

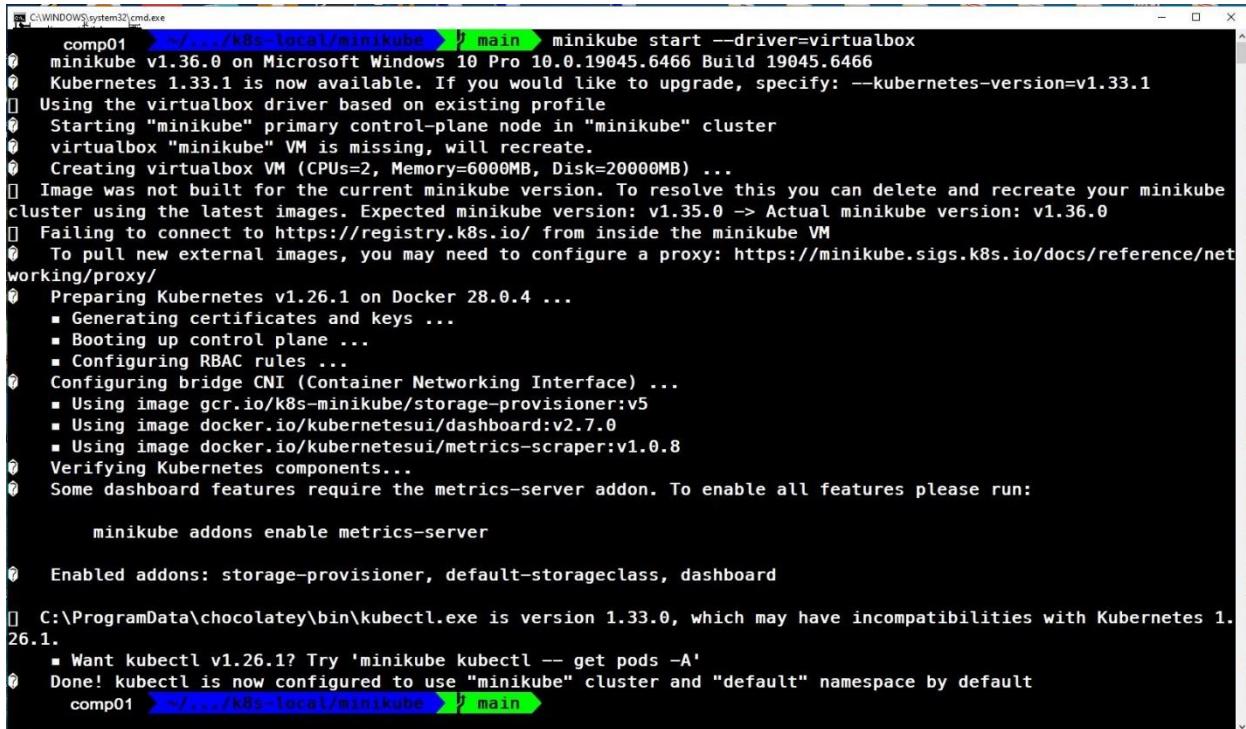
Start minikube on Windows – Virtual Box / HyperV environment

Run as administrator cmd.exe or PowerShell.

Virtual Box environment

To start single node minikube cluster, inside opened terminal run:

```
minikube start --driver=virtualbox
```

