

Installation and Upgrade 1

Installing DataPlane

Date of Publish: 2018-08-07

<http://docs.hortonworks.com>

Contents

DataPlane Platform support requirements.....	3
Installation overview.....	4
Pre-installation tasks.....	6
Prepare your clusters.....	6
Perform the pre-installation tasks.....	6
Setting up the local repository for DataPlane.....	8
Prepare the web server for the local repository.....	8
Set up a local repository for DataPlane.....	9
Create the repository configuration file.....	10
Install and Configure DataPlane.....	10
Install DP Platform.....	11
Configure an external database.....	12
(Optional) Configure a TLS certificate.....	13
Initialize DP Platform.....	13
Log in and configure DP Platform.....	15
Configure Knox and Ranger for DataPlane.....	16
Configure Knox SSO for DataPlane.....	16
Configure Knox Gateway for DataPlane.....	20
Configure Ranger to restrict access to DataPlane.....	22
Upgrade DataPlane from 1.1.0 to 1.2.0.....	23
Troubleshooting DataPlane Installation.....	24
Cluster Registration Error Messages.....	24
Cluster is not reachable.....	24
Knox is not set up on the HDP cluster, or Ambari credentials are incorrect for 'seeded user' mode.....	24
Knox setup is incorrect on the HDP cluster.....	25
Cannot register a cluster, other causes.....	25
Cluster status displays incorrectly on Details page.....	25
Logging in using the DataPlane local admin role.....	26
wget command is not available.....	26
Delete and clean up Docker containers.....	26

DataPlane Platform support requirements

You should review the support requirements for the DP Platform to ensure your environment meets those requirements. Additionally, you must consider various aspects of your clusters and prepare those clusters as part of your DataPlane installation in order to register the clusters with DP Platform.

Important:

The specific DataPlane Apps you plan to install into your environment might bring additional requirements. Review the App-specific documentation thoroughly to ensure you can meet the App-specific requirements. For example, depending on your choice of Apps, your cluster requirements might change. This includes (but is not limited to) a minimal HDP version, setup and configuration of Knox, and other cluster requirements.

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>.

DP Platform Host requirements

DP Platform must be installed on a dedicated host that is not part of an existing HDP cluster, to prevent potential port conflicts with other cluster services.

DataPlane is certified for use with specific versions of CentOS and RHEL. These operating systems include support for Docker 1.12 or higher.

Table 1: Requirements for the DP Platform host

Item	Versions
Container infrastructure	Docker 1.12.x or higher
Processing and Memory Requirements	<ul style="list-style-type: none"> • Multicore processor, with minimum 8 cores • Minimum 16 GB RAM
Software	<ul style="list-style-type: none"> • yum, rpm • wget • tar • bash shell
Authentication	Existing LDAP or Active Directory (AD)
Environment	Disable SELinux
Ports	<ul style="list-style-type: none"> • 443 (Used by DataPlane for SSL access) • 80 (Redirected to port 443 for SSL) • 8500 (Used by Consul which handles the Docker container networking) <p>When configuring the software repositories, to avoid a potential port conflict, be sure you either use a distinct host or do not host the local repositories on port 80.</p>

Related Tasks

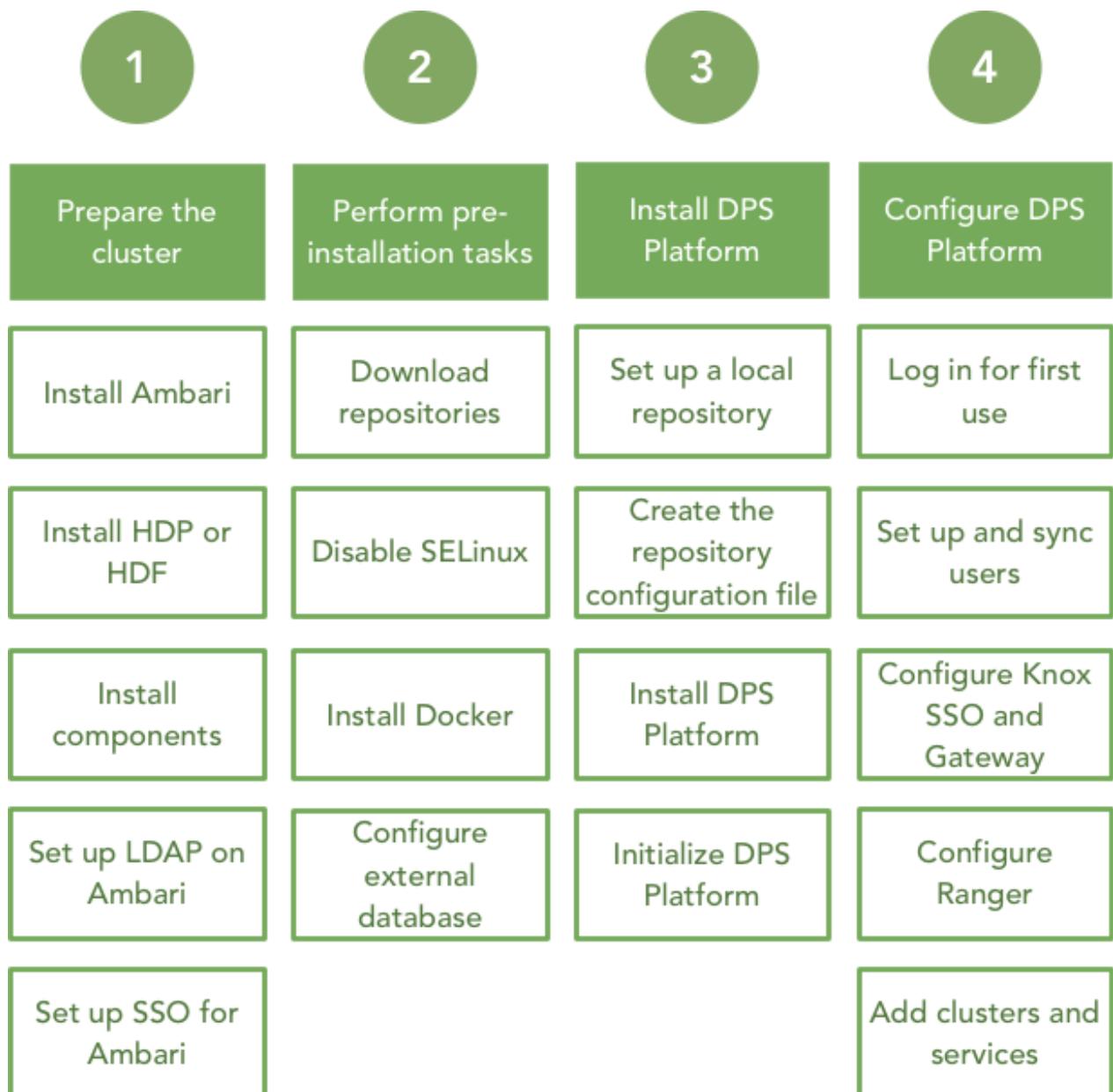
[General requirements for clusters](#)

