TOPIC: INTRODUCTION TO CONDA
WORKSHOP DATE: January 19, 2022
INSTRUCTOR: Jillian Rumore

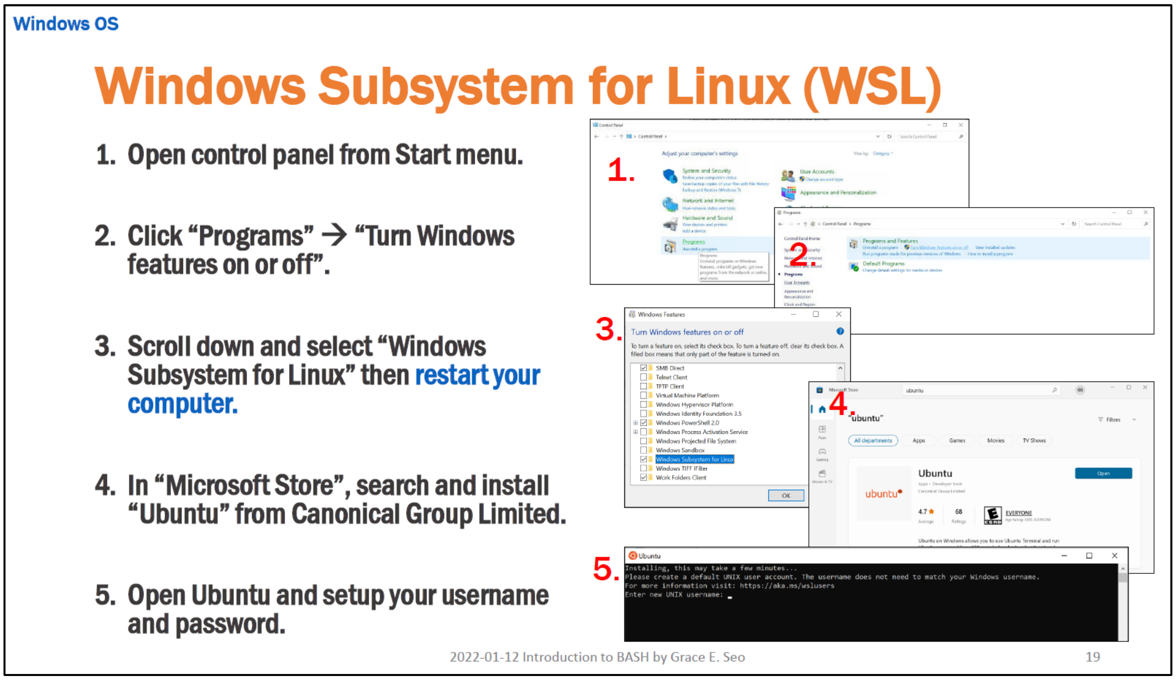
Due to time constraints, the workshop format is not intended to be a demo style tutorial. However, if you would like to follow along and try the commands out during the workshop, please ensure you have installed conda according to the instructions listed below that are appropriate for your operating system.

GETTING STARTED - INSTALLING A BASH TERMINAL

Using Bash in the Terminal is **a powerful way of interacting with your computer**, allowing you to perform many tasks more efficiently, as well as automate, and replicate workflows across different operating systems.

Instructions included below for installing the BASH terminal are from MMID Coding Workshop - Introduction to BASH.

WINDOWS USERS



Windows OS

Windows Subsystem for Linux (WSL)

1. Open control panel from Start menu.
2. Click "Programs" → "Turn Windows features on or off".
3. Scroll down and select "Windows Subsystem for Linux" then restart your computer.
4. In "Microsoft Store", search and install "Ubuntu" from Canonical Group Limited.
5. Open Ubuntu and setup your username and password.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: _
```

1.
2.
3.
Turn Windows features on or off
☑ Windows Subsystem for Linux
OK
4.
"ubuntu"
Ubuntu
Open
5.
Ubuntu

2022-01-12 Introduction to BASH by Grace E. Seo 19

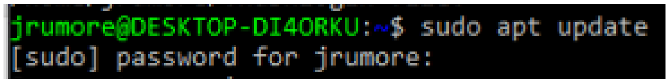
An instructional video on how to setup the Windows Subsystem for Linux can be found here:

https://www.youtube.com/watch?v=IAIMD_NpXCk

To ensure all existing packages on the Linux system are up to date, open the terminal and run the following command:

```
sudo apt update
```

You will be asked to enter your password for Ubuntu, which was created when installing the Windows Subsystem for Linux (WSL).



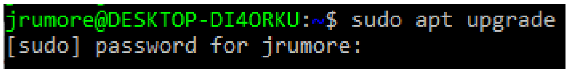
```
jrumore@DESKTOP-DI4ORKU:~$ sudo apt update
[sudo] password for jrumore:
```

A list of the packages and their version that are available for updating will be printed to the terminal. Please note, this may take a while to complete.

After updating the lists, run the following command to install/ update the packages:

```
sudo apt upgrade
```

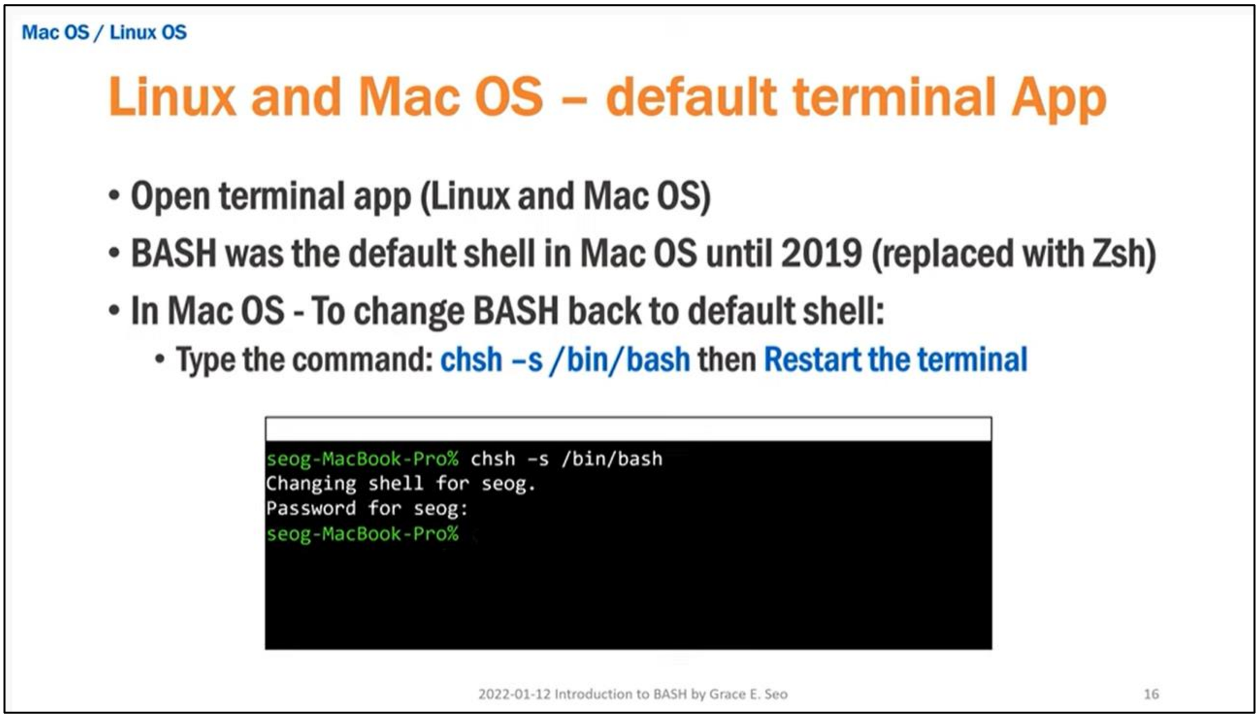
You will be asked to enter your password for Ubuntu, which was created when installing the Windows Subsystem for Linux (WSL).



