Creating Lab environment for Ansible

By – **Yathish Nagaraj**
Infra Dev Specialist
Automation PS Team

Components

Vagrant

Vagrant: Virtual Machine Controller

Define VM's to startup, and initial configs (ip, hostname, etc)

VirtualBox

VirtualBox: Virtual Machine Provider

Environment to run virtual machines

Ansible

Ansible: Automation / Provisioning

Application to push configuration and automation to remote systems

Install Vagrant

Go to :

www.vagrantup.com to download the installation package of your respective OS family

Go to :

Www.vagrantcloud.com to pick the VM that has to be built.

Install VirtualBox

Go to :

www.virtualbox.org to download the installation package of your respective OS family

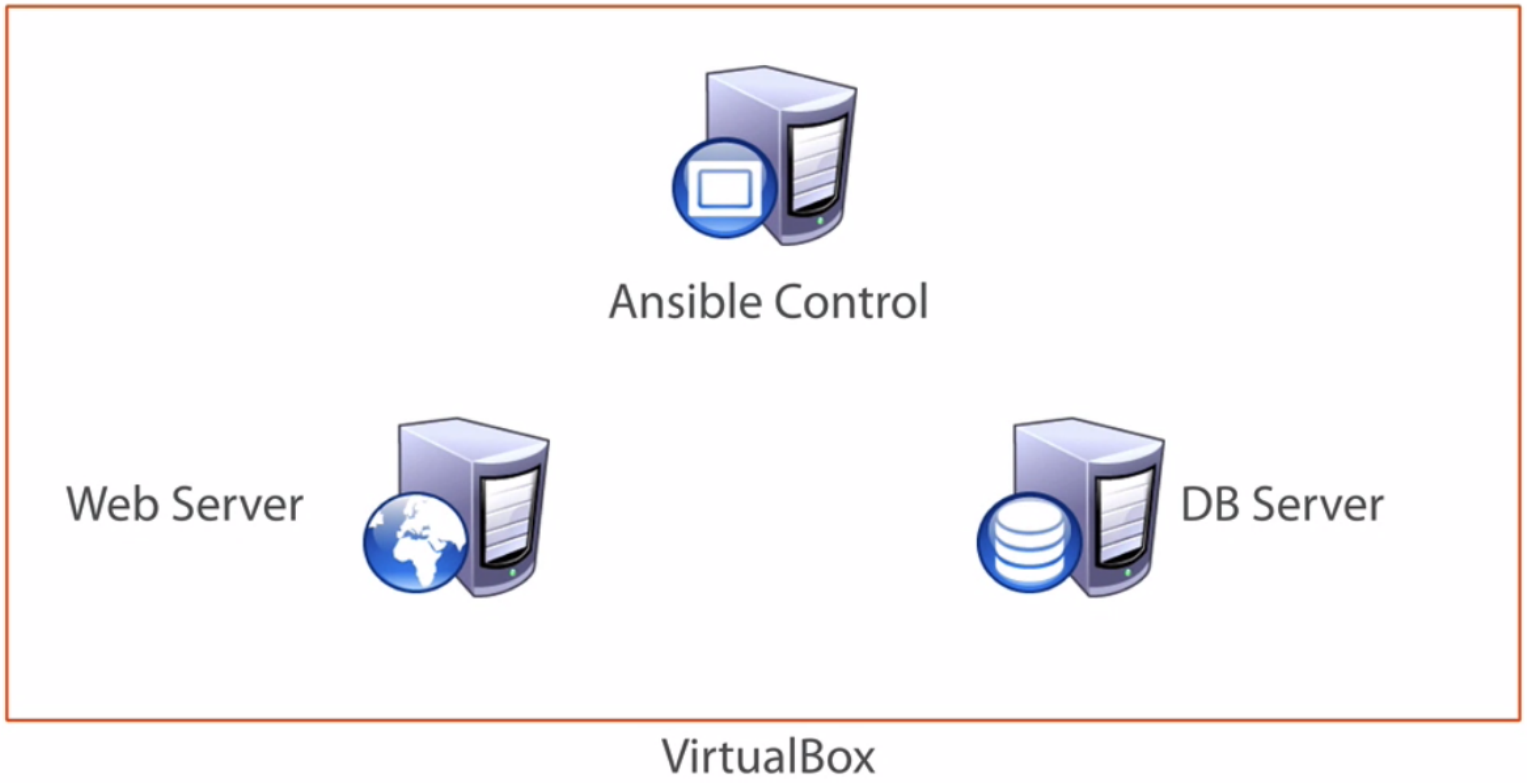
Execute :

“vboxmanage list runningvms” to get the list of Vms running in virtualBox.

