



Zenoss Core Configuration Guide

Release 5.1.2

Zenoss, Inc.

www.zenoss.com

Zenoss Core Configuration 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: 1041.16.118

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

Contents

Chapter 1: Configuring Zenoss Core.....	5
Changing default passwords.....	5
Deleting the RabbitMQ guest user account.....	6
Creating a weekly maintenance script.....	6
MariaDB database utilities.....	7
Assigning a virtual IP address to a resource pool.....	8
Configuring SSL (Optional).....	9
Monitoring IPv6 targets (Optional).....	9
Installing Quilt for patch management (Optional).....	10
Configuring OpenTSDB compaction (Optional).....	12
Chapter 2: Preparing Your Infrastructure for Monitoring.....	13
Extending Monitoring with ZenPacks.....	13
Preparing Network Devices.....	14
Preparing Storage Devices.....	15
Preparing Server Devices.....	17
Preparing Hypervisor Devices.....	17
Chapter 3: Starting Zenoss Core.....	18
Starting Zenoss Core from Control Center.....	18
Starting Zenoss Core from the command-line.....	19
Public Endpoints.....	19
Chapter 4: Setting up Zenoss Core.....	26
Completing the Startup Wizard.....	26
Setting Up Users.....	27
Discovering the Network.....	27
Adding Infrastructure.....	28
Classifying Discovered Devices.....	31
Updating Device Authentication Details.....	32
Adding or Editing Information on a Device Record.....	32
Chapter 5: Modeling Devices.....	34
Configuring Windows Devices to Provide Data Through SNMP.....	34
Configuring Linux Devices to Provide Data Through SNMP.....	35
Modeling Devices Using SSH/COMMAND.....	35
Using Device Class to Monitor Devices Using SSH.....	36
Using the /Server/Scan Device Class to Monitor with Port Scan.....	36
Modeling Devices Using Port Scan.....	36
About Modeler Plugins.....	37
Debugging the Modeling Process.....	38
Next Steps.....	38

Appendix A: External HBase configuration.....	39
Configuring OpenTSDB for an external HBase cluster.....	39
Configuring the OpenTSDB service startup command.....	40
Disabling the Zenoss Core HBase cluster.....	41

1

Configuring Zenoss Core

This chapter contains configuration procedures that you perform after Zenoss Core is installed. Some of the procedures are optional, and indicated as such in the section title. For installation and deployment instructions, refer to the *Zenoss Core Installation Guide*.

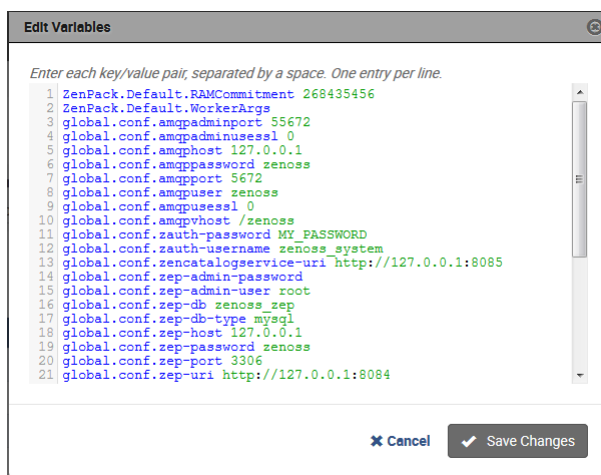
Changing default passwords

Zenoss Core includes several services with independent authentication systems, each of which have default passwords defined by Zenoss. Global configuration information, including passwords, is located in the `global.conf` file. This procedure shows you how to modify this file using Control Center.

Note You may change any default password. However, Zenoss recommends that you do not change account names.

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click **Zenoss.core**.
- 3 In the application title line, click **Edit Variables**.

The Edit Variables dialog is opened and displays the `global.conf` file:



- 4 Change the default password of the RabbitMQ service:
 - a In the **Edit Variables** dialog, locate the `global.conf.amqppassword` variable.
 - b Replace the default value, `zenoss`, with a new password.
- 5 Change the default password of the Zenoss authentication proxy:

- a In the **Edit Variables** dialog, locate the `global.conf.zauth-password` variable.
- b Replace the default value, `MY_PASSWORD`, with a new password.
- 6 Edit other passwords as desired, and then click **Save Changes**.
- 7 To pick up the new passwords, click **Restart** to restart the **Zenoss.core** service.

Note You can follow this basic procedure to edit the variables of other services. Simply locate the service in the **Services** table, click **Edit Variables**. You can then make the necessary changes, save the file, and restart the service.

Deleting the RabbitMQ guest user account

By default, RabbitMQ distributions include the `guest` user account. To prevent security issues, Zenoss recommends deleting the account.

- 1 Log in to the Control Center master host as a user with `serviced` CLI privileges.
- 2 Attach to the RabbitMQ container.

```
serviced service attach rabbitmq
```

- 3 Delete the guest user account.

```
rabbitmqctl delete_user guest
```

- 4 Exit the container session.

```
exit
```

- 5 Restart the RabbitMQ service.

```
serviced service restart rabbitmq
```

Creating a weekly maintenance script

The Zenoss Core databases require regular maintenance to perform optimally. This procedure creates a script for `cron` to run once a week, to perform the required maintenance.

- 1 Log in to the Control Center master host as `root`, or as a user with superuser privileges.
- 2 Create a shell script for `cron` to invoke.
 - a Open `/etc/cron.weekly/serviced` with a text editor.
The file is empty.
 - b Add the following content to the file.

```
#!/bin/sh

/bin/serviced service run zope zenosddbpack
```

- c Save the file, and then close the text editor.
- 3 Set file permissions.

```
chmod 0755 /etc/cron.weekly/serviced
```

MariaDB database utilities

The *Percona Toolkit* is a collection of helpful utilities for MySQL and MariaDB databases. For licensing reasons, Zenoss can not distribute it. Zenoss strongly recommends that all installations of Zenoss Core install the Percona Toolkit.

Installing the Percona Toolkit with internet access

To perform this procedure, you need one of the following:

- a login account on the master host that is a member of the `docker` group
- the password of the `root` user account

For more information, refer to the *Zenoss Core Installation Guide*.

- 1 Log in to the Control Center master host.
- 2 Install the package.

```
serviced service run zope install-percona
```

At the end of the installation process, the message `Container not committed` is displayed. This is normal. The tools are installed in the distributed file system, not in an image.

Installing the Percona Toolkit without internet access

To perform this procedure, you need one of the following:

- a login account on the master host that is a member of the `docker` group
- the password of the `root` user account

In addition, you need the Percona Toolkit package file. This procedure includes steps for downloading it to a client system, and then copying it to the Control Center master host.

- 1 On a client system, use a web browser to download *the latest version of the Percona Toolkit package*.
- 2 Log in to the Control Center master host.
- 3 Prepare the package for installation.
 - a On the Control Center master host, create a directory for the package, and then change directory.

```
mkdir /tmp/percona && cd /tmp/percona
```

- b Copy the package to the temporary location.
You may use a file transfer utility such as *WinSCP*.
- c Update the access permissions of the file and directory.

```
chmod -R 777 /tmp/percona
```

- 4 Start a shell as the `zenoss` user in a Zope container.
 - a Change directory to the location of the Percona Toolkit file.

```
cd /tmp/percona
```

- b Start an interactive shell in a Zope container and save a snapshot named `PerconaToolkit`.

```
mySnap=InstallPerconaToolkit
```

```
serviced service shell -i -s $mySnap zope bash
```

- c Switch user to zenoss.

```
su - zenoss
```

- 5 Install the package and exit the Zope container.

- a Create a directory for the package.

```
PERCONADIR=/var/zenoss/percona
mkdir -p $PERCONADIR
```

- b Extract the package files.

Replace *Version* with the version number of the package file:

```
tar --strip-components=1 -C $PERCONADIR -xzvf \
/mnt/pwd/percona-toolkit-Version.tar.gz
```

- c Exit the zenoss shell.

```
exit
```

- d Exit the Zope container.

```
exit
```

- 6 Commit the named snapshot.

```
serviced snapshot commit $mySnap
```

- 7 Restart the zeneventserver service.

```
serviced service restart zeneventserver
```

Assigning a virtual IP address to a resource pool

Resource pools are assigned to a specific host's IP address. If the server that provides the IP address to that host is not available, Control Center is unable to deploy a service that requires the IP address. You can avoid this issue by assigning virtual IP addresses to the Resource Pool. The virtual IP address must be on the same subnet and must not be used by other applications or services. Using the virtual IP address, Control Center can "float" a deployed service on any host in the resource pool.

Before you begin, Zenoss recommends that you contact your network administrator to assign and reserve an IP address on the resource pool's subnet. Reserving this IP address helps avoid future conflicts.

To assign a virtual IP address to a resource pool:

- 1 Log in to the Control Center browser interface.
- 2 Click **Resource Pools** at the top of the page.
- 3 In the Resource Pools table, click a resource pool name.
For example, click the **default** resource pool to see the Virtual IPs and Hosts for this pool.
- 4 To the right of the Virtual IPs table, click **Add Virtual IP**.
- 5 In the **Add Virtual IP** dialog, enter an additional, available IP address on the appropriate subnet. Contact your network administrator to reserve this address and avoid future conflicts.
- 6 Specify the **Netmask** and **Interface** fields.

Note The specific netmask and interface name must be the same on all hosts in the pool.

