

Hitachi Content Platform for Cloud Scale

v2.4.0

HCP for Cloud Scale Administration Guide

This document contains instructions for configuring, managing, monitoring, and troubleshooting the Hitachi Content Platform for cloud scale (HCP for cloud scale) software, storage components, and users. It describes the HCP for cloud scale Object Storage Management and System Management applications.

© 2020, 2022 Hitachi Vantara LLC. 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 LLC (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 LLC 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 LLC.

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 your 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.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, Lotus, MVS, OS/390, PowerHA, PowerPC, RS/6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Edge, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft corporate logo, the Microsoft Edge 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 website are properties of their respective owners.

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found at <https://www.hitachivantara.com/en-us/company/legal.html>.

Contents

Preface	12
About this document.....	12
Intended audience.....	12
Product version.....	12
Release notes.....	12
Related documents.....	13
Document conventions.....	13
Accessing product documentation.....	15
Getting help.....	15
Comments.....	15
Chapter 1: Getting started	16
Introducing HCP for cloud scale.....	16
Storage components, buckets, and objects.....	16
Data access.....	18
Data access control.....	18
Data security.....	18
Bucket synchronization.....	20
Object locking.....	20
Capacity monitoring.....	21
Chargeback reports.....	23
S3 Select.....	23
S3 event notification.....	24
Supported limits.....	24
High availability.....	25
Scalability of instances, service instances, and storage components....	25
Site availability.....	26
Service availability.....	26
Metadata availability.....	27
Object data availability.....	27
Network availability.....	27
Failure recovery.....	27
Instance failure recovery.....	28
Service failure recovery.....	28

- Storage component failure recovery..... 28
- Support for the Amazon S3 API..... 28
- HCP for cloud scale APIs..... 35
 - Object Storage Management API..... 36
 - System Management API..... 36
- Security and authentication..... 37
 - User accounts..... 37
 - API access..... 38
 - Network isolation and port mapping..... 38
- Logging in..... 39
 - HCP for cloud scale applications..... 40
 - Switching between applications..... 40
- Providing unseal keys to KMS service..... 41
- Serial number..... 41
 - Entering your serial number..... 42
 - Object Storage Management application instructions..... 42
 - Related REST API methods..... 42
 - Displaying your serial number..... 42
 - Object Storage Management application instructions..... 42
 - Related REST API methods..... 42
- License..... 42
 - Uploading your license..... 43
 - Object Storage Management application instructions..... 43
 - Related REST API methods..... 43
- Enabling encryption..... 43
 - Object Storage Management application instructions..... 44
 - Related REST API methods..... 45
- Defining subdomain for S3 Console application..... 45
- About page..... 46
- Chapter 2: Storage components..... 47**
 - Adding a storage component..... 47
 - Object Storage Management application instructions..... 47
 - Related REST API methods..... 49
 - Verifying a storage component..... 49
 - Object Storage Management application instructions..... 49
 - Modifying a storage component..... 49
 - Object Storage Management application instructions..... 49
 - Related REST API methods..... 50
 - Activating a storage component..... 50
 - Object Storage Management application instructions..... 50
 - Related REST API methods..... 51

Deactivating a storage component.....	51
Object Storage Management application instructions.....	51
Related REST API methods.....	51
Marking a storage component as read-only.....	51
Object Storage Management application instructions.....	52
Related REST API methods.....	52
Marking a storage component as read-write.....	52
Object Storage Management application instructions.....	52
Related REST API methods.....	53
Viewing storage components.....	53
Object Storage Management application instructions.....	53
Related REST API methods.....	53
Displaying storage component analytics.....	54
Displaying counts of storage components.....	56
Object Storage Management application instructions.....	56
Related REST API methods.....	56
Metrics.....	57
Displaying the active object count.....	57
Object Storage Management application instructions.....	57
Displaying metrics.....	57
Object Storage Management application instructions.....	57
Available metrics.....	58
Examples of metric expressions.....	70
Tracing requests and responses.....	72
Displaying traces.....	72
Traceable operations.....	72
Chapter 3: Services.....	78
Service categories.....	78
HCP for cloud scale services.....	78
Viewing services.....	93
Viewing all services.....	93
Viewing individual service status.....	94
Related CLI commands.....	95
Related REST API methods.....	95
Listing service ports.....	95
Managing services.....	96
Moving and scaling services.....	96
Relocating services.....	97
Related CLI commands.....	99
Related REST API methods.....	99
Scaling Metadata Gateway instances.....	99

Configuring service settings.....	100
Configuring service settings.....	100
Related CLI commands.....	100
Related REST API methods.....	100
Repairing services.....	100
Configuring TLS cipher suite.....	101
Avoiding Message Queue shutdown.....	101
Service restart after internal network interruption.....	102
Chapter 4: Monitoring.....	103
Monitoring instances.....	103
Viewing all instances.....	103
Viewing the services running on an instance.....	104
Related CLI commands.....	105
Related REST API methods.....	105
Monitoring services.....	105
Viewing all services.....	106
Viewing individual service status.....	106
Related CLI commands.....	107
Related REST API methods.....	107
Monitoring processes.....	107
Monitoring service operations.....	107
Related CLI commands.....	108
Related REST API methods.....	108
Monitoring system processes.....	108
Related CLI commands.....	108
Related REST API methods.....	108
Monitoring objects.....	109
Generating a system chargeback report.....	109
Related REST API methods.....	109
Generating a user chargeback report.....	109
Related REST API methods.....	109
System events.....	109
Related CLI commands.....	109
Related REST API methods.....	110
HCP for cloud scale events.....	110
Alerts.....	116
Viewing alerts.....	121
Object Storage Management application instructions.....	121
Related CLI commands.....	122
Related REST API methods.....	122
Related REST API methods.....	122

Email notification rules.....	122
Creating email notification rules.....	123
Related CLI commands.....	124
Related REST API methods.....	124
Creating syslog notification rules.....	124
Creating syslog notification rules.....	125
Related CLI commands.....	126
Related REST API methods.....	126
Logs and diagnostic information.....	126
Log levels.....	126
Log management.....	127
Retrieving logs and diagnostic information.....	127
Default log locations.....	128
Chapter 5: Security.....	132
Granting access to users.....	132
Setting the session timeout limit.....	132
Related CLI commands.....	132
Related REST API methods.....	133
Setting the refresh token timeout limit.....	133
Related CLI commands.....	133
Related REST API methods.....	133
Setting the CORS Allowed Origins.....	133
Related CLI commands.....	134
Related REST API methods.....	134
Identity providers.....	134
Adding identity providers.....	134
Related CLI commands.....	134
Related REST API methods.....	135
Identity provider configuration settings.....	135
User information caching.....	138
Related CLI commands.....	138
Related REST API methods.....	138
Viewing identity providers.....	138
Related CLI commands.....	139
Related REST API methods.....	139
Deleting identity providers.....	139
Related CLI commands.....	139
Related REST API methods.....	139
Groups.....	139
Adding groups.....	140
Related CLI commands.....	140

Related REST API methods.....	140
Viewing groups.....	140
Related CLI commands.....	141
Related REST API methods.....	141
Assigning roles to groups.....	141
Related CLI commands.....	141
Related REST API methods.....	141
Deleting groups.....	141
Related CLI commands.....	142
Related REST API methods.....	142
Roles.....	142
Creating roles.....	142
Related CLI commands.....	143
Related REST API methods.....	143
Viewing roles.....	143
Related CLI commands.....	143
Related REST API methods.....	144
Editing roles.....	144
Related CLI commands.....	145
Related REST API methods.....	145
Deleting roles.....	145
Related CLI commands.....	145
Related REST API methods.....	145
Permissions.....	145
Revoking user account credentials.....	149
Changing a local user's account password.....	151
Related CLI commands.....	151
Related REST API methods.....	151
Certificates.....	152
Viewing installed certificates.....	152
Related CLI commands.....	153
Related REST API methods.....	153
Client certificates.....	153
Client certificate considerations.....	153
Retrieving the client certificate for an HCP S Series Node.....	153
Uploading client certificates manually.....	154
Related CLI commands createCertificate.....	154
Related REST API methods.....	155
Changing the system certificate.....	155
System certificate considerations.....	155
Installing a certificate you created.....	155

Installing a new self-signed certificate.....	156
Creating a CSR and installing the returned certificate.....	157
Chapter 6: System management.....	160
Setting host name.....	160
Changing host name.....	160
Related CLI commands.....	161
Related REST API methods.....	161
System scaling.....	161
Networking.....	162
Load balancing.....	164
Script to enable S3 traffic redirection.....	164
Script to disable S3 traffic redirection.....	165
Script to list S3 traffic redirection.....	166
Handling network changes.....	167
Safely changing an instance IP address.....	167
After a network change.....	168
Volumes.....	168
Viewing volumes.....	169
Viewing service volumes.....	170
Instances.....	170
About master and worker instances.....	170
Single-instance systems versus multi-instance systems.....	171
Requirements for running system instances.....	172
Hardware requirements.....	172
Software requirements.....	172
Operating system and Docker minimum requirements.....	173
Operating system and Docker qualified versions.....	173
Docker considerations.....	173
SELinux considerations.....	174
Supported browsers.....	174
Time source.....	174
Adding new instances.....	174
Before adding a new instance.....	175
Install Docker on each server or virtual machine	175
Configure Docker on each server or virtual machine.....	175
(Optional) Configure Docker volume drivers.....	176
Configure maximum map count setting.....	176
Optional: Enable or disable SELinux on each server or virtual machine.....	176
Configure the firewall rules on each server or virtual machine.....	177
Install and configure NTP.....	177

Run Docker on each server or virtual machine.....	177
Unpack the installation package.....	177
Set up networking.....	178
Run the setup script on each server or virtual machine.....	179
Start the application on each server or virtual machine.....	181
Configure services and jobs on the new instances.....	181
Viewing instances.....	181
Viewing all instances.....	182
Viewing the services running on an instance.....	183
Related CLI commands.....	183
Related REST API methods.....	183
Removing instances.....	184
(Optional) Shut down the instance you want to remove.....	184
Remove the shut-down instance from the system.....	185
Replacing a failed instance.....	185
Plugins.....	186
Viewing installed plugins.....	186
Related CLI commands.....	186
Related REST API methods.....	186
Upgrading plugin bundles.....	187
Related CLI commands.....	187
Related REST API methods.....	187
Setting the active plugin bundle version.....	187
Related CLI commands.....	188
Related REST API methods.....	188
Deleting plugin bundles.....	188
Related CLI commands.....	188
Related REST API methods.....	188
Packages.....	188
Exporting packages.....	189
Related CLI commands.....	189
Related REST API methods.....	189
Importing packages.....	189
Related CLI commands.....	190
Related REST API methods.....	190
Setting a login welcome message.....	191
Related CLI commands.....	191
Related REST API methods.....	191
Updating the system.....	191
Applying a system update.....	192
Related CLI commands.....	193

Related REST API methods.....	193
Viewing update history.....	193
Related CLI commands.....	194
Related REST API methods.....	194
Removing the system.....	194
Chapter 7: Best practices.....	196
Best practices for system sizing and scaling.....	196
Best practices for maintaining system availability.....	198
Chapter 8: Reference.....	199
Troubleshooting.....	199
CLI reference.....	200
Accessing the CLI tools on a system instance.....	200
Installing CLI tools on your computer.....	200
Syntax.....	201
Viewing available commands.....	202
Viewing request models.....	202
Editing configuration preferences.....	203
System error responses.....	204
REST API Reference.....	205
Getting started with the management APIs.....	205
Input and output formats.....	206
Access and authentication.....	206
Requesting and submitting an XSRF token.....	207
Viewing and using MAPI methods.....	208
Including XSRF tokens in MAPI Swagger requests.....	209
Error responses.....	210

Preface

About this document

This document contains instructions for configuring, managing, monitoring, and troubleshooting the Hitachi Content Platform for cloud scale (HCP for cloud scale) software, storage components, and users. It describes the HCP for cloud scale Object Storage Management and System Management applications.

Intended audience

This document is intended for people who are configuring, managing, or monitoring HCP for cloud scale systems. It assumes you have some experience using computer software through a graphical user interface (GUI) or application programming interface (API).

Product version

This document applies to v2.4.0 of Hitachi Content Platform for cloud scale.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document. Release notes are available on the Hitachi Vantara Support Website: <https://knowledge.hitachivantara.com/Documents>.

Related documents

The following list describes documents containing information about v2.4.0 of HCP for cloud scale. You should have these documents available before using the product. Refer to the latest version of the *Hitachi Content Platform for Cloud Scale Release Notes* for information on document version numbers.

- *Hitachi Content Platform for Cloud Scale Release Notes* (RN-HCPCS004): This document is for customers and describes new features, product documentation, and resolved and known issues, and gives other useful information about this release of the product.
- *Installing Hitachi Content Platform for Cloud Scale* (MK-HCPCS002): This document gives you the information required to install or update the HCP for cloud scale software.
- *Hitachi Content Platform for Cloud Scale Administration Guide* (MK-HCPCS008): This document explains how to use the HCP for cloud scale Object Storage Management and System Management applications to configure and operate a common object storage interface for clients to interact with; configure HCP for cloud scale for users; enable and disable system features; and monitor the system and its connections.
- *Hitachi Content Platform for Cloud Scale S3 Console Guide* (MK-HCPCS009): This document is for end users and explains how to use the HCP for cloud scale S3 Console application to use S3 credentials and to simplify the process of creating, monitoring, and maintaining S3 buckets and the objects they contain.
- *Hitachi Content Platform for Cloud Scale Management API Reference* (MK-HCPCS007): This document is for customers and describes the object storage management application programming interface (API) methods available for customer use.





Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	<ul style="list-style-type: none"> ▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click OK. ▪ Indicates emphasized words in list items.
<i>Italic</i>	<ul style="list-style-type: none"> ▪ Indicates a document title or emphasized words in text. ▪ Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: <code>pairdisplay -g group</code> <p>(For exceptions to this convention for variables, see the entry for angle brackets.)</p>

Convention	Description
Monospace	Indicates text that is displayed on screen or entered by the user. Example: <code>pairdisplay -g oradb</code>
< > angle brackets	Indicates variables in the following scenarios: <ul style="list-style-type: none"> Variables are not clearly separated from the surrounding text or from other variables. Example: <pre>Status-<report-name><file-version>.csv</pre> Variables in headings.
[] square brackets	Indicates optional values. Example: [a b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a b } indicates that you must choose either a or b.
vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a b] indicates that you can choose a, b, or nothing. { a b } indicates that you must choose either a or b.

This document uses the following icons to draw attention to information:

Icon	Label	Description
	Note	Calls attention to important or additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Accessing product documentation

Product user documentation is available on the Hitachi Vantara Support Website: <https://knowledge.hitachivantara.com/Documents>. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Getting help

The [Hitachi Vantara Support Website](https://support.hitachivantara.com/en_us/contact-us.html) is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to the Hitachi Vantara Support Website for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

[Hitachi Vantara Community](https://community.hitachivantara.com) is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send us your comments on this document to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Chapter 1: Getting started

Hitachi Content Platform for cloud scale (HCP for cloud scale) is a software-defined object storage solution that is based on a massively parallel microservice architecture and is compatible with the Amazon Simple Storage Service (S3) application programming interface (API).

Introducing HCP for cloud scale

HCP for cloud scale is especially well suited to service applications requiring high bandwidth and compatibility with the Amazon S3 API.

HCP for cloud scale can federate S3 compatible storage from virtually any private or public source and present the combined capacity in a single, centrally managed, global name space.

You can install HCP for cloud scale on any server, in the cloud or on premise, that supports the minimum requirements.

HCP for cloud scale supports S3 event notification through a graphical user interface or through GET and PUT Bucket Notification configuration.

HCP for cloud scale lets you manage and scale storage components. You can add storage components, monitor their states, and take them online or offline for purposes of maintenance or repair. HCP for cloud scale provides functions to send notification of alerts, track and monitor throughput and performance, and trace actions through the system.

Storage components, buckets, and objects

Storage components

A *storage component* is an Amazon S3 compatible storage provider, running independently, that HCP for cloud scale manages as a back end to store object data. To an S3 client using HCP for cloud scale, the existence, type, and state of storage components are transparent.

HCP for cloud scale supports the following storage systems:

- Amazon S3
- Hitachi Content Platform (HCP)
- HCP S Series Node
- Any Amazon S3 compatible storage service

Buckets

A *bucket* is a logical collection of secure data objects that is created and managed by a client application. An HCP for cloud scale bucket is modeled on a storage service bucket. HCP for cloud scale uses buckets to manage storage components. You can think of an HCP for cloud scale system as a logical collection of secure buckets.

Buckets have associated metadata such as ownership and lifecycle status. HCP for cloud scale buckets are owned by an HCP for cloud scale user and access is controlled on a per-bucket basis by Amazon access control lists (ACL) supporting the S3 API. Buckets are contained in a specific region; HCP for cloud scale supports one region.



Note:

1. HCP for cloud scale buckets are not stored in storage components, so HCP for cloud scale clients can create buckets even before adding storage components.
2. Storage component buckets are created by storage component administrators and are not visible to HCP for cloud scale clients.
3. To empty and reuse a bucket, don't just delete the bucket and create a new one with the same name. After a bucket is deleted, the name becomes available for anyone to use and another account might take it first. Instead, empty and keep the bucket.

Objects

An *object* consists of data and associated metadata. The metadata is a set of name-value pairs that describe the object. Every object is contained in a bucket. An object is handled as a single unit by all HCP for cloud scale transactions, services, and internal processes.

For information about Amazon S3, see [Introduction to Amazon S3](#).

S3 Console application

HCP for cloud scale includes an S3 Console application that provides convenient functions for bucket users as an alternative to using S3 API methods:

- Obtaining S3 credentials
- Managing bucket synchronization, policies, and rules
- Creating S3 event notifications to synchronize buckets
- Managing objects in buckets, singly and in bulk

For more information, see the *S3 Console Guide*.

Strongly consistent object listing

HCP for cloud scale supports strong consistency in object listing. After a write, upload, or delete operation, a list operation shows the changes immediately. Strong consistency supports big-data analytics applications and applications originally written for storage environments of smaller scale.

Data access

HCP for cloud scale supports the Amazon S3 API, which lets client applications store and retrieve unlimited amounts of data from configured storage services.

Data access control

HCP for cloud scale uses ownership and access control lists (ACLs) as data access control mechanisms for the S3 API.

Ownership is implemented as follows:

- An HCP for cloud scale bucket is owned by the user who creates the bucket and the owner cannot be changed.
- A user has full control of the buckets that user owns.
- A user has full control of the objects that user creates.
- A user can list only the buckets that user owns.

ACLs allow the assignment of privileges (read, write, or full control) for access to buckets and objects to other user accounts besides the owner's.

Data security

HCP for cloud scale supports encryption of data sent between systems (that is, data "in flight") and, as a licensed feature, data stored persistently within the system (that is, data "at rest").

Certificate management

HCP for cloud scale uses Secure Sockets Layer (SSL) to provide security for both incoming and outgoing communications. To enable SSL security, two types of certificates are needed:

- System certificate: the certificate that HCP for cloud scale uses for its GUIs and APIs (for incoming communications)
- Client certificates: the certificates of IdPs, storage components, and SMTP servers (for outgoing communications)

When the HCP for cloud scale system is installed, it generates and automatically installs a self-signed SSL server system certificate. This certificate is not automatically trusted by web browsers. You can choose to trust this self-signed certificate or replace it by using one of these options:

1. Upload a PKCS12 certificate chain and password and apply it as the active system certificate.
2. Download a certificate signing request (CSR) and then use it to obtain, upload, and apply a certificate signed by a certificate authority (CA).
3. Generate a new self-signed certificate and apply it as the active system certificate.

For client certificates, upload the certificate of each client HCP for cloud scale needs to access using SSL.

You can manage certificates, and view details of installed certificates, using the System Management application.

Data-in-flight encryption

HCP for cloud scale supports data-in-flight encryption for the HTTPS protocol for all external communications. Data-in-flight encryption is always enabled for these data paths:

- S3 API (HTTP is also enabled on a different port)
- Management API
- System Management application graphical user interface (GUI)
- Object Storage Management application GUI

You can enable or disable data-in-flight encryption for the data paths between HCP for cloud scale and:

- An identity provider (IdP) server
- Each application using TLS or SSL
- Each managed storage component
- Each SMTP server using SSL or STARTTLS

Communication between HCP for cloud scale instances does not use data-in-flight encryption. Depending on the security needs, you might need to set up an isolated internal network for HCP for cloud scale at the site.

Data-at-rest encryption

HCP for cloud scale stores these kinds of data persistently:

- HCP for cloud scale services data
- HCP for cloud scale metadata and user-defined metadata
- User data (object data)

The first two kinds of data are handled by the hardware on which HCP for cloud scale instances are installed. If needed, you can install HCP for cloud scale on servers with encrypted disks.

Object data is handled by storage components. HCP for cloud scale supports system-wide encryption, using AWS SDK client-side encryption and strong encryption ciphers, as a licensed feature. Encryption and decryption are transparent to users. Each storage component has a separate master key. Storage components that use hardware acceleration for encryption and decryption are supported.

To manage encryption master keys, HCP for cloud scale supports the HashiCorp Vault key management system (KMS) through a KMS client that is automatically deployed as a service when encryption is enabled. After you set up a Vault server, you can enable encryption support on HCP for cloud scale as a global setting and then manage the encryption client service as needed. For information about Vault and how to set up a server, see <https://www.hashicorp.com/products/vault>.



Caution: Once enabled, encryption can't be disabled.

As an alternative, you can use individual storage components that support data-at-rest encryption. Storage components can self-manage their keys, or HCP for cloud scale can facilitate keys you supply following the Amazon S3 API specification.

Bucket synchronization

Bucket synchronization to a bucket (*bucket sync-to* or mirroring) allows automatic, asynchronous copying of objects in a bucket in an HCP for cloud scale system to an external storage system. Bucket synchronization from a bucket (*bucket sync-from* or mirroring back) allows automatic, asynchronous copying of objects in a bucket in an external storage system to an HCP for cloud scale bucket. Objects larger than 5 GB are synchronized (both sync-to and sync-from) using multi-part uploads.

An external storage system can be another HCP for cloud scale system, AWS, or any S3 compatible system.

Bucket sync-to offers the following advantages:

- **Data protection:** Data is well protected against the unavailability or catastrophic failure of a system. A bucket can be synchronized to a remote system of a different type. This arrangement can provide geographically distributed data protection (called *geo-protection*).



Note: All rules must share a single, common destination bucket. If more than one destination appears in the collection of rules the policy is rejected.

- **Data availability:** AWS services can access synchronized data directly from AWS.

Bucket sync-from offers the following advantages:

- **Data consolidation:** Transformed data can be stored on an HCP for cloud scale system. An HCP for cloud scale system can synchronize data from multiple remote systems of different types.
- **External update:** Data can be updated directly in an external system and stored on an HCP for cloud scale system.

Access to bucket synchronization is controlled on a per-user basis by *role-based access control (RBAC)*. Use the System Management application to define users, groups, and roles.

Access to an external resource might need an SSL certificate. You can upload an SSL certificate using the System Management application, the same as for uploading SSL certificates for storage components and IdPs.

For information on configuring bucket synchronizations, see the *S3 Console Guide*.

Object locking

HCP for cloud scale supports object locking, which prevents specified objects from being deleted. A bucket owner can lock or unlock objects or lock them for a specified time period. This feature implements legal hold and retention period requirements.

Object locking is enabled at the bucket level, either when or after a bucket is created. Once enabled, object locking can't be disabled.

Object locking offers the following advantages:

- Locked objects can't be deleted. This implements write once, read many (WORM) behavior, which protects objects from accidental or malicious changes.
- A bucket owner can lock objects until a specified date and time. This implements retention periods, which complies with record retention policy. The retention period can be up to 100 years in the future.



Note: Once set, a retention period can be extended, but not shortened or turned off.

- A bucket owner can lock an object indefinitely, and then turn the lock off. This complies with legal hold requirements. If a legal hold is placed on an object it can't be modified, versioned, moved or deleted, even if it has an expired retention period (that is, a legal hold overrides a retention period). A legal hold never expires, but must instead be removed. An object can have multiple legal holds placed on it.

HCP for cloud scale implements compliance mode as described by the Amazon S3 specification. It does not support governance mode.



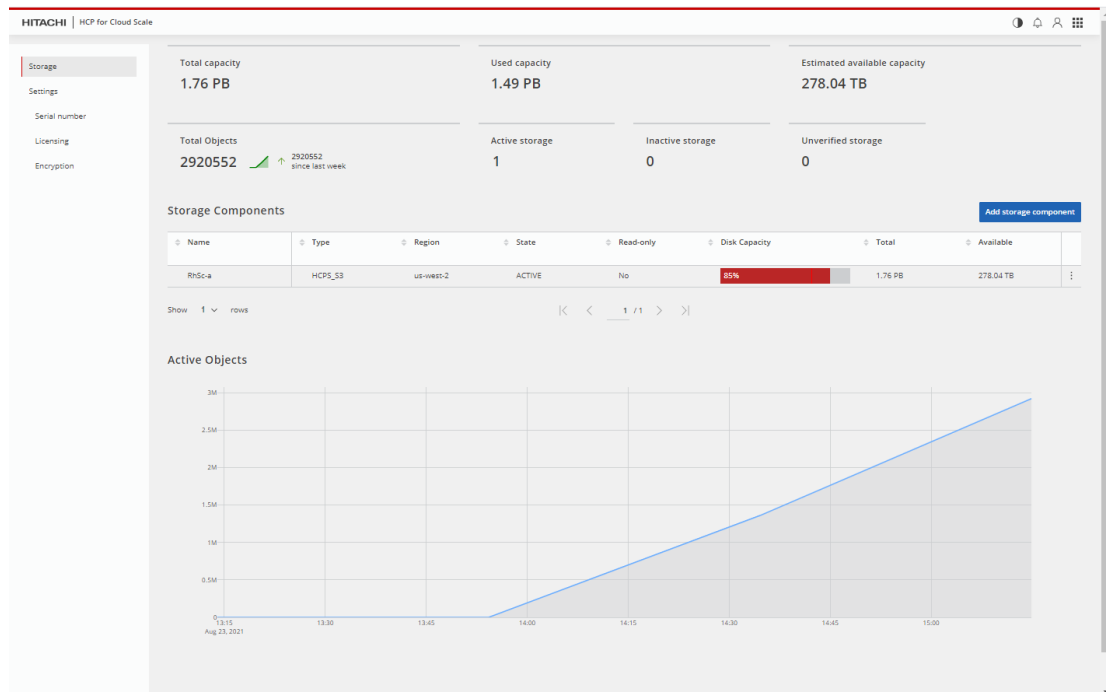
Note: Using S3 PUT Object Lock methods in HCP for cloud scale v1.4 and earlier is not supported. Using the methods might return an HTTP status code of 200 but will not produce the expected behavior. Only use S3 object lock methods after updating to v1.5 or later.

