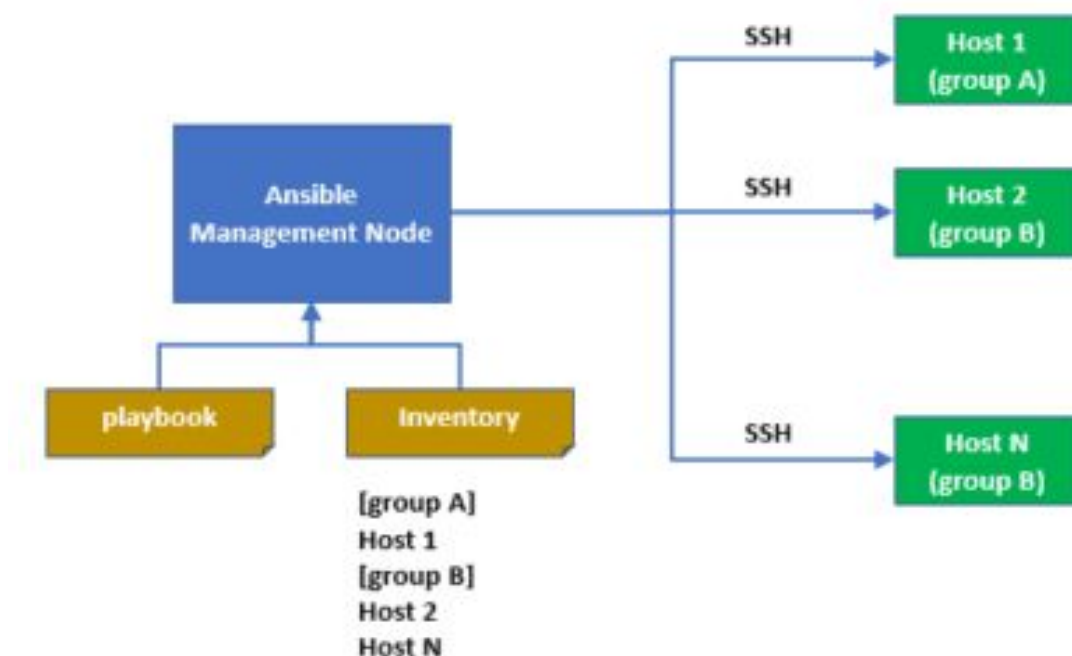


Requirements:

- ❖ Create an ansible playbook to install apache on a CentOS 7 server and make it listen on port 8080 and present the default web page there
- ❖ Configure the firewall on the same linux machine to allow TCP connections to port 8080 using ansible
- ❖ Disable the SELinux on machine using the ansible playbook
- ❖ Install MySQL server 5.7 on the machine using ansible playbook
- ❖ You will need to provide full instructions for someone to take your playbook and execute it successfully. Assume that the person has no knowledge of cloud or ansible.

Ansible architecture



- **Ansible** is an open-source platform for automating configuration management, service orchestration, cloud provisioning and application deployment.

- Ansible is **agentless**, using SSH to push changes from a single source to multiple remote resources.
- **Playbooks** are the files where the Ansible code is written in yml format. It has a list of tasks that need to be executed on node machine.
- **Inventory** file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate. The default location for the inventory file is /etc/ansible/hosts.

Tested Environment (Google Cloud)

IP Address	OS	Purpose
35.223.96.94	CentOS 7	Controlling Machine (Ansible management node)
34.66.245.56	CentOS 7	Client Node (managed node)

Note: If you already installed ansible server, you can skip step 1 and start with step 2.

Step 1: Setup Management Machine

First, login into the management node through SSH command

```
ssh -i server-private-key.pem ansible@35.223.96.94
```

Here `server-private-key.pem` is the private key of the machine.

`ansible` is the user of the machine, user can be anything according to the Linux machine.

`35.223.96.94` is the IP address of the management node.

