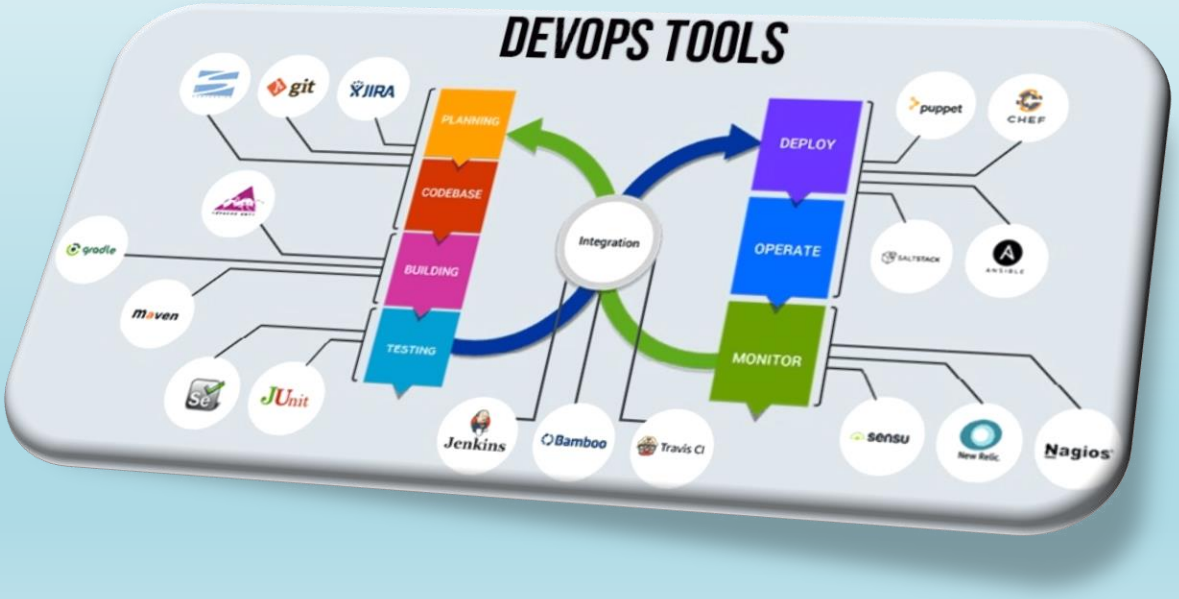
CLOUD TRAIN
ACCELERATE YOUR GROWTH

Container Orchestration [Kubernetes]



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Agenda

- INTRODUCTION TO KUBERNETES
- DOCKER SWARM VS. KUBERNETES
- KUBERNETES ARCHITECTURE
- KUBERNETES INSTALLATION
- WORKING OF KUBERNETES
- DEPLOYMENTS IN KUBERNETES
- SERVICES IN KUBERNETES
- INGRESS IN KUBERNETES
- KUBERNETES DASHBOARD

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Introduction to Kubernetes

Introduction to Kubernetes

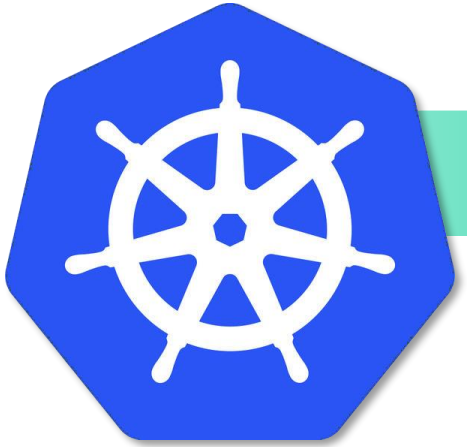
CLOUD TRAIN
ACCELERATE YOUR GROWTH

- Kubernetes is an open-source container orchestration software.
- It was originally developed by Google.
- It was first released on July 21, 2015.
- It is the ninth most active repository on GitHub in terms of number of commits.

