

Data Analytics Studio installation 1

# Data Analytics Studio Installation

**Date of Publish:** 2019-03-20



<https://docs.hortonworks.com>

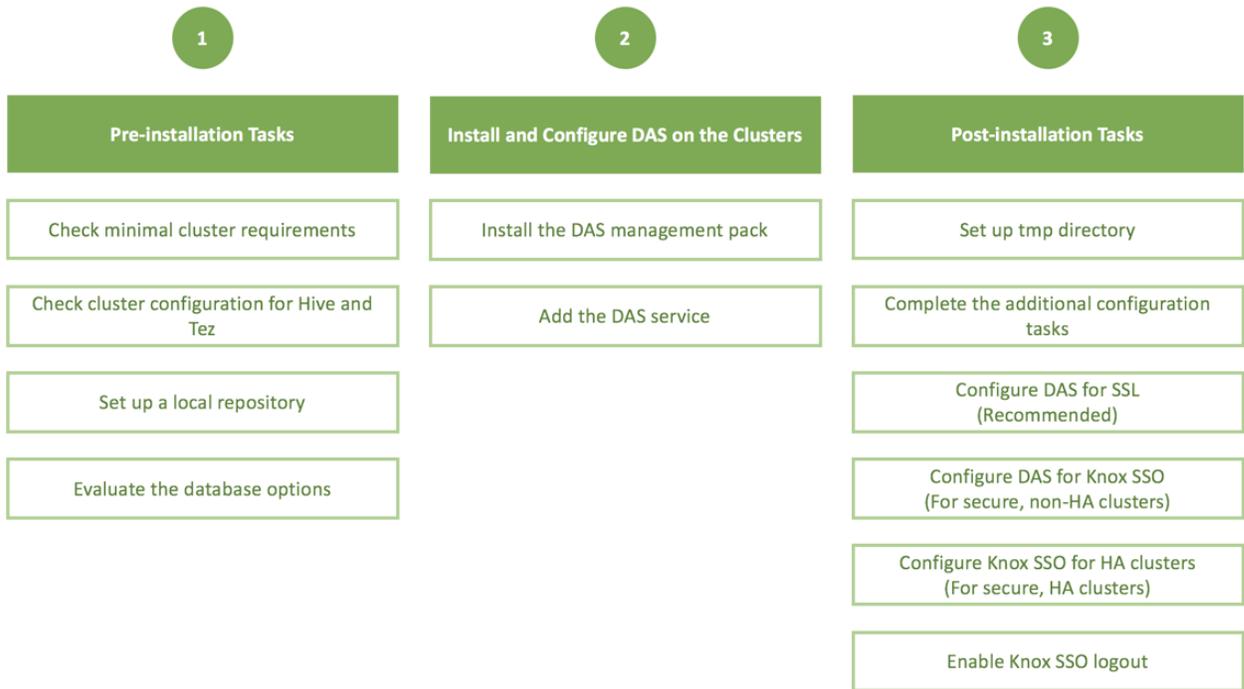
# Contents

<b>Installation roadmap.....</b>	<b>3</b>
<b>Support requirements.....</b>	<b>5</b>
<b>System requirements.....</b>	<b>7</b>
<b>Version recommendation for DAS.....</b>	<b>8</b>
<b>Understanding and obtaining DAS binaries.....</b>	<b>8</b>
<b>Pre-installation tasks.....</b>	<b>9</b>
Install the prerequisite components and services.....	9
Check cluster configuration for Hive and Tez.....	9
Check cluster configuration for Hive and Tez (HDP 2.6.5).....	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
Evaluate your database options.....	13
Configure Postgres database.....	13
<b>Installing Data Analytics Studio.....</b>	<b>16</b>
Installing the DAS cluster components.....	16
Adding the DAS service through the Ambari UI.....	17
<b>Post-installation tasks.....</b>	<b>18</b>
Setting up the tmp directory.....	18
Additional configuration tasks.....	18
(Recommended) Configuring DAS for SSL/TLS.....	19
Set up trusted CA certificate.....	19
Set up self-signed certificates.....	19
Configure SSL/TLS in Ambari.....	20
Configuring DAS for Knox SSO.....	21
Configuring Knox SSO for HA clusters.....	21
Enabling logout option for secure clusters.....	23
<b>(Optional) Installing DAS in DataPlane.....</b>	<b>23</b>
Installing the DAS-DP App in DP.....	24
Enabling the clusters for DAS on the DP Platform.....	25
Adding users and assigning roles for the DAS application.....	25

## 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. Optionally, you can install the DAS-DP App within the DataPlane environment. The DAS-DP App communicates with the DAS components within the cluster.

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.



