

Hitachi Content Platform for Cloud Scale

System Management Guide

© 2017 -2019 Hitachi Vantara Corporation. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara Corporation (collectively, "Hitachi"). Licensee may make copies of the Materials, provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video, and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara Corporation at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara Corporation.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API, or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, IBM, Lotus, MVS, OS/390, PowerPC, RS6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or web site are properties of their respective owners.



Contents

Chapter 1: Services	9
Services Overview	9
Service categories	9
Applications	9
Service instances	10
Floating services	10
Networking	10
Storage for services	10
Service list	11
Viewing services	36
Viewing all services	37
Viewing individual service status	38
Managing services	39
Moving and scaling services	39
Configuring service settings	43
Repairing services	43
Chapter 2: Jobs	45
Job types	45
Storage for jobs	45
Networking	45
Managing jobs	46
Configuring where jobs run	46
Configuring job settings	48
Scheduling jobs	49

Chapter 3: Monitoring	51
Monitoring instances	51
Viewing all instances	52
Viewing the services running on an instance	54
Monitoring services	55
Viewing all services	56
Viewing individual service status	57
Monitoring jobs	58
Monitoring all jobs	58
Monitoring job types	59
Monitoring individual jobs	60
Monitoring processes	62
Monitoring service operations	62
Monitoring system processes	62
System events	63
Alerts	64
List of alerts	65
Creating email notification rules	68
SMTP settings	69
Message settings	70
Recipient settings	70
Creating syslog notification rules	71
Syslog settings	71
Message settings	72
Severity filter	72
Logs and diagnostic information	73
Chapter 4: Security	79
Granting access to users	79
Setting the session timeout limit	79
Identity providers	80
Adding identity providers	80
Identity provider configuration settings	81
User information caching	85
Viewing identity providers	85

Deleting identity providers	86
Groups	86
Adding groups	86
Viewing groups	87
Assigning roles to groups	87
Deleting groups	88
Roles	88
Creating roles	88
Viewing roles	90
Editing roles	90
Deleting roles	91
Permissions	92
Changing the admin account password	95
Certificates	96
Viewing installed certificates	96
Adding data source certificates	97
Changing the system certificate	97
System certificate considerations	98
Installing a certificate you created	98
Installing a new self-signed certificate	99
Creating a CSR and installing the returned certificate	100
Chapter 5: System management	103
Setting the system hostname	103
System scaling	106
Networking	106
Instance IP address requirements	107
Network types	107
Allowing access to external resources	107
Ports	108
Handling network changes	113
Volumes	114
Example	115
Creating and managing volumes	116
Volume drivers	116

Viewing volumes	117
Instances	118
About master and worker instances	118
Single-instance systems versus multi-instance systems	118
Requirements for running system instances	120
Hardware resources	120
Operating system and Docker requirements	121
Docker considerations	122
SELinux considerations	123
Supported browsers	123
Time source requirements	123
Adding new instances	123
Viewing instances	132
Removing instances	137
Replacing a failed instance	139
Plugins	140
Installing new plugin bundles	140
Viewing installed plugins	141
Upgrading plugin bundles	142
Setting the active plugin bundle version	142
Deleting plugin bundles	143
Packages	143
Exporting packages	144
Importing packages	145
Setting a login welcome message	146
Updating the system	146
Before updating	147
During an update	147
Checking update status	147
Results of an update	148
Update errors	148
Update history	148
New services and components added during an update	148
Viewing update history	149
Uninstalling the system	150

Chapter 6: Best practices	151
Best practices for distributing services	151
Best practices for maintaining system availability	152
Run master instances on separate physical hardware	152
Run services on more than one instance	152
Chapter 7: Reference	153
Troubleshooting	153
System event issues	153
CLI reference	153
Accessing the CLI tools on a system instance	154
Accessing the CLI tools from your computer	154
Syntax	155
Viewing available commands	156
Viewing request models	157
Editing configuration preferences	159
System error responses	161
REST API reference	163
Input and output formats	163
Access and authentication	164
Viewing and using REST API methods	166
Error responses	168

Chapter 1: Services

This section describes the services of HCP for cloud scale as well as how to view, configure, move, and repair services.

Services Overview

Services perform functions essential to the health or functionality of the system. For example, the Metrics service stores and manages system events, while the Watchdog service ensures that other services remain running. Internally, services run in Docker containers on the instances in the system.

Service categories

Services are grouped into these categories depending on what actions they perform:

- **Services** – Enable product functionality. You can scale, move, and reconfigure these services.
- **System services** – Maintain the health and availability of the system. You cannot scale, move, or reconfigure these services.

Some System services run only on master instances. See "[About master and worker instances](#)" on page 118.

Applications

Some services are classified as applications. These are the services with which users interact. Services that are not applications typically interact only with other services.

Service instances

Services run on instances in the system. Most services can run simultaneously on multiple instances. That is, you can have multiple instances of a service running on multiple instances in the system. Some services run on only one instance.

Each service has a recommended and required number of instances on which it should run. For information, see "[Service list](#)" on the facing page.

You can configure where Product services run, but not System services.

Floating services

If a service supports **floating**, you have flexibility in configuring where new instances of that service are started when service instances fail.

Non-floating (or **persistent**) services run on the specific instances that you specify. If one of those service instances fails, the system does not automatically bring up a new instance of that service on another system instance.

With a service that supports floating, you specify a pool of eligible system instances and the number of service instances that should be running at any time. If a service instance fails, the system brings up another on one of the system instances in the pool that doesn't already have an instance of that service running.

For services with multiple types, the ability to float can be supported on a per-type basis.

For information on changing the instance pool for a floating service, see "[Moving and scaling services](#)" on page 39.

Networking

Each service binds to a number of ports and to one type of network, either internal or external. Networking for each service is configured during system installation and cannot be changed once a system is running. For more information, see "[Networking](#)" on page 106.

Storage for services

Services can use volumes for storing data. For more information, see "[Volumes](#)" on page 114.

Service list

The table below describes the services that your system runs. Each service runs within its own Docker container. For each service, the table lists:

- **Configuration settings:** The settings you can configure for the service. For information on configuration, see "[Configuring service settings](#)" on page 43.
- **RAM needed per instance:** The amount of RAM that, by default, the service needs on each instance on which it's deployed. For all services except for System services, this value is also the default Docker **Container Memory** value for the service.
- **Number of instances:** Shows both:
 - The required number of instances on which a service must run for the system to function properly.
 - The recommended number of instances that you should run a service on. These are recommended minimums; if your system includes more instances, you should take advantage of them by running services on them.
- Whether the service is persistent (that is, it must run on a specific instance) or supports floating (that is, it can run on any instance).
- Whether the service is scalable or not.



Note: For system services, you cannot set the **Container Memory** size larger than the **Max Heap Size** setting. For other services, you should not set the **Container Memory** size larger than the **Max Heap Size** setting.

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
Product services: These services perform system functions. You can move and reconfigure these services.		

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Cassandra Decentralized database that can be scaled across large numbers of hardware nodes.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 2.4 GB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	2.4 GB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	<p>Service Options</p> <ul style="list-style-type: none"> • Max heap: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1800m. • Heap new size: Heap size for the Java virtual machine. Valid values are integers representing a number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	Number of instances:	Required: 3 Recommended: All
	<p>Advanced Options:</p> <p>Compaction Frequency: How often database compaction operations are run. The options are Weekly (default) and Daily.</p> <p>Caution: Changing this setting can negatively affect the service. Use with caution.</p>	Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Chronos Job scheduling.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 712 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	712 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Heap settings: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	Number of instances:	<ul style="list-style-type: none"> • 1 required • 1 recommended
Persistent or floating?		Floating	
Supports volume configuration?		Yes	
Single or multiple types?		Single	
Scalable?		No	

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Elasticsearch Data indexing and search platform.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 2 GB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	2 GB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	<p>Service Options</p> <ul style="list-style-type: none"> • Heap settings: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB. • Days to keep logs: The number of days to keep service logs, including access and metrics indexes. The default is 30 days. • Index Protection Level: The number of additional replicas (copies) to keep of each index file (shard). Replicas are kept on separate instances. You can set this value for every shard. The default is 1 replica (which means that two copies are kept). The maximum is the number of instances less one. 	Number of instances:	Required: 3 Recommended: All
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Kafka Stream processing platform for handling real-time data streams.	Container Options: Default <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 2 GB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	2 GB
	Service Options <ul style="list-style-type: none"> • Heap settings: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is -Xmx1800m-Xms512m. 	Number of instances:	Required: 3 Recommended: All
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Logstash Collection engine for event data. Can perform transformations on the data it collects and then send that data to a number of outputs.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 700 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	700 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max heap: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
MAPI Gateway Serves MAPI endpoints.	Container Options: Default <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	Service Options <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 64 MB. 	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Metadata Cache Cache for system metadata.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Metadata Coordination Coordinates the different metadata gateway services, and coordinates metadata partitions.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 64 MB. 	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Metadata Gateway Stores and protects metadata and serves it to other services.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 4096 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 2048 MB. 	Number of instances:	Required: 3 Recommended: all
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Metadata Policy Engine Executes asynchronous metadata updates.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 2048 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB. 	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Metrics Data indexing and search platform.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	<p>Service Options</p> <ul style="list-style-type: none"> • Prometheus Scrape Interval: The time interval between runs of the metrics collection task. Enter an integer number of seconds. You can optionally specify the suffix s (seconds). The default is 10 seconds. • Prometheus Database Path: Storage location for prometheus local time-series db. Enter a path. The default is tsdb/. • Prometheus Database Retention: The number of days to retain files. Enter an integer number of days. You can optionally specify the suffix d (days). The default is 15 days. 	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>S3 Gateway Serves S3 API endpoints and communicates with storage components.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Max Heap Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	Number of instances:	Required: 1 Recommended: All

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	<p>HTTP Options:</p> <ul style="list-style-type: none"> • Enable HTTP: Select to enable HTTP connections. • Max Http Request Headers: The maximum number of HTTP request headers to allow. Enter an integer. The default is 100 request headers. 	Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Tracing Agent Provides end-to-end distributed tracing for S3 API calls and MAPI calls.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • Collector TChannel Hostname: Enter a host name. The default is localhost. • Collector TChannel Port: Enter a port number. The default is 14267. 	Number of instances:	Required: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Tracing Collector Provides end-to-end distributed tracing for S3 API calls and MAPI calls.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • ElasticSearch Hostname: Enter a host name. The default is localhost. • ElasticSearch Port: Enter a port number. The default is 9200. • Sampling Rate: The sampling rate for all clients implementing remote sampling. Enter a number between 0 and 1 inclusive. The default is 1. 	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Tracing Query Provides end-to-end distributed tracing for S3 API calls and MAPI calls.</p>	<p>Container Options: Default</p> <ul style="list-style-type: none"> • Container Memory: The hard memory limit for the service's Docker container, in MB. The default is 768 MB. • CPU: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1. 	RAM needed per instance:	768 MB
	<p>Service Options</p> <ul style="list-style-type: none"> • ElasticSearch Hostname: Enter a host name. The default is localhost. • ElasticSearch Port: Enter a port number. The default is 9200. 	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

System services: These services manage system resources and ensure that the system remains available and accessible. These services cannot be moved or reconfigured.

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
System Management application The system management application.	Service Options <ul style="list-style-type: none"> • Max heap size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Cluster Coordination Manages hardware resource allocation.	N/A	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service list

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Cluster Worker Receives and performs work from other services.	N/A	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Network Proxy Network request load balancer.	Security Protocol: Select which TLS versions to use: <ul style="list-style-type: none"> • TLS 1.0 • TLS 1.1 • TLS 1.2 • TLS 1.3 	RAM needed per instance:	N/A
	SSL Ciphers: If you want to provide your own cipher suite, enter it here.	Number of instances:	N/A
	Custom Global Configuration: Select Enable Advanced Global Configuration to enable adding your own parameters to the HAProxy "global" section.	Persistent or floating?	Persistent
	Custom Defaults Configuration: Select Enable Defaults Configuration to enable adding your own parameters to the HAProxy "global" section.	Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
<p>Sentinel Runs internal system processes and monitors the health of the other services.</p>	<p>Service Options</p> <ul style="list-style-type: none"> • Max heap size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 256 MB. 	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
<p>Service Deployment Handles deployment of high-level services (that is, the services that you can configure).</p>	N/A	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Synchronization Coordinates service configuration settings and other information across instances.	Service Options <ul style="list-style-type: none"> • Max heap size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB. 	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Watchdog Monitors the other System services and restarts them if necessary. Also responsible for initial system startup.	Service Options <ul style="list-style-type: none"> • Max heap size: Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB. • Heap new size: Heap size for the Java virtual machine. Valid values are integers representing a number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Yes
		Scalable?	No

Viewing services

You can use System Management application, CLI, and REST API to view the status of all services for the system.

Related topics:

- ["Service list"](#) on page 11
- Service units

System Management application instructions

Viewing all services

To view the status of all services, in the System Management application, click on **Services**.

For each service, the page shows:

- The service name
- The service state. One of these:
 - **Healthy** – The service is running normally.
 - **Unconfigured** – The service has yet to be configured and deployed.
 - **Deploying** – The system is currently starting or restarting the service. This can happen when:
 - You move the service to run on a completely different set of instances.
 - You repair a service.

For information on viewing the status service operations, see "[Monitoring service operations](#)" on page 62.

- **Balancing** – The service is running normally, but performing some background maintenance operations.
- **Under-protected** – In a multi-instance system, one or more of the instances on which a service is configured to run are offline.
- **Failed** – The service is not running or the system cannot communicate with the service.
- **CPU Usage** – The current percentage CPU usage for the service across all instances on which it's running.
- **Memory** – The current RAM usage for the service across all instances on which it's running.
- **Disk Used** – The current total amount of disk space that the service is using across all instances on which it's running.

Viewing individual service status

To view the detailed status for an individual service, click on the service on the **Services** page.

In addition to the information above, the page shows:

- **Instances** – A list of all instances on which the service is running.
- **Volumes** – To view a list of volumes used by the service, click on the row for an instance in the **Instances** section. For more information, see "[Viewing volumes](#)" on page 117.
- **Network: [Internal | External]** – Which network type this service uses to receive communications.

This section also displays a list of the ports that the service uses.

For more information, see "[Networking](#)" on page 106.

- **Configuration settings** – The settings you can configure for the service.
- **Service Units** – The total number of service units currently being spent to run this service. This value is equal to the service's service unit cost times the number of instances on which the service is running. For more information, see [Service units](#).
- **Service unit cost** – The number of service units required to run the service on one instance. For more information, see [Chapter 1: "Services"](#) on page 9.
- **Service Instance Types** – For services that have multiple types, the types that are currently running.
- **Instance Pool** – For floating services, the instances that this service is eligible to run on. For more information, see [Chapter 1: "Services"](#) on page 9.
- **Events** – A list of all system events for the service.

Related CLI command(s)

getService

listServices

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Managing services

This section describes how you can reconfigure, restart, and otherwise manage the services running on your system.

Moving and scaling services

You can change a service to run on:

- Additional instances (for example, if you need improved service performance and availability)
- Fewer instances (for example, if you want to free up resources on an instance for running other services)
- A different set of instances (for example, if you are retiring the piece of hardware on which an instance is installed)

Moving and scaling floating services

For floating services, instead of specifying the specific instances on which the service runs, you can specify a pool of eligible instances, any of which can run the service. For more information, see [Chapter 1: "Services"](#) on page 9.

Moving and scaling services with multiple types

When moving or scaling a service that has multiple types, you can simultaneously configure separate rebalancing operations for each type.

Considerations

- You cannot remove a service from an instance if doing so would cause or risk causing data loss.
- Service relocation operations may take a long time to complete and may impact system performance while they are running.
- Instance requirements vary from service to service. Each service defines the minimum and maximum number of instances on which it can run.

For information on:

- Individual services and the number of instances they should run on, see ["Service list"](#) on page 11.

- Monitoring service relocation operations, see "[Monitoring service operations](#)" on page 62.



Tip: Use the **Available Instances** option to have a floating service be eligible to run on any instance in the system, including any new instances added in the future.

