Loggin On to the Coli Server and the Clusters¶

- `$ ssh your_coli_username@login.coli.uni-saarland.de`
- your password is your Coli account password
- to get to a specific server: `ssh your_coli_username@serverName-X` For exampple, `ssh your_coli_username@jones-3`

Refer to the Coli Wiki (Intranet or VPN required)for more details on using the server and clusters.

Installing Anaconda Python and Packages on coli Jones-X¶

These are streamlined instructions based on how previous years' students walked me through that may be useful to others for installing python and other the packages for class on the server.

Things that will be installed:

- Anaconda Python 3.8 2020.07 release for Linux and default packages

The jones-X machines are separate entities, you need a separate install for each jones-X you want to work in.

Directory Organization¶

Skip this if you already have `/home/CE/yourusername` as your home directory when ssh jones-x

1. Create the /home/CE/yourusername directory:
   - The first time in the working directory after ssh to jones-x, `$ pwd` will simply show your directory as: `yourusername@jones-X:/$ /`
   - The `$ ls` command should show directories such as `home var bin root lib etc...`
   - Make `yourusername` directory in the `$ /home/CE/` directory by:
     `$ mkdir /home/CE/yourusername` (yourusername is your coli login username)
   - `/home/CE/yourusername` is now your default home directory
   - **You should NOT install your Python or run stuff here (500MB limits)**
2. Create your username directory in /local: `$ mkdir /local/yourusername`
   - **This is where you want your Python installation (among other things) to go**

Download and install Anaconda¶

1. Make sure you are in your **/local/username** directory on jones-X:
   `username@jones-X:/local/username $`

2. Create a /temp subfolder to hold your Anaconda installer or just download it here. We used the link to the latest release Linux x86_64 command line installer from the Anaconda Website:

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2020.07-Linux-x86_64.sh
```

1. Install with command: `$ bash Anaconda3-2020.07-Linux-x86_64.sh`
2. Keep pressing ENTER to go through the License Agreement after this:

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

1. When the following prompt appears, specify your installation location as followed:

Anaconda3 will now be installed into this location:
/home/CE/username/anaconda3 <-- **should NOT go there**

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

[/home/CE/username/anaconda3] >>> **/local/username/anaconda3**

1. Activate your installation by: `$ source ~/.bashrc`
2. You now have Python 3.8.4 and the default packages that come with Anaconda distribution and the Conda environment manager in your base environment on jones-X

Virtual Environments and Installing Packages¶

As is on your own machine, it's probably a good idea to work in virtual environments to to work with project specific packages.

Creating Virtual Environments with Conda and Required Packages¶

1. `$ conda info --env` should show you're in your base environment:

   # conda environments:
   #
   base * /local/username/anaconda3

2. Create a new virtual environment where you will install packages.

   `$ conda create --name` **`your_env_name`**`anaconda python=3.8` (this will create a virtual environment with all default packages and python version 3.8)

3. `$ conda info --env` should show the environment you created:

   # conda environments:
   #
   base * /local/username/anaconda3
   your_env_name /local/username/anaconda3/envs/**your_env_name**

4. `$ conda activate` **`your_env_name`** to work in the virtual environment .

5. To install additional packages:

   `$ conda install` **`your_package_name_here`**

6. Test your install. To check that your package(s) were installed.

   - `$ conda list -n your_env_name` will list the installed packages in your_env_name environment

     or

   - From **within** the activated virtual environment with the packages installed:

     - start ipython: `$ ipython`
     - You should be able to do the following without getting and error output

       ```
       In [1]: import your_package_module_name
       ```

Other useful commands to work on the Coli server¶

Moving files to/from the server¶

2 ways that I've used are listed below. This is NOT exhaustive, you're welcome to follow your own methods of transfering files to/from remote servers.

1. **Proxy Jump**

   - Local machine to Server:

     ```
     $ scp -o 'ProxyJump [username]@login.coli.uni-saarland.de'
     /path/to/file/on/my/computer [username]@serverNode-X:/path/to/send/in/node
     ```

   - Server to Local machine:

     ```
     $ scp -o 'ProxyJump [username]@login.coli.uni-saarland.de'
     [username]@serverNode-X:/path/to/send/in/node/path/to/file/on/my/computer
     ```

2. **Via GitHub**
   - edit your code on your editor of choice on your machine
   - update to your online GitHub repo
   - wget that repo to the server
   - add/update any new result/output files from server to GitHub repo
   - update your local machine from GitHub

Detaching from server without losing your work¶

Below are screen session commands for working on the server, allowing you to run a script in the session, detach from the session, and logging off from the server while your script is in progress.

1. To start a detachable screen session:

   `$ screen -S your_session_name` (option flags are case sensitive)

2. To detach from this session:

   `$ control + a + d` (on a Mac, this is still the 'control' key not the 'command' key)

3. To re-attach to a session:

   ```
   $ screen -r your_session_name
   ```

4. To list sessions and status:

   ```
   $ screen -ls
   ```

5. To quit a session:

   `$ screen -XS your_session_name_or_ID quit` (option flags are case sensitive)

For full list of options: `screen -h`