7 Click **Add Virtual IP**.

The IP address is now assigned to the resource pool and Control Center will automatically configure the IP address on one of the hosts in the Resource Pool. The Virtual IP then appears in the IP address assignments list.

Configuring SSL (Optional)

To use an SSL certificate with Control Center and Zenoss Core, the `serviced` service must know the location of the SSL key and cert files. This procedure describes how to specify these file locations in the `/etc/default/serviced` file.

Note If your environment uses a reverse proxy, contact Zenoss Support for further assistance.

Before you perform this procedure:

- Obtain a digitally signed SSL certificate from a Certificate Authority or generate one with a utility such as OpenSSL.
- Verify that you have access to the `root` account on the Control Center master host.

1 Log in to the Control Center master host.

2 Using a text editor, open the `/etc/default/serviced` file.

3 Locate the following lines and enter the appropriate path for each file. If these lines do not currently exist, add them to the end of the file.

```
SERVICED_KEY_FILE=<path_to_key_file>
```

```
SERVICED_CERT_FILE=<path_to_cert_file>
```

Note Files that require a passphrase are currently not supported.

4 Save the file and exit the text editor.

5 Reload `serviced`:

```
sudo systemctl reload serviced
```

`serviced` is restarted with the SSL configuration changes.

Monitoring IPv6 targets (Optional)

The following procedure describes how to monitor devices on an IPv6 network using a routed subnet. In this procedure you will route an IPv6 address block from your network to the Control Center using the `docker0` interface. You can monitor IPv6 targets assigned out of the routed block.

This example assumes a single resource pool with the following network parameters:

- Router IP address: 2001:0db8:200b::1/64
- Resource pool IP address: 2001:0db8:200b::2/64
- Routed subnet (to resource pool's IP): 2001:0db8:dce3::/80 (minimum of /80 required)
- IPv6 DNS servers: 2001:0db8:200b::100, 2001:0db8:200b::200

Replace the example addresses with real addresses from your network.

1 Log on to the master host as user with root privileges.

2 Enable IPv6 packet forwarding:

a Use a text editor to open the `/etc/sysctl.conf` file.

b Locate and uncomment the following line:

```
net.ipv6.conf.all.forwarding=1
```

- c Save the file.
- 3 Activate IPv6 packet forwarding without rebooting the host.


```
sysctl -w net.ipv6.conf.all.forwarding=1
```
- 4 Add IPv6 resolvers to the networking options for Docker using the `DOCKER_OPTS` variable.

Replace the IP address below with the appropriate value for your network.

- CentOS:

```
DOCKER_OPTS="--dns 2001:0db8:200b::100 --dns 2001:0db8:200b::200 \  
--ipv6 --fixed-cidr-v6=" 2001:0db8:dce3::/80" >> \  
/etc/sysconfig/docker
```

- Ubuntu:

```
DOCKER_OPTS="--dns 2001:0db8:200b::100 --dns 2001:0db8:200b::200 \  
--ipv6 --fixed-cidr-v6=" 2001:0db8:dce3::/80" >> \  
/etc/default/docker
```

- 5 Connect to an IPv6 container and send a ping to a valid IPv6 address.

```
serviced service attach zenping \  
ping6 -c 1 www.google.com
```

If the ping is successful, Docker is able to resolve IPv6 addresses and you can monitor devices with IPv6 addresses. If you are not able to ping an IPv6 address successfully, or you need help with this procedure, contact Zenoss Support.

Installing Quilt for patch management (Optional)

Quilt is a free patch management utility. You can use Quilt to apply or remove patches, and also keep track of the changes a patch makes. If your Zenoss Core installation contains any customizations, Zenoss recommends that you install Quilt.

To perform this procedure, you need one of the following:

- A login account on the master host that is a member of the `docker` group
- The password of the `root` user account

For more information about these prerequisites, refer to the *Zenoss Core Installation Guide*.

Perform this procedure to add the Quilt patch management system on a Zenoss Core image.

- 1 Log in to the Control Center master host.
- 2 Install the Quilt package.

```
serviced service run Zope install-quilt
```

Installing the Quilt package with internet access

To perform this procedure, you need one of the following:

- a login account on the master host that is a member of the `docker` group
- the password of the `root` user account

Perform this procedure to add the Quilt patch management system to a Zenoss Core image.

- 1 Log in to the Control Center master host.
- 2 Install the Quilt package.

```
serviced service run zope install-quilt
```

Installing the Quilt package without internet access

To perform this procedure, you need the password of the `root` user account. In addition, you need the Quilt package file. This procedure includes steps for downloading it to a client system, and then copying it to the Control Center master host.

Perform this procedure to add the Quilt patch management system to a Zenoss Core image.

- 1 On a client system, use a web browser to download *the latest version of the Quilt package*.
- 2 Log in to the Control Center master host.
- 3 Prepare the package for installation.
 - a On the Control Center master host, create a directory for the package, and then change directory.

```
mkdir /tmp/quilt && cd /tmp/quilt
```

- b Copy the package to the temporary location.
You may use a file transfer utility such as *WinSCP*.
- c Update the access permissions of the file and directory.

```
chmod -R 777 /tmp/quilt
```

- 4 Start a shell as the `zenoss` user in a Zope container.
 - a Change directory to the location of the Quilt package file.

```
cd /tmp/quilt
```

- b Start an interactive shell in a Zope container and save a snapshot named `InstallQuilt`.

```
mySnap=InstallQuilt
serviced service shell -i -s $mySnap zope bash
```

- c Switch user to `zenoss`.

```
su - zenoss
```

- 5 Extract the package files, and then compile and install Quilt.
 - a Extract the package files.

```
tar xzvf /mnt/pwd/quilt-*.tar.gz -C /tmp
```

- b Compile and install the package.

```
cd /tmp/quilt-* && ./configure --prefix=/opt/zenoss/var/ext \
&& make && make install
```

- 6 Exit the container.
 - a Exit the `zenoss` shell.

```
exit
```

- b Exit the Zope container.

```
exit
```

- 7 Commit the named snapshot.

```
serviced snapshot commit $mySnap
```

Configuring OpenTSDB compaction (Optional)

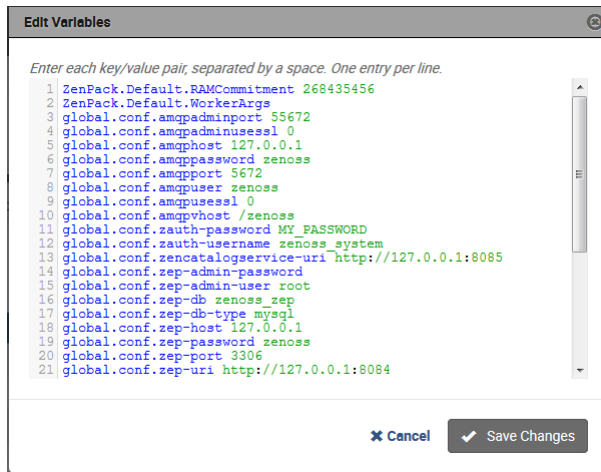
Zenoss Core uses an OpenTSDB database to store the monitoring data it collects. When OpenTSDB compaction is enabled, multiple columns in an HBase row are merged into a single column to reduce disk space. In testing, Zenoss has observed that these merges result in duplicate data points, so by default, compaction is disabled. Duplicate data points do not affect the integrity of the data.

Note Enabling compaction may degrade system performance. For technical assistance, contact Zenoss Support.

Perform this procedure to enable OpenTSDB compaction.

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click **Zenoss.core**.
- 3 In the application title line, click **Edit Variables**.

The Edit Variables dialog is opened and displays the `global.conf` file:



- 4 In the **Edit Variables** dialog, scroll to the bottom of the list.
- 5 Change the value of the `tsd.storage.enable_compaction` variable from `False` to `True`.
- 6 Click **Save Changes**.

Preparing Your Infrastructure for Monitoring

2

Zenoss Core uses standard management APIs to collect performance data, and therefore does not install proprietary agents on your infrastructure devices to collect monitoring data. However, Zenoss recommends that you review the information in this chapter to verify that the devices to you want to monitor are ready to respond to requests for data.

Note This chapter describes how to prepare the most common IT infrastructure. If the infrastructure you want to monitor is not described here, please refer to the corresponding ZenPack documentation in the [Zenpack Catalog](#).

When your infrastructure is ready to monitor, the Zenoss Core Setup Wizard guides you through the process of automatically discovering devices on your network and then adding devices by category and type.

Extending Monitoring with ZenPacks

Datacenters typically contain many different types of hardware, software, and cloud services from a long list vendors. To keep your company's data secure, all devices, network infrastructure, and services must be monitored.

Zenoss Core is ready to monitor a large number of common devices and network infrastructure as soon it is installed. However, you can monitor an even larger number of devices in Zenoss Core through the use of Zenpacks. A ZenPack is a plug-in that extends not only monitoring capabilities, but also adds new capabilities to the Zenoss Core itself. This can be as simple as adding new device classes or monitoring templates, or as complex as extending the data model and providing new collection daemons.

There are hundreds of ZenPacks available, some developed, supported, and maintained by Zenoss, and many others that are developed and maintained by the Zenoss user community.

You can use ZenPacks to add:

- Monitoring templates
- Data sources
- Graphs
- Event classes
- User commands
- Reports
- Model extensions
- Product definitions

Simple ZenPacks can be created completely within the Zenoss Core. More complex ZenPacks require development of scripts or daemons, using Python or another programming language. ZenPacks can also be distributed for

installation on other Zenoss Core systems. For information on how to create a new ZenPack, refer to *Zenoss Core Administration Guide*.