Features of Kubernetes

CLOUD TRAIN
ACCELERATE YOUR GROWTH

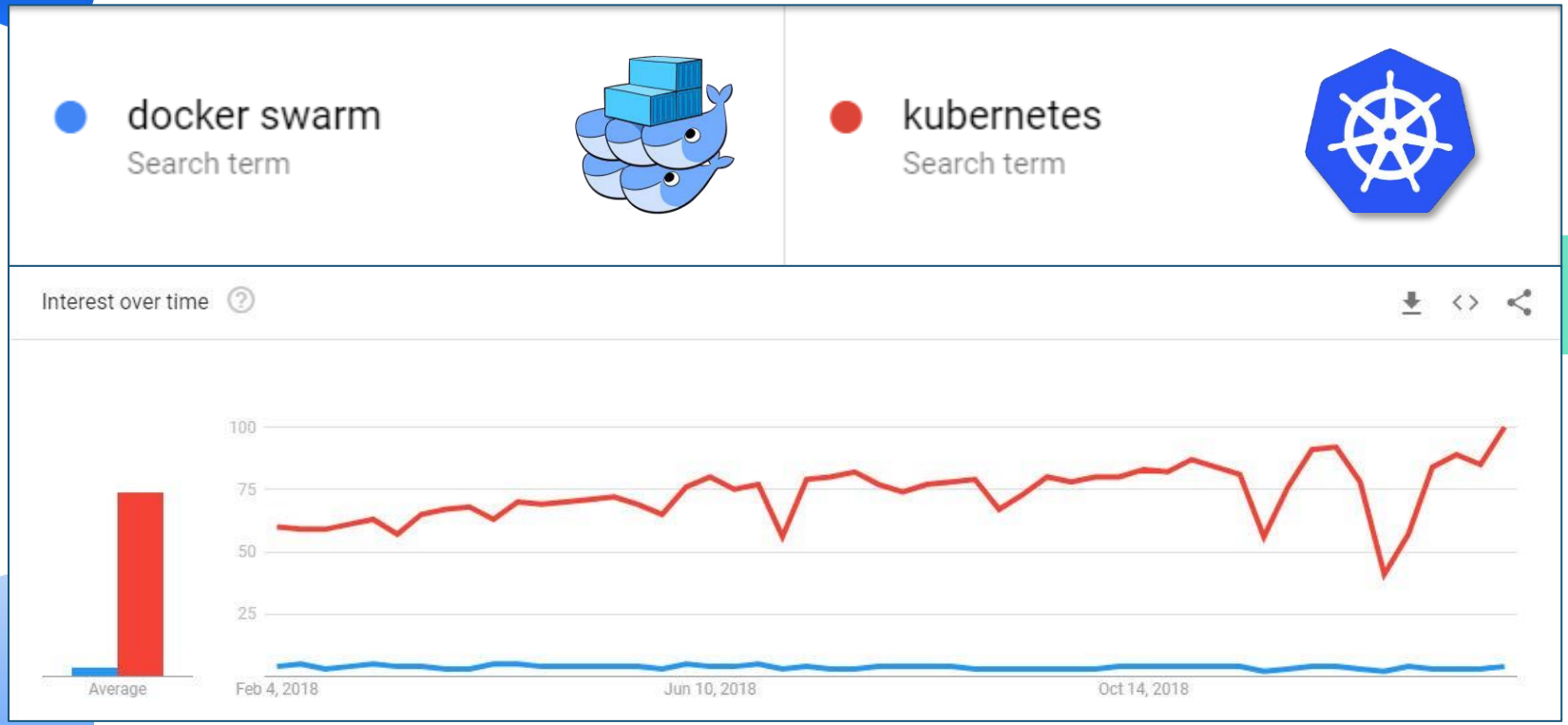
- Pods
- Replication Controller
- Storage Management
- Resource Monitoring
- Health Checks
- Service Discovery
- Networking
- Secret Management
- Rolling Updates



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Docker Swarm Vs. Kubernetes

Docker Swarm Vs. Kubernetes



Source: trends.google.com

Docker Swarm Vs. Kubernetes

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Docker Swarm

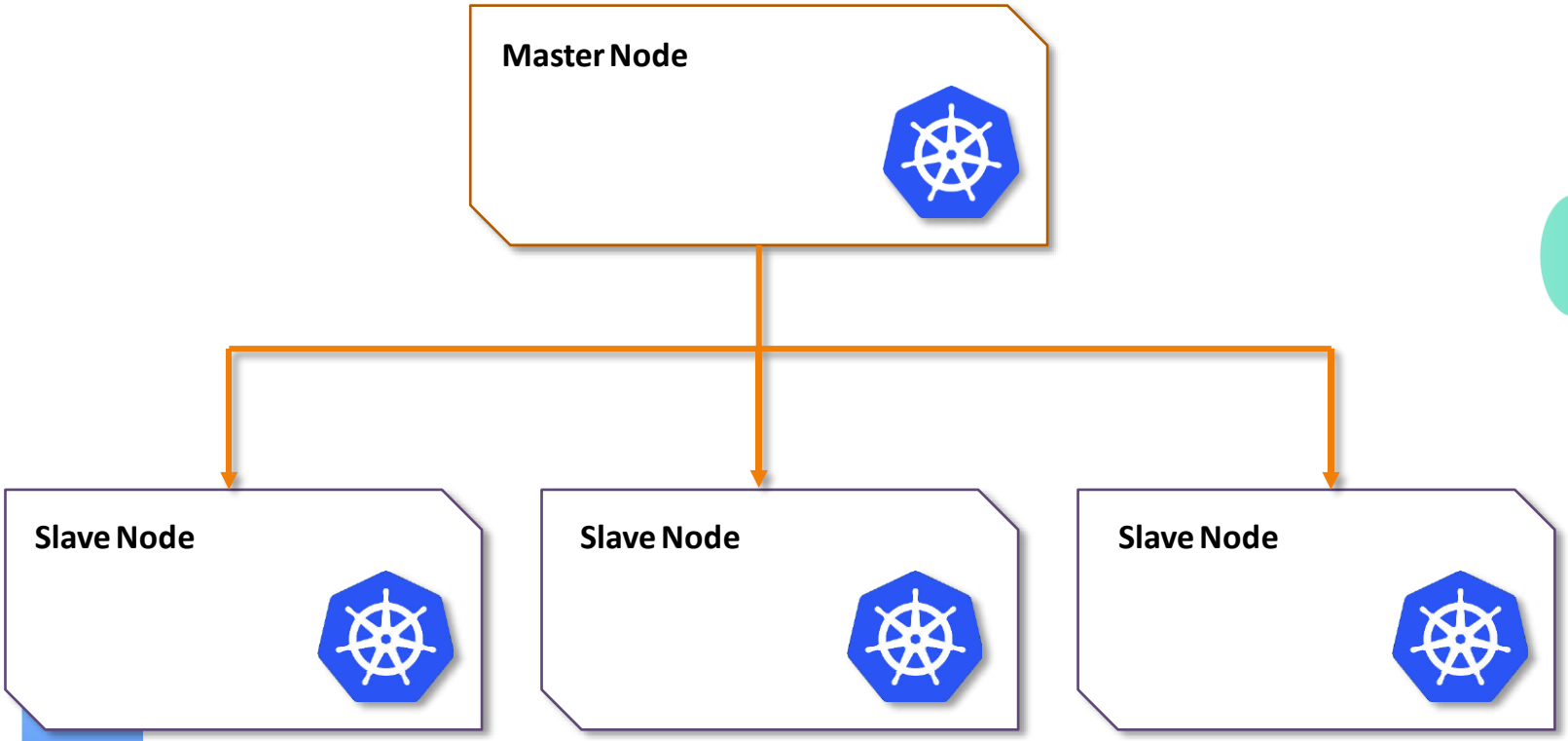
- Easy to install and initialize
- Faster when compared to Kubernetes
- Not reliable and has less features

