Hitachi Content Platform for Cloud Scale

v2.0.0

Installing HCP for Cloud Scale

This document contains instructions for installing or updating the Hitachi Content Platform for cloud scale (HCP for cloud scale) software. It describes how to install single- and multi-instance HCP for cloud scale systems.

MK-HCPCS002-07
July 2020

© 2020 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, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or 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.

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

Contents

Preface 6
- About this document 6
- Intended audience 6
- Product version 6
- Release notes 6
- Related documents 6
- Document conventions 7
- Accessing product documentation 8
- Getting help 8
- Comments 9

Chapter 1: Hitachi Content Platform for cloud scale overview 10
- About HCP for cloud scale 10
- System scaling 10
  - Single-instance systems vs. multi-instance systems 11
  - About master and worker instances 12
- Services 12
- Volumes 13

Chapter 2: System requirements and sizing 15
- Hardware requirements 15
- Operating system and Docker minimum requirements 16
- Operating system and Docker qualified versions 16
  - Docker considerations 16
  - SELinux considerations 17
- Virtual machine host requirements 17
- Networking 18
- Time source 20
- Supported browsers 20

Chapter 3: Installing HCP for cloud scale 21
- Items and information you need 21
- HCP for cloud scale installation steps 21
- Decide how many instances to deploy 22
- Configure your networking environment 22

(Optional) Select master instances....22
Install Docker on each server or virtual machine....23
Configure Docker on each server or virtual machine....23
(Optional) Install Docker volume drivers....24
(Optional) Enable or disable SELinux on each server or virtual machine....24
Configure maximum map count setting....24
Configure the firewall rules on each server or virtual machine....25
Run Docker on each server or virtual machine....25
Unpack the installation package....25
(Optional) Reconfigure network.config on each server or virtual machine....27
(Optional) Reconfigure volume.config on each server or virtual machine....28
Run the setup script on each server or virtual machine....31
Start the application on each server or virtual machine....33
(Optional) Configure NTP....34
Use the service deployment wizard....34
(Optional) Configure networks for services....36
(Optional) Configure volumes for services....37
(Optional) Verify the created volumes....40
(Optional) Distribute services among system instances....40
Moving and scaling floating services....41
Moving and scaling services with multiple types....41
Best practices....41
Considerations....41
Troubleshooting....41
Relocating services....42
Configure the system for your users....44

Chapter 4: Updating HCP for cloud scale....45

Items and information you need....45
Verify and place the update archive....45
Use the System Management application to update the system....46
Configure the system for your users....48
Troubleshooting....48

Appendix A: Logs and diagnostic information....50

Appendix B: Services list....55

HCP for cloud scale services....55

Appendix C: Handling network changes....73

Safely changing an instance IP address....73
After a network change....73

Preface

About this document

This document contains instructions for installing or updating the Hitachi Content Platform for cloud scale (HCP for cloud scale) software. It describes how to install single- and multi-instance HCP for cloud scale systems.

Intended audience

This document is intended for people who are installing or updating HCP for cloud scale systems. It assumes you have some experience creating Docker configurations and installing computer software.

Product version

This document applies to v2.0.0 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 Hitachi Vantara Support Connect: https://knowledge.hitachivantara.com/Documents.

Related documents

This is the set of documents containing information about HCP for cloud scale. You should have these documents available before using the product.

- *Hitachi Content Platform for Cloud Scale Release Notes* (RN-HCPCS004-07): This document is for customers and describes new features, product documentation, and resolved and known issues, and provides other useful information about this release of the product.
- *Installing Hitachi Content Platform for Cloud Scale* (MK-HCPCS002-07): This document gives you the information you need to install or update the HCP for cloud scale software.

- *Hitachi Content Platform for Cloud Scale Administration Guide* (MK-HCPCS008-03): This document explains how to use the HCP for cloud scale applications to configure and operate a common object storage interface for clients to interact with; configure HCP for cloud scale for your users; enable and disable system features; and monitor the system and its connections.
- *Hitachi Content Platform for Cloud Scale S3 Console Guide* (MK-HCPCS009-00): 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.
- *Hitachi Content Platform for Cloud Scale Management API Reference* (MK-HCPCS007-05): This document is for customers and describes the management application programming interface (API) methods available for customer use.

Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click **OK**. ▪ Indicates emphasized words in list items.
Italic	▪ 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: `pairdisplay -g group` (For exceptions to this convention for variables, see the entry for angle brackets.)
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`
< > angle brackets	Indicates variables in the following scenarios: ▪ Variables are not clearly separated from the surrounding text or from other variables. Example: `Status-<report-name><file-version>.csv` ▪ Variables in headings.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.

Convention	Description
{ } 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 Hitachi Vantara Support Connect: 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

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

Hitachi Vantara Community 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: Hitachi Content Platform for cloud scale overview

This is a description of Hitachi Content Platform for cloud scale (HCP for cloud scale) and its main use cases.

About HCP for cloud scale

HCP for cloud scale is a software-only data storage platform that rests on top of physical or cloud-based data storage systems, such as Hitachi Content Platform (HCP) and Amazon Web Services (AWS). HCP for cloud scale acts as a common interface between the storage systems that manages all storage objects, including buckets, objects, and metadata. HCP for cloud scale can scale to accommodate for any number of storage systems, and its data storage limitations are defined only by its underlying technologies.

System scaling

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

Instances

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

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

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

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.

Single-instance systems vs. multi-instance systems

An HCP for cloud scale system can have a single instance or can have multiple instances (four or more).

Note: Every instance must meet RAM, CPU, and disk space minimums.

One instance

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

However, a single-instance system has these drawbacks:

- A single-instance system 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.

Multiple instances

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

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

  For information on services, see Services list (on page 55) .
- 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.

For information about adding instances to an existing HCP for cloud scale system, see the HCP for cloud scale online help.

Four-instance system considerations

The minimum HCP for cloud scale configuration has four instances. Four-instance systems should have three master instances.

For information about master instances, see About master and worker instances (on page 12) .

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.

Services

Services perform functions essential to the health or functionality of the system. For example, the Cluster Coordination service manages hardware resource allocation, while the Policy Engine service runs synchronous and asynchronous policies triggered by S3 API requests. Internally, services run in Docker containers on the instances in the system.

Service categories

Depending on what actions they perform, services are grouped into these categories:

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

Some system services run only on master instances.

For a complete list of services, see Services list (on page 55) .

Applications

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

Service instances

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

Each service has a best and required number of instances on which it should run.

You can configure where Hitachi Content Platform for cloud scale services run, but not system services.

Floating services

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

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

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

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

Note: HCP for cloud scale has no services with multiple types.

Networking

Each service binds to a number of ports and to one type of network, either internal or external. Networking for each service is configured during system installation and cannot be changed after a system is running.

For information on configuration, see Networking (on page 18) .

Storage for services

Services can use volumes for storing data.

For information on volumes, see Volumes (on page 13) .

Volumes

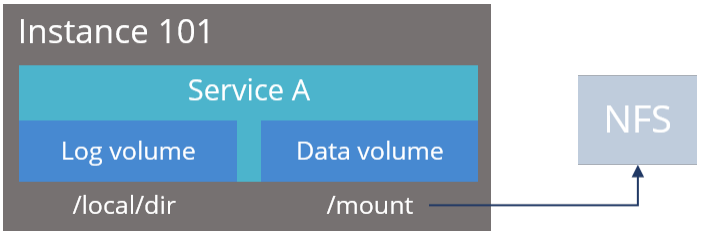
Volumes are properties of services that specify where and how a service stores its data.

