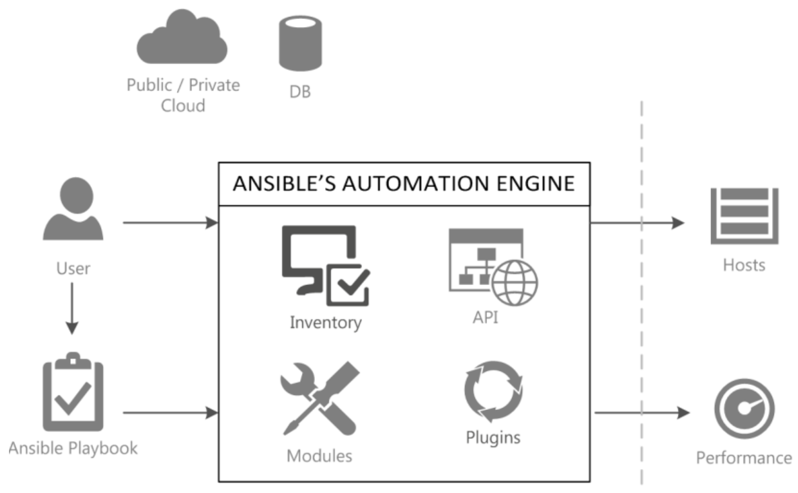
Ansible



The graphic here shows the relationship between all the main pieces of Ansible.

Contents

Make master and agent VM

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

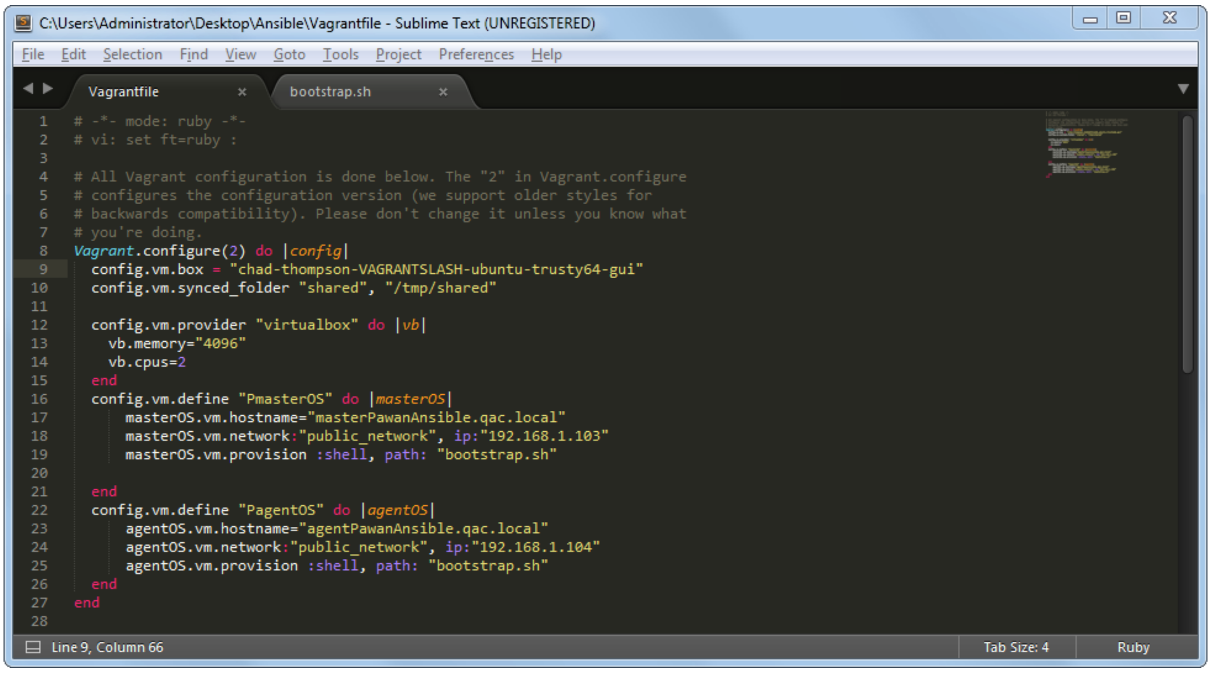
Create a directory **\Ansible**

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.