System Management application instructions

Configuring the service relocation operations manually

To manually configure a service relocation operation, in the System Management application:

1. Click on the **Services** panel.
2. Click on the service that you want to scale or move.
3. Click on the **Scale** tab.
4. If the service has more than one type, select the instance type that you want to scale.
5. If the service is a floating service, you are presented with options for configuring an instance pool:
 - a. In the **Service Instances** field, specify the number of instances on which the service should be running at any time.
 - b. Configure the instance pool:
 - To have the service run on any instance in the system, select the **All Available Instances** option.
 - With this option, the service can be restarted on any instance, including instances that were added to the system after the service was configured.
 - To have the service run on a specific set of instances, deselect the **All Available Instances** option. Then:
 - To remove an instance from the pool, select it from the **Instance Pool** list on the left. Then click on the **Remove Instances** button.
 - To add an instance to the pool, select it from the **Available Instances** list on the right. Then click on the **Add Instances** button.
6. If the service is a non-floating service, you are presented with options for selecting the specific instances that the service should run on. Do one or both of these:
 - To remove the service from the instances it's currently on, select one or more instances in the lefthand list. Then click on the **Remove Instances** button.

- To add the service to other instances, select one or more instances from the **Available Instances** list on the right. Then click on the **Add Instances** button.

7. Click on the **Update** button.

Configuring the service operations automatically

The system can check whether your services are currently deployed on the recommended minimum number of instances. If they aren't, the system can configure service scale operations for you.

To have the system do this for you, in the System Management application:

8. Click on the **Services** panel.

1. Click on the **Manage Services** button.

2. Click on the **Auto Scale** option. Then click on the **Next** button.

The system examines your service layout to ensure that each service is running on the minimum recommended number of instances. If one or more services are not running on enough instances, the system automatically creates scale operations for them.

3. Click on the **Update Service** button.

Related CLI command(s)

updateServiceConfig

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Related REST API method(s)

POST /services/configure

For information on specific REST API methods, in the System Management application, click on the help icon (❓). Then:

- To view the administrative REST API methods, click on **REST API - Admin**.

For general information about the administrative REST API, see "[REST API reference](#)" on page 163.

Configuring service settings

You can configure settings for some of the services that the system runs. For information on the settings you can configure for each service, see "[Service list](#)" on page 11.

For information on adding user-managed volumes to a service that supports them, see [Creating user-managed volumes](#).



Note: If you make an unwanted change to a service configuration, wait for the operation to finish before creating a new operation to correct the service configuration.

System Management application instructions

1. Click on the **Services** panel.
2. Click on the service you want to configure.
3. On the **Configuration** tab, configure the service.

For information on the settings available for each service, see "[Service list](#)" on page 11.

4. Click on the **Update** button.

Related CLI command(s)

updateServiceConfig

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Repairing services

If a service becomes slow, unresponsive, or shows a status of **Failed**, you can run a service operation to repair it. Repairing a service stops and restarts the service on each instance on which it's running.

For information on viewing service health and activity, see "[Monitoring services](#)" on page 55.



Important: Depending on which service you're repairing, parts of the system will be unavailable until the repair operation finishes.

To repair a service:

Managing services

1. Click on the **Services** panel.
2. Select the service you want to repair.
3. Click on the **Repair** button.

Chapter 2: Jobs

Jobs are operations that services run to perform some, usually transient, work. Like services, jobs are run in Docker containers on system instances. However when the job completes its work, its container exits.

Jobs are run by services; you cannot start or stop them yourself on demand, but you can schedule the times when they are allowed to run and specify which instances in the system that they are allowed to run on. For information, see "[Managing jobs](#)" on the next page



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Job types

Jobs are grouped into **job types**. All jobs in a type share the same default configuration settings. Newly created jobs inherit their settings from their job type. However, each job in a type can be configured with settings different from the job type default settings.

Storage for jobs

You can configure storage usage for jobs by associating volumes with job types. For information on volumes, see "[Volumes](#)" on page 114.

Networking

You can configure network ports for job types and individual jobs. For information, see "[Configuring job settings](#)" on page 48.

Managing jobs

This section describes how you can reconfigure, move, schedule, and otherwise manage the jobs that can run on your system.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Configuring where jobs run

The system lets you limit the system instances that jobs are allowed to run on. You can specify lists of allowed instances for all jobs, for jobs of a specific type, or for each individual job.

For example, if your system includes instances numbered 101 through 104, you could specify that:

- Only instances 101, 102, and 103 can run jobs
- Only instances 101 and 102 can run Example-type jobs
- Only instance 101 can run the Example-type job named Example1



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

System Management application instructions

To specify where jobs can run:

1. Click on **Jobs**.
2. Select the job you want and click on **Configure**.
3. Specify which instances can run jobs:
 - a. On the **Scale** tab:
 1. From the **Available Instances** list, select the instances where you want jobs to run. Then click on the **Add Instances** button.
 2. Optionally, enable the **Add newly selected instances to existing jobs** option.

With this option enabled, any new instances that you selected in the previous step will automatically be added as selected instances to all existing jobs. When disabled, existing jobs of this type keep their current lists of selected instances.

3. From the **Selected Instances** list, select any instances where you don't want jobs to run. Then click on the **Remove Instances** button.



Note:

- If you remove instances where jobs are already running, any existing jobs remain running on those instances, but new jobs will not run on the removed instances.
 - You cannot remove all selected instances from the **All Jobs** menu. If you don't want jobs to run anywhere, you need to remove the selected instances for each job type.
-

- b. Click on the **Update** button.

4. Specify which instances can run jobs of a particular type:

- a. Select a job type and click on **Configure**.
- b. On the **Scale** tab, from the **Eligible Instances** list, select the instances where you want this type of job to run. Then click on the **Add Instances** button.
- c. Optionally, enable the **Add newly selected instances to existing jobs** option.

With this option enabled, any new instances that you selected in the previous step will automatically be added as selected instances to all existing jobs of this type. When disabled, existing jobs of this type keep their current lists of selected instances.

- d. From the **Selected Instances** list, select any instances where you don't want this type of job to run. Then click on the **Remove Instances** button.



Note: If you remove instances where any jobs of this type are already running, the existing jobs remain running on those instances, but new jobs of this type will not run on the removed instances.

- e. Click on the **Update** button.

- f. Repeat this step for any other job types you want to configure.
5. Specify which instances can run a particular job:
 - a. On the **Scale** tab, select the instances you want from the **Available Instances** list and click on the **Add Instances** button. Here, the **Available Instances** list displays all instances where jobs of this type are allowed to run.
 - b. Click on the **Update** button.
 - c. Repeat this step for any other jobs you want to configure.

Related CLI command(s)

editJob

editJobSettings

editJobType

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Configuring job settings

Configurable settings

You can configure these Docker container settings for all job types and individual jobs:

- **Container Memory** – The hard memory limit for a job's Docker container, in megabytes (MB).
- **CPU** – The relative CPU usage weight for a job's Docker container. Generally, a higher value means that the container receives more CPU resources than other processes (including other service Docker containers) running on the instance.

Valid values are decimal numbers.

Other configurable settings are determined by the jobs and job types themselves.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Job setting inheritance

When you configure settings for a job type, those settings are inherited by newly created jobs, not by the jobs that exist at the time you make the configuration change.

System Management application instructions

1. Click on **Jobs**.
2. Select a job type.
3. On the **Configuration** tab, configure the settings you want.
4. Click on the **Update** button.

Related CLI command(s)

editJob

editJobType

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Scheduling jobs

You can configure a schedule for both job types and individual jobs.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Job type schedules

When you configure a schedule for a job type, the schedule is inherited by newly created jobs, not by the jobs that exist at the time you make the configuration change.

Individual job schedules

If a job has a schedule, the job enters the Running state automatically when a scheduled time period begins, regardless of the job's current state. A job can also be started by a service outside of the job's scheduled time periods.

When the end of a scheduled time period arrives:

- If the job is still running, the job stops and enters the Idle state.
- If the job is not running (for example, it's in the Completed or Failed state), the job remains in its current state.

For information on viewing a job's status, see "[Monitoring individual jobs](#)" on page 60

System Management application instructions

1. Click on **Jobs**.

2. Select a job type or individual job.
3. On the **Schedule** tab, use the calendar tool to specify when the job or job type should run.

	00:00	02:00	04:00	06:00	08:00	10:00	12:00	14:00	16:00	18:00	20:00	22:00
Monday	00:00 - 08:00									18:00 - 24:00		
Tuesday	00:00 - 08:00									18:00 - 24:00		
Wednesday	00:00 - 08:00									18:00 - 24:00		
Thursday	00:00 - 08:00									18:00 - 24:00		
Friday	00:00 - 08:00									18:00 - 24:00		
Saturday	00:00 - 24:00											
Sunday	00:00 - 24:00											

In this tool, click on a day to add a block of time. Click and drag the block to cover the hours you want the job or jobs to run.

To remove a block, right-click on it.

4. Click on the **Update** button.

Related CLI command(s)

editJob

editJobType

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Chapter 3: Monitoring

Your system provides a number of mechanisms that allow you to monitor the health and performance of the system and all of its instances and services.

Monitoring instances

You can use the System Management application, CLI, and REST API to view a list of all instances in the system.

Related topics:

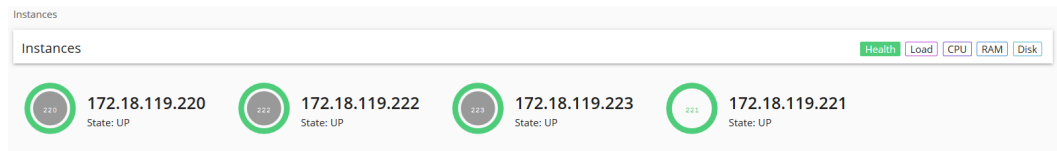
- ["Service list"](#) on page 11
- [Chapter 1: "Services"](#) on page 9

System Management application instructions


Viewing all instances

To view all instances, in the System Management application, click on **Dashboard > Instances**

The page shows all instances in the system. Each instance is identified by its IP address.



This table describes the information shown for each instance.

Property	Description
State	<p>One of these:</p> <ul style="list-style-type: none"> • Up – The instance is reachable by other instances in the system. • Down – The instance cannot be reached by other instances in the system.
Services	The number of services running on the instance.
Service Units	<p>The total number of service units for all services and job types running on the instance, out of the recommended service unit limit for the instance.</p> <p>An instance with a higher number of service units is likely to be more heavily utilized by the system than an instance with a lower number of service units.</p> <p>The Instances page displays a blue bar for instances running less than the recommended service unit limit.</p> <p>The Instances page displays a red bar for instances running more than the recommended service unit limit.</p> <p>Service Units  138 of 100 recommended Service Units</p> <p>For more information on service units and service unit recommendations, see Service units.</p>

Property	Description
Load Average	The load averages for the instance for the past one, five, and ten minutes.
CPU	The sum of the percentage utilization for each CPU core in the instance.
Memory Allocated	<p>This section shows both:</p> <ul style="list-style-type: none"> • The amount of RAM on the instance that's allocated to all services running on that instance. • The percentage of this allocated RAM to the total RAM for the instance.
Memory Total	The total amount of RAM for the instance.
Disk Used	The current amount of disk space that your system is using in the partition on which it is installed.
Disk Free	The amount of free disk space in the partition in which your system is installed.

Viewing the services running on an instance

To view the services running on an individual instance, in the System Management application:

1. Click on **Dashboard > Instances**.
2. Click on the instance you want.

The page lists all services running on the instance.

For each service, the page shows:

- The service name
- The service state. One of these:
 - **Healthy** – The service is running normally.
 - **Unconfigured** – The service has yet to be configured and deployed.
 - **Deploying** – The system is currently starting or restarting the service. This can happen when:
 - You move the service to run on a completely different set of instances.
 - You repair a service.

For information on viewing the status service operations, see ["Monitoring service operations"](#) on page 62.

- **Balancing** – The service is running normally, but performing some background maintenance operations.
 - **Under-protected** – In a multi-instance system, one or more of the instances on which a service is configured to run are offline.
 - **Failed** – The service is not running or the system cannot communicate with the service.
- **CPU Usage** – The current percentage CPU usage for the service across all instances on which it's running.

- **Memory** – The current RAM usage for the service across all instances on which it's running.
- **Disk Used** – The current total amount of disk space that the service is using across all instances on which it's running.

Related CLI command(s)

getInstance

listInstances

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Monitoring services

You can use System Management application, CLI, and REST API to view the status of all services for the system.

System Management application instructions

Viewing all services

To view the status of all services, in the System Management application, click on **Services**.

For each service, the page shows:

- The service name
- The service state. One of these:
 - **Healthy** – The service is running normally.
 - **Unconfigured** – The service has yet to be configured and deployed.
 - **Deploying** – The system is currently starting or restarting the service. This can happen when:
 - You move the service to run on a completely different set of instances.
 - You repair a service.

For information on viewing the status service operations, see "[Monitoring service operations](#)" on page 62.

- **Balancing** – The service is running normally, but performing some background maintenance operations.
- **Under-protected** – In a multi-instance system, one or more of the instances on which a service is configured to run are offline.
- **Failed** – The service is not running or the system cannot communicate with the service.
- **CPU Usage** – The current percentage CPU usage for the service across all instances on which it's running.
- **Memory** – The current RAM usage for the service across all instances on which it's running.
- **Disk Used** – The current total amount of disk space that the service is using across all instances on which it's running.

Viewing individual service status

To view the detailed status for an individual service, click on the service on the **Services** page.

In addition to the information above, the page shows:

- **Instances** – A list of all instances on which the service is running.
- **Volumes** – To view a list of volumes used by the service, click on the row for an instance in the **Instances** section. For more information, see "[Viewing volumes](#)" on page 117.
- **Network: [Internal | External]** – Which network type this service uses to receive communications.

This section also displays a list of the ports that the service uses.

For more information, see "[Networking](#)" on page 106.

- **Configuration settings** – The settings you can configure for the service.
- **Service Units** – The total number of service units currently being spent to run this service. This value is equal to the service's service unit cost times the number of instances on which the service is running. For more information, see [Service units](#).
- **Service unit cost** – The number of service units required to run the service on one instance. For more information, see [Chapter 1: "Services"](#) on page 9.
- **Service Instance Types** – For services that have multiple types, the types that are currently running.
- **Instance Pool** – For floating services, the instances that this service is eligible to run on. For more information, see [Chapter 1: "Services"](#) on page 9.
- **Events** – A list of all system events for the service.

Related CLI command(s)

getService

listServices

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Monitoring jobs

You can use System Management application, CLI, and REST API to monitor jobs. You can monitor information for all jobs, for all jobs of a type, or for each job individually.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Monitoring all jobs

You can use System Management application, CLI, and REST API to view the status of all jobs for the system. For information on jobs, see [Chapter 2: "Jobs"](#) on page 45.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

System Management application instructions

To view the status of all jobs, in the System Management application, click on **Jobs**.

On this page, the **All Jobs** section shows cumulative information for all jobs in the system.

- **Service Units** – The total number of service units currently being spent for running jobs.

This value is determined by where your system's job types are allowed to run.

Each job type has its own service unit cost. If an instance is configured to run multiple job types, only the job type with the highest service unit cost counts.

For example, suppose that your system has 4 instances and supports two job types: X, which costs 50 service units, and Y, which costs 25. Job type X is configured to run on 3 instances. Job type Y is configured to run on those same 3 instances, plus an additional instance (4 total). In this case, your total service unit cost for jobs is equal to 175:

$$50 + 50 + 50 + 25 = 175$$

For information on configuring which instances job types can run on, see ["Configuring where jobs run"](#) on page 46.

- **Total CPU** – The total CPU percentage utilization for all jobs across all instances in the system.
- **Total Memory** – The total RAM consumed by all jobs across all instances in the system.
- **Total Disk** – The total disk space consumed by all jobs across all instances in the system.

The **Job Types** section shows a box for each job type that the system supports.

Related CLI command(s)

listJobs

For information on running CLI commands, see ["CLI reference"](#) on page 153.