**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 <a href="#">Install the prerequisite components and services</a> .
2	Check the cluster configuration for Hive and Tez.	Check the configuration settings for Hive and Tez in Ambari. <ul style="list-style-type: none"> <li>For HDP 2.6.5, see <a href="#">Check cluster configuration for Hive and Tez (HDP 2.6.5)</a></li> <li>For HDP 3.x or higher, see <a href="#">Check cluster configuration for Hive and Tez</a></li> </ul>
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 <a href="#">Create and set up a local repository</a> .
4	Evaluate the database options.	DAS requires a PostgreSQL database for storing query event information. See <a href="#">Evaluate your database options</a> .
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 <a href="#">Installing the DAS cluster components..</a>
6	Add the DAS service.	After installing the DAS MPack through the Ambari console, add the DAS service through the Ambari UI. See <a href="#">Adding the DAS Service through the Ambari UI</a> .
Post-installation tasks		
7	Set up tmp directory.	You need write permission on the /tmp directory to write logs for secure clusters. See <a href="#">Setting up the tmp directory</a> .
8	Complete the additional configuration tasks.	See <a href="#">Additional configuration tasks</a> .
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 <a href="#">(Recommended) Configuring DAS for SSL/TLS</a> .
10	Configure DAS for Knox SSO.	To enable DAS to work with the HDP cluster SSO, configure the Knox settings as described in <a href="#">Configuring DAS for Knox SSO</a> .
11	Configure Knox SSO for HA clusters.	To enable Knox SSO for HA clusters, see <a href="#">Configuring Knox SSO for HA clusters</a> .
12	Enable Knox SSO logout.	See <a href="#">Enabling logout option for secure clusters</a> .

## 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	<ul style="list-style-type: none"> <li>• HDP 2.6.5 (requires Patch #1050)<sup>1</sup></li> <li>• HDP 3.0</li> <li>• HDP 3.1</li> </ul>
Cluster services	<ul style="list-style-type: none"> <li>• YARN</li> <li>• HDFS</li> <li>• Hive</li> <li>• Knox</li> <li>• Ranger</li> </ul>
Cluster operating systems	<ul style="list-style-type: none"> <li>• HDP 3.0 and HDP 3.1 <ul style="list-style-type: none"> <li>• RHEL/CentOS/OEL 7</li> <li>• Debian 9</li> <li>• Ubuntu 16</li> </ul> </li> <li>• HDP 2.6.5 <ul style="list-style-type: none"> <li>• RHEL/CentOS/OEL 7</li> <li>• Ubuntu 16</li> </ul> </li> </ul>
Databases	PostgreSQL 9.6
DataPlane Platform (Optional)	DAS-DP App runs on DP Platform 1.2.0 or later.

**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	

<sup>1</sup> DAS only (HDP 2.6.5 support is not available with DAS-Lite).

Default Port Number	Description	Additional Information
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	<ul style="list-style-type: none"> <li>Minimum: 2 cores</li> <li>Recommended: 2 cores</li> </ul> <p>*The number of cores that you allocate need to be proportional to U.</p>	<ul style="list-style-type: none"> <li>Minimum: 4 GB</li> <li>Recommended: 8 GB</li> </ul> <p>*The amount of memory that you allocate need to be proportional to U.</p>	<ul style="list-style-type: none"> <li>Minimum: 5 GB</li> <li>Recommended: 10 GB</li> </ul> <p>*The amount of disk space that you allocate need to be proportional to U.</p>
Event Processor	<ul style="list-style-type: none"> <li>Minimum: 2 cores</li> <li>Recommended: 4 cores</li> </ul> <p>*The number of cores that you allocate need to be proportional to P.</p>	<ul style="list-style-type: none"> <li>Minimum: 4 GB</li> <li>Recommended: 8 GB</li> </ul> <p>*The amount of memory that you allocate need to be proportional to P.</p>	<ul style="list-style-type: none"> <li>Minimum: 5 GB</li> <li>Recommended: 5 GB</li> </ul> <p>*The disk space is primarily used for logs, and can remain constant.</p>
Database	<ul style="list-style-type: none"> <li>Minimum: 2 cores</li> <li>Recommended: 4 cores</li> </ul> <p>*The number of cores that you allocate need to be proportional to (P + U).</p>	<ul style="list-style-type: none"> <li>Minimum: 4 GB</li> <li>Recommended: 8 GB</li> </ul> <p>*The amount of memory that you allocate need to be proportional to (T + Q).</p>	<ul style="list-style-type: none"> <li>Minimum: 5 GB</li> <li>Recommended: 20 GB</li> </ul> <p>*The amount of disk space that you allocate need to be proportional to (T + U + Q).</p>

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



**Note:** The managed Postgres database is created on the same machine as the Webapp. Therefore, you must calculate and provision the resources for the database instance on that machine accordingly.

