



Zenoss Core Upgrade Guide

Release 5.0.10

Zenoss, Inc.

www.zenoss.com

Zenoss Core Upgrade Guide

Copyright © 2016 Zenoss, Inc. All rights reserved.

Zenoss and the Zenoss logo are trademarks or registered trademarks of Zenoss, Inc., in the United States and other countries. All other trademarks, logos, and service marks are the property of Zenoss or other third parties. Use of these marks is prohibited without the express written consent of Zenoss, Inc., or the third-party owner.

Flash is a registered trademark of Adobe Systems Incorporated.

Oracle, the Oracle logo, Java, and MySQL are registered trademarks of the Oracle Corporation and/or its affiliates.

Linux is a registered trademark of Linus Torvalds.

RabbitMQ is a trademark of VMware, Inc.

SNMP Informant is a trademark of Garth K. Williams (Informant Systems, Inc.).

Sybase is a registered trademark of Sybase, Inc.

Tomcat is a trademark of the Apache Software Foundation.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

All other companies and products mentioned are trademarks and property of their respective owners.

Part Number: 1091.16.050

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

Contents

Preface	4
Chapter 1: Supported software and upgrade paths	5
Supported combinations.....	5
Supported upgrade paths.....	5
Chapter 2: Upgrading to 1.0.10 / 5.0.10	7
Upgrade overview.....	7
Upgrading Control Center.....	8
Upgrading Zenoss Core.....	12
Overview of converting the Docker storage driver.....	12
Chapter 3: Upgrading to 1.0.3 / 5.0.3	18
Upgrade script.....	18
Upgrading Control Center.....	19
Upgrading Zenoss Core.....	20

Preface

Zenoss Core Upgrade Guide provides detailed instructions for upgrading Zenoss Core from one minor or micro version to a more recent version.

Related publications

Title	Description
<i>Zenoss Core Administration Guide</i>	Provides an overview of Zenoss Core architecture and features, as well as procedures and examples to help use the system.
<i>Zenoss Core Installation Guide</i>	Provides detailed information and procedures for installing and upgrading Zenoss Core.
<i>Zenoss Core Release Notes</i>	Describes known issues, fixed issues, and late-breaking information not already provided in the published documentation set.

Additional information and comments

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.

1

Supported software and upgrade paths

Beginning with version 5.0.0, distributions of Zenoss Core include an additional component, Control Center. Each component is developed and maintained separately, and each has its own version number. This chapter identifies the combinations of component versions that Zenoss supports, and the supported upgrade paths between the combinations.

Supported combinations

The following table shows the Control Center and Zenoss Core release dates and the corresponding version combination for that release:

Release Date	Control Center	Zenoss Core
19 Feb 2016	1.0.10	5.0.10
02 Dec 2015	1.0.9	5.0.9
16 Nov 2015	1.0.8	5.0.8
10 Oct 2015	1.0.7	5.0.7
14 Sep 2015	1.0.6	5.0.6
05 Aug 2015	1.0.5	5.0.5
10 Jul 2015	1.0.4	5.0.4
27 May 2015	1.0.3	5.0.3
20 Apr 2015	1.0.2	5.0.2
03 Apr 2015	1.0.1	5.0.1
24 Feb 2015	1.0.0	5.0.0

Supported upgrade paths

Upgrade to 1.0.10 / 5.0.10

From combination	To combination
Control Center 1.0.3 and	Control Center 1.0.10 and

From combination	To combination
Zenoss Core 5.0.3	Zenoss Core 5.0.10
Control Center 1.0.4 and Zenoss Core 5.0.4	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.5 and Zenoss Core 5.0.5	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.6 and Zenoss Core 5.0.6	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.7 and Zenoss Core 5.0.7	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.8 and Zenoss Core 5.0.8	Control Center 1.0.10 and Zenoss Core 5.0.10

Upgrade to 1.0.3 / 5.0.3

From combination	To combination
Control Center 1.0.0 and Zenoss Core 5.0.0	Control Center 1.0.3 and Zenoss Core 5.0.3
Control Center 1.0.1 and Zenoss Core 5.0.1	Control Center 1.0.3 and Zenoss Core 5.0.3
Control Center 1.0.2 and Zenoss Core 5.0.2	Control Center 1.0.3 and Zenoss Core 5.0.3

2

Upgrading to 1.0.10 / 5.0.10

This chapter describes how to upgrade the combinations of Control Center and Zenoss Core shown in the following table.

