Cloudian HyperStore Installation Guide

Version 8.2.2

This page left intentionally blank.

This page left intentionally blank.

Contents

Chapter 1. HyperStore Installation Introduction

This documentation describes how to do a **fresh installation** of Cloudian® HyperStore® 8.2.2.

> **Note** For instructions on **upgrading** to 8.2.2 from an older HyperStore version see Chapter 3 "Upgrading Your HyperStore Software Version" in the ***Cloudian HyperStore Administrator's Guide***.

To do a fresh installation you need the **HyperStore product package**. To obtain the HyperStore 8.2.2 package:

- If you have an active Support contract you can download the package from the Cloudian Support Portal (https://cloudian-support.force.com).
- If you are doing an evaluation and do not have a Support contract you can obtain the package from your Cloudian sales representative.

To install and run HyperStore software you need a **HyperStore license file** — either an evaluation license or a production license. If you do not have a license file you can obtain one from your Cloudian sales representative or by registering for a free trial on the Cloudian website.

This page left intentionally blank

Chapter 2. Preparing Your Environment

Before installing HyperStore, Cloudian recommends that you prepare these aspects of your environment:

2.1. DNS Set-Up

Note As an alternative to setting up your DNS system to directly resolve HyperStore service endpoints (as described in this section), you have the option to use HyperStore's integrated nameserver **hsDNS** in combination with DNS delegation. For more information, in the "Service Features" section of your Cloudian HyperStore Administrator's Guide, see "hsDNS".

Subjects covered in this section:

For your HyperStore system to be accessible to external clients, you must configure your DNS name servers with entries for the HyperStore service endpoints. **Cloudian recommends that you complete your DNS configuration prior to installing the HyperStore system.** This section describes the required DNS entries.

2.1.1. HyperStore Service Endpoints

HyperStore includes a variety of services each of which is accessible to clients by way of a web service endpoint. On your name servers you will need to configure a DNS entry for each of these service endpoints.

By default the HyperStore system uses a standard format for each service endpoint, building on two values that are specific to your environment:

- Your organization's domain (for example *enterprise.com*)
- The name or names of your HyperStore service region or regions (for example *boston* for a single-region system, or *boston* and *chicago* for a multi-region system). Only lower case alphanumeric characters and dashes are allowed in region names, the minimum allowed region name length is 3 characters and the maximum length is 52 characters.

During HyperStore installation you will supply your domain and your service region name(s), and the interactive installer will show you the default service endpoints derived from the domain and region name(s). During installation you can accept the default endpoints or specify custom endpoints instead. The table that follows below is based on the default endpoint formats.

Note
* Do not include the string "s3" in your domain or in your region name(s). By default HyperStore generates S3 service endpoints by prepending an "*s3-*" prefix to your *<regionname>.<domain>* combination. If you include "s3" within either your domain or your region name, this will result in two instances of "s3" in the system-generated S3 service endpoints, and this may cause S3 service requests to fail for some S3 clients.
* If you specify custom endpoints during installation, do not use IP addresses in your endpoints.
* HyperStore by default derives the S3 service endpoint(s) as *s3-<regionname>.<domain>*. However HyperStore also supports the format *s3.<regionname>.<domain>* (with a dot after the "s3" rather than a

dash) if you specify custom endpoints with this format during installation. This latter format is used by AWS S3 currently.

The table below shows the default format of each service endpoint. The examples show the service endpoints that the system would automatically generate if the domain is *enterprise.com* and the region name is *boston*.

Service Endpoint	Default Format and Example	Description
S3 service endpoint (**one per service region**)	*s3-<regionname>.<domain>* *s3-boston.enterprise.com*	This is the service endpoint to which S3 client applications will submit requests. If you are installing a HyperStore system across multiple service regions, each region will have its own S3 service endpoint, and therefore you must create a DNS entry for each of those region-specific endpoints — for example *s3-boston.enterprise.com* and *s3-chicago.enterprise.com*.
S3 service endpoint wildcard (**one per service region**)	**.s3-<regionname>.<domain>* **.s3-boston.enterprise.com*	This S3 service endpoint wildcard entry is necessary to resolve virtual-hosted-style S3 requests, wherein the bucket name is specified as a sub-domain -- for example *bucket1.s3-boston.enterprise.com* and *bucket2.s3-boston.enterprise.com* and so on.
S3 static website endpoint (**one per service region**)	*s3-website-<regionname>.<domain>* *s3-website-boston.enterprise.com*	This S3 service endpoint is used for buckets configured as static websites. **Note** Your S3 static website endpoint cannot be the same as your S3 service endpoint. They must be different, or else the website endpoint will not work properly.
S3 static website endpoint wildcard (**one per service region**)	**.s3-website-<regionname>.<domain>* **.s3-website-boston.enterprise.com*	This S3 static website endpoint wildcard entry is necessary to resolve virtual-hosted-style S3 requests, wherein the bucket name is specified as a sub-domain, for buckets configured as static websites.
Admin Service endpoint (**one per entire system**)	*s3-admin.<domain>* *s3-admin.enterprise.com*	This is the service endpoint for HyperStore's Admin API. The Cloudian Management Console accesses this API, and you can also access this API directly with a third party client (such as a command line tool like *cURL*).
IAM Service endpoint (**one per entire**	*iam.<domain>* *iam.enterprise.com*	This is the service endpoint for accessing HyperStore's implementation of the Identity and Access Management API.

Service Endpoint	Default Format and Example	Description
system)		**Note** The IAM service endpoint can also be used for making Security Token Service (STS) API calls.
SQS Service endpoint (**one per entire system**)	*sqs.<domain>* *sqs.enterprise.com*	This is the service endpoint for accessing HyperStore's implementation of the Simple Queue Service (SQS) API. **Note** The SQS Service is disabled by default. For information about enabling this service, see the SQS section of the ***Cloudian HyperStore AWS APIs Support Reference***.
Cloudian Management Console (CMC) endpoint (**one per entire system**)	*cmc.<domain>* *cmc.enterprise.com*	The CMC is HyperStore's web-based console for performing system administrative tasks. The CMC also supports actions such as creating storage buckets or uploading objects into buckets.

2.1.2. Configuring Resolution of Service Endpoints

IMPORTANT ! Cloudian Best Practices suggest that a highly available load balancer be used in production environments where consistent performance behavior is desirable. For environments where a load balancer is unavailable, other options are possible. Please consult with your Cloudian Sales Engineer for alternatives.

For a production environment, in your DNS configuration each HyperStore service endpoint should resolve to the virtual IP address(es) of two or more load balancers that are configured for high availability. For more detail see **"Load Balancing"** (page 13).

2.1.3. Using Customized Service Endpoints

If you do not want to use the default service endpoint formats, the HyperStore system allows you to specify custom endpoint values during the installation process. If you intend to create custom endpoints, configure DNS entries to resolve the custom endpoint values that you intend to use, rather than the default-formatted endpoint values shown in the **"HyperStore Service Endpoints"** (page 10) table. Make a note of the custom endpoints for which you configure DNS entries, so that later you can correctly specify those custom endpoints when you perform the HyperStore installation.

If you want to use a **custom S3 endpoint** that does not include a region string, the installer allows you to do so. Note however that if your S3 endpoints lack region strings the system will not be able to support the region name validation aspect of AWS Signature Version 4 authentication for S3 requests (but requests can still succeed without the validation).

If you want to use **multiple S3 endpoints per service region** -- for example, having different S3 endpoints resolve to different data centers within one service region -- the installer allows you to do this. For this approach, the recommended syntax is *s3-<regionname>.<dcname>.<domain>* -- for example *s3-boston.dc1.enterprise.com* and *s3-boston.dc2.enterprise.com*.

Note If you want to change HyperStore service endpoints after the system has already been installed, you can do so as described in the "Changing S3, Admin, CMC, or IAM Service Endpoints" section of the ***Cloudian HyperStore Administrator's Guide***. If you change any endpoints, be sure to update your DNS configuration.

2.2. Load Balancing

IMPORTANT ! Cloudian recommends that a highly available load balancer be used in production environments where consistent performance behavior is desirable. For environments where a load balancer is unavailable, other options are possible (including using HyperStore's built-in hsDNS component in combination with DNS delegation --for more information see "hsDNS Feature Overview" in the ***Cloudian HyperStore Administrator's Guide***). Please consult with your Cloudian Sales Engineer. The discussion below assumes that you are using a load balancer.

HyperStore uses a peer-to-peer architecture in which each node in the cluster can service requests to the S3, Admin, CMC, IAM, STS, and SQS service endpoints. In a production environment you should use load balancers to distribute S3, Admin, CMC, IAM, STS, and SQS service endpoint requests evenly across all the nodes in your cluster. In your DNS configuration the S3, Admin, CMC, IAM, STS, and SQS service endpoints should resolve to the virtual IP address(es) of your load balancers; and the load balancers should in turn distribute request traffic across all your nodes. Cloudian recommends that you **set up your load balancers prior to installing the HyperStore system**.

For high availability it is preferable to use two or more load balancers configured for failover between them (as versus having just one load balancer which would then constitute a single point of failure). The load balancers could be commercial products or you can use open source technologies such as **HAProxy** (load balancer software for TCP/HTTP applications) and **Keepalived** (for failover between two or more load balancer nodes). If you use software-defined solutions such as these open source products, for best performance you should install them on dedicated load balancing nodes -- not on any of your HyperStore nodes.

For a **single-region HyperStore system**, for each service configure the load balancers to distribute request traffic across all the nodes in the system.

For a **multi-region HyperStore system**:

- Configure each region's S3 service endpoint to resolve to load balancers in that region, which distribute traffic across all the nodes within that region.
- Configure the Admin, IAM, STS, SQS, and CMC service endpoints to resolve to load balancers in the **default service region**, which distribute traffic to all the nodes in the default service region. (You will specify a default service region during the HyperStore installation process. For example, you might have service regions *boston* and *chicago*, and during installation you can specify that *boston* is the default service region.)

For detailed guidance on load balancing set-up, request a copy of the ***HyperStore Load Balancing Best Practice Guide*** from your Cloudian Sales Engineering representative.

Note The HyperStore S3 Service supports **PROXY Protocol** for incoming connections from a load balancer. This is disabled by default, but after HyperStore installation is complete you can enable it by configuration if you wish. For more information search on "s3.enable.proxyProtocol" in the ***Cloudian HyperStore Administrator's Guide***.

Note For information about how to perform health checks of HyperStore's HTTP(S) services such as the S3 Service and the CMC, see the "Checking HTTP(S) Responsiveness" section in the ***Cloudian HyperStore Administrator's Guide***.

Chapter 3. Preparing Your Nodes

This section covers these topics to help you select and prepare your HyperStore host machines:

3.1. Host Hardware and Operating System

Subjects covered in this section:

These host recommendations and requirements apply if you are doing a fresh installation of a HyperStore system or if you are adding new nodes to an existing HyperStore system.

3.1.1. Hardware Recommendations

The table below shows the recommended specifications for individual host machines in a HyperStore system.

	Recommended Server Spec		Recommended VM Spec	
Component	**Description**	**Qty**	**Description**	**Qty**
Processors	Intel Xeon 5218R or better	1-2	20 core CPU or greater	1-2
RAM	128GB or greater	4-12 DIMM slots	128GB or greater	1
OS Drives	NVMe Storage 480GB or greater	2	NVMe Storage 480GB or greater	2
Flash Metadata Drives	NVMe Storage 1920GB or greater	2	SAS SSD / NVMe Storage 1920GB or greater	2
Object Data Capacity Drives	7200RPM SAS 512e HDD or PCIe4 NVMe Flash/SSD	4-45	7200RPM SAS 512e HDD or PCIe4 NVMe Flash/SSD	4-12
Networking	Dual-port 10/25GbE	1-2	Dual-port 10/25GbE	1-2

Note that:

- For production environments the recommended specifications in the table are a starting point. For guidance on scaling a HyperStore deployment to meet your workload requirements, consult with your Cloudian sales representative.
- HyperStore supports using **Quad-Level Cell (QLC) SSDs** as the object data capacity drives, so long as those drives are large capacity drives (15TB or higher). Consult with your Cloudian sales representative for details about the supported QLC SSD types and sizes. QLC SSDs are **not** recommended for using as the metadata drives, since these will be subject to very frequent writes of small files. By contrast, QLC SSDs are best suited for read-heavy workloads and large objects. Note that if

you are using QLC SSDs as your object data drives:

- It is recommended that you have two non-QLC NVMe drives dedicated to HyperStore metadata storage, as indicated in the table above.
- When you launch the HyperStore installer (as described in **"Installing a New HyperStore System" (page 27)**) use the *configure-qlc-rdb* command line option.

- For virtualization environments, running HyperStore on VMware ESXi and vSphere is supported so long as the VMs have specs meeting or exceeding those in the table. However, avoid KVM or Xen as there are known problems with running HyperStore in those virtualization environments. For more guidance on deploying HyperStore on VMware, ask your Cloudian representative for the "Best Practices Guide: Virtualized Cloudian HyperStore on VMware vSphere and ESXi".

3.1.2. Operating System Requirements

To perform a fresh installation of HyperStore 8.2.2, on each of your host machines the operating system must be **Rocky Linux 8.9 or 8.10** (8.10 is recommended). Alternatively your hosts can be running Red Hat Enterprise Linux 8.9 or 8.10. HyperStore 8.2.2 does **not** support fresh installation on:

- Other versions of Rocky Linux or RHEL
- CentOS
- Other types of Linux distribution
- Non-Linux operating systems

If you have not already done so, for each host machine on which you intend to install HyperStore, install Rocky Linux 8.9 or 8.10 (or RHEL 8.9 or 8.10) in accordance with your hardware manufacturer's recommendations.

Below, see these additional requirements related to host systems on which you intend to install HyperStore:

- **"Partitioning of Disks Used for the OS and Metadata Storage" (page 16)**
- **"Host Firewall Services Must Be Disabled" (page 17)**
- **"Do Not Mount /tmp Directory with 'noexec'" (page 17)**
- **"root User umask Must Be 0022" (page 17)**
- **"Note: Automatic Exclusions to OS Package Updates" (page 17)**

3.1.2.1. Partitioning of Disks Used for the OS and Metadata Storage

For the disks used for the OS and metadata storage -- typically two mirrored SSDs as noted in the hardware requirements table above -- **do not accept the default partition schemes offered by Rockly Linux / RHEL**:

- By default Rocky Linux / RHEL allocates a large portion of disk space to a */home* partition. This will leave inadequate space for HyperStore metadata storage.
- By default Rocky Linux / RHEL proposes using LVM. Cloudian recommends using standard partitions instead.

