Data Analytics Studio Installation

Date of Publish: 2019-10-04

https://docs.hortonworks.com

Contents

Installation roadmap..... 3

Support requirements.....5

System requirements..... 7

Version recommendation for DAS.....7

Understanding and obtaining DAS binaries.....8

Pre-installation tasks..... 9

Install the prerequisite components and services.....9
Check cluster configuration for Hive and Tez.....10
Create and set up a local repository..... 10
Prepare the web server for the local repository.....10
Setting up the local repository..... 11
Creating the repository configuration file.....12
Configure PostgreSQL database.....13
For CentOS..... 13
For Debian..... 14
For Ubuntu.....15

Installing Data Analytics Studio..... 16

Installing the DAS cluster components.....16
Adding the DAS service through the Ambari UI.....17

Post-installation tasks.....18

Setting up the tmp directory.....18
Additional configuration tasks.....18
(Recommended) Configuring DAS for SSL/TLS.....18
Set up trusted CA certificate.....19
Set up self-signed certificates.....19
Configure SSL/TLS in Ambari..... 20
Configuring user authentication..... 21
Configuring user authentication using Knox SSO.....21
Configuring user authentication using Knox proxy.....22
Configuring user authentication using SPNEGO.....24
Configuring Knox SSO for HA clusters.....25
Enabling logout option for secure clusters..... 26

Installation roadmap

The DAS cluster components comprise of the DAS Webapp and a DAS Event Processor. These components are installed on the Ambari cluster to use with DAS.

To install DAS/DAS-Lite, review the installation roadmap and follow the steps. We strongly encourage you to read the support requirements and the pre-installation tasks before you start the installation.

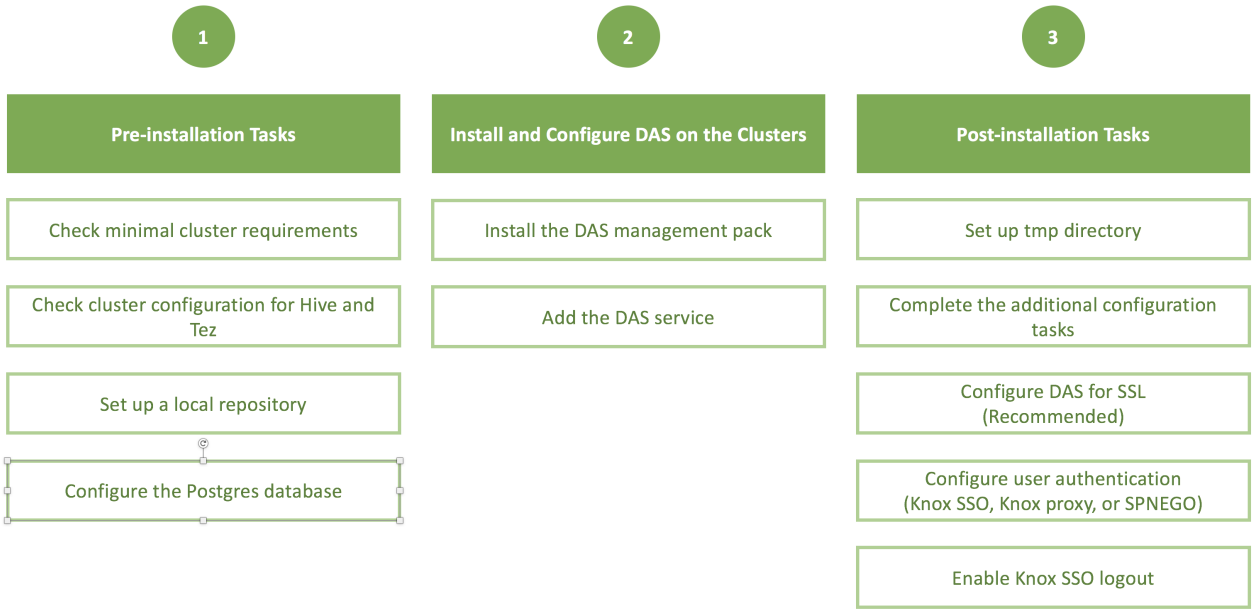

1
Pre-installation Tasks
Check minimal cluster requirements
Check cluster configuration for Hive and Tez
Set up a local repository
Configure the Postgres database
2
Install and Configure DAS on the Clusters
Install the DAS management pack
Add the DAS service
3
Post-installation Tasks
Set up tmp directory
Complete the additional configuration tasks
Configure DAS for SSL (Recommended)
Configure user authentication (Knox SSO, Knox proxy, or SPNEGO)
Enable Knox SSO logout

Table 1: DAS/DAS-Lite installation roadmap

No.	Task	Description
Pre-installation tasks		
1	Check the minimal cluster requirements.	To install and set up the necessary components and services, see Install the prerequisite components and services.
2	Check the cluster configuration for Hive and Tez.	Check the configuration settings for Hive and Tez in Ambari. See Check cluster configuration for Hive and Tez.
3	Create and set up a local repository.	Hortonworks does not host any public repository for DAS. Therefore, you need to setup a local repository to install the binaries. See Create and set up a local repository.
4	Configure the PostgreSQL database.	DAS requires a PostgreSQL database for storing query event information. See Configure PostgreSQL database.
Installing and configuring DAS on the HDP clusters		
5	Install the DAS management pack.	Install DAS cluster components (the DAS Webapp and the DAS Event Processor) on the same machine on which you have installed the Ambari server, using an Ambari management pack (MPack). See Installing the DAS cluster components.
6	Add the DAS service.	After installing the DAS MPack through the Ambari console, add the DAS service through the Ambari UI. See Adding the DAS Service through the Ambari UI.
Post-installation tasks		
7	Set up tmp directory.	You need write permission on the /tmp directory to write logs for secure clusters. See Setting up the tmp directory.
8	Complete the additional configuration tasks.	You can increase or decrease the time for which the audit and event logs are retained in the system, after which they are auto-purged. You can also make DAS work with HiveServer2 in case your Hive deployment is not LLAP-enabled and does not have Hive Server Interactive. See Additional configuration tasks.
9	Configure DAS for SSL.	It is recommended that you configure DAS for SSL. You can configure SSL if your HDP cluster is SSL-enabled. See (Recommended) Configuring DAS for SSL/TLS.
10	Configure user authentication.	To authenticate users using either Knox SSO, Knox proxy, or SPNEGO, configuring Knox SSO for HA clusters, and to enable Knox SSO logout option, see Configuring user authentication.

Support requirements

Before installing Data Analytics Studio (DAS/DAS-Lite), you must consider various aspects of your HDP environment and prepare your clusters.