Once you enter into the machine, execute this command to avoid any permission issue further.

```
sudo su
```

Check whether python is installed or not, it should at least have Python 2 (version 2.6 or later) or Python 3 (version 3.5 or later).

```
python --version
```

```
[root@ansible-server-001 ansible]# python --version  
Python 2.7.5
```

Many of the Linux flavours have python installed in it if it is not installed, Install it with the following below command.

```
yum install python -y
```

Now, let's install Ansible.

```
yum install ansible -y
```

Once Ansible is installed, verify the version of Ansible by executing below command.

```
ansible --version
```

```
[root@ansible-server-001 ansible]# ansible --version  
ansible 2.9.6  
config file = /etc/ansible/ansible.cfg  
configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']  
ansible python module location = /usr/lib/python2.7/site-packages/ansible  
executable location = /bin/ansible
```

Step 2: Setup Managed Nodes

Login to the client node with the following command.

```
ssh -i client-private-key.pem ansible@34.66.245.56
```

Here `client-private-key.pem` is the private key of the machine.

`ansible` is the user of the machine, user can be anything according to the Linux machine.

`34.66.245.56` is the IP address of the management node.

Once you enter into the client machine, execute the below command to avoid any permission issue further.

```
sudo su
```

Client machines should at least have Python 2 (version 2.6 or later) or Python 3 (version 3.5 or later).

```
python --version
```

```
[root@ansible-client-001 ansible]# python --version  
Python 2.7.5
```

Step 3: SSH Authentication

As said earlier, Ansible uses OpenSSH for remote communication. So let's create password less passwordless authentication to execute commands on managed nodes.

Login into the management node through the above command.

Generate public and private key pair using the following `ssh-keygen` command.

```
ssh-keygen
```

```
[root@ansible-server-001 .ssh]# ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
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:riXXxLHC8fpLIx6OMUL8PTMjHm22cONH0+eBjXUxIek root@ansible-server-001  
The key's randomart image is:  
+----[RSA 2048]-----+  
|         . . . . |  
|         ..o  |  
|          . . . o|  
|   .   . + o E.  |  
|   o   S =. = .  |  
|   .   + =o + +  |  
|   . O ^ . +. o . |  
|   o ^ %.. .    |  
|   + =.o.       |  
+----[SHA256]-----+  
[root@ansible-server-001 .ssh]# |
```

Press 3 times Enter when it prompt to enter.

Once you have created, you will find two files `id_rsa` and `id_rsa.pub` inside the `.ssh` directory - `/root/.ssh/` We are going to use `id_rsa.pub` for the passwordless login.

<< Press Enter

```
ls -l /root/.ssh/
```

```
[root@ansible-server-001 .ssh]# ls -l /root/.ssh/
total 12
-rw-----. 1 root root 1679 Apr 11 19:02 id_rsa
-rw-r--r--. 1 root root 405 Apr 11 19:02 id_rsa.pub
-rw-r--r--. 1 root root 174 Apr 11 18:42 known_hosts
```

`id_rsa` is the private key and `id_rsa.pub` that allow authorization of the user on a server and public key should be registered on the node machine.

Now, log into the client node machine mentioned in step 2 and create the `.ssh` directory in the home directory of the user, if it doesn't exist.

```
mkdir /home/ansible/.ssh
```

Ansible users may be different in your case.

```
chmod 700 /home/ansible/.ssh
```

Create an `authorized_keys` file if it doesn't exist.

```
touch /home/ansible/.ssh/authorized_keys
```

Update the permission of the file.

```
chmod 600 /home/ansible/.ssh/authorized_keys
```

Copy the content of `id_rsa.pub` key by logging into the ansible server node by this command

```
cat id_rsa.pub
```

Paste the content in `authorized_keys` file, which we have copied above, Press `i` for insert mode in vi editors.

```
vi /home/ansible/.ssh/authorized_keys
```

To save and exit - Press `Esc`, and write `:wq!`

To test passwordless login, Now access the managed node using SSH.

```
ssh ansible@34.66.245.56
```

```
[root@ansible-server-001 .ssh]# ssh ansible@34.66.245.56
Last login: Sat Apr 11 18:53:21 2020 from 94.96.223.35.bc.googleusercontent.com
[ansible@ansible-client-001 ~]$ |
```

It will take you to the shell directly without asking the password or pem key

Step 4: Create Ansible Inventory

Edit (or create) `/etc/ansible/hosts` file. This file holds the inventory of remote hosts to which Ansible will connect through SSH for managing them.

```
vi /etc/ansible/hosts
```

```
[stack]
```

```
host1 ansible_ssh_client=34.66.245.56
```

```
[stack]
host1 ansible_ssh_client=34.66.245.56
```

Put one or more client nodes and group it. Here, I have added client node IP to this stack group.

