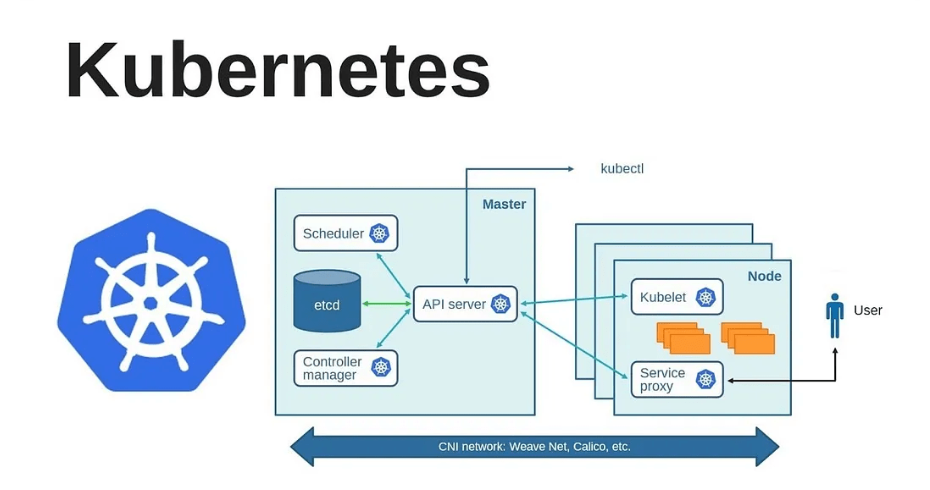
**Kubernetes Architecture Explained**

* [Control Plane (Master Node Components)](https://github.com/Imtiyaj5791/kubestarter/blob/main/kubernetes_architecture.md#control-plane-master-node-components)
* [Worker Node Components](https://github.com/Imtiyaj5791/kubestarter/blob/main/kubernetes_architecture.md#worker-node-components)
* [Other Components](https://github.com/Imtiyaj5791/kubestarter/blob/main/kubernetes_architecture.md#other-components)

**Control Plane (Master Node Components)  
  
API Server: -** This is the "front desk" of Kubernetes. Whenever you want to interact with your cluster, your request goes through the API Server. It validates and processes these requests to the backend components.  
  
**etcd: -** Think of this as the "database" of Kubernetes. It stores all the information about your cluster, what nodes are part of the cluster, what pods are running, what their statuses are, and more.  
  
**Scheduler: -** The "event planner" for your containers. When you ask for a container to be run, the Scheduler decides which machine (Node) in your cluster should run it. It considers resource availability and other constraints while making this decision.

### **Controller Manager: -** Imagine a bunch of small robots that continuously monitor the cluster to make sure everything is running smoothly. If something goes wrong (e.g., a Pod crashes), they work to fix it, ensuring the cluster state matches your desired state. **Cloud Controller Manager: -** This is a specialized component that allows Kubernetes to interact with the underlying cloud provider, like AWS or Azure. It helps in tasks like setting up load balancers and persistent storage.

## **Worker Node Components: -**

### **Kubelet: -** This is the "manager" for each worker node. It ensures all containers on the node are healthy and running as they should be. **kube-proxy: -** Think of this as the "traffic cop" for network communication either between Pods or from external clients to Pods. It helps in routing the network traffic appropriately. **Container Runtime: -** This is the software used to run containers. Docker is commonly used, but other runtimes like containerd can also be used.

## **Other Components: -**

### **Pod: -** The smallest unit in Kubernetes, a Pod is a group of one or more containers. Think of it like an apartment in an apartment building. **Service: -** This is like a phone directory for Pods. Since Pods can come and go, a Service provides a stable "address" so that other parts of your application can find them. **Volume: -** This is like an external hard-drive that can be attached to a Pod to store data. **Namespace: -** A way to divide cluster resources among multiple users or teams. Think of it as having different folders on a shared computer, where each team can only see their own folder. (Namespace is a logical group in which you can have resources like container) **Ingress: -** Think of this as the "front door" for external access to your applications, controlling how HTTP and HTTPS traffic should be routed to your services.

**How to Create a K8s Cluster: -**

EC2 with kubeadm  
Minikube (K8s in a Docker container)  
KinD (Master and worker are container)  
Managed Kubernetes (EKS, AKS, GKE)  
KillerKoda  
Rancher

**KIND Cluster Setup Guide: -**

Sudo apt-get update

**Installing Docker and add current user in docker group**

sudo apt install docker.io

sudo usermod -aG docker $USER

newgrp docker  
  
**Installing KIND: -**

[ $(uname -m) = x86\_64 ] && curl -Lo ./kind <https://kind.sigs.k8s.io/dl/v0.27.0/kind-linux-amd64>

chmod +x ./kind

sudo mv ./kind /usr/local/bin/kind

kind –version  
  
A computer screen with text and numbers

AI-generated content may be incorrect.

**Installing kubectl: -**

curl -LO [https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl](https://dl.k8s.io/release/$(curl%20-L%20-s%20https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl)

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version

A screen shot of a computer

AI-generated content may be incorrect.

**Setting Up the KIND Cluster: -**

* Create a kind-cluster-config.yaml file:
* mkdir k8s
* cd k8s
* vim cluster.yml  
  A computer screen shot of a program

  AI-generated content may be incorrect.

