CyVerse Documentation

Release 2.0

CyVerse

Apr 17, 2023

Contents

 Learning Center Home

Times: 09:00 - 14:00 MST, UTC-7

Dates: Thursday 5th, Friday 6th, and Saturday 7th (optional day) November 2020

Ready to join the workshop? Follow these steps:

Step 1: Create CyVerse Account (free)

Please use your institutional email address, and if you don't have an - sign up for one of those too, they're super important and valuable!

Step 2: Sign up for Workshop

Step 3: Review this website – all of our training materials will be posted or linked from here

CHAPTER 1

Goal

The goals of the workshop are to allow you as new data scientists to leave with an understanding of the NEON Data API and working with NEON AOP data and to introduce CyVerse as a platform for conducting data intensive scientific research.

You will have opportunities to work in your preferred Integrated Development Environment (IDE) in the public research cyberinfrastructure. CyVerse enables us to work with large and very large analyses. You will be able to work with NEON AOP data across many sites and many years worth of data without ever having to "download" anything over your local internet service provider.

Tutorial Maintainer(s)

Who to contact if this guide needs fixing.

Maintainer	Institution	GitHub Username
Tyson Swetnam	CyVerse / University of Arizona	`tyson-swetnam`
Bridget Hass	NEON / Battelle Inc	`bridgethass`
Donal O'Leary	NEON / Battelle Inc	`donal-at-NEON`

Content Links

Use the table of contents on the left side of the page to navigate

CYVERSE®
LEARNING

🏠 Learning Center Home

3.1 Code of Conduct

All attendees, speakers, sponsors and volunteers are required to agree with the following code of conduct. Organisers will enforce this code throughout the event. We expect cooperation from all participants to help ensure a safe, inclusive, and collaborative environment for everybody.

Harassment by any individual will not be tolerated and may result in the individual being removed from the Workshop.

Harassment includes: offensive verbal comments related to gender, gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, religion, technology choices, sexual images in public spaces, deliberate intimidation, stalking, following, harassing photography or recording, sustained disruption of talks or other events, inappropriate physical contact, and unwelcome sexual attention.

Workshop staff are also subject to the anti-harassment policy. In particular, staff should not use sexualised images, activities, or other material that conflicts with the code of conduct.

Participants who are asked to stop any harassing behavior are expected to comply immediately. If a participant engages in harassing behavior, the workshop organisers may take any action they deem appropriate, including warning the offender or expulsion from the workshop with no refund.

If you are being harassed, or notice that someone else is being harassed, or have any other concerns, please contact a member of the workshop staff immediately. Workshop staff will be happy to help participants contact local law enforcement, provide escorts, or otherwise assist those experiencing harassment to feel safe for the duration of the workshop. We value your attendance.

We expect participants to follow these rules at conference and workshop venues and conference-related social events.

See http://www.ashedryden.com/blog/codes-of-conduct-101-faq or The Carpentries https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html for more information on Codes of Conduct.

Fix or improve this documentation:

- On Github:

- Send feedback: Tutorials@CyVerse.org

Learning Center Home

3.2 Agenda

We will attempt to cover each of these Content pages during the workshop.

As we run out of time for each section, you may need to complete the material self-paced as homework after the in-person sessions end each day.

Zoom Info:

URL	Meeting ID	Password
Zoom US	812 0462 6687	`cyverse`

3.2.1 *Day 1 2020-11-05*

Time (AZT MST)	Activity	Lead Personnel	Content	Additional Links
0900-0910	Briefing	Tyson	Overview of course materials	This website
0910-0930	CyVerse walkthrough	Tyson	Discovery Environment Overview	https://de.cyverse.org
0930-0935	Launching an application	Tyson	Visual Interactive Computing	
0935-1000	Managing your code for reproducibility	Tyson	Version Control with GitHub	
1000-1010	Break			
1010-1100	Intro to NEON Data API	Donal	Download and Explore NEON Data	https://www.neonscience.org/download-explore-neon-data
1100-1200	Brunch, Early-, Regular-, Late- Lunch Break		Eat. Chat.	
1200-1250	RStudio Activity	Donal	Accessing and working with NEON geolocation data	https://www.neonscience.org/neon-spatial-data-basics
1250-1300	Break			
1300-1330	RStudio Activity (cont.)	Tyson	RStudio & Shiny Apps	GitHub CyVerse NEON Shiny App
1330-1350	Data Management	Tyson	Managing data in the cloud	
1350-1400	Wrap-up		After-action review (what went well, wrong, could be done better)	

3.2.2 *Day 2 2020-11-06*

Time (AZT MST)	Activity	Lead Personnel	Content	Links
0900-0910	Briefing	Tyson	Overview of course materials	This website
0910-1000	RStudio Activity	Donal	Creating a Canopy Height Model from lidar-derived rasters	https://www.neonscience.org/create-chm-rasters-r
1000-1010	Break			
1010-1100	RStudio Activity	Donal	Calculating Forest Structural Diversity Metrics from NEON lidar data	https://www.neonscience.org/structural-diversity-discrete-return
Optional:	RStudio Activity	Donal	NEON AOP lidar w/ gedi	https://katharynduffy.github.io/neon-aop.html#matching-gedi-waveforms-with-neon-aop-lidar-pointclouds
1100-1200	Brunch, Early-, Regular-, Late-Lunch Break	Eat. Chat.		
1200-1250	RStudio Activity	Donal	Introduction to Hyperspectral Remote Sensing Data	https://www.neonscience.org/intro-hsi-r-series
1250-1300	Break			
1300-1350	RStudio Activity		Free time to explore tools, ask questions.	
1350-1400	Wrap-up		After-action review (what went well, wrong, could be done better)	

3.2.3 *Day 3 2020-11-07*

Time (AZT MST)	Activity	Lead Personnel	Content	Links
0900-0910	Briefing		Overview of course materials	This website
0910-09:30	Jupyter, Remote Desktops	Tyson	More on VICE apps in Discovery Environment	
0930-1000	Jupyter Lab	Bridget	Working with NEON AOP data in Python	Introduction to NEON Data in Python using Jupyter Labs
1000-1010	Break			
1010-1100	Jupyter Lab	Bridget	Working with NEON Hyperspectral	
1100-1200	Brunch, Early-, Regular-, Late- Lunch Break	Eat. Chat.		
1200-1250	Remote Desktops	Tyson	QGIS and Desktops	
1250-1300	Break			
1300-1350	Batch processing lidar	Tyson	PDAL and Entwine Point Tiles	
1350-1400	Wrap-up		After-action review (what went well, wrong, could be done better)	

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click ![icon] on the lower right-hand side of the page

- Report an issue or submit a change:

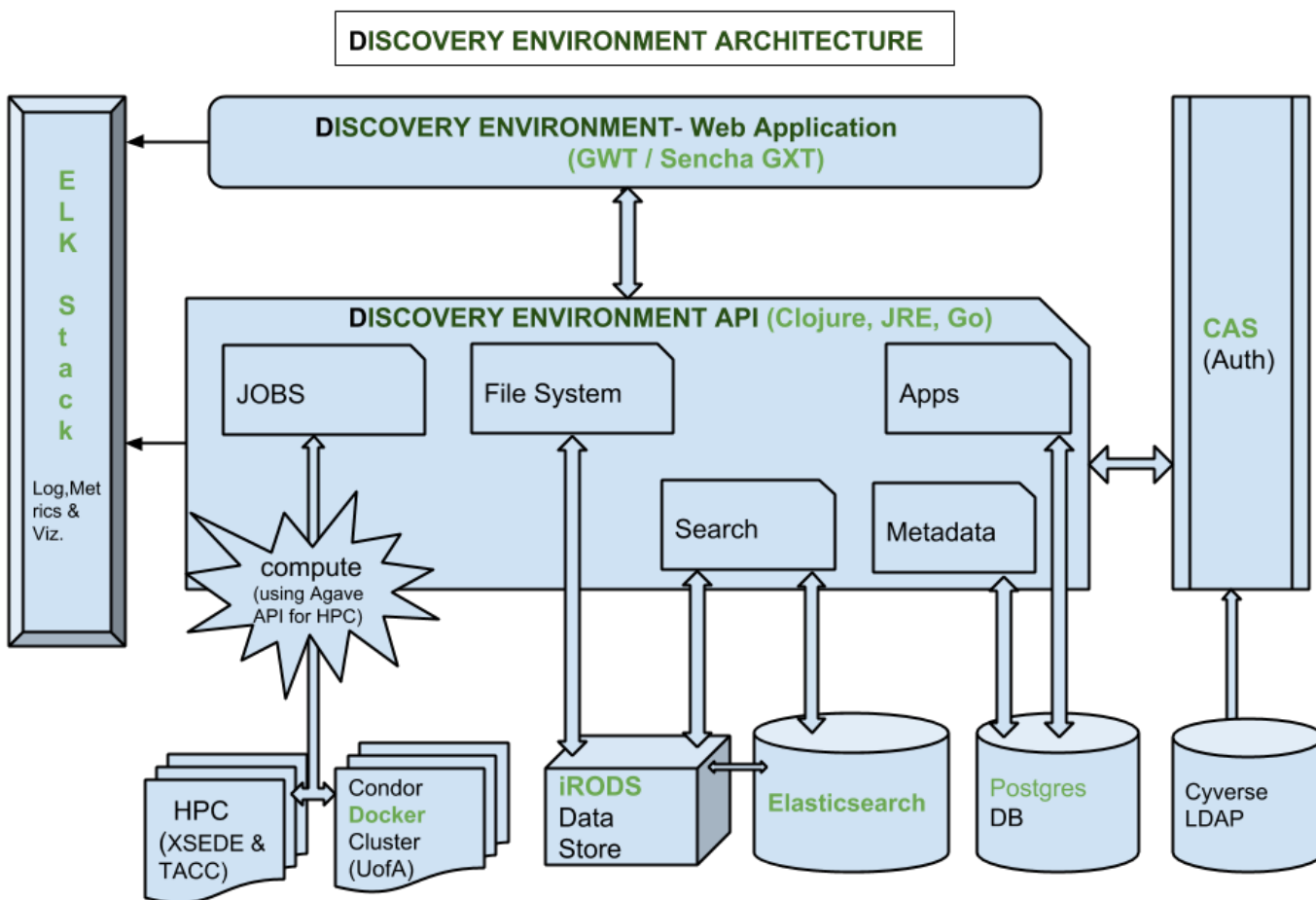- Send feedback: learning@CyVerse.org

![home icon] Learning Center Home

![home icon]

3.3 Discovery Environment

Description:

After you have created your CyVerse account and been granted access to the visual interactive computing environment (VICE) portion of the Discovery Environment data science workbench, you'll be able to start a GUI based app.

Figure: More than you wanted to know about how this stuff works.

3.3.1 *The Data Store*

The CyVerse Data Store uses iRODS as a cloud storage system. When you see the data in the browser, it looks like a conventional file tree with folders and filenames.

CyVerse started out with the project name "iPlant Collaborative", and our data store still retains the `iplant` zone name in iRODS.

Windows vs Linux

If you're a Windows user, you're used to your file path looking something like this:

```
C:\Documents\Folder Name\File Name.pdf
```

The volume or drive is assigned a letter, e.g. `C:\`, and there may be spaces in the folders and file names.

In Linux, the folder paths use a forward slash `/` , and do not add a letter to the root volume name. Spaces in folders and file names are highly discouraged and are unusable on the command line as a space is a special character which separates arguments.

```
/home/username/documents/folder_name/file_name.pdf
```
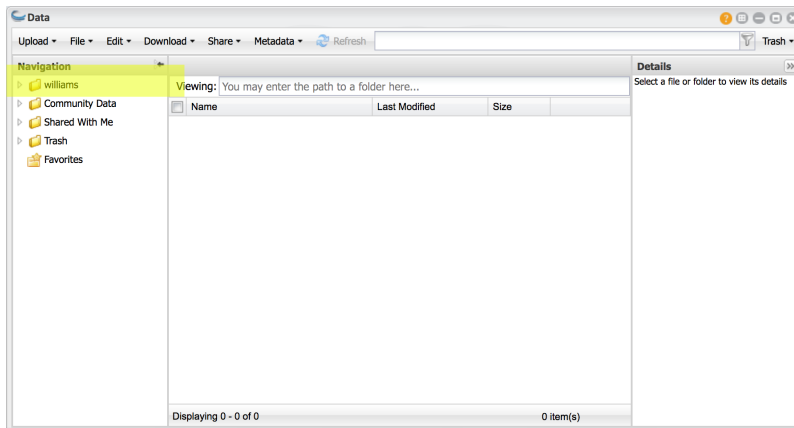
Using special case styles like `camelCase`, `snake_case`, `PascalCase`, or `kebab-case` helps to differentiate between words in folder and filenames.

Walkthrough

1. Log into the Discovery Environment: https://de.cyverse.org

2. Click the icon labeled "**Data**"

3. This opens a file explorer, in your personal username space

Your space has a path in the data store, e.g.

```
/iplant/home/username
```

This is your personal space, it is private to you.

You can create new folders and upload or download files.

You can change the permissions of these files and folders to share them with your collaborators or the public.

4. Create a new folder called `NEON_Downloads`

The new folder should be located:

```
/iplant/home/username/NEON_Downloads
```

This folder is private, only you can see it.

