Kubernetes

Kubernetes (K8s) is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It provides a framework for running distributed systems resiliently, ensuring high availability, load balancing, and fault tolerance across multiple containers.

Minikube

Minikube is a lightweight Kubernetes implementation designed for local development and testing. It enables developers to run a single-node Kubernetes cluster on their local machine, making it easier to experiment with Kubernetes features before deploying applications to a full-scale production cluster.

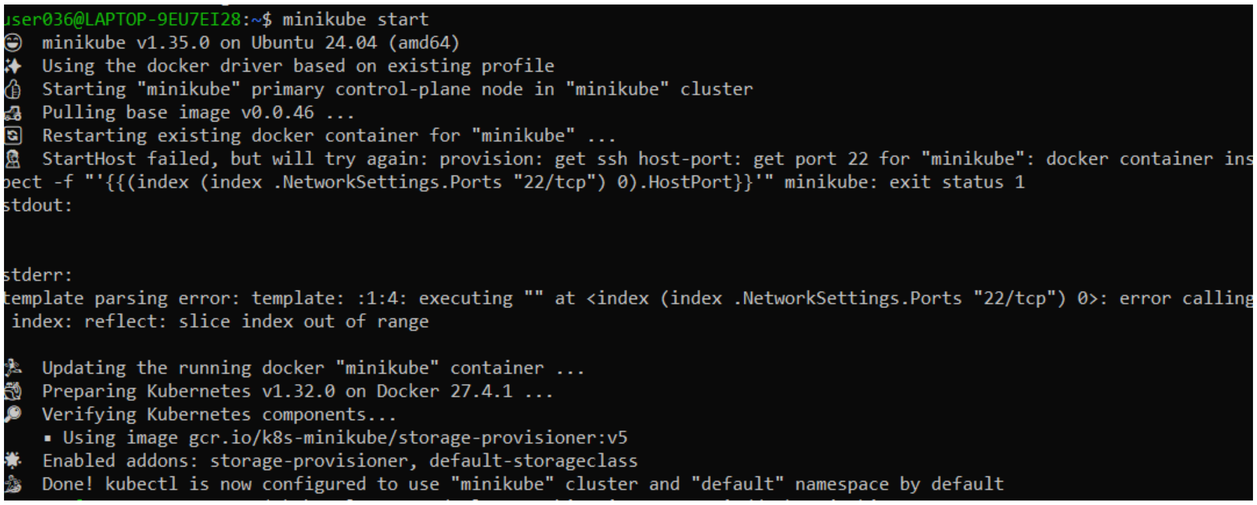
Prerequisites

★ Ubuntu installed on your system
★ Docker installed and running
★ Minikube installed and configured
★ kubectl (Kubernetes CLI) installed

Steps to Run a Docker Image in Minikube

Step 1: Start Minikube

Open a terminal and start Minikube with:

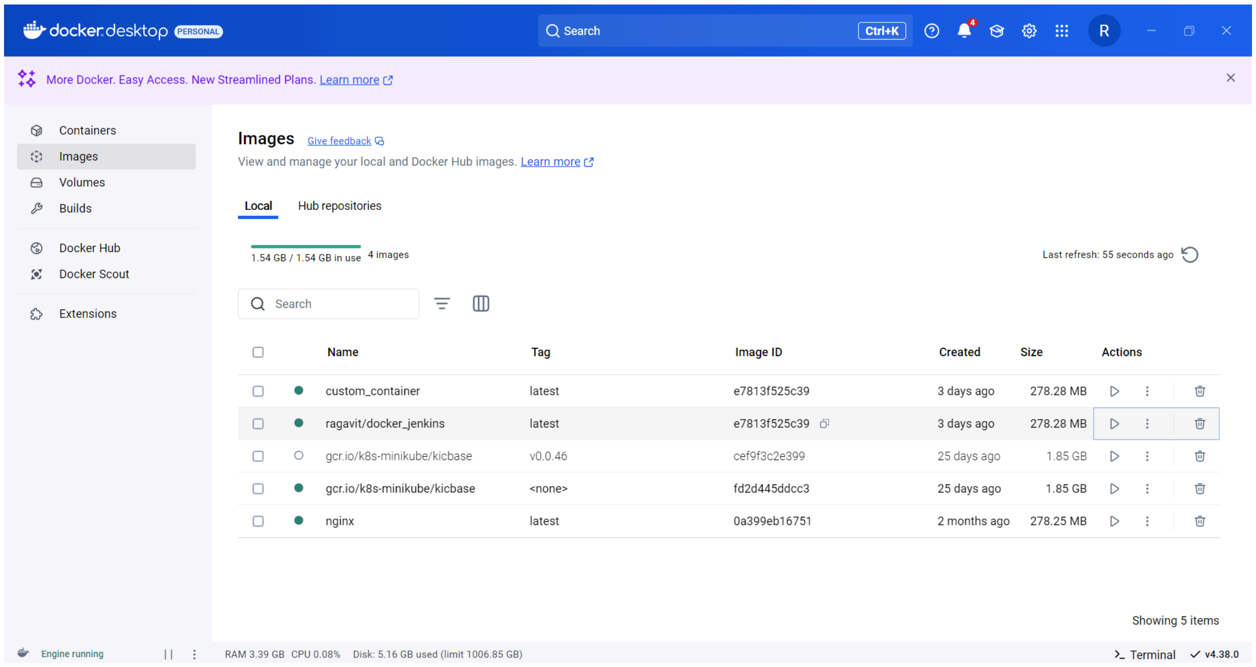


```
user036@LAPTOP-9EU7EI28:~$ minikube start
   minikube v1.35.0 on Ubuntu 24.04 (amd64)
   Using the docker driver based on existing profile
   Starting "minikube" primary control-plane node in "minikube" cluster
   Pulling base image v0.0.46 ...
   Restarting existing docker container for "minikube" ...
   StartHost failed, but will try again: provision: get ssh host-port: get port 22 for "minikube": docker container ins
pect -f "'{{(index (index .NetworkSettings.Ports "22/tcp") 0).HostPort}}'" minikube: exit status 1
stdout:


stderr:
template parsing error: template: :1:4: executing "" at <index (index .NetworkSettings.Ports "22/tcp") 0>: error calling
 index: reflect: slice index out of range

   Updating the running docker "minikube" container ...
   Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
   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 "minikube" cluster and "default" namespace by default
```

This initializes a local Kubernetes cluster.

