Cloudian HyperStore Installation Guide

Version 7.1

This page left intentionally blank

This page left intentionally blank

Contents

This page left intentionally blank

Chapter 1. HyperStore Installation Introduction

This installation documentation is for Cloudian HyperStore^{R} version 7.1. The main part of this documentation describes how to do a **fresh installation** of Cloudian HyperStore 7.1.

If you are **upgrading** to 7.1 from HyperStore 6.2 or later, you can go directly to **"Upgrading an Existing HyperStore System"** (page 25). (For upgrading to 7.1 from a version older than 6.2, contact Cloudian Support.)

If you do not yet have the HyperStore 7.1 package, you can obtain it from the Cloudian FTP site *ftp.cloudian.com*. You will need a login ID and password (available from Cloudian Support). Once logged into the FTP site, change into the *Cloudian_HyperStore* directory and then into the *cloudian-7.1* sub-directory. From there you can download the HyperStore software package, which is named *CloudianHyperStore-7.1.bin*.

Note The HyperStore ISO file (with file name extension *.iso*) is intended for setting up a HyperStore Appliance machine. Do not use this on other host hardware.

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.

Note **Users of the AWS MMS version of HyperStore:** Use the *HyperStore for AWS MMS Quick Start Guide* to set up your system. That document includes information specific to setting up HyperStore to work with AWS MMS, as well as high level instructions for installing an on-premise HyperStore cluster. You can refer to this *HyperStore Installation Guide* if you need more detail on the topic of installing a HyperStore cluster.

This page left intentionally blank

Chapter 2. Preparing Your Environment

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

- **"DNS Set-Up"** (page 4)
- **"Load Balancing"** (page 6)

2.1. DNS Set-Up

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.

Note If you are doing just a small evaluation and do not require that external clients be able to access any of the HyperStore services, you have the option of using the lightweight domain resolution utility *dnsmasq* which comes bundled with HyperStore -- rather than configuring your DNS environment to support HyperStore service endpoints. If you're going to use *dnsmasq* you can skip ahead to **"Preparing Your Nodes"** (page 9). You will subsequently use the *configure-dnsmasq* option when you launch the HyperStore installation script, as described later in this document.

The table that follows shows the DNS entries that you must configure on your name servers to resolve HyperStore service endpoints. By default the HyperStore system automatically derives service endpoint values from your organization’s domain, which you will supply when you run the HyperStore interactive installer. The table shows the default format of each service endpoint. The default S3 endpoint formats are consistent with the format that Amazon uses for its S3 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. 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 below. 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.

In a production environment, each of these service endpoints should resolve to the IP addresses of two or more **load balancers** (configured for fail-over), with the load balancers in turn distributing request traffic across **all** the nodes in the cluster. In a multi-region system, the regional S3 endpoints should resolve to local region load balancers which distribute traffic across local region nodes; while the Admin and CMC service endpoints should resolve to load balancers in the default service region which distribute traffic to nodes in the default service region. For background information on service regions see "Service Regions Feature Overview" in the "Major Features" section of the HyperStore Administrator's Guide.

Note In an evaluation or testing environment, using round-robin DNS is an acceptable alternative to using load balancers. See the Example that follows the table.

DNS Entry	Default Format and Example	Description
S3 service endpoint (**one per service region**)	*s3-<region>.<your-domain>* *s3-tokyo.enterprise.com*	This is the service endpoint to which S3 client applications will submit requests. The *<region>* segment indicates the HyperStore service region. You must choose a service region name for your HyperStore installation, even if you intend to have only one service region. The region name must be lower case with no dots, dashes, underscores, or spaces. Make sure that the region name you supply when doing the installation

DNS Entry	Default Format and Example	Description
		matches the region name you used in your DNS configuration. 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-tokyo.enterprise.com* and *s3-osaka.enterprise.com*. 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. 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.
S3 service endpoint wildcard (**one per service region**)	_*.s3-<region>.<your-domain>_ _*.s3-tokyo.enterprise.com_	This S3 service endpoint wildcard entry is necessary to resolve S3 requests pertaining to specific storage buckets (which is nearly all S3 requests).
S3 static website service endpoint (**one per service region**)	*s3-website-<region>.<your-domain>* *s3-website-tokyo.enterprise.com*	This S3 service endpoint is used for buckets configured as static websites.
S3 static website endpoint wildcard (**one per service region**)	_*.s3-website-<region>.<your-domain>_ _*.s3-website-tokyo.enterprise.com_	This S3 static website endpoint wildcard entry is necessary to make S3 requests resolvable, for buckets configured as static websites.
Admin Service endpoint (**one per entire system**)	*s3-admin.<your-domain>* *s3-admin.enterprise.com*	This is the service endpoint for HyperStore's Admin API. The Cloudian Management Console accesses this RESTful HTTP API, and you can also access the API directly with a command line tool such as *cURL* or a client application of your own creation.
Cloudian Management Console (CMC) domain (**one per entire**	*cmc.<your-domain>* *cmc.enterprise.com*	The CMC is HyperStore's web-based console for making S3 requests (such as creating storage buckets or uploading objects) and performing system administration tasks.

DNS Entry	Default Format and Example	Description
system)		

2.1.0.1. Example

Below is an example of a round-robin DNS configuration for a three-node HyperStore system that has only one service region. The organization's top-level domain is *enterprise.com*. Note that HyperStore uses a peer-to-peer architecture in which the S3 Service, Admin Service, and CMC all run on every HyperStore node, and thus in the example all of the nodes are part of the round-robin for each service endpoint.

```
s3-tokyo.enterprise.com IN A 10.1.1.1
                             10.1.1.2
                             10.1.1.3
*.s3-tokyo.enterprise.com IN A  10.1.1.1
                                10.1.1.2
                                10.1.1.3
s3-website-tokyo.enterprise.com IN A 10.1.1.1
                                     10.1.1.2
                                     10.1.1.3
*.s3-website-tokyo.enterprise.com IN A 10.1.1.1
                                       10.1.1.2
                                       10.1.1.3
s3-admin.enterprise.com IN A 10.1.1.1
                             10.1.1.2
                             10.1.1.3
cmc.enterprise.com IN A 10.1.1.1
                        10.1.1.2
                        10.1.1.3
```

Note In a production environment you should use load balancers rather than round-robin DNS.

2.2. Load Balancing

HyperStore uses a peer-to-peer architecture in which each node in the cluster can service requests to the S3, Admin, or CMC service endpoints. In a production environment you should use load balancers to distribute S3, Admin, and CMC service endpoint requests evenly across all the nodes in your cluster. In your DNS configuration the S3, Admin, and CMC service endpoints should resolve to the IP addresses 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 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 For a non-production environment round-robin DNS is an acceptable alternative to using load balancers. For more information see **"DNS Set-Up"** (page 4).

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 see s3_proxy_protocol_enabled in common.csv.

This page left intentionally blank

Chapter 3. Preparing Your Nodes

To prepare your hosts for HyperStore software installation first confirm that they meet HyperStore **"Host Hardware and OS Requirements"** (page 9).

Then complete these node preparation tasks in this order:

1. **"Installing HyperStore Prerequisites"** (page 11)
2. **"Configuring Data Disks"** (page 13)
3. **"Running the Pre-Install Checks Script"** (page 15)

3.1. Host Hardware and OS Requirements

3.1.1. Hardware

The table below shows the recommended and minimum hardware specifications for individual host machines in a HyperStore system. Only Intel x86-64 systems are supported. (AMD x86-64 may work, but has not been tested.)

> **Note** For guidance regarding how many nodes you should use to meet your initial workload requirements, consult with your Cloudian sales or support representative. For guidance about ongoing HyperStore capacity management and cluster resizing, see "Cluster Resizing Feature Overview" in the *HyperStore Administrator's Guide*.

Recommended for production systems	• 1 CPU, 8 cores • 64GB RAM • 2 x 300GB SSD (for RAID-1 mirrored hosting of the OS as well as Cassandra and Redis databases storing system metadata) • 12 x 4TB HDD (for *ext4* file systems storing object data) (JBOD, no RAID) • 2x10GbE Ports **Note** If you plan to use **erasure coding** for object data storage, 2 CPUs per node is recommended. Also, be aware that the higher your erasure coding *m* value (such as with *k+m* = 9+3 or 8+4), the higher the need for Cassandra metadata storage capacity. Consult with your Cloudian representative to ensure that you have adequate Cassandra storage capacity to support your desired *m* value.
Minimum for production systems	• 1 CPU, 8 cores • 32GB RAM • 2 x 160GB SSD (for RAID-1 mirrored hosting of the OS as well as Cassandra and Redis databases storing system metadata) • 12 x 2TB HDD (for *ext4* file systems storing object data) (JBOD, no RAID) • 1x1GbE Port

Minimum for installation	HyperStore software can be installed on a single host that has just one data drive. The host should have at least 1GB of hard drive space, at least 16GB RAM, and preferably at least 8 processor cores. If you install HyperStore software on a host with less resources than this, the install script will display a warning about the host having less than recommended resources. If you try to install HyperStore software on a host with less 100MB hard drive space or less than 2GB RAM, the installation will abort.

3.1.2. Operating System

To perform a fresh installation of HyperStore 7.1 you must have a **RHEL 7.x or CentOS 7.x** Linux operating system on each host. RHEL/CentOS 6.x is not supported for fresh installs of HyperStore 7.1. HyperStore does not support other types of Linux distribution, or non-Linux operating systems.

If you have not already done so, install RHEL 7.x or CentOS 7.x in accordance with your hardware manufacturer's recommendations.

The port access restricting services **firewalld, SELinux, and iptables must be disabled** on each host. HyperStore nodes sometimes communicate with each other via JMX, and when they do, after initial connection establishment on the designated JMX port (see **"HyperStore Listening Ports"** (page 33)) a random port is used for continued communication. Therefore there cannot be any port restrictions on internal communication between HyperStore nodes. The HyperStore installation will abort if *firewalld*, *SELinux*, or *iptables* is running on a host.

To disable *filewalld*:

```
[root]# systemctl stop firewalld
[root]# systemctl disable firewalld
```

Note RHEL/CentOS 7 uses *firewalld* by default rather than the *iptables* service, so you do not need to take action in regard to *iptables* unless you installed and enabled the *iptables* service. 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.

After verifying that your hosts meet hardware and OS requirements, proceed to **"Installing HyperStore Prerequisites"** (page 11).

