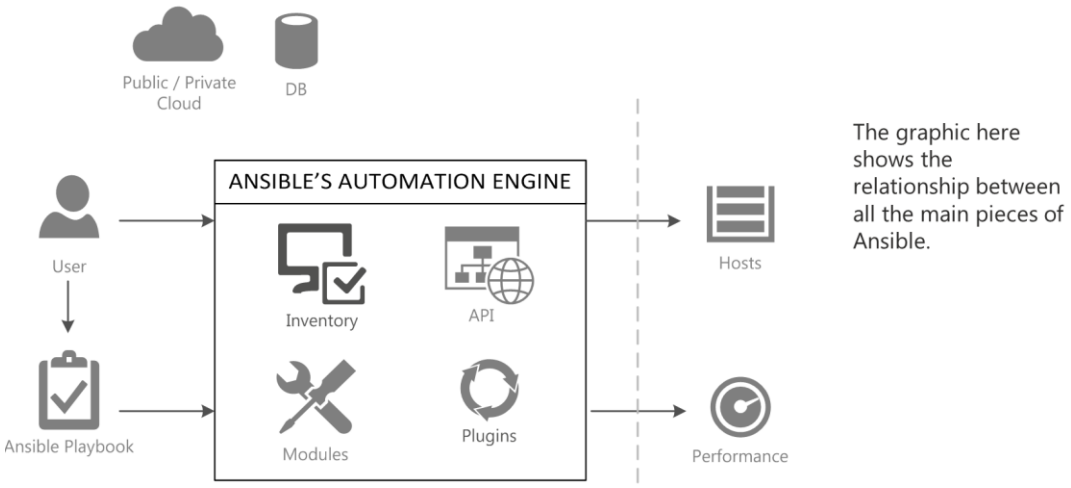
Ansible



Make master and agent VM

Install 2 virtual machines using vagrant. One acts as master, another as agent.