Step 2: Run the Image in Docker

Go to docker and run the image that you want to deploy



More Docker. Easy Access. New Streamlined Plans. Learn more

Images Give feedback

View and manage your local and Docker Hub images. Learn more

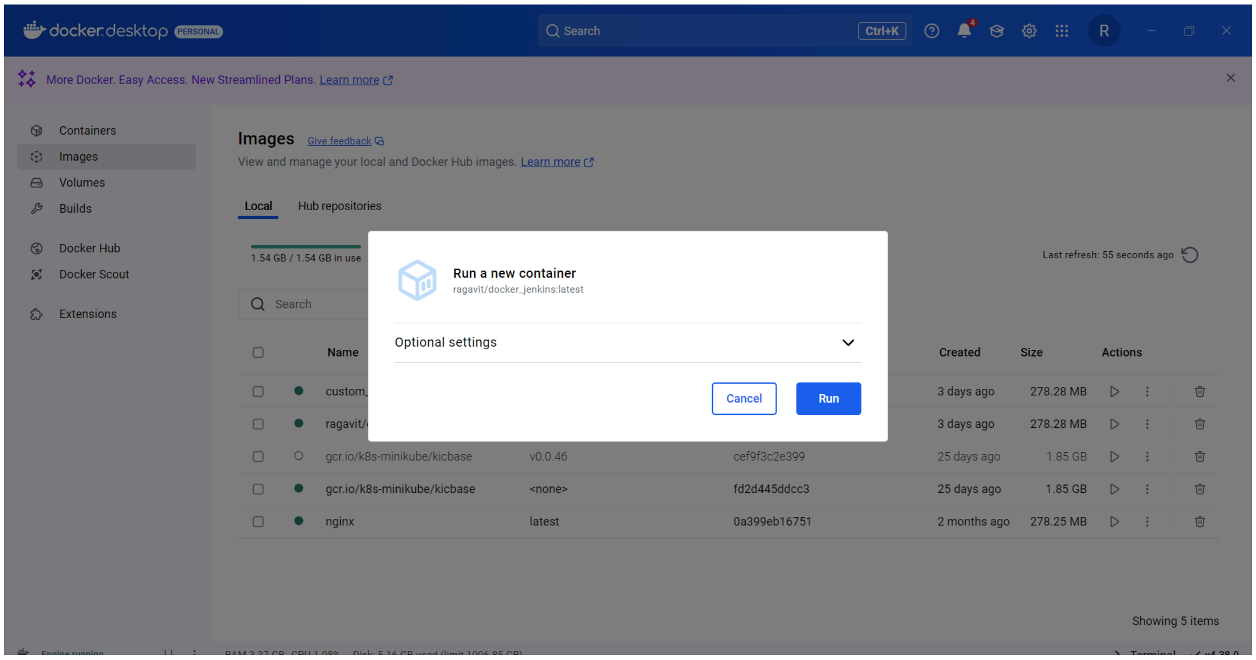
1.54 GB / 1.54 GB in use 4 images

Last refresh: 55 seconds ago

	Name	Tag	Image ID	Created	Size	Actions
●	custom_container	latest	e7813f525c39	3 days ago	278.28 MB	
●	ragavit/docker_jenkins	latest	e7813f525c39	3 days ago	278.28 MB	
○	gcr.io/k8s-minikube/kicbase	v0.0.46	cef9f3c2e399	25 days ago	1.85 GB	
●	gcr.io/k8s-minikube/kicbase	<none>	fd2d445ddcc3	25 days ago	1.85 GB	
●	nginx	latest	0a399eb16751	2 months ago	278.25 MB	

Showing 5 items

Engine running RAM 3.39 GB CPU 0.08% Disk: 5.16 GB used (limit 1006.85 GB) Terminal v4.38.0



Images Give feedback
View and manage your local and Docker Hub images. Learn more

1.54 GB / 1.54 GB in use

Last refresh: 55 seconds ago

Run a new container
ragavit/docker_jenkins:latest

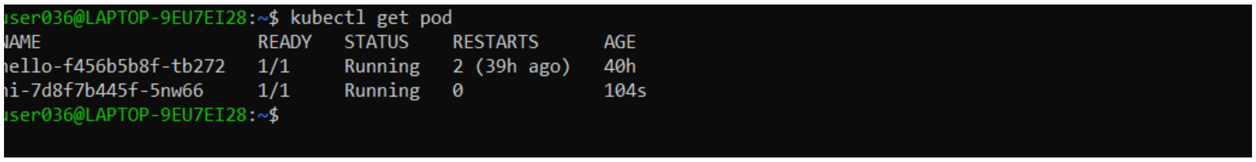
Optional settings

Name			Created	Size	Actions
custom_			3 days ago	278.28 MB	
ragavit/			3 days ago	278.28 MB	
gcr.io/k8s-minikube/kicbase	v0.0.46	cef9f3c2e399	25 days ago	1.85 GB	
gcr.io/k8s-minikube/kicbase	<none>	fd2d445ddcc3	25 days ago	1.85 GB	
nginx	latest	0a399eb16751	2 months ago	278.25 MB	

