CyVerse Documentation

CyVerse

Sep 18, 2020

Contents

|CyVerse_logo|_

 Learning Center Home

2018-07-12 PREZI

CHAPTER 1

Moving Geospatial Data Analysis onto hybrid Cloud & HPC

The purpose of this tutorial is to set up a linux-based Virtual Machine running on CyVerse Atmosphere or XSEDE Jetstream to do interactive geospatial analyses in Jupyter Notebooks and RStudio-Server.

The VM(s) can also be run in any other number of ways that the user sees fit.

CHAPTER 2

Contents

|CyVerse_logo|_

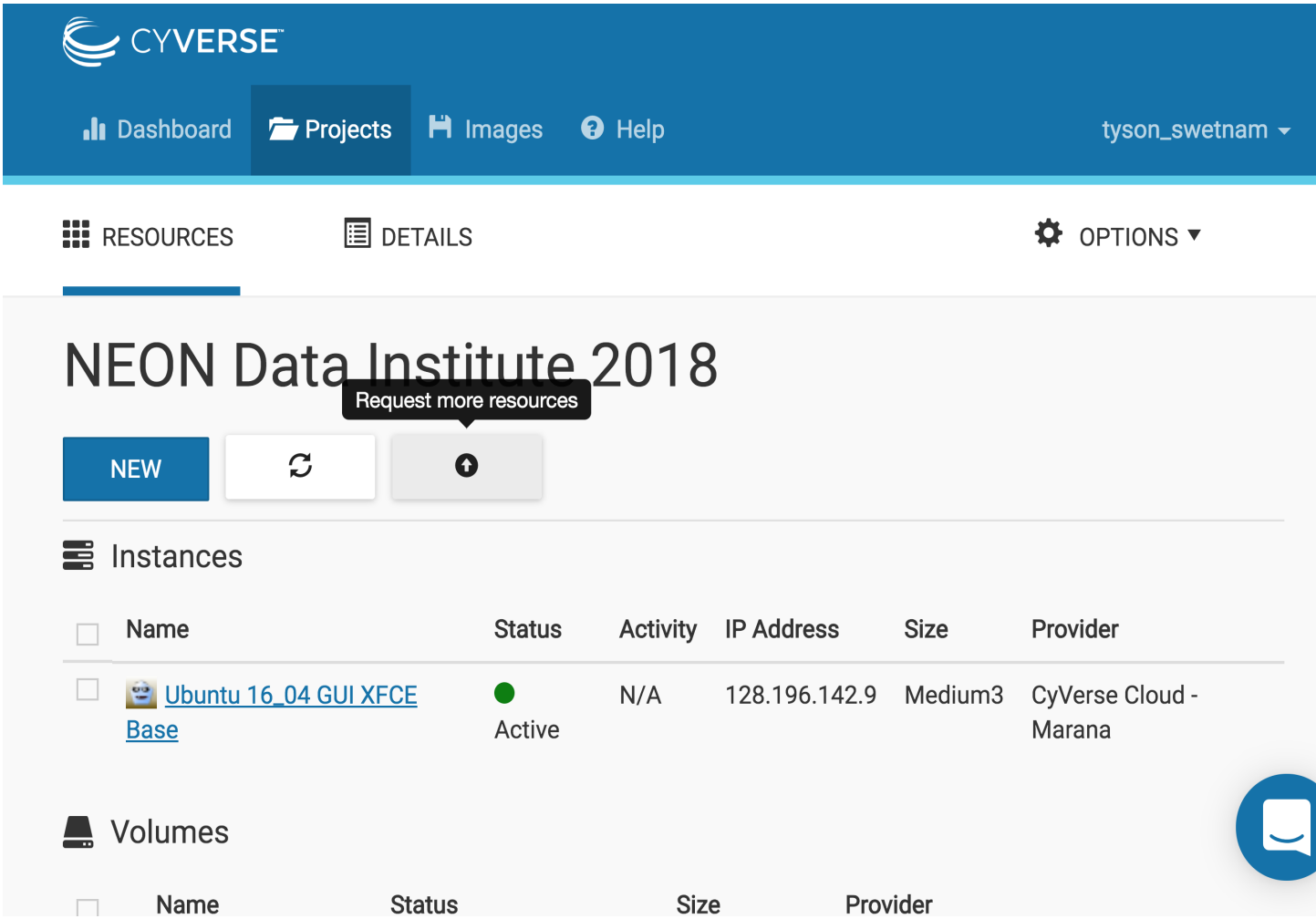
 Learning Center Home

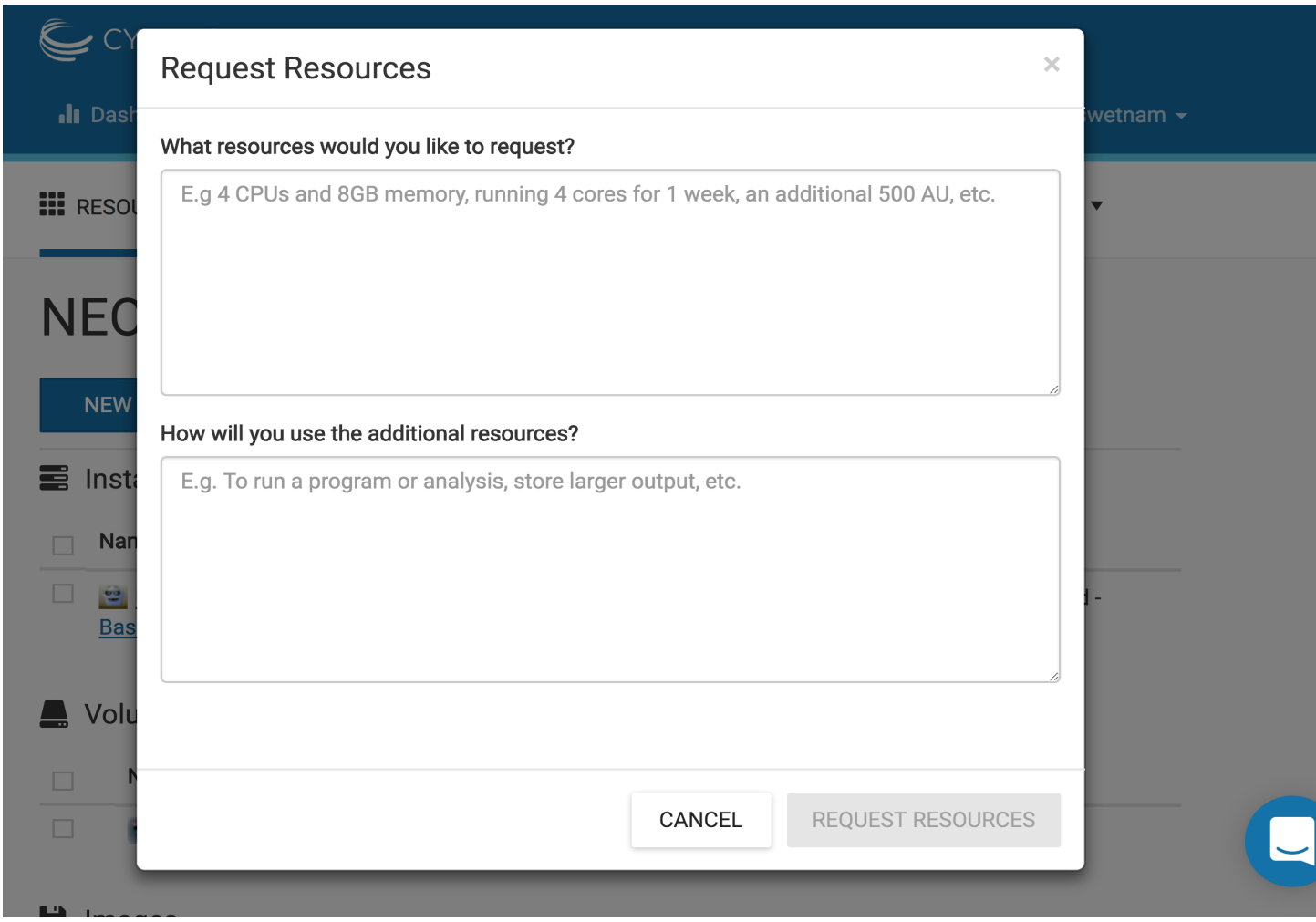
2.1 Data Science Virtual Machines on Atmosphere or Jetstream

Description: Provision VM for analyzing NEON AOP data

CyVerse operates a cloud service called Atmosphere. Users can request up to 2,000 allocation units [units are hours (hr)] per month. E.g. a 1-core instance uses 1 AU/hr, a 4-core instance uses 4 AU/hr, and a 16-core instance uses 16 AU/hr. Allocations are automatically reset to 128 AU on the 1st of each month.

Users can request more AU by clicking the Request More Resources button in the Atmosphere UI. You can also get help by asking questions in the Intercom (blue button in the lower right of the CyVerse website pages).


CYVERSE
Dashboard
Projects
Images
Help
tyson_swetnam
RESOURCES
DETAILS
OPTIONS
NEON Data Institute 2018
Request more resources
NEW
Instances
