

---

# **Resen**

***Release v2019.1.0rc2***

**Jun 14, 2019**



---

## Contents

---

<b>1</b>	<b>Resen</b>	<b>1</b>
1.1	Quickstart . . . . .	2
1.1.1	Installation . . . . .	2
1.1.2	Usage . . . . .	2
1.2	Documentation . . . . .	2
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	General Instructions . . . . .	3
2.1.1	Python 3 . . . . .	3
2.1.2	Docker . . . . .	3
2.1.3	Resen . . . . .	4
2.2	Windows Gotchas . . . . .	4
2.2.1	Install Anaconda and Resen . . . . .	4
2.2.2	Docker . . . . .	4
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Resen Workflow . . . . .	7
3.1.1	Setup a New Bucket . . . . .	8
3.1.2	Work with a Bucket . . . . .	9
3.1.3	Remove a Bucket . . . . .	10



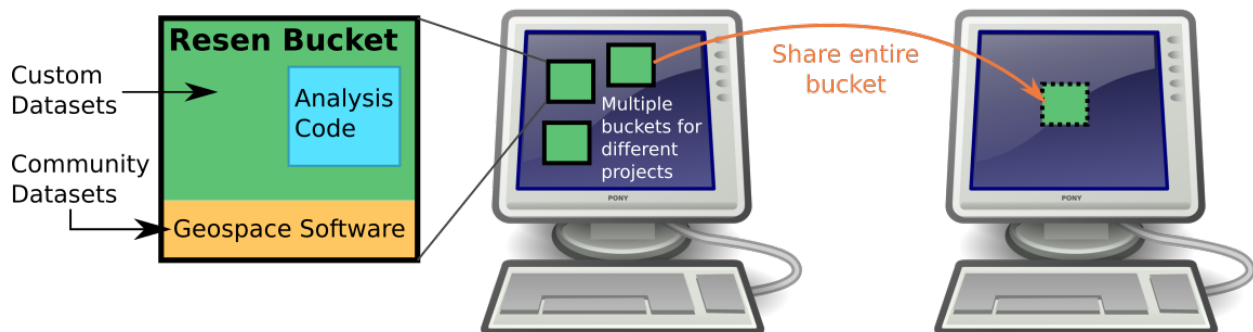
# CHAPTER 1

---

## Resen

---

Resen (REproducible Software ENvironment), is a tool that enables reproducible scientific data analysis, built using python and docker. It is designed to make it easier for geospace researchers to share analysis and results, as well as build off of work others have done. Resen was developed under the InGeO project, currently supported by the National Science Foundation's Cyberinfrastructure for Sustained Scientific Innovation (CSSI) program (Grant #1835573). For more information about the InGeO project, please visit the [InGeO website](#).



Resen is based on the concept of portable environments, or buckets, where code can be developed and run independent of a users system. When you start a resen bucket, it has a variety of common geospace software packages preinstalled and ready for use. This means you have easy access to common models and datasets, and can start using them in your analysis immediately. You can also set up your bucket to access your own datasets, locally stored on your machine.

After you have completed your analysis, you can share an entire bucket with other researchers. Within the bucket, your analysis code will always run exactly the same way, regardless of what system the bucket is on. This means that other researchers should be able to reproduce your work and start building off of it immediately, instead of spending time configuring their system, installing new packages, and setting up file paths so their environment is compatible with your code.

## 1.1 Quickstart

### 1.1.1 Installation

Resen requires both python 3 and docker to be installed.

1. Install [Python 3](#)
2. Install [docker](#)
3. Clone the [resen](#) git repository and install with `pip install .`

Please refer to the [installation documentation](#) for more detailed instructions.

### 1.1.2 Usage

Resen is a command line tool. To start resen, simply enter `resen` at a command prompt:

```
$ resen
```

For a list of available commands, use the `help` command:

```
[resen] >>> help
```

[Resen Workflow Example](#)

## 1.2 Documentation

Complete documentation for Resen is available at <https://resen.readthedocs.io/>.

### 2.1 General Instructions

Resen is built off of both python 3 and docker, so you must have both of these installed for Resen to function.

#### 2.1.1 Python 3

Python (<https://www.python.org/>) is an open source, interpreted programming language that is both powerful and easily to learn. There are many ways you can install python on your system. For new users, we recommend downloading and installing the latest Python 3 Anaconda Distribution (<https://www.anaconda.com/distribution/>) for your system. This will save you the trouble of building a python distribution from scratch.

#### 2.1.2 Docker

Docker CE is the recommended version of Docker to use with Resen. [Installation instructions](#) can be found in the docker documentation. Please read installation instructions carefully! For convenience, some OS specific links are provided below:

**MacOS:** [Install Docker Desktop for Mac](#)

Important! Docker Desktop is only for MacOS 10.10 Yosemite and later. Earlier versions of MacOS should install [Docker Toolbox](#).