Showing 5 items

Step 3: Verify Docker Images

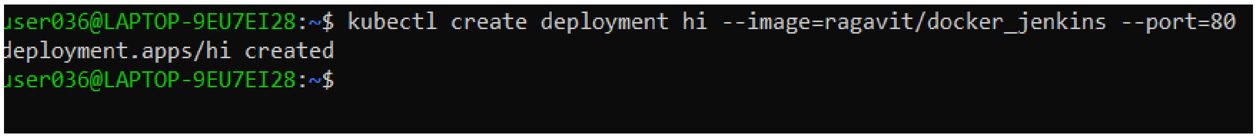
Check if your previously created Docker image is available inside Minikube's Docker environment



```
ser036@LAPTOP-9EU7EI28:~$ kubectl get pod
AME                       READY   STATUS    RESTARTS      AGE
ello-f456b5b8f-tb272      1/1     Running   2 (39h ago)   40h
i-7d8f7b445f-5nw66        1/1     Running   0             104s
ser036@LAPTOP-9EU7EI28:~$
```

Step 4: Create a Kubernetes Deployment

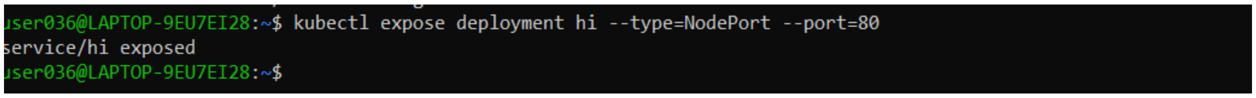
Create a Kubernetes deployment using your Docker image:



```
ser036@LAPTOP-9EU7EI28:~$ kubectl create deployment hi --image=ragavit/docker_jenkins --port=80
eployment.apps/hi created
ser036@LAPTOP-9EU7EI28:~$
```

Step 5: Expose the Deployment

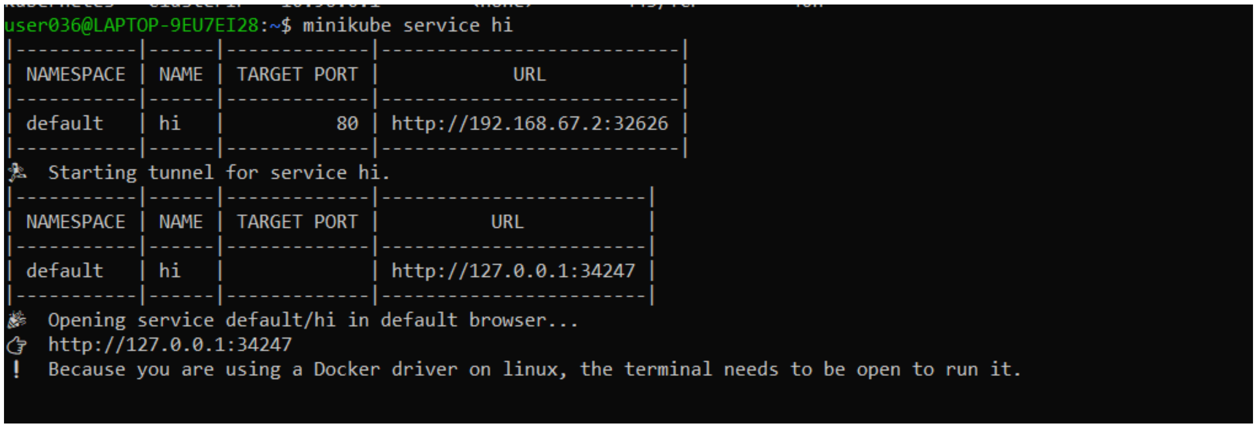
Expose the deployment as a service so it can be accessed:



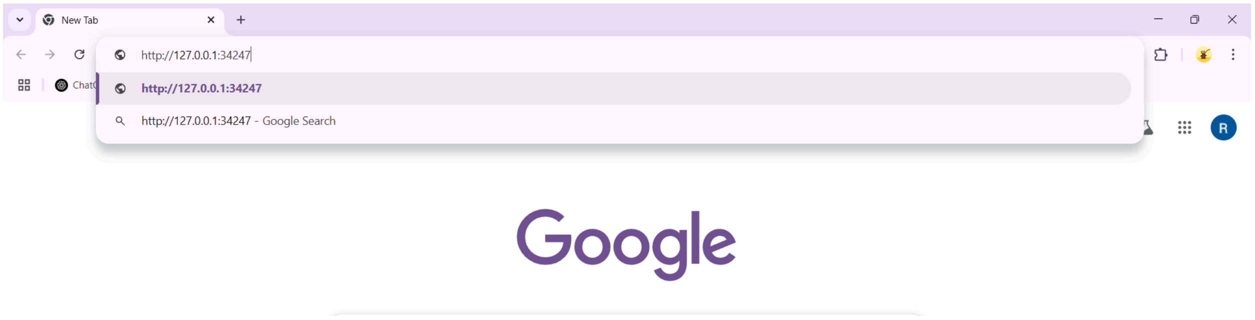
```
ser036@LAPTOP-9EU7EI28:~$ kubectl expose deployment hi --type=NodePort --port=80
ervice/hi exposed
ser036@LAPTOP-9EU7EI28:~$
```