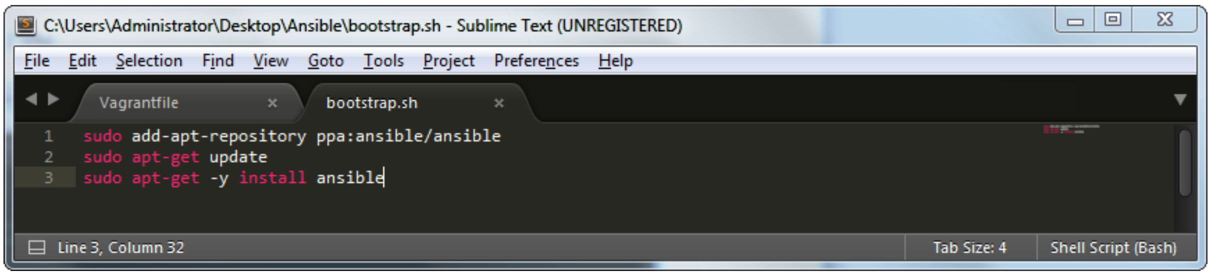


C:\Users\Administrator\Desktop\Ansible\Vagrantfile - Sublime Text (UNREGISTERED)

```ruby
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure(2) do |config|
  config.vm.box = "chad-thompson-VAGRANTSLASH-ubuntu-trusty64-gui"
  config.vm.synced_folder "shared", "/tmp/shared"

  config.vm.provider "virtualbox" do |vb|
    vb.memory="4096"
    vb.cpus=2
  end
  config.vm.define "PmasterOS" do |masterOS|
      masterOS.vm.hostname="masterPawanAnsible.qac.local"
      masterOS.vm.network:"public_network", ip:"192.168.1.103"
      masterOS.vm.provision :shell, path: "bootstrap.sh"

  end
  config.vm.define "PagentOS" do |agentOS|
      agentOS.vm.hostname="agentPawanAnsible.qac.local"
      agentOS.vm.network:"public_network", ip:"192.168.1.104"
      agentOS.vm.provision :shell, path: "bootstrap.sh"
  end
end

```

Line 9, Column 66 | Tab Size: 4 | Ruby

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

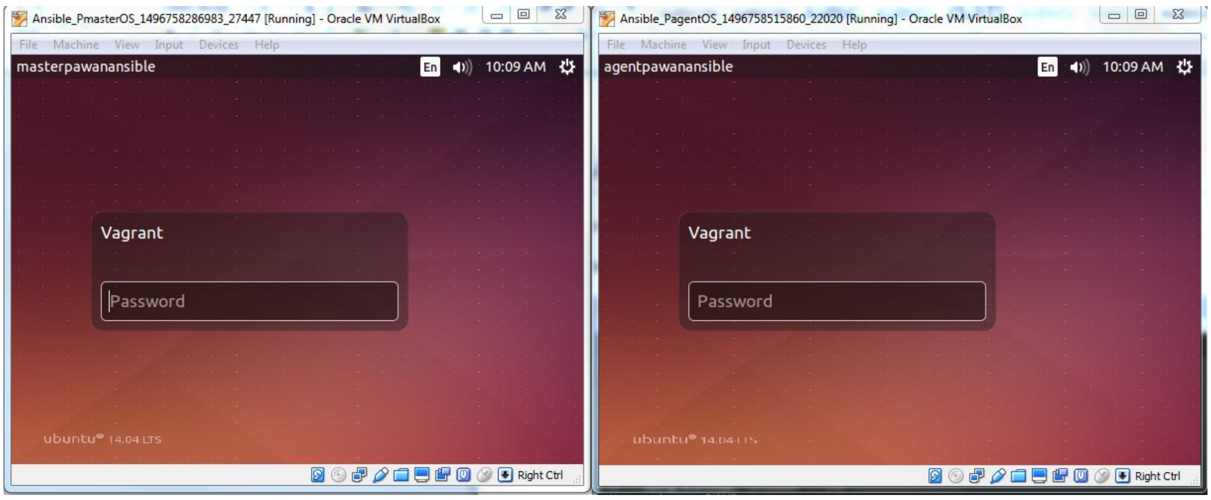


C:\Users\Administrator\Desktop\Ansible\bootstrap.sh - Sublime Text (UNREGISTERED)

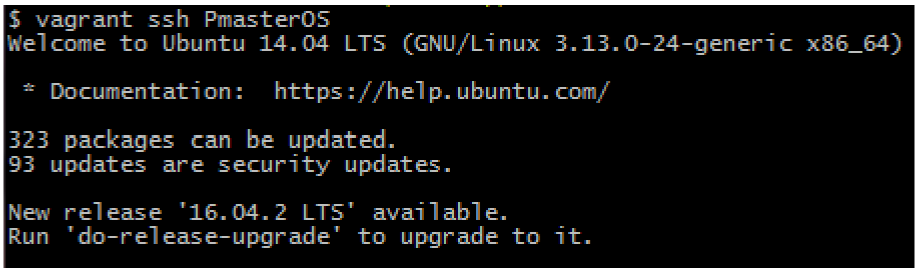
```bash
sudo add-apt-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get -y install ansible
```

Line 3, Column 32 | Tab Size: 4 | Shell Script (Bash)

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


Ansible_PmasterOS_1496758286983_27447 [Running] - Oracle VM VirtualBox
masterpawanansible
10:09 AM
Vagrant
Password
ubuntu 14.04 LTS
Ansible_PagentOS_1496758515860_22020 [Running] - Oracle VM VirtualBox
agentpawanansible
10:09 AM
Vagrant
Password
ubuntu 14.04 LTS

