

```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tirth>minikube start
* minikube v1.16.0 on Microsoft Windows 10 Home Single Language 10.0.19042 Build 19042
* Using the virtualbox driver based on existing profile
* minikube 1.21.0 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.21.0
* To disable this notice, run: 'minikube config set WantUpdateNotification false'

* Starting control plane node minikube in cluster minikube
* Restarting existing virtualbox VM for "minikube" ...
* Preparing Kubernetes v1.20.0 on Docker 20.10.0 ...
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\tirth>
```

Login to Minikube VM.

User Name: **docker**

Password: **tcuser**

```
minikube [Running] - Oracle VM VirtualBox

$ cd /etc/kubernetes/
$ ls
addons      controller-manager.conf  manifests
admin.conf  kubelet.conf             scheduler.conf

$ sudo cp admin.conf /home/docker/

$ cd /home/docker/
$ ls
admin.conf

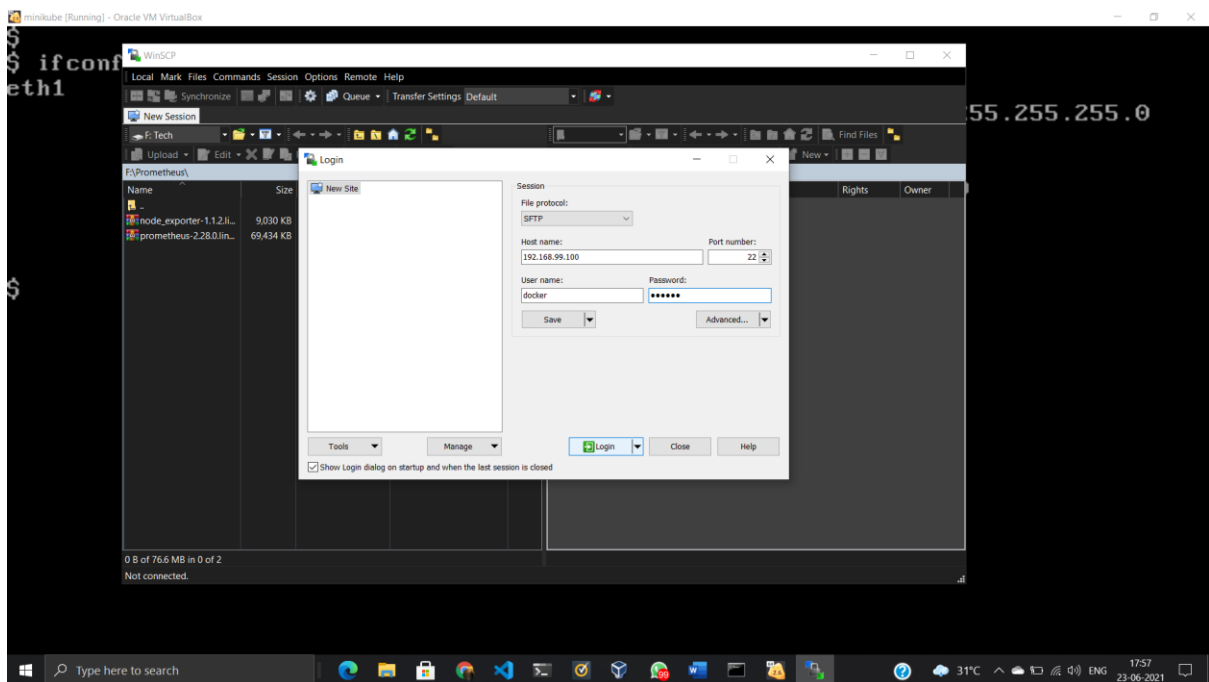
$ sudo chmod 777 admin.conf
$ ls -lah
total 8.0K
drwxr-xr-x 3 docker docker 80 Jun 23 12:29 .
drwxr-xr-x 3 root  root  60 Dec 14 2020 ..
drwx----- 2 docker docker 80 Jan 1 1970 .ssh
-rwxrwxrwx 1 root  root  5.5K Jun 23 12:29 admin.conf
$
```

