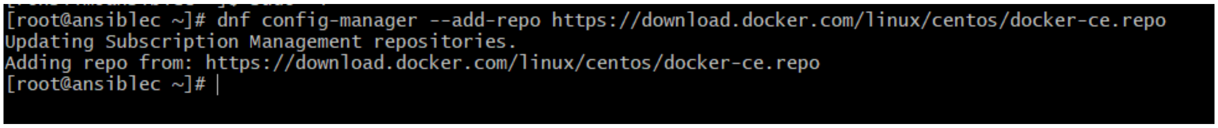
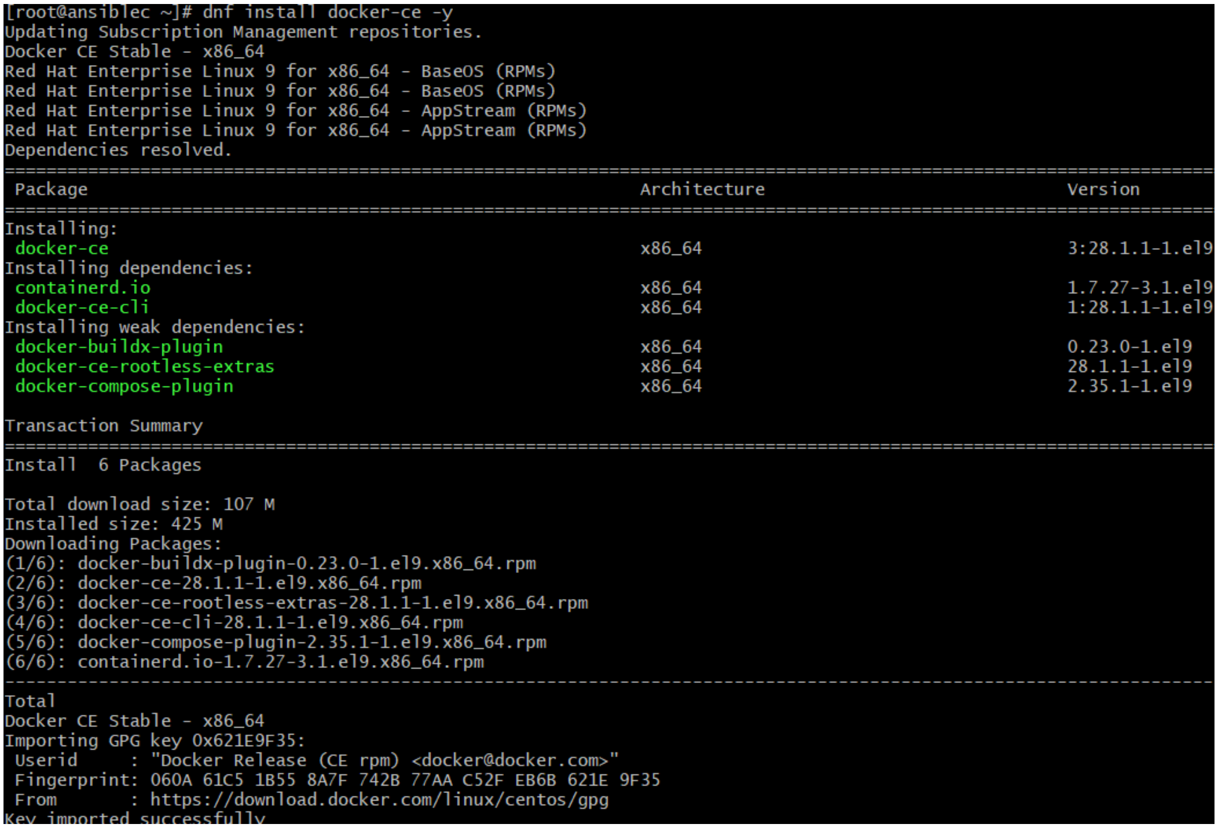
https://infotechys.com/install-minikube-on-rhel9-or-centos9/

```
dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```



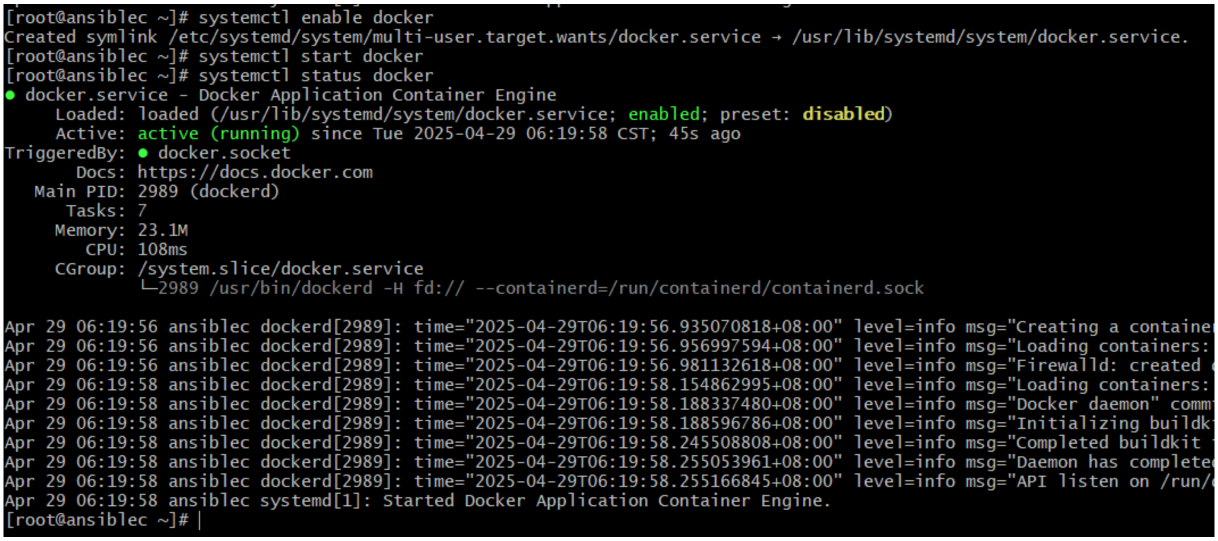
```
[root@ansiblec ~]# dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Updating Subscription Management repositories.
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
[root@ansiblec ~]# |
```

