Upgrade Guide for High-Availability Deployments

Release 6.5.0

Zenoss, Inc.

www.zenoss.com

Upgrade Guide for High-Availability Deployments

Part Number: 1950.20.160.53

Zenoss, Inc.
11305 Four Points Drive
Bldg 1 - Suite 300
Austin, Texas 78726

Contents

About this guide 6
- Documented upgrade paths 6
- Documentation feedback 6
- Change history 6

Chapter 1: Downloading and staging required files 7
- Downloading required files 7
- Installing the repository mirror 9
- Staging Docker image files 10
- Staging a Docker image file on ZooKeeper ensemble nodes 11

Part I: Upgrading 1.6.5 to 1.7.0 12
- Scope 12

Chapter 2: Before upgrading 1.6.5 to 1.7.0 13
- New features that affect upgrades 13
- Understanding DRBD 8.4 kernel dependencies 13
- Upgrade best practices 13

Chapter 3: Stopping a Control Center deployment 15
- Stopping a master host node 15
- Stopping a delegate host 16

Chapter 4: Upgrading master hosts from 1.6.5 to 1.7.0 19
- Updating Control Center on the master host 19

Chapter 5: Upgrading delegates from 1.5.0 to 1.7.0 21
- Updating Control Center on delegate hosts 21

Part II: Upgrading 1.5.1 or 1.6.3 to 1.7.0 23
- Scope 23

Chapter 6: Before upgrading 1.5.1 or 1.6.3 to 1.7.0 24
- New features that affect upgrades 24
- Understanding DRBD 8.4 kernel dependencies 24
- Upgrade best practices 24

Chapter 7: Stopping a Control Center deployment 26
- Stopping a master host node 26
- Stopping a delegate host 27

Chapter 8: Upgrading master hosts from 1.5.1 or 1.6.3 to 1.7.0..... 30
Updating Docker..... 30
Loading image files..... 31
Updating Control Center on the master host..... 32

Chapter 9: Upgrading delegates from 1.5.1 or 1.6.3 to 1.7.0..... 33
Updating Docker..... 33
Configuring NFS 4.0..... 34
Updating Control Center on delegate hosts..... 35
Updating the ZooKeeper image on ensemble nodes..... 36

Chapter 10: After upgrading..... 37
Removing unused images..... 37

Part III: Upgrading Resource Manager..... 38
Scope..... 38

Chapter 11: Before upgrading a high-availability deployment..... 39
Importing Resource Manager image files..... 39
Managing custom MariaDB passwords..... 40

Chapter 12: Upgrading Resource Manager..... 42
Stopping Resource Manager..... 42
Upgrading Resource Manager..... 42

Chapter 13: After upgrading a high-availability deployment..... 44
Removing the pre-upgrade snapshot..... 44
Moving MariaDB services into the master resource pool..... 44
Clearing heartbeat events..... 45

Appendix A: Starting and stopping Control Center deployments..... 46
Stopping Control Center..... 46
Starting Control Center..... 49

Appendix B: Resolving package dependency conflicts..... 51
Resolving device mapper dependency conflicts..... 51
Resolving other dependency conflicts..... 52

Appendix C: Updating the cluster management software..... 54
Downloading and staging cluster software..... 54
Updating cluster software..... 55

Appendix D: Installing the Resource Manager application template.....57
Downloading and installing the template package..... 57

About this guide

The *Upgrade Guide for High-Availability Deployments* provides detailed instructions for upgrading a high-availability deployment of recent versions of Control Center and Resource Manager. To perform the procedures in this guide, users must be able to download packages from the Zenoss Delivery site, which is available only to Zenoss customers.

Documented upgrade paths

Table 1: Control Center upgrade paths

From	To
Control Center 1.6.5	Control Center 1.7.0
Control Center 1.6.3	Control Center 1.7.0
Control Center 1.5.1	Control Center 1.7.0

For information about updating earlier releases of Control Center, please refer to the *Control Center Upgrade Guide for High-Availability Deployments (Release 1.5.1)*.

Table 2: Resource Manager upgrade paths

From	To
Resource Manager 6.4.1	Resource Manager 6.5.0
Resource Manager 6.4.0	Resource Manager 6.5.0
Resource Manager 6.3.2	Resource Manager 6.5.0

For information about updating earlier releases of Resource Manager, please refer to the *Zenoss Resource Manager (Release 6.3.2)*.

Documentation feedback

To provide feedback about this document, or to report an error or omission, please send an email to `docs@controlcenter.io`. In the email, please include the document title (*Upgrade Guide for High-Availability Deployments*) and part number (1950.20.160.53) and as much information as possible about the context of your feedback.

Change history

The following list associates document part numbers and the important changes to this guide since the previous release. Some of the changes involve features or content, but others do not.

1950.20.160.53 (6.5.0)

New publication combining Control Center and Resource Manager procedures.

1 Downloading and staging required files

This chapter describes how to download and install or stage required software and operating system dependencies. The procedures in this chapter are required to perform an upgrade.

Note The Pacemaker resource agents for Control Center are unchanged for this release.

The following table identifies where to perform each procedure in this chapter.

Procedure	Where to perform
Downloading required files on page 7	A workstation with internet access
Installing the repository mirror on page 9	All Control Center hosts
Staging Docker image files on page 10	Both master host nodes
Staging a Docker image file on ZooKeeper ensemble nodes on page 11	Delegate hosts that are ZooKeeper ensemble nodes

Downloading required files

To perform this procedure, you need:

- A workstation with internet access.
- Permission to download files from *delivery.zenoss.com*. Customers can request permission by filing a ticket at the *Zenoss Support* site.
- A secure network copy program.

Use this procedure to

- download the required files to a workstation
- copy the files to the hosts that need them

Perform these steps:

1. In a web browser, navigate to the download site, and then log in.
   The download site is *delivery.zenoss.com*.
2. Download the self-installing Docker image files.

Note These files are not needed to update release 1.6.5 to 1.7.0. For more information, refer to *Control Center releases and image tags*.

```
install-zenoss-serviced-isvcs-v63.run
install-zenoss-isvcs-zookeeper-v11.run
```

3 Download the Control Center RPM file.

```
serviced-1.7.0-1.x86_64.rpm
```

4 Identify the operating system release on Control Center hosts.

Enter the following command on each Control Center host in your deployment, if necessary. All Control Center hosts should be running the same operating system release and kernel.

```
cat /etc/redhat-release
```

5 Download the RHEL/CentOS repository mirror file for your deployment.

The download site provides repository mirror files containing the packages that Control Center requires. For RHEL/CentOS 7.7 and 7.8, use the 7.6 file.

```
yum-mirror-centos7.2-1511-serviced-1.7.0.x86_64.rpm
yum-mirror-centos7.3-1611-serviced-1.7.0.x86_64.rpm
yum-mirror-centos7.4-1708-serviced-1.7.0.x86_64.rpm
yum-mirror-centos7.5-1708-serviced-1.7.0.x86_64.rpm
yum-mirror-centos7.6-1708-serviced-1.7.0.x86_64.rpm
```

Note Currently, Red Hat Enterprise Linux (RHEL) 8.x and CentOS 8.x are unsupported.

6 Download the self-installing Docker image files for Resource Manager.

- `install-zenoss-hbase-24.0.8.run`
- `install-zenoss-opentsdb-24.0.8.run`
- `install-zenoss-resmgr_6.5:6.5.0_1.run`
- `install-zenoss-mariadb-resmgr:6.5.0.run`

7 Optional: Download the Zenoss GNU Privacy Guard (GPG) key, if desired.

You can use the Zenoss GPG key to verify Zenoss RPM files and the `yum` metadata of the repository mirror.

a Download the key.

```
curl --location -o /tmp/tmp.html \
'http://keys.gnupg.net/pks/lookup?op=get&search=0xED0A5FD2AA5A1AD7'
```

b Determine whether the download succeeded.

```
grep -Ec '^\-\-\-\-\-BEGIN PGP' /tmp/tmp.html
```

- If the result is `0`, return to the previous substep.
- If the result is `1`, proceed to the next substep.

c Extract the key.

```
awk '/^-----BEGIN PGP.*$/,/^-----END PGP.*$/' \
  /tmp/tmp.html > ./RPM-GPG-KEY-Zenoss
```

8 Use a secure copy program to copy the files to Control Center hosts.

- Copy all files to both master nodes. This guide relies on the following convention for file locations:
  - Place Control Center files in `/tmp`.
  - Place Resource Manager files in `/tmp/rm`.
- Copy the RHEL/CentOS RPM file, the Control Center RPM file, and the Zenoss GPG key file to all delegate hosts.
- Copy the Docker image file for ZooKeeper to delegate hosts that are ZooKeeper ensemble nodes.

