

Lab 5.1 Setup Apache Web Server Using Ansible

This section will guide you to:

Setup apache web server using Ansible

This lab has four sub-sections:

- 5.1.1 Install Ansible on Ubuntu
- 5.1.2 Establish connectivity between ansible controller and node machine
- 5.1.3 Write Ansible YAML script to install ansible software
- 5.1.4 Run Ansible YAML script

Step 5.1.1: Install Ansible on Ubuntu

Use the below commands on Ubuntu system to install ansible software.

```
sudo apt-get install software-properties-common
```

```
sudo apt-add-repository ppa:ansible/ansible
```

```
sudo apt-get update
```

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

```
root@docker:~# apt-get install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.24.32.5).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
root@docker:~# apt-add-repository ppa:ansible/ansible
  Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applicationsâautomate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or Ctrl-C to cancel adding it.

Hit:1 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Hit:7 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:8 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease
Hit:9 http://archive.canonical.com/ubuntu bionic InRelease
Reading package lists... Done
root@docker:~# apt install -y ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data python-certifi python-chardet python-jmespath python-kerberos python-libcloud python-lockfile python-netaddr python-openssl python-requests python-selinux
  python-simplejson python-urllib3 python-xmldict
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
```

Establish SSH key pair in linux system to have SSH connectivity with localhost using the following commands:

```
ssh-keygen -t rsa
```

```
cat root/.ssh/id_rsa.pub >> root/.ssh/authorized_keys
```

```
ssh localhost "date"
```

```
root@docker:~# ssh-keygen -t rsa
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:X1G0tNYLQxUf+kvDwbu2L9YBSOFuC+s13JjoK/bi4g8 root@docker
The key's randomart image is:
+---[RSA 2048]---+
|           .+.+.o |
|           .+ *o..|
|           ..O.o..|
|           .o =o.o|
|           S. o. o* |
|           .*. = ..+|
|           E   o.B . =.|
|           ..+o . . + o|
|           ..=o+=. . oo|
+---[SHA256]-----+
root@docker:~# cat .ssh/id_rsa.pub >> .ssh/authorized_keys
root@docker:~# ssh localhost "date"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Smx2lBsnY/AK7a0ZlkC42k8xKVBZZQ2Sah+Hp5PYnlU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Thu Nov 29 14:50:49 UTC 2018
root@docker:~#
```

Now, add the host localhost in ansible host file /etc/ansible/hosts.

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

When the file opens, add the following content:

```
[webservers]
```

localhost

Once that is done, validate using Ansible command:

```
root@docker:~# cat /etc/ansible/hosts
[webservers]
localhost
root@docker:~# ansible -m ping webservers
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@docker:~#
```

Step 5.1.2: Establish connectivity between Ansible controller and node machine
execute the below command to validate host inventory file.

```
ansible -m ping webservers
```

```
root@docker:~# ansible -m ping webservers
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Step 5.1.3: Write Ansible YAML script to install Ansible software

Once connectivity is established, add the following code in the apache.yaml file and proceed with execution.

```
sudo vi apache2.yaml
```

```

- - -
- hosts: webservers
  sudo: yes
  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enabled mod_rewrite
      apache2_module: name=rewrite state=present
      notify:
        - restart apache2

  handlers:
    - name: restart apache2
      service: name=apache2 state=restarted

```

-- INSERT --

15,44

A

Step 5.1.4: Run Ansible YAML script

Run apache.yaml file using below command:

```
ansible-playbook apache2.yaml
```

```

root@docker:~# ansible -m ping webservers
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@docker:~# ansible-playbook apache2.yaml
[DEPRECATION WARNING]: Instead of sudo/sudo_user, use become/become_user and make sure become_method is 'sudo'.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [webservers] *****

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

TASK [install apache2] *****
changed: [localhost]

TASK [enabled mod_rewrite] *****
changed: [localhost]

RUNNING HANDLER [restart apache2] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=4    changed=3    unreachable=0    failed=0

root@docker:~#

```

Validate installation using the command:

```
ansible -m shell -a "service apache2 status" webserver
```

```
root@docker:~# ansible -m shell -a "service apache2 status" webserver
[WARNING]: Consider using the service module rather than running service. If you need to use command because
command task or set command_warnings=False in ansible.cfg to get rid of this message.

localhost | SUCCESS | rc=0 >>
# apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            â€”â€”apache2-systemd.conf
   Active: active (running) since Tue 2018-11-27 09:49:40 UTC; 12min ago
     Process: 6942 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
     Process: 6947 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 6959 (apache2)
    Tasks: 55 (limit: 4401)
   CGroup: /system.slice/apache2.service
           â€”â€”6959 /usr/sbin/apache2 -k start
           â€”â€”6960 /usr/sbin/apache2 -k start
           â€”â€”6961 /usr/sbin/apache2 -k start

Nov 27 09:49:40 docker systemd[1]: Stopped The Apache HTTP Server.
Nov 27 09:49:40 docker systemd[1]: Starting The Apache HTTP Server...
Nov 27 09:49:40 docker systemd[1]: Started The Apache HTTP Server.

root@docker:~#
```