ZenPack Information Resources

Zenoss Core includes a link (the question mark icon) to the documentation of the ZenPacks that are included in your installation of Zenoss Core. It also provides access to the [ZenPack Catalog](#), which provides detailed descriptions of all of the ZenPacks developed by Zenoss.

You may also create your own ZenPacks, or download and install ZenPacks developed by others. The following list identifies ZenPack resources:

- [ZenPack Discussion Forum](#)
- [ZenPack Development Forum](#)
- [Public Zenoss repositories on GitHub](#)

Displaying Installed ZenPacks in Zenoss Core

To display the pre-installed ZenPacks on Zenoss Core:

- 1 In the browser interface, select the **ADVANCED** tab.
- 2 In the left column, select **ZenPacks**.

The following figure shows an example list of ZenPacks.

Pack	Package	Author	Version	Egg
ZenPacks.zenoss.AdvancedSearch	zenoss	Zenoss	1.1.4	Yes
ZenPacks.zenoss.AixMonitor	zenoss	Zenoss	1.3.0	Yes
ZenPacks.zenoss.ApacheMonitor	zenoss	Zenoss	2.1.4	Yes
ZenPacks.zenoss.AuditLog	zenoss	Zenoss	1.3.0	Yes
ZenPacks.zenoss.BigIpMonitor	zenoss	Zenoss	2.6.3	Yes
ZenPacks.zenoss.BrocadeMonitor	zenoss	Zenoss	2.1.1	Yes
ZenPacks.zenoss.CatalogService	zenoss	Zenoss	3.0.9	Yes
ZenPacks.zenoss.CheckPointMonitor	zenoss	Zenoss	2.0.0	Yes
ZenPacks.zenoss.CiscoMonitor	zenoss	Zenoss	5.3.1	Yes
ZenPacks.zenoss.CiscoUCS	zenoss	Zenoss	1.9.1	Yes
ZenPacks.zenoss.ControlCenter	zenoss	Zenoss	1.0.0	Yes
ZenPacks.zenoss.Dashboard	zenoss	Zenoss	1.0.3	Yes
ZenPacks.zenoss.DellMonitor	zenoss	Zenoss	2.2.0	Yes

- 3 To monitor infrastructure that does not appear in the **Loaded ZenPacks** list, download the required ZenPack from the [ZenPack Catalog](#).

Once the ZenPack is installed, you can then add the infrastructure to Zenoss Core.

Preparing Network Devices

Preparing Switches and Routers

To prepare a switch or router device for monitoring, verify that an SNMP agent is installed and currently running on the device.

Note This rest of this section describes how to prepare Cisco network devices for monitoring. For other device types, refer to the [ZenPack catalog](#) documentation.

Preparing Cisco UCS Network Devices

Zenoss Core uses SNMP to provide customized or generalized support for many Cisco products.

The following table associates Cisco products with the customized Zenoss Core device types that support them. Device types are listed in the **Network** area of the **Add Infrastructure** wizard, which is both part of the setup wizard and available through the Zenoss Core browser interface.

Note Some of the supported products, such as the Cisco Nexus 7000 and 9000 switches, represent a large number of discrete monitoring endpoints. If you are unsure which Zenoss Core virtual machine size supports the number of high-density devices you wish to monitor, contact your Zenoss representative.

Note In order to monitor Cisco Nexus 9000 Series devices, you must first enable NX-API with the **feature** manager CLI command on the device. For detailed instructions on performing this task, see the following Cisco documentation: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/6-x/programmability/guide/b_Cisco_Nexus_9000_Series_NX-OS_Programmability_Guide/b_Cisco_Nexus_9000_Series_NX-OS_Programmability_Configuration_Guide_chapter_0101.html#concept_BCCB1EFF9C4A4138BECE9ECC0C4E38DF

Cisco product	Device type
Cisco Catalyst 6500 and 3560 Series Switches	Cisco 6500 (SNMP)
Cisco Nexus 5000 Series Switches	Cisco Nexus 5000 (SNMP + Netconf)
Cisco Nexus 7000 Series Switches	Cisco Nexus 7000 (SNMP + Netconf)
Cisco Nexus 1000v Series Switches	Cisco Nexus 1000V (SNMP + Netconf)
Cisco Nexus 3000 Series Switches	Cisco Nexus 3000 (SNMP + Netconf)
Cisco Nexus 9000 Series Switches	Cisco Nexus 9000 (NX-API)
Cisco Catalyst 6500 Series Virtual Switching Systems	Cisco VSS (SNMP)
Cisco MDS 9000 Series Multilayer Switches	Cisco MDS 9000 (SNMP)

In addition, Zenoss Core provides two generalized device types.

Cisco product	Device type
Cisco CatOS-based switches or routers	Generic Switch/Router (SNMP)
Cisco IOS-based switches or routers	Cisco IOS (SNMP)

Preparing Storage Devices

Note This section describes how to prepare NetApp and EMC storage devices for monitoring. For other device types, refer to the [ZenPack catalog](#) documentation.

Legacy NetApp Filers

Zenoss Core uses SNMP to monitor legacy NetApp Filers that do not support the Data ONTAP® API (ZAPI).

Note The data gathered are approximate, because the values for many objects (Aggregate, Volume, Plex, and RAID group) are not exposed by the NetApp MIB.

To prepare a legacy NetApp Filer for monitoring, verify that SNMPv2 is installed, and then start an SNMP agent.

Recent NetApp Filers

Zenoss Core uses HTTP to monitor NetApp Filers that support the Data ONTAP® API (ZAPI).

To prepare a recent NetApp Filers for monitoring, verify the following conditions:

- The Filer is running in 7-Mode or C-Mode.
- ZAPI is installed and enabled. Version 8.x, or a more recent version, is required.

Also, you need the username and password of an account on the Filer that is authorized to use ZAPI.

EMC Storage Arrays

Zenoss Core uses the Web-Based Enterprise Management (WBEM) protocol to send queries to EMC Storage Management Initiative Specification (SMI-S) providers associated with EMC VMAX and VNX storage arrays.

To prepare EMC arrays for monitoring, verify that at least one EMC SMI-S provider is running for each type of array to monitor. (The VMAX and VNX data models are different.) In addition, you need the following information:

- The username and password of a user account that is authorized to collect data on each SMI-S provider.
- The IP address of each SMI-S provider.
- The port number at which each SMI-S provider listens for requests.
- Whether or not to use SSL.

Zenoss recommends verifying that an SMI-S provider is responding to requests before adding it to Zenoss Core.

Note Many of the graphs for components types of EMC arrays display NaN when statistics logging is disabled on the EMC device. The logging feature has a low default timeout value, and must be set to a higher value or turned on again periodically.

Verifying an SMI-S provider on EMC devices

To perform this procedure, you need a Linux host that has a network path to the SMI-S providers of the arrays to monitor.

Note Do not perform this procedure on the Zenoss Core host.

Perform this procedure to verify that the SMI-S providers associated with EMC arrays are configured correctly, and are responding to WBEM queries from command line tools.

- 1 Log in to a Linux host as `root`, or as a user with superuser privileges.
- 2 Install a WBEM command-line interface package, such as `wbemcli`.
- 3 Verify the SMI-S provider. Replace the variables with values that are valid in your environment.

```
wbemcli IP-Address:Port -u admin -p 'Password' -n root/emc --no-ssl
ei('EMC_DiskDrive')
```

The expected result is a list of Disk Drive classes.

Preparing Server Devices

Note This section describes how to prepare Linux and Windows servers for monitoring. For other device types, refer to the [ZenPack catalog](#) documentation.

Linux Servers

Zenoss Core uses SNMP or SSH to monitor Linux servers.

To prepare a Linux server for SNMP monitoring, install an SNMP package on the server (for example, [Net-SNMP](#)) and start the agent.

To prepare a Linux server for SSH monitoring, install an SSH server package (for example, [OpenSSH](#)) and start the SSH daemon. Also, obtain the username and password of a user account on the server that has standard user privileges (root privileges are not required).

Windows Servers

Zenoss Core uses SNMP or WinRM to monitor the following Microsoft Windows systems:

- Microsoft Windows Server 2012 and 2012 R2
- Microsoft Windows Server 2008 R2

To prepare a Windows system for SNMP monitoring, start the SNMP service.

To prepare a Windows system for WinRM monitoring, <https://support.zenoss.com/hc/en-us/articles/202432249-How-To-Monitor-Windows-2012-2012-R2-2008R2-And-2003-Standard-Edition-Servers-With-The-Microsoft-Windows-Zenpack-Using-WinRM->

Preparing Hypervisor Devices

Note This section describes how to prepare vSphere and Hyper-V hypervisors for monitoring. For other device types, refer to the [ZenPack catalog](#) documentation.

vSphere EndPoint

Zenoss Core uses SOAP to monitor VMware vSphere servers running vSphere 4.1, 5.0, 5.1, 5.5, or 6.0.

To prepare a VMware vSphere server for monitoring, verify the software version, and obtain the username and password of an account on the server that is authorized to use the vSphere API and determine whether or not to use SSL.