Name
Status
Activity
IP Address
Size
Provider
Ubuntu 16_04 GUI XFCE Base
Active
N/A
128.196.142.9
Medium3
CyVerse Cloud - Marana
Volumes
Name
Status
Size
Provider



Requests are typically approved in <1 hour during business hours, and <24 hours on nights and weekends.

XSEDE Jetstream uses the same UI interface as Atmosphere. Startup allocations typically range from 25,000 - 250,000 AU per year. Research allocations between 250,000 to several million AU are also available through XSEDE.

2.1.1 *Login*

Log into CyVerse Atmosphere

Atmosphere Manual

Alternately, log into XSEDE Jetstream

Fill in your `username` and `password` and click "LOGIN"

2.1.2 *Create a Project*

This is something you only need to do once.

- Click on the "Projects" tab on the top and then click "CREATE NEW PROJECT"
- Enter a name, e.g. "NEON2018" into the Project Name field.

- the Description can be something complex and long (like an extended abstract, or tutorial), or something short like "Data Institute 2018".
- Select the newly created project

2.1.3 *Start a new Instance*

From your Project folder, you can select "New" and "Instance"

1. Suggest you select a featured image with a Graphic User Interface (GUI).

Suggested Atmosphere Image(s):

Atmosphere Image(s):

Here are the tested Ubuntu images.

Warning: The latest version of Ubuntu (18.04) may not have current packages for some software.

