VMware NSX Advanced Load Balancer Administration Guide

VMware NSX Advanced Load Balancer 22.1



vmware by Broadcom

You can find the most up-to-date technical documentation on the VMware by Broadcom website at:

https://docs.vmware.com/

VMware by Broadcom
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2023 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to https://www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

About This Guide 8

1 Web Interface Access Settings 9

2 CLI Access Settings 12

- Access the Controller CLI 12
- CLI Options 13
  - Install the NSX Advanced Load Balancer CLI Shell 17
  - CLI - Linux Command Line Mode 20
  - CLI - Script Mode 22
  - CLI Top-Level Commands 23
  - Common NSX Advanced Load Balancer CLI Tasks 33
- CLI and Shell Prompts Available for NSX Advanced Load Balancer 39
- Access Using Remote Authentication 40
- Accessing NSX Advanced Load Balancer Linux CLI as a Non-Admin User Using an SSH client 40
  - Instructions 41
- Adding Required HMAC to the Allowed HMACs Lists using NSX Advanced Load Balancer UI 43
- Securing Management IP Access 43
- Login Banner and Message of the Day 50
- Certificates 52
  - Changing the Default Certificate of the Controller 52
  - Renew Default (Self-Signed) Certificates on NSX Advanced Load Balancer 53
  - Configure Stronger SSL Cipher 56
  - User Interface Inaccessible Due to Wrong Certificate 57
  - Import Encrypted Private Key to Install an SSL Certificate 58
  - Venafi Integration 60
  - Integrating Let's Encrypt Certificate Authority with NSX Advanced Load Balancer 63
  - Certificate Management Integration for CSR Automation 69
  - SSL Certificates 72

3 Licensing 86

- NSX Advanced Load Balancer Editions 86
  - NSX Advanced Load Balancer Enterprise with Cloud Services Edition 89
  - NSX Advanced Load Balancer - Enterprise Edition 89
  - NSX Advanced Load Balancer - Basic Edition 107
  - NSX Advanced Load Balancer Essentials for Tanzu 113

Terms of NSX Advanced Load Balancer Software License 117
Removing Trial Licenses from an NSX Advanced Load Balancer Controller 119
Updating License on each Node in an NSX Advanced Load Balancer Controller Cluster 120

4 Controller Settings 121
NTP/DNS Settings 121
Time synchronization for the Service Engine 128
Email-SMTP Settings 130

5 NSX Advanced Load Balancer Controller Cluster 134
Deploying an NSX Advanced Load Balancer Controller Cluster 136
Using the UI 136
Using the CLI 136
Deploying an NSX Advanced Load Balancer Controller Cluster (Additional) 137
Deploy SEs in Different Data Centers from Controllers 139
Clustering Patched Controllers 139
Controller Cluster IP 140
Cluster IP Advertisement 141
OpenStack Controller Cluster 141
Configure the Cluster IP Using the UI 144
Assigning NSX Advanced Load Balancer Controller IP in Different Deployments 144
Cluster Configuration with DNS Hostnames 146
Changing NSX Advanced Load Balancer Controller Cluster Configuration 147
Remove both Followers (Dismantling the Cluster) Using the UI 147
Change a Follower Node Using the UI 148
Replace the Leader Node 150
Replace a Follower Node with a New Node 150
Updating the Configuration Following NSX Advanced Load Balancer Controller IP Address Change 151
Configure the Script 151
Updating IP Information for a Single-node Deployment 152
Updating IP Information for an NSX Advanced Load Balancer Controller Cluster 152
Change Controller IPs on Nutanix Cluster 153
Customizing NSX Advanced Load Balancer Configuration 154
Clustering NSX Advanced Load Balancer Controllers of Different Networks 159
High Availability for NSX Advanced Load Balancer Controllers 160
Operation of NSX Advanced Load Balancer Controller HA 160
Converting a Single-Node Deployment to a 3-node Cluster 161
Control Plane High Availability 162
Operation of NSX Advanced Load Balancer Controller High Availability 162
Converting a Single-Node Deployment to a Three-node Cluster 164
Switching Among Controllers Using the Controller Site Selector 166

Controller Cluster Deployment Across Two Availability Zones 170
Impact of a Controller Failure 172
Recover a Non-Operational Controller Cluster 173
API - Configuring the NSX Advanced Load Balancer Controller Cluster 174
NSX Advanced Load Balancer ControllerCluster Configuration in AWS 177
Configure Azure Cluster IP 178
Changing Controller IP Addresses in a Bare-metal Environment 179
OpenStack Network Configuration for NSX Advanced Load Balancer Controller Cluster 180
FAQs on Controller Cluster 184
FAQs on Database Replication 187

6 Tenant Settings 190
Create a Tenant 191
Add an Existing User to a Tenant 192
Enable Use Single Role for All Tenants 192
Configuring Tenant Settings 192
Using the UI 193
Using the CLI 193
Switching Between Tenants 194
Sharing Certificates Across Tenants 195
Tenant-Scoped Clouds 204
All Tenants View 206
Using the UI 206
Using the CLI 206
Using the API 206
Tenants Versus SE Group Isolation 207
Sharing Admin Profiles Across Tenants 208

7 User Authentication and Authorization 210
Auth Profile 210
LDAP Authentication Profile 211
TACACS+ Authentication 221
Enabling SAML Authentication in NSX Advanced Load Balancer 233
Configure SAML with Workspace One for NSX Advanced Load Balancer 238
NSX Advanced Load Balancer Go SDK Support 250
User Roles 255
Manage User Roles 255
Create a Role 255
Assign a Role to a Local User Account 258
LDAP or TACACS+ Users to Roles 259
Extended Granular RBAC 260

Granular Role-Based Access Controls Per Field 266
Tenant and Role Mapping Examples 268
Maximum Concurrent Login Sessions 302
Configure No-Lockout-User-Account-Profile on NSX Advanced Load Balancer 303
User Accounts 305
Default System Accounts 311
Create a User Account 312
User Account Security 313
User Account Self-Service 314
User Account Lockout 315
User Credentials Timeout 316
Strong Default Admin Password 317
Strong Password Enforcement 318
Password History Enforcement 319
User Profile 322

8 NSX Advanced Load Balancer Controller Operations 324
Backup and Restore of NSX Advanced Load Balancer Configuration 326
Backing Up the NSX Advanced Load Balancer Configuration 326
Restore the VMware NSX Advanced Load Balancer Configuration 331
Upgrade and Patches 333
Upgrade Overview 333
Patch Overview 334
Prerequisites 334
Determine Software Version 338
Pre/Post Upgrade Object Operational State Verification 338
Upgrade Guide 344
Upgrading SE Group 362
Patch Guide 367
Rollback 382
Rollback - Patch 383
Reset to Factory Defaults 385
Locking a Linux System to a Specific OS Version 386
Increasing Docker Container Size 388

9 Migrating from F5 BIG-IP to NSX Advanced Load Balancer 391

10 Testing NSX Advanced Load Balancer with Load Generators 395

11 Maintaining NSX Advanced Load Balancer Controller Single Node Cluster Availability using VMware 397

12 NSX Advanced Load Balancer SaaS 404

Configuring NSX Advanced Load Balancer SaaS Controller Using Service Account 409

Accessing the NSX Advanced Load Balancer SaaS REST API 411

Accessing the NSX Advanced Load Balancer SaaS CLI Using SSH 413

About This Guide

The NSX Advanced Load Balancer Administration Guide provides information about configuring and managing networking for VMware NSX Advanced Load Balancer. This guide helps you set up user accounts and manage roles and permissions, and understand how to protect the network from unauthorized users. This guide also discusses how to control, manage, and monitor applications within NSX Advanced Load Balancer and control user access to objects. Further, you can schedule periodic backups and restore data in case of an unlikely disaster.

Intended Audience

This information is intended for administrator users who want to configure NSX Advanced Load Balancer. The information is written for experienced system administrators who are familiar with virtual machine technology, networking, and security operations.

VMware Technical Publications Glossary

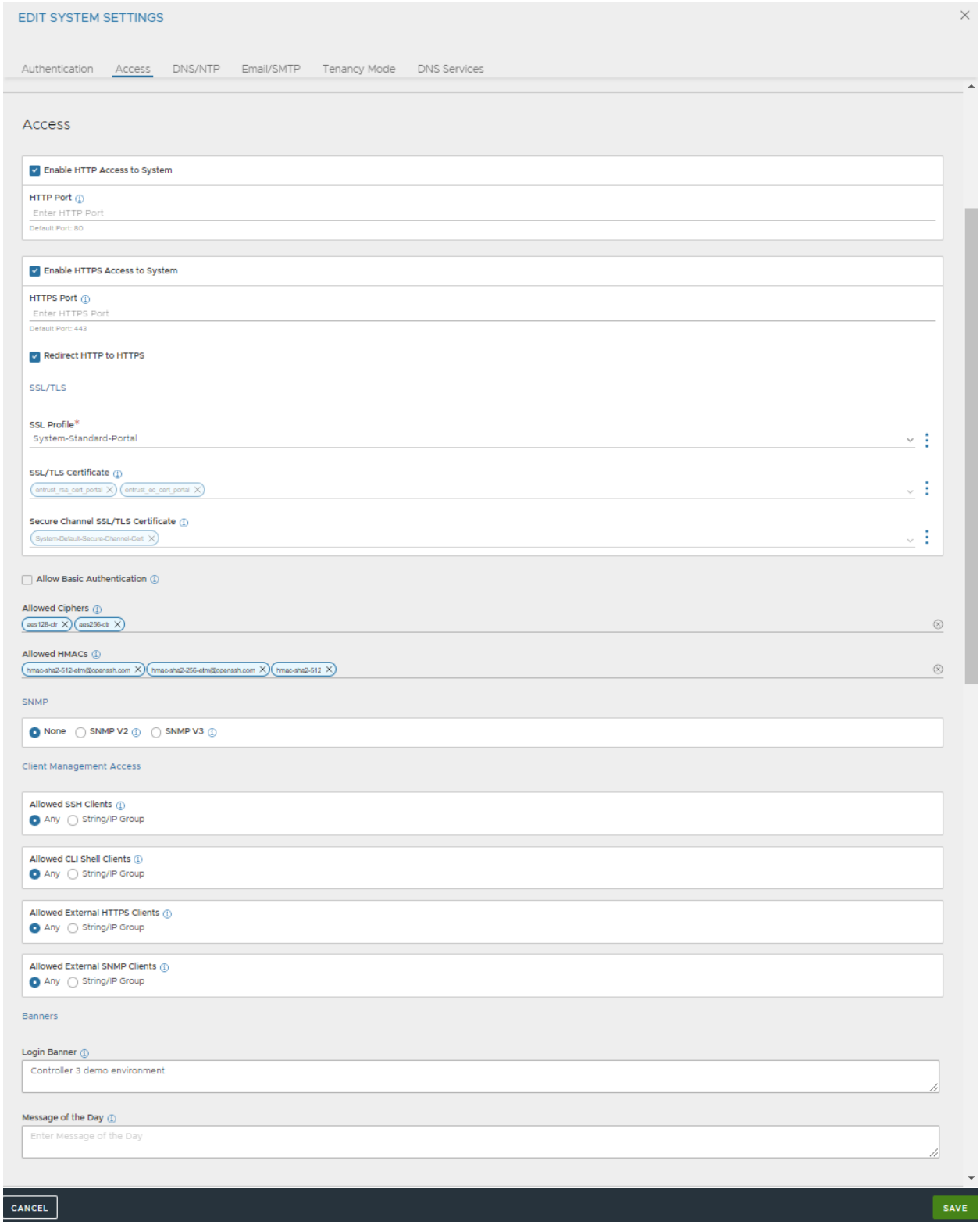
VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, see VMware Glossary.

Web Interface Access Settings

1

Access Settings allows users to view or edit settings mainly related to accessing the NSX Advanced Load Balancer Controller from the outside.

Navigate to **Administration > System Settings > EDIT > Access**.



The following options are classified as System Access Settings:

- **Enable HTTP Access to System**: It allows HTTP access to the NSX Advanced Load Balancer web interface and REST API. This option is insecure and not recommended.

- **Enable HTTPS Access to System**: Enables SSL/TLS access to NSX Advanced Load Balancer GUI and REST API. When the option is enabled, the **SSL Profile** and **SSL/TLS Certificate** fields must be populated.
- **Redirect HTTP to HTTPS**: When HTTP Access to System is disabled, enabling this option will automatically redirect administrators to the HTTPS interface for the web interface and API.
- **SSL Profile**: Select an **SSL Profile** to complete the HTTPS Access. This profile is from **Templates > Security > SSL Profiles**, which is also referenced by SSL-enabled virtual services.
- **SSL/TLS Certificate**: Select an SSL certificate from **SSL/TLS Certificate** drop-down menu to present to clients connecting to the web interface. RSA and Elliptic Curve (EC) are supported.
- **Allow Basic Authentication**: Uses HTTP to prompt the NSX Advanced Load Balancer user for a username and password and to return the values to NSX Advanced Load Balancer for authentication and authorization.
- **Allowed Ciphers**: List of the ciphers supported for HTTP basic authentication.
- **Allowed HMACs**: List of the HMACs supported for HTTP basic authentication.

The following option is related to SNMP:

- **SNMP**: Select **None**, **SNMP V2**, or **SNMP V3** as required. The string to be furnished by an external SNMP v2c manager wishing to query the SNMP daemon running on the NSX Advanced Load Balancer Controller leader. For more information, see SNMP Support in NSX Advanced Load Balancer.

The **Client Management Access** to NSX Advanced Load Balancer Controller lists four different client types that are authorized system users:

1 SSH Clients

2 CLI Shell Clients

3 External HTTP(S) Clients

4 External SNMP Clients

Note Enter the controller management IPs for HTTP(s) settings using the field Allowed External HTTP(S) Clients. Internal Analytics APIs will fail if the management IPs of the cluster nodes are not included in the list of IPs allowed for external HTTP(s).

Each one can flexibly specify clients by IP address and/or string/IP groups.

The following options govern two Banners that NSX Advanced Load Balancer will display if set:

- Message of the Day: Displayed to users after a successful login, be it through the UI, CLI, or SSH.
- Login Banner: Displayed before logging in through SSH or UI.

CLI Access Settings

2

When accessing the NSX Advanced Load Balancer CLI, an administrator needs to SSH through port 22 to the IP address of an NSX Advanced Load Balancer Controller or the cluster IP.

NSX Advanced Load Balancer Controller has two levels of CLI access. The first level of access is when the administrator connects to the NSX Advanced Load Balancer Controller and logs in, and they are granted access to a Linux Bash shell. The admin can then access the NSX Advanced Load Balancer CLI shell through a procedure in the Access the Controller CLI section. In some authentication modes, non-admin accounts cannot access Linux, and are instead forwarded directly to the shell.

Read the following topics next:

- Access the Controller CLI
- CLI Options
- CLI and Shell Prompts Available for NSX Advanced Load Balancer
- Access Using Remote Authentication
- Accessing NSX Advanced Load Balancer Linux CLI as a Non-Admin User Using an SSH client
- Adding Required HMAC to the Allowed HMACs Lists using NSX Advanced Load Balancer UI
- Securing Management IP Access
- Login Banner and Message of the Day
- Certificates

Access the Controller CLI

Local admin accounts default to Linux bash. In this case, enter the NSX Advanced Load Balancer shell by typing `shell`.

```
username@avi:~$ | shell
```

To exit the shell, enter in to Linux, type `exit`.

```
: > | bash
```

To switch between Linux Bash and Shell, type `exit`.

```
username@avi:~$ | exit
```

Note The SE can be accessed using CLI. However, this is not recommended except for troubleshooting.

CLI Options

NSX Advanced Load Balancer can be managed through GUI, RESTful API, or CLI. Both the GUI and CLI are built on top of the API, which means every CLI command maps to a corresponding API call (or calls) to be run.

Interfacing to NSX Advanced Load Balancer through CLI Commands

In NSX Advanced Load Balancer, there are three entities with which the authorized user can have a CLI-based conversation:

1 A Controller's Linux operating system

2 NSX Advanced Load Balancer processes running on a Controller

3 An SE's Linux operating system

Conversations with the first two entities are common. Conversations of the third kind are infrequent and typically undertaken by customer support personnel for troubleshooting purposes.

CLI Conversations with the Controller's Operating System

Below is a command-line example:

The one argument (user@hostname) passed to the `ssh` command is a combination of admin and the Controller's IP address, `10.144.130.195`. Every NSX Advanced Load Balancer Controller recognizes the admin user; others can be defined if necessary. The response to the password prompt is not echoed. However, in this example, it is represented by XXXXX.

The bash command-line interpreter gives access to the Controller's underlying operating system and file system. One use case would be to analyze various logs in the `/opt/avi/log` and `/var/log/upstart` directories.

```
$>ssh admin@100.65.9.10
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
Management:   100.65.9.10/20        UP
Gateway:      100.65.15.254         UP
admin@100.65.9.10@100.65.9.10
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
```

```
Management:   100.65.9.10/20        UP
Gateway:      100.65.15.254         UP
admin@100.65.9.10@100.65.9.10
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
Management:   100.65.9.10/20        UP
Gateway:      100.65.15.254         UP
admin@100.65.9.10's password:
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Wed Apr  5 16:24:41 2023 from 100.65.1.5
admin@100-65-9-10:~$
```

CLI Conversations with NSX Advanced Load Balancer Processes Running on the Controller

NSX Advanced Load Balancer-specific commands are not directly accessible from the Controller's bash interface.

Enter the NSX Advanced Load Balancer shell command-line interpreter by typing the `shell` command in response to the bash prompt and input the login credentials as follows:

```
shell
Login: admin
Password: XXXXX
```

In response to the shell prompt, press the Tab key twice to reveal the commands specific to NSX Advanced Load Balancer. In the CLI snippet shown below, `TABTAB` represents the Tab key pressed twice.

```
TABTAB
attach          forcedelete     reprogram       terminal
clear           gslb            resync          test
configure       import          retryplacement  upgrade
controller      migrate         rollback        upload
convert         nsx             rotatekeys      upload_to_avi
debug           passwd          scalein         verifylogin
delete          reboot          scaleout        vinfra
do              rediscover      show            watch
exec            redistribute    switchover
export          reimage         switchto
```

The `show` command helps analyze various issues and collect information. For example, `show virtualservice my-vs` displays essential data about the virtual service named `my-vs`.

For insight into what the aforementioned commands do, see CLI Top-Level Commands.

To exit the NSX Advanced Load Balancer shell and return to the Controller's bash prompt, type `bash`.

```
bash
```

CLI Conversations with an NSX Advanced Load Balancer SE's Linux Operating System

Note While it is possible to access a SE's CLI directly, it must only be used for basic troubleshooting. All configuration management must instead be done from and by the Controller.

SSH'ing to a SE or running the `attach serviceengine name-of-service-engine` command from the NSX Advanced Load Balancer shell places one in the SE's Linux CLI. There is no equivalent to the Controller's shell prompt; there are no `show` commands in the SE CLI. Use the SE's CLI to look into SE-specific logs in various directories, such as `/opt/avi/log`.

Navigation and Help

After entering into the NSX Advanced Load Balancer shell, press the TAB key twice to see a list of available commands.

- While typing a command, pressing the TAB key auto-completes the command. Double TAB returns a list of available options for the command in the left column. Most options include a brief help description, which is shown in the right column.

  ```
  export configuration
  export configuration serviceengine
  export serviceengine ova file from controller virtualservice
  export virtual service
  ```

- Commands or parameters might require multiple words or options. If there is only a single word or option, pressing TAB auto-completes the next word in the command.

  ```
  export configuration [TAB]
  export configuration file [TAB]
  WORD (required)
  export configuration file mybackup
  Completed writing the export configuration to mybackup
  ```

Other navigational commands:

- The up-arrow key cycles through and enables the reuse of previously run commands.
- The history command presents commands in a list format.
- Pipe filters results, as in
  - | grep address
  - | more

These are useful in watch commands too.

Sub-mode Navigation

Many NSX Advanced Load Balancer CLI commands contain sub-modes, which are nested sub-sections of the current command.

To enter the sub-mode, enter the relevant command. Within the context of a sub-mode, changes are not committed until explicitly saved. Type `save` to exit the sub-mode while committing changes.

To exit the sub-mode without saving changes, type `cancel`. When in a sub-mode or a nested sub-mode, the command prompt will change to reflect the current sub-mode.

```
debug virtualservice Test-VS
debug_ip
cancel
cancel
```

It is possible to enter a command which enters a sub-mode while also adding applicable flags. This will simultaneously navigate into the sub-mode and run the command. Subsequent commands within the sub-mode do not use the initial sub-mode command.

```
debug_ip addrs 10.1.1.1
addrs 10.1.1.2
save
```

The where Command

When operating within a sub-mode, multiple changes can be made to parameters. To see the current status of the configured parameters, use the `where` command.

```
debug_ip addrs 10.1.1.1
addrs 10.1.1.2
addrs 10.1.1.3
where
Tenant: admin
+----------+----------+
| Field    | Value    |
+----------+----------+
| addrs[1] | 10.1.1.1 |
| addrs[2] | 10.1.1.2 |
| addrs[3] | 10.1.1.3 |
+----------+----------+
```

Revealing the REST API Calls Behind CLI Commands

Any NSX Advanced Load Balancer CLI command may include the `--api-detail` flag, which echoes the API call (or calls) the command is performing. The command runs as it would without this flag. This can be useful when building API-driven automation scripts.

```
show serviceengine --api-detail
REST API Request
API: /api/serviceengine?owned_by_controller=True&join_subresources=runtime
```

API-echoed output may be enabled for every command run during a single CLI session by typing the terminal `display_api_details` command, as shown below:

```
terminal display_api_details
show serviceengine
REST API Request
Method: GET

API: /api/serviceengine?owned_by_controller=True&join_subresources=runtime

+--------------------------+---------------+------------------+---------------+------------+
| Name                     | SE Group      | Mgmt IP          | Cloud         | Oper State |
+--------------------------+---------------+------------------+---------------+------------+
| se3                      | glsbSEG       | 192.168.38.56    | Default-Cloud | OPER_UP    |
| se1                      | Default-Group | 192.168.38.52    | Default-Cloud | OPER_UP    |
| se2                      | Default-Group | 192.168.38.53    | Default-Cloud | OPER_UP    |
+--------------------------+---------------+------------------+---------------+------------+
```

Install the NSX Advanced Load Balancer CLI Shell

NSX Advanced Load Balancer can be managed through the web interface, REST API, or command-line interface (CLI). This section explains how to install the CLI shell onto a client's PC.

The CLI shell provides access to the NSX Advanced Load Balancer Controller through a PC client version of the Controller's CLI. Two versions of the CLI shell installation package are available:

- avi_shell-18.2.1-9010.tar.gz (or later) – Can be used with all infrastructure types. For installing this version of the CLI shell package, continue referring to the later Install the CLI Shell on a Ubuntu Docker Container.
- avi_lbaas-18.2.1-9010.tar.gz (or later) – Can be used if the infrastructure type is OpenStack and Keystone support is enabled. (This option is chosen during the initial NSX Advanced Load Balancer Controller setup and can also be configured later). To install this version of the CLI shell, see *Installing LBaaS Driver CLI Shell for OpenStack* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.

Note The CLI packages are available on the VMware customer portal, below the NSX Advanced Load Balancer Controller download option for each version.

For more information on VMware's customer portal, see VMware CUSTOMER CONNECT.

Requirements to Enable Remote CLI Shell

CLI shell server listens on TCP port 5054. To use the remote CLI shell, port 5054 must be permitted in the firewall rules between the CLI client and the NSX Advanced Load Balancer Controller.

OS Versions Supported

Versions of the CLI shell available for Linux and Mac are as follows:

- Linux Ubuntu Docker container
- Linux (not in a Docker container)
- Mac

The steps are the same for each OS.

Prerequisites

The NSX Advanced Load Balancer CLI shell requires the following software:

1 pip (install package manager for Python).

2 Virtual environment (virtualenv): Command syntax included below.

3 NSX Advanced Load Balancer CLI shell installation file: From AWS S3.

The following sections provide steps for installing the NSX Advanced Load Balancer CLI shell.

Install the CLI Shell on a Ubuntu Docker Container

To install the NSX Advanced Load Balancer CLI shell on a Ubuntu Docker container, download the shell package onto the host and enter the following commands. Edit the "`/tmp`" in "`/tmp ubuntu`" to the directory where the image is downloaded.

```
docker run -it -v /tmp:/tmp ubuntu
   sudo apt-get update
   sudo apt-get install python-pip
```

Log in to the CLI Shell

Log in to the NSX Advanced Load Balancer CLI shell as shown below.

```
avi_shell --address 10.10.10.99
 Login: admin
 Password *****
```

The IPADDR is the Controller's IP address (10.10.10.99 in this example).

After logging in, NSX Advanced Load Balancer CLI commands can be entered into the shell. The `show version controller` command in the following example displays the NSX Advanced Load Balancer version:

```
show version controller
+-----------------+----------------------------------------+
| Controller Name | Version                                |
+-----------------+----------------------------------------+
| 10.10.25.44     | 18.2.1 (9010) 2019-12-03 22:42:48 UTC |
+-----------------+----------------------------------------+
```

Leave the CLI Virtual Environment

Exit the CLI shell virtual environment as shown below.

```
deactivate
```

Restart the CLI Shell

After the CLI shell is installed, enter the following command to start it the next time.

```
 $> avi_shell/bin/avi_shell

show version controller
+-----------------+----------------------------------------+
| Controller Name | Version                                |
+-----------------+----------------------------------------+
| node-1          | 18.2.1 (9010) 2019-12-04 16:45:38 UTC |
| node-2          | 18.2.1 (9010) 2019-12-04 16:45:38 UTC |
| node-3          | 18.2.1 (9010) 2019-12-04 16:45:38 UTC |
+-----------------+----------------------------------------+
```

Install the CLI Shell on Linux or Mac

This section provides steps for installing the NSX Advanced Load Balancer CLI shell onto the Linux or Macintosh. The steps are the same for either OS.

Procedure

1. Install `pip`, if not already installed.
2. Install `virtualenv`, if not already installed.

```
sudo pip install virtualenv
 Downloading virtualenv-14.0.6-py2.py3-none-any.whl (1.8MB)
 100% |████████████████████████████████| 1.8MB 178kB/s
 Installing collected packages: virtualenv
 Successfully installed virtualenv-14.0.6
```

3 Create a virtual environment for the CLI shell.

```
virtualenv avi_shell
 New python executable in /home/user/git/clean/avi-dev/build/avi_shell/bin/python
 Installing setuptools, pip, wheel...done.
```

4 Go to the CLI shell virtual environment.

```
cd avi_shell/
 source ./bin/activate
```

5 Install the CLI shell package.

```
pip install /tmp/avi_shell-18.2.1-9010.tar.gz
 Processing /tmp/avi_shell-18.2.1-9010.tar.gz
 Collecting cmd2==0.6.8 (from shell-client==18.2)
 Collecting iso8601==0.1.11 (from shell-client==18.2)
 Using cached iso8601-0.1.11-py2.py3-none-any.whl
 ...
 ...
 Successfully installed cmd2-0.6.8 commentjson-0.6 iso8601-0.1.11 prettytable-0.7.2
pyparsing-2.1.0 pytz-2015.7 requests-2.9.1 requests-toolbelt-0.5.1 shell-client-17.2.17
urllib3-1.14 virtualenv-13.1.2 wheel-0.26.0 wrapt-1.10.6
```

Note For installing a CLI shell to manage an OpenStack write access mode deployment with Keystone support enabled, see *Install the LBaaS CLI Shell (OpenStack with Keystone)* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

CLI - Linux Command Line Mode

The CLI mode, `linux_command_line` allows the user to interact with the CLI similar to a traditional Linux command line program, where all the arguments are presented in `--parameter value` format. Multi-line configuration of object uses a Bash heredoc style.

The CLI enhancements are introduced with the following considerations:

- Easy automation using the CLI
- Search for specific keywords in the configuration using the CLI
- Quick cut-and-paste to make modifications or resolve some issues with the CLI

In the `linux_command_line` mode, the entire object is presented in the form of `--parameter value`, where each parameter represents a field in the object specified in a fully qualified path. An example of this is:

IP address of a server in a pool is specified as `--servers.index.ip`. This is equivalent to referring to this attribute as pool['servers'][index]['ip'].

```
terminal mode linux_command_line
terminal --linux_cmd_output single_line
show pool p2
--uuid pool-b0cb56dc-cc24-4b87-9d19-7bf790d2e582 --default_server_port
```

```
80 --graceful_disable_timeout 1 --connection_ramp_duration 10 --
max_concurrent_connections_per_server 0 --servers.1.ip 1.1.1.1 --servers.1.hostname
1.1.1.1 --servers.1.enabled True --servers.1.ratio 1 --servers.1.verify_network
False --servers.1.resolve_server_by_dns False --servers.1.static False --
servers.1.rewrite_host_header False --servers.2.ip 2.2.2.2 --servers.2.hostname
2.2.2.2 --servers.2.enabled True --servers.2.ratio 1 --servers.2.verify_network
False --servers.2.resolve_server_by_dns False --servers.2.static False --
servers.2.rewrite_host_header False --servers.3.ip 3.3.3.3 --servers.3.hostname
3.3.3.3 --servers.3.enabled True --servers.3.ratio 1 --servers.3.verify_network
False --servers.3.resolve_server_by_dns False --servers.3.static False
--servers.3.rewrite_host_header False --server_count 3 --lb_algorithm
lb_algorithm_least_connections --inline_health_monitor True --use_service_port False
--capacity_estimation False --server_auto_scale False --vrf_ref global --
fewest_tasks_feedback_delay 10 --enabled True --request_queue_enabled False
--request_queue_depth 128 --host_check_enabled False --sni_enabled True --
rewrite_host_header_to_sni False --rewrite_host_header_to_server_name False --tenant_ref
admin --cloud_ref Default-Cloud
```

Note The index starts with 1 in the CLI but is converted to 0 based on the API interactions.

Configure using the Linux Command Line Mode

```
configure pool p2 --uuid pool-7fce112f-c340-4c31-8c65-95e52f4b85ba --servers.1.ip 3.3.3.3 --
servers.1.hostname 3.3.3.3 --servers.2.ip 2.2.2.2 --servers.2.hostname 2.2.2.2 --servers.3.ip
1.1.1.1 --servers.3.hostname 1.1.1.1 --servers.4.ip 4.4.4.4 --servers.4.hostname 4.4.4.4 --
server_count 4 --tenant_ref admin
Updating an existing object.
p2 --uuid pool-7fce112f-c340-4c31-8c65-95e52f4b85ba --servers.1.ip 3.3.3.3 --
servers.1.hostname 3.3.3.3 --servers.2.ip 2.2.2.2 --servers.2.hostname 2.2.2.2 --servers.3.ip
1.1.1.1 --servers.3.hostname 1.1.1.1 --servers.4.ip 4.4.4.4 --servers.4.hostname 4.4.4.4 --
server_count 4 --tenant_ref admin
```

Some objects are relatively large, so interacting with the CLI with all these parameters specified in a single line becomes cumbersome. To address this, use `multi_line`, the CLI setting to change the Linux command output. This enables the CLI interaction one parameter per line.

Consider the example given below:

```
terminal --linux_cmd_output multi_line

show pool p2
--uuid pool-7fce112f-c340-4c31-8c65-95e52f4b85ba
--servers.1.ip 3.3.3.3
--servers.1.hostname 3.3.3.3
--servers.2.ip 2.2.2.2
--servers.2.hostname 2.2.2.2
--servers.3.ip 1.1.1.1
--servers.3.hostname 1.1.1.1
--server_count 3
--tenant_ref admin

configure pool p2 << END
--uuid pool-7fce112f-c340-4c31-8c65-95e52f4b85ba
```

```
--servers.1.ip 3.3.3.3
--servers.1.hostname 3.3.3.3
--servers.2.ip 2.2.2.2
--servers.2.hostname 2.2.2.2
--servers.3.ip 1.1.1.1
--servers.3.hostname 1.1.1.1
--servers.4.ip 4.4.4.4
--servers.4.hostname 4.4.4.4
--tenant_ref admin
END

Updating an existing object.
--uuid pool-7fce112f-c340-4c31-8c65-95e52f4b85ba
--servers.1.ip 3.3.3.3
--servers.1.hostname 3.3.3.3
--servers.2.ip 2.2.2.2
--servers.2.hostname 2.2.2.2
--servers.3.ip 1.1.1.1
--servers.3.hostname 1.1.1.1
--servers.4.ip 4.4.4.4
--servers.4.hostname 4.4.4.4
--server_count 4
--tenant_ref admin
```

CLI - Script Mode

In `script` mode, the input and output of the CLI command are expected to be in YAML format. This is a direct conversion of what is presented in the API input and output as JSON into YAML.

YAML was chosen for presentation as it allows an easier interaction without having to consider issues about the syntax such as commas, quotes, curly braces, and so on. It is easier to work with YAML primarily for cut-and-paste and incremental changes to an existing object. Multi-line configuration of object uses a Bash heredoc style. You can find more information on this under https://en.wikipedia.org/wiki/Here_document.

```
terminal mode script
show pool p1
name: p1
server_count: 3
servers:
- hostname: 1.1.1.1
ip:
addr: 1.1.1.1
type: V4
- hostname: 2.2.2.2
ip:
addr: 2.2.2.2
type: V4
tenant_ref: https://localhost/api/tenant/admin
uuid: pool-b0cb56dc-cc24-4b87-9d19-7bf790d2e582
```

Configure a Pool using the Script Mode

```
configure pool p1 << END
name: p1
server_count: 3
servers:
- hostname: 1.1.1.1
ip:
addr: 1.1.1.1
type: V4
- hostname: 2.2.2.2
ip:
addr: 2.2.2.2
type: V4
- hostname: 3.3.3.3
ip:
addr: 3.3.3.3
type: V4
tenant_ref: https://localhost/api/tenant/admin
uuid: pool-b0cb56dc-cc24-4b87-9d19-7bf790d2e582
END
Updating an existing object
name: p1
server_count: 3
servers:
- hostname: 1.1.1.1
ip:
addr: 1.1.1.1
type: V4
- hostname: 2.2.2.2
ip:
addr: 2.2.2.2
type: V4
- hostname: 3.3.3.3
ip:
addr: 3.3.3.3
type: V4
tenant_ref: https://localhost/api/tenant/admin
uuid: pool-b0cb56dc-cc24-4b87-9d19-7bf790d2e582
```

CLI Top-Level Commands

This section elaborates the top-level commands and their application.

Following are top-level commands shown when pressing the Tab twice from the shell:

Command	Description
attach	Connect to a remote Controller or SE. Similar to SSH.
clear	Clear the statistics or value of a designated object.
configure	Create or modify a new or existing object, such as a virtual service, pool, profile, and so on.

Command	Description
convert	Import and convert a configuration from non-VMware load balancers.
copy	Copy a file, such as a packet capture or tech-support file.
debug	Change debug settings or perform packet captures.
delete	Delete an object. Some objects may have dependencies which must be resolved first.
do	Execute a `show` command without exiting the current location or sub-mode.
export	Backup the system configuration or a single virtual service configuration.
import	Import a backed up (exported) complete or virtual service specific configuration.
initialplacement	Move a virtual service using manual placement mode to a different SE.
purge	Remove a file, such as a packet capture or tech-support file.
rebalance	Realign the SEs handing virtual services within an SE group.
reboot	Reboot part or all of the system. It can also wipe configuration.
rediscover	VMware-specific: Initiate discovery of networks and VMs.
restart	Deactivate, then re-activate a virtual service.
scalein	Reduce by one the number of SEs handling a virtual service in manual placement mode.
scaleout	Increase by one the number of SEs handling a virtual service in manual placement mode.
show	Show detailed information and stats on any NSX Advanced Load Balancer object.
switchto	Switch into a different tenant.
terminal	Alter the shell's terminal settings.
upgrade	Initiate an upgrade of the NSX Advanced Load Balancer system.
upload	Upload a specified tech-support debug file to VMware.
verifylogin	Validate login settings to a remote orchestrator such as vCenter, APIC, or OpenStack.
watch	Update the result of a command, such as `show`, every few seconds.

attach

Description	Connect to a remote Controller or SE. Similar to SSH.
Example	`Attach serviceengine Avi-se-arjnz`
Top Level Flags:	
controller	Attach to a Controller shell.
serviceengine	Attach to a SE shell.

configure

Description	Create or modify a new or existing object, such as a virtual service, Pool, or Profile.
Example	`: > configure pool Test`
Top-Level Flags:	
actiongroupconfig	Create or modify an Action Group Config.
alertconfig	Create or modify an Alert Config.
alertemailconfig	Create or modify an Alert Email Config.
alertscriptconfig	Create or modify an Alert Script Config.
alertsyslogconfig	Create or modify an Alert Syslog Config.
analyticsprofile	Create or modify an Analytics Profile.
application	Create or modify an Application.
applicationpersistenceprofile	Create or modify an Application Persistence Profile.
applicationprofile	Create or modify an Application Profile.
authprofile	Create or modify an Auth Profile.
cloud	Create or modify a Cloud.
cluster	Create or modify a Cluster.
controller	Create or modify Controller properties.
healthmonitor	Create or modify a Health Monitor.
httppolicyset	Create or modify an HTTP Policy Set.
ipaddrgroup	Create or modify an IP Address Group.
network	Create or modify a Network.
networkprofile	Create or modify a Network Profile.
networksecuritypolicy	Create or modify a Network Security Policy.
pkiprofile	Create or modify a PKI Profile.
pool	Create or modify a Pool.
role	Create or modify a Role.
serviceengine	Create or modify a Service Engine.
serviceenginegroup	Create or modify a Service Engine Group.
serviceengineproperties	Create or modify Service Engine properties.

sslkeyandcertificate	Create or modify an SSL Key and Certificate Request.
sslprofile	Create or modify an SSL Profile.
stringgroup	Create or modify a String Group.
systemconfiguration	Create or modify a System Configuration.
tenant	Create or modify a Tenant.
user	Create or modify a User.
virtualservice	Create or modify a Virtual Service.
vrfcontext	Create or modify a VRF Context.
vsdatascriptset	Create or modify a virtual service DataScript Set.

convert

Description	Import and convert a configuration from non-VMware load balancers. Supports conversion from F5 BIG-IP Local Traffic Manager configuration. Imported Virtual Services start in a disabled state to avoid IP conflicts.
Example	`convert bigip_configuration`
Top-Level Flags:	
bigip_ip_addr	BIGIP IP address
filename	NSX Advanced Load Balancer config file name.
password	BIGIP Password
username	BIGIP Username
virtualservername	Convert virtualserver. Name is of the form `/partition/virtualservername`

copy

Description	Copy a file, such as a packet capture or tech-support file.
Example	`copy file source/tmp/old destination /tmp/new`
Top-Level Flag:	
source	The source of the original file and path.
destination	The destination of the new file and path.

debug

Description	Change debug settings or perform packet captures.
Example	`: > debug virtualservice Test-VS`
Top-Level Flags:	
controller	Controller-specific debug options.
serviceengine	Service Engine-specific debug options.
virtualservice	Virtual service specific debug options, including packet capture.

delete

Description	Delete an object. Some objects may have dependencies that must be resolved first.
Example	`delete pool Test-Pool`
Top-Level Flags:	
actiongroupconfig	Delete Action Group Config.
alert	Delete Alert.
alertconfig	Delete Alert Config.
alertemailconfig	Delete Alert Email Config.
alertscriptconfig	Delete Alert Script Config.
alertsyslogconfig	Delete Alert Syslog Config.
application	Delete Application.
applicationpersistenceprofile	Delete Application Persistence Profile.
applicationprofile	Delete Application Profile.
authprofile	Delete Auth Profile.
cloud	Delete Cloud.
healthmonitor	Delete Health Monitor.
httppolicyset	Delete HTTP Policy Set.
ipaddrgroup	Delete IP Address Group.
networkprofile	Delete Network Profile.
networksecuritypolicy	Delete Network Security Policy.
pkiprofile	Delete PKI Profile.
pool	Delete Pool.
role	Delete Role.
serviceengine	Delete Service Engine.
serviceenginegroup	Delete Service Engine Group.
sslkeyandcertificate	Delete SSL Key and Certificate Request.
sslprofile	Delete SSL Profile.
stringgroup	Delete String Group.
tenant	Delete Tenant.
user	Delete User.
virtualservice	Delete Virtual Service.
vrfcontext	Delete VRF Context.
vsdatascriptset	Delete virtual service DataScript Set.

do

Description	Execute a show command without exiting the current location or sub-model.
Example	`do show debug flags virtualservice Test-VS`
Top-Level Flags:	
show	Show detailed information and stats for any NSX Advanced Load Balancer object.

export

Description	Back up the system configuration or a single virtual service configuration.
Example	`export configuration file /tmp/backup_config` `export virtualservice Test-VS file /tmp/Test-VS`
Top-Level Flags:	
configuration	Export the entire NSX Advanced Load Balancer configuration in JSON format.
serviceengine	Export the SE OVA file from Controller for manual install.
virtualservice	Export a specific virtual service configuration file including child objects.

import

Description	Import a backed up (exported) complete or virtual service specific configuration.
Example	`import configuration file /tmp/backup_config`
Top-Level Flags:	

initialplacement

Description	Move a virtual service using manual placement mode to a different SE.

Example	`initialplacement`
	`virtualservice Test-VS`
	`servicengine Avi-se-arjni`

Top-Level Flags:	
virtualservice	Specify the virtual service to be assigned to an SE.
serviceengine	Specify the name of the SE to receive the virtual service.

migrate

Description	Move a virtual service using manual placement (No Access or Read Access) mode to a different SE.

Example	`migrate`
	`virtualservice Test-VS`
	`serviceengine Avi-se-arjni`

Top-Level Flags:	
from_se_ref	Specify the name of the source SE that has the virtual service.
to_host_ref	An option used with `to_new_se`, specifying the host upon which to create the new SE.

to_new_se	Create a new SE and migrate to it.
to_se_ref	Migrate to a specific existing SE.

purge

Description	Remove a file, such as a packet capture or tech-support file.
Example	`purge file source /tmp/backup_config`
Top-Level Flags:	

rebalance

Description	In auto scale mode, sets the frequency upon which the Controller will Inspect an SE group to see if a virtual service to SE mapping mudt be adjusted, potentially resulting in a scale in, scale out, or migration.
Example	`rebalance interval 10 se_group_ref My_SE_Group`
Top-Level Flags:	
interval	The frequency, in minutes. Default is 5.
se_group_ref	The name of the SE group to alter.

reboot

Description	Reboot part or all of the NSX Advanced Load Balancer system. It can also delete configuration. With no flags specified, all Controllers and SEs are rebooted.
Example	`reboot`
Top-Level Flags:	
clean	Reset the NSX Advanced Load Balancer system's configuration and reboot the cluster. Consider making a backup first.
node	Reboot the virtual machine of a Controller within a cluster.
serviceengine	Reboot a specific SE. Virtual service disruption will depend on the high availability settings for the SE group.

rediscover

Description	VMware-specific: Initiate discovery of networks and VMs.
Example	`rediscover vcenter My-vCenter`
Top-Level Flags:	

restart

Description	Disable then immediately re-enable a virtual service.
Example	`restart virtualservice Test-VS`
Top-Level Flags:	

scalein

Description	Reduce by one the number of SEs handling a virtual service in manual placement mode. There must be a minimum of one SE.
Example	`scalein virtualservice Test-VS`
Top-Level Flags:	
from_se_ref	Specify a non-primary SE to stop using for this virtual service.
scalein_primary	Migrate from the primary SE and discontinue use of the SE for this virtual service.

scaleout

Description	Increase by one the number of SEs handling a virtual service in manual placement.
Example	`scaleout virtualservice Test-VS`
Top-Level Flags:	
to_host_ref	An option used with `to_new_se`, specifying where to create the new SE.
to_new_se	Create a new SE and scale out to it.
to_se_ref	Scale out to an existing SE.

show

Description	Show detailed information and stats on any NSX Advanced Load Balancer object.
Example	`show virtualservice Test-VS summary`
Top-Level Flags:	
actiongroupconfig	Show info on an Action Group Config.
alert	Show info on an Alert.
alertconfig	Show info on an Alert Config.
alertemailconfig	Show info on an Alert Email Config.
alertscriptconfig	Show info on an Alert Script Config.
alertsyslogconfig	Show info on the an Syslog Config.
analyticsprofile	Show info on an Analytics Profile.
apic	Show info on the APIC Graph Instances.
application	Show info on an Application Folder.
applicationpersistenceprofile	Show info on an Application Persistence Profile.
applicationprofile	Show info on an Application Profile.
authprofile	Show info on an Auth Profile.
backups	Show available backup files.
cloud	Show info on the Cloud.
cluster	Show info on the Cluster.
config-consistency-check	Show config-consistency-check status.

config_events	Show info on the Event Log.
configuration	Show configuration.
controller	Show Controller properties.
cpuusage	Show Controller CPU usage.
debug	Show Virtual Service capture file.
debug-log	Show Service Engine debugs.
diskusage	Show Controller disk usage.
events	Show info on an Event Log.
file	Show files.
healthmonitor	Show info on a Health Monitor.
httppolicyset	Show info on an HTTP Policy Set.
ipaddrgroup	Show info on an IP Address Group.
jobs	Show all duration based jobs pending expiry.
License	Show info on the Controller License.
logcontrollermapping	Show mapping of Log Controllers for each virtual service.
logs-status	Show logs subsystem status.
memoryusage	Show Controller memory usage.
metricsmgr	Show info on a Metrics Entity Runtime.
network	Show info on a Network.
networkprofile	Show info on a Network Profile.
networksecuritypolicy	Show info on a Network Security Policy.
openstack_audit	Show OpenStack LBaaS vs Avi config audit reports.
pkiprofile	Show info on a PKI Profile.
placement	Show info on a Rm VRF Proto.
pool	Show info on a Pool.
role	Show info on a Role.
serviceengine	Show info on a Service Engine.
serviceenginegroup	Show info on a Service Engine Group.
serviceengineproperties	Show the Service Engine Properties.
seupgrade	Show an ongoing SE upgrade status.
sslkeyandcertificate	Show info on an SSL Key and Certificate.
sslprofile	Show info on an SSL Profile.
stringgroup	Show info on a String Group.
systemconfiguration	Show info on the System Configuration.
systemconfigurationruntime	Show info on the System Configuration Runtime.
tech-support	Show full tech support.

tenant	Show info on a Tenant.
terminal	Show the terminal settings.
transaction	Show more info on Transaction Stats.
upgrade	Show upgrade status if one is in-progress.
user	Show info on the User.
vcenter	Show info on the specified VMs.
version	Show a Controller node's version.
vinfra	Show info on the VI Datastore Contents.
virtualservice	Show info on a Virtual Service.
vrfcontext	Show info on a VRF Context.
vsdatascriptset	Show info on the virtual service DataScript Set.

switchto

Description	Switch into a different tenant.
Example	`switchto tenant Tenant2`
Top-Level Flags:	

terminal

Description	Alter the shell's terminal settings.
Example	`terminal session_timeout 240`
Top-Level Flags:	
length	Number of rows to show for pagination output. Greater than will pipe to more. Choose `0` for no pagination.
session_timeout	Alter the default 15 min timeout to keep an idle terminal session open.
timezone	Display timestamps in specified time zone.
unhide	Commands show additional flags. It is not recommended for casual use.

upgrade

Description	Initiate an upgrade of the NSX Advanced Load Balancer system. This may be done passively (by migrating each SE while upgrading) or disruptively (fast, but it will terminate existing client connections then begin the upgrade).
Example	`upgrade system image_path /tmp/new_file`
Top-Level Flags:	
system	Upgrade the NSX Advanced Load Balancer system.
ui	Upgrade the NSX Advanced Load Balancer UI.

upload

Description	Generate and upload a tech-support debug file to VMware.
Example	`upload tech-support debuglogs filter exclude_logs`
Top-Level Flags:	
exclude_archive	Exclude archived backups.
exclude_logs	Exclude client log files (virtual service logs), which could be quite large.
include_se	Include (non-significant) logs may be stored on the SEs.

verifylogin

Description	Validate the username, password, and path to a remote orchestrator such as vCenter, APIC, or OpenStack.

Example	`verifylogin`
	`vcenter username`
	`admin password secret`
	`vcenter_url 10.1.1.1/login`

Top-Level Flags:	
apic	Verify access to an APIC controller.
openstack	Verify access to OpenStack.
vcenter	Verify access to a VMware vCenter controller.

watch

Description	Updates the result of a command such as `show` every few seconds.
Example	`watch show pool Test-pool server detail | grep -e 'ip| open_conns'`
Top-Level Flags:	
show	Select a valid show command syntax to update.

Common NSX Advanced Load Balancer CLI Tasks

This section illustrates examples of executing common management tasks with the NSX Advanced Load Balancer CLI.

CLI: Virtual Service and Pool Creation

During virtual service creation, a pool must be configured. This pool has to be created before creating the virtual service.

Create the following using the CLI:

1 The Pool

2 The VsVIP

Note The examples below shows only the minimum configuration required to successfully deploy a new application. Many additional options are available for customizing virtual services and pools.

Example: Create the Pool

```
: > configure pool Test
: pool> servers ip 10.1.1.100 port 80
: pool:servers> save
: pool> servers ip 10.1.1.101 port 80
: pool:servers> save
: pool> where
Tenant: admin
+------------+------------+
| Field      | Value      |
+------------+------------+
| name       | Test       |
| servers[1] |            |
| ip         | 10.1.1.100 |
| port       | 80         |
| servers[2] |            |
| ip         | 10.1.1.101 |
| port       | 80         |
+------------+------------+
: pool> save
```

Example: Create the VsVIP

```
: > configure vsvip Test_vsVip
: vsvip> vip ip_address 10.10.10.10
: vsvip:vip> save
: vsvip> save
```

Example: Create the Virtual Service

```
: > configure virtualservice Test_VS
: virtualservice> vsvip_ref Test_vsVip
: virtualservice> pool_ref Test
: virtualservice> services port 80
: virtualservice:services> save
: virtualservice> save
```

Modify Pool Servers

A common task is to add and remove servers from an existing pool or to activate or deactivate servers within the pool. While servers can be added to a pool without specifying a port (they will inherit the port of the pool or virtual service), the CLI requires the IP and port to manipulate the server for tasks such as enable and disable.

The following commands will delete the `.101` server and add a new `.102` server to the pool named Test.

```
configure pool Test
no servers ip 10.1.1.101 port 80
servers ip 10.1.1.102 port 80
[admin]: pool:servers save
```

The following commands will enter the sub-mode for the `.100` server and disable it.

```
ip 10.1.1.100 port 80
where
Tenant: admin
+-------+------------+
| Field | Value      |
+-------+------------+
| ip    | 10.1.1.100 |
| port  | 80         |
+-------+------------+
no enabled
+---------+------------+
| Field   | Value      |
+---------+------------+
| ip      | 10.1.1.100 |
| port    | 80         |
| enabled | False      |
+---------+------------+
save
```

Export / Import Configuration

The NSX Advanced Load Balancer Controller has a database to store its configuration information. This includes all configurations related to tenants, virtual services, pools, policies, and accounts. Export this configuration as a JSON file for backup or to migrate parts of the config to another NSX Advanced Load Balancer Controller.

The configuration may be backed up or moved to another Controller cluster through the CLI. The exported configuration might be the entire system configuration, a more limited version based on the access rights of the current user and tenant of the administrator, or a single virtual service and its child properties.

To export only a single virtual service and its child objects, such as pools, use the `virtualservice` flag instead of the `configuration` flag with the `export` command.

The following commands will export the configuration to a file named `config_export`, SCP it to a remote location, and then return it to the NSX Advanced Load Balancer shell.

```
export configuration file config_export
Completed writing the export configuration to config_export
bash

pwd
```

```
/home/admin
ls
config_export
scp ./config_export root@10.1.1.1:/root
root@10.1.1.1's password:
config_export 100% 232KB 431.8KB/s 00:00
exit
```

Before starting a configuration import:

- Ensure that all Controller members of the cluster are up and the cluster leader has the following configuration:
  - Admin Account
  - Cluster Configuration
  - OpenStack Infrastructure (OpenStack only)
- Edit the configuration file and remove the `_last_modified` field.

The following commands will import a backed-up configuration to a Controller cluster.

```
import configuration file /home/admin/myconfig keep_uuid
Successfully imported the configuration file
```

You can restore this configuration information to an empty NSX Advanced Load Balancer Controller. The restoration steps are:

- Deploy three new Controller nodes; the image type (OVA, qcow2, ami) should match that of the original Controller node from which the configuration was exported.
- Choose one Controller as the leader and go through the initial setup page to enter the initial setup information and the redundant Controller cluster information.
- Use the CLI or API to import the configuration.

After these steps, the cluster will return to the same running state as the original one.

Export Configuration Filters

You can also tune the exported file by using filters. The following filters are supported:

Filter Name	Description
full_system	Exports the complete configuration (including persistent runtime) in the format documented in Restore Configuration. This configuration cannot be imported through the CLI or the REST API.
obj_type	Limits exported objects to a specific set of object types.
recurse	Exports the entire tree of dependencies (including some runtime) for objects in filtered config.
passphrase	Encrypts sensitive data. The same passphrase must be provided to import the configuration.

Filter Name	Description
search	Only exports objects that contain a certain string.
skip_default	Prunes defaulted values of objects in exported configuration.
tenant_ref	Only exports objects in the specified tenant.

Example: Examples

To export all virtual services and dependencies, use the following command:

```
export configuration file config\_export obj\_type virtualservice recurse
```

To export all objects in tenant t1 and filter out defaulted values, use the following command:

```
export configuration file config\_export tenant\_ref t1 skip\_default
```

Packet Capture

The NSX Advanced Load Balancer Controller provides a convenient mechanism to capture data plane traffic processed by SEs. An administrator can initiate a `capture` command from the Controller CLI while the actual capture occurs on the SEs. The packet capture is saved in `pcap` format on the Controller. The SE captures packets on both the client-side and server-side of the SE. If a virtual service is scaled across multiple SEs, then all applicable SEs will participate in the packet capture. The Controller will aggregate the captures and sort the entries according to time. Once a capture is complete, it is uploaded to a personal computer or other systems for analysis with an appropriate tool such as tcpdump or Wireshark.

Enter the packet capture sub-mode for the specified virtual service:

```
debug virtualservice Test-VS
Updating an existing object. Currently, the object is:
+-------+--------------------+
| Field | Value              |
+-------+--------------------+
| uuid  | virtualservice-0-1 |
| name  | Test-VS            |
+-------+--------------------+
```

Parameters may be defined for the packet capture. By default, the capture is performed within the context of the selected virtual service, and it is also performed on all SEs handling the virtual service traffic, including the packets from the client and server sides.

Parameter Name	Description
`capture_params duration`	Time, in minutes. Default is unlimited.
`capture_params num_pkts`	Maximum number of packets to collect. Default is unlimited.

Parameter Name	Description
`capture_params pkt_size`	Packet size, or snap length, to capture. Default is unlimited.
`debug_ip addrs`	IP4 Address format <x.x.x.x>
`debug_ip prefixes`	IP4 Prefix format <x.x.x.x/x>

The `debug_ip` command enters a sub-mode. This allows multiple IP addresses or IP subnets to be entered (omit the `debug_ip` for subsequent entries). Save to commit the desired IPs and return to the previous menu.

Note By default, no maximum packets, or duration of time to be captured are defined. It is recommended to include a maximum packet capture, as shown in the following example. Without a limit, the capture will run until the SE disk is filled, potentially disrupting the service.

Specify parameters, including the maximum number of packets to capture:

```
capture_params num_pkts 1000
debug_ip addrs 10.10.10.10
save
```

Begin capturing based on the previously configured parameters:

```
capture
save
+----------------+--------------------+
| Field          | Value              |
+----------------+--------------------+
| uuid           | virtualservice-0-1 |
| name           | Test-VS            |
| debug_ip       |                    |
| addrs[1]       | 10.10.10.10        |
| capture        | True               |
| capture_params |                    |
| duration       | 0 mins             |
| num_pkts       | 1000               |
+----------------+--------------------+
```

Re-enter the packet capture sub-mode and stop an ongoing packet capture:

```
debug virtualservice Test-VS
no capture
save
```

Export the packet capture to a remote system to view it using tools such as tcpdump or Wireshark:

```
show debug virtualservice Test-VS capture
Please specify the destination directory: /tmp
Downloaded the attachment to /tmp/vs_virtualservice.20141205_192033.pcap
```

```
bash
scp /tmp/vs_virtualservice.192033.pcap user@10.1.1.1:/tmp
```

Monitor Concurrent Connections to a Server

To track the number of concurrent connections to a server, the server must be viewed within the context of a pool. By prepending the `watch` command in front of the `show`, it is possible to see a near real-time updates of the connection count. Keep in mind there is some delay between the time the connection is established and the polling interval when the SE sends information back to the Controller. Cancel the `watch` with `Ctrl+C`.

```
watch show pool Test-pool server detail | grep -e 'ip\|open_conns'
| ip_addr    | 10.1.1.17 |
| open_conns | 50        |
| ip_addr    | 10.1.1.16 |
| open_conns | 49        |
```

CLI and Shell Prompts Available for NSX Advanced Load Balancer

The authorized user may engage in three conversations through two command-line interfaces. Conversations with the first two entities are common. Conversations of the third kind are infrequent and typically undertaken by NSX Advanced Load Balancer Customer Support personnel for troubleshooting purposes.

In conversation	Your commands address	With this frequency
1	NSX Advanced Load Balancer Controller's Linux OS using Bash	Frequently Used
2	NSX Advanced Load Balancer processes running on Controller using the NSX Advanced Load Balancer shell	Frequently Used
3	The SE's Linux operating system	Rarely

Accessing NSX Advanced Load Balancer Controller using Linux Shell

- Access the NSX Advanced Load Balancer Controller's OS through the `bash` Linux CLI.
- Either SSH (Secure Shell) to the Controller or access it through the console from an orchestrator such as vCenter.
- Example: If the Controller's address is `10.144.130.195`, type the `ssh` command: `ssh admin@10.144.130.195`, and when prompted, supply the password for the admin user.

NSX Advanced Load Balancer Controller CLI is used to analyze various logs in the `/opt/avi/log and /var/log/upstart` directories.

Accessing NSX Advanced Load Balancer Controller using NSX Advanced Load Balancer Shell

- Start with conversation 1.
- To enter NSX Advanced Load Balancer-specific commands, type the shell command and furnish credentials.
- For example: `show virtualservice <name-of-virtual-service>`

Accessing NSX Advanced Load Balancer Controller using NSX Advanced Load Balancer Service Engines

- SSH to an NSX Advanced Load Balancer SE or execute the attached `service engine <name-of-service-engine>` command from the NSX Advanced Load Balancer shell to enter the SE's Linux CLI.
- Use it to look into SE-specific logs in various directories, such as `/opt/avi/log`.

Shell prompt access is not available for NSX Advanced Load Balancer Service Engines. NSX Advanced Load Balancer SE's Linux CLI does not provide the option to run `show` commands.

For more information, see CLI Options.

Access Using Remote Authentication

The following users' SSH into the Controller using local or remote authentication:

- Admin
- CLI

The admin account is the standard administrative account for the system and is maintained as a locally authenticated account, even in a system configured for remote auth.

The cli account has no password. Any non-admin account must use the account name cli, which will forward the user through Linux to the NSX Advanced Load Balancer shell. From the shell, the user must log in through their standard account. Non-admin accounts do not have access to Linux.

For more information on remote authentication, see Auth Profile.

Accessing NSX Advanced Load Balancer Linux CLI as a Non-Admin User Using an SSH client

When a non-admin user logs in to the NSX Advanced Load Balancer Linux CLI, the non-admin user can have admin-like privileges.

An SSH session to Linux CLI is available only for admin username. Even if a user is configured as a super-user, the user cannot log in to Linux CLI. Users other than admin, including super-users (whether local or remote), can only log in using `cli@<Avi Controller IP>` command.

If a non-admin user, even if it is configured as a super-user, tries to SSH to NSX Advanced Load Balancer Controller IP address, the system will return an Access Denied error, as shown below.

```
login as: testuser
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
Version: 17.1.8
Date: 2017-09-21 06:03:07 UTC
Build: 9020
Management: 10.10.1.1/23 UP
Gateway: 10.10.1.10 UP
Esx and Openstackgslb@10.10.1.1's password:
Access denied
testuser@10.10.30.55's password:
```

Instructions

Follow the steps to SSH into NSX Advanced Load Balancer CLI using a non-admin user account.

In this example, the non-admin user is configured as a super-user too.

- Open an SSH client and use the `cli@<Avi Controller IP>` command. Replace the NSX Advanced Load Balancer Controller IP with the IP of the Controller for which access is required.
- Provide the credentials when prompted for a username. In the below example, a user account with the username **`testuser`** is used, which is also configured as a super-user on NSX Advanced Load Balancer.

```
Using username "cli".
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
Version: 17.1.8
Date: 2017-09-21 06:03:07 UTC
Build: 9020
Management: 10.10.1.1/23 UP
Gateway: 10.10.1.1 UP
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Fri Oct 27 10:32:02 2017 from 10.10.8.11
Launching a CLI shell in a container
No handlers could be found for logger "docker.auth.auth"
```

```
Login: testuser
Password:
```

- After providing the password, as shown in the above CLI snippet, you can get access to the NSX Advanced Load Balancer shell.

```
[admin:avi-controller]: >
```

From the NSX Advanced Load Balancer shell prompt, you can run all the show commands and shell commands.

Checking Logs Using a Super-User Account

Use the account mentioned in the previous steps and use the `attach controller <controller-ip>` command to go to the `linux bash` prompt. As it is a container with no persistent storage, none of the log files are visible when the `ls` command is used.

```
[admin:avi-controller]: > bash
root@04de723c268a:/#
root@04de723c268a:/# cd /opt
root@04de723c268a:/opt# ls
root@04de723c268a:/opt# <- No directory in /opt as seen here
```

Using Username avidebuguser

A non-admin user (who is also a super-user) can be associated with the NSX Advanced Load Balancer Controller by using `attach <Avi Controller IP>` command. This will provide the Controller container access to the user as an **`avidebuguser`**. The **`avidebuguser`** is also a sudo user. `Attach` option is available only if the user (local or remote) is configured as a super-user.

```
[admin:avi-controller]: > attach controller 10.10.1.10
No handlers could be found for logger "root"
Warning: Permanently added '10.10.1.10' (ECDSA) to the list of known hosts.
Avi Cloud Controller
Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
All rights reserved.
Version: 17.1.8
Date: 2017-09-21 06:03:07 UTC
Build: 9020
Management: 10.10.1.10/23 UP
Gateway: 10.10.1.1 UP
Esx and OpenstackWelcome, this is your controller!!!
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Fri Oct 27 10:32:36 2017 from 172.17.0.2
avidebuguser@avi-controller:~$
```

Use the `ls` command to check the log files as shown below.

```
avidebuguser@avi-controller-2:/opt$ ls
*avi  zookeeper-3.4.6*
```

```
avidebuguser@avi-controller-2:/opt/avi/log$ pwd
/opt/avi/log
```

Additional Information

For more information on NSX Advanced Load Balancer Linux CLI and NSX Advanced Load Balancer CLI access, see CLI - Linux Command Line Mode and Chapter 2 CLI Access Settings.

Adding Required HMAC to the Allowed HMACs Lists using NSX Advanced Load Balancer UI

Sometimes the virtual machine (VM) for the NSX Advanced Load Balancer Controller is accessible through the UI but is not accessible through SSH, for example, Putty (Windows) and Microsoft Certification Test Tool for Azure (Windows).

By default, NSX Advanced Load Balancer does not allow **hmac-sha1-96** for management access to NSX Advanced Load Balancer Controller and Service Engines. To access the NSX Advanced Load Balancer Controller through an SSH Client (for example, Putty), which needs to use **`hmac-sha1-96`**, add **`hmac-sha1-96`** to the Allowed HMACs using UI. If there is no entry in the **Allowed HMACs**, all the default HMACs are allowed.

1 Navigate to **Administration > System Settings**, and click **edit**.

2 Under **Access**, add **hmac-sha1-96** to the **Allowed HMACs** option.

3 Click **Save**.

Once the required HMAC is allowed for the management access to the NSX Advanced Load Balancer Controller, SSH will be accessible to the various SSH clients. For the list of allowed HMACs for the management access of the NSX Advanced Load Balancer Controller, see Restricting the Allowed HMACs.

Securing Management IP Access

This topic describes how to restrict access to both the NSX Advanced Load Balancer Controller's management interfaces and the list of ciphers and HMACs that are allowed for management sessions.

By default, the NSX Advanced Load Balancer Controller does not restrict the client IP addresses that are allowed to attempt access to the Controller through its management interfaces. The NSX Advanced Load Balancer provides a way to define the set of IP addresses allowed to attempt management access to the Controller.

Additionally, access through all the management interfaces can be further restricted by explicitly specifying the ciphers and keyed-hash message authentication codes (HMACs) that are allowed.

To establish a management connection with the Controller through any management interface, the client must support a cipher and HMAC that are allowed by the Controller.

Note

- When configuring the IP access list for API and SSH, ensure that the Controller nodes can talk to each other by either placing them on the same subnet or explicitly adding the Controller IP addresses.
- NSX Advanced Load Balancer does not currently support SSH and System Internal Access control types if controller IPs are FQDN based.

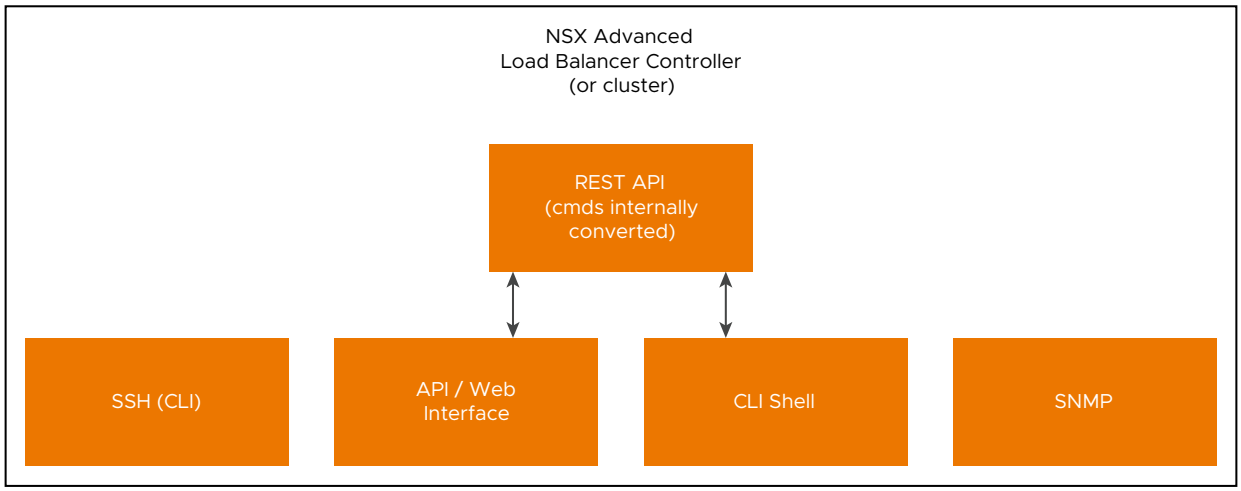
IP Access Lists

When the configuration is modified to specify allowed IP addresses for accessing specific management interfaces, the NSX Advanced Load Balancer programs the Linux IP tables on the NSX Advanced Load Balancer Controller host to allow the specified IP addresses at the network layer.

Separate IP access lists can be configured for each of the following management interfaces:

- REST API / Web interface (or any script or other means of automation that uses the REST API)
- SSH daemon
- CLI Shell (allows remote CLI access)
- SNMP

The following is a logical view of the management interfaces.



Note NSX Advanced Load Balancer internally converts commands that are entered through the web interface or the CLI shell into REST API commands. Similarly, the system responses are converted back into web interface or CLI format when presented in return to the NSX Advanced Load Balancer user. To restrict IP access through the interfaces, separate IP lists must be entered for each interface.

IP Access List Format

For each management interface, the set of client IP addresses that are allowed to access that interface can be specified in any of the following ways:

- Individual IP addresses; example: 1.1.1.1, 2.2.2.2
- IP address ranges; example: 1.1.1.1-1.1.1.100
- IP address prefixes (subnets); example: 1.1.1.0/24
- IP address groups

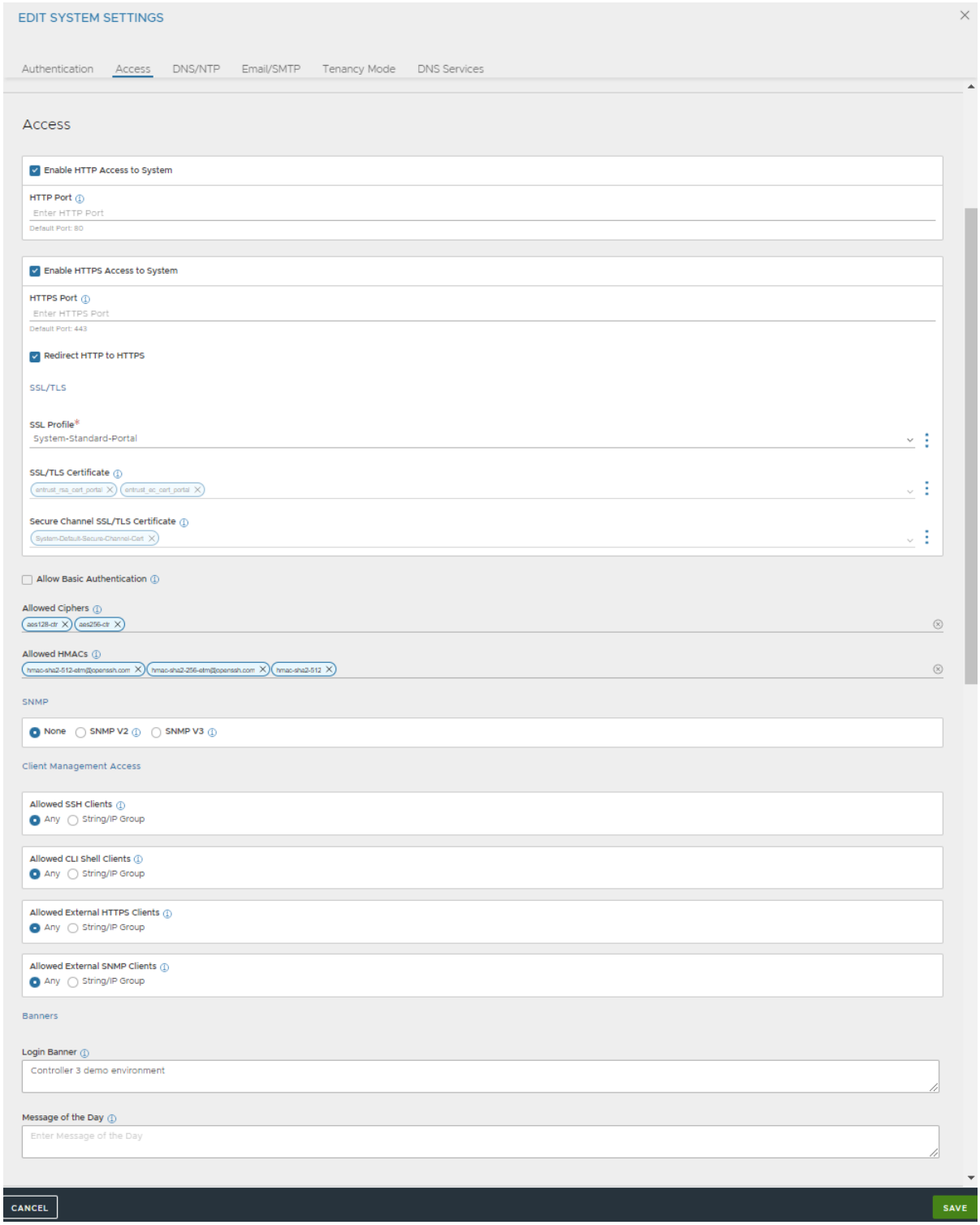
Caution

- When changing the IP access list for the management interface, ensure to include the IP address in the access list. Else, the management session will end after the change is saved.
- If no IP addresses are added to the access list of the management service, any IP address is allowed to attempt access to that service.

Restricting IP Access to Management Interfaces through Web Interface

1 Navigate to **Administration > System Settings**.

2 Click the **edit** icon to open the **EDIT SYSTEM SETTINGS** pop-up window.


EDIT SYSTEM SETTINGS
Authentication
Access
DNS/NTP
Email/SMTP
Tenancy Mode
DNS Services
Access
Enable HTTP Access to System
HTTP Port
Enter HTTP Port
Default Port: 80
Enable HTTPS Access to System
HTTPS Port
Enter HTTPS Port
Default Port: 443
Redirect HTTP to HTTPS
SSL/TLS
SSL Profile*
System-Standard-Portal
SSL/TLS Certificate
Secure Channel SSL/TLS Certificate
Allow Basic Authentication
Allowed Ciphers
aes128-ctr
aes256-ctr
Allowed HMACs
hmac-sha2-512
SNMP
None
SNMP V2
SNMP V3
Client Management Access
Allowed SSH Clients
Any
String/IP Group
Allowed CLI Shell Clients
Any
String/IP Group
Allowed External HTTPS Clients
Any
String/IP Group
Allowed External SNMP Clients
Any
String/IP Group
Banners
Login Banner
Controller 3 demo environment
Message of the Day
Enter Message of the Day
CANCEL
SAVE

3 In the **Client Management Access** section, configure the fields **Allowed SSH Clients**, **Allowed CLI Shell Clients**, **Allowed External HTTP(S) Clients**, and **Allowed External SNMP Clients** as required, and enter or select the IP addresses that are allowed access.

   - Host, range, or subnet: Choose **Select From Available** or **Enter Custom Value** to activate the field, and select or type the address(es). If you are listing multiple addresses, use commas to delimit them. For entering a range, use a hyphen between the starting (lowest) and ending (highest) addresses.
   - IP Group: Select the IP group (if already configured) or click **Create** to create the group (list). Enter the group name and address information and click **Save** to return to the **EDIT SYSTEM SETTINGS** pop-up window.

   Note Enter the Controller management IPs for HTTP(s) settings using the field **Allowed External HTTP(S) Clients**. Internal Analytics APIs fail if the management IPs of the cluster nodes are not included in the list of IPs allowed for external HTTP(s).

4 After specifying the allowed client IP addresses for each management service, go to the next section or click **Save** to save the changes and close the popup.

Restricting the Allowed Ciphers

By default, NSX Advanced Load Balancer allows management sessions to use any of the following ciphers:

- aes128-ctr
- aes256-ctr
- arcfour256
- arcfour128
- aes128-cbc
- 3des-cbc
- blowfish-cbc
- aes192-cbc
- aes256-cbc

Ciphers arcfour128 and arcfour256 are not supported. To restrict access to a subset of these ciphers, specify the individual ciphers:

- In the **Update System Access Settings** popup, in the **Allowed Ciphers** field, enter the cipher names. The names must be spelled as shown above. Use commas between the names.
- Go to the next section or click **Save**.

Restricting the Allowed HMACs

The NSX Advanced Load Balancer allows management sessions to use any of the following HMACs:

- hmac-md5
- hmac-md5-96
- hmac-sha1
- umac-128-etm@openssh.com
- hmac-sha256@ssh.com
- umac-64-etm@openssh.com
- hmac-sha256-96@ssh.com
- hmac-sha2-512
- hmac-md5-96-etm@openssh.com
- hmac-md5-etm@openssh.com
- hmac-sha2-512-etm@openssh.com
- hmac-sha2-256-etm@openssh.com
- hmac-sha1-96-etm@openssh.com
- hmac-sha1-etm@openssh.com

To support fewer HMACs, specify the individual HMACs:

1 In the **EDIT SYSTEM SETTINGS** pop-up window, in the **Allowed HMACs** field, enter the HMAC names. The names must be spelled as shown above. Use commas between the names.

2 After updating, click **Save** to save the changes and close the popup.

REST API -IP Addresses That Are Allowed Management Access

The following request to the NSX Advanced Load Balancer REST API retrieves the current system settings, including the `mgmt_ip_access_control` section. This section specifies the client IP addresses that are allowed to access the Controller through the management services.

In this example, access through the web interface or REST API is restricted to addresses in the 10.10.0.0/16 subnet, and to IP addresses in the range 3.3.3.1-100. IP access for the other management services is not included in this output, because IP access has not been explicitly defined for them.

```
API: GET /api/systemconfiguration
Data:
{
    "email_configuration": {
        "from_email": "admin@avicontroller.net",
        "mail_server_name": "localhost",
```

```
        "smtp_type": "SMTP_NONE",
        "mail_server_port": 25
    },
    ...

    "mgmt_ip_access_control": {
        "api_access": {
            "ranges": [
                {
                    "begin": {
                        "type": "V4",
                        "addr": "3.3.3.0"
                    },
                    "end": {
                        "type": "V4",
                        "addr": "3.3.3.100"
                    }
                }
            ],
            "prefixes": [
                {
                    "ip_addr": {
                        "type": "V4",
                        "addr": "10.10.0.0"
                    },
                    "mask": 16
                }
            ],
            "match_criteria": "IS_IN"
        }
    },
}
```

Allowed Ciphers and HMACs

The following REST API request retrieves the list of allowed ciphers and HMACs.

```
API: GET /api/systemconfiguration
Data:
{
    "email_configuration": {
        "from_email": "admin@avicontroller.net",
        "mail_server_name": "localhost",
        "smtp_type": "SMTP_NONE",
        "mail_server_port": 25
    },
        ...

    "ssh_ciphers": [
        "aes128-ctr",
        "aes256-ctr",
        "aes192-cbc",
        "aes256-cbc"
    ],
```

```
    "ssh_hmacs": [
        "hmac-md5",
        "hmac-md5-96",
        "hmac-sha1",
        "hmac-sha256-96@ssh.com",
        "hmac-sha2-512"
    ],
}
```

For more information, see Renew Default (Self-Signed) Certificates on NSX Advanced Load Balancer.

For more information on how to Change the Default Portal Certificates, see *Change the Default Certificate of the Controller* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Login Banner and Message of the Day

This section discusses the steps to configure Message of the Day and Login Banner.

NSX Advanced Load Balancer supports the configuration of the following types of management greeting messages:

Message of the Day

A greeting that appears to users who log in through the NSX Advanced Load Balancer UI, SSH, or CLI.

Login Banner

Message that appears before login through the web interface or CLI.

Configure Message of the Day and Login Banner

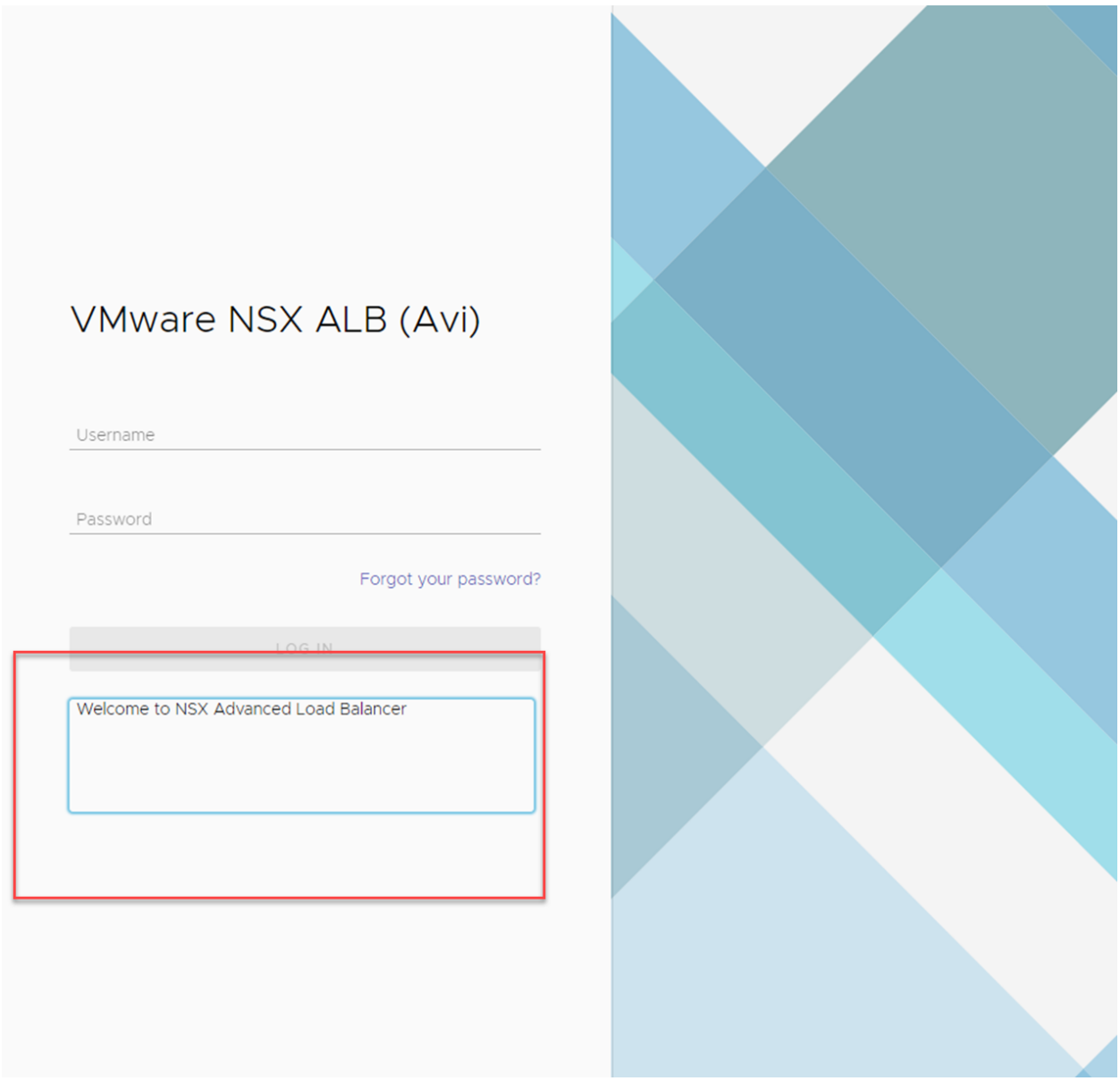
1 Navigate to **Administration > System Settings**.

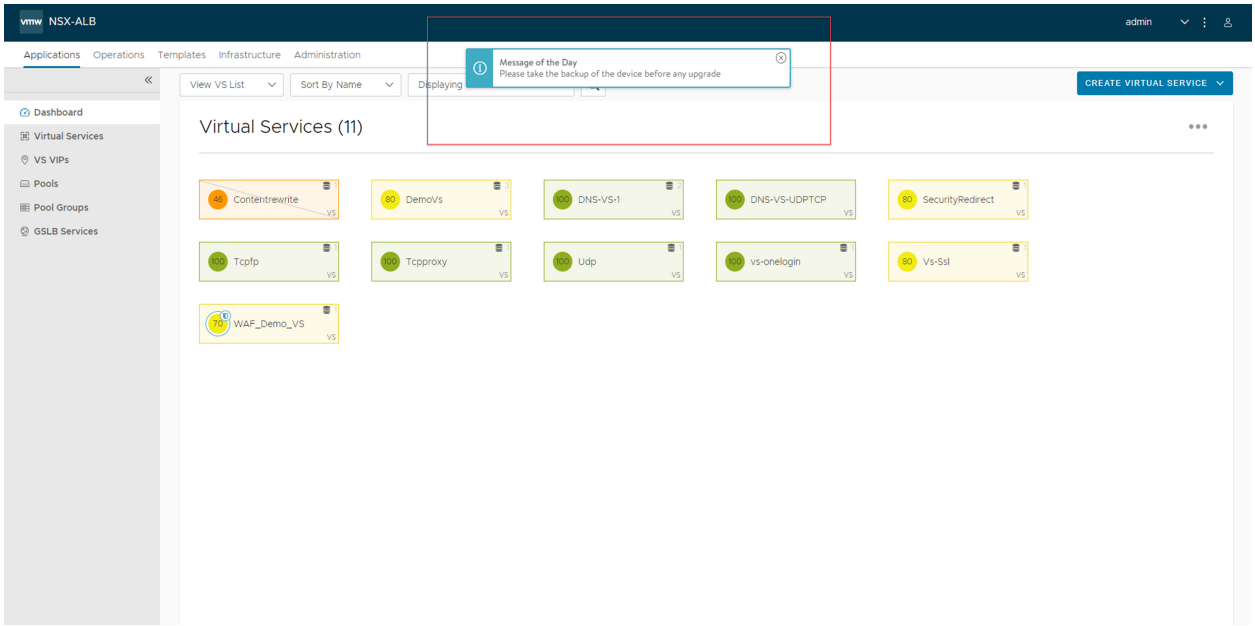
2 Click the **edit** icon next to **System Settings** to display the **EDIT SYSTEM SETTINGS** pop-up window.

3 In the **Message of the Day** and **Login Banner** fields, enter the text messages to be displayed after login through the web interface, SSH, or CLI.

4 Click **Save**.

The changes done in the previous steps will reflect in the next login session. Logout from the NSX Advanced Load Balancer UI, and login again to see the changes reflected as shown below


VMware NSX ALB (Avi)
Username
Password
Forgot your password?
LOG IN
Welcome to NSX Advanced Load Balancer



Certificates

Changing the Default Certificate of the Controller

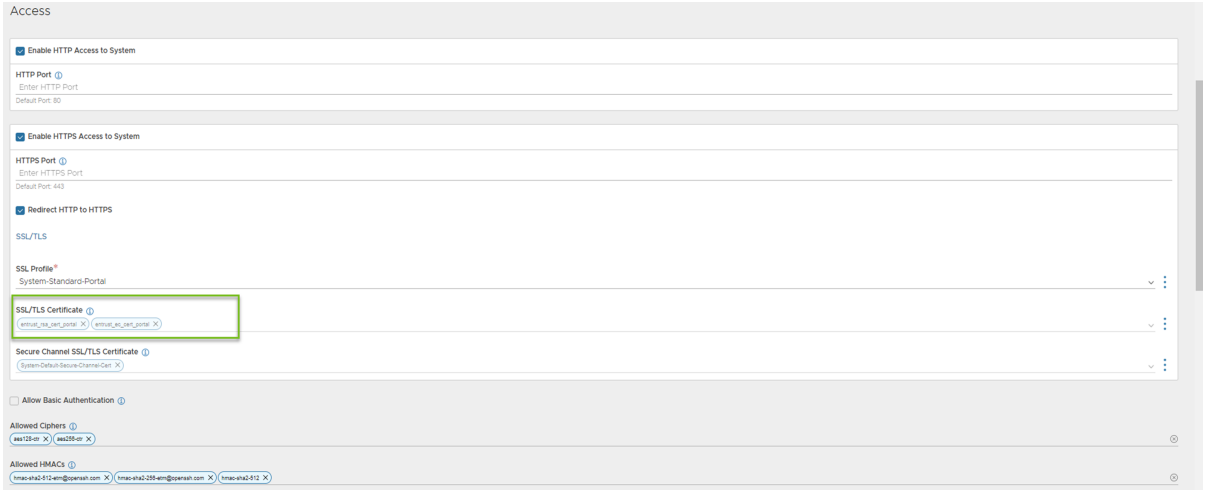
The NSX Advanced Load Balancer Controller can be accessed through the UI using the default certificate associated with it, but users get a warning message regarding certificate mismatch or certificate trust. To avoid a browser warning message while accessing the Controller, install the complete certificate chain matching the FQDN of the Controller and replace the default Controller certificate with the new certificate.

To change the default certificate for the Controller, and import or create a new Controller certificate.

Procedure

1. Navigate to **Templates > Security > SSL/TLS Certificates**, click **Create** and select **Controller Certificate**.
2. Create or Import the existing certificate.
3. Navigate to **Administration > System Settings**, and click the edit icon to edit the **System Settings**.

4 Replace the default or existing certificate with the new certificate in the **SSL/TLS Certificate** drop-down menu.



5 Click **Save**.

Results

Note To avoid any certificate trust issue, make sure that the certificate chain is complete on the Controller and the client browser. Install the complete certificate chain (the root and the intermediate certificates) on the Controller and the client browser. Try accessing the Controller using the UI to confirm it is opening without any error.

Renew Default (Self-Signed) Certificates on NSX Advanced Load Balancer

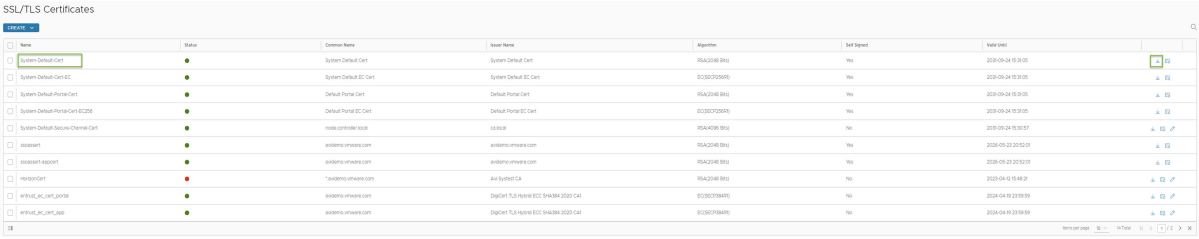
The default certificate on NSX Advanced Load Balancer is self-signed. This section explains how to replace the default certificate when the certificate is expired or is about to expire. The following steps can also be used to replace the self-signed certificate with a third-party signed certificate.

Prerequisites

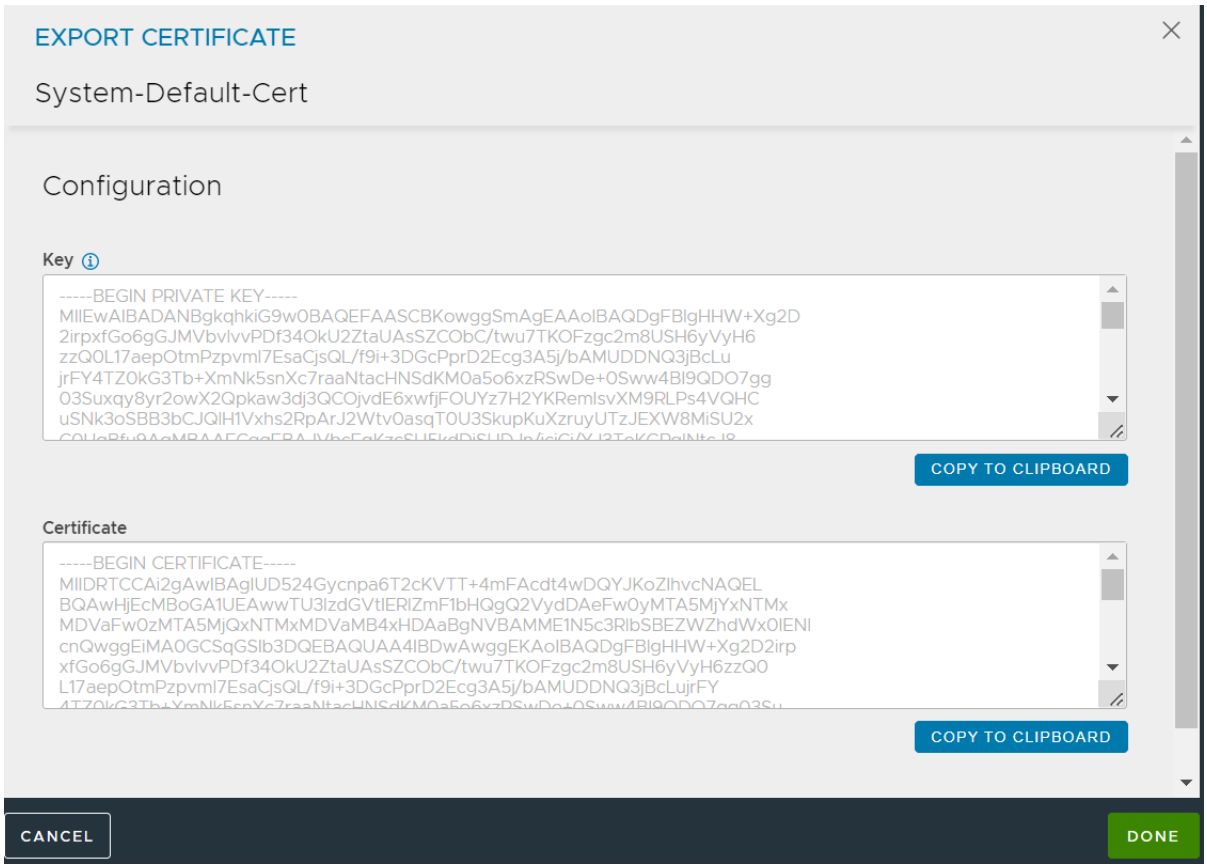
OpenSSL 1.1.x or later.

Configure using NSX Advanced Load Balancer UI

- Navigate to **Templates > Security > SSL/TLS Certificates** and click the **Export** icon (right) of System-Default-Cert entry.



- Copy data from the **Key** and **Certificate** fields to two new files using the **COPY TO CLIPBOARD** option. Name the new files as `system-default.key` and `system-default.cer`, respectively.



Configure using OpenSSL

- Run the following command in OpenSSL to check the expiration date of the certificate.

```
openssl x509 -in system-default.cer -noout -enddate
```

- Run the following command to generate a new CSR with the `system-default.key`.

```
openssl req -new -key system-default.key -out system-default.csr
```

- Run the following command to generate a new certificate with the new expiration date. In this example, the new certificate is named as `system-default2.cer`.

```
openssl x509 -req -days 365 -in system-default.csr -signkey system-default.key -out system-default2.cer
```

- Verify the expiration date on the new certificate `(system-default2.cer)`.

```
openssl x509 -in system-default2.cer -noout -enddate
```

Configure using NSX Advanced Load Balancer CLI and UI

- Copy the `system-default2.cer` and the `system-default.key` to the Controller.

  Optional Step: Before performing the next steps, you can deactivate any virtual services that are configured to use the `System-Default-Cert`.

- Log in to the CLI, and execute the following command to perform the changes for the default certificate on NSX Advanced Load Balancer (System-Default-Cert).

```
[admin:cntrl1]: > configure sslkeyandcertificate System-Default-Cert
```

- Execute the certificate command and click **Enter**. Run certificate file `<path to system-default2.cer>/system-default2.cer`. Enter the `save` command to save the changes.

```
[admin-cntrl1]: sslkeyandcertificate> certificate
[admin-cntrl1]: sslkeyandcertificate:certificate> certificate file:<path to system-default2.cer>/system-default2.cer
[admin-cntrl1]: sslkeyandcertificate> save
```

- Enter the key file `<path to system-default.key>/system-default.key`. Enter the `save` command.

```
[admin-cntrl1]: sslkeyandcertificate> key file:<path to system-default.key>/system-default.key
[admin-cntrl1]: sslkeyandcertificate> save
```

- Enable the virtual services if they were deactivated before the changes (optional).

- Log in to the UI, navigate to **Templates > Security > SSL/ TLS Certificates** and check the expiry date for the renewed certificate.

Configure Stronger SSL Cipher

This section explains how to Configure Stronger SSL Cipher.

Strength

SSL ciphers are defined by the **Templates > Security > SSL/TLS** Profile. Within a profile, List view and string view are the two modes for configuring ciphers.

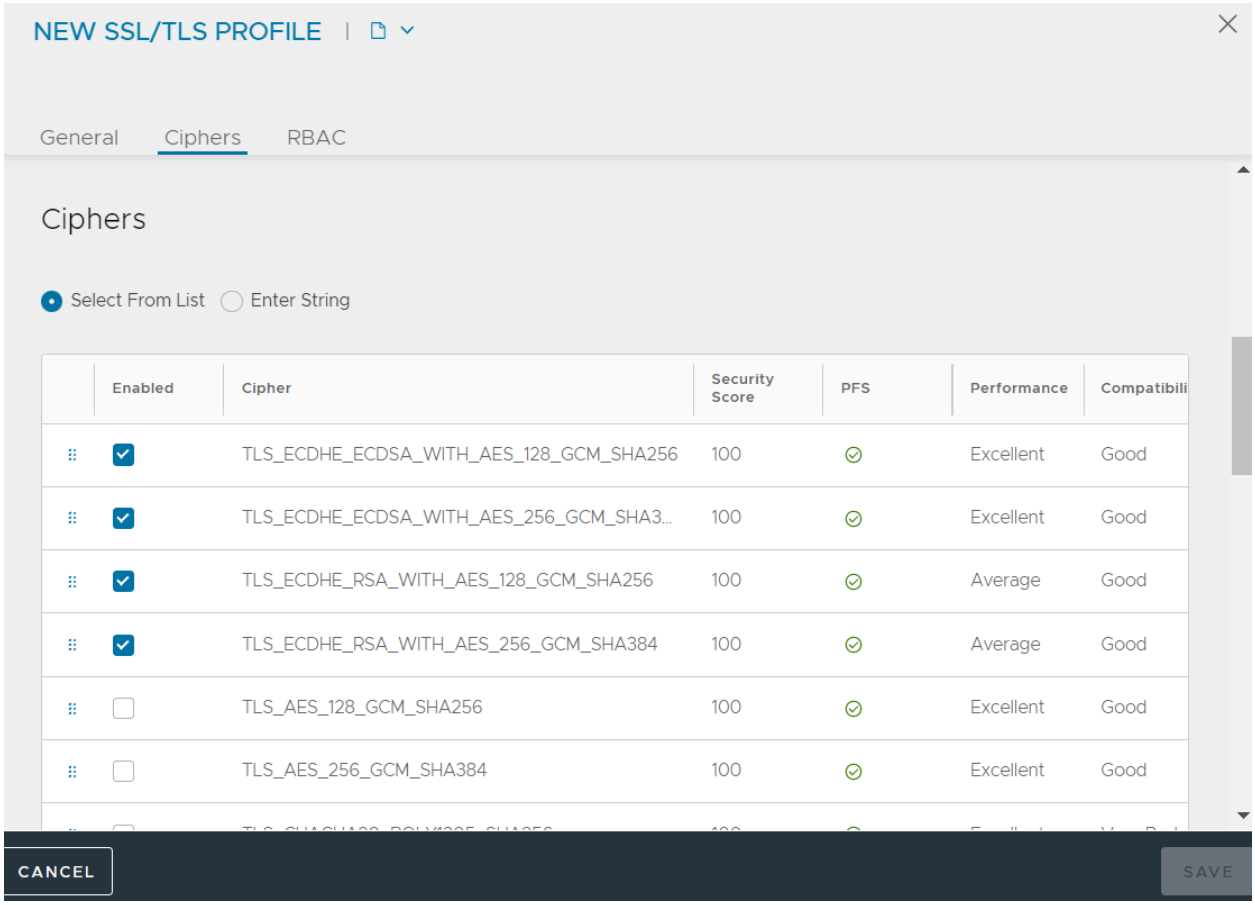
For more information, see *Apple's App Transport Security* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

SSL Rating

Modifying or reordering the list will alter the associated SSL rating in the top-right corner of the SSL/ TLS Profile edit window. This provides insight into the encryption performance, security, and client compatibility of the selected ciphers. This ranking is only made against the validated ciphers from the **List View** mode.

List View

The default cipher list view shows common ciphers in order of priority. Enable or deactivate ciphers using the check box, and reorder them using the up/ down arrows or drag and drop. List view provides a static list of validated ciphers. If alternate ciphers not listed are required, consider using String View. The ciphers included in this list are considered reasonably strong. If a cipher is later deemed to be insecure or less secure, its security score rating will drop to indicate it has fallen out of favor.



String View

The second cipher configuration mode allows accepted ciphers to be added as a string, similar to the OpenSSL syntax for viewing and setting ciphers. For this mode, NSX Advanced Load Balancer accepts all TLS 1.0 - 1.2, and Elliptic Curve ciphers from `https://www.openssl.org/docs/man1.0.2/apps/ciphers.html`. In this mode, the administrator must determine if the enabled ciphers are secure. You can set strong security by employing a known cipher suite, such as HIGH.

```
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES256
```

User Interface Inaccessible Due to Wrong Certificate

There can be instances where the NSX Advanced Load Balancer user interface is not accessible, but the Controller is reachable through CLI access.

Resolution

If the NSX Advanced Load Balancer UI is associated with a revoked certificate, the UI becomes inaccessible. The CLI can still be accessed as it uses SSH. Add the new certificate to the default *NGNIX* configuration file. Use the `/etc/nginx/sites-enabled/default` command to enable the Controller to use the correct certificate.

```
server {
listen 443;
server_tokens off;
more_clear_headers Server;
add_header Strict-Transport-Security "max-age=31536000;
includeSubdomains";
ssl on;
ssl_certificate /var/lib/avi/nginxd/certstore/server_00.cert;
ssl_certificate_key /var/lib/avi/nginxd/certstore/server_00.key;
ssl_certificate /var/lib/avi/nginxd/certstore/server_01.cert;
ssl_certificate_key /var/lib/avi/nginxd/certstore/server_01.key;
```

Test the reachability of the NSX Advanced Load Balancer UI by using the following curl command.

```
> curl -k -v <URL>
```

Import Encrypted Private Key to Install an SSL Certificate

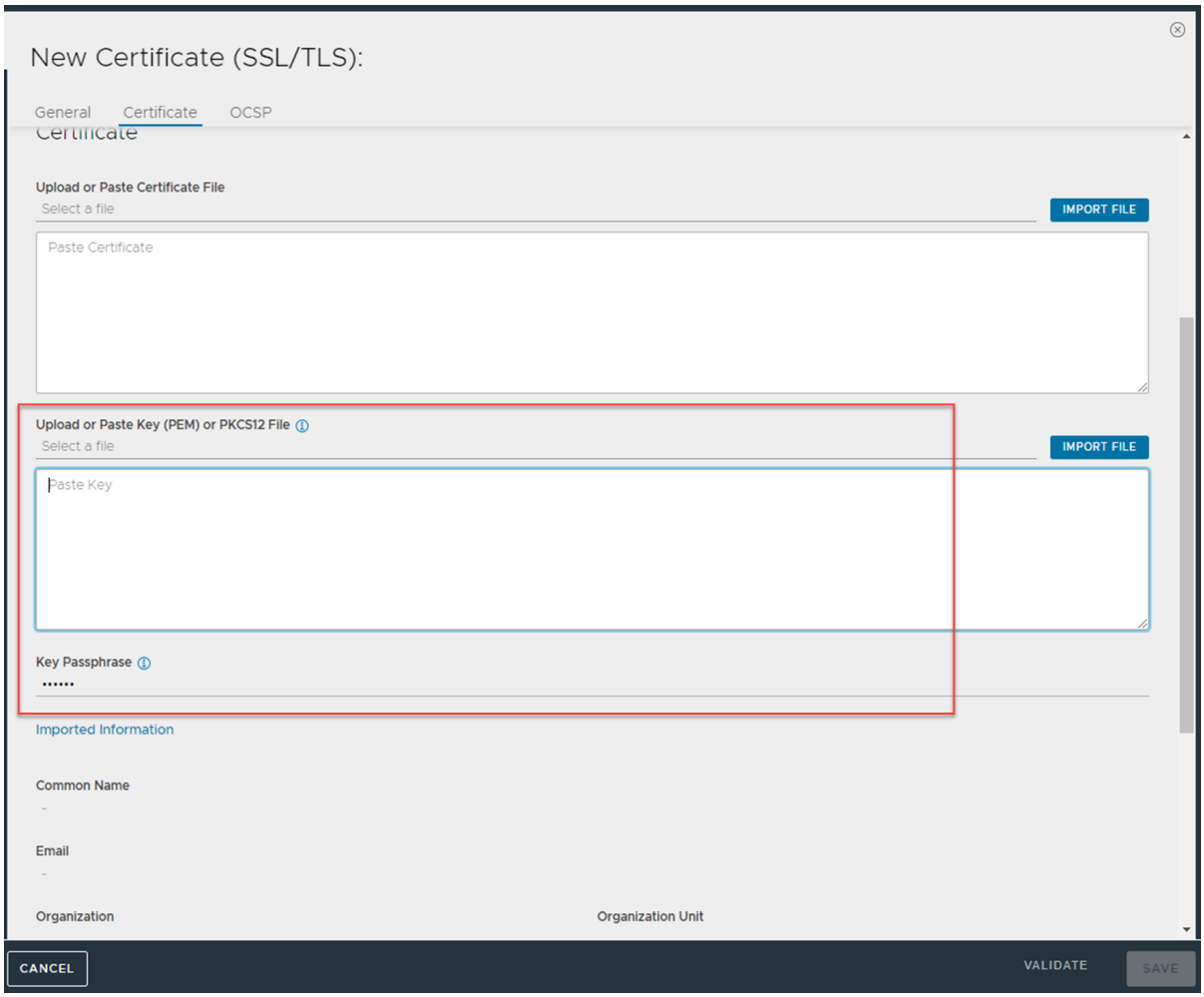
You can import an encrypted private key to install an SSL certificate on NSX Advanced Load Balancer. Password associated with an encrypted private key can be provided using the `Key Passphrase` option, available in the NSX Advanced Load Balancer UI while creating an SSL certificate. NSX Advanced Load Balancer uses the provided passphrase to decrypt the private key while installing the SSL certificate.

The steps to validate the Key Passphrase provided during the SSL certificate creation are as follows:

Procedure

1. Navigate to **Templates > Security > SSL /TLS Certificates**.
2. Click **Create** and choose **Application Certificate** or **Controller Certificate** according to the requirement.
3. From the **Type** drop-down menu, select **Import**.

4 Paste the key text in the **Upload or Paste Key (PEM) or PKCS12 File** text box.

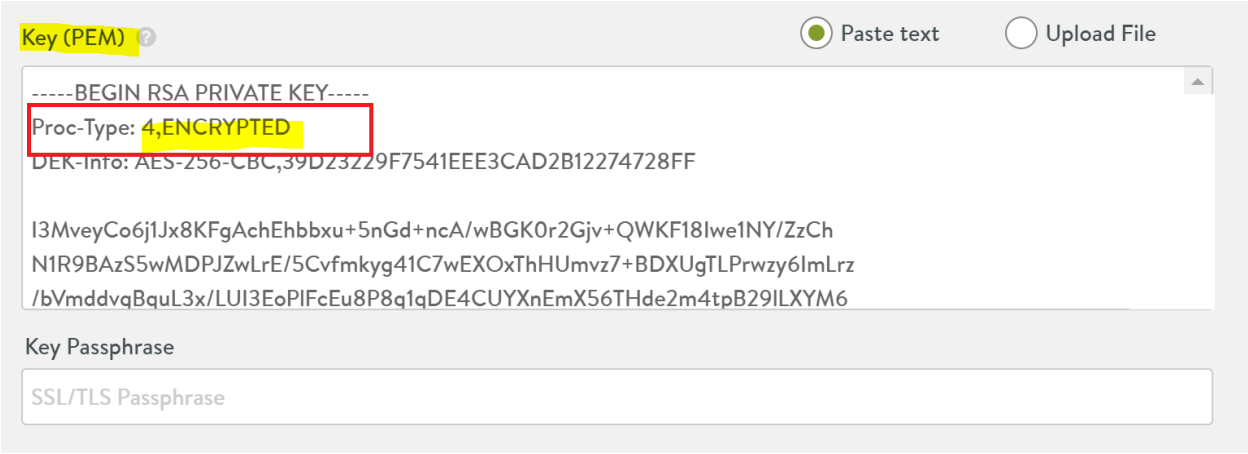


5 Enter the password in the **Key Passphrase**.

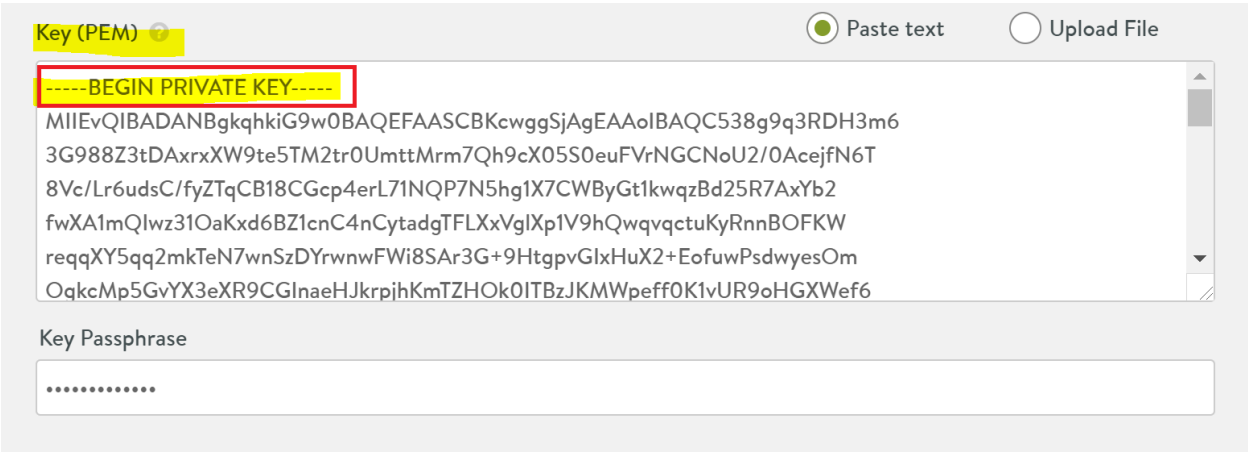
6 Click **Validate** to get the password validated by NSX Advanced Load Balancer.

Example: Sample Encrypted and Decrypted SSL key

Below is an example of an encrypted SSL key.



Note A decrypted SSL key file starts with BEGIN PRIVATE KEY as follows.



Venafi Integration

The NSX Advanced Load Balancer can be set up to integrate with the Venafi Trust Protection Platform™ for automation of SSL and TLS certificate life-cycle management. All certificates will be protected and controlled through TPP. This process is transparent to the NSX Advanced Load Balancer Controllers.

Note Minimum Venafi release is Trust Protection Platform 16.3.

Configuration

The Venafi Trust Protection Platform leverages the NSX Advanced Load Balancer REST API for all communications, including creating and updating SSL certificate and keys. No configuration changes are required on NSX Advanced Load Balancer. TPP requires a user name, password, and an IP address of the Controller. If the Controller is configured with a floating Controller cluster IP, this address must be used. If no floating IP is configured, the IP address of any Controller can be used.

For help in configuring Trust Protection Platform, contact the Venafi support team. You can get the latest version of the required driver, along with instructions for TPP configuration from the team.

Workflow

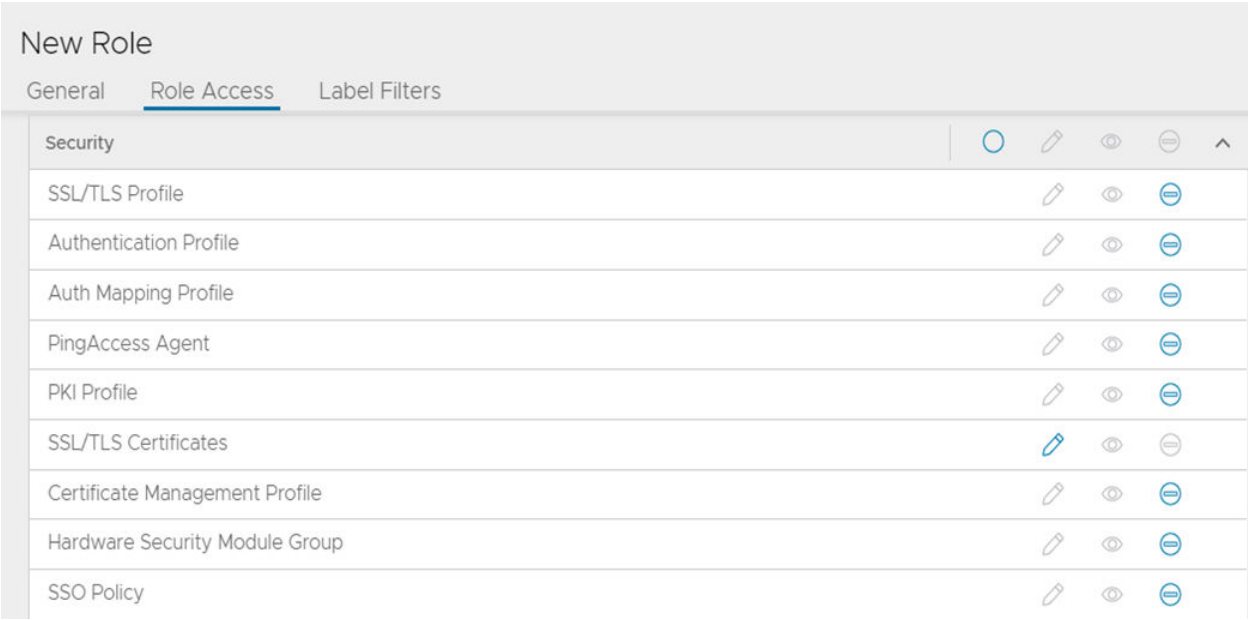
When a new application is created or a new SSL/ TLS certificate is required, use the following workflow.

- From the Trust Protection Platform, create a new certificate for the NSX Advanced Load Balancer. Behind the scenes, the following happens:
  - TPP sends the API calls necessary to generate a CSR on NSX Advanced Load Balancer and import it.
  - TPP forwards the CSR to the configured certificate authority.
  - TPP receives the signed certificate and key from the CA.
  - TPP pushes the certificate and key to NSX Advanced Load Balancer, along with any required chain certificates.
- From the NSX Advanced Load Balancer, create a new application virtual service, attaching the certificate.
- Any subsequent updates to the certificate received by the Controller will automatically and transparently be pushed to Service Engines using the certificate.
- TPP will learn the mapping of the virtual service name to the certificate. It will automatically handle certificate and chain certificate renewals.

IP Access

As a best practice, you can lock down the NSX Advanced Load Balancer administrative UI to known IP addresses. If this has been done, ensure that you include the source IP addresses of the Trust Protection Platform in the allowed-IP list of NSX Advanced Load Balancer for administrative access.

Admin Access



TPP can use any administrator account that has write-access to SSL/ TLS certificates for the required tenants. Best practice is to create a new role for SSL administration, with only write access for SSL/ TLS Certificates enabled. Create a new admin account mapped to this role. This limits the exposure of this account, and provides better audit logs.

Onboard discovery

The Onboard Discovery feature automates the process of importing certificates into Trust Protection Platform from network devices where you can monitor, validate, and provision them.

In case of having virtual services setup on Controller with SSL support and a blank setup or outdated configurations on Venafi TPP, onboard discovery job can be used to generate the corresponding certificate and application objects for the virtual services in Venafi TPP with pre-defined parameter values. This makes provisioning possible immediately after the discovery step. The job can be run automatically at regular intervals of time from Venafi TPP.

Provisioning

Provisioning is the process of pushing attached certificate in NSX Advanced Load Balancer Adaptable App in Venafi TPP to the corresponding virtual service under corresponding tenant, based on parameters of the Adaptable App. Provisioning is automatically triggered when a certificate is renewed in Venafi TPP.

The following are the two types of provisioning:

- Central
  - In central provisioning, since Venafi completely takes care of generation of certificate, the certificate is already ready while pushing.

    - New certificates are uploaded to Controller with the naming convention: **<Common Name><Valid To as yyMMMdd><Last 4 of Serial>**.
- Remote
    - In remote provisioning, CSR is generated by the NSX Advanced Load Balancer and imported to Venafi using API calls, which is then used by Venafi to generate certificate.
    - New certificates are uploaded to the Controller with the naming convention: **<Common Name>RGEN<Current Date/Time as yyMMdd-HHmmss>**.

When a new application is created or a new SSL/ TLS certificate is required, use the following workflow.

- From the Trust Protection Platform, create a new certificate for NSX Advanced Load Balancer. Behind the scenes, the following happens:
    - TPP sends the API calls necessary to generate a CSR on NSX Advanced Load Balancer and import it.
    - TPP forwards the CSR to the configured certificate authority.
    - TPP receives the signed certificate and key from the CA.
    - TPP pushes the certificate and key to the NSX Advanced Load Balancer, along with any required chain certificate.
- From the NSX Advanced Load Balancer, create a new application virtual service, attaching the certificate.
- Any subsequent updates to the certificate received by the Controller will automatically and transparently be pushed to Service Engines using the certificate.
- TPP will learn the mapping of the virtual service name to the certificate. It will automatically handle certificate and chain certificate renewals.

Integrating Let's Encrypt Certificate Authority with NSX Advanced Load Balancer

Let’s Encrypt is a free, automated (automates both issuing and renewing the certificate) and open certificate authority. The following section describes integration of Let's Encrypt Certificate Authority with NSX Advanced Load Balancer.

SSL/TLS protocol helps keeping an internet connection secure and safeguard any sensitive data sent between two machines, systems or devices, preventing intruders from reading, and modifying any information transferred between two machines, systems or devices. Though SSL/TLS certificate ensures secure, encrypted connections between systems, following are some challenges around it.

- Manually getting a certificate.
- The cost associated with a certificate signed by CA.

Let’s Encrypt resolves the above challenges. For more information, see Let’s Encrypt.

Working with Let's Encrypt

Before issuing a certificate, Let's Encrypt servers validate that the requester controls the domain names in that certificate using challenges as defined by the ACME standard. Let's Encrypt uses the ACME protocol to verify that you control a given domain name and to issue you a certificate. There are different ways that the agent or client can prove control of the domain.

- Provisioning a DNS record under the domain (as per CSR's common name).
- Provisioning an HTTP resource under a well-known URI.

The NSX Advanced Load Balancer supports HTTP-01 challenge for domain validation.

HTTP-01 Challenge

- Let's Encrypt gives a token to the ACME client that puts a file on the web server at `http://<YOUR_DOMAIN>/well-known/acme-challenge/<TOKEN>`. This file contains the token and a thumbprint of account key.
- Once the ACME client tells Let's Encrypt that the file is ready, Let's Encrypt tries retrieving it (potentially multiple times from multiple vantage points).
- If validation checks get the right responses from the web server, the validation is considered successful, and a certificate is issued.

Note

- As Let's Encrypt CA communicates on port 80 for HTTP-01 challenge, port 80 must be opened on the firewall and Let's Encrypt CA must be able to reach user network (network where the NSX Advanced Load Balancer System is deployed). Let's Encrypt CA connects through public network to user's NSX Advanced Load Balancer System on port 80.
- The script automatically creates a virtual service on port 80 for the respective virtual service listening on port 443/custom SSL Port, only if there is no virtual service already listening on port 80.

For more information on domain validation, see the following links.

- Challenge Types
- Let's Encrypt - How It Works

Configuring Let's Encrypt

Following is the configuration summary for the Let's Encrypt integration with the NSX Advanced Load Balancer.

