Hitachi Content Intelligence

Installing Hitachi Content Intelligence

This document contains information on installing an HCI system, either on physical servers or virtual machines that you provide.

MK-HCI002-16
April 2022

Legal notice

Contents

Preface

This book is the installation guide for HCI. It describes how to install both multi and single-instance HCI systems that power either the HCM or Hitachi Content Search use cases.

Please read this document carefully to understand how to install these products, and maintain a copy for your reference.

Intended audience

This document contains information on installing an HCI system, either on physical servers or virtual machines that you provide.

Product version

This document applies to Hitachi Content Intelligence 1.6 or later.

Release notes

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

Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click **OK**. ▪ Indicates emphasized words in list items.

Convention	Description
Italic	▪ Indicates a document title or emphasized words in text. ▪ Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: `pairdisplay -g group` (For exceptions to this convention for variables, see the entry for angle brackets.)
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`
< > angle brackets	Indicates variables in the following scenarios: ▪ Variables are not clearly separated from the surrounding text or from other variables. Example: `Status-<report-name><file-version>.csv` ▪ Variables in headings.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.

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

Icon	Label	Description
	Note	Calls attention to additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Important	Highlights information that is essential to the completion of a task.

Icon	Label	Description
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	CAUTION	Warns the user of a hazardous situation that, if not avoided, could result in major or minor injury.
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Conventions for storage capacity values

Physical storage capacity values (for example, disk drive capacity) are calculated based on the following values:

Physical capacity unit	Value
1 kilobyte (KB)	1,000 (10^3) bytes
1 megabyte (MB)	1,000 KB or $1{,}000^2$ bytes
1 gigabyte (GB)	1,000 MB or $1{,}000^3$ bytes
1 terabyte (TB)	1,000 GB or $1{,}000^4$ bytes
1 petabyte (PB)	1,000 TB or $1{,}000^5$ bytes
1 exabyte (EB)	1,000 PB or $1{,}000^6$ bytes

Logical capacity values (for example, logical device capacity, cache memory capacity) are calculated based on the following values:

Logical capacity unit	Value
1 block	512 bytes
1 cylinder	Mainframe: 870 KB Open-systems: ▪ OPEN-V: 960 KB ▪ Others: 720 KB
1 KB	1,024 (2^{10}) bytes
1 MB	1,024 KB or $1{,}024^2$ bytes

Logical capacity unit	Value
1 GB	1,024 MB or $1,024^3$ bytes
1 TB	1,024 GB or $1,024^4$ bytes
1 PB	1,024 TB or $1,024^5$ bytes
1 EB	1,024 PB or $1,024^6$ bytes

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

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

Hitachi Vantara Community 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!

HCI Community

For HCI-specific product support, discussions, announcements, and FAQs, visit us at the HCI Community portal: https://community.hitachivantara.com/s/hitachi-content-intelligence

Chapter 1: Overview

This chapter introduces Hitachi Content Intelligence (HCI) and its main use cases: Hitachi Content Search and Hitachi Content Monitor (HCM).

A single HCI system can be installed for only one of these use cases.

About Hitachi Content Search

Hitachi Content Intelligence (HCI) powers Hitachi Content Search, a full-fledged search and data processing solution. It handles all steps in making your data searchable, regardless of where that data lives or what formats it's in. HCI also gives users tools for examining, understanding, normalizing, migrating, and editing their data.

System scaling

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

Instances

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

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

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

Services

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

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

Single-instance systems vs. multi-instance systems

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

Note:

- Every instance must meet the minimum RAM, CPU, and disk space requirements.
- Three instances are sufficient to perform leader election for distributing work. However, a multi-instance system needs a minimum of four instances because, with the minimum hardware requirements, three instances are not sufficient for running all HCI services at their recommended distributions.
- Hitachi Vantara has qualified HCI systems with up to 16 instances.

One instance

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

However, a single-instance system has these drawbacks:

- Only 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.
- 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.

  Note: For a search index to survive an instance outage:

  - The system must have at least two instances running the Index service.
  - The Index Protection Level for the index must be at least 2.

  For more information, see the *HCI Administrator Help*, which is available in the Admin App.
- 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 Content Intelligence 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 HCI system, see the Content Intelligence Administrator Help, which is available from the Admin App.

Two-instance system considerations

Two-instance systems are a viable option for the HCM use case, but not recommended for Hitachi Content Search.

Three-instance system considerations

Three-instance systems should have only a single master instance. If you deploy a three-instance system where all three instances are masters, the system might not have enough resources to do much beyond running the master services.

About master and worker instances

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

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

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

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

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

Services

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

Service categories

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

- Services: Enable product functionality. For example, the Index service performs functions that allow the system to be used to search for data. You can scale, move, and reconfigure these services.
- System services: Maintain the health and availability of the system. You cannot scale, move, or reconfigure these services.

  Some System services run only on master instances.

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 Hitachi Content Intelligence services run, but not system services.

Services with multiple types

Some services can have multiple service instance types. That is, a service can run on two system instances, but those two service instances can perform different functions from one another.

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.

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.

Storage for services

Services can use volumes for storing data.

Jobs

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

Jobs are run by services; you cannot start or stop them yourself on demand, but you can schedule the times when they are allowed to run and specify which instances in the system that they are allowed to run on.

Workflow jobs

Beginning with release 1.3, each HCI workflow is associated with a job. Running the workflow causes its job to run and process documents.

Job types

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

Workflow-Agent job type

HCI has a single type of job, the Workflow-Agent job type. Jobs of this type are run to perform:

- A single workflow task.
- A pipeline test.
- A workflow test.
- Tasks to restart workflow failures.

Storage for jobs

You can configure storage usage for jobs by associating volumes with job types.

Volumes

WARNING: When mounting and unmounting directories on systems with HCI installed, do not use the `-a` option, as it may cause unintended performance and functionality issues.

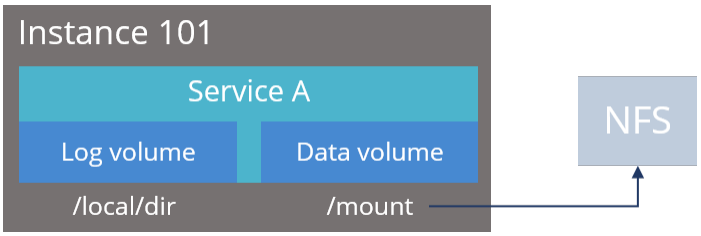
Volumes are properties of services that specify where and how a service stores its data.

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

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

Example

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



Important: In order for the connectors in the Workflow Designer App to run correctly when pipelines are executed, mount points for the associated drives need to be created *before* starting HCI.

Creating and managing volumes

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

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

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

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

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

Volume drivers

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

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

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

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

Note: HCI has been qualified with these volume drivers:

- local: The default Docker volume driver
- local-persist: A Docker volume driver plugin available from https://github.com/CWSpear/local-persist

Updating HCI

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.

Update consists of multiple steps and might take several hours to complete. During this time:

- Multiple varities of `Loading` and `Reconnecting` messages will appear.
- The window or its progress might appear stalled or stuck.
- `Severe` and `Warning` events might occur.

This is typical update and deployment behavior. You will be notified when the process has officially completed.

Important:

- Hitachi Vantara does not provide updates or security fixes for the host operating systems running on HCI instances.
- During Update and Deployment, if you're installing the Monitor-App service, each specific signal needs a different set of ports and protocol. For reference on which ports to use, see System ports for Monitor-App (on page 33).

Chapter 2: System requirements and sizing

The hardware, networking, and operating system requirements for running an HCI system with one or more instances.

Sizing guidance for Hitachi Content Search

Simple sizing

This table shows the minimum and recommended hardware requirements for each instance in an HCI running Hitachi Content Search.

Resource	Minimum	Recommended
RAM	16 GB	32 GB
CPU	4-core	8-core
Available disk space	50 GB	500 GB

Important:

- A large number of factors determine how many documents your system can index and how fast it can process them, including: the number of documents to be indexed; the contents of those documents; what search features (such as sorting) the index supports; the number of fields in the index; the number of users querying the system; and so on.

  Depending on how you use your system, you might require additional hardware resources to index all the documents you want and at the rate you require.
- Each instance uses all available RAM and CPU resources on the server or virtual machine on which it's installed.

Detailed sizing

If you are installing HCI to run Hitachi Content Search, you should size your system based on the number of documents you need to index and the rate at which you need documents to be processed and indexed.

Important: This sizing guide details the resources required for a system with a single Index Protection Level (IPL). To scale your system accordingly, you will need to double the recommended values to accommodate IPL 2, triple the recommended values to accommodate IPL 3, etc.

To determine the system size that you need:

Procedure

1. Determine how many documents you need to index.
2. Based on the number of documents you want to index, use the following tables to determine:
   - How many instances you need
   - How much RAM each instance needs
   - The Index service configuration needed to support indexing the number of documents you want

<table>
<tr><th colspan="3">Total documents to be indexed</th><th colspan="2">System configuration</th></tr>
<tr><td>15 million</td><td>25 million</td><td>50 million[a]</td><td>Total instances required: 1[b]</td><td>Instances running the Index service: 1
Index service configuration required:
Shards per index: 1
Index Protection Level per index: 1
Container memory: 200MB greater than Heap settings
Heap settings: Depends on instance RAM.
<table><tr><th>Instance RAM</th><th>Heap setting</th></tr><tr><td>16 GB</td><td>1800m</td></tr><tr><td>32 GB</td><td>9800m</td></tr><tr><td>64 GB</td><td>25800m</td></tr></table></td></tr>
<tr><td>16 GB</td><td>32 GB</td><td>64 GB</td><td colspan="2"></td></tr>
</table>

Total documents to be indexed	System configuration
Instance RAM needed (for each instance running the Index service)	
[a] Contact Hitachi Vantara for guidance before trying to index this many documents on this number of instances. At this scale, your documents and required configuration settings can greatly affect the number of documents you can index. [b] Single-instance systems are suitable for testing and development, but not for production use.	

