

# Hoffman2 Happy Hour: Anaconda for HPC

Charles Peterson

# Overview

Welcome to Hoffman2 Happy Hour!

The H<sup>2</sup>HH are designed to be short interactive talks that focus on a certain aspect of HPC.



- In this H<sup>2</sup>HH we will go over using Anaconda on Hoffman2
- This information can be applied to other HPC resources



Any suggestions for  
upcoming workshops,  
email me at  
[cpeterson@oarc.ucla.edu](mailto:cpeterson@oarc.ucla.edu)

# Files for this Presentation

This presentation can be found on our UCLA OARC's github repo under **H2HH\_anaconda\_06282022** folder

[https://github.com/ucla-oarc-hpc/hpc\\_workshops](https://github.com/ucla-oarc-hpc/hpc_workshops)

The **slides** folder has this slides.

- PDF format: **H2HH\_anaconda.pdf**
- html format: **html** directory
  - You can open the **H2HH\_anaconda.html** file in your web browser

This presentation was build with [Quarto](#) and RStudio.

- Quarto file: [H2HH\\_anaconda.qmd](#)

# What is Anaconda

- Anaconda is a very popular Python and R distribution.
- Great option for simplifying package management and pipelines.
- Easily install popular Python and R packages.



# Why use Anaconda

- Easy install many python and R packages with simple `conda` commands
- Create isolated python/R environments for different projects
- Checks and solve for possible version conflicts when installing packages
- Share conda env on different systems.
  - Version control!

# Starting Anaconda

On Hoffman2, Anaconda is installed and can be used by loading modules

- See available anaconda versions

```
1 module av anaconda
```

- Load anaconda in your environment

```
1 module load anaconda3/2020.11
```

- Loading the anaconda module will setup anaconda in your environment and ready to be used!

 Important



By using anaconda, you do **NOT** need to load any other python/R modules. The python/R builds will be available via anaconda.

Using other python build might cause conflicts with your anaconda python. (or R)

# Common conda commands

## Creating a new conda environment

```
1 conda create
2 conda create -n myconda
3 conda create -n myconda python=3.9
4 conda create --clone myconda -n myclone
```

## See list of all your environments that you can load

```
1 conda env list
```

## Start (activate) your conda environment

```
1 conda activate myconda
```

## Install packages to your activated conda environment

```
1 conda install python=>3.9
```



Don't run `conda init` on H2. While this does setup conda, it will change `~/.bashrc` and may cause conflicts using different versions/envs.

Loading the anaconda module will already setup conda.

# Creating new conda env

On Hoffman2, after loading the anaconda module, you can create new conda env

```
1 conda create -n myconda
```

Then you can **activate** this new conda env by running

```
1 conda activate myconda
```

Install conda packages on **conda create** command

```
1 conda create -n myconda python=3.9 pandas scipy tensorflow -c conda-forge
```

In this example, it will create a conda env, named **myconda** and will install python (v3.9), scipy and tensorflow all inside the conda env.

This version of `python` is installed locally in your conda env and is different from the builds of python on Hoffman2.

- So you do **NOT** need to load the python module if you installed python via anaconda.

# Installing packages

Install conda with `conda install`

```
1 conda create -n myconda
2 conda activate myconda
3 conda install python=3.9 pandas scipy tensorflow -c conda-forge
```

## Note

The `-c` option in conda is for the “conda channel”. The conda channels are different locations where packages are stored. Examples are ‘conda-forge’, ‘bioconda’, ‘defaults’, etc. Conda will search through the available channels for the request packages to install.

You can use `pip` when you are in a conda env

```
1 conda activate myconda
2 pip3 install scipy
```



## Tip

When using pip/pip3 in a conda env, you do **NOT** need to have `--user`. Using just `pip` will install the package inside the conda env. If you use `--user`, it will install the package outside of the conda env, inside of `~/.local` and may cause conflicts with other python builds or conda env's you have.

# Tips for running on HPC

You maybe familiar with using Anaconda on your local machine. Running on HPC may be different.



# Other Tips

By default, when you install a conda env, it will install it at `~/conda`

You can change this location, esp if you are low in space at `$HOME`

```
1 conda create -p $SCRATCH/mypython python=3.9
2 conda activate $SCRATCH/myptyhon
```

- Some detailed information on using Anaconda on Hoffman2 can be found on our website  
<https://www.hoffman2.idre.ucla.edu/Using-H2/Software/Software.html#anaconda>

# Job examples

```
1  #!/bin/bash
2  #$ -cwd
3  #$ -j y
4  #$ -l h_rt=1:00:00,h_data=5G
5  #$ -pe shared 1
6
7  # load the anaconda module
8  . /u/local/Modules/default/init/modules.sh
9  module load anaconda3/2020.11
10 # Activate the 'myconda' conda env
11 conda activate myconda
12
13 #Running python code
14 python3 test.py > test.out
```

# Searching for anaconda packages

Find software that is available on Anaconda's package repo

- <https://anaconda.org/anaconda/repo>

Here, you can search for software and other packages. It will also explain what conda commands you need to install them inside your conda env.

# Using yml files

You can create a conda file from a `.yml` file

```
1 conda env create -f environment.yml
```

The `environment.yml` file has the packages that are needed to create the conda env.

```
1 name: myconda
2 dependencies:
3   - numpy
4   - pandas
5   - python=3.9
```

An `.yml` file can be created from an existing conda env

```
1 conda activate myconda
2 conda env export > environment.yml
```

This file can be shared with others to reproduce any conda env.

- Creating a `environment.yml` is very useful if you want to make sure you keep the same versions of packages when running anaconda on different HPC resources.

# Installing Anaconda

While Hoffman2 already has Anaconda install, you may need to install yourself if you are using other HPC resources.

Visit <https://repo.anaconda.com/archive/> for all the versions of Anaconda that are available.

```
1 #Download anaconda script for Linux
2 wget https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh
3 #Run Anaconda installer
4 bash Anaconda3-2021.11-Linux-x86_64.sh -p /home/charlie/apps/anaconda/2021.
```

In this example, anaconda is install at  
[/home/charlie/apps/anaconda/2021.11](#)

```
1 source /home/charlie/apps/anaconda/2021.11/etc/profile.d/conda.sh
2 conda create -n myconda python=3.9
3 conda activate myconda
```

# Installing Anaconda



Tip

Don't run `conda init`

Instead, `source /CONDA/PATH/etc/profile.d/conda.sh`

This will setup Anaconda without changing the `~/.bashrc` file



Tip

`Miniconda` is a good alternative to Anaconda.

It is a Minimal installer for conda that is smaller than Anaconda.

# Thank you!

Questions? Comments?

Charles Peterson [cpeterson@oarc.ucla.edu](mailto:cpeterson@oarc.ucla.edu)

<https://charlespeterson3.com>

- Look at for more Hoffman2 workshps at <https://idre.ucla.edu/calendar>
  - Search for **HPC**