Installing the repository mirror

Use this procedure to install the Zenoss repository mirror on a Control Center host. Repeat this procedure on each host in your deployment.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Move the RPM files and the Zenoss GPG key file to `/tmp`.
3. Optional: Remove the existing repository mirror, if necessary.
   a. Search for the mirror.

   ```
   yum list --disablerepo=* | awk '/^yum-mirror/ { print $1}'
   ```

   b. Remove the mirror.
   Replace *Old-Mirror* with the name of the Zenoss repository mirror returned in the previous substep:

   ```
   yum remove Old-Mirror
   ```

4. Install the repository mirror.

   ```
   yum install /tmp/yum-mirror-*.rpm
   ```

   The `yum` command copies the contents of the RPM file to `/opt/zenoss-repo-mirror`.
5. Optional: Install the Zenoss GPG key, and then test the package files, if desired.
   a. Move the Zenoss GPG key to the mirror directory.

   ```
   mv /tmp/RPM-GPG-KEY-Zenoss /opt/zenoss-repo-mirror
   ```

   b. Install the key.

   ```
   rpm --import /opt/zenoss-repo-mirror/RPM-GPG-KEY-Zenoss
   ```

   c. Test the repository mirror package file.

   ```
   rpm -K /tmp/yum-mirror-*.rpm
   ```

   On success, the result includes the file name and the following information:

   ```
   (sha1) dsa sha1 md5 gpg OK
   ```

   d. Test the Control Center package file.

   ```
   rpm -K /tmp/serviced-1.7.0-1.x86_64.rpm
   ```

6. Optional: Update the configuration file of the Zenoss repository mirror to enable GPG key verification, if desired.

a. Open the repository mirror configuration file (`/etc/yum.repos.d/zenoss-mirror.repo`) with a text editor, and then add the following lines to the end of the file.

```
repo_gpgcheck=1
gpgkey=file:///opt/zenoss-repo-mirror/RPM-GPG-KEY-Zenoss
```

b. Save the file, and then close the editor.

c. Update the `yum` metadata cache.

```
yum makecache fast
```

The cache update process includes the following prompt:

```
Retrieving key from file:///opt/zenoss-repo-mirror/RPM-GPG-KEY-
Zenoss
Importing GPG key 0xAA5A1AD7:
 Userid     : "Zenoss, Inc. <dev@zenoss.com>"
 Fingerprint: f31f fd84 6a23 b3d5 981d a728 ed0a 5fd2 aa5a 1ad7
 From       : /opt/zenoss-repo-mirror/RPM-GPG-KEY-Zenoss
Is this ok [y/N]:
```

Enter `y`.

7. Move the Control Center package file to the mirror directory.

```
mv /tmp/serviced-1.7.0-1.x86_64.rpm /opt/zenoss-repo-mirror
```

8. Optional: Delete the mirror package file, if desired.

```
rm /tmp/yum-mirror-*.rpm
```

Staging Docker image files

Before performing this procedure, verify that approximately 2GB of temporary space is available on the file system where `/root` is located.

Use this procedure to stage Docker image files on a Control Center host. The files are used when Docker is fully configured. Perform this procedure on both master hosts.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Move Control Center files to `/root/cc`.
   a. Create the target directory.

```
mkdir /root/cc
```

   b. Move the files.

```
mv /tmp/*.run /root/cc
```

   NOTE: Subsequent procedures rely on the `/root/cc` location.
3. Move Resource Manager files to `/root/rm`.
   a. Create the target directory.

```
mkdir /root/rm
```

b Move the files.

```
mv /tmp/rm/*.run /root/rm
```

NOTE: Subsequent procedures rely on the `/root/rm` location.

4 Add execute permission to the files.

```
chmod +x /root/cc/*.run ; chmod +x /root/rm/*.run
```

Staging a Docker image file on ZooKeeper ensemble nodes

Before performing this procedure, verify that approximately 170MB of temporary space is available on the file system where `/root` is located.

Use this procedure to add a Docker image file to the Control Center delegate hosts that are ZooKeeper ensemble nodes. Delegate hosts that are not ZooKeeper ensemble nodes do not need the file.

1 Log in to a delegate host as `root`, or as a user with superuser privileges.

2 Copy or move the `install-zenoss-isvcs-zookeeper-v11.run` file to `/root`.

3 Add execute permission to the file.

```
chmod +x /root/*.run
```

Upgrading 1.6.5 to 1.7.0
Scope

The chapters in this part provide instructions for performing an upgrade of a high-availability deployment of Control Center, from version 1.6.5 to 1.7.0.

2 Before upgrading 1.6.5 to 1.7.0

This chapter provides information and procedures to prepare your high-availability Control Center deployment for an upgrade.

New features that affect upgrades

This release includes no new features or requirements that affect the upgrade process.

Note The Docker image files for Control Center are unchanged for this release. You only need to download and install the new `serviced` RPM package.

Understanding DRBD 8.4 kernel dependencies

In RHEL/CentOS, Distributed Replicated Block Device (DRBD) is a kernel module. When Resource Manager 6.5.0 was released, the module available on the Enterprise Linux packages repository (ELRepo) was for Linux kernel 3.10.0-1127. You must ensure that the version of DRBD you install works with the kernel you are using on both Control Center master hosts.

Upgrade best practices

The following list outlines recommended best practices for upgrading high-availability Control Center deployments:

1. Upgrade only what needs upgrading. For example, you can update DRBD and Pacemaker/Corosync independently. For more information, see *Updating the cluster management software* on page 54.
2. Compare the Docker images that accompany this release with the images that accompany the installed release, and determine whether the image files need to be downloaded and installed. For more information, refer to *Control Center releases and image tags*.
3. The contents of `/etc/default/serviced` on the master nodes must be identical. Use a utility like `sum` to compare the files quickly.
4. On delegate hosts, most of the upgrade steps are identical. Use `screen`, `tmux`, or a similar program to establish sessions on each delegate host and perform the steps at the same time.
5. Review and verify the settings in delegate host configuration files (`/etc/default/serviced`) before starting the upgrade. Ideally, the settings on all delegate hosts are identical, except on ZooKeeper nodes and delegate hosts that do not mount the DFS.

6 Review the procedures in this guide before performing them. Every effort is made to avoid mistakes and anticipate needs; nevertheless, the instructions may be incorrect or inadequate for some requirements or environments.

7 Download and stage the required files for your upgrade. For more information, see *Downloading and staging required files* on page 7.

3 Stopping a Control Center deployment

This chapter includes procedures for stopping a high-availability Control Center deployment.

Note The procedures in this chapter assume that Control Center is the only source of Docker containers that are run on Control Center hosts.

Perform the procedures in this chapter in order.

Stopping a master host node

Use this procedure to stop the Control Center service (`serviced`) on the master host in a high-availability deployment.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as `root`, or as a user with superuser privileges.
2. Display the public hostname of the current node.

   ```
   uname -n
   ```

   Make a note of which node (primary or secondary) is the current node, for use in a subsequent step.
3. Stop the top-level service `serviced` is managing, if necessary.
   a. Show the status of running services.

      ```
      serviced service status
      ```

      The top-level service is the service listed immediately below the headings line.