**Windows:** [Install Docker Desktop for Windows](#)

Important! If you are already using virtualbox, do NOT install Docker Desktop. Instead, install [Docker Toolbox](#).

**CentOS:** [Get Docker CE for CentOS](#)

**Debian:** [Get Docker CE for Debian](#)

**Fedora:** [Get Docker CE for Fedora](#)

**Ubuntu:** [Get Docker CE for Ubuntu](#)

## 2.1.3 Resen

Install Resen by first cloning the resen GitHub repo (<https://github.com/EarthCubeInGeo/resen>):

```
git clone https://github.com/EarthCubeInGeo/resen.git
```

Change into the resen directory:

```
cd resen
```

In a python 3 environment, use pip to install Resen:

```
pip install .
```

## 2.2 Windows Gotchas

Resen requires both python 3 and docker to function. Here we provide a basic guide for installing both python 3 and docker. We have tested this procedure and know it works. Python 3 and Resen are easy to install. Docker is also fairly easy, but there are some subtle details that need to be emphasized for a smooth installation process.

### 2.2.1 Install Anaconda and Resen

**Anaconda:** We recommend downloading and installing the Python 3 Anaconda Distribution (<https://www.anaconda.com/distribution/>). This simplifies the installation and usage of several common software tools needed to install and run Resen.

**Resen:** Using the start menu search, open the “Anaconda Powershell Prompt” and navigate to a directory where you wish to host the Resen source code. Next, install Resen by first cloning the resen GitHub repo (<https://github.com/EarthCubeInGeo/resen>):

```
git clone https://github.com/EarthCubeInGeo/resen.git
```

Change into the resen directory:

```
cd resen
```

Finally, install Resen:

```
pip install .
```

Once complete, this will provide the command line command `resen`. Next, we need to install Docker.

### 2.2.2 Docker

For Windows, there are 2 options for installing Docker, which depends on what else you use and do with your Windows system:

1. [Docker Desktop for Windows](#)



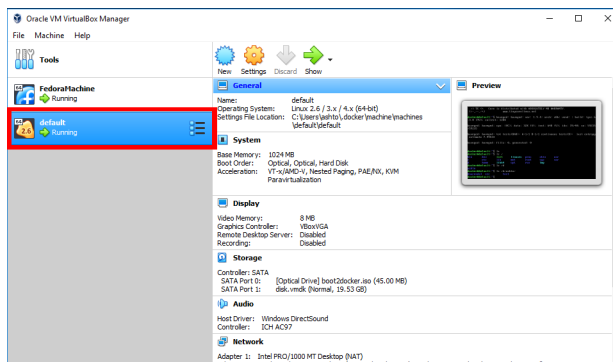
## 2. Docker Toolbox

If you use [Oracle VM VirtualBox](#) to run virtual machines on your Windows system, DO NOT install Docker Desktop. You must instead install Docker Toolbox. Docker Desktop uses Hyper-V, which is not compatible with VirtualBox.

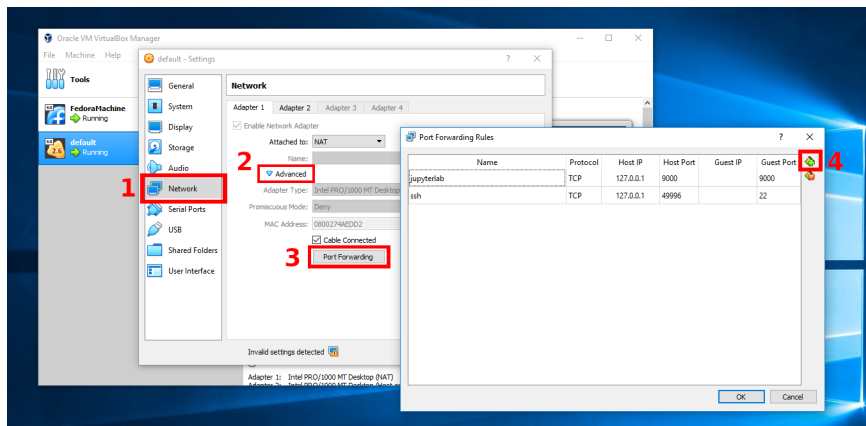
**Docker Desktop TODO.** If you can help fill this in, please make a PR to the develop branch for resen!