For information on how to lock and unlock objects, see the *S3 Console Guide*.

Capacity monitoring

You can monitor estimated available system-wide or per-storage component capacity.

The **Storage** page, in the Object Storage Management application, displays the total, used, and estimated available (free) capacity for HCP S Series Node storage components configured in the system, as well as the changes in the past week. If all the storage components are HCP S Series nodes, the page displays the total, used, and estimated available capacity of the entire system. You can set capacity threshold alarms that visually indicate, and can also send an alert message or email notification, if the capacity of an HCP S Series Node storage component, or the system as a whole, reaches a threshold. This page provides a single monitoring point for the entire HCP for cloud scale system, and the information displayed helps you with capacity planning.



The Metadata Gateway service periodically gathers storage component capacity metrics. If the used capacity for a single HCP S Series Node storage component or the entire system of HCP S Series Node storage components rises above a specified level, the system displays an alert. You can configure the alert threshold.

The calculation of used capacity includes:

- HCP S Series Node storage components configured for capacity monitoring
- Storage components set to read-only status
- Storage components that are inactive

The calculation of available system capacity does not include:

- HCP S Series Node storage components not configured for capacity monitoring
- Storage components other than HCP S Series Node storage components
- Storage components set to read-only status
- Storage components that are inactive



Note: Metrics for capacity usage are for Metadata Gateway instances only, so adding used capacity to estimated available capacity will not equal the total capacity on the system. Also, multiple services are running on a system instance, and all sharing the disk capacity. Therefore, the available capacity for the Metadata Gateway service on one node can be consumed by a different service running on the same node.

Using capacity information, you can be alerted and take action if a storage component is reaching capacity. You can determine if the system can support an increase in stored data (for example, as expected from a new customer). You can understand the balance of usage and capacity across storage components. You can plan for the orderly addition of additional capacity.

Chargeback reports

Chargeback reports detail how system storage capacity is used, per user or bucket.

HCP for cloud scale provides storage usage reports for objects on the system. Authorized users can generate a report for one or more of the buckets they own. Authorized administrators can generate a report for a user or a list of one or more buckets. Reports can detail hourly, daily, or monthly usage.

Chargeback reports let you create invoices or bills for bucket owners, or delegate that task to others.

How usage is calculated

Metrics for bucket size and number of objects are stored persistently. Storage usage is calculated at the granularity of byte-hours and can be reported by hour, day, or month.

For example, if a user stores 100 GB (107374182400 bytes) of standard storage data in a bucket for the first 15 days in March, and 100 TB (109951162777600 bytes) of standard storage data for the final 16 days in March, the usage is 42259901212262,400 byte-hours. The calculation is as follows:

First calculate the total byte-hour usage:

```
[107374182400 bytes × 15 days × (24 hours/day)] +
[109951162777600 bytes × 16 days × (24 hours/day)] =
42259901212262400 byte-hours
```

Then convert byte-hours to GB-months:

```
42259901212262400 byte-hours ÷
    (1073741824 bytes/GB) ÷
        (24 hours/day) ÷
            (31 days in March) =
                52900 GB-months
```

Usage reports

Storage capacity usage is reported in either a user report or a system report.

- The user report gives storage usage for any or all buckets defined in the system that the user owns.
- The system report gives storage usage for any or all buckets defined in the system.

Within each report you can specify which fields appear.

S3 Select

HCP for cloud scale supports the S3 Select feature.

HCP for cloud scale supports the S3 Select Object Content method, which allows retrieval of a portion of a structured object by an S3 client such as Apache Spark, Apache Hive, and Presto. The portion of the object returned is selected based on a structured query language (SQL) query sent in the request. The query is performed by S3 storage components that support pushdown. Selecting only the data needed within an object can significantly improve costs, time, and performance.

A request can select serialized object data in these formats:

- Apache Parquet

A request can return data in these formats:

- Comma-separated values (CSV)

The client application must have the permission `s3:GetObject`. S3 Select supports reading encrypted data. The SQL expression can be up to 256 KB, and can return up to 1 MB of data.

Here is a simple example of a SQL query against a Parquet object. The query returns data for salaries greater than 100000:

```
select salary from s3object s where s.salary > 100000
```

S3 event notification

HCP for cloud scale supports the S3 `PUT Bucket notification configuration` and `GET Bucket notification configuration` methods.

HCP for cloud scale can send notifications of specified events in a bucket to a message server for applications to consume. This is a more efficient way to signal changes than periodically scanning objects in a bucket.

HCP for cloud scale supports event notification to signal specified events in buckets. Notifications can be sent to AWS SQS Standard services, Kafka, or RabbitMQ. A retry mechanism assures highly reliable notifications.

Supported limits

HCP for cloud scale limits the number of instances (nodes) in a system to 160.

HCP for cloud scale does not limit the number of the following entities.

Entity	Minimum	Maximum	Notes
Buckets	None	Unlimited	A user can own up to 1000 buckets.
Users (external)	None	Unlimited	The local user has access to all functions including MAPI and S3 API methods. However, it's best to configure HCP for cloud scale with an identity provider (IdP) with users to enforce role-based access control.
Groups (external)		Unlimited	

Entity	Minimum	Maximum	Notes
Roles		Unlimited	
Objects	None	Unlimited	The size limit for an object is 5 TB.
Storage components	1	Unlimited	

High availability

HCP for cloud scale supports high availability for multi-instance sites.

High availability needs at least four service instances: three master instances, which run essential services, and at least one worker instance. The best practice is to run the three master instances on separate physical hardware (or, if running on virtual machines, on at least three separate physical hosts) and to run HCP for cloud scale services on more than one instance.

Scalability of instances, service instances, and storage components

You can increase or decrease the capacity, performance, and availability of HCP for cloud scale by adding or removing the following:

- Instances: physical computer nodes or virtual machines
- Service instances: copies of services running on additional instances
- Storage components: S3 compatible systems used to store object data

In a multi-instance site, you might add instances to improve system performance or if you are running out of storage space on one or more instances. You might remove instances if you are retiring hardware, if an instance is down and cannot be recovered, or if you decide to run fewer instances.

When you add an instance, you can also scale floating services (such as the S3 Gateway) to the new instance. When you scale a floating service, HCP for cloud scale automatically rebalances itself.

In a multi-instance site, you can manually change where a service instance runs:

- You can configure it to run on additional instances. For example, you can increase the number of S3 Gateway service instances to improve throughput of S3 API transactions.
- You can configure it run on fewer instances. For example, you can free computational resources on an instance to run other services.
- You can configure it to run on different instances. For example, you can move the service instances off a hardware instance to retire the hardware.
- For a floating service, instead of specifying a specific instance on which it runs, you can specify a pool of eligible instances, any of which can run the service.

Some services have a fixed number of instances and therefore cannot be scaled:

- Metadata Coordination

You might add storage components to a site under these circumstances:

- The existing storage components are running out of available capacity
- The existing storage components do not provide the performance you need
- The existing storage components do not provide the functionality you need

Site availability

An HCP for cloud scale site has three master instances and thus can tolerate the failure of one master instance without interruption of service.

If a site with only two healthy master instances experiences an outage of another master instance (for example, if services restart or the entire instance or operating system restarts), it goes into a degraded state until all three master instances are restored.

Service availability

HCP for cloud scale services provide high availability as follows:

- The Metadata Gateway service always has at least three service instances. When the system starts, the nodes "elect a leader" using the raft consensus algorithm. The other service instances follow the leader. The leader processes all GET and PUT requests. If the followers cannot identify the leader, they elect a new leader. The Metadata Gateway service tolerates the failure of one service instance without interruption. If more than one service instance is unavailable, some data can become unavailable until the instance recovers.
- The Metadata Coordination service always has one service instance. If that instance fails, HCP for cloud scale automatically starts another instance. Until startup is complete, the Metadata Gateway service cannot scale.
- The Metadata Cache service always has one service instance. If that instance fails, HCP for cloud scale automatically starts another instance. Until startup is complete, overall performance decreases.
- To protect messaging consistency, the Message Queue service always has three service instances. To prevent being split into disconnected parts, the service shuts down if half of the service instances fail. In practice, messaging stops if two of the three instances fail. Do not let the service run with only two instances, because in that scenario if one of the remaining instances fails, the service shuts down. However, when one of the failed instances restarts, messaging services recover and resume.
- To maintain access to the encryption key vault, the Key Management Server service uses an active-standby model. One service instance is the active instance and any other service instances are kept as standbys. If the active vault node becomes sealed or unavailable, one of the standbys takes over as active. You can scale up to the number of instances in the HCP for cloud scale system or your acceptable performance limits.

The rest of the HCP for cloud scale services remain available if HCP for cloud scale instances or service instances fail, as long as at least one service instance remains healthy. Even if a service that has only one service instance fails, HCP for cloud scale automatically starts a new service instance.

Metadata availability

Metadata is available as long as these services are available:

- S3 Gateway
- Metadata Gateway

Object data availability

Object data is available as long as these items are available:

- The S3 Gateway service (at least one instance)
- The storage component containing the requested object data
- At least two functioning Metadata Gateway service instances (of the required three)

For high availability of object data or data protection, you should use a storage component with high availability, such as HCP, HCP S Series Node, or AWS S3.

Network availability

You can install each HCP for cloud scale instance with both an internal and an external network interface. To avoid single points of networking failure, you can:

- Configure two external network interfaces in each HCP for cloud scale instance
- Use two switches and connect each network interface to one of them
- Bind the two network interfaces into one virtual network interface in an active-passive configuration
- Install HCP for cloud scale using the virtual network interface

Failure recovery

HCP for cloud scale actively monitors the health and performance of the system and its resources, gives real-time visual health representations, issues alert messages when needed, and automatically takes action to recover from the failure of:

- Instances (nodes)
- Product services (software processes)
- System services (software processes)
- Storage components

Instance failure recovery

If an instance (a compute node) fails, HCP for cloud scale automatically adds new service instances to other available instances (compute nodes) to maintain the minimum number of service instances. Data on the failed instance is not lost and remains consistent. However, while the instance is down, data redundancy might degrade.

HCP for cloud scale adds new service instances automatically only for floating services. Depending on the remaining number of instances and service instances running, you might need to add new service instances or deploy a new instance.

Service failure recovery

HCP for cloud scale monitors service instances and automatically restarts them if they are not healthy.

For floating services, you can configure a pool of eligible HCP for cloud scale instances and the number of service instances that should be running at any time. You can also set the minimum and maximum number of instances running each service. If a service instance failure causes the number of service instances to go below the minimum, HCP for cloud scale starts another service instance on one of the HCP for cloud scale instances in the pool that doesn't already have that service instance running.

Persistent services run on the specific instances that you specify. If a persistent service fails, HCP for cloud scale restarts the service instance in the same HCP for cloud scale instance. HCP for cloud scale does not automatically bring up a new service instance on a different HCP for cloud scale instance.

Storage component failure recovery

HCP for cloud scale performs regular, periodic health verifications to detect storage component failures.

If HCP for cloud scale detects a storage component failure, it sets the storage component state to INACCESSIBLE, so that HCP for cloud scale will not try to write new objects to the storage component, and sends an alert. While a storage component is unavailable, the data in it is not accessible.

HCP for cloud scale continues to verify a failed storage component and, when it detects that the storage component is healthy again, automatically sets its state to ACTIVE. HCP for cloud scale sends an alert when this event happens as well. After the storage component is repaired and brought back online, the data it contains is again accessible and HCP for cloud scale can write new objects to it.

Support for the Amazon S3 API

HCP for cloud scale is compatible with the Amazon Simple Storage Service (Amazon S3) REST API, which lets clients store objects in buckets. A bucket is a container of objects that has its own settings, such as ownership and lifecycle. Using HCP for cloud scale, users can perform common reads and writes on objects and buckets and manage ACL settings through the client access data service.

For information about using Amazon S3, see the [Amazon S3 API documentation](#).

For information about obtaining S3 user credentials, see the *S3 Console Guide*.

The following tables list the supported Amazon S3 API features and describe any implementation differences between the Amazon and HCP for cloud scale S3 APIs.

Authentication and addressing

Feature	Implementation differences
Authentication with AWS Signature Version 4	Fully implemented.
Addressing virtual host style (such as <code>http://bucket.server/object</code>)	Fully implemented.
Addressing path style (such as <code>http://server/bucket/object</code>)	Fully implemented.
Signed/unsigned payload	Fully implemented.
Chunked request	Fully implemented.
Presigned URL	Fully implemented.

Service

Feature	Implementation differences
LIST buckets (GET Service)	Fully implemented.

Buckets

Feature	Implementation differences
GET Bucket (list objects) V1	Fully implemented.
GET Bucket (list objects) V2	Fully implemented.
PUT Bucket	To support legacy S3 buckets, HCP for cloud scale supports bucket names of less than three characters. When anonymous requests to create or remove a bucket use a bucket name that isn't valid, Amazon S3 verifies access first and returns 403. HCP for cloud scale returns 400 if the bucket name validation fails.
DELETE Bucket	
HEAD Bucket	
PUT Bucket ACL	In Amazon S3 each grantee is specified as a type-value pair, where the type is one of the following:

Feature	Implementation differences
GET Bucket ACL	<ul style="list-style-type: none"> ▪ <code>emailAddress</code> if the value specified is the email address of an AWS account ▪ <code>id</code> if the value specified is the canonical user ID of an AWS account ▪ <code>uri</code> if granting permission to a predefined group <p>HCP for cloud scale does not support <code>emailAddress</code>. HCP for cloud scale fully supports <code>id</code>. HCP for cloud scale supports <code>uri</code> for the predefined groups <code>Authenticated Users</code> and <code>All Users</code>.</p> <p>HCP for cloud scale does not support the Amazon S3 predefined grant ("canned ACL") <code>aws-exec-read</code>.</p> <p>HCP for cloud scale supports the canned ACL <code>authenticated-read-write</code>.</p> <p>HCP for cloud scale does not mirror or mirror back ACLs or policies.</p>
List Multipart Uploads	Fully implemented.
GET Bucket Lifecycle (except transition action)	HCP for cloud scale supports the latest API for bucket lifecycle management. Old and deprecated V1.0 methods are not supported.
PUT Bucket Lifecycle (except transition action)	HCP for cloud scale does not support Object Transition actions. Including these actions causes a Malformed XML exception.
DELETE Bucket Lifecycle (except transition action)	
PUT Bucket Notification Configuration	<p>A configuration can have to up 100 rules.</p> <p>Amazon S3 considers that two rules overlap if both apply to the same object and share at least one event type. HCP for cloud scale supports notification from the same object to multiple targets. However, rules are blocked if they send a message for the same event to the same target.</p> <p>All notification message fields are returned except <code>Region</code> and <code>Glacier Storage</code>. The field <code>awsRegion</code> is returned but left empty.</p>
PUT Bucket Replication	<p>HCP for cloud scale supports one-to many mirroring and many-to-one mirroring back. The bucket Amazon Resource Name (ARN) is replaced by configuration settings.</p> <p>For mirroring back, HCP for cloud scale supports one queue server, <code>AMAZON_SQS</code>.</p> <p>Sending encrypted data to a remote bucket is not supported.</p>
GET Bucket Versioning	Returns the bucket versioning configuration and status (always <code>on</code>).

Feature	Implementation differences
GET Bucket Object Versions	Version listing requests do not strictly comply with documented behavior for NextKeyMarker/NextVersionIdMarker. Amazon S3 documentation currently states that these values specify "the first key not returned that satisfies the search criteria." However, HCP for cloud scale specifies the last key returned in the current response. S3 V1 object listings do not call out as specific a requirement and V2 object listings use a continuation token that is opaque to the caller. Internally, HCP for cloud scale shares the same listing logic across all listing types.
GET Bucket Location	You must be the bucket owner.
GET Bucket Notification Configuration	Fully implemented.

Object



Note: The characters null (NUL) and backslash (\) are not supported in object keys for S3 operations. If you include those characters in an operation it fails with the error 400 `BadRequest`. Don't use null or backslash characters in object keys.

Feature	Implementation differences
GET Object	<p>If a lifecycle policy is configured for a bucket, HCP for cloud scale displays the expiration date of an object (in the <code>x-amz-expiration</code> header) fetched using the subresource <code>?versionId</code>.</p> <p>Legal hold is fully implemented.</p> <p>Object retention is fully implemented.</p> <p>Object names cannot contain NUL or backslash (\) characters. GET methods on objects so named fail with a 400 error.</p>
HEAD Object	<p>If a lifecycle policy is configured for a bucket, HCP for cloud scale displays the expiration date of an object (in the <code>x-amz-expiration</code> header) fetched using the subresource <code>?versionId</code>.</p>
PUT Object	<p>In HCP for cloud scale, the maximum file size for a single PUT object call is configurable through the management API. The maximum and default is 5 GB.</p> <p>Amazon S3 is liberal in what is accepted for the Content-Type of an object. HCP for cloud scale adds additional content-type validation.</p> <p>Bucket synchronization is supported.</p>

Feature	Implementation differences
	<p>Legal hold is fully implemented. AWS object lock permissions are not supported; that is, a bucket owner can set a legal hold without restriction.</p> <p>Object retention is implemented, but not governance mode; that is, once a retain-until date is set, it can be extended but not removed. AWS object lock permissions are not supported; that is, a bucket owner can set object retention without restriction.</p> <p>Object locking can be applied to a bucket even after it's created. To enable object locking, in the S3 API <code>PUT Bucket ObjectLockConfiguration</code>, include the URI request parameter <code>x-amz-bucket-object-lock-token</code> (with any string).</p> <p>Object names cannot contain NUL or backslash (\) characters. PUT methods on objects so named fail with a 400 error.</p>
PUT Object (Copy)	Conditional headers are not supported. Server-side encryption is not supported.
PUT Object (Part Copy)	Conditional headers are not supported. Server-side encryption is not supported.
Object and version encoding	<p>Amazon S3 object and version listing documentation mentions the ability to pass an encoding parameter so the object name in the response XML to the client can be escaped to avoid names containing XML characters that are not valid. Encoding is documented only as it applies to object names, not Owner/DisplayNames. Also, escaping for Bucket Listing requests is not mentioned.</p> <p>The Owner/DisplayName is a concern because user display names might contain characters that can cause XML parsing issues. Amazon S3 does not currently return a display name for all regions. HCP for cloud scale uses IdPs and thus does not control restriction.</p> <p>Bucket name restrictions should prevent problematic bucket names from being created. For security, HCP for cloud scale passes the user display name through a URI encoder before returning the name in an XML response.</p>
Object tagging	<p>Amazon S3 wraps eTags in double quotes. For XML listings (v1 object, v2 object, version) double quotes are escaped. For example:</p> <pre><ETag>&quot;32c81604d07395b1aa39a7e206c3af06\$&quot;; </ETag></pre> <p>HCP for cloud scale does not do this because only attributes, not double quotes, need to be escaped within content.</p>

Feature	Implementation differences
	<p>Expiration date URL encoding (x-amz-expiration header)</p> <p>HCP for cloud scale URL-encodes the <code>RuleID</code> portion of the header <code>x-amz-expiration</code> using the same encoding strategy that Amazon suggests for V4 authentication. This strategy can result in encoded strings that do not exactly match how Amazon encodes RuleIDs. However, decoding them should always return the original strings.</p> <p>HCP for cloud scale synchronizes (mirrors and mirrors back) existing object tags. Subsequent PUT Object tagging requests are not synchronized, so tags added to, removed from, or updated on a synchronized object are not synchronized. To ensure that tags are processed as expected, it's best to set them when creating objects rather than creating objects and then setting their tags.</p>
GET Object ACL	Bucket synchronization is not supported.
PUT Object ACL	<p>In Amazon S3, each grantee is specified as a type-value pair, where the type is one of the following:</p> <ul style="list-style-type: none"> ▪ <code>emailAddress</code> if the value specified is the email address of an AWS account ▪ <code>id</code> if the value specified is the canonical user ID of an AWS account ▪ <code>uri</code> if granting permission to a predefined group <p>HCP for cloud scale does not support <code>emailAddress</code>. HCP for cloud scale fully supports <code>id</code>. HCP for cloud scale supports <code>uri</code> for the predefined groups <code>Authenticated Users</code> and <code>All Users</code>.</p> <p>HCP for cloud scale does not support the <code>aws-exec-read</code> canned ACL.</p>
DELETE Object	<p>Bucket synchronization or removal of an object or a specific version of an object is not supported.</p> <p>To improve performance, if the current version of an object is a delete marker, HCP for cloud scale does not create another delete marker.</p>
DELETE Multiple Objects	<p>Fully implemented.</p> <p>Bucket synchronization is not supported.</p>
POST Object	<p>Fully implemented.</p> <p>In HCP for cloud scale, the maximum file size for a single POST object call is configurable through the management API. The maximum and default is 5 GB.</p> <p>Bucket synchronization is supported.</p>

Feature	Implementation differences
POST Select Object Content	<p>Scan range is supported.</p> <p>HCP for cloud scale supports the use of * by itself with no alias reference. For example, this SQL query is supported:</p> <pre>select *, first_name from s3object s where s.salary > 100000 limit 10</pre> <p>HCP for cloud scale supports a wider range of date-time formats than AWS. The full list is available at https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html.</p> <p>HCP for cloud scale supports nested aggregate functions. For example, this expression is supported: <code>count (sum (s.salary))</code></p> <p>HCP for cloud scale SQL queries on columns are case sensitive, while AWS SQL queries are case insensitive. For example, given an object <code>s</code> with the columns <code>ID</code>, <code>id</code>, and <code>id</code>, an SQL query to select <code>s.id</code> will return column <code>id</code> in HCP for cloud scale but column <code>ID</code> in AWS.</p> <p>Only input serialization of Parquet is supported. Requests for CSV or JSON objects are not supported and return an error.</p> <p>Parquet compression is managed automatically, so the <code>CompressionType</code> argument is not needed, and if specified returns an error.</p> <p>Only CSV output is supported. Specifying another output format returns an error.</p> <p>AWS calculates the size of a record returned in an S3 Select query as the total size of the record, including any delimiters. HCP for cloud scale calculates the size as the total data of each column returned. These calculations can sometimes differ slightly.</p>
Initiate/Complete/Abort Multipart Upload	Fully implemented.
Upload Part	Fully implemented.
List Multipart Uploads	Fully implemented.

Unsupported S3 API methods

HCP for cloud scale does not support the following Amazon S3 API methods:

Authentication

- Authentication v2 (deprecated by AWS)

Bucket

- GET/PUT/DELETE Bucket website
- GET/PUT/DELETE Bucket policy
- GET/PUT/DELETE Bucket tagging
- GET/PUT/DELETE Bucket CORS (cross-origin resource sharing)
- PUT Bucket versioning (with HCP for cloud scale versioning is always on)
- GET/PUT Bucket logging
- GET Bucket notification
- GET/PUT Bucket requestPayment
- GET/PUT/DELETE Bucket Inventory
- List Bucket Inventory Configurations
- GET/DELETE Bucket metrics
- List Bucket Metrics Configurations
- GET/PUT/DELETE Bucket analytics
- List Bucket Analytics Configurations
- PUT/GET Bucket accelerate
- Server-side encryption with customer-provided encryption keys (SSE-C)
- Server-side encryption with storage-managed encryption keys (SSE-S3)

Object

- Options object
- GET/POST Object torrent

HCP for cloud scale APIs

The Hitachi Content Platform for cloud scale (HCP for cloud scale) system includes RESTful application programming interfaces (APIs) that you can use for writing applications that exercise its functions and manage the system.

Anything you can do in the Object Storage Management, S3 Console, or System Management application GUIs you can also do using APIs.

Object Storage Management API

The Object Storage Management application includes a RESTful API to administrative functions such as managing storage components, configuring Amazon S3 settings, and obtaining or revoking S3 user credentials. For more information on the Object Storage Management API, see the *MAPI Reference*.

System Management API

The System Management application includes a RESTful API to system management functions such as system monitoring, service monitoring, user registration, and configuration. For more information on the System Management API, see the Swagger interface in the System Management application.

Amazon S3 API

Unless otherwise noted, HCP for cloud scale is fully compatible with the Amazon S3 API.

Object Storage Management API

The Object Storage Management application includes a RESTful API interface for the following functions:

- Managing storage components and Amazon S3 settings
- Managing administrative resources such as serial numbers and system events
- Managing user resources such as S3 user credentials and OAuth tokens

The Object Storage Management API is served by the MAPI Gateway service from any HCP for cloud scale node.

You can execute all functions supported in the Object Storage Management application using the API.



Note: The system configuration, management, and monitoring functions included in the System Management application can be performed using the System Management API.

All URLs for the API have the following base, or root, uniform resource identifier (URI):

```
https://hcpcs_ip_address:9099/mapi/v1
```

System Management API

The System Management application provides a RESTful API for managing the following:

- Alerts
- Business objects
- Certificates
- Events
- Instances
- Jobs
- Licenses
- Notifications
- Packages
- Plugins
- Security

- Services
- Setup
- Tasks
- Updates

You can execute all functions supported in the System Management application using the API.

Security and authentication

HCP for cloud scale controls access to system functions through user accounts, roles, permissions, and, where user accounts are stored in an external identity provider, by OAuth tokens. All browser pages that make up the system are protected and cannot be reached without authentication. Users who try to reach a system page without authentication are redirected to the login page.

HCP for cloud scale controls access to data by S3 API requests through S3 credentials, ownership, and access control lists. HCP for cloud scale supports in-flight encryption (HTTPS) for all external communications.

User accounts

The initial user account, which has all permissions, is created when you install HCP for cloud scale. The initial user account can perform all HCP for cloud scale functions. After the initial user account is created, you can change its password any time, but you cannot disable the account and you cannot change its permissions.

The initial user is the only local account allowed and is intended only to let you configure an identity provider (IdP). HCP for cloud scale can communicate with IdPs using HTTP or HTTPS. HCP for cloud scale supports multiple IdPs:

- Active Directory
- OpenLDAP
- 389 Directory Server
- LDAP compatible

HCP for cloud scale supports external users defined in the IdP. External users with the appropriate permissions can perform some or all of these functions:

- Log in to the Object Storage Management application and use all functions
- Log in to the System Management application and use all functions
- Get an OAuth token to use all API calls for the Object Storage Management and System Management applications
- Log in to the S3 Console application and get S3 credentials to use the S3 API

HCP for cloud scale discovers the groups in each IdP and allows assigning roles to groups.

HCP for cloud scale uses OAuth2 as a service provider to authenticate single sign-on (SSO) access. SSO lets you use one set of login credentials for all HCP for cloud scale applications, so you can switch between applications without logging in again.

API access

Object Storage Management application API methods need a valid OAuth access token for a user account with suitable permissions, or else the requests are rejected. With one exception, System Management application API methods also require a valid OAuth access token for a user account with suitable permissions, or else the requests are rejected. (The exception is the API method to generate an OAuth token, which requires only a username and password in the body of the request.)

Before using either the Object Storage Management or System Management APIs, you need to obtain an OAuth token. You can generate an OAuth token by sending a request to the OAuth server with your account credentials. Then you can supply the OAuth token in the Authorization header in each request. OAuth tokens are valid for five hours.