1 Get the script which would assist in getting and renewing the certificate.

2 Add the script as controller script on the NSX Advanced Load Balancer System.

3 Add user account with customer (limited access only).

4 Create certificate management profile on NSX Advanced Load Balancer System.

5 Add virtual service on NSX Advanced Load Balancer System.

6 Make sure that the FQDN resolves to public IP and port 80 is open at Firewall.

7 Create CSR and select the configured certificate management profile.

8 Review the list of certificates. Let's Encrypt CA would push signed certificate.

9 Associate the certificate to the configured virtual service.

Configuring the NSX Advanced Load Balancer System

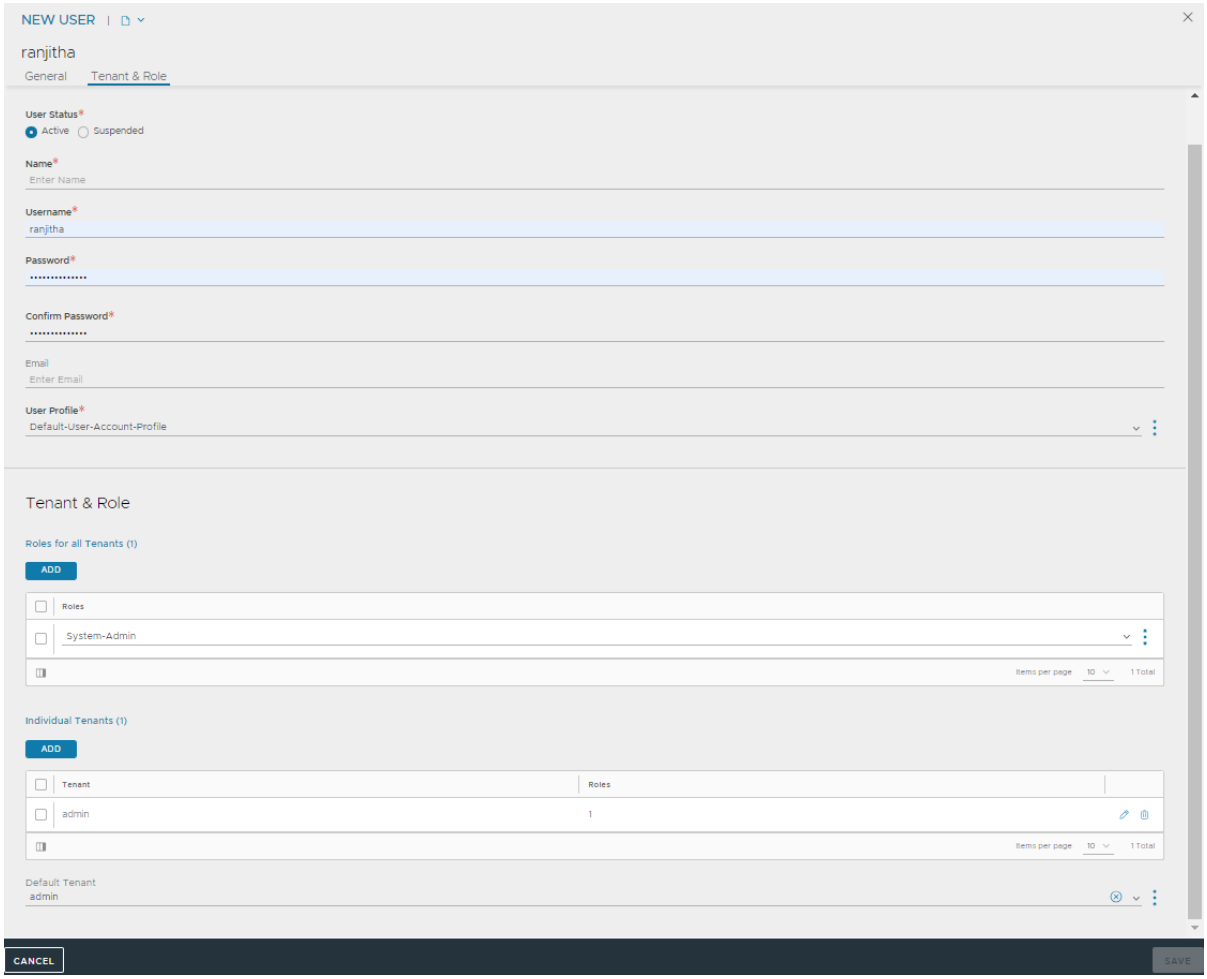
To configure Let's Encrypt for NSX Advanced Load Balancer.

1 Download the script available at letsencrypt_mgmt_profile. To download the file, click **Raw** option. Copy the code available.

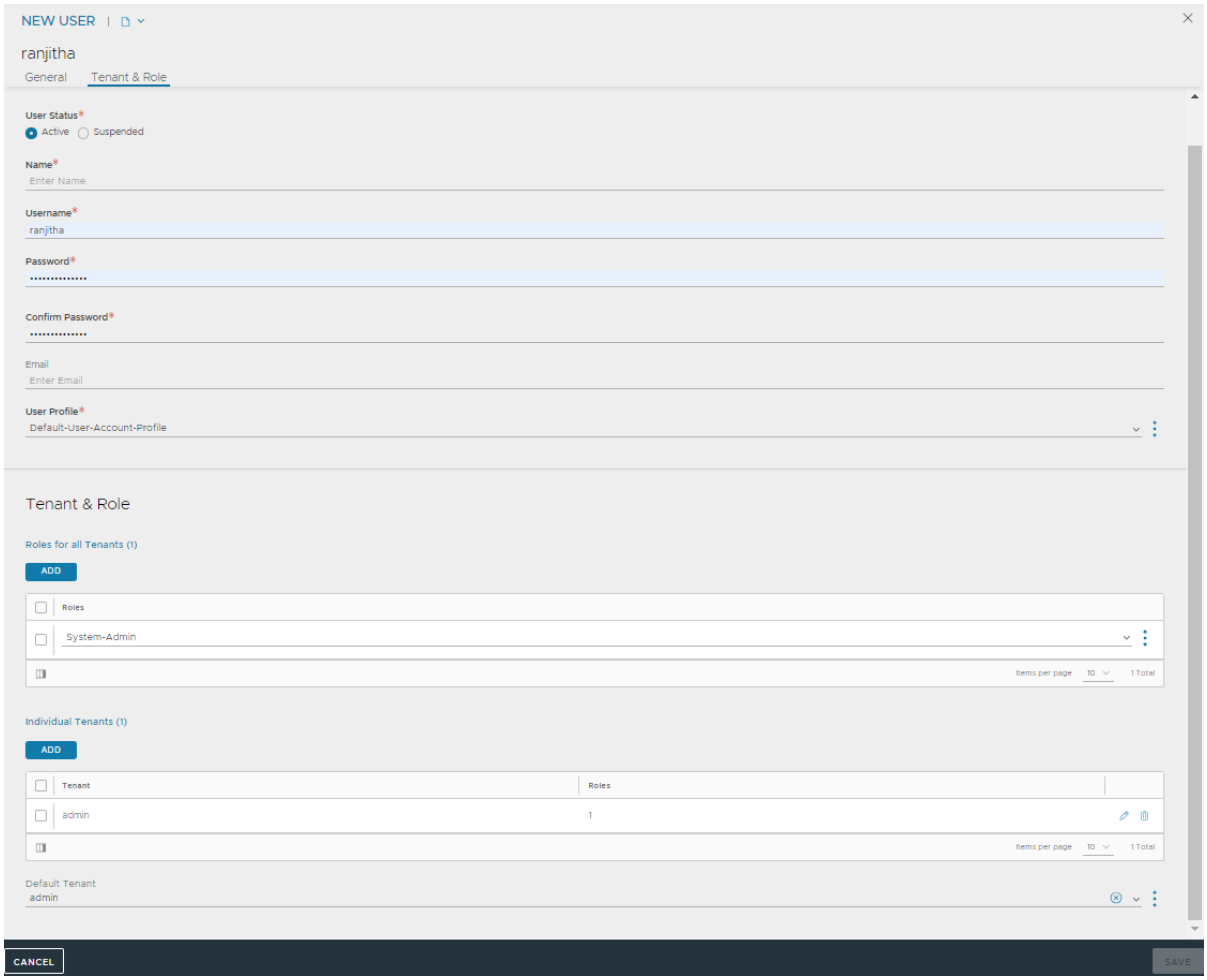
2 In the NSX Advanced Load Balancer, navigate to **Templates > Scripts > ControlScripts** and click **Create**.

3 Add a meaningful name and paste the code copied in step 1 in the **Import or Paste Control Script** field. **Save** the configuration.

4 Configure a custom role by navigating to **Administration > Roles**. Ensure that read and write access is enabled for Virtual Service, Application Profile, SSL/TLS Certificates and Certificate Management Profile, for this role.

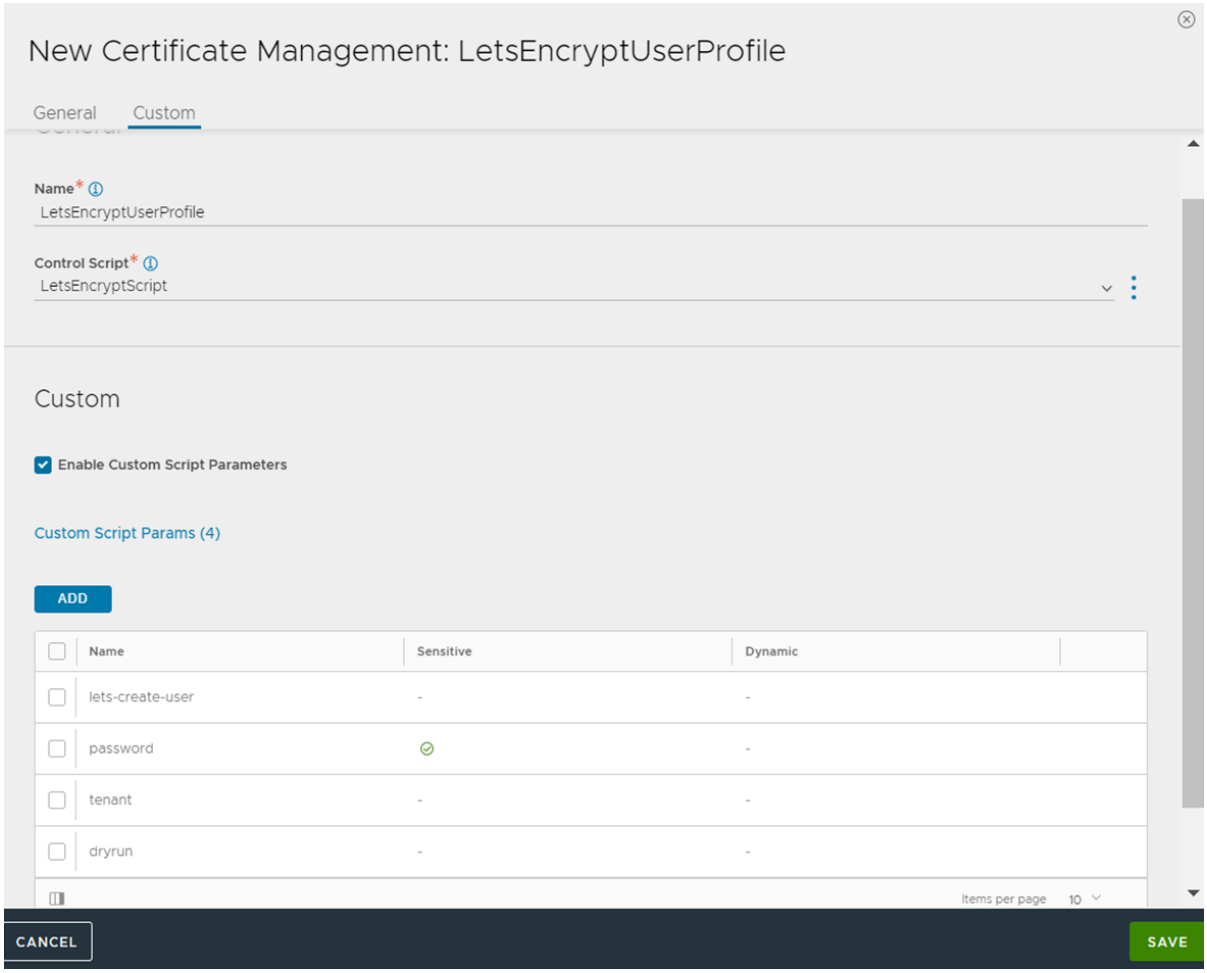


5 Add a user by navigating to **Administration > Users > CREATE**, enter all the required details and select the configured custom role.



6 Navigate to **Templates > Security > Certificate Management** and click **Create**.

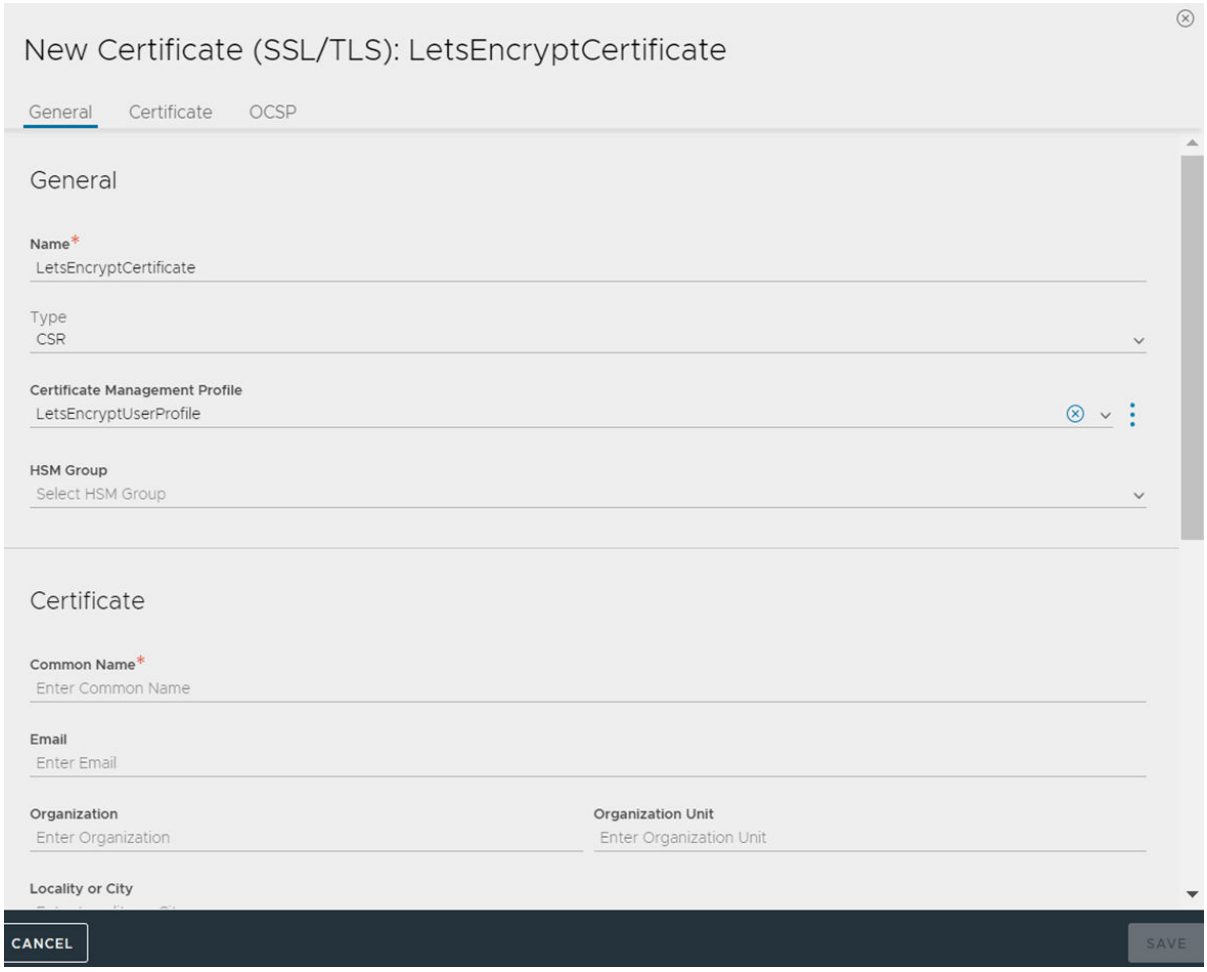
7 Enter a meaningful name, select the configured control script and enable custom parameters, add custom parameters as shown in the following example:



Note It is recommended not to use admin account. Always add a user account which has custom role (with limited access).

8 Navigate to **Templates > Security > SSL/TLS Certificates**, click **Create** and select **Application Certificate**.

9 Enter meaningful name, common name, select the configured certificate management profile, add all relevant details and save the configuration.

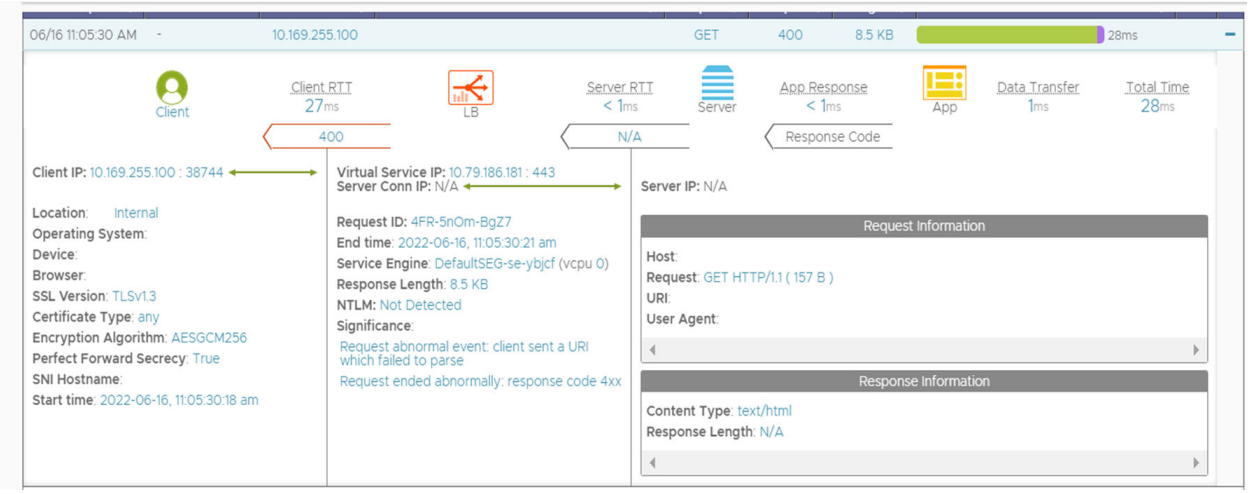


Ensure that a virtual service is configured with the Application Domain Name as Common Name (CN) of certificate. CN of certificate must match with the Application Domain Name of the virtual service. FQDN (CN of certificate or Application Domain Name of virtual service must resolve to IP address and the domain must be reachable).

After few minutes, review the list of the certificates. You can see the certificate pushed by Let's Encrypt CA. Associate the certificate to the configured virtual service.

Logs

To view the logs, enable non-significant logs for the configured virtual service and generate the certificate. Following is an example of the log.



Automation of certificate renewal

`Controllerproperties` has the configuration for the *ssl_certificate_expiry_warning_days*. In the default configuration, the value for *ssl_certificate_expiry_warning_days* is 30 days, 7 days, and 1 day. This setting can be modified if required. When certificate renewal is required (based on configured settings), the script gets activated and automatically takes care of certificate renewal.

Note Let's Encrypt CA imposes a rate limit. So ensure that the renewal of the certificate does not hit the rate limit.

Additional Information

- For more information on rate limit, see Rate Limits.
- For more information on SSL/TLS Certificate, see SSL Certificates.

Certificate Management Integration for CSR Automation

NSX Advanced Load Balancer supports automation of the process for requesting and installing a certificate signed by a certificate authority (CA). This feature handles initial certificate registration and renewal of certificates based on certificate expiration. This topic explains configuring certificate management integration and automatic certificate renewal.

Create a certificate management profile which provides a way to configure a path to a certificate script. Along with the set of parameters the script needs (CSR, common name, and others) to integrate with a certificate management service within the customer's internal network. The script itself is left opaque by design to accommodate the various certificate management services different customers may have.

For SSL certificate configuration, select CSR and fill in the necessary fields for the certificate, and select the certificate management profile to which this certificate is bound. The NSX Advanced Load Balancer Controller will then use the CSR and the script to obtain the certificate and also renew the certificate upon expiration. As a part of the renewal process, a new key pair is generated and a certificate corresponding to this is obtained from the certificate management service.

As a part of the SSL certificate configuration, only select CSR, fill in the necessary fields for the certificate, and select the certificate management profile to which this certificate is bound. The NSX Advanced Load Balancer Controller will then use the CSR and the script to obtain the certificate and also renew the certificate upon expiration. As a part of the renewal process, a new key pair is generated and a certificate corresponding to this is obtained from the certificate management service.

Without the addition of this automation, the process for sending the CSR to the external CA, then installing the signed certificate and keys, must be performed by the NSX Advanced Load Balancer user.

Configuring Certificate Management Integration

This section explains the details to configure certificate management integration.

The following are the steps to configure certificate management integration:

1 Prepare a Python script that defines a `certificate_request()` method. The method must accept the following input as a dictionary:
   a CSR
   b Hostname for the **Common Name** field.
   c Parameters defined in the certificate management profile.
2 Create a certificate management profile that calls the script.

For more information, click here.

Prepare the Script

The script must use the `def certificate_request` command. For instance,

```
def certificate_request(csr, common_name, args_dict):
    """
    Check if a token exists that can be used:
    If not, authenticate against the service with the provided credentials.
    Invoke the certificate request and get back a valid certificate.
    Inputs:
      @csr : Certificate signing request string. This is a multi-line string output like what
             you get from openssl.
      @common_name: Common name of the subject.
      @args_dict: Dictionary of the key value pairs from the certificate management profile.
    """
```

The specific parameter values to be passed to the script are specified within the certificate management profile.

Sensitive Parameters

For parameters that are sensitive, for instance, passwords, the values can be hidden. Marking a parameter sensitive prevents its value from being displayed in the web interface or being passed by the API.

Dynamic Parameter

The value for a certificate management parameter can be assigned within the profile or within individual CSRs.

- If the parameter value is assigned within the profile, the value applies to all CSRs generated using this profile.
- To dynamically assign a parameter's value, indicate that the parameter is dynamic within the certificate management profile. This leaves the parameter's value unassigned. In this case, the dynamic parameter's value is assigned when creating an individual CSR using the profile. The parameter value applies only to that CSR.

Create the Certificate Management Profile

This section explains how to create the certificate management profile.

Procedure

1. Navigate to **Templates > Security > Certificate Management** and click **Create**.
2. Specify the name for the profile.
3. Select an alert script configuration object type for the certificate management profile from the drop-down list.
4. If the profile need to pass some parameter values to the script, check **Enable Custom Parameters** box, and specify their names and values.

   In this example, the location (URL) of the CA service and the login credentials for the service, will be passed to the script. For parameters that are sensitive, such as, passwords, select **Sensitive**. Marking a parameter sensitive prevents its value from being displayed in the web interface or being passed by the API. For parameters that are to be dynamically assigned during CSR creation, select **Dynamic**. This leaves the parameter unassigned within the profile.
5. Click **Save**.

Use the Certificate Management Profile to get Signed Certificates

After adding the script and creating the certificate management profile, the profile can be used to easily obtain and install CA-signed certificates.

Procedure

1. Navigate to **Templates > Security > SSL/TLS Certificates**. Select **Application Certificate** option from the **CREATE** dropdown menu.
2. Specify the certificate name and select the type as **CSR** in the **Type** field.
3. Select the profile configured in the previous section from the **Certificate Management Profile** dropdown menu.
4. Specify the certificate details and click **Save**

   The NSX Advanced Load Balancer Controller generates a key pair and CSR, executes the script to request the CA-signed certificate from the NSX Advanced Load Balancer PKI service, and saves the signed certificate in persistent storage.

Automatic Certificate Renewal

This section explains about automatic certificate renewal.

You can choose to customize when certificate expiry notifications are sent; see the topic Certificate Management Integration for CSR Automation section. If the certificate management profile is configured for a certificate, a renewal is attempted in the last-but-one interval. By default, NSX Advanced Load Balancer Controller generates events 30 days, seven days, and one day before expiry. In this setting, certificate renewal will be attempted seven days before expiry.

If the certificate management profile is configured for automatic certificate renewal, a renewal is attempted just prior to the penultimate notification (in the above example, that will be just prior to the seven-day notification). If the renewal succeeds, the last two notifications are not sent. If the renewal fails, the penultimate notification is sent. Thereafter, if a manual renewal succeeds prior to the last notification, it is skipped. Otherwise, the final notification will be sent (with no accompanying final attempt to renew).

When a certificate renewal occurs, a new expiration date is set and yet another notification schedule is established per the values within the `ssl_certificate_expiry_warning_days` array in force at the time.

SSL Certificates

NSX Advanced Load Balancer supports terminating client SSL and TLS connections at the virtual service, which requires it to send a certificate to clients that authenticate the site and establishes secure communications.

A virtual service that handles secure connections requires the following:

- SSL/ TLS profile
  - This determines the supported ciphers and versions. For more information on SSL/ TLS profiles, see *SSL/TLS Profile* topic in the *VMware NSX Advanced Load BalancerConfiguration Guide*.
- SSL Certificate
  - This is presented to clients connecting to the site.
  - SSL certificates can be used to present to administrators connecting to the NSX Advanced Load Balancer web interface or API, and also for the NSX Advanced Load Balancer SE to present to servers when SE-to-server encryption is required with client (the SE) authentication.

The **SSL/TLS Certificates** page on **Templates > Security > SSL/TLS Certificates** allows the import, export, and generation of new SSL certificates or certificate requests. Newly-created certificates may be either self-signed by NSX Advanced Load Balancer or created as a Certificate Signing Request (CSR) that must be sent to a trusted Certificate Authority (CA), that generates a trusted certificate.

- Creating a self-signed certificate generates both the certificate and a corresponding private key.
- Imported existing certificates are not valid until a matching key has been supplied.
- NSX Advanced Load Balancer supports PEM and PKCS #12 formats for certificates.

SSL/ TLS Certificates Page

Navigate to **Templates > Security > SSL/TLS Certificates** to open the **SSL/TLS Certificates** page. This tab includes the following functionalities:

- **Search**: Search across the list of objects.
- **Create**: Shows the list of certificates from the drop-down menu to create a certificate.
- **Edit**: Opens the **Edit Certificate**window. Only incomplete certificates that do not have a corresponding key can be edited.
- **Export**: The down arrow icon exports a certificate and corresponding private key.
- **Delete**: A certificate may only be deleted if it is not currently assigned to a virtual service. An error message will indicate the virtual service referencing the certificate.

The table on this tab contains the following information for each certificate:

- **Name**: This displays the name of the certificate. Mouse over the name of the certificate will display any intermediate certificate that has been automatically associated with the certificate.

- **Status**: This shows the status of the certificate. Status in 'green' indicates it is good, in 'yellow'/orange/red' indicates the certificate is expiring soon or has already expired, and 'gray' indication, if the certificate is incomplete.
- **Common Name**: This displays the fully qualified name of the site to which the certificate applies. This entry must match the hostname the client will enter in their browser in order for the site to be considered trusted.
- **Issuer Name**: This displays the name of the certificate authority.
- **Algorithm**: This displays the algorithm as either EC (Elliptic Curve) or RSA.
- **Self Signed**: This displays whether the certificate is self-signed by NSX Advanced Load Balancer or signed by a Certificate Authority.
- **Valid Until**: This displays the date and time when the certificate expires.

Create Certificate

Navigate to **Templates > Security > SSL/TLS Certificates**. Click **Create** to open the **New Certificate (SSL/ TLS)**window.

When creating a new certificate, you can select any of the following certificates:

- **Root/Intermediate CA Certificate**: This certificate is used to automatically create the certificate chain for application certificates. There are no configuration options other than importing the certificate through a file or pasting the text. The root/ intermediate certificate will show up in a separate table at the bottom of the **SSL Certificates** page. It is recommended to import the root/intermediate certificate prior to importing an application certificate that relies on the intermediate for the chain.
- **Application Certificate**: This certificate is used for normal SSL termination and decryption on NSX Advanced Load Balancer. This option is also used to import or create a client certificate for NSX Advanced Load Balancer to present to a back-end server when it needs to authenticate itself.
- **Controller Certificate**: This certificate is used for the GUI and API for the Controller cluster. Once uploaded, select the certification through **Administration > Settings > Access Settings**.

To create a new certificate, follow the steps below:

- **Name**: Specify the name of the certificate.
- **Type**: Select the type of certificate to create from the drop-down menu. The following are the options:
  - **Self Signed**: Quickly create a test certificate that is signed by NSX Advanced Load Balancer. Client browsers will display an error that the certificate is not trusted. If the HTTP application profile has **HTTP Strict Transport Security (HSTS)** enabled, clients will not be able to access a site with a self signed certificate.

  - **CSR**: Create a valid certificate by first creating the certificate request. This request must be sent to a certificate authority, which will send back a valid certificate that must be imported back into NSX Advanced Load Balancer.
  - **Import**: Import a completed certificate that was either received from a certificate authority or exported from another server.
- **Common Name**: Specify the fully qualified name of the site, such as `www.vmware.com`. This entry must match the hostname the client entered in the browser in order for the site to be considered trusted.
- Input the required information required for the type of certificate you are creating:
  - Self-Signed Certificates
  - CSR Certificates
  - Importing Certificates

Note The OCSP stapling can be enabled and configured using the UI. For more information, see *OCSP Stapling in NSX Advanced Load Balancer* topic in the *VMware NSX Advanced Load BalancerConfiguration Guide*.

Self-Signed Certificates

NSX Advanced Load Balancer can generate self-signed certificates. Client browsers do not trust these certificates and will warn the user that the virtual service's certificate is not part of a trust chain.

Self-signed certificates are good for testing or environments where administrators control the clients and can safely bypass the browser's security alerts. Public websites must never use self-signed certificates.

If you have selected **Self Signed** option from **Type** drop-down menu in the **New Certificate** window, then specify the following details:

- **Common Name**- Specify the fully-qualified name of the site, such as `www.vmware.com`. For the site to be considered trusted, this entry must match the hostname that the client entered in the browser.
- **Organization**: Company or entity registering the certificate, such as NSX Advanced Load Balancer Networks, Inc. (optional).
- **Organization Unit**: Group within the organization that is responsible for the certificate, such as Development (optional).
- **Country**: Country in which the organization is located (optional).
- **State Name or Province**: State in which the organization is located (optional).
- **Locality or City**: City of the organization (optional).
- **Email**: The email contact for the certificate (optional).

- **Subject Alternate Name (SAN)**- The Subject Alternate Name (SAN) lets you specify additional host names to be protected by a single SSL certificate, such as `example.com` and `example.org`. The are essentially the alternative identities of the subject that is specified in the Certificate.
- **Algorithm**: Select either EC (Elliptic Curve) or RSA. RSA is older and considered less secure than EC, but is more compatible with a broader array of older browsers. EC is new, less expensive computationally, and generally more secure; however, it is not yet accepted by all clients. NSX Advanced Load Balancer allows a virtual service to be configured with two certificates at a time, one each of RSA and EC. This will enable it to negotiate the optimal algorithm with the client. If the client supports EC, then the NSX Advanced Load Balancer will prefer this algorithm, which gives the benefit of natively supporting Perfect Forward Secrecy for better security.
- **Key Size**: Select the level of encryption to be used for handshakes, as follows:
  - 2048-bit is recommended for RSA certificates. Higher values may provide stronger encryption, but dramatically increase the CPU resources required by both NSX Advanced Load Balancer and the client. For stronger encryption, use ECC certificates instead.
  - SECP256R1 is recommended for EC certificates.

  Higher values may provide better encryption but increase the CPU resources required by both NSX Advanced Load Balancer and the client.
- **Enable HSM Certificate**: Rather than store the private key locally on the Controller or Service Engine, it is maintained in an external hardware security module. This option enables referencing an HSM profile containing information about communicating with the HSM.
- After specifying the necessary details, click **Save**.

CSR Certificates

The Certificate Signing Request (CSR) is the first of three steps involved in creating a valid SSL/TLS certificate. The request contains the same parameters as a Self-Signed Certificate. However, NSX Advanced Load Balancer does not sign the completed certificate. Rather, it must be signed by a Certificate Authority that is trusted by client browsers.

You can select **CSR** option from **Type** drop-down menu in the **New Certificate** window. The configuration options for a certificate signing request are the same as for self-signed certificates. See the descriptions above for definition of each field.

Once completed, forward the completed CSR to any trusted Certificate Authority (CA), such as Thawte or Verisign by selecting the **Certificate Signing Request** at the bottom left of the **Add Certificate** popup. Then either copy-paste it directly to the CA's website or save it to a file for later use.

Once the CA issues the completed certificate, either paste or upload it into **Certificate** field at the bottom right of **Add Certificate** popup. The certificate is not usable on NSX Advanced Load Balancer until this step is complete.

Note It can take several days for the CA to return the finished certificate. Meanwhile, you can close the **New Certificate** window to return to the **SSL/TLS Certificates** page. The new certificate will appear in the table with the notation **Awaiting Certificate Valid Until** column.

When you receive the completed certificate, click **Edit** icon for the certificate to open the **Edit Certificate**, and then paste the certificate and click **Save** to generate the CSR certificate. NSX Advanced Load Balancer will generate a key from the completed certificate automatically.

Import Certificates

You can directly import an existing PEM or PKCS /#12 SSL/ TLS certificate into NSX Advanced Load Balancer (such as from another server or load balancer). A certificate will have a corresponding private key, which must also be imported.

Note NSX Advanced Load Balancer generates the key for self-signed or CSR certificates automatically.

The following are the steps to import the certificate:

1 Navigate to **Templates > Security > SSL/TLS Certificates**.

2 Click **CREATE** and select the certificate type such as **Application Certificate**.

3 Click **Type** and select **Import**. Certificate or Private Key can be imported by copying-pasting or uploading a file.

- **PEM File** – PEM files contain certificate or private key in plain-text Base64 encoded format. Certificate and private key can be provided in separate PEM files or combined in a single PEM file.
- If certificate and private key are provided in a single PEM file, navigate to **Paste Key** text box and add the private key by following any one of the methods listed below:
  - **Upload File**: Click the **Upload File** button, select the PEM or PKCS12 file, then click **Validate** button to parse the file. If the upload is successful then, the **Key** field will be populated.
  - **Paste**: Copy and paste a PEM key into the **Key** field. Ensure that you do not introduce extra characters in the text, which can occur while using some e mail clients or rich text editors. If you copy and paste the key and certificate together as one file then, click **Validate** button to parse the text and populate the **Certificate** field.
- If certificate and private key are provided in two separate PEM files, follow the below steps to import each individually:
  - **Certificate** - Add the certificate in the **Paste Certificate** text box by copying-pasting or file upload, as described above.

    - **Key** – Add the private key in the **Paste Key** field by copying-pasting or file upload.
- **PKCS#12 File** - PKCS #12 files contain both the certificate and key, PKCS #12 is a binary format, which cannot be copied-pasted, and hence it can be uploaded only. Navigate to the **Paste Key** and follow the below step to import the PKCS #12 file.
- **Upload File** - Click the **Import File** button, select the PKCS #12 file, click the **Validate** button to parse the file. If the upload is successful, both the **Key** and **Certificate** fields will be populated.
- **Key Passphrase**: You can also add and validate a key passphrase to encrypt the private key.
- **Import**: Select **Import** to finish adding the new certificate and key. The key will be embedded with the certificate and treated as one object within the NSX Advanced Load Balancer UI.

Certificate Authority

Certificates require a trusted chain of authority to be considered as valid. If the certificate used is directly generated by a certificate authority that is known to all client browsers then, no certificate chain is required. However, if there are multiple levels required, an intermediate certificate may be necessary. Clients will often traverse the path indicated by the certificate to validate on their own if no chain certificate is presented by a site, but this adds additional DNS lookups and time for the initial site load. The ideal scenario is to present the chain certificates along with the site cert.

If a chain certificate, or rather a certificate for a certificate authority, is uploaded through the **Certificate > Import** in the certificates page, it will be added to the **Certificate Authority** section. NSX Advanced Load Balancer will automatically build the certificate chain if it detects a next link in the chain exists.

To validate a certificate that has been attached to a chain certificate, hover the pointer over the certificate's name in the SSL Certificates table at the top of the page. NSX Advanced Load Balancer supports multiple chain paths. Each may share the same CA issuer, or they can be chained to different issuers.

SSL/ TLS Profile

NSX Advanced Load Balancer supports the ability to terminate SSL connections between the client and the virtual service, and to enable encryption between NSX Advanced Load Balancer and the back-end servers. The SSL/ TLS profile contains the list of accepted SSL versions and the prioritized list of SSL ciphers.

Both an SSL/ TLS profile and an SSL certificate must be assigned to the virtual service while configuring it to terminate client SSL/ TLS connections. If you encrypt traffic between NSX Advanced Load Balancer and the servers then, an SSL/ TLS profile must be assigned to the pool. While creating a new virtual service through the basic mode, the default system SSL/TLS profile is used automatically.

SSL termination can be performed on any service port. However, browsers assume that the default port as 443. The best practice is to configure a virtual service to accept both HTTP and HTTPS by creating a service on port 80, by selecting the **+** icon to add an additional service port, and then set the new service port to 443 with SSL enabled. A redirect from HTTP to HTTPS is generally preferable, which can be done through a policy or by using the **System-HTTP-Secure** application profile.

Each SSL/ TLS profile contains default groupings of supported SSL ciphers and versions that may be used with RSA or an Elliptic Curve certificate, or both. Ensure that any new SSL/TLS profile you create, includes ciphers that are appropriate for the certificate type that will be used later. The default SSL/ TLS profiles included with NSX Advanced Load Balancer provides a broad range of security. For instance, the Standard Profile works for typical deployments.

Creating a new SSL/ TLS profile or using an existing profile entails various trade-offs between security, compatibility, and computational expense. For instance, increasing the list of accepted ciphers and SSL versions increases the compatibility with clients while also lowering security potentially.

Note NSX Advanced Load Balancer can accommodate a broader set of security needs within a client community by associating multiple SSL profiles with a single virtual service, and have the Service Engines choose which to use based on the client's IP address. For more information, see 'Client-IP-based SSL Profiles' section in this guide.

SSL Profile Settings

Navigate to **Templates > Profiles > SSL/TLS Profile**.

- **Search**: Searches across the list of objects.
- **Create**: Opens the new application or `systemprofile`.
- **Edit**: Opens the existing profile to edit.
- **Delete**: An SSL/ TLS profile can only be deleted, if it is not currently assigned to a virtual service. An error message will indicate the virtual service referencing the profile. The default system profiles can be modified, but not deleted.

The table on this tab provides the following information for each SSL/ TLS profile:

- **Name**: Name of the profile.
- **Accepted Versions**: SSL and TLS versions accepted by the profile.

Create an SSL Profile

The following are the steps to create or edit an SSL profile:

- **Name**: Specify a unique name for the SSL/ TLS Profile.
- **Type**: Select the type of profile from the drop-down menu.

- **Accepted Versions**: Select one or more SSL/ TLS versions from the drop-down menu to add to this profile. Chronologically, TLS v1.0 is the oldest supported, and TLS v1.2 is the newest. SSL v3.0 is no longer support as of NSX Advanced Load Balancer v15.2. In general with SSL, older versions have many known vulnerabilities while newer versions have many undiscovered vulnerabilities. As with any security, NSX Advanced Load Balancer recommends diligence to understand the dynamic nature of security and to ensure that NSX Advanced Load Balancer is always up to date. Some SSL ciphers are dependent on specific versions of SSL or TLS supported. For more information, see OpenSSL.
- **Accepted Ciphers**: Enter the list of accepted ciphers in the **Accepted Ciphers** field. Each cipher entered must conform to the cipher suite names listed at OpenSSL. Separate each cipher with a colon. For instance, AES:3DES means that this Profile will accept the AES and 3DES ciphers. When negotiating ciphers with the client, NSX Advanced Load Balancer will prefer ciphers in the order listed. You may use an SSL/ TLS profile with both an RSA and an Elliptic Curve certificate. These two types of certificates can use different types of ciphers, so it is important to incorporate ciphers for both types. Selecting only the most secure ciphers may incur higher CPU load on NSX Advanced Load Balancer and may also reduce compatibility with older browsers.

PKI Profile

For more information on PKI Profile, see *PKI Profile* and *PKI Profile Settings* topics in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Certificate Management

To create a new certificate, follow the steps below:

1 From the UI, navigate to the **Templates > Security > Certificate Management**.

2 Click **Create**.

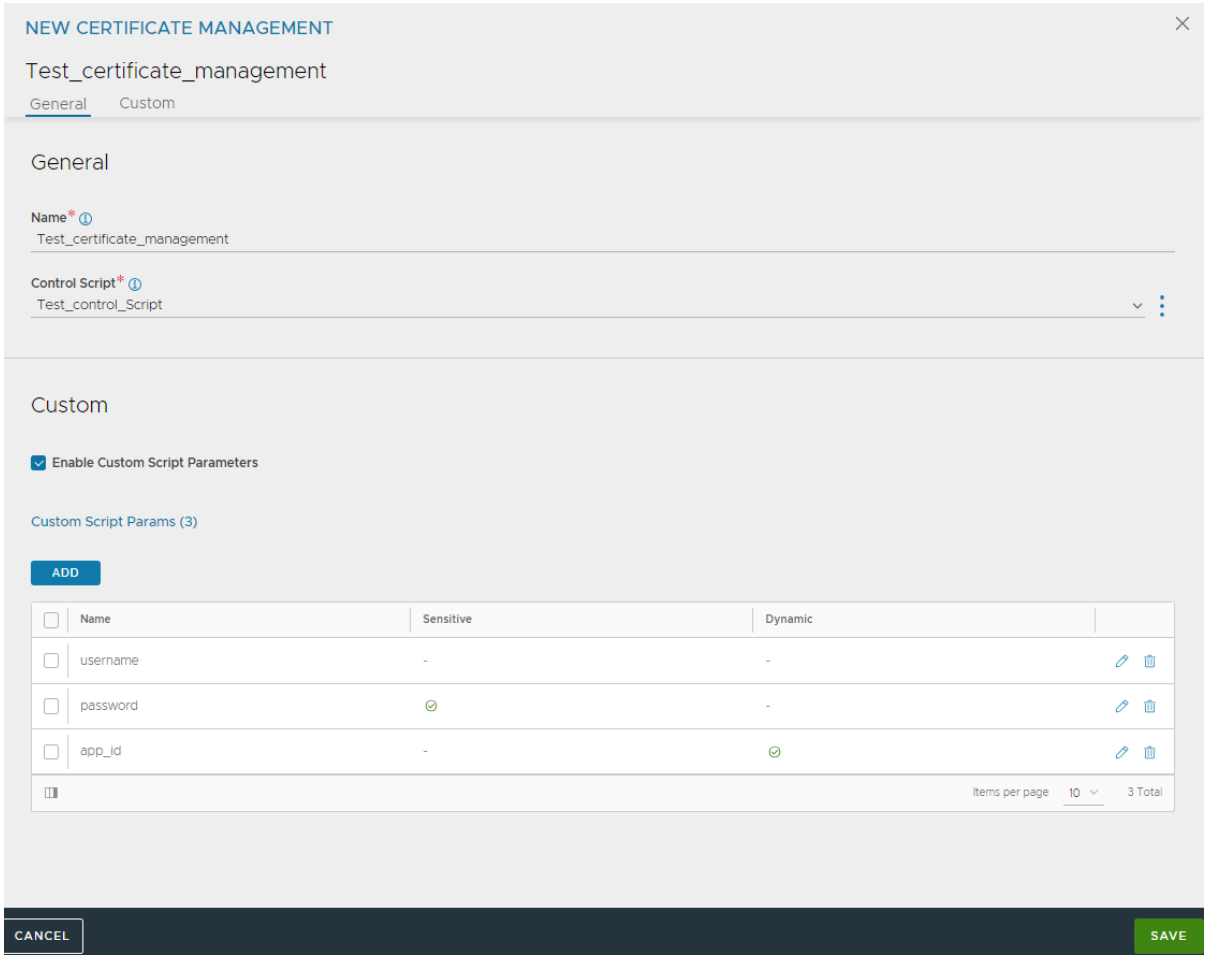
3 In the **New Certificate Management** screen, enter the **Name** of the profile.

4 In the **Control Script** field, select the required alert script configuration, as required.

Note Click **Create** button in the drop down, to create a new Control Script (if required).

5 If the profile needs to pass some parameter values to the script, select **Enable Custom Parameters**.

6 Enter the **Name** and **Value** for the parameters.



Note Re-upload the Control Script, if the file has been modified after uploading for the changes to reflect.

The location (URL) of the Trust Anchor service and the login credentials for the service, will be passed to the script. For parameters that are sensitive (for instance, passwords), select the **Sensitive** check box.

Marking a parameter sensitive prevents its value from being displayed in the web interface or being passed by the API. For parameters that are to be dynamically assigned during CSR creation, select the **Dynamic** check box. This leaves the parameter unassigned within the profile.

7 Click **Save**.

Authentication Profile

The Authentication profile (“auth profile”) allows configuration of clients into a virtual service through HTTP basic authentication.

The authentication profile is enabled through the HTTP basic authentication setting of a virtual service’s **Advanced Properties** tab.

NSX Advanced Load Balancer also supports client authentication through SSL client certificates, which is configured the HTTP Profile's Authentication section.

Auth Profile Settings

Select **Templates > Security > Auth Profile** to open the **Auth** tab. This tab includes the following functions:

- **Search**: Search across the list of objects.
- **Create**: Opens the Create/ Edit window.
- **Edit**: Opens the Create/ Edit window.
- **Delete**: An Auth profile may only be deleted if it is not currently assigned to a virtual service or in use by NSX Advanced Load Balancer for administrative authentication.

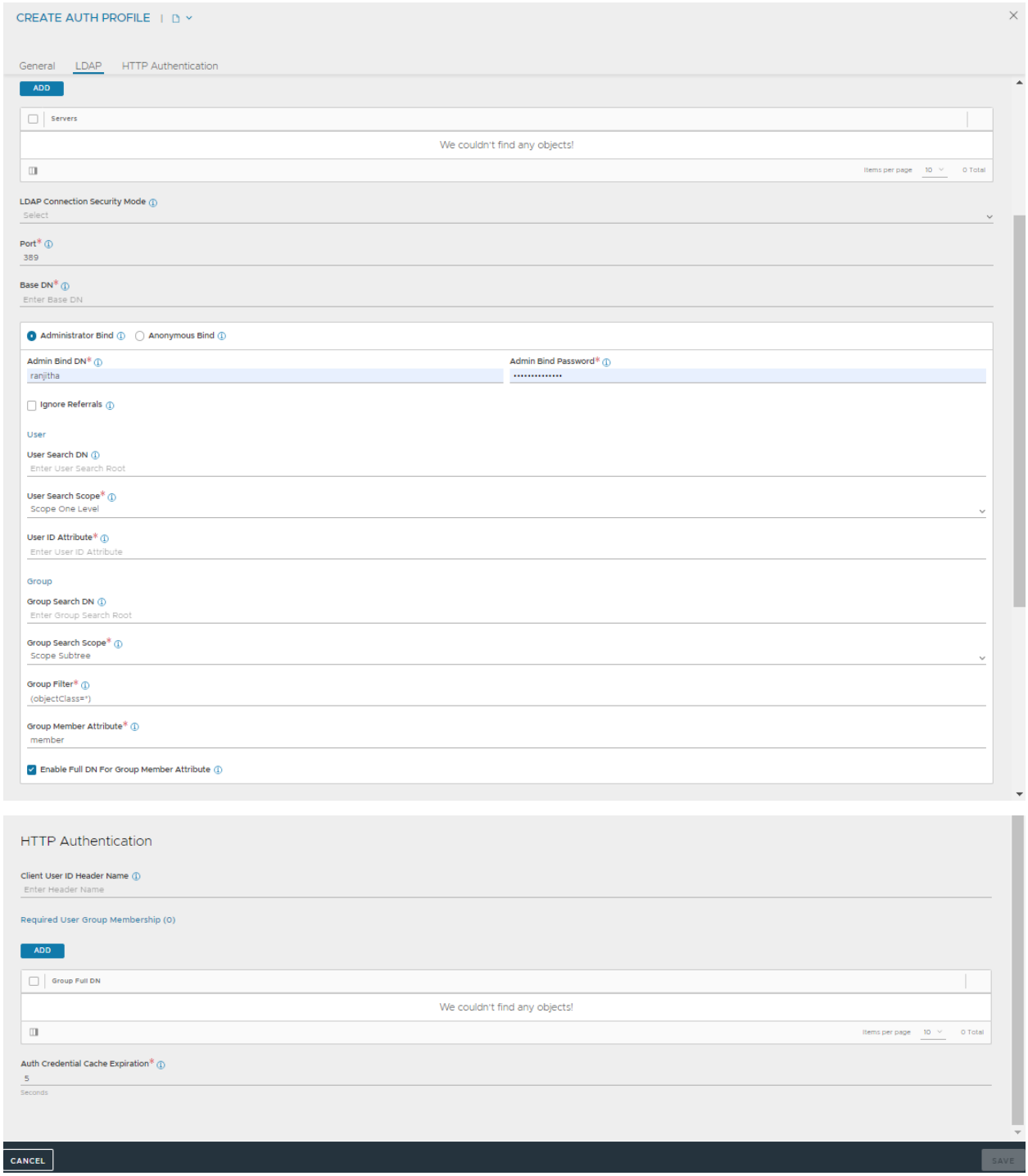
The table on this tab provides the following information for each authorization profile:

- **Name**: Name of the profile.
- **Type**: The Type will be LDAP.

Create an Authentication Profile

To create or edit an authentication profile, do the following:





- **Name**: Enter a unique name.
- **LDAP Servers**: Configure one or more LDAP servers by adding their IP addresses.
- **LDAP Port**: This is the service port to use when communicating with the LDAP servers. This is typically 389 for LDAP or 636 for LDAPS (SSL).
- **Secure LDAP using TLS**: Enable **startTLS** for secure communications with the LDAP servers. This may require a service port change.

- **Base DN**: LDAP Directory Base Distinguished Name. Used as default for settings where DN is required but was not populated like User or Group Search DN.
- **Anonymous Bind**: Minimal LDAP settings that are required to verify User authentication credentials by binding to LDAP server. This option is useful when you do not have access to administrator account on the LDAP server.
  - **User DN Pattern**: LDAP user DN pattern is used to bind an LDAP user after replacing the user token with real user name. The pattern must match the user record path in the LDAP server. For instance, cn=,ou=People,dc=myorg,dc=com is a pattern where we expect to find all user records under ou “People”. When searching LDAP for a specific user, we replace the token with user name.
  - **User Token**: An LDAP token is replaced with real user name in the user DN pattern. For instance, in**User DN Pattern** is configured as “cn=-user-,ou=People,dc=myorg,dc=com”, the token value must be -user-.
  - **User ID Attribute**: LDAP user ID attribute is the login attribute that uniquely identifies a single user record. The value of this attribute must match the user name used at the login prompt.
  - **User Attributes**: LDAP user attributes to fetch on a successful user bind. These attributes are used only for debugging purpose.
- **Administrator Bind**: LDAP administrator credentials configured under **LDAP Directory Settings** below are used to bind NSX Advanced Load Balancer as an admin when querying LDAP for Users or Groups.
  - **Admin Bind DN**: Full DN of LDAP administrator. Admin bind DN is used to bind to an LDAP server. Administrators must have sufficient privileges to search for users under user search DN or groups under group search DN.
  - **Admin Bind Password**: Administrator password. Password expiration or change is not handled. The password is hidden from Rest API and CLI.
  - **User Search DN**: LDAP user search DN is the root of search for a given user in the LDAP directory. Only user records present in this LDAP directory sub-tree are allowed for authentication. Base DN value is used if this value is not configured.
  - **Group Search DN**: LDAP group search DN is the root of search for a given group in the LDAP directory. Only matching groups present in this LDAP directory sub-tree will be checked for user membership. Base DN value is used if this value is not configured.
  - **User Search Scope**: LDAP user search scope defines how deep to search for the user starting from user search DN. The options are search at base, search one level below or search the entire subtree. The default option is to search one-level deep under user search DN.
  - **Group Search Scope**: LDAP group search scope defines how deep to search for the group starting from the group search DN. The default value is the entire subtree.

    - **User ID Attribute**: LDAP user ID attribute is the login attribute that uniquely identifies a single user record. The value of this attribute must match the user name used at the login prompt.
    - **Group Member Attribute**: LDAP group attribute that identifies each of the group members. For instance, member and memberUid are commonly used attributes.
    - **User Attributes**: LDAP user attributes to fetch on a successful user bind. These attributes are only for debugging.
- **Insert HTTP Header for Client UserID**: Insert HTTP header into the client request before it is sent to the destination server. This field is used to name the header. The value will be the client's User ID. This same User ID value will also be used to populate the User ID field in the Virtual Service's logs.
- **Required User Group Membership**: User must be a member of these groups. Each group is identified by the DN. For instance, `cn=testgroup,ou=groups,dc=LDAP,dc=example,dc=com`
- **Auth Credentials Cache Expiration**: The maximum allowed length of time a client's authentication is cached.
- **Group Member Attribute Is Full DN**: Group member entries contain full DNs instead of only User ID attribute values.

Licensing

3

The NSX Advanced Load Balancer software requires a license to enable and utilize the available load balancing features. This section lists the NSX Advanced Load Balancer license editions and describes the steps to obtain, add, and manage them.

Read the following topics next:

- NSX Advanced Load Balancer Editions
- Terms of NSX Advanced Load Balancer Software License
- Removing Trial Licenses from an NSX Advanced Load Balancer Controller
- Updating License on each Node in an NSX Advanced Load Balancer Controller Cluster

NSX Advanced Load Balancer Editions

The following are the various editions supported on NSX Advanced Load Balancer:

- VMware NSX Advanced Load Balancer Enterprise with Cloud Services Edition
- VMware NSX Advanced Load Balancer Enterprise Edition
- VMware NSX Advanced Load Balancer – Basic Edition
- VMware NSX Advanced Load Balancer Essentials for Tanzu

The next section explains the detailed feature list for each of the editions.

Feature Comparison

Categories	Feature	Essentials	Basic	Enterprise	Enterprise with Cloud Services Edition
Local Traffic Management	L4 LB	TCP & UDP Fast Path No SSL/TLS	TCP & UDP Fast Path & ProxyNo L4 SSL/TLS	TCP, UDP, DNS, SIP, RADIUS DSR, TLS support, PROXY protocol support	TCP, UDP, DNS, SIP, RADIUS DSR, TLS support, PROXY protocol support
	L7 LB	X	Limited to basic L7 features and no HTTP/2	HTTP/2, content rewrite, compression, caching	HTTP/2, content rewrite, compression, caching

Categories	Feature	Essentials	Basic	Enterprise	Enterprise with Cloud Services Edition
	L3/4 Policies	X	Policies limited to connection drops No DataScript support	Full-featured, including DataScripts and live IP Reputation	Full-featured, including DataScripts and live IP Reputation
	L7 Policies	X	Basic match and actions	Match on: IP Reputation DB, string groups, and so on Actions to: Rate-limit, ICAP, and so on	Match on: IP Reputation DB, string groups, and so on Actions to: Rate-limit, ICAP, and so on
Global Traffic Management	Global Server Load Balancing	X	X	Enterprise grade GSLB	Enterprise grade GSLB
Application Security	SSL/TLS	X	Very limited feature-set	Dynamic CRLs, OCSP stapling, TLSv1.3, HSM, and cert management	Dynamic CRLs, OCSP stapling, TLSv1.3, HSM, and cert management
	Application Rate Limiting	X	X	Rate limiters can be applied for L4, L7, DNS, and WAF	Rate limiters can be applied for L4, L7, DNS, and WAF
	DDoS Protection	X	X	L3, L4, and L7 DDoS protection	L3, L4, and L7 DDoS protection
	Intelligent WAF	X	X	CRS, learning/PSM, IPReputation, application signatures	CRS, learning/PSM, IPReputation, application signatures, bot detection For more information on bot detection, see *BOT Management Service* topic in the VMware NSX Advanced Load Balancer *Cloud Services Guide*
Container Ingress	Service Type LB	Yes	Yes	Yes	Yes
	Container Ingress	X	Limited Ingress	Full Ingress DNS and GSLB integration Multi K8s cluster & multi-AZ support	Full Ingress DNS and GSLB integration Multi K8s cluster & multi-AZ support

Categories	Feature	Essentials	Basic	Enterprise	Enterprise with Cloud Services Edition
Software Defined Platform	Administration	Basic LB administration	Basic LB administration	Fully multi-tenant & granular RBAC Rich alerts and events Various 3rd part integrations Integrations with various IDPs	Fully multi-tenant & granular RBAC Rich alerts and events Various 3rd part integrations Integrations with various IDPs
	Scale and HA	Active/ Standby only	Active/ Standby only	Active/Active deployments BGP and ECMP support Autoscale of load balancers	Active/Active deployments BGP and ECMP support Autoscale of load balancers
	Flexible upgrades	X	X	Flexible LCM across tenants/clouds/se-groups	Flexible LCM across tenants/clouds/se-groups
	Ecosystem automation	No Access Cloud vCenter	No Access Cloud NSX-T Cloud	Native integrations with all major public & private clouds and container orchestration platforms	Native integrations with all major public & private clouds and container orchestration platforms
	Pulse	X	X	Enhanced case management Live Security Threat Updates	Enhanced case management **Note** Proactive case creation is supported. For more information on Proactive case creation, see *Proactive Support* topic in the VMware NSX Advanced Load Balancer *Cloud Services Guide* Live Security Threat Updates
Advanced Analytics	Application Visibility/ Analytics	X	X	Advanced application telemetry Log insights	Advanced application telemetry Log insights

Functioning of Licensing Editions

The following are a few points on how the licensing editions work in the NSX Advanced Load Balancer Controller:

- The Enterprise with Cloud Services Edition is the default licensing tier for an NSX Advanced Load Balancer Controller. A new Controller is set up to be in the Enterprise with Cloud Services Edition licensing tier.
- Customers can request trial service units (saas units) valid for up to 90 days to try out features in the Cloud Services edition. Contact the NSX Advanced Load Balancer representative for more details.

- VMware continues to ship trial licenses for the Enterprise edition as part of the packaging. The customer must change the license tier to Enterprise to utilize these licenses.
- The NSX Advanced Load Balancer can be set up in any one of the licensing tier. However, the entire Controller cluster operates in the same license edition tier to which all applications and SEs belong..
- The License tier for the Controller can be changed from one edition to another without any disruptions of traffic or downtime.

NSX Advanced Load Balancer Enterprise with Cloud Services Edition

The NSX Advanced Load Balancer with Cloud Services provides multi-cloud load balancing, web application firewall, application analytics, and container services from the data center to the cloud, with enhanced operations delivered through SaaS.

For more information, see VMware NSX Advanced Load Balancer Cloud Services Guide.

NSX Advanced Load Balancer - Enterprise Edition

The new license enforcement model enables product enhancement to incorporate unified licensing around Service Units.

The Enterprise Edition license is based on Service Units. On upgrade to NSX Advanced Load Balancer release 20.1.1, the licenses prior to NSX Advanced Load Balancer release 20.1.1 are converted to Service Unit licenses.

If you are upgrading from version 18.x to NSX Advanced Load Balancer version 22.1.3, use the following table to understand how the various license types are converted to the Enterprise Edition licenses.

License prior to NSX Advanced Load Balancer 20.1.1 (1 unit)	Corresponding Service Unit License
Socket	5
Core (Avi Vantage Core - Prior to 2020)	1
Host	1
25 Mbps Bandwidth	0.4
200 Mbps Bandwidth	0.7

Depending on the configuration of NSX Advanced Load Balancer, the appropriate number of Service units are consumed by the SEs.

The configuration depends on the following parameters, which are explained in the later sections:

- `Bandwidth restriction`
- `Per-app mode`
- `Virtual Machines parameters`

- `Bare-metal parameters`
- `Cloud-specific configuration`

The license is downloaded from the NSX Advanced Load Balancer web portal and is administered centrally by the NSX Advanced Load Balancer Controller to all SEs. Licenses are consumed by the SEs and not by the Controllers.

The following environments support NSX Advanced Load Balancer - Enterprise Edition:

- Azure VMware Solution
- Google Cloud VMware Engine
- Oracle Cloud VMware Solution
- VMware Cloud on AWS

The license edition provides the following advantages:

- Aligns NSX Advanced Load Balancer with VMware price list
- Simplifies commercial model
- Provides flexibility – Same Service Unit license can be used on any ecosystem and using any configuration
- Simplifies reporting and projection

Enterprise Edition License

The Enterprise Edition is a fully-featured NSX Advanced Load Balancer license that includes load balancing, GSLB, WAF, and all other features available on NSX Advanced Load Balancer.

Note

- Future-dated license is not supported.
- Enterprise Edition comes with two units of a free perpetual license. This license does not include product support and is only intended for PoC or trial use.

License Duration

NSX Advanced Load Balancer is available in the following license types, based on the duration for which the license is valid.

License Type	Description
Term	■ Available for 1-year and 3-year pre-paid terms. ■ It has all the features such as load balancing, GSLB, WAF, and so on. ■ Features related to Cloud Services are not available (Cloud Services features such as Centralized Licensing are available in Subscription SKUs only). ■ Support included.
Perpetual	■ Perpetual License with features like Term. ■ Also needs additional per-year Support and Subscription (SnS). **Note** Perpetual SKUs are deprecated in December 2021.
Subscription	■ Available for 1-year and 3-year pre-paid terms. ■ Includes all features and the NSX Advanced Load Balancer Cloud Services features (such as Centralized Licensing). ■ Includes Support.

Obtaining an NSX Advanced Load Balancer License

NSX Advanced Load Balancer licenses can be downloaded by authorized persons from the customer portal or obtained by contacting the NSX Advanced Load Balancer sales team.

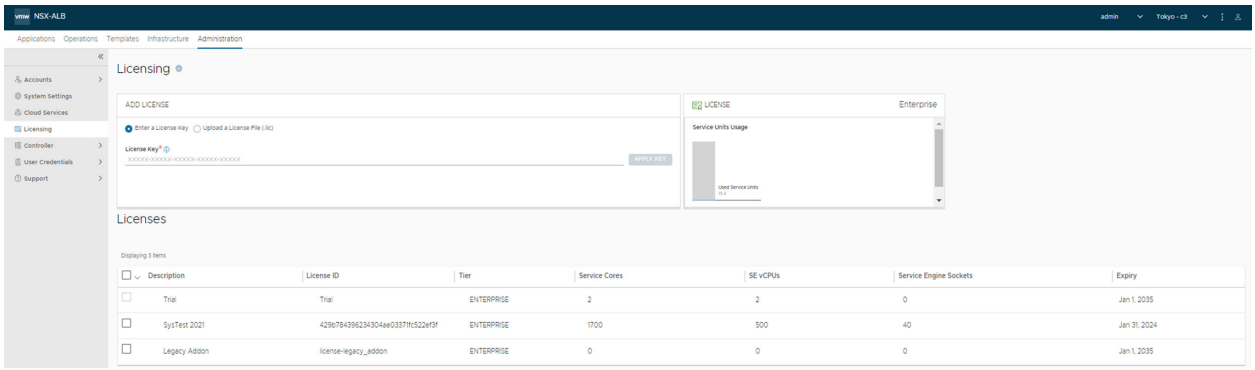
To activate a license, you must first add the license to the NSX Advanced Load Balancer Controller.

Adding a New License

You can add a new license using one of the following options:

Using VMware Serial Key

You can activate the license manually by entering the VMware serial key.



After adding a 25-character VMware serial key, the system will display information about the license, such as start date, expiry date, resource count, and so on, available in the NSX Advanced Load Balancer Controller license list.

Note Subscription-based serial keys with more than 90-day validity can have trouble functioning on the Controllers running 18.2.10 or lower versions.

The start date of such licences appears to be in the future, even when the serial key is correctly added. For instance, if the serial key is issued on 01-JAN-2021 for a 1-year validity, the start date can appear as 01-OCT-2021. However, the expiration date will be accurately set to 31-DEC-2021. You must update to version 18.2.11 or later to fix it.

You must reconfigure (by deleting the added but unusable serial key) such future-dated serial keys after the upgrade.

Using NSX Advanced Load Balancer License File

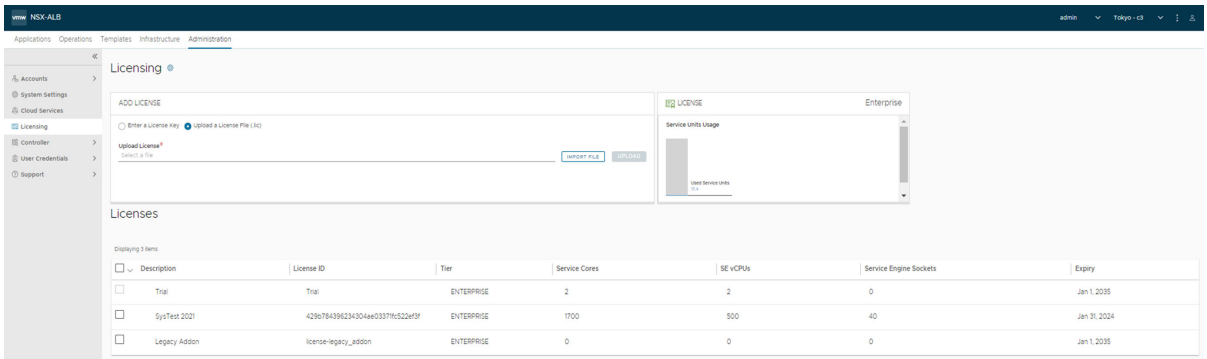
After obtaining a license from NSX Advanced Load Balancer, you can activate the license by adding it to the NSX Advanced Load Balancer Controller.

Procedure

1 Navigate to **Administration > Licensing**.

2 Click **Upload a License File (.lic)**.

3 Click **IMPORT FILE** and navigate to the file.



4 After the license file is uploaded, the new license appears in the license list of the Controller.

Results

The system displays information about the license, including the start date and expiration date.

Divide or Combine Licenses

You can divide or combine VMware serial key licenses.

For more information on divide and combine licenses, see How to Divide or Combine License Keys guide.

Note If you are using old YAML-based licenses generated by the NSX Advanced Load Balancer, contact your accounts team or VMware support for the new ones.

Various Parameters and Configuration Impacting License Consumption

The following is the list of configurations and parameters that impact Enterprise Edition license allocation:

- Bandwidth mode licensing
- Per-App mode licensing
- Bare-metal server licensing
- Public cloud licensing
- Azure cloud licensing

Bandwidth Mode Licensing

Bandwidth mode licensing is used to restrict the use of bandwidth on SEs. Using this licensing method, the number of datapath processes on SEs is restricted. The SEs no longer require a separate bandwidth-based license to enable the bandwidth mode. When the bandwidth mode is enabled on the SEs, the bandwidth is consumed from the central pool of licenses, which is the capacity of the Service Units license.

Note

- 1 Gbps bandwidth mode is no longer available.
- Per-App mode is not supported on SE Groups when the Bandwidth mode is enabled.

Bandwidth Mode	Consumed License of Service Units	Number of Datapath Processes
25 Mbps	0.4	1
200 Mbps	0.7	2

Per-App Mode Licensing

In the Per-App mode licensing, only two virtual services are supported per SE.

The key points regarding Per-app Mode licensing are as follows:

- It is not supported for SEs enabled with the Bandwidth Mode.
- It is not supported by the Azure Marketplace.
- It cannot be used to host DNS Virtual Services (both standalone DNS and GSLB).

- 0.25 Service Units license is required to license one core of load balancing license.
  - For an SE VM with eight physical cores, two Service Units licenses are required for the load balancing.
- For SNI or EVH shared virtual services, Per-App is limited to only one parent and one child virtual service.

Bare-Metal Server Licensing

The following are the key points regarding Bare-Metal server licensing:

- The socket license is no longer available.
- When you upgrade a bare-metal server, one socket license gets converted to five Service Unit licenses.
- Bare-metal servers are also licensed following the Service units.
- The number of Service Unit licenses consumed by a bare-metal server depends on the following:
  - the number of sockets associated with the bare-metal server
  - the number of cores associated with each socket
- Calculating Service Units license for various bare-metal servers based on their configuration:
  - For 16 cores, dual-socket, bare-metal server (16 cores on each socket), the following is applicable:
    - For each socket, and five licenses are required, as up to 20 cores, five Service Units licenses are required for one socket.
  - After the limit of 20 cores per socket, one Service Units license for every four additional cores is required.
    - For a bare-metal server configured with 24 cores, six Service Units licenses are required.
    - For a bare-metal server configured with 28 cores, seven Service Units licenses are required.
      - For a dual-socket, 28 core bare-metal servers, and 14 Service Units licenses are required.

Public Cloud Licensing

The following are the key points regarding the public cloud licensing for the SEs:

- 1 Gbps bandwidth SKU is no longer available.
- Hyper-threaded vCPUs are not accounted for licensing, regardless of the ecosystems.
- Ability to define custom number of datapath processes (number of cores used for load balancing). It allows for cost-effective use of larger instances in a public cloud.

- You can change bandwidth configuration in SE Group even when it has SEs in it.
  - SE might have to be rebooted for the bandwidth configuration to take effect.
- The restrictions based on the instance flavor has been removed.

Azure Marketplace Licensing

The Enterprise Edition license impacts the Azure Marketplace licensing or Azure pay-as-you-go (Azure PAYG) licensing in the following ways:

- 1 Gbps bandwidth SKU is not supported.
- Supported license types are as follows:
  - 25 Mbps bandwidth license
  - 200 Mbps bandwidth license
- Existing SEs on 1 Gbps SKU will continue to be billed following the older SKU.
- The bandwidth restriction for the SEs is removed.
- New SEs creation with 1 Gbps SKU is not supported.
- Instance-level restriction is removed. Any instance type can be used to deploy an NSX Advanced Load Balancer Controller.

For the PAYG licenses, the recommended sizes for an NSX Advanced Load Balancer SE are as follows:

License Unit	Maximum vCPU Allowed for Service Engine	Recommended VM Sizes	Other Allowed VM Sizes
25 Mbps	2	F2s_V2, DS2_v3	Any VM <= 2 vCPUs
200 Mbps	4	F4s_v2, DS4_v3	Any VM <= 4 vCPUs

For more information on PAYG licenses, see *Azure Pay-as-you-go (Azure PAYG)* section in the VMware NSX Advanced Load Balancer*Installation Guide*.

Updating an Existing License

Updating an existing license follows the same process as adding a new license. When a license file has the same license ID (with an updated limit and expiration date), uploading this license file will update the existing entry.

Upgrading to Enterprise Edition

All the previously deployed licenses are converted to equivalent Service unit-based licenses when an NSX Advanced Load Balancer Controller is upgraded.

The points to consider while upgrading to an Enterprise Edition license are as follows:

- The number of licenses consumed will be based on the number of SE_DPs running and not the number of cores on the SE.

- SEs associated with the Controller before the upgrade will continue to work. After the upgrade, all previous license types will be migrated to Service Units licenses. The number of consumed Enterprise Edition licenses is based on the number of SE_DPs running on the SEs and not on the number of cores on the SEs. Modification of the number of datapath processes for the SE is supported.
- A 1 Gbps bandwidth license is no longer available.
- The Enterprise Edition license supports a 10% buffer, over and above the licensed resources on the Controller. This 10% buffer is relative to the Service Units license available on the Controller. It is recommended to use the buffer allowance temporarily and for emergency requirements like a scale-out event.
    - For a Controller with 20 Service Units licenses, the effective Service Units after adding the buffer allowance is 22 Service Units (20 + 2).

Impacts on Service Engine Group

For SE groups, all the existing Enterprise_18 edition licenses are migrated to the Enterprise Edition license. All the existing license types are migrated to the Service Units licenses, except for pay-as-you-go (Azure marketplace).

A 1 Gbps bandwidth is set to unlimited, and the corresponding number of SE_DPs is set to `4`. The 25 Mbps bandwidth and 200 Mbps bandwidth licenses are not migrated.

Impact on Cloud Objects

The default licensing tier for the NSX Advanced Load Balancer Controller cluster is updated to Enterprise.

Migration of Service Engines using 1 Gbps Bandwidth

One unit of 1 Gbps bandwidth license is converted to four units of Service Units licenses.

The following are applicable to the SE groups on which the SEs (with 1 Gbps bandwidth license) are running.

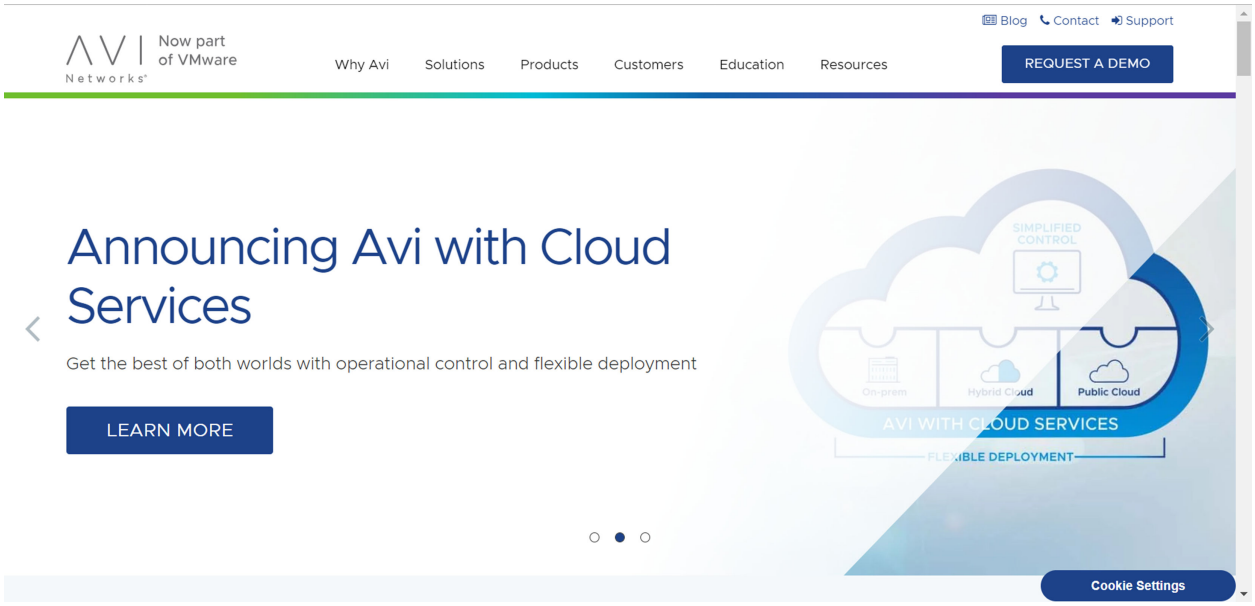
- The bandwidth mode is set to unlimited.
- Bandwidth restriction is removed.
- SE is set to consume only four Service Units licenses, as the value of the `max_se_dp` setting on the SE Group is set to `4`. The number of cores used for load balancing is set to `4`. Therefore, a maximum of four Service Units licenses are used.
- 1 Gbps license is no longer available.

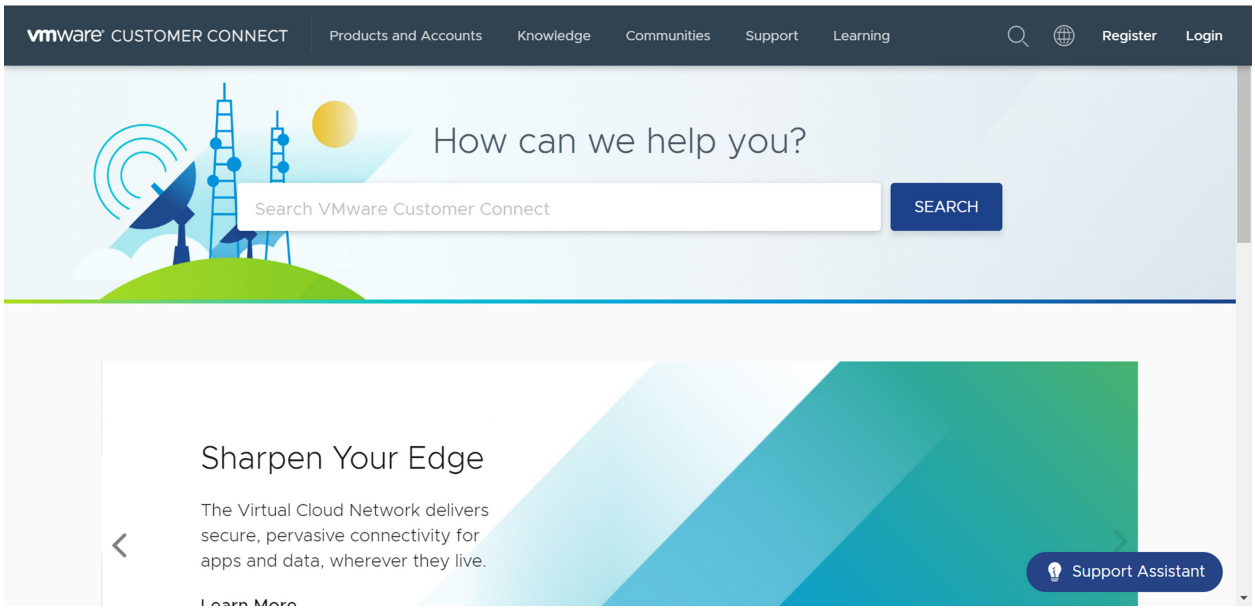
VMware Customer Connect Portal Self-Registration

This section lists step-by-step instructions to self-register using the customer portal.

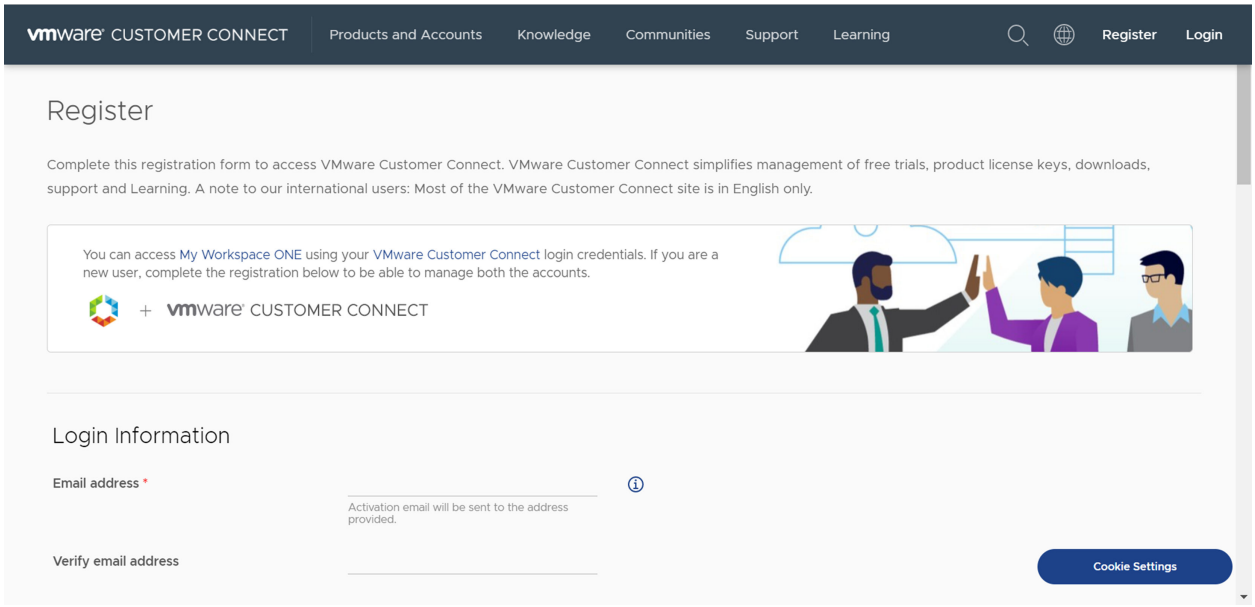
Registration

- Navigate to the home page and click **Support**. This guides you to the **VMware Customer Connect**portal. Click **Register** to navigate to the **Login Information** page.





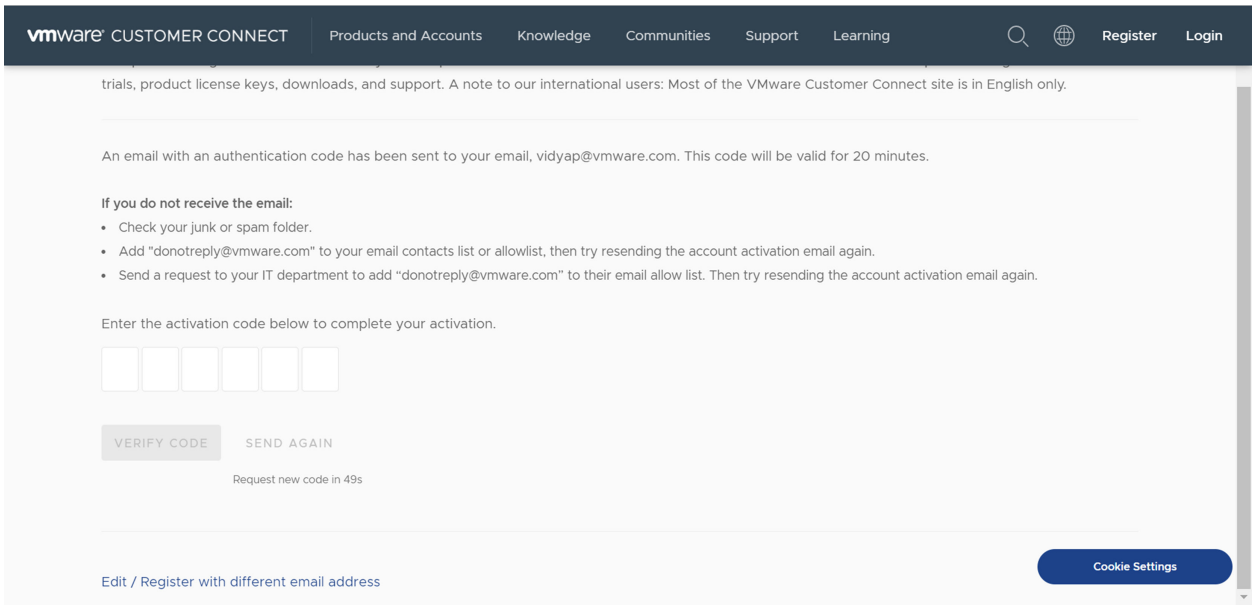
- Fill in the required details in the **Login Information** page.



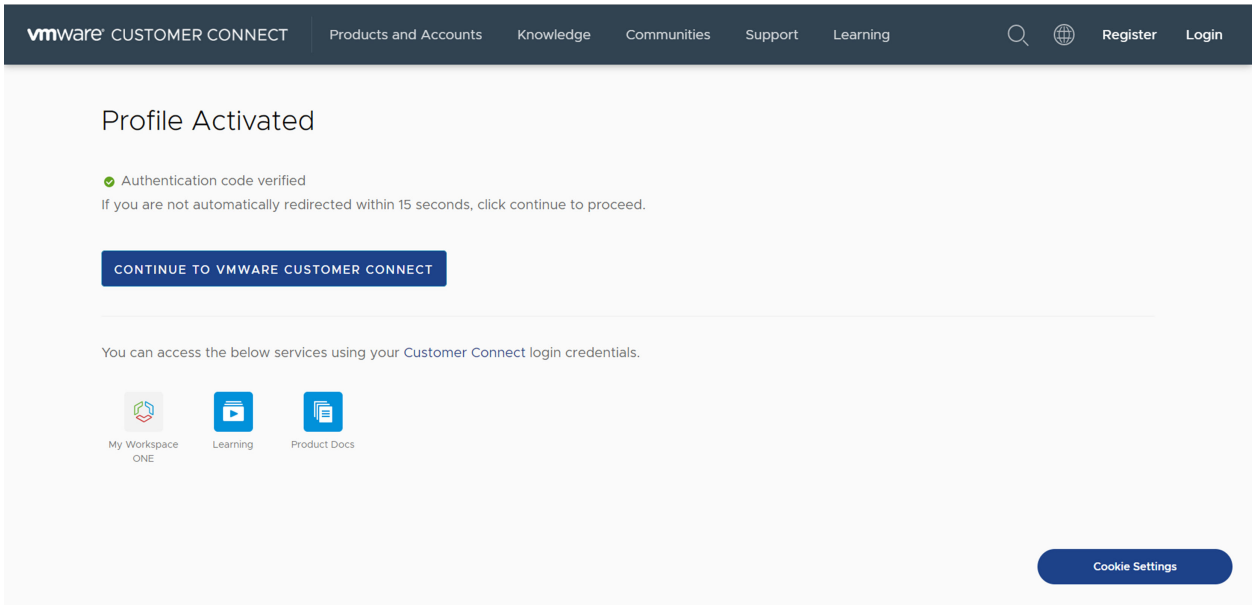
- Fill out the mandatory fields and click **REGISTER**.
- Upon submission of the form, an activation email is sent to the provided email address.

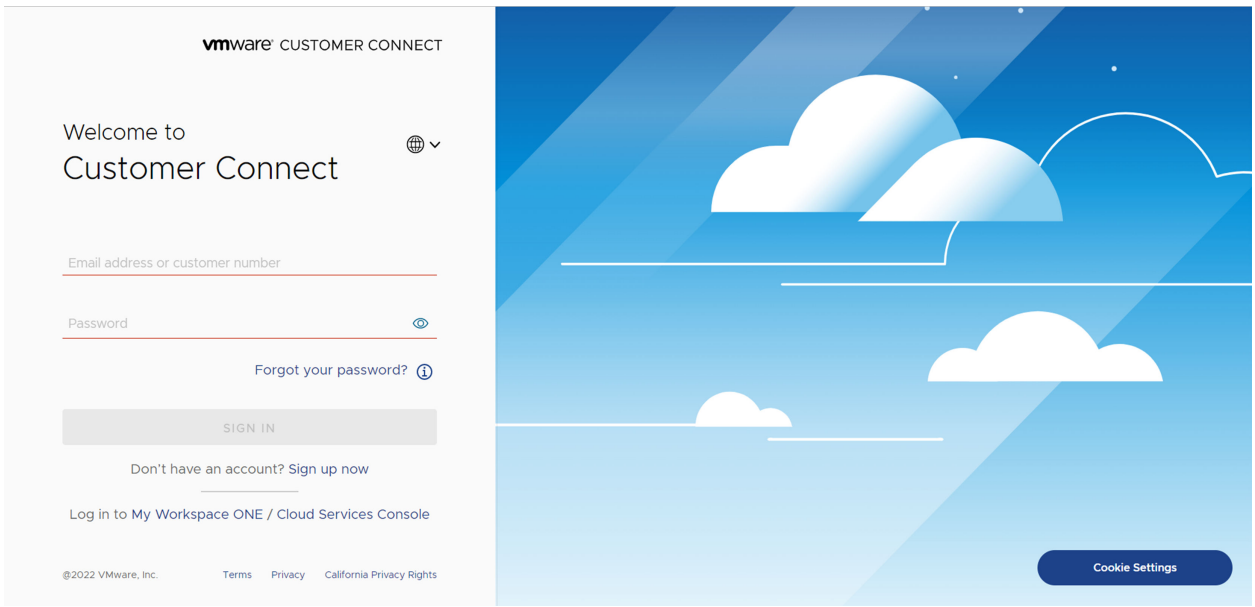
Activation

- Type the Authentication Code received in the email, in the space provided.



- If activation is successful, you are redirected to the **Profile Activated** page and subsequently to the **vmware Customer Connect** login page as shown below.





Login using the registered email id and password to access the **VMware Customer Connect** portal.

Additional Configuration Options

The following are the additional configuration options:

Limiting Number of Service Units Used by SEs

The number of Service Units used for the load balancing can be customized by using the `max_num_se_dps` option on the SE group. Customization of the number of cores for the load balancing feature enables the use of larger VM sizes but with fewer cores for load balancing.

The use of a customized core has the following benefits:

- Saving on licensing costs.

- Useful in cases where a larger VM is required for better packet per second (PPS) or VIP/VS density.

SE creates one datapath process per core. To restrict the number of cores used for load balancing, set the `max_num_se_dps` setting to the desired value on the SE Group. The `max_num_se_dps` parameter represents all cores on the VM that is used for load-balancing and its default value is zero.

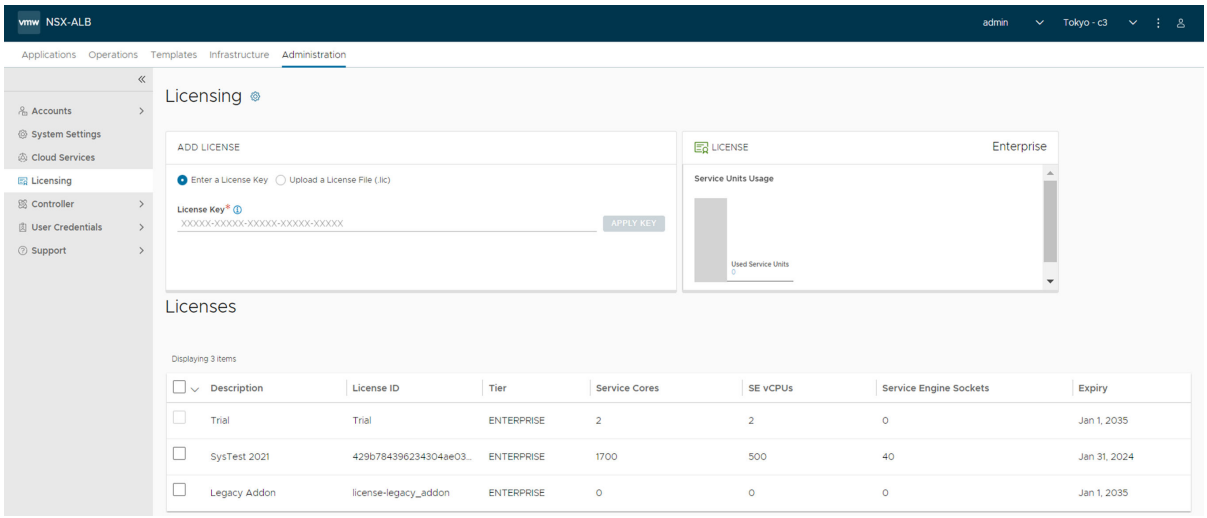
Note

- The configuration of the `max_num_se_dps` parameter is supported on all the ecosystems. The number of Service Units licenses assigned to the SE depends on the number of `SE_DPs` running. By default, an eight-core SE will consume eight Service Units licenses. However, if `max_num_se_dps` is set to `4`, even on an eight-core SE VM, only four Service Units licenses are consumed.
- When an SE group with a 1 GB bandwidth limit is migrated, the bandwidth limit is set to unlimited. The value of the `max_num_se_dps` attribute is set to `4`. For all the other SE Groups, the value of `max_num_se_dps` is set to `0`.
- Hyper-threaded cores are not accounted for licensing on all the ecosystems. For more information on Service Engine groups, see *Service Engine Group* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Checking License Usage

To check license usage information:

1 Navigate to **Administration > Licensing**.



The system displays the license limits and current usage.

Note To check license usage information using CLI, use `show license ledger details` command.

Reaching the License Limit

When a license limit is reached, you can neither create nor register new NSX Advanced Load Balancer SEs with the NSX Advanced Load Balancer Controller. However, you can use the existing SEs for existing or new virtual services.

In the case of container cloud environments, after the license limit has been reached, you cannot create new SEs even if you add new hosts to the cluster. Any new tasks or containers created on new container hosts can have degraded access to the existing load-balanced applications because SEs are not present on those hosts.

License Expiration

If multiple licenses are attached to the NSX Advanced Load Balancer, and one of the licenses expire, the system immediately prevents the capacity from exceeding the total resources allotted by the remaining valid licenses.

When all applied licenses expire, the NSX Advanced Load Balancer returns to an unlicensed state. In the unlicensed state, the following changes become applicable:

- There is no impact on existing virtual services.
- You cannot create new SEs.
- You can still create and edit virtual services and other load-balancing configurations.
- The system defaults to the free, perpetual limits.
- When a valid license is reapplied, the system will immediately be able to utilize the total resources allocated by the new license.

New licenses can be managed and downloaded from the Customer Portal. The interval for expiration notification is two months.

License Expiry Events

The NSX Advanced Load Balancer automatically generates a **LICENSE_EXPIRY** event once every day, starting one month before expiry. The events are included in the events table.

Procedure

1. Navigate to **Operations > Events > All Events**.
2. Search for **license_expiry** under **Event Code (event_id)** in the table.
3. click the **+** sign at the right of the row.

Results

The details of the particular event are displayed.

License Expiry Alerts

In addition to system-generated license expiry events, you can also configure the NSX Advanced Load Balancer to generate license-expiration alerts. Based on the alert configuration, the alerts

are triggered by license expiration events and are defined as you prefer, namely, through emails, Syslog messages, SNMP traps, and ControlScripts.

FAQs on Expired License

This section answers FAQs on expired NSX Advanced Load Balancer licenses.

1 Why does the NSX Advanced Load Balancer generate alerts for the expired license even though the valid license is applied?

   Each license present on NSX Advanced Load Balancer is considered as a separate object. The NSX Advanced Load Balancer will generate alerts for both types of licenses – active license and expired license. To stop generation of alerts for the expired licenses, delete or uninstall the expired licenses installed on NSX Advanced Load Balancer.

2 What is the procedure to delete the expired licenses?

   Follow the steps mentioned below to delete the expired license:

   a Navigate to **Administration > Licensing**.

   b Select the expired license from the list, and click **Delete** to uninstall the expired license.

3 Is it safe to delete the expired license?

   Yes, it is safe to delete the expired licenses from the NSX Advanced Load Balancer.

Configuration Objects Restricted in the NSX Advanced Load Balancer UI for Essentials and Basic License

This section elaborates the configuration objects restricted in the NSX Advanced Load Balancer UI for Essentials and Basic License.

NSX Advanced Load Balancer supports the following license types:

- VMware NSX Advanced Load Balancer Enterprise Edition - It is a fully-featured VMware NSX Advanced Load Balancer license, which includes load balancing, GSLB, WAF, and all other features available on NSX Advanced Load Balancer.
- VMware NSX Advanced Load Balancer - Basic Edition
- NSX Advanced Load Balancer Essentials for Tanzu - VMware NSX Advanced Load Balancer essentials for Tanzu provides Layer 4 load balancing services for Tanzu Basic and Standard users.

Note The Basic Edition of Avi Load Balancer (formerly known as NSX Advanced Load Balancer) has reached its End of Availability (EoA) date on Jan 30th, 2024. The changes only apply to the Basic Edition of Avi Load Balancer. For additional details, please visit VMware Knowledge Base .

In VMware NSX Advanced Load Balancer essentials for Tanzu, a few configuration options are restricted or not editable in the NSX Advanced Load Balancer UI. Some configuration knobs allow the create and edit options using UI in the essentials edition, whereas others only have the view access.

NSX Advanced Load Balancer UI Object List for Essentials, Basic, and Enterprise Edition

See the below table for more information on permissions available for various UI objects in VMware NSX Advanced Load Balancer essentials for Tanzu, VMware NSX Advanced Load Balancer - Basic Edition, and VMware NSX Advanced Load Balancer Enterprise Edition.

Menu	Sub Menu	Permissions(Essentials)*	Permissions(Basic)*	Permissions(Enterprise)
Applications	Dashboard	View	View	View
	Virtual Service	View	Create/Edit/View	Create/Edit/View
	VS VIPs	View	Create/Edit/View	Create/Edit/View
	Pools	View	Create/Edit/View	Create/Edit/View
	Pool Groups	View	Create/Edit/View	Create/Edit/View
Operations	Alerts - All Alerts	View	Create/Edit/View	Create/Edit/View
	Alerts - Alert Config	View	Create/Edit/View	Create/Edit/View
	Alerts - Alert Actions	View	Create/Edit/View	Create/Edit/View
	Events - All Events	View	Create/Edit/View	Create/Edit/View
	Events - Config Audit Trail	View	Create/Edit/View	Create/Edit/View
	Notifications - Syslog	View	Create/Edit/View	Create/Edit/View
	Notifications - Email	View	Create/Edit/View	Create/Edit/View
	Notifications - SNMP Trap	View	Create/Edit/View	Create/Edit/View
	Traffic Capture - Traffic Capture	Create/Edit/View	Create/Edit/View	Create/Edit/View
Templates	Profile - Application	View	Create/Edit/View	Create/Edit/View
	Profile - TCP/UDP	View	Create/Edit/View	Create/Edit/View
	Profile - Persistence	View	Create/Edit/View	Create/Edit/View
	Profile - Health Monitor	View	Create/Edit/View	Create/Edit/View
	Profile - Analytics	View	Create/Edit/View	Create/Edit/View
	Profile - IPAM/DNS Profiles	Create/Edit/View	Create/Edit/View	Create/Edit/View

Menu	Sub Menu	Permissions(Essentials)*	Permissions(Basic)*	Permissions(Enterprise)
	Profile - Custom IPAM/DNS	View	Create/Edit/View	Create/Edit/View
	Profile - Traffic Clone	View	Create/Edit/View	Create/Edit/View
	Profile - ICAP Profile	View	Create/Edit/View	Create/Edit/View
Templates	Policies - NAT Policy	View	Create/Edit/View	Create/Edit/View
	Policies - L4 Policy Set	View	Create/Edit/View	Create/Edit/View
Templates	Groups - IP Groups	View	Create/Edit/View	Create/Edit/View
	Groups - String Groups	View	Create/Edit/View	Create/Edit/View
Templates	Security - SSL/TLS Certificates	Create/Edit/View	Create/Edit/View	Create/Edit/View
	Security - SSL/TLS Profile	View	Create/Edit/View	Create/Edit/View
	Security - PKI Profile	View	Create/Edit/View	Create/Edit/View
	Security - Auth Profile	View	Create/Edit/View	Create/Edit/View
	Security - JWT Server Profile	View	Create/Edit/View	Create/Edit/View
	Security - Ping Access Agent	View	Create/Edit/View	Create/Edit/View
	Security - Certificate Management	View	Create/Edit/View	Create/Edit/View
	Security - IP Reputation	View	Create/Edit/View	Create/Edit/View
	Security - Geo DB	View	Create/Edit/View	Create/Edit/View
	Security - HSM Group	View	Create/Edit/View	Create/Edit/View
	Security - SSO Policy	View	Create/Edit/View	Create/Edit/View
	Security - Bot Management Policy	View	Create/Edit/View	Create/Edit/View
Templates	Scripts - DataScripts	View	Create/Edit/View	Create/Edit/View
	Scripts - ControlScripts	View	Create/Edit/View	Create/Edit/View
	Scripts - ProtocolParserScripts	View	Create/Edit/View	Create/Edit/View
Templates	Autoscale - Server Autoscale	View	Create/Edit/View	Create/Edit/View

Menu	Sub Menu	Permissions(Essentials)*	Permissions(Basic)*	Permissions(Enterprise)
	Autoscale - Launch Config	View	Create/Edit/View	Create/Edit/View
	Autoscale - WebHook	View	Create/Edit/View	Create/Edit/View
	Autoscale - PoolGroup Deployment	View	Create/Edit/View	Create/Edit/View
Templates	WAF - WAF Policy	View	Create/Edit/View	Create/Edit/View
	WAF - WAF Profile	View	Create/Edit/View	Create/Edit/View
	WAF - Application Rules	View	Create/Edit/View	Create/Edit/View
	WAF - Application Rules	View	Create/Edit/View	Create/Edit/View
	WAF - CRS	View	Create/Edit/View	Create/Edit/View
Templates	Error Page - Error Page Profile	View	Create/Edit/View	Create/Edit/View
	Error Page - Error Page Body	View	Create/Edit/View	Create/Edit/View
Infrastructure	Dashboard	View	View	View
	Clouds	Create/Edit/View	Create/Edit/View	Create/Edit/View
	Cloud Resources - Service Engine	View	Create/Edit/View	Create/Edit/View
	Cloud Resources - Service Engine Group	Create/Edit/View	Create/Edit/View	Create/Edit/View
	Cloud Resources - Networks	Edit/View	Create/Edit/View	Create/Edit/View
	Cloud Resources - VRF Context	Create/Edit/View	Create/Edit/View	Create/Edit/View
	GSLB - Site Configuration	View	Create/Edit/View	Create/Edit/View
Administration	Accounts - Users	View	Create/Edit/View	Create/Edit/View
	Accounts - User Profiles	View	Create/Edit/View	Create/Edit/View
	Accounts - Roles	View	Create/Edit/View	Create/Edit/View
	Accounts - Tenants	View	Create/Edit/View	Create/Edit/View
Administration	Settings – Authentication/ Authorization	Edit/View	Edit/View	Edit/View

Menu	Sub Menu	Permissions(Essentials)*	Permissions(Basic)*	Permissions(Enterprise)
	Settings – Access Settings	Edit/View	Edit/View	Edit/View
	Settings – DNS/NTP	Edit/View	Edit/View	Edit/View
	Settings – Licensing	Edit/View	Edit/View	Edit/View
	Settings – Email/ SMTP	Edit/View	Edit/View	Edit/View
	Settings – Tenancy Mode	Edit/View	Edit/View	Edit/View
	Settings – Upload HSM Packages	Edit/View	Edit/View	Edit/View
	Settings – DNS Service	View	Edit/View	Edit/View
	Settings – Cloud Services	View	Create/Edit/View	Create/Edit/View
Administration	Controller - Nodes	Edit/View	Edit/View	Edit/View
	Controller - Events	View	Create/Edit/View	Create/Edit/View
	Controller - Alerts	View	Create/Edit/View	Create/Edit/View
	Controller - Software	View	Create/Edit/View	Create/Edit/View
	Controller - System Update	Edit/View	Create/Edit/View	Create/Edit/View
	Controller - SEG Update	View	Create/Edit/View	Create/Edit/View
Administration	System - Configuration Backup	Edit/View	Edit/View	Edit/View
	System - Tech Support	Create/Edit/View	Create/Edit/View	Create/Edit/View
	System - Crash Reports	View	Create/Edit/View	Create/Edit/View
Administration	User Credential - User Credentials	View	Create/Edit/View	Create/Edit/View
Administration	Support - Cases	View	Create/Edit/View	Create/Edit/View

Note The above table lists all the UI options, but not all features are supported with the Essentials and Basic editions.

Some of the examples are as follows:

- WAF and GSLB are not supported with the essentials edition and the Basic Edition, but WAF and GSLB are available with the Enterprise edition.
- Layer 7 load balancing is not available with the Essentials edition, limited capability for Layer 7 load balancing is available with the Basic edition, and fully-featured Layer 7 support is available with the Enterprise edition.
- Similarly, only the vCenter cloud and No Orchestrator clouds can be created or edited with the essentials and the Basic editions. Whereas the Enterprise Edition supports clouds, like Google, Azure, AWS, and Oracle.
- Only HTTP and Layer 4 profiles can be created or edited in the Basic edition.
- The GSLB Services option is available under the **Applications** tab only when the GSLB feature is configured on the NSX Advanced Load Balancer Controller. GSLB is available with the Enterprise edition only.

In the UI, you might be able to edit or view the options, but to check if a feature is supported for your NSX Advanced Load Balancer edition, see NSX Advanced Load Balancer Editions.

NSX Advanced Load Balancer - Basic Edition

This topic explains how the NSX Advanced Load Balancer - Basic edition is deployed.

Note

- The Basic Edition of Avi Load Balancer (formerly known as NSX Advanced Load Balancer) has reached its End of Availability (EoA) date on Jan 30th, 2024. The changes only apply to the Basic Edition of Avi Load Balancer. For additional details, please visit VMware Knowledge Base.
- NSX Advanced Load Balancer-Basic edition is not available as a separate SKU and is not saleable.
- NSX Advanced Load Balancer-Basic Edition supports the following deployment modes:
  - NSX-T Cloud Connector
  - No Orchestrator Mode

For more information on setting up NSX Advanced Load Balancer - Basic Edition, see Installing NSX Advanced Load Balancer - Basic Edition.

Feature Comparison (Essentials vs Basic vs Enterprise)

For more information on the feature list for each of the editions, See NSX Advanced Load Balancer Editions.

Installing NSX Advanced Load Balancer - Basic Edition

This topic explains how the NSX Advanced Load Balancer - Basic edition is installed.

Installing the NSX Advanced Load Balancer Controller

The NSX Advanced Load Balancer Controller OVA can be downloaded from My VMware customer portal. SE creation will be handled by the NSX Advanced Load Balancer Controller.

Note The Basic Edition of Avi Load Balancer (formerly known as NSX Advanced Load Balancer) has reached its End of Availability (EoA) date on Jan 30th, 2024. The changes only apply to the Basic Edition of Avi Load Balancer. For additional details, see VMware Knowledge Base.

Deploying NSX Advanced Load Balancer Controller

Log in to the vCenter server through a vSphere Web Client. Use the client to deploy NSX Advanced Load Balancer Controller OVA file by following the steps mentioned below:

1 Click the **File** option in the top menu and select **Deploy OVF Template**.

2 Follow the **Deploy OVA Template wizard** instructions:

   a Select **Thick Provision Lazy Zeroed** for disk format.

   b Select a port group for Destination Networks in **Network Mapping**. This port group will be used by the NSX Advanced Load Balancer Controller to communicate with vCenter.

   c Specify the management IP address and default gateway. In the case of DHCP, leave this field empty (Only static IP addresses must be used in production environment).

   d Leave the **Sysadmin login authentication key** field blank.

3 Power on the VM.

Repeat the above steps to deploy three NSX Advanced Load Balancer Controller VMs to create an NSX Advanced Load Balancer Controller cluster set-up.

Performing the NSX Advanced Load Balancer Controller Initial Setup

You can change or customize settings following initial deployment using the NSX Advanced Load Balancer ControllerUI. Navigate to the NSX Advanced Load Balancer Controller IP on the browser.

Note While the system is booting up, a 503 status code or a page with following message will appear:

- `Controller is not yet ready. Please try again after a couple of minutes.` Wait for about 5 to 10 minutes and refresh the page. Follow the instructions below for the set-up wizard.

To perform the NSX Advanced Load Balancer Controller initial setup:

- Configure the basic system settings:
  - Administrator account.

    Note A valid email address is required for the admin password reset in case of user account lockout.

  - DNS and NTP server information
  - Configure Email or SMTP information
- Select the **No Orchestrator** option for Orchestrator Integration.
- Select **No** for **Support multiple Tenants?** to complete the set-up.

Preparing NSX Advanced Load Balancer - Basic Edition

The NSX Advanced Load Balancer Controller boots up in the Enterprise Edition by default. Delete the features that are not supported in the Basic Edition before changing the license tier.

Perform Configuration Audit

Run the `configuration audit` API to identify the Enterprise Edition features that need to be removed.

All configuration violations need to be corrected before license tier change is allowed.

The following is a sample output of the `configuration audit` API performed right after a fresh install.

```
[admin:cntrl]: > show configuration audit tier basic
+--------------------+---------------
+-------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
| Object Type        | Name          | License
Violations
                                            |
+--------------------+---------------
+-------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
| ServiceEngineGroup | Default-Group | Field ServiceEngineGroup.ha_mode cannot have
HA_MODE_SHARED as its value in BASIC license tier. Allowed value(s):
HA_MODE_LEGACY_ACTIVE_STANDBY. |
|                    |               | Field ServiceEngineGroup.hm_on_standby cannot have
True as its value in BASIC license tier. Allowed value: False.
|
|                    |               | Field ServiceEngineGroup.app_cache_percent cannot have
10 as its value in BASIC license tier. Allowed value: 0.                                    |
|                    |
|
                                                        |
+--------------------+---------------
```

```
+-----------------------------------------------------------------------------------------------
-------------------------------------------------+
```

The following are the observations from the above output:

- The default Service Engine Group (Default-Group) has a default HA mode of N+M. The NSX Advanced Load Balancer Basic Edition only supports the legacy Active/Standby HA mode and this must be changed.
- The NSX Advanced Load Balancer Basic Edition does not support configurable cache memory, and this value must be edited to zero.
- The NSX Advanced Load Balancer Basic Edition does not support health monitoring from the standby SE, and this must be disabled.

The above command will display any features that must be disabled before proceeding, if the NSX Advanced Load Balancer Controller is used and other features are configured.

Setup Service Engine Group to Legacy Active/ Standby

Before configuring the NSX Advanced Load Balancer Controller to be in the Basic Edition tier, the Service Engine group needs to be re-configured to adhere to the Basic Edition restrictions. Log in to the NSX Advanced Load Balancer UI using the Controller IP address on your browser.

Navigate to **Infrastructure > Cloud Resources > Service Engine Group**, edit the **Default-Group** and perform the following steps:

- Under **Placement**, select the **Active/Standby** option from **High Availability Mode**.
- Deselect the **Enable Health Monitoring on Standby SE(s)** check box.
- Under **Resources**, set `0` as the value for **Memory for Caching**.
- Click **Save**.

Run the configuration audit API to validate that there are no configuration errors.

```
[admin:ctrl]: > show configuration audit tier basic
License audit passed successfully
[admin:ctrl]: >
```

Perform Configuration Audit

The NSX Advanced Load Balancer Controller enforces feature restrictions for each edition. Therefore, configuration needs to adhere to Basic edition for the transition from Enterprise edition to Basic edition to be allowed.

The following is the sample output for the `configuration audit` API. The following output exhibits the configuration and features which are not supported in the Basic Edition. Perform the required actions as per the output, before proceeding with the license edition change.

```
[admin:ctrl]: > show configuration audit tier basic
+--------------------+---------------
+-------------------------------------------------------------------------------------------------
-------------------------------------------------------+
| Object Type        | Name          | License
Violations
                                           |
+--------------------+---------------
+-------------------------------------------------------------------------------------------------
-------------------------------------------------------+
| VirtualService     | vs-1          | Object refers to system default object marked for
removal: StringGroup System-Rewritable-Content-
Types                                              |
|                    |               | Field
VirtualService.analytics_policy.full_client_logs.enabled cannot have True as its value in
BASIC license tier. Allowed value: False.        |
|                    |               | Field VirtualService.enable_autogw cannot have True as
its value in BASIC license tier. Allowed value: False.                                    |
|                    |               | Field VirtualService.content_rewrite cannot be set in
BASIC license tier.                                                                        |
|                    |               | Field VirtualService.sideband_profile cannot be set in
BASIC license tier.                                                                       |
|                    |
|
                                                        |
| ServiceEngineGroup | Default-Group | Field ServiceEngineGroup.ha_mode cannot have
HA_MODE_SHARED as its value in BASIC license tier. Allowed value(s):
HA_MODE_LEGACY_ACTIVE_STANDBY. |
|                    |               | Field ServiceEngineGroup.hm_on_standby cannot have
True as its value in BASIC license tier. Allowed value: False.
|
|                    |               | Field ServiceEngineGroup.app_cache_percent cannot have
10 as its value in BASIC license tier. Allowed value: 0.                                  |
|                    |
|
                                                        |
| Pool               | vs-1-pool     | Field Pool.connection_ramp_duration cannot have 10 as
its value in BASIC license tier. Allowed value: 0.                                        |
|                    |
|
                                                        |
+--------------------+---------------
+-------------------------------------------------------------------------------------------------
-------------------------------------------------------+
```

The following are the additional points to check while changing the default licensing tier from the Enterprise Edition to the Basic Edition.

- Save the backup configuration before changing the licensing edition.

- The changing of license editions is supported only for NSX Advanced Load Balancer Controllers and SEs on the same base version. The Controller and SEs on different patch versions are also supported for the license tier change option.

Prerequisites

A config audit tool is introduced to check the configurations that are not supported for the target edition. Run the configuration audit API to identify the Enterprise Edition features that need to be removed and resolve them before changing the edition from Enterprise to Basic.

The configuration audit API performs licensing checks on all the objects in the system and reports various violations. It ignores the violations in system default objects because they are handled internally.

Use the NSX Advanced Load Balancer CLI or the REST API to check the unsupported configuration and take the required actions:

- CLI: `show configuration audit tier <tier_name>`
- REST API: `/api/config-audit/tier/ <tier_name>`

Converting from Basic Edition to Enterprise Edition

There is no disruption of traffic or downtime while changing the licensing tier from the Basic Edition to the Enterprise Edition. Once the Controller is in the Enterprise edition, all the features available on the NSX Advanced Load Balancer can be utilized.

Procedure

1. Check the available Enterprise Service Units on the Controller. The available Enterprise Service Units must be sufficient to license the existing SEs. Changing from the Basic Edition to the Enterprise Edition is only allowed if sufficient licenses are available on the Controller.
2. Import and upload the NSX Advanced Load Balancer Enterprise license using UI if the Enterprise license is not available on the Controller.
3. Set the value of the `default_license_tier` to the **`enterprise`**.

```
[admin:ctrl]: > configure systemconfiguration
[admin:ctrl]: systemconfiguration > default_license_tier enterprise
overwriting the previous entered value for default_license_tier
[admin:ctrl]: systemconfiguration > save
```

4. Once the mode change is successful, the SEs are licensed with the Enterprise Service Units, and the Basic Service Units are freed up.

Results

The new system-default objects are created according to the Enterprise edition support. The configurations and features available on the Controller will be as per the Enterprise Edition feature enforcement. All the features on NSX Advanced Load Balancer is available when the NSX Advanced Load Balancer is in the Enterprise Edition.

FAQ

This topic answers basic questions about NSX Advanced Load Balancer-Basic edition.

- Would deploying x Basic LB SEs and y Enterprise LB SEs on the same NSX Advanced Load Balancer Controller be allowed?

  No, the Basic edition and the Enterprise edition licenses cannot exist on the same NSX Advanced Load Balancer Controller.

- Will NSX Advanced Load Balancer-Basic edition allow AKO for ingress?

  No. Enterprise edition license is required to use AKO.

- Is Per-App supported in the Basic Edition?

  No. Customers can create multiple SE Groups if this functionality is needed.

- Can a Controller be upgraded from Basic edition to Enterprise edition?

  Yes, the Basic to Enterprise edition upgrade on the NSX Advanced Load Balancer Controller is allowed. The only requirement is to have sufficient Enterprise edition licenses to license the existing SEs in the cluster. Upgrade to the Enterprise edition would be seamless and cause zero traffic disruption.

Note The Basic Edition of Avi Load Balancer (formerly known as NSX Advanced Load Balancer) has reached its End of Availability (EoA) date on Jan 30th, 2024. The changes only apply to the Basic Edition of Avi Load Balancer. For additional details, please visit VMware Knowledge Base

NSX Advanced Load Balancer Essentials for Tanzu

This topic explains how the NSX Advanced Load Balancer Essentials for Tanzu is deployed.

Note

- NSX Advanced Load Balancer Essentials for Tanzu is not an SKU and is not saleable.
- NSX Advanced Load Balancer Essentials for Tanzu does not require any licenses to be used.
- Content Library is not supported with NSX Advanced Load Balancer Essentials for Tanzu.

Licensing Requirement

Any VMware NSX Tanzu customer with active entitlements for Basic or Standard SKUs is eligible to use VMware NSX Advanced Load Balancer Essentials for Tanzu. When the NSX Advanced Load Balancer Controller is in the Essentials Edition, any number of SEs can be used without requirements for any license, but it requires an active support entitlement for VMware Tanzu Basic or Standard.

Converting from Enterprise Edition to Essentials Edition

Follow the steps mentioned in the following sections to change an NSX Advanced Load Balancer Controller from the Enterprise Edition to the essentials Edition.

Configuration Objects Restricted in the NSX Advanced Load Balancer Essentials UI

In VMware NSX Advanced Load Balancer essentials for Tanzu, a few configuration options are restricted or not editable in the NSX Advanced Load Balancer UI. For more information on permissions available for various UI objects in VMware NSX Advanced Load Balancer essentials for Tanzu, see Configuration Objects Restricted in the NSX Advanced Load Balancer UI for Essentials and Basic License.

Check Licenses and Import NSX Serial Key

Setting up an NSX Advanced Load Balancer Controller in the Essentials Edition does not require any licenses.

Perform Configuration Audit

NSX Advanced Load Balancer Controller enforces feature restrictions for each edition. Therefore, the configuration needs to adhere to Essentials Edition for allowing the transition from Enterprise Edition to Essentials Edition.

The following is the sample output for the configuration audit command when the target tier is set as `essentials`. The following output exhibits the configurations and features which are not supported in the Essentials Edition. Take the required actions as per the output of the configuration output command and then change the license tier edition.

```
   [admin:ctrl]: > show configuration audit tier essentials
   +--------------------+---------------
+-------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
   | Object Type        | Name          | License
Violations
                                            |
   +--------------------+---------------
+-------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
   | VirtualService     | vs-1          | Object refers to system default object marked for
removal: StringGroup System-Rewritable-Content-
Types                                              |
   |                    |               | Field
VirtualService.analytics_policy.full_client_logs.enabled cannot have True as its value in
ESSENTIALS license tier. Allowed value: False.        |
   |                    |               | Field VirtualService.enable_autogw cannot have True
as its value in ESSENTIALS license tier. Allowed value:
False.                                      |
   |                    |               | Field VirtualService.content_rewrite cannot be set
in ESSENTIALS license
tier.                                                                          |
   |                    |               | Field VirtualService.sideband_profile cannot be set
in ESSENTIALS license
tier.                                                                        |
   |                    |
|
```

```
                                                                |
   | ServiceEngineGroup | Default-Group | Field ServiceEngineGroup.ha_mode cannot have
HA_MODE_SHARED as its value in ESSENTIALS license tier. Allowed value(s):
HA_MODE_LEGACY_ACTIVE_STANDBY. |
   |                    |               | Field ServiceEngineGroup.hm_on_standby cannot have
True as its value in ESSENTIALS license tier. Allowed value:
False.                                |
   |                    |               | Field ServiceEngineGroup.app_cache_percent cannot
have 10 as its value in ESSENTIALS license tier. Allowed value:
0.                                  |
   |                    |
|
                                                                |
   | Pool               | vs-1-pool     | Field Pool.connection_ramp_duration cannot have 10
as its value in ESSENTIALS license tier. Allowed value:
0.                                        |
   |                    |
|
                                                                |
   +--------------------+---------------
+-------------------------------------------------------------------------------------------------
--------------------------------------------------+
```

The following are the additional points to check while changing the licensing tier from the Enterprise edition to the Essentials edition:

- Save the backup configuration before changing the licensing edition.
- Ensure that the base versions of NSX Advanced Load Balancer Controllers and NSX Advanced Load Balancer SEs are same.

  Note Controllers and SEs can run on different patch versions.

Prerequisites

A config audit tool is introduced to check the configurations that are not supported for the target edition. Run the config audit tool to check the violations in the configurations. Resolve all the violations before proceeding to change from the Enterprise Edition to the Essentials edition. The configuration audit API performs licensing checks on all the objects in the system and reports various violations. It ignores the violations in system default objects as they are handled internally.

Use the following NSX Advanced Load Balancer CLI or the REST API to check the unsupported configuration and take the required actions.

- NSX Advanced Load Balancer CLI: `show configuration audit tier <tier_name>`
- NSX Advanced Load Balancer REST API: `/api/config-audit/tier/ <tier_name>`

Changing Licensing Tier from the Enterprise Edition to the Essentials Edition

Once all the features and configurations specific to the Enterprise edition are removed, the NSX Advanced Load Balancer can be switched to the desired licensing tier.

You can change the licensing tier from the Enterprise edition to the Essential edition as follows.

1 Navigate to **Administration > Licensing**.

2 Click gear icon next to **Licensing**.

3 Select **Essentials License**.

4 Click **Save**.

After changing the licensing tier to the essentials edition, SEs do not require any licenses. The Enterprise Service Units are freed up and are no more used when NSX Advanced Load Balancer Controller runs in the essentials edition. The Controller will start enforcing essentials edition feature restrictions.

The new **System-Default** objects are created according to the NSX Advanced Load Balancer Essentials for Tanzu edition support.

The events related to the system objects and the configuration changes are available under **Operations > Events**. Various events related to the deletion of network profiles, WAF Profiles, and so on are listed here.

Converting from Essentials Edition to Enterprise Edition

There is no disruption of traffic or downtime while changing the licensing tier from the essentials edition to the Enterprise edition. Once the NSX Advanced Load Balancer Controller is in the Enterprise edition, all the features available on the NSX Advanced Load Balancer can be utilized.

Follow the steps mentioned below to set up the Controller in the Enterprise edition.

Procedure

1 Check the available Enterprise Service Units on the Controller. The available Enterprise Service Units must be sufficient to license the existing SEs. Changing from the essentials edition to the Enterprise edition is only allowed if sufficient licenses are available on the Controller.

2 Import and upload the NSX Advanced Load Balancer Enterprise license using UI if the Enterprise license is not available on the Controller.

3 Change the system mode to Enterprise mode.

```
[admin:ctrl]: > configure systemconfiguration
[admin:ctrl]: systemconfiguration > default_license_tier enterprise
overwriting the previous entered value for default_license_tier
[admin:ctrl]: systemconfiguration > save
```

4 Once the mode change is successful, the SEs are entitled to the Enterprise Service Units.

5 The new **system-default** objects are created according to the Enterprise edition support. The configurations and features available on the Controller will be as per the Enterprise Edition feature enforcement. All the features on NSX Advanced Load Balancer are available when the NSX Advanced Load Balancer is in the Enterprise edition.

How to Get Support

A customer that has an active entitlement for VMware Tanzu Basic, or VMware Tanzu Standard is eligible for NSX ALB essentials for Tanzu support. This can be supported by the NSX Advanced Load Balancer and VMware Technical Support teams.

FAQs

- Is NSX Advanced Load Balancer Essentials an SKU?

  No, NSX Advanced Load Balancer Essentials do not require any licenses. Support entitlement is through VMware Tanzu SKUs (Basic or Standard).

- Can a Controller be upgraded from the Essentials Edition to the Enterprise Edition?

  Yes, the Essentials to Enterprise Edition upgrade is allowed on the Controller. The only requirement is to have sufficient Enterprise Edition licenses to license the existing SEs in the Controller Cluster. Upgrade to Enterprise Edition can be seamless and cause zero traffic disruption.