Monitoring job types

You can use System Management application, CLI, and REST API to view the status of all jobs of a particular type. For information on jobs, see [Chapter 2: "Jobs"](#) on page 45.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

System Management application instructions

To view the detailed status for all jobs of a type, in the System Management application, click on **Jobs**. Then click on the box for the job type you want.

The **Job Type** page shows cumulative information for all jobs of the selected type.

- **Total CPU** – The total CPU percentage utilization for all jobs of this type across all instances in the system.
- **Total Memory** – The total RAM consumed by all jobs of this type across all instances in the system.
- **Total Disk** – The total disk space consumed by all jobs of this type across all instances in the system.

The **Jobs** section lists each individual job of this type. For more information, see ["Monitoring individual jobs"](#) below.

The **Instances** section shows a list of all system instances where jobs of this type are currently running.

The **Pool** section shows the instances that jobs of this type are allowed to run on. For information on configuring this, see ["Configuring where jobs run"](#) on page 46.

The **Events** section shows a list of all events related to this job type.

Related CLI command(s)

getJobTypes

For information on running CLI commands, see ["CLI reference"](#) on page 153.

Monitoring individual jobs

You can use System Management application, CLI, and REST API to view the status of individual jobs. For information on jobs, see [Chapter 2: "Jobs"](#) on page 45.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

System Management application instructions

To view the detailed status for all jobs of a type, in the System Management application:

1. Click on **Jobs**.
2. Click on the box for the job type you want.
3. Click on the box for the job you want.

The **Job** page shows this information about the job:

- **Error Count** – A tally of the number of errors encountered by the system when running the job or relocating it to other instances.
- **Status** – One of these:
 - **Idle** – The job has not run yet or, for jobs that have a schedule, the job is not scheduled to be running at the current time.
 - **Pending** – The job has been submitted for execution but is not yet running.

- **Running** – The job is being executed.
 - **Completed** – The job finished execution without error.
 - **Failed** – The job finished execution with errors.
 - **Canceled** – The job has been canceled and is no longer running.
- If the task is configured to run according to a schedule, the **Status** panel shows this icon:

For more information, see "[Scheduling jobs](#)" on page 49.

- **Service Units** – The per-instance cost for this job's type. Individual jobs do not cost any service units to run. For more information, see "[Monitoring job types](#)" on page 59.
- **Total CPU** – The total CPU percentage utilization for this job across all instances in the system.
- **Total Memory** – The total RAM consumed by this job across all instances in the system.
- **Total Disk** – The total disk space consumed by this job across all instances in the system.

The **Instances** section shows a list of all system instances where the job is currently running.

To view information about the volumes that the job uses, click on the row for an instance in the **Instances** section. For more information, see "[Viewing volumes](#)" on page 117.

The **Pool** section shows the instances that job is allowed to run on. For information on configuring this, see "[Configuring where jobs run](#)" on page 46.

The **Events** section shows a list of all events related to this job.

Related CLI command(s)

getJobStatus

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Monitoring processes

The **Processes** page lets you view information about what the system is doing. This includes any service operations you started and any internal maintenance processes the system needs to run.

Monitoring service operations

You can use the System Management application, CLI, and REST API to monitor all service operations. These operations include:

- The initial deployments of services when the system was installed.
- Service relocation operations that you initiate. For information, see "[Moving and scaling services](#)" on page 39.

For each service operation, the system shows:

- The name of the service involved
- The status of the operation
- The number of steps completed out of the total number of steps for the operation

System Management application instructions

1. Click on the **Processes** panel.

The **Service Operations** tab shows information about in-progress and completed service operations.

Related CLI command(s)

`listSystemTasks`

`getSystemTask`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Monitoring system processes

You can use System Management application, REST API, and CLI to view the progress of internal system processes. These include package installation tasks and regularly scheduled system maintenance activities such as log rotation.

For each process, your system shows:

- The process name
- The process state
- The times at which each step in the process run occurred

System Management application instructions

1. In the System Management application, click on **Processes**.
2. To view the currently running processes, click on the **System** tab.
3. To view the scheduled processes, click on the **Scheduled** tab.

Related CLI command(s)

listSystemTasks

getSystemTask

System processes have a type of SCHEDULED or ONE-TIME.

For information on running CLI commands, see "[CLI reference](#)" on page 153.

System events

Your system maintains a log of system events that you can view through the System Management application, CLI, and REST API.

System Management application instructions

To view all system events, in the System Management application, click on **Events**.

Related CLI command(s)

queryEvents

To view events through the CLI, your requests need to specify which events you want to retrieve.

For example, this JSON request body searches the event log for all events that have a severity level of `warning`:

```
{
  "severities": [
    "warning"
  ]
}
```

```
]
}
```

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Alerts

Your system displays alert messages to notify you of situations that require your attention. You can view these alerts through the System Management application, CLI, and REST API.

Each alert corresponds to a system event. For information on:

- Viewing system events, see "[System events](#)" on the previous page.
- Receiving emails when events occur, see "[Creating email notification rules](#)" on page 68.

List of alerts


Severity	Alert Description	Action
Severe	Instance <ip-address> disk usage severe threshold	<p>The specified instance has less than 10% free disk space. Add additional storage to the instance.</p> <p>Important: If an instance runs out of disk space, the system may become unresponsive.</p>
Severe	Master Instance <ip-address> is down	<p>One of these:</p> <ul style="list-style-type: none"> Restart the instance hardware or virtual machine. Restart the <code>run</code> script on the instance. This script is located in the <code>bin</code> directory in the install directory.
Severe	Service is down	<p>Check the health of your instances. If one is down, do one of these:</p> <ul style="list-style-type: none"> Restart the instance hardware or virtual machine. Restart the <code>run</code> script on the instance. This script is located in the <code>bin</code> directory in the install directory. <p>Otherwise, if your instances are healthy and the problem persists, contact support.</p>
Severe	Worker Instance <ip-address> is down	<p>One of these:</p> <ul style="list-style-type: none"> Restart the instance hardware or virtual machine. Restart the <code>run</code> script on the instance. This script is located in the <code>bin</code> directory in the install directory.

Severity	Alert Description	Action
Warning	Instance <ip-address> disk usage warning threshold	<p>The specified instance has less than 25% free disk space. Add additional storage to the instance.</p> <p>Important: If an instance runs out of disk space, the system may become unresponsive.</p>
Warning	Package installation failed	<p>Your system failed to install a package that you uploaded.</p> <p>For information on</p> <ul style="list-style-type: none"> • Packages, see "Packages" on page 143. • Viewing the status of the package installation task, see "Monitoring system processes" on page 62.
Warning	Service below recommendation	<p>The service is currently running on fewer than the recommended number of instances. Configure this service to run on additional instances.</p> <p>For information on:</p> <ul style="list-style-type: none"> • The number of instances that each service should run on, see "Service list" on page 11. • Scaling services, see "Moving and scaling services" on page 39. • Adding new instances, see "Adding new instances" on page 123.

Severity	Alert Description	Action
Warning	Service under-protected	<p>A service has lost redundancy; that is, one or more instances on which that service is running are unresponsive.</p> <p>Check the health of your instances. If one is down, do one of these:</p> <ul style="list-style-type: none"> Restart the instance hardware or virtual machine. Restart the <code>run</code> script on the instance. This script is located in the <code>bin</code> directory in the install directory. <p>Otherwise, if your instances are healthy and the problem persists, contact support.</p>
Warning	SSL server certificate chain expires soon	<p>A certificate in the SSL server certificate chain for this system expires soon. If the certificate chain expires, users will be unable to access the system.</p> <p>See "Changing the system certificate" on page 97.</p>
Warning	SSL server certificate chain expired	<p>The SSL server certificate chain for this system contains an expired certificate. Users cannot access the system until the certificate chain is replaced.</p> <p>See "Changing the system certificate" on page 97.</p>

Severity	Alert Description	Action
Info	Package installation in progress	<p>Your system is currently installing a package that you uploaded. Depending on the contents of the package, this may take a while.</p> <p>For information on</p> <ul style="list-style-type: none"> • Packages, see "Packages" on page 143. • Viewing the status of the package installation task, see "Monitoring system processes" on page 62.
Warning	The certificate for the storage component (storage_id) is about to expire in n days	Renew the storage component certificate.
Info	The storage component (storage_id) is unavailable	Verify that the storage component id is correct and valid and that the storage component is active.

System Management application instructions

To view alerts, click on the user icon () in the top righthand corner of each System Management application page, then click on **Notifications**.

Related CLI command(s)

listAlerts

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Creating email notification rules

For the system to send email notifications, you need to create a rule that specifies who to email, what email server to use, what events to send emails about, and what information to include in email messages.

SMTP settings

- **Enable** – Turns on email notifications.
- **Host** – The hostname or IP address of the email server.
- **Port** – The port on which the email server listens for email messages.
- **Security** – The security protocol used by the email server (**SSL** or **STARTTLS**) or **None** if the email server doesn't use a security protocol.
- **Authenticated** – Enable this if the email server requires authentication, then specify:
 - In the **Username** field, the username for an email account that's authorized to establish the connection between the system and the email server.
 - In the **Password** field, the password for the email account.

Message settings

You use the email notification message settings to configure a template for formatting all email notifications sent by the system.

- **From** – The email address from which you want email notifications to be sent.
- **Subject** – The email subject.
- **Body** – The email message body.

Message variables

This table lists the variables you can use to construct the email notification template. When it sends an email notification, the system replaces the variables with event-specific information.

Variable	Description
\$severity	Event severity: INFO, WARNING, or SEVERITY
\$subject	A short description of the event
\$message	Event message text
\$userName	Name of the user responsible for the event
\$objectId	Unique identifier for component affected by the event
\$subsystem	Category for the component affected by the event
\$objectSourceId	Unique identifier of the internal system component or process that was the source of the event. Value is [unknown] for most events.

Recipient settings

- **Email addresses** – A comma-separated list of email addresses to send notification emails to.
- **Severity Filter** – The event severities about which to send email notifications. Can be one or more of these: INFO, WARNING, SEVERITY.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Notifications**.
3. Click on the **Create** button.

4. In the **Type** field, select **Email**.
5. Enter a name for the notification rule.
6. Configure the SMTP settings and message settings for the notification rule.
7. Specify a comma-separated list of emails to send notifications to.
8. Specify a severity filter for the notification role.
9. Click on the **Create** button.

Related CLI command(s)`createNotificationRule`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Creating syslog notification rules

When you create a syslog notification rule, the system sends log messages to your syslog server for each applicable system event.

Syslog settings

- **Enable** – Turns on syslog notifications.
- **Host** – The hostname or IP address of the syslog server.
- **Port** – The port on which the syslog server listens for log messages.
- **Facility** – Category for the messages sent by this notification rule.

Message settings

You use the syslog notification message settings to configure a template for formatting all syslog notifications sent by this notification rule.

- **Message** – The message to send. You can use these variables as part of the message.

Variable	Description
\$severity	Event severity: INFO, WARNING, or SEVERITY
\$subject	A short description of the event
\$message	Event message text
\$time	Time at which the event occurred
\$userName	Name of the user responsible for the event
\$subsystem	Category for the component affected by the event
\$objectId	Unique identifier for component affected by the event
\$objectType	The type of the component affected by the event.
\$objectSourceId	Unique identifier of the internal system component or process that was the source of the event. Value is [unknown] for most events.
\$objectSourceType	Type of the internal system component or process that was the source of the event. Value is [unknown] for most events.

- **Sender Identity** – Identity of the sender for the event. Sent with every syslog message.

Severity filter

The event severities about which to send email notifications. Can be one or more of these: INFO, WARNING, SEVERITY.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Notifications**.
3. Click on the **Create** button.

4. In the **Type** field, select **Syslog**.
5. Enter a name for the notification rule.
6. Configure the settings for the notification rule.
7. Specify a severity filter for the notification rule.
8. Click on the **Create** button.

Related CLI command(s)

createNotificationRule

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Logs and diagnostic information

Each service and job type maintains its own set of logs. By default, the logs are maintained in the `/<install-directory>/log` directory on each instance in the system. During installation, each service and job type can be configured to store its logs in a different, non-default location.

Log management

You can manage any of the system log files yourself if you want to. That is, you can delete or archive them as necessary.



Note: Deleting log files may make it more difficult for system support personnel to resolve issues you may encounter.

System logs are managed automatically in these ways:

- All log files are periodically added to a compressed file and moved to `/<install-directory>/retired/`. This occurs at least once a day, but can also occur:
 - Whenever you run the `log_download` script.
 - Hourly, if the system instance's disk space is more than 60% full.
- When a log file grows larger than 10MB in size, the system stops writing to that file, renames it, and begins writing to a new file. For example, if `exampleService.log.0` grows too large, it is renamed to `exampleService.log.1` and the system creates a new `exampleService.log.0` to write to.

Retrieving logs and diagnostic information

The `log_download` tool lets you easily retrieve logs and diagnostic information from all instances in the system. This tool is located at this path on each instance:

```
<install-directory>/bin/log_download
```

For information on running the tool, run:

```
<install-directory>/bin/log_download -h
```



Note:

- When using the `log_download` tool, if you specify the `--output` option, do not specify an output path that contains colons, spaces, or symbolic links. If you omit the `--output` option, you cannot run the script from within a directory path that contains colons, spaces, or symbolic links.
- When you run the `log_download` script, all log files are automatically compressed and moved to the `<install-directory>/retired/` directory.
- If an instance is down, you need to specify the `--offline` option to collect the logs from that instance. If your whole system is down, you need to run the `log_download` script with the `--offline` option on each instance.

Default log locations

By default, each service stores its logs in its own directory at this path:

```
<install-directory>/log
```

This table shows the default log directory names for each service. Depending on how your system was configured when first deployed, your system's logs may not be stored in these directories.

For information on the services, see "[Service list](#)" on page 11.

Default log directory name	Related service	Contains information about
<code>com.hds.ensemble.plugins.service.adminApp</code>	System-Management-application	The System Management application.

Default log directory name	Related service	Contains information about
com.hds.ensemble.plugins.service.cassandra	Database	<ul style="list-style-type: none"> • System configuration data. • Document fields and values.
com.hds.ensemble.plugins.service.chronos	Scheduling	Workflow task scheduling.
com.hds.ensemble.plugins.service.elasticsearch	Metrics	<p>The storage and indexing of:</p> <ul style="list-style-type: none"> • System events • Performance and failure metrics for workflow tasks
com.hds.ensemble.plugins.service.haproxy	Network-Proxy	Network requests between instances.
com.hds.ensemble.plugins.service.kafka	Message Queue	Transmission of data between instances.

Default log directory name	Related service	Contains information about
com.hds.ensemble.plugins.service.logstash	Logging	The transport of system events and workflow task metrics to the Metrics service.
com.hds.ensemble.plugins.service.marathon	Service-Deployment	The deployment of high-level services across system instances. High-level services are the ones that you can move and configure (such as Index), not the services grouped under System Services.
com.hds.ensemble.plugins.service.mesosAgent	Cluster-Worker	Work ordered by the Cluster-Coordination service.
com.hds.ensemble.plugins.service.mesosMaster	Cluster-Coordination	Hardware resource allocation.
com.hds.ensemble.plugins.service.remoteAction	Watchdog	Internal system processes.

Default log directory name	Related service	Contains information about
com.hds.ensemble.plugins.service.sentinel	Sentinel	Internal system processes.
com.hds.ensemble.plugins.service.watchdog	Watchdog	General diagnostic information.
com.hds.ensemble.plugins.service.zookeeper	Synchronization	Coordination of actions and database operations across instances.
com.hitachi.aspen.foundry.service.mapi.gateway	MAPI-Gateway	Management of API requests.
com.hitachi.aspen.foundry.service.metadata.cache	Metadata-Cache API requests	Transmission of system metadata between instances.
com.hitachi.aspen.foundry.service.metadata.coordination	Metadata-Coordination	Transmission of system metadata between instances.
com.hitachi.aspen.foundry.service.metadata.gateway	Metadata-Gateway	Transmission of system metadata between instances.
com.hitachi.aspen.foundry.service.metadata.async.policy.engine	Metadata-Policy-Engine	Update operations.