Note: An administrator can revoke all OAuth tokens for any other HCP for cloud scale user. You would do this, for example, if an employee leaves the company, you delete the user account, and you do not want to wait for the account tokens to expire.

S3 API requests generally require valid S3 credentials for users with the right privileges, that is, access control lists (ACLs). (Exceptions are methods configured to allow anonymous access and pre-signed requests.) HCP for cloud scale supports AWS Signature version 4 authentication to include S3 credentials in S3 requests.

Users with a valid account and suitable permissions can generate S3 credentials. You can generate an unlimited number of S3 credentials, but only the last credentials generated are valid. These credentials are associated only with your account. S3 credentials do not have an expiration date, so they are valid until revoked.

Users with a valid account and suitable permissions can revoke all S3 credentials of any user. That is, you can revoke your own S3 credentials or the S3 credentials of any other user. Revocation removes all S3 credentials associated with the account.



Note: Deleting a user account from the IdP does not revoke S3 credentials, and if a user's S3 credentials are revoked the user can still generate new credentials. The best practice is to delete the user account from the IdP and then revoke the S3 credentials.

Network isolation and port mapping

When you install HCP for cloud scale, you can set up network isolation by configuring one external network and one internal network.

HCP for cloud scale software creates a cluster using commodity x86 servers that are networked using Ethernet. The software uses two networks on the operating system hosting the HCP for cloud scale software. These networks can also use link aggregation defined by the OS administrator.

While two networks provide optimal traffic isolation, you can deploy the software using a single network. The OS administrator must make and implement networking decisions before you install HCP for cloud scale.

HCP for cloud scale services use a range of network ports. You can configure services to use different ports instead of the default ports. Installation is the only opportunity to change the default ports used by services.



Note: The following services must be deployed with their default port values:

- Message Queue
- Tracing Agent
- Tracing Collector
- Tracing Query

For information about installing HCP for cloud scale, see *Installing Hitachi Content Platform for Cloud Scale*.

Logging in

HCP for cloud scale provides one locally defined administrative user account. Any other user accounts reside in a *realm* provided by external identity providers (IdPs). To log in you need this information:

- The cluster hostname, instance, or IP address of the HCP for cloud scale system that you're using
- Your user name as assigned by your system administrator
- Your password as assigned by your system administrator
- The realm where your user account is defined

Procedure

1. Open a web browser and go to `https://system_address:8000`
system_address is the address of the HCP for cloud scale system that you're using
2. Type your username and password.
3. In the **Security Realm** field, select the location where your user account is defined. To log in using the local administrator account, without using an external IdP, select **Local**. If no IdP is configured yet, **Local** is the only available option.
4. Click **LOGIN**.

Result

The Applications page opens.

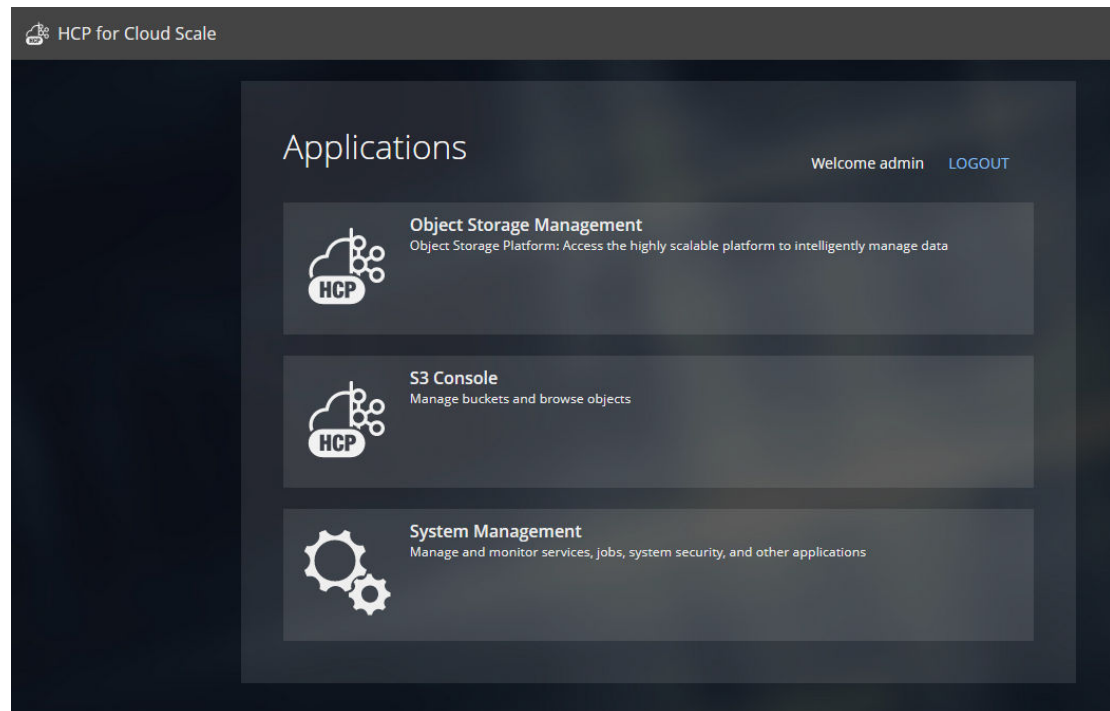


Note: When a new user is created and added to a group, that user might not have immediate access to HCP for cloud scale. Instead, login fails with the message "Not authorized. Please contact your system administrator." Verify the credentials. If the condition persists, the system administrator can use the API method `security/clearCache` to allow immediate login.

HCP for cloud scale applications

After you log in, the HCP for cloud scale **Applications** page shows you the applications you are authorized to use, such as:

- **Object Storage Management:** Manage and monitor storage components, data objects, alerts, and regions
- **S3 Console:** Generate S3 access and secret keys; conveniently create and manage buckets, bucket synchronization, and bucket policies; manage S3 event notification; and browse objects in buckets
- **System Management (sometimes referred to in the application as the Admin App):** Manage and monitor cluster instances, software services, system security, user accounts, and other cluster configuration parameters





From the **Applications** page, or from within each application, you can switch back and forth between applications as needed.

Switching between applications

HCP for cloud scale uses OAuth2 as a service provider to authenticate single sign-on (SSO) access. You only need one set of login credentials for all HCP for cloud scale applications, so you can switch between applications without logging in again.

Depending on the permissions assigned to your account role, you can have access to one or more HCP for cloud scale application. To switch between applications:

Procedure

1. Depending on the application you are currently using:
 - In the Object Storage Management application, click the **app switcher** menu () and select another application.
 - In the System Management application, click the **Open** menu () in the right corner of the top navigation bar, and select another application.



Note: The System Management application is also identified in the user interface as **Admin App**.

2. Select the application you want to use.
The application opens.

Providing unseal keys to KMS service

When encryption is enabled for a HCP for cloud scale system, the Key Management System service provides encryption keys for storage components. If the service restarts, the key repository vault closes and data objects can't be decrypted. If a vault instance becomes sealed, you must provide a quorum of unseal keys (three of the five provided when encryption was first enabled) to reopen the vault and resume encryption and decryption.



Caution: Don't try to initialize the vault manually outside of HCP for cloud scale. Doing so results in data loss.

Procedure

1. From the Object Storage Management application, select **Settings > Encryption**.
The **ENCRYPTION** page opens.
2. In the **UNSEAL VAULT INSTANCES** section, enter the first unseal key into the **Unseal key 1** field.
The key is validated. You can't leave the field blank.
3. Enter a second unseal key into the **Unseal key 2** field.
The key is validated. You can't leave the field blank. Each key must be different.
4. Enter a third unseal key into the **Unseal key 3** field.
The key is validated. You can't leave the field blank. Each key must be different.
5. Click **Unseal vault**.

Result

The vault is unsealed.

Serial number

You can use the Object Storage Management application or an API method to enter and display your HCP for cloud scale serial number.

A serial number is required to activate the HCP for cloud scale software. You must enter the serial number before you can use the system or its licensed features.

Entering your serial number

The Object Storage Management application displays the product serial number. An administrative account with appropriate permissions can enter or edit this number.

Object Storage Management application instructions

Procedure

1. From the Object Storage Management application, select **Settings** > **Serial number**. The **SERIAL NUMBER** page opens.
2. Enter your serial number into the **Serial number** field.
3. Click **Save**.

Related REST API methods

```
POST /serial_number/set
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Displaying your serial number

You can use the Object Storage Management application or an API method to displays the product serial number.

Object Storage Management application instructions

The product serial number is displayed in the Object Storage Management application on the **SERIAL NUMBER** page.

Related REST API methods

```
POST /serial_number/get
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

License

You can use the Object Storage Management application or an API method to enter and validate your HCP for cloud scale license.

A license is required before you can activate certain HCP for cloud scale features. You must enter your serial number before you can upload your license.

Uploading your license

The Object Storage Management application displays your product licenses. An administrative account with appropriate permissions can upload a license file.

Object Storage Management application instructions

Procedure

1. From the Object Storage Management application, select **Settings > Licensing**. The **Licensing** page opens.
2. Click **Upload license**. The **UPLOAD LICENSE** page opens, displaying the **Select file** area
3. Do one of the following:
 - Drag and drop a license file into the **Select file** area.
 - Click **Select file**, select a license file, and then click **Open**.

The license file is decrypted and validated and appears on the **Licensing** page.

Related REST API methods

```
POST /license/list
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Enabling encryption

Encryption is an available licensed feature. You must obtain and upload a license to enable encryption.

An administrative account with appropriate permissions can enable encryption. All objects added to the system after encryption is enabled are encrypted on all storage components.



Note: Encryption is a global setting. Once enabled, you can't turn off encryption or decrypt either storage components or the objects stored on them.

Before you enable encryption, set up a Vault server. Enabling encryption generates encryption keys, an initial root token, and a set of five unseal keys. When establishing a connection to the Vault server, HCP for Cloud Scale provides the initial root token for authentication and root access.

Vault doesn't store the generated master key. Instead, each time the Vault server starts it uses the unseal keys to regenerate the master key, which is then used to return storage component encryption keys. If the Vault server goes down, it seals the vault, and to regenerate the master key you must provide a quorum of at least three valid unseal keys.



Caution: If you don't provide a quorum of unseal keys to reconstruct the master key, Vault remains sealed, so the master key is unavailable and encrypted storage components can't be decrypted. To ensure encryption security, the best practice is to encrypt and store unseal keys separately.

You can enable encryption using the Object Storage Management application or a management API method.



Caution: If two accounts try to set the encryption flag simultaneously, either using the Object Storage Management application or the management API method `/s3_encryption/set`, existing storage components can become inaccessible.

- If you intend to use encryption, set it before defining storage components.
- If you have already defined storage components and intend to use encryption, do not try to set encryption from multiple accounts, or by multiple calls to the API method `/s3_encryption/set`, simultaneously.

After enabling encryption, restart (repair) the S3 Gateway and Policy Engine services.

Object Storage Management application instructions

Procedure

1. From the Object Storage Management application, select **Settings > Encryption**. The **ENCRYPTION** page opens. The page displays information about the key management server connection.
2. In the **ENCRYPTION** section, click **Enable**. You are prompted that turning on encryption is permanent.
3. Click **Enable** to confirm.



Note: You receive an error message if the KMS service has stopped or is unable to complete the request. You might also receive an error message if the key management server is not yet available. In this case, try again when the server is available.

The **Vault unsealing** window opens, displaying your initial root token and five unseal keys.

4. Click **Copy** for the initial root token and save it elsewhere.
5. Click **Copy** for each unseal key and save the keys elsewhere.



Important: This window is the only time that all of this data is ever known by Vault and also the only time that the unseal keys should ever appear together. To minimize the possibility of multiple keys becoming unavailable, the best practice is securely distribute, encrypt, and store the unseal keys in as many different locations.

6. Click **Close**.
You are warned that you won't have another opportunity to record the unseal keys and the initial root token.
7. Click **Continue**.
The Key Management Server service is initialized, a connection to the KMS server is established, the storage component encryption keys are generated and applied, and encryption is enabled.

Next steps

After enabling encryption, restart (repair) the S3 Gateway and Policy Engine services.

Related REST API methods

```
POST /s3_encryption/set
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Defining subdomain for S3 Console application

The S3 Console application uses a subdomain of the HCP for cloud scale system.

The S3 Console application uses a subdomain within the HCP for cloud scale system, such as `s3.hcpcs.Company.com`. For user convenience, you can modify the `hosts` file on systems used to call the S3 Console application.

Procedure

1. On a system that calls the S3 Console application, open the `hosts` file in an editor.
On a Windows system, the `hosts` file is normally located at `C:\Windows\System32\drivers\etc\hosts`. On a Linux system, the `hosts` file is normally located at `/etc/hosts`.
2. Associate the IP address of the HCP for cloud scale system with the S3 subdomain.
`10.24.19.54 s3.hcpcs.Company.com`
3. Save the file.
4. Repeat Steps 1-3 for every system used to call the S3 Console application.

About page

The Object Storage Management application **About** page displays the product version number and a link to the software license terms.

The **About** page is available from the user profile icon.



Note: Version information is also displayed on the main login page.

Chapter 2: Storage components

Within the Hitachi Content Platform for cloud scale (HCP for cloud scale) system, the Object Storage Management application lets you manage and monitor storage components.

The starting point for storage component management is the page **Storage** in the application Object Storage Management. The procedures in this module begin at this page.

Adding a storage component

You can use the Object Storage Management application or an API method to add a storage component to the HCP for cloud scale system.



Tip: To improve performance and availability, and to avoid transfer fees, add storage components that are local to the HCP for cloud scale site.

The storage component must contain an HCP for cloud scale bucket before you can add the storage component to the HCP for cloud scale system.

To add a storage component, it must be available and you need the following information about it:

- Storage component type
- Endpoint information (host name or IP address)
- If an HCP S Series Node storage component, the cluster name, management host name, and administrative user credentials
- If used, the proxy host and port and the proxy user name and password
- API port
- S3 credentials (the access key and secret key to use for access to the storage component bucket)



Tip: You can use the HCP S Series Management Console or management API to generate S3 credentials. Only you can generate the S3 compatible API credentials for your user account.

Object Storage Management application instructions




Note: The storage component must contain an HCP for cloud scale bucket before you can add it.

Procedure

1. From the **Storage** page, click **Add storage component**.
The **ADD STORAGE COMPONENT** page opens.
2. Specify the following:
 - a. **Name** (optional): Type the display name you choose for the storage component, up to 1024 alphanumeric characters.
If you leave this blank, the storage component is listed without a name.
 - b. **Type**: Select **AMAZON_S3**, **HCP_S3**, **HCPS_S3** (HCP S Series Node), or **GENERIC_S3**.
 - c. **Region** (optional): Type a region name of up to 1024 characters.
 - d. **Endpoint**: Type either the IP address or the cluster host name of the storage component. Type as many as 255 URI unreserved characters using only A-Z, a-z, 0-9, hyphen (-), period (.), underscore (_), and tilde (~). The final segment of a host name must not begin with a number.

For an HCP S Series Node storage component, the host name must be `hs3.cluster_name`.
3. In the **S3 CONNECTION** section, specify the following:
 - a. Select the **Protocol** used, either **HTTPS** (the default) or **HTTP**.
 - b. If **Use Default** is selected, the applicable default port number is filled in. If you cancel the selection **Use Default**, type the **Port** number.
4. In the **PROXY** section, specify the following:
 - a. If you select **Use Proxy**, type values in the **Host** and **Port** boxes, and if the proxy needs authentication, type the **Username** and **Password**.
5. In the **BUCKET** section, specify the following:
 - a. **Bucket Name**: Type the name of the bucket on the storage component. The name can be from 3 to 63 characters long and must contain only lowercase characters (a-z), numbers (0-9), periods (.), or hyphens (-).

 **Note:** The bucket must already exist on the storage component and should be empty.
 - b. (Optional) To use path-style URLs to access buckets, select **Use path style always** (the default).
6. In the **AUTHENTICATION** section, specify the following:
 - a. **Type**: Select the AWS Signature version, either **V2** or **V4**.
 - b. Type the **Access Key**.
 - c. Type the **Secret Key**.
7. When you are finished, click **Save**.
The storage component is added to the **Storage components** section of the **Storage** page with the state **ACTIVE**.

Result

You have defined a storage component.

If the storage component state is **INACTIVE**, a configuration value might be incorrect. Select **Verify** from the **More** menu for the storage component and click **Activate** from the window that appears. If configuration errors are detected, correct them and try again.

Related REST API methods

```
POST /storage_component/create
```



Note: After you define the storage component, if its state is **UNVERIFIED**, check the parameters you used when defining it.

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select **REST API**.

Verifying a storage component

You can use the Object Storage Management application to verify a storage component.

The storage component must be in the state **Available** before it can be used by the HCP for cloud scale system.

Object Storage Management application instructions

To verify a storage component, select it from the list on the **Storage** page, and from the **More** menu select **Verify**.

The verification process checks for these possible configuration errors:

- The specified bucket is already in use.
- The specified bucket does not exist.
- The endpoint is incorrect.
- The secret key or access key is incorrect.
- Path style addressing is configured but the storage component cannot use it.
- The authorization type is incorrect.


If configuration errors are detected, edit the storage component configuration to correct them and try again.

Modifying a storage component

You can use the Object Storage Management application or an API method to modify the configuration of a storage component after defining it.

Object Storage Management application instructions

Procedure

1. From the **Storage** page, navigate to the storage component you want to edit.
2. Click the more icon () icon by the storage component and select **Edit**.
The storage component's configuration page appears.
3. Edit the connection information as needed. When you're finished, click **Save**.
The **Username** field is blank, but the configured value is used unless you are change it.

Result

The storage component is modified.

If the storage component state becomes INACTIVE, a configuration value might be incorrect. Select Verify from the More menu for the storage component and click Activate from the window that appears. If configuration errors are detected, correct them and try again.

Related REST API methods

```
POST /storage_component/update
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Activating a storage component


You can use the Object Storage Management application or an API method to activate a storage component.

A storage component is displayed as UNVERIFIED if HCP for cloud scale cannot reach the storage component with the supplied parameters or if the storage component is misconfigured.

Object Storage Management application instructions

Note: You can only activate a storage container that is in the state INACTIVE.

Procedure

1. From the **Storage** page, navigate to the storage component you want to activate.
2. Click the more icon () of the storage component and then select **Set active**.
A message appears and prompts you to confirm your action.
3. Click **Yes, active**
The storage component state changes to **ACTIVE**.

Result

The storage component is activated.

If the storage component state remains **INACTIVE**, a configuration value might be incorrect. Select **Verify** from the **More** menu for the storage component and click **Activate** from the window that appears. If configuration errors are detected, correct them and try again.

Related REST API methods

```
POST /storage_component/update_state
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select **REST API**.

Deactivating a storage component

You can use the Object Storage Management application or an API method to deactivate a storage component.

You might deactivate a storage component for maintenance purposes.

After you mark a storage component as **INACTIVE**, read, write, and healthcheck requests are rejected.

Object Storage Management application instructions



Note: You can only deactivate a storage container that is in the state **ACTIVE**.

Procedure

1. From the **Storage** page, navigate to the storage component you want to deactivate.
2. Click the more icon (⋮) of the storage component and then select **Set inactive**. A message appears and prompts you to confirm your action.
3. Click **Yes, inactivate**. The storage component state changes to **INACTIVE**.

Related REST API methods

```
POST /storage_component/update_state
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select **REST API**.

Marking a storage component as read-only

You can use the Object Storage Management application or API methods to mark a storage component as read-only.


You might mark a storage component as read-only if it is nearly full.

Once you mark a storage component as read-only, write requests are directed to different storage components.

You can only mark a storage component as read-only if it is marked read-write and in the state ACTIVE.

Object Storage Management application instructions

Procedure

1. From the **Storage** page, navigate to the storage component you want to mark.
2. Click the more icon () of the storage component and then select **Set read-only**. A message appears and prompts you to confirm your action.
3. Click **Mark read-only**.
The storage component is marked as read-only.

Related REST API methods

```
PATCH /storage_component/update
POST /storage_component/update_state
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Marking a storage component as read-write


You can use the Object Storage Management application or API methods to mark a storage component as read-write.

This makes the storage component available for writing new objects.

You can only mark a storage component as read-write if it is marked read-only and in the state ACTIVE.

Object Storage Management application instructions

Procedure

1. From the **Storage** page, navigate to the storage component you want to mark.
2. Click the more icon () of the storage component and then select **Open for writes**. A message appears and prompts you to confirm your action.
3. Click **Open for writes**.
The **Read-only** flag for the storage component is marked as **No**.

Related REST API methods

```
PATCH /storage_component/update
POST /storage_component/update_state
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Viewing storage components

You can use the Object Storage Management application or an API method to view information about the storage components defined in the system.

For each storage component, you can get information about its name, type, region, and current state.

The storage component types are:

- AMAZON_S3: An Amazon Web Services S3 compatible node
- HCP_S3: A Hitachi Content Platform node
- HCPS_S3: An HCP S Series node
- GENERIC_S3: An S3 compatible node

The possible storage component states are:

- Active: Available to serve requests
- Inactive: Not available to serve requests (access is administratively paused)
- Inaccessible: Available to serve requests, but HCP for cloud scale is having access issues (for example, network, authentication, or certificate issues)
- Unverified: Not available to serve requests (unreachable by specified parameters, miconfigured, or awaiting administrative activation)

The storage component state Read-only can be on or off.

Object Storage Management application instructions

The storage components defined in the HCP for cloud scale system are listed in the Storage components section of the **Storage** page.

Related REST API methods

```
POST /storage_component/list
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Displaying storage component analytics

The **Storage** page displays counts of active, inactive, and unverified storage components, the total count of active objects, and information about system-wide total, used, and estimated available storage capacity. The page also displays information about individual storage components and their current capacity.

The **Storage** page displays several areas of information.

System-wide information

The top area of the page displays the following rolled-up information for HCP S Series Node storage components configured in the system.

HITACHI HCP for Cloud Scale				
<ul style="list-style-type: none"> Storage Settings Serial number Licensing Encryption 	Total capacity 1.77 PB	Used capacity 241.25 GB	Estimated available capacity 1.77 PB	
	Total Objects 193 193 since last week	Active storage 2	Inactive storage 0	Unverified storage 0

- Total capacity - the total number of bytes available, as well as the change over the past week
- Used capacity - the total number of bytes used, as well as the change over the past week
- Estimated available capacity - the total number of bytes unused, as well as the change over the past week
- Total objects - the total count of objects stored across all storage components, as well as the change over the past week
- Active storage - the number of storage components that can receive objects
- Inactive storage - the number of storage components that cannot receive objects
- Unverified storage - the number of storage components whose state can't be determined

The calculation of used capacity includes:

- HCP S Series Node storage components configured for capacity monitoring
- Storage components set to read-only status
- Storage components that are inactive

Metrics for capacity usage are for Metadata Gateway instances only, so adding used capacity to estimated available capacity will not equal the total capacity on the system. Also, multiple services are running on a system instance, all sharing the disk capacity. Therefore, the estimated available capacity for the Metadata Gateway service on one node can be consumed by a different service running on the same node.



Note: If the MAPI Gateway service restarts, capacity values are shown as 0 until fresh metrics are obtained.

The calculation of estimated available system capacity does not include:

- HCP S Series Node storage components not configured for capacity monitoring
- Storage components other than HCP S Series Node storage components
- Storage components set to read-only status
- Storage components that are inactive

Per-storage component information

The central area of the page displays information for each HCP S Series Node storage component configured for capacity monitoring in the system.

Name	Type	Region	State	Read-only	Disk Capacity	Total	Available
R73 - 1	HCP S_3		ACTIVE	No	0%	1.77 PB	1.77 PB
R73 - 2	HCP S_3		ACTIVE	No	0%	1.77 PB	1.77 PB

- User-defined name.
- Type (HCP S Series Node, displayed as HCP S_3).
- AWS region.
- State:
 - Active - Available to serve requests
 - Inactive - Not available to serve requests (access is administratively paused)
 - Unverified - Not available to serve requests (unreachable by specified parameters, or awaiting administrative activation)
- Whether or not the storage component is set to read-only status.
- Disk capacity: A graphical display of used capacity as a percentage of total capacity. You can configure a warning threshold, which is displayed as a red line. If the used capacity is below the threshold the bar is displayed in blue, and if the used capacity exceeds the threshold the bar is displayed in red. If no capacity is used the bar is displayed in gray. For example:

Disk Capacity	Total	Available
87%	1.76 PB	229.27 TB
87%	1.76 PB	229.27 TB

- Total capacity (used plus free).
- Available (estimated) capacity.

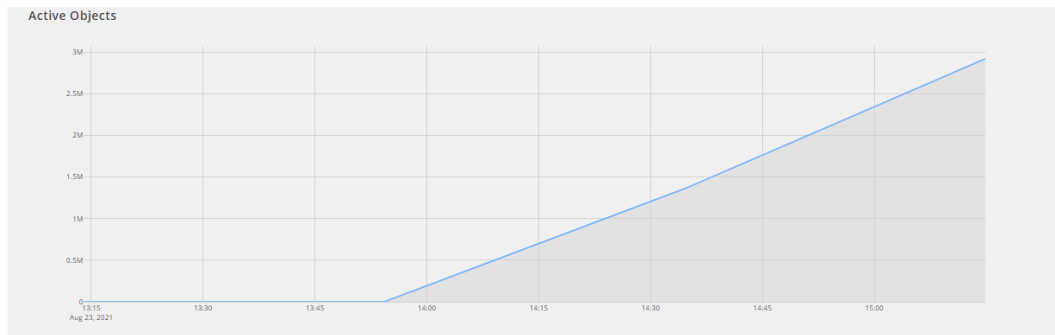
Capacity alerts are generated by the MAPI Gateway service. Use the System Management application to configure the capacity alert threshold for individual storage components or the overall system.

A more button (⋮), to the right of each storage component, opens a menu of actions that you can perform on that storage component:

- Edit - edit the configuration of the storage component
- Set inactive | Set active- change the state of the storage component between active and inactive
- Set read-only | Set read-write - change the status of the storage component between read-only and read-write

Active object information

The bottom area of the page displays a graph over time of the count of active objects stored in the system. The maximum time period is the previous week.



Displaying counts of storage components

You can use the Object Storage Management application or an API method to display counts of storage components in the system.

The page displays the following rolled-up information for HCP S Series Node storage components configured in the system:

- Active storage - the number of storage components that can receive objects
- Inactive storage - the number of storage components that cannot receive objects
- Unverified storage - the number of storage components that are misconfigured or whose state can't be determined

Object Storage Management application instructions

To display storage counts, select Storage.

The infographic displays the count of active, inactive, and unverified storage components.

Related REST API methods

```
POST /storage_component/list
```


For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Metrics

HCP for cloud scale uses a third-party, open-source software tool, running over HTTPS as a service, to provide storage component metrics through a browser.