Kubernetes

- Complex procedure to install
- Slower when compared to Docker Swarm
- More reliable and has more features

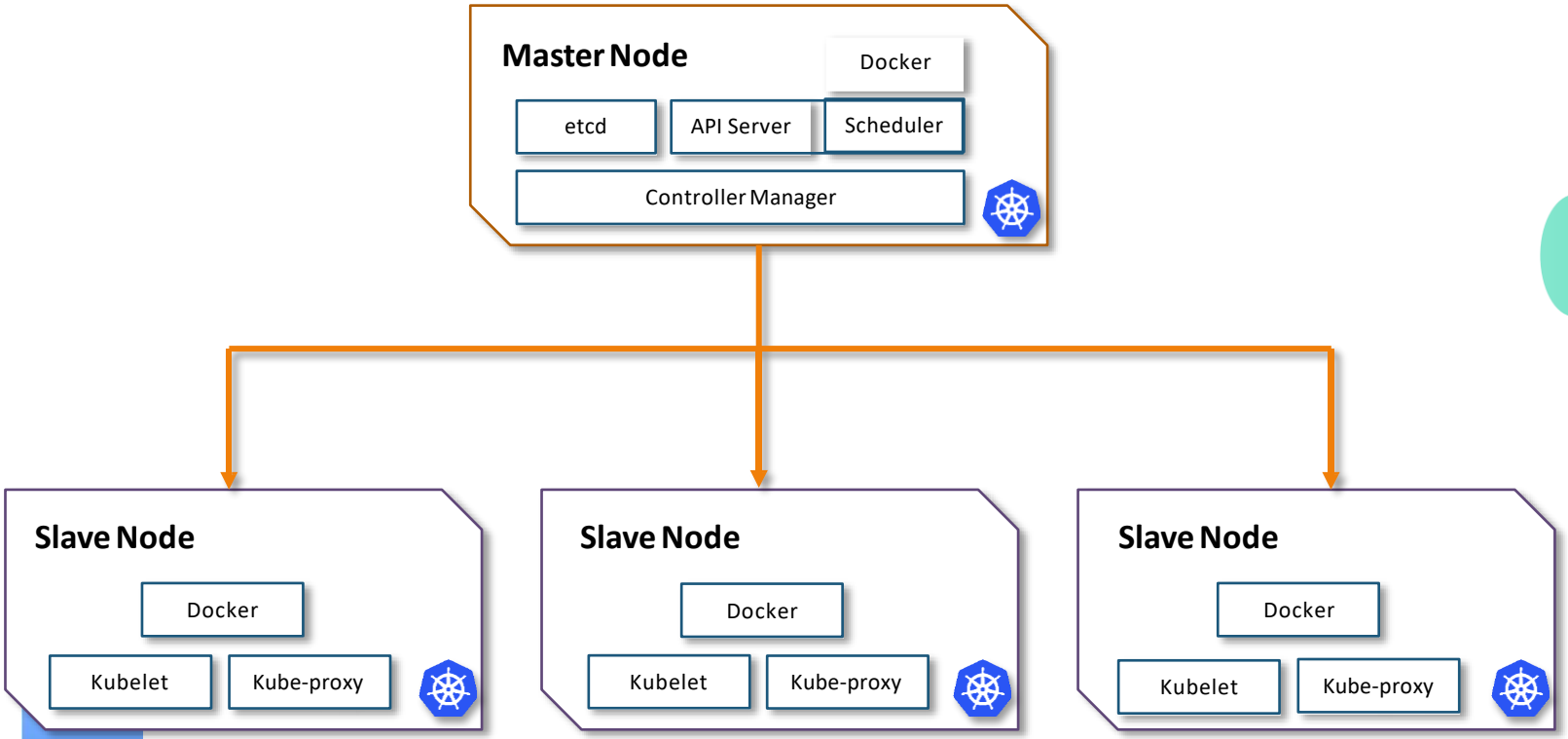
Kubernetes Architecture

Kubernetes Architecture



Kubernetes Architecture

CLOUD TRAIN
ACCELERATE YOUR GROWTH



CLOUD TRAIN
ACCELERATE YOUR GROWTH

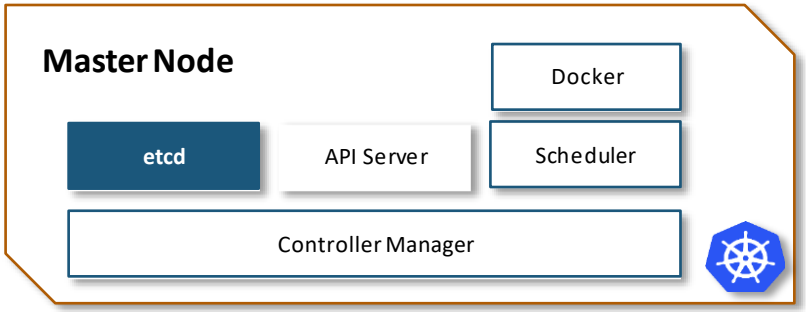
Kubernetes Architecture: Master Components

Kubernetes Architecture: Master Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- **etcd**
- API Server
- Scheduler
- Controller Manager

It is a highly available distributed key–value store, which is used to store cluster wide secrets. It is only accessible by the Kubernetes API server, as it has sensitive information.

