**Hands-on Ansible**

<https://www.pluralsight.com/courses/hands-on-ansible>

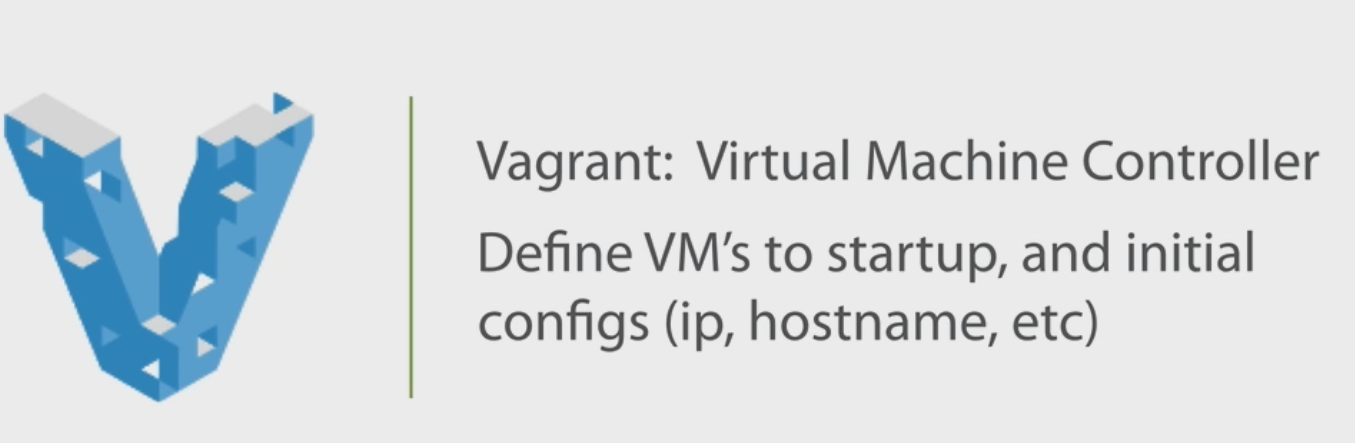
**-2- Creating environment**

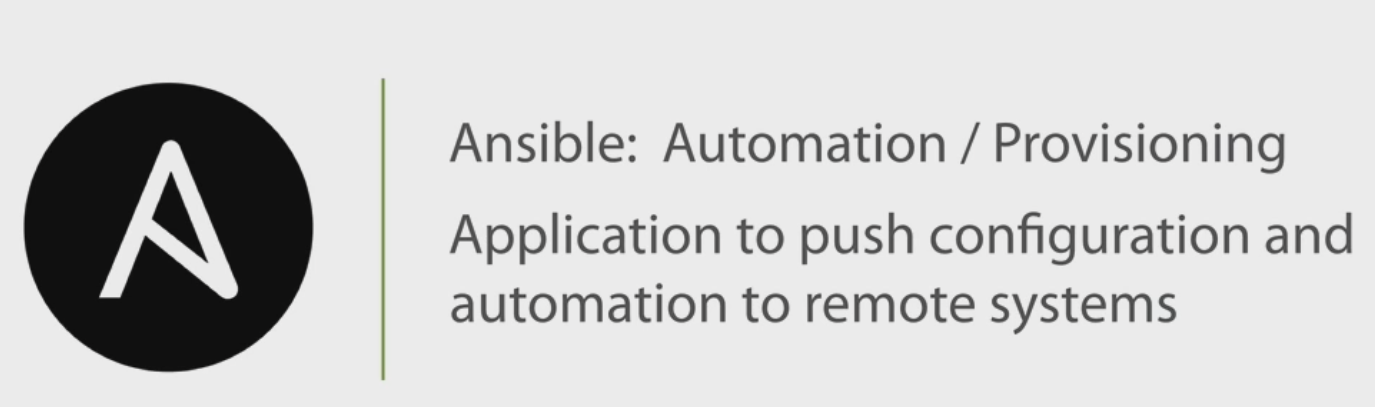
**- Vagrant Installation**

**- Virtual Box Installation**

**- Ansible Installation**

* **Vagrant** is our system controller
* **Virtual Box**: Virtual machine provider

****

****

**Configure Vagrant:**

> mkdir ansible-hands-one

> cd ansible-hands-one

> vagrant init

> This will create a Vagrantfile

- We will set up 3 Servers:

- ACS server. (Ansible Control Server)

