IBM Cloud Object Storage System™
Version 3.15.0

Appliance Docker Container Guide

This edition applies to IBM Cloud Object Storage System™ and is valid until replaced by new editions.

Contents

Chapter 1. Getting Started

The Docker Appliance Container provides the Manager Node and Accesser® Node as Docker-compatible Container images.

The Appliance Container image can be deployed to customer-managed hardware on a customer-managed operating system. The Appliance Container can be monitored, managed, and deployed by using the Manager Web Interface and Manager API.

⚠ **Attention:** The information that is provided presumes a working knowledge of Docker and the deployment of Docker-based applications.

Note: For more information on Docker, see the Docker web site. The Slicestor® Node is not available as an Appliance Container.

New capabilities

Support for Docker images was developed to provide flexible deployment of a system Manager. Customers can use their familiar hardware and operating systems to run IBM Cloud Object Storage System software alongside existing applications.

Of the Cloud Object Storage System hardware nodes, the Accesser Node was containerized first as it provided the most clear benefit and the best fit. The Manager Node was containerized next as follow-on work from the Accesser Container.

Benefits

General benefits of appliance containers

Appliance Containers allow customers to:

- Use existing infrastructure instead of purchasing dedicated appliances
- Use existing customer workflows which utilize Docker to deploy and manage other containers in their existing infrastructure
- Apply the benefits of virtualization with significantly lower overhead than a traditional hypervisor

Specific benefits of Accesser Containers

The Accesser Container allows customers to:

- Add Accesser capabilities to a Manager application with less on-device configuration than either an Accesser Appliance or the Accesser Application
- Combines the flexibility of the Accesser Application with the manageability and ease-of-use of an Accesser Appliance

Specific benefits of manager containers

The Manager Container allows customers to:

- Migrate from Manager Appliance to Manager Container
- Back up the Manager Appliance to a Manager Container, which could then be automated
- Use the Manager Container on any compatible generic hardware

Terms and components

Common terms that are used throughout.

Accesser Container

A Container that uses the Accesser software.

Appliance Container

A Container that uses either the Accesser or Manager software.

Container

A running instance of a Docker image, representing a virtual environment.

Docker

A collection of tools that are used to deploy, run, and monitor applications that are running inside Linux Containers. See the Docker web site, for more information.

Image

In Docker, an `image` is composed of a file system archive and metadata used to describe the file system archive. The **`docker run`** command starts a Container from a base image, where the image remains unmodified.

Manager Container

A Container that uses the Manager software.

Workflow

An Appliance Container resides in a customer-supplied operating system. An administrator can load an Appliance Container image into the Docker daemon then deploy one or more instances of that image as a container subject to networking restrictions.

A running Appliance Container supports all the same client interfaces as its respective Appliance. The Appliance Container appears in the Manager Web Interface in a nearly indistinguishable manner to the corresponding Appliance.

Note: Docker Containers cannot be run on a device running the ClevOS operating system.

Use cases

Increase deployment flexibility

An Appliance Container can be run on the same physical hardware as other applications using an existing operating system. This better utilizes existing infrastructure.

Accesser container augments or replaces the Accesser® Application

The Accesser Container requires less hand-configuration than the Accesser Application, and can be managed using the Manager Web Interface or Manager API. If the customer is not using a Docker-compatible Linux distribution, the Accesser Application is still available.

If there are no port conflicts and the operating system supports Docker, an Accesser Container may run on the same hardware as an Accesser Application.

Feature impact and remediation

Version upgrades

Appliance Containers cannot be upgraded through the Manager Web Interface.

Note: See “Upgrading a container” on page 15 for details.

Chapter 2. System and Network Requirements

Accesser container specific requirements

Accesser container memory requirements

The Accesser Container tries to scale RAM usage automatically based on the amount of system memory.

The maximum amount of RAM allocated to Accesser Container can be set through the environment variable *MAX_MEMORY*.

