**Setting Python 3.8**

Learn how to install Python 3 using [pyenv](https://github.com/pyenv/pyenv) on a CentOS 7 that has code-server pre-installed to provide a full development environment.

*Note:* This course uses Python 3.8 and you will *definitely* run into issues if you are using Python < 3.6.

Picking the Right Cloud Playground Image

If you plan on following along with the course on your local workstation, you'll want to make sure that you have a good development environment setup. But, if you want to follow along exactly with the course, then you'll want to create a Cloud Playground server (2 or 3 units) using the CentOS 7 w/ code-server image. This image will give us a server with [code-server](https://github.com/cdr/code-server) pre-installed (VS Code running on the server and accessible through the browser).

Using code-server to Program on the Server

By using the domain name of the server followed by port 8080, we can access code-server from our browser while having a full development environment and terminal available to us. We'll be redirected to the code-server page via HTTPS. However, depending on our browser, we'll need to click the advanced link to acknowledge that this certificate is self-signed.

Installing pyenv

Installing Python from source can be a great learning experience, but it is a little tedious. For this course, we're going to install [pyenv,](https://github.com/pyenv/pyenv) which will allow us to install and switch between multiple different Python versions more easily.

To get started, we need to make sure that we have some development dependencies installed so that we can pull down the pyenv repository. We're using the --skip-broken flag because the CentOS 7 w/ code-server playground image already has Git installed. But, if you're using a different image, you can install Git using the package manager for that system.

*Note:* You can get to the terminal in VS by clicking the hamburger icon (3-line menu bar) on the top left.

sudo yum install -y --skip-broken git gcc zlib-devel bzip2-devel readline-devel sqlite-devel

We also need to clone the [pyenv](https://github.com/pyenv/pyenv) repository:

$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv

For pyenv to be useful, we'll need to set a few environment variables and run a command when our shell is loading. We'll add those to our ~/.bashrc file so that it's set as soon as our shell is initialized:

$ echo 'export PYENV\_ROOT="$HOME/.pyenv"' >> ~/.bashrc

$ echo 'export PATH="$PYENV\_ROOT/bin:$PATH"' >> ~/.bashrc

$ echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n eval "$(pyenv init -)"\nfi' >> ~/.bashrc

Before we can use pyenv, we'll need to reload our shell:

$ exec $SHELL

Finally, let's install Python 3.8.2:

$ pyenv install 3.8.2

We can check and switch between versions of Python using pyenv. To see the versions available to us, we'll use the pyenv versions command:

$ pyenv versions

\* system (set by /home/cloud\_user/.pyenv/version)

3.8.2

To change our active version, we'll use pyenv shell <VERSION>:

$ pyenv shell 3.8.2

$ python --version

Python 3.8.2

We also have executables for python3 and python3.8 that we can use. To make it apparent what version is being used throughout the course, we'll use the python3.8 executable.

Since we're going to be using Python 3.8.2 for this whole course, we'll initiate 3.8.2 as our default Python version in our ~/.bashrc file:

~/.bashrc:

# previous lines omitted

if command -v pyenv 1>/dev/null 2>&1; then

eval "$(pyenv init -)"

pyenv shell 3.8.2

fi

Don't forget to save before exiting!

Upgrade Pip

The version of pip that we have might be up-to-date, but it's good practice to try to update it after the installation. Let's update that now:

$ pip3.8 install --upgrade pip

Collecting pip

Downloading https://files.pythonhosted.org/packages/57/36/67f809c135c17ec9b8276466cc57f35b98c240f55c780689ea29fa32f512/pip-20.0.1-py2.py3-none-any.whl (1.5MB)

|????????????????????????????????| 1.5MB 3.1MB/s

Installing collected packages: pip

Found existing installation: pip 19.2.3

Uninstalling pip-19.2.3:

Successfully uninstalled pip-19.2.3

Successfully installed pip-20.0.1

To use the improved REPL:

pip3.8 install bpython