## Version recommendation for DAS

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

DAS 1.2.1 is compatible with HDP 2.6.5.1050 and HDP 3.1 releases. HDP 2.6.5.1050 release has fixes in different components of HDP like Hive, Ranger, and Atlas to enable the DAS functionality. 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.

If you are on an older release of HDP, then use a standard full upgrade to HDP 2.6.5.1050. Contact the Hortonworks support team to upgrade to HDP 2.6.5.1050.

## Understanding and obtaining DAS binaries

You must download the DAS binaries from the Hortonworks Customer Portal by following the instructions provided as part of the subscription fulfillment process.

Make sure that you download the DAS/DAS-Lite binaries according to the HDP version and operating system as given in the following table. If you choose to use DP, then you'll need to download an additional tarball as outlined in a separate table.

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

Versions and Operating Systems	CentOS 7	Ubuntu	Debian
HDP 2.6.5	<ul style="list-style-type: none"> <li>HDP 2 MPack for CentOS</li> <li>HDP 2 DAS tarball for CentOS</li> </ul>	<ul style="list-style-type: none"> <li>HDP 2 MPack for Ubuntu</li> <li>HDP 2 DAS tarball for Ubuntu</li> </ul>	<ul style="list-style-type: none"> <li>HDP 2 MPack for Debian</li> <li>HDP 2 DAS tarball for Debian</li> </ul>
HDP 3.x	<ul style="list-style-type: none"> <li>HDP 3 MPack for CentOS</li> <li>HDP 3 DAS/DAS Lite tarball for CentOS</li> </ul>	<ul style="list-style-type: none"> <li>HDP 3 MPack for Ubuntu</li> <li>HDP 3 DAS/DAS Lite tarball for Ubuntu</li> </ul>	<ul style="list-style-type: none"> <li>HDP 3 MPack for Debian</li> <li>HDP 3 DAS/DAS Lite tarball for Debian</li> </ul>

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.

Optionally, you can install the DAS-DP App in to the DataPlane environment and is available for RHEL/CentOS/OEL 7 and is for use on the DP Platform 1.2.0 or higher.

**Table 7: Obtaining binaries: DAS managed through DP**

Versions and Operating Systems	CentOS 7	Ubuntu	Debian
HDP 2.6.5	CentOS 7 DAS tarball.	CentOS 7 DAS/DAS-Lite tarball. It is mandatory to download this file if you want to install and use the DAS-DP App.	CentOS 7 DAS/DAS-Lite tarball. It is mandatory to download this file if you want to install and use the DAS-DP App.
HDP 3.x	CentOS 7 DAS tarball.	CentOS 7 DAS/DAS-Lite tarball. It is mandatory to download this file if you want to install and use the DAS-DP App.	CentOS 7 DAS/DAS-Lite tarball. It is mandatory to download this file if you want to install and use the DAS-DP App.

**Table 8: 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.x)	Yes

**Table 9: Supported platform - DAS-DP application**

DAS/DAS-Lite	CentOS 7
DAS-DP App	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
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/

## Check cluster configuration for Hive and Tez (HDP 2.6.5)

Check the configuration settings for Hive and Tez in this topic if you are using HDP 2.6.5.

### Procedure

1. Go to **Ambari > Services > Hive > CONFIGS > ADVANCED**. Make sure that the Hive configurations are as follows:
  - **tez-interactive-site:**  
tez.history.logging.service.class=org.apache.tez.dag.history.logging.proto.ProtoHistoryLoggingService
  - **hive-site:** hive.hook.proto.base-directory={hive\_metastore\_warehouse\_dir}/sys.db/query\_data
  - **hiveserver2-interactive-site:**  
hive.exec.post.hooks=org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook  
hive.exec.pre.hooks=org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook  
hive.exec.failure.hooks=org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook  
hive.server2.transport.mode=http
2. Go to **Ambari > Services > Tez > CONFIGS > Custom tez-site**. Make sure that the Tez configuration is as follows:
 

**custom tez-site.xml:** tez.history.logging.proto-base-dir={hive\_metastore\_warehouse\_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. If you already have a DataPlane deployment, then you have this local repository set up as part of the DP setup process, and you can use the same for the DAS 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. Do not use port 80 for the web server, if you choose to install the DAS-DP App in DataPlane. Port 80 is used by DataPlane, and using it for the web server can cause a conflict.
- 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.

- 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
```

- 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

- Navigate to the repository directory.

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

- Create a repository file.

```
vi das.repo
```

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

- 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
```

```

pgpkey=http://<your_webserver>:port/DAS-APP/centos7/1.2.0.0-x/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1

```

## Evaluate your database options

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.

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.

The default database name is das.

## Configure Postgres database

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.