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

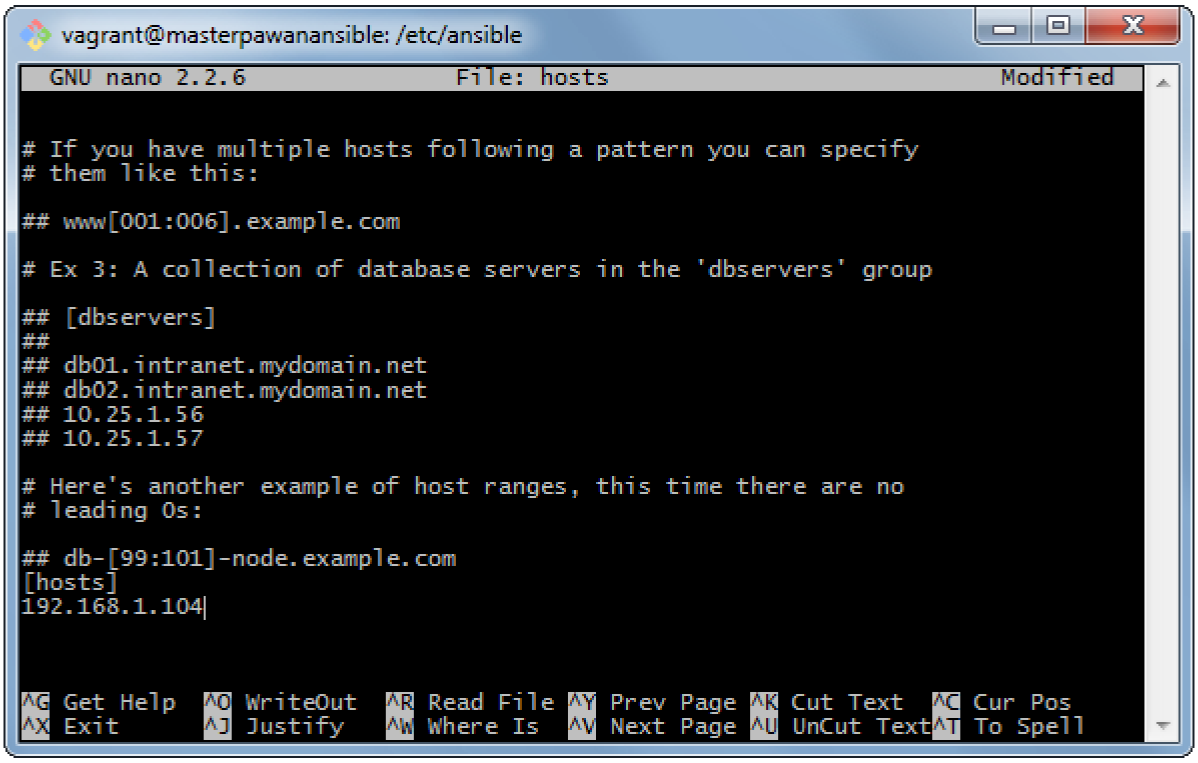


```
$ vagrant ssh PmasterOS
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

323 packages can be updated.
93 updates are security updates.

New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

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

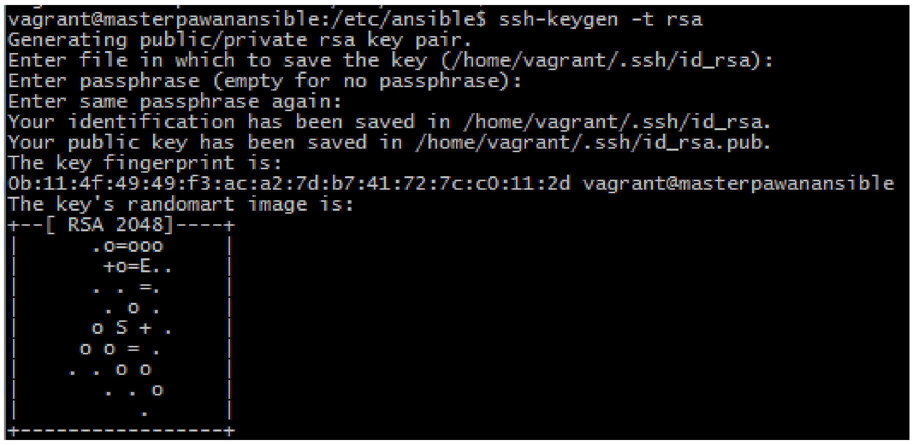
sudo nano hosts

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



```
vagrant@masterpawanansible: /etc/ansible
  GNU nano 2.2.6              File: hosts                          Modified


# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[hosts]
192.168.1.104



^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

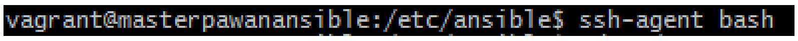
Create a ssh key with **ssh-keygen -t rsa**



```
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

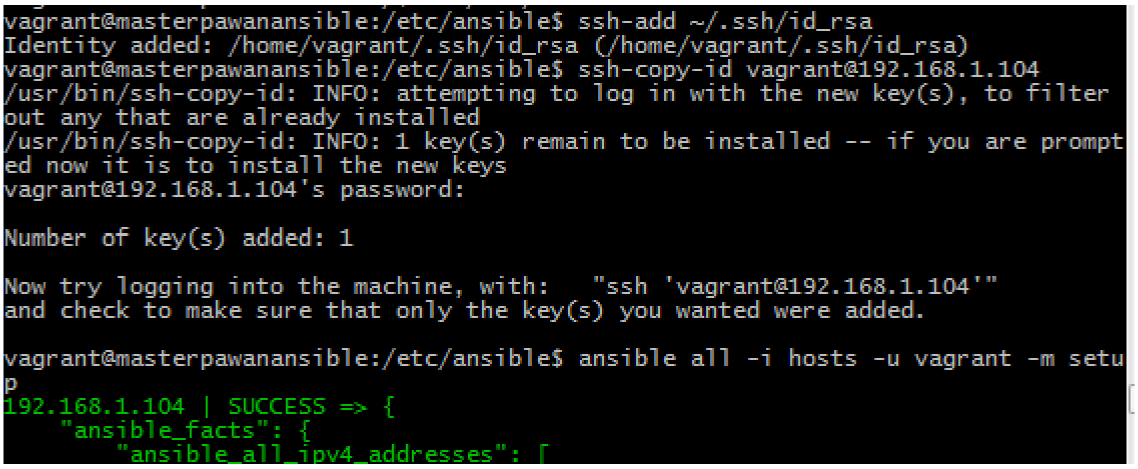
ssh-add adds the private key to the ssh-agent

ssh-copy-id vagrant@NODEIP

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

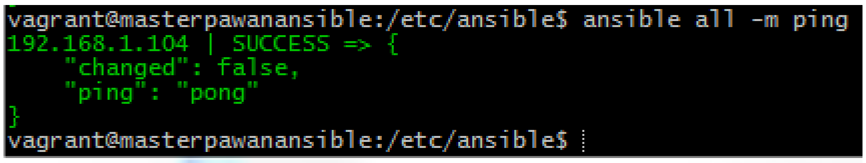
To test this was successful you can execute the following command from the master. You should receive the success message.

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



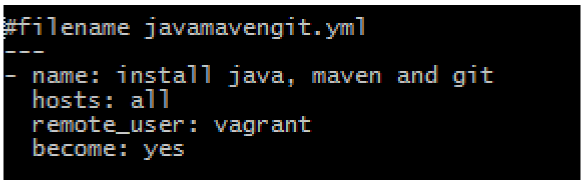
```
vagrant@masterpawanansible:/etc/ansible$ ansible all -m ping
192.168.1.104 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
vagrant@masterpawanansible:/etc/ansible$
```

Ansible playbook

Install Java, Maven and Git

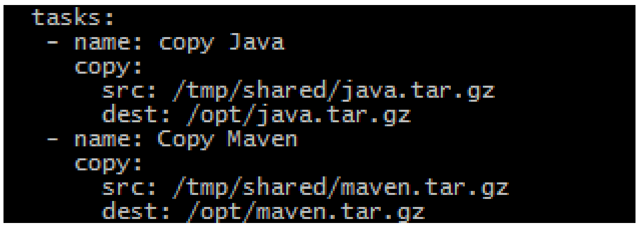
Create a yml file on **/etc/ansible/** on master where the **hosts** inventory file is located.

--- 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



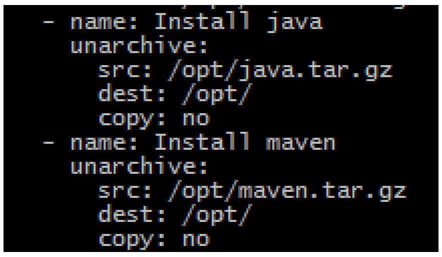
```
#filename javamavengit.yml
---
- name: install java, maven and git
  hosts: all
  remote_user: vagrant
  become: yes
```

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
```

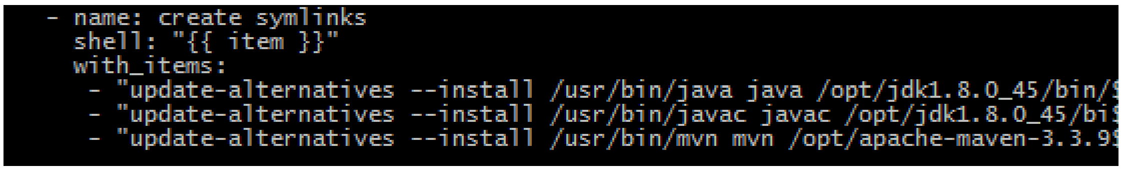
Use unarchive to unzip the tar files



```
   - 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**.

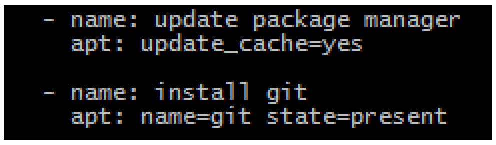
Create symlinks so we can use java, javac, maven from cmd using **shell** keyword.