```
jrumore@DESKTOP-DI4ORKU:~$ sudo apt upgrade
[sudo] password for jrumore:
```

Please note, updating packages may take a while to complete.

MacOS USERS



Mac OS / Linux OS

Linux and Mac OS – default terminal App

- Open terminal app (Linux and Mac OS)
- BASH was the default shell in Mac OS until 2019 (replaced with Zsh)
- In Mac OS - To change BASH back to default shell:
  - Type the command: chsh –s /bin/bash then Restart the terminal

```
seog-MacBook-Pro% chsh -s /bin/bash
Changing shell for seog.
Password for seog:
seog-MacBook-Pro%
```

2022-01-12 Introduction to BASH by Grace E. Seo 16

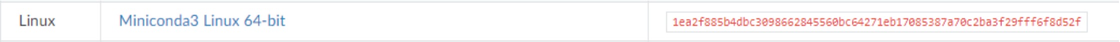
INSTALLING MINICONDA

The instructions listed below are for installation of Miniconda, a free minimal installer for conda. Installing Miniconda will provide you with fast access to working in conda and allow you to create conda environments tailored to your individual needs, as well as boost portability and reproducibility of your research and workflows.

INSTRUCTIONS FOR PARTICIPANTS WHO DO NOT HAVE ACCESS TO THE HIGH PERFORMANCE COMPUTING CLUSTER AT NATIONAL MICROBIOLOGY LABORATORY

WINDOWS USERS

1. Navigate to https://docs.conda.io/en/latest/miniconda.html and click on the following file to download:



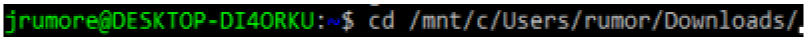
Linux	Miniconda3 Linux 64-bit	1ea2f885b4dbc3098662845560bc64271eb17085387a70c2ba3f29fff6f8d52f

2. Save it in the Downloads folder.
3. Open up the Ubuntu terminal from the Start menu.
4. Navigate to your Downloads directory using the following command:

```
cd /mnt/c/Users/USERNAME/Downloads
```

```
▪ USERNAME: user name on your computer
```

Example:



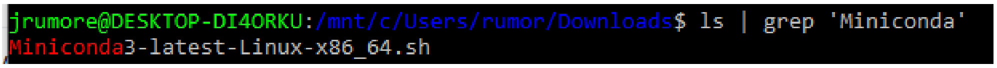
```
jrumore@DESKTOP-DI4ORKU:~$ cd /mnt/c/Users/rumor/Downloads/
```

5. Verify the `Miniconda3-latest-Linux-x86_64.sh` file was downloaded by searching for it in the Downloads directory.

```
ls | grep ‘Miniconda’
```

```
▪ ls: list files and directories
▪ grep: search for a particular pattern
```

Example:

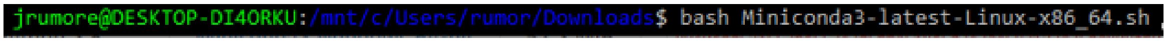


```
jrumore@DESKTOP-DI4ORKU:/mnt/c/Users/rumor/Downloads$ ls | grep 'Miniconda'
Miniconda3-latest-Linux-x86_64.sh
```

6. To install Miniconda, run the command:

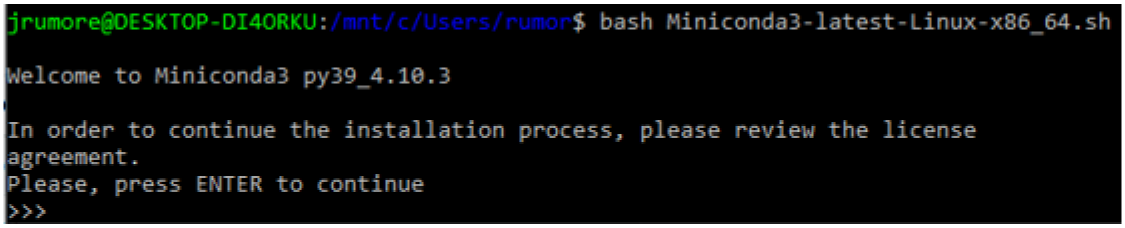
```
bash Miniconda3-latest-Linux-x86_64.sh
```

Example:



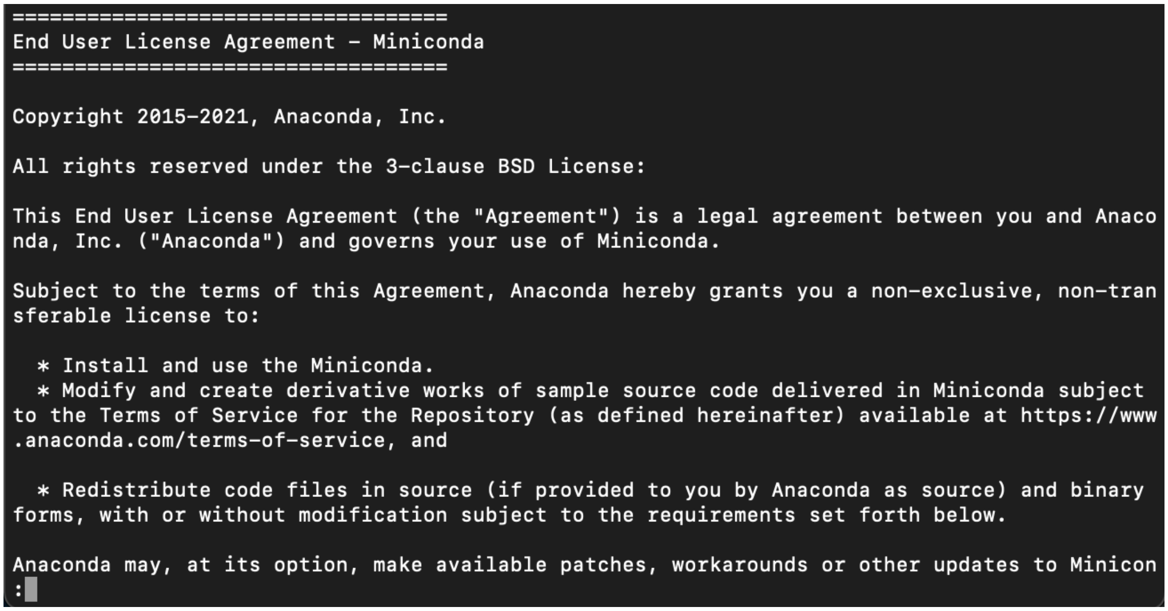
```
jrumore@DESKTOP-DI4ORKU:/mnt/c/Users/rumor/Downloads$ bash Miniconda3-latest-Linux-x86_64.sh
```

7. A terminal prompt will appear asking you to press **ENTER** to start the installation process. Press **ENTER**.



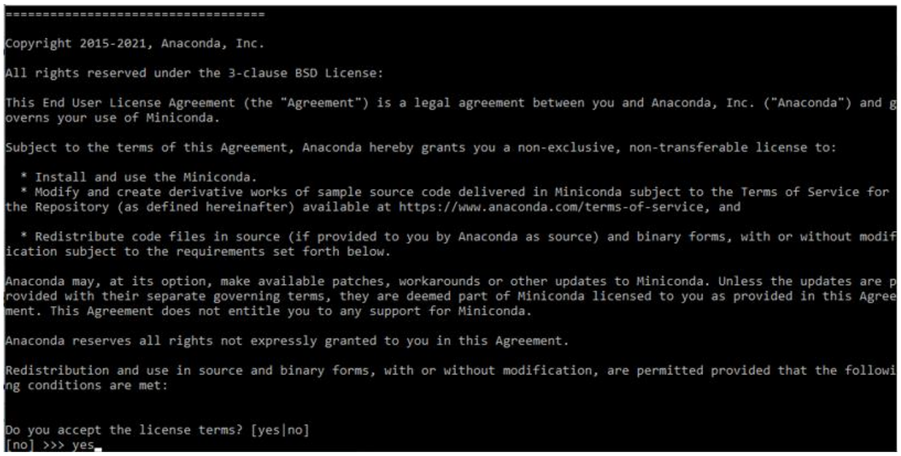
```
jrumore@DESKTOP-DI4ORKU:/mnt/c/Users/rumor$ bash Miniconda3-latest-Linux-x86_64.sh