5. Sharing a folder

Click on the 'Share' tab and 'Share with Collaborators' option.

Type in a user's given name and it should be searched and pop up. You will not see their `username`, only their identity and institution information.

You have three options in granting privileges to others: `read` `write` or `own`

- `read` permissions allows the users to see and download the files and folder
- `write` permissions allows the user to modify the file and folder name.
- `own` permissions allows the user to modify the file and folder **and the ability to create and delete**

Type in 'Public User' – adding this user will share the directory with all other CyVerse users when they are authenticated.

Type in 'Anonymous User' – adding this user will share the directory with the open internet (it will become visible on the internet via https://data.cyverse.org/dav-anon/

6. Look into the Community Data folder

These are public folders that have been 'shared' with all CyVerse users or with the open internet (via the Anonymous User group):

```
/iplant/home/shared/
```

Navigate to 'NEON_workshop/' and 'data/'

```
/iplant/home/shared/NEON_workshop/data
```

There are some sample NEON AOP Data in here that we'll get to tomorrow.

There are many more Community Data folders in CyVerse that you cannot see – that's because they have not been shared with the 'Public' or 'Anonymous' user groups.

You do not have `write` or `own` permission on any Community folders, so you cannot change them.

7. Look into the 'Shared with Me' folder

These folders are private user accounts that have public data in them or have been shared with you personally.

8. Access the Data Store from Cyberduck (Windows and Mac OS X only)

Download program onto your local computer.

Add the file to your installation. This will request your CyVerse credentials.

View the contents of your Data Store. Drag and drop files and Cyberduck will upload / download them for you.

9. Access the Data Store from WebDav (browser based)

In your browser, navigate to https://data.cyverse.org

WebDav is a read-only space for viewing data that are already in the data store

The `https://data.cyverse.org/dav/` folder path requires authentication with your CyVerse username and password

The `https://data.cyverse.org/dav-anon/` folder path is public and anonymous read only to anyone on the interent.

Where does your data live?

When you download data from the internet to your local computer they're isolated. How do you share them back with your team?

Many of us use services like Box or Google Drive to hold our files. CyberDuck and its command line client duck.sh also access these platforms.

These services are incredibly useful.

However, file storage and sharing platforms like Box and Drive were not designed for machine readability and rapid requests for many (i.e. thousands to millions) of requests by anonymous users or even by trusted users. (see Google Drive vs Google Cloud for an explanation)

Conventional file services like `ftp://` (file transfer protocol), function over HTTP and HTTPS. The same is true for Amazon Web Services `s3` storage object buckets. (S3 explained)

How to work with your data in CyVerse

Downloading data from commercial cloud storage providors directly into CyVerse Data Store requires you have a running instance (virtual machine, or container in Discovery Environment) where the data can be staged before moving them onto the Data Store.

Uploading data to CyVerse is dependent upon your local internet service provider.

3.3.2 *The App Catalog*

If you signed up for the workshop, you will have already been added to the NEON Community group. We have added a couple of apps that have all of the tools needed for the workshop.

These Apps are yours to use! You can install new packages and software into them, but if that becomes too time consuming, consider learning about how to integrate your own Tools and Apps using the .

App – a graphical interface for starting a "Tool" here in the Discovery Environment. The App window can be customized to use any set of conditionals, parameters, resource requirements, input data, or output folders needed to do your analysis. An App can be "**interactive**" like the RStudio or Jupyter Lab, "**executable**" like a command line script, or "**OSG**" for high throughput parallel computing on the Open Science Grid.

Tool – a "Tool" is a Docker container which has been added to the Discovery Environment tool manager. It must be public on the Docker Hub or another Docker Registry (e.g. quay.io, NVIDIA NGC, etc.). After the tool manager template has been completed, the container will be added to the Discovery Environment. Click the "Manage Tools" with the wrench icon in the Apps window, to add your containers. After the Tool is integrated a private App can be created.

Walkthrough

9. Click the icon labeled **Apps**

10. Under **Categories** and "My Apps" and "My Communities" you should see a group called NEON
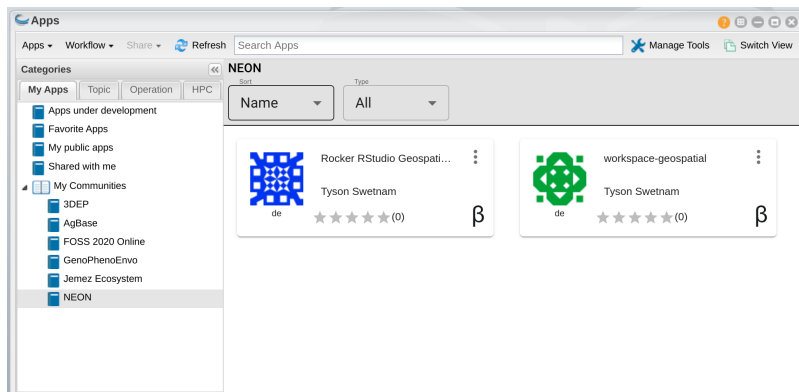
11. In the group are a couple of shared apps with the |**beta**| tag.

In the next section, we're going to go over starting one of these apps, and beginning an Analysis, which you can view using the icon labeled 'Analyses'

3.3.3 *Analyses*

Walkthrough

12. Click the icon labeled **Analyses**

13. In the next section, we'll cover running your own Analysis. When you start an "App" the running analysis will appear in the

Description of output and results

You should now understand the basics of the Classic Discovery Environment Interface.

- Data Store
- Apps
- Analyses

Fix or improve this documentation

- Search for an answer:
- Ask us for help: click on the lower right-hand side of the page
- Report an issue or submit a change:
- Send feedback: learning@CyVerse.org

 Learning Center Home

[Learning Center Home](#)

3.4 Your Workbench

Description:

We have integrated several Docker containers into the CyVerse Discovery Environment workbench platform, and made them "Public" for anyone with a CyVerse account to use.