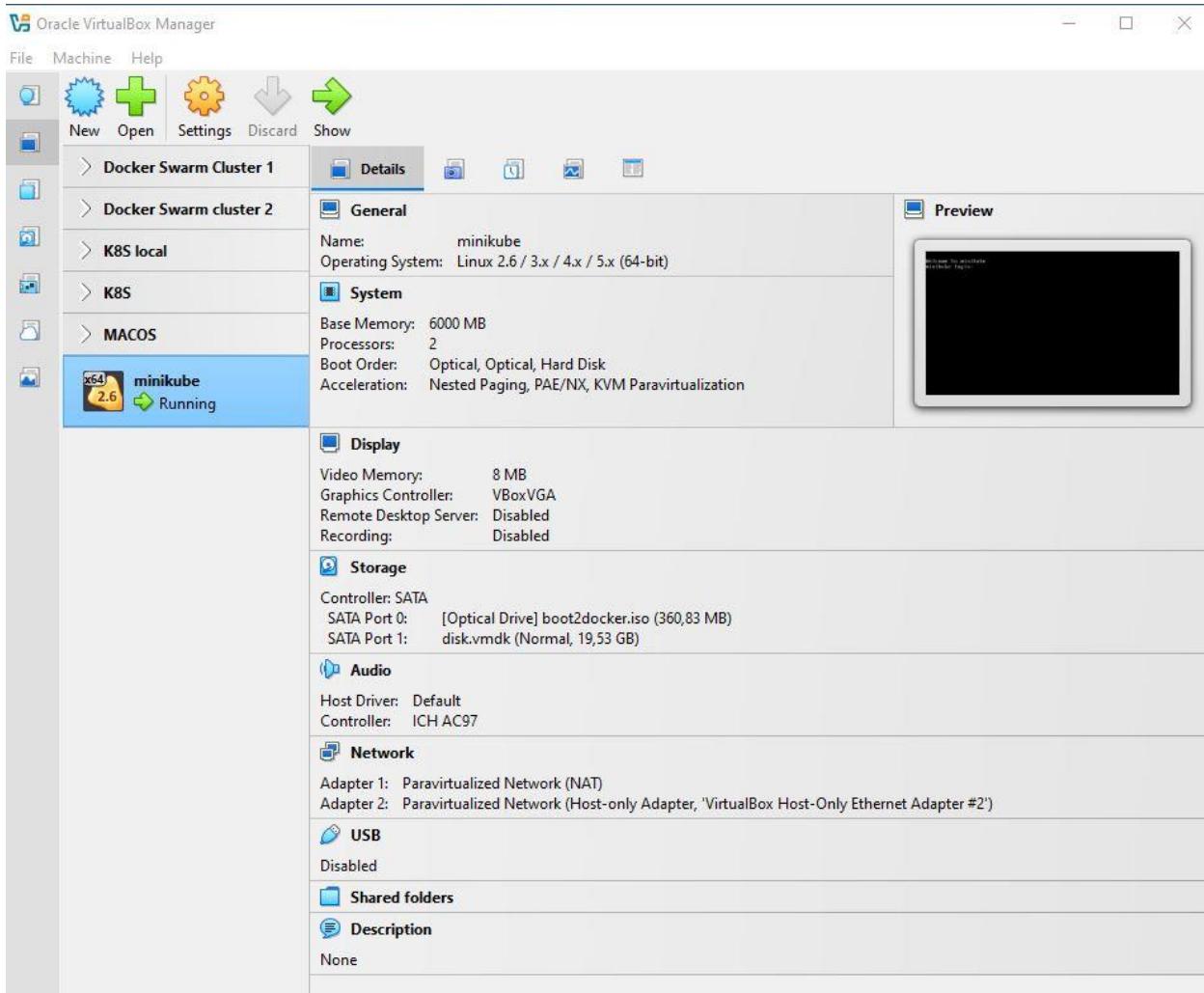


```
comp01 > cd C:\k8s-local\minikube > main > minikube start --driver=virtualbox
minikube v1.36.0 on Microsoft Windows 10 Pro 10.0.19045.6466 Build 19045.6466
Kubernetes 1.33.1 is now available. If you would like to upgrade, specify: --kubernetes-version=v1.33.1
Using the virtualbox driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
virtualbox "minikube" VM is missing, will recreate.
Creating virtualbox VM (CPUs=2, Memory=6000MB, Disk=20000MB) ...
Image was not built for the current minikube version. To resolve this you can delete and recreate your minikube cluster using the latest images. Expected minikube version: v1.35.0 -> Actual minikube version: v1.36.0
Failing to connect to https://registry.k8s.io/ from inside the minikube VM
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.26.1 on Docker 28.0.4 ...
  Generating certificates and keys ...
  Booting up control plane ...
  Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
  Using image docker.io/kubernetesui/dashboard:v2.7.0
  Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
Verifying Kubernetes components...
Some dashboard features require the metrics-server addon. To enable all features please run:

  minikube addons enable metrics-server

Enabled addons: storage-provisioner, default-storageclass, dashboard

C:\ProgramData\chocolatey\bin\kubectl.exe is version 1.33.0, which may have incompatibilities with Kubernetes 1.26.1.
  Want kubectl v1.26.1? Try 'minikube kubectl -- get pods -A'
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
comp01 > cd C:\k8s-local\minikube > main
```