```
dnf install docker-ce
```



```
[root@ansiblec ~]# dnf install docker-ce -y
Updating Subscription Management repositories.
Docker CE Stable - x86_64
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Dependencies resolved.
==========================================================================================================================
 Package                                                 Architecture                              Version
==========================================================================================================================
Installing:
 docker-ce                                               x86_64                                    3:28.1.1-1.el9
Installing dependencies:
 containerd.io                                           x86_64                                    1.7.27-3.1.el9
 docker-ce-cli                                           x86_64                                    1:28.1.1-1.el9
Installing weak dependencies:
 docker-buildx-plugin                                    x86_64                                    0.23.0-1.el9
 docker-ce-rootless-extras                               x86_64                                    28.1.1-1.el9
 docker-compose-plugin                                   x86_64                                    2.35.1-1.el9

Transaction Summary
==========================================================================================================================
Install  6 Packages

Total download size: 107 M
Installed size: 425 M
Downloading Packages:
(1/6): docker-buildx-plugin-0.23.0-1.el9.x86_64.rpm
(2/6): docker-ce-28.1.1-1.el9.x86_64.rpm
(3/6): docker-ce-rootless-extras-28.1.1-1.el9.x86_64.rpm
(4/6): docker-ce-cli-28.1.1-1.el9.x86_64.rpm
(5/6): docker-compose-plugin-2.35.1-1.el9.x86_64.rpm
(6/6): containerd.io-1.7.27-3.1.el9.x86_64.rpm
--------------------------------------------------------------------------------------------------------------------------
Total
Docker CE Stable - x86_64
Importing GPG key 0x621E9F35:
 Userid     : "Docker Release (CE rpm) <docker@docker.com>"
 Fingerprint: 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35
 From       : https://download.docker.com/linux/centos/gpg
Key imported successfully
```

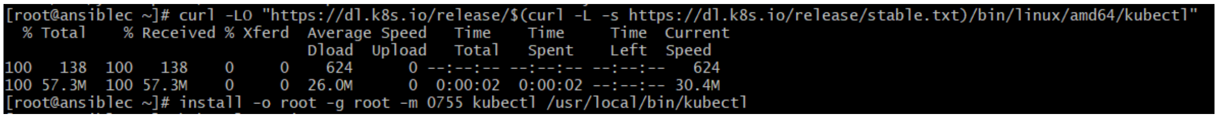
```
systemctl enable docker
systemctl start docker
Systemctl status docker
```