Default log directory name	Related service	Contains information about
com.hitachi.aspen.foundry.service.clientaccess.data	S3-Gateway	Transmission of S3 requests to endpoints.
com.hitachi.aspen.foundry.service.jaeger.agent	Tracing-Agent	Usage of tracing operations.
com.hitachi.aspen.foundry.service.jaeger.collector	Tracing-Collector	Usage of tracing collections.
com.hitachi.aspen.foundry.service.jaeger.query	Tracing-Query	Usage of tracing queries.

Chapter 4: Security

This section contains information on configuring system security features, including user authentication.

Granting access to users

These are the general steps you need to take to grant users access to the system:

1. Add one or more identity providers to the system.

For information, see "[Adding identity providers](#)" on the next page.

2. Add one or more groups from your identity providers to the system.

For information, see "[Adding groups](#)" on page 86.

3. Create a role that contains the system permissions you want to associate with a group of users.

For information, see "[Creating roles](#)" on page 88.

4. Associate roles with groups.

For information, see "[Assigning roles to groups](#)" on page 87.

Setting the session timeout limit

You can use the System Management application, REST API, or CLI to set the system session timeout limit. This limit affects user sessions in all applications that your system runs and also affects the length of time that REST API authorization tokens are valid.

For information on REST API authorization tokens, see "[REST API reference](#)" on page 163.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Settings** tab, type a number of minutes in the **Session Timeout** field.
4. Click on the **Update** button.

Related CLI command(s)

editSecuritySettings

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Identity providers

The system supports these identity provider types for user authentication:

- Active Directory (AD)
- OpenLDAP
- 389 Directory Server
- LDAP Compatible – Other LDAP-compatible identity providers not listed above.

To use one of these systems to authenticate users with your system, you need to first add your identity provider to the system.

Adding identity providers

For information on the types of identity providers you can add, see "[Identity provider configuration settings](#)" on the facing page.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Identity Providers** tab, click on the **Create** button.

4. Select an identity provider type and configure it. For information, see "[Identity provider configuration settings](#)" below.
5. Click on the **Create** button.

Related CLI command(s)`createIdentityProvider`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Identity provider configuration settings

These sections describe the configuration settings for each type of identity provider that your system supports.

All types

Security Realm Name – The name by which to identify this identity provider in the system. This name appears as an option in the Security Realm drop-down on login pages.



Tip: To ensure that your users can easily log into the system, pick security realm names that your users will recognize and understand.

Active Directory

- **Identity Provider Hostname** – Hostname or IP address for the identity provider.
- **Transport Security** – The protocol to use for securing communications between the system and the identity provider. Options are:
 - **None**
 - **TLS Security (Transport Layer Security)**
 - **SSL (Secure Sockets Layer)**
- **Identity Provider Host Port** – Network port used to communicate with the identity provider. The default value depends on the **Transport Security** setting:
 - For **None** or **TLS Security (Transport Layer Security)**, 389
 - For **SSL (Secure Sockets Layer)**, 636
- **User Name** – A user account on the identity provider. Your system uses this user account to read information from the identity provider.
- **Password** – The user account password.
- **Domain** – The AD domain in which the user account is defined.



Note: Use the short name for the AD domain. For example, use `MYACTIVEDIRECTORY` instead of `MYACTIVEDIRECTORY.local`.

- **Search Base DN** – The distinguished name (DN) of the identity provider location where you want your system to begin its searches for users and groups.

For example, if you specify a value of

OU=Users,DC=corp,DC=example,DC=com, the system searches for users and groups in the organization unit called **Users** in the **corp.example.com** domain.

- **Default Domain Name** – The default domain for users logging into the System Management application and Search App. For example, if you specify a default domain name of **east.example.com**, the user **jdoe@east.example.com** needs to specify only **jdoe** when logging into either app.

LDAP Compatible

- **Identity Provider Hostname** – Hostname or IP address for the identity provider.
- **Transport Security** – The protocol to use for securing communications between the system and the identity provider. Options are:
 - **None**
 - **TLS Security (Transport Layer Security)**
 - **SSL (Secure Sockets Layer)**
- **Identity Provider Host Port** – Network port used to communicate with the identity provider. The default value depends on the **Transport Security** setting:
 - For **None** or **TLS Security (Transport Layer Security)**, 389
 - For **SSL (Secure Sockets Layer)**, 636
- **User Name** – A user account on the identity provider. Your system uses this account to read information from the identity provider.
- **Password** – The user account password.
- **User DN Template** – A template on the LDAP server. When a user logs into their system, the provided username is inserted into this template to determine the user's LDAP distinguished name (DN).
- **Unique ID** – The unique identifier for the specified LDAP server.
- **Member Name Attribute** – The name of the attribute that each group on the identity provider uses to list its members.
- **Search Base DN** – The distinguished name (DN) of the identity provider location where you want your system to begin searching for users and groups.

For example, if you specify a value of

OU=Users,DC=corp,DC=example,DC=com, your system searches for users and groups in the organization unit called **Users** in the **corp.example.com** domain.

- **Group Object Class** – The objectClass value for groups on the LDAP server.

OpenLDAP and 389 Directory Server:

- **Identity Provider Hostname** – Hostname or IP address for the identity provider.
- **Transport Security** – The protocol to use for securing communications between the system and the identity provider. Options are:
 - **None**
 - **TLS Security (Transport Layer Security)**
 - **SSL (Secure Sockets Layer)**
- **Identity Provider Host Port** – Network port used to communicate with the identity provider. The default value depends on the **Transport Security** setting:
 - For **None** or **TLS Security (Transport Layer Security)**, 389
 - For **SSL (Secure Sockets Layer)**, 636
- **User Name** – A user account on the identity provider. Your system uses this account to read information from the identity provider.
- **Password** – The user account password.
- **User DN Template** – A template on the LDAP server. When a user logs into their system, the provided username is inserted into this template to determine the user's LDAP distinguished name (DN).
- **Unique ID** – The unique identifier for the specified LDAP server.
- **Member Name Attribute** – The name of the attribute that each group on the identity provider uses to list its members.
- **Search Base DN** – The distinguished name (DN) of the identity provider location where you want your system to begin searching for users and groups.

For example, if you specify a value of

OU=Users,DC=corp,DC=example,DC=com, your system searches for users and groups in the organization unit called **Users** in the **corp.example.com** domain.

User information caching

The system caches the following information from each of your identity providers:

- The names of users who access the system
- The groups that each user belongs to

As long as this information is in the system's cache, your users can perform any activities for which they have permissions, without the system needing to reconnect to the identity provider.

User information remains in the cache for four hours.

Clearing the cache

Any changes that you make on the identity provider are not reflected in the system until the information is removed from the cache. For example, if you delete a user from the identity provider, that user will be able to access the system for up to four hours, or until the cache is cleared.

Related CLI command(s)

`clearCache`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Viewing identity providers

You can use the System Management application, REST API, and CLI to view the identity providers that have been added to your system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. Click on the **Identity Providers** tab.

Related CLI command(s)

`getIdentityProvider`

`listIdentityProviders`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Deleting identity providers

When you delete an identity provider from your system, all users from that provider lose access to the system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Identity Providers** tab, click on the delete icon () for the server you want to remove.

Related CLI command(s)

`deleteIdentityProvider`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Groups

To allow user access to your system, you need to add groups to your system. These groups are defined on your organization's identity providers. Once you've added a group to your system, you can specify what roles its members have.

For information on:

- Adding identity providers to your system, see "[Adding identity providers](#)" on page 80.
- Roles, see "[Roles](#)" on page 88.

Adding groups

You use the REST API, System Management application, or CLI to add groups from your identity providers to your system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Groups** tab, click on the **Create** button.

4. Select an identity provider and type a string on which to query the identity provider for groups.
5. Click on the **Discover Groups** button.
6. Click on the **Continue** button.
7. Select one or more roles to associate with the group.
8. Click on the **Create** button.

Related CLI command(s)`createGroup`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Viewing groups

You use the REST API, CLI, or System Management application to view all the groups that have been created for your system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. Click on the **Groups** tab.

Related CLI command(s)`getGroup``listGroups`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Assigning roles to groups

You use the REST API, System Management application, and CLI to assign roles to the groups that you've added your system.

For information on roles, see "[Roles](#)" on the next page.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.

3. On the **Groups** tab, click on the group you want to edit.
4. On the **Roles** tab, select one or more roles to enable for the group.
5. Click on the **Update** button.

Related CLI command(s)

editGroup

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Deleting groups

When you delete a group, all users in the group lose access to your system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. Click on the **Groups** tab.
4. Click on the delete icon () for the group you want to remove.

Related CLI command(s)

deleteGroup

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Roles

Roles determine what actions a group of users can perform. You create your own roles, each of which can grant permission to perform any combination of actions.

For information on associating a role with a group of users, see "[Assigning roles to groups](#)" on the previous page.

Creating roles

You can use the REST API, System Management application, and CLI to create roles and select which permissions the roles contain.

About permissions

Each permission in a role grants a user the ability to perform an action in some area of the system. For example, the **admin:services:read** permission grants the ability to view services through the System Management application.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Roles** tab, click on the **Create** button.
4. Specify a name and, optionally, a description for the role.
5. Use the **Individual** and **Wildcard** tabs to edit the permissions for the role.

On the **Individual** tab, you can enable individual permissions or categories of permissions:

- Click on a category of permissions and select one or more individual permissions within the category.

For example, with the permissions selected in this image, a user can read, create, and update certificates, but cannot delete them.

Permissions Group - Certificates

<input checked="" type="checkbox"/>	Name ↑	Description
<input checked="" type="checkbox"/>	workflow:certificates:create	Create certificate
<input type="checkbox"/>	workflow:certificates:delete	Delete certificates
<input checked="" type="checkbox"/>	workflow:certificates:read	Read certificates(s)
<input checked="" type="checkbox"/>	workflow:certificates:update	Generate certificate

On the **Wildcard** tab, you can enable permissions for multiple categories at the same time. To do this:

- a. Click on the **Add Permission** button.
 - a. Use the drop-down menus to select a category of permissions.
 - b. Leave the last drop-down menu set to the wildcard character (*).
6. Click on the **Create** button.

7. Click on the **Update** button.

Related CLI command(s)

createRole

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Viewing roles

You can use the REST API, CLI, and System Management application to view all the roles that have been created for your system.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. Click on the **Roles** tab.

Related CLI command(s)

getRole

listRoles

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Editing roles

You can use the REST API, System Management application, and CLI to change the permissions that a role contains.

About permissions

Each permission in a role grants a user the ability to perform an action in some area of the system. For example, the **admin:services:read** permission grants the ability to view services through the System Management application.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. Click on the **Roles** tab.
4. Click on the role you want to edit.
5. Use the **Individual** and **Wildcard** tabs to edit the permissions for the role.

On the **Individual** tab, you can enable individual permissions or categories of permissions:

- Click on a category of permissions and select one or more individual permissions within the category.

For example, with the permissions selected in this image, a user can read, create, and update certificates, but cannot delete them.

Permissions Group - Certificates

<input checked="" type="checkbox"/>	Name ↑	Description
<input checked="" type="checkbox"/>	workflow.certificates.create	Create certificate
<input type="checkbox"/>	workflow.certificates.delete	Delete certificates
<input checked="" type="checkbox"/>	workflow.certificates.read	Read certificate(s)
<input checked="" type="checkbox"/>	workflow.certificates.update	Generate certificate

On the **Wildcard** tab, you can enable permissions for multiple categories at the same time. To do this:

- Click on the **Add Permission** button.
 - Use the drop-down menus to select a category of permissions.
 - Leave the last drop-down menu set to the wildcard character (*).
- Click on the **Create** button.
 - Click on the **Update** button.

Related CLI command(s)

editRole

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Deleting roles

When you delete a role, all groups associated with that role lose the permissions that the role granted.

System Management application instructions

- Click on the **Configuration** panel.
- Click on **Security**.

3. Click on the **Roles** tab.
4. Click on the delete icon () for the role you want to remove.

Related CLI command(s)

deleteRole

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Permissions

The following tables list the permissions available for system roles. The words **Yes** and **No** indicate whether or not the permission is assigned to a default role.

MAPI Alerts		
Permission name	Description	Default admin role permission?
mapi:alert:list	List all active alerts	Yes

MAPI Job Configurations		
Permission name	Description	Default admin role permission?
mapi:job_configuration:list	List all job configurations	Yes
mapi:job_configuration:run	Run a job configuration immediately	Yes
mapi:job_configuration:update	Modify a job configuration	Yes

MAPI S3 Settings		
Permission name	Description	Default admin role permission?
mapi:s3_settings:get	Read S3 settings	Yes
mapi:s3_settings:set	Modify SS3 settings	Yes

MAPI Storage Component		
Permission name	Description	Default admin role permission?
mapi:storage_component:activate	Activate a storage component	Yes
mapi:storage_component:create	Create a storage component	Yes
mapi:storage_component:list	List storage component(s)	Yes
mapi:storage_component:test	Test a storage component	Yes
mapi:storage_component:update	Modify a storage component	Yes
mapi:storage_component:update_state	Modify state of a storage component	Yes

MAPI Stored Objects		
Permission name	Description	Default admin role permission?
mapi:client_object:lookup	List stored objects	Yes

MAPI System		
Permission name	Description	Default admin role permission?
mapi:system:info	List system information	Yes

MAPI User		
Permission name	Description	Default admin role permission?
mapi:user:list	List system information	Yes
mapi:user:revoke_credentials	Revoke S3 credentials	Yes
mapi:user:revoke_tokens	Revoke OAuth tokens	Yes

S3 User		
Permission name	Description	Default admin role permission?
s3:user:generate_credentials	Generate S3 credentials	Yes

Serial Number		
Permission name	Description	Default admin role permission?
mapi:serial_number:get	Read serial number	Yes
mapi:serial_number:set	Modify serial number	Yes

Changing the admin account password

Your system includes a single local user account called admin, which is available when you first install the system. You can use the REST API, System Management application, or CLI to change the password for this account.

System Management application instructions

1. When logged into the System Management application with the admin user account, click on the user icon in the top righthand corner of the screen.
2. Click on **Change Password**.
3. Confirm your current password and specify a new password.
4. Click on the **Change Password** button.

Related CLI command(s)

`updateCurrentUserPassword`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Certificates

Your system uses SSL to provide security for the System Management application. To enable SSL security, you need a valid SSL server certificate or chain of certificates.

Your system comes with its own self-signed SSL server certificate, which is generated and installed automatically when the system is installed. This certificate is not automatically trusted by web browsers.

You can choose to trust this self-signed certificate or to replace it with one from a certificate authority (CA) or one that you create yourself. You can also have the system generate and install a new self-signed SSL server certificate. You would do this, for example, if the current certificate is close to expiring and you are waiting to retrieve a new one from your CA.

Viewing installed certificates

You can use the REST API, CLI, and System Management application to view information about:

- The system certificate. That is, the certificate used to secure communications for your system's applications, CLIs, and REST APIs.
- Data source certificates. These are the certificates retrieved from the systems that your system has connected to using a data connection. For information on data connections, see s.

For each certificate, you can view:

- The distinguished name of the certificate
- The date and time when the certificate goes (or went) into effect
- The date and time when the certificate expires (or expired)

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Certificates**.

The **System** tab displays the currently active system certificate.

3. To view the data source certificates, click on the **Client** tab.

Related CLI command(s)

listCertificates

getCertificate

getSystemCertificate

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Adding data source certificates

For your system to retrieve documents from a data source that uses SSL-protected communication, it must accept the certificate from the data source. Your system prompts you to accept a data source certificate when it tests the connection to the data source. You can also upload data source certificates manually.

System Management application instructions**Uploading data source certificates manually**

1. Retrieve the SSL certificate from your data source.
2. In the System Management application, click on **Configuration**.
3. Click on **Certificates**.
4. On the **Client** tab, click on Upload **Client Certificate**.
5. Click and drag the certificate file into the **Upload License** box.

Related CLI command(s)

testDataSource

createCertificate

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Changing the system certificate

By default, your system includes a self-signed certificate when the system is first installed.