To test if minikube working fine:

`kubectl get namespace`

or

`kubectl get ns`

```
comp01 ~/.kube/minikube % main ➔ kubectl get namespace
NAME          STATUS   AGE
default       Active   2m7s
kube-node-lease Active   2m16s
kube-public    Active   2m16s
kube-system   Active   2m16s
kubernetes-dashboard Active  2m
```

To access Kubernetes console, inside terminal type:

`minikube dashboard`

this will open inside you default browser Kubernetes default console:

Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	CPU capacity (cores)	Memory requests (bytes)	Memory limits (bytes)	Memory capacity (bytes)	Pods	Created
minikube	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	850.00m (42.50%)	0.00m (0.00%)	2.00	240.00Mi (4.15%)	340.00Mi (5.88%)	5.65Gi	10 (9.09%)	7 minutes ago

Create a test app (Deployment)

Let's deploy the classic **nginx** container.

Open second terminal and type:

```
kubectl create deployment test-nginx --image=nginx
```

What this does:

- Creates a **Deployment**
- With **1 Pod**
- Using the official nginx image

Verify from terminal:

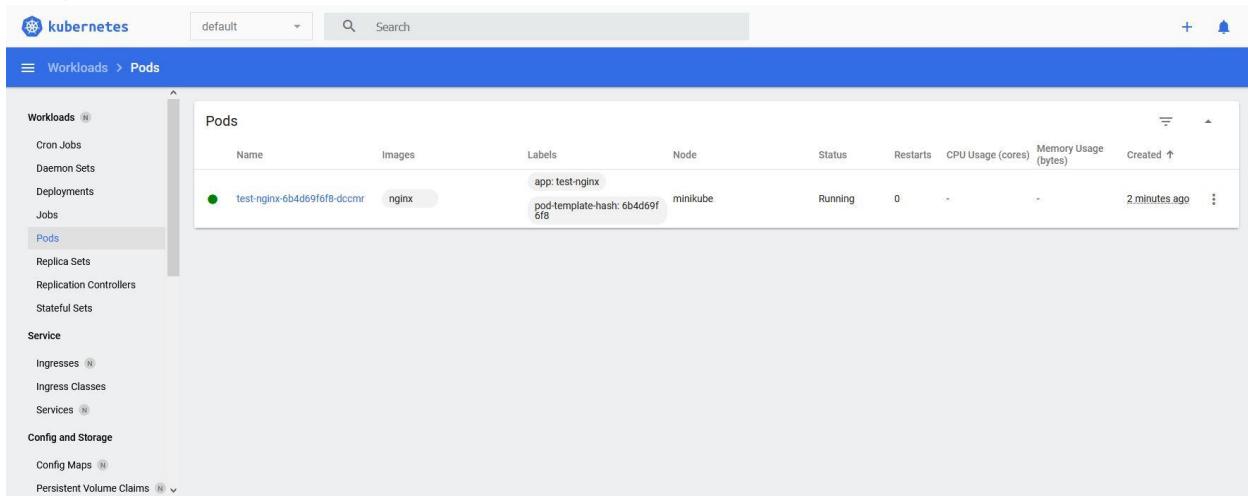
```
kubectl get deployments
```

```
kubectl get pods
```

