Cloudera Runtime 7.0.2

Configuring Advanced Security Options for Apache Ranger

Date published: 2019-11-01
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Configure Kerberos authentication for Apache Ranger

How to configure Kerberos Authentication for Apache Ranger

About this task

Kerberos authentication for Apache Ranger is automatically configured when HDFS Kerberos authentication is configured in Cloudera Manager (typically using the Cloudera Manager Kerberos Wizard). In this way, the actions that Ranger authorizes are sure to be requested by authenticated users.

Specifically, Ranger depends on the HDFS hadoop.security.authentication property to enable or disable Kerberos authentication. When the hadoop.security.authentication property is updated, the Ranger service gets a restart indicator for the core-site.xml file that resides inside the Ranger service conf directory generated by Cloudera Manager.

Ranger Kerberos authentication is automatically enabled when HDFS Kerberos authentication is enabled.

Related Information

Enabling Kerberos Authentication for CDP

Configure TLS/SSL for Apache Ranger

How to configure TLS/SSL for Apache Ranger

About this task

Procedure

1. In Cloudera Manager, select Ranger, then click the Configuration tab.
2. Under Category, select Security.
3. Set the following properties.

Table 1: Apache Ranger TLS/SSL Settings

Configuration Property	Description
Enable TLS/SSL for Ranger Admin ranger.service.https.attrib.ssl.enabled	Select this check box to encrypt communication between clients and Ranger Admin using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Ranger Admin TLS/SSL Server JKS Keystore File Location ranger.https.attrib.keystore.file	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Ranger Admin is acting as a TLS/SSL server. The keystore must be in JKS format.
Ranger Admin TLS/SSL Server JKS Keystore File Password ranger.service.https.attrib.keystore.pass	The password for the Ranger Admin JKS keystore file.
Ranger Admin TLS/SSL Client Trust Store File ranger.truststore.file	The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that Ranger Admin might connect to. This is used when Ranger Admin is the client in a TLS/SSL connection. This trust store must contain the certificate(s) used to sign the connected service(s). If this parameter is not provided, the default list of well known certificate authorities is used.

Configuration Property	Description
Ranger Admin TLS/SSL Client Trust Store Password ranger.truststore.password	The password for the Ranger Admin TLS/SSL Certificate trust store file. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.
Enable TLS/SSL for Ranger Tagsync	Select this check box to encrypt communication between clients and Ranger Tagsync using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Ranger Tagsync TLS/SSL Server JKS Keystore File Location xasecure.policymgr.clientssl.keystore	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Ranger Tagsync is acting as a TLS/SSL server. The keystore must be in JKS format.
Ranger Tagsync TLS/SSL Server JKS Keystore File Password xasecure.policymgr.clientssl.keystore.password	The password for the Ranger Tagsync JKS keystore file.
Ranger Tagsync TLS/SSL Client Trust Store Password xasecure.policymgr.clientssl.truststore.password	The password for the Ranger Tagsync TLS/SSL Certificate trust store file. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.
Ranger Usersync TLS/SSL Client Trust Store File ranger.usersync.truststore.file	The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that Ranger Usersync might connect to. This is used when Ranger Usersync is the client in a TLS/SSL connection. This trust store must contain the certificate(s) used to sign the connected service(s). If this parameter is not provided, the default list of well known certificate authorities is used.
Ranger Usersync TLS/SSL Client Trust Store Password ranger.usersync.truststore.password	The password for the Ranger Usersync TLS/SSL certificate trust store File. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.

4. Click Save Changes.

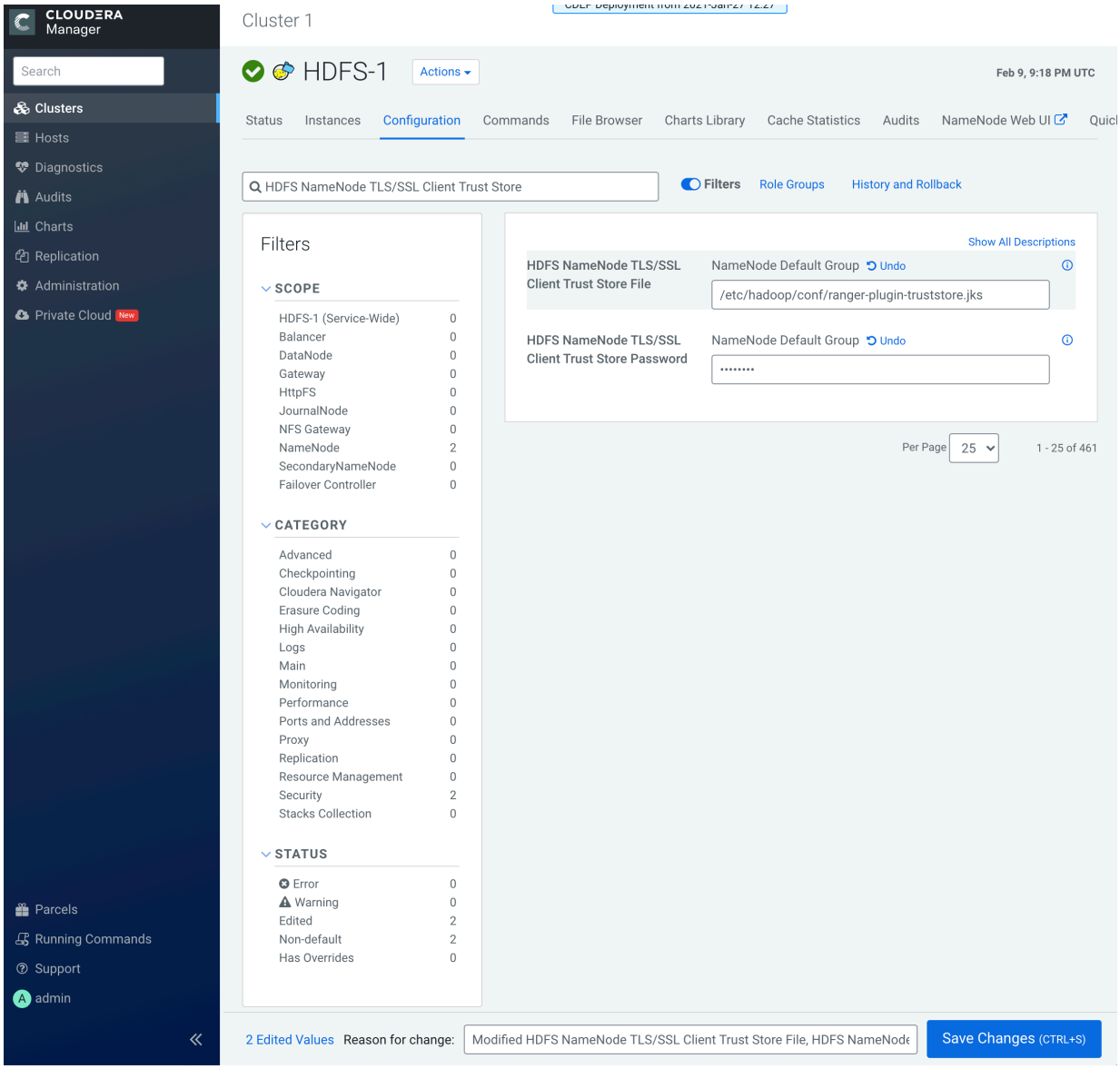
5. In order for services to communicate successfully with Ranger, you must set the following properties in each service that has Ranger authorization enabled to ensure that the Ranger Admin certificate is imported into the trust store.