```
[root@ansiblec ~]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ansiblec ~]# systemctl start docker
[root@ansiblec ~]# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-04-29 06:19:58 CST; 45s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2989 (dockerd)
      Tasks: 7
     Memory: 23.1M
        CPU: 108ms
     CGroup: /system.slice/docker.service
             └─2989 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 29 06:19:56 ansiblec dockerd[2989]: time="2025-04-29T06:19:56.935070818+08:00" level=info msg="Creating a containe
Apr 29 06:19:56 ansiblec dockerd[2989]: time="2025-04-29T06:19:56.956997594+08:00" level=info msg="Loading containers:
Apr 29 06:19:56 ansiblec dockerd[2989]: time="2025-04-29T06:19:56.981132618+08:00" level=info msg="Firewalld: created
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.154862995+08:00" level=info msg="Loading containers:
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.188337480+08:00" level=info msg="Docker daemon" comm
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.188596786+08:00" level=info msg="Initializing buildk
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.245508808+08:00" level=info msg="Completed buildkit
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.255053961+08:00" level=info msg="Daemon has complete
Apr 29 06:19:58 ansiblec dockerd[2989]: time="2025-04-29T06:19:58.255166845+08:00" level=info msg="API listen on /run/
Apr 29 06:19:58 ansiblec systemd[1]: Started Docker Application Container Engine.
[root@ansiblec ~]# |
```

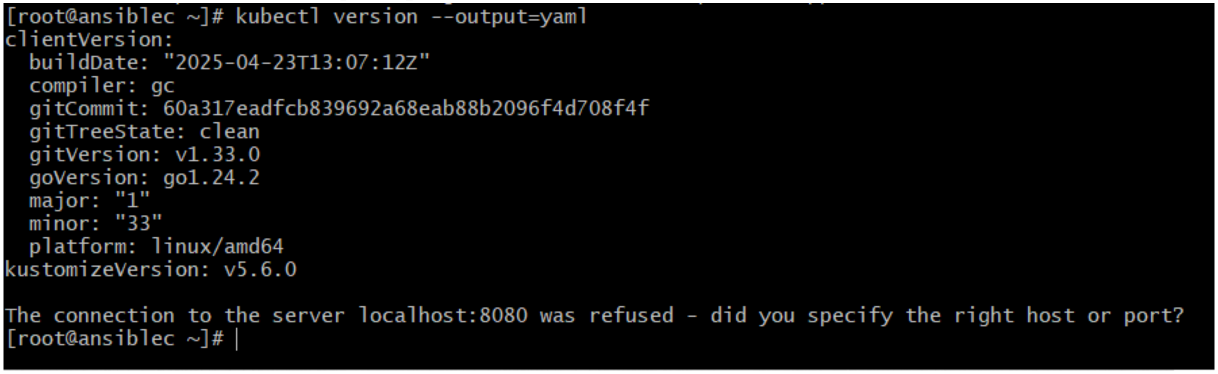
```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```



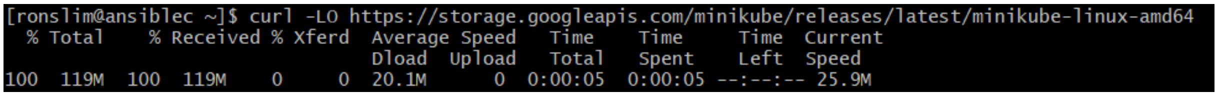
```
[root@ansiblec ~]# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    624      0 --:--:-- --:--:-- --:--:--   624
100 57.3M  100 57.3M    0     0  26.0M      0  0:00:02  0:00:02 --:--:-- 30.4M
[root@ansiblec ~]# install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

```
kubectl version --output=yaml
```



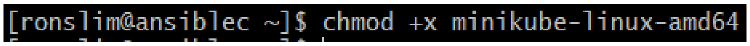
```
[root@ansiblec ~]# kubectl version --output=yaml
clientVersion:
  buildDate: "2025-04-23T13:07:12Z"
  compiler: gc
  gitCommit: 60a317eadfcb839692a68eab88b2096f4d708f4f
  gitTreeState: clean
  gitVersion: v1.33.0
  goVersion: go1.24.2
  major: "1"
  minor: "33"
  platform: linux/amd64
kustomizeVersion: v5.6.0

