Dashboard / Security Deployment RunBook

Installing Transparent Data Encryption

Created by Alexandru Ciobanu, last modified on 05/01/2017

eTo configure TDE significant preparation of the servers is required before installation can begin. We will first need to install Cloudera Navigator Key Trustee Server before anything else can be done. It is important that the machines housing Key Trustee Server do not run any Hadoop services as that might affect our ability to security harden the machines. Full instructions available here

Cloudera Navigator Key Trustee Server and Key Trustee KMS requires a certain amount of entropy to be available on the system. We can use these instructions to help bolster the randomness of the systems while will house Cloudera Key Trustee Server and Key Trustee KMS.

To determine the amount of randomness in the system use the command

```
cat /proc/sys/kernel/random/entropy_avail
```

If a value of less than 500 is found, try to install rng-tools

```
yum -y install rng-tools
echo 'EXTRAOPTIONS="-r /dev/urandom"' >> /etc/sysconfig/rngd
systemctl start rngd.service
systemctl enable rngd.service
```

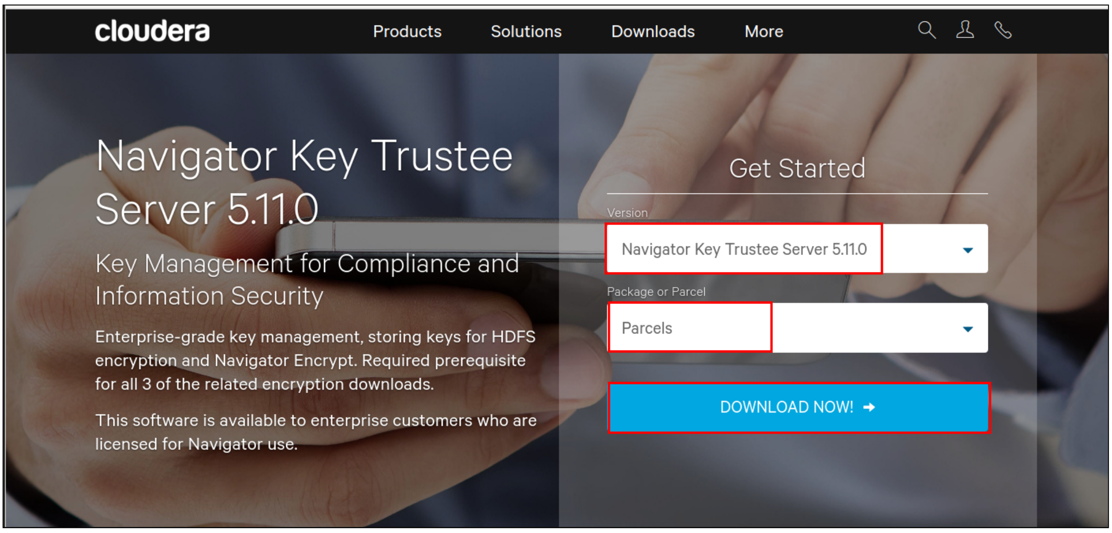
Next we will need to install a private parcels repository. If there is already an internal parcels repository you can skip this task. This is required as the Cloudera Key Trustee Server is not available through any public repository. It must be downloaded from the Cloudera Website via password authentication then deployed on an internal trusted parcels repository.

Navigate to the URL

```
https://www.cloudera.com/downloads/navigator/key-trustee-server/5-11-0.html
```

Download the Navigator Key Trustee Server 5.11 parcels.

IMPORTANT: The KEYTRUSTEE parcel in Cloudera Manager is not the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE_SERVER.



Navigator Key Trustee Server 5.11.0

Key Management for Compliance and Information Security

Enterprise-grade key management, storing keys for HDFS encryption and Navigator Encrypt. Required prerequisite for all 3 of the related encryption downloads.

This software is available to enterprise customers who are licensed for Navigator use.

Get Started

Version: Navigator Key Trustee Server 5.11.0

Package or Parcel: Parcels

Next copy the parcel on the edge node where you will install the parcel repository

```
scp keytrustee-server-5.11.0-parcels.tar.gz root@edge1:
```

Next we will need to install httpd to enable an internal parcels repository. Do this on any edge node of the cluster or on a separate server.

```
ssh root@edge1
yum -y install httpd
systemctl start httpd
```

After that you can untar the downloaded artifact and move the results into the read directory of the httpd server. Sample command would be:

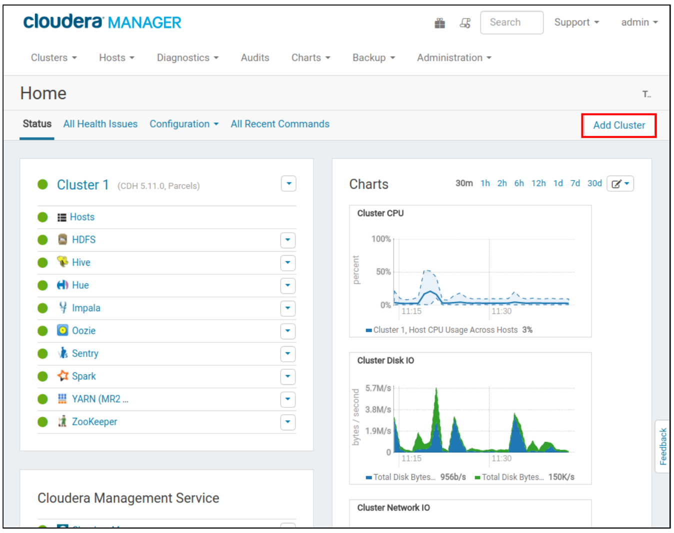
```
tar -zxvf keytrustee-server-5.11.0-parcels.tar.gz
mv keytrustee-server-5.11.0-parcels /var/www/html/keytrustee
chmod -R ugo+rX /var/www/html/keytrustee
```

We can now go to Cloudera Manager and install Key Trustee Server

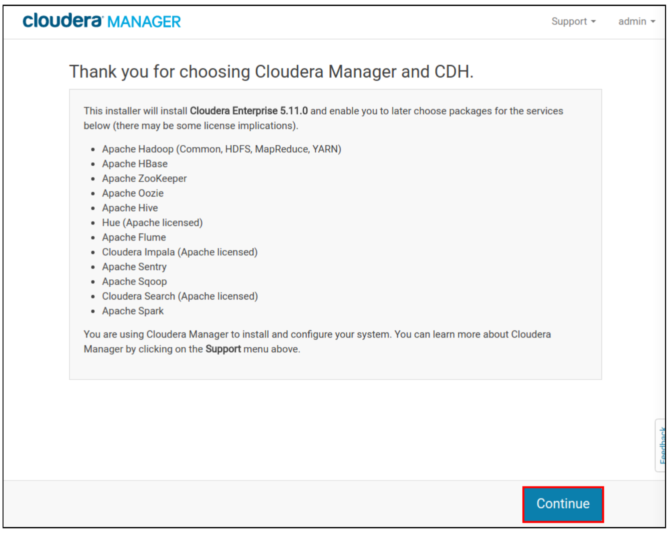
First we need to create a new cluster for the nodes housing Key Trustee Server.

Open a web browser and go to the home page of Cloudera Manager.

On the home page of Cloudera manager click on **Add Cluster**.



On the first introduction page click **Continue**



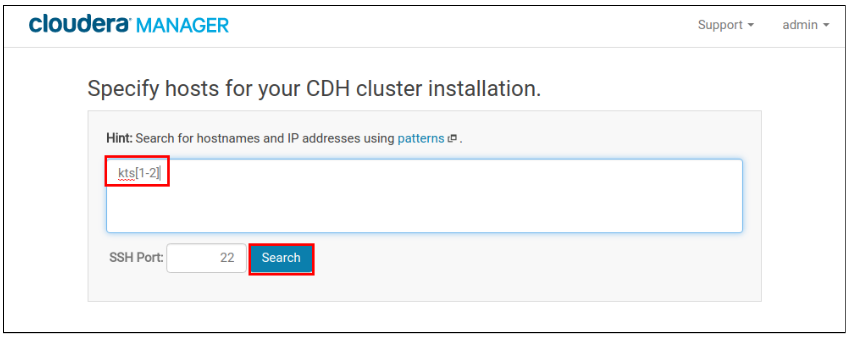
Thank you for choosing Cloudera Manager and CDH.

This installer will install **Cloudera Enterprise 5.11.0** and enable you to later choose packages for the services below (there may be some license implications).

- Apache Hadoop (Common, HDFS, MapReduce, YARN)
- Apache HBase
- Apache ZooKeeper
- Apache Oozie
- Apache Hive
- Hue (Apache licensed)
- Apache Flume
- Cloudera Impala (Apache licensed)
- Apache Sentry
- Apache Sqoop
- Cloudera Search (Apache licensed)
- Apache Spark

You are using Cloudera Manager to install and configure your system. You can learn more about Cloudera Manager by clicking on the **Support** menu above.

Next Enter the regular expression need to find the servers which are to house key trustee, then click **Search**



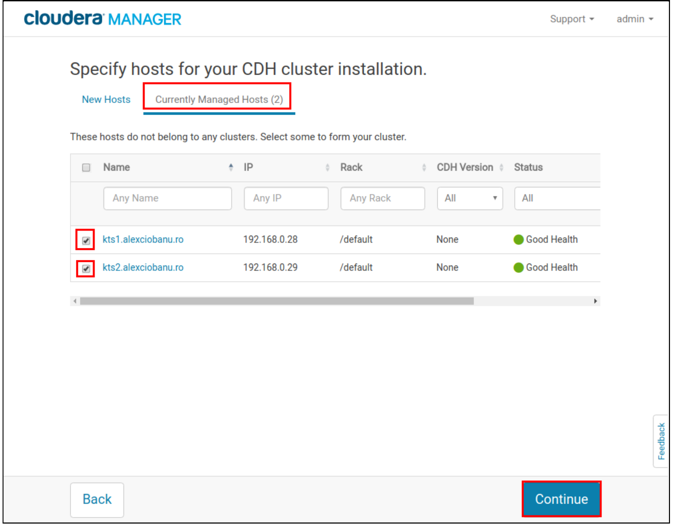
Specify hosts for your CDH cluster installation.

Hint: Search for hostnames and IP addresses using patterns.

kts[1-2]

SSH Port: 22

If you have previously added these hosts to the cluster, go the second tab **Currently Managed Hosts** and select the hosts from there. Them and click **Continue**

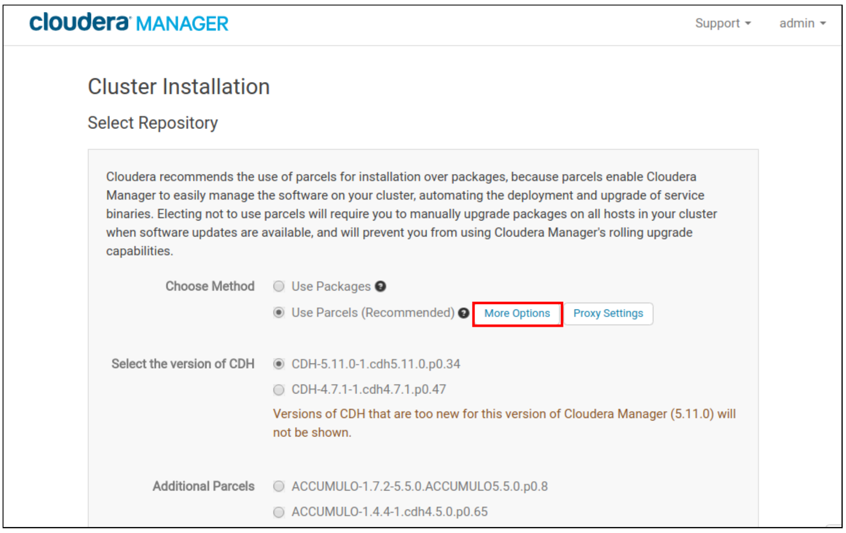


Specify hosts for your CDH cluster installation.

These hosts do not belong to any clusters. Select some to form your cluster.

□	Name	IP	Rack	CDH Version	Status
	Any Name	Any IP	Any Rack	All	All
☑	kts1.alexciobanu.ro	192.168.0.28	/default	None	● Good Health
☑	kts2.alexciobanu.ro	192.168.0.29	/default	None	● Good Health

When asked to choose the parcels click on **More Options** to allow us to enter out internal repository.



Cluster Installation

Select Repository

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Choose Method ○ Use Packages

● Use Parcels (Recommended)

Select the version of CDH ● CDH-5.11.0-1.cdh5.11.0.p0.34

○ CDH-4.7.1-1.cdh4.7.1.p0.47

Versions of CDH that are too new for this version of Cloudera Manager (5.11.0) will not be shown.

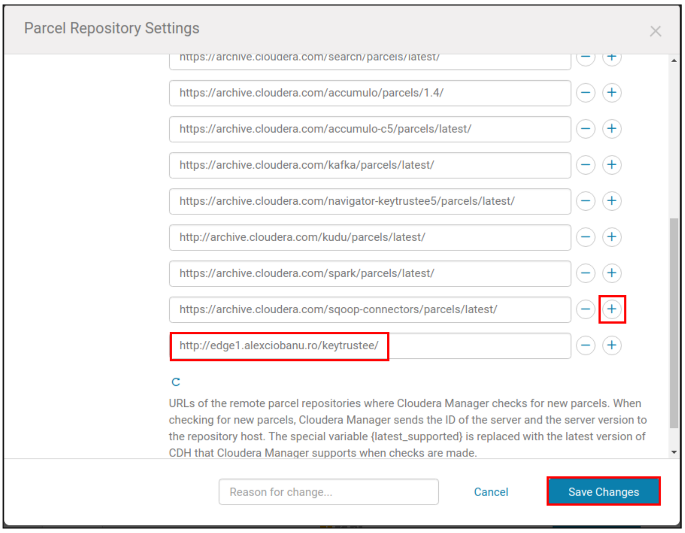
Additional Parcels ○ ACCUMULO-1.7.2-5.5.0.ACCUMULO5.5.0.p0.8

○ ACCUMULO-1.4.4-1.cdh4.5.0.p0.65

In the window that pops up in the section **Remote Parcel Repository URLs** enter the URL of the internal parcels server. You might need to scroll down to find this section

Remote Parcel Repository URLs: http://edge1.alexciobanu.ro/keytrustee/5.11.0/

Then click **Save Changes**


Parcel Repository Settings

https://archive.cloudera.com/search/parcels/latest/

https://archive.cloudera.com/accumulo/parcels/1.4/

https://archive.cloudera.com/accumulo-c5/parcels/latest/

https://archive.cloudera.com/kafka/parcels/latest/

https://archive.cloudera.com/navigator-keytrustee5/parcels/latest/

http://archive.cloudera.com/kudu/parcels/latest/

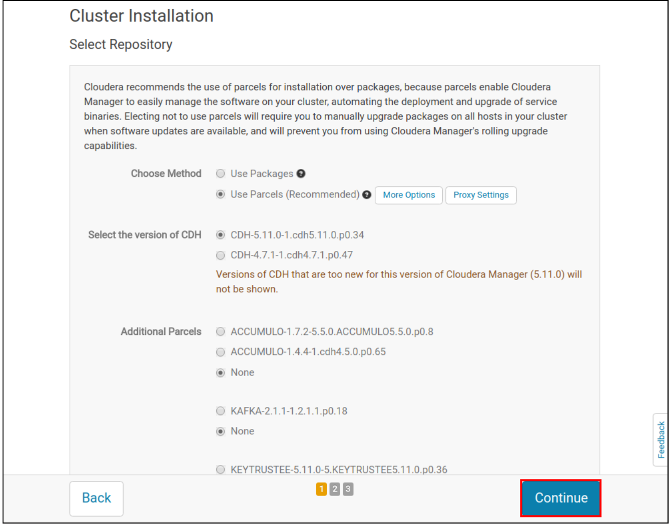
https://archive.cloudera.com/spark/parcels/latest/

https://archive.cloudera.com/sqoop-connectors/parcels/latest/

http://edge1.alexciobanu.ro/keytrustee/

URLs of the remote parcel repositories where Cloudera Manager checks for new parcels. When checking for new parcels, Cloudera Manager sends the ID of the server and the server version to the repository host. The special variable {latest_supported} is replaced with the latest version of CDH that Cloudera Manager supports when checks are made.