The Metrics service collects metrics for these HCP for cloud scale services:

- S3 Gateway
- MAPI Gateway
- Policy Engine
- Metadata Coordination
- Metadata Gateway

By default the Metrics service collects all storage component metrics and you cannot disable collection. By default, the Metrics service collects data every ten seconds (the Scrape Interval) and retains data for 15 days (the Database Retention); you can configure these values in the service by using the System Management application.



Note: Metrics related to HCP for cloud scale instances and services are collected and provided by the System Management application. Collection of these metrics cannot be disabled.

Displaying the active object count

The Object Storage Management application displays a count of active objects stored in the system.

Object Storage Management application instructions

To display the Active objects report, select Storage. The **Storage** page opens.

The page displays a line graph showing the total number of active objects in the system over time. The maximum time period is one week.

Displaying metrics

You can use the metrics service to display or graph metrics, or use the service API to obtain metrics.

Object Storage Management application instructions

You can display and graph metrics using the metrics GUI.

To display metrics, click the app switcher menu (☰) and then select Prometheus. The metrics tool opens in a separate browser window.

The metrics tool is a third-party, open-source package. For information about using the metrics tool, see the documentation provided with the tool.

Available metrics

Metrics provide information about the operation of a service. Metrics are collected while the service is active. If a service restarts, its metrics are restarted.



Note: If a metric is measured over an interval (for example, `http_s3_servlet_requests_latency_seconds`), but doesn't have at least two data points, the value is reported as NaN.

Metrics from all services

The following metrics are available from all services.

Metric	Description
<code>http_healthcheck_requests_total</code>	The count of the total number of requests made to the health verification API.
<code>http_monitoring_requests_total</code>	The count of the total number of requests made to the monitoring API.
<code>scrape_duration_seconds</code>	The duration in seconds of the scrape (collection interval).
<code>scrape_samples_post_metric_relabeling</code>	The count of samples remaining after metric relabeling was applied.
<code>scrape_samples_scraped</code>	The count of samples the target exposed.
<code>up</code>	1 if the instance is healthy (reachable) or 0 if collection of metrics from the instance failed.

Data Lifecycle

The following metrics are available from the Data Lifecycle service. For these metrics, a policy can be any of these:

- CHARGEBACK_POPULATION
- DELETE_BACKEND_OBJECTS
- EXPIRE_FAILED_WRITE
- INCOMPLETE_MPU_EXPIRATION
- VERSION_EXPIRATION

Metric	Description
lifecycle_policy_accept_latency_seconds	The lifecycle policy acceptance processing latency in seconds
lifecycle_policy_completed	The total number of lifecycle policies completed
lifecycle_policy_conflicts	The total number of lifecycle policy conflicts
lifecycle_policy_deleted_backend_objects_count	The total number of objects deleted from backend storage by the policy DELETE_BACKEND_OBJECTS
lifecycle_policy_errors	<p>The total number of errors that occurred while executing lifecycle policy actions, in the categories:</p> <ul style="list-style-type: none"> ▪ General ▪ Listing ▪ Metadata ▪ S3
lifecycle_policy_examine_latency_seconds	The lifecycle policy examination processing latency in seconds
lifecycle_policy_expiration_completed_count	The total number of objects completely processed by the expiration policies DELETE_MARKER and PERMANENT_DELETE)
lifecycle_policy_list_latency_seconds	The lifecycle policy listing latency in seconds
lifecycle_policy_splits	The total number of lifecycle policy splits
lifecycle_policy_started	The total number of lifecycle policies started
lifecycle_policy_submitted	The total number of lifecycle policies submitted
rmq_all_sync_from_pending	The number of pending messages in the s3.mirroringEvents queue awaiting sync-from processing
rmq_all_sync_to_pending	The number of pending messages in the s3.mirroringEvents queue awaiting sync-to processing

Metric	Description
rmq_queued_messages	<p>The queue depth (number of messages) in these product queues:</p> <ul style="list-style-type: none"> ▪ s3.all - Unprocessed messages resulting from S3 operations. ▪ s3.mirroringEvents - Messages for objects that require mirroring out to an external bucket. This is the Sync-To backlog. ▪ s3.notificationEvents - Messages for objects that require S3 notification to external entities. This is the S3 Notification backlog. ▪ s3.mirrorInEvents - Messages for objects that require mirroring in from an external bucket. This is the Sync-From backlog. Note: This queue is limited in size to 1 million entries. There might be additional backlog in SQS. ▪ lifecycle.client-object-policy - Messages that define tasks for client object policies, including version expiration, delete marker expiration, and tombstone expiration. Expiration for versions and delete markers are defined using a bucket lifecycle policy. ▪ lifecycle.delete-backend - Messages that define tasks for reclaiming space from storage components. ▪ lifecycle.expire-mpu - Messages that define tasks for expiring multipart uploads. Expiration is defined using a bucket lifecycle policy. ▪ lifecycle.chargeback - Messages that define tasks for aggregating chargeback data. <p>Note: A task represents a range of objects. Each range can have many thousands of objects.</p>
s3_operation_count	The count of S3 operations (READ, WRITE, DELETE, and HEAD) per storage component

Metric	Description
s3_operation_error_count	The count of failed S3 operations (READ, WRITE, DELETE, and HEAD) per storage component
s3_operation_latency_seconds	The latency of storage component operations (READ, WRITE, DELETE, and HEAD) in seconds

MAPI Gateway

The following metrics are available from the MAPI Gateway service. These metrics are collected every five minutes.

Metric	Description
storage_available_capacity_bytes	The number of bytes free on an HCP S Series Node.
storage_total_capacity_bytes	The number of bytes total, available and used, on an HCP S Series Node.
storage_total_objects	The number of objects on an HCP S Series Node.
storage_used_capacity_bytes	The number of bytes used on an HCP S Series Node.

Each metric is reported with a label, `store`, identifying it as being either from a specific HCP S Series Node or the aggregate total. You can also retrieve the metrics using this label. For example, to retrieve the used storage capacity of the storage component `hcps10.company.com`, you would specify:

```
storage_used_capacity_bytes{store="hcps10.company.com"}
```

To retrieve the number of objects stored on the HCP S Series Node storage component `snode67.company.com`, you would specify:

```
storage_total_objects{instance="hcpcs_cluster:9992",job="MAPI-Gateway",store="snode67.company.com"}
```

To retrieve the used storage capacity of all available storage components, you would specify:

```
storage_used_capacity_bytes{store="aggregate"}
```



Note: If storage components other than HCP S Series Nodes are configured, aggregate totals aren't reported.

Message Queue

The Message Queue service supports a large number of general metrics. Information on these metrics is available at <https://github.com/rabbitmq/rabbitmq-prometheus/blob/master/metrics.md>.

Metadata Coordination

The following metrics are available from the Metadata Coordination service.

Metric	Description
mcs_copies_per_partition	The count of number of copies of each metadata partition of each key space (to verify protection).
mcs_disk_usage_per_instance	The total disk usage of each metadata instance.
mcs_disk_usage_per_partition	The disk usage of each metadata partition of each key space.
mcs_failed_moves_per_keyspace	The count of number of unsuccessful requests for metadata partition moves per keyspace.
mcs_failed_splits_per_keyspace	The count of number of unsuccessful requests for metadata partition splits per keyspace.
mcs_moves_per_keyspace	The count of number of successful requests for metadata partition moves per keyspace.
mcs_partitions_per_instance	The count of total number of metadata partitions per metadata instance (to verify balance).
mcs_splits_per_keyspace	The count of number of successful requests for metadata partition splits per keyspace.

Metadata Gateway

The following metrics are available from the Metadata Gateway service.

**Note:**

1. Client count metrics are an approximation and might not correspond to the actual count.
2. Depending on when garbage collection tasks run, the ratio of client objects size to stored objects size might show a discrepancy.

Metric	Description
async_action_count	The count of actions performed.
async_action_latency_seconds_bucket	A histogram for the duration, in seconds, of actions on buckets. For actions comprising multiple steps, this is the total of all steps.
async_action_latency_seconds_count	The count of action latency measurements taken.
async_action_latency_seconds_sum	The sum of action latency in seconds.
async_concurrency	A gauge for the number of concurrent actions.
async_duq_latency_seconds_bucket	A histogram for the duration, in seconds, of operations on the durable update queue.
async_duq_latency_seconds_count	The count of durable update queue latency measurements.
async_getwork_database_count	The number of driver work checks accessing the database.
async_getwork_optimized_count	The number of driver work checks avoiding the database.
async_duq_latency_seconds_sum	The sum of actions on durable update queue in seconds.
metadata_available_capacity_bytes	The free bytes per instance (node) for the Metadata Gateway service. The label <code>store</code> is either the instance or <code>aggregate</code> . Note: Because multiple service instances can run on a node, all consuming the same shared disk space, the value returned by this metric might be more than the actual capacity available.
metadata_clientobject_active_count	The count of client objects in metadata that are in the ACTIVE state.
metadata_clientobject_active_encrypted_count	The count of encrypted client objects in metadata that are in the ACTIVE state.

Metric	Description
metadata_clientobject_active_unencrypted_count	The count of unencrypted client objects in metadata that are in the ACTIVE state.
metadata_clientobject_and_part_active_space	the space occupied by client objects and parts in metadata that are in the ACTIVE state.
metadata_clientobject_part_active_count	The count of client object parts in metadata that are in the ACTIVE state.
metadata_storedObject_active_space	The space occupied by stored objects on the back-end storage components.
metadata_used_capacity_bytes	The used bytes per instance (node) for the Metadata Gateway service. The label <code>store</code> gives the domain name of the instance. Note: Because multiple service instances can run on a node, all consuming the same shared disk space, combining this value with the value of <code>metadata_available_capacity_bytes</code> won't give the total capacity of the service.
update_queue_inprogress	The count of update queue entries in progress.
update_queue_size	The size of the update queue.

Policy Engine

The following metrics are available from the Policy Engine service.

Metric	Description
mirror_failed	The count of objects unsuccessfully synchronized. The mirror (synchronization) types are IN and OUT.
mirror_general_errors	The count of general synchronization errors. The mirror (synchronization) types are IN and OUT. The client types are EXTERNAL_S3, HCPCS, and TRANSFER.

Metric	Description
	The error categories are AUTHENTICATION, METADATA, OPERATION_ABORTED, RESOURCE_NOT_FOUND, S3, SERVICE_UNAVAILABLE, and GENERAL.
mirror_mpu_bytes	The number of bytes synchronized as part of multi-part uploads. The mirror (synchronization) types are IN and OUT.
mirror_mpu_errors	The count of multi-part upload synchronization errors. The mirror (synchronization) types are IN and OUT. The client types are EXTERNAL_S3, HCPCS, and TRANSFER. The error categories are AUTHENTICATION, METADATA, OPERATION_ABORTED, RESOURCE_NOT_FOUND, S3, SERVICE_UNAVAILABLE, and GENERAL.
mirror_mpu_objects	The count of objects synchronized using multi-part uploads. The mirror (synchronization) types are IN and OUT.
mirror_skipped	The count of objects not synchronized. The mirror (synchronization) types are IN and OUT.
mirror_success	The count of objects successfully synchronized. The mirror (synchronization) types are IN and OUT.
mirror_whole_bytes	The number of bytes synchronized as whole objects. The mirror (synchronization) types are IN and OUT.
mirror_whole_errors	The count of non-multipart synchronization errors.

Metric	Description
	<p>The mirror (synchronization) types are IN and OUT.</p> <p>The client types are EXTERNAL_S3, HCPCS, and TRANSFER.</p> <p>The error categories are AUTHENTICATION, METADATA, OPERATION_ABORTED, RESOURCE_NOT_FOUND, S3, SERVICE_UNAVAILABLE, and GENERAL.</p>
mirror_whole_objects	<p>The count of objects synchronized as whole objects.</p> <p>The mirror (synchronization) types are IN and OUT.</p>
sync_from_bytes	The count of bytes synchronized from external storage (sync-from) by this instance.
sync_from_bytes_copied	The count of bytes synchronized by full copy from external storage (sync-from) by this instance.
sync_from_bytes_putcopied	The count of bytes synchronized by put-copy from external storage (sync-from) by this instance.
sync_from_object_count_failed	<p>The count of objects that failed to synchronize from external storage (sync-from) by this instance, grouped by class of error. The error classes are AUTHENTICATION, METADATA, OPERATION_ABORTED, RESOURCE_NOT_FOUND, S3, SERVICE_UNAVAILABLE, and UNKNOWN.</p>
sync_from_object_count_succeeded	The count of objects synchronized from external storage (sync-from) by this instance.
sync_from_object_size_total	Total size of object data synchronized from external storage (sync-from) by this instance.
sync_from_objects	Total size of object data synchronized from external storage (sync-from) by this instance.

Metric	Description
sync_to_bytes	The count of bytes synchronized to external storage (sync-to) by this instance.
sync_to_bytes_copied	The count of bytes synchronized by full copy to external storage (sync-to) by this instance.
sync_to_bytes_putcopied	The count of bytes synchronized by put-copy (previously copied) to external storage (sync-to) by this instance.
sync_to_object_count_failed	The count of objects that failed to synchronize to external storage (sync-to) by this instance, grouped by class of error. The error classes are AUTHENTICATION, METADATA, OPERATION_ABORTED, RESOURCE_NOT_FOUND, S3, SERVICE_UNAVAILABLE, and UNKNOWN.
sync_to_object_count_succeeded	The count of objects synchronized to external storage (sync-to) by this instance.
sync_to_objects	The count of objects synchronized to external storage (sync-to) by this instance.
sync_to_object_size_total	The total size of object data synchronized to external storage (sync-to) by this instance.

S3 Gateway

The following metrics are available from the S3 Gateway service.

Metric	Description
http_s3_servlet_errors_total	The count of total number of errors returned by the s3 servlet, grouped by error.
http_s3_servlet_get_object_response_bytes_per_bucket	The count of total bytes in the body of S3 GET object responses per bucket.
http_s3_servlet_get_object_response_bytes_total	The count of total bytes in the body of S3 GET object responses.
http_s3_servlet_ingest_object_bytes_per_bucket	The count of total objects ingested for the specified bucket.
http_s3_servlet_operations_total	The count of total number of S3 operations made to the s3 servlet for each method, grouped by operation.

Metric	Description
http_s3_servlet_post_object_bytes_total	The count of total bytes of objects posted to S3.
http_s3_servlet_put_copied_bytes_total	The count of total bytes of objects PUT copied (previously copied) to S3.
http_s3_servlet_put_object_bytes_total	The count of total bytes of objects PUT (previously copied) to S3.
http_s3_servlet_put_object_part_bytes_total	The count of total bytes of PUT part operations (previously copied) to S3.
http_s3_servlet_requests_histogram_latency_seconds	The latency in seconds as measured by a histogram timer, grouped by operation.
http_s3_servlet_requests_histogram_latency_seconds_bucket	The latency in seconds as measured by a histogram timer, grouped by bucket.
http_s3_servlet_requests_histogram_latency_seconds_count	The count of s3 servlet request observations; used with sum to determine average.
http_s3_servlet_requests_histogram_latency_seconds_sum	Sum of s3 servlet request latency in seconds; used with count to determine average.
http_s3_servlet_requests_latency_seconds	The latency in seconds as measured by a summary timer, grouped by operation.
http_s3_servlet_requests_latency_seconds:hour_average	The latency in seconds over the last hour as measured by a summary timer.
http_s3_servlet_requests_latency_seconds_count	
http_s3_servlet_requests_latency_seconds_sum	The sum of request latency in seconds.
http_s3_servlet_requests_per_bucket	The count of total put, get, or deletion requests made to the specified bucket.
http_s3_servlet_requests_total	The count of total number of requests made to the s3 servlet, grouped by method.
http_s3_servlet_unimplemented_api_request_total	The count of total number of requests made for unimplemented S3 methods.
http_s3_servlet_unimplemented_bucket_api_request_total	The count of total number of requests made for unimplemented S3 methods per bucket, grouped by API.

Metric	Description
http_s3_servlet_unimplemented_object_api_request_total	The count of total number of requests made for unimplemented S3 methods per object, grouped by API.
http_s3_servlet_unimplemented_service_api_request_total	The count of total number of requests made for unimplemented S3 methods per service, grouped by API.
http_s3_servlet_unknown_api_requests_total	The count of total number of requests made for unknown S3 methods, grouped by API.
notification_events_considered	The count of events considered that could potentially lead to notifications.
notification_events_notification_attempted	The count of events that had at least one notification message attempted.
notification_message_failures	The count of notification messages that were attempted but failed.
notification_message_parsing_failures	The count of candidate object events that could not be parsed.
notification_messages_sent	The count of notification messages that were successfully sent.
s3_operation_count	The count of S3 operations (READ, WRITE, DELETE, and HEAD) per storage component
s3_operation_error_count	The count of failed S3 operations (READ, WRITE, DELETE, and HEAD) per storage component
s3_operation_latency_seconds	The latency of storage component operations (READ, WRITE, DELETE, and HEAD) in seconds
s3select_total_bytes_scanned	The count of bytes scanned in the object
s3select_total_bytes_processed	The count of bytes processed by the request
s3select_total_bytes_returned	The count of bytes returned from the request
s3select_input_type	Count of requests by file type
s3select_output_type	Count of responses by file type

Examples of metric expressions

By using metrics in formulas, you can generate useful information about the behavior and performance of the HCP for cloud scale system.

Available capacity

The following expression graphs the total capacity of the storage component `store54.company.com` over time. Information is returned for HCP S Series Node storage components only. The output includes the label `store`, which identifies the storage component by domain name. The system collects data every five minutes.

```
storage_total_capacity_bytes{store="store54.company.com"}
```

The following expression graphs the used capacity of all HCP S Series Node storage components in the system over time. (This is similar to the information displayed on the **Storage** page.) Information is returned only if all storage components in the system are HCP S Series nodes. The output includes the label `aggregate`. The system collects data every five minutes.

```
storage_used_capacity_bytes(store="aggregate")
```

Growth of active-object count

The following expression graphs the count of active objects (`metadata_clientobject_active_count`) over time. (This is similar to the graph displayed on the **Storage** page.) You can use this formula to determine the growth in the number of active objects.

```
sum(metadata_clientobject_active_count)
```

Sum of update queues

The following expression graphs the size of all update queues. You can use this formula to determine whether the system is keeping up with internal events that are processed asynchronously in response to S3 activity. If this graph increases over time, you might want to increase capacity.

```
sum(update_queue_size)
```

Changes in S3 put requests over time

The following expression graphs the count of S3 put requests, summed across all nodes, at one-minute intervals. If you remove the specifier `{operation="S3PutObjectOperation"}` the expression graphs all S3 requests.

```
sum(rate(http_s3_servlet_operations_total{operation="S3PutObjectOperation"}[1m]))
```

Request time service levels

The following expression divides the latency of requests (`async_duq_latency_seconds_bucket`) in seconds by the number of requests (`async_duq_latency_seconds_count`), for the bucket `getWork` and requests less than or equal to 10 ms, and graphs it over time. You can use this formula to determine the percentage of requests completed in a given amount of time.

```
sum(rate(async_duq_latency_seconds_bucket{op="getWork",le="0.01"}[1m])) /
sum(rate(async_duq_latency_seconds_count{op="getWork"}[1m]))
```

Here is a sample graph of data from a lightly loaded system:



Request time quantile estimates

The following expression estimates the quantile for the latency of requests (`async_duq_latency_seconds_bucket`) in seconds for the bucket `getWork`. You can use this formula to estimate the percentage of requests completed in a given amount of time.

```
histogram_quantile(.9,
sum(rate(async_duq_latency_seconds_bucket{op="getWork"}[1m])) by (le))
```

Here is a sample graph of data from a lightly loaded system:



Tracing requests and responses

HCP for cloud scale uses an open-source software tool, running over HTTPS as a service, for service tracing through a browser.

The Tracing service supports end-to-end, distributed tracing of S3 requests and responses by HCP for cloud scale services. Tracing helps you monitor performance and troubleshoot possible issues.

Tracing involves three service instances:

- Tracing Query: serves traces
- Tracing Agent: receives spans from tracers
- Tracing Collector: receives spans from Tracing Agent service using Tchannel

Displaying traces

You can display traces using the tracing service GUI.

To begin tracing, click the app switcher menu (☰) and then select Jaeger. The tracing tool opens in a separate browser window.

When tracing, you can specify:

- Service to trace
- Operation to trace (all or specific) for each service
- Tags
- Lookback period (by default, over the last hour)
- Minimum duration
- Number of results to display (by default, 20)

The service displays all found traces with a chart giving the time duration for each trace. You can select a trace to display how the trace is served by different services in cascade and the time spent on each service.

For information about the tracing tool, see the documentation provided with the tool.

Traceable operations

The following operations are traceable.

Component	Operation
async-policy-engine	Action Pipeline Action: BucketIdToNameMapAction
	Action Pipeline Action: BucketLookupForAsyncPolicyAction

Component	Operation
	Action Pipeline Action: BucketOwnerIdToNameMapAction
	Action Pipeline Action: BucketUpdateSecondaryAction
	Action Pipeline Action: ClientObjectDispatchRemoveBackReferencesAction
	Action Pipeline Action: ClientObjectLookupAction
	Action Pipeline Action: ClientObjectModifyInProgressListAction
	Action Pipeline Action: ClientObjectModifyListAction
	Action Pipeline Action: ClientObjectUpdateSecondaryAction
	Action Pipeline Action: DequeueAction
	Action Pipeline Action: MetadataAction
	BUCKET
	CLIENT_OBJECT
	STORED_OBJECT_BACK_REFERENCE
balance-engine	BalanceCluster
	BalanceEngineOperation
	controlApi.ControlApiService
	RefreshCluster
client-access-service	Action Pipeline Action: BucketAuthorizationAction
	Action Pipeline Action: BucketCountLimitAction
	Action Pipeline Action: BucketCreateAction
	Action Pipeline Action: BucketRegionValidationAction
	Action Pipeline Action: BucketUpdateAclAction
	Action Pipeline Action: ClientObjectInitiateMultipartAction
	Action Pipeline Action: ClientObjectListInProgressMultipartAction
	Action Pipeline Action: ClientObjectListVersionsAction

Component	Operation
	Action Pipeline Action: ClientObjectSizeLimitAction
	Action Pipeline Action: ClientObjectTableLookupAction
	Action Pipeline Action: ClientObjectUpdateAclAction
	Action Pipeline Action: CompleteMultipartUploadAction
	Action Pipeline Action: DataContentAction
	Action Pipeline Action: DataDeletionAction
	Action Pipeline Action: NotAnonymousAuthorizationAction
	Action Pipeline Action: ObjectAuthorizationAction
	Action Pipeline Action: ObjectDataPlacementAction
	Action Pipeline Action: ObjectGetCurrentExpirationAction
	Action Pipeline Action: ObjectGetMultipartAbortDateAction
	Action Pipeline Action: ObjectGetUndeterminedExpirationAction
	Action Pipeline Action: ObjectLookupAction
	Action Pipeline Action: PartDataPlacementAction
	Action Pipeline Action: PutAclAction
	Action Pipeline Action: RequestBucketLookupAction
	Action Pipeline Action: RequestVersionIdValidationAction
	Action Pipeline Action: UploadIdValidationAction
	Action Pipeline Action: UserLookupBucketsAction
	Action Pipeline Action: VersionIdNotEmptyValidationAction
expiration-rules-engine	EvaluateOperation
foundry-auth-client	FoundryAuthorizeOperation
	FoundryValidateOperation

Component	Operation
jaeger-query	/api/dependencies
	/api/services
	/api/services/{service}/operations
	/api/traces
mapi-service	GET
	POST
metadata-client	BucketService/Create
	BucketService/List
	BucketService/ListBucketOwnerListing
	BucketService/LookupBucketNameById
	BucketService/LookupByName
	BucketService/UpdateACL
	ClientObjectService/CloseNew
	ClientObjectService/ClosePart
	ClientObjectService/DeleteSpecific
	ClientObjectService/List
	ClientObjectService/LookupLatest
	ClientObjectService/LookupSpecific
	ClientObjectService/OpenNew
	ClientObjectService/OpenPart
	ClientObjectService/setACLOnLatest
	ClientObjectService/Delete
	ConfigService/List
	ConfigService/LookupById
	ConfigService/Set
	StoredObjectService/Close
StoredObjectService/Delete	
StoredObjectService/List	

Component	Operation
	StoredObjectService/Lookup
	StoredObjectService/MarkForCleanup
	StoredObjectService/Open
	UpdateQueueService/SecondaryEnqueue
	UserService/LookupById
	UserService/LookupOrCreate
	UserService/UpdateAddAuthToken
metadata-coordination-service	Status.Service/GetStatus
metadata-gateway-service	Status.Service/GetStatus
	BucketService/Create
	BucketService/List
	BucketService/ListBucketOwnerListing
	BucketService/LookupBucketNameById
	BucketService/LookupByName
	BucketService/UpdateACL
	ClientObjectService/CloseNew
	ClientObjectService/ClosePart
	ClientObjectService/DeleteSpecific
	ClientObjectService/List
	ClientObjectService/LookupLatest
	ClientObjectService/LookupSpecific
	ClientObjectService/OpenNew
	ClientObjectService/OpenPart
	ClientObjectService/setACLOnLatest
	ConfigService/Delete
	ConfigService/List
	ConfigService/LookupById
	ConfigService/Set

Component	Operation
	StoredObjectService/Close
	StoredObjectService/Delete
	StoredObjectService/List
	StoredObjectService/Lookup
	StoredObjectService/MarkForCleanup
	StoredObjectService/Open
	UpdateQueueService/SecondaryEnqueue
	UserService/LookupById
	UserService/LookupOrCreate
	UserService/UpdateAddAuthToken
metadata-policy-client	PolicyService/ExecutePolicy
metadata-policy-service	ServiceStatus/GetStatus
	PolicyService/ExecutePolicy
	ScheduledDeleteBackendObjectsJob
	ScheduledDeleteFailedWritesJob
	ScheduledExpirationJob
	ScheduledIncompleteMultipartExpirationJob
storage-component-client	InMemoryStorageComponentVerifyOperation
	InMemoryStorageDeleteOperation
	InMemoryStorageReadOperation
	InMemoryStorageWriteOperation
storage-component-manager	StorageComponentManager Operation: Create
	StorageComponentManager Operation: List
	StorageComponentManager Operation: Lookup
	StorageComponentManager Operation: Update
tomcat-servlet	S3 Operation

Chapter 3: Services

Services perform functions essential to the health and function of the Hitachi Content Platform for cloud scale (HCP for cloud scale) system. The System Management application enables management of services.

For example, the S3 Gateway service serves S3 API methods and communicates with storage components, while the Watchdog service ensures that other services remain running.

Services provide cluster management and coordination, metadata coordination and caching, and external gateways.

Internally, services run in Docker containers on the instances of the system. The container orchestration framework supports cloud or on-premise deployment.

HCP for cloud scale is designed around an adaptive service deployment model that changes based on workload.

The starting point for storage component management is the **Dashboard** page of the System Management application. The procedures in this module begin at this page.