From combination	To combination
Control Center 1.0.3 and Zenoss Core 5.0.3	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.4 and Zenoss Core 5.0.4	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.5 and Zenoss Core 5.0.5	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.6 and Zenoss Core 5.0.6	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.7 and Zenoss Core 5.0.7	Control Center 1.0.10 and Zenoss Core 5.0.10
Control Center 1.0.8 and Zenoss Core 5.0.8	Control Center 1.0.10 and Zenoss Core 5.0.10

For information about upgrading other combinations, see [Supported upgrade paths](#) on page 5.

Upgrade overview

The upgrade process include two required phases and one optional phase:

- 1 Upgrade Control Center. This phase includes stopping all of the applications that Control Center manages, upgrading Control Center on the master host, and then upgrading all of the resource pool hosts in your Control Center cluster.
- 2 Upgrade Zenoss Core. This phase is only performed on the Control Center master host.
- 3 Convert the storage driver that Docker uses from `btrfs` to `devicemapper`. The conversion is needed on all of the hosts in your Control Center cluster, and includes creating a new file system for the new storage driver. If you converted the storage driver to `devicemapper` during a previous upgrade, you may skip this phase.

The amount of time required to complete the upgrade process depends largely on environmental factors such as the size of your cluster. You can reduce the amount of time that Zenoss Core is down by downloading images from Docker Hub and reviewing the storage driver conversion process, in advance. For more information, see the following topics:

- [Downloading upgrade images](#) on page 8
- [Overview of converting the Docker storage driver](#) on page 12

Downloading upgrade images

By performing this procedure, you can reduce the amount of time required to upgrade Control Center and Zenoss Core by approximately 35-60 minutes.

- 1 Log in to the Control Center master host as `root` or as a user with `sudo` privileges.
- 2 Download the Docker image of Zenoss Core for this release.

The download takes approximately 10-20 minutes.

Ubuntu:

```
sudo docker run -it --rm -v /root:/mnt/root \
  zenoss/core_5.0:5.0.10 rsync -a /root/5.0.x /mnt/root
```

RHEL/CentOS:

```
docker run -it --rm -v /root:/mnt/root \
  zenoss/core_5.0:5.0.10 rsync -a /root/5.0.x /mnt/root
```

When the download completes, the `rsync` command copies scripts that perform the upgrade to a new directory, `/root/5.0.x`.

- 3 Download the Control Center internal services, HBase, and OpenTSDB images for this release.

Ubuntu: `sudo /root/5.0.x/pull-docker-images.sh`

RHEL/CentOS: `/root/5.0.x/pull-docker-images.sh`

The download takes approximately 25-50 minutes.

Upgrading Control Center

This section describes how to upgrade Control Center on all of the hosts in a Control Center cluster. The upgrade process includes three procedures:

- 1 Stop the application or applications that Control Center is managing, and then stop Control Center on the master host.
- 2 Upgrade Docker and the operating system, if necessary, on the master host.
- 3 Upgrade Control Center on the master host.
- 4 Upgrade Docker, the operating system (if necessary), and Control Center on all hosts that are configured as agents only.

Stopping applications and Control Center

Perform this procedure to stop all Control Center applications, and Control Center, on the master host.

- 1 Log in to the Control Center browser interface.

The screenshot shows the Control Center web interface. The top navigation bar includes 'Applications', 'Resource Pools', 'Hosts', 'Logs', and 'Backup / Restore'. The user is logged in as 'ecuser'. The main content area is divided into two sections: 'Applications' and 'Application Templates'.

