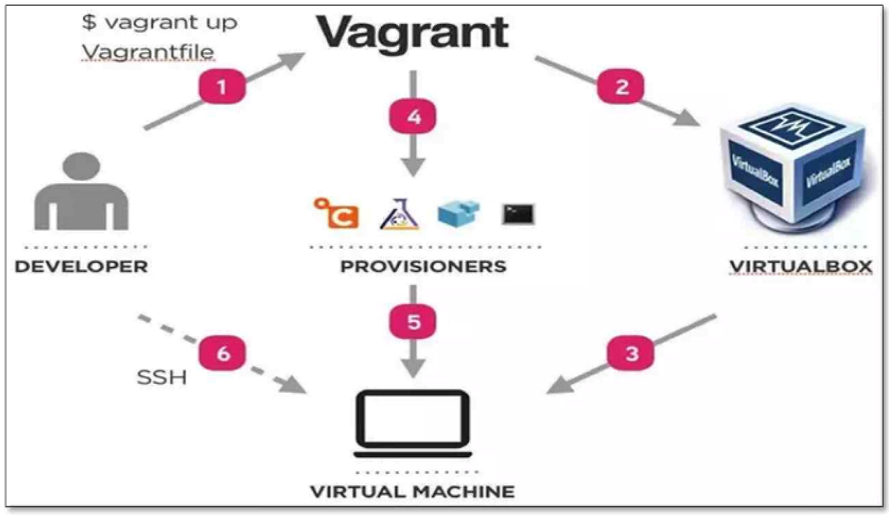
3rd September 2018

Agenda

- Prerequisites
- Ansible Introduction
- Basic Ansible Terminologies
- SSH communication with Ansible
- Ansible Architecture
- Install and Configure Ansible
- Ansible ad hoc commands
- Yaml Introduction
- Role based folder structure in Ansible
- Try Roles - Install Tomcat using Ansible
- Jenkins With Ansible

What is Vagrant ?

Vagrant is a tool for building and managing virtual machine environments in a single workflow. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the "works on my machine" excuse a relic of the past.



Prerequisites

Machine Proxy

- Add proxy in internet options
  - http://www-proxy.us.oracle.com Port:80
- Add environment variables
  - HTTP_PROXY → http://www-proxy.us.oracle.com:80
  - HTTPS_PROXY → http://www-proxy.us.oracle.com:80

Vagrant

- Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.
  - https://www.vagrantup.com/downloads.html

Oracle Virtual Box

- VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use.
  - https://www.virtualbox.org/wiki/Downloads

- Step 1
  - Download the vagrant file
  - {Check your inbox}
- Step 2
  - Open cmd or gitbash
  - Go to the file location and run below command.
    - vagrant validate
    - vagrant plugin install vagrant-proxyconf
    - vagrant up
- Step 3
  - Your environment are ready to use
    - vagrant ssh hostvm

Vagrantfile.dat

For Power Shell Upgradation from Version 2 to 3

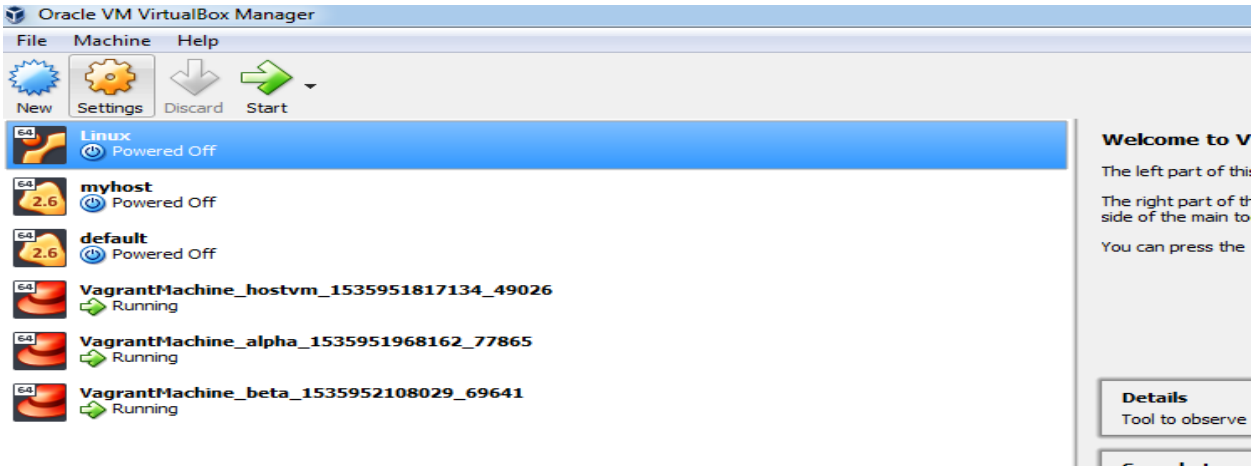
https://www.microsoft.com/en-us/download/details.aspx?id=34595

Vagrantfile

```
# -*- 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|

  if Vagrant.has_plugin?("vagrant-proxyconf")
    config.proxy.http     = "http://www-proxy.us.oracle.com:80/"
    config.proxy.https    = "http://www-proxy.us.oracle.com:80/"
    config.proxy.no_proxy = "localhost,127.0.0.1,.example.com"
  end


  config.vm.define :hostvm do |hostvm|
    hostvm.vm.box = "centos/7"
    hostvm.vm.network :private_network, ip: "10.0.0.10"
    hostvm.vm.hostname = "hostvm"
  end

  config.vm.define :alpha do |alpha|
    alpha.vm.box = "centos/7"
    alpha.vm.network :private_network, ip: "10.0.0.11"
    alpha.vm.hostname = "alpha"
  end

  config.vm.define :beta do |beta|
    beta.vm.box = "centos/7"
    beta.vm.network :private_network, ip: "10.0.0.12"
    beta.vm.hostname = "beta"
  end
end
```