Installation overview

DP Platform and its associated DP Apps are installed on a single host. The DP Platform and the DP Apps run as a set of “containers” on Docker on this host. It is recommended to have this as a dedicated host distinct from other software or cluster hosts. Let us refer to this host as your DP Instance (or “DP Host”).

Hosts from clusters that you plan to register into DataPlane must be accessible from this host. The hostname of a cluster node must be DNS addressable from the DataPlane host. In addition, for any DP Apps you plan to use with these clusters, you must install the requisite Cluster Agent for that DP App (for example: DLM Engine or DSS Profiler). Be sure your clusters meet the hardware and software requirements for that particular Agent. See the DataPlane Support Matrix and the support matrix for each of the DP Apps that you want to install.

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.



Pre-installation tasks

Prepare your cluster by installing or upgrading to the latest version of HDP or HDF and make sure you perform all the other prerequisite tasks.

Procedure

1. Prepare your clusters.
2. Perform the pre-installation tasks.

Prepare your clusters

Make sure you complete these basic preparation steps for any cluster you plan to use with DataPlane.

1. Install or upgrade to the supported version of Ambari. See [Apache Ambari Installation](#) for more details. See [Support Matrix](#) for details of the supported Ambari versions.
2. Install or upgrade to the supported versions of HDP or HDF on your cluster using Ambari. See the [Ambari Installation documentation](#) for more details. See [DP App specific Support Matrix](#) for details of the supported HDP and HDF versions.
3. Install and configure the required cluster services in your cluster. Make sure the DP App-specific components are installed in the cluster. For each app-specific list of requirements, see the [Product Interoperability details](#) in the apps' installation documentation.
4. Set up LDAP Authentication for Ambari.
 - a. For Ambari 2.7: See [Configuring Ambari Authentication with LDAP/AD](#).
 - b. For Ambari 2.6: See [Configuring Ambari for LDAP Authentication](#).
5. Set up Knox SSO with LDAP Authentication for Ambari.
 - a. For Ambari 2.7: See [Configuring Knox SSO](#).
 - b. For Ambari 2.6: See [Setting up Knox SSO for Ambari](#).

Perform the pre-installation tasks

Complete the pre-installation tasks before you install DP Platform.

Download product binaries

Download the DataPlane repository tarballs (or the product binaries) from the [Hortonworks Customer Portal](#) following the instructions provided as part of the subscription fulfillment process. DP Platform and the DP Apps (and related Cluster Agents) are provided as RPMs in tarball repositories.

Check DNS

Your system must be configured for both forward and reverse DNS.

Every host name used with DataPlane must be resolvable by DNS or configured in the `/etc/hosts` file on the DataPlane container, so that host names can be resolved between all cluster nodes. Using a DNS server is the recommended method, but if the instances are added to `/etc/hosts`, you must explicitly register the cluster host names within the DataPlane Docker containers. It is not sufficient to have the host names included in the `/etc/hosts` file on the DP Platform host. See the [DP Platform Administration guide](#) for instructions.

Note: If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

Check ports

The DP Host requires the following ports to be available on the host:

Port	Description
80, 443	The DP Instance Web UI for end-user access. Port 80 is redirected to port 443 for SSL. By default, DataPlane configures a self-signed SSL certificate. Refer to <i>Configuring DP Platform</i> to configure your own certificate.
8500	Used by Consul which handles the Docker container networking. This port is used internally in the DP instance deployment and does not need to be end-user accessible.

Disable SELinux

You must disable SELinux enforcement of permissions before installing DataPlane.

If you do not disable SELinux, then DataPlane will not install and run properly, and you will have to destroy and reinstall the containers.

```
setenforce 0    #A zero, not a letter
sed -i 's/^SELINUX=.* /SELINUX=disabled/g' /etc/sysconfig/selinux
```

Important: The second command prevents SELinux from being automatically re-enabled after a reboot.

Install Docker

Docker containers are used to install DataPlane. You must install either Docker Enterprise Edition (EE) or Community Edition (CE). You might be required to reboot your system after installing Docker.

For general information about installing Docker, see [Install Docker](#).

For Docker installation instructions for your operating system, access the appropriate Docker instructions:

- [Get Docker EE for Red Hat Enterprise Linux](#)
- [Get Docker EE for CentOS](#)
- [Get Docker EE for Oracle Linux](#)
- [Get Docker CE for CentOS](#)

Make sure you start the Docker service after installation using the following command:

```
service docker start
```

Configure LDAP

You need access to an enterprise LDAP setup when configuring DataPlane. Refer to *Enterprise LDAP requirements* for more information on the LDAP settings and options.

Configure external database

By default, DP Platform includes an embedded PostgreSQL instance for testing and evaluation purposes only.

Note: You should configure your DP instance to use an external PostgreSQL instance. We strongly recommend configuring DP Platform with an existing external database when running in production, and not use the embedded PostgreSQL. Refer to *Configuring DP Platform* for more information.