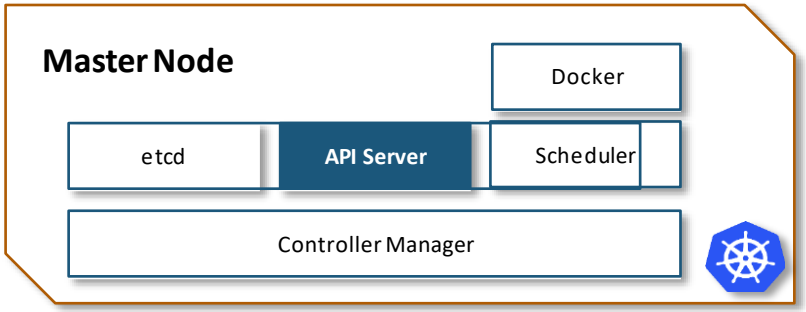


Kubernetes Architecture: Master Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- etcd
- **API Server**
- Scheduler
- Controller Manager

It exposes Kubernetes API. Kubernetes API is the front-end for the Kubernetes Control Plane and is used to deploy and execute all operations in Kubernetes.

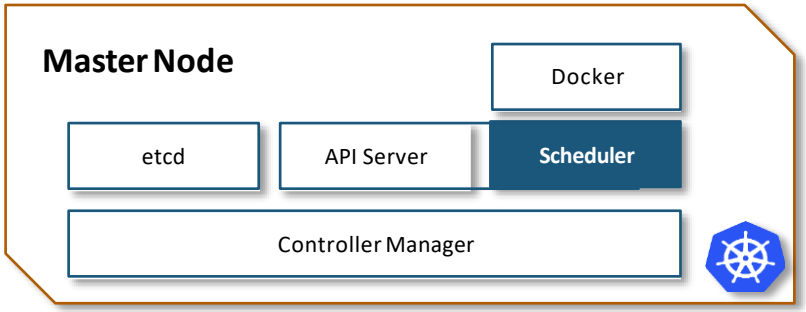


Kubernetes Architecture: Master Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- etcd
- API Server
- **Scheduler**
- Controller Manager

The scheduler takes care of scheduling of all processes and the dynamic resource management and manages present and future events on the cluster.

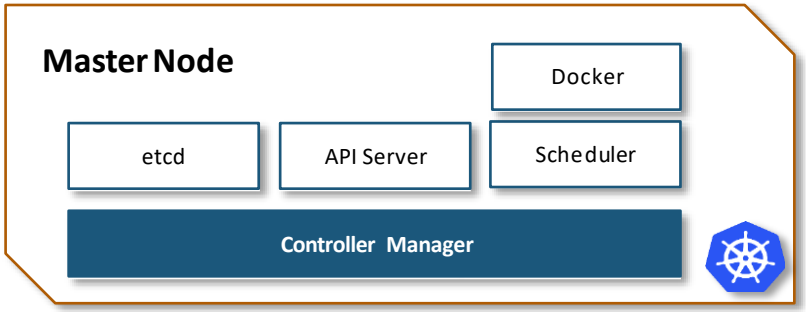


Kubernetes Architecture: Master Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

- etcd
- API Server
- Scheduler
- **Controller Manager**

The controller manager runs all controllers on the Kubernetes cluster. Although each controller is a separate process, to reduce complexity, all controllers are compiled into a single process. They are as follows: **Node Controller, Replication Controller, Endpoints Controller, Service Accounts and TokenControllers**.



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Kubernetes Architecture: Slave Components

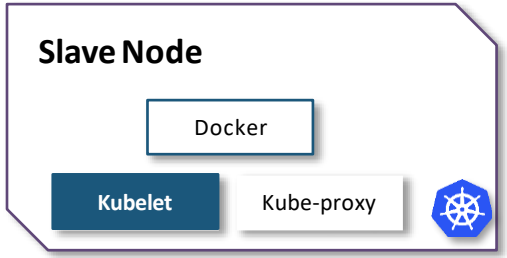
Kubernetes Architecture: Slave Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Kubelet takes the specification from the API server and ensures that the application is running according to the specifications which were mentioned. Each node has its own kubelet service.

Kubelet

Kube-proxy



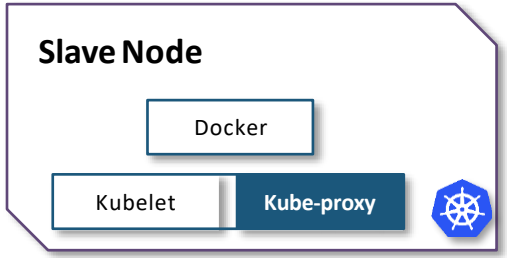
Kubernetes Architecture: Slave Components

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Kubelet

Kube-proxy

This proxy service runs on each node and helps in making services available to the external host. It helps in connection forwarding to the correct resources. It is also capable of doing primitive load balancing.

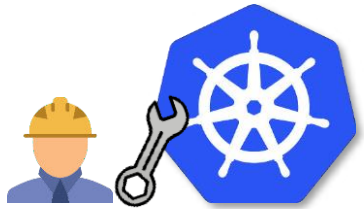


Kubernetes Installation

Kubernetes Installation

There are numerous ways to install Kubernetes. Following are some of the popular ways:

- **Kubeadm**: Bare Metal Installation
- **Minikube**: Virtualized Environment for Kubernetes
- **Kops**: Kubernetes on AWS
- **Kubernetes on GCP**: Kubernetes running on Google Cloud Platform(GKE)



CLOUD TRAIN
ACCELERATE YOUR GROWTH

Hands-on: Installing Kubernetes Using Kubeadm

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Working of Kubernetes

Working of Kubernetes

Pods can have one or more containers coupled together. They are the basic unit of Kubernetes. To increase high availability, we always prefer pods to be in replicas.

Working of Kubernetes

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Services are used to load balance the traffic among the pods. It follows round-robin distribution among the healthy pods.