CAUTION: The Accesser Container does not attempt to use more than 32 GB of RAM, regardless of the *MAX_MEMORY* setting.

Table 1. Memory Requirements

Deployment Scenario	Suggested System Memory (GB)	Suggested *MAX_MEMORY* setting (MB)
Server Class Systems	16+	4000
Server Class Systems (Using Vault Mirrors)	32+	8000

Note: See "Container environment variables" on page 8, for more information.

CAUTION: For scenarios where the file size is large (> 10 GB) and the number of concurrent uploads/downloads is large (> 50), contact IBM customer support for more guidance.

Accesser container storage requirements

The Accesser Container needs a modest amount of storage for logs and other state information.

Note: IBM recommends 60 GB capacity.

Manager container specific requirements

Manager container memory requirements

The Manager Container tries to scale RAM automatically usage based on the amount of memory in the system.

The maximum amount of RAM allocated to the Manager Container can be set through the environment variables *DS_MYSQL_MAX_MEMORY* and *DS_MANAGER_MAX_MEMORY*.

Note: There is approximately a 2 GB more base memory requirement in addition to the listed settings.

Table 2. Memory Requirements

Deployment Scenario	Suggested System Memory (GB)	Suggested *DS_MYSQL_MAX_MEMORY* setting (MB)	Suggested *DS_MANAGER_MAX_MEMORY* setting (MB)
Server Class Systems	16+	25% of System RAM	25% of System RAM

Manager container storage requirements

Note: IBM recommends 1 TB per 1,000 vaults.

System and network configuration

System clocks

Nodes running Appliance Containers must be configured for clock synchronization through NTP with other Nodes in the system.

The host OS should synchronize to the same NTP server as the remainder of the system.

⚠ **CAUTION:** Unlike the Manager Appliance, the Manager Container cannot provide NTP synchronization services. Devices that are managed by a Manager Container must be configured to use an external NTP server.

Note: For more information on NTP synchronization, see the *Time Synchronization Configuration Guide* and the documentation for the preferred NTP client.

Network ports

Appliance Containers need connectivity to all system nodes.

Note: Some Container services might require more network configuration for access. See "Container network configuration" on page 10, for more information.

Table 3. Port usage for appliance use

Destination	Port	Purpose
Slicestor Nodes	`TCP 5000`	Data operations and registry lookup
Slicestor Nodes	`TCP 7 (OPEN or REJECT)`	Round-trip time calculation
NTP	`TCP or UDP 123`	NTP messaging (configured in the host)
Manager Node	`TCP 443 (by default)`	Manager API Vault usage query via HTTPS
Manager CNC	`TCP 8088`	Management control port (non-Manager Nodes)

⚠ **CAUTION:** TCP port 7 may be closed, but any firewall rules should send REJECT messages and not drop packets.

Chapter 3. Configure the Appliance Container

Docker parameters

Note: For more information on parameters, see the Docker documentation.

Table 4. Docker parameters

Parameter	Purpose	Required	See Also
-i	Run interactively. Pair with **-t**.	No. However, not specifying these parameters can result in terminal errors on the console that uses **docker exec**.	
-t	Run with a tty session. Pair with **-i**.	No. However, not specifying these parameters can result in terminal errors on the console that uses **docker exec**.	
-d	Run a Container that is detached in the background.	No. However, the Accesser Container runs in the foreground if this parameter is not specified.	
--env	Each environment variable to be specified. Each needs its own **--env** parameter.	Yes	"Container environment variables" on page 8
-v	Volume to bind and mount from the host OS into the Accesser Container persistent area. The host directory must be writeable, the directory cannot be shared between Container instances and the directory must be unique.	No, for basic Container functions. Yes, for a Container upgrade or viewing Container application logs with host tools.	"Upgrading a container" on page 15
--net	Used to change the networking mode of the Container. However, appliance Containers support both **--net="host"** and **--net="bridge"**. Only one Container started with **--net="host"** can be running on a single machine at a time.	Yes	"Container network configuration" on page 10
-p	Host port to bind a Container port.	Yes, when using **--net="bridge"**	"Network ports" on page 6

