Data Analytics Studio Installation

Date of Publish: 2018-11-28

http://docs.hortonworks.com

Contents

Installation Overview (DAS/DAS-Lite)

You must install DataPlane, Data Analytics Studio Engine, and the Data Analytics Studio application (DAS/DAS-Lite) in the same order.

Important: DAS and DAS-Lite have the same release versions, as well as the same Support Matrix for a version number. However, the binaries for DAS and DAS-Lite are different for a given version. Make sure that you download the appropriate binary before starting the installation.

To install DAS or DAS-Lite, you must install the following components in this order:

1. DataPlane
2. Data Analytics Studio engine
3. Data Analytics Studio app

The DAS cluster agents are installed on the Ambari host using and Ambari management pack (MPack). Whereas, the DAS DP app is installed on the DP environment, and the required rpm files are available in the form of tarballs. You must download the appropriate tarballs for the supported OS versions as per the following matrix:

DAS/DAS-Lite	CentOS 7	Debian 9	Ubuntu 16
DAS Cluster Agents (DAS Event Processor and DAS Webapp)	✔	✔	✔
DAS DP Application	✔	✘	✘

You are strongly encouraged to read completely through this entire document before starting the installation process, so that you understand the interdependencies and order of the steps.

Installing Data Analytics Studio Engine on Clusters

Install the DAS Engine on clusters to begin the installation of Data Analytics Studio service.

Procedure

1. Make sure all the prerequisites are met.
2. Configure an external database.
3. Install the Data Analytics Studio Engine.

Prerequisites for Data Analytics Studio Engine

Perform these tasks before installing the Data Analytics Studio Engine on a cluster.

Before you begin

- Ensure that the clusters are running the latest version of HDP
- Ensure that the following HDP components are installed and configured:
  - Hive
  - Knox

- Ensure that you have installed DataPlane before installing DAS or DAS-Lite. For installing DataPlane, see Installing DataPlane.
- Make sure that you have installed and configured Knox SSO. For more information, see Setting up Knox SSO.

  After you set up Knox SSO, validate that it is configured correctly by signing out of Ambari. Upon signing out from Ambari, you should see the Knox SSO login page, and you should be able to log in using your credentials.
- Go to **Ambari** > **Services** > **Hive** > **CONFIGS** > **ADVANCED**. Make sure that the following Hive configurations are as follows:
  - **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
- Go to **Ambari** > **Services** > **Tez** > **CONFIGS** > **Custom tez-site**. Make sure that the following Tez configuration is as follows:
  - **tez.history.logging.service.class**: org.apache.tez.dag.history.logging.proto.ProtoHistoryLoggingService
- 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.

Configure Postgres database

DAS requires a PostgreSQL database for storing query event information. During the installation, you can choose to have DAS install and configure a default, embedded PostgreSQL database for use, or you can configure an external PostgreSQL database. You can do this by checking or unchecking the **Create Data Analytics Studio database** option.

About this task

If you want to use and manage your own database instead of the default database, you must configure the Postgres database and create the required roles in the database.

Although DAS provides an option to use the default, embedded database, the embedded database is intended for non-production use. It is strongly recommended to use an external database for production environments.

Note:

- The default, embedded database is created on the same host as the DAS Webapp component. It should not be installed on the Ambari server host because it could conflict with the Ambari embedded PostgreSQL instance.
- The external database that is supported for use is PostgreSQL 9.6.
- When creating an external database, the database name should be the same as the database username.

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
```

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

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbuser>" | 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 <dbuser>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbuser> TO <dbuser>;")
```

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

Note: The database user and database name must be the same. It should be the one that is used as the database username in the DAS configuration on Ambari.

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 /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 /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:

```
/usr/lib/postgresql/9.6/bin/pg_ctl start -D /var/lib/postgresql/9.6/main/
```

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

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbuser>" | 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 <dbuser>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbuser> TO <dbuser>;")
```

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

   Note: The database user and database name must be the same. It should be the one that is used as the database username in the DAS configuration on Ambari.

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 /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 /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:

```
/usr/lib/postgresql/9.6/bin/pg_ctl start -D /var/lib/postgresql/9.6/main/
```

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