   - TLS/SSL Client Trust Store File
   - TLS/SSL Client Trust Store Password

   For example, for HDFS select HDFS > Configuration in Cloudera Manager, then search for "HDFS NameNode TLS/SSL Client Trust Store", or use the Security Category to find and set the following properties:

   - HDFS NameNode TLS/SSL Client Trust Store File
   - HDFS NameNode TLS/SSL Client Trust Store Password

   Important: Repeat this procedure for all services that have Ranger authorization enabled.



6. Click Save Changes.

Configuring Apache Ranger High Availability

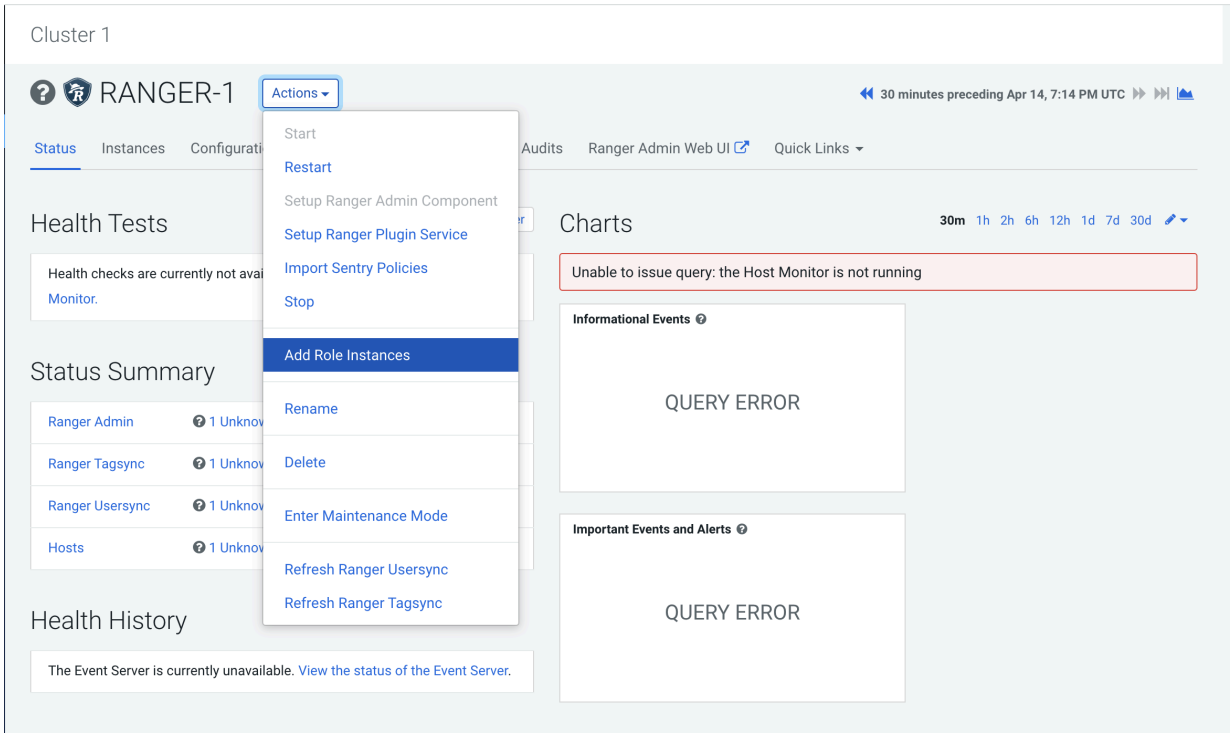
How to configure High Availability (HA) for Apache Ranger.

Configure Ranger Admin High Availability

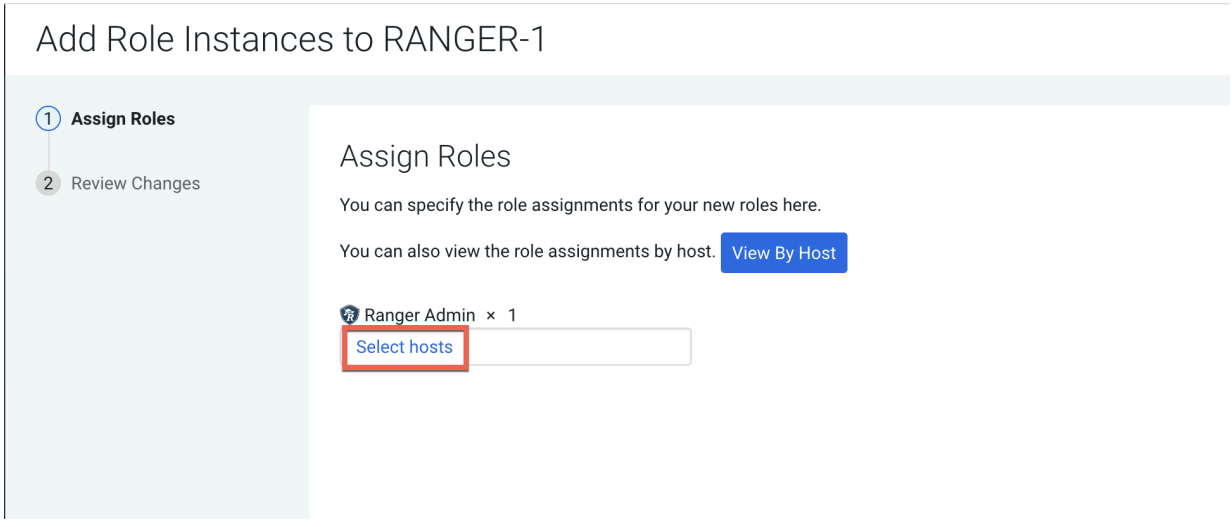
How to configure Ranger Admin High Availability (HA) by adding additional Ranger Admin role instances.