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.

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

Depending on how they are created and managed, volumes are separated into these groups:

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

Volume drivers

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

Because services 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.

For information about volume drivers, see the Docker documentation.

For information about services, see Services (on page 12) .

Chapter 2: System requirements and sizing

The hardware, networking, and operating system requirements for running an HCP for cloud scale system with one or more 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 minimum and best-practice hardware requirements for each instance in an HCP for cloud scale system.

Resource	Minimum	Best
RAM	32 GB	128 GB
CPU	8-core	24-core
Available disk space	500 GB 10k SAS RAID	2000 GB 15k SAS RAID
Network interface controller (NIC)	(1) 10 Gb Ethernet	(2) 10 Gb Ethernet
IP addresses	(1) static	(2) static
Firewall Port Access	Port 443 for S3 API and S3 Console application Port 8000 for System Management application GUI Port 8443 for S3 Gateway API Port 9084 for MAPI and Object Storage Management application GUI	Same
Internal IP Ports	See Networking (on page 18)	Same
Network Time	IP address of time service (NTP)	Same

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

Operating system and Docker minimum requirements

Each server or virtual machine you provide must have the following:

- 64-bit Linux distribution
- Docker version 1.13.1 or later installed
- 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: Docker versions after 18.09 are not supported.

Operating system	Docker version	Docker storage configuration	SELinux setting
Fedora 31	Docker 18.09.0-ce	direct-lvm	Enforcing
Red Hat Enterprise Linux 7.4	Docker 18.09.0-ce	direct-lvm	Enforcing
Ubuntu 16.04-LTS	Docker 17.03.0-ce	aufs	N/A
CentOS 7.4	Docker 18.09.1-ce	overlay2	Enforcing

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.

Virtual machine host requirements

You can deploy HCP for cloud scale on virtual machines from a .tgz file. Instances deploying HCP for cloud scale cannot run other software and multiple HCP for cloud scale nodes cannot be deployed on a single instance. To deploy multiple instance on the same hardware, use a hypervisor and guest VMs and deploy each HCP for cloud scale instance on a separate guest.

The HCP for cloud scale software has been qualified on these virtual machine host platforms:

- Hitachi Unified Compute Platform (UCP) 4.1.0

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 your networking environment changes such that the system can no longer function with its current networking configuration, you need to reinstall the system.

Site hostname

The HCP for cloud scale site hostname is configured during installation. This hostname is very important because it's needed for access to:

- The HCP for cloud scale user interface
- The S3 API

Instance IP address requirements

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

Important: If the IP address of any instance changes, you must reinstall the system.

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 your 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 your 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 your system must be addressable by two IP addresses, one on your internal network and one on your external network. If you use only one network type, each instance needs only one IP address.

Allowing access to external resources

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

- The storage components where your 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.

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	Object persistence and access
8000	Admin App	System Management application GUI
9099	MAPI Gateway	Object Storage Management application GUI

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.

Supported browsers

The HCP for cloud scale web applications support these web browsers:

- Google Chrome latest
- Mozilla Firefox latest

Chapter 3: Installing HCP for cloud scale

The following procedures describe how to install the HCP for cloud scale software.

After you installed the software, log in and deploy the system.

Items and information you need

To install an HCP for cloud scale system, you need the appropriate installation package containing the product installation tarball (archive) file `hcpcs-`*`version_number`*`.tgz`.

This document shows the path to the HCP for cloud scale folder as *install_path*. The best folder path is `/opt`.

HCP for cloud scale installation steps

The installation process for HCP for cloud scale consists of these steps. Some steps are required and some are optional, but you must do them in the order shown.

1. Decide how many instances to deploy (on page 22)
2. Configure your networking environment (on page 22)
3. (Optional) Select master instances (on page 22)
4. Install Docker on each server or virtual machine (on page 23)
5. Configure Docker on each server or virtual machine (on page 23)
6. (Optional) Install Docker volume drivers (on page 24)
7. (Optional) Enable or disable SELinux on each server or virtual machine (on page 24)
8. Configure maximum map count setting (on page 24)
9. Configure the firewall rules on each server or virtual machine (on page 25)
10. Run Docker on each server or virtual machine (on page 25)
11. Unpack the installation package (on page 25)
12. (Optional) Reconfigure network.config on each server or virtual machine (on page 27)
13. (Optional) Reconfigure volume.config on each server or virtual machine (on page 28)
14. Run the setup script on each server or virtual machine (on page 31)
15. Start the application on each server or virtual machine (on page 33)
16. (Optional) Configure NTP (on page 34)
17. Use the service deployment wizard (on page 34)

   a. (Optional) Configure networks for services (on page 36)
   b. (Optional) Configure volumes for services (on page 37)

18. (Optional) Verify the created volumes (on page 40)
19. (Optional) Distribute services among system instances (on page 40)
20. Configure the system for your users (on page 44)

Decide how many instances to deploy

Before installing a system, you need to decide how many instances the system will have.

The minimum for a production system is four instances.

Procedure

1. Decide how many instances you need.
2. Select the servers or virtual machines in your environment that you intend to use as HCP for cloud scale instances.

Configure your networking environment

Before installing the system, you need to determine the networks and ports each HCP for cloud scale service will use.

Procedure

1. Determine what ports each HCP for cloud scale service should use. You can use the default ports for each service or specify different ones.

   In either case, these restrictions apply:

   - Every port must be accessible from all instances in the system.
   - Some ports must be accessible from outside the system.
   - All port values must be unique; no two services, whether System services or HCP for cloud scale services, can share the same port.

2. Determine what types of networks, either internal or external, to use for each service.

   If you're using both internal and external networks, each instance in the system must have IP addresses on both your internal and external networks.

(Optional) Select master instances

You need to select which of the instances in your system will be master instances.

If you are installing a multi-instance system, the system must have either one or three master instances, regardless of the total number of instances it includes.

Important:

- For a production system, use three master instances.
- You cannot add master instances to a system after it's installed. You can, however, add any number of worker instances.

If you are deploying a single-instance system, that instance will automatically be configured as a master instance and run all services for the system.

Procedure

1. Select which of the instances in your system are intended as master instances.
2. Make note of the master instance IP addresses.

   Note: To ensure system availability, run master instances on separate physical hardware from each other, if possible.

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) Install Docker volume drivers

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

Procedure

1. If any services on your system are using Docker volume drivers (not the bind-mount setting) for storing data, install those volume drivers on the new instance that you are adding.

   If you don't, services might fail to run on the new instance.
2. If any services on your system use Docker volume drivers for storing data (instead of using the default bind-mount setting), install those volume drivers on all instances in the system.

(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 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.

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.