Cloudian recommends that you manually create a partition scheme like this:

For Software RAID

- 1x 1G as */boot*, Device Type RAID1, label *boot*, fs: *ext4*
- 1x 8G as SWAP, Device Type RAID1, label *swap*
- 1x remaining space as */* , Device Type RAID 1, label *root*, fs: *ext4*

For Hardware SUDO RAID with UEFI

- 1x 1G as */boot/efi*, label *efi*
- 1x 1G as */boot*, label *boot*, fs: *ext4*
- 1x 8G as SWAP, label *swap*
- 1x remaining space as */* , label *root*, fs: *ext4*

3.1.2.2. Host Firewall Services Must Be Disabled

To install HyperStore the following services **must be disabled on each HyperStore host machine**:

- *firewalld*
- *iptables*
- *SELinux*

To disable ***filewalld***:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

Rocky Linux / RHEL uses *firewalld* by default rather than the *iptables* service (*firewalld* uses *iptables* commands but the *iptables* service itself is not installed on Rocky Linux / RHEL by default). So you do not need to take action in regard to ***iptables*** unless you installed and enabled the *iptables* service on your hosts. If that's the case, then disable the *iptables* service.

To disable **SELinux**, edit the configuration file */etc/selinux/config* so that *SELINUX=disabled*. Save your change and then restart the host.

HyperStore includes a built-in firewall service (a HyperStore-custom version of the *firewalld* service) that is configured to protect HyperStore internal services while keeping HyperStore public services open. For more information see the "HyperStore Firewall" section in the ***Cloudian HyperStore Administrator's Guide***.

> **Note** For information about HyperStore port usage see **"HyperStore Listening Ports"** (page 50).

3.1.2.3. Do Not Mount */tmp* Directory with 'noexec'

The */tmp* directory on your host machines must not be mounted with the '*noexec*' option. If the */tmp* directory is mounted with '*noexec*', you will not be able to extract the HyperStore product package file and the HyperStore installer (installation script) will not function properly.

3.1.2.4. root User umask Must Be 0022

On hosts on which you will install HyperStore, the *root* user umask value must be '0022' (which is the default on Linux hosts). If the *root* user umask is other than '0022' the HyperStore installation will abort.

3.1.2.5. Note: Automatic Exclusions to OS Package Updates

As part of HyperStore installation, the HyperStore installation script will install prerequisites including Java and Salt on your HyperStore host machines. If you subsequently use *dnf* to update your OS packages, HyperStore automatically excludes Java and Salt related packages from the update. This is to ensure that only the correct, tested versions of these packages are used together with HyperStore. After HyperStore installation, this auto-

exclusion is configured in the */etc/dnf/plugins/versionlock.list* file on your host machines. You can review that file if you wish to see specifically which packages are "locked" at which versions, but do not remove any entries from the lock list.

3.1.3. Cloudian COSI Driver for Kubernetes

After you've deployed a HyperStore cluster, if your organization is also operating one or more Kubernetes clusters be aware that Cloudian has developed an open source Cloudian COSI driver for Kubernetes.

Devised by the Kubernetes organization, the Container Object Storage Interface (COSI) enables Kubernetes based applications to use the Kubernetes API to provision, configure, and manage buckets in an external S3-compatible object storage system. A key component in the COSI framework is COSI drivers. Operating in the Kubernetes cluster, a COSI driver is specific to a particular S3 object storage vendor, accepting incoming vender-neutral COSI requests from other components in the framework and converting those requests into calls to that particular external S3 object storage system.

The Cloudian COSI Driver is open sourced and is available at the following Github repository:

https://github.com/cloudian/cloudian-cosi-driver

That repository includes a "Getting Started" document (referenced in the repo's README file).

3.2. Preparing Your Nodes For HyperStore Installation

Subjects covered in this section:

Note These instructions assume that you have already configured basic networking on each of your nodes. In particular, each node must already be configured with a hostname and IPv4 address, and the nodes must be able to reach each other in the network.

3.2.1. Installing HyperStore Prerequisites

After verifying that your nodes meet HyperStore's **hardware and OS requirements**, follow these steps to install and configure HyperStore prerequisites on all of your nodes:

1. Log into any one of your nodes as *root*. This will be the node through which you will orchestrate the HyperStore installation for your whole cluster. Also, as part of the HyperStore installation, **Salt** configuration management software will be installed and configured in the cluster, and this HyperStore node will become the **Config Controller node** for purposes of ongoing cluster configuration management. Note that the Config Controller node must be one of your HyperStore nodes. It cannot be a separate node outside of your HyperStore cluster.
2. On the node that you've logged into, download or copy the HyperStore product package (*CloudianHyperStore-8.2.2.bin* file) into a working directory. Also copy your Cloudian license file (**.lic* file) into that same directory. Pay attention to the license file name since you will need the license file name in the next step.

> **Note** The license file must be your cluster-wide license that you have obtained from Cloudian, not a license for an individual HyperStore Appliance machine (not a *cloudian_appliance.lic* file).

3. In the working directory run the commands below to unpack the HyperStore package:

```
# chmod +x CloudianHyperStore-8.2.2.bin
# ./CloudianHyperStore-8.2.2.bin <license-file-name>
```

This creates an **installation staging directory** named */opt/cloudian-staging/8.2.2*, and extracts the HyperStore package contents into the installation staging directory.

> **Note** The installation staging directory must persist for the life of your HyperStore system. Do not delete the installation staging directory after completing the install.

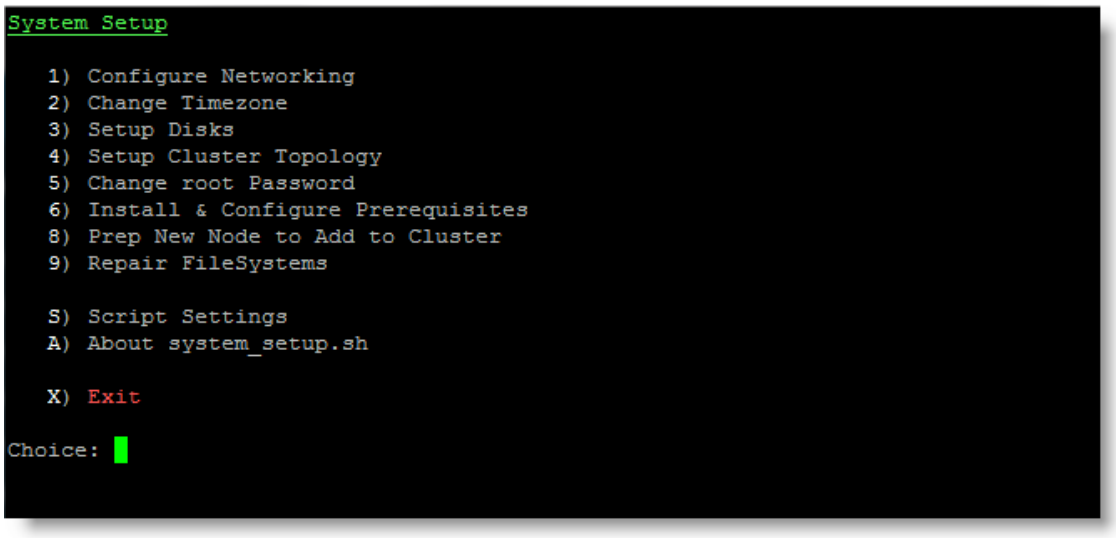
4. Change into the installation staging directory:

```
# cd /opt/cloudian-staging/8.2.2
```

5. In the installation staging directory, launch the *system_setup.sh* tool:

```
# ./system_setup.sh
```

This displays the tool's main menu.



6. From the setup tool's main menu, choose ***4)*** *Setup Cluster Topology*. This will display the Topology Editor menu. From this menu choose ***a)*** *Generate a new topology file for Object nodes*. Then provide all the information for which the tool prompts you: .

Category	Prompt for Information	Description
Cluster information	Domain name	Your organization's domain name. For example, *enterprise.com*.
	Region name	Name of the service region in which you are currently installing HyperStore nodes. The HyperStore system supports having multiple service regions with each region having its own independent S3 object inventory, and with S3 users able to choose a region when

Category	Prompt for Information	Description
		they create buckets. Even if you will have only one region you must give it a name. Only lower case alphanumeric characters and dashes are allowed in region names; and the first character must be alphanumeric (the name cannot start with a dash). The minimum region name length is 3 characters and the maximum length is 52 characters. Do not include the string "s3" in the region name. Make sure the region name matches the region name that you use in your S3 endpoints in your **"DNS Set-Up"** (page 10). Note that if you intend to have multiple service regions in your HyperStore system, this first region that you are installing will be the "**default region**". After completing the installation of this first service region you can subsequently use the CMC to add more service regions if you wish, as described in the "Operations" section of the system administration documentation.
	Data center name	Name of the data center in which you are currently installing HyperStore nodes. Only alphanumeric characters and dashes are allowed in the name; and the first character must be alphanumeric (the name cannot start with a dash). Note that after entering the data center name you will be prompted for the name of the region in which the data center is located, and this must be the name of the region that you had entered at the earlier region name prompt (above). After completing the installation of this first data center's nodes you can subsequently use the CMC to add more data centers if you wish, as described in the "Operations" section of the system administration documentation.
	Rack name	Name of the rack in which you are currently installing HyperStore nodes. Only alphanumeric characters and dashes are allowed in the name. **Note** In this topology use the same rack name for all the HyperStore nodes within a data center (even if the nodes are in different physical racks).
Node information	Node name	This must be the 'short' hostname, not an FQDN. After entering the node name you will be prompted for the name of the data center in which the node is located, and this must be the name of the data center that you had entered at the earlier data center name prompt (above).

Category	Prompt for Information	Description
		Note Each node must have a short hostname that is unique within your entire HyperStore system, even if you will eventually deploy multiple data centers and service regions.
	Node IP address	IP address of the node. This must be the physical IP address assigned to the host's front end port, and it must be an IPv4 address. IPv6 is not supported.
	Internal interface	For each node, you can optionally specify the name of the interface that the node uses for internal cluster communications: • For each node for which you do **not** specify an internal interface name here in the topology, HyperStore will use a default internal interface name that you will supply later in the HyperStore installation process. • For each node for which you **do** specify an internal interface name here in the topology, HyperStore will use that internal interface name for that node. The node-specific internal interface name in the topology overrides the default internal interface name that you will supply later in the HyperStore installation process.
	Add another node?	This yes/no prompt gives you the option to loop back through the node information prompts again, to include more nodes in your initial HyperStore cluster installation. Complete this process for each node in your initial HyperStore cluster, including the Config Controller node (the node from which you are directing the cluster installation).

After you've provided the information for every node in your HyperStore cluster, and you enter "n" (for no) at the "Add another node?" prompt, the Topology Editor menu displays again. Choose ***d)*** *Show topology entries* to review your node entries. After doing so, back at the Topology Editor menu choose ***f)*** *Save topology file* if the entries are correct, or ***e)*** *Edit topology* if you need to make corrections to any node entries. Once your topology is correct and saved, return to the system setup tool's main menu.

Note Based on your input at the prompts, the setup tool generates a topology file named *topology.yaml* in your installation staging directory.

7. Back at the system setup tool's main menu, if you want to change the root password for your nodes you can do so by choosing ***5)*** *Change Root Password* and following the prompts. It's recommended to use the same root password for each node. Otherwise the pre-installation cluster validation tool described later in the procedure will not be fully functional.

> **Note** If your host machines are "hardened" HyperStore Appliances -- which have the HyperStore Shell already enabled and the root password disabled -- then the "Change Root Password" option will not appear in the setup tool's main menu.

8. Back at the setup tool's main menu choose *Install & Configure Prerequisites*. When prompted about whether you want to perform this action for all nodes in your topology file enter "**yes**". The tool will connect to each of your nodes in turn and install the prerequisite packages. You will be prompted to provide the root password for the cluster nodes (unless an SSH key is present, in which case that will be used rather than a password). When the prerequisite installation completes for all nodes, return to the setup tool's main menu.

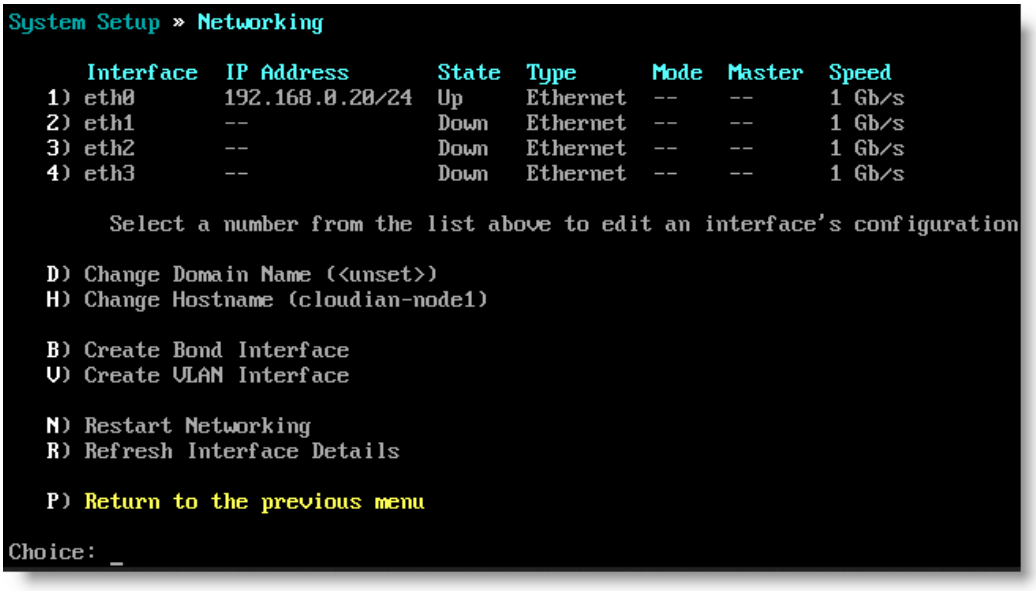
> **Note** If *firewalld* is running on your hosts the setup tool prompts you for permission to disable it. And if *Selinux* is enabled on your hosts, the tool automatically disables it without prompting for permission (or more specifically, changes it to "permissive" mode for the current running session and changes the configuration so it will be disabled for future boots of the hosts). For information on why these services must be disabled on HyperStore host machines see **"Operating System Requirements"** (page 16).