Applications Table:

Application	Description	Status	Deployment ID	Resource Pool	Virtual Host Names	Actions
Internal Services	Internal Services	Running (blue checkmark)	Internal	N/A	N/A	N/A
Zenoss.core (v5.0)	Zenoss Core	Stopped (grey circle)	Test	default	https://zenoss5x.c70	Start Stop Delete

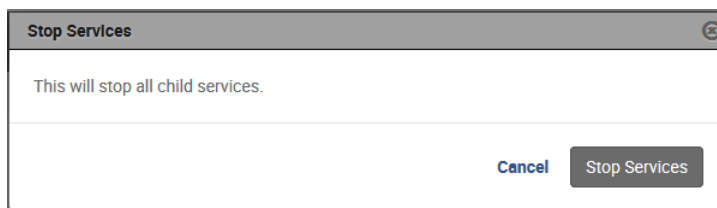
Application Templates Table:

Application Template	ID	Description	Actions
Zenoss.core (v5.0)	44441b24f9761b063d9f5ce6ec3d4eb7	Zenoss Core	Delete

- 2 Identify the application to stop.
For Zenoss Core, the application to stop is `Zenoss.core`.

Note The **Internal Services** application is part of Control Center, and is not stopped when Control Center is running.

- 3 In the **Actions** column of the **Applications** table, click the **Stop** control of each application to stop.
- 4 In the **Stop Services** dialog, click **Stop Services**.



- 5 Verify that the application's child services are stopped.
In the **Application** column of the **Applications** table, click the application name, and then scroll down to the **Services** table.
Stopped services have a grey circle icon in the **Status** column. Zenoss recommends waiting until all of the child services are stopped before proceeding to the next step.
- 6 Stop Control Center on the master host.
 - a Log in to the Control Center master host as `root` or as a user with `sudo` privileges.
 - b Stop Control Center.

```
Ubuntu: sudo stop serviced
RHEL/CentOS: systemctl stop serviced
```

Upgrading Docker and the host operating system

Perform this procedure to upgrade Docker from version 1.5 to 1.8.2 on a Control Center cluster host.

Note For this release, RHEL/CentOS systems must be upgraded from 7.0 to 7.1. So this procedure includes a step to upgrade the operating system, if necessary.

- 1 Log in to the Control Center master host as `root` or as a user with `sudo` privileges.
- 2 Remove the Docker 1.5 package, without removing Control Center.

RHEL/CentOS:

```
rpm -e --nodeps zenoss-docker-1.5.0-2
```

Ubuntu:

```
sudo dpkg -r --force-depends lxc-docker-1.5.0
```

- 3 On RHEL/CentOS systems, upgrade the operating system to version 7.1.

- a Update the repository cache.

```
yum makecache fast
```

- b Perform the upgrade.

```
yum update
```

- 4 Add the Docker repository to the host's repository list.

RHEL/CentOS:

```
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
```

Ubuntu:

```
myRepo='http://apt.dockerproject.org/repo/pool/main/d/docker-engine'
sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 \
  --recv-keys 58118E89F3A912897C070ADBF76221572C52609D
curl -sO $myRepo/docker-engine_1.8.2-0~trusty_amd64.deb
```

- 5 Update the repository cache.

RHEL/CentOS:

```
yum makecache fast
```

Ubuntu:

```
sudo apt-get update
```

- 6 Install Docker 1.8.2.

RHEL/CentOS:

```
yum install -y docker-engine-1.8.2
```

Ubuntu:

```
sudo dpkg -i docker-engine_1.8.2-0~trusty_amd64.deb
```

- 7 On RHEL/CentOS systems, edit the docker service definition.

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

- b Add the following line immediately after the line that contains [Service].

```
EnvironmentFile=-/etc/sysconfig/docker
```

- c Add `DOCKER_OPTS` to the `ExecStart` definition.
The result should look like the following example:

```
ExecStart=/usr/bin/docker daemon $DOCKER_OPTS -H fd://
```

8 Configure Docker.

- a Identify the IPv4 address and subnet Docker has selected for its virtual Ethernet bridge.

```
ip addr | grep -A 2 'docker0:' | grep inet
```

Note Typically, the address and subnet is 172.17.42.1/16. For more information about changing the selection, refer to Docker's [advanced network configuration](#) article.