```
   - 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

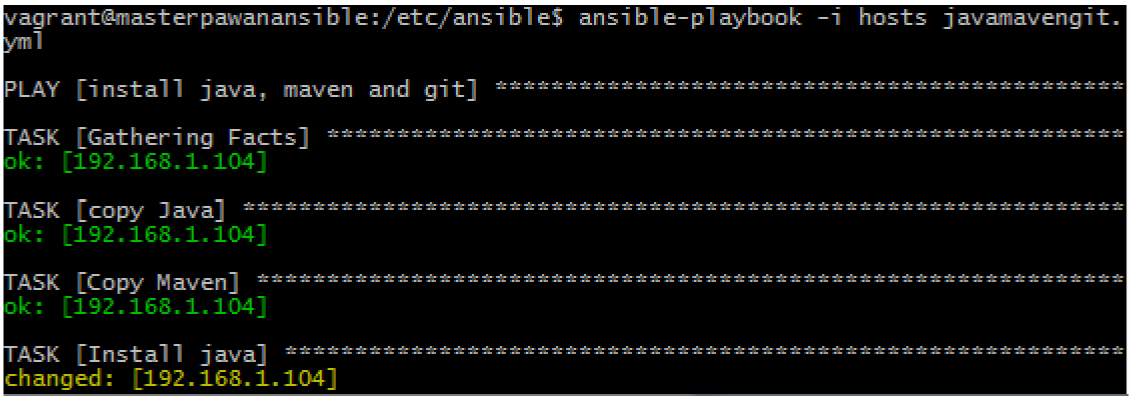
Then install git from updated package repo



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

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

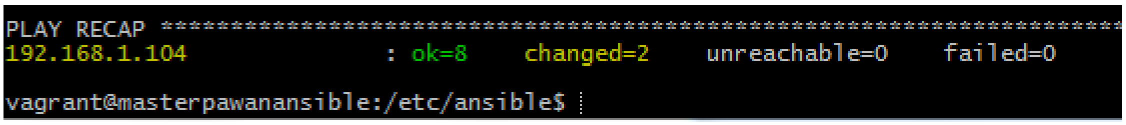
Run ansible-playbook on -i (inventory) hosts and run the javamavengit.yml instruction

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

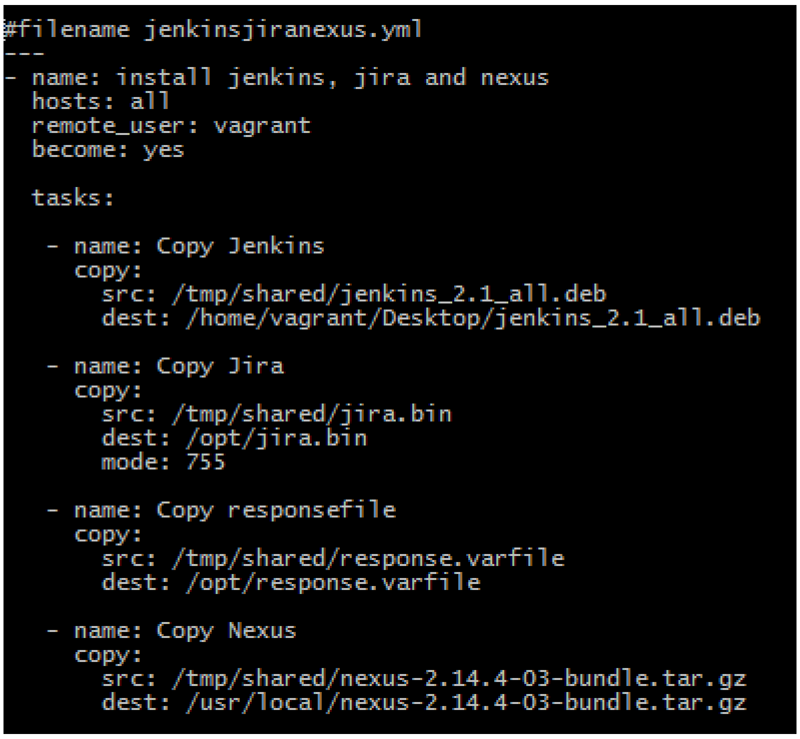


```
PLAY RECAP *********************************************************************
192.168.1.104              : ok=8    changed=2    unreachable=0    failed=0

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

Install Jenkins, Jira and Nexus

Copy over local files

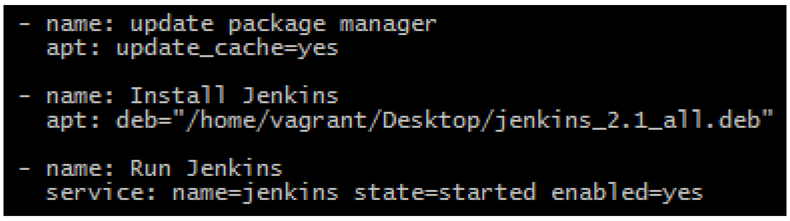
response.varfile contains response required by jira



