Zenoss Resource Manager Installation Guide

Release 5.1.5

Zenoss, Inc.

www.zenoss.com

Zenoss Resource Manager Installation Guide

Part Number: 1052.16.207

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

Contents

About this guide

Zenoss Resource Manager Installation Guide provides detailed procedures for installing Zenoss Resource Manager (Resource Manager).

Note Zenoss strongly recommends reviewing the *Zenoss Resource Manager Planning Guide* carefully before using this guide.

Related publications

Title	Description
Zenoss Resource Manager Administration Guide	Provides an overview of Resource Manager architecture and features, as well as procedures and examples to help use the system.
Zenoss Resource Manager Configuration Guide	Provides required and optional configuration procedures for Resource Manager, to prepare your deployment for monitoring in your environment.
Zenoss Resource Manager Installation Guide	Provides detailed information and procedures for creating deployments of Control Center and Resource Manager.
Zenoss Resource Manager Planning Guide	Provides both general and specific information for preparing to deploy Resource Manager.
Zenoss Resource Manager Release Notes	Describes known issues, fixed issues, and late-breaking information not already provided in the published documentation set.
Zenoss Resource Manager Upgrade Guide	Provides detailed information and procedures for upgrading deployments of Resource Manager.

Additional information and comments

If you have technical questions about this product that are not answered in this guide, please visit the *Zenoss Support* site or contact Zenoss Support.

Zenoss welcomes your comments and suggestions regarding our documentation. To share your comments, please send an email to `docs@zenoss.com`. In the email, include the document title and part number. The part number appears at the end of the list of trademarks, at the front of this guide.

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. For information about new or changed features, refer to the *Zenoss Resource Manager Release Notes*.

1052.16.207

Update release numbers.

1052.16.176

Update release numbers.

1052.16.153

Update release numbers.

1052.16.146

Update release numbers.

1052.16.125

Refine the procedure for creating the application data thin pool.

1052.16.118

Add support for Resource Manager 5.1.2.

Add a substep to create the docker override directory.

1052.16.111

Add this document change history.

Add chapters describing how to install the Resource Manager appliance.

Chapters are organized into parts.

Docker configuration steps now add the storage driver flag (`-s devicemapper`) to the `/etc/sysconfig/docker` file.

Docker needs a longer startup timeout value, to work around a known Docker issue with the `devicemapper` driver. Docker configuration steps now include adding `TimeoutSec=300`.

Rather than editing `/lib/systemd/system/docker.service`, Docker configuration steps now include adding a `systemd` override file.

Add a symlink to `/tmp` in `/var/lib/docker`.

Update the commands for starting and testing a ZooKeeper ensemble.

Add a procedure for updating the *SERVICED_ZK* value on resource pool hosts that are not members of a ZooKeeper ensemble.

Add a reference topic for the ZooKeeper variables required on hosts in a Control Center cluster.

Add procedures for configuring an NTP server and clients for offline deployments.

Add step to install the Nmap Ncat package, which is used to check ZooKeeper ensemble status.

1052.16.060

Planning information is now in the *Zenoss Resource Manager Planning Guide*.

Information about how to start and configure Resource Manager is now in the *Zenoss Resource Manager Configuration Guide*.

New procedures are included, for installing without internet access, and for installing high-availability deployments.

Part I: Customized deployments

The chapters in this part describe how to install Control Center and Resource Manager on real or virtual hosts, with or without internet access. The instructions include the full range of options for customizing your deployment for your environment.

1 Installing on hosts with internet access

The procedures in this chapter install Control Center and Resource Manager on one or more Red Hat Enterprise Linux (RHEL) 7.1 or 7.2 hosts, or one or more CentOS 7.1 or 7.2 hosts. To use the procedures in this chapter, all Control Center cluster hosts must have internet access.

You may create a single-host or a multi-host deployment. For production use, Zenoss strongly recommends creating a multi-host deployment that includes a minimum of three real or virtual machines. For more information about deploying Control Center and Resource Manager, refer to the *Zenoss Resource Manager Planning Guide*.

Note For optimal results, review this chapter thoroughly before starting the installation process.

Installing a master host

Perform the procedures in this section to install Control Center and Resource Manager on a master host.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center master host.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

   ```
   uname -m
   ```

   - If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
   - If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.
3. Verify that name resolution works on this host.

   ```
   hostname -i
   ```

   If the result is not a valid IPv4 adddress, add an entry for the host to the network nameserver, or to `/etc/hosts`.
4. Verify that the host's numeric identifier is unique.

   Each host in a Control Center cluster must have a unique host identifier.

   ```
   hostid
   ```

5 Determine whether the available, unused storage is sufficient.
   a Display the available storage devices.

```
lsblk --output=NAME,SIZE
```

   b Compare the available storage with the amount required for a Control Center master host.
   For more information, refer to the *Zenoss Resource Manager Planning Guide*.
6 Determine whether the available memory and swap is sufficient.
   a Display the available memory.

```
free -h
```

   b Compare the available memory with the amount required for a master host in your deployment.
   For more information, refer to the *Zenoss Resource Manager Planning Guide*.
7 Update the operating system, if necessary.
   a Determine which release is installed.

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

   If the result includes `7.0`, perform the following substeps.
   b Update the operating system.

```
yum update -y
```

   c Restart the system.

```
reboot
```

Preparing storage for the master host

In addition to the storage required for its operating system, a Control Center master host requires the following storage areas:

- A local partition for Docker data, configured as a device mapper thin pool.
- A local partition for Control Center internal services data, formatted with the XFS file system.

> **Note** Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

- A local or remote primary partition for Resource Manager data, configured as a device mapper thin pool.
- A local primary partition, a remote primary partition, or a remote file server, for backups of Resource Manager data. The local or remote primary partition is formatted with the XFS file system. A remote file server must provide a file system that is compatible with XFS.

> **Note** If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

For storage sizing information, refer to the *Zenoss Resource Manager Planning Guide*.

For device mapper thin pools, no formatting is required—simply create primary partitons, which are configured in subsequent procedures. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

To create the required storage, perform the following procedures.

Note Data present on the primary partitions you select are destroyed in these procedure. Please ensure that data is backed up elsewhere, or no longer needed, before proceeding.

Creating a file system for internal services

This procedure creates an XFS file system on a primary partition. For more information about primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

Note Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Identify the target primary partition for the file system to create.

   ```
   lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
   ```

   For more information about the output of the `lsblk` command, and about creating primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.
3. Create an XFS file system.
   Replace *Partition* with the path of the target primary partition:

   ```
   mkfs -t xfs Partition
   ```
4. Add an entry to the `/etc/fstab` file.
   Replace *Partition* with the path of the primary partition used in the previous step:

   ```
   echo "Partition \
     /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
   ```
5. Create the mount point for internal services data.

   ```
   mkdir -p /opt/serviced/var/isvcs
   ```
6. Mount the file system, and then verify it mounted correctly.

   ```
   mount -a && mount | grep isvcs
   ```

   Example result:

   ```
   /dev/xvdb1 on /opt/serviced/var/isvcs type xfs
    (rw,relatime,seclabel,attr2,inode64,noquota)
   ```

Creating a file system for backups

To perform this procedure, you need a host with at least one unused primary partition, or a remote file server.

The Control Center master host requires local or remote storage space for backups of Control Center data. This procedure includes steps to create an XFS file system on a primary partition, if necessary, and steps to mount a file system for backups. For more information about primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Optional: Identify the target primary partition for the file system to create, if necessary.

   Skip this step if you are using a remote file server.

   ```
   lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
   ```

   For more information about the output of the `lsblk` command, and about creating primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.
3. Optional: Create an XFS file system, if necessary.

   Skip this step if you are using a remote file server.

   Replace *Partition* with the path of the target primary partition:

   ```
   mkfs -t xfs Partition
   ```
4. Create an entry in the `/etc/fstab` file.

   Replace *File-System-Specification* with one of the following values:

   - the path of the primary partition used in the previous step
   - the remote server specification

   ```
   echo "File-System-Specification \
     /opt/serviced/var/backups xfs defaults 0 0" >> /etc/fstab
   ```
5. Create the mount point for backup data.

   ```
   mkdir -p /opt/serviced/var/backups
   ```
6. Mount the file system, and then verify it mounted correctly.

   ```
   mount -a && mount | grep backups
   ```

   Example result:

   ```
   /dev/sdb3 on /opt/serviced/var/backups type xfs
    (rw,relatime,seclabel,attr2,inode64,noquota)
   ```

Preparing the master host operating system

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center master host.

1. Log in to the candidate master host as `root`, or as a user with superuser privileges.
2. Add an entry to `/etc/hosts` for localhost, if necessary.
   a. Determine whether `127.0.0.1` is mapped to `localhost`.

      ```
      grep 127.0.0.1 /etc/hosts | grep localhost
      ```

      If the preceding commands return no result, perform the following substep.
   b. Add an entry to `/etc/hosts` for `localhost`.

      ```
      echo "127.0.0.1   localhost" >> /etc/hosts
      ```

3 Disable the firewall, if necessary.

This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

- If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
- If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

4 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

5 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations. **If you are installing a single-host deployment, skip this step**.

```
yum install -y nmap-ncat
```

8 Install and configure the NTP package.

a Install the package.

```
yum install -y ntp
```

b Set the system time.

```
ntpd -gq
```

c Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

d Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot. The following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

9 Install the Zenoss repository package.

a Install the package.

```
rpm -ivh http://get.zenoss.io/yum/zenoss-repo-1-1.x86_64.rpm
```

b Clean out the `yum` cache directory.

```
yum clean all
```

10 Reboot the host.

```
reboot
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

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

2 Install Docker 1.9.0, and then disable accidental upgrades.

a Add the Docker repository to the host's repository list.

```
cat > /etc/yum.repos.d/docker.repo <<-EOF
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

b Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install -y docker-engine-1.9.0
```

c Open `/etc/yum.repos.d/docker.repo` with a text editor.

d Change the value of the `enabled` key from `1` to `0`.

e Save the file and close the text editor.

3 Create a symbolic link for the Docker temporary directory.

Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

a Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

b Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4 Create a `systemd` override file for the Docker service definition.

a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-stable install -y serviced-1.1.7
```

6 Create a device mapper thin pool for Docker data.

a Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

b Create the thin pool.

Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

7 Configure and start the Docker service.

a Create variables for adding arguments to the Docker configuration file.

The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

8 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

d Restart the Docker service.

```
systemctl restart docker
```

Installing Resource Manager

This procedure installs Resource Manager and configures the NFS server.

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

2 Install Resource Manager.

```
yum --enablerepo=zenoss-stable install -y zenoss-resmgr-service
```

3 Authenticate to the Docker Hub repository.

Replace *USER* and *EMAIL* with the values associated with your Docker Hub account.

```
docker login -u USER -e EMAIL
```

The `docker` command prompts you for your Docker Hub password, and saves a hash of your credentials in the `$HOME/.dockercfg` file (root user account).

4 Configure and restart the NFS server.

Currently, *an unresolved issue* prevents the NFS server from starting correctly. The following commands provide a workaround to ensure that it does.

a Open `/lib/systemd/system/nfs-server.service` with a text editor.

b Change `rpcbind.target` to `rpcbind.service` on the following line:

```
Requires= network.target proc-fs-nfsd.mount rpcbind.target
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

Configuring Control Center

This procedure creates a thin pool for application data and customizes key configuration variables of Control Center.

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

2 Configure Control Center to serve as the master and as an agent.

The following variables configure `serviced` to serve as both master and agent:

`SERVICED_AGENT`

Default: `0` (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: `0` (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_AGENT=0
```

c Remove the number sign character (#) from the beginning of the line.

d Find the *SERVICED_MASTER* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_MASTER=0
```

e Remove the number sign character (#) from the beginning of the line.

f Save the file, and then close the editor.

3 Create a thin pool for Resource Manager data.

   a Identify the primary partition for the thin pool to create, and the amount of space available on the primary partition.

   ```
   lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
   ```

   For more information about the output of the `lsblk` command and primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

   b Create a variable for 50% of the space available on the primary partition for the thin pool to create.

   The thin pool stores application data and snapshots of the data. You can add storage to the pool at any time.

   Replace *Half-Of-Available-Space* with 50% of the space available in the primary partition, in gigabytes. Include the symbol for gigabytes (`G`) after the numeric value.

   ```
   myFifty=Half-Of-Available-SpaceG
   ```

   c Create the thin pool.

   Replace *Path-To-Device* with the path of the target primary partition:

   ```
   serviced-storage create-thin-pool -o dm.basesize=$myFifty \
     serviced Path-To-Device
   ```

   On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

4 Configure Control Center with the name of the thin pool for Resource Manager data.

The Control Center configuration file is `/etc/default/serviced`. (For more information about `serviced` configuration options, refer to the Control Center online help.)

   a Open `/etc/default/serviced` in a text editor.

   b Locate the *SERVICED_FS_TYPE* declaration.

   c Remove the number sign character (#) from the beginning of the line.

   d Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

   Replace *Thin-Pool-Name* with the name of the thin pool created previously:

   ```
   SERVICED_DM_THINPOOLDEV=Thin-Pool-Name
   ```

   e Save the file, and then close the editor.

5 Optional: Specify an alternate private subnet for Control Center, if necessary.

The default private subnet may already be in use in your environment. The following variable configures `serviced` to use an alternate subnet:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

   a Open `/etc/default/serviced` in a text editor.

   b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

   ```
   # SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
   ```

   c Remove the number sign character (#) from the beginning of the line.

   d Save the file, and then close the editor.

User access control

Control Center provides a browser interface and a command-line interface.

To gain access to the Control Center browser interface, users must have login accounts on the Control Center master host. (Pluggable Authentication Modules (PAM) is supported.) In addition, users must be members of the Control Center administrative group, which by default is the system group, `wheel`. To enhance security, you may change the administrative group from `wheel` to any non-system group.

To use the Control Center command-line interface, users must have login accounts on the Control Center master host, and be members of the `docker` user group. Members of the `wheel` group, including `root`, are members of the `docker` group.

Adding users to the default administrative group

This procedure adds users to the default administrative group of Control Center, `wheel`. Performing this procedure enables users with superuser privileges to gain access to the Control Center browser interface.

Note Perform this procedure or the next procedure, but not both.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Add users to the system group, `wheel`.

   Replace *User* with the name of a login account on the master host.

   ```
   usermod -aG wheel User
   ```

   Repeat the preceding command for each user to add.

   Note For information about using Pluggable Authentication Modules (PAM), refer to your operating system documentation.

Configuring a regular group as the Control Center administrative group

This procedure changes the default administrative group of Control Center from `wheel` to a non-system group.

Note Perform this procedure or the previous procedure, but not both.

1. Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2. Create a variable for the group to designate as the administrative group.

   In this example, the name of group to create is `serviced`. You may choose any name or use an existing group.

   ```
   GROUP=serviced
   ```

3. Create a new group, if necessary.

   ```
   groupadd $GROUP
   ```

4. Add one or more existing users to the new administrative group.

   Replace *User* with the name of a login account on the host:

   ```
   usermod -aG $GROUP User
   ```

   Repeat the preceding command for each user to add.
5. Specify the new administrative group in the `serviced` configuration file.

The following variable specifies the administrative group:

SERVICED_ADMIN_GROUP

Default: `wheel`

The name of the Linux group on the Control Center master host whose members are authorized to use the Control Center browser interface. You may replace the default group with a group that does not have superuser privileges.

a. Open `/etc/default/serviced` in a text editor.
b. Find the *SERVICED_ADMIN_GROUP* declaration, and then change the value from `wheel` to the name of the group you chose earlier.
The following example shows the line to change:

```
# SERVICED_ADMIN_GROUP=wheel
```

c. Remove the number sign character (#) from the beginning of the line.
d. Save the file, and then close the editor.

6. Optional: Prevent `root` users and members of the `wheel` group from gaining access to the Control Center browser interface, if desired.

The following variable controls privileged logins:

SERVICED_ALLOW_ROOT_LOGIN

Default: `1` (true)

Determines whether `root`, or members of the `wheel` group, may gain access to the Control Center browser interface.

a. Open `/etc/default/serviced` in a text editor.
b. Find the *SERVICED_ALLOW_ROOT_LOGIN* declaration, and then change the value from `1` to `0`.
The following example shows the line to change:

```
# SERVICED_ALLOW_ROOT_LOGIN=1
```

c. Remove the number sign character (#) from the beginning of the line.
d. Save the file, and then close the editor.

Enabling use of the command-line interface

This procedure enables users to perform administrative tasks with the Control Center command-line interface by adding individual users to the `docker` group.

1. Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2. Add users to the Docker group, `docker`.

Replace *User* with the name of a login account on the host.

```
usermod -aG docker User
```

Repeat the preceding command for each user to add.

Starting Control Center

This procedure starts the Control Center service, `serviced`.

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

```
systemctl start serviced
```

To monitor progress, enter the following command:

```
journalctl -flu serviced -o cat
```

The `serviced` daemon invokes `docker` to pull its internal services images from Docker Hub. The Control Center browser and command-line interfaces are unavailable until the images are installed and the services are started. The process takes approximately 5-10 minutes. When the message `Trying to discover my pool` repeats, Control Center is ready for the next steps.

3 **Note** Perform this step **only if you are installing a single-host deployment**.

Optional: Add the master host to the `default` resource pool.

Replace *Hostname-Or-IP* with the hostname or IP address of the Control Center master host:

```
serviced host add Hostname-Or-IP:4979 default
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Isolating the master host in a separate resource pool

Note If you are configuring a single-host deployment, skip this procedure.

Control Center enables or just performs rapid recovery from application service failures. When Control Center internal services and application services share a host, application failures can limit recovery options. Zenoss strongly recommends isolating the Control Center master host in a separate resource pool.

This procedure creates a new resource pool for the Control Center master host, and then adds the master host to the pool.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Create a new resource pool named `master`.

```
serviced pool add master
```

3 Add the master host to the `master` resource pool.

Replace *Hostname-Or-IP* with the hostname or IP address of the Control Center master host:

```
serviced host add Hostname-Or-IP:4979 master
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Installing resource pool hosts

Note If you are installing a single-host deployment, skip this section.

Control Center resource pool hosts run the application services scheduled for the resource pool to which they belong, and for which they have sufficent RAM and CPU resources.

Resource Manager has two broad categories of application services: Infrastructure and collection. The services associated with each category can run in the same resource pool, or can run in separate resource pools.

For improved reliability, two resource pool hosts are configured as nodes in an *Apache ZooKeeper* ensemble. The storage required for ensemble hosts is slightly different than the storage required for all other resource pool hosts: Each ensemble host requires a separate primary partition for Control Center internal services data, in addition to the primary partition for Docker data. Unless the ZooKeeper service on the Control Center master host fails, their roles in the ZooKeeper ensemble do not affect their roles as Control Center resource pool hosts.

Note The hosts for the ZooKeeper ensemble require static IP addresses, because ZooKeeper does not support hostnames in its configurations.

Repeat the procedures in the following sections for each host you wish to add to your Control Center deployment.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

   ```
   uname -m
   ```

   - If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
   - If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.
3. Verify that name resolution works on this host.

   ```
   hostname -i
   ```

   If the result is not a valid IPv4 adddress, add an entry for the host to the network nameserver, or to `/etc/hosts`.
4. Verify that the host's numeric identifier is unique.

   Each host in a Control Center cluster must have a unique host identifier.

   ```
   hostid
   ```
5. Determine whether the available, unused storage is sufficient.
   a. Display the available storage devices.

      ```
      lsblk --output=NAME,SIZE
      ```

   b. Compare the available storage with the amount required for a resource pool host in your deployment.

      In particular, resource pool hosts that are configured as nodes in a ZooKeeper ensemble require an additional primary partition for Control Center internal services data.

      For more information, refer to the *Zenoss Resource Manager Planning Guide*.
6. Determine whether the available memory and swap is sufficient.
   a. Display the available memory.

      ```
      free -h
      ```

   b. Compare the available memory with the amount required for a resource pool host in your deployment.

      For more information, refer to the *Zenoss Resource Manager Planning Guide*.
7. Update the operating system, if necessary.

a Determine which release is installed.

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

If the result includes `7.0`, perform the following substeps.

b Update the operating system.

```
yum update -y
```

c Restart the system.

```
reboot
```

Preparing a resource pool host

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center resource pool host.

1 Log in to the candidate resource pool host as `root`, or as a user with superuser privileges.
2 Add an entry to `/etc/hosts` for localhost, if necessary.
   a Determine whether `127.0.0.1` is mapped to `localhost`.

```
grep 127.0.0.1 /etc/hosts | grep localhost
```

   If the preceding commands return no result, perform the following substep.
   b Add an entry to `/etc/hosts` for `localhost`.

```
echo "127.0.0.1   localhost" >> /etc/hosts
```

3 Disable the firewall, if necessary.

   This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

   a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

   - If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
   - If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

   b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

   On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

4 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

5 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install and configure the NTP package.

a Install the package.

```
yum install -y ntp
```

b Set the system time.

```
ntpd -gq
```

c Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

d Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot. The following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

8 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations. **Perform this step only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**.

```
yum install -y nmap-ncat
```

9 Install the Zenoss repository package.

a Install the package.

```
rpm -ivh http://get.zenoss.io/yum/zenoss-repo-1-1.x86_64.rpm
```

b Clean out the `yum` cache directory.

```
yum clean all
```

10 Reboot the host.

```
reboot
```

Creating a file system for Control Center internal services

This procedure creates an XFS file system on a primary partition.

Note **Perform this procedure only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**. No other resource pool hosts run Control Center internal services, so no other pool hosts need a partition for internal services data.

---

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

2 Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

3 Create an XFS file system.
Replace *Isvcs-Partition* with the path of the target primary partition:

```
mkfs -t xfs Isvcs-Partition
```

4 Create the mount point for Control Center internal services data.

```
mkdir -p /opt/serviced/var/isvcs
```

5 Add an entry to the `/etc/fstab` file.
Replace *Isvcs-Partition* with the path of the primary partition used in the previous step:

```
echo "Isvcs-Partition \
  /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep isvcs
```

Example result:

```
/dev/xvdb1 on /opt/serviced/var/isvcs type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

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

2 Install Docker 1.9.0, and then disable accidental upgrades.

a Add the Docker repository to the host's repository list.

```
cat > /etc/yum.repos.d/docker.repo <<-EOF
[dockerrepo]
```

```
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

b Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install -y docker-engine-1.9.0
```

c Open `/etc/yum.repos.d/docker.repo` with a text editor.

d Change the value of the `enabled` key from `1` to `0`.

e Save the file and close the text editor.

3 Create a symbolic link for the Docker temporary directory.

Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

a Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

b Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4 Create a `systemd` override file for the Docker service definition.

a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-stable install -y serviced-1.1.7
```

6 Create a device mapper thin pool for Docker data.

a Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

b Create the thin pool.

Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

7 Configure and start the Docker service.

a Create variables for adding arguments to the Docker configuration file.

The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

8 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

---

d Restart the Docker service.

```
systemctl restart docker
```

Configuring and starting Control Center

This procedure customizes key configuration variables of Control Center.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Configure Control Center as an agent of the master host.

The following variable configures `serviced` to serve as agent:

`SERVICED_AGENT`

Default: `0` (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: `0` (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

In addition, the following lines need to be edited, to replace `{{SERVICED_MASTER_IP}}` with the IP address of the master host:

```
# SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
# SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
# SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
# SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
# SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
# SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
```

a Open `/etc/default/serviced` in a text editor.
b Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_AGENT=0
```

c Remove the number sign character (#) from the beginning of the line.
d Find the *SERVICED_MASTER* declaration, and then remove the number sign character (#) from the beginning of the line.
e Globally replace `{{SERVICED_MASTER_IP}}` with the IP address of the master host.

Note Remove the number sign character (#) from the beginning of each variable declaration that includes the master IP address.

---

f Save the file, and then close the editor.
3 Optional: Specify an alternate private subnet for Control Center, if necessary.

The default private subnet may already be in use in your environment. The following variable configures `serviced` to use an alternate subnet:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

4 Start the Control Center service (`serviced`).

```
systemctl start serviced
```

To monitor progress, enter the following command:

```
journalctl -flu serviced -o cat
```

To install additional resource pool hosts, return to *Verifying candidate host resources* on page 21.

ZooKeeper ensemble configuration

Note If you are installing a single-host deployment, or if your deployment includes fewer than two resource pool hosts, skip this section.

Control Center relies on *Apache ZooKeeper* to coordinate its services. The procedures in this section create a ZooKeeper ensemble of 3 nodes. To perform these procedures, you need a Control Center master host and a minimum of two resource pool hosts. Each resource pool host requires a separate primary partition for Control Center internal services, and each should have a static IP address. For more information about storage requirements, refer to the *Zenoss Resource Manager Planning Guide*.

Note Zenoss strongly recommends configuring a ZooKeeper ensemble for all production deployments.

A ZooKeeper ensemble requires a minimum of 3 nodes, and 3 nodes is sufficient for most deployments. A 5-node configuration improves failover protection during maintenance windows. Ensembles larger than 5 nodes are not necessary. An odd number of nodes is recommended, and an even number of nodes is strongly discouraged.

Note The Control Center ZooKeeper service requires consistently fast storage. Ideally, the primary partition for Control Center internal services is on a separate, high-performance device that has only one primary partition.

Control Center variables for ZooKeeper

This tables in this section associates the ZooKeeper-related Control Center variables to set in `/etc/default/serviced` with the roles that hosts play in a Control Center cluster.

Table 1: Control Center master host

SERVICED_ISVCS_ZOOKEEPER_ID The unique identifier of a ZooKeeper ensemble node. Value: 1
SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 2: Control Center resource pool host and ZooKeeper ensemble node

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 2 or 3

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ISVCS_START

The list of Control Center internal services to start and run on hosts other than the master host.
Value: `zookeeper`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 3: Control Center resource pool host only

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Configuring the master host as a ZooKeeper node

This procedure configures the Control Center master host as a member of the ZooKeeper ensemble.

Note For accuracy, this procedure constructs Control Center configuration variables in the shell and appends them to `/etc/default/serviced`. The last step is to move the variables from the end of the file to more appropriate locations.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

Note Define the variables identically on the master host and on each resource pool host.

Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=Master-Host-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

Note ZooKeeper requires IP addresses for ensemble configuration.

3 Set the ZooKeeper node ID to `1`.

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=1" >> /etc/default/serviced
```

4 Specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

5 Specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

You may copy the following of text and paste it in your console:

```
q1="1@0.0.0.0:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Clean up the Control Center configuration file.

a Open `/etc/default/serviced` with a text editor.

b Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

The value of this declaration specifies 3 hosts.

c Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

The value of this declaration is just the master host.

d Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.

e Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.

f Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

g Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.

h Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.

i Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

j Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.

k Save the file, and then close the text editor.

7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | egrep '(_ZOO|_ZK)'
```

Configuring a resource pool host as a ZooKeeper node

To perform this procedure, you need a resource pool host with an XFS file system on a separate partition, created previously.

This procedure configures a ZooKeeper ensemble on a resource pool host. Repeat this procedure on each Control Center resource pool host to add to the ZooKeeper ensemble.

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

2 Create a variable for each Control Center host to include in the ZooKeeper ensemble.

The variables are used in subsequent steps.

Note Define the variables identically on the master host and on each resource pool host.

Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=Master-Host-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

Note ZooKeeper requires IP addresses for ensemble configuration.

3 Set the ID of this node in the ZooKeeper ensemble.

For *Pool-Host-A-IP* (**node2**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=2" >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=3" >> /etc/default/serviced
```

4 Specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

5 Specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

For *Pool-Host-A-IP* (**node2**), use the following commands:

```
q1="1@${node1}:2888:3888"
```

```
q2="2@0.0.0.0:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@0.0.0.0:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Set the *SERVICED_ISVCS_START* variable, and clean up the Control Center configuration file.
   a Open `/etc/default/serviced` with a text editor.
   b Locate the *SERVICED_ISVCS_START* variable, and then delete all but `zookeeper` from its list of values.
   c Remove the number sign character (#) from the beginning of the line.
   d Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

      The value of this declaration specifies 3 hosts.
   e Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

      The value of this declaration is just the master host.
   f Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.
   g Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.
   h Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   i Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.
   j Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.
   k Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   l Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.
   m Save the file, and then close the text editor.
7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced \
  | egrep '(_ZOO|_ZK|_STA)'
```

8 Pull the required Control Center ZooKeeper image from the master host.
   a Identify the image to pull.

```
serviced version | grep IsvcsImages
```

   Example result:

```
IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
```

b Pull the Control Center ZooKeeper image.

Replace *Isvcs-ZK-Image* with the name and version number of the ZooKeeper image from the previous substep:

```
docker pull Isvcs-ZK-Image
```

Starting a ZooKeeper ensemble

This procedure starts a ZooKeeper ensemble.

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

1. Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the second node of the ZooKeeper ensemble (*Pool-Host-A-IP*).
3. In another separate window, log in to the third node of the ZooKeeper ensemble (*Pool-Host-B-IP*).
4. On all ensemble hosts, stop and start `serviced`.

```
systemctl stop serviced && systemctl start serviced
```

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

```
{ echo stats; sleep 1; } | nc localhost 2181 | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-A-IP 2181  | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-B-IP 2181  | grep Mode
```

   If `nc` is not available, you can use `telnet` with *interactive ZooKeeper commands*.
6. Optional: Log in to the Control Center browser interface, and then start Resource Manager and related applications, if desired.

   The next procedure requires stopping Resource Manager.

Updating resource pool hosts

The default configuration of resource pool hosts sets the value of the *SERVICED_ZK* variable to the master host only. This procedure updates the setting to include the full ZooKeeper ensemble.

Perform this procedure on each resource pool host in your Control Center cluster.

1. Log in to the resource pool host as `root`, or as a user with superuser privileges.
2. Update the variable.
   a Open `/etc/default/serviced` in a text editor.
   b Locate the *SERVICED_ZK* declaration, and then replace its value with the same value used in the ZooKeeper ensemble nodes.
   c Save the file, and then close the editor.
3. Restart Control Center.

```
systemctl restart serviced
```

Adding hosts to the default resource pool

Note If you are installing a single-host deployment, skip this section.

This procedure adds one or more resource pool hosts to the `default` resource pool.

1. Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2. Add a resource pool host.

   Replace *Hostname-Or-IP* with the hostname or IP address of the resource pool host to add:

   ```
   serviced host add Hostname-Or-IP:4979 default
   ```

   If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.
3. Repeat the preceding command for each resource pool host in your Control Center cluster.

Deploying Resource Manager

This procedure adds the Resource Manager application to the list of applications that Control Center manages, and pulls application images from Docker Hub.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Add the `Zenoss.resmgr` application to Control Center.

   ```
   myPath=/opt/serviced/templates
   serviced template add $myPath/zenoss-resmgr-*.json
   ```

   On success, the `serviced` command returns the template ID.
3. Deploy the application.

   Replace *Template-ID* with the template identifier returned in the previous step, and replace *Deployment-ID* with a name for this deployment (for example, `Dev` or `Test`):

   ```
   serviced template deploy Template-ID default Deployment-ID
   ```

   Control Center pulls Resource Manager images into the local registry. To monitor progress, enter the following command:

   ```
   journalctl -flu serviced -o cat
   ```

Control Center and Resource Manager are now installed, and Resource Manager is ready to be configured for your environment. For more information, refer to the *Zenoss Resource Manager Configuration Guide*.

Installing without internet access 2

The procedures in this chapter install Control Center and Resource Manager on one or more Red Hat Enterprise Linux (RHEL) 7.1 or 7.2 hosts, or one or more CentOS 7.1 or 7.2 hosts. The procedures in this chapter support hosts that do not have internet access.