      - If the status of the top-level service and all child services is `stopped`, proceed to the next step.
      - If the status of the top-level service and all child services is **not** `stopped`, perform the remaining substeps.

   b. Stop the top-level service.
      Replace *Service* with the name or identifier of the top-level service:

      ```
      serviced service stop Service
      ```

   c. Monitor the stop.

      ```
      serviced service status
      ```

When the status of the top-level service and all child services is `stopped`, proceed to the next step.

4 Stop Control Center with the cluster management tool.

```
pcs cluster standby --all
```

5 Monitor the status of cluster resources.

```
watch pcs status
```

Monitor the status until all resources report `Stopped`. Resolve any issues before continuing.

6 Ensure that no containers remain in the local repository.

a Start the Docker service.

```
systemctl start docker
```

b Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

If the command returns a result, enter the following command:

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

c Stop the Docker service.

```
systemctl stop docker
```

7 To ensure that no containers remain in both Docker repositories, log in to the other master node as `root`, or as a user with superuser privileges, and then perform the preceding step.

Stopping a delegate host

Use this procedure to stop the Control Center service (`serviced`) on a delegate host in a multi-host deployment. Repeat this procedure on each delegate host in your deployment.

1 Log in to the delegate host as `root`, or as a user with superuser privileges.

2 Stop the Control Center service.

```
systemctl stop serviced
```

3 Ensure that no containers remain in the local repository.

a Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

- If the command returns no result, proceed to the next step.
- If the command returns a result, perform the following substeps.

b Remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, the most likely cause is an NFS conflict. Perform the following substeps.

c Stop the NFS and Docker services.

```
systemctl stop nfs && systemctl stop docker
```

d Start the NFS and Docker services.

```
systemctl start nfs && systemctl start docker
```

e Repeat the attempt to remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, perform the remaining substeps.

f Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

g Reboot the host.

```
reboot
```

h Log in to the delegate host as `root`, or as a user with superuser privileges.

i Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

4 Dismount all filesystems mounted from the Control Center master host.

This step ensures no stale mounts remain when the storage on the master host is replaced.

a Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the following substeps.

b Force the filesystems to dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f $FS
done
```

c Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the following substeps.

d Perform a lazy dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f -l $FS
done
```

e Restart the NFS service.

```
systemctl restart nfs
```

f Determine whether any filesystems remain mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the remaining substeps.

g Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

h Reboot the host.

```
reboot
```

i Log in to the delegate host as `root`, or as a user with superuser privileges.

j Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

4 Upgrading master hosts from 1.6.5 to 1.7.0

Perform the procedures in this chapter to upgrade both Control Center 1.6.5 master hosts to 1.7.0. First, download and stage the required files. For more information, see *Downloading and staging required files* on page 7.

Note The Docker image files for Control Center are unchanged for this release. You only need to download and install the new `serviced` RPM package.

Updating Control Center on the master host

Use this procedure to update Control Center on the master host to version 1.7.0.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Save the current `serviced` configuration file as a reference.
   a. Rename the file.

```
mv /etc/default/serviced /etc/default/serviced-pre-1.7.0
```

   b. Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-pre-1.7.0
```

3. Install the new version of Control Center.

```
yum install --enablerepo=zenoss-mirror \
  /opt/zenoss-repo-mirror/serviced-1.7.0-1.x86_64.rpm
```

4. Make a backup copy of the new configuration file.
   a. Copy the file.

```
cp /etc/default/serviced /etc/default/serviced-1.7.0-orig
```

   b. Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-1.7.0-orig
```

5 Display the settings of the reference configuration file.

```
grep -E '^\b*[A-Z_]+' /etc/default/serviced-pre-1.7.0
```

6 Open the new configuration file with a text editor, and then update the file for your environment.
For more information about configuring the master host, refer to *Control Center configuration variables*.

5 Upgrading delegates from 1.5.0 to 1.7.0

Perform the procedures in this chapter to upgrade Control Center 1.6.5 delegate hosts to 1.7.0. First, download and stage the required files. For more information, see *Downloading and staging required files* on page 7.

Note The Docker image files for Control Center are unchanged for this release. You only need to download and install the new `serviced` RPM package.

Updating Control Center on delegate hosts

This procedure updates Control Center on delegate hosts to version 1.7.0.

Perform this procedure on each delegate host in your deployment.

1. Log in to a delegate host as `root`, or as a user with superuser privileges.
2. Save the current `serviced` configuration file as a reference.
   a. Rename the file.

   ```
   mv /etc/default/serviced /etc/default/serviced-pre-1.7.0
   ```

   b. Set permissions to read-only.

   ```
   chmod 0440 /etc/default/serviced-pre-1.7.0
   ```

3. Install the new version of Control Center.

   ```
   yum install --enablerepo=zenoss-mirror \
     /opt/zenoss-repo-mirror/serviced-1.7.0-1.x86_64.rpm
   ```

   If `yum` returns an error due to dependency issues, see *Resolving package dependency conflicts* on page 51 for potential resolutions.
4. Make a backup copy of the new configuration file.
   a. Copy the file.

   ```
   cp /etc/default/serviced /etc/default/serviced-1.7.0-orig
   ```

   b. Set permissions to read-only.

   ```
   chmod 0440 /etc/default/serviced-1.7.0-orig
   ```

5 Display the settings of the reference configuration file.

```
grep -E '^\b*[A-Z_]+' /etc/default/serviced-pre-1.7.0
```

6 Open the new configuration file with a text editor, and then update the file for your environment.

For more information about configuring a delegate host, refer to *Control Center configuration variables*.

When all delegate hosts are updated, start the cluster. For more information, see *Starting and stopping Control Center deployments* on page 46.

Upgrading 1.5.1 or 1.6.3 to 1.7.0

Scope

The chapters in this part provide instructions for performing an upgrade of a high-availability deployment of Control Center, from version 1.5.1 or 1.6.3 to 1.7.0.

6 Before upgrading 1.5.1 or 1.6.3 to 1.7.0

This chapter provides information and procedures to prepare your high-availability Control Center deployment for an upgrade.

New features that affect upgrades

This release includes new features and new requirements that affect the upgrade process. The following list provides an overview of the changes that are addressed during this upgrade.

- On RHEL/CentOS 7.4 and above, there may be a file locking defect in NFS 4.1. To avoid the issue, delegate hosts are configured to use NFS 4.0.
- This release replaces Docker Community Edition (CE) 17.09.0 with Docker CE 18.09.6.

Note The Pacemaker resource agents for Control Center are unchanged for this release.

Understanding DRBD 8.4 kernel dependencies

In RHEL/CentOS, Distributed Replicated Block Device (DRBD) is a kernel module. When Resource Manager 6.5.0 was released, the module available on the Enterprise Linux packages repository (ELRepo) was for Linux kernel 3.10.0-1127. You must ensure that the version of DRBD you install works with the kernel you are using on both Control Center master hosts.

Upgrade best practices

The following list outlines recommended best practices for upgrading high-availability Control Center deployments:

1. Upgrade only what needs upgrading. For example, you can update DRBD and Pacemaker/Corosync independently. For more information, see *Updating the cluster management software* on page 54.
2. Compare the Docker images that accompany this release with the images that accompany the installed release, and determine whether the image files need to be downloaded and installed. For more information, refer to *Control Center releases and image tags*.
3. The contents of `/etc/default/serviced` on the master nodes must be identical. Use a utility like `sum` to compare the files quickly.
4. On delegate hosts, most of the upgrade steps are identical. Use `screen`, `tmux`, or a similar program to establish sessions on each delegate host and perform the steps at the same time.

5 Review and verify the settings in delegate host configuration files (`/etc/default/serviced`) before starting the upgrade. Ideally, the settings on all delegate hosts are identical, except on ZooKeeper nodes and delegate hosts that do not mount the DFS.
6 Review the procedures in this guide before performing them. Every effort is made to avoid mistakes and anticipate needs; nevertheless, the instructions may be incorrect or inadequate for some requirements or environments.
7 Download and stage the required files for your upgrade. For more information, see *Downloading and staging required files* on page 7.

