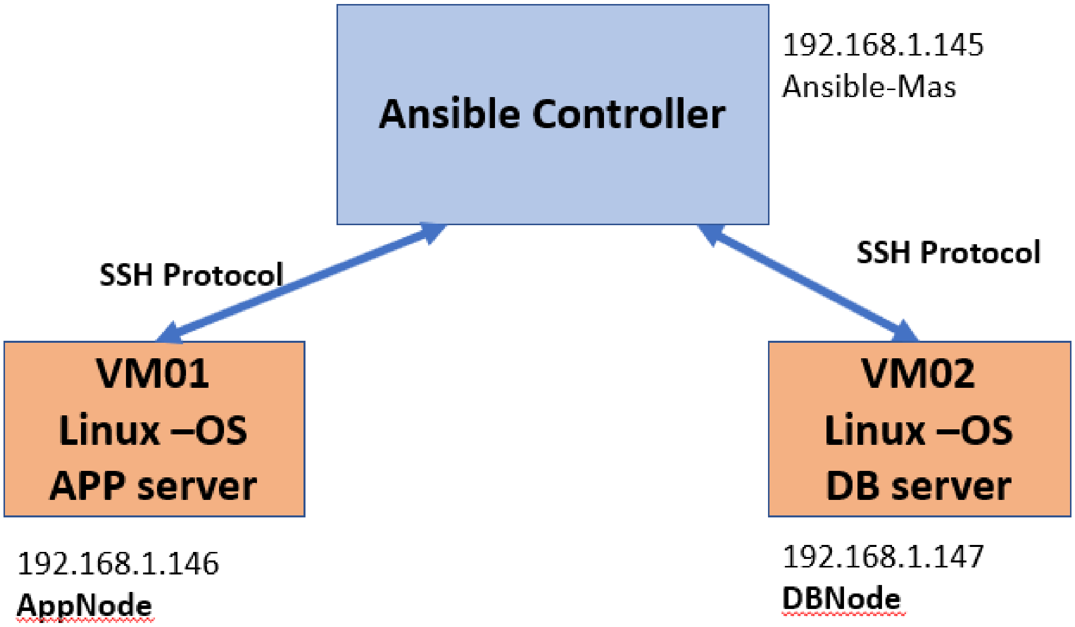
Pre Preparation for Ansible Setup.

3 Servers

Ansible LAB Setup Details

Server Name	Ip Address	Server Role
Ansible-control	192.168.1.145	Ansible Control Server
Appserver	192.168.0.146	App Server
DBServer	192.168.0.147	DBserver

Diagram



Steps:

1. **Installing Ansible Software on the Ansible Controller**
2. **Setup hostname’s all the nodes with “/etc/hosts”**
3. **Setup the SSH Keys for Ansible to login to Node (password less)**

Step1: Installing Ansible Software on the Ansible Controller

On RHEL and CentOS:

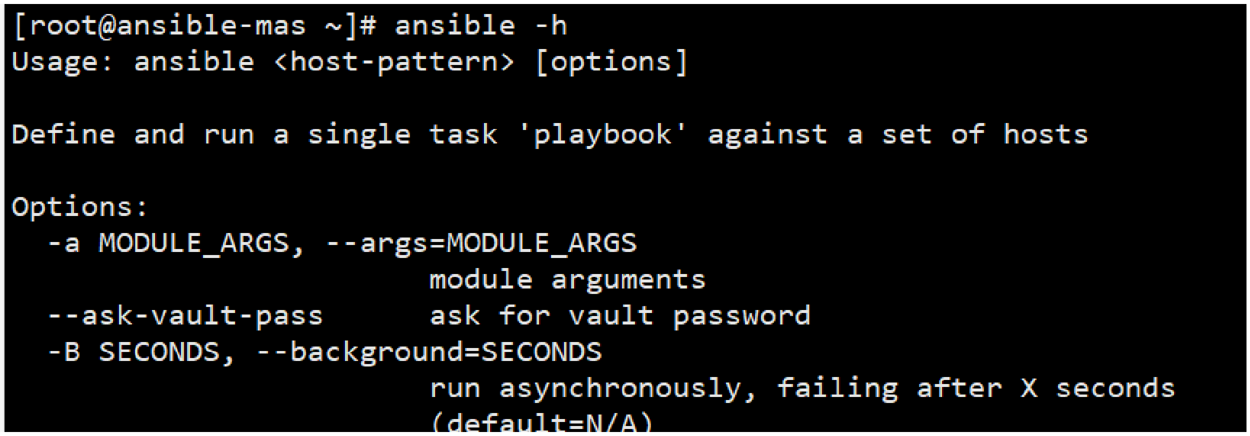
```
$ sudo yum update -y

$ sudo yum install ansible -y
```

This would successfully install the ansible.