Table 4. Docker parameters (continued)

Parameter	Purpose	Required	See Also
`--hostname`	Set the hostname of the Container.	Yes, when using **`--net="bridge"`**. A specified Container hostname is needed for Container upgrade when upgrading **`--net="bridge"`** containers. The default hostname that is provided by Docker (based on the Container ID) is not sufficient. The hostname must not be set to **`localhost`**.	"Upgrading a container" on page 15

Container environment variables

Runtime environment variables set after the **`docker run`** command by using **`--env`** flag parameters configure Appliance Containers.

Note: More settings can be set through the Manager Web Interface.

Example: Using Container Environment Variables

```
Set each parameter with its own --env flag.

# sudo docker run -itd --env="DS_MANAGER_IP=192.168.79.15" --env="DS_MANAGER_AUTOACCEPT=true" ...
{image-id or tag}
```

Appliance Container use

Accesser Container environment variables

DS_MANAGER_IP

The IP address of the Manager node. Accepts IPv4 or IPv6 addresses.

DS_MANAGER_AUTOACCEPT* or *DS_MANAGER_FINGERPRINT

If set to *DS_MANAGER_AUTOACCEPT=true*, the Container accepts any certificate the Manager Node provides from the IP address that is specified in *DS_MANAGER_IP*.

If *DS_MANAGER_FINGERPRINT* has any value that is specified, the Container accepts the Manager Node-provided certificate at the IP address that is specified in *DS_MANAGER_IP* only if the certificate fingerprint matches the fingerprint. The fingerprint should be provided in colon-separated pairs of hex digits.

For example, the *DS_MANAGER_FINGERPRINT=aa:bb:cc* accepts any Manager fingerprint beginning with `aa:bb:cc`.

Manual, interactive certificate approval (that is, `manager ip` at the `localadmin` shell) is not available for the Accesser Container.

Table 5. Situational variables

Variable	Variable Type	Purpose	Required?
DS_MANAGER_PORT	Integer	Port for Manager services	Yes when using a Manager Container on any port other than 443
DS_CNC_EXTERNAL_CONNECT_IP	String	External IP address Manager Node contacts	Yes when **`--net="bridge"`**. To remove the IP address, enter no value for this environment variable.
DS_CNC_EXTERNAL_CONNECT_IP_IPV6	String	External IPv6 address Manager Node contacts	Yes when **`--net="bridge"`**. To remove the IPv6 address, enter no value for this environment variable.
DS_CNC_EXTERNAL_CONNECT_PORT	Integer	Port number the Manager Node contacts	Yes when **`--net="bridge"`**
DS_PKI_SUBJECT	String	Specifies an alternative PKI subject	

Manager container environment variables

Variable	Variable Type	Purpose	Required?
DS_MANAGER_EXTERNAL_CONNECT_IP	String	Externally referenceable IP address for Manager services.	No, but not specifying can result in `manager verify` errors on appliances at the `localadmin` shell. To remove the IP address, provide the environment variable with no value.
DS_MANAGER_EXTERNAL_CONNECT_IP_IPV6	String	Externally referenceable IPv6 address for Manager services.	No, but not specifying can result in `manager verify` errors on appliances at the `localadmin` shell. To remove the IPv6 address, provide the environment variable with no value.

Variable	Variable Type	Purpose	Required?
DS_MANAGER_EXTERNAL_CONNECT_PORT	Integer	Forwarded port for Manager services	Yes, when **--net="bridge"** and forwarding a port other than 443
DS_MANAGER_MYSQL_AUTOEXTEND_MAX	Integer	Tunes the soft-limit of how much drive capacity is allocated to MySQL in the Manager Container. Specified in megabytes. If not specified, it is tuned according to the size of the storage volume that is mounted into the container with -v.	No
DS_MANAGER_MYSQL_MAX_MEMORY	Integer	Tunes how much memory is allocated to MySQL in the Manager Container. Specified in megabytes. If not specified, it is tuned according to the system RAM.	No
DS_MANAGER_MAX_MEMORY	Integer	Tuning parameter to limit the amount of memory that is allocated to the Manager process in the Manager Container. Specified in megabytes. If not specified, it is tuned according to the system RAM.	No

Container network configuration

IBM's implementation of Docker supports two network operation modes.