After the prerequisite installation completes for all nodes (as indicated by terminal messages from the setup tool), return to the setup tool's main menu and proceed to **"Configuring Network Interfaces, Time Zone, and Data Disks"** (page 22).

3.2.2. Configuring Network Interfaces, Time Zone, and Data Disks

After **"Installing HyperStore Prerequisites"** (page 18), you should be at the main menu of the *system_setup.sh* tool. Next follow these steps to configure network interfaces (if you haven't already fully configured them), set the time zone, and configure data disks on each node in your HyperStore cluster.

1. From the system setup tool's main menu, complete the setup of the Config Controller node itself:
   a. From the system setup tool's main menu, choose ***1)*** *Configure Networking*. This displays the Networking configuration menu.

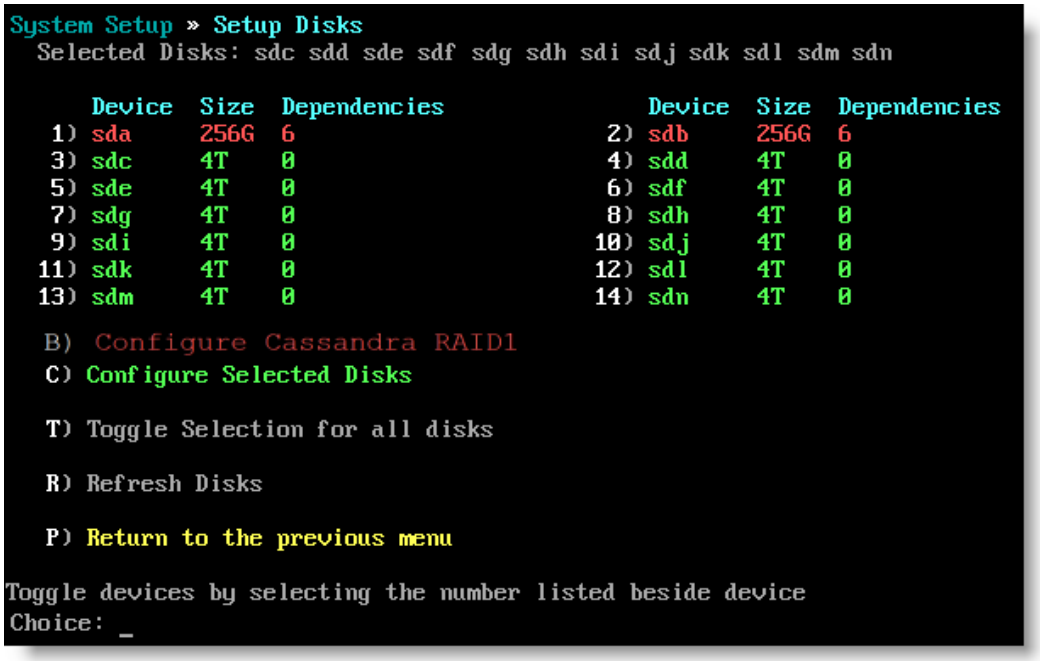


Here you can review the current network interface configuration for the Config Controller node, and if you wish, perform additional configuration such as configuring an internal/back-end interface. When you are done with any desired network interface configuration changes for this node, return to the setup tool's main menu.

> **Note** When setting/changing a node's hostname, if you enter a hostname that includes upper case letters the setup tool automatically converts the hostname to entirely lower case letters.

b. From the setup tool's main menu, choose ***2)*** *Change Timezone* and set the time zone for this node.

c. From the setup tool's main menu, choose ***3)*** *Setup Disks*. This displays the Setup Disks menu.



From the list of disks on the node select the disks to format as HyperStore data disks, for storage of S3 object data. By default the tool automatically selects all disks that are not already mounted and do not contain a */root*, */boot* or *[swap]* mount indication. Selected disks display in green font in the disk list. The tool will format these disks with *ext4* file systems and assign them mount points */cloudian1*, */cloudian2*, */cloudian3*, and so on. You can toggle (select/deselect) a disk by entering at the prompt the disk's number from the displayed list. Once you're satisfied with the selected list in green font, choose ***c)*** *Configure Selected Disks* and follow the prompts to have the tool configure the selected disks.

> **Note** In HyperStore version 8.1.1 and later, the Setup Disks menu also displays an option **b** for *Configure Cassandra RAID1*, which allows you to configure two same-sized drives into a RAID1 configuration for storage of the Metadata DB (in Cassandra). The recommended configuration is that these be NVMe drives dedicated to metadata storage (not the drives on which the OS is stored).

> **IMPORTANT !** Cloudian recommends using the *system_setup.sh* tool to format and mount your data disks (as described above). **If you have already formatted and mounted your data disks using third party tools**, then instead of using the disk configuration instructions in this section follow the guidelines and instructions in **"File System Requirements"** (page 55).

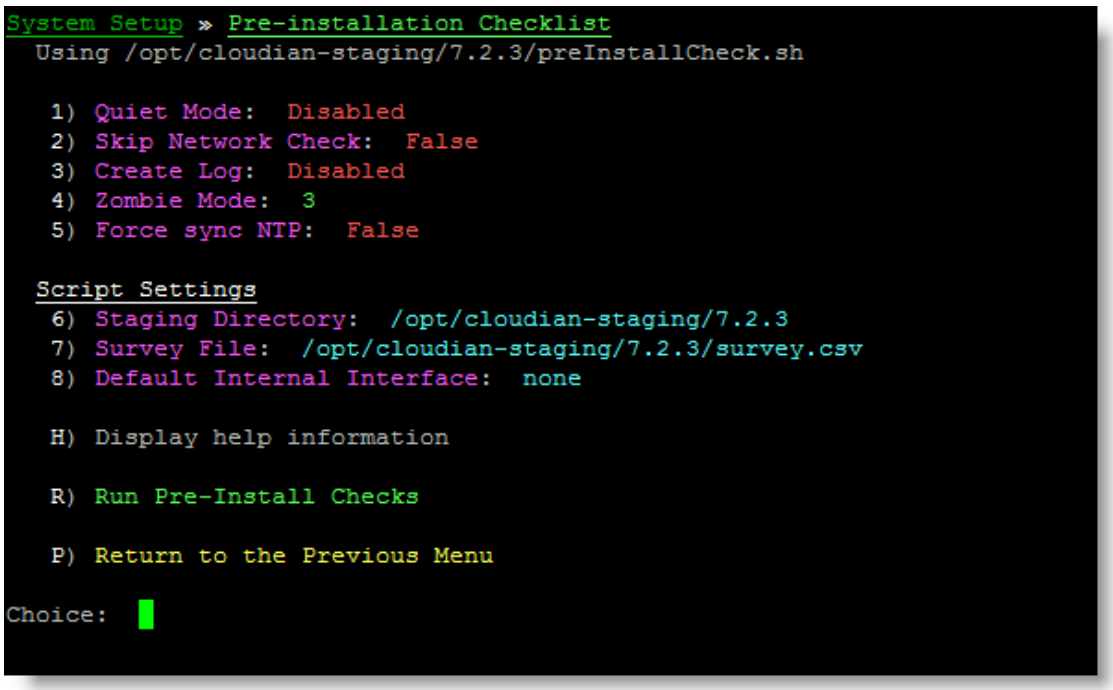
2. Next, complete the setup of the other nodes in your cluster:
    a. From the setup tool's main menu choose *Prep New Node to Add to Cluster*.
    b. When prompted enter the IP address of one of the remaining nodes (the nodes other than the Config Controller node), and then enter the login password for the node.
    c. Using the node preparation menu that displays:
        i. Review and complete network interface configuration for the node.
        ii. Set the time zone for the node.
        iii. Configure data disks for the node. Then return to the system setup tool's main menu.
    d. Repeat Steps "a" through "c" for each of the remaining nodes in your installation cluster.

After you've prepared all your nodes and returned to the setup tool's main menu, proceed to **"Running the Pre-Install Checks Script"** (page 24).

3.2.3. Running the Pre-Install Checks Script

Follow these steps to verify that your cluster now meets all HyperStore requirements for hardware, prerequisite packages, and network connectivity.

1. At the setup tool's main menu choose ***R**) Run Pre-Installation Checks*. This displays the Pre-Installation Checklist menu.



2. From the Pre-Installation Checklist menu choose ***r) Run Pre-Install Checks***. After prompting you for a cluster password the script checks to verify that your cluster meets all requirements for hardware, pre-requisite packages, and network connectivity.

   > **Note** The script only supports your providing one root password, so if some of your nodes do not use that password the script will not be able to check them and you may encounter errors during HyperStore installation if requirements are not met.

   At the end of its run the script outputs to the console a list of items that the script has evaluated and the results of the evaluation. You should review any "Warning" items but they don't necessarily require action (an example is if the hardware specs are less than recommended but still adequate for the installation to proceed). **You must resolve any "Error" items before performing the HyperStore software installation**, or the installation will fail.

   When you're done reviewing the results, press any key to continue and then exit the setup script. **If you make any system changes to resolve errors found by the pre-install check, run the pre-install check again afterward** to verify that your environment meets HyperStore requirements.

After your cluster has successfully passed the pre-install checks, proceed to **"Installing a New HyperStore System"** (page 27).

This page left intentionally blank

Chapter 4. Installing a New HyperStore System

This section describes how to do a fresh installation of HyperStore 8.2.2 software, **after** **"Preparing Your Environment"** (page 9) and **"Preparing Your Nodes For HyperStore Installation"** (page 18). From your Config Controller node you can install HyperStore software across your whole cluster.

1. On your Config Controller node, in your installation staging directory (*/opt/cloudian-staging/8.2.2*), launch the HyperStore installer. The basic launch command is as follows, but before launching the installer review the notes below regarding a couple of significant launch command options.

   ```
   [8.2.2]# ./cloudianInstall.sh
   ```

   Notes about launch command options:

   - If you want to use HyperStore's **hsDNS** component to resolve HyperStore service endpoints within the cluster (and also to resolve HyperStore service endpoints for external clients if you set up DNS delegation), launch the install script with the *configure-hsdns* option as shown below:

     ```
     [8.2.2]# ./cloudianInstall.sh configure-hsdns
     ```

     For more information about hsDNS, in the "Service Features" section of your *Cloudian HyperStore Administrator's Guide*, see "hsDNS".

   - If you are using **Quad-Level Cell (QLC) SSDs** as your object data drives, launch the install script with the *configure-qlcdata* option as shown below:

     ```
     [8.2.2]# ./cloudianInstall.sh configure-qlc-rdb
     ```

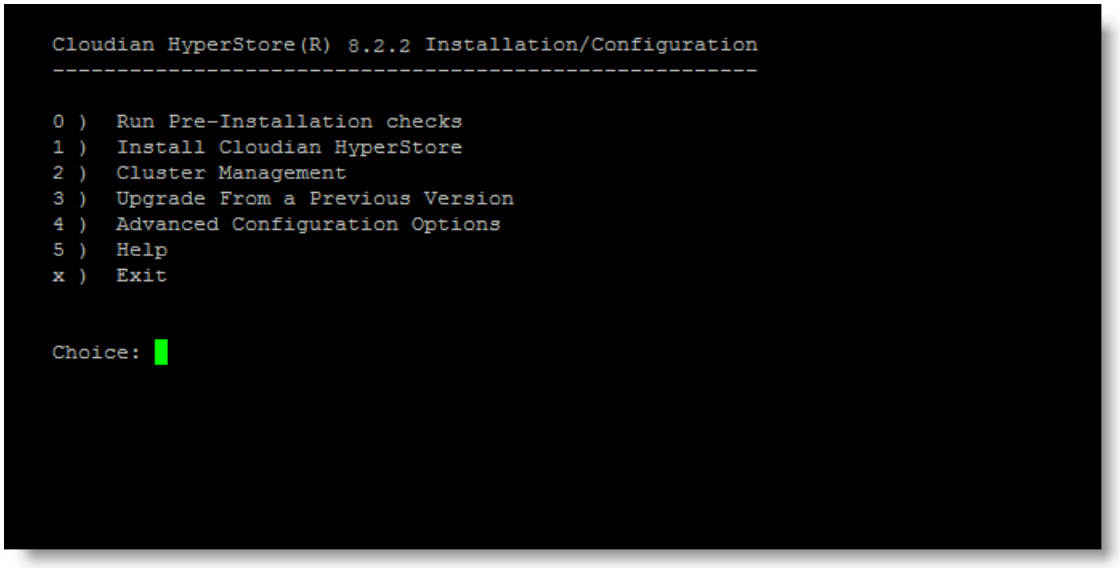
     Note If you use this option, then during the installation you will be prompted for the mount point for your Cassandra (Metadata DB) NVMe drives. RocksDB commit logs will be written to a */rdb* sub-directory under this mount point (rather than to the object data drives, as is done for a system that does not use QLC SSDs as the data drives).

   - You can use both options if you wish:

     ```
     [8.2.2]# ./cloudianInstall.sh configure-hsdns configure-qlc-rdb
     ```

   - For additional, less common script launch options, see the Installation Reference topic **"cloudianInstall.sh"** (page 59).

   When you launch the installer the main menu displays:



Note The installer menu includes an item "0" for Run Pre-Installation Checks. This is the same pre-installation check that you already ran from within the *system_setup.sh* tool as described in **"Running the Pre-Install Checks Script"** (page 24) -- so you can ignore this option in the installer menu. If you did **not** run the pre-install check already, then do so from the installer menu before proceeding any further.