### 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

```

- Open the postgresql.conf file for editing.

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

- Add, update, or uncomment the listen\_addresses line as follows:

```
listen_addresses = '*'
```

- 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.

- 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

- 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*.

- To make Postgres accessible to the DAS webapp and the DAS event processor host:

- 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.

- Add lines similar to the following lines:

```
local    all             <dbuser>
host     all             <dbuser>    0.0.0.0/0
host     all             <dbuser>    ::/0
```

```
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 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-DP App repository tarballs. You don't need to download the DAS-DP App repository tarball if you do not choose to use DP.

### 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 hdp<version>-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.

### Before you begin



**Note:** If you choose to use a managed Postgres database, then we recommend you to not install the DAS Webapp on the Ambari host, because there is a possibility that Ambari would have installed a Postgres database of its own. This can result into a conflict. However, if you are using a custom Postgres database, then the database resides on an isolated instance, and there is no chance of a conflict.

### 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. If you have installed Postgres database on your own, then:
    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, and the database will be installed on the same host as the Webapp.

**b.** 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 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.

**10.** If your HDP cluster is Kerberos enabled, then select the **authentication\_enabled** check-box under **Advanced data\_analytics\_studio-security-site**.

**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

Make sure you consider these additional configurations as part of your DAS installation.

**1.** Work with HiveServer2: 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`

**2.** Cleanup of Audit Log: Audit Log is written by Hive/Tez to HDFS for DAS to consume. 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:

**a.** On 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.

**b.** 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
<b>ssl_enabled</b>	Make sure it is checked.
<b>webapp_keystore_file</b>	Enter the keystore path on the webapp host.
<b>das_webapp_keystore_password</b>	Enter the password used in the previous procedure.
<b>event_processor_keystore_file</b>	Enter the keystore path on the event processor.

Field	Value
<code>das_event_processor_keystore_password</code>	Enter the password used in the previous procedure.

- In the **Advanced data\_analytics\_studio-webapp-properties** section, set **Data Analytics Studio Webapp server protocol** property to **https**.
- In the **Advanced data\_analytics\_studio-event\_processor-properties** section, set **Data Analytics Studio Event Processor server protocol** property to **https**.

## Configuring DAS for 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

- SSH in to the Knox gateway host with a root or a `knoxuser` user.
- 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.

- 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:
  - Select the **knox\_sso\_enabled** option.
  - Specify the Knox SSO URL in the **knox\_sso\_url** field in the following format:  
`https:knox-host>:8443/gateway/knoxssso/api/v1/websso`
  - Copy the contents of the PEM file that you exported earlier in the **knox\_publickey** field without the header and the footer.
  - Click **Save** and click through the confirmation pop-ups.
  - Restart DAS and any services that require restart by clicking **Actions > Restart All Required**.

## 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/kestores/<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/kestores/` 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) Select the **knox\_sso\_enabled** option.
  - b) 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`
  - c) Copy the contents of the Knox certificate file that you extracted earlier in the **knox\_publickey** field without the header and the footer.
  - d) Click **Save** and click through the confirmation pop-ups.
  - e) 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 `knoxsout.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 `knoxsout.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 `knoxsout`.

The host and the port that you specify here should be the same as those specified in the `knox_ssout_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**.

## (Optional) Installing DAS in DataPlane

This section is applicable only if you choose to manage DAS through the DataPlane platform.

## Installing the DAS-DP App in DP

After adding the DAS service in the Ambari UI, you can install the DAS-DP App in the DP environment.

### Before you begin

- Make sure that you have successfully installed DP Platform, and that DataPlane is running. To install DataPlane, thoroughly review the DataPlane [Getting Started](#) guide to understand the general requirements and basic dependencies, and then follow the [DataPlane Installation Guide](#).
- Verify whether you can access Ambari from DP without using your credentials. If you cannot access Ambari without using your credentials, then configure SSO from the DP cluster to the HDP cluster as per the steps listed in [Configure Knox SSO for DataPlane](#). This is different than having SSO enabled from the HDP cluster. To configure SSO from the HDP cluster, see [Configuring DAS for Knox SSO](#).
- Make sure that you download the DAS tarball for CentOS 7, even if you are on other platforms, such as Debian or Ubuntu.
- Enable network access from your target DP instance host to the local repository host. Do not use port 80 for the web server. Port 80 is used by DataPlane, and using it for the web server can cause a conflict.
- Verify whether you have the repository configuration file. To install the DAS-DP App in DataPlane, you must have a repository configuration file and use that file on the DataPlane host. 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 and you can use the same file that you had created while installing DAS.

### 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.

## Enabling the clusters for DAS on the DP Platform

After installing the DAS-DP 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.

## Adding users and assigning roles for the DAS application

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*.

### DAS Admin Users

You can add DAS Admin users using the DataPlane UI. DAS Admin users can view all the queries. For more information about adding DAS Admin users, see the [Add a user or group](#).