You may create a single-host or a multi-host deployment. For production use, Zenoss strongly recommends creating a multi-host deployment that includes a minimum of three real or virtual machines. For more information about deploying Control Center and Resource Manager, refer to the *Zenoss Resource Manager Planning Guide*.

Control Center requires a common time source. If you have an NTP time server inside your firewall, you may configure the hosts in your Control Center cluster to use it. If not, then you may use the procedures in this chapter to configure an NTP time server on the Control Center master host, and to configure all the other cluster hosts to synchronize with the master. However, the procedures require IP addresses. **Therefore, all of the hosts in your Control Center cluster require static IP addresses**.

Note For optimal results, review this chapter thoroughly before starting the installation process.

Installing a master host

Perform the procedures in this section to install Control Center and Resource Manager on a master host.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center master host.

To configure a private NTP cluster, the Control Center master host must have a static IP address.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

   ```
   uname -m
   ```

   - If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
   - If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.
3. Verify that the host's numeric identifier is unique.

Each host in a Control Center cluster must have a unique host identifier.

```
hostid
```

4 Determine whether the available, unused storage is sufficient.
   a Display the available storage devices.

   ```
   lsblk --output=NAME,SIZE
   ```

   b Compare the available storage with the amount required for a Control Center master host.
   For more information, refer to the *Zenoss Resource Manager Planning Guide*.
5 Determine whether the available memory and swap is sufficient.
   a Display the available memory.

   ```
   free -h
   ```

   b Compare the available memory with the amount required for a master host in your deployment.
   For more information, refer to the *Zenoss Resource Manager Planning Guide*.
6 Verify the operating system release.

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

If the result includes `7.0`, select another host or update the operating system.

Preparing storage for the master host

In addition to the storage required for its operating system, a Control Center master host requires the following storage areas:

- A local partition for Docker data, configured as a device mapper thin pool.
- A local partition for Control Center internal services data, formatted with the XFS file system.

Note Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

- A local or remote primary partition for Resource Manager data, configured as a device mapper thin pool.
- A local primary partition, a remote primary partition, or a remote file server, for backups of Resource Manager data. The local or remote primary partition is formatted with the XFS file system. A remote file server must provide a file system that is compatible with XFS.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

For storage sizing information, refer to the *Zenoss Resource Manager Planning Guide*.

For device mapper thin pools, no formatting is required—simply create primary partitons, which are configured in subsequent procedures. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

To create the required storage, perform the following procedures.

Note Data present on the primary partitions you select are destroyed in these procedure. Please ensure that data is backed up elsewhere, or no longer needed, before proceeding.

Creating a file system for internal services

This procedure creates an XFS file system on a primary partition. For more information about primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

Note Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Identify the target primary partition for the file system to create.

   ```
   lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
   ```

   For more information about the output of the `lsblk` command, and about creating primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.
3. Create an XFS file system.
   Replace *Partition* with the path of the target primary partition:

   ```
   mkfs -t xfs Partition
   ```
4. Add an entry to the `/etc/fstab` file.
   Replace *Partition* with the path of the primary partition used in the previous step:

   ```
   echo "Partition \
     /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
   ```
5. Create the mount point for internal services data.

   ```
   mkdir -p /opt/serviced/var/isvcs
   ```
6. Mount the file system, and then verify it mounted correctly.

   ```
   mount -a && mount | grep isvcs
   ```

   Example result:

   ```
   /dev/xvdb1 on /opt/serviced/var/isvcs type xfs
    (rw,relatime,seclabel,attr2,inode64,noquota)
   ```

Creating a file system for backups

To perform this procedure, you need a host with at least one unused primary partition, or a remote file server.

The Control Center master host requires local or remote storage space for backups of Control Center data. This procedure includes steps to create an XFS file system on a primary partition, if necessary, and steps to mount a file system for backups. For more information about primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Optional: Identify the target primary partition for the file system to create, if necessary.

Skip this step if you are using a remote file server.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

For more information about the output of the `lsblk` command, and about creating primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

3 Optional: Create an XFS file system, if necessary.

Skip this step if you are using a remote file server.

Replace *Partition* with the path of the target primary partition:

```
mkfs -t xfs Partition
```

4 Create an entry in the `/etc/fstab` file.

Replace *File-System-Specification* with one of the following values:

- the path of the primary partition used in the previous step
- the remote server specification

```
echo "File-System-Specification \
  /opt/serviced/var/backups xfs defaults 0 0" >> /etc/fstab
```

5 Create the mount point for backup data.

```
mkdir -p /opt/serviced/var/backups
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep backups
```

Example result:

```
/dev/sdb3 on /opt/serviced/var/backups type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Downloading files for offline installation

This procedure describes how to download RPM packages and Docker image files to your workstation.

To perform this procedure, you need:

- A workstation with internet access.
- A portable storage medium, such as a USB flash drive, with at least 5 GB of free space.
- Permission to download the required files from the *File Portal - Download Zenoss Enterprise Software* site. You may request permission by filing a ticket at the *Zenoss Support* site.

1 In a web browser, navigate to the *File Portal - Download Zenoss Enterprise Software* site.
2 Log in with the account provided by Zenoss Support.
3 Download archive files to your workstation.

Replace *Version* with the most recent version number available on the download page:

- `install-zenoss-hbase:v`*`Version`*`.run`
- `install-zenoss-isvcs-zookeeper:v`*`Version`*`.run`
- `install-zenoss-opentsdb:v`*`Version`*`.run`
- `install-zenoss-resmgr_5.1:5.1`*`Version`*`.run`

- `install-zenoss-serviced-isvcs:v`*`Version`*`.run`
- `serviced-resource-agents-`*`Version`*`.x86_64.rpm`

4 Download the RHEL/CentOS mirror package for your upgrade.

Note If you are planning to upgrade the operating system during your Control Center and Resource Manager upgrade, choose the mirror package that matches the RHEL/CentOS release to which you are upgrading, not the release that is installed now.

Replace *Version* with the most recent version number available on the download page, and replace *Release* with the version of RHEL/CentOS appropriate for your environment:

```
yum-mirror-centos7.Release-Version.x86_64.rpm
```

5 Copy the files to your portable storage medium.

Staging files for offline installation

Before performing this procedure, verify that approximately 4GB of temporary space is available on the file system where `/root` is located.

This procedure adds files for offline installation to the Control Center master host. The staged files are required in subsequent procedures.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Copy the archive files from your portable storage medium to `/root`.
3 Set the file permissions of the self-extracting archive files to execute.

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

4 Change directory to `/root`.

```
cd /root
```

5 Install the Resource Manager repository mirror.

```
yum install -y ./yum-mirror-*.x86_64.rpm
```

6 Optional: Delete the package file, if desired.

```
rm ./yum-mirror-*.x86_64.rpm
```

Preparing the master host operating system

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center master host.

1 Log in to the candidate master host as `root`, or as a user with superuser privileges.
2 Add an entry to `/etc/hosts` for localhost, if necessary.
   a Determine whether `127.0.0.1` is mapped to `localhost`.

```
grep 127.0.0.1 /etc/hosts | grep localhost
```

   If the preceding commands return no result, perform the following substep.
   b Add an entry to `/etc/hosts` for `localhost`.

```
echo "127.0.0.1   localhost" >> /etc/hosts
```

3 Disable the firewall, if necessary.

This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

- If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
- If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

4 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

5 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations. **If you are installing a single-host deployment, skip this step**.

```
yum --enablerepo=zenoss-mirror install -y nmap-ncat
```

8 Reboot the host.

```
reboot
```

Configuring an NTP master server

This procedure configures an NTP master server on the Control Center master host. If you have an NTP time server inside your firewall, you may configure the master host to use it; however, this procedure does not include that option.

1. Log in the Control Center master host as `root`, or as a user with superuser privileges.
2. Install the NTP package.

```
yum --enablerepo=zenoss-mirror install -y ntp
```

3. Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

4. Edit the NTP configuration file./
   a. Open `/etc/ntp.conf` with a text editor.
   b. Replace all of the lines in the file with the following lines:

```
# Use the local clock
server 127.127.1.0 prefer
fudge  127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift
broadcastdelay 0.008

# Give localhost full access rights
restrict 127.0.0.1

# Grant access to client hosts
restrict ADDRESS_RANGE mask NETMASK nomodify notrap
```

   c. Replace `ADDRESS_RANGE` with the range of IPv4 network addresses that are allowed to query this NTP server.

      For example, the following IP addresses are assigned to the hosts in a Control Center cluster:

      ```
      203.0.113.10
      203.0.113.11
      203.0.113.12
      203.0.113.13
      ```

      For the preceding addresses, the value for `ADDRESS_RANGE` is `203.0.113.0`.
   d. Replace `NETMASK` with the IPv4 network mask that corresponds with the address range.
      For example, the network mask for `203.0.113.0` is `255.255.255.0`.
   e. Save the file and exit the editor.
5. Enable and start the NTP daemon.
   a. Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

   b. Configure `ntpd` to start when the system starts.

      Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

c Start `ntpd`.

```
systemctl start ntpd
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install --enablerepo=zenoss-mirror -y docker-engine
```

3 Create a `systemd` override file for the Docker service definition.
   a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

   b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

   c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

4 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-mirror install -y serviced
```

5 Create a device mapper thin pool for Docker data.
   a Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

   b Create the thin pool.

   Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

   On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.
6 Configure and start the Docker service.
   a Create variables for adding arguments to the Docker configuration file.

   The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

7 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

d Restart the Docker service.

```
systemctl restart docker
```

8 Import the Control Center and Resource Manager images into the local `docker` registry.

The images are contained in the self-extracting archive files that are staged in the `/root` directory.

a Change directory to `/root`.

```
cd /root
```

b Extract the images.

```
for image in install-*.run
do
  echo -n "$image: "
  eval ./$image
done
```

Image extraction begins when you press the **y** key. If you press the **y** key and then **Return** key, the current image is extracted, but the next one is not.

c Optional: Delete the archive files, if desired.

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

Installing Resource Manager

This procedure installs Resource Manager and configures the NFS server.

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

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-mirror install -y zenoss-resmgr-service
```

3. Configure and restart the NFS server.

   Currently, *an unresolved issue* prevents the NFS server from starting correctly. The following commands provide a workaround to ensure that it does.

   a Open `/lib/systemd/system/nfs-server.service` with a text editor.

   b Change `rpcbind.target` to `rpcbind.service` on the following line:

```
Requires= network.target proc-fs-nfsd.mount rpcbind.target
```

   c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

Configuring Control Center

This procedure creates a thin pool for application data and customizes key configuration variables of Control Center.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Configure Control Center to serve as the master and as an agent.

   The following variables configure `serviced` to serve as both master and agent:

   `SERVICED_AGENT`

   Default: `0` (false)

   Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

   `SERVICED_MASTER`

   Default: `0` (false)

   Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

   a Open `/etc/default/serviced` in a text editor.

   b Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.

The following example shows the line to change:

```
# SERVICED_AGENT=0
```

c Remove the number sign character (#) from the beginning of the line.

d Find the *SERVICED_MASTER* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_MASTER=0
```

e Remove the number sign character (#) from the beginning of the line.

f Save the file, and then close the editor.

3 Create a thin pool for Resource Manager data.

a Identify the primary partition for the thin pool to create, and the amount of space available on the primary partition.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

For more information about the output of the `lsblk` command and primary partitions, refer to the *Zenoss Resource Manager Planning Guide*.

b Create a variable for 50% of the space available on the primary partition for the thin pool to create.

The thin pool stores application data and snapshots of the data. You can add storage to the pool at any time.

Replace *Half-Of-Available-Space* with 50% of the space available in the primary partition, in gigabytes. Include the symbol for gigabytes (`G`) after the numeric value.

```
myFifty=Half-Of-Available-SpaceG
```

c Create the thin pool.

Replace *Path-To-Device* with the path of the target primary partition:

```
serviced-storage create-thin-pool -o dm.basesize=$myFifty \
  serviced Path-To-Device
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

4 Configure Control Center with the name of the thin pool for Resource Manager data.

The Control Center configuration file is `/etc/default/serviced`. (For more information about `serviced` configuration options, refer to the Control Center online help.)

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_FS_TYPE* declaration.

c Remove the number sign character (#) from the beginning of the line.

d Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

Replace *Thin-Pool-Name* with the name of the thin pool created previously:

```
SERVICED_DM_THINPOOLDEV=Thin-Pool-Name
```

e Save the file, and then close the editor.

5 Optional: Specify an alternate private subnet for Control Center, if necessary.

The default private subnet may already be in use in your environment. The following variable configures `serviced` to use an alternate subnet:

SERVICED_VIRTUAL_ADDRESS_SUBNET

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

User access control

Control Center provides a browser interface and a command-line interface.

To gain access to the Control Center browser interface, users must have login accounts on the Control Center master host. (Pluggable Authentication Modules (PAM) is supported.) In addition, users must be members of the Control Center administrative group, which by default is the system group, `wheel`. To enhance security, you may change the administrative group from `wheel` to any non-system group.

To use the Control Center command-line interface, users must have login accounts on the Control Center master host, and be members of the `docker` user group. Members of the `wheel` group, including `root`, are members of the `docker` group.

Adding users to the default administrative group

This procedure adds users to the default administrative group of Control Center, `wheel`. Performing this procedure enables users with superuser privileges to gain access to the Control Center browser interface.

Note Perform this procedure or the next procedure, but not both.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Add users to the system group, `wheel`.

   Replace *User* with the name of a login account on the master host.

   ```
   usermod -aG wheel User
   ```

   Repeat the preceding command for each user to add.

Note For information about using Pluggable Authentication Modules (PAM), refer to your operating system documentation.

Configuring a regular group as the Control Center administrative group

This procedure changes the default administrative group of Control Center from `wheel` to a non-system group.

Note Perform this procedure or the previous procedure, but not both.

1. Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2. Create a variable for the group to designate as the administrative group.

In this example, the name of group to create is `serviced`. You may choose any name or use an existing group.

```
GROUP=serviced
```

3 Create a new group, if necessary.

```
groupadd $GROUP
```

4 Add one or more existing users to the new administrative group.

Replace *User* with the name of a login account on the host:

```
usermod -aG $GROUP User
```

Repeat the preceding command for each user to add.

5 Specify the new administrative group in the `serviced` configuration file.

The following variable specifies the administrative group:

SERVICED_ADMIN_GROUP

Default: `wheel`

The name of the Linux group on the Control Center master host whose members are authorized to use the Control Center browser interface. You may replace the default group with a group that does not have superuser privileges.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ADMIN_GROUP* declaration, and then change the value from `wheel` to the name of the group you chose earlier.
The following example shows the line to change:

```
# SERVICED_ADMIN_GROUP=wheel
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

6 Optional: Prevent `root` users and members of the `wheel` group from gaining access to the Control Center browser interface, if desired.

The following variable controls privileged logins:

SERVICED_ALLOW_ROOT_LOGIN

Default: `1` (true)

Determines whether `root`, or members of the `wheel` group, may gain access to the Control Center browser interface.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ALLOW_ROOT_LOGIN* declaration, and then change the value from `1` to `0`.
The following example shows the line to change:

```
# SERVICED_ALLOW_ROOT_LOGIN=1
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

Enabling use of the command-line interface

This procedure enables users to perform administrative tasks with the Control Center command-line interface by adding individual users to the `docker` group.

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

2 Add users to the Docker group, `docker`.

   Replace *User* with the name of a login account on the host.

   ```
   usermod -aG docker User
   ```

   Repeat the preceding command for each user to add.

Starting Control Center

This procedure starts the Control Center service, `serviced`.

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

   ```
   systemctl start serviced
   ```

   To monitor progress, enter the following command:

   ```
   journalctl -flu serviced -o cat
   ```

   The Control Center browser and command-line interfaces are unavailable until the Control Center images are tagged and the internal services are started. The process takes approximately 3 minutes. When the message `Trying to discover my pool` repeats, Control Center is ready for the next steps.

3 **Note** Perform this step **only if you are installing a single-host deployment**.

   Optional: Add the master host to the `default` resource pool.

   Replace *Hostname-Or-IP* with the hostname or IP address of the Control Center master host:

   ```
   serviced host add Hostname-Or-IP:4979 default
   ```

   If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Isolating the master host in a separate resource pool

Note If you are configuring a single-host deployment, skip this procedure.

Control Center enables or just performs rapid recovery from application service failures. When Control Center internal services and application services share a host, application failures can limit recovery options. Zenoss strongly recommends isolating the Control Center master host in a separate resource pool.

This procedure creates a new resource pool for the Control Center master host, and then adds the master host to the pool.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Create a new resource pool named `master`.

   ```
   serviced pool add master
   ```

3 Add the master host to the `master` resource pool.

Replace *Hostname-Or-IP* with the hostname or IP address of the Control Center master host:

```
serviced host add Hostname-Or-IP:4979 master
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Installing resource pool hosts

Note If you are installing a single-host deployment, skip this section.

Control Center resource pool hosts run the application services scheduled for the resource pool to which they belong, and for which they have sufficent RAM and CPU resources.

Resource Manager has two broad categories of application services: Infrastructure and collection. The services associated with each category can run in the same resource pool, or can run in separate resource pools.

For improved reliability, two resource pool hosts are configured as nodes in an *Apache ZooKeeper* ensemble. The storage required for ensemble hosts is slightly different than the storage required for all other resource pool hosts: Each ensemble host requires a separate primary partition for Control Center internal services data, in addition to the primary partition for Docker data. Unless the ZooKeeper service on the Control Center master host fails, their roles in the ZooKeeper ensemble do not affect their roles as Control Center resource pool hosts.

Note The hosts for the ZooKeeper ensemble require static IP addresses, because ZooKeeper does not support hostnames in its configurations. Likewise, to configure a private NTP cluster, all resource pool hosts must have static IP addresses.

Repeat the procedures in the following sections for each host you wish to add to your Control Center deployment.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

   ```
   uname -m
   ```

   - If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
   - If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.
3. Verify that name resolution works on this host.

   ```
   hostname -i
   ```

   If the result is not a valid IPv4 adddress, add an entry for the host to the network nameserver, or to `/etc/hosts`.
4. Verify that the host's numeric identifier is unique.

Each host in a Control Center cluster must have a unique host identifier.

```
hostid
```

5 Determine whether the available, unused storage is sufficient.

a Display the available storage devices.

```
lsblk --output=NAME,SIZE
```

b Compare the available storage with the amount required for a resource pool host in your deployment.

In particular, resource pool hosts that are configured as nodes in a ZooKeeper ensemble require an additional primary partition for Control Center internal services data.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

6 Determine whether the available memory and swap is sufficient.

a Display the available memory.

```
free -h
```

b Compare the available memory with the amount required for a resource pool host in your deployment.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

7 Verify the operating system release.

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

If the result includes `7.0`, select another host or upgrade the operating system.

Creating a file system for Control Center internal services

This procedure creates an XFS file system on a primary partition.

Note **Perform this procedure only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**. No other resource pool hosts run Control Center internal services, so no other pool hosts need a partition for internal services data.

---

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

2 Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

3 Create an XFS file system.
Replace *Isvcs-Partition* with the path of the target primary partition:

```
mkfs -t xfs Isvcs-Partition
```

4 Create the mount point for Control Center internal services data.

```
mkdir -p /opt/serviced/var/isvcs
```

5 Add an entry to the `/etc/fstab` file.
Replace *Isvcs-Partition* with the path of the primary partition used in the previous step:

```
echo "Isvcs-Partition \
  /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep isvcs
```

Example result:

```
/dev/xvdb1 on /opt/serviced/var/isvcs type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Staging files for offline installation

To perform this procedure, you need the portable storage medium that contains the archive files used in installing the master host.

This procedure adds files for offline installation to a resource pool host. The files are required in subsequent procedures.

Perform this procedure on each resource pool host in your deployment.

1 Log in to the target host as `root`, or as a user with superuser privileges.
2 Copy `yum-mirror-*.x86_64.rpm` from your portable storage medium to `/tmp`.
3 Install the Resource Manager repository mirror.

```
yum install -y /tmp/yum-mirror-*.x86_64.rpm
```

4 Optional: Delete the package file, if desired.

```
rm /tmp/yum-mirror-*.x86_64.rpm
```

Preparing a resource pool host

Perform this procedure to prepare a RHEL/CentOS 7.1 or 7.2 host as a Control Center resource pool host.

1 Log in to the candidate resource pool host as `root`, or as a user with superuser privileges.
2 Add an entry to `/etc/hosts` for localhost, if necessary.
   a Determine whether `127.0.0.1` is mapped to `localhost`.

```
grep 127.0.0.1 /etc/hosts | grep localhost
```

   If the preceding commands return no result, perform the following substep.
   b Add an entry to `/etc/hosts` for `localhost`.

```
echo "127.0.0.1   localhost" >> /etc/hosts
```

3 Disable the firewall, if necessary.

   This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.
   a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

   - If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
   - If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

4 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

5 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install and configure the NTP package.

a Install the package.

```
yum install -y ntp
```

b Set the system time.

```
ntpd -gq
```

c Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

d Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot. The following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

8 Reboot the host.

```
reboot
```

Configuring an NTP client

This procedure configures a resource pool host to synchronize its clock with the NTP server on the Control Center master host. If you have an NTP time server inside your firewall, you may configure the host to use it; however, this procedure does not include that option.

1 Log in the Control Center resource pool host as `root`, or as a user with superuser privileges.
2 Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

3 Edit the NTP configuration file./
   a Open `/etc/ntp.conf` with a text editor.
   b Replace all of the lines in the file with the following lines:

```
# Point to the master time server
server MASTER_ADDRESS

restrict default ignore
restrict 127.0.0.1
restrict MASTER_ADDRESS mask 255.255.255.255 nomodify notrap noquery

driftfile /var/lib/ntp/drift
```

   c Replace both instances of `MASTER_ADDRESS` with the IPv4 address of the host where the NTP server is running (the Control Center master host).
   d Save the file and exit the editor.
4 Synchronize the clock with the master server.

```
ntpd -gq
```

5 Enable and start the NTP daemon.
   a Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

   b Configure `ntpd` to start when the system starts.

   Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

   c Start `ntpd`.

```
systemctl start ntpd
```

Creating a file system for Control Center internal services

This procedure creates an XFS file system on a primary partition.

Note **Perform this procedure only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**. No other resource pool hosts run Control Center internal services, so no other pool hosts need a partition for internal services data.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

3. Create an XFS file system.
   Replace *Isvcs-Partition* with the path of the target primary partition:

```
mkfs -t xfs Isvcs-Partition
```

4. Create the mount point for Control Center internal services data.

```
mkdir -p /opt/serviced/var/isvcs
```

5. Add an entry to the `/etc/fstab` file.
   Replace *Isvcs-Partition* with the path of the primary partition used in the previous step:

```
echo "Isvcs-Partition \
  /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
```

6. Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep isvcs
```

   Example result:

```
/dev/xvdb1 on /opt/serviced/var/isvcs type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

1. Log in to the resource pool host as `root`, or as a user with superuser privileges.
2. Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install --enablerepo=zenoss-mirror -y docker-engine
```

3. Create a symbolic link for the Docker temporary directory.

   Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

   a. Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

   b. Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4. Create a `systemd` override file for the Docker service definition.

a. Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b. Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c. Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5. Install Control Center.

   Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-mirror install -y serviced
```

6. Create a device mapper thin pool for Docker data.

   a. Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

   b. Create the thin pool.

   Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

   On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

7. Configure and start the Docker service.

   a. Create variables for adding arguments to the Docker configuration file.

   The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

   Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

   b. Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

   c. Start or restart Docker.

```
systemctl restart docker
```

   The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

8 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

---

d Restart the Docker service.

```
systemctl restart docker
```

Configuring and starting Control Center

This procedure customizes key configuration variables of Control Center.

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

2 Configure Control Center as an agent of the master host.

The following variable configures `serviced` to serve as agent:

`SERVICED_AGENT`

Default: 0 (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: 0 (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

In addition, the following lines need to be edited, to replace `{{SERVICED_MASTER_IP}}` with the IP address of the master host:

```
# SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
# SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
# SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
# SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
```

```
# SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
# SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
```

a. Open `/etc/default/serviced` in a text editor.
b. Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_AGENT=0
```

c. Remove the number sign character (#) from the beginning of the line.
d. Find the *SERVICED_MASTER* declaration, and then remove the number sign character (#) from the beginning of the line.
e. Globally replace `{{SERVICED_MASTER_IP}}` with the IP address of the master host.

Note Remove the number sign character (#) from the beginning of each variable declaration that includes the master IP address.

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

3. Optional: Specify an alternate private subnet for Control Center, if necessary.

The default private subnet may already be in use in your environment. The following variable configures `serviced` to use an alternate subnet:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a. Open `/etc/default/serviced` in a text editor.
b. Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value.
The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c. Remove the number sign character (#) from the beginning of the line.
d. Save the file, and then close the editor.

4. Start the Control Center service (`serviced`).

```
systemctl start serviced
```

To monitor progress, enter the following command:

```
journalctl -flu serviced -o cat
```

To install additional resource pool hosts, return to *Verifying candidate host resources* on page 49.

ZooKeeper ensemble configuration

Note If you are installing a single-host deployment, or if your deployment includes fewer than two resource pool hosts, skip this section.

Control Center relies on *Apache ZooKeeper* to coordinate its services. The procedures in this section create a ZooKeeper ensemble of 3 nodes. To perform these procedures, you need a Control Center master host and a minimum of two resource pool hosts. Each resource pool host requires a separate primary partition for Control

Center internal services, and each should have a static IP address. For more information about storage requirements, refer to the *Zenoss Resource Manager Planning Guide*.

Note Zenoss strongly recommends configuring a ZooKeeper ensemble for all production deployments.

A ZooKeeper ensemble requires a minimum of 3 nodes, and 3 nodes is sufficient for most deployments. A 5-node configuration improves failover protection during maintenance windows. Ensembles larger than 5 nodes are not necessary. An odd number of nodes is recommended, and an even number of nodes is strongly discouraged.

Note The Control Center ZooKeeper service requires consistently fast storage. Ideally, the primary partition for Control Center internal services is on a separate, high-performance device that has only one primary partition.

Control Center variables for ZooKeeper

This tables in this section associates the ZooKeeper-related Control Center variables to set in `/etc/default/serviced` with the roles that hosts play in a Control Center cluster.

Table 4: Control Center master host

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 1

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 5: Control Center resource pool host and ZooKeeper ensemble node

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 2 or 3

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ISVCS_START

The list of Control Center internal services to start and run on hosts other than the master host.
Value: `zookeeper`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.

Value: *`IP-Address`*`:2181,...`

Table 6: Control Center resource pool host only

SERVICED_ZK The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it. Value: *`IP-Address`*`:2181,...`

Configuring the master host as a ZooKeeper node

This procedure configures the Control Center master host as a member of the ZooKeeper ensemble.

Note For accuracy, this procedure constructs Control Center configuration variables in the shell and appends them to `/etc/default/serviced`. The last step is to move the variables from the end of the file to more appropriate locations.

1. Log in to the master host as `root`, or as a user with superuser privileges.
2. Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

   Note Define the variables identically on the master host and on each resource pool host.

   Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

   ```
   node1=Master-Host-IP
   node2=Pool-Host-A-IP
   node3=Pool-Host-B-IP
   ```

   Note ZooKeeper requires IP addresses for ensemble configuration.
3. Set the ZooKeeper node ID to `1`.

   ```
   echo "SERVICED_ISVCS_ZOOKEEPER_ID=1" >> /etc/default/serviced
   ```
4. Specify the nodes in the ZooKeeper ensemble.
   You may copy the following text and paste it in your console:

   ```
   echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
     >> /etc/default/serviced
   ```
5. Specify the nodes in the ZooKeeper quorum.

   ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

   You may copy the following of text and paste it in your console:

   ```
   q1="1@0.0.0.0:2888:3888"
   ```

```
q2="2@${node2}:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Clean up the Control Center configuration file.
   a Open `/etc/default/serviced` with a text editor.
   b Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

      The value of this declaration specifies 3 hosts.
   c Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

      The value of this declaration is just the master host.
   d Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.
   e Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.
   f Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   g Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.
   h Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.
   i Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   j Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.
   k Save the file, and then close the text editor.
7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | egrep '(_ZOO|_ZK)'
```

Configuring a resource pool host as a ZooKeeper node

To perform this procedure, you need a resource pool host with an XFS file system on a separate partition, created previously.

This procedure configures a ZooKeeper ensemble on a resource pool host. Repeat this procedure on each Control Center resource pool host to add to the ZooKeeper ensemble.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

Note Define the variables identically on the master host and on each resource pool host.

Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=Master-Host-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

Note ZooKeeper requires IP addresses for ensemble configuration.

3 Set the ID of this node in the ZooKeeper ensemble.

For *Pool-Host-A-IP* (**node2**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=2" >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=3" >> /etc/default/serviced
```

4 Specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

5 Specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

For *Pool-Host-A-IP* (**node2**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@0.0.0.0:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@0.0.0.0:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Set the *SERVICED_ISVCS_START* variable, and clean up the Control Center configuration file.

a Open `/etc/default/serviced` with a text editor.

b Locate the *SERVICED_ISVCS_START* variable, and then delete all but `zookeeper` from its list of values.

c Remove the number sign character (#) from the beginning of the line.

d Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

The value of this declaration specifies 3 hosts.

e Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

The value of this declaration is just the master host.

f Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.

g Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.

h Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

i Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.

j Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.

k Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

l Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.

m Save the file, and then close the text editor.

7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced \
  | egrep '(_ZOO|_ZK|_STA)'
```

8 Pull the required Control Center ZooKeeper image from the master host.

a Identify the image to pull.

```
serviced version | grep IsvcsImages
```

Example result:

```
IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
```

b Pull the Control Center ZooKeeper image.

Replace *Isvcs-ZK-Image* with the name and version number of the ZooKeeper image from the previous substep:

```
docker pull Isvcs-ZK-Image
```

Starting a ZooKeeper ensemble

This procedure starts a ZooKeeper ensemble.

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

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

2 In a separate window, log in to the second node of the ZooKeeper ensemble (*Pool-Host-A-IP*).

3 In another separate window, log in to the third node of the ZooKeeper ensemble (*Pool-Host-B-IP*).

4 On all ensemble hosts, stop and start `serviced`.

```
systemctl stop serviced && systemctl start serviced
```

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

```
{ echo stats; sleep 1; } | nc localhost 2181 | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-A-IP 2181  | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-B-IP 2181  | grep Mode
```

If `nc` is not available, you can use `telnet` with *interactive ZooKeeper commands*.

6 Optional: Log in to the Control Center browser interface, and then start Resource Manager and related applications, if desired.

The next procedure requires stopping Resource Manager.

Updating resource pool hosts

The default configuration of resource pool hosts sets the value of the *SERVICED_ZK* variable to the master host only. This procedure updates the setting to include the full ZooKeeper ensemble.

Perform this procedure on each resource pool host in your Control Center cluster.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Update the variable.
   a Open `/etc/default/serviced` in a text editor.
   b Locate the *SERVICED_ZK* declaration, and then replace its value with the same value used in the ZooKeeper ensemble nodes.
   c Save the file, and then close the editor.
3 Restart Control Center.

```
systemctl restart serviced
```

Adding hosts to the default resource pool

Note If you are installing a single-host deployment, skip this section.

This procedure adds one or more resource pool hosts to the `default` resource pool.

1 Log in to the Control Center master host as `root`, or as a user with superuser privileges.
2 Add a resource pool host.