```
psql -tc "SELECT 1 FROM pg_database WHERE datname = <dbuser>" | 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 <dbuser>;" &&
psql -c "GRANT ALL PRIVILEGES ON DATABASE <dbuser> TO <dbuser>;")
```

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

   Note: The database user and database name must be the same. It should be the one that is used as the database username in the DAS configuration on Ambari.

Install the Data Analytics Studio Engine

Data Analytics Studio requires the DAS Engine to be installed on all the custers. The engine is installed on the Ambari host, 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 to perform this task.

Important: Download the required repository tarballs from the Hortonworks customer portal by following the instructions provided as part of the product procurement process. The repository tarballs for the DAS Engine are different from the DAS app repository tarballs.

Procedure

1. Log in as root to an Ambari host on a cluster.
2. Install the Data Analytics Studio MPack by running the following command, replacing <mpack-file-name> with the name of the MPack.

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

3. Restart the Ambari server.

   ```
   ambari-server restart
   ```

4. Launch Ambari in a browser and log in.
   http://<ambari-server-host>:8080

   Default credentials are:

   Username: admin
   Password: admin
5. In the Ambari Services navigation pane, click **Actions** > **Add Service**.
   The **Add Service Wizard** displays.
6. On the **Choose Services** page of the Wizard, select the Data Analytics Studio service to install in Ambari, and then follow the on-screen instructions.

   Other required services are automatically selected.
7. When prompted to confirm addition of dependent services, give a positive confirmation to all.

   This adds other required services.
8. On the **Assign Masters** page, you can choose the default settings.
9. 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. If you installed Postgres on your own:

   1. Uncheck **Create Data Analytics Studio database**.
   2. Set the database host in the **Data Analytics Studio database hostname**.
   3. Set the database username in **Data Analytics Studio database username**.

   

   Note: The hostname is ignored if the **Create Data Analytics Studio database** option is checked, the database will be installed on the same host as webapp.

   b. Database Password - Enter the password.

You can set credentials to whatever you want.

10. If Hive SSL is enabled, set the **Hive session params** in DAS configuration as follows:

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

11. If KNOX SSO is enabled, update **admin_users** under **Advanced data_analytics_studio-security-site**, with the list of users who need admin access to DAS.

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

12. Complete the remaining installation wizard steps and exit the wizard.
13. Ensure that all components required for your DAS service have started successfully.
14. Make sure to restart all the affected services in Ambari.

Configure 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.

- Setup trusted CA certificates
- Setup 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>
```

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.

3. Export the certificate.

```
keytool -exportcert -alias jetty -keystore /my/file.keystore -file
 <certificate file path> -storepass <keystore_password> -rfc
```

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.
webapp_keystore_password	Enter the password used in the previous procedure.
event_processor_keystore_file	Enter the keystore path on the event processor.
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**.

Configure Knox SSO for Data Analytics Studio

Update the Knox SSO settings under DAS Configuration in Ambari.

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
   ```

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

   ```
   Certificate gateway-identity has been successfully exported to: /usr/
   $REPO/$VERSION/knox/data/security/keystores/gateway-identity.pem
   ```

   Note the location where you save the gateway-identity.pem file.
3. Enable the Knox SSO topology settings:
   a) From **Ambari** > **Data Analytics Studio** > **Configs** > **Advanced data_analytics_studio-security-site**, check to select **knox_sso_enabled**.
   b) Set **knox_sso_url** value as **https://<knox-host>:8443/gateway/knoxsso/api/v1/websso**.
   c) Copy the contents of the PEM file exported in Step 1 to knox_publickey. Make sure the certificate headers are not copied.
   d) Click **Save** and click through the confirmation pop-ups.
   e) Restart Data Analytics Studio webapp.
   f) Select **Actions** > **Restart All Required** to restart all other services that require a restart.

Installing the Data Analytics Studio App

After installing the DP Platform, you must install the Data Analytics Studio application.

About this task

You must install the app on the same host as DP Platform.

Procedure

1. Set up a local repository.
2. Create the repository configuration file.
3. Install the Data Analytics Studio app.
4. Enable the clusters for DAS in the DP Platform.
5. Add users and assign roles for the DAS app.

Set up a local repository

Setting up a local repository involves moving the tarball to the selected mirror server and extracting 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. To install the DAS DP app, you must download the CentOS tarball.

Hortonworks does not host any public repository for DAS. Therefore, you need to setup a local repository to install the binaries. As part of the DataPlane setup, you must have set up this local repository already and the same can be used for the DAS binaries. For more information, see Prepare the web server for the local repository.

Procedure

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

      ```
      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 in /var/www/html/, such as DAS/centos7. These directories contain the repositories.

      Expanding the DAS app tarball can take several seconds.
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/centos7/1.0.0.0-X
```

Note the base URLs because you need them to install the DAS 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.

What to do next

Create the configuration file for the DAS repository.