The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@ansiblec ~]# |
```

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```



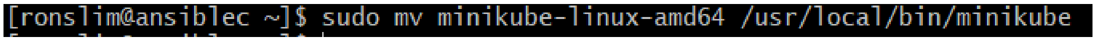
```
[ronslim@ansiblec ~]$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  119M  100  119M    0     0  20.1M      0  0:00:05  0:00:05 --:--:-- 25.9M
```

```
chmod +x minikube-linux-amd64
```



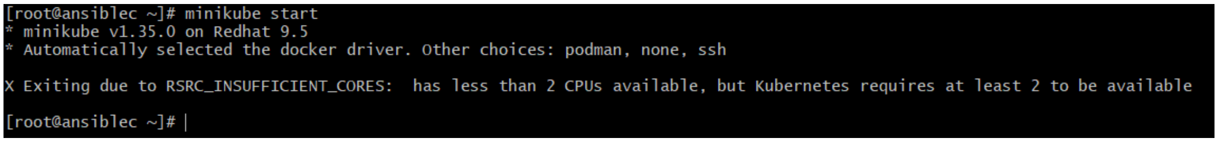
```
[ronslim@ansiblec ~]$ chmod +x minikube-linux-amd64
```

```
sudo mv minikube-linux-amd64 /usr/local/bin/minikube
```



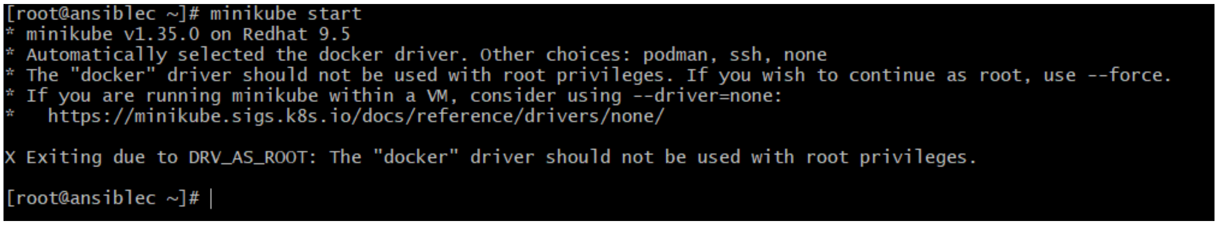
```
[ronslim@ansiblec ~]$ sudo mv minikube-linux-amd64 /usr/local/bin/minikube
```

Minikube start (Need 2 CPU)



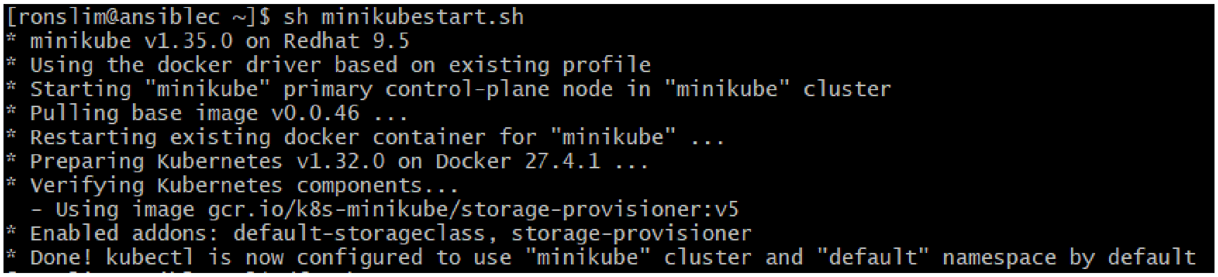
```
[root@ansiblec ~]# minikube start
* minikube v1.35.0 on Redhat 9.5
* Automatically selected the docker driver. Other choices: podman, none, ssh

X Exiting due to RSRC_INSUFFICIENT_CORES:  has less than 2 CPUs available, but Kubernetes requires at least 2 to be available

[root@ansiblec ~]# |
```

Minikube start (root acces not allowed)



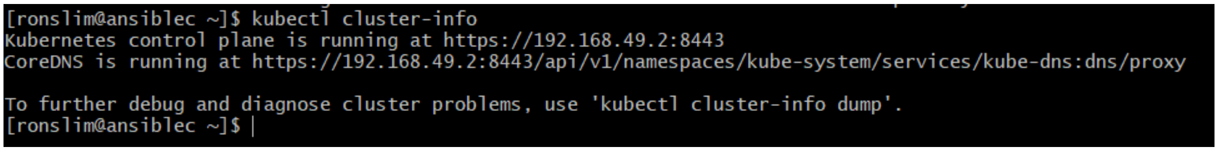
```
[root@ansiblec ~]# minikube start
* minikube v1.35.0 on Redhat 9.5
* Automatically selected the docker driver. Other choices: podman, ssh, none
* The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
* If you are running minikube within a VM, consider using --driver=none:
*   https://minikube.sigs.k8s.io/docs/reference/drivers/none/

X Exiting due to DRV_AS_ROOT: The "docker" driver should not be used with root privileges.