- b Add the `devicemapper` and `DNS` flags to the Docker startup options.
If you change the virtual bridge subnet, replace the IP address in the following command.

```
echo 'DOCKER_OPTS="-s devicemapper --dns=172.17.42.1"' \  
>> /etc/sysconfig/docker
```

Upgrading Control Center on the master host

Perform this procedure to upgrade Control Center on the master host.

Before performing this procedure, complete all of the steps in [Stopping applications and Control Center](#) on page 8 and [Upgrading Control Center on resource pool hosts](#) on page 11.

- 1 Log in to the Control Center master host as `root` or as a user with `sudo` privileges.
- 2 Upgrade Control Center.

```
Ubuntu:sudo apt-get update && sudo apt-get install -y serviced=1.0.10  
RHEL/CentOS:yum --enablerepo=zenoss-stable install -y serviced-1.0.10
```

During installation, you may see a message about `/etc/default/serviced.rpmnew`. For this release, no changes have been made to `/etc/default/serviced`, so the new version may be ignored.

- 3 Start Control Center.

```
Ubuntu:sudo start serviced  
RHEL/CentOS:systemctl start serviced
```

Upgrading Control Center on resource pool hosts

Perform this procedure on each Control Center host that is configured as an agent only.

Before performing this procedure, complete all of the steps in [Stopping applications and Control Center](#) on page 8.

- 1 Log in to the Control Center resource pool host as `root` or as a user with `sudo` privileges.
- 2 Stop Control Center.

```
Ubuntu:sudo stop serviced  
RHEL/CentOS:systemctl stop serviced
```

- 3 Upgrade Docker.

For more information, see [Upgrading Docker and the host operating system](#) on page 9.

4 Upgrade Control Center.

```
Ubuntu:sudo apt-get update && sudo apt-get install -y serviced=1.0.10
RHEL/CentOS:yum --enablerepo=zenoss-stable install -y serviced-1.0.10
```

During installation, you may see a message about `/etc/default/serviced.rpmnew`. For this release, no changes have been made to `/etc/default/serviced`, so the new version may be ignored.

5 Start Control Center.

```
Ubuntu:sudo start serviced
RHEL/CentOS:systemctl start serviced
```

To minimize downtime, proceed immediately to [Upgrading Zenoss Core](#) on page 12.

Upgrading Zenoss Core

Perform this procedure to upgrade Zenoss Core.

Before performing this procedure, complete all of the procedures in [Upgrading Control Center](#) on page 8.

- 1 Log in to the Control Center master host as `root` or as a user with `sudo` privileges.
- 2 Optional: Download the Docker image of Zenoss Core for this release, if necessary.

The download takes approximately 10-20 minutes.

Ubuntu:

```
sudo docker run -it --rm -v /root:/mnt/root \
  zenoss/core_5.0:5.0.10 rsync -a /root/5.0.x /mnt/root
```

RHEL/CentOS:

```
docker run -it --rm -v /root:/mnt/root \
  zenoss/core_5.0:5.0.10 rsync -a /root/5.0.x /mnt/root
```

When the download completes, the `rsync` command copies scripts that perform the upgrade to a new directory, `/root/5.0.x`.

- 3 Start the upgrade script.

```
Ubuntu:sudo /root/5.0.x/upgrade-core-5.0.x.sh
RHEL/CentOS:/root/5.0.x/upgrade-core-5.0.x.sh
```

The upgrade takes about 25-50 minutes.

- 4 Start Zenoss Core.
 - a Log in to the Control Center browser interface.
 - b In the **Actions** column of the **Applications** table, click **Start**.

Overview of converting the Docker storage driver

As of Control Center 1.0.4, Zenoss recommends using an `xfs` filesystem for Docker storage, located at `/var/lib/docker`, and the `devicemapper` Docker storage driver.

The procedures in this section allow you to convert the Docker storage driver of a Control Center deployment from `aufs` or `btrfs` to `devicemapper`.

Note

- If you have a large deployment and have experienced problems with `/var/lib/docker` on the `btrfs` filesystem, we recommend that you convert the existing filesystem to `xfs` and `devicemapper`.