We have also copied some of the NEON example data sets to the CyVerse Data Store, along with the completed `.ipynb` and `.Rmd` notebooks and `.r` scripts

In this section, we're going to walk through the required steps of starting an interactive analysis, and shutting it down when you're done.
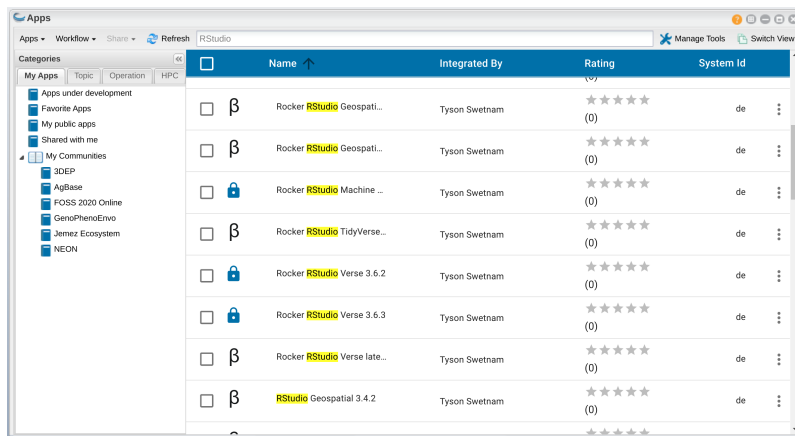
3.4.1 *Starting a VICE app*

1. Log into the Discovery Environment https://de.cyverse.org

Or click this quick launch button to be taken directly to the app:

- RStudio:

2. Click the 'Apps' button and a new window should open.

Search using the query "rstudio geospatial" and see what comes up.

Also, under the "My Apps" tab click on the "My Communities" section and "NEON" Group should appear.

Click on the Hyperlinked name and a new window should open.

3. In the pop-up for the particular App you have a few options.

i. `Analysis Name` - you can change the name of the analysis if you like, the default name is typically `<the-app-name>_analysis1`

Your analysis will run, and when it completes, it will write any data that you have in the WORKING DIRECTORY of the container back to the Data Store in your Analyses folder, e.g. `/iplant/home/<username>/analyses/<the-app-name>_analysis1-<DATE-TIME-of-job-starting>`

ii. The second section is `Input Data` and has options for adding (1) a folder, (2) a single file, or (3) multiple files.

For our use case today, we're going to add a folder from the data store.
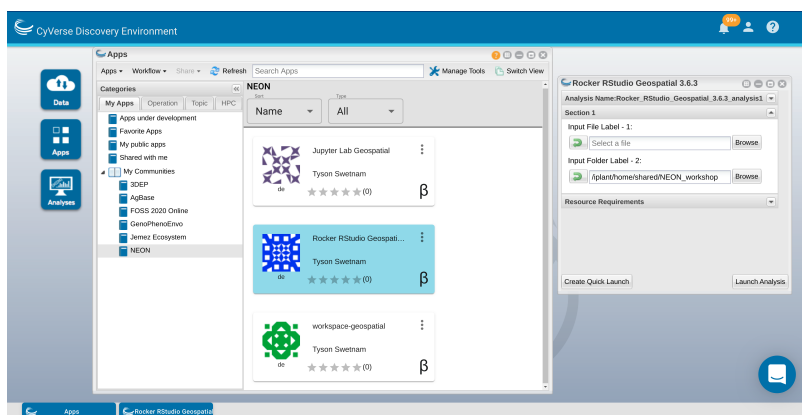
Add the path: `/iplant/home/shared/NEON_workshop/`
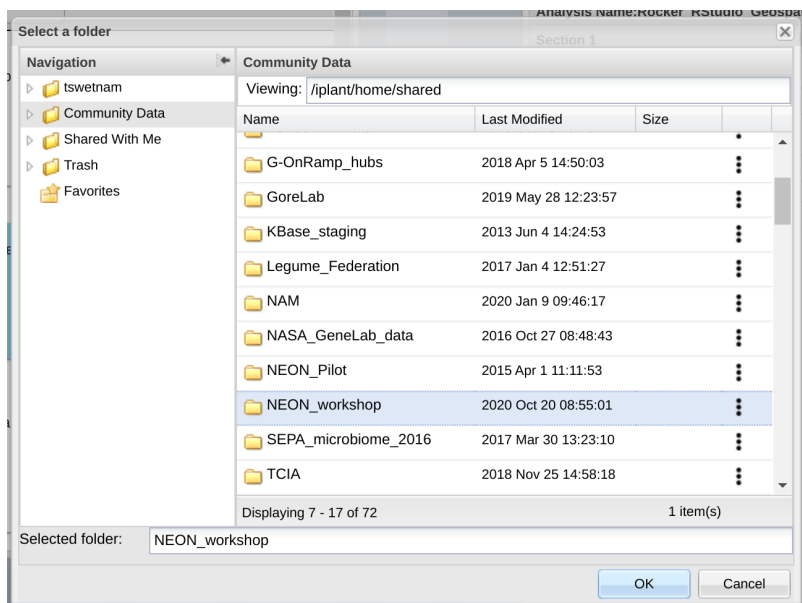
Input Data (not required)

When you launch a new VICE app, you can add data to it before it is launched. If you do this, it will slow down the launch, as the service must copy the data from the data store into your new instance before it becomes available.

A faster option is to start the container without the data, and then copy the data into the running container later using WebDav, iCommands, or a file system mount.

---

iii. The third section is `Resource Requirements`, again you can leave the default settings, or you can modify them.

You can set the `Minimum CPU` to the minimum number of cores your app requires. If you do not select anything, the app will still be able to use multiple cores on the shared node on which it is deployed.

You can set the `Minimum RAM` to the minimum number of GB of RAM you think your app requires.

You can set the `Minimum Disk Space` to the minimum amount of scratch space you think your data will need.
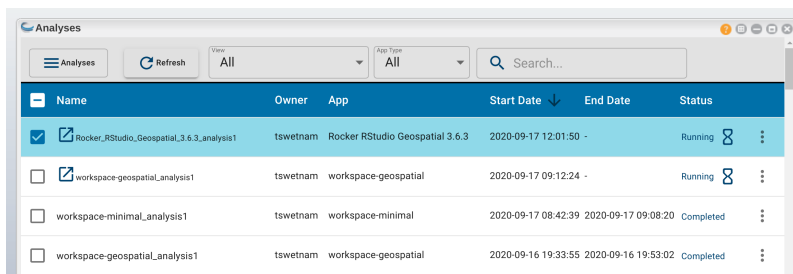
iv. After you've set the analysis name, input data, and resource requirements, click `Launch Analysis`

Watch closely, you'll see a notification at the top of your screen and in the Bell icon in the upper right.

4. Open the icon labeled "Analyses" to view your running analysis.

Look for your App by Name.

See the square icon with an arrow pointed up to the right? Click on that icon and a new browser tab will open.

Alternately, click the Bell icon in the upper right to see your notifications, you should see 'Access your running analysis here.' as an option. Click on that link and a new browser tab will open.

5. Having clicked on either of the hyperlinked icons in Step 4 should result in a new tab opening in your browser.

Your Discovery Environment Browser tab should still be open at `https://de.cyverse.org/de/`

The new URL for the App should be something like `https://af7664685.cyverse.run/`.

Your app is now ready for you to begin work.

