H2O.ai

H2O

Steam

Preview Release

Table of Contents

Introduction

This document describes how network admins can install and set up Steam. Refer to the following sections:

- Steam Installation and Setup
- User Management

Steam Installation and Setup

This section describes how to install and start Steam. Refer to the following topics:

- Requirements
- Installation
- RedHat 6.7 Installation
- Start Steam

Requirements

- Web browser with an Internet connection
- Steam tar for OS X, Linux, or RedHat 6.7

    - available from h2o.ai/steam/
- JDK 1.7 or greater

- PostgreSQL 9.1 or greater

    - available from PostgreSQL.org
- H2O jar file for version 3.10.0.3 or greater

    - available from the H2O Download page
    - If necessary, follow the instructions on the http://www.h2o.ai/download/h2o/python or http://www.h2o.ai/download/h2o/r page to upgrade H2O for Python or R.

Linux and Mac OS X Installation

Perform the following steps to install Steam on Linux or Mac OS X.

Note: This installation should only be performed on a YARN edge node.

Download the Software to the Edge Node

1. Go to h2o.ai/steam/ to download Steam. Be sure to accept the EULA.

2. Change directories to the Steam download folder for your OS (Linux or OS X), then copy the folder to your edge node. Enter the correct password when prompted.

```
cd ~/Downloads/steam-0/steam-0.1.0-darwin-amd64
scp -r steam-0.1.0-darwin-amd64 <user>@<yarn_edge_node>:~
```

You need to create the Steam superuser before you can start steam. The next sections describe starting PostgreSQL and then creating the Steam superuser.

Start PostgreSQL

Open a terminal window and run the following command to start PostgreSQL. This should be started from the folder where PostgreSQL was installed.

```
postgres -D /usr/local/var/postgres
```

Create the Steam Superuser

The Steam superuser is responsible for maintaining Steam clusters and for setting up roles, workgroups, and users. This step creates the superuser for the Steam database and then creates the database.

The example below creates a Steam superuser with a password `superuser`, and then creates the Steam database. *If prompted, do not enter a password*.

```
createuser -P steam
Enter password for new role:
Enter it again:
```

Create the Steam Database

The following commands show how to change directories to the Steam /var/master/scripts folder, and then create the database.

```
cd steam-master-darwin-amd64/var/master/scripts
./create-database.sh
```

RedHat 6.7 Installation

Perform the following steps to install Steam on RedHat 6.7. This is currently the only supported version of RedHat and can be downloaded using the following command:

```
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

Note: This installation should only be performed on a YARN edge node.

Install and Start PostgreSQL

1. SSH to the Steam edge node, and then add the following line to the **[main]** section of /etc/yum/pluginconf.d/rhnplugin.conf.

```
ssh <user>@<yarn_edge_node>
exclude=postgresql*
```

   i. Run the following commands to install PostgreSQL.

```
sudo yum localinstall http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-redhat94-9.4-2.noarch.rpm
sudo yum install postgresql94-server
```

   i. After the PostgreSQL server is installed, run the following command as the postgres user.

```
/usr/pgsql-9.4/bin/initdb -D /var/lib/pgsql/9.4/data
```

   i. Run the following command as root to start PostgreSQL.

```
sudo /etc/init.d/postgresql-9.4 start
```

   i. Run the following commands as the postgres user to create a Steam user. Note that a password is not required and can be left blank.

```
sudo -i -u postgres
createuser --interactive -P steam
Enter password for new role:  # remember this password
Enter it again:
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) y
Shall the new role be allowed to create more new roles? (y/n) n
```

Create the Steam Database

Now that PostgreSQL is running and a Steam user is crated, the following commands show how to change directories to the Steam /var/master/scripts folder, and then create the Steam database.

```
cd steam-master-darwin-amd64/var/master/scripts
./create-database.sh
```

Set Up .pgpass for PostgreSQL

As the Steam user, set up .pgpass for postgres. This is done by editing **~/.pgpass** and appending the line `*:*:*:steam:pa$$word`