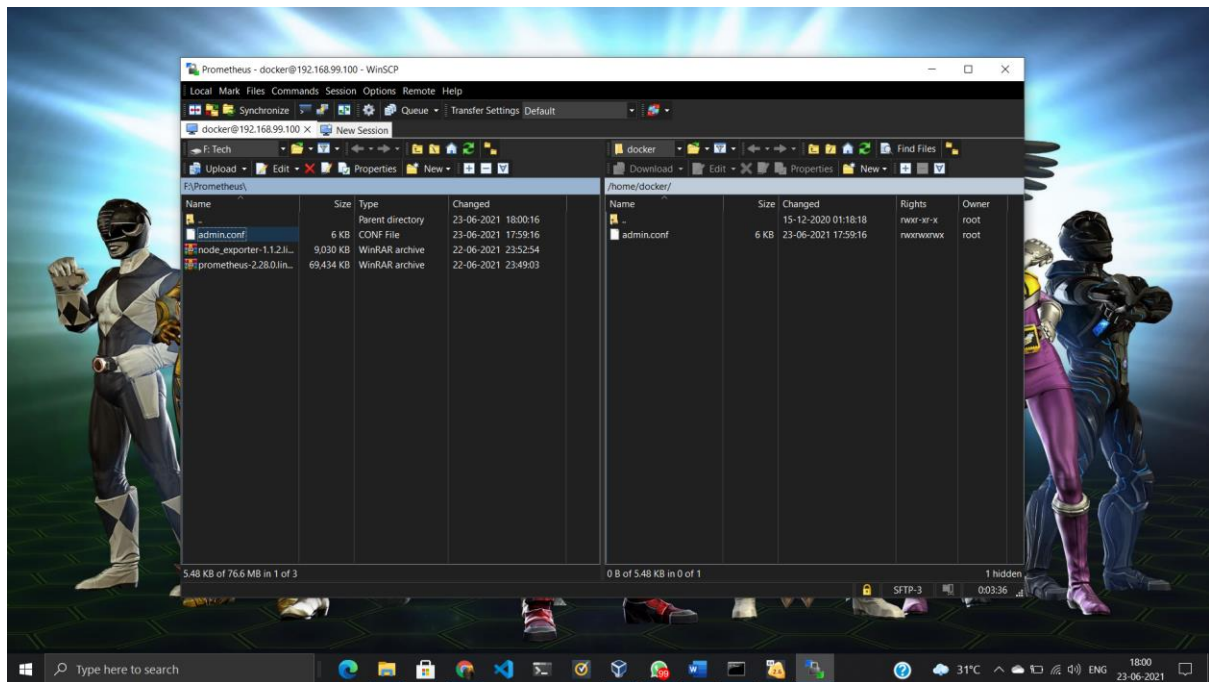
We change the permissions of admin.conf file so that we can copy this file from here to Windows Machine. And afterwards we need to transfer this file to RHEL8 VM.

```
minikube [Running] - Oracle VM VirtualBox
$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:2A:B5:97
          inet addr:192.168.99.100  Bcast:192.168.99.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:213 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28708 (28.0 KiB)  TX bytes:135849 (132.6 KiB)

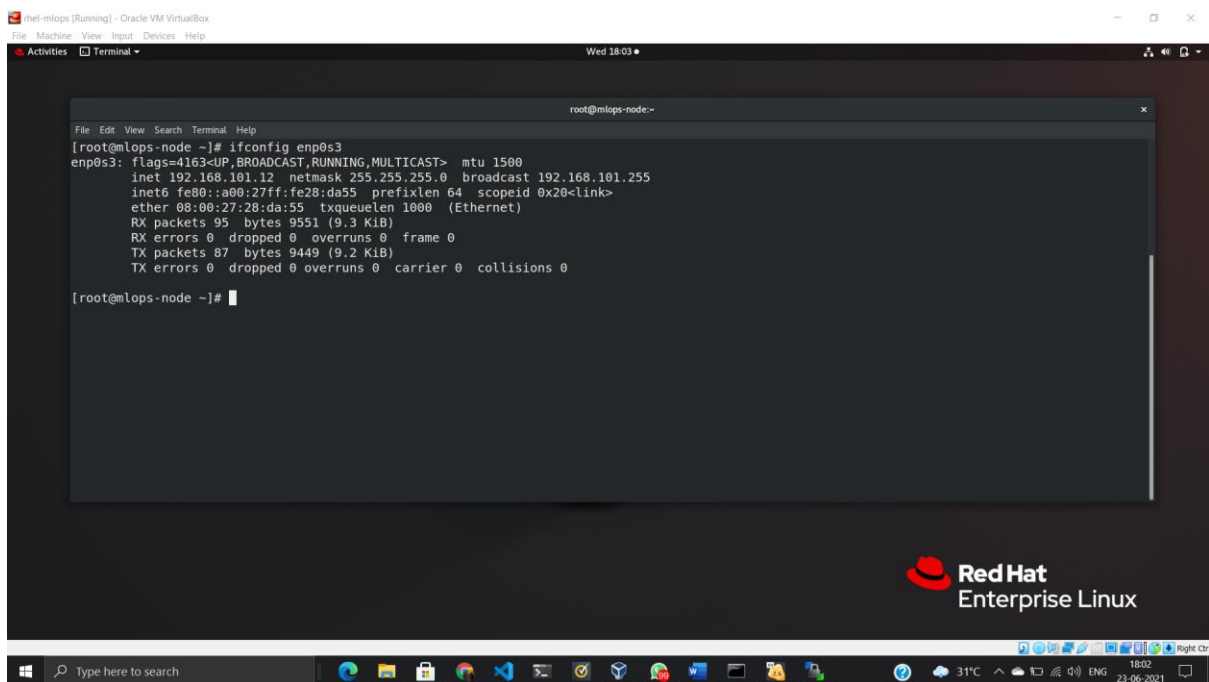
$ _
```