Demonstration of creating the ansible lab environment

Lets go set-up the Ansible lab environment.

Ansible Lab



Install Ansible on Debian linux

Install Ansible (Debian)

```
$ sudo apt-get install ansible
```

Install Ansible on CentOS linux

Install Ansible (CentOS)

```
$ sudo yum install epel-release
$ sudo yum install ansible
```

Basic Ansible Command syntax

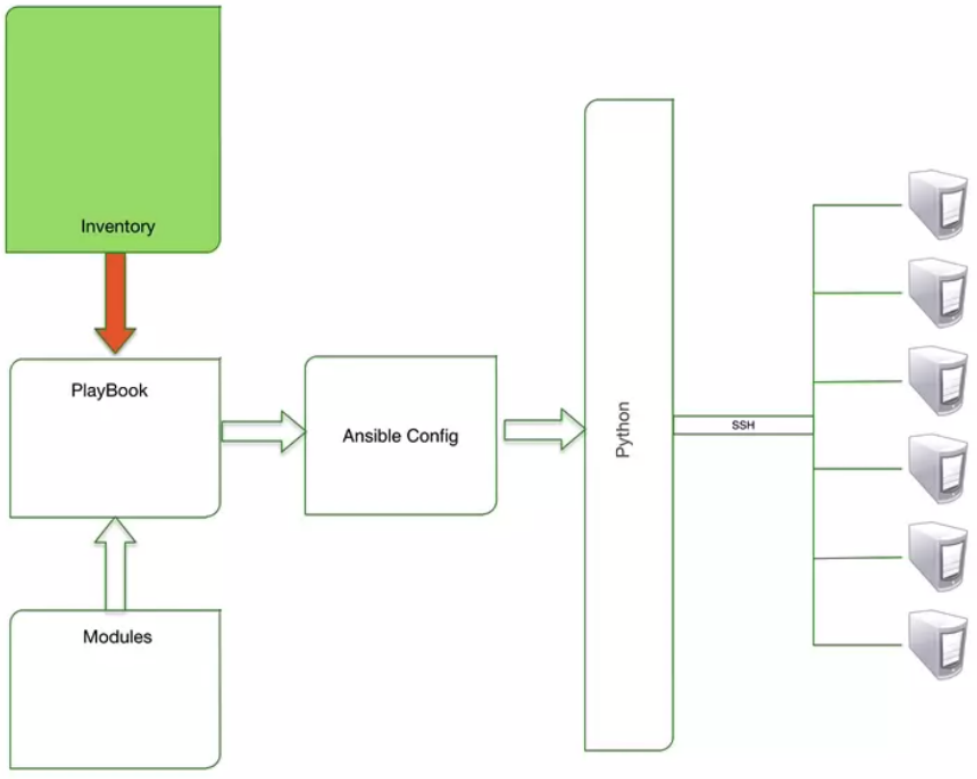
Include system, inventory file, module, and user