<table>
<tr><th colspan="3">Total documents to be indexed</th><th colspan="2">System configuration</th></tr>
<tr><td>45 million</td><td>75 million</td><td>150 million[a]</td><td>Total instances required: 4</td><td>Instances running the Index service: 3
Index service configuration required:
Shards per index: 3
Index Protection Level per index: 1
Container memory: 200MB greater than Heap settings
Heap settings: Depends on instance RAM.
<table><tr><th>Instance RAM</th><th>Heap setting</th></tr><tr><td>16 GB</td><td>1800m</td></tr><tr><td>32 GB</td><td>9800m</td></tr><tr><td>64 GB</td><td>25800m</td></tr></table></td></tr>
<tr><td>16 GB</td><td>32 GB</td><td>64 GB</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="3">Instance RAM needed (for each instance running the Index service)</td></tr>
<tr><td colspan="5">[a] Contact Hitachi Vantara for guidance before trying to index this many documents on this number of instances. At this scale, your documents and required configuration settings can greatly affect the number of documents you can index.</td></tr>
</table>

<table>
<tr><th colspan="3">Total documents to be indexed</th><th colspan="2">System configuration</th></tr>
<tr><td>75 million</td><td>125 million</td><td>250 million[a]</td><td>Total instances required: 8</td><td>Instances running the Index service: 5
Index service configuration required:
▪ Shards per index: 5
▪ Index Protection Level per index: 1
▪ Container memory: 200MB greater than Heap settings
▪ Heap[b] settings: Depends on instance RAM.
<table><tr><th>Instance RAM</th><th>Heap setting</th></tr><tr><td>16 GB</td><td>7800m</td></tr><tr><td>32 GB</td><td>15800m</td></tr><tr><td>64 GB</td><td>31000m</td></tr></table></td></tr>
<tr><td>16 GB</td><td>32 GB</td><td>64 GB</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="3">Instance RAM needed (for each instance running the Index service)</td></tr>
<tr><td colspan="5">[a] Contact Hitachi Vantara for guidance before trying to index this many documents on this number of instances. At this scale, your documents and required configuration settings can greatly affect the number of documents you can index.
[b] With an 8-instance system, the Index service should be the only service running on each of its 5 instances. With the Index service isolated this way, you can allocate more heap space to the service than you can on a single or 4-instance system.</td></tr>
</table>

Total documents to be indexed			System configuration	
195 million	325 million	650 million[a]	Total instances required: 16	Instances running the Index service: 13

<table>
<tr><th colspan="3">Total documents to be indexed</th><th colspan="2">System configuration</th></tr>
<tr><td colspan="3"></td><td></td><td>Index service configuration required:
Shards per index: 13
Index Protection Level per index: 1
Container memory: 200MB greater than Heap settings
Heap[b] settings: Depends on instance RAM.
<table><tr><th>Instance RAM</th><th>Heap setting</th></tr><tr><td>16 GB</td><td>7800m</td></tr><tr><td>32 GB</td><td>15800m</td></tr><tr><td>64 GB</td><td>31000m</td></tr></table></td></tr>
<tr><td>16 GB</td><td>32 GB</td><td>64 GB</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="3">Instance RAM needed (for each instance running the Index service)</td></tr>
<tr><td colspan="5">[a] Contact Hitachi Vantara for guidance before trying to index this many documents on this number of instances. At this scale, your documents and required configuration settings can greatly affect the number of documents you can index.
[b] With a 16-instance system, the Index service should be the only service running on each of its 13 instances. With the Index service isolated this way, you can allocate more heap space to the service than you can on a single or 4-instance system.</td></tr>
</table>

For example, if you need to index up to 150 million documents, you need at minimum a 4-instance system with 64 GB RAM per instance.

3. Determine how fast you need to index documents, in documents per second.

   For example:

   - To index 100 million documents in 2 days, you need an indexing rate of 578 documents per second.
   - To continuously index 1 million documents every day, you need an indexing rate of 12 documents per second.

4. Determine the base indexing rate for your particular dataset and processing pipelines:

   a. Install a single-instance HCI system with that has the minimum required hardware resources.
   b. Run a workflow with the pipelines you want and on a representative subset of your data.
   c. Use the workflow task details to determine the rate of documents processed per second.

5. To determine the number of cores you need per instance, replace **Base rate** in this table with the rate you determined in step 4.

Number of instances you need	Cores per instance	
	4 (minimum required)	8 (recommended)
1	Base rate	70% Base rate
4	300% Base rate	500% Base rate
8	600% Base rate	900% Base rate
More than 8	Contact Hitachi Vantara for guidance	

   For example, if you had previously determined that:

   - You need a 4-instance system.
   - You need to process 500 documents per second.
   - The base processing rate for your data and pipelines is 100 documents per second.

   You need 8 cores per instance.

6. Multiply the number of instances you need times the number of cores per instances to determine the total number of cores that you need for your system.
7. After your system is installed, configure it with the index settings you determined in step 2.

   For information on index shards, Index Protection Level, and moving the Index service, see the Administrator Help, which is available from the Admin App.

Sizing guidance for HCM

Minimum hardware requirements

If you are installing HCI to run HCM, each instance in the system must meet these minimum hardware requirements:

Documents per second	Cores	RAM (GB)	Disk (GB)
Up to 1200	8	28	600
1200-1600	12	32	800
1600-2000	16	40	1000
2000-2400	18	48	1400
2400-2800	20	56	1700
2800-3200	24	64	2000

Determining number of instances

The number of instances that your HCM system needs is based on:

- Whether you need the system to remain highly available.
- The number of documents being produced by the HCP system you want to monitor. In this case, each document represents a single piece of data about the HCP system. A more active HCP system will produce more documents than a less active one.
- The total number of documents you want HCM to store.

Number of instances: simple procedure

If you're monitoring a typically-active HCP system (roughly 75 operations per second per node), you can use this table to determine the number of HCM instances you need. This table lists the number of HCM instances you need based on the number of nodes in your HCP system and the number of days that you want your HCM system to retain the data it receives from HCP.

If your system is more active, see Number of instances: detailed procedure (on page 24).

HCP nodes	Data retention time on HCM	Instances needed
Up to 8	Up to 30 days	1*
Up to 8	Up to 60 days	3*
Up to 16	Up to 30 days	4
Up to 24	Up to 60 days	8
*An HCM system must have a minimum of 4 instances to maintain high system availability.		

Number of instances: detailed procedure

Procedure

1. Determine whether you need your HCM system to maintain high availability. If so, you need a minimum of 4 instances. For more information, see Single-instance systems versus multi-instance systems (on page 10).
2. Determine the number of documents per second being produced by the HCP system you want to monitor. You can easily do this if you already have an HCM system up and running:
   a. Go to the Monitor App: `https://system-hostname:6162`
   b. Add the HCP system as a source. For information, see the help that's available from the Monitor App.
   c. Go to the HCI Admin App: `https://system-hostname:8000`
   d. Go to **Workflows > Monitor App Workflow > Task > Metrics**.
   e. View the value for the **Average DPS** field.

   Tip: Let the workflow run for a while to get a more accurate measure for the **Average DPS** field.

   Otherwise, you can get an average documents per second value by doing this:
   a. Select a time period.
   b. Download the HCP **Internal Logs** for this time period. For more information, see the help that's accessible from the HCP System Management Console.
   c. In the downloaded logs for each node, count the number lines logged during the selected time period.
   d. Add the line value for each node and then divide the sum by the number of seconds in the time window you selected.
3. Use this table to determine the number of instances needed based on the number of documents per second produced by your HCP system.

Documents per second	Instances needed
≤ 3,200	1
3,201 to 7,200	3
7,201-10,500*	4
*This is the maximum documents per second that HCM currently supports.	

4. Based on your data availability requirements, determine the number of instances you need.

Data availability requirement	Index replicas needed	Instances needed	Impact on total documents stored
No failure tolerance	1	1	None
Survive 2 failed replicas	3	3	3x
Survive 3 failed replicas	4	4	4x

An index with multiple copies remains available in the event of an instance outage. For example, if an index has two copies stored on two instances and one of those instances fails, one copy of the index remains available for servicing requests.

5. Use this formula to determine the total number of documents your HCM system must be able to store:

   ```
   documents per second from step 2.
   x 3600 seconds in an hour
   x 24 hours in a day
   x number of days you want to store data (default is 30)
   x Impact from the data availability table in step 4.
   = Total document count
   ```

   For example, if your HCP system produces 1500 documents per second, you want to store data for 30 days, and you want to maintain two copies of each index containing the stored data, your system must have enough instances to be able to store roughly 8 billion documents:

   ```
   1500
   x 3600
   x 24
   x 30
   x 2
   = 7,776,000,000
   ```

6. Use this table to determine the number of instances needed based on the total number of documents your HCM must store.

Total document count	Instances needed
2 billion or less	1
6 billion or less	3
8 billion or less	4

7. Take the highest number of instances from steps 2, 3, and 6. That's the number instances you need.

Operating system and Docker requirements

To be an HCI 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.

Suggested Docker version

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 earlier than 1.13.1.

This table shows the operating systems, as well as the Docker and SELinux configurations, on which this HCI system has been qualified for:

Operating system	Docker version	Docker storage configuration	SELinux setting
CentOS 7.6	Docker 18.03.1-ce	device-mapper	Enforcing
CentOS 8.1.1911	Docker 19.03.9-ce	overlay2	Enforcing and Disabled
Red Hat Enterprise Linux 8.1	Docker 19.03.11-ce	overlay2	Enforcing
Ubuntu 18.04.4 LTS	Docker 18.03.1-ce	overlay2	Enforcing

Docker considerations

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

  After you install the product, to change the Docker storage driver you must reinstall the product.

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

  ```
  docker info
  ```

- 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. The product needs 20 GB to install successfully and an additional 20 GB to successfully update to a later version.