Support matrix information

You can find the most current information about interoperability for this release on the Support Matrix. The Support Matrix tool provides information about:

- Operating Systems
- Databases
- Browsers
- JDKs

To access the tool, go to: https://supportmatrix.hortonworks.com.

DAS requirements

DAS cluster components comprise of the DAS Webapp and a DAS Event Processor. These are installed into your HDP cluster.

Table 2: DAS/DAS-Lite cluster requirements

Item	Specification/Version
Hortonworks Data Platform	• HDP 3.1.4
Cluster services	• YARN • HDFS • Hive • Knox • Ranger

Cluster operating systems	• HDP 3.1.4
	  • RHEL/CentOS/OEL 7
	  • Debian 9
	  • Ubuntu 16

Databases	PostgreSQL 9.6

Table 3: DAS/DAS-Lite port specifications

Default Port Number	Description	Additional Information
30900	Data Analytics Studio Event Processor server port	
30901	Data Analytics Studio Event Processor admin server port	
30800	Data Analytics Studio Webapp server port	
30801	Data Analytics Studio Webapp admin port	
5432	Postgres database	DAS uses Postgres database to store information.

DAS interacts with the other services using the following ports:

Table 4: Services port specifications

Default Port Number	Description	Additional Information
8020	NameNode host	
50010	All DataNode hosts	
8080	Ambari server host	
10501	Hive interactive thrift port	
10500	Hive interactive HiveServer2 Port	
10000	HiveServer2 host	Binary mode port (Thrift)
10001	HiveServer2 host	HTTP mode port
2181	ZooKeeper hosts	
8050	YARN port	

System requirements

DAS is a memory-heavy and a disk-light application. For optimum performance, consider profiling the CPU cores, memory allocation, and disk space depending upon the number of users, the total number of databases and tables, and the number of queries in the system.

If you are setting up a high-availability cluster, then add additional cores and memory for the load balancer.

The following table provides component-wise recommendation for provisioning CPU, memory, and disk space. These recommendations are approximated considering 10 users, 10,000 Hive tables, 100 parallel Event Processor threads, and 40,000 queries.

Table 5: Hardware requirements for DAS/DAS-Lite

DAS/DAS-Lite component	CPU	Memory	Local Disk
Webapp	• Minimum: 2 cores • Recommended: 2 cores *The number of cores that you allocate need to be proportional to U.	• Minimum: 4 GB • Recommended: 8 GB *The amount of memory that you allocate need to be proportional to U and T.	• Minimum: 5 GB • Recommended: 10 GB *The amount of disk space that you allocate need to be proportional to U.
Event Processor	• Minimum: 2 cores • Recommended: 4 cores *The number of cores that you allocate need to be proportional to P.	• Minimum: 4 GB • Recommended: 8 GB *The amount of memory that you allocate need to be proportional to P and T.	• Minimum: 5 GB • Recommended: 5 GB *The disk space is primarily used for logs, and can remain constant.
Database	• Minimum: 2 cores • Recommended: 4 cores *The number of cores that you allocate need to be proportional to (P + U).	• Minimum: 4 GB • Recommended: 8 GB *The amount of memory that you allocate need to be proportional to (T + Q).	• Minimum: 5 GB • Recommended: 20 GB *The amount of disk space that you allocate need to be proportional to (T + U + Q).

Where,

U is the number of users concurrently accessing the DAS Webapp

T is the number of tables in Hive

P denotes the parallelism configured in the DAS Event Processor

Q is the total number of queries in the system

Version recommendation for DAS

You must ensure that you are using versions of Data Analytics Studio, HDP, and Ambari that are supported together.

DAS 1.4.1 is compatible with the HDP 3.1.4 release. If you are using older versions of Hive, then use the Ambari patch upgrade process, which will upgrade just the required components of HDP stack, so that other HDP components are not affected.

Understanding and obtaining DAS binaries

Access to Data Analytics Studio application for production purposes requires authentication. To access this application, you must have an active subscription agreement along with the required authentication credentials (namely, the username and password).

The authentication credentials are provided in an email sent to the customer account from Cloudera when a new license is issued. If you have an existing license with a Data Analytics Studio entitlement, you might not have received an email.

Note: If you do not have the authentication credentials, contact your account representative to receive the same.

Make sure that you download the DAS/DAS-Lite binaries according to the HDP version and operating system as given in the following table.

Table 6: Obtaining binaries: DAS deployed in a standalone cluster

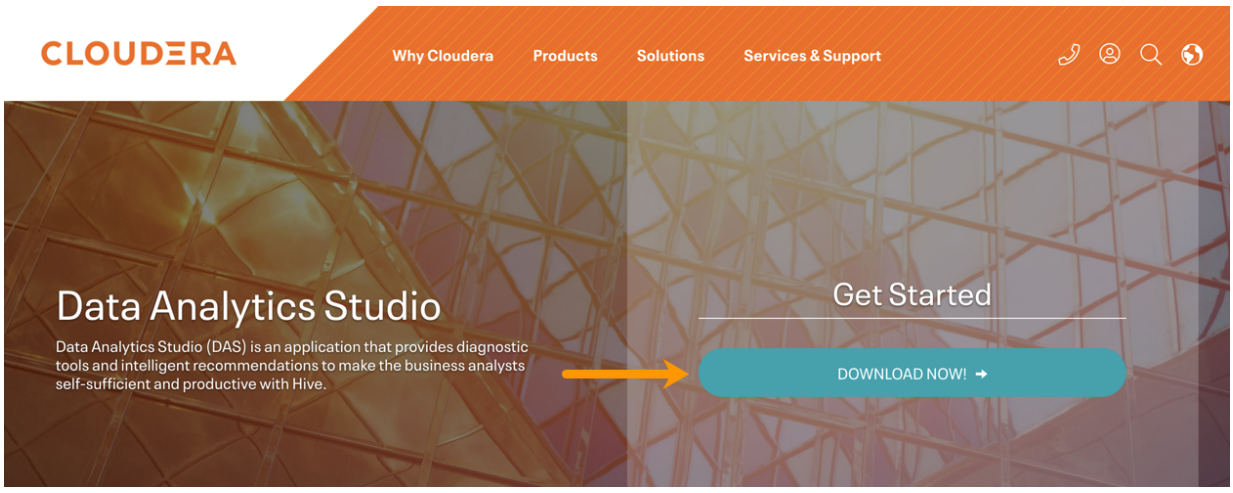
Versions and Operating Systems	CentOS 7	Ubuntu	Debian
HDP 3.x	• HDP 3 MPack for CentOS • HDP 3 DAS/DAS Lite tarball for CentOS	• HDP 3 MPack for Ubuntu • HDP 3 DAS/DAS Lite tarball for Ubuntu	• HDP 3 MPack for Debian • HDP 3 DAS/DAS Lite tarball for Debian

DAS is provided as RPMs in tarball repositories and as an Ambari Management Pack (MPack).