Procedure

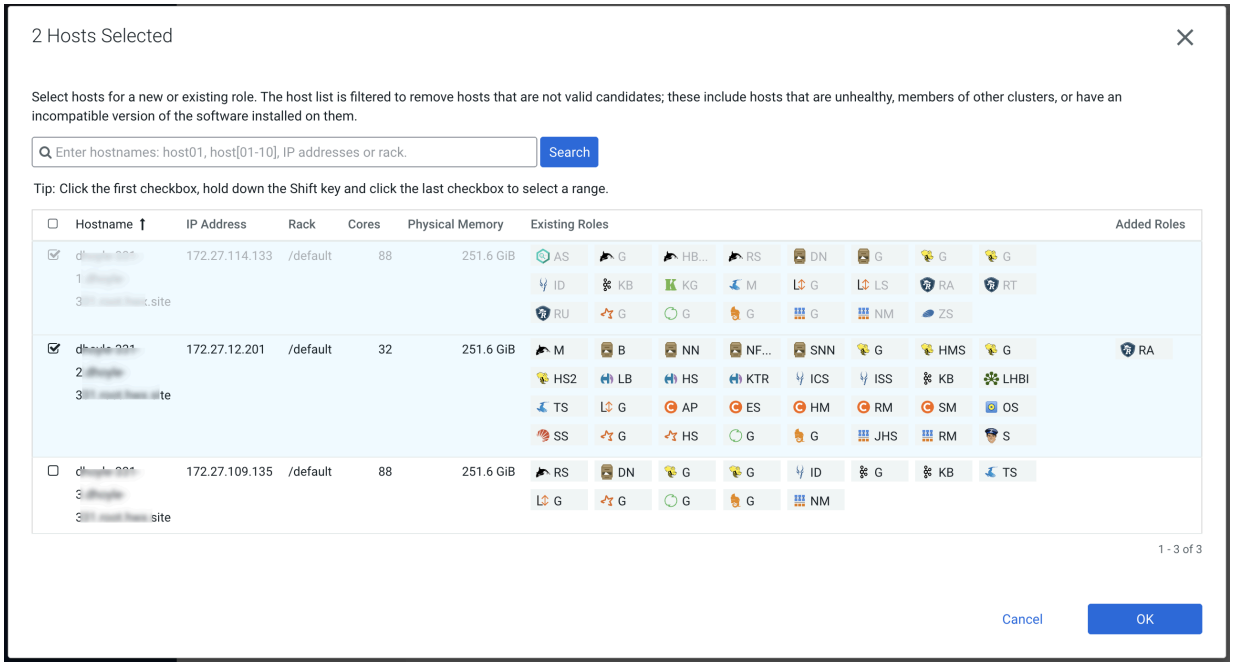
1. In Cloudera Manager, select Ranger, then select Actions > Add Role Instances.



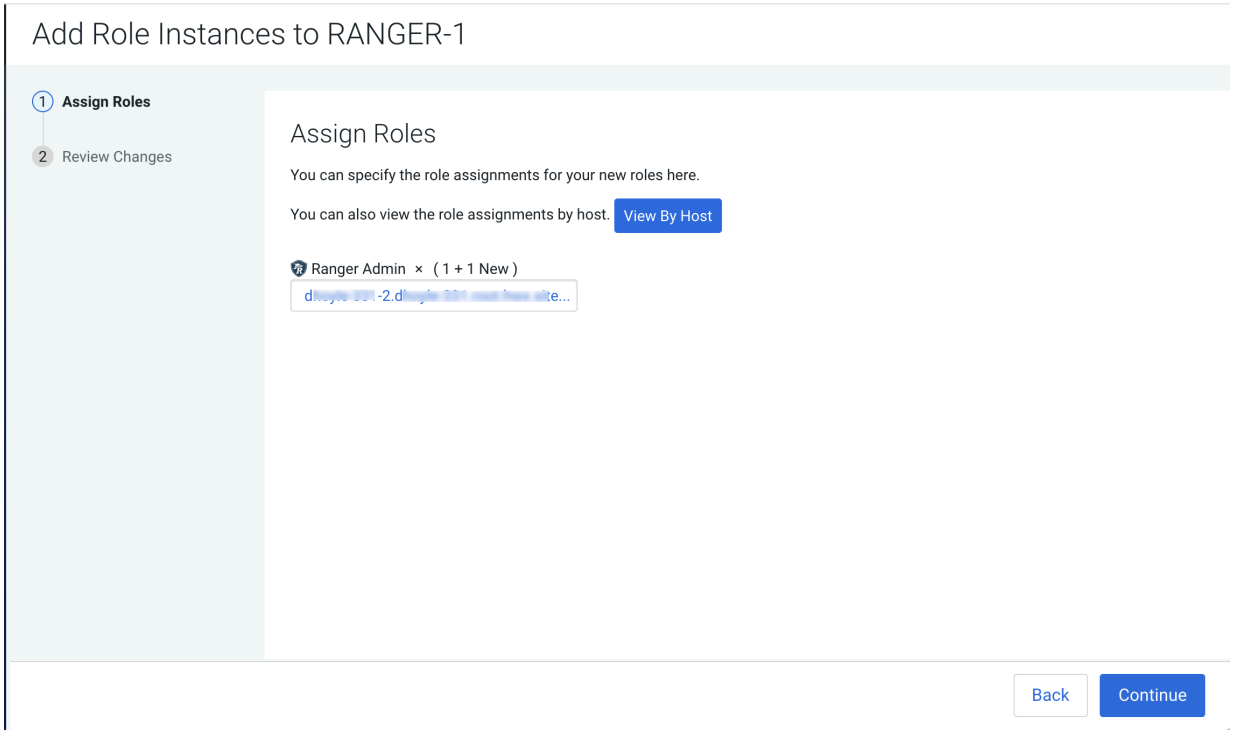
2. On the Add Role Instances page, click Select hosts.



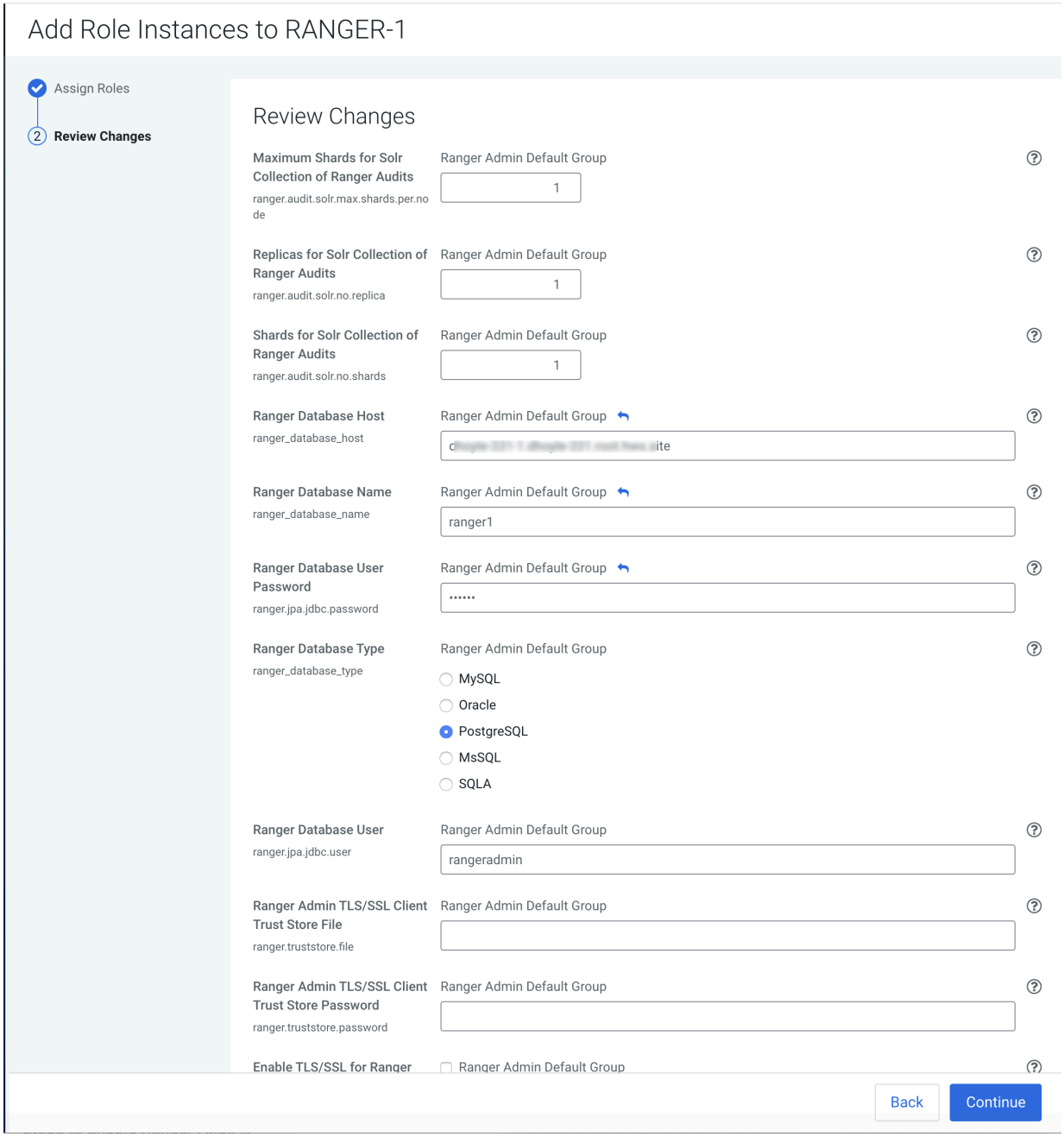
3. On the selected hosts page, the primary Ranger Admin host is selected by default. Select a backup Ranger host. A Ranger Admin (RA) icon appears in the Added Roles column for the selected backup host. Click OK to continue.