   Replace *Hostname-Or-IP* with the hostname or IP address of the resource pool host to add:

```
serviced host add Hostname-Or-IP:4979 default
```

   If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.
3 Repeat the preceding command for each resource pool host in your Control Center cluster.

Deploying Resource Manager

This procedure adds the Resource Manager application to the list of applications that Control Center manages.

1 Log in to the master host as `root`, or as a user with superuser privileges.
2 Add the `Zenoss.resmgr` application to Control Center.

```
myPath=/opt/serviced/templates
serviced template add $myPath/zenoss-resmgr-*.json
```

   On success, the `serviced` command returns the template ID.
3 Deploy the application.

Replace *Template-ID* with the template identifier returned in the previous step, and replace *Deployment-ID* with a name for this deployment (for example, `Dev` or `Test`):

```
serviced template deploy Template-ID default Deployment-ID
```

Control Center tags the Resource Manager images in the local registry. To monitor progress, enter the following command:

```
journalctl -flu serviced -o cat
```

Control Center and Resource Manager are now installed, and Resource Manager is ready to be configured for your environment. For more information, refer to the *Zenoss Resource Manager Configuration Guide*.

Part II: High-availability deployments

The chapters in this part describe how to install Control Center and Resource Manager on real or virtual hosts, with or without internet access, in a high-availability deployment. The instructions include the full range of options for customizing your deployment for your environment.

Creating a high-availability deployment with internet access 1

The procedures in this chapter create a high-availability deployment of Control Center and Resource Manager on Red Hat Enterprise Linux (RHEL) 7.1 or 7.2 hosts, or on CentOS 7.1 or 7.2 hosts. To use the procedures in this chapter, you must have a minimum of four hosts, and all of the hosts must have internet access.

For more information about deploying Control Center and Resource Manager, refer to the *Zenoss Resource Manager Planning Guide*.

Note For optimal results, review this chapter thoroughly before starting the installation process.

Master host storage requirements

In addition to the storage required for its operating system, both Control Center master hosts in the failover cluster require the following storage areas:

- A local primary partition for Docker data, configured as a device mapper thin pool.
- A local primary partition for Control Center internal services data, formatted with the XFS file system.

  Note Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

- A local primary partition for Control Center metadata, formatted with the XFS file system.
- A local primary partition for Resource Manager data, configured as a device mapper thin pool.

Note This chapter includes procedures for configuring and formatting all required storage areas.

In addition, the primary node of the failover cluster requires a local primary partition, a remote primary partition, or a remote file server, for backups of Resource Manager data. The local or remote primary partition is formatted with the XFS file system. A remote file server must provide a file system that is compatible with XFS.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

For storage sizing information, refer to the *Zenoss Resource Manager Planning Guide*.

Key variables used in this chapter

The following tables associate important features of a high-availability deployment with the variables used in this chapter.

Feature	Variable Name	
	Primary Node	**Secondary Node**
Public IP address of master node (static; known to all machines in the Control Center cluster)	*Primary-Public-IP*	*Secondary-Public-IP*
Public hostname of master node (returned by `uname -n`; resolves to the public IP address)	*Primary-Public-Name*	*Secondary-Public-Name*
Private IP address of master node (static; dual-NIC systems only)	*Primary-Private-IP*	*Secondary-Private-IP*
Private hostname of master node (resolves to the private IP address; dual-NIC systems only)	*Primary-Private-Name*	*Secondary-Private-Name*

Feature	Variable Name
Virtual IP address of the high-availability cluster (static; known enterprise-wide)	*HA-Virtual-IP*
Virtual hostname of the high-availability cluster (known enterprise-wide)	*HA-Virtual-Name*
Public IP address of resource pool host A (static; for ZooKeeper ensemble)	*Pool-Host-A-IP*
Public IP address of resource pool host B (static; for ZooKeeper ensemble)	*Pool-Host-B-IP*
Primary partition for Docker data (not mirrored)	*Docker-Partition*
Primary partition for Control Center internal services data (mirrored)	*Isvcs-Partition*
Primary partition for Control Center metadata (mirrored)	*Metadata-Partition*
Primary partition for Control Center application data (mirrored)	*App-Data-Partition*
Primary partition for Control Center backups (not mirrored)	*Backups-Partition*

Control Center on the master nodes

A high-availability deployment features two Control Center master nodes that are configured for failover. One host is the primary node, and the other host is the secondary node. Their configurations differ somewhat, but are mostly the same.

Perform all of the procedures in this section on the primary node and on the secondary node.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center master host.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

```
uname -m
```

- If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
- If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.

3 Verify that the host's numeric identifier is unique.

Each host in a Control Center cluster must have a unique host identifier.

```
hostid
```

4 Determine whether the available, unused storage is sufficient.

a Display the available storage devices.

```
lsblk --output=NAME,SIZE
```

b Compare the available storage with the amount required for a Control Center master host.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

5 Determine whether the available memory and swap is sufficient.

a Display the available memory.

```
free -h
```

b Compare the available memory with the amount required for a master host in your deployment.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

6 Update the operating system, if necessary.

a Determine which release is installed.

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

If the result includes `7.0`, perform the following substeps.

b Update the operating system.

```
yum update -y
```

c Restart the system.

```
reboot
```

Preparing the master host operating system

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center master host.

Perform this procedure on the primary node and on the secondary node.

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

2 Add an entry to `/etc/hosts` for localhost, if necessary.

a Determine whether `127.0.0.1` is mapped to `localhost`.

```
grep 127.0.0.1 /etc/hosts | grep localhost
```

If the preceding commands return no result, perform the following substep.

b Add an entry to `/etc/hosts` for `localhost`.

```
echo "127.0.0.1   localhost" >> /etc/hosts
```

3 Add the required hostnames and IP addresses of both the primary and the seconday node to the `/etc/hosts` file.

For a dual-NIC system, replace each variable name with the values designated for each node, and replace `example.com` with the domain name of your orgnanization:

```
echo "Primary-Public-IP Primary-Public-Name.example.com \
  Primary-Public-Name" >> /etc/hosts
echo "Primary-Private-IP Primary-Private-Name.example.com \
  Primary-Private-Name" >> /etc/hosts
echo "Secondary-Public-IP Secondary-Public-Name.example.com \
  Secondary-Public-Name" >> /etc/hosts
echo "Secondary-Private-IP Secondary-Private-Name.example.com \
  Secondary-Private-Name" >> /etc/hosts
```

For a single-NIC system, replace each variable name with the values designated for each node, and replace `example.com` with the domain name of your orgnanization:

```
echo "Primary-Public-IP Primary-Public-Name.example.com \
  Primary-Public-Name" >> /etc/hosts
echo "Secondary-Public-IP Secondary-Public-Name.example.com \
  Secondary-Public-Name" >> /etc/hosts
```

4 Disable the firewall, if necessary.

This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

- If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
- If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

5 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

6 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

7 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

8 Install and configure the NTP package.

a Install the package.

```
yum install -y ntp
```

b Set the system time.

```
ntpd -gq
```

c Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

d Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot. The following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

9 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations.

```
yum install -y nmap-ncat
```

10 Install the Zenoss repository package.

a Install the package.

```
rpm -ivh http://get.zenoss.io/yum/zenoss-repo-1-1.x86_64.rpm
```

b Clean out the `yum` cache directory.

```
yum clean all
```

11 Remove any file system signature from the required primary partitions.

Replace each variable name with the path of the primary partition designated for each storage area:

```
wipefs -a Docker-Partition
wipefs -a Isvcs-Partition
wipefs -a Metadata-Partition
wipefs -a App-Data-Partition
```

12 Add mount points for XFS file systems, which are created in subsequent steps.

```
mkdir -p /opt/serviced/var/isvcs /opt/serviced/var/volumes
```

13 Reboot the host.

```
reboot
```

Configuring a storage area for backups

The Control Center master host requires local or remote storage space for backups of Control Center data. This procedure includes steps to create an XFS file system on a primary partition, if necessary, and steps to mount a file system for backups. For more information about backups, refer to the *Zenoss Resource Manager Planning Guide*.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

Perform this procedure on the primary node and on the secondary node.

1 Log in to the primary node as `root`, or as a user with superuser privileges.
2 Optional: Remove any file system signature from the primary partition for Control Center backups, if necessary.
If you are using a remote file server for backups, skip this step.

Replace *Backups-Partition* with the path of the primary partition designated for Control Center backups:

```
wipefs -a Backups-Partition
```

3 Optional: Create an XFS file system, if necessary.
Skip this step if you are using a remote file server.
Replace *Backups-Partition* with the path of the primary partition designated for Control Center backups:

```
mkfs.xfs Backups-Partition
```

4 Create an entry in the `/etc/fstab` file.

Replace *File-System-Specification* with one of the following values:

- the path of *Backups-Partition*, used in the previous step
- the remote server specification

```
echo "File-System-Specification \
  /opt/serviced/var/backups xfs defaults 0 0" >> /etc/fstab
```

5 Create the mount point for backup data.

```
mkdir -p /opt/serviced/var/backups
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep backups
```

Example result:

```
/dev/sdb3 on /opt/serviced/var/backups type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Install Docker 1.9.0, and then disable accidental upgrades.
   a. Add the Docker repository to the host's repository list.

```
cat > /etc/yum.repos.d/docker.repo <<-EOF
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

   b. Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install -y docker-engine-1.9.0
```

   c. Open `/etc/yum.repos.d/docker.repo` with a text editor.
   d. Change the value of the `enabled` key from `1` to `0`.
   e. Save the file and close the text editor.
3. Create a symbolic link for the Docker temporary directory.

   Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.
   a. Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

   b. Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4. Create a `systemd` override file for the Docker service definition.
   a. Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

   b. Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c. Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5. Install Control Center.

   Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-stable install -y serviced-1.1.7
```

6. Disable automatic startup of Control Center by `systemd`.

   The cluster management software controls the Docker service.

```
systemctl disable serviced
```

7. Create a device mapper thin pool for Docker data.

   Replace *Docker-Partition* with the path of the primary partition designated for Docker data:

```
serviced-storage create-thin-pool docker Docker-Partition
```

   On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

8. Configure and start the Docker service.

   a. Create variables for adding arguments to the Docker configuration file.

      The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

      Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

   b. Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

   c. Start or restart Docker.

```
systemctl restart docker
```

      The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

9. Authenticate to the Docker Hub repository.

   Replace *USER* and *EMAIL* with the values associated with your Docker Hub account.

```
docker login -u USER -e EMAIL
```

   The `docker` command prompts you for your Docker Hub password, and saves a hash of your credentials in the `$HOME/.dockercfg` file (root user account).

10. Configure name resolution in containers.

    Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a. Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b. Open `/etc/sysconfig/docker` in a text editor.

c. Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

d. Restart the Docker service.

```
systemctl restart docker
```

11. Pull the required Control Center images from Docker Hub, and then stop and disable the Docker service.

a. Identify the images to pull.

```
serviced version | grep IsvcsImages
```

Example result:

```
IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
```

b. Pull Control Center images.

Replace *Isvcs-Image-Name* with one of the images named in the output of the previous substep:

```
docker pull Isvcs-Image-Name
```

Repeat the command for each required image.

c. Stop and disable the Docker service.

The cluster management software controls the Docker service.

```
systemctl stop docker && systemctl disable docker
```

Installing Resource Manager

This procedure installs Resource Manager and configures the NFS server.

Perform this procedure on the primary node and on the secondary node.

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

```
yum --enablerepo=zenoss-stable install -y zenoss-resmgr-service
```

3. Configure and disable the NFS service.

Currently, *an unresolved issue* prevents the NFS server from starting correctly. The following commands provide a workaround to ensure that it does.

a Open `/lib/systemd/system/nfs-server.service` with a text editor.

b Change `rpcbind.target` to `rpcbind.service` on the following line:

```
Requires= network.target proc-fs-nfsd.mount rpcbind.target
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

d Stop and disable the NFS service.

The cluster management software controls the NFS service.

```
systemctl stop nfs && systemctl disable nfs
```

Configuring Control Center

This procedure customizes key configuration variables of Control Center.

Perform this procedure on the primary node and on the secondary node.

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

2 Configure Control Center to serve as both master and agent, and to use the virtual IP address of the high-availability cluster.

The following variables define the roles `serviced` can assume:

`SERVICED_AGENT`

Default: `0` (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: `0` (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

In addition, replace `{{SERVICED_MASTER_IP}}` with *HA-Virtual-IP*, the virtual IP address of the high-availability cluster, in the following lines:

```
# SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
# SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
# SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
# SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
# SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
# SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
```

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.

c Remove the number sign character (`#`) from the beginning of the line.

d Locate the *SERVICED_MASTER* declaration, and then change the value from `0` to `1`.

e Remove the number sign character (#) from the beginning of the line.

f Globally replace `{{SERVICED_MASTER_IP}}` with the virtual IP address of the high-availability cluster.

Note Remove the number sign character (#) from the beginning of each variable declaration that includes the IP address.

---

g Save the file, and then close the editor.

3 Configure Control Center to send its responses to the virtual IP address of the high-availability cluster.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_OUTBOUND_IP* declaration, and then change its default value to *HA-Virtual-IP*.

Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster:

```
SERVICED_OUTBOUND_IP=HA-Virtual-IP
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

4 Optional: Specify an alternate private network for Control Center, if necessary.

Control Center requires a 16-bit, private IPv4 network for virtual IP addresses, independent of the private network used in a dual-NIC DRBD configuration. The default network is 10.3/16. If the default network is already in use in your environment, you may select any valid IPv4 16-bit network.

The following variable configures `serviced` to use an alternate network:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

User access control

Control Center provides a browser interface and a command-line interface.

To gain access to the Control Center browser interface, users must have login accounts on the Control Center master host. (Pluggable Authentication Modules (PAM) is supported.) In addition, users must be members of the Control Center administrative group, which by default is the system group, `wheel`. To enhance security, you may change the administrative group from `wheel` to any non-system group.

To use the Control Center command-line interface, users must have login accounts on the Control Center master host, and be members of the `docker` user group. Members of the `wheel` group, including `root`, are members of the `docker` group.

Adding users to the default administrative group

This procedure adds users to the default administrative group of Control Center, `wheel`. Performing this procedure enables users with superuser privileges to gain access to the Control Center browser interface.

Note Perform this procedure or the next procedure, but not both.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Add users to the system group, `wheel`.

   Replace *User* with the name of a login account on the master host.

   ```
   usermod -aG wheel User
   ```

   Repeat the preceding command for each user to add.

   Note For information about using Pluggable Authentication Modules (PAM), refer to your operating system documentation.

Configuring a regular group as the Control Center administrative group

This procedure changes the default administrative group of Control Center from `wheel` to a non-system group.

Note Perform this procedure or the previous procedure, but not both.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Create a variable for the group to designate as the administrative group.

   In this example, the name of group to create is `serviced`. You may choose any name or use an existing group.

   ```
   GROUP=serviced
   ```
3. Create a new group, if necessary.

   ```
   groupadd $GROUP
   ```
4. Add one or more existing users to the new administrative group.

   Replace *User* with the name of a login account on the host:

   ```
   usermod -aG $GROUP User
   ```

   Repeat the preceding command for each user to add.
5. Specify the new administrative group in the `serviced` configuration file.

   The following variable specifies the administrative group:

   `SERVICED_ADMIN_GROUP`

   Default: `wheel`

   The name of the Linux group on the Control Center master host whose members are authorized to use the Control Center browser interface. You may replace the default group with a group that does not have superuser privileges.

   a. Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ADMIN_GROUP* declaration, and then change the value from `wheel` to the name of the group you chose earlier.
The following example shows the line to change:

```
# SERVICED_ADMIN_GROUP=wheel
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

6 Optional: Prevent `root` users and members of the `wheel` group from gaining access to the Control Center browser interface, if desired.

The following variable controls privileged logins:

`SERVICED_ALLOW_ROOT_LOGIN`

Default: `1` (true)

Determines whether `root`, or members of the `wheel` group, may gain access to the Control Center browser interface.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ALLOW_ROOT_LOGIN* declaration, and then change the value from `1` to `0`.
The following example shows the line to change:

```
# SERVICED_ALLOW_ROOT_LOGIN=1
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

Enabling use of the command-line interface

This procedure enables users to perform administrative tasks with the Control Center command-line interface by adding individual users to the `docker` group.

Perform this procedure on the primary node and on the secondary node.

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

2 Add users to the Docker group, `docker`.

Replace *User* with the name of a login account on the host.

```
usermod -aG docker User
```

Repeat the preceding command for each user to add.

Configuring Logical Volume Manager

Control Center application data is managed by a device mapper thin pool created with Logical Volume Manager (LVM). This procedure adjusts the LVM configuration for mirroring by DRBD.

Perform this procedure on the primary node and on the secondary node.

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

2 Edit the LVM configuration file.

a Open `/etc/lvm/lvm.conf` with a text editor.

b Exclude the partition for Control Center application data.

The line to edit is in the `devices` section.

Replace *App-Data-Partition* with the path of the primary partition designated for Control Center application data.

```
filter = ["r|App-Data-Partition|"]
```

c Disable caching and the metadata daemon.

Set the value of the `write_cache_state` and `use_lvmetad` keys to 0.

```
write_cache_state = 0
use_lvmetad = 0
```

d Save the file and close the editor.

3 Delete any stale cache entries.

```
rm -f /etc/lvm/cache/.cache
```

4 Restart the host.

```
reboot
```

Installing DRBD

This procedure installs Distributed Replicated Block Device (DRBD) packages from the RPM repository for enterprise Linux packages, *ELRepo*.

Perform this procedure on the primary node and on the secondary node.

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

2 Add the ELRepo repostiory to the list of repositories.

```
r=http://www.elrepo.org
rpm --import $r/RPM-GPG-KEY-elrepo.org
rpm -Uvh $r/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
yum clean all
```

3 Install DRBD packages.

```
yum install -y drbd84-utils kmod-drbd84
```

DRBD configuration assumptions

The following list identifies the assumptions that inform the DRBD resource definition for Control Center:

- Each node has either one or two NICs. In dual-NIC hosts the private IP/hostnames are reserved for DRBD traffic. This is recommended configuration, which enables real-time writes for disk synchronization between the active and passive nodes, and no contention with application traffic. However, it is possible to use DRBD with a single NIC.
- The default port number for DRBD traffic is 7789.
- All volumes should synchronize and failover together. This is accomplished by creating a single resource definition.
- DRBD stores its metadata on each volume (`meta-disk/internal`), so the total amount of space reported on the logical device `/dev/drbdn` is always less than the amount of physical space available on the underlying primary partition.

- The `syncer/rate` key controls the rate, in bytes per second, at which DRBD synchronizes disks. Set the rate to 30% of the available replication bandwidth, which is the slowest of either the I/O subsystem or the network interface. The following example assumes 100MB/s available for total replication bandwidth (0.30 * 100MB/s = 30MB/s).

Configuring DRBD

This procedure configures DRBD for deployments with either one or two NICs in each node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On both nodes, identify the primary partitions to use.

```
lsblk --output=NAME,SIZE
```

Record the paths of the primary partitions in the following table. The information is needed in subsequent steps and procedures.

Node	*Isvcs-Partition*	*Metadata-Partition*	*App-Data-Partition*

4. On both nodes, edit the DRBD configuration file.
   a. Open `/etc/drbd.d/global_common.conf` with a text editor.
   b. Add the following values to the `global` and `common/net` sections of the file.

```
global {
  usage-count yes;
}
common {
  net {
    protocol C;
  }
}
```

   c. Save the file, and then close the editor.
5. On both nodes, create a resource definition for Control Center.
   a. Open `/etc/drbd.d/serviced-dfs.res` with a text editor.
   b. **For a dual-NIC system**, add the following content to the file.

   Replace the variables in the content with the actual values for your environment:

```
resource serviced-dfs {
   volume 0 {
      device /dev/drbd0;
      disk Isvcs-Partition;
      meta-disk internal;
   }
   volume 1 {
      device /dev/drbd1;
      disk Metadata-Partition;
      meta-disk internal;
   }
   volume 2 {
      device /dev/drbd2;
      disk App-Data-Partition;
      meta-disk internal;
```

```
    }
    syncer   {
       rate 30M;
    }
    net      {
       after-sb-0pri discard-zero-changes;
       after-sb-1pri discard-secondary;
    }
    on Primary-Public-IP {
       address Primary-Private-IP:7789;
    }
    on Secondary-Public-IP {
       address Secondary-Private-IP:7789;
    }
}
```

c **For a single-NIC system**, add the following content to the file.

Replace the variables in the content with the actual values for your environment:

```
resource serviced-dfs {
    volume 0 {
       device /dev/drbd0;
       disk Isvcs-Partition;
       meta-disk internal;
    }
    volume 1 {
       device /dev/drbd1;
       disk Metadata-Partition;
       meta-disk internal;
    }
    volume 2 {
       device /dev/drbd2;
       disk App-Data-Partition;
       meta-disk internal;
    }
    syncer   {
       rate 30M;
    }
    net      {
       after-sb-0pri discard-zero-changes;
       after-sb-1pri discard-secondary;
    }
    on Primary-Public-IP {
       address Primary-Public-IP:7789;
    }
    on Secondary-Public-IP {
       address Secondary-Public-IP:7789;
    }
}
```

d Save the file, and then close the editor.

6 On both nodes, create device metadata and enable the new DRBD resource.

```
drbdadm create-md all && drbdadm up all
```

Initializing DRBD

Perform this procedure to initialize DRBD and the mirrored storage areas.

Note Unlike the preceding procedures, most of the steps in this procedure are performed on the primary node only.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Synchronize the storage areas of both nodes.
   a. Start the synchronization.

      ```
      drbdadm primary --force serviced-dfs
      ```

      The command may return right away, while the synchronization process continues running in the background. Depending on the sizes of the partitions, this process can take several hours.
   b. Monitor the progress of the synchronization.

      ```
      drbd-overview
      ```

      Do not proceed until the status is `UpToDate/UpToDate`, as in the following example output:

      ```
      0:serviced-dfs/0 Connected Primary/Secondary UpToDate/UpToDate
      1:serviced-dfs/1 Connected Primary/Secondary UpToDate/UpToDate
      2:serviced-dfs/2 Connected Primary/Secondary UpToDate/UpToDate
      ```

      The `Primary/Secondary` values show that the command was run on the primary node; otherwise, the values are `Secondary/Primary`. Likewise, the first value in the `UpToDate/UpToDate` field is the status of the node on which the command is run, and the second value is the status of the remote node.
3. Format the partitions for Control Center internal services data and for Control Center metadata.

   The following commands use the paths of the DRBD devices defined previously, not the paths of the primary partions.

   ```
   mkfs.xfs /dev/drbd0
   mkfs.xfs /dev/drbd1
   ```

   The commands create XFS file systems on the primary node, and DRBD mirrors the file systems to the secondary node.
4. Create a device mapper thin pool for Control Center application data.

   Likewise, this command uses the path of the DRBD device defined previously.
   a. Create a variable for 50% of the space available on the DRDB device.

      The thin pool stores application data and snapshots of the data. You can add storage to the pool at any time.

      Replace *Half-Of-Available-Space* with 50% of the space available on the DRDB device, in gigabytes. Include the symbol for gigabytes (`G`) after the numeric value.

      ```
      myFifty=Half-Of-Available-SpaceG
      ```
   b. Create the thin pool.

      ```
      serviced-storage create-thin-pool -o dm.basesize=$myFifty \
        serviced /dev/drbd2 -v
      ```

   On success, DRBD mirrors the device mapper thin pool to the secondary node.
5. Configure Control Center with the name of the new thin pool.
   a. Open `/etc/default/serviced` in a text editor.
   b. Locate the *SERVICED_FS_TYPE* declaration.
   c. Remove the number sign character (#) from the beginning of the line.

d Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

```
SERVICED_DM_THINPOOLDEV=/dev/mapper/serviced-serviced--pool
```

e Save the file, and then close the editor.

6 Replicate the Control Center configuration on the secondary node.

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

b Open `/etc/default/serviced` in a text editor.

c Locate the *SERVICED_FS_TYPE* declaration.

d Remove the number sign character (#) from the beginning of the line.

e Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

Replace *Thin-Pool-Name* with the name of the thin pool created previously:

```
SERVICED_DM_THINPOOLDEV=Thin-Pool-Name
```

f Save the file, and then close the editor.

7 On the primary node, monitor the progress of the synchronization.

```
drbd-overview
```

Note Do not proceed until synchronization is complete.

8 On both nodes, stop DRBD.

```
drbdadm down all
```

Cluster management software

Pacemaker is an open source cluster resource manager, and Corosync is a cluster infrastructure application for communication and membership services. The Pacemaker/Corosync daemon (`pcs.d`) communicates across nodes in the cluster. When `pcs.d` is installed, started, and configured, the majority of PCS commands can be run on either node in the cluster.

Installing and configuring the cluster management software

Perform this procedure to install and configure the cluster management software.

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

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

3 On both nodes, install the cluster management software.

```
yum install -y corosync pacemaker pcs
```

4 On both nodes, install the Pacemaker resource agent for Control Center.

Pacemaker uses resource agents (scripts) to implement a standardized interface for managing arbitrary resources in a cluster. Zenoss provides a Pacemaker resource agent to manage the Control Center master host in a high-availability cluster.

```
yum --enablerepo=zenoss-stable install -y serviced-resource-agents
```

5 On both nodes, start and enable the PCS daemon.

```
systemctl start pcsd.service && systemctl enable pcsd.service
```

6 On both nodes, set the password of the `hacluster` account.

The Pacemaker package creates the `hacluster` user account, which must have the same password on both nodes.

```
passwd hacluster
```

Creating the cluster in standby mode

Perform this procedure to create the high-availability cluster in standby mode.

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

```
pcs cluster auth Primary-Public-Name Secondary-Public-Name
```

When prompted, enter the password of the `hacluster` account.

3 Generate and synchronize an initial (empty) cluster definition.

```
pcs cluster setup --name serviced-ha \
  Primary-Public-Name Secondary-Public-Name
```

4 Start the PCS management agents on both nodes in the cluster.

The cluster definition is empty, so starting the cluster management agents has no side effects.

```
pcs cluster start --all
```

The cluster management agents start, on both nodes.

5 Check the status.

```
pcs cluster status
```

The expected result is `Online`, for both nodes.

6 Put the cluster in standby mode.

Pacemaker begins monitoring and managing the different resources as they are defined, which can cause problems; standby mode prevents the problems.

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

7 Configure cluster services to start when the node starts.

For more information about cluster startup options, refer to the *Pacemaker documentation*.

```
systemctl enable corosync; systemctl enable pacemaker
```

8 Replicate the configuration on the secondary node.

a In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
b Configure cluster services to start when the node starts.

```
systemctl enable corosync; systemctl enable pacemaker
```

Property and resource options

Pacemaker provides options to support cluster configurations from small and simple to and large and complex. The following list identifies the options that support the two-node, active/passive configuration for Control Center.

`resource-stickiness=100`

Keep all resources bound to the same host.

`no-quorum-policy=ignore`

Pacemaker supports the notion of a voting quorum for clusters of three or more nodes. However, with just two nodes, if one fails, it does not make sense to have a quorum of one, therefore we disable quorums.

`stonith-enabled=false`

Fence or isolate a failed node. (The string "stonith" is an acronym for "shoot the other node in the head".)

Set this option to `false` only during the intial setup and testing period. For production use, set it to `true`. For more information about fencing, refer to the *Zenoss Resource Manager Planning Guide*.

Setting resource and property defaults

Perform this procedure to set resource and property defaults for the high-availability cluster.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Set resource and property defaults.

```
pcs resource defaults resource-stickiness=100
pcs property set no-quorum-policy=ignore
pcs property set stonith-enabled=false
```

3. Check resource defaults.

```
pcs resource defaults
```

Example result:

```
resource-stickiness: 100
```

4. Check property defaults.

```
pcs property
```

Example result:

```
Cluster Properties:
cluster-infrastructure: corosync
cluster-name: serviced-ha
dc-version: 1.1.12-a14efad
have-watchdog: false
no-quorum-policy: ignore
stonith-enabled: false
```

Defining resources

This procedure defines the following logical resources required for the cluster:

- DRBD Master/Secondary DFS set
- Two mirrored file systems running on top of DRBD:

    - `/opt/serviced/var/isvcs`
    - `/opt/serviced/var/volumes`
- `serviced` logical volume group running on top of DRBD
- Manage `serviced` storage
- The floating virtual IP address of the cluster (*HA-Virtual-IP*), which the management software assigns to the active node
- Docker
- NFS
- Control Center

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. Define a resource for the DRBD device, and a clone of that resource to act as the master.
    a. On the primary node, define a resource for the DRBD device.

    ```
    pcs resource create DFS ocf:linbit:drbd \
      drbd_resource=serviced-dfs \
      op monitor interval=30s role=Master \
      op monitor interval=60s role=Slave
    ```

    b. On the primary node, define a clone of that resource to act as the master.

    ```
    pcs resource master DFSMaster DFS \
      master-max=1 master-node-max=1 \
      clone-max=2 clone-node-max=1 notify=true
    ```

    For a master/slave resource, Pacemaker requires separate monitoring intervals for the different roles. In this case, Pacemaker checks the master every 30 seconds and the slave every 60 seconds.
4. Define the file systems that are mounted on the DRBD devices.
    a. On the primary node, define a resource for Control Center internal services data.

    ```
    pcs resource create serviced-isvcs Filesystem \
      device=/dev/drbd/by-res/serviced-dfs/0 \
      directory=/opt/serviced/var/isvcs fstype=xfs
    ```

    b. On the primary node, define a resource for Control Center metadata.

    ```
    pcs resource create serviced-volumes Filesystem \
      device=/dev/drbd/by-res/serviced-dfs/1 \
      directory=/opt/serviced/var/volumes fstype=xfs
    ```

    In the preceding definitions, `serviced-dfs` is the name of the DRBD resource defined previously, in `/etc/drbd.d/serviced-dfs.res`.
5. On the primary node, define the logical volume for `serviced` that is backed by a DRBD device.

    ```
    pcs resource create serviced-lvm ocf:heartbeat:LVM volgrpname=serviced
    ```

6. On the primary node, define the storage resource for `serviced`, to ensure that the device mapper device is deactivated and unmounted properly.

    ```
    pcs resource create serviced-storage ocf:zenoss:serviced-storage
    ```

7. On the primary node, define the resource that represents the floating virtual IP address of the cluster.

For dual-NIC deployments, the definition includes the `nic` key-value pair, which specifies the name of the network interface that is used for all traffic except the private DRBD traffic between the primary and seconday nodes. For single-NIC deployments, omit `nic` key-value pair.

For dual-NIC deployments, replace *HA-Virtual-IP* with the floating virtual IP address of the cluster, and replace *HA-Virtual-IP-NIC* with the name of the network interface that is bound to *HA-Virtual-IP*:

```
pcs resource create VirtualIP ocf:heartbeat:IPaddr2 \
  ip=HA-Virtual-IP nic=HA-Virtual-IP-NIC \
  cidr_netmask=32 op monitor interval=30s
```

For single-NIC deployments, replace *HA-Virtual-IP* with the floating virtual IP address of the cluster:

```
pcs resource create VirtualIP ocf:heartbeat:IPaddr2 \
  ip=HA-Virtual-IP cidr_netmask=32 op monitor interval=30s
```

8 Define the Docker resource.

a On the primary node, define the resource.

```
pcs resource create docker systemd:docker
```

b On both nodes, ensure that the automatic startup of Docker by `systemd` is disabled.

```
systemctl stop docker && systemctl disable docker
```

9 Define the NFS resource.

Control Center uses NFS to share configuration in a multi-host deployment, and failover will not work properly if NFS is not stopped on the failed node.

a On the primary node, define the resource.

```
pcs resource create nfs systemd:nfs
```

b On the primary node, disable Pacemaker monitoring of NFS health.