Welcome to Miniconda3 py39_4.10.3

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

8. The License Terms will appear. You can read through the agreement by pressing ENTER, or, you can skip it by typing **q** in the terminal.



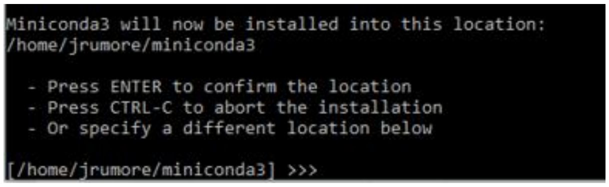
```
===================================
End User License Agreement - Miniconda
===================================

Copyright 2015-2021, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaco
nda, Inc. ("Anaconda") and governs your use of Miniconda.

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-tran
sferable license to:

  * Install and use the Miniconda.
  * Modify and create derivative works of sample source code delivered in Miniconda subject
to the Terms of Service for the Repository (as defined hereinafter) available at https://www
.anaconda.com/terms-of-service, and

  * Redistribute code files in source (if provided to you by Anaconda as source) and binary
forms, with or without modification subject to the requirements set forth below.

Anaconda may, at its option, make available patches, workarounds or other updates to Minicon
:
```

9. You will then be asked if you accept the license terms. Type **yes** to proceed and press **ENTER**.



```
===================================

Copyright 2015-2021, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and g
overns your use of Miniconda.

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

  * Install and use the Miniconda.
  * Modify and create derivative works of sample source code delivered in Miniconda subject to the Terms of Service for
the Repository (as defined hereinafter) available at https://www.anaconda.com/terms-of-service, and

  * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modif
ication subject to the requirements set forth below.

Anaconda may, at its option, make available patches, workarounds or other updates to Miniconda. Unless the updates are p
rovided with their separate governing terms, they are deemed part of Miniconda licensed to you as provided in this Agree
ment. This Agreement does not entitle you to any support for Miniconda.

Anaconda reserves all rights not expressly granted to you in this Agreement.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the followi
ng conditions are met:


Do you accept the license terms? [yes|no]
[no] >>> yes
```

10. By default, an installation folder will be created in the home directory of the system. To accept, press **ENTER.**

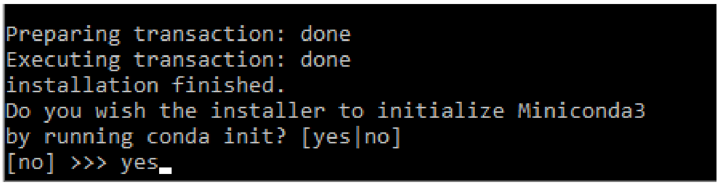


```
Miniconda3 will now be installed into this location:
/home/jrumore/miniconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/jrumore/miniconda3] >>>
```

In the screenshot above, miniconda is being installed in /home/jrumore/miniconda3.

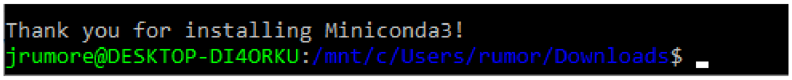
11. The software will now be installed. This will take a few minutes.

12. Once the installation has finished, it will ask you if you want the installer to initialize Miniconda3. Type **yes** and press **ENTER**. This will allow you to easily work with conda commands in the terminal.



```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

13. Once the initialization has completed, a prompt will appear in the terminal thanking you for installing Miniconda. Close and re-open the terminal for the changes to take effect.



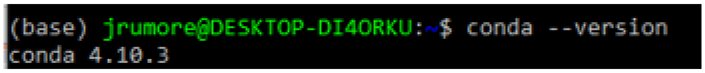
```
Thank you for installing Miniconda3!
jrumore@DESKTOP-DI4ORKU:/mnt/c/Users/rumor/Downloads$
```

14. Close the terminal, and re-open Ubuntu from the Windows Start menu.

15. To verify the installation has successfully completed, type the following command and press **ENTER**:

```
conda --version
```

Example:



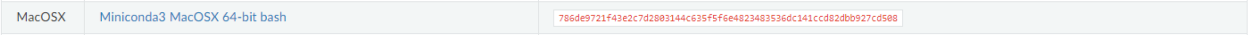
```
(base) jrumore@DESKTOP-DI4ORKU:~$ conda --version
conda 4.10.3
```

If miniconda was properly installed, the version number will appear. In this example, conda version 4.10.3 was successfully installed.

MacOS USERS

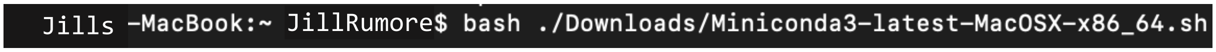
Source: https://conda.io/projects/conda/en/latest/user-guide/install/macos.html

1. Download the Miniconda installer for macOS (https://conda.io/miniconda.html)
2. Click on the Miniconda installer link. By default this file will likely be saved to your Downloads folder.



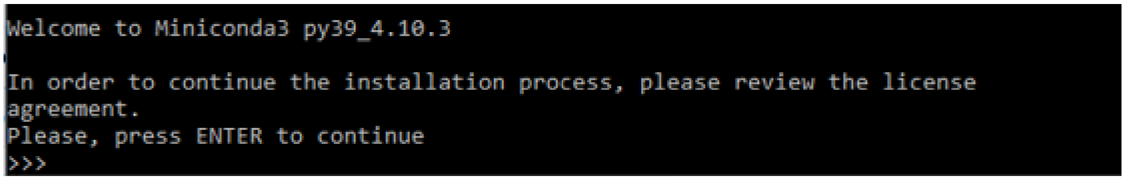
MacOSX	Miniconda3 MacOSX 64-bit bash	786de9721f43e2c7d2803144c635f5f6e4823483536dc141ccd82dbb927cd508

3. Open a terminal and type bash then navigate to the Miniconda3-latest-MacOSX-x86_64.sh file from the Downloads folder in the terminal.



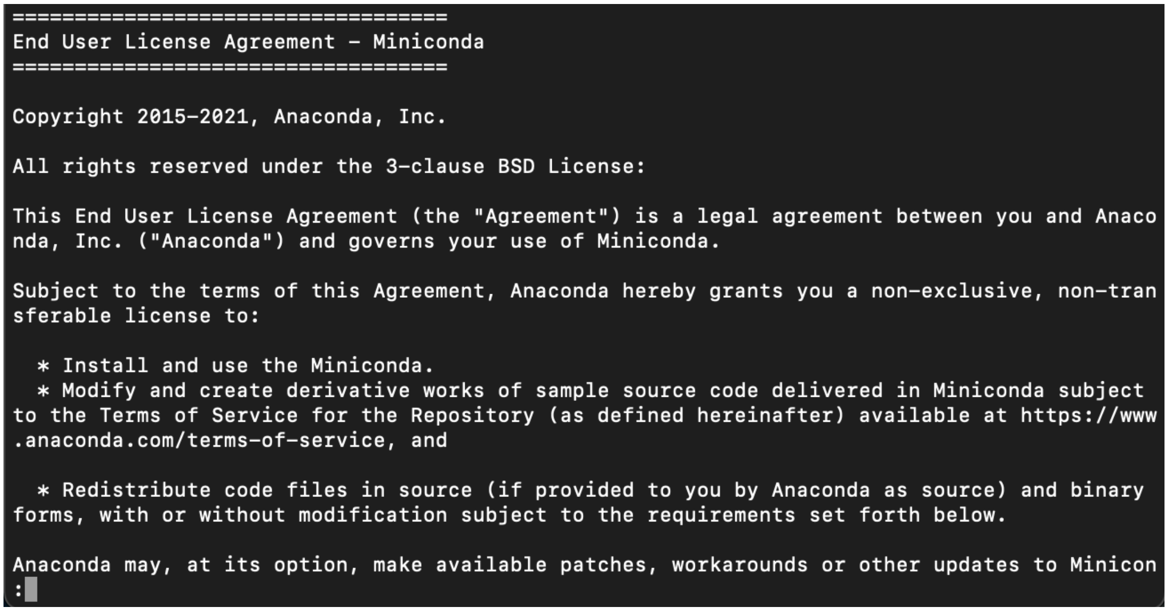
```
 Jills -MacBook:~ JillRumore$ bash ./Downloads/Miniconda3-latest-MacOSX-x86_64.sh
```

4. A terminal prompt will appear asking you to press **ENTER** to start the installation process. Press **ENTER**.



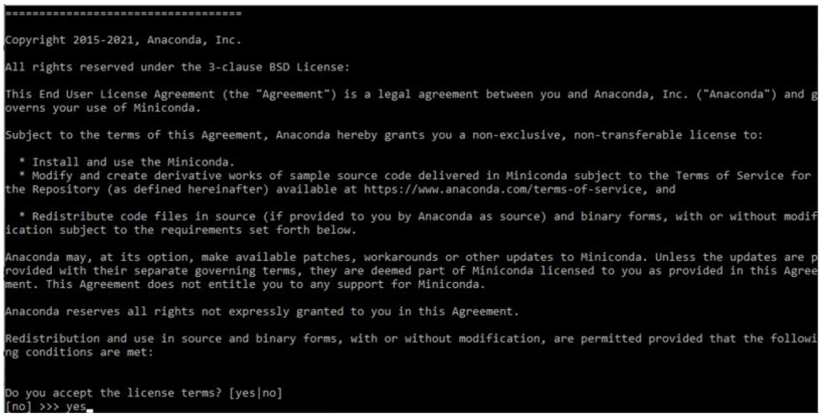
```
Welcome to Miniconda3 py39_4.10.3

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

5. The License Terms will appear. You can read through the agreement by pressing ENTER, or, you can skip it by typing **q** in the terminal.



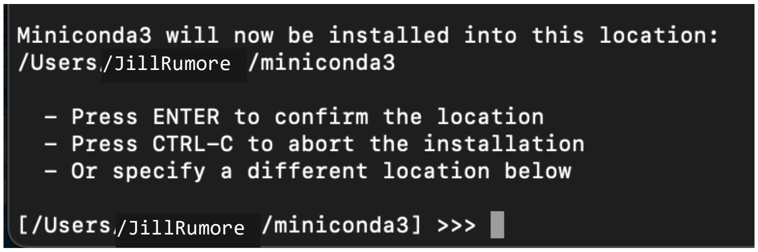
```
===================================
End User License Agreement - Miniconda
===================================

Copyright 2015-2021, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaco
nda, Inc. ("Anaconda") and governs your use of Miniconda.

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-tran
sferable license to:

  * Install and use the Miniconda.
  * Modify and create derivative works of sample source code delivered in Miniconda subject
to the Terms of Service for the Repository (as defined hereinafter) available at https://www
.anaconda.com/terms-of-service, and

  * Redistribute code files in source (if provided to you by Anaconda as source) and binary
forms, with or without modification subject to the requirements set forth below.

Anaconda may, at its option, make available patches, workarounds or other updates to Minicon
:
```

6. You will then be asked if you accept the license terms. Type **yes** to proceed and press **ENTER**.



```
==================================

Copyright 2015-2021, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and g
overns your use of Miniconda.

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

  * Install and use the Miniconda.
  * Modify and create derivative works of sample source code delivered in Miniconda subject to the Terms of Service for
the Repository (as defined hereinafter) available at https://www.anaconda.com/terms-of-service, and

  * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modif
ication subject to the requirements set forth below.

Anaconda may, at its option, make available patches, workarounds or other updates to Miniconda. Unless the updates are p
rovided with their separate governing terms, they are deemed part of Miniconda licensed to you as provided in this Agree
ment. This Agreement does not entitle you to any support for Miniconda.

Anaconda reserves all rights not expressly granted to you in this Agreement.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the followi
ng conditions are met:


Do you accept the license terms? [yes|no]
[no] >>> yes
```

7. By default, an installation folder will be created in the home directory of the system. To accept, press **ENTER.**

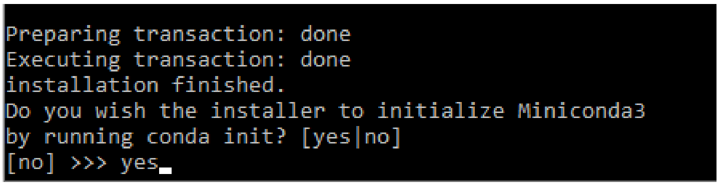


```
Miniconda3 will now be installed into this location:
/Users,/JillRumore /miniconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/Users,/JillRumore /miniconda3] >>>
```

In the screenshot above, miniconda is being installed in /Users/JillRumore/miniconda3.