You cannot delete the currently installed certificate. However, you can replace it by:

- Installing a new PKCS12 certificate (for instructions, see "[Installing a certificate you created](#)" on the next page)

- Generating and installing a new self-signed certificate (for instructions, see ["Installing a new self-signed certificate"](#) on the facing page)
- Generating a certificate signing request (CSR) and installing the certificate you receive in response to this request (for instructions, see ["Creating a CSR and installing the returned certificate"](#) on page 100)

System certificate considerations

Keep the following in mind when configuring SSL certificates for your system, especially if you are configuring the system to use one or more certificates that you create yourself:

- Do not allow any of the SSL certificates to expire.
- Adhere to the established best practices for setting up SSL certificates. For example, if you are using wildcards to identify hostnames in an SSL certificate, a wildcard should appear only at the beginning of the hostname, not in the middle.

For information on SSL best practices, see <http://tools.ietf.org/html/rfc5280> and <http://tools.ietf.org/html/rfc6125>.

- Ensure that the DNS name for the system matches the name defined in the certificate.
- When configuring a certificate chain, ensure that all intermediate issuers have the appropriate signing authority permissions so that the entire chain is signed.

Installing a certificate you created

You can create an SSL server certificate by using a third-party tool such as OpenSSL. When creating the certificate, you specify two passwords – one for the PKCS12 object containing the certificate and one for the private key for the certificate. To use the certificate with your system, these passwords must be the same.

When you create your own SSL server certificate, you can choose to have that certificate signed by a certificate authority (CA). In this case, the CA you use may provide you with one or more intermediate certificates. These certificates are used in conjunction with the SSL server certificate you created to establish a certificate chain, an ordered list of certificates in which each certificate is trusted by the next.

To preserve the chain of trust among the certificates, you need to upload the certificates in the correct order. That is, each certificate you upload must be immediately followed by the certificate that signs it. For information on the correct order for the certificate chain, see your CA.



Important: Read and understand the topic "[System certificate considerations](#)" on the previous page before creating your own SSL certificates and especially if you are using an in-house CA.

System Management application instructions

To install your certificates:

1. Click on the **Configuration** panel.
2. Click on **Certificates**.
3. Click on the **Update System Certificate** button.
4. On the **PKCS12** panel, click and drag your certificate into the **Upload Certificate Chain** box.
5. In the **PKCS12 Password** field, type the password for your certificate.
6. Click and drag the certificate into the **Upload Certificate Chain** box.
7. Click on the **Continue** button.
8. Click on the **Accept** button.

Related CLI command(s)

uploadPKCS12Certificate

applyCertificateChanges

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Installing a new self-signed certificate

Your system can generate and install a new self-signed SSL server certificate. The new certificate is good for five years.



Important: If the system is using a self-signed certificate, when you change the hostname name of the system, you need to generate a new SSL certificate. For information on changing the hostname, see "[Setting the system hostname](#)" on page 103.

System Management application instructions

To generate a new self-signed certificate:

1. Click on the **Configuration** panel.

2. Click on **Certificates**.
3. Click on **Update System Certificate**.
4. Click on the **Self-Signed** panel.
5. Click on the **Continue** button.

Your system generates a new self-signed server certificate.

6. Click on the **Accept** button.

Your system installs the new certificate.

7. To continue using the System Management application, log out and then log back in.

Related CLI command(s)

`generateSelfSignedCertificate`

`applyCertificateChanges`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Creating a CSR and installing the returned certificate

SSL server certificates are available from several trusted sources. To obtain a certificate created by a certificate authority (CA), you need to create a certificate signing request (CSR) and give it to the CA. The CA then generates the requested certificate and makes it available to you.

Creating a certificate signing request

You can create a CSR using the System Management application or a third-party tool. When you use the System Management application, the system securely stores the private key needed for installing the returned certificate, so you don't need to save it yourself.

To know exactly what information is required, check with the CA you plan to use.

System Management application instructions

To create a CSR:

1. Click on the **Configuration** panel.
2. Click on **Certificates**.

3. Click on the **System** tab.
4. Click on **Update System Certificate**.
5. Click on the **CSR** panel.
6. Choose **Generate a new certificate signing request** and click on the **Continue** button.
7. Fill in the fields as needed:

- In the **Common Name (CN)** field, type the DNS name of the system preceded by an asterisk (*) and a period (.) (for example, *.system.example.com).

The **Common Name (CN)** field is required.

- In the **Organizational Unit (OU)** field, type the name of the organizational unit that uses the system (for example, the name of a division or a name under which your company does business).
- In the **Organization (O)** field, type the full legal name of your organization.
- In the **Location (L)** field, type the name of the city in which your organization's headquarters are located.
- In the **State/Province (ST)** field, type the full name of the state or province in which your organization's headquarters are located.
- In the **Country (C)** field, type the two-letter ISO 3166-1 abbreviation for the country in which your organization's headquarters are located (for example, US for the United States).

8. Click on the **Generate CSR** button.

The page displays the generated certificate request.

9. Copy and paste the request text into a file and send that file to your CA.
10. Continue to "Installing the certificates returned for a system-generated CSR" on the next page.

Related CLI command(s)

generateCSR

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Installing the certificates returned for a system-generated CSR

In response to a CSR, your CA provides you with an SSL server certificate and any required **intermediate certificates**. These certificates are used in conjunction with the SSL server certificate to establish a **certificate chain**, an ordered list of certificates in which each certificate is trusted by the next. You need to upload and install these certificates on your system.

To preserve the chain of trust among the certificates, you need to upload the certificates in the correct order. That is, each certificate you upload must be immediately followed by the certificate that signs it. For information on the correct order for the certificate chain, see your CA.

System Management application instructions

To install the SSL server certificate and any intermediate certificates returned from a CA:

1. Click on the **Configuration** panel.
2. Click on **Certificates**.
3. Click on the **System** tab.
4. Click on **Update System Certificate**.
5. Click on the **CSR** panel.
6. Select the **I already generated a CSR and obtained a signed certificate** option and click on the **Continue** button.
7. Click and drag the certificate into the **Upload certificate obtained from Certificate Authority** box.
8. Click on the **Accept** button.

Related CLI command(s)

uploadCSR

applyCertificateChanges

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Chapter 5: System management

As an administrator, you play a role in ensuring the continued accessibility and performance of the system. You can use the System Management application, command line, or REST API to manage the system.

Your responsibilities for administering the system include:

- Managing and monitoring system performance and resource usage by configuring how instances are deployed in your infrastructure. For more information, see "[System scaling](#)" on page 106.
- Expanding functionality by writing and installing plugins. For information, see "[Plugins](#)" on page 140.
- Setting up email notifications. For information, see "[Creating email notification rules](#)" on page 68.
- Upgrading the system. For information, see "[Updating the system](#)" on page 146.

Setting the system hostname

After installing your system, you need to configure it with the hostname that you've assigned to it in your corporate DNS environment.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Settings** tab, specify the hostname in the **Cluster Hostname** field.
4. Click on the **Update** button.

Setting the system hostname

Related CLI command(s)

editSecuritySettings

For information on running CLI commands, see "[CLI reference](#)" on page 153.

System scaling

You manage how the system scales by adding or removing instances to the system and also by specifying which services run on those instances.

Instances

An **instance** is a server or virtual machine on which the software is running. A system can have either a single instance or multiple instances. Multi-instance systems have a minimum of four instances.

A system with multiple instances maintains higher availability in case of instance failures. Additionally, a system with more instances can run tasks concurrently and can typically process tasks faster than a system with fewer or only one instance.

A multi-instance system has two types of instances: **master instances**, which run an essential set of services, and non-master instances, which are called **workers**.

For more information, see "[Instances](#)" on page 118.

Services

Each instance runs a configurable set of services, each of which performs a specific function.

In a multi-instance system, services can be distributed across all instances in the system. In a single-instance system, that instance runs all services.

For more information, see [Chapter 1: "Services"](#) on page 9.

Networking

This topic describes the network usage and requirements for both system instances and services.



Note: You can configure the network settings for each service when you install the system. You cannot change these settings after the system is up and running.



Note: If your networking environment changes such that the system can no longer function with its current networking configuration, you need to reinstall the system. See "[Handling network changes](#)" on page 113.

Instance IP address requirements

All instance IP addresses must be static. This includes both internal and external network IP addresses, if applicable to your system.



Important: If the IP address of any instance changes, see "[Handling network changes](#)" on page 113.

Network types

Each of the Product services can bind to one type of network, either **internal** or **external**, for receiving incoming traffic. If your network infrastructure supports having two networks, you may want to isolate the traffic for most system services to a secured internal network that has limited access. You can then leave only the System-Management-application on your external network for user access.

You can use either a single network type for all services or a mix of both types. If you want to use both types, every instance in your system must be addressable by two IP addresses; one on your internal network, and one on your external network. If you use only one network type, each instance needs only one IP address.

Allowing access to external resources

Regardless of whether you're using a single network type or a mix of types, you need to configure your network environment to ensure that all instances have outgoing access to the external resources you want to use, such as:

- The data sources where your data is stored
- Identity providers for user authentication
- Email servers that you want to use for sending email notifications

Ports

Each service binds to a number of ports for receiving incoming traffic.

Before installing system, you can configure the services to use different ports, or use the default values shown in the tables below.

External ports

The following table contains information about the service ports that users use to interact with the system.

On every instance in the system, each of these ports:

- Must be accessible from any network that requires administrative or search access to the system
- Must be accessible from every other instance in the system



Note: Debuggings are accessible only when debug is set to true in `<installation-directory>/config/cluster.config`

Default Port Value	Used by Service	Purpose
80	S3 Gateway	HTTP communication
443 (S3 port)	S3 Gateway	HTTPS communication
8000	System Management application	System Management application GUI
9099	MAPI Gateway	Object Storage Management application GUI
9190	OAuth	OAuth port
12500	Metadata Gateway Raft	RPC communication
12501	Metadata Gateway	RPC communication
12510	Metadata Coordination	RPC communication
14268	Tracing Collector	HTTP port
16686	Tracing Query	HTTP port (APIs and user interface)

Internal ports

This table lists the ports used for intrasystem communication by the services. On every instance in the system, each of these ports:

- Must be accessible from every other instance in the system
- Should not be accessible from outside the system

You can find more information about how these ports are used in the documentation for the third-party software underlying each service. See [Service list](#).

Default Port Value	Used By	Purpose
2181	Synchronization	Primary port used to communicate with the service
2888	Synchronization	Server-server communication
3888	Synchronization	Leader elections
5000	Synchronization	Debugging
5001	System Management application	Debugging
5004	Watchdog	Debugging
5007	Sentinel	Debugging
5050	Cluster Coordination	Primary port used to communicate with the service
5051	Cluster Worker	Primary port used to communicate with the service
5555	Watchdog	Primary port used for inter-service communication
5778	Tracing Agent	Agent HTTP port
6831	Tracing Agent	UDP port
7000	Cassandra	TCP port for commands and data
7199	Cassandra	Used for JMX connections
7203	Kafka	Used for JMX connections

Default Port Value	Used By	Purpose
8005	System Management application	Tomcat shutdown port
8007	Sentinel	Tomcat shutdown port
8022	Watchdog	SSH
8080	Service Deployment	Primary port used to communicate with the service
8081	Chronos	Primary port used to communicate with the service
8889	Sentinel	Primary port used to communicate with the service
9042	Cassandra	Primary port used to communicate with the service
9091	Network Proxy	Primary port used to communicate with the service
9092	Kafka	Primary port used to communicate with the service
9191	Metrics	Primary port used to communicate with the service
9200	Elasticsearch	Used to communicate with Elasticsearch cluster
9201	Elasticsearch	Used to communicate with Elasticsearch nodes
9301	Elasticsearch	Elasticsearch intercluster communication
9600	Logstash	Primary port used to communicate with the service
9601	Logstash	Port used to listen for syslog connections
9750	S3 Gateway	Support
9751	Metadata Gateway	Support
9752	MAPI Gateway	Support
9753	Metadata Cache	Support
9758	Metadata Policy Engine	Support
9760	Metadata Coordination	Support
9990	S3 Gateway	Remote monitoring

Default Port Value	Used By	Purpose
9991	Metadata Gateway	Monitoring
9992	MAPI Gateway	Monitoring
9993	Metadata Cache	Monitoring
9998	Metadata Policy Engine	Monitoring
10000	Metadata Coordination	Monitoring
12000	S3 Gateway	Debugging
12001	Metadata Gateway	Debugging
12002	MAPI Gateway	Debugging
12003	Metadata Cache	Debugging
12004	Metrics	Debugging
12005	Tracing Collector	Debugging
12006	Tracing Query	Debugging
12007	Tracing Agent	Debugging
12008	Metadata Policy Engine	Debugging
12010	Metadata Coordination	Debugging
13300	Metadata Cache	Cache TCP discovery
13370	S3 Gateway	Cache TCP communication
13371	Metadata Gateway	Cache TCP communication
13372	MAPI Gateway	Cache TCP communication
13373	Metadata Cache	Cache TCP communication
13378	Metadata Policy Engine	Cache TCP communication

Default Port Value	Used By	Purpose
13380	Metadata Coordination	Cache TCP communication
13453	Metadata Cache	Cache TCP communication
13500	S3 Gateway	Cache client connector
13501	Metadata Gateway	Cache client connector
13502	MAPI Gateway	Cache client connector
13503	Metadata Cache	Cache client connector
13508	Metadata Policy Engine	Cache client connector
13510	Metadata Coordination	Cache client connector
14267	Tracing Collector	Collecting thrift spans from tracing agents
15050	Cluster Coordination	Local port to which the service directly binds
18000	System Management application	Local port to which the service directly binds
18080	Service Deployment	Local port to which the service directly binds
18889	Sentinel	Local port to which the service directly binds
31000 to 34000	Service Deployment	Port range used by both Service Deployment and Docker for running containers
47000	Cache	TCP cache communication
47008	Metadata Policy Engine	TCP cache communication
47500	Cache	TCP cache discovery
48000	Cache	TCP connector
48500	Cache	Client connector
48508	Metadata Policy Engine	Client connector

Handling network changes

Once your system is deployed, its network infrastructure and configuration should not change. Specifically:

- All instance IP addresses should not change
- All services should continue to use the same ports
- All services and instances should continue to use the same network types

If any of the above change, you will need to reinstall the system.

Safely changing an instance IP address

If you need to change the IP addresses for one or more instances in the system, use this procedure to manually change the IP addresses without risk of data loss.

For each instance whose IP address you need to change:

1. Move all services off of the instance. Distribute those services among all the other instances. For information, see ["Moving and scaling services"](#) on page 39.
2. On the instance from step 1, stop the `run` script using whatever tool or process you used to run it. For example, with `systemd`, run:

```
systemctl stop <service-name>
```
3. Remove the instance from the system. For information, see ["Removing instances"](#) on page 137.
4. Delete the installation directory from the instance.
5. Add the instance back to the system. For information, see ["Adding new instances"](#) on page 123.

After a network change

If a network infrastructure or configuration change occurs that prevents your system from functioning with its current network settings, you need to reinstall all instances in the system.

1. If the System Management application is accessible, back up your system components by exporting a package. For information, see "[Exporting packages](#)" on page 144.
2. On each instance in the system:
 - a. Navigate to the installation directory.
 - b. Stop the `run` script using whatever tool or process you used to run it. For example, with `systemd`, run:


```
systemctl stop <service-name>
```
 - c. Run `bin/stop`
 - d. Run the `setup` script, including the list of master instances:


```
sudo bin/setup -i <ip-address-for-this-instance> -m <comma-separated-list-of-master-instance-IP-addresses>
```
 - e. Run the `run` script using whatever methods you usually use to run scripts.
3. Log into System Management application and use the wizard to setup the system.
4. After the system has been setup, upload your package. For information, see "[Importing packages](#)" on page 145.

Volumes

Volumes are properties of services and job types that specify where and how a services and individual jobs store their data.

You can use volumes to configure jobs and services to store their data in external storage systems, outside of the system instances. This allows data to be more easily backed up or migrated.

