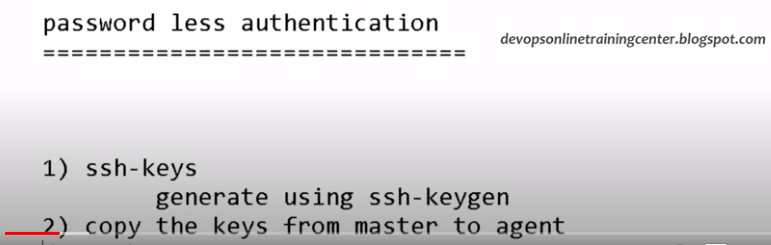
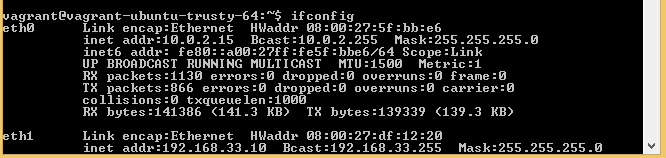
<https://www.youtube.com/watch?v=MwUSCcu9cZU&list=PL5nViEmyYI0bmfcM_s4P-dfzKrOpcFWWK&index=14>

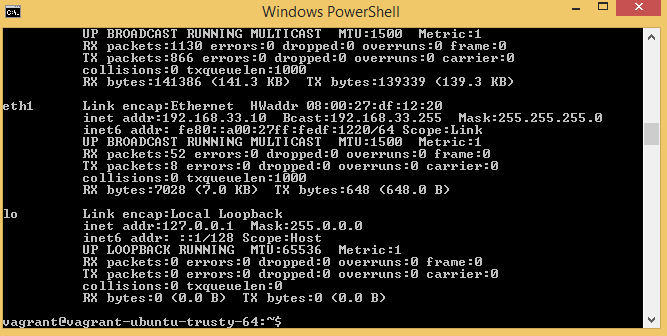
https://www.youtube.com/watch?v=ccya8HWg8ZA&list=PL5nViEmyYI0bmfcM\_s4P-dfzKrOpcFWWK&index=15

https://community.cloudera.com/t5/Support-Questions/Permission-denied-publickey-gssapi-keyex-gssapi-with-mic/td-p/146747



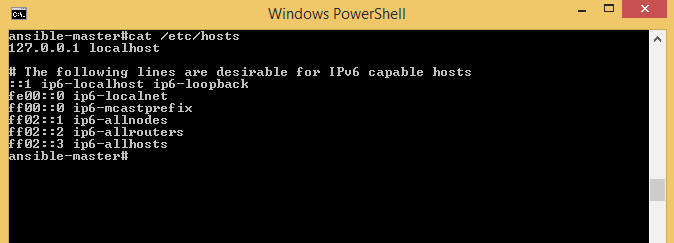






Make this system as:ansible-master#



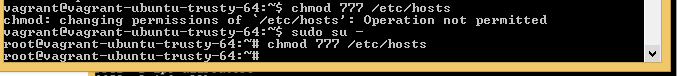


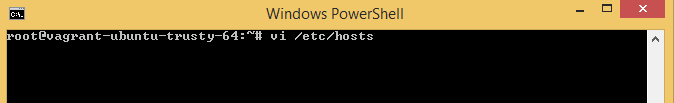
Make entries of 192.168.33.10 (ansible-master#) machine ipaddress and 192.168.33.11 (ansible-slave#) in /etc/hosts file.

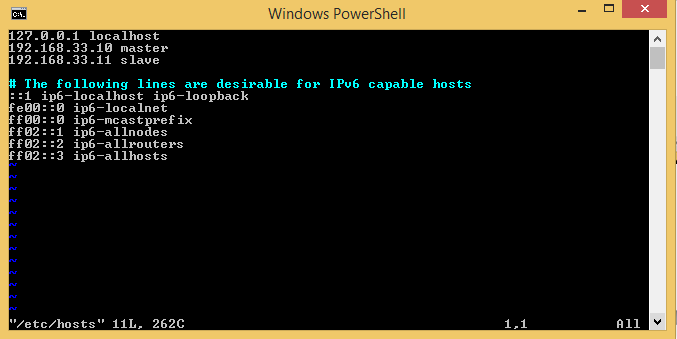
We can also Provide the alias name for both the ipaddress in host file.

Provide the permission to the file for write in the file(hosts),

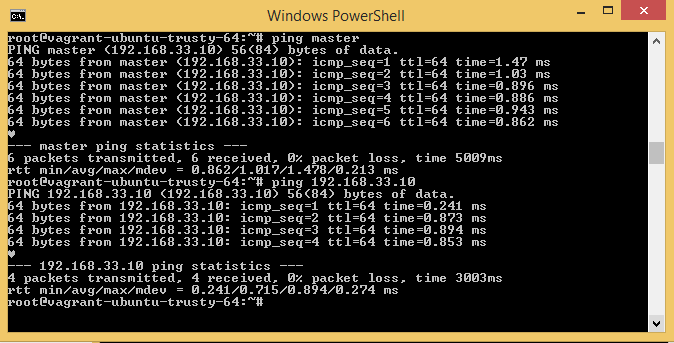
To provide permission We need to switch into root mode.

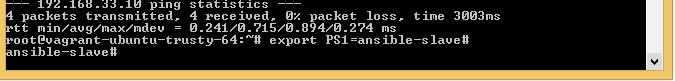




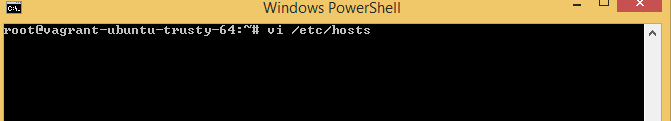


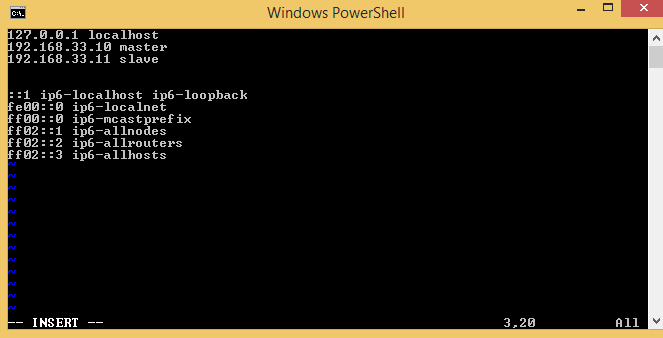
Now we are able to ping master machine from slave machine with alias name: master as well as with ipaddress 192.168.33.10

