HDP Security Audit 3

Managing Auditing

Date of Publish: 2018-07-15

http://docs.hortonworks.com

Contents

Audit Overview

The Audit section covers enabling audit settings in Ambari, auditing in Ranger, and using Apache Solr for Ranger audits.

Ranger provides a centralized framework for collecting access audit history and reporting data, including filtering on various parameters. HDP enhances audit information that is captured within different components within Hadoop and provides insights through this centralized reporting capability.

Related Information

Manually Enabling Audit Settings in Ambari Clusters

Using Apache Solr for Ranger Audits

Managing Auditing in Ranger

Manually Enabling Audit Settings in Ambari Clusters

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable queries from the Ranger Admin UI. HDFS is a long-term destination for audits; audits stored in HDFS can be exported to any SIEM system, or to another audit store.

Solr and HDFS audits are generally enabled as part of the standard Ambari installation procedure. This section describes how to manually update Ambari audit settings for Solr and HDFS.

Manually Update Ambari Solr Audit Settings

You can save and store Ranger audits to Solr if you have installed and configured the Solr service in your cluster.

Procedure

1. From the Ambari dashboard, select the Ranger service. Select Configs > Advanced, then scroll down and select Advanced ranger-admin-site.
2. Set the following property value: ranger.audit.source.type = solr.
3. On the Ranger Configs tab, select Ranger Audit. The SolrCloud button should be set to ON. The SolrCloud configuration settings are loaded automatically when the SolrCloud button is set from OFF to ON, but you can also manually update the settings.

   Note:

   Audits to Solr requires that you have already “Configure Externally-Managed SolrCloud”.
4. Restart the Ranger service: service ranger-admin restart.
5. After the Ranger service has been restarted, you will then need to make specific configuration changes for each plugin to ensure that the plugin's data is captured in Solr.
6. For example, if you would like to configure HBase for audits to Solr, perform the following steps:
   a) Select the Audit to Solr checkbox in Advanced ranger-hbase-audit.
   b) Enable the Ranger plugin for HBase.
   c) Restart the HBase component.
7. Verify that the Ranger audit logs are being passed to Solr by opening one of the following URLs in a web browser:
   - http://{RANGER_HOST_NAME}:6080/index.html#!/reports/audit/bigData
   - For HDP Search's Solr Instance: http:{SOLR_HOST}:8983/solr/ranger_audits

- For Ambari Infra's Solr Instance:http:{SOLR_HOST}:8886/solr/ranger_audits

Related Information

Configure Externally-Managed SolrCloud

Manually Update HDFS Audit Settings for Ambari installs

The following steps show how to save Ranger audits to HDFS for HBase. You can use the same procedure for other components.

Procedure

1. From the Ambari dashboard, select the HBase service. On the Configs tab, scroll down and select Advanced ranger-hbase-audit. Select the Audit to HDFS check box.
2. Set the HDFS path where you want to store audits in HDFS: xasecure.audit.destination.hdfs.dir = hdfs://$NAMENODE_FQDN:8020/ranger/audit.

   Refer to the fs.defaultFS property in the Advanced core-site settings.

   Note:

   For NameNode HA, NAMENODE_FQDN is the cluster name. In order for this to work, /etc/hadoop/conf/hdfs-site.xml needs to be linked under /etc/<component_name>/conf.
3. Enable the Ranger plugin for HBase.
4. Make sure that the plugin sudo user has permission on the HDFS Path: hdfs://NAMENODE_FQDN:8020/ranger/audit.

   For example, we need to create a Policy for Resource : /ranger/audit, all permissions to user hbase.
5. Save the configuration updates and restart HBase.
6. Generate some audit logs for the HBase component.
7. Check the HFDS component logs on the NameNode: hdfs://NAMENODE_FQDN:8020/ranger/audit.

Managing Auditing in Ranger

To explore options for auditing policies in Ranger, access the Ranger console, then click **Audit** in the top menu.

There are five tabs on the Audit page:

- Access
- Admin
- Login sessions
- Plugins
- Plugin Status

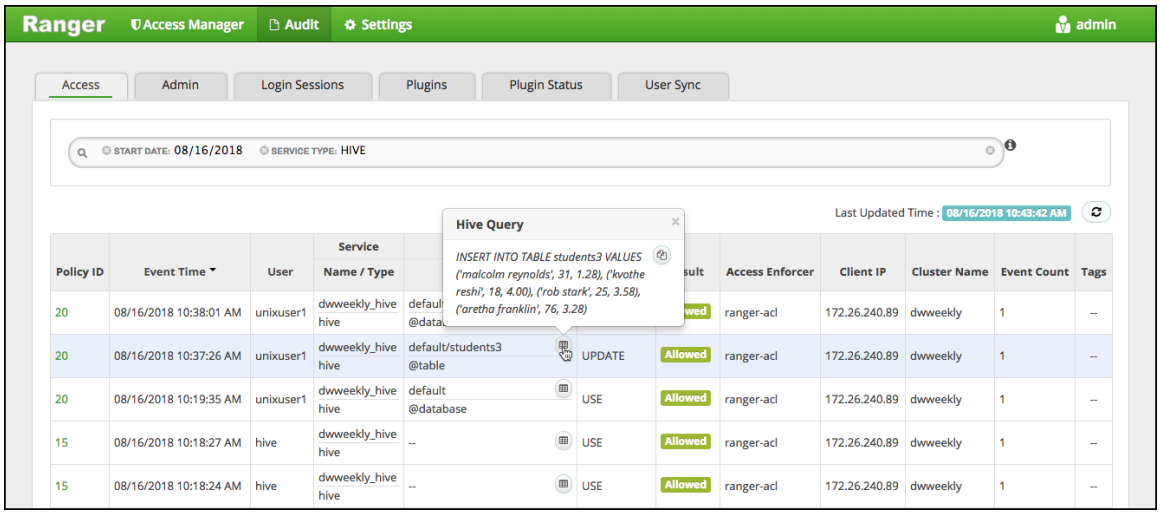
View Audit Details

How to view operation details in Ranger audits.