--net="host"

The Container shares networking with the host OS. Only one Appliance Container that uses **--net="host"** can run at one time on a single host. When using **--net="host"**, some ports that are used by services inside the Container can conflict with ports that are opened by services in your hosts network namespace. These ports can be remapped by using the *DS_CNC_** variables noted in "Container environment variables" on page 8 or reconfigured through the Manager Web Interface for the Accesser service.

Example: Command line to start an Accesser Container with **--net="host"**.

```
# sudo docker run -itd --env=“DS_MANAGER_IP=192.168.79.15” --env=“DS_MANAGER_AUTOACCEPT=true”
-v /home/data/container-data-1:/container-persistence:rw --net=“host” clevos-accesser:3.7.0.60
```

Note: The Docker host must not use `localhost` as its hostname when using an Appliance Container.

--net="bridge"

The Container uses a separate network namespace from the host OS. Docker automatically allocates an internal NAT-like network. Some ports must be published from the Container to the host.

Example: Command line with **--net="bridge"**.

This command starts an Accesser Container and maps:

- Host HTTP port 8080 to Container port 80
- Host HTTPS port 4430 to Container port 443
- Host Management Services port 8088 to Container port 27015

```
# sudo docker run -itd --hostname=“accesser-container.example.com” -p 8080:80 -p 4430:443 -p
27015:8088
--env=“DS_CNC_EXTERNAL_CONNECT_PORT=27015” --env=“DS_CNC_EXTERNAL_CONNECT_IP=192.168.79.99”
--env=“DS_MANAGER_IP=192.168.79.15” --env=“DS_MANAGER_AUTOACCEPT=true”
-v /home/data/container-data-1:/container-persistence:rw --net=“bridge” clevos-
accesser:3.7.0.60
```

Note: See the Port Mapping (`-p`) flag for Docker.

Note: The **`hostname`** parameter must be set to a valid host name other than `localhost`.

Container ports

Appliance Containers run services on several ports.

Depending on the use of `--net="host"` versus `--net="bridge"` (see “Container network configuration” on page 10), some of these ports might need to be published to host ports to access these services from outside the Docker host by using the `-p` flag.

Table 6. Running Ports for Appliance Container

TCP Port	Accesser Container Purpose	Manager Container Purpose
80	Accesser software HTTP	
443	Accesser software HTTPS	Manager software HTTPS
8080	Accesser software HTTP	
8088	Manager CNC services	
8192	Device API	
8443	Accesser software HTTPS	

Enabling IPv6 firewall, port forwarding, and IP masquerading on the host for bridged containers

Docker provides the needed firewall security, port forwarding, and IP masquerading for IPv4, but does not yet do so automatically for IPv6.

About this task

Until Docker upgrades the Daemon to automatically provide this functionality, you must manually configure the host to use `ip6tables` rules.

Procedure

1. Start the bridged ClevOS containers.
2. Run **`iptables -S`** and **`iptables -t nat -S`** to see how Docker implemented the rules for IPv4 and note the configuration.
3. Set the `ip6tables` rules according to the IPv4 implementation and adjust as necessary for IPv6.

Chapter 4. Deployment

Note: Docker commands must be run with `root` privileges. In the following examples, this is represented through the issuing of the sudo command before each `docker` command.

Prerequisites

- Docker-compatible Linux operating system installation
- NTP synchronization must be configured in the host operating system
- Docker 1.3 or later

API compatibility

- All APIs supported by the Accesser appliance are also supported by the Accesser Container.
- All APIs supported by the Manager appliance are also supported by the Manager Container.