3.1.2.1. Automatic Exclusions to OS Package Updates

As part of HyperStore installation, the HyperStore installation script will install prerequisites including Puppet, Facter, and Ruby on your HyperStore host machines. If you subsequently use *yum update* to update your OS packages, HyperStore automatically excludes Puppet, Facter, and Ruby 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 */etc/yum/pluginconf.d/versionlock.list* on your host machines. Do not remove any entries from the configured exclusion list.

3.2. Installing HyperStore Prerequisites

Follow these steps to install and configure HyperStore prerequisites on all of your nodes. Working from a single node you will be able to perform this task for your whole cluster.

1. Choose any one of your nodes to serve as the "Puppet Master" node, and log into the node. From this node you will orchestrate the HyperStore installation, and throughout the life of your HyperStore system you will use this node to orchestrate cluster configuration (leveraging the **Puppet** software that's integrated with HyperStore).

   > **Note:** The Puppet Master node must be one of your HyperStore nodes. It cannot be a separate node outside of your HyperStore cluster.

2. On the Puppet Master node create a directory */root/CloudianPackages* and copy the HyperStore product package (*.bin* file) and your Cloudian license file (*.lic* file) into that directory. This will be your "installation staging directory". Your installation staging directory must persist throughout the life of your HyperStore system -- do not delete this directory.

   > **Note** You can use a different directory as the installation staging directory if you wish, by placing your HyperStore product package *.bin* file and license file in a directory other than */root/CloudianPackages*. If you do so you will need to adjust the HyperStore setup tool settings as described later in this procedure. Remember that the staging directory must persist throughout your HyperStore system's lifetime.

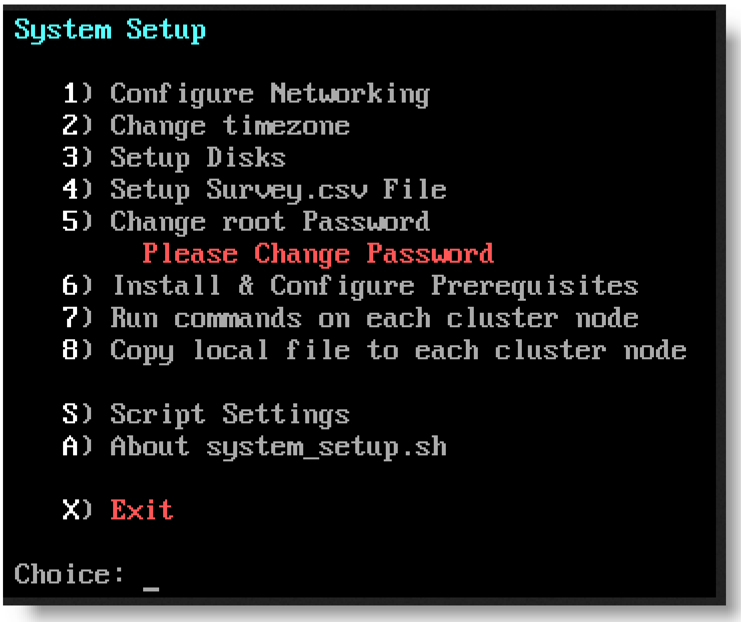
3. Change into the installation staging directory and then run the commands below to unpack the HyperStore package:

   ```
   [root]# chmod +x CloudianHyperStore-7.1.bin
   [root]# ./CloudianHyperStore-7.1.bin <license-file-name>
   ```

4. Still in the staging directory, launch the *system_setup.sh* tool.

   ```
   [root]# ./system_setup.sh
   ```

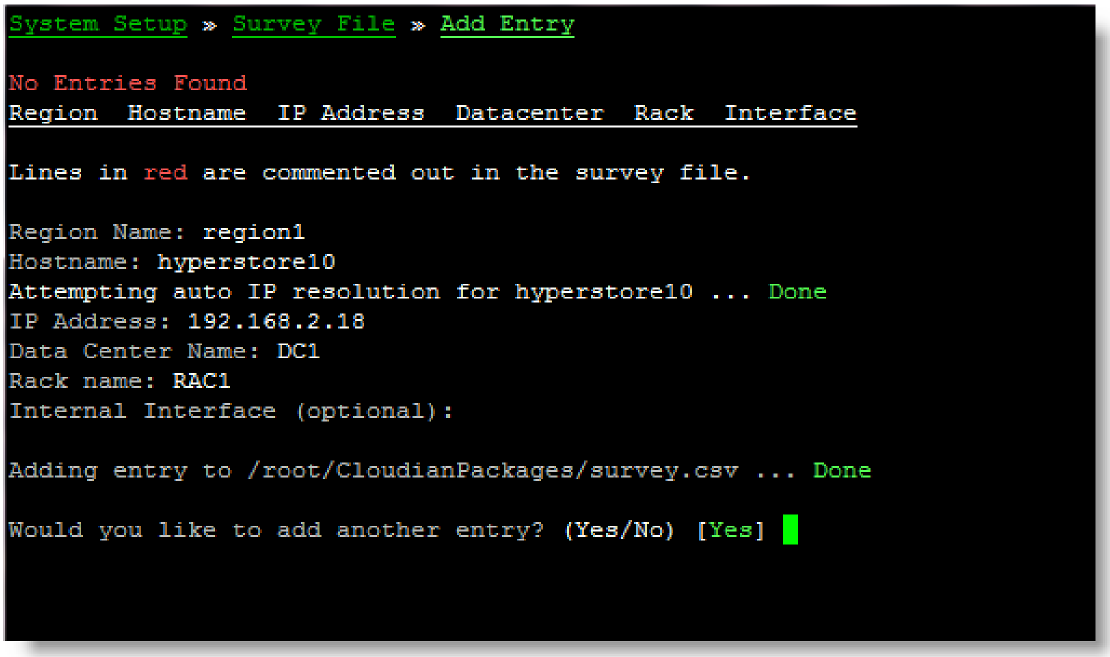
   This displays the tool's main menu.



```
System Setup

    1) Configure Networking
    2) Change timezone
    3) Setup Disks
    4) Setup Survey.csv File
    5) Change root Password
         Please Change Password
    6) Install & Configure Prerequisites
    7) Run commands on each cluster node
    8) Copy local file to each cluster node

    S) Script Settings
    A) About system_setup.sh

    X) Exit

Choice: _
```

> **Note** If you are using an installation staging directory other than */root/CloudianPackages*, from the setup tool's main menu enter "**s**" for Script Settings. Then in the Script Settings menu enter "**8**" for Staging Directory, and follow the prompts to specify your staging directory. Then return to the setup tool's main menu and continue with the steps below.

5. From the setup tool's main menu, enter "**4**" for Setup Survey.csv File and follow the prompts to create a cluster survey file with an entry for each of your HyperStore nodes (including the Puppet Master node). For each node you will enter a region name, hostname, public IP address, data center name, and rack name.



```
System Setup » Survey File » Add Entry

No Entries Found
Region  Hostname  IP Address  Datacenter  Rack  Interface

Lines in red are commented out in the survey file.

Region Name: region1
Hostname: hyperstore10
Attempting auto IP resolution for hyperstore10 ... Done
IP Address: 192.168.2.18
Data Center Name: DC1
Rack name: RAC1
Internal Interface (optional):

Adding entry to /root/CloudianPackages/survey.csv ... Done

Would you like to add another entry? (Yes/No) [Yes]
```

Note For each node the hostname that you enter must exactly match the node's hostname -- as would be returned by running the *hostname* command on the node -- including matching the case. HyperStore does not support having spaces within your hostnames, but does support hostnames that have periods, dashes, or underscores. In the region, data center, and rack names do not use spaces, periods, or underscores.

For each node there is also an optional prompt for specifying the node's internal interface name. You only need to provide this information if the node is using a different internal interface than the rest of the nodes in the cluster. If all nodes use the same internal interface you can leave this value empty for each node (later in the installation process you will specify a default internal interface name for the cluster).

After you've added an entry for each node, return to the setup tool's main menu.

Note Based on your input at the prompts, the setup tool creates a survey file named *survey.csv*, in your installation directory. This file must remain in your staging directory -- do not delete or move it. For more information about the survey file, see the Installation Reference topic **"Cluster Survey File (survey.csv)"** (page 40).

6. If you want to change the root password for your nodes, do so now by entering "**5**" from the main menu and following the prompts. It's recommended to use the same password for each node. Otherwise the pre-installation cluster validation tool described later in the procedure will not be fully functional.
7. Back at the setup tool's main menu enter "**6**" for Install & Configure Prerequisites. When prompted about whether you want to perform this action for all nodes in your survey 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 either for the whole cluster (if, as recommended, each node has the same root password) or for each node in turn (if the nodes have different passwords). When the prerequisite installation completes for all nodes, return to the setup tool's main menu.

Note: If Selinux was enabled on your hosts, Step 7 disabled it (or more specifically, changed it to "permissive" mode for the current running session and changed the configuration so it will be disabled for future boots of the host). For background information see **"Host Hardware and OS Requirements"** (page 9).

After installing HyperStore prerequisites, proceed to **"Configuring Data Disks"** (page 13).

3.3. Configuring Data Disks

Note These data disk setup instructions do not apply if you are doing a very simple HyperStore evaluation by installing on a host or hosts that have just a single disk (on which will be stored the OS and application data as well as S3 object data). If your host or hosts have just one disk each, skip ahead to **"Running the Pre-Install Checks Script"** (page 15).