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 product installation package and MD5 checksum file and store both in a folder on the server or virtual machine.
2. Verify the integrity of the installation package:
   `md5sum -c product-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 product installation folder.
   `mkdir install_path/product`
4. Move the installation package from the folder where you stored it to the product installation folder.
   `mv product-version_number.tgz install_path/product/product-version_number.tgz`
5. Navigate to the installation folder.
   `cd install_path/product`
6. Unpack the installation package:
   `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 more than one time during download. Use the following commands to fully extract the file:

   ```
   $ gunzip product-version_number.tgz
   $ mv product-version_number.tar product-version_number.tgz
   $ tar -zxf product-version_number.tgz
   ```

7. Run the installation script `install`, located within a folder matching the version number of system software used by the product software.
   `sudo ./cluster/sys_ver_num/bin/install`

   This version number is different from the product version number. The subfolder is the only one in the folder `cluster`.

   For example:

   `sudo ./cluster/1.4.0.260/bin/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.

(Optional) Reconfigure network.config on each server or virtual machine

Before you begin

Important: To reconfigure networking for the System services, you must complete this step before you run the setup script on each server or virtual machine.

You cannot change networking for System services after running the script `run` or after starting the service hcpcs.service using systemd.

To change the networking settings of System services, do so in this step, before running the product startup scripts. You configure networking for HCP for cloud scale services later when using the deployment wizard.

You can change these networking settings for each service in your product:

- The ports that the service uses.
- The network to listen on for incoming traffic, either internal or external.

To configure networking for the System services:

Procedure

1. On each server or virtual machine that is to be an HCP for cloud scale instance, use a text editor to open the file *install_path*`/hcpcs/config/network.config`.

   The file contains two types of lines for each service:

   - **Network type assignments:** For example:

     `com.hds.ensemble.plugins.service.`*service_name*`_interface=[internal|external]`

     `com.hds.ensemble.plugins.service.zookeeper_interface=internal`

   - **Port number assignments:** For example:

     `com.hds.ensemble.plugins.service.`*service_name*`.port.`*port_name*`=`*port_number*

     `com.hds.ensemble.plugins.service.zookeeper.port.PRIMARY_PORT=2181`

2. Type new port values for the services you want to configure.

   Note: If you reconfigure service ports, make sure that each port value you assign is unique across all services, both System services and HCP for cloud scale services.

Note: By default, all System services are set to `internal`.

If you're only using a single network, you can leave these settings as they are. This is because all system instances are assigned both internal and external IP addresses in HCP for cloud scale; if you're only using a single network type, the internal and external IP addresses for each instance are identical.

3. On the lines containing `_interface`, specify the network that the service should use. Valid values are **internal** and **external**.
4. Save your changes and exit the text editor.

Next steps

Important: Ensure that the file `network.config` is identical on all HCP for cloud scale instances.

(Optional) Reconfigure volume.config on each server or virtual machine

Before you begin

Caution: To reconfigure volumes for the System services, you must complete this step before you run the setup script on each server or virtual machine.

You cannot change volumes for System services after running the script run or after starting the service `hcpcs.service` using systemd.

By default, each of the System services is configured not to use volumes for storage (each service uses the bind-mount option). To change this configuration, you can do that now in this step, before running the product startup scripts.

Tip: System services typically do not store a lot of data, so you should favor keeping the default bind-mount setting for them.

You configure volumes for HCP for cloud scale services later when using the deployment wizard.

To configure volumes for the System services:

Procedure

1. On each server or virtual machine that is to be an HCP for cloud scale instance, use a text editor to open the file *`install_path`*`/hcpcs/config/volume.config`.

This file contains information about the volumes used by the System services. For each volume, the file contains lines that specify the following:

- The name of the volume:

  `com.hds.ensemble.plugins.service.`*`service_name`*`.`*`volume`*`_name=`*`volume_name`*

  > **Note:** Do not edit the volume names. The default volume name values contain variables (SERVICE_PLUGIN_NAME and INSTANCE_UUID) that ensure that each volume gets a unique name.

- The volume driver that the volume uses:

  `com.hds.ensemble.plugins.service.`*`service_name`*`.`*`volume`*`_driver=[`*`volume_driver_name`*` | bind-mount]`

- The configuration options used by the volume driver. Each option is listed on its own line: For example, these lines describe the volume that the Admin-App service uses for storing its logs:

  `com.hds.ensemble.plugins.service.`*`service_name`*`.`*`volume`*`_driver_opt_`*`option_number`*`=`*`volume_driver_option_and_value`*

```
com.hds.ensemble.plugins.service.adminApp.log_volume_name=SERVICE_PLUGIN_NAME.INSTANCE_UUID.log
com.hds.ensemble.plugins.service.adminApp.log_volume_driver=bind-mount
com.hds.ensemble.plugins.service.adminApp.log_volume_driver_opt_1=hostpath=/home/hcpcs/log/com.hds.ensemble.plugins.service.adminApp/
```

2. For each volume that you want to configure, you can edit the following:
   - The volume driver for the volume to use. To do this, replace `bind-mount` with the name of the volume driver you want.

     Volume drivers are provided by Docker and other third-party developers, not by the HCP for cloud scale system itself. For information on volume drivers, their capabilities, and their valid configuration settings, see the applicable Docker or third-party developer's documentation.
   - On the line that contains `_opt`, the options for the volume driver.

     For information about the options you can configure, see the documentation for the volume driver that you're using.

     Caution: Option/value pairs can specify where data is written in each volume. These considerations apply:
     - Each volume that you can configure here must write data to a unique location.
     - The SERVICE_PLUGIN and INSTANCE_UUID variables cannot be used in option/value pairs.
     - Make sure the options and values you specify are valid. Incorrect options or values can cause system deployment to fail or volumes to be set up incorrectly. For information on configuration, see the volume driver's documentation.

     Tip: Create test volumes using the command `docker volume create` with your option/value pairs. Then, to test the volumes you've created, run the command `docker run hello-world` with the option `--volume`.

Example

These lines show a service that has been configured to use the local-persist volume driver to store data:

```
com.hds.ensemble.plugins.service.marathon.data_volume_name=SERVICE_PLUGIN_NAME.INSTANCE_UUID.data
com.hds.ensemble.plugins.service.marathon.data_volume_driver=local-persist
com.hds.ensemble.plugins.service.marathon.data_volume_driver_opt_1=mountpoint=/home/hcpcs/data/com.hds.ensemble.plugins.service.marathon/
```

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/hcpcs/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 this table to determine which options you need 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`

Number of instances in the system	Network type usage	Options to use
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