2. From the installer main menu, enter "**1**" for Install Cloudian HyperStore. Follow the prompts to perform the HyperStore installation across all the nodes in your cluster topology file (which you created earlier during the node preparation task).

   During the HyperStore installation you will be prompted to provide cluster configuration information including the following:

   - The name of the **internal interface** that your nodes will use by default for internal cluster communications (communications between HyperStore nodes). For example, *eth1*.

     Note The system will use this default internal interface name for all nodes for which you did not specify an internal interface name in your cluster topology file (which you created during the **"Installing HyperStore Prerequisites"** (page 18) procedure). If in the topology file you specified internal interface names for some or all of your nodes, the system will use those internal interface names for those nodes, rather than the default internal interface name.

   - The starting "**replication strategy**" that you want to use to protect system metadata (such as usage reporting data and user account information). The replication strategy you enter must be formatted as "<datacenter_name>:<replication_#>". For example, "DC1:3" means that in the data center named DC1, three instances of each system metadata object will be stored (with each instance on a different host). If you are installing HyperStore into multiple data centers you must format this as a comma-separated list specifying the replicas per data center -- for example "DC1:3,DC2:3". The default is 3 replicas per service region, and then subsequently the system automatically adjusts the system metadata replication level based on the storage policies that you create. For more on this topic see "System Metadata Replication" in the ***Cloudian***

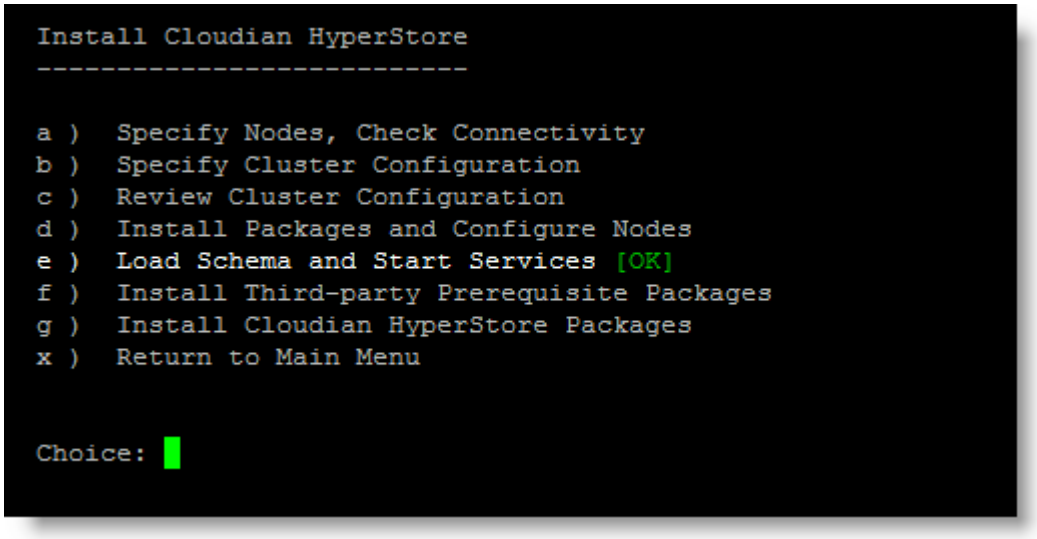
HyperStore Administrator's Guide.

- Your **organization's domain**. For example, *enterprise.com*. From this input that you provide, the installation script will automatically derive HyperStore service endpoint values. You can accept the derived endpoint values that the script presents to you, or optionally you can enter customized endpoint values at the prompts. For S3 service endpoint the default is to have one endpoint per service region, but you also have the option of entering multiple comma-separated endpoints within a service region -- if for example you want different data centers within the region to use different S3 service endpoints. If you want to have different S3 endpoints for different data centers within the same service region, the recommended S3 endpoint syntax is *s3-<region>.<dcname>.<domain>*. See **"DNS Set-Up"** (page 10) for more details about HyperStore service endpoints. Note that if you have multiple S3 service endpoints for your system, the first S3 service endpoint in your comma-separated list will be the "default" S3 service endpoint (and so will be used in public URLs if CMC users generate public URLs for some of their objects).

  IMPORTANT !
  * Do not use IP addresses in your service endpoints.
  * Including "s3" in the *<domain>* value is not recommended. By default HyperStore generates S3 service endpoints by prepending an "*s3-*" prefix to your *<region-name>.<domain>* combination. If you include "s3" within either your domain or your region name, this will result in two instances of "s3" in the system-generated S3 service endpoints, and this may cause S3 service requests to fail for some S3 clients.
  * HyperStore by default derives the S3 service endpoint(s) as *s3-<region-name>.<domain>*. However HyperStore also supports the format *s3.<regionname>.<domain>* (with a dot after the "s3" rather than a dash) if you specify a custom S3 endpoint with this format..
  * Your S3 static website endpoint cannot be the same as your S3 service endpoint. They must be different, or else the static website endpoint will not work properly.

- The **NTP servers** that HyperStore nodes should connect to for time synchronization. By default the public servers from the *pool.ntp.org* project are used. If you do not allow outbound connectivity from HyperStore hosts (and consequently public NTP servers cannot be reached) you must specify NTP server(s) within your environment that HyperStore hosts should connect to instead. The installation will fail if HyperStore hosts cannot connect to an NTP server.

At the conclusion of the installation an "Install Cloudian HyperStore" sub-menu displays, with indication of the installation status. If the installation completed successfully, the "Load Schema and Start Services" menu item should show an "OK" status:



After seeing that the "Load Schema and Start Services" status is OK, return to the installer's main menu.

> **Note** The "Install Cloudian HyperStore" sub-menu supports re-executing specific installation operations on specific nodes or on all nodes. This may be helpful if the installer interface indicates that an operation has failed. If one of the operations in the menu indicates an error status, retry that operation by specifying the menu option letter at the prompt (such as "**e**" for "Load Schema and Start Services"). **If the installation fails because you entered incorrect information in your cluster topology file** (during the process of **"Preparing Your Nodes For HyperStore Installation"** (page 18)) -- such as an incorrect and unresolvable IP address -- you can use the *system_setup.sh* tool to edit the topology file and then try the installation again. To do so, first exit the installer. Then launch *./system_setup.sh*, and from that tool's Topology Editor menu first choose ***c**) Load existing topology file*, then choose ***e**) Edit topology* and edit the topology entries as needed. After saving the edited topology file, exit the *system_setup.sh* tool and then launch *./cloudianInstall.sh* and try again to perform the install.

3. After installation has completed successfully, from the installer's main menu enter "**2**" for Cluster Management and then enter "**d**" for Run Validation Tests. This executes some basic automated tests to confirm that your HyperStore system is working properly. The tests include S3 operations such as creating an S3 user group, creating an S3 user, creating a storage bucket for that user, and uploading and downloading an S3 object.

After validation tests complete successfully, exit the installation tool.

> **Note** For troubleshooting information, see the Installation Reference topic **"Installation Error Logging"** (page 50).

> **Note** At any time after installation, if you wish you can output the topology files currently used by your active HyperStore system: the main object storage system topology plus (if applicable) File Service topology and Search Service topology. To do so, in a working directory -- not the installation staging directory -- run the command *hsctl topology dump*. Note that if you were to run the command in the installation staging directory, and if your current system topology is different from your originally installed

typology -- for example if you've added nodes to the system -- the dump will overwrite the original version of your topology files with the current version.

This page left intentionally blank

Chapter 5. Installing the HyperStore File Service

Subjects covered in this section:

Note For instructions on **upgrading** to HyperStore File Service 8.2.2 from an older HyperStore File Service version see Chapter 3 "Upgrading Your HyperStore Software Version" (particularly the sub-section on Upgrading HyperStore File Services) in the ***Cloudian HyperStore Administrator's Guide***.

5.1. Introduction

After you've installed your HyperStore system and verified that it's operating properly -- or anytime after you've put your HyperStore system into operation -- you can optionally install the HyperStore File Service. Once you've installed the HyperStore File Service in your HyperStore cluster, you can choose HyperStore users for whom to enable File Services functionality. Those users can then use the CMC to create and manage SMB and/or NFS file shares. The HyperStore File Service provides front end SMB and NFS access to these shares as well as a caching layer to speed read and write performance, while the users' HyperStore buckets provide back end archival storage for the shares. For more information about this feature, in the ***Cloudian HyperStore Administrator's Guide*** see *Setting Up Service Features -> File Services (SMB/NFS)* and *Preparing the File Services Feature*.

5.1.1. File Service Pools

When you add the File Service to your HyperStore system, you can deploy one or more **File Service pools**. Each File Service pool consists of either one File Service node or three File Service nodes, residing in a single data center. One node may be sufficient for a File Service pool that will be used only for development, testing, or evaluation purposes. One node may also be sufficient for production environments that will have only a small number of SMB or NFS clients and that do not require high-availability file services. Three nodes are required for a File Service pool that will be used in a production environment that has many concurrent SMB or NFS clients, or that needs high-availability file services.

If you wish you can deploy **multiple File Service pools, in different locations**. Each File Service pool will consist of either one node or three nodes. When you grant File Service privileges to a HyperStore user you can choose which File Service pool(s) the user will have access to. If you grant a HyperStore user access to multiple File Service pools, the HyperStore user can then publish different file shares to different File Service pools (and thereby to different sets of end users), or optionally the HyperStore user can publish a **geo-distributed file share** to multiple File Service pools and HyperStore will keep the share in sync across those different locations. For more information about geo-distributed shares, in the *HyperStore Administrator's Guide* see *Setting Up Service Features -> File Services (SMB/NFS)*.

When you install the HyperStore File Service you will perform an installation procedure for each File Service pool, one pool at a time. Note that a HyperStore File Service pool **does not support**:

- Running on two hosts or on more than three hosts
- Migrating from a single-node deployment to a three-node deployment
- Running on HyperStore object storage nodes

5.2. File Service Host Requirements and Recommendations

Each HyperStore File Service host must be either a Cloudian HFA-1100 appliance, or a bare metal server or VM with comparable specifications. The table below indicates the recommended specifications for HyperStore File Service hosts.

	Recommended Server Spec		Recommended VM Spec	
Component	**Description**	**Qty**	**Description**	**Qty**
Processors	Intel Xeon 5218R or better	1-2	20 core CPU or greater	1-2
RAM	128GB or greater	4-12 DIMM Slots	128GB or greater	1
OS Drives	NVMe Storage 480GB or greater	2	NVMe Storage 480GB or greater	2
Flash Metadata Drives	NVMe Storage 1920GB or greater	2	SAS SSD / NVMe Storage 1920GB or greater	2
Flash Write Cache Drives	NVMe Storage 1920GB or greater	2	SAS SSD / NVMe Storage 1920GB or greater	2
Capacity Read Cache Drives	7200RPM SAS 512e HDD or PCIe4 NVMe Flash/SSD 10TB-14TB per disk	4-12	7200RPM SAS 512e HDD or PCIe4 NVMe Flash/SSD 10TB-14TB per disk	4-12
Networking	Dual Port 10/25GbE	1-2	Dual Port 10/25GbE	1-2

5.2.1. Operating System Requirements

If you use HFA-1100s, these appliances will come with Rocky Linux 8.10 pre-installed.

If you want to use bare metal servers or VMs you will need to obtain the HyperStore Rocky Linux ISO file from Cloudian (this ISO is based on Rocky Linux 8.10). Use this ISO to install Rocky Linux on each of the machines that will host the HyperStore File Service. **Do not use the generic Rocky Linux ISO** from the Rocky Linux organization. You must use the HyperStore Rocky Linux ISO, in which the OS packages have been optimized for hosting Cloudian software.

The process of installing the Rocky Linux 8.10 OS from the HyperStore Rocky Linux ISO file is the same as doing a standard Rocky Linux installation, but note that:

- The BIOS needs to be configured with UEFI with secure boot disabled
- Choose 'Server' as the base install type (not Minimal or Custom OS)
- When assigning IP addresses to network interfaces, manually set static IP addresses (do not use DHCP)
- The default root password is *HSApplianceLinux8*

For more information on the Rocky Linux installation procedure, refer to the Rocky Linux website.

Note When obtaining the HyperStore Rocky Linux ISO file from Cloudian, also obtain the HyperStore File Service 8.2.2 product package (and corresponding signature file [*.sig* file]), which you will use to perform the File Service software installation as described later in this procedure.

5.2.2. Must Be Dedicated Host(s)

The HyperStore File Service requires a dedicated host or hosts. You cannot run the File Service on the same nodes as you're running HyperStore object storage software.

5.2.3. Networking Requirements

- The DNS server that you specify in each File Service host's configuration must resolve your HyperStore S3 service endpoint. If you have multiple S3 service endpoints for your HyperStore system, the DNS server must resolve all of the endpoints. The HyperStore File Service will use these endpoints when archiving file share contents to HyperStore S3 buckets. (This resolving of your HyperStore S3 service endpoint must be done through a DNS server and not just a */etc/hosts* file. Using HyperStore's *hsDNS* feature [which is disabled by default] is one way to satisfy this requirement; for more information about hsDNS see your ***Cloudian HyperStore Administrator's Guide***.)
- HyperStore File Service nodes will be part of your HyperStore cluster. Therefore, along with being accessible on your front end network (see open port requirements further below), File Service nodes must be on the same backend network as the rest of your HyperStore nodes -- if those nodes use a dedicated backend network as is the case in most HyperStore deployments. In particular, File Service nodes will need to be able to access the backend interface address of the HyperStore Config Controller node.
- On the backend network, communication among HyperStore nodes (including HyperStore File Service nodes) must be allowed on all ports.
- On the frontend network, the following ports must be open:

Port Number	Purpose	Transport Protocol(s)
22	SSH	TCP
443	File Service API server	TCP
80	File Service API server	TCP
137	Samba -- SMB over UDP for name services	UDP
138	Samba -- SMB over UDP for datagram services	UDP
139	Samba -- NetBIOS	TCP
445	Samba -- SMB protocol	TCP
111	NFS-Ganesha -- RPC (NFSv3 only)	TCP, UDP
38465	NFS-Ganesha -- MTN (NFSv3 only)	TCP, UDP
38466	NFS-Ganesha -- RQuota	TCP, UDP
38467	NFS-Ganesha -- Statd	TCP, UDP
38468	NFS-Ganesha -- NLM (NFSv3 only)	TCP, UDP
2049	NFS-Ganesha -- NFS v3/v4	TCP, UDP

5.3. Creating a File Service Pool Topology File

To deploy a File Service pool you will first create a topology file that has basic information about the pool and the host(s) on which it will run. You can easily generate this file using an interactive script that is part of HyperStore's *system_setup.sh* tool, as described in the steps below. If you want to know in advance what information you will need to provide you can look ahead to Step 3. After you've created the topology file for a File Service pool the topology file will be used as input to the File Services software installation procedure (as described in the next section, **"Installing File Service Software On the Target Host(s)"** (page 39)).