IMPORTANT: Cloudian recommends using the HyperStore system setup tool to format and mount your dedicated data disks on each host, as described in this section. **If you have already formatted and**

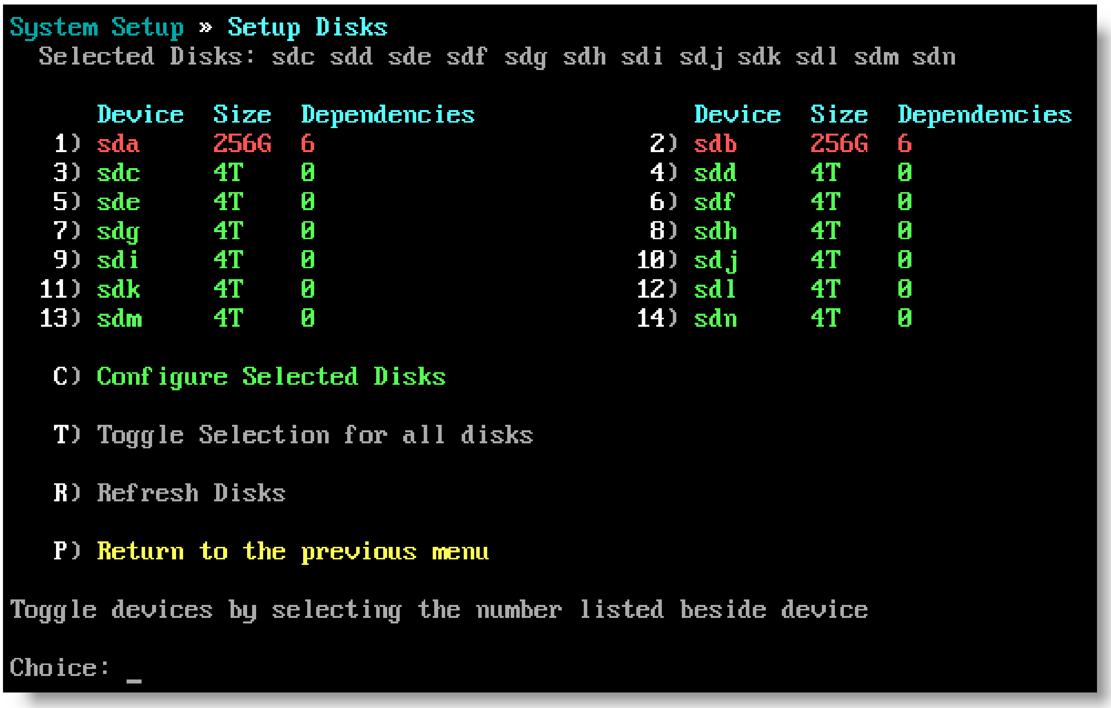
mounted your data disks using third party tools, then instead of using the instructions in this section do the following:

1. Confirm that your disk setup meets the requirements described in the Installation Reference topic **"File System Requirements"** (page 36).
2. Create a default mount point list for the cluster as described in the Installation Reference topic **"Data Directory Mount Point List (fslist.txt)"** (page 39).

Then proceed to **"Running the Pre-Install Checks Script"** (page 15).

Follow these steps to format and mount the data disks for each of your nodes one node at a time, using the HyperStore setup tool.

1. On the Puppet Master node, from the setup tool's main menu enter "**3**" for Setup Disks. This displays the Setup Disks menu.



```
System Setup » Setup Disks
  Selected Disks: sdc sdd sde sdf sdg sdh sdi sdj sdk sdl sdm sdn

      Device  Size  Dependencies                        Device  Size  Dependencies
   1) sda     256G  6                               2) sdb     256G  6
   3) sdc     4T    0                               4) sdd     4T    0
   5) sde     4T    0                               6) sdf     4T    0
   7) sdg     4T    0                               8) sdh     4T    0
   9) sdi     4T    0                              10) sdj     4T    0
  11) sdk     4T    0                              12) sdl     4T    0
  13) sdm     4T    0                              14) sdn     4T    0

   C) Configure Selected Disks

   T) Toggle Selection for all disks

   R) Refresh Disks

   P) Return to the previous menu

Toggle devices by selecting the number listed beside device

Choice: _
```

2. 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 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 (such as "**3**"). Once you're satisfied with the selected list in green font, enter "**c**" for Configure Selected Disks (or if you are configuring 15 or more disks on the node enter "**d**" for Configure Selected Disks with no ext4 lazy_init) and follow the prompts to have the tool configure the selected disks.

3. After you've configured all the data disks on the Puppet Master node, exit the setup tool and log out of the node.

4. Log into your other nodes one at a time and use the system setup tool to Setup Disks, as described above. On each node the tool will be in the same directory as it is on your Puppet Master node (the

"Install and Configure Prerequisites" operation that you completed previously creates that directory on each node and pushes a copy of the setup tool out to it).

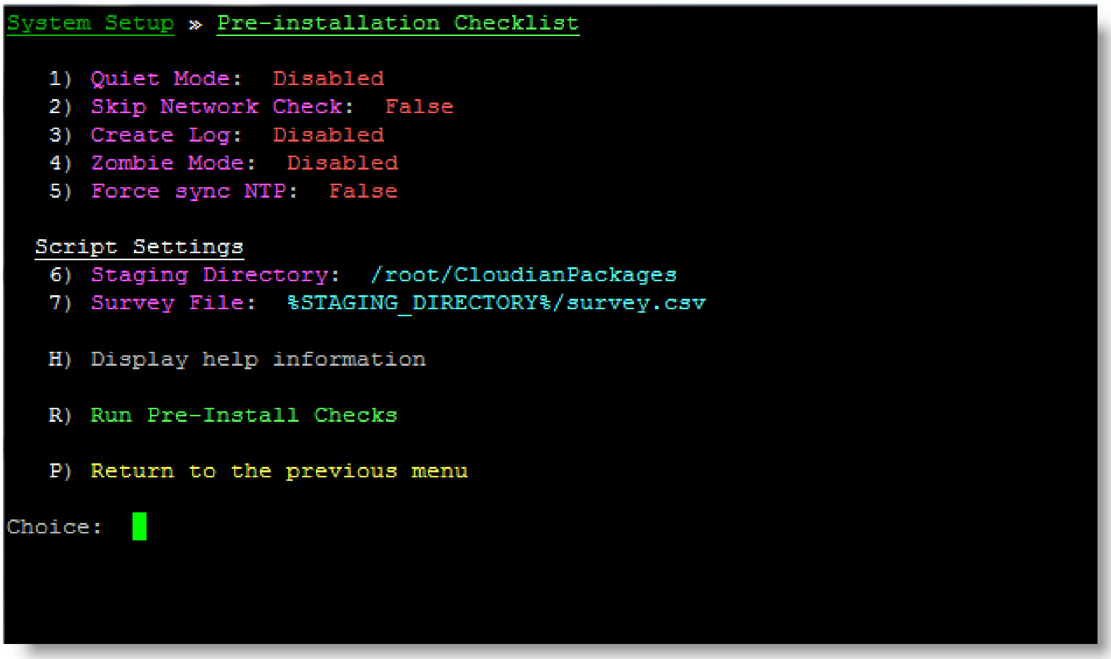
> **Note:** When configuring disks, the setup script by default invokes the *ext4* "lazy initialization" feature by which a background, low-priority process of disk initialization will continue for up to two to three hours after you've used the script to format the disks. You can continue with the installation and you can use the HyperStore system after it's installed, but you should wait for a few hours after the installation before running any performance tests.

After configuring disks on all your hosts, proceed to **"Running the Pre-Install Checks Script"** (page 15).

3.4. 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 enter "**r**" for Run Pre-Installation Checks. This displays the Pre-Installation Checklist menu.



```
System Setup » Pre-installation Checklist

   1) Quiet Mode:  Disabled
   2) Skip Network Check:  False
   3) Create Log:  Disabled
   4) Zombie Mode:  Disabled
   5) Force sync NTP:  False

  Script Settings
   6) Staging Directory:  /root/CloudianPackages
   7) Survey File:  %STAGING_DIRECTORY%/survey.csv

   H) Display help information

   R) Run Pre-Install Checks

   P) Return to the previous menu

Choice:
```

2. From the Pre-Installation Checklist menu enter "**r**" for Run Pre-Install Checks. The script then checks to verify that your cluster meets all requirements for hardware, prerequisite 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 17).

Chapter 4. Installing a New HyperStore System

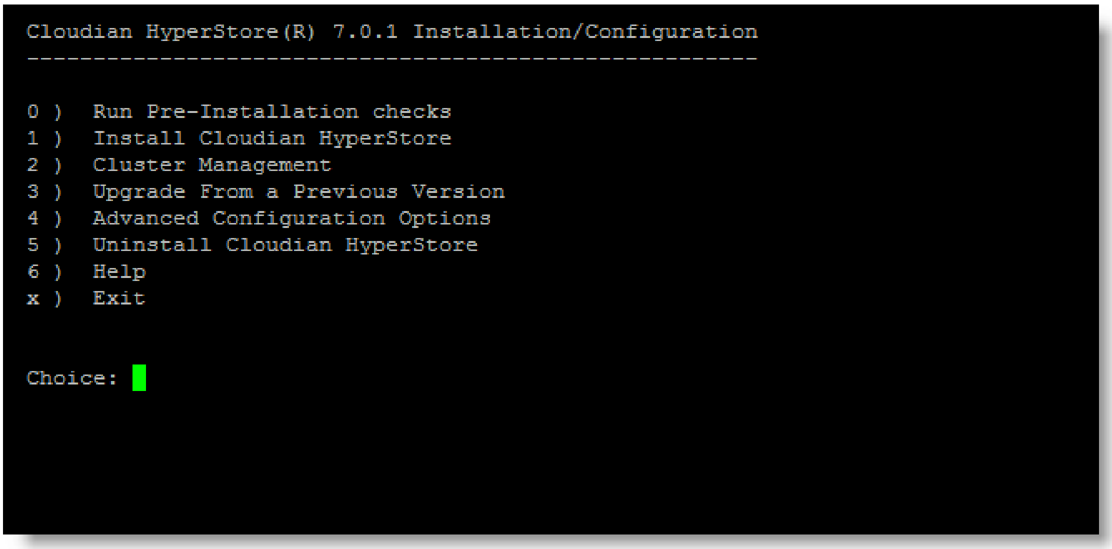
This section describes how to do a fresh installation of HyperStore 7.1 software, after **"Preparing Your Environment"** (page 3) and **"Preparing Your Nodes"** (page 9). From your Puppet Master node you can install HyperStore software across your whole cluster.

1. On your Puppet Master node, change into your installation staging directory. Then launch the HyperStore installation script as follows:

```
[root]# ./cloudianInstall.sh -s survey.csv
```

> **Note If you have not configured your DNS environment** for HyperStore (see **"DNS Set-Up"** (page 4)) and you want to instead use the included *dnsmasq* utility to resolve HyperStore service endpoints, launch the install script with the *configure-dnsmasq* option as shown below. This is not appropriate for production systems.
>
> *[root]# ./cloudianInstall.sh -s survey.csv configure-dnsmasq*
>
> For more script launch options, see the Installation Reference topic **"cloudianInstall.sh Command Line Options"** (page 41).