Service categories

Services are grouped into categories depending on what actions they perform.

Services are grouped into these categories:

- *Product services* enable HCP for cloud scale functions. For example, the S3 Gateway service serves S3 API methods and communicates with storage components. You can scale, move, and reconfigure product services.
- *System services* maintain the health and availability of the HCP for cloud scale system. For example, the Watchdog service ensures that other services remain running. You cannot scale, move, or reconfigure system services.

HCP for cloud scale services

The following table describes the services that HCP for cloud scale 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.
- 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 value of Container Memory for the service.

- Number of instances: Shows both:
 - The minimum number of instances on which a service must run to function properly.
 - The best number of instances on which a service should run. If the system includes more than the minimum number of instances, you should take advantage of the instances by running services on them.
- Service unit cost: For HCP for cloud scale, you can safely ignore these values.
- Whether the service is stateful (that is, it saves data permanently to disk) or stateless (that is, it does not save data to disk).
- 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 HCP for cloud scale services, you cannot set the size of Max Heap Size larger than the value of the setting Container Memory. For other services, you should not set the size of Max Heap Size larger than the value of the setting Container Memory.

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
Product services: These services perform HCP for cloud scale functions. You can move and reconfigure these services.		
Cassandra Decentralized database, used to stores some configuration and system update packages	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 2400.0 (2.4 GB). ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. 	RAM needed per instance: 2.4 GB Number of instances: minimum 3, best All Service unit cost: 10 Stateful or stateless? Stateful 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>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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1200m. ▪ 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. <p>Advanced Options</p> <p>Compaction Frequency: How often the database is compacted. The options are Weekly (default) and Daily.</p> <p>Caution: Changing this setting can negatively affect the service. Use with caution.</p>	
<p>Chronos Job scheduler</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 receives more CPU resources than other processes (including other service Docker containers) running on the instance. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 356 MB. 	<p>RAM needed per instance: 712 MB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 1</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>Data Lifecycle</p> <p>Processes lifecycle policies</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; a good initial value is 4 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 384 MB; a good initial value is 2 GB. 	<p>RAM needed per instance: 4 GB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes (but not recommended on master instances)</p>
<p>Elasticsearch</p> <p>Indexes metrics and event logs</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 2000.0 (2 GB); a good initial value is 10 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. 	<p>RAM needed per instance: 10 GB</p> <p>Number of instances: minimum 3, best All</p> <p>Service unit cost: 25</p> <p>Stateful or stateless? Stateful</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB; a good initial value is 8 GB. ▪ 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. 	
<p>Kafka</p> <p>Handles metrics and event logs</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 2000.0 (2 GB). ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1 GB. 	<p>RAM needed per instance: 2 GB</p> <p>Number of instances: minimum 3, best All</p> <p>Service unit cost: 5</p> <p>Stateful or stateless? Stateful</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>Key Management Server</p> <p>Manages storage component encryption keys</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 2000.0 (2 GB). ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.5. <p>Service Options</p> <p>None.</p>	<p>RAM needed per instance: 2 GB</p> <p>Number of instances: minimum 1, best 2 or more</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>
<p>Logstash</p> <p>Handles metrics and event logs</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 receives more CPU resources than other processes (including other service Docker containers) running on the instance. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 350 MB. 	<p>RAM needed per instance: 700 MB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>MAPI Gateway</p> <p>Serves MAPI endpoints</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; a good initial value is 2 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 384 MB; a good initial value is 1 GB. ▪ Total Storage Capacity Alert Threshold: Display an alert when the total storage capacity free to store object data in the system goes below this value. Type a threshold value. You must specify the suffix % (percent of total), K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). If blank, alerts are disabled. The default is 30%. ▪ Per Storage Component Capacity Alert Threshold: Display an alert when the storage capacity free to store object data in any storage component goes below this value. Type a threshold value. You must specify the suffix % (percent of total), K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes) If blank, alerts are disabled. The default is 250GB. 	<p>RAM needed per instance: 2 GB</p> <p>Number of instances: minimum 1, max 1</p> <p>Service unit cost: 5</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>Message Queue</p> <p>Coordinates and distributes messages 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 2048 MB; a good initial value is 8 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.5. <p>Service Options</p> <ul style="list-style-type: none"> ▪ Consumer Timeout: The time in milliseconds that the service waits before timing out an unacknowledged message, closing the consumer's channel, and returning the message to the queue. Type an integer number of milliseconds. The default is 172800000 ms (48 hours). If messages (for example, for synchronization of large files) take longer than that to be acknowledged by the consumer, increase this value. 	<p>RAM needed per instance: 8 GB</p> <p>Number of instances: minimum 3, best 3</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateful</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes (but not recommended on master instances)</p>
<p>Metadata Cache</p> <p>Cache for HCP for cloud scale 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 1024 MB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. 	<p>RAM needed per instance: 1024 MB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

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 Size: Maximum amount of memory to allocate to the Java heap for each instance of the service. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 384 MB. 	
<p>Metadata Coordination</p> <p>Coordinates Metadata Gateway service instances and coordinates scaling and balancing of 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; a good initial value is 4 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 384 MB; a good initial value is 2 GB. 	<p>RAM needed per instance: 4 GB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 5</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>Metadata Gateway</p> <p>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; a good initial value is 64 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 32 GB. 	<p>RAM needed per instance: 64 GB</p> <p>Number of instances: minimum 3, best All</p> <p>Service unit cost: 50</p> <p>Stateful or stateless? Stateful</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes (but not recommended on master instances)</p>
<p>Metrics</p> <p>Gathers metrics from all services and instances and supplies them to GUI and API</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; a good initial value is 6 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. 	<p>RAM needed per instance: 6 GB</p> <p>Number of instances: minimum 1, best 1</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateful</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

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. Type 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. Type a path. The default is <code>tsdb/</code>. ▪ Prometheus Database Retention: The number of days to retain files. Type an integer number of days. You can optionally specify the suffix d (days). The default is 15 days. 	
<p>Policy Engine</p> <p>Executes system policies</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; a good initial value is 4 GB (8 GB if synch-to, synch-from, or notifications are used). ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB; a good initial value is 2 GB (4 GB if synch-to, synch-from, or notifications are used). 	<p>RAM needed per instance: 4 GB</p> <p>Number of instances: minimum 1, best All</p> <p>Service unit cost: 25</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes (but not recommended on master instances)</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>S3 Gateway</p> <p>Serves S3 API methods 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; a good initial value is 16 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <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. It's best to set this value to half the size of the container memory. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 384 MB; a good initial value is 8 GB. <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. Type an integer. The default is 100 request headers. <p>HTTPS Options</p> <ul style="list-style-type: none"> ▪ SSL Ciphers: A comma-separated list of ciphers used to encode SSL traffic. Changing the list causes the service to redeploy. 	<p>RAM needed per instance: 16 GB</p> <p>Number of instances: minimum 1, best All</p> <p>Service unit cost: 25</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes (but not recommended on master instances)</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
<p>Tracing Agent</p> <p>Listens for incoming tracing of S3 API and MAPI calls, batches them, and sends them to Tracing Collector service</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; a good initial value is 2 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <p>Service Options</p> <ul style="list-style-type: none"> ▪ Collector TChannel Hostname: Type a host name. The default is localhost. ▪ Collector TChannel Port: Type a port number. The default is 14267. 	<p>RAM needed per instance: 2 GB</p> <p>Number of instances: minimum All</p> <p>Service unit cost: 1</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>
<p>Tracing Collector</p> <p>Collects traces from Tracing Agent service instances and stores them in tracing database</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; a good initial value is 8 GB. ▪ CPU: The relative CPU usage weight for the service'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. Type a decimal number. The default is 0.1. <p>Service Options</p> <ul style="list-style-type: none"> ▪ ElasticSearch Hostname: Type a host name. The default is localhost. ▪ ElasticSearch Port: Type a port number. The default is 9200. ▪ Sampling Rate: The sampling rate for all clients implementing remote sampling. Type a number between 0 and 1 inclusive. The default is 1. 	<p>RAM needed per instance: 8 GB</p> <p>Number of instances: minimum 1, best All</p> <p>Service unit cost: 10</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
	<ul style="list-style-type: none"> ▪ Max open index age: How long to keep tracing indexes open in the database, in days. Type a value from 1 to 365 days inclusive. The default is 30 days. ▪ Max index age: How long to keep tracing indexes in the database, in days. Type a value from 1 to 365 days inclusive. The default is 60 days. 	
<p>Tracing Query</p> <p>UI and API endpoint access for distributed tracing for S3 API 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 receives more CPU resources than other processes (including other service Docker containers) running on the instance. Type a decimal number. The default is 0.1. <p>Service Options</p> <ul style="list-style-type: none"> ▪ ElasticSearch Hostname: Type a host name. The default is localhost. ▪ ElasticSearch Port: Type a port number. The default is 9200. 	<p>RAM needed per instance: 768 MB</p> <p>Number of instances: minimum 1, best All</p> <p>Service unit cost: 5</p> <p>Stateful or stateless? Stateless</p> <p>Persistent or floating? Floating</p> <p>Supports volume configuration? No</p> <p>Single or multiple types? Single</p> <p>Scalable? Yes</p>
<p>System services: These services manage system resources and ensure that the HCP for cloud scale system remains available and accessible. These services are persistent and cannot be moved, scaled, or reconfigured.</p>		
<p>Admin App</p> <p>The System Management application</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. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB. 	<p>RAM needed per instance: N/A</p> <p>Number of instances: N/A</p> <p>Persistent or floating? Persistent</p> <p>Supports volume configuration? Yes</p> <p>Single or multiple types? Single</p> <p>Scalable? No</p>
<p>Cluster Coordination</p> <p>Manages hardware resource allocation</p>	<p>None.</p>	<p>RAM needed per instance: N/A</p> <p>Number of instances: N/A</p> <p>Persistent or floating? Persistent</p>

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
		Supports volume configuration? No Single or multiple types? Single Scalable? No
Cluster Worker Agent for Cluster Coordination on each instance; reports on resource utilization and availability, deploys services	None.	RAM needed per instance: N/A Number of instances: N/A Service unit cost: 5 Persistent or floating? Persistent Supports volume configuration? Yes Single or multiple types? Single Scalable? No
Network Proxy Network request load balancer	Security Protocol: Select which Transport Layer Security (TLS) versions to use: <ul style="list-style-type: none"> ▪ TLS 1.2 ▪ TLS 1.3 SSL Ciphers: To use another cipher suite, type it here. Custom Global Configuration: Select Enable Advanced Global Configuration to enable adding custom parameters to the HAProxy "global" section. Custom Defaults Configuration: Select Enable Defaults Configuration to enable adding custom parameters to the HAProxy "global" section.	RAM needed per instance: N/A Number of instances: N/A Service unit cost: 1 Persistent or floating? Persistent Supports volume configuration? Yes Single or multiple types? Single Scalable? No
Sentinel Runs internal system processes and monitors the health of other services	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. Type an integer number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 1024 MB; a good initial value is 8.6 GB. 	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 Deployment	None.	RAM needed per instance: N/A Number of instances: N/A

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
Handles deployment of high-level services (that is, the services that you can configure)		Persistent or floating? Persistent Supports volume configuration? Yes Single or multiple types? Single Scalable? No
Synchronization Coordinates service configuration settings and other information across service 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. Type 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 Responsible for initial system startup; monitors other System services and restarts them if necessary	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. Type 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 Service unit cost: 5 Persistent or floating? Persistent Supports volume configuration? Yes Single or multiple types? Single Scalable? No

Viewing services

You can use the Admin App, CLI commands, or REST API methods to view the status of all services for the system.

Viewing all services

Admin App instructions

Procedure

1. To view the status of all services, in the Admin App, click **Services**.

For each service, the page shows:

- The service name
- The service state:
 - **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 functions, see [Monitoring service operations \(on page 107\)](#).

- **Balancing:** The service is running normally, but performing background maintenance.
- **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

Procedure

1. To view the detailed status for an individual service, select the service in the **Services** window.

In addition to status information, the window 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 the row for an instance in the **Instances** section.

- **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:

- **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.
- **Service unit cost:** The number of service units required to run the service on one instance.
- **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.
- **Events:** A list of all system events for the service.

Related CLI commands

getService

listServices

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /services/query

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Listing service ports

You can list service port information for ports available for customer use.

You can list public service ports using an API without an access token.

Related API method

```
POST /public/discovery/get_service_port
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

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 more instances, fewer instances, or different instances.

You can change a service to run on:

- Additional instances (for example, to improve service performance and availability)
- Fewer instances (for example, to free up resources on an instance for running other services)
- A different set of instances (for example, to retire 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.

Moving and scaling services with multiple types

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

Best practices for distributing services

- Monitor resource usage (CPU, RAM, disk) and services such as Prometheus regularly and adjust the scaling of services across instances as needed.
- Ensure that there are enough instances that the cluster can still manage the volume and growth of objects if one or even two instances fail.
- Distributing instances of product services onto master instances is discouraged. For more information refer to [Best practices for system sizing and scaling \(on page 196\)](#).

Considerations

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



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

Relocating services

To manually relocate a service, in the Admin App:

Procedure

1. Select **Services**.
The **Services** page opens, displaying the services and system services.
2. Select the service that you want to scale or move.
Configuration information for the service is displayed.
3. Click **Scale**, and if the service has more than one type, select the instance type that you want to scale.

The next step depends on whether the service is floating or persistent (non-floating).

4. If the service is a floating service, you are presented with options for configuring an instance pool. For example:

The screenshot displays the configuration page for the MAPI-Gateway service. At the top, there are buttons for REPAIR and UPDATE. Below this, a summary bar shows Average CPU Usage at 1.02%, Memory Used at 531.2 MB of 768.0 MB, and Disk Used at 40.0 kB. A navigation bar includes tabs for INSTANCES, VOLUMES, NETWORK, CONFIGURATION, SCALE (which is active), and EVENTS. Below the navigation bar, a summary box shows Service Unit Cost, Total: 5 Per Instance: 5, Service Units In Use, and 574 of Unlimited. The main section is titled 'Service Instance Configuration' and shows 'Service Instances' set to 1, with a checkbox for 'All Available Instances' checked. A note below states: 'The number of service instances that should be run in the pool. Minimum: 1 Maximum: 3'. Below this is the 'Instance Pool' section, which lists three pools with their respective IP addresses, service counts, allocated service units, and load averages.

IP Address	Services	Allocated Service Units	Load Average
172.18.46.50	18	200	0.58
172.18.46.51	16	197	0.93
172.18.46.52	13	177	0.30

- a. In the box **Service Instances**, specify the number of instances on which the service should be running at any time.

b. Configure the instance pool:

- For the service to run on any instance in the system, select **All Available Instances**.

With this option, the service can be restarted on any instance in the instance pool, including instances that were added to the system after the service was configured.

- For the service to run on a specific set of instances, clear **All Available Instances**. Then:
 - To remove an instance from the pool, select it from the list **Instance Pool**, on the left, and then click **Remove Instances**.
 - To add an instance to the pool, select it from the list **Available Instances**, on the right, and then click **Add Instances**.

- If the service is a persistent (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, then click **Next**:

- To remove the service from the instances it's currently on, select one or more instances from the list **Selected Instances**, on the left, and then click **Remove Instances**.
- To add the service to other instances, select one or more instances from the list **Available Instances**, on the right, and then click **Add Instances**.

6. Click **Update**.

The **Processes** page opens, and the **Service Operations** tab displays the progress of the service update as "Running." When the update finishes, the service shows "Complete."

Next steps

After reconfiguration, the service might take a few minutes to appear on the **Services** page.

Related CLI commands

updateServiceConfig

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /services/configure

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Scaling Metadata Gateway instances

The HCP for cloud scale software lets you deploy an instance of the Metadata Gateway service on every node in your system. You can scale the number of instances up or down as needed.

The Metadata Coordination service manages Metadata Gateway scaling. The service does the following:

- Constantly monitors the Metadata Gateway service and balances data among Metadata Gateway instances as needed
- Moves data into new Metadata Gateway instances
- Moves data out of a Metadata Gateway instance set for removal

Scaling up

Use the System Management application to add new Metadata Gateway instances. You can add more than one instance at a time.

Scaling down

Use the System Management application to remove a Metadata Gateway instance. Before you scale down Metadata Gateway instances, consider the following:

- You can only remove a Metadata Gateway instance from the system when there is one or zero Metadata Gateway instances down.



Note: If more than one instance is down, call Support to remove a Metadata Gateway instance.

- You cannot remove a Metadata Gateway instance when there are only three instances. You first need to add a new Metadata Gateway instance.
- You can only remove one Metadata Gateway instance at a time.

Failure recovery

If a Metadata Gateway instance is down, the data in this instance becomes underprotected. To resolve this situation, remove the Metadata Gateway instance that is down so that the Metadata Gateway service can recover the data protection. You should first add a new Metadata Gateway instance before removing the instance that is down. This ensures that the system keeps the same performance and capacity usage and also that there is a suitable target instance to recover the data protection. When removing the Metadata Gateway instance, the considerations on scaling down services apply.

A snapshot shows the current state of the state machine from a leader node (service instance) to any follower service instance that is out of synch. If a leader node runs out of space to store snapshots and can't send out its latest snapshot, the follower node cannot resynchronize. If this happens, bring down the leader service instance, increase its storage space, and restart the service.

Configuring service settings

You can configure settings for some of the services that the system runs.



Note: If you make an unwanted change to a service configuration, wait for the configuration to finish before correcting the error.

Configuring service settings

Procedure

1. Select **Dashboard > Services**.
2. Select the service you want to configure.
3. On the **Configuration** tab, configure the service.
4. Click **Update**.

Related CLI commands

`updateServiceConfig`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST `/services/configure`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Repairing services

If a service becomes slow, unresponsive, or shows a status of Failed, you can repair it. If you change the configuration of a service you use the same process to restart it.

Repairing a service stops and restarts the service on each instance on which it's running.

If you change the cluster name (cluster hostname), you must repair the S3 Gateway services for the change to take effect.

If you regenerate or upload an SSL certificate, you must repair the S3 Gateway and MAPI services for the change to take effect.

If you upload an SSL certificate for access to a remote system for bucket synchronization, you must repair the Policy Engine and MAPI services for the change to take effect.



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

Procedure

1. Select **Dashboard > Services**.
2. Select the service you want to repair.
3. Click **Repair**.
The **Processes** window opens, displaying a progress bar for the repair process.

Configuring TLS cipher suite

By default, the S3 Gateway service supports the Transport Layer Security (TLS) 1.2 and 1.3 protocols to establish secure links to client nodes. TLS uses a suite of cryptographic algorithms to provide security for HTTPS traffic. The default cipher suite used by the S3 Gateway service is the Open Web Application Security Project (OWASP) security recommendation for Tomcat. You can provide your own cipher suite.



Note: Contact Support if you need to use the TLS 1.0 or 1.1 protocols.

Procedure

1. Select **Dashboard > Services**.
2. Select the service **S3-Gateway**.
3. On the **Configuration** tab, in the **HTTPS Options** section, enter the new cipher suite in the field **SSL Ciphers**.
4. Click **Update**.
The service redeploys.

Avoiding Message Queue shutdown

If two of the three Message Queue service instances fail, the service shuts down. To avoid the possible loss of queued messages, resolve any situation in which only two service instances are running.

To protect messaging consistency, the Message Queue service always has three service instances. To prevent being split into disconnected parts, the service shuts down if half of the service instances fail. In practice, messaging stops if two of the three instances fail.

Do not let the service run with only two instances, because in that scenario if one of the remaining instances fails, the service shuts down. However, when one of the failed instances restarts, messaging services recover and resume.

To protect the Message Queue service, immediately address a node failure where an instance cannot be restarted, because if two service instances are lost and cannot be recovered, the service cannot recover its previous state. You can still add new instances to form a new cluster, but messages that were queued are lost.

In the case of such a multi-node failure, the best practice is to restart the Policy Engine service instances one at a time after the Message Queue service cluster is re-formed. This forces the service instances to recover configurations that might have been missed while the Message Queue service was down. Additionally, after the Message Queue service cluster is re-formed, bucket sync-to events that were in the messaging queues are lost, so you might need to regenerate bucket sync-to events for such objects.

The cluster forms based on instance names, including the IP address of the node on which an instance runs. Therefore, changing node configurations such as IP addresses can cause nodes to be permanently removed from the cluster, possibly triggering a shutdown. If this happens, first add instances to the messaging service. Ensure the instances synchronize with the cluster before taking nodes offline or changing node configurations such as IP addresses. This way, the cluster can always keep over half of its instances running.

Service restart after internal network interruption

If the internal network is disconnected in one HCP for cloud scale node and then restored, the services Cassandra, Elasticsearch, and Kafka might not come back online automatically on the node, leaving the system underprotected.

If this happens, restart the `hcpcs` service on the node.

Chapter 4: Monitoring

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

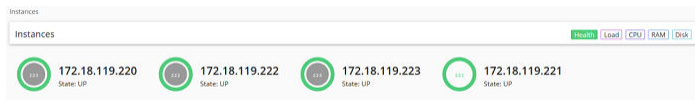
Monitoring instances

The **Instances** page lets you monitor instances (nodes) in the system. You can use the Admin App, CLI commands, or REST API methods to view a list of all instances in the system.

Viewing all instances


To view all instances, in the Admin App, click 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	<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 best-practice service unit limit for the instance.</p> <p>An instance with a higher number of service units is likely to be more heavily used 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 best-practice service unit limit.</p> <p>The Instances page displays a red bar for instances running more than the best-practice service unit limit.</p>

Property	Description
	Service Units  138 of 100 recommended Service Units
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"> ▪ 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 Admin App:

Procedure

1. Click **Dashboard > Instances**.
2. Select 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:
 - **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 \(on page 107\)](#).

- **Balancing:** The service is running normally, but performing background maintenance.
- **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 commands

getInstance

listInstances

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /instances

GET /instances/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Monitoring services

The **Services** page lets you view information about service instances. You can use the Admin App, CLI commands, or REST API methods to view the status of all services for the system.

Viewing all services

To view the status of all services, in the Admin App, click Services.

For each service, the page shows:

- The service name
- The service state:
 - 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. \(on page 107\)](#)

- 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, select the service on the Services page.

In addition to the status information, 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, select the row for an instance in the Instances section.
- 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.

- 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.
- Service unit cost: The number of service units required to run the service on one instance.

- 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.
- Events: A list of all system events for the service.

Related CLI commands

getService

listServices

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /services/query

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 Admin App, CLI commands, or REST API methods to monitor all service operations. These include:

- The initial deployments of services when the system was installed.
- Service relocations that you begin.

For each one, 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

Admin App instructions

Procedure

1. Select **Dashboard > Processes**.

Result

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

Related CLI commands

listSystemTasks

getSystemTask

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /tasks/system

GET /tasks/system/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Monitoring system processes

You can use the Admin App, CLI commands, or REST API methods to view the progress of internal system processes. These processes 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



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

Admin App instructions

Procedure

1. In the Admin App, select **Processes**.
2. To view the currently running processes, select the **System** tab.
3. To view the scheduled processes, select the **Scheduled** tab.

Related CLI commands

listSystemTasks

getSystemTask

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /tasks/system

GET /tasks/system/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Monitoring objects

You can use the REST API to configure and generate chargeback reports for objects on the system. Users can generate a report for one or more of the buckets they own. An administrator can generate a report for a user or a list of one or more buckets.

Generating a system chargeback report

You can use a REST API method to generate a system chargeback report. You can display a report for a specific user or a list of one or more buckets.

Related REST API methods

POST /chargeback/system/get_report

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Generating a user chargeback report

You can use a REST API method to generate a chargeback report for a user. Users can display a report for a specific bucket, a list of buckets, or all buckets that they own.

Related REST API methods

POST /chargeback/user/get_report

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

System events

Your system maintains a log of system events that you can view using the Admin App, CLI commands, or REST API methods.

Admin App instructions

Procedure

1. To view all system events, in the Admin App, click **Events**.

Related CLI commands

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 \(on page 200\)](#).

Related REST API methods

POST /events

To view events through the REST API, 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"
  ]
}
```

You can get help on specific REST API methods for the Admin App at [REST API - Admin](#).

HCP for cloud scale events

Most events are generated by and reported through the Object Storage Management application.

Events are written to `syslog`. Additionally, alerts corresponding to some events are displayed in the HCP for cloud scale applications.



Note: The System Management application reports service-related events using the IDs 6006 (service information), 6007 (service warning), and 6008 (service error).

The following table lists HCP for cloud scale events.

ID	Severity	Message	Description
1109	WARNING	Installation of package <i>package</i> failed: <i>reason</i>	The installation of the specified package failed for the specified reason.

ID	Severity	Message	Description
2004	SEVERE	<i>instance</i> instance with IP <i>ip_address</i> is error.	
2005	WARNING	Instance with IP <i>ip_address</i> value is at usage.	
2006	SEVERE	Instance with IP <i>ip_address</i> value is at usage.	
3002	WARNING	Low-level <i>service_name</i> service on instance <i>instance</i> exited abnormally. Restarting.	The specified service exited abnormally and is restarting.
5213	WARNING	A certificate in the SSL server certificate chain for this system expires soon. If the certificate chain expires, users won't be able to access the system.	This event applies only to system certificates, not client (storage component) certificates.
5214	WARNING	The SSL server certificate chain for this system contains an expired certificate. Users cannot access the system until the certificate chain is replaced.	This event applies only to system certificates, not client (storage component) certificates.
6001	WARNING	Service <i>service</i> is balancing.	
6002	WARNING	Service <i>service</i> is under-protected.	The number of service instances has fallen below the required minimum.
6003	SEVERE	Service <i>service</i> has failed.	
6006	INFO	Service Information: Default Retention configuration <i>policy_name</i> bucket ' <i>bucket_name</i> '	The default retention policy <i>policy_name</i> for the specified bucket has been updated.
6006	INFO	Service Information: Failed to Retrieve Storage Capacity Information	The system could not retrieve capacity information from storage component <i>id</i> . Verify the storage component configuration.

ID	Severity	Message	Description
6006	INFO	Service Information: Lifecycle policy {CREATE UPDATE DELETE} bucket ' <i>bucket_name</i> '	The lifecycle policy for the specified S3 bucket has been either created, updated, or removed.
6066	INFO	Service Information: Lifecycle policy deleted for bucket ' <i>bucket_name</i> '	The lifecycle policy for the specified S3 bucket has been removed.
6066	INFO	Service Information: Notifications configuration <i>notification_rule</i> bucket ' <i>bucket_name</i> '	Bucket notification has been updated.
6066	INFO	Service Information: Replication policy <i>policy_name</i> ' <i>bucket_name</i> '	The replication policy <i>policy_name</i> has been updated for the specified bucket.
6066	INFO	Service Information: Replication policy deleted for bucket ' <i>bucket_name</i> '	Bucket replication has been stopped.
6006	INFO	Service Information: S3 Encryption setting updated to <i>value</i>	The S3 encryption setting has been updated to the specified value.
6006	INFO	Service Information: Serial number updated to <i>value</i>	The HCP for cloud scale serial number has been changed to the specified value.
6006	INFO	Service Information: <i>setting_name</i> was set to <i>value</i>	The specified S3 setting has been changed to the specified value. If this was intended no action is needed.
6006	INFO	Service Information: Single Storage Component Available Capacity Low	The available capacity for object data of storage component <i>id</i> is now below the specified value. You might need additional capacity.
6006	INFO	Service Information: Storage component ' <i>id</i> ' created	The storage component <i>id</i> has been created.

