



The Zenoss Enablement Series:

Introduction to Zenoss Control Center

Document Version 500 p1

Zenoss, Inc.

www.zenoss.com

Copyright © 2015 Zenoss, Inc. 11305 Four Points Drive, Bldg 1 - Suite 300, Austin, Texas 78726, U.S.A.
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.

Cisco, Cisco UCS, Cisco Unified Computing System, Cisco Catalyst, and Cisco Nexus are trademarks or registered trademarks of Cisco and/or its affiliates in the United States and certain other countries.

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.

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.

vSphere is a 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.

Table of Contents

- Introduction to Zenoss Control Center 1**
- Control Center Concepts 1**
 - Resource Pool 1
 - Host 2
 - Service 2
 - Internal Services 2
 - Application 4
 - Service Definition 4
 - Application Template 4
 - Public Endpoints 4
- Resource Manager Public Endpoints 5**

Introduction to Zenoss Control Center

The Zenoss Control Center is a custom out-of-band management platform written and open-sourced by Zenoss. Control Center makes administration of the Resource Manager platform easier and more efficient.

Control Center is written in the Go programming language. It provides a central platform to orchestrate the Resource Manager distributed application. Control Center is built on top of the *Docker* platform (<https://www.docker.com/>) and leverages other open-source software.

Control Center is designed to simplify the installation, deployment and management of Zenoss Service Dynamics. It includes the Elastic product called *Kibana* (<https://www.elastic.co/products/kibana>). Kibana provides a user interface that enables you to search for, filter and display log messages from any service, regardless of the host on which it is running.

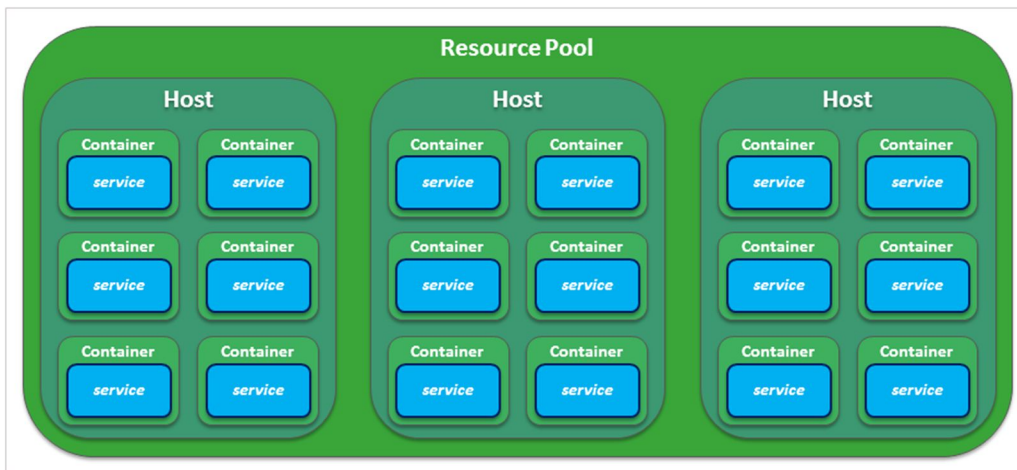
Control Center Concepts

Control Center includes the following core concepts, described below:

- [Resource Pool](#)
- [Host](#)
- [Service](#)
- [Internal Services](#)
- [Application](#)
- [Service Definition](#)
- [Application Template](#)
- [Public Endpoints](#)

Resource Pool

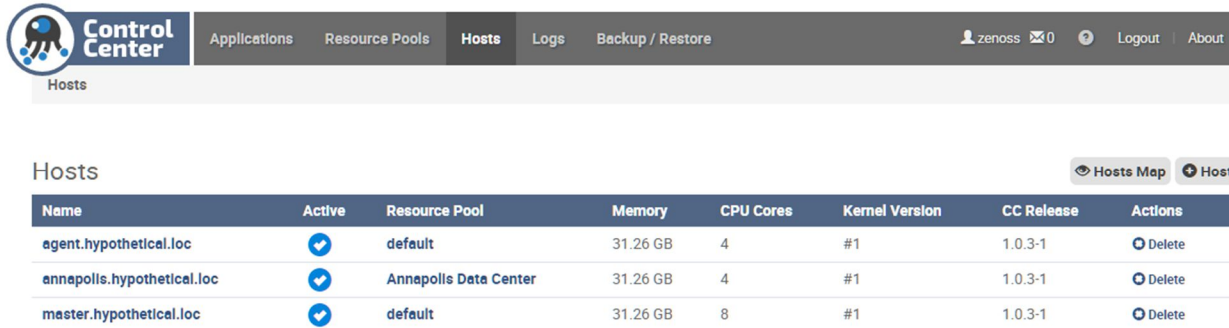
A resource pool is a collection of Hosts, their Containers, and associated services. For example, the following diagram shows three Hosts within a single Resource Pool:



A resource pool can be used to restrict the set of hosts where a service is run. This is typically done to ensure that a collector runs on the same network as the devices it monitors.