Create the repository configuration file

A repository configuration file must be created for the DAS Service on the DataPlane host. The file is required to identify the path to the repository data, and establish 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-1.0.0.0-59

```
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 in the task 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=1.0.0.0-1
[DAS-APP-1.0.0.0-59]
name=DAS-APP Version - DAS-APP-1.0.0.0-1
baseurl=http://<your_webserver>:port/DAS-APP/centos7/1.0.0.0
gpgcheck=1
gpgkey=http://<your_webserver>:port/DAS-APP/centos7/1.0.0.0/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

Install the Data Analytics Studio service app

Follow the instructions to install the Data Analytics Studio Service app.

Before you begin

Make sure that you have successfully installed DP Platform, and that DataPlane is running.

Procedure

1. Log in to the host on which you have set up the DataPlane repositories as a root user.
2. Install the RPMs for the DAS service application by entering the following command:

   ```
   yum install das_dp
   ```

   A folder is created that contains the Docker image tarball files and a configuration script.

   If the yum command fails, then the local repository was not set up correctly. Check the repository file /etc/yum.repos.d/das.repo on the host.
3. Navigate to the directory containing the installation scripts for the DAS service. For example:

   ```
   cd /usr/das/x.y.z.n-bb/das_dp/bin
   ```

   where x.y.z.n-bb refers to the version number of the DAS app that you installed in the earlier step.

4. Load the DAS Docker images and initialize the environment using the following commands:

```
./dasdeploy.sh load
```

```
./dasdeploy.sh init
```

It prompts for the master password that was used for initializing the DP Platform. Make sure you enter the same master password.

Images can take a while to load.

Note:

If you run into errors while deploying the DAS application, then destroy the deployment using the ./dasdeploy.sh destroy command and re-install the app. To check the logs of the das-app container, you can use the ./dasdeploy.sh logs command.

5. Verify that the container you installed is running by entering the following command:

```
./dasdeploy.sh ps
```

Make sure that the container with the name das-app is running.

Enable the clusters for DAS in the DP Platform

After installing the DAS app, you must enable clusters for it on the DP Platform.

Procedure

1. Log in to the DP Platform as a DataPlane Admin user.
2. Select the clusters from the list of clusters.

   The Services page is displayed.
3. Click the **Enable** button for the DAS service.

   A verification pop-up is displayed.

Results

The cluster is enabled for the DAS service.

Add users and assign roles for the DAS app

After you set up the LDAP configuration for DP Platform, you need to add users for the DAS app. During the LDAP configuration, you add users and groups that can log in as a DP admin. You must now assign roles to users and groups that allow the users to access the services that plug into DataPlane.

About this task

You must select the Data Analytics Studio User role for accessing the DAS Service. Users and groups should be assigned this role to access Data Analytics Studio service. To add users and groups, and to enable the Data Analytics Studio User role, see Managing Users and Groups in the *DataPlane Administration Guide*.

Upgrading DAS-Lite to DAS

If you are running the DAS-Lite version, you can upgrade to DAS to get the full feature set of DAS. With DAS, you can get comprehensive usage metrics about your tables and queries by generating reports.

Before you begin

- Download the required repository tarballs from the Hortonworks customer portal, following the instructions provided as part of the product procurement process.
- Make sure that you obtain the DAS Management pack (MPack) of the same release and version numbers as that of DAS-Lite that you are running.

Procedure

1. Log in to the Ambari UI as an admin user.
2. Stop the DAS service.
3. On the Ambari host:
   a) Download the DAS MPack corresponding to the DAS-Lite release version on your Ambari gateway host.
   b) Uninstall the DAS-Lite MPack by running the following command:

```
ambari-server uninstall-mpack --mpack=data-analytics-studio-lite-mpack
 --verbose
```

   c) Install the DAS MPack that you downloaded earlier in this process.

   Sample command:

```
ambari-server install-mpack --mpack=<mpack-filename>.tar.gz --verbose
```

   Replace the .tar.gz filename with the actual tarball that you have downloaded.
   d) Restart the Ambari server by entering the following command:

```
ambari-server restart
```

4. Carry out the following steps on each of your DAS host, which includes DAS webapp and the DAS events processor hosts:
   a) SSH in to your DAS host.
   b) Set up the DAS repository by copying the DAS repository configuration file to the following location:

   (On CentOS 7) /etc/yum.repos.d/

   (On Debian and Ubuntu) /etc/apt/sources.list.d
   c) (Only on CentOS): Search and remove the data_analytics_studio packages of the DAS-Lite version by running the following commands:

```
rpm -qa | grep data_analytics_studio
```

```
yum remove <data_analytics_studio_lite>
 <data_analytics_studio_lite_<version>>
```

   d) Install the DAS packages.

   (On CentOS 7)

```
yum install data_analytics_studio
```

   (On Debian and Ubuntu)

```
apt-get install data-analytics-studio
```

5. On the Ambari UI:
   a) Set up the SmartSense ID in DAS configurations.
      1. On the Ambari UI, go to **Data Analytics Studio** > **Configs**.

2. Under **Advanced data_analytics_studio-webapp-properties**, specify the SmartSense ID in the **Dataplane Smartsense id** field.

b) Restart the DAS service.