During normal operations, Control Center occasionally stops and restarts NFS, which could be misinterpreted by Pacemaker and trigger an unwanted failover.

```
pcs resource op remove nfs monitor interval=60s
pcs resource op add nfs monitor interval=0s
```

c On both nodes, ensure that the automatic startup of NFS by `systemd` is disabled.

```
systemctl stop nfs && systemctl disable nfs
```

10 Define the Control Center resource.

a On the primary node, define the resource.

```
pcs resource create serviced ocf:zenoss:serviced
```

b On both nodes, ensure that the automatic startup of `serviced` by `systemd` is disabled.

```
systemctl stop serviced && systemctl disable serviced
```

Pacemaker uses the default timeouts defined by the Pacemaker resource agent for Control Center to decide if `serviced` is unable to start or shutdown correctly. Starting with version 0.0.5 of the Pacemaker resource agent for Control Center, the default values for the start and stop timeouts are 360 and 130 seconds respectively.

The default startup and shutdown timeouts are based on the worst case scenario. In practice, Control Center typically starts and stops in much less time. However, this does not mean that you should decrease these timeouts. There are potential edge cases, especially for startup, where Control Center may take longer than usual to start or stop. If the start/stop timeouts for Pacemaker are set too low, and Control Center encounters one of those edge cases, then Pacemaker takes unnecessary or incorrect actions. For example, if the startup timeout is artificially set too low, 2.5 minutes for example, and Control Center startup encounters an unusual case where it requires at least 3 minutes to start, then Pacemaker initiates failover prematurely.

Defining the Control Center resource group

The resources in a resource group are started in the order they appear in the group, and stopped in the reverse order they appear in the group. The start order is:

1. Mount the file systems (`serviced-isvcs` and `serviced-volumes`)
2. Start the `serviced` logical volume.
3. Manage `serviced` storage.
4. Enable the virtual IP address of the cluster.
5. Start Docker.
6. Start NFS.
7. Start Control Center.

In the event of a failover, Pacemaker stops the resources on the failed node in the reverse order they are defined before starting the resource group on the standby node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Create the Control Center resource group.

```
pcs resource group add serviced-group \
  serviced-isvcs serviced-volumes \
  serviced-lvm serviced-storage \
  VirtualIP docker nfs \
  serviced
```

3. Define constraints for the Control Center resource group.

   Pacemaker resource constraints control when and where resources are deployed in a cluster.

   a. Ensure that `serviced-group` runs on the same node as `DFSMaster`.

   ```
   pcs constraint colocation add serviced-group with DFSMaster \
     INFINITY with-rsc-role=Master
   ```

   b. Ensure that `serviced-group` is only started after `DFSMaster` is started.

   ```
   pcs constraint order promote DFSMaster then \
     start serviced-group
   ```

Verification procedures

The cluster is created in standby mode while various configurations are created. Perform the procedures in the following sections to review the configurations and make adjustments as necessary.

Verifying the DRBD configuration

This procedure reviews the DRBD configuration.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On the primary node, display the full DRBD configuration.

```
drbdadm dump
```

The result should be consistent with the configuration created previously. For more information, see *DRBD configuration assumptions* on page 79.

4. On the primary node, display the synchronization status of mirrored storage areas.

```
drbd-overview
```

Do not proceed until the synchronization is complete. The process is complete when the status of the devices is `UpToDate/UpToDate`.

5. On both nodes, stop DRBD.

```
drbdadm down all
```

Verifying the Pacemaker configuration

This procedure reviews the resource and property defaults for Pacemaker.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Check resource defaverify-pacemaker.ditaults.

```
pcs resource defaults
```

Example result:

```
resource-stickiness: 100
```

3. Check property defaults.

```
pcs property
```

Example result:

```
Cluster Properties:
 cluster-infrastructure: corosync
 cluster-name: serviced-ha
 dc-version: 1.1.12-a14efad
 have-watchdog: false
 no-quorum-policy: ignore
 stonith-enabled: false
```

Note Set the `stonith-enabled` option to `false` only during the intial setup and testing period. For production use, set it to `true`. For more information about fencing, refer to the *Zenoss Resource Manager Planning Guide*.

4. Review the resource constraints.

The ordering constraint should show that `serviced-group` starts after `DFSMaster` (the DRBD master). The colocation constraint should show that `serviced-group` resource and `DFSMaster` are on the same active cluster node.

```
pcs constraint
```

Example result:

```
Location Constraints:
Ordering Constraints:
  promote DFSMaster then start serviced-group (kind:Mandatory)
Colocation Constraints:
  serviced-group with DFSMaster (score:INFINITY) (with-rsc-
role:Master)
```

5 Review the ordering of the `serviced-group` resource group.

```
pcs resource show --full
```

The resources in a resource group are started in the order they appear in the group, and stopped in the reverse order they appear in the group. The correct start order is:

1 serviced-isvcs
2 serviced-volumes
3 serviced-lvm
4 serviced-storage
5 VirtualIP
6 Docker
7 nfs
8 serviced

Verifying the Control Center configuration

This procedure verifies that the Control Center configuration is identical on both nodes.

1 Log in to the primary node as `root`, or as a user with superuser privileges.
2 In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3 On both nodes, compute the checksum of the Control Center configuration file.

```
cksum /etc/default/serviced
```

- If the result is identical on both nodes, the configurations are identical. Do not perform the next step.
- If the result is not identical on both nodes, there may be a difference in their configurations; proceed to the next step.

4 Optional: On both nodes, display the customized variables, if necessary.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | sort
```

Example result:

```
SERVICED_AGENT=1
SERVICED_DM_THINPOOLDEV=/dev/mapper/serviced-serviced--pool
SERVICED_DOCKER_REGISTRY=HA-Virtual-IP:5000
SERVICED_ENDPOINT=HA-Virtual-IP:4979
SERVICED_FS_TYPE=devicemapper
```

```
SERVICED_LOG_ADDRESS=HA-Virtual-IP:5042
SERVICED_LOGSTASH_ES=HA-Virtual-IP:9100
SERVICED_MASTER=1
SERVICED_OUTBOUND_IP=HA-Virtual-IP
SERVICED_STATS_PORT=HA-Virtual-IP:8443
SERVICED_ZK=HA-Virtual-IP:2181
```

Note There may only be insignificant differences between the files, such as an extra space at the beginning of a variable definition.

Verifying cluster startup

This procedure verifies the initial configuration by attempting to start the resources on one node only. With the other node in standby mode, Pacemaker does not automatically fail over to the other node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On the primary node, determine which node is the primary DRBD node.

```
pcs status
```

Example result:

```
Cluster name: serviced-ha
Last updated: Mon Feb 22 11:37:58 2016  Last change: Mon Feb 22
 11:35:19 2016 by root via crm_attribute on Secondary-Public-Name
Stack: corosync
Current DC: Primary-Public-Name (version 1.1.13-a14efad) - partition
 with quorum
2 nodes and 10 resources configured

Node Primary-Public-Name: standby
Node Secondary-Public-Name: standby

Full list of resources:

 Master/Slave Set: DFSMaster [DFS]
 Stopped: [ Primary-Public-Name Secondary-Public-Name ]
 Resource Group: serviced-group
     serviced-isvcs  (ocf::heartbeat:Filesystem):  Stopped
     serviced-volumes  (ocf::heartbeat:Filesystem):  Stopped
     serviced-lvm  (ocf::heartbeat:LVM):  Stopped
     serviced-storage  (ocf::zenoss:serviced-storage):  Stopped
     VirtualIP  (ocf::heartbeat:IPaddr2):  Stopped
     docker  (systemd:docker):  Stopped
     nfs  (systemd:nfs):  Stopped
     serviced  (ocf::zenoss:serviced):  Stopped

PCSD Status:
  Primary-Public-Name: Online
  Secondary-Public-Name: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

The line that begins with `Current DC` identifies the primary node. Review all of the command output for errors.

4 Start DRBD.

a On the secondary node, enter the following command:

```
drbdadm up all
```

b On the primary node, enter the following commands:

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

5 Start cluster resources.

You can run `pcs` commands on either node.

```
pcs cluster unstandby Primary-Public-Name
```

6 Monitor the status of cluster resources.

```
watch pcs status
```

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

Verifying cluster failover

This procedure simulates a failover.

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

2 Enable the DRBD secondary node.

a Take the secondary node out of standby mode.

Replace *Secondary-Public-Name* with the public hostname of the secondary node:

```
pcs cluster unstandby Secondary-Public-Name
```

b Monitor the status of the secondary node.

```
pcs status
```

Do not continue until the status of the secondary node is `Online`.

3 Verify that DRBD has completely synchonized all three volumes on the secondary node.

```
drbd-overview
```

Example result:

```
0:serviced-dfs/0  Connected Primary/Secondary UpToDate/UpToDate
1:serviced-dfs/1  Connected Primary/Secondary UpToDate/UpToDate
2:serviced-dfs/2  Connected Primary/Secondary UpToDate/UpToDate
```

4 Force a failover.

Pacemaker initiates a failover when the primary node is put in standby mode.

Replace *Primary-Public-Name* with the public hostname of the primary node:

```
pcs cluster standby Primary-Public-Name
```

5 Monitor the cluster status.

```
pcs status
```

Repeat the preceding command until all resources report a status of `Started`. Resolve any issues before continuing.

6 Restore the cluster.

Replace *Primary-Public-Name* with the public hostname of the primary node:

```
pcs cluster unstandby Primary-Public-Name
```

Creating new resource pools

This procedure creates a new resource pool for the Control Center master nodes, and one or more resource pools for other hosts.

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

2 Create a new resource pool named `master`.

```
serviced pool add master
```

3 Optional: Create additional resource pools, if desired.

No additional resource pools are required. However, many users find it useful to have pool names such as `infrastructure` and `collector-n` for groups of resource pool hosts.

Replace *Pool-Name* with the name of the pool to create:

```
serviced pool add Pool-Name
```

Repeat the preceding command as desired.

Adding master nodes to their resource pool

This procedure adds the Control Center master nodes to their resource pool, named `master`. The master nodes are added to the resource pool with their public hostnames, so that you can easily see which node is active when you log in to the Control Center browser interface.

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

2 Display the public hostname of the current node.

```
uname -n
```

The result is either *Primary-Public-Name* or *Secondary-Public-Name*.

3 Add the current node to the `master` resource pool.

Replace *Node-Hostname* with the public hostname of the current node:

```
serviced host add Node-Hostname:4979 master
```

4 Force a failover.

Replace *Node-Hostname* with the public hostname of the current node:

```
pcs cluster standby Node-Hostname
```

5 Monitor the cluster status.

```
watch pcs status
```

Do not proceed until all resources report a status of `Started`.

6 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

7 Display the public hostname of the current node.

```
uname -n
```

8 Add the current node to the `master` resource pool.

Replace *Node-Hostname* with the public hostname of the current node:

```
serviced host add Node-Hostname:4979 master
```

9 Restore the cluster.

Replace *Standby-Node-Hostname* with the public hostname of the node that is in standby mode:

```
pcs cluster unstandby Standby-Node-Hostname
```

Control Center on resource pool hosts

Control Center resource pool hosts run the application services scheduled for the resource pool to which they belong, and for which they have sufficent RAM and CPU resources. In a high-availability deployment, a resource pool host may belong to any resource pool other than `master`, and no application services are run in the `master` pool.

Resource Manager has two broad categories of application services: Infrastructure and collection. The services associated with each category can run in the same resource pool, or can run in separate resource pools.

For improved reliability, two resource pool hosts are configured as nodes in an *Apache ZooKeeper* ensemble. The storage required for ensemble hosts is slightly different than the storage required for all other resource pool hosts: Each ensemble host requires a separate primary partition for Control Center internal services data, in addition to the primary partition for Docker data. Unless the ZooKeeper service on the Control Center master host fails, their roles in the ZooKeeper ensemble do not affect their roles as Control Center resource pool hosts.

Note The hosts for the ZooKeeper ensemble require static IP addresses, because ZooKeeper does not support hostnames in its configurations.

Repeat the procedures in the following sections for each host you wish to add to your Control Center deployment.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the candidate host as `root`, or as a user with superuser privileges.
2. Verify that the host implements the 64-bit version of the x86 instruction set.

   ```
   uname -m
   ```

   - If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
   - If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.
3. Verify that name resolution works on this host.

   ```
   hostname -i
   ```

   If the result is not a valid IPv4 adddress, add an entry for the host to the network nameserver, or to `/etc/hosts`.
4. Verify that the host's numeric identifier is unique.

   Each host in a Control Center cluster must have a unique host identifier.

   ```
   hostid
   ```
5. Determine whether the available, unused storage is sufficient.
   a. Display the available storage devices.

      ```
      lsblk --output=NAME,SIZE
      ```

   b. Compare the available storage with the amount required for a resource pool host in your deployment.

      In particular, resource pool hosts that are configured as nodes in a ZooKeeper ensemble require an additional primary partition for Control Center internal services data.

      For more information, refer to the *Zenoss Resource Manager Planning Guide*.
6. Determine whether the available memory and swap is sufficient.
   a. Display the available memory.

      ```
      free -h
      ```

   b. Compare the available memory with the amount required for a resource pool host in your deployment.

      For more information, refer to the *Zenoss Resource Manager Planning Guide*.
7. Update the operating system, if necessary.
   a. Determine which release is installed.

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

      If the result includes `7.0`, perform the following substeps.
   b. Update the operating system.

      ```
      yum update -y
      ```

   c. Restart the system.

      ```
      reboot
      ```

Preparing a resource pool host

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the candidate resource pool host as `root`, or as a user with superuser privileges.
2. Add an entry to `/etc/hosts` for localhost, if necessary.
   a. Determine whether `127.0.0.1` is mapped to `localhost`.

   ```
   grep 127.0.0.1 /etc/hosts | grep localhost
   ```

   If the preceding commands return no result, perform the following substep.
   b. Add an entry to `/etc/hosts` for `localhost`.

   ```
   echo "127.0.0.1   localhost" >> /etc/hosts
   ```

3. Disable the firewall, if necessary.

   This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.
   a. Determine whether the `firewalld` service is enabled.

   ```
   systemctl status firewalld.service
   ```

   - If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
   - If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

   b. Disable the `firewalld` service.

   ```
   systemctl stop firewalld && systemctl disable firewalld
   ```

   On success, the preceding commands display messages similar to the following example:

   ```
   rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
   rm '/etc/systemd/system/basic.target.wants/firewalld.service'
   ```

4. Optional: Enable persistent storage for log files, if desired.

   By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

   ```
   mkdir -p /var/log/journal && systemctl restart systemd-journald
   ```

5. Disable Security-Enhanced Linux (SELinux), if installed.
   a. Determine whether SELinux is installed.

   ```
   test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
   ```

   If the preceding commands return a result, SELinux is installed.
   b. Set the operating mode to `disabled`.
   Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.
   c. Confirm the new setting.

   ```
   grep '^SELINUX=' /etc/selinux/config
   ```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations. **Perform this step only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**.

```
yum install -y nmap-ncat
```

8 Install and configure the NTP package.

a Install the package.

```
yum install -y ntp
```

b Set the system time.

```
ntpd -gq
```

c Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

d Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot. The following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

9 Reboot the host.

```
reboot
```

Creating a file system for Control Center internal services

This procedure creates an XFS file system on a primary partition.

Note **Perform this procedure only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**. No other resource pool hosts run Control Center internal services, so no other pool hosts need a partition for internal services data.

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

2 Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

3 Create an XFS file system.

Replace *Isvcs-Partition* with the path of the target primary partition:

```
mkfs -t xfs Isvcs-Partition
```

4 Create the mount point for Control Center internal services data.

```
mkdir -p /opt/serviced/var/isvcs
```

5 Add an entry to the `/etc/fstab` file.
Replace *Isvcs-Partition* with the path of the primary partition used in the previous step:

```
echo "Isvcs-Partition \
  /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep isvcs
```

Example result:

```
/dev/xvdb1 on /opt/serviced/var/isvcs type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

Perform this procedure on each resource pool host in your deployment.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Install Docker 1.9.0, and then disable accidental upgrades.
   a Add the Docker repository to the host's repository list.

```
cat > /etc/yum.repos.d/docker.repo <<-EOF
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

   b Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install -y docker-engine-1.9.0
```

   c Open `/etc/yum.repos.d/docker.repo` with a text editor.
   d Change the value of the `enabled` key from `1` to `0`.
   e Save the file and close the text editor.
3 Create a symbolic link for the Docker temporary directory.

   Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

   a Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

b Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4 Create a `systemd` override file for the Docker service definition.

a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-stable install -y serviced-1.1.7
```

6 Create a device mapper thin pool for Docker data.

a Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

b Create the thin pool.

Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

7 Configure and start the Docker service.

a Create variables for adding arguments to the Docker configuration file.

The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

8 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (**"**) is at the end of the line.

---

d Restart the Docker service.

```
systemctl restart docker
```

Configuring and starting Control Center

This procedure customizes key configuration variables of Control Center.

Perform this procedure on each resource pool host in your deployment.

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

2 Configure Control Center as an agent of the master host.

The following variable configures `serviced` to serve as agent:

`SERVICED_AGENT`

Default: `0` (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: `0` (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

In addition, replace `{{SERVICED_MASTER_IP}}` with *HA-Virtual-IP*, the virtual IP address of the high-availability cluster, in the following lines::

```
# SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
# SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
# SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
# SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
# SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
# SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
```

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
The following example shows the line to change:

```
# SERVICED_AGENT=0
```

c Remove the number sign character (#) from the beginning of the line.

d Find the *SERVICED_MASTER* declaration, and then remove the number sign character (#) from the beginning of the line.

e Globally replace `{{SERVICED_MASTER_IP}}` with the virtual IP address of the high-availability cluster (*HA-Virtual-IP*).

Note Remove the number sign character (#) from the beginning of each variable declaration that includes the virtual IP address.

---

f Save the file, and then close the editor.

3 Optional: Specify an alternate private network for Control Center, if necessary.

Control Center requires a 16-bit, private IPv4 network for virtual IP addresses, independent of the private network used in a dual-NIC DRBD configuration. The default network is 10.3/16. If the default network is already in use in your environment, you may select any valid IPv4 16-bit network.

The following variable configures `serviced` to use an alternate network:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value.
The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

4 Start the Control Center service (`serviced`).

```
systemctl start serviced
```

To monitor progress, open a separate window to the host, and then enter the following command:

```
journalctl -flu serviced -o cat
```

Deploying Resource Manager

This procedure adds all of the resource pool hosts to the Control Center cluster, and then deploys the Resource Manager application.

1. Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.
2. Display the public hostname of the current node.

   ```
   uname -n
   ```

   The result is either *Primary-Public-Name* or *Secondary-Public-Name*.
3. Place the other node in standby mode.

   This avoids potential conflicts and errors in the event of an unexpected `serviced` shutdown during the initial deployment.

   Replace *Other-Node-Hostname* with the public hostname of the other node:

   ```
   pcs cluster standby Other-Node-Hostname
   ```
4. Add resource pool hosts to resource pools.

   Replace *Hostname-Or-IP* with the hostname or IP address of the resource pool host to add, and replace *Resource-Pool-Name* with the name of a resource pool created previously, or with `default`:

   ```
   serviced host add Hostname-Or-IP:4979 Resource-Pool-Name
   ```

   If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

   Repeat this step for each resource pool host in your deployment.
5. Add the `Zenoss.resmgr` application to Control Center.

   ```
   myPath=/opt/serviced/templates
   serviced template add $myPath/zenoss-resmgr-*.json
   ```

   On success, the `serviced` command returns the template ID.
6. Deploy the application.

   Replace *Template-ID* with the template identifier returned in the previous step, and replace *Deployment-ID* with a name for this deployment (for example, `Dev` or `Test`):

   ```
   serviced template deploy Template-ID default Deployment-ID
   ```

   Control Center pulls Resource Manager images into the local registry. To monitor progress, open a separate window, and enter the following command:

   ```
   journalctl -flu serviced -o cat
   ```
7. Restore the cluster.

   Replace *Standby-Node-Hostname* with the public hostname of the node that is in standby mode:

   ```
   pcs cluster unstandby Standby-Node-Hostname
   ```

ZooKeeper ensemble configuration

Control Center relies on *Apache ZooKeeper* to coordinate its services. The configuration steps in this section create a ZooKeeper ensemble of 3 nodes.

A ZooKeeper ensemble requires a minimum of 3 nodes, and 3 nodes is sufficient for most deployments. A 5-node configuration improves failover protection during maintenance windows. Ensembles larger than 5 nodes are not necessary. An odd number of nodes is recommended, and an even number of nodes is strongly discouraged.

Control Center variables for ZooKeeper

This tables in this section associates the ZooKeeper-related Control Center variables to set in `/etc/default/serviced` with the roles that hosts play in a Control Center cluster.

Table 7: Control Center master nodes

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 1

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 8: Control Center resource pool host and ZooKeeper ensemble node

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 2 or 3

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ISVCS_START

The list of Control Center internal services to start and run on hosts other than the master host.
Value: `zookeeper`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.

Value: *IP-Address*:2181,...

Table 9: Control Center resource pool host

SERVICED_ZK The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (,). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it. Value: *IP-Address*:2181,...

Configuring a master node as a ZooKeeper node

This procedure configures both Control Center master nodes as members of the ZooKeeper ensemble.

Note For accuracy, this procedure constructs Control Center configuration variables in the shell and appends them to /etc/default/serviced. The last step is to move the variables from the end of the file to more appropriate locations.

1. Log in to the primary node as root, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as root, or as a user with superuser privileges.
3. On both nodes, create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

   Note Define the variables identically on both the primary and the secondary nodes, and on each resource pool host.

   Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

   ```
   node1=HA-Virtual-IP
   node2=Pool-Host-A-IP
   node3=Pool-Host-B-IP
   ```

   Note ZooKeeper requires IP addresses for ensemble configuration.

4. On both nodes, set the ZooKeeper node ID to 1.

   ```
   echo "SERVICED_ISVCS_ZOOKEEPER_ID=1" >> /etc/default/serviced
   ```

5. On both nodes, specify the nodes in the ZooKeeper ensemble.
   You may copy the following text and paste it in your console:

   ```
   echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
     >> /etc/default/serviced
   ```

6. On both nodes, specify the nodes in the ZooKeeper quorum.

   ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with 0.0.0.0.

   You may copy the following of text and paste it in your console:

   ```
   q1="1@0.0.0.0:2888:3888"
   q2="2@${node2}:2888:3888"
   ```

```
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

7 On both nodes, clean up the Control Center configuration file.
   a Open `/etc/default/serviced` in a text editor.
   b Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

      The value of this declaration specifies 3 hosts.
   c Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

      The value of this declaration is just the master node.
   d Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.
   e Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.
   f Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   g Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.
   h Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.
   i Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   j Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.
   k Save the file, and then close the editor.
8 On both hosts, verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | egrep '(_ZOO|_ZK)'
```

Configuring a resource pool host as a ZooKeeper node

To perform this procedure, you need a resource pool host with an XFS file system on a separate partition.

This procedure configures a ZooKeeper ensemble on a resource pool host. Repeat this procedure on each Control Center resource pool host to add to the ZooKeeper ensemble.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=HA-Virtual-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

3 Set the ID of this node in the ZooKeeper ensemble.

For *Pool-Host-A-IP* (**node2**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=2" >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=3" >> /etc/default/serviced
```

4 Specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

5 Specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

For *Pool-Host-A-IP* (**node2**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@0.0.0.0:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@0.0.0.0:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Set the *SERVICED_ISVCS_START* variable, and clean up the Control Center configuration file.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_ISVCS_START* variable, and then delete all but `zookeeper` from its list of values.

c Remove the number sign character (#) from the beginning of the line.

d Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

The value of this declaration specifies 3 hosts.

e Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

The value of this declaration is just the master node.

f Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.

g Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.

h Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

i Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.

j Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.

k Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

l Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.

m Save the file, and then close the editor.

7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced \
  | egrep '(_ZOO|_ZK|_STA)'
```

8 Pull the required Control Center ZooKeeper image from the master host.

a Identify the image to pull.

```
serviced version | grep IsvcsImages
```

Example result:

```
IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
```

b Pull the Control Center ZooKeeper image.

Replace *Isvcs-ZK-Image* with the name and version number of the ZooKeeper image from the previous substep:

```
docker pull Isvcs-ZK-Image
```

Starting a ZooKeeper ensemble

This procedure starts a ZooKeeper ensemble.

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

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

2 Display the public hostname of the current node.

```
uname -n
```

The result is either *Primary-Public-Name* or *Secondary-Public-Name*.

3 Place the other node in standby mode.

This avoids potential conflicts and errors in the event of an unexpected `serviced` shutdown during the ZooKeeper startup.

Replace *Other-Node-Hostname* with the public hostname of the other node:

```
pcs cluster standby Other-Node-Hostname
```

4 In a separate window, log in to the second node of the ZooKeeper ensemble (*Pool-Host-A-IP*).

5 In another separate window, log in to the third node of the ZooKeeper ensemble (*Pool-Host-B-IP*).

6 On all ensemble hosts, stop and start `serviced`.

```
systemctl stop serviced && systemctl start serviced
```

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

```
{ echo stats; sleep 1; } | nc localhost 2181 | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-A-IP 2181  | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-B-IP 2181  | grep Mode
```

If `nc` is not available, you can use `telnet` with *interactive ZooKeeper commands*.

8 Restore the cluster.

Replace *Other-Node-Hostname* with the public hostname of the primary node:

```
pcs cluster unstandby Other-Node-Hostname
```

Updating resource pool hosts

The default configuration of resource pool hosts sets the value of the *SERVICED_ZK* variable to the master host only. This procedure updates the setting to include the full ZooKeeper ensemble.

Perform this procedure on each resource pool host in your Control Center cluster.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Update the variable.
   a Open `/etc/default/serviced` in a text editor.
   b Locate the *SERVICED_ZK* declaration, and then replace its value with the same value used in the ZooKeeper ensemble nodes.
   c Save the file, and then close the editor.
3 Restart Control Center.

```
systemctl restart serviced
```

Creating a high-availability deployment without internet access 2

The procedures in this chapter create a high-availability deployment of Control Center and Resource Manager on Red Hat Enterprise Linux (RHEL) 7.1 or 7.2 hosts, or on CentOS 7.1 or 7.2 hosts. To use the procedures in this chapter, you must have a minimum of four hosts. None of the hosts require internet access.

For more information about deploying Control Center and Resource Manager, refer to the *Zenoss Resource Manager Planning Guide*.

Note For optimal results, review this chapter thoroughly before starting the installation process.

Master host storage requirements

In addition to the storage required for its operating system, both Control Center master hosts in the failover cluster require the following storage areas:

- A local primary partition for Docker data, configured as a device mapper thin pool.
- A local primary partition for Control Center internal services data, formatted with the XFS file system.

  Note Control Center internal services include ZooKeeper, which requires consistently fast storage. Zenoss recommends using a separate, high-performance storage resource for Control Center internal services. For example, a drive that is configured with only one primary partition, which eliminates contention by other services.

- A local primary partition for Control Center metadata, formatted with the XFS file system.
- A local primary partition for Resource Manager data, configured as a device mapper thin pool.

Note This chapter includes procedures for configuring and formatting all required storage areas.

In addition, the primary node of the failover cluster requires a local primary partition, a remote primary partition, or a remote file server, for backups of Resource Manager data. The local or remote primary partition is formatted with the XFS file system. A remote file server must provide a file system that is compatible with XFS.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

For storage sizing information, refer to the *Zenoss Resource Manager Planning Guide*.

Key variables used in this chapter

The following tables associate important features of a high-availability deployment with the variables used in this chapter.

Feature	Variable Name	
	Primary Node	**Secondary Node**
Public IP address of master node (static; known to all machines in the Control Center cluster)	*Primary-Public-IP*	*Secondary-Public-IP*
Public hostname of master node (returned by `uname -n`; resolves to the public IP address)	*Primary-Public-Name*	*Secondary-Public-Name*
Private IP address of master node (static; dual-NIC systems only)	*Primary-Private-IP*	*Secondary-Private-IP*
Private hostname of master node (resolves to the private IP address; dual-NIC systems only)	*Primary-Private-Name*	*Secondary-Private-Name*

Feature	Variable Name
Virtual IP address of the high-availability cluster (static; known enterprise-wide)	*HA-Virtual-IP*
Virtual hostname of the high-availability cluster (known enterprise-wide)	*HA-Virtual-Name*
Public IP address of resource pool host A (static; for ZooKeeper ensemble)	*Pool-Host-A-IP*
Public IP address of resource pool host B (static; for ZooKeeper ensemble)	*Pool-Host-B-IP*
Primary partition for Docker data (not mirrored)	*Docker-Partition*
Primary partition for Control Center internal services data (mirrored)	*Isvcs-Partition*
Primary partition for Control Center metadata (mirrored)	*Metadata-Partition*
Primary partition for Control Center application data (mirrored)	*App-Data-Partition*
Primary partition for Control Center backups (not mirrored)	*Backups-Partition*

Downloading files for offline installation

This procedure describes how to download RPM packages and Docker image files to your workstation.

To perform this procedure, you need:

- A workstation with internet access.
- A portable storage medium, such as a USB flash drive, with at least 5 GB of free space.
- Permission to download the required files from the *File Portal - Download Zenoss Enterprise Software* site. You may request permission by filing a ticket at the *Zenoss Support* site.

1. In a web browser, navigate to the *File Portal - Download Zenoss Enterprise Software* site.
2. Log in with the account provided by Zenoss Support.
3. Download archive files to your workstation.

   Replace *Version* with the most recent version number available on the download page:

   - `install-zenoss-hbase:v`*`Version`*`.run`
   - `install-zenoss-isvcs-zookeeper:v`*`Version`*`.run`
   - `install-zenoss-opentsdb:v`*`Version`*`.run`

- `install-zenoss-resmgr_5.1:5.1`*`Version`*`.run`
- `install-zenoss-serviced-isvcs:v`*`Version`*`.run`
- `serviced-resource-agents-`*`Version`*`.x86_64.rpm`

4 Download the RHEL/CentOS mirror package for your upgrade.

Note If you are planning to upgrade the operating system during your Control Center and Resource Manager upgrade, choose the mirror package that matches the RHEL/CentOS release to which you are upgrading, not the release that is installed now.

Replace *Version* with the most recent version number available on the download page, and replace *Release* with the version of RHEL/CentOS appropriate for your environment:

```
yum-mirror-centos7.Release-Version.x86_64.rpm
```

5 Copy the files to your portable storage medium.

Control Center on the master nodes

A high-availability deployment features two Control Center master nodes that are configured for failover. One host is the primary node, and the other host is the secondary node. Their configurations differ somewhat, but are mostly the same.

Note Both master nodes require the following non-standard packages:

- For DRBD: `drbd84-utils` and `kmod-drbd84`.
- For Pacemaker/Corosync: `corosync`, `pacemaker` and `pcs`.

The Control Center and Resource Manager offline artifacts do not include the preceding packages.

Perform all of the procedures in this section on the primary node and on the secondary node.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center master host.

Perform this procedure on the primary node and on the secondary node.

1 Log in to the candidate host as `root`, or as a user with superuser privileges.
2 Verify that the host implements the 64-bit version of the x86 instruction set.

```
uname -m
```

- If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
- If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.