Now we can continue with a regular Installation (we will use that repository at a later time). Click on **Continue**



Cluster Installation

Select Repository

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Choose Method ○ Use Packages
● Use Parcels (Recommended) More Options Proxy Settings

Select the version of CDH ● CDH-5.11.0-1.cdh5.11.0.p0.34
○ CDH-4.7.1-1.cdh4.7.1.p0.47

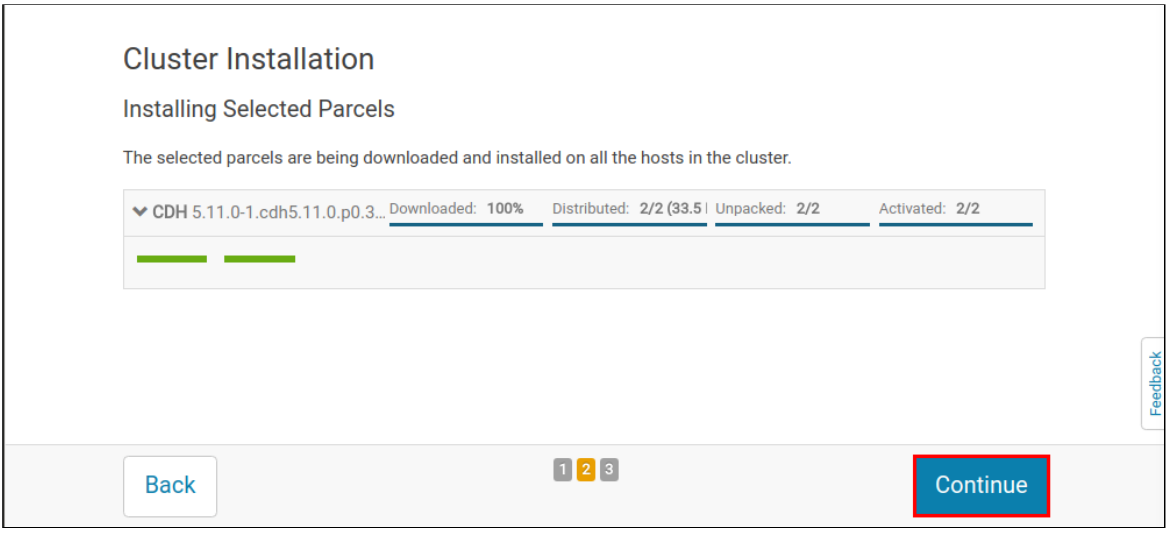
Versions of CDH that are too new for this version of Cloudera Manager (5.11.0) will not be shown.

Additional Parcels ○ ACCUMULO-1.7.2-5.5.0.ACCUMULO5.5.0.p0.8
○ ACCUMULO-1.4.4-1.cdh4.5.0.p0.65
● None

○ KAFKA-2.1.1-1.2.1.1.p0.18
● None

○ KEYTRUSTEE-5.11.0-5.KEYTRUSTEE5.11.0.p0.36

The CDH parcel will be deployed on the hosts. This will initialise our new cluster. Wait for the deployment to finish then click **continue.**



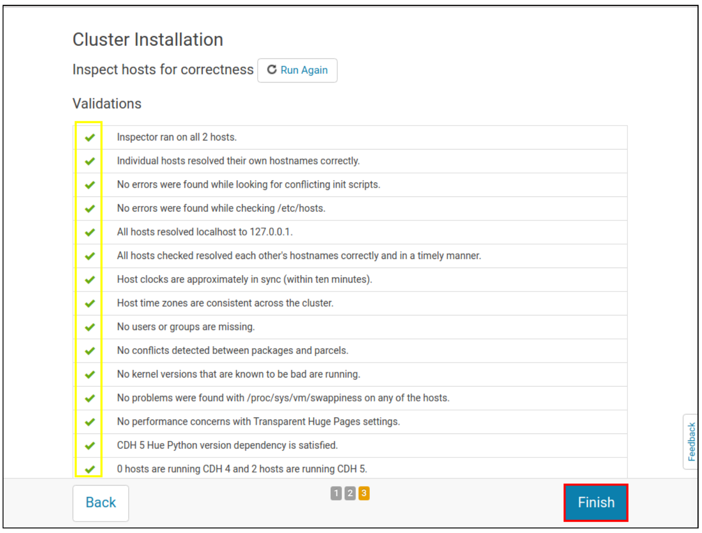
Cluster Installation

Installing Selected Parcels

The selected parcels are being downloaded and installed on all the hosts in the cluster.

CDH 5.11.0-1.cdh5.11.0.p0.3... Downloaded: 100% Distributed: 2/2 (33.5 | Unpacked: 2/2 Activated: 2/2

Host inspector will run to ensure the hosts are healthy. Once the inspector is done click **Finish**



Cluster Installation

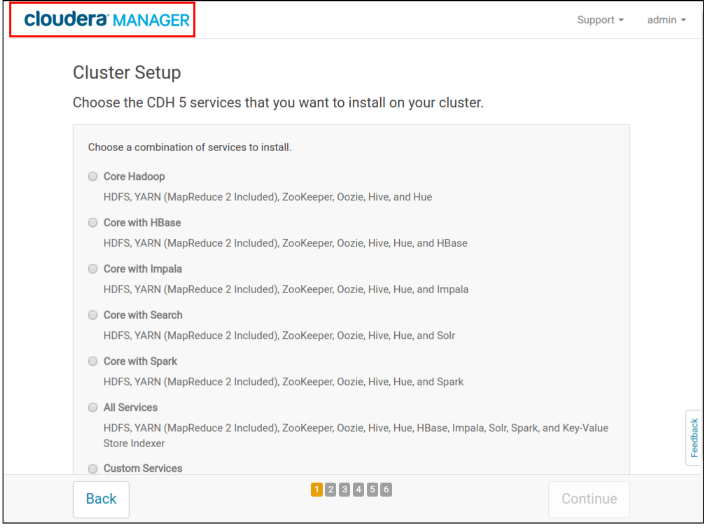
Inspect hosts for correctness Run Again

Validations

✓ Inspector ran on all 2 hosts.
✓ Individual hosts resolved their own hostnames correctly.
✓ No errors were found while looking for conflicting init scripts.
✓ No errors were found while checking /etc/hosts.
✓ All hosts resolved localhost to 127.0.0.1.
✓ All hosts checked resolved each other's hostnames correctly and in a timely manner.
✓ Host clocks are approximately in sync (within ten minutes).
✓ Host time zones are consistent across the cluster.
✓ No users or groups are missing.
✓ No conflicts detected between packages and parcels.
✓ No kernel versions that are known to be bad are running.
✓ No problems were found with /proc/sys/vm/swappiness on any of the hosts.
✓ No performance concerns with Transparent Huge Pages settings.
✓ CDH 5 Hue Python version dependency is satisfied.
✓ 0 hosts are running CDH 4 and 2 hosts are running CDH 5.

The next step will require the deployment of CDH components on the cluster. This step is not necessary as we will only deploy Keytrustee components on these nodes.

You can simply exit the menu by clicking on the **Cloudera Manager** logon on the right upper part of the screen.



Cluster Setup

Choose the CDH 5 services that you want to install on your cluster.

Choose a combination of services to install.

○ Core Hadoop

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, and Hue

○ Core with HBase

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and HBase

○ Core with Impala

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Impala

○ Core with Search

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Solr

○ Core with Spark

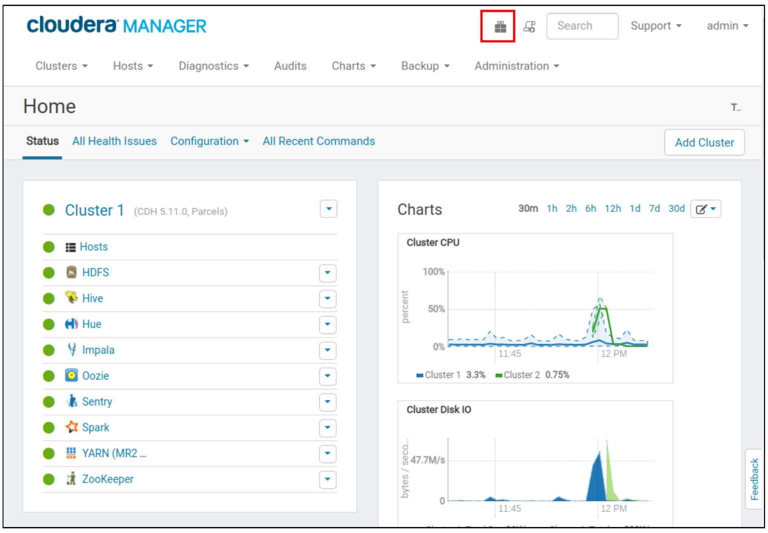
HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Spark

○ All Services

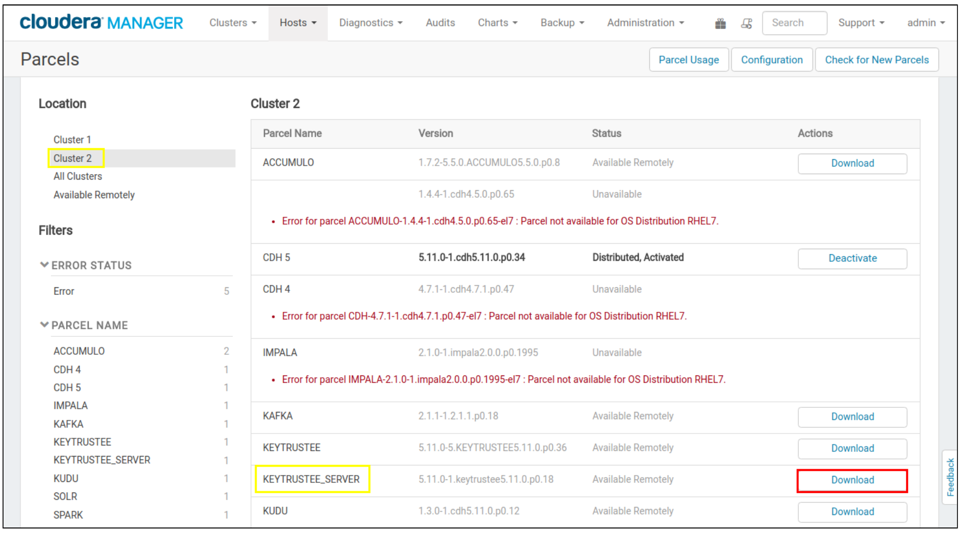
HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer

○ Custom Services

We can now deploy the KEYTRUSTEE_SERVER parcels to the newly deployed cluster. Click the parcels icon in the upper part of the screen.



In the Parcels window, make sure you select the correct cluster (**Cluster 2**, the newly created cluster) and click **download** next to the KEYTRUSTEE_SERVER parcels.



Location

Cluster 1
Cluster 2
All Clusters
Available Remotely

Filters

ERROR STATUS

Error	5

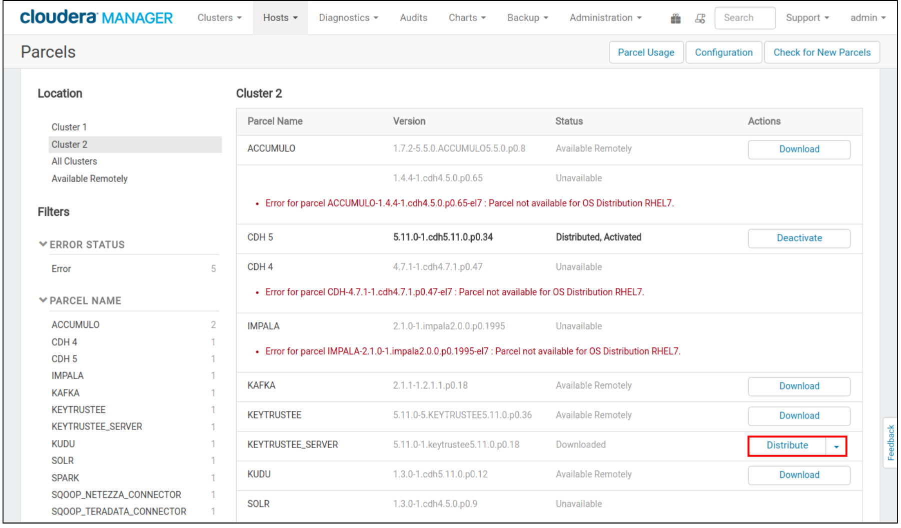
PARCEL NAME

ACCUMULO	2
CDH 4	1
CDH 5	1
IMPALA	1
KAFKA	1
KEYTRUSTEE	1
KEYTRUSTEE_SERVER	1
KUDU	1
SOLR	1
SPARK	1

Cluster 2

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	**5.11.0-1.cdh5.11.0.p0.34**	**Distributed, Activated**	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Available Remotely	Download
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Available Remotely	Download
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download

Once the parcels are downloaded, click the **Distribute** button.

