

How to setup Python environment on cdmHUB Linux workstation

Setup Python virtual environment

The purpose is to setup a Python virtual environment for code development and testing. This is realized by using the Anaconda environment.

Steps

Step 1: Use Anaconda

Start the Linux Workstation tool. A default terminal should be opened automatically. The command prompt is something like the following:

```
user_name@cdmhub_session:~$
```

To use the Anaconda environment, run

```
use anaconda-6
```

Then for the follow-up question, input y for yes:

```
Do you want to make 'use anaconda-6' persistent? (y/n) y
```

You can type conda to check if Anaconda is used properly. If not, you may need to restart the terminal.

Step 2: Create a virtual environment

Suppose that we want to create a Python 3.8 virtual environment with the name myenv.

In the terminal, run

```
conda create --name myenv python=3.8
```

HOW TO SETUP PYTHON ENVIRONMENT ON CDMHUB LINUX WORKSTATION

Anaconda will list all packages that will be downloaded and installed. Type y (or just press 'enter') to proceed:

```
Proceed ([y]/n)? y
```

Once finished, if there is no error, you can follow the instruction printed on screen to activate the virtual environment:

```
conda activate myenv
```

If this is the first time of activation, an error message could pop up:

```
CommandNotFoundError: Your shell has not been properly configured to use 'conda activate'.
```

Then go to Step 3.

Otherwise, the command prompt should be something like this:

```
(myenv) user_name@cdmhub_session:~$
```

And you can go to Step 4.

Step 3: Configure the shell start-up file

To let Anaconda configure the start-up file for the shell we are using, run

```
conda init bash
```

A short piece of script will be added to the file ~/.bashrc.

Then close and re-open the current shell.

If the command prompt is updated by prepending (base):

```
(base) user_name@cdmhub_session:~$
```

Then this step is done and you can proceed to Step 3.

Otherwise, one more thing needs to be done.

Since the default terminal on the workstation starts from the file `~/.bash_profile`, we need to add scripts to let `~/.bashrc` be executed when start the shell.

Use any text editor you are familiar with to open the file `~/.bash_profile`. Add the following script at the beginning of the file:

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

Then close and re-open the shell. Try activate the virtual environment again and everything should work now. The command prompt should now looks like

```
(myenv) user_name@cdmhub_session:~$
```

Step 4: Install packages

To install Python packages, activate and switch to the desired virtual environment first.

Suppose we want to install [NumPy](#) and [SciPy](#), run

```
conda install numpy scipy
```

Anaconda will list all dependences that will be downloaded and installed. Type y to proceed.

If no error occurs, then done.

Note

You can add as many package names as possible after install.

You can also specify versions for each package using =, e.g., numpy=1.22.

By default, Anaconda will use the latest but compatible versions for installed packages.

Other useful commands

To check all installed packages in the currently active virtual environment, run

```
conda list
```

To delete the virtual environment named myenv, run

```
conda remove --name myenv --all
```

To check all created virtual environments, run

```
conda env list
```

See also

- <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>