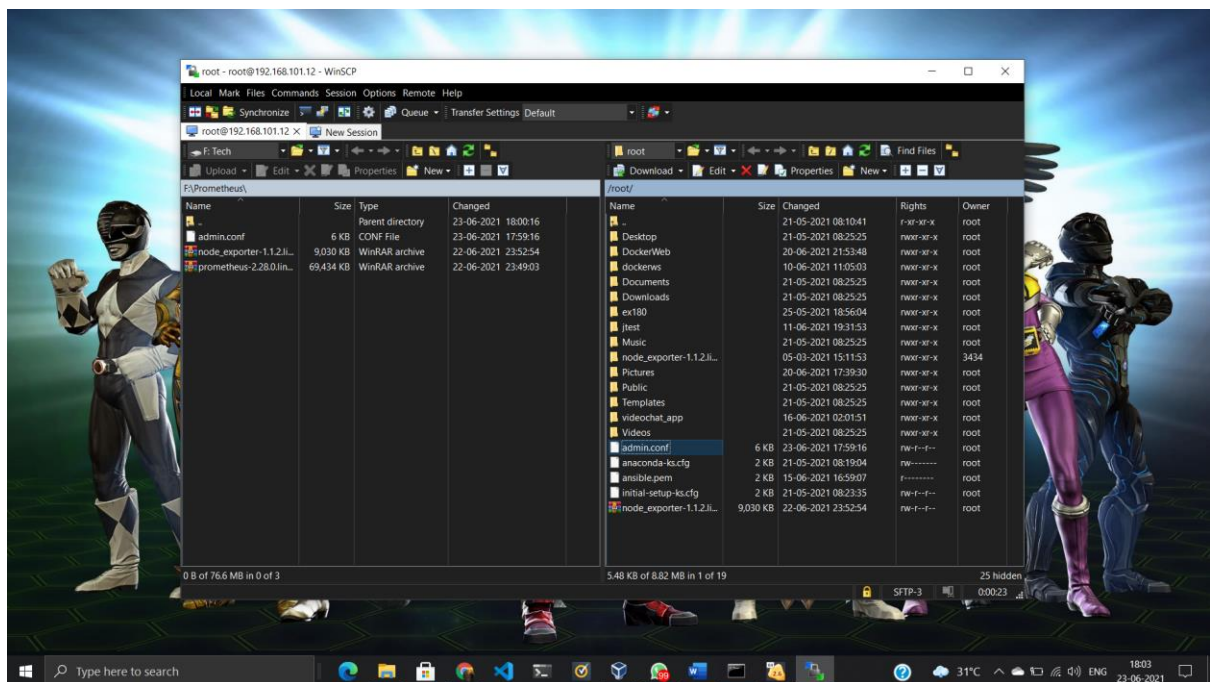
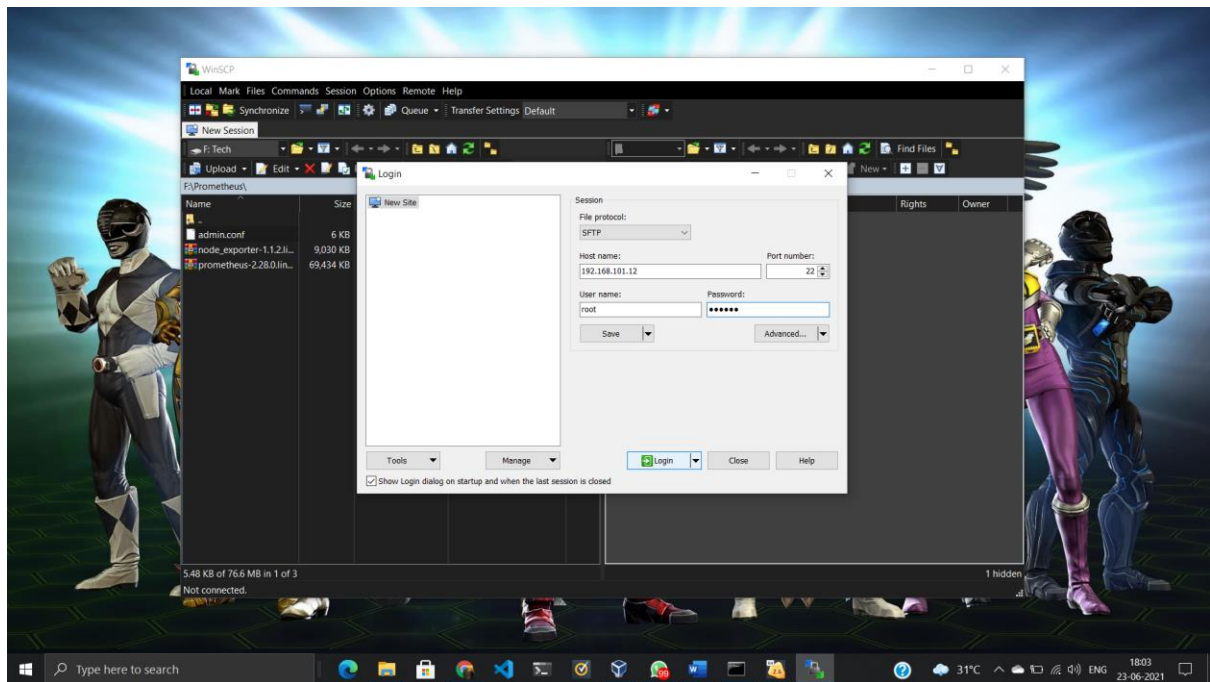
Using WinSCP we can transfer file from remote machines or VM's.