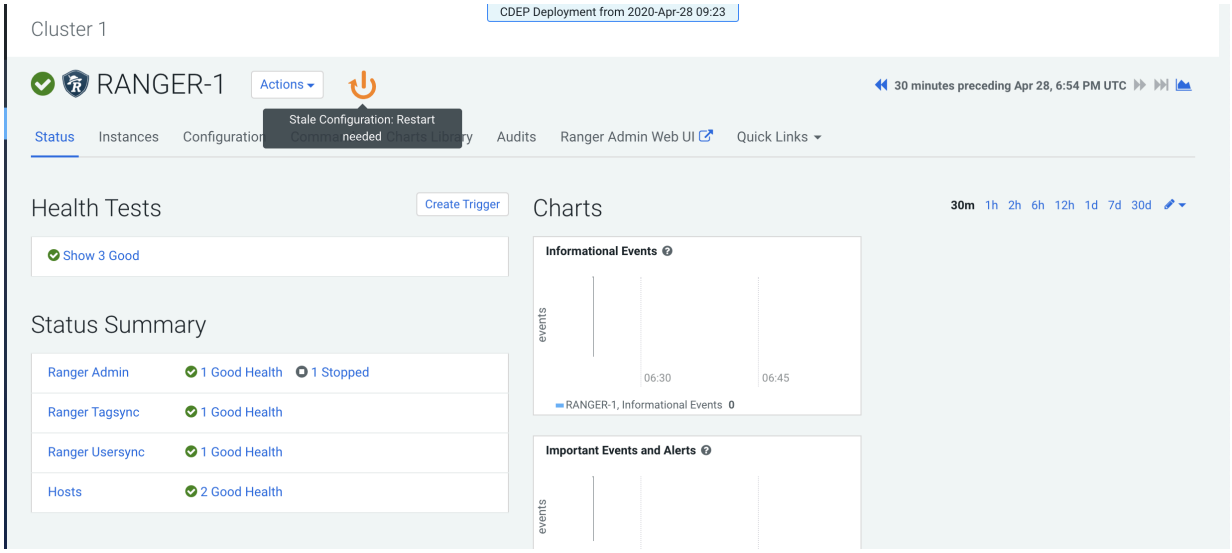
4. The Add Role Instances page is redisplayed with the new backup host. Click Continue.



5. Review the settings on the Review Changes page, then click Continue.

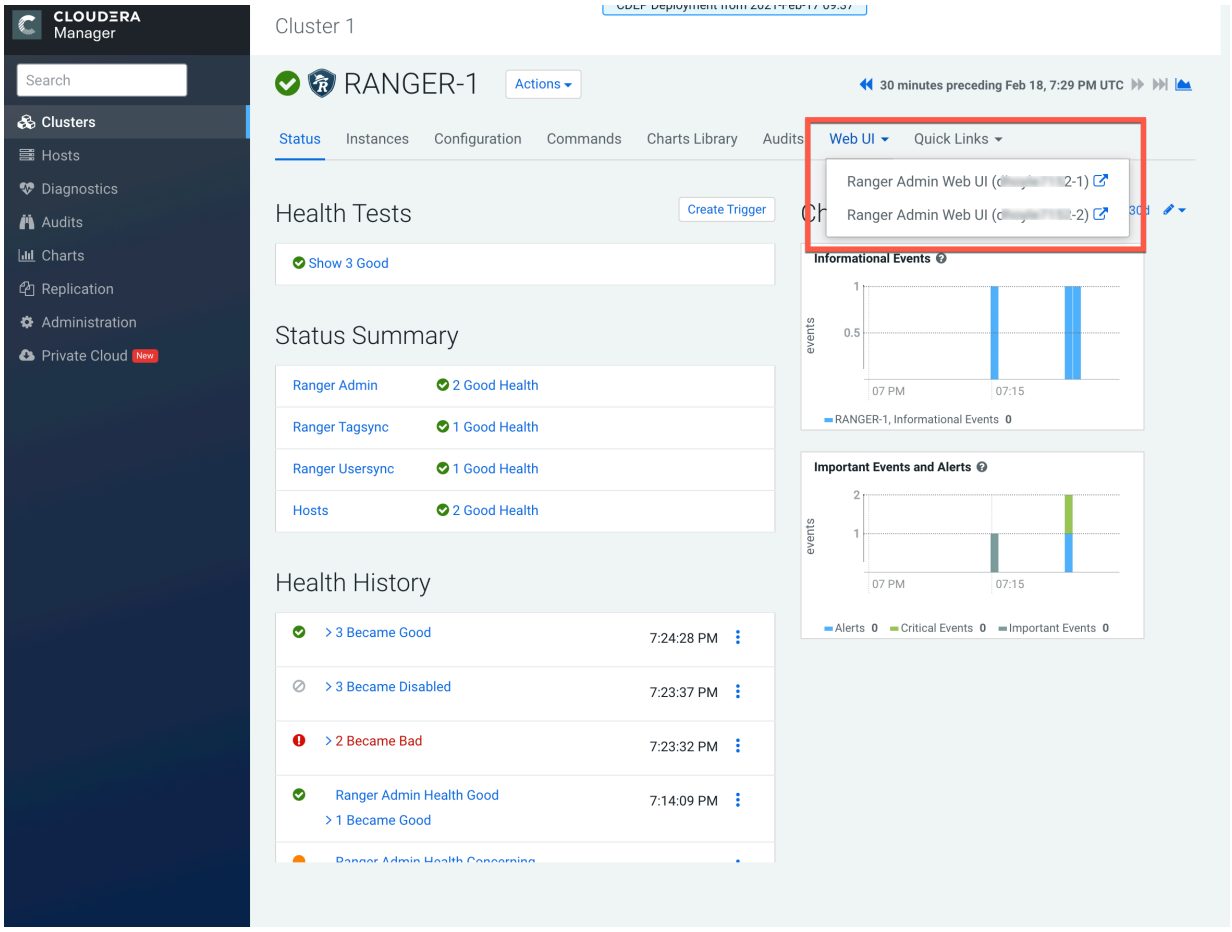


6. Restart the stale Ranger configuration, then click Finish.



7. After restart you will see two URLs for the Ranger Admin Web UI.

- Requests are distributed to the multiple Ranger Admin instances in a round-robin fashion.
- If a connection is refused (indicating a failure), requests are automatically rerouted to the alternate Ranger Admin instance. However, you must manually switch to the alternate Ranger Admin Web UI.
- For all services that have the Ranger plugin enabled, the value of the ranger.plugin.<service>.policy.rest.url property changes to http://<RANGER-ADMIN-1>:6080,http://<RANGER-ADMIN-2>:6080.