Also recommended to do is an update after the installation of ansible

```
$ sudo yum update -y
```

To Check if Ansible is installed



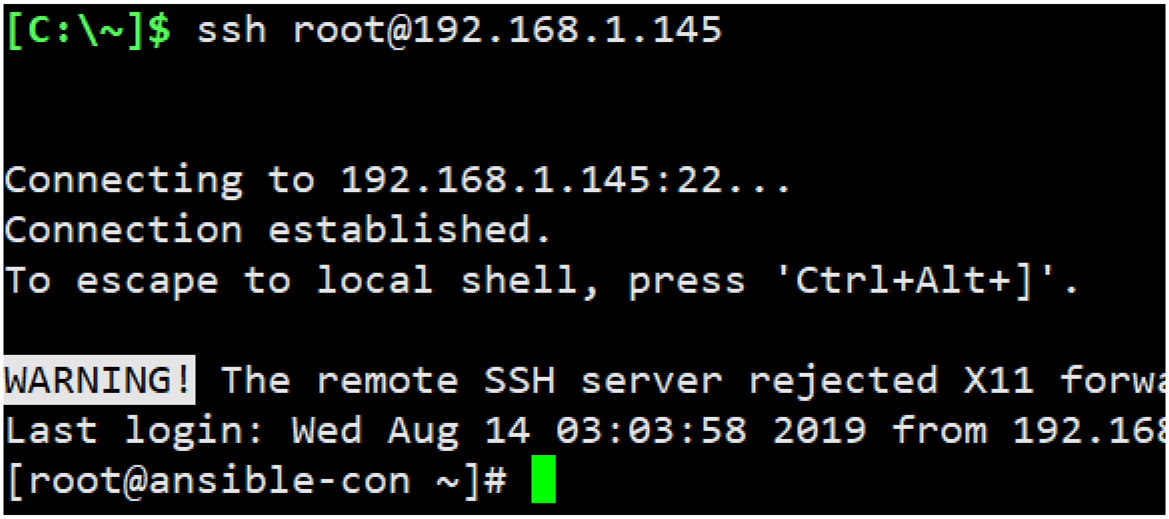
```
[root@ansible-mas ~]# ansible -h
Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:
  -a MODULE_ARGS, --args=MODULE_ARGS
                        module arguments
  --ask-vault-pass      ask for vault password
  -B SECONDS, --background=SECONDS
                        run asynchronously, failing after X seconds
                        (default=N/A)
```

Step2: Configure hostname on all the machines.

Ansible Controller.

Ssh to the Ansible in our case, **IP: 192.168.1.145**

```
$ hostnamectl set-hostname ansible-con
```

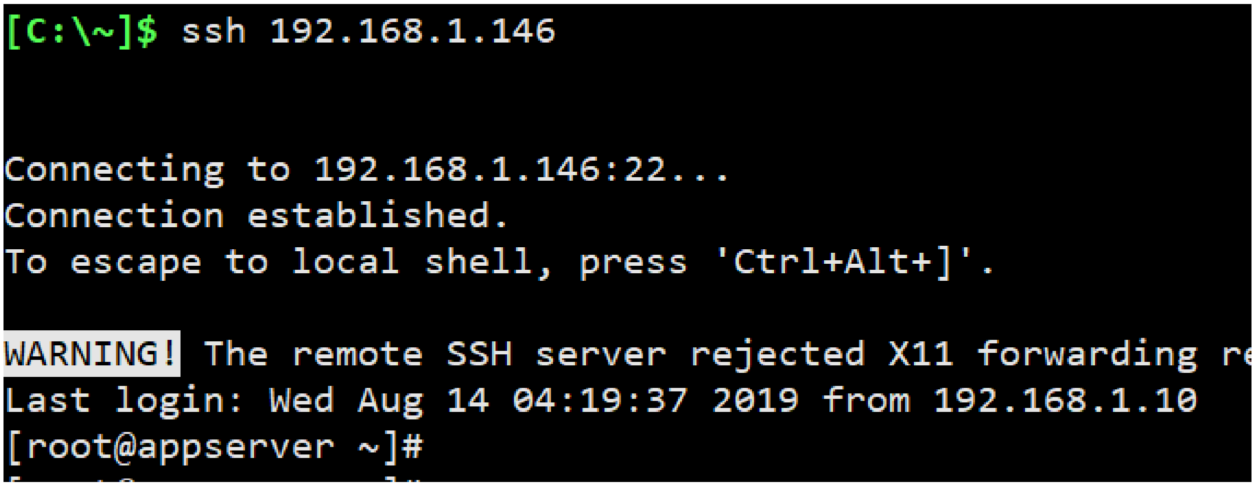


```
[C:\~]$ ssh root@192.168.1.145


Connecting to 192.168.1.145:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwa
Last login: Wed Aug 14 03:03:58 2019 from 192.168
[root@ansible-con ~]#
```