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

    ```
    docker info
    ```

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

SELinux considerations

- You should decide whether you want to run SELinux on system instances and enable or disable it before installing additional software on the instance.

  Enabling or disabling SELinux on an instance needs a restart of the instance.

  To view whether SELinux is enabled on an instance, run: `sestatus`
- To enable SELinux on the system instances, you need to use a Docker storage driver that supports it.

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

Networking

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

You can configure the network settings for each service when you install the system. You cannot change these settings after the system is up and running. If your networking environment changes such that the system can no longer function with its current networking configuration, you need to reinstall the system. See Handling network changes (on page 72).

WARNING:

The HCI product uses both internal and external ports to operate its services and the system-internal ports do not have authentication or Transport Layer Security (TLS). At a minimum, use your firewall to make these ports accesible only to other instances in the system. If any users have root access to your system, your network and its systems are vulnerable to unauthorized use.

To secure your data and HCI system, you need to manually use iptables or firewalld to restrict ports to only local communications that the HCI installer otherwise leaves open. See System-internal ports (on page 30) and Example HCI firewall setup (on page 74).

Additionally, you can use Internet Protocol Security (IPSec) or an equivalent to secure internode communications. Consult with your system administrator to configure your network with this added security.

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 72).

Network types

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

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

Allowing access to external resources

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

This includes:

- The data sources where your data is stored.
- Identity providers for user authentication.
- Email servers that you want to use for sending email notifications.
- Any external search indexes (for example, HDDS indexes) that you want to make accessible through HCI.

Ports

Each service binds to a number of ports for receiving incoming traffic. Before installing HCI, you can configure the services to use different ports, or use the default values shown in the following tables.

Port values can be reconfigured during system installation, so your system might not use the default values. You cannot change service port values when the system is up and running.

To view the ports that your system is using, view the Network tab for each service your system runs (Services > `service-name` > Network).

WARNING:

The HCI product uses both internal and external ports to operate its services and the system-internal ports do not have authentication or Transport Layer Security (TLS). At a minimum, use your firewall to make these ports accesible only to other instances in the system. If any users have root access to your system, your network and its systems are vulnerable to unauthorized use.

To secure your data and HCI system, you need to manually use iptables or firewalld to restrict ports to only local commnuications that the HCI installer otherwise leaves open. See System-internal ports (on page 30) and Example HCI firewall setup (on page 74).

Additionally, you can use Internet Protocol Security (IPSec) or an equivalent to secure internode communications. Consult with your system administrator to configure your network with this added security.

System-external ports

Important: To keep your system secure, HCI system-external ports require user authentication and utilize Transport Layer Security (TLS).

The following table contains information about the service ports that are used to interact with the system.

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

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

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

Default Port Value	Service	Purpose
6162	Monitor-App	Access to the HCM application, which is used to monitor the health of HCP systems. **WARNING:** The Monitor-App service will not function properly if it is assigned a port value lower than 1024.
8000	Admin-App	Access to administrative interfaces: ▪ Administration App ▪ Administrative REST API ▪ Administrative CLI
8888	Search-App	Access to search interfaces: ▪ Search App ▪ Workflow Designer ▪ Search REST API ▪ Workflow Designer REST API ▪ Search CLI ▪ Workflow Designer CLI

System-internal ports

This table lists the ports used for intra-system 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 on how these ports are used in the documentation for the third-party software underlying each service.

Note: For a secure and recommended firewall setup using these internet ports, see Example HCI firewall setup (on page 74).

Default Port Value	Used By	Purpose
2181	Synchronization service	Synchronization service client port.
2888	Synchronization service	Synchronization service internal communication.
3888	Synchronization service	Synchronization service leader election.
4040	Workflow jobs	Spark UI port.
5001	Admin-App service	Debug port for Admin-App service.
5005	Workflow jobs	The port to use for debugging the job driver.
5008	Workflow jobs	The port to use for debugging the job executor.
5002	Search-App service	Debug port used by the Search-App service.
5003	Index service	Debug port used by the Index service.
5050	Cluster-Coordination service	Primary port for communicating with Cluster-Coordination.
5051	Cluster-Worker service	Primary port for communicating with Cluster-Worker.
5123	Monitor-App service	The debug port used by the Monitor App.
5555	Watchdog service	Port for JMX connections to Watchdog service.
5601	Dashboard service	Primary port for communicating with the Dashboard service.
6175	Monitor-App service	The port used by the Monitor App for graceful shutdowns.
7000	Database service	TCP port for commands and data.
7199	Database service	Port for JMX connections to Database service.
7203	Message Queue service	Port for JMX connections to Message Queue service.
8005	Admin-App service	Port used by Admin-App for graceful shutdowns.
8006	Search App service	Port used by the Search App service for graceful shutdowns.

Default Port Value	Used By	Purpose
8080	Service-Deployment service	Primary port for communicating with Service-Deployment.
8081	Scheduling service	Primary port for communicating with the Scheduling service. **WARNING:** If you change the port number for the Scheduling service, in order for the changes to take effect, you will need to restart `HCI.service` on all system nodes.
5007	Sentinel service	Debug port used by Sentinel service.
8007	Sentinel service	Port used by the Sentinel service for graceful shutdowns.
8889	Sentinel service	Primary port for communicating with Sentinel.
8893	Monitor-App service	Port used for the Monitor App Analytics functionality.
8983	Index service	Primary port used to communicate with the Index service. **WARNING:** The port assigned to the Index service should not be below 1024.
9042	Database service	Primary port for communicating with the Database service.
9091	Network-Proxy service	Primary port for communicating with Network-Proxy.
9092	Message Queue service	Primary port for communicating with Message Queue service.
9200	Metrics service	Port used to communicate with the Metrics service cluster.
9201	Metrics service	Port used to communicate with an individual Metrics service node.
9301	Metrics service	Port that nodes in the Metrics service cluster should use when communicating with each other.

Default Port Value	Used By	Purpose
9600	Logging service	Primary port for communicating with Logging service.
9601	Logging service	The port used to receive syslog messages.
10000	Index service	Port used by the Index service for graceful shutdowns.
15050	Cluster-Coordination service	Cluster-Coordination internal communication
18000	Admin-App service	Admin-App internal communication.
18080	Service-Deployment service	Service-Deployment internal communication
18889	Sentinel service	Sentinel service internal communication.
31000-34000	Cluster-Coordination and Cluster-Worker services	High ports used by both Mesos and Docker.

System ports for Monitor-App

This table lists the ports used by Monitor-App during the Configuration and Deployment phases. Each signal needs the following port information to function properly:

Monitor-App signal	Port Type	Port Number
Node Status	TCP	443 (or 80 if not using SSL) inbound to HCP
MAPI	TCP	9090 inbound to HCP
SNMP	TCP/UDP	161 inbound to HCP
Syslog	UDP	9601 (the default listener port of Monitor-App) inbound to the HCM node

Time source

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

Supported browsers

The HCI web applications support these web browsers:

- The latest version of Google Chrome
- The latest version of Mozilla Firefox
- The latest version of Microsoft Edge

File ownership considerations

Within some of the Docker containers on each system instance, file ownership is assigned to this user and group:

- User: hci, UID: 10001
- Group: hci, GID: 10001

When you view such files in the instance operating system (for example, by running `ls -l`), the files appear to be owned by an unknown or undefined user and group. Typically, this causes no issues.

However, if you run applications on the system instances that change file ownership (for example, security hardening scripts), changing the ownership of files owned by the hci user and group can cause the system to become unresponsive.

To avoid these issues:

1. Create the expected user and group on each instance:

   ```
   sudo groupadd hci -g 10001
   sudo useradd hci -u 10001 -g 10001
   ```

2. Configure your applications to not change the ownership of files owned by the hci user and group.

Chapter 3: Installing HCI

This chapter describes how to install a system by deploying a number of software instances.

After you've set up all the instances that you want, you log into the Admin App to deploy the system.

Items you need

To install a system, you need the `HCI-<version-number>.tgz` file.

This archive file includes the software installation files needed to install your HCI instance.

Considerations for Solr backup and restore

To utilize the Solr backup and restore functionality of HCI, the following prerequisites need to be met on your system:

1. An external, dedicated NFS mount point for each HCI cluster.
2. A directory created on each node in your HCI cluster called `solrBackups`, located at the following path: `install_path/solrBackups`
3. The file system from step 1 mounted on each node of the directory listed in step 2.

   Note: The mechanism used to mount the file system needs to be able to persist through node reboot.
4. Sufficient disk space available on the mounted file system in order to successfully **backup** your HCI index.
5. Sufficient disk space reserved on your HCI nodes in order to successfully **restore** your HCI index.

It is recommended that all of the above mentioned mount points be set up *prior* to your installation of HCI.

For more information about Solr backup and restore after installing HCI, refer to the Workflow Designer Help.

HCI installation process

HCI installation consists of the following steps:

1. Decide how many instances to deploy (on page 36)

2. Configure your networking environment (on page 36)
3. (Optional) Select master instances (on page 37)
4. Install Docker on each server or virtual machine (on page 37)
5. Configure Docker on each server or virtual machine (on page 38)
6. (Optional) Install Docker volume drivers (on page 38)
7. (Optional) Enable or disable SELinux on each server or virtual machine (on page 39)
8. Configure maximum map count setting (on page 39)
9. Configure the firewall rules on each server or virtual machine (on page 40)
10. Run Docker on each server or virtual machine (on page 39)
11. Unpack the installation package (on page 40)
12. (Optional) Reconfigure network.config on each server or virtual machine (on page 41)
13. (Optional) Reconfigure volume.config on each server or virtual machine (on page 42)
14. Run the setup script on each server or virtual machine (on page 45)
15. Start the application on each server or virtual machine (on page 48)
16. (Optional) Configure NTP (on page 49)
17. Access deployment wizard (on page 50)
    a. (optional) Configure service networking (on page 51)
    b. (optional) Configure volumes for services and jobs (on page 51)
18. Deploy the system (on page 55)
19. Verify the created volumes (on page 55)
20. Distribute services among system instances (on page 56)
21. Configure the system for your users (on page 59)

Decide how many instances to deploy

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

The minimum for a production system is four instances.

