Active System Manager Integration for VMware vRealize Orchestrator

User's Guide

Notes, cautions, and warnings

NOTE: A NOTE indicates important information that helps you make better use of your product.

CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

2016 - 09

Rev. A01

Contents

1 Overview 5

2 About The Document 6

3 Executive Summary 7

4 What's new in the ASM vRO Plugin 2.0 8

5 Installing the ASM vRO Plugin for Integration 9
- Installation Prerequisites 9
- Installation Procedure 9

6 Validating the Installation 10
- Validating Package Content 10
  - Workflows 10
  - Actions 11
  - Inventory 11
- Validate the ASM vRO Plugin API 11

7 Upgrading ASM vRO 12

8 API for ASM vRO Integration 13
- API Types 13
- API Objects 14
  - ASMAppliance 14
  - ASMApplianceManager 16
  - ASM Service 17
  - ASMServiceComponent 17
  - ASMTemplate 18

9 Workflow and Actions 19
- Workflow 19
  - Configuration Workflows 19
  - Sample Workflows 19
- Actions 21
  - com.dell.library.asm 21
  - com.dell.library.asm.configuration 22

10 Using vRealize Orchestrator 23
- Adding a New ASM Appliance Object to vRO Inventory 23
- Deploy a New Service to vRO 23
- Add a Server to an Existing Service 24
- Teardown the Service 24

11 Integrating with vRealize Automation.....25
Prerequisite for vRA Integration.....25
Non-Interactive Workflow as vRA XaaS Blueprints.....25
Interactive Workflow as vRA XaaS Blueprints.....26
Workflow Output.....27
Upgrading the vRA XaaS Blueprint.....27

12 Troubleshooting.....28
vRO Exceptions.....28
vRO Application Server Log.....28
ASM Troubleshooting.....28

13 Uninstalling ASM vRO Integration.....30

14 Known Issues and Forewarning.....31

15 Compatibility Matrix.....32

1

Overview

The Dell Active System Manager integration for vRealize Orchestrator (vRO), formerly known as vCenter Orchestrator (vCO), enables key automation functions, which can be embedded into vRealize Orchestrator workflow. The workflow helps in provisioning infrastructure, capacity scaling, remediation actions, and service teardown. These features help IT teams to broaden their business solutions, improve responsiveness, and simplify complex IT processes.

The integration is developed using the VMware vRO SDK and Active System Manager's newly published REST APIs. The ASM Integration for vRO are given to ASM customers to install it into their vRealize Orchestrator environment. After installing the integration, customers can use the workflows packaged with the plugin or they can create their own workflows using the ASM attributes implemented in the creation of the plugin. IT administrators use the two main management products for this integration:

- VMware's vCenter Server
- VMware vRealize Automation Center

ASM integration for vRealize Orchestrator also enables the vRealize Automation (vRA) to integrate with VMware vRealize Orchestrator (vRO).

VMware vRealize Orchestrator (vRO) allows:

- Any template in ASM, whether infrastructure or workload, to be published as a service in vCAC (vCloud Automation Center) portal.
- Enables any ASM function to be called from vCO for broader orchestration.

vRealize Automation (vRA, formerly known as vCloud Automation Center) enables:

- Consumption portal and service catalog
- Broad workflow and process integration
- Aggregation of **anything as a service**, including any ASM-based template or ASM action such as scaling and remediation

This integration enables unique services not supported by VMware, physical, hybrid physical-virtual, multi-hypervisor, infrastructure scaling, and remediation. The ASM vRO Integration supports vCO and vRO versions 5.5 and 6.0.

2

About The Document

This documentation provides information regarding:

- Installing ASM vRO integration
- Validating the installation
- Upgrading ASM vRO
- Configuring ASM vRO API
- Workflow and Action
- Using vRealize Orchestrator
- Integrating with vRealize Automation
- Troubleshooting
- Uninstalling ASM vRO Integration

3

Executive Summary

ASM is a service-based management console that automates end-to-end template-based provisioning and lifecycle management of infrastructure and workloads in today's modern data center. ASM and its REST base API, enables end-to-end integration and automation with additional tools and management consoles in the data center. Active System Manager integration of VMware vRealize Orchestrator, automates customers, cloud and integrate VMware management systems with Active System Manager. Workflow packages are developed using the Orchestrator to achieve step-by-step process automation for greater flexibility and automation across physical and virtual infrastructure and multiple management products.

4

What’s new in the ASM vRO Plugin 2.0

In the previous releases, 1.0 ~ 1.1, the update was required for the support of ASM release 8.1.1 from release 8.1.0. This version has been implemented to support the ASM release 8.2.1 and new vRO version. Noticeably, the API objects are replaced and updated. The model objects provide more detail about the mirroring ASM objects than previous versions. The API Explorer has updated method signature and descriptions for commonly used objects and methods. Also, caching has been removed in this version, and the inventory data is handled by the vRO database. This means that AsmObjectNotFound exception will rarely be an issue if at all. Minor changes include the workflow icon update, version details, and so on.

5

Installing the ASM vRO Plugin for Integration

Before installing ASM vRO integration, ensure that you have met the installation prerequisites:

Installation Prerequisites

To install ASM vRO Integration, vRO environment must meet the following criteria before installing the Integration:

- The ASM vRO plugin 2.0 requires vRO 7.0.
- The vRO server must be able to communicate to the ASM appliance servers through HTTPS protocol (ports 443 and 8281).
- The vRO server must be in a clean state. Either the ASM vRO Integration has never been installed before or it should be removed completely. To ensure that the environment is clean, see the chapter **Uninstalling ASM vRO Integration**.
- The vRO control center must be running. if the control center is not running, run the command, `/etc/init.d/vco-configurator start`, on the vRO server to start the service.
- The workstation must be able to reach the vRO configuration server, and the o11nplugin-asm.vmoapp must be downloaded to the workstation.

NOTE: To avoid data loss, make sure that there is no workflow running in the vRO during the installation procedure, as restart is required.