App Server.

Ssh to the Ansible in our case**, IP: 192.168.1.146**

```
$ hostnamectl set-hostname appserver
```

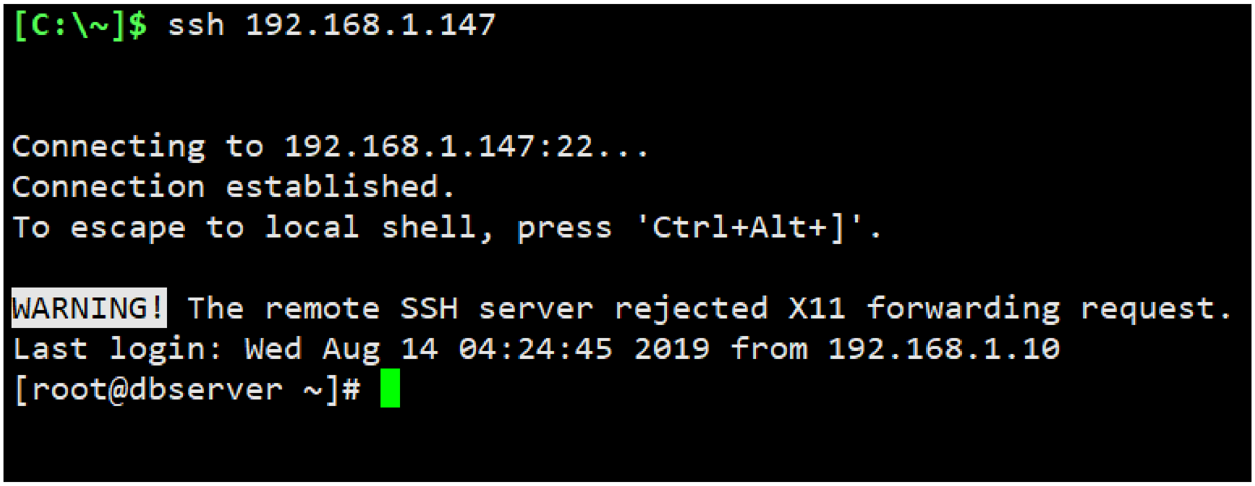


```
[C:\~]$ ssh 192.168.1.146


Connecting to 192.168.1.146:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding re
Last login: Wed Aug 14 04:19:37 2019 from 192.168.1.10
[root@appserver ~]#
```

DB Server.

Ssh to the Ansible in our case**, IP: 192.168.1.147**

```
$ hostnamectl set-hostname dbserver
```



```
[C:\~]$ ssh 192.168.1.147


Connecting to 192.168.1.147:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Wed Aug 14 04:24:45 2019 from 192.168.1.10
[root@dbserver ~]# 
```

Configure the "/etc/hosts" file on all the above 3 server's with below config

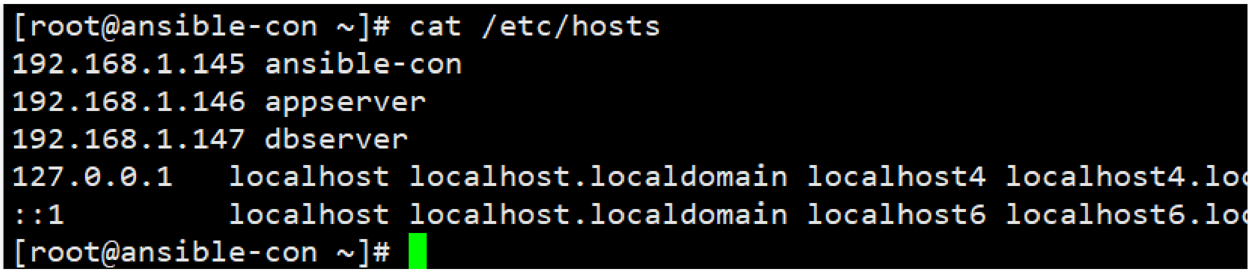
All the server need to have the below hosts file content.

