

Assignment 9 :: Deploy Ansible on One of EC2 instance and write a playbook to install httpd server and host website on two ansible hosts

Step 1 :: Create three EC2 instances and name it as below

1. Ansible\_Control\_Node
2. Ansible\_Manage\_Node\_1
3. Ansible\_Manage\_Node\_2

Instances (3) Info										
<input type="text" value="Search"/>										
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	
Ansible_Control_Node	i-0c08fe4b5df402684	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-50-16-62-167.com...	50.16.62.167	50.16.62.167	
Ansible_Manage_Node_1	i-0af63af1466c9486c	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-44-208-70-33.com...	44.208.70.33	44.208.70.33	
Ansible_Manage_Node_2	i-07cbfbed87ec2534	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-224-206-206.co...	3.224.206.206	3.224.206.206	

Step 2 :: Install Ansible in Ansible\_Control\_Node

Update the rhel repo to latest

# yum update -y

```
root@ip-172-31-80-132:~
Verifying : libcrypt-2.26-58.amzn2.x86_64
Verifying : curl-7.79.1-2.amzn2.0.1.x86_64
Verifying : libcurl-7.79.1-2.amzn2.0.1.x86_64
Verifying : glibc-minimal-langpack-2.26-58.amzn2.x86_64
Verifying : glibc-locale-source-2.26-58.amzn2.x86_64
Verifying : glibc-common-2.26-58.amzn2.x86_64
Verifying : glibc-all-langpacks-2.26-58.amzn2.x86_64
Verifying : amazon-ssm-agent-3.1.1188.0-1.amzn2.x86_64
Verifying : initscripts-9.49.47-1.amzn2.0.1.x86_64
Verifying : expat-2.1.0-12.amzn2.0.4.x86_64
Verifying : glibc-2.26-58.amzn2.x86_64

Installed:
kernel.x86_64 0:5.10.126-117.518.amzn2

Updated:
amazon-ssm-agent.x86_64 0:3.1.1575.0-1.amzn2    curl.x86_64 0:7.79.1-4.amzn2.0
glibc.x86_64 0:2.26-59.amzn2                    glibc-all-langpacks.x86_64 0:2
glibc-locale-source.x86_64 0:2.26-59.amzn2      glibc-minimal-langpack.x86_64
libcrypt.x86_64 0:2.26-59.amzn2                libcurl.x86_64 0:7.79.1-4.amzn
yum.noarch 0:3.4.3-158.amzn2.0.6

Complete!
[root@ip-172-31-80-132 ~]#
```

## Install Ansible using EPEL Repository

# wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

```
root@ip-172-31-80-132:~  
[root@ip-172-31-80-132 ~]# wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
--2022-07-15 05:34:13-- https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
Resolving dl.fedoraproject.org (dl.fedoraproject.org)... 38.145.60.22, 38.145.60.23, 38.145.60.24  
Connecting to dl.fedoraproject.org (dl.fedoraproject.org)|38.145.60.22|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 15608 (15K) [application/x-rpm]  
Saving to: 'epel-release-latest-7.noarch.rpm'  
  
100%[=====>] 15,608      --.-K/s   in 0s  
  
2022-07-15 05:34:13 (35.3 MB/s) - 'epel-release-latest-7.noarch.rpm' saved [15608/15608]  
  
[root@ip-172-31-80-132 ~]#
```