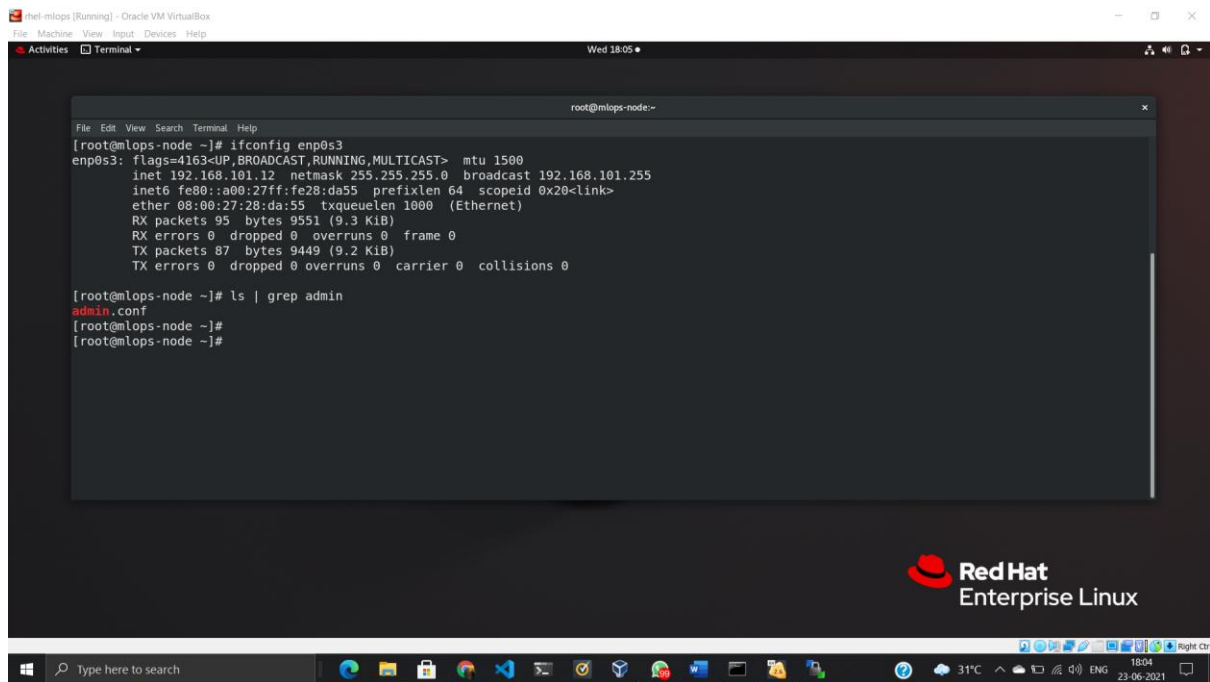




Launch the RHEL8 VM. Check the IP Address of system and again using WinSCP we can transfer admin.conf file from Windows to here i.e RHEL8 VM.