Note: For a full list of the supported features, see the *IBM Cloud Storage Object API 2.5 Development Guide*, the *OpenStack Object Storage API 1.0 Developer Guide*, or the the *IBM Simple Object over HTTP API 2.5 Developer Guide*.

Working with a container

Creating a new container

Procedure

1. Load the Container image into Docker running on your server.

   Example: Loading Appliance Containers

```
Loading an Appliance Container image stored on a remote HTTP server
# curl http://example.website/clevos-3.7.0.60-accesser-container.tar.gz | sudo docker load
Loading an Appliance Container image stored locally
# cat clevos-3.7.0.60-accesser-container.tar.gz | sudo docker load
```

2. List the Container images to find either the repository/tag pair or image ID to start a Container.

   Example: Loading Appliance Containers

```
The docker run command uses either the repository/tag pair or image ID to identify the
Container image.

# sudo docker images
REPOSITORY        TAG        IMAGE ID      CREATED         VIRTUAL SIZE
clevos-manager    3.7.0.60   0015d07492d0  4 days ago      1.795 GB
clevos-accesser   3.7.0.60   c260ab58b9ee  4 days ago      965.1 MB
scratch           latest     511136ea3c5a  23 months ago   0 B

To load an Appliance Container, the command would use either:

# sudo docker run ... clevos-accesser:3.7.0.60

or

# sudo docker run . . . c260ab58b9ee
```

3. Start a new container using **—env** to specify environment variables as documented in the *IBM Manager Administration Guide*. If successful, the Container ID is written to `stdout`.

   Example: Loading Appliance Containers

```
Starting an Appliance Container:

# sudo docker run -itd --env="DS_MANAGER_IP=192.168.79.15"
--env="DS_MANAGER_AUTOACCEPT=true"
-v /home/data/container-data-1:/container-persistence:rw
--net=host clevos-accesser:3.7.0.60

Returns:
c644228d71f84be86420b44ed53f9927401fcd5c638e4ce77fdd72ec28b47d43[1]

[1] This is the Container ID.
```

4. Approve the Container Instance in the Manager Web Interface according to the procedure found in the "Approve registered devices" section in the *IBM Manager Administration Guide*.

   Note: Once it is running, an Appliance Container can be used like its corresponding Appliance, subject to the other limitations described in this guide.

Stopping a running container

Procedure

Enter the **`docker stop`** command with the Container ID in the command line to stop the Container.

```
# sudo docker stop {container-id}
```

Note: *{container-id}* can be the first six characters of the Container ID.

Example: Loading Appliance Containers

```
This Container ID:

c644228d71f84be86420b44ed53f9927401fcd5c638e4ce77fdd72ec28b47d43

Can be entered in the docker stop command as:

# sudo docker stop c64422
```

Resuming a stopped container

Procedure

Enter the **`docker start`** command with the Container ID in the command line to resume the Container.

```
# sudo docker start {container-id}
```

Note: This preserves environment variables set during the initial **docker run** statement.

Executing an interactive shell

Procedure

To troubleshoot or debug a container, enter the **docker exec** command with the Container ID and a shell file and path in the command line.

```
# sudo docker exec -it {container-id} /bin/bash
```

Note: If **-i** and **-t** parameters are not specified in original **docker run** statement when starting the container, terminal-related error messages may be displayed while trying to use commands inside the Container.

Note: If the Appliance Container was started with **—net-host**, the prompt will change to `{hostname}#`.

Upgrading a container

Containers cannot be upgraded through the Manager Web Interface. A Container must be upgraded on the server on which it runs.

Before you begin

Note: The previous Container must have been run with a persistent volume mounted into `container-persistence` with **-v**, as shown in "Creating a new container" on page 13

Procedure

1. Load the new Container image into Docker

```
# gunzip -c clevos-3.7.0.99-accesser-container.tar.gz | sudo docker load
```

2. Stop the old Container

```
# docker stop {container-id}
```

3. Run the new Container image using the same persistent volume, environment variables and hostname used for the previous Container.

