
CyVerse Documentation

CyVerse

Dec 20, 2021

GETTING STARTED

1	About	3
2	Prerequisites	5
3	Applications	7
4	Cloud Shell	11
5	JupyterLab	17
6	RStudio	25
7	Sharing VICE apps with collaborators	33
8	Workflow	39
9	Guidelines for adding interactive tools in DE	41
10	Building DE tools and apps	43
11	Featured Apps	45
12	Instant Launches	47
13	Quick Launches	49
14	Frequently Asked Questions	51
15	VICE Best practices	53
16	Tool Troubleshooting	55
17	Contact	59



 [Learning Center Home](#)

The Visual and Interactive Computing Environment (VICE) is the latest feature in CyVerse's Discovery Environment (DE) for running interactive applications.

User Guide instructs users on basic functions of the DE: (a) how to launch an interactive app, (b) bring your data into the container, (c) run an analysis, and (d) save or upload your results to the cloud.

Developer Guide gives advanced tips about how to (a) build your own VICE-ready containers, (b) host them on a public registry, and (c) integrate them into VICE and make them available to your user community.

Last, we provide a brief list of featured VICE apps in the DE.



 [Learning Center Home](#)

ABOUT

VICE stands for **V**isual **I**nteractive **C**omputing **E**nvironment and is a part of CyVerse **D**iscovery **E**nvironment (**DE**). **VICE** allows verified CyVerse users to launch customized containers of their favorite data science platforms.

1.1 What is the big idea?

The DE supports executable applications which run as workflows on high performance or high throughput computing environments. **VICE** introduces Graphic User Interfaces (GUI) and common Integrated Development Environments (IDE) such as [Project Jupyter](#), [RStudio](#), & Remote Desktops to the DE.

Note: How is VICE different from other DE apps?

Applications or “Apps” in the DE are classified as:

- **executable** - meaning the user selects parameters and data for a particular analysis, and submits the job for execution ([HTCondor](#) , [Tapis](#), [OSG](#)).
 - **HPC** - executable apps which run on [Texas Advanced Computing Center Tapis](#) as high performance computing (HPC) jobs.
 - **OSG** - executable apps which run on the [OpenScienceGrid](#) as high throughput (HTC) jobs.
 - **VICE** - interactive, visual computing environments with GUI or IDE.
-



 [Learning Center Home](#)

Verified User Information

To access the interactive apps in the Discovery Environment, we ask users to provide sufficient information about themselves in their user profiles to ensure proper use of the platform.

We ask users to provide a valid email address from an organization, i.e., *.org*, an educational institution with the *.edu* ending, or a government *.gov* – we will not grant access to commercial email addresses, e.g., *@gmail.com* *@yahoo.com* *@msn.com* etc.

We also ask that users provide their **ORCID** – ORCID is the most widely adopted digital identifier for research in the world.

PREREQUISITES

Prerequisite	Preparation/Notes	Link/Download
CyVerse account	You will need a CyVerse account to complete this exercise	Register
Discovery Environment access	You will get access to Discovery Environment by default. Check the 'My Services' tab; if Discovery Environment is not listed, click the 'Available' tab, locate Discovery Environment and click the link to request access.	CyVerse User Portal

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org



APPLICATIONS

Currently, VICE apps are categorized as:

1. Integrated Development Environments (IDE) (e.g.,)
2. Interactive Apps (e.g., Shiny, WebGL, HTML5)
3. Virtual Desktop Environment (e.g., Ubuntu Desktops w/ Apache Guacamole, VNC, Xpra)

3.1 1. What is JupyterLab?

[JupyterLab](#) is an interactive development environment for working with notebooks, code and data.

JupyterLab enables you to use text editors, terminals, file viewers, and other custom components side-by-side with notebooks in a tabbed work area. JupyterLab provides a high level of integration between notebooks, documents, and activities:

- Drag-and-drop to reorder notebook cells and copy them between notebooks
- Run code blocks interactively from text files (.py, .R, .md, .tex, etc.)
- Link a code console to a notebook kernel to explore code interactively without cluttering up the notebook with temporary scratch work
- Edit popular file formats with live preview, such as Markdown, JSON, CSV, Vega, VegaLite, and more

3.1.1 1.1 What is a Jupyter Notebook?

The [Notebook](#) (formerly IPython Notebook) is Project Jupyter’s flagship project for creating reproducible computational narratives.

It enables users to create and share documents that combine live code with narrative text, mathematical equations, visualizations, interactive controls, and other rich output.

Notebook documents (or “notebooks”) are documents produced by the Jupyter Notebook App, which contains both computer code (e.g., python, r, julia) and rich text elements (paragraph, equations, figures, comments, images, links, etc.).

3.1.2 1.2 JupyterLab VICE

The apps are based on official integrated into the CyVerse Discovery Environment (DE).

3.2 2. What is RStudio?

RStudio is a free and open source integrated development environment for R, a programming language for statistical computing and graphics. Some of its features include:

- Customizable workbench with all of the tools required to work with R in one place (console, source, plots, workspace, help, history, etc.)
- Syntax highlighting editor with code completion
- Execute code directly from the source editor (line, selection, or file)
- Full support for authoring Sweave and TeX documents
- Runs on all major platforms (Windows, Mac, and Linux) and can also be run as a server, enabling multiple users to access the RStudio IDE using a web browser

3.2.1 2.1 RStudio VICE

The apps are based on official .

3.3 3. What is Shiny?

Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge. Some of its features include:

- Build useful web applications with only a few lines of code—no JavaScript required
- Shiny applications are automatically “live” in the same way that spreadsheets are live. Outputs change instantly as users modify inputs, without requiring you to reload your browser
- Shiny user interfaces can be built entirely using R, or can be written directly in HTML, CSS, and JavaScript for more flexibility
- Works in any R environment (Console R, Rgui for Windows or Mac, ESS, StatET, RStudio, etc.)
- Attractive default UI theme based on Twitter Bootstrap
- A highly customizable slider widget with built-in support for animation
- Pre-built output widgets for displaying plots, tables, and printed output of R objects
- Fast bidirectional communication between the web browser and R using the websockets package
- Uses a reactive programming model that eliminates messy event handling code, so you can focus on the code that really matters
- Develop and redistribute your own Shiny widgets that other developers can easily drop into their own applications

3.3.1 3.1 Shiny VICE

The apps are based on official .

Shiny Apps require the contents of a Shiny App be added as an input folder prior to launching a VICE app - otherwise the default Shiny App demo will be the only contents in the container.

Customized Shiny Apps can be created and deployed using the Tool and App builder.

3.4 4. What is Ubuntu Desktop?

The Ubuntu Desktop has a full Guacamole installation and Ubuntu XFCE desktop. This allows users to have a simple all-in-one desktop through their web browser. Users can run any interactive or visualization tool that can run on the most recent linux distros. Solutions to support the inevitable array of linux applications that user will want. Potential options include:

- Separate image per application
- Network fs (e.g. NFS, Ceph, etc) containing all applications
- Per-application network fs
- On-demand installation of application via script/ansible

3.4.1 4.1 Ubuntu Desktop VICE

Linux Desktops using Apache Guacamole, and Xpra have been integrated into the DE.

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org

 [Learning Center Home](#)

CLOUD SHELL

4.1 1. Log into Discovery Environment

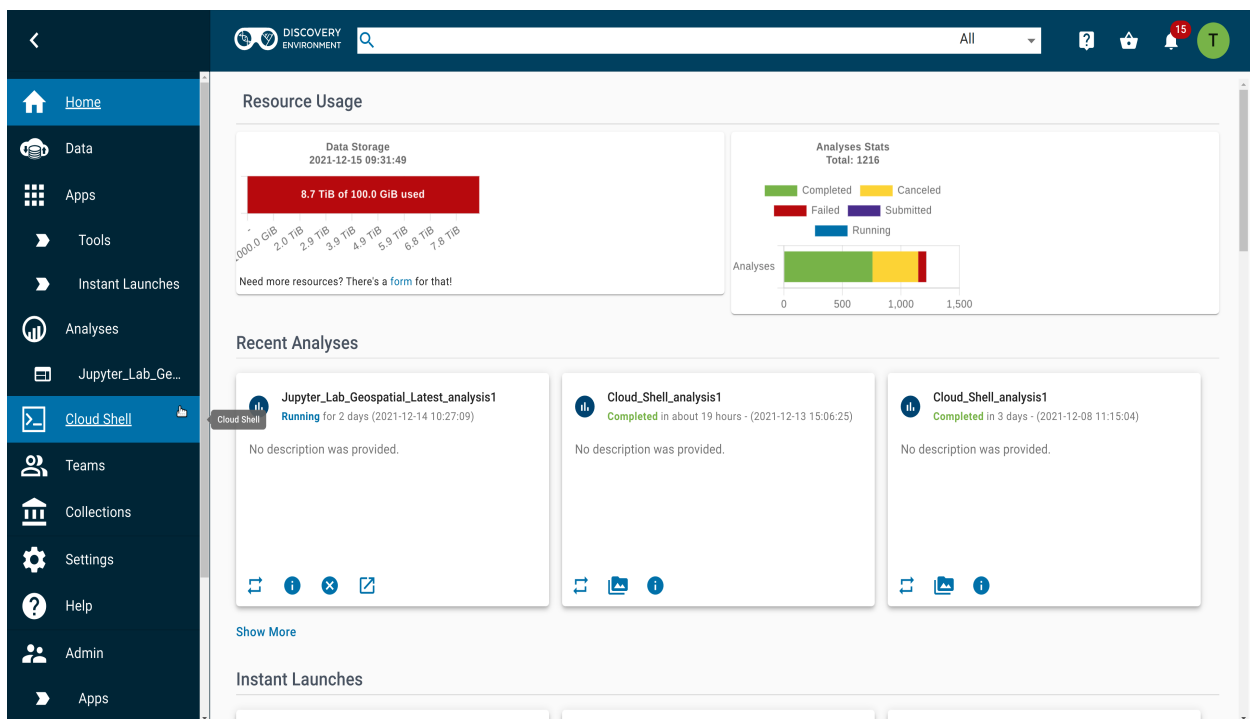
Log into [CyVerse DE](#) with your CyVerse credentials.

If you have not yet created an account, go to the [User Portal](#) and do so.

The Cloud Shell App is already pinned to the Discovery Environment Tool Bar (left side of screen):

Alternately, use the search to specify 'Apps' then type keyword *Shell* or *Cloud Shell*

You can see Cloud Shell listed when you expand the three bars Menu icon.



4.2 2. Launch analysis

Instant Launches start the App immediately without the opportunity to add any input data or increase the VM size.

Clicking on the **Apps** will show the Cloud Shell in “Featured Apps”.

By default the Analysis Name is the name of the App plus the date and time.

The output folder will use this same name and be written to the `/iplant/home/username/analyses/` directory **AFTER** the interactive analysis is completed.