# yum install -y epel-release-latest-7.noarch.rpm

```
root@ip-172-31-80-132:~  
=====Installing:  
epel-release      noarch      7-14      /epel-release-latest-7.noarch      25 k  
=====Transaction Summary  
=====Install 1 Package  
Total size: 25 k  
Installed size: 25 k  
Is this ok [y/d/N]: y  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : epel-release-7-14.noarch      1/1  
  Verifying   : epel-release-7-14.noarch      1/1  
  
Installed:  
epel-release.noarch 0:7-14  
  
Complete!  
[root@ip-172-31-80-132 ~]#
```

```
# yum update -y
```

```
root@ip-172-31-80-132:~  
From      : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : python2-simplejson-3.11.1-1.el7.x86_64      1/4  
  Installing : 1:python2-lockfile-0.11.0-17.el7.noarch    2/4  
  Erasing    : 1:python-lockfile-0.9.1-4.amzn2.noarch     3/4  
  Erasing    : python-simplejson-3.2.0-1.amzn2.0.2.x86_64 4/4  
  Verifying  : 1:python2-lockfile-0.11.0-17.el7.noarch    1/4  
  Verifying  : python2-simplejson-3.11.1-1.el7.x86_64     2/4  
  Verifying  : 1:python-lockfile-0.9.1-4.amzn2.noarch     3/4  
  Verifying  : python-simplejson-3.2.0-1.amzn2.0.2.x86_64 4/4  
  
Installed:  
  python2-lockfile.noarch 1:0.11.0-17.el7  
  python2-simplejson.x86_64 0:3.11.1-1.el7  
  
Replaced:  
  python-lockfile.noarch 1:0.9.1-4.amzn2  
  python-simplejson.x86_64 0:3.2.0-1.amzn2.0.2  
  
Complete!  
[root@ip-172-31-80-132 ~]#
```

```
# yum install python python-devel python-pip openssl ansible -y
```

```
root@ip-172-31-80-132:~  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : python2-httpplib2-0.18.1-3.el7.noarch      1/5  
  Installing : sshpass-1.06-1.el7.x86_64                 2/5  
  Installing : python-paramiko-2.1.1-0.10.el7.noarch      3/5  
  Installing : ansible-2.9.27-1.el7.noarch                4/5  
  Installing : python2-pip-20.2.2-1.amzn2.0.3.noarch      5/5  
  Verifying  : python-paramiko-2.1.1-0.10.el7.noarch      1/5  
  Verifying  : python2-pip-20.2.2-1.amzn2.0.3.noarch      2/5  
  Verifying  : sshpass-1.06-1.el7.x86_64                 3/5  
  Verifying  : python2-httpplib2-0.18.1-3.el7.noarch      4/5  
  Verifying  : ansible-2.9.27-1.el7.noarch                5/5  
  
Installed:  
  ansible.noarch 0:2.9.27-1.el7      python2-pip.noarch 0:20.2.2-1.amzn2.0.3  
  
Dependency Installed:  
  python-paramiko.noarch 0:2.1.1-0.10.el7  
  python2-httpplib2.noarch 0:0.18.1-3.el7  
  sshpass.x86_64 0:1.06-1.el7  
  
Complete!  
[root@ip-172-31-80-132 ~]#
```

# amazon-linux-extras install ansible2

```
root@ip-172-31-80-132:~  
43 livepatch available [ =stable ]  
44 python3.8 available [ =stable ]  
45 haproxy2 available [ =stable ]  
46 collectd available [ =stable ]  
47 aws-nitro-enclaves-cli available [ =stable ]  
48 R4 available [ =stable ]  
— kernel-5.4 available [ =stable ]  
50 selinux-ng available [ =stable ]  
51 php8.0 available [ =stable ]  
52 tomcat9 available [ =stable ]  
53 unbound1.13 available [ =stable ]  
54 mariadb10.5 available [ =stable ]  
55 kernel-5.10=latest enabled [ =stable ]  
56 redis6 available [ =stable ]  
57 ruby3.0 available [ =stable ]  
58 postgresql12 available [ =stable ]  
59 postgresql13 available [ =stable ]  
60 mock2 available [ =stable ]  
61 dnsmasq2.85 available [ =stable ]  
62 kernel-5.15 available [ =stable ]  
63 postgresql14 available [ =stable ]  
64 firefox available [ =stable ]  
† Note on end-of-support. Use 'info' subcommand.  
[root@ip-172-31-80-132 ~]#
```

# ansible --version

```
root@ip-172-31-80-132:~  
[root@ip-172-31-80-132 ~]# ansible --version  
ansible 2.9.27  
  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 = /usr/bin/ansible  
  python version = 2.7.18 (default, May 25 2022, 14:30:51) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]  
[root@ip-172-31-80-132 ~]#
```

Step 3 :: Configuring Ansible in all nodes

# useradd ansadmin

# passwd ansadmin

```
root@ip-172-31-80-132:~  
[root@ip-172-31-80-132 ~]# useradd ansadmin  
[root@ip-172-31-80-132 ~]# passwd ansadmin  
Changing password for user ansadmin.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@ip-172-31-80-132 ~]#
```

# visudo

Add below line

ansadmin      ALL=(ALL)      NOPASSWD: ALL

```
## Same thing without a password
# %wheel            ALL=(ALL)            NOPASSWD: ALL
ansadmin            ALL=(ALL)            NOPASSWD: ALL
```

# vi /etc/ssh/sshd\_config

Uncomment the below line

“PasswordAuthentication yes”

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no
```

# service sshd restart

