

## **How to set up web server in Managed node(s) using Ansible Playbook?**

**ARVIND BHIKAJI RAMUGADE**

*Corresponding Author: ARVIND BHIKAJI RAMUGADE*

-----  
Date of Submission: 01-08-2020

Date of Acceptance: 16-08-2020  
-----

Ansible is a tool for configuration management. It's provided by Python internally & comes with Python 3. Hence for Ansible installation we need to use pip3 python package manager. pip3 install Ansible is the command to install Ansible. We can verify installation using Ansible — version command. This should be the latest Ansible version i.e. 2.9.11.

### **Traditional Automation Approach:**

It uses imperative programming languages such as python, perl which handles following

1. What to do?
2. How to do? — For handling this the script is not intelligent enough to know the commands which needs to be run on different O.S. e.g. to install Firefox on Ubuntu/Rhel8 the command is not known to programming language this 2nd part is effectively handled using Intelligent Automation approach.

### **Intelligent Automation approach:**

This uses declarative language & is independent of the O.S. configuration on which it needs to be run. It's intelligent enough because it knows how to do the configuration/installation of softwares on other O.S. platforms. E.g. Ansible

Ansible uses inventory which is list of IP addresses of managed nodes. By default Config file name is ansible.cfg. It's stored under /etc. Pip does not create config file during Ansible installation.

Ansible has 2 types of nodes.

1. Controller node — on this node Ansible is installed.
2. Managed nodes — These nodes are managed by controller node on which Ansible performs configuration (software installation, any other config changes)
3. Inventory — this is a database which stores IP of managed nodes which is required by Ansible config file.
4. Ansible config file contains the path of inventory file to access managed nodes & perform configurations. Inventory path is given under defaults header. e.g.