Terms of NSX Advanced Load Balancer Software License

NSX Advanced Load Balancer licenses are based on annual or multi-year subscription. The licenses are valid for a year and can be renewed on a yearly basis.

Note For complete information on procuring, installing, and managing a license, see Chapter 3 Licensing.

NSX Advanced Load Balancer License Types

License Type	Infrastructure	Description
Data Plane (NSX Advanced Load Balancer Service Engine)		
vCPU	VMs, Baremetal, Container, Cisco CSP, Private, and public cloud	Defines the maximum number of vCPUs (cores) across all SEs. This is the default license mode. The SE vCPU allocation is configured in the SE group. ■ For example, a 10-vCPU license allows up to ten 1-vCPU SEs, or five 2-vCPU SEs, or a mix of four 1-vCPU SEs and three 2-vCPU SEs (across two SE groups), and so on. Per-VS license mode: If an SE group is configured in per-app SE mode, a vCPU counts at a 25% rate for licensing usage. For example, a 2-vCPU SE in the per-app SE group utilizes a 0.5 vCPU license (2 * 0.25). Per-app SE mode is limited to a maximum of two virtual services per SE. This mode is suitable for deployments with dedicated per-application load balancers. Per-VS license mode is not supported for DNS virtual services.
Socket	Baremetal, Cisco CSP	Defines the total number of sockets that an SE can be deployed on. This license is applicable for processors with up to 14 cores.

License Type	Infrastructure	Description
Data Plane (NSX Advanced Load Balancer Service Engine)		
High-core socket	Baremetal, Cisco CSP	Defines the total number of sockets that an SE can be deployed on. This license is applicable for processors with 16 cores or more.
Bandwidth	VMs, Baremetal, Container, Cisco CSP, Private, and public cloud	Defines bandwidth license applicable to a single SE. Available in the options of 25Mbps, 200Mbps, and 1Gbps (for public cloud only). Multiple Bandwidth licenses cannot be aggregated on a single SE.
Container-Core	Container East-West	Defines the maximum number of vCPUs (cores) across all SEs deployed for East-West traffic in a Container Cluster. This license is used for East-West traffic in a service mesh.
Control Plane (NSX Advanced Load Balancer Controller)		
SaaS unit	SaaS	Defines the number of NSX Advanced Load Balancer SaaS units needed in the SaaS instance to support the required number of SEs and application instances.

Note A free trial license with all features supported by an enterprise license is available for evaluation. Contact the NSX Advanced Load Balancer representative for more details.

License Overage Allowance

The NSX Advanced Load Balancer allows an overage allowance for the vCPU and bandwidth licenses.

The main features of the overage allowance are as follows:

- NSX Advanced Load Balancer provides an allowance of 10% of the configured license capacity. This allowance can be utilized once the regular capacity of the license is exhausted.
- If the 10% allowance results in a fractional number, it is rounded up to the next higher integer.
- In the case of bandwidth licenses, the allowance is for each bandwidth license SKU separately (25M, 200M, 1G BW).
- For an NSX Advanced Load Balancer Controller having 15 vCPU licenses, the overage allowance is 1.5 vCPU (10% of 15vCPU). This value will be rounded up to two vCPUs. The effective license value when the overage allowance is active is 17 vCPUs.
- For a Controller having 15 25M bandwidth licenses and ten 200M bandwidth licenses, the allowance is 1.5 additional 25M bandwidth licenses (rounded up to two bandwidth licenses) and one additional 200M bandwidth license.

Removing Trial Licenses from an NSX Advanced Load Balancer Controller

Licenses on an NSX Advanced Load Balancer Controller can be checked by navigating to **Administration > Licensing** on the UI. Trial licenses cannot be removed using UI or CLI. NSX Advanced Load Balancer REST API is used to remove trial license.

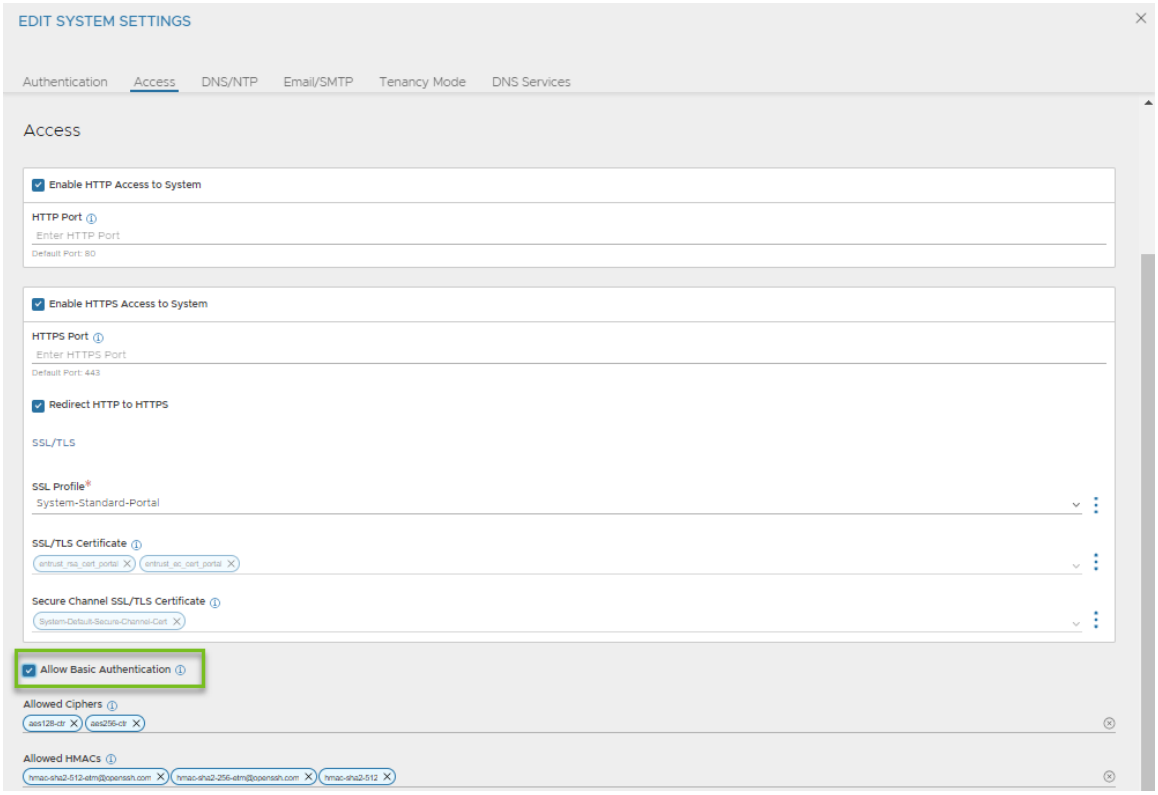
Follow the instructions mentioned below to remove trial licenses from an NSX Advanced Load Balancer Controller.

Instructions

1 Enable HTTP Basic authentication:

a Log into NSX Advanced Load Balancer UI, navigate to **Administration > System Settings** , and click **EDIT**.

b Under **Access**, select **Allow Basic Authentication** to activate HTTP basic authentication.



2 Use `cURL` command to delete the trial license

a Run the following `cURL` command from a host which can reach the NSX Advanced Load Balancer Controller:

```
curl -k --user admin -X DELETE --header 'Accept: application/json' --header
'X-Avi-Version: <Controller-Version>' 'https://<controller-address>/api/license/Trial'
```

b Example for the cURL command is shown below:

```
curl -k --user admin -X DELETE --header 'Accept: application/json' --header
'X-Avi-Version: 21.1.1' 'https://1.2.3.4/api/license/Trial'
```

c Above command will prompt for the admin password. Enter the password for the admin account.

3 Verify the changes:

a To verify that the trial license is removed from the system successfully, navigate to **Administration > Licensing** in the NSX Advanced Load Balancer Controller UI and ensure that it is not listed.

Updating License on each Node in an NSX Advanced Load Balancer Controller Cluster

It is not required to update the license on each node in an NSX Advanced Load Balancer Controller Cluster. The leader node in a Controller Cluster pushes its license to the remaining (follower) nodes. The leader node license in the NSX Advanced Load Balancer Controller Cluster can be updated using the NSX Advanced Load Balancer UI.

Updating an existing license follows the same process as adding a new license. When a license file has the same license ID (with an updated limit and expiration date), uploading this license file will update the existing entry in place.

Navigate to **Administration > Licensing** to update the existing license on a leader node in a Controller Cluster.

For more information about License management on NSX Advanced Load Balancer, see NSX Advanced Load Balancer - Enterprise Edition.

Controller Settings

4

The **System Settings** page of the NSX Advanced Load Balancer Controller UI provides the ability to configure administrative settings for the NSX Advanced Load Balancer.

The following settings can be configured:

- Authentication/Authorization: Choose local or remote authentication and authorization of NSX Advanced Load Balancer users. In the remote case, enables specification of an auth profile. For more information, see Auth Profile.
- Access Settings: Select options for securing management access to the NSX Advanced Load Balancer system.
- DNS / NTP: Specify the Domain Name System (DNS) and Network Time Protocol (NTP) servers used by NSX Advanced Load Balancer.
- Licensing: Display license information and upload license files.
- Email/SMTP: Specify the source for Simple Mail Transfer Protocol (SMTP) traffic from NSX Advanced Load Balancer.
- Tenant Settings: Specify system-wide settings for multitenancy.
- SSH Key Settings: Create or import SSH keys.
- Upload HSM Packages: Upload a customer's Hardware Security Module (HSM) software package, to enable HSM features within NSX Advanced Load Balancer.

For more information, see Application Profile topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Read the following topics next:

- NTP/DNS Settings
- Email-SMTP Settings

NTP/DNS Settings

NTP (Network Time Protocol) settings are critical for the functioning of the NSX Advanced Load Balancer. The analytics functionality in the Controller relies on synchronization between

the Controller in the cluster and SE. Controller synchronize time with the configured NTP servers and the SE in turn synchronize time with the Controller.

NSX Advanced Load Balancer requires access to valid DNS and NTP servers for operation. Using an NTP server is especially critical, without which NSX Advanced Load Balancer cannot function properly.

Port 123 must be open on the NSX Advanced Load Balancer Controller to receive timestamps over UDP.

Configure DNS/ NTP Settings using the UI

Configure NTP servers from the NSX Advanced Load Balancer UI as follows:

1 Navigate to **Administration > System Settings**.

2 Click the **EDIT** button to view the **EDIT SYSTEM SETTINGS** screen.

3 Under **DNS/NTP** tab, enter **DNS Resolver(s)**. This is a comma-delimited list of DNS server IP addresses. If a DNS server is not configured, NSX Advanced Load Balancer will not be able to accept names for load-balanced servers, virtual services, mail servers, and similar inputs.

4 Enter the **Search Domain** to use in DNS lookup.

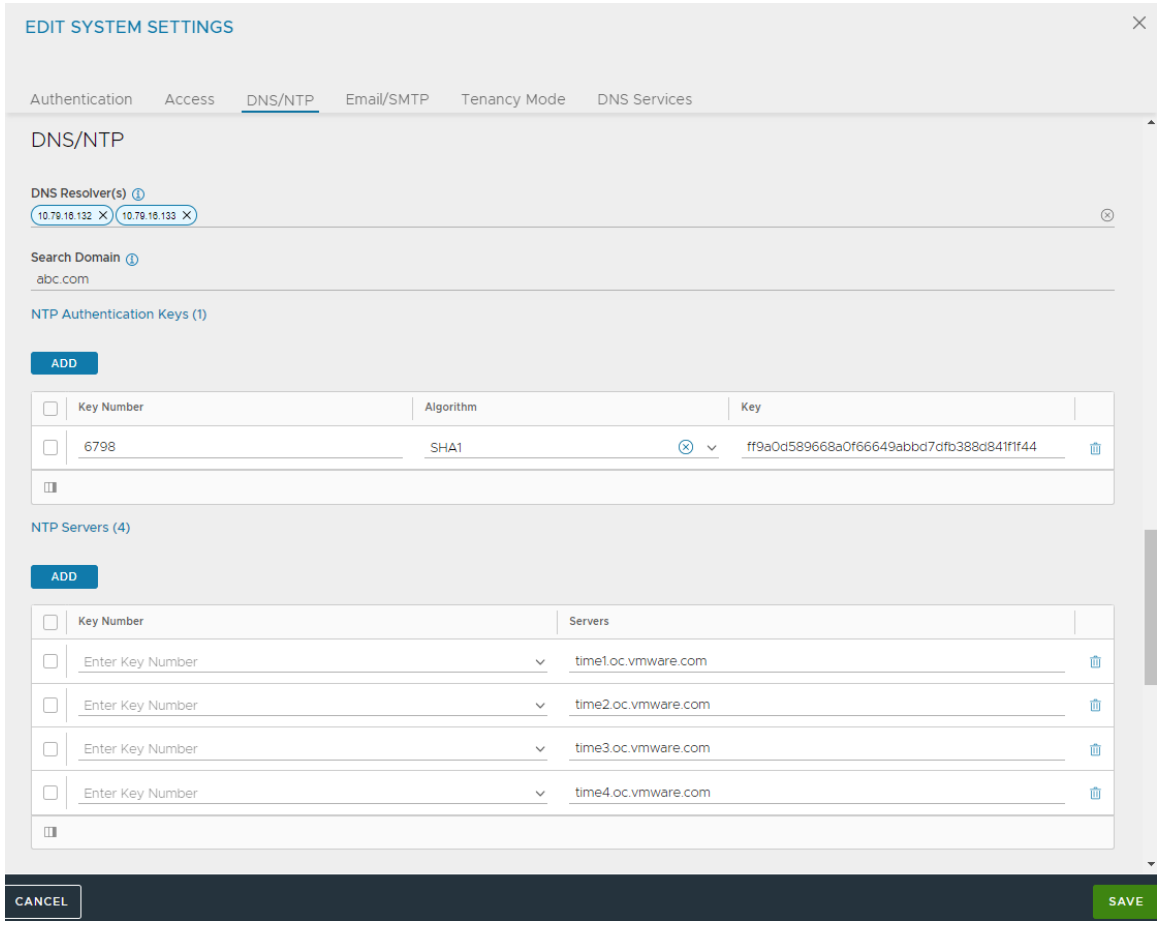
5 Configure **NTP Authentication Keys**.

   a Click **Add**.

   b Enter the **Key Number** from the list of trusted keys used to authenticate this server.

   c Select the Message Digest **Algorithm** used for NTP authentication. Message Digest (MD5) and Secure Hash Algorithm (SHA1) are selected.

   d Enter the NTP Authentication **Key**.



6 Under **NTP Servers**, select the **Key Number** for NTP authentication and the IP address of the NTP **Servers**.

7 Click **SAVE**.

Configure DNS/ NTP Settings using the CLI

Configure NTP servers from the CLI as follows.

```
: > configure systemconfiguration
: systemconfiguration> ntp_configuration
: systemconfiguration:ntp_configuration> ntp_server_list 23.239.26.89 ntp_server_list
69.89.207.99
: systemconfiguration:ntp_configuration> exit
: systemconfiguration> exit
+-----------------------------------+-----------------------------------+
| Field                             | Value                             |
+-----------------------------------+-----------------------------------+
| uuid                              | default                           |
| dns_configuration                 |                                   |
| search_domain                     |                                   |
| ntp_configuration                 |                                   |
| ntp_server_list[1]                | 23.239.26.89                      |
| ntp_server_list[2]                | 69.89.207.99                      |
```

```
| tech_support_uploader_configuration |                                    |
| auto_upload                         | False                              |
| portal_configuration                |                                    |
| enable_https                        | True                               |
| redirect_to_https                   | True                               |
| enable_http                         | True                               |
| sslkeyandcertificate_refs[1]        | System-Default-Portal-Cert         |
| sslkeyandcertificate_refs[2]        | System-Default-Portal-Cert-EC256   |
| use_uuid_from_input                 | False                              |
| sslprofile_ref                      | System-Standard                    |
| enable_clickjacking_protection      | True                               |
| allow_basic_authentication          | True                               |
| password_strength_check             | False                              |
| disable_remote_cli_shell            | False                              |
| global_tenant_config                |                                    |
| tenant_vrf                          | False                              |
| se_in_provider_context              | True                               |
| tenant_access_to_provider_se        | True                               |
| email_configuration                 |                                    |
| smtp_type                           | SMTP_LOCAL_HOST                    |
| from_email                          | admin@avicontroller.net            |
| mail_server_name                    | localhost                          |
| mail_server_port                    | 25                                 |
| docker_mode                         | False                              |
+-------------------------------------+------------------------------------+
```

The DNS Search Domain is the local domain name, which will be appended to a name that is not fully qualified. For instance, if the DNS search domain is set to avinetworks.com, and the name to be resolved is www, NSX Advanced Load Balancer will look up www.avinetworks.com.

Configure DNS Settings using the CLI

The `.local` domains are not resolvable by default through the configured DNS server (local domains are not routed to DNS servers). The search domains need to be configured explicitly for `.local` domains to make lookups work within this DNS domain. Configure the DNS settings from the CLI as shown below.

```
  [admin:avictrl]: > configure systemconfiguration
  [admin:avictrl]: systemconfiguration> dns_configuration
  [admin:avictrl]: systemconfiguration:dns_configuration> search_domain "test.domain1.local
test.domain2.com"
  Overwriting the previously entered value for search_domain
  [admin:avictrl]: systemconfiguration:dns_configuration> save
  [admin:avictrl]: systemconfiguration> save
  +---------------------------------+-------------------------------------+
  | Field                           | Value                               |
  +---------------------------------+-------------------------------------+
  | uuid                            | default                             |
  | dns_configuration               |                                     |
  |   server_list[1]                | 10.79.16.132                        |
  |   search_domain                 | test.domain1.local test.domain2.com|
  | ntp_configuration               |                                     |
  |   ntp_servers[1]                |                                     |
```

```
|     server                        | 0.us.pool.ntp.org                  |
|   ntp_servers[2]                  |                                    |
|     server                        | 1.us.pool.ntp.org                  |
|   ntp_servers[3]                  |                                    |
|     server                        | 2.us.pool.ntp.org                  |
|   ntp_servers[4]                  |                                    |
|     server                        | 3.us.pool.ntp.org                  |
| portal_configuration              |                                    |
|   enable_https                    | True                               |
|   redirect_to_https               | True                               |
|   enable_http                     | True                               |
|   sslkeyandcertificate_refs[1]    | System-Default-Portal-Cert         |
|   sslkeyandcertificate_refs[2]    | System-Default-Portal-Cert-EC256   |
|   use_uuid_from_input             | False                              |
|   sslprofile_ref                  | System-Standard-Portal             |
|   enable_clickjacking_protection  | True                               |
|   allow_basic_authentication      | True                               |
|   password_strength_check         | False                              |
|   disable_remote_cli_shell        | False                              |
|   disable_swagger                 | False                              |
|   api_force_timeout               | 24 hours                           |
|   minimum_password_length         | 8                                  |
| global_tenant_config              |                                    |
|   tenant_vrf                      | False                              |
|   se_in_provider_context          | False                              |
|   tenant_access_to_provider_se    | True                               |
| email_configuration               |                                    |
|   smtp_type                       | SMTP_LOCAL_HOST                    |
|   from_email                      | admin@avicontroller.net            |
|   mail_server_name                | localhost                          |
|   mail_server_port                | 25                                 |
|   disable_tls                     | False                              |
| docker_mode                       | False                              |
| ssh_ciphers[1]                    | aes128-ctr                         |
| ssh_ciphers[2]                    | aes256-ctr                         |
| ssh_hmacs[1]                      | hmac-sha2-512-etm@openssh.com      |
| ssh_hmacs[2]                      | hmac-sha2-256-etm@openssh.com      |
| ssh_hmacs[3]                      | hmac-sha2-512                      |
| default_license_tier              | ENTERPRISE                         |
| secure_channel_configuration      |                                    |
|   sslkeyandcertificate_refs[1]    | System-Default-Secure-Channel-Cert |
| welcome_workflow_complete         | False                              |
| fips_mode                         | False                              |
| enable_cors                       | False                              |
| common_criteria_mode              | False                              |
+-----------------------------------+------------------------------------+
```

Configure DNS Settings using the API

Configure NTP servers with the API as follows.

```
{
    },
    "ntp_configuration": {
```

```
        "ntp_server_list": [
            {
                "type": "V4",
                "addr": "23.239.26.89"
            },
            {
                "type": "V4",
                "addr": "69.89.207.99"
            }
        ]
    }
}
```

Configuring NTP Authentication using the CLI

NTP authentication can be enabled using either the CLI or the REST API. With NTP authentication, one can specify a set of trusted authentication keys and configure each NTP server peer with a specific authentication key. The NTP authentication key object consists of a key number, key algorithm (SHA1 or MD5) and the key itself.

Configure NTP and NTP authentication with the CLI as follows.

```
[admin:10-10-25-45]: > configure systemconfiguration
[admin:10-10-25-45]: systemconfiguration> ntp_configuration
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> ntp_authentication_keys
key_number 1 algorithm ntp_auth_algorithm_md5 key "=I&FBDl,WM,en5Mn~DaG"
New object being created
[admin:10-10-25-45]: systemconfiguration:ntp_configuration:ntp_authentication_keys> exit
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> ntp_authentication_keys
key_number 5 algorithm ntp_auth_algorithm_sha1 key ff9a0d589668a0f66649abbd7dfb388d841f1f44
New object being created
[admin:10-10-25-45]: systemconfiguration:ntp_configuration:ntp_authentication_keys> exit
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> exit
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> ntp_servers server 23.239.26.89
New object being created
[admin:10-10-25-45]: systemconfiguration:ntp_configuration:ntp_servers> exit
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> ntp_servers server 69.89.207.99
key_number 5
New object being created
[admin:10-10-25-45]: systemconfiguration:ntp_configuration:ntp_servers> exit
[admin:10-10-25-45]: systemconfiguration:ntp_configuration> exit
[admin:10-10-25-45]: systemconfiguration> exit
+-------------------------------------+-----------------------------------------+
| Field                               | Value                                   |
+-------------------------------------+-----------------------------------------+
| uuid                                | default                                 |
| dns_configuration                   |                                         |
| search_domain                       |                                         |
| ntp_configuration                   |                                         |
| ntp_authentication_keys[1]          |                                         |
| key_number                          | 1                                       |
| algorithm                           | NTP_AUTH_ALGORITHM_MD5                  |
| key                                 | =I&FBDl,WM,en5Mn~DaG                    |
| ntp_authentication_keys[2]          |                                         |
```

```
| key_number                          | 5                                        |
| algorithm                           | NTP_AUTH_ALGORITHM_SHA1                  |
| key                                 | ff9a0d589668a0f66649abbd7dfb388d841f1f44 |
| ntp_servers[1]                      |                                          |
| server                              | 23.239.26.89                             |
| ntp_servers[2]                      |                                          |
| server                              | 69.89.207.99                             |
| key_number                          | 5                                        |
| tech_support_uploader_configuration |                                          |
| auto_upload                         | False                                    |
| portal_configuration                |                                          |
| enable_https                        | True                                     |
| redirect_to_https                   | True                                     |
| enable_http                         | True                                     |
| sslkeyandcertificate_refs[1]        | System-Default-Portal-Cert               |
| sslkeyandcertificate_refs[2]        | System-Default-Portal-Cert-EC256         |
| use_uuid_from_input                 | False                                    |
| sslprofile_ref                      | System-Standard                          |
| enable_clickjacking_protection      | True                                     |
| allow_basic_authentication          | True                                     |
| password_strength_check             | False                                    |
| disable_remote_cli_shell            | False                                    |
| global_tenant_config                |                                          |
| tenant_vrf                          | False                                    |
| se_in_provider_context              | True                                     |
| tenant_access_to_provider_se        | True                                     |
| email_configuration                 |                                          |
| smtp_type                           | SMTP_LOCAL_HOST                          |
| from_email                          | admin@avicontroller.net                  |
| mail_server_name                    | localhost                                |
| mail_server_port                    | 25                                       |
| docker_mode                         | False                                    |
+-------------------------------------+------------------------------------------+
```

Configuring NTP Authentication using the API

Configure NTP and NTP authentication with the API as follows.

```
{
    },
    "ntp_configuration": {
        "ntp_servers": [
            {
                "server": {
                    "type": "V4",
                    "addr": "23.239.26.89"
                }
            },
            {
                "key_number": 5,
                "server": {
                    "type": "V4",
                    "addr": "69.89.207.99"
                }
```

```
            }
        ],
        "ntp_authentication_keys": [
            {
                "key_number": 1,
                "algorithm": "NTP_AUTH_ALGORITHM_MD5",
                "key": "=I&FBDl,WM,en5Mn~DaG"
            },
            {
                "key_number": 5,
                "algorithm": "NTP_AUTH_ALGORITHM_SHA1",
                "key": "ff9a0d589668a0f66649abbd7dfb388d841f1f44"
            }
        ]
    }
}
```

Time synchronization for the Service Engine

NSX Advanced Load Balancer supports configuring NTP servers on SEs. Prior to this, NTP synchronization for SEs relied on the Controller as the NTP server and performed time synchronization on UDP port 123 over the management interface. The SE presumes network connectivity with the Controller.

NTP Configuration for SE deployed as a Virtual Machine

The SE NTP servers can be configured for the virtual machine and LSC-based deployments. The SE will synchronize time with the configured servers at start-up and periodically monitor the time sync status.

NTP Configuration for Virtual Machine Deployments

When SE is deployed as a virtual machine, NTP servers can be configured using any of the methods below and applies configuration in the following order of priority:

DHCP

If DHCP through dhclient provides NTP servers over the management interface, the SE uses DHCP provided NTP servers as configuration for SE NTP synchronization.

Cloud Configuration

If DHCP does not provide NTP servers, NTP servers are acquired from the cloud configuration. NTP server's configuration through cloud configuration is a bootup property, and the SE must be restarted to apply this configuration.

Controller

NTP servers configuration is through system configuration

NTP Server configuration through Cloud configuration using CLI is as follows:

```
[admin:ctrl]: > configure cloud Default-Cloud
Updating an existing object. Currently, the object is:
+------------------------------+------------------------------------------+
| Field                        | Value                                    |
+------------------------------+------------------------------------------+
| uuid                         | cloud-666c8a8f-341d-4225-a189-c128981130c7 |
| name                         | Default-Cloud                            |
| vtype                        | CLOUD_NONE                               |
| dhcp_enabled                 | False                                    |
| mtu                          | 1500 bytes                               |
| prefer_static_routes         | False                                    |
| enable_vip_static_routes     | False                                    |
| license_type                 | LIC_CORES                                |
| state_based_dns_registration | True                                     |
| ip6_autocfg_enabled          | False                                    |
| dns_resolution_on_se         | False                                    |
| enable_vip_on_all_interfaces | False                                    |
| maintenance_mode             | False                                    |
| tenant_ref                   | admin                                    |
| license_tier                 | ENTERPRISE                               |
| autoscale_polling_interval   | 60 seconds                               |
| vmc_deployment               | False                                    |
| metrics_polling_interval     | 300 seconds                              |
+------------------------------+------------------------------------------+
[admin:ctrl]: cloud> ntp_configuration
[admin:ctrl]: cloud:ntp_configuration>
[admin:ctrl]: cloud:ntp_configuration> ntp_servers  index 1 server 23.239.26.89
New object being created
[admin:ctrl]: cloud:ntp_configuration:ntp_servers> save
[admin:ctrl]: cloud:ntp_configuration> save
[admin:ctrl]: cloud> save
+------------------------------+------------------------------------------+
| Field                        | Value                                    |
+------------------------------+------------------------------------------+
| uuid                         | cloud-666c8a8f-341d-4225-a189-c128981130c7 |
| name                         | Default-Cloud                            |
| vtype                        | CLOUD_NONE                               |
| dhcp_enabled                 | False                                    |
| mtu                          | 1500 bytes                               |
| prefer_static_routes         | False                                    |
| enable_vip_static_routes     | False                                    |
| license_type                 | LIC_CORES                                |
| state_based_dns_registration | True                                     |
| ip6_autocfg_enabled          | False                                    |
| dns_resolution_on_se         | False                                    |
| enable_vip_on_all_interfaces | False                                    |
| maintenance_mode             | False                                    |
| tenant_ref                   | admin                                    |
| license_tier                 | ENTERPRISE                               |
| autoscale_polling_interval   | 60 seconds                               |
| vmc_deployment               | False                                    |
| metrics_polling_interval     | 300 seconds                              |
| ntp_configuration            |                                          |
```

```
|   ntp_servers[1]             |                                            |
|     server                   | 23.239.26.89                               |
+------------------------------+--------------------------------------------+
[admin:ctrl]: >
```

NTP Configuration for LSC (Baremetal) Deployments

When SE is deployed as a container on the Bare Metal, the administrator is required to configure the NTP servers on the host.

LSC NTP synchronization only supports the following NTP daemons:

ntpd

The Network Time Protocol daemon (ntpd) is an operating system program that maintains the system time in synchronization with time servers using the NTP.

chronyd

chronyd is another implementation of the NTP and is used:

- To synchronize the system clock with NTP servers.
- To synchronize the system clock with a reference clock, for instance, a GPS receiver.
- To synchronize the system clock with a manual time input.

SE NTP operation

In both modes of deployments (SE deployed as a VM or as a container on LSC), SE periodically verifies if the NTP daemons can acquire and time sync with configured servers, and if SE is unable to sync time with the configured servers, an event is raised. The event is periodically repeated in 15 minutes unless the NTP time is synchronised.

Email-SMTP Settings

The NSX Advanced Load Balancer proactively sends emails for password reset operations, or as a result of a triggered alert action that calls for email notifications to be sent to administrators or monitoring systems. Emails are sent from the Controller, which requires the Controller to have DNS and network access to a destination email server. When the Controller sends an email, the email is sourced from the SMTP source.

Note The NSX Advanced Load Balancer supports non-TLS SMTP servers.

Modify Email or SMTP Settings

The steps to view or modify the email or SMTP settings are as follows.

1 Navigate to **Administration > System Settings**.

2 To edit the settings, click **EDIT**.

3 Under **Email/SMTP**, select **Anonymous SMTP**, **Local**, **SMTP**, or **None** as required for **SMTP Source**.

4 Configure the SMTP source settings.

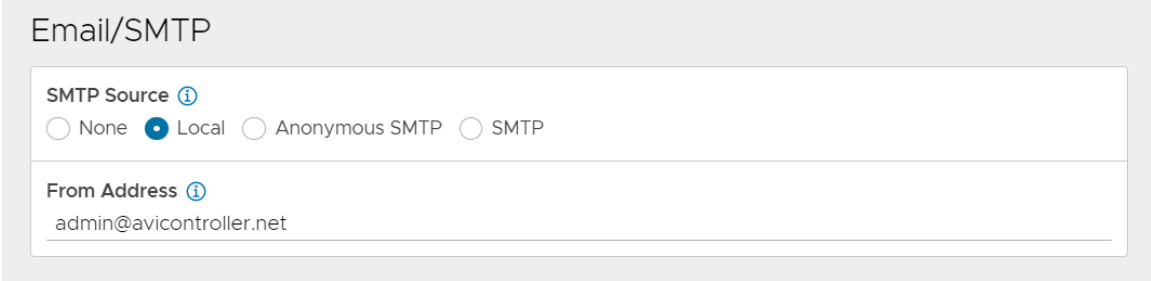
5 Click **SAVE**.

Configure Email or SMTP Source Settings using the UI

1 Select **None** from the **SMTP Source** option to not send emails.

2 Select **Local** from the **SMTP Source** option to send the email from a local host. Some enterprise email servers might not accept this method.

   a Enter **From Address**.



3 Select **SMTP** from the **SMTP Source** option to send the email from a remote SMTP server. This method is generally more trusted by the staff of security-conscious enterprise environments. Under **SMTP**, enter the following details:

   a Enter **Username** to authenticate to the mail server.

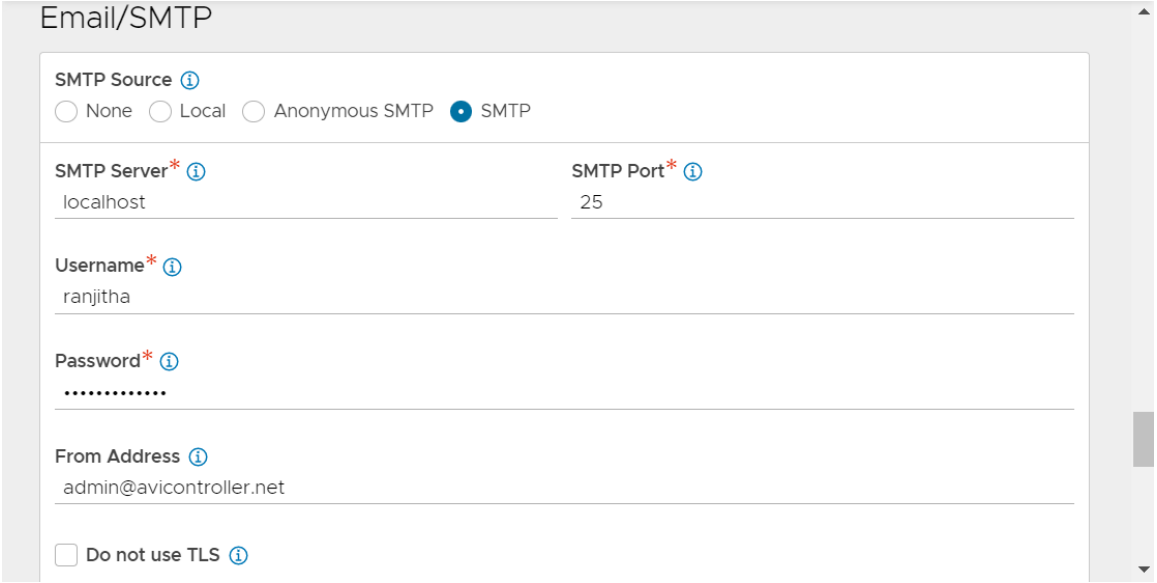
   b Enter **Password** to authenticate to the mail server.

   c Enter **SMTP Server** to send the email from a server host.

   d Enter **SMTP Port** number. The default service port for SMTP is 25.

   e Enter **From Address**.

   f Select **Do not use TLS** to disable the TLS connection to the mail server. By default, the option is not selected.



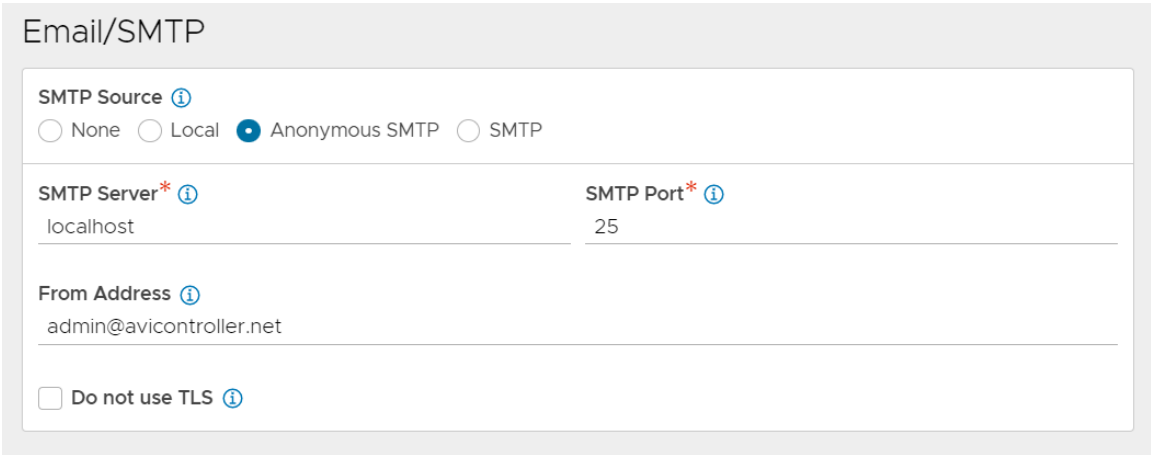
4 Select **Anonymous SMTP** from the **SMTP Source** option to log in to the SMTP server without the need for a username and password. The SMTP server must allow anonymous access. Under **Anonymous SMTP Server**, enter the following details:

   a Enter **From Address**.

   b Enter **SMTP Server** to send the email from a server host.

   c Enter **SMTP Port** number. The default service port for SMTP is 25.

   d Select **Do not use TLS** to disable the TLS connection to the mail server. By default, the option is deselected.



Configure Email or SMTP Source Settings using the CLI

Email/ SMTP settings are configured under System configuration as shown below.

```
[admin:ctrl]: systemconfiguration> email_configuration
[admin:ctrl]: systemconfiguration> email_configuration from_name "Admin Avinetworks"
[admin:ctrl]: systemconfiguration:email_configuration> save
```

```
[admin:ctrl]: systemconfiguration> save

| email_configuration              |                                  |
|   smtp_type                      | SMTP_LOCAL_HOST                  |
|   from_email                     | admin@avicontroller.net          |
|   mail_server_name               | localhost                        |
|   mail_server_port               | 25                               |
|   disable_tls                    | False                            |
|   from_name                      | Admin Avinetworks                |
```

Time zone for e-mail Alert Notifications

By default, the Controller uses UTC time when generating timestamps in the alert notification e-mails.

To configure the timestamps to reflect a specific timezone, use the following configuration through the CLI:

```
configure systemconfiguration
email_settings
email_timezone < timezone >
save
save
```

Using non-TLS SMTP Servers

To use non-TLS SMTP servers, the TLS option must be disabled implicitly.Log in to the CLI and run the following commands using the `configure systemconfiguration` mode.configure systemconfiguration email_configuration disable_tls save save

NSX Advanced Load Balancer Controller Cluster

5

To ensure complete system redundancy, the NSX Advanced Load Balancer Controller must be highly available. To provide high availability (HA) for the Controller, add two additional Controller nodes to create a three-node Controller cluster.

HA of the Controller requires three separate Controller instances configured as a 3-node cluster. Start with a single-node Controller deployment and use the following steps to add two additional Controller nodes to form a 3-node cluster.

Note If the cluster is already deployed and you want to modify its node membership or dismantle the cluster, see Changing NSX Advanced Load Balancer Controller Cluster Configuration.

Prerequisites for Cluster Deployment

A Controller cluster can have three nodes, one leader node, and two follower nodes.

Leader node

- The leader can be any single node with configuration or without configuration.
- The leader can have SEs connected.
- The node must have a static IP address.
  - Using DHCP can cause issues when nodes reboot and their IP addresses change.
  - The current release does not support the use of hostnames for cluster configuration.

Follower nodes

- An NSX Advanced Load Balancer Controller cluster can have 3 nodes, 1 leader node and 2 follower nodes.
- Follower nodes must have only factory-default configuration.
- In AWS environments, follower nodes must have an initial password configured. For more information, see *Changes for Cluster Set-up for AWS Deployments* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.
- In all other environments, follower nodes must be using the default initial admin credentials; do not run the initial setup wizard to set an admin password.

- Follower nodes are expected to be running the same NSX Advanced Load Balancer base+patch version as the leader.
- Follower typically is a VM or container created from the the NSX Advanced Load Balancer Controller installation package.
- Each follower node must have a static IP address.

Caveats

The current release does not support the use of hostnames for cluster configuration.

Read the following topics next:

- Deploying an NSX Advanced Load Balancer Controller Cluster
- Deploying an NSX Advanced Load Balancer Controller Cluster (Additional)
- Deploy SEs in Different Data Centers from Controllers
- Clustering Patched Controllers
- Controller Cluster IP
- Cluster Configuration with DNS Hostnames
- Changing NSX Advanced Load Balancer Controller Cluster Configuration
- Updating the Configuration Following NSX Advanced Load Balancer Controller IP Address Change
- Customizing NSX Advanced Load Balancer Configuration
- Clustering NSX Advanced Load Balancer Controllers of Different Networks
- High Availability for NSX Advanced Load Balancer Controllers
- Control Plane High Availability
- Switching Among Controllers Using the Controller Site Selector
- Controller Cluster Deployment Across Two Availability Zones
- Impact of a Controller Failure
- Recover a Non-Operational Controller Cluster
- API - Configuring the NSX Advanced Load Balancer Controller Cluster
- NSX Advanced Load Balancer ControllerCluster Configuration in AWS
- Configure Azure Cluster IP
- Changing Controller IP Addresses in a Bare-metal Environment
- OpenStack Network Configuration for NSX Advanced Load Balancer Controller Cluster
- FAQs on Controller Cluster
- FAQs on Database Replication

Deploying an NSX Advanced Load Balancer Controller Cluster

To deploy an NSX Advanced Load Balancer Controller cluster, first deploy a single NSX Advanced Load Balancer Controller node (the leader), and then add the follower nodes to the leader.

Ensure that after using the setup wizard to install NSX Advanced Load Balancer on the follower nodes, you do not make any other configuration changes on these nodes.

For more information about initial cluster deployment, see *Deploying a VMware NSX Advanced Load Balancer Controller Cluster* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.

Using the UI

The steps to deploy a cluster on the web interface of the NSX Advanced Load Balancer Controller that will serve as the leader are as follows:

Procedure

1. Navigate to **Administration > Controller > Nodes** and click **EDIT**.
2. Specify the Controller Cluster name and IP address.

   For more details on Controller cluster IP address, see Controller Cluster IP. This will be the shared management IP address of the cluster.
3. Specify the host IP addresses of each of the 3 nodes in the **Node IP** field of each **Cluster Node**.
   - Controller Node-1: Host IP address of the leader node.
   - Controller Node-2: Host IP address of one of the follower nodes.
   - Controller Node-3: Host IP address of one of the other follower nodes.
4. Click **SAVE**.

Using the CLI

Log in to the CLI (or CLI shell) and enter the commands shown in the following example.

Note Ensure that you specify the host IP addresses of the NSX Advanced Load Balancer Controller nodes instead of the IP addresses shown below:

Configure the cluster with the Controller nodes at [‘10.10.25.81’, ‘10.10.25.82’, ‘10.10.25.83’].

```
configure cluster
Updating an existing object. Currently, the object is:
+---------------+------------------------------------------------+
| Field         | Value                                          |
+---------------+------------------------------------------------+
```

```
| uuid          | cluster-eb01bf05-7313-4a4f-91b6-21e46d3c237d |
| name          | cluster-0-1                                  |
| nodes[1]      |                                              |
| name          | 10.10.25.81                                  |
| ip            | 10.10.25.81                                  |
| vm_ref        | EB01BF05-7313-4A4F-91B6-21E46D3C237D         |
| vm_mor        |                                              |
| vm_hostname   | node1.controller.local                       |
| tenant_ref    | admin                                        |
+---------------+----------------------------------------------+
: cluster> nodes name 10.10.25.82 ip 10.10.25.82
New object being created
: cluster:nodes> save
: cluster> nodes name 10.10.25.83 ip 10.10.25.83
New object being created
: cluster:nodes> save
: cluster> save
```

To add or remove nodes from the cluster, bring this Controller down and spin it up with the new configuration.

```
Waiting for the cluster to be ready...
Controller is ready.
```

Deploying an NSX Advanced Load Balancer Controller Cluster (Additional)

Begin with a single NSX Advanced Load Balancer Controller that is already installed and will become the leader node.

Procedure

1. Install two new NSX Advanced Load Balancer Controller nodes.

   It is recommended that each NSX Advanced Load Balancer Controller should be on the same management network as the Controller that was already installed. If they are not on the same management network, See Clustering NSX Advanced Load Balancer Controllers of Different Networks. Do not perform any configuration on these newly installed NSX Advanced Load Balancer Controller nodes.

2. Select an IP address to use as the Controller Cluster IP.

   For OpenStack deployments, choose one of the following options:

   a. Pick an unused IP address from the same network to which the Controllers are connected. During clustering, the Controller will automatically create a Neutron port with the Controller cluster IP as a fixed IP address. It ensures that Neutron does not assign that IP address for another purpose.

   b. Explicitly create a Neutron port and use the IP address assigned to that port as the cluster IP address.

3 Use a web browser to navigate to the management IP address of the NSX Advanced Load Balancer Controller that was already installed.

4 Navigate to **Administration > Controller > Nodes** and click **Edit**. The **EDIT CLUSTER** pop-up window appears.

5 In the **Controller Cluster IP** field, enter the shared IP address for the Controller cluster.

6 Configure the Deploying an NSX Advanced Load Balancer Controller Cluster.

Note

- When deploying a Controller Cluster in AWS, enter the Password that was created for the new Controller. For more information, see *Changes for Cluster Set-up for AWS Deployments* topic in the VMware NSX Advanced Load Balancer*Installation Guide*. For deployments in all other environments, the Password can be left blank to use the default initial admin credentials.
- Public IP is specified for all public clouds.

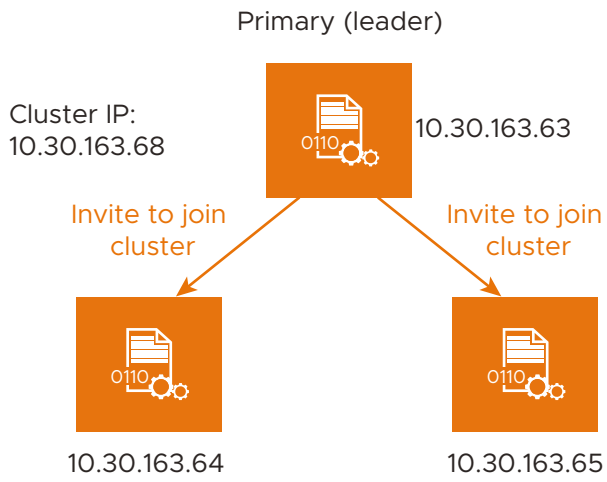
7 After these steps, the incumbent Controller becomes the leader of the cluster and invites the other Controllers to the cluster as follower members. NSX Advanced Load Balancer then performs a warm reboot of the cluster. This process can take 2-3 minutes. The configuration of the leader NSX Advanced Load Balancer Controller node is synchronized to the new follower nodes when the cluster comes online following the reboot.

a The leader NSX Advanced Load Balancer Controller assigns the Controller cluster IP address as a secondary IP address to its management interface.

b For OpenStack deployments, the leader also performs a Neutron API call to set Controller cluster IP address in the list of allowed-address-pairs for the Neutron port corresponding to its management interface.

c To use FQDNs instead of IP address of the Controller nodes, see the following section.

What to do next



The leader Controller assigns the Controller cluster IP address as a secondary IP address to its management interface.

For OpenStack deployments, the leader also performs a Neutron API call to set the Controller cluster IP address in the list of allowed address pairs for the Neutron port corresponding to its management interface.

To use FQDNs instead of IP addresses of controller nodes, see Cluster Configuration with DNS Hostnames.

Deploy SEs in Different Data Centers from Controllers

There is no strict requirement that will prevent Controllers from managing remote SEs. However, it is essential to understand the best practices and ensure the deployment will scale with resilience.

It is recommended to deploy the Controller cluster and its Service Engines in the same data center. Tasks such as SE deployment (copying the SE image), log and metric collection, config propagation, and state backup (such as persistence tables) require connectivity. Increased latency can impact these operations and cause potential failures for data plane traffic.

If the latency is low enough, and the intent is to have the Controllers manage SEs in two or more data centers, redundant Controllers are recommended to be congregated in one data center. If the connectivity to the remote data center is lost, those SEs will not be able to receive configuration updates, but will continue to operate until connectivity is restored.

Note The RTT (round-trip time) value between an NSX Advanced Load Balancer Controller and SE must be less than 70 ms.

Clustering Patched Controllers

Consider the special circumstances of a patched Controller.

For more information on patch process, see Patch Guide.

- The patch process requires logging into the NSX Advanced Load Balancer CLI as the admin user.
- Being able to log in as an admin user implies the factory-default admin password has been changed, as explained in the Strong Default Admin Password section.
- As mentioned, follower nodes are expected to have no configuration, and no changes to the NSX Advanced Load Balancer login credentials. If follower nodes do have credentials, a message of the following form is displayed if one attempts to cluster them:

  ```
  Cluster configuration failed. Unable to add node 1.2.3.4: Login with default credentials
  failed, clean reboot the node and retry.
  ```

  Therefore, before joining two patched Controller-followers to a patched Controller-leader, run reboot clean commands on each follower.

- A Controller which has had its admin password changed has a configuration.

Transition Process

- Trust is established between the leader and new followers over HTTPS.
- The leader exports cluster state information to each follower over SSH.
- Each node locally applies the cluster state information, and restarts services.
- Existing NSX Advanced Load Balancer SEs, originally connected to only the leader node, detect the cluster membership changes and connect to all the NSX Advanced Load Balancer Controller nodes in the cluster.

Note During this transition, the REST API is not available. Generally, the transition takes four to ten minutes.

Transition Status

The progress of transition to a 3-node cluster is indicated by the following status messages:

- `Configuration is in progress`: During this phase, NSX Advanced Load Balancer verifies the configuration and verifies the state information on each of the new nodes. Only nodes with no state can be added to the cluster. NSX Advanced Load Balancer also updates the cluster database, and launches a configuration script on each of the nodes.
- `Inaccessible`: NSX Advanced Load Balancer Controller processes are restarted.
- `Restoring`: NSX Advanced Load Balancer configuration is synchronized across all nodes.
- `Waiting for Service Engines to connect`: NSX Advanced Load Balancer waits for any known SEs that have not yet connected to and registered with any of the NSX Advanced Load Balancer Controller nodes.
- `Ready`

If dismantling a cluster (returning to only a single NSX Advanced Load Balancer Controller node), the status messages are the same. However, during the `Configuration is in progress phase`, the state information on the followers is ignored.

Controller Cluster IP

The NSX Advanced Load Balancer Controller cluster IP address is a single IP address shared by multiple Controllers within a cluster. It is the address to which the web interface, CLI commands, and REST API calls are directed. As a best practice, to access the Controller, you must log in to the cluster IP address instead of the IP addresses of individual Controller nodes.

For cluster IPs, the management IPs of the Controllers must all be in the same subnet.

Note You can use Route 53 with health checks to resolve the cluster's domain name to a Controller IP address directly in AWS deployments where Controllers are on different subnets. For complete information on cluster configuration in AWS, see NSX Advanced Load Balancer ControllerCluster Configuration in AWS.

Cluster IP Advertisement

The NSX Advanced Load Balancer Controller cluster IP is ARPed by the primary Controller (or leader, depending on the infrastructure type) within the cluster. When another Controller becomes the primary, it will send out a gratuitous ARP to claim ownership of the cluster IP.

Administrators can manage any of the Controllers within the cluster by directly accessing the cluster IP address. The Controllers will handle all replication, so there is no requirement to make changes specifically on the primary Controller.

Note In NSX Advanced Load Balancer, the cluster IP is not referred to as floating IP. The term floating IP applies only to OpenStack.

OpenStack Controller Cluster

The two ways to configure OpenStack Controller cluster are as follows:

Method 1

The following are the steps to configure OpenStack Controller cluster:

1 Spin up three or more NSX Advanced Load Balancer Controllers on OpenStack.

2 Consider one Controller for cloud or cluster configuration. If Controller is not reachable from outside OpenStack, assign Floating IP to Controller IP. You can access the controller using Floating IP and configure Cluster VIP on the Controller.

```
avi-dev-venv) ~ $> neutron floatingip-create public
 Created a new floatingip:
 +---------------------+--------------------------------------+
 | Field               | Value                                |
 +---------------------+--------------------------------------+
 | description         |                                      |
 | fixed_ip_address    |                                      |
 | floating_ip_address | 10.176.2.102                         |
 | floating_network_id | d11a54be-2de8-46be-a847-9402d3e2ea35 |
 | id                  | 55cbe4ce-97d4-44fc-ad38-78faf0cbe2d7 |
 | port_id             |                                      |
 | router_id           |                                      |
 | status              | DOWN                                 |
 | tenant_id           | 037e661ac0cb44c89449e5e9b76b9a00     |
 +---------------------+--------------------------------------+
```

```
 (avi-dev-venv) ~ $> neutron floatingip-associate 55cbe4ce-97d4-44fc-ad38-78faf0cbe2d7
2aecadeb-755a-495e-8f19-53301ee63d6b
 Associated floating IP 55cbe4ce-97d4-44fc-ad38-78faf0cbe2d7
```

3 Configure the cloud, wait for the status to be green indicating that the installation is successful.

4 Create a port in OpenStack for cluster VIP. It should be in the NSX Advanced Load Balancer management network.

```
 avi-dev-venv) ~ $> neutron port-show cvip1
 +-----------------------
+-------------------------------------------------------------------------------------+
 | Field                 |
Value                                                                                 |
 +-----------------------
+-------------------------------------------------------------------------------------+
 | admin_state_up        |
True                                                                                  |
 | allowed_address_pairs | {"ip_address": "172.16.0.3", "mac_address":
""}                                    |
 | binding:host_id
|                                                                                       |
 | binding:vif_details   | {"port_filter":
true}                                                        |
 | binding:vif_type      |
vrouter                                                                               |
 | binding:vnic_type     |
normal                                                                                |
 | description
|                                                                                       |
 | device_id
|                                                                                       |
 | device_owner
|                                                                                       |
 | fixed_ips             | {"subnet_id": "4982c62d-ada2-4067-879a-1c5b1ec94ec8",
"ip_address": "172.16.0.3"} |
 | id                    | 1fed2319-a179-4dc5-
b9e5-49853606e7a8                                          |
 | mac_address           |
02:1f:ed:23:19:a1                                                                     |
 | name                  |
cvip1                                                                                 |
 | network_id            | 9feb21ba-6c14-44a3-
a478-1f09e16b60df                                          |
 | port_security_enabled |
True                                                                                  |
 | security_groups       | 367987f6-f373-4637-8867-
aa5b31dc60d2                                              |
 | status                |
DOWN                                                                                  |
 | tenant_id             |
```

```
037e661ac0cb44c89449e5e9b76b9a00                                                    |
 +-----------------------
+-------------------------------------------------------------------------------------+
```

5 Assign Floating IP to cluster VIP if needed. If the NSX Advanced Load Balancer management network is reachable from outside, Floating IP is not required.

```
 (avi-dev-venv) ~ $> neutron floatingip-create public
 Created a new floatingip:
 +---------------------+--------------------------------------+
 | Field               | Value                                |
 +---------------------+--------------------------------------+
 | description         |                                      |
 | fixed_ip_address    |                                      |
 | floating_ip_address | 10.176.2.104                         |
 | floating_network_id | d11a54be-2de8-46be-a847-9402d3e2ea35 |
 | id                  | e5838127-f2f4-47d6-aaba-d5925d082514 |
 | port_id             |                                      |
 | router_id           |                                      |
 | status              | DOWN                                 |
 | tenant_id           | 037e661ac0cb44c89449e5e9b76b9a00     |
 +---------------------+--------------------------------------+

 (avi-dev-venv) ~ $> neutron floatingip-associate e5838127-f2f4-47d6-aaba-d5925d082514
1fed2319-a179-4dc5-b9e5-49853606e7a8
 Associated floating IP e5838127-f2f4-47d6-aaba-d5925d082514

 (avi-dev-venv) ~ $> neutron floatingip-list
 +--------------------------------------+------------------+---------------------
+--------------------------------------+
 | id                                   | fixed_ip_address | floating_ip_address |
port_id                              |
 +--------------------------------------+------------------+---------------------
+--------------------------------------+
 | 55cbe4ce-97d4-44fc-ad38-78faf0cbe2d7 | 172.16.0.2       | 10.176.2.102        |
2aecadeb-755a-495e-8f19-53301ee63d6b |
 | e5838127-f2f4-47d6-aaba-d5925d082514 | 172.16.0.3       | 10.176.2.104        |
1fed2319-a179-4dc5-b9e5-49853606e7a8 |
 +--------------------------------------+------------------+---------------------
+--------------------------------------+
```

6 Configure cluster VIP in NSX Advanced Load Balancer Controller.

7 Use the cluster VIP or the cluster Floating IP to log in to NSX Advanced Load Balancer Controller.

8 Disassociate the Floating IP from Controller IP. It is an optional step. (Because it is done in Step 2).

9 Add the other Controllers to the **Cluster Configuration** page.

Method 2

The following are the steps to configure OpenStack Controller cluster:

1 Spin up only one Controller.

2 Follow steps from 2 to 8 mentioned in method 1.

3 Bring another set of NSX Advanced Load Balancer Controller nodes.

4 Add the Controller nodes in the cluster configuration.

Configure the Cluster IP Using the UI

The steps to add the cluster IP using the NSX Advanced Load Balancer UI are as follows.

Procedure

1. Navigate to **Administration > Controller > Nodes** and click **EDIT**.
2. Enter **Name**.
3. Specify **Controller Cluster IP**.
4. Under **Cluster Node**, click **ADD**.
   a. Enter **Node IP**.
   b. Specify **Hostname**.
   c. Enter **Password**.
   d. Enter **Public IP Address**.
   e. Click **SAVE**.
5. Click **SAVE**.

Assigning NSX Advanced Load Balancer Controller IP in Different Deployments

For an NSX Advanced Load Balancer deployment in a DHCP environment, Controller IP is assigned using the DHCP option. The process for assigning IP to NSX Advanced Load Balancer Controller differs based on the configured cloud if NSX Advanced Load Balancer is configured using static IP.

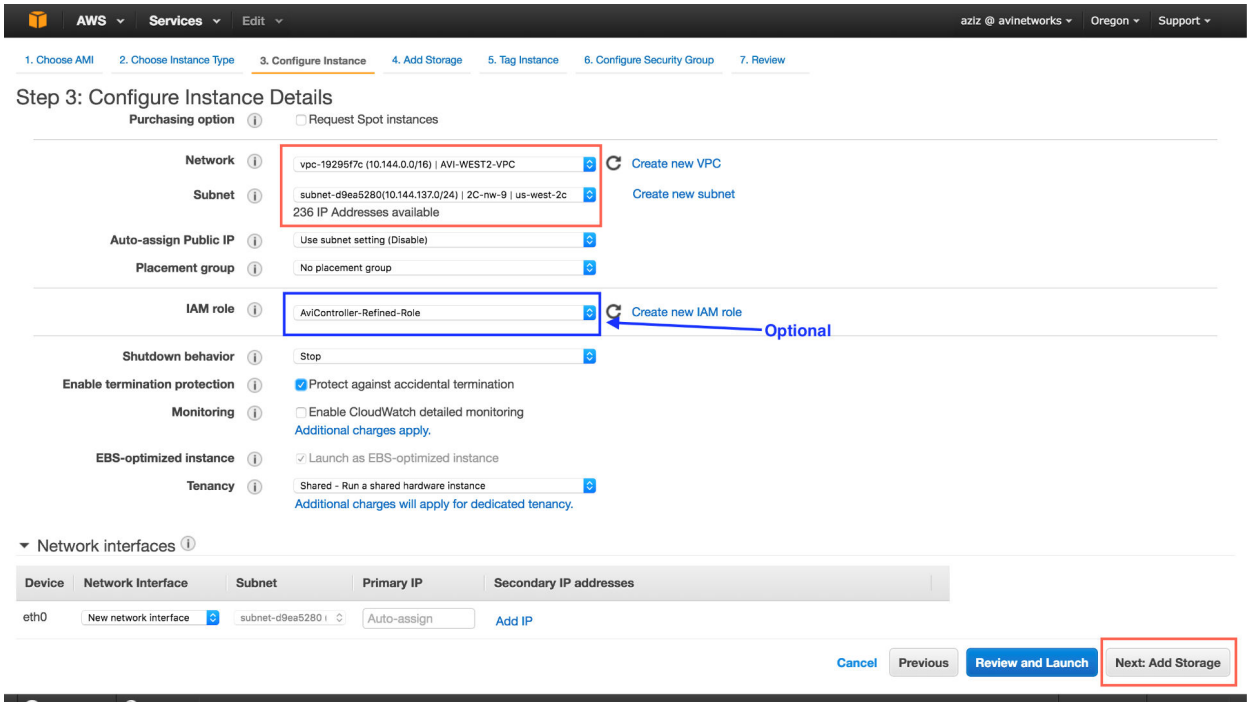
Assigning NSX Advanced Load Balancer IP in VMware Deployment

In any VMware environment, NSX Advanced Load Balancer Controller IP address is assigned using OVF properties during the installation. For more information, see *Installing NSX Advanced Load Balancer in VMware vSphere Environments* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

Assigning NSX Advanced Load Balancer Controller IP in AWS Deployment

In Amazon Web Services (AWS) or Cisco CSP integration, NSX Advanced Load Balancer Controller IP is assigned using metadata. For more information, see Installing NSX Advanced Load Balancer in Amazon Web Services.

The detailed information regarding assigning NSX Advanced Load Balancer Controller IP is available in *Deploying an EC2 Controller Instance* section of *Installing NSX Advanced Load Balancer in Amazon Web Services* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.



Assigning NSX Advanced Load Balancer Controller IP in Linux Server Cloud Deployment

In bare-metal integration or Linux Server Cloud integration (in the case of Docker), NSX Advanced Load Balancer Controller IP is assigned while running the `avi_baremetal_setup.py` script. Enter the Controller IP at the following prompt:

```
Please Enter Controller IP x.x.x.x
```

The following snapshot shows the section of the script when you will be prompted to enter NSX Advanced Load Balancer Controller IP.

7. To run in interactive mode, start by entering "avi_baremetal_setup.py". Here is an example:

```
./avi_baremetal_setup.pyWelcome to AVI Initialization ScriptDPDK Mode:
Pre-requisites(DPDK): This script assumes the below utilities are installed:
docker (yum -y install docker)
Supported Nics(DPDK): Intel 82599/82598 Series of Ethernet Controllers
Supported Vers(DPDK): OEL/CentOS/RHEL - 7.0,7.1,7.2Non-DPDK Mode:
Pre-requisites: This script assumes the below utilities are installed:
docker (yum -y install docker)
Supported Vers: OEL/CentOS/RHEL - 7.0,7.1,7.2Caution : This script deletes existing AVI docker containers &
Do you want to run AVI Controller on this Host [y/n] y
Do you want to run AVI SE on this Host [n] n
Enter The Number Of Cores For AVI Controller. Range [4, 39] 8
Please Enter Memory (in GB) for AVI Controller. Range [12, 125] 24
Please Enter directory path for Avi Controller Config (Default [/opt/avi/controller/data/])
Please Enter disk (in GB) for Avi Controller config (Default [30G])
Do you have separate partition for Avi Controller Metrics? If yes, please enter directory path, else leave
Do you have separate partition for Avi Controller Client Log? If yes, please enter directory path, else lea
Please Enter Controller IP 10.120.0.39
Run SE : No
Run Controller : Yes
Controller Cores : 8
Memory(mb) : 24
Controller IP : 10.120.0.39Disabling AVI Services...
Loading AVI CONTROLLER Image. Please Wait..
kernel.core_pattern = /var/crash/%e.%p.%t.coreInstallation Successful. Starting Services...
```

8. Start Avi Controller on the host to complete installation:

For more information, see step number 7 of *Installing NSX Advanced Load Balancer in a Linux Server Cloud* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

Cluster Configuration with DNS Hostnames

NSX Advanced Load Balancer Controller cluster can be configured with DNS hostnames instead of IP addresses.

Cluster configuration can be changed to use DNS hostnames instead of IP addresses by editing node management IP or hostname in the **Administration > Controller > Nodes** page. It triggers a new Cluster configuration process and updates all nodes to use DNS hostnames.

DNS is resolved when a secure connection is established between Controllers. This usually happens when the cluster is configured or any of the nodes is rebooted. SEs resolve DNS hostname of the Controller when establishing a secure connection to the Controller. This usually happens when the SE is first created and when either an SE or a Controller reboots.

Note If cluster VIP is configured there must be a valid IP address only. An FQDN cannot be configured for cluster VIP.

In addition, cluster configuration with DNS hostnames is not supported for write access vCenter or NSX-T clouds where the Service Engines' Management IP address is statically allocated by the Controller. It is supported for these cloud types if the Service Engines' Management IP address assignment is using DHCP and the DHCP server provides DNS server information that allows the Service Engine to resolve the Controller DNS hostnames.

The cluster configuration process takes the DNS information of the leader node and updates it in the new follower nodes. All nodes can reach the DNS server that is configured in the **Administration > System Settings > DNS/NTP** page of the leader.

SEs can initiate connection using IP address from the VM metadata but are automatically switched to use DNS hostname for the Controller nodes. So it is important for SEs to use the correct DNS server to resolve Controller DNS hostname.

Changing NSX Advanced Load Balancer Controller Cluster Configuration

This section discusses about how to change node membership or node IP address in an NSX Advanced Load Balancer Controller cluster.

Note If you change the IP address of an NSX Advanced Load Balancer Controller node, you must run a script to update the configuration. For more information, see Updating the Configuration Following NSX Advanced Load Balancer Controller IP Address Change.

Remove both Followers (Dismantling the Cluster) Using the UI

The steps to remove both followers from an NSX Advanced Load Balancer Controller cluster are as follows:

Procedure

1 Navigate to **Administration > Controller > Nodes**.

2 Click **EDIT**.

3 Remove each of the follower IP addresses in the **EDIT CLUSTER** pop-up window.

4 Click **SAVE**.

Using the CLI

Log in to the CLI (or CLI shell) and enter the commands shown as below.

Note Ensure that you specify the host IP addresses of theNSX Advanced Load Balancer Controller nodes instead of the IP addresses shown as below.

```
configure cluster
Updating an existing object. Currently, the object is:
+---------------+-----------------------------------------------+
| Field         | Value                                         |
+---------------+-----------------------------------------------+
| uuid          | cluster-eb01bf05-7313-4a4f-91b6-21e46d3c237d |
| name          | cluster-0-1                                   |
| nodes[1]      |                                               |
| name          | node-1                                        |
| ip            | 10.10.25.81                                   |
| vm_ref        | EB01BF05-7313-4A4F-91B6-21E46D3C237D          |
```

```
| vm_mor         |                                              |
| vm_hostname    | node1.controller.local                       |
| nodes[2]       |                                              |
| name           | node-2                                       |
| ip             | 10.10.25.82                                  |
| vm_ref         | EC123A05-7313-4A4F-91B6-21E46D3D46AF         |
| vm_mor         |                                              |
| vm_hostname    | node2.controller.local                       |
| nodes[3]       |                                              |
| name           | 10.10.25.83                                  |
| ip             | 10.10.25.83                                  |
| vm_ref         | EA12C05-7313-4A4F-91B6-21E46D3E256EA         |
| vm_mor         |                                              |
| vm_hostname    | node3.controller.local                       |
| tenant_ref     | admin                                        |
+----------------+----------------------------------------------+
: cluster> no nodes name node-2
Removed nodes with name=node-2
: cluster:nodes> save
: cluster> no nodes name node-3
Removed nodes with name=node-3
: cluster:nodes> save
: cluster> save
```

Configuring the cluster with the Controller nodes at [10.10.25.81'].

If you add or remove nodes from the cluster, you need to bring down this Controller and then spin up the Controller up with the new configuration.

```
Waiting for the cluster to be ready...
Controller is ready.
```

After saving,

- The cluster is dismantled and recovered into a single node.
- Each follower is re-imaged into its default state with no configuration and no access to the leader.
- The leader holds the configuration. The SEs continue to connect to the leader.

For more information about the transition process, including descriptions of the status messages that appear, see Clustering Patched Controllers.

Note During the transition from a cluster to a single node, the REST API will be unavailable for around two minutes.

Change a Follower Node Using the UI

The steps to remove one of the follower nodes and add another one are as follows.

Prerequisites

Note If the node is removed and replaced with a different node (different VM, container, or bare-metal server), see Changing NSX Advanced Load Balancer Controller Cluster Configuration. The cluster has to be Replace a Follower Node with a New Node, and recreated using the new node.

Procedure

1 Navigate to **Administration > Controller > Nodes**.

2 Click **EDIT**.

3 Edit the IP address for the follower node to change.

4 Click **SAVE**.

Using the CLI

Log in to the CLI (or CLI shell) and enter the commands shown in the following example.

Note Ensure that you enter the host IP addresses of the NSX Advanced Load Balancer Controller nodes instead of the IP addresses shown in the example.

```
configure cluster
Updating an existing object. Currently, the object is:
+---------------+-----------------------------------------------+
| Field         | Value                                         |
+---------------+-----------------------------------------------+
| uuid          | cluster-eb01bf05-7313-4a4f-91b6-21e46d3c237d |
| name          | cluster-0-1                                   |
| nodes[1]      |                                               |
| name          | node-1                                        |
| ip            | 10.10.25.81                                   |
| vm_ref        | EB01BF05-7313-4A4F-91B6-21E46D3C237D          |
| vm_mor        |                                               |
| vm_hostname   | node1.controller.local                        |
| nodes[2]      |                                               |
| name          | node-2                                        |
| ip            | 10.10.25.82                                   |
| vm_ref        | EC123A05-7313-4A4F-91B6-21E46D3D46AF          |
| vm_mor        |                                               |
| vm_hostname   | node2.controller.local                        |
| nodes[3]      |                                               |
| name          | 10.10.25.83                                   |
| ip            | 10.10.25.83                                   |
| vm_ref        | EA12C05-7313-4A4F-91B6-21E46D3E256EA          |
| vm_mor        |                                               |
| vm_hostname   | node3.controller.local                        |
| tenant_ref    | admin                                         |
+---------------+-----------------------------------------------+
: cluster> no nodes name node-3
Removed nodes with name=node-3
: cluster:nodes> save
```

```
: cluster> nodes name node-4 ip 10.10.25.84
Removed nodes with name=node-4
: cluster:nodes> save
: cluster> save
```

Configuring the cluster with the Controller nodes at [u'10.10.25.81', '10.10.25.82', '10.10.25.84']. If you add or remove nodes from the cluster, you should bring down this Controller and back up with the new configuration.

```
Waiting for the cluster to be ready...
Controller is ready.
```

After saving,

- The removed follower node is sent an API request asking it to gracefully clear its state.
- Since the old follower is not always expected to clear its state, the leader will forcibly remove it if necessary.
- The new follower node is added.

Replace the Leader Node

To replace the leader node, power it down to force one of the followers to take over leadership.

Procedure

1. Power down the leader node and leave it powered off for several minutes while one of the followers assumes leadership.
2. Use the steps for adding a new follower node.

   Note The leader cannot be directly removed. Instead, it must be replaced with another leader.

   To replace a follower, see Change a Follower Node Using the UI.

Replace a Follower Node with a New Node

This procedure is applicable for replacing a follower node with a completely different node that is not currently in the cluster. For instance, these steps apply to replacing a follower node with a new VM or bare metal server. The node cannot be replaced simply by inserting the node into the network and editing the cluster configuration information.

Procedure

1. Log in to the leader node, and use the steps to dismantle the cluster by Remove both Followers (Dismantling the Cluster) Using the UI nodes.
2. On the new node, install NSX Advanced Load Balancer. It involves spinning up a new virtual machine (VM) or container from the NSX Advanced Load Balancer Controller package (OVA, QCOW or Docker image) on the new node.

3 Follow the steps mentioned in the Deploying an NSX Advanced Load Balancer Controller Cluster section.

Updating the Configuration Following NSX Advanced Load Balancer Controller IP Address Change

The management IP addresses of each NSX Advanced Load Balancer Controller node should be static. It applies to single-node deployments and 3-node deployments.

The cluster configuration and runtime configuration each contain the IP information for the cluster. If the IP address of a leader or follower node changes (for example, due to DHCP), this script must be run to update the IP information. The cluster will not function properly until the cluster configuration is updated.

If the IP address of an NSX Advanced Load Balancer Controller node is changed for any reason (such as DHCP), the following script must be used to update the cluster configuration. This applies to single-node deployments and cluster deployments.

To repair the cluster configuration after an NSX Advanced Load Balancer Controller node's IP address is changed, run the `change_ip.py` script.

The script is located in the following directory:

`/opt/avi/python/bin/cluster_mgr/change_ip.py`

Note

1 The change IP script only changes the NSX Advanced Load Balancer cluster configuration. It does not change the IP address of the host or the virtual machine on which Controller services are running. For example, it does not update the `/etc/network/interfaces` file in a VMware-hosted Controller. One must change the IP address for the VM in the vApp properties in VMware.

2 Special consideration is required when changing the IP addresses of Controllers in a bare-metal configuration. For more information, see Changing Controller IP Addresses in a Bare-metal Environment.

Configure the Script

The script can run on the NSX Advanced Load Balancer Controller node whose management IP address changed, or on another NSX Advanced Load Balancer Controller node within the same cluster.

Note Before running the script, make sure that new IPs are working on all nodes and are reachable across nodes. If one or more IPs are not accessible, the script makes a best-effort update, but upon restoring connectivity, there is no guarantee the cluster will be back in sync.

The script must run on one of the nodes that is in the cluster. If the script is run on a node that is not in the cluster, the script will fail. The following is the list of parameters:

- `-i ipaddr`: Specifies the new IP address of the node on which the script is run.
- `-o ipaddr`: Specifies the IP address of another node in the cluster.
- `-m subnet-mask`: If the subnet also changed, use this option to specify the new subnet. Enter the mask in the following format: 255.255.255.0
- `-g gateway-ipaddr`: If the default gateway also changed, use this option to specify the new gateway.

Note The `-m` and `-g` options apply to all IP addresses in the cluster.

Updating IP Information for a Single-node Deployment

To update NSX Advanced Load Balancer Controller IP information for a single-node deployment, use a command string such as the following:

```
*change_ip.py -i **ipaddr*
```

This command is run on node 10.10.25.81. Whereas no other nodes are specified, this is assumed to be a single-node cluster (just this NSX Advanced Load Balancer Controller).

```
username@avi:~$ | change_ip.py -i 10.10.25.81
```

In the following example, the default gateway of the node also has changed.

```
username@avi:~$ | change_ip.py -i 10.10.25.81 -g 10.10.10.1
```

Updating IP Information for an NSX Advanced Load Balancer Controller Cluster

To update NSX Advanced Load Balancer Controller IP information for a cluster, use a command string such as the following:

Note Before executing `change_ip.py`, ensure that all new IPs are reachable from one another over SSH ports (22 for regular, 5098 for containers).

```
change_ip.py -i **ipaddr **-o ipaddr -o ipaddr
```

This command is run on node 10.10.25.81, which is a member of a 3-node cluster that also contains nodes 10.10.25.82 and 10.10.25.83.

```
username@avi:~$ | change_ip.py -i 10.10.25.81 -o 10.10.25.82 -o 10.10.25.83
```

The script can run on any of the nodes in the cluster. The following example is run on node 10.10.25.82.

```
username@avi:~$ | change_ip.py -i 10.10.25.82 -o 10.10.25.81 -o 10.10.25.83
```

Note After executing `change_ip.py`, in case of failure, use `recover.py` to convert nodes to single nodes and create the 3-node cluster again. For more information, see Recover a Non-Operational Controller Cluster.

To verify, go to the Controller nodes page and ensure all nodes are CLUSTER_ACTIVE.

Change Controller IPs on Nutanix Cluster

For the Controller cluster to come up with the new IPs, follow the steps below.

1 Change the IP address of each Controller node within the cluster to the new IP by manually editing the network-scripts on the host and changing the interface configuration.

   For instance, `/etc/network/interfaces/` file on NSX Advanced Load Balancer Controller virtual machine must be modified as follows (if using static IP):

   ```
   auto lo
   iface lo inet loopback

   auto eth0
   iface eth0 inet static
    address <ipv4 address>
    netmask 24
    gateway <ipv4 gw>
   ```

2 Ensure that the new Controller IP addresses are reachable in the network from the other Controller nodes.

3 Run `/opt/avi/python/bin/cluster_mgr/change_ip.py` on the Controller to reflect the above IP address change.

4 Reboot the Controller.

The above procedure is for a single node cluster specified in step 2. For a three-node cluster deployment, change the IPs on all the Controllers and run the command as shown below from any Controller node to update NSX Advanced Load Balancer Controller IP information for a cluster:

```
username@avi:~$ change_ip.py -i **ipaddr **-o ipaddr -o ipaddr
```

Where,

- `-i ipaddr`: Specifies the new IP address of the node on which the script is run.
- `-o ipaddr`: Specifies the IP address of another node in the cluster.

- `-m subnet-mask`: If the subnet also changed, use this option to specify the new subnet. Enter the mask in the following format: 255.255.255.0.
- `-g gateway-ipaddr`: If the default gateway also changed, use this option to specify the new gateway.

Customizing NSX Advanced Load Balancer Configuration

NSX Advanced Load Balancer Controller cluster instances are frequently required to be deployed from a common initial single Controller configuration that differs from the factory-default settings. This can be accomplished by creating a JSON file with the required object definitions and then using it to deploy subsequent controllers, which, as leaders, can then add followers to become clusters. You will have a set of identically initialized Controller clusters, ready to be individualized as needed.

Create `setup.json`

In most cases, these objects can be created by referring to the NSX Advanced Load Balancer REST API section.

The following example updates the system configuration by adding 8.8.8.8 to the DNS configuration:

```
{
    "SystemConfiguration": [
        {
            "dns_configuration": {
                "search_domain": "",
                "server_list": [
                    {
                        "type": "V4",
                        "addr": "8.8.8.8"
                    }
                ]
            }
        }
    ]
}
```

In the case of complex objects such as the `SSLKeyAndCertificate` object, the JSON file can be created by running a diff command against two configuration files. In a typical deployment, generating `setup.json` on a test Controller environment is recommended. This generated file can then be used as a template for actual deployments. An NSX Advanced Load Balancer Controller configuration snapshot can be taken using the `export` CLI command.

```
> export configuration file before.cfg
Please enter the passphrase to encrypt configuration:
Downloaded the attachment to before.cfg
Completed writing the export configuration to before.cfg
```

Configure objects using the UI or CLI, as required.

```
> export configuration file after.cfg
Please enter the passphrase to encrypt configuration:
Downloaded the attachment to after.cfg
Completed writing the export configuration to after.cfg
```

Beyond this, configuration diff can be taken using a Python script.NSX Advanced Load Balancer has written expressly to customize initial configuration of another Controller.

```
/opt/avi/scripts/diff_config.py -f before.cfg -t after.cfg > setup.json
```

User passwords can be encrypted using the following code when creating `setup.json` with the `User` object.

```
/opt/avi/scripts/avi_passwd_tool.py --password admin --salt
fF6ngAb3pvPgpbkdf2_sha256$100000$fF6ngAb3pvPg$ijkEue1M9fR/
qsLVgzvPe7N0VvOxIjDiJVmK9NIx+0Q=$6$fF6ngAb3pvPg$CqAKtNRZtgXtJchrPmoxUgdLFM7rFGmta1tWb7sobQI4iS
ZAY2QuAOBNtboVGrmDYPMCvqXXH6lARr9RedCJT.
```

Deployment using `setup.json`

It is recommended to take a configuration backup before deploying the Controller using the `setup.json` file created by the Python script. Use the following command to create an encrypted backup of the existing configuration.

```
/opt/avi/python/bin/portal/manage.py export_configuration --file ~/setup-old.json --
passphrase secret
```

For a Mesos/Bare-Metal Deployment

Copy `setup.json` to the persistent directory in the host mounted as `/vol` in the Controller container. If you are using the `avi_baremetal` script, the default location is `/opt/avi/controller/data` on the host. When deploying the Controller as a container, `setup.json` can be passed as an additional argument to the `avi_baremetal_setup.py` script. For example:

```
./avi_baremetal_setup.py -c -cc 4 -cm 12 -i 10.10.22.108 -m 10.10.22.108 --setup-json /root/
configs/avi-setup.json
```

For a Controller Deployment as a Virtual Machine

Wait until the Controller comes up. Place the config file on the Controller as `/var/lib/avi/etc/setup.json` (note the filename). Upon reboot or fresh-start, the NSX Advanced Load Balancer Controller will self-configure using the provided `setup.json` file.

```
reboot
```

For an OpenStack Deployment

UserData config size is limited to 48 Kb if the size of `setup.json` is within allowable limits.

```
>nova boot --config-drive true --image avicontroller --key-name mykey --flavor 4 --user-data /
root/my-avi-config.json --nic net-id=7402bf4f-240f-4172-99c1-90000ea45f86 avicontrollers

>nova boot --config-drive false --image avicontroller --key-name mykey --flavor 4 --user-
data /root/my-avi-config.json --nic net-id=7402bf4f-240f-4172-99c1-90000ea45f86 avicontrollers
```

If the `setup.json` size is bigger then than the allowable limit, setup.json can be uploaded and referred to in the deployment phase.

User data can refer to the file using the “`url`” or “`file`” tag. Following is an example of `my-avi-config-url.json` with URL.

```
{
 "META": {
     "init_config": {
         "url": "https://s3-us-west-2.amazonaws.com/avi-controller-configs/linuxserver-
awsipam-setup.json"
     }
   }
}
```

Following is an example of `my-avi-config-url.json` with filepath.

```
{
  "META": {
    "init_config": {
      "file": "/vol/linuxserver-awsipam-setup.json"
    }
  }
}
```

Following is an example of deployment.

```
>nova boot --config-drive true --image avicontroller --key-name mykey --flavor 4 --user-data /
root/my-avi-config-url.json --nic net-id=7402bf4f-240f-4172-99c1-90000ea45f86 --min-count=3 --
max-count=3 avicontrollers

>nova boot --config-drive false --image avicontroller --key-name mykey --flavor 4 --user-data /
root/my-avi-config-url.json --nic net-id=7402bf4f-240f-4172-99c1-90000ea45f86 --min-count=3 --
max-count=3 avicontrollers
```

For an AWS Deployment

UserData config size is limited to 16Kb. If the size of `setup.json` is within allowable limits, cut-paste the `my-avi-config.json` into the user-data section during launch from AWS Web Console.

```
# metasvc userdata  ec2-run-instances ami-b7ea27d7 -f /root/my-avi-config.json -t c4.2xlarge
-s subnet-62f1b707 -g sg-642d8d02
```

If the `setup.json` size is bigger than the allowable limit, cut-paste the `my-avi-config-url.json` into the user-data section during launch from AWS Web Console.

```
# metasvc userdata indirection ec2-run-instances ami-b7ea27d7 -f /root/my-avi-config-url.json
-t c4.2xlarge -s subnet-62f1b707 -g sg-642d8d02
```

`my-avi-config-url.json` follows similar formats as discussed in the OpenStack section. Following is a sample `my-avi-config-url.json` file for the S3 bucket.

```
{
  "META": {
    "init_config": {
      "s3": "avi-controller-configs/linuxserver-awsipam-setup.json"
    }
  }
}
```

For uploading `setup.json` on the S3 bucket:

- Public : use the `url` style or `s3` style
- Private through RBAC on VM: use the `s3` style. The VM role must have `s3:GetObject` action allowed to be able to s3-get the object using IAM.
- Private through RBAC on S3-bucket: use the `s3` style. The VM role must have AWS access. The S3 bucket must have permissions for the account or user or VM role to download the object.

Example Bucket Policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "AddPerm",
       "Effect": "Allow",
       "Principal": {
         "AWS": [
           "arn:aws:iam::139284885014:role/BM-AviController-Role",
           "arn:aws:iam::139284885014:root"
          ]
        },
       "Action": "s3:*",
       "Resource": "arn:aws:s3:::avi-controller-configs/*"
```

```
      }
    ]
  }
```

For an Azure Deployment

There are two ways to provide initial configuration in an Azure environment:

- Using Azure CLI
- Using ARM (Azure Resource Manager) template

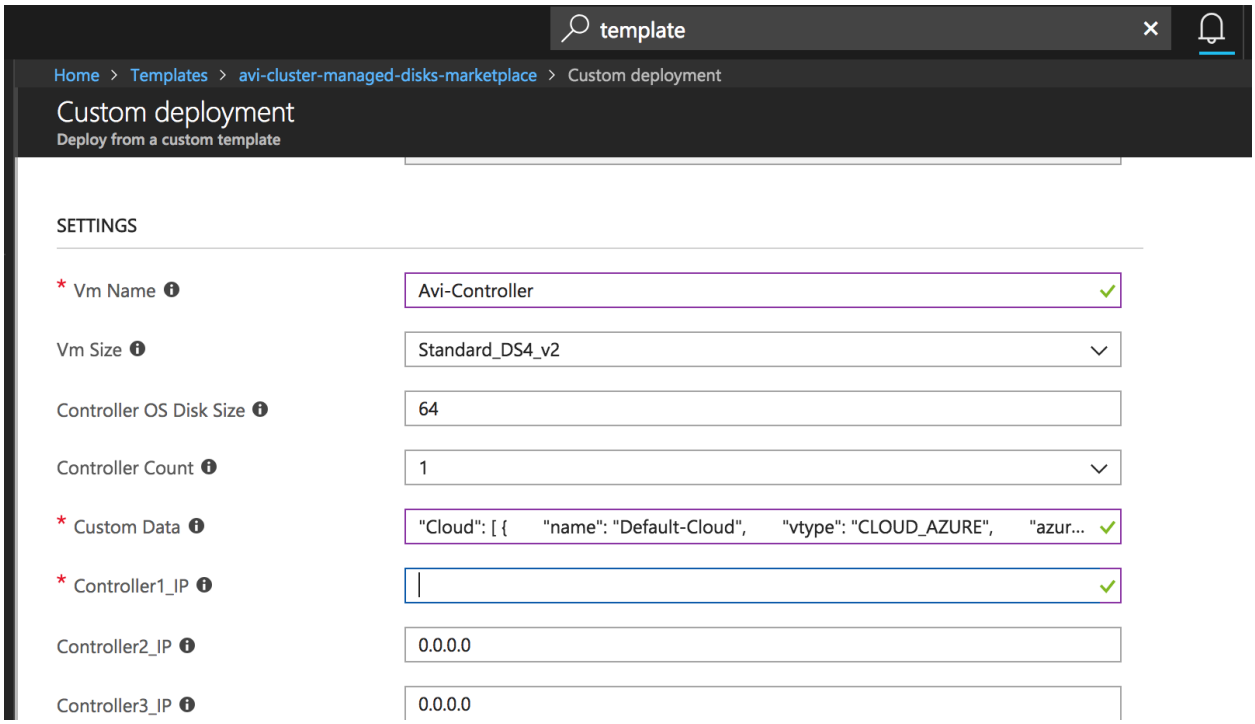
Using Azure CLI

If the NSX Advanced Load Balancer Controller is deployed using Azure CLI, the JSON file can be provided during the deployment.

```
az vm create --resource-group rahulr-jenkins-resource-group --location centralus --image
avi-networks:avi-vantage-adc:avi-vantage-adc-byol:17.2.7 --name Avi-Test-Controller --size
Standard_F8s --subnet /subscriptions/<subscription_id>/resourceGroups/<resource_group_name>/
providers/Microsoft.Network/virtualNetworks/<vnet_name>/subnets/rahulr-subnet --public-ip-
address "" --nsg "" --custom-data ./initial_config.json
```

Using Azure ARM Template

If the NSX Advanced Load Balancer Controller is deployed using the ARM template, the JSON data is provided as the **Custom Data** on the **Custom deployment** page of the Azure portal. Navigate to **Home > Templates > avi-cluster-managed-disks-market place > Custom deployment**.



Use the required JSON template in the **Custom Data** field. For reference, the below JSON template is for adding 8.8.8.8 to DNS configuration. Copy the JSON configuration mentioned below, and add it to the **Custom Data** field in the ARM template.

```
{
    "SystemConfiguration": [
        {
            "dns_configuration": {
                "search_domain": "",
                "server_list": [
                    {
                        "type": "V4",
                        "addr": "8.8.8.8"
                    }
                ]
            }
        }
    ]
}
```

Clustering NSX Advanced Load Balancer Controllers of Different Networks

Controller clusters provide high availability (HA), redundancy, and increased analytic workload scale.

Controllers communicate with each other over a single management IP address, the Controller Cluster IP. They also use this path to communicate with all SEs within the fabric.

Although Controllers do not have to exist within the same limitations, consider the following conditions:

- Controllers must be within the same region (ideally the same data center). It helps synchronize the databases and perform actions such as log indexing and data retrieval.
- Controllers have the option of sharing a cluster IP address. The cluster IP address is owned by the primary Controller within the cluster. To share an IP address, all Controllers must have a NIC in the same network.
- Each Controller must have access to the IP addresses of other Controllers through configured network routes.
- RTT (round-trip time) value between two Controller nodes must be less than 20 milliseconds.

Considerations

AWS

AWS Availability Zones (AZs) provide redundancy and separate fault domains. All AWS regions support a minimum of two AZs. To leverage HA provided by AWS AZs, it is recommended to deploy different NSX Advanced Load Balancer Controller instances of a cluster in different AZs.

Azure

The Controller cluster must be running inside the Azure cloud. Additionally, consider the following information:

- Azure credentials (username and password or application ID) which have **contri butor** privilege access over the Controller cluster VMs and AviController role access over the virtual network that is hosting the Controller cluster.
- `Subscription_id` of the subscription where the Controller virtual machines are running.

OpenStack

OpenStack requires NSX Advanced Load Balancer to maintain a cluster IP address. So, NSX Advanced Load Balancer deployed into an OpenStack cloud does not support clustering of NSX Advanced Load Balancer Controllers present in different networks.

High Availability for NSX Advanced Load Balancer Controllers

NSX Advanced Load Balancer can run with a single Controller (single-node deployment) or a three-node Controller cluster.

In a deployment that uses a single Controller, that Controller performs all administrative functions and all analytics data gathering and processing.

Adding two additional nodes to create a three-node cluster provides node-level redundancy for the Controller and maximizes performance for CPU-intensive analytics functions. Whereas the lone Controller in a single-node deployment performs all administrative functions and analytics data collection and processing, these tasks are distributed in three-node cluster.

In a three-node Controller cluster, one node is the primary (leader) node and performs the administrative functions. The other two nodes are followers (secondaries), and perform data collection for analytics, in addition to standing by as backups for the leader.

Operation of NSX Advanced Load Balancer Controller HA

This section explains how HA operates within an NSX Advanced Load Balancer Controller cluster.

Quorum

NSX Advanced Load Balancer Controller-level HA requires a quorum of Controller nodes to be up. In a three-node Controller cluster, quorum can be maintained if at least two of the three Controller nodes are up. If one of the Controllers fails, the remaining two nodes continue service and NSX Advanced Load Balancer continues to operate. However, if two of the three nodes go down, the entire cluster goes down, and NSX Advanced Load Balancer stops working.

Failover

Each Controller node in a cluster periodically sends heartbeat messages to the other Controller nodes in the cluster through an encrypted SSH tunnel using TCP port `22` (port `5098` if running as Docker containers).

The heartbeat interval is ten seconds. The maximum number of consecutive heartbeat messages that can be missed is four. If one of the Controllers does not hear from another Controller for 40 seconds (four missed heartbeats), the other Controller is assumed to be down.

If only one node is down, quorum is maintained, and the cluster can continue to operate.

- If a follower goes down but the leader node remains up, access to virtual services continues without interruption.
- If the primary (leader) node goes down, the member nodes form a new quorum and elect a cluster leader. The election process takes about 50-60 seconds, and during this period, there is no impact on the data plane. The SEs will continue to operate in the headless mode, but the control plane service will be unavailable. During this period, users will be unable to create a VIP through LBaaS or use the NSX Advanced Load Balancer UI, API, or CLI.

Converting a Single-Node Deployment to a 3-node Cluster

To convert a single-node NSX Advanced Load Balancer Controller deployment into a three-node deployment, use the following steps.

In this procedure, the Controller node that is already deployed in the singe-node deployment is referred to as the incumbent Controller.

Procedure

1. Install one new NSX Advanced Load Balancer Controller node. During installation, configure the following settings:
   - Node management IP address
   - Gateway address
2. Connect the management interface of each new Controller node to the same network as the incumbent Controller. After the incumbent Controller detects the two new Controller nodes, the incumbent Controller will become the primary (leader) Controller for the three-node cluster.

3 Use a web browser to navigate to the management IP address of the primary (leader) Controller.

4 Navigate to **Administration > Controller > Nodes**, and click **Edit**. The **Edit CLUSTER** pop-up window appears.

5 In the **Controller Cluster IP** field, enter the shared IP address for the Controller cluster.

6 In the **Public IP Address or Hostname** field, enter the management IP addresses of the new Controller nodes.

Note Password for the admin account is required for each node of the cluster for configuring cluster in AWS Cloud.

7 After these steps, the incumbent Controller becomes the primary (leader) for the cluster and invites the other Controllers to the cluster as members. NSX Advanced Load Balancer then performs a warm reboot of the cluster. This process can take two- three minutes. The configuration of the primary (leader) Controller is synchronized to the new member nodes when the cluster comes online following the reboot.

Control Plane High Availability

You can deploy a set of three NSX Advanced Load Balancer Controllers as a High Availability cluster, as a best practise. In a cluster deployment, one of the Controllers acts as a leader. It performs load balancing and configuration management for the cluster. While the other two Controllers are followers, they collaborate along with the leader to perform data collection from SEs and process analytics data.

High Availability for NSX Advanced Load Balancer Controllers

NSX Advanced Load Balancer can run with a single Controller (single-node deployment) or with a three-node Controller cluster. In a deployment that uses a single Controller, that Controller performs all the administrative functions, it also gathers and processes all the analytics data.

You can create a three-node cluster by adding two additional nodes. This three-node cluster provides a node-level redundancy for the Controller and maximizes performance for CPU-intensive analytics functions. However, for a single Controller in a single-node deployment, it performs all administrative functions; collects and processes all the analytics data. These tasks are distributed in a three-node cluster.

In a three-node Controller cluster, one node is the primary (leader) node and performs the administrative functions. The other two nodes are followers (secondaries), and perform data collection for analytics, in addition to standing by as backups for the leader.

Operation of NSX Advanced Load Balancer Controller High Availability

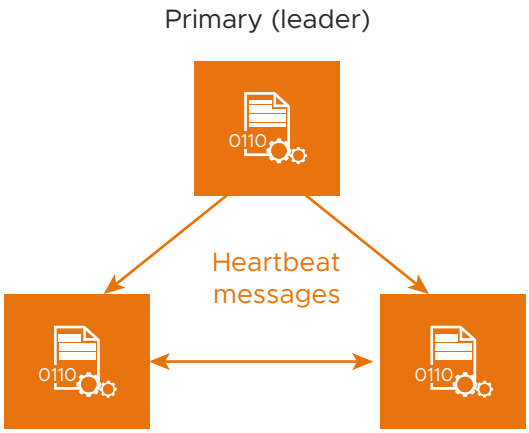
This section explains how high availability (HA) operates within NSX Advanced Load Balancer Controller cluster.

Quorum

The Controller level HA requires a quorum of NSX Advanced Load Balancer Controller nodes to be up. In a three-node Controller cluster, quorum can be maintained if at least two of the three Controller nodes are up. If one of the Controllers fail, the remaining two nodes continues service and NSX Advanced Load Balancer continues to operate. However, if two of the three nodes go down, then the entire cluster goes down, and NSX Advanced Load Balancer stops working.

Failover

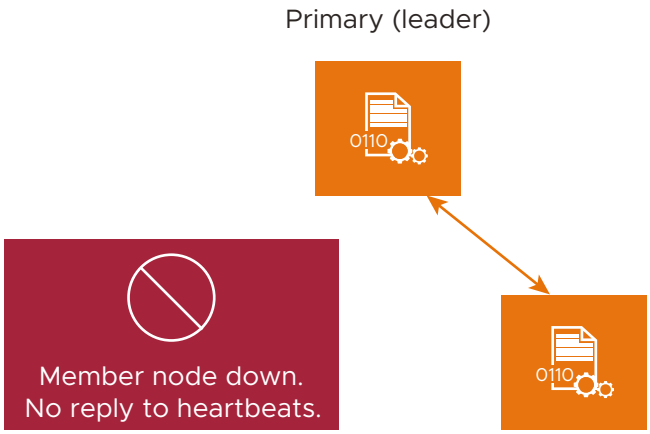
Each Controller node in a cluster periodically sends heartbeat messages to the other Controller nodes in a cluster, through an encrypted SSH tunnel using TCP port 22 (port 5098 if running as Docker containers).



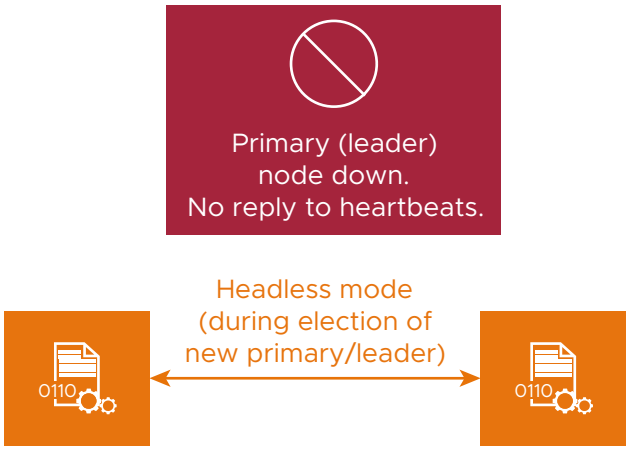
The heartbeat interval is ten seconds. The maximum number of consecutive heartbeat messages that can be missed is four. If one of the Controllers does not hear from another Controller for 40 seconds (four missed heartbeats), then the other Controller is assumed to be down.

If only one node is down, then the quorum is still maintained and the cluster can continue to operate.

If a follower node goes down but the primary (leader) node remains up, then the access to virtual services continues without any interruption.



- If the primary (leader) node becomes unavailable because of events such as a hardware failure or a snapshot of the leader node VM being taken, the member nodes will form a new quorum and elect a cluster leader. The election process takes about 50-60 seconds and during this period, there is no impact on the data plane. The SEs will continue to operate in the `Headless mode`, but the control plane service will not be available. During this period, you cannot create a VIP through LBaaS or use the NSX Advanced Load Balancer user interface, API, or CLI.



Converting a Single-Node Deployment to a Three-node Cluster

This section explains the process to convert a single-node deployment to a three-node cluster.

In this procedure, the NSX Advanced Load Balancer Controller node that is already deployed in the singe-node deployment is referred to as the incumbent NSX Advanced Load Balancer Controller.

To convert a single-node NSX Advanced Load Balancer Controller deployment into a three-node deployment, follow the steps below:

1 Install a single new Controller node. During installation, configure the following settings:

   - Node management IP address

- Gateway address

2 Connect the management interface of each new Controller node to the same network as the incumbent Controller. After the incumbent Controller detects the two new Controller nodes, the incumbent Controller becomes the primary (leader) Controller node for the three-node cluster.

3 Use a web browser to navigate to the management IP address of the primary (leader) Controller node.

4 Navigate to **Administration > Controller > Nodes** and click **Edit**. The **Edit Cluster** window appears.

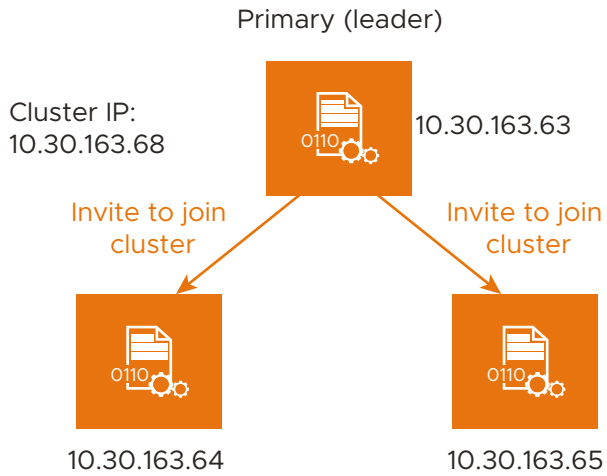
5 In the **Controller Cluster IP** field, specify the shared IP address for the Controller cluster.

6 In the **Public IP Address or Host Name** field, specify the management IP address of the new Controller node.

Note To configure cluster in AWS Cloud, each node of the cluster requires admin account credentials.

The incumbent Controller becomes the primary (leader) for the cluster and invites the other Controller to the cluster as members.

The NSX Advanced Load Balancer then performs a warm reboot of the cluster. This process can take two or three minutes. After the reboot, the configuration of the primary (leader) Controller is synchronized to the new member nodes once the cluster appears online.



To know more about cluster HA, see the links below:

- Controller Cluster IP
- Clustering NSX Advanced Load Balancer Controllers of Different Networks
- Impact of a Controller Failure

For more information on how to Enable Per-app SE mode for a Service Engine Group, see *Per-app SE Mode* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Switching Among Controllers Using the Controller Site Selector

This section discusses the steps to configure Controller sites and then switching between them through the NSX Advanced Load Balancer UI.

Synchronize the list of sites across all participating Controller sites using a script, Ansible or Terraform, to create a primary list of Controller sites. Thereafter, this information is propagated to every Controller deployment using one of the automation methods.

For example, to define the trio of Controller sites mentioned in the subsequent Controller Site Selector Use section, you must have two JSON payloads to POST to each `/api/controllersite`. The two POSTs from any given Controller must point to the other two. Thus, in this example, there is a total of six POSTs.

Note Adding the local Controller as a Controller site entry is no longer required. If at least one record does not match the current site, then NSX Advanced Load Balancer automatically shows the current site's hostname (if applicable) or IP address.

From the Earth site:

{

"name": "Mars",

"address": "192.0.2.20"

}, {

"name": "Venus",

"address": "192.0.2.30"

}

From the Mars site:

{

"name": "Earth",

"address": "192.0.2.10"

}, {

"name": "Venus",

"address": "192.0.2.30"

}

From the Venus site:

{

"name": "Earth",

"address": "192.0.2.10"

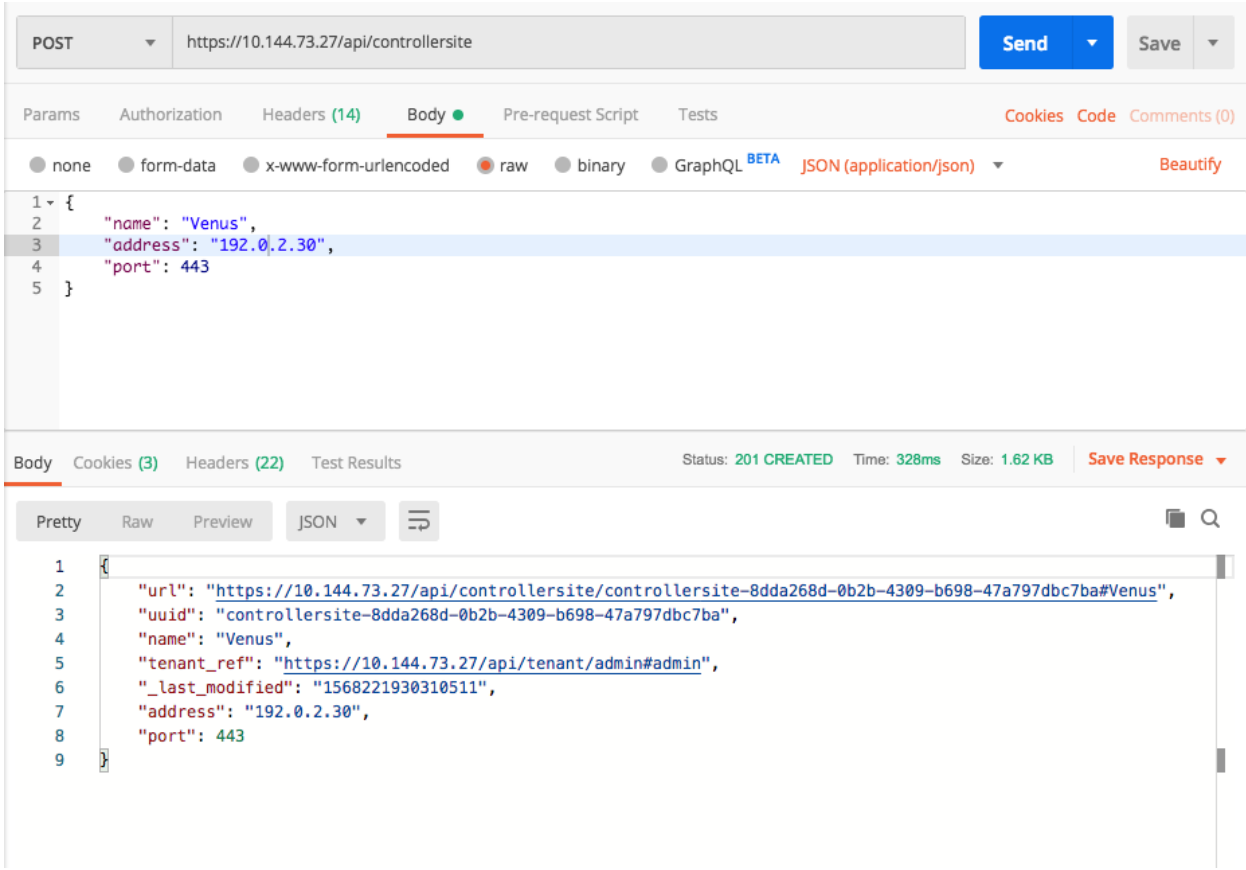
}, {

"name": "Mars",

"address": "192.0.2.20"

}

Example: POST (all releases)



Note

- A Controller site maps to one Controller name and IP address. Thus, for the feature to work with any Controller in a cluster, three local Controller names and corresponding IP addresses must be configured as sites. In the above example, consider three POSTs as shown below:

  {

  "name": "Earth",

  "address": "192.0.2.10"

  }, {

  "name": "Earth2",

  "address": "192.0.2.11"

  }, {

  "name": "Earth3",

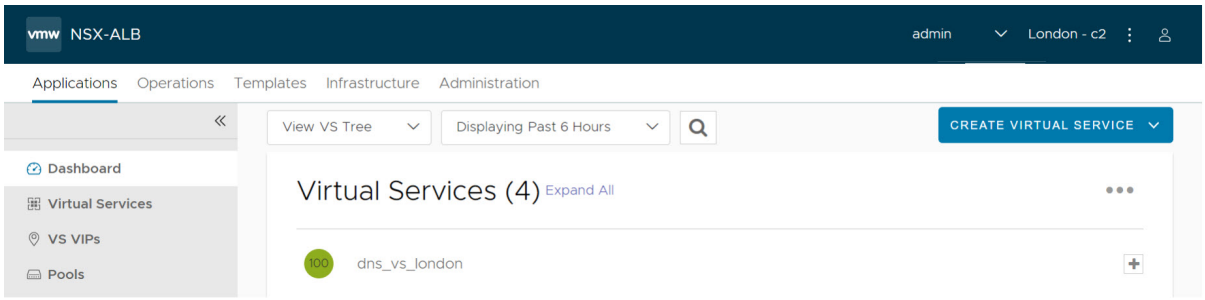
  "address": "192.0.2.12"

  }

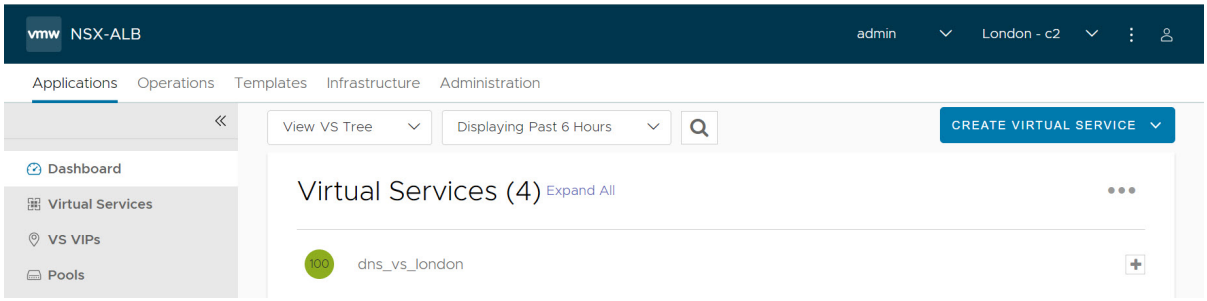
- The REST API call api/controllersite reveals the set to which the user has the `PERMISSION_CONTROLLERSITE` access right. Lack of `PERMISSION_CONTROLLERSITE` hides the Controller site name and drop-down icon from the main menu bar.
- Controller sites can only be added/modified through the CRUD call at `/api/controllersite` with `X-Avi-Version` set in the header. Adding a Controller site to a list of sites requires that at least two fields be given in the payload, name and address.
- Site configuration is locally significant, that is, adding site2 to site1's list does not automatically result in site1 appearing in site2's list.

Controller Site Selector Use

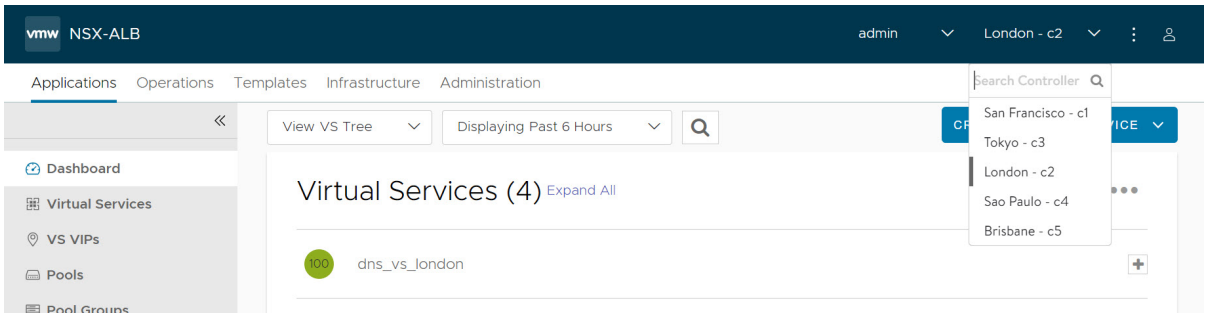
1 When there is but one Controller site record in the set, just the Controller's name appears to the left of the ellipsis (three dots).



2 On the other hand, if there are multiple records, a V-shaped drop-down icon appears to the left of the ellipsis (three dots).



3 Clicking on the drop-down icon reveals the Controller sites within the set to which the user has access.



4 Hovering over a particular Controller name causes it to be highlighted. Select the Controller to switch to that site.

5 The selected Controller opens in a new tab, while the previous tab remains open in the background. The user can now communicate directly with the new Controller. Due to SSO being active, no login screen was presented by the first Controller. The login was performed automatically, behind the scenes.

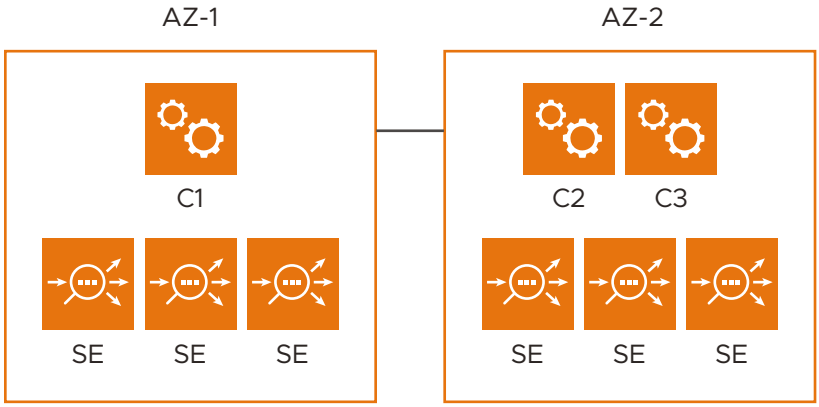
Controller Cluster Deployment Across Two Availability Zones

This section discusses Controller cluster deployments across two availability zones (AZ), what happens during failures, and the recovery actions needed during certain failures.

The NSX Advanced Load Balancer Controller is deployed as a three-node cluster for high availability and scalability. These nodes must be deployed such that the RTT (round-trip time) value between two Controller nodes is less than 20 milliseconds. In public cloud deployments such as AWS, a region has multiple availability zones. In such deployments, as a best practice, each Controller node should be deployed in a separate AZ.

Controller cluster deployment considerations are straightforward where the region has three or more AZs. However, in many disaster recovery (DR) deployments, there are only two AZ across which the three Controller cluster nodes are deployed, as depicted below:

Figure 5-1. Two Availability Zones within One Region



The Controller cluster will be `UP` in partition scenarios if at least two nodes are `UP` and connected. SEs will attempt to connect to the active partition. If they cannot connect to the active partition, they will operate without a Controller in a headless fashion and continue to serve application traffic.

In the above deployment, if AZ-2 goes `DOWN`, C1 will be brought down due to a lack of quorum. In such a scenario, manual intervention is needed to bring the Controller cluster `UP`.

At a high level, the manual workflow provides a way to recover the remaining node as a stand-alone cluster and permits two new nodes to be added when appropriate. The procedure is intentionally kept manual to force the user to recover the partitions carefully.

Recover a Non-Operational Cluster

Log in to Controller C1 and run the `/opt/avi/scripts/recover_cluster.py` script. It will reconfigure C1 as a standalone cluster, preserve the configuration and analytics data (logs and metrics), and bring the Controller cluster `UP`. Secure channel credentials for the other two Controller nodes will be revoked, and SEs having connectivity with C1 will be brought `UP`. As part of this, SEs will reconfigure themselves to connect only to C1.

What Happens When AZ-2 is Recovered

C2 and C3 will be able to form a Cluster. SEs in AZ-2 might connect to either partition (C1 or C2+C3). At this time, both the partitions are active, which could cause disruption if they run for too long. To prevent this, each Controller node monitors other Controller nodes in its configured members list. If a node identifies that it is not present in the configured members list of the other node, it will bring itself down for manual recovery. In this case, both C2 and C3 will detect that C1 has moved forward with manual recovery and will bring themselves down.

Secure channel credentials for all SEs will be revoked. SEs connecting to C2 or C3 during this time will detect that the Controller cluster is in manual recovery state; they will reboot themselves to no longer serve any application traffic. Once an SE comes `UP`, it will try to connect to C1 to establish normal operations. `REST API` commands sent to the Controllers, C2, and C3 will fail with a status code of `520`, indicating that these nodes must be reset to factory defaults using the `clean_cluster.py` script.

Clean a Non-Operational Cluster

It is assumed that C2 and C3 will be brought down soon after AZ-2 becomes operational. Log into C2 and C3 and run the script, `/opt/avi/scripts/clean_cluster.py`. This script will wipe out all the configuration and analytics and bring it up as a standalone node. Once C2 and C3 are cleaned, they can rejoin back to C1 to form a three-node cluster.

Impact of a Controller Failure

The impact of the loss of a Controller depends on whether the Controller is deployed as a standalone or as part of a three-node Controller cluster. A Controller failure can occur for various reasons, such as a software failure or failure in the underlying hardware. A Controller can also fail by losing connectivity to the networks containing the rest of the NSX Advanced Load Balancer infrastructure.

Controller Cluster

In a Controller cluster, the three Controllers divide the workload among themselves. If one Controller fails, the remaining two Controllers continue operations as normal with no impact to data plane traffic.

If the failed Controller was the cluster leader, one of the remaining Controllers takes over as cluster leader. This change in leadership has no direct impact on operations. If the Controller cluster IP address has been created, that IP address will move to the new leader, which will begin ARPing (advertising) the address.

Cluster Quorum

A Controller cluster requires at least two of its three nodes to be up, in order to maintain quorum. This eliminates the “split brain” scenario in which two devices are simultaneously active (primary) with potentially conflicting configuration updates.

If a Controller is still online, but has merely lost contact with the other Controllers, it will no longer be part of the cluster until it has regained connectivity. So if one Controller and multiple Service Engines are created in each of the three data centers, and one data center has lost connectivity to the other two data centers, the Controller at the isolated data center will not accept configuration changes as a standalone. The isolated Controller must either reconnect with the cluster or be formally demoted to a standalone.

Replacing a Failed Controller

If Controller failure is a permanent state, take the following actions to remove the Controller and restore full high availability to the cluster.

1 If the Controller was in a virtual machine, delete the VM from the cloud orchestrator.

2 From the web interface of another Controller that is still up, delete the IP address of the failed Controller. Navigate to **Administration > Controller**.

3 Install a new Controller.

Note Do not log into the web interface of the new Controller. Only perform initial setup such as selecting the cloud orchestrator.

4 From the existing Controller, add the IP address of the new Controller. After a few minutes, the status of the cluster must turn green.

Standalone Controller

In a standalone Controller configuration, a Controller failure leaves the NSX Advanced Load Balancer system in a headless state. In a headless state, existing Service Engines (SEs) will continue to operate with the last instructions they were given.

No new configuration changes are possible until a Controller is restored. This can be done by rebuilding a new Controller, which is disruptive to existing connections, or by bringing the existing Controller back online, which is transparent to existing data traffic.

While in a headless state, SEs will attempt to buffer client logs if sufficient disk space is available. If a Controller is temporarily offline, such as when the Controller's host is rebooted, the SEs will reconnect after the Controller returns, allowing the buffered client logs to be retrieved.

Recover a Non-Operational Controller Cluster

When two of the three NSX Advanced Load Balancer Controller nodes within a cluster are permanently down and not recoverable, the remaining Controller node in the cluster will be marked operationally down due to the lack of a cluster quorum.

Note All SEs continue to operate in headless mode.

Follow the steps below to return to a highly available three-node cluster:

1 To recover the cluster, you must first convert the remaining healthy Controller node to a single-node cluster configuration. Thereafter, two new nodes can be added to the cluster.

2 There are two ways of recovering a Controller, that is, with configuration and without configuration. It is important to recover one node with configuration to ensure it is made the Controller leader while other nodes are added as followers to the cluster:

- To recover a Controller with configuration, use the `/opt/avi/scripts/recover_cluster.py` script.

- To recover a Controller without configuration (essentially a factory reset; rarely necessary), use the `/opt/avi/scripts/clean_cluster.py` script instead. It is not reversible. The Controller will take a longer time to recreate the database. The`/opt/avi/scripts/clean_cluster.py` script performs the below tasks:
  - By default, this script reboots the connected SEs, unless the script is run with the switch.

    `/opt/avi/scripts/clean_cluster.py --skip-se-reboot`
  - The only way to login to the Controller node after running the script is to reset the admin password through the UI.

Typical Recovery

To convert the remaining Controller node to a single-node cluster while preserving the NSX Advanced Load Balancer configuration, execute the following script from the root account. If you attempt to execute it from a non-root account, the script will fail with a `Permission denied` message. Run `sudo` and enter the admin password to be promoted to root before running the script.

```
root@controller1:/home/admin# /opt/avi/scripts/recover_cluster.py
```

The script will request confirmation as a precaution and remind the user must run the script as root.

It is highly recommended to power off the other Controllers that were part of the cluster when running the `recover_cluster.py` script. Failure to do so can put the current and other nodes in an inoperable state.

The script stops all services on the Controller and restarts them. The Controller will be down and inaccessible for a few minutes.

Once the script finishes, you can log into the Controller node as a single-node cluster. To make this a highly available three-node cluster, add two new, unconfigured Controllers nodes to the cluster.

Note Ensure that the Controllers are on the same base and patch version.

API - Configuring the NSX Advanced Load Balancer Controller Cluster

The NSX Advanced Load Balancer REST API provides commands for managing the Controller cluster. This section discusses the steps to configure the Controller cluster using the API.

Getting the NSX Advanced Load Balancer Controller Cluster Configuration

To get the existing cluster configuration, send an API request as follows:

`GET /api/cluster`

The following is a sample output.

```
{
    url: "https://10.10.5.27/api/cluster",
    uuid: "cluster-005056ac9e91",
    name: "cluster-0-1",
    tenant_ref: "https://10.10.5.27/api/tenant/admin",
    virtual_ip: {
        type: "V4",
        addr: "10.10.5.27"
    },
    nodes: [
        {
            ip: {
                type: "V4",
                addr: "10.10.5.16"
            },
            vm_hostname: "node1.controller.local",
            vm_uuid: "005056ac9e91",
            name: "10.10.5.16",
        }
    ]
}
```

In this example, the cluster contains only a single member. When NSX Advanced Load Balancer is installed, the Controller created during installation is placed in a cluster as the primary Controller. The management IP address is configured as the cluster IP address.

Adding NSX Advanced Load Balancer Controller Nodes to the Cluster

To create a 3-node cluster, send a request such as the following to add two more Controller nodes to the cluster.

For `controller-ip`, specify the management IP address of the individual Controller node, not the IP address to be assigned to the cluster. The cluster IP is specified under `virtual_ip`.

`PUT /api/cluster`

The following is a PUT Data example.

```
{
    uuid: "cluster-005056ac9e91",
    name: "cluster-0-1",
    virtual_ip: {
        type: "V4",
        addr: "10.10.5.27"
    },
    nodes: [
        {
            ip: {
                type: "V4",
```

```
                addr: "10.10.5.16"
            },
            vm_hostname: "node1.controller.local",
            vm_uuid: "005056ac9e91",
            name: "10.10.5.16",
        },
        {
            ip: {
                type: "V4",
                addr: "10.10.5.15"
            },
            name: "10.10.5.15",
        },
        {
            ip: {
                type: "V4",
                addr: "10.10.5.17"
            },
            name: "10.10.5.17",
        }
    ]
}
```

Removing NSX Advanced Load Balancer Controller Nodes from the Cluster

To remove Controller nodes from the cluster, use a request such as the following.