Host

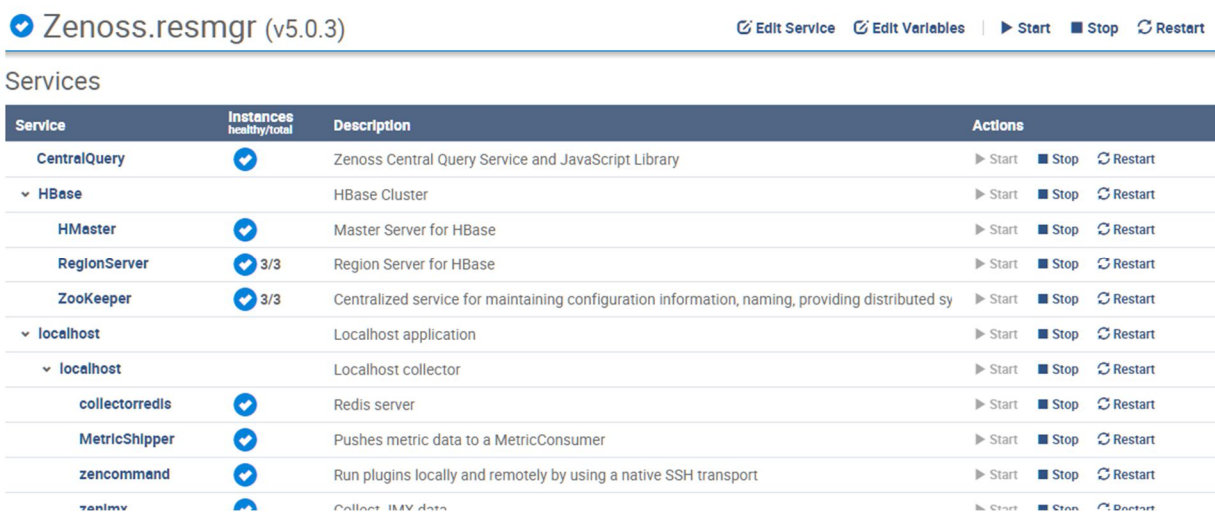
In the context of Control Center, a *Host* is a compute node, either physical or virtual, that runs services as part of a Resource Pool. The Host runs the `serviced` daemon in the Node Controller (Agent) or Master mode. When you add a Host to Control Center, you assign it to a particular Resource Pool. For example:



Name	Active	Resource Pool	Memory	CPU Cores	Kernel Version	CC Release	Actions
agent.hypothetical.loc	✓	default	31.26 GB	4	#1	1.0.3-1	⊙ Delete
annapolis.hypothetical.loc	✓	Annapolis Data Center	31.26 GB	4	#1	1.0.3-1	⊙ Delete
master.hypothetical.loc	✓	default	31.26 GB	8	#1	1.0.3-1	⊙ Delete

Service

A *Service* is one or more processes, with their supporting files, that are run within a single container to provide specific functionality as part of an application. Many Resource Manager services support multiple instances, each running in its own container. For example:



Service	Instances healthy/total	Description	Actions
CentralQuery	✓	Zenoss Central Query Service and JavaScript Library	▶ Start ■ Stop ⌛ Restart
▼ HBase		HBase Cluster	▶ Start ■ Stop ⌛ Restart
HMaster	✓	Master Server for HBase	▶ Start ■ Stop ⌛ Restart
RegionServer	✓ 3/3	Region Server for HBase	▶ Start ■ Stop ⌛ Restart
ZooKeeper	✓ 3/3	Centralized service for maintaining configuration information, naming, providing distributed sy	▶ Start ■ Stop ⌛ Restart
▼ localhost		Localhost application	▶ Start ■ Stop ⌛ Restart
▼ localhost		Localhost collector	▶ Start ■ Stop ⌛ Restart
collectorredis	✓	Redis server	▶ Start ■ Stop ⌛ Restart
MetricShipper	✓	Pushes metric data to a MetricConsumer	▶ Start ■ Stop ⌛ Restart
zencommand	✓	Run plugins locally and remotely by using a native SSH transport	▶ Start ■ Stop ⌛ Restart
zenimv	✓	Collect IMV data	▶ Start ■ Stop ⌛ Restart

Internal Services

Internal Services are essential support services required for Control Center to function, regardless of the applications deployed on it.