If you intend to deploy multiple File Service pools, you will deploy one File Service pool at a time -- first creating a topology file for a pool, then installing File Service software on that pool's hosts; and then repeating the topology creation and software installation procedures for the next pool.

Follow the steps below to use the HyperStore *system_setup.sh* tool to create a topology file for a File Service pool. You will then reference this topology file when executing the File Service installation as described in the next section.

Note If you prefer working directly with a *.yaml* file rather than answering prompts from the *system_setup.sh* tool, then instead of following the steps below you can edit the *sample-file-pool-1.yaml* file in the staging directory on your HyperStore Config Controller node, save the edited file as *<your-file-pool-name>.yaml*, then proceed to the instructions for **"Installing File Service Software On the Target Host(s)"** (page 39).

1. On your HyperStore Config Controller node, change into the installation staging directory (*/opt/cloudian-staging/8.2.2*) and then launch the *system_setup.sh* tool:

   ```
   # ./system_setup.sh
   ```

   If you are using the HyperStore Shell

   If you are using the **HyperStore Shell (HSH)** you can launch the system setup tool with this command:

   ```
   $ hspkg setup
   ```

   To run this command using the HSH you must be an HSH **Trusted user**.

2. From the tool's main menu choose ***4)*** *Setup Cluster Topology*. Then from the Topology Editor menu choose ***b)*** *Generate a new topology file for File Service nodes*.
3. Provide all the information for which the tool prompts you:

Category	Prompt for Information	Description
Pool name and location	File Service pool name	Name that you want to give to this File Service pool. The name must be at least 2 and no more than 32 characters long, and only lower case alphanumeric characters and dashes are supported (no other special characters and no spaces). The name must start with a letter, and must end with a letter or number. This name will be used in the topology file name (*<pool-name>.yaml*), and in the CMC interface as a way of identifying this File Service pool.
	Region name	HyperStore service region in which this File Service pool will be installed. This must be one of the

Category	Prompt for Information	Description
		service region names from your existing HyperStore system.
	Data center name	HyperStore data center in which this File Service pool will be installed. This must be one of the data center names from your existing HyperStore system. After entering the data center name you will be prompted for the name of the service region in which the data center is located -- this must be the same region as you entered at the earlier prompt.
	Rack name	Name of the rack in which this File Service pool will be installed.
Node information	Node name	Hostname of a node that will belong to this File Service pool. This must be the 'short' hostname, not an FQDN. After entering the node name you will be prompted for the name of the data center in which the node is located -- this must be the same data center as you entered at the earlier prompt.
	Node IP address	IP address of the node. This must be the physical IP address assigned to the host's front end port.
	Internal interface	The name of the internal interface that your nodes will use for internal cluster communications (communications between HyperStore nodes). For example, *eth1*.
	Add another node?	This prompt gives you the option to loop back through the node information prompts again, to add more nodes to the File Service pool. A File Service pool can have either one node or three nodes. Other node quantities such as two nodes or four nodes are not supported. If you are creating a three-node File Service pool, all nodes must be in the same data center.
File Service pool IP addresses	Kubernetes API address (only applicable to 3-node File Service pools)	An IP address available for the File Service system to use as the Kubernetes intranet address (which facilitates Kubernetes communications within the File Service pool). This IP address must be from the same subnet as the physical IP addresses of the nodes in this File Service pool.
	First SMB/NFS address	First IP address in an available IP address range large enough to fit: • One virtual IP address that the system will use as the File Service API address for this File Service pool. • One virtual IP **for each HyperStore user that you will allow to publish file shares to this File Service pool**. The system will auto-

Category	Prompt for Information	Description
		matically allocate an IP address to each File Services-enabled HyperStore user when the user publishes shares to this File Service pool, for the purpose of making the user's file shares available to end users via SMB or NFS. At the next prompt after this one you will enter the last address in this IP range. This IP address range must be from the same subnet as the physical IP addresses of the nodes in this File Service pool. Note that it's better to have this configured IP address range be larger than needed rather than to have it turn out to be too small for your needs. In the current version of HyperStore, expanding this IP range after installation of the File Service pool is difficult and requires assistance from Cloudian Support.
	Last SMB/NFS address	See description of "First SMB/NFS address" above.
Disk information	Devices for read cache	On all hosts in this File Service pool, the storage device or devices to use for the pool's read cache (for example */dev/sda*). End users' SMB or NFS read requests are serviced from the read cache if the data is present in the cache; if not present, the data is retrieved from the appropriate HyperStore bucket, served to the user, and cached. To specify more than one device use a comma-separated list (and do not enclose with quotes). For recommended specs for this storage purpose (and for the other storage purposes covered by subsequent prompts), see the **"File Service Host Requirements and Recommendations"** (page 34) section.
	Devices for write cache	The storage device or devices to use for the pool's write cache (for example */dev/sdb*). The write cache is used for newly written data, not yet archived to a HyperStore bucket. To specify more than one device use a comma-separated list.
	Devices for logs (optional)	The storage device or devices to use for File Service logs. Using a dedicated device for this purpose is optional. If you do not specify a device here, the logs will be stored on the read cache device(s).
	Devices for database (optional)	The storage device or devices to use for File Service metadata (Cassandra and RabbitMQ databases). Using a dedicated device for this purpose is recommended but optional. To specify more

Category	Prompt for Information	Description
		than one device use a comma-separated list. If you do not specify a device here, the databases will be stored on the OS device. **Note** If you are installing on an **HFA-1100 Appliance** host or hosts, specifying at least two devices here is mandatory.

4. After you respond to the last prompt described in the table above, you are automatically returned to the Topology Editor menu. Choose ***d)*** *Show topology entries* to review your entries. Then, back at the Topology Editor menu choose ***f)*** *Save topology file* if the entries are correct, or ***e)*** *Edit topology* if you need to make corrections to any entries (and then do **f**) Save topology file).

   Once your topology is correct and saved, exit the system setup tool and proceed to **"Installing File Service Software On the Target Host(s)"** (page 39).

Note Based on your input at the prompts, the system setup tool generates a topology file named *<pool-name>.yaml* in your installation staging directory.

5.4. Installing File Service Software On the Target Host(s)

Before you can perform the File Service software installation you must:

- Create at least one storage policy in your HyperStore system. If you have not yet done so see Add a Storage Policy for instructions. The File Service installation will create a bucket or buckets for internal system use, and in order for buckets to be created there must be at least one storage policy in your HyperStore system. If you have multiple storage policies the default storage policy will be used.
- Obtain the HyperStore File Service version 8.2.2 product package (and corresponding signature file [*.sig* file]) from Cloudian. It's best to download these files directly to a working directory on the HyperStore Config Controller node. (If you download the files on to a Mac as an intermediary step before transferring it to the Config Controller node, some Macs may automatically convert the File Service *.tgz* package file into a *.tar* file, and the package needs to be a *.tgz* file for the installation to work.)

  IMPORTANT ! Do not install File Service software that has a different version number than your currently running HyperStore system. To function properly within your HyperStore system, the File Service software that you install must have the same version number as your HyperStore system software (not an older or newer version).

To install File Service software on the hosts that you identified in your File Service pool topology file:

If you are using the HyperStore Shell

The instructions below are for installing File Service software as the root user. If you want to install File Service software as a **HyperStore Shell (HSH)** user with the root user password disabled, contact Cloudian Support for assistance.

1. Put the File Service package into the HyperStore Config Controller node's installation staging directory (*/opt/cloudian-staging/8.2.2*).
2. While in the Config Controller node's installation staging directory run this *hsctl* command:

   ```
   # hsctl file install <package-name> -t <topology-file>
   ```

   For example:

   ```
   # hsctl file install CloudianFileService-8.2.2.tgz -t file-pool-1.yaml
   ```

Note that:

- All required software and supporting components are in the product package. The installation does not need or use internet access.
- If the installation fails because you entered incorrect information in your File Service pool topology file -- such as an incorrect IP address -- you can use the *system_setup.sh* tool to edit the topology file and then try the installation again. From the Topology Editor menu first choose ***c**) Load existing topology file*, then choose **e***) Edit topology* and edit the file entries as needed, and then ***f**) Save topology file*
- After the File Service installation is completed, any system administrators who were logged into the CMC during the File Service installation should log out of the CMC and log back in, in order to see File Service information and functionality.
- For each File Services pool installation, one or more S3 buckets are automatically created for internal system use (associated with an automatically created S3 user named *_system-file* within a user group also named *_system-file*):
  - For each 3-node File Services pool, one bucket is created in support of logs management for the pool.
  - For each single-node File Services pool, three buckets are created -- one supporting logs management, one for file system metadata backup, and one for Kubernetes resources backup.
  - The *_system-file* group and user, and these buckets for internal system use, will not appear in the CMC.
  - If you have a multi-region system the bucket(s) will be created in the default region.

For information about additional File Service setup tasks to complete after installing the service, see **Setting Up Service Features -> File Services (SMB/NFS) -> Preparing the File Services Feature** in either the *HyperStore Administrator's Guide* or the CMC's online Help.

Note At any time after installation, if you wish you can output the topology files currently used by your active HyperStore system: the main object storage system topology plus (if applicable) File Service topology and Search Service topology. To do so, in a working directory -- not the installation staging directory -- run the command *hsctl topology dump*. Note that if you were to run the command in the installation staging directory, and if your current system topology is different from your originally installed typology -- for example if you've added object nodes to the system -- the dump will overwrite the original version of your topology files with the current version.

5.5. Uninstalling a File Service Pool

If you want to uninstall a File Service pool from your HyperStore system you can do so by running this command on the HyperStore Config Controller node:

Note If you are decommissioning a node in a single-node File Service pool and plan to restore its state to a new replacement node, follow the procedure described in "Restoring a Replacement Node" in the "File Service Operations" section of the *Cloudian HyperStore Administrator's Guide*.

```
# hsctl file uninstall -n <file-pool-name> [--skip-backup-delete]
```

Note that:

- If you have installed **multiple File Service pools** and you want to uninstall all of them, run the above command separately for each pool -- letting the uninstall process complete for each pool (as indicated by output written to your terminal) before moving on to uninstalling the next pool.
- If you are uninstalling a **File Service pool that consists only of a single File Service node** and if you want to preserve the backups that have been created for that node you can do so by running the uninstall command like this:

  ```
  # hsctl file uninstall -n <file pool name> --skip-backup-delete
  ```

  The *--skip-backup-delete* option is applicable only for a single-node File Service pool.

- The uninstall operation removes the bucket(s) that the system had automatically created (see the **"Installing File Service Software On the Target Host(s)"** (page 39) section for bucket details), with the exception that in the case of a single-node File Service pool the backup buckets are preserved if you use the *--skip-backup-delete* option. If all system-created buckets for the File Service have been removed, the *_system-file* group and user are also removed.
- The *hsctl file uninstall* command will only work with a File Service pool that you have installed in HyperStore 8.1 or newer or that you have upgraded to HyperStore 8.1 or newer.For information about upgrading your HyperStore File Service see Chapter 3 "Upgrading Your HyperStore Software Version" (particularly the sub-section on Upgrading HyperStore File Service) in the ***Cloudian HyperStore Administrator's Guide***)

This page left intentionally blank

Chapter 6. Installing the HyperStore Search Service

Subjects covered in this section:

6.1. Introduction

After you've installed your HyperStore system and verified that it's operating properly -- or anytime after you've put your HyperStore system into operation -- you can optionally install the HyperStore Search Service. Once you've installed the HyperStore Search Service in your HyperStore cluster, you can enable Search for new storage policies as you create them, or for any of your existing storage policies. Then, owners of buckets that use those storage policies can use the CMC's **Search** page to search for objects within a bucket, based on object metadata. For more information about this feature, in the ***Cloudian HyperStore Administrator's Guide*** see *Setting Up Service Features -> Object Metadata and Search Feature Overview* and *Preparing the Metadata Search Feature*.

6.1.1. Search Service Pool

When you add the Search Service to your HyperStore system, you can deploy the service in either a single-node configuration or a three-node configuration. This deployment of either one or three Search nodes is known as a Search Service "pool". This follows the terminology convention used for the HyperStore File Service.

Note that a HyperStore Search Service pool **does not support**:

- Running on two hosts or on more than three hosts
- Migrating from a single-node deployment to a three-node deployment
- Running on HyperStore object storage nodes

6.2. Search Service Host Requirements and Recommendations

Cloudian does not offer a Search Service appliance. Your Search Service host machine(s) must be either a bare metal server or a virtual machine (VM). **For detailed resource requirements and recommendations based on your use case, consult with your Cloudian sales representative.** A general recommendation for a VM that will host the Search Service is a minimum of:

- 4 vCPUs
- 16GB RAM

- 2 virtual disks, one at 250GB (for OS and software) and one at 300GB (for search index storage)

Note If you are installing the Search Service on a host with less than 8 CPUs or less than 120GB RAM, the installer will display a warning but will allow the installation to proceed.

6.2.1. Operating System Requirements

To install and use the HyperStore Search Service you must have Rocky Linux as the OS on the host machine (s). To install Rocky Linux on your bare metal server or VM you must obtain the HyperStore Rocky Linux ISO file from Cloudian (this ISO is based on Rocky Linux 8.10). **Do not use the generic Rocky Linux ISO** from the Rocky Linux organization. You must use the HyperStore Rocky Linux ISO, in which the OS packages have been optimized for hosting Cloudian software.