Hyper-V

Zenoss Core uses WinRM to monitor the following Microsoft Hyper-V systems:

- Microsoft Hyper-V Server 2012 and 2012 R2
- Microsoft Hyper-V Server 2008 and 2008 R2

To prepare a Hyper-V Server for WinRM monitoring, refer to the appendix, <https://support.zenoss.com/hc/en-us/articles/202432249-How-To-Monitor-Windows-2012-2012-R2-2008R2-And-2003-Standard-Edition-Servers-With-The-Microsoft-Windows-Zenpack-Using-WinRM->

Starting Zenoss Core

This chapter contains information on how to start Zenoss Core and its related services from Control Center or the command line, how to open the Zenoss Core browser interface, and how to create public endpoints.

Starting Zenoss Core from Control Center

Before you can log into Zenoss Core browser interface, you must first start the Zenoss Core application. This procedure describes how to start the application and open the browser interface from the Control Center.

To perform this procedure, you need:

- A supported client system and browser
- A name resolution entry for the Control Center master host on your DNS server or in your local `/etc/hosts` file
- A user account on the master host that has access to the Control Center browser interface

For more information on fulfilling these pre-requisites, refer to the *Zenoss Core Installation Guide*.

To start Zenoss Core from Control Center:

- 1 Log in to the Control Center browser interface.

The screenshot shows the Zenoss Control Center interface. At the top, there is a navigation bar with the Control Center logo and tabs for Applications, Resource Pools, Hosts, Logs, and Backup / Restore. Below the navigation bar, there is a section for Applications. The Applications table has the following data:

Application	Description	Status	Deployment ID	Resource Pool	Public Endpoints	Actions
Internal Services	Internal Services	Internal	N/A	N/A	N/A	N/A
Zenoss.core (v5.1.1)	Zenoss Core	Test	default	https://zenoss5.ip-10-111-23-202.zenoss.loc:443	Start Stop Delete	

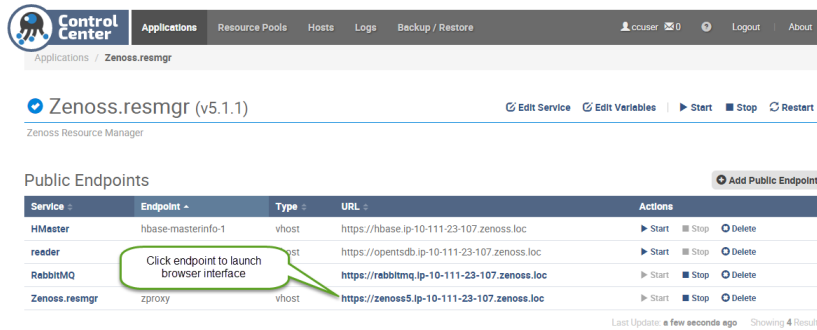
Below the Applications table, there is a section for Application Templates. The Application Templates table has the following data:

Application Template	ID	Description	Actions
Zenoss.core (v5.1.1)	673635369567a02388d4122bf61e4424	Zenoss Core	Delete

- 2 In the **Actions** column of the **Applications** table, click **Start** for Zenoss.resmgr.
- 3 In the **Start Service** dialog, click **Start Service and all 52 child services**.
- 4 Optional: Monitor the startup.
 - a In the **Applications** table, click **Zenoss.core**.
 - b Scroll down to the **Services** table and review the **Instances** icon for each service.

As services are started the Instance icon changes from a dash on a grey background to a check mark on a blue background.

- When all Zenoss Core services are started, click the **Public Endpoint** link to launch the browser interface.



The Zenoss Core Startup Wizard is displayed. For more information, see [Completing the Startup Wizard](#).

Starting Zenoss Core from the command-line

Before you can log into Zenoss Core browser interface, you must first start the Zenoss Core application. This procedure describes how to start the application from the command-line, and then open the browser interface.

To perform this procedure, you need one of the following:

- A login account on the master host that is a member of the `docker` group
- The password of the `root` user account

For more information about these pre-requisites, refer to the [Zenoss Core Installation Guide](#).

To start Zenoss Core from Control Center command-line:

- 1 Log in to the Control Center master host.
- 2 Start Zenoss Core:

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

- 3 Optional: Monitor the startup:

```
serviced service status Zenoss.core
```

- 4 When all Zenoss Core services are started, you can launch the browser interface by entering either the hostname or IP address of the public endpoint in your web browser. For example:

```
https://hostname
https://IP_address
```

The Zenoss Core Startup Wizard is displayed. For more information, see [Completing the Startup Wizard](#).

Public Endpoints

Public endpoints provide access to Zenoss applications and services using one of the following methods:

- A virtual hostname (the default)

Control Center provides a virtual hostname public endpoint for Zenoss Core (`Zenoss.core`) and several internal services (`hbase-masterinfo-1`, `opentsdb-reader`, and `rabbitmq_admin`).

- IP address and port

Optionally, you can create a port-based public endpoint to avoid the name resolution requirement when using a virtual hostname public endpoint, or if you want to access an application or service with a third-party tool.

Virtual host public endpoints

Control Center, the management and orchestration tool for Zenoss Core, proxies the IP addresses of the services and applications that it manages. These addresses can change during normal operations. Therefore, Control Center provides a virtual hostname to facilitate access to the Zenoss Core web server and child services.

The default virtual hostname for the Zenoss Core public endpoint is `core`.

To access applications using the virtual hostname, you must resolve the virtual hostname to the physical host. You can enable network-wide access, or configure access on individual client systems.

- To configure network-wide access, add the Zenoss Core virtual hostname to the DNS servers in your environment.
- To configure client systems individually, add the virtual hostname to the `C:\Windows\System32\drivers\etc\hosts` file (Windows systems) or the `/etc/hosts` file (Linux and OS/X systems), as described in the following sections.

If your environment requires a specific syntax for the virtual hostname, you can create a new virtual hostname public endpoint. For more information, refer to [Creating a public endpoint \(Optional\)](#) on page 21.

Configuring name resolution on a Windows 7 system

Perform this procedure to resolve the Zenoss Core virtual hostname to its physical host.

Note You must perform this procedure before you launch the Zenoss Core user interface.

To perform this procedure, you need Windows Administrator privileges on the client system.

To configure name resolution on a Windows 7 system:

- 1 Log in to the Windows 7 system as a user with Administrator privileges.
- 2 From the **Start** menu, highlight **All Programs > Accessories > Notepad**.
- 3 Right click, and then select **Run as administrator**.
- 4 From the Notepad **File** menu, select **Open**.
- 5 In the **File name** field of the **Open** window, enter `C:\Windows\System32\drivers\etc\hosts`.
- 6 Add the name resolution entry for your Control Center master host to the end of the file.

For example, the following entry identifies a Control Center master host at IP address `192.0.2.12`, hostname `cc-master`, in the `example.com` domain.

```
192.0.2.12 cc-master.example.com cc-master zenoss5.cc-master
```

- 7 Save the file, and then exit Notepad.

Configuring name resolution on a Linux or OS/X system

Perform this procedure to resolve the Zenoss Core virtual hostname to its physical host.

Note You must perform this procedure before you launch the Zenoss Core user interface.

To perform this procedure, you need superuser privileges on the client system.

To configure name resolution on a Linux/Mac OS/X system:

- 1 Log in to the client system as `root` or as a user with `sudo` privileges.
- 2 Open the `/etc/hosts` file in a text editor.
- 3 Add the name resolution entry for your Control Center master host to the end of the file.

For example, the following entry identifies a Control Center master host at IP address `192.0.2.12`, hostname `cc-master`, in the `example.com` domain.

```
192.0.2.12 cc-master.example.com cc-master zenoss5.cc-master
```

- 4 Save the file, and then close the editor.

Port public endpoints

If you cannot easily modify your DNS server and you do not want to modify the `/etc/hosts` file on individual clients, you can enable access to Zenoss Core and other services using an IP address and port number.

Using a port as the public endpoint, allows you to use a third-party tool or utility to access an application or service. For example, if you would like to run MySQL commands on the Mariadb, create a port-based public endpoint for the Mariadb service, and then specify the IP address and port number as needed.

Note By default, SSL is not enabled on new port public endpoints. To use a port public endpoint for Zenoss Core, Zenoss strongly recommends that a reverse proxy with SSL enabled is placed in front of Zenoss Core.

You can specify any unused port number between 1025-65535, except for ports 5000 and 8080, which are reserved ports. If necessary, contact your network administrator to assign a port number and allow traffic to and from the port.

For information on how to create a port-based public endpoint, refer to [Creating a public endpoint \(Optional\)](#) on page 21.

Creating a public endpoint (Optional)

A public endpoint provides an entry point to applications and services.

This optional procedure describes how to create a new public endpoint (either virtual hostname or port number) to access services. For information on how to create a port public endpoint for Zenoss Core, refer [Creating a port endpoint for Zenoss Core \(Optional\)](#) on page 22

To create a new public endpoint:

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click **Zenoss.core**.

The application's **Public Endpoints** are displayed:

The screenshot shows the Control Center web interface. The top navigation bar includes 'Applications', 'Resource Pools', 'Hosts', 'Logs', 'Backup / Restore', and user information 'ccuser 2'. The main content area is titled 'Zenoss.core (v5.1.1)' and includes buttons for 'Edit Service', 'Edit Variables', 'Start', 'Stop', and 'Restart'. Below this, the 'Public Endpoints' section is displayed with a table and an 'Add Public Endpoint' button.