   Example: Using **—net="host"**

```
The hostname (—hostname) was not specified as —net="host" is used;
the Container inherits the hostname of the host OS.

# sudo docker run -itd --env="DS_MANAGER_IP=192.168.79.15" --env="DS_MANAGER_AUTOACCEPT=true"
-v /home/data/container-data-1:/container-persistence:rw --net="host" clevos-accesser:3.7.0.99
```

   Example: Using **—net="bridge"**

```
# sudo docker run -itd --hostname="accesser-container.example.com"
--env="DS_MANAGER_IP=192.168.79.15" --env="DS_MANAGER_AUTOACCEPT=true"
--env="DS_CNC_EXTERNAL_CONNECT_PORT=27015" --env="DS_CNC_EXTERNAL_CONNECT_IP=192.168.79.99"
-p 8080:80 -p 4430:443 -p 27015:8088 -v /home/data/container-data-1:/container-persistence:rw
--net="bridge" clevos-accesser:3.7.0.99
```

   The initial environment variables (**—env**) do not need to be specified for upgraded Containers but port presentations (using **-p**) do.

4. Once the new Container has started, remove the old Container

```
# docker rm {container-id}
```

⚠ **Attention:** This will remove the Container Instance from the Docker application, but will not remove the Container Image from the host operating system. That must be done separately.

Migrating container devices to IPv6

Migrate your container devices to IPv6 to guard against the eventual depletion of IPv4 addresses and ensure that no outage in continuous connectivity occurs.

About this task

The system can run several network implementations:

- Single-stack IPv4 (The Manager appliance and all devices are configured with IPv4 addresses only).
- Dual-stack IPv4/IPv6 (The Manager appliance and all devices are configured with both IPv4 and IPv6 addresses).
- Single-stack IPv6 (The Manager appliance and all devices are configured with IPv6 addresses). IPv4 addresses are removed from the Manager and all devices.
- Mixed IPv4/IPv6 (The Manager appliance is configured with both IPv4 and IPv6 addresses). Some storage pools have devices that are configured with IPv6 addresses, while other storage pools have devices that are configured with IPv4 addresses only.

Procedure

1. Stop the Docker container.
2. Start a new container with the same persistent data while you change the appropriate IP environment variables:

   - *DS_MANAGER_IP*
   - *DS_CNC_EXTERNAL_CONNECT_IP*
   - *DS_CNC_EXTERNAL_CONNECT_IP_IPV6*
   - *DS_MANAGER_EXTERNAL_CONNECT_IP*
   - *DS_MANAGER_EXTERNAL_CONNECT_IP_IPV6*

   Note: To remove an IP address, provide the environment variable with no value. For example, **`--env="DS_CNC_EXTERNAL_CONNECT_IP="`**.

Convert a Manager Container to or from a Manager Appliance

An existing Manager Appliance installation can be converted to a Manager Application installation, and vice versa.

Converting from Manager Appliance into a Manager Container

Procedure

1. Back up the Manager Appliance.
2. Configure and start a Manager Container instance that is running the same software version as the Manager Appliance.
3. Restore the backup of the Manager Appliance on the Manager Container.
4. Reconfigure any appliances joined to the system served by the Manager Appliance, when an IP address or port changed regarding where Manager services originate.

Converting from Manager Container into a Manager Appliance

Manager Appliance installations do not support the use of ports other than 443 for Manager services.

Procedure

1. Back up the Manager Container.
2. Image and configure a Manager Appliance running the same software version as the Manager Container.
3. Restore the backup of the Manager Container on the Manager Appliance.
4. Reconfigure any appliances joined to the system served by the Manager Container, when an IP address or port changed regarding where Manager services originate.