3 Verify that the host's numeric identifier is unique.

Each host in a Control Center cluster must have a unique host identifier.

```
hostid
```

4 Determine whether the available, unused storage is sufficient.

a. Display the available storage devices.

```
lsblk --output=NAME,SIZE
```

b. Compare the available storage with the amount required for a Control Center master host.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

5. Determine whether the available memory and swap is sufficient.

a. Display the available memory.

```
free -h
```

b. Compare the available memory with the amount required for a master host in your deployment.

For more information, refer to the *Zenoss Resource Manager Planning Guide*.

6. Verify the operating system release.

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

If the result includes `7.0`, select another host or upgrade the operating system.

7. Determine whether required packages are installed.

```
for pkg in drbd84-utils kmod-drbd84 corosync pacemaker pcs
do
  echo "Result for $pkg:"
  rpm -qa | grep $pkg
done
```

Install missing packages before continuing.

Staging files for offline installation

Before performing this procedure, complete all of the steps in *Downloading files for offline installation* on page 110. In addition, verify that approximately 4GB of temporary space is available on the file system where `/root` is located.

This procedure adds files for offline installation to the master node. The staged files are required in subsequent procedures.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Copy the archive files from your portable storage medium to `/root`.
3. Set the file permissions of the self-extracting archive files to execute.

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

4. Change directory to `/root`.

```
cd /root
```

5. Install the Resource Manager repository mirror.

```
yum install -y ./yum-mirror-*.x86_64.rpm
```

6. Optional: Delete the package file, if desired.

```
rm ./yum-mirror-*.x86_64.rpm
```

Preparing the master host operating system

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center master host.

Perform this procedure on the primary node and on the secondary node.

1 Log in to the host as `root`, or as a user with superuser privileges.
2 Add an entry to `/etc/hosts` for localhost, if necessary.
   a Determine whether `127.0.0.1` is mapped to `localhost`.

```
grep 127.0.0.1 /etc/hosts | grep localhost
```

   If the preceding commands return no result, perform the following substep.
   b Add an entry to `/etc/hosts` for `localhost`.

```
echo "127.0.0.1   localhost" >> /etc/hosts
```

3 Add the required hostnames and IP addresses of both the primary and the seconday node to the `/etc/hosts` file.

For a dual-NIC system, replace each variable name with the values designated for each node, and replace `example.com` with the domain name of your orgnanization:

```
echo "Primary-Public-IP Primary-Public-Name.example.com \
  Primary-Public-Name" >> /etc/hosts
echo "Primary-Private-IP Primary-Private-Name.example.com \
  Primary-Private-Name" >> /etc/hosts
echo "Secondary-Public-IP Secondary-Public-Name.example.com \
  Secondary-Public-Name" >> /etc/hosts
echo "Secondary-Private-IP Secondary-Private-Name.example.com \
  Secondary-Private-Name" >> /etc/hosts
```

For a single-NIC system, replace each variable name with the values designated for each node, and replace `example.com` with the domain name of your orgnanization:

```
echo "Primary-Public-IP Primary-Public-Name.example.com \
  Primary-Public-Name" >> /etc/hosts
echo "Secondary-Public-IP Secondary-Public-Name.example.com \
  Secondary-Public-Name" >> /etc/hosts
```

4 Disable the firewall, if necessary.

This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.

   a Determine whether the `firewalld` service is enabled.

```
systemctl status firewalld.service
```

   - If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
   - If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

   b Disable the `firewalld` service.

```
systemctl stop firewalld && systemctl disable firewalld
```

On success, the preceding commands display messages similar to the following example:

```
rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
rm '/etc/systemd/system/basic.target.wants/firewalld.service'
```

5 Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

6 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

7 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

8 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations.

```
yum --enablerepo=zenoss-mirror install -y nmap-ncat
```

9 Remove any file system signature from the required primary partitions.

Replace each variable name with the path of the primary partition designated for each storage area:

```
wipefs -a Docker-Partition
wipefs -a Isvcs-Partition
wipefs -a Metadata-Partition
wipefs -a App-Data-Partition
```

10 Add mount points for XFS file systems, which are created in subsequent steps.

```
mkdir -p /opt/serviced/var/isvcs /opt/serviced/var/volumes
```

11 Reboot the host.

```
reboot
```

Configuring an NTP master server

This procedure configures an NTP master server on the master nodes. If you have an NTP time server inside your firewall, you may configure the master nodes to use it; however, this procedure does not include that option.

Perform this procedure on the primary node and on the secondary node.

1. Log in the master node as `root`, or as a user with superuser privileges.
2. Install the NTP package.

```
yum --enablerepo=zenoss-mirror install -y ntp
```

3. Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

4. Edit the NTP configuration file./
   a. Open `/etc/ntp.conf` with a text editor.
   b. Replace all of the lines in the file with the following lines:

```
# Use the local clock
server 127.127.1.0 prefer
fudge  127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift
broadcastdelay 0.008

# Give localhost full access rights
restrict 127.0.0.1

# Grant access to client hosts
restrict Address-Range mask Netmask nomodify notrap
```

   c. Replace *Address-Range* with the range of IPv4 network addresses that are allowed to query this NTP server.

      For example, the following IP addresses are assigned to the hosts in a Control Center cluster:

      ```
      203.0.113.10
      203.0.113.11
      203.0.113.12
      203.0.113.13
      ```

      For the preceding addresses, the value for *Address-Range* is `203.0.113.0`.
   d. Replace *Netmask* with the IPv4 network mask that corresponds with the address range.
      For example, the network mask for `203.0.113.0` is `255.255.255.0`.
   e. Save the file and exit the editor.
5. Enable and start the NTP daemon.
   a. Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

   b. Configure `ntpd` to start when the system starts.

      Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

   c. Start `ntpd`.

```
systemctl start ntpd
```

Configuring a storage area for backups

The Control Center master host requires local or remote storage space for backups of Control Center data. This procedure includes steps to create an XFS file system on a primary partition, if necessary, and steps to mount a file system for backups. For more information about backups, refer to the *Zenoss Resource Manager Planning Guide*.

Note If you are using a primary partition on a local device for backups, ensure that the primary partition for Control Center internal services data is not on the same device.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Optional: Remove any file system signature from the primary partition for Control Center backups, if necessary.

   If you are using a remote file server for backups, skip this step.

   Replace *Backups-Partition* with the path of the primary partition designated for Control Center backups:

   ```
   wipefs -a Backups-Partition
   ```
3. Optional: Create an XFS file system, if necessary.

   Skip this step if you are using a remote file server.

   Replace *Backups-Partition* with the path of the primary partition designated for Control Center backups:

   ```
   mkfs.xfs Backups-Partition
   ```
4. Create an entry in the `/etc/fstab` file.

   Replace *File-System-Specification* with one of the following values:

   - the path of *Backups-Partition*, used in the previous step
   - the remote server specification

   ```
   echo "File-System-Specification \
     /opt/serviced/var/backups xfs defaults 0 0" >> /etc/fstab
   ```
5. Create the mount point for backup data.

   ```
   mkdir -p /opt/serviced/var/backups
   ```
6. Mount the file system, and then verify it mounted correctly.

   ```
   mount -a && mount | grep backups
   ```

   Example result:

   ```
   /dev/sdb3 on /opt/serviced/var/backups type xfs
    (rw,relatime,seclabel,attr2,inode64,noquota)
   ```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

Perform this procedure on the primary node and on the secondary node.

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

2 Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install --enablerepo=zenoss-mirror -y docker-engine
```

3 Create a symbolic link for the Docker temporary directory.

Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

a Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

b Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4 Create a `systemd` override file for the Docker service definition.

a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

```
yum clean all && yum makecache fast
yum --enablerepo=zenoss-mirror install -y serviced
```

6 Disable automatic startup of Control Center by `systemd`.

The cluster management software controls the Docker service.

```
systemctl disable serviced
```

7 Create a device mapper thin pool for Docker data.

Replace *Docker-Partition* with the path of the primary partition designated for Docker data:

```
serviced-storage create-thin-pool docker Docker-Partition
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

8 Configure and start the Docker service.

a Create variables for adding arguments to the Docker configuration file.

The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

9 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

d Restart the Docker service.

```
systemctl restart docker
```

10 Import the Control Center and Resource Manager images into the local `docker` registry.

The images are contained in the self-extracting archive files that are staged in the `/root` directory.

a Change directory to `/root`.

```
cd /root
```

b Extract the images.

```
for image in install-*.run
do
  echo -n "$image: "
  eval ./$image
```

```
done
```

Image extraction begins when you press the **y** key. If you press the **y** key and then **Return** key, the current image is extracted, but the next one is not.

c Optional: Delete the archive files, if desired.

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

11 Stop and disable the Docker service.

The cluster management software controls the Docker service.

```
systemctl stop docker && systemctl disable docker
```

Installing Resource Manager

This procedure installs Resource Manager and configures the NFS server.

Perform this procedure on the primary node and on the secondary node.

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

2 Install Resource Manager.

```
yum install -y zenoss-resmgr-service
```

3 Configure and disable the NFS service.

Currently, *an unresolved issue* prevents the NFS server from starting correctly. The following commands provide a workaround to ensure that it does.

a Open `/lib/systemd/system/nfs-server.service` with a text editor.

b Change `rpcbind.target` to `rpcbind.service` on the following line:

```
Requires= network.target proc-fs-nfsd.mount rpcbind.target
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

d Stop and disable the NFS service.

The cluster management software controls the NFS service.

```
systemctl stop nfs && systemctl disable nfs
```

Configuring Control Center

This procedure customizes key configuration variables of Control Center.

Perform this procedure on the primary node and on the secondary node.

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

2 Configure Control Center to serve as both master and agent, and to use the virtual IP address of the high-availability cluster.

The following variables define the roles `serviced` can assume:

`SERVICED_AGENT`

Default: `0` (false)

Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

`SERVICED_MASTER`

Default: `0` (false)

Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

In addition, replace `{{SERVICED_MASTER_IP}}` with *HA-Virtual-IP*, the virtual IP address of the high-availability cluster, in the following lines:

```
# SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
# SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
# SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
# SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
# SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
# SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
```

a Open `/etc/default/serviced` in a text editor.
b Locate the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
c Remove the number sign character (#) from the beginning of the line.
d Locate the *SERVICED_MASTER* declaration, and then change the value from `0` to `1`.
e Remove the number sign character (#) from the beginning of the line.
f Globally replace `{{SERVICED_MASTER_IP}}` with the virtual IP address of the high-availability cluster.

Note Remove the number sign character (#) from the beginning of each variable declaration that includes the IP address.

---

g Save the file, and then close the editor.

3 Configure Control Center to send its responses to the virtual IP address of the high-availability cluster.
a Open `/etc/default/serviced` in a text editor.
b Locate the *SERVICED_OUTBOUND_IP* declaration, and then change its default value to *HA-Virtual-IP*.

Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster:

```
SERVICED_OUTBOUND_IP=HA-Virtual-IP
```

c Remove the number sign character (#) from the beginning of the line.
d Save the file, and then close the editor.

4 Optional: Specify an alternate private network for Control Center, if necessary.

Control Center requires a 16-bit, private IPv4 network for virtual IP addresses, independent of the private network used in a dual-NIC DRBD configuration. The default network is 10.3/16. If the default network is already in use in your environment, you may select any valid IPv4 16-bit network.

The following variable configures `serviced` to use an alternate network:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

User access control

Control Center provides a browser interface and a command-line interface.

To gain access to the Control Center browser interface, users must have login accounts on the Control Center master host. (Pluggable Authentication Modules (PAM) is supported.) In addition, users must be members of the Control Center administrative group, which by default is the system group, `wheel`. To enhance security, you may change the administrative group from `wheel` to any non-system group.

To use the Control Center command-line interface, users must have login accounts on the Control Center master host, and be members of the `docker` user group. Members of the `wheel` group, including `root`, are members of the `docker` group.

Adding users to the default administrative group

This procedure adds users to the default administrative group of Control Center, `wheel`. Performing this procedure enables users with superuser privileges to gain access to the Control Center browser interface.

Note Perform this procedure or the next procedure, but not both.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Add users to the system group, `wheel`.

   Replace *User* with the name of a login account on the master host.

   ```
   usermod -aG wheel User
   ```

   Repeat the preceding command for each user to add.

   Note For information about using Pluggable Authentication Modules (PAM), refer to your operating system documentation.

Configuring a regular group as the Control Center administrative group

This procedure changes the default administrative group of Control Center from `wheel` to a non-system group.

Note Perform this procedure or the previous procedure, but not both.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Create a variable for the group to designate as the administrative group.

In this example, the name of group to create is `serviced`. You may choose any name or use an existing group.

```
GROUP=serviced
```

3 Create a new group, if necessary.

```
groupadd $GROUP
```

4 Add one or more existing users to the new administrative group.

Replace *User* with the name of a login account on the host:

```
usermod -aG $GROUP User
```

Repeat the preceding command for each user to add.

5 Specify the new administrative group in the `serviced` configuration file.

The following variable specifies the administrative group:

SERVICED_ADMIN_GROUP

Default: `wheel`

The name of the Linux group on the Control Center master host whose members are authorized to use the Control Center browser interface. You may replace the default group with a group that does not have superuser privileges.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ADMIN_GROUP* declaration, and then change the value from `wheel` to the name of the group you chose earlier.
The following example shows the line to change:

```
# SERVICED_ADMIN_GROUP=wheel
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

6 Optional: Prevent `root` users and members of the `wheel` group from gaining access to the Control Center browser interface, if desired.

The following variable controls privileged logins:

SERVICED_ALLOW_ROOT_LOGIN

Default: `1` (true)

Determines whether `root`, or members of the `wheel` group, may gain access to the Control Center browser interface.

a Open `/etc/default/serviced` in a text editor.

b Find the *SERVICED_ALLOW_ROOT_LOGIN* declaration, and then change the value from `1` to `0`.
The following example shows the line to change:

```
# SERVICED_ALLOW_ROOT_LOGIN=1
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

Enabling use of the command-line interface

This procedure enables users to perform administrative tasks with the Control Center command-line interface by adding individual users to the `docker` group.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Add users to the Docker group, `docker`.

   Replace *User* with the name of a login account on the host.

   ```
   usermod -aG docker User
   ```

   Repeat the preceding command for each user to add.

Configuring Logical Volume Manager

Control Center application data is managed by a device mapper thin pool created with Logical Volume Manager (LVM). This procedure adjusts the LVM configuration for mirroring by DRBD.

Perform this procedure on the primary node and on the secondary node.

1. Log in to the host as `root`, or as a user with superuser privileges.
2. Edit the LVM configuration file.
   a. Open `/etc/lvm/lvm.conf` with a text editor.
   b. Exclude the partition for Control Center application data.

      The line to edit is in the `devices` section.

      Replace *App-Data-Partition* with the path of the primary partition designated for Control Center application data.

      ```
      filter = ["r|App-Data-Partition|"]
      ```

   c. Disable caching and the metadata daemon.

      Set the value of the `write_cache_state` and `use_lvmetad` keys to 0.

      ```
      write_cache_state = 0
      use_lvmetad = 0
      ```

   d. Save the file and close the editor.
3. Delete any stale cache entries.

   ```
   rm -f /etc/lvm/cache/.cache
   ```

4. Restart the host.

   ```
   reboot
   ```

DRBD configuration assumptions

The following list identifies the assumptions that inform the DRBD resource definition for Control Center:

- Each node has either one or two NICs. In dual-NIC hosts the private IP/hostnames are reserved for DRBD traffic. This is recommended configuration, which enables real-time writes for disk synchronization between the active and passive nodes, and no contention with application traffic. However, it is possible to use DRBD with a single NIC.
- The default port number for DRBD traffic is 7789.
- All volumes should synchronize and failover together. This is accomplished by creating a single resource definition.

- DRBD stores its metadata on each volume (`meta-disk/internal`), so the total amount of space reported on the logical device `/dev/drbd`*n* is always less than the amount of physical space available on the underlying primary partition.
- The `syncer/rate` key controls the rate, in bytes per second, at which DRBD synchronizes disks. Set the rate to 30% of the available replication bandwidth, which is the slowest of either the I/O subsystem or the network interface. The following example assumes 100MB/s available for total replication bandwidth (0.30 * 100MB/s = 30MB/s).

Configuring DRBD

This procedure configures DRBD for deployments with either one or two NICs in each node.

1 Log in to the primary node as `root`, or as a user with superuser privileges.
2 In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3 On both nodes, identify the primary partitions to use.

```
lsblk --output=NAME,SIZE
```

Record the paths of the primary partitions in the following table. The information is needed in subsequent steps and procedures.

Node	*Isvcs-Partition*	*Metadata-Partition*	*App-Data-Partition*

4 On both nodes, edit the DRBD configuration file.
   a Open `/etc/drbd.d/global_common.conf` with a text editor.
   b Add the following values to the `global` and `common/net` sections of the file.

```
global {
  usage-count yes;
}
common {
  net {
    protocol C;
  }
}
```

   c Save the file, and then close the editor.
5 On both nodes, create a resource definition for Control Center.
   a Open `/etc/drbd.d/serviced-dfs.res` with a text editor.
   b **For a dual-NIC system**, add the following content to the file.

   Replace the variables in the content with the actual values for your environment:

```
resource serviced-dfs {
   volume 0 {
      device /dev/drbd0;
      disk Isvcs-Partition;
      meta-disk internal;
   }
   volume 1 {
      device /dev/drbd1;
      disk Metadata-Partition;
      meta-disk internal;
   }
   volume 2 {
```

```
      device /dev/drbd2;
      disk App-Data-Partition;
      meta-disk internal;
   }
   syncer   {
      rate 30M;
   }
   net      {
      after-sb-0pri discard-zero-changes;
      after-sb-1pri discard-secondary;
   }
   on Primary-Public-IP {
      address Primary-Private-IP:7789;
   }
   on Secondary-Public-IP {
      address Secondary-Private-IP:7789;
   }
}
```

c **For a single-NIC system**, add the following content to the file.

Replace the variables in the content with the actual values for your environment:

```
resource serviced-dfs {
   volume 0 {
      device /dev/drbd0;
      disk Isvcs-Partition;
      meta-disk internal;
   }
   volume 1 {
      device /dev/drbd1;
      disk Metadata-Partition;
      meta-disk internal;
   }
   volume 2 {
      device /dev/drbd2;
      disk App-Data-Partition;
      meta-disk internal;
   }
   syncer   {
      rate 30M;
   }
   net      {
      after-sb-0pri discard-zero-changes;
      after-sb-1pri discard-secondary;
   }
   on Primary-Public-IP {
      address Primary-Public-IP:7789;
   }
   on Secondary-Public-IP {
      address Secondary-Public-IP:7789;
   }
}
```

d Save the file, and then close the editor.

6 On both nodes, create device metadata and enable the new DRBD resource.

```
drbdadm create-md all && drbdadm up all
```

Initializing DRBD

Perform this procedure to initialize DRBD and the mirrored storage areas.

Note Unlike the preceding procedures, most of the steps in this procedure are performed on the primary node only.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Synchronize the storage areas of both nodes.
   a. Start the synchronization.

      ```
      drbdadm primary --force serviced-dfs
      ```

      The command may return right away, while the synchronization process continues running in the background. Depending on the sizes of the partitions, this process can take several hours.
   b. Monitor the progress of the synchronization.

      ```
      drbd-overview
      ```

      Do not proceed until the status is `UpToDate/UpToDate`, as in the following example output:

      ```
      0:serviced-dfs/0 Connected Primary/Secondary UpToDate/UpToDate
      1:serviced-dfs/1 Connected Primary/Secondary UpToDate/UpToDate
      2:serviced-dfs/2 Connected Primary/Secondary UpToDate/UpToDate
      ```

      The `Primary/Secondary` values show that the command was run on the primary node; otherwise, the values are `Secondary/Primary`. Likewise, the first value in the `UpToDate/UpToDate` field is the status of the node on which the command is run, and the second value is the status of the remote node.
3. Format the partitions for Control Center internal services data and for Control Center metadata.

   The following commands use the paths of the DRBD devices defined previously, not the paths of the primary partions.

   ```
   mkfs.xfs /dev/drbd0
   mkfs.xfs /dev/drbd1
   ```

   The commands create XFS file systems on the primary node, and DRBD mirrors the file systems to the secondary node.
4. Create a device mapper thin pool for Control Center application data.

   Likewise, this command uses the path of the DRBD device defined previously.
   a. Create a variable for 50% of the space available on the DRDB device.

      The thin pool stores application data and snapshots of the data. You can add storage to the pool at any time.

      Replace *Half-Of-Available-Space* with 50% of the space available on the DRDB device, in gigabytes. Include the symbol for gigabytes (`G`) after the numeric value.

      ```
      myFifty=Half-Of-Available-SpaceG
      ```
   b. Create the thin pool.

      ```
      serviced-storage create-thin-pool -o dm.basesize=$myFifty \
        serviced /dev/drbd2 -v
      ```

   On success, DRBD mirrors the device mapper thin pool to the secondary node.
5. Configure Control Center with the name of the new thin pool.

a. Open `/etc/default/serviced` in a text editor.
b. Locate the *SERVICED_FS_TYPE* declaration.
c. Remove the number sign character (#) from the beginning of the line.
d. Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

```
SERVICED_DM_THINPOOLDEV=/dev/mapper/serviced-serviced--pool
```

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

6. Replicate the Control Center configuration on the secondary node.
   a. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
   b. Open `/etc/default/serviced` in a text editor.
   c. Locate the *SERVICED_FS_TYPE* declaration.
   d. Remove the number sign character (#) from the beginning of the line.
   e. Add *SERVICED_DM_THINPOOLDEV* immediately after *SERVICED_FS_TYPE*.

      Replace *Thin-Pool-Name* with the name of the thin pool created previously:

      ```
      SERVICED_DM_THINPOOLDEV=Thin-Pool-Name
      ```

   f. Save the file, and then close the editor.
7. On the primary node, monitor the progress of the synchronization.

   ```
   drbd-overview
   ```

   Note Do not proceed until synchronization is complete.

8. On both nodes, stop DRBD.

   ```
   drbdadm down all
   ```

Cluster management software

Pacemaker is an open source cluster resource manager, and Corosync is a cluster infrastructure application for communication and membership services. The Pacemaker/Corosync daemon (`pcs.d`) communicates across nodes in the cluster. When `pcs.d` is installed, started, and configured, the majority of PCS commands can be run on either node in the cluster.

Installing and configuring the cluster management software

Perform this procedure to install and configure the cluster management software.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On both nodes, install the Pacemaker resource agent for Control Center.

   Pacemaker uses resource agents (scripts) to implement a standardized interface for managing arbitrary resources in a cluster. Zenoss provides a Pacemaker resource agent to manage the Control Center master host in a high-availability cluster.

   ```
   yum install -y /root/serviced-resource-agents-*.x86_64.rpm
   ```

4 Optional: Delete the package file, if desired.

```
rm /root/serviced-resource-agents-*.x86_64.rpm
```

5 On both nodes, start and enable the PCS daemon.

```
systemctl start pcsd.service && systemctl enable pcsd.service
```

6 On both nodes, set the password of the `hacluster` account.

The Pacemaker package creates the `hacluster` user account, which must have the same password on both nodes.

```
passwd hacluster
```

Creating the cluster in standby mode

Perform this procedure to create the high-availability cluster in standby mode.

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

2 Authenticate the nodes.

```
pcs cluster auth Primary-Public-Name Secondary-Public-Name
```

When prompted, enter the password of the `hacluster` account.

3 Generate and synchronize an initial (empty) cluster definition.

```
pcs cluster setup --name serviced-ha \
  Primary-Public-Name Secondary-Public-Name
```

4 Start the PCS management agents on both nodes in the cluster.

The cluster definition is empty, so starting the cluster management agents has no side effects.

```
pcs cluster start --all
```

The cluster management agents start, on both nodes.

5 Check the status.

```
pcs cluster status
```

The expected result is `Online`, for both nodes.

6 Put the cluster in standby mode.

Pacemaker begins monitoring and managing the different resources as they are defined, which can cause problems; standby mode prevents the problems.

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

7 Configure cluster services to start when the node starts.

For more information about cluster startup options, refer to the *Pacemaker documentation*.

```
systemctl enable corosync; systemctl enable pacemaker
```

8 Replicate the configuration on the secondary node.

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

b Configure cluster services to start when the node starts.

```
systemctl enable corosync; systemctl enable pacemaker
```

Property and resource options

Pacemaker provides options to support cluster configurations from small and simple to and large and complex. The following list identifies the options that support the two-node, active/passive configuration for Control Center.

`resource-stickiness=100`

Keep all resources bound to the same host.

`no-quorum-policy=ignore`

Pacemaker supports the notion of a voting quorum for clusters of three or more nodes. However, with just two nodes, if one fails, it does not make sense to have a quorum of one, therefore we disable quorums.

`stonith-enabled=false`

Fence or isolate a failed node. (The string "stonith" is an acronym for "shoot the other node in the head".)

Set this option to `false` only during the intial setup and testing period. For production use, set it to `true`. For more information about fencing, refer to the *Zenoss Resource Manager Planning Guide*.

Setting resource and property defaults

Perform this procedure to set resource and property defaults for the high-availability cluster.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Set resource and property defaults.

```
pcs resource defaults resource-stickiness=100
pcs property set no-quorum-policy=ignore
pcs property set stonith-enabled=false
```

3. Check resource defaults.

```
pcs resource defaults
```

Example result:

```
resource-stickiness: 100
```

4. Check property defaults.

```
pcs property
```

Example result:

```
Cluster Properties:
cluster-infrastructure: corosync
cluster-name: serviced-ha
dc-version: 1.1.12-a14efad
have-watchdog: false
no-quorum-policy: ignore
stonith-enabled: false
```

Defining resources

This procedure defines the following logical resources required for the cluster:

- DRBD Master/Secondary DFS set
- Two mirrored file systems running on top of DRBD:
  - `/opt/serviced/var/isvcs`
  - `/opt/serviced/var/volumes`
- `serviced` logical volume group running on top of DRBD
- Manage `serviced` storage
- The floating virtual IP address of the cluster (*HA-Virtual-IP*), which the management software assigns to the active node
- Docker
- NFS
- Control Center

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. Define a resource for the DRBD device, and a clone of that resource to act as the master.
   a. On the primary node, define a resource for the DRBD device.

```
pcs resource create DFS ocf:linbit:drbd \
  drbd_resource=serviced-dfs \
  op monitor interval=30s role=Master \
  op monitor interval=60s role=Slave
```

   b. On the primary node, define a clone of that resource to act as the master.

```
pcs resource master DFSMaster DFS \
  master-max=1 master-node-max=1 \
  clone-max=2 clone-node-max=1 notify=true
```

   For a master/slave resource, Pacemaker requires separate monitoring intervals for the different roles. In this case, Pacemaker checks the master every 30 seconds and the slave every 60 seconds.
4. Define the file systems that are mounted on the DRBD devices.
   a. On the primary node, define a resource for Control Center internal services data.

```
pcs resource create serviced-isvcs Filesystem \
  device=/dev/drbd/by-res/serviced-dfs/0 \
  directory=/opt/serviced/var/isvcs fstype=xfs
```

   b. On the primary node, define a resource for Control Center metadata.

```
pcs resource create serviced-volumes Filesystem \
  device=/dev/drbd/by-res/serviced-dfs/1 \
  directory=/opt/serviced/var/volumes fstype=xfs
```

   In the preceding definitions, `serviced-dfs` is the name of the DRBD resource defined previously, in `/etc/drbd.d/serviced-dfs.res`.
5. On the primary node, define the logical volume for `serviced` that is backed by a DRBD device.

```
pcs resource create serviced-lvm ocf:heartbeat:LVM volgrpname=serviced
```

6 On the primary node, define the storage resource for `serviced`, to ensure that the device mapper device is deactivated and unmounted properly.

```
pcs resource create serviced-storage ocf:zenoss:serviced-storage
```

7 On the primary node, define the resource that represents the floating virtual IP address of the cluster.

For dual-NIC deployments, the definition includes the `nic` key-value pair, which specifies the name of the network interface that is used for all traffic except the private DRBD traffic between the primary and seconday nodes. For single-NIC deployments, omit `nic` key-value pair.

For dual-NIC deployments, replace *HA-Virtual-IP* with the floating virtual IP address of the cluster, and replace *HA-Virtual-IP-NIC* with the name of the network interface that is bound to *HA-Virtual-IP*:

```
pcs resource create VirtualIP ocf:heartbeat:IPaddr2 \
  ip=HA-Virtual-IP nic=HA-Virtual-IP-NIC \
  cidr_netmask=32 op monitor interval=30s
```

For single-NIC deployments, replace *HA-Virtual-IP* with the floating virtual IP address of the cluster:

```
pcs resource create VirtualIP ocf:heartbeat:IPaddr2 \
  ip=HA-Virtual-IP cidr_netmask=32 op monitor interval=30s
```

8 Define the Docker resource.

a On the primary node, define the resource.

```
pcs resource create docker systemd:docker
```

b On both nodes, ensure that the automatic startup of Docker by `systemd` is disabled.

```
systemctl stop docker && systemctl disable docker
```

9 Define the NFS resource.

Control Center uses NFS to share configuration in a multi-host deployment, and failover will not work properly if NFS is not stopped on the failed node.

a On the primary node, define the resource.

```
pcs resource create nfs systemd:nfs
```

b On the primary node, disable Pacemaker monitoring of NFS health.

During normal operations, Control Center occasionally stops and restarts NFS, which could be misinterpreted by Pacemaker and trigger an unwanted failover.

```
pcs resource op remove nfs monitor interval=60s
pcs resource op add nfs monitor interval=0s
```

c On both nodes, ensure that the automatic startup of NFS by `systemd` is disabled.

```
systemctl stop nfs && systemctl disable nfs
```

10 Define the Control Center resource.

a On the primary node, define the resource.

```
pcs resource create serviced ocf:zenoss:serviced
```

b On both nodes, ensure that the automatic startup of `serviced` by `systemd` is disabled.

```
systemctl stop serviced && systemctl disable serviced
```

Pacemaker uses the default timeouts defined by the Pacemaker resource agent for Control Center to decide if `serviced` is unable to start or shutdown correctly. Starting with version 0.0.5 of the Pacemaker resource agent for Control Center, the default values for the start and stop timeouts are 360 and 130 seconds respectively.

The default startup and shutdown timeouts are based on the worst case scenario. In practice, Control Center typically starts and stops in much less time. However, this does not mean that you should decrease these timeouts. There are potential edge cases, especially for startup, where Control Center may take longer than usual to start or stop. If the start/stop timeouts for Pacemaker are set too low, and Control Center encounters one of those edge cases, then Pacemaker takes unnecessary or incorrect actions. For example, if the startup timeout is artificially set too low, 2.5 minutes for example, and Control Center startup encounters an unusual case where it requires at least 3 minutes to start, then Pacemaker initiates failover prematurely.

Defining the Control Center resource group

The resources in a resource group are started in the order they appear in the group, and stopped in the reverse order they appear in the group. The start order is:

1. Mount the file systems (`serviced-isvcs` and `serviced-volumes`)
2. Start the `serviced` logical volume.
3. Manage `serviced` storage.
4. Enable the virtual IP address of the cluster.
5. Start Docker.
6. Start NFS.
7. Start Control Center.

In the event of a failover, Pacemaker stops the resources on the failed node in the reverse order they are defined before starting the resource group on the standby node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Create the Control Center resource group.