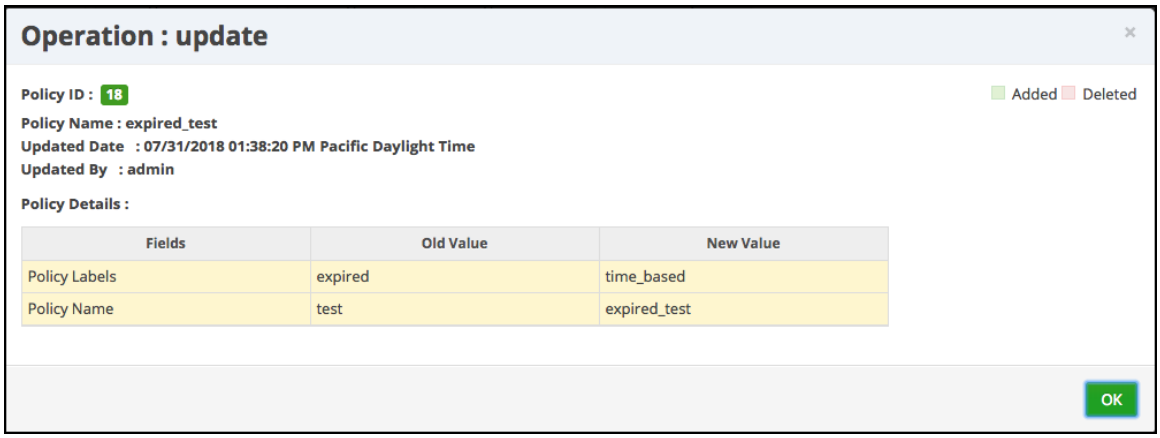
Procedure

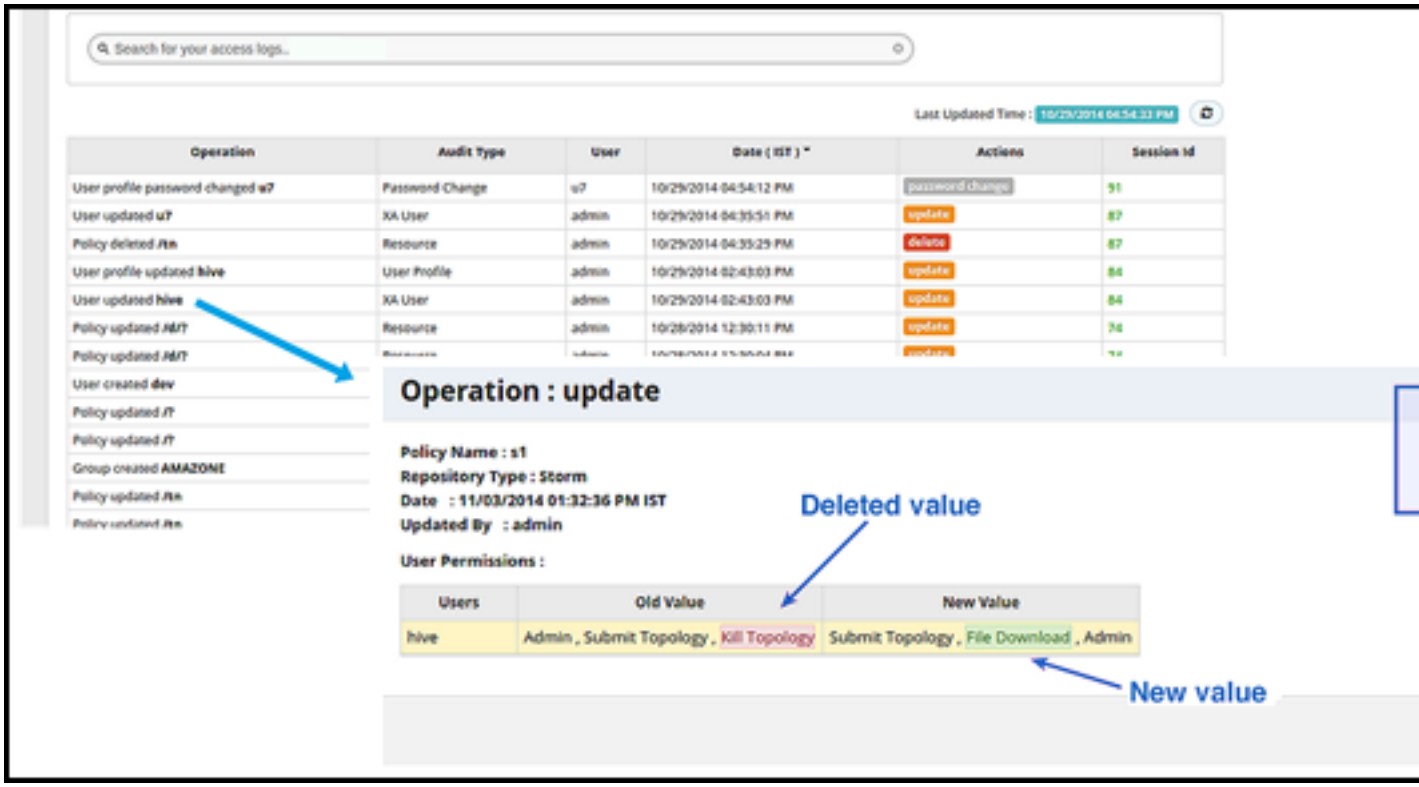
To view details on a particular operation, click any tab, then Policy ID, Operation name, or Session ID.