6. Return to your Discovery Environment Browser tab.

On the right side of the Analysis window there is an icon with three vertical dots. Click on that icon.

You have multiple options available here:
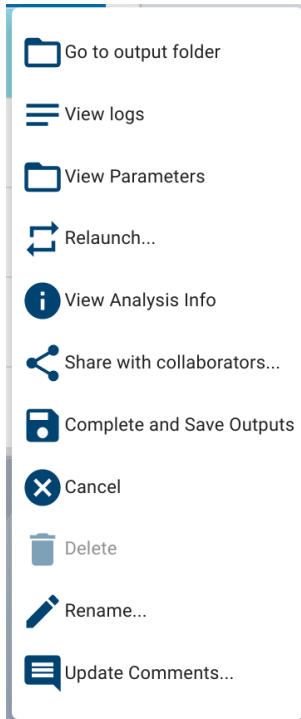
`Go to output folder` will take you to the output folder in yout `/analyses` Data store path. This output folder will not exist until AFTER you end the running analysis.

`View Logs` will show you the echo of the container job starting – this is very useful for debugging containers that are not acting appropriately.

`View Parameters` will show you the parameters you set when launched – good for reviewing stopped containers.

`Relaunch` – if you have stopped your analysis, you can re-launch it with the same parameters as before. Useful for coming back to a second or third day.

`View Analysis Info` will show you the status and Analysis ID – this is useful when asking for support from the CyVerse team.

`Share with Collaborators` you can share your running analysis with other CyVerse users or team members.

`Complete and Save Outputs` **DANGER Zone** – when you click this, your job will be stopped and your analysis results will be copied to the `/analyses` folder

`Cancel` **DANGER Zone** – when you click this, your job will be stopped and your results will not be written back to the data store.

`Delete` – this option is only available after you have stopped your App and want to remove the information from your Analysis window

`Rename` You can rename your analysis

`Update Comments` You can update your comments about your analysis.

3.4.2 *Stopping a VICE app*

1. After you've finished working on your instance, it is conscientious to shut down your analysis and free up share resources for other researchers.

2. Make sure that your data that you've copied into the running Analysis have been moved to another storage location on the internet, or back into your data store in another file path.

3. Analysis data that are in the single working directory "WORKDIR" of the container that the App has set in its configuration file will be saved back to the Data Store in your `/iplant/home/<username>/analyses/` `<the-app-name>_analysis1-<DATE-TIME-of-job-starting>` folder.

4. When the job ends, any data that are in other `/temp` or `/home/username` folders in the running App WILL NOT BE SAVED or written back to the Data Store.

5. Click on the icon with the three vertical dots or select the app with the check box and click on the `Analyses` buton in the upper left of the Analyses window.

> `Complete and Save Outputs` **DANGER Zone** – when you click this, your job will be stopped and your analysis results will be copied to the `/analyses` folder
>
> `Cancel` **DANGER Zone** – when you click this, your job will be stopped and your results will not be written back to the data store.

6. Depending on how much data you have in the WORKDIR folder of the running App, the analysis may take a few minutes to complete and shut down.

7. When the Analysis completes, you will now have the option to Delete the finished job.

Description of output and results

IMPORTANT: These data are currently running in a virtual machine, which will soon be going away, either when it times out, or you turn it off. You need to make sure that your data are moved to a data repository, like the CyVerse Data Store, or downloaded to your localhost, before you turn this analysis off.

Any data which are in the working directory of `/iplant/home/<username>/analyses/` `<the-app-name>_analysis1-<DATE-TIME-of-job-starting>` directory). Any data which are in other paths will not be preserved

---

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

 Learning Center Home

 Learning Center Home

3.5 Version Control with GitHub

Description:

We want to keep track of our work in a place that we can also share it with our collaborators, and eventually with the public.

GitHub is a version control system which tracks changes to code and shares it across teams. While it is predominantly used by software engineers, it is increasingly used by researchers in open science, for the same reasons.

3.5.1 *Create a new Repository*

1. Log into GitHub with your GitHub username

2. click on the + icon in the upper right of the screen.

> Select 'New Repository'

3. Give the new repository a name, e.g. `neon-aop-workshop`

> If you have an existing repo with this name it will be disallowed.

4. Allow the repository to be "Public" – if you want, you can make it "Private", and it will add the requirement that you authenticate to GitHub everytime you want to clone the repository somewhere else.

5. Select the option to create a `README.md` file with the repository. This will create a blank markdown file that you can populate with text later.

6. (Optional) Select a License for your code

> Choosing a license is important to ensuring that your work is properly attributed and cited in the future.

> ---
> **Choosing a License**
>
> choosealicense.com offers support and answers questions about selecting the right license.
> ---

7. You're ready to go back to your CyVerse RStudio browser tab.

3.5.2 *Clone your Repository into RStudio Server running in CyVerse*

RStudio can create a R Project using Version Control with `git` or `svn` (another version control platform).

Creating a R project with Version Control will allow you to sync changes back up to GitHub while you're working and when you're finished.

`git` also allows you to pull other GitHub repositories and work with existing code and analysis notebooks, thus enabling repeatability.

8. In RStudio, select 'File' then 'New Project' and then 'Version Control'

9. The repository will be copied onto your instance and you'll be in a directory with a new R project file.

This is a local copy of the `git` repository from GitHub.

Any changes thare are made locally on this machine will not affect the GitHub repository from which you got this.

10. Adding new files to the repository.

   Now that there is a copy of the repo on your instance, you're ready to start making changes and adding new scripts.

11. Updating a `.gitignore` file

   `git` is useful for tracking code – but it is not intended to track your data files.

   The best practice is to NOT keep your data in the same directory as the `git` repository.

   However, you can add a `.gitignore` file and update it with the various types of files you want `git` to not track or to submit back to GitHub when you commit your changes.

   When you use R Studio to create a Version Control project, it will generate a `.gitignore` file for you. The default files that it will ignore are related to your local R Studio environment:

```
.Rproj.user
.Rhistory
.RData
.Ruserdata
```

   You can update the `.gitignore` file so that it will also NOT track data type files:

```
.Rproj.user
.Rhistory
.RData
.Ruserdata
*.csv
*.tif
*.laz
*.las
*.hdf5
*.hd
*.txt
```

   this `.gitignore` will ignore all files with the `*` and given file extensions.

   An alternate way of making sure that you track your files is to include `!` only certain file types:

```
# ignores everything ...
/*
# ... but the following
!*.R
!*.Rmd
!*.Rproj
```

   If you own the GitHub repository, you will be able to make changes "commits" to the repository and "push" them back to the GitHub.

   If you pulled this repository from someone else, and you make commits and submit a "push" it back to the other person's GitHub, it will ask you to enter some user identification.

