1.Create a Ubuntu-based VM using Vagrant, Install any web-server on the VM and access using browser

Step 1 : Create VM using Vagrant

```
PS C:\vag vm\test7> vagrant init ubuntu/xenial64
```

--> This initializes the current directory to be a Vagrant environment by creating an initial Vagrantfile.

--> Vagrantfile contains the config informatiom for the VM

--> Using Vagrant cloud we get predefined operating package format for Vagrant environments

Vagrant File :

```
Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/xenial64"

  config.vm.network "forwarded_port", guest: 80, host: 8080

  config.vm.network "private_network", ip:"172.19.1.100"

end
```

- config.vm.box = "ubuntu/xenial64" --> taking ubuntu os from vagrant cloud
- config.vm.network :forwarded_port, guest: 80, host: 8089 --> forwarded ports allows to access a port on your host machine & all data forwarded to via port 80(guest machine) to 8080(host machine)
- config.vm.network "private_network", ip:"172.19.1.100" --> we set the ip address for the VM

```
PS C:\vag vm\test7> vagrant up --provider virtualbox
```

--> Using 'vagrant up' - creates and configures guest machine according to the Vagrantfile

--> '--provider' Vagrant will verify the box provider

```
PS C:\vag vm\test7> vagrant ssh
```

--> Connects via SSH to an environment

Step 2 : Install any web-server on the VM

```
  $sudo apt-get update      --> update the ubuntu
  $sudo apt-get install nginx-light   --> install the nginx web server on ubuntu
  $cd /var/www/html  -->  default Ubuntu document root
  $sudo su  --> login as a root user
  $cat > index.html
<html lang="en">
<meta charset="UTF-8">
<title>Page Title</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet" href="">
<style>
html,body {font-family:"Verdana",sans-serif}
h1,h2,h3,h4,h5,h6 {font-family:"Segoe UI",sans-serif}
</style>
<script src=""></script>
<body>


<h1>Welcome to DevOps World</h1>
```

```
<p>Let's Build the future from DevOps.</p>

<p>Quate for the day : Try until you win.</p>


</body>

</html>
```

$exit --> logout from root user

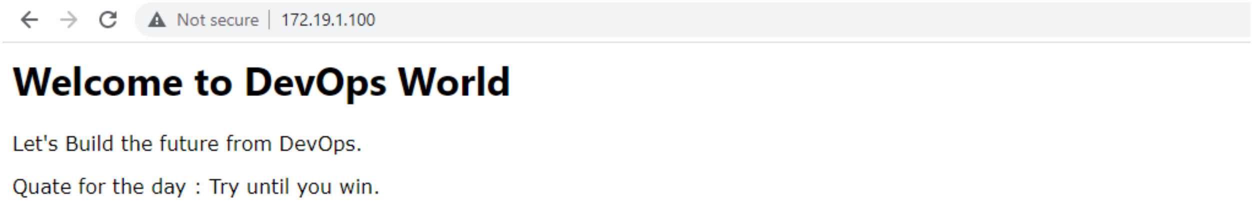
$exit --> logout from the session

Step 3 : Access webpage using browser on host machine

PS C:\vag vm\test7> vagrant reload --> it check Vagrantfile changes and restart the VM

On Browser :

http://172.19.1.100/

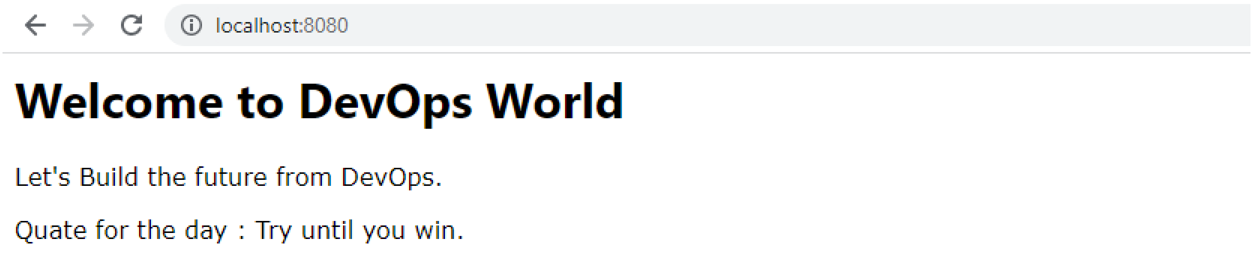


Welcome to DevOps World

Let's Build the future from DevOps.