ID	Severity	Message	Description
6006	INFO	Service Information: Storage component ' <i>id</i> ' is now <i>state</i>	The specified storage component is in one of the following states: <ul style="list-style-type: none"> ▪ ACTIVE ▪ INACTIVE ▪ UNVERIFIED
6006	INFO	Service Information: Storage component ' <i>id</i> ' updated: <i>configuration</i>	The specified storage component has been updated. <i>configuration</i> lists the changes.
6006	INFO	Service Information: System Available Capacity Low	The available capacity for object data of the system is now below the specified value. You might need to plan for additional capacity.
6007	WARNING	Certificate for SubjectDN <i>dn</i> will expire in <i>n</i> days	The SSL certificate for the specified client sync-to or sync-from target (specified by its Distinguished Name) is set to expire in <i>n</i> days. If the certificate expires, HCP for cloud scale will not be able to synchronize to or from the target. You might need to obtain a new client certificate.
6007	WARNING	Service Warning: Certificate for Storage component ' <i>id</i> ' is about to expire in ' <i>n</i> ' days	The SSL certificate for the specified storage component is set to expire in <i>n</i> days. If the certificate expires, HCP for cloud scale will not be able to read from or write to the storage component. You might need to obtain a new certificate.
6007	WARNING	Service Warning: Metadata-Coordination cannot communicate with Sentinel service to get state information	The Metadata Coordination service can't communicate with the Sentinel service.

ID	Severity	Message	Description
6007	WARNING	Service Warning: Storage component ' <i>id</i> ' is now INACCESSIBLE	The specified storage component is inaccessible.HCP for cloud scale cannot read from or write to the storage component.
6008	SEVERE	Certificate for SubjectDN <i>dn</i> expired on <i>dd-mmm-yyyy</i>	The SSL certificate for the specified client sync-to or sync-from target (specified by its Distinguished Name) had expired. HCP for cloud scale cannot synchronize to or from the target. You must obtain a new client certificate.
6008	SEVERE	Service Error: Storage Component Certificate has expired.	The SSL certificate for a storage component has expired. HCP for cloud scale cannot read from or write to the storage component. You must obtain a new certificate.
6008	SEVERE	Service Error: There is a critical issue with the Metadata Gateway database. Shutting down the Metadata Gateway Service.	
6008	SEVERE	Service Error: The vault service cannot be reached.	No connection to the active vault node can be established.
6008	SEVERE	Service Error: The vault service has a node that we can't connect to. Node IP: <i>ip_address</i>	One of the vault nodes can't be reached. If other active nodes are available service continues, but attend to this issue immediately.
6008	SEVERE	Service Error: The vault service has a sealed node. Please unseal it using the unseal keys you obtained when you turned on encryption. Node IP: <i>ip_address</i>	One of the vault nodes is sealed. If other active nodes are available service continues, but attend to this issue immediately. Unseal it using the unseal keys you obtained when you turned on encryption.

ID	Severity	Message	Description
6008	SEVERE	Service Error: Vault Service Completely Sealed. Please unseal it using the unseal keys you obtained when you turned on encryption	All nodes of the vault service (Key Management Server service) are sealed. Unseal using the unseal keys you obtained when you turned on encryption.
8001	WARNING	Starting update from <i>version</i> to <i>version</i> .	
8002	WARNING	Update in progress from <i>version</i> to <i>version</i> .	
8003	SEVERE	Update from <i>version</i> to <i>version</i> prechecks failed.	Update failed because a pre-update verification failed.
8004	SEVERE	Update from <i>version</i> to <i>version</i> failed.	The update failed.
8007	WARNING	Completed update from <i>version</i> to <i>version</i> .	The update succeeded.
9001	WARNING	Signal Source <i>source</i> failed. <i>Reason</i> .	
9002	WARNING	Workflow <i>workflow</i> is not running and will be restarted.	The Monitor-App workflow is not running. It will be restarted.
9003	WARNING	The Monitor-App is not processing data fast enough. Dashboard data is more than <i>n</i> minutes behind the latest data from the source. While in this state, monitors might not be triggered or might be triggered unexpectedly. If this alert persists, the system might be undersized. Consider adding more instances.	The Monitor-App is starting to fall behind.

ID	Severity	Message	Description
9004	SEVERE	The Monitor-App is not processing data fast enough. Dashboard data is more than <i>n</i> minutes behind the latest data from the source. While in this state, monitors might not be triggered or might be triggered unexpectedly. If this alert persists, the system might be undersized. Consider adding more instances.	The Monitor-App has fallen behind.

Alerts

Alert messages notify you of situations that need attention. Alerts can have a severity of Info, Warning, Severe, or Critical. You can view system alerts through the Admin App, CLI, or REST API, and storage component alerts through the Object Storage Management app.

Each alert corresponds to a system event.



Tip: Before you deal with an alert, refresh the page. The underlying condition might have changed since the alert was raised.

System alerts

Severity	Alert Description	Action
Severe	Instance <i>ip-address</i> disk usage severe threshold	The specified instance has less than 10% free disk space. Add additional storage to the instance. Important: If an instance runs out of disk space, the system can become unresponsive.
Severe	Master Instance <i>ip-address</i> is down	Do one of these: <ul style="list-style-type: none"> ▪ Restart the instance hardware or virtual machine. ▪ Restart the script <code>run</code> on the instance. This script is located in the folder <code>bin</code> in the installation folder.

Severity	Alert Description	Action
Severe	Service is down	<p>Verify the health of the 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 script <code>run</code> on the instance. This script is located in the folder <code>bin</code> in the installation folder. <p>Otherwise, if the instances are healthy and the problem persists, contact Support.</p>
Severe	Worker Instance <i>ip-address</i> is down	<p>Do one of these:</p> <ul style="list-style-type: none"> ▪ Restart the instance hardware or virtual machine. ▪ Restart the script <code>run</code> on the instance. This script is located in the folder <code>bin</code> in the installation folder.
Warning	Instance <i>ip-address</i> 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 can become unresponsive.</p>
Warning	Package installation failed	Your system failed to install a package that you uploaded.
Warning	Service below recommendation	The service is currently running on fewer than the minimum number of instances. Configure this service to run on additional instances.
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>Verify the health of the 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 script <code>run</code> on the instance. This script is located in the folder <code>bin</code> in the installation folder. <p>Otherwise, if the instances are healthy and the problem persists, contact Support.</p>
Warning	SSL server certificate chain expires soon	A certificate in the SSL server certificate chain for this system expires soon. If the certificate chain expires, users can't access the system.
Warning	SSL server certificate chain expired	The SSL server certificate chain for this system contains an expired certificate. Users cannot access the system until the certificate chain is replaced.

Severity	Alert Description	Action
Info	Package installation in progress	Your system is currently installing a package that you uploaded. Depending on the contents of the package, this might take a while.
Warning	The certificate for the storage component (<i>storage-id</i>) is about to expire in <i>n</i> days	Renew the storage component certificate.
Info	The storage component (<i>storage-id</i>) is unavailable	Verify that the storage component ID is correct and valid and that the storage component is active.
Info	Update migration in progress. Current state: <i>state</i>	Update migration from a version before v2.3 is in progress. The state of migration is either <code>OLD</code> , <code>MIGRATING</code> , or <code>CLEANUP</code> .

Storage component alerts

Severity	Message	Description
Warning	Available capacity is below <i>n</i> {% bytes} in the system for object data	The free capacity on the HCP for cloud scale system has fallen below the specified threshold (either a percentage of the total or a byte value).
Warning	Certificate for Storage component <i>id</i> is about to expire in <i>n</i> days	The SSL certificate for the storage component <i>id</i> is set to expire in <i>n</i> days. If the certificate expires, HCP for cloud scale will not be able to read from or write to the storage component.
Warning	Storage component <i>id</i> is now inaccessible	The storage component <i>id</i> is in the state <code>INACCESSIBLE</code> . HCP for cloud scale cannot read from or write to the storage component.
Severe	Certificate for Storage component <i>id</i> expired	The SSL certificate for the storage component <i>id</i> has expired. HCP for cloud scale cannot read from or write to the storage component. Install a new certificate.

Severity	Message	Description
Severe	Error communicating with a vault node. Node IP: <i>ip_address</i>	<p>One of the vault nodes can't be reached. If other active nodes are available service continues, but attend to this issue immediately.</p> <p>Examine the vault instance logs to determine the cause of this issue.</p>
Severe	Failed to connect to KMS server	<p>One of the vault nodes can't be reached. If other active nodes are available service continues, but attend to this issue immediately.</p> <p>If ingest is halted, then investigate why the KMS service is failing to run on all nodes. If ingest is still working, the original active node has failed over. Examine the vault instance logs to determine the cause of the failure.</p>
Severe	Failed to connect to KMS server as it is completely sealed	<p>The vault service (Key Management Server service) is completely sealed.</p> <p>Unseal it using the unseal keys you obtained when you turned on encryption.</p>
Severe	Service error: There is a critical issue with the Metadata Gateway database. Shutting down the Metadata Gateway Service.	<p>A Metadata Gateway instance has encountered an issue and shut down. Use the System Management Services function Repair to restart it.</p> <p>If restarting the service doesn't resolve the issue, contact Support.</p>
Severe	Vault node is sealed. Node IP: <i>ip_address</i>	<p>One of the vault nodes is sealed. If other active nodes are available service continues, but attend to this issue immediately.</p> <p>Unseal it using the unseal keys you obtained when you turned on encryption.</p>

Severity	Message	Description
Critical	Available capacity is below n {% bytes} in Storage component id	The free capacity on the named HCP S Series Node storage component has fallen below the specified threshold (either a percentage of the total or a byte value).
Critical	Failed to connect to KMS server	<p>The Key Management System service is not available. Until the service is available, data on encrypted storage components can't be read or written.</p> <p>When KMS service restarts, if there is only one active instance log in to HCP for cloud scale on port 8200 and provide unseal keys to reopen the vault.</p>
Critical	Failed to retrieve capacity usage from Storage component id	<p>System can't retrieve metrics from an HCP S Series Node storage component. Possible reasons are:</p> <ul style="list-style-type: none"> ▪ The storage component is not reachable ▪ The system was upgraded from before v2.1 ▪ The management username or password is not valid ▪ HCP S Series Node version is not supported
Critical	Failed verification for retrieved encryption key for StorageComponent_ID{ $uuid=uuid$ }	<p>The encryption key returned from the Key Management System server doesn't match the key for the storage component $uuid$.</p> <p>Verify that the KMS service is available. If the service is available, verify that you have provided the service with a quorum of unseal keys. If objects on the storage component still can't be read, contact Support.</p>


Severity	Message	Description
Critical	Metadata-Coordination cannot communicate with Sentinel service to get state information	The Sentinel service is not responding to requests for state information. Using the System Management application, immediately review the health of the Metadata-Coordination and Sentinel services and ensure that the Sentinel container has adequate heap size for the configuration of the cluster.

Client certificate alerts

Severity	Message	Description
Warning	Certificate for SubjectDN <i>dn</i> will expire in <i>n</i> days	The SSL certificate for the specified client sync-to or sync-from target (specified by its Distinguished Name) is set to expire in <i>n</i> days. If the certificate expires, HCP for cloud scale will not be able to synchronize to or from the target. You might need to obtain a new client certificate.
Severe	Certificate for SubjectDN <i>dn</i> expired on <i>dd-mmm-yyyy</i>	The SSL certificate for the specified client sync-to or sync-from target (specified by its Distinguished Name) had expired. HCP for cloud scale cannot synchronize to or from the target. You must obtain a new client certificate.

Viewing alerts

Procedure

1. To view alerts, click the user icon () in the top right corner of each Admin App page and then click **Notifications**.

Object Storage Management application instructions

The Object Storage Management application displays alerts about storage components. If an alert is raised the alert icon displays a badge with the number of active alerts. For example:



Click the icon to display a window listing alert text.

Related CLI commands

listAlerts

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /alerts

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Related REST API methods

```
POST /alert/list
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

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 needs 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 make the email notification template. When the system sends an email notification, it replaces the variables in the notification 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.

Creating email notification rules

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Notifications**.
3. Click **Create**.
4. In the **Type** field, select **Email**.
5. Type a name for the notification rule.

6. Under **SMTP settings**, click **Enable** to enable the rule.
7. Configure the SMTP and message settings for the notification rule.
8. Specify a comma-separated list of emails to send notifications to.
9. Click **Create**.

Related CLI commands

createNotificationRule

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /notifications

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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, or SEVERITY.

Creating syslog notification rules

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Notifications**.
3. Click **Create**.
4. In the **Type** field, select **Syslog**.
5. Type a name for the notification rule.
6. Under **Syslog settings**, click **Enable** to enable the rule.
7. Configure the settings for the notification rule.
8. Specify a severity filter for the notification rule.
9. Click **Create**.

Related CLI commands

createNotificationRule

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /notifications

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Logs and diagnostic information

Each service maintains its own set of logs. By default, log files are maintained in the folder `install_path/hcpcs/log` on each instance in the system. During installation, you can configure each service to store its logs in a different (that is, non-default) location.

Log levels

The following table lists the available log levels.



Note: Raising the log level (for example, from WARN to INFO) results in writing more data to the log file, but the file size increases more rapidly. Lowering the log level (for example, from WARN to ERROR) results in the file size increasing more slowly, but results in writing less data to the log file.

Level	Levels included
ALL	FATAL, ERROR, WARN, INFO, DEBUG, TRACE
TRACE	FATAL, ERROR, WARN, INFO, DEBUG, TRACE
DEBUG	FATAL, ERROR, WARN, INFO, DEBUG
INFO	FATAL, ERROR, WARN, INFO
WARN	FATAL, ERROR, WARN (default)
ERROR	FATAL, ERROR
FATAL	FATAL
OFF	None

Log management

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



Caution: Deleting log files can make it more difficult for support personnel to resolve issues you might encounter.

System logs are managed automatically in these ways:

- Retirement: All log files are periodically added to a compressed file and moved to `install_path/hcpcs/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.
 - At the optimum time for a specific service.
- Rotation: 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 the file `exampleService.log.0` grows to 10 MB, it is renamed to `exampleService.log.1` and the system creates a new file named `exampleService.log.0` to write to.
- Removal: When a log file becomes older than 90 days, it is removed. If the system instance's disk space is more than 70% full, log files are deleted when they become older than one day.
- When an optimum number of log files for a specific service is reached, the system can overwrite the oldest file. For example, if a service is limited to 20 log files, when the file `exampleService.log.19` is filled, the system overwrites the file named `exampleService.log.0`.

Retrieving logs and diagnostic information

The tool `log_download` 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_path/hcpcs/bin/log_download
```

For information about running the tool, use this command:

```
install_path/hcpcs/bin/log_download -h
```

**Note:**

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

Default log locations

Default log locations

By default, each service stores its logs on each instance on which the service instance runs, in its own folder at this path:

```
install_path/hcpcs/log
```

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

For information about services, see [HCP for cloud scale services \(on page 78\)](#).

Service	Default log folder name	Contains information about
Admin-App	com.hds.ensemble.plugins.service.adminApp	The System Management application.
Database	com.hds.ensemble.plugins.service.cassandra	<ul style="list-style-type: none"> ▪ System configuration data. ▪ Document fields and values.
Scheduling	com.hds.ensemble.plugins.service.chronos	Workflow task scheduling.
N/A	com.hds.ensemble.plugins.service.containerAction	Created by custom actions run by service plugins.

Service	Default log folder name	Contains information about
Metrics	com.hds.ensemble.plugins.service.elasticsearch	The storage and indexing of: <ul style="list-style-type: none"> ▪ System events ▪ Performance and failure metrics for workflow tasks
Network-Proxy	com.hds.ensemble.plugins.service.haproxy	Network requests between instances.
Message Queue	com.hds.ensemble.plugins.service.kafka	The transmission of data between instances.
Logging	com.hds.ensemble.plugins.service.logstash	The transport of system events and workflow task metrics to the Metrics service.
Service-Deployment	com.hds.ensemble.plugins.service.marathon	The deployment of high-level services across system instances. High-level services are the ones that you can move and configure, not the services grouped under System Services.

Service	Default log folder name	Contains information about
Cluster-Worker	com.hds.ensemble.plugins.service.mesosAgent	The work ordered by the Cluster-Coordination service.
Cluster-Coordination	com.hds.ensemble.plugins.service.mesosMaster	Hardware resource allocation.
Watchdog	com.hds.ensemble.plugins.service.remoteAction	Internal system processes.
Sentinel	com.hds.ensemble.plugins.service.sentinel	The internal system processes.
Watchdog	com.hds.ensemble.plugins.service.watchdog	General diagnostic information.
Synchronization	com.hds.ensemble.plugins.service.zookeeper	The coordination of actions and database activities across instances.
S3-Gateway	com.hitachi.aspen.foundry.service.clientaccess.data	The client access data service.
Data-Lifecycle	com.hitachi.aspen.foundry.service.data-lifecycle.service	The data lifecycle service.
Tracing-Agent	com.hitachi.aspen.foundry.service.jaeger.agent	The tracing agent service.
Tracing-Collector	com.hitachi.aspen.foundry.service.jaeger.collector	The tracing collector service.
Tracing-Query	com.hitachi.aspen.foundry.service.jaeger.query	The tracing query service.

Service	Default log folder name	Contains information about
MAPI-Gateway	com.hitachi.aspen.foundry.service.mapi.gateway	The management API gateway.
Policy-Engine	com.hitachi.aspen.foundry.service.metadata.async.policy.engine	The metadata asynchronous policy engine.
Metadata-Cache	com.hitachi.aspen.foundry.service.metadata.cache	The metadata cache.
Metadata-Coordination	com.hitachi.aspen.foundry.service.metadata.coordination	Metadata coordination.
Metadata-Gateway	com.hitachi.aspen.foundry.service.metadata.gateway	The metadata gateway.
Telemetry-Service	com.hitachi.aspen.foundry.service.metrics.prometheus	Telemetry.
Message-Queue	com.hitachi.aspen.foundry.service.rabbitmq.server	The message broker.
Key-Management-Server	com.hitachi.aspen.foundry.service.vault.vault	The key management server.

Chapter 5: Security

The System Management application supports 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 \(on page 134\)](#).
2. Add one or more groups from your identity providers to the system.
For information, see [Adding groups \(on page 140\)](#).
3. Create a role that contains the system permissions you want to associate with a group of users.
For information, see [Creating roles \(on page 142\)](#).
4. Associate roles with groups.
For information, see [Assigning roles to groups \(on page 141\)](#).

Setting the session timeout limit

You can use the Admin App, CLI commands, or REST API methods 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.

Admin App instructions

Procedure

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

Related CLI commands

`editSecuritySettings`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /security/settings

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Setting the refresh token timeout limit

You can use the Admin App, CLI commands, or REST API methods to set the refresh token timeout limit. The refresh token timeout limit must be greater than the session timeout limit so that if the access token expires, the refresh token will still be active and you can request a new session token. When your refresh token expires, you will need to resubmit your credentials to access your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. On the **Settings** tab, type a number of minutes in the **Refresh Token Timeout** field.
4. Click **Update**.

Related CLI commands

editSecuritySettings

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /security/settings

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Setting the CORS Allowed Origins

You can use the Admin App, CLI commands, or REST API methods to set CORS (cross-origin resource sharing) origins that are allowed on your system. Specifying multiple origins allows you to access restricted resources.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.

2. Click **Security**.
3. On the **Settings** tab, type a list of origins in the **CORS Allowed Origins** field.
4. Click **Update**.

Related CLI commands

editSecuritySettings

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /security/settings

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 \(on page 135\)](#).

Admin App instructions

Procedure

1. Select the **Configuration** window.
2. Click **Security**.
3. On the **Identity Providers** tab, click **Create**.
4. Select and configure an identity provider type. For information, see [Identity provider configuration settings \(on page 135\)](#).
5. Click **Create**.

Related CLI commands

createIdentityProvider

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /security/identityProviders

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Identity provider configuration settings

The following sections describe the configuration settings for each type of identity provider that the system supports.

All types

Security Realm Name: The name by which to identify this identity provider in the system. The name appears as an option in the Security Realm list on Admin App login pages.



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

Active Directory

- **Identity Provider Hostname:** Host name 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)
 - **Use Suffix For Hostname Verification:** When enabled, if the client host name doesn't match the certificate host name, host name verification will instead verify whether the ending of the client host name matches the provided suffix.



WARNING: This option can impact security and should only be enabled if the client host name is expected to differ from the certificate host name.

- **Hostname Suffix:** The suffix used for host name verification if the default host name verification fails.
 - SSL (Secure Sockets Layer)
- Note:** When selecting TLS Security, the setting Use Suffix For Hostname Verification appears.
- **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. The 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 the 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 Admin App and Search App. For example, if you specify a default domain name of `east.example.com`, the user `jdoh@east.example.com` needs to specify only `jdoh` when logging into either app.

LDAP Compatible

- Identity Provider Hostname: Host name 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)
 - Use Suffix For Hostname Verification: When enabled, if the client host name doesn't match the certificate host name, host name verification verifies whether the ending of the client host name matches the provided suffix.



WARNING: This option can impact security and should only be enabled if the client host name is expected to differ from the certificate host name.

- Hostname Suffix: The suffix used for host name verification if the default host name verification fails.
- 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 the system to begin searching 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.

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

OpenLDAP and 389 Directory Server

- **Identity Provider Hostname:** Host name 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)
 - **Use Suffix For Hostname Verification:** When enabled, if the client host name doesn't match the certificate host name, host name verification verifies whether the ending of the client host name matches the provided suffix.



WARNING: This option can impact security and should only be enabled if the client host name is expected to differ from the certificate host name.

- **Hostname Suffix:** The suffix used for host name verification if the default host name verification fails.
 - 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 the system to begin searching 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.

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.

LDAP 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 add a user to an LDAP identity provider, that user cannot access the system for up to four hours, or until the cache is cleared. If you delete a user from an LDAP identity provider, that user will be able to access the system for up to four hours, or until the cache is cleared.

To ensure that a change is reflected immediately, use the `clearCache` command or API.

Related CLI commands

`clearCache`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /security/clearCache

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Viewing identity providers

You can use the Admin App, CLI commands, or REST API methods to view the identity providers that have been added to your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.

2. Click **Security**.
3. Select the **Identity Providers** tab.

Related CLI commands

```
getIdentityProvider
```

```
listIdentityProviders
```

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

```
GET /security/identityProviders/{uuid}
```

```
GET /security/identityProviders
```


You can get help on specific REST API methods for the Admin App at REST API - Admin.

Deleting identity providers

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

Admin App instructions

Procedure

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

Related CLI commands

```
deleteIdentityProvider
```

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

```
DELETE /security/identityProviders/{uuid}
```

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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. After you add 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 \(on page 134\)](#).
- Roles, see [Roles \(on page 142\)](#).

Adding groups

You use the Admin App, CLI commands, or REST API methods to add groups from your identity providers to your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. On the **Groups** tab, click **Create**.
4. Select an identity provider and type a string on which to query the identity provider for groups.
5. Click **Discover Groups**.
6. Click **Continue**.
7. Select one or more roles to associate with the group.
8. Click **Create**.

Related CLI commands

`createGroup`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

`POST /security/groups`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Viewing groups

You use the Admin App, CLI commands, or REST API methods to view all the groups that have been created for your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. Select the **Groups** tab.

Related CLI commands

getGroup

listGroups

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /security/groups/{uuid}

GET /security/groups

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Assigning roles to groups

You use the Admin App, CLI commands, or REST API methods to assign roles to the groups that you've added your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. On the **Groups** tab, select the group you want to edit.
4. On the **Roles** tab, select one or more roles to enable for the group.
5. Click **Update**.

Related CLI commands

editGroup

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /security/groups/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Deleting groups

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

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.

3. Select the **Groups** tab.
4. Click the delete icon () for the group you want to remove.

Related CLI commands

deleteGroup

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

DELETE /security/groups/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 \(on page 141\)](#).

Creating roles

You can use the Admin App, CLI commands, or REST API methods 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 Admin App.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. On the **Roles** tab, click **Create**.
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:

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

- On the **Wildcard** tab, you can enable permissions for multiple categories at the same time. To do this:
 - a. Click **Add Permission**.
 - b. Use the menus to select a category of permissions.
 - c. Leave the last menu set to the wildcard character (*).
6. Click **Create**.
 7. Click **Update**.

Related CLI commands

createRole

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /security/roles

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Viewing roles

You can use the Admin App, CLI commands, or REST API methods to view all the roles that have been created for your system.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. Select the **Roles** tab.

Related CLI commands

getRole

listRoles

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /security/roles/{uuid}

GET /security/roles

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Editing roles

You can use the Admin App, CLI commands, or REST API methods 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 permission `admin:services:read` grants the ability to view services through the Admin App.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. Select the **Roles** tab.
4. Select 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:

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

- On the **Wildcard** tab, you can enable permissions for multiple categories at the same time. To do this:
 - a. Click **Add Permission**.
 - b. Use the menus to select a category of permissions.
 - c. Leave the last menu set to the wildcard character (*).

6. Click **Create**.
7. Click **Update**.

Related CLI commands

`editRole`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT `/security/roles/{uuid}`


You can get help on specific REST API methods for the Admin App at REST API - Admin.

Deleting roles

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

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. Select the **Roles** tab.
4. Click the delete icon () for the role you want to remove.

Related CLI commands

`deleteRole`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

DELETE `/security/roles/{uuid}`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Permissions

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

The `set` permissions override corresponding `get` permissions. That is, if a user has permission to set (configure) a function, the user is also granted permission to get (view) the function.

