

Creating Your First Conda Environment

First, enable the conda command. We recommend adding this to your `.bashrc` file so it happens automatically every time you log in:

Using your favorite text editor, open your `.bashrc` file:

```
vi .bashrc
```

Once in the text editor, move your cursor all the way to the bottom of the page/text and press the letter `o`. This places you in insert mode within `vi`. Paste the following text in and then press the `esc` key to exit insert mode.

```
source /optnfs/common/miniconda3/etc/profile.d/conda.sh
```

Now save the file via `:wq!`

There is a bug in the conda command that requires a one-time workaround. If you already have a `.conda` directory in your home, this step is unnecessary, but it won't hurt anything if you run it again:

```
mkdir -p .conda/pkgs/cache .conda/envs
```

When creating a Python environment, you'll likely want to specify your own Python version at creation time. This is done with the `conda create` command and the `python=version` argument. The version can be

- Nothing: In this case, you'll get the latest version of Python available in the Conda repository.
- A major version (e.g., 2 or 3): You'll get the latest available release of that major version.
- A specific version (e.g., 3.4.3): You'll get exactly that version.

Other packages work the same way. You may specify multiple packages when you create the environment, and of course, you can also add packages later. Here is an example that creates an environment named `mytest` with Python 3.9.5 and the latest version of the `numpy` package:

```
conda create --name discovery_class python=3.9.5 numpy
```

Be sure to activate your new environment before trying to use it:

```
[john@andes8 discovery]$ conda activate discovery_class
```

To verify we are using the new version of python we can run `python -V`

```
(mytest) [john@andes8 discovery_class_website]$ python -V  
Python 3.9.5
```