   This process creates something called a "pull request" on GitHub, where the owner of the repository can see who made the changes, and review whether or not they agree with these changes. They can then choose to approve the request, and the changes will update their repository.

3.5.3 *Add your code to the Repository*

12. Copy the contents of our other NEON exercises into the new directory.

    These new files are not yet added to the `git` module tracking.

13. Configure `git` for the first time

    Because we're working on a remote instance, the `git` configuration has not been set.

    To configure `git` for sending requests to GitHub, set the email address and a name for your remote:

    ```
    git config --global user.name "Your Name"
    git config --global user.email "Your@email.address"
    ```

14. Create a new "Branch"

    The repository has a version called `master`

    Create a new branch called `main`

    We'll do this in the RStudio Git menu, but it can also be done in the terminal:

    ```
    git checkout -b main
    ```

15. Add tracking for the new files in the repository and create a "commit" message

    RStudio's Git integration should show you which files are not tracked by `git`. You can select the check boxes for each file and add them.

    You need to create a "commit" message which briefly explains the changes you're about to make.

16. Push your changes to the GitHub.

    Your updates are now ready to be submitted to the GitHub from RStudio.

    Because you're sending these files from a remote computer, your changes will not automatically be accepted by the GitHub.

3.5.4 *Review and Accept your own Pull Request*

17. Review and Accept your own "Pull Request"

    Go to your GitHub user profile and select the repository.

    Your changes should now be registered as a "Pull Request"

    You will see your "commit" message, and be able to review the new files and file changes.

18. Your files are now saved in GitHub, under a new branch called `main`

    You can safely delete the `master` branch in RStudio and on GitHub.

    On GitHub, above the list of files, click the `branches` hyperlink, select the `master` branch and delete it.

    Official instructions

Black Lives Matter

Github has stated that they will remove the use of the term `master` as the default branch name from their platform, but that is not yet the case.

Website js script for changing `master` to `main`

Description of output and results

You should now have a tracked version control of your workshop project, with all of the pre-existing scripts. You can re-use this repository on your local computer, or somewhere else in the future.

Hopefully, you now understand how GitHub can be used to share code and analyses, and to do your science!

NEON maintains a large library of pre-written scripts on their GitHub repository: https://github.com/neonscience

---

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click ![icon] on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

![CyVerse Learning logo] Learning Center Home

![home icon] Learning Center Home

3.6 Using the NEON Shiny App in RStudio-Server

Description:

The Download and Explore NEON Data API tutorial covers the basics of downloading data direcly into R and RStudio via the `neonUtilities` R package.

Our team created a NEON Shiny App for interfacing with the NEON API in R Studio in a graphical manner.

The Shiny app can be launched using Docker, in RStudio or an RStudio-Server.

When started the Shiny app will create a new folder called `~/NEON_Downloads` in the home working directory. For the R Studio instance on CyVerse, this is set as `/home/rstudio/NEON_Downloads`.

Data that are selected and downloaded go into this folder and are organized using the same ontology as the NEON Data API.

3.6.1 *Prerequisite*

In the previous section, you should have started a RStudio Server in the Discovery Environment https://de.cyverse.org

If you have not, do so again

3.6.2 *Download the Shiny App*

1. Have an instance of RStudio-Server in VICE already started, see Your Workspace for details.

2. Click on the `Terminal` tab in the R Console

3. Type or copy this text into the linux terminal:

```
git clone https://github.com/cyverse-gis/neon-shiny-browser
```

4. The contents of the Git Repository should now be in your workspace files in the lower right of RStudio

5. Set the working directory to the `neon-shiny-browser` directory

3.6.3 *Start the Shiny App*

RStudio-Server has a feature called "**Jobs**"" which can run the Shiny app as a background process. This will keep the R Console active and allow us to continue work in the R Studio while the App is running at the same time.

6. Select the `Jobs` option in the Console.

7. Set the working directory as `~/neon-shiny-browser` and select the `background.R` script.

8. Start the Job as a background process.

9. This particular App will install a few missing R packages before it starts. Don't worry, it will do this automatically. After it has installed the missing package dependencies, it should begin to echo out logs and then start to run on the `localhost` address number `127.0.0.1` using a randomly assigned PORT number:

```
(A bunch of stuff above this...)

Reading       layer `D19_HEAL_R3_P1_v1' from data source `/home/
↪rstudio/     neon-shi     ny-browser/NEON-data/Flightdata/        ↵
↪Flight_b     oundaries_2017/D19_HEAL_R3_P1_v1      .geojson'↵
↪using driver       `GeoJSON      '
Simple f     e      ature collection with 1 feature      and 4↵
↪fields
geometry               type:  MULTIPOLYGON
dimensio      n      :       XY
bbox:                xmin: -149.3151 ymin: 63      .82981 xmax: -
↪149.1116      ymax: 63.93015
CRS:          4326

Listening on http://127.0.0.1:5716
```

The part we want to copy is the `http://127.0.0.1:5716`, the `5716` number here will be randomly assigned

10. Back in the R Console, type the following using the PORT number from the job:

```
rstudioapi::viewer("http://127.0.0.1:5716")
```

11. The App should start to run and appear in the lower right "Viewer". You can pop the app out into its own browser tab, and begin to use it.

3.6.4 *Downloading Data via the NEON API*

12. Browse the App and find a dataset that you're interested in downloading.

13. AOP data use a slightly different protocol in the NEON Data API, so make sure to select the AOP data check box when you are ready.

14. After you've initiated the download, the data will begin being downloaded to the `~/NEON_Downloads` folder.

When the data download is complete you will get a notification.

15. Important: Your data are in a directory which is not set as the WORKDIR by the Docker container in the Discovery Environment. This means that your downloaded data will NOT be saved back to the Data Store when your analysis completes.

Output/Results

Output	Description	Notes
NEON Data!	Data should be in the `/home/rstudio/NEON_Downloads` directory	These data use the same file-tree hierarchy as the NEON Data API.

Description of output and results

IMPORTANT: These data are currently running in a virtual machine, which will soon be going away, either when it times out, or you turn it off. You need to make sure that your data are moved to a data repository, like the CyVerse Data Store, or downloaded to your localhost, before you turn this analysis off.

Any data which are in the working directory of the instance (likely the `/home/rstudio/` directory will be preserved in the `/iplant/home/<username>/analyses/<the-app-name>_analysis1-<DATE-TIME-of-job-starting>` directory). Any data which are in other paths will not be preserved

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

 Learning Center Home

3.7 Managing your data in the cloud

Description:

After you've become familiar with downloading data from NEON Data API, or from other resources on the internet, into your cloud instances, you're going to be in a situation where you need to move them and store them somewhere more permanently.

Its important to accept that many of these public data repositories are stable and that data will be available from them in the future.

