HITACHI
Inspire the Next

Hitachi Content Platform for Cloud Scale

Version 1.0

Installing HCP for cloud scale

MK-HCPCS002-01
May 2019

© 2019 Hitachi, Ltd. All rights reserved.

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

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

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

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

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

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that 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, IBM, Lotus, MVS, OS/390, PowerPC, RS/6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

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

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

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

Contents

Preface 6

Intended audience 6
Product version 6
Release notes 6
Related documents 6
Accessing product documentation 7
Getting help 7
Comments 7

Chapter 1: Hitachi Content Platform for cloud scale overview 8

About HCP for cloud scale 8
System scaling 8
Single-instance systems versus multi-instance systems 9
About master and worker instances 10
Services 10
Volumes 11
Updating HCP for cloud scale 12

Chapter 2: System requirements and sizing 14

Hardware requirements 14
Operating system and Docker requirements 14
Docker considerations 15
SELinux considerations 16
Virtual machine host requirements 16
Networking 16
Time source requirements 22
Supported browsers 22

Chapter 3: Installing HCP for cloud scale 23

Items and information you need 23
HCP for cloud scale installation steps 23
Decide how many instances to deploy 24
Configure your networking environment 24
(Optional) Select master instances 24
Install Docker on each server or virtual machine 25

Configure Docker on each server or virtual machine....25
(Optional) Install Docker volume drivers....26
(Optional) Enable or disable SELinux on each server or virtual machine....26
Configure maximum map count setting....27
Configure the firewall rules on each server or virtual machine....27
Run Docker on each server or virtual machine....27
Unpack the installation package....27
(Optional) Reconfigure network.config on each server or virtual machine....29
(Optional) Reconfigure volume.config on each server or virtual machine....30
Run the setup script on each server or virtual machine....32
Start the application on each server or virtual machine....35
(Optional) Configure NTP....35
Use the service deployment wizard....35
(Optional) Configure networks for services....37
(Optional) Configure volumes for services....38
(Optional) Check the created volumes....41
(Optional) Distribute services among system instances....42
Moving and scaling floating services....42
Moving and scaling services with multiple types....42
Recommendations....42
Considerations....42
Configuring the service relocation operations manually....42
Configure the system for your users....43

Appendix A: Logs and diagnostic information....44

Appendix B: Services list....48

Appendix C: Handling network changes....68
Safely changing an instance IP address....68
After a network change....68

Preface

This document contains instructions for installing 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.

This document refers to other documents that you should have available before beginning the build process.

Intended audience

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

Product version

This document applies to release 1.0.0 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

The following documents contain additional information about HCP for cloud scale:

- *Hitachi Content Platform for Cloud Scale Object Storage Management Application Help* (MK-HCPCS000-00): This Help system explains how to use the HCP for cloud scale Object Storage Management application to configure and operate a common object storage interface for clients to interact with.
- *Hitachi Content Platform for Cloud Scale System Management Application Help* (MK-HCPCS001-00): This Help system contains the instructions for using the HCP for cloud scale System Management application to 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 Copyrights and Third-party Licenses* (MK-HCPCS003-00): This document contains copyright and license information for third-party software distributed with or embedded in the HCP for cloud scale operating system, core software, and applications.
- *Hitachi Content Platform for Cloud Scale Release Notes* (RN-HCPCS004-00): 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.

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 Corporation.

Thank you!

Chapter 1: Hitachi Content Platform for cloud scale overview

This chapter introduces Hitachi Content Platform for cloud scale (HCP for cloud scale) and explains 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 an HCP for cloud scale system scales by adding or removing instances to the system and also by specifying which services run on those instances.

Instances

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

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

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

For information on instances, see About master and worker instances (on page 10) .

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.

For information on services, see Services (on page 10) .

Single-instance systems versus 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 the minimum RAM, CPU, and disk space requirements. For information on the requirements, see Hardware requirements (on page 14) .

One instance

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

However, a single-instance system has these drawbacks:

- It has a single point of failure. If the instance hardware fails, you lose access to the system.
- With no additional instances, you cannot choose where to run services. All services run on 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 48) .
- 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 System Management Help, which is available from the System Management application.