Service	Endpoint	Type	URL	Actions
HMester	hbase-masterinfo-1	vhost	https://hbase-ip-10-111-23-202.zenoss.loc	Start Stop Delete
opentsdb	opentsdb-reader	vhost	https://opentsdb-ip-10-111-23-202.zenoss.loc	Start Stop Delete
RabbitMQ	rabbitmq_admin	vhost	https://rabbitmq-ip-10-111-23-202.zenoss.loc	Start Stop Delete
Zenoss.core	zproxy	vhost	https://zenoss5-ip-10-111-23-202.zenoss.loc	Start Stop Delete

At the bottom right of the table, it says 'Last Update: a few seconds ago Showing 4 Results'.

- 3 Click the **+ Add Public Endpoint** button, located on the right side of the table.

The Add Public Endpoint dialog is displayed:

- 4 Define the new endpoint.
 - a Click an endpoint **Type**:
 - **VHost** to create a virtual host endpoint.
 - **Port** to create a port endpoint.
 - b From the **Service - Endpoint** list, select a service.
 - c Enter the hostname or port:
 - For **VHost Hostname**, enter a fully-qualified domain name for the new alias.
 - For **Port**, enter an IP address and a port number larger than 1024. If you omit the IP address or enter 0.0.0.0, the port is opened across all interfaces.

Note For Chrome browsers, see [Unsafe ports on Chrome](#) for a list of unsafe ports that should not be used.

If you enter a port number that is used by another resource, a conflict will occur and the endpoint will not start.

- d Click **Add Public Endpoint**.

The new public endpoint is enabled and added to the Public Endpoints table. If the service is currently running, it is restarted.

Note You cannot edit the new endpoint from the Control Center user interface; however you can delete the endpoint and create a new one.

Creating a port endpoint for Zenoss Core (Optional)

This optional procedure describes how to create a port endpoint for Zenoss Core. As part of this procedure you will disable HTTPS in the `zope.conf` file.

Note By default, SSL is not enabled on new port endpoints. To use a port endpoint for Zenoss Core, Zenoss strongly recommends that a reverse proxy with SSL enabled is placed in front of Zenoss Core.

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click on **Zenoss.core**.

The application's **Public Endpoints** are displayed:

The screenshot shows the Zenoss Control Center interface. The top navigation bar includes 'Applications', 'Resource Pools', 'Hosts', 'Logs', and 'Backup / Restore'. The main content area displays the 'Zenoss.core (v5.1.1)' service page. Below the service name, there is a table titled 'Public Endpoints' with columns for Service, Endpoint, Type, URL, and Actions. The table lists four endpoints: HMaster, opentsdb, RabbitMQ, and Zenoss.core. Each endpoint has a Start, Stop, and Delete action button. A '+ Add Public Endpoint' button is located at the top right of the table.

Service	Endpoint	Type	URL	Actions
HMaster	hbase-masterinfo-1	vhost	https://hbase-ip-10-111-23-202.zenoss.loc	Start Stop Delete
opentsdb	opentsdb-reader	vhost	https://opentsdb-ip-10-111-23-202.zenoss.loc	Start Stop Delete
RabbitMQ	rabbitmq_admin	vhost	https://rabbitmq-ip-10-111-23-202.zenoss.loc	Start Stop Delete
Zenoss.core	zproxy	vhost	https://zenoss5-ip-10-111-23-202.zenoss.loc	Start Stop Delete

- 3 Scroll down to **Services**, locate **User Interface**, and click **Zope**.
- 4 On the **Zope** page, locate the `/opt/zenoss/etc/zope.conf` file and click **Edit**.
- 5 In the `zope.conf` file, locate the `HTTPS ON` line and insert `"#"` at the beginning of the line, then save the file.

The screenshot shows the 'Edit Configuration - /opt/zenoss/etc/zope.conf' dialog. The configuration file content is displayed in a text area. The line `#| HTTPS ON` is highlighted with a green box, and a '#' character has been inserted at the beginning of the line, changing it to `#|#| HTTPS ON`. The dialog has 'Cancel' and 'Save' buttons at the bottom right.

```

364 # Description.
365 #   A section which allows a user to define arbitrary key-value pairs for
366 #   use as the initial CGI environment variables. This is useful
367 #   when you want to proxy requests from another web server to Zserver,
368 #   and would like Zserver's CGI environment to reflect the CGI
369 #   environment of the other web server.
370 #
371 # Default: unset
372 #
373 # Example:
374 #
375 # <cgi-environment>
376 #   HTTPS_SERVER FooBar Server 1.0
377 #   HTTPS_PORT 443
378 # </cgi-environment>
379 #
380 <cgi-environment>
381 #|#| HTTPS ON
382 </cgi-environment>
383 #
384 # Directive: dns-server
385 #
386 # Description:
387 #   Specify the IP address of your DNS server in order to cause resolved
388 #   hostnames to be written to Zope's access log. By default, Zope will
389 #   not resolve hostnames unless this is set.
390 #
391 # Default: unset
392 #
393 # Example:
394 #
395 #   dns-server 127.0.0.1
396 #
397 #
398 #
399 # Directive: ip-address

```

- 6 Return to the **Zenoss.core** page.
- 7 Click the **+ Add Public Endpoint** button.

The Add Public Endpoint dialog is displayed:

8 Define the new endpoint:

- a Click **Port** for the endpoint **Type**.
- b From the **Service - Endpoint** list, select **core - zproxy**.
- c Enter the IP address and a port number larger than 1024. If you omit the IP address or enter 0.0.0.0, the port is opened across all interfaces.

Note For Chrome browsers, see [Unsafe ports on Chrome](#) for a list of unsafe ports that should not be used.

If you enter a port number that is used by another resource, a conflict will occur and the endpoint will not start.

- d Click **Add Public Endpoint**.

The new public endpoint is enabled and added to the Public Endpoints table. If the service is currently running, it is restarted.

Note The virtual host endpoint for Zenoss Core will no longer be accessible.

You cannot edit the new endpoint from the Control Center user interface; however you can delete the endpoint and create a new one.

Removing a public endpoint

To remove a public endpoint:

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click **Zenoss.core**.
- 3 Click **Delete** for the public endpoint you want to delete.

The Remove Public Endpoint dialog is displayed:

- 4 Click **Remove and Restart Service**.

The public endpoint is removed and the service for that endpoint is restarted.

4

Setting up Zenoss Core

This section describes how to use the Zenoss Core Setup Wizard for initial system configuration.

To complete the Setup Wizard, you need the following items:

- A password for the default administrative account (`admin`).
- A username and password for one additional administrative account.
- The username and password of an account on each server that is authorized for read access to the resources you plan to monitor.

Completing the Startup Wizard

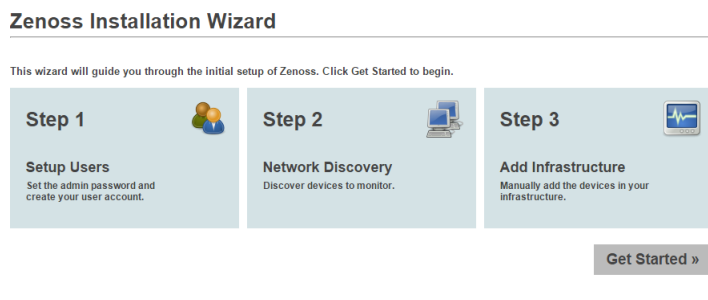
The first time you log in to Zenoss Core, you will immediately be taken to a startup wizard where you will perform the following tasks:

- Set your admin password
- Set your personal login
- Discover devices (optional)
- Add Infrastructure (optional)

Note The Setup Wizard times out after 20 minutes if you have not completed it. To start it again, close its browser window or tab, and then log in again.

- 1 Launch your Zenoss Core application the first time by clicking on the Virtual Host Name in Control Center. You will be presented with the following page showing you the initial steps to follow:

Figure 1: Setup Wizard



- 2 Click **Get Started** to begin the wizard. The Setup Users page appears.

Note When you launch Zenoss Core in the future, you will go directly to the login screen.

Setting Up Users

Follow this procedure to create a password for the admin user and create at least one additional user account.

- 1 In the **Set admin password** area, enter and confirm a password for the admin user account.

Passwords must contain a minimum of 8 characters, including one capital letter and one digit.

Step 1: Setup Users

Set admin password

The admin account has extended privileges, similar to Linux's root or Windows' Administrator. Its use should be limited to administrative tasks.

Password Must:

- Contain 8 or more characters
- Contain at least one number
- Contain at least one upper and lower case character

Admin password:

Confirm password:

Create your account

Enter information for your personal user account. You'll use this to perform most tasks.

Username:

Password:

Retype password:

Your email:

« Previous Next »

- 2 In the **Create your account** area, create one additional administrative user account name and password.
- 3 Click **Next** to go **Network Discovery**.

Discovering the Network

This procedure describes how to complete the **Network Discovery** page of the Setup Wizard. In the network discovery phase, Zenoss Core searches your network for devices to monitor based on an IP range and SSH or Windows credentials.

If you are not ready to discover devices, you can skip this page and add devices later.

Figure 2: Step 2: Network Discovery

Step 2: Network Discovery

Devices found via Discovery will be placed in the /Discovered Device Class

Networks/Range

Enter one or more networks (such as 10.0.0.0/24) or IP ranges (such as 10.0.0.1-50):

Discover

SNMP

Community Strings:

public
private

SSH Authentication

Username:

Password:

Windows Authentication

Administrator Username:

Password:

Discoveries

Status	Networks	Credentials	Duration	Job Log	Remove
Add network discoveries using the above form					

« Previous Next »

To complete the **Network Discovery** page:

- 1 For each network or IP range on which you want to discover devices, enter a network address in CIDR notation in the **Networks/Range** field. For example:

192.0.2.0/24
192.0.2.1-50

Note A /16 or /8 network can take a very long time, and may have unintended consequences.