The screenshot shows a terminal window titled 'root@mlops-node:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal output is as follows:

```
[root@mlops-node ~]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.101.12  netmask 255.255.255.0  broadcast 192.168.101.255
    inet6 fe80::a00:27ff:fe28:da55  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:28:da:55  txqueuelen 1000  (Ethernet)
    RX packets 95  bytes 9551 (9.3 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 87  bytes 9449 (9.2 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

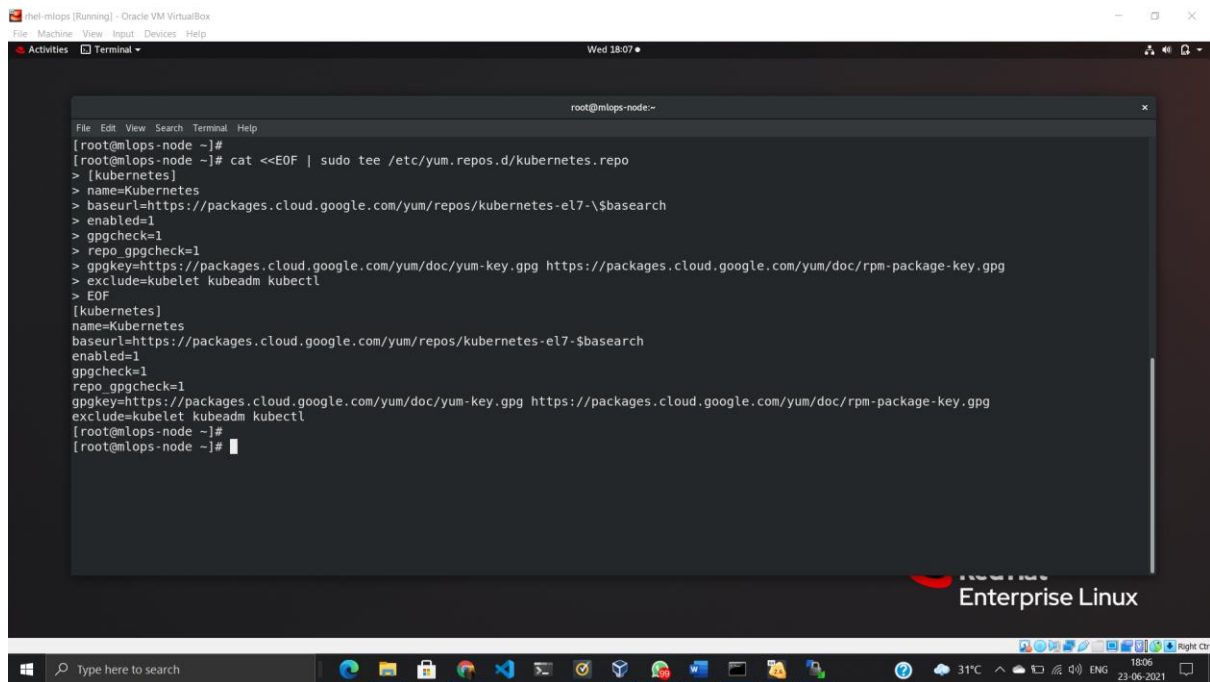
[root@mlops-node ~]# ls | grep admin
admin.conf
[root@mlops-node ~]#
[root@mlops-node ~]#
```

The terminal window is part of a Red Hat Enterprise Linux environment, as indicated by the logo in the bottom right corner. The desktop environment includes a taskbar with various application icons and a system tray showing the date and time (Wed 18:05).

Now, we need kubectl software which acts as a client for k8s cluster. Copy below code in terminal and it will create a Kubernetes repo from which we can install kubectl.

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

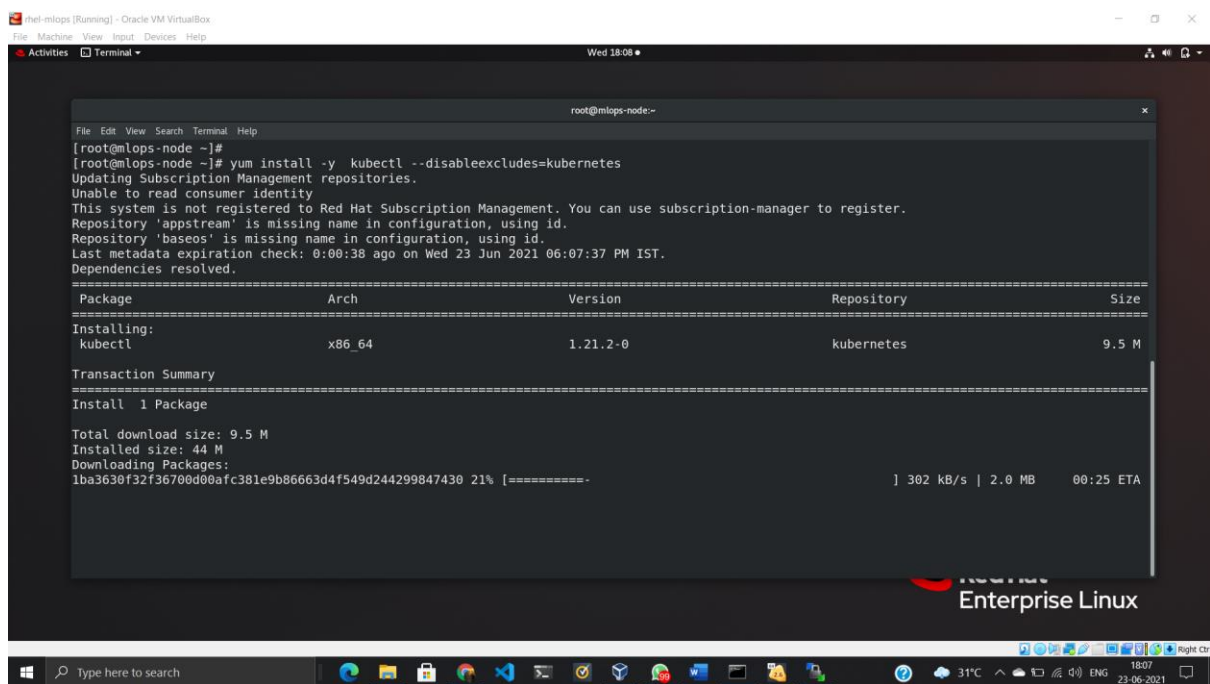
```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-`$basearch`
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
EOF
```

```
File Edit View Search Terminal Help
root@mlops-node:~
[root@mlops-node ~]#
[root@mlops-node ~]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
> enabled=1
> gpgcheck=1
> repo_gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> exclude=kubelet kubeadm kubectl
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
[root@mlops-node ~]#
[root@mlops-node ~]#
```