**Docker Toolbox** Docker Toolbox essentially works by running docker inside of a Linux virtual machine using VirtualBox. The VM that gets installed is name “default” and we will refer to this “default” Docker virtual machine as the “Docker VM”. To install it Docker Toolbox, do the following:

1. Shutdown any VirtualBox VMs that you currently have running and take note of the VirtualBox version you have installed. Docker Desktop installs an older version of VirtualBox on your system, but this version you are currently running can be upgraded back to the version you are currently running.
2. [Follow the instructions on here to install Docker Toolbox](#). Once installed, restart your computer and then run the Docker Quickstart Terminal from the start menu. TODO: insert screenshot
3. Now we need to add port forwarding and check the shared folders for the Docker VM in VirtualBox. To do this, open VirtualBox and open the “Settings” for the “default” VM, like so:

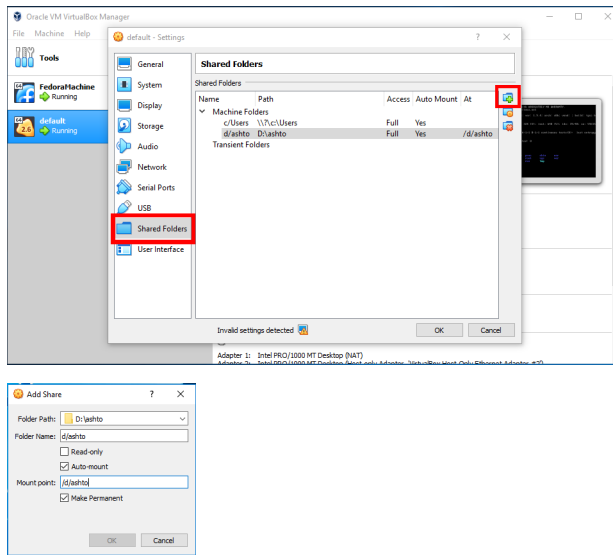


Add a new port forwarding rule by navigating to Settings->Network->Adapter 1->Advanced->Port Forwarding:



Here, we need to add a port forwarding rule for each bucket we create in Resen. Resen requires port 9000 for one bucket and then increments by 1 for every new bucket created. This means that if you have 5 buckets, you will need to make a port forward rule for ports 9000, 9001, 9002, 9003, and 9004. Change both the Host and Guest Ports as seen in the above screenshot.

Now we can optionally add Shared Folders. By default, Docker Toolbox shares the `C:\Users` directory with the Docker VM at `/c/Users`. This means that directories in `C:\Users` will be available to mount into a Resen bucket via the `/c/Users` Shared Folder in VirtualBox. If additional shared directory locations are desired add them. For example:



makes an additional location, `D:\ashto` available to the Docker VM at the location `/d/ashto` so that any directories in `D:\ashto` can be mounted into a resen bucket via `/d/ashto`. After adding or removing Shared Folders, you must restart the Docker VM. This can be done by running:

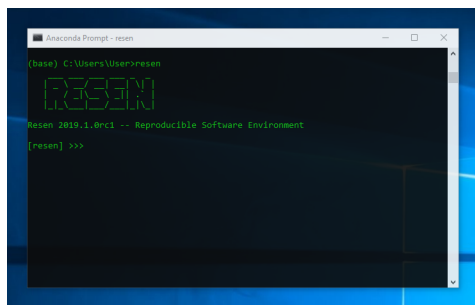
```
docker-machine restart
```

in the “Docker Quickstart Terminal”.

4. Optionally, you can now re-install the newer versions of VirtualBox that you had previously installed. Before doing this, shutdown the Docker Toolbox VM. After re-installing VirtualBox, restart your computer and then open the “Docker Quickstart Terminal” again.

## Running Resen

Now you can run Resen! To do this, open an “Anaconda Powershell Prompt” and type “resen” and hit enter! You should see something similar to:



## CHAPTER 3

---

### Usage

---

To use resen, simply enter `resen` at the command line:

```
$ resen
```

This will open the resen tool:

```
  _ _ _ _ _ _ _ _ _ _
 | _ \ _ / _ | _ | \ | |
 | / _ \ _ \ _ | | . ` |
 | _ \ _ | _ / _ | _ | _ |

Resen 2019.1.0rc2 -- Reproducible Software Environment

[resen] >>>
```

Type `help` to see available commands:

```
[resen] >>> help
```

This will produce a list of resen commands you will use to manage your resen buckets:

```
Documented commands (type help <topic>):
=====
EOF          exit  quit          start_jupyter  stop_jupyter
create_bucket  help  remove_bucket  status
```

To get more information about a specific command, enter `help <command>`.

### 3.1 Resen Workflow

Use Resen to create and remove buckets. Buckets are portable, system independent environments where code can be developed and run. Buckets can be shared between Windows, Linux, and macos systems and all analysis within the

bucket will be run exactly the same. Resen buckets come preinstalled with a variety of common geospace software that can be used immediately in analysis.