The DAS cluster components are installed in to your cluster using the Ambari MPack. The DAS cluster components are available for RHEL/CentOS/OEL 7, Debian 9 (HDP 3.x), and Ubuntu 16 platforms.

To download the Data Analytics Studio application, follow these steps:

1. Log into the cloudera.com account associated with the Data Analytics Studio license agreement.
2. On the Data Analytics Studio download page, click **DOWNLOAD NOW!**.



A table listing the release packages with the associated release numbers, supported operating system, and downloadable link is displayed.

DAS			
1.0.0.0-89	centos7	https://archive.cloudera.com/web/DAS/1.0.0.0-89/centos7/DAS-1.0.0.1.0.0.0-89-centos7-rpm.tar.gz	rpm
1.0.0.0-89	centos7	https://archive.cloudera.com/web/DAS/1.0.0.0-89/centos7/DAS-1.0.0.1.0.0.0-89-centos7-tars-tarball.tar.gz	tarball
1.0.0.0-89	centos7	https://archive.cloudera.com/web/DAS/1.0.0.0-89/centos7/data-analytics-studio-mpack-1.0.0.1.0.0.0-89.tar.gz	mpack
1.0.0.0-89	centos7	https://archive.cloudera.com/web/DAS/1.0.0.0-89/centos7/data_analytics_studio-1.0.0.1.0.0.0-89-release_source.tar.gz	tarball
1.0.2.0-6	centos7	https://archive.cloudera.com/web/DAS/1.0.2.0-6/oE7aK/centos7/DAS-1.0.2.0-6-centos7-rpm.tar.gz	rpm
1.0.2.0-6	centos7	https://archive.cloudera.com/web/DAS/1.0.2.0-6/oE7aK/centos7/data-analytics-studio-mpack-1.0.2.1.0.2.0-6.tar.gz	mpack
1.0.2.0-6	centos7	https://archive.cloudera.com/web/DAS/1.0.2.0-6/oE7aK/centos7/data_analytics_studio-1.0.2.1.0.2.0-6-source.tar.gz	tarball
1.0.2.0-6	debian9	https://archive.cloudera.com/web/DAS/1.0.2.0-6/oE7aK/debian9/DAS-1.0.2.0-6-debian9-deb.tar.gz	deb
1.0.2.0-6	ubuntu16	https://archive.cloudera.com/web/DAS/1.0.2.0-6/oE7aK/ubuntu16/DAS-1.0.2.0-6-ubuntu16-deb.tar.gz	deb

3. Download the DAS MPack and the DAS RPM corresponding to your operating system and the DAS version that you want to install on your computer by clicking the downloadable link.

Table 7: Supported platforms - DAS cluster components

DAS/DAS-Lite	CentOS 7	Debian 9	Ubuntu 16
DAS Cluster Components (DAS Event Processor and DAS Webapp)	Yes	Yes (HDP 3.1.5)	Yes

Pre-installation tasks

Review and complete the following tasks before installing DAS:

Install the prerequisite components and services

You need an Ambari cluster with the following components and services to install and use DAS: YARN, HDFS, Hive, and Knox and Ranger for secure clusters.

Before you begin

- Set up a cluster managed by Ambari that includes at least the following components and services: YARN, HDFS, Hive.

  Configure Knox and Ranger if you choose to set up secure clusters.

  To install Ambari and the related services, see Installing Ambari in the *Apache Ambari Installation Guide*.
- (For secure clusters) Configure Knox SSO for Ambari, Hive, and Ranger. See Setting up Knox SSO.

  After you set up and enable Knox SSO through the Ambari CLI, test whether you have configured it correctly by using the following steps:

  1. Sign out from Ambari.

You should be redirected to the Knox SSO login page.

2. Sign in using your credentials.

   You should be able to log in using your credentials.

Check cluster configuration for Hive and Tez

Check the configuration settings for Hive and Tez in the Ambari UI.

Procedure

1. Go to **Ambari** > **Services** > **Hive** > **CONFIGS** > **ADVANCED**. Make sure that the Hive configurations are as follows:
   - **hive.hook.proto.base-directory**: {hive_metastore_warehouse_external_dir}/sys.db/query_data/
   - **hive.exec.failure.hooks**: org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook
   - **hive.exec.post.hooks**: org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook
   - **hive.exec.pre.hooks**: org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook
   - **hive.metastore.transactional.event.listeners**: org.apache.hive.hcatalog.listener.DbNotificationListener

   Note: If you have upgraded your cluster from HDP 2.6.x to HDP 3.1.x, then specify org.apache.hive.hcatalog.listener.DbNotificationListener in the **hive.metastore.transactional.event.listeners** property under **Advanced hivemetastore-site**.
2. Go to **Ambari** > **Services** > **Tez** > **CONFIGS** > **Custom tez-site**. Make sure that the Tez configuration is as follows:
   - **tez.history.logging.service.class**: org.apache.tez.dag.history.logging.proto.ProtoHistoryLoggingService
   - **tez.history.logging.proto-base-dir**: {hive_metastore_warehouse_external_dir}/sys.db/

Create and set up a local repository

Hortonworks does not host any public repository for DAS. Therefore, you need to setup a local repository to install the binaries.

Prepare the web server for the local repository

The first step is to configure a web server on which you want to create the local repository.

Before you begin

- Set up a local repository host that runs a supported operating system.
- Enable network access from your target Ambari host to the local repository host.
- Ensure that the web server host has a package manager, such as yum (for RHEL, CentOS, or Oracle Linux), installed.

Procedure

1. Create an HTTP server.
   a) On the local repository host, install an HTTP server (such as Apache httpd) using the instructions provided on the Apache community website.
   b) Activate the server.
   c) Ensure that any firewall settings allow inbound HTTP access from your cluster nodes to your local repository host.

   Note: If you are using Amazon EC2, make sure that SELinux is disabled.

2. On your local repository host, create a directory for your web server by entering the following command:

```
mkdir -p /var/www/html/
```

3. Optional: If you are using a symlink, enable the followsymlinks on your web server.

Setting up the local repository

The second step is to set up a local repository. In this step, you move the tarball to the selected mirror server and extract the tarball to create the repository.

Before you begin

Ensure that you have downloaded the required tarballs from the Hortonworks customer portal by following the instructions provided as part of the product procurement process.

Procedure

1. Copy the repository tarballs to the web server directory and expand (uncompress) the archive file:
   a) Change to the web server directory that you created earlier.