Related Tasks

[Advanced: Add host entries to the DataPlane environment](#)

Related reference

[DataPlane Support Matrix](#)

[Enterprise LDAP requirements](#)

Related Information

[DataPlane Administration](#)

Setting up the local repository for DataPlane

You must download the DataPlane repository tarballs from the Hortonworks Customer Portal following the instructions provided as part of the subscription fulfillment process. DP Platform and the DP Apps (and related Cluster Agents) are provided as RPMs in tarball repositories.

Before installing DataPlane, you must set up a server to host the RPMs in a local repository that can then be used to install the product binaries.

Note: You can create the local repository on the same host as your DP Instance, but do not host the local repository on port 80 since that will conflict with your DP Instance.

Prepare the web server for the local repository

Before setting up your local repository, you must properly configure an HTTP web server, on which you will create the repositories.

Before you begin

Prior to preparing the web server, you must have:

- Selected a server host that runs a supported operating system. This will be the local repository host.
- Enabled network access from your target DP Instance host to local repository host.
- Ensured that the web server is not using port 80. This port is used by DataPlane and will cause a conflict if in use by your web server.
- Ensured that the hosts have a package manager installed such as yum (for RHEL, CentOS, or Oracle Linux).

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.

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

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

What to do next

You next must set up your local repository.

Related Information

[Downloading the Apache HTTP Server](#)

Set up a local repository for DataPlane

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 tarball from the customer portal, following the instructions provided as part of the product procurement process.
- You must have completed the preparatory tasks before setting up a repository.

Procedure

1. Copy the repository tarballs to the web server directory and expand (uncompress) the archive file:

- a) Navigate 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 tarball to the current directory and expand each of the repository tarball.

Replace <filename> with the actual name of the RPM tarball that you are expanding.

```
tar zxvf <filename>.tar.gz
```

During expansion of the tarball, subdirectories are created in /var/www/html/, such as DP/centos7. These directories contain the repositories.

Expanding the tarball takes several seconds.

2. Confirm that you can browse to the newly created local repositories by using the *Base URL*:

```
http://<your_webserver>:port/<repo_name>/<OS>/<version>
```

- <your_webserver>:port

This is the FQDN and port of the web server host.

- <repo_name>

The repository name, usually the abbreviated name of the DataPlane component, for example DP for *DP Platform*.

- <OS>

The operating system.

- <version>

The version number of the downloaded component.

DP Platform Base URL example:

```
http://<your_webserver>:port/DP/centos7/1.2.0.0
```

Remember this base URL. You need it to set up the repository configuration file in subsequent steps.

3. If you have multiple repositories configured in your environment, deploy 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 local repository is now set up and ready for use.

What to do next

Create the repository configuration file for the newly created local repository.

Create the repository configuration file

A repository configuration file (.repo file) must be created on the DP host. The file is required to identify the path to the repository data, establish whether a GPG signature check should be performed on the repository packages, etc. Only one repository configuration file is needed.

Procedure

1. Navigate to the repository directory.
`cd /etc/yum.repos.d/`
2. Create a repository file.
`vi dp.repo`
Alternatively, you can copy an existing repository file to edit.
3. Add the following content to the repository file.

Important: Be sure to use the *Base URL* you created when setting up the local repository.

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

Results

You are now ready to install the DataPlane software.

Related Tasks

[Install DP Platform](#)

Install and Configure DataPlane

After setting up the local repository, install DP platform. You must configure an external database and then initialize the DP platform.

Procedure

1. Install DP Platform.

2. Configure an external database.
3. Configure a TLS certificate.
4. Initialize DP Platform.
5. Log in and configure DP Platform.

Install DP Platform

When installing DataPlane in a production environment, DP Platform and all associated DP apps must be installed on a separate host that is not part of any cluster.

About this task

DP Platform is required as a baseline, but you can install any combination of DP apps on top of the platform. After installing the platform, see the installation instructions for each DP app for instructions on how to install and configure the app on the platform.

- You will be installing the DP Platform software using the local repositories.

Before you begin

- You need root user access on the DP Host to perform this task.
- You must have completed the actions identified in *DataPlane installation prerequisites*.
- The host must meet the requirements identified in the *DataPlane Support Matrix*.
- The host must be a dedicated host that is not part of an existing cluster. This prevents potential port conflicts with other cluster services.
- You must have the FQDN or IP address of the host available.
- You will be initializing the DP Platform software, and optionally configuring an external database or your own SSL certificate.
- If you plan to use an external database, refer to the DataPlane Support Matrix for requirements and supported databases. Be sure to have that database URI, username and password available and to perform the steps in *Configure External Database* prior to performing the install steps below. We strongly recommend for any production deployment that you configure an external database.
- If you plan to use an SSL certificate, have the full path and file name for the public key and private key (.pem files) and the certificate password available. Be sure to perform the steps in *Configure a TLS Certificate* prior to performing the install steps below.

Procedure

1. Log in as root to the host on which you set up the DataPlane repositories.

```
sudo su
```

2. Verify that SELinux is disabled.

```
sestatus
```

If SELinux is not disabled, stop now and verify that all installation prerequisites were successfully completed, then continue with the installation.

3. Be sure Docker is installed, configured, and started on the host.

Refer to *Installation Prerequisites* for more information.

4. Install the DP Platform.

```
yum install dp-core
```

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/dp.repo` on the DP host.

What to do next

Proceed to initializing the DP Platform.

Related Tasks

[Perform the pre-installation tasks](#)

[Setting up the local repository for DataPlane](#)

Related Information

[Hortonworks Support Matrix](#)

Configure an external database

Although DataPlane includes an embedded PostgreSQL database, the embedded database is intended for nonproduction use. It is strongly recommended to use an external database for production environments. After installing the database following the instructions provided with the database software, you must set up the database for use with DataPlane.

About this task

- PostgreSQL database is supported in this version.
- You will be configuring an external database or your own SSL certificate.
- Refer to the *DataPlane Support Matrix* for requirements and supported databases.

Be sure to have the database URI, username, and password available.

Before you begin

- You need root user access on the DP Host to perform this task.
- You must have the proper role to perform this task.
- The PostgreSQL database must have been installed and properly configured for remote access.
- A database must have been created.
- A database user must have been created and assigned permissions for the new database.