This means that you **should not create copies of original data** unless you are in a situation where the data are very large and downloading them again is prohibitive of your time.

3.7.1 *Setting up iCommands*

CyVerse Data Store uses a platform called iRODS to manage its data. iRODS has a command line application called iCommands for moving data over the terminal.

First, we need to initiate a connection to the CyVerse iRODS.

1. In the Terminal type in `iinit`

> This should echo out a set of information in the terminal:

```
One or more fields in your iRODS environment file (irods_environment.json)␣
↪are missing;
please enter them.

Enter the host name (DNS) of the server to connect to:
```

2. Enter in the following data for each field:

```
Enter the host name (DNS) of the server to connect to: data.cyverse.org
Enter the port number: 1247
Enter your irods user name: user_name
Enter your irods zone: iplant
Those values will be added to your environment file (for use by
other iCommands) if the login succeeds.

Enter your current iRODS password:
```

- host name (DNS): `data.cyverse.org`

- port number: `1247`

- irods user name: `<your CyVerse username>`

- irods zone: `iplant`

- current iRODS password: `<your current password>`

3. You should now be authenticated to the Data Store.

To test, try typing `ils`

If you do not echo back anything, try Step 2. again

3.7.2 *Uploading with iCommands*

4. Type in `ils`

```
rstudio@a4bdcc31:~$ ils

/iplant/home/username:
C- /iplant/home/username/analyses
C- /iplant/home/username/NEON_Downloads
```

You should now see the contents of your personal Data Store

5. Upload a single file to the Data Store using `iput`

You need to select the file you want to copy, and the location in the Data Store you want to copy it to.

```
iput -KPvf /home/rstudio/neon-shiny-browser/background.R /iplant/home/
↪username/NEON_Downloads/
```

This command will take a single file `background.R` and copy it from the container to the Data Store folder `/iplant/home/username/NEON_Downloads/`

The flags `K`, `P`, `v`, and `f` are described in the help file.

6. Upload a folder with recursive sub-folders and files

Next, we want to upload an entire directory with many folders and files in it.

```
iput -KPbrvf /home/rstudio/NEON_Downloads/NEON_HARV_DP1.30003.001_
↪2019 /iplant/home/<your-user-name>/NEON_Downloads/
```

I have added the flags `b` for bulk, and `r` for recursive to the `iput` command. This will upload the entire directory `NEON_HARV_DP1.30003.001_2019` to the data store.

7. The `P` flag for Progressive and `v` flag for verbose will echo out the progress of the upload until it completes.

When it is complete, the terminal should be available again.

To test whether your files are now in CyVerse try:

```
ils /iplant/home/<your-user-name>/NEON_Downloads/

# and then

ils /iplant/home/<your-user-name>/NEON_Downloads/NEON_HARV_DP1.30003.001_2019
```

You should be able to see the contents of your directory in the Data Store

8. These files are now in your private user space. No one can see them, but if you did want to share them, you can do so by modifying their permissions directly in the Discovery Environment, as shown in Step 1, or by using the following commands:

```
ichmod
```

Follow the instructions in the help menu to set the user privileges and ownership.

This example makes your data directory public on the internet as a read-only archive:

```
ichmod read anonymous /iplant/home/<your-user-name>/NEON_Downloads/
```

3.7.3 *Downloading with iCommands*

It is also likely that you're going to download data from the Data Store into your running Apps

9. Use the `ils` command to look for some shared data in the Data Store

```
ils /iplant/home/username/NEON_Downloads
```

10. Download a file using `iget`

```
iget -KPvf /iplant/home/username/NEON_Downloads/benchmarking.rmd
```

This should download an Rmd file into your local instance (whatever current working directory you're in in terminal)

11. Download a directory using `iget`

```
time iget -KPbvrf /iplant/home/username/NEON_Downloads/NEON_HARV_DP1.30003.
↪001_2019/
```

Here we're using the `time` flag to tell us how long the download takes

3.7.4 *Downloading with WebDav*

CyVerse Data Store also uses WebDav, an https based protocol for read-only data downloads from the Data Store.

We can use `wget` or `curl` commands in the terminal to download files this way.

12. Download a directory using `wget`

```
time wget -r -nH --cut-dirs=5 --no-parent -l8 --reject="index.html*"␣
↪https://data.cyverse.org/dav-anon/iplant/home/username/NEON_
↪Downloads/NEON_HARV_DP1.30003.001_2017/
```

again, we're using the `time` function to monitor the download speeds.

We're also using some `wget` flags to just get the data and folders back from the Data Store.

3.7.5 *Other Services: Downloading with S3*

Many organizations are hosting data on Amazon Web Services S3, Google Cloud Storage, or Microsoft Azure.

Cloud buckets, like S3, use HTTPS protocols, just like WebDav.

OpenTopography.org (re)hosts some NEON lidar data, e.g. NEON D17 Pacific Southwest- California

We can download these using their Point Cloud Bulk Data Download option:

```
aws s3 cp s3://pc-bulk/NEON_D17/ . --recursive --endpoint-url https://
→opentopography.s3.sdsc.edu --no-sign-request
```

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click ⬛ on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

 Learning Center Home

 Learning Center Home

3.8 Jupyter Lab, Desktop Environments, and Text Editors

Description:

Earlier in the workshop you launched a VICE image with RStudio. Follow the same process again using our Jupyter Lab Geospatial container instead.

We have integrated Project Jupyter's Data Science Jupyter Lab images with geospatial packages.

We have also integrated an all-in-one container which supports a desktop environment, Jupyter, VS Code (or your favorite text editors), and even RStudio. This container represents a 'Workspace' where you can do almost all of your daily data science tasks.

3.8.1 *Starting the VICE app*

1. Log into the Discovery Environment: https://de.cyverse.org

2. Click one of our quick launch buttons:

- Jupyter Lab Geospatial:

- Workspace:

Or, search using the ⊞ Apps 'Apps' search bar.

query "jupyter lab geospatial" or "workspace geospatial" and see what comes up.

Or, click the App button and a new window should open.

Under "My Apps" section click on the "My Communities" section and "NEON" Group should appear

3. In the App Window you have a few options.

i. `Analysis Name` - you can change the name of the analysis if you like, the default name is typically `<the-app-name>_analysis1`

> Your analysis will run, and when it completes, it will write any data that you have in the WORKING DIRECTORY of the container back to the Data Store in your Analyses folder, e.g. `/iplant/home/<username>/analyses/<the-app-name>_analysis1-<DATE-TIME-of-job-starting>`

ii. The second section is `Input Data` and has options for adding (1) a folder, (2) a single file, or (3) multiple files.