The `stack` is an organizational tag that lets you refer to any servers listed under it with one word.

Specifically, we have to tell Ansible that it should connect to 34.66.245.56 in the “stack” group with the **ansible** user.

```
sudo mkdir /etc/ansible/group_vars
```

Within this folder, we can create YAML-formatted files for each group we want to configure:

```
vi /etc/ansible/group_vars/stack
```

```
---  
ansible_ssh_user: ansible
```

```
---  
ansible_ssh_user: ansible
```

YAML files start with “---”, so make sure you don’t forget that part.

Ping stack group that you have configured by typing:

```
ansible -m ping stack
```

```
[root@ansible-server-001 group_vars]# ansible -m ping stack  
host1 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python"  
  },  
  "changed": false,  
  "ping": "pong"  
}
```

Step5: Create Ansible Playbook

First, move to the opt directory.

```
cd /opt
```

create a playbook called stack.yml, use it any name.

```
vi stack.yml
```

Press i for insert mode and paste this configuration.

```
---
- hosts: stack
  remote_user: ansible
  become: yes
  become_method: sudo

  tasks:

    - name: Install Apache
      yum: pkg=httpd state=latest

    - name: Copy index.html to this location - /var/www/html
      template: src=/etc/ansible/index.html dest=/var/www/html
               owner=apache group=apache mode=0644

    - name: Change the apache default Port from 80 to 8080.
      lineinfile: path=/etc/httpd/conf/httpd.conf regexp="Listen 80"
                 line="Listen 8080" state=present

    - name: Allow 8080 behind the Firewall
      firewallld: port=8080/tcp permanent=yes state=enabled

    - name: Enable and restart Apache on System Boot
      service: name=httpd enabled=yes state=restarted

    - name: Disable SELinux
      selinux: state=disabled

    - name: Reboot the machine to disable SELinux
```



```
reboot: reboot_timeout=300
```

- name: Create a directory if it does not exist
file: path=/opt/mysql5.7 state=directory mode='0755'
- name: Download RPM package for MySQL 5.7
get_url: url=https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm dest=/opt/mysql5.7
- name: Add the new MySQL repository.
yum: name=/opt/mysql5.7/mysql57-community-release-el7-9.noarch.rpm state=present
- name: Install Mysql 5.7
yum: name=mysql-server state=present
- name: Start and Enable MySQL Service
service: name=mysqld enabled=yes state=restarted

Create a demo HTML file (/etc/ansible/index.html). This HTML file will be placed in the default DocumentRoot of Apache server on the managed node.

```
vi /etc/ansible/index.html
```

```
<html>
<head>
  <title>Apache</title>
</head>
<body>
<h1>Welcome to the apache world!!</h1>
<p>Now, Apache is managed through Ansible and Services is UP and Running.</p>
</body>
</html>
```

Step6: Breaking Down Ansible Playbook

Now, we will go through each section of the playbook `stack.yml` file to understand what these are.

```
---
- hosts: stack
  remote_user: ansible
  become: yes
  become_method: sudo

  tasks:
```

All YAML files should begin with (Three dashes) `---` and this indicates the start of a document.

The `hosts'` line is a list of one or more groups (stack which is defined in inventory), separated by colons. Along with the host.

`ansible` is the remote user client node.

If you install any package in Linux machine you need root privileges, so here, `sudo` is the privilege escalations methods.

```
tasks:
- name: Install Apache
  yum: pkg=httpd state=latest

- name: Copy index.html to this location - /var/www/html
  template: src=/etc/ansible/index.html dest=/var/www/html
            owner=apache group=apache mode=0644

- name: Change the apache default Port from 80 to 8080.
  lineinfile: path=/etc/httpd/conf/httpd.conf regexp="Listen 80"
              line="Listen 8080" state=present

- name: Allow 8080 behind the Firewall
  firewallld: port=8080/tcp permanent=yes state=enabled

- name: Enable and restart Apache on System Boot
  service: name=httpd enabled=yes state=restarted

- name: Disable SELinux
  selinux: state=disabled

- name: Reboot the machine to disable Selinux
  reboot: reboot_timeout=300
```

Now, we have a set of tasks.

Each play contains a list of tasks, those are executed in order, one at a time, against the node matched by the host pattern, before moving on to the next task.