You should see a Pod in Running state.

```
C:\WINDOWS\system32\cmd.exe
comp01 ~/.../k8s-local/minikube ➔ main ➔ kubectl create deployment test-nginx --image=nginx
deployment.apps/test-nginx created
comp01 ~/.../k8s-local/minikube ➔ main ➔ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
test-nginx 1/1     1           1           55s
comp01 ~/.../k8s-local/minikube ➔ main ➔ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
test-nginx-6b4d69f6f8-dccmr 1/1     Running   0          64s
comp01 ~/.../k8s-local/minikube ➔ main ➔
```

Verify from kubernetes Web UI



Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
test-nginx-6b4d696f8-dccmr	nginx	app: test-nginx pod-template-hash: 6b4d69f6f8	minikube	Running	0	-	-	2 minutes ago

2 Expose the app (Service)

Expose it inside the cluster:

```
kubectl expose deployment test-nginx \
--type=ClusterIP \
--port=80
kubectl expose deployment test-nginx --type=NodePort --port=80
service/test-nginx exposed
kubectl get svc
```

```
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP      29m
test-nginx  NodePort  10.100.219.92  <none>        80:31249/TCP  11s
minikube ip
192.168.59.106
```

Verify:

```
kubectl get svc
```

At this point:

- The app is reachable **inside the cluster**
- Not yet accessible from your browser

```
■ W      comp01 ~/.k8s-local/minikube ➔ main ➔ kubectl expose deployment test-nginx \
Don>   --type=ClusterIP \
vladan>   --port=80
NAME      service/test-nginx exposed
default   comp01 ~/.k8s-local/minikube ➔ main ➔ kubectl get svc
kube-noNAME  TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kube-pukubernetes  ClusterIP  10.96.0.1    <none>        443/TCP      14m
kube-sytest-nginx  ClusterIP  10.105.43.29  <none>        80/TCP      12s
kuberne   comp01 ~/.k8s-local/minikube ➔ main ➔
```

2 Use the NodePort

From your previous kubectl get svc output:

```
test-nginx  NodePort  10.100.219.92 <none>  80:31249/TCP
```

- NodePort = 31249
- ClusterIP = internal only (ignore)

So the URL to open in your browser is:

http://<minikube-ip>:<node-port>

With the example above:

http://192.168.59.106:31249

Open that in your browser — you should see **the nginx default page**.



4 Scale the app (still imperative)

```
kubectl scale deployment test-nginx --replicas=3  
deployment.apps/test-nginx scaled
```

Verify:

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
test-nginx-6b4d69f6f8-cv7x6	1/1	Running	0	30s
test-nginx-6b4d69f6f8-dccmr	1/1	Running	0	29m
test-nginx-6b4d69f6f8-xbvjr	1/1	Running	0	30s

You'll now see **3 Pods**.

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
test-nginx-6b4d69f6f8-cv7x6	nginx	app: test-nginx pod-template-hash: 6b4d69f6f8	minikube	Running	0	-	-	a minute ago
test-nginx-6b4d69f6f8-xbvjr	nginx	app: test-nginx pod-template-hash: 6b4d69f6f8	minikube	Running	0	-	-	a minute ago
test-nginx-6b4d69f6f8-dccmr	nginx	app: test-nginx pod-template-hash: 6b4d69f6f8	minikube	Running	0	-	-	29 minutes ago

5 Clean up (important habit)

```
kubectl delete deployment test-nginx  
kubectl delete svc test-nginx
```

🧠 Mental model recap

Imperative commands:

- Are **great for learning and testing**
- Are **not reproducible**
- Should not be used for production

Think of them as:

“Typing directly into the cluster”

⚡ One-command quick smoke test

If you want the shortest possible test:

```
kubectl create deployment hello --image=nginx && \  
kubectl expose deployment hello --type=NodePort --port=80 && \  
minikube service hello
```

To stop minikube:

```
minikube stop
```

```
* Stopping node "minikube" ...
```

```
* 1 node stopped.
```