Four-instance system considerations

The minimum recommended 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 10).

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 above. For information on services, see Services.

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 Metadata 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

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

- **HCP for cloud scale services** — Enable product functionality. For example, the Metadata Gateway service stores metadata persistently. You can scale, move, and reconfigure these services.
- **System services** — Maintain the health and availability of the system. You cannot scale or move these services.

Some system services run only on master instances. See About master and worker instances (on page 10).

For a complete list of services, see Services list (on page 48).

Applications

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

Service instances

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

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

You can configure where HCP 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 once a system is running. For information on configuration, see Networking (on page 16).

Storage for services

Services can use volumes for storing data. For information on volumes, see Volumes (on page 11).

Volumes

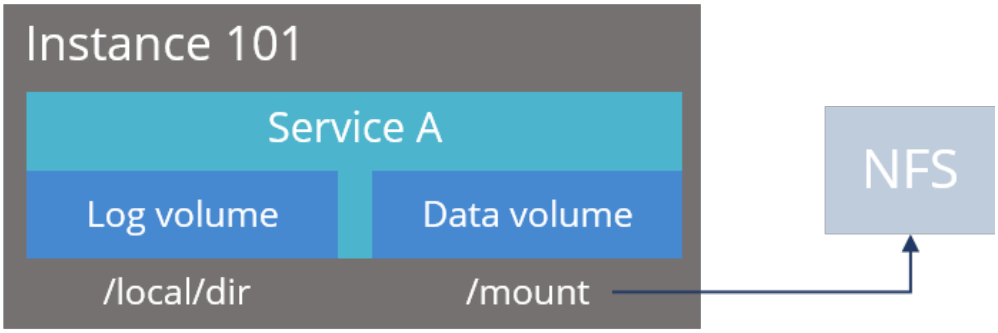
Volumes are properties of services that specify where and how a service stores its data.

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 may 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 directory on the system instance and the service's Data volume stores data in an NFS mount.



Creating and managing volumes

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

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

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

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

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 HCP for cloud scale system, volume drivers are provided by Docker and other third-party developers, not by the HCP for cloud scale 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 HCP for cloud scale installation directory on each instance where the service runs.

For information about volume drivers, see the Docker documentation.

For information about services, see Services (on page 10) .

Updating HCP for cloud scale

You can update system software by installing an update package through the System Management application. For more information, see the System Management Help, which is accessible from the System Management application.

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

Chapter 2: System requirements and sizing

This section lists 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 recommended hardware requirements for each instance in an HCP for cloud scale system.

Resource	Minimum	Recommended
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 Port 8000 for Admin App GUI Port 9084 for MAPI and Storage Management App GUI	Same
Internal IP Ports	See Networking (on page 16)	Same
Network Time	IP address of time service (NTP)	Same

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

Operating system and Docker requirements

To be an HCP for cloud scale instance, each server or virtual machine you provide:

- Must run a 64-bit Linux distribution
- Must have Docker version 1.13.1 or later installed
- Must be configured with IP and DNS addresses

Additionally, you should install all relevant patches on the operating system and perform appropriate security hardening tasks.

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

This table shows the operating systems and Docker and SELinux configurations with which HCP for cloud scale has been qualified.

Operating System	Docker Version	Docker Storage Configuration	SELinux setting
Fedora 27	Docker 18.03.0-ce	direct-lvm	Enforcing
Red Hat Enterprise Linux 7.4	Docker 18.03.0-ce	direct-lvm	Enforcing
Ubuntu 16.04-LTS	Docker 17.03.0-ce	aufs	N/A
CentOS 7.4	Docker 18.03.1-ce	overlay	Enforcing

Docker considerations

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

  After installing HCP for cloud scale, changing the Docker storage driver requires a reinstallation of HCP for cloud scale.