```
#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
```

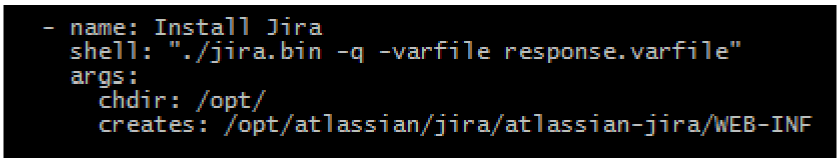
Update the package manager apt

Unpackage Jenkins with apt so it can be run as service



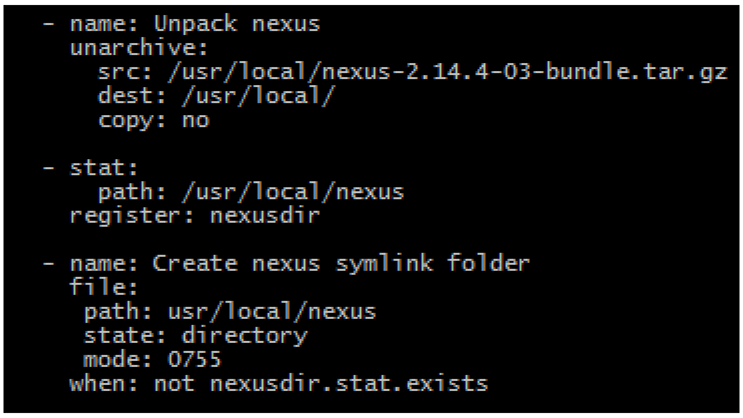
```
 - name: update package manager
   apt: update_cache=yes

 - 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



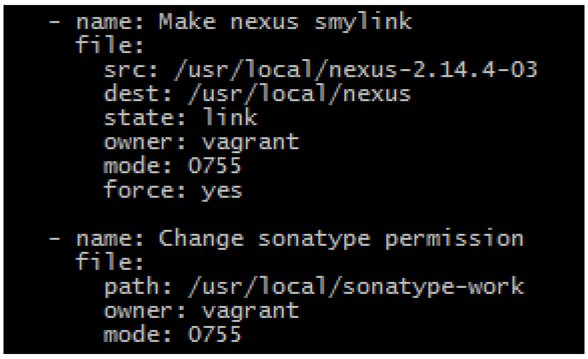
```
   - name: Install Jira
     shell: "./jira.bin -q -varfile response.varfile"
     args:
       chdir: /opt/
       creates: /opt/atlassian/jira/atlassian-jira/WEB-INF
```

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



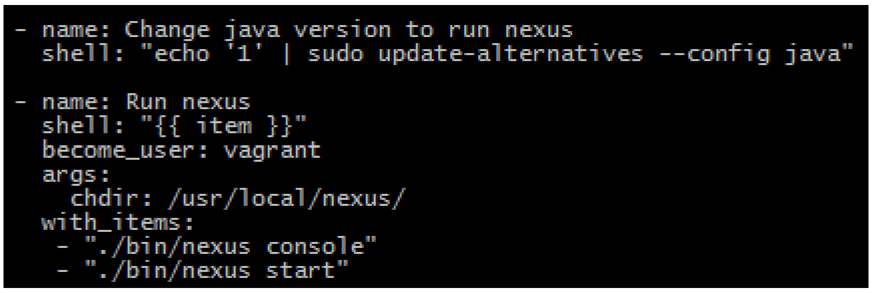
```
  - name: Unpack nexus
    unarchive:
      src: /usr/local/nexus-2.14.4-03-bundle.tar.gz
      dest: /usr/local/
      copy: no

  - 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.



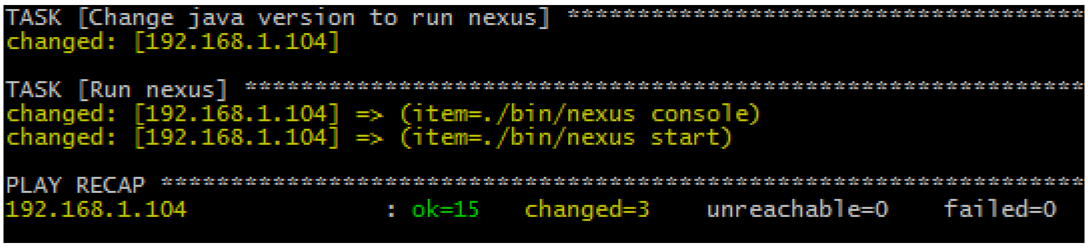
```
  - name: Make nexus smylink
    file:
      src: /usr/local/nexus-2.14.4-03
      dest: /usr/local/nexus
      state: link
      owner: vagrant
      mode: 0755
      force: yes