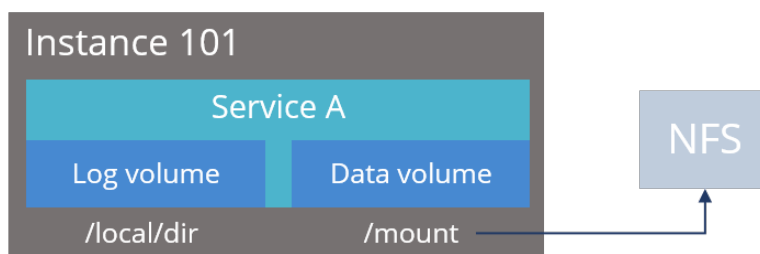
Volumes can also allow services or jobs to store different types of data in different locations. For example, a service may use two separate volumes, one for storing its logs and the other for storing all other data.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

Example

In this example, service A runs on instance 101. The service's Log volume stores data in a directory on the system instance and the service's Data volume stores data in an NFS mount.



Creating and managing volumes

Volumes are separated into these groups, depending on how they are created and managed:

- **System-managed volumes** are created and managed by the system. When you deploy the system, you can specify the volume driver and options that the system should use when creating these volumes.

Once the system is deployed, you cannot change the configuration settings for these volumes.

- **User-managed volumes** can be added to services and job types after the system has been deployed. These are volumes that you manage; you need to create them on your system instances before you can configure a job or service to use them.



Note: As of release 1.3.0, none of the built-in services support adding user-managed volumes.

For more information, see `addVolumesToJobs`.

Volume drivers

When configuring a volume, you specify the volume driver that it should use. The volume driver determines how and where data is stored.

Because services and jobs run in Docker containers on instances in the system, volume drivers are provided by Docker and other third-party developers, not by the system itself. For information on volume drivers you can use, see the applicable Docker or third-party developer's documentation.

By default, all services and jobs do not use volume drivers but instead use the **bind-mount** setting. With this setting, data for each service or job is stored within the system installation directory on each instance where the service runs.

For more information on volume drivers, see the Docker documentation.

For more information:

- On Jobs, see [Chapter 2: "Jobs"](#) on page 45.
- On services, see [Chapter 1: "Services"](#) on page 9.

Viewing volumes

The Administration App shows this information about the Docker volumes used by jobs and services:

- **Name** – The unique identifier for the volume.
- **Type** – Either of these:
 - **System** – The volume is managed automatically for you by the system.
 - **User** – You need to manage the volume yourself.
- **Capacity** – Total storage space available in the volume.
- **Used** – Space used by the job or service.
- **Pool** – The volume category, as defined by the service or job that uses the volume.



Note: Some functions described here are not used with HCP for cloud scale. They are not visible in the System Management application, or have no effect when used.

For each volume, you can also view this information about the volume driver that controls how the volume stores data:

- **Volume driver** – The name of the volume driver.
- **Option/Value** – The command-line options used to create the volume, and their corresponding values. The available options and valid values for those options are determined by the volume driver.

Viewing job volumes

To view the volumes being used by a job:

1. In the System Management application, click on the **Jobs** panel.
2. On the **Job Type** page, click on the job you want.
3. Click on the **Volumes** tab.

Viewing service volumes

To view the volumes being used by a service:

1. In the System Management application, click on the **Services** panel.
2. Click on the service you want.
3. Click on the **Volumes** tab.

Instances

A system is made up of one or more instances of the software. This section includes information on adding and removing instances to the system.

For more information on instances, see "[System scaling](#)" on page 106.

About master and worker instances

Master instances are special instances that run an essential set of services, including:

- System-Management-application service
- Cluster-Coordination service
- Synchronization service
- Service-Deployment service

Non-master instances are called **workers**. Workers can run any services except for those listed above. For information on services, see [Chapter 1: "Services"](#) on page 9.

Single-instance systems have one master instance while multi-instance systems have either one or three master instances.



Important: You cannot add master instances to a system after it's installed. You can, however, add any number of worker instances.

Single-instance systems versus multi-instance systems

A system can have a single instance or can have multiple instances (four or more).

**Note:**

Every instance must meet the minimum RAM, CPU, and disk space requirements. For information, see "[Hardware resources](#)" on the next page.

One instance

A **single-instance system** is useful for testing and demonstration purposes. It requires only a single server or virtual machine and can perform all product functionality.

However, a single-instance system has these drawbacks:

- It has a single point of failure. If the instance hardware fails, you lose access to the system.
- With no additional instances, you cannot choose where to run services. All services run on that one instance.

Multiple instances

A multi-instance system is suitable for use in a production environment because it offers these advantages over a single-instance system:

- You can control how services are distributed across the multiple instances, providing improved service redundancy, scale out, and availability.

For more information, see "[Service list](#)" on page 11.

- A multi-instance system can survive instance outages. For example, with a four-instance system running the default distribution of services, the system can lose one instance and still remain available.

- Performance is improved as work can be performed in parallel across instances.
- You can add additional instances to the system at any time.



Note: You cannot change a single-instance system into a production-ready multi-instance system by adding new instances. This is because you cannot add master instances. **Master instances** are special instances that run a particular set of system services. Single instance systems have one master instance. Multi-instance systems have three.

By adding additional instances to a single-instance system, your system still has only one master instance, meaning there is a single point of failure for the essential services that only a master instance can run.

For more information, see "[Adding new instances](#)" on page 123.

Three-instance system considerations

Three-instance systems should have only a single master instance. If you deploy a three-instance system where all three instances are masters, the system may not have enough resources to do much beyond running the master services. For information on master instances, see "[About master and worker instances](#)" on page 118.

Requirements for running system instances

This section lists the hardware and operating system requirements for running system instances. Also see "[Networking](#)" on page 106 for information on network requirements for both instances and services.

Hardware resources

To install system on on-premises hardware for production use, you must provision at least four instances (nodes) with sufficient CPU, RAM, disk space, and networking capabilities. This table shows the minimum and recommended hardware requirements for each instance in an system system.

Resource	Minimum	Recommended
RAM	32 GB	128 GB
CPU	8-core	24-core

Resource	Minimum	Recommended
Available disk space	500 GB 10k SAS RAID	2000 GB 15k SAS RAID
Network interface controller (NIC)	(1) 10 Gb Ethernet	(2) 10 Gb Ethernet
IP addresses	(1) static	(2) static
Firewall Port Access	Port 443 for S3 API Port 8000 for System Management application GUI Port 9084 for MAPI and Storage Management App GUI	Same
Internal IP Ports	See " Networking " on page 106	Same
Network Time	IP address of time service (NTP)	Same



Important:

Each instance uses all available RAM and CPU resources on the server or virtual machine on which it's installed.

Operating system and Docker requirements

To be a system instance, each server or virtual machine you provide:

- Must run a 64-bit Linux distribution
- Must have Docker version 1.13.1 or later installed



Important: Install the current Docker version suggested by your operating system, unless that version is earlier than 1.13.1. The system cannot run with Docker versions prior to 1.13.1.

This table shows the operating systems and Docker and SELinux configurations that system has been qualified with. For more information, see "[Docker considerations](#)" on the next page and "[SELinux considerations](#)" on page 123.

Operating System	Docker Version	Docker Storage Configuration	SELinux setting
Fedora 27	Docker 1.13.1-58.git87f2fab.el7.x86_64	direct-vm	Enforcing
Red Hat Enterprise Linux 7.4	Docker 1.13.1-58.git87f2fab.el7.x86_64	direct-vm	Enforcing
Ubuntu 16.04-LTS	Docker 17.03.0-ce	aufs	N/A
CentOS 7.4	Docker 18.03.1-ce	overlay2	Enforcing

Docker considerations

- The Docker installation directory on each instance must have at least 20 GB available for storing the system Docker images.
- Make sure that the Docker storage driver is configured correctly on each instance before installing the system.

After installing the system, changing the Docker storage driver requires a reinstallation of the system.

To view the current Docker storage driver on an instance, run:

```
docker info
```

- If you want to enable SELinux on the system instances, you need to use a Docker storage driver that supports it. The storage drivers that SELinux supports differ depending on the Linux distribution you're using. For more information, see the Docker documentation.
- If you are using the Docker `devicemapper` storage driver:
 - Make sure that there's at least 40 MB of Docker metadata storage space available on each instance. The system requires 20 MB to install successfully and an additional 20 MB to successfully update to a later version.

To view Docker metadata storage usage on an instance, run:

```
docker info
```

- On a production system, do not run `devicemapper` in `loop-lvm` mode. This can cause slow performance or, on certain Linux distributions, the system may not have enough space to run.

SELinux considerations

- You should decide whether you want to run SELinux on the new instance. Then, enable or disable it before installing system on the instance.

Enabling or disabling SELinux on an instance requires you to reboot the instance.

To view whether SELinux is enabled on an instance, run:

```
sestatus
```

- If you want to enable SELinux on the system instances, you need to use a Docker storage driver that supports it.

The storage drivers that SELinux supports differ depending on the Linux distribution you're using. For more information, see the Docker documentation.

Supported browsers

- Google Chrome latest
- Mozilla Firefox latest

Time source requirements

If you are installing a multi-instance system, each instance should run NTP (network time protocol) and use the same external time source. For information, see support.ntp.org.

Adding new instances

You may want to add additional instances to the system if:

- You want to improve system performance.
- You are running out of disk space on one or more instances.



Important:

- You cannot add new master instances, only new worker instances.
-

However, these situations may also be improved by adding additional CPU, RAM, or disks to the instances you already have. For guidance, see "[Hardware resources](#)" on page 120.

Before adding a new instance

- Obtain the product installation file. When adding an instance, you unpack and deploy this file on a bare-metal server or a pre-existing Linux virtual machine.
- Record the IP address(es) of at least one of the master instances in the system.

If your system uses internal and external networks, you need to record both the internal and external IP addresses for the master instances.

You can view instance IP addresses on the **Instances** page in the System Management application.

- Ensure that the new instances you are adding meet the minimum hardware, OS, and networking requirements. For information, see "[Requirements for running system instances](#)" on page 120.
- Record the Docker volume drivers currently used by services and jobs across all existing instances. You need to install all of these volume drivers on the new instance that you're adding.

To find the volume drivers currently in use by your system, run this command on each system instance:

```
docker volume ls
```

Take note of each value for the **DRIVER** field.

Step 1: Install Docker

On each server or virtual machine that is to be a system instance:

1. In a terminal window, check whether Docker 1.13.1 or later is installed:

```
docker --version
```

2. If Docker is not installed or if you have a version prior to 1.13.1, install the current Docker version suggested by your operating system.

The installation method you use depends on your operating system. See the [Docker website](#) for instructions.

Step 2: Configure Docker

Configure Docker with settings suitable for your environment. For guidance on configuring and running Docker, see the applicable Docker documentation.



Important:

- The Docker installation directory on each instance must have at least 20 GB available for storing the system Docker images.
- Make sure that the Docker storage driver is configured correctly on each instance before installing the system.

After installing the system, changing the Docker storage driver requires a reinstallation of the system.

To view the current Docker storage driver on an instance, run:

```
docker info
```

- If you want to enable SELinux on the system instances, you need to use a Docker storage driver that supports it. The storage drivers that SELinux supports differ depending on the Linux distribution you're using. For more information, see the Docker documentation.
- If you are using the Docker `devicemapper` storage driver:
 - Make sure that there's at least 40 MB of Docker metadata storage space available on each instance. The system requires 20 MB to install successfully and an additional 20 MB to successfully update to a later version.

To view Docker metadata storage usage on an instance, run:

```
docker info
```

- On a production system, do not run `devicemapper` in `loop-lvm` mode. This can cause slow performance or, on certain Linux distributions, the system may not have enough space to run.

Step 3: Conditionally, configure Docker volume drivers

If any services or jobs on your system are using Docker volume drivers (that is, not the bind-mount setting) for storing data, you need to install those volume drivers on the new instance that you are adding. If you don't, jobs and services may fail to run on the new instance.

Volume drivers are provided by Docker and other third-party developers, not by the system itself. For information on volume drivers, their capabilities, and their valid configuration settings, see the applicable Docker or third-party developer's documentation.

For more information, see "[Volumes](#)" on page 114.

Step 4: Configure mmap count

On each server or virtual machine that is to be a system instance, add this line to the `/etc/sysctl.conf` file:

```
vm.max_map_count = 262144
```

If the line already exists, ensure that the value is greater than or equal to 262144.

Step 5: Conditionally, enable or disable selinux

You should decide whether you want to run SELinux on the new instance. Then, enable or disable selinux on that instance before installing system.

Enabling or disabling SELinux on an instance requires you to reboot the instance.

For information on running system with SELinux enabled, see "[Operating system and Docker requirements](#)" on page 121 and "[SELinux considerations](#)" on page 123.

Step 6: Configure your firewall rules

1. Determine the port values currently used by your system. To do this, on any instance, view the `<installation-directory>/config/network.config` file.
2. On each server or virtual machine that is to be a system instance:
 - a. Edit the firewall rules to allow communication over all network ports that your system currently uses. You do this using a tool such as `firewalld`.
 - b. Restart the server or virtual machine.

Step 7: Install and configure NTP

Install NTP (Network Time Protocol) on the new server or virtual machine and configure it to use the same time source as the other system instances. For information, see support.ntp.org.

Step 8: Run Docker

On each server or virtual machine that is to be a system instance, you need to start Docker and keep it running.

You can use whatever tools you typically use for keeping services running in your environment.

For example, to run Docker using `systemd`:

1. To check if Docker is running, run this command:

```
systemctl status docker
```

2. If Docker is not running, start the `docker` service:

```
sudo systemctl start docker
```

3. Optionally, configure the `docker` service to start automatically when you restart the server or virtual machine:

```
sudo systemctl enable docker
```

Step 9: Unpack the installation package

On each server or virtual machine that is to be a system instance:

1. Retrieve the product installation file and store it in a directory on the server or virtual machine.
2. In the largest disk partition, create a directory. This is the product installation directory.
3. Move the installation package to the product installation directory:

```
mv <path>/<installation-file> /<path>/<installation-directory>
```

4. Navigate to the installation directory:

```
cd /<path>/<installation-directory>
```

5. Unpack the installation package:

```
tar -zxvf <installation-file>
```

6. Run the `install` script in the version-specific directory:

```
sudo ./cluster/<version-number>/bin/install
```

For example:

```
sudo ./cluster/1.0.0.123/bin/install
```

**Notes:**

- Don't change directories after running the `install` script. The following steps are performed in your current directory, not under the version-specific directory.
 - The `install` script can be run only once on each instance. You cannot rerun this script to try to repair or upgrade a system instance.
-

Step 10: Set up networking

On each server or virtual machine that is to be a system instance, edit the `<installation-directory>/config/network.config` file to be identical to the copies of the same file on the existing system instances.

Step 11: Run the setup script on each new instance

On each server or virtual machine that is to be a system instance, run the `setup` script with the applicable options to match the network configuration of the existing system:

- `-i` – The external network IP address for the instance that you're adding.
- `-I` – The internal network IP address for the instance that you're adding.
- `-m` – The external IP address for an existing master instance.
- `-M` – The internal IP address for an existing master instance.

For example, if you're adding an instance to a system that uses both internal and external networks, use this syntax:

```
sudo bin/setup -i <external-ip-for-new-instance> -I <internal-ip-
for-new-instance> -m <list-of-external-master-ips> -M <list-of-
internal-master-ips>
```

Such as:

```
sudo bin/setup -i 192.0.2.4 -I 10.236.1.0 -m
192.0.2.0,192.0.2.1,192.0.2.3 -M 10.236.1.1,10.236.1.2,10.236.1.3
```



Important: For the `-m` and `-M` options:

- Do not specify the IP address for the instance you are trying to add. You cannot add new master instances to an existing system.
- You can optionally specify multiple master IP addresses. If you do, don't separate them with spaces. For example:

```
sudo bin/setup -i 192.0.2.4 -m 192.0.2.0,192.0.2.1,192.0.2.3
```

Example

This table shows the example commands used to create a four-instance system. Each command is entered on a different server or virtual machine that is to be a system instance.

The resulting system in this example contains three master instances, one worker instance, and uses both internal and external networks.

