MINIKUBE INSTALLED AND STARTED IN MY LOCAL


```
incubus@incubus-ThinkCentre-M93p:~$ minikube start --cpus=2 --memory=2g --disk-size=2g --cache-images=true
😄  minikube v1.30.1 on Ubuntu 22.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...

❗  Docker is nearly out of disk space, which may cause deployments to fail! (87% of capacity). You can pass '--force' to skip this check.
💡  Suggestion:

    Try one or more of the following to free up space on the device:

    1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
    2. Increase the storage allocated to Docker for Desktop by clicking on:
    Docker icon > Preferences > Resources > Disk Image Size
    3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
🍿  Related issue: https://github.com/kubernetes/minikube/issues/9024

🔄  Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
🔄  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
incubus@incubus-ThinkCentre-M93p:~$ kubectl create namespace muthu
namespace/muthu created
incubus@incubus-ThinkCentre-M93p:~$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   5h34m
kube-node-lease   Active   5h34m
kube-public       Active   5h34m
kube-system       Active   5h34m
muthu             Active   12s
incubus@incubus-ThinkCentre-M93p:~$ kubectl create -f pod.yaml --namespace muthu
pod/nginx-pod created
incubus@incubus-ThinkCentre-M93p:~$ kubectl get pods
```

NAME SPACE CREATED


```
🔄  Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
🔄  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
incubus@incubus-ThinkCentre-M93p:~$ kubectl create namespace muthu
namespace/muthu created
incubus@incubus-ThinkCentre-M93p:~$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   5h34m
kube-node-lease   Active   5h34m
kube-public       Active   5h34m
kube-system       Active   5h34m
muthu             Active   12s
incubus@incubus-ThinkCentre-M93p:~$ kubectl create -f pod.yaml --namespace muthu
pod/nginx-pod created
incubus@incubus-ThinkCentre-M93p:~$ kubectl get pods
No resources found in default namespace.
incubus@incubus-ThinkCentre-M93p:~$ kubectl get pods --namespace muthu
NAME        READY   STATUS             RESTARTS   AGE
nginx-pod   0/1     ContainerCreating  0          18s
incubus@incubus-ThinkCentre-M93p:~$ kubectl port forward --namespace muthu nginx-pod 80:80
error: unknown command "port" for "kubectl"

Did you mean this?
        port-forward
incubus@incubus-ThinkCentre-M93p:~$ kubectl port-forward --namespace muthu nginx-pod 80:80
Unable to listen on port 80: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:80: bind: p
ermission denied unable to create listener: Error listen tcp6 [::1]:80: bind: permission denied]
error: unable to listen on any of the requested ports: [{80 80}]
incubus@incubus-ThinkCentre-M93p:~$ kubectl port-forward --namespace muthu nginx-pod 5000:80
Forwarding from 127.0.0.1:5000 -> 80
Forwarding from [::1]:5000 -> 80
Handling connection for 5000
```

APPLICATION RUNNING INSIDE NAME SPACE ---- LOOK AT THE LOCAL HOST PORT

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.