```
pcs resource group add serviced-group \
  serviced-isvcs serviced-volumes \
  serviced-lvm serviced-storage \
  VirtualIP docker nfs \
  serviced
```

3. Define constraints for the Control Center resource group.

   Pacemaker resource constraints control when and where resources are deployed in a cluster.

   a Ensure that `serviced-group` runs on the same node as `DFSMaster`.

   ```
   pcs constraint colocation add serviced-group with DFSMaster \
     INFINITY with-rsc-role=Master
   ```

   b Ensure that `serviced-group` is only started after `DFSMaster` is started.

   ```
   pcs constraint order promote DFSMaster then \
     start serviced-group
   ```

Verification procedures

The cluster is created in standby mode while various configurations are created. Perform the procedures in the following sections to review the configurations and make adjustments as necessary.

Verifying the DRBD configuration

This procedure reviews the DRBD configuration.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On the primary node, display the full DRBD configuration.

   ```
   drbdadm dump
   ```

   The result should be consistent with the configuration created previously. For more information, see *DRBD configuration assumptions* on page 79.
4. On the primary node, display the synchronization status of mirrored storage areas.

   ```
   drbd-overview
   ```

   Do not proceed until the synchronization is complete. The process is complete when the status of the devices is `UpToDate/UpToDate`.
5. On both nodes, stop DRBD.

   ```
   drbdadm down all
   ```

Verifying the Pacemaker configuration

This procedure reviews the resource and property defaults for Pacemaker.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. Check resource defaverify-pacemaker.ditaults.

   ```
   pcs resource defaults
   ```

   Example result:

   ```
   resource-stickiness: 100
   ```
3. Check property defaults.

   ```
   pcs property
   ```

   Example result:

   ```
   Cluster Properties:
    cluster-infrastructure: corosync
    cluster-name: serviced-ha
    dc-version: 1.1.12-a14efad
    have-watchdog: false
    no-quorum-policy: ignore
    stonith-enabled: false
   ```

Note Set the `stonith-enabled` option to `false` only during the intial setup and testing period. For production use, set it to `true`. For more information about fencing, refer to the *Zenoss Resource Manager Planning Guide*.

4 Review the resource constraints.

The ordering constraint should show that `serviced-group` starts after `DFSMaster` (the DRBD master). The colocation constraint should show that `serviced-group` resource and `DFSMaster` are on the same active cluster node.

```
pcs constraint
```

Example result:

```
Location Constraints:
Ordering Constraints:
  promote DFSMaster then start serviced-group (kind:Mandatory)
Colocation Constraints:
  serviced-group with DFSMaster (score:INFINITY) (with-rsc-
role:Master)
```

5 Review the ordering of the `serviced-group` resource group.

```
pcs resource show --full
```

The resources in a resource group are started in the order they appear in the group, and stopped in the reverse order they appear in the group. The correct start order is:

1 serviced-isvcs
2 serviced-volumes
3 serviced-lvm
4 serviced-storage
5 VirtualIP
6 Docker
7 nfs
8 serviced

Verifying the Control Center configuration

This procedure verifies that the Control Center configuration is identical on both nodes.

1 Log in to the primary node as `root`, or as a user with superuser privileges.
2 In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3 On both nodes, compute the checksum of the Control Center configuration file.

```
cksum /etc/default/serviced
```

- If the result is identical on both nodes, the configurations are identical. Do not perform the next step.
- If the result is not identical on both nodes, there may be a difference in their configurations; proceed to the next step.

4 Optional: On both nodes, display the customized variables, if necessary.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | sort
```

Example result:

```
SERVICED_AGENT=1
SERVICED_DM_THINPOOLDEV=/dev/mapper/serviced-serviced--pool
SERVICED_DOCKER_REGISTRY=HA-Virtual-IP:5000
SERVICED_ENDPOINT=HA-Virtual-IP:4979
SERVICED_FS_TYPE=devicemapper
SERVICED_LOG_ADDRESS=HA-Virtual-IP:5042
SERVICED_LOGSTASH_ES=HA-Virtual-IP:9100
SERVICED_MASTER=1
SERVICED_OUTBOUND_IP=HA-Virtual-IP
SERVICED_STATS_PORT=HA-Virtual-IP:8443
SERVICED_ZK=HA-Virtual-IP:2181
```

Note There may only be insignificant differences between the files, such as an extra space at the beginning of a variable definition.

Verifying cluster startup

This procedure verifies the initial configuration by attempting to start the resources on one node only. With the other node in standby mode, Pacemaker does not automatically fail over to the other node.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On the primary node, determine which node is the primary DRBD node.

```
pcs status
```

Example result:

```
Cluster name: serviced-ha
Last updated: Mon Feb 22 11:37:58 2016  Last change: Mon Feb 22
 11:35:19 2016 by root via crm_attribute on Secondary-Public-Name
Stack: corosync
Current DC: Primary-Public-Name (version 1.1.13-a14efad) - partition
 with quorum
2 nodes and 10 resources configured

Node Primary-Public-Name: standby
Node Secondary-Public-Name: standby

Full list of resources:

 Master/Slave Set: DFSMaster [DFS]
 Stopped: [ Primary-Public-Name Secondary-Public-Name ]
 Resource Group: serviced-group
     serviced-isvcs  (ocf::heartbeat:Filesystem):  Stopped
     serviced-volumes  (ocf::heartbeat:Filesystem):  Stopped
     serviced-lvm  (ocf::heartbeat:LVM):  Stopped
     serviced-storage  (ocf::zenoss:serviced-storage):  Stopped
     VirtualIP  (ocf::heartbeat:IPaddr2):  Stopped
     docker  (systemd:docker):  Stopped
     nfs  (systemd:nfs):  Stopped
     serviced  (ocf::zenoss:serviced):  Stopped

PCSD Status:
  Primary-Public-Name: Online
  Secondary-Public-Name: Online
```

```
Daemon Status:
  corosync: active/disabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

The line that begins with `Current DC` identifies the primary node. Review all of the command output for errors.

4 Start DRBD.

a On the secondary node, enter the following command:

```
drbdadm up all
```

b On the primary node, enter the following commands:

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

5 Start cluster resources.

You can run `pcs` commands on either node.

```
pcs cluster unstandby Primary-Public-Name
```

6 Monitor the status of cluster resources.

```
watch pcs status
```

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

Verifying cluster failover

This procedure simulates a failover.

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

2 Enable the DRBD secondary node.

a Take the secondary node out of standby mode.

Replace *Secondary-Public-Name* with the public hostname of the secondary node:

```
pcs cluster unstandby Secondary-Public-Name
```

b Monitor the status of the secondary node.

```
pcs status
```

Do not continue until the status of the secondary node is `Online`.

3 Verify that DRBD has completely synchonized all three volumes on the secondary node.

```
drbd-overview
```

Example result:

```
0:serviced-dfs/0  Connected Primary/Secondary UpToDate/UpToDate
1:serviced-dfs/1  Connected Primary/Secondary UpToDate/UpToDate
2:serviced-dfs/2  Connected Primary/Secondary UpToDate/UpToDate
```

4 Force a failover.

Pacemaker initiates a failover when the primary node is put in standby mode.

Replace *Primary-Public-Name* with the public hostname of the primary node:

```
pcs cluster standby Primary-Public-Name
```

5 Monitor the cluster status.

```
pcs status
```

Repeat the preceding command until all resources report a status of `Started`. Resolve any issues before continuing.

6 Restore the cluster.

Replace *Primary-Public-Name* with the public hostname of the primary node:

```
pcs cluster unstandby Primary-Public-Name
```

Creating new resource pools

This procedure creates a new resource pool for the Control Center master nodes, and one or more resource pools for other hosts.

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.
2 Create a new resource pool named `master`.

```
serviced pool add master
```

3 Optional: Create additional resource pools, if desired.

No additional resource pools are required. However, many users find it useful to have pool names such as `infrastructure` and `collector-n` for groups of resource pool hosts.

Replace *Pool-Name* with the name of the pool to create:

```
serviced pool add Pool-Name
```

Repeat the preceding command as desired.

Adding master nodes to their resource pool

This procedure adds the Control Center master nodes to their resource pool, named `master`. The master nodes are added to the resource pool with their public hostnames, so that you can easily see which node is active when you log in to the Control Center browser interface.

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.
2 Display the public hostname of the current node.

```
uname -n
```

The result is either *Primary-Public-Name* or *Secondary-Public-Name*.

3 Add the current node to the `master` resource pool.

Replace *Node-Hostname* with the public hostname of the current node:

```
serviced host add Node-Hostname:4979 master
```

4 Force a failover.

Replace *Node-Hostname* with the public hostname of the current node:

```
pcs cluster standby Node-Hostname
```

5 Monitor the cluster status.

```
watch pcs status
```

Do not proceed until all resources report a status of `Started`.

6 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

7 Display the public hostname of the current node.

```
uname -n
```

8 Add the current node to the `master` resource pool.

Replace *Node-Hostname* with the public hostname of the current node:

```
serviced host add Node-Hostname:4979 master
```

9 Restore the cluster.

Replace *Standby-Node-Hostname* with the public hostname of the node that is in standby mode:

```
pcs cluster unstandby Standby-Node-Hostname
```

Control Center on resource pool hosts

Control Center resource pool hosts run the application services scheduled for the resource pool to which they belong, and for which they have sufficent RAM and CPU resources. In a high-availability deployment, a resource pool host may belong to any resource pool other than `master`, and no application services are run in the `master` pool.

Resource Manager has two broad categories of application services: Infrastructure and collection. The services associated with each category can run in the same resource pool, or can run in separate resource pools.

For improved reliability, two resource pool hosts are configured as nodes in an *Apache ZooKeeper* ensemble. The storage required for ensemble hosts is slightly different than the storage required for all other resource pool hosts: Each ensemble host requires a separate primary partition for Control Center internal services data, in addition to the primary partition for Docker data. Unless the ZooKeeper service on the Control Center master host fails, their roles in the ZooKeeper ensemble do not affect their roles as Control Center resource pool hosts.

Note The hosts for the ZooKeeper ensemble require static IP addresses, because ZooKeeper does not support hostnames in its configurations.

Repeat the procedures in the following sections for each host you wish to add to your Control Center deployment.

Verifying candidate host resources

This procedure determines whether a hosts's hardware resources and operating system are sufficient to serve as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1 Log in to the candidate host as `root`, or as a user with superuser privileges.
2 Verify that the host implements the 64-bit version of the x86 instruction set.

```
uname -m
```

- If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step
- If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host.

3 Verify that name resolution works on this host.

```
hostname -i
```

If the result is not a valid IPv4 adddress, add an entry for the host to the network nameserver, or to `/etc/hosts`.

4 Verify that the host's numeric identifier is unique.

Each host in a Control Center cluster must have a unique host identifier.

```
hostid
```

5 Determine whether the available, unused storage is sufficient.
   a Display the available storage devices.

```
lsblk --output=NAME,SIZE
```

   b Compare the available storage with the amount required for a resource pool host in your deployment.

   In particular, resource pool hosts that are configured as nodes in a ZooKeeper ensemble require an additional primary partition for Control Center internal services data.

   For more information, refer to the *Zenoss Resource Manager Planning Guide*.

6 Determine whether the available memory and swap is sufficient.
   a Display the available memory.

```
free -h
```

   b Compare the available memory with the amount required for a resource pool host in your deployment.

   For more information, refer to the *Zenoss Resource Manager Planning Guide*.

7 Verify the operating system release.

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

If the result includes `7.0`, select another host or upgrade the operating system.

Staging files for offline installation

To perform this procedure, you need the portable storage medium that contains the archive files used in installing the master host.

This procedure adds files for offline installation to a resource pool host. The files are required in subsequent procedures.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the target host as `root`, or as a user with superuser privileges.
2. Copy `yum-mirror-*.x86_64.rpm` from your portable storage medium to `/tmp`.
3. Install the Resource Manager repository mirror.

```
yum install -y /tmp/yum-mirror-*.x86_64.rpm
```

4. Optional: Delete the package file, if desired.

```
rm /tmp/yum-mirror-*.x86_64.rpm
```

Preparing a resource pool host

This procedure prepares a RHEL/CentOS 7.1 or 7.2 host as a Control Center resource pool host.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the candidate resource pool host as `root`, or as a user with superuser privileges.
2. Add an entry to `/etc/hosts` for localhost, if necessary.
   a. Determine whether `127.0.0.1` is mapped to `localhost`.

   ```
   grep 127.0.0.1 /etc/hosts | grep localhost
   ```

   If the preceding commands return no result, perform the following substep.
   b. Add an entry to `/etc/hosts` for `localhost`.

   ```
   echo "127.0.0.1   localhost" >> /etc/hosts
   ```

3. Disable the firewall, if necessary.

   This step is required for installation but not for deployment. For more information, refer to the *Zenoss Resource Manager Planning Guide*.
   a. Determine whether the `firewalld` service is enabled.

   ```
   systemctl status firewalld.service
   ```

   - If the result includes `Active: inactive (dead)`, the service is disabled. Proceed to the next step.
   - If the result includes `Active: active (running)`, the service is enabled. Perform the following substep.

   b. Disable the `firewalld` service.

   ```
   systemctl stop firewalld && systemctl disable firewalld
   ```

   On success, the preceding commands display messages similar to the following example:

   ```
   rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
   rm '/etc/systemd/system/basic.target.wants/firewalld.service'
   ```

4. Optional: Enable persistent storage for log files, if desired.

By default, RHEL/CentOS systems store log data only in memory or in a small ring-buffer in the `/run/log/journal` directory. By performing this step, log data persists and can be saved indefinitely, if you implement log file rotation practices. For more information, refer to your operating system documentation.

```
mkdir -p /var/log/journal && systemctl restart systemd-journald
```

5 Disable Security-Enhanced Linux (SELinux), if installed.

a Determine whether SELinux is installed.

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the preceding commands return a result, SELinux is installed.

b Set the operating mode to `disabled`.
Open `/etc/selinux/config` in a text editor, and change the value of the *SELINUX* variable to `disabled`.

c Confirm the new setting.

```
grep '^SELINUX=' /etc/selinux/config
```

6 Enable and start the Dnsmasq package.

```
systemctl enable dnsmasq && systemctl start dnsmasq
```

7 Install the *Nmap Ncat* utility.

The utility is used to verify ZooKeeper ensemble configurations.

```
yum --enablerepo=zenoss-mirror install -y nmap-ncat
```

8 Reboot the host.

```
reboot
```

Configuring an NTP client

This procedure configures a resource pool host to synchronize its clock with the NTP server on the Control Center master host. If you have an NTP time server inside your firewall, you may configure the host to use it; however, this procedure does not include that option.

1 Log in the Control Center resource pool host as `root`, or as a user with superuser privileges.

2 Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

3 Edit the NTP configuration file./

a Open `/etc/ntp.conf` with a text editor.

b Replace all of the lines in the file with the following lines:

```
# Point to the master time server
server HA-Virtual-IP

restrict default ignore
restrict 127.0.0.1
restrict HA-Virtual-IP mask 255.255.255.255 nomodify notrap noquery
```

```
driftfile /var/lib/ntp/drift
```

c Replace both instances of *HA-Virtual-IP* with the virtual IP address of the high-availability cluster.

d Save the file and exit the editor.

4 Synchronize the clock with the master server.

```
ntpd -gq
```

5 Enable and start the NTP daemon.

a Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

b Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

c Start `ntpd`.

```
systemctl start ntpd
```

Creating a file system for Control Center internal services

This procedure creates an XFS file system on a primary partition.

Note **Perform this procedure only on the two resource pool hosts that are designated for use in the ZooKeeper ensemble**. No other resource pool hosts run Control Center internal services, so no other pool hosts need a partition for internal services data.

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

2 Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

3 Create an XFS file system.
Replace *Isvcs-Partition* with the path of the target primary partition:

```
mkfs -t xfs Isvcs-Partition
```

4 Create the mount point for Control Center internal services data.

```
mkdir -p /opt/serviced/var/isvcs
```

5 Add an entry to the `/etc/fstab` file.
Replace *Isvcs-Partition* with the path of the primary partition used in the previous step:

```
echo "Isvcs-Partition \
  /opt/serviced/var/isvcs xfs defaults 0 0" >> /etc/fstab
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep isvcs
```

Example result:

```
/dev/xvdb1 on /opt/serviced/var/isvcs type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Installing Docker and Control Center

This procedure installs and configures Docker, and installs Control Center.

Perform this procedure on each resource pool host in your deployment.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Install Docker 1.9.0.

```
yum clean all && yum makecache fast
yum install --enablerepo=zenoss-mirror -y docker-engine
```

3 Create a symbolic link for the Docker temporary directory.

Docker uses its temporary directory to spool images. The default directory is `/var/lib/docker/tmp`. The following command specifies the same directory that Control Center uses, `/tmp`. You can specify any directory that has a minimum of 10GB of unused space.

a Create the `docker` directory in `/var/lib`.

```
mkdir /var/lib/docker
```

b Create the link to `/tmp`.

```
ln -s /tmp /var/lib/docker/tmp
```

4 Create a `systemd` override file for the Docker service definition.

a Create the override directory.

```
mkdir -p /etc/systemd/system/docker.service.d
```

b Create the override file.

```
cat <<EOF > /etc/systemd/system/docker.service.d/docker.conf
[Service]
TimeoutSec=300
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=
ExecStart=/usr/bin/docker daemon \$OPTIONS -H fd://
EOF
```

c Reload the `systemd` manager configuration.

```
systemctl daemon-reload
```

5 Install Control Center.

Control Center includes a utility that simplifies the process of creating a device mapper thin pool.

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

```
yum --enablerepo=zenoss-mirror install -y serviced
```

6 Create a device mapper thin pool for Docker data.

a Identify the primary partition for the thin pool to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

b Create the thin pool.

Replace *Path-To-Device* with the path of an unused primary partition:

```
serviced-storage create-thin-pool docker Path-To-Device
```

On success, the result includes the name of the thin pool, which always starts with `/dev/mapper`.

7 Configure and start the Docker service.

a Create variables for adding arguments to the Docker configuration file.

The `--exec-opt` argument is a workaround for *a Docker issue* on RHEL/CentOS 7.x systems.

Replace *Thin-Pool-Device* with the name of the thin pool device created in the previous step:

```
myDriver="-s devicemapper"
myFix="--exec-opt native.cgroupdriver=cgroupfs"
myFlag="--storage-opt dm.thinpooldev"
myPool="Thin-Pool-Device"
```

b Add the arguments to the Docker configuration file.

```
echo 'OPTIONS="'$myDriver $myFix $myFlag'='$myPool'"' \
  >> /etc/sysconfig/docker
```

c Start or restart Docker.

```
systemctl restart docker
```

The initial startup takes up to a minute, and may fail. If the startup fails, repeat the previous command.

8 Configure name resolution in containers.

Each time it starts, `docker` selects an IPv4 subnet for its virtual Ethernet bridge. The selection can change; this step ensures consistency.

a Identify the IPv4 subnet and netmask `docker` has selected for its virtual Ethernet bridge.

```
ip addr show docker0 | grep inet
```

b Open `/etc/sysconfig/docker` in a text editor.

c Add the following flags to the end of the *OPTIONS* declaration.

Replace *Bridge-Subnet* with the IPv4 subnet `docker` selected for its virtual bridge, and replace *Bridge-Netmask* with the netmask `docker` selected:

```
--dns=Bridge-Subnet --bip=Bridge-Subnet/Bridge-Netmask
```

For example, if the bridge subnet and netmask is 172.17.0.1/16, the flags to add are `--dns=172.17.0.1 --bip=172.17.0.1/16`.

Note Leave a blank space after the end of the thin pool device name, and make sure the double quote character (") is at the end of the line.

---

d Restart the Docker service.

```
systemctl restart docker
```

Configuring and starting Control Center

This procedure customizes key configuration variables of Control Center.

Perform this procedure on each resource pool host in your deployment.

1. Log in to the resource pool host as `root`, or as a user with superuser privileges.
2. Configure Control Center as an agent of the master host.

   The following variable configures `serviced` to serve as agent:

   `SERVICED_AGENT`

   Default: `0` (false)

   Determines whether a `serviced` instance performs agent tasks. Agents run application services scheduled for the resource pool to which they belong. The `serviced` instance configured as the master runs the scheduler. A `serviced` instance may be configured as agent and master, or just agent, or just master.

   `SERVICED_MASTER`

   Default: `0` (false)

   Determines whether a `serviced` instance performs master tasks. The master runs the application services scheduler and other internal services, including the server for the Control Center browser interface. A `serviced` instance may be configured as agent and master, or just agent, or just master. Only one `serviced` instance in a Control Center cluster may be the master.

   In addition, replace `{{SERVICED_MASTER_IP}}` with *HA-Virtual-IP*, the virtual IP address of the high-availability cluster, in the following lines::

   ```
   # SERVICED_ZK={{SERVICED_MASTER_IP}}:2181
   # SERVICED_DOCKER_REGISTRY={{SERVICED_MASTER_IP}}:5000
   # SERVICED_ENDPOINT={{SERVICED_MASTER_IP}}:4979
   # SERVICED_LOG_ADDRESS={{SERVICED_MASTER_IP}}:5042
   # SERVICED_LOGSTASH_ES={{SERVICED_MASTER_IP}}:9100
   # SERVICED_STATS_PORT={{SERVICED_MASTER_IP}}:8443
   ```

   a Open `/etc/default/serviced` in a text editor.

   b Find the *SERVICED_AGENT* declaration, and then change the value from `0` to `1`.
   The following example shows the line to change:

   ```
   # SERVICED_AGENT=0
   ```

   c Remove the number sign character (#) from the beginning of the line.

   d Find the *SERVICED_MASTER* declaration, and then remove the number sign character (#) from the beginning of the line.

   e Globally replace `{{SERVICED_MASTER_IP}}` with the virtual IP address of the high-availability cluster (*HA-Virtual-IP*).

   Note Remove the number sign character (#) from the beginning of each variable declaration that includes the virtual IP address.

   f Save the file, and then close the editor.
3. Optional: Specify an alternate private network for Control Center, if necessary.

Control Center requires a 16-bit, private IPv4 network for virtual IP addresses, independent of the private network used in a dual-NIC DRBD configuration. The default network is 10.3/16. If the default network is already in use in your environment, you may select any valid IPv4 16-bit network.

The following variable configures `serviced` to use an alternate network:

`SERVICED_VIRTUAL_ADDRESS_SUBNET`

Default: `10.3`

The 16-bit private subnet to use for `serviced`'s virtual IPv4 addresses. RFC 1918 restricts private networks to the 10.0/24, 172.16/20, and 192.168/16 address spaces. However, `serviced` accepts any valid, 16-bit, IPv4 address space for its private network.

a Open `/etc/default/serviced` in a text editor.

b Locate the *SERVICED_VIRTUAL_ADDRESS_SUBNET* declaration, and then change the value. The following example shows the line to change:

```
# SERVICED_VIRTUAL_ADDRESS_SUBNET=10.3
```

c Remove the number sign character (#) from the beginning of the line.

d Save the file, and then close the editor.

4 Start the Control Center service (`serviced`).

```
systemctl start serviced
```

To monitor progress, open a separate window to the host, and then enter the following command:

```
journalctl -flu serviced -o cat
```

Deploying Resource Manager

This procedure adds all of the resource pool hosts to the Control Center cluster, and then deploys the Resource Manager application.

1 Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.

2 Display the public hostname of the current node.

```
uname -n
```

The result is either *Primary-Public-Name* or *Secondary-Public-Name*.

3 Place the other node in standby mode.

This avoids potential conflicts and errors in the event of an unexpected `serviced` shutdown during the initial deployment.

Replace *Other-Node-Hostname* with the public hostname of the other node:

```
pcs cluster standby Other-Node-Hostname
```

4 Add resource pool hosts to resource pools.

Replace *Hostname-Or-IP* with the hostname or IP address of the resource pool host to add, and replace *Resource-Pool-Name* with the name of a resource pool created previously, or with `default`:

```
serviced host add Hostname-Or-IP:4979 Resource-Pool-Name
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Repeat this step for each resource pool host in your deployment.

5 Add the `Zenoss.resmgr` application to Control Center.

```
myPath=/opt/serviced/templates
serviced template add $myPath/zenoss-resmgr-*.json
```

On success, the `serviced` command returns the template ID.

6 Deploy the application.

Replace *Template-ID* with the template identifier returned in the previous step, and replace *Deployment-ID* with a name for this deployment (for example, `Dev` or `Test`):

```
serviced template deploy Template-ID default Deployment-ID
```

Control Center pulls Resource Manager images into the local registry. To monitor progress, open a separate window, and enter the following command:

```
journalctl -flu serviced -o cat
```

7 Restore the cluster.

Replace *Standby-Node-Hostname* with the public hostname of the node that is in standby mode:

```
pcs cluster unstandby Standby-Node-Hostname
```

ZooKeeper ensemble configuration

Control Center relies on *Apache ZooKeeper* to coordinate its services. The configuration steps in this section create a ZooKeeper ensemble of 3 nodes.

A ZooKeeper ensemble requires a minimum of 3 nodes, and 3 nodes is sufficient for most deployments. A 5-node configuration improves failover protection during maintenance windows. Ensembles larger than 5 nodes are not necessary. An odd number of nodes is recommended, and an even number of nodes is strongly discouraged.

Control Center variables for ZooKeeper

This tables in this section associates the ZooKeeper-related Control Center variables to set in `/etc/default/serviced` with the roles that hosts play in a Control Center cluster.

Table 10: Control Center master nodes

SERVICED_ISVCS_ZOOKEEPER_ID The unique identifier of a ZooKeeper ensemble node. Value: 1
SERVICED_ISVCS_ZOOKEEPER_QUORUM The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0. Value: *ZooKeeper-ID*`@`*IP-Address*`:2888:3888,...`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 11: Control Center resource pool host and ZooKeeper ensemble node

SERVICED_ISVCS_ZOOKEEPER_ID

The unique identifier of a ZooKeeper ensemble node.
Value: 2 or 3

SERVICED_ISVCS_ZOOKEEPER_QUORUM

The ZooKeeper node ID, IP address, peer communications port, and leader communications port of each host in an ensemble. Each quorum definition must be unique, so the IP address of the "current" host is 0.0.0.0.
Value: *`ZooKeeper-ID`*`@`*`IP-Address`*`:2888:3888,...`

SERVICED_ISVCS_START

The list of Control Center internal services to start and run on hosts other than the master host.
Value: `zookeeper`

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Table 12: Control Center resource pool host

SERVICED_ZK

The list of endpoints in the Control Center ZooKeeper ensemble, separated by the comma character (`,`). Each endpoint includes the IP address of the ensemble node, and the port that Control Center uses to communicate with it.
Value: *`IP-Address`*`:2181,...`

Configuring a master node as a ZooKeeper node

This procedure configures both Control Center master nodes as members of the ZooKeeper ensemble.

Note For accuracy, this procedure constructs Control Center configuration variables in the shell and appends them to `/etc/default/serviced`. The last step is to move the variables from the end of the file to more appropriate locations.

1. Log in to the primary node as `root`, or as a user with superuser privileges.
2. In a separate window, log in to the secondary node as `root`, or as a user with superuser privileges.
3. On both nodes, create a variable for each Control Center host to include in the ZooKeeper ensemble.
   The variables are used in subsequent steps.

Note Define the variables identically on both the primary and the secondary nodes, and on each resource pool host.

Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=HA-Virtual-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

Note ZooKeeper requires IP addresses for ensemble configuration.

4 On both nodes, set the ZooKeeper node ID to `1`.

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=1" >> /etc/default/serviced
```

5 On both nodes, specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

6 On both nodes, specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

You may copy the following of text and paste it in your console:

```
q1="1@0.0.0.0:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

7 On both nodes, clean up the Control Center configuration file.

a Open `/etc/default/serviced` in a text editor.

b Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

The value of this declaration specifies 3 hosts.

c Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

The value of this declaration is just the master node.

d Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.

e Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.

f Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

g Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.

h Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.

i Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

This declaration is commented out.

j Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.

k Save the file, and then close the editor.

8 On both hosts, verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced | egrep '(_ZOO|_ZK)'
```

Configuring a resource pool host as a ZooKeeper node

To perform this procedure, you need a resource pool host with an XFS file system on a separate partition.

This procedure configures a ZooKeeper ensemble on a resource pool host. Repeat this procedure on each Control Center resource pool host to add to the ZooKeeper ensemble.

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

2 Create a variable for each Control Center host to include in the ZooKeeper ensemble.

Replace *HA-Virtual-IP* with the virtual IP address of the high-availability cluster, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

```
node1=HA-Virtual-IP
node2=Pool-Host-A-IP
node3=Pool-Host-B-IP
```

3 Set the ID of this node in the ZooKeeper ensemble.

For *Pool-Host-A-IP* (**node2**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=2" >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following command:

```
echo "SERVICED_ISVCS_ZOOKEEPER_ID=3" >> /etc/default/serviced
```

4 Specify the nodes in the ZooKeeper ensemble.
You may copy the following text and paste it in your console:

```
echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
  >> /etc/default/serviced
```

5 Specify the nodes in the ZooKeeper quorum.

ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

For *Pool-Host-A-IP* (**node2**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@0.0.0.0:2888:3888"
q3="3@${node3}:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@0.0.0.0:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

6 Set the *SERVICED_ISVCS_START* variable, and clean up the Control Center configuration file.
   a Open `/etc/default/serviced` in a text editor.
   b Locate the *SERVICED_ISVCS_START* variable, and then delete all but `zookeeper` from its list of values.
   c Remove the number sign character (#) from the beginning of the line.
   d Navigate to the end of the file, and cut the line that contains the *SERVICED_ZK* variable declaration at that location.

      The value of this declaration specifies 3 hosts.
   e Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

      The value of this declaration is just the master node.
   f Paste the *SERVICED_ZK* variable declaration from the end of the file in the location of the just-deleted declaration.
   g Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration at that location.
   h Locate the *SERVICED_ISVCS_ZOOKEEPER_ID* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   i Paste the *SERVICED_ISVCS_ZOOKEEPER_ID* variable declaration from the end of the file in the location of the just-deleted declaration.
   j Navigate to the end of the file, and cut the line that contains the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration at that location.
   k Locate the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable near the end of the file, and then delete the line it is on.