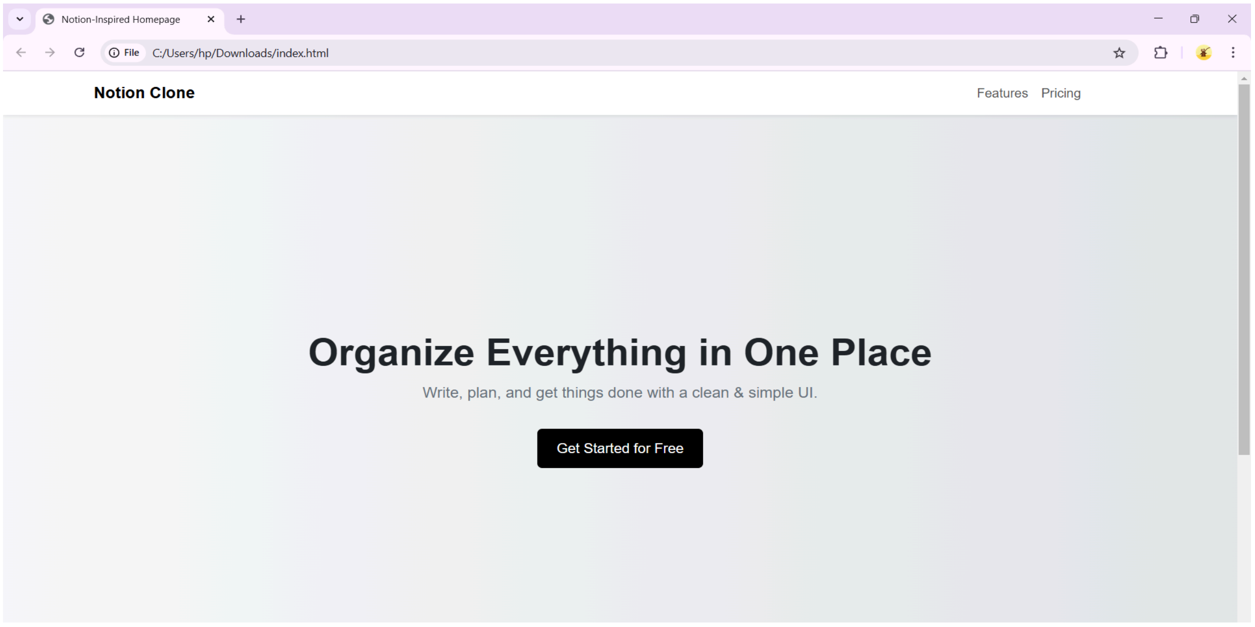
Step 6: Get the Service URL

To access your running container, get the service URL with:



```
user036@LAPTOP-9EU7EI28:~$ minikube service hi
|-----------|------|-------------|---------------------------|
| NAMESPACE | NAME | TARGET PORT |            URL            |
|-----------|------|-------------|---------------------------|
| default   | hi   |          80 | http://192.168.67.2:32626 |
|-----------|------|-------------|---------------------------|
🏃  Starting tunnel for service hi.
|-----------|------|-------------|------------------------|
| NAMESPACE | NAME | TARGET PORT |          URL           |
|-----------|------|-------------|------------------------|
| default   | hi   |             | http://127.0.0.1:34247 |
|-----------|------|-------------|------------------------|
🎉  Opening service default/hi in default browser...
👉  http://127.0.0.1:34247
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```





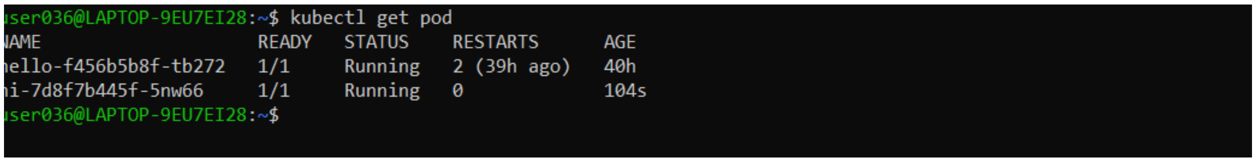
Organize Everything in One Place

Write, plan, and get things done with a clean & simple UI.

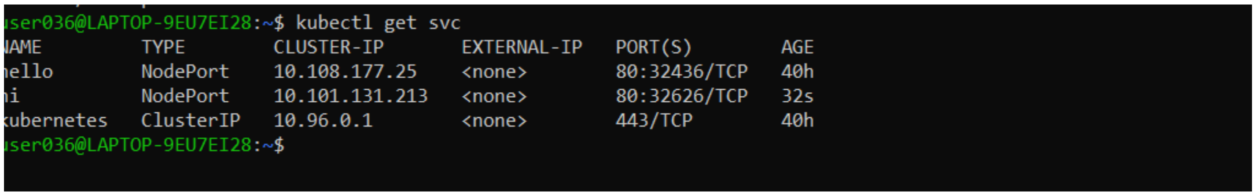
Copy and paste the provided URL into your browser to access the application.

Step 7: Verify the Deployment

Check the status of your deployment using:



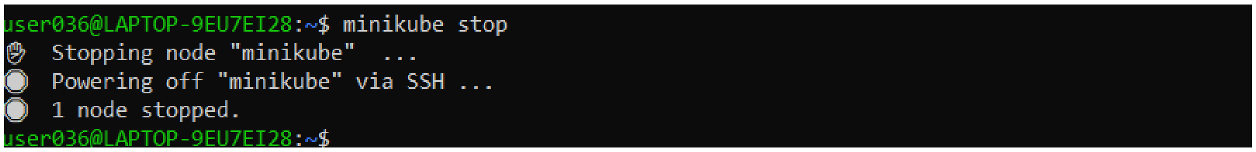
```
ıser036@LAPTOP-9EU7EI28:~$ kubectl get pod
IAME                    READY   STATUS    RESTARTS       AGE
ıello-f456b5b8f-tb272   1/1     Running   2 (39h ago)    40h
ıi-7d8f7b445f-5nw66     1/1     Running   0              104s
ıser036@LAPTOP-9EU7EI28:~$
```



```
ıser036@LAPTOP-9EU7EI28:~$ kubectl get svc
IAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
ıello        NodePort    10.108.177.25    <none>        80:32436/TCP   40h
ıi           NodePort    10.101.131.213   <none>        80:32626/TCP   32s
ubernetes    ClusterIP   10.96.0.1        <none>        443/TCP        40h
ıser036@LAPTOP-9EU7EI28:~$
```

Step 8: Stop Minikube (Optional)

Once testing is complete, you can stop Minikube with:



```
user036@LAPTOP-9EU7EI28:~$ minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
user036@LAPTOP-9EU7EI28:~$
```

This structured procedure ensures that the Docker image runs successfully in Minikube on Ubuntu.