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

  ```
  docker info
  ```

- If you want to enable SELinux on the system instances, you need to use a Docker storage driver that supports it. The storage drivers that SELinux supports differ depending on the Linux distribution you're using. For more information, see the Docker documentation.
- If you are using the Docker `devicemapper` storage driver:
  - Make sure that there's at least 40 GB of Docker metadata storage space available on each instance. HCP for cloud scale requires 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, HCP for cloud scale may 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 HCP for cloud scale.

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

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

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

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

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. If you want to deploy multiple instance on the same hardware, you need to 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 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.

Site hostname

The HCP for cloud scale site hostname is configured during installation. This hostname is very important because it is required 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, see Handling network changes (on page 68) .

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 may want to isolate the traffic for most system services to a secured internal network that has limited access. You would then leave the following services on your external network for user access:

- Admin-App
- Metadata-Cache
- Metadata-Coordination
- Metadata-Gateway
- Metadata-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. If you want to use both types, every instance in your system must be addressable by two IP addresses; one on your internal network, and one on your external network. If you use only one network type, each instance needs only one IP address.

Allowing access to external resources

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

- The 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 the services to use different ports, or use the default values shown in the tables below. For information, see (Optional) Reconfigure network.config on each server or virtual machine (on page 29).

External ports

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

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

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

Note: Debugging ports are accessible only when `debug` is set to `true` in *`install_path`*`/hcpcs/config/cluster.config`

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

Internal ports

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

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

You can find more information about how these ports are used in the documentation for the third-party software underlying each service, see Services (on page 10).

Default Port Value	Used By	Purpose
2181	Synchronization	Primary port used to communicate with the service
2888	Synchronization	Server-server communication
3888	Synchronization	Leader elections
5000	Synchronization	Debugging
5001	Admin App	Debugging
5004	Watchdog	Debugging
5007	Sentinel	Debugging
5050	Cluster Coordination	Primary port used to communicate with the master service
5051	Cluster Worker	Primary port used to communicate with the worker service
5555	Watchdog	Primary port used for JMX inter-service communication
5778	Tracing Agent	Agent HTTP port
6831	Tracing Agent	UDP port
7000	Cassandra	TCP port for database commands and data
7199	Cassandra	Used for database JMX connections
7203	Kafka	Used for message queue JMX connections
8005	Admin App	Tomcat shutdown port
8007	Sentinel	Tomcat shutdown port
8022	Watchdog	SSH
8080	Service Deployment	Primary port used to communicate with the service
8081	Chronos	Primary port used to communicate with the scheduling service
8889	Sentinel	Primary port used to communicate with the service
9042	Cassandra	Primary port used to communicate with the database service

Default Port Value	Used By	Purpose
9091	Network Proxy	Primary port used to communicate with the HA proxy service
9092	Kafka	Primary port used to communicate with the message queue service
9190	OAuth	OAuth port
9191	Metrics	Primary port used to communicate with the service
9200	Elasticsearch	Used to communicate with Elasticsearch cluster
9201	Elasticsearch	Used to communicate with Elasticsearch nodes
9301	Elasticsearch	Elasticsearch intercluster communication
9600	Logstash	Primary port used to communicate with the logging service
9601	Logstash	Port used to listen for syslog connections
9750	S3 Gateway	Support
9751	Metadata Gateway	Support
9752	MAPI Gateway	Support
9753	Metadata Cache	Support
9758	Metadata Policy Engine	Support
9760	Metadata Coordination	Support
9990	S3 Gateway	Remote monitoring
9991	Metadata Gateway	Monitoring
9992	MAPI Gateway	Monitoring
9993	Metadata Cache	Monitoring
9998	Metadata Policy Engine	Monitoring
10000	Metadata Coordination	Monitoring
12000	S3 Gateway	Debugging
12001	Metadata Gateway	Debugging
12002	MAPI Gateway	Debugging