Service

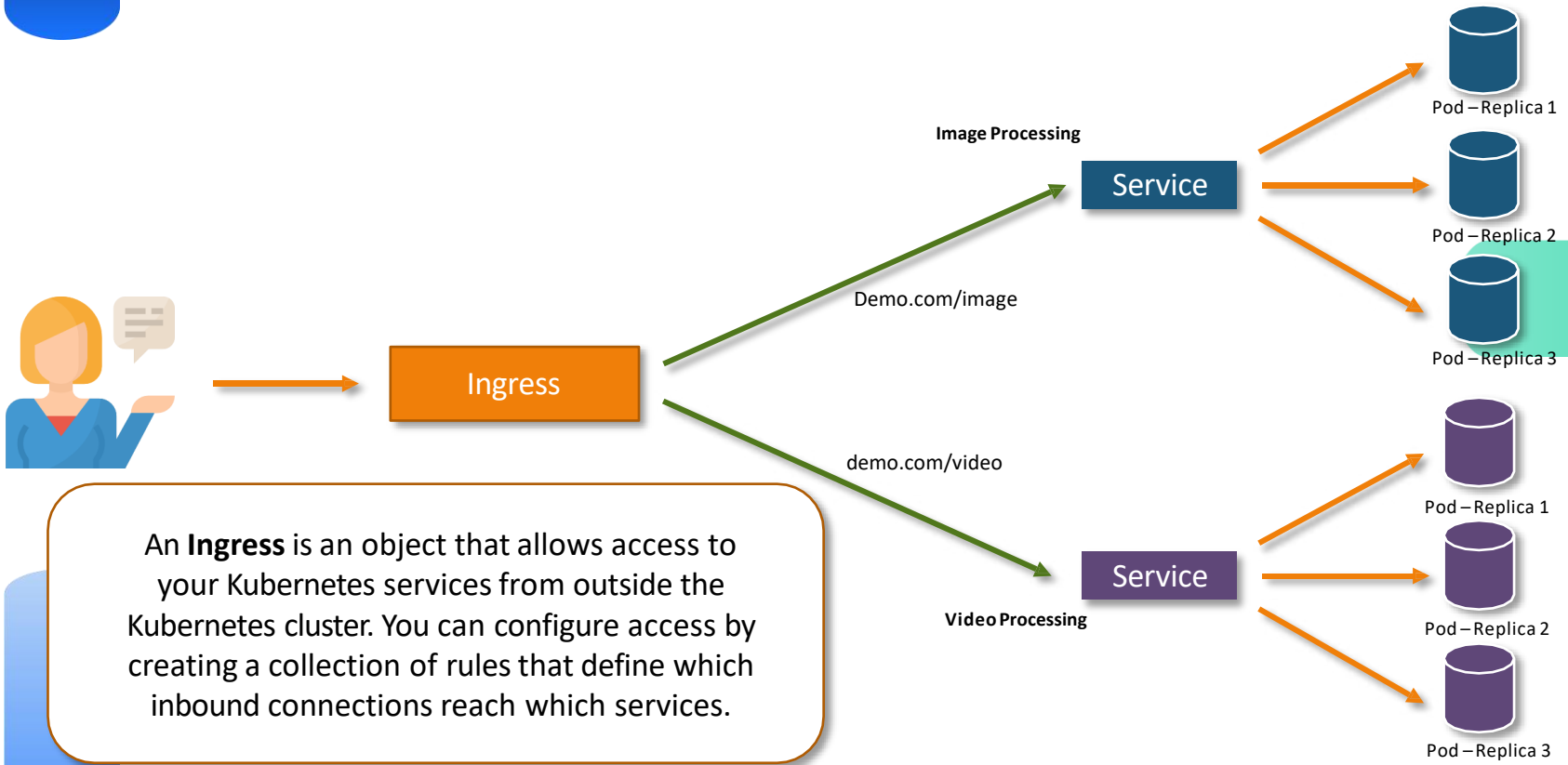
Pod – Replica 1

Pod – Replica 2

Pod – Replica 3

Working of Kubernetes

CLOUD TRAIN
ACCELERATE YOUR GROWTH



An **Ingress** is an object that allows access to your Kubernetes services from outside the Kubernetes cluster. You can configure access by creating a collection of rules that define which inbound connections reach which services.

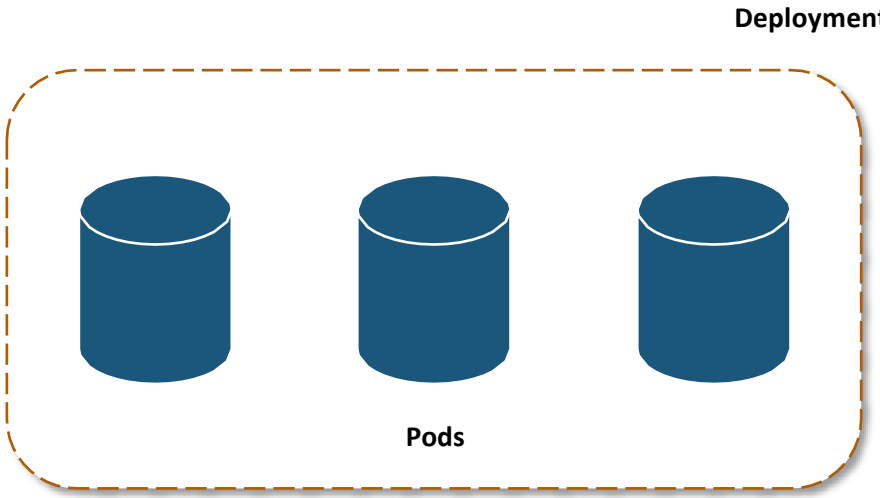
CLOUD TRAIN
ACCELERATE YOUR GROWTH

Deployments in Kubernetes

Deployments in Kubernetes

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Deployment in Kubernetes is a controller which helps your applications reach the desired state; the desired state is defined inside the deployment file.