Below is an example

```
[defaults]
```

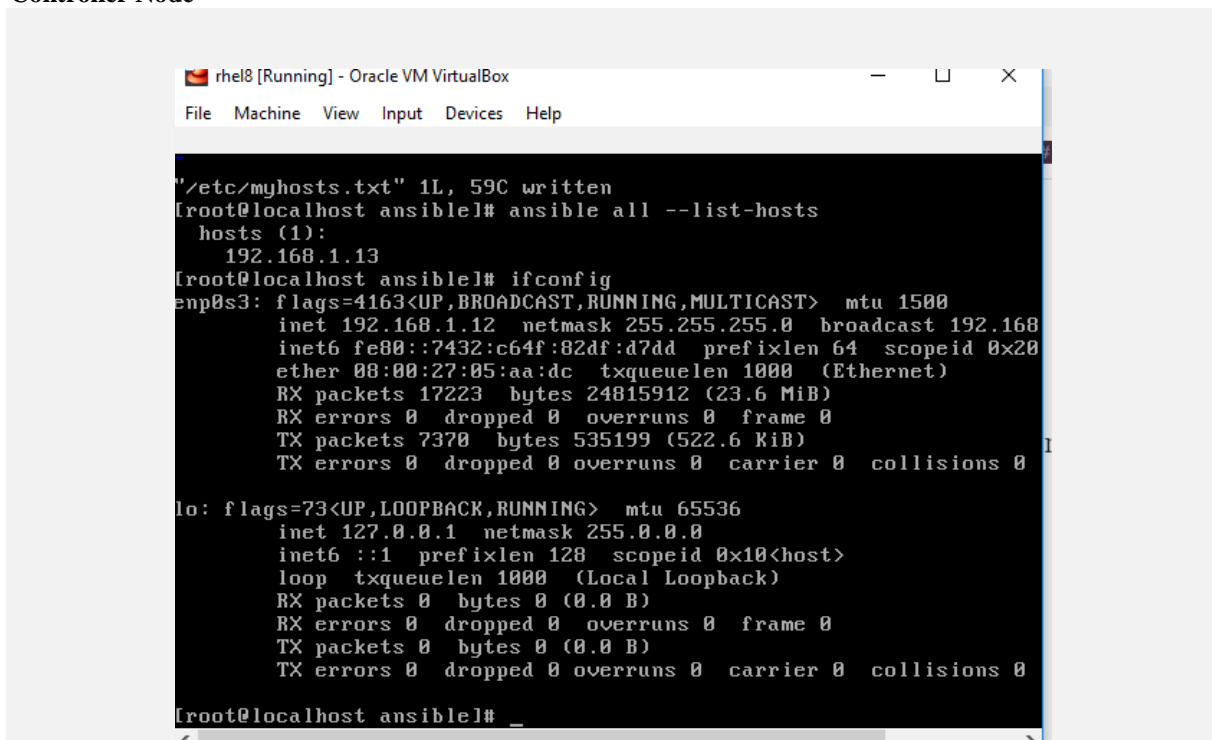
```
inventory =/etc/myhosts.txt
```

Where myhosts.txt contains IP of managed nodes

Internal protocol of Ansible uses SSH for Linux systems. In order to access managed nodes we need to get SSH credentials from controller node. We can use Ssh password setup for that. rpm file can be downloaded from internet for it & we can use below command for installation

```
rpm -ivh<rpm file name>
```

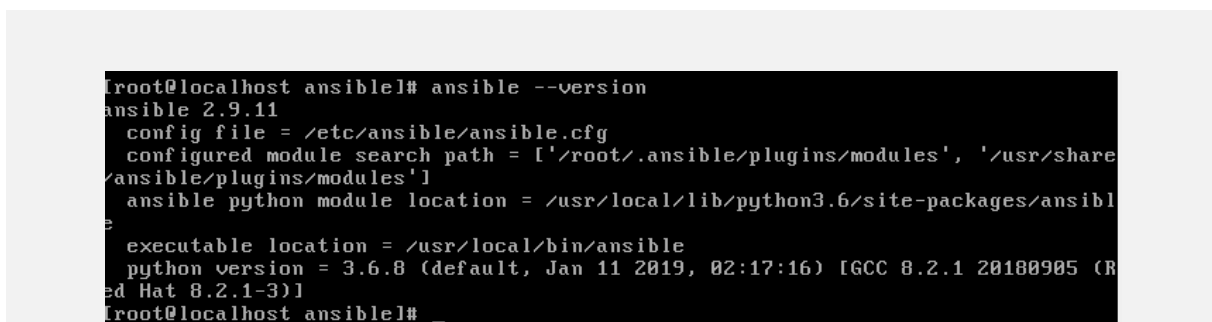
## Controller Node



```
rhel8 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

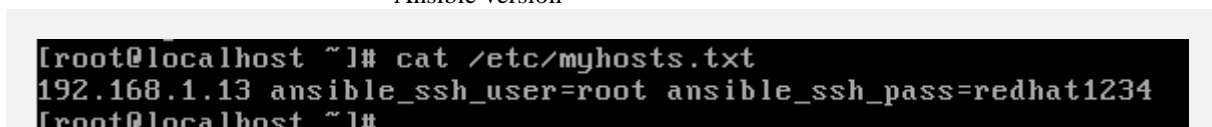
"/etc/myhosts.txt" 1L, 59C written
[root@localhost ansible]# ansible all --list-hosts
hosts (1):
  192.168.1.13
[root@localhost ansible]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.1
  inet6 fe80::7432:c64f:82df:d7dd prefixlen 64 scopeid 0x20<link>
  ether 08:00:27:05:aa:dc txqueuelen 1000 (Ethernet)
  RX packets 17223 bytes 24815912 (23.6 MiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 7370 bytes 535199 (522.6 KiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@localhost ansible]# _
```



```
[root@localhost ansible]# ansible --version
ansible 2.9.11
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.6/site-packages/ansible
  executable location = /usr/local/bin/ansible
  python version = 3.6.8 (default, Jan 11 2019, 02:17:16) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]
[root@localhost ansible]# _
```

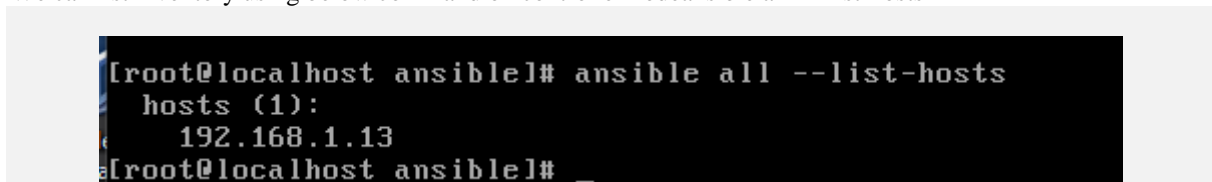
Ansible version



```
[root@localhost ~]# cat /etc/myhosts.txt
192.168.1.13 ansible_ssh_user=root ansible_ssh_pass=redhat1234
[root@localhost ~]#
```