```
root@ip-172-31-80-132:~
[root@ip-172-31-80-132 ~]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-80-132 ~]#
```

Step 4 :: Setup PasswordLess login to all the Manage Nodes from Control node via ansadmin user

# su - ansadmin

\$ ssh-keygen

```
ansadmin@ip-172-31-80-132:~  
[ansadmin@ip-172-31-80-132 ~]$ pwd  
/home/ansadmin  
[ansadmin@ip-172-31-80-132 ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ansadmin/.ssh/id_rsa):  
Created directory '/home/ansadmin/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ansadmin/.ssh/id_rsa.  
Your public key has been saved in /home/ansadmin/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:nwWJTw8oON4rlw6SrVNmjNrHz01KnMxasWbg2dj/U8 ansadmin@ip-172-31-80-132.ec2  
.internal  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  
| . o .  
| o . o =  
| . o . + +  
| . . S * =  
| o.o.o* @ .  
| +.==O / . E|  
| .=*+ O + . .|  
| .o.=o ...|  
+---[SHA256]-----+  
[ansadmin@ip-172-31-80-132 ~]$
```

Copy the public key to Manage Nodes

\$ ssh-copy-id -i /home/ansadmin/.ssh/id\_rsa.pub [ansadmin@172.31.83.149](#)

\$ ssh-copy-id -i /home/ansadmin/.ssh/id\_rsa.pub [ansadmin@172.31.91.139](#)

```
ansadmin@control-node:~  
[ansadmin@control-node ~]$ ssh-copy-id -i /home/ansadmin/.ssh/id_rsa.pub ansadmin@172.31.91.139  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"  
The authenticity of host '172.31.91.139 (172.31.91.139)' can't be established.  
ECDSA key fingerprint is SHA256:5vDSqRqpxfuyAZ3L4JNLhiU6kfRqcZWHP5Jz2A8b15k.  
ECDSA key fingerprint is MD5:e1:02:19:c7:89:c9:f8:b3:ae:23:5a:03:b2:96:a0:9a.  
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 already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
ansadmin@172.31.91.139's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'ansadmin@172.31.91.139'"  
and check to make sure that only the key(s) you wanted were added.
```

Test the PasswordLess login from control node to manage nodes

```
ansadmin@manage-node-2:~  
[ansadmin@control-node ~]$ ssh ansadmin@172.31.91.139  
Last login: Fri Jul 15 06:19:22 2022 from ip-172-31-80-132.ec2.internal  
  
  _ |  _ |  )  
 _ | ( _ | /  Amazon Linux 2 AMI  
 _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
[ansadmin@manage-node-2 ~]$
```

```
ansadmin@manage-node-1:~  
[ansadmin@control-node ~]$ ssh ansadmin@172.31.83.149  
Last login: Fri Jul 15 06:13:17 2022 from ip-172-31-80-132.ec2.internal  
  
  _ |  _ |  )  
 _ | ( _ | /  Amazon Linux 2 AMI  
 _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
[ansadmin@manage-node-1 ~]$
```

Step 5 :: Managing inventory file on Master

Add the below lines at end of file /etc/ansible/hosts

# vi /etc/ansible/hosts

```
172.31.91.139  
172.31.83.149  
[webserver]  
172.31.91.139  
[nginx]  
172.31.83.149
```

```
root@control-node:~  
# Ex 3: A collection of database servers in the 'dbservers' group  
## [dbservers]  
##  
## db01.intranet.mydomain.net  
## db02.intranet.mydomain.net  
## 10.25.1.56  
## 10.25.1.57  
  