Image Name	Version	Description	Link
Ubuntu 16.04 GUI	2.1	Ubuntu 16.04 GUI XFCE Base	Image
Ubuntu 16.04 non-GUI	1.6	Ubuntu 16.04 non-GUI Base	Image
Ubuntu 18.04 GUI	1.0	Ubuntu 18.04 GUI XFCE Base	Image
Ubuntu 18.04 non-GUI	1.0	Ubuntu 18.04 non-GUI Base	Image

Suggested Jetstream Image(s):

Image Name	Version	Description	Link
Ubuntu 16.04 GUI	1.13	Ubuntu 16.04 LTS Development + GUI support + Docker	Image
Ubuntu 14.04 GUI	1.17	Base Ubuntu 14.04.3 + Xfce + Xfce-goodies, firefox, icon sets and themes	Image

- Find the "Ubuntu 16.04" image, click on it
- Give it a short name that is distinct "my_first_vm"
- Select 'tiny1 (CPU: 1, Mem: 4GB, Disk: 30GB)'. Because this is your first attempt at provisioning a virtual machine it doesn't need to be a workhorse (yet).
- Leave rest of the fields as default.
- Wait for it to become active
- Be Patient (but not too patient - if it takes >10 minutes the system may be at capacity, if you're trying to launch a large or extra large VM, try something smaller).
- You can click on your new instance to get more information.

2.1.4 *Accessing the Shell*

Once the instance is *active*, you can access it via `ssh` or by using the Web Shell provided by Atmosphere.

- Click "Open Web Shell", *or*, if you know how to use ssh,

you can ssh in with your CyVerse username on the IP address of the machine

```
ssh CyVerseUserName@<INSTANCE-IP-ADDRESS>
```

You should see something like this

```
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-81-generic x86_64)

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

155 packages can be updated.
0 updates are security updates.


*** System restart required ***
Welcome to
    _   _                                 _
   / \ | |_ _ __ ___   ___  ___ _ __ | |__   ___ _ __ ___
  / _ \| __| '_ ` _ \ / _ \/ __| '_ \| '_ \ / _ \ '__/ _ \
 / ___ \ |_| | | | | | (_) \__ \ |_) | | | |  __/ | |  __/
/_/   \_\__|_| |_| |_|\___/|___/ .__/|_| |_|\___|_|  \___|

cyverse_username@vm142-39:~$
```

Note, this instance is running an older version of Ubuntu 16.

A good practice before installing any new software is to run:

```
sudo apt-get update && sudo apt-get upgrade
```

After the new updates are installed you can reboot the machine from terminal or from the Atmosphere UI

```
sudo reboot
```

If you're using the Web Shell, the instance will exit. Wait a few minutes for the instance to reboot and refresh the screen.

Note: To access the Clipboard in an Apache Guacamole Web Shell:

- Open Clipboard and virtual keyboard - On a standard keyboard: *ctrl* + *alt* + *shift* key - On a MAC OS X keyboard: *control* + *command* + *shift* key
- Select your text or paste text into the clipboard window.
- Close the Clipboard window by selecting *control* + *command* + *shift* keys again
- Right click with your mouse or double tap fingers on touchpad to paste in the web shell or Desktop

---

Suspending an instance

- When you're done using an instance it is wise to 'Suspend' the instance in the Actions.
- This will kill any process that is still running.
- Your data and all of your programs will be fine. It is however wise to move your data back onto your DataStore or back it up somewhere else so it will be available.
- Suspending the instance will leave it ready for reuse when you want to "resume" working on it.
- You will not be charged any AU while the instance is suspended.

Deleting your instance

- To completely remove your instance, you can select the "delete" buttom from the instance details page.
- This will open up a dialogue window. Select the "Yes, delete this instance" button.
- It may take Atmosphere a few minutes to process your request. The instance should disappear from the project when it has been successfully deleted.

Imaging an instance

The use of Docker and Singularity take a lot of the problems out of building unique software stacks on cloud - but sometimes these cannot be avoided.

- Have you created a unique software stack that you need to launch on a larger number of future instances?
- Does it take a long time to compile your software stack each time you launch a new instance?
- Only create images from the smallest possible versions of your instance. A larger imaged instance cannot be run on a smaller instance.

To request that your instance be imaged click the "Image" button from Actions.

Note: It is advisable to delete the machine if you are not planning to use it in future to save valuable resources. However if you want to use it in future, you can suspend it.

If you want to keep the instance for a future project, you can also "shelve" the instance. It will take a longer period of time to resume a shelved instance.

2.1.5 *EZ Installation of Project Jupyter*

We install Project Jupyter (Notebooks and Lab) using the Anaconda distribution. Within the Anaconda distribution is the `conda` package manager which can be used to both build and install software.

Anaconda is different than a basic Python installation. It serves as both a package manager and an environment. While this has many benefits, it also adds some complexity to running your Python environments. Still confused? Read about the myths and misconceptions of Anaconda.

For more details about installing software on Atmosphere visit the CyVerse Data Science Quickstart Tutorial or the Jetstream EZ Tutorial. There are instructions for `ez` installation of Docker, Singularity, and Anaconda.

If you're on an instance which already has Anaconda installed, you'll still need to re-run `ez` to restart the Anaconda virtual environment.