After vagrant up run **vagrant plugin install vagrant-proxyconf**

To reload the machine with proxy: **vagrant reload hostvm**



To Know PowerShell version: $PSVersionTable.PSVersion

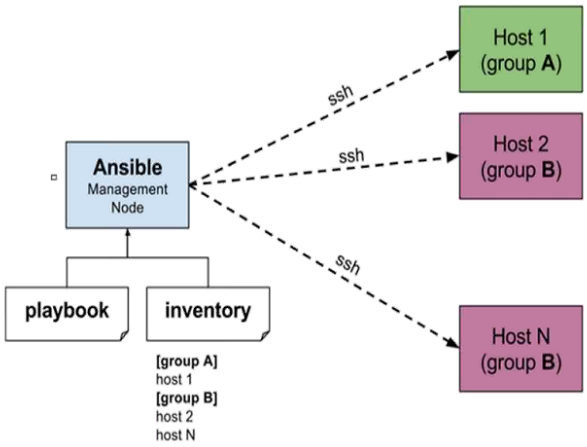
Why Anisble?

Working in IT, you're likely doing the same tasks over and over. What if you could solve problems once and then automate your solutions going forward? Ansible is here to help.

To avoid repetitive work Ansible is helpful.

Ansible Introduction

- Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.
- Why Do We Need Ansible?
  - PROVISIONING
  - COMPLETE IT AUTOMATION
  - APPLICATION DEPLOYMENT
  - CONTINUOUS DELIVERY
  - SECURITY & COMPLIANCE
  - ORCHESTRATION



It should be installed in a dedicated system where playbook and inventory should be installed

Basic Ansible Terminologies

- **Controller Machine**: The machine where Ansible is installed, responsible for running the provisioning on the servers you are managing.
- **Inventory**: An initialization file that contains information about the servers you are managing.
- **Playbook**: The entry point for Ansible provisioning, where the automation is defined through tasks using YAML format.
- **Task**: A block that defines a single procedure to be executed, e.g. Install a package.
- **Module**: A module typically abstracts a system task, like dealing with packages or creating and changing files. Ansible has a multitude of built-in modules, but you can also create custom ones.
- **Role**: A pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of a provisioning.

Basic Ansible Terminologies

- **Play**: A provisioning executed from start to finish is called a play. In simple words, execution of a playbook is called a play.
- **Facts**: Global variables containing information about the system, like network interfaces or operating system.
- **Handlers**: Used to trigger service status changes, like restarting or stopping a service.

SSH communication with Ansible

- Ansible has a default inventory file (**/etc/ansible/hosts**) used to define which remote servers.
- It will be managing. Our public SSH key should be located in **authorized_keys** on remote systems.

Ansible Architecture

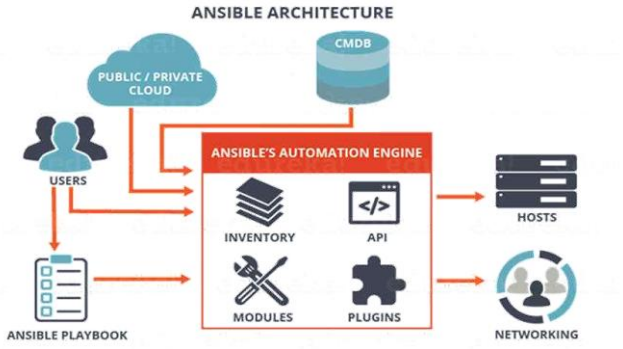
The Ansible automation engine has a direct interaction with the users who write playbooks to execute the Ansible Automation engine. It also interacts with cloud services and Configuration Management Database (CMDB).

The Ansible Automation engine consists of:

- Inventories
- APIs
- Modules
- Plugins
- Networking
- Hosts
- Playbooks
- CMDB and Cloud

Ansible Documentation

- Website [https://docs.ansible.com/]
- Inline help [ansible –help]



Installation of Ansible

Install Ansible and Configure (CentOS 7)

1. vagrant ssh hostvm
2. sudo su root
3. yum install epel-release
4. yum update
5. yum install –y vim git python python-devel python-pip openssl ansible
   - vim /etc/ansible/ansible.cfg – Review ansible configuration and uncomment hosts path and sudo_user=root.
   - vim /etc/ansible/hosts – Add or edit hosts
6. visudo – Edit the file and add below statement (perform this step on nodes as well)
   - vagrant ALL=(ALL) NOPASSWD: ALL
7. su vagrant -
8. ssh-keygen
9. ssh-copy-id vagrant@10.0.0.11 (run this on ansible host for all nodes and localhost)
10. ssh vagrant@10.0.0.11 (If ssh connection doesn't work do it manually)
11. ansible --version

1) To go inside any machine:

```
vagrant ssh <MachineName>
Ex: vagrant ssh hostvm
```

2) Sudo su root → Skip this step
3) Installing a repository:

```
sudo yum install epel-release
```

4) sudo yum update
5) sudo yum install -y vim git python python-devel python-pip openssl ansible
6) Go to /etc/ansible folder
   sudo vim ansible.cfg

```
inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
sudo_user      = root
#ask_sudo_pass = True
```

Click insert to edit. To save a file: **wq!**

In the host file: sudo vim hosts

```
[demoservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
10.0.0.11
10.0.0.12
# If you have multiple hosts foll
```

Click insert to edit. To save a file: **wq!**

7) **Skip this step**
8) **Go to /home/vagrant/.ssh/**
9) **sudo su vagrant -**
10) **ssh-keygen** to generate key. Make sure key is generated in /home/vagrant/.ssh/id_rsa.pub

```
[vagrant@hostvm .ssh]$ ssh-keygen
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:
SHA256:IEhPwmaxS1iSocrlJzZxQKhfO3yEBrjORyt9IqZeLxs vagrant@hostvm
The key's randomart image is:
+---[RSA 2048]----+
|o=B+.            |
|+*=*.            |
|++++oo.          |
|=.oo*...         |
|+o+O.+  S        |
| *o=B..          |
|o +Eoo           |
|. ..o            |
|.. .o.           |
+----[SHA256]-----+
```