# Here's another example of host ranges, this time there are no  
# leading 0s:  
  
## db-[99:101]-node.example.com  
  
172.31.91.139  
172.31.83.149  
  
[webserver]  
172.31.91.139  
  
[nginx]  
172.31.83.149  
-- INSERT --
```

52,14

Bot

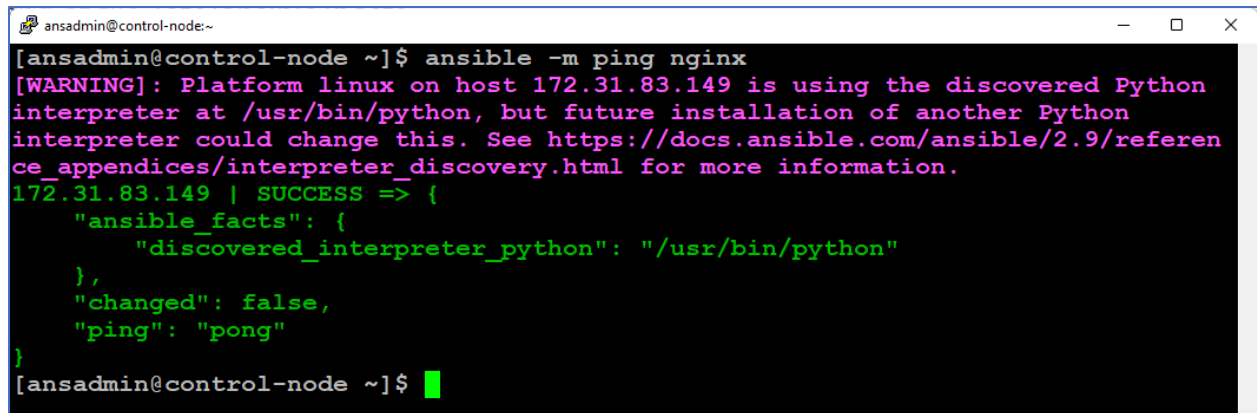
Perform Ping test from Control Node to Manage Nodes from ansadmin user

\$ ansible -m ping webserver

```
ansadmin@control-node:~  
[ansadmin@control-node ~]$ ansible -m ping webserver  
[WARNING]: Platform linux on host 172.31.91.139 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
172.31.91.139 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
[ansadmin@control-node ~]$
```

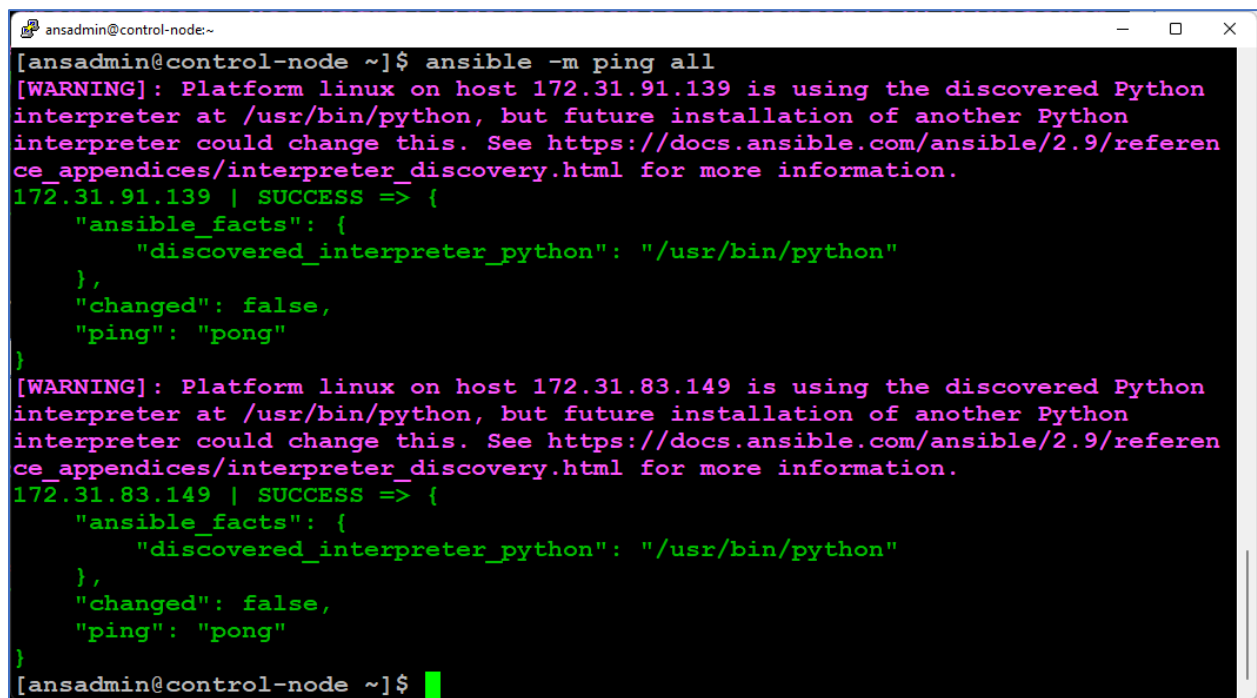


\$ ansible -m ping nginx

A terminal window titled 'ansadmin@control-node:~' showing the command 'ansible -m ping nginx' and its output. The output includes a warning about the Python interpreter and a success message for host 172.31.83.149.

```
[ansadmin@control-node ~]$ ansible -m ping nginx
[WARNING]: Platform linux on host 172.31.83.149 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.83.149 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[ansadmin@control-node ~]$
```

\$ ansible -m ping all

A terminal window titled 'ansadmin@control-node:~' showing the command 'ansible -m ping all' and its output. The output includes warnings for two hosts and success messages for both 172.31.91.139 and 172.31.83.149.