To install kubectl:

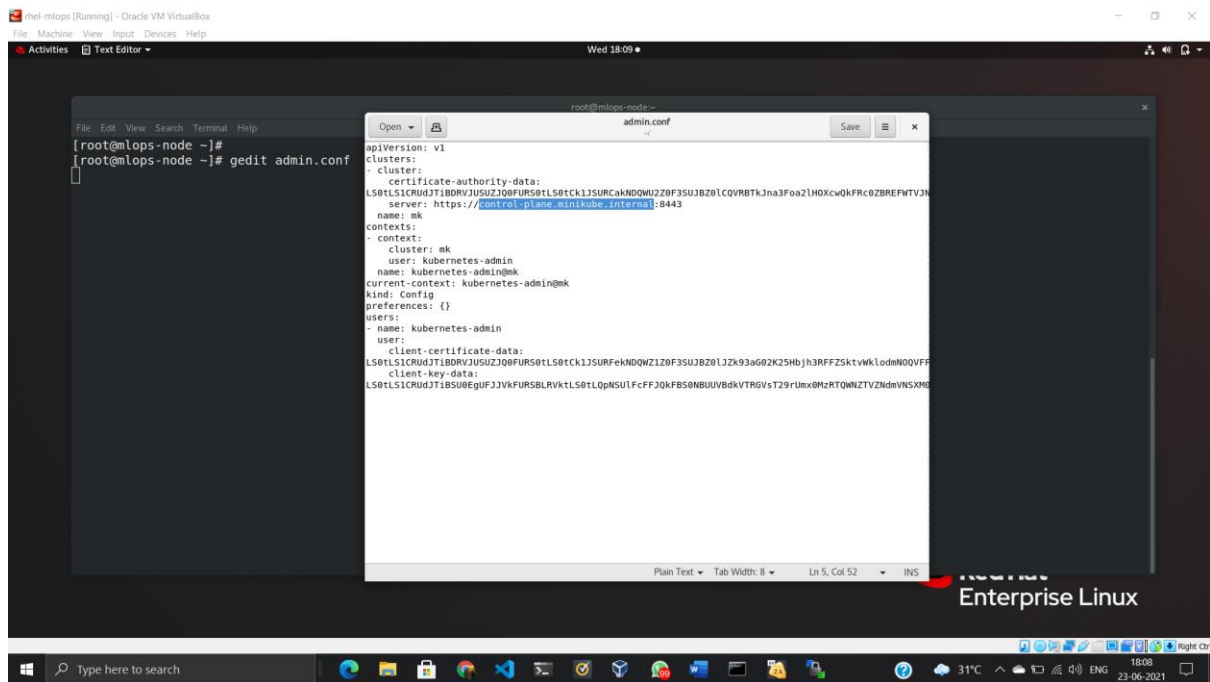
yum install kubectl



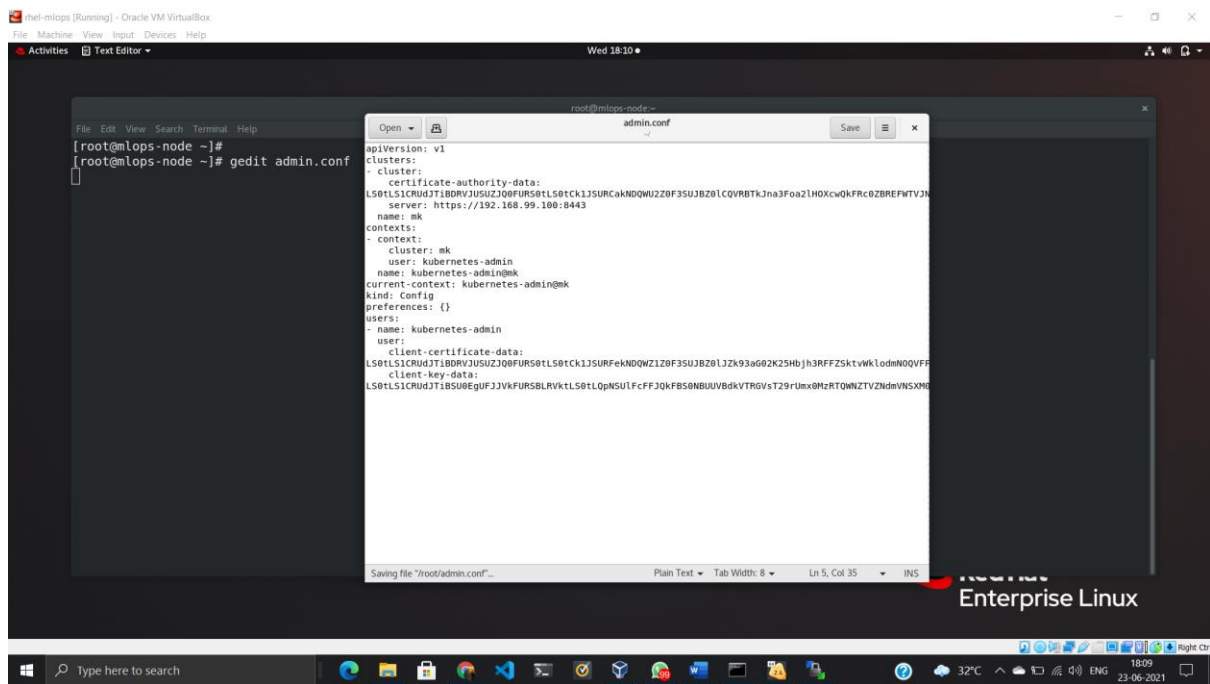
```
File Edit View Search Terminal Help
root@mlops-node:~
[root@mlops-node ~]#
[root@mlops-node ~]# yum install -y kubectl --disableexcludes=kubernetes
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Repository 'appstream' is missing name in configuration, using id.
Repository 'baseos' is missing name in configuration, using id.
Last metadata expiration check: 0:00:38 ago on Wed 23 Jun 2021 06:07:37 PM IST.
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
 kubectl x86_64 1.21.2-0 kubernetes 9.5 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 9.5 M
Installed size: 44 M
Downloading Packages:
1ba3630f32f3670d00a0fc381e9b86663d4f549d244299847430 21% [=====
] 302 kB/s | 2.0 MB 00:25 ETA
```

Now, we have to make a small change inside admin.conf file. By default it has the hostname written as the server. So, we need to replace it by the IP Address of the Minikube VM.