1. Install Anaconda with Python3 (`ez` comes preloaded on featured instances on Atmosphere and Jetstream) by typing:

```
ezj
```

2. Once the installation completes, you're done! A Jupyter Notebook should now be running on the VM.

```
tswetnam@js-19-4:~$ ezj
/usr/bin/python3
DEBUG: using python version 3
DEBUG: downloading anaconda binary, may take a few minutes
DEBUG: Anaconda already installed to /opt/anaconda3
/opt/anaconda3/bin/python3
DEBUG: using python version 3
[I 07:44:31.522 NotebookApp] JupyterLab beta preview extension loaded from /opt/anaconda3/lib/python3.6/site-packages/jupyter
[I 07:44:31.523 NotebookApp] JupyterLab application directory is /opt/anaconda3/share/jupyter/lab
[I 07:44:31.549 NotebookApp] my module enabled!
[I 07:44:31.552 NotebookApp] Serving notebooks from local directory: /home/tswetnam
[I 07:44:31.552 NotebookApp] 0 active kernels
[I 07:44:31.552 NotebookApp] The Jupyter Notebook is running at:
[I 07:44:31.552 NotebookApp] http://js-19-4.jetstream-cloud.org:8888/?token=70d484e87467a7981c20c2344cf6fe5a3124e739057e2d9e
[I 07:44:31.552 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 07:44:31.552 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://js-19-4.jetstream-cloud.org:8888/?token=70d484e87467a7981c20c2344cf6fe5a3124e739057e2d9e&token=70d484e87467a79
c2344cf6fe5a3124e739057e2d9e
```

3. Click the link showing the notebook URL (notice this is not the localhost:8888).

Note: To install your own packages you'll need to change ownership of the Anaconda installation:

```
sudo chown $(id -u):$(id -g) /opt/anaconda3 -R
```

Down version Python 3.6 to 3.5

To use GDAL you may need to reverse version Python to an earlier version

Kernel installation instructions

```
python -m pip install ipykernel
```

```
conda create -n ipykernel_py35 python=3.5 ipykernel
source activate ipykernel_py35    # On Windows, remove the word 'source'
python -m ipykernel install --user
```

List of Jupyter Kernels

R

```
conda install -c r irkernel
```

JavaScript

```
sudo apt-get install nodejs-legacy npm ipython ipython-notebook
sudo npm install -g ijavascript
ijsinstall
```

Ruby

Add Jupyter PPA

```
sudo add-apt-repository ppa:chronitis/jupyter -y
sudo apt-get update
sudo apt-get install -y iruby
```

Python2 Kernel

```
conda create -n ipykernel_py2 python=2 ipykernel
source activate ipykernel_py2
python -m ipykernel install --user
source deactivate ipykernel_py2
conda activate base # switch back to base Python3 environment
```

Julia Kernel

First, install Julia, here we are installing v0.6.

Once Julia as been installed, run `julia` from the prompt.

```
wget https://julialang-s3.julialang.org/bin/linux/x64/0.6/julia-0.6.3-linux-
↪x86_64.tar.gz
tar xvzf julia-0.6.3-linux-x86_64.tar.gz
sudo mv julia-d55cadc350/ /opt/julia
rm -rf julia-0.6.3-linux-x86_64.tar.gz
sudo ln -s /opt/julia/bin/julia /usr/local/bin/julia
julia
```

Now, from Julia prompt install the iJulia Kernel.

```
Pkg.add("IJulia")
ENV["JUPYTER"] = "/opt/anaconda3/bin/jupyter"
Pkg.add("Feather")
Pkg.add("DataFrames")
Pkg.add("NamedArrays")
```

Bash Kernel

```
pip install bash_kernel
python -m bash_kernel.install
```

Geospatial dependencies

```
conda install -c conda-forge gdal
```

```
sudo add-apt-repository -y ppa:ubuntugis/ubuntugis-unstable
sudo apt update
sudo apt install gdal-bin python-gdal python3-gdal libgdal1-dev
```

Script of Scripts

Official documentation

```
pip install sos
pip install sos-notebook
python -m sos_notebook.install
```

2.1.6 *Installing RStudio-Server*

RStudio can be installed in several ways.

First, you can follow the RStudio-Server instructions for Linux

Second, you can use Docker (following the same `ez` documentation as for Anaconda). We suggest using containers from Docker Hub Rocker on the instance.

```
ezd
sudo usermod -aG docker $USER
exit
docker pull rocker/geospatial
docker run -d -p 8787:8787 rocker/geospatial
```

Third, you can use Anaconda

Here we are going to use `ezj` to install both Anaconda (Jupyter) and R

```
ezj -R
```

This will trigger the Ansible playbook to install `r-base`, `r-essentials`, and a few other commonly used R Data Science packages.

After `ezj -R` has finished, you can install RStudio-Server

Install these misc. dependencies

```
export PATH="/opt/anaconda3/bin":$PATH
sudo chown $(id -u):$(id -g) /opt/anaconda3/ -R
conda update conda
conda install gxx_linux-64
conda install gcc_linux-64
```

Set Path and install `gdebi`

```
sudo apt-get install gdebi-core
```

Install RStudio-Server with `gdebi`:

```
echo "export RSTUDIO_WHICH_R='/opt/anaconda3/bin/R'" >> ~/.bash_profile
wget https://download2.rstudio.org/rstudio-server-1.1.447-amd64.deb
sudo gdebi --non-interactive rstudio-server-1.1.447-amd64.deb
```

The installation of RStudio-Server is going to fail because we haven't told it which R to use. Because we are using Anaconda's installation of R, and not the basic installation of R, we have to reassign RStudio to look for Anaconda

```
sudo sh -c 'echo "rsession-which-r=/opt/anaconda3/bin/R" >> /etc/rstudio/
↪rserver.conf'
# export RSTUDIO_WHICH_R='/opt/anaconda/lib/R/bin/R'
# sudo sh -c 'echo "launchctl setenv RSTUDIO_WHICH_R $RSTUDIO_WHICH_R" >> ~/.
↪bash_profile'
```

Restart the server

```
sudo rstudio-server start
```

4. You can launch Jupyter Lab by exiting the notebook and typing *jupyter lab* - but this will allow Lab to only be available on the localhost, with no way to connect from a remote terminal. Exit the notebook by pressing *ctrl* + *c* twice, and then start a Jupyter Lab.