Note When obtaining the HyperStore Rocky Linux ISO file from Cloudian, also obtain the HyperStore Search Service 8.2.2 product package (and corresponding signature file [*.sig* file]), which you will use to perform the Search Service software installation as described later in this procedure.

The process of installing the Rocky Linux 8.10 OS from the HyperStore Rocky Linux ISO file is the same as doing a standard Rocky Linux installation, but note that:

- Choose 'Server' as the base install type (not Minimal or Custom OS)
- When assigning IP addresses to network interfaces, manually set static IP addresses (do not use DHCP)
- The default root password is *HSApplianceLinux8*

For more information on the Rocky Linux installation procedure, refer to the Rocky Linux website.

6.2.2. Must Be Dedicated Host(s)

The HyperStore Search Service requires a dedicated host or hosts. You cannot run the Search Service on the same nodes as you're running HyperStore object storage software.

6.2.3. Networking Requirements

- The HyperStore Search Service node will be part of your HyperStore cluster. Therefore, along with being accessible on your front end network (see open port requirements further below), the Search Service node must be on the same backend network as the rest of your HyperStore nodes -- if those nodes use a dedicated backend network as is the case in most HyperStore deployments.
- On the backend network, communication among HyperStore nodes (including HyperStore Search Service nodes) must be allowed on all ports.
- On the frontend network, the following ports must be open:
  - 22
  - 80
  - 443
  - 9200
  - 9300

6.3. Creating a Search Service Pool Topology File

Like with deploying the **HyperStore File Service**, with the HyperStore Search Service there is the notion of a service "pool" as the basic deployment unit. However, while for the File Service you can deploy multiple service pools if you wish, in the current HyperStore release you can only deploy one Search Service pool -- which can consist of either one node or three nodes.

Follow the steps below to use the HyperStore *system_setup.sh* tool to create a topology file for your desired Search Service pool. You will then use this topology file as input to the Search Service installation as described in the next section, **"Installing Search Service Software On the Target Host"** (page 47).

Note If you prefer working directly with a *.yaml* file rather than answering prompts from the *system_setup.sh* tool, then instead of following the steps below you can edit the *sample-search-pool.yaml* file in the staging directory on your HyperStore Config Controller node, save the edited file as *<your-search-pool-name>.yaml*, then proceed to the instructions for **"Installing Search Service Software On the Target Host"** (page 47).

1. On your HyperStore Config Controller node, change into the installation staging directory (*/opt/cloudian-staging/8.2.2*) and then launch the *system_setup.sh* tool:

   ```
   # ./system_setup.sh
   ```

   If you are using the HyperStore Shell

   If you are using the **HyperStore Shell (HSH)** you can launch the system setup tool with this command:

   ```
   $ hspkg setup
   ```

   To run this command using the HSH you must be an HSH **Trusted user**.

2. From the tool's main menu choose ***4)*** *Setup Cluster Topology*. Then from the Topology Editor menu choose ***c)*** *Generate a new topology file for Search Service nodes*.
3. Provide all the information for which the tool prompts you:

Category	Prompt for Information	Description
Pool name and location	Search Service pool name	Name that you want to give to this Search Service pool. The name must be at least 2 and no more than 32 characters long, and only lower case alphanumeric characters and dashes are supported (no other special characters and no spaces). The name must start with a letter, and must end with a letter or number. This name will be used in the topology file name (*<pool-name>.yaml*), and in the CMC interface as a way of identifying this Search Service pool.
	DNS name	Fully qualified domain name to use for this Search Service pool (for example, *search.my-enterprise.com*).
	Region name	HyperStore service region in which this Search Service pool will be installed. This must be one of the service region names from your existing HyperStore system.

Category	Prompt for Information	Description
	Data center name	HyperStore data center in which this Search Service pool will be installed. This must be one of the data center names from your existing HyperStore system. After entering the data center name you will be prompted for the name of the service region in which the data center is located -- this must be the same region as you entered at the earlier prompt.
	Rack name	Name of the rack in which this Search Service pool will be installed.
Node information	Node name	Hostname of a node that will belong to this Search Service pool. This must be the 'short' hostname, not an FQDN. After entering the node name you will be prompted for the name of the data center in which the node is located -- this must be the same data center as you entered at the earlier prompt.
	Node IP address	IP address of the node. This must be the physical IP address assigned to the host's front end port.
	Internal interface	The name of the internal interface that your nodes will use for internal cluster communications (communications between HyperStore nodes). For example, *eth1*.
	Add another node?	This prompt gives you the option to loop back through the node information prompts again, to add more nodes to the Search Service pool. A Search Service pool can have either one node or three nodes. Other node quantities such as two nodes or four nodes are not supported. If you are creating a three-node Search Service pool, all nodes must be in the same data center. If you reply "no" after just one node (i.e. you're installing just a single-node Search Service pool), you'll be returned to the Topology Editor menu and you can proceed directly to Step 4 below. If you reply "no" after a third node, you will be prompted for additional information that's applicable only to a three-node pool.
For three-node pool only:		
Disk information	Disk name	The device path (for example */dev/sdb*) for the disk on which to store search index data on each node. This device path must be the correct path for all three of the hosts that you've included in the topology (each of the three nodes must have a disk at this same device path). After entering one disk name the next prompt will give you the option to specify a second disk if you wish (if you want the

Category	Prompt for Information	Description
		search index data on each node to be split among multiple disks). The total space on your specified disk(s) should be at least 300GB per node and potentially more depending on the guidance that you've received from your Cloudian sales representative. In your three-node Search Service pool, the search index data will be replicated on each of the three nodes for redundancy.
Kubernetes cluster information	PV size	Persistent volume size in GBs. Leave at the default of 10 unless otherwise instructed by your Cloudian representative.
	Kubernetes API address	An IP address available for the Search Service system to use as the Kubernetes intranet address (which facilitates Kubernetes communications within the Search Service pool). This IP address must be from the same subnet as the physical IP addresses of the nodes in this Search Service pool. By default the topology generator will propose the next IP address in sequence above that of the last node IP address that you entered (for the three nodes that will comprise this pool).
	Load balancer address	Virtual IP address that the system will use as the Search Service API address for this three-node Search Service pool. Search Service API requests submitted to this address will be load balanced among the three nodes. This IP address must be from the same subnet as the physical IP addresses of the nodes in this Search Service pool. By default the topology generator will propose the next IP address in sequence above that of the Kubernetes API address.

4. After you respond to the last prompt described in the table above, you are automatically returned to the Topology Editor menu. Choose ***d)*** *Show topology entries* to review your entries. Then, back at the Topology Editor menu choose ***f)*** *Save topology file* if the entries are correct, or ***e)*** *Edit topology* if you need to make corrections to any entries. Once your topology is correct and saved, return to the system setup tool's main menu and then exit the tool.

Note Based on your input at the prompts, the system setup tool generates a topology file named *<pool-name>.yaml* in your installation staging directory.

6.4. Installing Search Service Software On the Target Host

Before you can perform the Search Service software installation you must obtain the HyperStore Search Service version 8.2.2 product package (and corresponding signature file [*.sig* file]) from Cloudian.

IMPORTANT ! Do not install Search Service software that has a different version number than your currently running HyperStore system. To function properly within your HyperStore system, the Search Service software that you install must have the same version number as your HyperStore system software (not an older or newer version).

To install Search Service software on the host(s) that you identified in your Search Service pool topology file:

If you are using the HyperStore Shell

The instructions below are for installing Search Service software as the root user. If you want to install Search Service software as a **HyperStore Shell (HSH)** user with the root user password disabled, contact Cloudian Support for assistance.

1. Put the Search Service product package into the HyperStore Config Controller node's installation staging directory (*/opt/cloudian-staging/8.2.2*).
2. While in the Config Controller node's installation staging directory run this *hsctl* command:

   ```
   # hsctl search install <package-name> -t <topology-file>
   ```

   For example:

   ```
   # hsctl search install CloudianSearchService-8.2.2.bin -t my-search-pool.yaml
   ```

Note that:

- All required software and supporting components are in the product package. The installation does not need or use internet access.
- If the installation fails because you entered incorrect information in your Search Service pool topology file -- such as an incorrect IP address -- you can use the *system_setup.sh* tool to edit the topology file and then try the installation again. From the Topology Editor menu first choose ***c**) Load existing topology file*, then choose ***e**) Edit topology* and edit the file entries as needed, and then ***f**) Save topology file*.

For information about additional Search Service setup tasks to complete after installing the service, see *Setting Up Service Features -> Object Metadata and Search -> Preparing the Metadata Search Feature* in either the ***Cloudian HyperStore Administrator's Guide*** or the CMC's online Help.

Note At any time after installation, if you wish you can output the topology files currently used by your active HyperStore system: the main object storage system topology plus (if applicable) File Service topology and Search Service topology. To do so, in a working directory -- not the installation staging directory -- run the command *hsctl topology dump*. Note that if you were to run the command in the installation staging directory, and if your current system topology is different from your originally installed typology -- for example if you've added object nodes to the system -- the dump will overwrite the original version of your topology files with the current version.

6.5. Uninstalling a Search Service Pool

If you want to uninstall a Search Service pool from your HyperStore system you can do so by running this command on the HyperStore Config Controller node:

```
# hsctl search uninstall -n <search-pool-name>
```

Chapter 7. HyperStore Installation Reference

This section of the installation documentation provides reference information that you may find useful in some installation scenarios and circumstances.

7.1. Installation Error Logging

7.1.1. Installation Logs

When you run the HyperStore installer it generates the following logs that may be helpful for troubleshooting installation problems:

On the Config Controller node (the HyperStore node on which you're running the install script):

- *<installation-staging-directory>/cloudian-installation.log*
- */var/log/puppetserver/puppetserver.log*

On every HyperStore node:

- */tmp/puppet_agent.log*

Scanning these logs for error or warning messages should help you identify the stage at which the installation encountered a problem, and the nature of the problem. This information can further your own troubleshooting efforts, and also can help Cloudian Support pinpoint the problem in the event that you need assistance from Support.

Note When you use *system_setup.sh* to prepare your nodes for HyperStore installation, that tool writes its logging output to *system_setup.sh.log*, in the same directory as the *system_setup.sh* tool is located (typically your installation staging directory). This log can also be a source of troubleshooting information.

7.1.2. Debug Mode

Another potentially useful source of troubleshooting information is to run the installer in debug mode. In the installation staging directory:

```
# ./cloudianInstall.sh -d
```

For example, if you encounter an error while running the installer in regular (non-debug) mode, you can exit the installer menu and then launch the installer again in debug mode. You can then either re-execute the installation starting from the beginning, or re-execute the installation starting from the step that had previously failed. If you had partially run the installation, then when you subsequently select Install Cloudian HyperStore at the main menu a sub-menu will display to let you choose from among several installation tasks to run again.

When run in debug mode, the installer will write highly granular messages to both the console and the installation log (*cloudian-installation.log*).

7.2. HyperStore Listening Ports

The HyperStore system uses the listening ports specified in the table below. Only the service ports for the CMC, S3, IAM, SQS, and Admin services -- the port numbers in italics in the "Listening Port" column -- should be open to traffic originating from outside the HyperStore system. All other ports must be closed to traffic from outside the system, for system security.

Each HyperStore node includes a built-in HyperStore Firewall that implements port restrictions appropriate to a HyperStore cluster. The HyperStore Firewall is disabled by default in HyperStore systems that were originally installed as a version older than 7.2; and enabled by default in HyperStore systems that originally installed as version 7.2 or newer. You can enable/disable the firewall on all HyperStore nodes by using the installer's Advanced Configuration Options. For instructions see "HyperStore Firewall" in the ***Cloudian HyperStore Administrator's Guide***.

> **Note** If you are installing HyperStore across **multiple data centers and/or multiple service regions**, the HyperStore nodes in each data center and region will need to be able to communicate with the HyperStore nodes in the other data centers and regions. This includes services that listen on the internal interface (such as Cassandra, the HyperStore Service, and Redis). Therefore you will need to configure your networking so that the internal networks in each data center and region are connected to each other (for example, by using a VPN).

Service	Listening Port	Interface(s) Bound To	Purpose
Cloudian Management Console (CMC)	*8888*	All	Requests from administrators' or end users' browsers over HTTP
	8443	All	Requests from administrators' or end users' browsers over HTTPS
S3 Service	*80*	All	Requests from the CMC or other S3 client applications over HTTP
	443	All	Requests from the CMC or other S3 client applications over HTTPS
	81	All	Requests relayed by an HAProxy load balancer using the PROXY Protocol
	4431	All	Requests relayed by an HAProxy load balancer using the PROXY Protocol with SSL (if enabled by configuration)
	19080	Internal	JMX access
IAM Service and STS Service	*16080*	All	Requests from the CMC or other Identity and Access Management (IAM) or Security Token Service (STS) clients over HTTP **Note** In the current HyperStore release, the STS Service uses the same listening ports as

Service	Listening Port	Interface(s) Bound To	Purpose
			the IAM Service.
	16443	All	Requests from the CMC or other IAM or STS clients over HTTPS
	19084	Internal	JMX access
SQS Service	*18090*	All	Requests from Simple Queue Service (SQS) clients over HTTP
	18443	All	Requests from SQS clients over HTTPS
	19085	Internal	JMX access
Admin Service	*18081*	All	Requests from the CMC or other Admin API clients over HTTP
	19443	All	Requests from the CMC or other Admin API clients over HTTPS (Note: The CMC by default uses HTTPS to access the Admin Service) **IMPORTANT !** The Admin Service is intended to be accessed only by the CMC and by system administrators using other types of clients (such as cURL). Do not expose the Admin Service to a public network.
	19081	Internal	JMX access
Redis Monitor	9078	Internal	Communication between primary and backup Redis Monitor instances **Note** Not applicable in a HyperStore Single-Node system.
	19083	Internal	JMX access
HyperStore Service	19090	Internal	Data operation requests from the S3 Service