7 Stopping a Control Center deployment

This chapter includes procedures for stopping a high-availability Control Center deployment.

Note The procedures in this chapter assume that Control Center is the only source of Docker containers that are run on Control Center hosts.

Perform the procedures in this chapter in order.

Stopping a master host node

Use this procedure to stop the Control Center service (`serviced`) on the master host in a high-availability deployment.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as `root`, or as a user with superuser privileges.
2. Display the public hostname of the current node.

   ```
   uname -n
   ```

   Make a note of which node (primary or secondary) is the current node, for use in a subsequent step.
3. Stop the top-level service `serviced` is managing, if necessary.
   a. Show the status of running services.

      ```
      serviced service status
      ```

      The top-level service is the service listed immediately below the headings line.

      - If the status of the top-level service and all child services is `stopped`, proceed to the next step.
      - If the status of the top-level service and all child services is **not** `stopped`, perform the remaining substeps.

   b. Stop the top-level service.
      Replace *Service* with the name or identifier of the top-level service:

      ```
      serviced service stop Service
      ```

   c. Monitor the stop.

      ```
      serviced service status
      ```

When the status of the top-level service and all child services is `stopped`, proceed to the next step.

4 Stop Control Center with the cluster management tool.

```
pcs cluster standby --all
```

5 Monitor the status of cluster resources.

```
watch pcs status
```

Monitor the status until all resources report `Stopped`. Resolve any issues before continuing.

6 Ensure that no containers remain in the local repository.

a Start the Docker service.

```
systemctl start docker
```

b Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

If the command returns a result, enter the following command:

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

c Stop the Docker service.

```
systemctl stop docker
```

7 To ensure that no containers remain in both Docker repositories, log in to the other master node as `root`, or as a user with superuser privileges, and then perform the preceding step.

Stopping a delegate host

Use this procedure to stop the Control Center service (`serviced`) on a delegate host in a multi-host deployment. Repeat this procedure on each delegate host in your deployment.

1 Log in to the delegate host as `root`, or as a user with superuser privileges.

2 Stop the Control Center service.

```
systemctl stop serviced
```

3 Ensure that no containers remain in the local repository.

a Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

- If the command returns no result, proceed to the next step.
- If the command returns a result, perform the following substeps.

b Remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, the most likely cause is an NFS conflict. Perform the following substeps.

c Stop the NFS and Docker services.

```
systemctl stop nfs && systemctl stop docker
```

d Start the NFS and Docker services.

```
systemctl start nfs && systemctl start docker
```

e Repeat the attempt to remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, perform the remaining substeps.

f Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

g Reboot the host.

```
reboot
```

h Log in to the delegate host as `root`, or as a user with superuser privileges.

i Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

4 Dismount all filesystems mounted from the Control Center master host.

This step ensures no stale mounts remain when the storage on the master host is replaced.

a Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the following substeps.

b Force the filesystems to dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f $FS
done
```

c Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the following substeps.

d Perform a lazy dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f -l $FS
done
```

e Restart the NFS service.

```
systemctl restart nfs
```

f Determine whether any filesystems remain mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the remaining substeps.

g Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

h Reboot the host.

```
reboot
```

i Log in to the delegate host as `root`, or as a user with superuser privileges.

j Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

Upgrading master hosts from 1.5.1 or 1.6.3 to 1.7.0

8

Perform the procedures in this chapter to upgrade both Control Center 1.5.1 or 1.6.3 master hosts to 1.7.0. First, download and stage the required files. For more information, see *Downloading and staging required files* on page 7.

Updating Docker

Use this procedure to update Docker to version 18.09.6.

1. Log in as `root`, or as a user with superuser privileges.
2. Update the operating system, if necessary.
   a. Determine which release is installed.

      ```
      cat /etc/redhat-release
      ```

      - If the result is greater than `7.2`, proceed to the next step.
      - If the result is `7.1`, perform the remaining substeps.

   b. Disable automatic start of `serviced`.

      ```
      systemctl disable serviced
      ```

   c. Update the operating system, and then restart the host.

      The following commands require internet access or a local mirror of operating system packages.

      ```
      yum makecache fast && yum update && reboot
      ```

   d. Log in as `root`, or as a user with superuser privileges.
   e. Enable automatic start of `serviced`.

      ```
      systemctl enable serviced
      ```

3. Update the Linux kernel, if necessary.
   a. Determine which kernel version is installed.

      ```
      uname -r
      ```

      If the result is lower than `3.10.0-327.22.2.el7.x86_64`, perform the following substeps.

b Disable automatic start of `serviced`.

```
systemctl disable serviced
```

c Update the kernel, and then restart the host.

Note You may need to install a specific kernel, rather than update to the latest kernel. For more information, see *Understanding DRBD 8.4 kernel dependencies* on page 13.

The following commands require internet access or a local mirror of operating system packages.

```
yum makecache fast && yum update kernel && reboot
```

d Log in as `root`, or as a user with superuser privileges.

e Enable automatic start of `serviced`.

```
systemctl enable serviced
```

4 Stop the Docker service.

```
systemctl stop docker
```

5 Remove Docker 17.09.0.

a Remove without checking dependencies.

```
rpm -e --nodeps docker-ce
```

b Clean the `yum` databases.

```
yum clean all
```

6 Install Docker CE 18.09.6.

```
yum install --enablerepo=zenoss-mirror docker-ce-18.09.6-3.el7
```

If `yum` returns an error due to dependency issues, see *Resolving package dependency conflicts* on page 51 for potential resolutions.

7 Start the Docker service.

```
systemctl start docker
```

Loading image files

Use this procedure to load images into the local registry.

1 Log in to the master host as `root`, or as a user with superuser privileges.

2 Start the Docker service.

```
systemctl start docker
```

3 Change directory to `/root`.

```
cd /root
```

4 Load the images.

```
for image in install-zenoss-*.run
do
  /bin/echo -en "\nLoading $image..."
  yes | ./$image
done
```

5 List the images in the registry.

```
docker images
```

The result should show one image for each archive file.

6 Optional: Delete the archive files, if desired.

```
rm -i ./install-zenoss-*.run
```

7 Stop the Docker service.

```
systemctl stop docker
```

Updating Control Center on the master host

Use this procedure to update Control Center on the master host to version 1.7.0.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Save the current `serviced` configuration file as a reference.
   a Rename the file.

```
mv /etc/default/serviced /etc/default/serviced-pre-1.7.0
```

   b Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-pre-1.7.0
```

3 Install the new version of Control Center.

```
yum install --enablerepo=zenoss-mirror \
  /opt/zenoss-repo-mirror/serviced-1.7.0-1.x86_64.rpm
```

4 Make a backup copy of the new configuration file.
   a Copy the file.

```
cp /etc/default/serviced /etc/default/serviced-1.7.0-orig
```

   b Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-1.7.0-orig
```

5 Display the settings of the reference configuration file.

```
grep -E '^\b*[A-Z_]+' /etc/default/serviced-pre-1.7.0
```

6 Open the new configuration file with a text editor, and then update the file for your environment.
For more information about configuring the master host, refer to *Control Center configuration variables*.

Upgrading delegates from 1.5.1 or 1.6.3 to 1.7.0

9

Perform the procedures in this chapter to upgrade Control Center 1.5.1 or 1.6.3 delegate hosts to 1.7.0. First, download and stage the required files. For more information, see *Downloading and staging required files* on page 7.

Updating Docker

Use this procedure to update Docker to version 18.09.6.

1. Log in as `root`, or as a user with superuser privileges.
2. Update the operating system, if necessary.
   a. Determine which release is installed.

      ```
      cat /etc/redhat-release
      ```

      - If the result is greater than `7.2`, proceed to the next step.
      - If the result is `7.1`, perform the remaining substeps.

   b. Disable automatic start of `serviced`.

      ```
      systemctl disable serviced
      ```

   c. Update the operating system, and then restart the host.