```
PUT /api/cluster
```

The following is a PUT Data example.

```
{
    uuid: "cluster-005056ac9e91",
    name: "cluster-0-1",
    virtual_ip: {
        type: "V4",
        addr: "10.10.5.27"
    },
    nodes: [
        {
            ip: {
                type: "V4",
                addr: "10.10.5.16"
            },
            name: "10.10.5.16",
        }
    ]
}
```

Getting Runtime Information for the Cluster

The following request gets runtime information for the cluster.

The cluster is ready for operation when the cluster_state is `CLUSTER_UP_HA_ACTIVE` (for a 3-node cluster) or `CLUSTER_UP_NO_HA` (for a 1-node cluster).

This example shows `CLUSTER_UP_HA_ACTIVE`.

```
GET /api/cluster/runtime
```


```
{
    node_info: {
        uuid: "005056ac115e",
        mgmt_ip: "10.10.5.15",
        has_config: "True",
        ip: "node2.controller.local",
        vm_mor: "vm-54736",
        version: "16.1.1(9014) 2016-03-26 01:05:26 UTC",
        vm_uuid: "005056ac115e"
    },
    node_states: [
        {
            up_since: "2016-03-26 16:06:21",
            state: "CLUSTER_ACTIVE",
            role: "CLUSTER_LEADER",
            name: "10.10.5.15"
        },
        {
            up_since: "2016-03-26 16:07:08",
            state: "CLUSTER_ACTIVE",
            role: "CLUSTER_FOLLOWER",
            name: "10.10.5.17"
        },
        {
            up_since: "2016-03-26 16:07:31",
            state: "CLUSTER_ACTIVE",
            role: "CLUSTER_FOLLOWER",
            name: "10.10.5.16"
        }
    ],
    service_states: [],
        cluster_state: {
        up_since: "2016-03-26 16:06:21",
        progress: 100,
        state: "CLUSTER_UP_HA_ACTIVE"
    }
}
```


NSX Advanced Load Balancer ControllerCluster Configuration in AWS

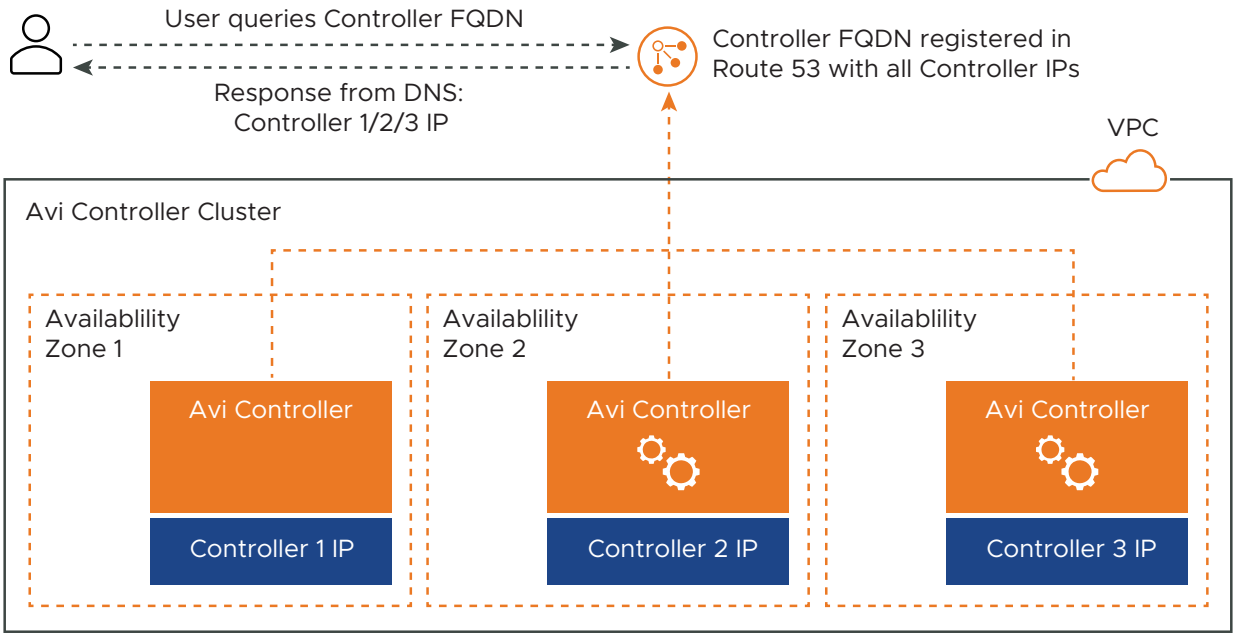
The NSX Advanced Load Balancer Controller is the management and orchestration engine for NSX Advanced Load Balancer ADC.

To provide HA and resilience, it is recommended to deploy a cluster of three Controller instances. Once the Controller cluster is formed, the controllers synchronize the state, irrespective of the Controller instance used to configure NSX Advanced Load Balancer features or retrieve operational data.

For more information about Controller cluster architecture, see High Availability for NSX Advanced Load Balancer Controllers.

In AWS environments, AWS Availability Zones (AZs) provide redundancy and separate fault domains. All AWS regions support a minimum of two AZs. To leverage the HA provided by AWS AZs, it is recommended to deploy different Controller instances of a cluster in different AZs.

Managing an NSX Advanced Load Balancer Controller Cluster across AZs



Each NSX Advanced Load Balancer Controller will receive an IP address from a different subnet given that an AWS subnet does not span across AZs.

In this scenario, it is recommended to create a FQDN in AWS Route 53, and associate all three Controller IPs with this FQDN. In addition, Route 53 health checks can be used in conjunction with multivalue routing when the FQDN is added to a public zone. This ensures that only healthy Controller IPs are returned.

Configure Azure Cluster IP

NSX Advanced Load Balancer Controller cluster IP configuration is integrated into the configuration workflow and no longer requires ControlScripts.

Procedure

1 Configure MSI based authentication for the Controller virtual machine.

   For more information on MSI based authentication, see *Support for Managed Services Identify (MSI) based Authentication for Microsoft Azure* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

2 Configure the Controller cluster IP. To add the cluster IP within the UI, navigate to **Administration > Controller > Nodes > Edit**. Enter the Name and the new address to the **Controller Cluster IP** field.

3 Click **Save**.

   Note The configured cluster IP must belong to the same VNet as the Controller nodes.

4 Remove a pre-configured cluster IP. If a cluster IP was configured using ControlScripts for a release prior to 17.2.5, follow the steps below to ensure that it is removed:

   a Remove the ControlScript, if configured.

   b Delete the cluster IP from the cluster IP configuration page.

   c Remove the cluster IP from the Azure portal.

Changing Controller IP Addresses in a Bare-metal Environment

Following are the steps to change the IP addresses of Controllers running in a bare-metal (Docker) environment.

Procedure

1 Change the IP address of each Controller node within the cluster to the new IP by manually editing the network-scripts on the host and changing the interface configuration.

2 Ensure that the new Controller IP addresses are reachable in the network from the other Controller nodes.

3 Edit the `/etc/systemd/system/avicontroller.service` or `/usr/sbin/avicontroller` file on each host running the Controller Docker container and overwrite the old Controller IP with the new Controller IP wherever applicable.

   The long command line below contains the Controller IP in a sample `/etc/systemd/system/avicontroller.service` file.

```
code>ExecStartPre=/usr/bin/docker run --name=avicontroller -m 48g --cpu-period=100000
--cpu-quota=2400000 -p 5098:5098 -p 8443:8443 -p 5054:5054 -p 80:80
-p 443:443 -p 161:161/udp -d --privileged -e "CONTAINER_NAME=avicontroller"
-e "MANAGEMENT_IP=10.122.0.112" -e NUM_CPU=24 -e NUM_MEMG=48 -e
```

```
DISK_GB=30 -e CNTRL_SSH_PORT=5098 -e SYSINT_PORT=8443 -e HTTP_PORT=80 -e
HTTPS_PORT=443 -v /:/hostroot/ -v /dev:/dev -v /var/run/docker.sock:/var/run/docker.sock
-v /run/xtables.lock:/run/xtables.lock -v /opt/avi/controller/data:/vol/ avinetworks/
controller:18.2.5-5116-20190708.063533</code>
```

4 On each Controller host the administrator would run this trio of commands:

```
systemctl daemon-relaod
systemctl stop avicontroller
systemctl start avicontroller
```

5 Corresponding to the sample in step 3, on any one of the Controller containers, the administrator would run these commands:

```
cd /opt/avi/python/bin
python change_ip.py -i 10.122.0.111 -o 10.122.0.112 -o 10.122.0.113
```

Results

The Controller cluster comes back up with the new IPs.

OpenStack Network Configuration for NSX Advanced Load Balancer Controller Cluster

This section explains how to configure a cluster in NSX Advanced Load Balancer for an OpenStack cloud. To provide NSX Advanced Load Balancer Controller HA, add two additional Controller nodes to create a three-node Controller cluster.

For more information on deploying a cluster, see Chapter 5 NSX Advanced Load Balancer Controller Cluster.

Prerequisites for Cluster Deployment

There are certain prerequisites defined for the leader and follower nodes in a cluster. For complete information, see Prerequisites for Cluster Deployment.

From an OpenStack perspective, consider the following:

1 A Neutron port is created and is available for cluster VIP.

2 A floating IP is available for Neutron port.

Deploying an NSX Advanced Load Balancer Controller Cluster

For complete information on configuring Controller's management interfaces and cluster IP, see Deploying an NSX Advanced Load Balancer Controller Cluster (Additional).

The following sections are for creating OpenStack floating IP and binding that with the cluster IP:

Write Mode

1 Access OpenStack Horizon CLI.

a List the Network

`openstack network list` — This indicates the configured requisite networks.

```
root@openstack-mitaka:/root# openstack network list
+--------------------------------------+---------------
+------------------------------------------------------+
| id                                   | name
| subnets                                              |
+--------------------------------------+---------------
+------------------------------------------------------+
| 10a514a3-d843-499d-80fd-28274d4a4912 | webserver-
net | 3ebfb2ef-9b47-44f7-9da5-5245e1d0ed53 192.168.10.0/24 |
| 5dd0b1cb-ebba-4ff9-84fd-74dcf13c7f86 | client-
net    | a9a00d61-6ee8-4fac-80df-4e0bb8c8b4f3 192.168.11.0/24 |
| c1c045f5-2d0f-43e3-ab43-55f990cde9b7 | provider1
| 1b65c0da-38c7-4c85-88a9-30c52c6a4558 10.130.128.0/18 |
| dd9dab27-9228-4765-96f2-d56194136ba0 | avimgmt
| 5785c1cf-a222-4b0a-9343-003153f37a65 172.16.0.0/24   |
+--------------------------------------+---------------
+------------------------------------------------------+
```

b Create a floating IP.

`openstack floating ip create provider1` — `provider1` is the network used.

```
root@openstack-mitaka:/root# openstack floating ip create provider1
```

New floating IP is created.

```
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| description         |                                      |
| fixed_ip_address    |                                      |
| floating_ip_address | 10.130.170.86                        |
| floating_network_id | c1c045f5-2d0f-43e3-ab43-55f990cde9b7 |
| id                  | 4ec57a12-7357-461a-80f6-d87ae7536335 |
| port_id             |                                      |
| router_id           |                                      |
| status              | DOWN                                 |
| tenant_id           | 904fb201a92f443297bffca3b354d52d     |
+---------------------+--------------------------------------+
```

c Get the port-id for cluster IP.

```
 openstack port list -c ID -c 'Fixed IP Addresses'|grep 172.16.0.65
     95665123-64a4-453a-abde-70fdb3d2ae2a| ip_address='172.16.0.65',
subnet_id='5785c1cf-a222-4b0a-9343-003153f37a65'
```

d Associate the cluster IP with the floating IP.

Using the port-id from the command above (`95665123-64a4-453a-abde-70fdb3d2ae2a` in this case), associate it with the floating IP created in step b.

```
root@openstack-mitaka:/root# openstack floating ip set --port 95665123-64a4-453a-
abde-70fdb3d2ae2a 4ec57a12-7357-461a-80f6-d87ae7536335
```

```
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| description               |                                      |
| fixed_ip_address          | 172.16.0.65                          |
| floating_ip_address       | 10.130.170.86                        |
| floating_network_id       | c1c045f5-2d0f-43e3-ab43-55f990cde9b7|
| id                        | 4ec57a12-7357-461a-80f6-d87ae7536335|
| port_id                   | 95665123-64a4-453a-abde-70fdb3d2ae2a|
| router_id                 | 2d3b93a2-7804-4841-90c4-be15b148d099|
| status                    | ACTIVE                               |
| tenant_id                 | 904fb201a92f443297bffca3b354d52d     |
+---------------------------+--------------------------------------+
```

2 Add the cluster IP and the secondary IP for the cluster leader.

```
root@172-16-0-66:~# ip a
eth0: (BROADCAST,MULTICAST,UP,LOWER_UP) mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:50:56:bd:5a:0f brd ff:ff:ff:ff:ff:ff
inet 172.16.0.66/24 brd 172.16.0.255 scope global eth0
valid_lft forever preferred_lft forever
inet 172.16.0.65/32 scope global eth0:1 Cluster IP
```

No-Access Mode

For OpenStack No-Access cloud type, the AAP entries need to be configured manually using the following command.

An example is shown in the code block below.

```
root@openstack-mitaka:/root# openstack port set --allowed--address ip-address=172.16.0.133
Controller_Port
```

```
root@openstack-mitaka:/root#  openstack port set --allowed--address ip-address=172.16.0.133
d0bf0bda-02e2-46bf-abd2-0d05cc4654df
root@openstack-mitaka:/root# openstack port show d0bf0bda-02e2-46bf-abd2-0d05cc4654df
        +-------------------------------
+-------------------------------------------------------------------------------------+
 | Field                     |
Value                                                                                 |
 +---------------------------
+-------------------------------------------------------------------------------------+
 | admin_state_up            |
True                                                                                  |
 | allowed_address_pairs     |   {"ip_address": "172.16.0.131", "mac_address":
```

```
"fa:16:3e:47:6b:70"}                    |
 | binding:host_id          |   openstack-
mitaka                                                                  |
 | binding:profile          |
{}                                                                                |
 | binding:vif_details      |   {"port_filter":
true}                                                          |
 | binding:vif_type         |
bridge                                                                            |
 | binding:vnic_type        |
normal                                                                            |
 | created_at               |
2018-01-12T13:58:02                                                               |
 | description
|                                                                                     |
 | device_id                |   2adedfc3-75d6-4296-ad18-
bfc38873485c                                          |
 | device_owner             |
compute:nova                                                                      |
 | extra_dhcp_opts
|                                                                                     |
 | fixed_ips                |   {"subnet_id": "5785c1cf-a222-4b0a-9343-003153f37a65",
"ip_address": "172.16.0.133"}  |
 | id                       |   d0bf0bda-02e2-46bf-
abd2-0d05cc4654df                                             |
 | mac_address              |
fa:16:3e:47:6b:70                                                                 |
 | name
|                                                                                     |
 | network_id               |   dd9dab27-9228-4765-96f2-
d56194136ba0                                          |
 | port_security_enabled    |
True                                                                              |
 | security_groups          |   3cc1092e-538c-4ff7-b4ac-
eeff84731f75                                          |
 | status                   |
ACTIVE                                                                            |
 | tenant_id                |
904fb201a92f443297bffca3b354d52d                                                  |
 | updated_at               |
2018-01-12T14:19:06                                                               |
 +--------------------------
+-----------------------------------------------------------------------------------------+
```

Create the neutron port for the VIP by using the following command.

```
openstack port create --network "neutron_network_name" --
allowed-address mac-address="fa:16:3e:52:81:03",ip-address="172.16.0.63" --allowed-
address mac-address="fa:16:3e:52:81:04",ip-address="172.16.0.64" --allowed-address mac-
address="fa:16:3e:52:81:06",ip-address="172.16.0.66" --fixed-ip ip-address="172.16.0.65" --
project "904fb201a92f443297bffca3b354d52d"
```

The following is an example.

```
openstack port create --network "neutron_network_name" --
allowed-address mac-address="controller_mac1",ip-address="controller_ip1" --allowed-
address mac-address="controller_mac2",ip-address="controller_ip2" --allowed-address mac-
address="controller_mac3",ip-address="controller_ip3" --fixed-ip ip-address="cluster_ip" --
project "project-id"
```

Note When the leader Controller fails (or reboots), a follower Controller will take over the cluster IP (in this case, 172.16.0.65), and the mapping between floating IP (10.130.170.86) and cluster IP (172.16.0.65) will not change. Therefore without intervention, the floating IP and cluster IP association will work as expected.

FAQs on Controller Cluster

This topic answers basic questions about NSX Advanced Load Balancer Controller Clusters.

- How many nodes are supported in an NSX Advanced Load Balancer Controller cluster?

  NSX Advanced Load Balancer Controller clusters can include either 1 (standalone mode) or 3 Controller nodes in a cluster.

- How many nodes are needed for the NSX Advanced Load Balancer Controller cluster to be operational?

  A three-node cluster requires a quorum (majority) of the nodes to be present for the cluster to be operational. It means at least two nodes must be up.

- Can you explain how the three nodes in an NSX Advanced Load Balancer Controller are used during normal operation?

  Among the three nodes, a leader node is elected and orchestrates the process startup or shutdown across all the active members of the cluster. Configuration and metrics databases are configured to be active on the leader node and placed in standby mode on the follower nodes.

  Streaming replication is configured to synchronize the active database to the standby databases. Analytics functionality is shared among all the nodes of the cluster. A given virtual service streams the logs and metrics to one of the nodes of the cluster.

- What happens if a follower node goes down?

  This node is removed from the active member list. The work that was performed on this node is re-distributed among the remaining nodes in the NSX Advanced Load Balancer Controller cluster.

- What happens if the leader node goes down?

One of the follower nodes is promoted as a leader. This triggers a warm restart of the processes among the remaining nodes in the cluster. The warm standby is required to promote the standby configuration and metrics databases from standby to active on the new leader.

Note During warm restart, the Controller REST API is unavailable for a period of 2-3 minutes.

- What happens if two nodes go down?

  The entire cluster becomes non-operational until at least one of the nodes up. A quorum (two out of three) of the nodes must be up for the cluster to be operational.

- Will there be a disruption to traffic during cluster convergence (single or multiple nodes go down and come back)?

  When single or multiple nodes go down and come back, the cluster is non-operational, the SEs continue to forward traffic for the configured virtual services. This is referred to as headless mode.

  The analytics (logs and metrics) are buffered on the SEs. When the Controller cluster is once again operational, the cluster re-synchronizes with the SEs and initiates the collection of metrics and logs. Data plane traffic continues to flow normally throughout this time.

- Recover a non-operational cluster where two of the three nodes are permanently down and not recoverable

  For detailed information, see Recover a Non-Operational Controller Cluster.

- Recover the system if all three NSX Advanced Load Balancer Controller nodes are permanently down and not recoverable.

  For detailed information, see Backup and Restore of NSX Advanced Load Balancer Configuration.

- Re-configure the Controller Cluster membership.

  For detailed information, see Changing NSX Advanced Load Balancer Controller Cluster Configuration.

- Can the NSX Advanced Load Balancer Controller nodes in a cluster have different resource allocations (memory, CPU, and drive space)?

  It is recommended to allocate these same resources to each of the three nodes within the Controller cluster. In course of time, if there is a need to re-size (change resource allocations) the NSX Advanced Load Balancer Controller in the cluster, to accommodate growth, change resource allocations on one NSX Advanced Load Balancer Controller node at a time. However, it is expected that all nodes within the cluster will eventually be resized to the same allocations.

- Can the NSX Advanced Load Balancer Controllers in a cluster be in different subnets?

  Yes. This configuration is supported and can even be preferred for certain topologies for improved fault tolerance. However, a limitation with placing the Controllers in separate subnets is that the cluster IP address is not supported. The cluster IP address requires all Controller nodes to be in the same subnet.

- Can the cluster leader be changed manually?

  No. Currently, this is not a supported operation. The leader is chosen during initial deployment of the cluster or when recovering a fully down cluster. However, the leader cannot be manually changed while the cluster is operational.

  NSX Advanced Load Balancer Controller Controllers participate in election of their leader, and automatically decides who the new leader must be in case of failure.

- What timers are used during cluster failover? Are the cluster failover timers configurable?

  Leader failure

  If the leader Controller of the cluster fails, this triggers an internal warm restart of the Controller processes. Typically, this takes around 2-3 minutes after it is detected that the leader has failed.

  Graceful reboot of leader

  Failure detection is almost instantaneous in cases of a graceful reboot.

  Ungraceful power-off of leader

  In the case of an ungraceful power-off of the leader, it can take up to 30 seconds to detect that the leader has failed.

  These timers are tuned based on testing but are not configurable.

- Can configuration changes be made directly on follower Controllers?

  Yes. Nearly any configuration changes can be made on any of the Controller nodes.

  Exceptions:

  Configuration of the cluster itself (node IP addresses and cluster address).

  Upgrade, which is performed only on the leader node.

- What is the recommended cluster deployment option for multiple data centers in different regions?

  NSX Advanced Load Balancer recommends deploying the Controller Cluster only within a single region. However, the member nodes are deployed across multiple availability zones within the region. If there are multiple regions, it is recommended to deploy one Controller Cluster per region. (For more information, see Clustering NSX Advanced Load Balancer Controllers of Different Networks).

- How to import and export NSX Advanced Load Balancer metrics database?

  Prior to upgrading to version 22.1.2, it is recommended to export the metrics database. To export the metrics database, copy the `/var/lib/postgresql/10/pg_metrics/` directory and store it outside the Controller VM. In case you need to roll back to a previous release, the exported metrics can be imported. Thus, preventing the loss of data.

  To import the metrics database,

  - Stop the process supervisor by running the following command first on the follower nodes and then on the leader.

    ```
    systemctl stop process-supervisor
    ```

  - Move the copied `pg_metrics` directory to /`var/lib/postgresql/10/pg_metrics/` on the leader.
  - Run the following to first start the process supervisor on leader and then on follower so that the imported `pg_metrics` data is replicated from leader to the followers.

    ```
    python3 /opt/avi/scripts/pg_rec_fix_cluster.py --nodelist leader_ip follower_1_ip
    follower_2_ip
    ```

  Note It will take some time for the cluster to become HA_ACTIVE based on the metrics database size.

FAQs on Database Replication

This section discusses the FAQs and resolutions on Database Replication.

What are the observations leader node disconnects from the NSX Advanced Load Balancer cluster?

Following are the observations when a leader node disconnects from the NSX Advanced Load Balancer cluster:

1. The leader node restarts and breaks the NSX Advanced Load Balancer cluster due to a lost quorum.
2. One of the follower nodes posts a message that the DB replication is not completed and it cannot become a leader due to that.
3. Another follower node will post `DB replication from the leader` and becomes the leader node.

What will happen when none of the follower is not able to complete database replication in an NSX Advanced Load Balancer cluster?

1. On the follower node, the following process is used to replicate the database:

Check if streaming replication of the database can be enabled. In general, it is expected to succeed as NSX Advanced Load Balancer allows for a reasonable number of WAL record files to satisfy this. If this fails or if it is the first time a follower node is added to the cluster, the follower node does a full backup, which usually takes time.

2 Once the streaming replication is set up, NSX Advanced Load Balancer monitors that the replication continues to occur every minute. If it detects a failure five consecutive times, then it declares that the replication has failed and attempt a full replication.

3 On trying to do a full replication, NSX Advanced Load Balancer always does a full copy of the entire database data directory into the next directory. Only when it is complete it moves this to the current directory for the database. If there is a failure as a part of the full copy of the database files, then it will continue to have a valid, current directory.

4 On deciding to do a full backup, it is essential to record this to ensure that this node does not become a leader if there is a leader failure during this time. With this scheme, a follower node must always be in sync and can take over as a leader. If the full sync fails, the safety mechanism will prevent the node from taking over as leader, but with manual intervention, you can promote one of the nodes as the leader.

Does the cluster_mgr logs show the sync up every minute once the full replication attempts?

Logs for the full replication process can be found in `/var/lib/avi/log/postgres_service_main.log` and `postgres_service_metrics.log`.

What are the common causes of replication failure?

Permanent connectivity issues can cause replication failures. Streaming replication always succeeds when the postgres on the follower node can recover from its checkpoint and is behind the leader. In some instances, when the postgres is not brought down gracefully, it may not be able to replicate from leader. This will force us to do a full backup from the leader.

Which directory is the ‘current’ directory and which directory is the ‘next’ directory during the database replication process?

For config database: next is `/var/lib/postgresql/9.3/main_backup` and curl is `/var/lib/postgresql/9.3/main/`.

For metrics database: next is `/var/lib/postgresql/9.3/pg_metrics/metrics_backup` and `/var/lib/postgresql/9.3/pg_metrics/metrics`.

What are the symptoms of replication failures? Are they only noticeable during leader election? Are internal events generated for such events?

DB REPLICATION FAILED events are observed on the NSX Advanced Load Balancer, when it detects replication failures.

Is there any relation to cluster convergence? If so, what are the recent improvements on this in detail?

When NSX Advanced Load Balancer detects replication failure, a full backup is triggered. As a part of this, it touches a file in `/var/lib/avi/etc` to indicate that the replication has not been completed. So, the specific node does not participate in becoming a leader if there is a convergence. Also, the state for this service goes to COPYING_DB_FROM_LEADER, which will move the node state from CLUSTER_ACTIVE to CLUSTER_STARTING. The overall cluster state is not observed as CLUSTER_HA_ACTIVE.

Does the recent versions of NSX Advanced Load Balancer still use zookeeper ? If not, why does it still run on NSX Advanced Load Balancer and what has changed?

The recent versions of the NSX Advanced Load Balancer do not use zookeeper. The zookeeper process is run for backward compatibility. When NSX Advanced Load Balancer is upgraded from previous versions to 17.2, the Controllers will have to publish the leadership information in ZK for the SEs that are not yet in 17.2. Once the upgrade is complete, SE will use the new scheme for leadership notifications. At some point, ZK will be removed when all upgrades will be from 17.2.x.

Tenant Settings

6

A tenant is an isolated instance of NSX Advanced Load Balancer. Each NSX Advanced Load Balancer user account is associated with one or more tenants.

The tenant associated with a user account defines the resources that the user can access within the NSX Advanced Load Balancer. When a user logs in, the NSX Advanced Load Balancer restricts access to only those resources that are in the same tenant. If a user is associated with multiple tenants, each tenant still isolates the resources that belong to that tenant from the resources in other tenants. To access resources in another tenant, the user must switch the focus of the management session to the other tenant.

Note

- For information on switching a management session from one tenant to another, see Switching Between Tenants.
- For information on creating a cloud inside a tenant, see Tenant-Scoped Clouds.
- Certificates from the admin tenant can be shared by non-admin tenants. For more information, see Sharing Certificates Across Tenants.

Default Tenant

By default, all resources belong to a single, global tenant, namely, admin. The admin tenant contains all NSX Advanced Load Balancer resources.

The default admin user account belongs to the admin tenant and so, can access all resources.

If no additional tenants are created, all new NSX Advanced Load Balancer user accounts are automatically added to the admin tenant.

Tenant-to-Role Mapping

Within each user account, the role selected for the user is mapped to a tenant. If only one tenant is defined (the default admin tenant), this tenant is automatically mapped to the role selected for the user. It allows the user to access all resources to the extent (write, read, or no access) allowed by their role.

Creating additional tenants allows a user account to have multiple roles. In this case, each role can be mapped individually to a tenant within the user account. Optionally, a single role can be mapped to all tenants.

If a single role is mapped to all tenants, the user can switch the management session to other tenants as needed.

All Tenants View for Super Users

The NSX Advanced Load Balancer user accounts that are enabled for superuser access automatically have access to a special read-only tenant: All Tenants. The All Tenants view provides read-only access to all resources within the NSX Advanced Load Balancer.

The All Tenants View tenant cannot be mapped to any roles within a user account. The All Tenants tenant is automatically made available to all superuser accounts.

Read the following topics next:

- Create a Tenant
- Add an Existing User to a Tenant
- Enable Use Single Role for All Tenants
- Configuring Tenant Settings
- Switching Between Tenants
- Sharing Certificates Across Tenants
- Tenant-Scoped Clouds
- All Tenants View
- Tenants Versus SE Group Isolation
- Sharing Admin Profiles Across Tenants

Create a Tenant

This task helps you to create a tenant.

Procedure

1. Navigate to **Administration > Accounts > Tenants**, and click **Create**.
2. Enter a name for the new tenant.
3. Enter an optional description.
4. By default, **Tenant Access to Provider Service Engine** is selected.

5 Select **Tenant VRF** if required.

When **Per Tenant IP Domain** is selected, each tenant gets its own routing domain that is not shared with any other tenant. When **Share IP Domain across all tenants** is selected, all tenants share the same routing domain.

6 Click **Save**.

The new tenant appears on the tenant table.

Note The admin account is automatically added to each new tenant.

Add an Existing User to a Tenant

This task helps you to add an existing user to a tenant.

Procedure

1 Navigate to **Administration > Accounts > Users**.

2 Click the edit icon next to the user name.

3 Under **Tenant & Role**, click **Add** and select the role from the **Role** drop-down menu.

4 Click **Add** and select the new tenant.

a Click **Create** if the new tenant does not exist.

A new set of mapping fields appears.

5 Select the mapping that is no longer needed and click **REMOVE**.

6 Click **Save**.

Results

The tenants table reappears, displaying the change.

Enable Use Single Role for All Tenants

This option allows the user account to access any tenant from the same role. In this case, the default tenant must be selected. The default tenant is the one into which the user is placed after logging in.

Configuring Tenant Settings

A tenant is a logical grouping of configuration. An administrator might belong to one or multiple tenants, but can only view and edit one tenant at a time. Each tenant has its own isolated control plane configuration.

Each tenant can optionally have its own isolated data plane. This will depend on the global configuration of the NSX Advanced Load Balancer deployment.

Tenants can be deployed within a Provider Context or a Tenant Context:

- Provider Context mode: SE groups are shared across tenants
- Tenant Context mode: SE groups are exclusive to each tenant

In NSX Advanced Load Balancer, Virtual Routing Forwarding (VRF) contexts can be created to isolate traffic within a system.

Using the UI

This task helps you to configure the tenant settings using UI.

Procedure

1 Navigate to **Administration > System Settings > EDIT > Tenancy Mode**.

2 Select **IP Route Domain** as **Per Tenant** or **Share Across Tenants** as required.

When **Per Tenant IP Domain** is selected, each tenant gets its own routing domain that is not shared with any other tenant. When **Share IP Domain across all tenants** is selected, all tenants share the same routing domain.

3 Select the context mode as applicable:

- **Service Engines are managed within the tenant context, not shared across tenants** to enable the **Tenant Context** mode.
- **Service Engines are managed within the provider context, shared across tenants** to enable the **Provider Context** mode.

4 Select **Read** or **No Access** as required in the **Tenant Service Engine Access** field.

5 Click **Save**.

Note The configuration made using the UI will remain as the global configuration when creating a tenant. These values can be overridden for a specific tenant using the CLI or API.

Using the CLI

Log in to the shell and enter the configuration as shown.

```
[admin:10-140-16-178]: > configure tenant t2

[admin:10-140-16-178]: tenant> config_settings
[admin:10-140-16-178]: tenant:config_settings> no se_in_provider_context
+-------------------------+-------+
| Field                   | Value |
+-------------------------+-------+
| se_in_provider_context | False |
+-------------------------+-------+
sav[admin:10-140-16-178]: tenant:config_settings> save
[admin:10-140-16-178]: tenant> save
+--------------------------------+------------------------------------------------+
| Field                          | Value                                          |
+--------------------------------+------------------------------------------------+
| uuid                           | tenant-7136c2d8-f903-4b3b-a2bb-23336aa0149f |
| name                           | t2                                             |
| local                          | True                                           |
| config_settings                |                                                |
|   tenant_vrf                   | False                                          |
|   se_in_provider_context       | False                                          |
|   tenant_access_to_provider_se | True                                           |
+--------------------------------+------------------------------------------------+
```

Field	Value	Description
se_in_provider_context	True	SEs are managed within the provider context, shared across tenants
se_in_provider_context	False	SEs are managed within the tenant context, not shared across tenants
tenant_access_to_provider_se	True	Tenant has read access to SEs
tenant_access_to_provider_se	False	Tenant has no access to SEs
tenant_vrf	True	Configure IP route domain per tenant. The system creates a unique VRF for each tenant that is only usable within that tenant
tenant_vrf	False	Share IP route domain across tenants. VRFs are defined in the admin tenant and are usable by any tenant

Note In the global configuration, `se_in_provider_context` is set to `True`. However, this can be changed for a tenant.

Switching Between Tenants

User accounts that have access to multiple tenants are placed into a single, default tenant after log in.

Procedure

1 In the UI, navigate to **Applications > Dashboard**.

2 Click the username in the upper right corner of the display, and select **All Tenants** from the drop-down menu.

The resources of newly selected tenant are now accessible through the web interface instead of the previously selected tenant.

Results

Note If you are logged in with a superuser account, a read-only All Tenants View tenant is also available.

Note When you switch the tenant, the corresponding Application dashboard opens.

Sharing Certificates Across Tenants

In NSX Advanced Load Balancer, certificates from the admin tenant can be shared by non-admin tenants when the `shared_ssl_certificates` flag is set to `True` in the Controller.

Default Behavior

System default certificates are used by objects in any tenant. For example, these include `System-Default-Cert`, `System-Default-Cert-EC`, `System-Default-Portal-Cert`, `System-Default-Portal-Cert-EC256`, `System-Default-Root-CA`, and `System-Default-Secure-Channel-Cert`, a set of objects that can be expected to expand over time. Objects created in a specific tenant (including the admin tenant) can only be viewed and used in their respective tenant. Certificates are automatically chained and will only be chained to certificates in the respective tenant.

Shared SSL Certificates

In NSX Advanced Load Balancer, the `shared_ssl_certificates` are added to the `Controller Properties` object. By default, this is set to `False`. If `shared_ssl_certificates` is set to `True`, the following behavior applies:

- All certificates from the admin tenant are viewed from non-admin tenants.
- Certificates from the admin tenant can be used in non-admin objects, that is, virtual services, pools, and so on.
- Application certificates in non-admin tenants will be chained to issuer certificates in the admin tenant.
- NSX Advanced Load Balancer will not chain certificates from the admin tenant to issuer certificates in non-admin tenants. As a result, if there is an Intermediate certificate in the admin tenant and the corresponding CA certificate is in the non-admin tenant, these objects will not be linked.
- If there are any cross-tenant links (that is, an Intermediate certificate in the admin tenant and Application certificate in the non-admin tenant), the NSX Advanced Load Balancer will prevent changing the `shared_ssl_certificates` flag.

- For unchained Application certificate in a non-admin tenant and the corresponding Intermediate certificate in the admin tenant, the user toggles the `shared_ssl_certificates` flag from `False` to `True`, and the Intermediate and Application certificates will not be chained. If you want these certificates to be chained, delete and recreate the application certificate.
- You can configure this feature using NSX Advanced Load Balancer REST API or CLI. This feature is currently not supported on the NSX Advanced Load Balancer UI.

Note

- When certificate sharing is enabled in NSX Advanced Load Balancer before version 21.1.4, the certificate with the most days to expiry is always selected.
- When certificate sharing is enabled in NSX Advanced Load Balancer version 21.1.4, the Intermediate or CA certificate with the highest expiry in the current tenancy is always selected. If the current tenant has no Intermediate or CA, the corresponding Intermediate or CA from the admin tenant (if any) is selected.

Usage Guidelines

The following guidelines are applicable as the certificates in the admin tenant can be chained to any certificate in the system:

- Toggle the `shared_ssl_certificates` flag to `True` and create shared Intermediate or Root certificates in the admin tenant before creating Application certificates.
- Application certificates must be in the tenant with the corresponding application.
- Although certificate additions or updates in the admin tenant are CPU-intensive, these actions must have minimal impact, as they are infrequent operations.

CLI Configuration

```
[admin:10-10-28-16]: > configure controller
properties

Updating an existing object. Currently, the object is:
+-----------------------------------------+--------------------+
| Field                                   | Value              |
+-----------------------------------------+--------------------+
| uuid                                    | global             |
| unresponsive_se_reboot                  | 300 sec            |
| crashed_se_reboot                       | 900 sec            |
| se_offline_del                          | 172000 sec         |
| vs_se_create_fail                       | 1500 sec           |
| vs_se_vnic_fail                         | 300 sec            |
| vs_se_bootup_fail                       | 480 sec            |
| se_vnic_cooldown                        | 120 sec            |
| vs_se_vnic_ip_fail                      | 120 sec            |
| fatal_error_lease_time                  | 120 sec            |
| upgrade_lease_time                      | 360 sec            |
| query_host_fail                         | 180 sec            |
```

```
| vnic_op_fail_time                           | 180 sec            |
| dns_refresh_period                          | 60 min             |
| se_create_timeout                           | 900 sec            |
| max_dead_se_in_grp                          | 1                  |
| dead_se_detection_timer                     | 360 sec            |
| api_idle_timeout                            | 15 min             |
| allow_unauthenticated_nodes                 | False              |
| cluster_ip_gratuitous_arp_period            | 60 min             |
| vs_key_rotate_period                        | 360 min            |
| secure_channel_controller_token_timeout     | 60 min             |
| secure_channel_se_token_timeout             | 60 min             |
| max_seq_vnic_failures                       | 3                  |
| vs_awaiting_se_timeout                      | 60 sec             |
| vs_apic_scaleout_timeout                    | 360 sec            |
| secure_channel_cleanup_timeout              | 60 min             |
| attach_ip_retry_interval                    | 360 sec            |
| attach_ip_retry_limit                       | 4                  |
| persistence_key_rotate_period               | 0 min              |
| allow_unauthenticated_apis                  | False              |
| warmstart_se_reconnect_wait_time            | 480 sec            |
| vs_se_ping_fail                             | 60 sec             |
| se_failover_attempt_interval                | 300 sec            |
| max_pcap_per_tenant                         | 4                  |
| ssl_certificate_expiry_warning_days[1]      | 30 days days       |
| ssl_certificate_expiry_warning_days[2]      | 7 days days        |
| ssl_certificate_expiry_warning_days[3]      | 1 days days        |
| seupgrade_fabric_pool_size                  | 20                 |
| seupgrade_segroup_min_dead_timeout          | 360 sec            |
| allow_ip_forwarding                         | False              |
| appviewx_compat_mode                        | False              |
| upgrade_dns_ttl                             | 5 sec              |
| bm_use_ansible                              | True               |
| vs_se_attach_ip_fail                        | 600 sec            |
| max_seq_attach_ip_failures                  | 3                  |
| cleanup_expired_authtoken_timeout_period    | 60 min             |
| cleanup_sessions_timeout_period             | 60 min             |
| consistency_check_timeout_period            | 60 min             |
| process_locked_useraccounts_timeout_period | 1 min              |
| process_pki_profile_timeout_period          | 1440 min           |
| enable_memory_balancer                      | True               |
| warmstart_vs_resync_wait_time               | 300 sec            |
| api_perf_logging_threshold                  | 10000 milliseconds |
| se_from_marketplace                         | IMAGE              |
| cloud_reconcile                             | True               |
| enable_api_sharding                         | True               |
| vs_scaleout_ready_check_interval            | 60 sec             |
| shared_ssl_certificates                     | False              |
+---------------------------------------------+--------------------+
[admin:10-10-28-16]: controllerproperties> shared_ssl_certificates
Overwriting the previously entered value for shared_ssl_certificates
[admin:10-10-28-16]: controllerproperties> save
+---------------------------------------------+--------------------+
| Field                                       | Value              |
+---------------------------------------------+--------------------+
| uuid                                        | global             |
```

unresponsive_se_reboot	300 sec
crashed_se_reboot	900 sec
se_offline_del	172000 sec
vs_se_create_fail	1500 sec
vs_se_vnic_fail	300 sec
vs_se_bootup_fail	480 sec
se_vnic_cooldown	120 sec
vs_se_vnic_ip_fail	120 sec
fatal_error_lease_time	120 sec
upgrade_lease_time	360 sec
query_host_fail	180 sec
vnic_op_fail_time	180 sec
dns_refresh_period	60 min
se_create_timeout	900 sec
max_dead_se_in_grp	1
dead_se_detection_timer	360 sec
api_idle_timeout	15 min
allow_unauthenticated_nodes	False
cluster_ip_gratuitous_arp_period	60 min
vs_key_rotate_period	360 min
secure_channel_controller_token_timeout	60 min
secure_channel_se_token_timeout	60 min
max_seq_vnic_failures	3
vs_awaiting_se_timeout	60 sec
vs_apic_scaleout_timeout	360 sec
secure_channel_cleanup_timeout	60 min
attach_ip_retry_interval	360 sec
attach_ip_retry_limit	4
persistence_key_rotate_period	0 min
allow_unauthenticated_apis	False
warmstart_se_reconnect_wait_time	480 sec
vs_se_ping_fail	60 sec
se_failover_attempt_interval	300 sec
max_pcap_per_tenant	4
ssl_certificate_expiry_warning_days[1]	30 days days
ssl_certificate_expiry_warning_days[2]	7 days days
ssl_certificate_expiry_warning_days[3]	1 days days
seupgrade_fabric_pool_size	20
seupgrade_segroup_min_dead_timeout	360 sec
allow_ip_forwarding	False
appviewx_compat_mode	False
upgrade_dns_ttl	5 sec
bm_use_ansible	True
vs_se_attach_ip_fail	600 sec
max_seq_attach_ip_failures	3
cleanup_expired_authtoken_timeout_period	60 min
cleanup_sessions_timeout_period	60 min
consistency_check_timeout_period	60 min
process_locked_useraccounts_timeout_period	1 min
process_pki_profile_timeout_period	1440 min
enable_memory_balancer	True
warmstart_vs_resync_wait_time	300 sec
api_perf_logging_threshold	10000 milliseconds
se_from_marketplace	IMAGE
cloud_reconcile	True

```
| enable_api_sharding                         | True               |
| vs_scaleout_ready_check_interval            | 60 sec             |
| shared_ssl_certificates                     | True               |
+---------------------------------------------+--------------------+
[admin:10-10-28-16]: > configure sslkeyandcertificate admin-intermediate

[admin:10-10-28-16]: sslkeyandcertificate> certificate
[admin:10-10-28-16]: sslkeyandcertificate:certificate> certificate --
-----BEGIN
CERTIFICATE-----

[280/18075]
MIIFZzCCA0+gAwIBAgICEAIwDQYJKoZIhvcNAQELBQAwPzELMAkGA1UEBhMCVVMx
CzAJBgNVBAgMAkNBMQwwCgYDVQQKDANBdmkxFTATBgNVBAMMDEludGVybWVkaWF0
ZTAeFw0xNzEyMjAyMzM0MzVaFw0zNzEyMTUyMzM0MzVaMEkxCzAJBgNVBAYTAlVT
MQswCQYDVQQIDAJDQTEMMAoGA1UECgwDQXZpMR8wHQYDVQQDDBZTYW1lLU5hbWUt
SW50ZXJtZWRpYXRlMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAy0kq
S48Ngxg1KJ1hmwMxbSEJnGuz0bfxf/FbcVK0OQZzOfl7K1nrg8CIjLyywEkgzBqf
/b1GwEwNRNvCxAgIP78kCw39chdGzW2jRcjiWPV6OrOizrkXHKlhCJ7LnONSeQH1
rGehFSzpLT8g6KY+DCkeVQBVscV4cFJFTL484EoOhgxMuqj0jij3T+GctqsK5p2Y
VCy71ZEbJvvET3x6/rDNIJU9njJxCvlJyk3T78sTSsW7+xjhCRVsvBAHyUhUGWuC
9ol6EcJdOBUVUKIJX8t+qT1iGtMEd2oV0rUv+2cvHJrhZW24BSVnebW05n32z9Je
oPcHgdrH0ZJN9O0DV46QP1HTdVe7GvY1Fd+UjUFh4oIjwQyYSpO/smBHUffCmtyX
wljCbmjYM2yKyQe04C/+s8ZO+AFFtqx6srvnElQTXtfxkTWYPSrodDKmxqY81aR9
TFd5wWtApMeFT9DK5dDlneBpqn0gDE+JixlEx+pEZM6SDdO1arAg3PKZotuzndo0
1c0mqG6Lp5r464xi5g4kbPHNe1PFe+2tDCEW9BuYADe0v8PvpHMbGJNxOt+w8CcV
R/muH/KoKYs8Y9Ej03MRob1r7Xpv4/NO/1KLHhggxlihiUib1GVDguRNJmMYloo+
8FfoSMixPRJxUg03yZA479e4QSNI+5AryxzohXUCAwEAAaNjMGEwHQYDVR0OBBYE
FEamhG8kGg3PCElsgH8XYIWO04BXMB8GA1UdIwQYMBaAFEqs0+NaumvRXZkP+sTw
NMNvbr61MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3
DQEBCwUAA4ICAQCVeQKhOIDK0z8XdohL/vkypGGayBtU17lFfwZEiWuIeeLnZDrB
vzz1T1j91tx6MBWFEbP2FoJYCwaU9YSuOP8mhtmJM4v1MgC3aOGdMa3nKo2PbS9M
ECMLFB6Jpo7zVjVxwEz7WXA7/YJgR0g5ft/turJnbbUis0K0FBO/aYzc9gyBvg8I
GTB6GX6DDNuwT5EOjkynT3SqnRrnD2piZ0oQ2IIDMaYm/r/DFaMLoU6GRLmj74N0
P3Lefks4JX5C2KKEuM3/6/udMlmNrObjkIACe34icImkdxSXjmKj8Mg4YG8PBRU1
/j1yizB6GokGq2//0BkRMzBLJfUifOVa9mH/C303kA/CvJ42nQyDPLU77nunng3f
T//+/dQYk+OuMTTuVul2WSef+wW+kEspE8uTo/GH1ZmMRV0T7aPxt8/ASDbhEcQM
Okhbo49AhxuTHlOWS3xKxVIbxJ4P/P0v8c5bb/4D5gdGgBCoXQptiBRtS2suBt1M
g0eCtusMuUqPkwB5o5IU2MPGbHiiPzB4up5ZJHYe97rtKduM1cD+0v+w7ZxDrqdD
ebfAJjqaZLKNWEmy5fYt0lWUgDsA8aWUSLN2j/R3BbtXHcClmsZap3CSFzJlhbPz
9tQBVsfx6UJYZR2eAXTpEtEMYous6tKHcRS04/mCPBq+WhoYG39aX85g2Q==
-----END CERTIFICATE-----
END
[admin:10-10-28-16]: sslkeyandcertificate:certificate> save
[admin:10-10-28-16]: sslkeyandcertificate> save
+-------------------------
+-------------------------------------------------------------------------------+
| Field                  |
Value                                                                          |
+-------------------------
+-------------------------------------------------------------------------------+
| uuid                   | sslkeyandcertificate-2348ba24-1a56-4e9d-9833-
c8c3c1158714                    |
| name                   | admin-
intermediate                                                                   |
```

```
| type                   |
SSL_CERTIFICATE_TYPE_CA                                                    |
| certificate
|                                                                              |
|   version              |
2                                                                          |
|   serial_number        |
4098                                                                       |
|   self_signed          |
False                                                                      |
|   issuer
|                                                                              |
|     common_name        |
Intermediate                                                               |
|     organization       |
Avi                                                                        |
|     state              |
CA                                                                         |
|     country            |
US                                                                         |
|     distinguished_name | C=US, ST=CA, O=Avi,
CN=Intermediate                                        |
|   subject
|                                                                              |
|     common_name        | Same-Name-
Intermediate                                                     |
|     organization       |
Avi                                                                        |
|     state              |
CA                                                                         |
|     country            |
US                                                                         |
|     distinguished_name | C=US, ST=CA, O=Avi, CN=Same-Name-
Intermediate                                  |
|   signature_algorithm  |
sha256WithRSAEncryption                                                    |
|   not_before           | 2017-12-20
23:34:35                                                         |
|   not_after            | 2037-12-15
23:34:35                                                         |
|   fingerprint          | SHA1
Fingerprint=CD:96:22:87:B2:58:39:7C:7A:26:4B:3A:18:B2:99:CD:DB:73:B5:79 |
|
|                                                                              |
|   expiry_status        |
SSL_CERTIFICATE_GOOD                                                       |
|   days_until_expire    |
365                                                                        |
| key_params
|                                                                              |
|   algorithm            |
SSL_KEY_ALGORITHM_RSA                                                      |
|   rsa_params
|                                                                              |
|     key_size           |
```

```
SSL_KEY_4096_BITS                                                              |
|     exponent           |
65537                                                                          |
| status                 |
SSL_CERTIFICATE_FINISHED                                                       |
| ca_certs[1]
|                                                                                 |
|   name                 |
Intermediate                                                                   |
| format                 |
SSL_PEM                                                                        |
| certificate_base64     |
False                                                                          |
| key_base64             |
False                                                                          |
| tenant_ref             |
admin                                                                          |
+------------------------
+-------------------------------------------------------------------------------+
[admin:10-10-28-16]: > switchto tenant t1
Switching to tenant t1
[t1:10-10-28-16]: > show sslkeyandcertificate
+-----------------------------------+------------------------+------------------------+------
+-----------+
| Name                              | Issuer                 | Subject                | Self
| Algorithm |
+-----------------------------------+------------------------+------------------------+------
+-----------+
| System-Default-Cert               | System Default Cert    | System Default Cert    | True
| -         |
| System-Default-Cert-EC            | System Default EC Cert | System Default EC Cert | True
| -         |
| System-Default-Portal-Cert        | Default Portal Cert    | Default Portal Cert    | True
| -         |
| System-Default-Portal-Cert-EC256  | Default Portal EC Cert | Default Portal EC Cert | True
| -         |
| System-Default-Root-CA            | ca.local               | ca.local               | True
| -         |
| System-Default-Secure-Channel-Cert | ca.local              | node.controller.local  | -
| -         |
| admin-intermediate                | Intermediate           | Same-Name-Intermediate | -
| -         |
+-----------------------------------+------------------------+------------------------+------
+-----------+
[t1:10-10-28-16]: > configure sslkeyandcertificate t1-app

[t1:10-10-28-16]: sslkeyandcertificate> key --
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC+CGpOcfqxuUvl
sCa1+iYUu7EyrvJObDSdorIbjbu5qXpqL7lScGQq6uhbKKMAGM/JIiI75hOAeHN9
hYa/0v8BndV/AJ1zGpK3K5ahuVfrtsNIHk6q0SSw3YtB63/8nhwUiz/ZBgthfCJ/
eroG7RBEh8uOpPhXLJf88o1UOF5FcbrFsW5qvXQHMRSKK2I9wFkSSgNMCoOGOB7W
X+6aDG0ZAZt9eoQkPQNOxw9dEavqOZFqkHTSkyfWyHuw605dmRs2Cz8IZMhhEZvq
LUpe6HMFopxwTzt/5NyW0FJJW1K81WS46ab/tOIOLbkgNV9wLWMNeKvAEWzYPONS
QDzBOhrlAgMBAAECggEAEwNKh5C10WRFqLRoGxrtBnQE9Zo1Wg1Pclod0c3rc1b2
```

```
jXs64nmmO/kGyGAXduIEoA4POMj7OIZUn8FlSvn0U5gUDUHlfuewuCzfRE0D8+x0
O1n06vhD4II59Z13T7IOAywvdio5p0ZBOVnxFNJRJ1oizqHIywgGKOOnqj59iBr9
pw/LTthM1mozfUxVYxftSwDr91C7PTaYDE9prmw8wH1TL6I4skxKRVmagFwY0rtr
ViNhNigPjUB3xlEtv6RuwFeEmfcZMzkLCAoXbg1yv6Av5tGJwdCVwDwrpP6I/FHz
PwQdFmZRGZJI8QqdEcWYI/ewXYevCfDrQIWH+gFVIQKBgQDrcCmclzSqQt4xczJ2
ajXAaxnxLSJC/WYOIsIp3L5b/gqs+SUAIJXoVZMinOcygtJs3J4f0Zuy9NkddNn9
JVeMXs7rr7quXKSzX0100acB1NR4Sfq1RWboOxoiSgrUSx8D/ooaJE0JSlj0DtHl
+FVlSECAK2wpM8dFEMf9cAEIeQKBgQDOoRlQzkdnoDVL+gyIXnsA3ArnXDcig1x1
tSj0VqCEaGHhjngYHsmissaIw9ABlwZkt9maylX9PrLaAceGXPzeBvlK0PcgImZ+
2hYVp00znj4//JOsFe9joruKfaXrTLPvY8N0jYAmip6FJJ1eq4x8rL8gU/NdlMQf
5diVimhizQKBgCGs82bAgfnwgpOUJJ2nZ3TUXOuQRxxJ3nUbJ6aROnEyDxjash4o
iwimZNtIkhE5gRutGrj2ZEzelMeP1TZORw1+6h3wDsWt3qkBcrTI4Bh09scV3dRb
zvJcscpByPbAn/kUSXCfzJ0Nk1elXwSD1sMb6I3sqBXkoBYS5mgrwxoRAoGAXJmB
uN7YzS3U9LmYiDyfLyFtmYWQB92KwA1xzx5LTUtiIi0w0M5rWoh3xK7MNwoxiU2D
LYVjx9wjVuPZQPPHNtE1Qzwmo7YG7O5bW1TgmjNeflp463PhFmvFVCk/BBYZxTyW
SVNojN0ucUiZZeXHTdA0zw4QUG3s/saIq2udoDkCgYBS9FJxYZV/3eWZTV7E8RHO
4ABpujonzZcrxB/pIlQJhehVABopbMAGE0aGc7gGacu0DKsLNYL8Wkdqgs6WN9Yo
erlGXlJelgs4CSlZulInntFgdqC9Rj0sHjx6gCVEgg1lGkB++YrCLj2YuYN7L9JW
wk/YYUmjGLjqcHvBNDl0Gw==
-----END PRIVATE KEY-----
END
[t1:10-10-28-16]: sslkeyandcertificate> certificate
[t1:10-10-28-16]: sslkeyandcertificate:certificate> certificate --
-----BEGIN CERTIFICATE-----
MIIFAzCCAuugAwIBAgICEAEwDQYJKoZIhvcNAQELBQAwSTELMAkGA1UEBhMCVVMx
CzAJBgNVBAgMAkNBMQwwCgYDVQQKDANBdmkxHzAdBgNVBAMMFlNhbWUtTmFtZS1J
bnRlcm1lZGlhdGUwHhcNMTcxMjIwMjMzNDU2WhcNMzcxMjE1MjMzNDU2WjA3MQsw
CQYDVQQGEwJVUzELMAkGA1UECAwCQ0ExDDAKBgNVBAoMA0F2aTENMAsGA1UEAwwE
QXBwMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL4Iak5x+rG5S+Ww
JrX6JhS7sTKu8k5sNJ2ishuNu7mpemovuVJwZCrq6FsoowAYz8kiIjvmE4B4c32F
hr/S/wGd1X8AnXMakrcrlqG5V+u2w0geTqrRJLDdi0Hrf/yeHBSLP9kGC2F8In96
ugbtEESHy46k+Fcsl/zyjVQ4XkVxusWxbmq9dAcxFIorYj3AWRJKA0wKg4Y4HtZf
7poMbRkBm316hCQ9A07HD10Rq+o5kWqQdNKTJ9bIe7DrTl2ZGzYLPwhkyGERm+ot
Sl7ocwWinHBPO3/k3JbQUklbUrzVZLjppv+04g4tuSA1X3AtYw14q8ARbNg841JA
PME6GuUCAwEAAaOCAQUwggEBMAkGA1UdEwQCMAAwEQYJYIZIAYb4QgEBBAQDAgZA
MDMGCWCGSAGG+EIBDQQmFiRPcGVuU1NMIEdlbmVyYXRlZCBTZXJ2ZXIgQ2VydGlm
aWNhdGUwHQYDVR0OBBYEFBFU4BzZ35LC8gZlhXQG6pB9sDxNMGgGA1UdIwRhMF+A
FEamhG8kGg3PCElsgH8XYIWO04BXoUOkQTA/MQswCQYDVQQGEwJVUzELMAkGA1UE
CAwCQ0ExDDAKBgNVBAoMA0F2aTEVMBMGA1UEAwwMSW50ZXJtZWRpYXRlggIQAjAO
BgNVHQ8BAf8EBAMCBaAwEwYDVR0lBAwwCgYIKwYBBQUHAwEwDQYJKoZIhvcNAQEL
BQADggIBAAfp7d3STNGOQvPxMb+w9b4MjxXAdcFLCLiowcnh6wRS5/ALIjr+7oAt
5T+SzFx1jiZltRf7wk5Ot48+lKwSJ93oaqow82QAZZFeNvkYecL/HHqW7squC7Su
lmdxQ0DT/fkpedKu3koWjUvf90zb/LotdKN9GN4R2KwKY+p/73w1cDMxyqyPiSOH
dCX1fkG1du4HEujG+zVTlEO5Wc94zer4+C9g/QwTVkBH11MOLd9RSlStadYzy8Qs
wu1pPEXZbePA7urZGqgiYUTYKbW+Ck/EKqt8NxvyqvmYBqmfuEnOW1W7XH7Zlzli
dAFEfZ5U9we1YlduDT7KUHizBn8Uex1O1TCjn2XMt+5KhJ8yfNjqbwTyg7G1pHcG
ifl+u/PYTyrLwnf0s09/iw27oacSczDxB/yRe5W6wmhsgL0Rry1tZvAcIHPR2c5t
xstiAJVZVp+WSqJRbCR+KZYZS7IX3J09gtZy8ZDaEhCGtiE/liin4yxLEP4cgbCd
ctIdYP+3pYFC7Ij4BvT+cHtKFAIQ8gD3pSx+NHjX/cWnhjQIo4ljt+ash9YQz+70
hbsp3zDB+Qbnc6j1MuITHQneKKxVPBkvYK7bcqKmKRfjOIpFgtClWd9+YRBriBKo
CayuZ7LuJYYgVqnU6waCJaA9eZC/BSNUqqHzBYV49oBUpyDIWOTW
-----END CERTIFICATE-----
END
[t1:10-10-28-16]: sslkeyandcertificate:certificate> save
[t1:10-10-28-16]: sslkeyandcertificate> save
```

```
+-------------------------
+-------------------------------------------------------------------------------+
| Field                  |
Value                                                                          |
+-------------------------
+-------------------------------------------------------------------------------+
| uuid                   | sslkeyandcertificate-9ec6948b-f57c-49ac-
b9da-28092a3fd72a                    |
| name                   | t1-
app                                                                          |
| type                   |
SSL_CERTIFICATE_TYPE_VIRTUALSERVICE                                            |
| certificate
|                                                                                |
|   version              |
2                                                                              |
|   serial_number        |
4097                                                                           |
|   self_signed          |
False                                                                          |
|   issuer
|                                                                                |
|     common_name        | Same-Name-
Intermediate                                                 |
|     organization       |
Avi                                                                            |
|     state              |
CA                                                                             |
|     country            |
US                                                                             |
|     distinguished_name | C=US, ST=CA, O=Avi, CN=Same-Name-
Intermediate                                |
|   subject
|                                                                                |
|     common_name        |
App1                                                                           |
|     organization       |
Avi                                                                            |
|     state              |
CA                                                                             |
|     country            |
US                                                                             |
|     distinguished_name | C=US, ST=CA, O=Avi,
CN=App1                                                      |
|   signature_algorithm  |
sha256WithRSAEncryption                                                        |
|   not_before           | 2017-12-20
23:34:56                                                                 |
|   not_after            | 2037-12-15
23:34:56                                                                 |
|   fingerprint          | SHA1
Fingerprint=18:B1:FD:DC:AF:F0:62:0C:73:E1:56:FC:75:AE:86:93:2E:56:1E:75 |
|
|                                                                                |
|   expiry_status        |
```

```
SSL_CERTIFICATE_GOOD                                                    |
|   days_until_expire    |
365                                                                     |
| key_params
|                                                                         |
|   algorithm            |
SSL_KEY_ALGORITHM_RSA                                                   |
|   rsa_params
|                                                                         |
|     key_size           |
SSL_KEY_2048_BITS                                                       |
|     exponent           |
65537                                                                   |
| status                 |
SSL_CERTIFICATE_FINISHED                                                |
| ca_certs[1]
|                                                                         |
|   name                 | Same-Name-
Intermediate                                                  |
|   ca_ref               | admin-
intermediate                                                       |
| ca_certs[2]
|                                                                         |
|   name                 |
Intermediate                                                            |
| format                 |
SSL_PEM                                                                 |
| certificate_base64     |
False                                                                   |
| key_base64             |
False                                                                   |
| tenant_ref             |
t1                                                                      |
+-------------------------
+-------------------------------------------------------------------------------+
[t1:10-10-28-16]: >
```

Tenant-Scoped Clouds

For additional configuration flexibility, isolation, and delegation of ADC administration, the NSX Advanced Load Balancer supports the creation of a cloud inside a tenant.

The Tenant-scoped clouds feature is currently restricted to AWS, Azure, Linux Server Clouds, NSX-T, and no-orchestrator clouds. Attempts to choose other cloud types result in an error code.

For more information on no-orchestrator (no-access clouds), in which adding, removing, or modifying properties of an SE requires an administrator's manual intervention, see *Orchestrator Access Modes* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

Features

Configuration Flexibility

Users with Tenant-Admin role access can create/read/update/delete (CRUD) cloud infrastructure (VRF contexts, SE groups, and SEs) within their tenant (the one to which the cloud is scoped). In addition, tenant administrators can access and use the infrastructure of clouds in the admin tenant, as before, based on tenant settings. If the infrastructure in an admin-tenant cloud is configured in provider mode, tenant administrators can use it. It is used in addition to, and independent of any tenant-scoped clouds under their direct control.

- Provider Mode — It is a configuration in which SEs are shared across tenants. In provider mode, all the network resources of the cloud remain in the admin tenant and cannot be moved. To configure VRF contexts and move port groups into them, the NSX Advanced Load Balancer user must have write privileges for the admin tenant.

Isolation

The tenant-scoped cloud and its associated objects (VRF context, SEs, SE groups) are only visible in the cloud's tenant. They are not visible in other tenants or the admin tenant.

Administration

The admin user is authorized and responsible for performing the highest-level configuration operations, such as user creation and authorization, tenant creation, and all infrastructure associated with the admin tenant. However, the admin user can and likely prefers to delegate the creation of clouds scoped to their respective tenants to tenant administrators.

When creating a tenant-scoped cloud, NSX Advanced Load Balancer automatically:

1. Creates two VRF contexts for the cloud in the particular tenant. One is assigned data-plane traffic and the other is control-plane (management) traffic.
2. Creates the first SE group for the cloud in this tenant (it will be named Default-Group).
3. Changes default permissions to enable the user having the Tenant-Admin role for creating and managing tenant-scoped clouds and associated objects.

NSX Advanced Load Balancer UI Tenant-Scoped Cloud Creation

- Log in to the admin user, click **admin** on the top right of the toolbar to show the complete list of tenants defined and select a tenant from the **All Tenants** drop-down menu. The administrator's intent is to switch the context to that tenant.
- When a new tenant is selected, the name of the selected tenant is displayed in place of **admin** on the top right of the toolbar.

  Note The admin user is still logged in, and only the tenant context is changed.

- After navigating to **Infrastructure > Clouds**, click the **CREATE** drop-down menu. Select the desired cloud the user intends to create.

  Note It is not necessary for the admin user to create the cloud within the tenant.

- Cloud creation can proceed as normal.

- Typically, the tenant administrator of the selected tenant is logged in. No other tenants are listed when the selected tenant is clicked in the top right corner of the toolbar. It is preferrable to be limited to a single tenant.
- The user can also opt to create other clouds with the selected tenant.

Note The steps to create a cloud are same as for clouds that are not a tenant-scoped. It depends on the tenant context of the user.

All Tenants View

The All Tenants view provides a global view of all NSX Advanced Load Balancer objects across all tenants. This includes all the tenants, including the admin tenant.

The All Tenants view is intended for viewing, updating, and deleting objects. New objects cannot be created in this view.

Note The All Tenants view is supported only for administrative user accounts configured to be a superuser. To use the feature, you must log in with an NSX Advanced Load Balancer user account that has the role System-Admin and the superuser attribute enabled.

To switch the focus of the management session to the All Tenants view, use any of the following methods.

Using the UI

This task helps you to switch the focus of the management to the All tenants View using UI.

Procedure

1. Click the username in the upper right corner of the display.
2. Select **All Tenants** from the drop-down menu.

Using the CLI

This task helps you to switch the focus of the management to the All tenants View using CLI.

Enter the following command.

```
switchto tenant *
Switching to tenant *
```

Specifying /* (wildcard) instead of a tenant name gives access to objects across all tenants.

Using the API

This task helps you to switch the focus of the management to the All tenants View using API.

Set the value of X-Avi-Tenant in the HTTP request header to `/*` (wildcard).

Tenants Versus SE Group Isolation

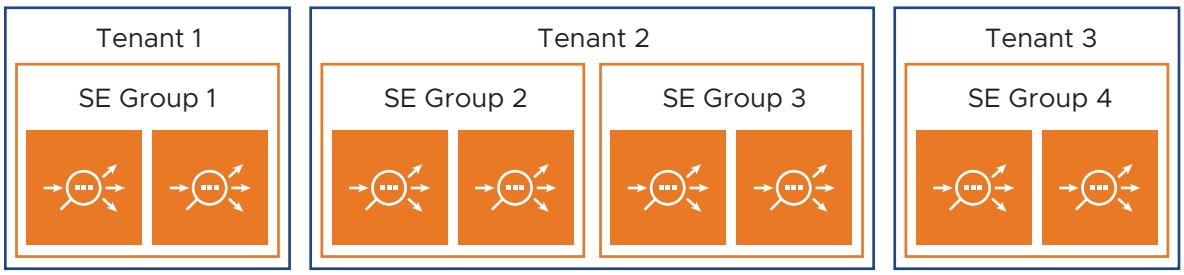
There are multiple ways to create isolation within the NSX Advanced Load Balancer. This section discusses the difference between tenants and SE groups and their relationship to data plane isolation and control plane isolation.

Isolation	Tenant: Provider Context	Tenant: Tenant Context	SE Group
Control Plane	Yes	Yes	No
Data Plane	No	Yes	Yes

Service Engine Groups

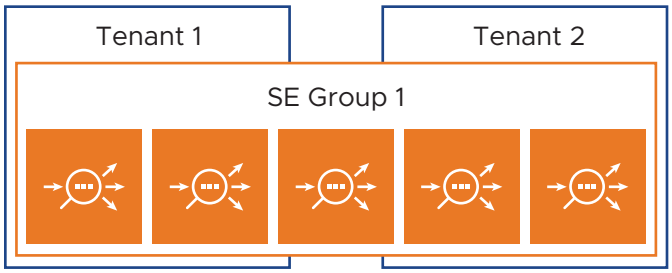
SE groups are an inherent method of grouping SEs to provide data plane isolation. A single tenant can have one or more SE groups. Multiple tenants can also point to one or more SE Groups. Only one SE group can serve a virtual service. If one of its SEs fails, another SE can take over within the same SE group. SEs in other SE groups cannot be pulled in to provide capacity for another SE group. This ensures data plane isolation.

Example: Example 1:



An administrator manages an application in both test and production environments. The virtual service of each application must be deployed on a different SE group. For ease of management, both applications can be in the same tenant (tenant 2 in the diagram), though arguments can be made for separating these different environments into two separated tenants (such as tenant 1 and 3 in the diagram).

Example: Example 2:



A cloud service provider manages multiple customer applications. Each customer is assigned to a unique tenant, ensuring complete configuration isolation. The service provider can allow all tenants to have isolated SEs, or they can choose to place multiple tenants in the same SE group to reduce idle resources.

Sharing Admin Profiles Across Tenants

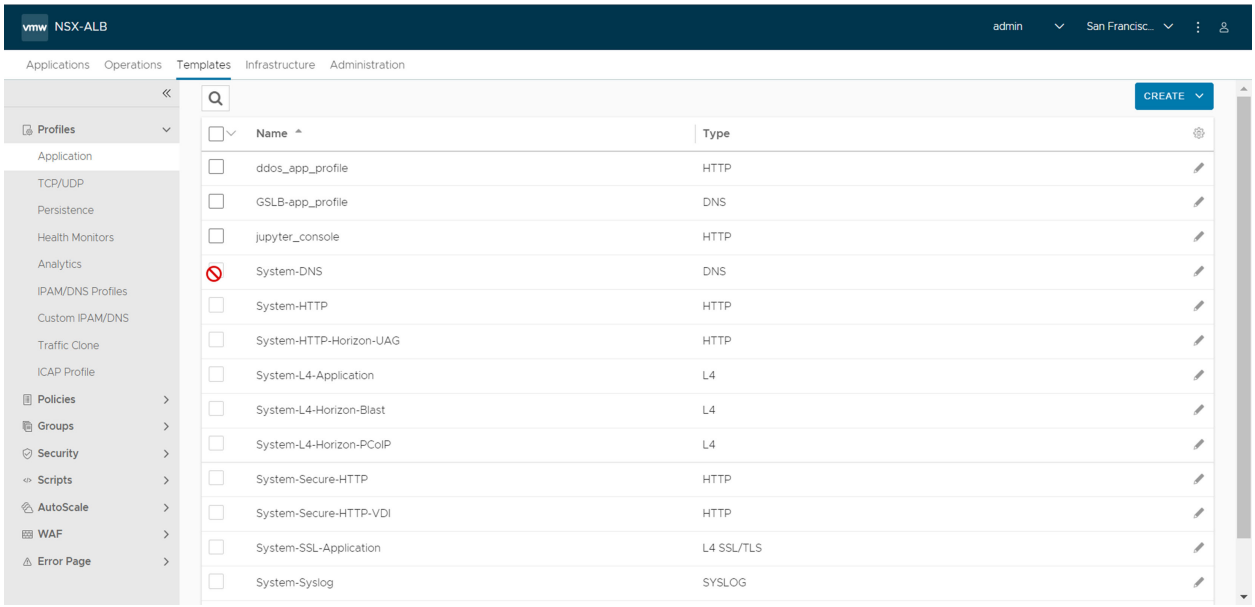
Profile objects in the admin tenant are shared across all tenants in the system. This includes objects visible under the **Templates** section of the NSX Advanced Load Balancer UI such as Health Monitor, Application Profile, SSL Profile, Network Profile, PKI Profile, and so on.

The NSX Advanced Load Balancer defines a set of system default profiles as a part of the installation. Though these objects can be edited, they can neither be renamed nor deleted. In addition to these profiles, the administrator can create, edit and delete custom profile objects suited to their specific deployment. Both the system default profiles and any custom profiles that are created in the admin tenant are automatically shared across all the tenants in the system. Tenants are able to view and use these profiles, but cannot delete the shared admin profiles.

Tenants can chose to override the default profile parameters to that which best suits their application deployment. A tenant can do this by either editing the shared admin profile or creating new custom profiles under their tenant context. Editing the shared admin profile creates a copy-on-write effect, such that a new profile with the same name (but a different UUID) is created in the specific tenant's context. Making changes to this profile object does not affect other tenants in the system. Deleting such a profile created using copy-on-write brings the shared admin profile back into the tenant context.

The following example demonstrates the sharing functionality.

Navigate to **Templates > Profiles > Application**. In the Application Profile page, a couple of custom profiles have been created by the administrator. Note that the default profiles (System-xxx) cannot be deleted (check box is disabled for selection), but the custom profiles can be selected for deletion (adjacent check box can be selected).



In the context of tenant *avidev*, though the shared admin profiles are visible, none of the profiles can be deleted.

By clicking the edit icon, tenant *avidev* edits the *System-HTTP* and *Custom-HTTP* profiles.

The formerly disabled check boxes turn selectable, indicating that the profiles have undergone a copy-on-write operation and can now be selected for deletion.

Tenant *avidev* deletes the *System-HTTP* and *Custom-HTTP* profiles. The shared admin profiles are now visible and cannot be deleted.

User Authentication and Authorization

7

To access NSX Advanced Load Balancer through the GUI, REST API, or CLI, a valid user account is required. Each user is assigned a role which grants permissions and access to read or write to the objects in NSX Advanced Load Balancer. You can restrict accounts to specific tenants and grant configure different roles within each tenant.

User accounts are maintained locally within NSX Advanced Load Balancer or remotely using an external AAA server.

When the same user account exists locally and remotely, NSX Advanced Load Balancer authenticates the credentials locally first and then authenticates the remote user account.

For SSH access, the NSX Advanced Load Balancer Controller will also attempt to authenticate the user after failing to find the user in the local or remote auth databases. Local and remote users are not created in Linux and may not have Linux access, with the exception of admin account.

Note If remote authentication (LDAP, TACACS, SAML, and more) is enabled, you can deactivate local authentication in the NSX Advanced Load Balancer Controller by deactivating the **Enable Local User Login** option in the **Edit System Settings** screen.

For authenticating users remotely to log into NSX Advanced Load Balancer. Auth Profile discusses different ways of remote authentication supported in NSX Advanced Load Balancer.

Read the following topics next:

- Auth Profile
- User Roles
- User Accounts

Auth Profile

An Auth profile is a set of Authentication, Authorization, and Accounting (AAA) attributes used by NSX Advanced Load Balancer users to log into NSX Advanced Load Balancer.

To configure an Auth profile, perform the following steps.

1 Navigate to the **Templates > Security > Auth Profile** and click **Create**.

2 Enter a name for the profile and select the **Type**.

3 Configure the settings specific to the type of profile selected.

4 Click **Save**.

LDAP Authentication Profile

NSX Advanced Load Balancer supports user authentication using Lightweight Directory Access Protocol (LDAP). LDAP is a commonly used protocol for accessing a directory service. A directory service is a hierarchical object oriented database view of an authentication system. LDAP settings can be configured in an authentication profile.

Creating an LDAP Auth Profile

To create an LDAP Auth Profile, perform the following steps.

1. Navigate to **Templates > Security > Auth Profile**.

2. Under the **General** tab, enter the **Name** of the profile and select the **Type** of the Auth Profile as LDAP.

3. Configure the **LDAP** settings.

4. Configure the **HTTP Authentication** settings.

5. Click **Save**.

Configuring LDAP Settings

1. Under the LDAP tab, click **Add** to enter the LDAP server IPs.

Note For Controller authentication, multiple LDAP servers are supported only if they belong to the same cluster. Otherwise, the Controller tries to authenticate with the first reachable server.

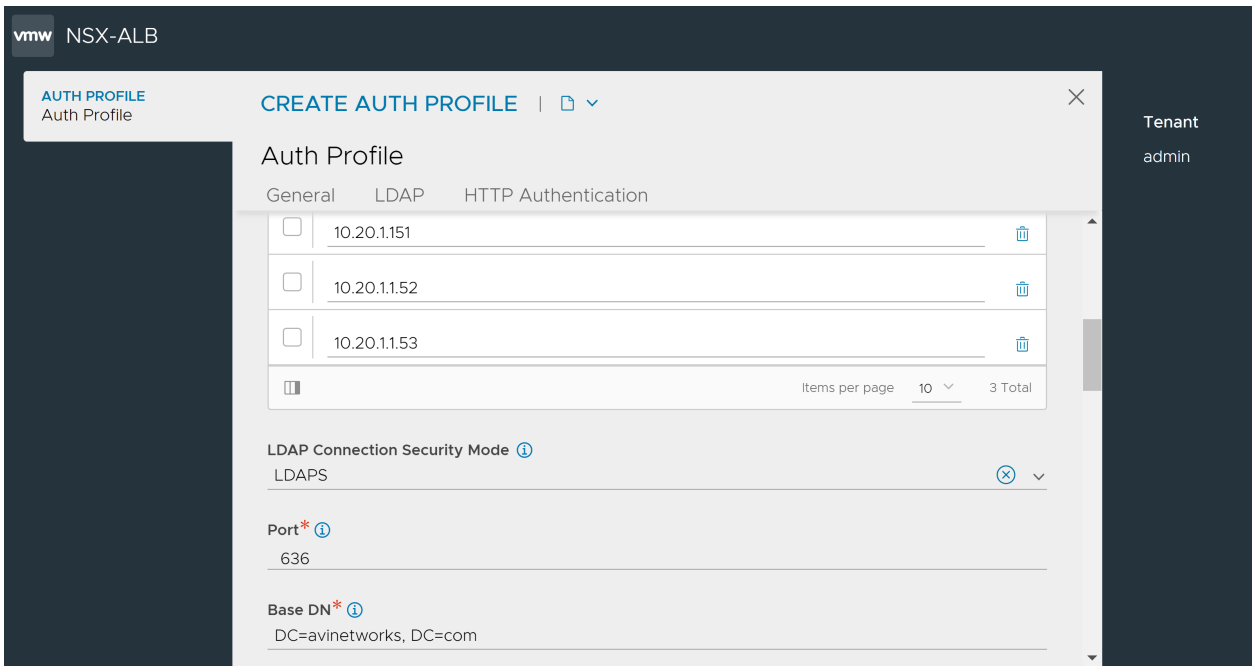
2. Select the **LDAP Connection Security Mode**, if required.

3. Enter the **Port** on which NSX Advanced Load Balancer queries the LDAP servers. Use port 389 for LDAP and 636 for LDAPS (SSL).

Note Just configuring the port does not determine if the LDAP security mode is LDAPS. Hence, explicitly select **LDAPS** as the **LDAP Connection Security Mode** to ensure secure LDAP is in use.

4. In the **Base DN** field, enter the top level path in the directory server object hierarchy.

Note Using a specific base DN expedites all queries to the LDAP server in contrary to a generic, high level base DN which makes the queries reach out to more parts of the directory's hierarchy. The base DN must be set to the topmost (most general) path to which the account has access.



5. Select the **LDAP Bind Settings**.

Administrator Bind

- The administrator account provided will be used to search for users and user group memberships across the LDAP server.

Note Admin Bind is the recommended LDAP Bind option.

- Enter the **Admin Bind DN** and **Admin Bind Password**. The account used must have access to search the directory tree for both users and user groups.
  - NSX Advanced Load Balancer uses the configuration to search for users or groups. An LDAP search typically requires:
  - Top-level directory hierarchy (search DN) to start the search.
  - The scope value limits the search to one of the following: base (one-level deep) or entire subtree.
  - Filter to match only on entries of a given class or category

Note If the LDAP admin bind password is changed, the user must restart the NSX Advanced Load Balancer portal for the authentication to work due to a limitation in the uwsgi server.

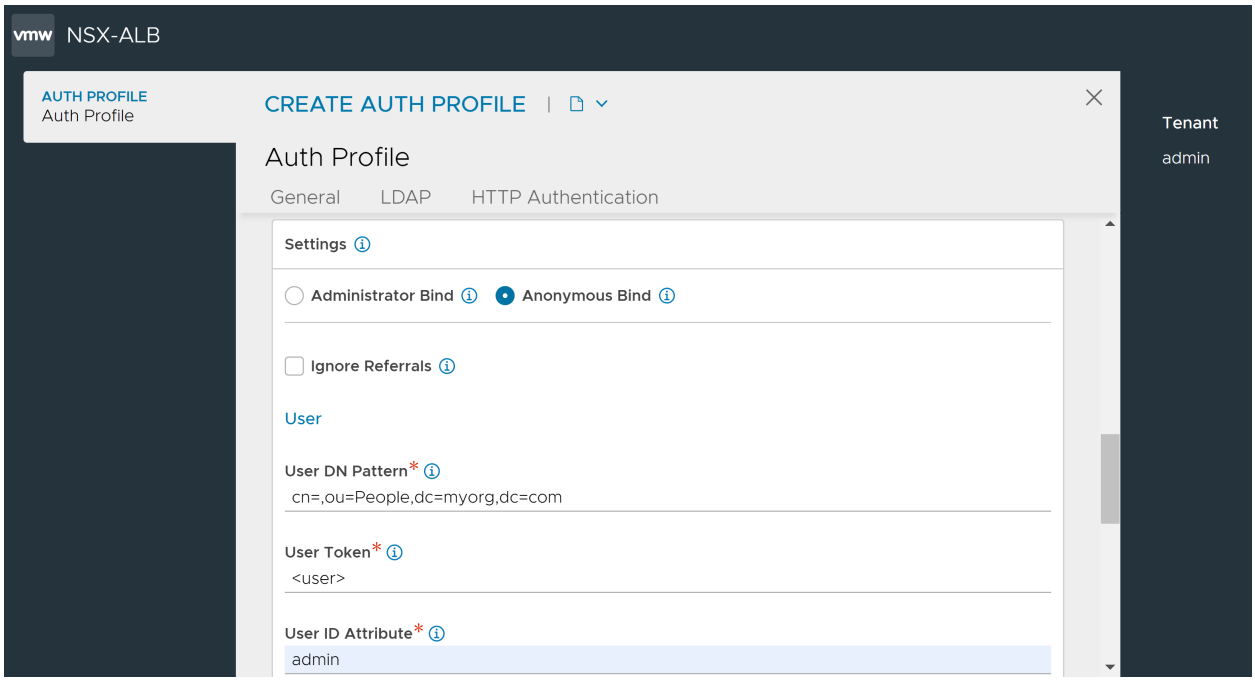
Anonymous Bind

- Anonymous bind is useful when you do not have access to an administrator account on the LDAP server(s). Anonymous bind can only be used to authenticate the user and cannot be used to authorize the user.

- To configure Anonymous Bind, enter the following:
  - **User DN Pattern** used to bind an LDAP user after replacing the user token with the real user name. The pattern must match the user record path in the LDAP server.
  - **User Token** which is replaced with the real user name in the user DN pattern.
  - **User ID Attribute** is the unique property that identifies a single user record. This value must match with the user name used at the login prompt.

Note Authentication profiles that use anonymous bind cannot be used for role or tenant mapping.

An LDAP profile configured with Anonymous Bind is as shown below:



6. Enable **Ignore Referrals** for the group search to skip referrals links that connect to another LDAP server.

Note Enabling the option **Ignore Referrals** can quicken the group searches by preventing delay in LDAP group search due to unnecessary referral searches.

7. Under **User**, define the scope of search for users who log into NSX Advanced Load Balancer.

User Search DN

LDAP user search DN is the root of search for a given user in the LDAP directory. Only user records present in this LDAP directory sub-tree are allowed for authentication. Base DN value is used if this value is not configured.

User Search Scope

LDAP user search scope defines how deep to search for the user starting from user search DN.

User ID Attribute

LDAP user ID attribute is the login attribute that uniquely identifies a single user record. The value of this attribute must match the user name used at the login prompt.

8. Under **Group**, define the parameters to search for a user's group membership.

Group Search DN

- LDAP group search DN is the root of search for a given group in the LDAP directory. Only matching groups present in this LDAP directory sub-tree will be checked for user membership.
- If this value is not defined, the base DN value is used.

Group Search Scope

- Define the LDAP group search scope using filters for the group starting from the group search DN.
- The different levels of group search available are:
  - Scope Base
  - Scope One Level
  - Scope Subtree

Group Filter

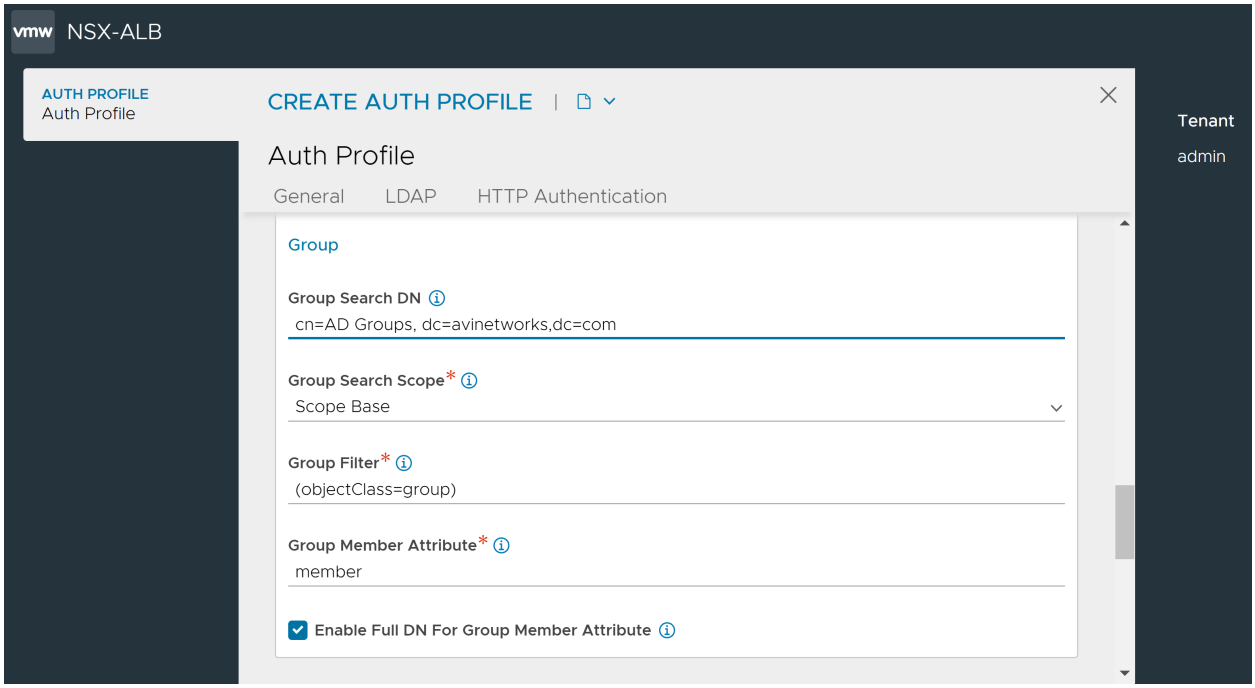
- A search can find many different types of objects under the search DN. Specify the Group Filter to pick up objects specific to the group only.
- NSX Advanced Load Balancer appends a user-specific group membership filter to the configured group filter to check a specific user’s group membership.

Group Member Attribute

- Enter the LDAP group attribute that identifies each of the group members.
- For example, member and memberUid are commonly used attributes.

Enable Full DN For Group Member Attribute

- Enable this option if the group member entries have full DNs instead of just user ID attributes.
- For example, a configured group filter (objectClass=group) is extended by NSX Advanced Load Balancer to a full filter when user “bob” logs in to something such as the following: (&(objectClass=group)(member=bob))



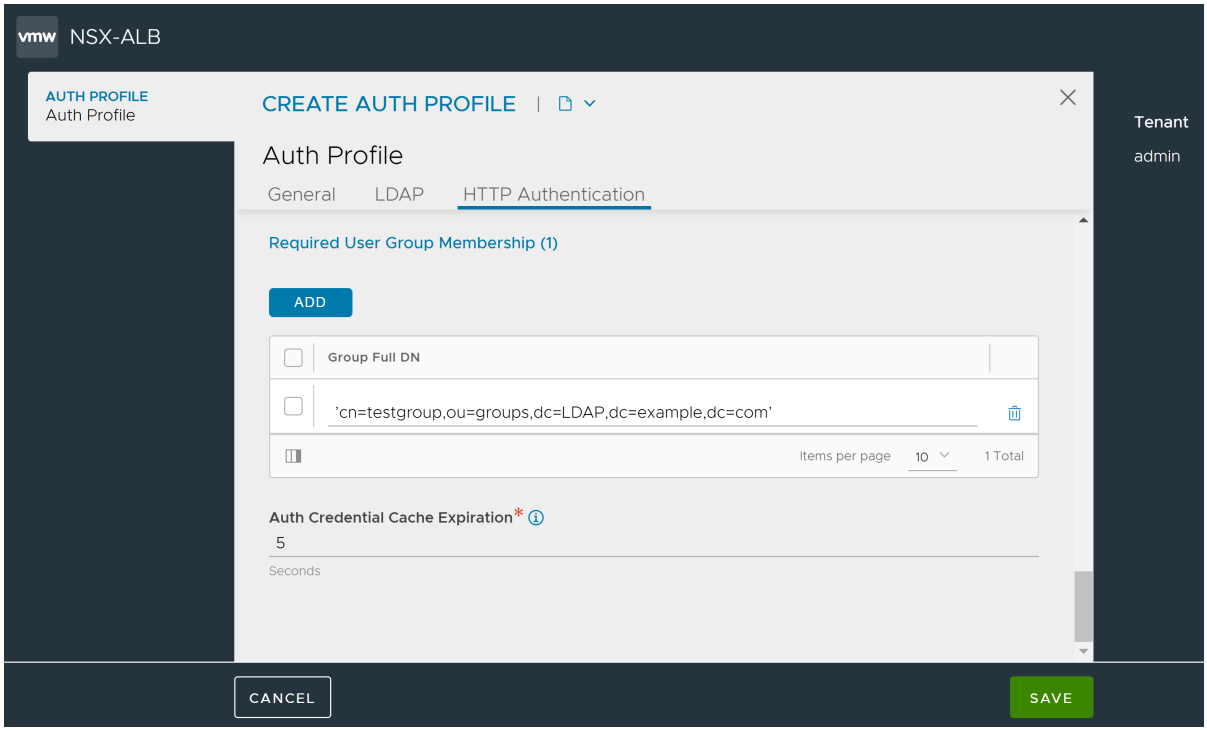
After configuring the LDAP settings, configure the settings under the **HTTP Authentication** tab.

Configuring HTTP Authentication

1 Enter the **Client User ID Header Name** to insert an HTTP header into the client request before it is sent to the destination server. This field is used to name the header. The value will be the client's User ID. This same User ID value will be used to populate the User ID field in the Virtual Service's logs.

2 Under **Required User Group Membership**, click **Add** to identify the groups which the user must be a member of. Each group is identified by the DN, for example, 'cn=testgroup,ou=groups,dc=LDAP,dc=example,dc=com'.

3 In the field **Auth Credential Cache Expiration**, specify the maximum allowed length of time for which a client's authentication is cached.



4 Click **Save** to complete the LDAP Auth profile creation.

LDAP Authentication Profile Testing

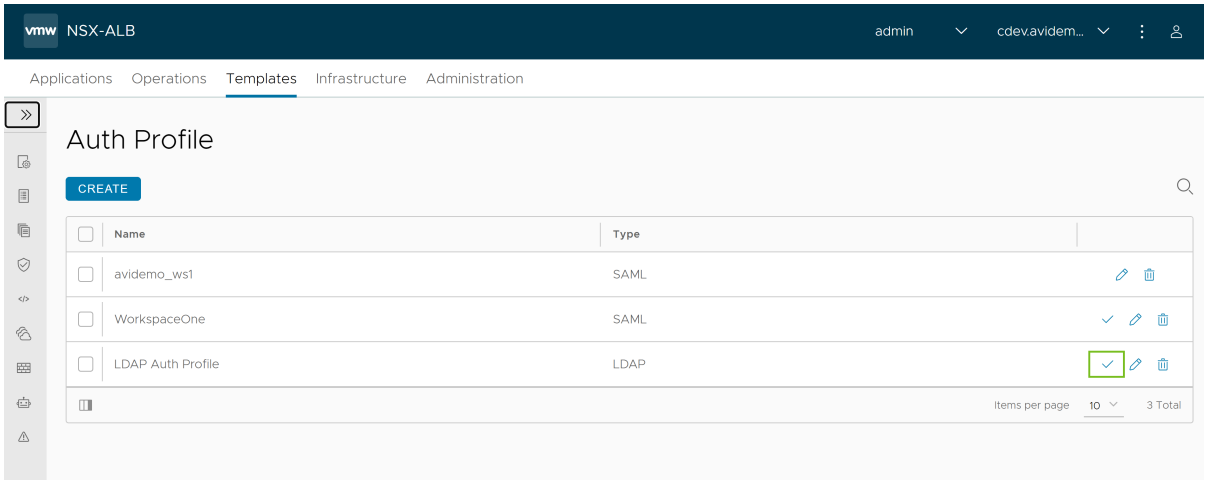
NSX Advanced Load Balancer provides an option for testing authentication profiles configured on the NSX Advanced Load Balancer Controller.

Testing an Authentication Profile

To verify the LDAP profile, perform the following steps.

1 Navigate to **Templates > Security > Auth Profile**.

2 Click the verify icon to view the **VERIFY AUTH PROFILE**.

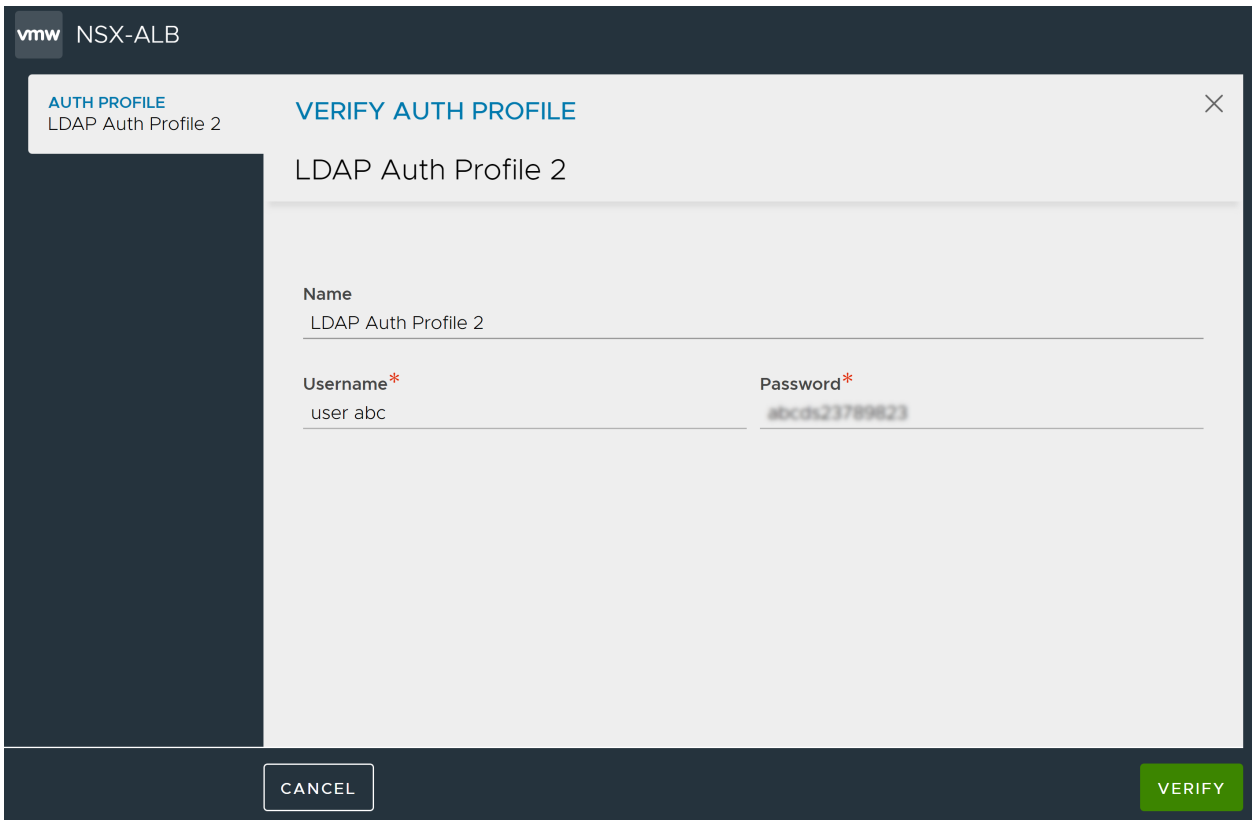


Note Depending on whether the LDAP Auth Profile has Administrator Bind or Anonymous Bind configured, the **VERIFY AUTH PROFILE** screen displays a different set of options.

3 Enter the details required in the **VERIFY AUTH PROFILE**.

Verifying LDAP Profile with Anonymous Bind

If the LDAP authentication profile is configured to use anonymous binding for authentication requests, the popup for testing the profile prompts for the LDAP user's user name and password.



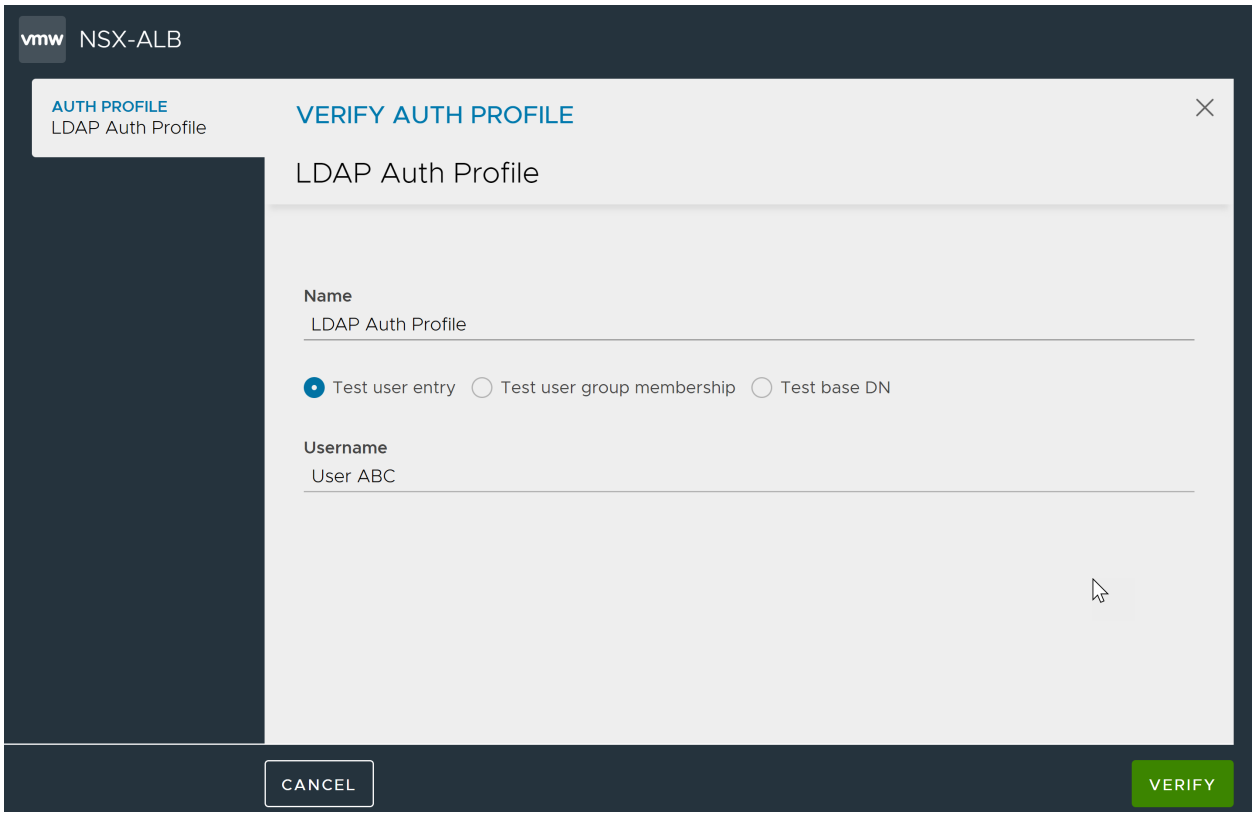
In the **VERIFY AUTH PROFILE** screen, enter the **Username** and **Password** , and click **Verify**.

Testing whether a user can bind successfully verifies that the LDAP authentication profile is configured correctly to authenticate users with the same user DN pattern.

Verifying LDAP Profile with Administrator Bind

If the LDAP authentication profile is configured to use administrator binding for authentication requests, one of the following types of information can be specified on the verification popup for the profile.

In the **VERIFY AUTH PROFILE** screen, select the required option to verify and enter the **Username**.



Test user entry

- Searches the LDAP server's database for the specified user name, and returns the corresponding user entry from the LDAP database.
- This option is useful for listing all attribute key-value pairs for any given user. The user search settings configured in the authentication profile are used.

Test user group membership

- Lists all group memberships for the specified user. The group search settings configured in the authentication profile are used.

Test base DN

- Returns all objects under the base DN.

- This option is useful for testing administrator permissions and for reading the DN tree of the LDAP server.

This process can identify some common error scenarios like:

- LDAP server IP/port is incorrect
- Bad user name or user search settings are incorrect
- User is either not a member of any group or the group search settings are incorrect

LDAP Configuration Examples

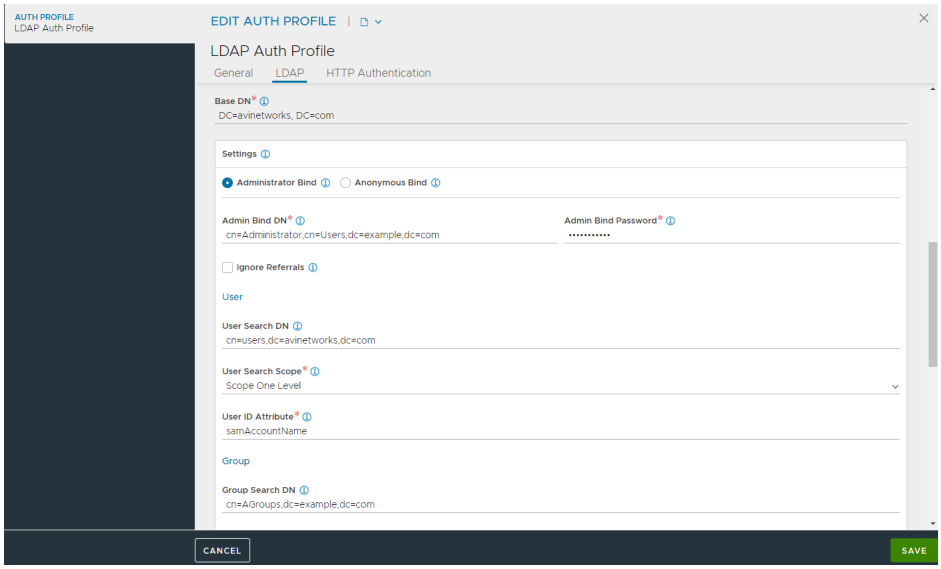
This section has several sample LDAP profile configurations that can be used in different scenarios.

For more information on creating an LDAP profile, see Configuring LDAP Settings.

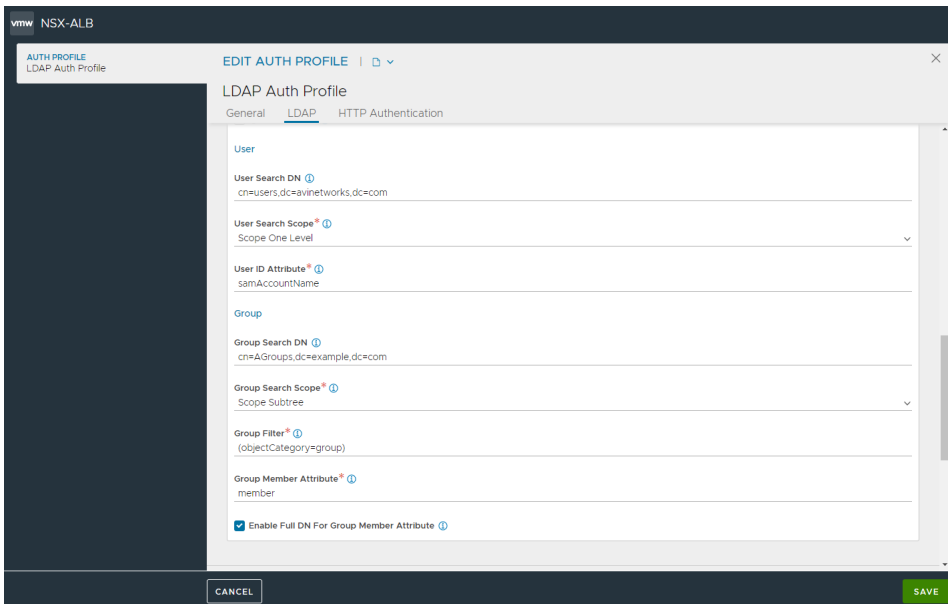
Example: Example 1

Use Case	Binding Option	Configuration Details
Active Directory common settings	Administrator Bind	With the configuration for administrator bind, the group membership includes the full DN.

The LDAP profile configured with Administrator Bind and user search configuration is as shown below:



The Group search configuration for the LDAP profile is as shown below:

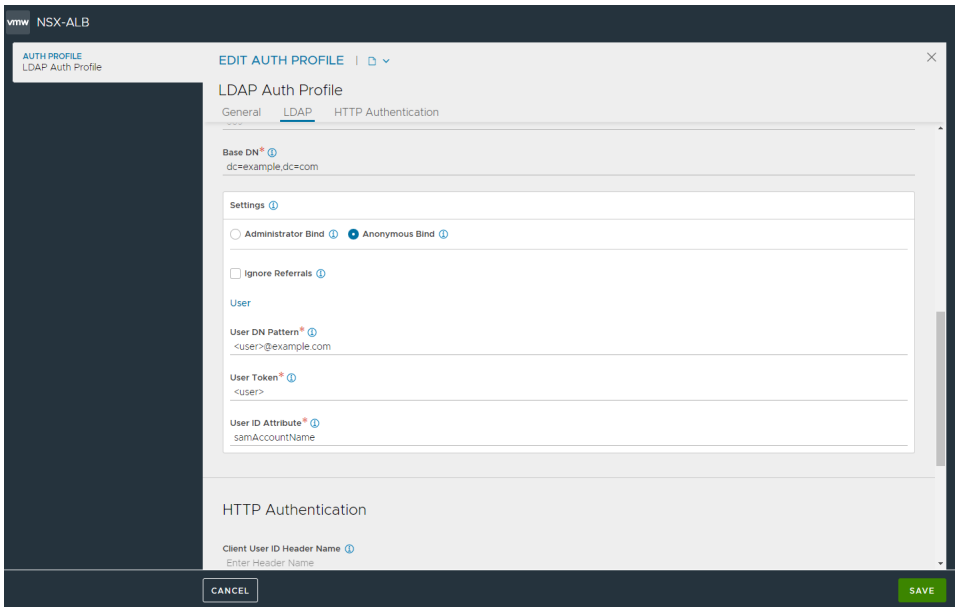


Note The **User ID Attribute** and the **Group Member Attribute** must match as configured in the Active Directory. The quickest way to verify is to check the response by testing the **Base DN**.

Example: Example 2

Use Case	Binding Option	Configuration Details
Active Directory common settings	Anonymous Bind	If the LDAP/AD user can bind with the DN jdoe@example.com and password, it validates the user login.

The LDAP Profile configured with Anonymous Bind is as shown below:



TACACS+ Authentication

NSX Advanced Load Balancer supports authentication and authorization of NSX Advanced Load Balancer users with Terminal Access Controller Access Control System (TACACS+). TACACS+ is an open standards protocol that handles authentication and accounting.

Creating a TACACS Plus Profile

In NSX Advanced Load Balancer, TACACS+ settings are configured in an auth profile. To create a TACACS+ auth profile, complete the following steps.

1 Navigate to the **Templates > Security > Auth Profile**.

2 Click **Create** to view the **CREATE AUTH PROFILE** screen.

3 Select the **Type** as TACACS_PLUS.

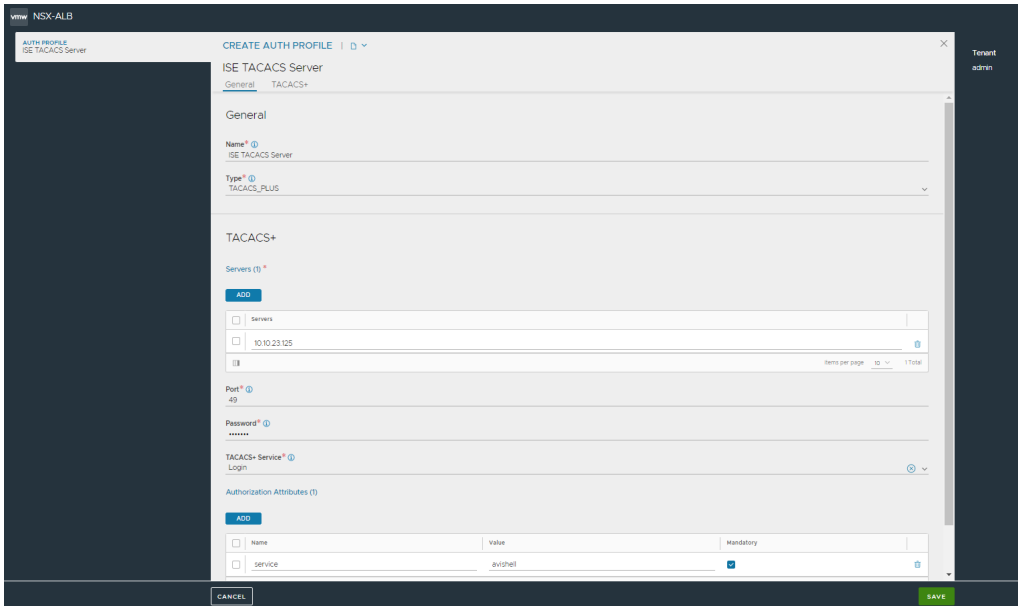
4 Under the **TACACS+** tab, click **Add** to enter TACACS+ server IP address.

> **Note** You can add multiple servers. If the first server does not respond, NSX Advanced Load Balancer tries the next server. If there is no response from this server as well, NSX Advanced Load Balancer tries to connect to the next server. Each server is tried once. If all the servers fail or timeout, the request is failed.

5 Enter the TACACS+ server **Port**. By default, the value is 49.

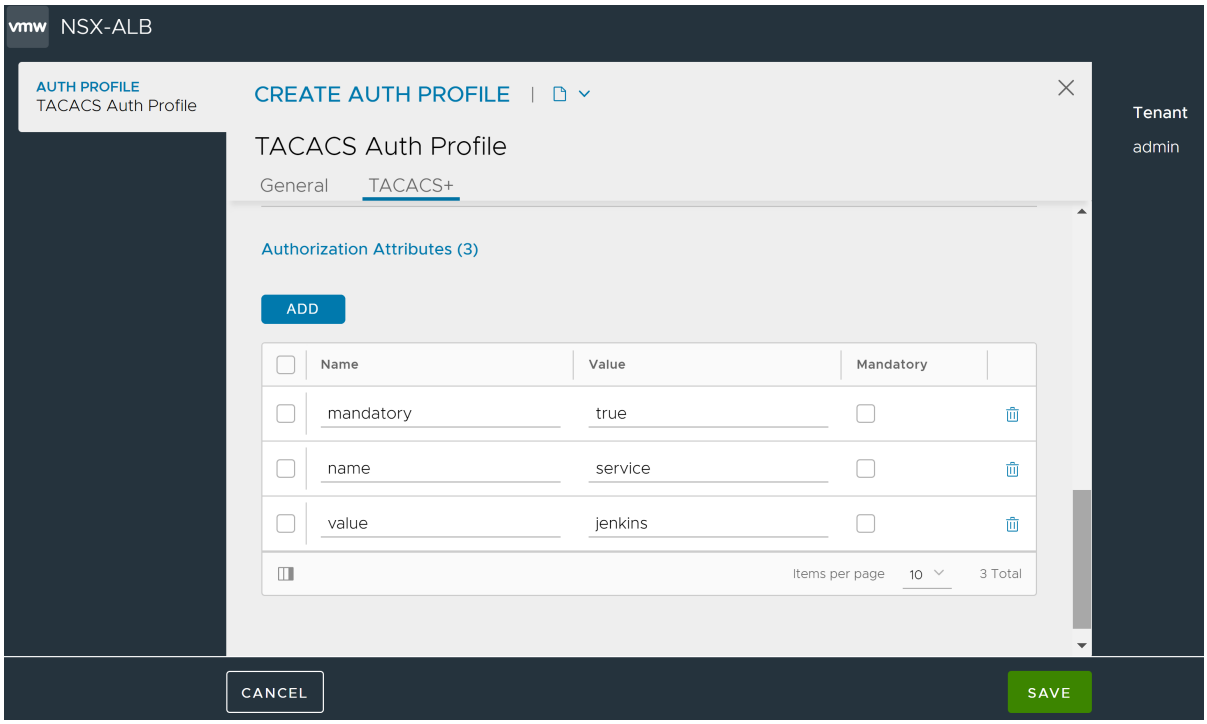
6 Enter the TACACS+ server shared secret as **Password**.

7 Select the **TACACS+ Service** type used in all authentication and authorization queries.



8 Under **Authorization Attributes**, click **Add** and enter the set of attribute value pairs under **Name** and **Value** to identify the host. The TACACS+ server configures user-level authorization based on these attributes. Cisco Access Control Servers (ACSs) typically expect authorization attribute values for “service” and “protocol” to be populated to identify and authorize an NSX Advanced Load Balancer user. Authorization attributes from a TACACS+ server can be used to map users to various roles and tenants.

9 If the attribute is required, check **Mandatory**.



10 Click **Save**.

Authentication and Authorization

Authentication and authorization of an NSX Advanced Load Balancer user with TACACS+ takes place as follows:

1 AUTHEN START packet from NSX Advanced Load Balancer to TACACS+ server. Contains:

- action=login
- authen_type=ascii
- service=
- user=
- remote_addr=

2 AUTHEN REPLY packet from TACACS+ server to NSX Advanced Load Balancer. It contains status of type GETPASS indicating that password needs to be supplied for the user message field with text “Password.”

3 AUTHEN CONTINUE packet from NSX Advanced Load Balancer to TACACS+ server. It contains user message field with actual password from user.

4 AUTHEN REPLY packet from TACACS+ server to NSX Advanced Load Balancer. Contains:

- SUCCESS status if password is valid and user is allowed
- FAILED status

5 AUTHOR START packet from NSX Advanced Load Balancer to TACACS+ server. Contains:

- User name of the user
- Remote address of the user
- Authorization attribute name, value and whether or not they are mandatory
- An authorization attribute string “abc=xyz” that indicates an attribute named “abc” is mandatory and has value “xyz”
- An authorization attribute string “abc*xyz” that indicates an attribute named “abc” is optional and has value “xyz”

6 AUTHOR REPLY packet from TACACS+ server to NSX Advanced Load Balancer. It contains one of the following:

- PASS_ADD or PASS_REPL status, which authorizes the successfully authenticated user with attribute value pairs to be added or replaced.
- FAIL status, indicating the user is not authorized.

Encryption

All TACACS+ packets are encrypted, whereas the 12-byte header is passed in the clear. Encryption is part of the TACACS+ standard and is compatible with all TACACS+ servers.

Error Handling

If an error is indicated in the **Status** field of any reply packet during this process, the user login is rejected and results in a failure.

Examples of TACACS+ Configuration

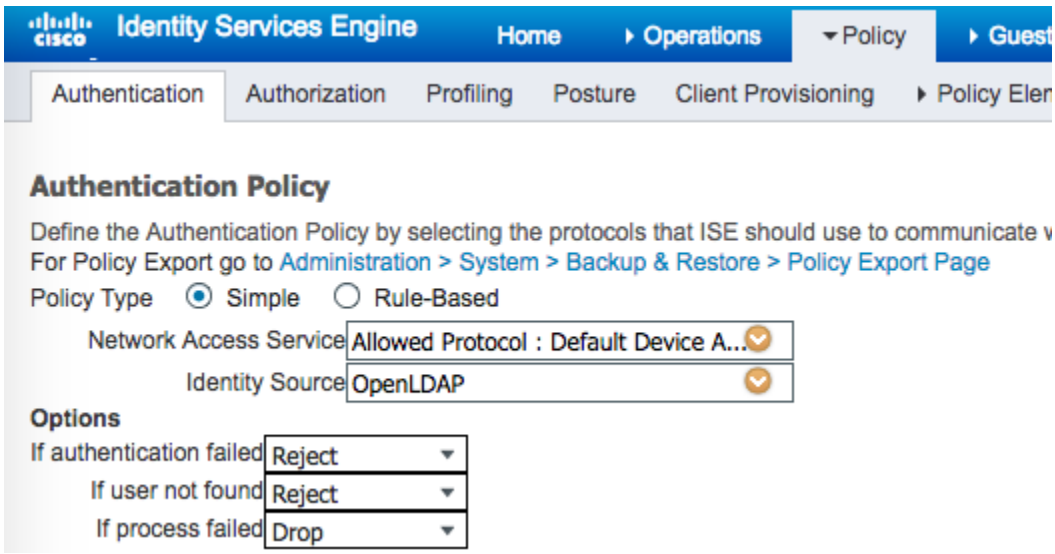
This section explains the examples of TACACS+ Configuration.

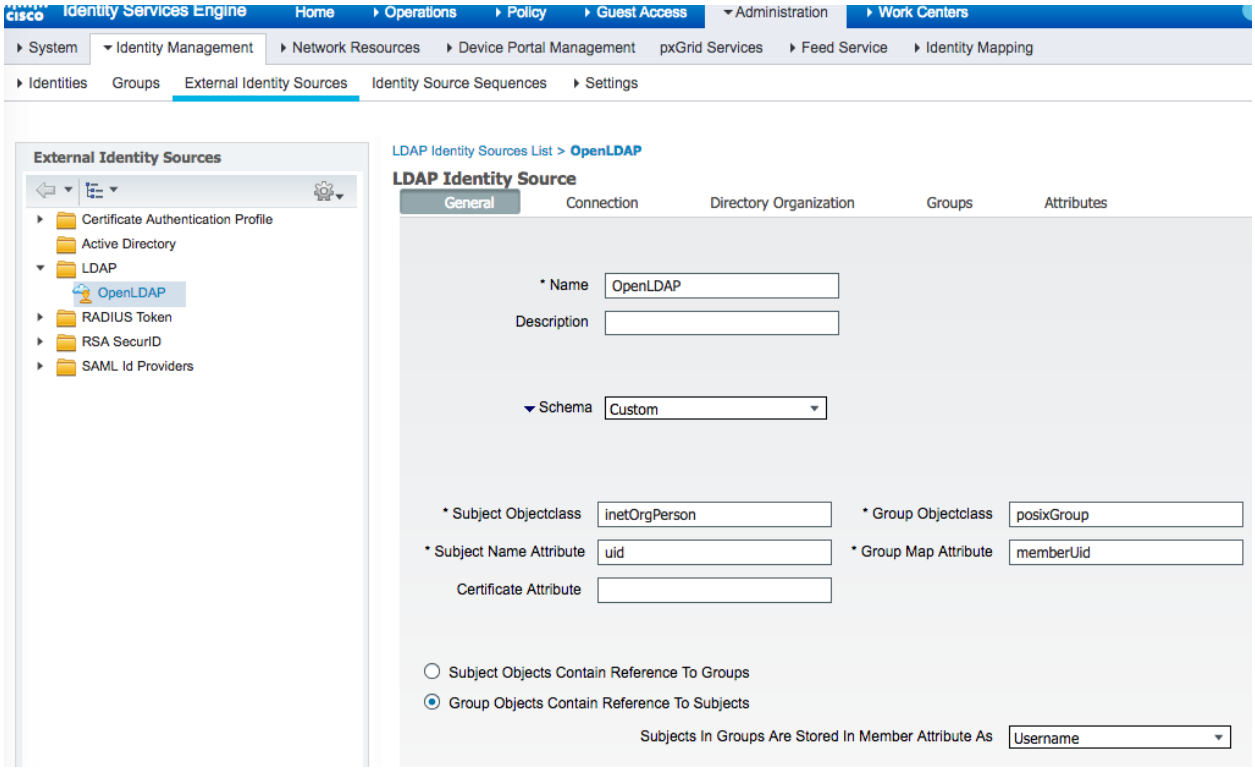
ISE TACACS+ Server

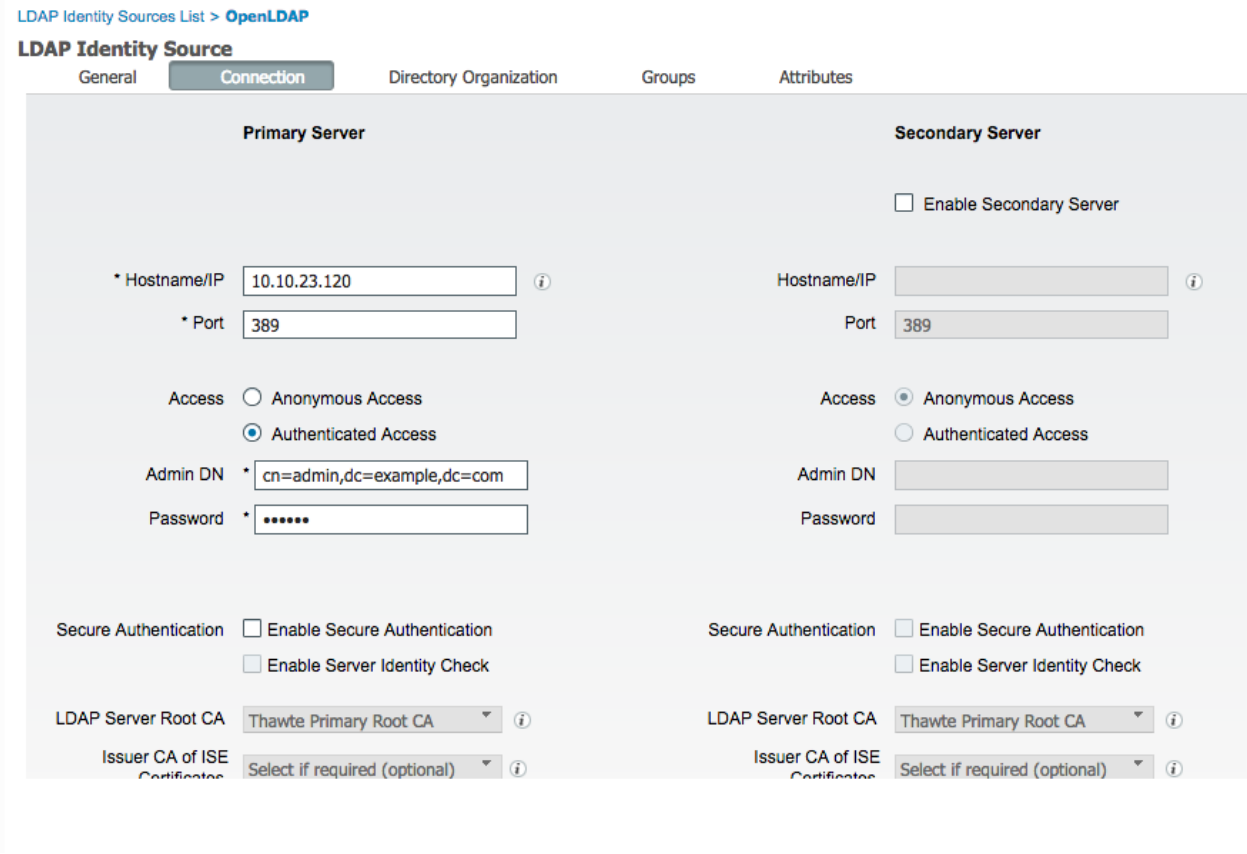
Cisco ISE is a security policy management platform that provides secure access to network resources. Cisco ISE functions as a policy decision point and enables enterprises to ensure compliance, enhance infrastructure security, and streamline service operations.

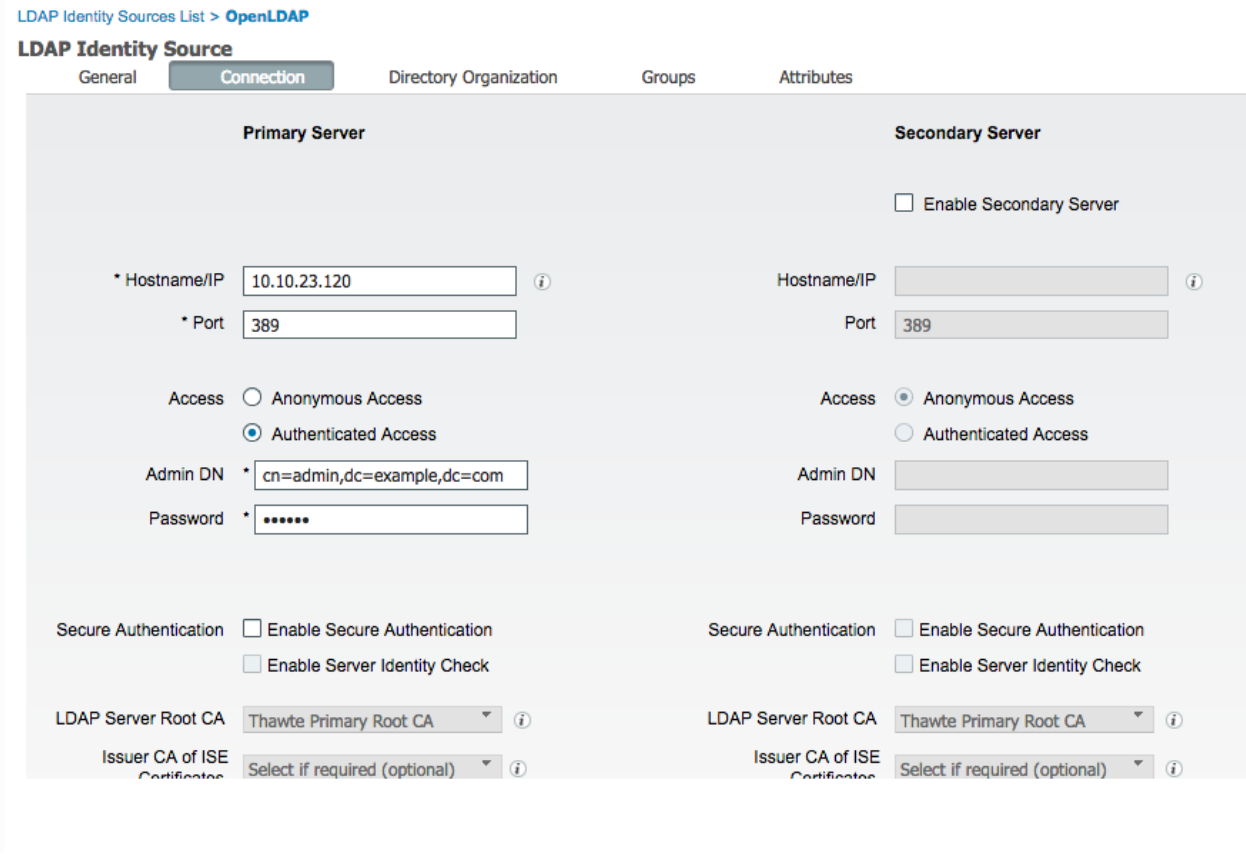
To set up an ISE TACACS+ server as a remote authentication and authorization system for NSX Advanced Load Balancer, follow the steps given below:

The ISE server is generally configured with external Identity Sources (in this case OpenLDAP).

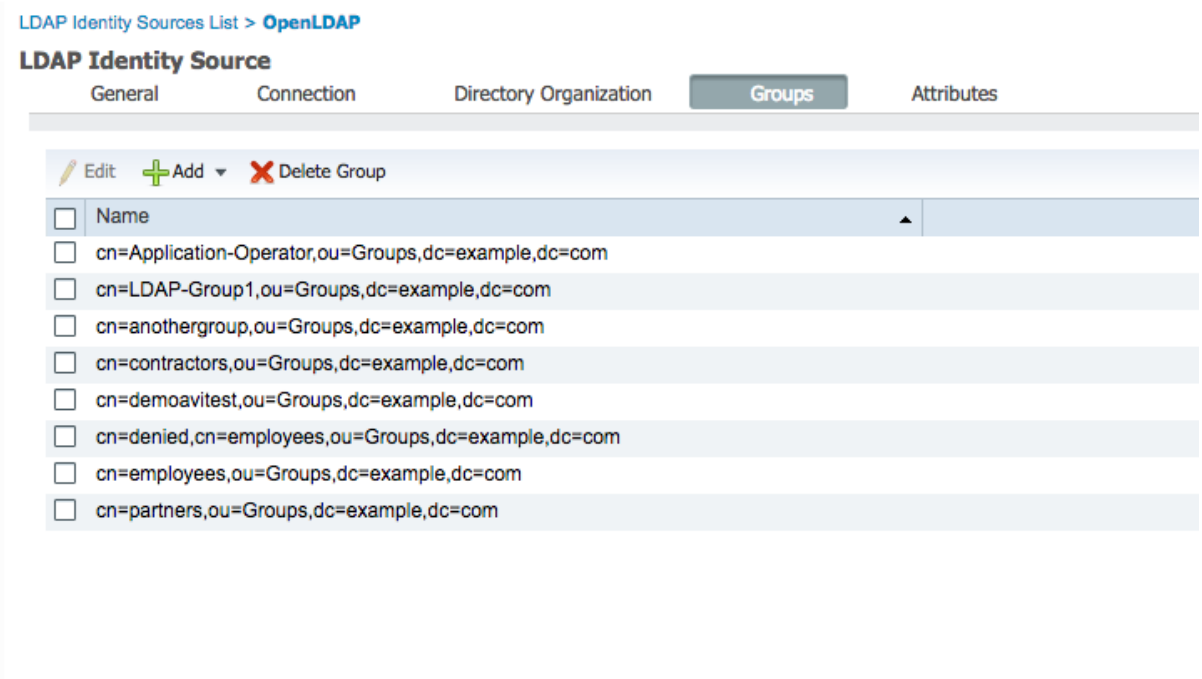

Identity Services Engine
Home
Operations
Policy
Guest
Authentication
Authorization
Profiling
Posture
Client Provisioning
Policy Elen
Authentication Policy
Define the Authentication Policy by selecting the protocols that ISE should use to communicate v
For Policy Export go to Administration > System > Backup & Restore > Policy Export Page
Policy Type
Simple
Rule-Based
Network Access Service
Allowed Protocol : Default Device A...
Identity Source
OpenLDAP
Options
If authentication failed
Reject
If user not found
Reject
If process failed
Drop


Identity Services Engine
Home
Operations
Policy
Guest Access
Administration
Work Centers
System
Identity Management
Network Resources
Device Portal Management
pxGrid Services
Feed Service
Identity Mapping
Identities
Groups
External Identity Sources
Identity Source Sequences
Settings
External Identity Sources
Certificate Authentication Profile
Active Directory
LDAP
OpenLDAP
RADIUS Token
RSA SecurID
SAML Id Providers
LDAP Identity Sources List > OpenLDAP
LDAP Identity Source
General
Connection
Directory Organization
Groups
Attributes
* Name
OpenLDAP
Description
Schema
Custom
* Subject Objectclass
inetOrgPerson
* Group Objectclass
posixGroup
* Subject Name Attribute
uid
* Group Map Attribute
memberUid
Certificate Attribute
Subject Objects Contain Reference To Groups
Group Objects Contain Reference To Subjects
Subjects In Groups Are Stored In Member Attribute As
Username

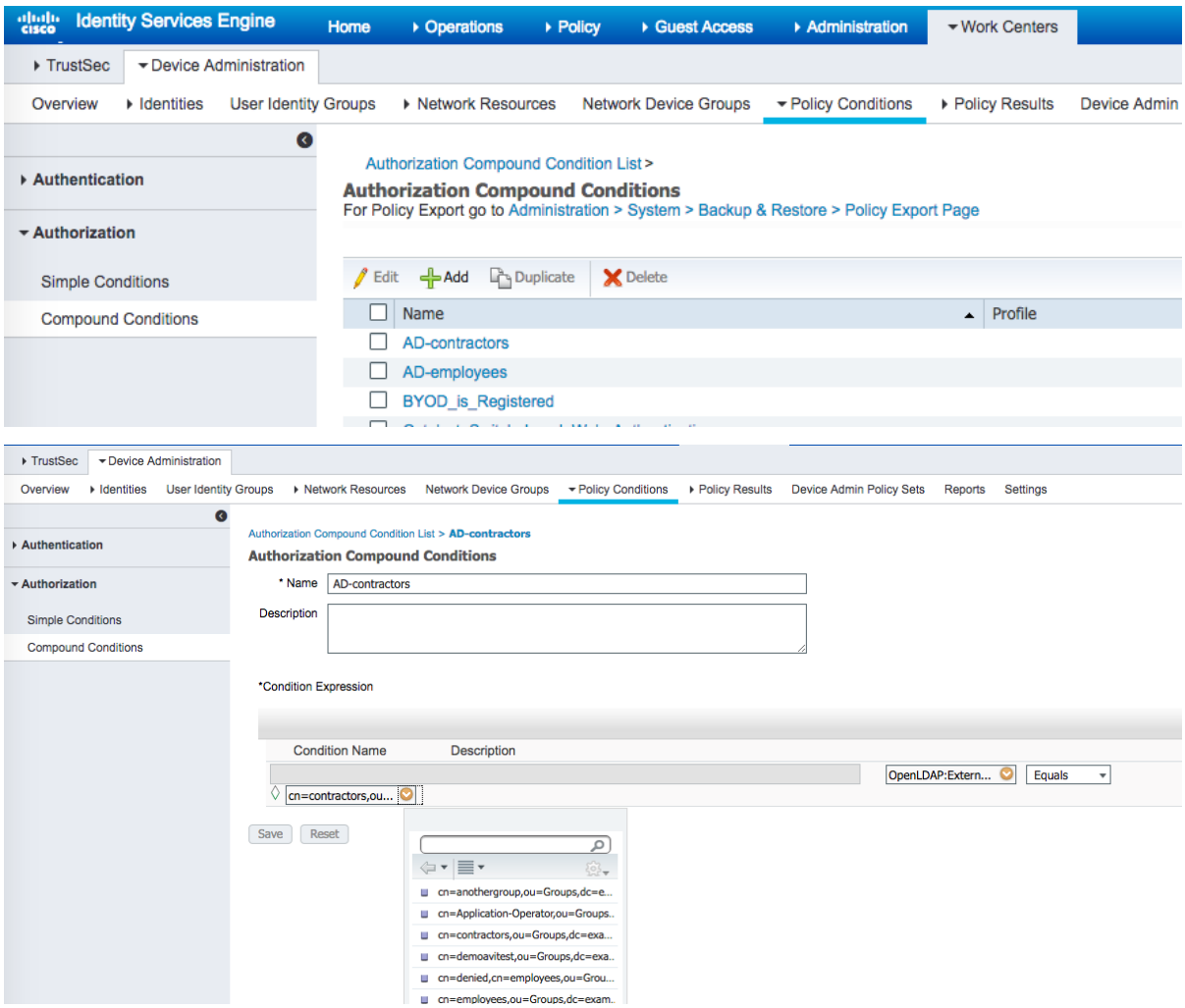

LDAP Identity Sources List > OpenLDAP
LDAP Identity Source
General
Connection
Directory Organization
Groups
Attributes
Primary Server
Secondary Server
Enable Secondary Server
* Hostname/IP
10.10.23.120
Hostname/IP
* Port
389
Port
389
Access
Anonymous Access
Authenticated Access
Access
Anonymous Access
Authenticated Access
Admin DN
cn=admin,dc=example,dc=com
Admin DN
Password
Password
Secure Authentication
Enable Secure Authentication
Enable Server Identity Check
Secure Authentication
Enable Secure Authentication
Enable Server Identity Check
LDAP Server Root CA
Thawte Primary Root CA
LDAP Server Root CA
Thawte Primary Root CA
Issuer CA of ISE
Select if required (optional)
Issuer CA of ISE
Select if required (optional)



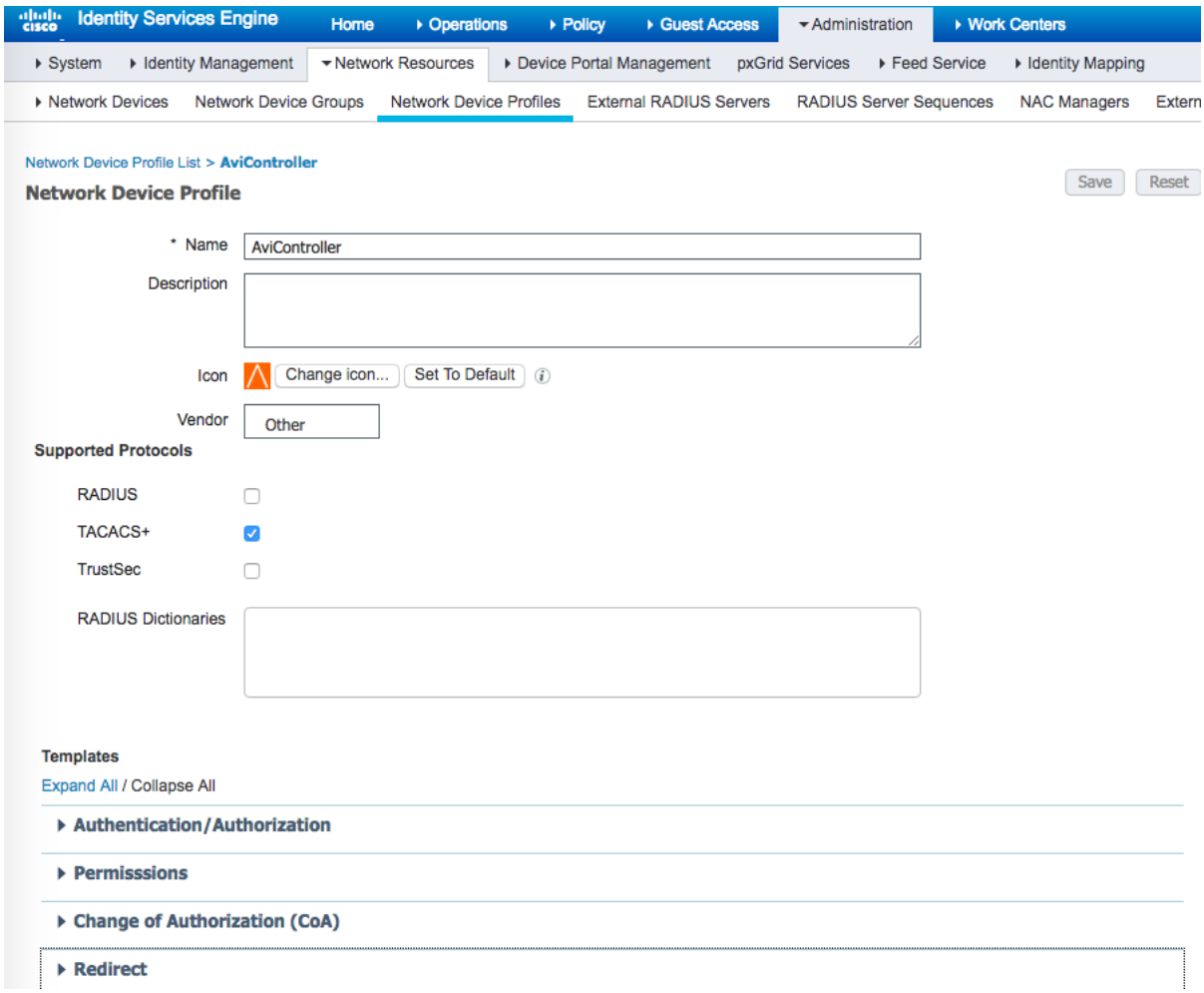
- The ISE LDAP settings used to fetch LDAP groups to use them for Authorization conditions are as shown below:

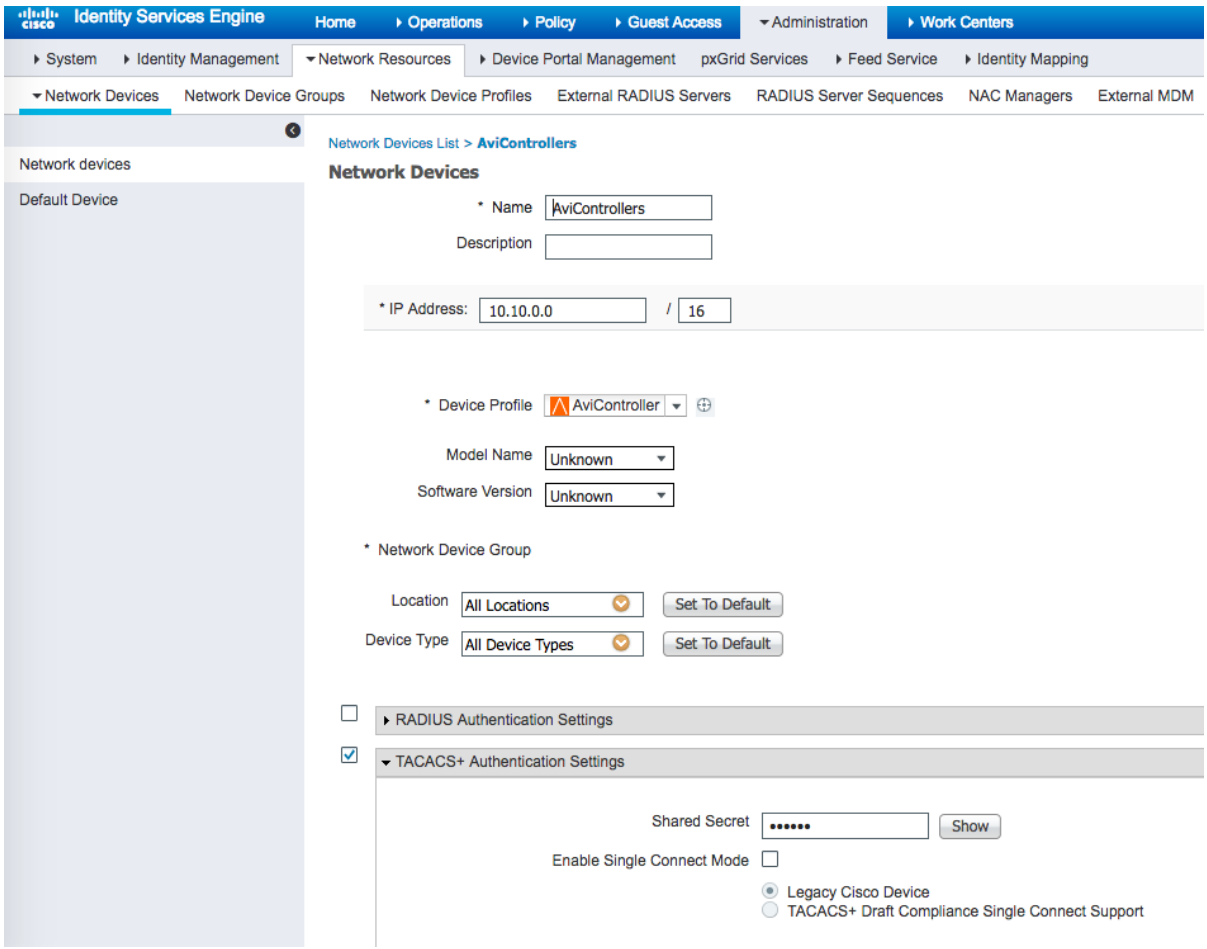


- ISE Authorization conditions added for Users in the AD groups.

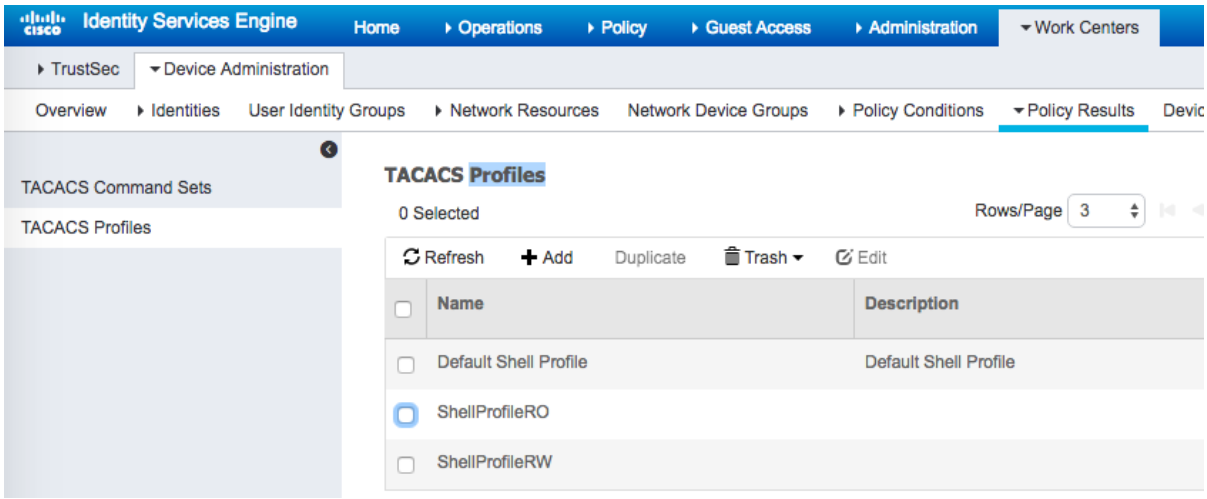


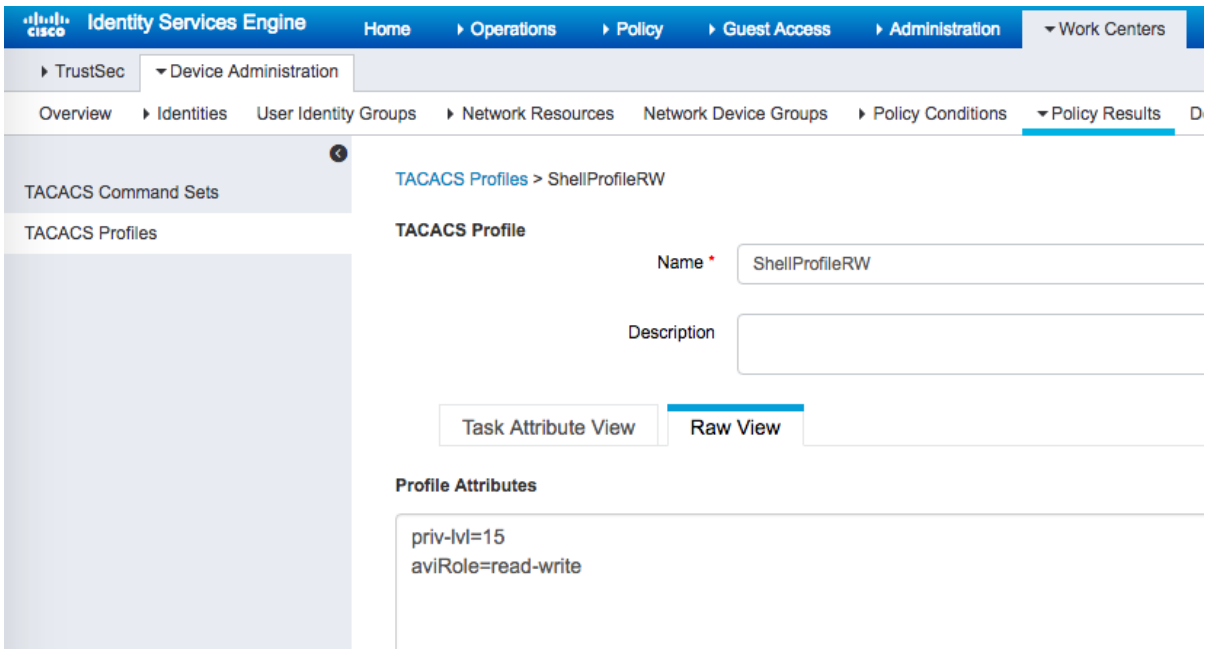
- ISE server recognizes all NSX Advanced Load Balancer Controller cluster nodes as valid Network Devices.


Identity Services Engine
Home
Operations
Policy
Guest Access
Administration
Work Centers
System
Identity Management
Network Resources
Device Portal Management
pxGrid Services
Feed Service
Identity Mapping
Network Devices
Network Device Groups
Network Device Profiles
External RADIUS Servers
RADIUS Server Sequences
NAC Managers
Extern
Network Device Profile List > AviController
Network Device Profile
Save
Reset
* Name
AviController
Description
Icon
Change icon...
Set To Default
Vendor
Other
Supported Protocols
RADIUS
TACACS+
TrustSec
RADIUS Dictionaries
Templates
Expand All / Collapse All
Authentication/Authorization
Permisssions
Change of Authorization (CoA)
Redirect

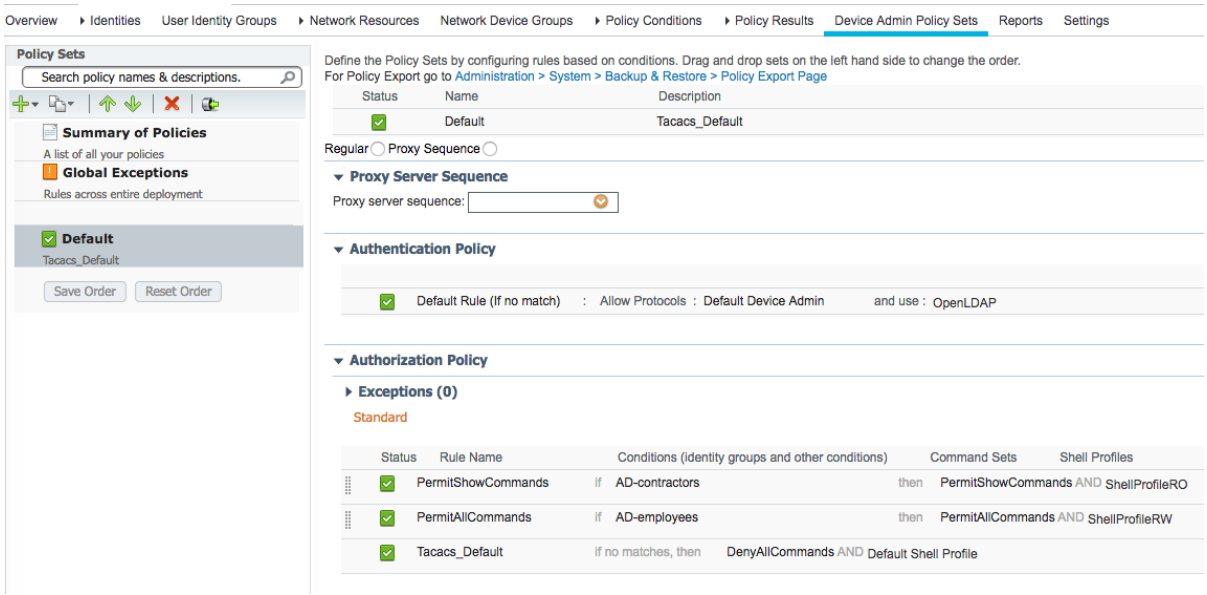


- Configuring ISE requires shell profiles and TACACS+ profiles.



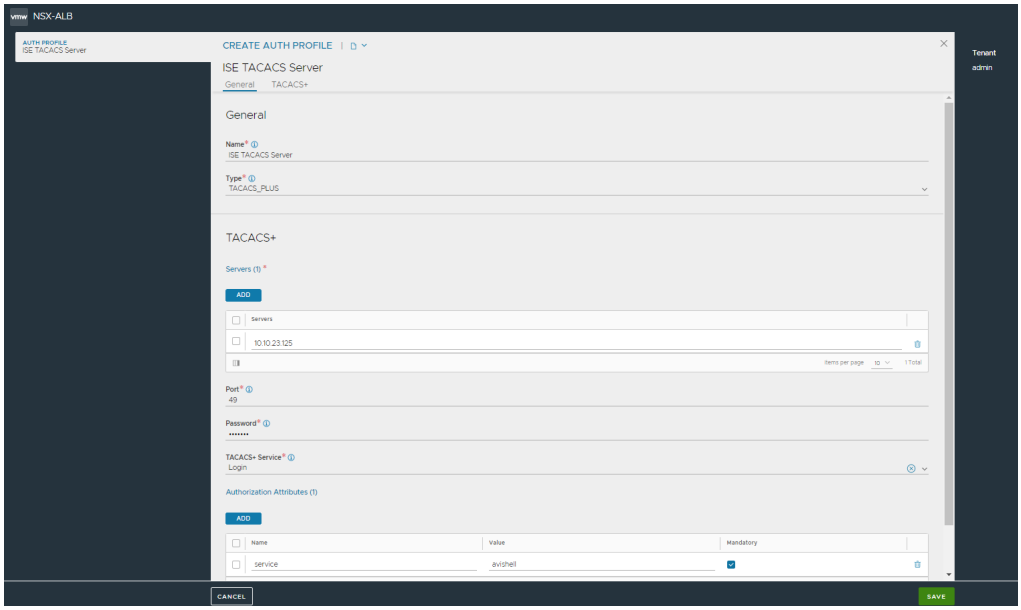


- ISE device policy sets default condition updated to assign different shell profiles based on group membership.



- The NSX Advanced Load Balancer TACACS+ auth profile must be configured with the same shared secret that was assigned to the device in ISE. The “service” attribute is generally required to identify and authorize a NSX Advanced Load Balancer user. Authorization attributes from a TACACS+ server can be used to map NSX Advanced Load Balancer users to various roles and tenants.

  In the case of an ACS server, service=avishell is required for user authorization; while in the case of an ISE server, service=avishell is known to cause authorization failure.



- NSX Advanced Load Balancer TACACS+ authorization role and tenant mapping configured to assign different roles based on TACACS+ attribute value.

Shrubbery TAC_PLUS

- TAC_PLUS server is a much simpler alternative to ISE/ACS. This is relevant in development or testing environments. Conceptually, users are assigned to groups and groups have request and response attributes.