Make sure key is generated in home/vagrant/.ssh folder not in root

Copy the generated key from host vm by going to **/home/vagrant/.ssh/id_rsa.pub**:
Copy the generated key

Login to alpha and beta machine to paste the generated key in the respective VM

1. vagrant ssh alpha
2. sudo vi /home/vagrant/.ssh/authorized_keys
   Edit the file to save the generated key

Login to beta machine to paste the generated key in the respective VM

1. vagrant ssh beta
2. sudo vi /home/vagrant/.ssh/authorized_keys
   Edit the file to save the generated key

```
bispatel@BISPATEL-IN MINGW64 /d/training_devops/vagrant_learning
$ vagrant ssh alpha
[vagrant@alpha ~]$ cd /home/vagrant/.ssh
[vagrant@alpha .ssh]$ ls
authorized_keys
[vagrant@alpha .ssh]$ sudo vi authorized_keys
[vagrant@alpha .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCsUFg7V/S17v4Qgln7oO/Cxl9PbzU37xQHK9+Da60yJJcAs0DfXFauOF8+Kon/Q/fGh6Vjzz5LL1juajlf4GQQkp8DboOPPJ3d/IJbCmioZnvx3k6s1bKhPIU2v+p2DYIQQ
GmTL13z7x9UYaxFd5+hwZbZtrlnY2KFBzug0XPXhaGlVNyDdFP6ha7dd3ZR5SVpwhHEJ8COHiUx8w9mG2aLH1akRv1muOQghh4gsFkTkv4iWS3g4Fm17CUrtTSdBnL7KcMeEAxm+ZBKGSl3gGn9C13KooNU+lxL2cQajiwGuq
LVGnXKLZrhY2m4EWyS/BDc883OJludLlM1tZW5 vagrant

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCw9myH3EQO8d2AYfXXGn6HuGgrz1kDUs8nHWWtPQSfcFRfMbj6G83BOW+RLTdm+jBI9am4BuVbZXsqbY4gvHwpy5O/MJe7jqgYCS1kihdIYuObG8NnsMszTJXeVpU8+GuxQ
gaDro4zhPJSo4ct7uYQWvZKrMExqFbzL2NFHaAURAbQVGbu1y48zLhzhxregr9jPmmnMPmKOOlQr/8sM7sBaVysE6AXWs3qSs5sru/IU/8WQAEh4CL7Bbie7CE2MUgeFwgKnXG4a1ZOyNQEGe57rFMr/72mdFwnVN+3+kaRRL
ZspXE/vMOaqMRIu5T+q71HtXo48/u6mmx+CYYr vagrant@hostvm

[vagrant@alpha .ssh]$ exit
logout
Connection to 127.0.0.1 closed.

bispatel@BISPATEL-IN MINGW64 /d/training_devops/vagrant_learning
$ vagrant ssh beta
[vagrant@beta ~]$ cd /home/vagrant/.ssh
[vagrant@beta .ssh]$ ls
authorized_keys
[vagrant@beta .ssh]$ sudo vi authorized_keys
[vagrant@beta .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDxY8zI67rDLvc7xApxZPS1A/I9DIcyVQEgvwg+LJ/oMEJxvnmRbKMd6tlvNQu5uJigl/VZobTAFjk92QTRGK3R5iFhHRnmKcJmiHgs3QYRRM4G/Q5OoK6mbE1CD/EpjyTXz
7TRlveID4LJXHK5NZWsIFXau9EVvsxRoLv05EroqxrF+3Omr1O1mFZMJyaozNel15y8kQLX+IS06cuhO7aax7YNcd3LA5tmBBJjwwfSKJfWM8VSsfbyXbdN1GQAqVcLCcue6GIgf3/HftILjUnhd1vR+DRcuVhQZ9aLLWzVHt
1SxrwcJJN6Aqhj3etknMyxM4N8CX0wclWH/y8p vagrant

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCw9myH3EQO8d2AYfXXGn6HuGgrz1kDUs8nHWWtPQSfcFRfMbj6G83BOW+RLTdm+jBI9am4BuVbZXsqbY4gvHwpy5O/MJe7jqgYCS1kihdIYuObG8NnsMszTJXeVpU8+GuxQ
gaDro4zhPJSo4ct7uYQWvZKrMExqFbzL2NFHaAURAbQVGbu1y48zLhzhxregr9jPmmnMPmKOOlQr/8sM7sBaVysE6AXWs3qSs5sru/IU/8WQAEh4CL7Bbie7CE2MUgeFwgKnXG4a1ZOyNQEGe57rFMr/72mdFwnVN+3+kaRRL
ZspXE/vMOaqMRIu5T+q71HtXo48/u6mmx+CYYr vagrant@hostvm
```

11) In location /home/vagrant/.ssh/
ssh-copy-id vagrant@10.0.0.11

```
[vagrant@hostvm .ssh]$ ssh-copy-id vagrant@10.0.0.13
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '10.0.0.13 (10.0.0.13)' can't be established.
ECDSA key fingerprint is SHA256:QQoMCUxznGOPaENElNWQGm7gTl3XECCYsECL65MYmrc.
ECDSA key fingerprint is MD5:6a:d5:af:6d:1f:b7:c7:43:63:47:08:58:2f:91:df:4e.
Are you sure you want to continue connecting (yes/no)? yes
/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: WARNING: All keys were skipped because they already exist on the remote system.
                (if you think this is a mistake, you may want to use -f option)

[vagrant@hostvm .ssh]$ ssh-copy-id vagrant@10.0.0.14
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '10.0.0.14 (10.0.0.14)' can't be established.
ECDSA key fingerprint is SHA256:xlTmhiYWaLdii8rSCD4W92GhWv3tX/djW/jjL3PRkZM.
ECDSA key fingerprint is MD5:0d:3a:78:6f:57:09:e8:81:7f:d2:1d:db:54:b8:79:84.
Are you sure you want to continue connecting (yes/no)? yes
/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: WARNING: All keys were skipped because they already exist on the remote system.
                (if you think this is a mistake, you may want to use -f option)
```