Audit> Access: User-run Hive query

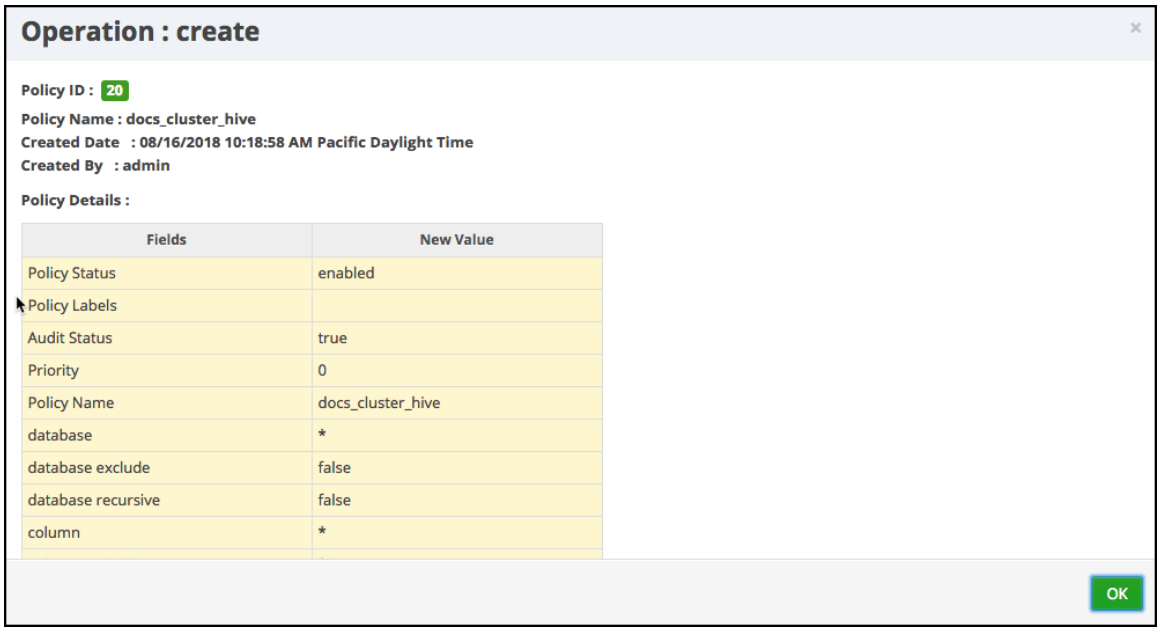


Audit> Admin: Update

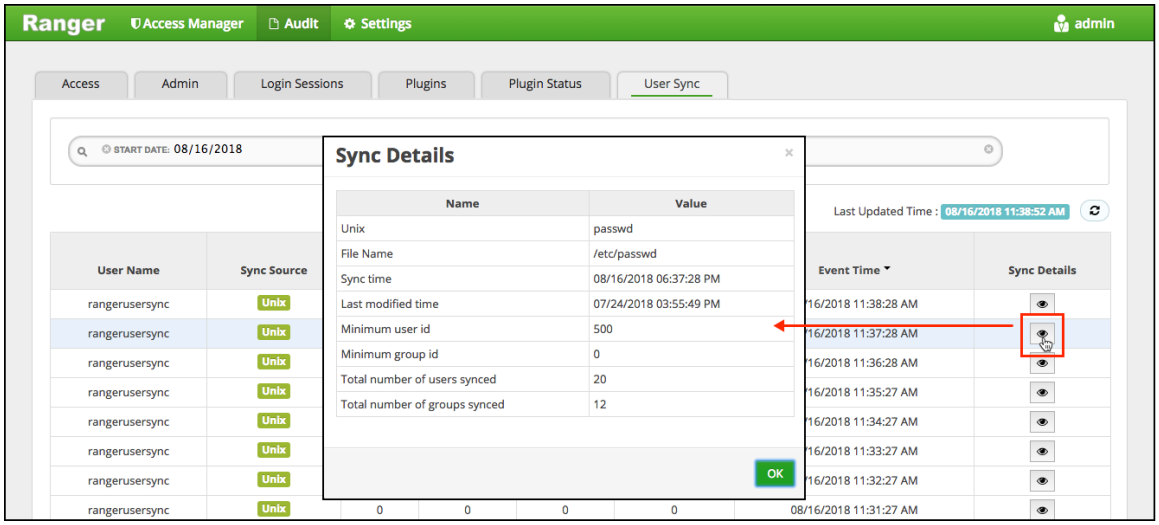




Audit> Admin: Create



Audit>User Sync: Sync details



Differentiate Events from Multiple Clusters

How to differentiate events from multiple clusters when managing audits in Ranger.

About this task

To differentiate the audit events generated by multiple ephemeral clusters that are stored in the same shared service cluster, you can define a cluster name. The cluster name is visible in Ranger>Auditing>Access and Ranger>Auditing>Plugins.

Procedure

1. From **Ambari** > **$Component** > **Configs** > **Advanced** > **ranger-$component-audit**, define a value for ranger.plugin.$component.ambari.cluster.name=$cluster_name .
2. Restart Ranger and the component.

Example

For example, in Ambari>HDFS>Configs>Advanced>ranger-hdfs-audit: ranger.plugin.hdfs.ambari.cluster.name=cluster_c6700.

Using Apache Solr for Ranger Audits

Apache Solr is an open-source enterprise search platform. Apache Ranger can use Apache Solr to store audit logs, and Solr can also to provide a search capability of the audit logs through the Ranger Admin UI.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable search queries from the Ranger Admin UI. HDFS is a long-term destination for audits -- audits stored in HDFS can be exported to any SIEM system, or to another audit store.

Apache Ranger uses Apache Solr to store audit logs and provides UI searching through the audit logs. Solr must be installed and configured before installing Ranger Admin or any of the Ranger component plugins. The default configuration for Ranger Audits to Solr uses the shared Solr instance provided under the Ambari Infra service. Solr is both memory and CPU intensive. If your production system has high volume of access requests, make sure that the Solr host has adequate memory, CPU, and disk space.

SolrCloud is the preferred setup for production usage of Ranger. SolrCloud, which is deployed with the Ambari Infra service, is a scalable architecture that can run as a single node or multi-node cluster. It has additional features such as

replication and sharding, which is useful for high availability (HA) and scalability. You should plan your deployment based on your cluster size. Because audit records can grow dramatically, plan to have at least 1 TB of free space in the volume on which Solr will store the index data. Solr works well with a minimum of 32 GB of RAM. You should provide as much memory as possible to the Solr process.

It is highly recommended to use SolrCloud with at least two Solr nodes running on different servers with replication enabled. You can use the information in this section to configure additional SolrCloud instances.