      The following commands require internet access or a local mirror of operating system packages.

      ```
      yum makecache fast && yum update && reboot
      ```

   d. Log in as `root`, or as a user with superuser privileges.
   e. Enable automatic start of `serviced`.

      ```
      systemctl enable serviced
      ```

3. Update the Linux kernel, if necessary.
   a. Determine which kernel version is installed.

      ```
      uname -r
      ```

      If the result is lower than `3.10.0-327.22.2.el7.x86_64`, perform the following substeps.

b Disable automatic start of `serviced`.

```
systemctl disable serviced
```

c Update the kernel, and then restart the host.

Note You may need to install a specific kernel, rather than update to the latest kernel. For more information, see *Understanding DRBD 8.4 kernel dependencies* on page 13.

The following commands require internet access or a local mirror of operating system packages.

```
yum makecache fast && yum update kernel && reboot
```

d Log in as `root`, or as a user with superuser privileges.

e Enable automatic start of `serviced`.

```
systemctl enable serviced
```

4 Stop the Docker service.

```
systemctl stop docker
```

5 Remove Docker 17.09.0.

a Remove without checking dependencies.

```
rpm -e --nodeps docker-ce
```

b Clean the `yum` databases.

```
yum clean all
```

6 Install Docker CE 18.09.6.

```
yum install --enablerepo=zenoss-mirror docker-ce-18.09.6-3.el7
```

If `yum` returns an error due to dependency issues, see *Resolving package dependency conflicts* on page 51 for potential resolutions.

7 Start the Docker service.

```
systemctl start docker
```

Configuring NFS 4.0

Use this procedure to configure NFS 4.0 on delegate hosts if the operating system release is 7.4. There may be a file locking defect in NFS 4.1 with RHEL/CentOS 7.4 and above.

1 Log in to the host as `root`, or as a user with superuser privileges.

2 Determine which release is installed.

```
cat /etc/redhat-release
```

- If the result includes `7.4` or higher, perform the remaining steps of this procedure.
- If the result includes `7.2` or `7.3`, continue to the next procedure.

3 Change the NFS configuration file.

a. Open `/etc/nfsmount.conf` with a text editor.
b. Locate the `Defaultvers` directive.
c. Remove the number sign character (`#`) from the beginning of the line.
d. Change the value from `4` to `4.0`.
   The line should appear as follows:

```
Defaultvers=4.0
```

e. Save the file, and then close the editor.

4. Restart the NFS server.

```
systemctl restart nfs-server
```

Updating Control Center on delegate hosts

This procedure updates Control Center on delegate hosts to version 1.7.0.

Perform this procedure on each delegate host in your deployment.

1. Log in to a delegate host as `root`, or as a user with superuser privileges.
2. Save the current `serviced` configuration file as a reference.
   a. Rename the file.

```
mv /etc/default/serviced /etc/default/serviced-pre-1.7.0
```

   b. Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-pre-1.7.0
```

3. Install the new version of Control Center.

```
yum install --enablerepo=zenoss-mirror \
  /opt/zenoss-repo-mirror/serviced-1.7.0-1.x86_64.rpm
```

   If `yum` returns an error due to dependency issues, see *Resolving package dependency conflicts* on page 51 for potential resolutions.
4. Make a backup copy of the new configuration file.
   a. Copy the file.

```
cp /etc/default/serviced /etc/default/serviced-1.7.0-orig
```

   b. Set permissions to read-only.

```
chmod 0440 /etc/default/serviced-1.7.0-orig
```

5. Display the settings of the reference configuration file.

```
grep -E '^\b*[A-Z_]+' /etc/default/serviced-pre-1.7.0
```

6. Open the new configuration file with a text editor, and then update the file for your environment.
   For more information about configuring a delegate host, refer to *Control Center configuration variables*.

When all delegate hosts are updated, start the cluster. For more information, see *Starting and stopping Control Center deployments* on page 46.

Updating the ZooKeeper image on ensemble nodes

Use this procedure to install a new Docker image for ZooKeeper into the local repository of delegate hosts that are ZooKeeper ensemble nodes.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Identify the hosts in the ZooKeeper ensemble.

   ```
   grep -E '^\b*SERVICED_ZK=' /etc/default/serviced
   ```

   The result is a list of 3 or 5 hosts, separated by the comma character (`,`). The master host is always a node in the ZooKeeper ensemble.
3. Log in to a ZooKeeper ensemble node as `root`, or as a user with superuser privileges.
4. Start the Docker service.

   ```
   systemctl start docker
   ```
5. Change directory to `/root`.

   ```
   cd /root
   ```
6. Extract the ZooKeeper image.

   ```
   ./install-zenoss-isvcs-zookeeper-v11.run
   ```

   At the prompt, press **y**.
7. Optional: Delete the archive file, if desired.

   ```
   rm -i ./install-zenoss-isvcs-zookeeper-v11.run
   ```
8. Repeat the preceding five steps on each ZooKeeper node in the ensemble.

10 After upgrading

Perform the procedures in this chapter after Control Center is upgraded.

Removing unused images

Use this procedure to identify and remove unused Control Center images.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Identify the images associated with the installed version of `serviced`.

   ```
   serviced version | grep Images
   ```

   Example result:

   ```
   IsvcsImages: [zenoss/serviced-isvcs:v63 zenoss/isvcs-zookeeper:v11]
   ```
3. Start Docker, if necessary.

   ```
   systemctl status docker || systemctl start docker
   ```
4. Display the `serviced` images in the local repository.

   ```
   docker images | awk '/REPO|isvcs/'
   ```

   Example result (edited to fit):

   ```
   REPOSITORY                      TAG       IMAGE ID
   zenoss/serviced-isvcs           v40       88cd6c24cc82
   zenoss/serviced-isvcs           v63       0aab5a2123f2
   zenoss/isvcs-zookeeper          v3        46fa0a2fc4bf
   zenoss/isvcs-zookeeper          v11       0ff3b3117fb8
   ```

   The example result shows the current versions and one set of previous versions. Your result may include additional previous versions and will show different images IDs.
5. Remove unused images.
   Replace *Image-ID* with the image ID of an image for a previous version.

   ```
   docker rmi Image-ID
   ```

   Repeat this command for each unused image.

Upgrading Resource Manager Scope

The chapters in this part describe how to upgrade high-availability deployments of Resource Manager. The procedures in this part are valid for all of the upgrade paths listed in *Documented upgrade paths* on page 6.

This part does not include information about updating Control Center. For more information, see one of the preceding parts of this document.

Before upgrading a high-availability deployment 11

Use the procedures in this chapter to prepare your high-availability Resource Manager deployment for an upgrade.

Importing Resource Manager image files

Use this procedure to import Resource Manager image from self-installing archive files.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as a user with `serviced` CLI privileges.
2. In a separate terminal session, log in to the other master host as `root`, or as a user with superuser privileges.
3. On both nodes, change directory to `/root/rm`.

```
cd /root/rm
```

4. On the secondary node, start Docker.

```
systemctl start docker
```

5. On both nodes, import the images.

```
for image in install-zenoss-*.run
do
  /bin/echo -en "\nLoading $image..."
  yes | ./$image
done
```

6. On both nodes, list the images in the registry.

```
docker images
```

   The result should include one image for each archive file.
7. Optional: On both nodes, delete the archive files, if desired.

```
rm -i ./install-*.run
```

8. On both nodes, copy the upgrade scripts from the new Resource Manager image to `/root/6.5.x`.

```
docker run -it --rm -v /root:/mnt/root \
```

```
  zenoss/resmgr_6.5:6.5.0_1 rsync -a /root/6.5.x /mnt/root
```

9 On the secondary node, stop Docker.

```
systemctl stop docker
```

Managing custom MariaDB passwords

If you are using custom database passwords for the `mariadb-events` and `mariadb-model` services, the passwords must be managed before starting the update process for this release. To determine whether custom passwords are in use, follow these steps:

1 Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as a user with `serviced` CLI privileges.
2 Display the MariaDB passwords.