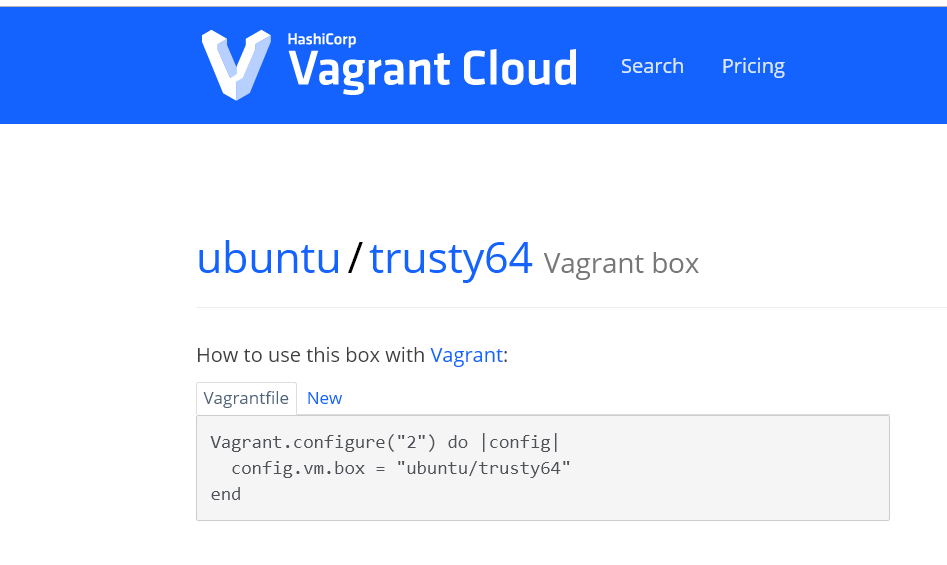
- Web Server

- DB Servers

- We copy this ubuntu config into our file.

1- Our ACS server will be running in an ubuntu.

If you don’t have these images in your repo, vagrant will go out and download them.



2- Our Web Server will be running in a CentOS 7.

* Keep in mind, this is web Server. We will be testing if we can see Web Pages or not.
* If I have multiple Web Servers running in a VM, if I go to access one, most of them, but one.. will be already in use.
  + **Solution: Port Forwarding**

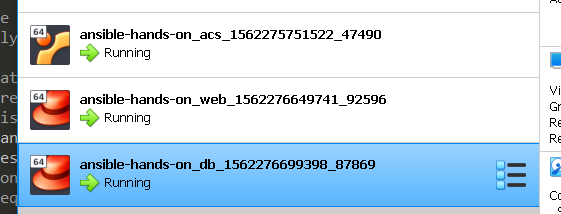


**This will allow accessing port 80 on the guest via port 8080 on the host.**

Whenever I access port 8080 on my machine, it will be forwarded to 80 inside of this host.

<https://www.vagrantup.com/docs/networking/forwarded_ports.html>

* *vagrant up*



**Installing Ansible on the ACS Server - Ubuntu**

* *Vagrant ssh acs*
* $ sudo apt update
* $ sudo apt install software-properties-common
* $ sudo apt-add-repository --yes --update ppa:ansible/ansible
* $ sudo apt install ansible

**On the WEB Server - CentOS**

*λ vagrant ssh web*

*[vagrant@web ~]$ sudo yum install epel-release*

*[vagrant@web ~]$ sudo yum install ansible*

* This will install the repository that has all the enterprise Linux software’s on it.
* **On the DB Server – CentOS**
  + - We will try to compile python on the DB Sever.
    - We need the GCC to compile C files.

*[vagrant@db ~]$ sudo yum install python-setup-tool*

* This will allow us to install and download ansible source code.

*[vagrant@db ~]$ sudo easy\_install pip*

*[vagrant@db ~]$ sudo yum install python-devel*

*[vagrant@db ~]$ sudo pip install ansible*

*Download the source code of Ansible and compile it.*

**04 Testing Lab with Your First Ansible Commands**

* Go to ACS
  + *λ vagrant ssh acs*
* Create an **inventory** files with 2 @ip addresses
  + *192.168.33.20*
  + *192.168.33.30*
* Ping a remote server with ansible