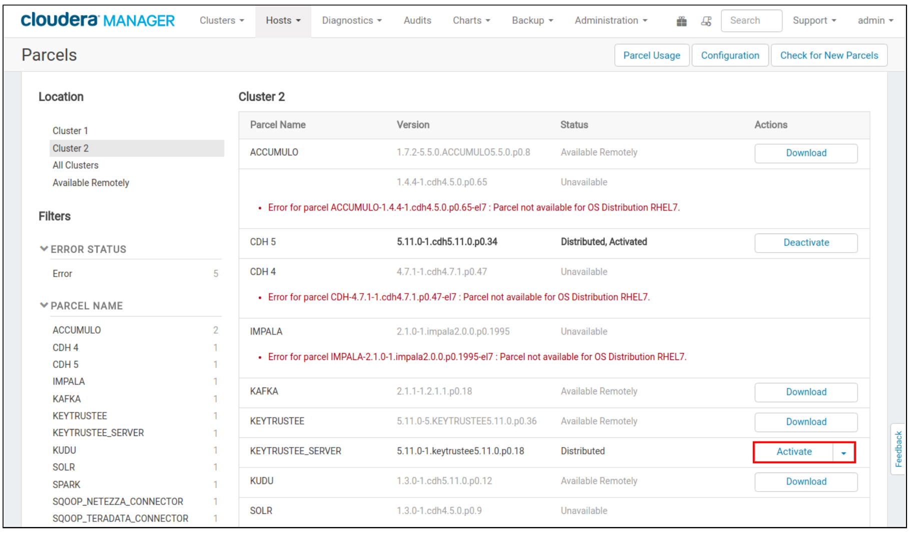


Parcels

Cluster 2

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Available Remotely	Download
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Downloaded	Distribute
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

And finally **activate** the Parcel.

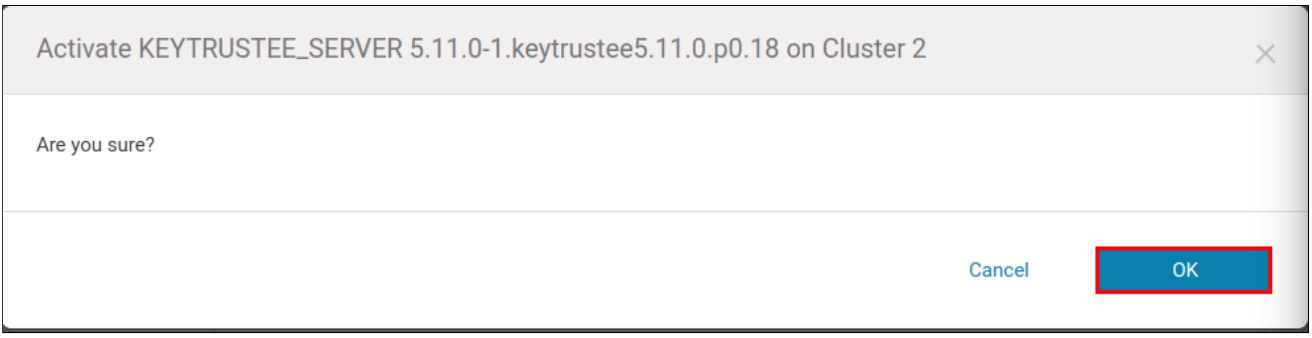


Parcels

Cluster 2

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Available Remotely	Download
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Distributed	Activate
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

And click **OK** on the confirmation dialog.

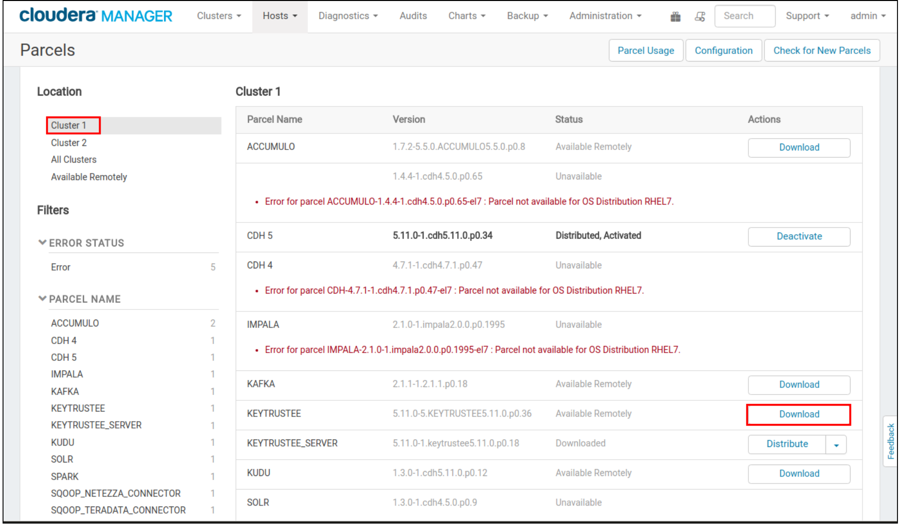


Activate KEYTRUSTEE_SERVER 5.11.0-1.keytrustee5.11.0.p0.18 on Cluster 2

Are you sure?

Switch to cluster 1 and repeat the process for the KEYTRUSTEE parcel.

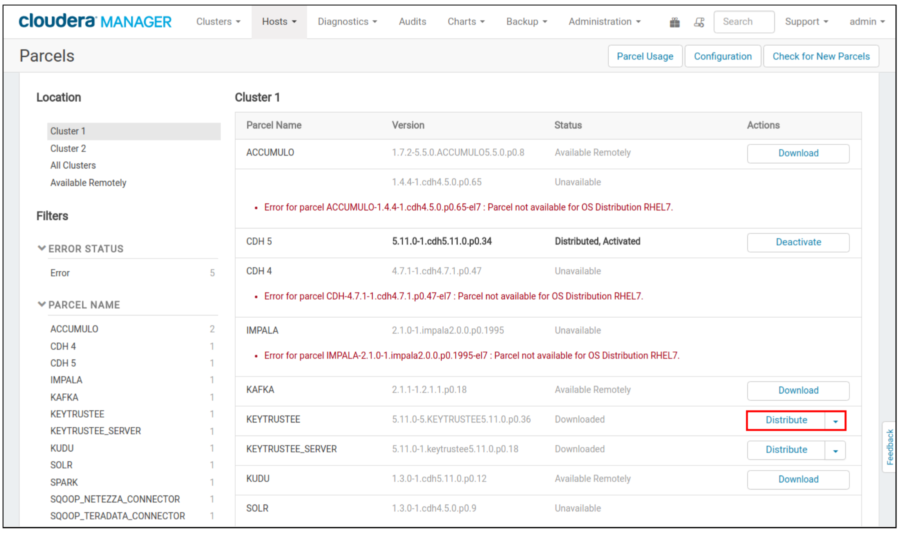
Select **Cluster 1** and click **Download** for the KEYTRUSTEE parcel



Cluster 1

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Available Remotely	Download
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Downloaded	Distribute
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

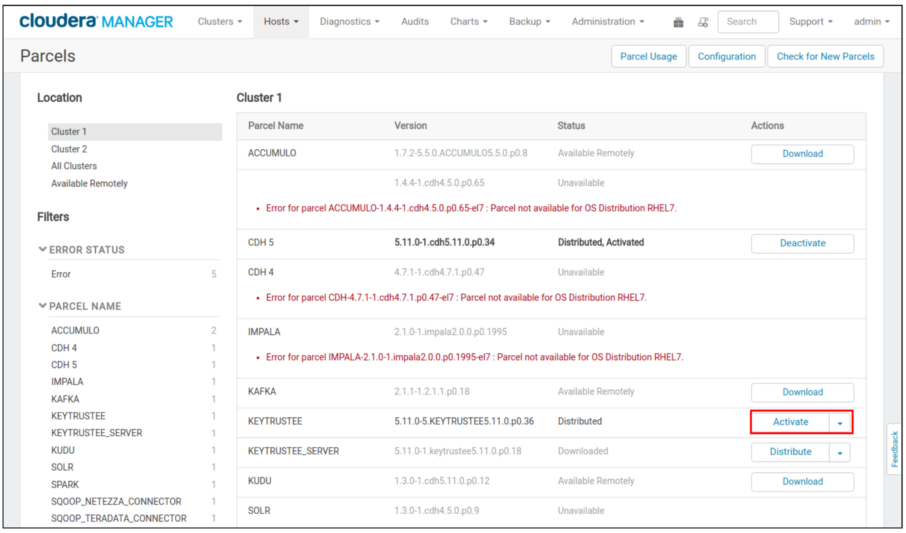
Next click on **Distribute**



Cluster 1

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Downloaded	Distribute
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Downloaded	Distribute
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

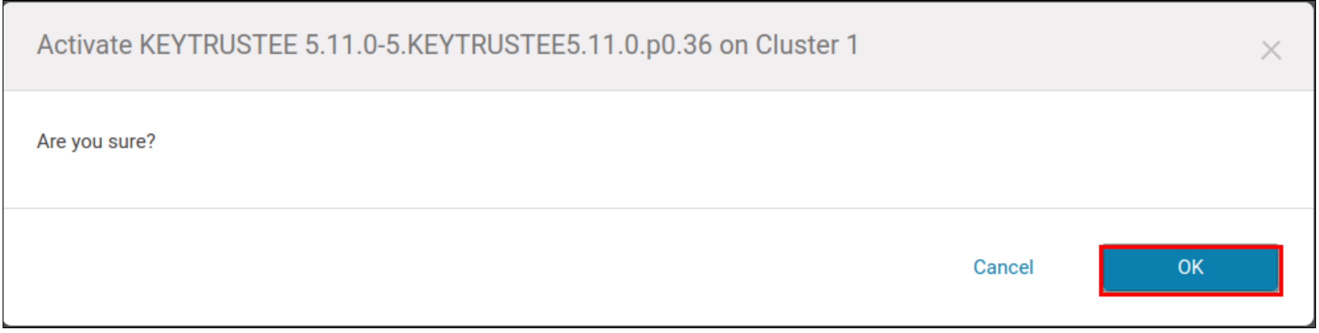
Finally we click **activate** for the keytrustee parcel



Cluster 1

Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
	1.4.4-1.cdh4.5.0.p0.65	Unavailable	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Distributed	Activate
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Downloaded	Distribute
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

On the pop up question click **OK**

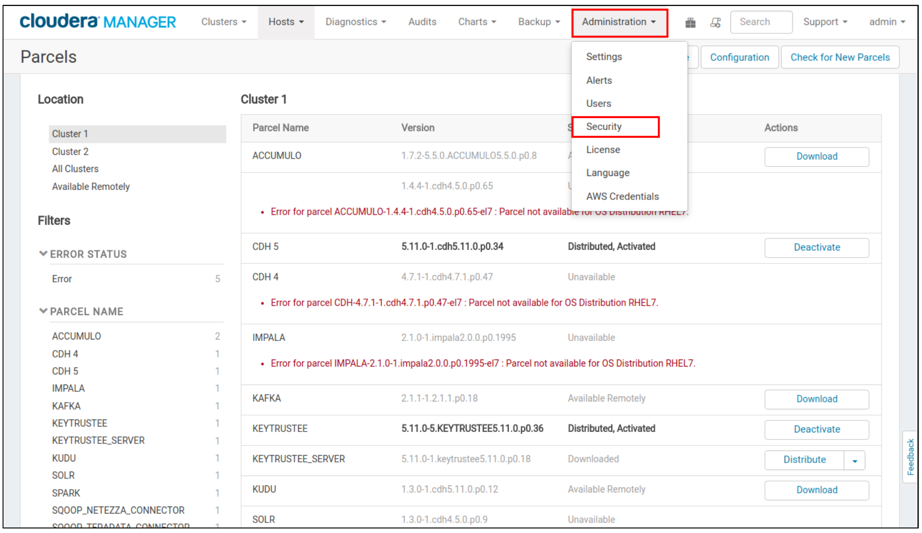


Activate KEYTRUSTEE 5.11.0-5.KEYTRUSTEE5.11.0.p0.36 on Cluster 1

Are you sure?

Once the parcel is activated, we can now start the Data At Rest Wizard.

At the top of the screen click **Administration -> Security**



Parcels

Location

- Cluster 1
- Cluster 2
- All Clusters
- Available Remotely

Filters

ERROR STATUS

Error	5

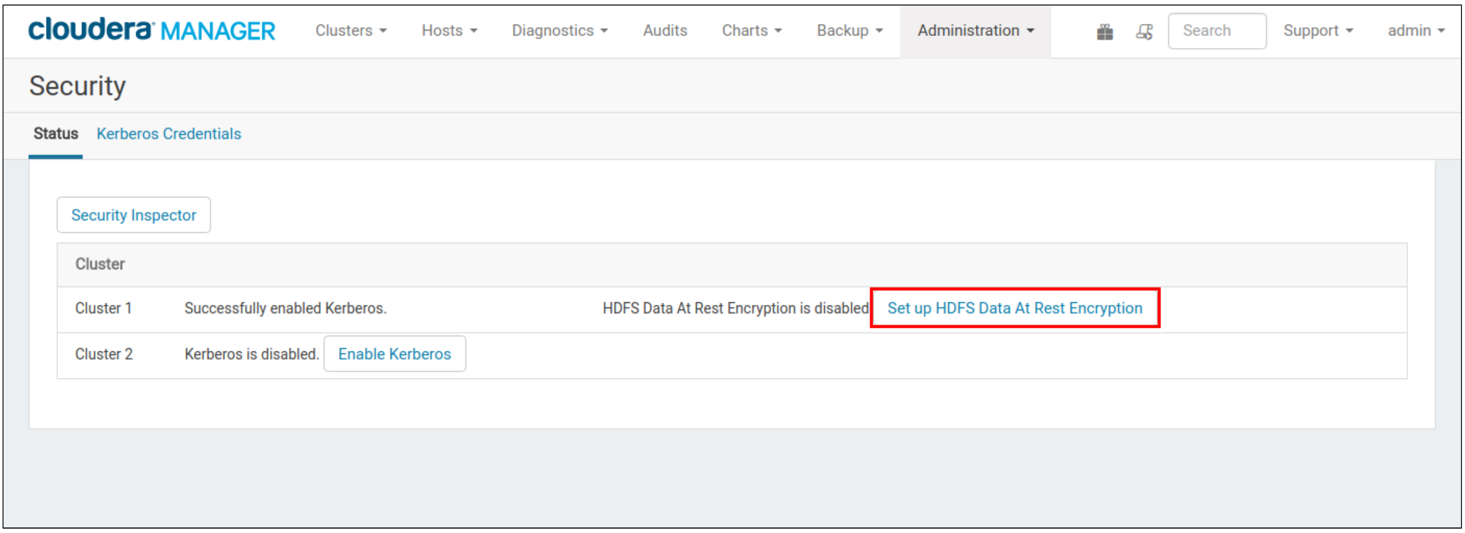
PARCEL NAME

ACCUMULO	2
CDH 4	1
CDH 5	1
IMPALA	1
KAFKA	1
KEYTRUSTEE	1
KEYTRUSTEE_SERVER	1
KUDU	1
SOLR	1
SPARK	1
SQOOP_NETEZZA_CONNECTOR	1
SQOOP_TERADATA_CONNECTOR	1

