Steam Standalone Installation

This document describes how to get up and running with Steam without the need for a local running instance of YARN. These instructions will walk through the following procedures:

- Installing and starting Steam, the Compilation Service, and H2O
- Adding Roles, Workgroups, and Users to the database
- Building a simple model in Python (Optional for users who don't have an existing demo.)
- Deploying the model using Steam
- Making predictions using the Steam Prediction Service

During this demo, four terminal windows will remain open for the Steam, Scoring, H2O, and postgres services. A fifth terminal window will be used to run H2O commands in the Python or R example.

Finally, these steps were created using H2O version 3.10.0.3, and that version resides in a Downloads folder. Wherever used, this version number and path should be adjusted to match your version and path.

Requirements

- Web browser with an Internet connection
- JDK 1.7 or greater
- PostgreSQL 9.1 or greater

    - available from [PostgreSQL.org (https://www.postgresql.org/)](https://www.postgresql.org/)
    - Note that if you get a permissions error when running `brew install postgres`, then you may need to change permissions using either `rm ~/.steam/config` or `brew doctor`.

- Steam tar for Linux or OS X

    - available from [www.h2o.ai/steam (http://www.h2o.ai/steam)](http://www.h2o.ai/steam)

- H2O jar file for version 3.10.0.3 or greater

    - available from the [H2O Download (http://www.h2o.ai/download/h2o/choose)](http://www.h2o.ai/download/h2o/choose) page

Optional

The following are required if you use a Python or R demo.

Python

- A dataset that will be used to generate a model. This demo uses the well-known iris.csv dataset with headers (available online), and the dataset is saved onto the desktop.
- Python 2.7

R

- A dataset that will be used to generate a model.
- Comprehensive R Archive Network (R). Available from [https://cran.r–](https://cran.r-)

.

Starting Steam

This section describes how to set up and start Steam and start the Steam CLI for user management. Five terminal windows will be open the first time you run this setup; four terminal windows will be open for subsequent logins.

1. Go to www.h2o.ai/steam (http://www.h2o.ai/steam) and download the Steam package. Be sure to accept the EULA.

2. Open a terminal window and untar the steamY binary. Note that the command below untars the OS X binary. Replace `darwin` with `linux` in the steps that follow to build on Linux.

   ```
   tar xvf steamY-master-darwin-amd64.tar.gz
   ```

3. Open a second terminal window and start PostgreSQL. This should be started from the folder where PostgreSQL was installed.

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

4. Open a third terminal window to create a new user for the Steam database and then create the database. The commands below only need to be performed once. The example below creates a steam **superuser**. *If prompted, do not enter a password*.

   ```
   createuser -P steam
   Enter password for new role:
   Enter it again:
   # Change directories to the Steam /var/master/scripts folder.
   cd steam-master-darwin-amd64/var/master/scripts
   ./create-database.sh
   ```

5. Change directories to your Steam directory, and start the Jetty server.

   ```
   cd steam-master-darwin-amd64
   java -jar var/master/assets/jetty-runner.jar var/master/assets/ROOT.war
   ```

   > **Note**: The Jetty server defaults to port 8080. You can optionally provide a `--port` value for **jetty-runner.jar**.

6. Open a fourth terminal window. From within the **steam-master-darwin-amd64** folder, 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 --superuser-name=superuser --superuser-password=superuser
   ```

> **Note**: This starts the Steam web service on `localhost:9000`, the compilation service on `localhost:8080` (same as the Jetty server), and the scoring service on `localhost`. You can change these using `--compilation-service-address=<ip_address:port>` and `--scoring-service-address=<ip_address>`. Use `./steam help serve master` or `./steam serve master -h` to view additional options.

7. Open a fifth terminal window to run CLI commands. From within the Steam folder, log in to the machine running Steam (localhost:9000). Use the superuser login and password that you created in the previous step.

```
./steam login localhost:9000 --username=superuser --password=superuser
```

8. Run the following to verify that the CLI is working correctly.

```
./steam help
```

At this point, you can open a browser and navigate to localhost:9000. Note that you may be prompted to once more provide the login credentials supplied in Step 7.

The next section describes how to add additional users to the Steam database.

Adding Roles, Workgroups, and Users

The following example creates sample roles, workgroups, and users using the CLI. Refer to the [CLI Command Reference Appendix (../docs/CLIAppendix.md)](../docs/CLIAppendix.md) for information about all of the commands available in the CLI. These commands are run from the terminal window used to log in to Steam ([Step 7](#) above).