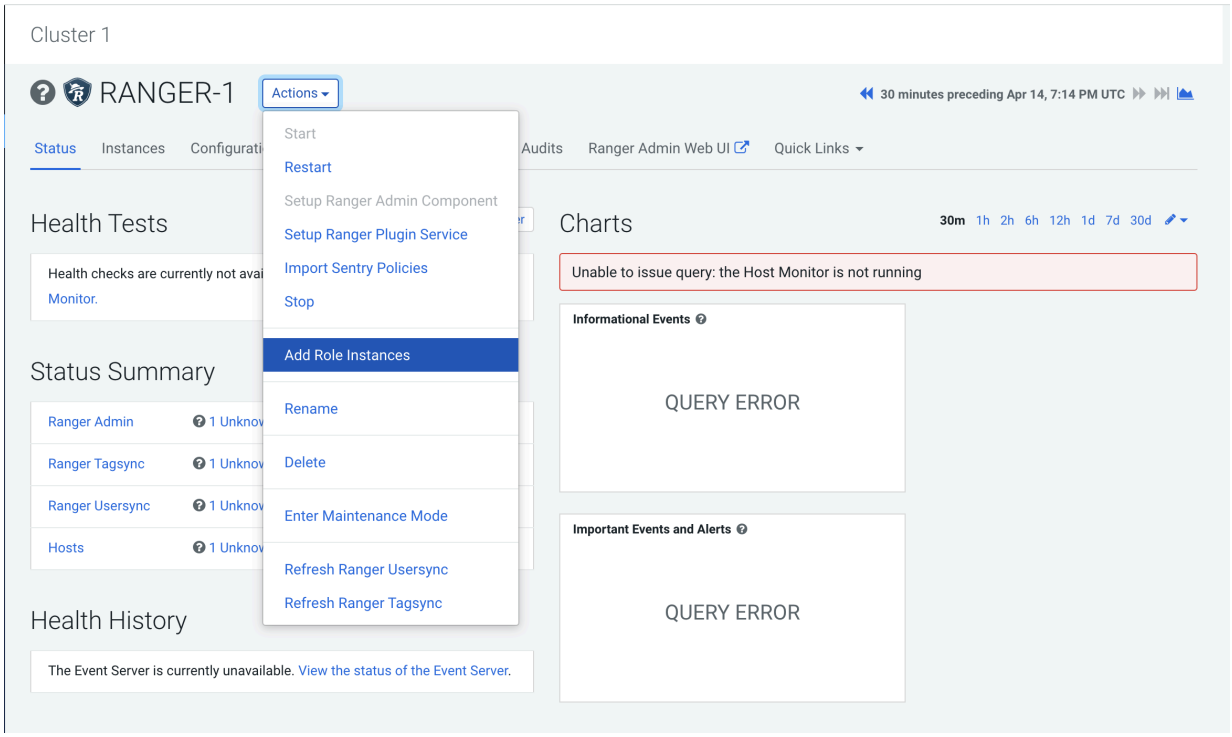


Configure Ranger Admin High Availability with a Load Balancer

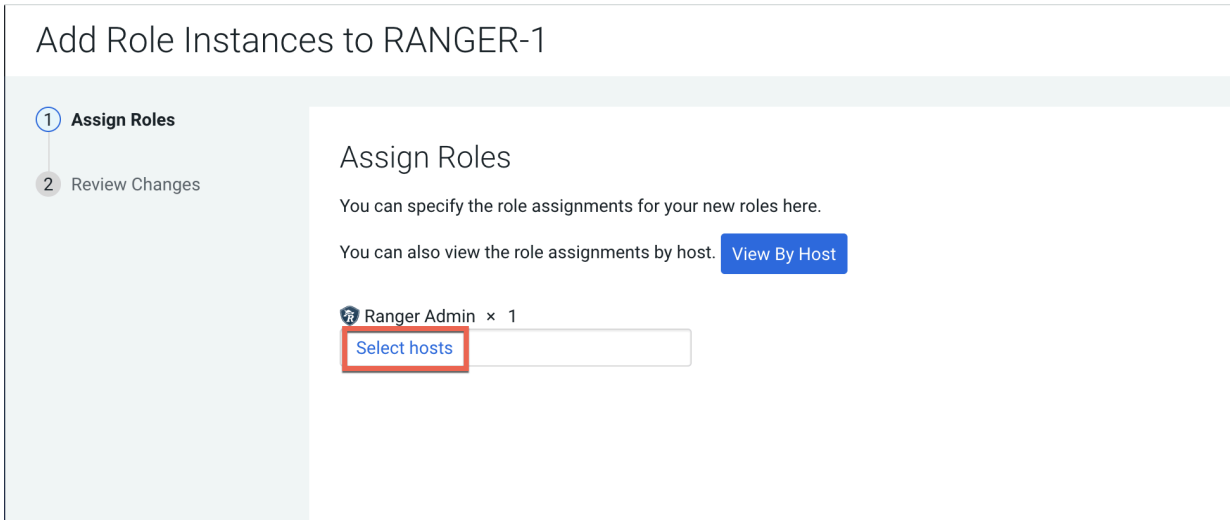
For clusters that have multiple users and production availability requirements, you may want to configure Ranger high availability (HA) with a load-balancing proxy server to relay requests to and from Ranger.