When you launch the installer the main menu displays:



```
Cloudian HyperStore(R) 7.0.1 Installation/Configuration
-------------------------------------------------------

0 )  Run Pre-Installation checks
1 )  Install Cloudian HyperStore
2 )  Cluster Management
3 )  Upgrade From a Previous Version
4 )  Advanced Configuration Options
5 )  Uninstall Cloudian HyperStore
6 )  Help
x )  Exit


Choice:
```

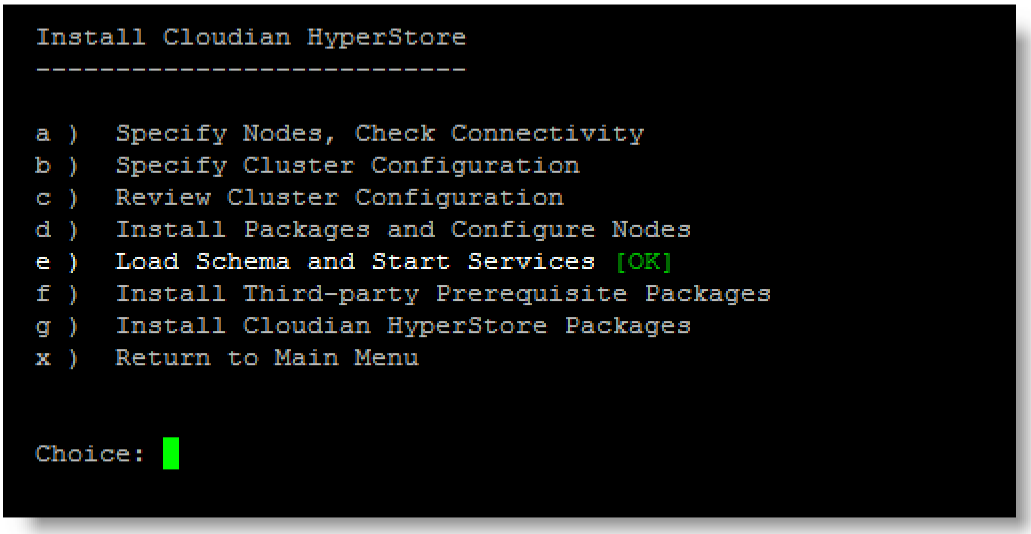
> **Note** The installer menu includes an item "0" for Run Pre-Installation Checks. This is the same pre-installation check that you already ran from within the *system_setup.sh* tool as described in **"Running the Pre-Install Checks Script"** (page 15) -- so you can ignore this option in the installer 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 survey file (which you created earlier during the node preparation task).

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

   - The name of the **internal interface** that your nodes will use by default for internal cluster communications. For example, *eth1*. Cassandra, Redis, and the HyperStore Service are among the services that will utilize the internal interface for intra-cluster communications.
   - The "**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, like "DC1:3,DC2:2". The default is 3 replicas per DC.
   - 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 you can enter 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. See **"DNS Set-Up"** (page 4) for details about HyperStore service endpoints.

   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:



```
Install Cloudian HyperStore
---------------------------

a )  Specify Nodes, Check Connectivity
b )  Specify Cluster Configuration
c )  Review Cluster Configuration
d )  Install Packages and Configure Nodes
e )  Load Schema and Start Services [OK]
f )  Install Third-party Prerequisite Packages
g )  Install Cloudian HyperStore Packages
x )  Return to Main Menu


Choice:
```

   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").

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. You can then further exercise your HyperStore system using the Cloudian Management Console, as described in **"Getting Started with a Newly Installed HyperStore System"** (page 21).

Note For troubleshooting information, see the Installation Reference topic **"Installation Troubleshooting"** (page 32).

This page left intentionally blank

Chapter 5. Getting Started with a Newly Installed HyperStore System

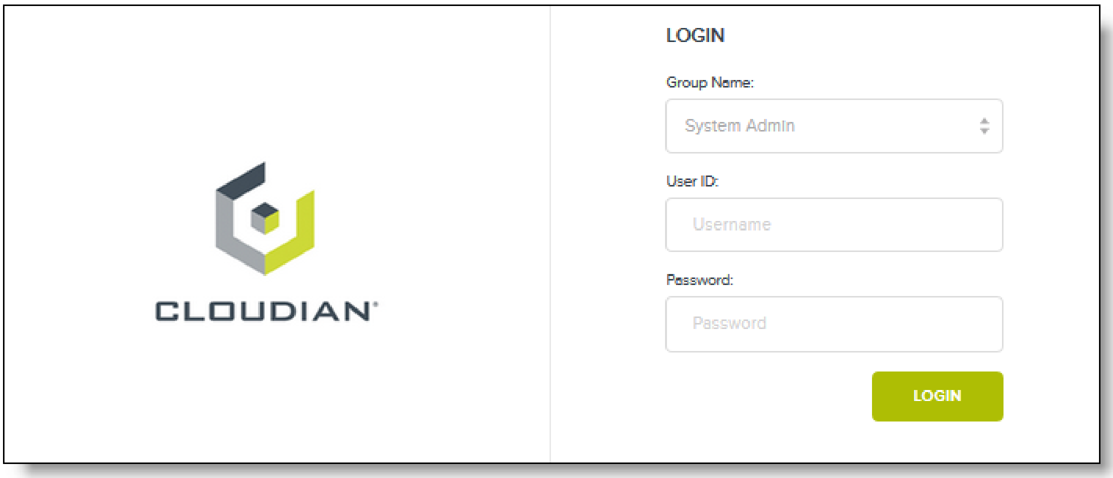
> **Note** For complete HyperStore documentation see the HyperStore Help, which you can access through the CMC's "Help" button or by opening the file *<installation_staging_directory>/doc/HyperStoreHelp/HyperStoreHelp.html* on your Puppet master node.

Once a newly installed HyperStore system is up and running, you can use the Cloudian Management Console (CMC) to take the system for a test drive. Follow these steps:

1. Point a browser to *http://<CMC_host_IP_address>:8888/Cloudian*

   Since the CMC runs on all of your HyperStore nodes by default, you can use any node's IP address.

2. The connection will automatically switch over to SSL and you will get a certificate warning. Follow the prompts to add an exception for the certificate and accept it. You should then see the CMC’s login screen.



LOGIN

Group Name: System Admin

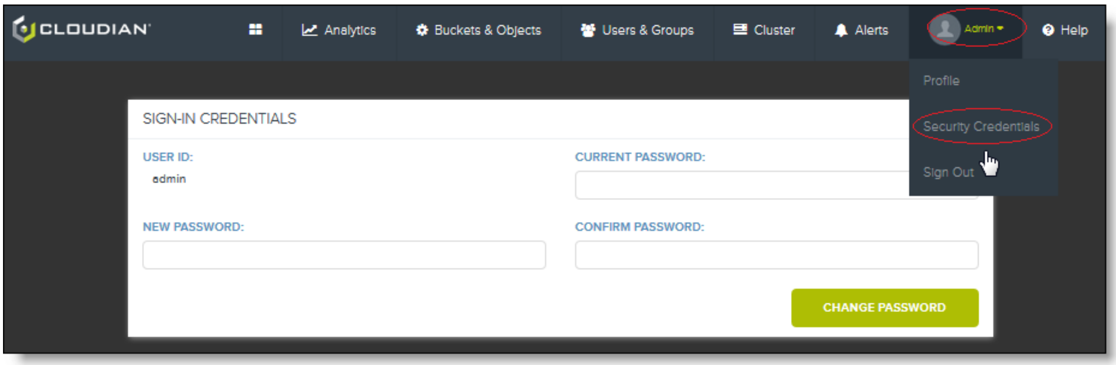
User ID: Username

Password: Password

LOGIN

3. Log in with the system administrator user ID *admin* and default password *public*.

4. In the upper right corner of the CMC UI, hold your cursor over your user ID ("admin"). From the drop-down menu that displays, select **Security Credentials** to open the **Sign-In Credentials** interface. For security, change the system administrator password.



SIGN-IN CREDENTIALS

USER ID: admin

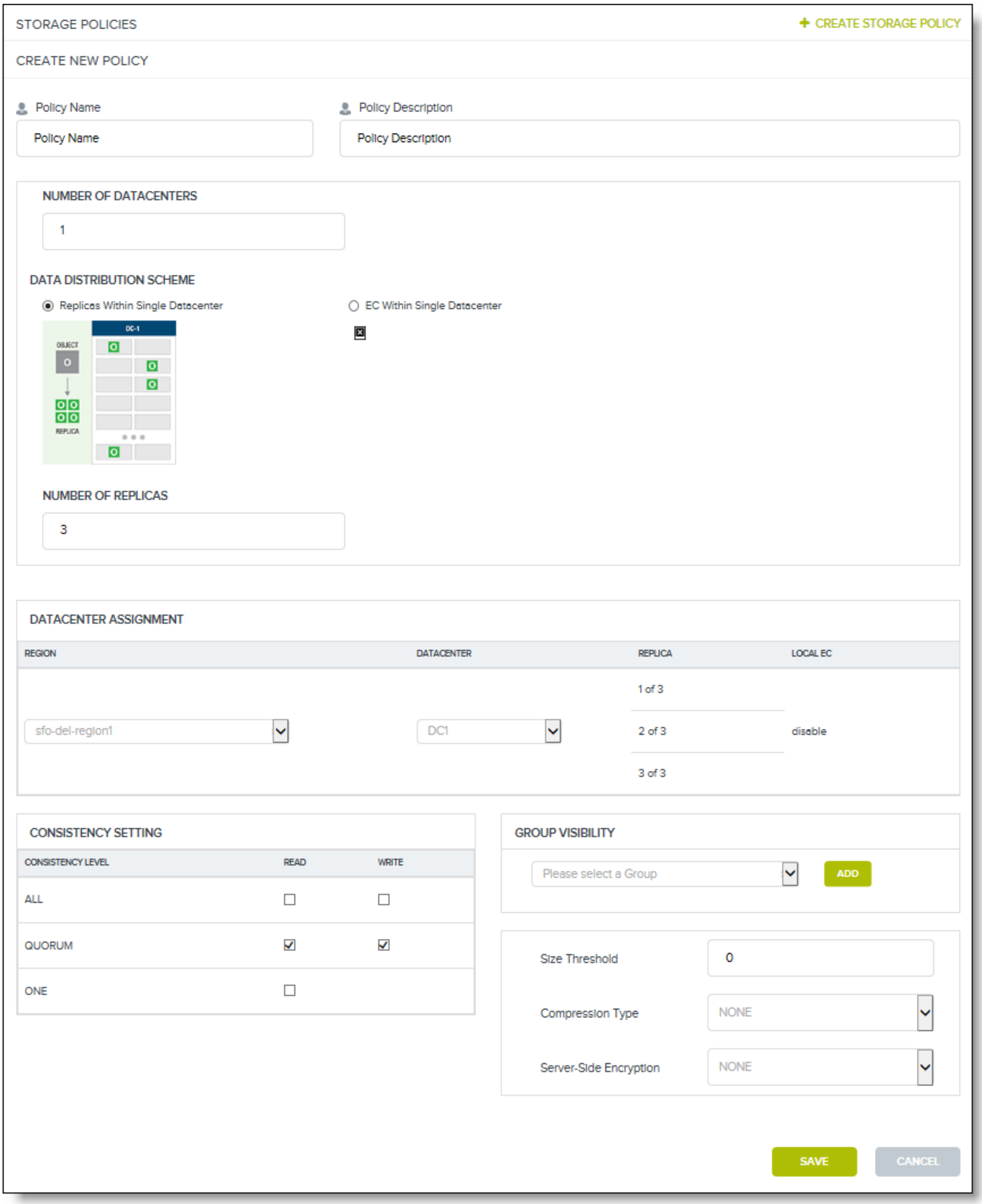
CURRENT PASSWORD:

NEW PASSWORD:

CONFIRM PASSWORD:

CHANGE PASSWORD

5. Select the **Cluster** tab, then **Storage Policies**, then **Create Storage Policy**. This opens the **Create New Policy** interface. Create and **Save** a default storage policy for your system. A storage policy is a method of storing and protecting S3 object data and object metadata. Leave the "Group Visibility" unspecified so that this policy is visible to all groups. (At a later time you can edit this policy; create additional policies; and assign a different policy to be the system default policy if you wish).



Policy Name: Policy Name

Policy Description: Policy Description

NUMBER OF DATACENTERS: 1

DATA DISTRIBUTION SCHEME

● Replicas Within Single Datacenter ○ EC Within Single Datacenter

NUMBER OF REPLICAS: 3

DATACENTER ASSIGNMENT

REGION	DATACENTER	REPLICA	LOCAL EC
sfo-del-region1	DC1	1 of 3 2 of 3 3 of 3	disable

CONSISTENCY SETTING

CONSISTENCY LEVEL	READ	WRITE
ALL	☐	☐
QUORUM	☑	☑
ONE	☐	

GROUP VISIBILITY

Please select a Group ADD

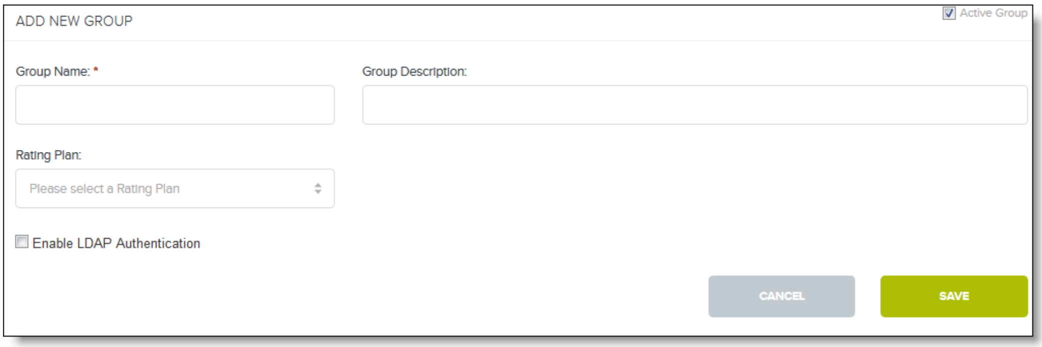
Size Threshold	0
Compression Type	NONE
Server-Side Encryption	NONE

For detail about the available options when you configure a new storage policy see Add a Storage Policy.

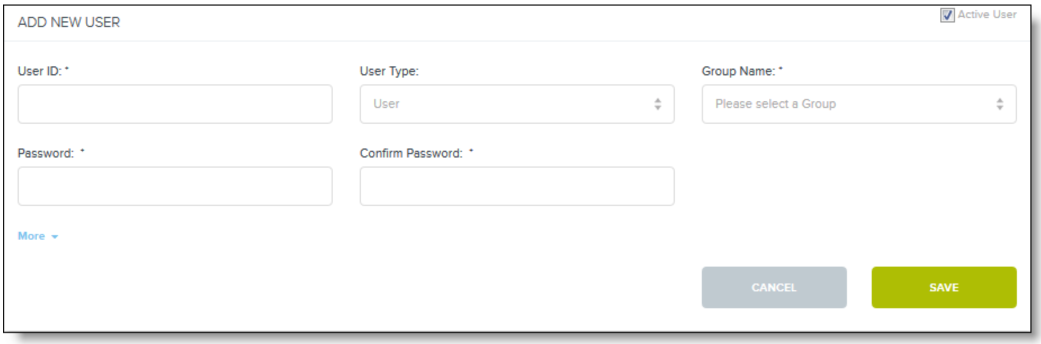
6. Create a regular S3 service user account that will enable you to test the system’s S3 object storage services.

Note The system administrator role cannot have its own S3 storage service account.

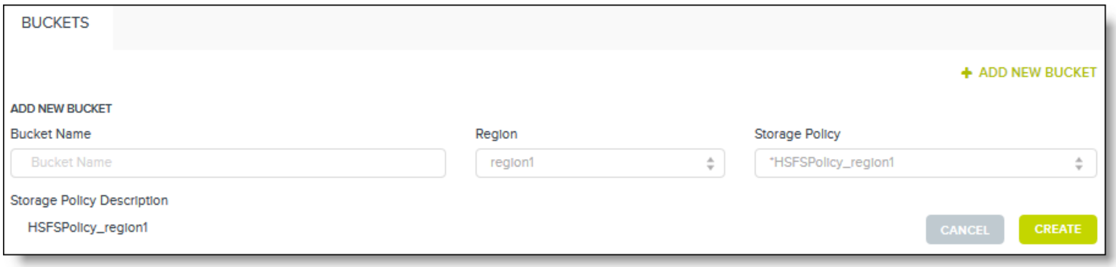
a. Select **Users & Groups** → **Manage Groups** → **New Group** to open the **Add New Group** interface. Create a new user group.



b. Select **Users & Groups** → **Manage Users** → **New User** to open the **Add New User** interface. Create a new regular user, assigned to the user group that you created in the previous step. Make a note of the group name, user ID, and the password that you assign to the user so that you will be able to log in as that user.



7. Log out of the CMC, and then log back in as the new user that you created. The CMC now displays only the functions that are available to regular service users, including the **Buckets & Objects** interface.

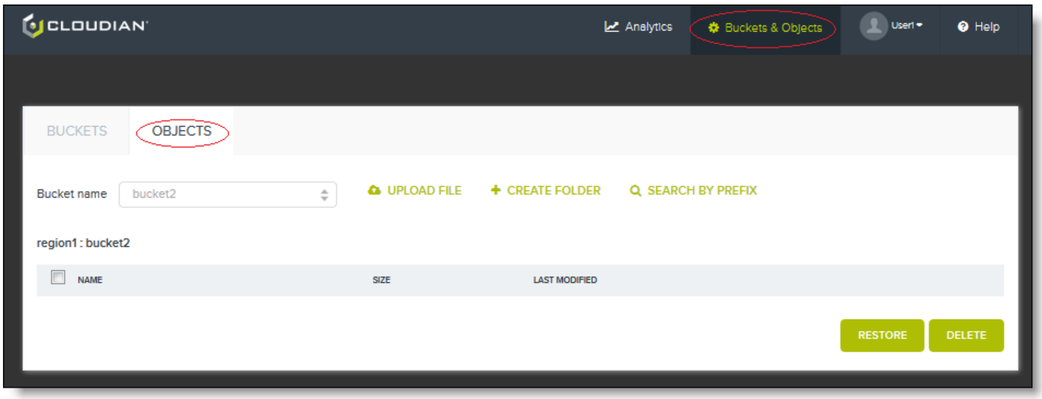


Note As an alternative to logging out of the CMC, you can remain logged in as an administrator and use the **Manage Users** page to retrieve the user that you just created. Then click the "View User Data" link for that user. The **Buckets & Objects** interface will display for you as if you were that user.

8. Experiment with the CMC **Buckets & Objects** interface. For example:

a. Add a new storage bucket by entering a bucket name and clicking **Create**. (You must create a bucket before you can upload any objects.)

b. Use the **Objects** tab to create a folder in the bucket by entering a folder name and clicking **Create Folder**.



a. Open the folder then upload a file into it by using the **Upload File** function.

b. Verify that the uploaded file appears in the folder contents list, then verify that you can download the file by clicking on the file name.

Chapter 6. Upgrading an Existing HyperStore System

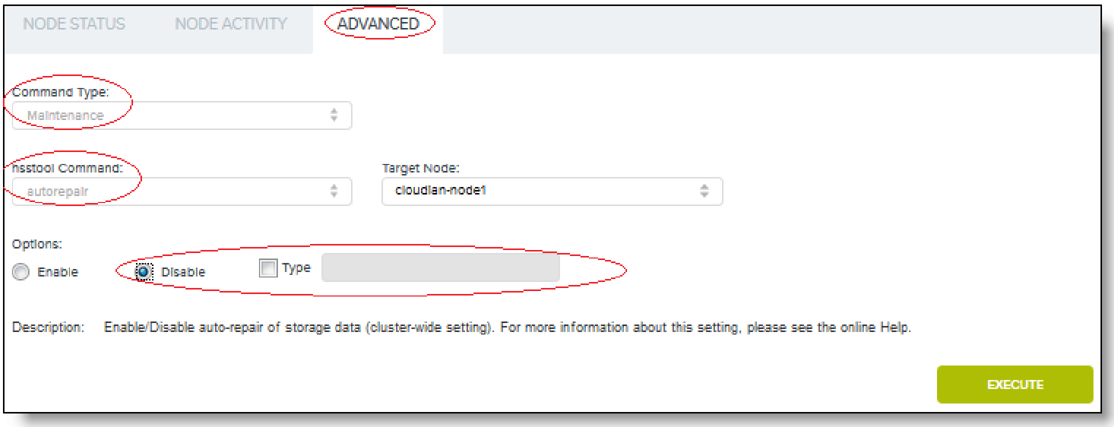
The instructions that follow are for upgrading to HyperStore version 7.1 **from version 6.2 or newer**.