Cluster 1

Parcel Name	Version	S	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	A	Download
	1.4.4-1.cdh4.5.0.p0.65	U	
• Error for parcel ACCUMULO-1.4.4-1.cdh4.5.0.p0.65-el7 : Parcel not available for OS Distribution RHEL7.			
CDH 5	5.11.0-1.cdh5.11.0.p0.34	Distributed, Activated	Deactivate
CDH 4	4.7.1-1.cdh4.7.1.p0.47	Unavailable	
• Error for parcel CDH-4.7.1-1.cdh4.7.1.p0.47-el7 : Parcel not available for OS Distribution RHEL7.			
IMPALA	2.1.0-1.impala2.0.0.p0.1995	Unavailable	
• Error for parcel IMPALA-2.1.0-1.impala2.0.0.p0.1995-el7 : Parcel not available for OS Distribution RHEL7.			
KAFKA	2.1.1-1.2.1.1.p0.18	Available Remotely	Download
KEYTRUSTEE	5.11.0-5.KEYTRUSTEE5.11.0.p0.36	Distributed, Activated	Deactivate
KEYTRUSTEE_SERVER	5.11.0-1.keytrustee5.11.0.p0.18	Downloaded	Distribute
KUDU	1.3.0-1.cdh5.11.0.p0.12	Available Remotely	Download
SOLR	1.3.0-1.cdh4.5.0.p0.9	Unavailable	

On this page, click the button for **Set up HDFS Data At Rest Encryption**

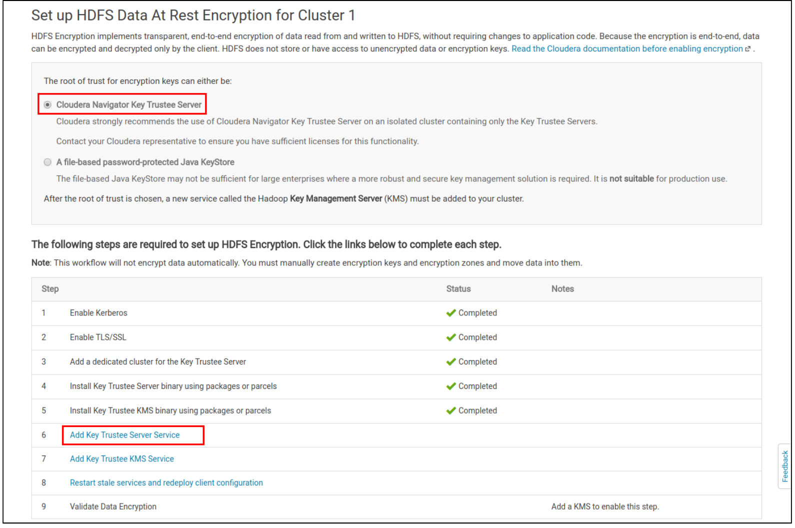


Security

Cluster		
Cluster 1	Successfully enabled Kerberos.	HDFS Data At Rest Encryption is disabled
Cluster 2	Kerberos is disabled.	

In the wizard page ensure **Cloudera Navigator Key Trustee Server** is selected in the root of trust for encryption keys section.

Next click on the **Add Key Trustee Server Service** link to perform said action



Set up HDFS Data At Rest Encryption for Cluster 1

HDFS Encryption implements transparent, end-to-end encryption of data read from and written to HDFS, without requiring changes to application code. Because the encryption is end-to-end, data can be encrypted and decrypted only by the client. HDFS does not store or have access to unencrypted data or encryption keys. Read the Cloudera documentation before enabling encryption .

The root of trust for encryption keys can either be:

● Cloudera Navigator Key Trustee Server

Cloudera strongly recommends the use of Cloudera Navigator Key Trustee Server on an isolated cluster containing only the Key Trustee Servers.

Contact your Cloudera representative to ensure you have sufficient licenses for this functionality.

○ A file-based password-protected Java KeyStore

The file-based Java KeyStore may not be sufficient for large enterprises where a more robust and secure key management solution is required. It is **not suitable** for production use.

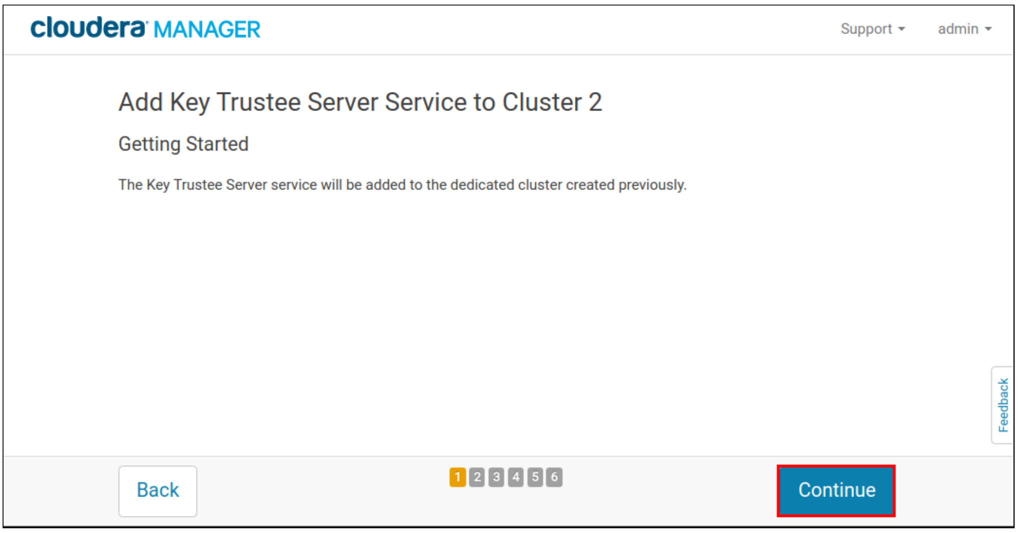
After the root of trust is chosen, a new service called the Hadoop **Key Management Server** (KMS) must be added to your cluster.

The following steps are required to set up HDFS Encryption. Click the links below to complete each step.

Note: This workflow will not encrypt data automatically. You must manually create encryption keys and encryption zones and move data into them.

Step		Status	Notes
1	Enable Kerberos	✓ Completed	
2	Enable TLS/SSL	✓ Completed	
3	Add a dedicated cluster for the Key Trustee Server	✓ Completed	
4	Install Key Trustee Server binary using packages or parcels	✓ Completed	
5	Install Key Trustee KMS binary using packages or parcels	✓ Completed	
6	Add Key Trustee Server Service		
7	Add Key Trustee KMS Service		
8	Restart stale services and redeploy client configuration		
9	Validate Data Encryption		Add a KMS to enable this step.

We will now enter the Key Trustee Server installation wizard. On the first window there is nothing to select so simply click **Continue**

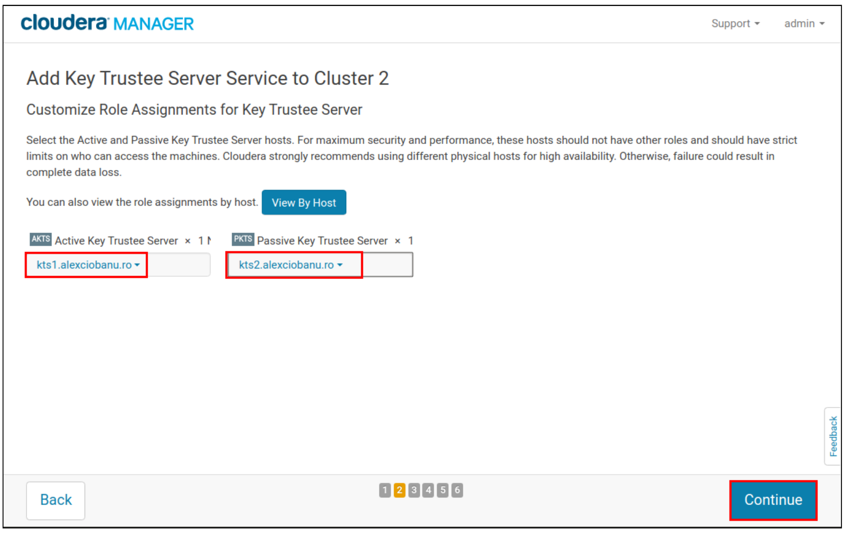


Add Key Trustee Server Service to Cluster 2

Getting Started

The Key Trustee Server service will be added to the dedicated cluster created previously.

In the next window you will need to select which servers will host Key Trustee Server. Select the first server to be the **master** and the second server to be the **replica**, then click **Continue**



Add Key Trustee Server Service to Cluster 2

Customize Role Assignments for Key Trustee Server

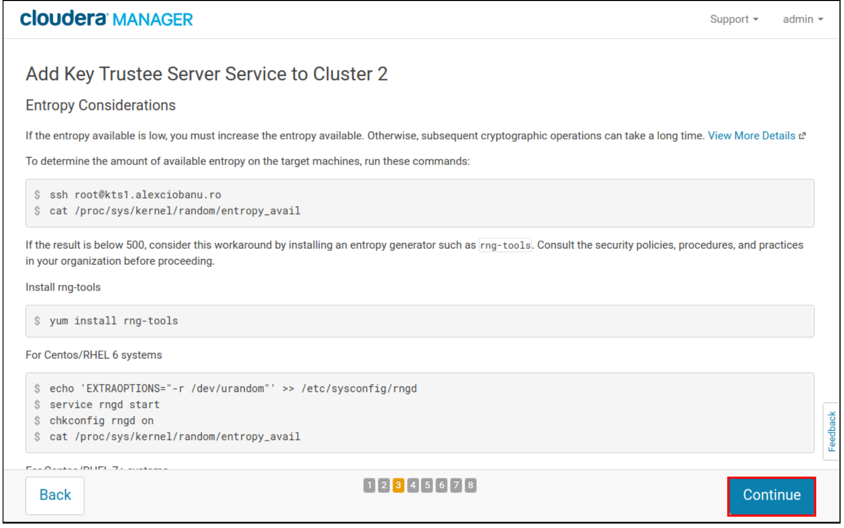
Select the Active and Passive Key Trustee Server hosts. For maximum security and performance, these hosts should not have other roles and should have strict limits on who can access the machines. Cloudera strongly recommends using different physical hosts for high availability. Otherwise, failure could result in complete data loss.

You can also view the role assignments by host. View By Host

AKTS Active Key Trustee Server × 1 N: kts1.alexciobanu.ro

PKTS Passive Key Trustee Server × 1: kts2.alexciobanu.ro

Next you will get instructions on how to increase the entropy of the system. We have already performed these instructions as a preconfigs step. We can simple click **Continue**



Add Key Trustee Server Service to Cluster 2

Entropy Considerations

If the entropy available is low, you must increase the entropy available. Otherwise, subsequent cryptographic operations can take a long time. View More Details

To determine the amount of available entropy on the target machines, run these commands:

```
$ ssh root@kts1.alexciobanu.ro
$ cat /proc/sys/kernel/random/entropy_avail
```

If the result is below 500, consider this workaround by installing an entropy generator such as `rng-tools`. Consult the security policies, procedures, and practices in your organization before proceeding.

Install rng-tools

```
$ yum install rng-tools
```

For Centos/RHEL 6 systems

```
$ echo 'EXTRAOPTIONS="-r /dev/urandom"' >> /etc/sysconfig/rngd
$ service rngd start
$ chkconfig rngd on
$ cat /proc/sys/kernel/random/entropy_avail
```

On the next window we get instructions on initialising the Ket Trustee servers against the same private keys. We need to follow these instructions.

NOTE: if rsync in not installed on the server run the commands, on both kts servers:

```
yum install rsync
```

Enter the commands as seen on the webpage. For example, on the system used to write this document, the instructions were:

```
ssh root@kts1.alexciobanu.ro
ktadmin init
```

Output should look similar to the one below

```
[root@kts1 user]# ktadmin init
INFO:keytrustee.server.util:Creating self-signed cert
INFO:keytrustee.util:`/bin/openssl req -nodes -new -days 3650 -subj /C=US/ST=TX/L=Austin/CN=kts1.alexciobanu.ro/E=keytrustee@kts1.alexciobanu.r
INFO:keytrustee.server.util:Generating PGP key, this may take a while
Initialized directory for 4096R/EB9C63EFCD294264E79BDF31F0788AC403074A55
```

After the GPG key is created transfer the private key across to the other Key Trustee Server. A more secure method then rsync is recommended if the channel is not secure.

```
rsync -zav --exclude .ssl /var/lib/keytrustee/.keytrustee kts2.alexciobanu.ro:/var/lib/keytrustee/
```

The output should look similar to that below

```
[root@kts1 user]# rsync -zav --exclude .ssl /var/lib/keytrustee/.keytrustee kts2.alexciobanu.ro:/var/lib/keytrustee/
Password:
sending incremental file list
.keytrustee/
.keytrustee/gpg.conf
.keytrustee/keytrustee.conf
.keytrustee/logging.conf
.keytrustee/pubring.gpg
.keytrustee/pubring.gpg~
.keytrustee/random_seed
.keytrustee/secring.gpg
.keytrustee/trustdb.gpg

sent 11169 bytes  received 168 bytes  839.78 bytes/sec
total size is 12239  speedup is 1.08
```

And finally initialise the Passive Key Trustee Server

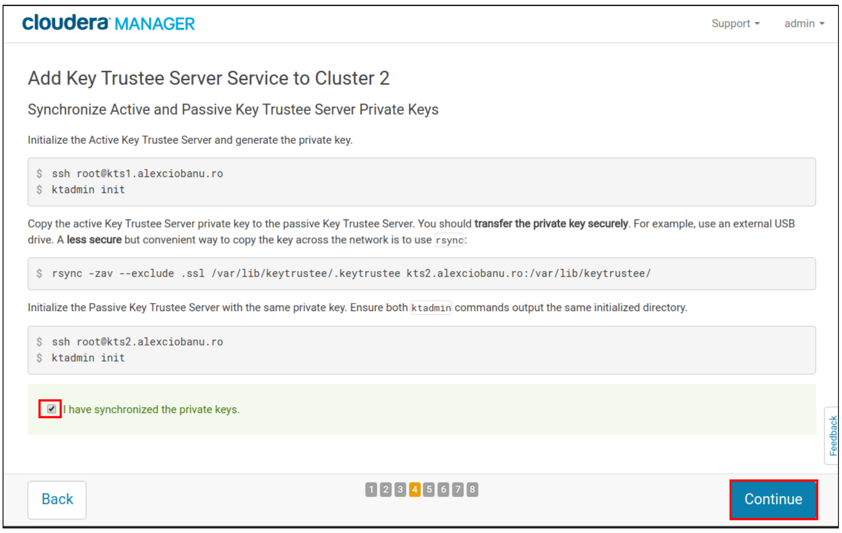
```
ssh root@kts2.alexciobanu.ro
ktadmin init
```

You should get a result similar to the one below

```
[root@kts2 ~]# ktadmin init
INFO:keytrustee.server.util:Creating self-signed cert
INFO:keytrustee.util:`/usr/bin/openssl req -nodes -new -days 3650 -subj /C=US/ST=TX/L=Austin/CN=kts2.alexciobanu.ro/E=keytrustee@kts2.alexcioba
Initialized directory for 4096R/EB9C63EFCD294264E79BDF31F0788AC403074A55
```

Ensure both ktadmin commands output the same **Initialized directory for** on both servers.

Once this is complete, go back to Cloudera Manager, tick the box **I have synchronized the private keys**., and click **Continue**



Add Key Trustee Server Service to Cluster 2

Synchronize Active and Passive Key Trustee Server Private Keys

Initialize the Active Key Trustee Server and generate the private key.

```
$ ssh root@kts1.alexciobanu.ro
$ ktadmin init
```

Copy the active Key Trustee Server private key to the passive Key Trustee Server. You should **transfer the private key securely**. For example, use an external USB drive. A **less secure** but convenient way to copy the key across the network is to use `rsync`:

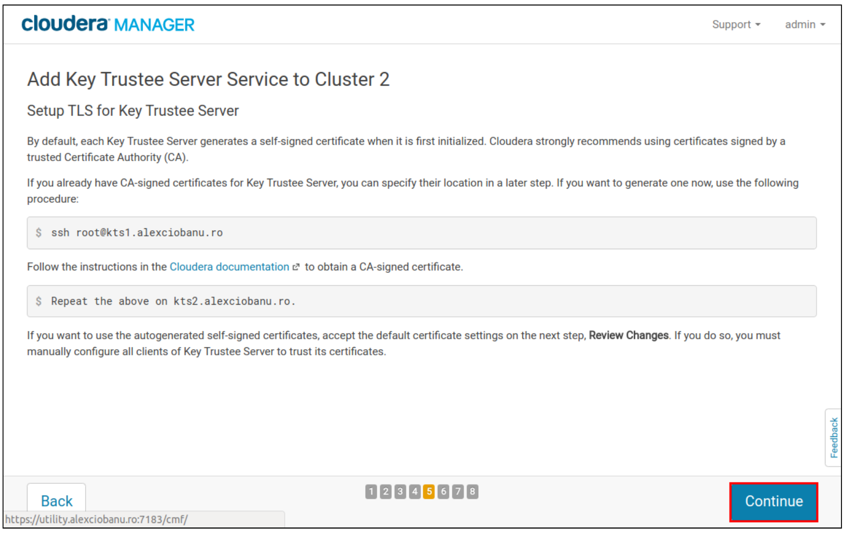
```
$ rsync -zav --exclude .ssl /var/lib/keytrustee/.keytrustee kts2.alexciobanu.ro:/var/lib/keytrustee/
```

Initialize the Passive Key Trustee Server with the same private key. Ensure both `ktadmin` commands output the same initialized directory.

```
$ ssh root@kts2.alexciobanu.ro
$ ktadmin init
```

☑ I have synchronized the private keys.

Next steps ask us to generate TLS for each of the Key Trustee Servers. As we have already done this as part of TLS configuration so we can skit this step.
Just lick **Continue**



Add Key Trustee Server Service to Cluster 2

Setup TLS for Key Trustee Server

By default, each Key Trustee Server generates a self-signed certificate when it is first initialized. Cloudera strongly recommends using certificates signed by a trusted Certificate Authority (CA).

If you already have CA-signed certificates for Key Trustee Server, you can specify their location in a later step. If you want to generate one now, use the following procedure:

```
$ ssh root@kts1.alexciobanu.ro
```

Follow the instructions in the Cloudera documentation to obtain a CA-signed certificate.

```
$ Repeat the above on kts2.alexciobanu.ro.
```

If you want to use the autogenerated self-signed certificates, accept the default certificate settings on the next step, **Review Changes**. If you do so, you must manually configure all clients of Key Trustee Server to trust its certificates.

We are now asked to enter the location of the pem files for the cluster. Fill in the following information.

Active Key Trustee Server TLS/SSL Server Private Key File (PEM Format): /opt/cloudera/security/x509/key.pem

Active Key Trustee Server TLS/SSL Server Certificate File (PEM Format): /opt/cloudera/security/x509/cert.pem

Active Key Trustee Server TLS/SSL Server CA Certificate (PEM Format): /opt/cloudera/security/truststore/ca-truststore.pem

Active Key Trustee Server TLS/SSL Private Key Password: password

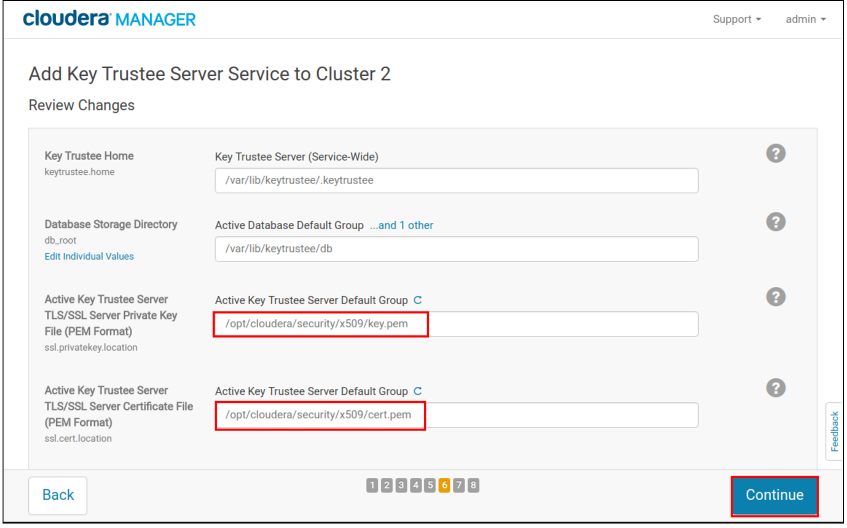
Passive Key Trustee Server TLS/SSL Server Private Key File (PEM Format): /opt/cloudera/security/x509/key.pem

Passive Key Trustee Server TLS/SSL Server Certificate File (PEM Format): /opt/cloudera/security/x509/cert.pem

Passive Key Trustee Server TLS/SSL Server CA Certificate (PEM Format): /opt/cloudera/security/x509/cert.pem

Passive Key Trustee Server TLS/SSL Private Key Password: password

Then click **Continue**



Add Key Trustee Server Service to Cluster 2

Review Changes

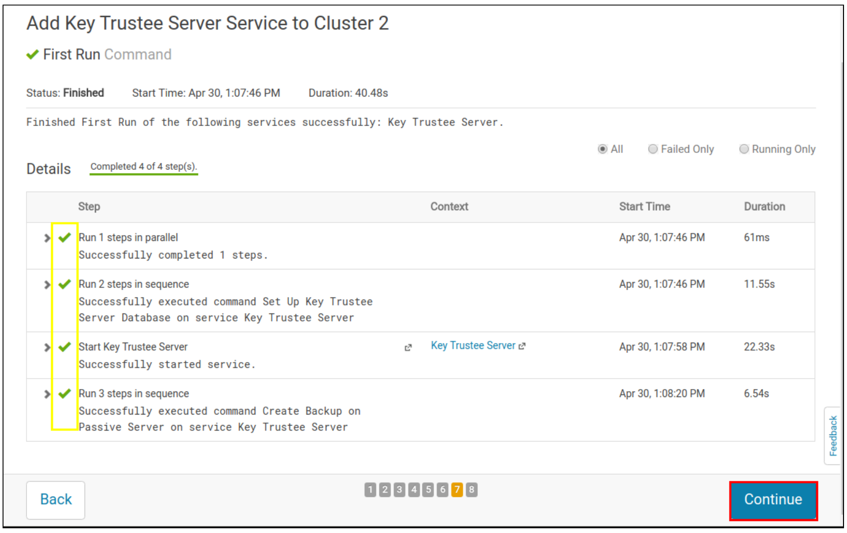
Key Trustee Home
keytrustee.home
Key Trustee Server (Service-Wide)
/var/lib/keytrustee/.keytrustee

Database Storage Directory
db_root
Edit Individual Values
Active Database Default Group ...and 1 other
/var/lib/keytrustee/db

Active Key Trustee Server TLS/SSL Server Private Key File (PEM Format)
ssl.privatekey.location
Active Key Trustee Server Default Group
/opt/cloudera/security/x509/key.pem

Active Key Trustee Server TLS/SSL Server Certificate File (PEM Format)
ssl.cert.location
Active Key Trustee Server Default Group
/opt/cloudera/security/x509/cert.pem

Cloudera Manager will now go through configuring and installing Key Trustee Sever. Once the process is complete click **Continue**



Add Key Trustee Server Service to Cluster 2

✓ First Run Command

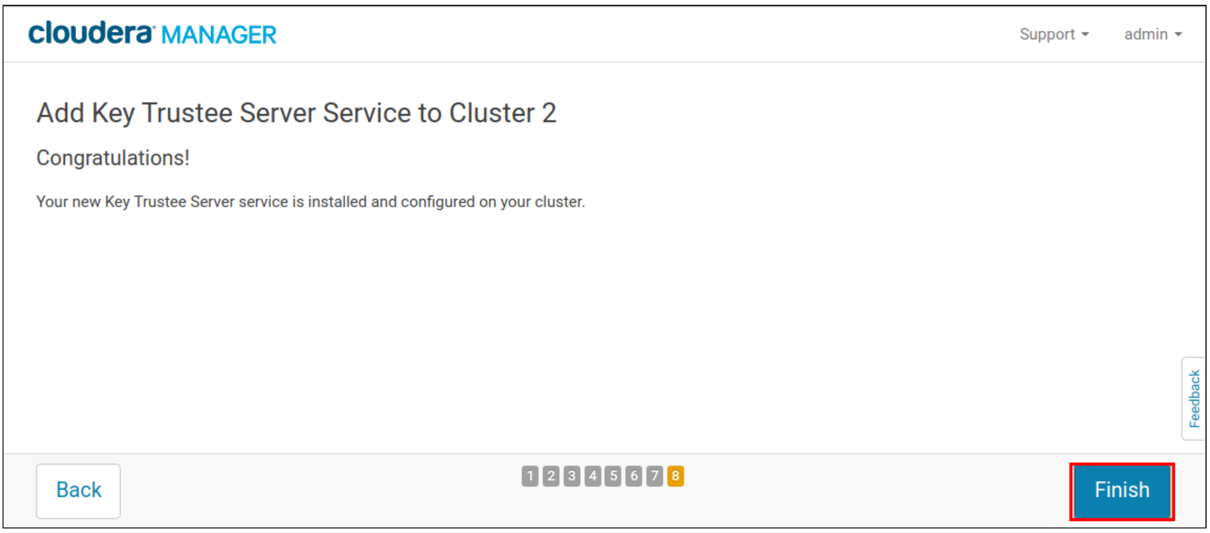
Status: Finished   Start Time: Apr 30, 1:07:46 PM   Duration: 40.48s

```
Finished First Run of the following services successfully: Key Trustee Server.
```

Details   Completed 4 of 4 step(s).

	Step	Context	Start Time	Duration
✓	Run 1 steps in parallel Successfully completed 1 steps.		Apr 30, 1:07:46 PM	61ms
✓	Run 2 steps in sequence Successfully executed command Set Up Key Trustee Server Database on service Key Trustee Server		Apr 30, 1:07:46 PM	11.55s
✓	Start Key Trustee Server Successfully started service.	Key Trustee Server	Apr 30, 1:07:58 PM	22.33s
✓	Run 3 steps in sequence Successfully executed command Create Backup on Passive Server on service Key Trustee Server		Apr 30, 1:08:20 PM	6.54s

You have now successfully installed Key Trustee Server. Click the **Finish** button.



Add Key Trustee Server Service to Cluster 2

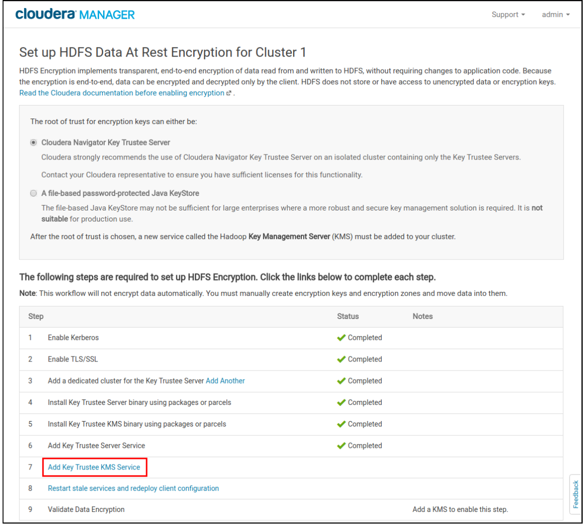
Congratulations!

Your new Key Trustee Server service is installed and configured on your cluster.

NOTE: It is very important to harden the servers where Key Trustee Server is running. Some instructions can be found here. It is strongly recommended they are followed. Additional information can be found here.

Once we are back to the home screen of the HDFS Data At Rest Encryption wizard we will need to deploy the Key Management Service (KMS) on our main cluster. This will act as a gateway between Key Trustee Server and the rest of the processes.

In the wizard click on the link: **Add Key Trustee KMS Service**



Set up HDFS Data At Rest Encryption for Cluster 1

HDFS Encryption implements transparent, end-to-end encryption of data read from and written to HDFS, without requiring changes to application code. Because the encryption is end-to-end, data can be encrypted and decrypted only by the client. HDFS does not store or have access to unencrypted data or encryption keys. Read the Cloudera documentation before enabling encryption.

The root of trust for encryption keys can either be:

● Cloudera Navigator Key Trustee Server

Cloudera strongly recommends the use of Cloudera Navigator Key Trustee Server on an isolated cluster containing only the Key Trustee Servers.

Contact your Cloudera representative to ensure you have sufficient licenses for this functionality.

○ A file-based password-protected Java KeyStore

The file-based Java KeyStore may not be sufficient for large enterprises where a more robust and secure key management solution is required. It is **not suitable** for production use.

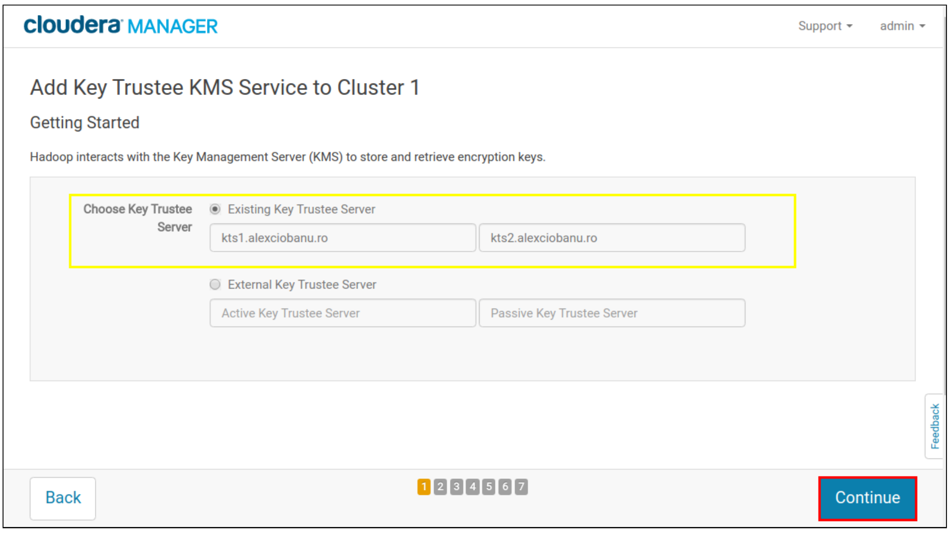
After the root of trust is chosen, a new service called the Hadoop **Key Management Server** (KMS) must be added to your cluster.

The following steps are required to set up HDFS Encryption. Click the links below to complete each step.

Note: This workflow will not encrypt data automatically. You must manually create encryption keys and encryption zones and move data into them.

Step		Status	Notes
1	Enable Kerberos	✓ Completed	
2	Enable TLS/SSL	✓ Completed	
3	Add a dedicated cluster for the Key Trustee Server Add Another	✓ Completed	
4	Install Key Trustee Server binary using packages or parcels	✓ Completed	
5	Install Key Trustee KMS binary using packages or parcels	✓ Completed	
6	Add Key Trustee Server Service	✓ Completed	
7	Add Key Trustee KMS Service		
8	Restart stale services and redeploy client configuration		
9	Validate Data Encryption		Add a KMS to enable this step.

In next window Cloudera Manager should auto detect your Key Trustee Server Installation. Ensure that happened and click **Continue**



Add Key Trustee KMS Service to Cluster 1

Getting Started

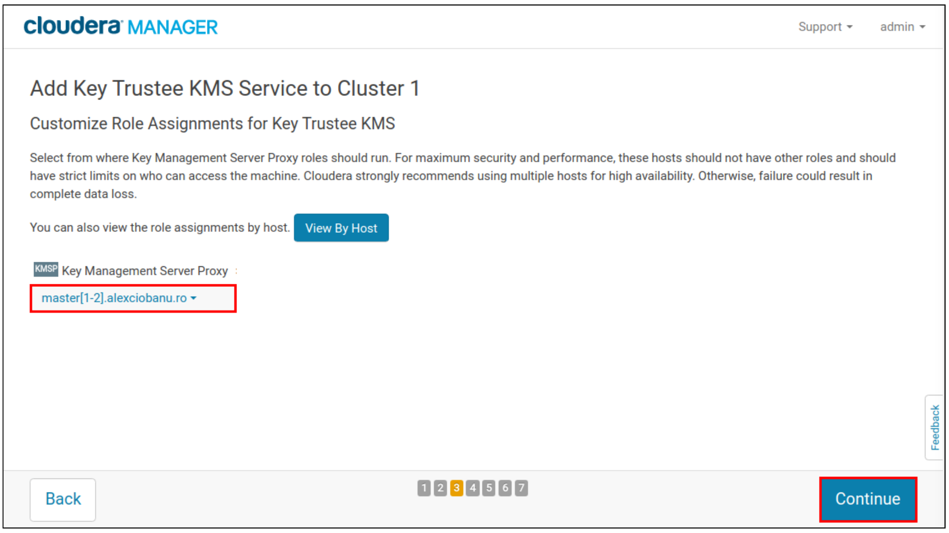
Hadoop interacts with the Key Management Server (KMS) to store and retrieve encryption keys.

Choose Key Trustee Server ● Existing Key Trustee Server
kts1.alexciobanu.ro kts2.alexciobanu.ro

○ External Key Trustee Server
Active Key Trustee Server Passive Key Trustee Server

In the next window you will need to chose the server on which to install the KMS. Please select the servers and click **Continue**.

In the Key Management Server Proxy section add the two servers which were added to the main cluster with no services for the KMS purposes. In non production environments, if you do not have these 2 dedicated servers you can use 2 master servers for this services.



Add Key Trustee KMS Service to Cluster 1

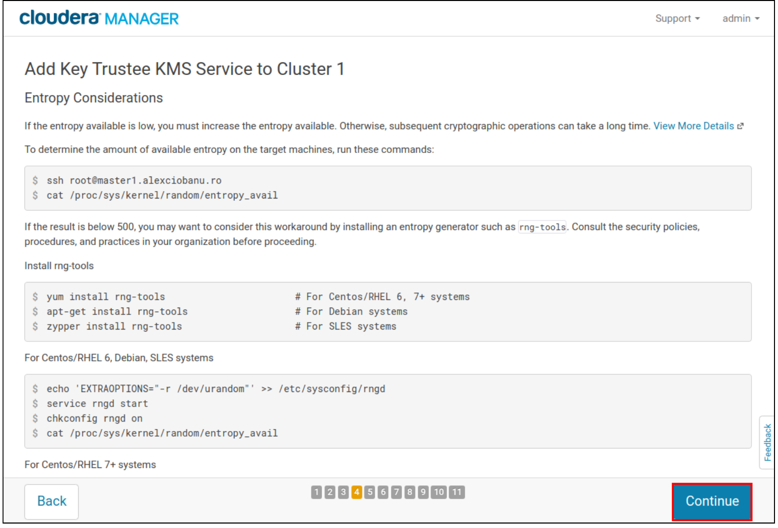
Customize Role Assignments for Key Trustee KMS

Select from where Key Management Server Proxy roles should run. For maximum security and performance, these hosts should not have other roles and should have strict limits on who can access the machine. Cloudera strongly recommends using multiple hosts for high availability. Otherwise, failure could result in complete data loss.

You can also view the role assignments by host. View By Host

KMSP Key Management Server Proxy
master[1-2].alexciobanu.ro ▾

The next window will ask us to install the entropy boosting application. We have done this already as part of the pre-requisites so we can simply click **Continue**



Add Key Trustee KMS Service to Cluster 1

Entropy Considerations

If the entropy available is low, you must increase the entropy available. Otherwise, subsequent cryptographic operations can take a long time. View More Details

To determine the amount of available entropy on the target machines, run these commands:

```
$ ssh root@master1.alexciobanu.ro
$ cat /proc/sys/kernel/random/entropy_avail
```

If the result is below 500, you may want to consider this workaround by installing an entropy generator such as `rng-tools`. Consult the security policies, procedures, and practices in your organization before proceeding.

Install rng-tools

```
$ yum install rng-tools                      # For Centos/RHEL 6, 7+ systems
$ apt-get install rng-tools                  # For Debian systems
$ zypper install rng-tools                   # For SLES systems
```

For Centos/RHEL 6, Debian, SLES systems

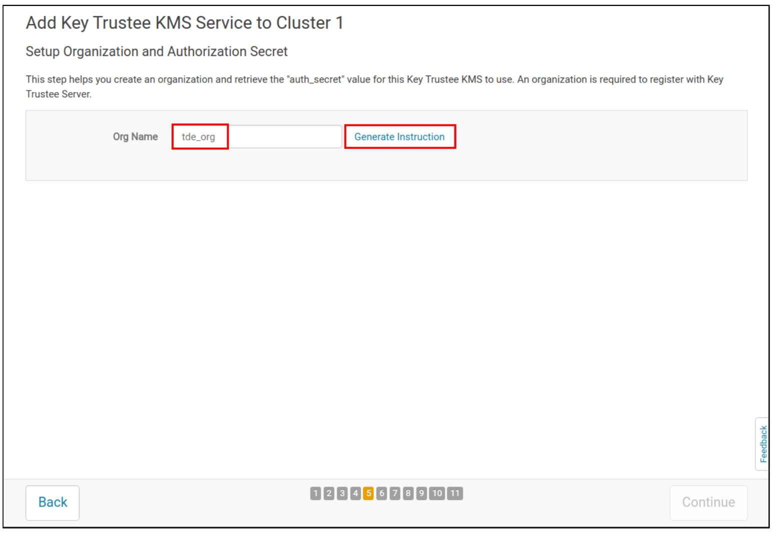
```
$ echo 'EXTRAOPTIONS="-r /dev/urandom"' >> /etc/sysconfig/rngd
$ service rngd start
$ chkconfig rngd on
$ cat /proc/sys/kernel/random/entropy_avail
```

For Centos/RHEL 7+ systems

In the next screen you will be asked to enter an organisation name. This is a logical container to group keys for encryption at rest in the Key Trustee Server. We will only use a single Org, hence this name does not mater, and will not be needed after setup, but it is required for the KTS to function correctly.Enter

Org name: tde_org

Then click **Generate Instructions**



Add Key Trustee KMS Service to Cluster 1

Setup Organization and Authorization Secret

This step helps you create an organization and retrieve the "auth_secret" value for this Key Trustee KMS to use. An organization is required to register with Key Trustee Server.

Org Name: tde_org

Follow the instructions displayed on the screen to authenticate the systems together

```
ssh root@kts1.alexciobanu.ro
keytrustee-orgtool add -n tde_org -c root@localhost
keytrustee-orgtool list
```

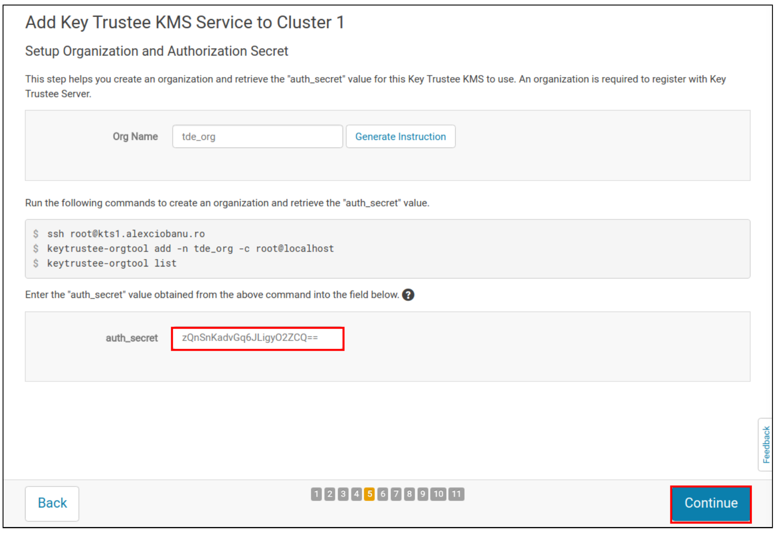
You should get output similar to the one below.

```
alex@alex-laptop:~$ ssh root@kts1.alexciobanu.ro
[root@kts1 ~]# keytrustee-orgtool add -n tde_org -c root@localhost
Dropped privileges to keytrustee
[root@kts1 ~]# keytrustee-orgtool list
Dropped privileges to keytrustee
{
    "tde_org": {
        "auth_secret": "zQnSnKadvGq6JLigyO2ZCQ==",
        "contacts": [
            "root@localhost"
        ],
        "creation": "2017-04-30T13:37:20",
        "expiration": "9999-12-31T18:59:59",
        "key_info": null,
        "name": "tde_org",
        "state": 0,
        "uuid": "UQNBZ3k0Bjo8rWYVfof6KQSz5f6cYja4sBoQXhNEjt0"
    }
}
```

Copy the "auth_secret" from the terminal into the appropriate window in your browser

auth_secret: zQnSnKadvGq6JLigyO2ZCQ==

Then click **Continue**



Add Key Trustee KMS Service to Cluster 1

Setup Organization and Authorization Secret

This step helps you create an organization and retrieve the "auth_secret" value for this Key Trustee KMS to use. An organization is required to register with Key Trustee Server.

Org Name: tde_org

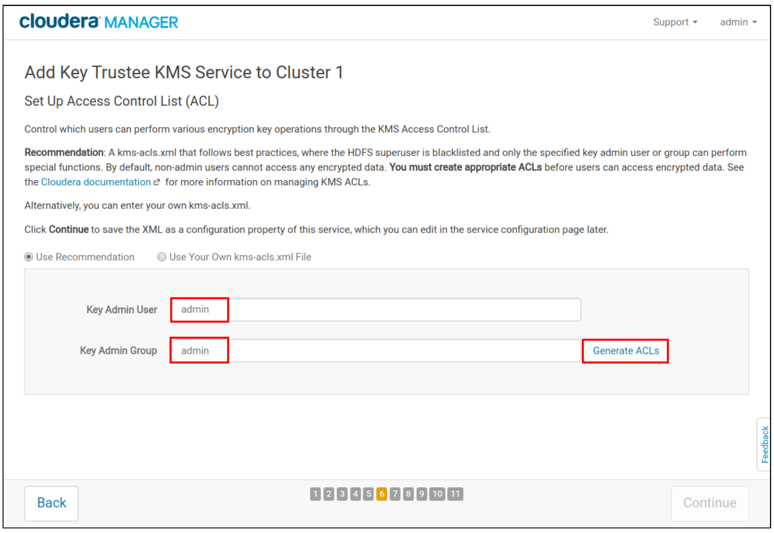
Run the following commands to create an organization and retrieve the "auth_secret" value.

```
$ ssh root@kts1.alexciobanu.ro
$ keytrustee-orgtool add -n tde_org -c root@localhost
$ keytrustee-orgtool list
```

Enter the "auth_secret" value obtained from the above command into the field below.

auth_secret: zQnSnKadvGq6JLigyO2ZCQ==

Next you will be able to configure the KMS Access Control List, which will enable fine control on who is able to decrypt data. Enter the KMS admin account in this page and click **Generate ACLs**.



Add Key Trustee KMS Service to Cluster 1

Set Up Access Control List (ACL)

Control which users can perform various encryption key operations through the KMS Access Control List.

Recommendation: A kms-acls.xml that follows best practices, where the HDFS superuser is blacklisted and only the specified key admin user or group can perform special functions. By default, non-admin users cannot access any encrypted data. **You must create appropriate ACLs** before users can access encrypted data. See the Cloudera documentation for more information on managing KMS ACLs.

Alternatively, you can enter your own kms-acls.xml.