Inventory

We can list inventory using below command on controller node



```
[root@localhost ansible]# ansible all --list-hosts
hosts (1):
  192.168.1.13
[root@localhost ansible]# _
```

Listing inventory

```
root@localhost ansible1# ping goo.gl
PING goo.gl (172.217.174.238) 56(84) bytes of data:
4 bytes from bom12s03-in-f14.1e100.net (172.217.174.238): icmp_seq=1 ttl=64 time=14.5 ms
4 bytes from bom12s03-in-f14.1e100.net (172.217.174.238): icmp_seq=2 ttl=64 time=15.9 ms
4 bytes from bom12s03-in-f14.1e100.net (172.217.174.238): icmp_seq=3 ttl=64 time=16.5 ms
C
-- goo.gl ping statistics --
 3 packets transmitted, 3 received, 0% packet loss, time 7ms
 rtt min/avg/max/mdev = 14.460/15.628/16.493/0.857 ms
root@localhost ansible1#
```

Ensure Connectivity to internet

How to install Firefox software on Managed node?

Ansible uses push mechanism, to config managed nodes. It pushes the code to managed nodes and run it there. Ansible do not require any software to run on managed node. This is known as agent less. Other configuration management tools like puppet and chef uses pull mechanism. They pull the code from the controller node and run it. Managed node should have puppet or chef installed on it. They are known as agents.

When Ansible runs the code on managed node it first check the current state and match it with the desired state. If current state matches with desired state, ansible do not run the code, but if they don't match ansible run the code and configure managed nodes according to desired state. This is known as Idem-potency.

To install Firefox on managed node run

“ansible all -m package -a “name=firefox state=present”

Where -m stands for module,

- a stands for attribute.
- name and state are the attributes, where name gives the package name and state tells to install or remove package from managed node.

```
root@localhost ansible1# ansible all -m package -a "name=firefox state=present"
192.168.1.13 | FAILED! => {
  "msg": "Using a SSH password instead of a key is not possible because Host Key Checking is enabled and sshpass does not support this. Please add this host's fingerprint to your known_hosts file to manage this host."
}
```

Install firefox on managed nodes

To handle above error we need to include Host\_key\_checking as false in ansible config file.

```
[root@localhost ~]# cat /etc/ansible/ansible.cfg
[defaults]
inventory=/etc/myhosts.txt
Host_key_checking=false
[root@localhost ~]#
```