Note: -- It is best practice to use the hostname's of the client in the Ansible setup

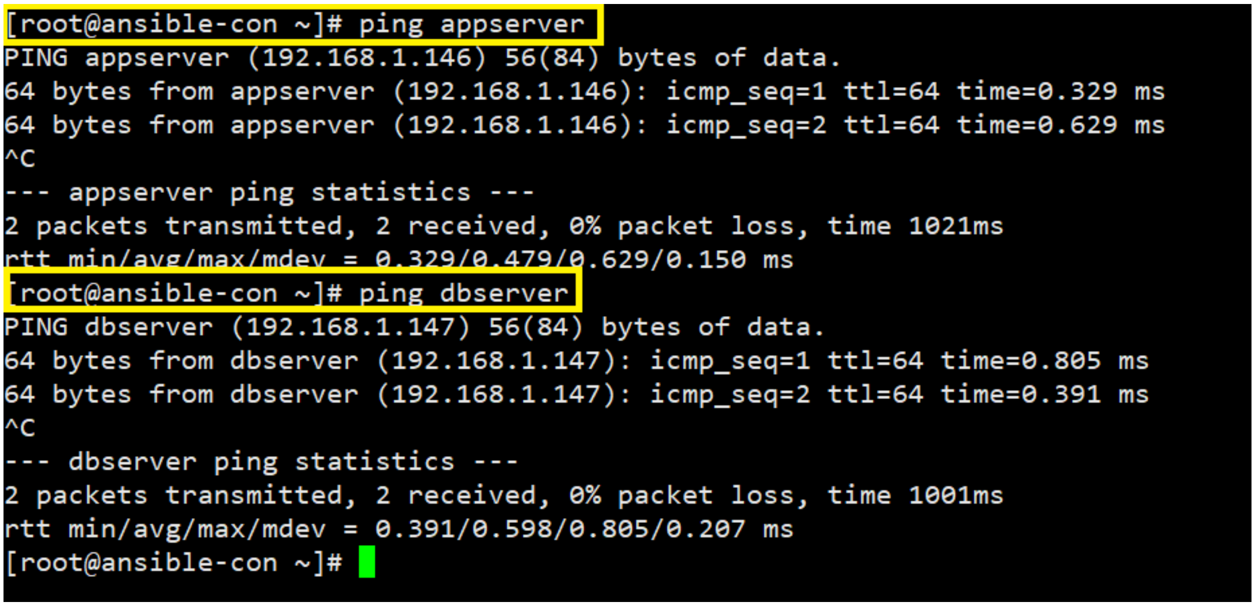
But its ***NOT*** *compulsory.*

```
192.168.1.145 ansible-con
192.168.1.146 appserver
192.168.1.147 dbserver
```

On Ansible Controller



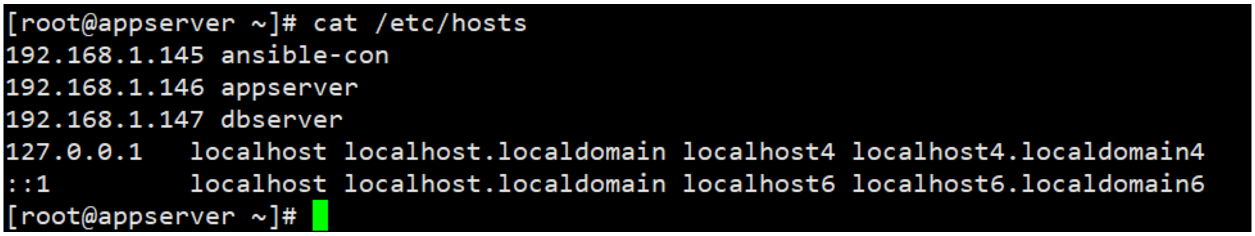
```
[root@ansible-con ~]# cat /etc/hosts
192.168.1.145 ansible-con
192.168.1.146 appserver
192.168.1.147 dbserver
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.loc
::1         localhost localhost.localdomain localhost6 localhost6.loc
[root@ansible-con ~]#
```



```
[root@ansible-con ~]# ping appserver
PING appserver (192.168.1.146) 56(84) bytes of data.
64 bytes from appserver (192.168.1.146): icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from appserver (192.168.1.146): icmp_seq=2 ttl=64 time=0.629 ms
^C
--- appserver ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.329/0.479/0.629/0.150 ms
[root@ansible-con ~]# ping dbserver
PING dbserver (192.168.1.147) 56(84) bytes of data.
64 bytes from dbserver (192.168.1.147): icmp_seq=1 ttl=64 time=0.805 ms
64 bytes from dbserver (192.168.1.147): icmp_seq=2 ttl=64 time=0.391 ms
^C
--- dbserver ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.391/0.598/0.805/0.207 ms
[root@ansible-con ~]#
```

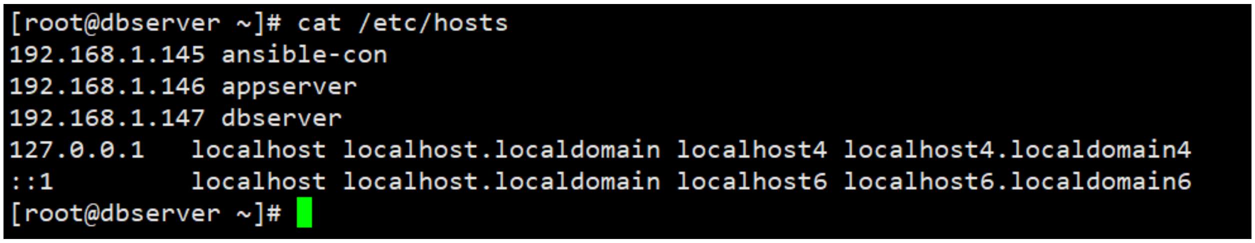
This confirms that the hostname's are working fine.

On app server



```
[root@appserver ~]# cat /etc/hosts
192.168.1.145 ansible-con
192.168.1.146 appserver
192.168.1.147 dbserver
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@appserver ~]#
```

On DB server



```
[root@dbserver ~]# cat /etc/hosts
192.168.1.145 ansible-con
192.168.1.146 appserver
192.168.1.147 dbserver
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@dbserver ~]#
```

COOL….. We are ready for the next step to setup the SSH keys between all the 3 nodes.

Step3: Configure all the 3 VM's with same SSH key.

Method 1:

1. Run **“ssh-keygen”** on the Ansible controller.
2. Copy the content of **“.ssh/id_rsa.pub”** in the **Ansible controller** to the file “**authorized_key**” on the VM “**appserver**” inside the folder “**.ssh**” of **same user** profile.
3. Set the permission of the file “**authorized_key**” to “600”
   $ chmod 600 **authorized_key**
4. Similarly to step “2 and 3” for the VM “**dbserver**” as well.

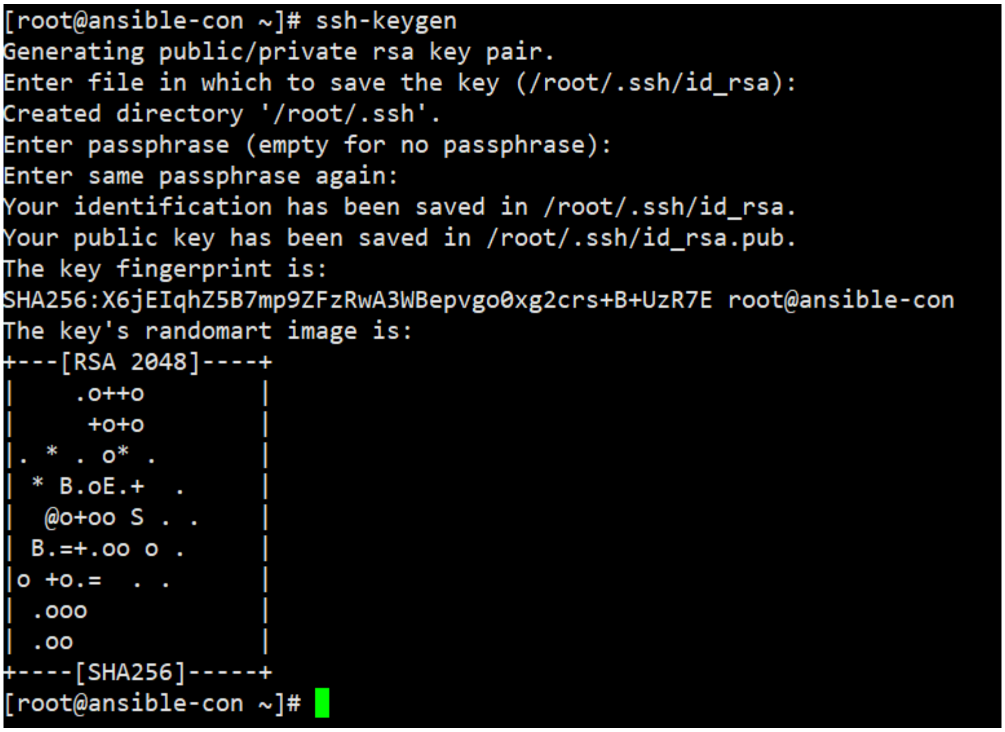
But this is a little tedious process……

Let’s take a short cut and do the same in a simple manner.

Method 2.

a. On the **Ansible controller** , login as **root**

b. run “**ssh-keygen**”



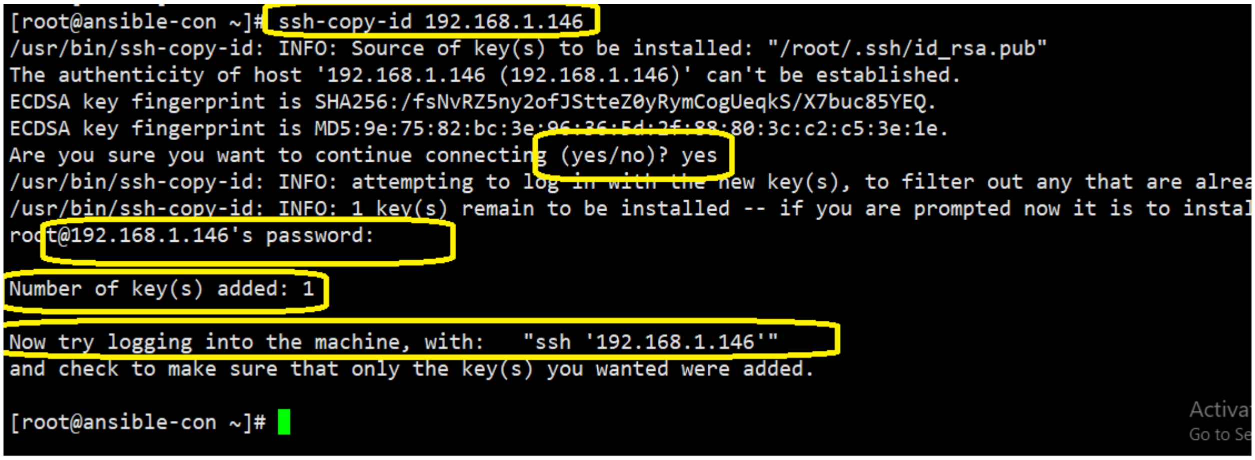
```
[root@ansible-con ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:X6jEIqhZ5B7mp9ZFzRwA3WBepvgo0xg2crs+B+UzR7E root@ansible-con
The key's randomart image is:
+---[RSA 2048]----+
|    .o++o        |
|     +o+o        |
|. * . o* .       |
| * B.oE.+  .     |
|  @o+oo S . .    |
| B.=+.oo o .     |
|o +o.=  . .      |
| .ooo            |
| .oo             |
+----[SHA256]-----+
[root@ansible-con ~]#
```

Now, let’s do the magic, with the below command.