```
[ansadmin@control-node ~]$ ansible -m ping all
[WARNING]: Platform linux on host 172.31.91.139 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.91.139 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.83.149 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.83.149 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[ansadmin@control-node ~]$
```

Step 6 :: Write a playbook to install httpd and copy the index.html in manage node 1 (webserver) and install nginx in manage node 2 (nginx)

[ansadmin@control-node ~]\$ cat hari\_playbook.yml

```
- name: Creating a Webserver
  become: yes
  remote_user: ansadmin
  hosts: webserver
  tasks:
  - name: install httpd package
    yum:
      name:
```

- httpd
  - state: present
- name: start service httpd
  - service:
    - name: httpd
    - state: started
    - enabled: yes
- name: create a directory
  - file:
    - path: /devweb
    - state: directory
    - mode: 02775
    - setype: httpd\_sys\_content\_t
- name: create file
  - file:
    - path: /devweb/index.html
    - state: touch
- name: copy the contents
  - copy:
    - content: "Welcome to Hariharan's WebServer page !!!\n"
    - dest: /devweb/index.html
- name: link a file
  - file:
    - src: /devweb
    - dest: /var/www/html/devweb
    - state: link
- name: Install nginx package in nginx server
  - become: yes
  - remote\_user: ansadmin
  - hosts: nginx
  - tasks:
    - name: Get the EPEL repo
      - get\_url:
        - url: <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>
        - dest: /home/ansadmin/epel-release-latest-7.noarch.rpm
    - name: install EPEL repo
      - yum:
        - name: /home/ansadmin/epel-release-latest-7.noarch.rpm
        - state: present
    - name: install nginx package
      - yum:
        - name:
          - nginx
        - state: present
    - name: start service nginx
      - service:
        - name: nginx

```
state: started
enabled: yes
[ansadmin@control-node ~]$
```

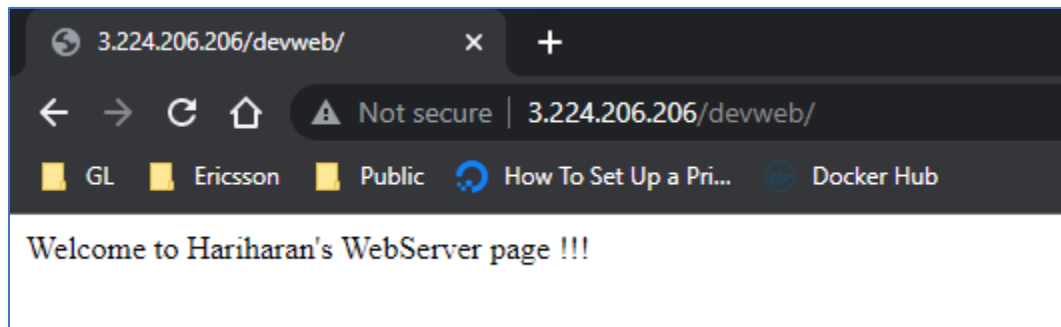
Run the playbook

```
$ ansible-playbook hari_playbook.yml
```

```
ansadmin@control-node ~$ ansible-playbook hari_playbook.yml
PLAY [Creating a Webserver] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.91.139 is using the discovered Python interpreter at /usr/bin/python, but future installation of another
Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.91.139]
TASK [install httpd package] *****
ok: [172.31.91.139]
TASK [start service httpd] *****
ok: [172.31.91.139]
TASK [create a directory] *****
ok: [172.31.91.139]
TASK [create file] *****
changed: [172.31.91.139]
TASK [copy the contents] *****
ok: [172.31.91.139]
TASK [link a file] *****
ok: [172.31.91.139]
PLAY [Install nginx package in nginx server] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.83.149 is using the discovered Python interpreter at /usr/bin/python, but future installation of another
Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.83.149]
TASK [Get the EPEL repo] *****
ok: [172.31.83.149]
TASK [install EPEL repo] *****
ok: [172.31.83.149]
TASK [install nginx package] *****
ok: [172.31.83.149]
TASK [start service nginx] *****
changed: [172.31.83.149]
PLAY RECAP *****
172.31.83.149 : ok=5 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
172.31.91.139 : ok=7 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ansadmin@control-node ~]$
```

## Result:

Webpage is configured in webserver



Nginx is successfully installed in nginx server