Quate for the day : Try until you win.

http://localhost:8080/



Welcome to DevOps World

Let's Build the future from DevOps.

Quate for the day : Try until you win.

2.Create 2 VM using Vagrant File

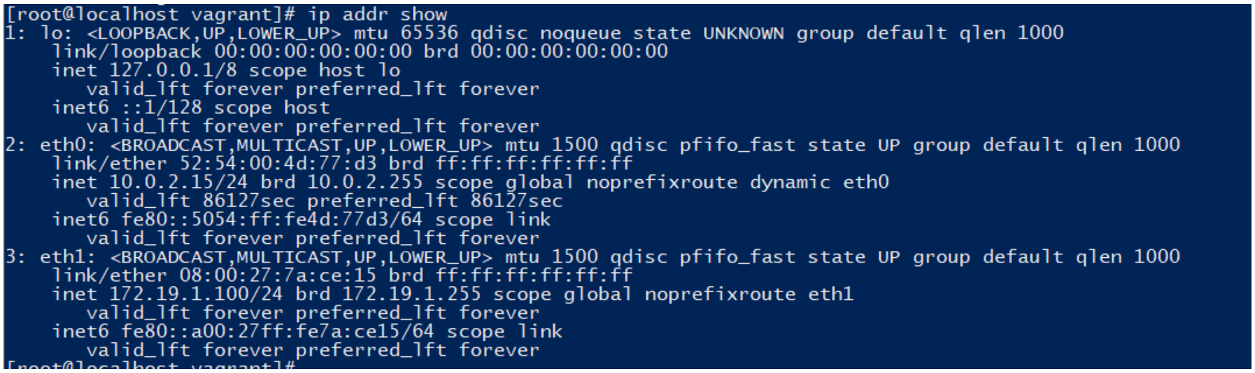
Vagrant File 1 :

```
Vagrant.configure("2") do |config|
  config.vm.box = "centos/7"
  config.vm.network "private_network", ip:"172.19.1.100"
end
```

```
PS C:\vag vm\test9> vagrant init centos/7
PS C:\vag vm\test9> vagrant up --provider virtualbox
PS C:\vag vm\test9> vagrant shh
```

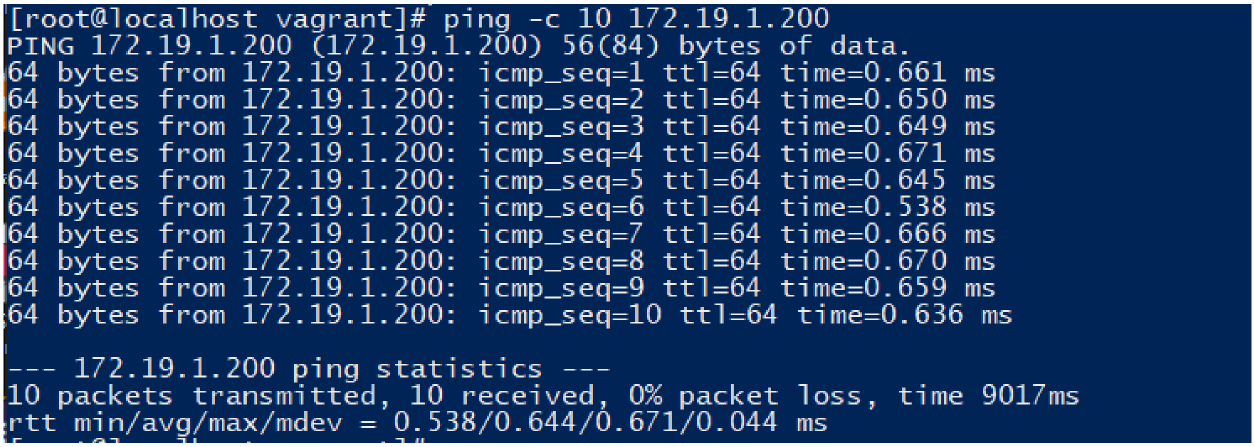
Logged into VM1 :

```
$ip addr show
```



```
[root@localhost vagrant]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:4d:77:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic eth0
       valid_lft 86127sec preferred_lft 86127sec
    inet6 fe80::5054:ff:fe4d:77d3/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:7a:ce:15 brd ff:ff:ff:ff:ff:ff
    inet 172.19.1.100/24 brd 172.19.1.255 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7a:ce15/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost vagrant]#
```