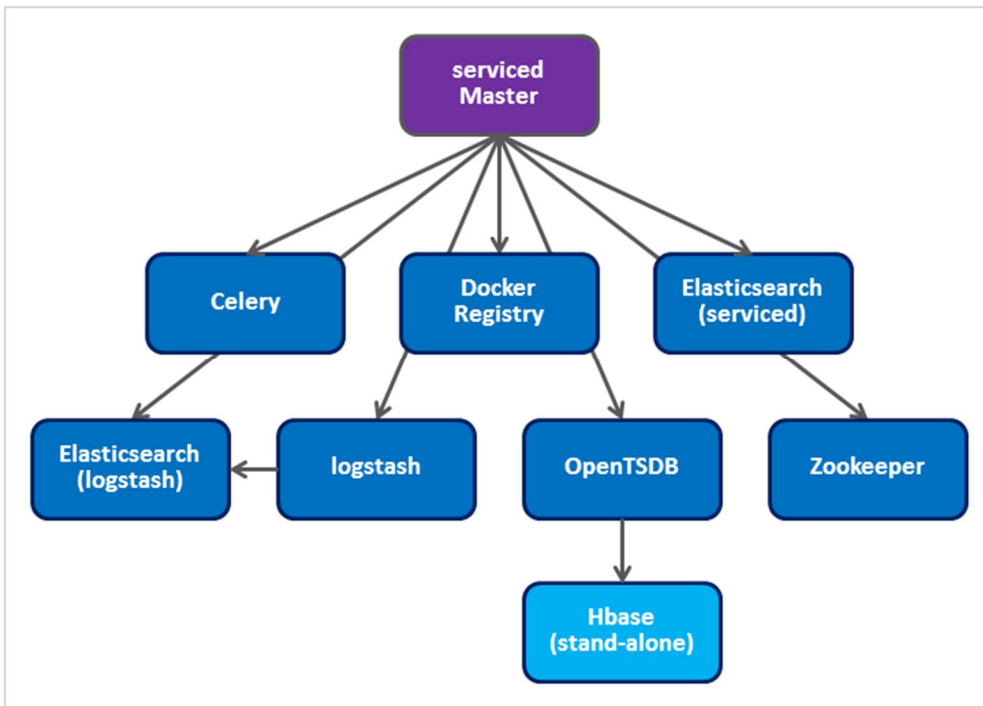
The Control Center Internal Services include:

- **Celery** – An asynchronous task queue and scheduler.
- **Docker Registry** – A central repository of Docker images in a multi-host environment.

Zenoss, Inc.

- **Elasticsearch** (2 instances) –
A distributed, real-time search and analytics engine. Control Center uses it to perform the following:
 - 1) index log files
 - 2) store service definitions
- **HBase** -
An internal service instance that runs within the OpenTSDB container and stores Control Center service metrics. Note that it does not store device performance metrics.
- **Logstash** –
A log file collector and aggregator that can forward parsed log file entries to Elasticsearch.
- **OpenTSDB** –
A Time Series Database that runs on top of HBase to store and search Control Center metrics.
- **Zookeeper** –
A centralized service that maintains distributed application configuration information.

The following diagram shows the Control Center Internal Services:



Note: Internal Services are not associated with any particular application.

The following diagram shows Internal Services displayed in the Control Center UI:

The screenshot shows the Control Center UI for the 'Internal Services' application. The navigation bar includes 'Applications', 'Resource Pools', 'Hosts', 'Logs', and 'Backup / Restore'. The user is logged in as 'zenoss'. The main content area shows a table of services:

Service	Instances Healthy/Total	Description
Celery	1/1	Internal Celery
Docker Registry	1/1	Internal Docker Registry
Elastic Search - LogStash	1/1	Internal Elastic Search - LogStash
Elastic Search - Serviced	1/1	Internal Elastic Search - Serviced
Logstash	1/1	Internal Logstash
OpenTSDB	1/1	Internal Open TSDB
Zookeeper	1/1	Internal ZooKeeper

Application

Within Control Center, an *Application* refers to a set of services that act together to provide a high-level service. Zenoss applications include Resource Manager and Analytics. Note that Service Impact runs as an optional service within the Resource Manager application instead of running as a top-level Control Center application.

An *Application Template* defines an application (see *Application Template*, below).

