Lab - Explore Python Development Tools (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Answers: 3.1.12 Lab - Explore Python Development Tools

Objectives

Part 1: Launch the DEVASC VM

Part 2: Review the Python Installation

Part 3: PIP and Python Virtual Environments

Part 4: Sharing Your Virtual Environment

Background / Scenario

This course has some basic expectations for students, that they have some background in Python and some hands-on experience with programming. Lab practice with Python gives your fingers the "muscle memory" to work more efficiently in Python. However, initially it is important for you to know how to maintain a Python development environment.

In this lab, you review Python installation, PIP, and Python virtual environments.

Required Resources

- 1 PC with operating system of your choice
- Virtual Box or VMWare
- DEVASC Virtual Machine

Instructions

Part 1: Launch the DEVASC VM

If you have not already completed the **Lab - Install the Virtual Machine Lab Environment**, do so now. If you have already completed that lab, launch the **DEVASC VM**.

Part 2: Review the Python Installation

These commands give you basic methods for learning more about the local Python environment.

a. In the DEVASC VM, you can check the version of Python already installed with the **python3 -V** command.

```
devasc@labvm:~$ python3 -V
Python 3.8.2
devasc@labvm:~$
```

b. To see the directory for the local Python environment, use the **which python3** command.

```
devasc@labvm:~$ which python3
/usr/bin/python3
devasc@labvm:~$
```

Part 3: PIP and Python Virtual Environments

PIP stands for Pip Installs Packages. Many people first learn about PIP and start using **pip3 install** commands on the system wide Python installation. When you run **pip3 install** command on your system, you might introduce competing dependencies in your system installation that you may or may not want for all Python projects. Therefore, the best practice is to enable a Python virtual environment. Then install only the packages that are needed for the project in that virtual environment. That way, you know exactly which packages are installed in a given setting. You can switch those package dependencies easily when switching to a new virtual environment, and not break or cause problems due to competing versions of software.

To install a Python virtual environment, use the **venv** tool in Python 3 and then activate the virtual environment, as shown in the following steps.

Step 1: Create a Python 3 virtual environment.

a. Inside the DEVASC VM, change the **labs/devnet-src/python** directory.

```
devasc@labvm:~$ cd labs/devnet-src/python/
devasc@labvm:~/labs/devnet-src/python$
```

b. Enter the following command to use the **venv** tool to create a Python 3 virtual environment with the name **devfun**. The **-m** switch tells Python to run the **venv** module. The name is chosen by the programmer.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devfun
devasc@labvm:~/labs/devnet-src/python$
```

Step 2: Activate and test the Python 3 virtual environment.

a. Activate the virtual environment. The prompt changes to indicate the name of the environment you are currently working in, which is **devfun** in this example. Now when you use the **pip3 install** command form here, the system will only install packages for the active virtual environment.

```
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

b. Run the **pip3 freeze** command to verify that there are no additional Python packages currently installed in the **devfun** environment.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

c. Now install the Python **requests** package within the **devfun** environment.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 install requests
Collecting requests
  Downloading requests-2.23.0-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 290 kB/s
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
  Downloading urllib3-1.25.9-py2.py3-none-any.whl (126 kB)
     |████████████████████████████████| 126 kB 1.7 MB/s
Collecting idna<3,>=2.5
  Downloading idna-2.9-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 18.3 MB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2020.4.5.1-py2.py3-none-any.whl (157 kB)
     |████████████████████████████████| 157 kB 19.8 MB/s
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
```

```
     |████████████████████████████████| 133 kB 59.2 MB/s
Installing collected packages: urllib3, idna, certifi, chardet, requests
Successfully installed certifi-2020.4.5.1 chardet-3.0.4 idna-2.9 requests-2.23.0
urllib3-1.25.9
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

d. Re-enter the **pip3 freeze** command to see the packages now installed in the **devfun** environment.

Note: Your list of packages and version numbers may be slightly different.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2020.4.5.1
chardet==3.0.4
idna==2.10
requests==2.24.0
urllib3==1.25.9
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

e. To deactivate the virtual environment and go back to your system, enter the **deactivate** command.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$
```

Step 3: Check the current packages installed in the system environment.

a. Enter the system wide **python3 -m pip freeze** command to see what packages are installed in the system environment.

Note: Because Python 3 is invoked with the following command, you only use **pip** instead of **pip3**.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze
ansible==2.9.4
apache-libcloud==2.8.0
appdirs==1.4.3
argcomplete==1.8.1
astroid==2.3.3
bcrypt==3.1.7
blinker==1.4
certifi==2019.11.28
<output omitted>
xmltodict==0.12.0
zipp==1.0.0
devasc@labvm:~/labs/devnet-src/python$
```

b. If you want to quickly find the version of a package installed, pipe the output to the **grep** command. Enter the following to see the version of the requests package currently installed.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze | grep requests
requests==2.22.0
requests-kerberos==0.12.0
requests-ntlm==1.1.0
requests-toolbelt==0.9.1
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$
```

Part 4: Sharing Your Virtual Environment

The output of the **pip3 freeze** command is in a specific format for a reason. You can use all the dependencies listed so that other people who want to work on the same project as you can get the same environment as yours.

A developer can create a requirements file, such as **requirements.txt**, by using the **pip3 freeze > requirements.txt** command. Then another developer can, from another activated virtual environment, use this **pip3 install -r requirements.txt** command to install the packages required by the project.

a. Re-activate the **devfun** virtual environment.

```
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:
```

b. Send the output of the **pip3 freeze** command to a text file called **requirements.txt**.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze >
requirements.txt
```

c. Deactivate the **devfun** virtual environment. You can use the **ls** command to see that the **requirements.txt** file is in the /python directory.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$ ls
devfun       file-access-input.py  if-acl.py       requirements.txt
devices.txt  file-access.py        if-vlan.py      while-loop.py
devnew       hello-world.py        person-info.py
```

d. Create and activate a new Python virtual environment called **devnew**.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devnew
devasc@labvm:~/labs/devnet-src/python$ source devnew/bin/activate
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

e. Use the **pip3 install -r requirements.txt** command to install the same packages that are installed in the **devfun** virtual environment.

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 install -r
requirements.txt
Collecting certifi==2020.4.5.1
  Using cached certifi-2020.4.5.1-py2.py3-none-any.whl (157 kB)
Collecting chardet==3.0.4
  Using cached chardet-3.0.4-py2.py3-none-any.whl (133 kB)
Collecting idna==2.9
  Using cached idna-2.9-py2.py3-none-any.whl (58 kB)
Requirement already satisfied: pkg-resources==0.0.0 in ./devnew/lib/python3.8/site-
packages (from -r requirements.txt (line 4)) (0.0.0)
Collecting requests==2.23.0
  Using cached requests-2.23.0-py2.py3-none-any.whl (58 kB)
Collecting urllib3==1.25.9
  Using cached urllib3-1.25.9-py2.py3-none-any.whl (126 kB)
Installing collected packages: certifi, chardet, idna, urllib3, requests
Successfully installed certifi-2020.4.5.1 chardet-3.0.4 idna-2.9 requests-2.23.0
urllib3-1.25.9
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

f. When entering **pip3 freeze** in the **devnew** environment, you should see the following output.

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2020.4.5.1
chardet==3.0.4
idna==2.9
pkg-resources==0.0.0
requests==2.23.0
urllib3==1.25.9
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

g. Deactivate the **devnew** virtual environment.

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$
```