```
key = xxxxxxx

group = netadmin {
        default service = permit
        login = file /etc/passwd
        service = exec {
                priv-lvl = 15
        }
}

group = admin {
        default service = permit
}

group = jenkinsattrs {
        default service = permit
        service = jenkins {
                avirole = Tacacs-Admin
                avitenant = Tacacs-Tenant1
        }
}

group = jenkinsunknown {
        default service = permit
        service = jenkins {
                avirole = "Unknown Role"
                avitenant = "Unknown Tenant"
        }
}

group = jenkinsnoattrs {
        default service = permit
        service = jenkins {
        }
}

user = aviuser {
        member = netadmin
}

user = jenkinstest1 {
        login = cleartext "password"
        member = jenkinsattrs
}

user = jenkinstest2 {
        login = cleartext "password"
        member = jenkinsattrs
}
```

```
[root@localhost ~]# cat /etc/systemd/system/tac_plus.service
[Unit]
Description=TACACS+ Service
After=syslog.target

[Service]
Type=simple
ExecStart=/usr/local/sbin/tac_plus -C /etc/tac_plus/tac_plus.conf -L -p 49 -d 65535 -Gt -l /var/log/tac_plus.log
KillMode=process
Restart=always
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

The NSX Advanced Load Balancer TACACS+ auth profile can be configured in the same way as an ISE or ACS.

Enabling SAML Authentication in NSX Advanced Load Balancer

After creating a SAML profile in NSX Advanced Load Balancer and a SAML catalog item in Workspace One Access, enable SAML and grant superuser rights to SAML users.

Note You can configure granular role-based access control by adding application parameters into the Workspace One Access catalog item and then mapping those parameters to different roles in NSX Advanced Load Balancer. For more information, see Tenant and Role Mapping Examples.

To enable SAML and map user roles, follow the steps below.

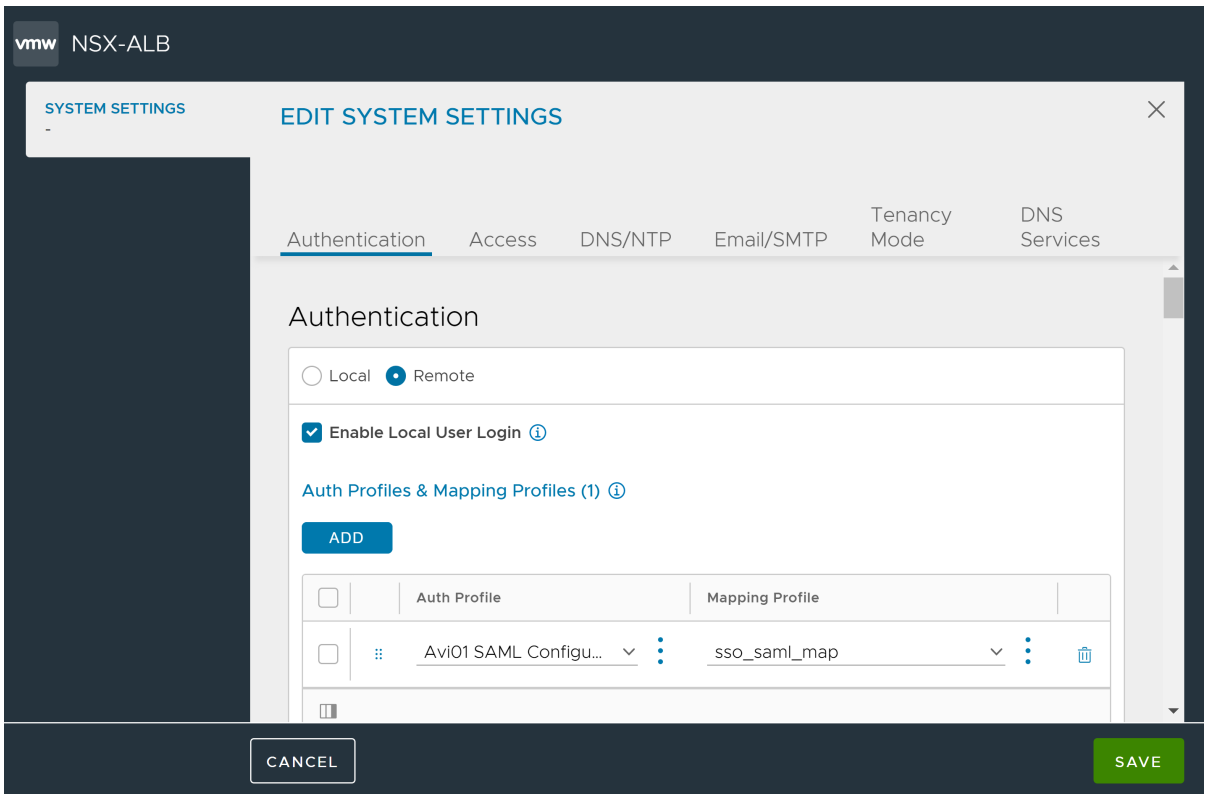
1 Log in to the NSX Advanced Load Balancer Controller with admin credentials.

2 Navigate to **Administration > System Settings** and click **Edit**.

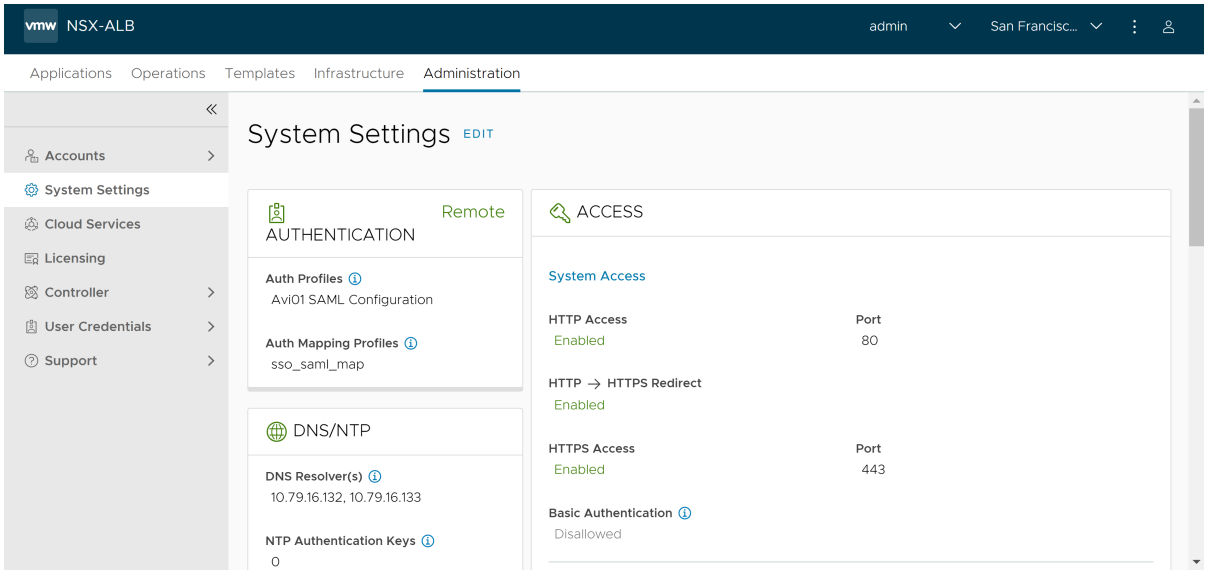
3 Under **Authentication**, select **Remote**.

4 Select the **Enable Local User Login** option. If this option is not selected, and there is a configuration issue, you will not be able to log back into the Controller.

5 Select the **Auth Profile** created with SAML as the **Type** and select the required Mapping Profile.



6 Click **Save**.



SAML authentication is now configured on the NSX Advanced Load Balancer Controller.

SAML Support for NSX Advanced Load Balancer SDK

NSX Advanced Load Balancer supports SDKs to use IdP credentials for it and a REST API login. SAML authentication profile be set up on the NSX Advanced Load Balancer Controller to be used by the Python SDK to establish a connection and access resources.

Note

- Logging into the NSX Advanced Load Balancer CLI using IdP credentials is not yet supported.
- SAML-based authentication using the Python SDK is supported for Okta and OneLogin.
- The service provider (SP) never directly interacts with the identity provider. A browser or the Python SDK acts as the agent to carry out all redirections.
- The service provider needs to know to which identity provider to redirect before it has any idea who the user is.
- The service provider does not know who the user is until the SAML assertion comes back from the identity provider.
- SAML authentication flow is asynchronous. The SP does not know if the IdP will ever complete the entire flow. Owing to this, the SP does not maintain any state of any authentication requests generated. When the SP receives a response from an IdP, the response must contain all necessary information.

For more information, see *SAML Authentication for Single Sign-On* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

SAML Python SDK

Under the SDK, a file named `saml_avi_api.py` contains the IdP class definition for each supported IdP. IdP-specific classes are inherited from the `ApiSession` base class. An IdP-specific class definition has its own authentication method to be called to authenticate a given user. URL redirection and SAML assertion are handled in this class. This class returns the Controller session after successful authentication from the given IdP.

Example of Okta:

In this collection of code snippets, the `OktaSAMLApiSession` class is used to authenticate a user for Okta IdP, get the Controller session, and create the VS. From `avi.sdk.saml_avi_api import OktaSAMLApiSession`:

Create NSX Advanced Load Balancer API Session

```
api = OktaSAMLApiSession("10.10.10.42", "okta_username", "okta_password")
```

OR

```
api = ApiSession.get_session("controller_ip", username="foo", password="foo",
idp=OktaSAMLApiSession)
```

Create VS Using Pool sample_pool

```
pool_obj = api.get_object_by_name('pool', 'sample_pool')
pool_ref = api.get_obj_ref(pool_obj)
services_obj = [{'port': 80, 'enable_ssl': False}]
vs_obj = {'name': 'sample_vs', 'ip_address': {'addr': '11.11.11.42', 'type': 'V4'},
          'services': services_obj, 'pool_ref': pool_ref}
resp = api.post('virtualservice', data=vs_obj)
```

Print List of all Virtual Services

```
resp = api.get('virtualservice')
for vs in resp.json()['results']:
    print vs['name']
```

Delete a Virtual Service

```
resp = api.delete_by_name('virtualservice', 'sample_vs')
```

OneLogin Example

In this collection of code snippets, the `OneloginSAMLApiSession` class is used to authenticate a user for OneLogin IdP, get the Controller session, and create the virtual service.

From `avi.sdk.saml_avi_api import OneloginSAMLApiSession`

Create NSX Advanced Load Balancer API Session

```
api = OneloginSAMLApiSession("10.10.10.42", "onelogin_username", "onelogin_password")
```

OR

```
api = ApiSession.get_session("controller_ip", username="foo", password="foo",
idp=OneloginSAMLApiSession)
```

Create VS Using Pool sample_pool

```
pool_obj = api.get_object_by_name('pool', 'sample_pool')
pool_ref = api.get_obj_ref(pool_obj)
services_obj = [{'port': 80, 'enable_ssl': False}]
vs_obj = {'name': 'sample_vs', 'ip_address': {'addr': '11.11.11.42', 'type': 'V4'},
          'services': services_obj, 'pool_ref': pool_ref}
resp = api.post('virtualservice', data=vs_obj)
```

Print List of all Virtual Services

```
resp = api.get('virtualservice')
for vs in resp.json()['results']:
    print vs['name']
```

Delete a Virtual Service

```
resp = api.delete_by_name('virtualservice', 'sample_vs')
```

NSX Advanced Load Balancer CLI Access to a SAML-Authenticated NSX Advanced Load Balancer Controller

NSX Advanced Load Balancer supports single sign-on (SSO) to the Avi Controller's UI using Security Assertion Markup Language (SAML). However, during debugging or even normal day-to-day operations, accessing the Avi Controller's CLI is often needed using SSH. SAML credentials cannot be used to log into the CLI.

To access the NSX Advanced Load Balancer Controller through SSH, a registered user must have a valid token. Once a token is created, one can initiate an SSH connection to the Controller using CLI as the SSH user. A CLI shell will be created. Once the shell has been created, a login prompt will be presented. Provide the required username and the token as the password.

Token Authentication

For debugging or even normal day-to-day operations, accessing the Controller's CLI is often needed using SSH. To access the NSX Advanced Load Balancer Controller through SSH, the system admin or a registered user must have a valid token. Note that this token is not the same as the Oauth token. It is an alternative to a password. The system admin can create a temporary token for a user to access resources for a few hours. After that time expires, the token will not work and the user loses access, so there is no need to delete the token. Once a token is created, you can initiate an SSH connection to the Controller using CLI as the SSH user. A CLI shell is created. Once the shell has been created, a login prompt will be displayed. Provide the required username and the token as the password.

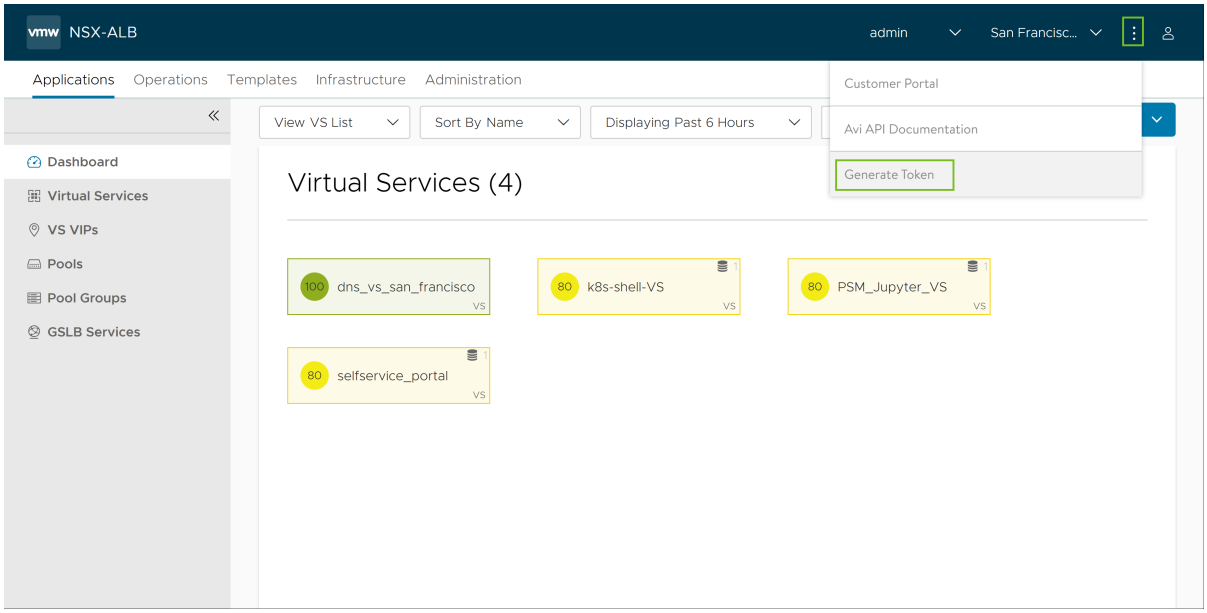
To generate the SSO token through the UI, perform the following steps.

Procedure

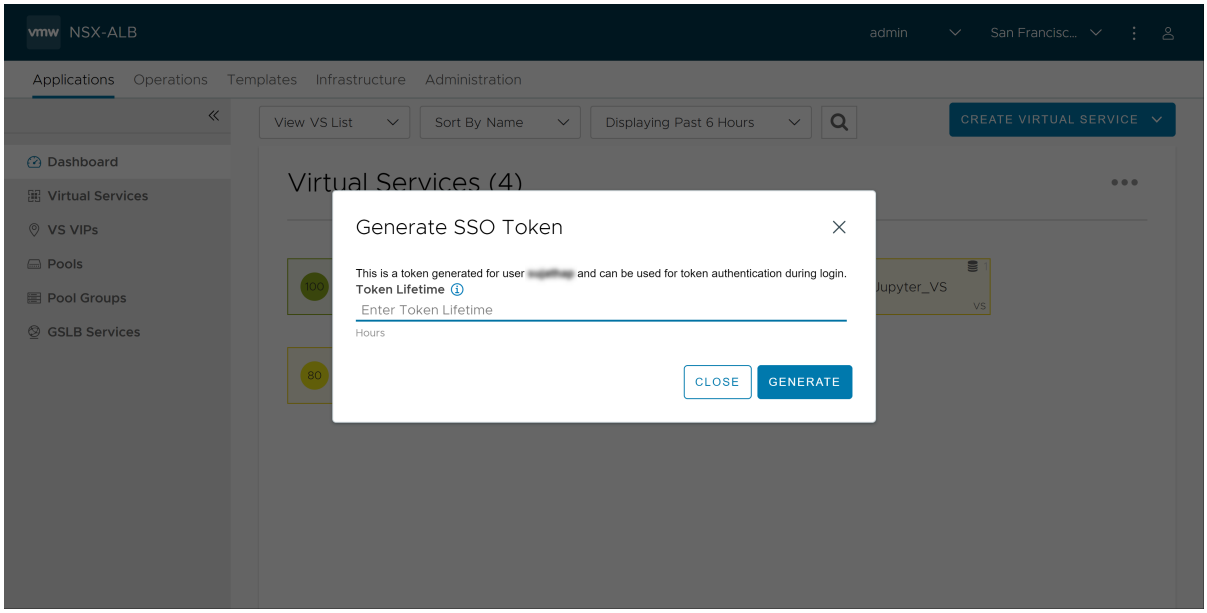
1 Login to the NSX Advanced Load Balancer UI.

2 Click the three dots in the top right corner of the dashboard.

3 Select **Generate Token** from the drop down menu.



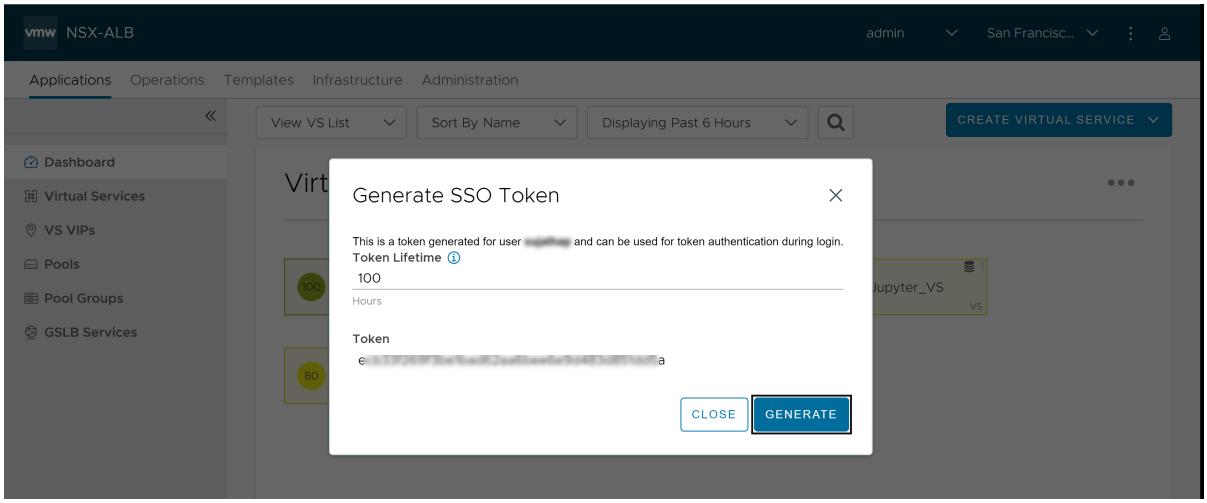
A pop-up screen appears as shown below.



4 Enter the **Token Lifetime** for the token's validity in hours and click **Generate**.

Note

- To generate a single use token, enter 0.
- The maximum value that can be entered in this field is 87600 hours.
- In case another token is generated before the first one expires, the first token still remains valid.



5 Copy the generated token for CLI/SSH access.

6 To delete a token, for example, if you want to decommission an active token, use the DELETE command in the CLI.

Access the CLI Using the Token

This section explains the steps to access CLI/SSH using the generated SSO token.

Procedure

1 Open an SSH or Putty session to `cli@.saas.avinetworks.com</code>`.

2 Login with your username.

3 Paste the token that was generated using the CLI as the Password.

Results

You have now successfully logged into your Controller using your account or Service Account.

Configure SAML with Workspace One for NSX Advanced Load Balancer

SAML enables integration into VMware Workspace ONE (WS1) and takes advantage of the App Catalog, network access restrictions, and step-up authentication when administrators sign in.

In this section, you will understand how to configure SAML for NSX Advanced Load Balancer control plane authentication with WS1 access being the IDP.

Note You can configure granular role-based access control by adding application parameters into the Workspace One Access catalog item and then mapping those parameters to different roles in NSX Advanced Load Balancer. For more information, see Tenant and Role Mapping Examples.

Prerequisites

Before initiating the configuration, complete the following prerequisites:

- Configure a DNS record for the NSX Advanced Load Balancer Controller. This will be used for the fully qualified domain name (FQDN) that is used when signing into the system.
- Get the Workspace ONE Access IDP metadata.

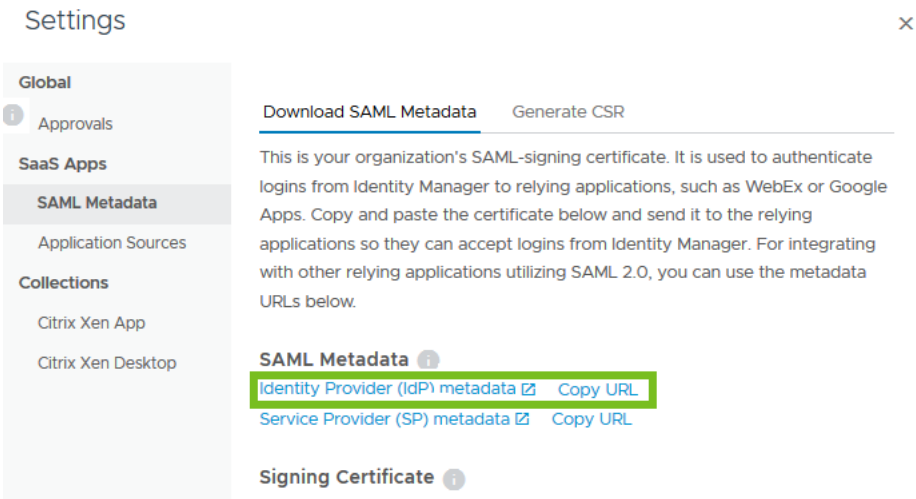
Follow the steps below to download the `idp.xml` file:

1 Log in to the Workspace ONE Access administrator console.

2 Navigate to the **Catalog > Settings**.

3 Click **SAML Metadata** under **SaaS Apps**.

4 In the **Download SAML Metadata** tab, click **Copy URL** next to **Identity Provider (IdP) metadata**.



SAML Configuration in NSX Advanced Load Balancer

To configure an authentication profile to support SAML on the NSX Advanced Load Balancer Controller, follow the below:

1 Log in to the NSX Advanced Load Balancer Controller with admin credentials.

2 Navigate to the **Templates > Security > Auth Profile** and click **CREATE**.

3 Enter the **Name** of the auth profile.

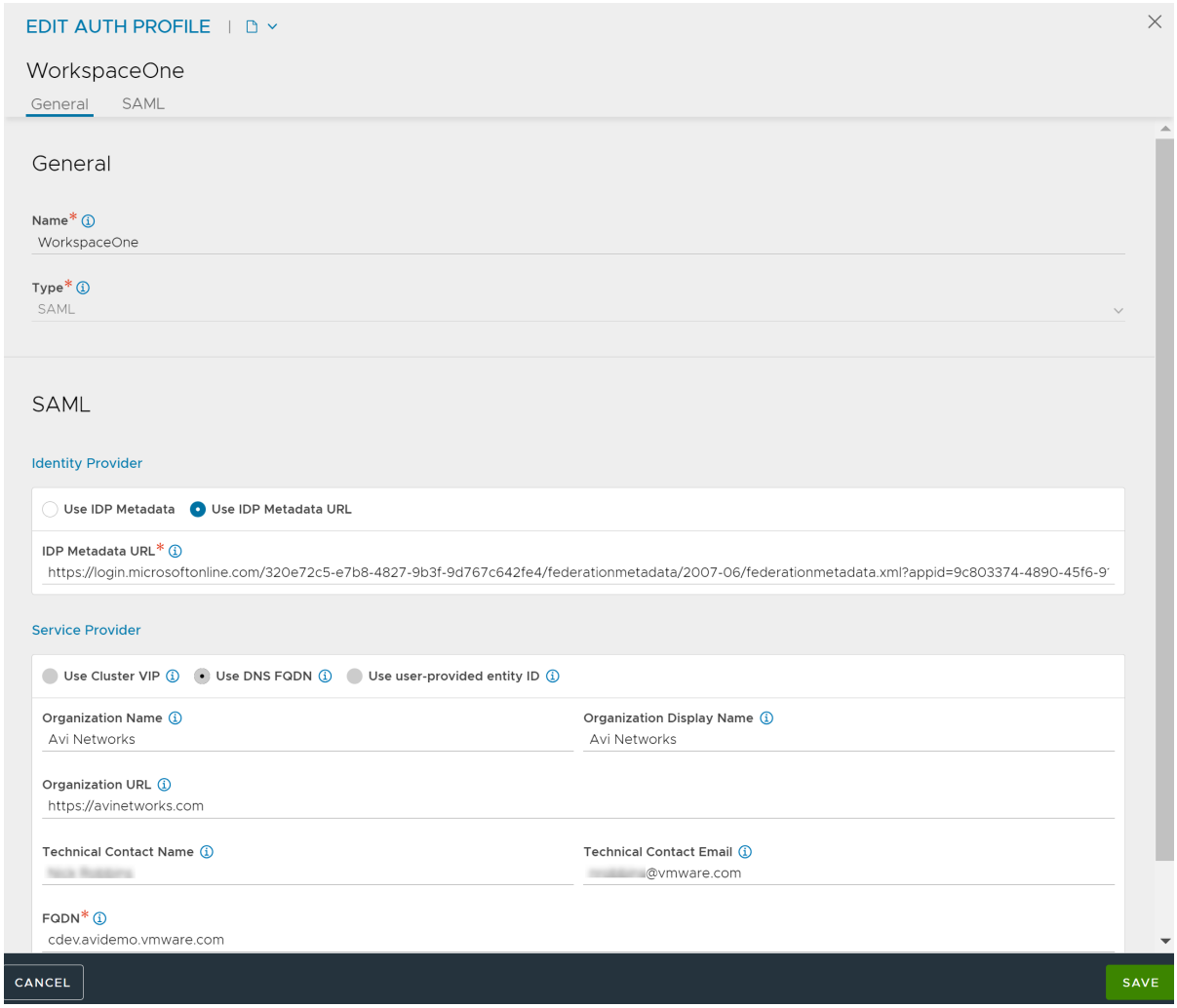
4 Select **SAML** as the **Type** of auth profile.

5 Select **Use IDP Metadata URL** option and paste the URL in the**Enter IDP Metadata URL** field.

6 Under **Service Provider**, select **Use DNS FQDN**.

7 Enter the service provider organization details, as required.

8 Enter the **FQDN** to be used for the SAML configuration.



9 Click **Save**.

Collecting Service Provider Metadata

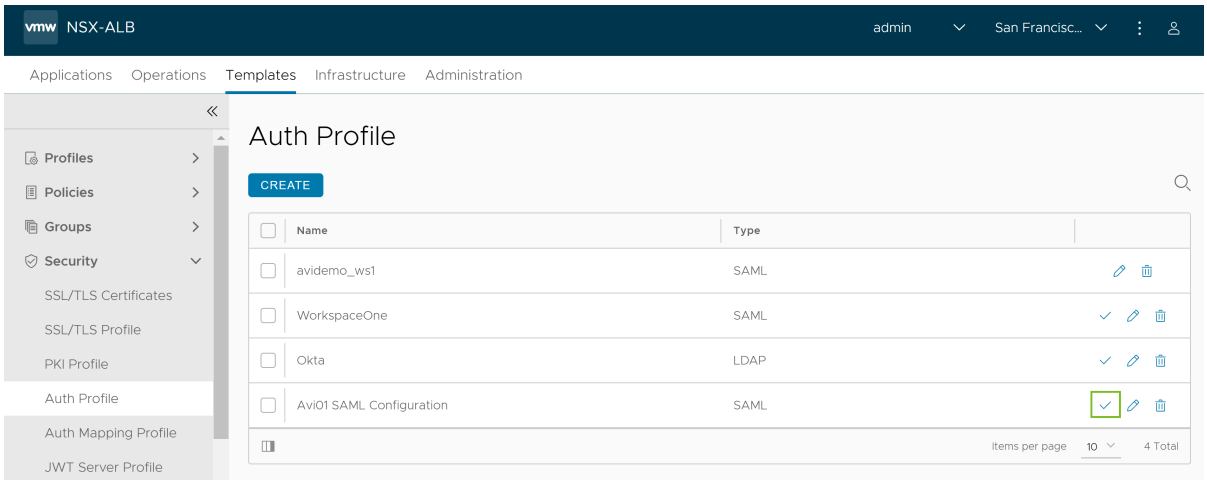
NSX Advanced Load Balancer does not generate an xml file that can be imported into Workspace ONE Access. So, the metadata must be entered manually. Collect the following details:

- Entity ID
- SSO URL
- Signing Certificate

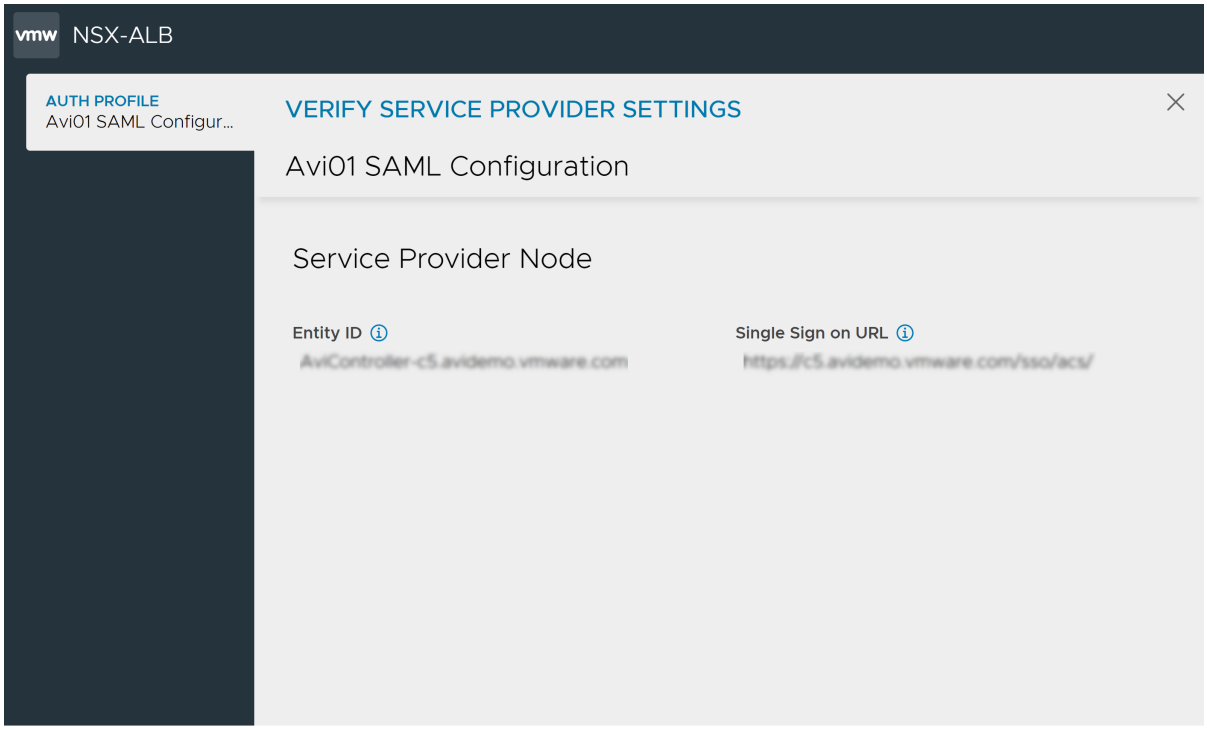
Get the entity ID and the SSO URL from the **VERIFY SERVICE PROVIDER SETTINGS** screen as shown below.

1 Navigate to **Templates > Security > Auth Profile**.

2 Identify the authentication profile created and click **verify** icon.



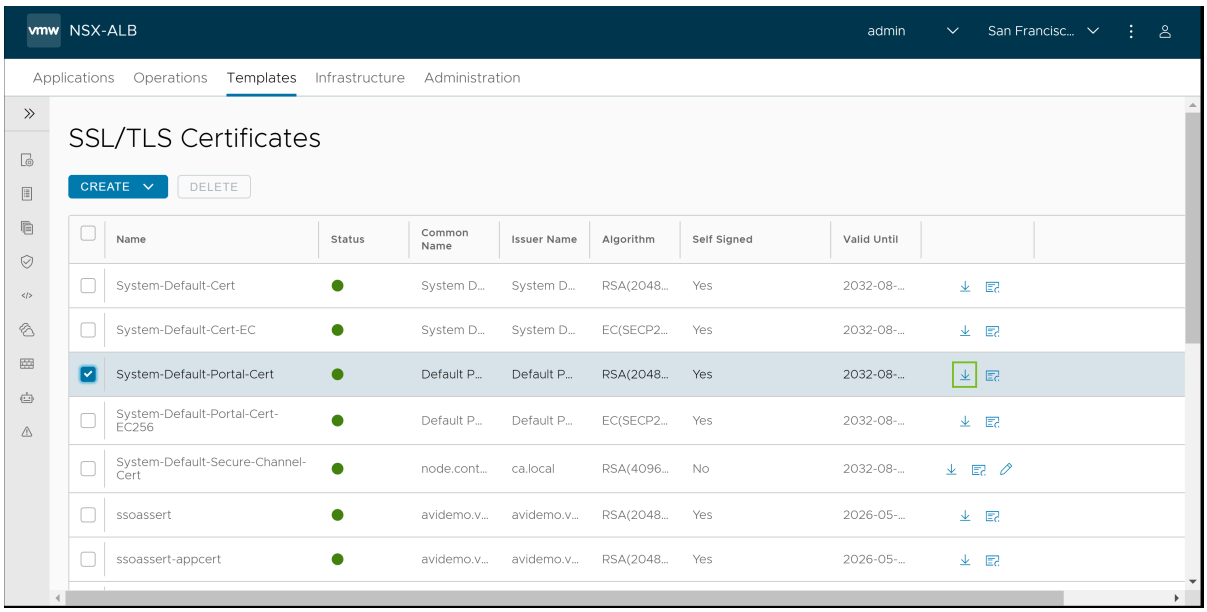
3 The **VERIFY SERVICE PROVIDER SETTINGS** screen displays the **Entity ID** and the **Single Sign on URL**. Copy this information and paste in a text editor.



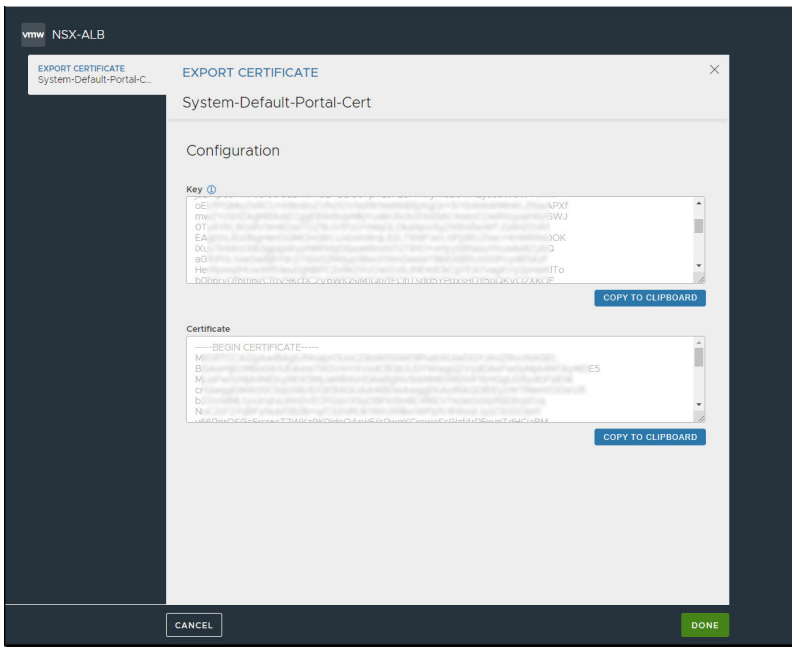
Get the signing certificate from SSL/TLS Certificates as shown below.

1 From the NSX Advanced Load Balancer UI, navigate to **Templates > Security > SSL/TLS Certificates**.

2 Identify the `System-Default-Portal-Cert` and click **Export** icon.



3 From the **Export Certificate** screen, click **Copy to clipboard** to copy the **Key** and the **Certificate**.



4 Paste the details into a text editor.

5 Click **DONE**.

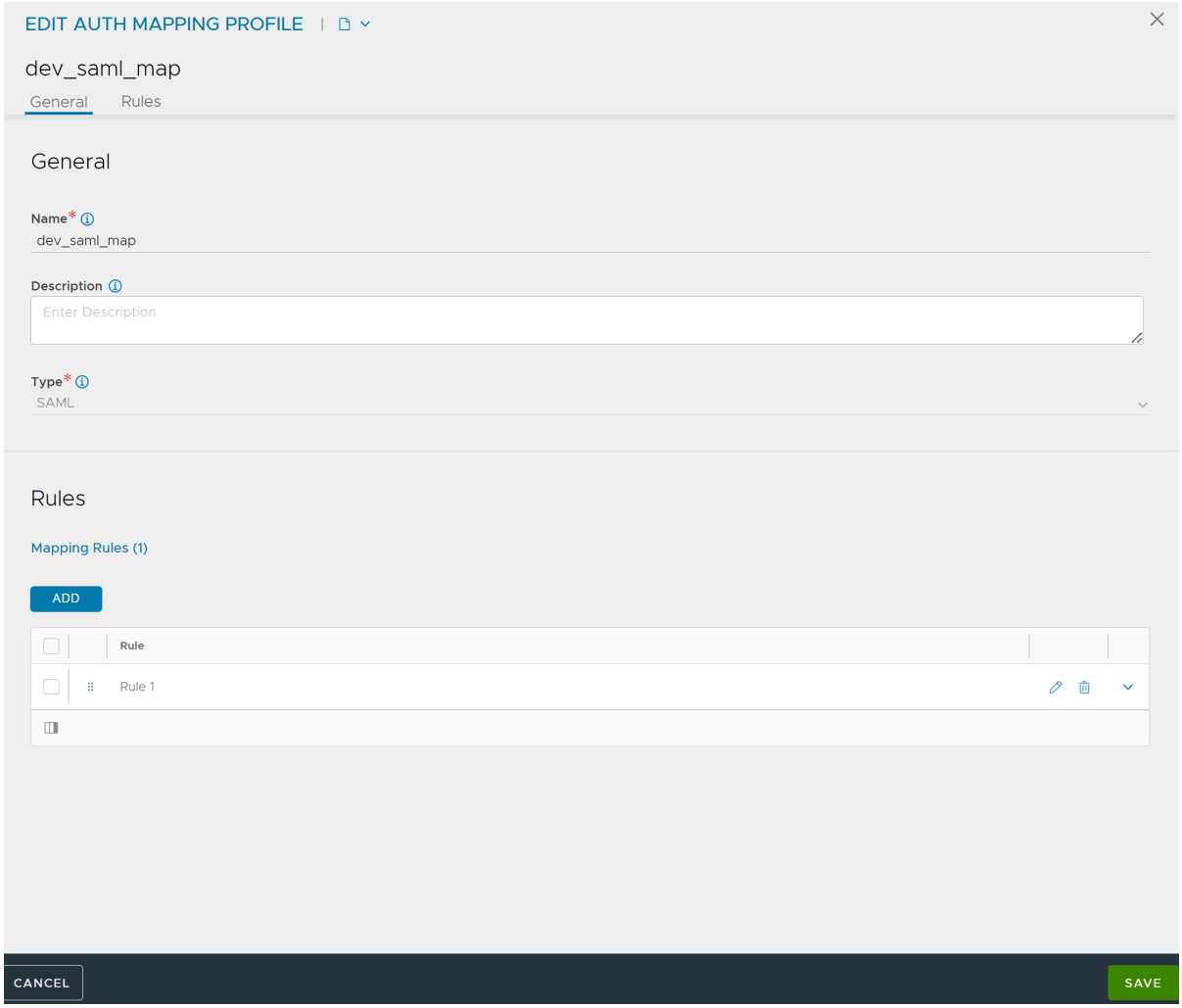
Configuring Auth Mapping Profile

1 From the NSX Advanced Load Balancer UI, navigate to **Templates > Security > Auth Mapping Profile**.

2 Click **CREATE** or edit an existing auth mapping rule.

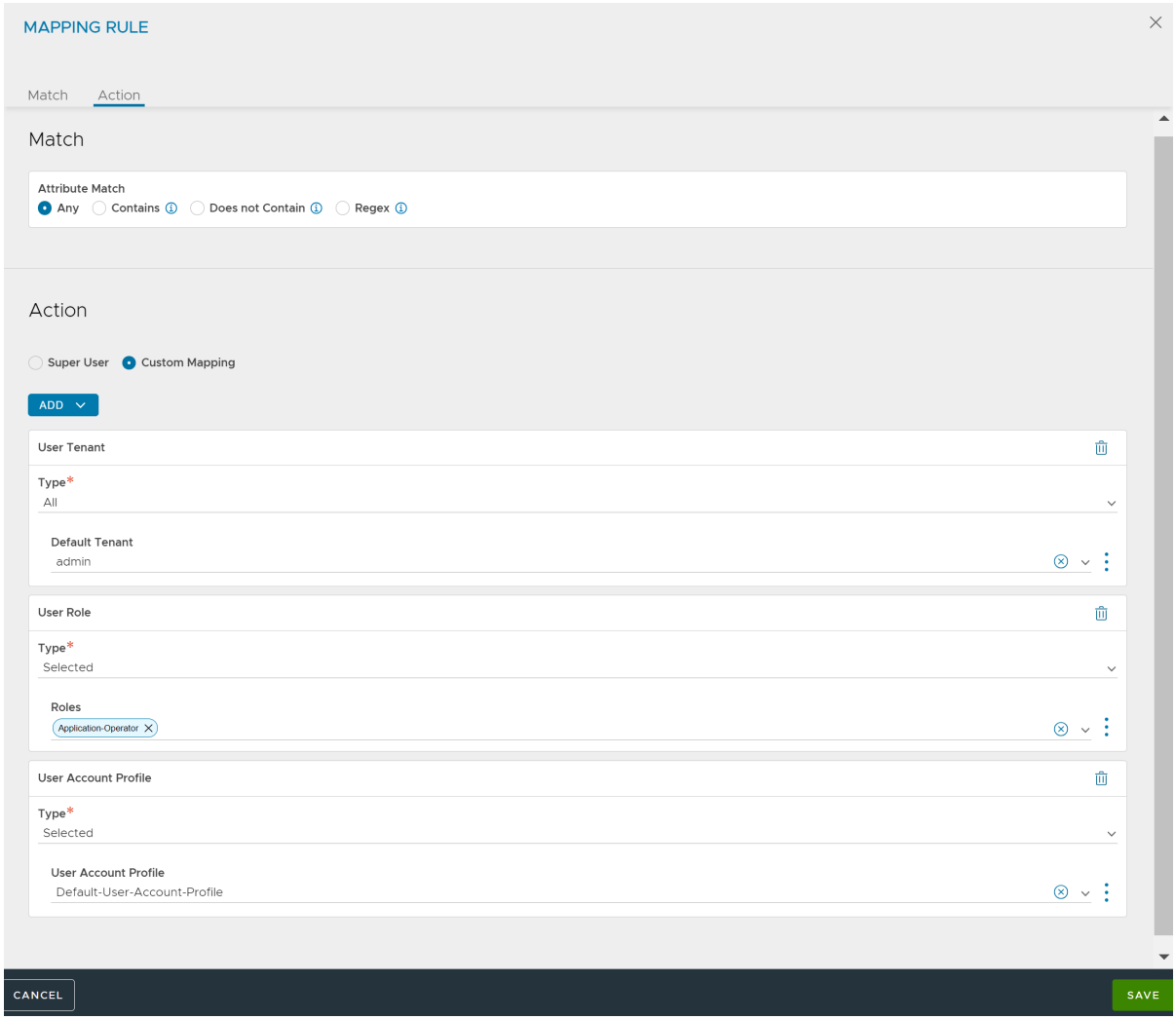
3 Enter **Name** and **Description**.

4 Select **SAML** as the **Type**.



Note Depending on the type selected, the auth profile settings are displayed.

5 Under **Rules**, Click **ADD**. A new **Mapping Rule** screen appears.



6 Under **Match**, select the filter for the **Attribute Match**.

   a **Any**: Users match regardless of attributes or their values.

   b **Contains**: The user must have the specified attribute, and the attribute must have one of the specified values.

   c **Does Not Contain**: The user must not have the specified attribute and value(s).

   d **Regex**:

7 Under **Action**, select **Custom Mapping**.

8 Select and configure **User Tenant**, **User Role**, or **User Account Profile** from the **ADD** drop-down menu as required. For more information, see Configuring Default Tenant for Remote Users, User Roles, and User Profile.

9 Click **Save**.

Configuring Remote Authentication

By default, a Controller will have only local authentication established (Authentication/ Authorization: Local).

To configure remote authentication using the NSX Advanced Load Balancer UI, follow the steps below:

1 Navigate to **Administration > System Settings > EDIT > Authentication**.

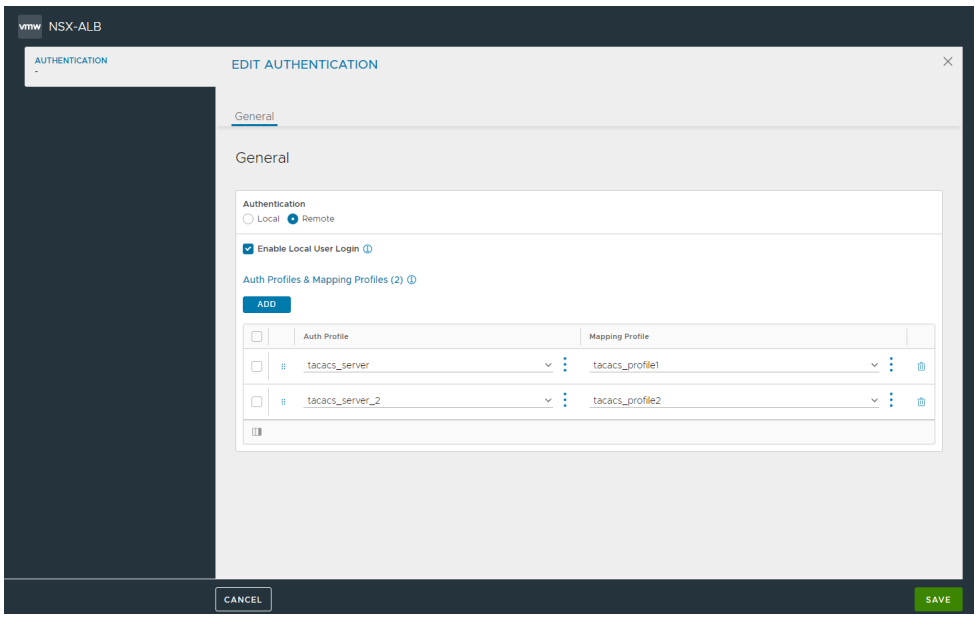
2 In the **Edit System Settings** screen, select **Remote** as the Authentication method.

3 Select**Enable Local User Login** to allow users from the local user database to log in with their user credentials.

4 Under **Auth Profiles & Mapping Profiles**, click **Add**.

5 From the **Select Auth Profile** drop-down menu, select the previously created remote Auth Profile.

6 From the **Select AUTH Mapping Profile** drop-down menu, select an existing **Mapping Profile** or click the vertical menu icon (three dots) to create a new **Mapping Profile**. For more information, see Configuring Auth Mapping Profile.



7 Click **Save**.

Note Tenant and role mapping are available only with remote authentication.

Configuring the NSX Advanced Load Balancer Catalog Item in Workspace ONE Access

Once the SAML profile is created in the NSX Advanced Load Balancer Controller, the Workspace ONE catalog entry can be created.

To create the Workspace ONE catalog entry, do the following:

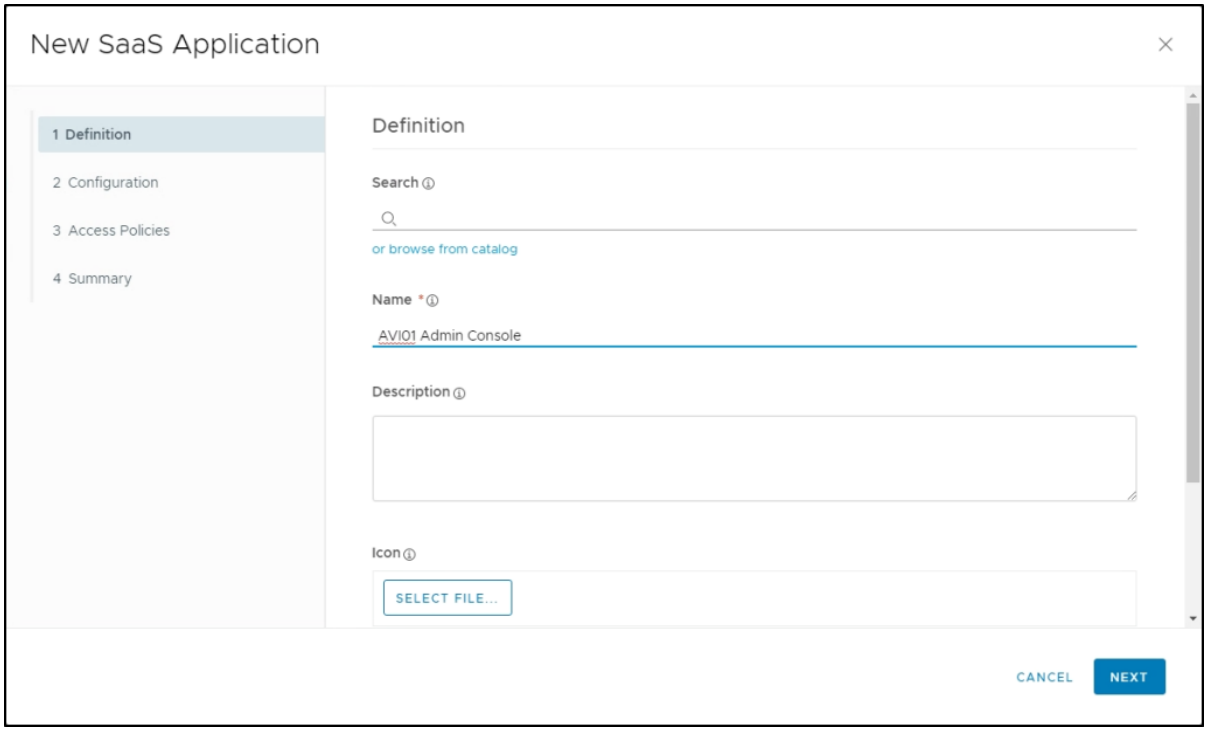
1 Log in to your Workspace ONE Access administrator console.

2 Navigate to the **Catalog** tab.

3 Click **New**.

4 In the **New SaaS Application** screen, enter a **Name** for the new NSX Advanced Load Balancer entry in the App Catalog.

5 Under the **Definition** tab, if you have an icon to use, click **Select File**, upload the icon for the application, and click **Next** to view the **Configuration** tab.

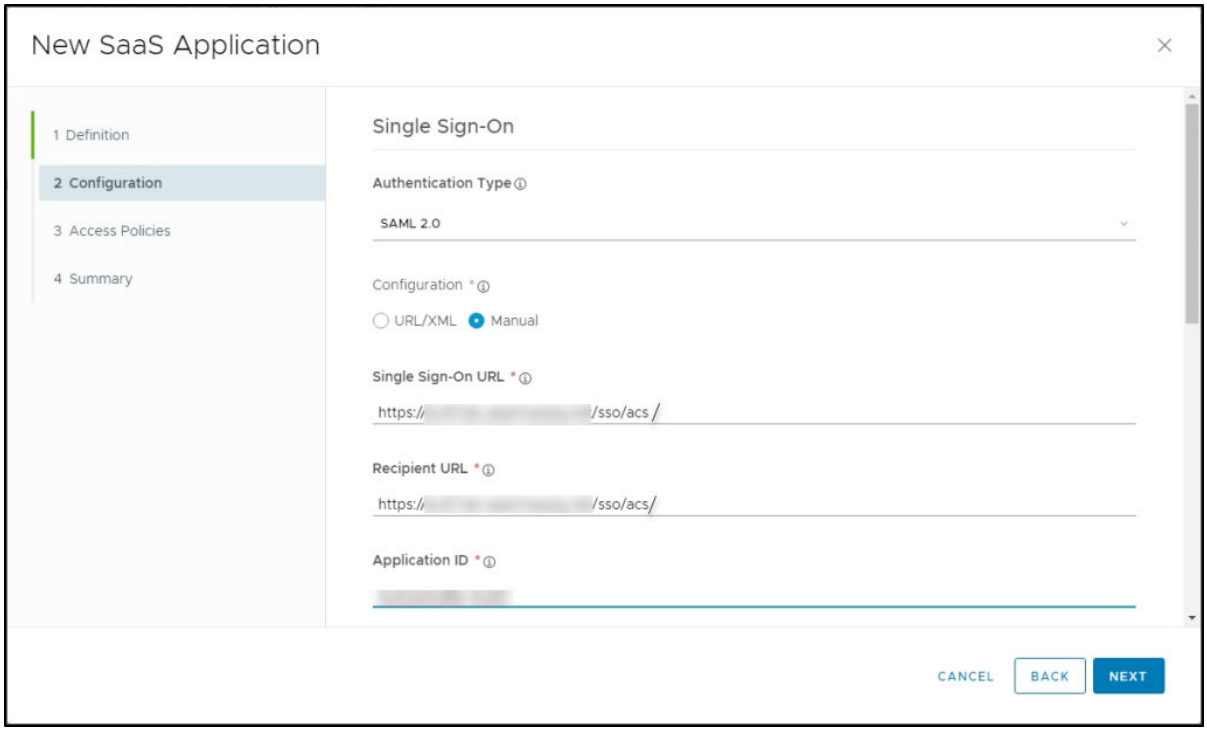


6 In the **Configuration** tab, enter the details as shown below:

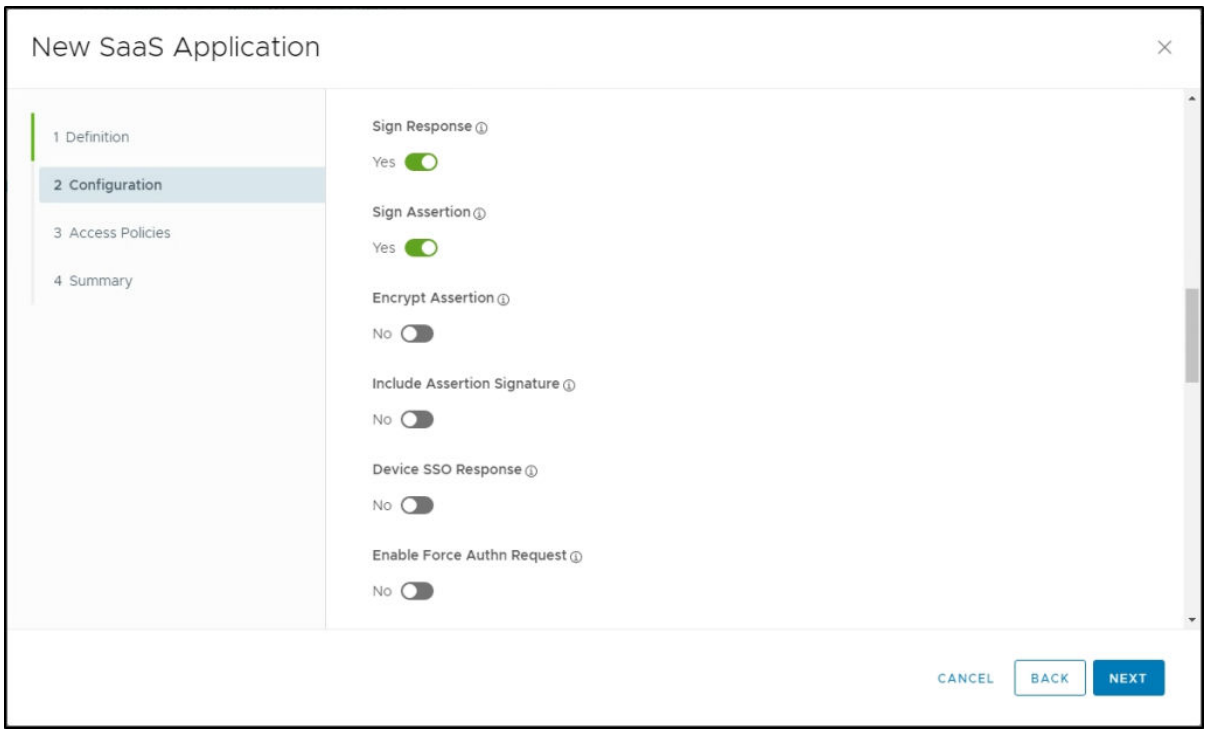
Field	Description
Single Sign-On URL	Use the **Single Sign on URL** copied from the **VERIFY SERVICE PROVIDER SETTINGS** screen in NSX Advanced Load Balancer. **Note** The trailing slash (/) after acs is mandatory.
Recipient URL	Use the **Single Sign-On URL**.
Application ID	Use the **Entity ID** copied from the **VERIFY SERVICE PROVIDER SETTINGS** screen in NSX Advanced Load Balancer.
Username Format	Unspecified.

Field	Description
Username Value	${user.email}.
Relay State URL	The FQDN or IP address of your appliance.

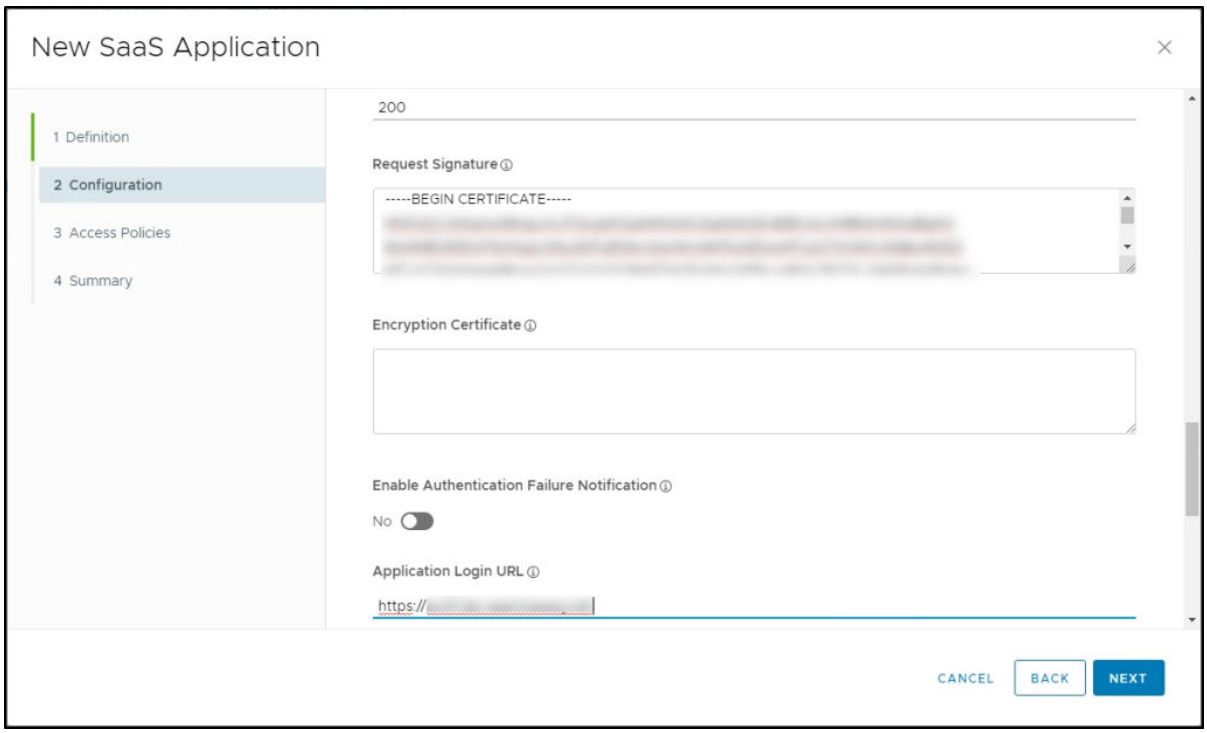
The **Configuration** tab in the **New SaaS Application** screen is as shown below.



7 Click **Advanced Properties** and configure the properties as shown below:

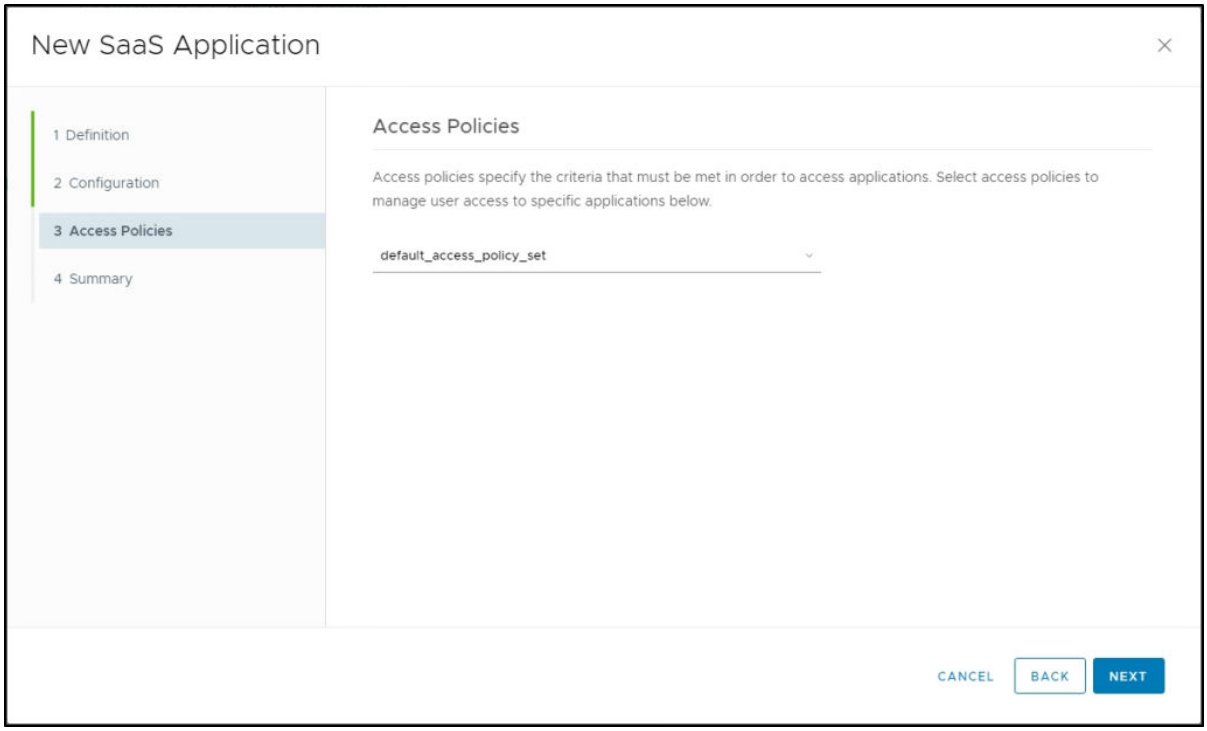


8 Copy the value of the **System-Default-Portal-Cert** certificate and paste it into the **Request Signature** field.



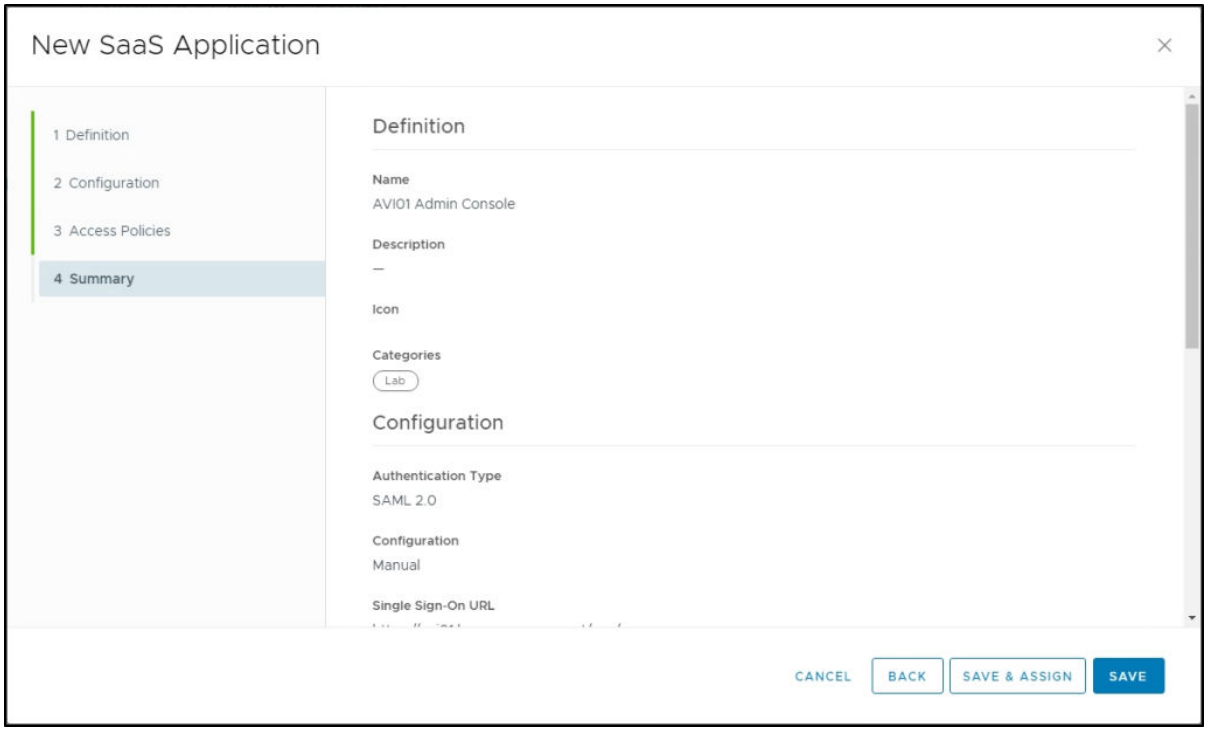
9 Enter the FQDN or IP address of the appliance as the **Application Login** URL. This enables SP-initiated login workflows.

10 Click Next to select **Access Policies** to use for this application. This determines the rules used for authentication and access to the application.



11 Click **Next** to review the **Summary** of the configuration.

12 Click **Save & Assign** and select the users or groups that will have access to this application and the deployment type.



13 Click **Save**.

NSX Advanced Load Balancer Go SDK Support

This section explains how to install, configure, and use the NSX Advanced Load Balancer Go SDK package for the various automation requirements.

NSX Advanced Load Balancer Go SDK is a Go Package that provides other APIs to communicate with the NSX Advanced Load Balancer REST APIs. NSX Advanced Load Balancer GO SDK package also provides utilities, which can be used to simplify and automate load balancing configurations on an NSX Advanced Load Balancer Controller. The following are the essential features for the NSX Advanced Load Balancer GO SDK package:

- Uses NSX Advanced Load Balancer session class and provides utilities to simplify integration with the NSX Advanced Load Balancer Controller.
- Helps in session authentication and keeps a cache of sessions to avoid multiple connections and tear downs across the different API sessions invocation.
- Automatically updates session cookies and CSRF Tokens from the NSX Advanced Load Balancer Controllers and provides helper APIs and templates for NSX Advanced Load Balancer Objects.
- X-AVI-TENANT (tenant) header handling and provides the sample source code for common load balancing examples.

SDK Directories

NSX Advanced Load Balancer Go SDK package is the source for the Go SDK for NSX Advanced Load Balancer Controller configuration. It is not required to download packages. Go will take care of package installations.

The following are the important NSX Advanced Load Balancer Go SDK directories:

1 examples: Go samples are in the `go/examples` directory. `create_vs.go` provides automation samples for the following configuration on NSX Advanced Load Balancer Controller:
   - Create session
   - Generic client to NSX Advanced Load Balancer Controller
   - Create a pool with one server, create an SSL virtual service, and fetch an object by name
2 clients: Contains NSX Advanced Load Balancer clients to establish a connection between the Go SDK and NSX Advanced Load Balancer Controller using the NSX Advanced Load Balancer session class. Each resource has its client to establish the connection.
3 session: Contains all the generic codes of the REST API calls for the NSX Advanced Load Balancer session and helper routines from REST API calls. It creates and maintains a session for the given resource.
4 models: Contains all models required to capture the API response. These are the structures to grab and store data of corresponding NSX Advanced Load Balancer REST API calls.

Prerequisites

Go Lang package must be installed before installing the NSX Advanced Load Balancer Go SDK package. To install the Go Lang package, see Go Lang Installation.

For more information on the Go programming language, see Go Lang Documentation.

Installing NSX Advanced Load Balancer Go SDK Package

The following is the syntax for installing NSX Advanced Load Balancer Go SDK package:

```
$ mkdir -p src/github.com/avinetworks/
$ cd src/github.com/avinetworks/
$ git clone https://github.com/avinetworks/sdk.git
#GOPATH will be path till src dir.
$ export GOPATH=~/src
```

Usage Examples

To create a session, a pool, and a basic virtual service (my-test-vs), execute `create_vs.go` file. Before executing this script, specify NSX Advanced Load Balancer Controller IP address, username, password, and tenant in `create_vs.go` file.

Importing NSX Advanced Load Balancer Sessions, Models, and Clients

```
package main

import (
    "github.com/avinetworks/sdk/go/clients"
    "github.com/avinetworks/sdk/go/models"
    "github.com/avinetworks/sdk/go/session"
    )
```

Creating NSX Advanced Load Balancer API Session

The sample syntax mentioned below creates a session with the following attributes:

- Client session IP address: 10.10.25.25
- Tenant: admin
- Session Type: Not secure

```
aviClient, err := clients.NewAviClient("10.10.25.25", "admin",
        session.SetPassword("something"),
        session.SetTenant("admin"),
        session.SetInsecure)
```

Creating a Pool

The syntax mentioned below creates a pool with the following attributes:

- Pool Name: my-test-pool

- Server IP Address: 10.90.20.12

```
pobj := models.Pool{}
pobj.Name = "my-test-pool"
serverobj := models.Server{}
serverobj.Enabled = true
serverobj.IP = &models.IPAddr{Type: "V4", Addr: "10.90.20.12"}
pobj.Servers = append(pobj.Servers, &serverobj)
npobj, err := aviClient.Pool.Create(&pobj)
if err != nil {
    fmt.Println("Pool creation failed: ", err)
    return
}
```

Creating a Virtual Service

The syntax mentioned below created a virtual service with the following attributes:

- Name: my-test-vs
- IP Address: 10.90.20.51
- Port Number: 80
- The pool associated with the virtual service: my-test-pool

```
vsobj := models.VirtualService{}
vsobj.Name = "my-test-vs"
vipip := models.IPAddr{Type: "V4", Addr: "10.90.20.51"}
vsobj.Vip = append(vsobj.Vip, &models.Vip{VipID: "myvip", IPAddress: &vipip})
vsobj.PoolRef = npobj.UUID
vsobj.Services = append(vsobj.Services, &models.Service{Port: 80})

nvsobj, err := aviClient.VirtualService.Create(&vsobj)
if err != nil {
    fmt.Println("VS creation failed: ", err)
    return
}
fmt.Printf("VS obj: %+v", *nvsobj)
```

Fetching an Object by the Name

The sample syntax mentioned below fetches the virtual service details with the name my-test-vs and provides the following output: `Cloud UUID: f39f950a-e6ca-442d-b546-fc31520991bb`.

```
var obj interface{}
err = aviClient.AviSession.GetObjectByName("virtualservice", "my-test-vs", &obj)
fmt.Printf("VS obj: %v\n", obj)

err = aviClient.AviSession.GetObject(
    "virtualservice", session.SetName("my-test-vs"), session.SetResult(&obj),
    session.SetCloudUUID("cloud-f39f950a-e6ca-442d-b546-fc31520991bb"))
fmt.Printf("VS with CLOUD_UUID obj: %v", obj)
```

Deleting a Virtual Service

The syntax mentioned below delete the desired virtual service using the Object ID for the virtual service.

```
aviClient.VirtualService.Delete(nvsobj.UUID)
```

Deleting a Pool

Use the following syntax to delete the desired pool.

```
aviClient.Pool.Delete(npobj.UUID)
```

Creating a Basic Virtual Service (my-test-vs) using `create_vs.go`

```
$ go run create_vs.go
```

Metric and Inventory API Example

```
package main

import (
    //"flag"
    "fmt"
    "github.com/avinetworks/sdk/go/clients"
    "github.com/avinetworks/sdk/go/session"
)

type MetricRequest struct {
    Step           int    `json:"step"`
    Limit          int    `json:"limit"`
    EntityUUID     string `json:"entity_uuid"`
    MetricID       string `json:"metric_id"`
    IncludeName    string `json:"include_name"`
    IncludeRefs    string `json:"include_refs"`
    PadMissingData string `json:"pad_missing_data"`
}

type Metrics struct {
    MetricRequests []MetricRequest `json:"metric_requests"`
}

func main() {
    // Create a session and a generic client to Avi Controller
    aviClient, err := clients.NewAviClient("10.10.25.42", "admin",
        session.SetPassword(""),
        session.SetTenant("admin"),
        session.SetInsecure)
    if err != nil {
        fmt.Println("Couldn't create session: ", err)
        return
    }
    mr := MetricRequest{Step: 1, Limit: 1, EntityUUID: "*", MetricID:
```

```
"l7_server.max_concurrent_sessions", IncludeName: "True", IncludeRefs: "True",
PadMissingData: "False"}
    sr := []MetricRequest{}
    sr = append(sr, mr)
    req := Metrics{MetricRequests: sr}
    path := "/api/analytics/metrics/collection"
    var rsp interface{}
    aviClient.AviSession.Post(path, req, &rsp)
    fmt.Printf("response %v\n", rsp)
}
```

Compilation

Use the following syntax to compile the NSX Advanced Load Balancer Go SDK package.

```
$ go build -o /usr/bin/create_vs create_vs.go
```

Including Go SDK package in a third party code

The NSX Advanced Load Balancer Go SDK package can also be integrated with any third-party Go code package. The following is an example of importing the NSX Advanced Load Balancer Go SDK package into a third-party Go code package.

```
import
 (
    "github.com/avinetworks/sdk/go/clients"
    "github.com/avinetworks/sdk/go/session"
```

The following is an example entry of `vendor.json` file for the Terraform provider:

```
"package": [ {
        "path": "github.com/avinetworks/sdk/go/clients",
        "revision": "796ddcccdc37a5a9771bfbb716b159ae5c9b4b11",
        "revisionTime": "2018-04-06T16:51:27.185773",
        "version": "18.1.3",
        "versionExact": "18.1.3"
    },
    {
        "path": "github.com/avinetworks/sdk/go/session",
        "revision": "796ddcccdc37a5a9771bfbb716b159ae5c9b4b11",
        "revisionTime": "2018-04-06T16:51:27.185773",
        "version": "18.1.3",
        "versionExact": "18.1.3"
    } ]
```

Additional Information

The location on GitHub for NSX Advanced Load Balancer Go SDK package: https://github.com/vmware/alb-sdk/tree/eng/go

User Roles

The role, in combination with the tenant, comprises the authorization settings for an NSX Advanced Load Balancer user. When assigned to a user, the role defines the type of access the user has to each area of the NSX Advanced Load Balancer system. Roles provide granular RBAC within NSX Advanced Load Balancer.

Access Types

For each NSX Advanced Load Balancer resource (object type) and within each group of resources (system area), the user can have the following privileges:

- **Write**: The user has full access to create, read, modify, and delete resources.
- **Read**: The user can only read the existing configuration of resources. For example, the user can see how a virtual service is configured and view the health and analytics data of the virtual service but is unable to modify the configuration or delete the virtual service.
- **No Access**: The user has no access to the resources and cannot even read or list these resources.
- **Assorted**: The user has a mixture of the above privileges for different resources within the system area.

Manage User Roles

Each user must be associated with at least one role. The role can be either predefined or a custom role.

Procedure

1. From the UI, navigate to **Administration > Accounts > Roles**.

   The pre-defined roles in NSX Advanced Load Balancer are listed. Each user role has different levels of access to different settings in NSX Advanced Load Balancer.
2. Click the **+** icon in the table to expand the access settings for a role.
3. Click the edit icon against a role to configure the access settings for the role according to the requirement.

Results

If multi-tenancy is configured, a user can be assigned to more than one tenant and have a separate role for each tenant.

Create a Role

If none of the pre-defined roles fit the access requirements for some user accounts, you can create custom roles according to the requirement.

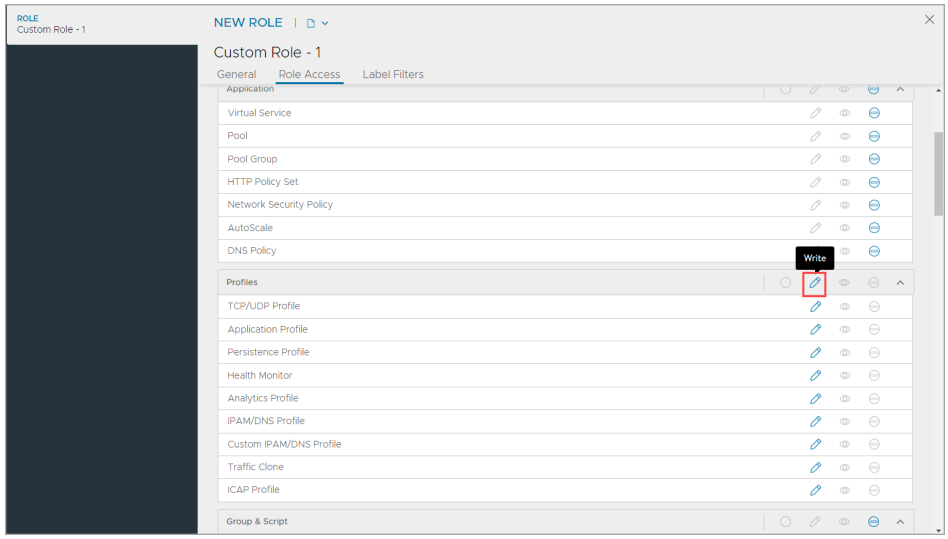
Procedure

1. Navigate to **Administration > Accounts > Roles**.
2. Click **CREATE** to open the **New Role** screen.
3. Under **General**, enter a **Name** for the role.

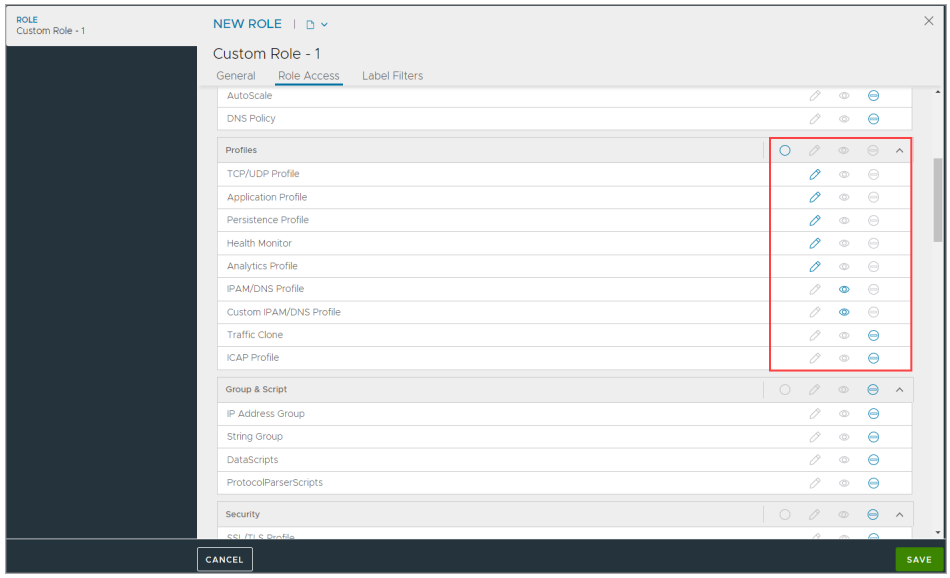
4 Configure the access for each system area under **Role Access**. By default, all the roles are set to no access.

a To set access to all the resources within a system area, click the required access icon on the title row of the system row.

To provide write access to all the profiles, click the write icon on the title row of the **Profiles** system area:



b To give write, read, or no access for some resources, click the relevant icon for each resource. Automatically, the assorted access is enabled on the title row of the system area, indicating multiple types of access.



5 Optionally, configure **Label Filters** for the role to provide object-based granular access.

a Click **Add** to open the **LABEL FILTER** sub-screen.

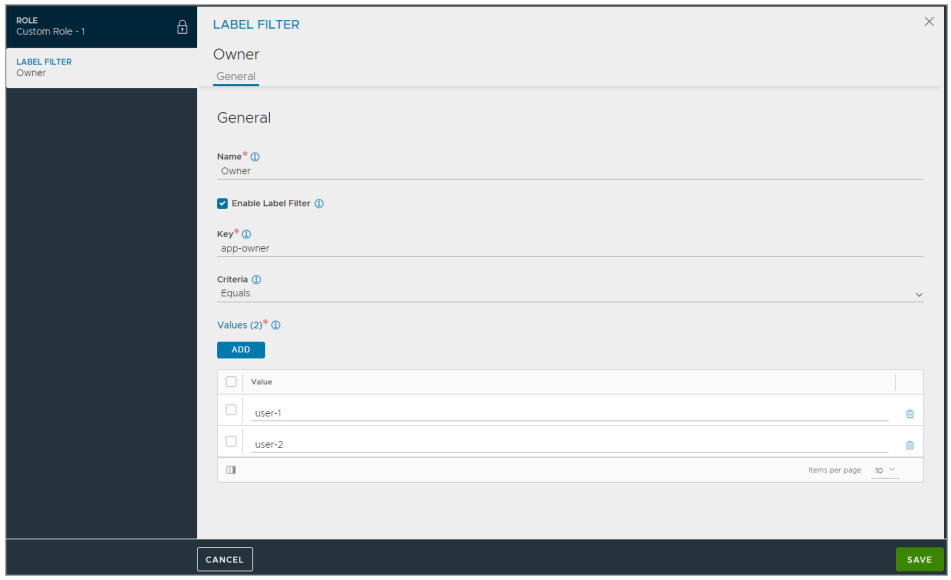
b Enter the **Name** for the filter.

c Select **Enable Label Filter** to apply the label.

d Enter the **Key**.

e Select the **Criteria** to match the key:value pair.

f Under the **Values**, click **Add** to add values.



g Click **Save**.

6 Click **Save** to save the new role.

Note Roles can only be created in the admin tenants.

Assign a Role to a Local User Account

You can assign a role to local and remote user accounts (LDAP, TACACS+). The procedure differs depending on where the account is maintained.

Roles can be assigned to a user account when the account is created or at any time later. In either case, select the **Role** from the **Role** drop-down menu in the configuration popup for the user account.

Note User accounts are case-sensitive. LDAP users are an exception.

Procedure

1 Navigate to **Administration > Accounts > Users**.

2 To configure a new account, click **Create**. To edit an existing account, click the edit icon.

3 Under **Tenant & Role**, click **Add** and select the **Role** from the **Roles** drop-down menu. To create a custom role, click **Create**.

LDAP or TACACS+ Users to Roles

You can assign roles to users if LDAP or TACACS+ remote authentication is used.

Roles are assigned to users based on the following:

LDAP group

A role is assigned to users in any group or specifically to users who are either members or not members of specific groups.

LDAP attributes

For users who match the LDAP group filter, the role is assigned to those who either do or do not have specific attributes and values.

The mappings are configured within NSX Advanced Load Balancer rather than the LDAP or TACACS+ server.

To map LDAP or TACACS+ users to NSX Advanced Load Balancer roles, use the following steps. Multiple mappings can be configured if needed for any combination of user group name and attribute:value pair.

Prerequisites

- NSX Advanced Load Balancer authentication or authorization must be set to remote, and an LDAP or TACACS+ Auth profile is applied.
- Group names are case-sensitive for LDAP mapping.

Procedure

1. Navigate to **Templates > Security > Auth Mapping Profile** and click **CREATE**.
2. In the **CREATE AUTH MAPPING PROFILE** under the **General** tab, enter the profile **Name**.
3. Select the **Type** of auth mapping profile to which these rules are applied and click **ADD** button. Select **LDAP** to define **LDAP Group Match**.

   Note Depending on the type selected, the auth profile settings are displayed.

4. Select the filter for the **LDAP Group Match**.
   - **Any**: Users in any LDAP group match the filter.
   - **Member**: Users must be members of the specified groups. When specifying a group, you must use the common name (CN). If entering multiple group names, use commas between the names.
   - **Not a Member**: Users must not be members of the specified groups.
5. Select the filter for the **Attribute Match**.
   - **Any**: Users match regardless of attributes or their values.

- **Contains**: The user must have the specified attribute, and the attribute must have one of the specified values.
- **Does Not Contain**: The user must not have the specified attribute and value(s).
- **Regex**:

6 Under **Action**, select **Custom Mapping**.

7 Select **User Role** from the **ADD** drop-down menu.

8 Select the **Type** from the drop-down menu.

- **All**: Assigns all roles.
- **Attribute Regex**:
- **Attribute Value**: Assigns the role whose name matches an attribute value from the LDAP server.

  Note **Attribute Match Value** is case sensitive. If there is a mismatch, the following error message will be displayed: `DEBUG [auth_rules.update_user_access:502] ldap: NO Tenant/Role assigned`

- **Group Name**: Assigns the role whose name matches a group name on the LDAP server.
- **Group Regex**:
- **Selected**: Displays a Roles drop-down menu. Select the role from the menu.

9 If using multitenancy, users also can be mapped to tenants. Tenants can be mapped by selecting **User Tenant** from the **ADD** drop-down menu. For more information, see Chapter 6 Tenant Settings.

10 Click **Save**.

Results

The new mapping appears in the **Tenant and Role Mapping** table.

Extended Granular RBAC

This section explains extended granular Role Based Access Control (RBAC) feature.

Note The terms labels and markers are used interchangeably throughout the section.

- Multiple values (value 1, value 2, value 3...), mapped to a single key for an object.

  Example:

  User 1 has to transfer a pre-prod application to prod, for which User 2 has access.

  The object has label “app”:[“pre-prod”].

  User 1 has access to “app”:”pre-prod”.

  User 1 needs to transfer this object to “prod“.

In this case, the object has to be configured with “key” : “list of labels” for transferring the application.

Any user who has access to the object with labels configured in the role, can use additional labels which the user does not have access to.

Here, an additional label has to be applied for the object.

“app“: [“pre-prod”, “prod”] After the transfer, the new app owner (User 2) can remove “app“: “pre-prod“ from the object to discontinue access to User 1.

- The access privilege granted by setting the `allow_unlabelled_ access` to `true` in the role object depends on the privilege type of the role.
- A new object, `LabelGroup` is introduced to track the labels allowed or suggested for a given tenant. For each tenant, there can be associated label groups. At the tenant level, the labels can either be enforced or deprecated.

Using Markers

The steps to configure the object using markers are as follows.