YAML Syntax for Deployments

This YAML file will deploy 3 pods for nginx and will maintain the desired state, which is 3 pods, until this deployment is deleted.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
  matchLabels:
     app: nginx
  template:
  metadata:
     labels:
       app:nginx
   spec:
     containers:
     - name: nginx
       image:nginx:1.7.9
       ports:
       - containerPort:80
```

Creating a Deployment

Once the file is created, to deploy this deployment use the following syntax:

Syntax

```
kubectl create –f nginx.yaml
```

```
ubuntu@master:~$ kubectl create -f nginx.yaml
deployment.apps/nginx-deployment created
```

Listing the Pods

To view the pods, type the following command:

Syntax

```
kubectl get po
```

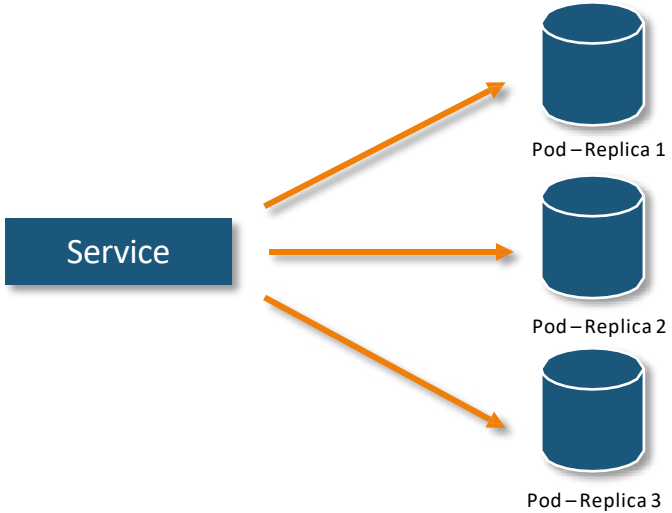
```
ubuntu@master:~$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-585449566-89kmp    1/1     Running   0          9s
nginx-deployment-585449566-bpn8x    1/1     Running   0          9s
nginx-deployment-585449566-fkjdt    1/1     Running   0          9s
```

As you can see, the number of pods are matching with the number of replicas specified in the deployment file.

Creating a Service

Creating a Service

A Service is basically a round-robin load balancer for all pods, which matches with its name or selector. It constantly monitors the pods; in case a pod gets unhealthy, the service will start deploying the traffic to other healthy pods.



Service Types

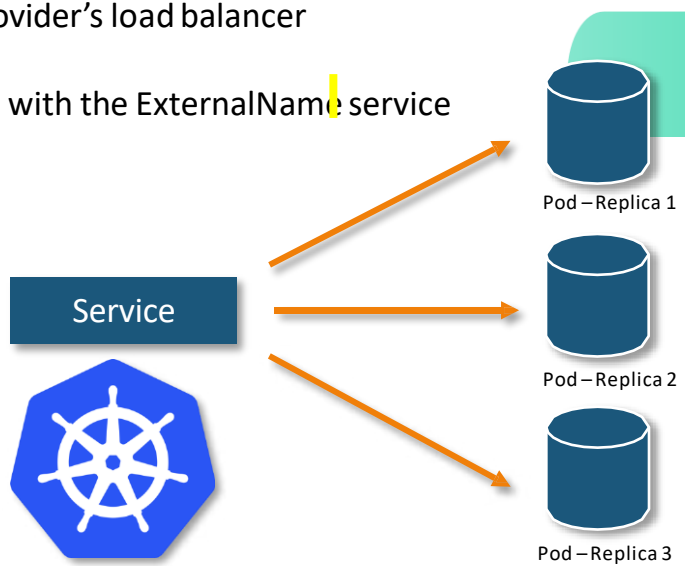
CLOUD TRAIN
ACCELERATE YOUR GROWTH

ClusterIP: Exposes the service on cluster-internal IP

NodePort: Exposes the service on each Node's IP at a static port

LoadBalancer: Exposes the service externally using a cloud provider's load balancer

ExternalName: Maps the service to the DNS Name mentioned with the ExternalName service



Creating a NodePort Service

We can create a NodePort service using the following syntax:

Syntax

```
kubectl create service nodeport <name-of-service> --tcp=<port-of-service>:<port-of-container>
```

```
ubuntu@master:~$ kubectl create service nodeport nginx --tcp=80:80
service/nginx created
```

Creating a NodePort Service

To know the port, on which the service is being exposed, type the following command:

Syntax