- If you choose to convert your Docker storage, perform the procedures in this section only after upgrading your deployment to Control Center 1.0.4 or later and Zenoss Core 5.0.4 or later.
- After converting the Docker storage, you do not need to perform this procedure following subsequent upgrades.

Preparing to convert the Docker storage driver

Perform this procedure to

- Verify that the Docker storage driver needs to be changed
- Determine whether a new, separate file system needs to be created for `/var/lib/docker`, or an existing file system may be converted

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

2 Determine whether the storage driver needs to be changed.

a Determine which storage driver Docker is using.

```
Ubuntu: sudo grep DOCKER_OPTS /etc/default/docker
RHEL/CentOS: grep DOCKER_OPTS /etc/sysconfig/docker
```

b Interpret the text returned by `grep`.

- If the text includes `-s aufs`, the storage driver is `aufs`. Continue this procedure.
- If the text includes `-s btrfs`, the storage driver is `btrfs`. Continue this procedure.
- If the text does not include the `-s` flag and the operating system is Ubuntu, the storage driver is `aufs`. Continue this procedure.
- If the text includes `-s devicemapper`, the storage driver is `devicemapper`. Discontinue this procedure. No changes are required on this host.

3 Determine whether `/var/lib/docker` is mounted on a separate `Btrfs` file system.

```
Ubuntu: sudo mount -t btrfs | awk '{ print $1 " " " $3 }'
```

```
RHEL/CentOS: mount -t btrfs | awk '{ print $1 " " " $3 }'
```

Example result:

```
/dev/sda3 /opt/serviced/var/volumes
/dev/sda3 /var/lib/docker
```

The first field shows the same device name, and one of the second fields contains `/var/lib/docker`. Therefore, `/var/lib/docker` is not mounted on its own file system, and you must create a partition for a new file system. For more information, refer to the host's operating system documentation.

Converting the storage driver of the master host

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

2 Stop Zenoss Core.

```
serviced service stop Zenoss.core
```

3 **Note** Perform this step only if the master host does not have internet access.

Optional: Create a backup of the Control Center internal services image.

The following command requires approximately 600MB in `/tmp`.

```
docker save zenoss/serviced-isvcs:v27.1 | gzip -9 > /tmp/isvcs.hold
```

4 Stop the Control Center and Docker services.

- If the operating system is RHEL or CentOS, enter the following commands:

```
systemctl stop serviced && systemctl stop docker
```

- If the operating system is Ubuntu, enter the following commands:

```
stop serviced && stop docker
```

- 5 Unmount `/var/lib/docker`.

```
umount /var/lib/docker
```

- 6 Create a new file system for `/var/lib/docker`.

- If `/var/lib/docker` was mounted on its own partition, replace *Device-Name* with the device identified previously.
- If you created a new partition for `/var/lib/docker`, replace *Device-Name* with the new partition's device.

RHEL/CentOS systems:

```
mkfs -t xfs -f Device-Name
```

Ubuntu systems:

```
mkfs.ext4 Device-Name
```

- 7 Comment the existing entry for `/var/lib/docker` in the file system table.
Replace *Device-Name* with the value used in the previous step.

```
sed -i -e '\|/var/lib/docker| s|^|#|' /etc/fstab
```

- 8 Add the new file system for `/var/lib/docker` to the file system table.
Replace *Device-Name* with the value used in the previous step.

RHEL/CentOS systems:

```
echo "${Device-Name} /var/lib/docker xfs defaults 1 2" >> /etc/fstab
```

Ubuntu systems:

```
echo "${Device-Name} /var/lib/docker ext4 defaults 1 2" >> /etc/fstab
```

- 9 Mount `/var/lib/docker`.

```
mount /var/lib/docker
```

- 10 Change the Docker storage driver.

- If the operating system is RHEL or CentOS, enter the following command:

```
sed -i -e '/DOCKER_OPTS/ s/btrfs/devicemapper/' \  
/etc/sysconfig/docker
```

- If the operating system is Ubuntu, open `/etc/default/docker` with a text editor, and then add `-s devicemapper` to the `DOCKER_OPTS` variable declaration.
- 11 Start the Docker service.

