Development Environment

Installation

Disclaimer

Set up development environment: Linux

In this lab you'll find walkthroughs on how to install a set of common development tools on an Ubuntu Desktop 18.0.4 workstation or latest.

Objectives

1. Install a basic development toolset on your local workstation
2. Verify the tools are all working as expected

Step 1: Ubuntu specific preparation

To use Linux as your development environment, you should have a good graphical desktop interface setup.

1. The standard/default Ubuntu desktop environment for Ubuntu 18.04 LTS is Gnome Shell, but...this is Linux - you have choices!
2. Now your workstation should startup and provide a GUI login to a desktop environment
3. Install some basic Linux tools and utilities

```
sudo apt install curl
sudo apt install libssl-dev
 # (equivalent to openssl-dev on other distributions)
```

   (wget is already installed, so we do not need to install it)
4. Install the typical developer tools and utilities (For example the GCC C/C++ compiler):

```
sudo apt install build-essential
```

Step 2: Source control systems

Git

Installation:

1. Git needs to be installed as a separate package, but is easily done:

```
sudo apt install git
```

verification

Let us verify Git is working as expected:

1. Open a terminal
2. From within the terminal, run:

```
git --version
```

You should get output indicating the version of git installed:

```
# Example output
git version 2.17.1
```

3. Attempt to clone a repository from GitHub:

```
git clone https://github.com/RSTForum/knowledgebase.git
# Expected Output
Cloning into
remote: Counting objects: 11, done.
remote: Total 11 (delta 0), reused 0 (delta 0), pack-reused 11
Unpacking objects: 100% (11/11), done.
```

Step 3: Python and Node.js

Python 3 is the recommended version of Python.

Note: Python 2 is no longer supported. As of January 1st, 2020 no new bug reports, fixes, or changes will be made to Python 2. You can read Python 2 instructions in the appendix to this Lab, but it is **not** required or recommended.

Installing Python

You may already have Python 3 installed. Verify your Python installation by running this command at the terminal prompt:

```
python3


#Expected output
Python 3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

1. If Python3 is not installed, you can install it:

   ```
   sudo apt install python3
   ```

2. Verify Python3 was correctly installed:

   ```
   cd ~
   python3 -V
   # Expected Output
   Python 3.6.9
   ```

Python virtual environments

Before leaving the Python setup, you need to know how to create a Python virtual environment. Python virtual environments are a method of creating isolated "environments" where specific versions of Python can be installed along with independent sets of libraries and dependencies.

Virtual environment usage is very common and is recommended practice when working in Python, and most DevNet labs encourage you to create and work within virtual environments.

1. First download and install the Python 3 virtual environment package.

```
sudo apt install python3-virtualenv
```

2. Create a Python 3 virtual environment using the `virtualenv` module.

```
python3 -m venv py3-venv
```

3. Now "activate" the environment. Look for the name of the virtual environment to be enclosed in parenthesis after activation.

```
source py3-venv/bin/activate
# Expected Output
(py3-venv) [timmc@ubuntu ~]$
```

4. Now verify that `python` is now linked to Python 3.

```
python -V
Python 3.6.9
```

5. Deactivate the virtual environment.

```
Deactivate
```

Step 4: Nodejs

There are two different Node.js options when installing. It's possible to use the `distro-stable` package with APT, or to branch out and use version-specific versions of Node.js using NVM, the Node Version Manager.

For our lab purposes, the `distro-stable` version will probably be fine since it is 8.x.

1. Install nodejs

```
sudo apt install nodejs
```

2. Install NPM (Node Package Manager)

```
sudo apt install npm
```

3. Check the version installed.

```
nodejs -v
#Expected output
v8.10.0
```

Step 5: Text editors and IDE

Atom Installation

1. Because Atom only offers install via Debian package in Ubuntu (.deb) we can't use `apt` to do the install. Luckily, we can use `snap` installer instead. This should already be installed, but if not we can install it:

```
sudo apt install snapd
```

2. Using snap, installing Atom is, well, a snap:

```
sudo snap install atom --classic
```

Verification

1. Once the program is finished installing, we can launch it by typing `atom` from the terminal, or find it under Applications

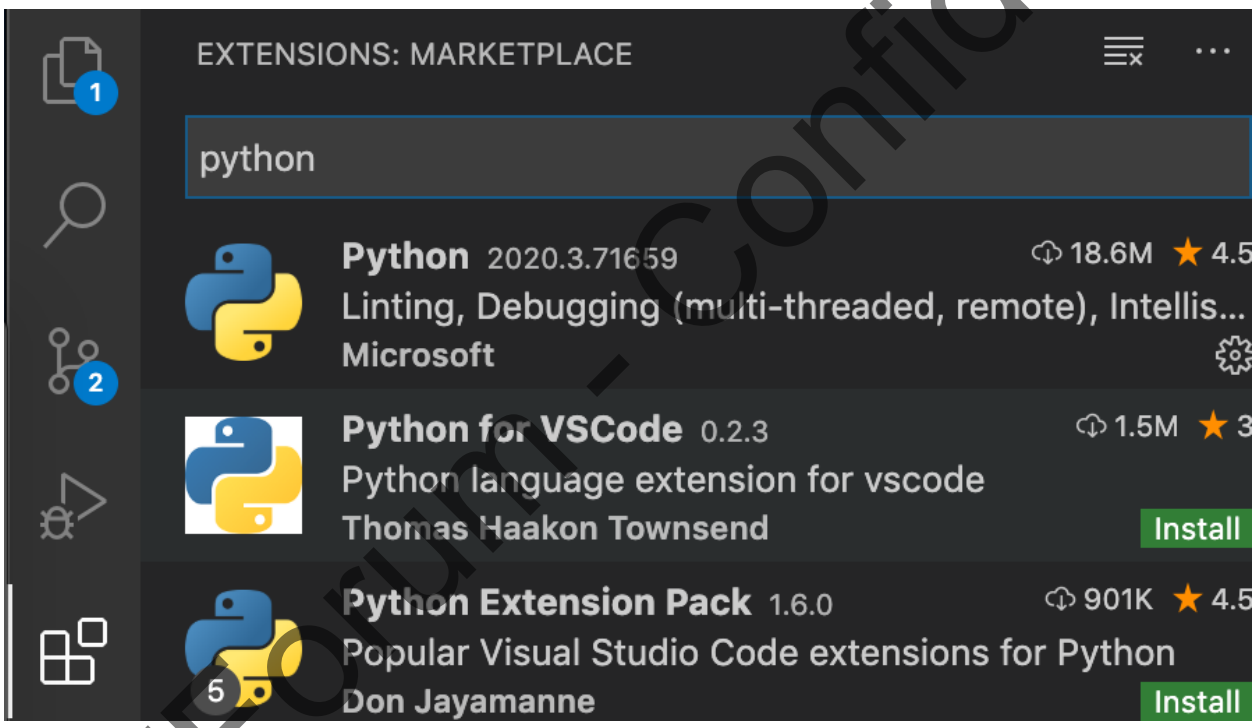
Visual Studio Code

Installation

1. VS code can also been installed from the Snap store:

```
sudo snap install atom --classic
```

2. Open VS Code to display the main interface
3. In the upper left, select the **Extensions** view, search for "`python`" and install the top hit, i.e. the "Python" extension (by Microsoft)



All done!

Step 6: Development tools and clients

Postman Installation

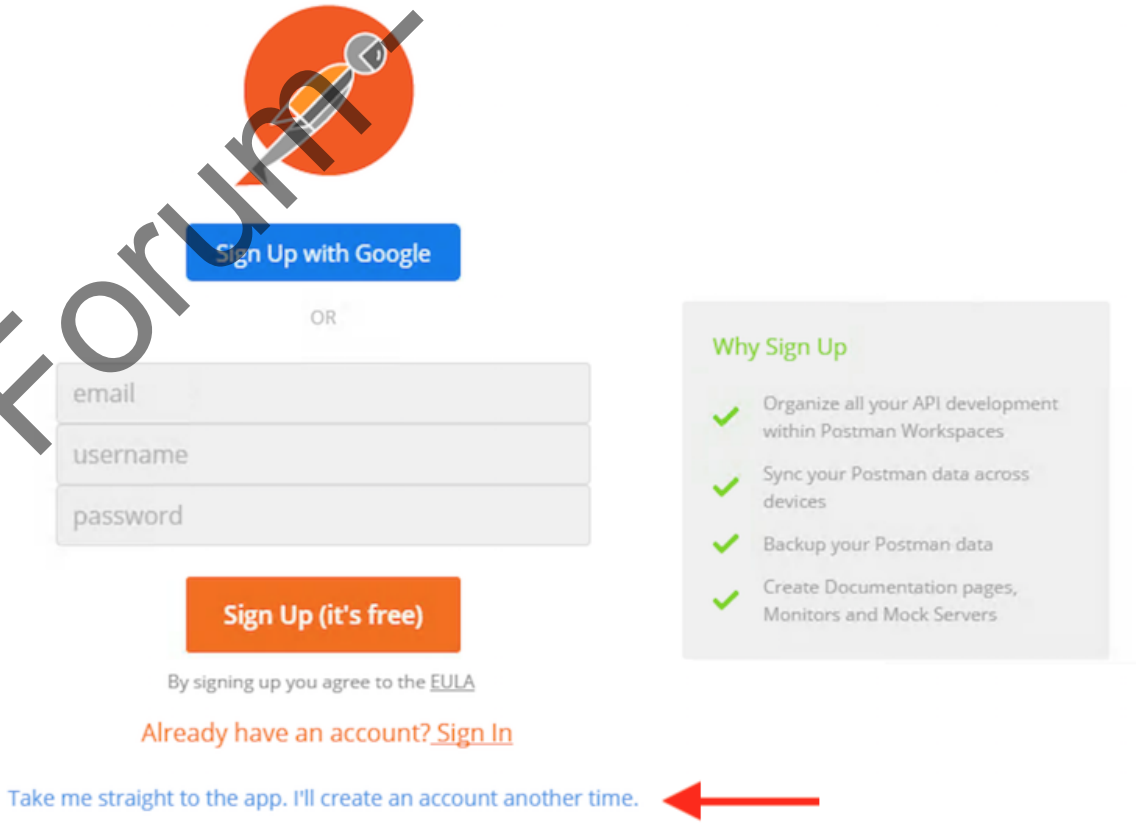
1. Postman has a complicated install on some distributions, but we can leverage the snap installer that we used for Atom to install Postman as well.

```
sudo snap install postman
```

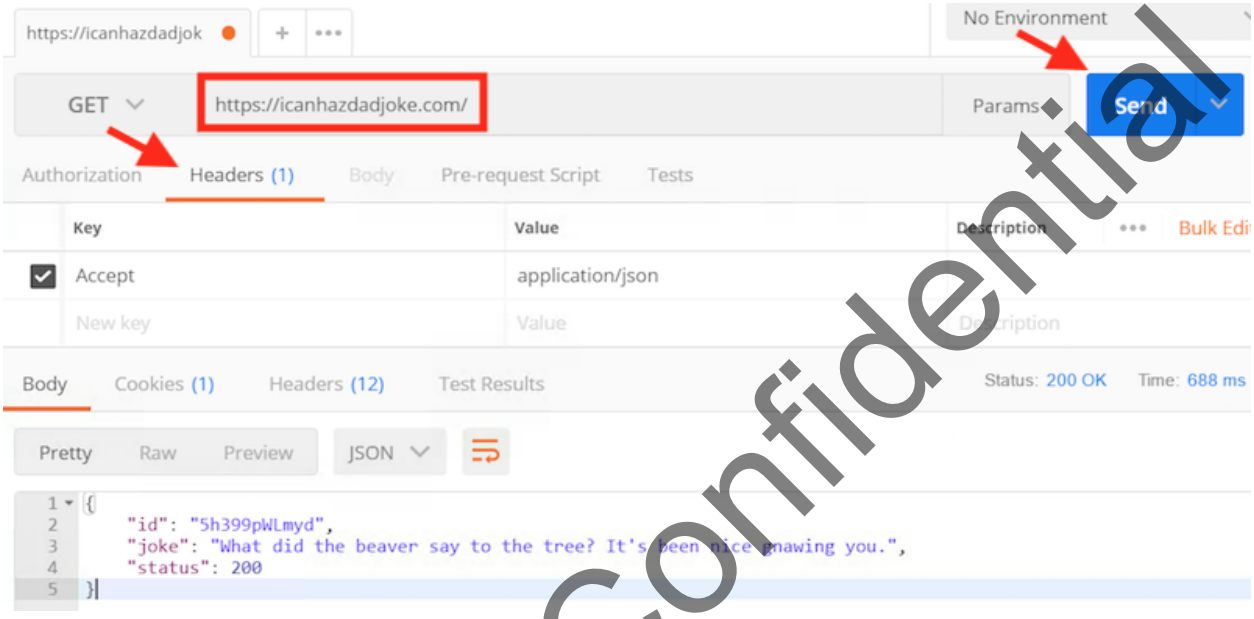
2. Now you can activate Postman either from the terminal window by using the `postman` command, or find it under Applications.

Verification

1. Once the installation completes, find Postman in the application launcher.
2. Postman will open and allow you to sign-up or sign-in. You do **NOT** need to sign in to use Postman, you can simply click the "Take me straight to the app. I'll create an account another time." link to bypass login.



3. Test that you can make REST API calls with Postman with this fun "Dad Joke" API.
   Enter `https://icanhazdadjoke.com/` into the address bar. Click the "Headers" tab and add an entry for `Accept` with a value of `application/json`. Then click "Send" and enjoy your joke :-)



ngrok Installation

1. ngrok is another app we can get from the handy Snap store:

```
snap install ngrok
```

Google Chrome Installation

1. Download the `.deb` Google Chrome installer from google.com/chrome
2. Install it using `apt`:

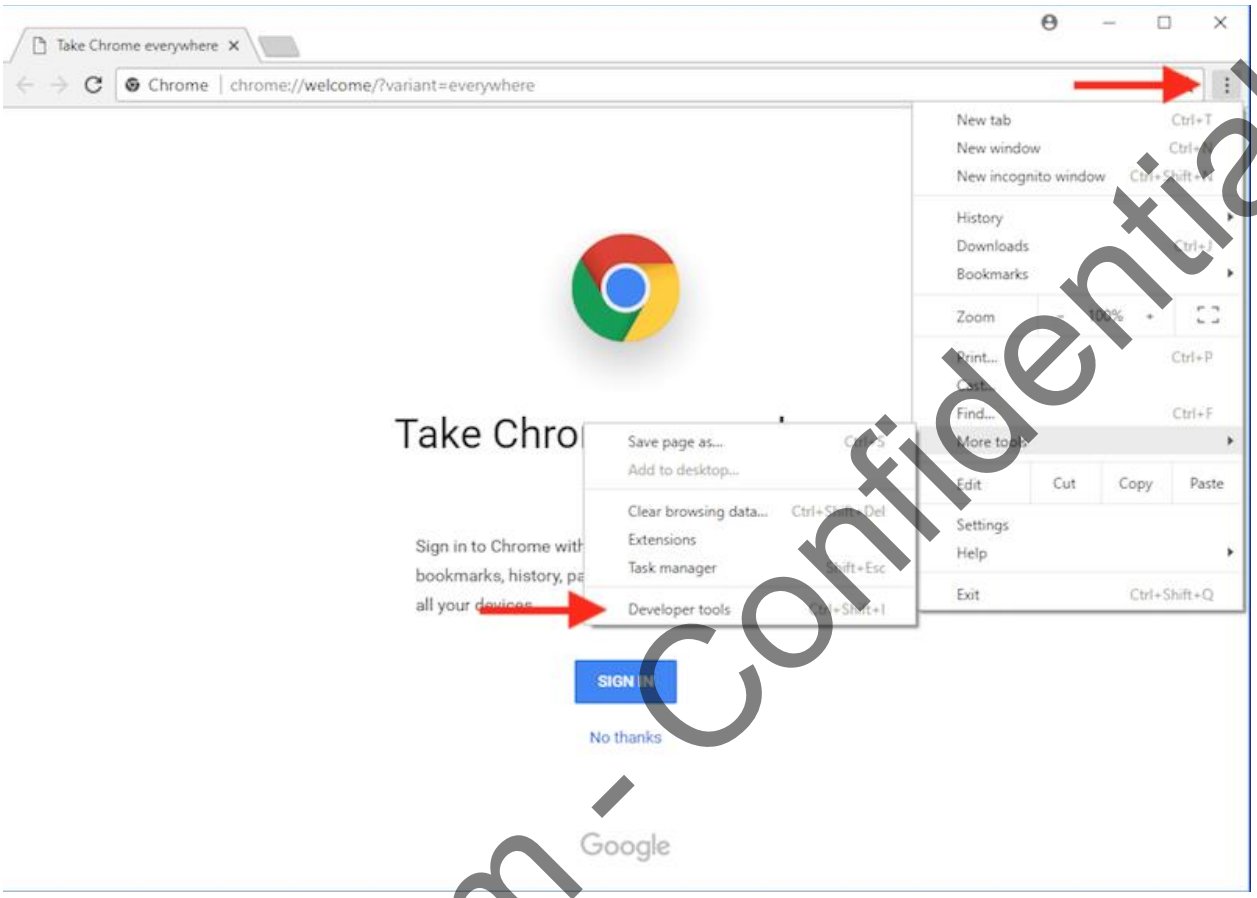
```bash
sudo apt install google-chrome-stable_current_amd64.deb
```

3. Start Google Chrome from the terminal with the `google-chrome-stable` command or from Applications

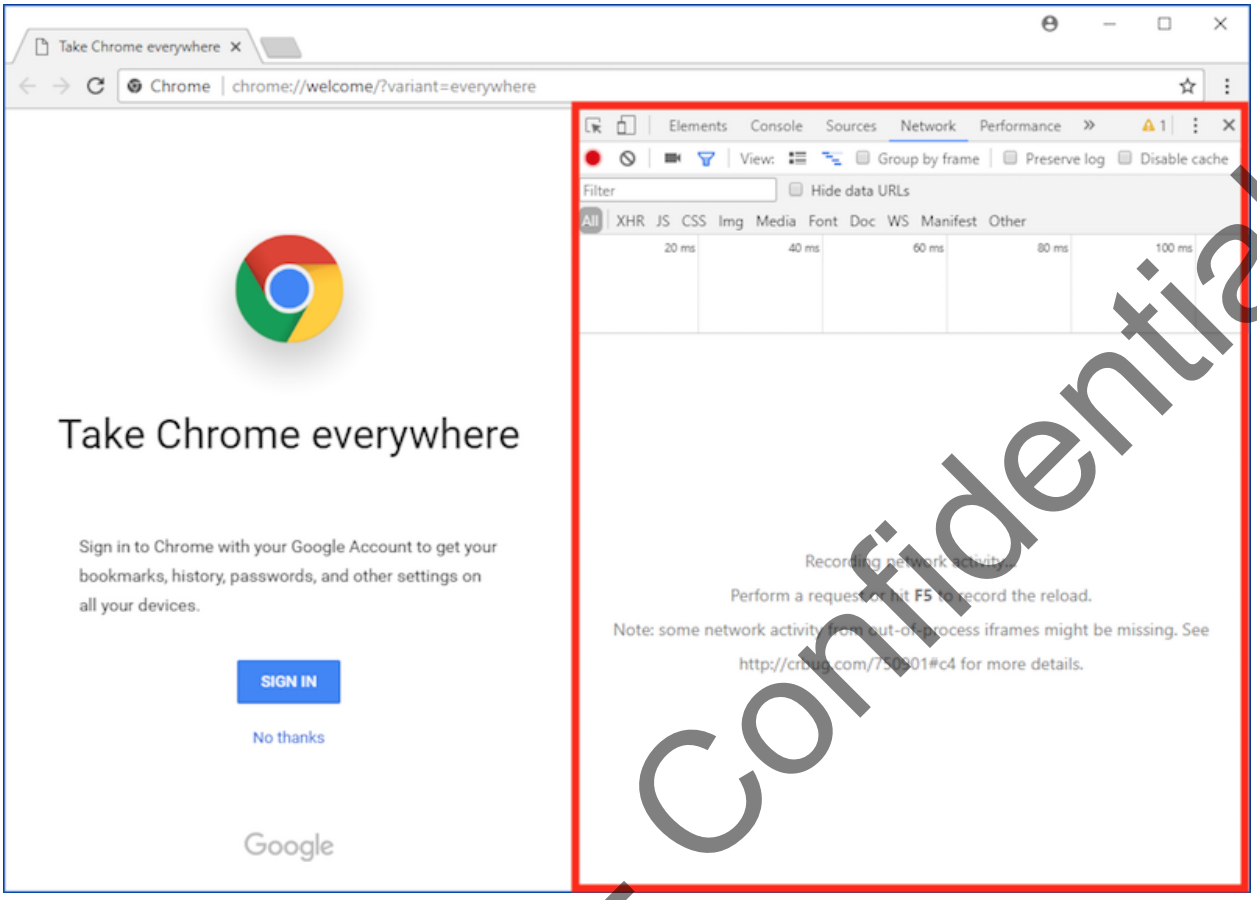
Verification

To access the Chrome Developer tools:

1. Open Google Chrome, and click the menu icon ("three dots") to the right of the address bar
2. Under **More tools**, click the link for **Developer tools**



3. You will now see the developer tools open within the window:



OpenConnect Installation

1. Install the OpenConnect client:

```
sudo apt install openconnect
```