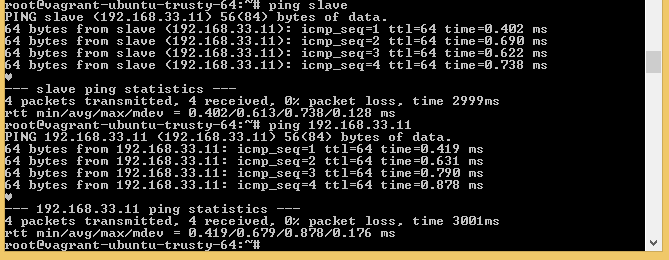




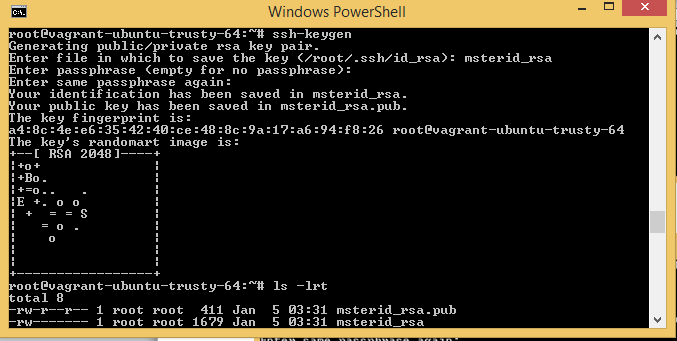
Same steps need to perform on master machine:

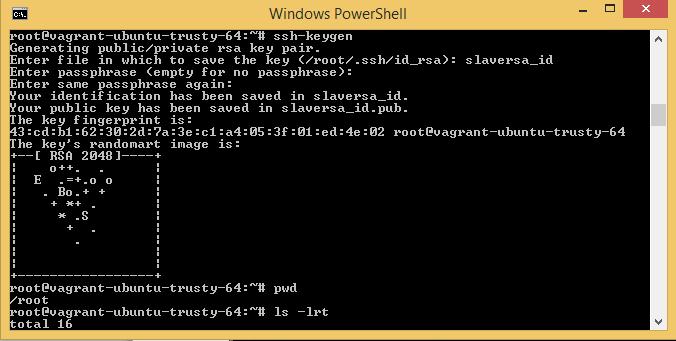


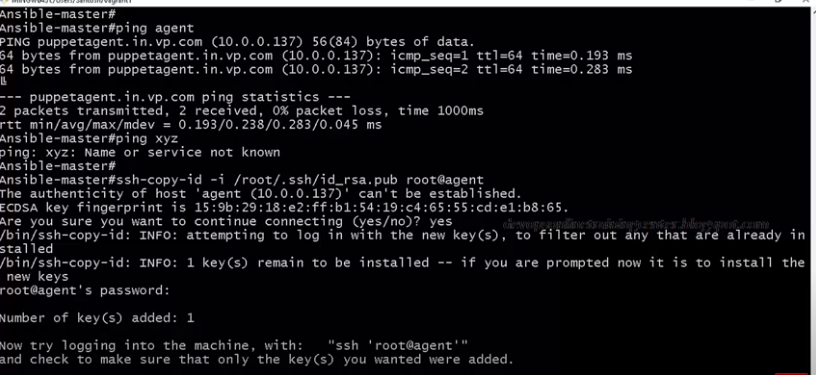


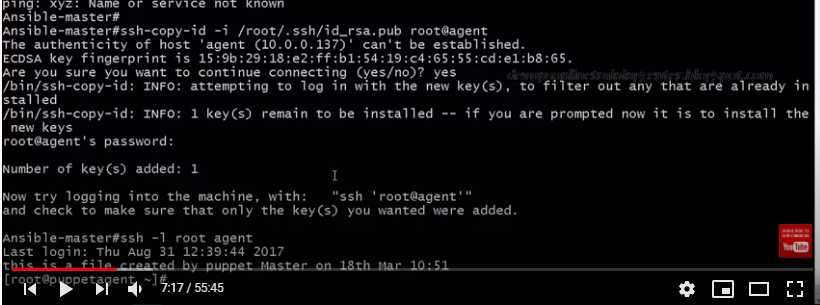


Now generate the SSH-KEYGEN on both the machines:







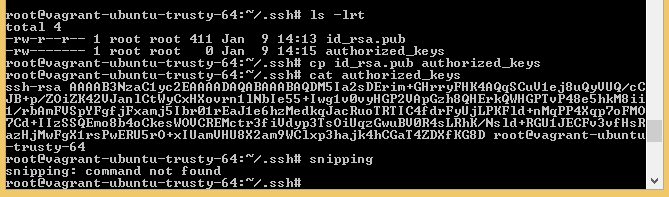


Check the file content authorized\_keys on slave(/root/.ssh), if it is blank, then copy content of the master machine file id\_rsa.pub into authorized\_keys file.

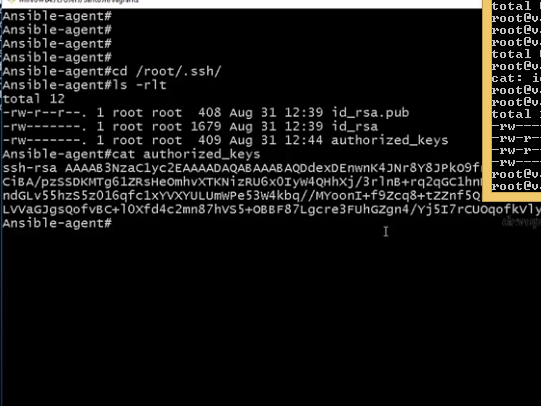
Options:

1. Copy the file id\_rsa.pub file into slave/agent machine from file of master machine(id\_rsa.pub)

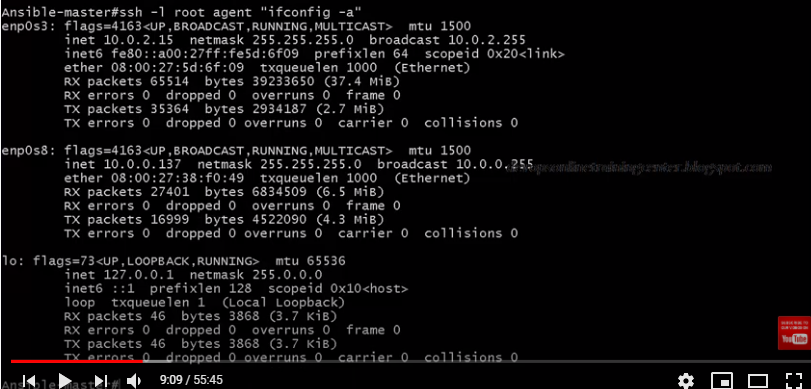
Then copy the contend of id\_rsa.pub into authorized\_keys file on slave/agent machine.



To check the keys for master in Agent/slave machine



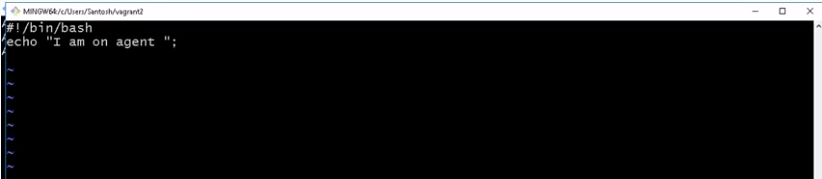
To check hostname or ipaddress of agent/slave machine without asking password.