Configuration Options

- Ambari Infra Managed Solr (default) -- Audits to Solr defaults to use the shared Solr instance provided under the Ambari Infra service. There are no additional configuration steps required for this option. SolrCloud, which is deployed with the Ambari Infra service, is a scalable architecture which can run as a single node or multi-node cluster. This is the recommended configuration for Ranger. By default, a single-node SolrCloud installation is deployed when the Ambari Infra Service is chosen for installation. Hortonworks recommends that you install multiple Ambari Infra Solr Instances in order to provide distributed indexing and search for Atlas, Ranger, and LogSearch (Technical Preview). This can be accomplished by simply adding additional Ambari Infra Solr Instances to existing cluster hosts by selecting Actions > Add Service on the Ambari dashboard.
- Externally Managed SolrCloud -- You can also install and manage an external SolrCloud that can run as single or multi-node cluster. It includes features such as replication and sharding, which are useful for high availability (HA) and scalability. With SolrCloud, customers need to plan the deployment based on the cluster size.
- Externally Managed Solr Standalone -- Solr Standalone is NOT recommended for production use, and should be only used for testing and evaluation. Solr Standalone is a single instance of Solr that does not require ZooKeeper.

  Attention:

  Solr Standalone is NOT recommended and support for this configuration will be deprecated in a future release.
- SolrCloud for Kerberos -- This is the recommended configuration for SolrCloud in Kerberos environments.

Related Information

Using Ambari Core Services>Ambari Infra

Solr Reference Guide > SolrCloud

Solr Reference Guide > Cross Data Center Replication (CDCR)

Using Apache Solr for Ranger Audits: Prerequisites

Your cluster must meet some prerequisites before it can use Apahce Solr for Ranger audits.

Solr Prerequisites

- Ranger supports Apache Solr 5.2 or higher.
- Apache Solr requires the Java Runtime Environment (JRE) version 1.7 or higher.
- 1 TB free space in the volume where Solr will store the index data.
- 32 GB RAM.

SolrCloud Prerequisites

- SolrCloud supports replication and sharding. It is highly recommended that you use SolrCloud with at least two Solr nodes running on different servers with replication enabled.
- SolrCloud requires Apache ZooKeeper.
- SolrCloud with Kerberos requires Apache ZooKeeper and MIT Kerberos.

Migrating Audit Logs from DB to Solr in Ambari Clusters

How to migrate your audit logs from DB to Solr. It is recommended that you store audits in both HDFS and Solr.

Before you begin

Before you migrate your audit logs from DB to Solr, make sure your cluster meets the following requirements:

- Solr must be installed and running (see “Using Apache Solr for Ranger Audits”).
- All plug-ins must be upgraded and writing audit logs to Solr (i.e., plugins must not be writing audit logs to DB.)
- The DB server must be running, and the credentials used to connect Audit to DB must be available.
- Ranger must be running with the audit source as Solr, and the Solr URL must be configured.

About this task

Audit to DB is no longer recommended and the option is disabled in the Ambari UI. If your logs were previously stored on DB, you can migrate the logs to Solr.

Procedure

1. Configure the properties ranger.audit.source.type and ranger.audit.solr.urls:

Property Name	Sample Value	Location
ranger.audit.source.type	solr	Ranger>Configs>Advanced>Advanced ranger-admin-site
ranger.audit.solr.urls	Syntax: http://<solr_host>:<port>/ solr/ranger_audits Example: http://192.168.0.2:8983/ solr/ranger_auditsExample: http://192.168.0.2:8886/solr/ranger_audits	Ranger>Configs>Ranger Audit

2. Verify or enter the ranger.jpa.audit.jdbc.url value.
3.
4. After upgrading Ranger and changing the audit log destination from DB to Solr, Ambari may not automatically populate the required property values. If necessary, you can add these as custom properties from Ambari.
   a) Select Ranger>Configs>Advanced>Custom ranger-admin-site, then click **Add Property...**.
   b) Enter the following information on the **Add Property** pop-up:
      - Type: preloaded with the value ranger-admin-site.xml
      - Key: enter ranger.jpa.audit.jdbc.url
      - Value: enter the JDBC audit string for your DB platform:

Table 1: JDBC Audit String

DB Platform	Syntax	Example Value
MySQL	jdbc:mysql://DB_HOST:PORT/audit_name	jdbc:mysql:// c6401.ambari.apache.org:3306/ ranger_audit
Oracle	For Oracle SID: jdbc:oracle:thin:@AUDIT_HOST:PORT:SID	jdbc:oracle:thin:@c6401.ambari.apache.org:1521:ORCL
	For Oracle Service Name: jdbc:oracle:thin:@//AUDIT_HOST[:PORT] [/ServiceName]	jdbc:oracle:thin:@// c6401.ambari.apache.org:1521/XE
PostgreSQL	jdbc:postgresql://AUDIT_HOST/audit_name	jdbc:postgresql:// c6401.ambari.apache.org:5432/ ranger_audit
MS SQL	jdbc:sqlserver://AUDIT_HOST;databaseName=audit_name	jdbc:sqlserver:// c6401.ambari.apache.org:1433;databaseName=ranger_audit
SQLA	jdbc:sqlanywhere:host=AUDIT_HOST;database=audit_name	jdbc:sqlanywhere:host=c6401.ambari.apache.org:2638;databas

5. Restart Ranger Admin: service ranger-admin restart.

6. Navigate to the Ranger admin directory and run the following command:

```
$/path/to/java -Dlogdir=ews/logs -Dlog4j.configuration=db_patch.log4j.xml
 -cp ews/webapp/WEB-INF/classes/conf:ews/webapp/WEB-INF/classes/
lib/*:ews/webapp/WEB-INF/:ews/webapp/META-INF/:ews/webapp/
WEB-INF/lib/*:ews/webapp/WEB-INF/classes/:ews/webapp/WEB-INF/
classes/META-INF:/usr/share/java/mysql-connector-java.jar
 org.apache.ranger.patch.cliutil.DbToSolrMigrationUtil
```

If the script succeeds, it prints the following details on the screen:

- Processing batch 'n' of total 'noOfBatches' (Where each batch contains 10000 rows.)
- Total number of migrated audit logs.