   ```
   cd /var/www/html/
   ```

   All content in this directory is served by the web server.
   b) Move the tarballs to the current directory and expand each of the repository tarballs that you downloaded.
   Replace <file-name> with the actual name of the RPM tarball that you are expanding.

   ```
   tar zxvf <file-name>.tar.gz
   ```

   When you expand the tarball, subdirectories are created under /var/www/html/, such as DAS/centos7. These directories contain the repositories.
2. Confirm that you can browse to the newly created local repositories by using the base URLs: http://<webserver-host-name>/<repo-name>/<OS>/<service-version-X>
   - <webserver-host-name>

     This is the FQDN of the web server host.
   - <repo-name>

     This is composed of the abbreviated name of the repository, such as DAS.
   - <OS>

     This is the operating system version.
   - <service-version-X>

     This is the version number of the downloaded repository, appended with a unique version number.

   Base URL Examples

   DAS Base URL:

   ```
   http://webserver.com:port/DAS/centos7/1.2.0.0-X
   ```

   If you choose to use DAS with DP, then note the base URLs. You need the base URLs to install the DAS-DP App on the host and to install the associated agent on the clusters.
3. If you have configured multiple repositories in your environment, then install the following plugin on all the nodes in your cluster:

   ```
   yum install yum-plugin-priorities
   ```

4. Edit the /etc/yum/pluginconf.d/priorities.conf file to add the following values:

```
[main]
enabled=1
gpgcheck=0
```

Results

The repositories for DAS are now prepared for installation.

Creating the repository configuration file

The final step is to create the repository configuration file. The file is required to identify the path to the repository data, and check whether a GPG signature check should be performed on the repository packages. You need only one repository configuration file.

Procedure

1. Navigate to the repository directory.

```
cd /etc/yum.repos.d/
```

2. Create a repository file.

```
vi das.repo
```

Alternatively, you can copy an existing repository file to edit.

3. Add the following content in the repository file:
#VERSION_NUMBER=<downloaded-version#> [<service-name-abbreviation>]

This is composed of the service name abbreviation and version number (includes the build number). Example: DAS-APP-1.2.0.0-x.

```
name=<service-name-abbreviation> Version - <service-name-abbreviation>
```

```
baseurl=http://<webserver-host-name>/<directory-containing-repo>
```

<webserver-host-name> is the FQDN of the web server host that contains the repository. This is the same base URL that you used earlier to prepare the repositories.

<directory-containing-repo> is the path expanded from the tarball.

```
gpgcheck=1
gpgkey=http://<webserver-host-name>/<directory-containing-repo>/RPM-GPG-
KEY/RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

Example Repository File:

```
#VERSION_NUMBER=1.2.0.0-x
[DAS-APP-1.2.0.0-x]
name=DAS-APP Version - DAS-APP-1.2.0.0-x
baseurl=http://<your_webserver>:port/DAS-APP/centos7/1.2.0.0-x
gpgcheck=1
gpgkey=http://<your_webserver>:port/DAS-APP/centos7/1.2.0.0-x/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

Configure PostgreSQL database

DAS requires a PostgreSQL database for storing query event information. During the installation, uncheck the **Create Data Analytics Studio database** option.

Before you begin

Note:

- Cloudera no longer provides support for any managed Postgres instances on production clusters. However, you can use the managed Postgres database for running a POC. It is strongly recommended to use a custom PostgreSQL database for production environments.
- The custom database that is supported for use is PostgreSQL 9.6.

The default database name is das.

For CentOS

The commands to configure Postgres database are different for CentOS, Debian, and Ubuntu. Refer to the respective section to view the procedure for your platform.

Procedure

1. Install the supported version of Postgres using the following commands:

```
yum install https://download.postgresql.org/pub/repos/yum/9.6/redhat/
rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
```

```
yum install postgresql96-contrib postgresql96-server
```

   For more information about the supported version of Postgres, see the *DAS Support Matrix*.
2. Initialize the Postgres database by running the following command:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

3. Open the pg_hba.conf file for editing by entering the following command:

```
vi /var/lib/pgsql/9.6/data/pg_hba.conf
```

4. Add lines similar to the following lines:

```
local   all             <dbuser>                                md5
host    all             <dbuser>      0.0.0.0/0                 md5
host    all             <dbuser>      ::/0                      md5
local   all             postgres         ident
```

5. Open the postgresql.conf file for editing.

```
vi /var/lib/pgsql/9.6/data/postgresql.conf
```

6. Add, update, or uncomment the listen_addresses line as follows:

```
listen_addresses = '*'
```

7. Start the Postgres database by running the following command:

```
service postgresql-9.6 start
```

   Wait for the command to finish in some time or kill the command if it does not respond.

8. Create roles in Postgres by running the following commands as a Postgres user:

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbname>" | grep 1 ||
 (
psql -c "CREATE ROLE <dbuser> WITH LOGIN PASSWORD <dbpass>;" &&
psql -c "ALTER ROLE <dbuser> SUPERUSER;" &&
psql -c "ALTER ROLE <dbuser> CREATEDB;" &&
psql -c "CREATE DATABASE <dbname>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbname> TO <dbuser>;")
```

Replace <dbname> with the database name, <dbuser> with the database username and <dbpass> with the database password.

For Debian

The commands to configure Postgres database are different for CentOS, Debian, and Ubuntu. Refer to the respective section to view the procedure for your platform.

Procedure

1. Install the supported version of Postgres using the following commands:

```
echo deb http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main > /
etc/apt/sources.list.d/postgresql.list
```

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
 apt-key add -
```

```
apt-get update
```

```
apt-get install postgresql-9.6
```

For more information about the supported version of Postgres, see the *DAS Support Matrix*.

2. To make Postgres accessible to the DAS webapp and the DAS event processor host:
   a) Open the pg_hba.conf file for editing.

```
vi /etc/postgresql/9.6/main/pg_hba.conf
```

   If the configuration file is not available at the above location, it could be located at /var/lib/postgresql/9.6/main/pg_hba.conf.
   b) Add lines similar to the following lines:

```
local   all             <dbuser>                                md5
host    all             <dbuser>      0.0.0.0/0                 md5
host    all             <dbuser>      ::/0                      md5
local   all             postgres         ident
```

   c) Open the postgresql.conf file for editing.

```
vi /etc/postgresql/9.6/main/postgresql.conf
```

   If the configuration file is not available at the above location, it could be located at /var/lib/postgresql/9.6/main/postgresql.conf.
   d) Add, update, or uncomment the listen_addresses line as follows:

```
listen_addresses = '*'
```

3. Start the Postgres database by running the following command as a Postgres user:

```
service postgresql start
```

Wait for the command to finish in some time or kill the command if it does not respond.