- If the operating system is RHEL or CentOS, enter the following command:

```
systemctl start docker
```

- If the operating system is Ubuntu, enter the following command:

```
start docker
```

- 12 **Note** Perform this step only if the master host does not have internet access, and you created a backup of the the Control Center internal services image in a previous step.

Optional: Restore the backup of the Control Center internal services image.

```
cat /tmp/isvcs.hold | gunzip - | docker load
```

- If your deployment includes resource pool hosts, proceed to [Converting the storage driver of a resource pool host](#) on page 15.
- If your deployment does not include resource pool hosts, proceed to [Starting Control Center and Zenoss Core](#) on page 16.

Converting the storage driver of a resource pool host

Convert the storage driver of the master host before converting the storage driver of resource pool hosts. For more information, see [Converting the storage driver of the master host](#) on page 13.

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

- 1 Log in to a Control Center resource pool host as `root`, or as a user with superuser privileges.
- 2 Stop the Control Center and Docker services.

- If the operating system is RHEL or CentOS, enter the following commands:

```
systemctl stop serviced && systemctl stop docker
```

- If the operating system is Ubuntu, enter the following commands:

```
stop serviced && stop docker
```

- 3 Unmount `/var/lib/docker`.

```
umount /var/lib/docker
```

- 4 Create a new file system for `/var/lib/docker`.

- If `/var/lib/docker` was mounted on its own partition, replace *Device-Name* with the device identified previously.
- If you created a new partition for `/var/lib/docker`, replace *Device-Name* with the new partition's device.

RHEL/CentOS systems:

```
mkfs -t xfs -f Device-Name
```

Ubuntu systems:

```
mkfs.ext4 Device-Name
```

- 5 Comment the existing entry for `/var/lib/docker` in the file system table.

Replace *Device-Name* with the value used in the previous step.

```
sed -i -e '\|/var/lib/docker| s|^|#|' /etc/fstab
```

- 6 Add the new file system for `/var/lib/docker` to the file system table.

Replace *Device-Name* with the value used in the previous step.

RHEL/CentOS systems:

```
echo "${Device-Name} /var/lib/docker xfs defaults 1 2" >> /etc/fstab
```

Ubuntu systems:

```
echo "${Device-Name} /var/lib/docker ext4 defaults 1 2" >> /etc/fstab
```

- 7 Mount `/var/lib/docker`.

```
mount /var/lib/docker
```

- 8 Change the Docker storage driver.

- If the operating system is RHEL or CentOS, enter the following command:

```
sed -i -e '/DOCKER_OPTS/ s/btrfs/devicemapper/' \
/etc/sysconfig/docker
```

- If the operating system is Ubuntu, open `/etc/default/docker` with a text editor, and then add `-s devicemapper` to the `DOCKER_OPTS` variable declaration.

- 9 Start the Docker service.

- If the operating system is RHEL or CentOS, enter the following command:

```
systemctl start docker
```

- If the operating system is Ubuntu, enter the following command:

```
start docker
```

- To perform this procedure on another resource pool, return to step 1.
- To start Zenoss Core, proceed to [Starting Control Center and Zenoss Core](#) on page 16.

Starting Control Center and Zenoss Core

Perform this procedure after all hosts in your Control Center deployment are converted to the `devicemapper` storage driver.

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

- If the operating system is RHEL or CentOS, enter the following command:

```
systemctl start serviced
```

- If the operating system is Ubuntu, enter the following command:

```
start serviced
```

3 Start Zenoss Core.

```
serviced service start Zenoss.core
```

Upgrading to 1.0.3 / 5.0.3

This chapter describes how to upgrade the combinations of Control Center and Zenoss Core shown in the following table.

From combination	To combination
Control Center 1.0.0 and Zenoss Core 5.0.0	Control Center 1.0.3 and Zenoss Core 5.0.3
Control Center 1.0.1 and Zenoss Core 5.0.1	Control Center 1.0.3 and Zenoss Core 5.0.3
Control Center 1.0.2 and Zenoss Core 5.0.2	Control Center 1.0.3 and Zenoss Core 5.0.3

For information about upgrading other combinations, see [Supported upgrade paths](#) on page 5.