```
serviced service list Zenoss.resmgr | awk '/(zep|zodb).*-password/'
```

When the default passwords are in use, the result looks like this:

```
"global.conf.zep-admin-password": "",
"global.conf.zep-password": "zenoss",
"global.conf.zodb-admin-password": "",
"global.conf.zodb-password": "zenoss",
```

To perform this procedure, you need:

- A workstation with internet access.
- Permission to download files from *delivery.zenoss.com*. Customers can request permission by filing a ticket at the *Zenoss Support* site.
- A secure network copy program.

Note In addition, Resource Manager must be running normally.

To maintain customized MariaDB passwords before updating Resource Manager, follow these steps:

1 In a web browser, navigate to *delivery.zenoss.com*, and then log in.
2 Download the `addGlobalConfToMariadb.py` script.

The script is located in the `resource-manager-65x` folder.

3 Use a network copy utility to copy the script to the Control Center master node.
4 Log in to the Control Center master node as a user with `serviced` CLI privileges.
5 Create a variable for the ID of the Zope/0 container.

```
zope0=$(docker container ls --no-trunc --format "{{.ID}} {{.Command}}"
 \
  | awk '/runzope/ { if ($4 == "0" && $9 !~ "CONFIG") print $1 }') \
  && echo $zope0
```

6 Copy the script to the temporary directory of the Zope/0 container.

```
docker cp addGlobalConfToMariadb.py ${zope0}:/tmp
```

7 Start an interactive session in the Zope/0 container as user `zenoss`.

```
serviced service attach zope/0 su - zenoss
```

8 Change directory to the migration directory.

```
cd /opt/zenoss/Products/ZenModel/migrate
```

9 Copy the `addGlobalConfToMariadb.py` script from the temporary directory.

```
cp /tmp/addGlobalConfToMariadb.py .
```

10 Migrate the passwords.

- **To migrate 6.3.2 systems:**

  ```
  zenmigrate --step=AddGlobalConfToMariadb --dont-bump
  ```

- **To migrate 6.4.x systems:**

  ```
  zenmigrate --step=AddGlobalConfToMariadb
  ```

11 Exit the container.

```
exit
```

12 Upgrading Resource Manager

This chapter contains the procedures for upgrading a high-availability deployment of Resource Manager. Before starting the procedures in this chapter, complete the procedures in *Before upgrading a high-availability deployment* on page 39.

Note Before performing an upgrade or installing a ZenPack, Zenoss strongly recommends that you check the integrity of Resource Manager databases. For more information, refer to the *Zenoss Toolbox* documentation.

Stopping Resource Manager

Use this procedure to stop Resource Manager in a high-availability Control Center deployment.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as a user with `serviced` CLI privileges.
2. Check the status of Resource Manager.

   ```
   serviced service status --show-fields 'Name,ServiceID,Status'
   ```

   - If the status of all services is `stopped`, this procedure is complete. Continue to the next procedure.
   - If the status is `running`, perform the remaining steps.
3. Stop Resource Manager.

   ```
   serviced service stop Zenoss.resmgr
   ```
4. Check the status of Resource Manager.

   ```
   serviced service status --show-fields 'Name,ServiceID,Status'
   ```

   Repeat the command until the status of all services is `stopped`.

Upgrading Resource Manager

Use this procedure to upgrade a high-availability deployment of Resource Manager.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as a user with `serviced` CLI privileges.

2 Start the upgrade script.

```
/root/6.5.x/upgrade-resmgr.sh
```

The upgrade process begins. If you encounter errors, see *A snapshot with the given tag already exists* on page 43.

3 Restart Resource Manager.

Some Resource Manager services are started during the upgrade, and they must be restarted.

```
serviced service restart Zenoss.resmgr
```

A snapshot with the given tag already exists

When an upgrade attempt fails, the upgrade script does not remove the snapshot it creates at the beginning of the upgrade process. Use this procedure to remove the tag of the pre-upgrade snapshot and restart the upgrade. Untagged snapshots are removed when their time-to-live (TTL) expires. The TTL value is defined by the *SERVICED_SNAPSHOT_TTL* variable in the Control Center configuration file.

1 Log in to the Control Center master host as a user with `serviced` CLI privileges.

2 Create a variable for the identifier of the tenant application.

```
myTenant=$(serviced service list Zenoss.resmgr --format='{{.ID}}')
```

3 Display a list of all Control Center snapshots, with their tags.

```
serviced snapshot list -t
```

Example result, truncated to save space:

```
Snapshot                          Description  Tags
xm5mtezbyo2_20160211-220535.480                preupgrade-resmgr-6.5.0
```

The snapshot identifier is shown in the first column.

4 Remove the tag of the pre-upgrade snapshot.
Replace *Tag-Name* with the name of the pre-upgrade snapshot that was displayed in the previous step:

```
serviced snapshot untag ${myTenant} Tag-Name
```

5 Restart the upgrade script.

```
/root/6.5.x/upgrade-resmgr.sh
```

After upgrading a high-availability deployment 13

After Resource Manager is upgraded, perform the procedures in this chapter.

Removing the pre-upgrade snapshot

The Resource Manager upgrade script uses Control Center to create and tag a snapshot of the system before it begins the upgrade process. Tagged snapshots persist until they are explicitly removed, and grow over time. When you are satisfied the new release is working properly, remove the pre-upgrade snapshot.

1. Log in to the Control Center master host as a user with `serviced` CLI privileges.
2. Display a list of all Control Center snapshots, with their tags.

   ```
   serviced snapshot list -t
   ```

   Example result, truncated to save space:

   ```
   Snapshot                          Description  Tags
   xm5mtezbyo2_20160211-220535.480                preupgrade-resmgr-6.5.0
   ```

   The snapshot identifier is shown in the first column.
3. Remove the pre-upgrade snapshot.

   Replace *Snapshot-ID* with the identifier of the pre-upgrade snapshot returned in the previous step:

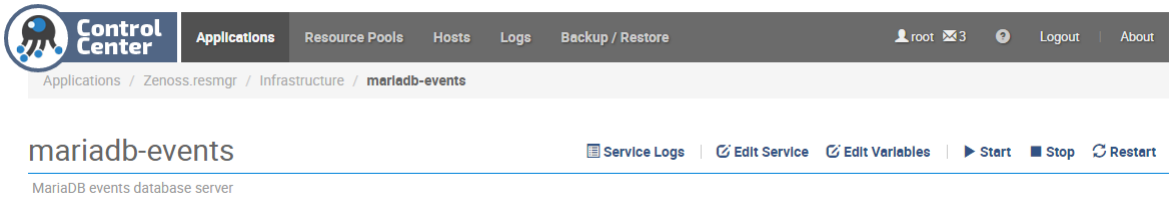
   ```
   serviced snapshot remove Snapshot-ID
   ```

Moving MariaDB services into the master resource pool

To perform this procedure, both Control Center master host nodes must have sufficient RAM and CPU resources for Control Center services and for the MariaDB services. For more information about the amount of RAM and CPU resources that are required, please contact your Zenoss representative.

Use this procedure to move the **mariadb-model** and **mariadb-events** services to the **master** resource pool. If the services are already in the **master** resource pool, skip this procedure.

1. Log in to the Control Center browser interface.
2. In the **Applications** table, click **Zenoss.resmgr**.
3. Scroll down to the **Services** table, and then expand the **Infrastructure** entry.
4. In the **Service** column of the **Services** table, click **mariadb-events**.



5 On the **mariadb-events** page, click **Edit Service**.
6 In the **Edit Service** dialog box, select **master** from the list in the **Resource Pool ID** field.
7 At the bottom of the **Edit Service** dialog box, click **Save Changes**.
8 On the **mariadb-events** page, click **Restart**.
9 Repeat steps 3-8 for the **mariadb-model** service.

Clearing heartbeat events

If you are using the **Daemon Process Down** portlet, `zencatalogservice` may be listed as down immediately after upgrading to this release. The status is incorrect and can be corrected by using this procedure to clear heartbeat events.

1 Log in to the Resource Manager browser interface as a user with ZenManager or Manager privileges.
2 Navigate to **ADVANCED** > **Settings**.
3 In the left panel, select **Events**.
4 At the bottom of the **Event Configuration** page, click the **Clear** button.

Starting and stopping Control Center deployments A

This appendix includes procedures for stopping and starting a high-availability Control Center deployment.

Note The procedures in this appendix assume that Control Center is the only source of Docker containers that are run on a host.

Stopping Control Center

To stop Control Center in a high-availability deployment, perform the procedures in this section, in order.

Stopping a master host node

Use this procedure to stop the Control Center service (`serviced`) on the master host in a high-availability deployment.

1. Use the virtual hostname or virtual IP address of the high-availability cluster to log in to the Control Center master node as `root`, or as a user with superuser privileges.
2. Display the public hostname of the current node.

   ```
   uname –n
   ```

   Make a note of which node (primary or secondary) is the current node, for use in a subsequent step.
3. Stop the top-level service `serviced` is managing, if necessary.
   a. Show the status of running services.

      ```
      serviced service status
      ```

      The top-level service is the service listed immediately below the headings line.

      - If the status of the top-level service and all child services is `stopped`, proceed to the next step.
      - If the status of the top-level service and all child services is **not** `stopped`, perform the remaining substeps.

   b. Stop the top-level service.
      Replace *Service* with the name or identifier of the top-level service:

      ```
      serviced service stop Service
      ```

c Monitor the stop.