Procedure

1. Open the `<installer_home>/config.env.sh` file for editing.

```
vi /usr/dp/current/core/bin/config.env.sh
```

2. Modify the DB Configs settings to add the appropriate connection information.

```
USE_EXTERNAL_DB="yes"  
DATABASE_URI="jdbc:postgresql://<host_name>:5432/<database_name>"  
DATABASE_USER="<user_name>"  
DATABASE_PASS="<password>"
```

Results

Your external database is now set up so that you can configure it for DataPlane during DataPlane installation.

(Optional) Configure a TLS certificate

If you choose to use the default TLS (formerly SSL) certificates provided with DataPlane, then DataPlane generates self-signed certificates. If using your own certificates, then you must modify certificate configuration settings.

About this task

DP Platform is required as a baseline, but you can install any combination of DP apps on top of the platform. After installing the platform, see the installation instructions for each DP app for instructions on how to install and configure the app on the platform.

- You will be configuring your own SSL certificate.

Before you begin

- You need root user access on the DP Host to perform this task.
- Have the full path and file name for the public key and private key (.pem files) and the certificate password available.

Procedure

1. Open the configuration file:

```
/usr/dp/current/core/bin/config.env.sh
```

2. Uncomment and modify the following properties:

```
USE_TLS="true"  
USE_PROVIDED_CERTIFICATES="yes"  
DATAPLANE_CERTIFICATE_PUBLIC_KEY_PATH="<absolute_path_of_public_key_file>"  
DATAPLANE_CERTIFICATE_PRIVATE_KEY_PATH="<absolute_path_of_encrypted_private_key_file>"
```

3. Save the file.

Initialize DP Platform

After installing the RPMs and optionally configuring your external database and TLS certificate, you must initialize DataPlane.

About this task

You will be initializing and configuring the DP Platform.

Before you begin

If you plan to use an external database or use your own TLS (SSL) certificate, be sure to configure those options prior to initializing the DP Platform. Refer to [Configure an external database](#) and [Configure a TLS certificate](#) for more details.

Procedure

1. Navigate to the folder containing the DataPlane configuration script.

```
cd /usr/dp/current/core/bin
```

2. Initialize the software.

This loads the Docker images into your local system and prompts for configuration options.

```
./dpdeploy.sh init --all
```

3. Create the password for a Super User and a Master Password for the system.

Ensure that you remember these passwords, as they cannot be retrieved or reset. If you forget the password, contact Hortonworks Support for further assistance.

```
DataPlane will now setup a Super User with a username 'admin'.
This user can configure LDAP, add DataPlane Service Admins, enable services,
etc.
Setup a password for this user.
```

```
Enter password for DataPlane Services Super User:
Re-enter password:
```

```
DataPlane Services will now setup a Master password that is used to secure
the secret storage for the system.
```

```
Caution: The master password can be setup only once and cannot be reset
easily.
```

```
You will need to provide it for various admin operations. Hence please
remember what you enter here.
```

```
Enter master password for DataPlane Service (Minimum 6 characters long):
Reenter password:
```

4. When the initialization process completes, you can check the status of the docker containers using the following command:

```
docker ps
```

Sample output:

CONTAINER ID	IMAGE	STATUS	PORTS	COMMAND
a9ce135f1b4a	hortonworks/dp-app:1.2.0.0-388	Up 33 seconds	0.0.0.0:80->80/tcp,	"/bootstrap.sh"
	34 seconds ago		0.0.0.0:443->443/tcp, 9000/tcp	dp-app
3605425f7337	hortonworks/dp-cluster-service:1.2.0.0-388	Up 33 seconds	9009-9010/tcp	"/."
docker_service_...	35 seconds ago		dp-cluster-service	
80bff7d6d4f4	hortonworks/dp-db-service:1.2.0.0-388	Up 34 seconds	9000/tcp	"/."
docker_service_...	35 seconds ago		dp-db-service	
55658aefelf5	hortonworks/dp-gateway:1.2.0.0-388	Up 35 seconds	8762/tcp	"/."
docker_service_...	36 seconds ago		dp-gateway	
888df9acb2ea	hortonworks/dp-knox:1.2.0.0-388	Up 36 seconds	53/udp, 8300-8302/tcp,	"/usr/dp-
scripts/k..."	38 seconds ago		8400/tcp, 8301-8302/udp, 8500/tcp	knox
4146fe3dc416	consul:1.0.1	Up 38 seconds	8300-8302/tcp,	"docker-
entrypoint..."	39 seconds ago		8301-8302/udp, 8600/tcp, 8600/udp, 0.0.0.0:8500->8500/tcp	dp-consul-
			server	
d6elfc2c6d28	postgres:9.6.3-alpine	Up 51 seconds	5432/tcp	"docker-
entrypoint..."	53 seconds ago		dp-database	

Output descriptions:

Docker Container	Description
dp-app	DP Instance application (UI, web, etc.), accessible from port 443 (port 80 redirects to port 443)
dp-cluster-service	Powers how DP Platform talks to clusters
dp-db-service	Backend data store API in support of DP Apps
dp-gateway	Handles routing between DP Platform, DP Apps, Knox, etc.
knox	Runs a Knox instance that wraps the DataPlane instance. This is the AuthN enforcement point for DataPlane (SSO).
dp-consul-server	Handles the networking of the containers
dp-database	The PostgreSQL database instance that the DP instance uses by default. This is not used if you configure an external database.

- Browse to your DP Instance host and proceed to log in using the Super User admin account.

Note:

If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

```
https://<DP_host_FQDN>
```

As part of the installation process, data collection using cookies and other telemetry mechanisms is turned on by default. To disable all data telemetry, see the *DataPlane Administration Guide* for disabling data telemetry.

What to do next

You can now complete configuration of the DP Platform.

Related Tasks

(Optional) [Configure a TLS certificate](#)

Log in and configure DP Platform

To complete your installation, you must log in and configure the DP Instance for LDAP.

About this task

You must configure your DP Instance for LDAP and set up your initial DataPlane Admin users and groups.