Note: To ensure your session doesn't die when you close your terminal use *tmux* or *screen* to start your remote sessions and to detach the screen before exiting.

- detach screen: *ctrl* + *b* then *d*
- list tmux sessions: `tmux ls`
- re-attach screen: `tmux attach -t <session id #>`

2.1.7 *Establishing a Secure Connection*

1. On the VM start the Lab in terminal (don't forget to use *tmux*)

```
jupyter lab --no-browser --ip=* --port=8888
```

Option 1: SSH tunnel

You must have the ability to use `ssh` on your localhost to use this method.

1. Start Jupyter

```
jupyter lab --no-browser --ip=127.0.0.1 --port=8888
```

2. Open a new terminal on your localhost.

```
ssh -nNT -L 8888:localhost:8888 CyVerseUserName@<IPADDRESS>
```

Enter your password when prompted.

The terminal should stop responding after this.

3. In your browser, open a new tab and go to `http://localhost:8888`

Option 2: Caddy

You can use this method with `tmux` in the Web Shell

1. Follow the same step #1 above
2. In the terminal start a new `tmux` session. Then copy/paste the following:

```
echo "$(hostname)
proxy / 127.0.0.1:8888 {
    websocket
    transparent
}
" > Caddyfile
curl https://getcaddy.com | bash -s personal http.nobots
caddy
```

The Caddyserver will output a secure URL *https://* for the Atmosphere VM which you can then connect in a new browser tab.

3. Copy / Paste the URL `https://vm142-xx.cyverse.org` into a new browser tab.

Description of output and results

Congratulations - you've got a Virtual Machine ready to do some serious data science!

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home

|CyVerse_logo|_

 Learning Center Home

2.2 Data Download

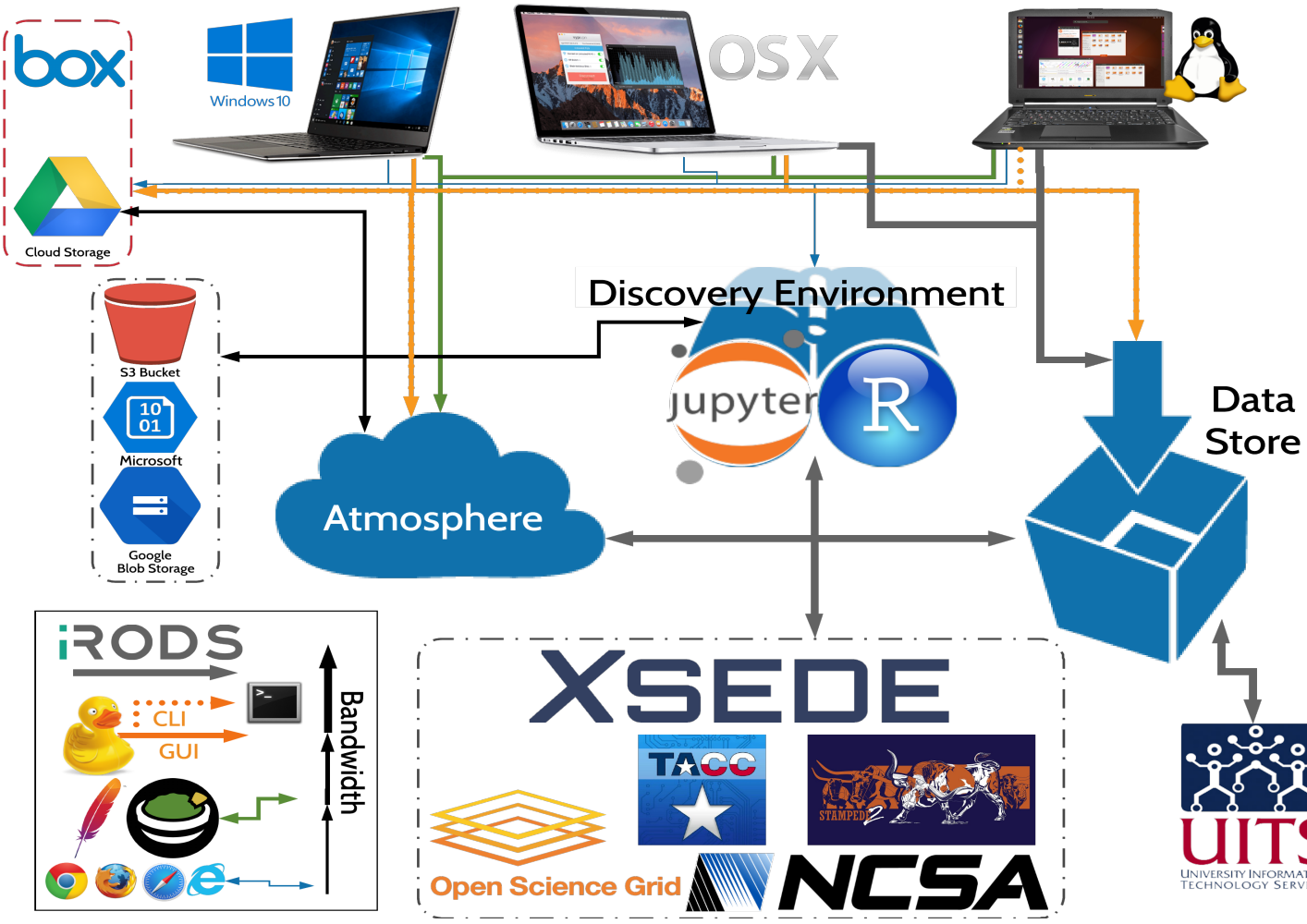
Description: Managing your data on CyVerse