$ping -c 10 172.19.1.200



```
[root@localhost vagrant]# ping -c 10 172.19.1.200
PING 172.19.1.200 (172.19.1.200) 56(84) bytes of data.
64 bytes from 172.19.1.200: icmp_seq=1 ttl=64 time=0.661 ms
64 bytes from 172.19.1.200: icmp_seq=2 ttl=64 time=0.650 ms
64 bytes from 172.19.1.200: icmp_seq=3 ttl=64 time=0.649 ms
64 bytes from 172.19.1.200: icmp_seq=4 ttl=64 time=0.671 ms
64 bytes from 172.19.1.200: icmp_seq=5 ttl=64 time=0.645 ms
64 bytes from 172.19.1.200: icmp_seq=6 ttl=64 time=0.538 ms
64 bytes from 172.19.1.200: icmp_seq=7 ttl=64 time=0.666 ms
64 bytes from 172.19.1.200: icmp_seq=8 ttl=64 time=0.670 ms
64 bytes from 172.19.1.200: icmp_seq=9 ttl=64 time=0.659 ms
64 bytes from 172.19.1.200: icmp_seq=10 ttl=64 time=0.636 ms

--- 172.19.1.200 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/dev = 0.538/0.644/0.671/0.044 ms
[root@localhost vagrant]#
```

Vagrant File 2 :

```
Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/xenial64"

  config.vm.network "private_network", ip:"172.19.1.200"

end
```

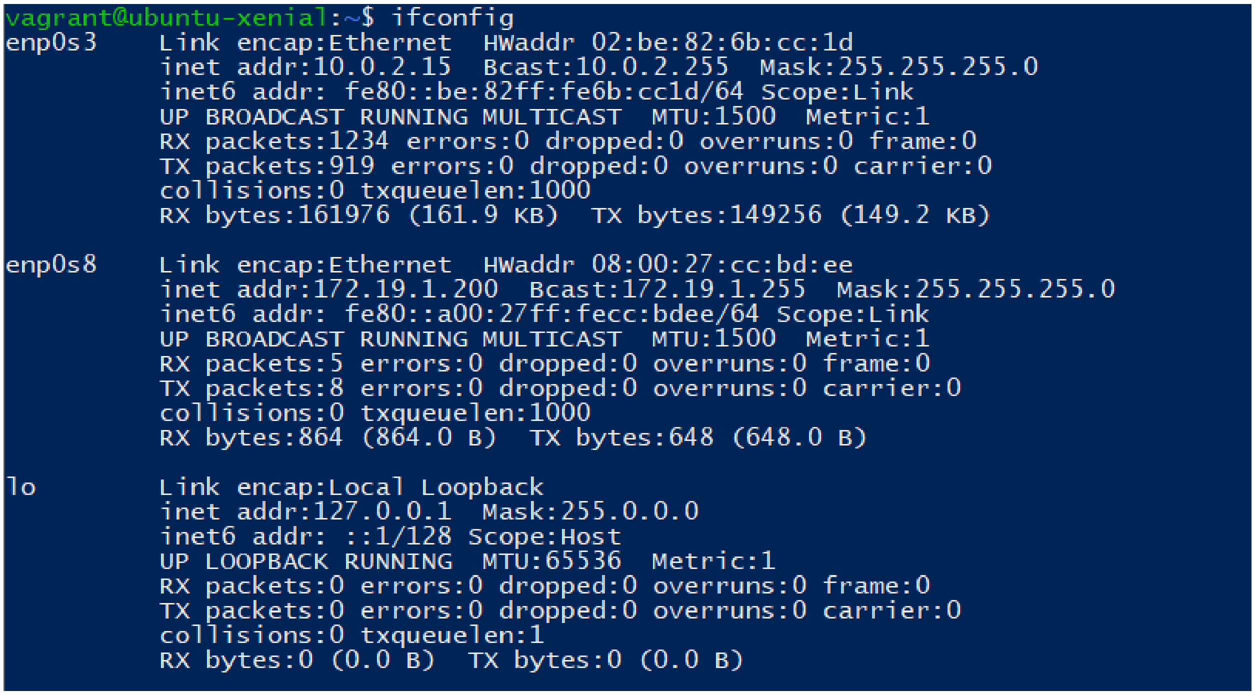
```
PS C:\vag vm\test9> vagrant init ubuntu/xenial64

PS C:\vag vm\test9> vagrant up --provider virtualbox

PS C:\vag vm\test9> vagrant shh
```