Procedure

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

Configure your networking environment

Before installing the system, you need to determine the networks and ports each HCI service will use.

Procedure

1. Determine what ports each HCI 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 HCI services, can share the same port.

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

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

(Optional) Select master instances

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

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

Important:

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

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

Procedure

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

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

Install Docker on each server or virtual machine

On each server or virtual machine that is to be an HCI instance:

Procedure

1. In a terminal window, verify whether Docker 1.13.1 or later is installed:
   ```
   docker --version
   ```

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

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

Configure Docker on each server or virtual machine

Before installing the product, configure Docker with settings suitable for your environment. For guidance on configuring and running Docker, see the applicable Docker documentation.

Procedure

1. Ensure that the Docker installation folder on each instance has at least 20 GB available for storing the product Docker images.
2. Ensure that the Docker storage driver is configured correctly on each instance. After installation, changing the Docker storage driver needs reinstallation of the product.

   To view the current Docker storage driver on an instance, run: `docker info .`
3. To enable SELinux on the system instances, use a Docker storage driver that 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.

   The product needs 20 GB to install successfully and an additional 20 GB to successfully update to a later version.

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

Next steps

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

(Optional) Install Docker volume drivers

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

Procedure

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

   If you don't, services might fail to run on the new instance.

2. If any services on your system use Docker volume drivers for storing data (instead of using the default bind-mount setting), install those volume drivers on all instances in the system.

(Optional) Enable or disable SELinux on each server or virtual machine

You should decide whether you want to run SELinux on system instances before installation.

Procedure

1. Enable or disable SELinux on each instance.
2. Restart the instance.

Configure maximum map count setting

You need to configure a value in the file `sysctl.conf`.

Procedure

1. On each server or virtual machine that is to be a system instance, open the file `/etc/sysctl.conf`.
2. Append this line: `vm.max_map_count = 262144`

   If the line already exists, ensure that the value is greater than or equal to `262144`.
3. Save and close the file.

Run Docker on each server or virtual machine

On each server or virtual machine that is to be a system instance, you need to start Docker and keep it running. You can use whatever tools you typically use for keeping services running in your environment.

For example, to run Docker using systemd:

Procedure

1. Verify that Docker is running:
   ```
   systemctl status docker
   ```
2. If Docker is not running, start the `docker` service:
   ```
   sudo systemctl start docker
   ```
3. (Optional) Configure the Docker service to start automatically when you restart the server or virtual machine:
   ```
   sudo systemctl enable docker
   ```

Unpack the installation package

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

Procedure

1. Download the product installation package and MD5 checksum file and store both in a folder on the server or virtual machine.
2. Verify the integrity of the installation package:
   `md5sum -c HCI-version_number.tgz.md5`
   If the package integrity is verified, the command displays `OK`.
3. In the largest disk partition on the server or virtual machine, create a product installation folder.
   `mkdir install_path/hci`
4. Move the installation package from the folder where you stored it to the product installation folder.
   `mv HCI-version_number.tgz install_path/hci/HCI-version_number.tgz`
5. Navigate to the installation folder.
   `cd install_path/hci`
6. Unpack the installation package:
   `tar -zxf HCI-version_number.tgz`
   A number of directories are created within the installation folder.
7. Run the `install` script:
   `sudo ./install`

Notes:

- Don't change directories after running the installation script. The following tasks are performed in your current folder.
- The installation script can be run only one time on each instance. You cannot rerun this script to try to repair or upgrade a system instance.

Configure the firewall rules on each server or virtual machine

Before you begin

Determine the port values currently used by your system. To do this, on any instance, view the file `install_path/config/network.config`.

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

Procedure

1. Edit the firewall rules to allow communication over all network ports that you want your system to use. You do this using a firewall management tool such as `firewalld`.
2. Restart the server or virtual machine.

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

Before you begin

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

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

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 HCI instance, use a text editor to open the file *install_path*`/hci/config/network.config`.

   The file contains two types of lines for each service:

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

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

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

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

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

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

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

   Note: If you reconfigure service ports, make sure that each port value you assign is unique across all services, both System services and HCI 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 HCI; 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 HCI instances.

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

Before you begin

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

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

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

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

You configure volumes for HCI 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 HCI instance, use a text editor to open the file *`install_path`*`/hci/config/volume.config`.

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

- The name of the volume:

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

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

- The volume driver that the volume uses:

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

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

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

```
com.hds.ensemble.plugins.service.adminApp.log_volume_name=SERVICE_PLUGIN_NAME.INSTANCE_UUID.log
com.hds.ensemble.plugins.service.adminApp.log_volume_driver=bind-mount
com.hds.ensemble.plugins.service.adminApp.log_volume_driver_opt_1=hostpath=/home/hci/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 HCI system itself. For information on volume drivers, their capabilities, and their valid configuration settings, see the applicable Docker or third-party developer's documentation.
   - On the line that contains `_opt`, the options for the volume driver.

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

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

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

Example

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

```
com.hds.ensemble.plugins.service.marathon.data_volume_name=SERVICE_PLUGIN_NAME.INSTANCE_UUID.data
com.hds.ensemble.plugins.service.marathon.data_volume_driver=local-persist
com.hds.ensemble.plugins.service.marathon.data_volume_driver_opt_1=mountpoint=/home/hci/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/hci/bin/setup -i 192.0.2.4
  -m 192.0.2.0,192.0.2.1,192.0.2.3
  ```

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

Procedure

1. Run the script `setup` with the applicable options:

Option	Description
`-i`	The external network IP address for the instance on which you're running the script.
`-I`	The internal network IP address for the instance on which you're running the script.
`-m`	Comma-separated list of external network IP addresses of each master instance.
`-M`	Comma-separated list of internal network IP addresses of each master instance.
`-i IPADDRESS`	Displays the external instance IP address. If not specified, this value is discovered automatically.
`-I IPADDRESS`	Displays the internal instance IP address. If not specified, this value is the same as the external IP address.
`-d`	Attempts to automatically discover the real master list from the provided masters.
`--hci_uid UID`	Allows you to set the desired user ID (UID) for the HCI USER at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.

Option	Description
`--hci_gid GID`	Allows you to set the desired group ID (GID) for the HCI GROUP at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--mesos_uid UID`	Allows you to set the desired user UID for the MESOS USER at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--mesos_gid GID`	Allows you to set the desired GID for the MESOS GROUP at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--haproxy_uid UID`	Allows you to set the desired UID for the HAPROXY USER at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--haproxy_gid GID`	Allows you to set the desired GID for the HAPROXY GROUP at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--zk_uid UID`	Allows you to set the desired UID for the ZOOKEEPER USER at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.
`--zk_gid GID`	Allows you to set the desired GID for the ZOOKEEPER GROUP at *install time only*. **Important:** This value needs to be greater than 1000, less than or equal to 65533, and the same on all nodes in a cluster.

Use the following table to determine which options to use:

Number of instances in the system	Network type usage	Options to use
Multiple	Single network type for all services	Either: `-i` and `-m` or `-I` and `-M`
Multiple	Internal for some services, external for others	All of these: `-i`, `-I`, `-m`, `-M`
Single	Single network type for all services	Either `-i` or `-I`
Single	Internal for some services, external for others	Both `-i` and `-I`

Result

Note: If the terminal displays Docker errors when you run the `setup` script, ensure that Docker is running.

Example

The following example sets up a single-instance system that uses only one network type for all services:

```
sudo install_path/hci/bin/setup -i 192.0.2.4
```

To set up a multi-instance system that uses both internal and external networks, type the command in this format:

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

The following table shows sample commands to create a four-instance system. Each command is entered on a different server or virtual machine that is to be a system instance. The resulting system contains three master instances and one worker instance and uses both internal and external networks.

Instance internal IP	Instance external IP	Master or worker	Command
192.0.2.1	10.236.1.1	Master	`sudo install_path/hci/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/hci/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/hci/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/hci/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

Procedure

1. Start the application script `run` using whatever methods you usually use to run scripts.

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

- You can run the script in the foreground: `sudo install_path/hci/bin/run`

  When executed this way, the `run` script does not automatically complete, but instead remains running in the foreground.

- You can run the script as a service using `systemd`:

  a. Open the **HCI.service** file in a text editor, located in `install_path/bin`.

  b. Verify that the following two lines have the correct `install_path`:

  ```
  ExecStart=install_path/hci/bin/run
  ExecStopPost=install_path/hci/bin/stop
  ```

  c. Save the file.

  d. Copy the **HCI.service** file to the appropriate location for your OS:

  ```
  cp install_path/hci/bin/HCI.service /etc/systemd/system
  ```

  e. Enable and start **HCI.service**:

  ```
  sudo systemctl daemon-reload
  sudo systemctl enable HCI.service
  sudo systemctl start HCI.service
  ```

Note: When you enable HCI.service, systemctl might display this message:

```
The unit files have no [Install] section. They are not
meant to be enabled using systemctl. Possible reasons
for having this kind of units are:

1) A unit may be statically enabled by being symlinked
from another unit's .wants/ or .requires/ directory.

2) A unit's purpose may be to act as a helper for some
other unit which has a requirement dependency on it.