4. Create roles in Postgres by running the following commands as a Postgres user:

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbname>" | grep 1 ||
 (
psql -c "CREATE ROLE <dbuser> WITH LOGIN PASSWORD <dbpass>;" &&
psql -c "ALTER ROLE <dbuser> SUPERUSER;" &&
psql -c "ALTER ROLE <dbuser> CREATEDB;" &&
psql -c "CREATE DATABASE <dbname>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbname> TO <dbuser>;")
```

Replace <dbname> with the database name, <dbuser> with the database username and <dbpass> with the database password.

For Ubuntu

The commands to configure Postgres database are different for CentOS, Debian, and Ubuntu. Refer to the respective section to view the procedure for your platform.

Procedure

1. Install the supported version of Postgres using the following commands:

```
echo deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg main > /etc/
apt/sources.list.d/postgresql.list
```

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
 apt-key add -
```

```
apt-get update
```

```
apt-get install postgresql-9.6
```

For more information about the supported version of Postgres, see the *DAS Support Matrix*.

2. To make Postgres accessible to the DAS webapp and the DAS event processor host:
   a) Open the pg_hba.conf file for editing.

```
vi /etc/postgresql/9.6/main/pg_hba.conf
```

If the configuration file is not available at the above location, it could be located at /var/lib/postgresql/9.6/main/pg_hba.conf.

   b) Add lines similar to the following lines:

```
local   all             <dbuser>                                md5
host    all             <dbuser>      0.0.0.0/0                 md5
host    all             <dbuser>      ::/0                      md5
local   all             postgres         ident
```

   c) Open the postgresql.conf file for editing.

```
vi /etc/postgresql/9.6/main/postgresql.conf
```

If the configuration file is not available at the above location, it could be located at /var/lib/postgresql/9.6/main/postgresql.conf.

d) Add, update, or uncomment the listen_addresses line as follows:

```
listen_addresses = '*'
```

3. Start the Postgres database by running the following command as a Postgres user:

```
service postgresql start
```

Wait for the command to finish in some time or kill the command if it does not respond.

4. Create roles in Postgres by running the following commands as a Postgres user:

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbname>" | grep 1 ||
 (
psql -c "CREATE ROLE <dbuser> WITH LOGIN PASSWORD <dbpass>;" &&
psql -c "ALTER ROLE <dbuser> SUPERUSER;" &&
psql -c "ALTER ROLE <dbuser> CREATEDB;" &&
psql -c "CREATE DATABASE <dbname>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbname> TO <dbuser>;")
```

Replace <dbname> with the database name, <dbuser> with the database username and <dbpass> with the database password.

Installing Data Analytics Studio

First, install the DAS cluster components on the Ambari server using the MPacks. Then add the DAS service through the Ambari UI. Finally, configure DAS to use Knox SSO to enable DAS to work with the HDP cluster SSO.

Installing the DAS cluster components

You install the DAS cluster components (the DAS Webapp and the DAS Event Processor) by using an Ambari management pack (MPack). An MPack bundles service definitions, stack definitions, and stack add-on service definitions.

About this task

This task must be completed on all the clusters to be used with DAS.

Before you begin

You must have root access to the Ambari Server host node and you must perform this task as a root user.

Important: Download the DAS binaries from the Data Analytics Studio download page after logging into the cloudera.com account associated with the Data Analytics Studio license agreement.

Procedure

1. Log in to the Ambari host on a cluster.
2. Install the Data Analytics Studio MPack by running the following command:

```
ambari-server install-mpack --mpack=<mpack-name> --verbose
```

Replace the <mpack-name> with the name of the MPack. Typically, the format of the MPack is hdpx-data-analytics-studio-mpack-X.X.X.tar.gz.

3. Restart the Ambari server by entering the following command:

```
ambari-server restart
```

Adding the DAS service through the Ambari UI

After you successfully install the DAS MPack through the Ambari console, add the DAS service through the Ambari UI.

Procedure

1. Sign in to the Ambari host with your credentials.
   http://<ambari-server-host>:8080
2. If you are using a local repository, then specify it in the **dasbn-repo** field.
   a) Go to **Ambari** > **Admin** > **Stacks and Versions** > **Versions** and click **Show Details**.
   b) On the **Version Details** pop-up, click the **Edit Repositories** icon.
   c) On the **Repositories** screen, specify the local repository URL in the **dasbn-repo** field and click **Save**.
3. In the Ambari Services navigation pane, click **Actions** > **Add Service**.
   The **Add Service Wizard** is displayed.
4. On the **Choose Services** page of the Wizard, select **Data Analytics Studio**.
   Other required services are automatically selected.
5. When prompted to confirm the addition of dependent services, give a positive confirmation to all.
   This adds the other required services.
6. On the **Assign Masters** page, assign the nodes for the Webapp and the Event Processor as per your configuration outline.
7. On the **Customize Services** page, expand **Advance_data_analytics_studio-database** and fill in the database details and other required fields that are highlighted.
   a. Uncheck **Create Data Analytics Studio database**.
   b. Set the database host in the **Data Analytics Studio database hostname**.
   c. Set the database username in **Data Analytics Studio database username**.
   d. Database Password - Enter the password.

   You can set the credentials as per your requirement.
8. If Hive SSL is enabled, set the **Hive session params** in DAS configuration as follows:

```
sslTrustStore=/etc/security/serverKeys/
hivetruststore.jks;trustStorePassword=your_password
```

9. If you are running HSI in an active-passive HA mode, then specify a configuration string in the **Hive session params** field as shown in the following example along with the rest of the parameters such as truststore and password:

```
serviceDiscoveryMode=zooKeeperHA;hive.server2.zookeeper.namespace=hs2ActivePassiveHA
```

10. In the **user_authentication** field, specify the authentication method that you want to use. You can specify one of the following:
    - NONE: Specify NONE if you do not want to enable user authentication. In this case, all the queries will be executed as the hive service user.
    - KNOX_SSO: Specify KNOX_SSO to enable single sign-on experience to authenticate users as configured in the Knox SSO topology.
    - KNOX_PROXY: Specify KNOX_PROXY to use DAS behind a Knox proxy. The user is authenticated as configured in the Knox proxy topology. It also enables SPNEGO authentication.

- SPNEGO: Specify SPNEGO to authenticate users using a token-based authentication.

You can configure the user authentication separately, after you have added the DAS service. See Configuring user authentication.

11. Complete the remaining installation wizard steps and exit the wizard.
12. Ensure that all components required for your DAS service have started successfully.
13. Restart all the affected services in Ambari.

Post-installation tasks

Complete the following tasks after installing DAS/DAS-Lite:

Setting up the tmp directory

DAS allows you to download logs for secure clusters. To download logs, make sure that the DAS service user has write permission to the /tmp directory. Also make sure that the /tmp directory has sufficient storage space to hold logs from a query for the download logs feature to work.

Additional configuration tasks

You can increase or decrease the time for which the audit and event logs are retained in the system, after which they are auto-purged. You can also make DAS work with HiveServer2 in case your Hive deployment is not LLAP-enabled and does not have Hive Server Interactive.

Making DAS work with HiveServer 2

If your Hive deployment is not LLAP-enabled and does not have Hive Server Interactive, you need to update the configuration for DAS to work with HiveServer 2 as follows:

Go to **Ambari** > **Data Analytics Studio** > **Configs** and change use.hive.interactive.mode=true to use.hive.interactive.mode=false

Customizing the retention period of audit and event logs

Audit and event logs are written by Hive/Tez to HDFS for DAS to consume to generate DAG information. Once DAS consumes them, this information on HDFS is no longer required. They are cleaned automatically via a thread in HMS (Hive metastore).

The time to live count of the audit logs is defined in the hive.hook.proto.events.ttl parameter. By default, the logs are retained for seven days. To configure a custom value:

1. From the Ambari UI, go to **Hive** > **HiveMetaStore** > **Configs** and specify the time to live in the hive.hook.proto.events.ttl parameter.

   For example, to retain the logs for 30 days, specify 30d.

   Typically, a 30-day duration is useful if you want to download older DAG data.
2. Restart Hive service.

(Recommended) Configuring DAS for SSL/TLS

If your HDP cluster is SSL enabled, then you can configure SSL. You can use one of the two options to set up SSL certificates.

- Set up trusted CA certificates

- Set up self-signed certificates

Set up trusted CA certificate

You can enable SSL for the DAS Engine using a certificate from a trusted Certificate Authority (CA). Certificates from a trusted CA are primarily used in production environments. For a test environment, you can use a self-signed certificate.

Before you begin

- You must have root user access to the clusters on which DAS Engine is installed.
- You must have obtained a certificate from your CA, following their instructions.

Procedure

1. Log in as root user on the cluster with DAS Engine installed.
2. Import the Certificate Chain Certificate and the certificate you obtained from your CA.

```
keytool -import -alias root -keystore <path_to_keystore_file> -
trustcacerts -file <certificate_chain_certificate>
```

```
keytool -import -alias jetty -keystore <path_to_keystore_file> -file
 <certificate_from_CA>
```

Note: Ignore the following warning:

```
The JKS keystore uses a proprietary format. It is recommended
 to migrate to PKCS12 which is an industry standard format using
 "keytool -importkeystore -srckeystore <keystore_file_path> -
destkeystore <keystore_file_path> -deststoretype pkcs12".
```

Set up self-signed certificates

You can enable SSL for the DAS Engine using a self-signed certificate. Self-signed certificates are primarily used in test environments. For a production environment, you should use a certificate from a trusted CA.

Before you begin

You must have root user access to the clusters on which DAS Engine is installed.

Procedure

1. Log in as root user on the cluster with DAS Engine installed.
2. Generate a key pair and keystore for use with DAS Engine.

```
keytool -genkey -alias jetty -keystore <certificate_file_path> -storepass
 <keystore_password> -dname 'CN=das.host.com, OU=Eng, O=ABC Corp, L=Santa
 Clara, ST=CA, C=US' -keypass <key_password> -keyalg RSA
```

Note: Ignore the following warning:

```
The JKS keystore uses a proprietary format. It is recommended
 to migrate to PKCS12 which is an industry standard format using
 "keytool -importkeystore -srckeystore <keystore_file_path> -
destkeystore <keystore_file_path> -deststoretype pkcs12".
```

Follow the prompts and enter the required information.

- CN must be the FQDN of the DAS Engine host.

- Default value for the key password is *password*.

  If you change the password, then you have to update the DAS configuration.

Following is a sample command output:

```
keytool -genkey -alias jetty -keystore ~/tmp/ks -storepass password
What is your first and last name?
  [Unknown]:  das.host.com
What is the name of your organizational unit?
  [Unknown]:  Eng
What is the name of your organization?
  [Unknown]:  ABC Corp
What is the name of your City or Locality?
  [Unknown]:  Santa Clara
What is the name of your State or Province?
  [Unknown]:  CA
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=das.host.com, OU=Eng, O=ABC Corp, L=Santa Clara, ST=CA, C=US correct?
  [no]:  yes

Enter key password for <jetty>
  (RETURN if same as keystore password):
```

Note: You will have to use this keystore file while configuring the DAS Engine for TLS in Ambari.

Configure SSL/TLS in Ambari

In the Ambari UI, you enable TLS for DAS Engine and update the DAS Engine configuration if settings change.

Procedure

1. Copy the keystore files generated in the earlier procedures to webapp and event processor hosts. Make sure they are owned by configured user for DAS. The default user is hive.

   For example:

   ```
   /etc/security/certs/das-cert.jks
   ```

2. Navigate to **Data Analytics Studio** > **Configs**.
3. Set the following properties in **Advanced data_analytics_studio-security-site** section.

Field	Value
ssl_enabled	Make sure it is checked.
webapp_keystore_file	Enter the keystore path on the webapp host.
das_webapp_keystore_password	Enter the password used in the previous procedure.
event_processor_keystore_file	Enter the keystore path on the event processor.
das_event_processor_keystore_password	Enter the password used in the previous procedure.

4. In the **Advanced data_analytics_studio-webapp-properties** section, set **Data Analytics Studio Webapp server protocol** property to **https**.
5. In the **Advanced data_analytics_studio-event_processor-properties** section, set **Data Analytics Studio Event Processor server protocol** property to **https**.

Configuring user authentication

You can authenticate the users by using either Knox SSO, Knox proxy, or SPNEGO. You can also choose not to set up user authentication. In this case, all the queries are executed as a hive service user.

Configuring user authentication using Knox SSO

To enable DAS to work with the HDP cluster SSO, configure the Knox settings as described here.

About this task

Note: Follow these instructions only if you choose to configure secure clusters.

Before you begin

You need to export the Knox certificate from the Knox gateway host. To find the Knox gateway host, go to **Ambari** > **Services** > **Knox** > **CONFIGS** > **Knox Gateway hosts**.

Procedure

1. SSH in to the Knox gateway host with a root or a knoxuser user.
2. Export the Knox certificate by running the following command:

```
/usr/hdp/current/knox-server/bin/knoxcli.sh export-cert --type PEM
```

Note: If you have already integrated Knox SSO earlier, then the gateway-identity.pem file would exist. Check whether the gateway-identity.pem file exists or not before running this command.