*>> ansible 192.168.33.20 -i inventory -u vagrant -m ping -k*

* + *Password prompt: vagrant*

**192.168.33.20 | FAILED => to use the 'ssh' connection type with passwords, you must install the sshpass program**

*>> sudo apt-get install* ***sshpass***

*>> ansible 192.168.33.20 -i inventory -u vagrant -m ping -k*

* + *Password prompt: vagrant*

***192.168.33.20 | FAILED => Using a SSH password instead of a key is not possible because Host Key checking is enabled and sshpass does not support this. Please add this host's fingerprint to your known\_hosts file to manage this host.***

**Ssh fingerprint:**

*>> ssh* [*vagrant@192.168.33.20*](mailto:vagrant@192.168.33.20)

***Permission denied (publickey,gssapi-keyex,gssapi-with-mic).***

* We need to enable some options in the **target server web**
* From you target server **‘’web’’**:

**Enable root login over SSH:**

1. As root, edit the sshd\_config file in /etc/ssh/sshd\_config:

nano /etc/ssh/sshd\_config

1. Add a line in the Authentication section of the file that says *PermitRootLogin and PasswordAuthentication* ***yes***. This line may already exist and be commented out with a "#". In this case, remove the "#".

# Authentication:

#LoginGraceTime 2m

**PermitRootLogin yes**

**PasswordAuthentication yes**

#StrictModes yes

#MaxAuthTries 6

#MaxSessions 10

**[vagrant@web ~]$ sudo /bin/systemctl restart sshd.service**

**[vagrant@web ~]$ sudo /bin/systemctl status sshd.service**

**>>> vagrant@acs:~$ ansible 192.168.33.20 -i inventory -u vagrant -m ping -k -vvv**

**192.168.33.20 | SUCCESS => {**

**"ansible\_facts": {**

**"discovered\_interpreter\_python": "/usr/bin/python"**

**},**

**"changed": false,**

**"invocation": {**

**"module\_args": {**

**"data": "pong"**

**}**

**},**

**"ping": "pong"**

**}**

Another way to do is to use a COMMAND module - There is also a Shell module - See difference:

[**https://docs.ansible.com/ansible/latest/modules/shell\_module.html**](https://docs.ansible.com/ansible/latest/modules/shell_module.html)

**# Restart all machines**

**>> ansible all -i inventory -u vagrant -m command -a "/sbin/shutdown -h"**

**# Update all machines**

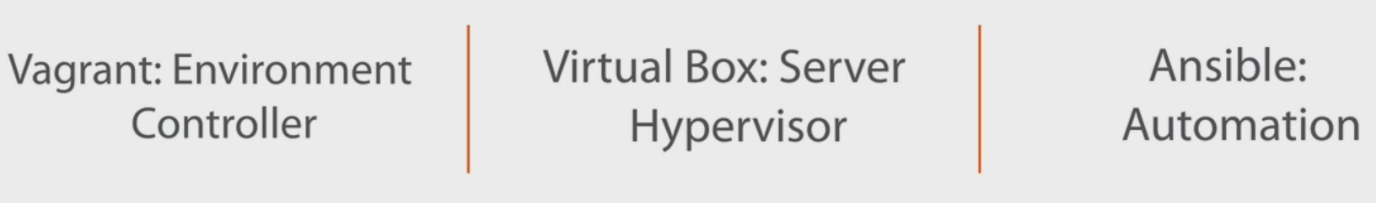
**>> ansible all -i inventory -u vagrant -m command -a "/ures/sbin/ yum update -u"**