```
kubectl get svc nginx
```

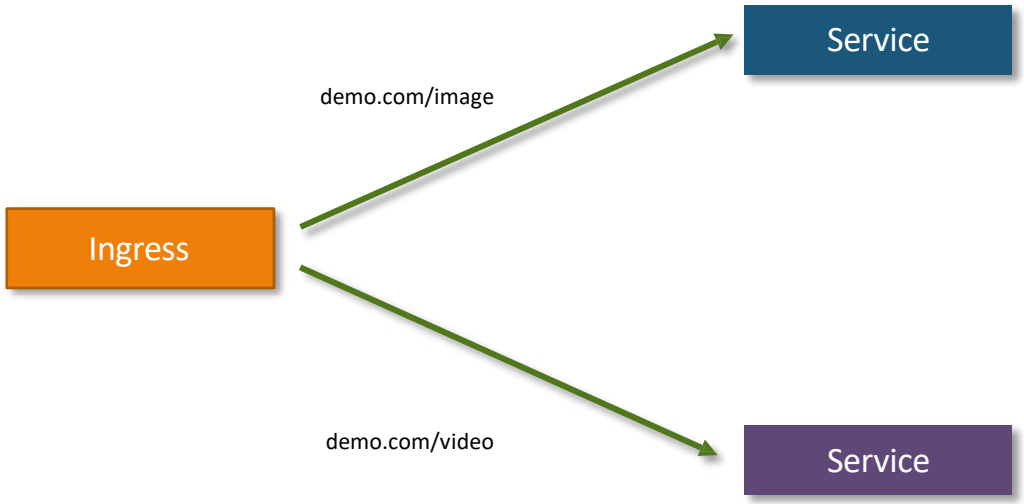
```
ubuntu@master:~$ kubectl get svc nginx
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
nginx   NodePort    10.104.95.243   <none>        80:32256/TCP   31s
```

Creating an Ingress

What is an Ingress?

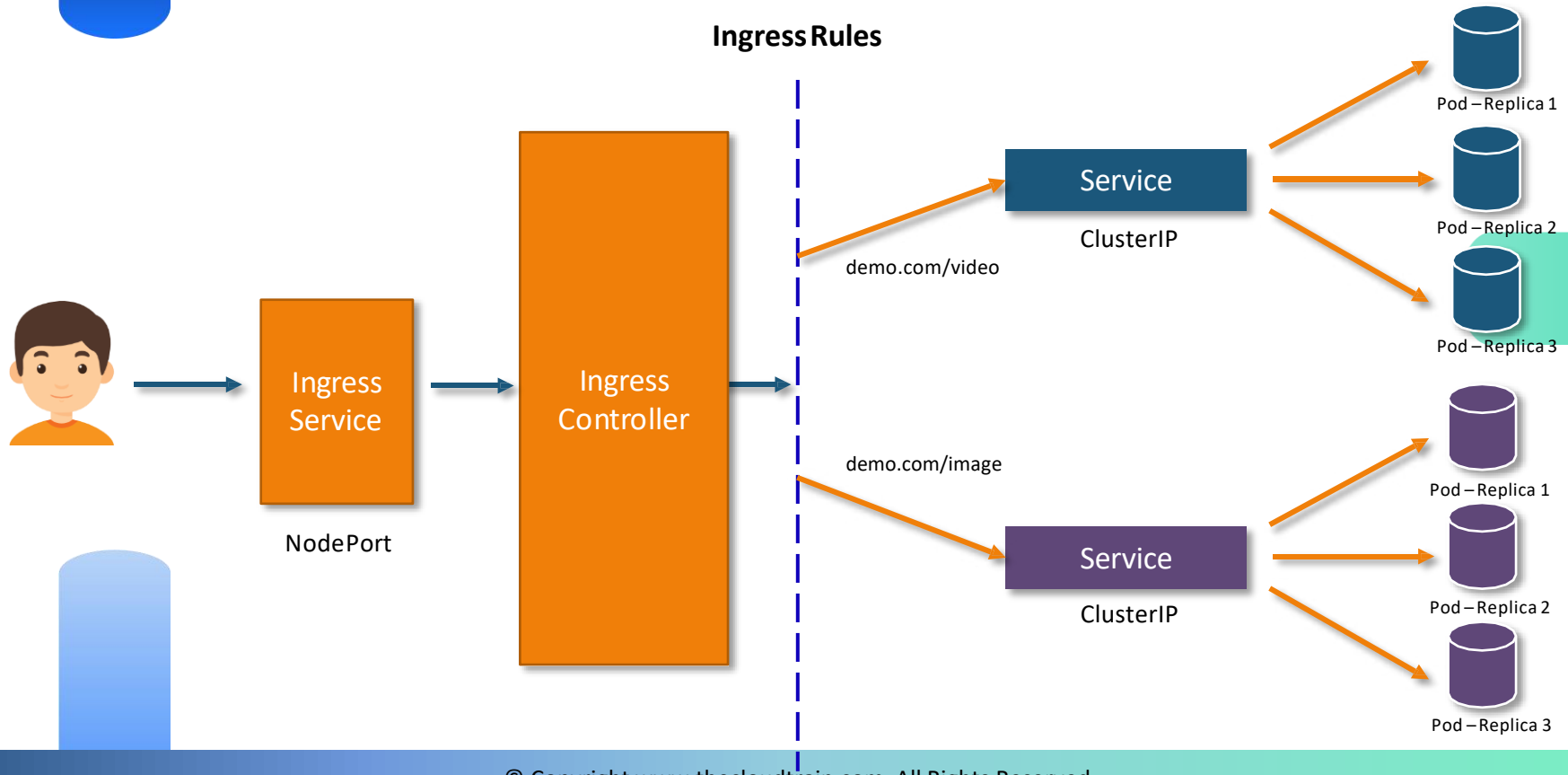
CLOUD TRAIN
ACCELERATE YOUR GROWTH

Kubernetes ingress is a collection of routing rules that govern how external users access services running in a Kubernetes cluster.



What is an Ingress?

CLOUD TRAIN
ACCELERATE YOUR GROWTH



Installing Ingress Controller

We will be using the nginx ingress controller for our demo. We can download it from the following link:

Link

https://github.com/kubernetes/ingress-nginx/blob/master/docs/deploy/index.md

Defining Ingress Rules

CLOUD TRAIN
ACCELERATE YOUR GROWTH

The following rule, will redirect traffic which asks for /foo to nginx service. All other requests will be redirected to ingress controller's default page.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout-example
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  -http:
      paths:
      - path: /foo
        backend:
          serviceName: nginx
          servicePort: 80
```

Deploying Ingress Rules

CLOUD TRAIN
ACCELERATE YOUR GROWTH

To deploy ingress rules, we use the following syntax:

Syntax

```
kubectl create –f ingress.yaml
```

```
ubuntu@ip-172-31-17-194:~$ kubectl create -f ingress.yaml
ingress.extensions/simple-fanout-example created
ubuntu@ip-172-31-17-194:~$
```

```
ubuntu@slave2:~$ kubectl get svc -n ingress-nginx
NAME                                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
ingress-nginx-controller             NodePort    10.105.151.14    <none>        80:30817/TCP,443:30521/TCP   8m14s
ingress-nginx-controller-admission   ClusterIP   10.102.113.226   <none>        443/TCP                      8m14s
```

Viewing Ingress Rules

To list the ingress rules we use the followingsyntax:

Syntax

```
kubectl get ing
```

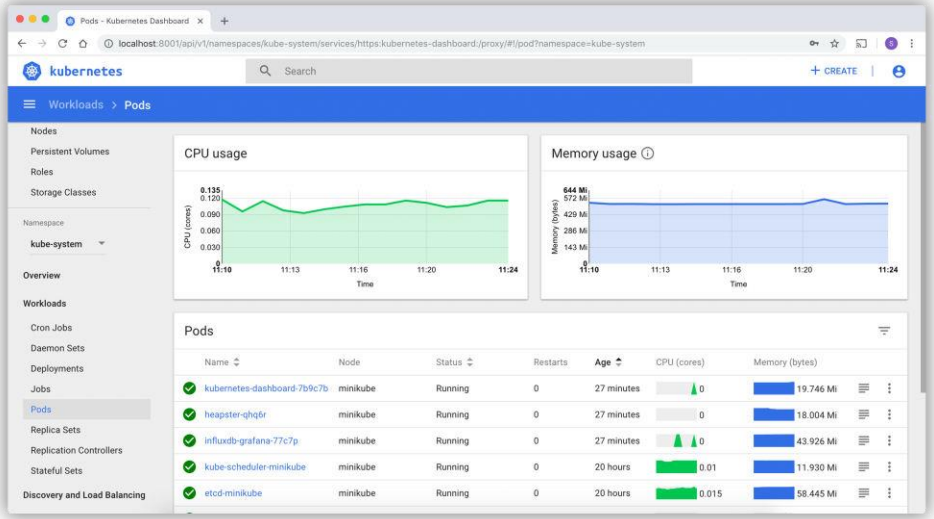
```
ubuntu@ip-172-31-17-194:~$ kubectl get ing
NAME                    HOSTS   ADDRESS   PORTS   AGE
simple-fanout-example   *                 80      2m5s
ubuntu@ip-172-31-17-194:~$
```

Kubernetes Dashboard

Kubernetes Dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application and manage cluster resources.



Installing Kubernetes Dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

To install Kubernetes Dashboard, execute the following command:

Syntax

```
kubectl create -f
https://raw.githubusercontent.com/kubernetes/dashboard/master/aio/deploy/recommended/kubernetes-dashboard.yaml
```

```
ubuntu@ip-172-31-17-194:~$ kubectl create -f https://raw.githubusercontent.com/k
ubernetes/dashboard/master/aio/deploy/recommended/kubernetes-dashboard.yaml
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
serviceaccount/kubernetes-dashboard created
role.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
deployment.apps/kubernetes-dashboard created
service/kubernetes-dashboard created
ubuntu@ip-172-31-17-194:~$
```

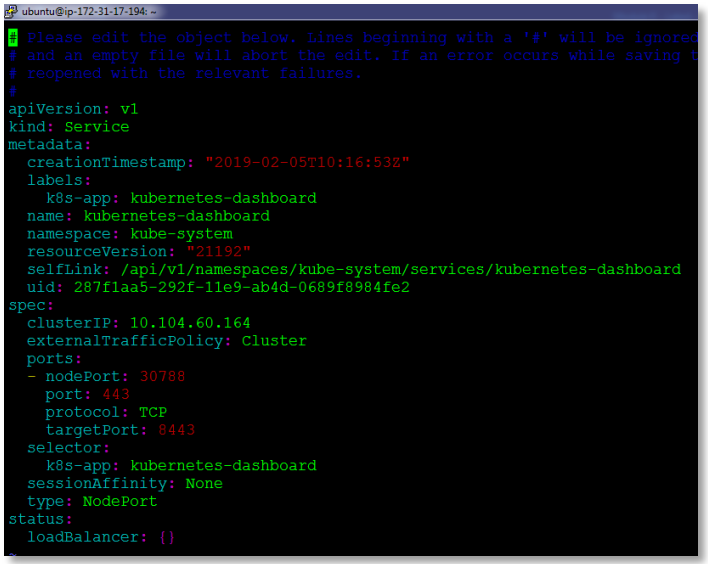
Accessing Kubernetes Dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Change the service type for kubernetes-dashboard to NodePort

Syntax

```
kubectl -n kube-system edit service kubernetes-dashboard
```



Logging into Kubernetes Dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

1. Check the NodePort from the kubernetes-dashboard service
2. Browse to your cluster on the Internet browser, and enter the IP address
3. Click on Token, which will ask you for the token entry
4. Generate a token using the following command:

```
$ kubectl create serviceaccount cluster-admin-dashboard-sa
$ kubectl create clusterrolebinding cluster-admin-dashboard-sa \
  --clusterrole=cluster-admin \
  --serviceaccount=default:cluster-admin-dashboard-sa

$ TOKEN=$(kubectl describe secret $(kubectl -n kube-system get secret | awk '/^cluster-admin-dashboard-sa-token-/{print $1}') | awk '$1=="token:"{print $2}')

$ echo $TOKEN
```

5. Finally, enter the token and login to your dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Hands-on: Deploying an App Using Dashboard

CLOUD TRAIN
ACCELERATE YOUR GROWTH

Got queries or need more info?

Contact us

TO ACCELERATE YOUR CAREER GROWTH

For questions and more details:

please call @ +91 98712 72900 or

visit https://www.thecloudtrain.com/ or

email at join@thecloudtrain.com or

WhatsApp us >>