If the script fails to migrate data, it returns the error: Migration process failed, Please refer ranger_db_patch.log file.

Related Information

Using Apache Solr for Ranger Audits

Install Externally-Managed SolrCloud

How to install externally-managed SolrCloud when using Apache Solr for Ranger audits.

About this task

You can download the Solr package from Apache Solr Downloads (link below). Make sure that the Solr version is 5.2 or above. You can also use the Ranger setup.sh script to automatically download, install, and configure Solr. If you would like to use the setup.sh script to install Solr, set the following properties in the install.properties files, along with the settings from the one of the configuration options in the following sections.

Procedure

1. Run the following commands:

```
cd $HOME
git clone https://github.com/apache/incubator-ranger.git
cd incubator-ranger/security-admin/contrib/solr_for_audit_setup
```

2. Edit the install.properties file (see the instructions in the following sections).

Table 2: Solr install.properties Values for setup.sh script

Property Name	Value	Description
SOLR_INSTALL	true	When set to true, hesetup.sh script will download the Solr package and install it.
SOLR_DOWNLOAD_URL	http://archive.apache.org/dist/lucene/solr/5.2.1/solr-5.2.1.tgz	It is recommended that you use one for Apache mirror sites to download the Solr package. You can choose a mirror site at http://lucene.apache.org/solr/mirrors-solr-latest-redir.html
SOLR_INSTALL_FOLDER	/opt/solr	The Solr install folder.

3. Run the ./setup.sh script.
4. Refer to $SOLR_RANGER_HOME/install_notes.txt for additional instructions.

Related Information

Apache Solr Downloads

Configure Externally-Managed SolrCloud

How to configure externally-managed SolrCloud when using Apache Solr for Ranger audits.

Procedure

1. Open the install.properties file in the vi text editor. vi install.properties.
2. Set the following property values, then save the changes to the install.properties file.

Table 3: Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: /usr/jdk64/jdk1.8.0_40	Provide the path to the JDK install folder. For Hadoop, you can check/etc/hadoop/conf/hadoop-env.sh for the value of JAVA_HOME. As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	solr	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	/opt/lucidworks-hdpsearch/solr	The Solr installation directory.
SOLR_RANGER_HOME	/opt/solr/ranger_audit_server	The location where the Ranger-related configuration and schema files will be copied.
SOLR_RANGER_PORT	For HDP Search's Solr Instance: 8983 For Ambari Infra's Solr Instance: 8886	The Solr port for Ranger.
SOLR_DEPLOYMENT	solrcloud	The deployment type.
SOLR_ZK	<ZooKeeper_host>:2181/ranger_audits	The Solr ZooKeeper host and port. It is recommended to provide a sub-folder to create the Ranger Audit related configurations so you can also use ZooKeeper for other Solr instances. Due to a Solr bug, if you are using a path (sub-folder), you can only specify one ZooKeeper host.
SOLR_SHARDS	1	If you want to distribute your audit logs, you can use multiple shards. Make sure the number of shards is equal or less than the number of Solr nodes you will be running.
SOLR_REPLICATION	1	It is highly recommend that you set up at least two nodes and replicate the indexes. This gives redundancy to index data, and also provides load balancing of Solr queries.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder for the Solr log files.
SOLR_MAX_MEM	2g	The memory allocation for Solr.

3. Run the set up script: ./setup.sh.
4. Run the following command only once from any node. This command adds the Ranger Audit configuration (including schema.xml) to ZooKeeper: /opt/solr/ranger_audit_server/scripts/add_ranger_audits_conf_to_zk.sh.
5. Log in as the solr or root user and run the following command to start Solr on each node: /opt/solr/ranger_audit_server/scripts/start_solr.sh.
   When Solr starts, a confirmation message appears. Started Solr server on port 8983/8886 (pid=). Happy searching!
6. Run the following command only once from any node. This command creates the Ranger Audit collection. /opt/solr/ranger_audit_server/scripts/create_ranger_audits_collection.sh.
7. You can use a web browser to open the Solr Admin Console at the following address:
   a) For HDP Search's Solr Instance: http:<solr_host>:8983/solr.