Instances

Instance internal IP	Instance external IP	Master or worker	Command
192.0.2.1	10.236.1.1	Master	sudo bin/setup -i 192.0.2.1 -I 10.236.1.1 -m 192.0.2.1,192.0.2.2,192.0.2.3 -M 10.236.1.1,10.236.1.2,10.236.1.3
192.0.2.2	10.236.1.2	Master	sudo bin/setup -i 192.0.2.2 -I 10.236.1.2 -m 192.0.2.1,192.0.2.2,192.0.2.3 -M 10.236.1.1,10.236.1.2,10.236.1.3
192.0.2.3	10.236.1.3	Master	sudo bin/setup -i 192.0.2.3 -I 10.236.1.3 -m 192.0.2.1,192.0.2.2,192.0.2.3 -M 10.236.1.1,10.236.1.2,10.236.1.3
192.0.2.4	10.236.1.4	Worker	sudo bin/setup -i 192.0.2.4 -I 10.236.1.4 -m 192.0.2.1,192.0.2.2,192.0.2.3 -M 10.236.1.1,10.236.1.2,10.236.1.3

Step 12: Start the application

On each virtual machine that is to be a system instance, start the `run` application script using whatever methods you usually use to run scripts.



Important: Ensure that the method you use can keep the `run` script running and can automatically restart it in case of a server reboot or other availability event.

Here are some examples of how you can start the script:

- **Example 1** – You could run the script in the foreground:

```
sudo bin/run
```

When you run the `run` script this way, the script does not automatically complete. It remains running in the foreground.

- **Example 2** – You could run the script as a service using `systemd`:
 - a. Copy the `<product-name>.service` file to the appropriate location for your OS. For example:

```
cp /<file-path>/bin/<product-name>.service /etc/systemd/system
```

b. Enable and start the search service:

```
sudo systemctl enable <product-name>.service
```

```
sudo systemctl start <product-name>.service
```



Note: When you enable the search service, `systemctl` may display this message:

```
The unit files have no [Install] section. They are not meant to be enabled using systemctl.
```

```
Possible reasons for having this kind of units are:
```

- 1) A unit may be statically enabled by being symlinked from another unit's `.wants/` or `.requires/` directory.
- 2) A unit's purpose may be to act as a helper for some other unit which has a requirement dependency on it.
- 3) A unit may be started when needed via activation (socket, path, timer, D-Bus, udev, scripted systemctl call, ...).

Depending on your OS, the search service may or may not have successfully been enabled.

To avoid this, make sure that you moved the `<product-name>.service` to the appropriate location, typically **`/etc/systemd/system`**.

Once the service starts, the server or virtual machine automatically joins the system as a new instance.

Step 13: Configure services and jobs on the new instances

The system does not automatically begin running services on the instances you've added. You need to manually configure services to run on those new instances. For information, see "[Moving and scaling services](#)" on page 39.

Also, depending on how your jobs are configured, jobs may not run on the new instances that you've added. You need to manually configure jobs to run on them. For information, see "[Configuring where jobs run](#)" on page 46.

Viewing instances

You can use the System Management application, CLI, and REST API to view a list of all instances in the system.

Related topics:

- "[Service list](#)" on page 11

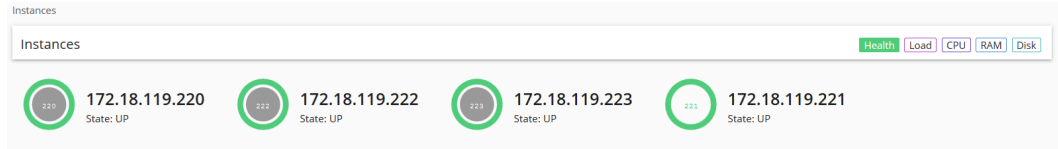
- [Chapter 1: "Services"](#) on page 9

System Management application instructions

Viewing all instances

To view all instances, in the System Management application, click on **Dashboard > Instances**

The page shows all instances in the system. Each instance is identified by its IP address.



This table describes the information shown for each instance.

Property	Description
State	<p>One of these:</p> <ul style="list-style-type: none"> Up – The instance is reachable by other instances in the system. Down – The instance cannot be reached by other instances in the system.
Services	The number of services running on the instance.
Service Units	<p>The total number of service units for all services and job types running on the instance, out of the recommended service unit limit for the instance.</p> <p>An instance with a higher number of service units is likely to be more heavily utilized by the system than an instance with a lower number of service units.</p> <p>The Instances page displays a blue bar for instances running less than the recommended service unit limit.</p> <p>The Instances page displays a red bar for instances running more than the recommended service unit limit.</p> <p>For more information on service units and service unit recommendations, see Service units.</p>

Property	Description
Load Average	The load averages for the instance for the past one, five, and ten minutes.
CPU	The sum of the percentage utilization for each CPU core in the instance.
Memory Allocated	This section shows both: <ul style="list-style-type: none"><li data-bbox="345 499 1409 573">• The amount of RAM on the instance that's allocated to all services running on that instance.<li data-bbox="345 615 1271 646">• The percentage of this allocated RAM to the total RAM for the instance.
Memory Total	The total amount of RAM for the instance.
Disk Used	The current amount of disk space that your system is using in the partition on which it is installed.
Disk Free	The amount of free disk space in the partition in which your system is installed.

Viewing the services running on an instance

To view the services running on an individual instance, in the System Management application:

1. Click on **Dashboard > Instances**.
2. Click on the instance you want.

The page lists all services running on the instance.

For each service, the page shows:

- The service name
- The service state. One of these:
 - **Healthy** – The service is running normally.
 - **Unconfigured** – The service has yet to be configured and deployed.
 - **Deploying** – The system is currently starting or restarting the service. This can happen when:
 - You move the service to run on a completely different set of instances.
 - You repair a service.

For information on viewing the status service operations, see ["Monitoring service operations"](#) on page 62.

- **Balancing** – The service is running normally, but performing some background maintenance operations.
 - **Under-protected** – In a multi-instance system, one or more of the instances on which a service is configured to run are offline.
 - **Failed** – The service is not running or the system cannot communicate with the service.
- **CPU Usage** – The current percentage CPU usage for the service across all instances on which it's running.

- **Memory** – The current RAM usage for the service across all instances on which it's running.
- **Disk Used** – The current total amount of disk space that the service is using across all instances on which it's running.

Related CLI command(s)`getInstance``listInstances`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Removing instances

You would typically remove an instance from your system in these situations:

- You are retiring the hardware on which the instance runs
- The instance is in the Down state and cannot be recovered
- You want to run a system with fewer instances

Step 1: Conditionally, shut down the instance you want to remove

If the instance has already shut down as a result of a failure, the instance is in the Down state. Your system automatically attempts to move all services from that instance to other instances in the system. After all services have been moved, the instance is eligible for removal. Continue to "[Step 2: Remove the shut down instance from the system](#)" on the facing page.

If the instance that you want to remove is in the Up state, you need to shut it down yourself before you can remove it from the system.

Procedure:

1. Move all the services that the instance is currently running to the other instances in the system. For information, see "[Moving and scaling services](#)" on page 39.



Important: Shutting down an instance without first moving its services can cause data loss.

2. If the system has jobs configured to run on only the failed instance, configure those jobs to run on other instances. For information, see "[Configuring where jobs run](#)" on page 46.
3. Stop the `run` script from running. You do this using whatever method you're currently using to run the script.
4. Run this command to stop all system Docker containers on the instance:

```
sudo <installation-directory>/bin/stop
```

5. Delete the system Docker containers:

- a. List all Docker containers:

```
sudo docker ps
```

- b. Note the container IDs for all containers that use a **com.hds.ensemble** or **com.hitachi.aspen** image.

- c. Delete each of those containers:

```
sudo docker rm <container-id>
```

6. Delete the system Docker images:

a. List all Docker images:

```
sudo docker images
```

b. Note the image IDs for all images that use a **com.hds.ensemble** or **com.hitachi.aspen** repository.

c. Delete each of those images:

```
sudo docker rmi <image-id>
```

7. Delete the system installation directory:

```
rm -rf /<installation-directory>
```

Step 2: Remove the shut down instance from the system

System Management application instructions

1. Click on the **Instances** panel.
2. Click on the instance you want to remove.
3. Click on **Remove Instance**.

Related CLI command(s)

```
deleteInstance
```

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Replacing a failed instance

If an instance suffers an unrecoverable failure, you need to replace that instance with a new one.

Steps:

1. In the System Management application, view the **Instances** page to determine whether the failed instance was a master instance.
2. Select a new server or virtual machine to add as a new instance to the system. For information on instance requirements, see "[Requirements for running system instances](#)" on page 120.

3. Remove the failed instance from the system. For information, see "[Removing instances](#)" on page 137.



WARNING! If the failed instance was a master, after removing it, you have only two master instances remaining. If any other instance fails while you are in this state, the system becomes completely unavailable until you add a third master back to the system by completing this procedure.

4. Add the replacement instance to the system. For information, see "[Adding new instances](#)" on page 123.



Important: If the instance you are replacing was a master instance, when you run setup on the replacement instance, the list of masters that you specify for the **-m** option needs to include:

- The IP addresses of the two remaining healthy master instances.
- The IP address of the new instance that you're adding.

For example, in a system with master instance IPs ranging from 192.0.2.1 to 192.0.2.3 and you are replacing instance 192.0.2.3 with 192.0.2.5, run setup with these options:

```
sudo bin/setup -i 192.0.2.5 -m 192.0.2.1,192.0.2.2,192.0.2.5
```

This does not apply when you're replacing a worker instance. In that case, specify the IP addresses of the 3 existing masters.

Plugins

Plugins are modular pieces of code that allow your system to perform specific activities.

Plugins are organized in groups called **plugin bundles**. When adding or removing plugins from your system, you work with plugin bundles, not individual plugins.

Installing new plugin bundles

You can use the System Management application, administrative REST API, or administrative CLI to upload your custom plugin bundles.

For information on writing and packaging your own plugins, see [Writing plugins](#).



Tip: Before uploading your plugin bundles, use the plugin test harness to ensure that your plugin bundle is correctly formatted. For more information, see [Writing plugins](#).



Note: If you encounter errors, make sure you are installing a plugin bundle and not a workflow component .bundle file. For information on those files, see [Importing and exporting components](#).

1. Click on the **Configuration** panel.
2. Click on **Plugins**.
3. Click on the **Upload Bundle** button.
4. Do one of these:
 - If the plugin bundle you want to import is stored on your computer, click and drag the plugin bundle into the **Upload** panel.

If the bundle you uploaded doesn't appear, click on the **Reload Plugins** button.

Related CLI command(s)

uploadPlugin

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Viewing installed plugins

Use the System Management application, REST API, and CLI to view all plugin bundles and individual plugins that have been installed. You can view all individual plugins at the same time or filter the list based on plugin type.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Plugins**.

The **Plugin Bundles** tab shows all installed plugin bundles.

3. To view all individual plugins, click on the **All Plugins** tab.

Related CLI command(s)

listPlugins

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Upgrading plugin bundles

To upgrade plugins, you upload a new version of the bundle that contains those plugins.

You can select which version of the plugin bundle is the active one (that is, the one that connectors or stages will use). If you select the new version, all connectors and stages immediately begin using the new versions of the plugins in the bundle.

You can change the active plugin bundle version at any time. For information on doing this, see "[Setting the active plugin bundle version](#)" below.

System Management application instructions

To install a plugin:

1. Click on the **Configuration** panel.
2. Click on **Plugins**.
3. Click on the **Upload Bundle** button.
4. In the **Upload Plugins** window, drag and drop the new version of the plugin bundle.
5. In the list of plugin bundles, click on the row for the plugin bundle version that you want.

If the bundle you uploaded isn't listed, click on the **Reload Plugins** button.

6. Click on the **Set Active** button.

Related CLI command(s)

uploadPlugin

setPluginBundleActive

Setting the active plugin bundle version

If you've uploaded multiple versions of a plugin bundle, only one version can be active at a time. The active plugin bundle version is the one that the system uses.

System Management application instructions

To set the active plugin version:

1. Click on the **Configuration** panel.
2. Click on **Plugins**.
3. Click on the row for the plugin bundle version that you want.
4. Click on the **Set Active** button.

Related CLI command(s)

setPluginBundleActive

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Deleting plugin bundles

To delete plugins from your system, you delete plugin bundles from the system. You cannot delete individual plugins.

You cannot delete a plugin bundle, or any of its versions, if any of that bundle's plugins are currently in use by the system.

System Management application instructions

To delete a plugin bundle:

1. Click on the **Configuration** panel.
2. Click on **Plugins**.
3. Click on the delete icon () for the plugin bundle you want to remove.

Related CLI command(s)

deletePluginBundle

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Packages

You can back up all of your system configuration by exporting packages. You can back up these package files and use them to restore your configurations in case of a system failure.

Exporting packages

You can export the configurations for system components as package files. You can back up these package files and use them to restore your configurations in case of a system failure.

After exporting a package, you can store it in one of your data sources. When you want to import the package, your system can retrieve it directly from the data source.

For information on:

- Importing packages, see "[Importing packages](#)" on the facing page

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Packages**.
3. Click on **Export**.
4. Under **Customize Package Description**, give your package a name and an optional description.
5. Under **Configuration**, select any configuration items to export.
6. Under **Plugins**, select any plugin bundles to export.
7. Under **Components**, select any available components to export.

If you select one component but not the components it depends on, the System Management application prompts you to add those missing components to the package.

8. Under **Validate**, make sure your package is valid and click on the **Download Package** button.
9. Once your package downloads, click on the **Download Package** button to download it again, or click on the **Finish** button to exit.

Related CLI command(s)

`buildPackage`

`downloadPackage`

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Importing packages

To import a package, you can upload it from your computer or have your system retrieve it from one of your data sources. After you import the package, your system runs a system task to synchronize the package components across all instances in your system.

The system can have only one imported package at a time.



Notes:

- Importing a component that already exists on your system may cause conflicts and should be avoided.
 - You need to manually resolve conflicts with Components, while conflicts with Configuration are handled automatically by the system.
-

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Packages**.
3. Click on **Import**.
4. Do one of these:
 - If the package you want to import is stored on your computer, click and drag the package file into the **Upload Package** panel.
 - If the package you want to import is stored in one of your data sources, click on the **Click to Upload** panel. Then, browse for the package file.
5. Under **Package Description**, review the description and click on the **Continue** button.
6. Under **Configuration**, select any configuration items to import.
7. Under **Plugins**, select any plugin bundles to import.
8. Under **Components**, select any available components to import.
9. Under **Validate**, make sure your package is valid and click on the **Install Package** button.

Setting a login welcome message

Your system starts a system task to install the package components on all instances in the system.

You can monitor the task from the current page or from the **Processes** page.

10. Once the task has completed and all package components have been installed, clicking on the **Complete Install** button will delete the package from the system.

Related CLI command(s)

uploadPackage

loadPackage – (Loads a package from a data connection)

installPackage

getPackageStatus

deletePackage

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Setting a login welcome message

You can use the System Management application, REST API, and CLI to set a welcome message for the System Management application. The message appears on the app's login page.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Security**.
3. On the **Settings** tab, type a message in the **Single Sign-on Welcome Message** field.
4. Click on the **Update** button.

CLI instructions

editSecuritySettings

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Updating the system

You can update system software by uploading new update packages.



Important: Hitachi Vantara does not provide updates or security fixes for the host operating systems running on system instances.

Before updating

In order for a system to be updated:

- All instances and services must be healthy.
- Each service must be running on its recommended number of instances. See "[Service list](#)" on page 11.
- Each instance must have enough disk space for the update.
- All required network ports must be available on each instance.
- There can be no in-progress package uploads or installations. See "[Packages](#)" on page 143.

During an update

- All running jobs are paused.
- System availability considerations:
 - Instances shut down and restart one at a time during the upgrade. Other instances remain online and able to service requests.
 - The System Management application remains available but is in a read-only state. You can monitor the progress of the update, but you cannot make any other changes to the system.



Note: Systems with two instances are more susceptible to availability outages during an update than systems with three or more instances.

Checking update status

As an update runs, you can view its progress on the **Configuration > Update** page. Also on this page, you can view all system events related to system updates.