- For **SNMP**, Zenoss Core searches both public and private community strings. This is field in informational only. You do not need to do anything in the **SNMP** area.
- In the **SSH Authentication** and **Windows Authentication** fields, enter the credentials Zenoss Core will use to discover devices.

You can enter only one set of credentials for each type of authentication.

Zenoss Core will attempt to use the same credentials on each device it discovers within the networks or IP ranges specified, but will not try to automatically classify the devices.

- Click **Discover**.

Zenoss Core begins the discovery process. The Discoveries table shows the status for each IP range.

You can view a list of discovered devices on Zenoss Core **Infrastructure** page under the **Discovered** category.

The discovery process iterates through every IP address in the networks and IP ranges you specify, adding each device that responds to a ping request. Further, the process adds information to any device that responds to an SNMP, WinRM, or SSH request.

Note Zenoss Core uses Advanced Encryption Standard (AES) with a 256-bit key size to encrypt all passwords, and stores them in the Zope object database.

The system places discovered routers in the device path /Network/Router. Devices are placed in the /Discovered device class.

- When you finished discovering your network, click **Next**.

Adding Infrastructure

The **Add Infrastructure** step is optional, as you may add devices through the **Add Infrastructure** page in Zenoss Core at any time.

Figure 3: Add Infrastructure

Step 3: Add Infrastructure

Category

- CiscoUCS
- ControlCenter
- HTTP
- KVM
- Network
- Ping
- Power
- Printer

Type

-
-
-
-
-

Connection Information

Enter multiple similar devices, separated by a comma, using either hostname or IP Address:

Manager User: admin

Manager Password: *****

Manager Port: 443

Devices

Status	Host	Credentials	Type	Duration	Job Log	Remove	Retry
Add infrastructure using the above form							

« Previous ✓ Finish

If you wish to exit the wizard and infrastructure at a later time, click **Finish**. You will then be taken to the Dashboard.

Adding Network Devices

This optional procedure is for the **Add Infrastructure** step of the Setup Wizard.

- 1 In the **Category** area, select **Network**.

Step 5: Add Infrastructure

- 2 In the **Type** list, select the product model of the switch or router to add.
The protocol used to gather data from the device is included in the list, in parentheses.

Note Some of the devices in the **Type** list, such as the Nexus 7000 and 9000 switches, represent a large number of discrete monitoring endpoints. If you are unsure whether the Zenoss Core virtual machine size you have selected supports the number of high-density devices you wish to monitor, contact your Zenoss representative.

- 3 In the **Connection Information** area, specify the devices to add. Depending on the type of network device you select, you will have different connection information fields to enter. If the field described below is not present, then it does not apply to your selection.
 - a In the **Enter multiple similar devices, separated by a comma, using either hostname or IP Address** field, enter the hostname or IP address of one or more switch or router devices on your network.
 - b In the **SNMP Community String** field, change the default (`public`) if necessary.
This field is not used if the selected device supports both SNMP and NETCONF, and you provide a user name and password.
 - c In the **Username** or **Netconf Username** field, enter the name of a user account on the device.
 - d In the **Password** or **Netconf Password** field, enter the password of the user account specified in the previous field.
 - e Click **Add**.

If you are finished adding network devices, click **Next**.

Adding Storage Devices

This option is part of step 5 of the Setup Wizard.

- 1 In the **Category** area, select **Storage**.

Step 5: Add Infrastructure

- 2 In the **Type** list, select the product model of the storage device to add.
The protocol used to gather data from the device is included in the list, in parentheses.
- 3 In the **Connection Information** area, specify the devices to add.
 - a In the **Enter multiple similar devices, separated by a comma, using either hostname or IP Addresses** field, enter the hostname or IP address of one or more storage devices on your network.

- b** Optional: In the **Username** field, enter the name of a user account on the device.
This field is not present when the device protocol is SNMP.
- c** Optional: In the **Password** field, enter the password of the user account specified in the previous field.
This field is not present when the device protocol is SNMP.
- d** Optional: In the **Port** field, enter the port at which the device listens for data collection requests.
This field is present only when the device protocol is SMIS Proxy.
- e** Check the **Use SSL?** check box to use secure communications to collect data, or uncheck the check box to use insecure communications.
This field is not present when the device protocol is SNMP.
- f** Click **Add**.

If you are finished adding storage devices, click **Next**.

Adding Server Devices

This option is part of step 5 of the Setup Wizard.

- 1 In the **Category** area, select **Servers**.

Step 5: Add Infrastructure

- 2 In the **Type** list, select the operating system and monitoring protocol of the server to add.
The protocol used to gather data from the device is included in the list, in parentheses.
- 3 In the **Connection Information** area, specify the servers to add.
 - a** In the **Enter multiple similar devices, separated by a comma, using either hostname or IP Addresses** field, enter the hostname or IP address of one or more server devices on your network.
 - b** Optional: In the **SNMP Community String** field, change the default (`public`) if necessary.
This field is only present when the device protocol is SNMP.
 - c** Optional: In the **Username** field, enter the name of a user account on the device.
This field is not present when the device protocol is SNMP.
 - d** Optional: In the **Password** field, enter the password of the user account specified in the previous field.
This field is not present when the device protocol is SNMP.
 - e** Optional: In the **AD Domain Controller** field, enter the IP address or hostname of the Active Directory Domain Controller on your network.
This field is only present when the device protocol is WinRM.
 - f** Click **Add**.

If you are finished adding server devices, click **Next**.

Adding Hypervisor Devices

This option is part of step 5 of the Setup Wizard.

- 1 In the **Category** area, select **Hypervisor**.

Step 5: Add Infrastructure

- 2 In the **Type** list, select the hypervisor service to add.
- 3 In the **Connection Information** area, specify the service to add.
 - a In the **Device Name** field, enter the name of the hypervisor service.
 - b In the **Hostname / IP Address** field, enter the hostname or IP address of the hypervisor service.
 - c In the **Username** field, enter the name of a user account on the host.
 - d In the **Password** field, enter the password of the user account specified in the previous field.
 - e Optional: Check the **Use SSL?** check box to use secure communications to collect data (recommended).
This field is only present when the device protocol is SOAP.
 - f Optional: Enter information in the **AD Domain Controller**, **Version**, **HTTP or HTTPS**, and **Port** fields.
These fields are only present when the device protocol is WinRM.
 - g Click **Add**.

If you are finished adding hypervisor devices, click **Finish**.

Adding Control Center

The Control Center is the internal application management and orchestration system for Zenoss Core. You can add Control Center as a managed resource so that you can see the internal components and their performance data.

Click **Finish** if you are done adding your devices. You can always add more devices at a later date.

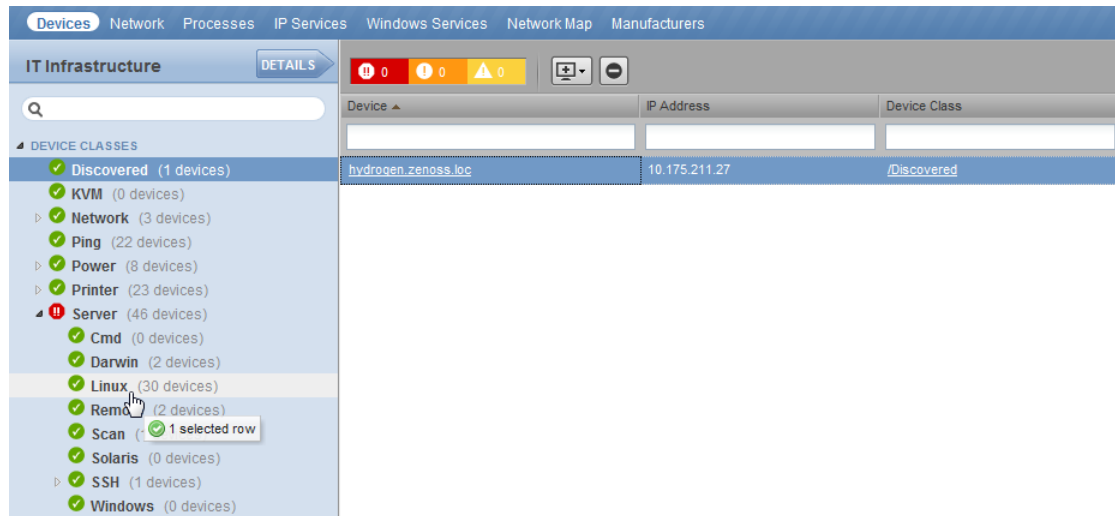
Classifying Discovered Devices

To begin monitoring discovered devices (placed in the /Discovered class, by default) you must move the devices to the appropriate device class.

Servers are organized by operating system. If the system discovers Windows devices, for example, you might choose to relocate them to /Server/Windows. Similarly, you might choose to classify discovered Linux devices in /Server/Linux (if you want to monitor and model using SNMP), or /Server/SSH/Linux (if you want to monitor and model using SSH).

To classify discovered devices:

- 1 Click **Infrastructure**, and select one or more discovered devices (highlight one or more rows) in the **Discovered** device list.
- 2 Drag the selected devices to the new device class in the tree view, for example, the **Linux** device class.