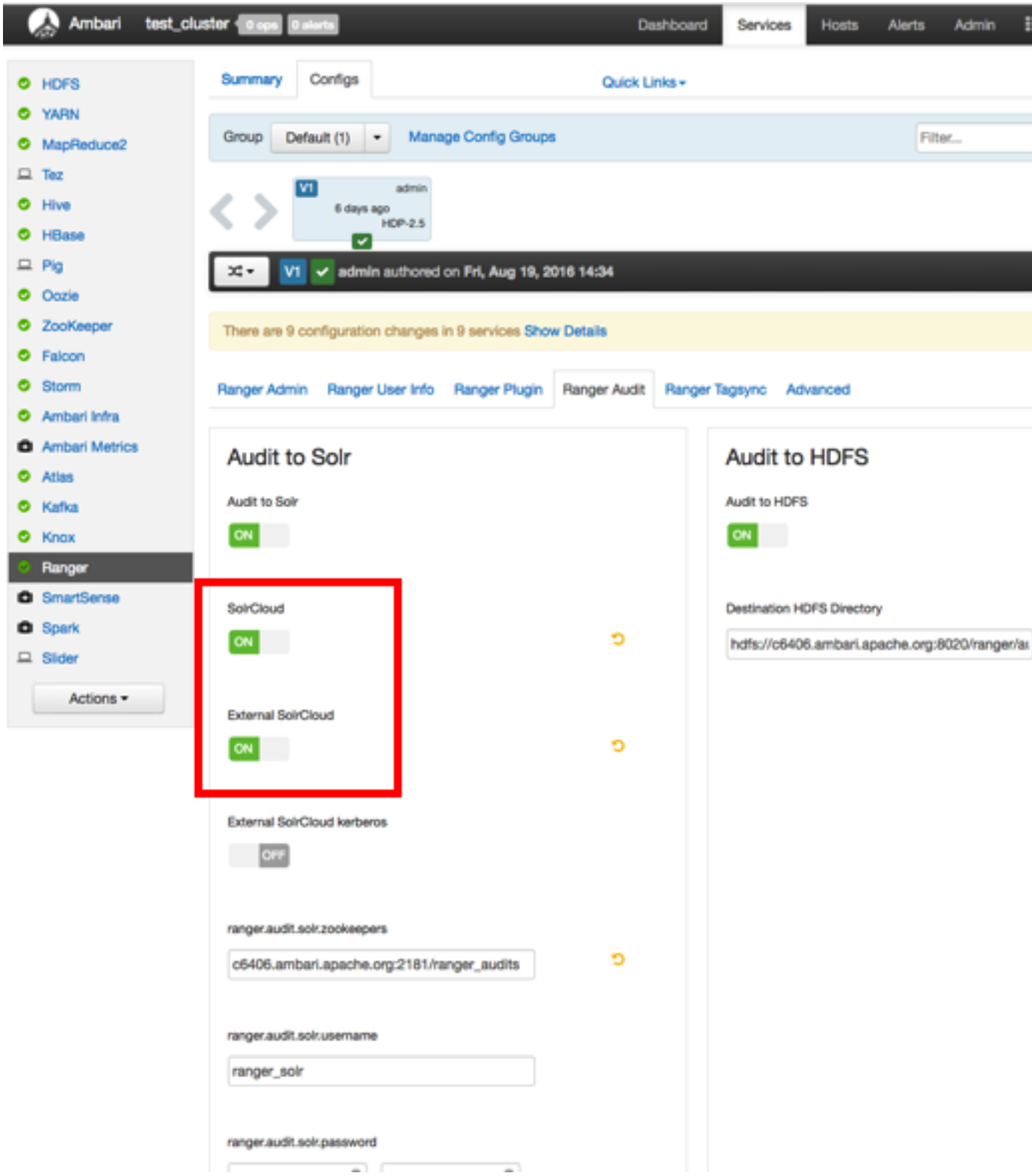
b) For Ambari Infra's Solr Instance:http:<solr_host>:8886/solr.

Note:

You can use the following command to stop Solr:

```
/opt/solr/ranger_audit_server/scripts/stop_solr.sh
```

8. On the Ambari dashboard, select Ranger > Configs > Ranger Audit, then enable SolrCloud and External SolrCloud by clicking the OFF buttons. The button labels change to ON when SolrCloud and External SolrCloud are enabled.



9. Set the value of the ranger.audit.solr.zookeepers property to <host_name>:2181/ranger_audits.
10. Select Ranger > Configs > Advanced, then select Advanced ranger-env and set the following properties:
    a) ranger_solr_replication_factor – set this to the same value used in the install.properties file.
    b) ranger_solr_shards – set this to the same value used in the install.properties file.
11. Click Save, then restart Ranger and all required services.

Configure Externally-Managed Solr Standalone

How to configure externally-managed Solr Standalone when using Apache Solr for Ranger audits.

About this task

Attention:

This configuration is NOT recommended for new installs of HDP-2.5 and is intended for non-production use. Support for this configuration will be deprecated in a future release.

Procedure

1. Open the install.properties file in the vi text editor. vi install.properties.
2.
3. Set the following property values, then save the changes to the install.properties file.

Table 4: Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: /usr/jdk64/jdk1.8.0_60	Provide the path to the JDK install folder. For Hadoop, you can check/etc/hadoop/conf/hadoop-env.sh for the value of JAVA_HOME. As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	solr	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	/opt/solr	The Solr installation directory.
SOLR_RANGER_HOME	/opt/solr/ranger_audit_server	The location where the Ranger-related configuration and schema files will be copied.
SOLR_RANGER_PORT	For HDP Search's Solr Instance: 8983	The Solr port for Ranger.
SOLR_DEPLOYMENT	standalone	The deployment type.
SOLR_RANGER_DATA_FOLDER	/opt/solr/ranger_audit_server/data	The folder where the index data will be stored. The volume for this folder should have at least 1 TB free space for the index data, and should be backed up regularly.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder for the Solr log files.
SOLR_MAX_MEM	2g	The memory allocation for Solr.

4. Run the Solr for Ranger setup script. ./setup.sh.
5. To start Solr, log in as the solr or root user and run the following command. /opt/solr/ranger_audit_server/scripts/start_solr.sh.
   When Solr starts, a confirmation message appears. Started Solr server on port 8983/8886 (pid=). Happy searching!
6. You can use a web browser to open the Solr Admin Console at the following address:
   a) For HDP Search's Solr Instance: http:<solr_host>:8983/solr.

Configuring SolrCloud for Kerberos

How to configure SolrCloud for Kerberos when using Apache Solr for Ranger Audits.

Configure Kerberos for SolrCloud

How to configure Kerberos for SolrCloud.

Procedure

1. Create a principal "solr" in your KDC. You can make it host-specific or headless.
2. Log in as the root user to the KDC server and create the keytabs for users "solr" and HTTP.

```
kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local:  addprinc -randkey solr@EXAMPLE.COM
WARNING: no policy specified for solr@EXAMPLE.COM; defaulting to no policy
Principal "solr@EXAMPLE.COM" created.
kadmin.local:  xst -k solr.service.keytab solr@EXAMPLE.COM
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type aes256-
cts-hmac-sha1-96 added to keytab WRFILE:solr.service.keytab.
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type aes128-
cts-hmac-sha1-96 added to keytab WRFILE:solr.service.keytab.
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type des3-
cbc-sha1 added to keytab WRFILE:solr.service.keytab.
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type arcfour-
hmac added to keytab WRFILE:solr.service.keytab.
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type des-
hmac-sha1 added to keytab WRFILE:solr.service.keytab.
Entry for principal solr@EXAMPLE.COM with kvno 2, encryption type des-cbc-
md5 added to keytab WRFILE:solr.service.keytab.
kadmin.local:  quit
```

The example above creates a headless keytab for the "solr" service user. You should create one keytab per host. You should also create a principal for each host on which Solr is running. Use the procedure shown above, but use the principal name with the host. For example:

```
kadmin.local:  addprinc -randkey solr/<SOLR_HOST_NAME>@EXAMPLE.COM
```