12) ssh vagrant@10.0.0.11

```
[vagrant@hostvm .ssh]$ ssh vagrant@10.0.0.13
Last login: Tue Sep  4 08:22:28 2018 from 10.0.2.2
[vagrant@alpha ~]$ exit
logout
Connection to 10.0.0.13 closed.
[vagrant@hostvm .ssh]$ ssh vagrant@10.0.0.14
Last login: Tue Sep  4 08:23:47 2018 from 10.0.2.2
[vagrant@beta ~]$ exit
logout
Connection to 10.0.0.14 closed.
```

13) ansible --help

Ansible ad hoc commands

1. ansible all –a “ls –al /home/ansible”
2. ansible all **–s** –a “cat var/log/messages” (-s for sudo)
3. ansible webservers –m **copy** –a “src=test.xml dest=/temp/test.xml” (copy module)
4. ansible webserver **–s** –m **user** –a “name=test” (user module)
5. ansible webservers –m setup **–a ‘filter=*ipv4*’** (Gathering facts and filtering it)

```
[vagrant@hostvm .ssh]$ ansible all -m ping
[DEPRECATION WARNING]: DEFAULT_SUDO_USER option, In favor of Ansible B
removed in version 2.8. Deprecation warnings can be disabled by settin
10.0.0.14 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