> **IMPORTANT:** If you are currently on a version older than 6.2, do not use the procedure described here. Instead contact Cloudian Support to request instructions for your particular upgrade path.

6.1. Before Upgrading Your System

Before upgrading your HyperStore system, log into the CMC and do the following:

1. Use's the CMC's **Node Advanced** page to temporarily **shut down the scheduled auto-repair feature**, so that no new auto-repairs kick off while you are performing the upgrade. On the Node Advanced page execute the Maintenance command *autorepair* with the Disable option. The target node can be any node in the cluster. Leave the "Type" option unspecified so that all auto-repair types (Replicas, EC, and Cassandra) are disabled.



Command Type: Maintenance

hsstool Command: autorepair

Target Node: cloudian-node1

Options: ○ Enable ● Disable □ Type

Description: Enable/Disable auto-repair of storage data (cluster-wide setting). For more information about this setting, please see the online Help.

2. Use the CMC's **Repair Status** page to **verify that no repairs or rebalancing operations of any type are currently in progress** in your system. All nodes should have an "All Clear" status (green cube icon). If any type of repair (including proactive repair) or rebalance is in progress, wait until it completes before you proceed with the upgrade.

   > **Note:** The upgrade will abort if a repair or rebalance is running in your system. In a multi-region system the upgrade will abort if a repair or rebalance is running in any of your HyperStore regions.

3. Use the CMC's **Dashboard** page to **verify that no services are currently down** in your cluster -- the dashboard's "Cluster Health" section should show a green circle and status should say "Healthy". If the circle is orange then one or more services are down. In that case, check the CMC's **Alerts** page for alerts regarding downed services. **Resolve any errors** identified by alerts, working with Cloudian Support if necessary. **Restart any services** that are down.

IMPORTANT: If you are using Xen, Amazon EC2, or Logical Volume Manager (LVM), contact Cloudian Support before upgrading your system.

6.2. Upgrading Your System

To perform the upgrade to HyperStore version 7.1:

1. On your Puppet master node, create a **new staging directory**, with a different name than the directory that you used for your existing cluster. From here forward you will be working in the new staging directory.

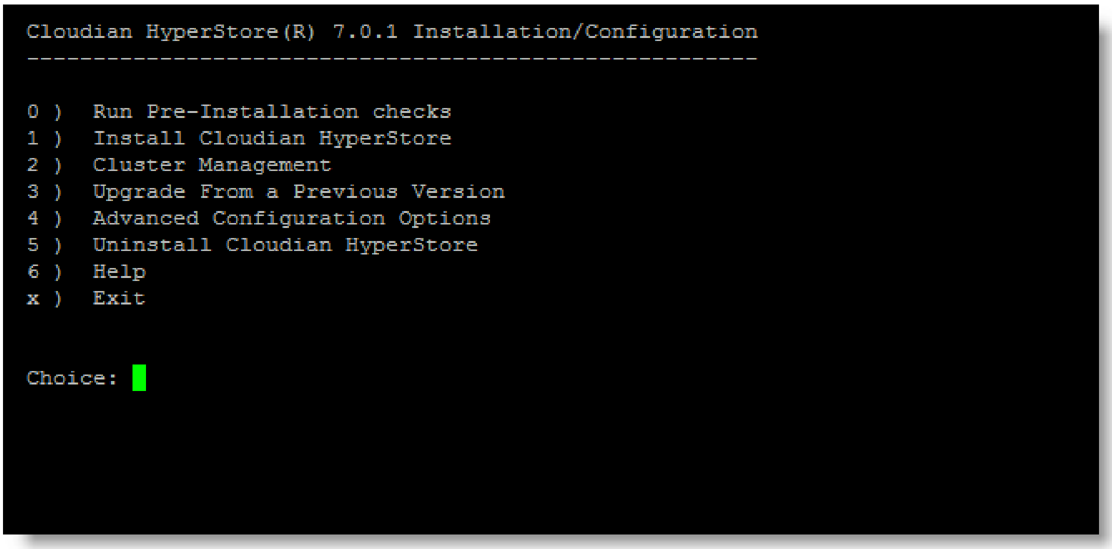
   Note: Do not make the new staging directory a sub-directory under the staging directory of your existing HyperStore version.

2. Download the HyperStore 7.1 package. (If you need download access information see **"HyperStore Installation Introduction"** (page 1)). Place the package into your new staging directory.
3. Copy your HyperStore license file into the new staging directory. (Your current license file is located under */etc/cloudian-<your-current-version>-puppet/modules/baselayout/files* and ends with suffix *.lic*. Copy this file to the new staging directory.)
4. In your new staging directory, run the following command to unpack the HyperStore package:

```
root# ./CloudianHyperStore-7.1.bin <license-file-name>
```

5. In your new staging directory, launch the installer:

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



```
Cloudian HyperStore(R) 7.0.1 Installation/Configuration
-------------------------------------------------------

0 )  Run Pre-Installation checks
1 )  Install Cloudian HyperStore
2 )  Cluster Management
3 )  Upgrade From a Previous Version
4 )  Advanced Configuration Options
5 )  Uninstall Cloudian HyperStore
6 )  Help
x )  Exit


Choice: 
```

6. From the installer main menu enter "**3**" for "Upgrade from <your version number> to 7.1". Then at the prompt, confirm that you wish to continue with the automated upgrade.

The upgrade script will first check your Puppet configuration template files (**.erb* files) from your current HyperStore system to determine whether you made any customizations to those settings (changes from the default values). If you have made such configuration changes, the installer creates a text file that lists those changes and prompts you to review the text file. Depending on which **.erb* file setting(s) you customized, you may need to manually edit the new HyperStore version's configuration templates (under */etc/cloudian-<new-version-#>-puppet/modules* on the Puppet master node) before continuing with the upgrade process, if you want to carry your customizations forward to your upgraded system. To see which **.erb* file settings support automatic migration of customized values and which require you to make manual edits in order to migrate your customizations, see **"Puppet Template (.erb) Settings Automatically Migrated During Upgrade"** (page 28). If you do make manual edits to the new HyperStore version's **.erb* template settings during the upgrade, you do not need to do a Puppet push -- just edit the configuration templates and then resume the automated upgrade process.

After the upgrade successfully completes, proceed to **"After Upgrading Your System"** (page 27).

> **Note**
> * Once you've started the upgrade, you cannot *<ctrl>-c* out of it.
>
> * If you have initiated the upgrade through a remote terminal, and the connection between the terminal and the Puppet master node is subsequently lost, the upgrade will continue.
>
> * If the upgrade fails and rolls back, you can get details about the failure from the *cloudian-installation.log* and the *cloudian-upgrade.log*, both of which are generated in the installation staging directory. The upgrade process also generates an *upgrade-logNconfig*.tgz* "S.O.S" tar file (which packages together multiple upgrade-related files) that you can provide to Cloudian Support in the event of upgrade problems.

6.3. After Upgrading Your System

After all HyperStore nodes have been upgraded, verify that all services are running and that the HyperStore version is now 7.1:

1. After the automated upgraded completes, you should be taken back to the main menu of the HyperStore installer. The first post-upgrade step is to confirm that all your HyperStore services are up and running:
   a. From the installer’s main menu select "Cluster Management".
   b. From the Service Management sub-menu that displays select "Manage Services".
   c. At the “Select a service to manage:” prompt, select All Services.
   d. At the “Enter command” prompt, type *status*.

   All services on all nodes should then indicate that they are running.
2. Next, confirm that the HyperStore software version is correct:
   a. Still on the Service Management menu, at the “Select a service to manage:” prompt select the S3 Service.
   b. At the “Enter command” prompt, type *version*.

   On all nodes the S3 version should indicate version 7.1.

   After confirming the version you can exit the installer.
3. Use the CMC to check on your upgraded cluster:

- On the **Node Advanced** page, select command type Info then execute the "repairqueue" command to **verify that auto-repair is enabled** for replica, EC, and Cassandra data. (Although you disabled auto-repair prior to doing the upgrade, the system automatically re-enables auto-repair at the end of the upgrade process).
- On the **Manage Users** page, confirm that you can retrieve users.
- Log out of the CMC as system admin and log back in as a regular user, and then confirm that you can successfully download and upload objects.

4. If prior to the upgrade you had made any customizations to the branding of the CMC interface, only your customized logos and customized application name will be retained after the upgrade. You will need to re-implement any changes that you had made to the browser tab title and/or the color scheme, by again following the instructions for Rebranding the CMC UI.

You are now done with upgrading to HyperStore 7.1.

6.4. Upgrading Your Hosts' OS to RHEL/CentOS 7.x

After upgrading to HyperStore 7.0, you must upgrade your host operating systems to RHEL/CentOS 7.x. Although HyperStore systems upgraded to HyperStore 7.0 will run on RHEL/CentOS 6.x as well as RHEL/CentOS 7.x, **HyperStore 7.1 and later will only run on RHEL/CentOS 7.x**. Cloudian recognizes that upgrading your hosts to RHEL/CentOS 7.x will take days or weeks, but you must complete the process before you will be able to upgrade to HyperStore 7.1 or later. Contact Cloudian Professional Services for guidance and assistance on upgrading your HyperStore hosts to RHEL/CentOS 7.x**.**

6.5. Puppet Template (.erb) Settings Automatically Migrated During Upgrade

If in your existing HyperStore system you have made changes to the settings in Puppet template (**.erb*) files, then during upgrade the installer creates a text file that lists those changes -- a "diff" file -- and prompts you to review the file.

First, open a separate terminal instance in which you can open and review the "diff" file that the installer created. Then, whether or not this requires any further action on your part depends on whether your customized settings identified in the "diff" file are among those listed in the middle column of the table below. The table lists **.erb* file settings that, starting with HyperStore 7.0, are under control by new settings in *common.csv*. (This 7.0 change will make future upgrades easier in regard to configuration migration.)