This 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
```

This 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	`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`
192.0.2.2	10.236.1.2	Master	`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`

Instance internal IP	Instance external IP	Master or worker	Command
192.0.2.3	10.236.1.3	Master	`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`
192.0.2.4	10.236.1.4	Worker	`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`

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 in the event of a server restart or other availability event.

Result

After 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
     ```

(Optional) Configure NTP

If you are installing a multi-instance system:

Procedure

1. Configure NTP (network time protocol) so that each instance uses the same time source.

   For information on NTP, see http://support.ntp.org/.

Use the service deployment wizard

After creating all of your instances and starting HCP for cloud scale, use the service deployment wizard. This wizard runs the first time you log in to the system.

To run the service deployment wizard:

Procedure

1. Open a web browser and go to `https://instance_ip_address:8000`.
   The Deployment Wizard starts.
2. Set and confirm the password for the main **admin** account.

   Important: Do not lose or forget this password.

   When you have defined the password, click **Continue**.

3. On the next page of the deployment wizard, type the cluster host name in the **Cluster Hostname/IP Address** field, then click **Continue**.

   Omitting this can cause links in the System Management application to function incorrectly.
4. On the next page of the deployment wizard, confirm the cluster topology. Verify that all instances that you expect to see are listed.

   If some instances are not displayed, in the **Instance Discovery** window, click **Refresh instances** until they appear.

   When you have confirmed the cluster topology, click **Continue**.
5. On the next page of the deployment wizard, confirm the advanced configuration settings.

   Important: If you decide to reconfigure networking for the HCP for cloud scale services or volume usage or services, you must do so now, before deploying the system.

   For information on configuration, see Networking (on page 18) .

   - For information on networking settings for the HCP for cloud scale services, see (Optional) Configure networks for services (on page 36) .
   - For information on storage volumes used for services, see (Optional) Configure volumes for services (on page 37) .

   When you have confirmed the configuration settings, click **Continue**.
6. On the last page of the deployment wizard, to deploy the cluster, click **Deploy Cluster**.
   After a brief delay, the message "Deployment in progress" is displayed and instances of services are started.
7. When the wizard is finished, the message "Setup Complete" is displayed. Click **Finish**.
   The **Applications** page opens.

Result

Service instances are deployed and the HCP for cloud scale system is ready to use.

Note: If you configured the System services networking incorrectly, the System Management application might not appear as an option on the **Applications** page. This can happen, for example, if the `network.config` file is not identical on all instances. For error information, view the file *`install_path`*`/hcpcs/config/cluster.config` or the output information logged by the script `run`.

To fix this issue, do the following:

1. Stop the script `run`. You can do this using whatever method you're currently using to run the script.
2. Run this command to stop all HCP for cloud scale Docker containers on the instance:

   `sudo` *`install_path`*`/hcpcs/bin/stop`
3. Delete the contents of the folder *`install_path`*`/hcpcs` from all instances.
4. Delete any Docker volumes created during the installation:

   `docker volume rm` *`volume-name`*
5. Begin the installation again from the step where you unpack the installation package.

Note: The following messages indicate that the deployment process failed to initialize a Metadata Gateway service instance:

- If the deployment process repeatedly tries and fails to reach a node, it displays this message: "`Failed to initialize all MetadataGateway instances. Please re-deploy the system.`"
- If the deployment process detects an existing Metadata Gateway partition on a node, it displays this message: "`Found existing metadata partitions on nodes, please re-deploy the system.`"

If you see either message, you can't resolve the issue by clicking Retry. Instead, you must reinstall the HCP for cloud scale software.

(Optional) Configure networks for services

To change networking settings for the HCP for cloud scale services:

Procedure

1. On the **Advanced Configuration** page, select the service to configure.
2. On the **Network** tab:
   a. Configure the ports that the service should use.

   Note: If you reconfigure service ports, make sure that each port value you assign is unique across all services, both System services and HCP for cloud scale services.

b. For each service, specify the network, either **Internal** or **External**, to which the service should bind.

> **Note:** By default, the HCP for cloud scale services have the **External** network selected and the System services have the **Internal** network selected.
>
> If you're only using a single network, you can leave these settings as they are. This is because all system instances are assigned both internal and external IP addresses in HCP for cloud scale; if you're only using a single network type, the internal and external IP addresses for each instance are identical.

(Optional) Configure volumes for services

To change volume usage:

Procedure

1. On the **Advanced Configuration** page, select a service to configure.
2. Click the **Volumes** tab. This tab displays the system-managed volumes that the service supports. By default, each built-in service has both Data and Log volumes.
3. For each volume, provide Docker volume creation information:
   a. In the **Volume Driver** field, specify the name of the volume driver that the volume should use. To configure the volume not to use any volume driver, specify **bind-mount**, which is the default setting.

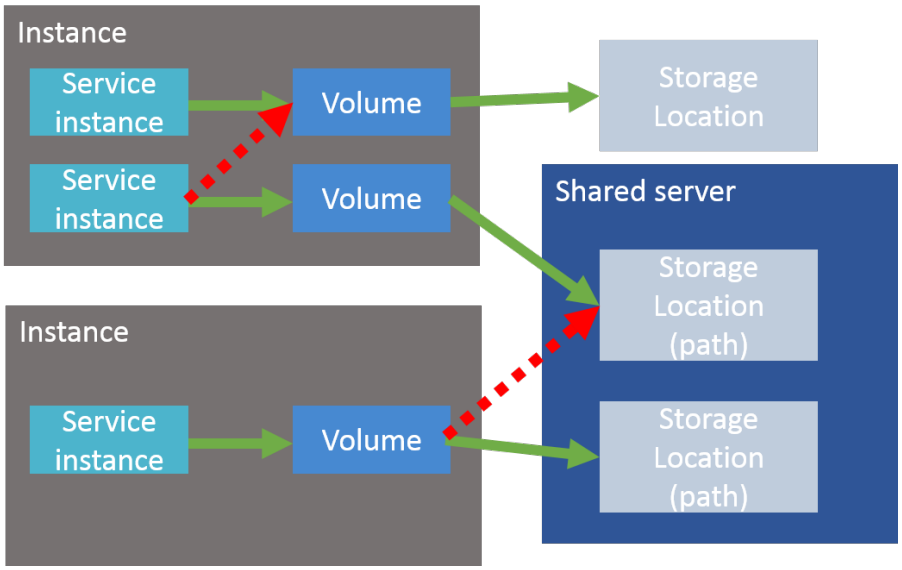
   > **Note:** Volume drivers are provided by Docker and other third-party developers, not by the HCP for cloud scale system itself. For information on volume drivers, their capabilities, and their valid configuration settings, see the applicable Docker or third-party developer's documentation.

   b. In the **Volume Driver Options** section, in the **Option** and **Value** fields, specify any optional parameters and their corresponding values for the volume driver:
      - If you're using the **bind-mount** setting, you can edit the value for the `hostpath` option to change the path where the volume's data is stored on each system instance. However, this must be a path within the HCP for cloud scale installation folder.
      - If you're using a volume driver:
        i. Click the trashcan icon to remove the default **hostpath** option. This option applies only when you are using the **bind-mount** setting.
        ii. Type the name of a volume driver option in the **Option** field. Then type the corresponding parameter for that option in the **Value** field.
        iii. Click the plus-sign icon to add the option/value pair.
        iv. Repeat this procedure for each option/value pair you want to add.

Option/value pairs can specify where data is written to in each volume. These considerations apply:

- Each service instance must write its data to a unique location. A unique location can be a file system or a unique path on a shared external storage server.

  In this illustration, green arrows show acceptable configurations and red arrows show unacceptable configurations where multiple service instances are writing to the same volume, or multiple volumes are backed by the same storage location:



- For persistent (that is, non-floating) services, favor using the `${container_inst_uuid}` variable in your option/value pairs. For persistent services, this variable resolves to a value that's unique to each service instance.

  This is especially useful if the volume driver you're using is backed by a shared server. By providing a variable that resolves to a unique value, the volume driver can use the resolved variable to create unique directories on the shared server.

  However, some volume drivers, such as Docker's local volume driver, do not support automatic Folder creation. If you're using such a volume driver, you need to create volume folders yourself. For an example of how to handle this, see the following Docker local volume driver example.

- Floating services do not support volumes that are backed by shared servers, because floating services do not have access to variables that resolve to unique values per service instance.

- Make sure the options and values you specify are valid. Options or values that are not valid can cause system deployment to fail or volumes to be set up incorrectly. For information on volumes, see the volume driver's documentation.

Tip: Create test volumes by use the command `docker volume create` with your option/value pairs. Then, to test the volumes you created, use the command `docker run hello-world --volume`.

You can include these variables when configuring volume options:

- `${install_dir}` is the product installation folder.
- `${data_dir}` is equal to `${install_dir}/data`
- `${log_dir}` is equal to `${install_dir}/log`
- `${volume_def_name}` is the name of the volume you are configuring.
- `${plugin_name}` is the name of the underlying service plugin.
- `${container_inst_uuid}` is the UUID for the Docker container in which the service instance runs. For floating services, this is the same value for all instances of the service.
- `${node_ip}` is the IP address for the system instance on which the service is running. This cannot be used for floating services.
- `${instance_uuid}` is the UUID for the system instance. This cannot be used for floating services. For services with multiple types, this variable resolves to the same value for all instances of the service, regardless of their types.

4. Repeat this procedure for each service that you want to configure.

bind-mount configuration for Database service log volume

Docker local volume driver for Database service log volume

The built-in Database service has a volume called log, which stores the service's logs. The log volume has this default configuration:

- Volume driver: bind-mount
- Option: hostname, Value: ${log_dir}/${plugin_name}/${container_inst_uuid}

With this configuration, after the system is deployed, logs for the Database service are stored at a unique path on each system instance that runs the Database service:

install_path`/hcpcs/log/com.hds.ensemble.plugins.service.cassandra/`*service-instance-uuid*

Alternatively, you can configure the Database service to use Docker's built-in local volume driver to store logs on an NFS server. To do this:

1. Log in to your NFS server.
2. Create a folder.
3. Within that folder, create one folder for each of the instances in your system. Name each one using the instance IP address.

   Note: In this example, you need to create these folders yourself because the local storage driver will not create them automatically.

4. Back in the system deployment wizard, in the Volume Driver field, specify local

5. Specify these options and values:

Option	Value
type	**nfs**
o	**addr=** ***nfs-server-ip*****,rw**
device	**:/*path-to-folder-from-step-ii*/${node_ip}**

With this configuration, each instance of the Database service stores its logs in a different folder on your NFS server.

(Optional) Verify the created volumes

Before you begin

If you configured the service volumes to use volume drivers, use these commands to list and view the Docker volumes created on all instances in the system:

```
docker volume ls
```

```
docker volume inspect volume_name
```

If volumes were created incorrectly, you need to redo the system installation:

Procedure

1. Stop the `run` script from running. You do this using whatever method you're currently using to run the script.
2. Stop all HCP for cloud scale Docker containers on the instance:
   `sudo install_path/hcpcs/bin/stop`
3. Delete the contents of the folder `install_path/hcpcs` from all instances.
4. Delete any Docker volumes created during the installation:
   `docker volume rm volume_name`
5. Begin the installation again from the point where you unpack the installation package.

(Optional) Distribute services among system instances

By default, when you install and deploy a multi-instance system, the system automatically runs each service (except Dashboard) on its normal number of instances.

However, if you've installed more than four instances, some instances may not be running any services at all. As a result, these instances are under-used. You should manually distribute services to run across all instances in your system.

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

Here are some guidelines for distributing services across instances:

- Avoid running multiple services with high service unit costs together on the same instance.
- On master instances, avoid running any services besides those classified as System services.

Considerations

- Instance requirements vary from service to service. Each service defines the minimum and maximum number of instances on which it can run.
- You cannot remove a service from an instance if doing so causes or risks causing data loss.
- Service relocation might take a long time to complete and can impact system performance.

Troubleshooting

You might encounter these issues during installation.

Service doesn't start

Rarely, a system deployment, service management action, or system update fails because a service fails to start. When this happens, the System Management application is inaccessible from the instance where the failure occurred.

The logs in the watchdog-service log folder contain this error:

`Error response from daemon: Conflict. The name "`*`service-name`*`" is already in use by container `*`Docker-container-id`*`. You have to remove (or rename) that container to be able to reuse that name.`

To resolve this issue, restart the Docker service on the instance where the service failed to start. For example, if you are using `systemd` to run Docker, run:

```
systemctl restart docker
```

After restarting Docker, try the system deployment, service management action, or system update again.

Relocating services

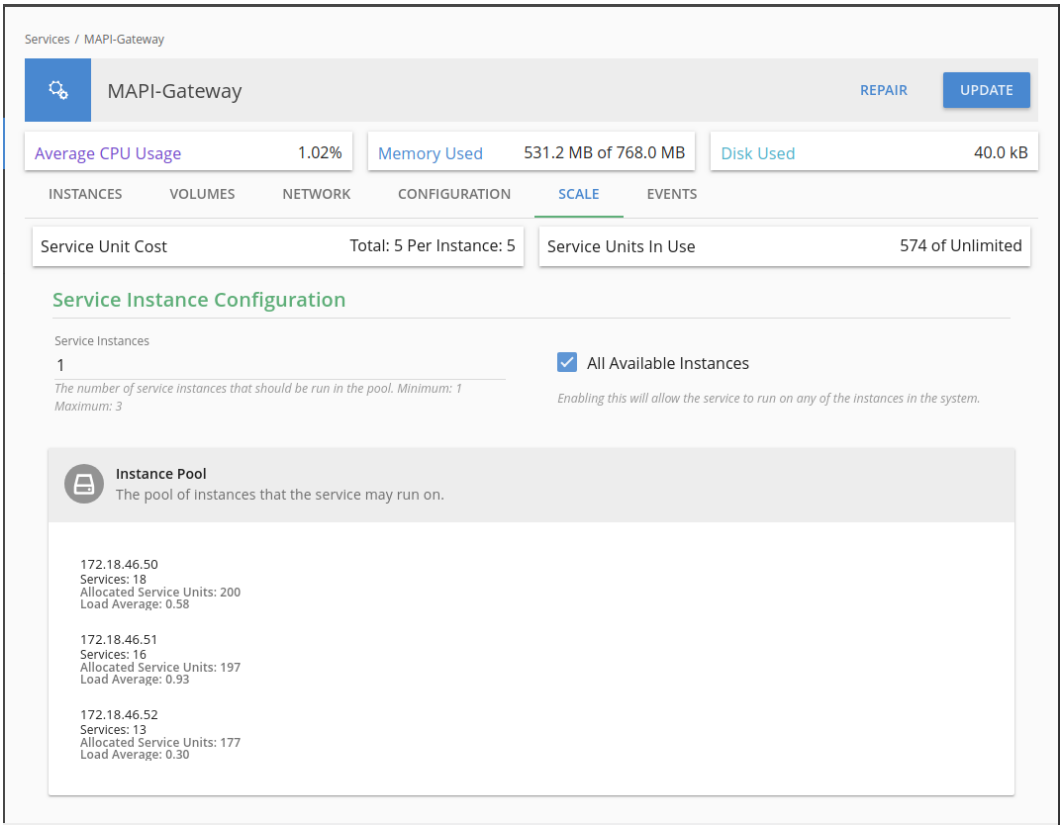
To manually relocate a service, in the Admin App:

Procedure

1. Select **Services**.
   The **Services** page opens, displaying the services and system services.
2. Click on 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:



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

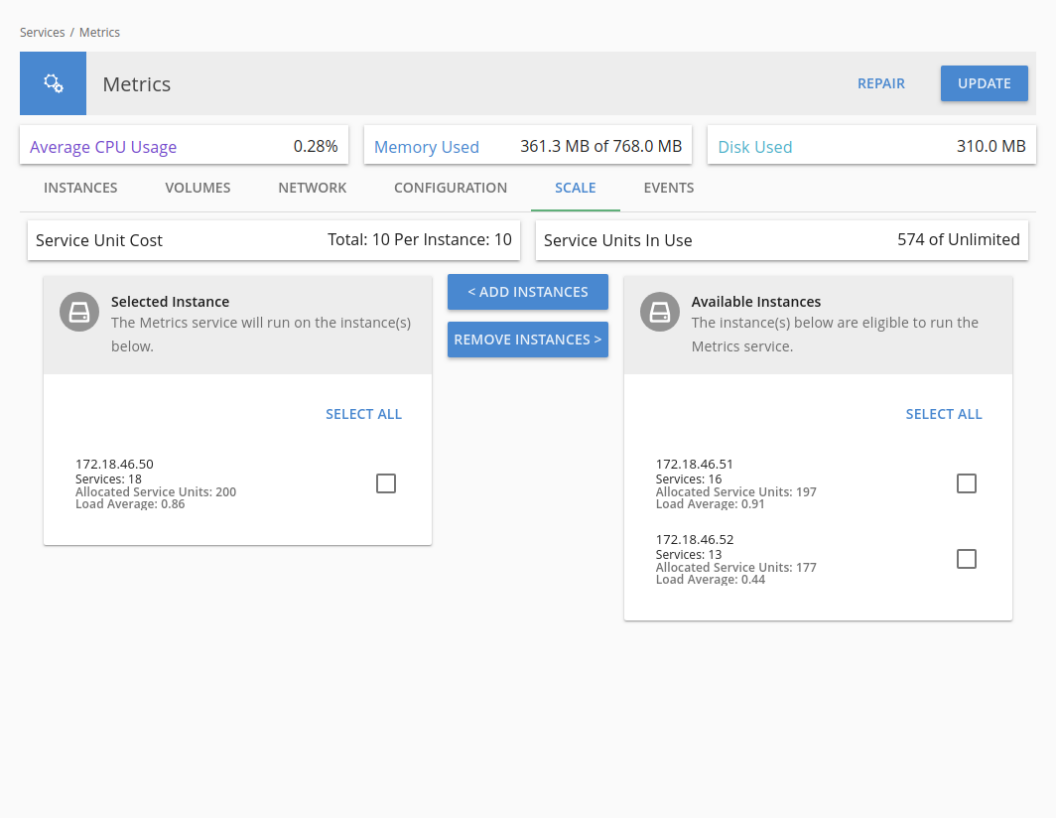
b. Configure the instance pool:

- 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, deselect **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**.

5. 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 may take a few minutes to appear on the Services page.

Configure the system for your users

After your system is up and running, you can begin configuring it for your users.

For information about these procedures, see the applicable topic in the help that's available from the HCP for cloud scale application.

The overview of tasks is:

Procedure

1. Configure the connection to an IdP and create user accounts.
2. Define storage components.
3. Configure DNS servers to resolve both the fully qualified domain name for your HCP for cloud scale cluster and the wildcard `*.`*`hcpcs_cluster_name`*.
4. Obtain S3 authorization credentials.

Chapter 4: Updating HCP for cloud scale

The following procedures describe how to update the HCP for cloud scale software.

Updates are managed by the System Management (Admin) application. Instances are shut down, updated, and restarted one at a time automatically, so you can update the HCP for cloud scale software to a newer version without interrupting availability or reingesting data. S3 API methods remain available, so that users can continue to read and write data and create and configure buckets.

During an update, management API methods that don't change the configuration remain available. Tracing and the collection of metrics aren't affected.

Restrictions

You cannot downgrade HCP for cloud scale to a previous version.

You cannot upgrade to v1.3.0 from any previous version.

During an update, you cannot make changes to the configuration. After an update, you might need to reconfigure services.

Items and information you need

To update an HCP for cloud scale system, you need the appropriate update archive file `hcpcs-`*`version_number`*`.update` and, for verification purposes, its MD5 checksum file `hcpcs-`*`version_number`*`.md5`.

This document shows the path to the HCP for cloud scale folder as *install_path*. The best folder path is `/opt`.

Verify and place the update archive

On the server or virtual machine from which you want to start the update:

Procedure

1. Download the update archive and MD5 checksum file and store the files in a folder on the server or virtual machine.
2. The best practice is to verify the integrity of the update archive:
   `md5sum -c `*`product-version_number`*`.update.md5`
   If the archive integrity is verified, the command displays *`product-version_number`*`.update: OK`.

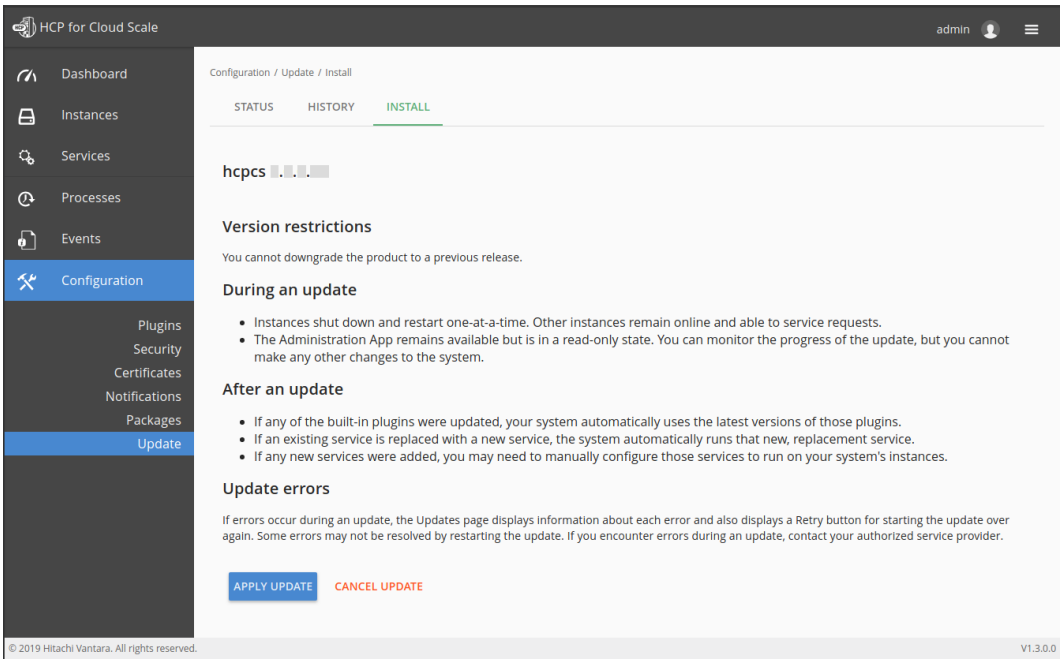
3. In the largest disk partition on the server or virtual machine, create a product update folder:
   `mkdir update_path/product`
4. Move the update archive from the folder where you stored it to the product update folder:
   `mv product-version_number.update update_path/product/product-version_number.update`
5. Navigate to the update folder:
   `cd update_path/product`

Use the System Management application to update the system

The update process is controlled by the System Management application (the Admin App).

Procedure

1. From the System Management application, select **Configuration**.
   The **Configuration** page opens.
2. Click **Update**.
   The **Update** page opens.
3. Select the **Install** tab.
   The **Upload** area opens.
4. Click and drag the update file into the **Upload** area, or click **Click to Upload**, select the update archive, and click **Open**.
   The update archive is uploaded and verified. These processes will take several minutes. When the archive is verified, the page displays information about the update process. If the update contains a new service, you can optionally configure network, volume, or log file settings.

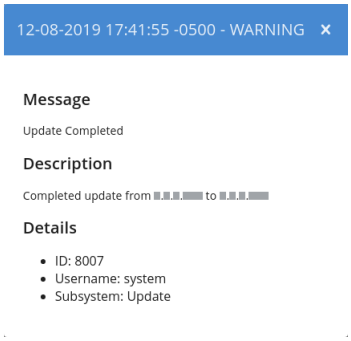


5. When you're ready to begin, click **Apply Update**.
   The System Management application displays the message `Successfully started update package installation` and then applies the update to the cluster (each server or virtual machine in the HCP for cloud scale system).

Result

When the system is updated:

- The `Successfully started` banner message is removed.
- In the Status tab, the Update Status displays `None` and the event `Update Completed` appears.



- The
- The update version appears in the lower right corner of the **Login** page.

Note: If the update process encounters an error, update stops and the **Install** tab displays the message `The update has failed! Click retry to attempt the update again.` On the **Status** tab, click on the error message to get more details about the error. Click **Retry**, or click **Cancel**, correct the problem, and then restart the update. Some errors might not be resolvable by restarting the update. If a retry doesn't work, contact Support.

Configure the system for your users

After your system is up and running, you can begin configuring it for your users.

For information about these procedures, see the applicable topic in the help that's available from the HCP for cloud scale application.

The overview of tasks is:

Procedure

1. Configure the connection to an IdP and create user accounts.
2. Define storage components.
3. Configure DNS servers to resolve both the fully qualified domain name for your HCP for cloud scale cluster and the wildcard `*.`*`hcpcs_cluster_name`*.
4. Obtain S3 authorization credentials.

Troubleshooting

You might encounter these issues during an update.

Service doesn't start

Rarely, a system deployment, service management action, or system update fails because a service fails to start. When this happens, the System Management application is inaccessible from the instance where the failure occurred.

The logs in the watchdog-service log folder contain this error:

`Error response from daemon: Conflict. The name "`*`service-name`*`" is already in use by container `*`Docker-container-id`*`. You have to remove (or rename) that container to be able to reuse that name.`

To resolve this issue, restart the Docker service on the instance where the service failed to start. For example, if you are using systemd to run Docker, run:

```
systemctl restart docker
```

After restarting Docker, try the system deployment, service management action, or system update again.

PUT and GET calls on encrypted objects fail

After you turn on encryption, PUTS and GETS of objects require the key management server (KMS) to be up and unsealed. During an update the KMS can restart multiple times, including when the master nodes and services are upgraded and when the Vault service is updated. When the KMS service restarts, it is sealed, which can interrupt service.

If you are using encryption, monitor the Vault service closely during an update to prevent interruptions. Whenever the service restarts and gets sealed, unseal it.

If you have access to the Aspen administration app, you can monitor the health of the KMS by checking for the alert "`Failed to connect to KMS server`." When you see this alert, you know that the KMS is either down or sealed.

Another approach is to load the KMS page, which is at port 8200 of the system. The status of the KMS is displayed in the upper right corner. A red dot indicates that it is sealed.

Appendix A: Logs and diagnostic information

Each service maintains its own set of logs. By default, the logs 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, non-default location.

Log management

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

Note: 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:

- 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.
- 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 too large, it is renamed to exampleService.log.1 and the system creates a new file named exampleService.log.0 to write to.

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

By default, each service stores its logs 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 directories.

For information about services, see Services list (on page 55).

Default log folder name	Related service	Contains information about
com.hds.ensemble.plugins.service.adminApp	Admin-App	The System Management application.
com.hitachi.aspen.foundry.service.Aspen	Aspen	The HCP for cloud scale application.
com.hds.ensemble.plugins.service.cassandra	Database	▪ System configuration data. ▪ Document fields and values.
com.hds.ensemble.plugins.service.chronos	Scheduling	Workflow task scheduling.
com.hitachi.aspen.foundry.service.clientaccess.data	S3-Gateway	The client access data service.

Default log folder name	Related service	Contains information about
com.hds.ensemble.plugins.service.elasticsearch	Metrics	The storage and indexing of: ▪ System events ▪ Performance and failure metrics for workflow tasks
com.hds.ensemble.plugins.service.haproxy	Network-Proxy	Network requests between instances.
com.hitachi.aspen.foundry.service.tracing.agent	Tracing-Agent	The tracing agent service.
com.hitachi.aspen.foundry.service.tracing.collector	Tracing-Collector	The tracing collector service.
com.hitachi.aspen.foundry.service.tracing.query	Tracing-Query	The tracing query service.
com.hds.ensemble.plugins.service.kafka	Message Queue	The transmission of data between instances.
com.hds.ensemble.plugins.service.logstash	Logging	The transport of system events and workflow task metrics to the Metrics service.
com.hitachi.aspen.foundry.service.mapi.gateway	MAPI-Gateway	The management API gateway.
com.hitachi.aspen.foundry.service.vault.vault	Key-Management-Server	The key management server.

Default log folder name	Related service	Contains information about
com.hds.ensemble.plugins.service.marathon	Service-Deployment	The deployment of high-level services across system instances. High-level services are the ones that you can move and configure (such as Index), not the services grouped under System Services.
com.hds.ensemble.plugins.service.mesosAgent	Cluster-Worker	The work ordered by the Cluster-Coordination service.
com.hds.ensemble.plugins.service.mesosMaster	Cluster-Coordination	Hardware resource allocation.
com.hitachi.aspen.foundry.service.metadata.policy.engine	Policy-Engine	The metadata asynchronous policy engine.
com.hitachi.aspen.foundry.service.metrics.prometheus	Telemetry-Service	Telemetry.
com.hitachi.aspen.foundry.service.rabbitmq.server	Message Queue	The message broker.
com.hds.ensemble.plugins.service.sentinel	Sentinel	The internal system processes.
com.hds.ensemble.plugins.service.watchdog	Watchdog	General diagnostic information.

Default log folder name	Related service	Contains information about
com.hds.ensemble.plugins.service.zookeeper	Synchronization	The coordination of actions and database activities across instances.

Appendix B: Services list

This is a description of the HCP for cloud scale services and how to configure them.

HCP for cloud scale services

This 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 your system includes more than the minimum number of instances, you should take advantage of the instances by running services on them.
- Whether the service is persistent (that is, it must run on a specific instance) or supports floating (that is, it can run on any instance).
- Whether the service is scalable or not.

Note: For HCP for cloud scale services, you cannot set the size of Container Memory larger than the value of the setting Max Heap Size. For other services, you should not set the size of Container Memory larger than the value of the setting Max Heap Size.

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.		

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Cassandra Decentralized database that can be scaled across large numbers of hardware nodes.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2.4 GB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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
	Service Options ▪ **Max heap**: 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 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.	Number of instances:	Minimum: 3 Best: All
		Service unit cost:	10
	Advanced Options: **Compaction Frequency**: How often the database is compacted. The options are **Weekly** (default) and **Daily**. **Caution:** Changing this setting can negatively affect the service. Use with caution.	Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	10
Chronos Job scheduling.	**Container Options: Default**	RAM needed per instance:	712 MB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **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.	Service unit cost:	1
	Service Options ▪ **Heap settings**: 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 356 MB.	Number of instances:	Minimum: 1 Best: 1
		Persistent or floating?	Floating
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	1
Elasticsearch Data indexing and search platform.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2 GB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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 GB
		Service unit cost:	25
	Service Options	Number of instances:	Minimum: 3 Best: All

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **Heap settings**: 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. ▪ **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.	Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	25
Kafka Stream processing platform for handling real-time data streams.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2 GB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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 GB
		Service unit cost:	5
	Service Options ▪ **Heap settings**: 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 1 GB.	Number of instances:	Minimum: 3 Best: All
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	5
Key Management Server Manages storage component encryption keys	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2 GB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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.	RAM needed per instance:	2 GB
		Service unit cost:	10
		Number of instances:	Minimum: 1 Best: 2 or more
		Persistent or floating?	Floating
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	10
Logstash Collection engine for event data. Can perform transformations on the data it collects and then send that data to a number of outputs.	**Container Options: Default** ▪ **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.	RAM needed per instance:	700 MB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	Service Options ▪ **Max heap**: 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 350 MB.	Number of instances:	Minimum: 1 Best: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	10
MAPI Gateway Serves MAPI methods.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
		Service unit cost:	5
	Service Options ▪ **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 384 MB.	Number of instances:	Minimum: 1 Best: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Service unit cost:	5
Message Queue Provides system queueing services.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2048 MB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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.	RAM needed per instance:	2048 MB
		Service unit cost:	10
	Service Options None.	Number of instances:	Minimum: 3 Best: 3
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	10

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Metadata Cache Cache for HCP for cloud scale metadata	**Container Options: Default** ▪ **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.	RAM needed per instance:	1024MB
	Service Options ▪ **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 384 MB.	Number of instances:	Minimum: 1 Best: 1
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	10
Metadata Coordination Coordinates Metadata Gateway service instances and coordinates scaling and balancing.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	Service Options ▪ **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 384 MB.	Number of instances:	Minimum: 1 Best: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	5
Metadata Gateway Stores and protects metadata and serves it to other services.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 4096 MB. ▪ **CPU**: The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container 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:	768 MB
		Service unit cost:	50
	Service Options ▪ **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 2048 MB.	Number of instances:	Minimum: 3 Best: All
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Service unit cost:	50
Metrics	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
		Service unit cost:	10
	Service Options ▪ **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 tsdb/. ▪ **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.	Number of instances:	Minimum: 1 Best: 1
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	10
Policy Engine Executes system policy operations and asynchronous metadata updates.	**Container Options: Default** ▪ **Container Memory**: The hard memory limit for the service's Docker container, in MB. The default is 2048 MB.	RAM needed per instance:	768 MB

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **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.	Service unit cost:	25
	Service Options ▪ **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.	Number of instances:	Minimum: 1 Best: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
S3 Gateway Serves S3 API methods and communicates with storage components.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
		Service unit cost:	25
	Service Options ▪ **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 384 MB.	Number of instances:	Minimum: 1 Best: All

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	HTTP Options: ▪ **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.	Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
	HTTPS Options: ▪ **SSL Ciphers**: A comma-separated list of ciphers used to encode SSL traffic. Changing the list causes the service to redeploy.	Scalable?	Yes
		Service unit cost:	25
Tracing Agent Supports end-to-end distributed tracing for S3 API calls and MAPI calls.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
		Service unit cost:	1
	Service Options ▪ **Collector TChannel Hostname**: Type a host name. The default is localhost. ▪ **Collector TChannel Port**: Type a port number. The default is 14267.	Number of instances:	Minimum: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Service unit cost:	1
Tracing Collector Supports end-to-end distributed tracing for S3 API calls and MAPI calls.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
	Service Options ▪ **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. ▪ **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.	Number of instances:	Minimum: 1 Best: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	10

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Tracing Query Supports end-to-end distributed tracing for S3 API calls and MAPI calls.	**Container Options: Default** ▪ **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.	RAM needed per instance:	768 MB
	Service Options ▪ **ElasticSearch Hostname**: Type a host name. The default is localhost. ▪ **ElasticSearch Port**: Type a port number. The default is 9200.	Number of instances:	Minimum: 1 Best: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service unit cost:	5
System services: These services manage system resources and ensure that the HCP for cloud scale system remains available and accessible. These services cannot be moved or reconfigured.			
Admin App The system management application.	**Service Options** ▪ **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.	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes

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

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Network Proxy Network request load balancer.	**Security Protocol**: Select which Transport Layer Security (TLS) versions to use: ▪ **TLS 1.2** ▪ **TLS 1.3**	RAM needed per instance:	N/A
	SSL Ciphers: To use your own cipher suite, type it here.	Number of instances:	N/A
		Service unit cost:	1
	Custom Global Configuration: Select **Enable Advanced Global Configuration** to enable adding your own parameters to the HAProxy "global" section.	Persistent or floating?	Persistent
	Custom Defaults Configuration: Select **Enable Defaults Configuration** to enable adding your own parameters to the HAProxy "global" section.	Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Sentinel Runs internal system processes and monitors the health of the other services.	**Service Options** ▪ **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
Service Deployment Handles deployment of	N/A	RAM needed per instance:	N/A

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
high-level services (that is, the services that you can configure).		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Synchronization Coordinates service configuration settings and other information across instances.	**Service Options** ▪ **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 Monitors the other System services and restarts them if necessary. Also responsible for initial system startup.	**Service Options** ▪ **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
		Service unit cost:	5
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **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.	Single or multiple types?	Single
		Scalable?	No
		Service unit cost:	5

Appendix C: 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 examples change, you will need to reinstall the system.

Safely changing an instance IP address

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

For each instance whose IP address you need to change:

Procedure

1. Move all services off of the instance. Distribute those services among all the other instances.

   For information on distribution, see (Optional) Distribute services among system instances (on page 40) .
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. For information on exporting packages, see the Administrator Help, which is accessible from the Admin App

2. On each instance in the system:
   a. Navigate to the installation folder.
   b. Stop the script `run` using whatever tool or process you used to run it.
      For example, with systemd, run: `systemctl stop hcpcs.service`
   c. Run `bin/stop`
   d. Run the script `setup`, including the comma-separated list of master instances:
      `sudo bin/setup -i` *`instance_ip_addr`* `-m` *`master_instance_ip_addrs`*
   e. Run the script `run` using whatever methods you usually use to run scripts.
3. Use the Admin App setup wizard.
4. After the system has been set up, upload your package.

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