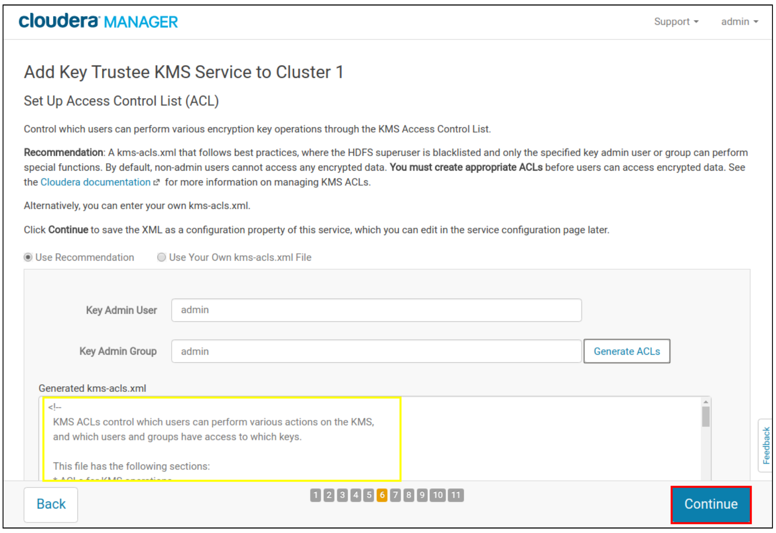
Click **Continue** to save the XML as a configuration property of this service, which you can edit in the service configuration page later.

● Use Recommendation ○ Use Your Own kms-acls.xml File

Key Admin User: admin

Key Admin Group: admin

Ensure the generated kms-acls.xml meets requirements and click **Continue**



Add Key Trustee KMS Service to Cluster 1

Set Up Access Control List (ACL)

Control which users can perform various encryption key operations through the KMS Access Control List.

Recommendation: A kms-acls.xml that follows best practices, where the HDFS superuser is blacklisted and only the specified key admin user or group can perform special functions. By default, non-admin users cannot access any encrypted data. **You must create appropriate ACLs** before users can access encrypted data. See the Cloudera documentation for more information on managing KMS ACLs.

Alternatively, you can enter your own kms-acls.xml.

Click **Continue** to save the XML as a configuration property of this service, which you can edit in the service configuration page later.

● Use Recommendation ○ Use Your Own kms-acls.xml File

Key Admin User: admin

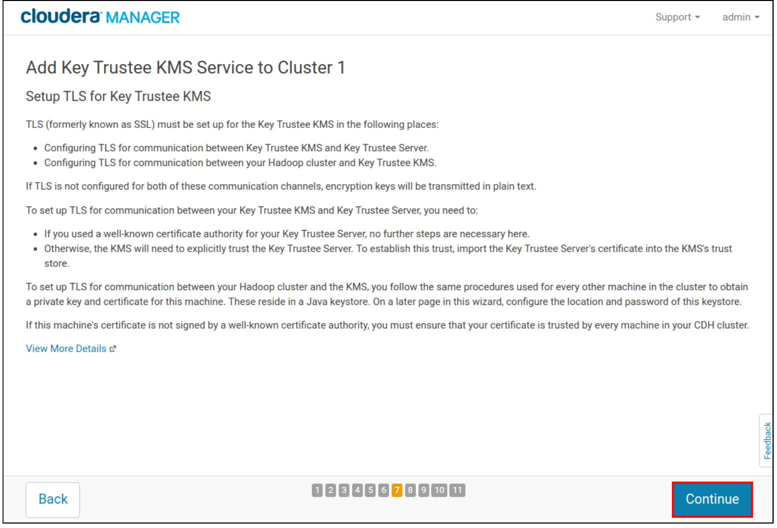
Key Admin Group: admin

Generated kms-acls.xml

```
<!--
  KMS ACLs control which users can perform various actions on the KMS,
  and which users and groups have access to which keys.

  This file has the following sections:
```

Next page discusses the requirements and restrictions of TLS. As we have already configured TLS in a previous section we can just click **Continue**.



Add Key Trustee KMS Service to Cluster 1

Setup TLS for Key Trustee KMS

TLS (formerly known as SSL) must be set up for the Key Trustee KMS in the following places:

- Configuring TLS for communication between Key Trustee KMS and Key Trustee Server.
- Configuring TLS for communication between your Hadoop cluster and Key Trustee KMS.

If TLS is not configured for both of these communication channels, encryption keys will be transmitted in plain text.

To set up TLS for communication between your Key Trustee KMS and Key Trustee Server, you need to:

- If you used a well-known certificate authority for your Key Trustee Server, no further steps are necessary here.
- Otherwise, the KMS will need to explicitly trust the Key Trustee Server. To establish this trust, import the Key Trustee Server's certificate into the KMS's trust store.

To set up TLS for communication between your Hadoop cluster and the KMS, you follow the same procedures used for every other machine in the cluster to obtain a private key and certificate for this machine. These reside in a Java keystore. On a later page in this wizard, configure the location and password of this keystore.

If this machine's certificate is not signed by a well-known certificate authority, you must ensure that your certificate is trusted by every machine in your CDH cluster.

View More Details

In the next page you will need to fill in the TLS options. Scroll down to the bottom and fill in the following options:

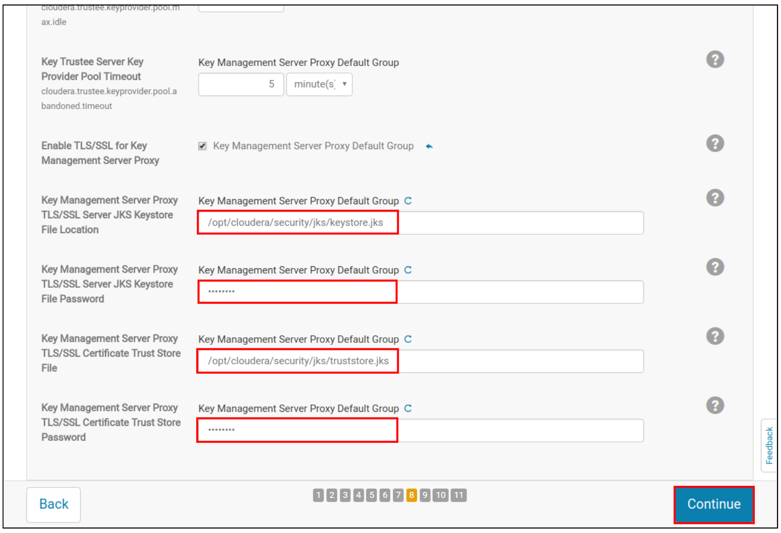
Key Management Server Proxy TLS/SSL Server JKS Keystore File Location: /opt/cloudera/security/jks/keystore.jks

Key Management Server Proxy TLS/SSL Server JKS Keystore File Password: password

Key Management Server Proxy TLS/SSL Certificate Trust Store File: /opt/cloudera/security/jks/truststore.jks

Key Management Server Proxy TLS/SSL Certificate Trust Store Password: password

Then click **Continue**



cloudera.trustee.keyprovider.pool.max.idle

Key Trustee Server Key Provider Pool Timeout
cloudera.trustee.keyprovider.pool.abandoned.timeout
Key Management Server Proxy Default Group
5 minute(s) ▾

Enable TLS/SSL for Key Management Server Proxy
☑ Key Management Server Proxy Default Group

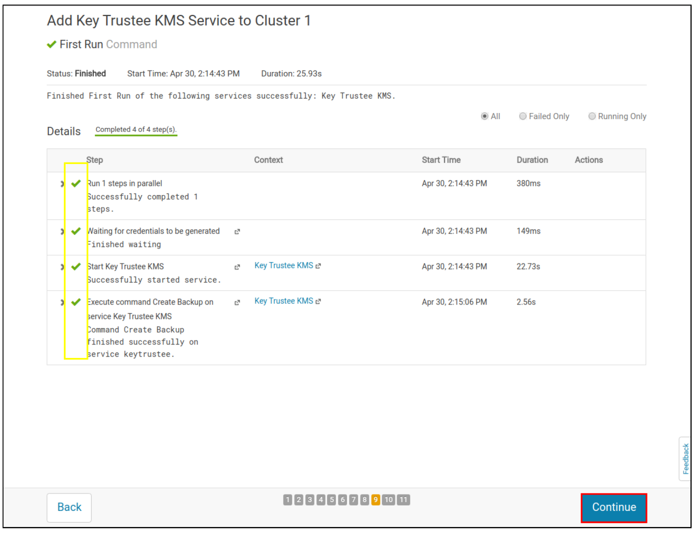
Key Management Server Proxy TLS/SSL Server JKS Keystore File Location
Key Management Server Proxy Default Group
/opt/cloudera/security/jks/keystore.jks

Key Management Server Proxy TLS/SSL Server JKS Keystore File Password
Key Management Server Proxy Default Group
••••••••

Key Management Server Proxy TLS/SSL Certificate Trust Store File
Key Management Server Proxy Default Group
/opt/cloudera/security/jks/truststore.jks

Key Management Server Proxy TLS/SSL Certificate Trust Store Password
Key Management Server Proxy Default Group
••••••••

Now Cloudera Manager will start deploying the service. Wait for things to complete successful and click **Continue**



Add Key Trustee KMS Service to Cluster 1

✓ First Run Command

Status: **Finished**   Start Time: Apr 30, 2:14:43 PM   Duration: 25.93s

```
Finished First Run of the following services successfully: Key Trustee KMS.
```

● All ○ Failed Only ○ Running Only

Details Completed 4 of 4 step(s).

	Step	Context	Start Time	Duration	Actions
✓	Run 1 steps in parallel Successfully completed 1 steps.		Apr 30, 2:14:43 PM	380ms	
✓	Waiting for credentials to be generated Finished waiting		Apr 30, 2:14:43 PM	149ms	
✓	Start Key Trustee KMS Successfully started service.	Key Trustee KMS	Apr 30, 2:14:43 PM	22.73s	
✓	Execute command Create Backup on service Key Trustee KMS Command Create Backup finished successfully on service keytrustee.	Key Trustee KMS	Apr 30, 2:15:06 PM	2.56s	

You will next have to synchronise the private keys across the 2 KMS servers.

NOTE: if rsync in not installed on the server run the commands, on both KMS servers:

```
yum install rsync
```

Follow the instructions the screen once rsync is installed. For example run the commands

```
ssh root@master2.alexciobanu.ro
rsync -zavc /var/lib/kms-keytrustee/keytrustee/.keytrustee master3.alexciobanu.ro:/var/lib/kms-keytrustee/keytrustee/
```

You should see output similar to that below on your screen

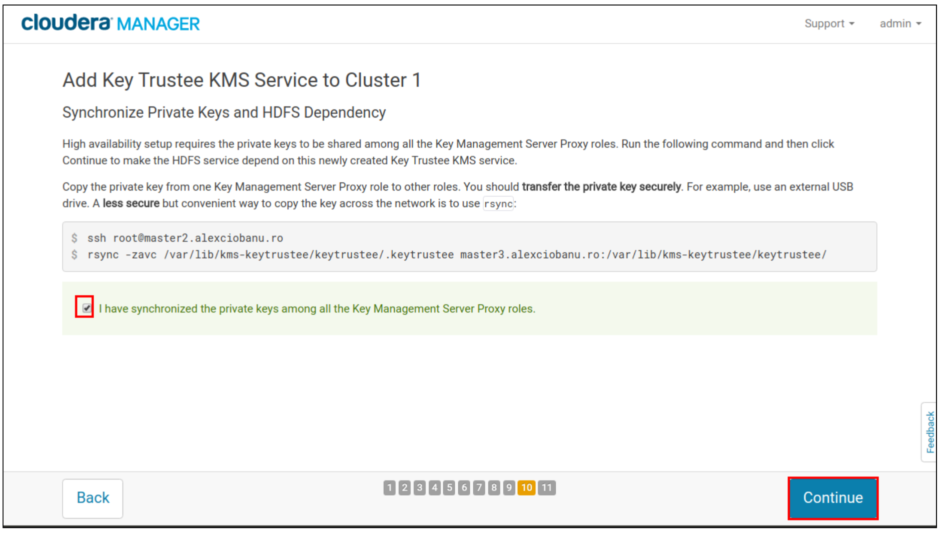
```
[root@master2 ~]# rsync -zavc /var/lib/kms-keytrustee/keytrustee/.keytrustee master3.alexciobanu.ro:/var/lib/kms-keytrustee/keytrustee/
Password:
sending incremental file list
.keytrustee/
.keytrustee/keytrustee.conf
.keytrustee/pubring.gpg
.keytrustee/secring.gpg

sent 9727 bytes  received 169 bytes  2199.11 bytes/sec
total size is 10477  speedup is 1.06
```

Once that is successful, back in your browser window select the option:

I have synchronized the private keys among all the Key Management Server Proxy roles: **Tick**

Then click **Continue**



Add Key Trustee KMS Service to Cluster 1

Synchronize Private Keys and HDFS Dependency

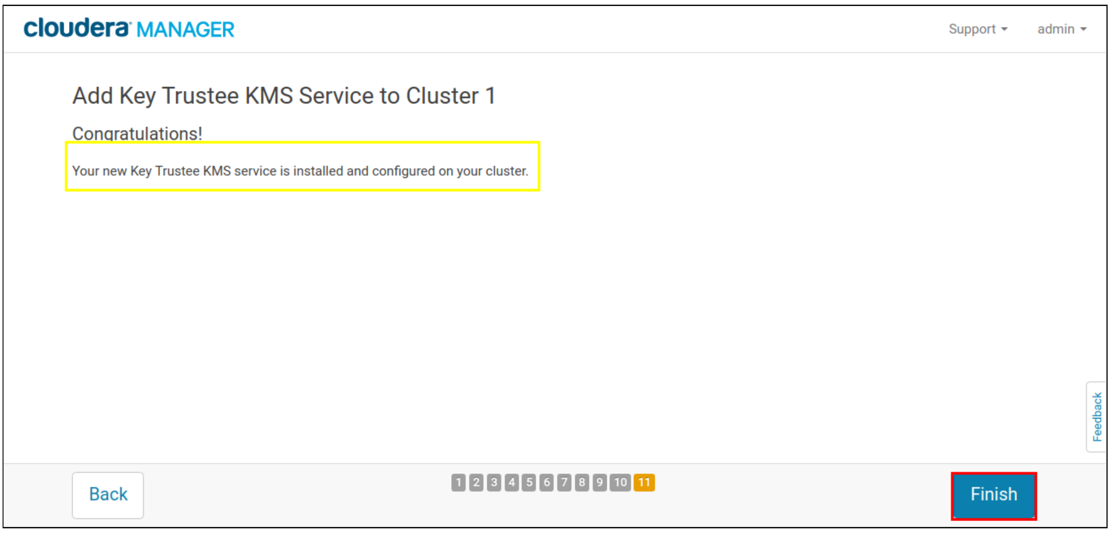
High availability setup requires the private keys to be shared among all the Key Management Server Proxy roles. Run the following command and then click Continue to make the HDFS service depend on this newly created Key Trustee KMS service.

Copy the private key from one Key Management Server Proxy role to other roles. You should **transfer the private key securely**. For example, use an external USB drive. A **less secure** but convenient way to copy the key across the network is to use `rsync`:

```
$ ssh root@master2.alexciobanu.ro
$ rsync -zavc /var/lib/kms-keytrustee/keytrustee/.keytrustee master3.alexciobanu.ro:/var/lib/kms-keytrustee/keytrustee/
```

☑ I have synchronized the private keys among all the Key Management Server Proxy roles.