8. The software will now be installed. This will take a few minutes.

9. Once the installation has finished, it will ask you if you want the installer to initialize Miniconda3. Type **yes** and press **ENTER**. This will allow you to easily work with conda commands in the terminal.



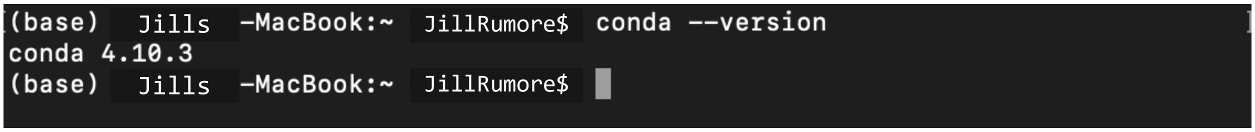
```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

10. Close and re-open the terminal for the changes to take effect.

11. To verify the installation has successfully completed, type the following command in the terminal and press **ENTER**:

```
conda --version
```

Example:



```
(base)  Jills  -MacBook:~  JillRumore$ conda --version
conda 4.10.3
(base)  Jills  -MacBook:~  JillRumore$ 
```

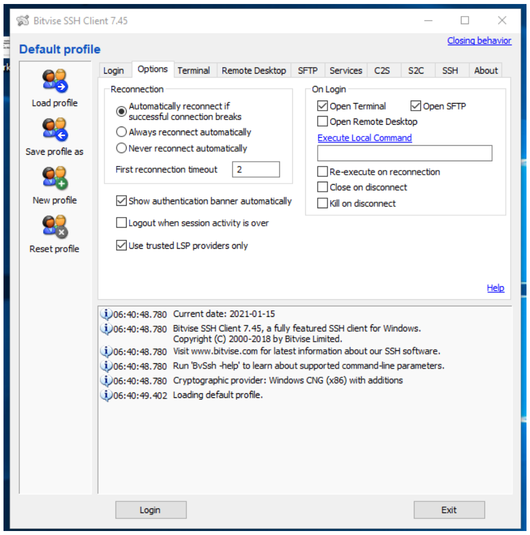
If miniconda was properly installed, the version number will appear. In this example, conda version 4.10.3 was successfully installed.

INSTRUCTIONS FOR PARTICIPANTS WHO <u>HAVE ACCESS</u> TO THE HIGH PERFORMANCE COMPUTING CLUSTER AT NATIONAL MICROBIOLOGY LABORATORY

System administrators have already installed Miniconda for you on Waffles. The instructions described below are for participants who are accessing the cluster from a Government of Canada computer.

A variety of SSH client software are available, and you are free to use whichever platform you are most comfortable with. Below I have provided instructions on how to install the BitVise SSH client.

1. Install Bitvise SSH Client (https://www.bitvise.com/ssh-client-download). Accept the defaults.
2. Once installation has finished, connect to the Science Network VPN using your science network credentials.
3. Open Bitvise SSH Client.
4. In the Login tab, type in the following:
    a. Host: waffles.cscscience.ca
    b. Port: 22
    c. Username: science network user name
5. Click on the Options tab and ensure the following are selected:



Reconnection

● Automatically reconnect if successful connection breaks
○ Always reconnect automatically
○ Never reconnect automatically

First reconnection timeout: 2

☑ Show authentication banner automatically
☐ Logout when session activity is over
☑ Use trusted LSP providers only

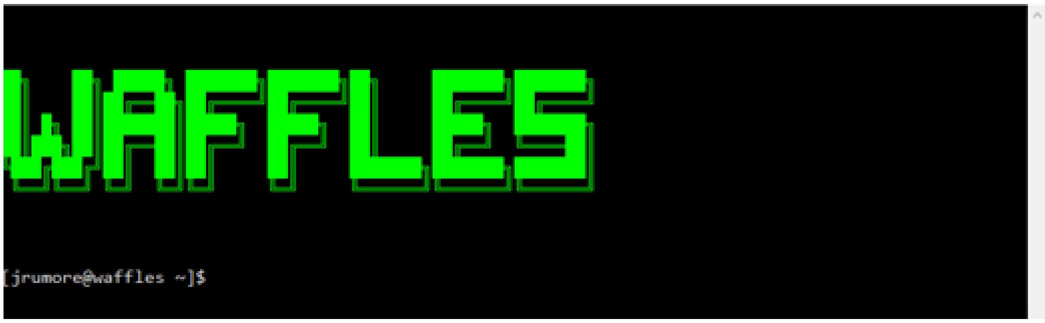
On Login

☑ Open Terminal ☑ Open SFTP
☐ Open Remote Desktop
Execute Local Command
☐ Re-execute on reconnection
☐ Close on disconnect
☐ Kill on disconnect

```
06:40:48.780  Current date: 2021-01-15
06:40:48.780  Bitvise SSH Client 7.45, a fully featured SSH client for Windows.
              Copyright (C) 2000-2018 by Bitvise Limited.
06:40:48.780  Visit www.bitvise.com for latest information about our SSH software.
06:40:48.780  Run 'BvSsh -help' to learn about supported command-line parameters.
06:40:48.780  Cryptographic provider: Windows CNG (x86) with additions
06:40:49.402  Loading default profile.
```

6. Click Login. The User Authentication window will open. Click OK to open the terminal.

7. If accessing Waffles for the first time via the Bitvise SSH client, it may ask you what directory you want to save your *rsa* key to, which is always your home directory. Press *Enter* twice until you see a blank terminal with your username pointing to your home directory (~) as shown below:



```
[jrumore@waffles ~]$
```

SLURM WORKLOAD MANAGER (PARTICIPANTS WITH ACCESS TO WAFFLES)

The slurm workload manager is a job scheduler used at the National Microbiology Laboratory to efficiently manage computational jobs that are submitted to the high performance-computing cluster, commonly referred to as Waffles. This type of system helps streamline jobs, while preventing jobs from running on the head node, which acts as the central managing server that provides workload and job scheduling services across the cluster to worker nodes.

Waffles is a powerful resource, but it is also a SHARED resource. Therefore, as users, we cannot solely be concerned with our own interests, but must also be respectful of the interests of others. Therefore, ALL USERS must practice good cluster etiquette when using the high performance computing cluster.

Submitting Jobs

To submit jobs to Waffles, use the *sbatch* command (note, *srun* can also be used, however, to simplify job submission for participants, this document will only focus on *sbatch*). This will run jobs in the background and appropriately allocate them to the worker nodes and NOT the head node. The *sbatch* command should be used when:

- *Creating conda environments*
- *Installing packages*
- *Updating conda environments*
- *Removing packages and environments*
- *Sharing conda environments (especially if they are very large)*
- *Running packages*

Basic structure

```
sbatch -c NCPUS --mem MEMORY -p PARTION --wrap="COMMAND TO RUN PACKAGE/TOOL"
```

Options

-c : Number of cores (e.g., CPUs – Central Processing Units) requested for the job

- Use 1 core UNLESS the package/ tool offers multithreading*

--mem : Memory requested to run the job

- Units must be specified along with quantity (e.g., 4G = 4 gigabytes)

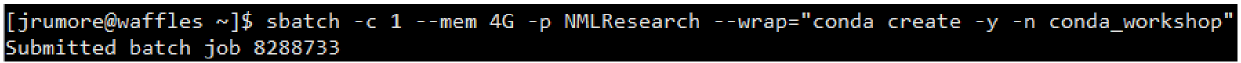
-p : Partition to which the job will be submitted

- OutbreakResponse**
- Reference/Surveillance**
- NMLResearch
- ExternalResearch

--wrap : Script to run the package placed in double quotes (“”)

**Mulithreading will not be discussed in this workshop; for more advanced users.*
***High priority partitions reserved for routine activities; using these partitions for RESEARCH activities is NOT PERMITTED. As the name would imply, partitions NMLResearch (users affiliated with the NML) and ExternalResearch (users outside of the NML), are designated for research related jobs (lower priority).*

Example:



```
[jrumore@waffles ~]$ sbatch -c 1 --mem 4G -p NMLResearch --wrap="conda create -y -n conda_workshop"
Submitted batch job 8288733
```

In the example shown above, the user is creating a new conda environment called conda_workshop using the slurm workload manager. To run the job they are requesting 1 core and 4G of memory and are submitting it through the NMLResearch partition.

Job Management

When submitting jobs to Waffles using *sbatch*, a JobID is created.

```
Submitted batch job JOBID
```

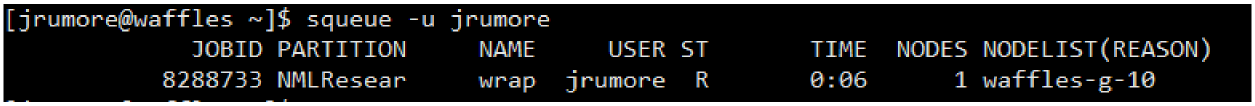
You can check the status of your job by running *squeue*, which is used to view job information for jobs managed by Slurm. To view job information, run the following command:

```
squeue -u username
```

OPTIONS

- `-u:` Science Network user name

Example:

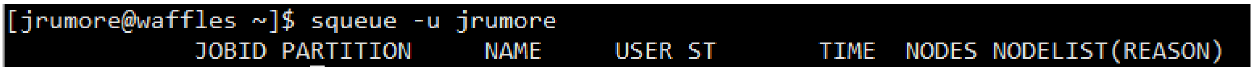


```
[jrumore@waffles ~]$ squeue -u jrumore
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           8288733 NMLResear     wrap  jrumore  R       0:06      1 waffles-g-10
```

When a job is no longer visible in the terminal window after running *squeue*, it means either the job has run to completion or it has failed.

```
squeue -u username
```

Example:



```
[jrumore@waffles ~]$ squeue -u jrumore
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

In the example shown above, the user jrumore has no active jobs running.

The command *sacct* can be used to determine whether the job has successfully completed (or failed). To display information from a finished job, run the following command:

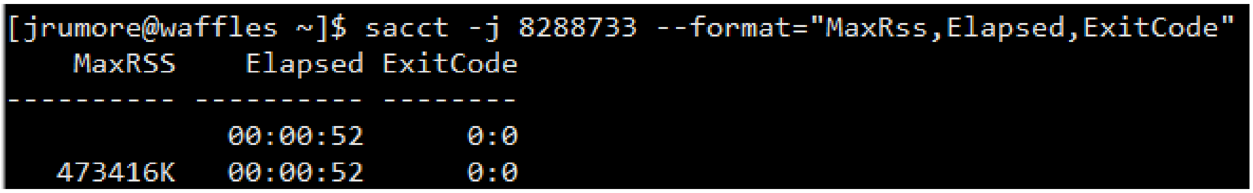
`sacct -j JOBID --format="MaxRss,Elapsed,ExitCode"`

`OPTIONS`

- `-j:` JobID
- `--format:` Comma separated list of fields to display
    - MaxRss - Max memory for ALL tasks
    - Elapsed - Elasped time [DD-HH:MM:SS]
    - ExitCode - Value other than 0 indicates job failure

*Additional options for `--format` can be obtained by running `sacct -e`

Example:



```
[jrumore@waffles ~]$ sacct -j 8288733 --format="MaxRss,Elapsed,ExitCode"
    MaxRSS    Elapsed ExitCode
---------- ---------- --------
             00:00:52      0:0
   473416K   00:00:52      0:0
```

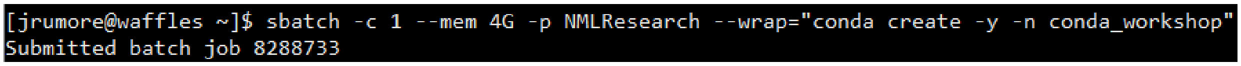
In the above example, JobID 8288733 required 473416 kilobytes (0.47 gigabytes) of memory and took 52 seconds to complete with no errors (ExitCode 0:0).

Optimizing Resource Usage

The information supplied by *sacct* can be very useful for optimizing resource usage when submitting jobs to Waffles. Since it is a shared resource, as users, we want to make sure we are not requesting more memory than we need.

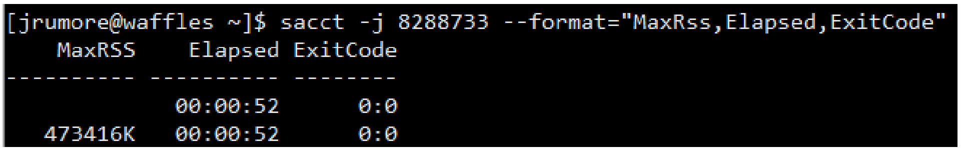
Example:

In the screenshot shown below, the user is trying to create a new conda environment called conda_workshop using the slurm workload manager. To accomplish this task, the user asks for 4G = 4 gigabytes of memory.



```
[jrumore@waffles ~]$ sbatch -c 1 --mem 4G -p NMLResearch --wrap="conda create -y -n conda_workshop"
Submitted batch job 8288733
```

Once the job has completed, the user verifies whether the job ran successfully (ExitCode), as well how much memory it required (MaxRss), and how long it took (Elapsed). From the output, the user confirms the job ran successfully (i.e., ExitCode = 0:0), however, they also realize that the job required less memory than what was requested (i.e., the job only required 473416 kilobytes = 0.47 gigabytes). As a result, the user will make a note to request less memory when creating a new conda environment in the future.