Figure 4: Classifying Discovered Devices

The Move Devices dialog appears.

- 3 Click **OK**.

The list of devices refreshes, and the devices now appear in the newly selected class.

Updating Device Authentication Details

For each device added to the database and set to its proper device class, Zenoss Core may require additional or different authentication information before it can gather device information and monitor the device.

For example, for a device in the /Server/Windows class, you must supply your Windows user name and password before the system can monitor the device. To do this:

- 1 Click a device name in the devices list.

The Device summary page appears.

- 2 Select Configuration Properties from the left panel.
- 3 Double-click the zWinRMUser configuration property to display the Edit Config Property dialog.
- 4 Enter your Windows user name in the Value field, and then click **Submit**.
- 5 Double-click the zWinRMPasswd configuration property to display the Edit Config Property dialog.
- 6 Enter your Windows password in the Value field, and then click **Submit**.

Similarly, for a device in the /Server/SSH/GenericLinux class, you must supply your SSH user name and password. Set these values in the device's zCommandUsername and zCommandPassword configuration properties.

Note After making changes, you should remodel the device to ensure the authentication changes are valid.

Note Zenoss Core uses Advanced Encryption Standard (AES) with a 256-bit key size to encrypt all passwords, and stores them in the Zope object database.

Adding or Editing Information on a Device Record

You may want to add or edit details about a device.

To add or edit information:

- 1 Click a device name in the devices list. The Device overview page appears.
- 2 You can select values to change, or click the "edit" link adjacent to a label to edit that value. Enter or change information in one or more areas, and then click **Save** to save your changes.

Modeling Devices

To model devices, the system can use:

- SSH
- WinRM
- SNMP (legacy option)

Note SSH and WinRM are the recommended options.

The modeling method you select depends on your environment, and on the types of devices you want to model and monitor.

By default the system remodels each known device every 720 minutes (12 hours).

Note You can change the frequency with which devices are remodeled. Edit the value of the Modeler Cycle Interval in the collector's configuration.

For larger deployments, modeling frequency may impact performance. In such environments, you should stop the `zenmodeler` daemon and run the modeling process once daily from a cron job.

Configuring Windows Devices to Provide Data Through SNMP

By default, Windows may not have SNMP installed. To install SNMP on your particular version of Windows, please refer to the Microsoft documentation.

After setting up and configuring the SNMP service, you must set the `zSnmpCommunity` string in Zenoss Core to match, to obtain SNMP data.

If you want processor and memory monitoring, install SNMP-Informant on the device. Go to <http://www.snmp-informant.com> and download SNMP for Windows.

To collect Windows event logs or log files from a Windows box using syslog, you can use the SyslogAgent Windows add-on, available from:

<http://syslogserver.com/syslogagent.html>

Configuring Linux Devices to Provide Data Through SNMP



To configure a Linux machine for monitoring, it must have SNMP installed. A good Linux SNMP application is net-snmp. Download, install, and configure net-snmp to then use SNMP to monitor Linux devices.

Modeling Devices Using SSH/COMMAND

You can gather additional information by running commands on the remote device and interpreting the results. This provides a more scalable and flexible way to gather information that may not be available through any other means.

Each built-in modeling command plugin is differentiated by the platform on which it runs. To determine the platform for the device you want to model, run the `uname` command in a shell on the device.

To model a device using command plugins, first add the device by using the protocol "none," and then choose the plugins you want to apply:

- 1 From the Navigation menu, select **Infrastructure**.
- 2  Click the Add Devices  icon and select **Add a Single Device** from the drop-down list. The Add a Single Device window appears.
- 3 Enter values for Name or IP and Device Class.
- 4 De-select the Model Device option.
- 5 Click **Add**.
- 6 After adding the device, select the device name in the devices list.

The Device Overview page appears.

- 7 In the left panel, select **Configuration Properties**.
- 8 If necessary, set the values of the `zCommandUsername` and `zCommandPassword` configuration properties to the user name and password of the device (or set up authentication by using RSA/DSA keys.)

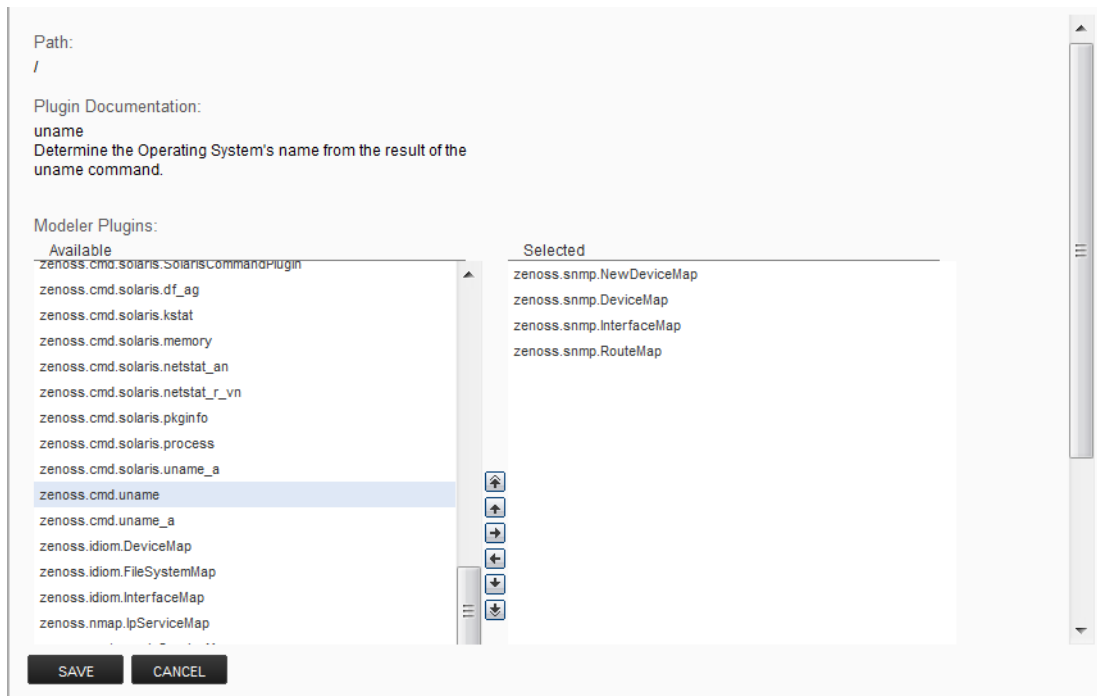
Note If using RSA keys for a device or device class, change the value of the `zKeyPath` configuration property to:

```
~/.ssh/id_rsa
```

- 9 In the left panel, select **Modeler Plugins**.

The list of plugins appears. The left column displays available plugins; the right column shows those currently selected.

- 10 Select `zenoss.cmd.uname` from the Available list, and then use the right arrow control to move it to the Selected list on the right. Use the controls to place it at the top of the list.

Figure 5: Add Plugin

- 11 Use the left arrow control to move the other Selected plugins from the Selected list to the Available list.
- 12 Click **Save**.
- 13 Model the device by clicking the **Model Device** button.

Using Device Class to Monitor Devices Using SSH

The `/Server/Command` device class is an example configuration for modeling and monitoring devices using SSH. The `zCollectorPlugins` have been modified (see the section titled "Modeling Using SSH/Command"), and the device, file system, and Ethernet interface templates used to gather data over SSH have been created. You can use this device class as a reference for your own configuration; or, if you have a device that needs to be modeled or monitored via SSH/Command, you can place it in this device class to use the pre-configured templates and configuration properties. You also must set the `zCommandUsername` and `zCommandPassword` configuration properties to the appropriate SSH login information for each device.

Using the `/Server/Scan` Device Class to Monitor with Port Scan

The `/Server/Scan` device class is an example configuration for modeling devices by using a port scan. You can use this device class as a reference for your own configuration; or, if you have a device that will use only a port scan, you can place it under this device class and remodel the device.

Modeling Devices Using Port Scan

You can model IP services by doing a port scan, using the Nmap Security Scanner (<http://nmap.org/>). You must provide the full path to your system's `nmap` command.

To determine where `nmap` is installed, at the command line, enter:

```
which nmap
```

If your system returns a result similar to:

```
/usr/bin/which: no nmap in (/opt/zenoss/bin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/opt/zenoss/bin)
```

then nmap is not installed. Install it, and then try again.

After locating the nmap command (including the directory beginning with /), enter the following as the zenoss user on the Zenoss Core server:

```
cd $ZENHOME/libexec ln -s
    Full_Path_to_nmap
```

Note In order to execute a command using `$ZENHOME` (`/opt/zenoss` for the zenoss user), you must be attached to the container holding the Zenoss Core application. See the Control Center documentation for `serviced` commands.

To model a device using a port scan:

- 1 Select the device in the device list.
- 2 In the left panel, select **Modeler Plugins**.
- 3 Select the `zenoss.nmap.ipServiceMap` plugin in the list of Available plugins, and then use the right arrow control to move it to the list of Selected plugins.
- 4 Click **Save**.
- 5 Remodel the device by clicking the **Model Device** button.

About Modeler Plugins

Zenoss Core uses plug-in maps to map real world information into the standard model. Input to the plug-ins can come from SNMP, SSH or Telnet. Selection of plug-ins to run against a device is done by matching the plug-in name against the `zCollectorPlugins` configuration property.