10.0.0.13 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Execute the above commands in hostvm

To create User:

ansible demoservers -s -m user -a "name=test"

ansible all -s -a "cat /var/log/messages"

ansible all -s -a "cat /vagrant/Vagrantfile"

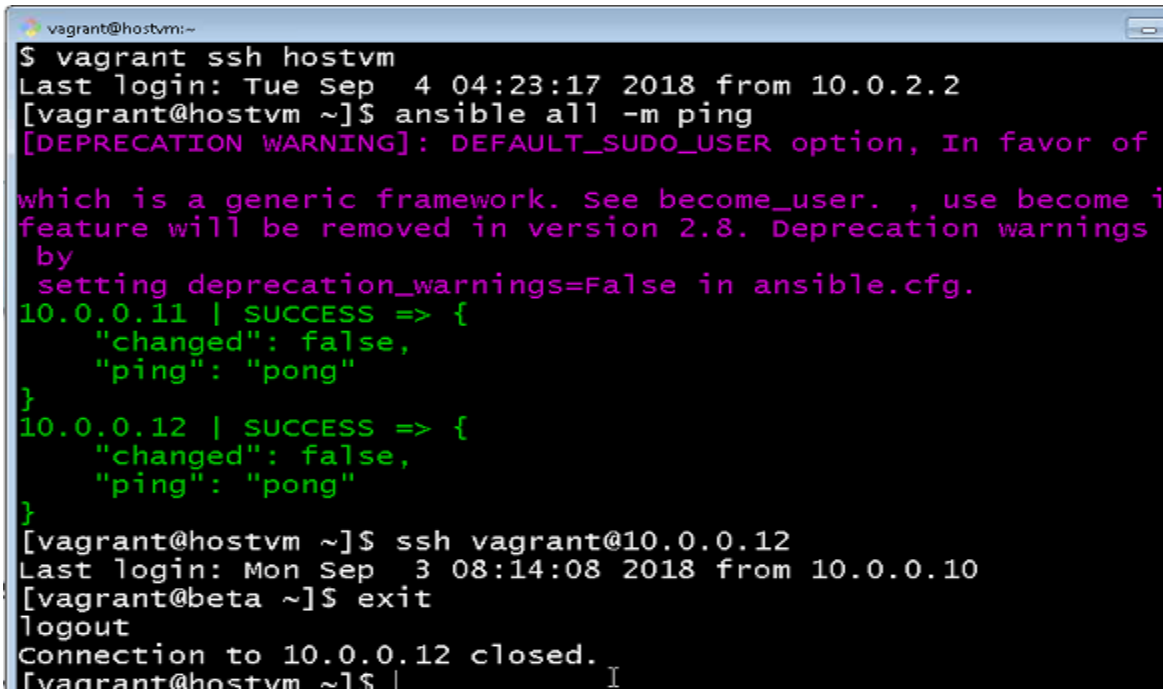
ansible demoservers -m setup -a 'filter=*ipv4*'

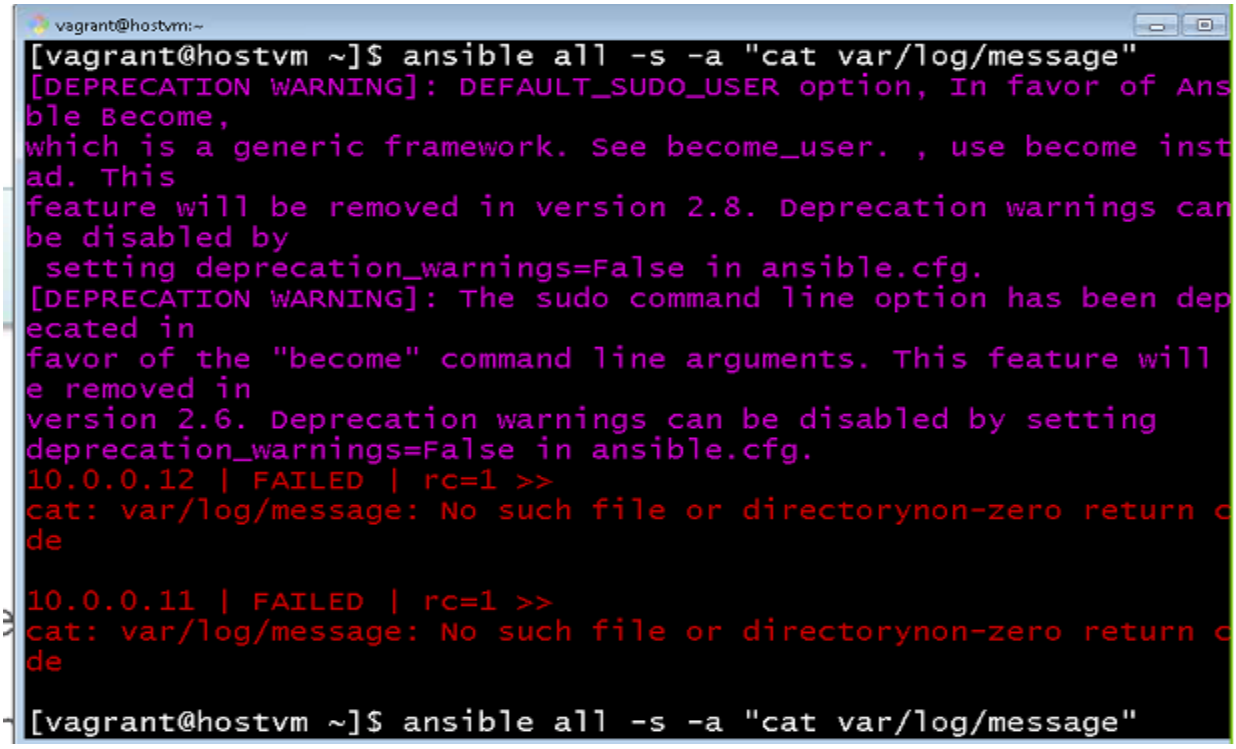
Yaml Introduction

- YAML is a data serialization language designed to be human-readable and working well with modern programming languages for everyday tasks.

- Specifically, we could note the list of ingredients for a breakfast as follows:

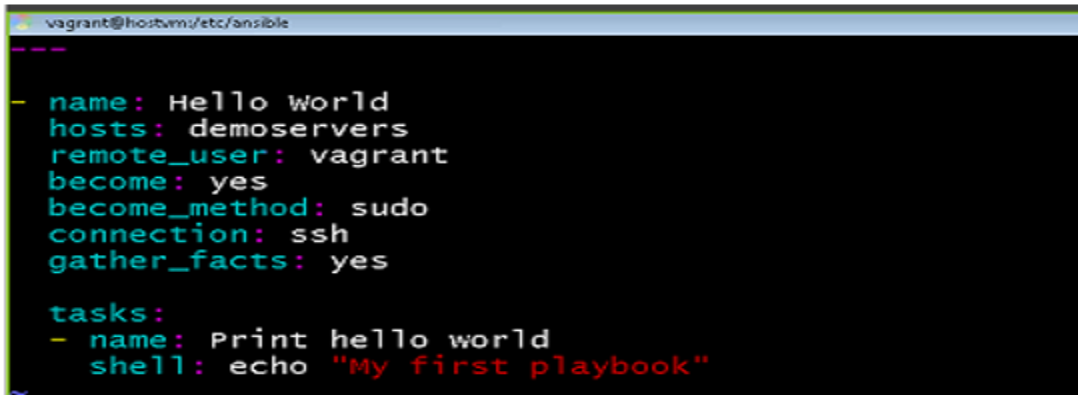
```
- croissants
- chocolate breads
- ham
- eggs
```



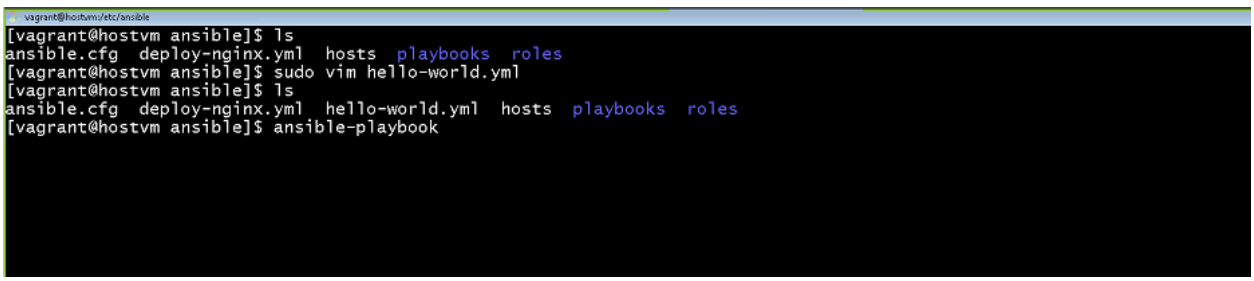


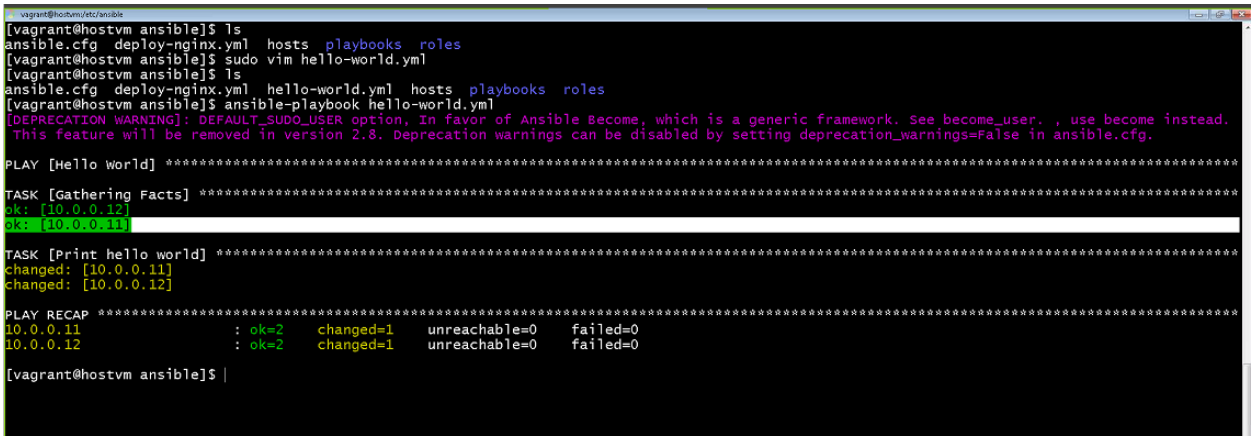
Creating a playbook create a yml file in /etc/ansible folder:

1. sudo vim hello-world.yml



Run the playbook





Creating advanced Playbook: **YAML FILE**