Procedure

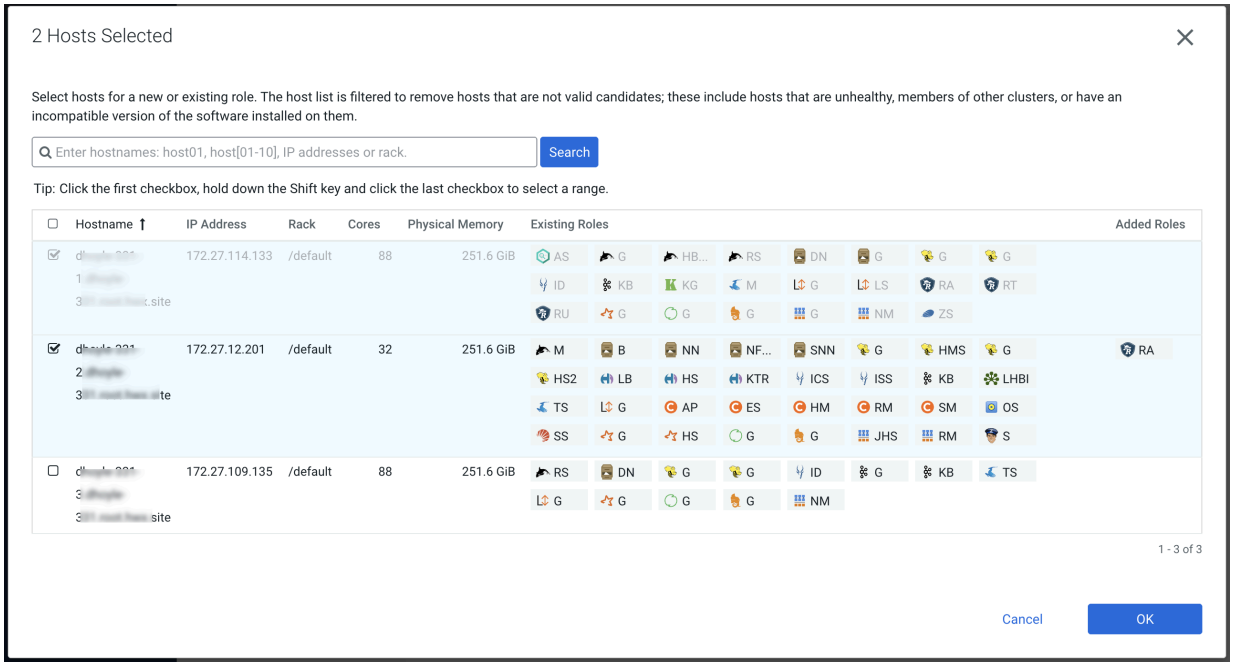
1. Configure an external load balancer to use with Ranger HA.
2. In Cloudera Manager, select Ranger, then select Actions > Add Role Instances.



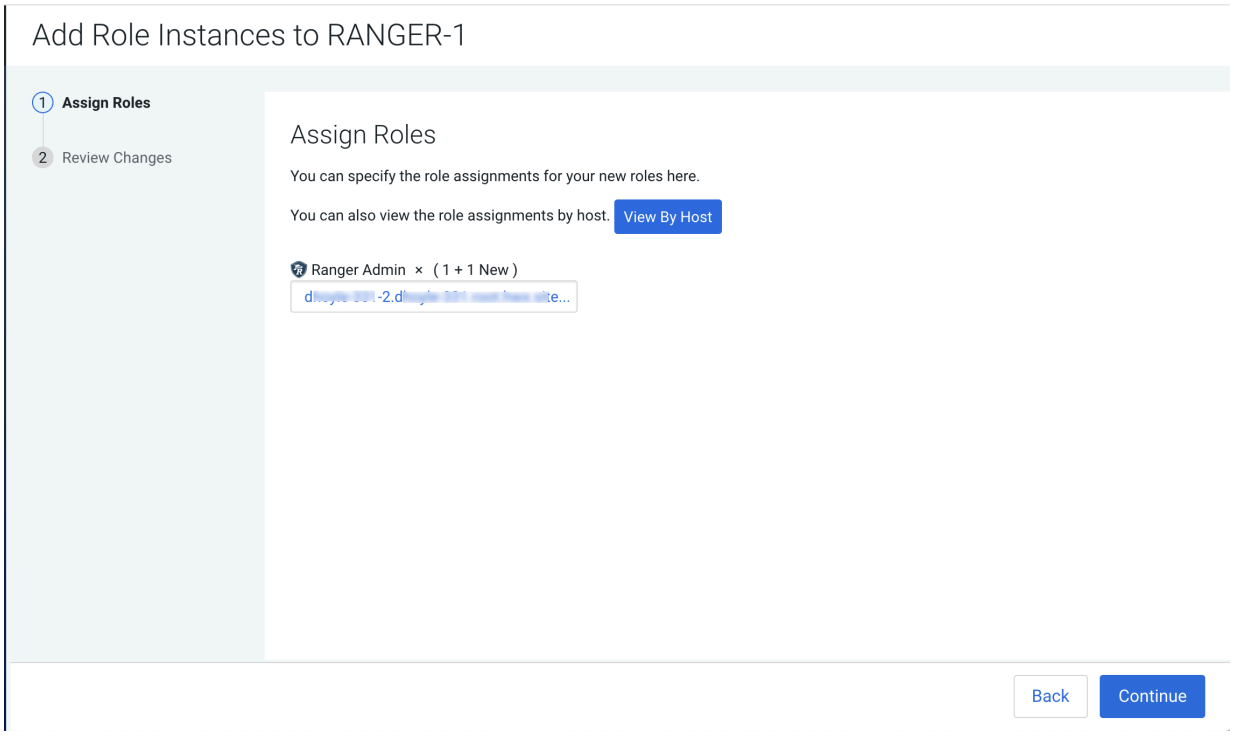
3. On the Add Role Instances page, click Select hosts.



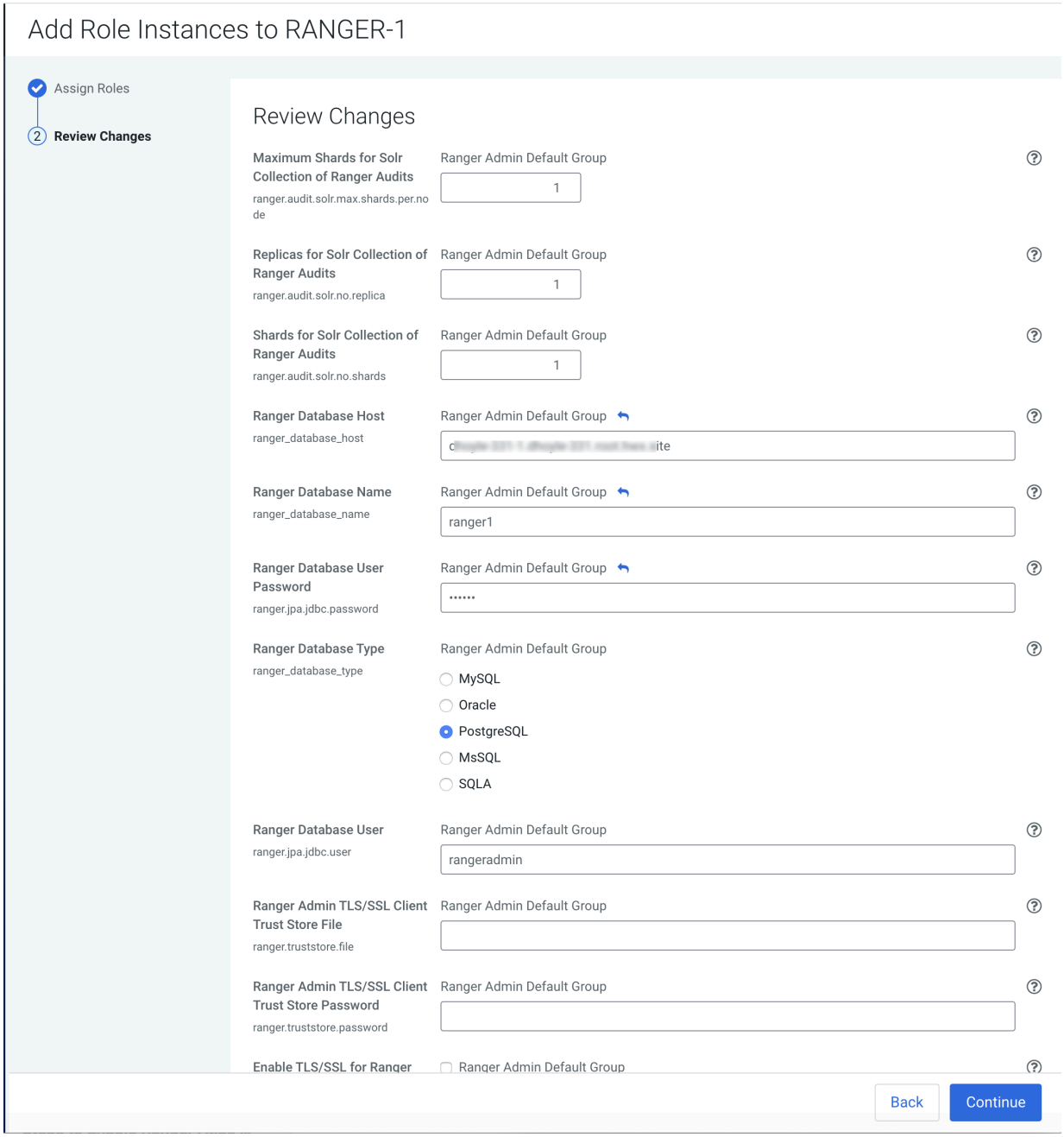
4. On the selected hosts page, the primary Ranger Admin host is selected by default. Select your configured backup Ranger host (ranger-host2-fqdn). A Ranger Admin (RA) icon appears in the Added Roles column for the selected backup host. Click OK to continue.