```
serviced service status
```

When the status of the top-level service and all child services is `stopped`, proceed to the next step.

4 Stop Control Center with the cluster management tool.

```
pcs cluster standby --all
```

5 Monitor the status of cluster resources.

```
watch pcs status
```

Monitor the status until all resources report `Stopped`. Resolve any issues before continuing.

6 Ensure that no containers remain in the local repository.

a Start the Docker service.

```
systemctl start docker
```

b Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

If the command returns a result, enter the following command:

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

c Stop the Docker service.

```
systemctl stop docker
```

7 To ensure that no containers remain in both Docker repositories, log in to the other master node as `root`, or as a user with superuser privileges, and then perform the preceding step.

Stopping a delegate host

Use this procedure to stop the Control Center service (`serviced`) on a delegate host in a multi-host deployment. Repeat this procedure on each delegate host in your deployment.

1 Log in to the delegate host as `root`, or as a user with superuser privileges.

2 Stop the Control Center service.

```
systemctl stop serviced
```

3 Ensure that no containers remain in the local repository.

a Display the identifiers of all containers, running and exited.

```
docker ps -qa
```

- If the command returns no result, proceed to the next step.
- If the command returns a result, perform the following substeps.

b Remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, the most likely cause is an NFS conflict. Perform the following substeps.

c Stop the NFS and Docker services.

```
systemctl stop nfs && systemctl stop docker
```

d Start the NFS and Docker services.

```
systemctl start nfs && systemctl start docker
```

e Repeat the attempt to remove all remaining containers.

```
docker ps -qa | xargs --no-run-if-empty docker rm -fv
```

- If the remove command completes, proceed to the next step.
- If the remove command does not complete, perform the remaining substeps.

f Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

g Reboot the host.

```
reboot
```

h Log in to the delegate host as `root`, or as a user with superuser privileges.

i Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

4 Dismount all filesystems mounted from the Control Center master host.

This step ensures no stale mounts remain when the storage on the master host is replaced.

a Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the following substeps.

b Force the filesystems to dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f $FS
done
```

c Identify filesystems mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.

- If the preceding command returns a result, perform the following substeps.

d Perform a lazy dismount.

```
for FS in $(awk '/serviced/ { print $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs')
do
  umount -f -l $FS
done
```

e Restart the NFS service.

```
systemctl restart nfs
```

f Determine whether any filesystems remain mounted from the master host.

```
awk '/serviced/ { print $1, $2 }' < /proc/mounts \
  | grep -v '/opt/serviced/var/isvcs'
```

- If the preceding command returns no result, stop. This procedure is complete.
- If the preceding command returns a result, perform the remaining substeps.

g Disable the automatic startup of `serviced`.

```
systemctl disable serviced
```

h Reboot the host.

```
reboot
```

i Log in to the delegate host as `root`, or as a user with superuser privileges.

j Enable the automatic startup of `serviced`.

```
systemctl enable serviced
```

Starting Control Center

Use this procedure to start Control Center in a high-availability deployment. The default configuration of the Control Center service (`serviced`) is to start when the host starts. This procedure is only needed after stopping `serviced` to perform maintenance tasks.

1. Log in to the primary node as `root`, or as a user with superuser privileges.

   In this context, the primary node is the node that was the current node when you stopped Control Center.
2. Identify the hosts in the ZooKeeper ensemble.

   ```
   grep -E '^\b*SERVICED_ZK=' /etc/default/serviced
   ```

   The result is a list of 1, 3, or 5 hosts, separated by the comma character (`,`). The master host is always a node in the ZooKeeper ensemble.
3. In separate windows, log in to each of the delegate hosts that are nodes in the ZooKeeper ensemble as `root`, or as a user with superuser privileges.
4. Take the cluster out of standby mode.

   ```
   pcs cluster unstandby --all
   ```

5 On the other ZooKeeper ensemble hosts, start `serviced`.

The window of time for starting a ZooKeeper ensemble is relatively short. The goal of this step is to start Control Center on each ensemble node at about the same time, so that each node can participate in electing the leader.

```
systemctl start serviced
```

6 Monitor the status of cluster resources.

```
watch pcs status
```

Monitor the status until all resources report `Started`. Resolve any issues before continuing.

7 On the master host, check the status of the ZooKeeper ensemble.

a Attach to the container of the ZooKeeper service.

```
docker exec -it serviced-isvcs_zookeeper bash
```

b Query the master host and identify its role in the ensemble.
Replace *Master* with the hostname or IP address of the master host:

```
{ echo stats; sleep 1; } | nc Master 2181 | grep Mode
```

The result includes `leader` or `follower`. When multiple hosts rely on the ZooKeeper instance on the master host, the result includes `standalone`.

c Query the other delegate hosts to identify their role in the ensemble.
Replace *Delegate* with the hostname or IP address of a delegate host:

```
{ echo stats; sleep 1; } | nc Delegate 2181 | grep Mode
```

d Detach from the container of the ZooKeeper service.

```
exit
```

If none of the nodes reports that it is the ensemble leader within a few minutes of starting `serviced`, reboot the ensemble hosts.

8 Log in to each of the delegate hosts that are not nodes in the ZooKeeper ensemble as `root`, or as a user with superuser privileges, and then start `serviced`.

```
systemctl start serviced
```

9 Optional: Monitor the startup, if desired.

```
journalctl -u serviced -f -o cat
```

Once Control Center is started, it is ready to start managing applications. For more information, refer to the documentation of your application.

Resolving package dependency conflicts B

This appendix includes procedures for resolving common Docker CE and Control Center dependency conflicts.

Resolving device mapper dependency conflicts

To perform this procedure, you need:

- An RHEL/CentOS system with internet access and the same operating system release and kernel as the Control Center hosts in your deployment.
- A secure network copy program.

Use this procedure to resolve dependency issues in which the installed versions of device mapper libraries are newer than the versions included in the Zenoss mirror. The following example shows a typical `yum` error of this type:

```
Error: Package: 7:device-mapper-event-1.02.107-5.el7.x86_64 (zenoss-
mirror)
Requires: device-mapper = 7:1.02.107-5.el7
Installed: 7:device-mapper-1.02.107-5.el7_2.5.x86_64 (@updates)
device-mapper = 7:1.02.107-5.el7_2.5
```

Follow these steps:

1. Display the version number of the installed device mapper package.

   ```
   rpm -q device-mapper | cut -d - -f 3-
   ```

   Example result:

   ```
   1.02.135-1.el7_3.1.x86_64
   ```

   Record the version number for subsequent use.
2. Log in to a compatible host that is connected to the internet as `root`, or as a user with superuser privileges.