[root@ansiblec ~]# |
```

Minikube start (non-root)



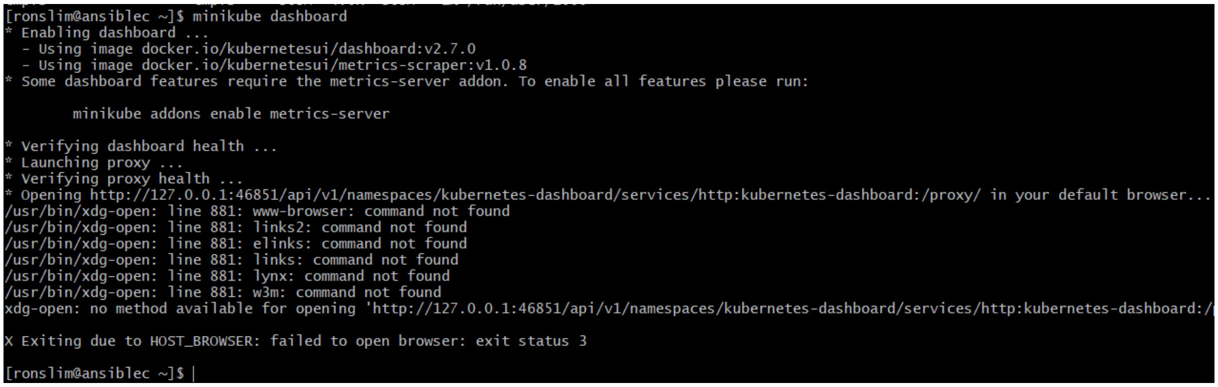
```
[ronslim@ansiblec ~]$ sh minikubestart.sh
* minikube v1.35.0 on Redhat 9.5
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Restarting existing docker container for "minikube" ...
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

kubectl cluster-info



```
[ronslim@ansiblec ~]$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[ronslim@ansiblec ~]$ |
```

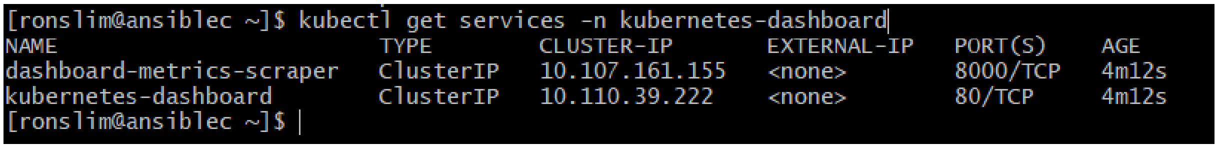
Minikube dashboard



```
[ronslim@ansiblec ~]$ minikube dashboard
* Enabling dashboard ...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:46851/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
/usr/bin/xdg-open: line 881: www-browser: command not found
/usr/bin/xdg-open: line 881: links2: command not found
/usr/bin/xdg-open: line 881: elinks: command not found
/usr/bin/xdg-open: line 881: links: command not found
/usr/bin/xdg-open: line 881: lynx: command not found
/usr/bin/xdg-open: line 881: w3m: command not found
xdg-open: no method available for opening 'http://127.0.0.1:46851/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/

X Exiting due to HOST_BROWSER: failed to open browser: exit status 3

[ronslim@ansiblec ~]$ |
```