> For our use case today, we're going to add a folder from the data store.
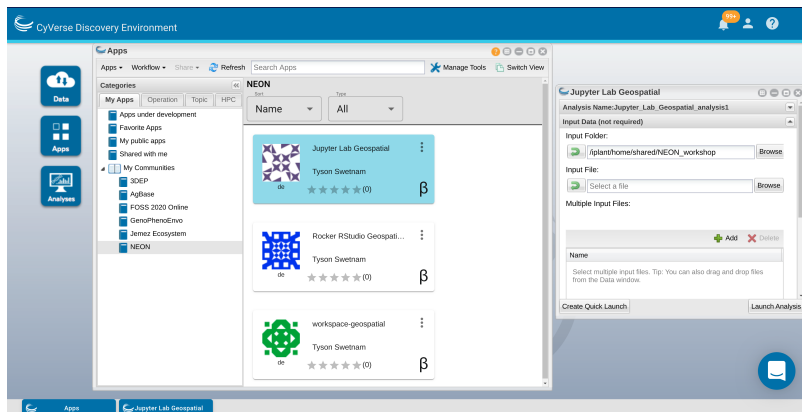
Add the path: `/iplant/home/shared/NEON_workshop/`
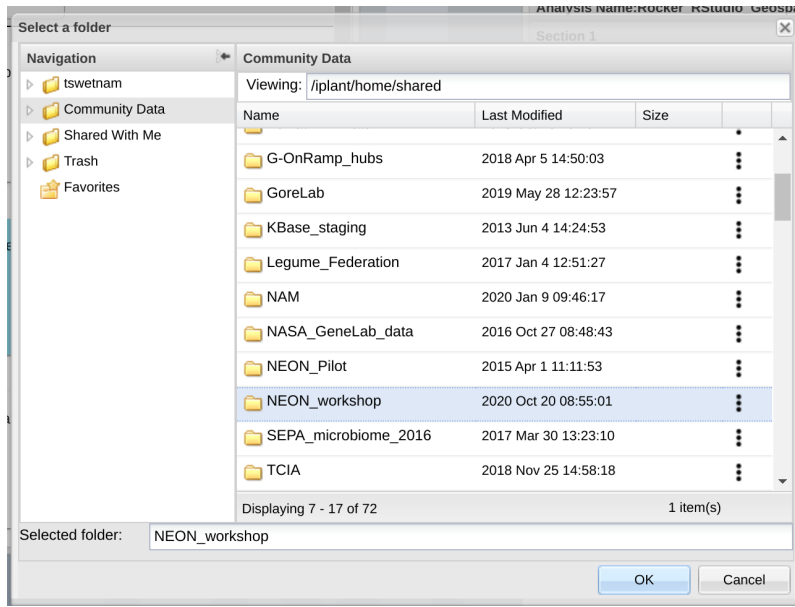
iii. The third section is `Resource Requirements`, again you can leave the default settings.

> You can set the `Minimum CPU` to the minimum number of cores your app requires. If you do not select anything, the app will still be able to use multiple cores on the shared node on which it is deployed.
>
> You can set the `Minimum RAM` to the minimum number of GB of RAM you think your app requires.
>
> You can set the `Minimum Disk Space` to the minimum amount of scratch space you think your data will need.
>
> **iv.** After you've set the analysis name, input data, and resource requirements, click `Launch Analysis`

Watch closely, you'll see a notification at the top of your screen and in the Bell icon in the upper right.

4. Open the icon labeled "Analyses" to view your running analysis.

> Look for your App Name. See the square icon with an arrow pointed up to the right? Click on that icon and a new tab will open.
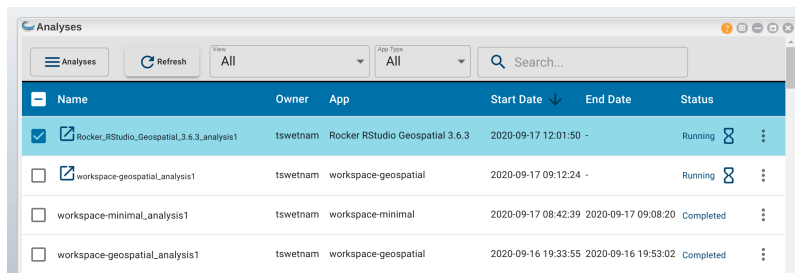
> Alternately, click the Bell icon in the upper right to see your notifications, you should see 'Access your running analysis here.' as an option. Click on that link and a new tab will open.

5. Having clicked on either of the hyperlinked icons in Step 4 should result in a new tab opening in your browser.

> Your Discovery Environment Browser tab should still be open.

The new URL for the instance has changed from `https://de.cyverse.org/de/` to something like `https://af7664685.cyverse.run/`.

This is a new instance is running a Virtual Machine for you on CyVerse cloud.

You can now begin working in your running instance.

Description of output and results

You should now be ready to run the

Note, there are pre-completed `.ipynb` notebooks in the `/tutorials` folder which you added to the Instance when it was created. These should be in the working directory under the `/NEON_workshop` directory. You can use these for guidance, particularly if you fall behind.

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click ![icon] on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

![home icon] Learning Center Home

Prerequisites

4.1 Downloads, access, and services

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

Prerequisite	Preparation/Notes	Link/Download
	You will need a CyVerse account to use our tools	
	CyberDuck File manager (Windows and Mac OS X only)	
	GitHub allows you to create your own version controlled repositories	
	Google Earth Engine (GEE) code editor account	

4.2 Platform(s)

We provide ready-to-use examples of (1) Docker containers with pre-configured geospatial software environments, (2) Notebook examples of NEON AOP geospatial data analyses, & (3) ReadTheDocs style documentation that will allow for self-paced asynchonous learning as well as opportunities for in-person live coding.

We will use the following CyVerse platform(s):

Platform	Interface	Link	Platform Tour
Data Store	GUI/Command line		
Discovery Environment	Web/Point-and-click		
Learning Center	ReadTheDocs	This website	

4.3 Application(s) used

Discovery Environment App(s):

App name	Version	Description	Quick Launch	GitHub repositories
Workspace	latest			
RStudio	3.6.3	Rocker Project RStudio with geospatial applications pre-installed		
Jupyter-Lab	2.2.5	Jupyter Lab Data Science Notebook with Geospatial applications pre-installed		
GIS Desktop	latest	Ubuntu Desktop with QGIS, GRASS-GIS, SAGA-GIS, PDAL, & GDAL tools		

4.4 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
	various	Use in browser or R Studio Shiny App	
Sample Datasets	various	Example datasets cached on the CyVerse Data Store	

Fix or improve this documentation

- Search for an answer:

- Ask us for help: click on the lower right-hand side of the page

- Report an issue or submit a change:

- Send feedback: learning@CyVerse.org

Learning Center Home