CARC\_onDemand\_setup

This link is very helpful for all the setups:

<https://www.carc.usc.edu/user-information/user-guides/hpc-basics/getting-started-ondemand>

Here are the steps for macOS users, if you are a Windows or Linux user, please check the link for more details. I’m using my username for examples, so please remember to change it to yours!

CARC OnDemand will only be accessible via a connection to either USC secure network or a USC VPN.

VPN link:

<https://itservices.usc.edu/vpn/>

**Files/directories**

Every user will have access to their **/home1, /scratch1, and /scratch2** directories. These three directories are default.

We have our project directory which is **/project/aealmada\_561/**, and you might also have other project directories.

* The two scratch directories are intended for temporary files, needed periodically backup. Better to store all your work under the project directory.

A screenshot of a computer

Description automatically generated

**Shell access**

OnDemand provides **Discovery, Endeavour, and Data Transfer clusters**. I use Discovery Cluster. To access Discovery cluster, select Cluster 🡪 Discovery Cluster Shell Access from the dropdown.

A screenshot of a computer

Description automatically generated

**Getting started with Discovery**

This link is helpful:

<https://www.carc.usc.edu/user-information/user-guides/hpc-basics/getting-started-discovery>

1. **Connecting to USC network of VPN**
2. **Logging in**

You will need to use a secure shell client. This enables you to connect to a remote computer via SSH (Secure SHell). You will need your USC NetID to log in. You don’t need to use SSH every time after your first log-in.

For macOS user:

Connect to Discovery using the **Terminal** application. Open a new terminal window and enter:

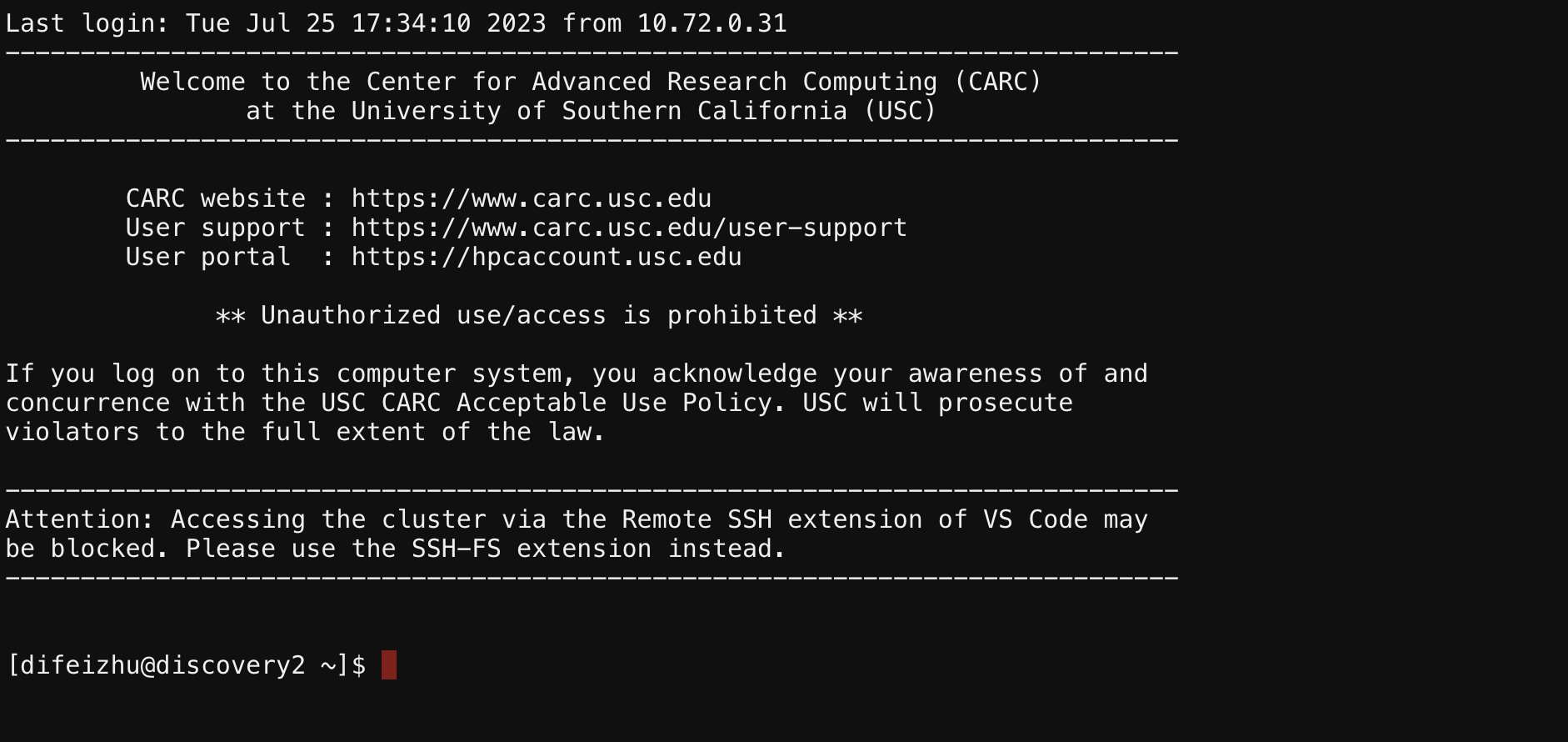
ssh difeizhu@discovery.usc.edu

# username is your USC NetID, which is the same username for your USC email account.

After entering the command, you will be prompted to enter your USC NetID password. (There will be no visual feedback for your password.)

For Windows and Linux users, you can find the commands here: <https://www.carc.usc.edu/user-information/user-guides/hpc-basics/getting-started-discovery>

Usually, you will see something like this after logging in:



1. **Organizing files and making your own subdirectory**

You can access the four directories on Discovery Cluster:

/home1

/project/aealmada\_561

/scratch1

/scratch2

You can find more detailed introduction about all the file systems through the link under “Getting started with Discovery”. Here are the steps for a quick setup.

Use the myquota command to see the directories you have access to along with quota information.

Home file system:

You will always start in your home directory when you log in to Discovery. Home directory can be accessed by:

cd

pwd

/home1/difeizhu

Use cd command to quickly change to your home directory.

Project file system:

Our project directory can be accessed by:

cd /project/aealmada\_561

pwd

/project/aealmada\_561

You can create our own subdirectory within the project directory. Like this:

Remember to use pwd command to find where you are!

# if you are in the project directory now:

mkdir judyz

# if you are in somewhere else:

mkdir /project/aealmada\_561/judyz

Scratch file systems:

The /scratch1 and /scratch2 file systems are intended for temporary files and I/O operations, so the files stored in both scratch file systems are not backed up in any way. I highly recommend you store all your work in our project folder!

The /scratch1 can be quickly accessed by:

cds

pwd

/scratch1/difeizhu

The /scratch2 can be quickly accessed by:

cds2

pwd

/scratch2/difeizhu

1. **Transferring files**

Sometimes you need to transfer files between your local machine and Discovery or from the internet to Discovery. Here are some commands. You can read more information about transferring files here:

<https://www.carc.usc.edu/user-information/user-guides/data-management/transferring-files-command-line>

Between your local machine and Discovery:

The available options are sftp and rsync for macOS and Linux. Windows users please check the link to see more information.

I prefer sftp command that requires authenticating only once, but feel free to check the link to see rsync instructions if you’d like to try that one.

From your local computer (**macOS Terminal**), first log in to a CARC node hpc-transfer1 and authenticate via Duo:

sftp difeizhu@hpc-transfer1.usc.edu

You will be prompted to enter the password of your USC NetID like this:

A screen shot of a computer

Description automatically generated

After entering your USC NetID, you will authenticate via Duo:

A screenshot of a computer code

Description automatically generated

If it is your first time logging in, you will be asked “Are you sure you want to continue connecting (yes/no)?”. Enter “yes”. You will see sftp> once you are connected.



Navigating locally (**macOS Terminal**):

Remember to add a “l” in front of all the commands, indicating the local navigation.

sftp> lpwd

/Users/zhudifei

sftp> lcd Desktop

sftp> lls

course.pdf Almada\_lab

Navigating remotely (**CARC system):**

Use all the commands as usual.

sftp> pwd

Remote working directory: /home1/difeizhu

sftp> cd /project/aealmada\_561

sftp> ls

judyz

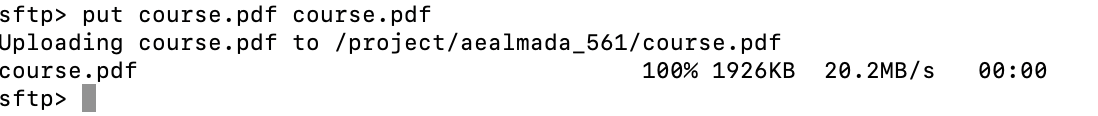
Uploading file/directory from local machine to CARC system:

Use put command. Remember to use lpwd or pwd to check where you are on both platforms before operating!

sftp> put course.pdf course.pdf

# to upload a directory, add the -R option and specify the path to the local directory

You will see something like this:



Some files are too big, so it might take some time to upload. Once your progress showed 100%, you can refresh your CARC OnDemand server, and you will see the file you just uploaded.

A screenshot of a device

Description automatically generated

Downloading file/directory from CARC systems to local computer:

It is basically same as uploading files, but use get command. Remember to use lpwd or pwd to check where you are on both platforms before operating!