To delete minikube:

```
minikube delete
```

To start minikube 3 node cluster:

```
minikube start --nodes 3 -p multimode --driver=virtualbox --profile=mk-vb-3nodes
```

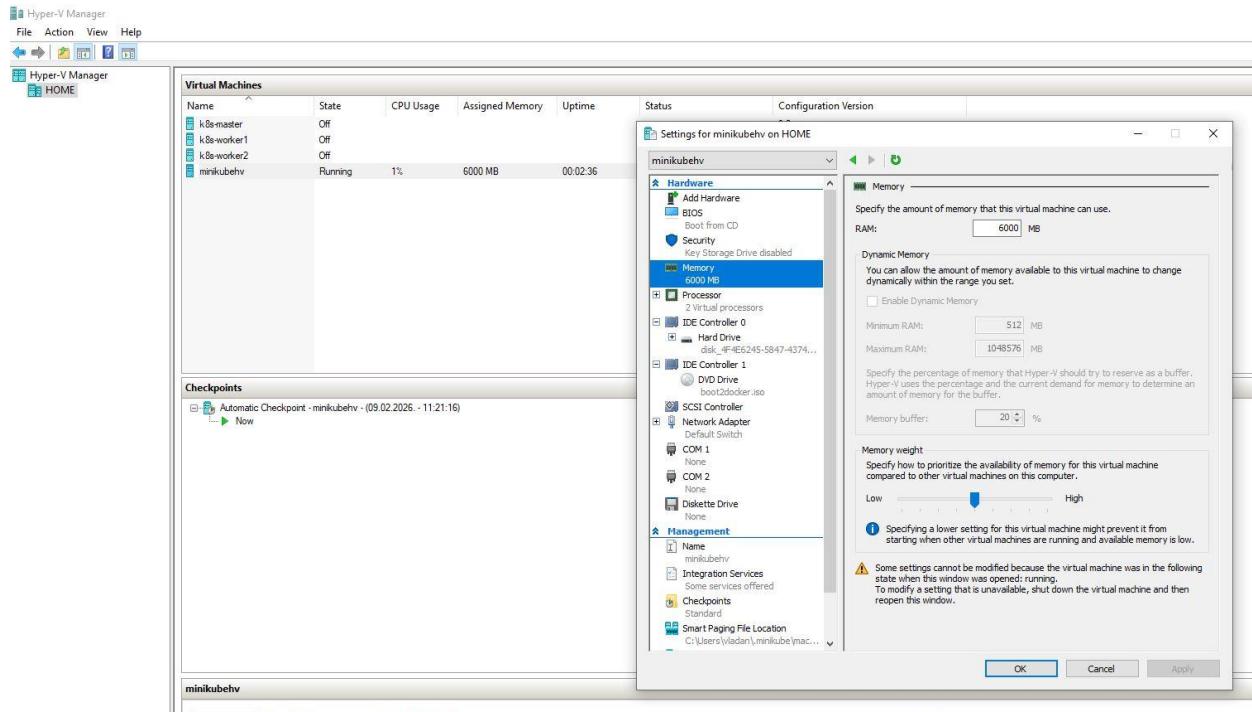
```
C:\Users\user01\Desktop\Linkedin\minikube>minikube start --driver=hyperv --profile=minikubehv
* [minikubehv] minikube v1.36.0 on Microsoft Windows 10 Pro 10.0.19045.6466 Build 19045.6466
  Using the hyperv driver based on user configuration
Starting "minikubehv" primary control-plane node in "minikubehv" cluster
Creating hyperv VM (CPUs=2, Memory=6000MB, Disk=20000MB) ...
! Image was not built for the current minikube version. To resolve this you can delete and recreate your minikube cluster using the latest images. Expected minikube version: v1.35.0 -> Actual minikube version: v1.36.0
* Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.33.1 on Docker 28.0.4 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikubehv" cluster and "default" namespace by default
```