Official CyVerse Data Management

Using CyVerse iCommands

Official iCommands User

Data Sharing Services:



Input	Description	Link
Google Drive	Cloud Data Storage Service	Jupyter Client
Cyverse Data Store	integrated Rule-Oriented Data System (iRODS)	Jupyter Client
CyberDuck	File Share UI	Using with CyVerse
Guacamole	Atmosphere Browser Tabs	

2.2.1 *Google Drive Jupyter Client*

Note: As of July 2018, the Google Drive Jupyter Client has been deprecated.

Update Conda to latest

```
conda update conda
conda update conda-build
```

Dependency: Node.js 5+

```
conda install -c conda-forge nodejs
```

1. Install Google Drive to Jupyter Lab

Google Drive requires port `8888` or `8889` with port forwarding to work

```
jupyter labextension install @jupyterlab/google-drive
```

2.2.2 *iRODS Jupyter Client*

CyVerse has a developed a Jupyter Lab iRODS client similar to the Google Drive extension.

Update to the latest version of Jupyter Lab

```
conda install -c conda-forge jupyterlab
```

```
pip install jupyterlab_irods
jupyter serverextension enable --py jupyterlab_irods
jupyter labextension install @towicode/jupyterlab_irods
```

2.2.3 *iRODS iCommands CLI*

CyVerse Instructions

Instructions from iRODS

Download from iRODS

1. Install iCommands on the VM

```
wget -qO - https://packages.irods.org/irods-signing-key.asc | sudo apt-
↪key add -
echo "deb [arch=amd64] https://packages.irods.org/apt/ $(lsb_release -sc)␣
↪main" | sudo tee /etc/apt/sources.list.d/renci-irods.list
sudo apt-get update
sudo apt-get install irods-icommands
```

2. Initialize iRODS-iCommands

```
iinit
```

You will be queried to set up your *irods_environment.json*

Enter the following:

statement	input
DNS	*data.cyverse.org*
port	*1247*
username	*your CyVerse username*
zone	*iplant*

Note: Set up auto-complete for iCommands instructions

Download i-commands-auto.bash

In your home directory, rename `i-commands-auto.bash` to `.i-commands-auto.bash`

In your `.bashrc` or `.bash_profile`, enter the following:

```
source .i-commands-auto.bash
```

2.2.4 *CyberDuck*

CyberDuck is a GUI client that works in Windows and Mac OS X. It can connect to almost every type of proximate file sharing system (DropBox, Drive, AWS S3, SFTP, iRODS, etc).

Using CyVerse and CyberDuck

CyberDuck also has a CLI client that can be used in Linux.

To install on the VM:

```
echo -e "deb https://s3.amazonaws.com/repo.deb.cyberduck.io stable main" |␣
↪sudo tee /etc/apt/sources.list.d/cyberduck.list > /dev/null
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys␣
↪FE7097963FEFBE72
sudo apt-get update
sudo apt-get install duck
```

2.2.5 *Guacamole*

Open the Web Desktop or Web Shell from the Atmosphere UI

`Ctrl + Alt + Shift` to open the side bar.

Click on `Devices`

Select the `Home` folder (or a folder that your $USER has ownership of) and upload data.

Description of output and results

Congratulations! You've established ways to get data to and from your VM!

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home

|CyVerse_logo|_

 Learning Center Home

2.3 NEON Data API w/ Python

NEON developed an R and Python API for downloading data from their data store.

2.3.1 *Cloning Jupyter Tutorials from Github*

We provide some example Python3 Notebooks and R Markdown Notebooks for downloading lidar and hyperspectral data.

Prerequisite: Installed Anaconda and RStudio-Server, launched Jupyter Notebook or Lab

In the terminal:

1. Clone notebooks from NEON Data Science or CyVerse GIS to a location on the VM (e.g. `/home/user/`)

```
git clone https://github.com/cyverse-gis/neon_data_science
cd neon_data_science/lessons
```

2. From Jupyter Notebook or Lab select a data download notebook.
3. Follow the notebook instructions.

2.3.2 *Download data from CyVerse DataStore in Bash*

CyVerse uses a system called iRODS to move files onto and off of its Data Store.

iRODS uses multi-threaded file transfers for faster downloads and uploads than traditional `wget` or `curl`

Prerequisite: Installed iRODS iCommands and initiated connection

1. Use the `ils` command to view your files on the Data Store
2. Change ownership of the directory where you want to download the data.

```
sudo chown $USER:iplant-everyone /scratch -R
```