Upgrade script

Zenoss provides an upgrade script for Zenoss Core called `version-upgrade-core.txt`, where *version* is the version that you are upgrading to. For example, if you are upgrading to version `x.y.z`, the file name would be `x.y.z-upgrade-core.txt`.

The upgrade script performs the underlying tasks required to upgrade the application binary and is provided as part of the new application images. When you pull the new Zenoss Core image from Docker hub, the script is automatically extracted and placed in the `root` directory of the master host.

The `version-upgrade-core.txt` script performs the following actions:

- Pulls the latest application images from Docker hub.

Note If maintenance downtime is an issue for your environment, you can perform this step manually to reduce the downtime window. For more information, see [Optional: Reducing the downtime maintenance window](#) on page 20.

- Preserves your custom data on the DFS.
- Re-installs all currently installed Zenpacks.

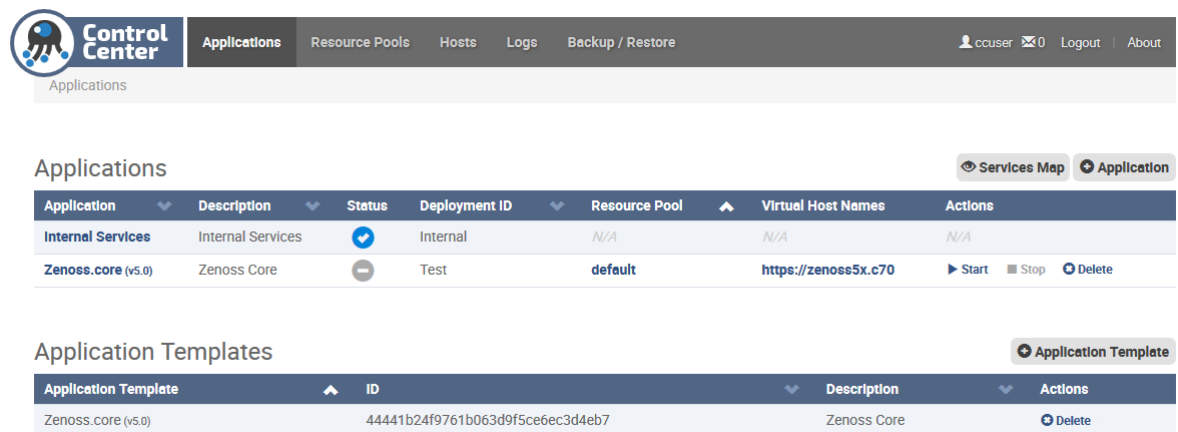
- Checks your deployment for any removed files, and if necessary, removes the corresponding file in the upgraded application.
- Checks for any locally installed patches and re-installs them.

Upgrading Control Center

This information describes how to upgrade Control Center (`serviced`) to v1.0.3.

To upgrade Control Center (`serviced`), you will use the web interface to stop all applications that are currently running. Then depending on your operating system environment, you will either use `yum` or `apt-get` to install the latest package from the Zenoss `apt` or `yum` repository to the master host. Once the package is installed, you will open Control Center and restart the stopped applications.

- 1 Log in to the Control Center web interface.
- 2 In the **Applications** table, identify the name of the Zenoss Core instance.
- 3 Stop the instance, and wait until all subservices are stopped.
 - a In the **Actions** column of the **Applications** table, click **Stop**.



The screenshot shows the Control Center web interface. The top navigation bar includes the Control Center logo, a user profile for 'ccuser', and links for 'Logout' and 'About'. The main content area is divided into two sections: 'Applications' and 'Application Templates'.