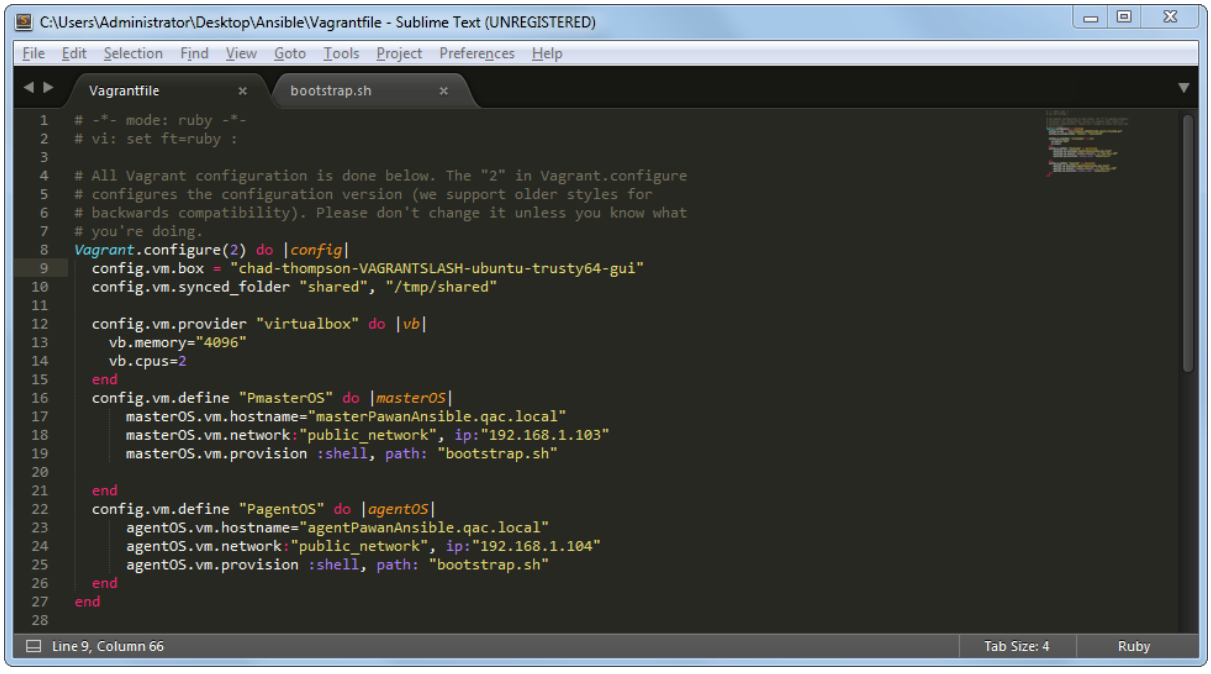
Create a directory **\Ansible**

Right click and use **Git Bash**

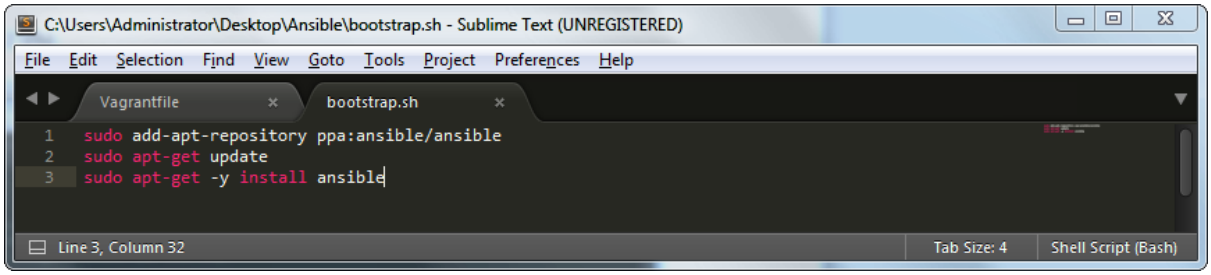
Use **vagrant init** command

Create directory **\Ansible\shared** for shared folder

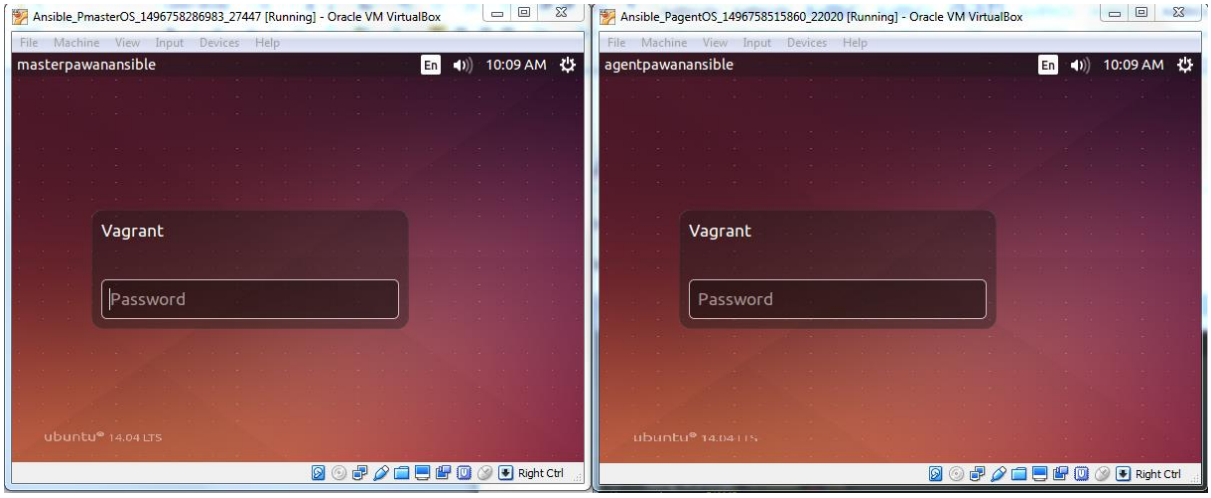
Edit the **vagrantfile** to install the 2 VM.