- If an **.erb* file setting that you have changed is **listed in the middle column** of the table below, the installer will automatically migrate your customization to the new version of the setting in *common.csv*. This requires no action on your part. For example, if in your existing HyperStore system you had set the *hyperstore-server.properties.erb* setting *auto.repair.computedigest.run.number* to "3" (rather than its default value of "0"), then the installer during upgrade to 7.0 will automatically set the new *common.csv* setting *auto_repair_computedigest_run_number* to "3".
- If an **.erb* file setting that you have changed is **not listed in the middle column** of the table, then to carry forward your customization you need to manually edit the setting in the new HyperStore version's configuration templates (under */etc/cloudian-<new-version-#>-puppet/modules*) before continuing. For example, if in your existing HyperStore system you had set a custom value for a *hyperstore-server.properties.erb* setting that's not listed in the table, edit that same setting in */etc/cloudian-7.1-puppet/modules/cloudians3/templates/hyperstore-server.properties.erb*. After saving your change (and

doing likewise for any of your other customizations to **.erb* settings that are not listed in the table), return to the terminal instance in which you are running the upgrade, and at the installer prompt continue with the upgrade.

.erb File	.erb File Setting Being Migrated	New Version of Setting in common.csv
hyperstore-server.properties.erb	disk.check.interval auto.repair.computedigest.run.number	hyperstore_disk_check_interval auto_repair_computedigest_run_number
mts-ui.properties.erb	storageuri.ssl.enabled grouplist.enabled login.grouplist.enabled bucket.tiering.aws.url bucket.tiering.google.url	cmc_storageuri_ssl_enabled cmc_grouplist_enabled cmc_login_grouplist_enabled cmc_bucket_tiering_aws_url cmc_bucket_tiering_google_url
mts.properties.erb	cloudian.s3.serverside.encryption.keylength phonehome_proxy_host phonehome_proxy_port phonehome.uri phonehome.bucket phonehome.accessKey phonehome.secretKey cassandra.tombstone_cleanup_threshold cassandra.tombstone_gcgrace phonehome_proxy_username phonehome_proxy_password bucketstats.enabled awsmmsproxy.host	cloudian_s3_aes256encryption_enabled phonehome_proxy_host phonehome_proxy_port phonehome.uri phonehome_bucket phonehome_access_key phonehome_secret_key cassandra_tombstone_cleanup_threshold cassandra_tombstone_gcgrace phonehome_proxy_username phonehome_proxy_password bucketstats_enabled awsmmsproxy_host
cassandra.yaml.erb	tombstone_warn_threshold tombstone_failure_threshold	cassandra_tombstone_warn_threshold cassandra_tombstone_failure_threshold
watchcronhost.erb	MINUTES_BEFORE_FAILOVER	watchcronhost_minutes_before_failover

Note Customizations that you may have made to the configuration file *common.csv* are handled differently. The installer detects such customizations and automatically applies the same customizations to the new version's *common.csv* file, without you having to do anything.

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.

Note The install script's "Advance Configuration Options" are covered in the "System Configuration" section of the HyperStore Administrator's Guide.

7.1. Installation Troubleshooting

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 Puppet master node (on which you're running the install script):

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

On each Puppet agent node (each node on which you're installing HyperStore):

- */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.

7.1.2. Debug Mode

Another potentially useful source of troubleshooting information is to run the installer in debug mode:

```
[root]# <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*).

7.1.3. Specific Issues

ISSUE: You encounter the following warnings:

```
Warning: Could not retrieve fact fqdn
Warning: Host is missing hostname and/or domain: cloudian-singlenode
```

Solution

As suggested by the warning messages, the domain part is missing for the host named "cloudian-singlenode". To resolve this edit the */etc/hosts* file or */etc/resolv.conf* file.

1. Edit the */etc/hosts* file and make sure the following entry exists:

   ```
   Ip-address   cloudian-singlenode.MyDomain.Com  cloudian-singlenode
   ```

   - *Ip-address* should be replaced with host's real IP address
   - *MyDomain.Com* should be replaced with your domain name of choice.

2. Edit the */etc/resolv.conf* file and make sure the following entry exists:

   ```
   Domain MyDomain.Com
   ```

   MyDomain.Com should be replaced with your domain name of choice.

Verify that the *facter fqdn* and *hostname –f* commands output ‘ cloudian-singlenode.MyDomain.Com’ to the console.

ISSUE: Puppet is unable to propagate configuration settings to the agent nodes, and in the *puppet_agent.log* and/or *puppet_server.log* you see errors indicating certificate problems or access failures.

Solution

Try going to the installer’s "Advanced Options" sub-menu and executing task [**h**] — "Remove Existing Puppet SSL Certificates". Then go back to the main menu and choose the appropriate action below, depending on what you were doing when you encounted the Puppet run failure:

- If you are doing the initial installation of your HyperStore cluster, choose "Install Cloudian HyperStore", then execute task "Install Packages and Configure Nodes [includes Run Puppet]".
- If you are performing post-installation configuration tasks, choose "Cluster Management", then execute task "Push Configuration Settings to the Cluster [Run Puppet]".

ISSUE: While working with the installation script, you get a console message indicating that Puppet access is locked.

Solution

The Puppet process can sometimes end up left in a "locked" state if a Puppet run is interrupted, such as by a Ctrl-<c> command or a host shutdown.

To unlock Puppet, go to the installer’s "Advanced Options" sub-menu and execute task [**j**] — "Remove Puppet Access Lock". Then go back to the main menu and choose the appropriate Puppet-running action below, depending on what you were doing when you encountered the Puppet lock error:

- If you are doing the initial installation of your HyperStore cluster, choose "Install Cloudian HyperStore", then execute task "Install Packages and Configure Nodes [includes Run Puppet]".
- If you are performing post-installation configuration tasks, choose "Cluster Management", then execute task "Push Configuration Settings to the Cluster [Run Puppet]".

7.2. HyperStore Listening Ports

The HyperStore system uses the listening ports specified in the table below. **These ports must be open on each host on which you run HyperStore.** Only the CMC Service and S3 Service (with listening ports marked in italics below) should be exposed to traffic that ultimately originates from end users outside the data center.

Service	Listening Port	Purpose
Cloudian Management Console (CMC)	*8888*	Requests from administrators' or end users' browsers via HTTP
	8443	Requests from administrators' or end users' browsers via HTTPS

Service	Listening Port	Purpose
S3 Service	*80*	Requests from the CMC or other S3 client applications via HTTP
	443	Requests from the CMC or other S3 client applications via HTTPS
	81	Requests relayed by an HAProxy load balancer using the PROXY Protocol (if enabled by configuration; see s3_proxy_protocol_enabled in common.csv)
	4431	Requests relayed by an HAProxy load balancer using the PROXY Protocol with SSL (if enabled by configuration)
	19080	JMX access
Admin Service	18081	Requests from the CMC or other Admin API clients via HTTP
	19443	Requests from the CMC or other Admin API clients via HTTPS (Note: The CMC by default uses HTTPS to access the Admin Service)
	19081	JMX access
IAM Service	16080	Requests from the CMC or other Admin API clients via HTTP
	16443	Requests from the CMC or other Admin API clients via HTTPS (Note: The CMC by default uses HTTPS to access the Admin Service)
	19084	JMX access
Redis Monitor	9078	Communication between primary and backup Redis Monitor instances
	19083	JMX access
HyperStore Service	19090	Data operation requests from the S3 Service
	19050	Communication between HyperStore Service instances
	19082	JMX access
Redis DBs	6379	Requests to the Redis Credentials DB from the S3 Service, HyperStore Service, or Admin Service; and communication between Redis Credentials instances
	6380	Requests to the Redis QoS DB from the S3 Service, HyperStore Service, or Admin Service; and communication between Redis QoS instances
Cassandra	9160	Data operations requests from the S3 Service, HyperStore Service, or Admin Service
	7000	Communication between Cassandra instances
	7199	JMX access
Cloudian Monitoring Agent	19070	Requests from the Cloudian Monitoring Data Collector

Service	Listening Port	Purpose
Puppet Master	8140	On your Puppet Master node (the HyperStore node from which you will manage cluster installation and configuration) this port will service incoming requests from Puppet agents on your other HyperStore nodes
SSH	22	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	NTP port for time synchronization between nodes
Echo	7	The Cloudian Monitoring Data Collector uses this port on each node to check whether the node is reachable, if ICMP is unavailable.
ICMP	n/a	The Cloudian Monitoring Data Collector uses ICMP to check whether each node is reachable.

IMPORTANT: HyperStore nodes sometimes communicate with each other via JMX, and when they do, after initial connection establishment on the designated JMX port a random port is used for continued communication. Therefore **there cannot be any port restrictions on internal communication between HyperStore nodes**.

7.2.1. 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).

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. 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 *mts.properties.erb: phonehome.proxy.**).
- **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.
- **Pre-Configured ntpd** — Accurate, synchronized time across the cluster is vital to HyperStore service. In 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 doesn’t support configuring an explicit forward proxy, but you can use transparent forward proxying if you wish.

To see which of your HyperStore nodes are internal NTP servers, after HyperStore installation log into the CMC and go to **Cluster → Cluster Config → Cluster Information**.

For more information on HyperStore's NTP set-up, see "System Configuration" -> "Configuration Special Topics" -> "NTP Automatic Set-Up" in the *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 33) for HyperStore requirements regarding listening port access.

7.4. File System Requirements

When you are installing a new HyperStore cluster you can use the *system_setup.sh* tool to configure the data disks and mount points on your nodes, as described in **"Configuring Data Disks"** (page 13). The tool is part of the HyperStore product package (the *.bin* file). **If you do not use the system setup tool for disk setup** -- such as would be the case if you're adding nodes to an existing HyperStore cluster -- review the information below to make sure that your host machines meet HyperStore requirements.

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 OS Requirements"** (page 9)). On host machines with multiple hard drives:

- HyperStore will by default use the drive that the OS is on for storing system metadata (in Cassandra and Redis databases). If a host machine has 10 or more drives in total, 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.
- 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. Format each of the 12 HDDs with *ext4* file systems and configure mount points such as */cloudian1*, */cloudian2*, */cloudian3* and so on.

HyperStore **does not support** XFS file systems; VirtIO disks; Logical Volume Manager (LVM); or Multipathing. For questions regarding these unsupported technologies, contact Cloudian Support:

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.