We can execute slave script(1.sh) on master, so first of all create 1.sh script on slave

And execute it from master.





Change the permission of file on slave/agent machine:

Chmod 777 1.sh



We can login into the master on slave and execute any file from master as well.



Out from Agent/Slave machine by following commands:



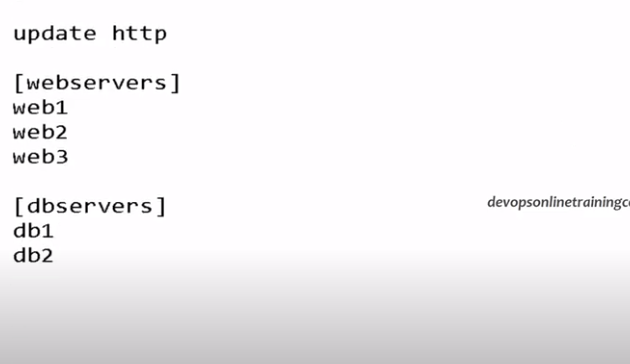


Web server mean: tomcat/apache server installed on 30 webserver

Data vase server mean: sql server/mysql installed on 30 webserver

App servers: meane some application installed on 30 app server

Monitoring server mean: Nagios



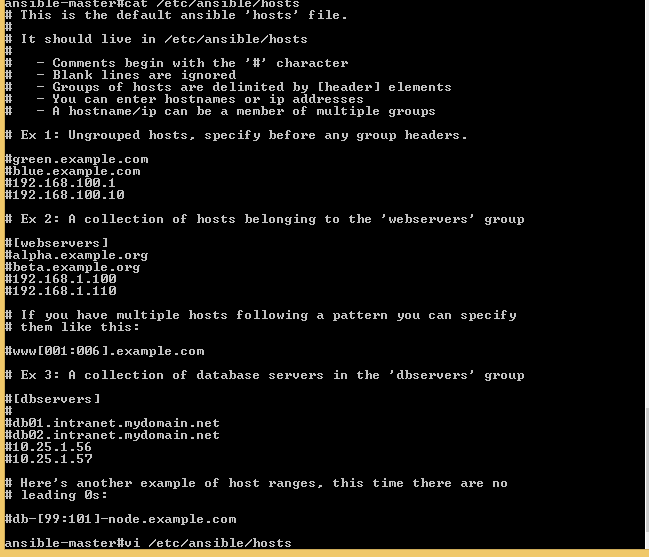
Combining all the servers into group.

Add it into ansible host from master machine.

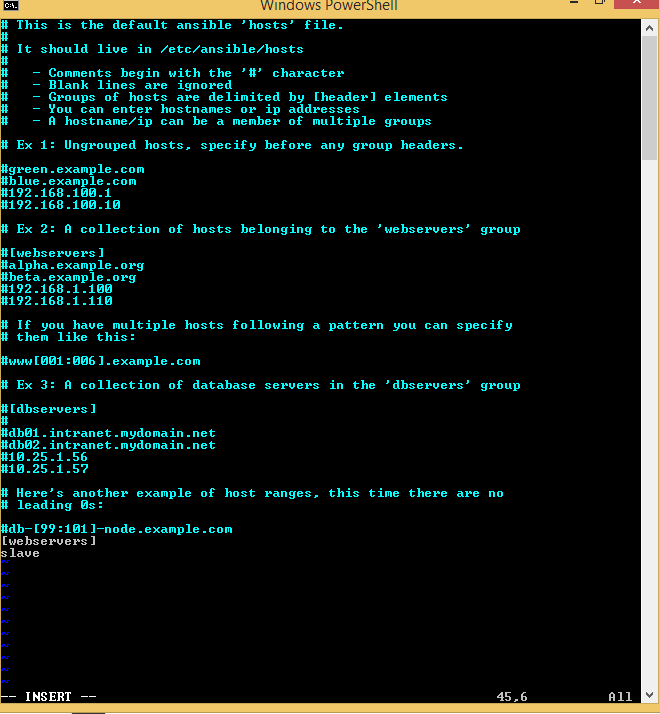
/etc/ansible/hosts

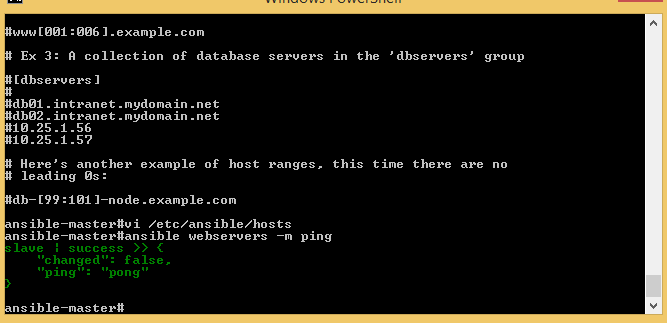
Install ansible on master machine





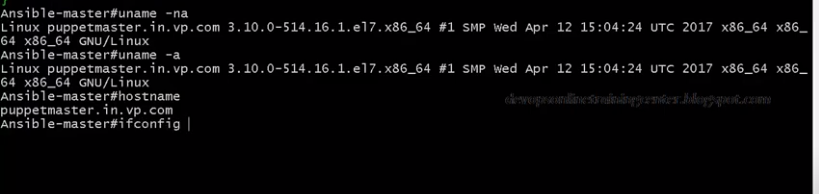
Add the slave machine name in webserver group.





Ansible webservers –m ping: m is here module

So in similar fashion add other slave/agent machine, and ping.It will ping simply all the machine name which are add into inventory file(hosts) of ansible.

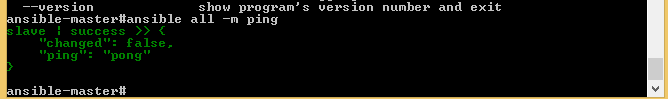


If machine ip is added in DNS then it will ping the machine without adding into hosts file otherwise we need to add it into hosts file.

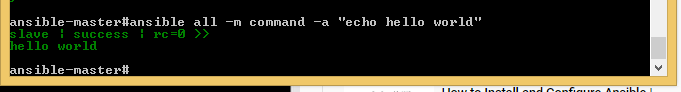
We can also add ip in host file, it will work.

This will ping all the machine.

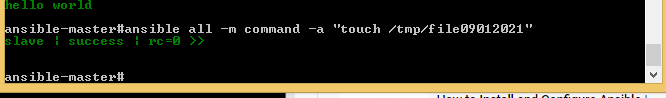




Module command executed from master but actually executed on agent and print the output hello world.



Where is this file created?



This file is created in slave/agent machine.



Created file from master on slave/agent machine via ansible installed on master machine.

Note: INSTALL PAYTHON on AGENT/SLAVE MACHINE , before doing ansible command from master for agent/slave.