Configure the Object

Let us consider configuring the pool object pool-123 with two owners, engineer, and marketing. Here, `“Key”: [“value1”, “value2”] :: “Owner”: [“eng”, “marketing”].`

```
[admin:ctrl10]: > configure pool pool-123
[admin:ctrl10]: pool> markers
New object being created
[admin:ctrl10]: pool:markers> key owner
[admin:ctrl10]: pool:markers> values eng
[admin:ctrl10]: pool:markers> values marketing
[admin:ctrl10]: pool:markers> save
[admin:ctrl10]: pool> save
```

The pool configuration shows that the key and the corresponding values are assigned as shown below.

```
+--------------------------------------+---------------------------------+
| Field                                | Value                           |
+--------------------------------------+---------------------------------+
| uuid                                 | pool-0f373267-
d62d-47b5-90e6-486abdd5da53                                                |
| name                                 | pool-123                        |
| default_server_port                  | 80                              |
| graceful_disable_timeout             | 1 min                           |
| connection_ramp_duration             | 10 min                          |
| max_concurrent_connections_per_server | 0                               |
| lb_algorithm                         | LB_ALGORITHM_LEAST_CONNECTIONS|
| lb_algorithm_hash                    |
LB_ALGORITHM_CONSISTENT_HASH_SOURCE_IP_ADDRESS                             |
| inline_health_monitor                | True                            |
| use_service_port                     | False                           |
```

```
| capacity_estimation                   | False                          |
| capacity_estimation_ttfb_thresh       | 0 milliseconds                 |
| vrf_ref                               | global                         |
| fewest_tasks_feedback_delay           | 10 sec                         |
| enabled                               | True                           |
| request_queue_enabled                 | False                          |
| request_queue_depth                   | 128                            |
| host_check_enabled                    | False                          |
| sni_enabled                           | True                           |
| rewrite_host_header_to_sni            | False                          |
| rewrite_host_header_to_server_name    | False                          |
| lb_algorithm_core_nonaffinity         | 2                              |
| lookup_server_by_name                 | False                          |
| analytics_profile_ref                 | System-Analytics-Profile       |
| markers[1]                            |                                |
|   key                                 | owner                          |
|   values[1]                           | eng                            |
|   values[2]                           | marketing                      |
| tenant_ref                            | admin                          |
| cloud_ref                             | Default-Cloud                  |
| server_timeout                        | 0 milliseconds                 |
| delete_server_on_dns_refresh          | True                           |
| enable_http2                          | False                          |
| ignore_server_port                    | False                          |
| routing_pool                          | False                          |
+---------------------------------------+--------------------------------+
```

Create Roles

Create the Role named Eng with write access to the pool object.

```
[admin:ctrl10.79.169.184]: > configure role role-eng
[admin:ctrl10.79.169.184]: role> privileges
New object being created
[admin:ctrl10.79.169.184]: role:privileges> type write_access
[admin:ctrl10.79.169.184]: role:privileges> resource permission_pool
[admin:ctrl10.79.169.184]: role:privileges> save
[admin:ctrl10.79.169.184]: role> filters
New object being created
[admin:ctrl10.79.169.184]: role:filters> match_operation role_filter_glob_match
[admin:ctrl10.79.169.184]: role:filters> match_label
[admin:ctrl10.79.169.184]: role:filters:match_label> key owner
[admin:ctrl10.79.169.184]: role:filters:match_label> values *eng*
[admin:ctrl10.79.169.184]: role:filters:match_label> save
[admin:ctrl10.79.169.184]: role:filters> save
[admin:ctrl10.79.169.184]: role> no allow_unlabelled_access
[admin:ctrl10.79.169.184]: role> save
```

The role is viewed as shown below.

```
+-------------------------+------------------------------------------+
| Field                   | Value                                    |
+-------------------------+------------------------------------------+
| uuid                    | role-870880cf-6093-4dbb-83bb-b6e0566dfc83 |
| name                    | role-eng                                 |
```

```
| privileges[1]           |                                          |
|   type                  | WRITE_ACCESS                             |
|   resource              | PERMISSION_POOL                          |
| filters[1]              |                                          |
|   match_operation       | ROLE_FILTER_GLOB_MATCH                   |
|   match_label           |                                          |
|     key                 | owner                                    |
|     values[1]           | *eng*                                    |
|   enabled               | True                                     |
| allow_unlabelled_access | False                                    |
| tenant_ref              | admin                                    |
+-------------------------+------------------------------------------+
```

Note For this role, `allow_unlabelled_access` is disabled. This means, the unlabelled objects are not visible to the user. For unlabelled objects to be visible, this option has to be set to `True`.

Similarly, the role marketing can be configured with the required permissions to the object.

Create a Label Group

Create label group-123 which is a new object that holds a list of `["key1": ["value1", "value2', "value3", …]`.

```
[admin:ctrl]: > configure labelgroup labelgroup-123
[admin:ctrl]: labelgroup> labels
New object being created
[admin:ctrl]: labelgroup:labels> match_operation role_filter_equals
[admin:ctrl]: labelgroup:labels> match_label
[admin:ctrl]: labelgroup:labels:match_label> key owner
[admin:ctrl1]: labelgroup:labels:match_label> values eng
[admin:ctrl1]: labelgroup:labels:match_label> values marketing
[admin:ctrl1]: labelgroup:labels:match_label> values testing
[admin:ctrl1]: labelgroup:labels:match_label> save
[admin:ctrl1]: labelgroup:labels> save
[admin:ctrl1]: labelgroup> save
```

The label group object is as shown below.

```
+------------------+------------------------------------------------+
| Field            | Value                                          |
+------------------+------------------------------------------------+
| uuid             | labelgroup-dee35ef6-b3c3-4eae-956a-9b32b6a87d26 |
| name             | labelgroup-123                                 |
| labels[1]        |                                                |
|   match_operation | ROLE_FILTER_EQUALS                            |
|   match_label    |                                                |
|     key          | owner                                          |
|     values[1]    | eng                                            |
|     values[2]    | marketing                                      |
|     values[3]    | testing                                        |
+------------------+------------------------------------------------+
```

Associate Label Group to a Tenant

```
[admin:ctrl]: > configure tenant t-1
[admin:ctrl]: tenant> enforce_label_group
[admin:ctrl]: tenant> label_group_refs labelgroup-123
[admin:ctrl]: tenant> save
```

The configured tenant is as shown below.

```
+--------------------------------+--------------------------------------+
| Field                          | Value                                |
+--------------------------------+--------------------------------------+
| uuid                           | tenant-b7a85c33-26c3-40eb-a25c-
f86a58d3e5ff                                                          |
| name                           | t-1                                  |
| local                          | True                                 |
| config_settings                |                                      |
|   tenant_vrf                   | False                                |
|   se_in_provider_context       | True                                 |
|   tenant_access_to_provider_se | True                                 |
| enforce_label_group            | True                                 |
| label_group_refs[1]            | labelgroup-123                       |
+--------------------------------+--------------------------------------+
```

Creating an object with markers that does not qualify the assigned key:value rules in the label group, displayed as error.

For example, if the pool object is configured with the marker `"Key": ["sales"]`, an error is displayed as shown below:

```
[admin:ctrl]: > configure pool pool-4
[admin:ctrl]: pool> markers
New object being created
[admin:ctrl]: pool:markers> key owner
[admin:ctrl]: pool:markers> value sales
[admin:ctrl]: pool:markers> save
[admin:ctrl]: pool> save
Error: {"error": "Marker with key 'owner' to value 'sales' does not qualify the labelgroup
rules on this tenant."}
```

Supported Objects

The labels framework is supported for the following config objects:

- VIRTUALSERVICE
- POOL
- POOLGROUP

The other supported objects are as follows:

- NETWORKPROFILE
- HTTPPOLICYSET

- DNSPOLICY
- SECURITYPOLICY
- IPADDRGROUP
- STRINGGROUP
- SSLPROFILE
- SSLKEYANDCERTIFICATE
- NETWORKSECURITYPOLICY
- APPLICATIONPERSISTENCEPROFILE
- ANALYTICSPROFILE
- VSDATASCRIPTSET
- PKIPROFILE
- SERVERAUTOSCALEPOLICY
- AUTOSCALELAUNCHCONFIG
- HARDWARESECURITYMODULEGROUP
- PRIORITYLABELS
- POOLGROUPDEPLOYMENTPOLICY
- GSLBSERVICE
- GSLBGEODBPROFILE
- GSLBAPPLICATIONPERSISTENCEPROFILE
- TRAFFICCLONEPROFILE
- WAFPOLICY
- WAFPROFILE
- ERRORPAGEPROFILE
- ERRORPAGEBODY
- L4POLICYSET
- WAFPOLICYPSMGROUP
- PINGACCESSAGENT
- NETWORKSERVICE
- NATPOLICY
- SSOPOLICY
- PROTOCOLPARSER
- IPREPUTATIONDB

- IpamDnsProviderProfile
- SERVICEENGINEGROUP

Label-Based Permissions for Cloud Objects

Granular RBAC can be applied and enforced on cloud objects using the field `restrict_cloud_read_access in controllerproperties` using the CLI.

This field is set to `False` by default. To enforce label-based permissions on cloud objects, set the `field restrict_cloud_read_access` to `True` as shown below.

```
[admin:ctrl]: > configure controller properties
[admin:ctrl]: controllerproperties> restrict_cloud_read_access
Overwriting the previously entered value for restrict_cloud_read_access
[admin:ctrl]: controllerproperties> save
```

Unsupported Permissions

The following permissions classified under Operations and Administrations are not supported for role-based access control per app using labels:

- Operations
  - PERMISSION_ALERT, PERMISSION_ALERTCONFIG, PERMISSION_ALERTEMAILCONFIG, PERMISSION_ALERTSYSLOGCONFIG, PERMISSION_ACTIONGROUPCONFIG, PERMISSION_SNMPTRAPPROFILE, PERMISSION_TRAFFIC_CAPTURE
- Administration
  - PERMISSION_CONTROLLER, PERMISSION_INTERNAL, PERMISSION_TENANT, PERMISSION_SYSTEMCONFIGURATION, PERMISSION_USER, PERMISSION_ROLE, PERMISSION_UPGRADE, PERMISSION_REBOOT, PERMISSION_TECHSUPPORT

Caveats and Limitations

This section discusses the caveats and limitations in implementing this framework for RBAC.

- The number of labels for objects is limited to 4. There is no limit of values to each label or marker key.
- A maximum of 4 filters can be applied per role.
- The number of characters for the labels key and value is limited to 128.
- A user with no filters can access all objects as per the privileges configured in the role (as per the default behavior).

Granular Role-Based Access Controls Per Field

NSX Advanced Load Balancer provides RBAC to provide granular access to control, manage, and monitor applications within NSX Advanced Load Balancer.

RBAC can be implemented at a field-level. This section covers the use of sub-resources to implement RBAC per field.

Granular RBAC per-Field

Using Granular RBAC per Field, users can be allowed to update an object but restrict the updates to a specific set of fields.

For example, allow users to:

- Enable or disable GSLB service groups, but restrict updating any other fields in the GSLB object
- Enable or disable a virtual service, but restrict updating any other virtual service configuration
- Add, remove, or update the pool servers, but restrict updating any other pool configuration

Sub-resources

To implement per-field RBAC, sub-resources for the existing resources are introduced. These sub-resources are associated with a specific field, feature, or a set of fields within the object. When a sub-resource is configured on a resource with write access, it will allow update to the object only if those sub-resources are the only fields updated. Read access is allowed for the full object, but delete and create are not allowed from that permission. Sub-resources can be combined to allow users to configure multiple fields or features in an object.

To define access for sub-resources, the flags `allow edit to only [subresource(s)]` and `allow edit of entire object except for [subresource(s)]` are introduced.

For example, configure a role with sub-resources as shown below.

```
[admin:10]: > configure role Pool-Enabled-Role
[admin:10]: role> privileges
New object being created
[admin:10]: role:privileges> type write_access
[admin:10]: role:privileges> resource permission_pool
[admin:10]: role:privileges> subresource
[admin:10]: role:privileges:subresource> subresources subresource_pool_enabled
[admin:10]: role:privileges:subresource> save
[admin:10]: role:privileges> save
[admin:10]: role> save
```

The pool is configured as shown below.

```
+--------------------------+------------------------------------------+
| Field                    | Value                                    |
+--------------------------+------------------------------------------+
| uuid                     | role-c5d28445-995c-44b8-9677-610bb20cb2e7 |
| name                     | Pool-Enabled-Role                        |
| privileges[1]            |                                          |
|   type                   | WRITE_ACCESS                             |
|   resource               | PERMISSION_POOL                          |
|   subresource            |                                          |
|     exclude_subresources | False                                    |
```

```
|     subresources[1]      | SUBRESOURCE_POOL_ENABLED                  |
| tenant_ref               | admin                                     |
+--------------------------+-------------------------------------------+
```

Sub-resources enable the user to execute a specific function within the object.

All available sub-resources are listed below.

Sub-resource	Function
SUBRESOURCE_POOL_ENABLED	Add/update/disable pool servers
SUBRESOURCE_POOL_SERVERS	Add/update/remove pool servers
SUBRESOURCE_POOL_SERVER_ENABLED	Enable/disable pool servers
SUBRESOURCE_VIRTUALSERVICE_ENABLED	Enable/disable virtual servers
SUBRESOURCE_GSLBSERVICE_ENABLED	Enable/disable GSLB service objects
SUBRESOURCE_GSLBSERVICE_GROUPS	Update GSLBservice groups
SUBRESOURCE_GSLBSERVICE_GROUPS_ENABLED	Enable/disable GSLBservice groups
SUBRESOURCE_GSLBSERVICE_GROUP_MEMBERS	Update GSLBservice group members
SUBRESOURCE_GSLBSERVICE_GROUP_MEMBER_ENABLED	Enable/disable GSLBservice group members
SUBRESOURCE_VIRTUALSERVICE_AUTO_ALLOCATE_FLOATING_IP	Enable/ disable Auto allocate floating IP

Note Prior to NSX Advanced Load Balancer version 22.1.1, it was only possible to control the update (PUT) action on any resource field. Starting with NSX Advanced Load Balancer version 22.1.1, if the access is not allowed for any field, creation of objects is not permitted as well.

Tenant and Role Mapping Examples

Remote Auth requires the assignment of roles and tenants for every user login through the authorization mapping rules. Authorization is assessed on every login, and the user record is updated. Upon successful user login through an external authentication server, all mapping rules are evaluated.Tenant and role pairs are added to the user access list.

NSX Advanced Load Balancer supports user profile mapping for remote users.

Configuring Remote Authentication

By default, a Controller will have only local authentication established (**Authentication/ Authorization**: **Local**).

To configure remote authentication using the NSX Advanced Load Balancer UI, follow the steps below:

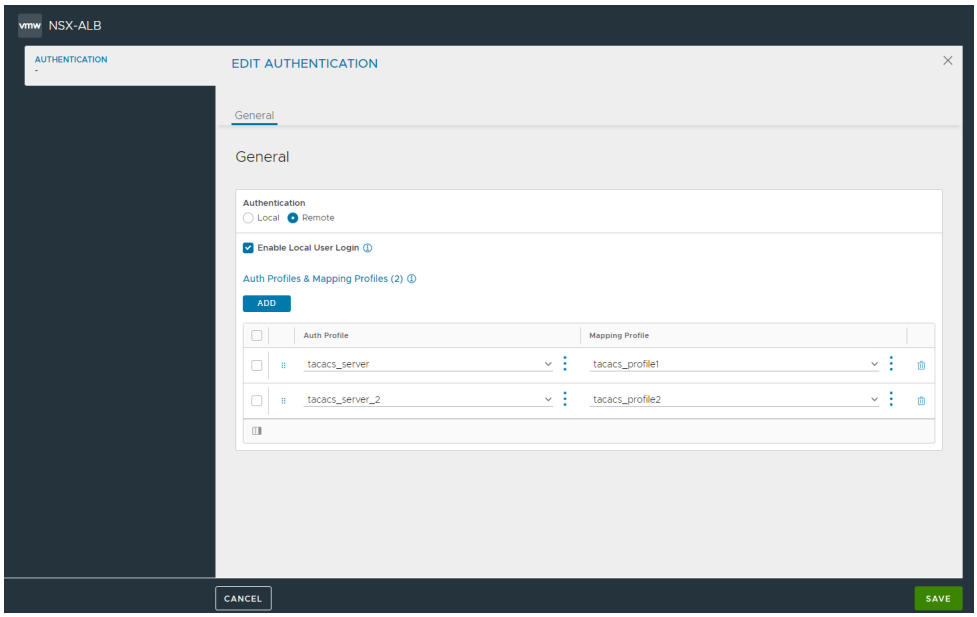
1 Navigate to **Administration > System Settings > EDIT > Authentication**.

2 In the **Edit System Settings** screen, select **Remote** as the Authentication method.

3 Select **Enable Local User Login** to allow users from the local user database to log in with their user credentials.

4 Under **Auth Profiles & Mapping Profiles**, click **Add**.

5 From the **Select Auth Profile**, select an existing remote auth profile or click the vertical menu icon (three dots) to create a new auth profile.



6 Click **Save**.

Note Tenant and role mapping are available only with remote authentication.

Multiple remote authentication profiles can be configured using the NSX Advanced Load Balancer UI under the following conditions:

- Any number of auth profiles (more than 2) can be created provided they are of the same type. Consider the following examples to understand this further:

Profile Types	Result	Reason
■ TACACS ■ TACACS ■ TACACS	Successful	All profiles are of the same type
■ TACACS ■ TACACS ■ LDAP	Fails with an error	All the profiles are not of the same type

- If there are only two auth profiles configured,
  - The primary auth profile has to be SAML auth profile.

- Both profiles cannot be SAMl. So if SAML is the first auth profile, then the second profile has to be any profile other than SAML

Consider the following examples to understand this further:

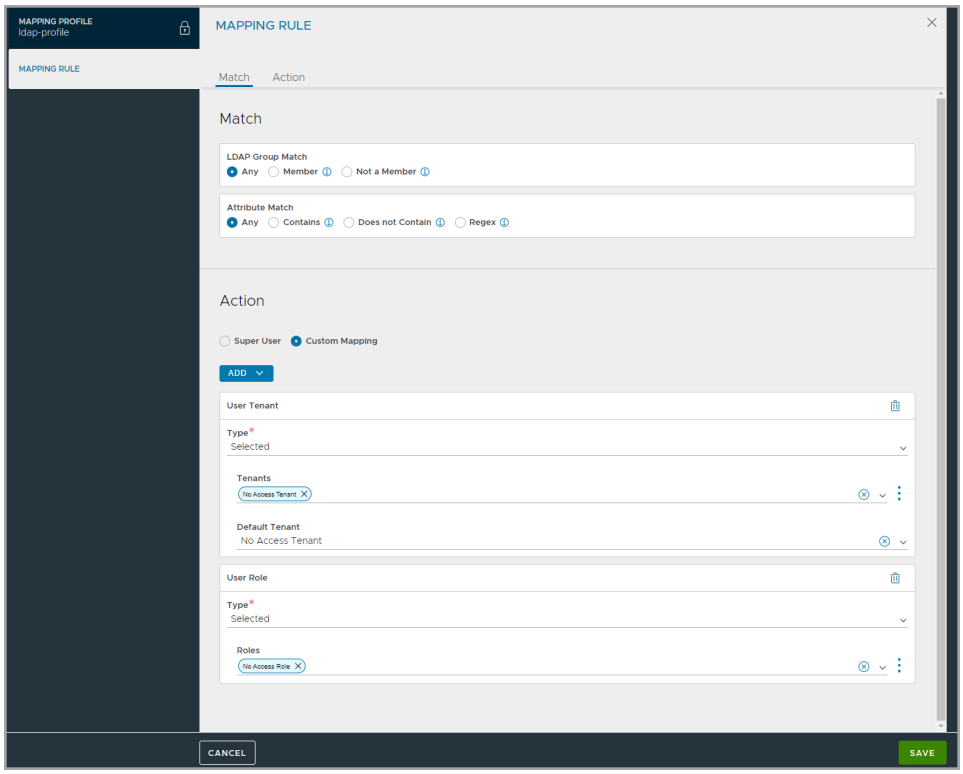
Primary Auth Profile	Secondary Auth Profile	Result	Reason
SAML	LDAP	Successful	The primary and the secondary profiles are of different types.
SAML	SAML	Fails with an error	The primary and secondary profiles are both of type SAML.
LDAP	TACACS	Fails with an error	The primary auth profile is not of type SAML.

Any Group or Any Attribute Rule

A rule with any group or any attribute applies to all users and can be used as a default option. The rule assigns every user to a least-privileged role and tenant.

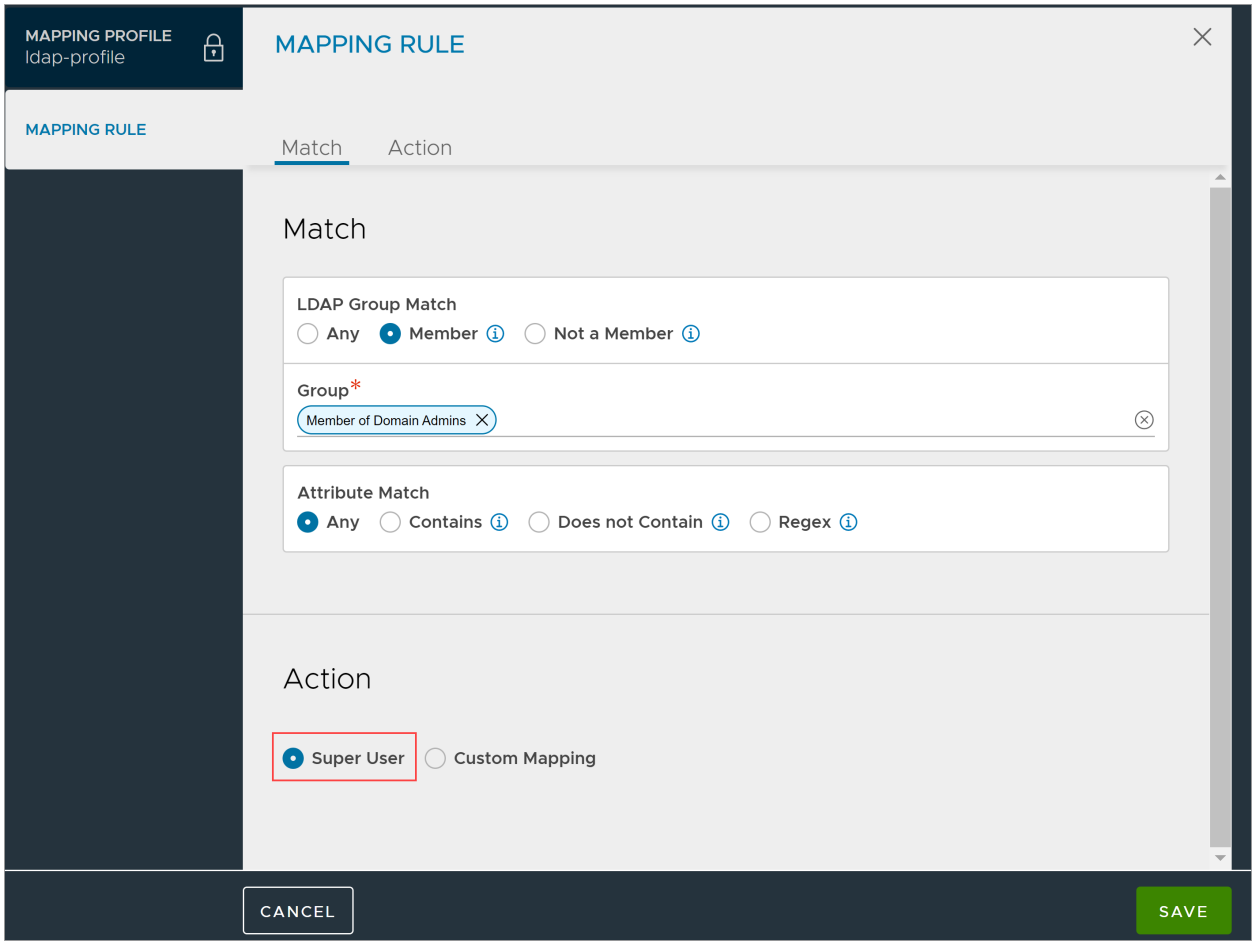
Note The role and tenant need to configured to allow only the least privileges.

If the user is not assigned additional roles or tenant pairs, the least privileged access will take effect after login.



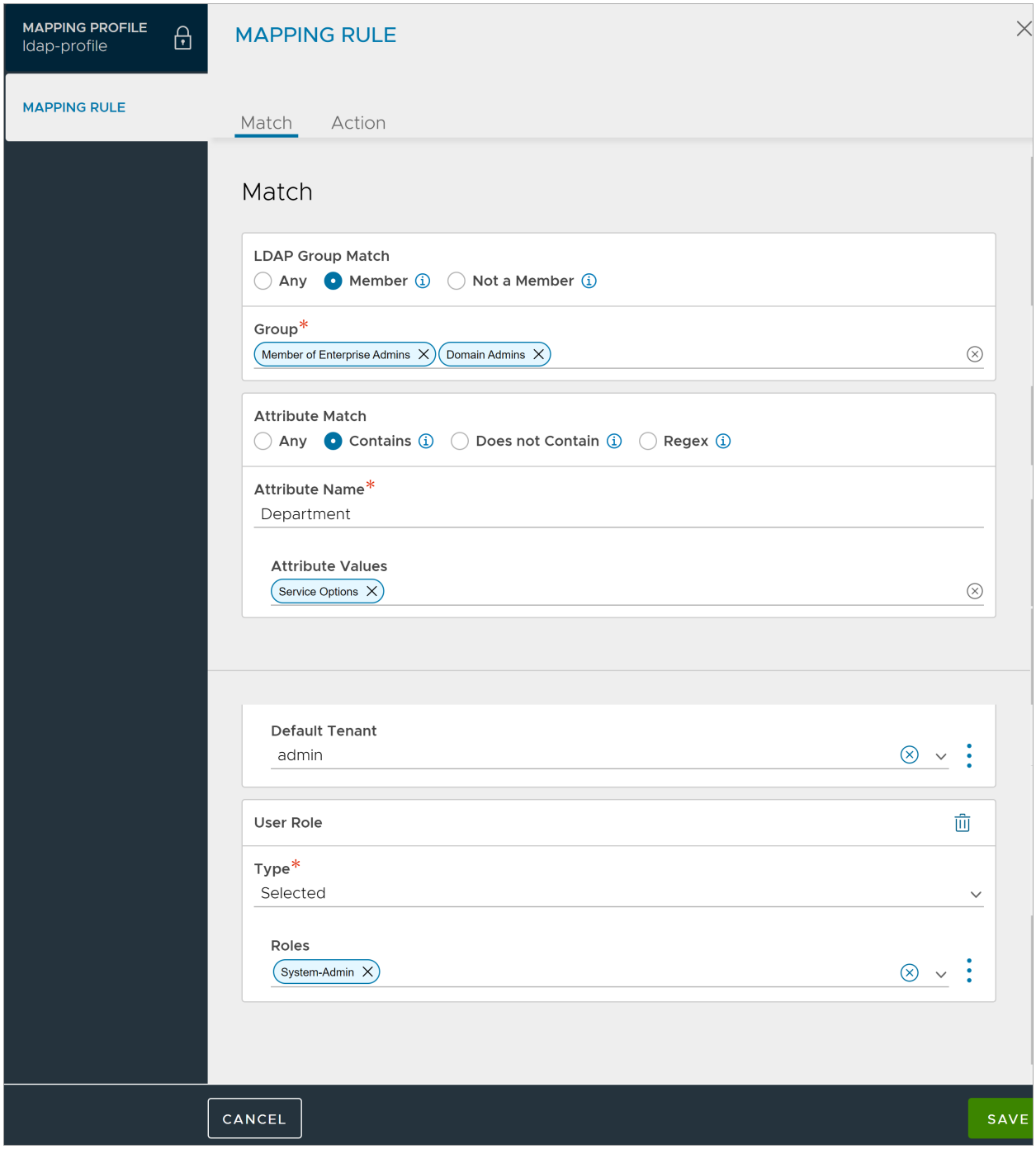
Super User Rule

A rule can be configured to assign **Super User** privileges to a user. This user will have access to all tenants with the most privileged role. Once a user is super user, no other tenant or role mapping assignments will make a difference to the user's access.



Attribute and Group Match

A mapping rule can be required to match both an attribute and group requirement. It will ensure a more specific assignment of role(s) and tenant(s).

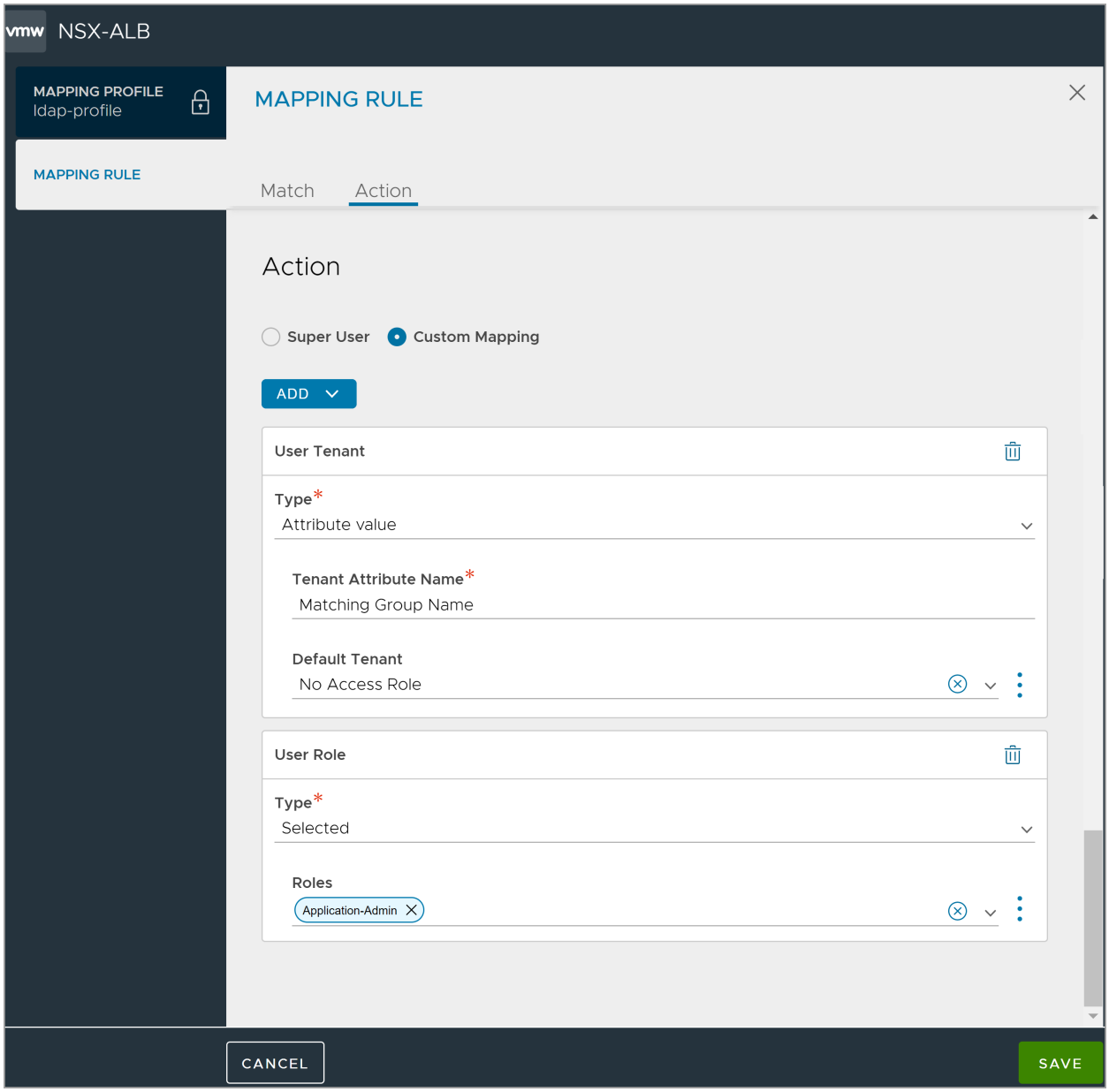


Assign Matching Attribute Values

LDAP/TACACS+ attribute vantageRole for a user can have one or more values. For each value, if there is a configured role with the same name, the role is assigned to the user with access to all tenants. A user session can end up with multiple roles, and the most privileged role will take effect.

Assign Matching Group Names

A user can be a member of multiple LDAP or AD groups. For each group, if there is a configured tenant, the user will be given access to the tenant with any other tenants the user might already have obtained access to using matching rules.



Configuring Default Tenant for Remote Users

When configuring auth mapping rules, you can now select default tenants. When a remote user is created, the tenant selected will be used as the default tenant.

1 From the NSX Advanced Load Balancer UI, navigate to **Templates > Security > Auth Mapping Profile**.

2 Click **CREATE** or edit an existing auth mapping rule.

3 Enter **Name** and **Description**.

4 Select the **Type** as required.

5 Under **Rules**, Click **ADD**.

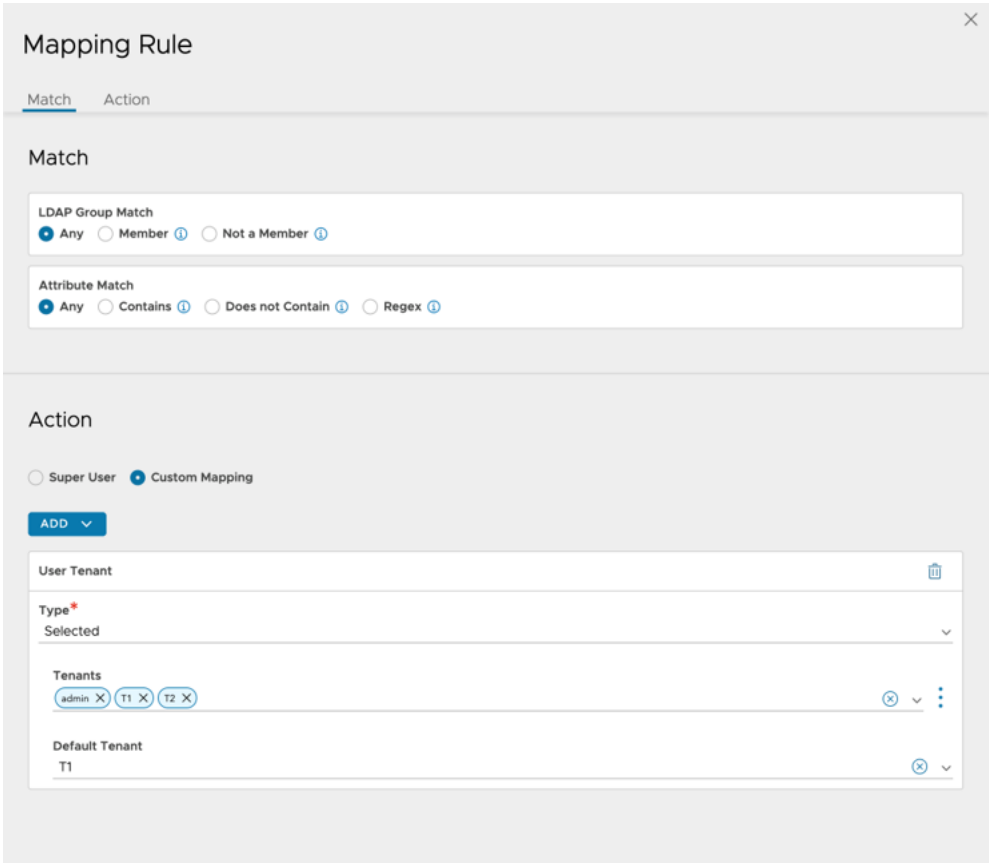
6 Under **Action**, select **Custom Mapping**.

7 Click **ADD** and select **User Tenant** drop-down menu

8 From **Type** drop-down menu, choose **Selected**.

9 Select the required **Tenants**.

Note If multiple tenants are created in the mapping rules, the first tenant selected is taken as the default tenant.



10 Click **Save**.

Alternatively, default tenant for remote users can be configured using the field `default_tenant_ref` through the CLI as shown below:

```
[admin:10-102-64-241]: authmappingprofile:mapping_rules> default_tenant_ref admin
[admin:10-102-64-241]: authmappingprofile:mapping_rules> save
[admin:10-102-64-241]: authmappingprofile> save
+---------------------+-------------------------------------------------------------+
| Field               | Value                                                       |
+---------------------+-------------------------------------------------------------+
```

```
| uuid                 | authmappingprofile-46400ba5-9f05-4ec9-9464-e188010ce7b7 |
| name                 | saml                                                    |
| type                 | AUTH_PROFILE_SAML                                       |
| mapping_rules[1]     |                                                         |
|   index              | 0                                                       |
|   is_superuser       | True                                                    |
|   default_tenant_ref | admin                                                   |
| tenant_ref           | admin                                                   |
+----------------------+---------------------------------------------------------+
[admin:10-102-64-241]: >
```

Default tenant mapping based on the tenant selected, is as shown below:

User Tenant	Behavior	Default Tenant	Description
All	Admin	All tenant values	Any user-specified tenant is used. In the absence of user-defined tenants, admin is selected as the default tenant.
Admin	Admin	All tenant values	Any user-specified tenant is used. In the absence of user-defined tenants, admin is selected as the default tenant.
Selected	Tenant ref	Only selected tenant values	Any User-specified Tenant that is part of the selected tenant list is used. If not, Must check will display an error while creating a new mapping. For example, if the selected tenant list has T1 and T2 but the user has provided T3 as the default tenant, then the must check will display the following error: `Default tenant is not in selected tenants list`. If the user has not selected any tenant but has provided default tenant ref, then the error `Please add at least one tenant in the selected list` is displayed.
Attribute Regex	Tenant ref	All tenant values	User-specified tenant.

User Tenant	Behavior	Default Tenant	Description
Attribute value	Tenant ref	All tenant values	User-specified tenant.
Group name	Tenant ref	All tenant values	User-specified tenant.

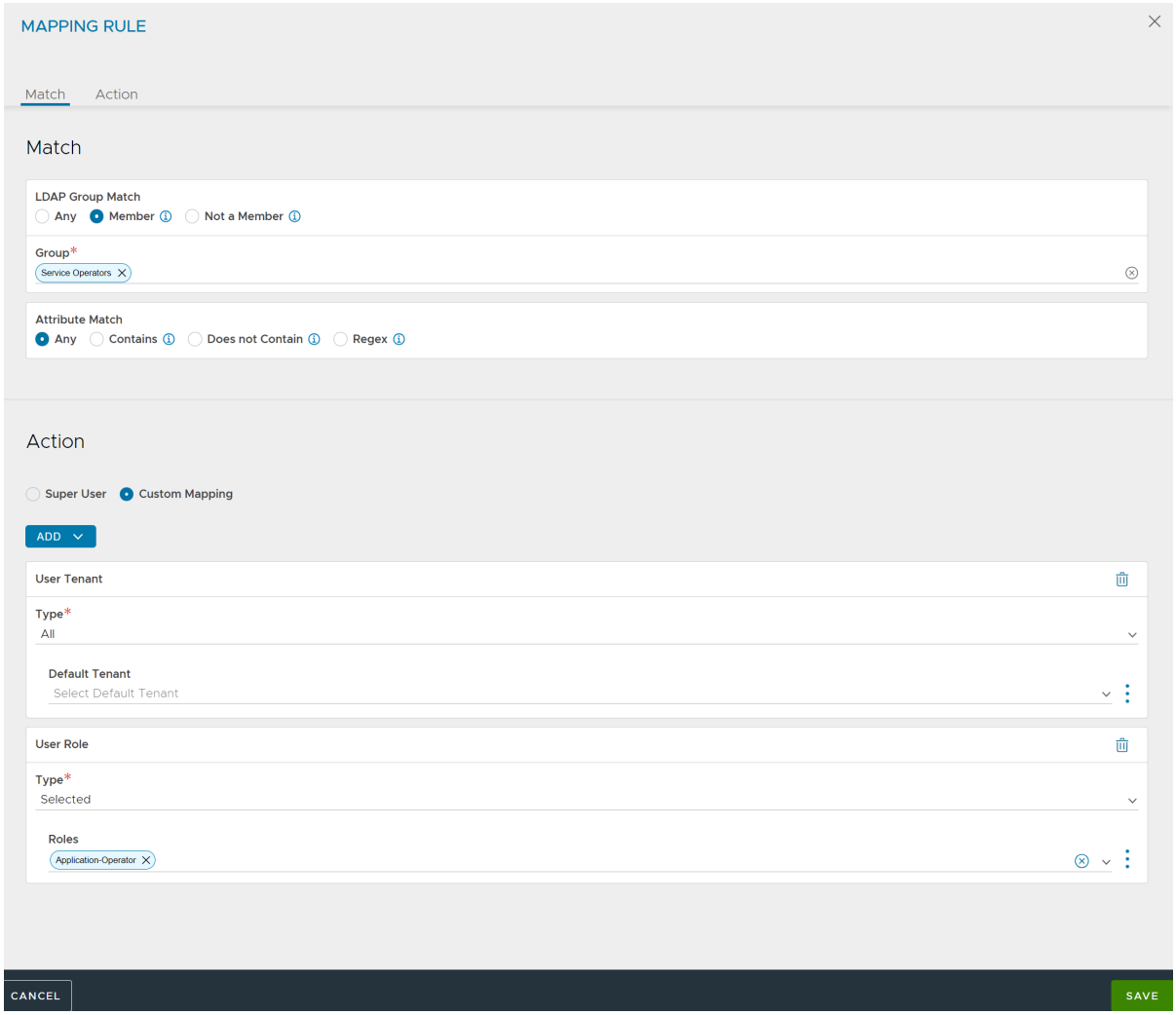
Note In case of Attribute Regex, Attribute value, Group Name, if the user has entered a tenant that is not part of the user tenant list (retrieved at runtime on the basis of Attribute Regex, Attribute Value, Group Name, Group Regex match), then the first entry in the retrieved list will be assigned as the default tenant from the tenant list. For example, User1 can be assigned T1 and T2, but T3 is provided as the default tenant. Depending on the first entry in the received tenant list, either T1 or T2 will become the default tenant.

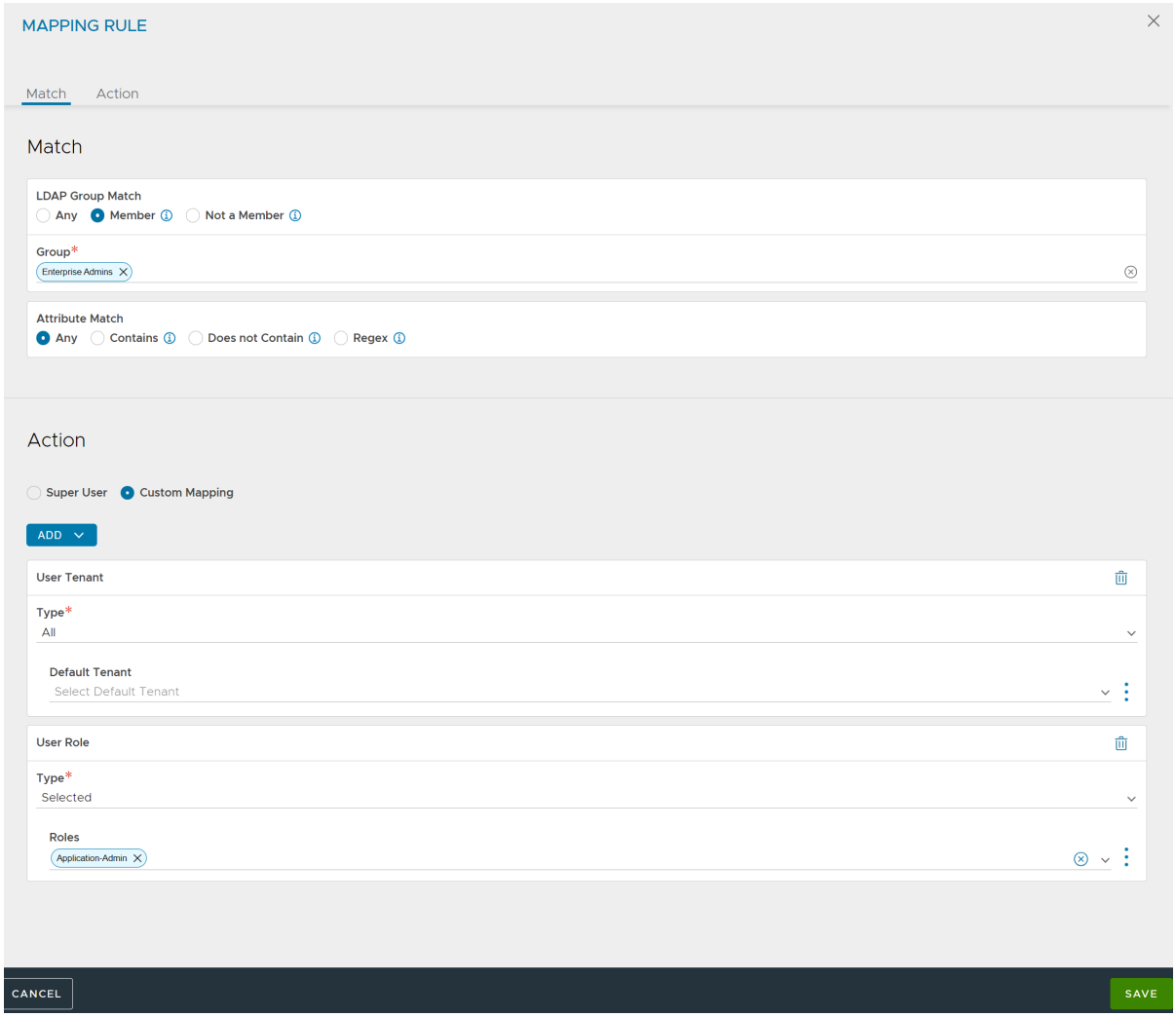
Example: Multiple Groups Mapping to Different Roles

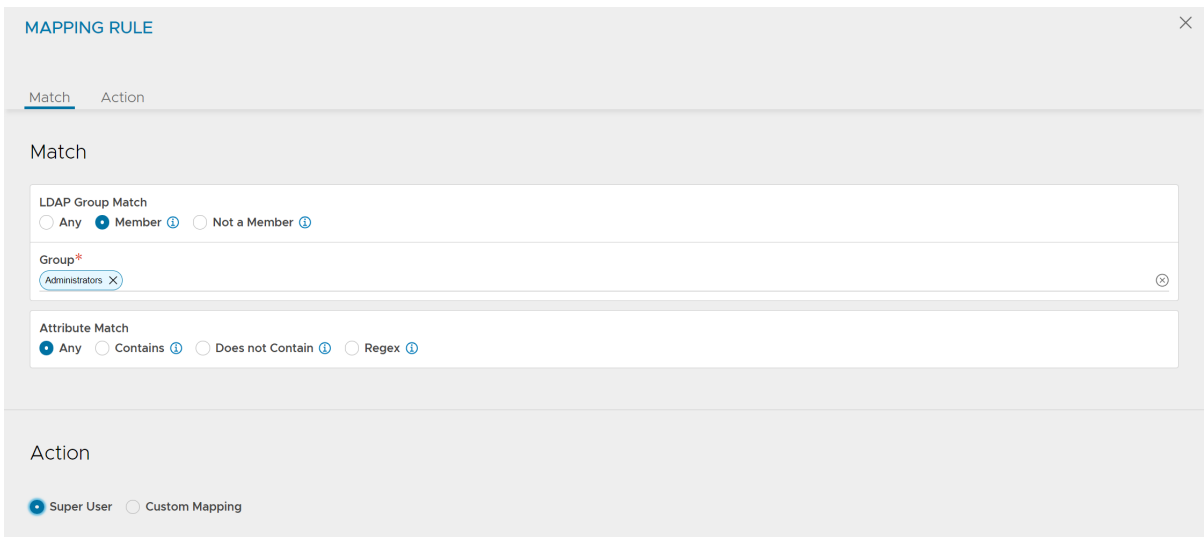
This example illustrates the case of an IT team with three user groups — super-admins, app-admins, and operations — where the following applies:

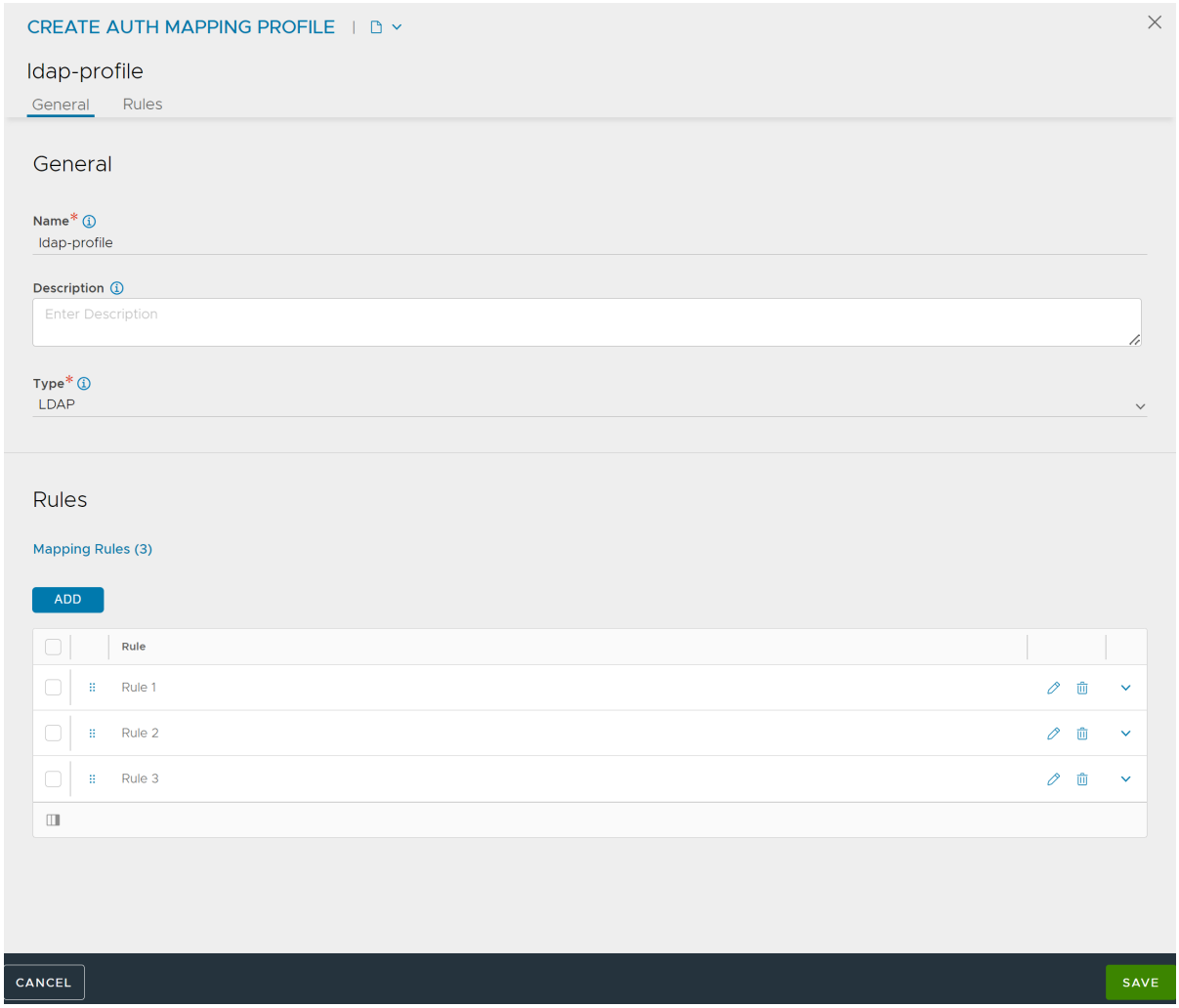
User Groups	Roles
Super Admins	The users can access all tenants and all settings.
Application Admins	■ The users can only create, read, update, and delete virtual services and other profiles. ■ The users can not create clouds.
Application Operators	The users has read-only access.

- Navigate to **Templates > Security > Auth Mapping Profile > CREATE**. Select **Type** and click **ADD** under the **Rules** section.

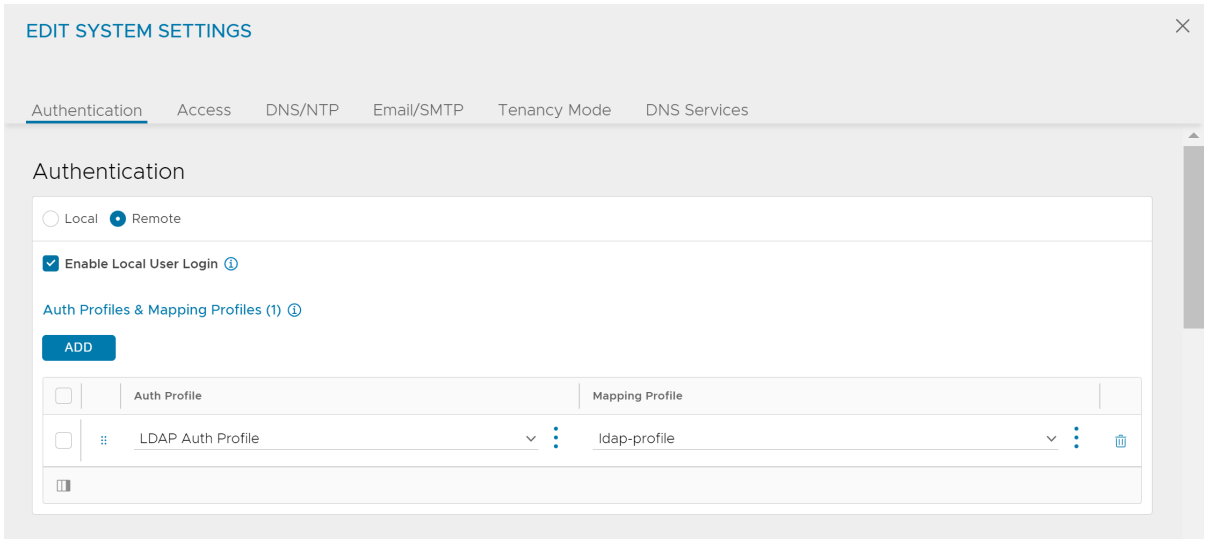

MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Operators
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
All
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
Application-Operator
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Enterprise Admins
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
All
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
Application-Admin
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Administrators
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping



- Navigate to **Administration > System Settings > EDIT**. Under **Authentication**, select **Remote** and click **ADD**.

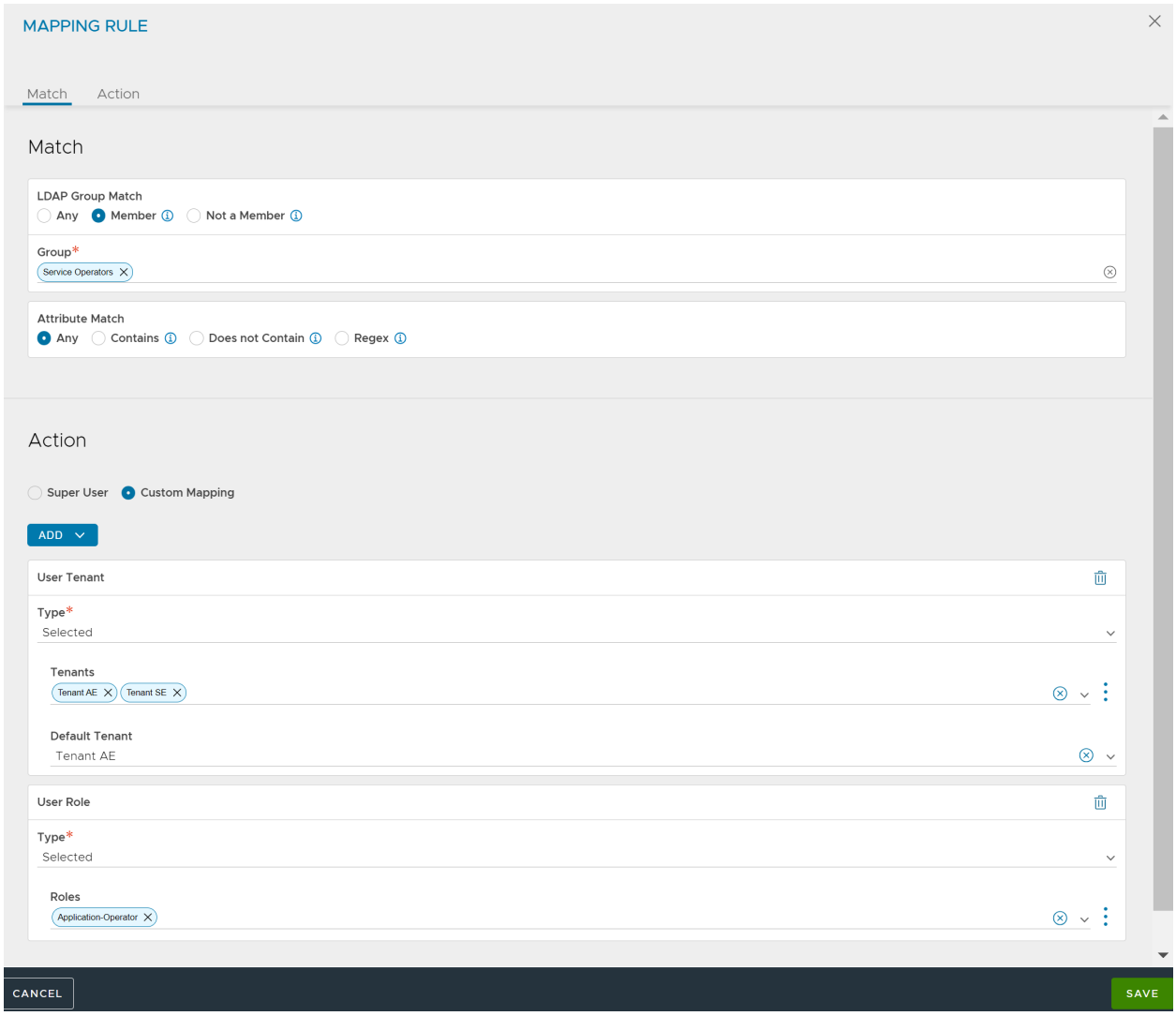


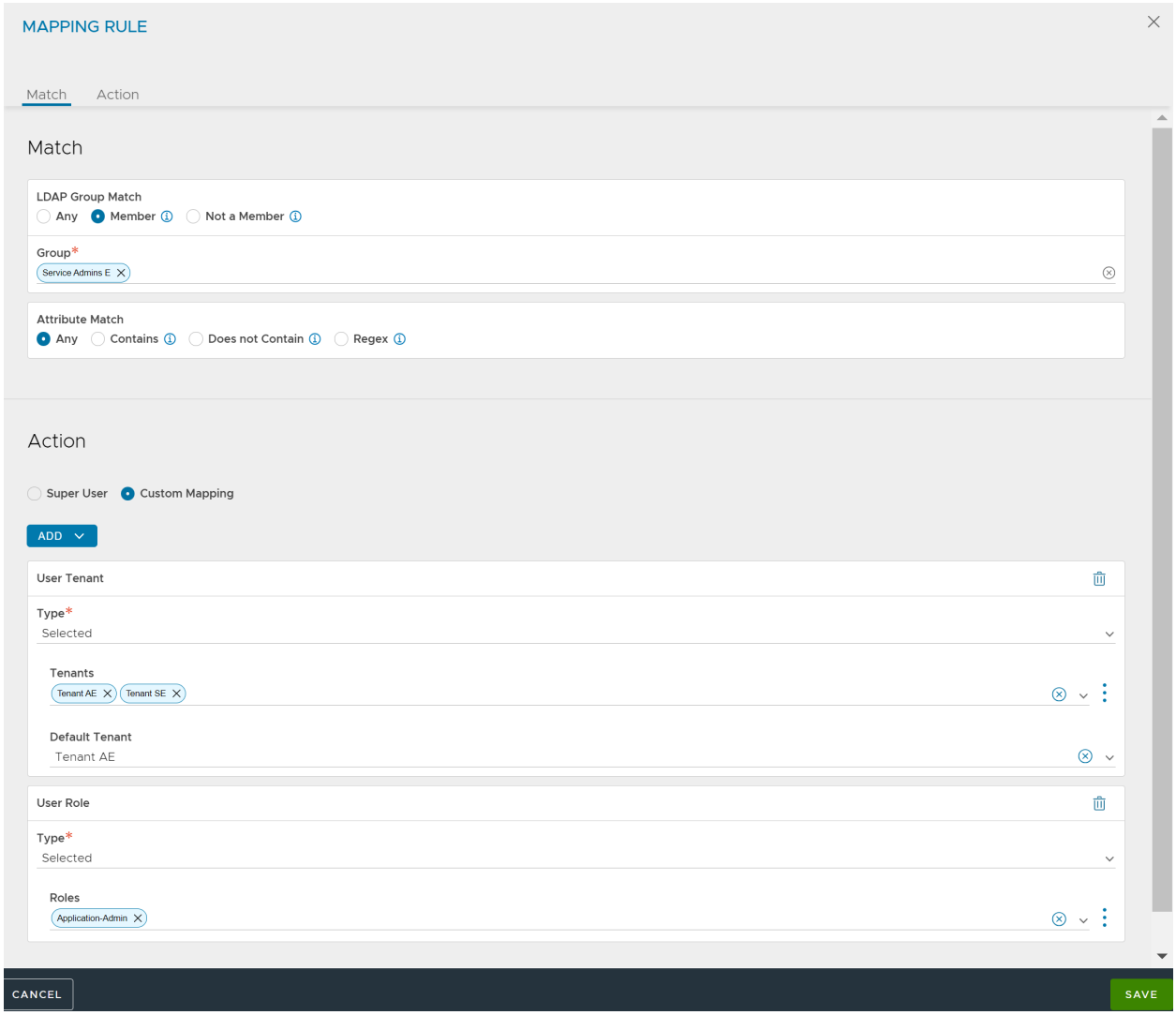
Separate mapping rules are required to map users from each group to different role or tenant assignments.

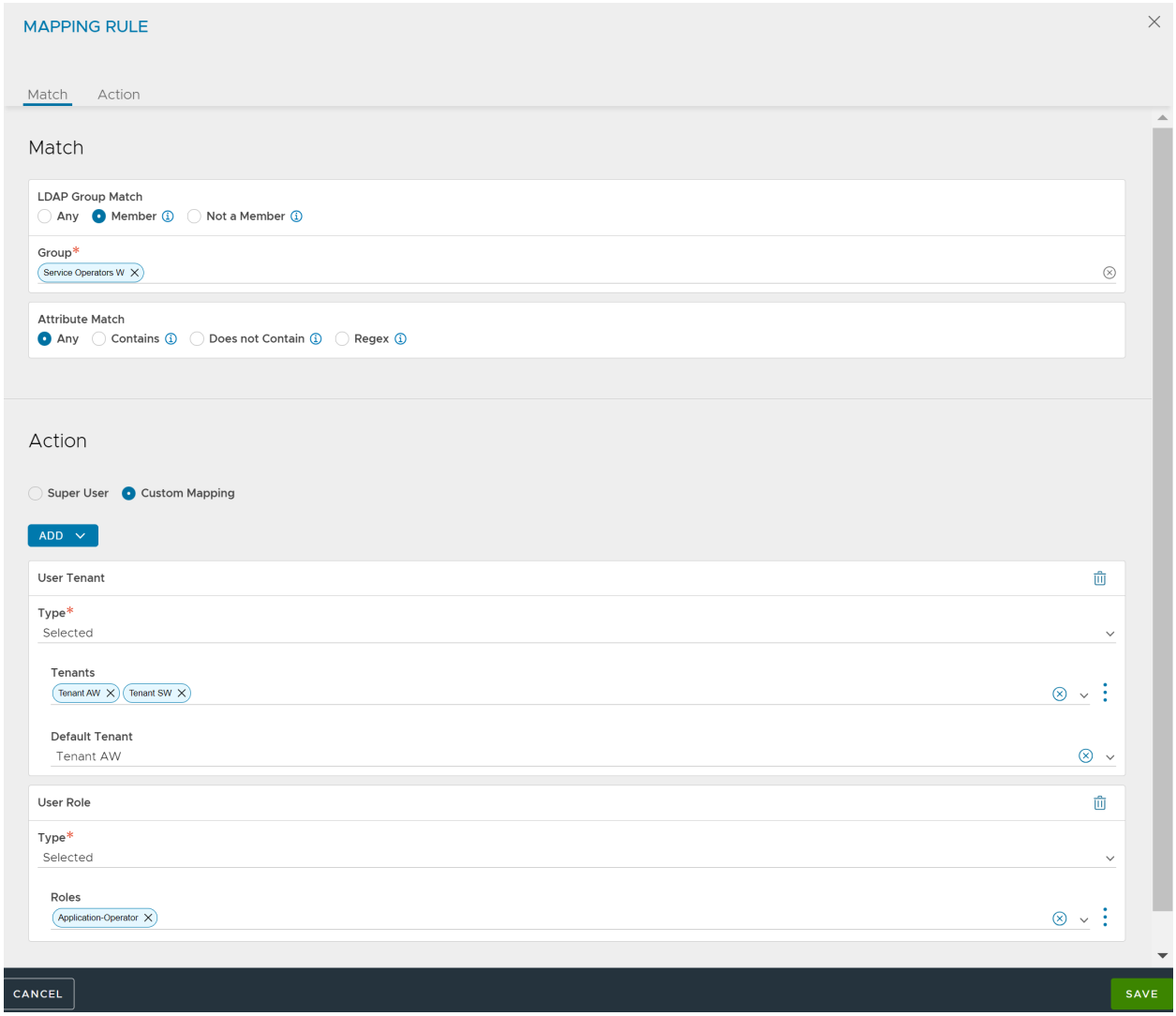
Example: Multiple Groups Mapping to Different Tenants

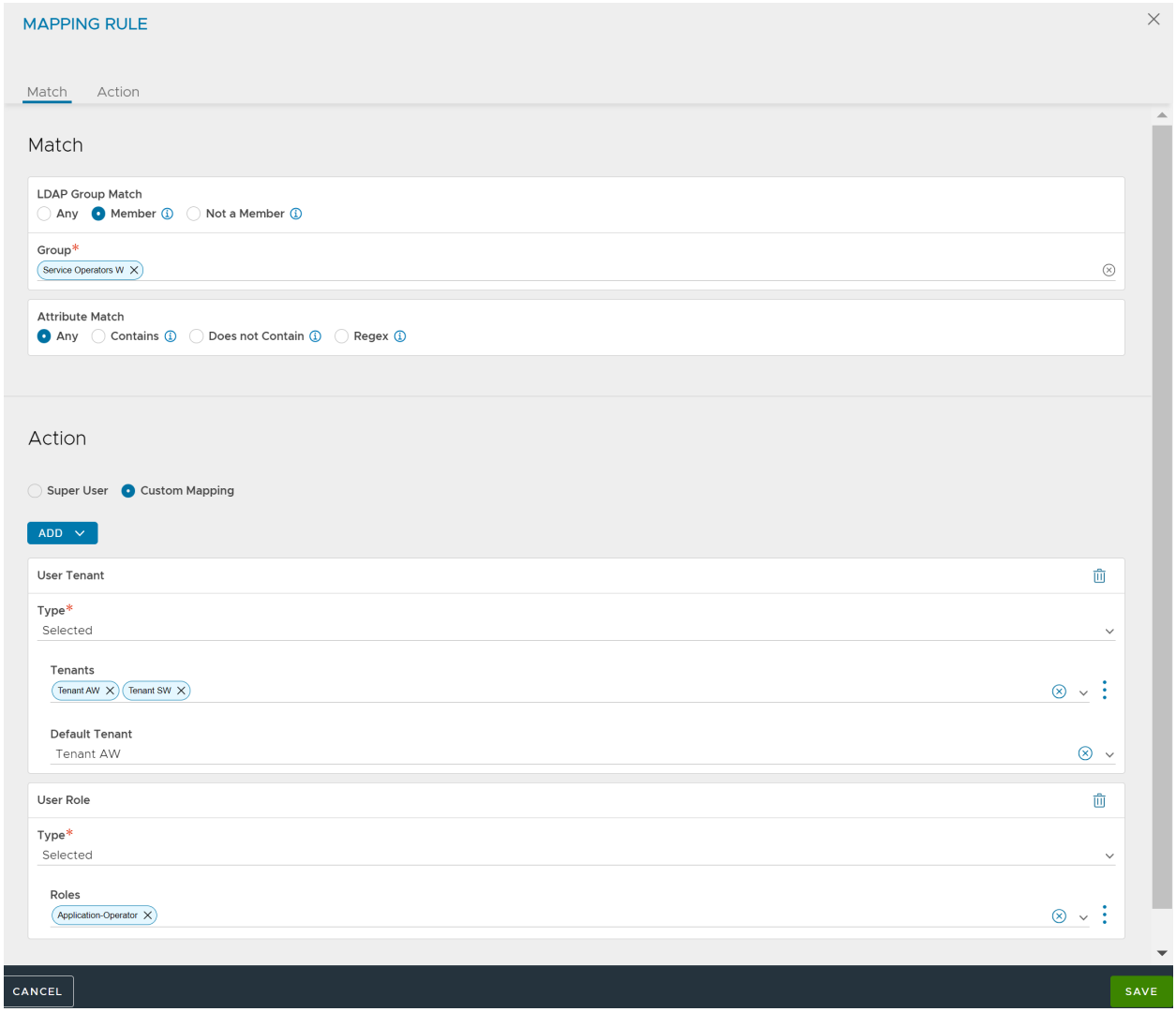
This example illustrates settings for an IT team that expects tenant isolation except for a few super users.

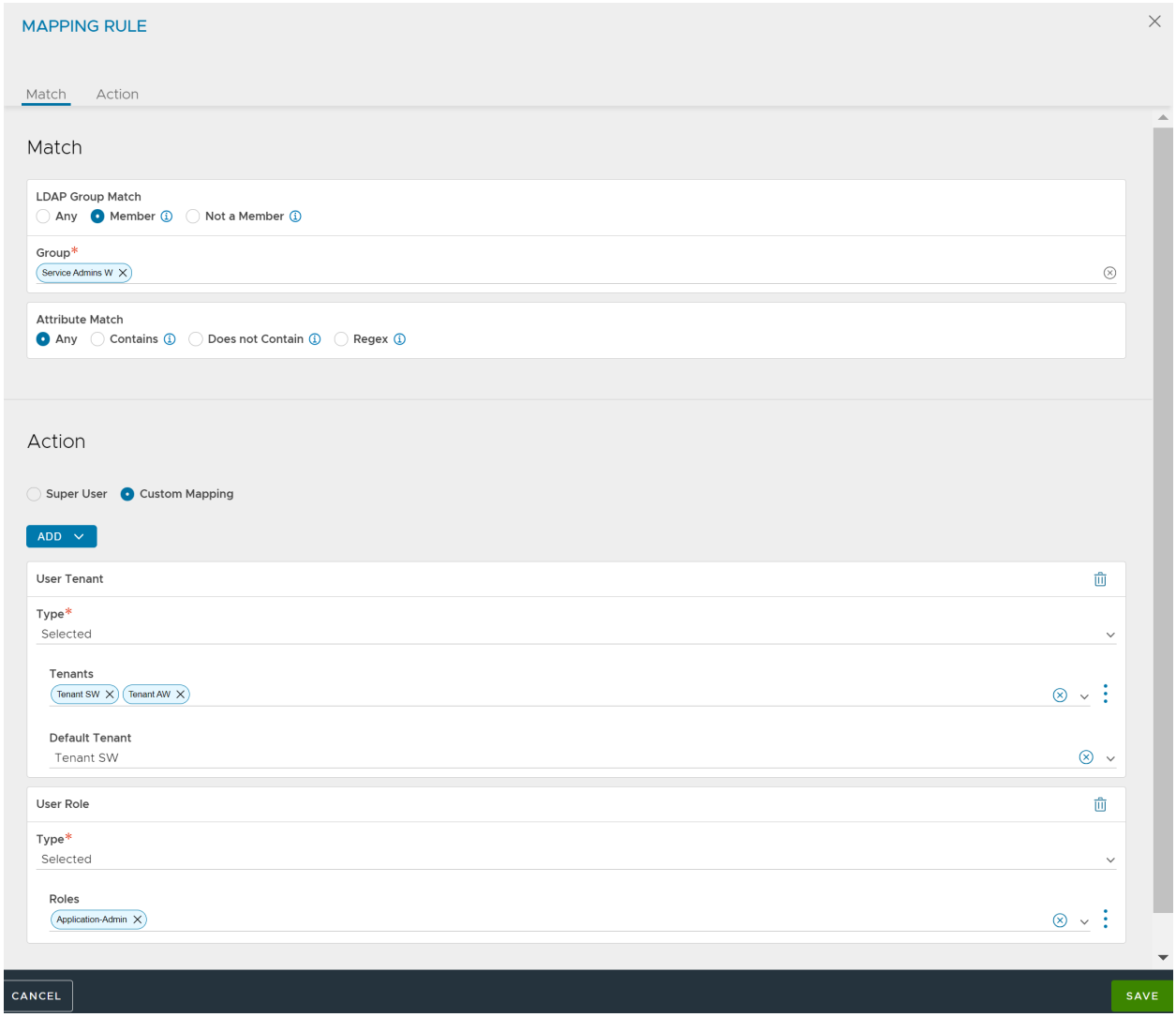
User Groups	Roles
Super Admins	The users can access all tenants and all settings.
Tenant Application Admins	The users have access to a few tenants — app owner for a few tenants.
Tenant Application Operators	The users have access to a few tenants — cannot modify anything.
Tenant Application Admins or Operators	The users have access to a few tenants as app owners and other tenants as app operator folks.

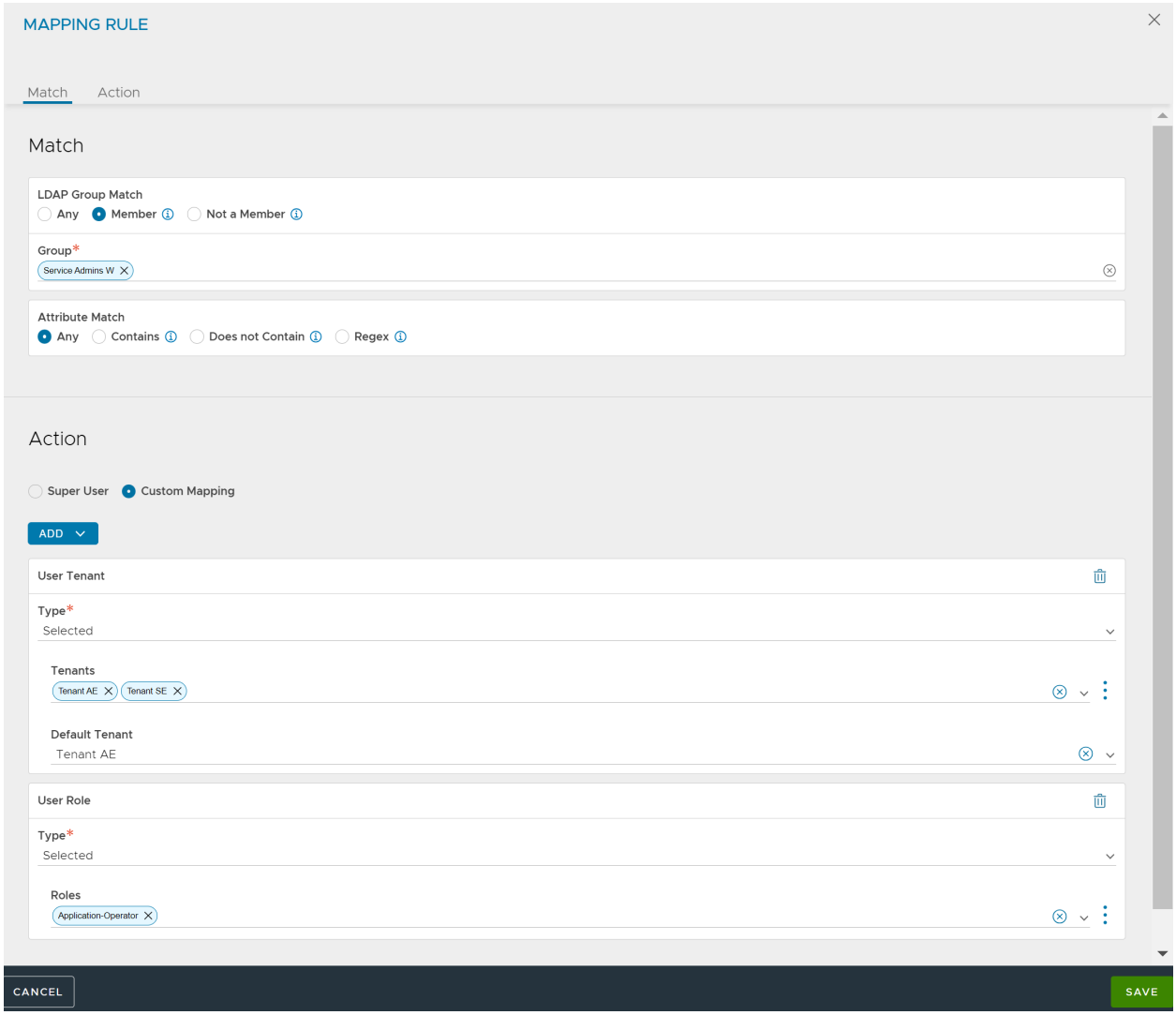


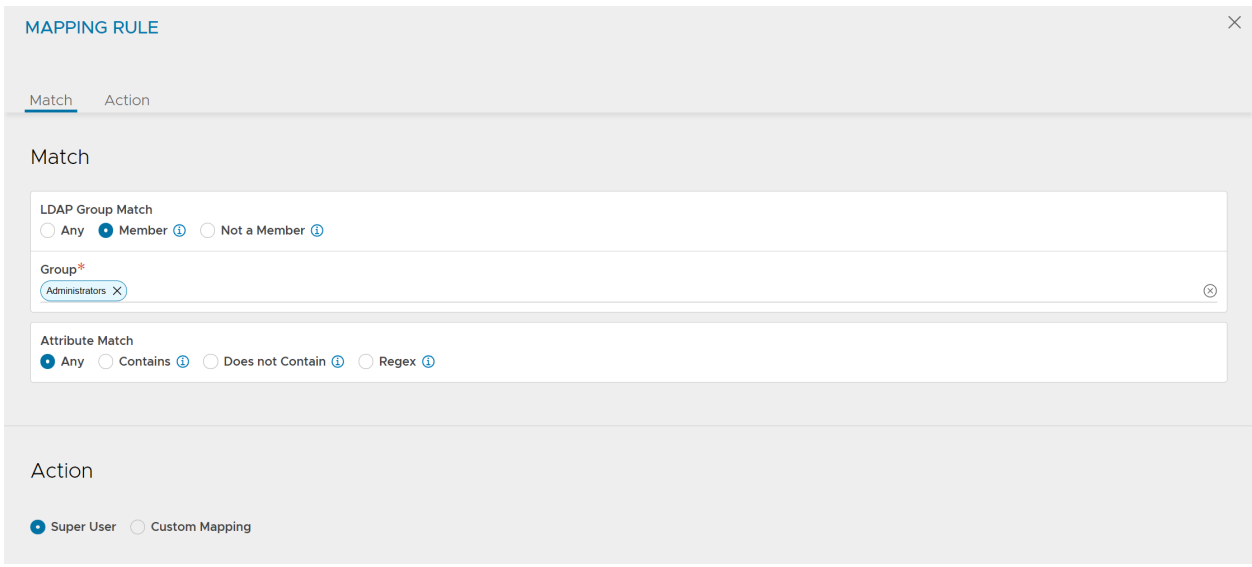

MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Admins E
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Tenant AE
Tenant SE
Default Tenant
Tenant AE
User Role
Type*
Selected
Roles
Application-Admin
CANCEL
SAVE

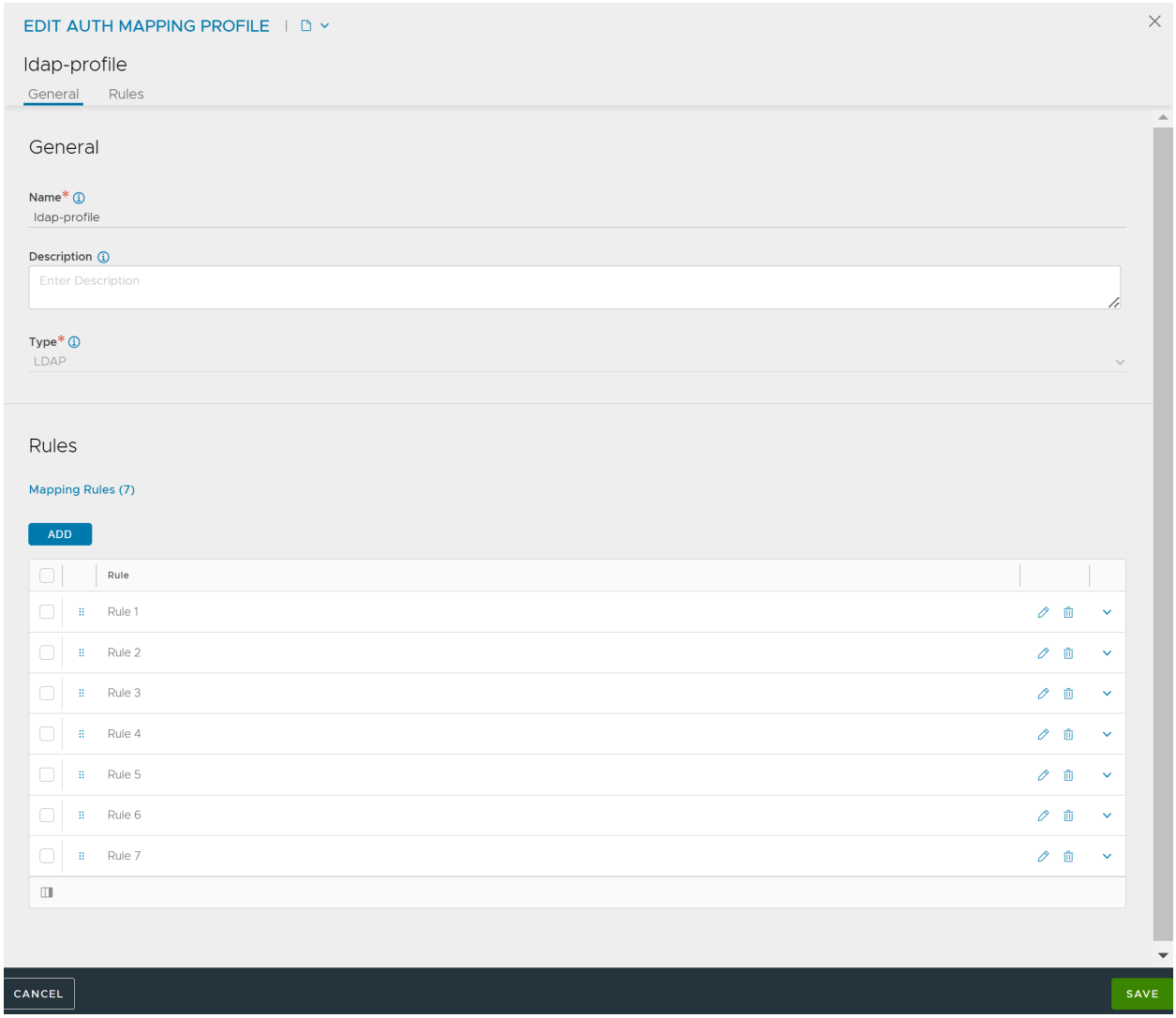

MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Operators W
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Tenant AW
Tenant SW
Default Tenant
Tenant AW
User Role
Type*
Selected
Roles
Application-Operator
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Operators W
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Tenant AW
Tenant SW
Default Tenant
Tenant AW
User Role
Type*
Selected
Roles
Application-Operator
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Admins W
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Tenant SW
Tenant AW
Default Tenant
Tenant SW
User Role
Type*
Selected
Roles
Application-Admin
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Admins W
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Tenant AE
Tenant SE
Default Tenant
Tenant AE
User Role
Type*
Selected
Roles
Application-Operator
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Administrators
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping

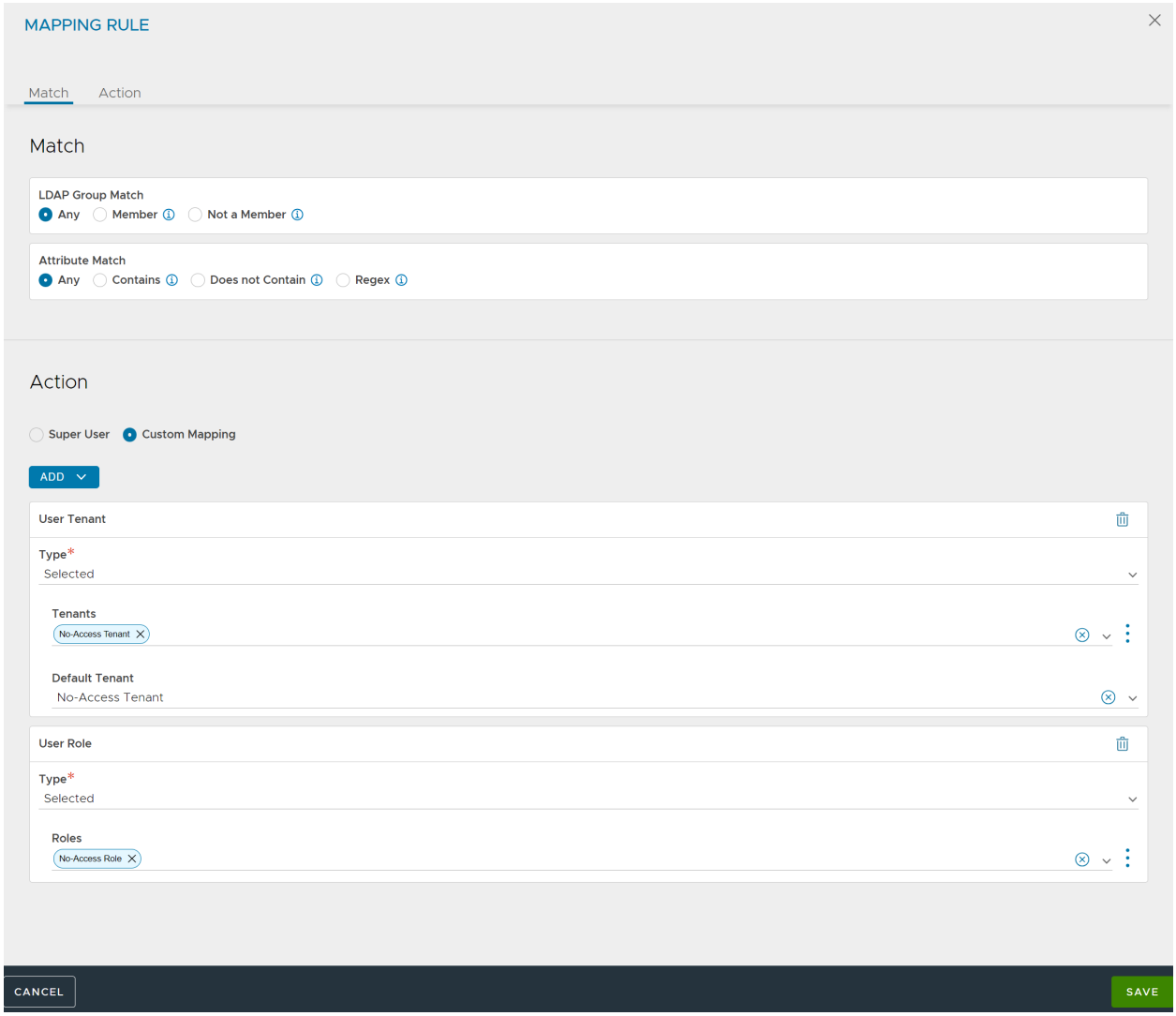


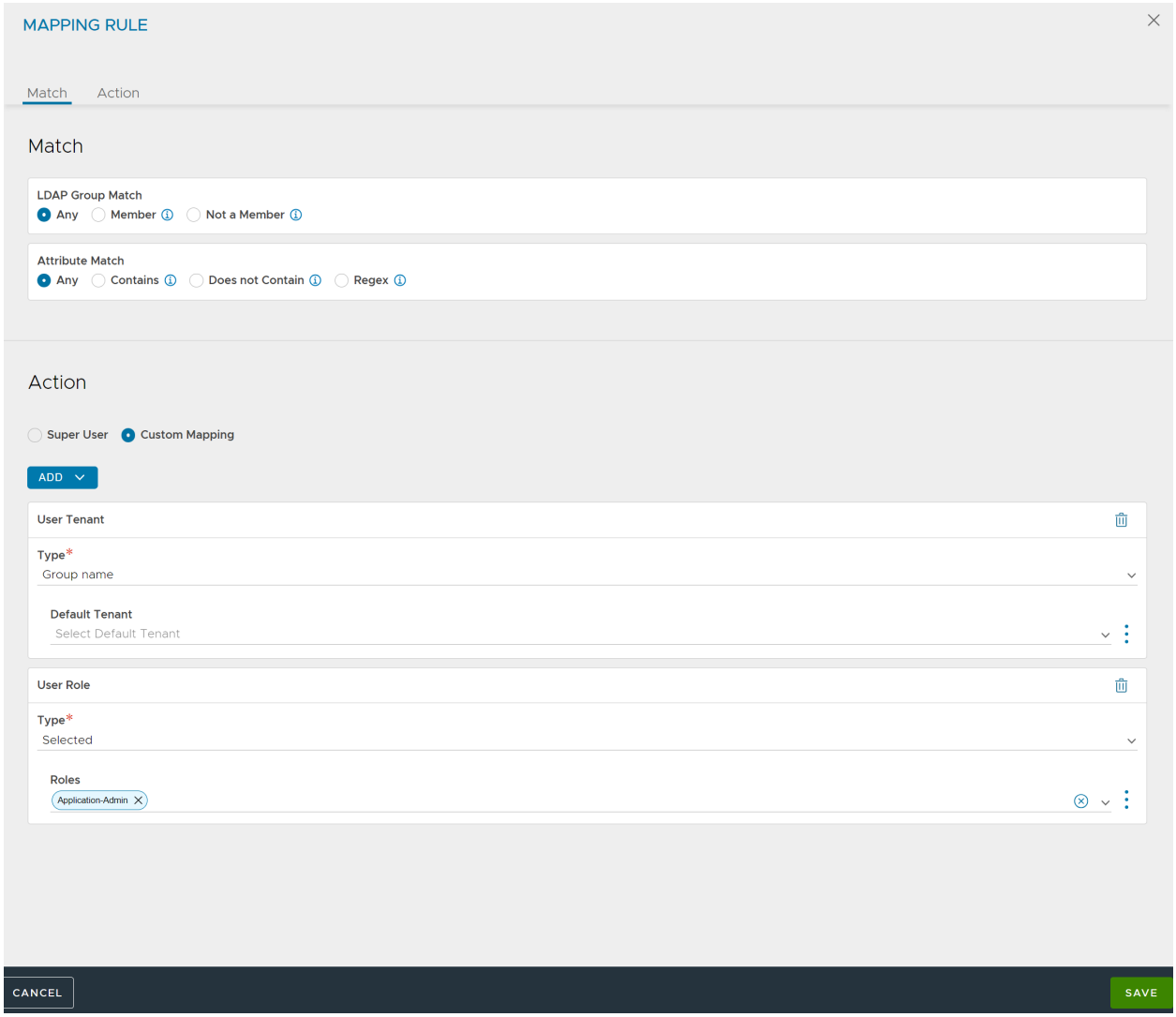
In this example, members of group Service Admins E have read or write access (Application-Admin role) in tenants Tenant AE and Tenant SE, whereas they have read-only access (Application-Operator role) in a few other tenants. Service Operators have only read-only access in their respective tenants.

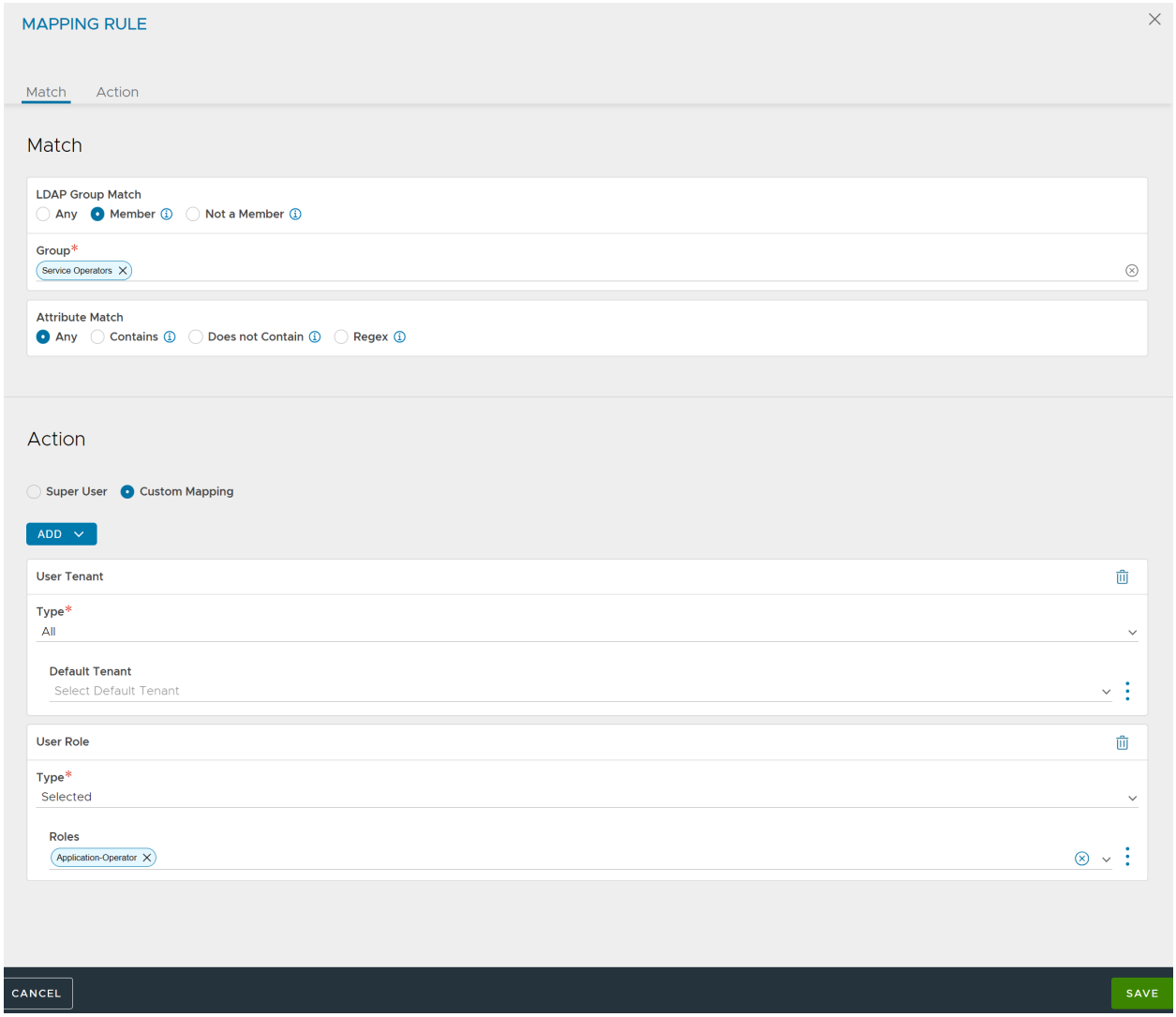
Example: Multiple Authorizations for a Single User

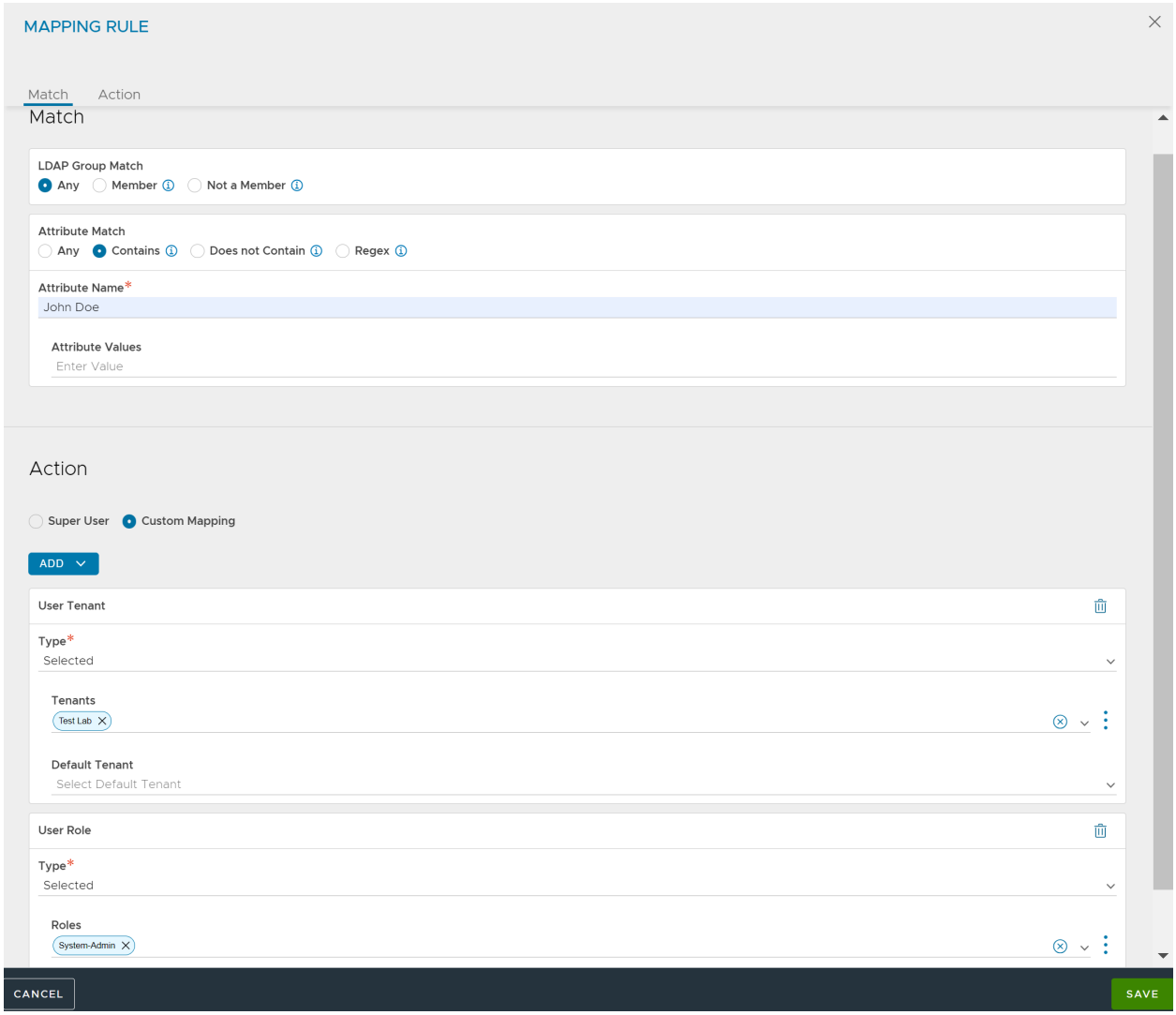
In this example, the login of user John Doe results in the user gaining access through multiple authorization mapping rules.

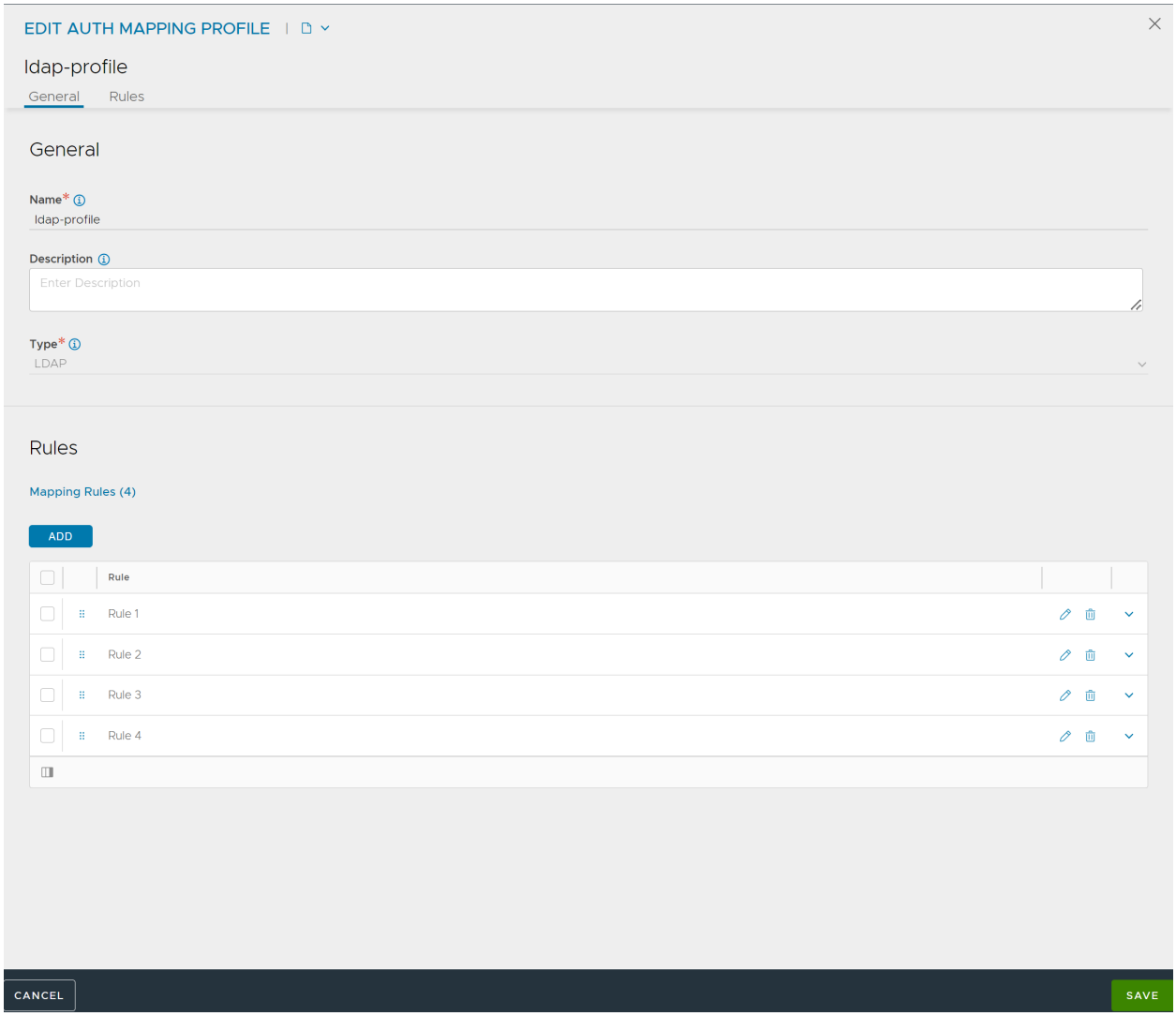
Multiple mapping rules are configured based on various group and attribute criteria.


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
No-Access Tenant
Default Tenant
No-Access Tenant
User Role
Type*
Selected
Roles
No-Access Role
CANCEL
SAVE

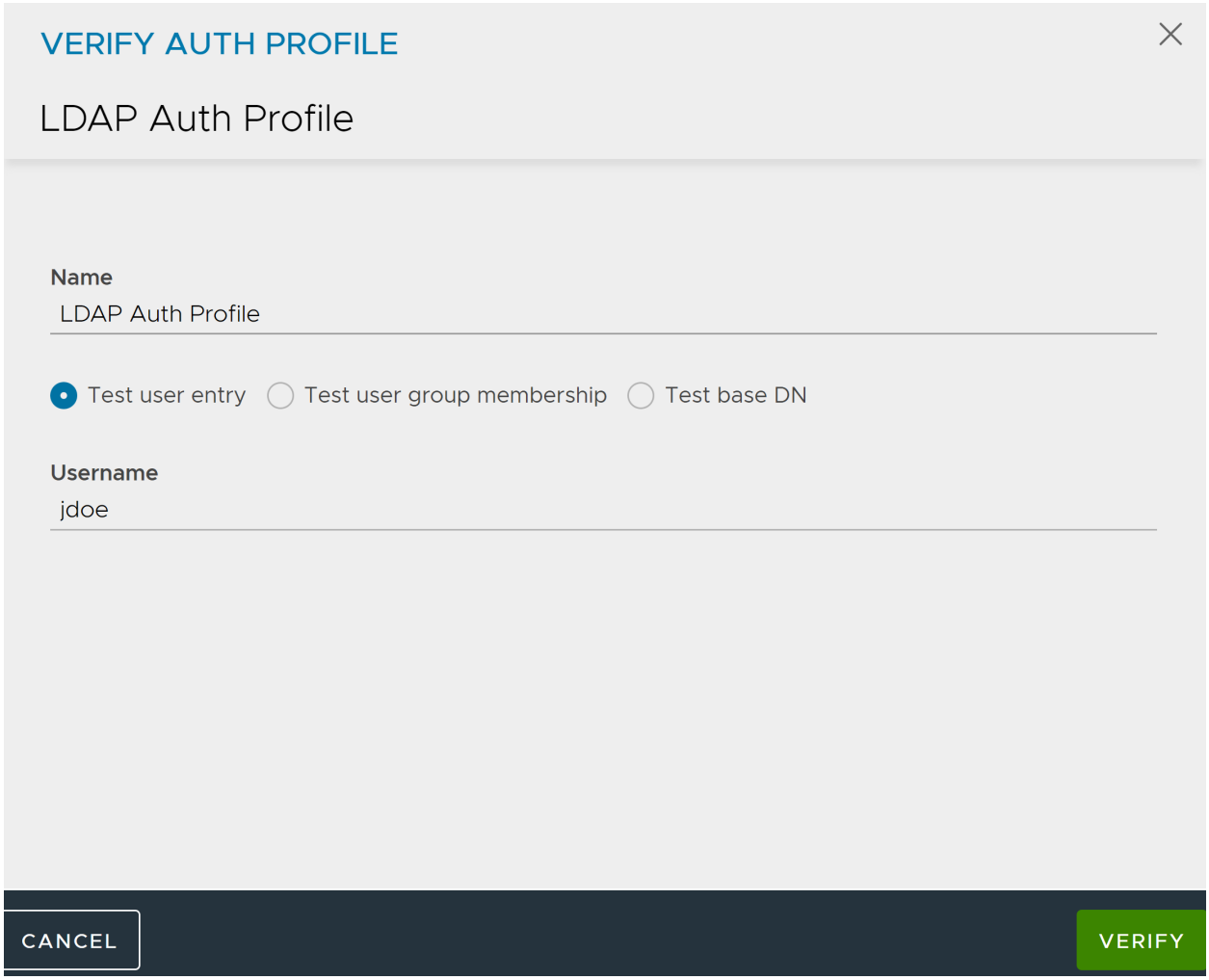

MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Group name
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
Application-Admin
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Operators
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
All
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
Application-Operator
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Attribute Name*
John Doe
Attribute Values
Enter Value
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Test Lab
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
System-Admin
CANCEL
SAVE



The LDAP server is configured with user John Doe.



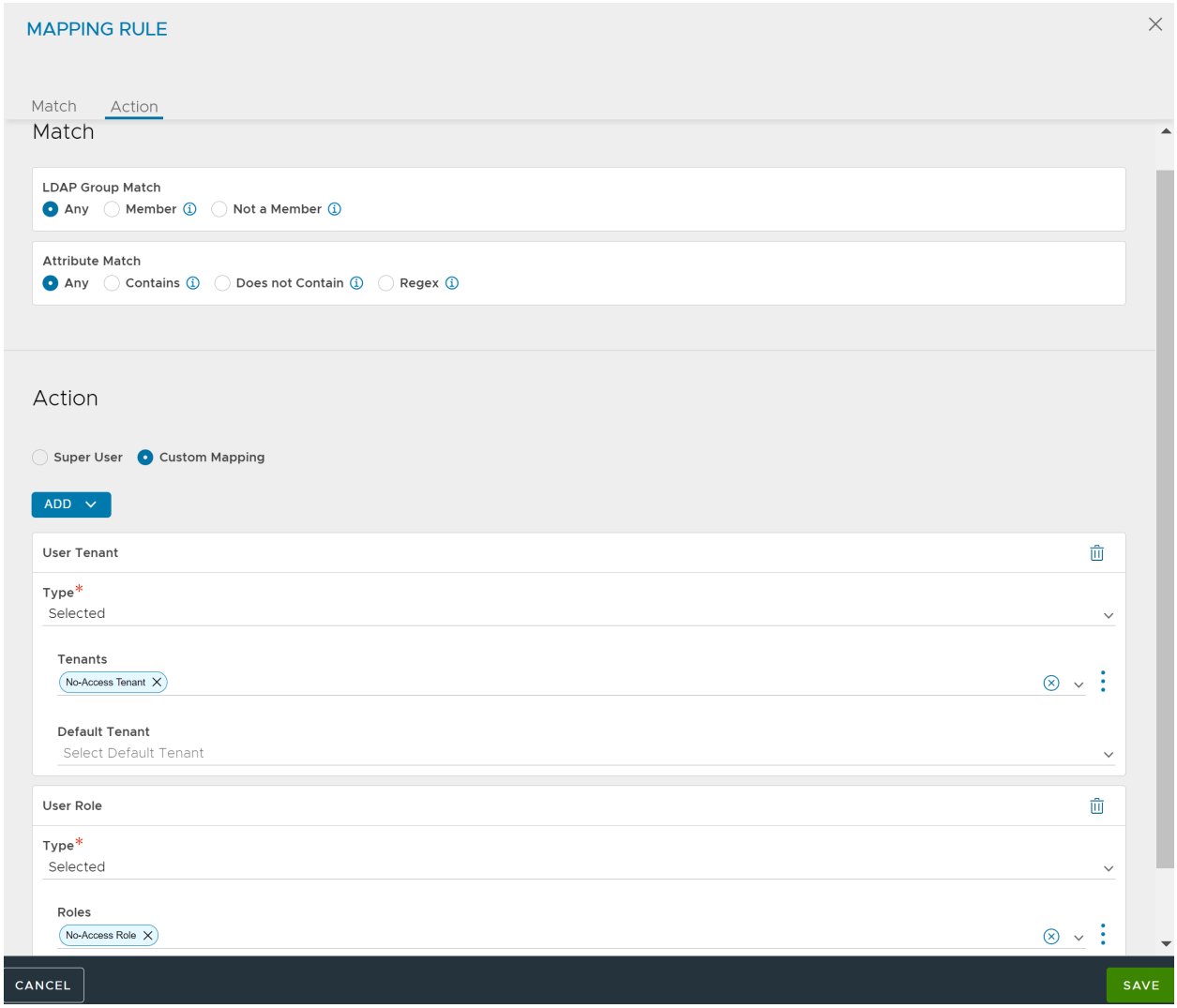
The LDAP server can be configured with John Doe as a member of the groups - Enterprise Admins and Service Operators.

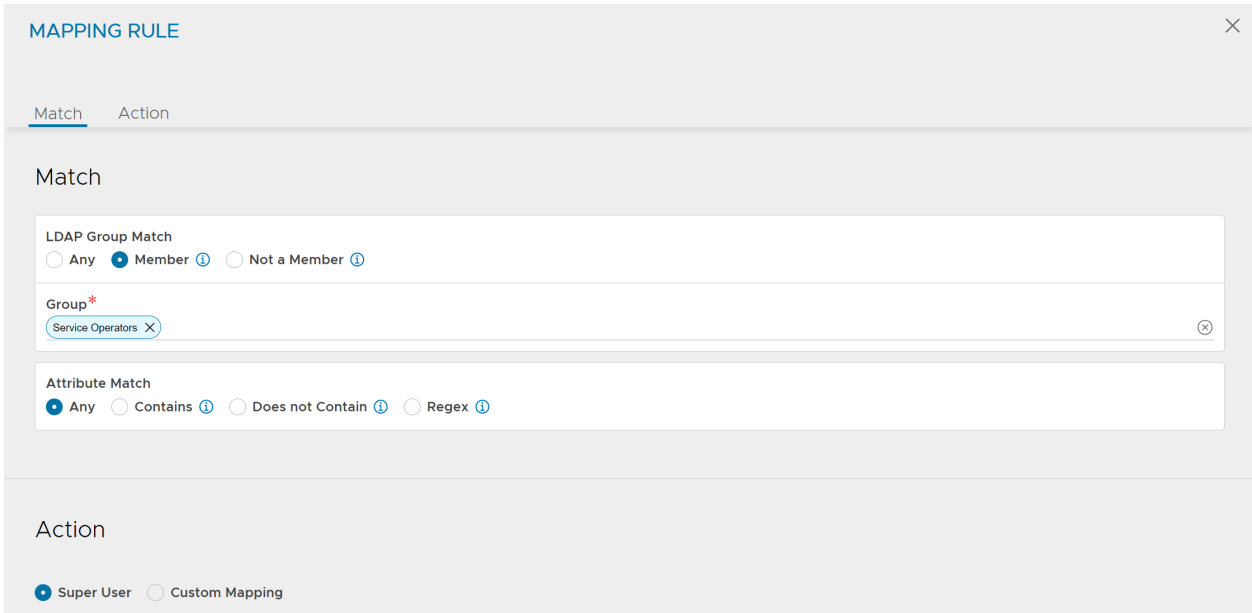
After the user John Doe logs in, and all authorization rules are applied to the user session. Multiple role or tenant combinations are used to determine user privileges during user login. The user record shows the user successfully matched all four rules and role or tenant pairs were appropriately applied.

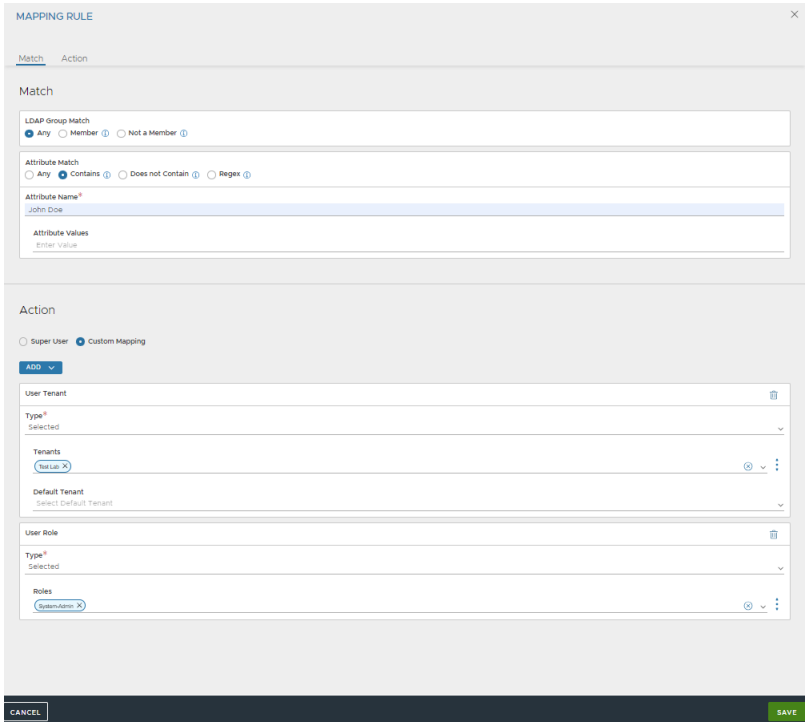
Example: Multiple Authorizations Resulting in a Super User

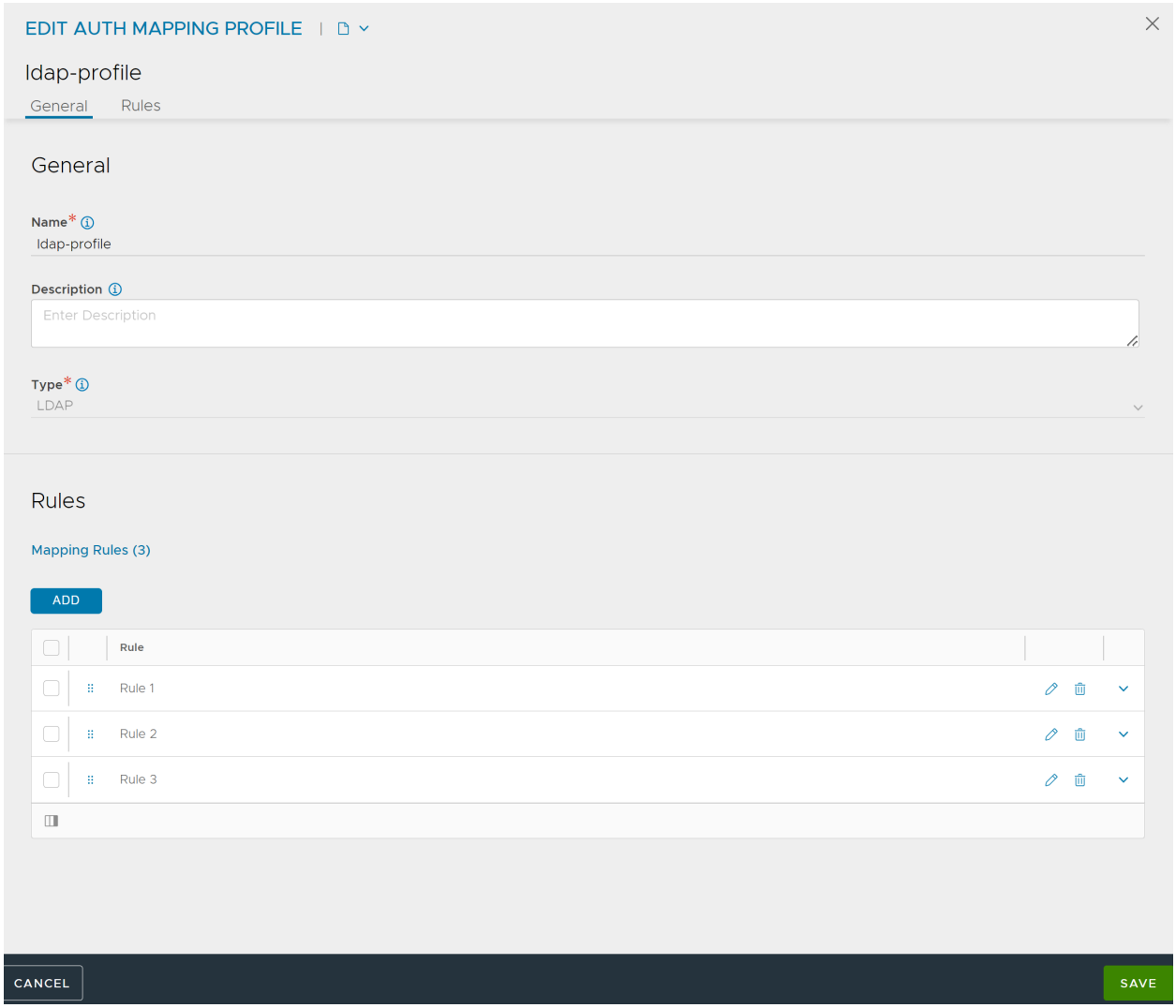
In this example, the login of user John Doe results in the user becoming a super user.

Mapping rules make a member of the group Service Operators a super user.