Default Port Value	Used By	Purpose
12003	Metadata Cache	Debugging
12004	Metrics	Debugging
12005	Tracing Collector	Debugging
12006	Tracing Query	Debugging
12007	Tracing Agent	Debugging
12008	Metadata Policy Engine	Debugging
12010	Metadata Coordination	Debugging
12500	Metadata Gateway	Raft RPC Communication
12501	Metadata Gateway	Metadata RPC Communication
12510	Metadata Coordination	RPC communication
12520	Metadata Policy Engine	RPC communication
13300	Metadata Cache	Cache TCP discovery
13370	S3 Gateway	Cache TCP communication
13371	Metadata Gateway	Cache TCP communication
13372	MAPI Gateway	Cache TCP communication
13373	Metadata Cache	Cache TCP communication
13378	Metadata Policy Engine	Cache TCP communication
13380	Metadata Coordination	Cache TCP communication
13453	Metadata Cache	Cache TCP communication
13500	S3 Gateway	Cache client connector
13501	Metadata Gateway	Cache client connector
13502	MAPI Gateway	Cache client connector
13503	Metadata Cache	Cache client connector
13508	Metadata Policy Engine	Cache client connector
13510	Metadata Coordination	Cache client connector
14267	Tracing Collector	Collecting thrift spans from tracing agents
14268	Tracing Collector	HTTP port

Default Port Value	Used By	Purpose
15050	Cluster Coordination	Local port to which the service directly binds
16686	Tracing Query	HTTP port (APIs and user interface)
18000	Admin App	Local port to which the service directly binds
18080	Service Deployment	Local port to which the service directly binds
18889	Sentinel	Local port to which the service directly binds
31000 to 34000	Service Deployment	Port range used by both Service Deployment and Docker for running containers
47000	Cache	TCP cache communication
47008	Metadata Policy Engine	TCP cache communication
47500	Cache	TCP cache discovery
48000	Cache	TCP connector
48500	Cache	Client connector
48508	Metadata Policy Engine	Client connector

Time source requirements

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

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

This chapter describes how to install the HCP for cloud scale software.

Once you've installed the software, you 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 directory path to the HCP for cloud scale directory as *install_path*. The recommended directory 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 24)
2. Configure your networking environment (on page 24)
3. (Optional) Select master instances (on page 24)
4. Install Docker on each server or virtual machine (on page 25)
5. Configure Docker on each server or virtual machine (on page 25)
6. (Optional) Install Docker volume drivers (on page 26)
7. (Optional) Enable or disable SELinux on each server or virtual machine (on page 26)
8. Configure maximum map count setting (on page 27)
9. Configure the firewall rules on each server or virtual machine (on page 27)
10. Run Docker on each server or virtual machine (on page 27)
11. Unpack the installation package (on page 27)
12. (Optional) Reconfigure network.config on each server or virtual machine (on page 29)
13. (Optional) Reconfigure volume.config on each server or virtual machine (on page 30)
14. Run the setup script on each server or virtual machine (on page 32)
15. Start the application on each server or virtual machine (on page 35)
16. (Optional) Configure NTP (on page 35)

17. Use the service deployment wizard (on page 35)
    a. (Optional) Configure networks for services (on page 37)
    b. (Optional) Configure volumes for services (on page 38)
18. (Optional) Check the created volumes (on page 41)
19. (Optional) Distribute services among system instances (on page 42)
20. Configure the system for your users (on page 43)

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 will 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 will be master instances.
2. Make note of the master instance IP addresses.

Tip: 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, check whether Docker 1.13.1 or later is installed:
   `docker -version`
2. If Docker is not installed or if you have a version prior to 1.13.1, install the current Docker version suggested by your operating system.

   The installation method you use depends on your operating system. See the Docker website for instructions.

Configure Docker on each server or virtual machine

Before installing HCP for cloud scale, 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 directory on each instance has at least 20 GB available for storing the HCP for cloud scale Docker images.
2. Ensure that the Docker storage driver is configured correctly on each instance.

   After installation, changing the Docker storage driver requires reinstallation of HCP for cloud scale.

   To view the current Docker storage driver on an instance, run: `docker info` .

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

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

   HCP for cloud scale requires 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, HCP for cloud scale may 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 (that is, 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 may fail to run on the new instance.
2. If you want any services on your system to use Docker volume drivers for storing data (that is, 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 installing HCP for cloud scale.