Basic Ansible Command

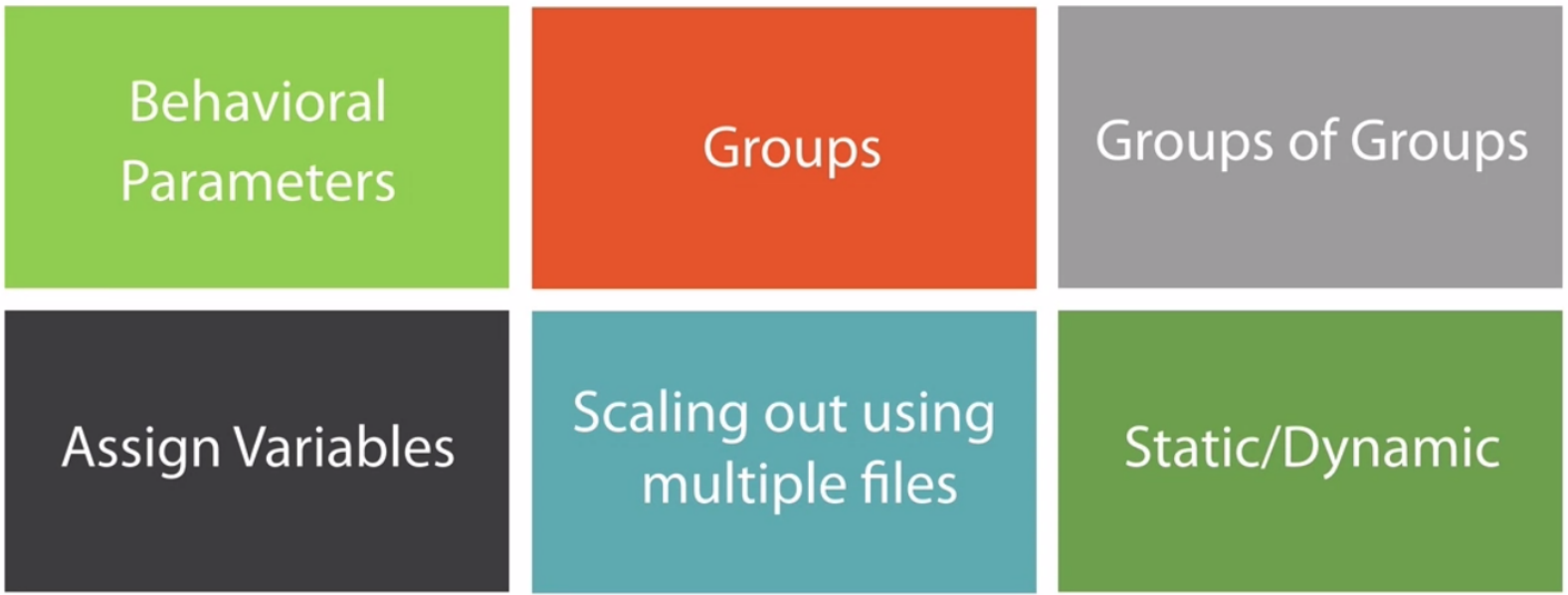
```
$ ansible <system>
  -i <inventoryFile>
  -m <module>
  -u <username>
  -k <password prompt>
  -v ( -vv debug level2 / -vvv debug level3)
```

Ansible Inventory and Configurations

Ansible Inventory



Ansible Inventory features



Ansible Inventory file

```
[db]
db1.company.com ansible_ssh_user=aaron ansible_ssh_pass=123
db2.company.com ansible_python_interpreter=/usr/bin/python

[datacenter-west:children]
db
[datacenter-west:vars]
ansible_ssh_user=ansible_user
ansible_ssh_pass=#45e!@Gh
ntp-server=5.6.7.8
```

Demonstration on Inventory file creation

- Add Behavioral Parameters
- Create host-based variables
- Create a Group
- Create group-based variables

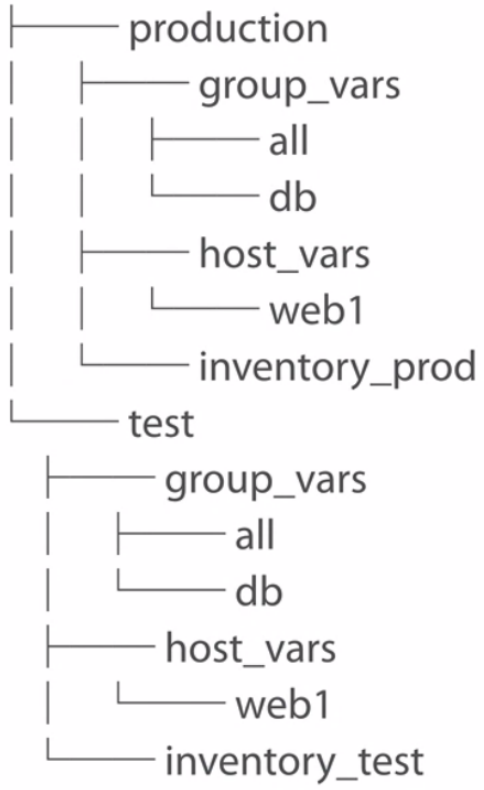
Scaling-out Inventory files

Using Directories

Can use to break-out long-running inventory files.