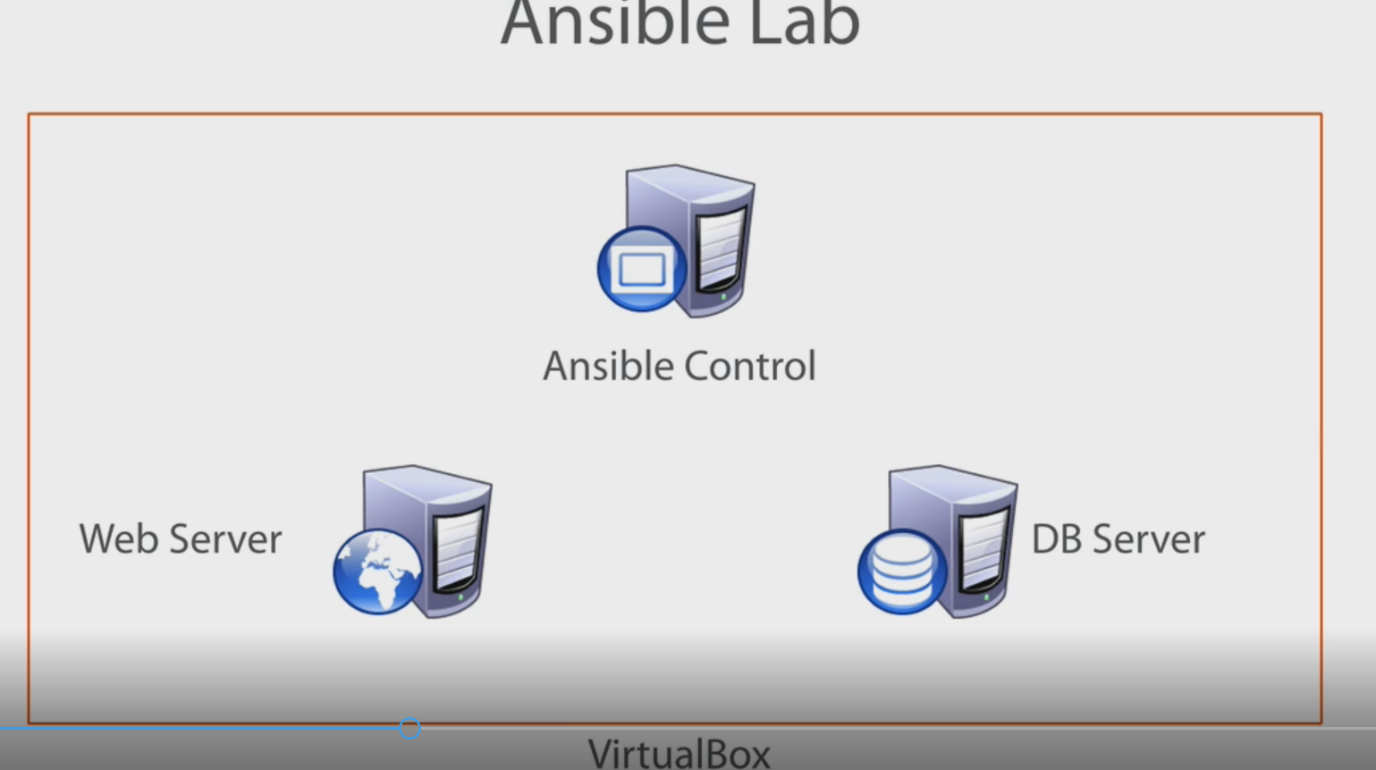
Since COMMAND module is the default you can just skip command:

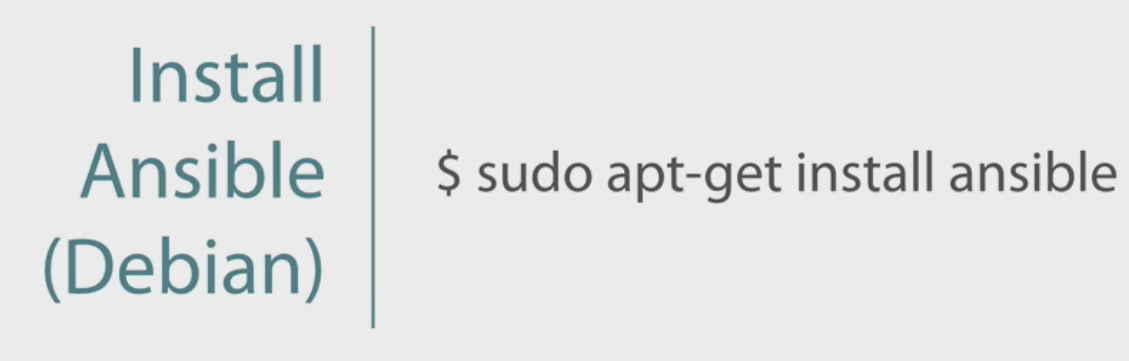
**>> ansible all -i inventory -u vagrant -a "/ures/sbin/ yum update -u"**