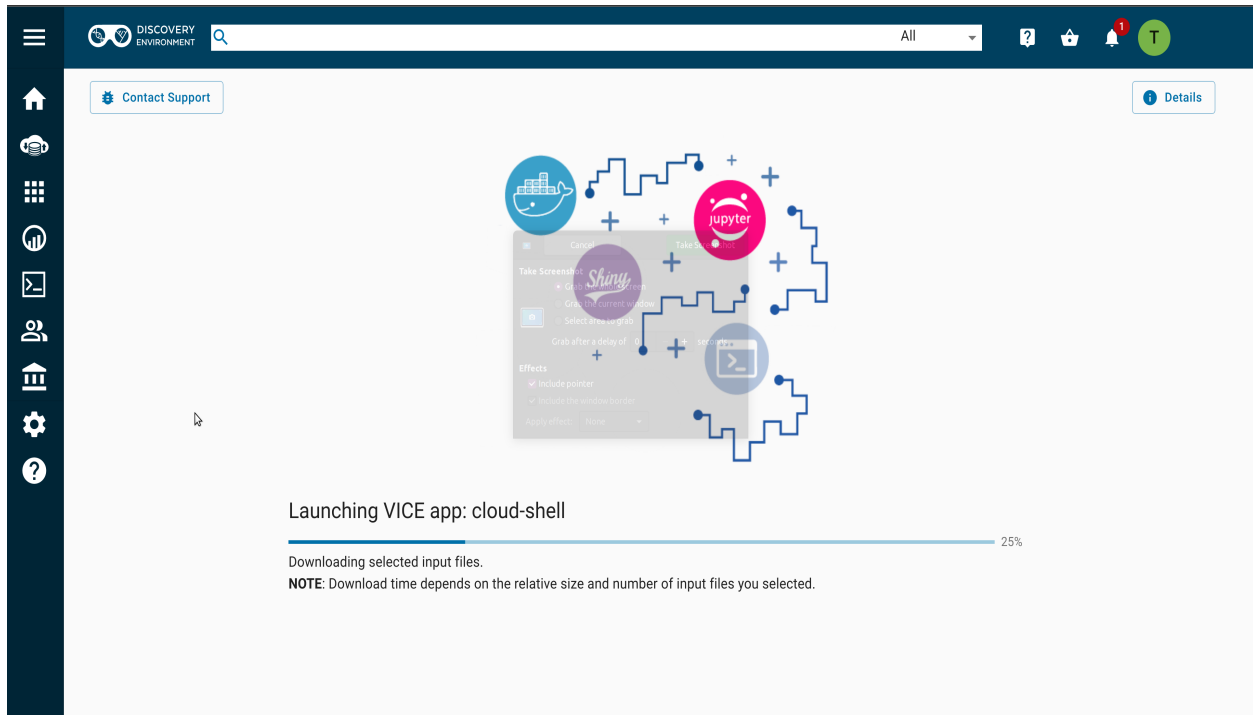
Note: Deleting a running Analysis will stop it from writing outputs to your data store `analyses/` directory.

Note: The input files and/or folders can be selected under the ‘Parameters’ tab.

Tip: If you have an existing set of files or directories, you can import these into the app using input files and/or folder in the Parameters .

4.3 3. Navigate to the App

After you have started a VICE App, your browser will open a new tab and automatically be taken to the loading screen.



Once the app is ready, it will transition to the user interface (in this example, a Linux terminal)

There should be a “message of the day”, some information about the machine you’re using, and your CyVerse username for when you initiate an iCommands connection.

Important: Normal wait times for a featured VICE App are less than 2 minutes. If you’re experiencing a significantly longer wait, consider terminating the app and re-starting.

If you have closed the window, and are returning to the Discovery Environment in a new login session, you can find the running App in the “Analyses” section

```

CyVerse VICE

..-/+00ssss00+/-..
`:+ssssssssssssssssss+`
+sssssssssssssssssssyssss+
.osssssssssssssssssdmmnnyssso.
/sssssssssssshdmmnNmmymNMMMMhssss+/
+ssssssssshmmymhhyyyhmmNMMMMhssss+/
/ssssssssshmmymhhyyyhmmNMMMMhssss+/
.sssssssdmmNMMhssssssssshNMMMdssssss.
+ssshhhyNMMhssssssssssyNMMMyssssss+
ossyNMMNymMMhssssssssssshmmhssssssso
ossyNMMNymMMhssssssssssshmmhssssssso
+ssshhhyNMMhssssssssssyNMMMyssssss+
.sssssssdmmNMMhssssssssshNMMMdssssss.
/ssssssssshmmymhhyyyhmmNMMhssssss+/
+ssssssssdmyhmmNMMMMhdddyssssss+
/sssssssssshdmmNmmymNMMMMhssss+/
.osssssssssssssssssdmmnnyssso.
+sssssssssssssssssyssss+
`:+ssssssssssssssss+`
..-/+00ssss00+/-..

user@ae1ee4fbb
-----
OS: Ubuntu 20.04.3 LTS x86_64
Host: X9DR3-F 0123456789
Kernel: 3.10.0-1160.49.1.el7.x86_64
Uptime: 9 days, 50 mins
Packages: 532 (dpkg)
Shell: bash 5.0.17
Resolution: 1024x768
CPU: Intel Xeon E5-2620 v2 (24) @ 2.600GHz
Memory: 4576MiB / 257930MiB

Welcome to the Cloud Shell.

Your CyVerse Username: tswetnam

To connect iCommands, type 'iinit'

Want to know more? Learning Center Documentation: https://learning.cyverse.org

About this Tool & App: https://github.com/cyverse-vice/cli

user@ae1ee4fbb:~/work$

```

4.4 4. Activate a conda environment

The Cloud Shell comes with multiple languages and package managers pre-installed. These include `go`, `python`, and `rust`.

Package managers include `conda` and `cargo` in addition to linux `apt` and `apt-get`.

Your identity inside the container is `user` and you have limited `sudo` privileges to install new packages into the container.

These changes are not saved after the analyses ends, or when you start a new Cloud Shell Analyses later.

Tip: The cloud shell is running a terminal multiplexer called *tmux* which keeps your session active even after you’ve closed your browser tab.

tmux keyboard commands should function normally.

To activate conda:

```
` conda init `
```

and then

```
` conda activate base `
```

If you receive a message about refreshing your screen, you can `exit` the cloud shell by typing “exit” and then hitting the refresh button on your browser tab.

4.5 5. Adding data to your analysis

To connect to the CyVerse DataStore, you can initiate an iRODS iCommands `iinit`

You should now be connected to your `/iplant/home/username` home directory.

4.5.1 ils

or

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

To view the 'shared' directory try:

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

Download data into your Cloud Shell with iCommands by running `iget`

```
` iget -KPbvrf /iplant/home/shared/cyverse_training/ `
```

```
***
```

4.6 6. Complete and Save Outputs

After finishing your analyses, you can save the outputs back to Data Store.

You can either use `iput` to copy your new files back to your user space, or if you've left your new work in the `/home/user/work` folder, it will be copied back to your `/iplant/home/username/Analyses/` directory.

You can find the outputs you generated (if any) using the same steps as before, but this time select the 'Go To Output Folder'.

4.7 7. Terminate your app

The Discovery Environment is a shared system. In fairness to the community, we ask users to "Terminate" any apps they have started when they are no longer running analyses.

In the Analyses window, select the app with the checkbox, then select "More Actions" and "Terminate" to shut the app down.

Any new data in the `/home/user/work` directory will begin copying back to your folder at this time.

Any input data which you added when the app started using the conventional launch feature will not be copied.

Warning: VICE apps only run for a pre-determined amount of time, typically between 4 and 48 hours. If you have opted for email notifications from DE, then you'll get a notification 1 day before and 1 hour before the app gets terminated. If you want to extend the time, you need to login to <https://de.cyverse.org>, find your analysis and then click the hour glass which automatically extends the app run time.

Fix or improve this documentation

- On Github:
- Send feedback: Tutorials@CyVerse.org



JUPYTERLAB

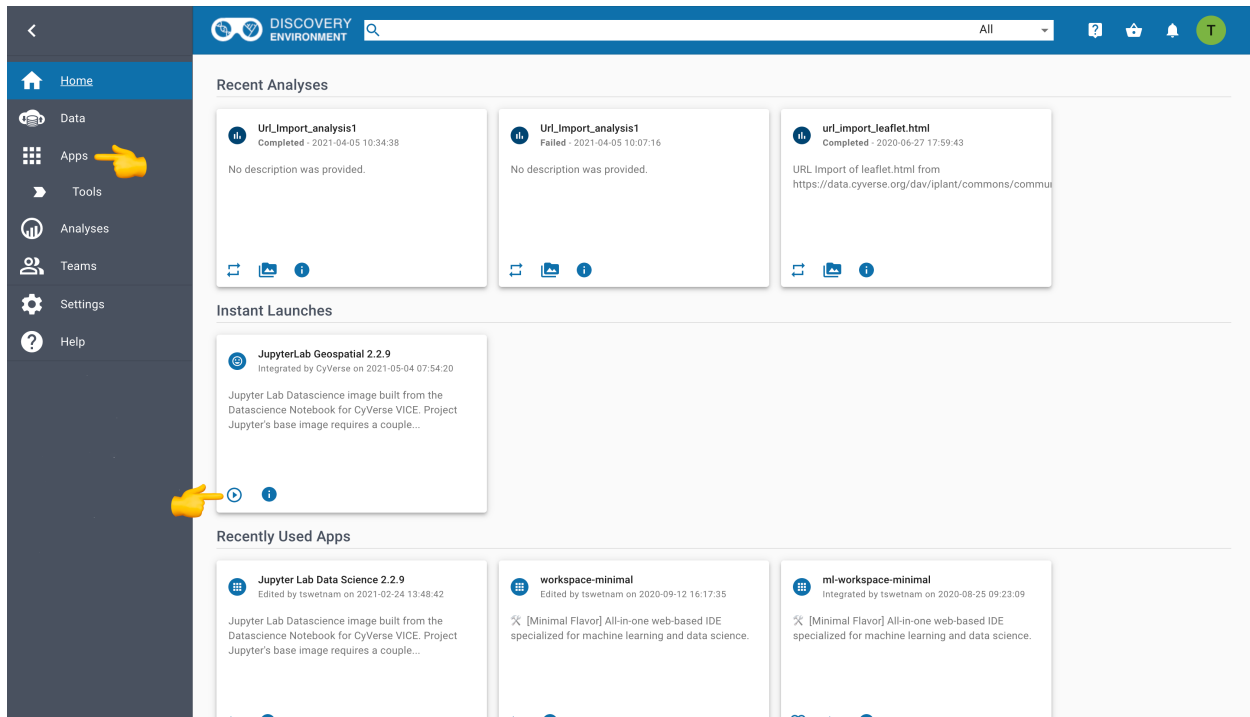
5.1 1. Search for JupyterLab

Log into CyVerse DE

Use the search bar to specify ‘Apps’ or type keywords *Jupyter* or *JupyterLab Datascience*

The screenshot shows the CyVerse DE Discovery Environment interface. A search bar at the top contains the text 'JupyterLab Datascience'. The left sidebar shows navigation options: Home, Data, Apps (selected), Tools, Analyses, Teams, Settings, and Help. The main content area displays search results for 'JupyterLab Datascience' under the 'Apps' tab. The results are organized into sections: 'Browse All Apps', 'Apps', 'Analyses', and 'Data'. The 'Apps' section shows a list of applications, including 'JupyterLab Datascience 2.2.9', '16sblaster', '3.1 dev GEA Le', '3.5.2-Shinyvers', '3.5.3-Shinyvers', '3.6.0-ShinyVers', 'AAARFv1.0.1', 'add column to file', 'add column to file2', 'Admixture', 'ALLMAPS', 'ALLMAPS merge', 'ALLMAPS path', 'Annotate transcripts', 'AntEpiSeeker 2.0', 'appendUnclustered', 'APPLES_conservation', and 'APPLES_rbh'. Each application entry includes a checkbox, a name, a description, a user name, a star rating, and a system ID. The 'Data' section shows a list of data items, including 'AAARFv1.0.1', 'add column to file', 'add column to file2', 'Admixture', 'ALLMAPS', 'ALLMAPS merge', 'ALLMAPS path', 'Annotate transcripts', 'AntEpiSeeker 2.0', 'appendUnclustered', 'APPLES_conservation', and 'APPLES_rbh'. Each data item entry includes a checkbox, a name, a description, a user name, a star rating, and a system ID. The bottom of the interface shows a pagination bar with the number 100.

5.2 2. Launch analysis



Instant Launches will start the App immediately.

Clicking on **Apps** find the JupyterLab you're interested in and then click on the blue run button.

By default the Analysis Name is the name of the App, and the Output Folder is where any work done in the container will be written to `/iplant/home/username/analyses/analysis_name/` directory **AFTER** the interactive analysis is completed.

Note: Deleting a running Analysis will stop it from writing outputs to your data store `analyses/` directory.

The screenshot displays the 'JupyterLab Datascience 2.2.9' configuration page. The left sidebar contains navigation options: Home, Data, Apps, Tools, Analyses, Teams, Settings (highlighted with a yellow hand icon), Help, Admin, Apps, DOI Requests, Reference Genomes, Tools, and VICE. The main content area shows a progress bar with four steps: 1. Analysis Info, 2. Parameters, 3. Advanced Settings (optional), and 4. Review and Launch. Step 1 is active. It includes a 'Back' button, a 'Next' button, and a 'Details' button. The 'Analysis Name' field is filled with 'JupyterLab_Datascience_2.2.9_analysis1'. The 'Output Folder' field is filled with '/iplant/home/tswetnam/analyses'. A yellow hand icon points to the 'Next' button at the bottom.

Note: The input files and/or folders can be selected under the 'Parameters' tab.

Tip: If you have an existing Jupyter workbook, you can import it into the app using input files and/or folder in the Parameters .

5.3 3. Navigate to JupyterLab url

After you start the VICE App, you will be taken to a loading screen.

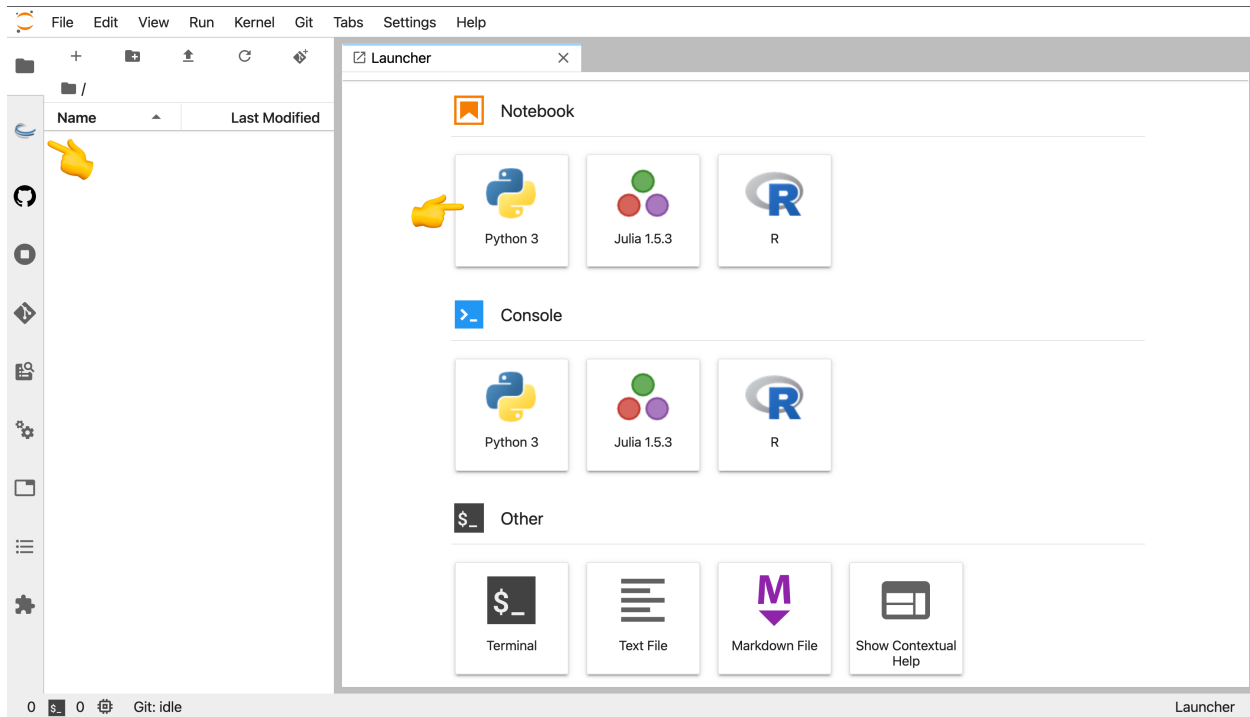
Once the app is ready, it will transition to the Jupyter Lab interface

Important: Normal wait times for a featured VICE App are less than 2 minutes. If you're experiencing a significantly longer wait, consider terminating the app and re-starting.

The Jupyter Lab Interface: Jupyter Lab provides flexible building blocks for interactive, exploratory computing. While Jupyter Lab has many features found in traditional integrated development environments (IDEs), it remains focused on interactive, exploratory computing. The Jupyter Lab interface consists of a main work area containing tabs of documents and activities, a collapsible left sidebar, and a menu bar. The left sidebar contains a file browser, the list of running kernels and terminals, the command palette, the notebook cell tools inspector, and the tabs list.

More information about the Jupyter Lab can be found [here](#).

5.4 4. Create Jupyter notebook



Jupyter notebooks (`.ipynb`) combine code with narrative text (Markdown), equations (LaTeX), images and interactive visualizations.

To create a notebook, click the + button, this opens the new Launcher tab.

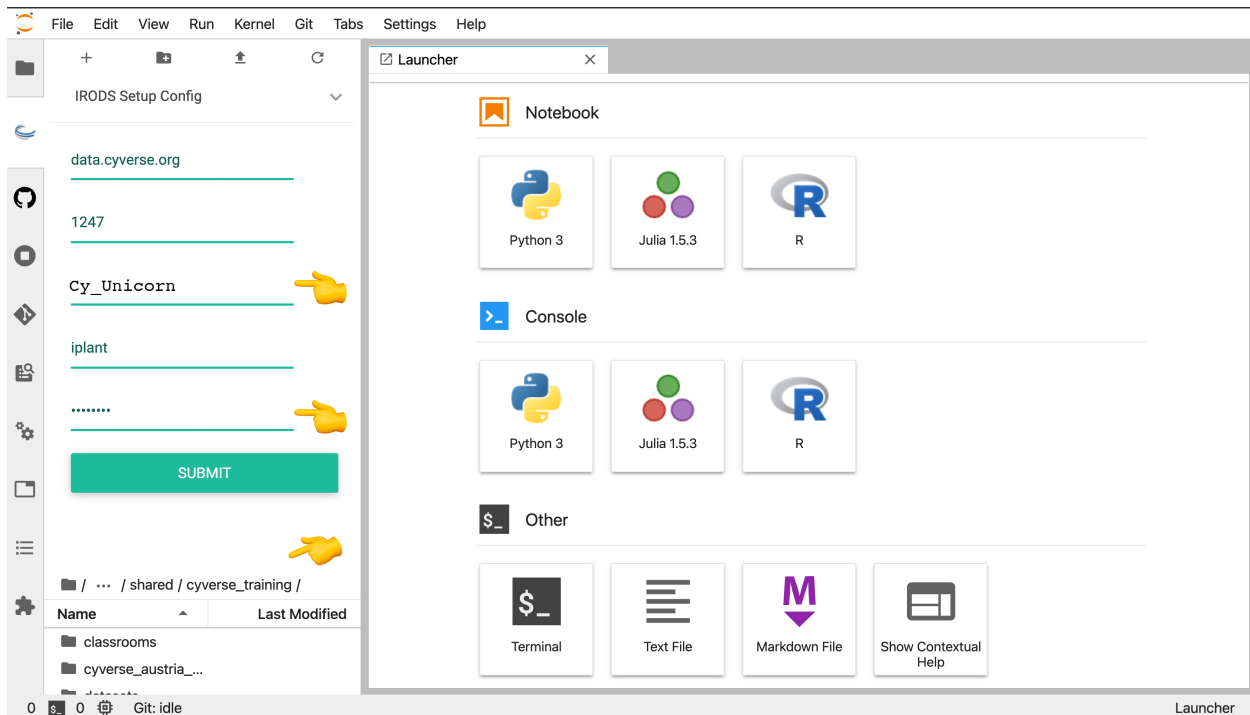
The JupyterLab Datascience containers have three pre-installed kernels: Python3, Julia, and R.

[Official Jupyter Notebooks](#)

Tip: To open the classic Notebook view from JupyterLab, select “Launch Classic Notebook” from the Help Menu.

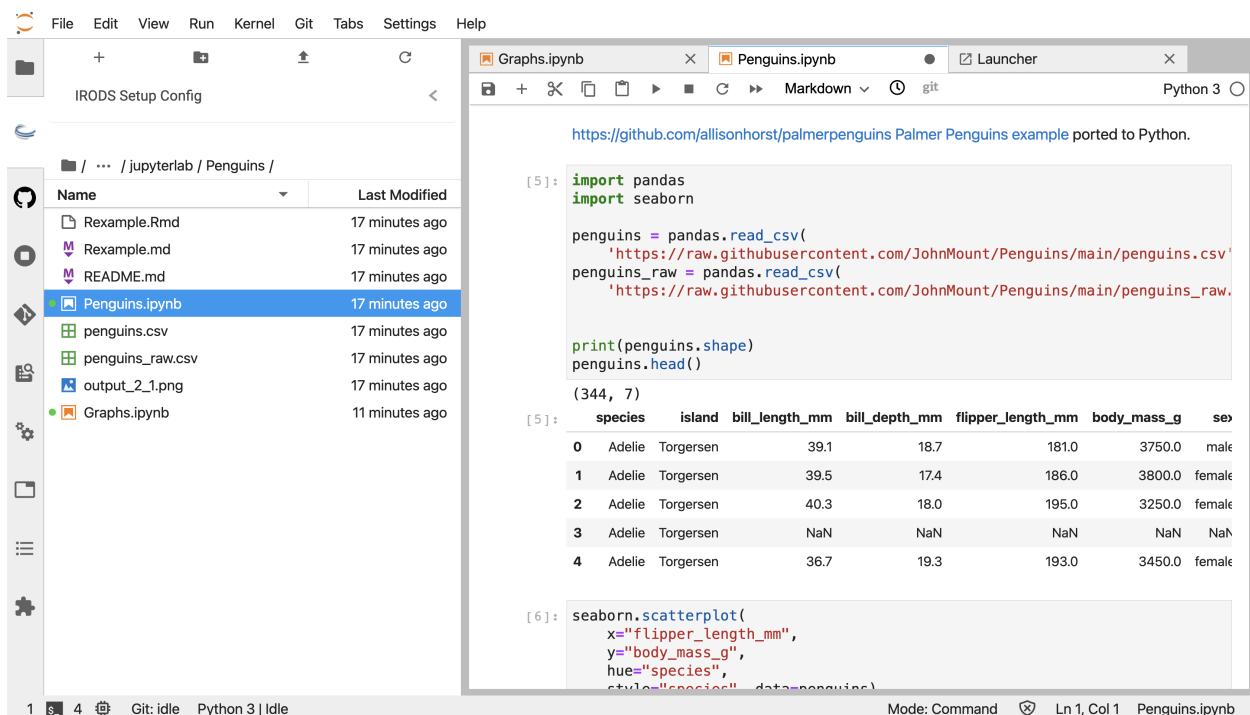
5.5 5. Adding data to your analysis

To connect to the CyVerse DataStore, click the little CyVerse orb in the left side of the Lab.



You should now be connected to your `/iplant/home/username` home directory. Navigate to the 'shared' directory by clicking one order higher on the `/home` directory, you should now see your username and the `/shared` path.

Navigate to `/shared/cyverse_training/platform_guides/discovery_environment/jupyterlab` and open the **Penguins** sample dataset

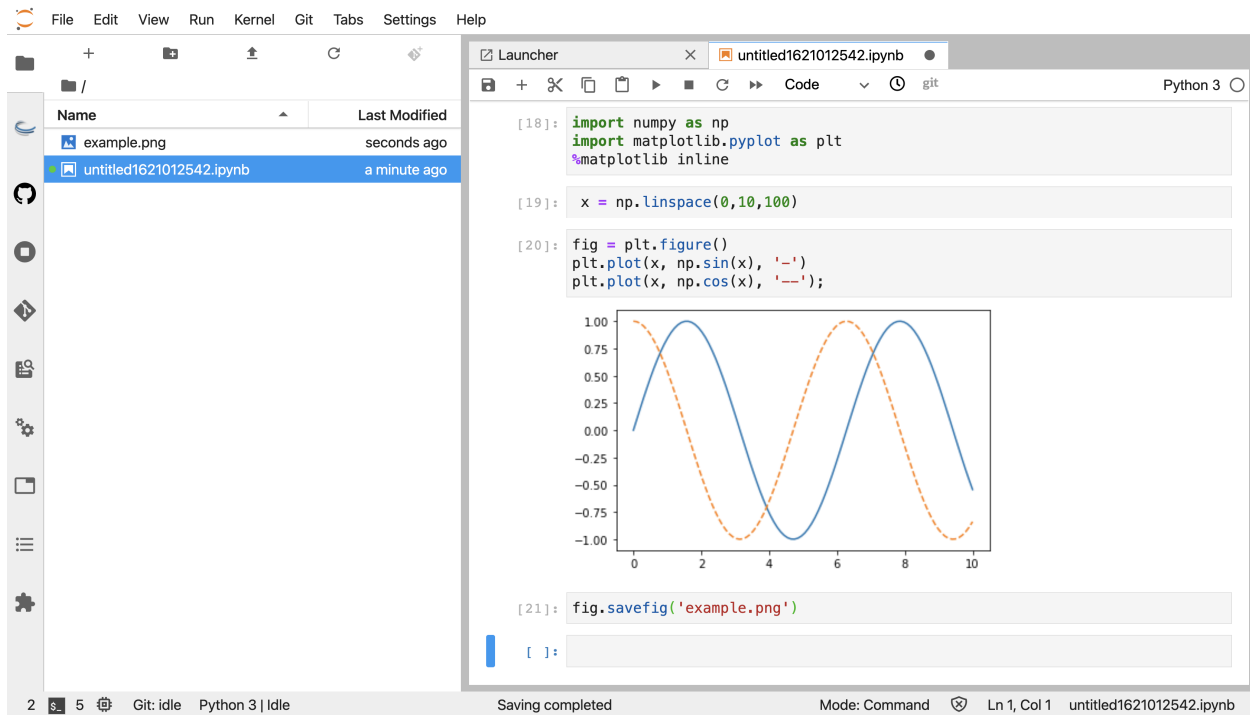


Note: There are plenty other cool stuff that you can do in Jupyter Lab such as using **consoles**, using **terminal** and

using [text editor](#).

5.6 6. Write your own code

Once you open a new notebook, you can start writing your code, put markdown text, generate plots, save plots etc.



The screenshot displays the JupyterLab environment. On the left is a file browser showing a directory with 'example.png' and 'untitled1621012542.ipynb'. The main area shows a notebook with the following code:

```
[18]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

[19]: x = np.linspace(0,10,100)

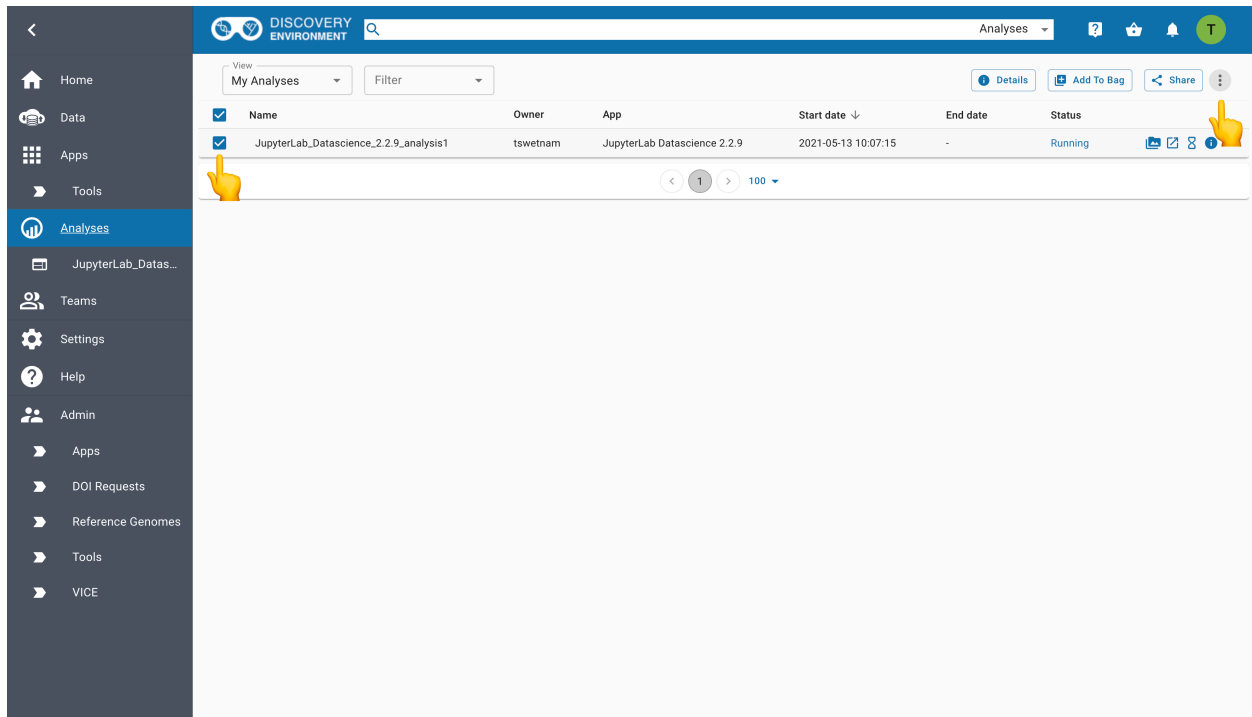
[20]: fig = plt.figure()
plt.plot(x, np.sin(x), '-')
plt.plot(x, np.cos(x), '-');

[21]: fig.savefig('example.png')
```

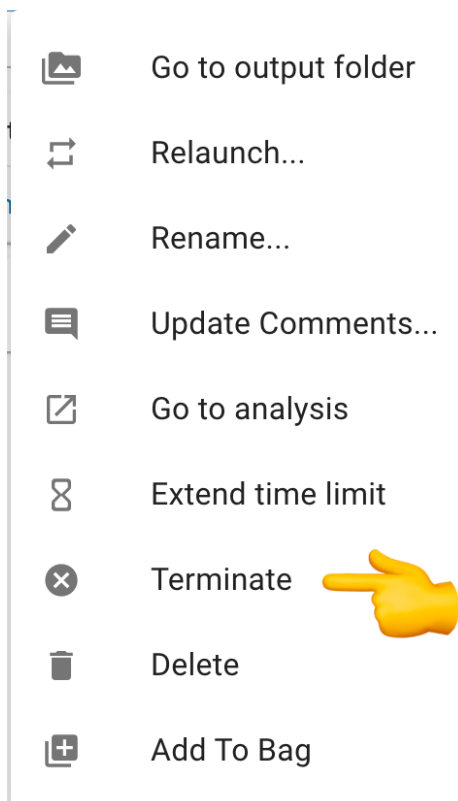
The code generates a plot of sine and cosine waves. The plot shows two periodic functions: a solid blue line for $\sin(x)$ and a dashed orange line for $\cos(x)$. The x-axis ranges from 0 to 10, and the y-axis ranges from -1.00 to 1.00. The plot is titled 'example.png'.

The status bar at the bottom indicates 'Saving completed', 'Mode: Command', and 'Ln 1, Col 1'.

5.7 6. Complete and Save Outputs



After finishing your analysis, you can save outputs to Data Store by clicking the Analysis window, then select the VICE analysis that you are running and select *Terminate* under the “Analyses” button.



After you had done this, you can find the outputs that you generated (if any) using the same steps as before, but this time selecting ‘Go To Output Folder’.

Warning: VICE apps only run for a pre-determined amount of time, typically between 4 and 48 hours. If you have opted for email notifications from DE, then you’ll get a notification 1 day before and 1 hour before the app gets terminated. If you want to extend the time, you need to login to <https://de.cyverse.org>, find your analysis and then click the hour glass which automatically extends the app run time.

Fix or improve this documentation

- On Github:
- Send feedback: Tutorials@CyVerse.org



Learning Center Home

RSTUDIO

RStudio is a free, open source IDE (integrated development environment) for R. Its interface is organized so that the user can clearly view graphs, data table, R code and output at the same time. It also offers an Import-Wizard-like feature that allows users to import CSV, Excel, SAS (*.sas7bdat*), SPSS (*.sav*), and Stata (**.dta*) files into R without having to write the code to do so.

6.1 1. Search for Rstudio App

First log-on CyVerse DE

Use the search bar to specify 'Apps' or search for 'rstudio'.

The screenshot shows the CyVerse DE interface. On the left is a dark sidebar with navigation icons and labels: Home, Data, Apps (highlighted with an orange arrow), Tools, Analyses, Rocker_RStudio_V..., Teams, Settings, and Help. The top header is blue with the 'DISCOVERY ENVIRONMENT' logo, a search bar containing 'Rstudio' (with an orange arrow pointing to it), and a dropdown menu set to 'All'. Below the header, a message states 'Searching for "Rstudio". Found 67 matching results.' There are four tabs: Data (30), Apps (28, selected), Analyses (9), and Teams (0). The main content area displays a table of RStudio applications.

Name ↑	Integrated By	System ID
Rocker RStudio Geospatial 3.6.3	Tyson Swetnam	de
Rocker RStudio Geospatial v3.5.3	Tyson Swetnam	de
Rocker RStudio TidyVerse v3.6.3	Tyson Swetnam	de
Rocker RStudio Verse 3.6.3	Tyson Swetnam	de
Rocker RStudio Verse 4.0.0-ubuntu18.04	Tyson Swetnam	de
Rocker RStudio Verse 4.0.5	Tyson Swetnam	de

At the bottom of the table is a 'Load More' button.

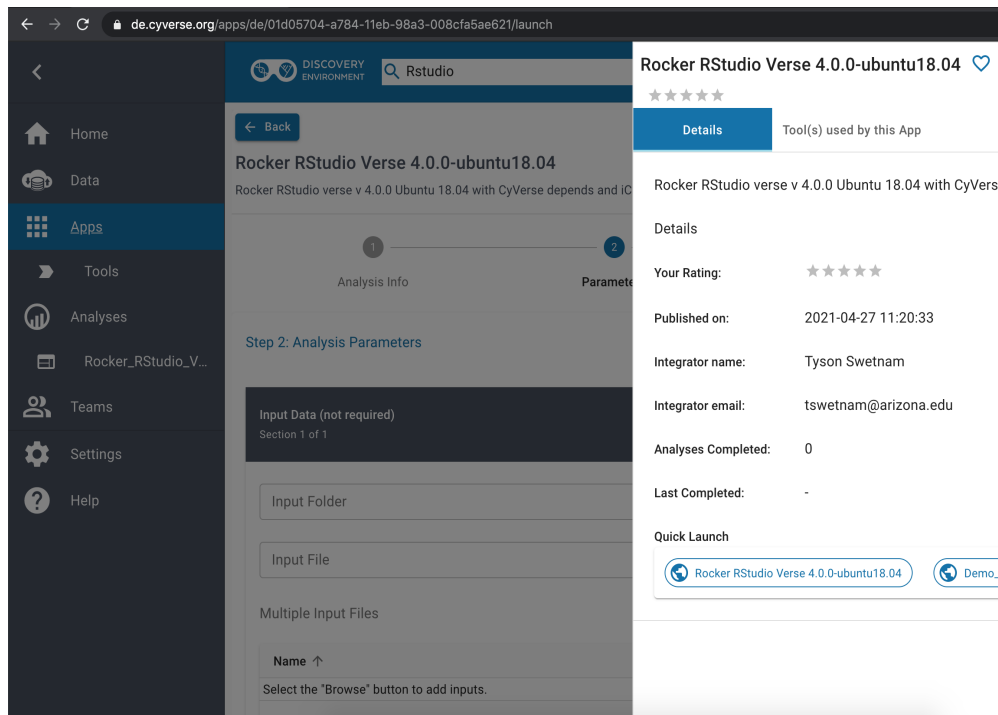
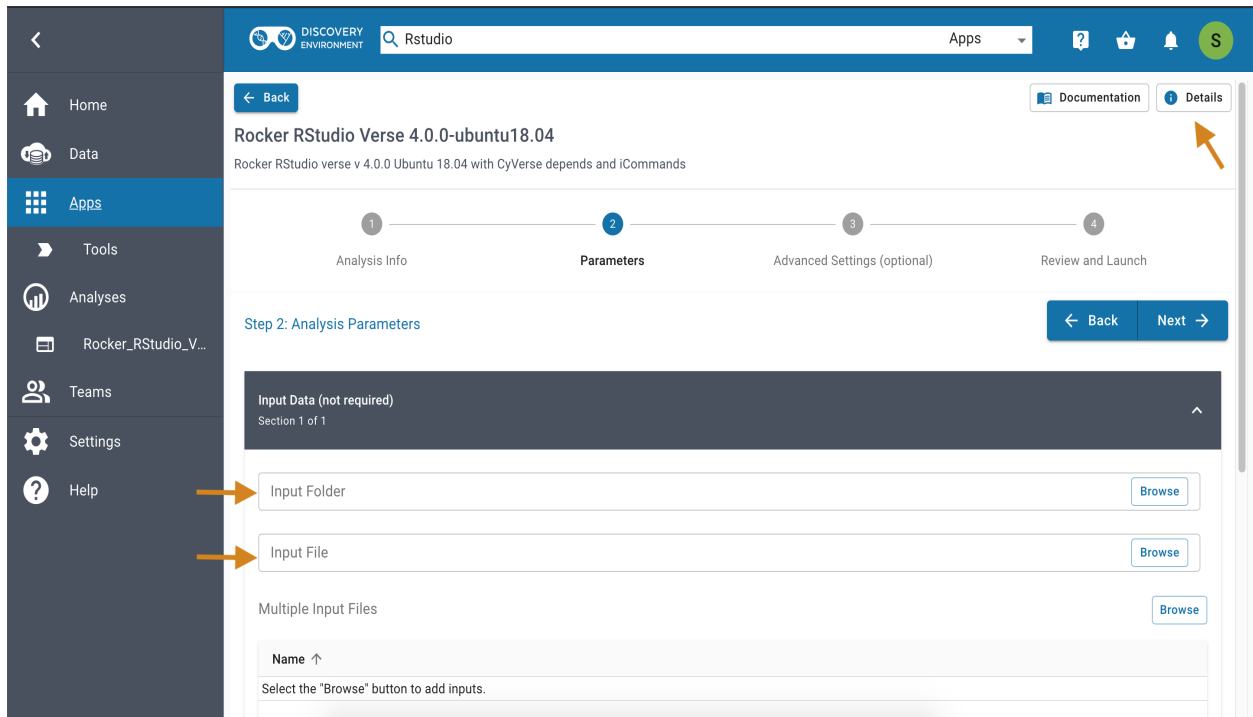
6.2 2. Launch analysis

Find the RStudio app you are interested in and click on the app. Change the analysis name and output folder or leave it to defaults.

The screenshot displays the 'Rocker RStudio Verse 4.0.0-ubuntu18.04' app configuration page in the CyVerse Discovery Environment. The interface includes a sidebar with navigation options: Home, Data, Apps (selected), Tools, Analyses, Rocker_RStudio_V..., Teams, Settings, and Help. The main content area shows a progress bar with four steps: 1. Analysis Info, 2. Parameters, 3. Advanced Settings (optional), and 4. Review and Launch. The 'Analysis Info' step is active, showing fields for 'Analysis Name' (pre-filled with 'Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1'), 'Enter analysis name', 'Comments', and 'Output Folder' (pre-filled with '/iplant/home/siuser/analyses'). There are 'Back' and 'Next' buttons at the bottom of the form.

Launch the Rstudio app by selecting an example folder or click on the details to quick launch example data. You can select different input files and/or folder.

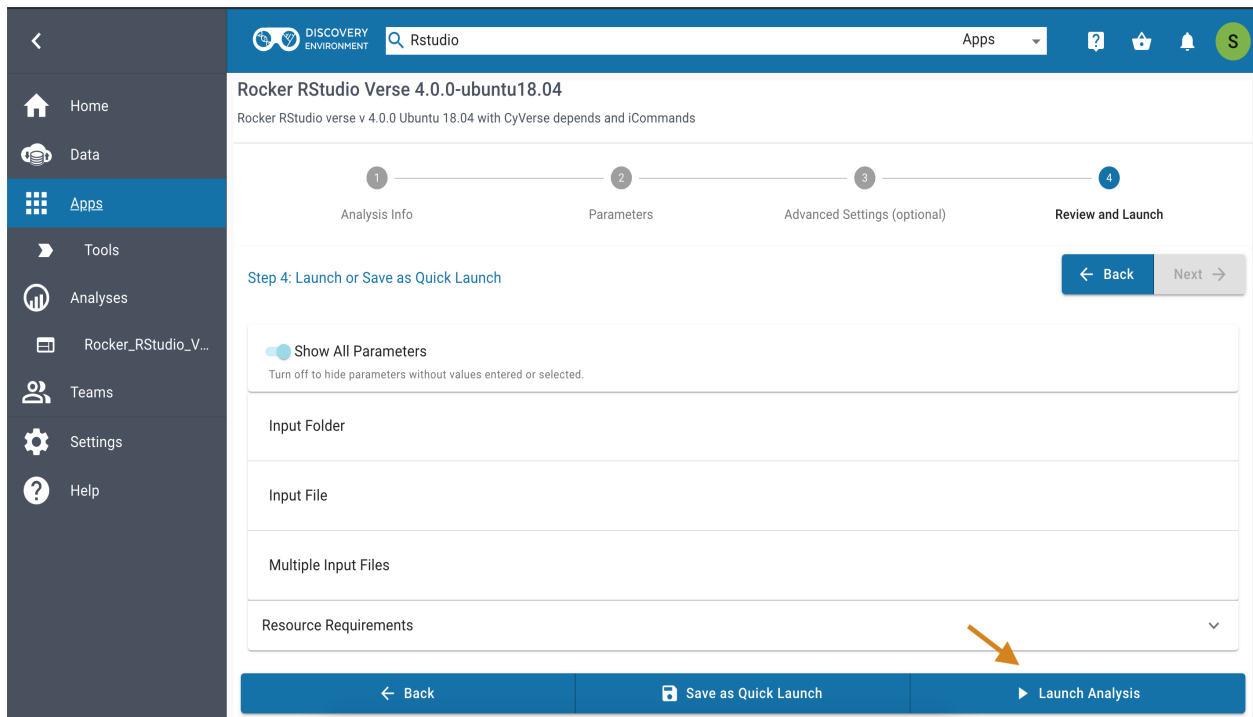
Tip: You can use input files to import a script into the app.



Use quick launch to input example data. . .

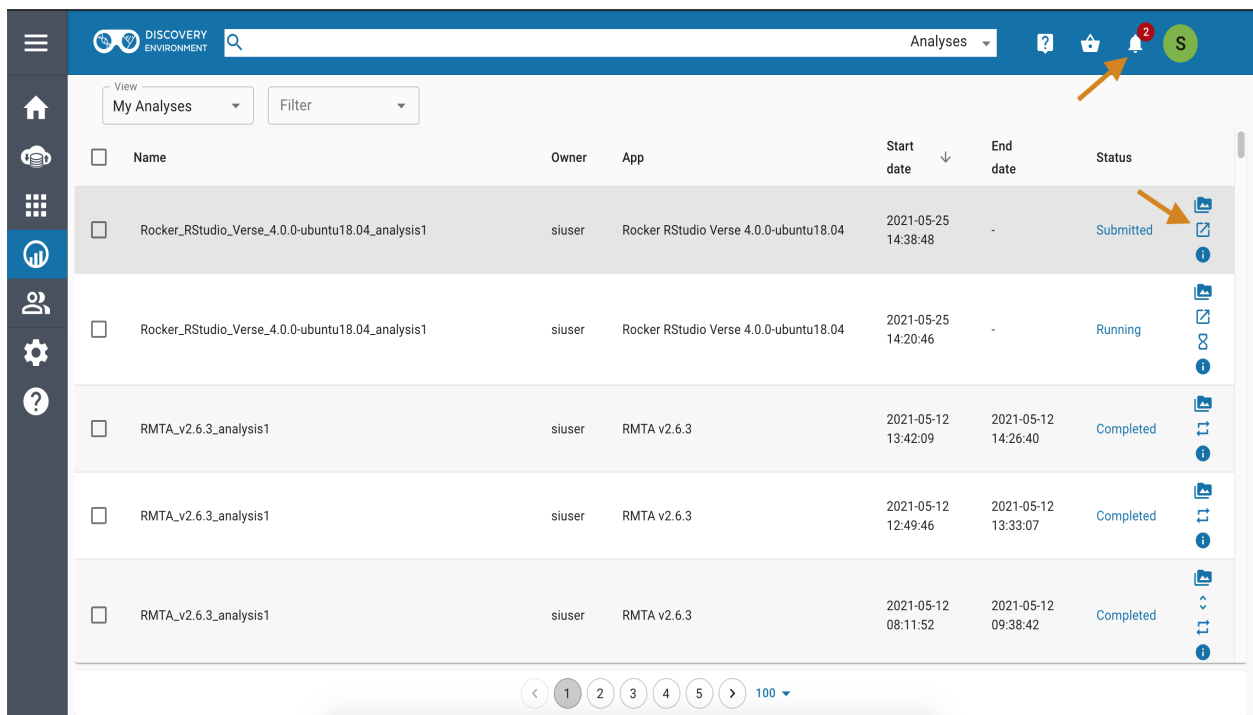
Note: You will not see any files when selecting the folder if you selected input by folder. Rest assured that they will be there once the app begins to run.

Launch the analysis after you are finished selecting the input files (if any).



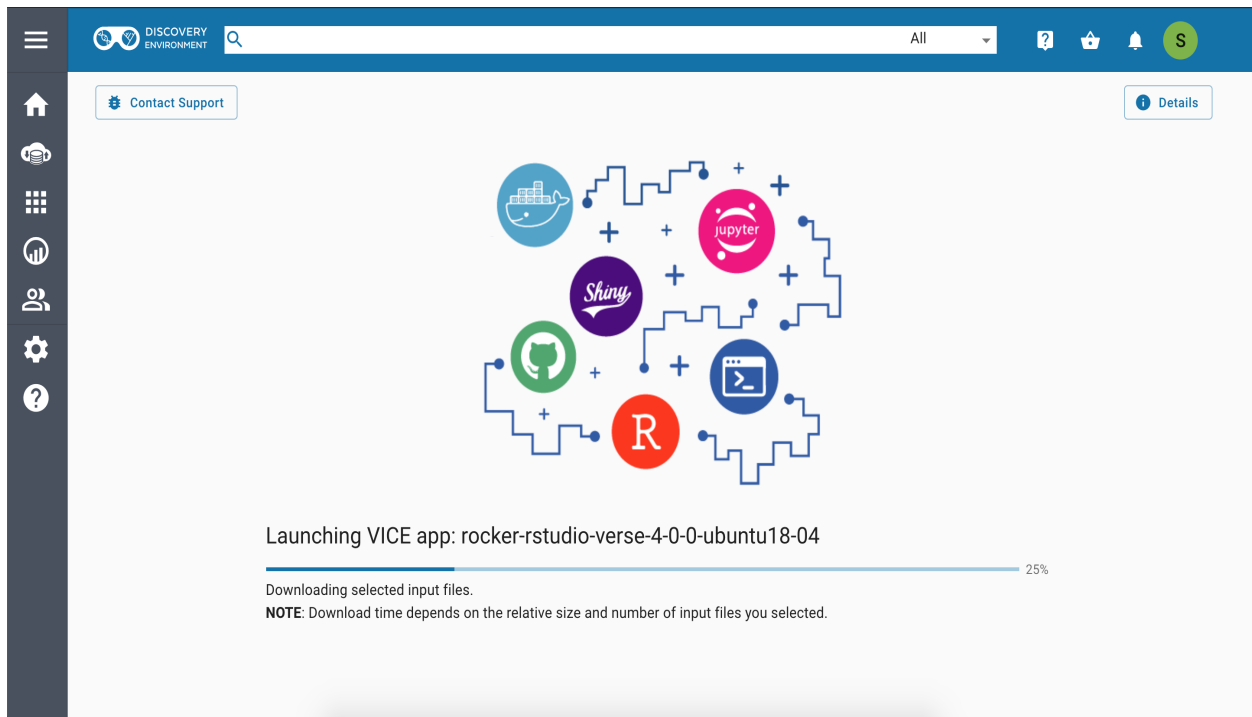
6.3 3. Navigate to rstudio app url

After the analysis starts running, open your notifications and click on the ‘Access your running Analysis here url’. Alternatively, select the app and click on the squarebox from your analysis window.



6.4 4. Launch Rstudio

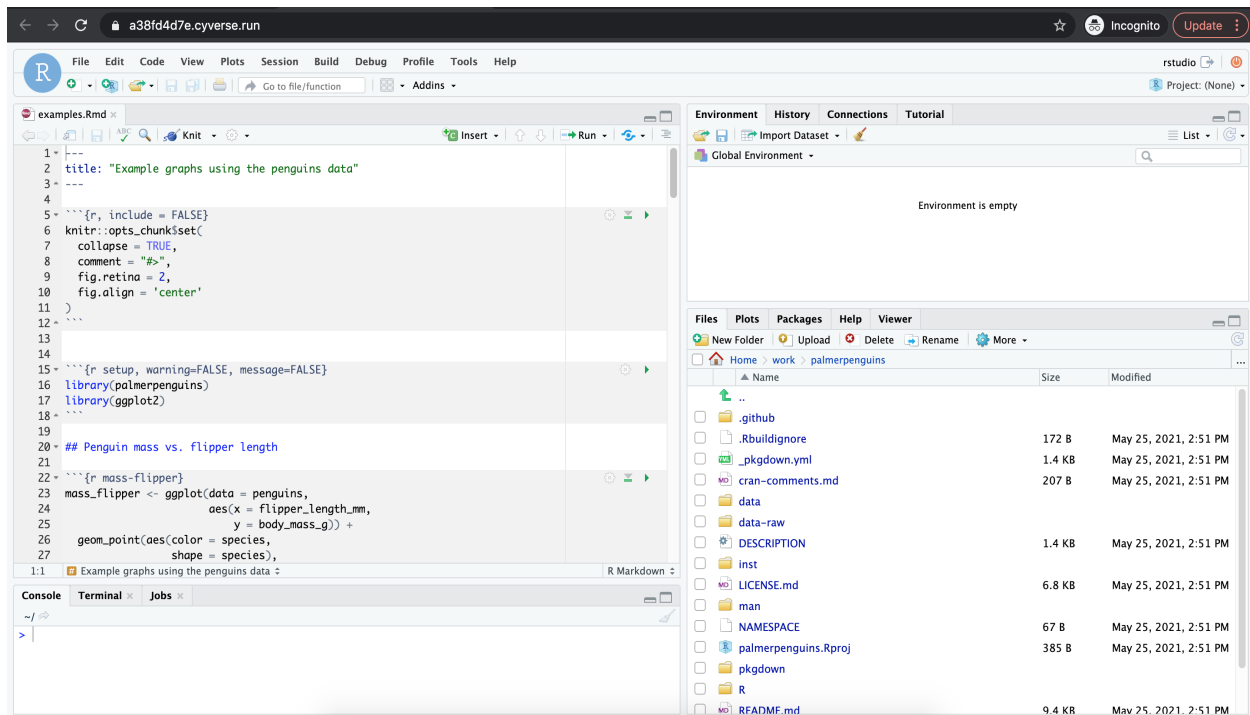
After you start the VICE App, you will be taken to a loading screen.



6.5 5. Write/Run your code

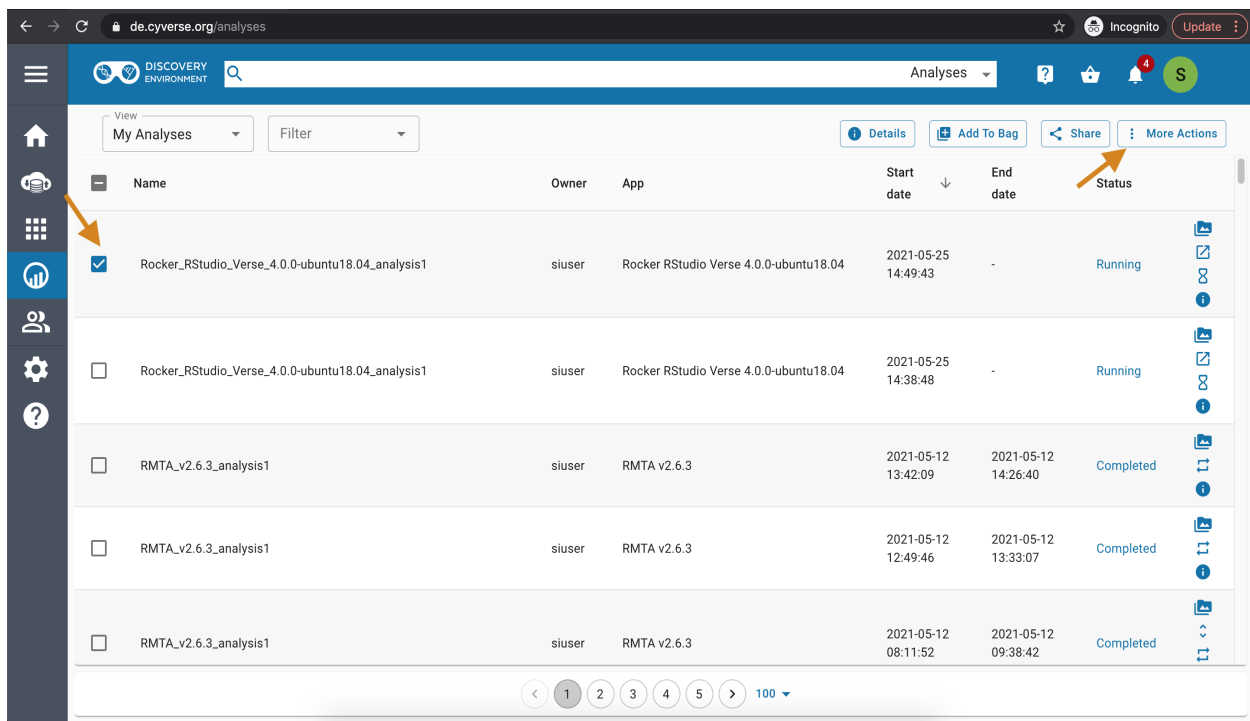
In the Rstudio script section, you can write your code, generate plots, save plots etc.

Tip: As a first step, check that the files you wanted to import are in the app. Go to the bottom right of the app, and check under 'Files' for your files.



6.6 6. Complete and Save Outputs

Complete your analysis by clicking the Analysis window, then select the rstudio analysis. From More Actions tab, click 'Terminate'.



The screenshot shows the 'My Analyses' view in the CyVerse Discovery Environment. The table lists several analyses with columns for Name, Owner, App, Start date, and End date. A context menu is open for the second analysis, 'Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1', with the 'Terminate' option highlighted by an orange arrow.

Name	Owner	App	Start date	End date
<input checked="" type="checkbox"/> Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:49:43	-
<input type="checkbox"/> Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:38:48	-
<input type="checkbox"/> RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 13:42:09	2021-05-12 14:26:40
<input type="checkbox"/> RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 12:49:46	2021-05-12 13:33:07
<input type="checkbox"/> RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 08:11:52	2021-05-12 09:38:42

After you had done this, you can find the outputs that you generated (if any) using the same steps as before, but this time selecting 'Go To Output Folder'.

Warning: Currently, VICE can run for 48 hrs beyond which the apps will be terminated. So make sure you run your analysis before 48 hrs.

Fix or improve this documentation

- On Github: [Github Repo Link](#)
- Send feedback: Tutorials@CyVerse.org

 [Learning Center Home](#)

SHARING VICE APPS WITH COLLABORATORS

You can share your running VICE workspace with colleagues (with a CyVerse account) who can see and edit your notebooks, logs, and outputs.

To share your running workspace

Click on the Analyses

<input type="checkbox"/>	Name	Owner	App	Start date ↓	End date	Status
<input type="checkbox"/>	Jupyter_Lab_SciPy_Notebook_Latest_analysis1	siuser	Jupyter Lab SciPy Notebook Latest	2021-05-25 16:38:21	-	Submitted
<input type="checkbox"/>	Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:49:43	2021-05-25 16:36:03	Completed
<input type="checkbox"/>	Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:38:48	2021-05-25 16:36:03	Completed
<input type="checkbox"/>	RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 13:42:09	2021-05-12 14:26:40	Completed
<input type="checkbox"/>	RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 12:49:46	2021-05-12 13:33:07	Completed

Select the running analysis and click on *Share* from the top bar.

Name	Owner	App	Start date	End date	Status
<input checked="" type="checkbox"/> Jupyter_Lab_SciPy_Notebook_Latest_analysis1	siuser	Jupyter Lab SciPy Notebook Latest	2021-05-25 16:38:21	-	Running
<input type="checkbox"/> Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:49:43	2021-05-25 16:36:03	Completed
<input type="checkbox"/> Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:38:48	2021-05-25 16:36:03	Completed
<input type="checkbox"/> RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 13:42:09	2021-05-12 14:26:40	Completed
<input type="checkbox"/> RMTA_v2.6.3_analysis1	siuser	RMTA v2.6.3	2021-05-12 12:49:46	2021-05-12 13:33:07	Completed

From the sharing window, search your collaborators by CyVerse username, email or group name.

Sharing

Search By CyVerse username, email, or group name...

siuser

Search Results

- siuser

Cancel Done

Opening workspaces shared with you

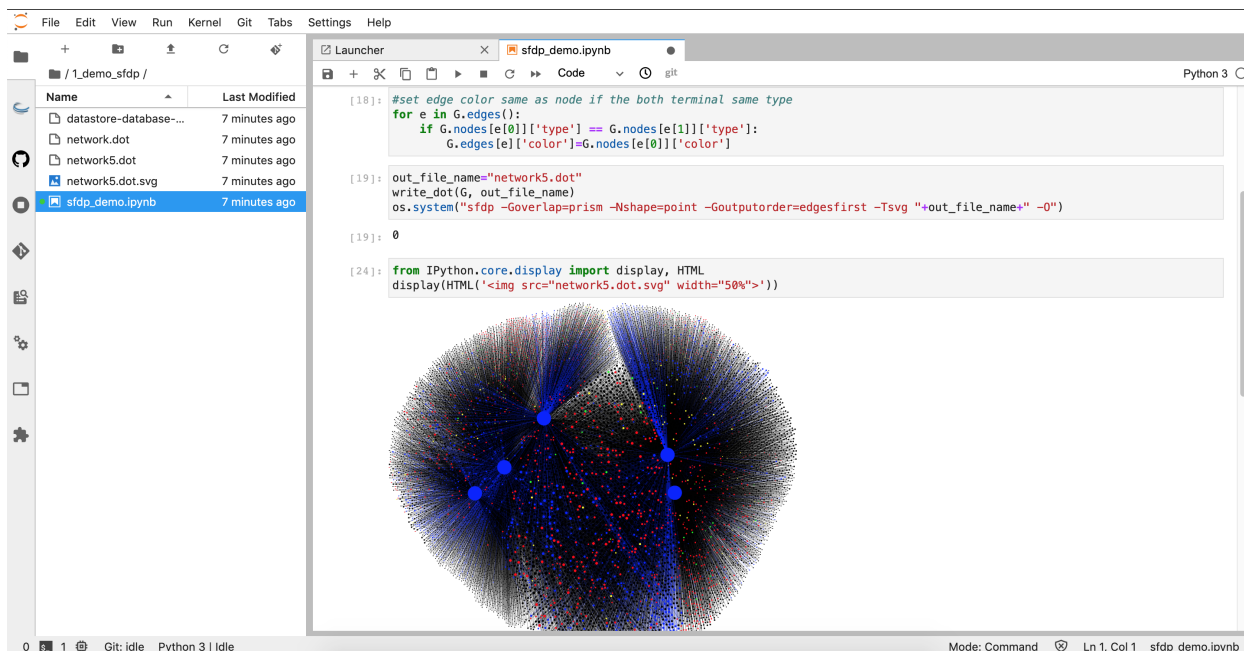
Click on the Analyses

Name	Owner	App	Start date	End date	Status
Jupyter_Lab_SciPy_Notebook_Latest_analysis1	siuser	Jupyter Lab SciPy Notebook Latest	2021-05-25 16:38:21	-	Submitted
Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:49:43	2021-05-25 16:36:03	Completed
Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:38:48	2021-05-25 16:36:03	Completed
RMATA_v2.6.3_analysis1	siuser	RMATA v2.6.3	2021-05-12 13:42:09	2021-05-12 14:26:40	Completed
RMATA_v2.6.3_analysis1	siuser	RMATA v2.6.3	2021-05-12 12:49:46	2021-05-12 13:33:07	Completed

Select the running analysis and click on *Go to analysis* (square arrow box on right).

Name	Owner	App	Start date	End date	Status
Jupyter_Lab_SciPy_Notebook_Latest_analysis1	siuser	Jupyter Lab SciPy Notebook Latest	2021-05-25 16:38:21	-	Running
Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:49:43	2021-05-25 16:36:03	Completed
Rocker_RStudio_Verse_4.0.0-ubuntu18.04_analysis1	siuser	Rocker RStudio Verse 4.0.0-ubuntu18.04	2021-05-25 14:38:48	2021-05-25 16:36:03	Completed
RMATA_v2.6.3_analysis1	siuser	RMATA v2.6.3	2021-05-12 13:42:09	2021-05-12 14:26:40	Completed
RMATA_v2.6.3_analysis1	siuser	RMATA v2.6.3	2021-05-12 12:49:46	2021-05-12 13:33:07	Completed

This will launch the shared analysis in a new window.



There are also different ways to share a VICE app without sharing a running instance of that app.

- Apps that have been made public in the Discovery Environment can be shared with the [public app's URL](#).
- Unpublished apps (those in your private workspace that have not yet been made public) can be shared with specific users or teams. See [Sharing your App or Workflow and Editing the User Manual](#).
- A Quick Launch configuration can be created, and then the URL to the Quick Launch can be shared.

Creating and sharing Quick Launch configurations

Quick Launches provide a way to set default parameter values for an analysis, which can make it much easier to launch similar jobs without having to select the parameter values that the jobs have in common for every new analysis.

To create a Quick Launch, open the app as if you are going to run it, fill in any default parameters for this Quick Launch, then click the **Create Quick Launch** button instead of the **Launch Analysis** button.

The screenshot shows a configuration window titled "Jupyter Lab SciPy Notebook Latest". It has a standard window header with a 3-dot menu, minimize, maximize, and close buttons. The window contains several sections:

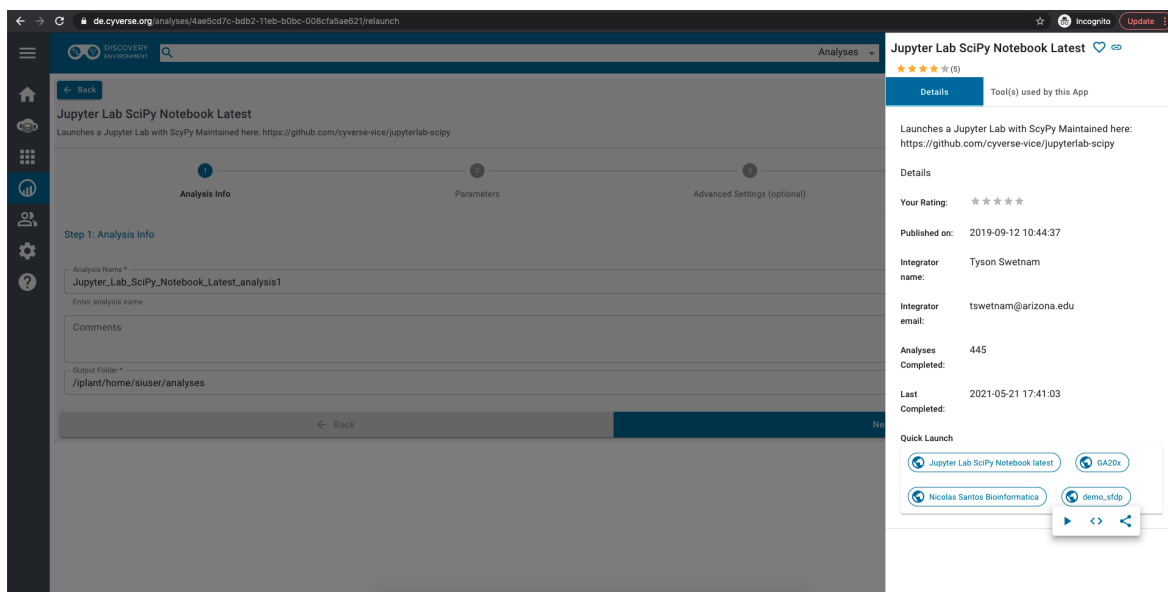
- Analysis Name:** A dropdown menu showing "Jupyter_Lab_SciPy_Notebook_Latest_analysis1".
- Add data at launch:** A section with two options:
 - Add a single file:** A text input field containing "/iplant/home/shared/cyverse_training/example/coffee_cake.txt" and a "Browse" button.
 - Add a folder with data:** A text input field containing "Select a folder" and a "Browse" button.
- Resource Requirements:** A section with a dropdown arrow.
- Buttons:** "Create Quick Launch" and "Launch Analysis" are located at the bottom of the window.

Then you can name the Quick Launch and make it public.

The screenshot shows a configuration dialog titled "Jupyter Lab SciPy Notebook Latest" with a close button (X). It contains the following elements:

- Name:** A text input field with the label "Name*" and the text "Coffee Cake public".
- Public:** A checkbox that is checked, with the label "Public".
- Buttons:** "Cancel" and "Create Quick Launch" are located at the bottom of the dialog.

To share the link for a public Quick Launch, or to copy its badge code for embedding in web pages or in repo markdown files, click the **Quick Launch** menu item from the app's 3-dot menu in the **Apps window**.



Note that private Quick Launch configurations cannot be shared, and clicking on them from this listing simply opens the app launch dialog with its parameters pre-filled, according to the Quick Launch configuration.

If you don't need to save any default inputs or parameters, you can also use the app's public URL for sharing the VICE app, which acts the same as a Quick Launch, but it doesn't have any parameters pre-filled.

Fix or improve this documentation:

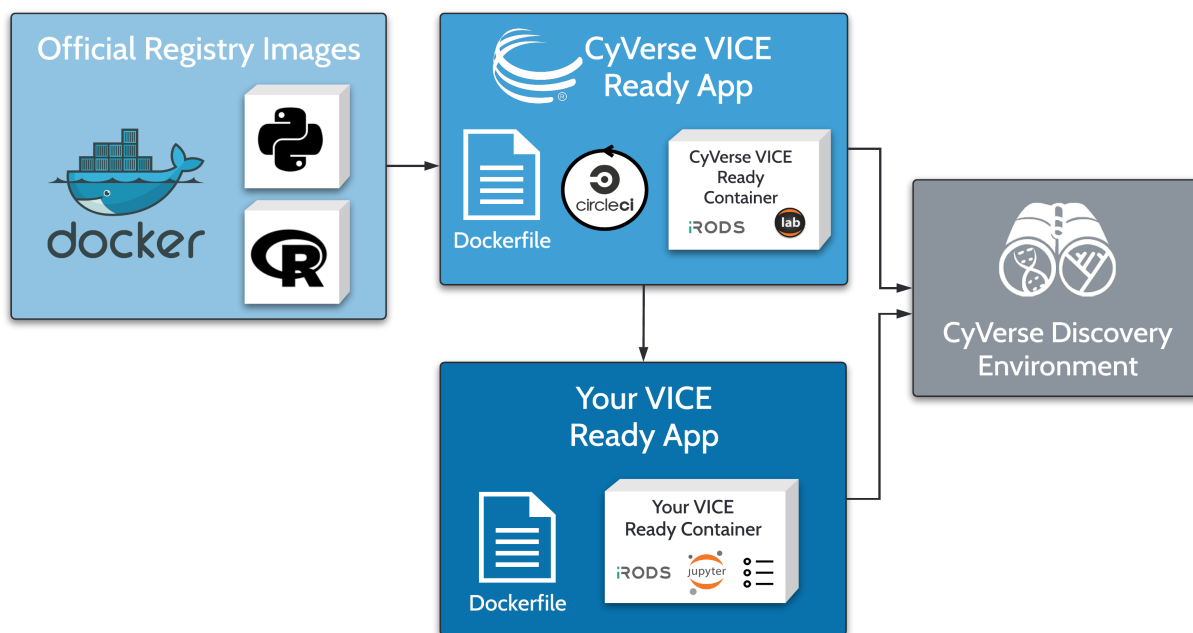
- On Github:
- Send feedback: Tutorials@CyVerse.org



WORKFLOW

CyVerse hosts many popular data science applications, e.g. Jupyter Lab, RStudio, and Shiny. These applications can be started in the Discovery Environment, and the researcher can install additional packages to the running application.

In cases where the installation may be complex, long, or require additional system administrator level access, the researcher can use the existing CyVerse VICE Ready container as a base image for their own new container. The researcher can add their own packages and then deploy the new app in the Discovery Environment.



Fix or improve this documentation

- On Github:
- Send feedback: Tutorials@CyVerse.org



GUIDELINES FOR ADDING INTERACTIVE TOOLS IN DE

9.1 Prerequisites

Adding VICE Tools in DE is different from non-interactive Tools. VICE applications like Jupyter and RStudio run on an open port for enabling their web UI.

1. Ensure that the listen port for the web UI has a sane default and is set in the Dockerfile.
2. The working directory is set
3. All commonly needed dependencies are installed in the container image - you will not have *root* privileges later
4. The default user set
5. Disable any additional authentication (CyVerse provides CAS authentication and authorization).
6. URLs will work sanely behind a reverse proxy. If they don't, you may need to add nginx to the container.=

9.2 Community images as your base image

If you need to set the configurations at all (see above), you'll need to create a new Dockerfile that uses the community-provided image as a base. Your new Dockerfile should deal with custom configurations and dependency installations.

- Jupyter Lab (<https://hub.docker.com/r/cyversevice/jupyterlab-base>)
- RStudio Verse (<https://hub.docker.com/r/cyversevice/rstudio-verse>)
- Shiny Verse (<https://hub.docker.com/r/cyversevice/shiny-verse>)

See some examples of VICE apps that uses community images as base image in the Dockerfile

- MMTF (<https://github.com/sbl-sdsc/mmtf-genomics/blob/master/vice/Dockerfile>)
- Rstudio-Bioconductor (<https://github.com/cyverse/docker-builds/blob/master/rstudio-bioconductor/Dockerfile>)
- Patmatch (<https://github.com/fomigtez/patmatch-binder/tree/master/vice>)

9.3 Test your Docker image

Since the images are built based using Dockerfile, make sure you test the Dockerfile before providing it to us. Dockerfile must have Entrypoint. If you cannot provide us the Dockerfile, you can request integration of the app by doing a tool request.

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org



Learning Center Home

BUILDING DE TOOLS AND APPS

Once you build your Docker image (following the guidelines), the next step is building the Tool.

For this you'll need a Docker image name, any port numbers `PORT`, User ID `UID`, working directory `WORKDIR`, and `ENTRYPOINT`.

10.1 Docker images

The Docker image of your tool is mandatory and it should be available on public registries such as [Dockerhub](#) or [quay.io](#). Alternatively you can provide us the Dockerfile and we will build the Docker image for you. If there is no Dockerfile for the tool that you are interested in, then tell us what tool you are interesting in making us as VICE app.

10.2 Add Tool in DE

The final step in building the VICE tool is to fill up the "Add Tool" form in DE.

Brifely here are the following steps.

- Log in CyVerse [Discovery Environment](#)
- Click on the Apps window and click Manage Tools button on the far right hand side of the window
- Click on Tools button and then finally Add Tools button

You'll see a Add Tool form like this

ladd-tool

- `Tool name` is the name of the tool. This will appear in the DE's tool listing dialog. This is mandatory field. Eg. "jupyterlab-circos"
- `description` is a brief description of the tool. This will appear in the DE's tool listing dialog. Eg. "Circos is a software package for visualizing data and information that was created by Martin Krzywinski"
- `version` is the version of the tool. This will appear in the DE's tool listing dialog. This is mandatory field. Eg. "1.0"
- `Image name` is the name of the image specifier minus the image tag. The image must exist on Dockerhub or quay.io. This is mandatory field. E.g "fomightez/circos-vice"
- `Tag` is the image tag. If you don't specify the tag, the DE will look for the "latest" tag which is the default tag.
- `Docker Hub URL` is the url of the image on the Dockerhub. E.g <https://hub.docker.com/r/fomightez/circos-vice>
- `Type` is the type of Tool. For VICE apps, chose "interactive".

- `OSG Image Path` is path of the image on the OSG. You can skip this for interactive tools.
- `Entrypoint` is the Entrypoint for your tool. Entrypoint should be present in the Docker image and if not, you should specify it here.
- `Working Directory` this is the working directory of the tool and must be filled in with the value you gathered above. E.g `/home/jovyan/vice`
- `UID` is a number and must be filled in with the value you gathered from above. E.g 1000
- `Max CPU Cores` is the number of cores for your tool. Eg. 16
- `Memory Limit` is the memory for your tool. Eg. 64 GB
- `Min Disk Space` is the minimum disk space. Eg. 200 GB
- `Container Ports` must be a list of maps with only a single entry. The key in that entry must be `container_port` and should be filled in with the number value you gathered above.

Warning: It is strongly recommended you do not set the *bind to host* as *true* for your added ports when creating a new App**

10.3 Creating VICE app for your tool

To create a new app, follow the instructions in [here](#)

Important: For VICE apps, be sure to check the box “Do not pass this argument to the command line” for each option you add (for VICE, this is usually just input files and folders).

|vice-do-not-pass|

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org



 [Learning Center Home](#)

FEATURED APPS

CyVerse hosts the recipes of its featured apps on GitHub: <https://github.com/cyverse-vice/>

These images are built from other official projects, and are maintained by CyVerse staff.

INSTANT LAUNCHES

From the Home tab there are several apps that have an **Instant Launch** feature which allows you to start the app in a single click.

These apps launch with their default number of cores, amount of RAM, and timeout, and without input data. You can always import data using *https* protocols or iCommands after launch.

QUICK LAUNCHES

Quick launch buttons are directed URLs that allow you to share an app with pre-set configurations. After selecting an app, you will be taken to the app launcher where you can select input data sets, and set size and time parameters. Any public app can have a quick launch URL generated for it.

To use one of the Featured Launches listed below in the table, copy the badge (button link) to add to where ever you collaborate (a webpage, project notes, documentation, etc.).

To create your on Saved Launch, proceed with launching the app you want to use. This will be a good time to Favorite the app. In the “Review & Launch” panel, click the “Create Saved Launch” button. You will be asked to name your Saved Launch and check the box when prompted if you would like it to be public. Remember which app you saved, You will find the link under Details of the app you saved.

App Name	Dockerfile	Saved Launch
Rocker Project RStudio Verse	GitHub	
Rocker Project RStudio Geospatial	GitHub	
JupyterLab Datascience	GitHub	
JupyterLab Geospatial	GitHub	

You can design your own badge at [Shields.io](#).

Fix or improve this documentation

- On Github:
- Send feedback: learning@CyVerse.org



FREQUENTLY ASKED QUESTIONS

1. What happens if my VICE app has been running for more than 48 hours?
2. Can I extend the 48hr time limit on the VICE app?
3. Can I request the CyVerse team to build the VICE app for my interactive tool if I don't have Docker image?
4. I'm getting this error (or similar) with my docker file:

You must set a unique PASSWORD (not 'rstudio') first! e.g. run with: `docker run -e PASSWORD=<YOUR_PASS> -p 8787:8787 rocker/rstudio`

A: Make sure the environmental password is set for rStudio: `ENV PASSWORD "rstudio1"`. You can also try using this base image for rStudio: `cyversevice/rstudio-base:latest` or `cyversevice/rstudio-verse:3.6.0`. For bioconductor images, use `upendradevisetty/bioconductor:1.0`.

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org



VICE BEST PRACTICES

1. Smaller docker containers are better. Larger images take longer to transfer and load.
2. Use our base images for complex programs that require additional configuration files other than the *Dockerfile*.
3. Create robust documentation with as much metadata as possible.

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org

TOOL TROUBLESHOOTING

16.1 1. Get the port

You'll need to figure out the port that the tool uses to present its web interface. This is mandatory and you can integrate a tool without knowing the port it runs on. If you don't know, you can find the ports that a container image exposes with this command

```
$ docker inspect <image-name>:<image-tag> -f '{{range $port, $val := .ContainerConfig.  
→ExposedPorts}}{{ $port}} {{end}}'
```

Note: Replace *<image-name>:<image-tag>* with the your Docker image

It's possible that multiple or no ports are listed. If that's the case you'll need to refer to the documentation for the tool to figure out the port it uses. Make a note of the port, you'll need it later when integrating the tool in DE. Here are the tools and their ports for common tools such as Jupyter notebook, Rstudio and Shiny. If you are developing any applications based on these tools, you can use these ports while integrating the tool in DE.

Type	Port
Jupyter	8888
Rstudio	80
Shiny	3838

16.2 2. Get the UID of the tool's user

You'll need to figure out the UID of the of the user the app runs as. Many tools will start up as root and then use another user for the actual process, so it might take a little investigation to figure this out. To start this figure out the user that the container is configured to start up using:

```
$ docker inspect <image-name>:<image-tag> -f '{{.ContainerConfig.User}}'
```

If you're lucky that will contain the numerical UID of the user. In that case you can make a note of the UID and move on. Otherwise you have more work to do. The User field can also be empty or set to the username. If its empty, then the user is `root`. If it's a username then you'll need to get the UID from inside the container.

To get the UID for a username run this:

```
$ docker run --rm -it --entrypoint "id" <image-name>:<image-tag> -u <username>
```

Note: Replace `<image-name>:<image-tag>` with the your Docker image

If the User field is empty or `root`, you need to be sure that the process inside the container actually runs as `root`. There are a few ways to check this:

- Fire up the container, exec into it, and do a `ps aux` to see the user the process is running as.

```
$ docker run -d --name app <image-name>:<image-tag>
$ docker exec -it app ps aux
```

- Print out the contents of `/etc/passwd` and check for hints:

```
docker run --rm -it --entrypoint "cat" <image-name>:<image-tag> /etc/passwd
```

- Alternatively check the documentation for the tool.

Note: The UID of the tool can be empty but setting the UID will make sure that the user can write to the input files in the container.

Make a note of the UID, you'll need it later when putting together the JSON for the tool and app.

16.3 3. Get the working directory

You'll need the working directory for the process in the tool container, which may not correspond to the default working directory for the container.

To get the default working directory for the container run this:

```
$ docker inspect <image-name>:<image-tag> -f '{{.ContainerConfig.WorkingDir}}'
```

- If that prints out an empty string, then the default working directory is `/`.
- If the container fires up as `root` but the tool runs as another user, then the working directory may need to be that user's home directory.
- If the container changes to another directory after it starts up, then the working directory may need to be that directory.
- If all else fails, check the documentation and/or try out the container locally to figure out what it does.

Important: Keep in mind that the working directory is where the input files will be made available. Similar to UID, working directory is not mandatory but given jupyter lab's default behavior of showing things in subdirectories of the place it's started. So if you're loading notebooks and data from the Datastore, you want the working directory (where those files are loaded into the container) to be in the right spot

Make a note of the working directory, you'll need it later when putting together the JSON for the tool and app.



CONTACT

17.1 Intercom


17.2 Ask Forum

- You can post your questions at Ask forum <http://ask.cyverse.org>

Fix or improve this documentation:

- On Github:
- Send feedback: Tutorials@CyVerse.org

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: Tutorials@CyVerse.org