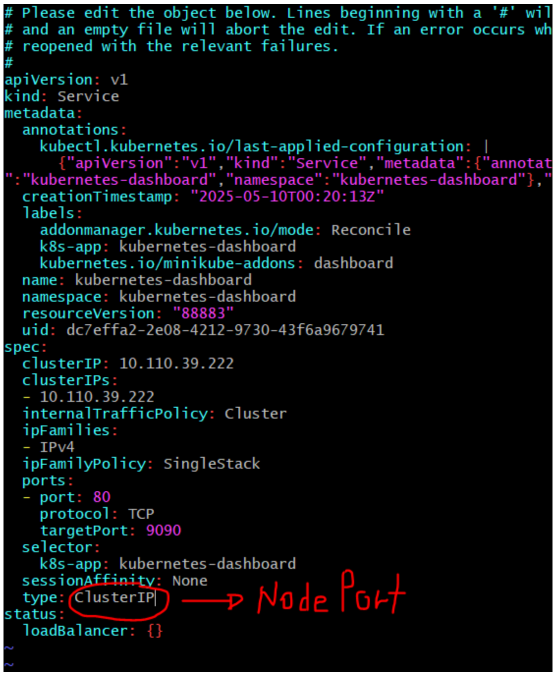
Accessing Dashboard outside browser (Reference = How to access/expose kubernetes-dashboard service outside of a cluster? - Stack Overflow):

Get the dashboard service
kubectl get services -n kubernetes-dashboard



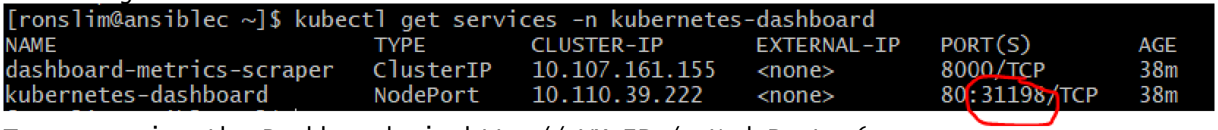
```
[ronslim@ansiblec ~]$ kubectl get services -n kubernetes-dashboard
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
dashboard-metrics-scraper   ClusterIP   10.107.161.155   <none>        8000/TCP   4m12s
kubernetes-dashboard        ClusterIP   10.110.39.222    <none>        80/TCP     4m12s
[ronslim@ansiblec ~]$ |
```

Edit Dashboard service
kubectl edit service kubernetes-dashboard -n kubernetes-dashboard and change type from ClusterIP to NodePort and save changes



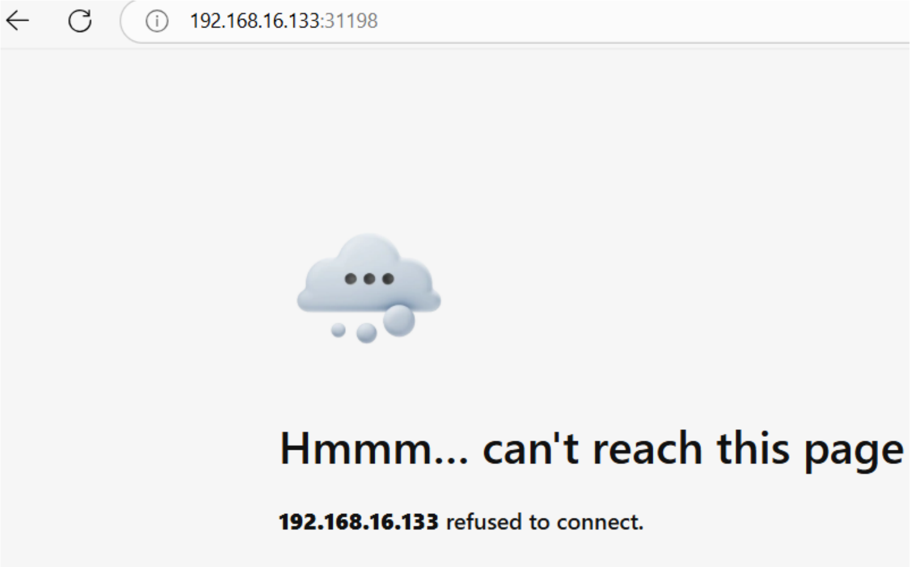
```
# Please edit the object below. Lines beginning with a '#' wil
# and an empty file will abort the edit. If an error occurs wh
# reopened with the relevant failures.
#
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotat
":"kubernetes-dashboard","namespace":"kubernetes-dashboard"},"
  creationTimestamp: "2025-05-10T00:20:13Z"
  labels:
    addonmanager.kubernetes.io/mode: Reconcile
    k8s-app: kubernetes-dashboard
    kubernetes.io/minikube-addons: dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
  resourceVersion: "88883"
  uid: dc7effa2-2e08-4212-9730-43f6a9679741
spec:
  clusterIP: 10.110.39.222
  clusterIPs:
  - 10.110.39.222
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 80
    protocol: TCP
    targetPort: 9090
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
~
~
```