Service	Listening Port	Interface(s) Bound To	Purpose
	19050	Internal	Communication between HyperStore Service instances **Note** Not applicable in a HyperStore Single-Node system.
	19082	Internal	JMX access
Credentials DB and QoS DB (Redis)	6379	Internal	Requests to the Credentials DB from the S3 Service, HyperStore Service, or Admin Service; and communication between Credentials DB nodes
	6380	Internal	Requests to the QoS DB from the S3 Service, HyperStore Service, or Admin Service; and communication between QoS DB nodes
Metadata DB (Cassandra)	9042	Internal	Data operations requests from the S3 Service, HyperStore Service, or Admin Service, using CQL protocol
	9160	Internal	Data operations requests from the S3 Service, HyperStore Service, or Admin Service, using Thrift protocol
	7000	Internal	Communication between Cassandra instances **Note** Not applicable in a HyperStore Single-Node system.
	7199	Internal	JMX access
Cloudian Monitoring Agent	19070	Internal	Requests from the Cloudian Monitoring Data Collector
Config Controller	4505	Internal	On your Config Controller node this is the port to which Salt agents ("minions") establish a persistent connection so that the Salt Master can publish to the minions.
	4506	Internal	Salt minions connect to this port on the Config Controller as needed to send results to the

Service	Listening Port	Interface(s) Bound To	Purpose
			Salt Master, and to request files and minion-specific data values.
SSH	22	All	The HyperStore installer accesses this SSH port on each node on which you are installing HyperStore software (during initial cluster install or if you subsequently expand your cluster)
NTP	123	All	NTP port for time synchronization between nodes **Note** Not applicable in a HyperStore Single-Node system.
Echo	7	Internal	The Cloudian Monitoring Data Collector uses Echo (port 7) to check whether each node is reachable.
HyperIQ	9999	All	If you use Cloudian HyperIQ to monitor your HyperStore system, HyperIQ accesses port 9999 on each HyperStore node

7.3. Outbound Internet Access

The HyperStore installation process does not require outbound internet access. However, the following HyperStore features do access the internet once the system is in operation; and HyperStore does need access to NTP server(s) during the installation (see the "Pre-Configured chronyd" bullet point below). If you use forward proxying in your environment, after HyperStore installation you may want to set up forward proxying to support these HyperStore features:

- **Smart Support** — The Smart Support feature (also known as "Phone Home") securely transmits HyperStore daily diagnostic information to Cloudian Support over the internet. HyperStore supports configuring this feature to use an explicit forward proxy for its outbound internet access (after installation, the relevant settings are *common.phoneHome.proxy.host*, *common.phoneHome.proxy.port*, *common.phoneHome.proxy.auth.username*, and *common.phoneHome.proxy.auth.password*). To use a forward proxy with this feature you should configure your forward proxy to support access to **.s3-support.cloudian.com* (that is, to any sub-domain of *s3-support.cloudian.com*).
- **Auto-Tiering and Cross-Region Replication** — If you want to use either the auto-tiering feature or the cross-region replication feature (CRR), the S3 Service running on each of your HyperStore nodes requires outbound internet access. These features do not support configuring an explicit forward proxy, but you can use transparent forward proxying if you wish. (Setting up transparent forward proxying is outside the scope of this documentation.) For more information on these features see the "Auto-Tering

Feature Overview" and "Cross-Region Replication Overview" sections in the ***Cloudian HyperStore Administrator's Guide***.

- **Pre-Configured *chronyd*** — Accurate, synchronized time across the cluster is vital to HyperStore service. When you install your HyperStore cluster, the installation script automatically configures a robust NTP set-up using *chronyd*. In each of your HyperStore data centers four of your HyperStore nodes are automatically configured to act as internal NTP servers. (If a HyperStore data center has only four or fewer nodes, then all the nodes in the data center are configured as internal NTP servers.) These internal NTP servers are configured to connect to external NTP servers — by default the public servers from the *pool.ntp.org* project. In order to connect to the external NTP servers, the internal NTP servers must be allowed outbound internet access. This feature does not support configuring an explicit forward proxy, but you can use transparent forward proxying if you wish. (Setting up transparent forward proxying is outside the scope of this documentation.)

  IMPORTANT ! If you do not allow HyperStore hosts to have outbound connectivity to the internet, then during the interactive installation process -- when you are prompted to specify the NTP servers that HyperStore hosts should connect to -- you must specify NTP servers within your environment, rather than the public NTP servers that HyperStore connects to by default. If HyperStore hosts cannot connect to any NTP servers, the installation will fail.

  After HyperStore installation, to see which of your HyperStore nodes are internal NTP servers, log into the CMC and go to **Cluster** → **Cluster Config** → **Cluster Information**. On that CMC page you can also see your configured list of NTP servers.

  For more information on HyperStore's NTP set-up, see the "NTP Automatic Set-Up" section in the ***Cloudian HyperStore Administrator's Guide***.

7.3.1. Multi-DC Considerations

If you are installing HyperStore across multiple data centers and/or multiple service regions, the HyperStore nodes in each data center and region will need to be able to communicate with the HyperStore nodes in the other data centers and regions. This includes services that listen on the internal interface (such as Cassandra, the HyperStore Service, and Redis). Therefore you will need to configure your networking so that the internal networks in each data center and region are connected to each other (for example, by using a VPN). See **"HyperStore Listening Ports"** (page 50) for HyperStore requirements regarding listening port access.

7.4. File System Requirements

Subjects covered in this section:

Cloudian recommends that you use the HyperStore *system_setup.sh* tool to configure the disks and mount points on your HyperStore nodes, as described in **"Configuring Network Interfaces, Time Zone, and Data Disks"** (page 22). The tool is part of the HyperStore product package (when you extract the *.bin* file).

If you do not use the system setup tool for disk setup, use the information below to make sure that your hosts meet HyperStore file system requirements.

7.4.1. OS/Metadata Drives and Data Drives

Although it's possible to install HyperStore on a host with just a single hard drive, for a rigorous evaluation or for production environments each host should have multiple drives (see **"Host Hardware and Operating System"** (page 15)). On host machines with multiple hard drives:

- HyperStore will by default use the drive that the OS is on for storing system metadata (in the Metadata DB, the Credentials DB, and the QoS DB). Cloudian recommends that you dedicate two drives to the OS and system metadata in a RAID-1 mirroring configuration. Preferably the OS/metadata drives should be SSDs. Also preferably, the metadata should be on dedicated drives as described in **"Option for Putting the Metadata DB on Dedicated Drives Rather Than the OS Drives"** (page 57)
- You must format all other available hard drives with *ext4* file systems mounted on raw disks. These drives will be used for storing S3 object data. RAID is not necessary on the S3 object data drives.

For example, on a machine with 2 SSDs and 12 HDDs:

- Mirror the OS on the two SSDs. For more detailed recommendations for partitioning these disks see **"Partitioning of Disks Used for the OS and Metadata Storage"** (page 16).
- Format each of the 12 HDDs with *ext4* file systems and configure mount points such as */cloudian1*, */cloudian2*, */cloudian3* and so on.

> **Note** On the HDDs for storing object data, HyperStore **does not support** XFS file systems; VirtIO disks; or Logical Volume Manager (LVM). For questions regarding these unsupported technologies, contact Cloudian Support.

7.4.2. Mount Point Naming Guidelines

If you are installing HyperStore on multiple hosts that each have multiple disks for object data storage, use the same mount point naming scheme on each of your hosts. If all your hosts have the same number of disks, then they should all have the identical set of mount points for HyperStore object storage. For example, if each host has 12 disks for object storage, then on all your hosts you could name the mount points */cloudian1*, */cloudian2*, */cloudian3*, and so on up through */cloudian12*.

If in your installation cluster some hosts have more disks than others, use as much overlap in mount point naming as possible. For example, suppose that most of your hosts have 10 disks for storing object data while one host has 12 disks. In this scenario, all of the hosts can have mount points */cloudian1*, */cloudian2*, */cloudian3*, and so on up through */cloudian10*, while the one larger host has those same mount points plus also */cloudian11* and */cloudian12*.

> **Note** Although uniformity of mount point naming across nodes (to the extent possible) is desirable for simplicity's sake, the HyperStore installation does support a way to accommodate differences in the number or names mount points across nodes -- this is described in **"A Data Directory Mount Point List (fslist.txt) Is Required"** (page 58)..

7.4.3. Option for Putting the Metadata DB on Dedicated Drives Rather Than the OS Drives

Regarding the Metadata DB (built on Cassandra), the recommended configuration is to put your Cassandra data on dedicated drives, rather than on the OS drives. In this case you would have:

- OS drives in RAID-1 configuration. The Credentials DB and QoS DB will also be written to these drives.
- Cassandra drives in RAID-1 configuration. On these drives will be written Cassandra data and also the Cassandra commit log. (And also the Digest DB commit log, in HyperStore 8.2 and later.)

  Note You must create a Cassandra data directory named as *<mountpoint>/cassandra* (for example *cassandradb/cassandra*) and a Cassandra commit log directory named as *<mountpoint>/cassandra_commit* (for example *cassandradb/cassandra_commit*).

- Multiple drives for S3 object data (with mount points for example */cloudian1*, */cloudian2*, */cloudian3* and so on), with no need for RAID protection.

For more information see **"A Data Directory Mount Point List (fslist.txt) Is Required"** (page 58).

Note If you are using **Quad-Level Cell (QLC) SSDs**, these should be used as the object data drives -- **not** as the metadata drives. The metadata drives will be subject to very frequent writes of small files. By contrast, QLC SSDs are best suited for read-heavy workloads and large objects.

7.4.4. You Must Use UUIDs in fstab

In your *fstab* file, **you must use UUIDs** to identify the devices to which you will mount HyperStore S3 object data directories. Do not use device names or LABELs.

If you are not using UUIDs in *fstab* currently, follow the instructions below to modify your *fstab* so that it uses UUIDs for the devices to which you will mount S3 object data directories (you do not need to do this for the OS/-metadata mount points).

As *root*, do the following:

1. Check whether your *fstab* is currently using UUIDs for your S3 object data drives. In the example below, there are two S3 object data drives and they are currently identified by device name, not by UUID.

   ```
   # cat /etc/fstab
   ...
   ...
   /dev/sdb1 /cloudian1  ext4  rw,noatime,barrier=0,data=ordered,errors=remount-ro 0 1
   /dev/sdc1 /cloudian2  ext4  rw,noatime,barrier=0,data=ordered,errors=remount-ro 0 1
   ```

2. Back up your existing fstab file:

   ```
   # cp /etc/fstab /etc/fstab.backup.<today's date>
   ```

3. Retrieve the UUIDs for your devices by using the *blkid* command.

   ```
   # blkid
   ...
   ...
   /dev/sdb1: UUID="a6fed29c-97a0-4636-afa9-9ba23e1319b4" TYPE="ext4"
   /dev/sdc1: UUID="rP38Ux-3wzO-sP3Y-2CoD-2TDU-fjpO-ffPFZV" TYPE="ext4"
   ```

4. Open *fstab* in an editor.
5. For each device that you are using for S3 object storage, replace the device name with *UUID="<UUID>"*, copying the device's UUID from the *blkid* response in the previous step. For example:

```
# Original line

/dev/sdb1 /cloudian1 ext4 rw,noatime,barrier=0,data=ordered,errors=remount-ro  0 1

# Revised line

UUID="a6fed29c-97a0-4636-afa9-9ba23e1319b4" /cloudian1  ext4  rw,noatime,barrier=0,
data=ordered,errors=remount-ro   0 1
```

6. After editing *fstab* so that each device on which you will store S3 data is identified by a UUID, save your changes and close the *fstab* file.
7. Remount the host's file systems:

```
# mount -a
```

Repeat this process for **each host on which you will install HyperStore**.

7.4.5. A Data Directory Mount Point List (fslist.txt) Is Required

If you do not use the HyperStore *system_setup.sh* script to configure the data disks and mount points on your nodes, you must manually create a data directory mount point list file and place it in your installation staging directory on the Config Controller node, as described below.

> **Note** If you use the *system_setup.sh* script to configure the disks and mount points on your nodes, the script creates the needed mount point list files automatically and you can ignore the instructions below.

If all your nodes have the same data mount points -- for example if all nodes have as their data mount points */cloudian1*, */cloudian2*, and so on through */cloudian12* -- you only need to create one mount point list file. If some nodes have a different set of mount points than do other nodes -- for example if some nodes have more data disks than other nodes -- you will need to create a default mount point list file and also a node-specific mount point list file for each node that differs from the default.

In your installation staging directory create a file named *fslist.txt* and in the file enter one line for each of your S3 data directory mount points, with each line using the format below.

```
<deviceName> <mountPoint>
```

Example of a properly formatted file (truncated):

```
/dev/sdc1 /cloudian1
/dev/sdd1 /cloudian2
...
```

> **Note** Use device names in your *fslist.txt* file, not UUIDs.

Optionally, you can also include an entry for the Cassandra data directory and an entry for the Cassandra commit log directory, if you do not want this data to be put on the same device as the operating system (see **"Option for Putting the Metadata DB on Dedicated Drives Rather Than the OS Drives"** (page 57)). If you do

not specify these Cassandra directory paths in *fslist.txt*, then by default the system automatically puts Cassandra data and commit log directories on the same device on which the operating system resides.

Do not use symbolic links when specifying your mount points. The HyperStore system does not support symbolic links for data directories.

If some of your hosts have data directory mount point lists that differ from the cluster default, in the installation staging directory create a *<hostname>_fslist.txt* file for each such host. For example, along with the default *fslist.txt* file that specifies the mount points that most of your hosts use, you could also have a *cloudian-node11_fslist.txt* file and a *cloudian-node12_fslist.txt* file that specify mount points for two non-standard nodes that have hostnames *cloudian-node11* and *cloudian-node12*.

7.4.6. Reducing Reserved Space to 0% for HyperStore Data Disks

By default Linux systems reserve 5% of file system space for root user and system services. On modern large-capacity disks this can be a waste of a considerable amount of storage space. Cloudian recommends that you set the reserved space to 0% for each drive on which you will store HyperStore object data (S3 object data).

For each HyperStore data drive do the following.

```
### Check current "Reserved block count":

# tune2fs -l <device>

### Set Reserved block count to 0%:

# tune2fs -m 0 <device>

### For example:

# tune2fs -m 0 /dev/sdc1
```

7.5. cloudianInstall.sh

The *cloudianInstall.sh* tool (also known as "the installer") serves several purposes including:

- Installation of a HyperStore cluster (for detail see **"Installing a New HyperStore System"** (page 27))

  > **Note** If you have a HyperStore Single-Node system, your system is installed for you by Cloudian Professional Services.