Before you begin

Be sure you have your enterprise LDAP configuration available. Refer to *Enterprise LDAP requirements* for more information.

Procedure

- Browse to your DP Instance host.

If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

```
https://<DataPlane-host-FQDN>
```

- Log in using the Super User admin account you configured during initialization:

Username: admin

Password: Use the password specified during DataPlane initialization

After login, the initial “Onboarding Welcome” screen displays.

- Click **Get Started** to proceed with setting up authentication.

The Onboard/Configure LDAP page displays the Setup Authentication settings.

4. Enter your LDAP information.

See Enterprise LDAP requirements for more details on these settings and options.

If you are using LDAPS and a self-signed certificate, be sure to click *Upload certificate* and provide your certificate in order for DataPlane to connect to your LDAP instance. Use the corporate LDAP values that you collected during the preparatory steps.

5. Click **Save**.

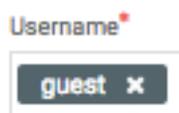
A success message displays on the page.

6. Add users and groups, which will be the initial members of the default DataPlane Admin role.

At least one user must be configured and must have at least the DataPlane Admin role, which is needed to add other users or groups, assign them roles, and configure services in DataPlane.

Tip:

You must click the name of the user when it displays and ensure it appears in the Users field on a dark background.



If the name appears on a white background, it means the name is not recognized and the action fails.

7. Click **Save & Login** to save your changes.

8. Log in as one of the DataPlane Admin users you added.

Results

You are now ready to proceed to managing your DP platform, including cluster registration and DP App installs.

What to do next

You can now install and configure additional DP Apps and manage your DP Instance.

Related reference

[Enterprise LDAP requirements](#)

Configure Knox and Ranger for DataPlane

Before proceeding further for using DataPlane, you must configure Knox SSO and Knox Gateway for DataPlane. It is also highly recommended to configure Ranger to restrict access to DataPlane.

Configure Knox SSO for DataPlane

This topic provides an overview of how to configure Knox SSO in your cluster to work with DataPlane. Refer to the Hortonworks Data Platform or Hortonworks DataFlow documentation for details that might be applicable to your specific cluster configuration and setup.

About this task

Note: As part of configuring Knox SSO to work with DataPlane, you will be setting up Knox topologies for token exchanges to allow your DP instance to communicate and handle SSO requests. It is strongly recommended that in your cluster, you configure Ranger to restrict access to these token topologies to be only from your DP instance. See Configure Ranger in your Cluster section for more information.

- You will be configuring Knox SSO in your cluster to work with your DP instance.

Before you begin

- You must have installed and configured DataPlane.
- Minimally, Knox SSO should be configured for Ambari.

Note: If you are using Ambari 2.7 or later, Ambari provides a helper “setup-sso” command to simplify the setup of Knox SSO for Ambari and certain cluster services. Refer to the *Ambari Security Guide* for more information.

- Knox host FQDN must be DNS addressable and available from your DataPlane environment. If your Knox configuration is setup for High Availability (HA) with more than one Knox instance running behind a proxy, the FQDN/IP of that proxy must be DNS addressable and available from your DataPlane environment.

If it is not, the Knox IP address must be in the `/etc/hosts` file on the DP environment. Refer to the *DP Administration Guide* for details on how to add Knox to the DataPlane environment hosts.

- You must have an SSL certificate (such as a `.pem` file) available and have access to the public key in the file.

Procedure

- In a terminal, SSH to the DP host.
- Navigate to `$DP_INSTALL_HOME/certs/`.

```
cd /usr/dp/current/core/bin/certs/
```

- Display the content of the `ssl-cert.pem` file.

```
cat ssl-cert.pem
```

- Copy and retain the DataPlane public key displayed in the certificate between “Begin Certificate” and “End Certificate”, because you need it in a succeeding step.

The public key looks similar to the following:

-----BEGIN CERTIFICATE-----

```
MIIEpDCCAowCCQDgyOmhg8r5wjANBgkqhkiG9w0BAQsFADAUMRIWEAYDVQQDDAlk
YXRhcGxhbmUwHhcVEBgwNjA2MDMzMzEyWhcNMTkwNjA2MDMzMzEyWjAUMRIWEAYD
VQQDDAlkYXRhcGxhbmUwggIiMyYu7ncDA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAo
fTf6/5drxlYa5EeHDetQo3I50Vx+Tj9jpd8t1x+3zJMO3xI6UCtHFizlxs8IToTw
V2BEOPH1K7qqRQjTagZtqNU7JNiEouBxO+1RYXdyaqhCIIUcgspso9W5C9T2aM6
HWx73MapoP5r4dRTpYCYITWJW7GkvJGhplIK51MhWD2dL9+bzsZ/sY9nwGJ2iUZ39
VGKMRc8TQlR2uwmRl2GziOCjMrfJIDDBxNm2xCbpYbNo8prle2EcntLgrw/EBNyA
Fbp8a506gL7TbbjngFM4iHhr7vVzbOzl14Qci+GLdTl51yNLER1k5sC/TEFpKBXP
IcY8UOuU6bbXrF/ZYR5FjYu7ncDOP0wDLz0lmLPwU27tE9D9SR+k0PYo8xvLIw31
iPsw0UuF4ouWJI9UaZ2i0vGJJcU8TtH7cOEyUL+ww0sCNHp5eI1hXgdCTkyFZv7s
2xWNA12L0TSD/49nHA9fIrXHYp5inmHrJiRP4wJlxFDvbZFltFepaG4I2pWhdrwY
2Sj0G084J294FNLad2xpoIacwaKWzdJ6HTvYo09Pqiu0YAchGicGCjwUU0omeK61
mZRmwENGYlzkXsrBAAGU1QuzZTFevA36d71iCctZOWKp7KeHL0e8RJ77ftfgEXvRT
WzmnVdOCfpflmaXoVhyATIRVAF5RDdnYdK+QUPDpVwIDAQABMA0GCSqGSIb3DQEBA
ViNN1eA4ICAQCNPcF0UJVAs70hAf3DE2IjdWoiUUr1x8jt5icWOS48Mw8mM9ADD
tGZ3IMrSB3qerixBN7t1FS/NCTXvS78imNXJT5xnJvNUYekWj/uLi4Vh9/Vg/1/2
zm7uLn5gTLwzDr9QGUF8Hb7+o/nW3pYLwrSoay5ApIysMd/mOykzFuLf4P7CHOME
cFImfN82xSD46W2zLbW/aJ5jdwXRa27L075TA91sqRR00tObn+vCBgsehvt1EBFy
U7puKS9fv2QHMS7rmt3ixhWw5AJ8wG9D4/nJq0cJhIKaqiDn9nqGiY8GBUwa/YAc
tAuIqo+LcvoIr/J+yc+7SXxDJXM+S1bS+1A0Fp/EdIMuDbey2T6Sabor4khzi78E
PeCBKHnZ2V8MTtcAXKw6RTSToBhIGozm9oRGU1xft61xAein+ba4UH1yZUanna7o
nBRQIKjSYC48oxYyWEA6HlZbtjy/uaBU3WP78mcoYUfFronK7fGkGvB8+xMYlYc+
aM1t87Z/KpY2d2CtDEG6qMl5wWTJqwocN5cYNwubgJM8vtluDlIhsezHjr1Tuv1x
988ztA/raNLF921ZAc5W1Xly5JF4z1lhZQZqDBpMsdo05HWKU4bgdLLoCN7bFVFPX
```

```
Sclm0TtyUSXblXfPqXqBfnPJNiimmLk1+Sm0xX9h+dOHfcNMSSNa5g==
```

```
-----END CERTIFICATE-----
```