Hyper-V environment

Run cmd.exe or PowerShell as Administrator:

```
minikube start --driver=hyperv --profile=minikubehv
```

```
C:\Users\user01\Desktop\Linkedin\minikube>minikube start --driver=hyperv --profile=minikubehv
* [minikubehv] minikube v1.36.0 on Microsoft Windows 10 Pro 10.0.19045.6466 Build 19045.6466
  Using the hyperv driver based on user configuration
Starting "minikubehv" primary control-plane node in "minikubehv" cluster
Creating hyperv VM (CPUs=2, Memory=6000MB, Disk=20000MB) ...
! Image was not built for the current minikube version. To resolve this you can delete and recreate your minikube cluster using the latest images. Expected minikube version: v1.35.0 -> Actual minikube version: v1.36.0
* Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.33.1 on Docker 28.0.4 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikubehv" cluster and "default" namespace by default
```

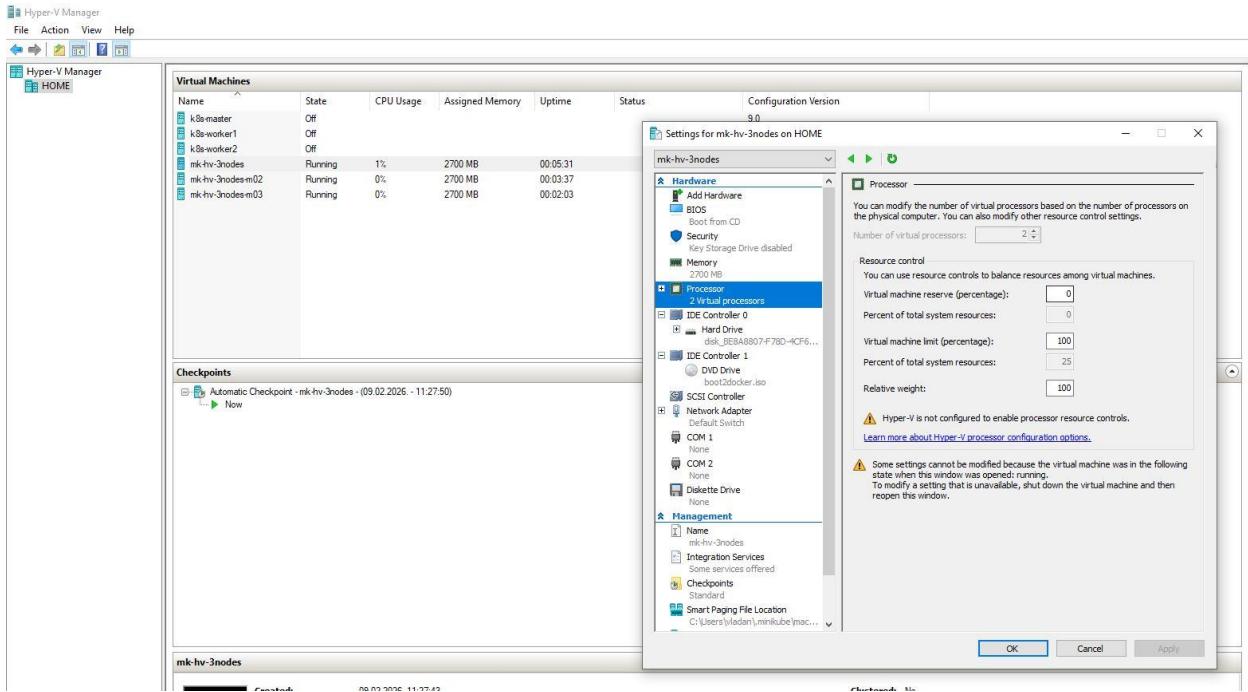


To delete:

```
C:\Users\user01\Desktop\Linkedin\minikube>minikube delete --profile=minikubehv
* Stopping node "minikubehv" ...
* Powering off "minikubehv" via SSH ...
* Deleting "minikubehv" in hyperv ...
* Removed all traces of the "minikubehv" cluster.
```