Installation Procedure

To install ASM vRO Integration:

1. Open control center on the browser. Usually, it is https://<VRO_HOSTNAME>:8283/vco-controlcenter/ or https://<vRA_HOSTNAME>:8283/vco-controlcenter/ depending on how the vRO has been set up. If the vRO is bundled in vRA appliance, use the vRA_HOSTNAME. If these URL's do not work, contact the IT administrator to verify the system setup. Alternatively, access the system setup page(https://<vRA_HOSTNAME>:5480) to verify the required port information.
2. In **VMware vRealize Orchestrator** window, log in as a user with administrative privileges (For example, "root" user) to the vRO control center. Click **Plug-ins** in the left pane of the **VMware vRealize Orchestrator configuration** page.
3. Find the "Install new plug-in" in the Plug-ins page.
4. Click on the find button (magnifying glass icon or "browse" box), locate and select the vmoapp file in the pop-out window, and click **Upload and install**.
5. Read and accept the EULA in the following page.
6. The Plug-ins page should display, "Dell Active System Manager (2.0.0 build X) New plug-in installed" (where 'X' is the actual build number) message after a successful installation. It will also alert that the vRO has to be restarted. Click the link provided on the page to the **Startup Options** page, or on the **VMware vRealize Orchestrator Configuration** page, in the left pane, click **Startup Options**..
7. Restart the vRO application server by clicking **Restart Service**. The page will update the vRO application server status. Even though the page shows that server is RUNNING, it will take a few seconds afterward to enable the service completely.
8. Now the installation is complete.

6

Validating the Installation

To validate the installation, make sure:

- The package contents are imported.
- ASM API/SDK is available.
- New Inventory object is created.

Validating Package Content

1. When the ASM vRO Integration is successfully installed, new contents should appear in vRO. To inspect these, launch the vRO client, and select the **Design** view.
2. You see six tabs in the left panel to inspect different types of vRO contents. Each tab selects its corresponding view (workflows, actions, resources, configurations, packages, or inventory). First select the **Packages** tab. It shows the list of existing packages and additional information about each package.
3. Locate the package, *com.dell.plugin.asm*. Now click the **Workflow** tab in the right pane of the vRO client. It should show 21 workflows that are imported through the plugin installation.
4. Click the **Actions** tab in the right panel, you can see action elements are listed.

   NOTE: The workflows and the actions show their version as 2.0.0.
5. Finally to check whether the ASM API is associated with these elements or not, click the **Used Plug-ins** tab in the right pane. It should list the ASM API with its name and version information.

Workflows

The ASM vRO workflows are organized under the Dell Active System Manager folder in the Library folder. Select the **Workflows** tab in the left panel of the vRO client. Expand the root folder, Library folder, Dell Active System Manager folder, and its subfolders to inspect the workflows:

- Root Folder
  - Library
    - Dell Active System Manager
      - Configuration Workflows
        - Add ASM Appliance
        - Remove ASM Appliance
      - Sample Workflows
        - Asynchronous
          - Deploy Service (Async)

    - Scaledown Service (Async)
    - Scaledown Service — App (Async)
    - Scaledown Service — Cluster (Async)
    - Scaledown Service — Server (Async)
    - Scaledown Service — Storage (Async)
    - Scaledown Service — VM (Async)
    - Scaleup Service — Server (Async)
    - Scaleup Service — Storage (Async)
    - Teardown Service — All (Async)
    - Teardown Service — Selective (Async)
  - Noninteractive
    - Deploy Service
    - Scaleup Service — Server
  - Synchronous
    - Deploy Service
    - Scaledown Service
    - Scaleup Service — Server
    - Scaleup Service — Storage
    - Teardown Service — All
    - Teardown Service Selectively

Actions

The actions are organized into two groups:

- **com.dell.library.asm**: The actions directly call the ASM plugin API methods.
- **com.dell.library.asm.configuration**: The actions help configuring the vRO resources and other helper actions used in the sample workflows.

To see these actions, click the **Actions** tab in the left panel of the vRO client. Verify that all 14 workflows are imported successfully.

Inventory

Make sure that a new Inventory object is created for the ASM vRO plugin. Click the **Inventory** tab in the left panel. Ensure that there is a root inventory object called **Dell Active System Manager**. This root inventory object can be expanded, but it should be empty.

Validate the ASM vRO Plugin API

Make sure that the ASM Integration API is available. One quick way to validate this is by finding the ASM API in the API Explorer of the vRO client.

1. In the top right corner of the vRO client, click the **Tools** tab and expand it.
2. Then click **API Explorer** from the drop-down menu.
   It brings up a window that shows the API Explorer, locate **ASM SDK** module from the explorer. The API Explorer can be helpful when users are scripting with the ASM vRO plugin API. The API Explorer shows Javadoc-style documentations for the types and scripting objects.

7

Upgrading ASM vRO

The following are the pre-requisites for upgrading ASM vRO:

- The ASM must be updated to release 8.2.1.
- The ASM requires vRO 7.0 instead of 5.5 or 6.0.

NOTE: When you uninstall the older version of the plugin, the workflows created using that version is also deleted. Hence, ensure that you recreate the workflows and update the vRA service blueprints with the new workflows after installing the newer version of the plugin.

8

API for ASM vRO Integration

The ASM vRO Integration includes a package with useful elements such as the actions, workflows, and inventory. The ASM vRO Integration is operated through the ASM plugin API. The API includes the SDK module, types, objects, and their methods. The types are used for representing the API objects into the inventory or finder view. The API objects contain attributes and methods that are useful for automating procedures which require ASM operations. For more information, see the topic **Workflows and Actions**. You can also see the description of the types, objects, attributes, and methods from the API Explorer.

NOTE: The types and objects that existed in the plugin versions 1.0 and 1.1 are modified, replaced, or removed in version 2.0.

API Types

The API types provide mapping of the API objects to the finder objects. In scripting, API types are also used to define the type of required input, output, and attribute.

These following types are removed from the plug in 2.0 because the vRO 7.0 SDK supports the custom module to relate, find, and categorize the types and objects without requiring placeholder type/objects: *ASM: ASM_PLUGIN, ASM: ServiceFolder*, and *ASM: TemplateFolder*.