You will also need another keytab for Spnego. This is used by Solr to authenticate HTTP requests. Follow the process shown above, but replace "solr" with "HTTP". For example:

```
kadmin.local
kadmin.local:  addprinc -randkey HTTP@EXAMPLE.COM
kadmin.local:  xst -k HTTP.keytab HTTP@EXAMPLE.COM
kadmin.local:  quit
```

3. After the keytabs are created, run the following commands to copy them to all of the hosts running Solr, chown to "solr", and chmod to 400.

```
mkdir -p /opt/solr/conf
#scp both the keytab files to the above folder
chown solr:solr /opt/solr/conf/solr.service.keytab
usermod -a -G hadoop solr
chmod 400 /opt/solr/conf/solr.service.keytab
chown solr:solr /opt/solr/conf/HTTP.keytab
chmod 400 /opt/solr/conf/HTTP.keytab
```

Note:

The usermod -a -G hadoop solr command is required if you are using the HTTP (Spnego) keytab that is generated by Ambari.

Configure SolrCloud for Kerberos

How to configure SolrCloud for Kerberos.

Procedure

1. Run the following commands:

```
cd /opt/solr
mkdir /opt/solr/conf
```

2. Create a new JAAS file in the /opt/solr/conf directory: vi /opt/solr/conf/solr_jaas.conf.
3. Add the following lines to the solr_jaas.conf file, but replace the REALM name @EXAMPLE.COM with your REALM.

```
Client {
       com.sun.security.auth.module.Krb5LoginModule required
       useKeyTab=true
       keyTab="/opt/solr/conf/solr.service.keytab"
       storeKey=true
       useTicketCache=true
       debug=true
       principal="solr@EXAMPLE.COM";
};
```

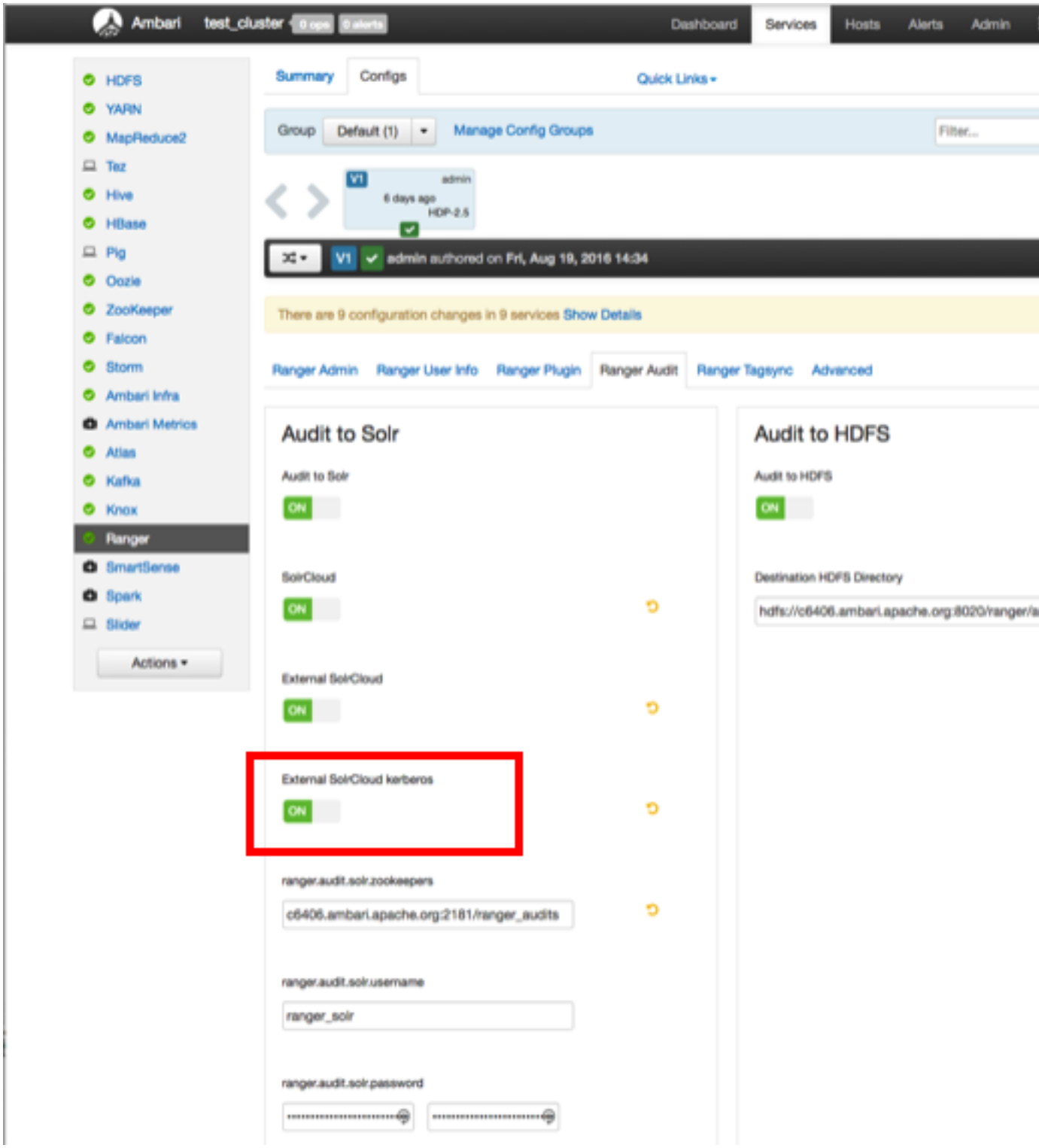
4. Copy the solr_jaas.conf file to all of the hosts on which Solr is running.
5. Edit the solr.in.sh file in the <SOLR_INSTALL_HOME>/bin/ directory:vi $SOLR_INSTALL_HOME/ranger_audit_server/scripts/solr.in.sh.
6. Add the following lines at the end of the solr.in.sh file:

```
SOLR_JAAS_FILE=/opt/solr/conf/solr_jaas.conf
SOLR_HOST=`hostname -f`
ZK_HOST="$ZK_HOST1:2181,$ZK_HOST2:2181,$ZK_HOST3:2181/ranger_audits"
KERBEROS_REALM="EXAMPLE.COM"
SOLR_KEYTAB=/opt/solr/conf/solr.service.keytab
SOLR_KERB_PRINCIPAL=HTTP@${KERBEROS_REALM}
SOLR_KERB_KEYTAB=/opt/solr/conf/HTTP.keytab
SOLR_AUTHENTICATION_CLIENT_CONFIGURER="org.apache.solr.client.solrj.impl.Krb5HttpClie
SOLR_AUTHENTICATION_OPTS=" -
DauthenticationPlugin=org.apache.solr.security.KerberosPlugin
 -Djava.security.auth.login.config=$SOLR_JAAS_FILE -
Dsolr.kerberos.principal=${SOLR_KERB_PRINCIPAL}
 -Dsolr.kerberos.keytab=${SOLR_KERB_KEYTAB} -Dsolr.kerberos.cookie.domain=
${SOLR_HOST} -Dhost=${SOLR_HOST}
 -Dsolr.kerberos.name.rules=DEFAULT"
```

