# Kubernetes Basics

What is Kubernetes:

‘*Kubernetes is a production-grade, open-source platform that orchestrates the placement (scheduling) and execution of application containers within and across computer clusters.’*

*How it works?*

Kubernetes coordinates a highly available cluster of computers that are connected to work as a single unit.

* The **Control Plane** coordinates the cluster
* **Nodes** are the workers that run applications

A node is a VM or a physical computer that serves as a worker machine in a Kubernetes cluster

Dashboard and minikube started:  
A screenshot of a computer

AI-generated content may be incorrect.

Deployment to manage a pod:  
A screen shot of a computer program

AI-generated content may be incorrect.

How to view the pods:

A screen shot of a computer

AI-generated content may be incorrect.

How to view the event log:

A computer screen shot of a black screen

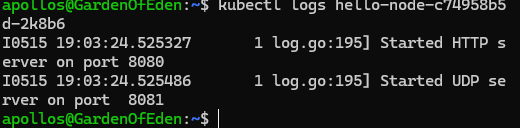
AI-generated content may be incorrect.

Config check:

A screenshot of a computer screen

AI-generated content may be incorrect.

How to view specific applications logs:



How to run a proxy for Kubernetes:  
A black screen with white text

AI-generated content may be incorrect.

You can now query the Kubernetes cluster directly:  
A computer screen shot of white text

AI-generated content may be incorrect.

Troubleshooting commands:

* kubectl get - list resources
* kubectl describe - show detailed information about a resource
* kubectl logs - print the logs from a container in a pod
* kubectl exec - execute a command on a container in a pod

how to view the containers within a pod:

kubectl describe pods:

A screenshot of a computer screen

AI-generated content may be incorrect.

How to set a pod name variable:

export POD\_NAME="**$(**kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}'**)**"

echo Name of the Pod: $POD\_NAME

A screen shot of a computer

AI-generated content may be incorrect.

How to execute commands on the container:

kubectl exec "$POD\_NAME" – env

A computer screen with white text

AI-generated content may be incorrect.

## Creating a service

Use the expose command to create a service and expose it to the outside world:

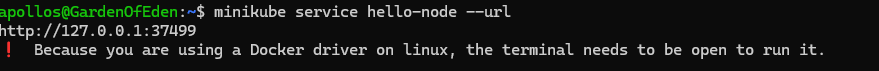
A black screen with white text

AI-generated content may be incorrect.

Test that it is exposed:

curl http://"**$(**minikube ip**)**:$NODE\_PORT"

or on docker:



A screenshot of a computer

AI-generated content may be incorrect.

## How to label a pod

A black background with white text

AI-generated content may be incorrect.

## Deleting a service:

kubectl delete service -l

A black background with white text

AI-generated content may be incorrect.

kubectl get deployments

A black background with white text

AI-generated content may be incorrect.

* *NAME* lists the names of the Deployments in the cluster.
* *READY* shows the ratio of CURRENT/DESIRED replicas
* *UP-TO-DATE* displays the number of replicas that have been updated to achieve the desired state.
* *AVAILABLE* displays how many replicas of the application are available to your users.
* *AGE* displays the amount of time that the application has been running.

A black background with white text

AI-generated content may be incorrect.

Get rs shows the replica set created with the deployment

Using the SCALE command we can create multiple instances of the application:

A screen shot of a computer program

AI-generated content may be incorrect.

## How to rolling update the application:

Using the set image subcommand, we can change the image to another with a new version:  
kubectl set image deployments/<deployment-name> <deployment-name>=docker.io/<new\_image>:<version>

You can also confirm the update by running the rollout status subcommand:

kubectl rollout status deployments/<deployment-name>

To roll back an update, use the rollout undo subcommand:

kubectl rollout undo deployments/<deployment-name>