ASM: Appliance:

- Scripting object: `ASMAppliance`
- This type is mapped to `ASMAppliance`, which is responsible for storing ASM appliance configurations and making REST API calls. The `FinderResult` object of this type is a child object of the ASM: root object. It also contains the child objects of the *ASM: Service* Service type and the *ASM: Template* type. For more information, see the topic **ASMAppliance**.

ASM: Service:

- Scripting object: `Service`
- This type is mapped to `Service` class. This object is a child object of the `ASM: Appliance` type object in the finder. It also contains child objects of the `ASM: ServiceComponent` type. For more information, see the topic **ASMService**.

ASM: ServiceComponent:

- Scripting object: `ServiceComponent`
- This type is mapped to `ServiceComponent`. The object of this type has the `ASM: Service` type object as the parent in finder. For more information, see the topic **ServiceComponent**.

ASM: Template:

- Scripting object: `Template`
- This type is mapped to `Template`. In finder view, the object of this type is structured under the *ASM: Appliance* type object. For more information, see the topic **Template**.

API Objects

There are eight API objects that are included in the ASM SDK module. This section covers each object and the associated attributes and methods.

NOTE: Not all API objects have attributes and/or methods.

ASMAppliance

This object establishes HTTPS connection to the ASM REST webservice to make the requests. This is the most crucial object for the Integration.

Attributes:

- **displayName (String)**: String value that represents the display name of the ASM Appliance
- **services (Array of ASM: Service objects)**: Array of `ASM: Service` objects that are children of the `ASM: Appliance` object.
- **templates (Array of ASM: Template objects)**: Array of `ASM: Template` objects that are children of the `ASM: Appliance` object.

Methods:

- **addDuplicateServer**: This method clones server from the configuration of an existing server component. It duplicates servers which have the same relations to other resources (For example, duplicated servers are added to the cluster where original server belongs to). Server pool must be able to allocate physical server resources for the chosen configuration. Returned object is the *Service* with updated components.
  - deployment — Target `Service` where new server components are added.
  - targetServer –`ServiceComponent` of SERVER type that provides the configuration for the duplicate servers.
  - newServerName — Name of the new servers; if more than one instance of servers is requested, sequence number is appended to the name.
  - numInstances — Specifies the number of server instances that are added to the Service.
- **addDuplicateStorage**: This method clones storage from the configuration of an existing storage component. Duplicated storages have the same relations to other resources (For example, duplicated storages are used by the servers that the original storage support). Target volume must be able to allocate resources for the chosen configuration. Returned object is the *Service* with updated components. The target storage component must be a new storage that was created through the ASM (For example, a new volume in EqualLogic was created by the ASM during the deployment or update).
  - deployment — Target Service where new server components are added.
  - targetStorage — ServiceComponent of STORAGE type that provides the configuration for the duplicate storages.
  - newStorageName — Name of the new storages; if more than one instance of storages is requested, sequence number is appended to the name.
  - numInstances — Specifies the number of storage instances that are added to the Service.
- **deployTemplate**: Deploy a new Service with the configuration of the given `Template`.

  NOTE: The target `Template` in ASM must autogenerate the required fields (For example, the host name of the servers) other than the deployment name and description. Returned *Service* object describes the ASM service that has been deployed.

  - template — Target `Template` that is to provide the configuration for the new `Service` being deployed.
  - deploymentName — Name of the new `Service`.
  - description — Optional parameter that describes the deployed `Service`.

- **getService**: Retrieves the `Service` of the given name.
  - name — Name of the `Service` that should be retrieved.
- **getServiceById**: Retrieves the `Service` of the given id.
  - **deploymentId** — id of the `Service` that is to be fetched.
- **getServiceStatus**: Returns the status of the target Service (For example, in_progress, complete, error).

  NOTE: The returned status is no longer capitalized as in versions 1.0 –1.1.

  - vsoDeployment — Service that is to be checked for its status.
- **getServiceTemplateComponentById**: Retrieves a ServiceComponent object that has the matching id from the given Service object.
  - **deployment** — the target Service object.
  - **componentId** — the component id.
- **getServiceTemplateComponents**: Retrieves a list of `ServiceComponent` objects from the given `Service`.
  - deployment — The `Service` to retrieve the `ServiceComponent` objects.
- **getServiceTemplateComponentsByType**: Retrieves a list of `ServiceComponent` objects, but the result is filtered by the selected resource types.
  - deployment — The `Service` to retrieve the `ServiceComponent` objects.
  - storage — Boolean value indicating that the returned list of `ServiceComponents` should include STORAGE type components if available.
  - server — Boolean value indicating that the returned list of `ServiceComponents` should include SERVER type components if available.
  - cluster — Boolean value indicating that the returned list of `ServiceComponents` should include CLUSTER type components if available.
  - vm — Boolean value indicating that the returned list of `ServiceComponents` should include VIRTUALMACHINE type components if available.
  - app — Boolean value indicating that the returned list of `ServiceComponents` should include SERVICE (application) type components if available.
- **getServices**: Returns a list of all the Services in the ASM server.

  This method is deprecated, and the method description does not show in the API Explorer. Use the attribute, services, instead.
- **getTemplateById**: This method takes the Template id and fetches the Template object of the matching id from the Appliance.
  - **templateId** — ID of the target Template.
- **getTemplates**: Returns a list of all the published Templates in the ASM server.

  This method is deprecated, and the method description does not appear on the API Explorer. Use the attribute, templates, instead.
- **scaleDownService**: This method removes the resources of the selected types from the chosen `Service`. If a resource type is selected and no resources of that type exists, it will simply not affect the overall operation of the scale down (For example, scaling down VIRTUALMACHINE type resources in a *Service* without any will not affect the *Service* after the operation has been completed). Returned object is the *Service* with updated components list.
  - deployment — The `Service` where the resources of the selected types are to be removed.
  - storage — Boolean value indicating that STORAGE type components are to be removed.
  - server — Boolean value indicating that SERVER type components are to be removed.
  - cluster — Boolean value indicating that CLUSTER type components are to be removed.

- vm — Boolean value indicating that VIRTUALMACHINE type components are to be removed.
- app — Boolean value indicating that SERVICE (application) type components are to be removed.

- **scaleDownServiceComponent**: Removes the target `ServiceComponent` object from the `Service` and returns its resources to the resource pool. Returned object is the *Service* with updated components list.
  - deployment — The `Service` to be down scaled.
  - component — The `ServiceComponent` object to be removed from the `Service`.
- **serviceExists**: Checks if the `Service` of the given name exists or not. This is useful for workflow that wait until the service is deleted.
  - serviceName — Name of the *Service* that is checked for its existence.
- **tearDownService**: This method deletes the Service and returns all the resources to the resource pools. It returns the *Service* object describing the ASM service that is to be deleted from this call.

  NOTE: If any of the `ServiceComponents` objects, in the target `Service`, is shared, the request fails during the validation (For example, there is a `Service` that defines a cluster and supporting servers and storages. There is another `Service` that relies on this cluster to provision VM's). The ASM server (8.2.0 and newer releases) does not allow the teardown of any shared component.
  - deployment — Target *Service* to be removed; its resources are returned to the resource pools by default.
- **tearDownServiceSelective**: This method deletes the `Service` and returns the selected resources to the resource pools. This method can be helpful when there are shared resources across `Services`. It returns the `Service` object describing the ASM service that is to be deleted from this call.

  NOTE: The sample workflows that use this method forces the teardown of SERVER and VIRTUALMACHINE types (the same behavior is expected from the ASM UI).
  - deployment — Target `Service` to be removed; its resources are returned to the resource pools only if they are of the selected types
  - storage — Boolean value indicating to remove resources of the STORAGE type.
  - server — Boolean value indicating to remove resources of the SERVER type.
  - cluster — Boolean value indicating to remove resources of the CLUSTER type.
  - vm — Boolean value indicating to remove resources of the VIRTUALMACHINE type.
  - app — Boolean value indicating to remove resources of the SERVICE (application) type.

ASMApplianceManager

This is the class that manages the `ApplianceHostConfigData` objects, which are plugin internal objects to persist the `ASMAppliance` objects. This object is used to save or remove the ASMAppliance with the vRO persistent database.

Method:

- **save**: This method creates (persists) a new vRO resource with the given ASMAppliance object. Persisted password is encrypted.
  - name —User-given name of the ASMAppliance configuration.
  - serverUrl —URL used to reach the ASM server (For example, https://asm_server.dell.com).
  - userName— The user name to authenticate the REST connection with.
  - password—The password for the given username.
  - domain—Authentication domain (For example, ASMLOCAL)
  - sslTrustAllCerts—Boolean value to decide whether to trust all SSL certificates or not.
- remove: This method removes the vRO resource that contains the ASMAppliance configuration detail. The same effect is achieved when the vRO administrator removes the vRO resource of the target appliance object.

- appliance – The ASMAppliance object to be removed.

ASM Service

The *Service* object describes the ASM Service with minimum essential information.

Attributes:

- deployedBy – Shows the ASM user that deployed the *Service*.
- deploymentDate — Shows when the *Service* was deployed.
- description — Shows the description of the *Service* if available.
- id — Identifier used by the vRO to relate and find objects.
- name — Name of the *Service*.
- serviceId — Service identifier given by the ASM server
- status — Status of the Service (values: in_progress, complete, pending, error, and cancelled).
- templateName - templateName — Shows the name of the *Template* that provided the initial configuration for the *Service* deployment.

Methods:

- **ASMService**: Constructor method that takes another *ASMService* object and clones another object from it.
  - **deployment** — The source *ASMService* object.
- **ASMService**: Constructor method that takes the configuration data and creates a *ASMService* object.
  - **serviceId** — Service identifier issued by the ASM server
  - **name** — Name of the *Service*
  - **applianceId** — Identifier of the parent Appliance object
  - **status** — The *Service* status
  - **desc** — The Service description
  - **deploymentDate** — The timestamp to mark the deployment event
  - **deployedBy** – Shows the user who requested the deployment
  - **templateName** — Shows the *Template* that provided the configurations for the *Service* deployment
- **ASMService**: Default constructor method.

ASMServiceComponent

The *ServiceComponent* object contains minimum essential data of the resources or components in *Services*. Resources that are described can be servers, storages, cluster, virtual machines, and applications.

Attributes:

- **applianceId (String)** — The identifier of the Appliance object where the parent Service object belongs.
- **componentId (String)**: The ServiceComponent identifier given by the ASM server.
- **deploymentId (String)** — The identifier of the parent Service object
- **id (String)** — Identifier used by the vRO to relate and find objects
- **ip (String)** — The IP address of the resource if available
- **name (String)** — Name of the *ServiceComponent*
- **resourceName (String)** — Name of the hardware resource
- **serviceTag (String)** — Shows the service tag

- **type (String)** — The *ServiceComponent* type — value can be SERVER, STORAGE, CLUSTER, VIRTUALMACHINE, or SERVICE (which is the application on servers or vm's)

Methods

- **ASMServiceComponent** — Constructor method that takes another *ASMServiceComponent* object and clones another object from it.
  - **component** — The source *ASMServiceComponent* object.
- **ASMServiceComponent** — Constructor method that takes the configuration data and creates a *ASMServiceComponent* object.
  - **componentId** — *ServiceTemplateComponent* identifier issued by the ASM server.
  - **name** — Name of the Service.
  - **type** — Type of the component.
  - **deploymentId** — Identifier of the parent Service object.
  - **applianceId** —Identifier of the ASMAppliance object where the parent object exists
- **ASMServiceComponent** — Default constructor method.

ASMTemplate

This object describes the ASM Template with essential data about the Template.

Attributes:

- category (String): The category that groups the published templates.
- description (String): Description of the *Template*.
- id (String): The identifier used by the vRO to relate and find objects.
- name (String): Name of the Template.
- templateId (string): The identifier of the Template given by the ASM server.