  - 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**



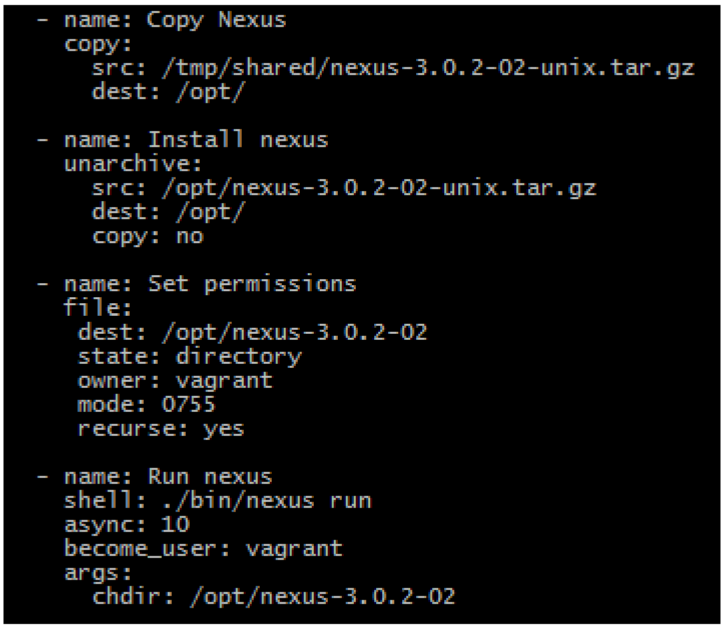
```
TASK [Change java version to run nexus] *************************************
changed: [192.168.1.104]

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
```

Nexus Version 3

Copy the tar file **nexus-3.0.2-02-unix.tar.gz**.

Make a user (e.g. **vagrant**) owner of file as it’s not recommended to run as root.

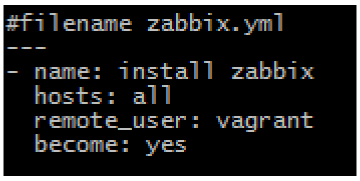
Execute the file with **./bin/nexus run**



```
  - name: Copy Nexus
    copy:
      src: /tmp/shared/nexus-3.0.2-02-unix.tar.gz
      dest: /opt/

  - name: Install nexus
    unarchive:
      src: /opt/nexus-3.0.2-02-unix.tar.gz
      dest: /opt/
      copy: no

  - name: Set permissions
    file:
     dest: /opt/nexus-3.0.2-02
     state: directory
     owner: vagrant
     mode: 0755
     recurse: yes

  - name: Run nexus
    shell: ./bin/nexus run
    async: 10
    become_user: vagrant
    args:
      chdir: /opt/nexus-3.0.2-02
```

Install Zabbix

Set the file headings



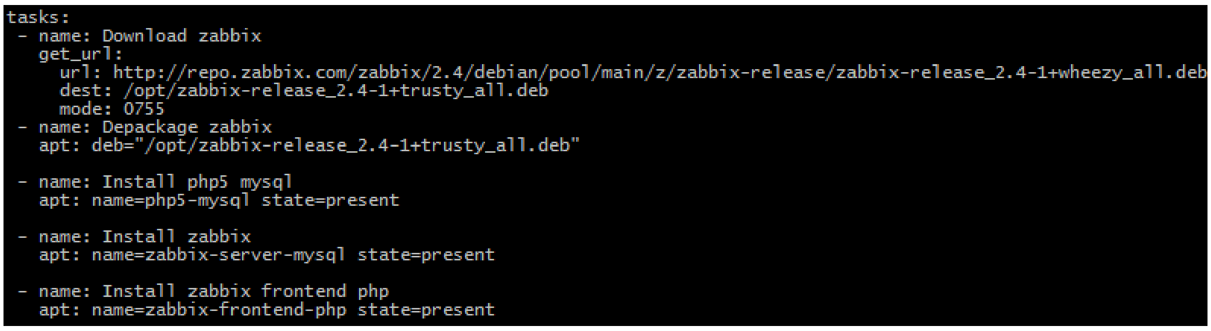
```
#filename zabbix.yml
---
- name: install zabbix
  hosts: all
  remote_user: vagrant
  become: yes
```

Define tasks

Download zabbix.

Depackage with apt so it can be seen as a service to make it idempotent.

Install php5 mysql and zabbix frontend



```
tasks:
 - name: Download zabbix
   get_url:
     url: http://repo.zabbix.com/zabbix/2.4/debian/pool/main/z/zabbix-release/zabbix-release_2.4-1+wheezy_all.deb
     dest: /opt/zabbix-release_2.4-1+trusty_all.deb
     mode: 0755
 - name: Depackage zabbix
   apt: deb="/opt/zabbix-release_2.4-1+trusty_all.deb"

 - name: Install php5 mysql
   apt: name=php5-mysql state=present

 - name: Install zabbix
   apt: name=zabbix-server-mysql state=present

 - name: Install zabbix frontend php
   apt: name=zabbix-frontend-php state=present
```

Next we need to connect php and mysql

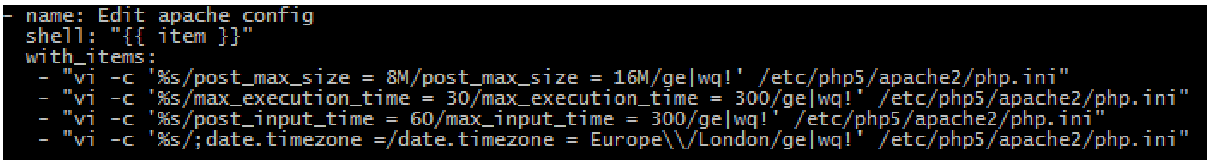
Edit the apache config file **/etc/php5/apache2/php.ini** with vim.

vi -c = perform a command

"%s/KEYWORD/NEW_KEYWORD/g" = vim can pattern match with the following command without opening a file where a keyword is replaced with new keyword. "%s/**…/../ge**" ge will force exit even if pattern not found

wq! = write and quit and force quit

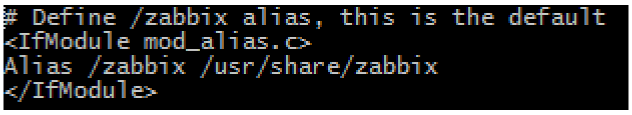
Europe\\/London = required string is "Europe/London" however / needs to be escaped on vim with a \ that makes it "Europe\/London" however this gives an error on yml files as \ has to be escaped. Therefore escape that with \ to make "Europe\\/London"



```
- name: Edit apache config
  shell: "{{ item }}"
  with_items:
    - "vi -c '%s/post_max_size = 8M/post_max_size = 16M/ge|wq!' /etc/php5/apache2/php.ini"
    - "vi -c '%s/max_execution_time = 30/max_execution_time = 300/ge|wq!' /etc/php5/apache2/php.ini"
    - "vi -c '%s/post_input_time = 60/max_input_time = 300/ge|wq!' /etc/php5/apache2/php.ini"
    - "vi -c '%s/;date.timezone =/date.timezone = Europe\\/London/ge|wq!' /etc/php5/apache2/php.ini"
```

Restart apache2 to make the config file set timezone and above setting.

Copy the configuration file apache.conf which sets alias to zabbix and apache2

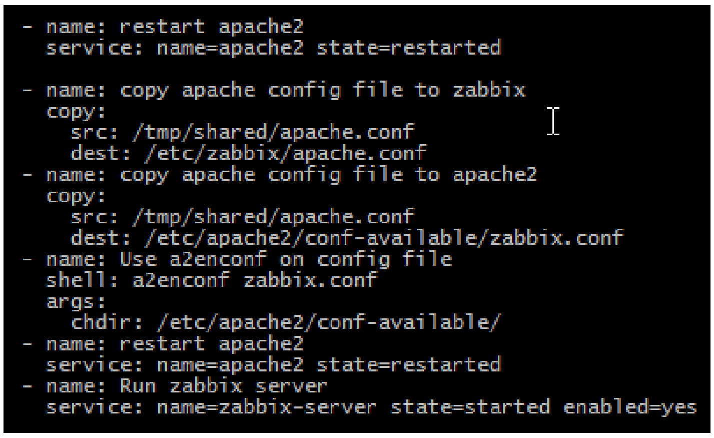


```
# Define /zabbix alias, this is the default
<IfModule mod_alias.c>
Alias /zabbix /usr/share/zabbix
</IfModule>
```

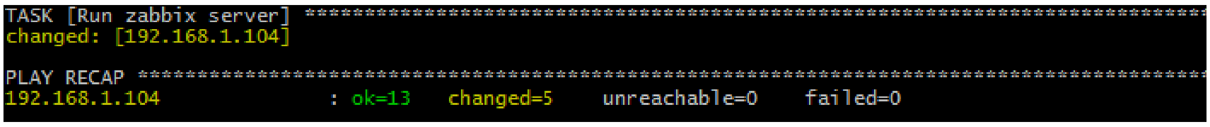
Run a2enconf to enable the config file for apache2

a2enconf *is a script that enables the specified configuration file within the* ***apache2*** *configuration. It does this by creating symlinks within* ***/etc/apache2/conf-enabled****.*

Next restart apache2 and run zabbix-server



```
 - name: restart apache2
   service: name=apache2 state=restarted

 - name: copy apache config file to zabbix
   copy:
     src: /tmp/shared/apache.conf
     dest: /etc/zabbix/apache.conf
 - name: copy apache config file to apache2
   copy:
     src: /tmp/shared/apache.conf
     dest: /etc/apache2/conf-available/zabbix.conf
 - name: Use a2enconf on config file
   shell: a2enconf zabbix.conf
   args:
     chdir: /etc/apache2/conf-available/
 - name: restart apache2
   service: name=apache2 state=restarted
 - name: Run zabbix server
   service: name=zabbix-server state=started enabled=yes
```



```
TASK [Run zabbix server] ****************************************************************************
changed: [192.168.1.104]

PLAY RECAP ******************************************************************************************
192.168.1.104              : ok=13   changed=5    unreachable=0    failed=0
```