- **DeviceMap**– Collects basic information about a device, such as its OS type and hardware model.
- **InterfaceMap**– Collects the list of network interfaces on a device.
- **RouteMap**– Collects the network routing table from the device.
- **IpServicesMap**– Collects the IP services running on the device.
- **FileSystemMap**– Collects the list of file systems on a device.

Viewing and Editing Modeler Plugins for a Device

Plugins are controlled by regular expressions that match their names. To view a list of plugins for any device:

- 1 Click the device name in the devices list.
The Device summary page appears.
- 2 In the left panel, select **Modeler Plugins**.
The Modeler Plugins page appears.

Adding Plugins

To add a plugin to a device:

- 1 Use the right arrow control to move one or more plugins from the Available list (on the left) to the Selected list (on the right).
- 2 Click **Save**.

Reordering Plugins

Plugins run in the order in which they are listed. To re-order plugins, use the up and down arrow controls, and then click **Save**.

Deleting Plugins from a Device

To delete a plugin from a device, use the left arrow control to move the plugin from the Selected list to the Available list.

Debugging the Modeling Process

You can run the modeler from the command line against a single device. This feature is useful when debugging issues with a plugin.

By passing the `--collect` command to the modeler, you can control which modeler plugins are used. For example, the following command runs only the interface plugin against the `build.zenoss.loc` device:

- 1 Login into the Control Center host as a user with `serviced` CLI privileges.
- 2 Attach to the `zenmodeler` service.

```
serviced service attach zenmodeler
```

- 3 Change to the `zenoss` user.
- 4 Run the `zenmodeler` command.

```
$ zenmodeler run -v10 --collect=IpInterface -d build.zenoss.loc
```

If the command returns any stack traces, check the community forums for assistance.

Next Steps

Zenoss Core is now configured to monitor your IT infrastructure. Your next steps in the monitoring process may include some of the following items:

- Customize the Dashboard
- Review events in the Events Console
- Organize devices and infrastructure in to logical groupings on the Infrastructure page
- View data collection graphs from the Infrastructure page
- Generate and review reports on the Reports page
- Refine data collection on the Advanced page

For more information on these and other procedures, refer to the *Zenoss Core Administration Guide*.

A

External HBase configuration

Zenoss Core may be configured to use an external HBase cluster, rather than the cluster that is included in the application.

Note If you do not already have an external HBase cluster, there is no need to create one. The procedures in this section are for customers who wish to use an existing HBase cluster for Zenoss Core data.

The version of HBase installed in your external HBase cluster must be compatible with the version of OpenTSDB used by the Zenoss Core application. For Zenoss Core versions 5.0.1 through 5.1.2, the required version of HBase is 0.92.

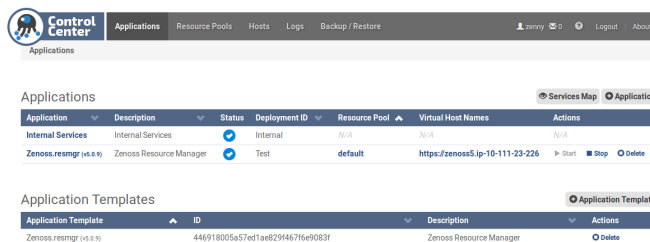
Perform the procedures in the following sections in order.

Configuring OpenTSDB for an external HBase cluster

To perform this procedure, install and start Zenoss Core.

This procedure configures OpenTSDB to use an external HBase cluster, rather than the HBase cluster that is included in the Zenoss Core application.

- 1 Log in to the Control Center browser interface.



- 2 In the **Applications** table, click **Zenoss.resmgr**.
- 3 Scroll down to the **Services** table, and then locate the OpenTSDB service.
Depending on your version of Control Center, the service is either **opentsdb** or two separate services, **reader** and **writer**.
- 4 Click **opentsdb**, or one of **reader** or **writer**.
- 5 On the service details page, scroll down to the **Configuration Files** table.

The screenshot shows the Zenoss Control Center interface for the 'reader' service. At the top, there are navigation tabs: Applications, Resource Pools, Hosts, Logs, Backup / Restore, and user information. Below the navigation, there are buttons for 'Edit Service', 'Edit Variables', 'Start', 'Stop', and 'Restart'. The 'Virtual Host Names' section contains a table with one entry:

Virtual Host Name	Service	Endpoint	URL	Actions
opentsdb	reader	opentsdb-reader	https://opentsdb.ip-10-111-23-226	Delete

The 'IP Assignments' section shows 'No Data Found'. The 'Configuration Files' section shows a table with one entry:

Path	Actions
/opt/zenoss/etc/opentsdb/opentsdb.conf	Edit

- 6 In the **Actions** column of the **Configuration Files** table, click **Edit**.
- 7 In the **Edit Configuration** dialog, replace the value of the `tsd.storage.hbase.zk_quorum` key with the ZooKeeper quorum of the external HBase cluster.
 - a Delete the existing value.

The default value is a *Go language template* expression.
 - b Specify the ZooKeeper quorum of the external HBase cluster.

To specify a ZooKeeper quorum, create a comma-separated list of all quorum members. Specify each member of the quorum with a hostname or IP address, the colon character (:), and then the port number on which the ZooKeeper service is listening.

Note If you use hostnames, the Control Center master host must be able to resolve them to IPv4 addresses, either through a nameserver on the network or through entries in `/etc/hosts`.

The following example shows the correct syntax for a 3-member ZooKeeper quorum:

```
zk-1.example.com:2181, zk-2.example.com:2181, zk-3.example.com:2181
```

- 8 In the **Edit Configuration** dialog, click **Save**.
- 9 At the top of the page, click the **Stop** button, and then click the **Start** button.
- 10 If your version of Zenoss Core includes two OpenTSDB services (**reader** and **writer**) repeat the preceding steps for the other service.

Configuring the OpenTSDB service startup command

This procedure configures the OpenTSDB service to use the external HBase cluster on startup.

- 1 Log in to the Control Center browser interface.
- 2 In the **Applications** table, click **Zenoss.resmgr**.
- 3 Scroll down to the **Services** table, and then locate the OpenTSDB service.

Depending on your version of Control Center, the service is either **opentsdb** or two separate services, **reader** and **writer**.
- 4 Click **opentsdb**, or one of **reader** or **writer**.
- 5 On the service details page, click the **Edit Service** link at the top of the page.

This screenshot is identical to the one above, showing the Zenoss Control Center interface for the 'reader' service. It includes the same navigation tabs, buttons, and tables for Virtual Host Names, IP Assignments, and Configuration Files.

- 6 In the **Edit Service** dialog, change the value of the **Startup Command** field.

- a Delete the *Go language template* expression.

The expression is everything after `start-opentsdb.sh`.

- b Specify the ZooKeeper quorum of the external HBase cluster.

To specify a ZooKeeper quorum, create a comma-separated list of all quorum members. Specify each member of the quorum with a hostname or IP address, the colon character (:), and then the port number on which the ZooKeeper service is listening.

Note If you use hostnames, the Control Center master host must be able to resolve them to IPv4 addresses, either through a nameserver on the network or through entries in `/etc/hosts`.

The following example shows the correct syntax for a 3-member ZooKeeper quorum:

```
zk-1.example.com:2181, zk-2.example.com:2181, zk-3.example.com:2181
```

The result should include at least one empty space between `start-opentsdb.sh` and the ZooKeeper quorum list.

- 7 In the **Edit Service** dialog, click **Save Changes**.
- 8 At the top of the page, click the **Stop** button, and then click the **Start** button.
- 9 If your version of Zenoss Core includes two OpenTSDB services (**reader** and **writer**) repeat the preceding steps for the other service.

Disabling the Zenoss Core HBase cluster

This procedure disables the HBase cluster that is included in the Zenoss Core application.

- 1 Log in to the Control Center master host as `root`, or as a user with superuser privileges.
- 2 Stop the Zenoss Core HBase cluster.

```
serviced service stop HBase
```

- 3 Disable automatic start of the HBase services.

- a Change the configuration of each service.

```
for svc in hmaster regionserver zookeeper
do
    serviced service list $svc | \
    sed -e 's/"Launch": "auto"/"Launch": "manual"/' | \
    serviced service edit $svc
done
```

The `serviced` command displays the new configuration after each edit.

- b Verify that each service is set to manual start.

```
for svc in hmaster regionserver zookeeper
do
    serviced service list $svc | egrep '"Launch":'
done
```

- 4 Remove the OpenTSDB prerequisite for the Zenoss Core HBase cluster.

Depending on your version of Control Center, the OpenTSDB service is either `opentsdb` or two separate services, `reader` and `writer`.

- a Edit `opentsdb`, or one of `reader` or `writer`.

```
serviced service edit reader
```

The `serviced` command opens the service's configuration in the default text editor.

- b** Locate the `Prereqs` section, and then remove everything between the left square bracket (`[`) and the right square bracket (`]`) characters.

The following lines show an example `Prereqs` section:

```
"Prereqs": [  
  {  
    "Name": "HBase Regionserver up",  
    "Script": "{{with $rss := (child (child (parent) \"HBase  
\"))}.Instances }}"  
  }  
],
```

After editing, the section should look like the following example:

```
"Prereqs": [],
```

- c** Save the file, and then exit the text editor.
- d** If your version of Zenoss Core includes two OpenTSDB services (**reader** and **writer**) repeat the preceding substeps for the other service.