```
[root@hyperstore1 etc]# 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:

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

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

```
[root]# 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,bar-
rier=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:

```
[root]# mount -a
```

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

You Must Manually Create a Data Directory Mount Point List (fslist.txt)

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 Puppet Master node. For detail see **"Data Directory Mount Point List (fslist.txt)"** (page 39).

Mount Point Naming Guidelines

If you are installing HyperStore on multiple hosts that each have multiple disk drives, 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. For example, if each host has 12 disks for S3 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 S3 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 **"Data Directory Mount Point List (fslist.txt)"** (page 39)..

Option for Putting Cassandra Data on Dedicated Disks Rather Than the OS Disk

Regarding Cassandra data, another supported configuration — for a host with many drives — is to put your Cassandra data directory and Cassandra commit log directories each on dedicated disks, rather than on the OS disk. In this case you would have:

- OS drive (with Redis also)
- Cassandra data directory drive (mount point path **must** include */cassandra*)
- Cassandra commit log directory drive (mount point path **must** include */cassandra_commit*)
- Multiple drives for S3 object data (with mount points for example */cloudian1*, */cloudian2*, */cloudian3* and so on).

In this configuration, where Cassandra data is on a different disk than the OS, it’s advisable to use RAID-1 for the OS disk. It’s not necessary to use RAID for a dedicated Cassandra disk.

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 data (S3 object data).

For each HyperStore data drive do the following.

```
# Check current “Reserved block count”:

root# tune2fs -l <device>

# Set Reserved block count to 0%:

root# tune2fs -m 0 <device>

# For example:
```

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

7.5. Data Directory Mount Point List (fslist.txt)

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 Puppet Master node. 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.

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

Note The requirement to create an *fslist.txt* file does not apply if you doing a simple HyperStore evaluation by installing on a host or hosts that have just a single disk (on which will be stored the OS and application data as well as S3 object data). The requirement applies only to the typical case where hosts have multiple disks some of which will be dedicated to S3 object storage.

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 specify one mount point for metadata stored in Cassandra (must include the string "cassandra" in the mount point path) and/or one mount point for the Cassandra commit log (must include "cassandra_commit" in the mount point path). If you do not specify these Cassandra mount points in *fslist.txt*, by default the system automatically puts Cassandra data and commit log directories on the same disk on which the operating system and application files reside.

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*.

7.6. Cluster Survey File (survey.csv)

During the **"Installing HyperStore Prerequisites"** (page 11) task you use the *system_setup.sh* script to create a cluster survey file which by default is named *survey.csv*. This file resides in your installation staging directory for the life of your HyperStore system. The survey file is automatically updated by the system if you subsequently use the CMC to add more nodes to your cluster; and it is automatically copied to your new installation staging directory when you execute a HyperStore version upgrade.

> **Note** The survey file must be kept in the installation staging directory, not in a different directory. Do not delete or move the survey file.

The survey file contains one line for each HyperStore host in your cluster (including the Puppet Master host), with each line using the format below.

```
<regionname>,<hostname>,<ip4-address>,<datacenter-name>,<rack-name>[,<internal-inter-
face>]
```

- *<region-name>* — The HyperStore system supports having multiple service regions with each region having its own independent storage cluster and own independent S3 object inventory, and with S3 clients able to choose a storage region when they create storage buckets. Even if you will have only one region you must give it a name. The maximum allowed length is 52 characters. The only allowed character types are ASCII alphanumerical characters and dashes (A-Za-z0-9 and dashes). For more information on regions see "Service Regions Feature Overview" in the "Major Features" section of the *HyperStore Administrator's Guide*.
- *<hostname>* — Hostname of a host machine on which you are installing HyperStore software. Be sure to include in your survey file one line for each node on which you are installing HyperStore. You can install to all your intended nodes at the same time, even nodes in a different data center or region. In the hostnames you can use periods (such as in FQDNs), dashes, and underscores -- but not spaces or special characters.
- *<ip4-address>* — IP address (v4) that the hostname resolves to. Do not use IPv6. This should be the IP address associated with the host's default, external interface -- not an internal interface.
- *<datacenter-name>* — Name of the data center in which the host machine is located. The maximum allowed length is 256 characters. The only allowed character types are ASCII alphanumerical characters and dashes (A-Za-z0-9 and dashes). If there is no official name you can just create one for purposes of HyperStore configuration.
- *<rack-name>* — Name of the server rack in which the host machine is located. The maximum allowed length is 256 characters. The only allowed character types are ASCII alphanumerical characters and dashes (A-Za-z0-9 and dashes). If there is no official name you can just create one for purposes of HyperStore configuration.
- *[<internal-interface>]* — Use this field only for hosts that will use a different network interface for internal cluster traffic than the rest of the hosts in the cluster do. For example, if most of your hosts will use "eth1" for internal cluster traffic, but two of your hosts will use "eth2" instead, use this field to specify "eth2" for each of those two hosts, and leave this field empty for the rest of the hosts in your survey file. (Later in the installation procedure you will have the opportunity to specify the default internal interface for the hosts in your cluster -- the internal interface used by all hosts for which you do not specify the *internal-interface* field in your survey file.) If all of your hosts use the same internal network interface — for example if all hosts use "eth1" for internal network traffic — then leave this field empty for all hosts in the survey file.

Note: Cassandra, Redis, and the HyperStore Service are among the services that will utilize the internal interface for intra-cluster communications.

The example survey file below is for a single-node HyperStore installation:

```
region1,arcturus,10.10.2.1,DC1,RAC1
```

This second example survey file is for a three-node HyperStore cluster with just one service region, one data center, and one rack:

```
tokyo,cloudian-vm7,10.10.1.33,DC1,RAC1
tokyo,cloudian-vm8,10.10.1.34,DC1,RAC1
tokyo,cloudian-vm9,10.10.1.35,DC1,RAC1
```

This third example survey file below is for a HyperStore installation that spans two regions, with the first region comprising two data centers and the second region comprising just one data center. Two of the hosts use a different network interface for internal network traffic than all the other hosts do.

```
boston,hyperstore1,10.1.0.1,DC1,RAC1
boston,hyperstore2,10.1.0.2,DC1,RAC1
boston,hyperstore3,10.1.0.3,DC1,RAC1
boston,hyperstore4,10.2.0.1,DC2,RAC5
boston,hyperstore5,10.2.0.2,DC2,RAC5
chicago,hyperstore6,10.3.0.1,DC3,RAC2
chicago,hyperstore7,10.3.0.2,DC3,RAC2
chicago,hyperstore8,10.3.2.1,DC3,RAC7,eth2
chicago,hyperstore9,10.3.2.2,DC3,RAC7,eth2
```

7.7. cloudianInstall.sh Command Line Options

The HyperStore installation script *cloudianInstall.sh* resides in your installation staging directory on your Puppet Master node. Typically you would launch the script either like this:

```
[root]# ./cloudianInstall.sh -s survey.csv
```

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

```
[root]# ./cloudianInstall.sh -s survey.csv configure-dnsmasq
```

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

```
[root]# ./cloudianInstall.sh [-s <survey-filename>] [-c <config-filename>]
[-k <ssh-private-key-filename>] [-l <install-log-filename>] [-d] [-h]
[no-hosts] [configure-dnsmasq] [force]
```

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

After using the installer for product installation or subsequently for system configuration tasks, 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.

- *[-s <survey-filename>]* — Name of your cluster survey file. If you do not specify the survey filename argument, the script will prompt you for the file name during installation.
- *[-c <config-filename>]* — The first time you run *cloudianInstall.sh*, it creates in the installation staging directory a cluster configuration file *CloudianInstallationConfiguration.txt* based on your survey file input and your interactive input. If you are subsequently doing multiple installations for testing purposes, you can edit this file, rename it, and then use your customized version as input to *cloudianInstall.sh* runs by using the *-c <config-filename>* option (perhaps in combination with the *-b* option).
- *[-k <ssh-private-key-filename>]* — The Puppet master employs SSH for secure communication with the rest of your HyperStore installation nodes. Use the *-k <ssh-private-key-filename>* option if you want to use your own existing SSH authentication key pair rather than having the HyperStore install tool generate a key pair for you. If you want to use this option, then before running the *cloudianInstall.sh* script:
    - Copy your private key into the installation staging directory (where the install script resides).
    - Copy your public key to each host on which you plan to install HyperStore (see standard SSH set-up documentation for guidance); or copy your public key to the installation staging directory and the install script will copy the public key to your target nodes automatically.
- *[-l <install-log-filename>]* — By default installation logging information is written to *cloudian-installation.log* in the staging directory. You can use this option if you'd prefer that installation logging information be written to a different file. Include the path if you want the file location to be other than in the staging directory.
- *[-d]* — Turn on debugging output.
- *[-h]* — Display Help information for the tool. This option causes the tool to print a usage message and exit.
- *[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.
- *[configure-dnsmasq]* — Use this option if you want the install tool to install and configure **dnsmasq**, a lightweight utility that can provide domain resolution services for testing a small HyperStore system. If you use this option the installer installs *dnsmasq* and automatically configures it for resolution of HyperStore service domains. If you did not create DNS entries for HyperStore service domains as described in **"DNS Set-Up"** (page 4), then you must use the *configure-dnsmasq* option in order for the system to be functional when you complete installation. Note that using *dnsmasq* is not appropriate in a production environment.

    > **Note:** If you do not have the installer install *dnsmasq* during HyperStore installation, and then later you decide that you do want to use *dnsmasq* for your already installed and running HyperStore system, do not use the *configure-dnsmasq* command line option when you re-launch the installer. Instead, re-launch the installer with no options and use the Installer Advanced Configuration Options menu to enable *dnsmasq* for your system. (See the System Configuration chapter of the HyperStore Administrator's Guide for instructions.)

- *[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 16GB
- OS is RHEL or CentOS 7.0 or newer
- System Architecture is x86 64-bit
- SELinux is disabled
- firewalld is disabled
- iptables is not running
- Directory permissions are as needed
- Hostnames of installation hosts are resolvable or installer is allowed to append to hosts' */etc/hosts* files
- S3 service and Admin service endpoints (which you will specify during the interactive install) are resolvable -- which they should be so long as you either create DNS entries for HyperStore service endpoints or else use the *configure-dnsmasq* option when launching the installer