After successfully setup 2nd Agent machine.

1. Make a copy of successful vagrant file in another floder say:centos1
2. Do vagrant UP and Vagrant ssh command
3. Make entires in /etc/hosts file in master machine as well as in agent machine

: 192.168.33.10 master

192.168.33.11 slave

192.168.33.12 agent

4) Change the agent root machine password, first of all move to root

Sudo sudo su –

Sudo passwd root

1. Generate ssh-keygen in master machine
2. copy the id\_rsa.pub file of master machine into agent machine

scp /root/.ssh/id\_rsa.pub [root@agent:/root/.ssh/](mailto:root@agent:/root/.ssh/)

5) copy the id\_rsa.pub content into authorized\_keys file, then it allow to login into agent machine from master machine without password.

5)Allow root permit as Yes in sshd\_config file in agent machine name.

6)sudo vi /etc/ssh/sshd\_config

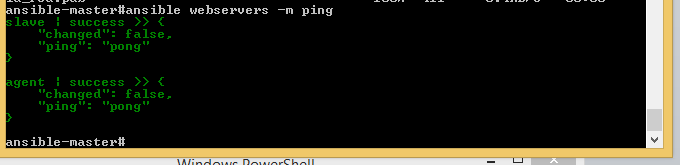
7) restart the service:sudo service ssh restart

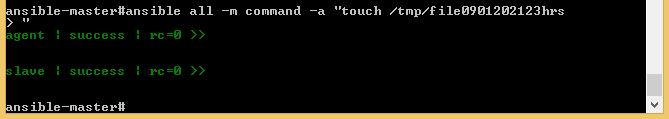
8) Check wthether the python installed or not in both machine agent,slave,master also, by default , it was installed.If not, then install by command:

**Sudo apt install python**

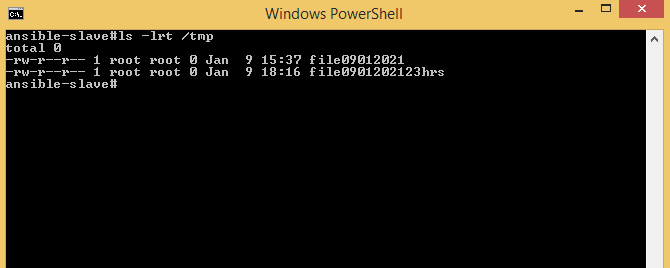
8) Make entries of machine “agent” name under [webservers] group in ansible host file. /etc/ansible/hosts

From master machine, we are able to ping all the machine at a time.

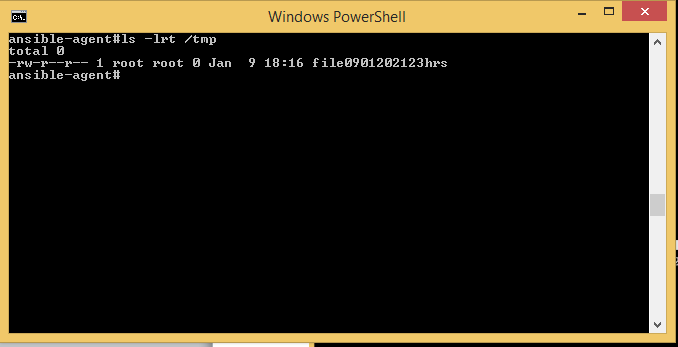


Create file in 2 machine(agent and slave) at one time, 

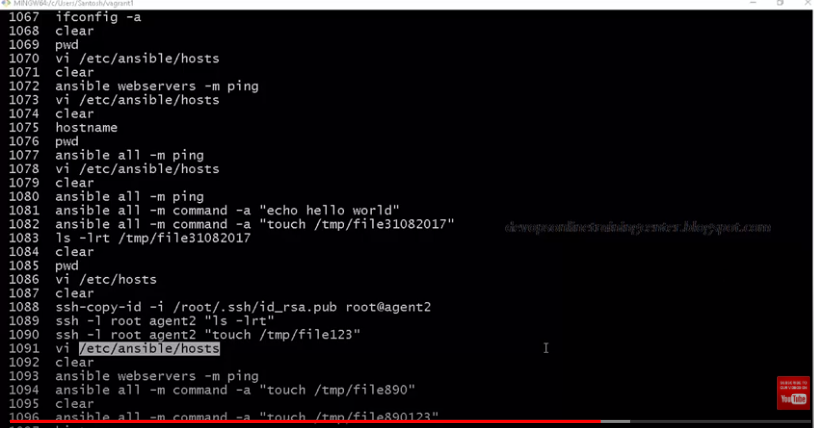
Slave machine where file0901202123hrs created.



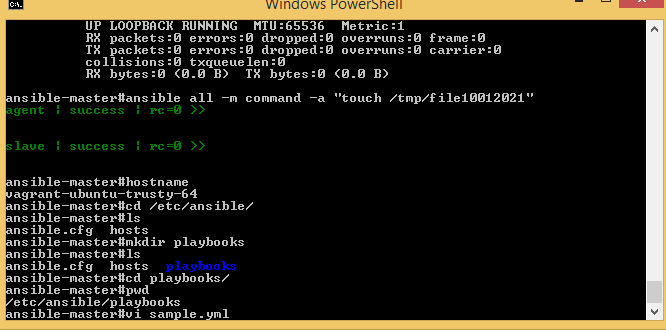
Slave machine where file0901202123hrs created.



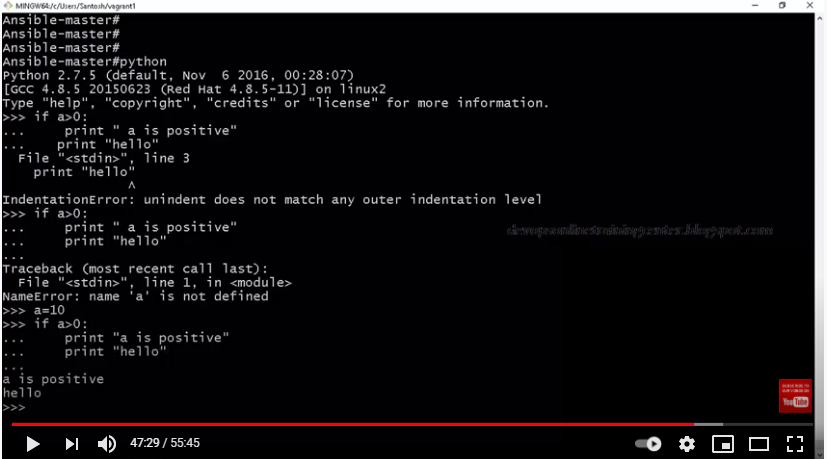
Few Useful Commands:



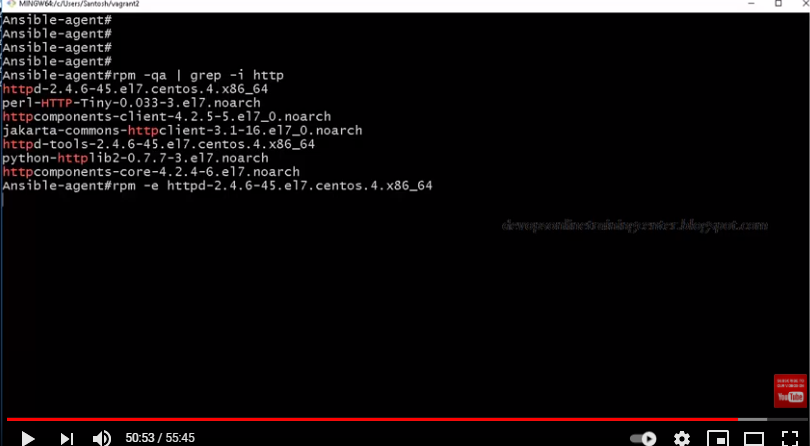
Another Way to execute the commands on all agent/slave machine by Playbooks.



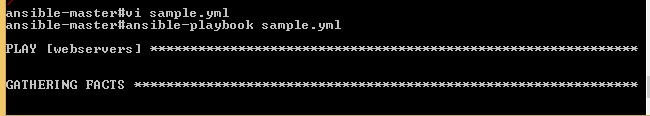
Python & YML file follow indentation while scripting.



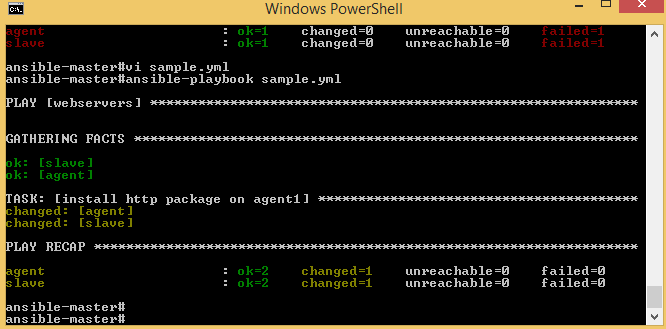
Check on agent machine whether httpd installed or not.If it is installed then remove it.

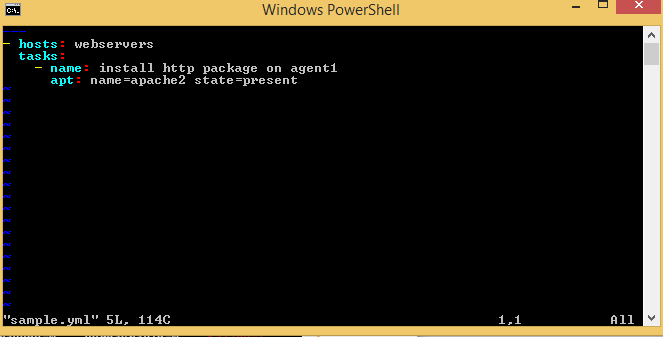


To check whether http install or not.

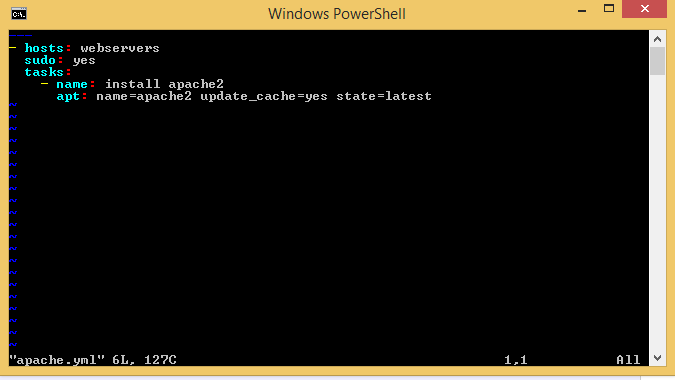


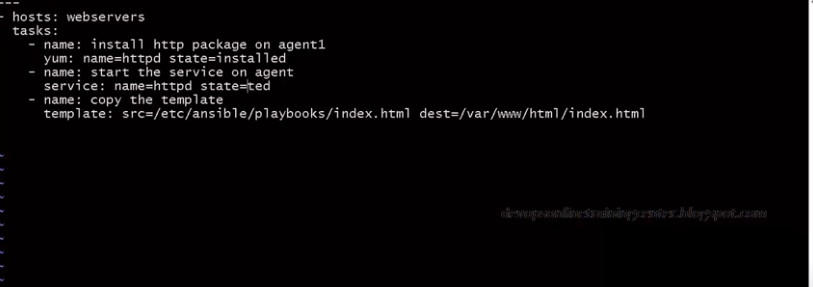
<https://www.middlewareinventory.com/blog/ansible-apt-examples/>