**Create the cluster using the configuration file:**

* kind create cluster --config cluster.yml --name tws-cluster

A computer screen with text on it

AI-generated content may be incorrect.

* **Verify the cluster:**

kubectl get nodes

A screen shot of a computer screen

AI-generated content may be incorrect.

* **If multiple clusters, then how to set or use current cluster**

kubectl config get-contexts

kubectl config set-context --cluster=kind-tws-cluster --current

A screen shot of a computer

AI-generated content may be incorrect.

Namespace---Logical Boundary or isolated network

Nginx-----Container-----Pod -----Deployment(scaling)----Kube-proxy (interact service from outside) -----User

A diagram of a diagram

AI-generated content may be incorrect.

**Create Namespace: -**vim namespace.yml  
kubectl apply -f namespace.yml  
kubectl get ns  
A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Create Pod: -**

vim pod.yml

kubectl apply -f pod.yml

kubectl get pods -n nginx-ns

kubectl get pods -o wide ---- find out which worker nodes your pods are running

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Create Deployment: -**Vim deployment.ymlkubectl apply -f deployment.ymlkubectl get deployment -n nginx-nskubectl get pods -n nginx-nskubectl delete pod nginx-deployment-657764f66c-6qcf8 -n nginx-nskubectl scale deployment nginx-deployment -n nginx-ns --replicas=3 (pods will be 3) **Deployment: -** is desired state of your pods that allows you to scale up and down as needed it.

**Rolling updates vs repilcaset: -**

**ReplicaSet** focuses on maintaining the number of pod replicas, a **Deployment** provides a higher-level management tool that includes features like rolling updates and rollbacks, making it more versatile for application lifecycle management.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Create Service: -**A Service is a way to expose your application running in pods to other applications or users, both inside and outside the cluster **,** There are different types of services, ClusterIP: - It exposes the service on an internal IP within the cluster.NodePort: - Exposes the service on a specific port on each node in the cluster.LoadBalancer (external access via cloud provider's load balancer)External IP: - Maps an external IP to a service within the cluster.Headless: -

A screen shot of a computer

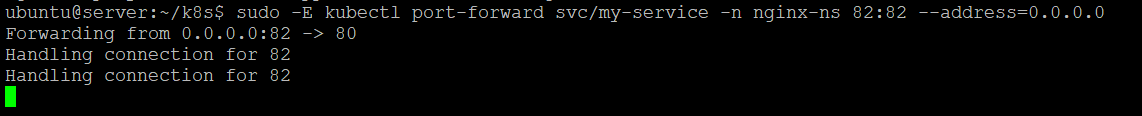
AI-generated content may be incorrect.  
target port is container port and port where you want to map.



A black screen with white text

AI-generated content may be incorrect.

sudo -E kubectl port-forward svc/my-service -n nginx-ns 82:82 --address=0.0.0.0



**Deploy online-shop: -**

git clone <https://github.com/Imtiyaj5791/online_shopping_app.git>

cd online\_shopping\_app/

docker build -t online-shop:latest .

docker image tag online-shop:latest flexis008/online-shop:latest

docker login

docker push flexis008/online-shop:latest

**create a namespace: -**  
A black screen with white text

AI-generated content may be incorrect.

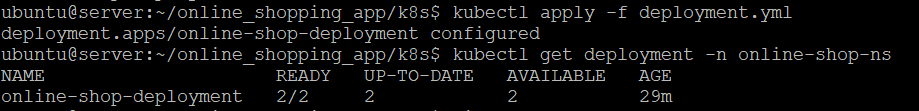
A screenshot of a computer

AI-generated content may be incorrect.

**Create a deployment: -**

A screen shot of a computer

AI-generated content may be incorrect.



**Create a service: -**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**sudo -E kubectl port-forward svc/online-shop-svc -n online-shop-ns 5173:5173 --address=0.0.0.0 &**

A screenshot of a computer

AI-generated content may be incorrect.

**Create Ingress: -** Path Based routing

kubectl apply -f <https://kind.sigs.k8s.io/examples/ingress/deploy-ingress-nginx.yaml>

A screen shot of a computer program

AI-generated content may be incorrect.

**kubectl get svc -n ingress-nginx  
kubectl get pods -n ingress-nginx  
sudo -E kubectl port-forward svc/ingress-nginx-controller -n ingress-nginx 80:80 --address=0.0.0.0**

A screen shot of a computer

AI-generated content may be incorrect.

**Now,  
We have nginx and Apache deployment and also online-shop deployment  
rewrite IP/nginx, IP/Apache , IP/shop  
apache-service.yml**

A screen shot of a computer

AI-generated content may be incorrect.

**apache.yml (deployment)**

A screenshot of a computer program

AI-generated content may be incorrect.

**Ingress.yml (for nginx and apache because they are on same namespace)**A computer screen shot of a black screen

AI-generated content may be incorrect.

**Ingress for online shops (this is different name space)**

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Secret and Config Map and Persistent Volumes: -**

A screen shot of a computer

AI-generated content may be incorrect.

**echo -n 'Admin123' | base64 ----- output (QWRtaW4xMjM=)**

A screen shot of a computer

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.