7. Copy the solr.in.sh file to all of the hosts on which Solr is running.
8. Run the following command to enable Kerberos as the authentication scheme by updating the security.json file in ZooKeeper.$SOLR_INSTALL_HOME/server/scripts/cloud-scripts/zkcli.sh -zkhost $ZK_HOST:2181 -cmd put /ranger_audits/security.json '{"authentication":{"class": "org.apache.solr.security.KerberosPlugin"}}'.
9. Run the following commands to restart Solr on all hosts:

```
/opt/solr/ranger_audit_server/scripts/stop_solr.sh
/opt/solr/ranger_audit_server/scripts/start_solr.sh
```

10. On the Ambari dashboard, select Ranger > Configs > Ranger Audit, then enable External SolrCloud Kerberos by clicking the OFF button. The button label changes to ON when External SolrCloud Kerberos is enabled.



11. Click Save, then restart Ranger and all required services.

Connecting to Kerberos enabled SolrCloud

How to connect to Kerberos-enabled Solr from your local machine.

Procedure

1. On both Linux and Mac, copy the /etc/krb5.conf file from the Solr host to your local /etc/krb5.conf. If you already have a local /etc/krb5.conf file, merge the two files.
2. Log in to the KDC host as root and run the following commands to create a KDC user:

```
kadmin.local
kadmin.local: addprinc $USERNAME@EXAMPLE.COM
kadmin.local: quit
```

3. Run the following command on your local machine: kinit $USERNAME@EXAMPLE.COM.
4. You can now use a browser to connect to the Solr URL.

Create Read-Only Admin User (Auditor)

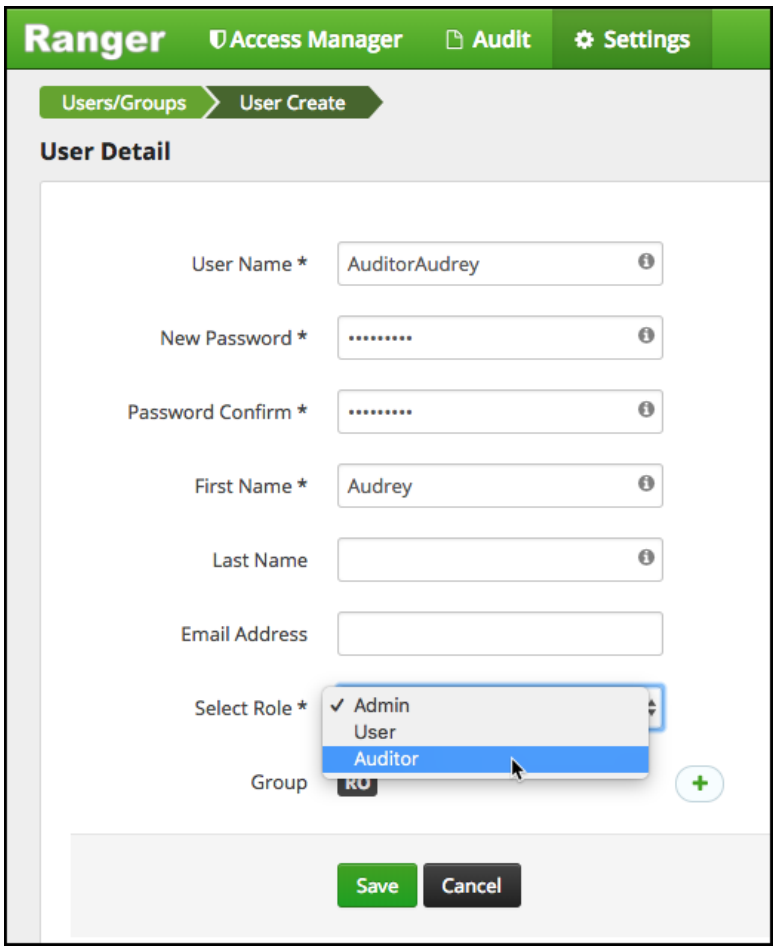
Creating a read-only admin user (auditor) enables compliance activities because the user can monitor policies and audit events without making changes.

About this task

When a user with the role “Auditor” logs in, they will see a read-only view of Ranger policies and audit events. That user can search and filter on access audit events, and access and view all tabs under Audit to understand access events. They cannot edit users or groups, export/import policies, or make changes of any kind.

Procedure

1. From the Ranger console, click **Settings > Users/Groups**.
2. Click **Add New User**.
3. Complete the **User Detail** section, selecting **Auditor** as the role:


Ranger
Access Manager
Audit
Settings
Users/Groups
User Create
User Detail
User Name *
AuditorAudrey
New Password *
Password Confirm *
First Name *
Audrey
Last Name
Email Address
Select Role *
Admin
User
Auditor
Group
Save
Cancel

4. Click **Save**.