3) A unit may be started when needed via activation
(socket, path, timer, D-Bus, udev, scripted systemctl
call, ...).
```

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

To avoid this, make sure that you move HCI.service to the appropriate location, typically `/etc/systemd/system`.

(Optional) Configure NTP

If you are installing a multi-instance system:

Procedure

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

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

Access deployment wizard

After creating all of your instances, you need to go to the service deployment wizard in the Admin App.

Important: You cannot change networking or volume settings for System services at this point.

Alternatively, if you configured the System services networking incorrectly, the Admin App might fail to appear. This can happen, for example, if the `network.config` file is not identical on all instances. For error information, view the file `install_path/hci/config/cluster.config` or the output information logged by the `run` script.

To fix this issue, do the following:

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

   ```
   sudo install_path/hci/bin/stop
   ```

3. Delete the contents of the folder `install_path/hci` 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 40).

To access the service deployment wizard:

Procedure

1. Open a web browser and go to: `https://instance_ip_address:8000`
2. On the **Welcome** page, set a password for the admin user account. Then click **Set Admin Password**.

   Important: Do not lose or forget this password.

3. On the **Licensing** page:
   - If you have your purchased license file, drag and drop it into the **Upload License** section.
   - If you've purchased a license but have not yet received it, make note of the value in the System ID section on the **Licensing** page and contact your sales representative.

- To use the system for a limited amount of time with the pre-installed trial license, click **Continue**.
- If for some reason the trial license failed to install, there is a copy included in the HCI-<version-number>.tgz installation package that you can upload to the **Licensing** page. The trial license is located in the installation package at:

  `install_path/product/<version>/trial-<version>.plk`

4. On the **Set Cluster Hostname/IP** page, specify the hostname for your system. Omitting this can cause links in the Admin App to function incorrectly.
5. On the **Choose Deployment** page, select the HCI deployment type that you purchased, either Hitachi Content Search or HCM. Then click **Continue**.
6. The **Confirm Cluster Topology** page shows all detected instances. If your system includes multiple instances, make sure that all instances that you expect to see are listed.

(optional) Configure service networking

To change networking settings for the HCI services:

Procedure

1. On the **Services** tab, select a service to configure.
2. On the **Networks** tab:
   a. Optionally, configure the ports that the service should use.

      If you reconfigure service ports, make sure that each port value you assign is unique across all services, both System services and HCI services.

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

      By default, the Search-App, Monitor-App, and Admin-App services have the **External** option selected and all other 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 HCI; 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 and jobs

To change volumes usage:

Procedure

1. Click the **Services** or **Jobs** tab and select a service or job type 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 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 Content Intelligence system itself. For information on volume drivers, their capabilities, and their valid configuration settings, see the applicable Docker or third-party developer's documentation.
- The Workflow-Agent job type supports only the default bindmount setting. You cannot specify a volume driver for this job type.

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

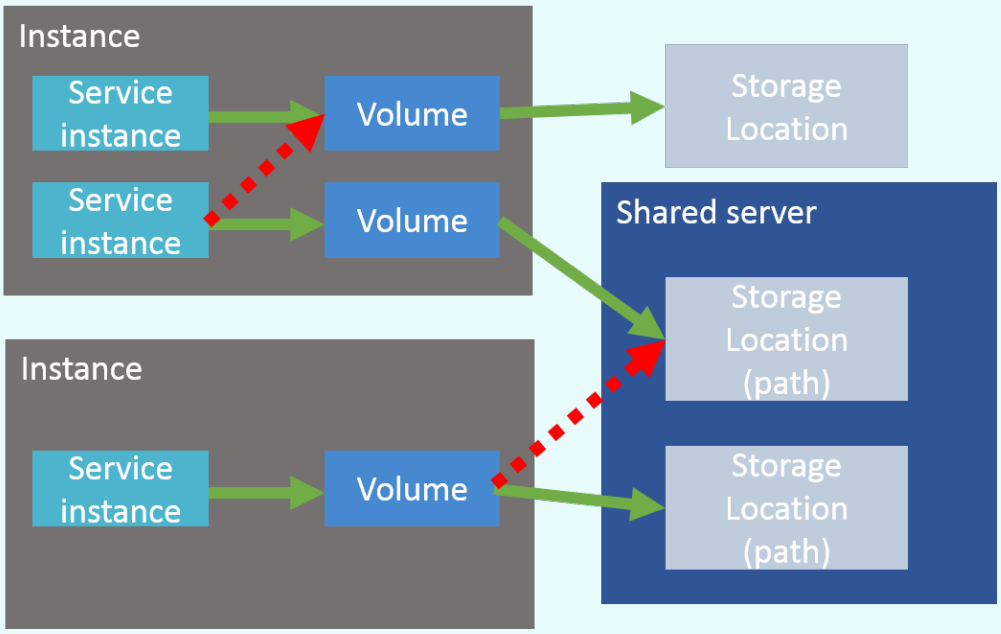
- If you're using the **bind-mount** setting, you can edit the value for the **hostpath** option to change the path where the volume's data is stored on each system instance. However, this must be a path within the Content Intelligence installation folder.
- If you're using a volume driver:
  - Click the delete icon to remove the default **hostpath** option. This option applies only when you are using the **bind-mount** setting.
  - Type the name of a volume driver option in the **Option** field. Then type the corresponding parameter for that option in the **Value** field.
  - Click the plus-sign icon to add the option/value pair.
  - Repeat this procedure for each option/value pair you want to add.
- For considerations regarding adding option/value pairs, see Considerations for option/value pairs (on page 53).

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

Considerations for option/value pairs

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

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



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

- For persistent (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 Example: Docker local volume driver for Database service log volume.

- 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.

  For services with multiple types, consider specifying the type name as a part of the path to where service instances of that type write their data:

  ```
  /example/typeA/${node_ip}
  ```

  ```
  /example/typeB/${node_ip}
  ```

  For information about the ${node_ip} variable, see Available variables.

- Make sure the options and values you specify are valid. Incorrect options or values can cause system deployment to fail or volumes to be set up incorrectly. For information on volumes, see the volume driver's documentation.

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 or job runs. For floating services, this is the same value for all instances of the service.
- `${node_ip}` is the IP address for the system instance on which the service or job is running. This cannot be used for floating services.
- `${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, after the system is deployed, logs for the Database service are stored at a unique path on each system instance that runs the Database service:

`/<install-dir>/log/com.hds.ensemble.plugins.service.cassandra/`*`service-instance-uuid`*

For example:

`/home/hci/log/com.hds.ensemble.plugins.service.cassandra/12345678-1234-1234-1234-123456789012`

Example: Docker local volume driver for Database service log volume

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

- Log in to your NFS server.
- Create a directory.

- 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.

- Back in the system deployment wizard, in the Volume Driver field, specify local
- Specify these options and values:

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

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

Deploy the system

After all your instances have been discovered and (optionally) configured:

Procedure

1. Click **Deploy Single Instance** or **Deploy Cluster** (multi-instance), as appropriate. The system deployment starts.
2. Click the link **View Deployment Details** to view the progress of the deployment.

Verify the created volumes

If you configured the service volumes to use volume drivers in (optional) Configure volumes for services and jobs (on page 51), 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. Run this command to stop all Content Intelligence Docker containers on the instance: `sudo install_path/hci/bin/stop`

3. Delete the contents of the folder `install_path/hci` 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 40).

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 required number of instances. For example, the Index service runs on three instances.

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

Moving and scaling floating services

For floating services, instead of specifying the specific instances on which the service runs, you can specify a pool of eligible instances, any of which can run the service.

Moving and scaling services with multiple types

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

Best Practices

- Moving or scaling services can cause document failures during a workflow task. Before moving or scaling a service, you should either pause all running workflow tasks or wait for them to complete.
- Avoid running multiple services with high service unit costs together on the same instance.

  Ideally, each of these services should run by itself on an instance:

  - Database
  - Index
- On master instances, avoid running any services besides those classified as System services.
- To use your instances evenly, try to deploy a comparable number of service units on each instance.

Considerations

- You cannot remove a service from an instance if doing so causes or risks causing data loss.
- Service relocation might take a long time to complete and can impact system performance 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.

Relocating services

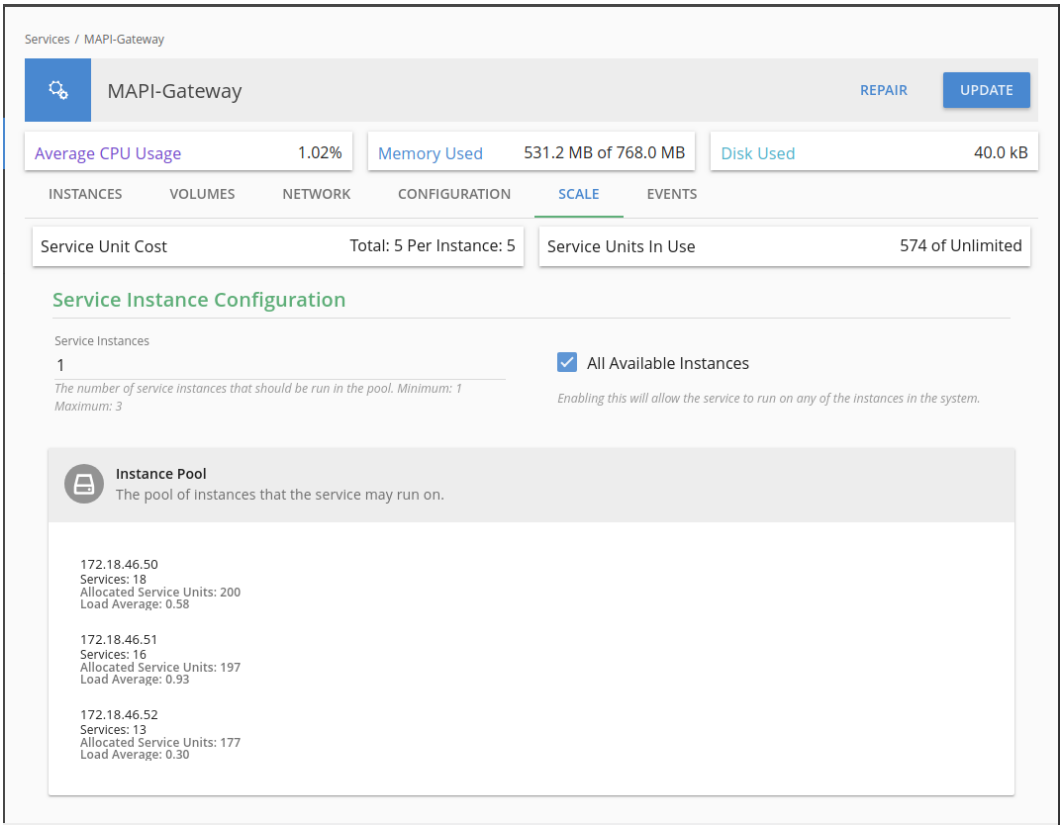
To manually relocate a service, in the Admin App:

Procedure

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

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

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



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

b. Configure the instance pool:

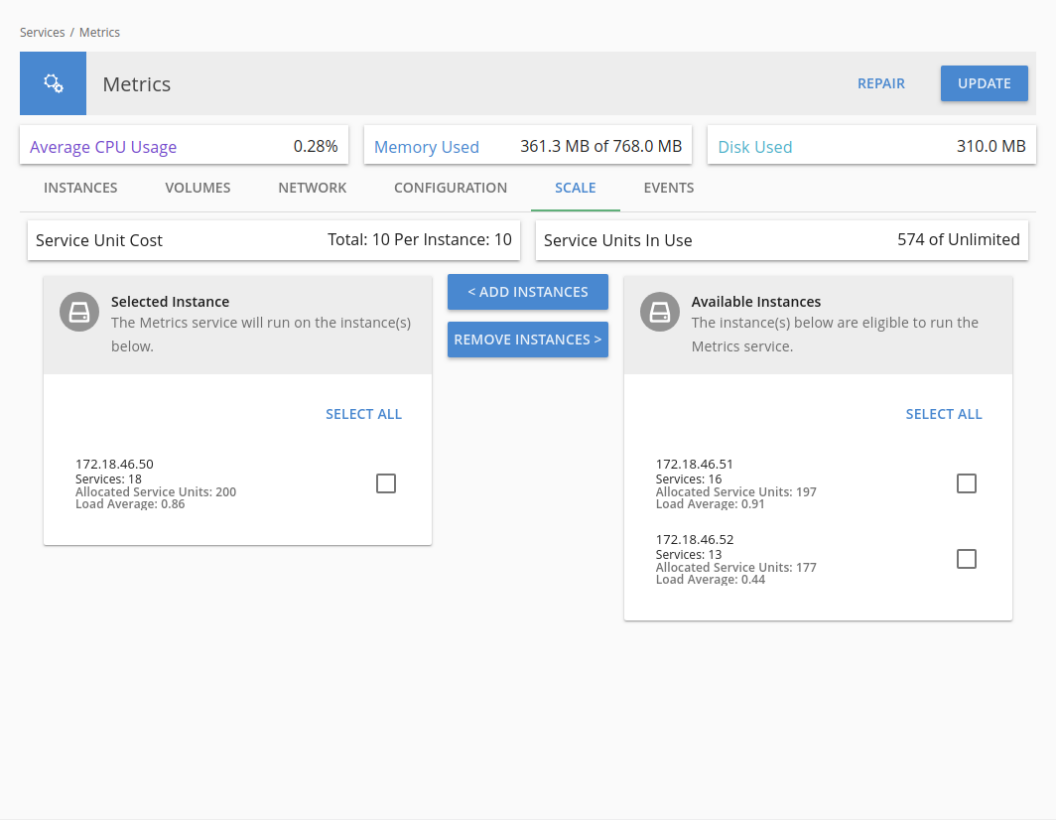
- For the service to run on any instance in the system, select **All Available Instances**.

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

- For the service to run on a specific set of instances, clear **All Available Instances**. Then:

  - To remove an instance from the pool, select it from the list **Instance Pool**, on the left, and then click **Remove Instances**.
  - To add an instance to the pool, select it from the list **Available Instances**, on the right, and then click **Add Instances**.

5. If the service is a persistent (non-floating) service, you are presented with options for selecting the specific instances that the service should run on. Do one or both of these, then click **Next**:



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

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

Next steps

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

Configure the system for your users

Once your system is up and running, you need to begin configuring it for your users. For information, see the applicable topic in the help that's available from the Admin App:

- *Administering Hitachi Content Search*
- *Administering Hitachi Content Monitor*

Appendix A: Logs and diagnostic information

Each service maintains its own set of logs. By default, the logs are maintained in the `install_path/hci/log` folder 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 HCI log files yourself, deleting or archiving them as necessary.

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

System logs are managed automatically in these ways:

- All log files are periodically added to a compressed file and moved to `install_path/hci/retired/`. This occurs at least one time 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.
- After a log file grows larger than 10MB in size, the system stops writing to that file, renames it, and begins writing to a new file. For example, if `exampleService.log.0` grows too large, it is renamed to `exampleService.log.1` and the system creates a new `exampleService.log.0` to write to.

Retrieving logs and diagnostic information

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

```
install_path/hci/bin/log_download
```

For information about running the tool, use this command:

```
install_path/hci/bin/log_download -h
```

Note:

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

Default log locations

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

`install_path/hci/log`

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

Related service	Default log directory name	Contains information about
Admin-App	com.hds.ensemble.plugins.service.adminApp	The System Management application.
Cluster-Coordination	com.hds.ensemble.plugins.service.mesosMaster	Hardware resource allocation.
Cluster-Worker	com.hds.ensemble.plugins.service.mesosAgent	Work ordered by the Cluster-Coordination service.
Dashboard	com.hds.ensemble.plugins.service.kibana	The advanced Dashboard Management service.
Database	com.hds.ensemble.plugins.service.cassandra	▪ System configuration data. ▪ Document fields and values.
Index	com.hds.ensemble.plugins.service.solr	Index collections and search indexes.

Related service	Default log directory name	Contains information about
Logging	com.hds.ensemble.plugins.service.logstash	The transport of system events and workflow task metrics to the Metrics service.
Metrics	com.hds.ensemble.plugins.service.elasticsearch	The storage and indexing of: ▪ System events ▪ Performance and failure metrics for workflow tasks
Message Queue	com.hds.ensemble.plugins.service.kafka	Transmission of data between instances.
Monitor-App	com.hds.ensemble.plugins.service.mapApp	The HCM application.
Network-Proxy	com.hds.ensemble.plugins.service.haproxy	Network requests between instances.
Scheduling	com.hds.ensemble.plugins.service.chronos	Workflow task scheduling.
Search-App	com.hds.ensemble.plugins.service.searchApp	The Search App
Sentinel	com.hds.ensemble.plugins.service.sentinel	Internal system processes.
Service-Deployment	com.hds.ensemble.plugins.service.marathon	The deployment of high-level services across system instances. High-level services are the ones that you can move and configure (such as Index), not the services grouped under System Services.
Synchronization	com.hds.ensemble.plugins.service.zookeeper	Coordination of actions and database operations across instances.