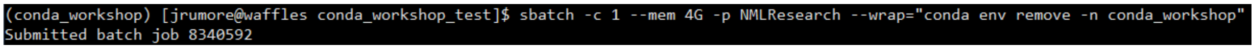
```
[jrumore@waffles ~]$ sacct -j 8288733 --format="MaxRss,Elapsed,ExitCode"
    MaxRSS    Elapsed ExitCode
---------- ---------- --------
             00:00:52      0:0
   473416K   00:00:52      0:0
```

Troubleshooting

For all scripts submitted using *sbatch*, a slurm-JobID.out file is created in the working directory. This is a great resource for troubleshooting failed jobs.

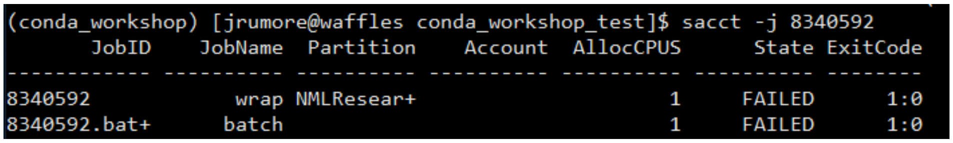
Example:

The user attempted to remove a conda environment using the following command:



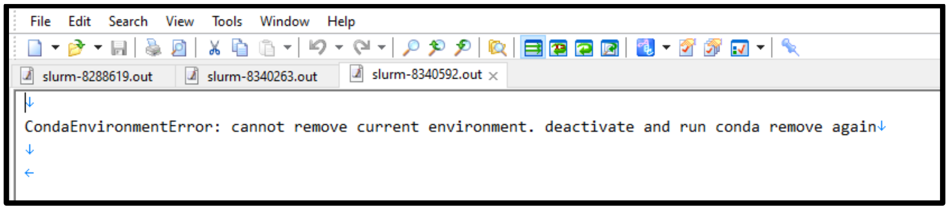
```
(conda_workshop) [jrumore@waffles conda_workshop_test]$ sbatch -c 1 --mem 4G -p NMLResearch --wrap="conda env remove -n conda_workshop"
Submitted batch job 8340592
```

After the job completed, the user wanted to verify whether it ran successfully using the following command:



```
(conda_workshop) [jrumore@waffles conda_workshop_test]$ sacct -j 8340592
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
8340592            wrap NMLResear+                     1     FAILED      1:0
8340592.bat+      batch                                1     FAILED      1:0
```

Based on the output for the completed job, the user realized it failed (i.e., State = Failed and ExitCode = 1.0). To investigate why the job failed, the user navigates to the slurm-JOBID.out file saved in the working directory, which in the above example is located in the conda_workshop_test folder.



```
CondaEnvironmentError: cannot remove current environment. deactivate and run conda remove again
```

Looking at the output file, the user realizes they forgot to deactivate the conda environment before trying to remove the conda environment, causing the job to fail (i.e., note the conda environment is still active in the example above; the environment name conda_workshop is displayed in the terminal prompt). The user then deactivates the conda environment and re-runs the command.

COMMAND LINE BASICS - SUPPLEMENTARY INFO

If you have not taken the Introduction to BASH workshop, it is highly recommended to review the workshop cheatsheet included with the workshop materials as it will provide valuable information on basic BASH commands. A video of the complete Introduction to BASH workshop can be found here: https://www.youtube.com/watch?v=_VUynJ_CBJo.

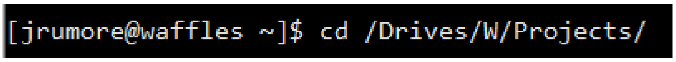
SETTING UP A PERSONAL DIRECTORY AND SUBDIRECTORIES

To create folders (i.e., directories), open a new terminal and navigate to the directory of interest using the following command:

```
cd /path/to/directory
```

- `cd: change directory`

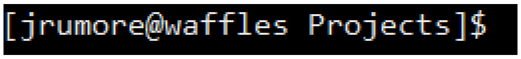
Example:



```
[jrumore@waffles ~]$ cd /Drives/W/Projects/
```

In the example above, the user is navigating to the Projects directory on the W drive. The current working directory will be displayed by your user name.

Example:



```
[jrumore@waffles Projects]$
```

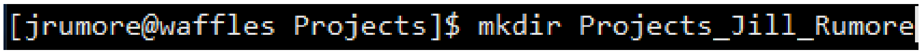
The user jrumore is currently in the Projects directory. The working directory changed from ~ (home directory) shown in the previous example to Projects.

Within the Projects directory (i.e., working directory), we can make another directory using the following command:

```
mkdir DIRECTORYNAME
```

- `mkdir: make directory`

Example:



```
[jrumore@waffles Projects]$ mkdir Projects_Jill_Rumore
```

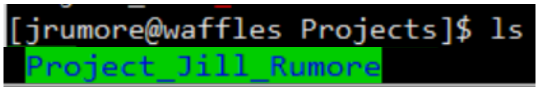
In this example, the user is making a NEW directory called Projects_Jill_Rumore in the Projects directory. Please note, when making a NEW directory DO NOT INCLUDE ANY SPACES in the name; replace spaces with either _ or -. Also, avoid the use of special characters; try to keep the directory names simple, BUT informative.

To verify the new directory has been created, from the current directory type the following command:

```
ls
```

- ls: list files and directories

Example:



```
[jrumore@waffles Projects]$ ls
 Project_Jill_Rumore
```

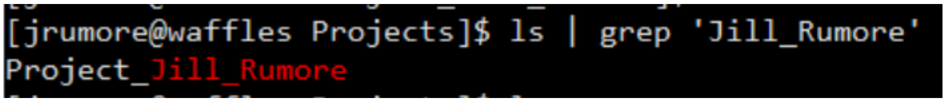
Reminder: A directory that has been successfully created will be highlighted in green. In contrast, files in a directory are indicated in green font (no highlighting).

If you are looking for a specific directory in a directory containing a large number of folders, you can search for it using the *grep* command:

```
ls | grep ‘DIRECTORYNAME’
```

- ls: list files and directories
- grep: search for lines matching a specific pattern

Example:



```
[jrumore@waffles Projects]$ ls | grep 'Jill_Rumore'
Project_Jill_Rumore
```

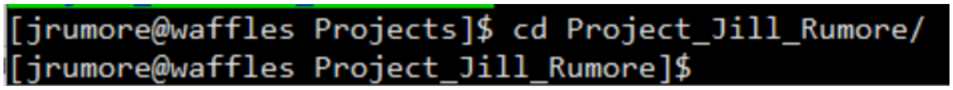
In the example, the user is looking for a folder with the pattern ‘Jill_Rumore’.