      This declaration is commented out.
   l Paste the *SERVICED_ISVCS_ZOOKEEPER_QUORUM* variable declaration from the end of the file in the location of the just-deleted declaration.
   m Save the file, and then close the editor.
7 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced \
  | egrep '(_ZOO|_ZK|_STA)'
```

8 Pull the required Control Center ZooKeeper image from the master host.
   a Identify the image to pull.

      ```
      serviced version | grep IsvcsImages
      ```

      Example result:

      ```
      IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
      ```
   b Pull the Control Center ZooKeeper image.

Replace *Isvcs-ZK-Image* with the name and version number of the ZooKeeper image from the previous substep:

```
docker pull Isvcs-ZK-Image
```

Starting a ZooKeeper ensemble

This procedure starts a ZooKeeper ensemble.

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

1. Use the virtual hostname (*HA-Virtual-Name*) or virtual IP address (*HA-Virtual-IP*) of the high-availability cluster to start a Bash shell on the Control Center master host as `root`, or as a user with superuser privileges.
2. Display the public hostname of the current node.

   ```
   uname -n
   ```

   The result is either *Primary-Public-Name* or *Secondary-Public-Name*.
3. Place the other node in standby mode.

   This avoids potential conflicts and errors in the event of an unexpected `serviced` shutdown during the ZooKeeper startup.

   Replace *Other-Node-Hostname* with the public hostname of the other node:

   ```
   pcs cluster standby Other-Node-Hostname
   ```
4. In a separate window, log in to the second node of the ZooKeeper ensemble (*Pool-Host-A-IP*).
5. In another separate window, log in to the third node of the ZooKeeper ensemble (*Pool-Host-B-IP*).
6. On all ensemble hosts, stop and start `serviced`.

   ```
   systemctl stop serviced && systemctl start serviced
   ```
7. On the master host, check the status of the ZooKeeper ensemble.

   ```
   { echo stats; sleep 1; } | nc localhost 2181 | grep Mode
   { echo stats; sleep 1; } | nc Pool-Host-A-IP 2181  | grep Mode
   { echo stats; sleep 1; } | nc Pool-Host-B-IP 2181  | grep Mode
   ```

   If `nc` is not available, you can use `telnet` with *interactive ZooKeeper commands*.
8. Restore the cluster.

   Replace *Other-Node-Hostname* with the public hostname of the primary node:

   ```
   pcs cluster unstandby Other-Node-Hostname
   ```

Updating resource pool hosts

The default configuration of resource pool hosts sets the value of the *SERVICED_ZK* variable to the master host only. This procedure updates the setting to include the full ZooKeeper ensemble.

Perform this procedure on each resource pool host in your Control Center cluster.

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

2. Update the variable.
   a. Open `/etc/default/serviced` in a text editor.
   b. Locate the *SERVICED_ZK* declaration, and then replace its value with the same value used in the ZooKeeper ensemble nodes.
   c. Save the file, and then close the editor.
3. Restart Control Center.

```
systemctl restart serviced
```

Part III: Appliance deployments

The chapters in this part describe how to install the Resource Manager appliance, a pre-configured virtual machine that is ready to deploy to your hypervisor. The instructions include a variety of options for customizing your deployment for your environment.

1 Installing a Control Center master host

This chapter describes how to install a Resource Manager appliance package as a Control Center master host, using either VMware vSphere or Microsoft Hyper-V.

The procedures in this chapter configure a Control Center master host that functions as both master and agent. Perform the procedures in this chapter whether you are configuring a single-host or a multi-host deployment. (For more information about configuring a multi-host deployment, see *Configuring a multi-host Control Center cluster* on page 178.)

The procedures in this chapter do not include adding storage for backups created by Control Center. Hypervisor backups of a Resource Manager host do not capture the information needed to restore a system successfully, and Zenoss strongly recommends using the Control Center backup and restore features instead of hypervisor backups. For more information about adding storage for backups, see *Adding storage for backups* on page 163. For more information about the Control Center backup and restore features, refer to the *Zenoss Resource Manager Administration Guide*.

Creating a virtual machine

You may create a virtual machine for the Resource Manager appliance with VMware vSphere or Microsoft Hyper-V. Choose one of the procedures in this section.

Creating a virtual machine with vSphere

To perform this task, you need:

- A VMware vSphere client
- Permission to download Resource Manager software from the *Zenoss Support* site

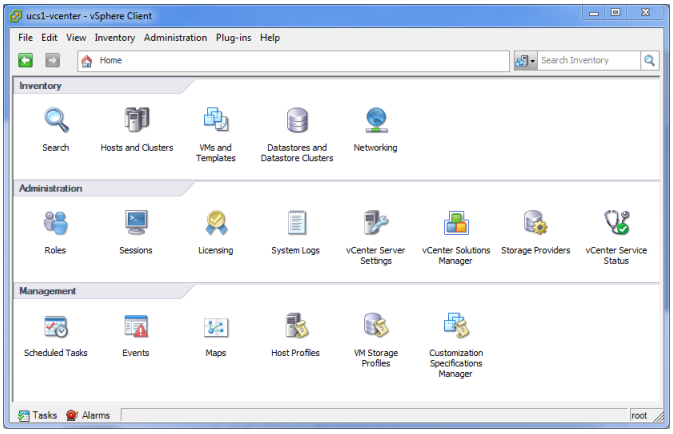
This procedure installs the Resource Manager OVA package as a virtual machine managed by vSphere Server version 5.0.0, using VMware vSphere Client 5.0.0. The procedure is slightly different with different versions of VMware vSphere Client.

Note VMware vSphere Client 5.0.0 does not include a library that is needed to deploy compressed OVA files. You may uncompress the OVA package and then deploy it, or download and install *the missing library*. Zenoss recommends installing the library.

---

1. Download the Resource Manager OVA file from the *Zenoss Support* site to your workstation.
2. Use the VMware vSphere Client to log in to vCenter as `root`, or as a user with superuser privileges, and then display the **Home** view.

Figure 1: vSphere client Home view



3 From the **File** menu, select **Deploy OVF Template...**.
4 In the **Source** panel, specify the path of the Resource Manager package, and then click **Next**.
5 In the **OVF Template Details** panel, click **Next**.
6 In the **Name and Location** panel, provide a name and a location for the server.
   a In the **Name** field, enter a new name or use the default.
   b In the **Inventory Location** area, select a data center for the virtual machine.
   c Click **Next**.
7 In the **Host / Cluster** panel, select a host system, and then click **Next**.
8 In the **Storage** panel, select a storage system with sufficient space for your deployment, and then click **Next**.
9 In the **Disk Format** panel, select select **Thin Provison**, and then click **Next**.
10 In the **Ready to Complete** panel, review the deployment settings, and then click **Finish**.

   Please do not check the check box labeled **Power on after deployment**.
11 Navigate to the new virtual machine's **Getting Started** or **Summary** tab, and then click the **Edit virtual machine settings** link.
12 Update the memory assigned to the machine.
   a In the **Virtual Machine Properties** dialog, select **Memory** in the **Hardware** table.
   b In the **Memory Configuration** area, set the **Memory Size** field to 16GB (multi-host deployments) or 32GB (single-host deployments).

      For single-host deployments, you may assign a greater amount of RAM.
13 Optional: Update the number of CPU sockets assigned to the machine, if desired.

   Only 4 CPUs are needed for multi-host deployments.
   a In the **Virtual Machine Properties** dialog, select **CPUs** in the **Hardware** table.
   b Set the **Number of virtual sockets** field to 4 (multi-host deployments), and set the **Number of cores per socket** field to 1.
14 At the bottom of the the **Virtual Machine Properties** dialog, click the **OK** button.
15 On the new virtual machine's **Getting Started** tab, click the **Power on virtual machine** link, and then click the **Console** tab.

Creating a virtual machine with Hyper-V

To perform this task, you need:

- A Microsoft Remote Desktop client
- Administrator privileges on a Microsoft Hyper-V server
- Permission to download Resource Manager software from the *Zenoss Support* site

This procedure installs the Resource Manager appliance as a virtual machine managed by Microsoft Hyper-V.

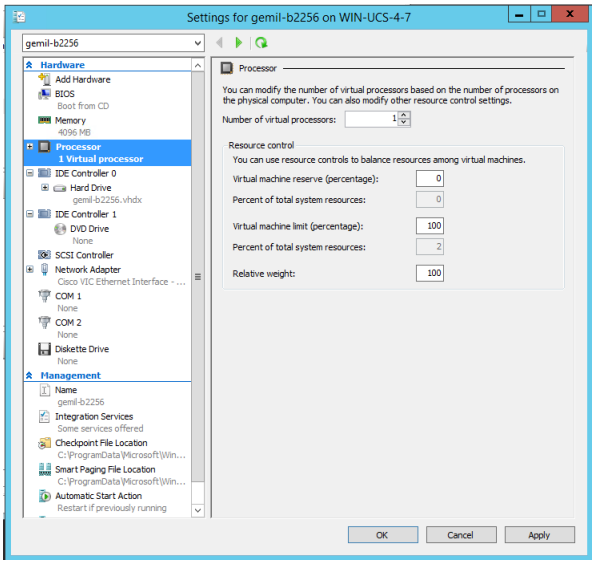
1 Use a Microsoft Remote Desktop client to log in to a Hyper-V host as Administrator, or as a user with Administrator privileges.
2 Download the Resource Manager ISO file from the *Zenoss Support* site to the Hyper-V host.
3 Start **Hyper-V Manager**.
4 In the left column, select a server to host the virtual machine.
5 From the **Action** menu, select **New** > **Virtual Machine...**.
6 In the **New Virtual Machine Wizard** dialog, display the **Specify Name and Location** panel.

   If the first panel displayed is the **Before You Begin** panel, click **Next**.
7 In the **Specify Name and Location** panel, provide a name for the virtual machine, and then click **Next**.
8 In the **Specify Generation** panel, select **Generation 1**, and then click **Next**.
9 In the **Assign Memory** panel, enter 16384 (16GB; multi-host deployments) or 32768 (32GB; single-host deployments) in the **Startup memory** field, and then click **Next**.

   For single-host deployments, you may assign a greater amount of RAM.
10 In the **Configure Networking** panel, select a virtual switch, and then click **Next**.
11 In the **Connect Virtual Hard Disk** panel, select **Create a virtual hard disk**, enter `335` in the **Size** field, and then click **Next**.
12 In the **Installation Options** panel, specify the Resource Manager ISO package.
   a Select **Install an operating system from a bootable CD/DVD-ROM**.
   b Select **Image file (.iso)**, and then specify the location of the Resource Manager ISO image file.
   c Click **Next**.
13 In the **Summary** panel, review the virtual machine specification, and then click **Finish**.

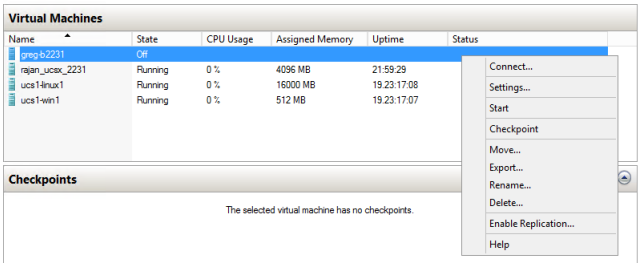
   Hyper-V Manager creates the new virtual machine, and then closes the **New Virtual Machine Wizard** dialog.
14 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Settings...**.
15 In the **Hardware** area of the **Settings** dialog, select **Processor**.

Figure 2: Settings dialog, Processor selected



16 In the **Processor** area, enter 4 (multi-host deployments) or 8 (single-host deployments) in the **Number of virtual processors** field, and then click **OK**.
17 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Start**.

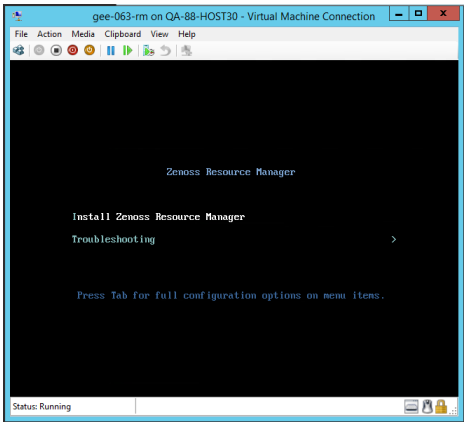
Figure 3: Starting a virtual machine



18 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Connect**.

19 In the **Virtual Machine Connection** window, press the **Enter** key.

Figure 4: Appliance installation start screen

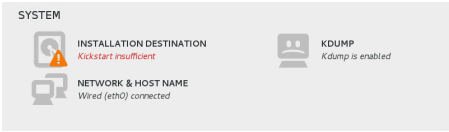


The appliance installation process takes about 15 minutes, and should complete with no additional input.

20 Optional: Select the installation destination, if necessary.

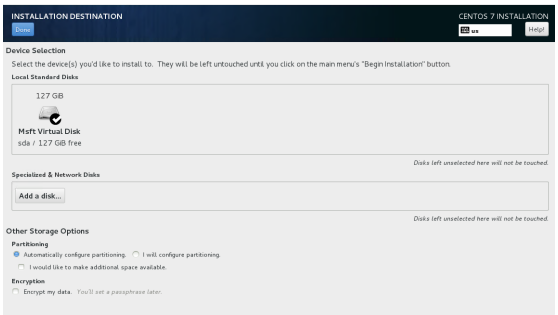
Occasionally, installation is interrupted with the `Kickstart insufficient` message.

Figure 5: Kickstart insufficient message



a In the **SYSTEM** area of the **INSTALLATION SUMMARY** page, click the **INSTALLATION DESTINATION** control.

Figure 6: INSTALLATION DESTINATION page



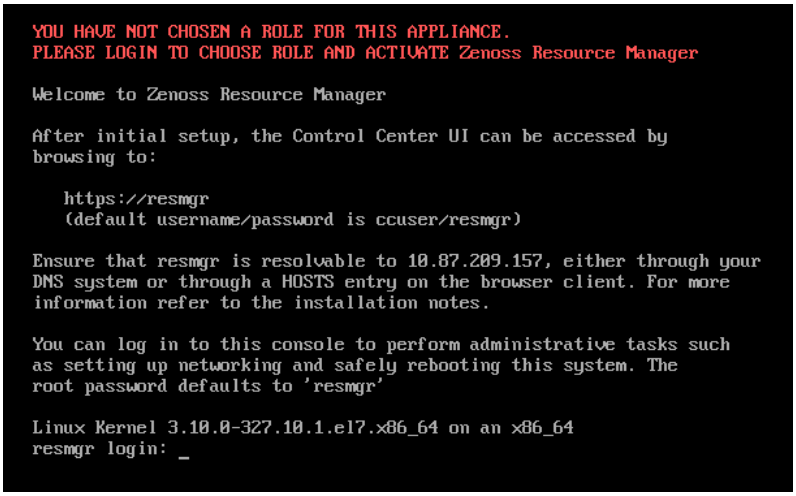
b On the **INSTALLATION DESTINATION** page, click the **Done** button, located at the upper-left corner of the page.
c On the **INSTALLATION SUMMARY** page, click the **Begin Installation** button, located at the bottom-right corner of the page.

Configuring the Control Center host mode

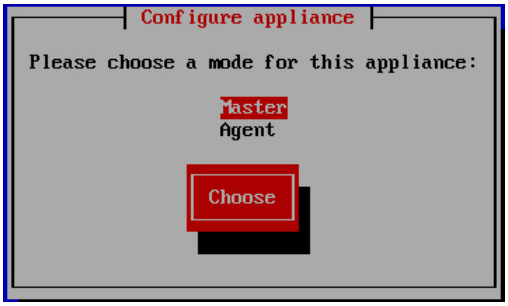
Perform this procedure immediately after creating and starting a Control Center host. All Control Center deployments must include one system configured as the master host.

1 Gain access to the console interface of the Control Center host through your hypervisor console interface.

Figure 7: Initial hypervisor console login prompt



2 Log in as the `root` user.
The initial password is provided in the console.
3 The system prompts you to enter a new password for `root`.
4 The system prompts you to enter a new password for `ccuser`.
The `ccuser` acount is the default account for gaining access to the Control Center browser interface.
5 Select the master role for the host.



a In the **Configure appliance** menu, press the **Tab** key to select the **Choose** button.
b Press the **Enter** key.

The system reboots.

Edit a connection

The default configuration for network connections is DHCP. To configure static IPv4 addressing, perform this procedure.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*.
2. Log in as the `root` user.
3. Select the **NetworkManager TUI** menu.
   a. In the **Appliance Administration** menu, select the **Configure Network and DNS** option.
   b. Press the **Tab** key to select the **Run** button.
   c. Press the **Enter** key.
4. On the **NetworkManager TUI** menu, select **Edit a connection**, and then press the **Return** key.
   The TUI displays the connections that are available on this host.

   Figure 8: Example: Available connections

   Note: Do not modify the `docker0` connection.
5. Use the down-arrow key to select the virtual connection, and then press the **Return** key.

   Figure 9: Example: Edit Connection screen

   Use the **Tab** key and the arrow keys to navigate among options in the **Edit Connection** screen, and use the **Return** key to toggle an option or to display a menu of options.
6. Optional: If the **IPv4 CONFIGURATION** area is not visible, select its display option (**<Show>**), and then press the **Return** key.
7. In the **IPv4 CONFIGURATION** area, select **<Automatic>**, and then press the **Return** key.

   Figure 10: Example: IPv4 Configuration options

8. Configure static IPv4 networking.
   a. Use the down arrow key to select **Manual**, and then press the **Return** key.
   b. Use the **Tab** key or the down arrow key to select the **<Add...>** option next to **Addresses**, and then press the **Return** key.
   c. In the **Addresses** field, enter an IPv4 address for the virtual machine, and then press the **Return** key.
   d. Repeat the preceding two steps for the **Gateway** and **DNS servers** fields.
9. Use the **Tab** key or the down arrow key to select the **<OK>** option at the bottom of the **Edit Connection** screen, and then press the **Return** key.
10. In the available connections screen, use the **Tab** key to select the **<Quit>** option, and then press the **Return** key.
11. Reboot the operating system.
    a. In the **Appliance Administration** menu, use the down-arrow key to select the **Reboot / Poweroff System** option.
    b. Use the **Down Arrow** key to select **Reboot**.
    c. Press the **Tab** key to select **OK**, and then press the **Return** key.

Set system hostname

The default hostname of a Resource Manager appliance host is `resmgr`. To change the hostname, perform this procedure.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Select the **NetworkManager TUI** menu.
   a. In the **Appliance Administration** menu, select the **Configure Network and DNS** option.
   b. Press the **Tab** key to select the **Run** button.
   c. Press the **Enter** key.
3. Display the hostname entry field.
   a. In the **NetworkManager TUI** menu, use the down-arrow key to select **Set system hostname**.
   b. Press the **Tab** key to select the **OK** button.
   c. Press the **Enter** key.
4. In the **Hostname** field, enter the new hostname.

   You may enter either a hostname or a fully-qualified domain name.
5. Press the **Tab** key twice to select the **OK** button, and then press the **Enter** key.
6. In the confirmation dialog, press the **Return** key.
7. Reboot the operating system.
   a. In the **Appliance Administration** menu, use the down-arrow key to select the **Reboot / Poweroff System** option.
   b. Use the **Down Arrow** key to select **Reboot**.
   c. Press the **Tab** key to select **OK**, and then press the **Return** key.

Adding the master host to a resource pool

This procedure adds the Control Center master host to the `default` resource pool, or to a new resource pool, named `master`.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.
   a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b. Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

   ```
   [root@resmgr ~]#
   ```
3. Optional: Create a new resource pool, if necessary.

   - For single-host deployments, skip this step.
   - For multi-host deployments with at least two resource pool hosts, perform this step.

   ```
   serviced pool add master
   ```
4. Add the master host to a resource pool.

   - For single-host deployments, add the master host to the `default` resource pool.
   - For multi-host deployments with at least two resource pool hosts, add the master host to the `master` resource pool.

Replace *Hostname-Or-IP* with the hostname or IP address of the Control Center master host, and replace *Resource-Pool* with `default` or `master`:

```
serviced host add Hostname-Or-IP:4979 Resource-Pool
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

Deploying Resource Manager

This procedure adds the Resource Manager application to the list of applications that Control Center manages.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.
   a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b. Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

   ```
   [root@resmgr ~]#
   ```
3. Add the `Zenoss.resmgr` application to Control Center.

   ```
   myPath=/opt/serviced/templates
   serviced template add $myPath/zenoss-resmgr-*.json
   ```

   On success, the `serviced` command returns the template ID.
4. Deploy the application.

   Replace *Template-ID* with the template identifier returned in the previous step, and replace *Deployment-ID* with a name for this deployment (for example, `Dev` or `Test`):

   ```
   serviced template deploy Template-ID default Deployment-ID
   ```

   Control Center tags Resource Manager images in the local registry.

If you are installing a single-host deployment, proceed to the *Zenoss Resource Manager Configuration Guide*.

Adding storage for backups 2

This chapter describes how to add storage for application data backups to an appliance-based Control Center master host. Most appliance-based deployments need additional storage for backups. On the Resource Manager appliance, the default partition for backup data is the same partition as the root (/) file system, which is not sized to store backups.

You can use a remote file server for backups—you do not have to add a virtual disk device to the Control Center host, you can simply mount a remote file system.

Note The procedures in this chapter do not include size recommendations for backups storage. For more information about sizing, refer to the *Zenoss Resource Manager Planning Guide*.

The procedures in this chapter may be performed only after a Control Center master host is installed and running.

Option	Procedure
Add a remote file server for backups	*Mounting a remote file system for backups* on page 163
Add a virtual disk for backups	1 *Identifying existing virtual disks* on page 164 ■ *Creating a virtual disk with vSphere* on page 164 ■ *Creating a virtual disk with Hyper-V* on page 165 2 *Identifying new virtual disks* on page 166 3 *Creating primary partitions* on page 166 4 *Preparing a partition for backups* on page 168

Mounting a remote file system for backups

This procedure mounts a remote file system for backups.

To perform this procedure, you need a Linux-compatible remote file server, and the file system specification for the file system to mount.

1 Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2 Start a command-line session as `root`.
   a In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b Press the **Tab** key to select **Run**, and then press the **Return** key.

The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

3 Create an entry in the `/etc/fstab` file.

Replace *File-System-Specification* with the remote server specification, and replace *File-System-Type* with the file system type (such as `xfs`):

```
echo "File-System-Specification \
  /opt/serviced/var/backups File-System-Type \
  defaults 0 0" >> /etc/fstab
```

4 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep backups
```

Example result:

```
fs12:/backups/zenoss on /opt/serviced/var/backups type xfs
 (rw,relatime,seclabel,attr2,inode64,noquota)
```

Identifying existing virtual disks

This procedure identifies the virtual disks attached to an appliance-based master host.

1 Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2 Start a command-line session as `root`.
   a In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

3 Identify the virtual disks attached to the host.

```
lsblk -pdo NAME,HCTL,SIZE
```

Example output:

```
NAME     HCTL        SIZE
/dev/sda 2:0:0:0     293G
/dev/sr0 1:0:0:0    1024M
```

The example output shows two devices:

- One disk drive (`/dev/sda`)
- One CD-ROM drive (`/dev/sr0`)

Make a note of the disk devices for later comparison.

Creating a virtual disk with vSphere

To perform this task, you need a VMware vSphere client.

1. Use the VMware vSphere Client to log in to vCenter as `root`, or as a user with superuser privileges, and then display the **Home Inventory** view.
2. In the left column, right-click on the Control Center master host virtual machine, and then select **Edit Settings...**.
3. On the **Hardware** tab, click the **Add...** button.
4. In the **Add Hardware** dialog, select **Hard Disk**, and then click the **Next** button.
5. In the **Select a Disk** pane, click the **Create a new virtual disk** radio button, and then click the **Next** button.
6. In the **Create a Disk** pane, configure the virtual disk.
   a. In the **Capacity** area, set the disk size.

      For more information, refer to the *Zenoss Resource Manager Planning Guide*.
   b. In the **Disk Provisioning** area, choose the option you prefer.
   c. In the **Location** area, choose the option you prefer.
   d. Click the **Next** button.
7. In the **Advanced Options** pane, configure the mode.
   a. In the **Mode** area, check the **Independent** check box.
   b. Click the **Persistent** radio button.
   c. Click the **Next** button.
8. In the **Ready to Complete** pane, confirm the virtual disk configuration, and then click the **Finish** button.
9. At the bottom of the **Virtual Machine Properties** dialog, click the **OK** button.

Creating a virtual disk with Hyper-V

To perform this task, you need:

- A Microsoft Remote Desktop client
- Administrator privileges on a Microsoft Hyper-V server

In addition, the virtual machine to modify must be stopped.

1. Use a Microsoft Remote Desktop client to log in to a Hyper-V host as Administrator, or as a user with Administrator privileges.
2. Start **Hyper-V Manager**.
3. In the left column, select the server that is hosting the Control Center master host, and then right-click to select **New** > **Hard Disk...**.
4. In the **New Virtual Hard Disk Wizard** dialog, navigate to the **Choose Disk Format** panel.
5. Click the **VHDX** radio button, and then click the **Next** button.
6. In the **Choose Disk Type** panel, click the **Dynamically expanding** radio button, and then click the **Next** button.
7. In the **Specify Name and Location** panel, enter a name for the disk in the **Name** field, and then click the **Next** button.
8. In the **Configure Disk** panel, click the **Create a new blank virtual hard disk** radio button, enter the disk size in the **Size** field, and then click the **Next** button.

   For more information, refer to the *Zenoss Resource Manager Planning Guide*.
9. In the **Summary** panel, review the virtual disk settings, and then click the **Finish** button.
10. In **Hyper-V Manager**, right-click the virtual machine of the Control Center master host, and then select **Settings...**.
11. In the **Settings** dialog, select **SCSI Controller** from the **Hardware** list in the left column.
12. In the **SCSI Controller** area on the right side, select **Hard Drive**, and then click the **Add** button.
13. In the **Hard Drive** area, click the **Virtual hard disk** radio button, and then click the **Browse** button.
14. In the **Open** dialog, select the hard disk image created previously, and then click the **Open** button.
15. In the **Settings** dialog, click the **OK** button.

Identifying new virtual disks

This procedure identifies the newly-attached virtual disks of an appliance-based master host.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.
   a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b. Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

3. Rescan all SCSI storage.

```
for h in $(ls /sys/class/scsi_host)
do
  echo "- - -" > /sys/class/scsi_host/${h}/scan
done
```

4. Identify the virtual disks attached to the host.

```
lsblk -pdo NAME,HCTL,SIZE
```

   Example output:

```
NAME     HCTL        SIZE
/dev/sda 2:0:0:0     293G
/dev/sdb 2:0:1:0     300G
/dev/sr0 1:0:0:0    1024M
```

   Compared to the previous example output, this example output shows a new drive, `/dev/sdb`.

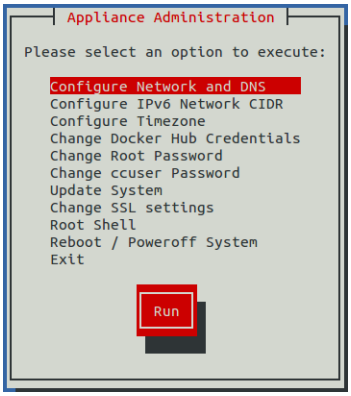
Creating primary partitions

To perform this procedure, you need a host with at least one disk device.

This procedure demonstrates how to create primary partitions on a disk. Each primary partition may be formatted as a file system or swap space, used in a device mapper thin pool, or reserved for future use. Each disk must have one primary partition, and may have four.

Note Data present on the disk you select is destroyed by this procedure. Please ensure that data present on the disk is backed up elsewhere, or no longer needed, before proceeding.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Log in as the `root` user.



3 Start a command-line session as `root`.

   a In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.

   b Press the **Tab** key to select **Run**, and then press the **Return** key.

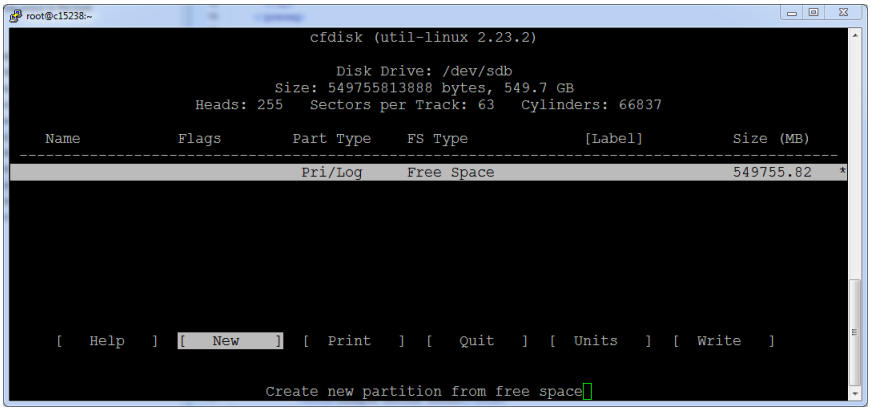
   The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

4 Start the partition table editor for the target disk.
In this example, the taget disk is `/dev/sdb`, and it has no entries in its partition table.

```
cfdisk /dev/sdb
```

Figure 11: Initial screen



The `cfdisk` command provides a text user interface (TUI) for editing the partition table. The following list describes how to navigate through the interface:

- To select an entry in the table, use the up and down arrow keys. The current entry is highlighted.
- To select a command from the menu at the bottom of the interface, use the left and right arrow keys, or **Tab** and **Shift-Tab**. The current command is highlighted.
- To choose a command, press the **Enter** key.
- To return to the previous level of the menu, press the **Esc** key.
- To exit the interface, select **Quit** from the menu, and then press the **Enter** key.

For more information about `cfdisk`, enter `man cfdisk`.

5 Create a new partition.

Repeat the following substeps for each primary partition to create. You may create four primary partitions on a disk.

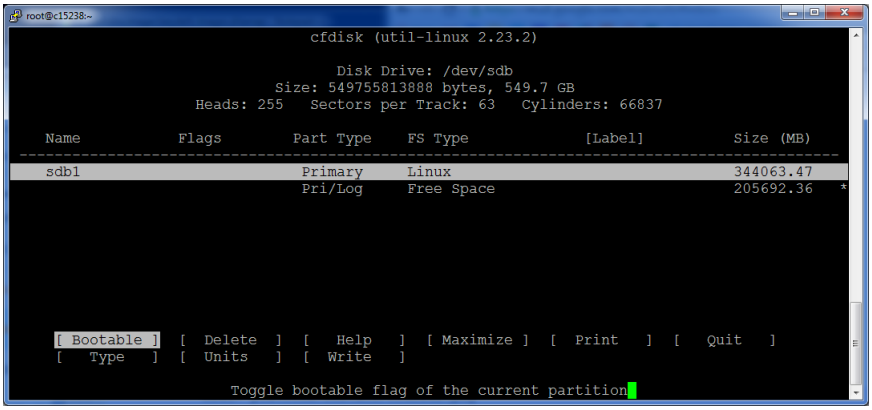
a Select the table entry with the value **Free Space** in the **FS Type** column.
b Select **[New]**, and then press the **Enter** key.
c Select **[Primary]**, and then press the **Enter** key.
d At the **Size (in MB)** prompt, enter the size of the partition to create in megabytes, and then press the **Enter** key.

   To accept the default value, which is all of the free space on the disk, just press the **Enter** key.

e **Note** If you created a single partition that uses all of the available disk space, skip this substep.

   Optional: Select **[Beginning]**, and then press the **Enter** key.

Figure 12: One primary partition



6 Write the partition table to disk, and then exit the partition table editor.
a Select **[Write]**, and then press the **Enter** key.
b At the **Are you sure...** prompt, enter `yes`, and then press the **Enter** key.

   You can ignore the warning about a bootable partition.

c Select **[Quit]**, and then press the **Enter** key.

Preparing a partition for backups

To perform this procedure, you need an unused primary partition.

This procedure prepares a partition for backups for a Control Center master host.

1 Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2 Start a command-line session as `root`.
a In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
b Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

3 Identify the partition to prepare.
Replace *Device* with the virtual disk added previously:

```
lsblk -p --output=NAME,SIZE,TYPE Device
```

Example output:

```
NAME            SIZE TYPE
/dev/sdb        300G disk
```

```
 |--/dev/sdb1  300G part
```

In this example, the partition to prepare is `/dev/sdb1`.

4 Create an XFS file system on the partition, and label the partition.
Replace *Partition* with the partition identified previously:

```
mkfs -t xfs -L BACKUPS Partition
```

5 Create an entry in the `/etc/fstab` file.
Replace *Partition* with the partition identified previously:

```
myPart=Partition
echo "$myPart /opt/serviced/var/backups xfs defaults 0 0" \
  >> /etc/fstab
```

6 Mount the file system, and then verify it mounted correctly.