```
---
- name: Install Apache
  hosts: demoservers
  remote_user: vagrant
  become: yes
  become_method: sudo
  connection: ssh
  gather_facts: yes
  tasks:
  - name: Print hello world
    shell: echo "My first playbook"
  - name: Install httpd
    yum:
      name: httpd
      state: latest
    notify:
       startservice
  handlers:
  - name: startservice
    service:
      name: httpd
      state: restarted
```

```
vagrant@hostvm:/etc/ansible
---

- name: Install Apache
  hosts: demoservers
  remote_user: vagrant
  become: yes
  become_method: sudo
  connection: ssh
  gather_facts: yes

  tasks:
  - name: Print hello world
    shell: echo "My first playbook"

  - name: Install httpd
    yum:
      name: httpd
      state: latest
    notify:
        startservice

  handlers:
  - name: startservice
    service:
      name: httpd
      state: restarted
```

```
[vagrant@hostvm ansible]$ sudo vim hello-world.yml
[vagrant@hostvm ansible]$ sudo vim hello-world.yml
[vagrant@hostvm ansible]$ ansible-playbook hello-world.yml
[DEPRECATION WARNING]: DEFAULT_SUDO_USER option, In favor of Ansible Become, which is a generic framework. See become_user. , use become i
 This feature will be removed in version 2.8. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [Install Apache] **********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [10.0.0.12]
ok: [10.0.0.11]

TASK [Print hello world] *******************************************************************************************************************
changed: [10.0.0.12]
changed: [10.0.0.11]

TASK [Install httpd] ***********************************************************************************************************************
changed: [10.0.0.12]
changed: [10.0.0.11]

RUNNING HANDLER [startservice] *************************************************************************************************************
changed: [10.0.0.12]
changed: [10.0.0.11]

PLAY RECAP *********************************************************************************************************************************
10.0.0.11                  : ok=4    changed=3    unreachable=0    failed=0
10.0.0.12                  : ok=4    changed=3    unreachable=0    failed=0
```

Command: **`ansible-playbook hello-world.yml`**

Parameterizing the packages:

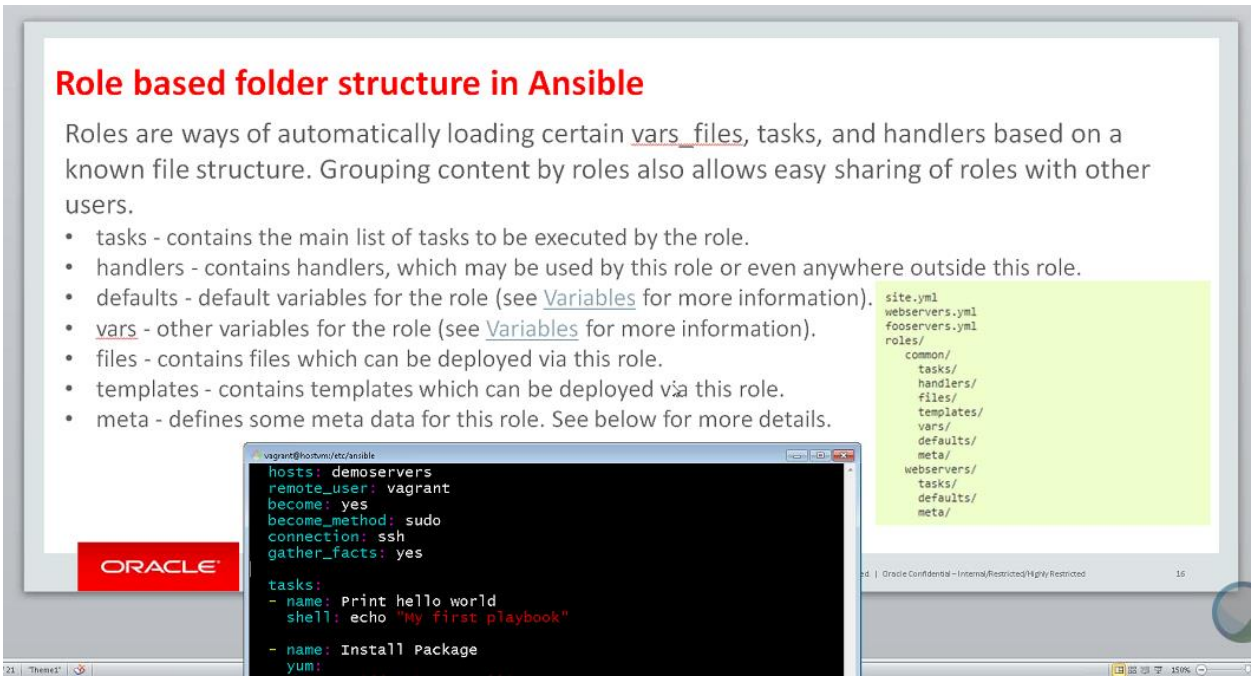
```
vagrant@hostvm:/etc/ansible
---