5. The Add Role Instances page is redisplayed with the new backup host. Click Continue.

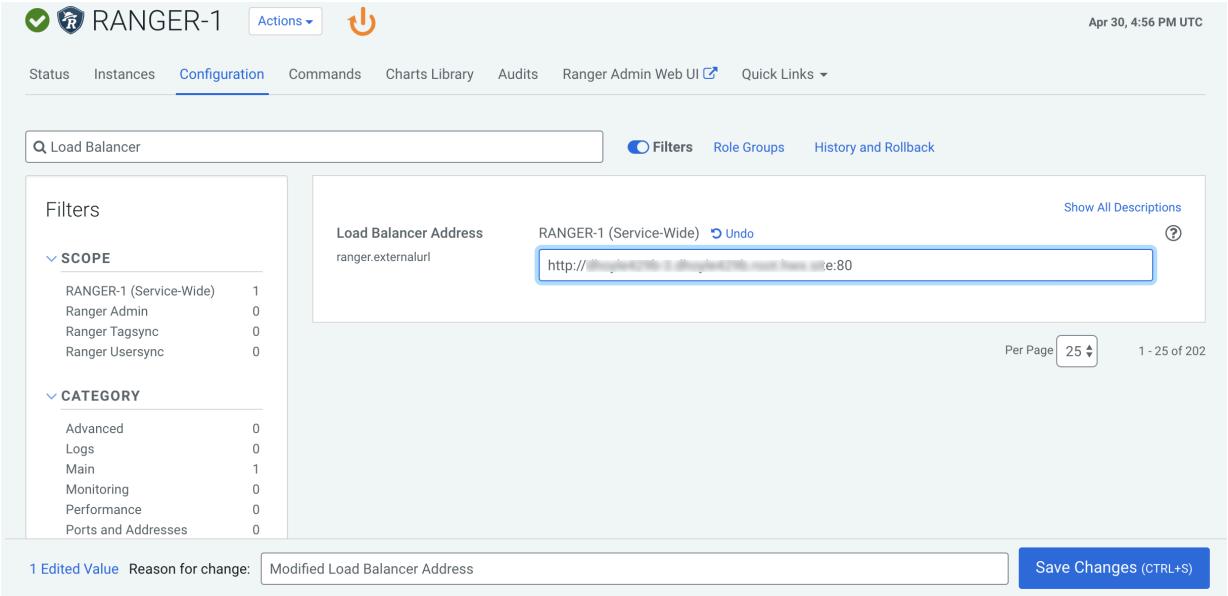


6. Review the settings on the Review Changes page, then click Continue.



7. Update the Ranger Load Balancer Address property (ranger.externalurl) with the load balancer host URL and port, then click Save Changes.

   Note: Do not use a trailing slash in the the load balancer host URL when updating the Ranger Load Balancer Address property.



8. If Kerberos is configured on your cluster, use SSH to connect to the KDC server host. Use the kadmin.local command to access the Kerberos CLI, then check the list of principals for each domain where Ranger Admin and the load-balancer are installed.

   Note: This step assumes you are using an MIT KDC (and kadmin.local). This step will be different if you are using AD or IPA.

```
kadmin.local
kadmin.local: list_principals
```

   For example, if Ranger Admin is installed on <host1> and <host2>, and the load-balancer is installed on <host3>, the list returned should include the following entries:

```
HTTP/ <host3>@EXAMPLE.COM
HTTP/ <host2>@EXAMPLE.COM
HTTP/ <host1>@EXAMPLE.COM
```

   If the HTTP principal for any of these hosts is not listed, use the following command to add the principal:

```
kadmin.local: addprinc -randkey HTTP/<host3>@EXAMPLE.COM
```

   Note:

   This step will need to be performed each time the Spnego keytab is regenerated.

9. If Kerberos is configured on your cluster, complete the following steps to create a composite keytab.

Note: These steps assume you are using an MIT KDC (and kadmin.local). These steps will be different if you are using AD or IPA.

a) SSH into the Ranger Admin host, then create a keytabs directory.

```
mkdir /etc/security/keytabs/
```

b) Copy the ranger.keytab from the current running process.

```
cp /var/run/cloudera-scm-agent/process/<current-ranger-process>/ranger.k
eytab /etc/security/keytabs/ranger.ha.keytab
```

c) Run the following command to invoke kadmin.local.

```
kadmin.local
```

d) Run the following command to add the SPNEGO principal entry on the load balancer node.

```
ktadd -norandkey -kt /etc/security/keytabs/ranger.ha.keytab HTTP/load-ba
lancer-host@EXAMPLE.COM
```

Note:

As shown above, the domain portion of the URL must be in capital letters. You can use list_principals * to view a list of all of the principals.

e) Run the following command to add the SPNEGO principal entry on the node where the first Ranger Admin is installed.

```
ktadd -norandkey -kt /etc/security/keytabs/ranger.ha.keytab HTTP/ranger-
admin-host1@EXAMPLE.COM
```

f) Run the following command to add the SPNEGO principal entry on the node where the second Ranger Admin is installed.

```
ktadd -norandkey -kt /etc/security/keytabs/ranger.ha.keytab HTTP/ranger-
admin-host2@EXAMPLE.COM
```

g) Run the following command to exit kadmin.local.

```
exit
```

h) Run the following command to verify that the /etc/security/keytabs/ranger.ha.keytab file has entries for all of the required SPNEGO principals.

```
klist -kt /etc/security/keytabs/ranger.ha.keytab
```

i) On the backup (ranger-admin-host2) Ranger Admin node, run the following command to create a keytabs folder.

```
mkdir /etc/security/keytabs/
```

j) Copy the ranger.ha.keytab file from the primary Ranger Admin node (ranger-admin-host1) to the backup (rang er-admin-host2) Ranger Admin node.

```
scp /etc/security/keytabs/ranger.ha.keytab root@ranger-host2-fqdn:/etc/s
ecurity/keytabs/ranger.ha.keytab
```