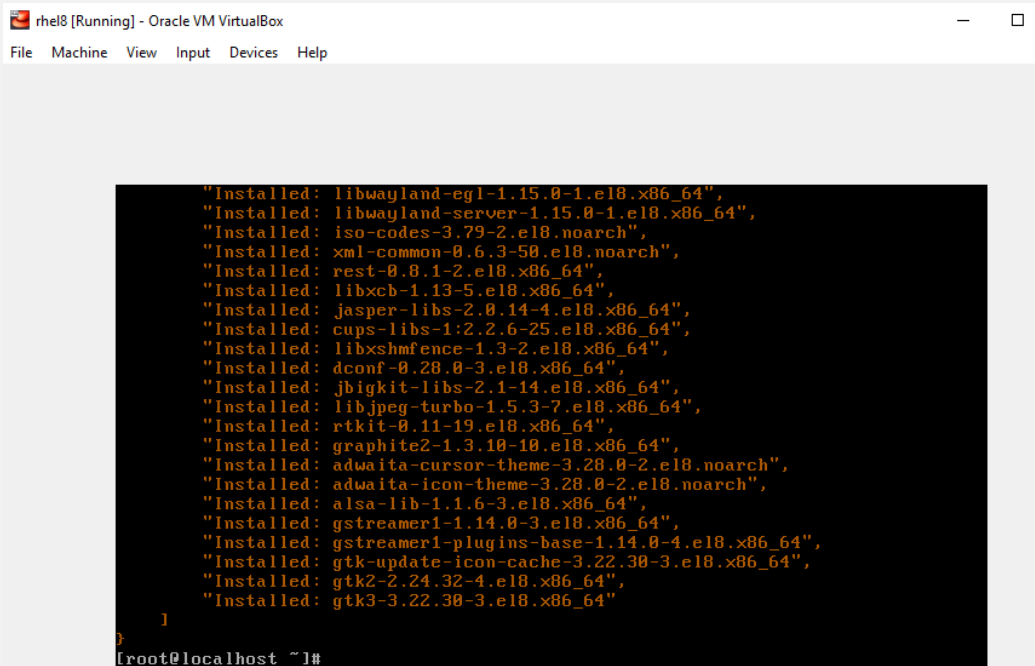
Ansible config file

To check connectivity with managed node(s) we can use below command

```
[root@localhost ~]# ansible all -m ping
192.168.1.13 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
```

Connectivity to managed nodes

Installation of Firefox will show following screen on Controller node



```
rhel8 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

"Installed: libwayland-egl-1.15.0-1.e18.x86_64",
"Installed: libwayland-server-1.15.0-1.e18.x86_64",
"Installed: iso-codes-3.79-2.e18.noarch",
"Installed: xml-common-0.6.3-50.e18.noarch",
"Installed: rest-0.8.1-2.e18.x86_64",
"Installed: libxcb-1.13-5.e18.x86_64",
"Installed: jasper-libs-2.0.14-4.e18.x86_64",
"Installed: cups-libs-1:2.2.6-25.e18.x86_64",
"Installed: libxshmfence-1.3-2.e18.x86_64",
"Installed: dconf-0.28.0-3.e18.x86_64",
"Installed: jbigkit-libs-2.1-14.e18.x86_64",
"Installed: libjpeg-turbo-1.5.3-7.e18.x86_64",
"Installed: rtkit-0.11-19.e18.x86_64",
"Installed: graphite2-1.3.10-10.e18.x86_64",
"Installed: adwaita-cursor-theme-3.28.0-2.e18.noarch",
"Installed: adwaita-icon-theme-3.28.0-2.e18.noarch",
"Installed: alsa-lib-1.1.6-3.e18.x86_64",
"Installed: gstreamer1-1.14.0-3.e18.x86_64",
"Installed: gstreamer1-plugins-base-1.14.0-4.e18.x86_64",
"Installed: gtk-update-icon-cache-3.22.30-3.e18.x86_64",
"Installed: gtk2-2.24.32-4.e18.x86_64",
"Installed: gtk3-3.22.30-3.e18.x86_64"
}
[root@localhost ~]#
```

Installation of Firefox on managed node

```
[root@localhost ~]# cat web.yml
- hosts: 192.168.1.13
  tasks:
  - package: "name=httpd state=present"
  - copy: "src=home.html dest=/var/www/html/"
  - service: "name=httpd state=started"
```