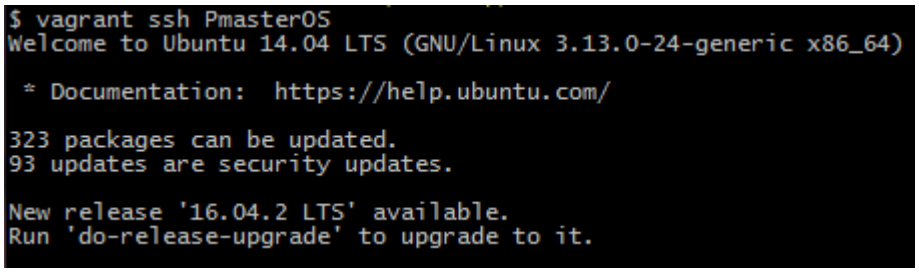
Write the script file to install ansible called **bootstrap.sh**



Use **vagrant up** command to run the 2 VM



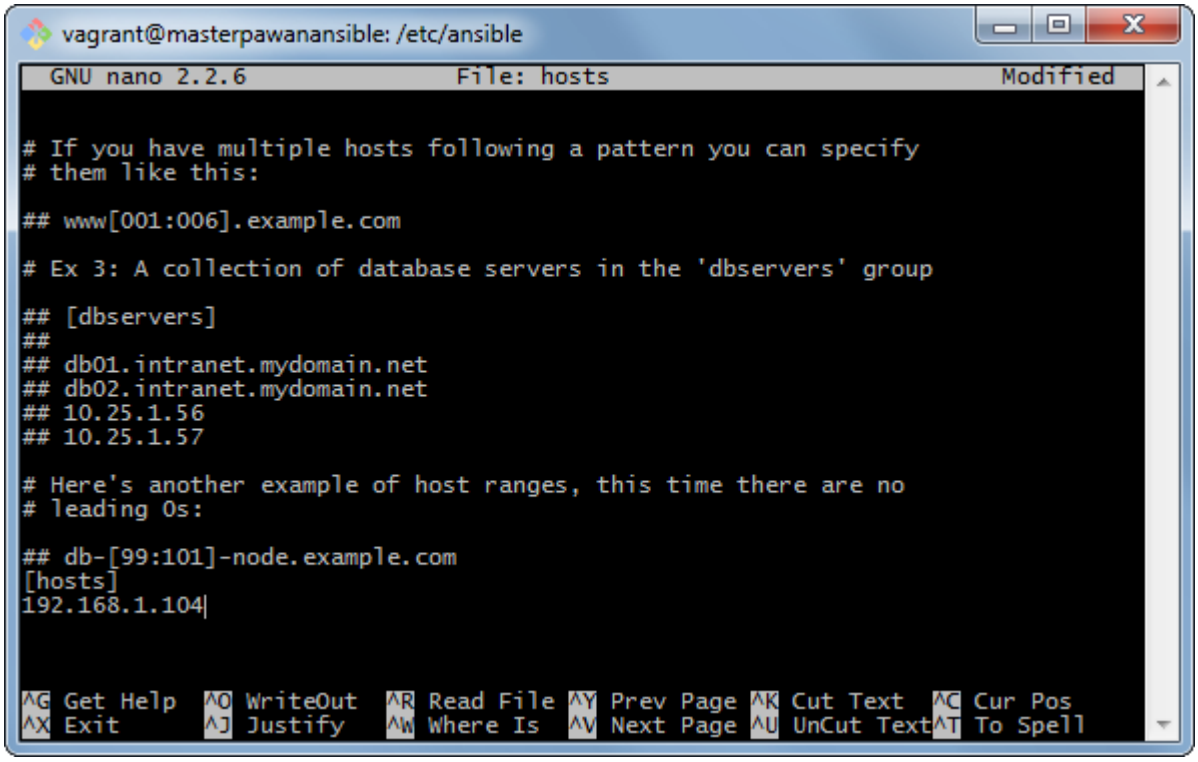
On windows host, do **vagrant ssh hostname** of the master VM