Related service	Default log directory name	Contains information about
Watchdog	com.hds.ensemble.plugins.service.remoteAction	Internal system processes
Watchdog	com.hds.ensemble.plugins.service.watchdog	General diagnostic information.
Workflow-Agent jobs	com.hds.ensemble.plugins.job.workflow	Workflows

Appendix B: Service list

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

- 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.
- Service unit cost per instance: The number of service units that it costs to run the service on one instance. This cost indicates how computationally expensive one service is compared to another.
- 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 has a single type or multiple.

Note: For services with both the Container Memory and Max Heap Size settings, the Container Memory setting should be larger than the Max Heap Size setting.

Service name and description	Service properties	
The services perform functions related to the system's supported use cases. You can move, scale, and reconfigure these services.		
Admin-App Runs the Admin App.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	10
	Persistent or floating	Persistent
	Supports volume configuration	Yes

Service name and description	Service properties	
	Single or multiple types	Single
Cluster-Coordination Mesos (master) - https://mesos.apache.org Hardware resource management solution for distributed systems. *How it's used* Manages hardware resource allocation.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	1
	Persistent or floating	Persistent
	Supports volume configuration	No
	Single or multiple types	Single
Cluster-Worker Mesos (slave) - https://mesos.apache.org Hardware resource management solution for distributed systems. *How it's used* Receives and performs work from other services. **Note**: Though the Cluster-Worker service has a low service unit cost, it can at times appear to be using a large amount of CPU resources. When other services use Cluster-Worker to perform their work, Cluster-Worker reflects the CPU usage of those services.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	5
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Dashboard https://www.elastic.co/products/kibana Visualizes information stored in Elasticsearch indexes. *How it's used*	RAM needed per instance	300 MB
	Number of instances	Required: 0 Optimal: 2
	Service unit cost per instance	5
	Persistent or floating	Persistent

Service name and description	Service properties	
Powers the advanced Dashboard Management service. **Note**: This service is in the Unconfigured state by default.	Supports volume configuration	Yes
	Single or multiple types	Single
Database http://cassandra.apache.org/ Decentralized database that can be scaled across large numbers of hardware nodes. *How it's used* Stores system configuration data. Also stores document discovery and failure data for workflow tasks.	RAM needed per instance	2.4 GB
	Number of instances	Required: 1 Optimal: 3
	Service unit cost per instance	10
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Index http://lucene.apache.org/solr/ Data indexing and search platform. *How it's used* The search engine that manages all internal search indexes.	RAM needed per instance	2 GB
	Number of instances	Required: 0 Optimal: 3 **Notes:** ▪ No instances are required to run this service, but without at least one, you cannot index data. ▪ If multiple copies of an index exist (with an Index Protection Level greater than one), each copy is managed by a separate instance of the Index service.
	Service unit cost per instance	25
	Persistent or floating	Persistent
	Supports volume configuration	Yes

Service name and description	Service properties	
	Single or multiple types	Single
Logging https://www.elastic.co/products/logstash Collection engine for event data. Can perform transformations on the data it collects and then send that data to a number of outputs. *How it's used* Transports system logs and metrics data to the Metrics service.	RAM needed per instance	700 MB
	Number of instances	Required: 1 Optimal: 1
	Service unit cost per instance	10
	Persistent or floating	Floating
	Supports volume configuration	Yes
	Single or multiple types	Single
Message Queue https://kafka.apache.org/ Stream processing platform for handling real-time data streams. *How it's used* Facilitates communication between instances.	RAM needed per instance	2 GB
	Number of instances	Required: 1 Optimal: 3
	Service unit cost per instance	5
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Metrics https://www.elastic.co/ Data indexing and search platform. *How it's used* Stores and manages: ▪ System events ▪ Workflow performance data ▪ Workflow failure data	RAM needed per instance	2000 MB
	Number of instances	Required: 1 Optimal: 3
	Service unit cost per instance	25
	Persistent or floating	Persistent
	Supports volume configuration	Yes

Service name and description	Service properties	
The service maintains this information in a number of internally-managed Metrics indexes.	Single or multiple types	Single
Monitor-App Powers the Monitor App.	RAM needed per instance	556 MB
	Number of instances	Required: 0 Optimal: 1 **Note**: Scaling the Monitor-App service does not affect any of the workflows that collect data from the systems you are monitoring. For example, if you scale the service to run on 0 instances, users cannot access the Monitor App, but HCI will continue to collect data.
	Service unit cost per instance	10
	Persistent or floating	Floating
	Supports volume configuration	Yes
	Single or multiple types	Single
Network-Proxy HAProxy - https://haproxy.org Load balancer for TCP and HTTP-based applications. *How it's used* Maps network requests to the instances where the applicable services are located.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	1
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Scheduling https://mesos.github.io/chronos/	RAM needed per instance	712 MB
	Number of instances	Required: 1 Optimal: 1

Service name and description	Service properties	
Job scheduler for Apache Mesos. *How it's used* Schedules workflow tasks.	Service unit cost per instance	1
	Persistent or floating	Floating
	Supports volume configuration	Yes
	Single or multiple types	Single
Search-App Powers the Search App	RAM needed per instance	556 MB
	Number of instances	Required: 0 Optimal: 2 **Note**: No instances are required to run this service, but without at least one, the Search App is unavailable.
	Service unit cost per instance	10
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Sentinel Runs internal system processes and monitors the health of the other services.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	5
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Service-Deployment Marathon - https://mesosphere.github.io/marathon/ Orchestration platform for Mesos applications.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	1
	Persistent or floating	Persistent

Service name and description	Service properties	
How it's used Handles deployment of high-level services (that is, the services that you can configure).	Supports volume configuration	Yes
	Single or multiple types	Single
Synchronization Apache Zookeeper - https://zookeeper.apache.org/ Coordinates configuration settings and other information between a number of distributed services. *How it's used* Coordinates actions and database operations across instances.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	5
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single
Watchdog Monitors the other System Services and restarts them if necessary. Also responsible for initial system startup.	RAM needed per instance	N/A
	Number of instances	N/A
	Service unit cost per instance	5
	Persistent or floating	Persistent
	Supports volume configuration	Yes
	Single or multiple types	Single

Appendix C: Service units

Your system license limits grants you a number of service units. These limit how and where you can run services and jobs.

- For services, each service costs a certain number of service units per instance to run. For example, a service with a cost of one service unit that's running on three instances counts for three service units against your licensed limit.
- For jobs, service unit cost is assessed based on where job types are allowed to run, not on the number of individual jobs that you run.

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

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