- name: Install Package using yum
  hosts: demoservers
  remote_user: vagrant
  become: yes
  become_method: sudo
  connection: ssh
  gather_facts: yes

  tasks:
  - name: Print hello world
    shell: echo "My first playbook"

  - name: Install Package
    yum:
      name: '{{packagename}}'
      state: latest
    notify:
        startservice

  handlers:
  - name: startservice
    service:
      name: httpd
      state: restarted
```

To supply parameters to file

```
[vagrant@hostvm ansible]$ ansible-playbook hello-world.yml --extra-vars "packagename=httpd"
```

Command: **ansible-playbook hello-world.yml --extra-vars "packagename=httpd"**

Roles:



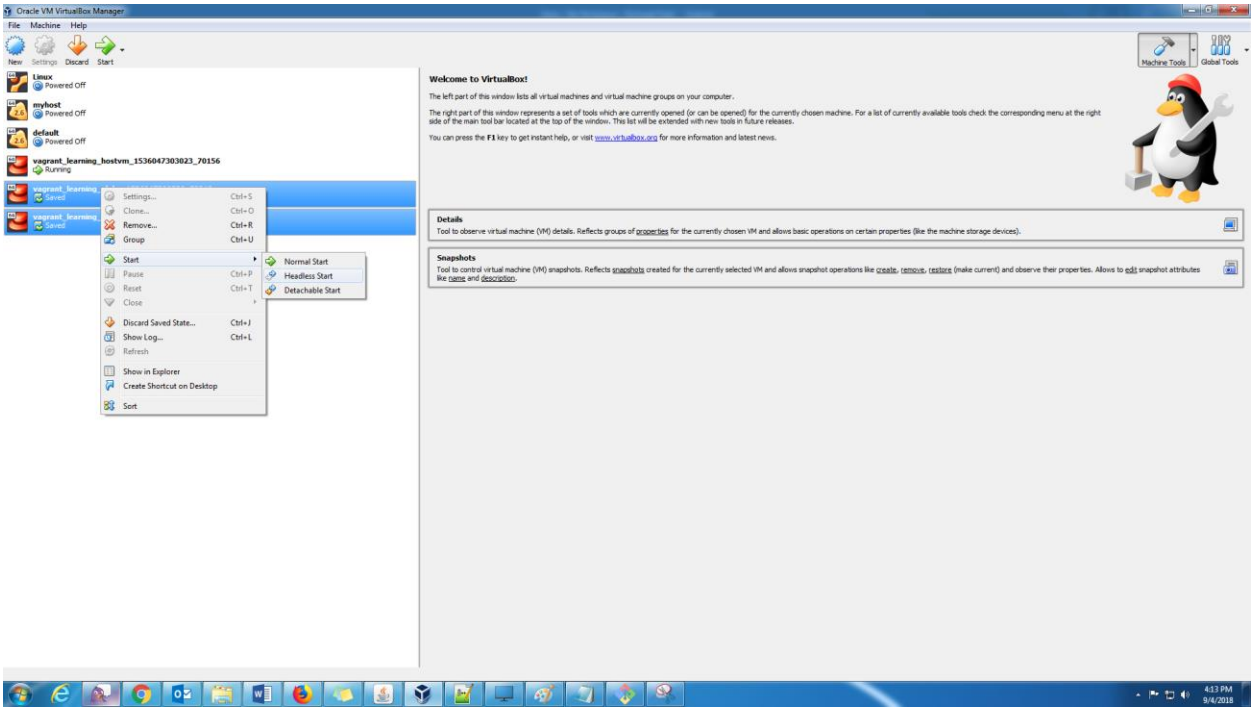
Task is the compulsory folder

```
[vagrant@hostvm ansible]$ sudo vim hello-world.yml
[vagrant@hostvm ansible]$ sudo vim hello-world.yml
[vagrant@hostvm ansible]$ ls -ltr
total 32
drwxr-xr-x. 2 root root     6 Aug 17 21:06 roles
-rw-r--r--. 1 root root 19547 Sep  3 06:56 ansible.cfg
drwxr-xr-x. 4 root root   101 Sep  4 04:33 playbooks
-rw-r--r--. 1 root root   415 Sep  4 04:38 deploy-nginx.yml
-rw-r--r--. 1 root root   960 Sep  4 04:58 hosts
-rw-r--r--. 1 root root   452 Sep  4 06:04 hello-world.yml
[vagrant@hostvm ansible]$ cd playbooks/
[vagrant@hostvm playbooks]$ ls -ltr
total 16
drwxr-xr-x. 2 root root   28 Sep  4 04:33 group_vars
drwxr-xr-x. 4 root root   35 Sep  4 04:33 roles
-rw-r--r--. 1 root root 1104 Sep  4 04:33 README.md
-rw-r--r--. 1 root root  217 Sep  4 04:33 LICENSE.md
-rw-r--r--. 1 root root  159 Sep  4 04:33 site.yml
-rw-r--r--. 1 root root   33 Sep  4 04:33 hosts
[vagrant@hostvm playbooks]$ cd roles/
[vagrant@hostvm roles]$ ls
selinux  tomcat
[vagrant@hostvm roles]$ cd ..
[vagrant@hostvm playbooks]$ vim site.yml
[vagrant@hostvm playbooks]$ cd roles/
[vagrant@hostvm roles]$ cd tomcat/
[vagrant@hostvm tomcat]$
```

```
[vagrant@hostvm playbooks]$ ls -ltr
total 16
drwxr-xr-x. 2 root root   28 Sep  4 04:33 group_vars
drwxr-xr-x. 4 root root   35 Sep  4 04:33 roles
-rw-r--r--. 1 root root 1104 Sep  4 04:33 README.md
-rw-r--r--. 1 root root  217 Sep  4 04:33 LICENSE.md
-rw-r--r--. 1 root root  159 Sep  4 04:33 site.yml
-rw-r--r--. 1 root root   33 Sep  4 04:33 hosts
[vagrant@hostvm playbooks]$ ansible-playbook site.yml
[DEPRECATION WARNING]: DEFAULT_SUDO_USER option, In favor of Ans
 This feature will be removed in version 2.8. Deprecation warnin
 [WARNING]: Could not match supplied host pattern, ignoring: web