3. Create a new directory in `/scratch`

```
mkdir -p /scratch/2016_Campaign/HARV/L1/DiscreteLidar/
```

4. Use the iget command to download files from the Data Store

```
iget -KPQbrvf /iplant/home/shared/NEON_data_institute_2018/2016_Campaign/
↪HARV/L1/DiscreteLidar/ClassifiedLaz /scratch/2016_Campaign/HARV/L1/
↪DiscreteLidar/ClassifiedLaz
```

In this example we are using the flags to:

-K	verify the checksum
-P	output the progress of the download.
-Q	use RBUDP (datagram) protocol for the data transfer
-b	bulk file transfer
-r	recursive - retrieve subcollections
-v	verbose
-f	force - write local files even it they exist already (overwrite them)

2.3.3 *Upload data to the CyVerse DataStore in Bash*

1. Use the iput command to upload files to the Data Store

```
iput -KPQbrvf /scratch/2016_Campaign/HARV/L1/DiscreteLidar/some_results /
↪iplant/home/$USER/neon/results
```

Note, we are using the same flags as the `iget` statement above.

2.3.4 *Download data from CyVerse DataStore with CyberDuck*

After you've set up Cyberduck to access your CyVerse DataStore, you can click and drag and drop files to your localhost; or drag and drop files into a second CyberDuck window that is connected to another data source.

Note: Dragging and dropping data with Cyberduck will cause the data to be streamed down to your localhost and then uploaded back to the second remotehost. This will greatly reduce the speed with which you transfer files.

It is strongly suggested you use the Cyberduck CLI tool to move files between two remote data stores.

2.3.5 *Jupyter Lab Google Drive Client*

Google Drive will ask for some authentication through your browser with a token. After you authenticate you can view files in your Google Drive and move them onto the VM.

If you have any data on Google Drive, you can drag and drop them onto your VM.

2.3.6 *Jupyter Lab iRODS Client*

After you've authenticated to CyVerse, you will be able to view your data store files.

The Jupyter iRODS Client is not suitable for downloading hundreds of files, but it is useful for finding files and copying their URLs.

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home

|CyVerse_logo|_

 Learning Center Home

2.4 Working with Docker and Singularity

CyVerse recently taught a Container BootCamp with in depth instructions for working with Docker and Singularity.

In these examples we'll use Singularity to launch both Docker and Singularity containers.

Description: Run CLI and GUI programs on VMs using Docker & Singularity Containers

If you're running on Windows OS you can set up the Windows-Linux subsystem to access a real Linux terminal. This will enable you to run secure shell connections to your VM.

Another option is to use the Atmosphere Web Desktop, which is running an XFCE Desktop.

2.4.1 *Build a Singularity Container yourself*

If you want to develop your own containers you can download the example Singularity file from my Github repository and make your own changes

EZ Install Singularity on Atmosphere or Jetstream

To install Singularity on linux follow these instructions.

As of early May 2018, Singularity is version *2.5.1*

```
VERSION=2.5.1
wget https://github.com/singularityware/singularity/releases/download/
↪$VERSION/singularity-$VERSION.tar.gz
tar xvf singularity-$VERSION.tar.gz
cd singularity-$VERSION
./configure --prefix=/usr/local
make
sudo make install
cd ..
sudo rm -rf singularity-$VERSION.tar.gz
```

Singularity build dependencies:

Get the Singularity file from terminal:

1. Clone github repository onto the VM (e.g. `/home/user/`)

```
git clone https://github.com/tyson-swetnam/osgeo-singularity
cd osgeo-singularity
```

2. Select `Singularity` file and view it if you like, make any changes you wish.
3. Build the container locally:

```
sudo singularity build osgeo.simg Singularity
```

2.4.2 *Download the Container from Singularity-Hub*

The container image is hosted on Singularity Hub and can be downloaded from there.

1. Pull the image from Singularity-Hub

```
singularity pull --name osgeo.simg shub://tyson-swetnam/osgeo-singularity
```

2.4.3 *Running CLI scripts*

To run the container from the CLI:

```
singularity shell osgeo.simg
```

Running a container from your Jupyter Notebook (Python3)

```
```

2.4.4 *Run GUI Applications*

Run the container with the `singularity exec` command to use the GUI applications, the interface for GRASS:

```
singularity exec osgeo.simg grass74
```

GRASS 7.4 has a problem with its environment variables not being set within the container. You can do this by hand while the container is running:

```
singularity shell osgeo.simg

GISBASE=/opt/osgeo/grass-7.4.0
GRASS_PROJSHARE=/usr/share/proj
LD_LIBRARY_PATH=/opt/osgeo/lib:/opt/osgeo/grass-7.4.0/lib
PATH=/opt/osgeo/bin:/opt/osgeo/grass-7.4.0/bin:$PATH
PYTHONPATH=/opt/osgeo/lib/python3.6/site-packages
export GISBASE GRASS_PROJSHARE LD_LIBRARY_PATH PATH PYTHONPATH

grass74
```

For QGIS:

```
singularity exec osgeo.simg qgis
```

For Saga-GIS:

```
singularity exec osgeo.simg saga_gui
```

Note: Running the GUI applications requires a stable, fast, internet connection, else loading large raster layers may be very slow.