```
50 + 50 + 50 + 25 = 175
```

Best practices for service unit limits

The system makes recommendations on the maximum number of service units that you should run on each instance. An instance that runs more than the recommended number of service units in use is likely to experience decreased performance.

The recommended service unit limits are based on whether an instance meets the recommended hardware requirements:

- If an instance meets the recommended hardware requirements, you can run up to 180 service units on that instance.
- If an instance does not meet the recommended hardware requirements, you can run up to 100 service units on that instance.

Appendix D: Handling network changes

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

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

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

After a network change

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

Procedure

1. If the Admin App is accessible, back up your system components by exporting a package. For information on exporting packages, see the Administrator Help, which is accessible from the Admin App
2. On each instance in the system:
   a. Navigate to the installation folder.
   b. Stop the script `run` using whatever tool or process you used to run it.
   For example, with systemd, run: `systemctl stop HCI.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.

Appendix E: About hardware and performance testing

This topic summarizes the system configurations and settings used by Hitachi Vantara to characterize HCI hardware and provide system sizing guidance for Hitachi Content Search.

Constant settings for all testing

- Documents indexed: Small text documents, average 5 KB
- Number of indexes per system: 1
- Index Protection Level per index: 1
- Workflow task settings: All defaults
- Data connection: HCP MQE data connection
- Initial schema setting for each index: Schemaless. Includes:
  - 151 defined fields
  - 73 dynamic fields
  - 3 copy fields

Variable settings

- Shards per index: Equal to the number of instances running the Index service.
- Instance and service configurations:

Number of instances	Instances running the Index service	Instances running the Workflow-Agent service
1	1	1
4	3	4
8	5	8

For information on:

- System sizing guidance, see Sizing guidance for Hitachi Content Search (on page 17).
- Index Protection Level settings, shards, and index schema options, see the HCI Administrator Help, which is available from the HCI Admin App.

Appendix F: Example HCI firewall setup

Important:

- This example details the steps required for a single node. This process must be repeated across all nodes in your system.
- Users upgrading their systems from HCI 1.6.1 to later versions of HCI who currently have existing signal sources and scripts executed will not receive syslog messages until these firewall scripts are rerun on their upgraded system.
- Prior to running the scripts, ensure that the firewall service is enabled.
- While running the scripts, users may enounter errors due to nmcli not working as a result of NetworkManager being disabled. To enable it, type: `systemctl start NetworkManager`
- After the scripts have concluded, you will need to restart HCI.

The following is an example of what a hardened HCI cluster running CentOS Linux 7.4.1708 (Core) would look like if it was set up to ONLY allow HCI to run from within it.

The following firewall scripts are now located in `<hci_install_directory>/bin`:

- hciConfigFirewallExample.sh
- hciFirewallExampleUtils
- hciProcessFirewall

To run the example script on your system, execute **hciConfigFirewallExample.sh**.

WARNING:

The following firewalld example was created using our proprietary script. It is compatible with HCI versions 1.5 and later.

This script IS NOT officially supported or licensed by Hitachi Vantara. Usage of this script assumes all risks and responsibilities associated with it. Also, based on your personal network and system settings, your mileage with its usage and implementation may vary. Contact your system administrator if you have any network security or firewall concerns.

Table 1 Set up two network interfaces to be used as a trusted network interface (for internal HCI traffic) and a non-trusted network interface (external HCI traffic).

Network interfaces examples	
`ens160 : 172.18.118.111`	In the following config example, this network interface is the external non-trusted interface.
`ens192 : 172.118.110.111`	In the following config example, this network interface is the internal trusted interface.

Table 2 Set up three active zones and a default zone.

Zone setup	
Default Zone	`drop`

Active Zones	`HCI-External`
	`trusted`
	`HCI-AdminApp-Mon`

Table 3 Firewalld configuration example: *drop*

To view your current settings: `firewall-cmd --list-all --zone=drop`	
target	`DROP`
icmp-block-inversion	`no`
interfaces	<blank>
sources	<blank>
services	<blank>
ports	<blank>
protocols	<blank>
masquerade	`no`
forward-ports	<blank>
source-ports	<blank>
icmp-blocks	<blank>
rich rules	<blank>

Table 4 Firewalld config example: *HCI-External*

To view your current settings: `firewall-cmd --list-all --zone=HCI-External`	
target	`DROP`
icmp-block-inversion	`no`
interfaces	`ens160`
sources	<blank>
services	`ssh`
ports	`8000/tcp 8888/tcp 6162/tcp`
protocols	<blank>
masquerade	`no`
forward-ports	<blank>
source-ports	<blank>
icmp-blocks	<blank>
rich rules	<blank>

Table 5 Firewalld config example: *trusted*

To view your current settings: `firewall-cmd --list-all --zone=trusted`	
target	`ACCEPT`
icmp-block-inversion	`no`
interfaces	`ens192`
sources	<blank>
services	<blank>
ports	<blank>
protocols	<blank>
masquerade	`no`
forward-ports	<blank>
source-ports	<blank>
icmp-blocks	<blank>
rich rules	<blank>

Table 6 Firewalld config example: *HCI-AdminApp-Mon*

To view your current settings: `firewall-cmd --list-all --zone=HCI-AdminApp-Mon`	
target	`default`
icmp-block-inversion	`no`
interfaces	<blank>
sources	`ipset:HCI-Cluster-External`
services	<blank>
ports	<blank>
protocols	`tcp`
masquerade	`no`
forward-ports	<blank>
source-ports	`18000/tcp`
icmp-blocks	<blank>
rich rules	<blank>

Table 7 Linux system example: *ipset* table

To view your current settings: `ipset list`	
Name	`default`
Type	`no`
Revision	<blank>
Header	`ipset:HCI-Cluster-External`
Size in memory	<blank>
References	<blank>

Members	`<IP_ADDRESS_FOR_NODE_1>`
	`<IP_ADDRESS_FOR_NODE_2>`
	`<IP_ADDRESS_FOR_NODE_3>`
	`<IP_ADDRESS_FOR_NODE_4>`

To view your current settings: `ipset list`	
	Note: These values would be filled with the specific IP addresses for each of your system nodes.

Table 8 The following is an example of what the *iptables* look like after completing the above:

To view your current settings: `iptables -S`
▪ -P INPUT ACCEPT ▪ -P FORWARD ACCEPT ▪ -P OUTPUT ACCEPT ▪ -N FORWARD_IN_ZONES ▪ -N FORWARD_IN_ZONES_SOURCE ▪ -N FORWARD_OUT_ZONES ▪ -N FORWARD_OUT_ZONES_SOURCE ▪ -N FORWARD_direct ▪ -N FWDI_HCI-AdminApp-Mon ▪ -N FWDI_HCI-AdminApp-Mon_allow ▪ -N FWDI_HCI-AdminApp-Mon_deny ▪ -N FWDI_HCI-AdminApp-Mon_log ▪ -N FWDI_HCI-External ▪ -N FWDI_HCI-External_allow ▪ -N FWDI_HCI-External_deny ▪ -N FWDI_HCI-External_log ▪ -N FWDI_drop ▪ -N FWDI_drop_allow ▪ -N FWDI_drop_deny

To view your current settings: `iptables -S`
▪ -N FWDI_drop_log ▪ -N FWDI_trusted ▪ -N FWDI_trusted_allow ▪ -N FWDI_trusted_deny ▪ -N FWDI_trusted_log ▪ -N FWDO_HCI-AdminApp-Mon ▪ -N FWDO_HCI-AdminApp-Mon_allow ▪ -N FWDO_HCI-AdminApp-Mon_deny ▪ -N FWDO_HCI-AdminApp-Mon_log ▪ -N FWDO_HCI-External ▪ -N FWDO_HCI-External_allow ▪ -N FWDO_HCI-External_deny ▪ -N FWDO_HCI-External_log ▪ -N FWDO_drop ▪ -N FWDO_drop_allow ▪ -N FWDO_drop_deny ▪ -N FWDO_drop_log ▪ -N FWDO_trusted ▪ -N FWDO_trusted_allow ▪ -N FWDO_trusted_deny ▪ -N FWDO_trusted_log ▪ -N INPUT_ZONES ▪ -N INPUT_ZONES_SOURCE ▪ -N INPUT_direct ▪ -N IN_HCI-AdminApp-Mon

To view your current settings: `iptables -S`
▪ -N IN_HCI-AdminApp-Mon_allow ▪ -N IN_HCI-AdminApp-Mon_deny ▪ -N IN_HCI-AdminApp-Mon_log ▪ -N IN_HCI-External ▪ -N IN_HCI-External_allow ▪ -N IN_HCI-External_deny ▪ -N IN_HCI-External_log ▪ -N IN_drop ▪ -N IN_drop_allow ▪ -N IN_drop_deny ▪ -N IN_drop_log ▪ -N IN_trusted ▪ -N IN_trusted_allow ▪ -N IN_trusted_deny ▪ -N IN_trusted_log ▪ -N OUTPUT_direct ▪ -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT ▪ -A INPUT -i lo -j ACCEPT ▪ -A INPUT -j INPUT_direct ▪ -A INPUT -j INPUT_ZONES_SOURCE ▪ -A INPUT -j INPUT_ZONES ▪ -A INPUT -m conntrack --ctstate INVALID -j DROP ▪ -A INPUT -j REJECT --reject-with icmp-host-prohibited ▪ -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT ▪ -A FORWARD -i lo -j ACCEPT

To view your current settings: `iptables -S`

- -A FORWARD -j FORWARD_direct
- -A FORWARD -j FORWARD_IN_ZONES_SOURCE
- -A FORWARD -j FORWARD_IN_ZONES
- -A FORWARD -j FORWARD_OUT_ZONES_SOURCE
- -A FORWARD -j FORWARD_OUT_ZONES
- -A FORWARD -m conntrack --ctstate INVALID -j DROP
- -A FORWARD -j REJECT --reject-with icmp-host-prohibited
- -A OUTPUT -j OUTPUT_direct
- -A FORWARD_IN_ZONES -i ens192 -j FWDI_trusted
- -A FORWARD_IN_ZONES -i ens160 -j FWDI_HCI-External
- -A FORWARD_IN_ZONES -j FWDI_drop
- -A FORWARD_IN_ZONES_SOURCE -m set --match-set HCI-Cluster-External src -g FWDI_HCI-AdminApp-Mon
- -A FORWARD_OUT_ZONES -o ens192 -j FWDO_trusted
- -A FORWARD_OUT_ZONES -o ens160 -j FWDO_HCI-External
- -A FORWARD_OUT_ZONES -j FWDO_drop
- -A FORWARD_OUT_ZONES_SOURCE -m set --match-set HCI-Cluster-External dst -g FWDO_HCI-AdminApp-Mon
- -A FWDI_HCI-AdminApp-Mon -j FWDI_HCI-AdminApp-Mon_log
- -A FWDI_HCI-AdminApp-Mon -j FWDI_HCI-AdminApp-Mon_deny
- -A FWDI_HCI-AdminApp-Mon -j FWDI_HCI-AdminApp-Mon_allow
- -A FWDI_HCI-AdminApp-Mon -p icmp -j ACCEPT
- -A FWDI_HCI-External -j FWDI_HCI-External_log
- -A FWDI_HCI-External -j FWDI_HCI-External_deny
- -A FWDI_HCI-External -j FWDI_HCI-External_allow
- -A FWDI_HCI-External -j DROP

To view your current settings: `iptables -S`
▪ -A FWDI_drop -j FWDI_drop_log ▪ -A FWDI_drop -j FWDI_drop_deny ▪ -A FWDI_drop -j FWDI_drop_allow ▪ -A FWDI_drop -j DROP ▪ -A FWDI_trusted -j FWDI_trusted_log ▪ -A FWDI_trusted -j FWDI_trusted_deny ▪ -A FWDI_trusted -j FWDI_trusted_allow ▪ -A FWDI_trusted -j ACCEPT ▪ -A FWDO_HCI-AdminApp-Mon -j FWDO_HCI-AdminApp-Mon_log ▪ -A FWDO_HCI-AdminApp-Mon -j FWDO_HCI-AdminApp-Mon_deny ▪ -A FWDO_HCI-AdminApp-Mon -j FWDO_HCI-AdminApp-Mon_allow ▪ -A FWDO_HCI-External -j FWDO_HCI-External_log ▪ -A FWDO_HCI-External -j FWDO_HCI-External_deny ▪ -A FWDO_HCI-External -j FWDO_HCI-External_allow ▪ -A FWDO_HCI-External -j DROP ▪ -A FWDO_drop -j FWDO_drop_log ▪ -A FWDO_drop -j FWDO_drop_deny ▪ -A FWDO_drop -j FWDO_drop_allow ▪ -A FWDO_drop -j DROP ▪ -A FWDO_trusted -j FWDO_trusted_log ▪ -A FWDO_trusted -j FWDO_trusted_deny ▪ -A FWDO_trusted -j FWDO_trusted_allow ▪ -A FWDO_trusted -j ACCEPT ▪ -A INPUT_ZONES -i ens192 -j IN_trusted ▪ -A INPUT_ZONES -i ens160 -j IN_HCI-External

To view your current settings: `iptables -S`

- -A INPUT_ZONES -j IN_drop
- -A INPUT_ZONES_SOURCE -m set --match-set HCI-Cluster-External src -g IN_HCI-AdminApp-Mon
- -A IN_HCI-AdminApp-Mon -j IN_HCI-AdminApp-Mon_log
- -A IN_HCI-AdminApp-Mon -j IN_HCI-AdminApp-Mon_deny
- -A IN_HCI-AdminApp-Mon -j IN_HCI-AdminApp-Mon_allow
- -A IN_HCI-AdminApp-Mon -p icmp -j ACCEPT
- -A IN_HCI-AdminApp-Mon_allow -p tcp -m conntrack --ctstate NEW -j ACCEPT
- -A IN_HCI-AdminApp-Mon_allow -p tcp -m tcp --sport 18000 -m conntrack --ctstate NEW -j ACCEPT
- -A IN_HCI-External -j IN_HCI-External_log
- -A IN_HCI-External -j IN_HCI-External_deny
- -A IN_HCI-External -j IN_HCI-External_allow
- -A IN_HCI-External -j DROP
- -A IN_HCI-External_allow -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
- -A IN_HCI-External_allow -p tcp -m tcp --dport 8000 -m conntrack --ctstate NEW -j ACCEPT
- -A IN_HCI-External_allow -p tcp -m tcp --dport 8888 -m conntrack --ctstate NEW -j ACCEPT
- -A IN_HCI-External_allow -p tcp -m tcp --dport 6162 -m conntrack --ctstate NEW -j ACCEPT
- -A IN_drop -j IN_drop_log
- -A IN_drop -j IN_drop_deny
- -A IN_drop -j IN_drop_allow
- -A IN_drop -j DROP
- -A IN_trusted -j IN_trusted_log

To view your current settings: `iptables -S`
▪ -A IN_trusted -j IN_trusted_deny ▪ -A IN_trusted -j IN_trusted_allow ▪ -A IN_trusted -j ACCEPT

Appendix G: Removing an HCI system

In order to remove an HCI system, complete the following steps across all instances.

Procedure

1. Stop the `run` script using whatever method you used to start it.

   For example, if you used systemd to run HCI.service, you would use the following commands to stop and disable it:

   ```
   sudo systemctl stop HCI.service
   sudo systemctl disable HCI.service
   ```

   Additionally, if you used systemd to run HCI, you must delete the HCI.service file you copied during install. Typically, it is copied to `/etc/systemd/system`.

2. Run the following command to stop all HCI Docker containers on the instance:

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

3. Delete the HCI Docker containers.
   a. List all Docker containers:

      ```
      sudo docker ps
      ```

   b. Note the container IDs for all containers that use a *com.hds.ensemble* or *com.hitachi.foundry* image.
   c. Delete the containers:

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

4. Delete the HCI Docker images.
   a. List all Docker images:

      ```
      sudo docker images
      ```

   b. Note the image IDs for all images that use a *com.hds.ensemble* or *com.hitachi.foundry* repository.
   c. Delete the images:

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

5. Delete the HCI installation folder:

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

   For example:

   ```
   rm -rf /opt/hci
   ```

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