Run the below command on the ansible controller.

```
$ ssh-copy-id 192.168.1.146
```

This command would copy the “public key” from the ansible controller to “appserver”(192.168.1.146)

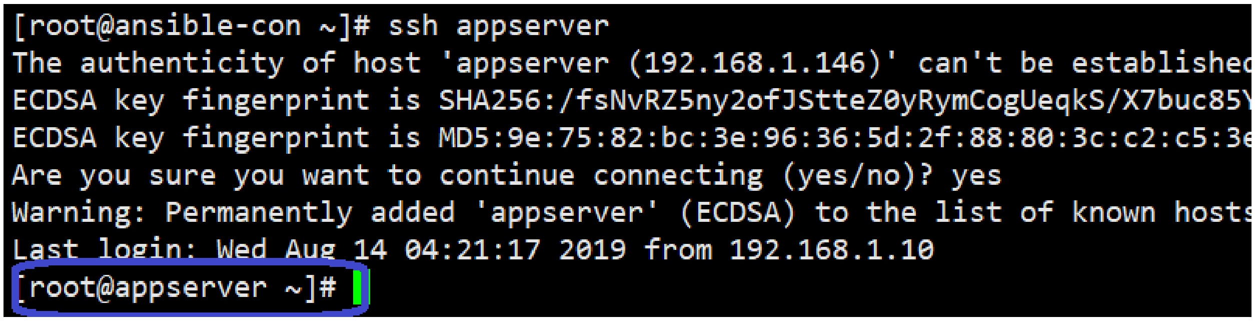


```
[root@ansible-con ~]# ssh-copy-id 192.168.1.146
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.146 (192.168.1.146)' can't be established.
ECDSA key fingerprint is SHA256:/fsNvRZ5ny2ofJStteZ0yRymCogUeqkS/X7buc85YEQ.
ECDSA key fingerprint is MD5:9e:75:82:bc:3e:96:36:5d:2f:88:80:3c:c2:c5:3e:1e.
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 alrea
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to instal
root@192.168.1.146's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.1.146'"
and check to make sure that only the key(s) you wanted were added.

[root@ansible-con ~]#
```

To check.

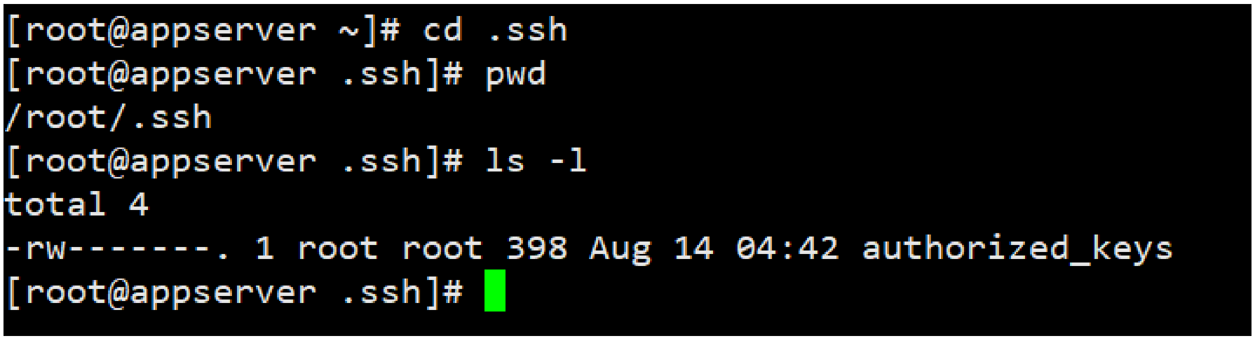
Do an ssh to appserver from ansible controller.

Note: -- IT SHOULD **NOT** ASK FOR PASSWORD



```
[root@ansible-con ~]# ssh appserver
The authenticity of host 'appserver (192.168.1.146)' can't be establishe
ECDSA key fingerprint is SHA256:/fsNvRZ5ny2ofJStteZ0yRymCogUeqkS/X7buc85Y
ECDSA key fingerprint is MD5:9e:75:82:bc:3e:96:36:5d:2f:88:80:3c:c2:c5:3e
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'appserver' (ECDSA) to the list of known hosts
Last login: Wed Aug 14 04:21:17 2019 from 192.168.1.10
[root@appserver ~]#
```