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
No-Access Tenant
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
No-Access Role
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Service Operators
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Attribute Name*
John Doe
Attribute Values
Enter Value
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Test Lab
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
System-Admin
CANCEL
SAVE

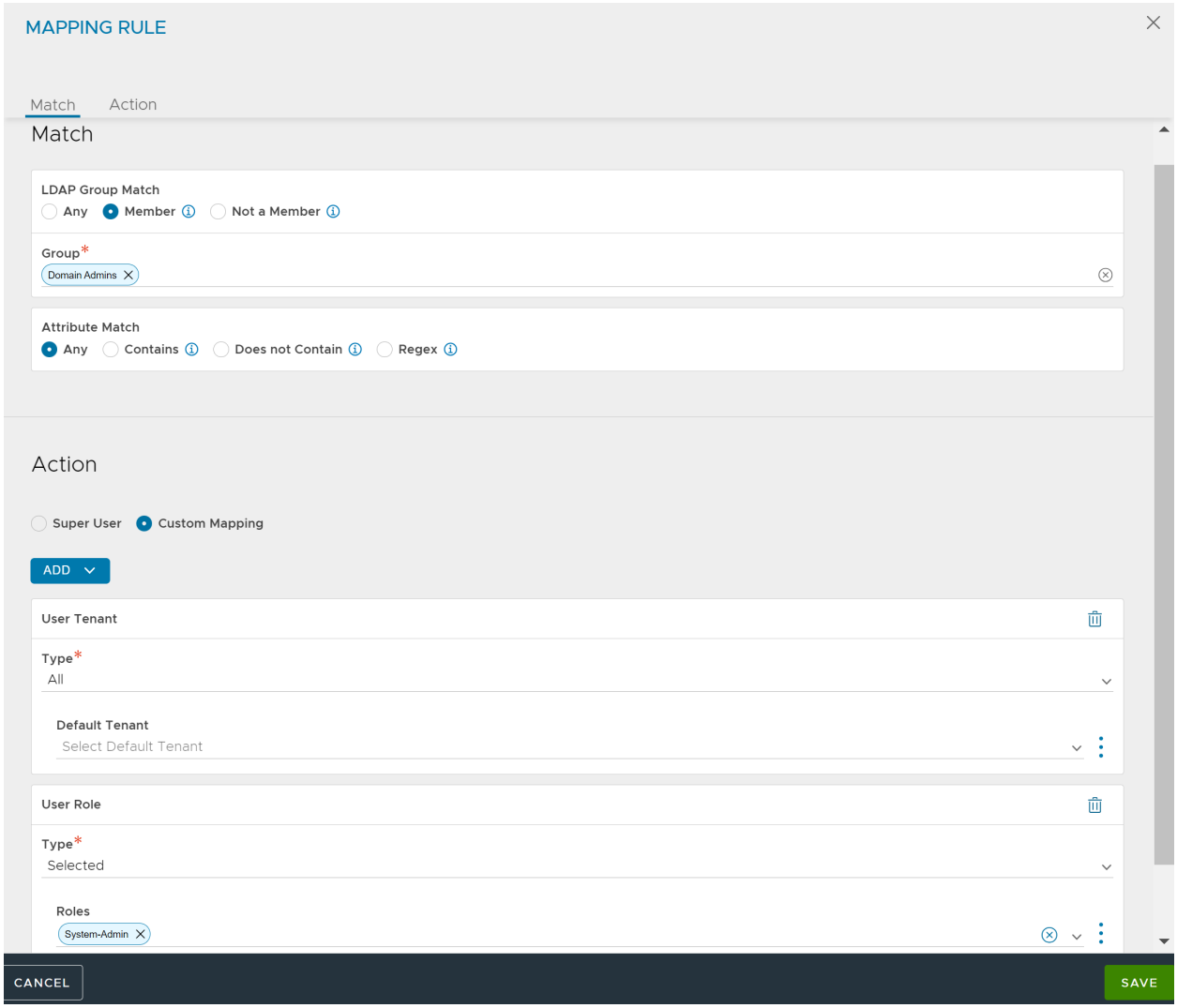


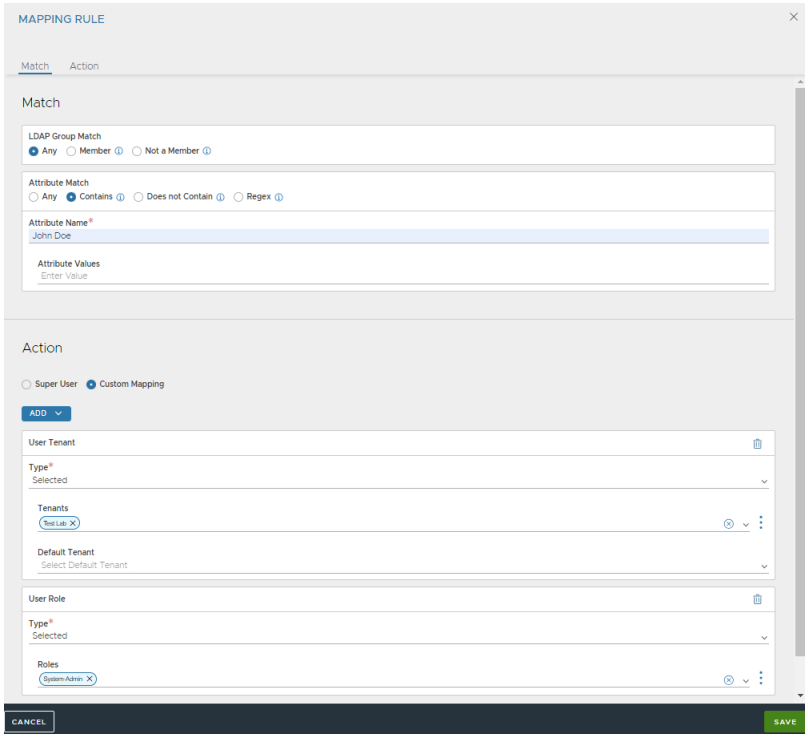
Due to the super user access, user John Doe gets access to all tenants with every role.

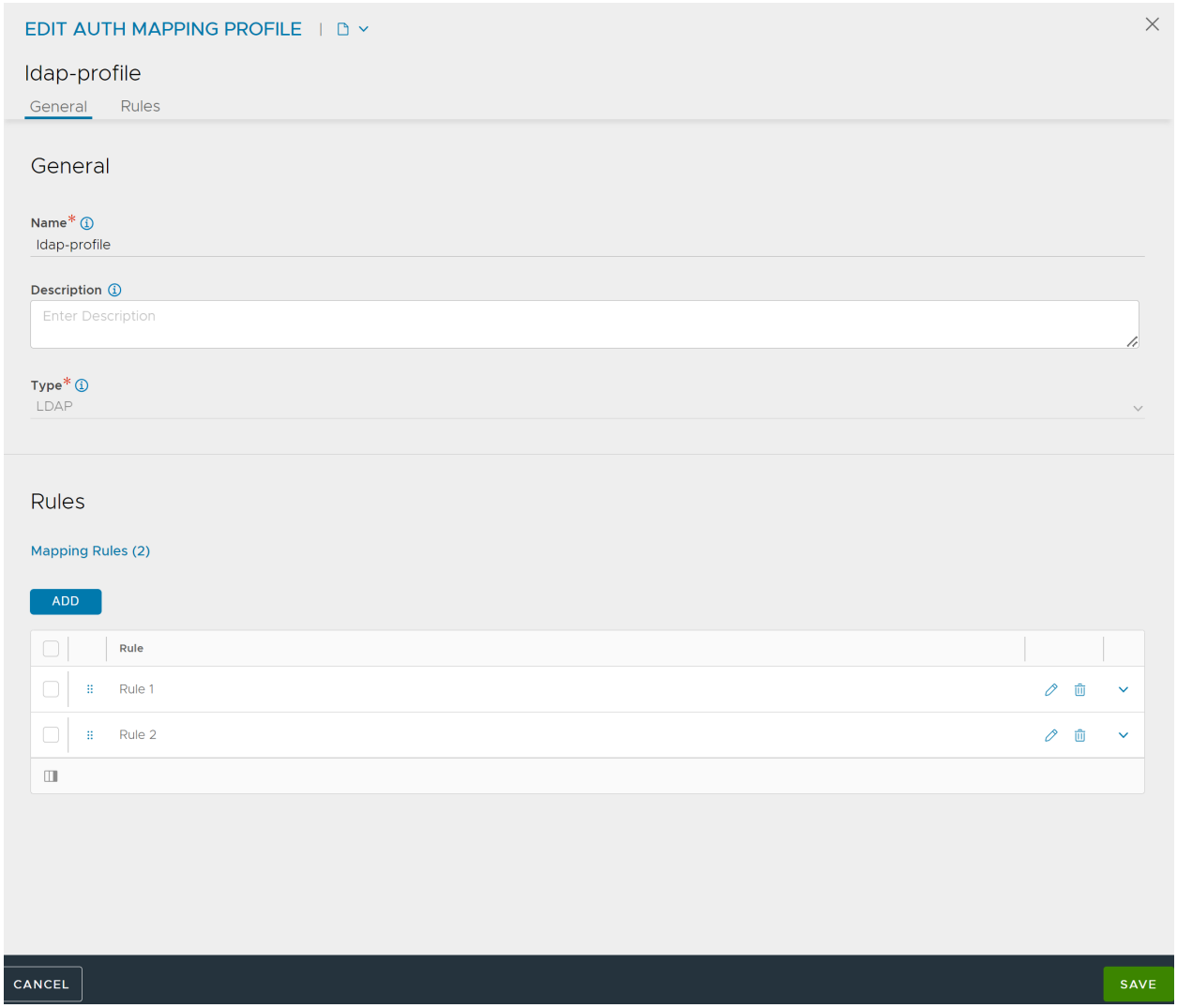
Example: No Authorizations for a Single User

In this example, the login of user John Doe results in the user not getting any roles or tenants.

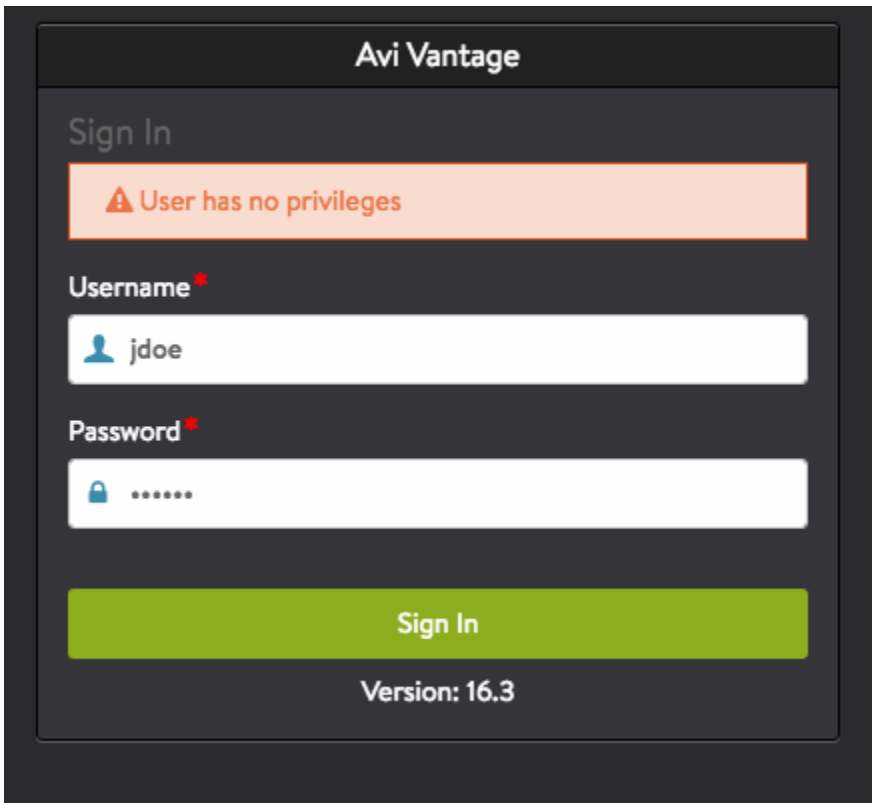
Mapping rules are updated to keep user John Doe from having any privileges.


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Group*
Domain Admins
Attribute Match
Any
Contains
Does not Contain
Regex
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
All
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
System-Admin
CANCEL
SAVE


MAPPING RULE
Match
Action
Match
LDAP Group Match
Any
Member
Not a Member
Attribute Match
Any
Contains
Does not Contain
Regex
Attribute Name*
John Doe
Attribute Values
Enter Value
Action
Super User
Custom Mapping
ADD
User Tenant
Type*
Selected
Tenants
Test Lab
Default Tenant
Select Default Tenant
User Role
Type*
Selected
Roles
System-Admin
CANCEL
SAVE



When user John Doe logs in, the user interface reports no privileges to login.



User record does not have any access entries.

```
[: > show user jdoe
+----------------------+------------------------------------------+
| Field                | Value                                    |
+----------------------+------------------------------------------+
| uuid                 | user-2d4348f3-fcf6-4af3-9c8b-ff54476ad7f6 |
| username             | jdoe                                     |
| name                 | jdoe                                     |
| email                |                                          |
| is_superuser         | False                                    |
| last_login_ip        | 10.10.221.128                            |
| last_login_timestamp | 2016-08-07 06:08:42                      |
| logged_in            | True                                     |
| local                | False                                    |
| full_name            | John Doe                                 |
+----------------------+------------------------------------------+
```

Dynamic Tenant and Role Mappings

NSX Advanced Load Balancer supports dynamically mapping tenants and/or roles based on regular expression matching against the user's group memberships or user LDAP attributes.

The mapping is achieved using regex Named Capture Groups of the form `(?P{tenant|role}<match>)`.

In the following example, the user is a member of the LDAP groups `lb_app1234_admin` and `lb_app7890_admin`, where `app1234` and `app7890` correspond to the names of tenants defined in NSX Advanced Load Balancer. The example will grant the user the specific Tenant-Admin role within these tenants.

```
configure authmappingprofile authmapping
[admin:avicontroller]: > configure authmappingprofile authmapping
[admin:avicontroller]: authmappingprofile> mapping_rules index 1
New object being created
[admin:avicontroller]: authmappingprofile:mapping_rules> group_match
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> criteria auth_match_regex
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> groups "lb_(?P{tenant}
\w+)_admin"
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> save
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_tenant
assign_matching_group_regex
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_role assign_from_select_list
[admin:avicontroller]: authmappingprofile:mapping_rules> role_refs Tenant-Admin
[admin:avicontroller]: authmappingprofile:mapping_rules> save
```

Roles may also be dynamically mapped using the `?P{role}` Named Capture Group. The example below will map a user that is a member of a group named `lb_app1234_appowner` to the role `appowner` and `tenant app1234`:

```
configure authmappingprofile authmapping
[admin:avicontroller]: > configure authmappingprofile authmapping
[admin:avicontroller]: authmappingprofile> mapping_rules index 1
New object being created
[admin:avicontroller]: authmappingprofile:mapping_rules> group_match
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> criteria auth_match_regex
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> groups "lb_(?P{tenant}
\w+)_(?P{role}\w+)"
[admin:avicontroller]: authmappingprofile:mapping_rules:group_match> save
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_tenant
assign_matching_group_regex
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_role
assign_matching_group_regex
[admin:avicontroller]: authmappingprofile:mapping_rules> save
```

Similarly, regex-based mapping based on the value of an LDAP attribute rather than group membership is also supported. The example below will map a user with LDAP attribute `business_unit` with value `bu_sales` to the tenant sales:

```
configure authmappingprofile authmapping
[admin:avicontroller]: > configure authmappingprofile authmapping
[admin:avicontroller]: authmappingprofile> mapping_rules index 1
New object being created
[admin:avicontroller]: authmappingprofile:mapping_rules> attribute_match
[admin:avicontroller]: authmappingprofile:mapping_rules:attribute_match> criteria
auth_match_regex
[admin:avicontroller]: authmappingprofile:mapping_rules:attribute_match> name business_unit
[admin:avicontroller]: authmappingprofile:mapping_rules:attribute_match> values "bu_(?
```

```
P{tenant}\w+)"
[admin:avicontroller]: authmappingprofile:mapping_rules:attribute_match> save
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_tenant
assign_matching_attribute_regex
[admin:avicontroller]: authmappingprofile:mapping_rules> tenant_attribute_name business_unit
[admin:avicontroller]: authmappingprofile:mapping_rules> assign_role assign_from_select_list
[admin:avicontroller]: authmappingprofile:mapping_rules> role_refs Tenant-Admin
[admin:avicontroller]: authmappingprofile:mapping_rules> save
```

Note: Currently group-based regex matching can only be configured through the CLI/API.

User Profile Mapping

Using user profile mapping, it is possible to choose a user profile for remote users based on certain conditions.

Use the following CLI to configure the user profile:

```
[admin:123]: systemconfiguration> admin_auth_configuration
[admin:123]: systemconfiguration:admin_auth_configuration> mapping_rules index 1
[admin:123]: systemconfiguration:admin_auth_configuration:mapping_rules> assign_userprofile
assign_from_select_list
[admin:123]: systemconfiguration:admin_auth_configuration:mapping_rules> userprofile_ref
Tacacs-Userprofile
[admin:123]: systemconfiguration:admin_auth_configuration:mapping_rules> save
```

Note Ensure that the user profile is already created. For more information on how to create and configure a user profile, see User Profile.

View the user profile configuration as shown below.

```
[admin:123]: > show systemconfiguration
|----------------------------------|-------------------------------------|
|Field                             |    Value                            |
|----------------------------------|-------------------------------------|
|                          Truncated Output
|
| admin_auth_configuration         |                                     |
|   auth_profile_ref               | tacacs1                             |
|   mapping_rules[1]               |                                     |
|     index                        | 1                                   |
|     assign_tenant                | ASSIGN_FROM_SELECT_LIST             |
|     tenant_attribute_name        |                                     |
|     tenant_refs[1]               | admin                               |
|     assign_role                  | ASSIGN_FROM_SELECT_LIST             |
|     role_attribute_name          |                                     |
|     role_refs[1]                 | Application-Admin                   |
|     is_superuser                 | False                               |
|     assign_userprofile           | ASSIGN_FROM_SELECT_LIST             |
|     userprofile_ref              | Tacacs-Userprofile                  |
|   allow_local_user_login         | True                                |
|----------------------------------|-------------------------------------|
```

Maximum Concurrent Login Sessions

This feature prevents having more than a configurable number of concurrent sessions. The default value is set to 0, which means the concurrent session check is bypassed. Additional logins are prevented if the maximum concurrent session count has been reached. Beyond this, the user can log in using shell `--clear-sessions` which will invalidate all the active sessions.

The administrator controls this feature through NSX Advanced Load Balancer CLI or REST API. The setting for it is maintained within the UserAccountProfile object. By default, all the users in the system are attached to Default-User-Account-Profile, as shown below. The admin can create a new user account profile with different thresholds if required.

```
admin:10-10-24-52]: > show useraccountprofile Default-User-Account-Profile
+-------------------------------+--------------------------------------------------------------+
| Field | Value |
+-------------------------------+--------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+-------------------------------+--------------------------------------------------------------+
To change the maximum number of concurrent sessions:
[admin:10-10-24-52]: > configure useraccountprofile Default-User-Account-Profile
Updating an existing object. Currently, the object is:
[admin:10-10-24-52]: useraccountprofile> max_concurrent_sessions 5
Overwriting the previously entered value for max_concurrent_sessions
[admin:10-10-24-52]: useraccountprofile> save
+-------------------------------+--------------------------------------------------------------+
| Field | Value |
+-------------------------------+--------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 5 |
| credentials_timeout_threshold | 0 |
+-------------------------------+--------------------------------------------------------------+
```

If the threshold has been reached, the user can choose to invalidate existing sessions using the `--clear-sessions` option of the shell command.

```
root@10-10-24-52:/home/admin# shell
Login: admin
Password:
```

Caution Maximum concurrent session count has been reached. Clear the sessions using shell –clear-sessions

```
root@10-10-24-52:/home/admin/# shell -clear-sessions Login: admin Password:
[admin:10-10-24-52]: >
```

The sessions are considered invalid when one of the following is observed:

- The user's UI session(s) end, and a login screen is displayed.
- The user's CLI sessions will end with no indication on-screen. The next command typed will trigger the re-validation of a new CLI session, and the command will be executed.
- A REST API user's next API call will fail to validate. Or, if the REST API user is executing calls within a session, the session is ended.

Configure No-Lockout-User-Account-Profile on NSX Advanced Load Balancer

The main difference between the **Default-User-Account-Profile** and **No-Lockout-User-Account-Profile** is the value set for **Max Login Failure Count**. The **Max Login Failure Count** is the number of login attempts allowed before the lockout of the user account. By default, this value is set to `3` for the default profile.

For the no lockout user profile, **Max login Failure Count** is set to 0. It means that a user can have unlimited login failures without the risk of an account getting locked.

In GSLB deployments, it is recommended to use **No-Lockout-User-Account-Profile**. It prevents the locking of the user account due to various reasons. Sometimes, an admin account gets locked when one node of the GSLB pair tries to reach another node with the admin credentials, but the other node is not reachable.

Configuring using UI

To check or edit the attributes for **No-Lockout-User-Account-Profile**, navigate to **Administration > Accounts > User Profiles** and click the pencil icon on the right side of **No-Lockout-User-Account-Profile**.

Note You can use the existing **No-Lockout-User-Account-Profile** available or create a new one. **Max Login Failure Count** must be set to 0 for any profile to work like a **No-Lockout-User-Account-Profile**.

To create a new user, follow the below steps:

1 Login to the NSX Advanced Load Balancer Controller using admin credentials.

2 Navigate to **Administration > Accounts > Users**.

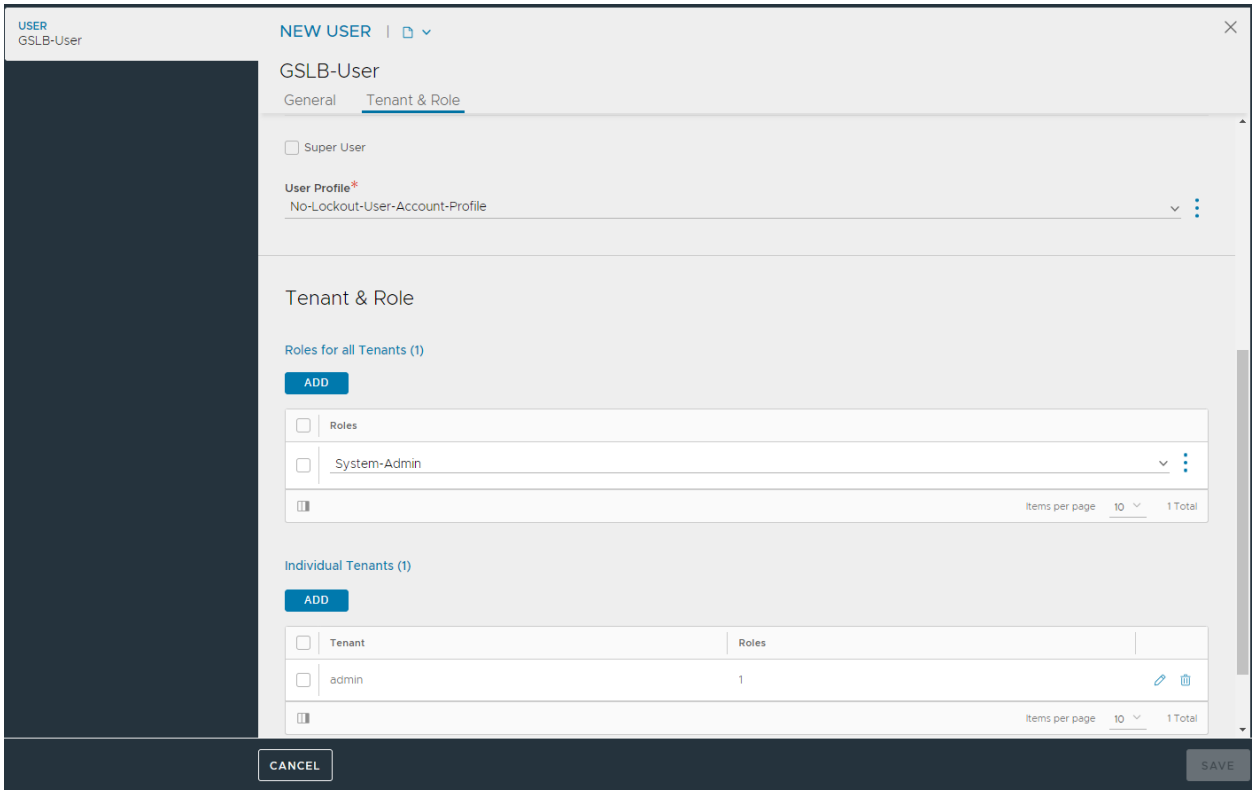
3 Click **Create**.

4 Provide the username of your choice.

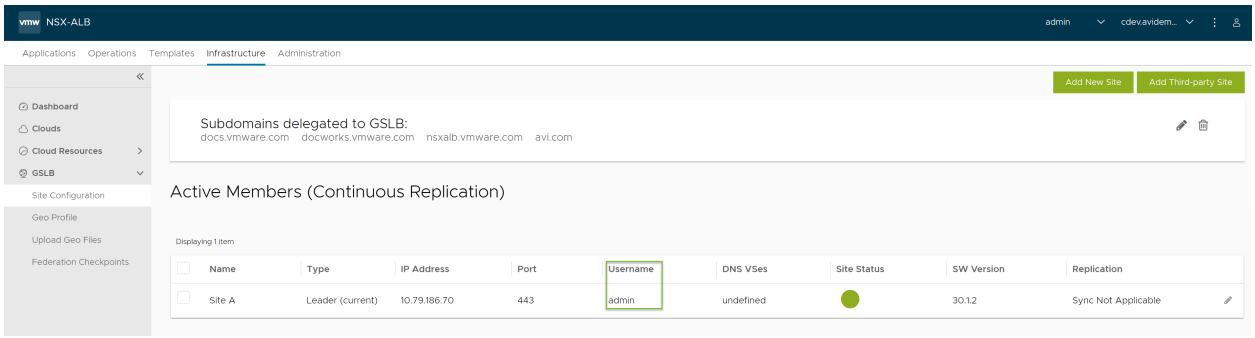
5 Select the **No-Lockout-User-Account-Profile** from the drop-down menu for **User Profile**.

6 Use desired **Tenant and Role** for this new user.

7 Click **Save**.



Note Use the same user created in the previous step while doing GSLB configuration. For example, the user **admin** must be replaced with the newly created user (GSLB-User) with **No-Lockout-User-Account-Profile**.



For more information, see GSLB Site Configuration.

User Accounts

A valid account is required for access to NSX Advanced Load Balancer through the web interface, REST API, or CLI. To configure or manage NSX Advanced Load Balancer user accounts, you need a user account with write access to the **Accounts** section. This is defined by the role assigned to the user account. The admin user account is a unique account used for initial setup of NSX Advanced Load Balancer.

Default System Accounts

By default, the NSX Advanced Load Balancer installation includes a number of pre-configured user accounts that serve specific purposes. Unlike custom user accounts, these accounts cannot be deactivated or removed.

admin

- The **admin** account exists on both the Controller and Service Engine of NSX Advanced Load Balancer.
- It is the default administrator user-name for the system and cannot be changed.
- The default shell is Linux bash.
  - From Linux prompt, use shell command to access the CLI shell.
  - admin is the only NSX Advanced Load Balancer account with its password automatically synchronized with Linux.
- **admin** account has the Super User Accounts field enabled in the Controller. Any changes to the admin account are not allowed.
- User is in the sudoers list.
- Default password for admin user - The initial default password of the Controller admin user is changed from admin to a strong password. This password is available in the portal where release images are uploaded and is accessible only to customers having an account on the portal. Additionally, SSH access to the Controller with this default password is not allowed until the user changes the default password of the admin user. Once the password is changed, SSH access to the admin user is permitted. For more information on default password, see Strong Default Admin Password.
  - Password is synchronized to the SEs.
- Account has super-user status with full access to all tenants.

- This account is always authenticated through the local user-db. It does not use any configured remote authentication.

cli

- This account is used to launch the CLI shell by logging into the Controller. Users will SSH to a Controller IP address, use **`cli`** as the username at the Linux prompt, and then be presented with the NSX Advanced Load Balancer CLI shell access username and password prompt, which requires their custom credentials.
- It is password-less from the Linux perspective with the CLI shell as the default shell that prompts for a user name/password.
- CLI shell is launched in a container with no persistent storage.

aviseuser

- This account exists on the Controller and SE.
- It is the internal user for SE-to-Controller communication through SSH tunnel.
- No password is required. It uses unique key-pair per SE.
- User is not in the sudoers list on the Controller.
- User is in the sudoers list on the SEs.

avictlruser

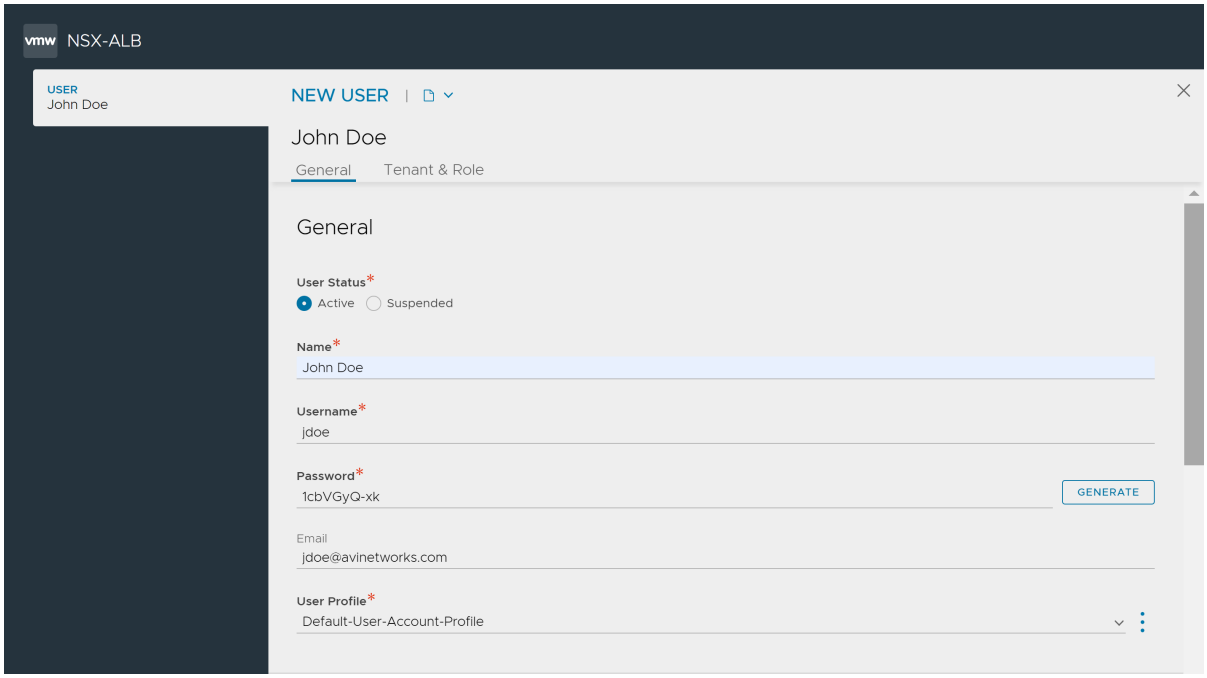
- This account exists on Controller only.
- It is the internal user for Controller-to-Controller communication through SSH.
- No password is required. It uses unique key-pair per Controller.
- User is in sudoers list.

Create a Local User Account

1 Navigate to **Administration > System Settings > EDIT**.

2 Under **Authentication**, select **Local**. For more information on configuring Access Settings, see Chapter 1 Web Interface Access Settings.

3 Navigate to **Administration > Accounts > Users** and click **Create**.



4 If the user requires the same privileges as the admin account, select the **Super User** check box. For more information, see Super User Accounts.

5 In the **Name** field, enter the full name of the user.

6 In the **Username** field, enter the name that the user will supply when signing into NSX Advanced Load Balancer, such as **`jdoe`** or **`jdoe@avinetworks.com`**.

7 In the **Password** field, either enter a case-sensitive password or click the **GENERATE** button to create a random password for the new user.

8 In the **Email** field, enter the email address of the user. This field is used when a user loses their password and requests to have it reset.

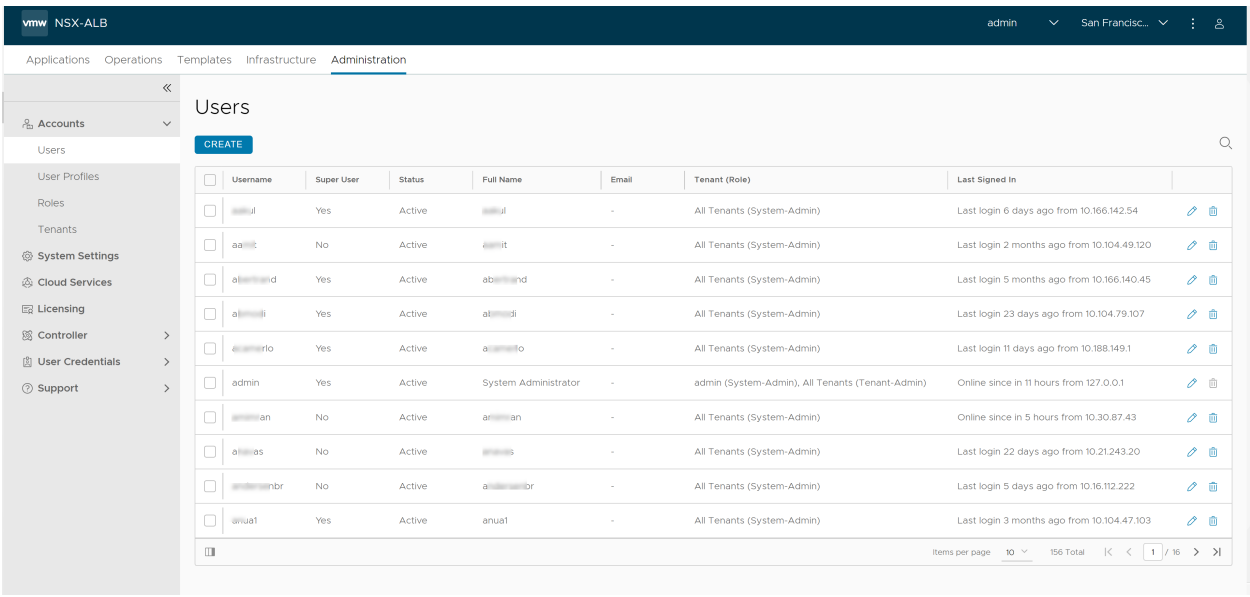
9 From the **User Profile** drop-down menu, select **Default-User-Account-Profile** or **No-Lockout-User-Account-Profile** as required. For more information, see User Profile.

10 Under **Tenant & Role**, select the areas of the NSX Advanced Load Balancer system to which the user account will be allowed access. For each system area, the role defines whether the user account has read, write, or no access. NSX Advanced Load Balancer comes with predefined roles. In addition, users who have write access to the Accounts section of NSX Advanced Load Balancer can customize the predefined roles and create new roles. For more information, see User Roles.

11 Click **Save**.

User Account Table

To view the NSX Advanced Load Balancer user accounts that are in its local user database, navigate to **Administration > Accounts > Users**.



For each local user account, the following information is listed:

Field	Description
Username	The account name used to log in to NSX Advanced Load Balancer through its management interfaces, that is, web interface, REST API, or CLI.
Super User	Super user access provides write access to all resources within NSX Advanced Load Balancer.
Status	Diplays the status of the user account, whether it is **Active** or **Suspended**.
Full Name	Full name of the user.
Email	Email address of the user.
Tenant (Role)	Access settings (write, read, or no access) for each type of resource within NSX Advanced Load Balancer.
Last Signed In	System time on the NSX Advanced Load Balancer Controller when the user most recently logged in.

Super User Accounts

The user accounts in NSX Advanced Load Balancer can be enabled for Super User access. Super User access provides write access to all resources within the NSX Advanced Load Balancer and also automatically provides access to all tenants.

The admin account that is created during the installation of the Controller automatically has Super User access enabled.

Optionally, super user access can be enabled in other user accounts, on an individual basis.

To enable super access for a NSX Advanced Load Balancer user account, select the **Super User** check box in the account configuration.

A non-admin user that is also a super-user can be associated with the NSX Advanced Load Balancer Controller by using the `attach <Avi Controller IP>` command. This provides the Controller container access to the user as an `avidebuguser`. The `avidebuguser` is also a sudo user. Attach option is available only if the local or remote user is configured as a super-user.

For detailed information about configuring user accounts, see Create a User Account.

SSH Access for the Super User

During debugging, there is a need to access different NSX Advanced Load Balancer entities such as Controllers and Service Engines. The NSX Advanced Load Balancer includes a mechanism whereby the Controller can self-authenticate a super-user access to any entity in the system.

A user can attach to the Service Engine using the attach `serviceengine <se_ip>` command. The admin user is authenticated using a token and logs into the SE as admin.

A sample illustration is as follows:

```
login as: admin
Pre-authentication banner message from server:
|
| Avi Cloud Controller
|
| Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
| All rights reserved.
|
| Management:   10.10.10.1/22              UP
| Gateway:      10.79.186.1                   UP
|
| Controller 1 demo environment
|
End of banner message from server
admin@10.79.187.241's password:
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Wed Aug 17 14:32:15 2022 from 10.16.113.99
admin@10-79-187-241:~$
```

Any non-admin super-user is also authenticated using tokens and logs into the SE as *avidebuguser*. This user also has super-user privileges on the SE.

A sample illustration is as follows:

```
login as: avidebuguser
Pre-authentication banner message from server:
|
| Avi Cloud Controller
```

```
|
| Avi Networks software, Copyright (C) 2013-2017 by Avi Networks, Inc.
| All rights reserved.
|
| Management:   10.10.10.1/22               UP
| Gateway:      10.79.186.1                    UP
|
| Controller 1 demo environment
|
End of banner message from server
admin@10.79.187.241's password:
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
Last login: Wed Aug 17 14:32:15 2022 from 10.16.113.99
avidebuguser@10-10-26-42:~$
```

SSH Users and Keys

The Controller and SEs use SSH for secure management communication. The communication falls into two categories, Controller-to-SE, and administrator-to-Controller.

Controller-to-SE Communication

This requires an SSH user who exists on both the Controller and the SEs, and a copy of the SSH user's public key on the NSX Advanced Load Balancer SEs. Though SSH setup is automated for some installation types such as installation into VMware with write access, other installation types require manual setup of these SSH resources. For more information, see Public Key Management on SE Hosts topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

Create SSH User

To create an SSH user on the NSX Advanced Load Balancer Controller, use the following steps.

Note An SSH user and key that already exists can be used. They must still be added to the Controller using these steps. When creating the user account, the existing key for the user can be added by copying and pasting it or by importing the key file.

1 Navigate to **Administration > Accounts > User Credentials**.

2 Click **Create**.

3 Enter the SSH user name (for example, root) in the **Name** field.

4 Select **SSH** as **Credentials Type**.

5 In **Authentication** section, select **SSH Key**.

6 In **Keys** section, select **Generate SSH Key Value Pair** to create a key pair for the user.

7 Click **Generate and Save**.

Note If the user already exists, you can add the user to the NSX Advanced Load Balancer by entering the user name, selecting **Import Private Key**, or either copying and pasting or importing the key file.

Download Public Key File

After creating an SSH user on the NSX Advanced Load Balancer Controller, perform the following steps to download the user public key from the Controller.

1 Navigate to **Administration > Accounts > User Credentials**.

2 Click the three dots next to the SSH user and select **Download Public Key**.

Default System Accounts

By default, the NSX Advanced Load Balancer installation includes a number of pre-configured user accounts that serve specific purposes. Unlike custom user accounts, these accounts cannot be deactivated or removed.

admin

- The **admin** account exists on both the Controller and Service Engine of NSX Advanced Load Balancer.
- It is the default administrator user-name for the system and cannot be changed.
- The default shell is Linux bash.
  - From Linux prompt, use shell command to access the CLI shell.
  - admin is the only NSX Advanced Load Balancer account with its password automatically synchronized with Linux.
- **admin** account is associated with super-user role in the Controller.
- User is in the sudoers list.
- Default password for admin user - The initial default password of the Controller admin user is changed from admin to a strong password. This password is available in the portal where release images are uploaded and is accessible only to customers having an account on the portal. Additionally, SSH access to the Controller with this default password is not allowed until the user changes the default password of the admin user. Once the password is changed, SSH access to the admin user is permitted. For more information on default password, see Strong Default Admin Password.
  - Password is synchronized to the SEs.
- Account has super-user status with full access to all tenants.

- This account is always authenticated through the local user-db. It does not use any configured remote authentication.

cli

- This account is used to launch the CLI shell by logging into the Controller. Users will SSH to a Controller IP address, use **`cli`** as the username at the Linux prompt, and then be presented with the NSX Advanced Load Balancer CLI shell access username and password prompt, which requires their custom credentials.
- It is password-less from the Linux perspective with the CLI shell as the default shell that prompts for a user name/password.
- CLI shell is launched in a container with no persistent storage.

aviseuser

- This account exists on the Controller and SE.
- It is the internal user for SE-to-Controller communication through SSH tunnel.
- No password is required. It uses unique key-pair per SE.
- User is not in the sudoers list on the Controller.
- User is in the sudoers list on the SEs.

avictlruser

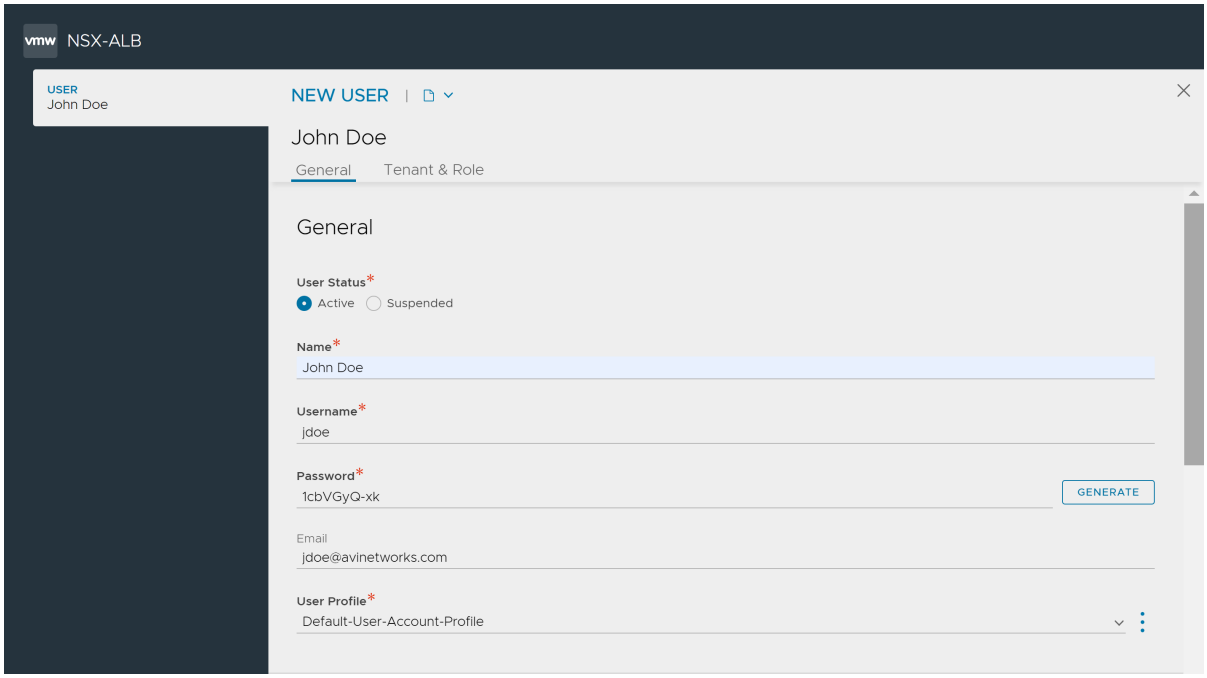
- This account exists on Controller only.
- It is the internal user for Controller-to-Controller communication through SSH.
- No password is required. It uses unique key-pair per Controller.
- User is in sudoers list.

Create a User Account

The steps to create a user account are as follows.

Procedure

1 Navigate to **Administration > Accounts > Users** and click **Create**.



2 In the **Name** field, enter the full name of the user.

3 In the **Username** field, enter the name that the user will supply when signing into NSX Advanced Load Balancer, such as jdoe or jdoe@avinetworks.com.

4 In the **Password** field, either enter a case-sensitive password or click the **Generate** button to create a random password for the new user.

5 In the **Email** field, enter the email address of the user. This field is used when a user loses their password and requests to have it reset. See Password Recovery.

6 In the **Role** field, select the areas of the NSX Advanced Load Balancer system to which the user account will be allowed access. For each system area, the role defines whether the user account has read, write, or no access. NSX Advanced Load Balancer comes with predefined roles. In addition, users who have write access to the Accounts section of NSX Advanced Load Balancer can customize the predefined roles and create new roles. For more information, see User Roles.

7 If the user requires the same privileges as the admin account, select the **Super User** check box.

8 Click **Save**.

User Account Security

The NSX Advanced Load Balancer includes several mechanisms to protect the control plane from overload or abuse. Some of these are based on feedback from penetration test findings.

Details about these mechanisms are discussed in the following sections:

- Super User Accounts

- Strong Password Enforcement
- Password History Enforcement
- Password-less SSH-based Login for Admin User
- User Account Lockout
- Maximum Concurrent Login Sessions
- User Credentials Timeout

User Account Self-Service

This topic explains the steps to configure the user account.

Editing User Account

When logged into the web interface, an NSX Advanced Load Balancer user can view or change information about their own account.

This applies only to accounts in local user database of NSX Advanced Load Balancer.

To configure your user account, use the following steps.

1 Log in to the web interface.

2 Click the **user icon** in the top, right corner of the application.

3 Click **My Account**.

4 In the **EDIT MY ACCOUNT** screen, enter the **Name** and **Username**.

5 Enter the **Email** address. This address is used by NSX Advanced Load Balancer when you request a password recovery.

6 To change the password, enter the **Old Password**, the **New Password**, and re-enter the **New Password** for confirmation.

7 Select the **Time Zone**. The format can be **UTC Time** or **Local Time**. The NSX Advanced Load Balancer obtains the time from the specified NTP server. This format is used in the timestamps that appear in logs or metrics.

8 Select the **Language** of display.

Note The UI supports Japanese and Simplified Chinese along with the default option English.

9 Select **Enable Tech Preview for Virtual Service Logs** checkbox to replace the existing legacy view. This option is deselected by default.

10 Under **Session Timeout**, enter the maximum number of minutes a management session can remain idle before being terminated by the Controller.

11 Under **DNS Refresh Period**, enter the period for refresh pool and GSLB DNS job in minutes.

12 Click **Save**.

Configuring Idle Timeout for Admin Access to GUI and API

By default, the Controller terminates idle connections to the GUI and API. You can configure this idle connection timeout value as a global setting, applied to all administrators across all tenants for the Controller.

Only administrators with write access can modify this parameter by navigating to **Administration > Controller**.

The account can timeout based on the global settings defined by an administrator with appropriate privileges.

You can change this setting through the GUI by logging in to the NSX Advanced Load Balancer UI and navigating to **My Account > Session Timeout**.

You can change this setting through the CLI by navigating to `show controller properties api_idle_timeout`.

User Account Lockout

This feature prevents users from logging in after 20 failed attempts. The user account is locked out for 30 minutes after the last failure login attempt. If the account has not been locked, the running count of failed login attempts is reset to 0 after a valid login.

The administrator controls this feature through the NSX Advanced Load Balancer CLI or REST API. The setting for it is maintained within the *UserAccountProfile* object. By default, all the users in the system are attached to *Default-User-Account-Profile*, as shown in the following example. If required, the admin can create a new user account profile with different thresholds.

Note This feature can be deactivated by setting the *max_login_failure_count* to zero.

```
admin:10-10-24-52]: > show useraccountprofile Default-User-Account-Profile
+-------------------------------+-------------------------------------------------------------+
| Field | Value |
+-------------------------------+-------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+-------------------------------+-------------------------------------------------------------+
```

To change user account lockout attributes, use the following.

```
[admin:10-10-24-52]: > configure useraccountprofile Default-User-Account-Profile
Updating an existing object. Currently, the object is:
[admin:10-10-24-52]: useraccountprofile> max_login_failure_count 30
Overwriting the previously entered value for max_login_failure_count
[admin:10-10-24-52]: useraccountprofile> account_lock_timeout 60
Overwriting the previously entered value for account_lock_timeout
```

```
[admin:10-10-24-52]: useraccountprofile> save
+------------------------------+------------------------------------------------------------+
| Field | Value |
+------------------------------+------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 30 |
| account_lock_timeout | 60 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+------------------------------+------------------------------------------------------------+
```

User Credentials Timeout

The admin can choose to expire user credentials after a configurable number of days. Once credentials have expired, all API calls will error out. Only API or user account is supported at this point, to enable the user to change the password. If the user has configured an email address, **Forgot Password** workflow can also be followed as this point to reset the password.

The administrator controls this feature through the NSX Advanced Load Balancer CLI or REST API. The setting for it is maintained within the *UserAccountProfile* object. By default, all the users in the system are attached to *Default-User-Account-Profile*, as shown in the following example. If required, the admin can create a new user account profile with different thresholds.

```
+------------------------------+------------------------------------------------------------+
| Field | Value |
+------------------------------+------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+------------------------------+------------------------------------------------------------+
CLI commands to set credentials_timeout_threshold:
[admin:10-10-24-52]: > configure useraccountprofile Default-User-Account-Profile
Updating an existing object. Currently, the object is:
[admin:10-10-24-52]: useraccountprofile> credentials_timeout_threshold 60
Overwriting the previously entered value for credentials_timeout_threshold
[admin:10-10-24-52]: useraccountprofile> save
+------------------------------+------------------------------------------------------------+
| Field | Value |
+------------------------------+------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 60 |
+------------------------------+------------------------------------------------------------+
```

Strong Default Admin Password

The initial default password of the admin user of the NSX Advanced Load Balancer Controller is set to a strong password. This password is available in the NSX Advanced Load Balancer portal where release images are uploaded and accessible only to customers having an account on the portal.

SSH access to the Controller with this default password is not allowed until the user changes the default password of the admin user. Once the password is changed, SSH access to the admin user is permitted. The primary reason behind this change is to mitigate the risk of having a newly created Controller being attacked through SSH using the simple password. The password can be changed through the Welcome screen in NSX Advanced Load Balancer UI, NSX Advanced Load Balancer REST API, or through a remote CLI shell. This change strikes a good balance between security and ease of automation and keeps the existing automation workflows such as the NSX Advanced Load Balancer UI welcome screen, adding Controller nodes to a cluster unchanged.

Customers must change the password of the admin user to a strong password first before any other change is implemented in the Controller.

NSX Advanced Load Balancer CLI

To change the password using CLI, you need to launch a remote CLI shell as follows.

```
avi_shell --addr 10.10.25.44
Login: admin
Password:

[admin:10-10-25-44]: > passwd
Changing password for admin
(current) password:
Enter new password:
Retype new password:
Password successfully changed
[admin:10-10-25-44]: >
```

NSX Advanced Load Balancer UI

If you use the NSX Advanced Load Balancer UI, a welcome screen is launched. The user's first step is to change the admin user password.

NSX Advanced Load Balancer REST API

If you plan to use the SDK or API directly, establish a login session with the Controller using the default password and execute the API calls below to change the password.

```
   PUT https://<controller-ip>/api/useraccount     DATA: {"old_password": <default password>,
"password": <new password>}
```

Strong Password Enforcement

The default deployment of NSX Advanced Load Balancer creates an admin account for access to the system. This initial account does not mandate any specific password requirements. Additional user accounts can be created, either local (username/password) or remote accounts (which are tied into an external auth system, such as LDAP). For local accounts, it is possible to enable strong password enforcement. Enabling this option does not impact the passwords of existing accounts. It only impacts newly created accounts, or existing accounts that are attempting to change their password.

The strong password enforcement feature does not affect remotely authenticated accounts. It also does not impact the password requirements for the underlying Linux operating system of the NSX Advanced Load Balancer.

Password Requirements (Strong Password Enforcement Enabled)

Following are the list of password requirements:

- Minimum of 8 characters
- It must contain at least one occurrence of three of the following four categories:
  - Uppercase letters
  - Lowercase letters
  - Digits
  - Special characters
- Dictionary words are not allowed.
- Passwords that are easy to guess. For example, the password **!QAZxsw2** is not allowed because it is entered using 8 adjacent keys on the keyboard.

You can specify the minimum permissible password length when password complexity is enforced. The configurable minimum password length is 8 characters by default, but can range from 6 to 32 characters.

You can configure minimum password length using the following CLI command.

```
[admin]: > configure systemconfiguration
[admin]: systemconfiguration > portal_configuration
[admin]: systemconfiguration:portal_configuration > minimum_password_length <value>
```

Strong password enforcement is enabled by default. You can deactivate it if required.

Note Only an account that has the System Administrator role can change this setting.

Enabling Strong Password Enforcement

Strong password enforcement can be enabled using the CLI commands shown below.

```
bash# shell
: > configure systemconfiguration
: systemconfiguration> portal_configuration
: systemconfiguration:portal_configuration> password_strength_check
Overwriting the previously entered value for password_strength_check
: systemconfiguration:portal_configuration> exit
: systemconfiguration> exit
```

The following is the truncated view of the results.

```
+-------------------------------------+----------------------------------+
| Field                               | Value                            |
+-------------------------------------+----------------------------------+
| uuid                                | default                          |
| portal_configuration                |                                  |
|   enable_https                      | True                             |
|   redirect_to_https                 | True                             |
|   enable_http                       | True                             |
|   enable_clickjacking_protection    | True                             |
|   allow_basic_authentication        | False                            |
|   password_strength_check           | True                             |
+-------------------------------------+----------------------------------+
```

Deactivating Strong Password Enforcement

Warning Deactivating strong password enforcement is not recommended.

```
bash# shell
: > configure systemconfiguration
: systemconfiguration> portal_configuration
: systemconfiguration:portal_configuration> no password_strength_check
Overwriting the previously entered value for password_strength_check
: systemconfiguration:portal_configuration> exit
: systemconfiguration> exit
```

Password History Enforcement

This feature prevents a user from updating their password to some password that was used in the recent past. A configurable number of previous password hashes are saved in the system. Any future proposed password update is compared against this list and marked invalid if there is a perfect match.

The administrator controls this feature through NSX Advanced Load Balancer CLI or REST API. The settings for this feature are maintained within the `UserAccountProfile` object. By default, all the users in the system are attached to *Default-User-Account-Profile*, as shown below. If required, the admin can create a new user account profile with different thresholds.

```
admin:10-10-24-52]: > show useraccountprofile Default-User-Account-Profile
+-------------------------------+--------------------------------------------------------------+
| Field | Value |
+-------------------------------+--------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 0 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+-------------------------------+--------------------------------------------------------------+
```

Use the following CLI to change the password history count.

```
[admin:10-10-24-52]: > configure useraccountprofile Default-User-Account-Profile
[admin:10-10-24-52]: useraccountprofile> max_password_history_count 5
Overwriting the previously entered value for max_password_history_count
[admin:10-10-24-52]: useraccountprofile> save
+-------------------------------+--------------------------------------------------------------+
| Field | Value |
+-------------------------------+--------------------------------------------------------------+
| uuid | useraccountprofile-6753548e-7ac5-4601-939b-ad4394405db4 |
| name | Default-User-Account-Profile |
| max_password_history_count | 5 |
| max_login_failure_count | 20 |
| account_lock_timeout | 30 |
| max_concurrent_sessions | 0 |
| credentials_timeout_threshold | 0 |
+-------------------------------+--------------------------------------------------------------+
```

Password-less SSH-based Login for Admin User

The NSX Advanced Load Balancer enables the admin user to log in using SSH keys, instead of having to supply a password. This section details ways for password-less SSH-based login for admin user.

Password-less Login using NSX Advanced Load Balancer CLI

- Use the following code in CLI to upload a key.

  ```
  upload adminkey public_key "<public key>
  ```

- To delete a specific key, use the following command.

  ```
  delete adminkey public_key "<public key>"
  ```

- Use the following command to delete all the keys.

```
delete adminkey
```

Password-less Login using NSX Advanced Load Balancer Rest API

- USe the following API request to upload a key.

```
POST https://<controller-ip>/api/adminkey JSON data: {"key":"<public key>"}
```

- To delete a specific key, use the following API request.

```
DELETE https://<controller-ip>/api/adminkey?key=<public key>
```

- To delete all the keys, use the following API call.

```
DELETE https://<controller-ip>/api/adminkey?key
```

Password Recovery

This section discusses ways to recover passwords for different user types.

Reset a Known Password

Locally authenticated users can change their own password by logging in, clicking their user name in the top-right corner of the GUI, and then clicking the **My Account** selection from the menu. When the user account editor appears, the user enters the old password followed by the new.

Note An administrator can change passwords of other users through the User account page, but can change their own password only through their account page.

Local User Password Recovery

If a locally-authenticated user forgets their password, the following two options are available.

- User-initiated

If SMTP email has been configured for NSX Advanced Load Balancer, locally authenticated users can make a reset-password request through the GUI. In the NSX Advanced Load Balancer UI, the **Forgot Your Password?** link is available below the **Password** field. Clicking the link brings up a window which prompts for the email addressof the user. If the provided email address has been configured through the **Administration > Account > User page**, an email containing a link to reset the password is sent to the email id. If SMTP has not been configured, the **Forgot your password?** link does not appear.

Note The user account must have a valid email configured for the account, and NSX Advanced Load Balancer must have access to an SMTP server.

- Admin-initiated

Any administrator or user who has write privileges to user objects can change the password for a local user. From the admin account, navigate to **Administration > Account > User** and edit the account to be reset. Input a new password or select the **Generate** button to create a random password for the user.

Note Though the password is reset on saving, the administrator must still copy and manually send this new password to the user.

Remote User Password Recovery

Password recovery for users who are remotely authenticated through LDAP or TACACS must be performed with the remote auth server. With remote authentication, the NSX Advanced Load Balancer does not maintain user passwords and does not participate in the password recovery process.

Admin User Password Recovery

The procedure to reset the administrator password is the same as that of any locally-authenticated user.

User Profile

The User Profiles section is used to control a user access to NSX Advanced Load Balancer. You can configure and control different attributes related to the user account. By default, there are two user profiles available:

1 Default-User-Account-Profile

2 No-Lockout-User-Account-Profile

Manage User Profiles

The User Profiles screen lists all the user accounts created.

Procedure

1 Navigate to **Administration > Accounts > User Profiles**.

2 From this screen, **Create** new user profiles, edit or **Delete** existing user profiles, as required.

Create a User Profile

The steps to create a new user profile are as follows.

Procedure

1 From the **User Profiles** screen, click **Create**.

You can view the **New User Profile** screen.

2 Enter a **Name**.

3 Under **Max Password History Count**, enter the maximum number of passwords to be maintained in the password history.

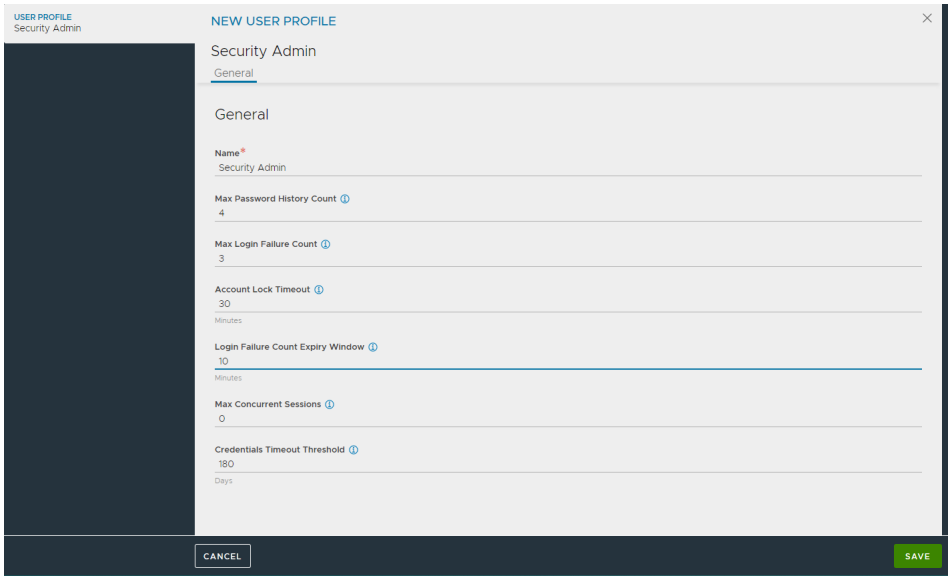
4 Enter the **Account Lock Timeout** period in minutes.

5 Under **Login Failure Count Expiry** Window, enter the time (in minutes) within which only `login_failure_attempts` from the past will be considered for account lockout. Set this to `0` if you do not want to do a time-based account lockout and consider all failed login attempts irrespective of time frame.

6 Enter the maximum number of concurrent sessions allowed.

7 Enter **Credentials Timeout Threshold** (in days), after which the credentials are invalid.

The **New User Profile** screen is as shown below:



8 Click **Save**.

What to do next

To completely avoid the risk of your account getting locked, you can use the Configure No-Lockout-User-Account-Profile on NSX Advanced Load Balancer. By default, this user profile has Max login Failure Count and Login Failure Count Expiry Window set to 0.

NSX Advanced Load Balancer Controller Operations

8

This section discusses the recommended steps and precautions required during the maintenance operations of an NSX Advanced Load Balancer Controller.

Shutting Down or Rebooting an NSX Advanced Load Balancer Controller

The Controller Virtual Machines (VMs) can be shut down or rebooted directly through the hypervisor or a public cloud console. If the Controller is part of a cluster, it will attempt to rejoin the cluster once it is operational. The Controller does not require user intervention for a reboot operation in a cluster set-up.

Increasing Controller Disk Size of an NSX Advanced Load Balancer Controller

There are scenarios where the Controller's disk size configured initially is insufficient and requires an increase in the disk size. The Controller VM's disk can be resized directly using the hypervisor or the public cloud console. If the Controller is part of a cluster, it is recommended to perform the changes on the follower nodes, followed by the leader node.

Note

- Increasing overall disk capacity by adding a new, additional disk to the Controller VM is not supported.
- It is recommended to configure the same disk size for all Controllers in a cluster.

Follow the below steps to increase Disk Size:

1. Shut down the Controller VM.
2. Increase the disk size through the hypervisor or public cloud console.
3. Start the Controller VM.
4. SSH to the Controller VM, and confirm that the new disk size using the command `sudo df -h`.
5. Perform the above steps on the follower nodes first, followed by the leader nodes.

Increasing Memory and vCPU of an NSX Advanced Load Balancer Controller

There are scenarios where the vCPU or memory configured initially is insufficient and requires an increase in the allotment. Another related use case it so select a different flavor or size for the Controller VM for a public cloud or OpenStack deployment. The resources can be edited using the hypervisor, or the public cloud console.

If the Controller is part of a cluster, it is recommended to perform the changes on the follower nodes, followed by the leader node.

Note

- It is recommended to configure the same resource parameters for all Controllers in a cluster.

Follow the below steps to increase vCPU or Memory of an NSX Advanced Load Balancer Controller:

1 Shut down the Controller VM.

2 Increase the vCPU or memory assigned to the VM.

3 In vCenter environments, increase the allocation using the settings.

4 In case of flavor-based selections, select the new flavor or VM size.

5 Start the Controller VM.

6 Perform the above steps on the follower nodes, followed by the leader nodes.

Increasing Controller Disk Capacity

Following is the process by which disk capacity for Controllers in a cluster can be increased.

1 Power down one of the Controller followers.

2 Increase disk size for the physical or virtual machine using the tools appropriate to your environment.

3 Power up the Controller on that machine.

4 Verify that it has joined into the cluster.

5 Repeat steps 1-4 for the second follower.

6 Power down the Controller cluster leader. One of the two already-upgraded followers automatically become the new leader.

7 Proceed with steps 2-4 for the third and final Controller. The third controller must have joined as the second follower.

Read the following topics next:

- Backup and Restore of NSX Advanced Load Balancer Configuration

- Upgrade and Patches
- Increasing Docker Container Size

Backup and Restore of NSX Advanced Load Balancer Configuration

Periodic backup of the NSX Advanced Load Balancer configuration database is recommended. This database defines all clouds, all virtual services, all users, and so on.

Any user capable of logging into the admin tenant is authorized to perform a backup of the entire configuration, that is, of all tenants. A restore operation spans all the same entities but can only be performed by the administrator(s) capable of logging into one of the Controllers using SSH or SCP.

It is a best practice to store backups in a safe external location in the unlikely event that a disaster destroys the entire NSX Advanced Load Balancer Controller (or cluster), with no possibility of remediation. A recommended backup schedule could be daily or even hourly, based on how often the configuration changes.

Backing Up the NSX Advanced Load Balancer Configuration

To back up the configuration, use the NSX Advanced Load Balancer UI, CLI commands, or API commands shown in this section. Backups can be scheduled or on-demand.

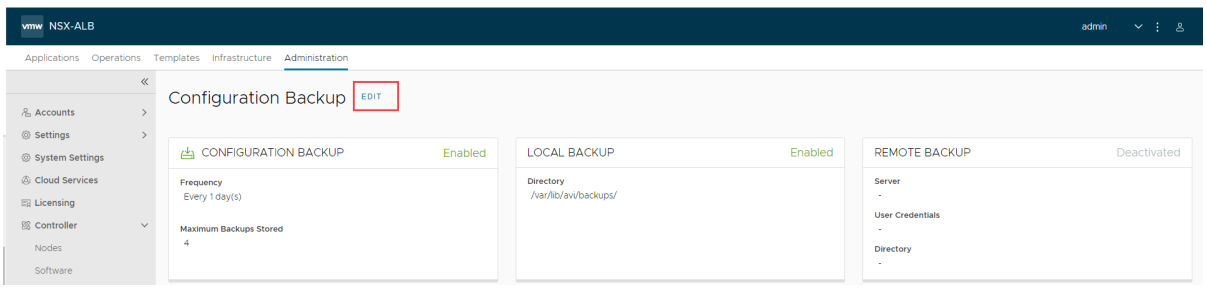
Note When backing up Avi Load Balance via the CLI method for system restore, it is required and recommended that you perform the backup with the *full_system* option enabled.

Scheduled Backup using the UI

To view or edit the configuration backup scheduler's current settings, an admin-tenant user first must navigate to **Administration > Controller > Configuration Backup**.

Note The scheduled backups get stored in `/var/lib/avi/backups/` on all NSX Advanced Load Balancer Controller in the cluster.

1 To configure the backup, click **EDIT** to view the **EDIT CONFIGURATION BACKUP** screen.



2 Select **Enable Configuration Backup** check box to schedule backups.

3 Enter the **Passphrase** and confirm the passphrase to encrypt all sensitive fields contained within the backup. Choose a phrase that is not easy to guess and guard it carefully. Data cannot be restored without the passphrase.

4 Enter a **File Prefix** to customize the backup archive filename.

5 Select a remote transfer Protocol from the drop-down menu.

   a Secure Copy Protocol (SCP)

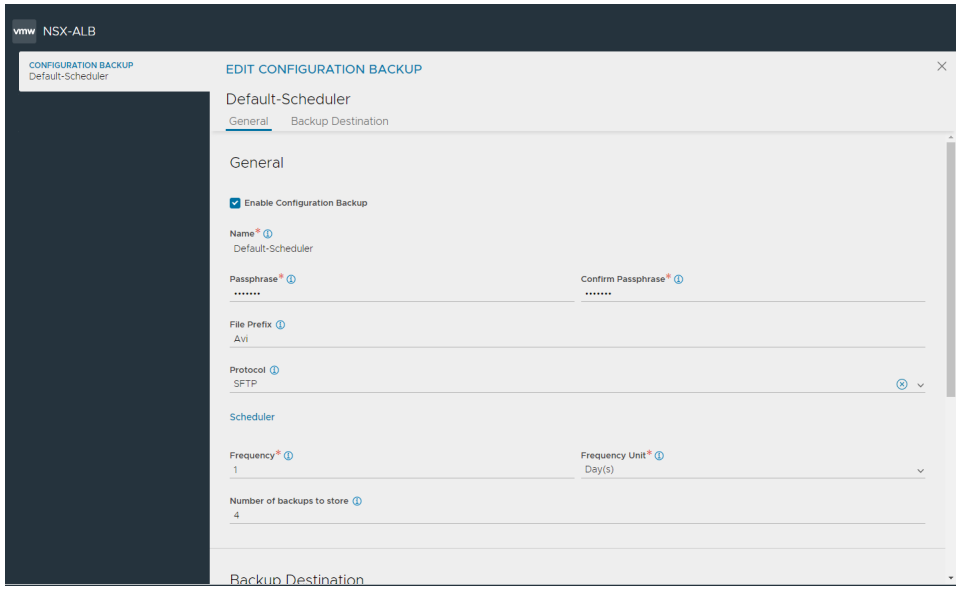
   b Secure File Transfer Protocol (SFTP)

6 Under **Scheduler**, configure the following:

   a Enter a value from 0 to 60 as the **Frequency** to determine how often backups are to be taken. 0 indicates the backup sequence has no end time.

   b Enter the **Frequency Unit** for backups to occur. By default, the **Frequency Unit** is taken every day. Use this field to change the unit to minutes, hours, weeks, or months.

   c Enter a number ranging from 0 to 20 as the **Number of backups to store**. 0 is equivalent to deselect the Local option. The oldest backup is deleted after the most recent backup successfully completes.



7 You can choose to save local (on Controller) and remote backup locations independently. In the **Backup Destination** section, configure the following:

   a Select **Enable Local Backups (On Controller)**, to preserve the number of indicated backups on the Controller.

   b Select **Enable Remote Server Backup** to save the backup on a remote server.

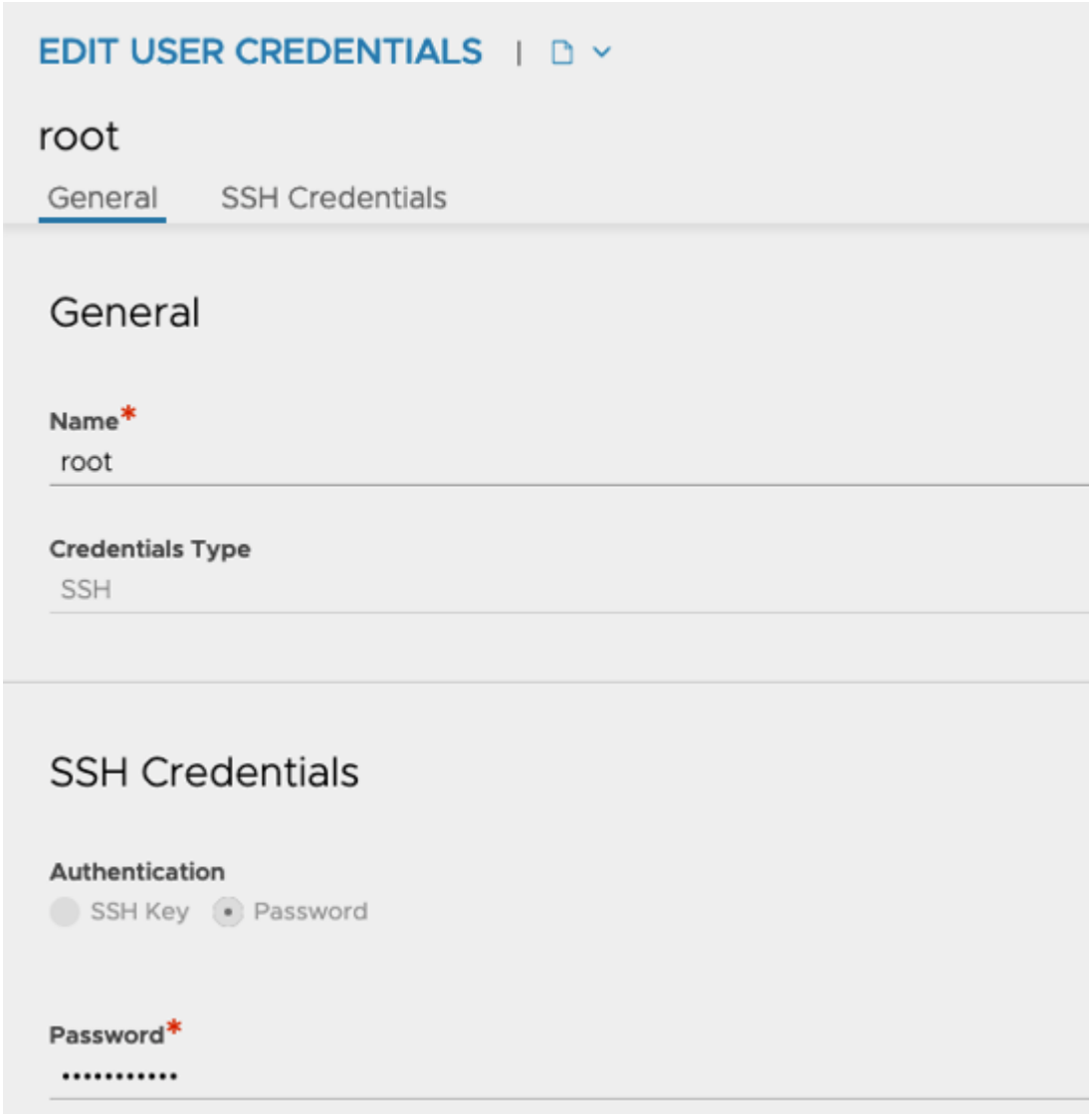
   Note This option is deactivated by default, but it is recommended to specify a remote destination in case the Controller cluster fails in a non-recoverable fashion.

   c In the **Server Address** field, enter an FQDN or IP address reachable from the Controller.

d In the **Home Directory** field, provide a remote destination address with write permissions.

e Use the drop-down menu to select **User Credentials** from a previously-defined SSH user. Alternatively, click the three dots at the end of the field to **Create** new user credentials.

f If you are creating a new SSH user, the following dialog is shown.



Enter a username in the **Name** field. NSX Advanced Load Balancer will use this username to transfer the file. In the **Authentication** field, select either **SSH Key** or **Password** and provide the required information.

8 Click **Save** to complete configuration backup settings.

Scheduled Backup using the CLI

Note When backing up Avi Load Balance via the CLI method for system restore, it is required and recommended that you perform the backup with the *full_system* option enabled.

```
[admin:10-102-64-234]: > show backupconfiguration Backup-Configuration
+-------------------------------+-------------------------------------------------------------+
| Field                         | Value                                                       |
+-------------------------------+-------------------------------------------------------------+
| uuid                          | backupconfiguration-0c5b26a0-a45d-47af-b230-42a8901aaf57 |
| name                          | Backup-Configuration                                        |
| save_local                    | True                                                        |
| maximum_backups_stored        | 4                                                           |
| upload_to_remote_host         | True                                                        |
| ssh_user_ref                  | aviuser                                                     |
| remote_directory              | /tmp                                                        |
| remote_hostname               | 10.102.64.235                                               |
| backup_passphrase             | <sensitive>                                                 |
| backup_file_prefix            | sftp                                                        |
| remote_file_transfer_protocol | SFTP                                                        |
| tenant_ref                    | admin                                                       |
+-------------------------------+-------------------------------------------------------------+
[admin:10-102-64-234]: >
```

You can specify the value of `start_date_time` from the CLI (not possible using the UI).

```
[admin:10-10-24-52]: > configure scheduler Default-Scheduler
[admin:10-10-24-52]: scheduler> no enabled
[admin:10-10-24-52]: scheduler> start_date_time 2017-05-11T00:00:00
Overwriting the previously entered value for start_date_time
[admin:10-10-24-52]: scheduler> save
[admin:10-10-24-52]: > configure scheduler Default-Scheduler
[admin:10-10-24-52]: scheduler> enabled
Overwriting the previously entered value for enabled
[admin:10-10-24-52]: scheduler> save
```

Scheduled Backup using the API

In this example, a `PUT` changes the scheduler frequency to one week.

```
PUT : api/scheduler/
{'_last_modified': u'1476209663670990',
'backup_config_ref': 'https://10.10.24.52/api/backupconfiguration/
backupconfiguration-5d65f12e-5da1-49e0-b703-ec65ae9a39c6',
 'enabled': True,
 'frequency': 1,
 'frequency_unit': u'SCHEDULER_FREQUENCY_UNIT_WEEK',
 'name': u'Default-Scheduler',
 'run_mode': u'RUN_MODE_PERIODIC',
 'scheduler_action': u'SCHEDULER_ACTION_BACKUP',
 'start_date_time': u'2016-10-09T15:35:46.220623',
 'tenant_ref': u'https://10.10.24.52/api/tenant/admin',
```

```
'url': 'https://10.10.24.52/api/scheduler/scheduler-b5f7e673-8818-44d1-8f74-45238cc08235',
'uuid': u'scheduler-b5f7e673-8818-44d1-8f74-45238cc08235'}
```

On-demand Backup using the CLI

To back up the NSX Advanced Load Balancer configuration on-demand, at any arbitrary time, use the following CLI command.

```
: > export configuration file /tmp/avi_config.json full_system
Please enter the passphrase to encrypt configuration:
Retype passphrase:
Downloaded the attachment to /tmp/avi_config.json
Completed writing the export configuration to /tmp/avi_config.json
```

On-demand Backup using the REST API

To back up the NSX Advanced Load Balancer configuration on-demand, at any arbitrary time, use the following API request.

```
GET https://[CONTROLLER-IP]/api/configuration/export?full_system=true
```

To include a passphrase, use one of the following options.

```
GET https://[CONTROLLER-IP]/api/configuration/export?full_system=true&passphrase=[PASSPHRASE]
```

Use the following POST method and include passphrase in the JSON data.

```
POST https://[CONTROLLER-IP]/api/configuration/export?full_system=true JSON data:
{"passphrase":"[PASSPHRASE]"}
```

Make sure to replace [CONTROLLER-IP] with the IP address of the NSX Advanced Load Balancer Controller (if using a single Controller node) or the IP address of the Controller cluster.

On-demand Backup Script Utilizing Python

For more information, see vmware/alb-sdk.

Configuring Backup using Amazon S3

A backup configuration can be stored on Amazon S3 bucket.

To enable backup configuration on Amazon S3, use the `configure backupconfiguration` command and set the value of the `upload_to_s3` flag to `true`.

Provide the value of the following attributes to save the backup file on the Amazon S3 bucket for the required instance.

- aws_access_key : Access Key ID
- aws_secret_access : Secret Access Key

- aws_bucket_id : Name of the S3 bucket

```
[admin:10-1-1-1]: configure backupconfiguration Backup-Configuration
```

Note For enabling backup of the NSX Advanced Load Balancer Controller, you must have write access permission to the S3 bucket.

For detailed information on the Access Key ID and Secret Access Key, see *AWS User Cross-Account AssumeRole* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.

Restore the VMware NSX Advanced Load Balancer Configuration

In case, an unlikely disaster destroys the NSX Advanced Load Balancer Controller (or entire cluster), the device or VM hosting the Controllers must first be restored to factory default using `flushdb.sh`. Failure to do so can prevent the Controller from coming up. If there is a prev partition, rename or delete the prev partition. The `prev` partition can either be `root1` or `root2mv root1/root2 prev_back`.

Steps to check the Partition Mapping are listed below:

1 `sudo cat/proc/cmdline`

You can observe an output with either root1 or root2 as a current partition.

For example, we see root1 below as current partition.

```
root=UUID=f4a947e1-7efb-4345-9eac-1ff680fc50e0 subroot=/root1 net.ifnames=0 biosdevname=0
console=tty0 console=ttyS0,115200n8
```

2 Go to `/host` directory and rename the `prev` partition as shown below.

```
cd /host
ls -lrth <- This is to see if you have root1 and root2 directories.
mv root2 prev_bak ----> as root2 is prev partition
```

Thereafter, the following script can be used to automate the configuration recovery process.

`/opt/avi/scripts/restore_config.py`

Note Prev partition must be removed.

This script imports the backup configuration onto the NSX Advanced Load Balancer Controller. If restoring a Controller cluster, this script restores the configuration and re-adds the other two nodes to the cluster.

1 Create three new Controllers with the same IP address as the original cluster members. (NSX Advanced Load Balancer currently supports only static IP addresses). At this point, other than having an IP address, each Controller node must be in its factory default state.

2 Log in to one of the NSX Advanced Load Balancer Controller nodes using SSH or SCP. Use the default administrator credentials.

3 Run the restore command or script:

- Copy backup file using SCP.

  ```
  scp /var/backup/avi_config.json admin@<controller-ip>://tmp/
  avi_config.json
  ```

4 Run the following restore command locally using SSH.

```
/opt/avi/scripts/restore_config.py --config CONFIG --passphrase PASSPHRASE --
do_not_form_cluster DO_NOT_FORM_CLUSTER --flushdb --vip VIP --followers FOLLOWER_IP
[FOLLOWER_IP ...]
```

In the above command line:

- `CONFIG` is the path of the configuration file.
- `PASSPHRASE` is the export configuration passphrase.
- `DO_NOT_FORM_CLUSTER` causes cluster formation to be skipped.
- `VIP` is the virtual `IP` address of the Controller.
- `FOLLOWER_IP [FOLLOWER_IP ...]` is a list of the IP addresses of the followers.
- `CLUSTER_UUID` is the old cluster `UUID` to be restored.

Note Starting with NSX Advanced Load Balancer 22.1.3, the following options are no longer supported.

1 --do_not_form_cluster

2 --vip

3 --followers

Follow the steps below to restore configuration in a cluster set-up.

1 Restore the configuration on one of the nodes

2 Reform the cluster by inviting the two new nodes to the cluster.

Restore Configuration for a Three-Node Cluster

For a three-node cluster, before running the script ensure the following conditions are met:

- The follower nodes need to be in a factory default state
- The two new Controllers need to be spawned (make sure not to make any changes to them).

Execute the following command to reset the nodes to factory default states.

```
sudo systemctl stop process-supervisor.service
rm /var/lib/avi/etc/flushdb.done
/opt/avi/scripts/flushdb.sh
sudo systemctl start process-supervisor.service
```

Optional arguments for the `restore_config` script:

- vip `<Virtual IP of controller>` (Virtual IP of the Controller in case of a cluster)
- followers `<follower_IP_1> <follower_IP_2>` (IP addresses of the follower nodes)
- `cluster_uuid <cluster_uuid>` (Old Cluster UUID will be restored)

Run the below command with the above optional arguments to restore a 3-node cluster.

```
/opt/avi/scripts/restore_config.py --config  --flushdb --passphrase  --followers
```

If the followers argument is not added in the above command, the cluster can be formed manually from the UI, CLI, or API after a successful restore configuration. The other Controllers need to be in a factory default state to be accepted for cluster formation.

During a restore config, the SE package is re-signed, and existing SE packages are deleted.

If a restore config fails due to a configuration import error, the logs can be found in `/opt/avi/log/portal-webapp.log`.

Upgrade and Patches

This section discusses about the upgrade and patches.

Upgrade Overview

NSX Advanced Load Balancer supports improved and more flexible methods for upgrading the NSX Advanced Load Balancer system.

The followings are the additional features for the Flexible Upgrades:

- The upgrade is possible per SE group. The transition of all the SE groups to the new version might occur over a long period.
- Upgrades of different SE groups are supported with different patch versions.
- Rollback to the previous versions of NSX Advanced Load Balancer is non-disruptive.

The following options are available with Flexible Upgrades.

Upgrades	Patch Ugrades	Rollback	Rollback Patch
System (NSX Advanced Load Balancer Controller and SE Groups)	System (NSX Advanced Load Balancer Controller and SE Groups)	System (NSX Advanced Load Balancer Controller and SE Groups)	System (NSX Advanced Load Balancer Controller and SE Groups)
NSX Advanced Load Balancer Controller only	NSX Advanced Load Balancer Controller only	NSX Advanced Load Balancer Controller only	NSX Advanced Load Balancer Controller only
Some or all the SE groups	Some or all the SE groups	Some or all the SE groups	Some or all the SE groups

Use Cases

- Scenarios where it is not possible to upgrade all SE groups to the newer version at the same time due to various business reasons such as logistics, confidence in the new software, and so on
- The configuration is blocked during the entire duration of the NSX Advanced Load Balancer Controller and SE upgrade. This is not acceptable in many deployments. With the new upgrade feature, the process is flexible and can be performed on an SE group basis. The configuration is blocked for the entire duration if a system upgrade is performed till all SEs are upgraded
- Using SE groups for data plane separation. Based on the SE group segmentation, the upgrade is performed depending on the following attributes:
  - Application or product offering
  - Tenant
  - Production, pre-production, and development environments
  - Cloud or environment (AWS, VMware, and so on.)
  - Provide patches to only applications or SE groups that need them
- Flexible scheduling
- Self-service upgrades

Patch Overview

NSX Advanced Load Balancer supports patch upgrades by which hotfixes are placed into effect.

The followings are the available options for patch upgrade:

- System — Patch upgrade for NSX Advanced Load Balancer Controller and all SE groups
- Controller — Patch upgrade for the NSX Advanced Load Balancer Controller
- SE group — Patch upgrade for some or all the SE groups

Note The following are a few points for a patch upgrade process:

- An image with a patch can be applied.
- The image and the patch must have the same base version.
- A patch cannot be applied without applying the image.
- Compatibility checks prevent incorrect patches from getting applied to different versions.

Prerequisites

This section discusses about the prerequisites of Image Management and Service.

Image Management and Service

Image service is the first step in the flexible upgrade workflow. It uploads the image after an upgrade operation is initiated. The NSX Advanced Load Balancer Controller hosts images of different versions because SE groups could be potentially in different versions.

The Controller must have additional disk space to host these images.

NSX Advanced Load Balancer Controller images for the major versions include the following:

- `controller.pkg` (for VM-based NSX Advanced Load Balancer Controller)
- `controller_docker.tgz` (For Docker-based Controller)

Images for the patches include the following:

- `avi_patch.pkg` (Full package)
- `controller_patch.pkg` (NSX Advanced Load Balancer Controller package)
- `se_patch.pkg` (SE patch package)

As a part of the upload process, the image service extracts files, and metadata from the package. This information is not only displayed to the user, but it is also used in the upgrade process.

Note

- Image service provides an ability to upload, query, and delete NSX Advanced Load Balancer image(s) to the system.
- Image service supports the upload of NSX Advanced Load Balancer patch packages.
- Image upload can happen only on the cluster leader. It is not allowed from a cluster member.

Image Bundling

NSX Advanced Load Balancer supports composite image or image bundle. The composite image consists of the following:

- Base image – Controller image (`controller_docker.tgz`, `controller.pkg`, `controller ova`, `controller.qcow2`, and so on).
- Controller package – It is an optional package.
- SE patch image – It is an optional package.

The upgrade workflow using the image bundle, or the composite image is the same as the standard image. When using the image bundle for an upgrade, a patch image can be applied in addition to the base image.

Note When upgrading from versions 18.x or lesser to 21.1 and higher, in the NSX Advanced Load Balancer Controller, change the `DefaultTimeoutStartSec` (File: `/etc/systemd/system.conf`) to 120 seconds to avoid timeout during the upgrade.

Uploading Image Using the CLI

The CLI provides better control of the upgrade operations leading to a consistent and predictable workflow.

For uploading the package, use the `upload image filename <path-of-the-package>` command as shown below.

```
[admin:controller]: > upload image filename /tmp/controller.pkg
```

The following show command returns the details of the image metadata. Use the `show image <image-name>` command as shown below.

```
[admin:-controller]: > show image
+------------------------------+------------------------------------------+----------------+
| Name                         | UUID                                     | Status         |
+------------------------------+------------------------------------------+----------------+
| 18.2.7-5000-20191009.205501 | image-fxxxx22-0f40-45de-8551-15xxxxxxx1fe | SYSERR_SUCCESS |
+------------------------------+-----
```

For more information about differences in CLI commands and APIs, see Comparison Table for Differences in CLIs Commands and APIs.

Uploading Image Service using the REST API

A POST operation is used for image upload. To get the image details in response, run a GET API request.

- Use the following REST API to upload the image for `controller.pkg`.
  - **URI**: `/api/image`
  - **Method**: `POST`

```
root@admin:-controller# curl -X POST -k  https://10.58.3.27/api/image  -u "admin:admin"
-F file=@controller.pkg -H "X-CSRFToken: P9e5H5aCtRwtAVIouzJW9eoxWNPSHptc" -H "Cookie:
csrftoken= P9e5H5aCtRwtAVIouzJW9eoxWNPSHptc; sessionid= 9ku3n1z3oqa62k7lwgbdm8zcatf75qag"
-H "Referer: https://10.58.3.27/api"
```

- Use the following REST API to upload the image for `controller_patch.pkg`.

```
root@admin:-controller# curl -X POST -k  https://10.58.3.27/api/image
-u "admin:admin"   -F file=@controller_patch.pkg -H "X-CSRFToken:
P9e5H5aCtRwtAVIouzJW9eoxWNPSHptc" -H "Cookie: csrftoken= P9e5H5aCtRwtAVIouzJW9eoxWNPSHptc;
sessionid= 9ku3n1z3oqa62k7lwgbdm8zcatf75qag" -H "Referer: https://10.58.3.27/api"
```

- Use the following API to delete the image provided, if it is not in use.

  `delete image <image-name>`

Must-Checks for Upgrade

Prior to upgrade operations, various must-checks are run to check the mandatory and optional requirements for upgrade. The output message is displayed either as an error message or a warning message. Warnings can be skipped, whereas Errors cannot be over-ridden. CLI/API provides the `skip_warnings` option to control the above behavior.

- For the CLI: This is directly integrated into the normal workflow, and there is no separate command.
- For the REST API: Add `/preview/` at the end of APIs to get previews for that particular flow.

Starting with NSX Advanced Load Balancer 22.1.3, initiating an upgrade or patch through the CLI will halt at the `UPGRADE_PRE_CHECK_WARNING` state if the command is executed without the `skip_warnings` parameter.

Review the warnings using the command `show upgrade status detail filter pre_check_status` and take any necessary actions.

After addressing the warnings, you can run the upgrade or patch command again with the `skip_warnings` parameter to allow the operation to continue.

```
[admin:10-10-10-1]: > upgrade system image_ref 30.3.3-7235-20230110.035149 skip_warnings
+-------------+-------------------------------------------------------------------------------+
| Field       | Value                                                                         |
+-------------+-------------------------------------------------------------------------------+
| status_code | SYSERR_UPGRADE_OPS_PREVIEW_RESPONSE                                           |
| status      | Checks preview for upgrade operations.                                        |
| checks      |                                                                               |
|             | Check Controller Cluster readiness for upgrade operations.                    |
|             | Check and inform user to take a backup prior to upgrade operations.           |
|             | Check if se linux is enabled on controller nodes.                             |
|             | Check if upgrade operation is already in progress.                            |
|             | Check ServiceEngineGroup has an ongoing upgrade operation.                    |
|             | Check image version compatibility for upgrade operations.                     |
|             | Check ServiceEngine reachability for upgrade operations.                      |
|             | Check ServiceEngine disk space for upgrade operations.                        |
|             | Check Controller Cluster disk space for upgrade operations.                   |
|             | Check and inform Virtual Service(s) disruption for upgrade operations.        |
|             | Check idempotent operations for upgrade operations.                           |
|             | Check active versions compatibility for upgrade operations.                   |
|             | Check ServiceEngineGroup error recovery options prior to upgrade operations.  |
|             | Check Image state across Cluster members for upgrade operations.              |
|             | Checks for the patch in image bundle.                                         |
|             | Checks if Gslb Feature is enabled and provides feature specific messages.     |
|             | Checks the system configuration.                                              |
|             | Check total number of alerts for upgrade operations.                          |
|             | Checks if the cloud api versions are compatible after upgrade.                |
|             | Checks if Docker version is compatible.                                       |
|             | Checks if configured IP type is DHCP or STATIC.                               |
|             | Checks if se has a valid license state.                                       |
+-------------+-------------------------------------------------------------------------------+
Starting upgrade
+-------------
```

```
+-------------------------------------------------------------------------------------------------
--------------+
| Field       |
Value
            |
+-------------
+-------------------------------------------------------------------------------------------------
--------------+
| status_code |
SYSERR_UPGRADE_SYSTEM_STARTED
            |
| status      | 'Upgrade of System (Controller + All SEGroup(s)) started. Use 'show upgrade
status' to check the status.' |
+-------------
+-------------------------------------------------------------------------------------------------
--------------+
```

Similarly, use the `skip_warning` option while performing the patch upgrade.

```
[admin:10-10-10-1]: > patch controller controller_patch_ref 23.1.1-7189-2p1-20221216.192828
skip_warnings
Previewing upgrade
```

Determine Software Version

This topic details the methods to determine the specific NSX Advanced Load Balancer software version running on the Controller and its SEs.

Finding Version through CLI

From the Controller shell, enter the following command.

```
: > show version controller
+-----------------+--------------------------------------+
| Controller Name | Version                              |
+-----------------+--------------------------------------+
| 10.130.162.14   | 15.2.4(9007) 2015-10-17 01:43:38 UTC |
+-----------------+--------------------------------------+
```

Alternatively, from bash, run the following command.

```
cat /bootstrap/VERSION
```

Finding Version through Web Interface

From the top right corner of the Controller web interface, click the user icon and select **About NSX ALB (Avi)** from the drop-down menu.

Pre/Post Upgrade Object Operational State Verification

This section compares the pre/post-upgrade operational state using the CLI.

For software upgrade operations, the NSX Advanced Load Balancer Controller provides visibility into the pre-operation and post-operation states of the system, summarizing and highlighting the significant differences in the state or health of the system caused by the performed operation.

When performing a software upgrade, the NSX Advanced Load Balancer Controller will take an operational state snapshot pre-upgrade and another post-upgrade to allow the admin to perform a quick validation of the current state of their environment compared to the state before the upgrade.

The operational snapshots are taken for the following objects:

- Virtual Services
- Service Engines
- Pools
- GSLB Services

The operational snapshots are taken during the following software operations:

- Upgrade
- Patch
- Rollback Patch

Compare Pre/Post Upgrade Operational State using the CLI

The following are the steps to compare pre/ post upgrade operational state using the CLI:

1 Check if all the state-diff operations exists in the system.

```
 > show statediff
 +-----------------------------------------------------+-----------------+
 | Name                                                | Status          |
 +-----------------------------------------------------+-----------------+
 | upgrade_controller_fitb_2021-06-22-08:20:40.957288 | FB_IN_COMPLETED|
 +-----------------------------------------------------+-----------------+
```

2 Check if the summarized state changes for a particular operation from the above listed operations.

```
> show statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288
 +----------------+----------------+
 | Object         | States Changed |
 +----------------+----------------+
 | SERVICEENGINE  | 0              |
 | VIRTUALSERVICE | 1              | // Indicates 1 VS changed state for the queried
operation
 | GSLBSERVICE    | 0              |
 | POOL           | 0              |
 +----------------+----------------+
```

3 Check for the detailed state changes for all entities.

```
> show statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288 filter detail
 +---------------+-----------------------------------------------------
+-------------------------------------------------
+-------------------------------------------------+
 | Name          | Uuid
| PreSnapshot                                     |
PostSnapshot                                    |
 +---------------+-----------------------------------------------------
+-------------------------------------------------
+-------------------------------------------------+
 | 10.79.168.138 | se-005056812adc
|
|                                                 |
 |               |                                                     |     state:
OPER_UP,                            |     state: OPER_UP,                             |
 |               |                                                     |     reason:
[                                   |     reason: [                                   |
 |               |                                                     |
Service Engine connected to controller |         Service Engine connected to controller |
 |               |
|     ],
|     ],                                          |
 |               |                                                     |     reason_code:
0,                              |     reason_code: 0,                               |
 |               |
|     last_changed_time: 21-06-22-08:16:02
|     last_changed_time: 21-06-22-08:16:02        |
 |               |
|
|                                                 |
 | vs-2          | virtualservice-7e22eec4-
fc6b-40d9-9306-84259f54948a |
|                                                 |
 |               |                                                     |     state:
OPER_UP,                            |     state: OPER_UP,                             |
 |               |
|     last_changed_time: 21-06-12-17:45:27
|     last_changed_time: 21-06-12-17:45:27        |
 |               |
|
|                                                 |
 | vs-1          | virtualservice-1aa470fa-
e43f-47f3-b720-0836bf87bad8 |
|                                                 |
 |               |                                                     |     state:
OPER_UP,                            |     state: OPER_DOWN,                           |
 |               |
|     last_changed_time: 21-06-12-17:45:27
|     last_changed_time: 21-06-12-17:45:27        |
 |               |
|
|                                                 |
 | pool-1        | pool-fb65d17e-
```

```
c28a-413c-a44e-17bdb705fd1b           |
|                                                |
 |               |                                                          |     state:
OPER_UP,                            |     state: OPER_UP,                            |
 |               |
|     last_changed_time: 21-06-12-17:45:27
|     last_changed_time: 21-06-12-17:45:27       |
 |               |
|
|                                                |
 | gslb-svc1     | gslbservice-
b3fed408-fa96-40a0-b240-df8e8c524dfb    |
|                                                |
 |               |                                                          |     state:
OPER_UP,                            |     state: OPER_UP,                            |
 |               |
|     last_changed_time: 21-06-12-17:55:33
|     last_changed_time: 21-06-12-17:55:33       |
 |               |
|
|                                                |
 +---------------+------------------------------------------------------
+------------------------------------------------
+------------------------------------------------+
```

4 Check for the summary of state changes for all the objects of a particular type (virtual service in this example).

```
 > show statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288 filter obj_type
fb_virtualservice
 +------+------------------------------------------------------+--------------
+----------------------+
 | Name | Uuid                                                 | State Change |
Summary              |
 +------+------------------------------------------------------+--------------
+----------------------+
 | vs-2 | virtualservice-7e22eec4-fc6b-40d9-9306-84259f54948a | NO           |
-                    |
 | vs-1 | virtualservice-1aa470fa-e43f-47f3-b720-0836bf87bad8 | YES          | OPER_UP ->
OPER_DOWN |
 +------+------------------------------------------------------+--------------
+----------------------+
```

5 Check for the detailed state changes for desired objects.

```
> show statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288 filter virtualservice
vs-1,vs-2 pool pool-1
 +--------+-----------------------------------------------------
+-------------------------------------------+-----------------------------------------+
 | Name   | Uuid                                                |
PreSnapshot                               | PostSnapshot                              |
 +--------+-----------------------------------------------------
+-------------------------------------------+-----------------------------------------+
 | vs-2   | virtualservice-7e22eec4-fc6b-40d9-9306-84259f54948a
```

```
|                                          |                                                |
 |        |                                                    |     state:
OPER_UP,                      |     state: OPER_UP,                      |
 |        |                                                    |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                |
 | vs-1   | virtualservice-1aa470fa-e43f-47f3-b720-0836bf87bad8
|                                          |                                                |
 |        |                                                    |     state:
OPER_UP,                      |     state: OPER_DOWN,                    |
 |        |                                                    |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                |
 | pool-1 | pool-fb65d17e-c28a-413c-a44e-17bdb705fd1b
|                                          |                                                |
 |        |                                                    |     state:
OPER_UP,                      |     state: OPER_UP,                      |
 |        |                                                    |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                |
 +--------+------------------------------------------------------
+------------------------------------------+------------------------------------------+
```

6 Check for the state changes for all objects under a particular `serviceenginegroup(parent obj)` object.

```
> show statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288 filter virtualservice
vs-1,vs-2 pool pool-1
 +--------+------------------------------------------------------
+------------------------------------------+------------------------------------------+
 | Name   | Uuid                                               |
PreSnapshot                              | PostSnapshot                             |
 +--------+------------------------------------------------------
+------------------------------------------+------------------------------------------+
 | vs-2   | virtualservice-7e22eec4-fc6b-40d9-9306-84259f54948a
|                                          |                                                |
 |        |                                                    |     state:
OPER_UP,                      |     state: OPER_UP,                      |
 |        |                                                    |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                |
 | vs-1   | virtualservice-1aa470fa-e43f-47f3-b720-0836bf87bad8
|                                          |                                                |
 |        |                                                    |     state:
OPER_UP,                      |     state: OPER_DOWN,                    |
 |        |                                                    |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                |
 | pool-1 | pool-fb65d17e-c28a-413c-a44e-17bdb705fd1b
```

```
|                                          |                                                  |
 |        |                                                  |     state:
OPER_UP,                       |     state: OPER_UP,                        |
 |        |                                                  |     last_changed_time:
21-06-12-17:45:27 |     last_changed_time: 21-06-12-17:45:27 |
 |        |
|                                          |                                                  |
 +--------+--------------------------------------------------
+-----------------------------------------+-------------------------------------------+
```

7 Clean up the changes, if required, using the following command.

```
> delete statediff upgrade_controller_fitb_2021-06-22-08:20:40.957288
```

Pre-upgrade Checklist

This section provides a list of checks to be performed before initiating upgrade of NSX Advanced Load Balancer.

NSX Advanced Load Balancer supports a simple system upgrade method wherein all NSX Advanced Load Balancer Controller nodes and SE nodes are upgraded to the newer software version in one upgrade sequence. Following are the list of checks to be performed before upgrading the NSX Advanced Load Balancer:

- Check if all the three nodes of the Controller cluster are Active.

  For more information about Controller clusters, see Chapter 5 NSX Advanced Load Balancer Controller Cluster and Changing NSX Advanced Load Balancer Controller Cluster Configuration.

- Check if there is adequate disk space on all the Controller nodes before the upgrade. Ideally, the disk usage must not be more than 60%. For more information, see *NSX Advanced Load Balancer Controller Sizing* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.

- If the disk usage is more (greater than 80%), look for any core archives or packages uploaded to the Controller nodes and clear them. Core archives are under `/var/lib/avi/` archives.

- Ensure all servers are connected. Run `Show serviceengine`. Check if all the service engines are `UP` unless they are disabled. For more information, see *De-activating SE* topic in the VMware NSX Advanced Load Balancer *Monitoring and Operability Guide*. The `upgrade` command does a pre-check to ensure that no Service Engines are disconnected from the Controller, so we do not inadvertently finish the upgrade without these Service Engines.

- Ensure that all the SEs are on the version required. Run `show version controller` and `show version serviceengine`.

- Check CPU and memory of all Controllers to ensure compliance with minimum recommendations.

Note For NSX Advanced Load Balancer (formerly Avi Vantage) lifecycle information, see the Product Life Cycle Matrix and the Lifecycle Policy.

Post-upgrade Checklist

This section provides a list of checks to be performed after successful upgrade of NSX Advanced Load Balancer.

NSX Advanced Load Balancer supports a simple system upgrade method wherein all NSX Advanced Load Balancer Controller nodes and SE nodes are upgraded to the newer software version in one upgrade sequence. The list of checks performed post-upgrade are:

- Execute `show upgrade status` to ensure that the upgrade is successful. The status should be `UPGRADE_COMPLETED`.
- Execute `show version controller` and `show version serviceengine` to ensure that the service engines are on the version expected.
- Ensure all the three nodes of the Controller cluster are active.
- Ensure that the Cloud Status is **Placement Ready**.
- Inspect the **Application Dashboard** to ensure the application's health is as expected. If you have traffic tools to test the availability of the virtual services, ensure that this is running successfully. For more information, see *Virtual Service Health Monitoring* topic in the VMware NSX Advanced Load Balancer*Monitoring and Operability Guide*.

Upgrade Guide

This section explains the steps to upgrade NSX Advanced Load Balancer System (NSX Advanced Load Balancer Controller and SE Groups)

Prechecks_only Flag

Starting with NSX Advanced Load Balancer Controller 22.1.6, a new flag `prechecks_only` is introduced for all upgrade operations (upgrade, patch upgrade, rollback). The `prechecks_only` flag enables upgrade APIs to perform upgrade checks prior to an upgrade maintenance window. Executing the `prechecks_only` flag does not trigger any upgrade operation and it is safe to perform upgrade operations with the `prechecks_only` flag outside of a maintenance window.

Use the `upgrade controller image_ref <image_name> prechecks_only` command to inititate the prechecks_only operation for NSX Advanced Load Balancer Controller.

```
[admin:100-65-10-137]: > upgrade controller image_ref 22.1.6-8143-20240115.080016
prechecks_only
Previewing upgrade
+-------------
+-----------------------------------------------------------------------------------------+
```

```
| Field       |
Value                                                                                                |
+-------------
+-----------------------------------------------------------------------------------------------+
| status_code |
SYSERR_UPGRADE_OPS_PREVIEW_RESPONSE                                                                  |
| status      | Checks preview for upgrade
operations.                                                 |
| checks
|                                                                                                       |
|             | 'Checking Controller Cluster readiness for upgrade
operations.'                          |
|             | 'Check if upgrade operation is already in
progress.'                                    |
|             | 'Checking active versions compatibility for upgrade
operations.'                         |
|             | 'Checking if system in Avi-ESSENTIALS/BASIC mode and allows only permitted
operations.' |
|             | 'Checking if the cloud api versions are compatible after
upgrade.'                      |
|             | 'Checking ServiceEngineGroup has an ongoing upgrade
operation.'                           |
|             | 'Checking Controller Cluster disk space for upgrade
operations.'                          |
|             | 'Checking image version compatibility for upgrade
operations.'                           |
|             | 'Checking idempotent operations for upgrade
operations.'                                  |
|             | 'Checking Image state across Cluster members for upgrade
operations.'                     |
|             | 'Checking the system
configuration.'                                                    |
|             | 'Checking config
migration.'                                                                 |
|             | 'Checking if Gslb Maintenance Mode is enabled prior to
upgrade.'                          |
|             | 'Checking if Gslb Feature is enabled and provides feature specific
messages.'           |
|             | 'Checking and inform user to take a backup prior to upgrade
operations.'                |
|             | 'Checking if Docker version is
compatible.'                                             |
|             | 'Checking for the mandatory patch in image
bundle.'                                      |
|             | 'Checking if configured IP type is DHCP or
STATIC.'                                      |
|             | 'Checking if se linux is enabled on controller
nodes.'                                    |
|             | 'Checking total number of alerts for upgrade
operations.'                                  |
+-------------
+-----------------------------------------------------------------------------------------------+
Starting upgrade
+--------------------
+-----------------------------------------------------------------------------------------------
```

```
---+
| Field              |
Value
 |
+--------------------
+-----------------------------------------------------------------------------------------------
---+
| status_code        |
SYSERR_EVAL_UPGRADE_CONTROLLER_STARTED
 |
| status             | 'Pre-checks for Upgrade of Controller started. Use 'show upgrade
status' to check the status.' |
| system_report_uuid | systemreport-6f499bff-fcbd-4fca-
a43a-483d1764eed5                                          |
+--------------------
+-----------------------------------------------------------------------------------------------
---+
```

Upgrading NSX Advanced Load Balancer System (NSX Advanced Load Balancer Controller and SE Groups)

The configuration and placement of virtual services are blocked if it is a system-level upgrade till all the SEs are upgraded. Once these operations are completed, configuration on NSX Advanced Load Balancer Controller (except the configuration of virtual service and VIP) is allowed, irrespective of the SE group upgrade status.

Note

- It is recommended to increase the default timeout value from 90 seconds to 120 seconds before performing the upgrade. The default timeout is available as the parameter name *DefaultTimeoutStartSec* and at (File: `/etc/systemd/system.conf`).This is to avoid the upgrade going to timeout.

Number of Maximum Active Versions Consideration for Upgrade

NSX Advanced Load Balancer supports a maximum of 2 active image versions in the system at any time.

It is essential to consider various deployment scenarios when planning system upgrades. This strategic approach aids in the effective execution of system upgrades.

Consider the following deployment scenarios:

- Case 1: The Controller's image version is 22.1.6, and one of SE Group's image versions is 22.1.5. So, the number of active versions used in the system is 2 (22.1.5 and 22.1.6).
  - In this case, the number of active versions of deployed images is 2. As the number of active versions is equal to the maximum active version supported, upgrading Controller to the next version is not supported. An upgrade to the next version, say 30.2.1 in this case, means total number of active versions at a time are 30.2.1, 22.1.5, and 22.1.6. It is not allowed as it exceeds maximum number active versions supported. Make sure to upgrade all the SE Groups' image versions to 22.1.6 before attempting the upgrade operation to 30.2.1 version.
- Case 2: When all SE Groups' image versions are 22.1.6, and the Controller's image version is also 22.1.6, the number of active versions is 1 (22.1.6).
  - In this scenario, upgrading the Controller to the next version is supported, as the number of active versions (1) is less than the maximum number of active versions supported (2).

Using the CLI

The auto-suggest option in the CLI provides available values on pressing tab on your keyboard. Use `skip_warnings` option to skip any warnings and optional must checks.

The following are the various options available for NSX Advanced Load Balancer system upgrade:

- Use the `upgrade system image_ref <image name>` command to upgrade the system to a base image.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9000-20191031.063017
```

- Use the following to upgrade the system to a base image and a controller patch.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535
controller_patch_ref 18.2.6-9134-2p1-20190806.011824
```

- Use the following to upgrade the system to a base image and an SE patch.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535 se_patch_ref
18.2.6-9134-2p1-20190806.011824
```

- Use the following to upgrade the system to a base image, an NSX Advanced Load Balancer Controller patch, and an SE patch.

```
[admin:-controller]: >upgrade system image_ref 18.2.6-9134-20191101.042535
controller_patch_ref 18.2.6-9134-2p1-20190806.011824 se_patch_ref
18.2.6-9134-2p1-20190806.011824
```

SE Upgrades

The Controller allows you to pick up the number of SE-groups per Controller node.

`seupgrade_fabric_pool_size`:

This property allows the Controller to pickup number of SE groups per Controller to upgrade.

For instance, if `seupgrade_fabric_pool_size` is set to 3, three SE-groups are picked up per Controller, that means 9 SE groups across the cluster.

The default value of seupgrade_fabric_pool_size is 20. However, you can update this based on the requirement or the load.

`seupgrade_copy_pool_size`:

This parameter defines the number of simultaneous SE image downloads in a SEGroup. It is used to pace the SE downloads so that Controller network/ CPU bandwidth is a bounded operation. A value of zero will disable the pacing scheme and all the SE(s) in the SEGroup will attempt to download the image.

`seupgrade_copy_pool_size` = n, where 'n' is the number of SE within SE group will be picked for copy.

For instance, if `seupgrade_copy_pool_size` = 3, the three SE in a picked up SE group will be picked for copy.

The default value of seupgrade_copy_pool_size is 5. However, you can update this based on the requirement or the load.

The following are the steps to configure this:

1 Configure the Controller properties.

2 Set `seupgrade_fabric_pool_size <number>`.

3 Set `seupgrade_copy_pool_size <number>`.

4 Save.