5. On your cluster Knox host, create three topology files - token.xml, redirect.xml, and redirecttoken.xml topology files.

```
vi /etc/knox/conf/topologies/token.xml
```

```
vi /etc/knox/conf/topologies/redirect.xml
```

```
vi /etc/knox/conf/topologies/redirecttoken.xml
```

Note: The redirecttoken.xml topology file should be exactly same as the token.xml topology file. For security purposes, the TTL of the token should be kept very low. It is recommended to keep the value at 10 seconds.

6. Add the required content to the token.xml file on each cluster host running a Knox instance:
 - a) Add the basic topology content.

You can copy and paste the following content into the file and modify the content as needed.

```
<?xml version="1.0" encoding="UTF-8"?>
<topology>
  <uri>https://$KNOX_HOSTNAME_FQDN:8443/gateway/token</uri>
  <name>token</name>
  <gateway>
    <provider>
      <role>federation</role>
      <name>SSOCookieProvider</name>
      <enabled>true</enabled>
      <param>
        <name>sso.authentication.provider.url</name>
        <value>https://$KNOX_HOSTNAME_FQDN:8443/gateway/knoxssso/api/
v1/websso</value>
      </param>
      <param>
        <name>sso.token.verification.pem</name>
        <value>
          $ADD_THE_PUBLIC_KEY_HERE
        </value>
      </param>
    </provider>
    <provider>
      <role>identity-assertion</role>
      <name>HadoopGroupProvider</name>
      <enabled>true</enabled>
    </provider>
  </gateway>

  <service>
    <role>KNOXTOKEN</role>
    <param>
      <name>knox.token.ttl</name>
      <value>100000</value>
    </param>
    <param>
      <name>knox.token.client.data</name>
      <value>cookie.name=hadoop-jwt</value>
    </param>
```

```

    <param>
      <name>main.ldapRealm.authorizationEnabled</name>
      <value>true</value>
    </param>
  </service>
</topology>

```

Provide the following details in the topology file:

Property	Values	Description
sso.token.verification.pem	Certificate	Paste in the public key value that you copied in a previous step, replacing \$ADD_THE_PUBLIC_KEY_HERE (be sure to exclude the BEGIN CERTIFICATE and END CERTIFICATE text).
knox.token.ttl	milliseconds	Expiry time of the token. A value of -1 means no expiry. For security purposes, the TTL of the token should be kept very low. It is recommended to keep the value at 10 seconds (10000).
sso.authentication.provider.url	Knox SSO URL	The URL to your cluster Knox SSO endpoint. Replace \$KNOX_HOSTNAME_FQDN with the fully qualified domain name of the host.
identity-assertion	true false	Enables the “HadoopGroupProvider” Hadoop user-group mapping, which identifies the groups to which users belong

Note: The `authorization=XASecurePDPKnox` parameter and `main.ldapRealm.authorizationEnabled=true` parameter enable Ranger authorization with the token topologies in Knox.

7. Add the required content to the `redirect.xml` file on each cluster host running a Knox instance:
 - a) Add the basic topology content.
 - b) You can copy and paste the following content into the files and modify the content as needed.

```

<topology>
  <name>tokensso</name>
  <gateway>
    <provider>
      <role>federation</role>
      <name>JWTProvider</name>
      <enabled>true</enabled>
    </provider>
    <provider>
      <role>identity-assertion</role>
      <name>Default</name>
      <enabled>true</enabled>
    </provider>
  </gateway>
  <service>
    <role>KNOXSSO</role>
    <param>
      <name>knoxsso.cookie.secure.only</name>
      <value>true</value>
    </param>
    <param>
      <name>knoxsso.token.ttl</name>
      <value>600000</value>
    </param>
  </service>
</topology>

```

```

        <name>knoxsso.redirect.whitelist.regex</name>
        <value>^https?:\\\/\/(DOMAIN_OF_CLUSTER|localhost|127\.0\.0\.1|
0:0:0:0:0:0:0:1|::1):[0-9].*$</value>
        </param>
    </service>
</topology>

```

Provide the following details in the topology file:

Property	Values	Description
knoxsso.cookie.secure	true false	This sets if a session cookie is require or not. If your cluster Ambari is configured for SSL, then set this value to true. Otherwise, set to false. This value is true if the secure cookie is required.
knox.token.ttl	milliseconds	Expiry time of the token. A value of -1 means no expiry.
knoxsso.redirect.whitelist.regex	regex	This should have the regex which matches the URL in the to or original Url query parameter for the two separate calls. Be sure to replace "DOMAIN_OF_CLUSTER" in the example regex provided.