PLAY [webservers] ***************************************************
skipping: no hosts matched

PLAY RECAP **********************************************************

[vagrant@hostvm playbooks]$
```

To validate the YML file:

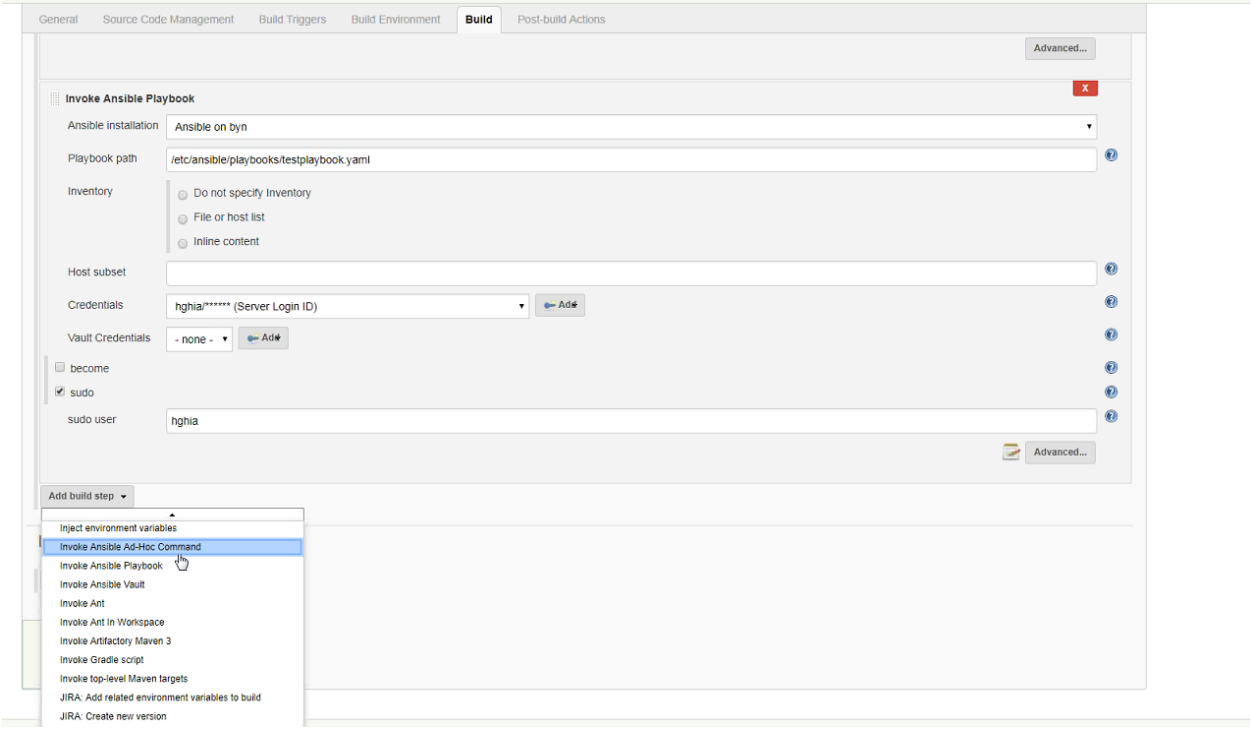
http://www.yamllint.com

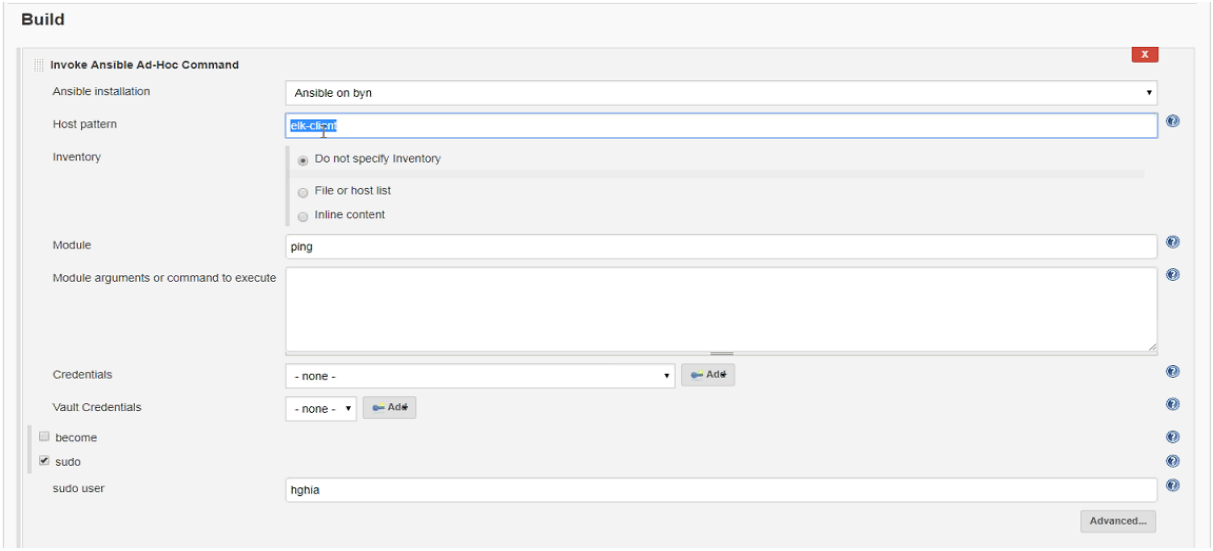


Always do Headless start

Ansible setup in Jenkins:

Ansible plugin





How to share files from windows: Create a folder in vagrant folder and it will be shared with vm