Go to **cd /etc/ansible** and edit the hosts file with

sudo nano hosts

Add the group name [hosts] and the agent ip address below it



Create a ssh key with **ssh-keygen -t rsa**

```
vagrant@masterpawanansible:/etc/ansible$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa.
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub.
The key fingerprint is:
0b:11:4f:49:49:f3:ac:a2:7d:b7:41:72:7c:c0:11:2d vagrant@masterpawanansible
The key's randomart image is:
+--[ RSA 2048]----+
|       .o=ooo    |
|        +o=E..   |
|       . . =.    |
|        . o .    |
|       o S + .   |
|      o o = .    |
|     . . o o     |
|         . . o   |
|            .    |
+-----------------+
```

ssh-agent bash

ssh-agent keeps the key in memory and bash makes it accessible to the terminal

```
vagrant@masterpawanansible:/etc/ansible$ ssh-agent bash
```

ssh-add ~/.ssh/id_rsa

ssh-add adds the private key to the ssh-agent

ssh-copy-id vagrant@NODEIP

ssh-copy-id can be used to install the ssh key as an authorized key on the agent machine

To test this was successful you can execute the following command from the master. You should receive the success message.

ansible all -i hosts -u vagrant -m setup

```
vagrant@masterpawanansible:/etc/ansible$ ssh-add ~/.ssh/id_rsa
Identity added: /home/vagrant/.ssh/id_rsa (/home/vagrant/.ssh/id_rsa)
vagrant@masterpawanansible:/etc/ansible$ ssh-copy-id vagrant@192.168.1.104
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
vagrant@192.168.1.104's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.1.104'"
and check to make sure that only the key(s) you wanted were added.

vagrant@masterpawanansible:/etc/ansible$ ansible all -i hosts -u vagrant -m setu
p
192.168.1.104 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
```

Ping to check the agent is running

```
vagrant@masterpawanansible:/etc/ansible$ ansible all -m ping
192.168.1.104 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
vagrant@masterpawanansible:/etc/ansible$
```

Ansible playbook

Install Java, Maven and Git

Create a yml file on **/etc/ansible/** on master where the **hosts** inventory file is located.

--- Insert triple dash which is like playbook interpreter, it not present the file won't run
name = description of what the file does
hosts = which agents to carry out the yml on, the group is listed on **hosts** inventory file
remote_user = to run as which user on agents
become = give sudo privilege

```
#filename javamavengit.yml
---
- name: install java, maven and git
  hosts: all
  remote_user: vagrant
  become: yes
```

Write the tasks. Copy JAVA and MAVEN to **/opt/** directory

```
tasks:
 - name: copy Java
   copy:
     src: /tmp/shared/java.tar.gz
     dest: /opt/java.tar.gz
 - name: Copy Maven
   copy:
     src: /tmp/shared/maven.tar.gz
     dest: /opt/maven.tar.gz
```

Use unarchive to unzip the tar files

```
  - name: Install java
    unarchive:
      src: /opt/java.tar.gz
      dest: /opt/
      copy: no
  - name: Install maven
    unarchive:
      src: /opt/maven.tar.gz
      dest: /opt/
      copy: no
```

By default, Ansible copies the file (src) from control machine to the remote machine and unarchives it. Since our machine is Windows which we use to ssh to master Ubuntu machine, set **copy: no**.

Create symlinks so we can use java, javac, maven from cmd using **shell** keyword.

```
  - name: create symlinks
    shell: "{{ item }}"
    with_items:
      - "update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_45/bin/$
      - "update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_45/bi$
      - "update-alternatives --install /usr/bin/mvn mvn /opt/apache-maven-3.3.9$
```

update_cache=yes is equivalent to apt-get update

Then install git from updated package repo

```
  - name: update package manager
    apt: update_cache=yes

  - name: install git
    apt: name=git state=present
```

Run ansible-playbook on -i (inventory) hosts and run the javamavengit.yml instruction

ansible-playbook -i hosts javamavengit.yml

```
vagrant@masterpawanansible:/etc/ansible$ ansible-playbook -i hosts javamavengit.
yml

PLAY [install java, maven and git] *********************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.104]

TASK [copy Java] ***************************************************************
ok: [192.168.1.104]

TASK [Copy Maven] **************************************************************
ok: [192.168.1.104]

TASK [Install java] ************************************************************
changed: [192.168.1.104]
```