8. Perform a secure copy of the three topology files to a Knox-enabled node on the cluster.
9. Verify that Knox has picked up the files:
 - a) Log in to the Knox-enabled node.
 - b) Ensure that a directory called token.topo.<number> is present in the path /var/lib/knox/data-<version>/deployments/.
If the files are not present, verify that the content in the token.xml file is correct. You can check the Knox gateway logs for error information.

Related Concepts

[Knox SSO with DataPlane clusters](#)

Related Tasks

[Advanced: Add host entries to the DataPlane environment](#)

Related Information

[Defining Cluster Topologies](#)

Configure Knox Gateway for DataPlane

DP Platform communicates with services on the cluster like DP Agents, Ambari, Atlas, Ranger, etc as well as DP Agents used by DP Apps (for example: DLM Engine for DLM and Profiler for DSS). To eliminate DataPlane communicating directly to all the cluster service endpoints, you can configure Knox Gateway as a proxy to your cluster services.

About this task

Note: If you are using TLS wire encryption on your clusters, you must configure Knox Gateway to proxy requests to and from DP host.

This topic provides an overview of how to configure Knox Gateway proxy in your cluster services for DataPlane communication. If you configure Knox Gateway as the proxy for communication, be sure all DP services are configured through the gateway. Refer to the Hortonworks Data Platform or Hortonworks DataFlow documentation for details that might be applicable to your specific cluster configuration and setup. .

- You will be configuring Knox Gateway proxy in your cluster to work with your DP instance.

Before you begin

- You must have installed and configured DataPlane.
- You must have configured Knox Gateway on your clusters.

Knox host FQDN must be DNS addressable and available from your DataPlane environment. If not, the Knox IP address must be in the `/etc/hosts` file on the DP environment. Refer to the *DataPlane Administration* guide for details on how to add Knox to the DataPlane environment hosts.

Procedure

1. On your cluster Knox host, navigate to the Knox topologies directory.

```
cd /etc/knox/conf/topologies
```

2. Create a DataPlane proxy topology file.

```
vi dp-proxy.xml
```

3. Add the host name for each of the services listed in the file, based on where that service is running in your cluster.

Tip: At this point, you can add to the file the DP service agents that you plan to install, or you can add them later.

Important:

- Do not modify the URL in the provider section of the file.
- Be sure to keep this file updated if you move services or add services in your cluster.

The `<localhost>` entry in the following example might be something like `ctr-e138-1518143905142-369209-01-000005.hw.x.site:20070`.

Topology `dp-proxy.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<topology>
  <gateway>
    <provider>
      <role>federation</role>
      <name>SSOCookieProvider</name>
      <enabled>>true</enabled>
      <param>
        <name>sso.authentication.provider.url</name>
        <value>https://localhost:8443/gateway/knoxssso/api/v1/webssso</
value>
      </param>
    </provider>
    <provider><role>identity-assertion</role>
      <name>Default</name>
      <enabled>>true</enabled>
    </provider>
  </gateway>

  <service>
    <role>WEBHDFS</role>
    <url>http://<localhost>:20070/webhdfs</url>
  </service>
  <service>
    <role>WEBHCAT</role>
    <url>http://<localhost>:20111/templeton</url>
  </service>
  <service>
    <role>AMBARI</role>
    <url>http://<localhost>:8080</url>
```

```

</service>
<service>
  <role>AMBARIUI</role>
  <url>http://<localhost>:8080</url>
</service>
<service>
  <role>RANGER</role>
  <url>http://<localhost>:6080</url>
</service>
<service>
  <role>RANGERUI</role>
  <url>http://<localhost>:6080</url>
</service>
<service>
  <role>ATLAS</role>
  <url>http://<localhost>:21000</url>
</service>
<service>
  <role>ATLAS-API</role>
  <url>http://<localhost>:21000</url>
</service>
<service>
  <role>BEACON</role>    ##The DLM Engine
  <url>http://<localhost>:25968</url>
</service>
<service>
  <role>HIVE</role>
  <url>http://<localhost>:10001/cliservice</url>
</service>
<service>
  <role>RESOURCEMANAGER</role>
  <url>http://<localhost>:8088/ws</url>
</service>

<service>
  <role>PROFILER-AGENT</role>    ##The DSS Agent
  <url>http://<localhost>:21900</url>
</service>
</topology>

```

Configure Ranger to restrict access to DataPlane

It is strongly recommended that in your cluster, you configure Ranger to restrict access to these token topologies to be only from your DP instance, in order to restrict access to only authorized users of DataPlane Platform.

About this task

As part of configuring Knox SSO to work with DataPlane, you setup Knox topologies to allow your DP instance to communicate and handle SSO request token between DP and your cluster.

Note: This is the basic Ranger policy setup to restrict access to the Knox topology to only DataPlane. Additional policies may be recommended or required based on the DP Apps (and their requisite Cluster Agents) you use.

- You will be configuring a Ranger policy to restrict access to Knox SSO token topologies to DataPlane users and your DP Instance.
- You must have installed and configured DataPlane.
- You must have configured Knox SSO for DataPlane. See *Configuring Knox SSO for DataPlane* for more information.

- You must have Ranger installed and configured in your cluster.
- Be sure to also add the authorization role to the token topologies you configured for DP in your Knox SSO setup.