### 3.1.1 Setup a New Bucket

1. Creating a new bucket is performed with the command:

```
[resen] >>> create_bucket
```

The `create_bucket` command queries the user for several pieces of information required to create a bucket. First it asks for the bucket name. Creating a bucket named `amber`:

```
Please enter a name for your bucket.
Valid names may not contain spaces and must start with a letter and be less than
↳20 characters long.``
>>> Enter bucket name: amber
```

Next, the user is asked to specify the version of `resen-core` to use:

```
Please choose a version of resen-core.
Available versions: 2019.1.0rc2
>>> Select a version: 2019.1.0.rc2
```

Optionally, one may then specify a local directory to mount into the bucket at `/home/jovyan/work`:

```
Local directories can be mounted to either /home/jovyan/work or /home/jovyan/
↳mount/ in
a bucket. The /home/jovyan/work location is a workspace and /home/jovyan/mount/
↳is intended
for mounting in data. You will have rw privileges to everything mounted in work,
↳but can
specify permissions as either r or rw for directories in mount. Code and data
↳created in a
bucket can ONLY be accessed outside the bucket or after the bucket has been
↳deleted if it is
saved in a mounted local directory.
>>> Mount storage to /home/jovyan/work? (y/n): y
>>> Enter local path: /some/local/path
```

Followed by additional local directories that can be mounted under `/home/jovyan/mount`:

```
>>> Mount storage to /home/jovyan/mount? (y/n): y
>>> Enter local path: /some/other/local/path
>>> Enter bucket path: /home/jovyan/mount/data001
>>> Enter permissions (r/rw): r
>>> Mount additional storage to /home/jovyan/mount? (y/n): n
```

Finally, the user is asked if they want `jupyterlab` to be started:

```
>>> Start bucket and jupyterlab? (y/n): y
```

after which `resen` will begin creating the bucket. Example output for a new bucket named `amber` with `jupyterlab` started is:

```
...adding core...
...adding mounts...
```

(continues on next page)

(continued from previous page)

```

Bucket created successfully!
...starting jupyterlab...
Jupyter lab can be accessed in a browser at: http://localhost:9000/?
↪token=61469c2ccef5dd27dbf9a8ba7c296f40e04278a89e6cf76a

```

## 2. Check the status of the bucket:

```

[resen] >>> status amber
{'bucket': {'name': 'amber'}, 'docker': {'image': '2019.1.0rc2', 'container':
↪'a6501d441a9f025dc7dd913bf6d531b6b452d0a3bd6d5bad0eedca791e1d92ca', 'port':
↪[[9000, 9000, True]], 'storage': [['/some/local/path', '/home/jovyan/work', 'rw
↪'], ['/some/other/local/path', '/home/jovyan/mount/data001', 'ro']], 'status':
↪'running', 'jupyter': {'token':
↪'61469c2ccef5dd27dbf9a8ba7c296f40e04278a89e6cf76a', 'port': 9000}, 'image_id':
↪'sha256:3ba43e401c1b1a8eca8969aec8426a22d99bca349fd837270fa06dbcaefae47',
↪'pull_image': 'earthcubeingeo/resen-
↪core@sha256:c3783e3b7f05ec17f9381a01009b794666107780d964e8087c62f7baaa00049d'}}

```

At this point, the bucket should have a name, an image, at least one port, and optionally one or more storage location. Status should be running if the user decided to have jupyterlab started, otherwise the status will be None.

## 3.1.2 Work with a Bucket

### 1. Check what buckets are available with status:

```

[resen] >>> status

```

Bucket Name	Docker Image	Status
amber	2019.1.0rc2	running

If a bucket is running, it will consume system resources accordingly.

### 2. Stop jupyter lab from a bucket:

```

[resen] >>> stop_jupyter amber

```

The status of amber should now be exited:

```

[resen] >>> status

```

Bucket Name	Docker Image	Status
amber	2019.1.0rc2	exited

The bucket will still exist and can be restarted at any time, even after quitting and restarting resen.

### 3. Start a jupyter lab in bucket amber that has been stopped:

```

[resen] >>> start_jupyter amber

```

The status of amber should now be running:

```

[resen] >>> status

```

Bucket Name	Docker Image	Status
amber	2019.1.0rc2	running

The jupyter lab server starts in the /home/jovyan directory, which should include the persistent storage directories work and mount. The user can alternate between the jupyter lab and the classic notebook view by

changing the url in the browser from `http://localhost:8000/lab` to `http://localhost:8000/tree`. Alternatively one can switch from the lab to the notebook through Menu -> Help -> Launch Classic Notebook.

### **3.1.3 Remove a Bucket**

The user can delete a bucket with the following command:

```
[resen] >>> remove_bucket amber
```

A bucket that is running needs to be stopped before removed. **WARNING:** This will permanently delete the bucket. Any work that was not saved in a mounted storage directory will be lost.