To change in to the new directory, run the following command:

```
cd DIRECTORYNAME
```

- cd: change directory

Example:



```
[jrumore@waffles Projects]$ cd Project_Jill_Rumore/
[jrumore@waffles Project_Jill_Rumore]$
```

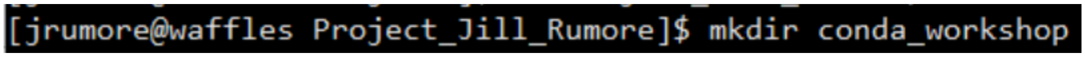
In the example shown, the user is moving from the Projects directory to the Project_Jill_Rumore directory. The current working directory is appended to the user name.

In the newly created project directory, create a new directory using the following command:

```
mkdir DIRECTORYNAME
```

- mkdir: make directory

Example:



```
[jrumore@waffles Project_Jill_Rumore]$ mkdir conda_workshop
```

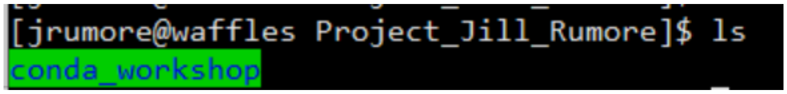
In the example shown above, the user has created a NEW directory called conda_workshop in the Project_Jill_Rumore directory.

To verify the new directory was created, from the current directory type the following command:

```
ls
```

- ls: list files and directories

Example:



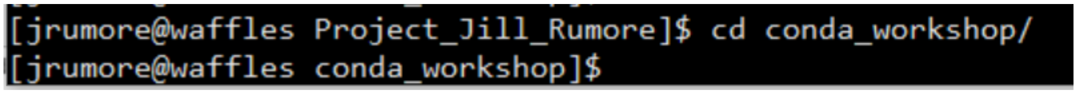
```
[jrumore@waffles Project_Jill_Rumore]$ ls
conda_workshop
```

To change in to the new directory, run the following command:

```
cd DIRECTORYNAME
```

- cd: change directory

Example:



```
[jrumore@waffles Project_Jill_Rumore]$ cd conda_workshop/
[jrumore@waffles conda_workshop]$
```

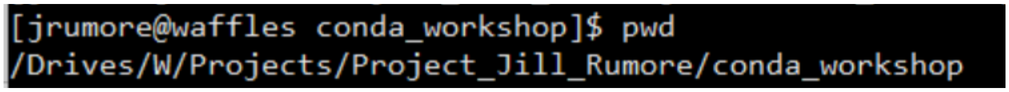
In the example shown, the user is moving from the Project_Jill_Rumore directory into the conda_workshop directory.

To display the full path to the current working directory use the following command:

```
pwd
```

- pwd = print working directory

Example:



```
[jrumore@waffles conda_workshop]$ pwd
/Drives/W/Projects/Project_Jill_Rumore/conda_workshop
```

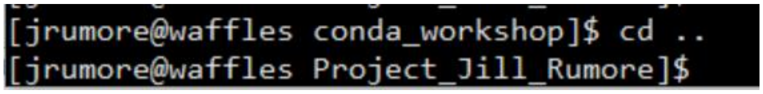
In this example, the path to the current working directory is /Drive/W/Projects/Project_Jill_Rumore/conda_workshop.

To move out of the current directory and back a single directory, use the following command:

```
cd ..
```

- `cd: change directory`
- `..: move back one directory`

Example:



```
[jrumore@waffles conda_workshop]$ cd ..
[jrumore@waffles Project_Jill_Rumore]$
```

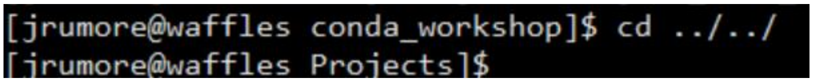
In this example, the user is moving up one directory from conda_workshop to Project_Jill_Rumore.

To move up multiple directories, run the following command:

```
cd ../../
```

- `cd: change directory`
- `../..: move back two directories`

Example:



```
[jrumore@waffles conda_workshop]$ cd ../../
[jrumore@waffles Projects]$
```

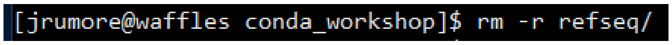
In this example, the user is moving up one directory from conda_workshop to the Projects directory. As a reminder, the current working directory WILL ALWAYS be displayed beside your user name.

To remove a directory and all its contents, use the following command:

```
rm -r DIRECTORYNAME
```

- `rm: remove`
- `-r: remove directories and their contents recursively`

Example:



```
[jrumore@waffles conda_workshop]$ rm -r refseq/
```

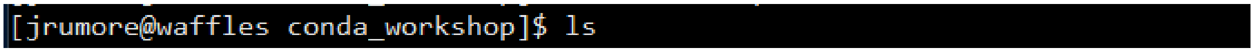
In this example, the user is removing the refseq directory from the conda_workshop directory.

Verify the directory has been removed using the following command:

```
ls
```

- `ls: list files and directories`

Example:



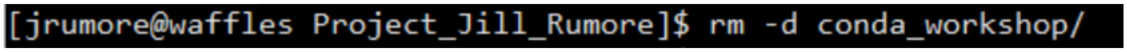
```
[jrumore@waffles conda_workshop]$ ls
```

To remove an empty directory, use the following command:

```
rm -d DIRECTORYNAME
```

- `rm: remove`
- `-d: remove empty directories`

Example:



```
[jrumore@waffles Project_Jill_Rumore]$ rm -d conda_workshop/
```

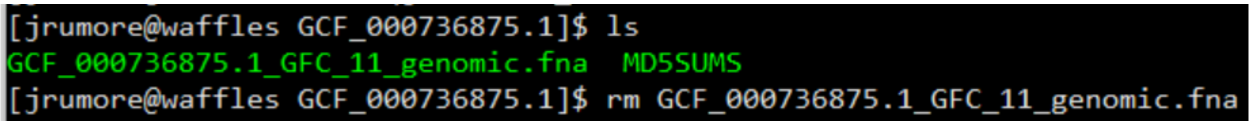
In this example, the user is removing the conda_workshop directory.

To remove a file in a directory, use the following command:

```
rm FILENAME
```

- `rm: remove`

Example:



```
[jrumore@waffles GCF_000736875.1]$ ls
GCF_000736875.1_GFC_11_genomic.fna  MD5SUMS
[jrumore@waffles GCF_000736875.1]$ rm GCF_000736875.1_GFC_11_genomic.fna
```

In this example, the user is removing the GCF_000736875.1_GFC_11_genomic.fna file from the GCF_000736875.1 directory.

HELPFUL RESOURCES

https://www.youtube.com/watch?v=_VUynJ_CBJo

https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments

https://docs.conda.io/projects/conda/en/latest/commands.html

https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf

https://slurm.schedmd.com/overview.html

https://slurm.schedmd.com/sbatch.html