|  |  |
| --- | --- |
| **Recommended PC configuration** | |
| **Windows**: Windows 10 or Higher  Minimum 8 GB RAM  Available 40 GB Hard Disk Space | **Mac**: Latest Mac OS  Minimum 8 GB RAM  Available 40 GB Hard Disk Space |

1. Download and install Mac/Windows versions of the following software:

[VirtualBox](https://www.virtualbox.org/wiki/Downloads) (The software that creates virtual machines)

[Vagrant](https://www.vagrantup.com/) (The software that deploys virtual machines into VirtualBox)

1. Setup Ubuntu Guest machines (Virtual machines) using Vagrant

* Create a folder DevOps/vagrant/ubuntu under preferred drive in your Mac/Windows PC
* Create a ‘Vagranfile’ file under ubuntu folder, past the sample vagrant file content from github links

<https://github.com/prkatta/dotraining/tree/master/installs/varant-vms/ubuntu/Vagrantfile>

* Open command prompts, cd to “DevOps\Vagrant\Ubuntu”
* (for MAC only) Go to VM folder where you created Vagrantfile and run “**vagrant init hashicorp/bionic64**”

C:\DevOps\Vagrant\Ubuntu>vagrant init hashicorp/bionic64

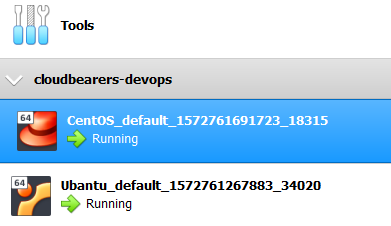
A `Vagrantfile` has been placed in this directory. You are now

ready to `vagrant up` your first virtual environment! Please read

the comments in the Vagrantfile as well as documentation on

`vagrantup.com` for more information on using Vagrant.

* (For MAC and Windows) Now from the same location, run “**vagrant up --provision**” for Ubuntu
* Let it complete the ubuntu machines up and running



**Login for Vagrant: vagrant / vagrant**

1. Now double click Ubuntu from Virtual Box and login to the Ubuntu with vagrant/vagrant

|  |
| --- |
| **Ubuntu** |
| **GIT**  sudo apt-get install git -y |
| **JDK**  sudo apt-get install openjdk-8-jdk -y  sudo apt-get install openjdk-11-jdk -y |
| **Ngnix**  sudo apt-get install nginx -y |
| **Tomcat**  sudo apt-get install tomcat -y |
| **Maven & Gradle**  sudo apt-get install maven -y  sudo apt-get install gradle -y |
| **My SQL**  sudo apt-get install mysql-server -y |
| **Postgress**  https://tecadmin.net/install-postgresql-server-on-ubuntu/  sudo apt-get install wget ca-certificates  wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb\_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'  sudo apt-get update  sudo apt-get install postgresql postgresql-contrib |
| **NodeJS & NPM**  sudo apt-get install nodejs -y  sudo apt-get install npm -y |
| **Docker**  sudo apt-get install docker -y |
| **Groovy**  sudo apt-get install groovy -y |
| **Python**  **Python 2.7.16**  https://tecadmin.net/install-python-2-7-on-ubuntu-and-linuxmint/  sudo apt-get update  sudo apt-get install build-essential checkinstall  sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev  cd /usr/src  sudo wget https://www.python.org/ftp/python/2.7.16/Python-2.7.16.tgz  cd Python-2.7.16  sudo tar xzf Python-2.7.16.tgz  cd Python-2.7.16  sudo ./configure --enable-optimizations  sudo make altinstall  python2.7 -V  **Python 3.7:**  wget https://www.python.org/ftp/python/3.7.1/Python-3.7.1.tgz  tar -xvf Python-3.7.1.tgz  sudo apt-get install gcc  sudo apt-get install libffi-dev  cd Python-3.7.1  ./configure --enable-optimizations  sudo make  sudo make install  python3.7 -V  sudo apt-get upgrade python3 |
| **Jenkins LTS**  wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -  sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'  sudo apt-get update  sudo apt-get install jenkins |
| **Grafana**  sudo apt-get install -y gnupg2 curl  curl https://packages.grafana.com/gpg.key | sudo apt-key add -  sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"  sudo apt-get update  sudo apt-get -y install grafana  sudo systemctl start grafana-server  sudo systemctl status grafana-server |
| **Ansible**  sudo apt-add-repository ppa:ansible/ansible  sudo apt-get update  sudo apt-get install ansible -y |
| **JQuery**  sudo apt-get install jq |
| **CheckMK**  sudo apt-get install checkmk |
| **ELK**  <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-18-04> |

**How to switch between JAVAs**

sudo alternatives --config java

set java 11

Puppet Environment:

Foreman: <https://www.theforeman.org/manuals/1.23/index.html#2.1Installation>

Ansible:

Ansible Environment: <https://www.theforeman.org/manuals/1.23/index.html#2.1Installation>

Jq : <https://stedolan.github.io/jq/>

**Reference Links:**

Git- > <https://www.digitalocean.com/community/tutorials/how-to-install-git-on-centos-7>

Maven-> <https://www.tecmint.com/install-apache-maven-on-centos-7/>

Mysql -> <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-centos-7>

Tomcat -> <https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-7-on-centos-7-via-yum>

Python -> <https://www.digitalocean.com/community/tutorials/how-to-set-up-python-2-7-6-and-3-3-3-on-centos-6-4>

Docker-> <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-centos-7>

Postgres -> <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-centos-7>

ELK- <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-centos-7>

CheckMK: <https://www.fosslinux.com/8424/install-and-configure-check_mk-server-on-centos-7.htm>

Linux Sheet Cheat : <https://www.linuxtrainingacademy.com/linux-commands-cheat-sheet/#12_8211_SSH_LOGINS>

**IDE:**

Intellij: <https://www.jetbrains.com/idea/download/#section=windows>

PyCharm: <https://www.jetbrains.com/pycharm/>

Atom: <https://atom.io>

**Java Build Tools**

* Maven: <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
* Gradle: <https://gradle.org/install/>

Example projects:

Jenkins: <https://github.com/jenkinsci/jenkins>

SonarQube: <https://github.com/SonarSource/sonarqube>

Spring pet-clinic: <https://github.com/spring-petclinic/spring-framework-petclinic>

Misc samples : <https://github.com/jfrog/project-examples/tree/master/python-example>

Sample Programs for All languages

* Python: <https://github.com/jfrog/project-examples/tree/master/python-example>