Results of an update

After an update, the system runs a new version of the software. Additionally:

- If any of the built-in plugins were updated, your system automatically uses the latest versions of those plugins.
- If an existing service is replaced with a new service, the system automatically runs that new, replacement service.
- If any new services were added, you may need to manually configure those services to run on the system instances.

For information on:

- Plugins, see "[Plugins](#)" on page 140
- Configuring where services run, see "[Moving and scaling services](#)" on page 39.

Update errors

If errors occur during an update, the **Update** page displays information about each error and also displays a **Retry** button for starting the update over again. Some errors may not be resolved by restarting the update.

If you encounter errors during an update, contact your authorized service provider.

Update history

For information on viewing the update history for the system, see "[Viewing update history](#)" on the facing page.

New services and components added during an update

A system update may add new services or plugins. You need to manually configure your system to start using these new components; your system does not start using them automatically.

System Management application instructions

To update system:

1. Click on the **Configuration** panel.
2. Click on **Update**.

3. Click on the **Install** tab.
4. Click and drag the file into the **Upload** panel.

The update file is uploaded and system checks to make sure the file is valid. This may take several minutes.

5. On the **Update** page, click on the **View** button in the **Update Status** panel.
6. The **Verify & Apply Update** page displays information about the contents of the update.
7. To start the update, click on the **Apply Update** button.

The system begins checking to make sure the system is ready to be updated. If it isn't, the update stops. In this case, you need to correct the problems before the update can continue.

Related CLI command(s)

getUpdateStatus

installUpdate

deleteUpdate

loadUpdate

uploadUpdate

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Viewing update history

You can view a list of all updates that have previously been applied to your system. For each update, you can view the corresponding version number and the date on which it was installed.

System Management application instructions

1. Click on the **Configuration** panel.
2. Click on **Update**.

The **History** tab lists previously installed versions and when each was installed.

Related CLI command(s)

getUpdateHistory

For information on running CLI commands, see "[CLI reference](#)" on page 153.

Uninstalling the system

To completely uninstall your system, do the following on all instances:

1. Stop the `run` script from running. You do this using whatever method you're currently using to run the script.

2. Run this command to stop all system Docker containers on the instance:

```
sudo <installation-directory>/bin/stop
```

3. Delete the system Docker containers:

- a. List all Docker containers:

```
sudo docker ps
```

- b. Note the container IDs for all containers that use a **com.hds.ensemble** or **com.hitachi.aspen** image.

- c. Delete each of those containers:

```
sudo docker rm <container-id>
```

4. Delete the system Docker images:

- a. List all Docker images:

```
sudo docker images
```

- b. Note the image IDs for all images that use a **com.hds.ensemble** or **com.hitachi.aspen** repository.

- c. Delete each of those images:

```
sudo docker rmi <image-id>
```

5. Delete the system installation directory:

```
rm -rf /<installation-directory>
```

Chapter 6: Best practices

This section contains topics that describe some best practices for administering your system.

Best practices for distributing services

Each service has an associated service unit cost. These costs indicate how computationally expensive one service is to run compared to another. You can use these costs as a guide for how to distribute services across the instances in your system.

Recommendations

- Avoid running multiple services with high service unit costs together on the same instance.
- On master instances, avoid running any services besides those classified as System services.
- To utilize your instances evenly, try to deploy a comparable number of service units on each instance. For more information, see [Service units](#).

Related topics

For information on:

- The services and their costs, see ["Service list"](#) on page 11
- Configuring where a service runs, see ["Moving and scaling services"](#) on page 39

Best practices for maintaining system availability

Run master instances on separate physical hardware

For a multi-instance system, master instances should run on separate physical hardware. If your instances run on virtual machines, run the master instances on separate physical hosts.

Run services on more than one instance

In a multi-instance system, you can choose which and how many instances that each service can run on. For redundancy, you should run each service on more than one instance.

For per-service deployment recommendations, see "[Service list](#)" on page 11.

Chapter 7: Reference

This section contains information about system limits, known issues, license and copyright information, and information on using the command-line interface and REST API.

Troubleshooting

System event issues

For information on viewing system events, see "[System events](#)" on page 63.

Issue	Description/Resolution
<p>The event log contains instances of event id 6007 with this message:</p> <pre>Service <service-name> health check against HTTP /health <port> cannot succeed because it is on a different network than service Cluster-Worker. Health check ignored to prevent service interruption.</pre>	<p>Your system uses both internal and external networks, but the service specified by the event is on a different network type from the Cluster-Worker service. Because of this, the system cannot perform an additional health check on the specified service.</p> <p>You can ignore this event. The additional health check is not required and does not affect the other health checks used to display the service status on the Monitoring > Services page in the System Management application.</p>

CLI reference

The system includes an administrative CLI for system management. This interface allows you to perform all tasks relating to system setup and configuration. Any

administrative activity that you can perform in the System Management application or REST API can be performed through the CLI.

Your system may also include additional, product-specific CLI tools.

Accessing the CLI tools on a system instance

You can access the CLI tools from any instance. To do this:

1. Log in or SSH into a system instance.
2. Navigate to the CLI tool directory:

```
cd <installation-directory>/cli/
```

3. Navigate to the directory for the CLI tool you want. For example:

```
cd admin
```

Accessing the CLI tools from your computer

You can install your system's CLI tools on your Linux computer. To do this, you must have version 1.8 of the Java Runtime Environment (JRE) installed.

Your system's CLI tools are distributed in .tgz files along with the software installation package.

To install a CLI tool:

1. Store the .tgz file in a directory on your computer.
2. Unpack the file on Linux:

```
tar zxvf filename
```

Syntax

The CLI tools have this syntax:

```
<tool-name> [options] [command] [command-specific-options]
```

Options

`-c, --command <command|category>`

Specifies the command you want to run. When used with the `--help` option, displays information about the specified command.

You can also use this option to specify a category of commands when using the `--help` option. Doing this displays information about all commands within the specified category.

`-d, --model-definition <name>`

Returns information about the specified request model. See "[Viewing request models](#)" on page 157.

`--debug`

Includes verbose debug output for troubleshooting purposes.

`-h, --help <all>`

Displays help information. If you specify the **all** argument, displays information on all commands. If you specify the `-c` option, displays information about commands in the specified category.

`-k, --check-ssl-cert <true|false>`

Whether to enable SSL security checking. When false, insecure connections are allowed.

`-m, --model-schema <ModelName>`

Returns the JSON-formatted schema for the specified request model. See "[Viewing request models](#)" on page 157.

`-p, --password <password>`

Password for the specified user account.

`--port`

The port for the system application that supports the CLI tool.

`-r, --realm <realm>`

Security realm where your user account is defined. For information, see "[Adding identity providers](#)" on page 80.

`-s, --server <server>`

The hostname or IP address of a system instance.

`-u, --username <username>`

Username for an account that has permission to access system.

`-V, --version`

Displays the CLI version.

Viewing available commands

- To view all available commands, run:

```
<cli-tool-name> --help all
```

- To view all command categories, run:

```
<cli-tool-name> --help
```

- To view all commands within a category, run:

```
<cli-tool-name> --help -c <category>
```

For example:

```
admincli --help -c instances
```

- To view all information about a single command, run:

```
<cli-tool-name> --help -c <command>
```

For example:

```
admincli --help -c listInstances
```

Viewing request models

Some commands require that you include a JSON-formatted request body along with the command. The command's request model determines how you need to format the request body.

The help command for an individual command indicates what request model it requires.

For example, this help command output indicates that a command to update a service requires a `ServiceUpdateModel` request:

```
# ./admincli -c updateServiceConfig -h

usage: updateServiceConfig

Name:

    updateServiceConfig

Description:

    Configure service instances

Added:

    1.0

Usage:

    admincli -c updateServiceConfig <options>

Options:

    --service-update-model <ServiceUpdateModel>

    File containing JSON text representing a ServiceUpdateModel for the
    command updateServiceConfig. Use the -m and -d options to retrieve
    information on request and response models.
```

Viewing request model information

To view detailed information about the contents of a request model, run:

```
<cli-tool-name> -d <ModelName>
```

For example:

```
admincli -d ServiceUpdateModel
```

Viewing request model formatting

To view the JSON format for the request model, run:

```
<cli-tool-name> -m <ModelName>
```

Editing configuration preferences

You can use the CLI tool's `.conf` file to specify settings to use every time you run a CLI command.

The CLI configuration file has this format:

```
{
  "defaultSettings": {
    "checkSSLCert": "[false|true]", (optional)
    "server": "<hostname>", (optional)
    "realm": "[local|<security-realm-name>]", (optional)
    "username": "<your-username>", (optional)
    "password": "<your-password>" (optional)
  }
}
```

For example, with the following configuration, all commands:

- Are run against the `system.example.com` system
- Check the SSL certificate for the system before connecting
- Uses the `exampleUsersEast` security realm to authenticate the specified username and password

```
{
  "defaultSettings": {
    "checkSSLCert": "true",
    "server": "system.example.com",
    "realm": "exampleUsersEast"
  }
}
```

File location

You can configure CLI preference by editing the existing `.conf` files in the CLI installation directory.

The options you specify explicitly in a CLI command override the options specified in the `.conf` file.

System error responses

If a CLI request reaches the system and the system returns an error, the CLI response contains:

- An HTTP status code
- Conditionally, a product-specific error code
- A JSON-formatted error response body

HTTP status codes

This table describes the typical reasons why these HTTP status codes are returned.

Status code	Description
400 (Bad Request)	<p>The request body contains one or more of these:</p> <ul style="list-style-type: none"> • An invalid entry • An invalid value for an entry • Invalidly formatted JSON <p>If the request includes a UUID, the UUID may be invalidly formatted.</p>
401 (Unauthorized)	The provided credentials are invalid.
403 (Forbidden)	You do not have permission to perform the request.
404 (File not found)	The resource you are trying to retrieve or edit cannot be found.
409 (Conflict)	The resource you are trying to create already exists.
500 (Server Error)	The system experienced an error.
599 (Network Connection Timeout Error)	The CLI request timed out while attempting to connect to the system or one of its instances.

Product-specific error codes

Some CLI requests return product-specific error codes in addition to an HTTP status code. These error codes are listed in the `errorCodes` field in the JSON response body. This table describes these error codes.

Error code	Description
4000	SSL certificate not trusted.

JSON response body

Error response bodies have this format:

```
{
  "statusCode": <HTTP-status-code>,
  "errorCode": <product-specific-error-code>,
  "errorMessage": <message>,
  "errorProperties": [
    {
      "name": <error-property>,
      "message": <error-property-message>
    }
  ]
}
```

REST API reference

Your system provides a RESTful API that you can use for writing applications that manage the system. Anything you can do in the System Management application can also be performed using the REST API.

Information is available about the APIs in two places:

- For information about system management APIs, see the topics in this Help document.
- For information about Object Storage Management and S3 APIs, see the Help for the Object Storage Management application.

Input and output formats

The REST API accepts and returns JSON.

Access and authentication

To use the REST API, you need a user account that has permission to perform the actions you want. For information on assigning permissions to users, see ["Roles"](#) on page 88.

Requesting an access token

Once you have a user account, you need to request an authentication token from the system. To do this, you send an HTTP POST request to the `/auth/oauth` endpoint.

When you generate a new access token, a refresh token also gets generated automatically.

Here's an example using the cURL command-line tool:

```
curl -ik -X POST https://mysystem.example.com:/auth/oauth/ \
-d grant_type=password \
-d username=user1 \
-d password=password1 \
-d scope=* \
-d client_secret=my-client \
-d client_id=my-client \
-d realm=marketingUsers
```

In response to this request, you receive a JSON response body containing an `access_token` field. The value for this field is your token. For example:

```
{"access_token" : "eyJr287bjle..."}
```



Notes:

- To get a list of security realms for the system, send an HTTP GET request to the `/setup` endpoint. For example, to do this with cURL:

```
curl -k -X GET --header 'Accept: application/json'
'https://mysystem.example.com:<System-Management-application-
port>/api/admin/setup'
```

For information on configuring security realms, see ["Adding identity providers"](#) on page 80.

- To get an access token for the local admin user account, you can omit the realm option for the request, or specify a realm value of **Local**.
-

Submitting your access token

You need to specify your access token as part of all REST API requests that you make. You do this by submitting an Authorization header along with your request. Here's an example that uses cURL:

```
curl -X GET --header "Accept:application/json"
https://mysystem.example.com:<System-Management-application-
port>/api/admin/instances --header "Authorization: Bearer
eyJr287bjle..."
```

Changing a password

You can use the REST API to change your system's password using the following cURL command.

```
$1=<server-name>
```

```
$2=<current-password>
```

```
$3=<new-password>
```

```
TOKEN=$(curl -ik -X POST https://$1.mysystem.com:8000/auth/oauth/ -d
grant_type=password -d username=admin -d password=$2 -d scope=* -d
client_secret=hci-client -d client_id=hci-client -d realm=local 2>&1
| grep access_token | awk -F: '{print $2}' | awk -F\" '{print $2}')
```

```
curl -v -X POST --header 'Content-Type: application/json' --header
"Authorization: Bearer $TOKEN" https://$1.mysys-
tem.com:8000/api/admin/setup/password -d '{"password": "'"$3"'"}'
```

Viewing and using REST API methods

Your system provides web-based documentation pages where you can view all supported REST API methods, including the request bodies, request URLs, response bodies, and return codes for each. You can also use these pages to run each REST API method.

Viewing the API documentation

To view the REST API documentation, in the System Management application, click on the help icon (❓). Then click on **REST API - Admin**.

To view the HCP for cloud scale API documentation, click REST API in the Object Storage Management app.

Making requests

You can use the REST API documentation pages to experiment with the REST API.



Note: Any requests you submit on the REST API page take effect on the system.

To use the REST API page to run a method:

1. Click on the row for the method you want.
2. If the method you want requires that you specify a UUID:
 - a. Click on the row for the GET method for the resource type that you want.
 - b. Click on the **Try it Out!** button.
 - c. In the JSON response body, copy the value for the **uuid** field for the resource that you want.



Note: If you specify UUIDs when creating resources, the UUIDs are ignored.

3. If the method you want requires that you specify a request body:
 - a. In the **Parameters** section, under **Model Schema**, click inside the JSON text box.

The JSON text is added to the **Value** field.

- b. Edit the JSON in the **Value** field.



Note: Some methods may require other information in addition to or instead of UUIDs or JSON-formatted text. Some require particular string values or require that you browse for and select a file to upload.

4. Click on the **Try it out!** button.

Error responses

When an API request fails, the API returns:

- An HTTP status code
- Conditionally, a system-specific error code
- A JSON-formatted error response body

HTTP status codes

This table describes the typical reasons why these HTTP status codes are returned. For information on the status codes for a particular method, view the system REST API web interface (see "[Viewing and using REST API methods](#)" on page 166).

Status code	Description
400 (Bad Request)	The request body contains one or more of these: <ul style="list-style-type: none"> • An invalid entry • An invalid value for an entry • Invalidly formatted JSON If the request includes a UUID, the UUID may be invalidly formatted.
403 (Forbidden)	You do not have permission to perform the request.
404 (File not found)	The resource you are trying to retrieve or edit cannot be found.
409 (Conflict)	The resource you are trying to create already exists.
500 (Server Error)	The system experienced an error.

System-specific error codes

Some API requests return system-specific error codes in addition to an HTTP status code. These error codes are listed in the `errorCodes` field in the JSON response body. This table describes these error codes.

Error code	Description
4000	SSL certificate not trusted.

JSON response body

REST API error responses have this format:

```
{
  "statusCode": <HTTP-status-code>,
  "errorCode": <system-specific-error-code>,
  "errorMessage": <message>,
  "errorProperties": [
    {
      "name": <error-property>,
      "message": <error-property-message>
    }
  ]
}
```


Hitachi Vantara



Corporate Headquarters

2845 Lafayette Street

Santa Clara, CA 95050-2639 USA

www.HitachiVantara.com | community.HitachiVantara.com

Regional Contact Information

Americas: +1 866 374 5822 or info@hitachivantara.com

Europe, Middle East and Africa: +44 (0) 1753 618000 or info.emea@hitachivantara.com

Asia Pacific: +852 3189 7900 or info.marketing.apac@hitachivantara.com