Methods

- **ASMTemplate**: Constructor method that takes another ASMTemplate object and clones another object from it.
  - **Template** —The source ASMTemplate object
- **ASMTemplate**: Constructor method that takes the configuration data and creates a new ASMTemplate object.
  - **templateId** —The identifier issued by the ASM server to the ServiceTemplate.
  - **name** —Name of the *Template*.
  - **applianceId** —The identifier of the parent ASMAppliance object.
  - **desc** —Description of the *Template*.
  - **category** —The category of the *Template*.

9

Workflow and Actions

The Dell ASM vRO Integration is prepackaged with a set of sample workflows and actions. These can be used as out of the box to immediately enable common use cases. They may also be used to serve as examples that can be extended by advanced users. Underlying actions and API's are updated, but the workflows provide the same user experience as with versions 1.0 and 1.1.

Workflow

The prepackaged ASM vRO Integration workflow is divided into two types:

- **Configuration workflow** — These can be used to store and remove ASM appliance configurations within vRO. Appliances can be viewed in the Inventory panel once it is stored.
- **Sample workflow** — These samples illustrate the various capabilities of the integration. They are designed to allow users to immediately begin testing the integration between vRO and their ASM appliance.

Configuration Workflows

Add ASM Appliance:

This workflow asks the user to provide various configuration parameters for the ASM appliance and subsequently persists the configuration as a vRO resource. Once this workflow has been run successfully, the appliance appears in the inventory and can be selected as an input for any of the other sample workflows.

Remove ASM Appliance:

This workflow removes the selected appliance from the inventory.

Sample Workflows

The sample workflow is divided into three categories:

- **Asynchronous** — This workflow is designed to make API calls ASM appliance without waiting for the completion of the corresponding operations.
- **Non-interactive** – The workflow is designed without any user inputs, and serves as examples for vRA integration or other automation purposes.
- **Synchronous** — The workflow makes use of their asynchronous counterparts but include polling functionality to defer completion until the corresponding operation either succeeds or fails on the ASM appliance.

Asynchronous

Deploy service

This workflow allows users to deploy a new Service based on the template configuration chosen by the user. This workflow is completed after making the REST call to the ASM Appliance. The Service status is not tracked.

Scaledown service

Following are the set of scaledown operation workflow:

- **Scaledown Service**: Scale down the Service by removing the selected ServiceComponent (prompted during the workflow run).
- **Scaledown Service – App**: Scale down the target service by removing all resources of app type.
- **Scaledown Service – Cluster**: Scale down the target service by removing all resources of cluster type.
- **Scaledown Service – VM**: Scale down the target service by removing all resources of VM type.
- **Scaledown Service – Server**: Scale down the target service by removing all resources of server type.
- **Scaledown Service – Storage**: Scale down the target service by removing all resources of storage type.

Scaleup Service

Following are the set of sample asynchronous scaleup service workflow:

- **Scaleup Service – Server**: Scale up the service by adding more server components/resources.
- **Scaleup Service – Storage**: Scale up the service by adding more storage components/resources. The source storage to be cloned must be a new volume storage created through ASM service deployment or service update event.

Teardown Service

Following are the set of sample asynchronous teardown service workflow:

- **Teardown service – All**: Delete the service selected (prompted during the workflow run) and return all resources to the resource pool.
- **Teardown service – Selective**: Removes the service and all of its resources selected by their type. This workflow can be used to leave resources which are utilized by other services.

NOTE: The SERVER and VIRTUALMACHINE type components are automatically chosen for removal in the plugin version 2.0.

Non-interactive

There are two sample non-interactive workflows. These serve as templates and can be modified by the user according to their specific environments and requirements:

- **Deploy service** — Deploy a new service based on predefined parameters that are required. To use this workflow, the attributes must be set as appliance, ServiceTemplate, and baseDeploymentName.
- **Scaleup service – Server**- Scaleup a service based on the predefined parameters. The attributes must be set before this workflow can be used as service, targetServiceComponent, newComponentName, numberOfServers, and asm.

To set the attributes, select the workflow and click the **Edit** button (keyboard shortcut: Ctrl + E for computer and Command + E for Apple Mac Computer). It is shown as a pencil icon. In the editing mode, select the **General** tab. You can set the values for the attributes and save it.

Synchronous

Deploy service

This workflow allows users to deploy a new service based on the template configuration chosen by the user. This workflow tracks the status of the new Service, and it completes its run when the service status is no longer displaying as **in_progress**.

NOTE: The user may change the interval between the status checks (*delay* and *initDelay* attributes).

Scaledown service

This workflow scales down the service by removing the selected Service Component (prompted during the workflow run). This workflow tracks the status of the target Service, and it completes its run when the service status is no longer in **in_progress** status.

Scaleup service

There are two sample synchronous scaleup service workflows:

- **Scaleup service – Server**: Scale up the service by adding more server components or resources. This workflow tracks the status of the target Service, and it completes its run when the Service status is no longer in **in_progress**.
- **Scaleup service – Storage**: Scale up the service by adding more storage components or resources. This workflow tracks the status of the target Service, and it completes its run when the Service status is no longer in **in_progress**.

Teardown service

There are two sample synchronous teardown service workflows:

- **Teardown service – All**: Delete the service selected (prompted during the workflow run) and remove all its resources. This workflow tracks the existence of the target Service, and it completes its run when the Service no longer exists.
- **Teardown service – Selectively**: Removes the Service and all its resources selected by their type. This workflow tracks the existence of the target Service, and it completes its run when the Service no longer exists. The SERVER and VIRTUALMACHINE type components are automatically set for removal.

Actions

The actions provide elements that can be used to build workflow by mapping the ASM vRO Integration API methods. It also provides actions that help configuring the vRO environment for managing the inventory objects. There are two groups of the actions:

- com.dell.library.asm
- com.dell.library.asm.configuration

com.dell.library.asm

Actions in com.dell.library.asm

These actions helps in mapping of the API methods from the ASMAppliance. It is responsible for making REST to sync with the ASM server:

addDuplicateServer

This is the action that calls the addDuplicateServer method of the ASMAppliance API object. It returns a service object that has the updated information.