Logged into VM2 :

$ifconfig

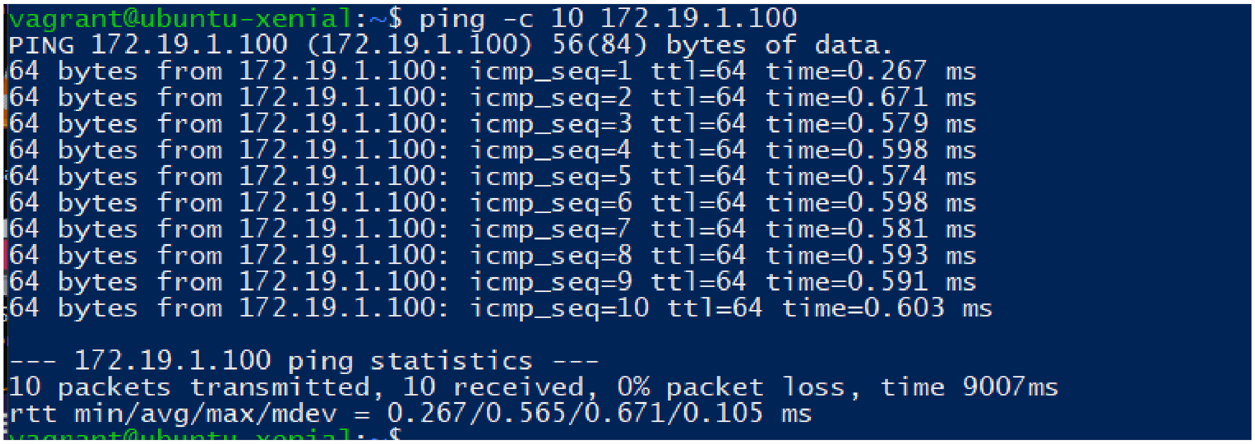


```
vagrant@ubuntu-xenial:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 02:be:82:6b:cc:1d
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::be:82ff:fe6b:cc1d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1234 errors:0 dropped:0 overruns:0 frame:0
          TX packets:919 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:161976 (161.9 KB)  TX bytes:149256 (149.2 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:cc:bd:ee
          inet addr:172.19.1.200  Bcast:172.19.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fecc:bdee/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:864 (864.0 B)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

$ping -c 10 172.19.1.100



```
vagrant@ubuntu-xenial:~$ ping -c 10 172.19.1.100
PING 172.19.1.100 (172.19.1.100) 56(84) bytes of data.
64 bytes from 172.19.1.100: icmp_seq=1 ttl=64 time=0.267 ms
64 bytes from 172.19.1.100: icmp_seq=2 ttl=64 time=0.671 ms
64 bytes from 172.19.1.100: icmp_seq=3 ttl=64 time=0.579 ms
64 bytes from 172.19.1.100: icmp_seq=4 ttl=64 time=0.598 ms
64 bytes from 172.19.1.100: icmp_seq=5 ttl=64 time=0.574 ms
64 bytes from 172.19.1.100: icmp_seq=6 ttl=64 time=0.598 ms
64 bytes from 172.19.1.100: icmp_seq=7 ttl=64 time=0.581 ms
64 bytes from 172.19.1.100: icmp_seq=8 ttl=64 time=0.593 ms
64 bytes from 172.19.1.100: icmp_seq=9 ttl=64 time=0.591 ms
64 bytes from 172.19.1.100: icmp_seq=10 ttl=64 time=0.603 ms

--- 172.19.1.100 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9007ms
rtt min/avg/max/mdev = 0.267/0.565/0.671/0.105 ms
vagrant@ubuntu-xenial:~$
```