```
[admin:ctrl]: > configure controller properties
[admin:ctrl]: controllerproperties> seupgrade_fabric_pool_size 2
Overwriting the previously entered value for seupgrade_fabric_pool_size
[admin:ctrl]: controllerproperties> seupgrade_copy_pool_size 2
Overwriting the previously entered value for seupgrade_copy_pool_size
[admin:ctrl]: controllerproperties> save

[admin:ctrl]: > show controller properties |grep pool
| seupgrade_fabric_pool_size                | 2                  |
| seupgrade_copy_pool_size                  | 2                  |
[admin:ctrl]: >
```

Using the REST API

Image UUID is obtained by using the `GET /api/image`.

The following are the various REST API options available for the NSX Advanced Load Balancer system upgrade:

- Use the following API to upgrade the system to a base image.

  API:`/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
  {
    'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
    'system': true
  }
  ```

- Use the following API to upgrade the system to a base image and a controller patch.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
  {
     'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
     'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
     'system': true
   }
  ```

- Use the following API to upgrade the system to a base image and an SE patch.

  API:`/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
  {
     'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
     'system': true,
     'se_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
     'skip_warnings': True
   }
  ```

- Use the following API to upgrade the system to a base image, an NSX Advanced Load Balancer Controller patch, and an SE patch

  API: `/api/upgrade`

  Method: `POST`

  `JSON` Data:

  ```
  {
   'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  ```

```
 'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
 'system': true,
 'se_patch_uuid': 'image-e88aaad68-5aaf-485a-8bd9-1db3ec562d6a'
}
```

Upgrading NSX Advanced Load Balancer Controller

This section explains about upgrading NSX Advanced Load Balancer Controller.

Using the UI

The NSX Advanced Load Balancer supports improved and more flexible methods for upgrading the system. The Controller supports all the upgrade workflows through the UI.

Following are the additional features for Flexible Upgrades using the NSX Advanced Load Balancer UI:

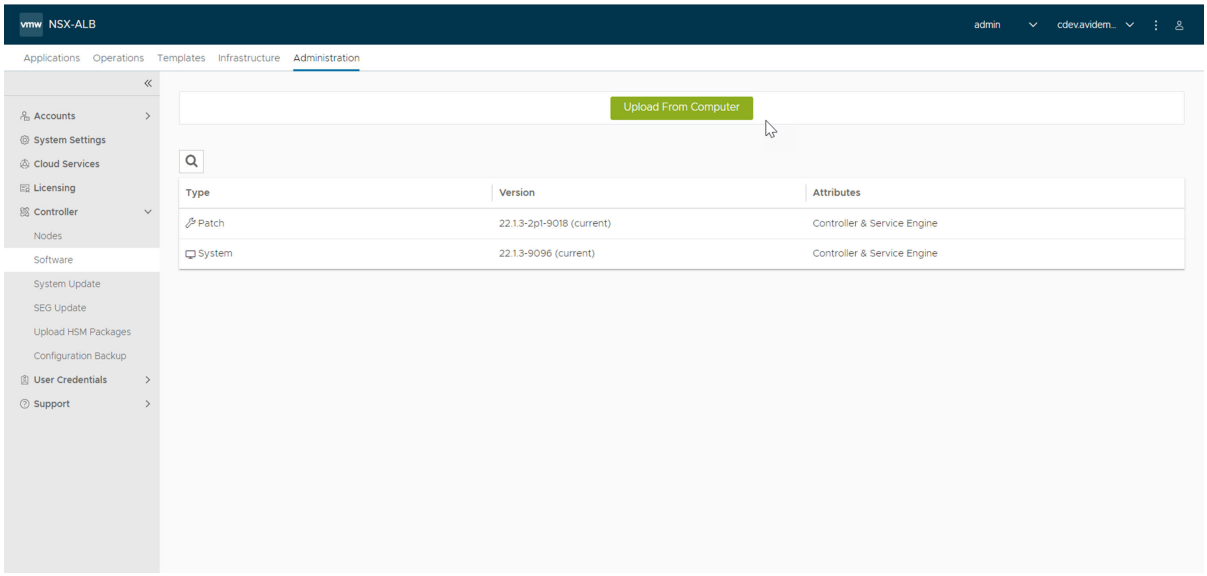
- Upgrading System
  - Upgrading only the Controller
  - Upgrading the Controller and the Service Engine groups
- Upgrading only Service Engine Group – You can upgrade a single SE group or multiple SE groups, as needed. The transition of all the SE groups to the new version can occur over a long period. Upgrades of different SE groups are supported with different patch versions.

The following sections explain the different options available for the NSX Advanced Load Balancer software's upgrade process through the UI.

Upload the Software

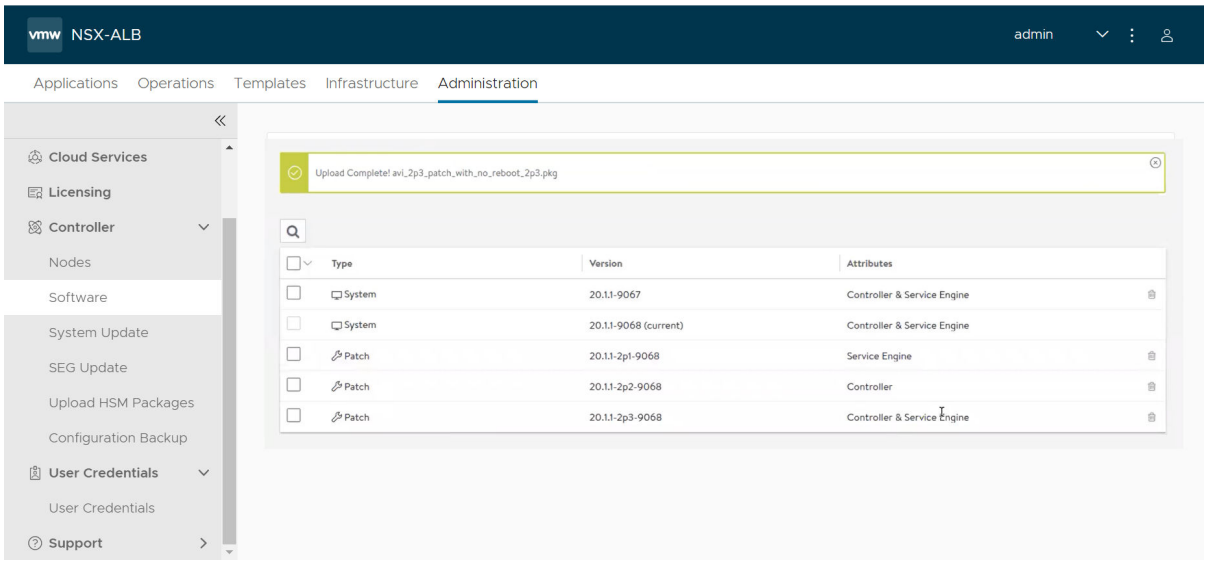
To upload the NSX Advanced Load Balancer software:

1 Login to the NSX Advanced Load Balancer UI, and navigate to **Administration > Controller > Software**. Click **Upload From Computer** to upload the NSX Advanced Load Balancer software to the Controller.



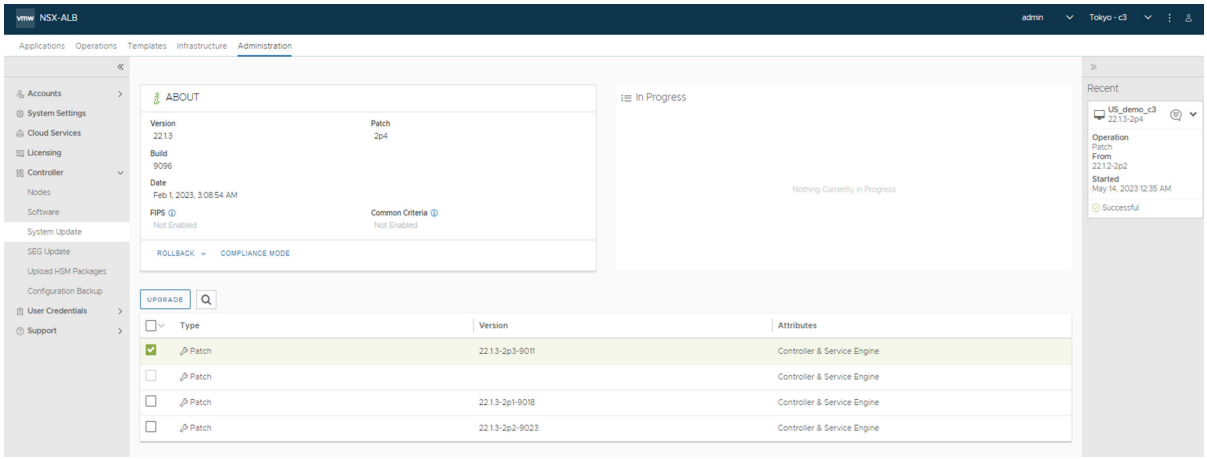
2 Once the file is selected, the upload of the software image starts. The status of the image upload progress is displayed on the UI.

3 After the image upload is complete, the software package is available under the **Software** tab.

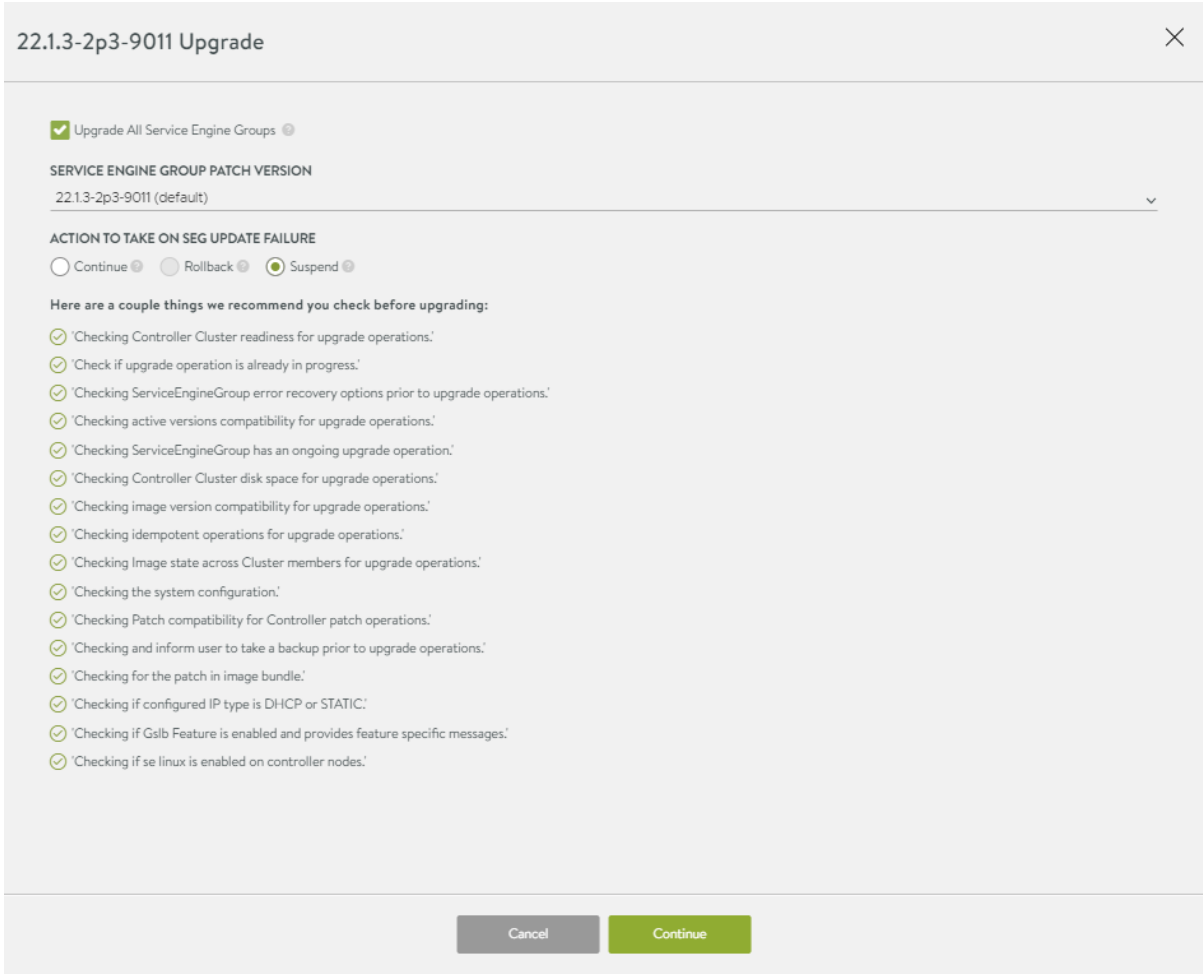


Upgrade the NSX Advanced Load Balancer System

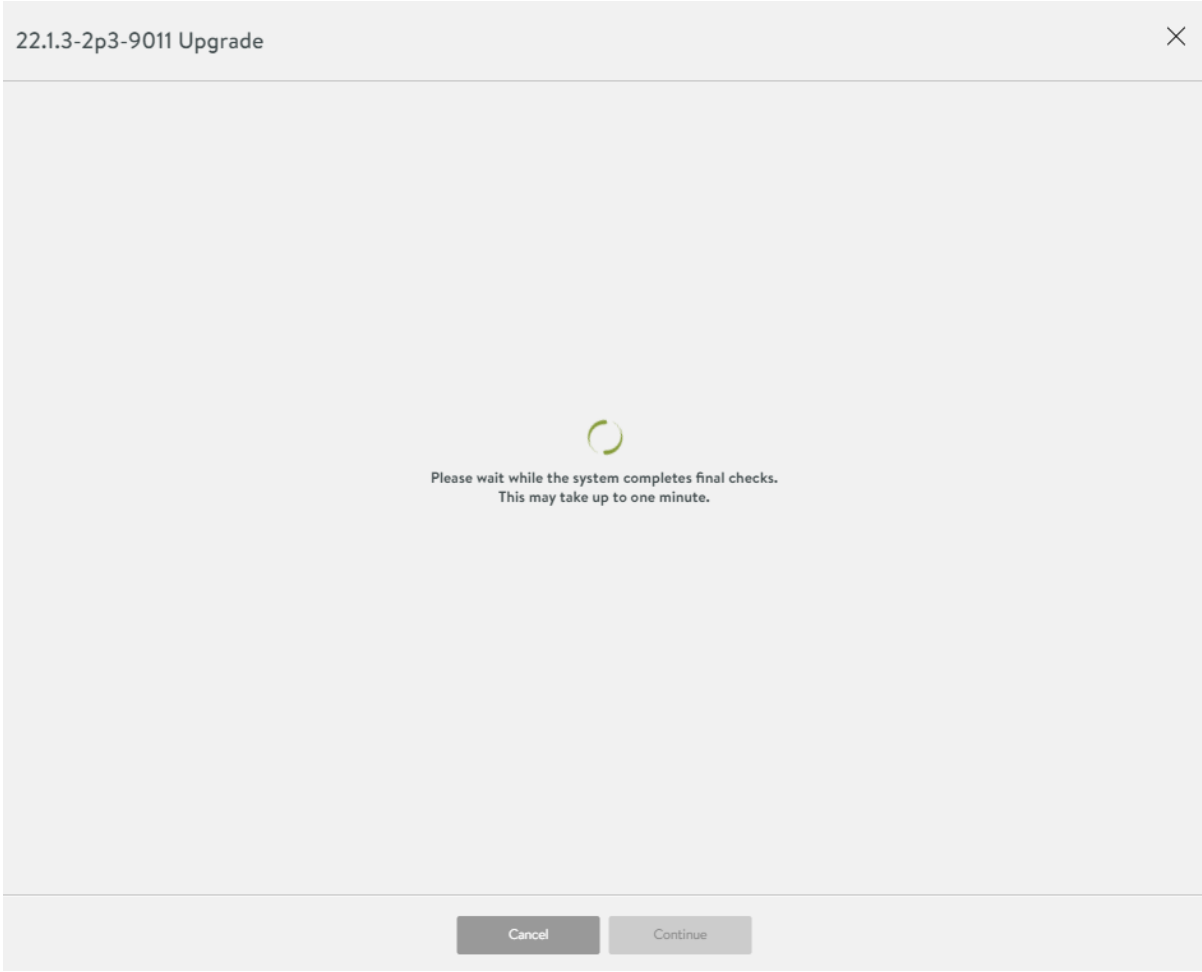
1 Navigate to **Administration > Controller > System Update**, select the image uploaded in the previous step and click **UPGRADE** to start the upgrade process.



2 Select the **Upgrade All Service Engine Groups** check-box for selecting the SE groups update along with the Controller upgrade. Click **Continue** to proceed with the Controller and SE groups software upgrades.



The following screen appears for the final checks before the upgrade proceeds.



3 Do not select the check-box for the SE group update if the requirement is to upgrade the Controller only.

4 When selecting the SE group update, the following two options are available if the SE groups updates fail:

- **Continue** – The upgrade process for SE groups continues in the event of an error, failure, or any other issues.
- **Suspend** – The upgrade process will not progress in the event of an error, failure, or other such issues.

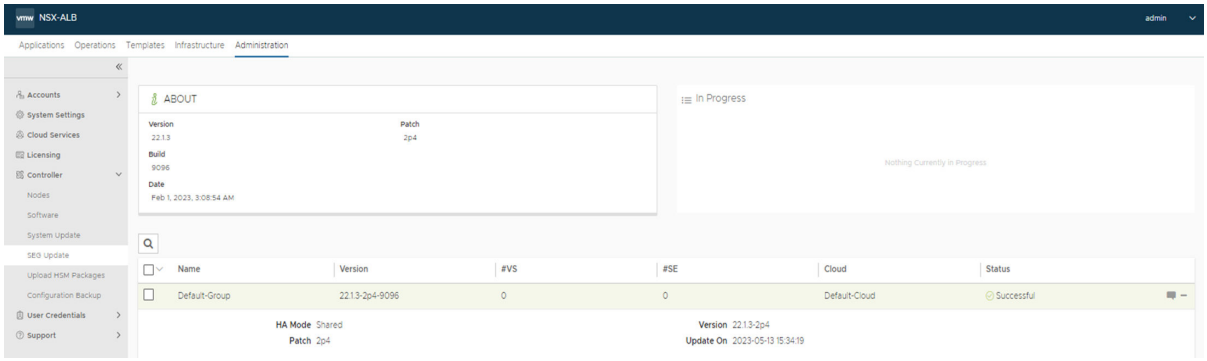
5 **Confirm** that the backup of configuration has been performed, when prompted for.

6 The progress for the update process is available on the UI under the **In Progress** section.

7 Once the upgrade process is complete, the latest software version will be available under the **Administration > Controller > System Update** tab. The current tag is displayed next to the updated software version.

8 Once the Controller upgrade is successful, the Service Engine group upgrade is initiated, as the SE group upgrade was chosen in step no. 2. The following screenshot exhibits the message regarding the SE group update status.

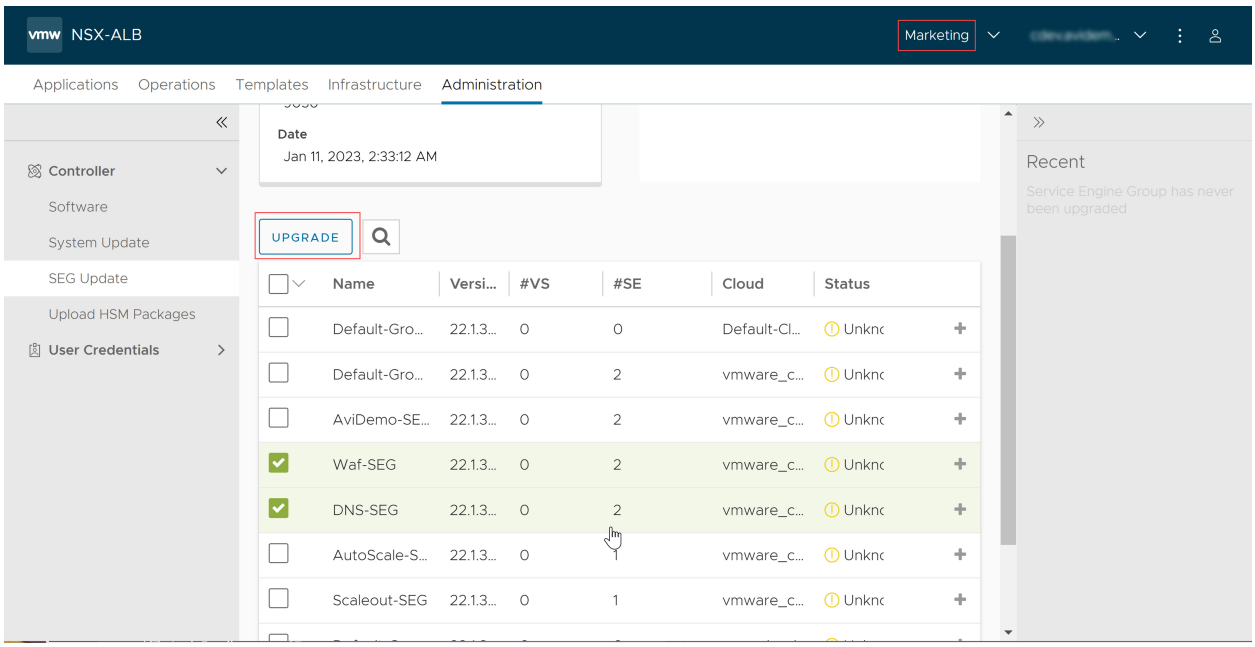
9 Once the SE group update is successful, the upgrade status gets changed to successful, as shown below.



SE Group Update

Use the same steps as mentioned in the previous section to update one or more than one SE group. Navigate to **Administration > Controller > SEG Update**, select the required SE group, and click on **UPGRADE** to proceed with the update process.

Starting with NSX Advanced Load Balancer 22.1.3, SE group upgrade can be initiated in both admin and non-admin tenants as required.



Using the CLI

Log in to the NSX Advanced Load Balancer shell prompt and use the following upgrade commands for various options.

- Use the `upgrade controller image_ref <image name>` command to upgrade the Controller to a base image.

```
[admin:-controller]: >upgrade controller image_ref 18.2.6-9000-20191031.063017
```

- Use the `upgrade controller image_ref <image name>controller_patch_ref <patch name>` command to upgrade the Controller to a base image and an NSX Advanced Load Balancer Controller patch.

```
[admin:-controller]: >upgrade controller image_ref 18.2.6-9134-20191101.042535
controller_patch_ref 18.2.6-9134-2p1-20190806.011824
```

Using the REST API

This section discusses about Using NSX Advanced Load Balancer REST API.

- Use the following API to upgrade the Controller to a base image.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

```
{
'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4'
}
```

- Use the following API to upgrade the Controller to a base image and the NSX Advanced Load Balancer Controller patch.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

```
{
  'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
  'controller_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a'
}
```

Tracking NSX Advanced Load Balancer Controller and Service Engine Upgrade Status in AWS Cloud

The upgrade version and the last upgrade date information are available for the Controller and SEs for deployment in the AWS cloud.

Tags To Check Upgrade Status

The following tags are available for the Controller and SEs to track the upgrade status:

- AVI_VERSION - the latest NSX Advanced Load Balancer version for the Controller and Service Engine
- AVI_UPGRADED_ON - the date and the timestamp when the software version upgrade was completed.

Tagging NSX Advanced Load Balancer Controller

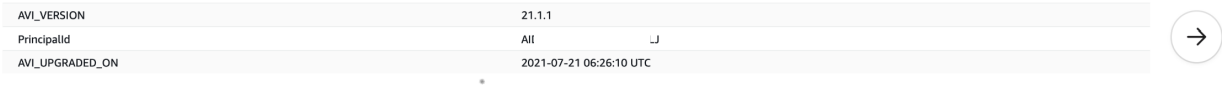
These tags are available only for Controllers, which are deployed with IAM roles attached. Upgrade tags are not supported for the deployment of the Controller using the access key and secret key. The upgrade version and the last upgrade date information are available for Controller and SEs for deployment in the AWS cloud.

Tagging NSX Advanced Load Balancer SEs

SEs have these tags populated irrespective of the access methods used for AWS cloud creation.

Checking Upgrade Status

The tags are available in the AWS Console for the Controller and SE instances.



The above screenshot depicts NSX Advanced Load Balancer at version 21.1.1, and the upgrade was successfully completed on 2021-07-21 at 06:26:10 UTC.

Note On initial deployment, AVI_UPGRADED_ON shows the build date for the Controller and SE software.

Flexible Upgrades Must-Checks

This section explains the must-checks that are executed prior to the upgrade operations. Must-checks are specific to upgrade operations. It returns the messages as warnings or errors. Warnings can be skipped while errors cannot be overridden. NSX Advanced Load Balancer REST API/NSX Advanced Load Balancer CLI presents the `skip_warnings` option to control the above behavior.

- For REST API: add /preview/ at the end of APIs described in V2 APIs to get previews for that particular flow
- For NSX Advanced Load Balancer CLI: This is directly integrated into the normal work-flow and there is no separate command.

Must-Checks

This section describes various must-checks. The following table describes the detailed information on various error and warning messages. Use the steps mentioned in the recovery steps to proceed further with upgrade process.

Must-Check Name and Type	Must-Check Message	Description	Recovery Steps	Operation
CheckLicense(Error)	License <name> is either expired or will expire within <num_days> days	This check ensures that there is a minimum of 7 days of license validity	Please reach out the customer success team to extend the license duration	All upgrade operations
CheckClusterState(Error)	Controller cluster is not ready. All the cluster nodes should be ACTIVE before initiating the process.	The cluster should be ready prior to initiate any upgrade operations.	1 Use the `show cluster` command to ensure that all the cluster members are up and in a consistent state. 2 Wait till the cluster is ready to attempt the upgrade operation.	All upgrade operations
ControllerUpgradeDiskSpace(Error)	Node <node> does not have enough disk space for upgrade operations, The available space is <avail_space>, Required space is <req_space>.	There is insufficient disk space on one or more of the cluster members. 1 Use the from imageproperties to determine if the existing free-disk size meets the requirements of the new image. 2 The from-image properties can be retrieved by `show image <from-image>` command	1 This requires the user to actively intervene and initiate cleanup. 2 Check for cores and initiate a cleanup. 3 Check for tech-support tarballs and initiate a cleanup. 4 Check for unused-images through show image and initiate a cleanup. 5 Check if images are present in non-standard directories (`/home/admin`, etc.) and clean up. 6 Check for `/tmp/` and initiate a cleanup. 7 Perform a `du -sh` from the root and cleanup the outliers.	All upgrade operations

Must-Check Name and Type	Must-Check Message	Description	Recovery Steps	Operation
CheckControllerDiskSpace(Erorr)	Node <node name> does not have enough disk space for upgrade operations, The available space is <avail_space>, Required space is <req_space>	There is insufficient disk space on one or more of the cluster members.	Please refer to the recovery steps mentioned for the ControllerUpgradeDiskSpace error.	Upgrade, patch upgrade, SE group, and SE group resume
CheckControllerRollbackDiskSpace	Node <node name> does not have enough disk space for upgrade operations, The available space is <avail_space>, Required space is <req_space>	There is insufficient disk space on one or more of the cluster members during rollback operations. The free-disk threshold for rollback operations is generally less than the free-disk threshold for upgrade operations.	Please refer to the recovery steps mentioned for the ControllerUpgradeDiskSpace error.	Rollback and rollback patch
SeGroupUpgradeDiskSpace(Error)	Node <node name> does not have enough disk space for upgrade operations, The available space is <avail_space>, Required space is <req_space>	There is insufficient disk space on one or more ServiceEngines. 1 Use thefrom imageproperties to determine the if the existing free-disk size meets the requirements of the new image. 2 The from-image properties can be retrieved by the `show image <from-image>` command.	1. Check for cores and initiate a cleanup. 2. Check for `/tmp` and initiate a cleanup. 3. Use the `du -sh` command from the root directory and clean up the outliers.	(Upgrade, patch) se group and (upgrade, patch) system

Additional Options for Flexible Upgrades

This section explains the following additional options available for Flexible Upgrades:

- Rollback - Error Recovery
- Stop Cleanup
- SE Group Resume Option

Rollback – Error Recovery

- If the upgrade process hits an error, then an automated rollback error recovery is initiated to bring the NSX Advanced Load Balancer Controller(s) into a known good state.
- If the SE group encounters an error, SE group will suspend upgrade if the `suspend_on_failure` flag is used. Otherwise, it will continue to upgrade the rest of the Service Engines in the SE group.
- If the error happens in the context of a patch, then the patch will be rollback.
- If an error is encountered during the execution of rollback mechanism, then it will be treated as upgrade cancelled.

Stop Cleanup

Whenever the rollback operation is triggered and it fails, then the NSX Advanced Load Balancer Controller or SE group will be moved to the abort state. In Flexible Upgrades, these states can be cleaned up and NSX Advanced Load Balancer Controller and SE group are transitioned to a known stable state.

Using SE Group Resume Option

Note SE group resume option is supported only from NSX Advanced Load Balancer release 18.2.8.

Whenever an SE group is upgraded with `suspend_on_failure` enabled and an issue is encountered, the upgrade process for that SE group is suspended. After the issue is resolved through manual intervention, use the following options to resume the upgrade:

- `Se-group-uuids` — Specify the SE group that needs to be resumed.
- `Ignore_failure` — This field overrides the earlier suspend on failure. The upgrade will take place in an unconditional manner and will proceed even if there is a failure in the subsequent upgrade iteration. Default value is false.
- `Skip-suspended` — This field will skip the SE(s) that are suspended in the previous upgrade iteration and proceed with the remaining SE(s) in the group. The default value is false.

```
[admin:controller]: >resume segroup se_group_refs <se-group-name>
```

```
[admin:controller]: >resume segroup se_group_refs seg-a
```

The following options are available for resuming SE Group Upgrade with Options:

```
[admin:controller]: >resume segroup skip_suspended se_group_refs Default-Group
action_on_error continue_upgrade_ops_on_error
```

```
[admin:controller]: >resume segroup skip_suspended se_group_refs Default-Group
action_on_error suspend_upgrade_ops_on_error
```

Use the following API POST method to resume SE group upgrade.

```
API: /api/segroup/resume
POST /api/segroup/resume
JSON data:
{
        "se_group_uuids": [
                "serviceenginegroup-ec9c8141-844d-467d-bdc0-d7855e9d8419"
        ],
        "skip_warnings": true
}
```

Note When `skip_warnings": true` is used, upgrade proceeds without exhibiting warnings messages and upgrade previews.

Resume with other options:

Option: Skip suspended SE's and continue with the upgrade, update the `se_group action_on_error` with `CONTINUE_UPGRADE_OPS_ON_ERROR`.

```
{
        "se_group_options": {
              "action_on_error": "CONTINUE_UPGRADE_OPS_ON_ERROR",
              "skip_suspended": true
         },
        "se_group_resume_options": {
              "action_on_error": "CONTINUE_UPGRADE_OPS_ON_ERROR",
              "skip_suspended": true
         },
        "se_group_uuids": [
              "serviceenginegroup-ec9c8141-844d-467d-bdc0-d7855e9d8419"
        ],
        "skip_warnings": true
}
```

Option: Skip suspended SE's and continue with the upgrade.

```
{
        "se_group_options": {
              "action_on_error": "SUSPEND_UPGRADE_OPS_ON_ERROR",
              "skip_suspended": true
        },
        "se_group_resume_options": {
              "action_on_error": "SUSPEND_UPGRADE_OPS_ON_ERROR",
              "skip_suspended": true
        },
        "se_group_uuids": [
              "serviceenginegroup-ec9c8141-844d-467d-bdc0-d7855e9d8419"
        ],
        "skip_warnings": true
}
```

Creating SE Group using SE Group Template

During cloud initiation and SE Group creation, it is checked if the cloud has a Service Engine group template. If the template is available, the base image or patch image is copied from the SE group template. Otherwise, the image information is picked from the NSX Advanced Load Balancer Controller.

For each cloud, there is a `se_group_template_uuid`. This is used to ensure that the newly created SE Group follow the `se_group_template_uuid`.

Any SE group can be designated as the template. If an SE group (*Seg1*) is assigned as the default SE group template, then the newly created SE group (*Seg2*) picks the base and patch image from Seg as shown below.

```
[admin:controller]: > show upgrade status filter serviceenginegroup Seg2
+------+--------+--------------+------------------+-----------+------------------------------
+----------------------------------+
| Name | Tenant | Cloud         | State            | Operation | Image
| Patch                            |
+------+--------+--------------+------------------+-----------+------------------------------
+----------------------------------+
| Seg3 | admin  | Default-Cloud | UPGRADE_FSM_INIT | None      | 18.2.9-9000-20200509.052234
| 18.2.9-9000-2p1-20200430.133146 |
+------+--------+--------------+------------------+-----------+------------------------------
+----------------------------------+
```

The patch rollback option should be enabled as shown below.

```
[admin:controller]: > show upgrade status detail filter serviceenginegroup Seg2
+-----------------------
+-------------------------------------------------------------------------------+
| Field                 |
Value                                                                          |
+-----------------------
+-------------------------------------------------------------------------------+
| uuid                  |
serviceenginegroup-d564305e-9db5-4ae6-941c-485a26af062a                        |
| name                  |
Seg2                                                                           |
| node_type             |
NODE_SE_GROUP                                                                  |
| version               |
18.2.9-9000-20200509.052234                                                    |
| image_ref             |
18.2.9-9000-20200509.052234                                                    |
| patch_version         |
2p1                                                                            |
| patch_image_ref       |
18.2.9-9000-2p1-20200430.133146                                                |
| state
|                                                                                 |
|   state               |
UPGRADE_FSM_INIT                                                               |
|   last_changed_time   |
```

```
Sat May  9 06:15:49 2020 ms(41) UTC                                          |
| seg_status
|                                                                              |
|   notes[1]            |
[2020-05-09 06:15:49] Init segroup(seg3 defaults to seg-template(seg1). |
| start_time            |
2020-05-09 06:15:49.406502                                                   |
| enable_rollback       |
False                                                                        |
| enable_patch_rollback |
True                                                                         |
| progress              |
0 percent                                                                    |
| tenant_ref            |
admin                                                                        |
| obj_cloud_ref         |
Default-Cloud                                                                |
+-----------------------
+------------------------------------------------------------------------------|
```

Upgrading SE Group

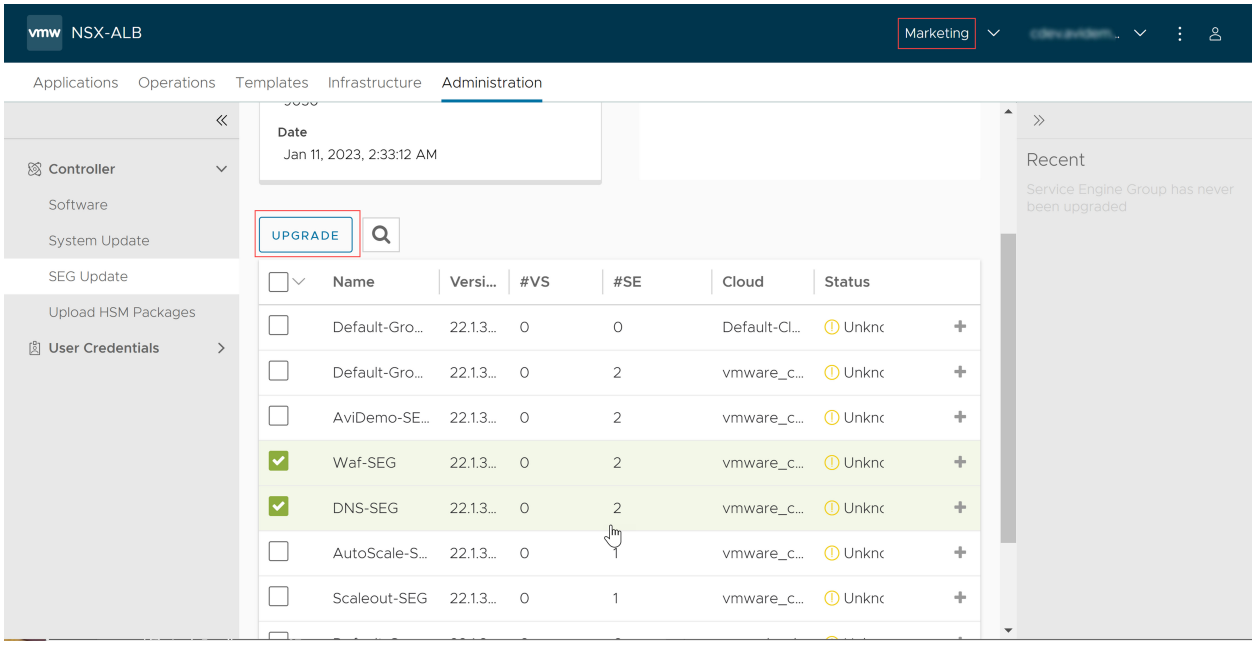
This interface is used to upgrade all or some of the SE groups.

Using UI

The following section details the steps to perform SE Group Update through the NSX Advanced Load Balancer Controller UI.

Navigate to **Administrator > Controller > SEG Update**, select the required SE group, and click **UPGRADE** to proceed with the update process.

Starting with NSX Advanced Load Balancer 22.1.3, SE group upgrade can be initiated in both admin and non-admin tenants as required.



Using the CLI

Log in to the NSX Advanced Load Balancer shell prompt to use the various options available for the SE group upgrade.

- Use the `upgrade segroup se_group_refs Default-Group image_ref<image name>` command to upgrade an SE group to the Controller image.

```
[admin:-controller]: >upgrade segroup se_group_refs Default-Group image_ref
18.2.6-9134-20191101.042535
```

- Use the `upgrade segroup se_group_refs Default-Group image_ref *lt;Controller image> se_patch_ref <SE patch name>`: command to upgrade an SE group to the Controller image and the SE patch image.

```
[admin:-controller]: >upgrade segroup se_group_refs Default-Group image_ref
18.2.6-9134-20191101.042535 se_patch_ref 18.2.6-9134-2p1-20190806.011824
```

Using the REST API

SE Group UUID is obtained by the `GET /api/serviceenginegroup API`.

The followings are the additional options for SE group upgrade:

Disruptive

This option disables a non-disruptive mechanism to facilitate a faster upgrade. If enabled, the SE(s) are upgraded in a disruptive manner. The default value is `false`.

Suspend-on-failure

This option suspends the upgrade of subsequent SE(s) within a SE-group when a failure is encountered in the SE upgrade path. The default value is `false`.

The followings are the different APIs for the SE group upgrade:

- Use the following API to upgrade the SE group to the Controller image.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
  {
     'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
     'se_group_uuids': [
       'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
     ]
  }
  ```

- Use the following with the additional SE Group options — Disruptive and Suspend_on_failure.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
  {
      'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
      'se_group_uuids': [
        'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
      ],
      'disruptive':true,
      'suspend_on_failure': true
   }
  ```

- Use the following API to upgrade the SE group to the Controller image and the SE patch image.

  API: `/api/upgrade`

  Method: `POST`

  `JSON Data`:

  ```
    {
        'image_uuid': 'image-b8adc2bd-d27f-469d-b78d-5e2bc14a14e4',
        'se_patch_uuid': 'image-e3aaad68-5aaf-485a-8bd9-1db3ec562d6a',
        'se_group_uuids': [
            'serviceenginegroup-e553b1a6-4851-4e82-ad12-cecc4bbda6c7'
        ]
    }
  ```

Additional Options for SE Group Upgrade

The following upgrade options are available for upgrading SE group:

Option	Behaviour	Notes
SUSPEND_UPGRADE_OPS_ON_FAILURE	This option is used to suspend the upgrade operations (Upgrade/Patch) on SE-Group if the SE(s) hit an issue and does NOT come up during the upgrade operations.	It is enabled by default.This option serializes the SE upgrades in the SE group upgrade. It increases the overall upgrade time for the entire SE group. Batch size is used to decrease the upgrade time. Even if the SEs do not have scaled-out virtual services, it still upgrades serially.
CONTINUE_UPGRADE_OPS_ON_FAILURE	This option is used to continue the upgrade or patch upgrade operations on the SE group even if the SE(s) hit an issue and does not come up during the upgrade operations. Service disruption can be observed.	This option parallelizes the SE upgrade in the SE group upgrade if SEs do not have scaled-out virtual services. If SEs have scaled-out virtual services, then they continue with serial upgrades.
Disruptive	This option is used to disable the non-disruptive nature of SE upgrades. It is used to upgrade all the SE(s) in the group to the next version irrespective of the traffic disruption.	This option is disabled by default. All SE(s) will be upgraded in parallel, irrespective of scaled-out virtual service existence. Traffic/Service disruption will take place.

Check Failed Error during Service Engine Upgrade

The NSX Advanced Load Balancer SE upgrade can fail with the error `Check failed`. This topic covers the troubleshooting and resolution steps for this error.

The following errors were observed in the SE log:

```
[127.1.0.3] Executing task 'se_upgrade_check'
cat: /host/prev/bootstrap/VERSION: No such file or directory
se_upgrade_check error exception: Error reading SSH protocol banner
```

The following error was observed in the `secure_channel.log` (in `/var/log/upstart` in SE CLI):

```
AVI-SE:/var/log/upstart
Authenticated to 10.1.1.1 ([10.1.1.1]:22).^M

connect_to localhost: unknown host (Name or service not known)^M

connect_to localhost: unknown host (Name or service not known)^M

connect_to localhost: unknown host (Name or service not known)^M

connect_to localhost: unknown host (Name or service not known)^M

connect_to localhost: unknown host (Name or service not known)^M
```

Resolution

The errors observed in the SE log file suggest that the entry for the localhost mapping to 127.0.0.1 is missing in `/etc/hosts`. This issue occurs when name resolution for the local host from SE is not successful as shown in the following snippet.

```
root@app-node3:/var/log/upstart# ping localhost

ping: unknown host localhost
```

Check `/etc/hosts` file by using `cat` command. Note that the localhost entry was missing as shown below.

```
app-node3.avi-systest.local-avitag-1root@app-node3:/# cat /etc/hosts

127.0.0.1 app-node3.avi-systest.local--nameless-abc-xyz

#used by abc_servers: node1.controller.local

{127.0.0.9, 10.140.88.197}

127.0.0.9 node1.controller.local

#used by abc_servers: node2.controller.local {127.0.0.8, 10.140.88.199}

127.0.0.8 node2.controller.local

#used by abc_servers: node3.controller.local {127.0.0.7, 10.140.88.198}

127.0.0.7 node3.controller.local

root@app-node3:/#
```

To solve this issue, add the localhost entry **`127.0.0.1 localhost`** to the localhost file. After adding the localhost entry, try again upgrading the SEs.

Additional Troubleshooting Tips

Usually, `/etc/hosts` content on the `root1` folder has this entry. From the NSX Advanced Load Balancer Controller, run the following command:

```
ssh -i /etc/ssh/id_se aviseuser@10.1.1.1
```

If ssh is successful, no issue is observed with the localhost entry.

Run the following command:

```
ssh -i /etc/ssh/id_se -p 5097 aviseuser@127.1.0.3
```

If ssh fails, add an entry to `/etc/hosts` file with the localhost 127.0.0.1 and retry ssh.

If ssh is successful, retry upgrading NSX Advanced Load Balancer SEs.

Patch Guide

This section discusses about the patch upgrade options.

Upgrading using Patch Release

The followings are the available options for patch upgrade:

- System — Patch upgrade for NSX Advanced Load Balancer Controller and all SE groups.
- Controller — Patch upgrade for the NSX Advanced Load Balancer Controller alone.
- SE group — Patch upgrade for some or all the SE groups.

Note The following are a few points for a patch upgrade process:

- An image with a patch can be applied.
- The image and the patch must have the same base version.
- A patch cannot be applied without applying the image.
- Compatibility checks prevent incorrect patches from getting applied to different versions.

Patch Upgrade Process for NSX Advanced Load Balancer

The patches are designed not to interrupt active services. In cases where an interruption is expected, the patch package is released with related documents and details. To ensure configuration integrity, changes to the configuration are locked out during a patch upgrade.

Patch Process

1 Download a patch package from the NSX Advanced Load BalancerCustomer Portal.

2 There can be as many as three packages for every patch release.

3 Use the patch shell command to apply the desired patch as follows:

   a The `controller_patch` and the `se_patch` provide the administrator with an option to patch some but not all aspects of the NSX Advanced Load Balancer.

   b By applying the `se_patch`, one has the flexibility to upgrade just some SE groups.

   c The `avi_patch` (system patch) applies to all the other patches.

For more information, see Patch Upgrade Options.

Preparing for the Patch

This section discusses the steps, prerequisites and restrictions while preparing for the patch.

Finding the Version

One or more patch packages may be applicable to a specific NSX Advanced Load Balancer version. Therefore, it is essential to know the version that the NSX Advanced Load Balancer is currently on. Check the NSX Advanced Load Balancer Controller or SE version(s) using the following commands:

- `show version controller`
- `show version serviceengine`

Prerequisites and Restrictions

- Choose the required patch package based on the Controller and SE versions.
- All Controllers must be on the same (base+patch) version to form a cluster. For instance, assume: A – is the major release and x.y – is the maintenance version. You cannot form a cluster with one Controller on A.x.y, one on A.x.y 2p1, and one on A.x.y 2p2.
- For more information, see Clustering Patched Controllers.
- All patches from a maintenance release are incorporated into successive maintenance releases.
- Once a NSX Advanced Load Balancer Controller is upgraded to a new maintenance release, all underlying SE groups must be upgraded to the same release as well.
- A patch family is the one in which the leading digit is the same. For instance, 1p1, 1p2, and 1p3 are patches in the 1px family.
- Fixes accumulate within a patch family. For instance, the 1p2 patch contains new fixes unique to it, plus all the fixes from 1p1. The 1p3 patch includes fixes from both the 1p1 and 1p2 patches. Additionally, the 2p1 patch is the first in a new patch family and does not contain 1px fixes.
- A given fix may appear in more than one patch family.
- The following options are allowed when selecting a patch version:
  - Choose any patch applicable to a particular maintenance release as the first patch to be applied to that base version.

    For example, in a patch family comprised of 1p1, 1p2 and 1p3, any one of the three can be the first applied.
  - Apply any subsequent patch, as long as it is within the same patch family. For instance, you can apply 1p5 to 1p1.
- The following options are not advisable while choosing a patch version:
  - Applying a patch from a patch family other than the one already chosen.

    For instance, you cannot apply patch 2p1 once any 1px patch has been applied.
  - Apply a patch that would imply an upgrade to a different NSX Advanced Load Balancer maintenance release.

For example, it is impossible to patch upgrade from 17.2.3 to 17.2.4-1p3.

- `.pkg` is the same for both container and non-container.
- For Controllers on BareMetal/LSC or legacy GCP, the upgrade package is available in `docker.tgz`.

Upload the Patch Package

Use WinSCP or any similar tool to upload the patch package to the Controller.

The following are the ways to upload patch image to the NSX Advanced Load Balancer Controller.

- Copy the downloaded patch image to the NSX Advanced Load Balancer Controller`/tmp` directory and then upload it on NSX Advanced Load Balancer Controller using image API.
- Use the `curl` command to upload the respective patch packages.

Note The leader Controller ensures that the follower Controllers are on the same version. The Controller machine on the base version of NSX Advanced Load Balancer might be previously patched. Upload patch package by using image the API `/api/image/`.

Images should be uploaded before starting the upgrade process.

1 Use the upload `image filename <file path>` command to start uploading the image.

```
[admin:controller]: > upload image filename /tmp/se_patch.pkg
 Starting image upload...
   +-------------------+-----------------------------------------------------+
   | Field             | Value                                               |
   +-------------------+-----------------------------------------------------+
   | status            | SYSERR_SUCCESS                                      |
   | se_info           |                                                     |
   |   path            | image://20.1.1-5000-2p2-20200217.063645/se_patch.pkg |
   |   hash            | e337b2024fe8b1647128af9da3c66c83                    |
   |   build           |                                                     |
   |     min_version   | 15.2                                                |
   |     tag           | 20.1.1-5000-20200217.063645                         |
   |     build_no      | 5000                                                |
   |     patch_version | 2p2                                                 |
   |     version       | 20.1.1                                              |
   |     date          | 2020-02-17 06:36:45 UTC                             |
   |   patch           |                                                     |
   |     patch_type    | se                                                  |
   |     reboot        | False                                               |
   | uuid              | image-b26182c2-92d9-4523-9c5e-676371664038          |
   | type              | IMAGE_TYPE_PATCH                                    |
```

```
| tenant_uuid       | admin                                                    |
| name              | 20.1.1-5000-2p2-20200217.063645                          |
+-------------------+----------------------------------------------------------+
Time Taken: 2.15626502037
```

Note Image upload is supported only on the Cluster Leader.

2 Use the `show image` command to view the image.

```
[admin:10-50-54-123]: > show  image
+------------------------------+-----------------------------------------
+-------------------+---------------------+
| Name                         | UUID
| Type              | State               |
+------------------------------+-----------------------------------------
+-------------------+---------------------+
| 20.1.7-9154-20210916.210140 | image-e4ffa292-be4e-45e0-b6f4-c4a5ee66fc66
| IMAGE_TYPE_SYSTEM | IMAGE_FSM_COMPLETED |
| 21.1.3-9003-20211202.115243 | image-f2325e62-cae2-47af-bfb0-7fd9ab00d5b4
| IMAGE_TYPE_SYSTEM | IMAGE_FSM_COMPLETED |
| 21.1.3-9007-20211204.000303 | image-d7764ab0-8ac3-4e58-8484-6ae5c77142f6
| IMAGE_TYPE_SYSTEM | IMAGE_FSM_COMPLETED |
+------------------------------+-----------------------------------------
+-------------------+---------------------+
```

3 Log in to the NSX Advanced Load Balancer shell using admin credentials. Use the `show upgrade status` and `show upgrade status detail` commands to check the upgrade status.

```
[admin:controller]: > show upgrade status
+---------------+---------------+-----------------------+-----------
+------------------------------+-------+
| Name          | Cloud         | State
| Operation | Image                        | Patch |
+---------------+---------------+-----------------------+-----------
+------------------------------+-------+
| cluster-0-1   | -             | UPGRADE_FSM_COMPLETED
| UPGRADE   | 18.2.8-9000-20200212.075158 | -     |
| Default-Group | Default-Cloud | UPGRADE_FSM_COMPLETED
| UPGRADE   | 18.2.8-9000-20200212.075158 | -     |
| se1 | Default-Cloud | UPGRADE_FSM_COMPLETED
| UPGRADE   | 18.2.8-9000-20200212.075158 | -     |
+---------------+---------------+-----------------------+-----------
+------------------------------+-------+
```

Patch Upgrade Options

This section discusses about the version upgrade and patch of the NSX Advanced Load Balancer.

Version Upgrade and Patch

NSX Advanced Load Balancer Controller can be upgraded to a more recent version along with the required patch as follows:

1 Use the `upgrade controller image_ref <image> controller_patch_ref <patch>` as shown below to upgrade the NSX Advanced Load Balancer Controller along with a patch.

```
[admin:controller]: > upgrade controller image_ref 18.2.7-5000-20200213.181331
controller_patch_ref 18.2.7-5000-2p1-20200213.182111
```

2 Use the `upgrade segroup image_ref <image> se_patch_ref <patch>` command to upgrade a SE group with a patch.

3 Use the `upgrade system image_ref <image> controller_patch_ref <patch> se_patch_ref <se_path>` command to upgrade the NSX Advanced Load Balancer System (Controller and SE groups) to the desired patch.

This ensures that the NSX Advanced Load Balancer Controller is upgraded and the desired patch is applied, at the same instance.

Note

- The patch should be of the same version as that of the Controller upgrade.
- `se_group_options` and `se_group_resume` options are available in the NSX Advanced Load Balancer CLI.

Additional Options for Patch Upgrade

Apart from this, the following are the four options for the patch command:

- Disruptive Patch
- Controller Patch
- SE Group Patch
- System Patch

Disruptive Patch

The disruptive patch option is set to `False` by default. The `se_group_refs` attribute governs the scope of the upgrade. If the non-disruptive rolling upgrade of Service Engines are not required, this flag can be set to `True` to go through the upgrade process quickly. This flag can be set to `True`, when the require.

The below command initiates an upgrade with the disruptive flag set to `True`.

```
[admin:controller]: > patch segroup
action_on_error   The error recovery action configured for a SE
Group.

disruptive        Disable non-disruptive
mechanism.
```

```
se_group_refs     SE Groups subjected to patch
operations.

se_patch_ref      Image name for identifying SE patch
image.

skip_warnings     This is flag when set as true skips few optional must checks.
```

Note The `action_on_error` option is supported for SE group upgrade.

Controller Patch

```
[admin:10-50-54-122]: > patch  controller controller_patch_ref
18.2.12-9110-2p1-20210220.231845
```

SE Group Patch

If the `se_group_refs` option is not enabled, all SE groups are upgraded. When enabled, it identifies a specific SE group for patching. If more than one SE group require patching, each will require a separate patch command.

```
[admin:controller]: > patch segroup se_group_refs Default-Group se_patch_ref
18.2.8-9000-1p2-20200219.121101
```

```
[admin:controller]: > patch segroup se_group_refs Default-Group se_group_refs Default-Group
abc-group se_patch_ref 18.2.8-9000-1p2-20200219.121101
```

Note If the `se_group_refs` option is not enabled, all SE groups are upgraded.

System Patch

Use the following command to patch the NSX Advanced Load Balancer Controller along with system patch.

```
[admin:controller]: > patch system controller_patch_ref 18.2.8-9000-1p3-20200219.121643
se_patch_ref 18.2.8-9000-1p3-20200219.121643


action_on_error   The error recovery action configured for a SE
Group.
disruptive        Disable non-disruptive
mechanism.
```

```
skip_warnings     This is flag when set as true skips few optional must checks.
```

Note

- SEs check for the version present on the Controller. In the event of a mismatch, the SE is rebooted and upgraded with the new patch available on the Controller.
- If a patch 18.2.6-5p1 is applied to the SE group, then all the entities in the system (SEs and Controller) — can only be upgraded to 5p1 or some member of the 5px patch series. For example, a different patch series 6p1 can be applied to the NSX Advanced Load Balancer Controller and 5p1 to the SE group.

Patch Upgrade for NSX Advanced Load Balancer System

Use the following CLI command for the base image upgrade with a patch image.

```
[admin:controller]: > upgrade system image <image-name>  controller_patch <controller-patch-name> se_patch <se-patch-name>
[admin:controller]: >upgrade system image 18.2.6 controller_patch 18.2.6-1p1 se_patch 18.2.6-1p1
```

1 Use the `upgrade system image_ref <image name > controller_patch_ref <SE patch name>` command for the NSX Advanced Load Balancer.

```
[admin:-controller]: upgrade system image_ref 18.2.6-9000-20191031.063017 controller_patch_ref 18.2.6-2p1-20191031.063017
```

2 Use the `upgrade system image_ref <image name> se_patch_ref <SE patch name>` command for the NSX Advanced Load Balancer.

```
[admin:-controller]:  upgrade system image_ref 18.2.6-9000-20191031.063017 se_patch_ref 18.2.6-2p1-20191031.063017
```

3 Use the `upgrade system image_ref <image name> controller_patch_ref <Controller patch image> se_patch_ref <SE patch image>` command for the system upgrade with both Controller and SE patch.

```
[admin:-controller]:upgrade system image_ref 18.2.6-9000-20191031.063017 controller_patch_ref 18.2.6-2p1-20191031.063017 se_patch_ref 18.2.6-2p1-20191031.063017
```

Patch Upgrade for NSX Advanced Load Balancer Controller

This interface is used to patch upgrades for the NSX Advanced Load Balancer Controller.

Patch Upgrade Using UI

Patch Upgrade involves uploading the patch image for the controller and service engines.

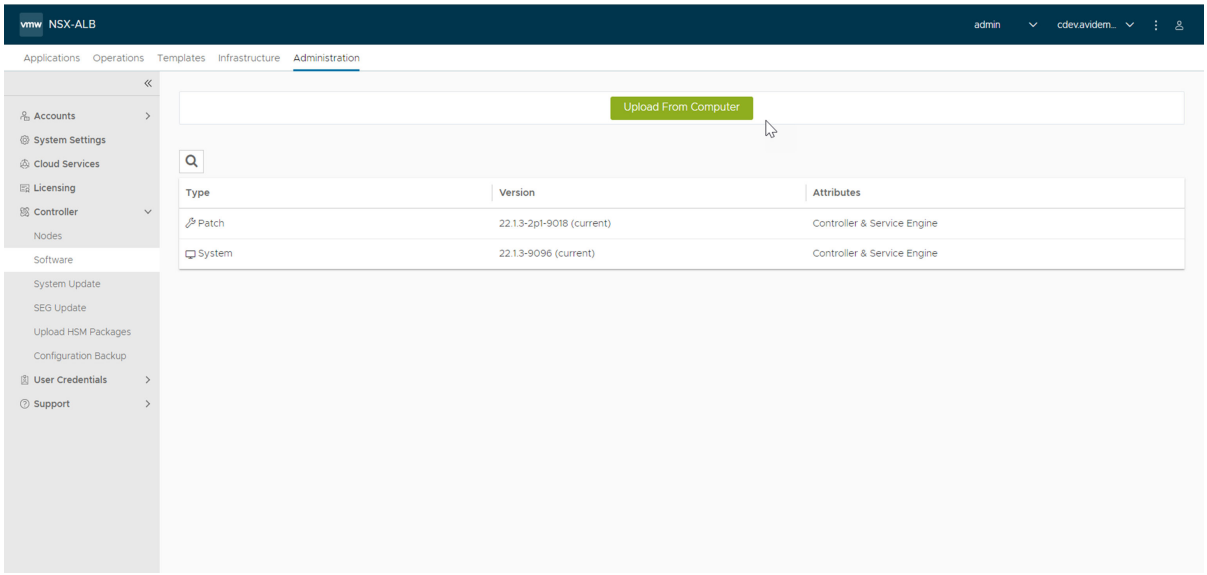
The following patch upgrade options are available:

- Patch upgrade for Controller

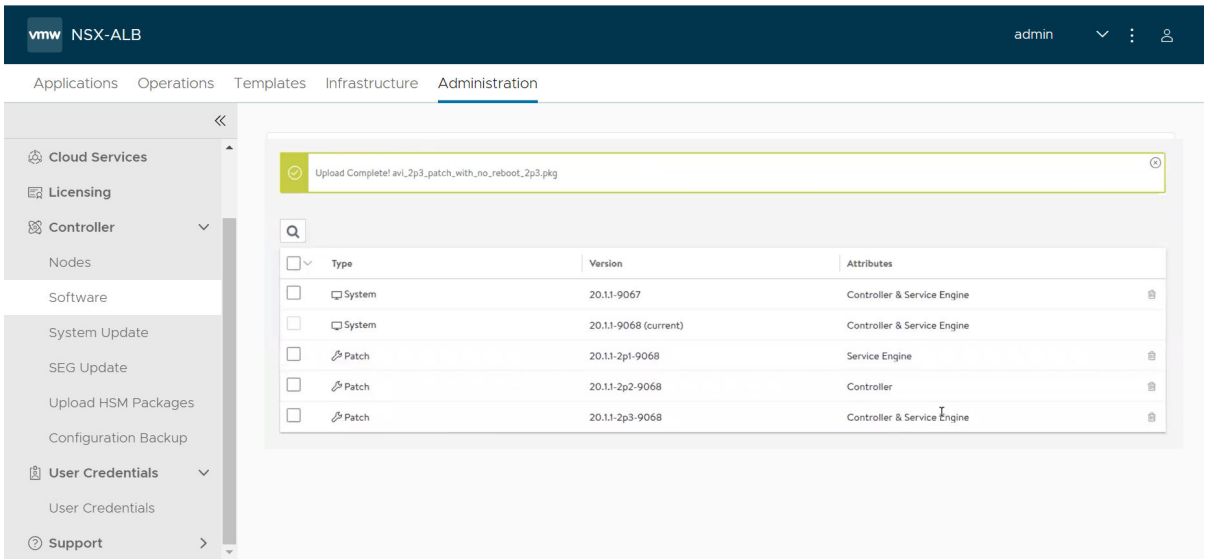
- Patch upgrade for Controller and Service Engines
- Patch upgrade for Service Engines

Upload the Patch Image

1 Navigate to **Administration > Controller > Software**. Click **Upload From Computer** to upload the patch image to the Controller. Once the patch file is selected, the upload of the patch image starts. The status of the patch image upload progress is available on the UI. In the following example, the Controller's patch image is uploaded.

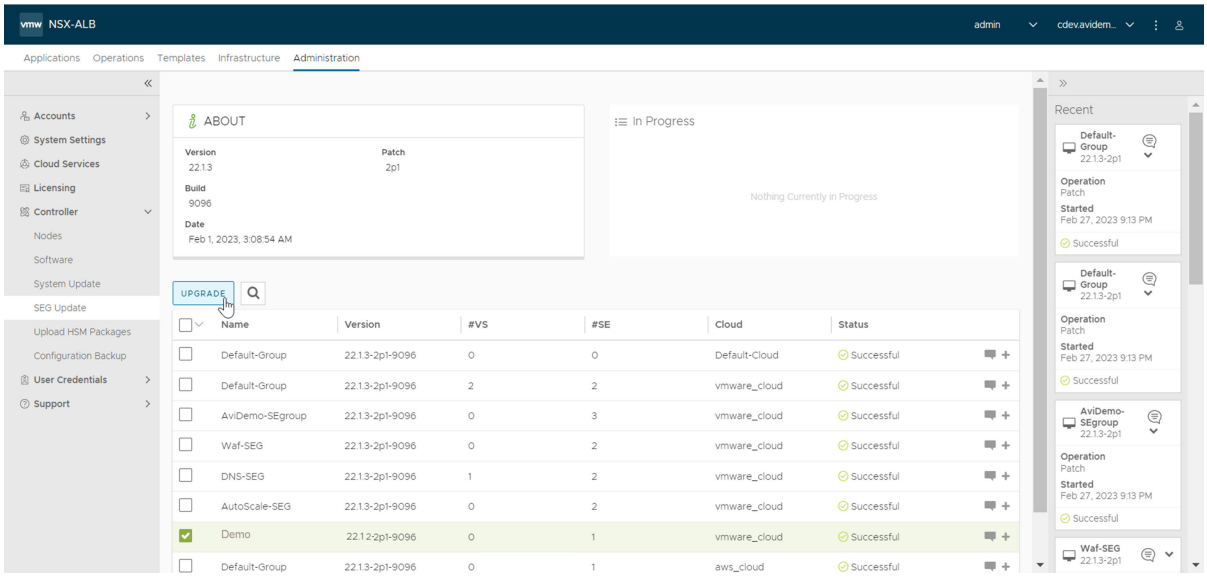


2 A software image containing patch files for the Controller and Service Engines can also be uploaded using the same process. The following screenshots exhibit the patch upload for the Controller and Service Engine. Patch upgrades for only Service Engines are also supported.

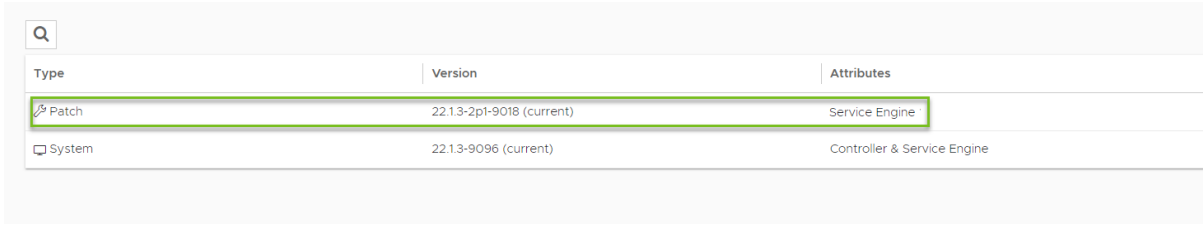


3 To upgrade the Service Engine group with the patch image, use the patch upgrade file for Service Engines. The process to patch upgrade the Controller or Service Engine Groups is the same as mentioned for the main software release. For more information, see Using the UI.

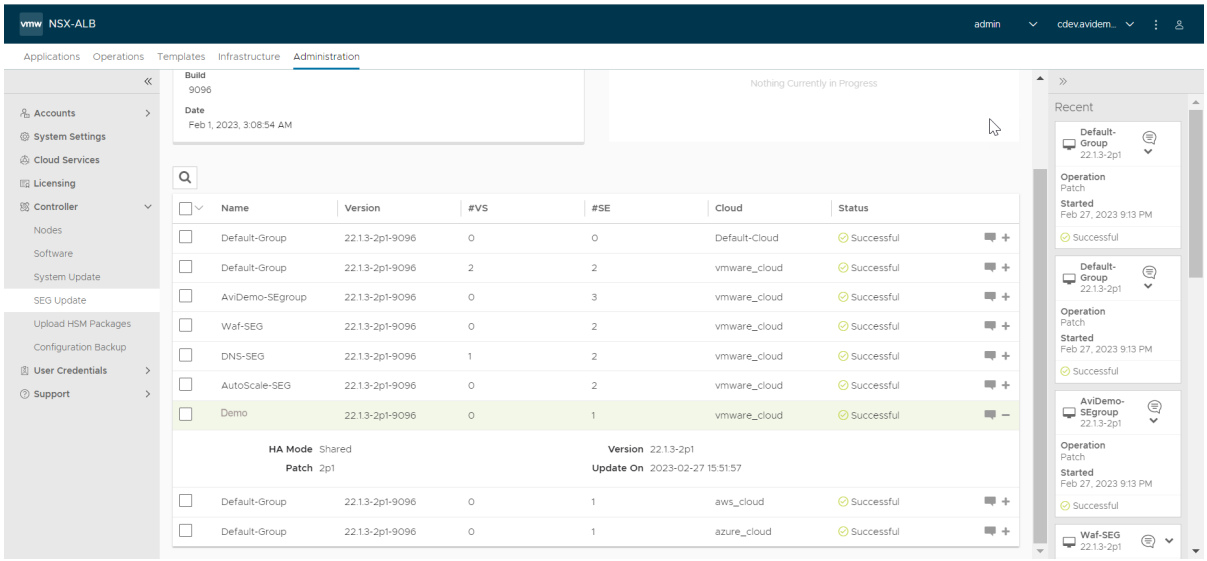
4 Navigate to **Administration > Controller > SEG Update**, select the required Service Engine Group, and click **UPGRADE**, as shown below.



5 For Service Engine Group update, select only the SE patch, as shown below.



6 Once the patch update for the SE group is completed, the UI exhibits the status as successful, as shown below. The selected SE group is updated with the 22.1.3-2p1 patch.



Using the CLI

Use the `upgrade controller image_ref <image name> controller_patch_ref <Controller patch image` command to upgrade the NSX Advanced Load Balancer Controller with a patch.

```
[admin:-controller]: upgrade controller image_ref 18.2.6-9000-20191031.063017
controller_patch_ref 18.2.6-2p1-20191031.063017
```

```
[admin:controller]: > patch controller <patch-name>

[admin:controller]: > patch controller controller_patch 18.2.5-5p1
```

Using the REST API

This section describes the using NSX Advanced Load Balancer REST API.

```
POST api/upgrade JSON data:{'controller_patch_uuid': <image-uuid>}
```

Patch Upgrade for SE Group

SE groups can be of different versions and different versions of patches can be applied.

Use the `upgrade segroup image_ref <image name> se_group_refs Default-Group se_patch_ref <patch for the SE Group>` command to upgrade specific SE groups along with a patch.

```
[admin:-controller]: upgrade segroup image_ref 18.2.6-9000-20191031.063017 se_group_refs
Default-Group se_patch_ref 18.2.6-2p1-20191031.063017
```

Note Patch name and patch UUID is retrieved from the image service.

API - HTTP PATCH Support

This section explains the usage of HTTP PATCH method.

NSX Advanced Load Balancer supports use of PATCH to perform the following operations:

- Update scalar fields (string, bool, int32, etc.).
- Add a new entry to a list in an object.
- Update an entry in a list in an object.
- Unset scalar fields - This causes the fields to be reset to their default values, if applicable.
- Delete an entry from a list in an object - According to section 5.1.1 of RFC2616, “The Method token indicates the method to be performed on the resource identified by the Request-URI. The method is case-sensitive.” So spell it “PATCH” to avoid “400 bad request” errors from NSX Advanced Load Balancer when executing the HTTP PATCH request.

Editing Nested Fields

In the current release, editing nested fields using PATCH is supported only to a field depth of 1. PATCH is not supported for editing objects with deep nesting characteristics. To accomplish an edit to a nested field, PATCH expects the data to be in the following form:

```
{
    "add": {

    }
}
```

or

```
{
    "replace": {

    }
}
```

or

```
{
    "delete": {

    }
}
```

For scalar fields, add or replace are equivalent as they replace the value of the scalar field with the value provided. In case of a list, add indicates that the specified set of entries needs to be added to the list and replace indicates that the list itself is replaced with what is specified in the request. Action delete is used to remove specified entries from the list.

Examples

The following examples use PATCH to update server information in a pool. The pool is identified by its UUID.

- Add Servers to a Pool- This request to the NSX Advanced Load Balancer API adds two new servers (1.1.1.1 and 1.1.1.2) to an existing pool.

```
API: /api/pool/pool_uuid
Data:
{
    "add": {
        "servers": [
            {
                "ip": {
                    "addr": "1.1.1.1",
                    "type": "V4"
                }
            },
            {
                "ip": {
                    "addr": "1.1.1.2",
                    "type": "V4"
                }
            }
        ]
    }
}
```

- Update Server Information in a Pool - This request to the NSX Advanced Load Balancer API updates one of the servers in an existing pool.

```
API: /api/pool/pool_uuid
Data:
{
    "add": {
        "servers": [
            {
                "ip": {
                    "addr": "1.1.1.1",
                    "type": "V4"
                },
                "ratio": 10
            }
        ]
```

```
    }
}
```

Note If a field is an array of structures, each structure is typically identified by a key (combination of the fields within the structure). This is used to identify the element in the list to update. In the case of pool servers, the server key is identified by ip, port.

- Replace Servers in a Pool with a new set of Servers - This request replaces the entire server list with a new server list. The other fields of the pool remain intact.

```
API: /api/pool/pool_uuid
Data:
{
    "replace": {
        "servers": [
            {
                "ip": {
                    "addr": "3.3.3.3",
                    "type": "V4"
                },
            },
            {
                "ip": {
                    "addr": "3.3.3.4",
                    "type": "V4"
                },
            }
        ]
    }
}
```

- Delete a Server from a Pool- This request deletes one of the servers from an existing pool.

```
API: /api/pool/pool_uuid
Data:
{
    "delete": {
        "servers": [
            {
                "hostname": "www.example.com",
                "ip": {
                    "addr": "3.3.3.3",
                    "type": "V4"
                },
                "resolve_server_by_dns": true
            }
        ]
```

```
    }
}
```

- Updating Scalar Fields - The examples in this section set some scalar fields. The following request enables HTTP request queuing and sets the default server port to 8080.

```
API: /api/pool/pool_uuid
Data:
{
    "delete": {
        "default_server_port": 8080
    }
}
```

The following request resets the default server port to the system default, by deleting its explicit setting from the pool.

```
API: /api/pool/pool_uuid
Data:
{
    "replace": {
        "request_queue_enabled": True,
        "default_server_port": 8080
    }
}
```

Supported Objects

PATCH is officially supported for the following objects and fields:

Object	Fields
Cloud	linuxserver.hosts
GslbService	enabled
IpAddrGroup	addrs
	prefixes
	ranges
MicroServiceGroup	service_refs
Pool	health_monitor_refs
	servers
StringGroup	kv
	type
SystemConfiguration	dns_virtualservice_refs
	global_tenant_config

Object	Fields
	snmp_configuration.community
VirtualService	http_policies

Asynchronous Patch

Adding pools and deleting pools at a high rate requires applying patches at a high rate. The size of the deployment often adds more load on the Controller and increases the latency.

The Asynchronous Patch feature allows patch requests to be queued, consolidated in fixed intervals, and updated together with minimal latency.

Note This feature is recommended only in case of large deployment size.

When *async_patch_merge_period* is 0 in Controller properties all PATCH calls on pool with `?async_mode=true` are converted into synchronous call and 200 response is returned.

Using the Asynchronous Patch Update

By default, this feature is disabled. To enable asynchronous patch:

1. Change the *async_patch_merge_period* in the Controller properties from 0 (default value) to the required interval (30-120 seconds).
2. Pass the `async_mode` as a query parameter in the patch request. For example, `PATCH /api/pool/<pool-uuid>?async_mode`. The response displays a unique *patch_id*, which can be used to view the status of the request. The patch consolidation is initiated, thereby merging the pending requests in the queue.
3. Review the status using a GET request on `/api/pool-async/status/<patch_id>`. The status is displayed as follows:

```
{
                "uuid": <uuid of the pool>,

                "status": "QUEUED | IN_PROGRESS | SUCCESS | FAIL",

                "created": <timestamp of request creation>,

                "error_message": <error message in case of failure>,

                "status_code": <status code for server PATCH>

            }
```

Caveats

- Currently, this feature is available only for the servers field (with IP4) in the pool object.
- This feature is limited to only adding or deleting servers in a pool

- All patch requests to the Controller within an *async_patch_merge_period* must have the same API version (*HTTP_X_AVI_VERSION*)
- Patch updates are not installed until the next update cycle, as defined in the time interval. So the patch updates are not real time.

Rollback

Rollbacks are non-disruptive. When a rollback operation is performed, the NSX Advanced Load Balancer Controller or SEs will transition to the previous major version of the software. Selective rollback is possible for the Controller and SE groups.

The following options are available:

- Rollback for System
- Rollback for NSX Advanced Load Balancer Controller only
- Rollback for some or all the SE groups

Note Rollback of the SE Group will be to the previous version.

Rollback for System

The rollback of the system will result in the rollback of the SE(s) followed by the rollback of the NSX Advanced Load Balancer Controller.

Use the following CLI and REST API for performing rollback for a patch version for the NSX Advanced Load Balancer system (Controller and SE groups).

Using the CLI

```
[admin:controller]: > rollback system
```

Using the REST API

```
POST api/rollback JSON data:{'system':true}
```

```
POST api/rollback JSON data:{'system':true,'rollback_type':2}
```

Rollback for NSX Advanced Load Balancer Controller

This interface is used to rollback the NSX Advanced Load Balancer Controller.

Using the CLI

```
[admin:controller]: > rollback controller
```

Using the REST API

```
POST api/rollback
```

Rollback for SE Groups

This interface is used to rollback for SE groups.

Using the CLI

```
[admin:controller]: > rollback segroup <se-group-name>
[admin:controller]: > rollback segroup  seg-a
```

Using the REST API

```
POST api/rollback JSON data:{'se_group_uuids': ['seg-a-uuid']}
```

Rollback - Patch

Rollback of a patch release transitions the software to a version without the specific patch. It will not roll back to the previous major version.

Selective ability to rollback the patch on the NSX Advanced Load Balancer Controller and SE groups is available.

This interface is used to roll back the patch and not the major version.

The followings are the available options:

- System: rollback patch for NSX Advanced Load Balancer Controller and all SE groups
- Controller: rollback patch the NSX Advanced Load Balancer Controller only.
- SE-group: rollback patch for all or some of the SE groups.

Rollback Patch for System

Use the following CLI and REST API for performing rollback for NSX Advanced Load Balancer System (NSX Advanced Load Balancer Controller and SE groups).

Using the CLI

```
[admin:controller]: > rollbackpatch system
```

Using the REST API

```
POST api/rollback JSON data:{'rollback_type':2}
```

Rollback Patch for NSX Advanced Load Balancer Controller

Use the following CLI and REST API for performing rollback for a patch version for an NSX Advanced Load Balancer Controller.

Using the CLI

```
[admin:controller]: > rollbackpatch controller
```

Rollback Patch for SE Groups

Use the following CLI and REST API for performing rollback for a patch version for an SE group.

Using the CLI

```
[admin:controller]: > rollbackpatch segroup <se-group-name>
[admin:controller]: > rollbackpatch segroup  seg-a
```

Using the REST API

```
POST api/rollback JSON data:{'rollback_type':2,'se_group_uuids': ['seg-a-uuid']
```

Note See Additional Options for Flexible Upgrades for the following additional options:

- Rollback - Error Recovery
- Abort Cleanup
- SE Group Resume Option

Show Commands

The following show commands provide software version visibility in the system:

- `show version controller`
- `show version serviceengine`
- `show version serviceenginegroup`

The following commands provide upgrade visibility in the system:

- `show upgrade status`: Various filters will be implemented as per UI workflow.
- `show upgrade history`: This command is deprecated.

Note

- If the Controller is at the highest version, whereas the SE groups are at lower versions, commands might not work due to the higher version of the Controller.
- Due to the API version semantics, fields might not be available as they are deprecated in the annotation.
- Due to API endpoint deprecation, some internal commands might not work.

Alerts and Events

The following events are available to provide visibility:

- Image upload/delete events
- Upgrade-specific events
- Patch-specific events

- Rollback-specific events
- Rollback patch-specific events.
- Failures will translate into alerts.

Additional APIs

The following GET API calls are applicable:

- The following REST API provides information about all the images present in the system.

```
Get API: api/image/
```

- The following API provides information about a specific image whose UUID is passed as a slug.

```
Get API: api/image/image-uuid
```

- Use the following API to delete the image provided if not in use.

```
Delete API: api/image/image-uuid
```

- Inventory API - `api/image-inventory`: This API provides the image inventory on the system. It filters based on various options such as retrieve all packages for a version and so on.

Reset to Factory Defaults

In some scenarios, it might be useful to reset the NSX Advanced Load Balancer to its factory default settings. With a few exceptions, factory reset is equivalent to completely reinstalling a fresh deployment. As part of the factory reset, the Controllers and SEs reboot and delete their load balancing and proxy configurations, while still maintaining their basic networking configurations.

Following a factory reset, on logging in to the Controller through the web interface, the initial setup wizard starts.

Configure the settings as required, including setting a new password.

Resetting to Factory Defaults

To reset to factory defaults, run the following command from the NSX Advanced Load Balancer shell.

```
: > | reboot clean
```

This is different from the reboot command in the Linux CLI.

On reboot clean, a new self-signed certificate is generated for the Controller UI. If you are logged into the UI, the browser might not redirect the page to login, as it sees a certificate change. Refresh the page in the browser to get back to the login page.

Note

- In the write access mode deployments, if an SE is not given a new configuration within a configured interval, it is deleted. In the web interface, this interval is configurable on the **Advanced** tab for the SE group, in the **Delete Unused Service Engines After** field.
- When reboot clean is issued, the SE virtual machines get picked up by the new Controller and are put in the default cloud. You cannot use or remove them other than from the underlying environment, e.g., VMware vCenter.
- In case of a cluster, reboot clean is applicable only from a leader node in the cluster.
  - On performing a reboot clean on a leader node in the cluster, the cluster remains intact, including the UUID. You have to reset the password from the NSX Advanced Load Balancer UI.
  - In case of a no-access deployment, the SEs are moved from the no-access cloud to the default cloud, as the cloud config is deleted.

Settings Retained After Factory Reset

The following are the settings retained after factory reset.

- SE and Controller virtual machines.
- SE and Controller management networking properties, such as IP addresses and default gateways.

Settings Removed After Factory Reset

The following are the settings removed after factory reset.

- Licenses.
- Virtual service configurations.
- Pool configurations.
- SSL certificates, including certificates that might have been used for the Controller UI.
- Backups of the configuration.
- Logs and metrics.
- RBAC, users, passwords.

Locking a Linux System to a Specific OS Version

This section explains the ways to prevent Specific OS Version from being updated.

CentOS Linux

To prevent CentOS from being updated beyond a release level, it's necessary to appropriate set the `$releasever` parameter. Create a new file, `/etc/yum/vars/releasever`, containing the value of the highest point release to which an update is acceptable.

This file creation can be done in two ways:

```
head -n1 /etc/centos-release | awk '{print $4}' > /etc/yum/vars/releasever
```

```
echo '7.2.1511' > /etc/yum/vars/releasever
```

The first code automatically restricts to the release that is currently running.

Red Hat Enterprise Linux

RHEL is a bit more complicated, as there are many possible options, which are detailed here: https://access.redhat.com/solutions/238533. To summarize all but the EUS Subscription details:

Systems not Registered to Customer Portal or Satellite

Any of the following will work.

- Modify the `/etc/yum.conf` file under the [main] heading:

  ```
  [main] distroverpkg=7.2
  ```

- Create the var file to override $releasever:

  ```
  head -n1 /etc/redhat-release | awk '{print $7}' > /etc/yum/vars/
  releasever
  ```

- Alternatively you can also use:

  ```
  echo '7.2' > /etc/yum/vars/releasever
  ```

Systems Registered to Customer Portal or Satellite

- See a list of possible releases:

  ```
  subscription-manager release --list
  ```

- Set the release:

  ```
  subscription-manager release --set=7.2
  ```

- Clean your yum cache:

  ```
  yum clean all
  ```

- Verify the system is set to the correct release:

  ```
  subscription-manager release --show
  ```

Increasing Docker Container Size

The devicemapper basesize changes will take effect only for containers created after the change. Existing containers will keep using the basesize that was originally present. Follow according to the type of host on which commands are being run.

For Controllers Without Any Co-Located SE

Follow the step-by-step procedure and run the commands from follower nodes to the leader node. Wait for the cluster to become active after each node is configured:

1 `$ systemctl stop avicontroller.`

2 Check if docker thinpool has more space to increase from 10GB to 30GB (more if devicemapper volume has the capacity).

3 Edit `/etc/docker/daemon.json` to have basesize set to at least 30G:

```
{

"storage-driver": "devicemapper",

"storage-opts": [

"dm.thinpooldev=/dev/mapper/docker-thinpool",

"dm.use_deferred_removal=true",

"dm.use_deferred_deletion=true",

"dm.basesize=30G" # <-- Add this

]
```

4 Run the following commands:

`$ sudo systemctl daemon-reload` and `$ reboot #`, for both kernel and devicemapper changes to take effect.

5 Verification:

`$ docker info # pool base device size` should show as 30GB (won't reflect inside container yet, new container needs to be created for that).

6 Verify cluster is in active state (from controller CLI: `show cluster nodes`).

For Controllers With Co-Located SE

Follow the step-by-step procedure and run the commands from follower nodes to the leader node:

1 `$ systemctl stop avicontroller.`

2 Check if docker thinpool has more space to increase from 10GB to 30GB (more if devicemapper volume has the capacity).

```
[root@vzn-ctrl-1 ~]# lvs -a # LSize is 37G

LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync

thinpool docker twi-aot--- <37.11g 20.08 4.30
```

3 Edit `/etc/docker/daemon.json` to have basesize set to at least 30G:

```
{

"storage-driver": "devicemapper",

"storage-opts": [

"dm.thinpooldev=/dev/mapper/docker-thinpool",

"dm.use_deferred_removal=true",

"dm.use_deferred_deletion=true",

"dm.basesize=30G" # <-- Add this

]

}
```

4 Run the following commands for both kernel and devicemapper changes to take effect:

`$ sudo systemctl daemon-reload`

`$ reboot #`

5 Verification:

`$ docker info # pool base device size` should show as 30GB (wont reflect inside container yet, new container needs to be created for that).

6 Verify cluster is in the active state (from controller CLI: `show cluster nodes`), SE is `OPER_UP` (`show servicengine </code>`), VLANs if any are still present on SE, basic traffic validation.

For SE Hosts

Follow the step-by-step procedure and run the commands from follower nodes to the leader node. Wait for the SE to become `OPER_UP` after each host is configured:

1 `$ systemctl stop avise`.

2 Verify SE is `OPER_UP` (from controller CLI: `show servicengine </code>`), VLANs if any are still present on SE, basic traffic validation.

Note

- The above changes will take effect for the Docker Container (controller/SE) on the next upgrade.
- You must rebuild the controller container if the upgrade is not being performed. As a result, the cluster will take effect for the above changes.

For more information on deleting and re-adding the cluster nodes, see Changing NSX Advanced Load Balancer Controller Cluster Configuration.

Migrating from F5 BIG-IP to NSX Advanced Load Balancer

9

When deploying NSX Advanced Load Balancer into existing environments, it is often required to migrate application workloads from other load balancers to NSX Advanced Load Balancer. NSX Advanced Load Balancer recognizes the stringent requirements of customers and the need to maintain up-time during a live migration. This section provides insight into the process of migrating from F5's BIG-IP LTM to NSX Advanced Load Balancer.

NSX Advanced Load Balancer provides a wide ranging breadth of functionality common in competitive application delivery controllers, or load balancers. But it also extends the load balancer's capabilities and value by incorporating extensive analytics data, a centralized control plane and modern distributed data plane architecture. Ultimately this creates an extremely intuitive load balancer fabric which is highly automated to reduce operational complexity and time to deploy, learn, and manage.

Concepts

While F5 and NSX Advanced Load Balancer provide many similar high level features, there are important differences architecturally, in the operations of various features, and even in the names of features and concepts.

To successfully migrate from F5, a VMware engineer will walk you through the differences to ensure the optimal deployment of NSX Advanced Load Balancer. Some customers choose to recreate the F5 configuration as exactly as possible. Other customers have room to re-architect and modernize their load balancing infrastructure.

Concept	F5 Term	NSX Advanced Load Balancer Term
application proxy	virtual server	Virtual Service For more information, see *Virtual Service* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*
group of servers	pool	Pool For more information, see *Server Pools* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*

Concept	F5 Term	NSX Advanced Load Balancer Term
Data plane scripting	iRules™	DataScript For more information, see *DataScripts Guide Overview* topic in the VMware NSX Advanced Load Balancer *DataScript Guide*
API	iControl™	REST API
Load balancer	BIG-IP™ LTM™ + GTM™	Service Engine For more information, see *Service Engine* topic in the VMware NSX Advanced Load Balancer *Installation Guide*
Connection aggregation	OneConnect™	Multiplexing For more information, see *Connection Multiplexing* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*
Central config manager	Enterprise Manager™ / BIG-IQ™	Controller For more information, see *Control Plane* topic in the VMware NSX Advanced Load Balancer *Installation Guide*
Orchestrator	none	Controller For more information, see *Control Plane* topic in the VMware NSX Advanced Load Balancer *Installation Guide*

Control Plane

Architecturally, NSX Advanced Load Balancer is managed by a Controller or a redundant cluster of Controllers. Rather than logging into and managing each pair of load balancer appliances, the NSX Advanced Load Balancer fabric is managed by the Controller cluster. A single Controller cluster may manage hundreds of Service Engines, even if they are deployed in different clouds such as VMware or OpenStack. Customers may also choose to deploy more than one Controller cluster, though this is usually done for geographically separate data centers. One cluster could manage both test and production environments separated by different tenants, or each could have their own cluster created.

Data Plane

NSX Advanced Load Balancer load balancers, called Service Engines (SEs), may be deployed similar to BIG-IP in active/standby pairs (using NSX Advanced Load Balancer's Legacy HA mode), or they may preferably be deployed in elastic HA mode, with fully active groups. There are a number of configuration options to carve out separate groups through tenants, VRFs, clouds,

and SE groups. Each application may have its own load balancer, or all applications may share a group of Service Engines. When migrating from BIG-IP, the appliance-pair versus fabric choice is one of the first architectural questions that should be answered to determine how to best consolidate and minimize unused load balancer capacity.

Migration

Migration from existing, live environments is a delicate process. NSX Advanced Load Balancer provides migration services, with a combination of migration tools and engineers with decades of experience in load balancing.

Automated

BIG-IP LTM configurations can be automatically imported into NSX Advanced Load Balancer's JSON config format. The BIG-IP to Vantage configuration migration tool imports virtual services, pools, and most adjacent functionality configured in the `bigip.conf` file. Additional configuration objects that can be automatically imported include groups, SSL keys and certificates. This eliminates the potential of errors when converting BIG-IP configuration files that are often tens of thousands of lines long. The configuration utility provides a complete output, showing every configuration setting that has been converted.

Manual

Some functionality from BIG-IP LTM cannot be automatically converted. For instance, F5's iRules are not migrated. NSX Advanced Load Balancer's experience is that about 75% of all iRules can be converted to native point-and-click features, though a professional services engineer will manually inspect the iRule to make that determination. iRules that cannot be performed as native features will be rewritten in NSX Advanced Load Balancer's DataScript format, which is similar in logic and function, but based on the more modern Lua language.

Only LTM configuration is migrated; other modules must be done manually. If a feature or functionality cannot be converted directly, the VMware engineer will work with the customer to provide a workaround or determine the best course of action.

Cutover

Once the configuration is migrated to the NSX Advanced Load Balancer deployment, it is time to begin testing. All virtual services are migrated and left in a disabled state so they do not cause an ARP conflict. There are various methods for testing the configuration prior to cutting over. Most often, the virtual service is given a new IP address and marked as enabled.

The virtual service is deployed onto a Service Engine and is available for testing. This can involve accessing the virtual service directly, by using an alternate name in DNS, or by altering a client host file. In this test scenario, SNAT is typically recommended to ensure no changes are required on the servers, yet traffic will synchronously return through the NSX Advanced Load Balancer's load balancers rather than the server’s default gateway.

Once the virtual service is ready to go live, the virtual service is disabled on the BIG-IP and enabled on NSX Advanced Load Balancer. If additional IP addresses are available, NSX Advanced Load Balancer can be configured to use a unique IP for the virtual service. The cutover is performed by changing the IP address advertised through DNS. Traffic will gracefully bleed off the BIG-IP as new traffic is processed through NSX Advanced Load Balancer with no disruption to live traffic.

This process can be repeated as necessary for all applications.

Testing NSX Advanced Load Balancer with Load Generators

10

Validating performance characteristics of Avi Vantage or any device under test (DUT) with load generation tools is a common practice. There are many load generators, and operating them is often more of an art than a science.

NSX Advanced Load Balancer has no specific preference for which load generators should be used. Our intent and goal is for any published HTTP performance numbers to be fully repeatable via the industry standard ApacheBench (AB), which is freely available for most Linux distributions.

NSX Advanced Load Balancer provides an Ubuntu virtual machine, downloadable from our Portal. This VM includes a default web server and AB, which can be used to quickly spin up clients and servers to test NSX Advanced Load Balancer load balancing and performance.

The most common issue when testing with a load generator is the client or server running out of capacity. When this happens, responses become slow, packets are dropped, and the load generator will report the device under test did not perform well. Load generators make no distinction in results when the bottleneck is the client, the device under test, or the server. If the performance doubles when adding a second load generator or a second server, then its a safe assumption that the Service Engine was not the bottleneck.

This section is not a definitive guide to load generators or their optimization. VMware strongly recommends working with one of our engineers to ensure best results.

Load Generators Versus Real Clients

Testing with a load generator is an important step in establishing trust with a load balancer. It is helpful to understand the behavior when a Service Engine will reach maximum capacity and the impact to connections or latency. However, load generators are not real clients, which behave very differently. For instance, clients connecting over a network will have some latency, while load generators often are configured for near zero latency via immediate adjacency to the device under test.

Free load generators generally are software that generate traffic and validate the timing and success of the responses. Commercial load generators, such as Ixia or Spirent provide both the clients and servers. With systems that include the servers, often the servers do not start until the test starts. Service Engines health checking the servers will find them down for a few seconds after clients begin sending requests.

Service Engine Optimization

Ensure the Service Engines to be tested have been configured with enough hardware resources to achieve the desired goals. This includes hardware such as CPU, memory, NICs etc. Every virtualized environment has different performance maximums, particularly in the packets per second able to be sent to a single VM. For instance, ESX 5.5 is about 500,000/s, ESX is about 950,000/s.

For more information on general guidelines on performance, see *Sizing Service Engines* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

NSX Advanced Load Balancer Configuration Recommendations

The following configuration changes should be made to NSX Advanced Load Balancer:

- Health monitors: Disable active and passive health monitors from within the pool. This ensures there is no lag between the servers being turned on, clients sending traffic, and NSX Advanced Load Balancer waiting for a successful health monitor response.
- Slow Ramp: Set the pool's Slow Ramp time to 0, effectively disabling the ramp. This ensures NSX Advanced Load Balancer sends connections as fast as it receives them. The assumption in a load test is the clients and servers will not be the bottleneck.
- Logs: Disable Non-Significant client logs if they have been enabled. It is never recommended to attempt to log hundreds of thousands of requests per second, as there is a performance penalty for this task. Significant logs (errors) will still be performed, which is the default state of a virtual service.

When deploying SEs in virtual environments such as ESX:

- Reserve CPU
- Reserve Memory

Maintaining NSX Advanced Load Balancer Controller Single Node Cluster Availability using VMware

11

It is recommended to deploy NSX Advanced Load Balancer Controller in a three-node cluster. Owing to the limited availability of resources, some businesses may not find this recommendation preferable. In such cases, NSX Advanced Load Balancer Controller can be deployed as a standalone node. In a standalone node deployment, the availability of NSX Advanced Load Balancer Controller is maintained by the underlying infrastructure rather than the clustering mechanism.

By deploying and hosting NSX Advanced Load Balancer Controller on VMware, VMware's native solutions can be utilized for maintaining high availability for NSX Advanced Load Balancer Controller. This section focuses on how a single node NSX Advanced Load Balancer Controller cluster is being hosted within VMware and how using native solutions we can maintain and restore Controller availability both proactively and reactively to an infrastructure impact.It focuses on how to use the VMware environment in conjunction with a deployed Controller. However, setting up the VMware environment is not within the scope of this section.

Note

- It is highly recommended to maintain up-to-date configuration archives. A single node cluster implies there is only one host maintaining the configuration. During disaster recovery, for example, after a storage failure, NSX Advanced Load Balancer Controller will have to be restored from the configuration archive.
- For more information on VMware HA/ DRS/ vMotion on Controller cluster and Service Engines, see *NSX Advanced Load Balancer High Availability in VMware vCenter Environment* topic in the VMware NSX Advanced Load Balancer*Installation Guide*.

VMware Solutions

VMware provides native mechanisms for maintaining availability of critical applications and virtual machines. vMotion and VMware High Availability (VMware HA) were tested, and any impact to availability of a single node NSX Advanced Load Balancer Controller was measured and data integrity validated.

vMotion

vMotion allows live migration of a virtual machine from one host to another with no downtime. This is a proactive approach to maintaining availability of a virtual machine's services, migrating virtual machines prior to server maintenance, or move virtual machines off a degraded or failing server.

For more information, see vMotion Overview and Best Practices for Configuring Resources for vMotion.

VMware High Availability (VMware HA)

VMware high availability provides failure protection for applications and virtual machines. In case of server failure, VMware HA allows virtual machines to be dynamically migrated and started up on another server restoring the application's availability.

For more information, see About VMware HA and Best practices for Configuring Resources for VMware.

Validation Scenarios

The following scenarios were tested for validating, restoring, and maintaining the availability of a single node NSX Advanced Load Balancer Controller running on VMware:

- vMotion live migration
- Server failure

Setup Specifications

The following setup specifications were used when validating the scenarios:

- VMware vSphere 6.0 update 3
- NSX Advanced Load Balancer 17.2.7
- NFS Shared Datastore

Use Cases

vMotion Live Migration

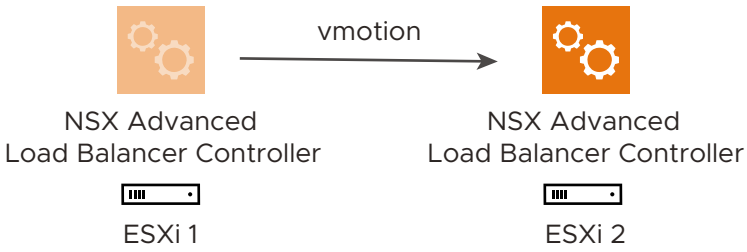
During the vMotion live migration, NSX Advanced Load Balancer Controller was migrated from one server to another, all the while being powered on.

Expectation

Availability maintained throughout with instances of increased API latency.

Result

During the process, NSX Advanced Load Balancer Controller remained available with intermittent occurrences of increased latency to the API. After migration, all services are restored and the Controller is back to functioning normally. There was no data path impact (load balanced applications) for the duration of the test scenario.

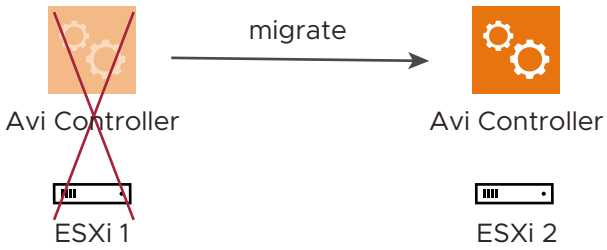


Server Failure

The ESXi server hosting the NSX Advanced Load Balancer Controller was made to fail.

Expectation

Unavailability of Controller for 3 – 5 minutes, with additional 2-3 minutes before the controller services are up and running. VMWare high availability restores the availability of the NSX Advanced Load Balancer Controller.



Result

On ESXi host failure, it took between 3-5 minutes for vCenter to detect the host failure, migrate the NSX Advanced Load Balancer Controller and power on.In the next 2-3 minutes, Controller services were up and the APIs were available.There was no data path impact (load balanced applications) for the duration of the test scenario.

NSX Advanced Load Balancer Deployment Recommendations

There are some best deployment recommendations to be considered when deploying using a single NSX Advanced Load Balancer Controller. These recommendations are applicable when the NSX Advanced Load Balancer Controller and Service Engines are hosted on the same VMware cluster. Owing to the potential fate sharing of the Service Engines and NSX Advanced Load Balancer Controller being hosted on the same cluster, both the control plane availability (NSX Advanced Load Balancer Controller) and the data plane (Service Engine) availability are taken into consideration. These recommendations may vary based on the:

For more information on configured SE group, see *Elastic HA for NSX Advanced Load Balancer Service Engines* topic in the VMware NSX Advanced Load Balancer *Installation Guide* or *Legacy HA for NSX Advanced Load Balancer Service Engines* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

For more information on VMware integration, see *Installing NSX Advanced Load Balancer in VMware vSphere Environments* topic in the VMware NSX Advanced Load Balancer *Installation Guide*.

Note These recommendations are applicable when the NSX Advanced Load Balancer Controller and NSX Advanced Load Balancer Service Engines are hosted on the same VMware server cluster.

Elastic HA - Write Access

When deploying Service Engines in an Elastic HA mode, it is recommended to host NSX Advanced Load Balancer Controller on different servers than the Service Engines. If the Service Engine and Controller are hosted on the same server, in a failure scenario, the virtual services hosted on that Service Engine will be impacted until the controller services are completely restored.

Note vMotion and DRS are not recommended for Service Engines.

Configurations in NSX Advanced Load Balancer

Modify the Service Engine group to specify which hosts/clusters the Service Engines can/can not be created on. By doing so, you can specify hosts that the Controller will not be hosted on. For more information, see *Performing an Include-Exclude Operation on a Cluster and Host* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Configurations in VMWare

To keep the NSX Advanced Load Balancer Controller separate from the Service Engines, we will be using VM Affinity rules within VMWare. VM Affinity rules are configured such that the controller runs on servers different from the servers hosting the Service Engines.

For more information, see VMware VM Affinity and VMware VM Affinity Setup.

Elastic HA - No Access

When deploying Service Engines in an elastic HA mode, it is recommended to host NSX Advanced Load Balancer Controller on different servers than the Service Engines. If the Service Engine and NSX Advanced Load Balancer Controller are hosted on the same server, then in a failure scenario, the virtual services hosted on that Service Engine will be impacted until the controller services are completely restored.

Note vMotion and DRS are not recommended for Service Engines.

Configurations in NSX Advanced Load Balancer

None required.

Configurations in VMWare

To keep the NSX Advanced Load Balancer Controller separate from the Service Engines, we will be using VM Affinity rules within VMWare. VM Affinity rules are configured such that the controller runs on servers different from the servers hosting the Service Engines. We will use one rule for defining which hosts are eligible for the controller and another for servers for which the Service Engines are eligible for. By doing so, you can specify different hosts for NSX Advanced Load Balancer Controller than those that are specified for the Service Engines. These rules should keep the Service Engines separated from the NSX Advanced Load Balancer Controllers, thus reducing the risk to data plane traffic in the event of a host failure.

For more information, see VMware VM Affinity and VMware VM Affinity Setup.

Legacy HA - write access

When deploying Service Engines in the Legacy HA mode, it is recommended that NSX Advanced Load Balancer Controller is hosted on different servers than the Service Engines. If the primary Service Engine and Controller are hosted on the same server, in a failure scenario, the virtual services hosted on that Service Engine will be impacted until the controller services are completely restored.

Note vMotion and DRS are not recommended for Service Engines.

Configurations in NSX Advanced Load Balancer

Modify the Service Engine group to specify which hosts or clusters in which the Service Engines can/cannot be created on. By doing so, you can specify hosts that NSX Advanced Load Balancer Controller will not be hosted on.

For more information, see *Performing an Include-Exclude Operation on a Cluster and Host* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Configurations in NSX Advanced Load Balancer

To keep the NSX Advanced Load Balancer Controller separate from the Service Engines, we will be using VM Affinity rules within VMWare. VM Affinity rules are configured such that NSX Advanced Load Balancer Controller runs on servers different from the servers hosting the Service Engines.

For more information, see VMware VM Affinity and VMware VM Affinity Setup.

Legacy HA - No Access

When deploying Service Engines in an elastic HA mode, it is recommended to host NSX Advanced Load Balancer Controller on different servers than the Service Engines. If the Service Engine and NSX Advanced Load Balancer Controller are hosted on the same server, in a failure scenario, the virtual services hosted on that Service Engine will be impacted until the NSX Advanced Load Balancer Controller services are completely restored.

Note vMotion and DRS are not recommended for Service Engines.

If the server resources are not sufficient to keep the NSX Advanced Load Balancer Controller separate from all the Service Engines, it is recommended to not select **Distribute Load** within the Legacy HA configuration. This will result in only a single Service Engine being primary for all virtual services.

Configurations in NSX Advanced Load Balancer

If there are not enough compute resources available preventing the controller from being hosted separately from all Service Engines, do not configure **Distribute Load** within the Legacy HA setup.

For more information, see *Legacy HA for NSX Advanced Load Balancer Service Engines* topic in the VMware NSX Advanced Load Balancer *Configuration Guide*.

Configuration in VMware

To host the NSX Advanced Load Balancer Controller separately we will be using the VM Affinity rules within VMware. We will use one rule for defining which hosts are eligible for the controller and another for servers for which the Service Engines are eligible for. By doing so, you specify different hosts for the controller than those specified for the Service Engines. These rules should keep the Service Engines separated from the Controllers, thus reducing the risk to data plane traffic in the event of a host failure.

If there are not enough server resources available to keep NSX Advanced Load Balancer Controller hosted separately from all the Service Engines, setup the Affinity rules to keep the controller separate from the primary Service Engine. This will prevent impact to the data plane traffic in the event of the server hosting NSX Advanced Load Balancer Controller fails.

For more information, see VMware VM Affinity and VMware VM Affinity Setup.

In conclusion, VMware's native solutions can be relied upon to provide High Availability for NSX Advanced Load Balancer Controller when a business decides that a single node Controller suits them the best.

NSX Advanced Load Balancer SaaS

12

The NSX Advanced Load Balancer SaaS complements the flagship NSX Advanced Load Balancer Platform, as the cloud-hosted option to deliver application services including distributed load balancing, web application firewall, global server load balancing (GSLB), network and application performance management across a multi-cloud environment. It helps ensure fast time-to-value, operational simplicity, and deployment flexibility in a highly secure manner.

NSX Advanced Load Balancer Architecture

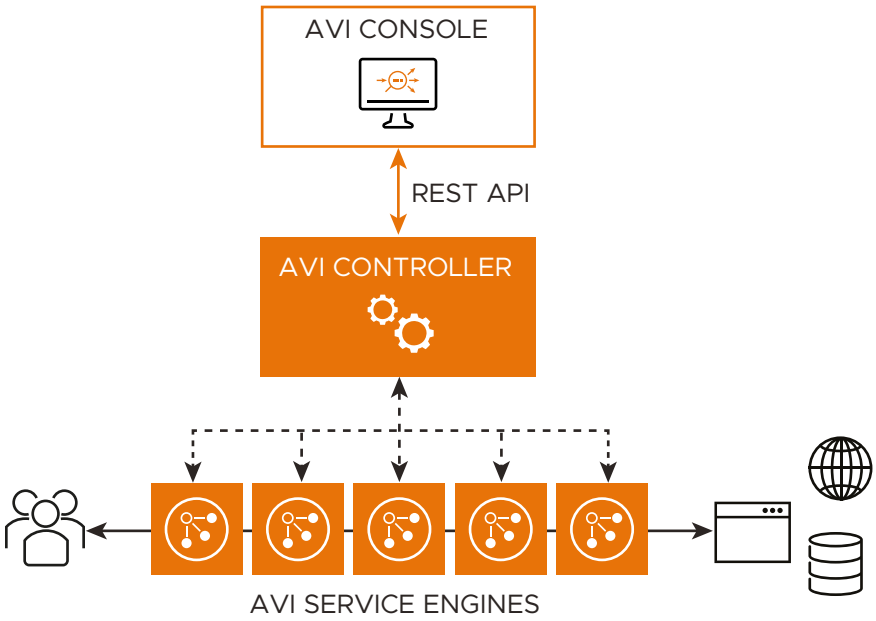
The NSX Advanced Load Balancer architecture comprises two key components:

- The Controller

  The NSX Advanced Load Balancer Controller is the control plane for the NSX Advanced Load Balancer solution. It provides a single point of management and operations, including configuration, visibility, analytics, and metrics. It uses big data analytics to analyze the data and present actionable insights to administrators on intuitive dashboards.

- The Service Engines

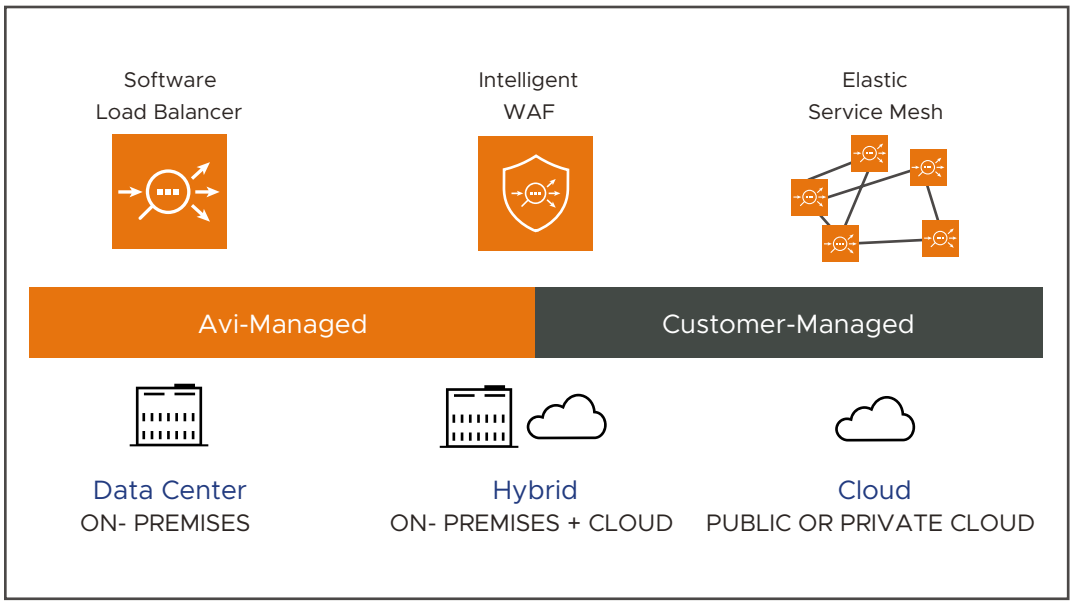
  The NSX Advanced Load Balancer Service Engines (SEs) are the data-plane component. SEs collect real-time application analytics from traffic flows between end users and applications. The SEs are high-performance data-plane entities which perform the actual load balancing (LB) and web application firewall (WAF) functions. All configurations for a given SE are managed by the respective Controller Cluster.



For more information on the NSX Advanced Load Balancer Architecture, see the *Load Balancing* topic in the *VMware NSX Advanced Load Balancer Configuration Guide*.

Introduction to NSX Advanced Load Balancer SaaS

Prior to the introduction of NSX Advanced Load Balancer SaaS, Controllers and Service Engines would be deployed in the customer environment. With the advent of NSX Advanced Load Balancer SaaS, an additional operational model is offered, control plane as a service. With NSX Advanced Load Balancer SaaS, the NSX Advanced Load Balancer hosts and operates the control plane. Service Engines (the data plane) continue to reside in the customer environment (on-premises or in a public cloud). The following diagram depicts a Controller managed by NSX Advanced Load Balancer.



The salient features are listed below:

- NSX Advanced Load Balancer-managed control plane, available as a service.
- Instantaneous on-boarding: NSX Advanced Load Balancer Controller deployed in minutes.
- The NSX Advanced Load Balancer provides continuous monitoring, configuration backups, software upgrades and patches, security alerts, and performance tune-ups
- Proactive support
- Multi-cloud solutions — data plane can be on-premises or in a public cloud.

Requirements for Connectivity to NSX Advanced Load Balancer SaaS

The following considerations are important while connecting your environment to NSX Advanced Load Balancer SaaS.

Case 1: No additional connectivity requirements from NSX Advanced Load Balancer SaaS to customer environment

In this scenario, NSX Advanced Load Balancer SaaS communicates with public API endpoints for respective Infrastructure as a service (IaaS) cloud infrastructure where the customer workloads are provisioned. This works automatically, with no additional configuration or access policy requirement. The NSX Advanced Load Balancer SaaS supports the following cloud types in this mode:

- Amazon Web Services
- Microsoft Azure
- OpenStack with public URLs (for example, Platform 9)

In addition, the NSX Advanced Load Balancer SaaS can work with no-orchestrator clouds. In this case, SE provisioning is done by the customer manually (or with the help of automation tools such as Ansible or Terraform). NSX Advanced Load Balancer SaaS does not require inbound access to the customer virtualization infrastructure. The cloud types supported in this mode are:

- Linux Server Cloud
- No-orchestrator clouds (for example, VMware read access)

Case 2: NSX Advanced Load Balancer SaaS requires connectivity to customer's orchestration environment

In this scenario, the NSX Advanced Load Balancer SaaS communicates with the infrastructure typically residing in a private network. Once NSX Advanced Load Balancer SaaS is allowed to communicate with the virtualization infrastructure, it can communicate continuously with the virtualization orchestrator and provide fully automated provisioning. The NSX Advanced Load Balancer SaaS supports the following cloud types in this mode:

- VMware write access
- OpenStack hosted on-premises
- VMware read and write access

Note

- In both Case 1 and 2 above, the Service Engines need to communicate with NSX Advanced Load Balancer SaaS. This might require allowing communication between the NSX Advanced Load Balancer Service Engines and the Controller IPs, over secure communication channels such as HTTPS and SSH.
- Linux Server Cloud can function in either of the modes above.

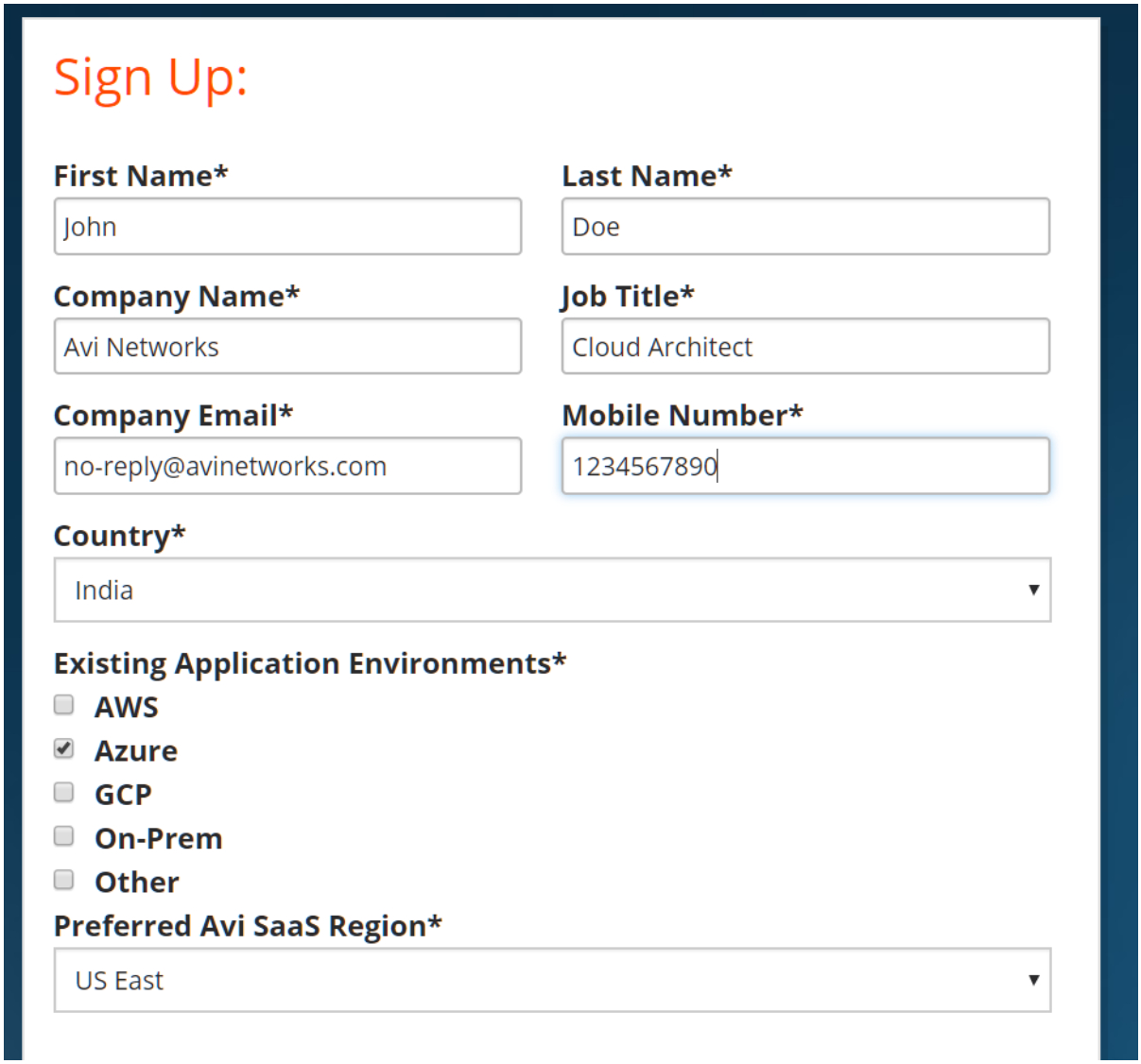
Deploying NSX Advanced Load Balancer SaaS

This section lists the process for deploying the NSX Advanced Load Balancer Platform with the NSX Advanced Load Balancer SaaS operational model. The full deployment process is divided into three steps:

- Registering for NSX Advanced Load Balancer SaaS
- Connecting NSX Advanced Load Balancer SaaS to your cloud
- Configuring application delivery features

Registering for NSX Advanced Load Balancer SaaS

1 To sign up for NSX Advanced Load Balancer SaaS, navigate to https://info.avinetworks.com/saas and register.



2 An NSX Advanced Load Balancer representative will contact you and provide a URL through which to connect. Log in to create an NSX Advanced Load Balancer account.

   Once the previous step completed, you will be provided with a login ID and the URL with which you can access NSX Advanced Load Balancer SaaS.

Connecting Your Cloud to NSX Advanced Load Balancer SaaS

In this section, NSX Advanced Load Balancer SaaS is configured to integrate with the network infrastructure where your applications are hosted (and where the NSX Advanced Load Balancer Service Engines will be provisioned).

Depending on the cloud type, please refer to the respective document mentioned below:

Cloud Type	Documentation
Microsoft Azure	See *Installing NSX Advanced Load Balancer in Microsoft Azure* topic in the *VMware NSX Advanced Load BalancerInstallation Guide*.
Amazon Web Service	See *Installing NSX Advanced Load Balancer in Amazon Web Services* topic in the *VMware NSX Advanced Load BalancerInstallation Guide*.
Linux server cloud	See *Considerations for Deploying NSX Advanced Load Balancer in Linux Server Cloud* topic in the *VMware NSX Advanced Load BalancerInstallation Guide*.

Configuring Application Delivery Features

Once the cloud is integrated with NSX Advanced Load Balancer SaaS, you can get started with creating your first virtual service. For more information, see *Create a Virtual Service* topic in the *VMware NSX Advanced Load Balancer Configuration Guide*.

Read the following topics next:

- Configuring NSX Advanced Load Balancer SaaS Controller Using Service Account
- Accessing the NSX Advanced Load Balancer SaaS REST API
- Accessing the NSX Advanced Load Balancer SaaS CLI Using SSH

Configuring NSX Advanced Load Balancer SaaS Controller Using Service Account

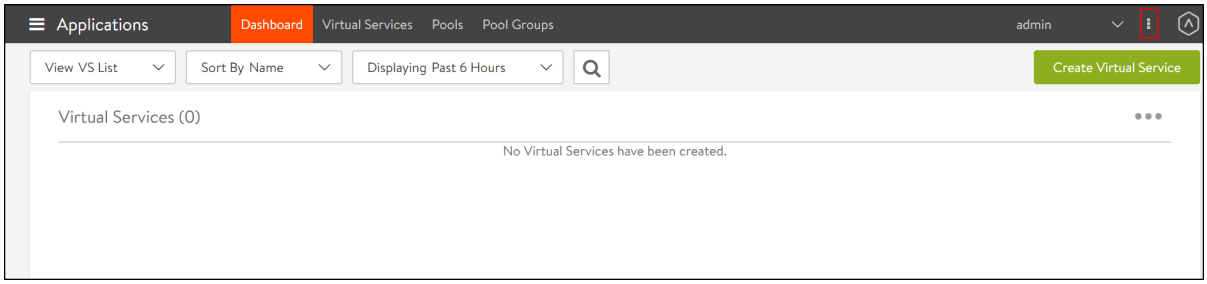
The NSX Advanced Load Balancer supports single sign-on (SSO) to the NSX Advanced Load Balancer Controller UI using Security Assertion Markup Language (SAML). However, during debugging or even normal day-to-day operations, there is often a need to access the Controller's CLI using SSH. SAML credentials cannot be used to login to the CLI.

To access the Controller through SSH, a registered user must have a valid token. Once a token has been created, one can initiate an SSH connection to the Controller using cli as the SSH user. A CLI shell will be created. Once the shell has been created, a login prompt will be presented. Provide the required username and the token as the password. This topic explains the process needed to configure a service account for use on an NSX Advanced Load Balancer SaaS Controller.

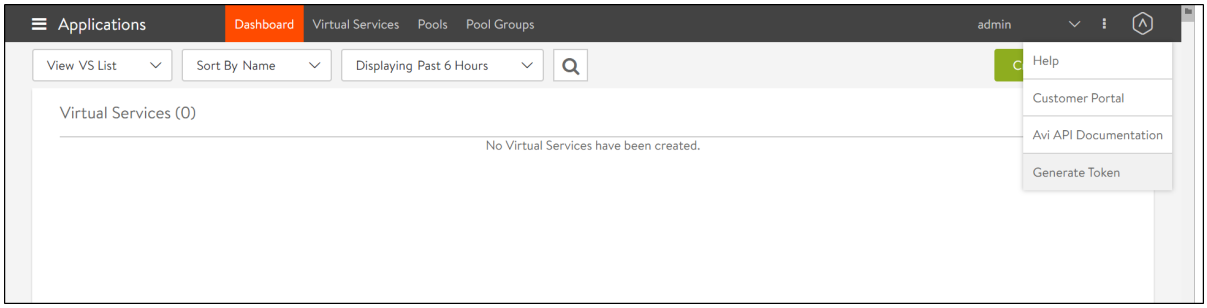
Generate the Authorization Token

1 Log in to the NSX Advanced Load Balancer UI.

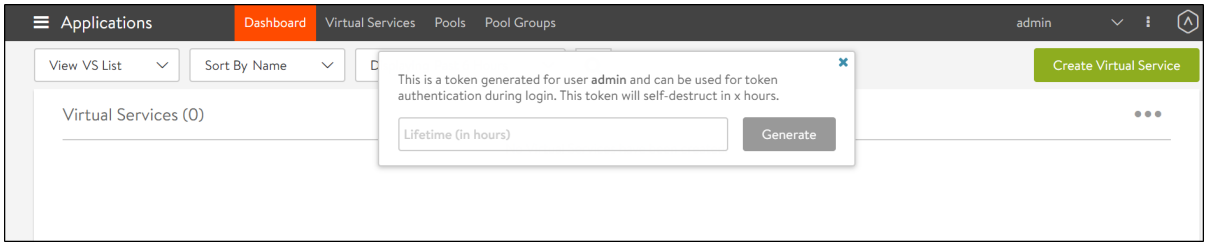
2 Click the three dots in the dashboard.



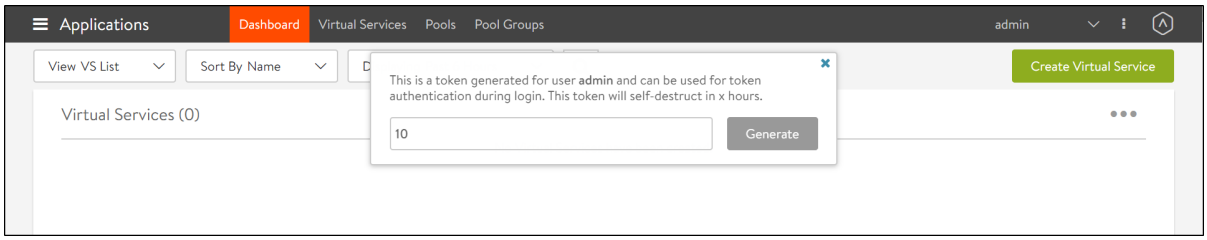
3 Click **Generate Token**.



A pop-up screen appears as shown below:



4 Enter the Lifetime for the token's validity in hours.



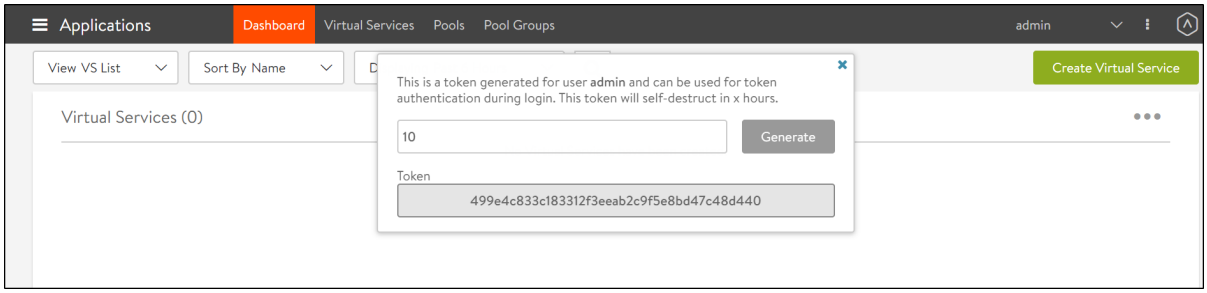
Note

a To generate a single use token, enter **0**.

b The maximum value that can be entered in this field is **87600** hours.

c In case another token is generated before the first one expires, the first token still remains valid.

5 Click **Generate**. The token is generated for your Service Account.



6 Save this token for your automation or API usage.

7 To test your credentials use the following Python code using the Requests library.

```
import requests
import urllib3

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

data = {
  'username': '<service account name>',
  'password': '<your token that was generated in step 5>
}
login = requests.post('https://<your controller name>.saas.avinetworks.com/login',
verify=False, data=data)
print(login.status_code)
```

The status code `200` is returned for a successful query, and the status code `401` is returned for the failed query.

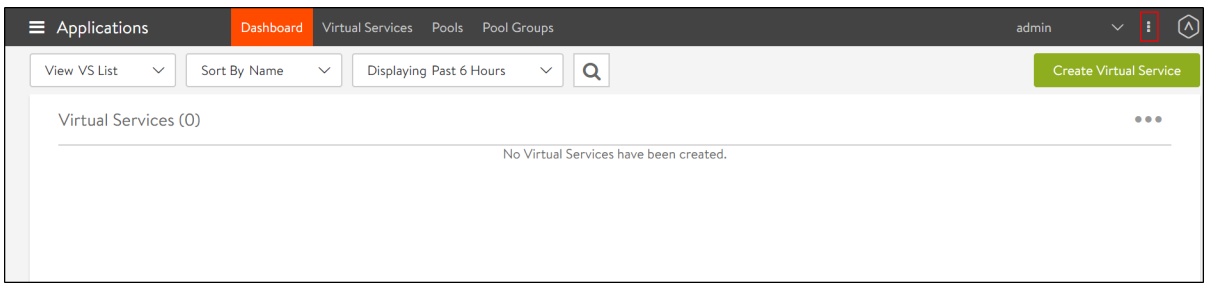
Accessing the NSX Advanced Load Balancer SaaS REST API

To access an NSX Advanced Load Balancer SaaS instance through the REST API, a registered user must have a valid token. Once that token has been created, the user can authenticate to the NSX Advanced Load Balancer Controller REST API using the token as the password.

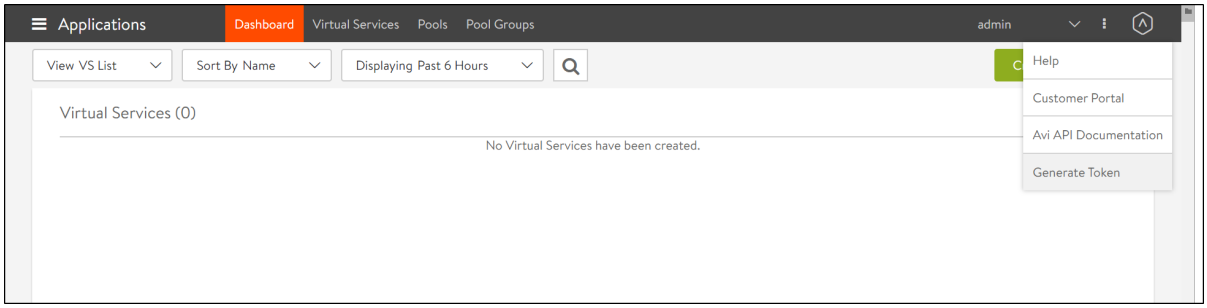
Generate the Authorization Token

1 Login to the NSX Advanced Load Balancer UI.

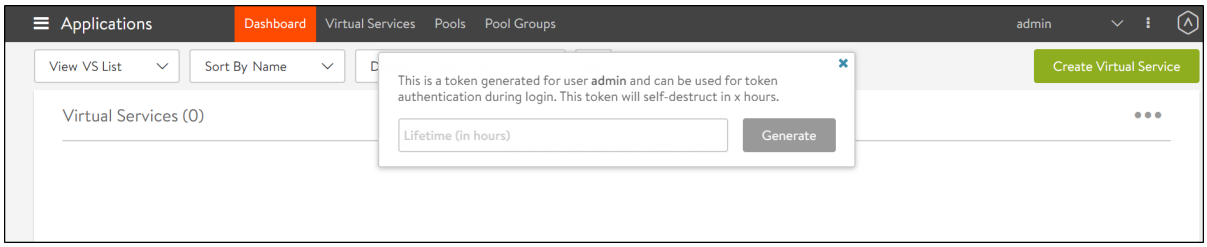
2 Click the three dots in the dashboard.



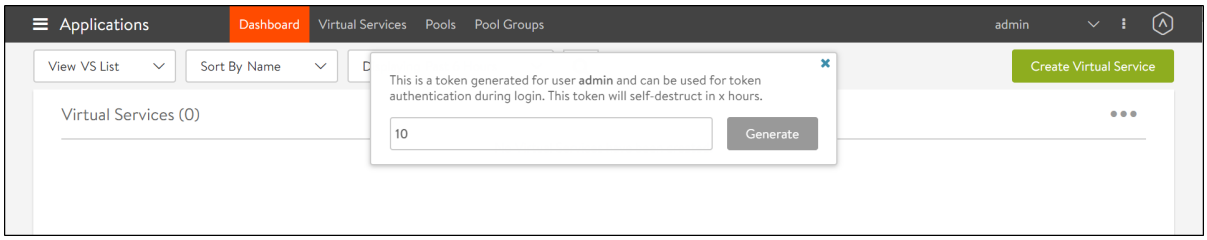
3 Click **Generate Token**.



A pop-up screen appears as shown below:



4 Enter the Lifetime for the token's validity in hours.



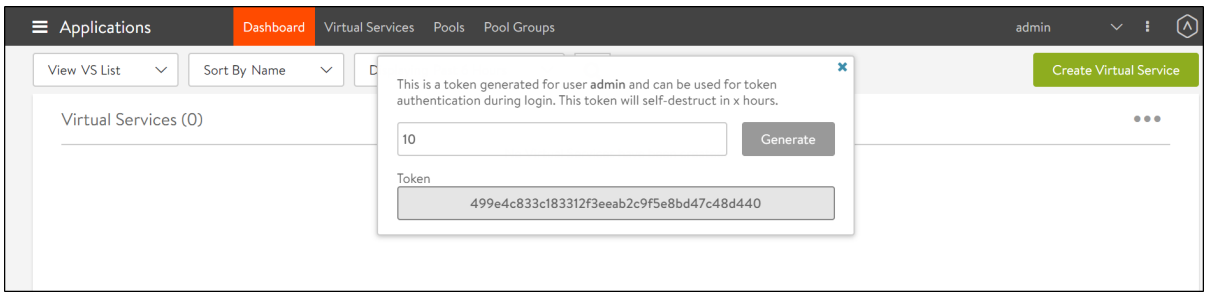
Note

a To generate a single use token, enter **0**.

b The maximum value that can be entered in this field is **87600** hours.

c In case another token is generated before the first one expires, the first token still remains valid.

5 Click **Generate**. The token is generated and displayed as shown below:



6 Copy the token.

Login to NSX Advanced Load Balancer SaaS REST API

As shown in the SDK example below, access the Controller API using the token as your password.

```
from avi.sdk.avi_api import ApiSession

api = ApiSession.get_session("avikb.saas.avinetworks.com", "admin@avinetworks.com",
"499e4c833c183312f3eeab2c9f5e8bd47c48d440", tenant="kb")

#----- retrieve virtualservices
resp = api.get('virtualservice')
```

Accessing the NSX Advanced Load Balancer SaaS CLI Using SSH

In an NSX Advanced Load Balancer SaaS environment, SAML authenticates the user to the NSX Advanced Load Balancer Controller UI. However, during debugging or even normal day-to-day operations, it can be useful to access the Controller's CLI using SSH instead.

For more information on Accessing the NSX Advanced Load Balancer SaaS CLI, see *NSX Advanced Load Balancer CLI Access to a SAML-Authenticated NSX Advanced Load Balancer Controller* topic in the *VMware NSX Advanced Load BalancerAdministration Guide*.