```
    # Create engineer role and link that role to permissions
    ./steam create role engineer --desc="a default engineer role"
    ./steam link role engineer ViewModel ViewProject ViewWorkgroup

    # Create 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

    # Create preparation and production workgroups
    ./steam create workgroup preparation --desc="data prep group"
    ./steam create workgroup production --desc="production group"

    # Create two users - Bob and Jim
    ./steam create identity bob bobSpassword
    ./steam create identity jim j1mSpassword

    # Link Bob to engineer role; link Jim to datascience role
    ./steam link identity bob role engineer
    ./steam link identity jim role datascience

    # Link Bob to preparation workgroup; link Jim to production workgroup
    ./steam link identity bob workgroup preparation
    ./steam link identity jim workgroup production
```

Starting H2O and Running a Model

In order to create a project in Steam, your cluster must already have at least a single dataset. This section describes how to start H2O and runs a small Python demo for adding a dataset and building a model.

Start H2O

1. Open another terminal window. Navigate to the folder with your H2O jar file and start H2O. This will create a one-node cluster on your local machine on port 54321.

```
cd ~/Downloads/h2o-3.10.0.3
java -jar h2o.jar
```

Build a Model

The following steps show how to build model using the Iris dataset and the GBM algorithm. The steps will be run using H2O in Python. Once created, the model can be selected in Steam when creating a new project.

> **Note**: The rest of section can be skipped if you already have demo steps that you use in R, Python, or Flow. If you use another demo, be sure that you initialize H2O on your local cluster so that the data will be available in Steam.

Additional demos for Python are available [here (https://github.com/h2oai/h2o-3/tree/master/h2o-py/demos)](https://github.com/h2oai/h2o-3/tree/master/h2o-py/demos).

Demos for R are available [here (https://github.com/h2oai/h2o-3/tree/master/h2o-r/demos)](https://github.com/h2oai/h2o-3/tree/master/h2o-r/demos).

A demo of Flow can be viewed [here (https://www.youtube.com/watch?feature=player_embedded&v=wzeuFfbW7WE)](https://www.youtube.com/watch?feature=player_embedded&v=wzeuFfbW7WE).

1. Open a terminal window. Change directories to the H2O folder, and start Python. Import the modules that will be used for this demo.

```
$ cd ~/Downloads/h2o-3.10.0.3
$ python
>>> import h2o
>>> from h2o.estimators.gbm import H2OGradientBoostingEstimator
```

2. Initialize H2O using localhost and port 54321. (Note that if started Steam on a different machine, then replace `localhost` with the IP address of that machine.)

```
>>> h2o.init(ip="localhost", port=54321)
----------------------------   -------------------------------------
H2O cluster uptime:            2 minutes 37 seconds 168 milliseconds
H2O cluster version:           3.8.2.8
H2O cluster name:              user
H2O cluster total nodes:       1
H2O cluster total free memory: 3.35 GB
H2O cluster total cores:       8
H2O cluster allowed cores:     8
H2O cluster healthy:           True
H2O Connection ip:             127.0.0.1
H2O Connection port:           54321
H2O Connection proxy:
Python Version:                2.7.9
----------------------------   -------------------------------------
```

3. Upload the Iris dataset. Note that in this example, Python is running from the Downloads folder, and the Iris dataset is on the Desktop:

```
>>> df=h2o.upload_file("../../Desktop/iris.csv")
```

4. Specify the configuration options to use when building a GBM model.

```
>>> gbm_regressor =
H2OGradientBoostingEstimator(distribution="gaussian", ntrees=10,
max_depth=3, min_rows=2, learn_rate="0.2")
```

5. Train the model using the Iris dataset (`df` object) and the GBM configuration options.

```
>>> gbm_regressor.train(x=range(1, df.ncol), y=0, training_frame=df)
```
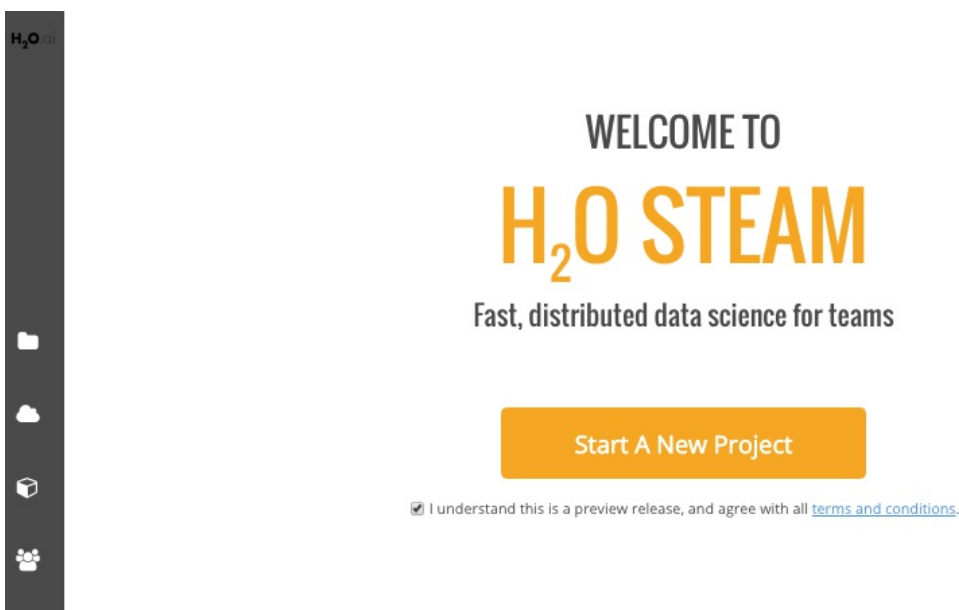
6. Optionally view the model details.

```
>>> gbm_regressor
```

Once created, the model will be visible in the Steam UI.

Creating a Steam Project

Now that you have added files to your H2O cluster, you can create your first Steam project. Point your browser to the Steam URL, for example, http://localhost:9000/.

The first time you log in to Steam, an empty Steam page will display, prompting you to start a project. Be sure to accept the terms and conditions in order to continue.

1. Click **Start a New Project**. This opens a page allowing you to start a new project from scratch or to begin importing models into your Steam environment.

NEW PROJECT

Steam organizes your data sets, your models, and your deployment configurations into one cohesive project. This enables you to:

- Visually compare all models within a project

- Manage how models from a project gets deployed

- Track the history of model deployment

Import Existing Models

Choose this option if you already have H2O models built and stored in a H2O cluster in your network.

Start Import

Start from Scratch

Choose this option if this is a completely new project.

Create New Project

2. To start a new project from scratch, click **Create New Project**. This opens a page showing you the available H2O clusters. When you first log in to Steam, the list of clusters will be empty. Enter your cluster IP address, then click **Connect**. Once connected, this will immediately populate the current list of clusters.
3. Select the H2O frame from the Datasets dropdown, then select the Category.
4. Select the checkbox beside the model(s) to import into the Steam project.
5. Specify a name for the project.

6. Click **Create Project** when you are done. Upon successful completion, the Models page will be populated with the model(s) that you added to your project.

You can perform the following actions directly from the models page:

- Import a new model
- View model details
- Label a model as a test, staging, or production model
- Deploy the model (See next section.)

Deploying a Model in Steam

1. On the Models page, click the **deploy model** link for the model that you want to deploy.
2. Specify a service name for the deployment, then click **Deploy**.

3. Upon successful completion, a scoring service will be created for this deployed model. Click the **Deployment** menu option on the left navigation to go to the Deployment page.

Making Predictions

1. To reach the scoring service, click the IP address link listed under the Deployed Services. This opens Steam Prediction Service tool. The fields that display on the Prediction Service tool are automatically populated with field information from the deployed model.

Prediction Service   Steam

Select input parameters, OR enter your own custom query string to predict

⇥ **MODEL INPUT PARAMETERS**   ▫ **PREDICTION RESULTS**

Parameters

1. sepal_wid

2. petal_len   **Model Runtime Stats**

3. petal_wid   Service started   2016-08-03 22:21:20 UTC
                                       Uptime   8 m 26 s
4. class   ↕

                                          MORE STATS

Query String

The parameters above gets automatically built into a
REST API query string. You can also input your own
string if that's easier for you.

http://172.16.2.124:46827/predict?   ▣

        PREDICT   CLEAR

2. Make predictions by specifying input values based on column data from the original
   dataset. This automatically populates the fields in the query string. (Note that you can
   optionally include input parameters directly in the query string instead of specifying
   parameters.)

3. Click **Predict** when you are done.

   Note: Use the **Clear** button to clear all entries and begin a new prediction. Use
   the **More Stats** button to view additional statistics about the scoring service
   results.

Stopping the Steam Database

When you are finished using Steam, press Ctrl+C in each of the Steam, Compilation Service, and
postgres terminal windows to stop the services end your session.

What's Next?

Now that you have completed your first demo, you are ready to begin creating models using your
own data.