Chargeback Reporting		
Permission name	Description	Default admin role permission?
chargeback:system:get_report	Generate chargeback report for any bucket	Yes
chargeback:user:get_report	Generate chargeback report for the user's buckets	Yes

Data Service		
Permission name	Description	Default admin role permission?
data:bucket:expirationlifecycle:get	View bucket expiration lifecycle policy configuration	Yes
data:bucket:expirationlifecycle:set	Configure bucket expiration lifecycle policy	Yes
data:bucket:notification:get	View bucket notification configuration	Yes
data:bucket:notification:set	Configure bucket notification	Yes
data:bucket:objectlock:get	View bucket object lock policy configuration	Yes
data:bucket:objectlock:set	Configure bucket object lock policy	Yes
data:bucket:sync:from:set	Create bucket sync-from rules for buckets the user owns or has access to	Yes
data:bucket:sync:get	View bucket sync-from and sync-to rules for buckets the user owns or has access to	Yes
data:bucket:sync:to:set	Create bucket sync-to rules for buckets the user owns or has access to	Yes

License		
Permission name	Description	Default admin role permission?
mapi:license:add	Add licensed feature	Yes
mapi:license:list	List all licensed feature	Yes

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

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

MAPI Storage Component		
Permission name	Description	Default admin role permission?
mapi:storage_component:activate	Activate a storage component	Yes
mapi:storage_component:get_capacity	Get storage component capacity	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		
Permission name	Description	Default admin role permission?
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:certificates:refresh	Refresh SSL certificates	
mapi:system:info	List system information	Yes

MAPI User		
Permission name	Description	Default admin role permission?
mapi:user:list	List existing users	Yes
mapi:user:list_buckets	List user's buckets	Yes
mapi:user:revoke_credentials	Revoke S3 credentials	Yes
mapi:user:revoke_tokens	Revoke OAuth tokens	Yes

S3 Encryption Setting		
Permission name	Description	Default admin role permission?
mapi:s3_encryption:get	Read S3 encryption settings	Yes
mapi:s3_encryption:set	Enable global encryption	Yes
mapi:s3_encryption:unseal	Unseal KMS service	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

Revoking user account credentials

To immediately and completely revoke a user's account credentials, it's best to remove credentials from the identity provider, clear the HCP for cloud scale user cache, revoke OAuth tokens, and revoke S3 credentials, in that order.

You might want to revoke a user's account credentials for a number of reasons:

- The user has left the organization.
- The user is part of a company that has ended a contract with the organization.
- The user had a temporary account that has expired.

Before you begin

To complete these tasks you must have permission to reset a user's password in the identity provider (IdP), such as Active Directory or LDAP, and you must have permission to execute HCP for cloud scale API methods.

If you remove the user account from the IdP, you might not be able to move or delete the user's data and buckets. If you must do that, reset the account password instead, delete or move the data as needed, and then remove or disable the account.



Caution: Execute the steps in this procedure in the order given.

Procedure

1. Prevent the user from signing in through the IdP.
 - Remove or disable the user account in the IdP.
 - Change or remove the user's password.

Removing or disabling the account creates a positive record of revocation but can prevent access to the user's data.

The user can now no longer log in to the system.

2. Clear the HCP for cloud scale user cache:
 - a. Obtain an OAuth token.
 - b. Use the API method `POST security/clearCache`.

The user can now no longer obtain fresh OAuth tokens.

3. Revoke the user's OAuth tokens.
 - a. Obtain an OAuth token (or use the token previously obtained).
 - b. Obtain an XSRF token.
 - c. Use the API method `POST user/list` to obtain the user ID for the user.
The response body contains the user ID.
 - d. Use the API method `POST user/revoke_tokens`, passing as a parameter the user ID previously obtained.
You should receive a 200 OK response with an empty response body.

The user's existing OAuth tokens are now revoked. The user can now no longer obtain S3 credentials.

4. Revoke the user's existing S3 credentials.
(Alternatively, you can generate fresh S3 credentials, but the listed steps create a positive record of revocation.)
 - a. Obtain an OAuth token (or use the token previously obtained).
 - b. Obtain an XSRF token (or use the token previously obtained).
 - c. Use the API method `POST user/list` to obtain the user ID for the user (or use value previously obtained).
 - d. Use the API method `POST user/revoke_credentials`, passing as a parameter the user ID previously obtained.
You should receive a 200 OK response with an empty response body.

Result

The user's account credentials are cleared from the system and revoked.

Next steps

If you now want to remove the user's data and buckets, reset the password in the IdP, obtain an OAuth token using the new password, and then obtain S3 credentials using the OAuth token. When you obtain access, you have two options:

- Remove all objects, versions, and delete markers in each of the user's buckets. When the buckets are empty, remove them.
- Set an expiration period of one second in each of the user's buckets. When the buckets are empty, remove them.




Note: Manual removal of objects and buckets can take a significant amount of time. It's best to write a script to do it. Alternatively, use an S3 client that support bulk deletions.

Changing a local user's account password

Your system includes a single local user account called `admin`, which is available when you first install the system. In addition, your system lets you create a secondary local admin account for reconciliation purposes. You can use the Admin App, CLI commands, or REST API methods to change the passwords for these accounts.

Admin App instructions**Procedure**

1. Click the user icon () in the top right corner of the window.
2. Click **Change Password**.
3. Confirm your current password for the chosen account and specify a new password.
4. Click **Change Password**.

Related CLI commands

`updateCurrentUserPassword`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

`POST /setup/password`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Certificates

The system uses SSL to provide secure incoming and outgoing communication for the product applications.

To enable SSL security, you need a valid server certificate or chain of certificates for incoming communication and a valid client certificate for outgoing communication.

The system comes with its own self-signed SSL system 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 system certificate or replace it by doing one of the following:

- Upload a PKCS12 format certificate chain from a certificate authority (CA).
- Download a certificate signing request (CSR) and use it to obtain, upload, and apply a certificate signed by a CA.
- Generate and apply a new self-signed SSL server certificate. You might do this, for example, if the current certificate is close to expiring and you are waiting to retrieve a new one from your CA.

For outgoing communication, such as to storage components, you need to upload the certificate used by clients. However, you don't need to upload the client certificate if it's valid and trusted by a CA.

Viewing installed certificates

You can use the Admin App, CLI commands, or REST API methods to view information about:

- The system certificate. This is the certificate used to secure communications for your system's applications, CLIs, and REST APIs.
- Client certificates. These are the certificates you upload to HCP for cloud scale that allow the system to communicate securely with clients such as AD and storage components.

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)

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Certificates**.
The **System** tab displays the currently active system certificate.
3. To view the client certificates, select the **Client** tab.

Related CLI commands

listCertificates

getCertificate

getSystemCertificate

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET /certificates

GET /certificates/system

GET /certificates/{subjectDn}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Client certificates

For the HCP for cloud scale system to communicate with Identity Providers (IdPs) and storage components that use SSL-protected communication, the system must accept the certificate from the clients. Your system prompts you to accept a client certificate when testing the connection to an IdP using the System Management application. It's best to use valid SSL certificates trusted by a certificate authority, but you can also upload client certificates manually.

Client certificate considerations

Keep the following in mind when configuring SSL certificates for a client (such as a storage component):

- Do not allow any of the SSL certificates to expire.
- Adhere to the established best practices for setting up SSL certificates.
For information on SSL best practices, see <http://tools.ietf.org/html/rfc5280> and <http://tools.ietf.org/html/rfc6125>.
- When configuring a certificate chain, ensure that all intermediate issuers have the appropriate signing authority permissions so that the entire chain is signed.
- If you regenerate or upload an SSL certificate you must repair (that is, restart) the S3 Gateway and MAPI Gateway services for the change to take effect.
- If you regenerate or upload an SSL certificate for an S3-compatible remote system used for bucket synchronization you must repair (that is, restart) the S3 Gateway and Policy Engine services for the change to take effect.

Retrieving the client certificate for an HCP S Series Node

It's best to use valid SSL certificates from a trusted certificate authority. If an HCP S Series Node storage component uses another kind of certificate, you can retrieve the certificate and upload it manually. This procedure uses the Firefox browser.

Procedure

1. Go to the HCP S Series Node management interface at `https://host:8000/` where *host* is the host name of the management system.
2. Right click and select **Page Info**.
3. Select **Security** and click **View Certificate**.
4. Click the **Details** tab.
5. Click **Export**, select a download location, and click **Save**.

Result

The certificate is retrieved.

Next steps

Upload the client certificate manually into the HCP for cloud scale system.

Uploading client certificates manually

If a client such as a storage component does not have a trusted SSL certificate, you can upload it manually.

Admin App instructions**Procedure**

1. Retrieve the SSL certificate from your client.
2. In the Admin App, click **Configuration**.
3. Click **Certificates**.
4. On the **Client** tab, click **Upload Client Certificate**.
5. Drag the certificate file into the **Upload Certificate** box.
6. Refresh the certificate by doing one of the following:
 - Use the Object Storage Management MAPI method `POST /certificates/refresh`
 - Click **Services**, select and repair the S3 Gateway service, and then select and repair the MAPI Gateway service.

Next steps

You can now use the Object Storage Management application to create the storage component. Use the option HTTPS and the same host name as in the uploaded SSL certificate.

Related CLI commands createCertificate

createCertificate

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /certificates (System Management)

POST /certificates/refresh (Object Storage Management)

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 the certificate by:

- Installing a new PKCS12 certificate (for instructions, see [Installing a certificate you created \(on page 155\)](#)).
- Generating and installing a new self-signed certificate (for instructions, see [Installing a new self-signed certificate \(on page 156\)](#)).
- 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 \(on page 157\)](#)).

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. If you rename the system you need a new certificate.
- When configuring a certificate chain, ensure that all intermediate issuers have the appropriate signing authority permissions so that the entire chain is signed.
- If you regenerate or upload an SSL certificate you must repair (that is, restart) the S3 Gateway and MAPI Gateway services for the change to take effect.
- If encryption is enabled you must also repair the Key Management Server service and unseal the vault.

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 the system, these passwords must be the same.

When you create an SSL server certificate, you can have that certificate signed by a certificate authority (CA). In this case, the CA you use might 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*, which is an ordered list of certificates in which each certificate is trusted by the next.

To preserve the chain of trust among the certificates, you must 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 the CA.



Important: Read and understand the topic [System certificate considerations \(on page 155\)](#) before creating SSL certificates, especially if you are using an in-house CA.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Certificates**.
3. Click **Update System Certificate**.
4. In the box **PKCS12 Password**, type the password for the certificate.
5. Drag the certificate into the box **Upload Certificate Chain**.
6. Click **Continue**.
7. Click **Accept**.
8. Restart (repair) the S3 Gateway.

Related CLI commands

uploadPKCS12Certificate

applyCertificateChanges

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /certificates/system/pkcs12

POST /certificates/system

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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.

**Note:**

- 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.
- When you install the new certificate, if you are using encryption, you must repair the Vault service and then unseal the vault.

Admin App instructions**Procedure**

1. Select **Dashboard > Configuration**.
2. Click **Certificates**.
3. Click **Update System Certificate**.
4. Select the **Self-Signed** window.
5. Click **Continue**.
Your system generates a new self-signed server certificate.
6. Click **Accept**.
Your system installs the new certificate.
7. To continue using the Admin App, log out and then log back in.

Related CLI commands

```
generateSelfSignedCertificate
```

```
applyCertificateChanges
```

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

```
POST /certificates/system/selfsigned
```

```
POST /certificates/system
```

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 Admin App or a third-party tool. When you use the Admin App, the system securely stores the private key needed for installing the returned certificate, so you don't need to save the key yourself.

Verify what information is required with the certificate authority (CA) that you plan to use.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Certificates**.
3. Select the **System** tab.
4. Click **Update System Certificate**.
5. Select the **CSR** window.
6. Choose **Generate a new certificate signing request** and click **Continue**.
7. Fill in the following as needed:

- In the **Common Name (CN)** box, type the cluster name or an asterisk (*) followed by the DNS name of the system (for example, `cluster.company.com` or `*.company.com`).

Common Name (CN) is required.

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

For example: `cluster.company.com *.cluster.company.com`

Subject Alternative Name is required for Chrome browsers.



Note: The System Management application supports a single SAN, so arrange with your CA to include two SANs in the final signed certificate.

8. Click **Generate CSR**.
The page displays the generated certificate request.
9. Copy and paste the request text into a file and send that file to the CA.

Next steps

Continue to [Installing the certificates returned for a system-generated CSR \(on page 159\)](#).

Related CLI commands

`generateCSR`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /certificates/system/csr

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Installing the certificates returned for a system-generated CSR

In response to a CSR, your CA gives you 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.

Admin App instructions**Procedure**

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

Related CLI commands

uploadCSR

applyCertificateChanges

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /certificates/system/csr

POST /certificates/system

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Chapter 6: System management

As an administrator, you play a role in ensuring the continued accessibility and performance of the system. You can use the Admin App, CLI commands, or REST API methods 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.
- Expanding functionality by writing and installing plugins.
- Setting up email notifications.
- Upgrading the system.

Setting host name

After installing your system, you need to configure it with the host name assigned to it in your corporate DNS environment.

The host name must be a fully qualified domain name (FQDN) using lowercase letters.

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.
3. On the **Settings** tab, specify the system or cluster host name in the **Cluster Hostname** field. Type a lowercase ASCII FQDN of as many as 255 characters only from the set a-z, 0-9, hyphen (-), period (.), underscore (_), and tilde (~).
4. Click **Update**.

Changing host name

If you change the system or cluster host name, you must update the system certificate and restart the S3 Gateway and MAPI Gateway services for the change to take effect.

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Security**.

3. On the **Settings** tab, change the system or cluster host name in the **Cluster Hostname** field.
Type a lowercase ASCII FQDN of as many as 255 characters only from the set a-z, 0-9, hyphen (-), period (.), underscore (_), and tilde (~).
4. Click **Update**.

Next steps

After changing the host name, do the following:

1. Update the system certificate. This applies to the default self-signed certificate as well.
For more information, see [Changing the system certificate \(on page 155\)](#).
2. Restart (repair) the S3 Gateway and MAPI Gateway services.
For information about restarting a service, see [Repairing services \(on page 100\)](#).
3. If encryption is enabled, restart (repair) the Key Management Server (KMS) service and unseal the vault.
For information on unsealing the vault, see [Providing unseal keys to KMS service \(on page 41\)](#).

Related CLI commands

`editSecuritySettings`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

`PUT /security/settings`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 if instances fail. 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*.

Services

Each instance runs a configurable set of services, each of which performs a specific function. For example, the Metadata Gateway service stores metadata persistently.

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

Networking

This topic describes the network usage by, 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.
- If the networking environment changes such that the system can no longer function with its current networking configuration, you must reinstall the system.

Cluster host name

The HCP for cloud scale cluster host name is configured during installation. The cluster host name is required because it's needed for access to both the HCP for cloud scale user interface and the S3 API.

Instance IP address requirements

All instance IP addresses must be static, including both internal and external network IP addresses if applicable to the system. If you replace an instance, you can reuse its IP address. By doing so you don't have to change DNS entries and you conserve the address.

Network types

Each of the HCP for cloud scale services can bind to one type of network, either **internal** or **external**, for receiving incoming traffic. If the network infrastructure supports having two networks, you might want to isolate the traffic for most system services to a secured internal network that has limited access. You can then leave the following services on the external network for user access:

- Admin-App
- Message Queue
- Metadata-Cache
- Metadata-Coordination
- Metadata-Gateway
- Policy-Engine
- Metrics
- S3-Gateway

- Tracing-Agent
- Tracing-Collector
- Tracing-Query
- MAPI-Gateway

You can use either a single network type for all services or a mix of both types. To use both types, every instance in the system must be addressable by two IP addresses, one on the internal network and one on the 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 must configure the network environment to ensure that all instances have outgoing access to the external resources you want to use, such as:

- The storage components where the object 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. Port mapping is visible from the Network tab for each service.

Before installing HCP for cloud scale, you can configure services to use different ports, or use the default values shown in the following tables.

The following services must be deployed with their default port values:

- Message Queue
- Metadata Cache
- Tracing Agent
- Tracing Collector
- Tracing Query

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 needs administrative or data access to the system
- Must be accessible from every other instance in the system

Default Port Value	Used by Service	Purpose
80 (S3 HTTP port, if enabled)	S3 Gateway	Object persistence and access
443 (S3 HTTPS port)	S3 Gateway S3 Console application	Object persistence and access Proxied by Network Proxy
8000	Admin App	System Management application GUI
8443 (S3 HTTPS port)	S3 Gateway	Object persistence and access Not proxied by Network Proxy, used by external load balancer
9099	MAPI Gateway	Object Storage Management application GUI

Load balancing

The supported options for load balancing S3 traffic affect performance.

The S3 Gateway service processes S3 traffic and can serve as an SSL termination point. It can listen on port 80, port 443 (the standard SSL port) or port 8443. The Network Proxy service balances the flow of S3 traffic to S3 Gateway instances. The Network Proxy service listens only on port 443. By default, Network Proxy passes S3 SSL traffic through to the S3 Gateway service.

To improve performance, you can configure an external load balancer and bypass Network Proxy. If your load balancer supports SSL termination, you can configure S3 Gateway instances to accept HTTP traffic on port 80.

If you want your load balancer to pass through SSL S3 traffic and your firewall rules permit traffic on port 8443, configure your load balancer to point to port 8443.

If you want your load balancer to pass through SSL S3 traffic but your firewall rules block traffic on port 8443, you can use IP tables to redirect the traffic from port 8443 to port 443.


HCP for cloud scale provides scripts to enable and disable IPtable redirection of S3 traffic. An additional script lists the IP addresses of affected instances. For more information, see [Script to enable S3 traffic redirection \(on page 164\)](#), [Script to disable S3 traffic redirection \(on page 165\)](#), and [Script to list S3 traffic redirection \(on page 166\)](#).

Script to enable S3 traffic redirection

A script is included to enable redirection of S3 traffic from port 443 to port 8443.

The script is written in Python and located in the folder `install_path/product/bin` (for example, `/opt/hcpcs/bin`).

The script redirects S3 traffic from port 443 to port 8443 using the file `iptables`.

 **Note:** The script requires Python 2.

Syntax

```
enable_s3_redirect.py
```

Options and parameters

None

Example

```
$ enable_s3_redirect.py
```

This example can produce the following output:

```
*** PREROUTING chain in NAT table before adding Redirect
Chain PREROUTING (policy ACCEPT 135 packets, 8100 bytes)
num  pkts bytes target    prot opt in     out     source
destination
1     14M  845M DOCKER    all  --  *     *     0.0.0.0/0
0.0.0.0/0          ADDRTYPE match dst-type LOCAL

-----


*** PREROUTING chain in NAT table after adding Redirect
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source
destination
1     14M  845M DOCKER    all  --  *     *     0.0.0.0/0
0.0.0.0/0          ADDRTYPE match dst-type LOCAL
2      0    0 REDIRECT tcp  --  *     *     0.0.0.0/0
0.0.0.0/0          tcp dpt:443 redir ports 8443
```

Script to disable S3 traffic redirection

A script is included to disable redirection of S3 traffic from port 443 to port 8443.

The script is written in Python and located in the folder `install_path/product/bin` (for example, `/opt/hcps/bin`).

The script redirects S3 traffic from port 8443 to port 443 using the file `iptables`.

 **Note:** The script requires Python 2.

Syntax

```
disable_s3_redirect.py
```

Options and parameters

None

Example

```
$ disable_s3_redirect.py
```

This example can produce the following output:

```
*** PREROUTING chain in NAT table before deleting Redirect
Chain PREROUTING (policy ACCEPT 3227 packets, 195K bytes)
num  pkts bytes target    prot opt in     out     source
destination
1     14M  845M DOCKER    all  --  *     *     0.0.0.0/0
0.0.0.0/0          ADDRTYPE match dst-type LOCAL
2      0    0 REDIRECT tcp  --  *     *     0.0.0.0/0
0.0.0.0/0          tcp dpt:443 redir ports 8443

-----

*** PREROUTING chain in NAT table after deleting Redirect
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source
destination
1     14M  845M DOCKER    all  --  *     *     0.0.0.0/0
0.0.0.0/0          ADDRTYPE match dst-type LOCAL
```

Script to list S3 traffic redirection

A script is included to list redirection of S3 traffic.

The script is written in Python and located in the folder *install_path/product/bin* (for example, */opt/hcpcs/bin*).

The script lists the instances running S3 Gateway system instances and writes the output to the file *s3NodeIPs.txt*.

Syntax

```
list_s3_node_ips.py username password
```

Options and parameters***username***

User name of an HCP for cloud scale user with administrative privileges.

password

Password for the administrative user.

Example

```
$ list_s3_node_ips.py username password
```

This example can produce the following output:

```
INSTALL_DIR: /opt/hcpcs
ADMIN_CLI: /opt/hcpcs/cli/admin/admincli
IPs of S3 Gateway nodes:
172.10.24.195
172.10.24.196
172.10.24.197
172.10.24.198
172.10.24.199
Output file is located at /opt/hcpcs/s3NodeIPs.txt
```

Handling network changes

After 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 these things 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.



Note: You can reuse the IP addresses of retired nodes for new nodes.

For each instance whose IP address you need to change:

Procedure

1. Move all services off of the instance. Distribute those services among all the other instances.
2. On the instance from step 1, stop the script `run` using whatever tool or process you used to run it.
For example, with `systemd`, run: `systemctl stop hcpcs.service`
3. Remove the instance from the system.
4. Delete the installation folder from the instance.
5. Add the instance back to the system.

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.

Procedure

1. If the Admin App is accessible, back up your system components by exporting a package.
2. On each instance in the system:
 - a. Navigate to the installation folder.
 - 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 Admin App and use the wizard to set up the system.
4. After the system has been set up, upload your package.

Volumes

Volumes are properties of services that specify where and how a service stores its data.

You can use volumes to configure 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.

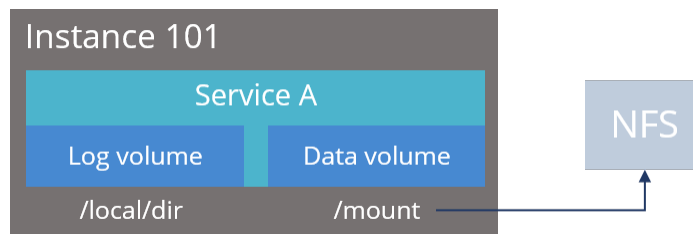
Volumes can also allow services to store different types of data in different locations. For example, a service might use two separate volumes, one for storing its logs and the other for storing all other data.



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 folder 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.

After 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 service or job to use them.



Note: The built-in services don't support adding user-managed volumes.

Volume drivers

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

Because services 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 about volume drivers you can use, see the applicable Docker or third-party developer's documentation.

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

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

Viewing volumes

The System Management application 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 free 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 service volumes

Procedure

1. Select **Dashboard > Services**.
2. Click the service you want.
3. Click 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.

About master and worker instances

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

- Admin-App service
- Cluster-Coordination service
- Synchronization service
- Service-Deployment service

Non-master instances are called workers. Workers can run any services except for those listed previously.

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.

Single instance

A single-instance system is useful for testing and demonstration purposes. A single-instance system 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 the single instance.

Therefore, a single-instance system is unsuitable for use in a production environment.

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.
- 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 HCP for cloud scale services. Single-instance systems have one master instance. Multi-instance systems have at least three.

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

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.

Requirements for running system instances

This section lists the hardware and operating system requirements for running system instances.

Hardware requirements

To install HCP for cloud scale 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 hardware resources required for each instance of an HCP for cloud scale system for a minimum qualified configuration and a standard qualified configuration.

Resource	Minimum configuration	Standard configuration
CPU	Single CPU, 10-core	Dual CPU, 20+-core
RAM	128 GB	256 GB
Available disk space	(4) 1.92 TB SSD, RAID10	(8) 1.92 TB SSD, RAID10
Network interface controller (NIC)	(2) 10 Gb Ethernet NICs	(2) 25 Gb Ethernet NICs or (4) 10 GB Ethernet NICs



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

Software requirements

The following table shows the minimum requirements and best-practice software configurations for each instance in an HCP for cloud scale system.

Resource	Minimum	Best
IP addresses	(1) static	(2) static
Firewall Port Access	Port 443 for SSL traffic Port 8000 for System Management App GUI Port 8888 for Content Search App GUI	Same
Network Time	IP address of time service (NTP)	Same

Operating system and Docker minimum requirements

Each server or virtual machine you provide must have the following:

- 64-bit Linux distribution
- Docker version installed: Docker Community Edition 18.09.0 or later
- IP and DNS addresses configured

Additionally, you should install all relevant patches on the operating system and perform appropriate security hardening tasks.

! **Important:** The system cannot run with Docker versions before 1.13.1.

To execute scripts provided with the product on RHEL, you should install Python.

Operating system and Docker qualified versions

This table shows the operating system, Docker, and SELinux configurations with which the HCP for cloud scale system has been qualified.

! **Important:** An issue in Docker Enterprise Edition 19.03.15 and resolved in 20.10.5 prevented HCP for cloud scale deployment. Do not install any version of Docker Enterprise Edition above 19.03.14 and below 20.10.5.

Operating system	Docker version	Docker storage configuration	SELinux setting
Red Hat Enterprise Linux 8.4	Docker Community Edition 19.03.12 or later	overlay2	Enforcing

If you are installing on Amazon Linux, before deployment, edit the file `/etc/security/limits.conf` on every node to add the following two lines:

```
* hard nofile 65535
* soft nofile 65535
```

Docker considerations

The Docker installation folder on each instance must have at least 20 GB available for storing the Docker images.

Make sure that the Docker storage driver is configured correctly on each instance before installing the product. After you install the product, to change the Docker storage driver you must reinstall the product. To view the current Docker storage driver on an instance, run:

```
docker info
```

Core dumps can fill a host's file system, which can result in host or container instability. Also, if your system uses the data at rest encryption (DARE) feature, encryption keys are written to the dump file. It's best to disable core dumps.

To enable SELinux on the system instances, you need to use a Docker storage driver that SELinux supports. 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 GB of Docker metadata storage space available on each instance. The product needs 20 GB to install successfully and an additional 20 GB 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 product might not have enough space to run.

SELinux considerations

- You should decide whether you want to run SELinux on system instances and enable or disable it before installing additional software on the instance.

Enabling or disabling SELinux on an instance needs a restart of the instance.

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

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

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

Supported browsers

The HCP for cloud scale web applications support these web browsers:

- Google Chrome latest
- Mozilla Firefox latest

Time source

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 might 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 might also be improved by adding additional CPU, RAM, or disks to the instances you already have.

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 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 Admin App.

- Ensure that the new instances you are adding meet the minimum hardware, OS, and networking requirements. For information, see [Requirements for running system instances \(on page 172\)](#).
- 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.

Install Docker on each server or virtual machine

On each server or virtual machine that is to be an HCP for cloud scale instance:

Procedure

1. In a terminal window, verify whether Docker 1.13.1 or later is installed:
`docker --version`
2. If Docker is not installed or if you have a version before 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.

Configure Docker on each server or virtual machine

Before installing the product, configure Docker with settings suitable for your environment. For guidance on configuring and running Docker, see the applicable Docker documentation.

Procedure

1. Ensure that the Docker installation folder on each instance has at least 20 GB available for storing the product Docker images.
2. Ensure that the Docker storage driver is configured correctly on each instance. After installation, changing the Docker storage driver needs reinstallation of the product.
To view the current Docker storage driver on an instance, run: `docker info`.
3. To enable SELinux on the system instances, use a Docker storage driver that SELinux supports.
The storage drivers that SELinux supports differ depending on the Linux distribution you're using. For more information, see the Docker documentation.
4. If you are using the Docker `devicemapper` storage driver, ensure that there's at least 40 GB of Docker metadata storage space available on each instance.
The product needs 20 GB to install successfully and an additional 20 GB to successfully update to a later version.
To view Docker metadata storage usage on an instance, run: `docker info`

Next steps

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

(Optional) 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 might 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.