```
<provider>
  <role>authorization</role>
  <name>XASecurePDPKnox</name>
  <enabled>true</enabled>
</provider>
```

Procedure

1. In your cluster, navigate to the Ranger UI and log in.
2. Click **Access Manager**, and then click the Knox repository link, for example: **<cluster-name> Policies**.
3. Click **Add New Policy**, and then enter the following values:

Parameter	Value
Policy Type	Access
Knox Topology	token
Knox Service	*
4. Enter groups or user names in **Select Group** or **Select User**.
5. Optional: Under Policy Conditions click **Add Condition** and enter the IP addresses of the DataPlane host. This adds an IP-based filter to ensure that only known DataPlane Core hosts can access cluster services through the token topology.
6. Under Permissions, click **Add Permission** and select **Allow**.

Upgrade DataPlane from 1.1.0 to 1.2.0

Make sure you take regular backups of the instance before you proceed with the upgrade procedure.

Procedure

1. Back up your existing DataPlane repository (dp.repo) file.
2. Download the upgrade repository tarball to the repository folder:

```
wget -nv <upgrade-repo-URL> -o /etc/yum.repos.d/dp.repo
```

3. Verify that the repository is downloaded:

```
yum search dp-core
```

You should see two dp-core repositories.

4. Update the repository by running the following command:

```
yum update dp-core
```

You should see two versions of the DP Platforms.

5. Run the upgrade command as follows:

```
./dpdeploy.sh upgrade --from /usr/dp/1.1.0.0-392/core/bin/
```

A message appears that the upgrade process is complete.

6. Once the upgrade process is successfully completed, add hosts using the following command:

```
./dpdeploy.sh utils add-host <Host URL> <Host FQDN>
```

Make sure you run this command for each host. The upgrade process does not automatically add hosts.

The following message appears:

```
This will update database schema which cannot be reverted. All backups
need to be made manually. Please confirm to proceed (yes/no):
```

7. Enter yes to continue.
8. When prompted, enter the master password for DataPlane.
Note: The master password should be the same as the password used in the previous version of DataPlane.
9. If asked, enter the password for supplied certificates. You will be prompted only if you have provided certificates in the config.env.sh file.

Troubleshooting DataPlane Installation

You can troubleshoot various installation issues such as cluster registration errors, logging issues, and Docker container errors.

Cluster Registration Error Messages

Following are errors you might encounter while registering a cluster in DataPlane on the Add Your Cluster page. Some possible causes and possible resolutions are also included.

Cluster is not reachable

DataPlane containers are not able to resolve a provided hostname or use the IP address to connect to the machine.

DNS resolution is not setup.

There are firewall or other networking restrictions that are preventing access.

Sample Message:

Failed: This is not a valid Ambari URL.

Procedure

1. Verify that the specified hostname or IP address is valid and reachable from the DP host machine.
2. If the hostname or IP address is reachable, try adding the hostname resolution to the DataPlane container using the `./dpdeploy.sh utils add-host <ip> <host>` command.
3. Verify if network connectivity settings, such as firewalls, are configured correctly.

Knox is not set up on the HDP cluster, or Ambari credentials are incorrect for 'seeded user' mode

This error occurs when the cluster is reachable, but authentication is failing.

Sample Message:

Unable to connect, please retry. DataPlane could not retrieve cluster information.

Possible Causes:

- Knox is not set up on the cluster.
- The user wants to use the less secure 'seeded user' mode, but the credentials of the seeded user (user name or password) are not setup in DataPlane.

Procedure

1. Validate that Knox is configured correctly as per documentation.
2. If seeded user mode is being used (for evaluation purposes), add the correct credentials to DataPlane using `./dpdeploy.sh utils update-user ambari`.

Knox setup is incorrect on the HDP cluster

This error indicates that the cluster being registered has Knox enabled, but the communication from DataPlane to Knox is failing.

Sample Message:

Failed: There was an error fetching information from Ambari.

Possible Causes:

- The Knox token service is not properly configured.
- The public key of DataPlane is not set up correctly in the Knox topology.

Procedure

Validate that the Knox configuration for the token topology is done correctly, following the instructions in *Configuring Knox and Ranger for DataPlane*.

Cannot register a cluster, other causes

If you cannot register a cluster with DataPlane and none of the errors mentioned are the cause, the following might apply.

Procedure

1. Verify that the hostname where Knox is running is valid and reachable from the DP host machine. If it is reachable, try adding the hostname resolution to the DP container using the `./dpdeploy.sh utils add-host <ip> <host>` command.
2. Verify that network connectivity settings, such as firewalls, are correctly configured.
3. Verify that the username is an Ambari Admin on the cluster. If not, make the user an Ambari Admin user, by logging into Ambari, selecting the user, and providing Admin privileges.
4. See the [DataPlane Support Matrices](#) and verify that you are running a supported configuration.

Cluster status displays incorrectly on Details page

On the Cluster Details page, sometimes the Status of a cluster displays in gray, instead of red or green.

This generally indicates a timeout issue, in which DataPlane is not able to refresh the cluster details correctly. Manually refreshing the cluster information should fix the problem.

Procedure

1. Click the **Actions** icon at the end of the row.
2. Click **Refresh**.

Refresh of the cluster status might take several seconds.

Logging in using the DataPlane local admin role

The local admin role allows you to perform administrative activities and troubleshoot problems when access through LDAP and Knox is not available. The local admin is also the role you use to log in to DataPlane the first time, before LDAP is configured in DataPlane for SSO.

About this task

When you log in as the local DataPlane Admin, you bypass Knox.

For login, the default username is “admin”. The password you use to log in is set during the installation process.

Procedure

Log in by appending /sign-in to the DataPlane login URL, for example:

```
http://dataplane-host-name/sign-in
```

wget command is not available

The wget command is not installed on the system.

Use the command `yum install wget` to install the wget tool.

Delete and clean up Docker containers

If you have problems with your installation or want to update a DataPlane container, you can delete the Docker containers and then install the new images.

About this task

Important: Performing this task deletes all of your DP Platform database content, so you will have to reconfigure the LDAP and cluster registration settings after reinstalling the Docker containers.

For information about the commands and options supported by `./dpdeploy.sh`, use the command-line help.

Before you begin

You must be root user to perform this task.

Procedure

1. `cd /usr/dp/current/apps/dlm/bin`
2. `./dlmdeploy.sh destroy`
3. `cd /usr/dp/current/core/bin`
4. `./dpdeploy.sh destroy --all`
5. `docker ps`

This ensures that no containers are running. If you see any, kill them with `docker kill`.

6. Go to *Initialize DataPlane* and run the original DataPlane deployment commands starting with `./dpdeploy.sh init --all`.