Successful message where tasks failed=0 and agents unreachable=0

```
PLAY RECAP *********************************************************************
192.168.1.104              : ok=8    changed=2    unreachable=0    failed=0

vagrant@masterpawanansible:/etc/ansible$
```

Install Jenkins, Jira and Nexus

Copy over local files

response.varfile contains response required by jira

```
#filename jenkinsjiranexus.yml
---
- name: install jenkins, jira and nexus
  hosts: all
  remote_user: vagrant
  become: yes

  tasks:

   - name: Copy Jenkins
     copy:
       src: /tmp/shared/jenkins_2.1_all.deb
       dest: /home/vagrant/Desktop/jenkins_2.1_all.deb

   - name: Copy Jira
     copy:
       src: /tmp/shared/jira.bin
       dest: /opt/jira.bin
       mode: 755

   - name: Copy responsefile
     copy:
       src: /tmp/shared/response.varfile
       dest: /opt/response.varfile

   - name: Copy Nexus
     copy:
       src: /tmp/shared/nexus-2.14.4-03-bundle.tar.gz
       dest: /usr/local/nexus-2.14.4-03-bundle.tar.gz
```

Update the package manager apt

Unpackage Jenkins with apt so it can be run as service

```
- name: update package manager
  apt: update_cache=yes

- name: Install Jenkins
  apt: deb="/home/vagrant/Desktop/jenkins_2.1_all.deb"

- name: Run Jenkins
  service: name=jenkins state=started enabled=yes
```

Use the **response.varfile** to run jira with previously taken user input; change directory to **/opt/** to run it; and check that directory created is present or to make jira installation idempotent

```
   - name: Install Jira
     shell: "./jira.bin -q -varfile response.varfile"
     args:
       chdir: /opt/
       creates: /opt/atlassian/jira/atlassian-jira/WEB-INF
```

Unpack nexus to **/usr/local/** according to nexus convention, file is **nexus-2.14.4-03**
Check the status of the nexus directory and register it to variable **nexusdir**
when nexus directory doesn't exist, create it

```
  - name: Unpack nexus
    unarchive:
      src: /usr/local/nexus-2.14.4-03-bundle.tar.gz
      dest: /usr/local/
      copy: no

  - stat:
      path: /usr/local/nexus
    register: nexusdir

  - name: Create nexus symlink folder
    file:
     path: usr/local/nexus
     state: directory
     mode: 0755
    when: not nexusdir.stat.exists
```

Create symlink between **nexus-2.14.4-03** and **nexus** folder and set permission to vagrant user. Do the same with **sonatype** folder required by nexus.

```
   - name: Make nexus smylink
     file:
       src: /usr/local/nexus-2.14.4-03
       dest: /usr/local/nexus
       state: link
       owner: vagrant
       mode: 0755
       force: yes

   - name: Change sonatype permission
     file:
       path: /usr/local/sonatype-work
       owner: vagrant
       mode: 0755
```

Use appropriate version of java using **update-alternatives**
Run nexus from directory **/usr/local/nexus/** and run nexus from **/usr/local/nexus/bin/nexus**

```
- name: Change java version to run nexus
  shell: "echo '1' | sudo update-alternatives --config java"

- name: Run nexus
  shell: "{{ item }}"
  become_user: vagrant
  args:
    chdir: /usr/local/nexus/
  with_items:
   - "./bin/nexus console"
   - "./bin/nexus start"
```

Run with **ansible-playbook -i hosts jenkinsjiranexus.yml**

```
TASK [Change java version to run nexus] ***********************************
changed: [192.168.1.104]

TASK [Run nexus] ***********************************************************
changed: [192.168.1.104] => (item=./bin/nexus console)
changed: [192.168.1.104] => (item=./bin/nexus start)

PLAY RECAP *****************************************************************
192.168.1.104              : ok=15   changed=3    unreachable=0    failed=0
```