- Implementing advanced, semi-automated system configuration changes (for detail see **"Installer Advanced Configuration Options"** (page 62))
- Pushing configuration file edits to the cluster and restarting services to apply the changes(for detail see "Pushing Configuration File Edits to the Cluster and Restarting Services" in the ***Cloudian HyperStore Administrator's Guide***)

The *cloudianInstall.sh* tool is in your installation staging directory on your Config Controller node. To perform advanced configurations, or to push configuration file changes to the system and restart services, you would launch the tool simply like this, without using additional command line options:

```
# ./cloudianInstall.sh
```

7.5.1. Command Line Options When Using cloudianInstall.sh for HyperStore Cluster Installation

To perform a HyperStore cluster installation you typically would launch the script either like this:

```
# ./cloudianInstall.sh
```

Or like this if you are not using your DNS environment to resolve HyperStore service endpoints and you want to use the bundled tool *hsDNS* instead (which is not appropriate for production systems):

```
# ./cloudianInstall.sh configure-hsdns
```

However the script does support additional command line options. The syntax is as follows:

```
# ./cloudianInstall.sh [-k <ssh-private-key-filename>] [-d] [-h]
[configure-hsdns] [configure-qlc-rdb] [no-hosts] [no-firewall] [force] [uninstall]
```

> **Note** If you use multiple options, on the command line place options that start with a "-" (such as *-k <ssh-private-key-filename>* or *-d*) before options that do not (such as *no-hosts* or *configure-hsdns*).

*If you are using the **HyperStore Shell***

If you are using the HyperStore Shell (HSH) as a Trusted user, from any directory on the Config Controller node you can launch the installer with this command:

```
$ hspkg install
```

The installer's options are the same regardless of whether it is launched from the HSH command line or the OS command line.

> **Note** After using the installer, exit the installer when you're done. Do not leave it running. Certain automated system tasks invoke the installer and cannot do so if it is already running.

The supported command line options are:

- *-k <ssh-private-key-filename>* — The Config Controller node uses SSH for secure communication with the rest of your HyperStore installation nodes. By default the install script automatically creates an SSH key pair for this purpose. But if instead you would prefer to use your own existing SSH key pair for this purpose, you can use the installer's *-k <ssh-private-key-filename>* option to specify the name of the private key file (including the full path to the file). When you run the install script it will copy the private key and corresponding public key to the installation staging directory, and in the staging directory the key file will be renamed to *cloudian-installation-key*. Then from the staging directory, the public key file *cloudian-installation-key.pub* will be copied to each node on which you are installing HyperStore.

  > **Note** Keys created with default *ssh-keygen* behavior will not work with *JSch* (a Java implementation of SSH2 that HyperStore uses). Instead use these options for key generation:
  >
  > *ssh-keygen -t ecdsa -m pem -b 256*

- *-d* — Turn on debugging output.
- *-h* — Display usage information for the install tool. This option causes the tool to print a usage message and exit.

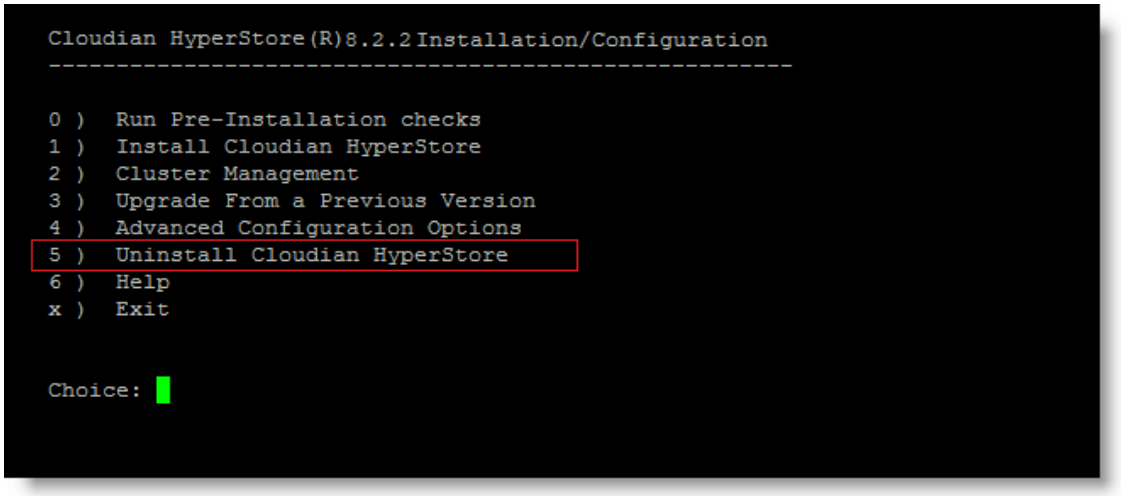
> **Note** This usage information mentions more command line options than are described here in this Help topic. This is because the usage information includes installer options that are meant for HyperStore internal system use, such as options that are invoked by the CMC when you use the CMC to add nodes to your cluster or remove nodes from your cluster. You should perform such operations through the CMC, not directly through the installer. The CMC implements automations and sanity checks beyond what is provided by the install script alone.

- *configure-hsdns* — Include this command-line option if you want to use HyperStore's hsDNS component to resolve HyperStore service endpoints within the cluster (and also externally if you set up DNS delegation). For more information about hsDNS, in the "Service Features" section of your *Cloudian HyperStore Administrator's Guide*, see "hsDNS".
- *configure-qlc-rdb* — Use this command-line option If you are using Quad-Level Cell (QLC) SSDs as your object data drives. Using this option results in the commit logs from RocksDB -- used for object digests -- being stored on the Cassandra (Metadata DB) drives rather than on the data drives (where they are stored in traditional non-QLC deployments). This helps to reduce frequent-write, small-file activity on your QLC drives.
- *no-hosts* — Use this option if you do not want the install tool to append entries for each HyperStore host on to the */etc/hosts* file of each of the other HyperStore hosts. By default the tool appends to these files so that each host is resolvable to the other hosts by way of the */etc/hosts* files.
- *no-firewall* — If this option is used, the HyperStore firewall will **not** be enabled upon HyperStore installation. By default the HyperStore firewall will be enabled upon completion of a fresh HyperStore installation. For more information about the HyperStore firewall see the "HyperStore Firewall" section in the ***Cloudian HyperStore Administrator's Guide***.
- *force* — By default the installer performs certain prerequisite checks on each node on which you are installing HyperStore and aborts the installation if any of your nodes fails a check. By contrast, if you use the *force* option when you launch the installer, the installer will output warning messages to the terminal if one or more nodes fails a prerequisite check but the installation will continue rather than aborting. The prerequisite checks that this feature applies to are:
  - CPU has minimum of 8 cores
  - RAM is at least 128GB
  - System Architecture is x86 64-bit
  - SELinux is disabled
  - firewalld is disabled
  - iptables is not running

> **Note** If you specify the *force* option when running the installer, the *force*option will "stick" and will be used automatically for any subsequent times the installer is run to install additional nodes (such as when you do an "Add Node" operation via the Cloudian Management Console, which invokes the installer in the background). To turn the *force*option off so that it is no longer automatically used when the installer is run to add more nodes, launch the installer and go to the Advanced Configuration Options. Then choose option **t** for **Configure force behavior** and follow the prompts.

> **Note** Even if the *force* option is used the installer will abort if it detects an error condition on the host that will prevent successful installation.

- *uninstall* — If you use this option when launching the installer, the installer main menu will include an additional menu item -- "Uninstall Cloudian HyperStore".



Use this menu option only if you want to **delete the entire HyperStore system, on all nodes, including any metadata and object data** stored in the system. You may want to use this Uninstall Cloudian HyperStore option, for example, after completing a test of HyperStore -- if you do not want to retain the test system.

> **IMPORTANT !** Do not use this option to uninstall a single node from a HyperStore system that you want to retain (such as a live production system). For instructions on removing a node from a HyperStore system see the "Removing a Node" section in the ***Cloudian HyperStore Administrator's Guide***.

7.6. Installer Advanced Configuration Options

The HyperStore installation tool supports several types of advanced system configurations which can be implemented at any time after initial installation of the system. To access the advanced configuration options, on the Config Controller node change into your installation staging directory (*/opt/cloudian-staging/8.2.2*) and launch the installer.

```
# ./cloudianInstall.sh
```

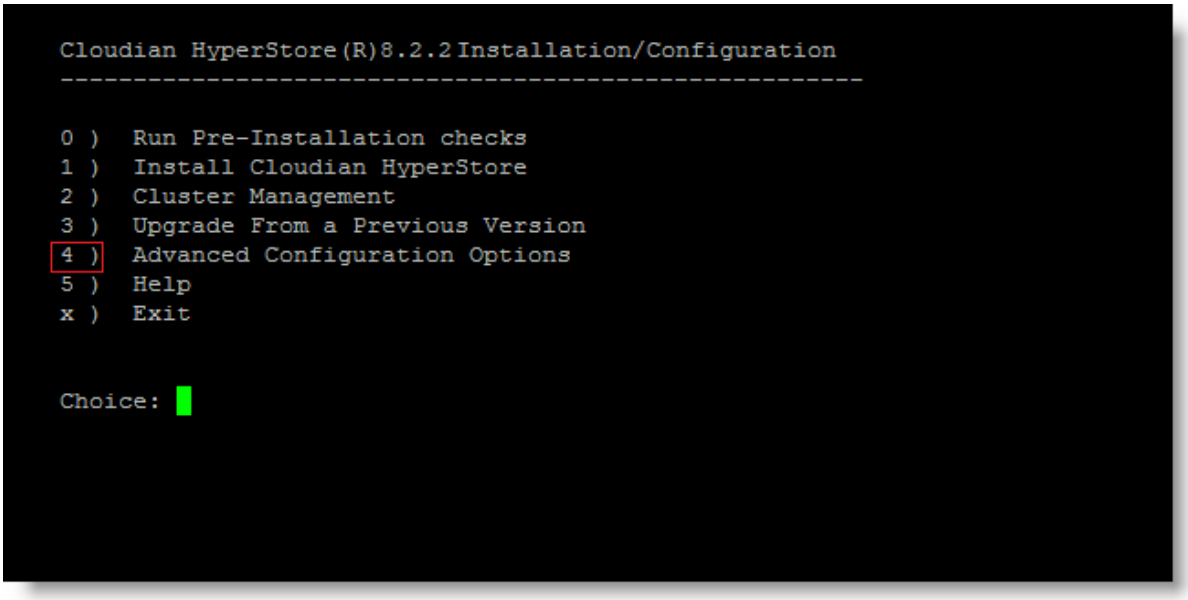
*If you are using the **HyperStore Shell***

If you are using the HyperStore Shell (HSH) as a Trusted user, from any directory on the Config Controller node you can launch the installer with this command:

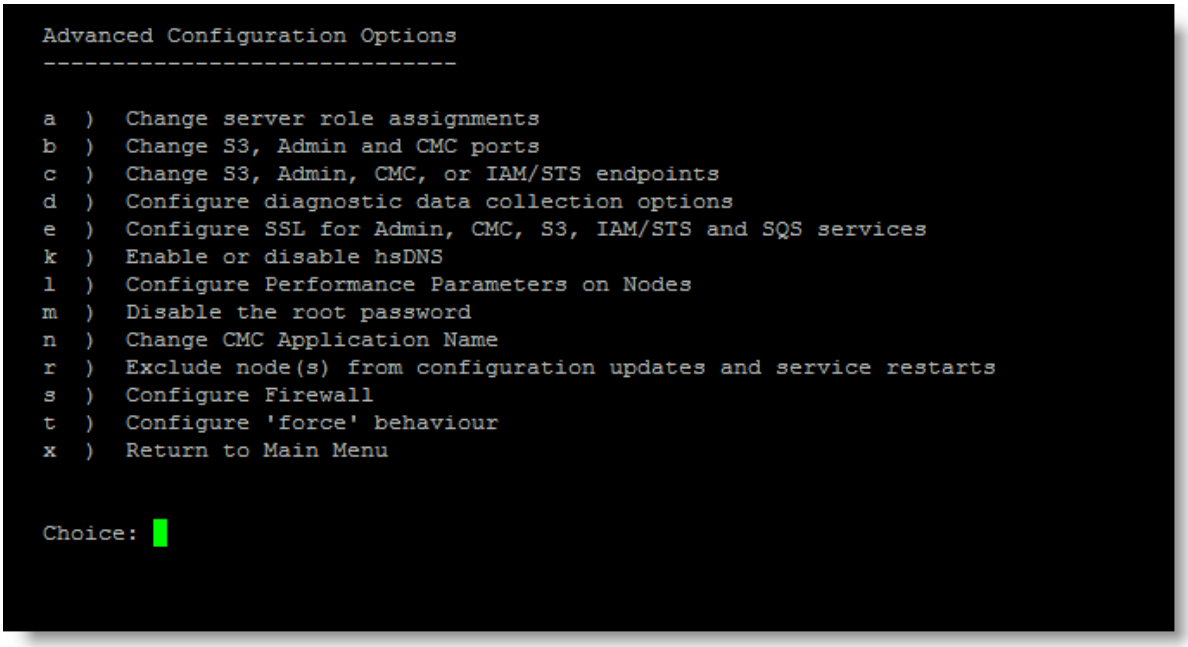
```
$ hspkg install
```

Once launched, the installer's menu options (such as referenced in the steps below) are the same regardless of whether it was launched from the HSH command line or the OS command line.

At the installer main menu's Choice prompt enter **4** for Advanced Configuration Options.



This opens the "Advanced Configuration Options" sub-menu.



From this menu you can choose the type of configuration change that you want to make and then proceed through the interactive prompts to specify your desired settings.

For information about each of these options, see the "Configuration Settings -> Installer Advanced Configuration Options" section of the ***Cloudian HyperStore Administrator's Guide***.

Note As a best practice, you should complete basic HyperStore installation first and confirm that the system is working properly (by running the installer's Validation Tests, under the "Cluster Management" menu) before you consider using the installer's advanced configuration options.