The Key Trustee KMS will now be started. Wait for the action to complete and click **Finish**

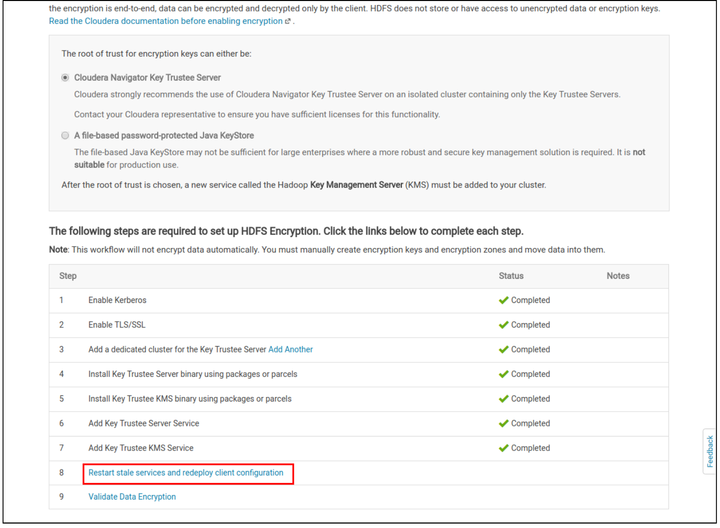


Add Key Trustee KMS Service to Cluster 1

Congratulations!

Your new Key Trustee KMS service is installed and configured on your cluster.

The final step in this configuration is to restart the cluster for the changes to take effect. In the HDFS Data At Rest Encryption wizard click on the link **Restart stale services and redeploy client configuration**



the encryption is end-to-end, data can be encrypted and decrypted only by the client. HDFS does not store or have access to unencrypted data or encryption keys. Read the Cloudera documentation before enabling encryption .

The root of trust for encryption keys can either be:

● Cloudera Navigator Key Trustee Server

Cloudera strongly recommends the use of Cloudera Navigator Key Trustee Server on an isolated cluster containing only the Key Trustee Servers.

Contact your Cloudera representative to ensure you have sufficient licenses for this functionality.

○ A file-based password-protected Java KeyStore

The file-based Java KeyStore may not be sufficient for large enterprises where a more robust and secure key management solution is required. It is **not suitable** for production use.

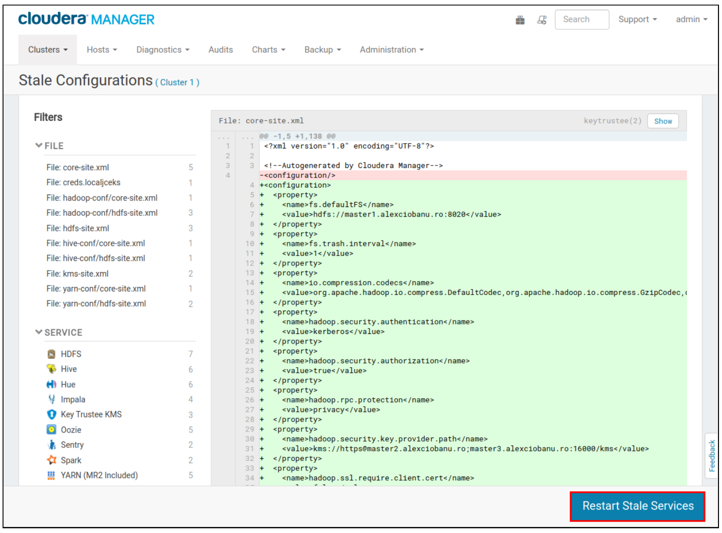
After the root of trust is chosen, a new service called the Hadoop **Key Management Server** (KMS) must be added to your cluster.

The following steps are required to set up HDFS Encryption. Click the links below to complete each step.

Note: This workflow will not encrypt data automatically. You must manually create encryption keys and encryption zones and move data into them.

Step		Status	Notes
1	Enable Kerberos	✓ Completed	
2	Enable TLS/SSL	✓ Completed	
3	Add a dedicated cluster for the Key Trustee Server Add Another	✓ Completed	
4	Install Key Trustee Server binary using packages or parcels	✓ Completed	
5	Install Key Trustee KMS binary using packages or parcels	✓ Completed	
6	Add Key Trustee Server Service	✓ Completed	
7	Add Key Trustee KMS Service	✓ Completed	
8	Restart stale services and redeploy client configuration		
9	Validate Data Encryption		

Now click on **Restart Stale Services** to restart the cluster



Stale Configurations (Cluster 1)

Filters

FILE

File: core-site.xml	5
File: creds.localjceks	1
File: hadoop-conf/core-site.xml	1
File: hadoop-conf/hdfs-site.xml	3
File: hdfs-site.xml	3
File: hive-conf/core-site.xml	1
File: hive-conf/hdfs-site.xml	1
File: kms-site.xml	2
File: yarn-conf/core-site.xml	1
File: yarn-conf/hdfs-site.xml	2

SERVICE

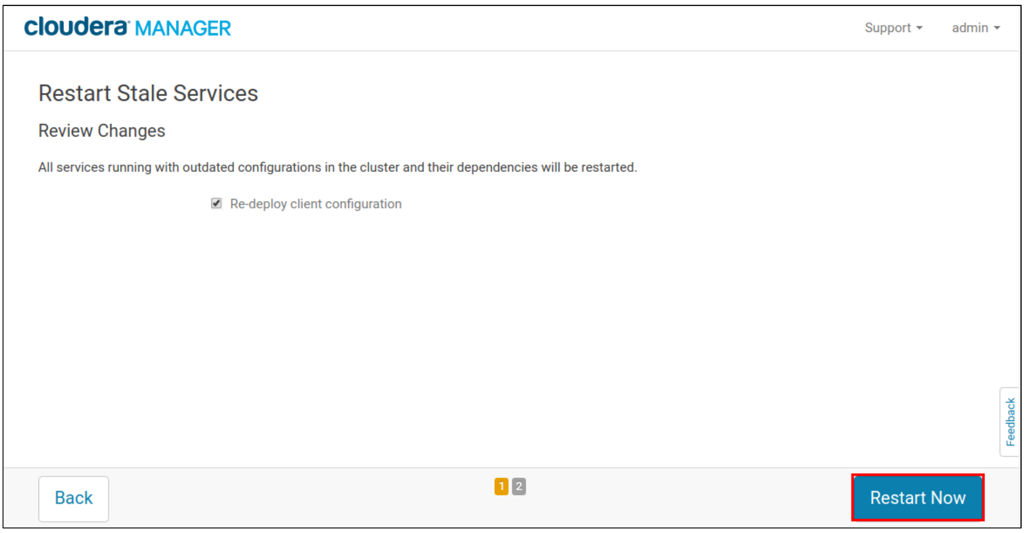
HDFS	7
Hive	6
Hue	6
Impala	4
Key Trustee KMS	3
Oozie	5
Sentry	2
Spark	2
YARN (MR2 Included)	5

File: core-site.xml   keytrustee(2)   Show

```
@@ -1,5 +1,138 @@
 <?xml version="1.0" encoding="UTF-8"?>

 <!--Autogenerated by Cloudera Manager-->
-<configuration/>
+<configuration>
+  <property>
+    <name>fs.defaultFS</name>
+    <value>hdfs://master1.alexciobanu.ro:8020</value>
+  </property>
+  <property>
+    <name>fs.trash.interval</name>
+    <value>1</value>
+  </property>
+  <property>
+    <name>io.compression.codecs</name>
+    <value>org.apache.hadoop.io.compress.DefaultCodec,org.apache.hadoop.io.compress.GzipCodec,
+  </property>
+  <property>
+    <name>hadoop.security.authentication</name>
+    <value>kerberos</value>
+  </property>
+  <property>
+    <name>hadoop.security.authorization</name>
+    <value>true</value>
+  </property>
+  <property>
+    <name>hadoop.rpc.protection</name>
+    <value>privacy</value>
+  </property>
+  <property>
+    <name>hadoop.security.key.provider.path</name>
+    <value>kms://https@master2.alexciobanu.ro;master3.alexciobanu.ro:16000/kms</value>
+  </property>
+  <property>
+    <name>hadoop.ssl.require.client.cert</name>
```

Now we can click **Restart Now** to restart the cluster



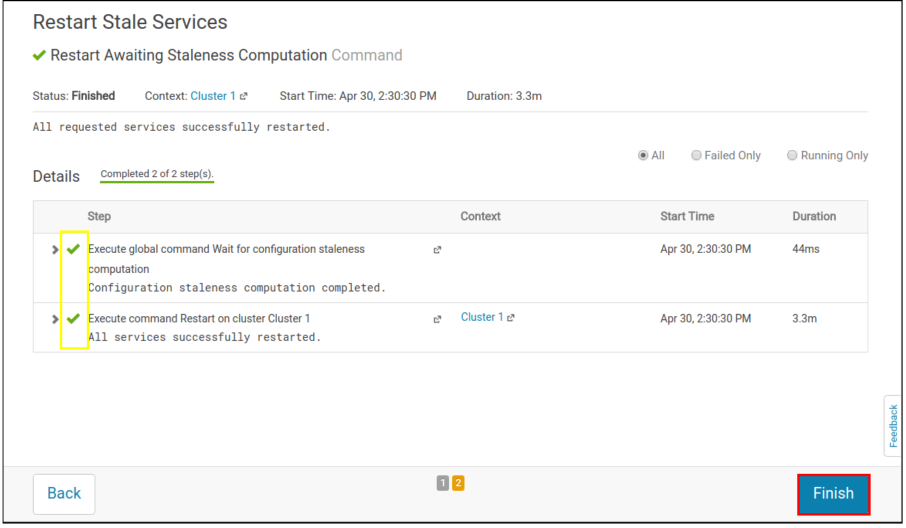
Restart Stale Services

Review Changes

All services running with outdated configurations in the cluster and their dependencies will be restarted.

☑ Re-deploy client configuration

Once the restart is complete, click the **Finish** button



Restart Stale Services

✓ Restart Awaiting Staleness Computation Command

Status: **Finished**   Context: Cluster 1   Start Time: Apr 30, 2:30:30 PM   Duration: 3.3m

```
All requested services successfully restarted.
```

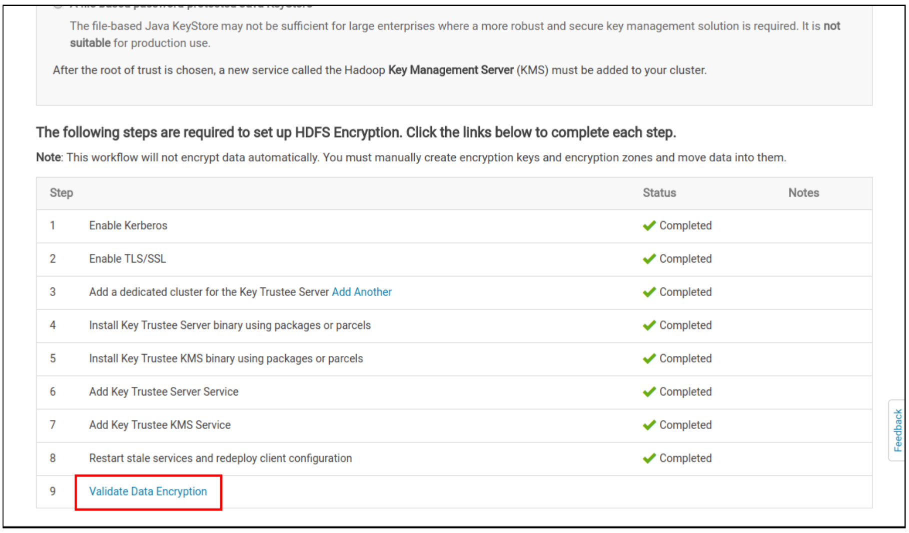
● All   ○ Failed Only   ○ Running Only

Details   Completed 2 of 2 step(s).

	Step	Context	Start Time	Duration
✓	Execute global command Wait for configuration staleness computation `Configuration staleness computation completed.`		Apr 30, 2:30:30 PM	44ms
✓	Execute command Restart on cluster Cluster 1 `All services successfully restarted.`	Cluster 1	Apr 30, 2:30:30 PM	3.3m

Transparent data encryption is now enabled. We can now test Data Encryption.

In the HDFS Data At Rest Encryption wizard click on the last link available to get instruction on smoke testing the system



○ A file-based password-protected Java KeyStore

The file-based Java KeyStore may not be sufficient for large enterprises where a more robust and secure key management solution is required. It is **not suitable** for production use.

After the root of trust is chosen, a new service called the Hadoop **Key Management Server** (KMS) must be added to your cluster.

The following steps are required to set up HDFS Encryption. Click the links below to complete each step.

Note: This workflow will not encrypt data automatically. You must manually create encryption keys and encryption zones and move data into them.

Step		Status	Notes
1	Enable Kerberos	✓ Completed	
2	Enable TLS/SSL	✓ Completed	
3	Add a dedicated cluster for the Key Trustee Server Add Another	✓ Completed	
4	Install Key Trustee Server binary using packages or parcels	✓ Completed	
5	Install Key Trustee KMS binary using packages or parcels	✓ Completed	
6	Add Key Trustee Server Service	✓ Completed	
7	Add Key Trustee KMS Service	✓ Completed	
8	Restart stale services and redeploy client configuration	✓ Completed	
9	Validate Data Encryption		

Follow the instructions on screen.

Smoke Testing Transparent Data Encryption

To test encryption we must first create an encrypted zone and an hdfs folder point to be encrypted. Use command similar to the ones below at achieve that task.

```
ssh user@edge1
kinit admin
hadoop key create mykey1
hdfs dfs -mkdir /tmp/zone1
```

Next we must make the HDFS path for encryption. The commands would look like:

```
kinit hdfs
hdfs crypto -createZone -keyName mykey1 -path /tmp/zone1
```

Create a file, put it in your zone and ensure the file can be encrypted.

```
kinit admin
echo "Hello World" > /tmp/helloWorld.txt
hdfs dfs -put /tmp/helloWorld.txt /tmp/zone1
hdfs dfs -cat /tmp/zone1/helloWorld.txt
```

Now we need to ensure the file that is stored is encrypted.

```
kinit hdfs
hadoop fs -cat /.reserved/raw/tmp/zone1/helloWorld.txt
```

You should get output similar to the following:

```
hadoop fs -cat /.reserved/raw/tmp/zone1/helloWorld.txt
```

```
nL:▒.(▒▒^L3
```

This proves the data is stored in encrypted format.

We can now clean up out folders

```
hadoop fs -rm -R /tmp/zone1
kdestroy
```

You now have Encryption at rest configured for data sitting in HDFS.

No labels

ATLASSIAN