The addDuplicateServer(s) method cannot scale up the server(s) unless the target server for duplication is configured in this way:

- The server must be created with hardware settings that specify the server source to be server pool instead of manual entry.
- The duplicated servers must be provisioned from the same server pool from where the original server was pulled out.

addDuplicateStorage

The method, addDuplicateStorage, is called from the ASMAppliance API object. It returns a Service object that has updated information.

DeployService

This action calls the deployTemplate method of the ASMAppliance API object. It returns a Service object that has been deployed.

getServices

This calls the getServices method from the ASMAppliance. An array of Service objects is returned.

getTemplates

This calls the getTemplates method from the ASMAppliance. An array of Template objects is returned.

scaleDownService

This action calls the scaleDownService method of the ASMAppliance. A Service object with updated information is returned.

scaleDownServiceComponent

This action calls the scaleDownServiceComponent method from the ASMAppliance. A Service object with updated information is returned.

serviceExists

This calls serviceExist method to check if the Service exists in the ASM appliance. A boolean value is returned to indicate whether it exists or not.

teardownService

This action calls teardownService method, and the target Service is returned.

teardownServiceSelective

This calls the teardownServiceSelective method from the ASMAppliance, and the target Service is returned.

com.dell.library.asm.configuration

Actions in com.dell.library.asm.configuration

The actions in this group are used for creating inventory items for the ASM appliance resources and other general helps:

addAsmApplianceConfig

This is an action that persists the given configuration of the ASMAppliance object by calling the create method of the ASMApplianceManager.

removeAsmApplianceConfig

This removes the vRO resource used for persisting the ASMAppliance configuration (which is added through addAsmApplianceConfig action or ASMApplianceManager.save method).

generateUniqueServiceName

This action takes a base name of Service and appends sequence number at the end to ensure the uniqueness of the given name. To get the list of all the Service names, it calls the getServices method from the ASMAppliance.

getDefaultAsmAppliance

This action is useful in a vRO environment that manages multiple instances of ASMAppliance objects. It picks the first persisted ASMAppliance object from the list of all the ASMAppliance resources. The action can be modified to pick a specific ASMAppliance by its ID or name. This action is used in the sample workflows through the attribute property.

10

Using vRealize Orchestrator

By using vRO with ASM integration, you can perform the following actions:

- Add a new ASM Appliance object to the vRO inventory
- Deploy a new service to vRA
- Add a server to an existing service
- Teardown a service with selected resource types

Adding a New ASM Appliance Object to vRO Inventory

For using ASM vRO plugin, the best way is to utilize the vRO workflows which are packaged with the plugin. The following instruction explains how to add an ASM Appliance object to the vRO inventory:

1. Add ASM Appliance workflow from the **Configuration Workflows** folder on the left pane, then click the **Start workflow** button on the right pane.
2. Enter the values for the name and the server URL under Common Parameter. Either enable or disable the setting to trust all SSL certificates from the server. If an unverified certificate is installed in the ASM server, this setting must be set to true in order for the vRO to communicate with the ASM.
3. Click **Next**, and set the values for the authentication domain (ASMLOCAL by default), username and password.
4. Click **Submit**.

   The workflow should run successfully. To validate whether the configuration has been added or not, go to the Resources view, expand the folders and locate the new resource item.
5. Select the **Viewer** tab to cross-check the values that were used as input parameters for the workflow.

   The password is encrypted.

The user should be able to see the inventory objects for this ASM appliance server. Any templates or services already present in the ASM environment are also displayed.

Deploy a New Service to vRO

After adding the ASMAppliance object as a vRO resource, you can start running the workflow to deploy and update Services. In this example, the following steps describe how to deploy a new Service using vRO.

1. In run mode, click **Workflows view**, navigate **Library → Dell Acive System Manager → Sample Workflow → Synchronous → Deploy Service → Start Workflow** (which allow vRO workflows to call ASM for infrastructure).

   NOTE: The Synchronous workflow is used for demonstration.

   The default ASMAppliance object should already be set as the input parameter
2. Click **Submit** to proceed.
3. Click the selection box. A message stating **Not set** is displayed. Select the template to deploy in the window that is displayed.
4. Click **Submit**.
5. Provide the service name and modify the description of the service.
6. Click **Submit**.

The workflow uses the ASM API to call the target ASM server. The workflow enters the waiting phase until the service is fully deployed.

The workflow is completed after the Service deployment is completed (For example: The status of the service is no longer **in_progress**).

NOTE: Once the cloud infrastructure is deployed, VMware vRealize Automation (vRA) can be used to grant users self-service access to deploy virtual machines in the environment.

NOTE: The deployment can be confirmed using the ASM UI or the vRO Inventory view. The vRO Inventory view should also show the *Service* object and its information

Add a Server to an Existing Service

To add additional resources through the workflow, follow these steps:

1. Navigate **Library → Dell Active System Manager → Sample Workflow → Synchronous → Scaleup Service — Server Workflow**.

   NOTE: The Synchronous workflow is used for demonstration.
2. Set the default ASMAppliance object if the object is already as a resource.
3. Click **Submit** and proceed to the following page and select the target Service.
4. Click **Submit**, and select the target server Service Component to clone and add to the Service.
5. Set the name and the number of instances (must be an integer (for example, 0,1,2,3 and so on) and not a rational number value even though the primary data type (Number) for vRO suggests it can be.
6. Click **Submit** to request the scaleup to the ASM server.
7. Confirm if the new resource is added to the service after completion.

Teardown the Service

1. Locate the **Teardown Service** workflow.

   NOTE: The synchronous workflow is used for demonstration.

   NOTE: The ASMAppliance object should be already set to the default value.
2. Continue and select the target Service.
3. The workflow enters the waiting phase until the service is removed. After its completion, the Service is removed, yet the cluster is not removed from the infrastructure.
4. Confirm that the service is removed through the ASM UI or by vRO Inventory.

11

Integrating with vRealize Automation

The Dell Active System Manager integration with vRealize Orchestrator and its workflows are embedded to publish vRealize Automation Service Blueprints. The result allows you to call ASM using the vRealize Orchestrator workflow which is used through the vRA self-service catalog items.

Prerequisite for vRA Integration

Make sure that you have met the following prerequisites for vRA Integration:

1. There are two distinctive modes of operations in vRA. The system administrator configures the general system settings such as tenants management, branding, email servers configuration . The system administrator's account cannot be used to manage self-service items for users. Each tenant manages its catalogs and services for vRA catalog management. Therefore, there must be tenant organizations in the target vRA environment other than the `vsphere.local` tenant.
2. The vRA is shipped with vRO running on the same appliance. By default, vRA is configured to use this vRO. However, if the administrator chooses to use a different vRO server, the vRA system setting must be modified to use the external vRO server where the ASM vRO Integration exists.
3. Finally, there has to be an account with sufficient privilege to add and modify the tenant service catalog items. Either the Tenant Administrator or the Service Architect role grants this privilege. To run the following workflow, either of these accounts must be in use.

Non-Interactive Workflow as vRA XaaS Blueprints

By using the non-interactive workflow, you can create and publish vRA XaaS Blueprints in the easiest way. Modify the non-interactive sample workflow in vRO by setting the attributes or create a workflow with all the attributes set (no input exists to the workflow). For better illustration, assume that a sample workflow has been modified. Navigation path for the workflow is **Deploy Service workflow → Dell Active System Manager → Sample Workflow → Non-interactive folder**. vRA's attributes are set to deploy an ASM Template into an ASM Service to provision virtual machines in a vCenter cluster.
The following instruction describes how to convert the ASM sample workflow (non-reactive) to vRA-friendly workflow:

1. Log in to the vRA tenant console (It is accessible at **https://VRA_HOSTNAME/vcac/org/TENANT_NAME**) using the privileged account.

   NOTE: This step is optional. Existing Service can be used instead of using a new Service to contain the XaaS Blueprints.
2. Click the **Administration** tab.
3. Select **Catalog Management**, and then the **Services**.
4. Create a service, and add the service to an entitlement.
5. Click the **Design** tab.
6. In the left pane, expand **XaaS**, and then click **Select XaaS Blueprint**.
7. Click **Add** on the right pane to add a new XaaS Blueprint.
8. First select the **Deploy Service** workflow that was modified earlier in vRO.
9. Click **Next** .
10. Edit the details as needed, then proceed.

    NOTE: A blueprint form is not required for a non-interactive workflow.
11. Click **Next**, and make sure that no resource is provisioned after running the workflow.

12. Click **Add** to complete creating the XaaS Blueprint.

    A new XaaS Blueprint should appear on the right pane with its given name, description, and the status.

13. Select this item, and click **Publish**.

    The XaaS Blueprint is now visible as a catalog item.

    NOTE: It still has to be added to an entitled Service else, the catalog item has to be added to an entitlement individually.

14. Select the **Administration** tab and click **Catalog Management**, then click **Catalog Items** from the left pane.

    NOTE: The XaaS Blueprint is available here for additional configuration. Its source is set to Advanced Designer Service where it was published.

15. Select the catalog item, and click **Configure**.
16. When the **Configure Catalog Item** page loads edit the description and upload a custom icon (optional), and select the Service to put the catalog item in.

Now the catalog item can be used by the users but it must have the correct entitlement settings.
To make sure that the users have access to the new self-service catalog item, check if an entitlement exists for the users or the user group. This entitlement must include the service that contains the XaaS Blueprint or the catalog item individually.

NOTE: When an user requests the service from this kind of catalog item, only the description of the request is required. The attributes on the workflow provide the rest of the information which is required to call ASM.

Interactive Workflow as vRA XaaS Blueprints

To embed interactive workflows in vRA XaaS Blueprints, all required data must be processed as input parameters, which requires that most of the workflow bundled with the integration cannot be populated to vRA directly. To populate the workflow to vRA, the workflow must be modified. Consider the ASM configuration workflow and add ASM Appliance. This workflow can be encapsulated as a vRA XaaS Blueprint without any issue as all required data are passed as input parameters. Most sample workflows contain user interaction in their schema. This causes the workflow to pause and wait when it is called through vRA.
As example, deploy Service (Asynchronous) workflow is chosen. To convert the ASM sample workflow (Interactive workflow) to vRA-friendly workflow, do the following:

1. Log in to vRO with privileged credential.
2. Duplicate the sample workflow into a folder where the vRA tenant has access.
3. Edit the duplicated workflow.
4. Remove all the attributes from the workflow.
5. Create input parameters for **ServiceTemplate** (ASM:Template type), **DeploymentName** (string), and **DeploymentDescription** (string). There should be four input parameters in total.
6. Select the **Schema** tab, and remove all the design elements except for the action element, **DeployService**.
7. Edit the **DeployService** element. Bind the local parameters and source parameters accordingly.
8. Save the workflow and close the editor mode.

   NOTE: This workflow is ready to be encapsulated into a vRA XaaS Blueprint.

9. Go back to the vRA tenant console, create a XaaS Blueprint by choosing the modified workflow. Proceed until the new XaaS Blueprint is added.

   NOTE: The blueprint form has auto populated fields. Updating the fields is not mandatory.

10. Publish the XaaS Blueprint, and grant access to the users to this catalog item.
11. The input parameters are required to grant access to user to use the catalog items. It can only be chosen from the vRO inventory for any ASM vRO Integration API objects, For more information, see **Adding a New ASM Appliance Object to vRO Inventory**.
12. As the User Interaction elements are removed from the workflow, the vRA user has to carefully select the **ASM:Template** object that is in the selected **ASM:Appliance** object.

    NOTE: Mismatching in selection will make the ASM server to refuse the deployment.

13. Provide the service name and description to complete all the fields (Optional).

14. Click **Submit**.

The vRA triggers the vRO workflow to call the ASM server for a new service deployment.

Workflow Output

The ASM vRO sample workflow may have an output parameter with a custom API object type. If the vRA needs to use the information in the output parameter, it cannot be parsed easily in the vRA layer. Therefore, the workflow should be modified to parse the parameter into primitive data type objects before running it (For example: Add a script action that takes the custom API object and outputs string objects). In this way the output data can be easily used by vRA.