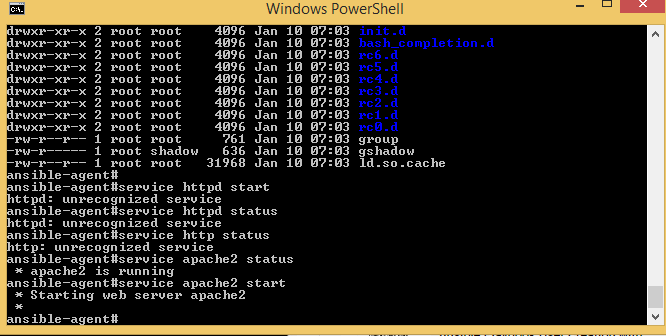


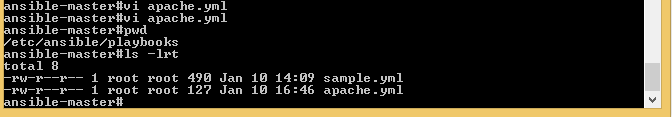


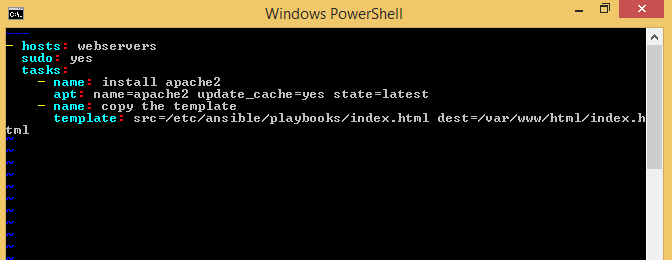
Playbook apache.yml

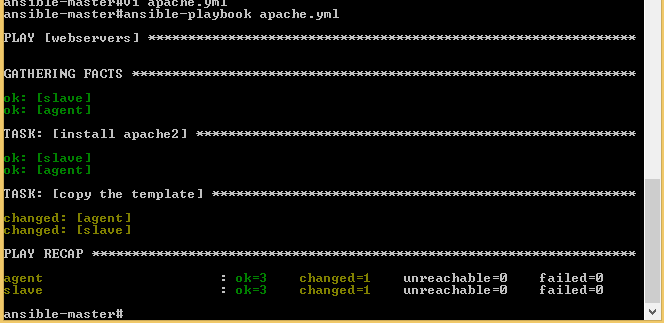




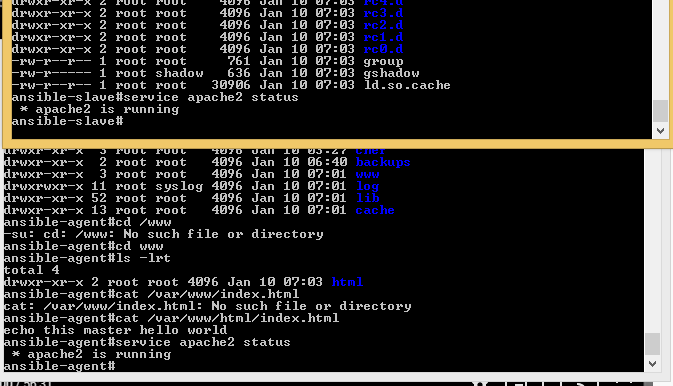






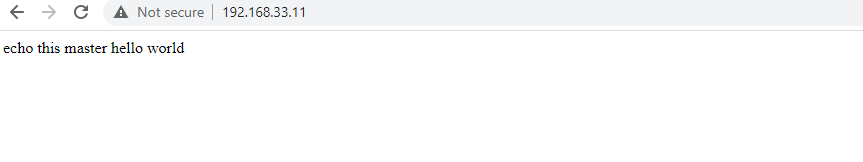


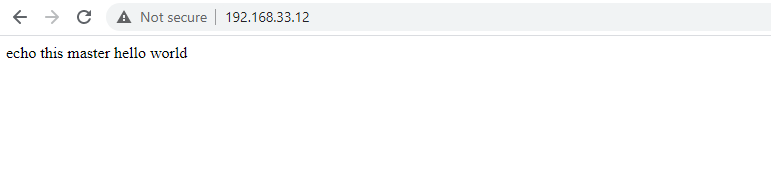
Apache2 server is running on Agent/Slave machine



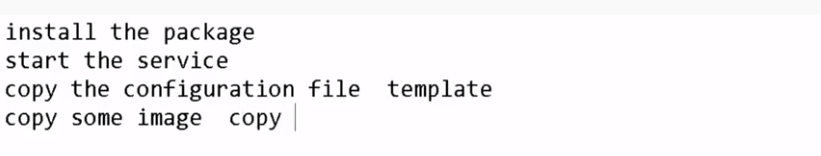
Once apache2 server is running, check on webbrowser with 192.168.33.11 or 192.168.33.12.

It is printing message “echo this master Hello world”



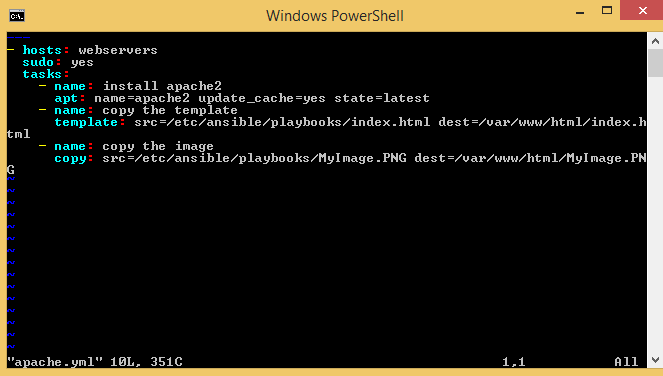


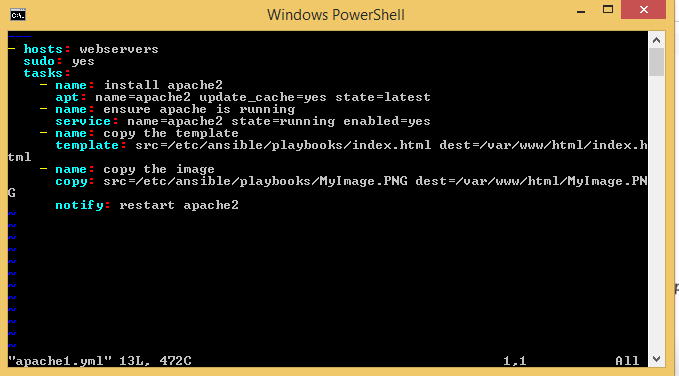
<https://www.scaleway.com/en/docs/how-to-install-apache-on-ansible/>

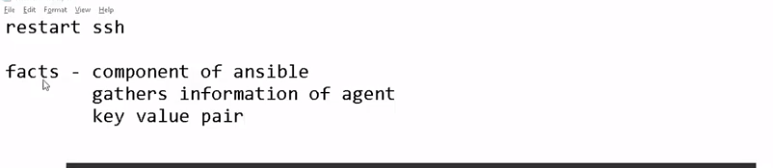


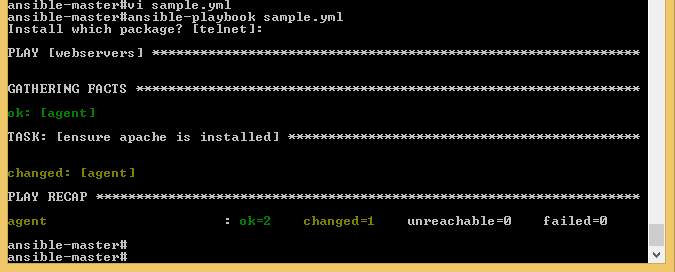
Copy the configuration: Using here template module

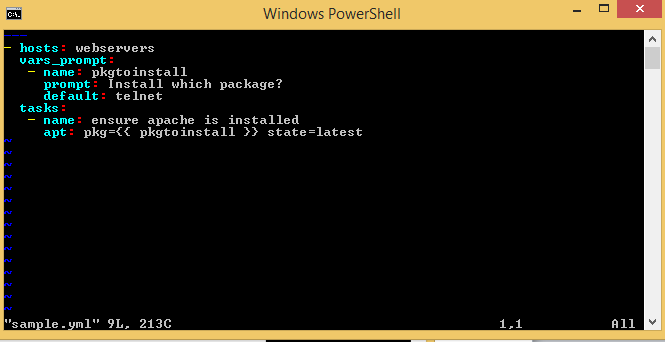
Copy some image: using copy module







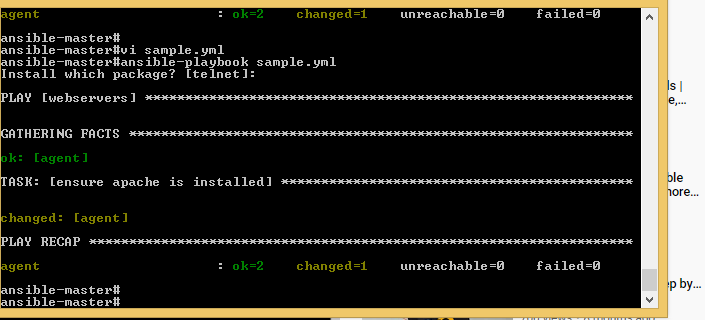




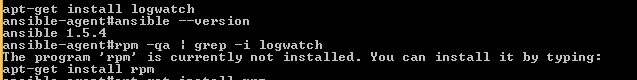
Facts—gathering the agent system information