Every task should have a name and is used to display the current activity while running the playbook.

Here, we will install the latest version of Apache, copy the demo HTML (`/etc/ansible/index.html`) to `/var/www/html` directory of the managed node.

Change the port 80 to 8080 in `httpd.conf` file and allow port 80 in firewall permanently.

Enable Apache service to start automatically during system boot and finally, we are disabling SELinux permanently by booting the node.

```
- name: Create a directory if it does not exist
  file: path=/opt/mysql5.7 state=directory mode='0755'

- name: Download RPM package for mysql 5.7
  get_url: url=https://dev.mysql.com/get/mysql57-community-release-e17-9.noarch.rpm dest=/opt/mysql5.7

- name: Add the new MySQL repository.
  yum: name=/opt/mysql5.7/mysql57-community-release-e17-9.noarch.rpm state=present

- name: Install MySQL 5.7
```

The above list of tasks creates a directory, downloads the rpm package, installs mysql5.7 and finally starts and enables MySQL 5.7 Service.

Step7: Execute Ansible Playbook

Move to the /opt directory where playbook resides with the following command

```
cd /opt
```

Now, execute the playbook with command.

```
ansible-playbook stack.yml
```

ansible-playbook: Command to run Ansible playbooks.

stack.yml: YAML file (Ansible Playbook file – that we created)

```
[root@ansible-server-001 opt]# ansible-playbook stack.yml

PLAY [stack] *****

TASK [Gathering Facts] *****
ok: [host1]

TASK [Install Apache] *****
ok: [host1]

TASK [Copy index.html to this location - /var/www/html] *****
changed: [host1]

TASK [Change the apache default Port from 80 to 8080.] *****
changed: [host1]

TASK [Allow 8080 behind the Firewall] *****
changed: [host1]

TASK [Enable and restart Apache on System Boot] *****
changed: [host1]

TASK [Disable SELinux] *****
[WARNING]: SELinux state temporarily changed from 'enforcing' to 'permissive'. State change will take effect next reboot.
changed: [host1]

TASK [Reboot the machine to disable Selinux] *****
changed: [host1]

TASK [Create a directory if it does not exist] *****
changed: [host1]

TASK [Download RPM package for mysql 5.7] *****
changed: [host1]

TASK [Add the new MySQL repository.] *****
changed: [host1]

TASK [Install Mysql 5.7] *****
changed: [host1]

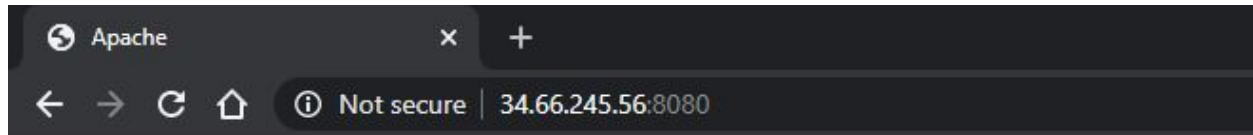
TASK [Strat and Enable mysql Service] *****
changed: [host1]

PLAY RECAP *****
host1                : ok=13  changed=11  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[root@ansible-server-001 opt]# |
```

Step 8: Let's Verify that apache and DB installation

Hit the Managed node IP with followed 8080 port -- `http://34.66.245.56:8080`



Now, Apache is managed through Ansible and Services is UP and Running.

You should now get above page **Apache is managed through Ansible**, and this page confirms us that Apache was installed by Ansible.

Also, check MySQL service by logging into the node

```
mysql -V
```

```
[root@ansible-client-001 ansible]# mysql -V
mysql Ver 14.14 Distrib 5.7.29, for Linux (x86_64) using EditLine wrapper
```

```
systemctl status mysqld
```

```
[root@ansible-client-001 ansible]# systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-04-12 07:34:16 UTC; 4min 59s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 1656 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
   Process: 1634 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
   Main PID: 1659 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─1659 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid

Apr 12 07:34:15 ansible-client-001 systemd[1]: Stopped MySQL Server.
Apr 12 07:34:15 ansible-client-001 systemd[1]: Starting MySQL Server...
Apr 12 07:34:16 ansible-client-001 systemd[1]: Started MySQL Server.
[root@ansible-client-001 ansible]#
```

That's all, We have successfully installed Apache and mysql 5.7 through playbook on Remote node.