Upgrading a system managed by a Manager Container

Before you begin

A Manager Container cannot be upgraded through the normal Manager UI orchestration. Upgrading of non-container devices in a Manager Container system requires that the Manager Container is upgraded externally before using the Manager Web Interface to upgrade the remaining supported devices.

Procedure

1. Upgrade the Manager Container to the wanted version per the procedure in "Upgrading a container" on page 15.
2. Upgrade any Accesser Container instances to the wanted version per the procedure in "Upgrading a container" on page 15.
3. Use the Manager Web Interface to upgrade the remaining hardware devices.

Chapter 5. Limitations

Monitoring

Monitoring the Appliance Container in the Manager Web Interface is nearly identical to monitoring an appliance.

As it is a software solution, the Appliance Container does not provide generalized hardware- and operating system-level monitoring.

Statistics that are not displayed (or provided in the Manager REST API or the Device API) include, but are not limited to:

- CPU temperatures
- Disk I/O
- Disk temperatures
- Fan speeds

Events/Monitoring not performed include, but are not limited to:

- RAID monitoring
- CPU/Disk temperature alerts
- Device reboot events
- Kernel dump events (does not apply to Containers).

Some graphs/stats for Appliance Containers might behave differently than expected:

CPU usage and system load

These graphs/stats reflect the CPU usage and system load of the host machine as a whole, not an individual Appliance Container.

Network I/O

These graphs reflect the interfaces visible to the Appliance Container and are different depending on the Containers network settings (that is, **`docker run --net={"host" | "bridge"}`**).

Extra rules and restrictions

- The operator cannot reboot a Device from within a Container.
- The Network Utility Tool (**`nut`**) commands do not work in a Container.
- Unlike the Manager Appliance, the Manager Container does not provide NTP synchronization services to other devices.
- For more information, see "System clocks" on page 6.

Chapter 6. Troubleshooting

Error scenarios

Since the Accesser and Manager Containers provide an HTTP interface, errors are returned to traditional HTTP clients as HTTP status codes.

Container logs can be examined after an HTTP error code is returned to the client to further diagnose a particular status code.

Note: For more information about the operation of the HTTP interfaces that are provided by the Appliance Container, see the appropriate API guide.

Error starting a container

Troubleshoot an appliance container that stops immediately or that won't start.

An appliance container started in the background stops immediately

When starting an Appliance Container in the background (**-d** flag when using the **`docker run`** command), the Container might stop immediately, even though it issued a Container ID to the console.

It generally occurs because the initialization routines in the Container fail due to incorrect or insufficient environment variables that are passed to the Container. These logs can be visible if the Container is run in the foreground, but can also be visible in Docker logs.

Note: For more information, see Docker run command and "Container environment variables" on page 8.

Example: Error when starting an Accesser Container

```
# sudo docker run -it clevos-accesser:3.7.0.60

[ ok ] Configuring Container persistence ... done.
Error: No manager IP specified
expected DS_MANAGER_IP="<ip>"
Error: /etc/rcS.Container/300-env-mapping failed with return code 1
```

In this case, the environment variable *DS_MANAGER_IP* was expected but was not present.

An appliance container won't start

Check `logs/dsnet-core/stdout.log` in container-persistence for one of the following errors:

-
  ```
  ulimit max open files configured to <CONFIGURED_OPEN_FILES> instead of <MAX_OPEN_FILES>
  ```

  where *<CONFIGURED_OPEN_FILES>* is the number of open files currently allowed by Docker and *<MAX_OPEN_FILES>* is the number of open files needed for the dsnet-core process.

-
  ```
  ulimit max locked memory configured to <CONFIGURED_LOCKED_MEMORY> instead of <MAX_LOCKED_MEMORY>
  ```

  where *<CONFIGURED_LOCKED_MEMORY>* is the amount of locked memory currently allowed by Docker and *<MAX_LOCKED_MEMORY>* is the amount of locked memory needed for the dnset-core process.

If either error appears, add `-ulimit nofile=1000000:1000000 --ulimit memlock=-1` to the **`docker run`** command.

Error mounting persistent volume on SELinux systems

In some cases, primarily when using a Linux distribution that uses SELinux to provide access control for Docker containers, when starting an Appliance Container and mounting a persistent volume, the container fails to start with `Permission denied` errors appearing in the Docker logs or as console output.

Use the **`z`** switch to the **`-v`** mount option to resolve it.

Note: For more information, see Using Volumes with Docker can cause problems with SELinus.

Example: SELinux Volume Error

```
[root@dc-a3100-3740 ~]# docker run -it --env=DS_MANAGER_EXTERNAL_CONNECT_IP=192.168.28.141
--env=DS_MANAGER_EXTERNAL_CONNECT_PORT=8503 -p 8503:443 -p 8502:80
-v /home/docker-persistence/docker-nat-mgr2:/container-persistence:rw
--hostname=“docker-nat-mgr2” 1bb97d13f249
Usage of loopback devices is strongly discouraged for production use.
Either use `--storage-opt dm.thinpooldev` or use `--storage-opt dm.no_warn_on_loop_devices=true`
to suppress this warning.
[....] Configuring container persistence...find: `/run/sendsigs.omit.d': No such file or directory
find: `/run/mysqld': No such file or directory
find: `/run/network': No such file or directory
find: `/run/apache2': No such file or directory
find: `/run/rdnssd': No such file or directory
find: `/run/lock/apache2': No such file or directory
cp: cannot create directory `/container-persistence/logs': Permission denied
cp: cannot create directory `/container-persistence/config': Permission denied
cp: cannot create directory `/container-persistence/dsnet-md': Permission denied
cp: cannot create directory `/container-persistence/dsnet-core': Permission denied
cp: cannot create directory `/container-persistence/dsnet-manager': Permission denied
cp: cannot create directory `/container-persistence/mysql': Permission denied
mkdir: cannot create directory `/container-persistence/tmp': Permission denied
chown: cannot dereference `/tmp/mysql': No such file or directory
mkdir: cannot create directory `/container-persistence/tmp': Permission denied
done.
Traceback (most recent call last):
 File “/etc/rcS.container/300-env-mapping”, line 135, in <module>
  sys.exit(main())
 File “/etc/rcS.container/300-env-mapping”, line 127, in main
  message_md.write('/security/bootstrap', md_parameters)
 File “/usr/lib/python2.7/dist-packages/dsnet/platform/message_md.py”, line 28, in write
  with open(tmp_name, 'w') as file_:
IOError: [Errno 2] No such file or directory: '/var/lib/dsnet-md/spool/tmp/
144292875158610106_message-md_1_583571'
Error: /etc/rcS.container/300-env-mapping failed with return code 1
```

Manager IP display

When using a Manager Container, the IP address listed for the Manager Container in the Manager Web Interface may not reflect the externally visible IP address and instead may display `127.0.0.1`. This is expected behavior.

Manager verify command failures on appliances

Using the **`manager verify`** command in the **`localadmin`** shell of a Manager Container might fail with `curl` error code 51.

```
dc-a2000-2424# manager verify 10.10.14.16:9443
Unable to reach manager https: Command '('curl', '-s', '--cacert', '/tmp/tmpSOEWOI', '-I',
'https://10.10.14.16:9443/manager/api/sandra/ca/certificate.pem')' returned non-zero exit status
51
```

Normal system functions are not affected.

Container logs

Container logs can be viewed in two locations.

- The host system of the persistent volume that is mounted into the Container in the `logs` folder.

  Example: Location of log files

```
If the Container was started with -v /home/admin/container-data:/container-persistence:rw,
the logs are visible in /home/admin/container-data/logs/.
```

- In the Container, either `/container-persistence/logs` or `/var/log`.

Note: The content of all paths that are listed is identical.

Events

In addition to logging, only non-hardware exceptions when using various APIs show as events in the Manager Web Interface.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM® in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785

US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.

Homologation statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.

Printed in USA