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

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 installation package `hcpcs-`*`version_number`*`.tgz` and the MD5 checksum file `hcpcs-`*`version_number`*`.tgz.md5` and store them in a directory on the server or virtual machine.
2. Verify the integrity of the installation package:
   `md5sum -c hcpcs-`*`version_number`*`.tgz.md5`
   If the package integrity is verified the command displays `OK`.
3. In the largest disk partition on the server or virtual machine, create a directory named *`install_path`*`/hcpcs`:
   `mkdir` *`install_path`*`/hcpcs`
4. Move the installation package from the directory where you stored it to the directory *`install_path`*`/hcpcs`:
   `mv hcpcs-`*`version_number`*`.tgz` *`install_path`*`/hcpcs/hcpcs-`*`version_number`*`.tgz`
5. Navigate to the installation directory:
   `cd` *`install_path`*`/hcpcs`
6. Unpack the installation package:
   `tar -zxf hcpcs-`*`version_number`*`.tgz`
   A number of directories are created within the installation directory.

   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 may have been packed more than once during download. Use the following commands to fully extract the file:

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

7. Run the installation script `install`, located within a directory matching the version number of system software used by the HCP for cloud scale software.
   `sudo ./cluster/`*`sys_ver_num`*`/bin/install`
   This version number is different from the HCP for cloud scale version number. It is the only subdirectory in the directory `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 directory.
   - The installation script can be run only once 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: If you want to reconfigure networking for the System services, you must complete this step before Run the setup script on each server or virtual machine (on page 32) .

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

If you want 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:

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

     For example:

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

   - Port number assignments:

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

     For example:

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

2. Enter 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

Important: If you want to reconfigure volumes for the System services, you must complete this step before Run the setup script on each server or virtual machine (on page 32) .

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 (that is, each service uses the bind-mount option). If you want 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:

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

  For example, these lines describe the volume that the Admin-App service uses for storing its logs:

```
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.

     Important: 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. Invalid options or values could 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	
`-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`
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. For information, see Run Docker on each server or virtual machine (on page 27) .

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, enter 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`
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 case of a server reboot or other availability event.

Example

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

- You can run the script in the foreground:

  `sudo install_path/hcpcs/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 hcpcs.service file to the appropriate location for your OS. For example:

     `cp install_path/hcpcs/bin/hcpcs.service /etc/systemd/system`

  2. Enable and start the hcpcs.service service:

     ```
     sudo systemctl enable hcpcs.service
     sudo systemctl start hcpcs.service
     ```

(Optional) Configure NTP

If you are installing a multi-instance system:

Procedure

1. Configure NTP (network time protocol) to have each instance use 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 HCP for cloud scale 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, enter the cluster host name or IP address in the **Cluster Hostname/IP Address** field and 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** panel, 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 want 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 16).

   - For information on networking settings for the HCP for cloud scale services, see (Optional) Configure networks for services (on page 37).
   - For information on storage volumes used for services, see (Optional) Configure volumes for services (on page 38).

   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 the services are started.
7. When the wizard finished, the message "Setup Complete" is displayed. Click **Finish**.
   The **Applications** page opens.

Result

The services 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 may 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 directory *`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 Unpack the installation package (on page 27).

(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 and 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 have the volume not 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.

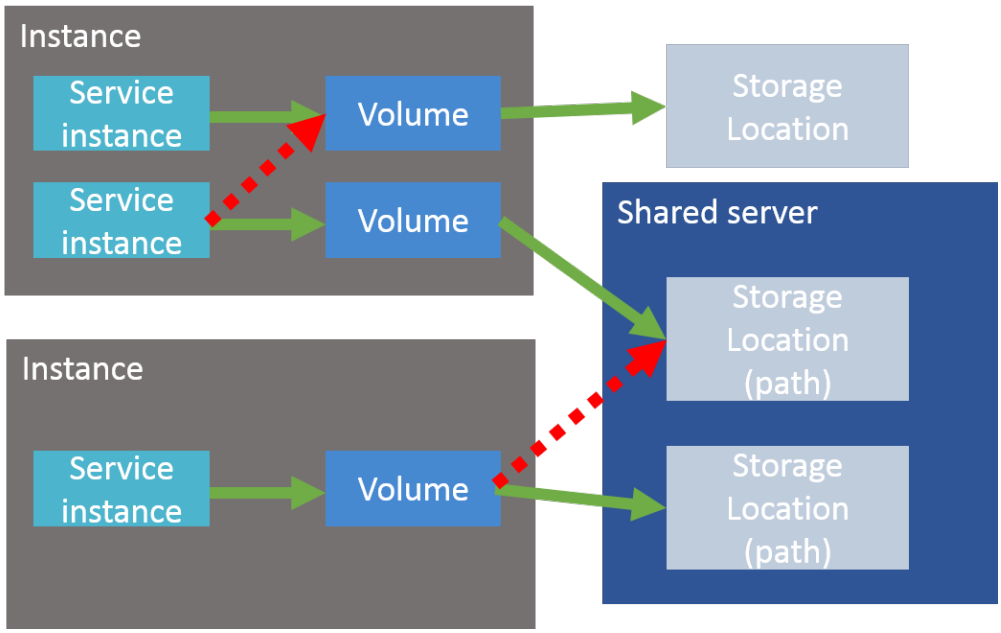
   b. In the **Volume Driver Options** section, in the **Option** and **Value** fields, specify any optional parameters and their corresponding values for the volume driver:

      - 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 directory.
      - If you're using a volume driver:
        i. Click on 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 on 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 could be a filesystem 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. See **Available variables**.

  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 directory creation. If you're using such a volume driver, you need to create volume directories yourself. For an example of how to handle this, see the Docker local volume driver example below.

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

- Make sure the options and values you specify are valid. Invalid options or values could 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`.

Available variables

You can include these variables when configuring volume options:

- `${install_dir}` is the product installation directory.
- `${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.

Example: bind-mount configuration 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, once 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*

Example: Docker local volume driver for Database service log volume

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

i. Log in to your NFS server.
ii. Create a directory.
iii. Within that directory, create one directory 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 directories yourself because the **local** storage driver will not create them automatically.

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

v. Specify these options and values:

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

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

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

(Optional) Check the created volumes

Before you begin

If you configured the service volumes to use volume drivers in (Optional) Configure volumes for services (on page 38), 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 directory `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 Unpack the installation package (on page 27).

(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 recommended 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-utilized. 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. For information on floating services, see Services (on page 10) .

Moving and scaling services with multiple types

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

Recommendations

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

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

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

Configuring the service relocation operations manually

To manually reconfigure a service relocation operation, in the Admin App:

Procedure

1. Select **Services**.
2. Locate a service that you want to scale or move and click **Configure**.
3. On the **Scale** tab, if the service has more than one type, select the instance type that you want to scale.
4. If the service is a floating service, you are presented with options for configuring an instance pool:
   a. In the **Service Instances** field, specify the number of instances on which the service should be running at any time.
   b. Configure the instance pool:
      - To have the service run on any instance in the system, select the **All Available Instances** option.

        With this option, the service can be restarted on any instance, including instances that were added to the system after the service was configured.
      - To have the service run on a specific set of instances, deselect the **All Available Instances** option. Then:
        - To remove an instance from the pool, select it from the **Instance Pool** list on the left and click **Remove Instances**.
