

# Red Cloud Documentation

This document is written by Jenny Chen as a tutorial for people who want to get started with using the CAC computing resource. If you have any questions or suggestions, please reach out to [jc3467@cornell.edu](mailto:jc3467@cornell.edu).

## Set up instances and connect through terminal

### Setting up

The Cornell CAC has a series of videos on how to set up an instance on [Youtube](#).

The website for managing all instance is <https://redcloud.cac.cornell.edu>

The basic workflow is the following:

1. Generate a key pair [link](#)
2. Generate a security group [link](#)
3. Launch an instance [link](#)

### Connect

The CAC channel also has a [video](#) about how to connect to the instance. I'll briefly talk about the process here.

To connect to the server, we need to make sure we're **in the directory that contains the key pair** we generated. In practice, I always need to connect to the Cornell VPN in order to connect to the server.

To connect, use the following command

```
ssh -i name_of_key username@instance_IP
```

An as example:

```
ssh -i cloud.key ubuntu@128.84.11.193
```

cloud.key is the name of my key, and the IP address after the @ symbol can be found on the website depending on the instance. Username is just ubuntu if you follow their instructions

(or if you use any ubuntu images).

<input type="checkbox"/>	Instance Name	Image Name	IP Address
<input type="checkbox"/>	test-instance	-	128.84.8.100
<input type="checkbox"/>	qslack-1	-	128.84.11.193

## Resizing the instance

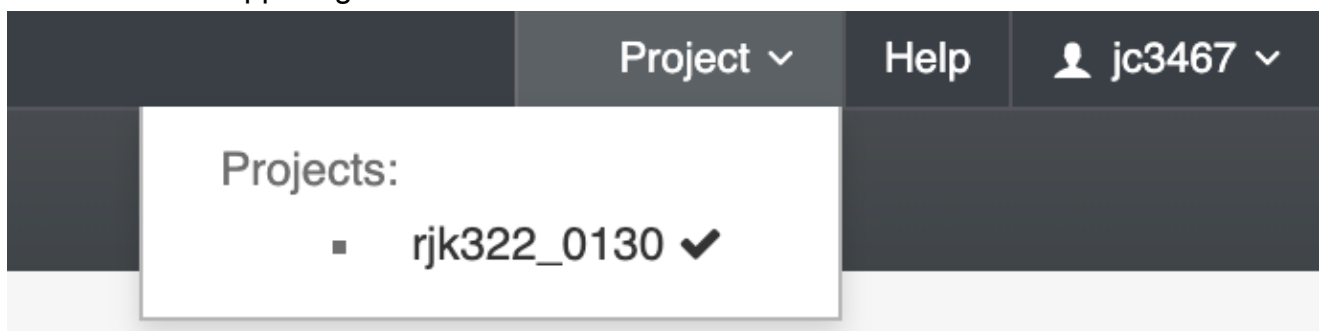
It's recommended to set up the instance with the smallest possible flavor and then scale it up later. Normally, we only need one core to set up (i.e. using c1.m8). To resize the instance, follow the instruction of this [link](#).

After you see "Confirm/revert the resizing", remember to press the confirm button.

When you buy the subscription, you will have a certain number of core hours to use. A core hour means "one core running for one hour". Therefore, if your instance is using  $x$  cores, you will use  $x$  core hours after running one hour. If the status of your instance is "Active", that means you're using your core hours. Make sure to use "Shelve Instance" when you're done using the cloud. See more detailed description [here](#).

## Selecting project

Since a person can be included in multiple projects, you need to select the right project to work on on the upper right corner of the dashboard.



## Install Miniconda

### Installing

We choose to use Miniconda instead of Conda in the server to save the disk space. For further information, check this [link](#).

To install Miniconda through terminal after connected to the instance, follow the tutorial for linux system [here](#).

First, run

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
-O ~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
```

Then, run

```
~/miniconda3/bin/conda init bash
~/miniconda3/bin/conda init zsh
```

Finally, use command `exit` to log out of the server and log in again to see conda is installed. If it is installed successfully, you should see "(base)" at the beginning of each line.

```
[(base) ubuntu@test-instance:~$
```

Run `conda list` to see if command conda can be used.

## Trouble-shooting

If after you re-login and you cannot use any conda commands, try run the following lines

```
export PATH="/home/ubuntu/miniconda3/bin:$PATH"
source ~/.bashrc
```

If conda is working fine after these two lines, add the following lines to your `.bash_profile`, and if needed, creating a `.bash_profile` ( `touch .bash_profile` ) in the process:

```
if [ -f ~/.bashrc ]; then
    source ~/.bashrc
fi
```

To modify any file, choose any terminal editor of your choice like vim, nano, etc.

If above procedure did not solve the problem, try to reach out to CAC help center. You can send a ticket to them [here](#).

## Setting up new conda environment

To start, use the following command

```
conda create --name myenv python=3.9 --yes
```

This way we automatically have python installed and we can use the pip command later. The `--yes` flag will automatically allow installation of dependencies. It's unnecessary to add if you don't want to.

After creating this environment, we need to activate it and then modify it.