   The host must have the same operating system (RHEL or CentOS) and release installed as the Control Center hosts in your deployment.
3. Install `yum` utilities, if necessary.

a. Determine whether the `yum` utilities package is installed.

```
rpm -qa | grep yum-utils
```

- If the command returns a result, the package is installed. Proceed to the next step.
- If the command does not return a result, the package is not installed. Perform the following substep.

b. Install the `yum-utils` package.

```
yum install yum-utils
```

4. Download the required dependencies, and then create a `tar` archive of the files.
   a. Create a variable for the dependency version to download.
   Replace *Device-Mapper-Version* with the version number displayed in a previous step:

```
myVersion=Device-Mapper-Version
```

   b. Create a temporary directory for the dependencies.

```
mkdir /tmp/downloads
```

   c. Download the dependencies to the temporary directory.

```
yum install --downloadonly --downloaddir=/tmp/downloads \
  device-mapper-event-$myVersion
```

   The `yum` command downloads two package files.
   d. Create a `tar` archive of the temporary directory.

```
cd /tmp && tar czf ./downloads.tgz ./downloads
```

5. Use a secure copy program to copy the archive file to the `/tmp` directory of the Control Center host or hosts that need the dependencies.
6. Log in to the host as `root`, or as a user with superuser privileges.
7. Install the device mapper dependencies.
   a. Extract the packages from the `tar` archive.

```
cd /tmp && tar xzf ./downloads.tgz
```

   b. Install the dependencies.

```
yum install $(ls /tmp/downloads/*.rpm)
```

Return to the procedure you were performing before turning to this appendix and retry the `yum` install command that failed previously.

Resolving other dependency conflicts

Use this procedure to resolve dependency issues in which the installed versions of one or more dependencies are newer than the versions included in the Zenoss mirror. The following example shows a typical `yum` error of this type:

```
Error: Package: policycoreutils-python-2.5-9.el7.x86_64 (zenoss-mirror)
Requires: policycoreutils = 2.5-9.el7
```

```
Installed: policycoreutils-2.5-11.el7_3.x86_64 (@updates)
```

Follow these steps:

1. Install the older package.
   Replace *Package-Name* with the name of the package displayed in the error message:

   ```
   rpm -Uvh --oldpackage Package-Name
   ```

2. Clean all `yum` caches.

   ```
   yum clean all
   ```

Return to the procedure you were performing before turning to this appendix and retry the `yum` install command that failed previously.

Updating the cluster management software C

The procedures in this appendix describe how to update the Distributed Replicated Block Device (DRBD) and Pacemaker/Corosync software, if desired. Typically, the software is updated only to resolve issues caused by the currently-installed versions of the packages.

Note Zenoss supports DRBD 8.4, not DRBD 9.0. The procedures in this section download the latest version of release 8.4.

Downloading and staging cluster software

To perform this procedure, you need:

- An RHEL/CentOS system with internet access and the same operating system release and kernel as the master host nodes.
- A secure network copy program.

Use this procedure to download packages for Distributed Replicated Block Device (DRBD) and Pacemaker/Corosync, and to bundle them for installation on master host nodes.

1. Log in to a compatible host that is connected to the internet as `root`, or as a user with superuser privileges.

   The host must have the same operating system (RHEL or CentOS) and release installed, and the same version of the Linux kernel, as the master host nodes.
2. Install `yum` utilities, if necessary.
   a. Determine whether the `yum` utilities package is installed.

   ```
   rpm -qa | grep yum-utils
   ```

   - If the command returns a result, the package is installed. Proceed to the next step.
   - If the command does not return a result, the package is not installed. Perform the following substep.

   b. Install the `yum-utils` package.

   ```
   yum install yum-utils
   ```
3. Add the Enterprise Linux packages repository (ELRepo), if necessary.
   a. Determine whether the ELRepo repository is available.

   ```
   yum repolist | grep elrepo
   ```

- If the command returns a result, the repository is available. Proceed to the next step.
- If the command does not return a result, the repository is not available. Perform the following substeps.

b Import the public key for the repository.

```
rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
```

c Add the repository to the download host.

```
rpm -Uvh \
  https://www.elrepo.org/elrepo-release-7.0-4.el7.elrepo.noarch.rpm
```

d Clean and update the `yum` caches.

```
yum clean all && yum makecache fast
```

4 Download the required packages and their dependencies, and then create a `tar` archive of the package files.

a Create a temporary directory for the packages.

```
mkdir /tmp/downloads
```

b Download the DRBD packages to the temporary directory.

```
repotrack -a x86_64 -r elrepo -p /tmp/downloads kmod-drbd84
```

c Download the Corosync/Pacemaker packages to the temporary directory.

```
repotrack -a x86_64 -p /tmp/downloads pcs
```

d Create a `tar` archive of the temporary directory.

```
cd /tmp && tar czf ./downloads.tgz ./downloads
```

5 Remove the Enterprise Linux packages repository.

```
yum remove elrepo-release && yum clean all
```

6 Use a secure copy program to copy the packages archive to the `/tmp` directory of each master host node.

Updating cluster software

Use this procedure to update Distributed Replicated Block Device (DRBD) and Pacemaker/Corosync, if desired.

1 Stop all applications, and then stop the high-availability cluster.
For more information, see *Stopping a Control Center deployment* on page 15.

2 Log in to the primary node as `root`, or as a user with superuser privileges.

3 In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.

4 On both nodes, stop the cluster software.

a Stop the PCS daemon.

```
systemctl stop pcsd.service
```

b Stop DRBD.

```
drbdadm down all
```

5 On both nodes, extract and install the cluster management packages.

a Extract the packages.

```
cd /tmp && tar xzf ./downloads.tgz
```

b Install the packages.

```
yum install ./downloads/*.rpm
```

6 On both nodes, install the Pacemaker resource agents for Control Center.

```
yum install \
  /opt/zenoss-repo-mirror/serviced-resource-agents-1.1.0-1.x86_64.rpm
```

7 Determine which node is the primary node.

```
pcs status
```

8 On the primary node, start the cluster software.

a Start DRBD.

```
drbdadm up all
```

b Start DRBD and assign the primary role.

```
drbdadm up all && drbdadm primary --force serviced-dfs
```

c Start the PCS daemon.

```
systemctl start pcsd.service
```

9 On the secondary node, start the cluster software.

a Start DRBD.

```
drbdadm up all
```

b Start the PCS daemon.

```
systemctl start pcsd.service
```

10 Start the high-availability cluster, and then start applications.

For more information, see *Starting Control Center* on page 49.

Installing the Resource Manager application template D

This appendix includes procedures for downloading and installing the most recent Resource Manager application template. The latest template is not needed to perform an upgrade, and is included in appliance upgrades. Use the procedures in this appendix if you plan to delete a deployed application template and then redeploy with a newer version of the template. (For example, in a development or staging environment.)

Downloading and installing the template package

To perform this procedure, you need:

- A workstation with internet access.
- Permission to download files from the *delivery.zenoss.com* site. Zenoss customers may request permission by filing a ticket at the *Zenoss Support* site.
- A secure network copy program.

Use this procedure to download the required files to a workstation and then install the files on both Control Center master hosts.

Perform these steps:

1. In a web browser, navigate to the *delivery.zenoss.com* site.
2. Log in with the account provided by Zenoss Support.
3. Download the template package file.
   `zenoss-resmgr-service-6.5.0-1.noarch.rpm`
4. Use a secure copy program to copy the package file to both Control Center master hosts.
5. Log in to the target host as `root`, or as a user with superuser privileges.
6. Change directory to the directory in which the template package file is located.
7. Install the template.

   ```
   yum install ./zenoss-resmgr-service-6.5.0-1.noarch.rpm
   ```

   The template file is stored in `/opt/serviced/templates`.