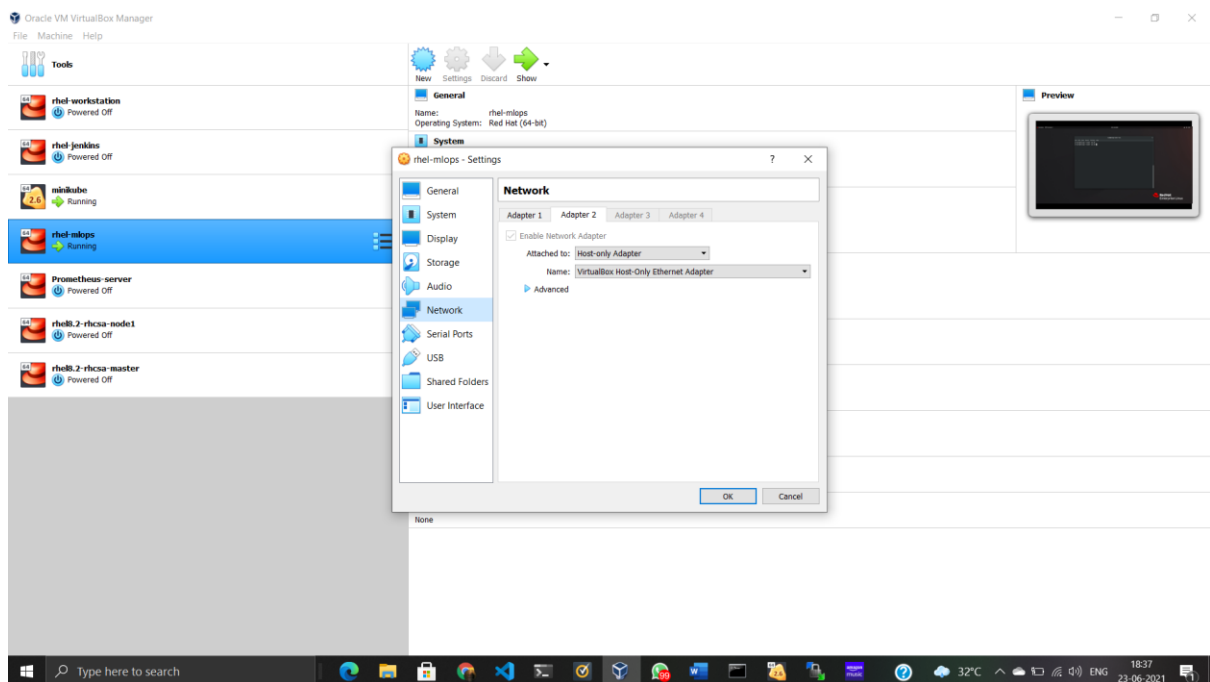
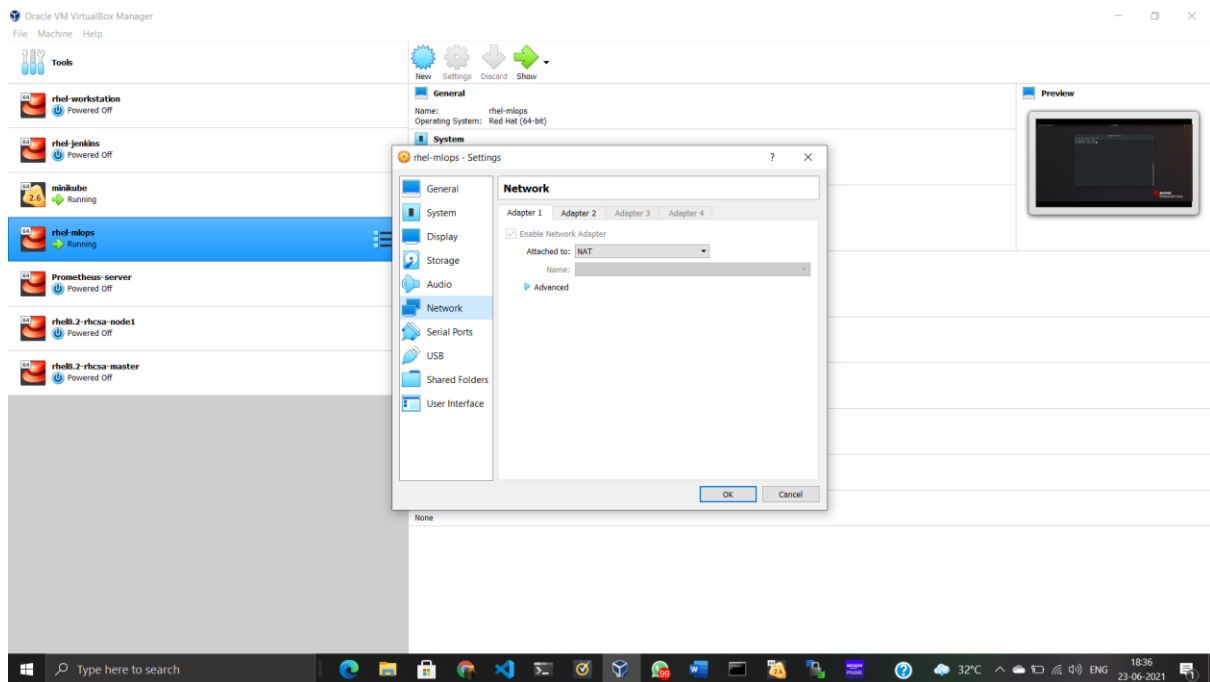


Put IP Address of Minikube VM.



Now, close the VM and change the Network Adapter of VM.

Put 1st Adapter as NAT and 2nd Adapter as Host Only



Start the VM. Create a working directory and put the admin.conf file in that directory.

And we can run kubectl commands by appending **--kubeconfig admin.conf** at last of the command.

Example:

kubectl create deployment web-test --image=httpd --kubeconfig admin.conf

rhel-mlops [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Wed 18:55

root@mlops-node:~/kubews

```
File Edit View Search Terminal Help
[root@mlops-node ~]#
[root@mlops-node ~]# mkdir kubews
[root@mlops-node ~]#
[root@mlops-node ~]# cp admin.conf kubews/
[root@mlops-node ~]#
[root@mlops-node ~]# cd kubews/
[root@mlops-node kubews]# ls
admin.conf
[root@mlops-node kubews]#
[root@mlops-node kubews]# kubectl create deployment web-test --image=httpd --kubeconfig admin.conf
deployment.apps/web-test created
[root@mlops-node kubews]#
[root@mlops-node kubews]# kubectl get pods --kubeconfig admin.conf
NAME                                READY   STATUS    RESTARTS   AGE
web-test-79c7d95567-wnv9x          1/1     Running   0           13s
[root@mlops-node kubews]#
[root@mlops-node kubews]# kubectl get deployment --kubeconfig admin.conf
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
web-test 1/1      1             1           25s
[root@mlops-node kubews]#
[root@mlops-node kubews]#
```

Type here to search

32°C 1853 23-06-2021