

## CHAPTER 2.2

### Installing Python 3.7 on a Cloud Playground Server

Learn how to install Python 3 using [pyenv](#) on CentOS 8 with code-server pre-installed to provide a full development environment.

**Note:**

*This course uses Python 3.7 and you will definitely run into issues if you are using Python < 3.7.*

#### Picking the Right Cloud Playground Image

If you plan on following along the course on your local workstation, you'll want to make sure to have a good development environment set up. But, if you want to follow along exactly with the course, then you'll want to create a Cloud Playground server (medium: 3 units) using the "CentOS w/ code-server" image. This image will give us a server with code-server pre-installed (VS Code running on the server and accessible through the browser). Be sure to log in the server and set your own password first.

#### Using code-server to Program on the Server

By using the public IP address (or domain name) of the server on port 8080, we can access [code-server](#) from our browser which has a full development environment with a terminal available to us.

We'll be redirected to the page being served over HTTPS and, depending on our browser, we'll need to click a few buttons acknowledging the certificate is self-signed.

You can acquire your password for your server by logging in via SSH to the host and navigating to the directory provided at `~/.config/code-server/config.yaml`.

#### Installing pyenv

Installing Python from source can be a great learning experience, but it is a little tedious.

Instead, for this course, we're going to install **pyenv** which will allow us to install and switch between multiple different Python versions more easily.

1. To get started, we need to make sure we have some development dependencies installed, so we can pull down the pyenv repository.

```
$ sudo yum install openssl-devel -y
```

2. Now we need to install the pyenv tool (NOTE: This will take a few moments to complete.).

```
$ curl https://pyenv.run | bash
```

3. 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 `~/.profile` and `~/.bashrc` file, so this is actioned as soon as our shell is initialized.

```
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(pyenv init --path)"' >> ~/.bashrc
$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc
```

4. Exit the terminal before proceeding by typing:

```
$ exit
```

5. Open a new terminal to test if our settings are loaded by typing (NOTE: You should see a list of commands for *pyenv*):

```
$ pyenv
```

6. Finally, let's install Python 3.7.6. (NOTE: This will take a few moments to complete.)

```
$ pyenv install 3.7.6
```

7. Now 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)
  3.7.6
```

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

```
$ pyenv shell 3.7.6
$ python --version
Python 3.7.6
```

We also have an executable for `python3` and `python3.7` available for use. To make it apparent what version is being used throughout the course, you'll see that the commands use the `python3.7` executable.

### Upgrade Pip

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

```
$ pip3.7 install --upgrade pip
Collecting pip
  Downloading
https://files.pythonhosted.org/packages/ca/31/b88ef447d595963c01060998cb329251648acf4a067721b0452c45527eb8/pip-21.2.4-py3-none-any.whl (1.6MB)
|████████████████████████████████████████████████████████████| 1.6MB 18.9MB/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-21.2.4
```

**Note:**

*If you chose to install the Python extension within code-server, be sure to press the blue button to reload. Click Install at the bottom right-hand corner to install Pylint. Pylint is a source-code, bug, and quality checker for the Python programming language.*

**Handy to know:**

You no longer need to install anything for the cli command "code-server" to work.  
You can test that you can open files via the cli by making a blank file

```
$ touch test.txt
```

Now you can test by opening a file in the editor.

```
$ code-server test.txt
```

And that is it, you are all ready to get started with this course.