Very useful when dealing with large environments.

Multi-Environment Directory Structure



Order-of-Operations (Precedence)

1). (Group_Vars) All
2). (Group_Vars) GroupName
3). (Host_Vars) HostName

Variable file example

```
---
# file: group_vars/dc1-west
ntp: ntp-west.company.com
syslog: logger-west.company.com
```

Variable files are written in YAML

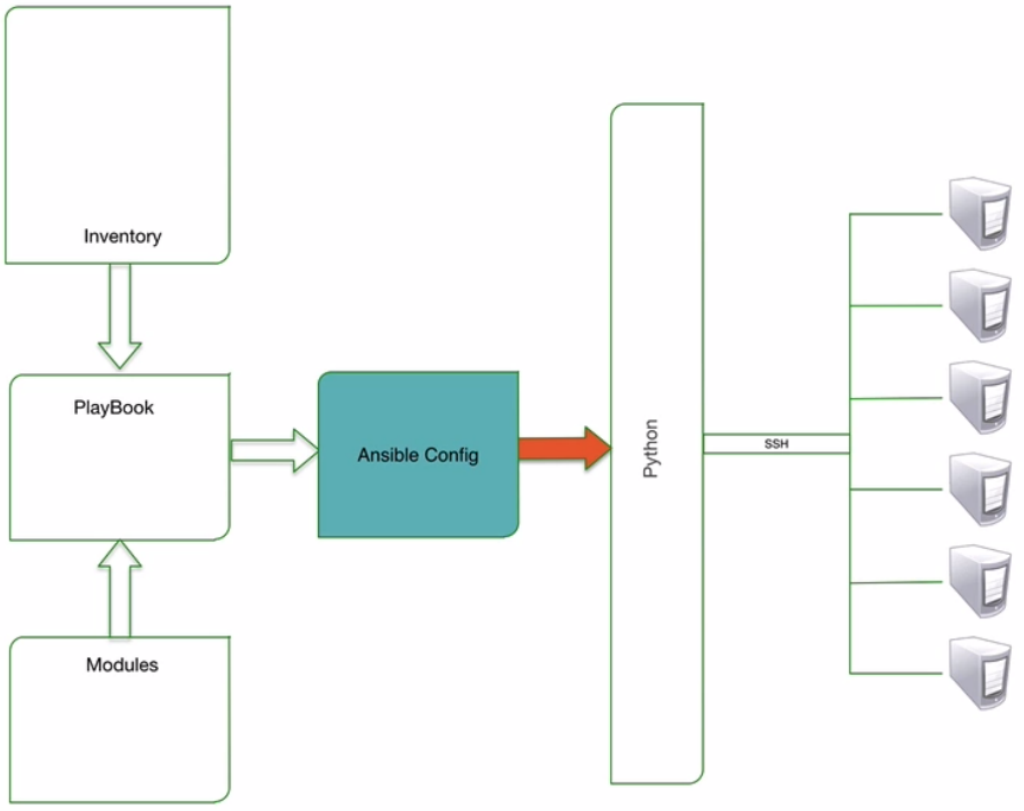
Comments use #

Key : Value pairs

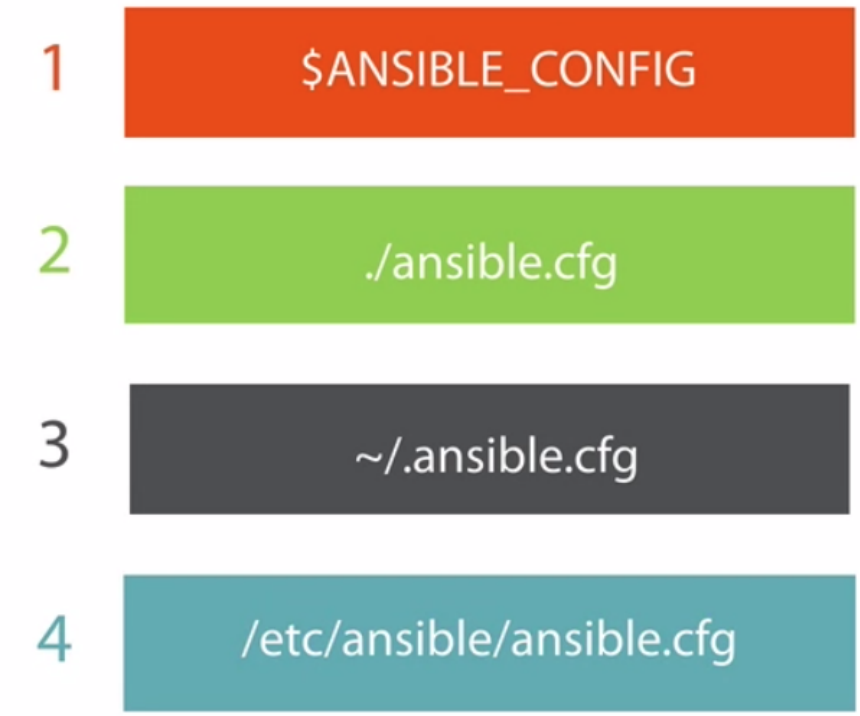
Demonstration on Scaling-out Inventory files

- Create Group Variables in separate file
- Show Order-of-Precedence

Ansible Configuration Basics



Configuration Settings Order-of-Operations



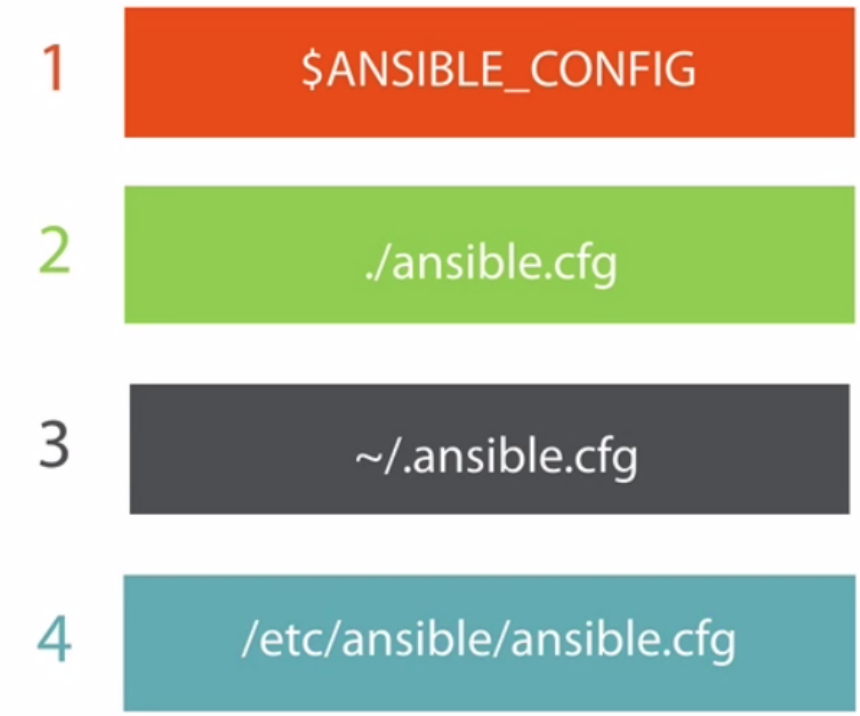
Configuration Settings Order-of-Operations

[defaults] forks	Default set to 5 Total number of parallel operations Ansible executes Production Recommendation: 20 Start at 20, and go up or down depending on performance
[defaults] host_key_checking	Default set to True For Production environments, do not change Development Environment: set to False Due to the dynamic environment of Dev, keeps it easy
[defaults] log_path	Default set to Null Write information on Ansible executions Set path to log file Make sure all users of Ansible has permissions to write

Demonstration on Ansible Configuration

- Define settings in configuration file
- Override setting in environment variable

Configuration Settings Order-of-Operations



Configuration Settings Order-of-Operations