        - To add an instance to the pool, select it from the **Available Instances** list on the right and click **Add Instances**.
5. If the service is a non-floating service, you are presented with options for selecting the specific instances that the service should run on. Do one or both of these, then click **Next**:
   - To remove the service from the instances it's currently on, select one or more instances from the list on the left and click **Remove Instances**.
   - To add the service to other instances, select one or more instances from the **Available Instances** list on the right and click **Add Instances**.
6. Click **Update Service**.

Configure the system for your users

Once 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.

Procedure

1. Set an IDP and creating user accounts.
2. Define storage objects.
3. Obtain S3 authorization credentials.

Appendix A: Logs and diagnostic information

Each service maintains its own set of logs. By default, the logs are maintained in the *install_path*/hcpcs/log directory 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 if you want to. That is, you can delete or archive them as necessary.

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

System logs are managed automatically in these ways:

- All log files are periodically added to a compressed file and moved to *install_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 directory 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 directory *`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 directory at this path:

`install_path``/hcpcs/log`

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

For information about the services, see Services list (on page 48) .

Default log directory 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	Client access data service.

Default log directory 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	Agent service
com.hitachi.aspen.foundry.service.tracing.collector	Tracing-Collector	Collector service
com.hitachi.aspen.foundry.service.tracing.query	Tracing-Query	Query service
com.hds.ensemble.plugins.service.kafka	Message Queue	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	Management API gateway
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.

Default log directory name	Related service	Contains information about
com.hds.ensemble.plugins.service.mesosAgent	Cluster-Worker	Work ordered by the Cluster-Coordination service.
com.hds.ensemble.plugins.service.mesosMaster	Cluster-Coordination	Hardware resource allocation.
com.hitachi.aspen.foundry.service.metadata.policy.engine	Metadata-Policy-Engine	Metadata asynchronous policy engine
com.hitachi.aspen.foundry.service.metrics.prometheus	Telemetry-Service	Telemetry
com.hds.ensemble.plugins.service.sentinel	Sentinel	Internal system processes.
com.hds.ensemble.plugins.service.watchdog	Watchdog	General diagnostic information.
com.hds.ensemble.plugins.service.zookeeper	Synchronization	Coordination of actions and database operations across instances.

Appendix B: Services list

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

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

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

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties
Product services: These services perform 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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	2.4 GB
	Service Options ▪ **Max heap:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 1800m. ▪ **Heap new size:** Heap size for the Java virtual machine. Valid values are integers representing a number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 512 MB.	Number of instances:	Required: 3 Recommended: All

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	Advanced Options: **Compaction Frequency:** How often database compaction operations are run. 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 units:	10
Chronos Job scheduling.	**Container Options: Default** ▪ **Container Memory:** The hard memory limit for the service's Docker container, in MB. The default is 712 MB. ▪ **CPU:** The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	712 MB
	Service Options	Number of instances:	▪ 1 required ▪ 1 recommended
		Persistent or floating?	Floating
		Supports volume configuration?	Yes

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. Enter an integer 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 units:	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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	2 GB
	Service Options ▪ **Heap settings:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 1024 MB.	Number of instances:	Required: 3 Recommended: All
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **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.	Single or multiple types?	Single
		Scalable?	Yes
		Service units:	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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	2 GB
	Service Options	Number of instances:	Required: 3 Recommended: 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. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is -Xmx1800m-Xms512m.	Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	Yes
		Service units:	5
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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	700 MB
	Service Options	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **Max heap:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 512 MB.	Single or multiple types?	Single
		Scalable?	No
MAPI Gateway Serves MAPI endpoints.	**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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 64 MB.	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Single or multiple types?	Single
		Scalable?	Yes
		Service units:	5
Metadata Cache Cache for HCP for cloud scale system metadata.	**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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 512 MB.	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service units:	10

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Metadata Coordination Coordinates the different metadata gateway services, and coordinates metadata partitions.	**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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 64 MB.	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 2048 MB.	Number of instances:	Required: 3 Recommended: 3
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service units:	50

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Metadata Policy Engine Executes 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. ▪ **CPU:** The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 1024 MB.	Number of instances:	Required: 1 Recommended: 1
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