To start minikube 3 node cluster:

```
minikube start --nodes 3 -p multimode --driver=hyperv --profile=mk-hv-3nodes
```



To open cubernetes console:

```
minikube dashboard -p mk-hv-3nodes
```

```
C:\Users\user01\Desktop\LinkedIn\minikube>minikube dashboard -p mk-hv-3nodes
* Enabling dashboard ...
- Using image docker.io/kubernetessui/metrics-scraper:v1.0.8
- Using image docker.io/kubernetessui/dashboard:v2.7.0
* Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube -p mk-hv-3nodes addons enable metrics-server

* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:52448/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```

Kubernetes Web UI

<http://127.0.0.1:52448/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/node?namespace=default>

Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	CPU capacity (cores)	Memory requests (bytes)	Memory limits (bytes)	Memory capacity (bytes)	Pods	Created
mk-hv-3nodes-m03	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	100.00m (5.00%)	100.00m (5.00%)	2.00	50.00Mi (1.92%)	50.00Mi (1.92%)	2.54Gi	3 (2.73%)	6 minutes ago
mk-hv-3nodes-m02	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	100.00m (5.00%)	100.00m (5.00%)	2.00	50.00Mi (1.92%)	50.00Mi (1.92%)	2.54Gi	3 (2.73%)	8 minutes ago
mk-hv-3nodes	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	850.00m (42.50%)	100.00m (5.00%)	2.00	220.00Mi (8.44%)	220.00Mi (8.44%)	2.54Gi	8 (7.27%)	10 minutes ago

`kubectl get nodes -o wide`

```

C:\Users\user01\Desktop\Linkedin\minikube>kubectl get nodes -o wide
NAME      STATUS   ROLES      AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE     KERNEL-VERSION   CONTAINER-RUNTIME
mk-hv-3nodes   Ready   control-plane   12m    v1.33.1   172.30.111.185 <none>        Buildroot   2025.02       5.10.207      docker://28.0.4
mk-hv-3nodes-m02   Ready   <none>      11m    v1.33.1   172.30.103.106 <none>        Buildroot   2025.02       5.10.207      docker://28.0.4
mk-hv-3nodes-m03   Ready   <none>      9m44s   v1.33.1   172.30.103.218  <none>        Buildroot   2025.02       5.10.207      docker://28.0.4

C:\Users\user01\Desktop\Linkedin\minikube>

```

To delete:

`minikube delete --profile=mk-hv-3nodes`

1 Stop the cluster (if running)

`minikube stop -p mk-hv-3nodes`

- Stops all nodes in the cluster
- Safe first step before deletion

2 Delete the Minikube cluster

`minikube delete -p mk-hv-3nodes`

- Deletes **all VMs** created for that profile:

- mk-hv-3nodes (control-plane)

- mk-hv-3nodes-m02 (worker)
 - mk-hv-3nodes-m03 (worker)
-
- Removes **Minikube state** (cluster metadata, configs) for that profile
-

3 Verify deletion

minikube profile list

- The mk-hv-3nodes profile should no longer exist
 - If there's still a profile listed, you can remove it manually:
-

minikube delete -p <profile-name>

4 Remove leftover Hyper-V VMs manually (optional)

Sometimes minikube delete **doesn't remove VMs** completely due to Hyper-V quirks:

1. Open **Hyper-V Manager**

2. Look for VMs named:

- mk-hv-3nodes
 - mk-hv-3nodes-m02
 - mk-hv-3nodes-m03
-

3. Right-click → **Delete**

5 Remove leftover Minikube config/data (optional)

Minikube keeps data in your **user directory**:

C:\Users\<YourUser>\.minikube

You can delete this folder entirely to **reset Minikube state**:

rmdir /s /q C:\Users\<YourUser>\.minikube

- Deletes all cached ISO images, configs, logs
 - Will require re-download when starting a new cluster
-