Applications are assigned to a base *Resource Pool* when deployed in Control Center. Services within the application can be configured to run in other Resource Pools. Configuring services to run in other resource pools is usually done to ensure that a Collector runs on the same network as the devices it monitors.

Service Definition

A *Service Definition* is a formal description of how an individual service is configured and run.

Application Template

An *Application Template* is a hierarchical set of service definitions that constitute a Control Center Application.

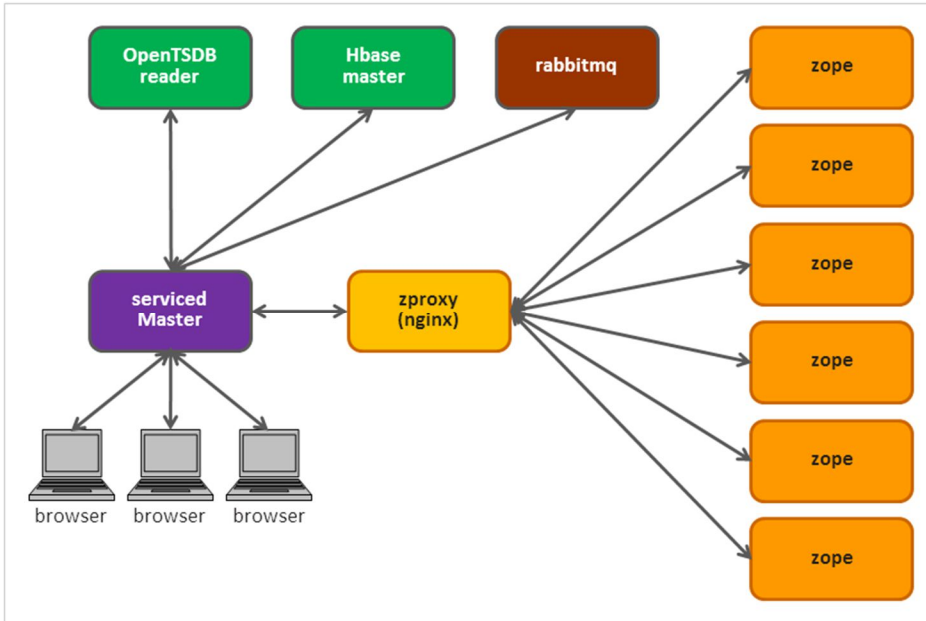
Public Endpoints

Within Control Center, a *Public Endpoint* is an URL that is interpreted and proxied by `serviced`. A Public Endpoint provides access to the web interface of an individual service within an application.

NOTE: Public Endpoints were called *Virtual Hostnames* in versions previous to version 5.1.1 of Control Center.

Zenoss, Inc.

You can access the following service user interfaces using a Resource Manager Public Endpoint:



The Public Endpoints include the Resource Manager UI:

[Zenoss.resmgr \(v5.1.1\)](#) Edit Service Edit Variables Start Stop Restart

Zenoss Resource Manager

Public Endpoints Add Public Endpoint

Service	Endpoint	Type	URL	Actions
HMaster	hbase-masterinfo-1	vhost	https://hbase.pre-train2.zenoss.com	Start Stop Delete
reader	opentsdb-reader	vhost	https://opentsdb.pre-train2.zenoss.com	Start Stop Delete
RabbitMQ	rabbitmq_admin	vhost	https://rabbitmq.pre-train2.zenoss.com	Start Stop Delete
Zenoss.resmgr	zproxy	vhost	https://zenoss5.pre-train2.zenoss.com	Start Stop Delete

Last Update: a few seconds ago Showing 4 Results

Resource Manager Public Endpoints

By default, Resource Manager defines four *Public Endpoints*. The virtual names/URLs are based on the fully-qualified host name of your Control Center master server. For example, a master server host name of **master.hypothetical.loc** results in the following Public Endpoints:

URL	Description
https://zenoss5.master.hypothetical.loc	Resource Manager web UI
https://hbase.master.hypothetical.loc	HBase management
https://opentsdb.master.hypothetical.loc	Time Series Database data point graphing
https://rabbitmq.master.hypothetical.loc	RabbitMQ reporting and management

Rabbitmq, OpenTSB and others can be accessed through *virtual hostnames*, or (in Control Center version 5.1.1 or later) a *port-based Public Endpoints*. This means:

- **Host Public Endpoints** are accessible by *hostname*, for example:
`https://zenoss.mckraken`
- **Port Public Endpoints** are accessible by *ip:port* or *hostname:port*, for example:
`myhost:54321` or `10.87.1.100:54321`

Note: Public Endpoints or virtual names must all resolve to the same IP address as the Control Center master host.