Web Server set up on managed node using Ansible Playbook

```
[root@localhost ~]# cat home.html
Hello World !!!
[root@localhost ~]#
```

```
[root@localhost ~]# ansible-playbook web.yml
PLAY [192.168.1.13] *****
TASK [Gathering Facts] *****
ok: [192.168.1.13]

TASK [package] *****
changed: [192.168.1.13]

TASK [copy] *****
changed: [192.168.1.13]

TASK [service] *****
changed: [192.168.1.13]

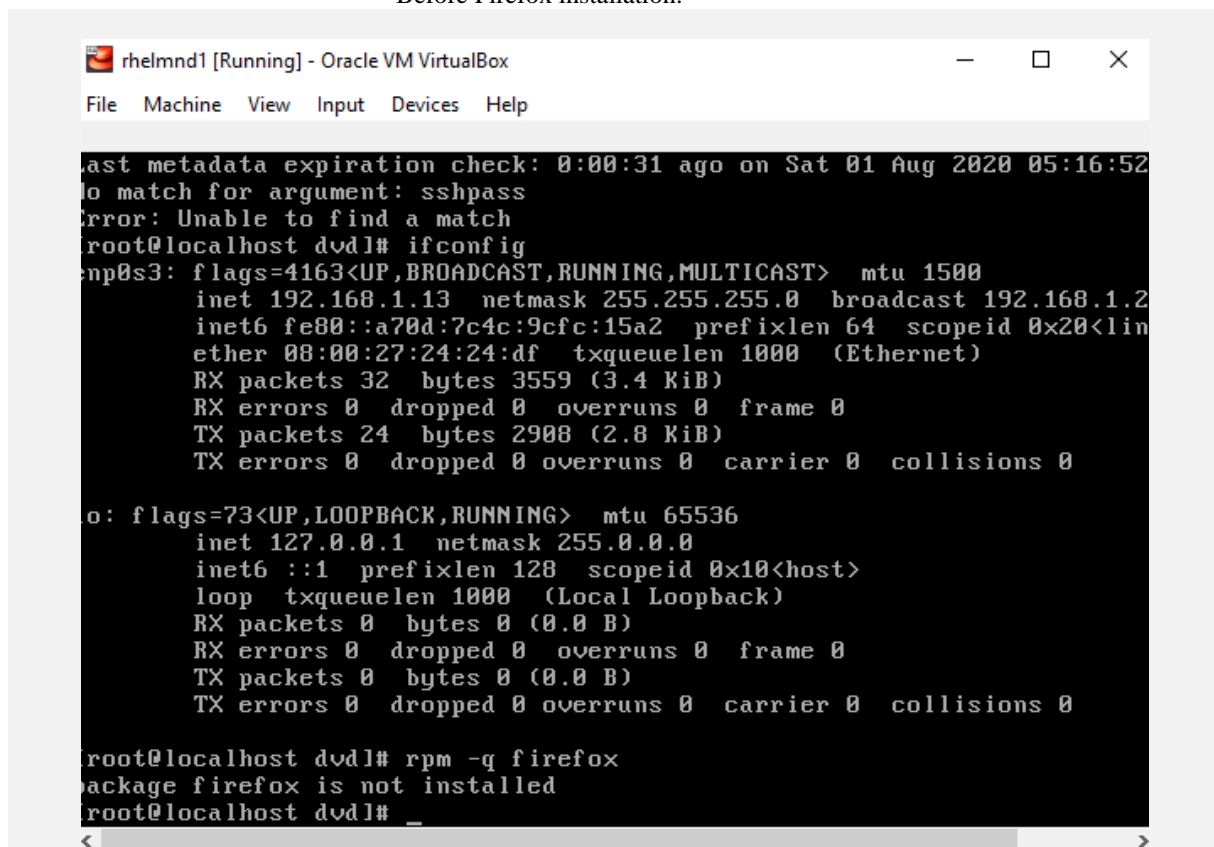
PLAY RECAP *****
192.168.1.13 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[root@localhost ~]# _
```

Ansible Playbook execution

### Managed Node

Before Firefox installation:



```
rhelcmd1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Last metadata expiration check: 0:00:31 ago on Sat 01 Aug 2020 05:16:52
No match for argument: sshpass
Error: Unable to find a match
root@localhost dvd1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.13 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a70d:7c4c:9cfc:15a2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:24:24:df txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3559 (3.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 2908 (2.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@localhost dvd1# rpm -q firefox
package firefox is not installed
root@localhost dvd1# _
```

Managed Node

After Firefox installation:

```
root@localhost dvd1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.13 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a70d:7c4c:9cfc:15a2 prefixlen 64 scopeid 0x20<link-local>
    ether 08:00:27:24:24:df txqueuelen 1000 (Ethernet)
    RX packets 631 bytes 452529 (441.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 287 bytes 40637 (39.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@localhost dvd1# rpm -q firefox
firefox-60.5.1-1.el8.x86_64
root@localhost dvd1# _
```

Successful installation of Firefox on managed node

Managed Node after Apache Web Server set up

```
[root@localhost dvd1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.13 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a70d:7c4c:9cfc:15a2 prefixlen 64 scopeid 0x20<link-local>
    ether 08:00:27:24:24:df txqueuelen 1000 (Ethernet)
    RX packets 1431 bytes 1193714 (1.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 614 bytes 85852 (83.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost dvd1# rpm -q httpd
httpd-2.4.37-10.module+el8+2764+7127e69e.x86_64
[root@localhost dvd1#
```