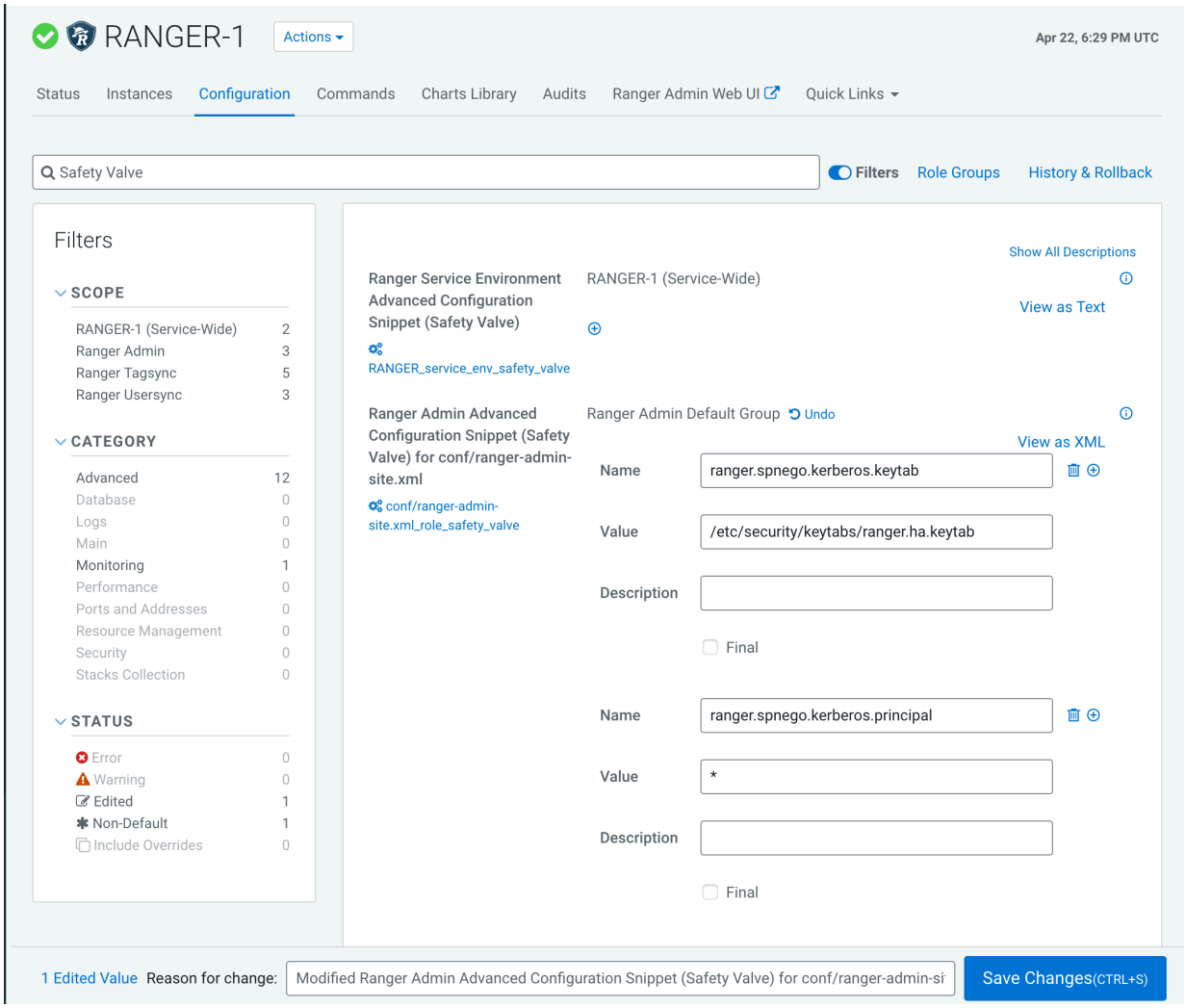
k) Run the following commands on all of the Ranger Admin nodes.

```
chmod 440 /etc/security/keytabs/ranger.ha.keytab
```

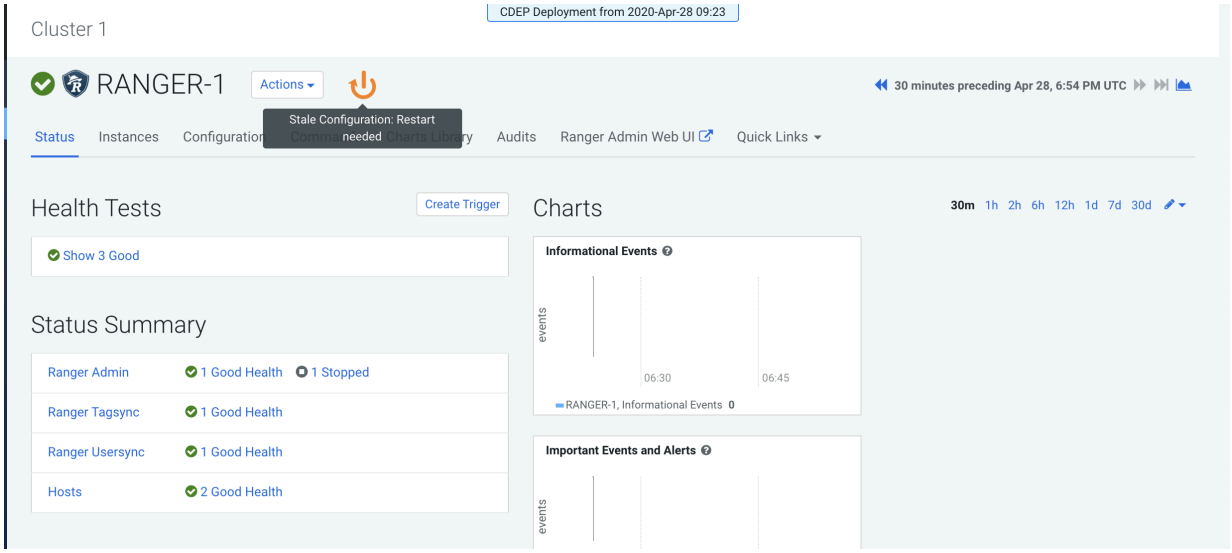
```
chown ranger:hadoop /etc/security/keytabs/ranger.ha.keytab
```

10. Update the following ranger-admin-site.xml configuration settings using the Safety Valve.

```
ranger.spnego.kerberos.keytab=/etc/security/keytabs/ranger.ha.keytab
ranger.spnego.kerberos.principal=*
```



11. Restart all cluster services that require a restart, then click Finish.



12. Use a browser to check the load-balancer host URL (with port). You should see the Ranger Admin page.