Configuring a Virtual Environment

Background

A Python *virtual environment* is a tool used to create and maintain isolated Python environments. This allows each environment to be customized with all necessary dependencies and packages installed, without affecting other users or system-wide configurations. It is commonly used in AI/ML/DL development projects.

There are a number of tools to accomplish this task. This document describes the use of **virtualenv** on 'ghost', a Deep Learning workstation.

Setup and Configuration (basic environment)

1. login to ghost with your assigned username and password
    - you need to be on a GVSU network, or setup and use the Pulse Secure VPN (instructions on GVSU website)
    - IP address: **35.39.163.55**

    Recommended settings if you use **x2go**:

    - **Session tab**
        - Session type: XFCE
    - **Connection tab**
        - Connection speed: ADSL
        - Method: 256k-tight
    - **Media tab**
        - Uncheck/disable sound support
        - Uncheck/disable client-side printing support

2. test system for compatible versions and permissions

    ```
    python3 --version
    pip3 --version
    virtualenv --version
    ```

3. create a new Python 3.x virtual environment (e.g. called 'myenv') using the **virtualenv** utility

    ```
    $ virtualenv --system-site-packages -p python3 ./myenv
    ```

4. enter/activate the new virtual environment; note that the command-line indicates this

   ```
   $ cd myenv
   $ source bin/activate
   (myenv) wolffe@ghost:~$
   ```

5. download the file '**requirements.txt**' into your virtual environment

6. clone the reference virtual environment using the file 'requirements.txt' (i.e. install the specified packages)

   ```
   pip install -r requirements.txt
   ```

7. display the python packages installed (locally) into your new virtual environment

   ```
   pip list -l
   ```

8. test software installation

   ```
   python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
   ```

   Expected output:

   ```
   Tensor("Sum:0", shape=(), dtype=float32)
   ```

9. exit/deactivate the virtual environment; your normal prompt will return

   ```
   (myenv) wolffe@ghost:~$ deactivate
   wolffe@ghost:~$
   ```

Further Customization

After the above operations, you are basically the administrator of your environment. However, at some point you may need to modify, extend, or update your software. Here are a few additional instructions to get you started:

# installing a new package (e.g. 'gr' graphics framework)

```
     pip install gr
```

# installing a specific version of a package (e.g. 'numpy')

```
     pip install numpy==1.16.5
```

# performing an upgrade of any package (e.g. 'matplotlib')

```
pip install --upgrade matplotlilb
```