Successful installation of Apache Web Server on Managed Node

```
[root@localhost dvd1# cat /var/www/html
cat: /var/www/html: Is a directory
[root@localhost dvd1# cd /var/www/html
[root@localhost html1# ls
home.html
[root@localhost html1# cat home.html
Hello World !!!
[root@localhost html1#
```

```
[root@localhost html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-08-01 18:03:12 IST; 11min ago
     Docs: man:httpd.service(8)
  Main PID: 11254 (httpd)
   Status: "Running, listening on: port 80"
    Tasks: 213 (limit: 6092)
   Memory: 24.7M
   CGroup: /system.slice/httpd.service
           └─11254 /usr/sbin/httpd -DFOREGROUND
             └─11255 /usr/sbin/httpd -DFOREGROUND
               └─11256 /usr/sbin/httpd -DFOREGROUND
                 └─11257 /usr/sbin/httpd -DFOREGROUND
                   └─11258 /usr/sbin/httpd -DFOREGROUND

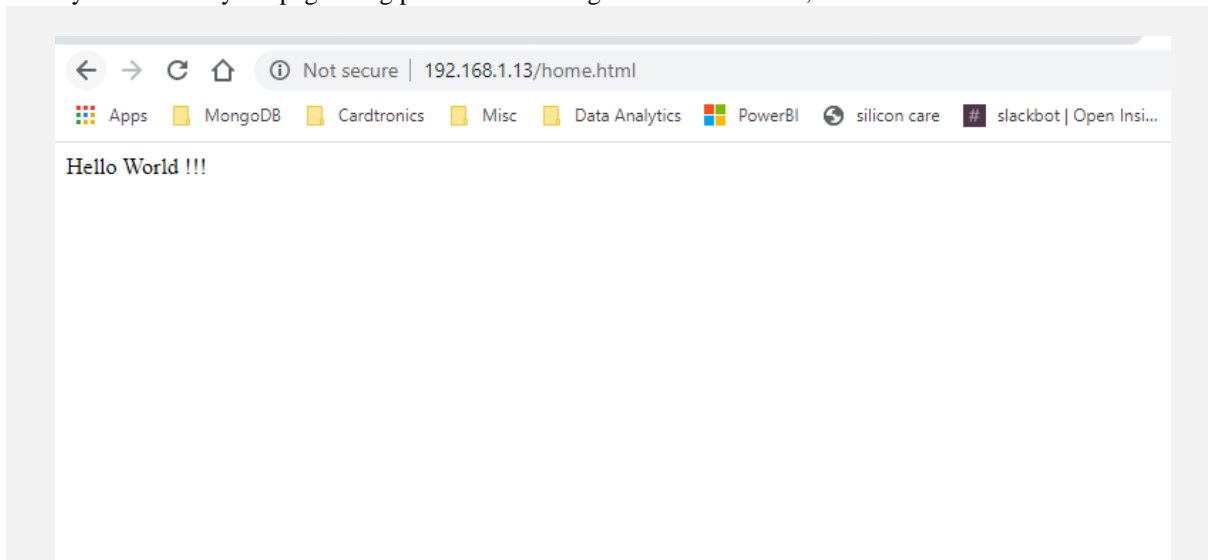
Aug 01 18:03:12 localhost.localdomain systemd[1]: Starting The Apache HTTP Server: httpd.
Aug 01 18:03:12 localhost.localdomain httpd[11254]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, because the
Aug 01 18:03:12 localhost.localdomain httpd[11254]: Server configured.
Aug 01 18:03:12 localhost.localdomain systemd[1]: Started The Apache HTTP Server: httpd.
lines 1-19/19 (END)
```

Statues of Web Server

```
[root@localhost html]# firewall-cmd --permanent --add-port=80/tcp
success
[root@localhost html]# firewall-cmd --reload
success
```

Add port 80

Finally we can verify the page using public IP of managed node as follows,.



Web page launched finally!!

This way we can configure/install software(s), folders & other configuration management using Ansible tool. Ansible uses modules which is intelligent enough to perform how to do action on managed nodes. This makes our life simpler as we don't have to learn commands of individual systems as Ansible managed that part using intelligent modules.

**Apache Web Server Set up on Managed Node(s) using Ansible Tool!!**



WRITTEN BY

**Arvind Ramugade**