Configure maximum map count setting

You need to configure a value in the file `sysctl.conf`.

Procedure

1. On each server or virtual machine that is to be a system instance, open the file `/etc/sysctl.conf`.
2. Append this line: `vm.max_map_count = 262144`
If the line already exists, ensure that the value is greater than or equal to 262144.
3. Save and close the file.

Optional: Enable or disable SELinux on each server or virtual machine

You should decide whether you want to run SELinux on system instances before installation.

Procedure

1. Enable or disable SELinux on each instance.
2. Restart the instance.

Configure the firewall rules on each server or virtual machine

Before you begin

Determine the port values currently used by your system. To do this, on any instance, view the file `install_path/config/network.config`.

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

Procedure

1. Edit the firewall rules to allow communication over all network ports that you want your system to use. You do this using a firewall management tool such as `firewalld`.
2. Restart the server or virtual machine.

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 <http://support.ntp.org>.

Run Docker on each server or virtual machine

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`:

Procedure

1. Verify that Docker is running:

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

```
sudo systemctl start docker
```
3. (Optional) Configure the Docker service to start automatically when you restart the server or virtual machine:

```
sudo systemctl enable docker
```

Unpack the installation package

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

Procedure

1. Download the installation package `hcpcs-version_number.tgz` and the MD5 checksum file `hcpcs-version_number.tgz.md5` and store them in a folder on the server or virtual machine.

2. Verify the integrity of the installation package. For example:

```
md5sum -c hcpcs-version_number.tgz.md5
```

 If the package integrity is verified, the command displays `OK`.
3. In the largest disk partition on the server or virtual machine, create a folder named `install_path/hcpcs`. For example:

```
mkdir /opt/hcpcs
```
4. Move the installation package from the folder where you stored it to `install_path/hcpcs`. For example:

```
mv hcpcs-version_number.tgz /opt/hcpcs/hcpcs-version_number.tgz
```
5. Navigate to the installation folder. For example:

```
cd /opt/hcpcs
```
6. Unpack the installation package. For example:

```
tar -zxf hcpcs-version_number.tgz
```

 A number of directories are created within the installation folder.

**Note:**

If you encounter problems unpacking the installation file (for example, the error message "tar: This does not look like a tar archive"), the file might have been packed multiple times during download. Use the following commands to fully extract the file:

```
$ gunzip hcpcs-version_number.tgz
$ mv hcpcs-version_number.tar hcpcs-version_number.tgz
$ tar -zxf hcpcs-version_number.tgz
```

7. Run the installation script `install`:

```
./install
```

**Note:**

- Don't change directories after running the installation script. The following tasks are performed in your current folder.
- The installation script can be run only one time on each instance. You cannot rerun this script to try to repair or upgrade a system instance.

Set up networking

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

Run the setup script on each server or virtual machine

Before you begin



Note:

- When installing a multi-instance system, make sure you specify the same list of master instance IP addresses on every instance that you are installing.
- When entering IP address lists, do not separate IP addresses with spaces. For example, the following is correct:

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

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

Procedure

1. Run the script `setup` with the applicable options:

Option	Description
-i	The external network IP address for the instance on which you're running the script.
-I	The internal network IP address for the instance on which you're running the script.
-m	Comma-separated list of external network IP addresses of each master instance.
-M	Comma-separated list of internal network IP addresses of each master instance.

Use the following table to determine which options to use:

Number of instances in the system	Network type usage	Options to use
Multiple	Single network type for all services	Either: -i and -m or -I and -M
Multiple	Internal for some services, external for others	All of these: -i, -I, -m, -M
Single	Single network type for all services	Either -i or -I
Single	Internal for some services, external for others	Both -i and -I

Result

Note: If the terminal displays Docker errors when you run the `setup` script, ensure that Docker is running.

Example

The following example sets up a single-instance system that uses only one network type for all services:

```
sudo install_path/hcpcs/bin/setup -i 192.0.2.4
```

To set up a multi-instance system that uses both internal and external networks, type the command in this format:

```
sudo install_path/hcpcs/bin/setup -i external_instance_ip -I
internal_instance_ip -m external_master_ips_list -M
internal_master_ips_list
```

For example:

```
sudo install_path/hcpcs/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
```

The following table shows sample commands 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 contains three master instances and one worker instance and uses both internal and external networks.

Instance internal IP	Instance external IP	Master or worker	Command
192.0.2.1	10.236.1.1	Master	<code>sudo install_path/hcpcs/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</code>
192.0.2.2	10.236.1.2	Master	<code>sudo install_path/hcpcs/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</code>
192.0.2.3	10.236.1.3	Master	<code>sudo install_path/hcpcs/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</code>
192.0.2.4	10.236.1.4	Worker	<code>sudo install_path/hcpcs/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</code>

Start the application on each server or virtual machine

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

Procedure

1. Start the application script `run` 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 if a server restarts or there is other availability event.

Result

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

Example

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

- You can run the script in the foreground:

```
sudo install_path/product/bin/run
```

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

- You can run the script as a service using `systemd`:

1. Copy the product `.service` file to the appropriate location for your OS. For example:

```
cp install_path/product/bin/product.service /etc/systemd/system
```

2. Enable and start the `product.service` service:

```
sudo systemctl enable product.service
sudo systemctl start product.service
```

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.

Also, depending on how your jobs are configured, jobs might not run on the new instances that you've added. You need to manually configure jobs to run on the instances.

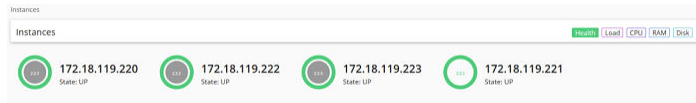
Viewing instances

You can use the Admin App, CLI, and REST API to view a list of all instances in the system.


Viewing all instances

To view all instances, in the Admin App, click 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	<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 best-practice service unit limit for the instance.</p> <p>An instance with a higher number of service units is likely to be more heavily used 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 best-practice service unit limit.</p> <p>The Instances page displays a red bar for instances running more than the best-practice service unit limit.</p> <p><i>Service Units</i>  138 of 100 recommended Service Units</p>
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 Admin App:

Procedure

1. Click **Dashboard > Instances**.
2. Select 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:
 - **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 \(on page 107\)](#).

- **Balancing:** The service is running normally, but performing background maintenance.
- **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 commands

`getInstance`

`listInstances`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

`GET /instances`

GET /instances/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Removing instances

You 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.

(Optional) Shut down the instance you want to remove

If the instance has already shut down because of a failure, the instance is in the Down state. Your system automatically tries 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 the next step ([Remove the shut down instance from the system \(on page 185\)](#)).

If the instance that you want to remove is in the Up state, you need to shut the instance 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.



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.
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-folder>/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 folder:

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

Remove the shut-down instance from the system

Admin App instructions

Procedure

1. Select **Dashboard > Instances**.
2. Click the instance you want to remove.
3. Click **Remove Instance**.

Related CLI commands

deleteInstance

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

DELETE /instances/{uuid}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Replacing a failed instance

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

Procedure

1. In the Admin App, 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.
3. Remove the failed instance from the system.



WARNING: If the failed instance was a master, after you remove the instance, 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.



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 three 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.

Viewing installed plugins

Use the Admin App, CLI commands, or REST API methods 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.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Plugins**.

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

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

Related CLI commands

`listPlugins`

Related REST API methods

`GET /plugins`

You can get help on specific REST API methods for the Admin App at [REST API - Admin](#).

Upgrading plugin bundles

To upgrade plugins, 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.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Plugins**.
3. Click **Upload Bundle**.
4. In the **Upload Plugins** window, drag and drop the new version of the plugin bundle.
5. In the list of plugin bundles, click the row for the plugin bundle version that you want.
If the bundle you uploaded isn't listed, click **Reload Plugins**.
6. Click **Set Active**.

Related CLI commands

uploadPlugin

setPluginBundleActive

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /plugins/upload

POST /plugins/bundles/{name}/{bundleVersion}/active

You can get help on specific REST API methods for the Admin App at [REST API - Admin](#).

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.

When you change the active version of a plugin bundle, any workflow tasks that contain connectors and stages that use the bundle immediately begin using the new active version.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Plugins**.
3. Click the row for the plugin bundle version that you want.

4. Click **Set Active**.

Related CLI commands

setPluginBundleActive

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /plugins/bundles/{name}/{bundleVersion}/active

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Deleting plugin bundles


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



Note: 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.

Admin App instructions

Procedure

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

Related CLI commands

deletePluginBundle

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

DELETE /plugins/bundles/{name}/{bundleVersion}

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 the event 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 the event 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.

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.
2. Click **Packages**.
3. Click **Export**.
4. Under **Customize Package Description**, give your package a name and, optionally, a 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 Admin App prompts you to add those missing components to the package.

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

Related CLI commands

`buildPackage`

`downloadPackage`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST `/package/build`

POST `/package/download`

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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.

**Note:**

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

Admin App instructions**Procedure**

1. Select **Dashboard > Configuration**.
2. Click **Packages**.
3. Click **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** window.
 - If the package you want to import is stored in one of your data sources, click the **Click to Upload** window and then browse for the package file.
5. Under **Package Description**, review the description and then click **Continue**.
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 then click **Install Package**.
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. When the task has completed and all package components have been installed, clicking **Complete Install** deletes the package from the system.

Related CLI commands

uploadPackage

loadPackage: loads a package from a data connection

installPackage

getPackageStatus

deletePackage

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

POST /package (Uploads a package)

POST /package/load (Loads a package from a data connection)

POST /package/install

GET /package (Gets the status of the imported package)

DELETE /package

You can get help on specific REST API methods for the Admin App at REST API - Admin.

Setting a login welcome message

You can use the Admin App, CLI commands, or REST API methods to set a welcome message for the Admin App. The message appears on the app's login page.

Admin App instructions

Procedure

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

Related CLI commands

editSecuritySettings

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

PUT /security/settings

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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 best-practice number of instances.
- 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.

During an update

- 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 Admin App 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.

Verifying 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 might need to manually configure those services to run on the system instances.

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 might not be resolved by restarting the update.

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

New services and components added during an update

A system update might 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.

Applying a system update

Admin App instructions

Procedure

1. Select **Dashboard > Configuration**.

2. Click **Update**.
3. Click the **Install** tab.
4. Click and drag the file into the **Upload** window.
The update file is uploaded and the system verifies that the file is valid. This might take several minutes.
5. On the **Update** page, click **View** in the **Update Status** window.
The **Verify & Apply Update** page displays information about the contents of the update.
6. To start the update, click **Apply Update**.

Result

The system verifies that it 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 commands

```
getUpdateStatus
installUpdate
deleteUpdate
loadUpdate
uploadUpdate
```

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

```
GET /update
POST /update/install
DELETE /update/package
POST /update/package
POST /update/package/load — (Retrieves update package from a data connection)
```

You can get help on specific REST API methods for the Admin App at REST API - Admin.

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.

Admin App instructions

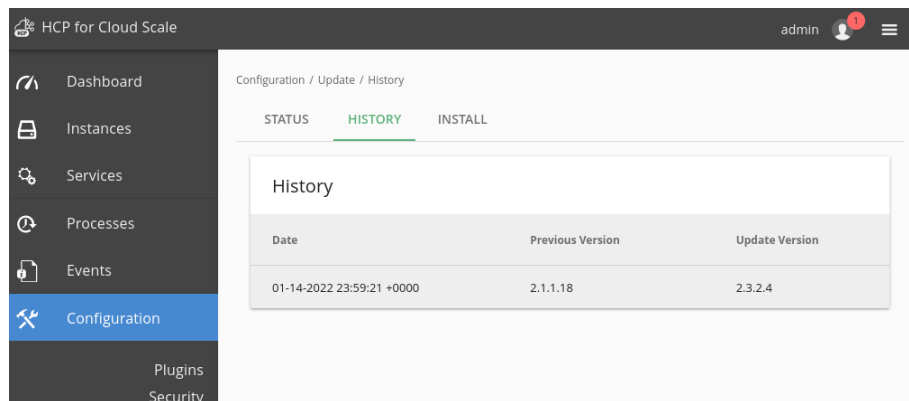
Procedure

1. Select **Dashboard > Configuration**.
2. Click **Update**.

Result

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

For example:

**Related CLI commands**

`getUpdateHistory`

For information on running CLI commands, see [CLI reference \(on page 200\)](#).

Related REST API methods

GET `/update/history`

You can get help on specific REST API methods for the Admin App at [REST API - Admin](#).

Removing the system

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

Procedure

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-folder>/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 folder:

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

Chapter 7: Best practices

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

Best practices for system sizing and scaling

Using master instances

These best practices apply specifically to the scaling of product services across master instances (nodes). If you do not plan to scale product services across master instances then you can skip this section.

Running product services on master instances can maximize the use of all physical resources in the site. However, since master instances are critical to the orchestration and management of HCP for cloud scale, it is important to avoid any impact on them. Here are best practices for master instances.

First, do not over-provision master instances:

- CPU/RAM - HCP for cloud scale can partition the usage of these resources, but an instance can still exhaust these resources.
- Network - HCP for cloud scale cannot control usage of network resources.
- Disk space - HCP for cloud scale cannot control usage of disk space. One service might consume all the free space and thus affect other services. This is most likely to happen with the Metadata Gateway and Message Queue services, but all stateful services consume disk space.

Second, consider the impact of running product services on master instances:

- When scaling product services across master instances, you must take into account the resources (CPU, memory, etc.) already consumed there by system services.
- If product services run into an issue, resolution could require restarting services (including system services) or even restarting the host (operating system). This can affect system services and thus the operation of the entire site.
- When designing a solution, consider hardware failures and other causes of outages. If you plan to use the master instances for product services, ensure there are enough instances (nodes) that the cluster will still be able to manage both existing objects and the anticipated throughput (addition and deletion) of objects if one or even two hosts fail.

Resource impact of stateful and stateless services

You should consider the differing impacts of stateful and stateless services on resource consumption when planning how to scale system and worker service instances across your HCP for cloud scale system.

A stateful service is a service that permanently saves data to disk. Any data that is stored by a stateful service is critical for the operation of the system and the integrity of customer data, so all data that is stored by a stateful service is protected by having three identical copies. Each copy of the data resides on a different instance (node). Each instance of a stateful service runs on a different physical instance.

Stateful services are also persistent. A persistent service runs on a specific instance (node) that you designate. If an instance of a persistent service fails, HCP for cloud scale restarts the instance on the same node. (In other words, HCP for cloud scale does not automatically bring up a new stateful service instance on a different node.)

Because every stateful service is also persistent, a failure or even a planned outage of an instance (node) affects the copy of the data of all stateful service instances that had been running on this instance (node). For more information refer to [Service failure recovery \(on page 28\)](#) and [Scaling Metadata Gateway instances \(on page 99\)](#).

Also, stateful services typically require more computing power to process, store, and read the data on disk securely, efficiently, and with high performance.

A stateless service is a service that does not save data to disk. Stateless services are usually also floating. A floating service can run on any node assigned to a pool of instances associated with the floating service. If an instance of a floating service fails, HCP for cloud scale restarts the instance on any node in the instance pool. Therefore, stateless floating service instances have less resource impact, and are typically easier to manage and recover, than stateful persistent service instances.

The impact of stateful and stateless services is as follows:

- The number of floating/stateless services affects the speed of operations. If there are too few of them operations slow down.
- The number of persistent/stateful services affects both the speed of operations and availability. If there are too few of them operations slow down and can fail entirely.
- Not every stateful service is critical to application-facing operations (for example, S3 operations).
- Not every stateful service is resource intensive.
- Not every stateless service is easy and lightweight in terms of resource usage.

The following services can be resource intensive and should be scaled across worker and master instances carefully:

- Data Lifecycle
- Metadata Gateway
- Message Queue
- Policy Engine
- S3 Gateway

Sizing and scaling models

Consider the following use cases:

- Ring buffer - Data has a short life cycle (a few weeks) and will be deleted after a certain period of time
- Synchronization - Buckets are configured to synchronize (mirror) data out to or in from external services
- Performance sensitive - The planned usage must achieve a specific level of performance (an SLA)

If your planned usage of the HCP for cloud scale system matches one of these use cases it's best to size and scale it as follows:

- The minimum cluster size is six instances (nodes).
- With fewer than 12 instances (nodes), do not scale resource-intensive services across more than one master node.
- With 12 or more instances (nodes), do not scale resource-intensive services across any master nodes.

If, however, your planned usage of the HCP for cloud scale system does not match any of these use cases it's best to size and scale it as follows:

- The minimum cluster size is four instances (nodes).
- With four or five instances (nodes), do not scale resource-intensive services across more than two master nodes.
- With six or seven instances (nodes), do not scale resource-intensive services across more than one master node.
- With eight or more instances (nodes), do not scale resource-intensive services across any master nodes.

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.

Chapter 8: Reference

This section contains information about troubleshooting, information on using the command-line interface, and information on using the REST API.

Troubleshooting

System event issues

For information on viewing system events, see [System events \(on page 109\)](#).

Issue	Description/Resolution
<p>The event log contains instances of event id 6007 with this message:</p> <pre>Service <i>service-name</i> health check against HTTP /health <i>port</i> 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 verification on the specified service.</p> <p>You can ignore this event. The additional health verification is not required and does not affect the other health verifications used to display the service status on the Monitoring > Services page in the System Management application.</p>
<p>The MAPI Gateway service becomes unresponsive.</p>	<p>If you're working with a storage component that is configured with multiple retries and long timeouts, and if the endpoint for the storage component is unreachable, and if as a result you send multiple verification or activation requests to the endpoint, the MAPI Gateway service can become unresponsive.</p> <p>If the MAPI Gateway service becomes unresponsive, use the System Management application Services function Repair on it.</p>

Issue	Description/Resolution
Users can't write to storage components, objects read from storage components are encrypted, and you receive the critical alert <code>Failed to connect to KMS server.</code>	The Key Management System service is down. The service should restart automatically. When it restarts, log in to HCP for Cloud Scale on port 8200 and provide a quorum of unseal keys.

CLI reference

An administrative CLI is provided for system management. This interface lets you perform all tasks relating to system setup and configuration. Any administrative activity that you can perform in the Admin App or REST API can be performed through the CLI. You can avoid using an internet connection or a browser.

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

Accessing the CLI tools on a system instance

You can access the CLI tools from any instance

Procedure

1. Log in directly to or use SSH to log in to a system instance.
2. Navigate to the CLI tool:
`cd <installation-folder>/cli/`
3. Navigate to the folder for the CLI tool you want. For example:
`cd admin`

Installing CLI tools on your computer

You can install your system's CLI tools on a Linux computer.

To install the CLI tools, 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.

Procedure

1. Store the .tgz file in a folder on your computer.
2. Unpack the file:
`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 \(on page 202\)](#).

```
--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 `--help` option, displays information about commands in the specified category.

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

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

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

Returns the JSON-formatted schema for the specified request model. See [Viewing request models \(on page 202\)](#).

```
-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 \(on page 134\)](#).

```
-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 need 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 needs.

For example, this help command output indicates that a command to update a service needs 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
- Verify 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 folder.

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	Meaning	Description
400	Bad Request	The request body contains one or more of these: <ul style="list-style-type: none"> ▪ An entry that isn't valid ▪ A value for an entry that isn't valid ▪ Incorrectly formatted JSON If the request includes a UUID, the UUID might be incorrectly formatted.
401	Unauthorized	The provided credentials are incorrect.
403	Forbidden	You do not have permission to perform the request.
404	No 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	Internal Server Error	The system experienced an error.
599	Network Connection Timeout Error	The CLI request timed out while trying 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.

Status 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 includes a RESTful API that you can use for writing applications that manage the system. Anything you can do in either the Object Storage Management application or the System Management application can also be performed using their REST APIs.

Getting started with the management APIs

HCP for cloud scale includes RESTful HTTP management application programming interfaces (APIs) for the Object Storage Management application and the System Management application. These APIs are separate and use separate ports.

System Management

You can execute all functions supported in the System Management application using MAPI methods. The System Management methods are served by the Admin service from any HCP for cloud scale node.

All URLs for the System Management MAPI methods have the following base, or root, uniform resource identifier (URI):

```
https://hcpcs_cluster:8000/api/admin/
```

The System Management MAPI is described in Swagger, available from the System Management user interface. Those methods not described in this document.

Object Storage Management

You can execute all functions supported in the Object Storage Management application and the S3 Console application using MAPI methods. The Object Storage Management management API (MAPI) supports management of the following:

- Storage components and Amazon Simple Storage Service (Amazon S3) settings
- Encryption of storage components
- Administrative resources such as serial numbers and system events

- User resources such as S3 user credentials and OAuth tokens
- Public information such as available public ports

The Object Storage Management MAPI methods are served by the MAPI Gateway service from any HCP for cloud scale node.

All URLs for the Object Storage Management MAPI methods have the following base, or root, uniform resource identifier (URI):

```
https://hcpcs_cluster:9099/mapi/v1/
```

The Object Storage Management MAPI is described in this document. It is also described in Swagger, available from the Object Storage Management user interface.

Input and output formats

The API accepts and returns JSON.

The REST API accepts and returns JavaScript Object Notation (JSON). It does not support HTTPS 1.0 requests; all HTTPS 1.0 requests are denied. When the body of the request has contents, the MAPI accepts and returns JSON; when the body is empty, JSON format is unnecessary.

Access and authentication

To use the Object Storage Management or System Management MAPIs, you need a user account that has permission to perform the actions you want.

Requesting an access token

After you have a user account, you must request an authentication token from the system. To do this, you send an HTTP POST request to the method `/auth/oauth`.

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:8000/auth/oauth/ \
-d grant_type=password \
-d username=user1 \
-d password=password1 \
-d scope=* \
-d client_secret=my-secret \
-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 the token. For example:

```
{
  "access_token": "eyJr287bjle..."
  "expires_in": 7200
}
```

**Note:**

- To get a list of security realms for the system, send an HTTP GET request to the method `/setup`. For example, to do this with cURL:

```
curl -k -X GET --header 'Accept: application/json' \
'https://mysystem.example.com:8000/api/admin/setup'
```

- 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 the access token

You must specify the access token as part of all REST API requests that you make. You do this by submitting an Authorization header along with the request. Here's an example that uses cURL:

```
curl -X GET --header "Accept:application/json" \
https://mysystem.example.com:8000/api/admin/instances \
--header "Authorization: Bearer eyJr287bjle..."
```

Changing a password

You can use the MAPI to change the system's password using the following cURL commands, where `$1=server_name`, `$2=current_password`, and `$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=client-secret -d client_id=client-id \
-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.mysystem.com:8000/api/admin/setup/password \
-d '{"password": "'"$3"'"}'
```

Requesting and submitting an XSRF token

To protect against cross-site-request forgery, the Object Storage Management MAPI requires an XSRF token in all requests.

The Object Storage Management MAPI requires you to pass the XSRF token both in the request header and as a cookie within the request.



Note: The XSRF token has a limited lifetime, so it's best to obtain a fresh token before issuing every group of requests.

To obtain the token and pass it as part of a request:

Procedure

1. Obtain the token.

The following command string obtains the token, locates it within the response, strips out extraneous text, and stores it as a variable:

```
xsrftoken=$(curl -skc - https://hcpcs_cluster.com:9099/ \
-H "user-agent: Mozilla" | grep XSRF-TOKEN | cut -d$'\t' -f 7)
```

2. Include the token both in the request header and as a cookie within a MAPI request. The following command string uses an OAuth token defined as the variable `$token` in a MAPI request to list the user `maria`:

```
curl -s -k -H "X-XSRF-TOKEN:${xsrftoken}" \
-H "Authorization: Bearer $token" \
-b "XSRF-TOKEN=${xsrftoken}" -H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{"nameFilter": "maria"}' \
-X POST https://hcpcs_cluster.com:9099/mapi/v1/user/list
```

Viewing and using MAPI methods

Your system includes web-based documentation pages where you can view the MAPI methods for both the Object Storage Management and System Management applications, including the request bodies, request URLs, response bodies, and return codes for each. You can also use these pages to run each MAPI method.

You can use the MAPI documentation pages to experiment with the MAPI. Any requests you submit on the page take effect on the system.



Note: If you specify UUIDs when creating resources, the UUIDs are ignored.

To use the MAPI page to run a method:

Procedure

1. In either the Object Storage Management App or the System Management App, select the user profile icon, in the upper right portion of the page.
2. Select:
 - In the Object Storage Management App, select **REST API**.
 - In the System Management App, select **REST API - Admin**.

A Swagger page opens for the selected MAPI.
3. Expand the category containing the method you want.
4. Select the row for the method you want.
5. To use an Object Storage Management method, enter the XSRF token in the field **X-XSRF-TOKEN Header**.
6. If the method you want needs a UUID:
 - a. Select the row for the `GET` method for the resource type that you want.

- b. Click **Try it out**.
 - c. In the JSON response body, copy the value in the field `uuid` for the resource that you want.
7. If the method you want needs a request body:
 - a. In the section **Parameters**, under **Model Schema**, click inside the JSON text box. The JSON text is added to the field **Value**.
 - b. Edit the JSON in the field **Value** as needed.



Note: Some methods might need other information in addition to or instead of UUIDs or JSON-formatted text. Some methods need particular string values or need you to browse for and select a file to upload.

8. Click **Execute**.
The method is executed and the results appear in the section **Responses**.

Including XSRF tokens in MAPI Swagger requests

Swagger does not automatically populate the X-XSRF-TOKEN header when executing Object Storage Management MAPI requests.

To obtain the token within a browser and include it in a MAPI request through Swagger:

Procedure

1. From the user profile icon on the top right of an Object Storage Management window, select **REST API**.
A Swagger page opens in a new tab.
2. Locate and copy the value of the cookie XSRF-TOKEN:
 - In Chrome, from the **Customize** menu (in the upper right corner), select **More tools > Developer tools**. From the **Developer Tools** window select **Storage > Cookies**. Select and copy the value of the cookie XSRF-TOKEN.
 - In Firefox, from the **Open** menu (in the upper right corner), select **Web Developer > Storage Inspector**. From the **Developer Tools** window select **Cookies**. Select and copy the value of the cookie XSRF-TOKEN.
3. Select the management API you want to execute and then click **Try it out**.
4. In the section **X-XSRF-TOKEN**, paste the value of the cookie into the field **X-XSRF-TOKEN Header**.

Result

You can now execute the method.

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.

Status code	Meaning	Description
400	Bad Request	The request body contains one or more of these: <ul style="list-style-type: none"> ▪ An entry that's not valid ▪ A value for an entry that's not valid ▪ JSON code that's not validly formatted If the request includes a UUID, the UUID might not be validly formatted.
403	Forbidden	You do not have permission to perform the request.
404	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)	Internal 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.

Status 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
2535 Augustine Drive
Santa Clara, CA 95054 USA
HitachiVantara.com | community.HitachiVantara.com

Contact Information
USA: 1-800-446-0744
Global: 1-858-547-4526
HitachiVantara.com/contact