```
/usr/hdp/current/knox-server/data/security/keystores/gateway-identity.pem
```

If the export is successfully, the following message is displayed:

```
Certificate gateway-identity has been successfully exported to: /usr/hdp/
current/knox-server/data/security/keystores/gateway-identity.pem
```

Note the location where you save the gateway-identity.pem file.

3. If not done already, specify KNOX_SSO in the **user_authentication** field under **Ambari** > **Data Analytics Studio** > **Config**.
4. Enable the Knox SSO topology settings. From the Ambari UI, go to **Data Analytics Studio** > **Config** > **Advanced data_analytics_studio-security-site** and make the following configuration changes:
   a) Specify KNOX_SSO in the **user_authentication** field.
   b) Specify the Knox SSO URL in the **knox_sso_url** field in the following format:
      https:knox-host>:8443/gateway/knoxsso/api/v1/websso
   c) Copy the contents of the PEM file that you exported earlier in the **knox_publickey** field without the header and the footer.
   d) Add the list of users in the **admin_users** field who need admin access to DAS.
      You can specify * (asterisk) in the **admin_users** field to make all users the admin users.

      Note: Only admin users have access to all the queries. Non-admin users can access only their queries.

   e) Click **Save** and click through the confirmation pop-ups.
   f) Restart DAS and any services that require restart by clicking **Actions** > **Restart All Required**.

Configuring user authentication using Knox proxy

As of HDP 3.0, Knox Proxy is configured via the Knox Admin UI. To set up proxy, you first define the provider configurations and descriptors, and the topologies are automatically generated based on those settings. For more information, see Set Up Knox Proxy.

Procedure

1. SSH in to the machine on which you have installed the Knox gateway.
2. Change directory to the following:

   /usr/hdp/current/knox-server/data/services/das/1.4.1/
3. Create the following two configuration files with the given content:

   rewrite.xml

```
<!--
   Licensed to the Apache Software Foundation (ASF) under one or more
   contributor license agreements.  See the NOTICE file distributed with
   this work for additional information regarding copyright ownership.
   The ASF licenses this file to You under the Apache License, Version 2.0
   (the "License"); you may not use this file except in compliance with
   the License.  You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 implied.
   See the License for the specific language governing permissions and
   limitations under the License.
-->
<rules>
  <rule dir="IN" name="DAS/das/inbound/root" pattern="*://*:*/**/das/">
    <rewrite template="{$serviceUrl[DAS]}/"/>
  </rule>
  <rule dir="IN" name="DAS/das/inbound/path" pattern="*://*:*/**/das/
{**}">
    <rewrite template="{$serviceUrl[DAS]}/{**}"/>
  </rule>
  <rule dir="IN" name="DAS/das/inbound/pathqp" pattern="*://*:*/**/das/
{**}?{**}">
    <rewrite template="{$serviceUrl[DAS]}/{**}?{**}"/>
  </rule>
  <rule dir="OUT" name="DAS/das/outbound/assets" pattern="assets/{**}">
    <rewrite template="{$frontend[path]}/das/assets/{**}"/>
  </rule>
  <rule dir="OUT" name="DAS/das/outbound/rooturl">
    <rewrite template="{$frontend[path]}/das/"/>
  </rule>
  <filter name="DAS/das/outbound/dasrooturl">
    <content type="application/javascript">
      <apply path="/dasroot/.." rule="DAS/das/outbound/rooturl"/>
    </content>
  </filter>
</rules>
```

   service.xml

```
<!--
   Licensed to the Apache Software Foundation (ASF) under one or more
   contributor license agreements.  See the NOTICE file distributed with
```

```
   this work for additional information regarding copyright ownership.
   The ASF licenses this file to You under the Apache License, Version 2.0
   (the "License"); you may not use this file except in compliance with
   the License.  You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 implied.
   See the License for the specific language governing permissions and
   limitations under the License.
-->
<service role="DAS" name="das" version="1.3.0">
  <routes>
    <route path="/das" />
    <route path="/das/**" />
    <route path="/das/**/hivestudio*.js">
      <rewrite apply="DAS/das/outbound/dasrooturl" to="response.body"/>
    </route>
  </routes>
</service>
```

If you have more than one Knox servers in a high availability configuration, then you need to create these files on every machine.

4. Restart the Knox server.
5. Sign in to the Knox Admin UI. From Ambari, go to **Knox** > **Quick Links** and click **Knox Admin UI**.
6. Create or use an existing Provider Configuration with the following configuration:
   a) (Required) Authentication provider: It provides authentication mechanism for Knox. For example, ShiroProvider. ShiroProvider integrates with LDAP.
   b) (Required) Identity Assertion provider: It authenticates the user name requesting access to DAS through SPNEGO.
   c) (Optional) Add any other provider as per your organization's requirements, such as Acls or WebAppSec.
7. Create a new descriptor for the DAS service with the Provider Configuration that you created earlier. If a descriptor is already present with the required Provider Configuration, then you can edit it.
   a) Add a new service named DAS.
   b) Under **Descriptor Detail** > **Services** > **DAS**, add the DAS Webapp URL in the **URLs** field and save the changes.
8. If not done already, specify KNOX_PROXY in the **user_authentication** field under **Ambari** > **Data Analytics Studio** > **Config**.
9. Next, go to **Ambari** > **Data Analytics Studio** > **CONFIGS** > **Advanced data_analytics_studio-security-site** and configure the following fields:
   a) das_knox_user: It specifies the Knox service user, and is picked up automatically.
   b) das_knox_doas_param_name: Specify the parameter name that you specified under the **Params** field in the Knox Admin UI. However, if you have selected the default Identity Assertion provider, then the **das_knox_doas_param_name** name is doAs, and is configured automatically.
   c) knox_ssout_url: Specify the KnoxSSOUT URL in the following format:

      https://{host}:{port}/{cluster-name}/{knox-sso-out-topo}/api/v1/webssout

      This step requires you to have a KnoxSSO out topology present in your Knox server. To create the Knox SSO out topology, see Enabling logout option for secure clusters.
   d) Add the list of users in the **admin_users** field who need admin access to DAS.

      You can specify * (asterisk) in the **admin_users** field to make all users the admin users.

Note: Only admin users have access to all the queries. Non-admin users can access only their queries.

10. Restart the DAS service.

Setting up quick links for the DAS UI

To use the DAS UI quick link from Ambari in case of Knox proxy clusters, you (the admin user) need to paste the following quicklink.json file in the Ambari server machine.

Procedure

1. Create the quicklink.json file as per the following template:

```
{
  "name": "default",
  "description": "default quick links configuration",
  "configuration": {
    "protocol":
    {
      "type": "HTTPS_ONLY"
    },
    "links": [
      {
        "name": "data_analytics_studio_ui",
        "label": "Data Analytics Studio UI",
        "requires_user_name": "false",
        "component_name": "KNOX_GATEWAY",
        "url": "%@://%@:%@/gateway/<das_ui_topology_name>/das/",
        "port": {
          "https_property": "gateway.port",
          "https_default_port": "8443",
          "regex": "^(\\d+)$",
          "site": "gateway-site"
        }
      }
    ]
  }
}
```

Replace the das_ui_topology_name with the actual DAS UI topology.

2. Change to the following directory:

/var/lib/ambari-server/resources/mpacks/data-analytics-studio-mpack-<VERSION>/hdp-addon-services/DATA_ANALYTICS_STUDIO/<VERSION>/quicklinks/

Replace the <VERSION> with the version of the DAS distribution that you have installed.

3. Replace the existing quicklink.json file with the file that you created in step 1.

Verify that you have correctly replaced the quicklink.json file.

4. Restart Ambari by using the following command:

```
ambari-server restart
```

Configuring user authentication using SPNEGO

SPNEGO uses a Kerberized environment for user authentication. To use SPNEGO to authenticate users, specify SPNEGO in the user_authentication field.

Procedure

1. From the Ambari UI, go to **Data Analytics Studio** > **CONFIGS** > **Advanced data_analytics_studio-security-site**.
2. Specify SPNEGO in the **user_authentication** field.
3. Configure the **das_spnego_name_rules** field if you have specific name mapping rules for your cluster.
4. Save the changes and restart DAS.

Configuring Knox SSO for HA clusters

In a High Availability set up, the load balancer distributes the incoming requests to multiple Knox instances.

About this task

Note: Follow these instructions only if you choose to configure secure clusters.

The format of the Knox SSO URL is as follows: https://<address>/gateway/knoxsso/api/v1/websso

where, the address is the host:port of the load balancer pointing to the Knox instance. You can obtain the value of the load balancer's host and port from the following parameter: hadoop.http.authentication.authentication.provider.url.

However, if you are unable to locate the URL, then contact the one who set up Knox in HA mode for you.

Before you begin

You need to obtain the Knox certificate (also known as the knox_publickey) from the Knox gateway host.

Procedure

1. SSH in to the Knox gateway host with a root or a knoxuser user.
2. Obtain the Knox certificate by running the following commands, depending on whether you have set the gateway.signing.keystore.name parameter under the Knox configurations:
   a) If you have set the gateway.signing.keystore.name parameter, go to the Knox data folder and run the following command:

   ```
   keytool -exportcert -alias <gateway.signing.key.alias> -keypass <knox-secret> -keystore security/keystores/<gateway.signing.keystore.name> -storepass <knox-secret> -rfc
   ```

   where,

   - gateway.signing.keystore.name is typically a filename with .jks extension. For example, knoxidentity.jks.
   - The keypass and storepass are the Knox secret passwords that you specified while creating the .jks file. For example, knoxsecret.
   - The value of gateway.signing.key.alias can be obtained from Knox Config in Ambari or in the /etc/knox/conf/gateway-site.xml file. For example, knoxidentity.

   b) If you have not set the gateway.signing.keystore.name parameter, extract the certificate from the gateway.jks file by running the following command:

   ```
   /usr/hdp/current/knox-server/bin/knoxcli.sh export-cert --type PEM
   ```

   Note:

   The gateway.jks file is automatically created when Knox is started for the first time. If you have already integrated Knox SSO earlier, then the gateway-identity.pem file would exist. Check whether the gateway-identity.pem file exists or not before running this command.

The certificate is extracted from the gateway.jks file and is stored in a file called gateway-identity.pem located under the /var/lib/knox/data-<version>-<build-no>/security/keystores/ directory.

3. Enable the Knox SSO topology settings. From the Ambari UI, go to **Data Analytics Studio** > **Config** > **Advanced data_analytics_studio-security-site** and make the following configuration changes:
   a) Specify KNOX_SSO in the **user_authentication** field.
   b) Add the list of users in the **admin_users** field who need admin access to DAS.
      You can specify * (asterisk) in the **admin_users** field to make all users the admin users.

      Note: Only admin users have access to all the queries. Non-admin users can access only their queries.

   c) Specify the Knox SSO URL in the **knox_sso_url** field in the following format:
      https://<host:port_of_load_balancer>/gateway/knoxsso/api/v1/websso
   d) Copy the contents of the Knox certificate file that you extracted earlier in the **knox_publickey** field without the header and the footer.
   e) Click **Save** and click through the confirmation pop-ups.
   f) Restart DAS and any services that require restart by clicking **Actions** > **Restart All Required**.

Enabling logout option for secure clusters

To sign out from the application (DAS Webapp) and the identity provider on secure clusters, you need to create a KnoxSSO out topology in the Ambari console and configure the knox_ssout_url through the Ambari UI.

About this task

This section can be used to configure the Knox SSO logout URL for both HA and non-HA clusters.

Procedure

1. SSH in to the Ambari console on which you have configured Knox as a root user.
2. If you are configuring KnoxSSOUT for the first time, then you need to create a KnoxSSO out topology called knoxssout.xml in the /etc/knox/x.x.x.x-xx/0/topologies directory.
   If you have already configured KnoxSSOUT, then you can skip this step.
   Add the following lines in the knoxssout.xml file:

```
<?xml version="1.0"?>
<topology>
  <gateway>
    <provider>
      <role>hostmap</role>
      <name>static</name>
      <enabled>true</enabled>
    </provider>
  </gateway>
  <service>
    <role>KNOXSSOUT</role>
  </service>
</topology>
```

   Note: If you have configured Knox SSO for HA clusters, then you must create the KnoxSSO out topology on all the Knox hosts.

3. On the Ambari UI, go to **Data Analytics Studio** > **Config** > **Advanced data_analytics_studio-security-site** and specify the Knox SSO logout URL in the **knox_ssout_url** field in the following format:

```
https://{host}:{port}/{cluster-name}/{knox-sso-out-topo}/api/v1/webssout
```

Where, knox-sso-out-topo is the name of the KnoxSSO out topology xml file that you created earlier. In this case, it is knoxssout.

The host and the port that you specify here should be the same as those specified in the knox_sso_url field.

To obtain the cluster_name, on the Ambari UI, click **admin** > **Manage Ambari** > **Cluster Information** and note the value from the **Cluster Name** field.

If you have HA clusters, then specify the host and the port of the load balancer pointing to the Knox instance.

4. Click **Save** and click through the confirmation pop-ups.
5. Restart DAS and any services that require restart by clicking **Actions** > **Restart All Required**.