Install logwatch from master machine to agent machine:

Enter logwatch as parameter when ask for user input. It will install logwatch on agent machine



Installing rpm on agent machine



To get a list of a specific package installed:

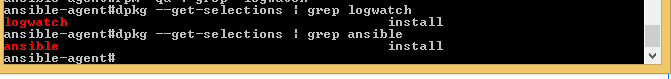
dpkg --get-selections | grep postgres

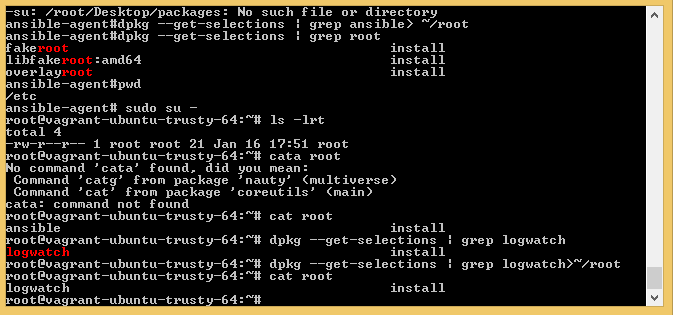
To save that list to a text file called packages on your desktop do this in your terminal:

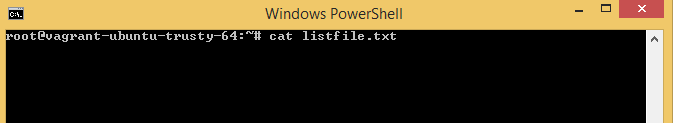
dpkg --get-selections | grep -v deinstall > ~/Desktop/packages

Alternatively, simply use

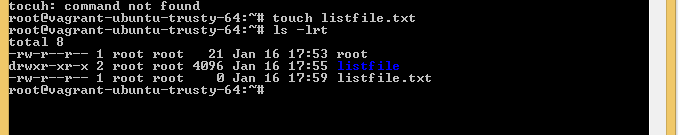
dpkg -l



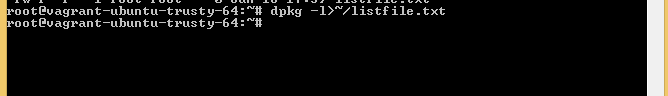


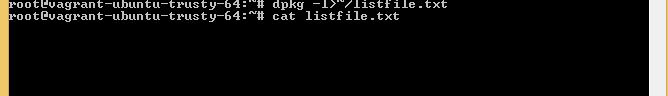
To save the list of packages are installed on agent machine:  


First Create file:



Below command copy all the package name in listfile.txt





Tags : It can used to execute the part of program



<https://linuxhint.com/fix_connection_refused_ubuntu/#:~:text=Sometimes%20while%20connecting%20to%20SSH,of%20the%20IP%20conflict%20issue>.

In order to check which on which port the OpenSSH server is listening to; use the following command in Terminal:

 sudo netstat -ltnp | grep sshd

To Login into the machine:

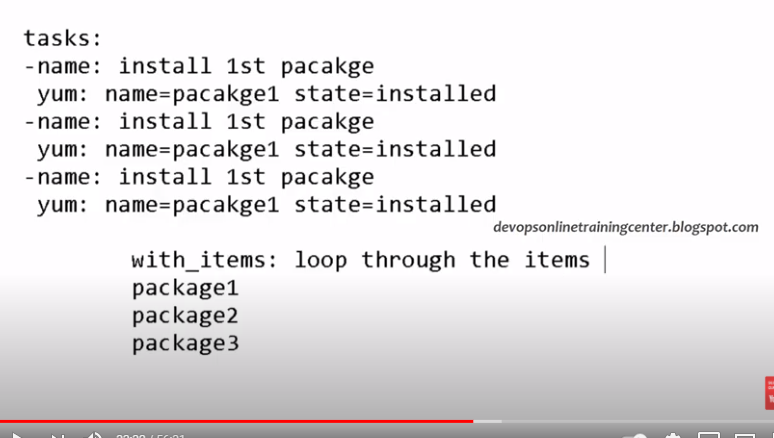
ssh username@machine

ssh vagrant@agent

If you want to login with Port then specify the Port number:

Ssh –p vagrant@agent







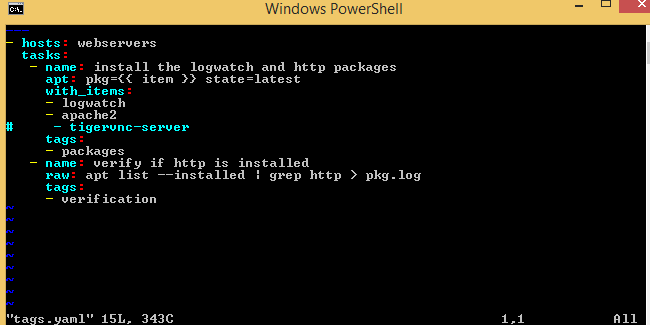
Check whether a specific package is installed in Ubuntu

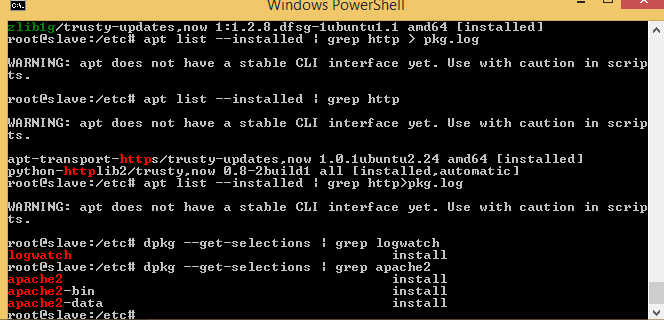
Since the list of installed packages is a huge one, it would be a better idea to use grep and filter the output for a certain package.

apt list --installed | grep program\_name

A better way is to use this command:

apt -qq list program\_name --installed



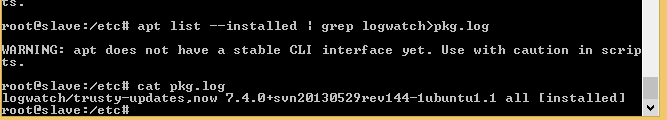


After searching it will store the name in pkg.log file

1st Way:



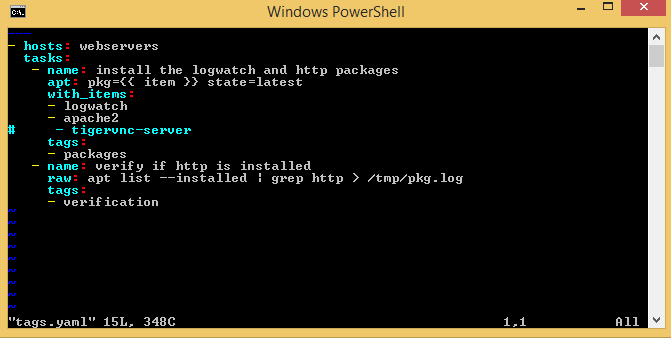
2nd Way:

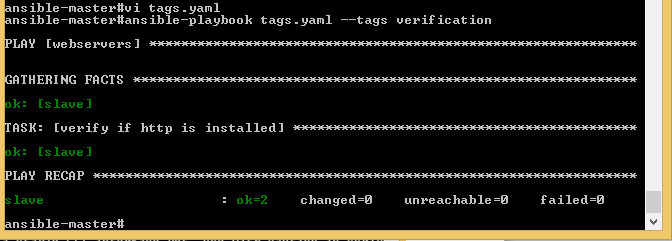


If you check on tmp floder : there is nothing in the floder



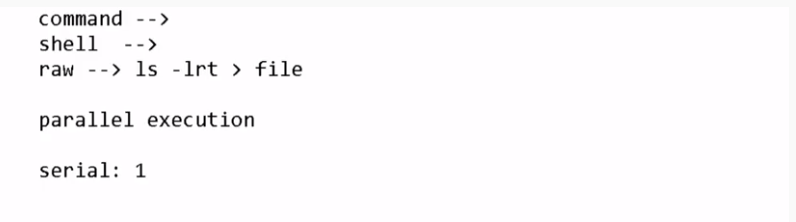
When we execute the playbook, it will create file in agent/slave temporary floder.







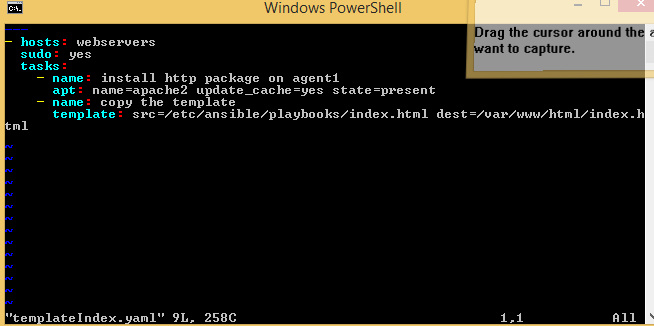
In playbook, if you mentioned serial , then playbook will install software/package sequential otherwise it is parallel execution.By default, it is parallel.

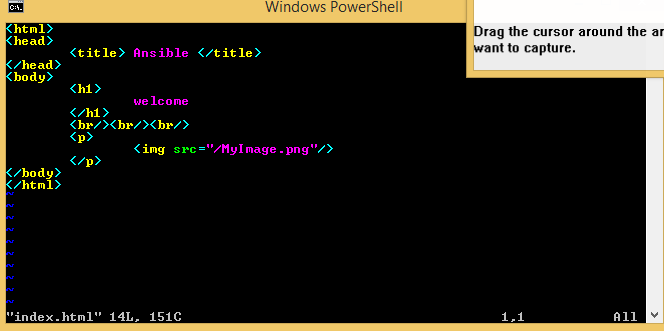


Display the image on browser of agent:

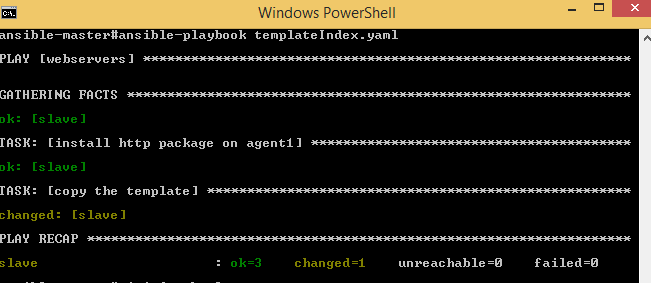
Copy the image file also in the slave machine@/var/www/html



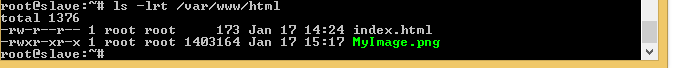


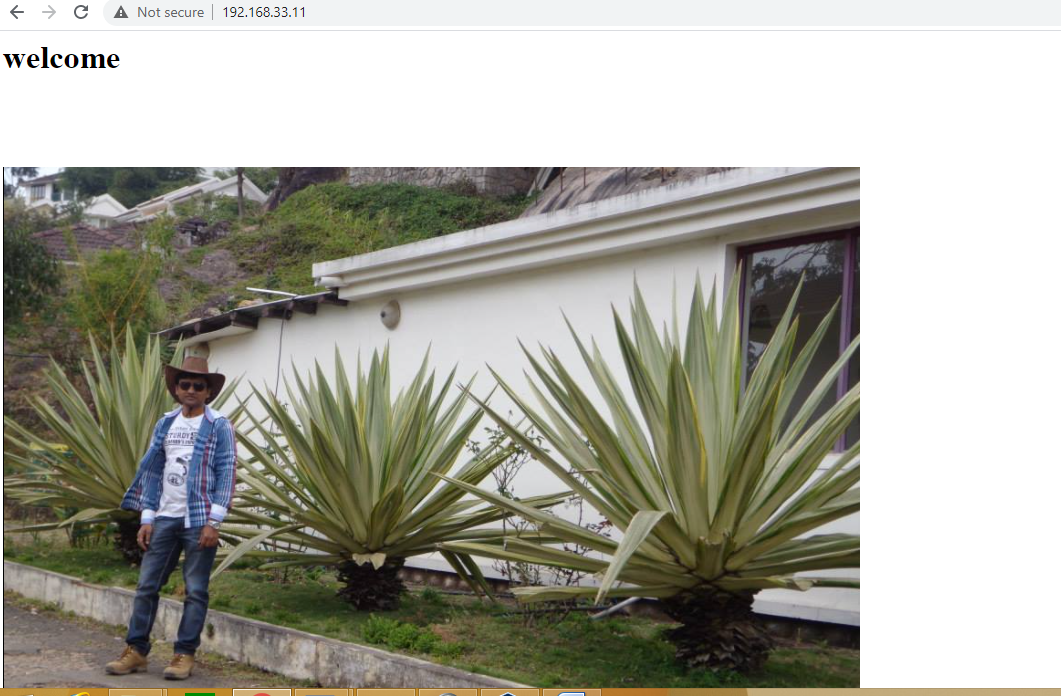


Now execute the playbook file:



Now open the browser with slave/agent ip address:eg:192.168.33.11 or 192.168.33.12







This is serial execution:



