**TCO Admin Manual**

**Summary**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <https://cyberinsight.hpc.uiowa.edu:8000>   |  |  |  | | --- | --- | --- | | **ID** | **PW** | **Type** | | eager | eagercyberinsight | admin | | testuser | testuser | user | | brogers | cyberinsight | user | | danareed | cyberinsight | user | | fleagle | cyberinsight | user | | kangplee | cyberinsight | user |   Admin manual (this document): <https://docs.google.com/document/d/12UU9sE7MJiLlsfl6Wddw_ACR7Bek1vasur8kyY7EcWA/edit?usp=sharing> |

**Launch an EC2 Instance on AWS**

1. Launch a t2.micro EC2 instance on AWS.

2. Open ports 8000 for JupyterHub and 22 for SSH.

**Add Users**

Do the following as root.

$ cd

$ vi add\_users.sh

useradd eager  
echo eager:eagercyberinsight | chpasswd  
mkdir /home/eager/notebooks  
chown eager:eager /home/eager/notebooks

useradd testuser  
echo testuser:testuser | chpasswd  
mkdir /home/testuser/notebooks  
chown testuser:testuser /home/testuser/notebooks

mkdir -p /home/testuser/.jupyter/custom

chown -R testuser:testuser /home/testuser/.jupyter/

useradd brogers  
echo brogers:cyberinsight | chpasswd  
mkdir /home/brogers/notebooks  
chown brogers:brogers /home/brogers/notebooks  
mkdir -p /home/brogers/.jupyter/custom

chown -R brogers:brogers /home/brogers/.jupyter/

useradd danareed  
echo danareed:cyberinsight | chpasswd  
mkdir /home/danareed/notebooks  
chown danareed:danareed /home/danareed/notebooks  
mkdir -p /home/danareed/.jupyter/custom

chown -R danareed:danareed /home/danareed/.jupyter/

useradd fleagle  
echo fleagle:cyberinsight | chpasswd  
mkdir /home/fleagle/notebooks  
chown fleagle:fleagle /home/fleagle/notebooks  
mkdir -p /home/fleagle/.jupyter/custom

chown -R fleagle:fleagle /home/fleagle/.jupyter/

useradd kangplee  
echo kangplee:cyberinsight | chpasswd  
mkdir /home/kangplee/notebooks  
chown kangplee:kangplee /home/kangplee/notebooks

mkdir -p /home/kangplee/.jupyter/custom

chown -R kangplee:kangplee /home/kangplee/.jupyter/

$ chmod 744 add\_users.sh

$ ./add\_users

**Install Anaconda**

Do the following as root.

1. Go to <https://www.anaconda.com/download/#linux> and download the Python 3.7 version Linux installer.

2. Install Anaconda under /opt/anaconda3. Make sure to answer "yes" to every question.

**Set Up and Run JupyterHub**

Do the following as root.

$ cd

$ source .bashrc

$ conda install -c conda-forge jupyterhub

$ pip install -U jupyterhub ipywidgets matplotlib notebook numpy pandas qgrid sklearn

$ jupyter nbextension enable --py --sys-prefix qgrid

$ jupyterhub --generate-config  
$ vi jupyterhub\_config.py  
c.Authenticator.whitelist = {'eager', ‘testuser’, 'brogers', 'danareed', 'fleagle', 'kangplee'}

c.Authenticator.admin\_users = {'eager'}

c.Spawner.notebook\_dir = '~/notebooks'

c.Spawner.args = ['--NotebookApp.iopub\_data\_rate\_limit=1000000000']

c.PAMAuthenticator.open\_sessions = False

To run Jupyter Hub,  
$ jupyterhub

You can access the JupyterHub at [http://IP\_ADDRESS:8000](http://ip_address:8000).

**Add A New User**

Do the following as root.

$ cd

$ vi add\_new\_user.sh

useradd $1  
echo $1:$2 | chpasswd  
mkdir /home/$1/notebooks  
chown $1:$1 /home/$1/notebooks  
mkdir -p /home/$1/.jupyter/custom  
chown -R $1:$1 /home/$1/.jupyter/

$ chmod 744 add\_new\_user.sh

$ ./add\_new\_user.sh USERID PASSWD

Update copy\_files.sh and remove\_files.sh accordingly.

Add the user to the whitelist in jupyterhub\_config.py.

Rerun JupyterHub

**Install and Run MySQL**

Do the following as root.

$ yum install mysql-server -y  
$ service mysqld start  
$ pip install mysql-connector

$ mysql -u root  
mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'eager'@'localhost' IDENTIFIED BY 'cyberinsightmysql';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'eager'@'%' IDENTIFIED BY 'cyberinsightmysql';  
Query OK, 0 rows affected (0.00 sec)

|  |
| --- |
| CREATE DATABASE tco;  CREATE TABLE IF NOT EXISTS tco.system\_info (  id INT(11) NOT NULL AUTO\_INCREMENT,  user\_id VARCHAR(45) NOT NULL,  ts VARCHAR(26) NOT NULL,    org\_name VARCHAR(45) NOT NULL,  sys\_name VARCHAR(45) NOT NULL,  sys\_desc VARCHAR(1000) NOT NULL,  sys\_type VARCHAR(10) NOT NULL,  deploy\_date DATE NOT NULL,  is\_public INT(1) NOT NULL,    PRIMARY KEY (id),  KEY (user\_id),  KEY (ts) ) ENGINE=InnoDB;  CREATE TABLE IF NOT EXISTS tco.benchmark\_info (  id INT(11) NOT NULL AUTO\_INCREMENT,  user\_id VARCHAR(45) NOT NULL,  ts VARCHAR(26) NOT NULL,    measure\_name VARCHAR(45) NOT NULL,  x\_name VARCHAR(45) NOT NULL,  y\_name VARCHAR(45) NOT NULL,  benchmark\_date DATE NOT NULL,    PRIMARY KEY (id),  KEY (user\_id),  KEY (ts) ) ENGINE=InnoDB;   CREATE TABLE IF NOT EXISTS tco.benchmark\_data (  id INT(11) NOT NULL AUTO\_INCREMENT,  user\_id VARCHAR(45) NOT NULL,  ts VARCHAR(26) NOT NULL,    x FLOAT NOT NULL,  y FLOAT NOT NULL,    PRIMARY KEY (id),  KEY (user\_id),  KEY (ts) ) ENGINE=InnoDB;  CREATE TABLE IF NOT EXISTS tco.tco\_data\_onpremise (  id INT(11) NOT NULL AUTO\_INCREMENT,  user\_id VARCHAR(45) NOT NULL,  ts VARCHAR(26) NOT NULL,   node\_count INT NOT NULL,  cost\_per\_node FLOAT NOT NULL,  rack\_infra\_cost FLOAT NOT NULL,  power\_infra\_cost FLOAT NOT NULL,  monthly\_data\_center\_cost FLOAT NOT NULL,  monthly\_power FLOAT NOT NULL,  power\_cost FLOAT NOT NULL,    personnel\_setup\_cost FLOAT NOT NULL,  monthly\_personnel\_cost FLOAT NOT NULL,    PRIMARY KEY (id),  KEY (user\_id),  KEY (ts) ) ENGINE=InnoDB;  CREATE TABLE IF NOT EXISTS tco.tco\_data\_cloud (  id INT(11) NOT NULL AUTO\_INCREMENT,  user\_id VARCHAR(45) NOT NULL,  ts VARCHAR(26) NOT NULL,    instance\_count INT NOT NULL,  hourly\_instance\_cost FLOAT NOT NULL,  storage\_requirement FLOAT NOT NULL,  monthly\_storage\_cost FLOAT NOT NULL,  monthly\_network\_transfer\_cost FLOAT NOT NULL,    personnel\_setup\_cost FLOAT NOT NULL,  monthly\_personnel\_cost FLOAT NOT NULL,    PRIMARY KEY (id),  KEY (user\_id),  KEY (ts) ) ENGINE=InnoDB;  CREATE TABLE IF NOT EXISTS tco.tco\_questions (  id INT(11) NOT NULL AUTO\_INCREMENT,  sys\_type VARCHAR(10) NOT NULL,  question VARCHAR(100) NOT NULL,  variable\_name VARCHAR(45) NOT NULL,  unit VARCHAR(10) NOT NULL,  PRIMARY KEY (id) ) ENGINE=InnoDB;  INSERT INTO tco.tco\_questions(sys\_type, question, variable\_name, unit) VALUES ("On-Premise", "Node Count", "node\_count", ""), ("On-Premise", "Cost per Node", "cost\_per\_node", "$"), ("On-Premise", "Rack Infrastructure Cost", "rack\_infra\_cost", "$"), ("On-Premise", "Power Infrastructure Cost", "power\_infra\_cost", "$"), ("On-Premise", "MONTHLY Data Center Cost", "monthly\_data\_center\_cost", "$"), ("On-Premise", "Expected MONTHLY Power", "monthly\_power", "kW/h"), ("On-Premise", "Power Cost per kW/h", "power\_cost", "$"), ("On-Premise", "Personnel Setup Cost", "personnel\_setup\_cost", "$"), ("On-Premise", "MONTHLY Personnel Cost", "monthly\_personnel\_cost", "$"),  ("Cloud", "Instance Count", "instance\_count", ""), ("Cloud", "HOURLY Instance Cost", "hourly\_instance\_cost", "$"), ("Cloud", "Storage Requirement", "storage\_requirement", "GB"), ("Cloud", "MONTHLY Storage Cost per GB", "monthly\_storage\_cost", "$"), ("Cloud", "Expected MONTHLY Network Transfer Cost", "monthly\_network\_transfer\_cost", "$"), ("Cloud", "Personnel Setup Cost", "personnel\_setup\_cost", "$"), ("Cloud", "MONTHLY Personnel Cost", "monthly\_personnel\_cost", "$"); |

**Create custom.js**

Do the following as root.

$ mkdir -p /home/eager/.jupyter/custom

$ vi /home/eager/.jupyter/custom/custom.js

IPython.OutputArea.prototype.\_should\_scroll = function(lines) {return false;}

This custom.js file will need to be copied to all users.

**Copy Files from eager to All Users**

Do the following as root.

$ cd

$ vi copy\_files.sh

cp /home/eager/notebooks/\* /home/brogers/notebooks/  
chown brogers:brogers /home/brogers/notebooks/\*

cp /home/eager/.jupyter/custom/custom.js /home/brogers/.jupyter/custom/

chown brogers:brogers /home/brogers/.jupyter/custom/custom.js

cp /home/eager/notebooks/\* /home/testuser/notebooks/  
chown testuser:testuser /home/testuser/notebooks/\*

cp /home/eager/.jupyter/custom/custom.js /home/testuser/.jupyter/custom/

chown testuser:testuser /home/testuser/.jupyter/custom/custom.js

cp /home/eager/notebooks/\* /home/danareed/notebooks/  
chown danareed:danareed /home/danareed/notebooks/\*

cp /home/eager/.jupyter/custom/custom.js /home/danareed/.jupyter/custom/

chown danareed:danareed /home/danareed/.jupyter/custom/custom.js

cp /home/eager/notebooks/\* /home/fleagle/notebooks/  
chown fleagle:fleagle /home/fleagle/notebooks/\*

cp /home/eager/.jupyter/custom/custom.js /home/fleagle/.jupyter/custom/

chown fleagle:fleagle /home/fleagle/.jupyter/custom/custom.js

cp /home/eager/notebooks/\* /home/kangplee/notebooks/  
chown kangplee:kangplee /home/kangplee/notebooks/\*

cp /home/eager/.jupyter/custom/custom.js /home/kangplee/.jupyter/custom/

chown kangplee:kangplee /home/kangplee/.jupyter/custom/custom.js

$ chmod 744 copy\_files.sh

$ ./copy\_files.sh

**Remove Files from All Users except for eager**

Do the following as root.

$ cd

$ vi remove\_files.sh

rm -rf /home/testuser/notebooks/\*

rm -rf /home/brogers/notebooks/\*

rm -rf /home/danareed/notebooks/\*

rm -rf /home/fleagle/notebooks/\*

rm -rf /home/kangplee/notebooks/\*

$ chmod 744 remove\_files.sh

$ ./remove\_files.sh

**Distribute Modules**

Do the following as root. Make sure to have the python script in /home/eager/notebooks/Backup.

$ cd

$ mkdir functions

$ cd functions

$ cp /home/eager/notebooks/Backup/my\_DB\_functions.py .  
$ vi setup.py  
from distutils.core import setup  
  
setup(  
 name = 'dbaccess',  
 version = '1.0.0',  
 py\_modules = ['my\_DB\_functions'],  
 author = 'kangpyolee',  
 author\_email = 'kangpyo-lee@uiowa.edu',  
 url = '',  
 description = '',  
 )  
$ python setup.py sdist  
$ python setup.py install

**Edit login.html**

<https://pythonforundergradengineers.com/assignments-dir-and-custom-login-page-to-jupyterhub.html>

Do the following as root.

$ cp -R /opt/anaconda3/pkgs/jupyterhub-0.9.4-py37\_1000/share/jupyterhub/templates/ /home/eager/templates/

$ vi /root/jupyterhub\_config.py

c.JupyterHub.template\_paths = ['/home/eager/templates/']

$ chown -R eager:eager /home/eager/templates/

$ vi /home/eager/templates/login.html

{% extends "page.html" %}  
{% if announcement\_login %}  
 {% set announcement = announcement\_login %}  
{% endif %}  
  
{% block login\_widget %}  
{% endblock %}  
  
{% block main %}  
  
{% block login %}  
<div id="login-main" class="container">  
{% if custom\_html %}  
{{ custom\_html | safe }}  
{% elif login\_service %}  
<div class="service-login">  
 <a role="button" class='btn btn-jupyter btn-lg' href='{{authenticator\_login\_url}}'>  
 Sign in with {{login\_service}}  
 </a>  
</div>  
{% else %}  
<form action="{{login\_url}}?next={{next}}" method="post" role="form">  
  
 <img src="https://nsf.gov/awardsearch/images/common/head.gif" alt="NSF"></img>  
 <p><br></p>  
 <p align="center"><a href="https://tco.sites.uiowa.edu/" target="\_new">CYBER-INSIGHT: Evaluating Cyberinfrastructure Total Cost of Ownership</a></p>  
 <p align="center">If you would like to request an account, click <a href="https://goo.gl/forms/mRhRE0JbVHNeFkzj1" target="\_new">here</a>.</p>  
 <p><br></p>  
  
 <div class="auth-form-header">  
 Sign in  
 </div>  
 <div class='auth-form-body'>  
  
 <p id='insecure-login-warning' class='hidden'>  
 Warning: JupyterHub seems to be served over an unsecured HTTP connection.  
 We strongly recommend enabling HTTPS for JupyterHub.  
 </p>  
  
 {% if login\_error %}  
 <p class="login\_error">  
 {{login\_error}}  
 </p>  
 {% endif %}  
 <label for="username\_input">Username:</label>  
 <input  
 id="username\_input"  
 type="text"  
 autocapitalize="off"  
 autocorrect="off"  
 class="form-control"  
 name="username"  
 val="{{username}}"  
 tabindex="1"  
 autofocus="autofocus"  
 />  
 <label for='password\_input'>Password:</label>  
 <input  
 type="password"  
 class="form-control"  
 name="password"  
 id="password\_input"  
 tabindex="2"  
 />  
  
 <input  
 type="submit"  
 id="login\_submit"  
 class='btn btn-jupyter'  
 value='Sign In'  
 tabindex="3"  
 />  
 </div>  
</form>  
{% endif %}  
</div>  
{% endblock login %}  
  
{% endblock %}  
  
{% block script %}  
{{ super() }}  
<script>  
if (window.location.protocol === "http:") {  
 // unhide http warning  
 var warning = document.getElementById('insecure-login-warning');  
 warning.className = warning.className.replace(/\bhidden\b/, '');  
}  
</script>  
  
{% endblock %}

**Enable SSL/HTTPS**

Get the two certificate files first with the DNS name cyberinsight.hpc.uiowa.edu, then do the following as root.

$ cd

$ mkdir certificate

Place the two cert files in the certificate directory.

$ vi jupyterhub\_config.py

c.JupyterHub.ssl\_cert = '/root/certificate/cyberinsight\_chain.cer'

c.JupyterHub.ssl\_key = '/root/certificate/cyberinsight.key'

Rerun JupyterHub

Access to <https://cyberinsight.hpc.uiowa.edu:8000>.