**Setup a Vagrant test box**

The Vagrant base box **hashicorp/precise32** is used for testing the application with **VirtualBox** provider.

Following are the sequence of commands run to deploy the app and install all the base dependencies.

sudo apt-get update

sudo apt-get -y install git

sudo apt-get -y install python-pip

sudo mkdir -p /Springer

sudo chmod 777 /Springer

git clone https://bitbucket.org/springersbm/pe\_springer\_nature\_assessment\_app.git /Springer

git clone https://github.com/pyenv/pyenv.git ~/.pyenv #installing pyenv

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

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

echo 'eval "$(pyenv init -)"' >> ~/.bashrc

source ~/.profile

sleep 5

git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyenv root)/plugins/pyenv-virtualenv

echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bashrc #installing virtualenv

source ~/.profile

sleep 5

cd /Springer

python setup.py test

python setup.py sdist

sudo pip install /Springer/dist/helloworld-springer-app-0.1.6.tar.gz

To automate the deployment of the application so that the VM created has the application already installed on it, the above sequence of codes are passed as inline shell commands in the Vagrantfile. The exact Vagrantfile used for deploying the application test box is attached.

After logging into the Vagrant VM the application needs to be configured with the two parameters

1. HELLOWORLD\_SPRINGER\_APP\_TIMEOUT
2. HELLOWORLD\_SPRINGER\_APP\_DATE\_ENDPOINT

export HELLOWORLD\_SPRINGER\_APP\_TIMEOUT=\*

export HELLOWORLD\_SPRINGER\_APP\_DATE\_ENDPOINT=<http://sleepy-thicket-8107.herokuapp.com/date>

\* should be the desired timeout time in secs. The time mentioned by default in the application installation instructions is 5 seconds.

xdg-utils and dependant packages are also installed. xdg-open is used to open the application at port 8080 of the guest VM machine. The following commands were thus also passed along with the application installation commands in the inline shell script.

sudo apt-get -y install xdg-utils

sudo apt-get -y install links

sudo apt-get -y install links2

sudo apt-get -y install lynx

The application can be started by the following command

python /Springer/helloworld\_springer\_app/\_\_main\_\_.py

The application response can be accessed by the following command

xdg-open <http://127.0.0.1:8080>

The server provisioning can also be similarly achieved by many other configuration management tools like Ansible local and Puppet agent. However the choice of **shell** provisioner was made because it comes by default with the Ubuntu server and would thus save the time of installing additional tools (Ansible/Puppet/..). However the other tools are a lot more appropriate for more complex applications with more dimensions for configuration and frequent upgradations.

**P.S: The timeout and endpoint parameter values are incorrectly shown (interchanged) in the Trello board card for this particular activity.**