**RECAP-**

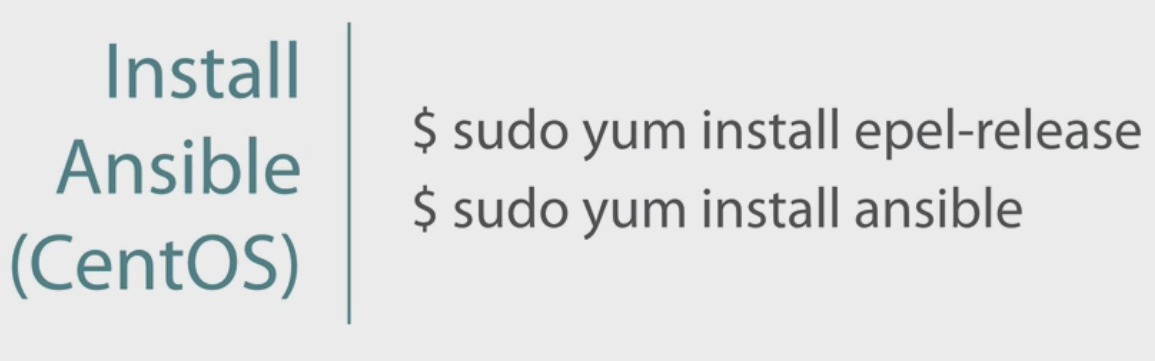
* Vagrant: Downloading and preconfiguring VM’s
* VBOX run VM’s
* Ansible automation: Running some commands







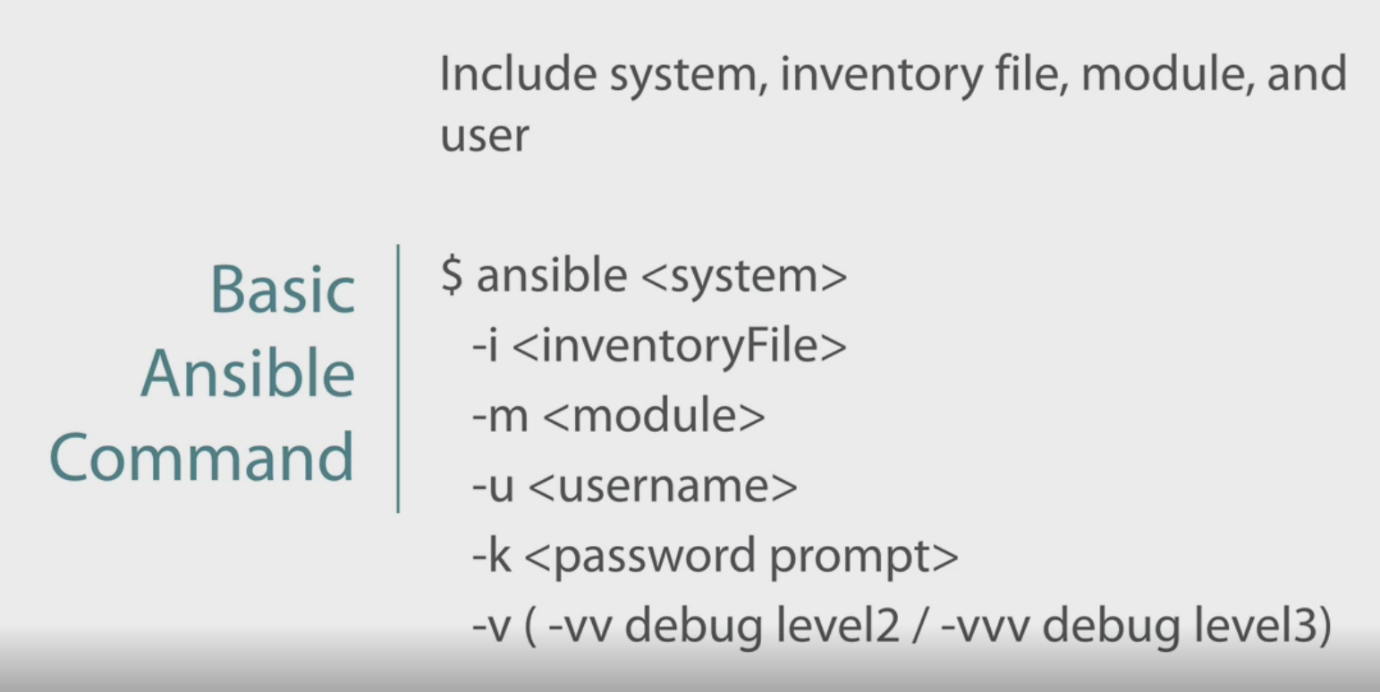
Install repository first - CentOS



You can install Ansible from **the Source code:**

****

**Some Ansible module commands:**

****