Get again the dashboard service and get Node Port
kubectl get services -n kubernetes-dashboard



```
[ronslim@ansiblec ~]$ kubectl get services -n kubernetes-dashboard
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
dashboard-metrics-scraper   ClusterIP   10.107.161.155   <none>        8000/TCP       38m
kubernetes-dashboard        NodePort    10.110.39.222    <none>        80:31198/TCP   38m
```

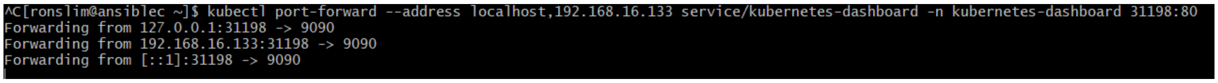
Try accessing the Dashboard via http://<VM IP>/:<NodePort> (e.g. http://192.168.16.133:31198). Notice that the Grafana is not yet accessible outside VM browser.


192.168.16.133:31198

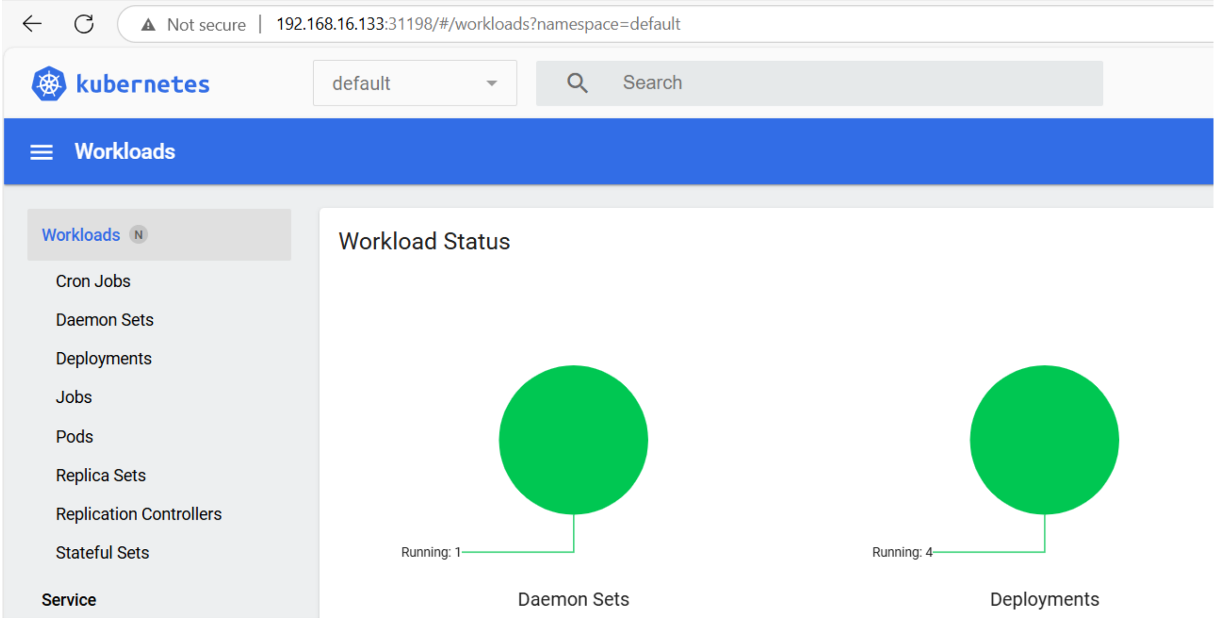
Hmmm... can't reach this page

192.168.16.133 refused to connect.

Then port-forward to access the site in the new terminal window.
Kubectl port-forward –address localhost,<VM IP> service/<grafana service> <NodePort>:80
e.g. kubectl port-forward --address localhost,192.168.16.133 service/grafana-np 31198:80


```
^C[ronslim@ansiblec ~]$ kubectl port-forward --address localhost,192.168.16.133 service/kubernetes-dashboard -n kubernetes-dashboard 31198:80
Forwarding from 127.0.0.1:31198 -> 9090
Forwarding from 192.168.16.133:31198 -> 9090
Forwarding from [::1]:31198 -> 9090
```

Try accessing the Dashboard again via http://<VM IP>/:<NodePort> (e.g. http://192.168.16.133:31198). Notice that the Dashboard is now accessible outside VM Browser.


⚠ Not secure | 192.168.16.133:31198/#/workloads?namespace=default