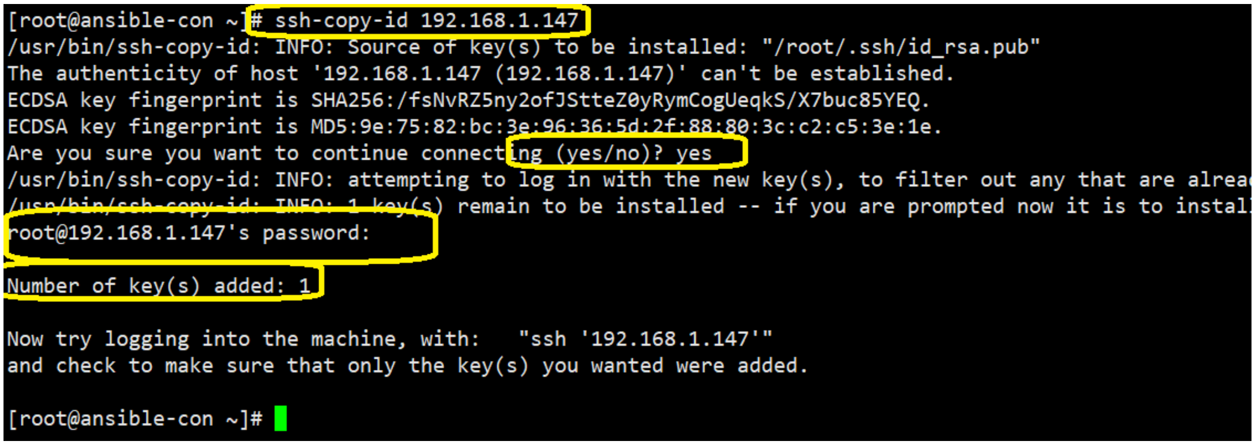
On the appserver.



```
[root@appserver ~]# cd .ssh
[root@appserver .ssh]# pwd
/root/.ssh
[root@appserver .ssh]# ls -l
total 4
-rw-------. 1 root root 398 Aug 14 04:42 authorized_keys
[root@appserver .ssh]#
```

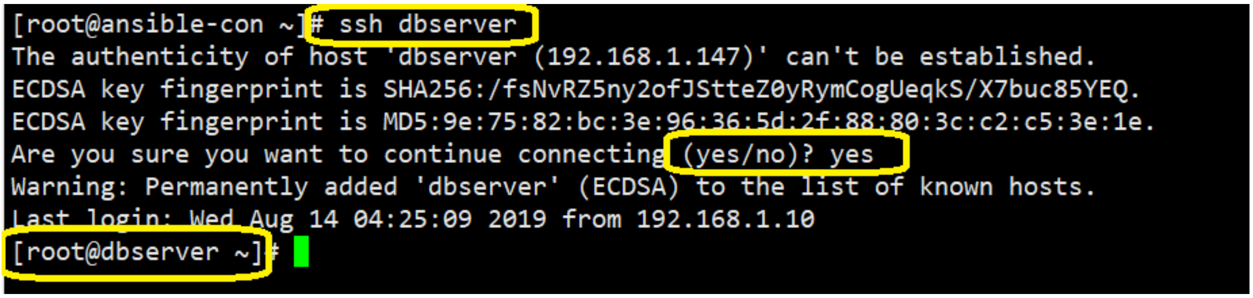
Automatically the "**authorized_key**" **file is created.**

Now, similarly let's do this on the "dbserver" (192.168.1.147)



```
[root@ansible-con ~]# ssh-copy-id 192.168.1.147
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.147 (192.168.1.147)' can't be established.
ECDSA key fingerprint is SHA256:/fsNvRZ5ny2ofJStteZ0yRymCogUeqkS/X7buc85YEQ.
ECDSA key fingerprint is MD5:9e:75:82:bc:3e:96:36:5d:2f:88:80:3c:c2:c5:3e:1e.
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 alrea
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to instal
root@192.168.1.147's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.1.147'"
and check to make sure that only the key(s) you wanted were added.

[root@ansible-con ~]#
```

Checking…. On the Ansible Controller.



```
[root@ansible-con ~]# ssh dbserver
The authenticity of host 'dbserver (192.168.1.147)' can't be established.
ECDSA key fingerprint is SHA256:/fsNvRZ5ny2ofJStteZ0yRymCogUeqkS/X7buc85YEQ.
ECDSA key fingerprint is MD5:9e:75:82:bc:3e:96:36:5d:2f:88:80:3c:c2:c5:3e:1e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'dbserver' (ECDSA) to the list of known hosts.
Last login: Wed Aug 14 04:25:09 2019 from 192.168.1.10
[root@dbserver ~]#
```

Successfully able to ssh without the password prompt.

Perfect… Our LAB is READY to ROCK and ROLL for the ANSIBLE commands……

LET'S START ANSIBLE → THE MOST USED CONFIGURATION MANAGEMENT TOOL