```
conda activate myenv
```

## Additional note on Jupyter notebook

I'm not sure how to run Jupyter notebook through the server. After running the commands, the list of files that I can read and modify are files on my local computer and not the server. This part is for people who want to install Jupyter notebook on their local computers. If you want to use Jupyter notebook on CAC, please reach out to their help center.

If you want to launch a jupyter notebook instance through this environment, first we need to install package jupyter. Use

```
pip install jupyter
```

If there's error saying

```
TypeError: warn() missing 1 required keyword-only argument: 'stacklevel'
```

Then try run

```
pip install --upgrade notebook
```

Use `jupyter notebook` to start running. This way, we can run jupyter notebooks using this new environment.

## Transfer files

A standard library to transfer files from local to a server or a server to local is the scp library. This is a [quick tutorial](#) on this topic. The general workflow should be very similar to using `cp` (copy) command on local machines.

## Local to server

To transfer a local file to the server, run the following command in the **local terminal**:

```
scp filename username@IP:/home/username/destination_dir
```

An example for the cac instance could be:

```
scp QSlack_cleaned/VQED_test.py  
ubuntu@128.84.11.193:/home/ubuntu/qs lack_home/
```

To transfer an entire directory to the server is similar (with an extra `-r` flag):

```
scp -r local_dir_name username@IP:/home/username/target_dir_name
```

An example could be:

```
scp -r QSlack_cleaned ubuntu@128.84.11.193:/home/ubuntu/
```

## Server to local

To transfer a file from the server to a local machine, run the following command in the **local terminal**:

```
scp username@ip_address:/home/username/filename local_dir_name
```

An example could be:

```
scp ubuntu@128.84.11.193:/home/ubuntu/test.txt ./test
```

For the final destination, using `.` means it will be transferred to the current directory of the local machine.

Similarly, using `-r` flag can transfer an entire directory

```
scp -r username@IP:/home/username/dir_name target_local_dir_name
```

An example could be:

```
scp -r ubuntu@128.84.11.193:/home/ubuntu/qs lack_home ./test
```

## Additional important notes

### A list of websites that can be helpful

1. <https://www.cac.cornell.edu/services/>
2. [https://www.cac.cornell.edu/wiki/index.php?title=Red\\_Cloud](https://www.cac.cornell.edu/wiki/index.php?title=Red_Cloud)
3. <https://www.cac.cornell.edu/TechDocs/redcloud/#important-pages>
4. <https://www.cac.cornell.edu/TechDocs/clusters/#getting-started-on-private-clusters>
5. <https://www.cac.cornell.edu/TechDocs/redcloud/#cac-account-first-time-login>
6. <https://www.cac.cornell.edu/TechDocs/clusterinfo/linuxtutorial/>
7. <https://rt.cac.cornell.edu/index.html> (send ticket)
8. <https://www.cac.cornell.edu/services/cu/Memberlimits.aspx> (check balance, Cornell user)
9. <https://www.cac.cornell.edu/cu/explore.aspx> (apply for exploratory account)

### What to do when you're done running the code

Always remember to **shelve the instance** when you're done using it. Otherwise, your core hours will be used even if you're not running any code.

### Exploratory Account

Cornell students, faculty, and staff are eligible for a 4-month Red Cloud **Exploratory Account** that provides you with 165 core hours, 50GB storage, and 1 hour consulting help for free. **However**, once a person is added to a regular project, they **cannot set up an exploratory account**. Note that for exploratory projects, CAC is the PI and you are the only project member. Make sure to include the right person in the project is important. To apply for an exploratory account, use this [link](#).

### Subscription rate

The current subscription rate is described on this [website](#). Check the **Subscription Unit** and **Subscriptions for Cornell Users** sections for more details.

1. By April 2024, with first subscription, Cornell users can buy 8,585 core hours and 50GB storage. Additional storage is \$100.00/TB/year or \$0.10/GB/year. Multiple subscriptions can have discounts.
2. Optional consulting hour is 81.71 dollars per hour if needed.

### How to check how many resources do I have left?

Cornell users should be able to see how many hours and storage you have left using [this link](#). External users can use [this link](#). You should be able to see the information for all projects you're in.

## Acknowledging CAC

According to the instruction on the CAC website:

When you publish a paper, make presentations, or are interviewed by the Cornell Chronicle, national news media, etc., please acknowledge the Center by including:

☞ "This research was conducted with support from the Cornell University Center for Advanced Computing."

Alternatively, the full acknowledgement is:

☞ "This research was conducted with support from the Cornell University Center for Advanced Computing, which receives funding from Cornell University, the National Science Foundation, and members of its Partner Program."