You must use the Atmosphere Web Shell or `ssh -X` in the terminal to access the Container's GUI applications.

```
ssh -X <USERNAME>@<IP-ADDRESS>
```

If you are using the Web Desktop, you can resize the screen by opening the terminal emulator and typing `xrandr`

```
 SZ:    Pixels          Physical       Refresh
 0   1024 x 768     ( 260mm x 195mm )   0
 1    800 x 600     ( 203mm x 152mm )   0
 2   1280 x 800     ( 325mm x 203mm )   0
```

(continues on next page)

(continued from previous page)

```
 3    1280 x 960     ( 325mm x 244mm )    0
 4    1280 x 1024    ( 325mm x 260mm )    0
 5    1680 x 1050    ( 427mm x 267mm )    0
 6    1920 x 1080    ( 488mm x 274mm )    0
*7    1920 x 1200    ( 488mm x 305mm )   *0
 8    3360 x 1050    ( 853mm x 267mm )    0
 9    1024 x 700     ( 260mm x 178mm )    0
 10   1200 x 740     ( 305mm x 188mm )    0
 11   1600 x 1000    ( 406mm x 254mm )    0
 12   1600 x 1200    ( 406mm x 305mm )    0
 13   3200 x 1000    ( 813mm x 254mm )    0
 14   3200 x 1200    ( 813mm x 305mm )    0
Current rotation - normal
Current reflection - none
Rotations possible - normal
Reflections possible - none
```

This will show you the list of possible screen resolutions. To reset the screen resolution to HD (1920x1080):

```
xrandr -s 6
```

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

Learning Center Home

|CyVerse_logo|_

Learning Center Home

2.5 Working with Google Earth Engine API

Description: Run Earth Engine from your Jupyter Notebook or Lab

Dr. Guillermo E. Ponce-Campos from the Tucson ARS Southwest Watershed Research Center has developed some NEON tuturials on Google Earth Engine and an Earth Engine App.

Samapriya Roy has provided us with a Planet Labs & Google Earth Engine Tutorial

2.5.1 *Install Earth Engine API*

Official Instructions

Requirements: Docker

Build Dependencies:

```
ezd -p # In this example I'm also installing Portainer.io
sudo usermod -aG docker $USER
```

Set the paths:

```
export GCP_PROJECT_ID=gee-projects
export CONTAINER_IMAGE_NAME=gcr.io/earthengine-project/datalab-ee:latest
export WORKSPACE=${HOME}/workspace/datalab-ee
mkdir -p $WORKSPACE
cd $WORKSPACE
```

Run the Container (detached):

```
docker run -it -d -p "127.0.0.1:8081:8080" -v "$WORKSPACE:/content" -e
↪"PROJECT_ID=$GCP_PROJECT_ID" $CONTAINER_IMAGE_NAME
```

Establish a secure connection with Caddy:

```
echo "$(hostname)
proxy / 127.0.0.1:8081 {
    websocket
    transparent
}
" > Caddyfile
curl https://getcaddy.com | bash -s personal http.nobots
caddy
```

2.5.2 *Use Google Colaboratory*

Google has its own Jupyter service called 'Colaboratory'

https://colab.research.google.com/

2.5.3 Download Data

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home

CHAPTER 3

Prerequisites

3.1 Downloads, access, and services

In order to complete this tutorial you will need access to the following services/software

Prerequisite	Preparation/Notes	Link/Download
CyVerse account	You will need a CyVerse account to complete this exercise	Register
Atmosphere access	You must have access to Atmosphere	Request Access
CyVerse Data Store allocation increase (Optional)	You must be registered for CyVerse	Request Increase (form #2)
Jetstream access (Optional)	You must have registered with XSEDE	Request Access
Cyberduck (Optional)	Standalone program for uploading/downloading data to Data Store	Download
Windows 10 Linux Subsystem (Optional)	Install Ubuntu Bash on a Windows OS	Installation Instructions

3.2 Platform(s)

We will use the following CyVerse platform(s):

Platform	Interface	Link	Platform Documentation	Quick Start
Data Store	GUI/Command line	Data Store	Data Store Manual	Guide
Discovery Environment	Jupyter Notebooks / non-interactive Docker jobs	Discovery Environment	DE Manual	Guide
Atmosphere	Command line (ssh) and/or Desktop (VNC)	Atmosphere	Atmosphere Manual	Guide

3.3 Application(s) used

Discovery Environment App(s):

Name	Version	Location	App Link	Notes/other
Jupyter Notebooks	0.1	TBA	TBA	TBA

Atmosphere Image(s):

Here are the tested Ubuntu images.

Warning: The latest version of Ubuntu (18.04) may not have current packages for some software.

Image Name	Version	Description	Link
Ubuntu 18.04 GUI	1.0	Ubuntu 18.04 GUI XFCE Base	Image
Ubuntu 18.04 non-GUI	1.0	Ubuntu 18.04 non-GUI Base	Image
Ubuntu 16.04 GUI	2.1	Ubuntu 16.04 GUI XFCE Base	Image
Ubuntu 16.04 non-GUI	1.6	Ubuntu 16.04 non-GUI Base	Image

Jetstream Image(s):

3.4 Input and example data

In order to complete this tutorial you will need to have the following inputs prepared

Input File(s)	Format	Preparation/Notes	Example Data
Discrete lidar	*.laz, *.las, *.xyz, *.bin	classified	Data Portal
Orthophotography	*.tif		
Hyperspectral	*.hdf		

Fix or improve this documentation

- Search for an answer: **|CyVerse Learning Center|**
- Ask us for help: click **|Intercom|** on the lower right-hand side of the page
- Report an issue or submit a change: **|Github Repo Link|**
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home