```
mount -a && mount | grep backups
```

Example result:

```
/dev/sdb1 on /opt/serviced/var/backups type xfs
 (rw,relatime,attr2,inode64,noquota)
```

3 Installing resource pool hosts

This chapter describes how to install appliance-based resource pool hosts. You may add as many resource pool hosts as you wish to a Control Center cluster.

Creating a virtual machine

You may create a virtual machine for the Resource Manager appliance with VMware vSphere or Microsoft Hyper-V. Choose one of the procedures in this section.

Creating a virtual machine with vSphere

To perform this task, you need a VMware vSphere client.

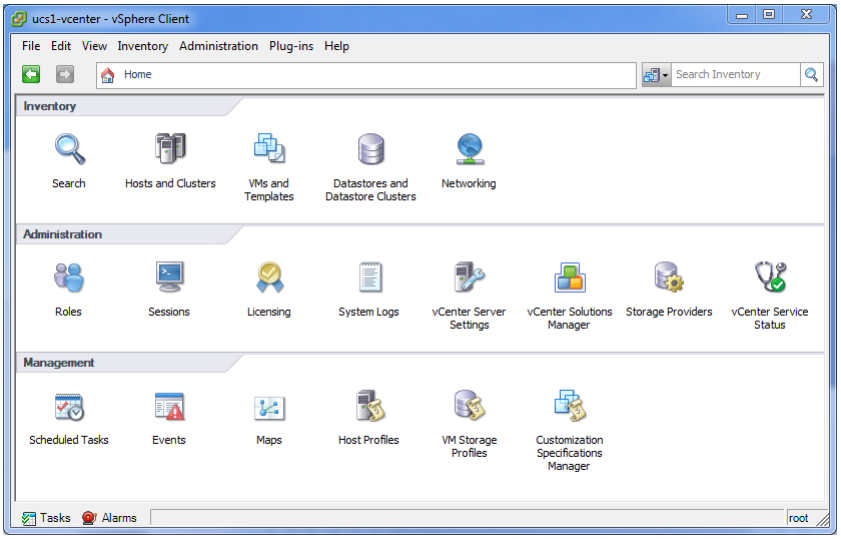
This procedure installs the Resource Manager OVA package as a virtual machine managed by vSphere Server version 5.0.0, using VMware vSphere Client 5.0.0. The procedure is slightly different with different versions of VMware vSphere Client.

1 Download the Resource Manager OVA file from the *Zenoss Support* site to your workstation, if necessary.

> **Note** The same OVA package is used for both master host and resource pool host virtual machines.

2 Use the VMware vSphere Client to log in to vCenter as `root`, or as a user with superuser privileges, and then display the **Home** view.

Figure 13: vSphere client Home view



3 From the **File** menu, select **Deploy OVF Template...**.
4 In the **Source** panel, specify the path of the Resource Manager package, and then click **Next**.
5 In the **OVF Template Details** panel, click **Next**.
6 In the **Name and Location** panel, provide a name and a location for the server.
   a In the **Name** field, enter a new name.
   b In the **Inventory Location** area, select a data center for the virtual machine.
   c Click **Next**.
7 In the **Host / Cluster** panel, select a host system, and then click **Next**.
8 In the **Storage** panel, select a storage system with sufficient space for the virtual machine, and then click **Next**.
9 In the **Disk Format** panel, select select **Thin Provison**, and then click **Next**.
10 In the **Ready to Complete** panel, review the deployment settings, and then click **Finish**.

   Please do not check the check box labeled **Power on after deployment**.
11 Navigate to the new virtual machine's **Getting Started** or **Summary** tab, and then click the **Edit virtual machine settings** link.
12 Update the memory assigned to the machine.
   a In the **Virtual Machine Properties** dialog, select **Memory** in the **Hardware** table.
   b In the **Memory Configuration** area, set the **Memory Size** field to 32GB.
   c At the bottom of the the **Virtual Machine Properties** dialog, click the **OK** button.
13 On the new virtual machine's **Getting Started** tab, click the **Power on virtual machine** link.

Creating a virtual machine with Hyper-V

To perform this task, you need:

- A Microsoft Remote Desktop client
- Administrator privileges on a Microsoft Hyper-V server

This procedure installs the Resource Manager appliance as a virtual machine managed by Microsoft Hyper-V.

1 Use a Microsoft Remote Desktop client to log in to a Hyper-V host as Administrator, or as a user with Administrator privileges.
2 Download the Resource Manager ISO file from the *Zenoss Support* site to the Hyper-V host, if necessary.

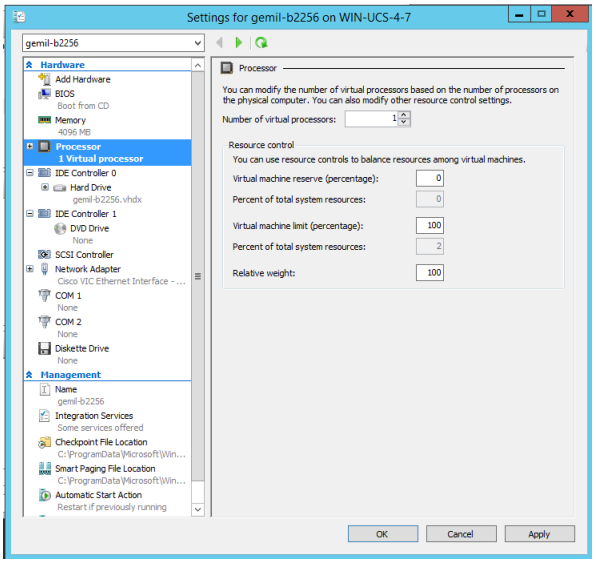
> **Note** The same OVA package is used for both master host and resource pool host virtual machines.

3 Start **Hyper-V Manager**.
4 In the left column, select a server to host the virtual machine.
5 From the **Action** menu, select **New** > **Virtual Machine...**.
6 In the **New Virtual Machine Wizard** dialog, display the **Specify Name and Location** panel.

   If the first panel displayed is the **Before You Begin** panel, click **Next**.
7 In the **Specify Name and Location** panel, provide a name for the virtual machine, and then click **Next**.
8 In the **Specify Generation** panel, select **Generation 1**, and then click **Next**.
9 In the **Assign Memory** panel, enter 32768 (32GB) in the **Startup memory** field, and then click **Next**.
10 In the **Configure Networking** panel, select **Cisco VIC Ethernet Interface - Virtual Switch**, and then click **Next**.
11 In the **Connect Virtual Hard Disk** panel, select **Create a virtual hard disk**, enter 200 in the **Size** field, and then click **Next**.
12 In the **Installation Options** panel, specify the Resource Manager ISO package.
   a Select **Install an operating system from a bootable CD/DVD-ROM**.
   b Select **Image file (.iso)**, and then specify the location of the Resource Manager ISO image file.
   c Click **Next**.
13 In the **Summary** panel, review the virtual machine specification, and then click **Finish**.

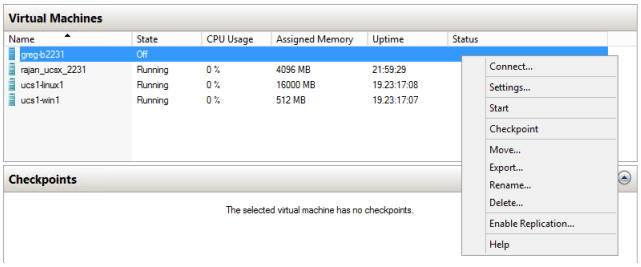
   Hyper-V Manager creates the new virtual machine, and then closes the **New Virtual Machine Wizard** dialog.
14 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Settings...**.
15 In the **Hardware** area of the **Settings** dialog, select **Processor**.

Figure 14: Settings dialog, Processor selected



16 In the **Processor** area, enter 8 in the **Number of virtual processors** field, and then click **OK**.
17 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Start**.

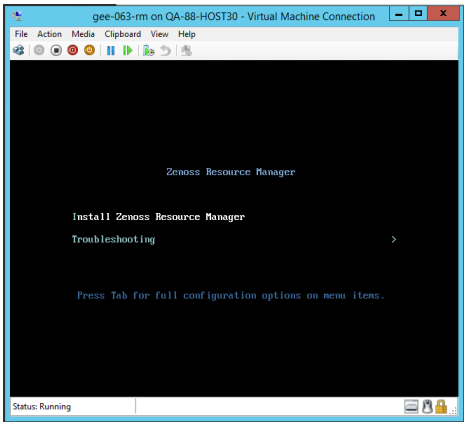
Figure 15: Starting a virtual machine



18 In the **Virtual Machines** area of Hyper-V Manager, select the new virtual machine, and then right-click to select **Connect**.

19 In the **Virtual Machine Connection** window, press the **Enter** key.

Figure 16: Appliance installation start screen

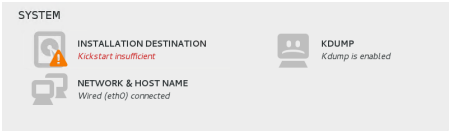


The appliance installation process takes about 15 minutes, and should complete with no additional input.

20 Optional: Select the installation destination, if necessary.

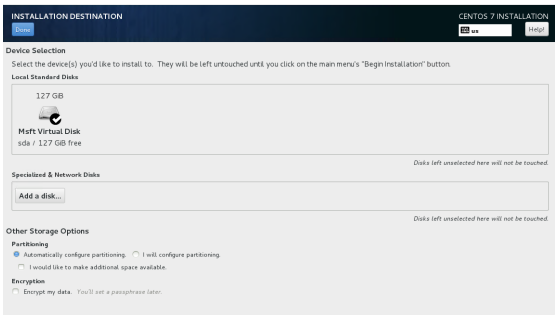
Occasionally, installation is interrupted with the `Kickstart insufficient` message.

Figure 17: Kickstart insufficient message



a In the **SYSTEM** area of the **INSTALLATION SUMMARY** page, click the **INSTALLATION DESTINATION** control.

Figure 18: INSTALLATION DESTINATION page



b On the **INSTALLATION DESTINATION** page, click the **Done** button, located at the upper-left corner of the page.

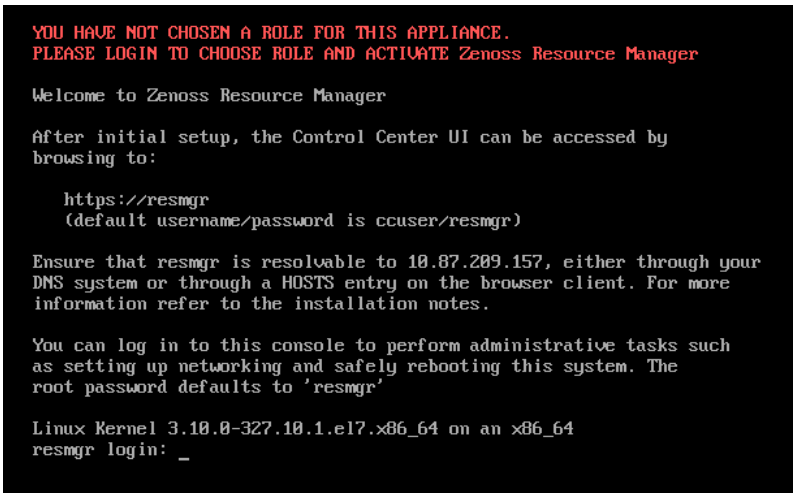
c On the **INSTALLATION SUMMARY** page, click the Begin Installation button, located at the bottom-right corner of the page.

Configuring the virtual machine mode

This procedure configures the new virtual machine as a resource pool host.

1 Gain access to the console interface of the Control Center host through your hypervisor console interface.

Figure 19: Initial hypervisor console login prompt



2 Log in as the `root` user.

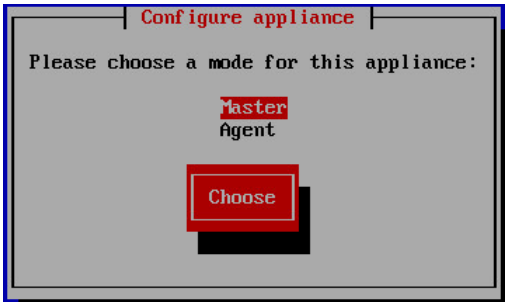
The initial password is provided in the console.

3 The system prompts you to enter a new password for `root`.

4 The system prompts you to enter a new password for `ccuser`.

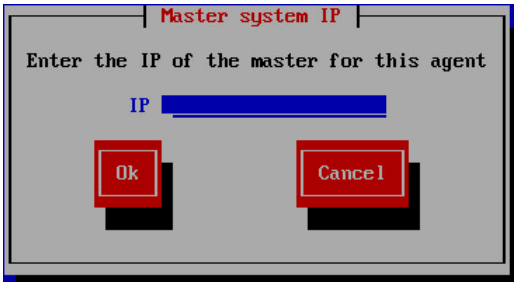
The `ccuser` acount is the default account for gaining access to the Control Center browser interface.

5 Select the **Agent** role for the virtual machine.



a In the **Configure appliance** menu, press the down-arrow key to select **Agent**.

b Press the the **Tab** key to select the **Choose** button, and then the **Enter** key.



c In the **IP** field, enter the hostname, fully-qualified domain name, or IPv4 address of the master host.

If you enter the hostname or fully-qualified domain name of the master host, you need an entry in the `/etc/hosts` file of the agent host, or a nameserver on your network, that resolves the name to its IPv4 address.

d Press the the **Tab** key to select the **Ok** button, and then the **Enter** key.

The system reboots.

Edit a connection

The default configuration for network connections is DHCP. To configure static IPv4 addressing, perform this procedure.

1 Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*.

2 Log in as the `root` user.

3 Select the **NetworkManager TUI** menu.

a In the **Appliance Administration** menu, select the **Configure Network and DNS** option.

b Press the **Tab** key to select the **Run** button.

c Press the **Enter** key.

4 On the **NetworkManager TUI** menu, select **Edit a connection**, and then press the **Return** key.
The TUI displays the connections that are available on this host.

Figure 20: Example: Available connections

Note Do not modify the `docker0` connection.

5 Use the down-arrow key to select the virtual connection, and then press the **Return** key.

Figure 21: Example: Edit Connection screen

Use the **Tab** key and the arrow keys to navigate among options in the **Edit Connection** screen, and use the **Return** key to toggle an option or to display a menu of options.

6 Optional: If the **IPv4 CONFIGURATION** area is not visible, select its display option (**<Show>**), and then press the **Return** key.

7 In the **IPv4 CONFIGURATION** area, select **<Automatic>**, and then press the **Return** key.

Figure 22: Example: IPv4 Configuration options

8 Configure static IPv4 networking.

   a. Use the down arrow key to select **Manual**, and then press the **Return** key.
   b. Use the **Tab** key or the down arrow key to select the **<Add...>** option next to **Addresses**, and then press the **Return** key.
   c. In the **Addresses** field, enter an IPv4 address for the virtual machine, and then press the **Return** key.
   d. Repeat the preceding two steps for the **Gateway** and **DNS servers** fields.
9. Use the **Tab** key or the down arrow key to select the **<OK>** option at the bottom of the **Edit Connection** screen, and then press the **Return** key.
10. In the available connections screen, use the **Tab** key to select the **<Quit>** option, and then press the **Return** key.
11. Reboot the operating system.
   a. In the **Appliance Administration** menu, use the down-arrow key to select the **Reboot / Poweroff System** option.
   b. Use the **Down Arrow** key to select **Reboot**.
   c. Press the **Tab** key to select **OK**, and then press the **Return** key.

Set system hostname

The default hostname of a Resource Manager appliance host is `resmgr`. To change the hostname, perform this procedure.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Select the **NetworkManager TUI** menu.
   a. In the **Appliance Administration** menu, select the **Configure Network and DNS** option.
   b. Press the **Tab** key to select the **Run** button.
   c. Press the **Enter** key.
3. Display the hostname entry field.
   a. In the **NetworkManager TUI** menu, use the down-arrow key to select **Set system hostname**.
   b. Press the **Tab** key to select the **OK** button.
   c. Press the **Enter** key.
4. In the **Hostname** field, enter the new hostname.

   You may enter either a hostname or a fully-qualified domain name.
5. Press the **Tab** key twice to select the **OK** button, and then press the **Enter** key.
6. In the confirmation dialog, press the **Return** key.
7. Reboot the operating system.
   a. In the **Appliance Administration** menu, use the down-arrow key to select the **Reboot / Poweroff System** option.
   b. Use the **Down Arrow** key to select **Reboot**.
   c. Press the **Tab** key to select **OK**, and then press the **Return** key.

Editing the /etc/hosts file

This procedure is optional. Perform this procedure only if you use hostnames or fully-qualified domain names instead of IPv4 addresses, and only after all resource pool hosts are installed and renamed. Perform this procedure on the Control Center master host and on each resource pool host.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.

a In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.

b Press the **Tab** key to select **Run**, and then press the **Return** key.

The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

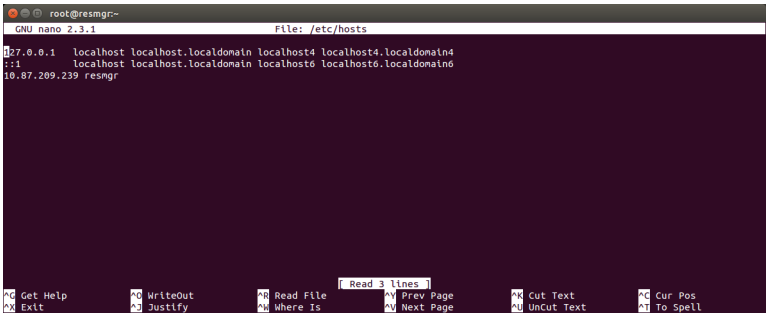
3 Open the `/etc/hosts` file in a text editor.

The following steps use the *nano* editor.

a Start the editor.

```
nano /etc/hosts
```

Figure 23: Example nano session



Use the up-arrow and down-arrow keys to select lines, and the right-arrow and left-arrow keys to select characters on a line.

b Optional: On resource pool hosts, the file may include two entries with the same the IP address. Remove the first of the two entries, which maps the IP address to the `resmgr` hostname.

c Add entries for the Control Center master host and for each resource pool host.

d Save the file and exit the editor.

To save, press **Control-o**. To exit, press **Control-x**.

4 Return to the **Appliance Administration** menu.

```
exit
```

5 Exit the **Appliance Administration** menu.

a Use the down-arrow key to select **Exit**.

b Press the **Tab** key, and then press the **Return** key.

Configuring a multi-host Control Center cluster 4

This chapter describes how to configure an appliance-based multi-host Control Center cluster. A multi-host Resource Manager deployment includes one Control Center master host and one or more resource pool hosts. Zenoss recommends deploying at least two resource pool hosts, to enable creating a ZooKeeper ensemble.

Task	Procedure
Install a master host	*Installing a Control Center master host* on page 155
Add storage for backups	*Adding storage for backups* on page 163
Install resource pool hosts	*Installing resource pool hosts* on page 170
Create a ZooKeeper ensemble	*ZooKeeper ensemble configuration* on page 178
Create a private NTP server (Hyper-V only)	*Enabling NTP on Microsoft Hyper-V guests* on page 182
Add resource pool hosts to the cluster	*Adding hosts to the default resource pool* on page 185

ZooKeeper ensemble configuration

Control Center relies on *Apache ZooKeeper* to coordinate its services. The procedures in this section create a ZooKeeper ensemble of 3 nodes. To perform these procedures, you need a Control Center master host and a minimum of two resource pool hosts. Each resource pool host should have a static IP address.

Note Zenoss strongly recommends configuring a ZooKeeper ensemble for all production deployments.

A ZooKeeper ensemble requires a minimum of 3 nodes, and 3 nodes is sufficient for most deployments. A 5-node configuration improves failover protection during maintenance windows. Ensembles larger than 5 nodes are not necessary. An odd number of nodes is recommended, and an even number of nodes is strongly discouraged.

Configuring the master host as a ZooKeeper node

This procedure configures the Control Center master host as a member of the ZooKeeper ensemble.

Note For accuracy, this procedure constructs Control Center configuration variables in the shell and appends them to `/etc/default/serviced`. The last step is to move the variables from the end of the file to more appropriate locations.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.
   a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b. Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

   ```
   [root@resmgr ~]#
   ```
3. Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

   Note Define the variables identically on the master host and on each resource pool host.

   Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

   ```
   node1=Master-Host-IP
   node2=Pool-Host-A-IP
   node3=Pool-Host-B-IP
   ```

   Note ZooKeeper requires IP addresses for ensemble configuration.
4. Set the ZooKeeper node ID to `1`.

   ```
   echo "SERVICED_ISVCS_ZOOKEEPER_ID=1" >> /etc/default/serviced
   ```
5. Specify the nodes in the ZooKeeper ensemble.
   You may copy the following text and paste it in your console:

   ```
   echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
     >> /etc/default/serviced
   ```
6. Specify the nodes in the ZooKeeper quorum.

   ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

   You may copy the following of text and paste it in your console:

   ```
   q1="1@0.0.0.0:2888:3888"
   q2="2@${node2}:2888:3888"
   q3="3@${node3}:2888:3888"
   echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
     >> /etc/default/serviced
   ```
7. Clean up the Control Center configuration file.
   a. Open `/etc/default/serviced` with a text editor.
   b. Locate the *SERVICED_ZK* variable near the beginning of the file, and then delete the line it is on.

      The value of this declaration is just the master host.
   c. Save the file, and then close the text editor.
8. Verify the ZooKeeper environment variables.

   ```
   egrep '^[^#]*SERVICED' /etc/default/serviced | egrep '(_ZOO|_ZK)'
   ```

Configuring a resource pool host as a ZooKeeper node

This procedure configures a ZooKeeper ensemble on a resource pool host. Repeat this procedure on each Control Center resource pool host to add to the ZooKeeper ensemble.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.
   a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.
   b. Press the **Tab** key to select **Run**, and then press the **Return** key.

   The menu is replaced by a command prompt similar to the following example:

   ```
   [root@resmgr ~]#
   ```
3. Create a variable for each Control Center host to include in the ZooKeeper ensemble.

   The variables are used in subsequent steps.

   Note Define the variables identically on the master host and on each resource pool host.

   Replace *Master-Host-IP* with the IP address of the Control Center master host, and replace *Pool-Host-A-IP* and *Pool-Host-B-IP* with the IP addresses of the Control Center resource pool hosts to include in the ensemble:

   ```
   node1=Master-Host-IP
   node2=Pool-Host-A-IP
   node3=Pool-Host-B-IP
   ```

   Note ZooKeeper requires IP addresses for ensemble configuration.
4. Set the ID of this node in the ZooKeeper ensemble.

   For *Pool-Host-A-IP* (**`node2`**), use the following command:

   ```
   echo "SERVICED_ISVCS_ZOOKEEPER_ID=2" >> /etc/default/serviced
   ```

   For *Pool-Host-B-IP* (**`node3`**), use the following command:

   ```
   echo "SERVICED_ISVCS_ZOOKEEPER_ID=3" >> /etc/default/serviced
   ```
5. Specify the nodes in the ZooKeeper ensemble.
   You may copy the following text and paste it in your console:

   ```
   echo "SERVICED_ZK=${node1}:2181,${node2}:2181,${node3}:2181" \
     >> /etc/default/serviced
   ```
6. Specify the nodes in the ZooKeeper quorum.

   ZooKeeper requires a unique quorum definition for each node in its ensemble. To achieve this, replace the IP address of the current node with `0.0.0.0`.

   For *Pool-Host-A-IP* (**`node2`**), use the following commands:

   ```
   q1="1@${node1}:2888:3888"
   q2="2@0.0.0.0:2888:3888"
   q3="3@${node3}:2888:3888"
   echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
   ```

```
  >> /etc/default/serviced
```

For *Pool-Host-B-IP* (**node3**), use the following commands:

```
q1="1@${node1}:2888:3888"
q2="2@${node2}:2888:3888"
q3="3@0.0.0.0:2888:3888"
echo "SERVICED_ISVCS_ZOOKEEPER_QUORUM=${q1},${q2},${q3}" \
  >> /etc/default/serviced
```

7 Set the *SERVICED_ISVCS_START* variable, and clean up the Control Center configuration file.
   a Open `/etc/default/serviced` with a text editor.
   b Add the *SERVICED_ISVCS_START* variable.
   The value of the variable is `zookeeper`:

   ```
   SERVICED_ISVCS_START=zookeeper
   ```

   c Save the file, and then close the text editor.
8 Verify the ZooKeeper environment variables.

```
egrep '^[^#]*SERVICED' /etc/default/serviced \
  | egrep '(_ZOO|_ZK|_STA)'
```

9 Pull the required Control Center ZooKeeper image from the master host.
   a Identify the image to pull.

   ```
   serviced version | grep IsvcsImages
   ```

   Example result:

   ```
   IsvcsImages: [zenoss/serviced-isvcs:v40 zenoss/isvcs-zookeeper:v3]
   ```

   b Pull the Control Center ZooKeeper image.

   Replace *Isvcs-ZK-Image* with the name and version number of the ZooKeeper image from the previous substep:

   ```
   docker pull Isvcs-ZK-Image
   ```

Starting a ZooKeeper ensemble

This procedure starts a ZooKeeper ensemble.

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

1 Start a command-line session as `root` on the Control Center master host.
2 In a separate window, start a command-line session as `root` on the second node of the ZooKeeper ensemble (*Pool-Host-A-IP*).
3 In another separate window, start a command-line session as `root` on the third node of the ZooKeeper ensemble (*Pool-Host-B-IP*).
4 On all ensemble hosts, stop and start `serviced`.

```
systemctl stop serviced && systemctl start serviced
```

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

```
{ echo stats; sleep 1; } | nc localhost 2181 | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-A-IP 2181  | grep Mode
{ echo stats; sleep 1; } | nc Pool-Host-B-IP 2181  | grep Mode
```

If `nc` is not available, you can use `telnet` with *interactive ZooKeeper commands*.

Updating resource pool hosts

The default configuration of resource pool hosts sets the value of the *SERVICED_ZK* variable to the master host only. This procedure updates the setting to include the full ZooKeeper ensemble.

Perform this procedure on each resource pool host in your Control Center cluster.

1 Log in to the resource pool host as `root`, or as a user with superuser privileges.
2 Update the variable.
   a Open `/etc/default/serviced` in a text editor.
   b Locate the *SERVICED_ZK* declaration, and then replace its value with the same value used in the ZooKeeper ensemble nodes.
   c Save the file, and then close the editor.
3 Restart Control Center.

```
systemctl restart serviced
```

Enabling NTP on Microsoft Hyper-V guests

Like most distributed applications, Control Center requires a common time source. The procedures in this section enable *NTP* to synchronize the system clocks of all hosts in your Control Center cluster.

You may configure NTP to rely on public time servers or on a private master server.

- If all of the hosts in your Control Center cluster can access the internet, configure NTP to rely on public time servers.
- If none of the hosts in your Control Center cluster can access the internet, configure NTP to rely on a private master server.

Note The procedures in this section are required only for multi-host deployments running as Microsoft Hyper-V guests. VMware vSphere guests use an hourly `cron` job to synchronize their system clocks with the host.

Configuring NTP for public time servers

This procedure uses the default configuration of NTP to synchronize system clocks with public time servers. If all of the hosts in your Control Center cluster can access the internet, repeat this procedure on each host in the cluster, starting with the Control Center master host.

Note Do not perform this procedure on VMware vSphere guests.

1 Stop Control Center.

```
systemctl stop serviced
```

2 Synchronize the system clock and enable the NTP daemon.

a Set the system time.

```
ntpd -gq
```

b Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

c Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

d Start `ntpd`.

```
systemctl start ntpd
```

3 Start Control Center.

```
systemctl start serviced
```

Configuring an NTP master server

This procedure configures an NTP master server on the Control Center master host. Perform this procedure only if the host does not have internet access.

Note Do not perform this procedure on VMware vSphere guests.

1 Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

2 Edit the NTP configuration file./

a Open `/etc/ntp.conf` with a text editor.

b Replace all of the lines in the file with the following lines:

```
# Use the local clock
server 127.127.1.0 prefer
fudge  127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift
broadcastdelay 0.008

# Give localhost full access rights
restrict 127.0.0.1

# Grant access to client hosts
restrict ADDRESS_RANGE mask NETMASK nomodify notrap
```

c Replace `ADDRESS_RANGE` with the range of IPv4 network addresses that are allowed to query this NTP server.

For example, the following IP addresses are assigned to the hosts in an Control Center cluster:

```
203.0.113.10
```

```
203.0.113.11
203.0.113.12
203.0.113.13
```

For the preceding addresses, the value for `ADDRESS_RANGE` is `203.0.113.0`.

d Replace `NETMASK` with the IPv4 network mask that corresponds with the address range. For example, the network mask for `203.0.113.0` is `255.255.255.0`.

e Save the file and exit the editor.

3 Stop Control Center.

```
systemctl stop serviced
```

4 Enable and start the NTP daemon.

a Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

b Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

c Start `ntpd`.

```
systemctl start ntpd
```

5 Start Control Center.

```
systemctl start serviced
```

Configuring NTP clients

This procedure configures agent hosts to synchronize their clocks with the NTP server on the Control Center master host. Perform this procedure only if the hosts do not have internet access, and repeat this procedure on each resource pool host in your Control Center cluster.

Note Do not perform this procedure on VMware vSphere guests.

1 Create a backup of the NTP configuration file.

```
cp -p /etc/ntp.conf /etc/ntp.conf.orig
```

2 Edit the NTP configuration file./

a Open `/etc/ntp.conf` with a text editor.

b Replace all of the lines in the file with the following lines:

```
# Point to the master time server
server MASTER_ADDRESS

restrict default ignore
restrict 127.0.0.1
restrict MASTER_ADDRESS mask 255.255.255.255 nomodify notrap noquery
```

```
driftfile /var/lib/ntp/drift
```

c. Replace both instances of `MASTER_ADDRESS` with the IPv4 address of the host where the NTP server is running (the Control Center master host).

d. Save the file and exit the editor.

3. Stop Control Center.

```
systemctl stop serviced
```

4. Synchronize the clock with the master server.

```
ntpd -gq
```

5. Enable and start the NTP daemon.

a. Enable the `ntpd` daemon.

```
systemctl enable ntpd
```

b. Configure `ntpd` to start when the system starts.

Currently, an unresolved issue associated with NTP prevents `ntpd` from restarting correctly after a reboot, and the following commands provide a workaround to ensure that it does.

```
echo "systemctl start ntpd" >> /etc/rc.d/rc.local
chmod +x /etc/rc.d/rc.local
```

c. Start `ntpd`.

```
systemctl start ntpd
```

6. Start Control Center.

```
systemctl start serviced
```

Adding hosts to the default resource pool

This procedure adds one or more resource pool hosts to the `default` resource pool.

1. Gain access to the Control Center host, through the console interface of your hypervisor, or through a remote shell utility such as *PuTTY*, and then log in as `root`.
2. Start a command-line session as `root`.

a. In the **Appliance Administration** menu, use the down-arrow key to select **Root Shell**.

b. Press the **Tab** key to select **Run**, and then press the **Return** key.

The menu is replaced by a command prompt similar to the following example:

```
[root@resmgr ~]#
```

3. Add a resource pool host.

Replace *Hostname-Or-IP* with the hostname or IP address of the resource pool host to add:

```
serviced host add Hostname-Or-IP:4979 default
```

If you enter a hostname, all hosts in your Control Center cluster must be able to resolve the name, either through an entry in `/etc/hosts`, or through a nameserver on your network.

4. Repeat the preceding command for each resource pool host in your Control Center cluster.