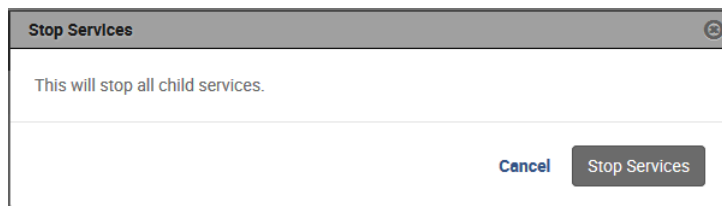
Applications Table:

Application	Description	Status	Deployment ID	Resource Pool	Virtual Host Names	Actions
Internal Services	Internal Services	Running (blue circle)	Internal	N/A	N/A	N/A
Zenoss.core (v5.0)	Zenoss Core	Stopped (grey circle)	Test	default	https://zenoss5x.c70	Start Stop Delete

Application Templates Table:

Application Template	ID	Description	Actions
Zenoss.core (v5.0)	44441b24f9761b063d9f5ce6ec3d4eb7	Zenoss Core	Delete

- b In the **Stop Services** dialog, click **Stop Services**.



- c Display the child services of Zenoss Core.

In the **Application** column of the **Applications** table, click Zenoss Core, and then scroll down to the **Services** table. Stopped services have a grey circle icon in the **Status** column. Do not proceed until all services are stopped.
- 4 Log in to the Control Center master host as a user with `sudo` privileges and install the latest Control Center (`serviced`) package.
 - RHEL:

```
sudo yum --enablerepo=zenoss-stable install -y serviced-1.0.3
```

- Ubuntu:

```
sudo apt-get update && sudo apt-get install -y serviced=1.0.3
```

The Control Center web interface and `serviced` binary are updated.

- 5 If you are upgrading another application, refer to the upgrade steps for that product. Otherwise, open the Control Center web interface, click **Start** in the **Actions** column of the **Applications** table to restart Zenoss Core.

Upgrading Zenoss Core

This section contains the steps to upgrade the Zenoss Core application.

Optional: Reducing the downtime maintenance window

To minimize downtime and verify that you have adequate disk space, you can pull the required images manually, prior to your planned maintenance window. If you choose not to pull the images manually, the upgrade script performs these actions for you.

Note Replace *Version* in the following commands with the appropriate version number. For version information, refer to the release notes.

- 1 Log in to the master host as a user with `serviced` privileges.
- 2 Pull the most recent Zenoss Core image to the local master and copy the the upgrade script to the root directory:

```
sudo docker run -it --rm -v /root:/mnt/root
zenoss/core_5.0:5.0.Version \
rsync -a /root/5.0.x /mnt/root
```

The Zenoss Core pull takes approximately 10 - 20 minutes.

- 3 Synchronize the image with the local `serviced` registry:

```
serviced docker sync
```

The sync for Zenoss Core takes approximately 5 minutes.

Upgrading the Zenoss Core application

This procedure describes how to upgrade Zenoss Core to 5.0.0, 5.0.1, or 5.0.2.

Note Replace *Version* in the following commands with the appropriate version number. For version information, refer to the release notes.

- 1 If Zenoss Core is currently running, log in to the Control Center web interface and click **Stop** in the **Actions** column of the **Applications** table.
- 2 Display the child services of Zenoss Core.
In the **Application** column of the **Applications** table, click Zenoss Core, and then scroll down to the **Services** table. Stopped services have a grey circle icon in the **Status** column. Do not proceed until all services are stopped.
- 3 Log in to the Control Center master host as a user with `serviced` privileges.
- 4 Pull the most recent Zenoss Core images to the local master and copy the upgrade script to the `root` directory.

```
sudo docker run -it --rm -v /root:/mnt/root
zenoss/core_5.0:5.0.Version \
```

```
rsync -a /root/5.0.x /mnt/root
```

The Zenoss Core pull takes approximately 10 - 20 minutes.

- 5 Run the upgrade script replacing *Version* in the examples below with the new micro version number. For the latest version information, see the release notes.

```
serviced script run /root/5.0.x/5.0.Version-upgrade-core.txt \  
--service Zenoss.core
```

The Zenoss Core application is upgraded and your custom data is preserved.

- 6 Log in to the Control Center web interface and click **Start** in the **Actions** column of the **Applications** table to start Zenoss Core.