		Service units:	25

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Metrics 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 768 MB. ▪ **CPU:** The relative CPU usage weight for the service's Docker container. Generally, a higher value means that the container received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Prometheus Scrape Interval:** The time interval between runs of the metrics collection task. Enter an integer number of seconds. You can optionally specify the suffix s (seconds). The default is 10 seconds. ▪ **Prometheus Database Path:** Storage location for prometheus local time-series db. Enter a path. The default is tsdb/. ▪ **Prometheus Database Retention:** The number of days to retain files. Enter an integer number of days. You can optionally specify the suffix **d** (days). The default is 15 days.	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service units:	10

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
S3 Gateway Serves S3 API endpoints 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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 512 MB.	Number of instances:	Required: 1 Recommended: All
	HTTP Options: ▪ **Enable HTTP:** Select to enable HTTP connections. ▪ **Max Http Request Headers:** The maximum number of HTTP request headers to allow. Enter an integer. The default is 100 request headers.	Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Scalable?	Yes
		Service units:	25
Tracing Agent Provides 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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **Collector TChannel Hostname:** Enter a host name. The default is localhost. ▪ **Collector TChannel Port:** Enter a port number. The default is 14267.	Number of instances:	Required: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	Yes
		Service units:	1

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Tracing Collector Provides 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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **ElasticSearch Hostname:** Enter a host name. The default is localhost. ▪ **ElasticSearch Port:** Enter a port number. The default is 9200. ▪ **Sampling Rate:** The sampling rate for all clients implementing remote sampling. Enter a number between 0 and 1 inclusive. The default is 1.	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No
		Single or multiple types?	Single

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **Max open Index age:** How long to keep tracing indexes open in the database, in days. Enter 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. Enter a value from 1 to 365 days inclusive. The default is 60 days.	Scalable?	Yes
Tracing Query Provides 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 received more CPU resources than other processes (including other service Docker containers) running on the instance. Enter a decimal number. The default is 0.1.	RAM needed per instance:	768 MB
	Service Options ▪ **ElasticSearch Hostname:** Enter a host name. The default is localhost. ▪ **ElasticSearch Port:** Enter a port number. The default is 9200.	Number of instances:	Required: 1 Recommended: All
		Persistent or floating?	Floating
		Supports volume configuration?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Single or multiple types?	Single
		Scalable?	Yes
System services: These services manage system resources and ensure that the system remains available and accessible. These services cannot be moved or reconfigured.			
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. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 512 MB.	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Cluster Coordination Manages hardware resource allocation.	N/A	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	No
		Single or multiple types?	Single
		Scalable?	No
Cluster Worker	N/A	RAM needed per instance:	N/A

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
Receives and performs work from other services.		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
		Service units:	5
Network Proxy Network request load balancer.	**Security Protocol:** Select which TLS versions to use: ▪ **TLS 1.0** ▪ **TLS 1.1** ▪ **TLS 1.2** ▪ **TLS 1.3**	RAM needed per instance:	N/A
	SSL Ciphers: If you want to provide your own cipher suite, enter it here.	Number of instances:	N/A
	Custom Global Configuration: Select **Enable Advanced Global Configuration** to enable adding your own parameters to the HAProxy "global" section.	Persistent or floating?	Persistent
	Custom Defaults Configuration: Select **Enable Defaults Configuration** to enable adding your own parameters to the HAProxy "global" section.	Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
		Service units:	1
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. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 256 MB.	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Service Deployment Handles deployment of high-level services (that is, the services that you can configure).	N/A	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Single
		Scalable?	No
Synchronization Coordinates service configuration settings and other information across instances.	**Service Options**	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent

Service name and description	Configuration settings (changes cause the service to redeploy)	Properties	
	▪ **Max Heap Size:** Maximum amount of memory to allocate to the Java heap for each instance of the service. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 1024 MB.	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. Enter an integer number of bytes. You can optionally specify the suffixes **k** (kilobytes), **m** (megabytes), or **g** (gigabytes). The default is 1024 MB. ▪ **Heap new size:** Heap size for the Java virtual machine. Valid values are integers representing a number of bytes. You can optionally specify the suffixes k (kilobytes), m (megabytes), or g (gigabytes). The default is 512 MB.	RAM needed per instance:	N/A
		Number of instances:	N/A
		Persistent or floating?	Persistent
		Supports volume configuration?	Yes
		Single or multiple types?	Yes
		Scalable?	No
		Service units:	5

Appendix C: Handling network changes

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

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

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

Safely changing an instance IP address

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

For each instance whose IP address you need to change:

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 42) .
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 directory 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 directory.
   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