Upgrading the vRA XaaS Blueprint

To upgrade the vRA XaaS Blueprint along with the vRO and the ASM vRO plugin, complete the following tasks:

1. Put both vRO and vRA servers under maintenance. No deployments should be made during this period. Take note or snapshot to save the non-interactive workflows' configurations.
2. Uninstall the old plugin and configurations.
3. Upgrade the vRO server. Optionally, vRA server can be upgraded during this period.
4. After the upgrade, install the vRO plugin 2.0.
5. Modify the non-interactive workflows as needed in the new plugin installed.
6. Update the XaaS Blueprint in vRA to point to the new workflows.

12

Troubleshooting

You can troubleshoot ASM vRO Integration issue using:

- vRO Exceptions
- vRO Application Server Log
- ASM Log

vRO Exceptions

When an API method is invoked through actions or scripts, the operation is requested by the ASM server. However, the ASM server may return an error message due to internal server issues or bad user inputs. In such cases, the message is mapped to an exception to be thrown. Typically, these messages are reported with the following format:

Message<sequence number>

- Message code
- Severity
- Category
- Display message
- AgentId
- Timestamp
- Sequence number

There may be multiple messages sometimes. Plugins throw exceptions with appropriate messages to indicate any problem during the operation.

vRO Application Server Log

There are exceptional cases where the exception returned to the vRO which does not include enough details. In this case, the best way to track down the issue is by checking the vRO application server logs. You need access to the vRO appliance shell and read privileges for the following files (different in custom installations):

- /var/log/vco/app/catalina.sh
- /var/log/vco/app/server.out

Alternatively, the user can view the logs from the vRO control center.

ASM Troubleshooting

If there is no obvious issue in the vRO, the last resort is to check the ASM appliance log itself.

See the ASM documentations for further troubleshooting.

Go to http://www.dell.com/asmdocs for additional supporting documents on ASM.

For more information about ASM including how to videos, white papers, blogs, and support forum, see Dell Active System Manager page on Dell TechCenter:

http://www.dell.com/asmtechcenter

13

Uninstalling ASM vRO Integration

Before uninstalling the ASM vRO integration, make sure that the integration is already installed. The **Validate the Installation** section describes the procedure of validating ASM vRO integration. If the integration is not installed, the inventory tab does not show the root FinderResult object. It has Dell logo with the name “ASM”. Also, you can check the API Explorer to see whether the ASM SDK module exists or not. The following steps describe how to uninstall the ASM vRO Integration:

1. Make sure that no workflow is running. Notify users not to use the vRO server during the maintenance period for the removal process.
2. Remove the resources that persist the ASMAppliance objects. If it is not removed, it may be used again later when the ASM vRO integration is re-installed. To remove it, you can either run the workflow, remove ASM Appliance or you can remove the resources directly. To remove the resources using the workflow, start the workflow and select the ASMAppliance object(s) to remove. Another way to remove the resources is, go to the Resources view. Select Design mode, click **Resources tab**, then expand the folders to locate the resources that define the ASMAppliance objects (**Root** → **Library** → **Dell Active System Manager** → **Configuration**). Right click the resource item, and click **Delete**.
3. Go to the **Packages** view, and delete the **com.dell.plugin.asm** package along with the contents of the package.

   This action also deletes the workflows, actions, and icon resources.
4. Log in to the vRO server shell using SSH or console.
5. Change the working directory to **/var/lib/vco/app-server/plugins/**. Remove the **o11nplugin-asm.dar** file.
6. Restart the vRO configurator and the application server. To restart the configurator server, run the command **/etc/init.d/vco-configurator restart** in the shell with root privilege. To restart the vRO application server, either run the command **/etc/init.d/vco-server restart** in the shell or use the vRO configurator. To restart the vRO application server through the configurator, login to the vRO control center web console. Click **Startup** Options. When the **Startup option** page is displayed. click the **Restart Service** .

   The ASM vRO Integration is uninstalled.
7. Start the vRO console to verify if the integration is uninstalled.

14

Known Issues and Forewarning

- In order to use the deployTemplate method in the ASMAppliance object, the target template in ASM should have all the required fields set to a static value or to auto-generate the value upon deployment request. In ASM UI, the deployment is not allowed unless the required fields are all set. However, the vRO API plugin does not check to see if all the fields are set. Therefore this can cause a deployment to start and fail later when the required fields are found to be not set.
- The current version of the plugin can add resources only by duplicating existing component(s) already defined in the Service. Component types other than SERVER and STORAGE are not supported for duplication at this point. For the STORAGE component scale up, the source component must be created from the ASM Service deployment or update event. Otherwise, the ASM throws exception for duplicating a reference storage volume in a single Service.
- Duplicating storage has not been fully tested for NetApp storage type.
- Non-interactive sample workflows that are packaged with the plugin must be updated with valid attributes before they can be used.
- As with the ASM UI, adding storage to a Service with a hyper-v cluster is disabled.
- Tearing down a Service with shared resources must be requested with precaution. ASM does not checkchecks whether the resources requested for teardown are shared or not. If shared component is requested for teardown, it will result in an exception. Selective teardown API method should be used in order to remove a Service without returning the resources to the pool.
- For best practice, disable the standard users of vRO from the workflows in the Configuration Workflows folder. Ideally, a group of privileged users and ASM administrators should have access to these workflows (especially the ones that can deploy, update, or remove Service). Right-click on the target workflow(s) and click on the Edit access rights button. See VMware vRO documentation for more detail how to control the access rights to workflows. These workflows are used to persist or remove ASMAppliance objects as vRO resources.
- The vRO fails to load ASMAppliance object from the resource item if the connection fails.

15

Compatibility Matrix

The matrix below describes the compatibility between different plugin versions, ASM releases, and vRO versions:

Table 1. Compatibility Matrix

ASM vRO Plugin Version	ASM Release	vRO Version
1.0	8.1.0 through 8.1.1	5.5 and 6.0
1.1	8.1.0 through 8.2.0	5.5 and 6.0
2.0	8.2.1 and newer releases	7.0

NOTE: Earlier versions, 1.0 ~ 1.1, are deprecated. For ongoing support, upgrade the ASM server and vRO to the latest version.