```
chmod 600 ~/.pgpass
```

Start Steam

After Steam is installed on the YARN edge node and a superuser is created, the next step is to provide the designated Steam superuser with the URL of the edge node and the superuser login credentials. The superuser can then start Steam and begin creating roles, workgroups, and users.

1. SSH into the YARN edge node where the Steam package was copied.

   ```
   ssh <user>@<yarn_edge_node>
   ```

2. Start the Steam compilation and scoring service. Be sure to include the `--superuser-name=superuser` and `--superuser-password=superuser` flags. (Or provide a more secure password.) This starts Steam on localhost:9000 and creates a Steam superuser. The Steam superuser is responsible for creating roles, workgroups, and users and maintains the H2O cluster.

   ```
   ./steam serve master --compilation-service-address=<yarn_edge_node>:<port> --scoring-service-address=<ip_address> --superuser-name=superuser --superuser-password=superuser
   ```

   Note: This starts the Steam web service on **:** and the scoring service on . Use `./steam help serve master` or `./steam serve master -h` to view additional help information.

Now that Steam is up and running, you can log in to the machine that is running Steam and use the CLI to create additional roles, workgroups, and users. Refer to the User Management section.

User Management Overview

Before using Steam, it is important to understand User Management within your YARN environment. In Steam, User Management is supported in a PostgreSQL database. The User Management functions in Steam determine the level of access that users have for Steam features. The Steam database supports setup via CLI commands. Refer to the CLI Command Reference Appendix for a list of all available CLI commands.

For more information on Steam User Management, refer to the following sections.

- Terms
- Privileges/Access Control
- Authorization
- User Management Workflow
- User Management Example
- Next Steps

Terms

The following lists common terms used when describing Steam User Management.

- **Entities** represent *objects* in Steam. Examples of entities include Roles, Workgroups, Identities, Clusters, Projects, Models, and Services (engines).

- **Identities** represent *users* in Steam. Users sign in using an Identity, and then perform operations in Steam.

- **Permissions** determine what operations you can perform. Examples of permissions include *Manage Clusters*, *View Clusters*, *Manage Models*, *View Models*, and so on.

- **Privileges** determine the entities that you can perform operations on (i.e., data / access control).

Privileges/Access Control

Privileges are uniquely identified by the entity in question and the kind of privilege you have on the entity.

The following privileges are available on an entity:

- **Own** privileges allow you to share, view, edit, and delete entities.

- **Edit** privileges allow you to view and edit entities, but not share or delete them.

- **View** privileges allow you to view entities, but not share, edit, or delete them.

When you create an entity, you immediately *Own* it. You can then share this entity with others and award them either *Edit* or *View* privileges. Entities are allowed to have more than one owner, so you can also add additional owners to entities.

The following table lists the kind of privileges you need in order to perform specific operations on entities:

```
Entity              Own  Edit View
----------------------------------
Role
  Read               x    x    x
  Update             x    x
  Assign Permission  x    x
  Delete             x
  Share              x

Workgroup
  Read               x    x    x
  Update             x    x
  Delete             x
  Share              x

Identity
  Read               x    x    x
  Assign Role        x    x
  Assign Workgroup   x    x
  Update             x    x
  Delete             x
  Share              x

Cluster
  Read               x    x    x
  Start/Stop         x

Project
  Read               x    x    x
  Assign Model       x    x
  Update             x    x
  Delete             x
  Share              x

Engine, Model
  Read               x    x    x
  Update             x    x
  Delete             x
  Share              x
```

Authorization

Permissions and privileges are set up using Roles and Workgroups, respectively.

- Identities cannot be linked directly to permissions. For that, you'll need Roles.

- Identities cannot be linked directly to privileges on entities. For that, you'll need Workgroups, i.e. when you share entities with others, you would be sharing those entities with workgroups, not individuals.

Roles

A **Role** is a named set of permissions. Roles allow you define a cohesive set of permissions into operational roles and then have multiple identities *play* those roles, regardless of access control. For example:

- a *Data Scientist* role can be composed of the permissions *View Clusters*, *Manage Models*, *View Models*.
- an *Operations* role can be composed of the permissions *View Models*, *View Services*, *Manage Services*,
- a *Manager* role can be composed of the permissions *Manage Roles*, *View Roles*, *Manage Workgroups*, *View Workgroups*

Workgroups

A **Workgroup** is a named set of identities. Workgroups allow you to form collections of identities for access control purposes. For example, a *Demand Forecasting* workgroup can be composed of all the users working on demand forecasting, regardless of their role. This workgroup can be then used to control access to all the clusters, projects, models and services that are used for demand forecasting.

User Management Workflow

The steps below provide a common workflow to follow when creating users. This workflow is followed in the example that follows.

1. Define roles based on operational needs.
2. Define workgroups based on data / access control needs.
3. Then add a new user:

   - Create the user's identity.
   - Associate the user with one or more roles.
   - Optionally, associate the user with one or more workgroups.

User Management Example

The following example creates sample roles, workgroups, and users using the CLI. Refer to the CLI Command Reference Appendix for information about all of the commands available in the CLI.

1. Log in as the Steam superuser on the machine that is running Steam.

   ```
   ./steam login <yarn_edge_node>:<port> --username=superuser --password=superuser
   ```

2. Create an engineer role, and link that role to permissions. Note that you can run `./steam get permissions` to view a list of available permissions.

   ```
   ./steam create role engineer --desc="a default engineer role"
   ```

   ```
   ./steam link role engineer ViewModel ViewProject ViewWorkgroup
   ```

3. Create a data scientist role, and link that role to permissions.

   ```
   ./steam create role datascience --desc="a default data scientist role"
   ```

   ```
   ./steam link role datascience ManageProject ManageModel ViewCluster
   ```

4. Create preparation and production workgroups.

   ```
   ./steam create workgroup preparation --desc="data prep group"
   ```

   ```
   ./steam create workgroup production --desc="production group"
   ```

5. Create two users - Bob and Jim.

   ```
   ./steam create identity bob bobSpassword
   ```

   ```
   ./steam create identity jim j1mSpassword
   ```

6. Link Bob to engineer role; link Jim to datascience role.

   ```
   ./steam link identity bob role engineer
   ```

   ```
   ./steam link identity jim role datascience
   ```

7. Link Bob to preparation workgroup; link Jim to production workgroup.

   ```
   ./steam link identity bob workgroup preparation
   ```

   ```
   ./steam link identity jim workgroup production
   ```

Next Steps

Now that you understand User Management, you can create your own roles, workgroups, and users. Once created, be sure to provide your users with their Steam login credentials.