

## EXPERIMENT NO.10

Aim :- To study and implement the container observation using Kubernetes.

Theory :-

Container orchestration :-

It automatically provisions, deploys, scales and manages containerized applications without worrying about the underlying infrastructure. Developers can implement container orchestration anywhere containers are, allowing them to automate the life cycle management of containers.

Need of Container Orchestration :-

- 1) Reliable application development
- 2) Scalability
- 3) Lower Costs
- 4) Enhanced Security
- 5) High Availability

Kubernetes :-

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem.

Services offered by Kubernetes :

- i) Container Orchestration
- ii) Service Discovery and Load Balancing
- iii) Automated Rollouts and Rollbacks
- iv) Storage Orchestration
- v) Automated Scaling

Kubernetes components :

- i) Control Plane Components
- ii) Kube-apiserver components
- iii) etcd
- iv) Nodes
- v) Image Registry and Pods

Kubernetes is a robust container orchestration platform and offers advanced features. Docker Swarm is a simple and lightweight container orchestration platform, and provides basic features.

Kubernetes follows modular architecture with multiple components, while Docker Swarm has simple architecture with fewer components, including Swarm manager and Swarm agents.

Conclusion :-

P<sup>A</sup>\* Thus, we were able to implement the container orchestration using Kubernetes.

5/3/29

Rishab Mandal  
2103110  
C23

## Cloud Computing Experiment 10

**Aim:** To study and Implement Container Orchestration using Kubernetes.

### Theory:

#### Introduction:

**Container orchestration** is the automation of much of the operational effort required to run containerized workloads and services. This includes a wide range of things software teams need to manage a containers lifecycle, including provisioning, deployment, scaling (up and down), networking, load balancing and more.

Because containers are lightweight and ephemeral by nature, running them in production can quickly become a massive effort. Particularly when paired with microservices—which typically each run in their own containers—a containerized application might translate into operating hundreds or thousands of containers, especially when building and operating any large-scale system.

This can introduce significant complexity if managed manually. Container orchestration is what makes that operational complexity manageable for development and operations—or DevOps—because it provides a declarative way of automating much of the work. This makes it a good fit for DevOps teams and culture, which typically strive to operate with much greater speed and agility than traditional software teams.

### Kubernetes:

The Kubernetes Project was open-sourced by Google in 2014 after using it to run production workloads at scale for more than a decade. Kubernetes provides the ability to run dynamically scaling, containerised applications, and

utilising an API for management. Kubernetes is a vendor-agnostic container management tool, minimising cloud computing costs whilst simplifying the running of resilient and scalable applications.

Kubernetes has become the standard for running containerised applications in the cloud, with the main Cloud Providers (AWS, Azure, GCE, IBM and Oracle) now offering managed Kubernetes services.

## Features Of Kubernetes

This is the right time to talk about Kubernetes features because, you already know what it does and how it compares against Docker Swarm.

### 1. Automatic Binpacking

Kubernetes automatically packages your application and schedules the containers based on their requirements and available resources while not sacrificing availability. To ensure complete utilization and save unused resources, Kubernetes balances between critical and best-effort workloads.

### 2. Service Discovery & Load balancing

With Kubernetes, there is no need to worry about networking and communication because Kubernetes will automatically assign IP addresses to containers and a single DNS name for a set of containers, that can load-balance traffic inside the cluster.

### 3. Storage Orchestration

With Kubernetes, you can mount the storage system of your choice. You can either opt for local storage, or choose a public cloud provider such as GCP or AWS, or perhaps use a shared network storage system such as NFS, iSCSI, etc.

### 4. Self-Healing

Personally, this is my favourite feature. Kubernetes can automatically restart containers that fail during execution and kills those containers that don't respond to user-defined health checks. But if nodes itself die, then it replaces and reschedules those failed containers on other available nodes.

### 5. Secret & Configuration Management

Kubernetes can help you deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.

## 6. Batch Execution

In addition to managing services, Kubernetes can also manage your batch and CI workloads, thus replacing containers that fail, if desired.

## 7. Horizontal Scaling

Kubernetes needs only 1 command to scale up the containers, or to scale them down when using the CLI. Else, scaling can also be done via the Dashboard (Kubernetes UI).

## 8. Automatic Rollbacks & Rollouts

Kubernetes progressively rolls out changes and updates to your application or its configuration, by ensuring that not all instances are worked at the same instance. Even if something goes wrong, Kubernetes will roll back the change for you.

These were some of the notable features of Kubernetes. Let me delve into the attractive aspects of Kubernetes with a real-life implementation of it and how it solved a major industry worry.

## 3. Explain Kubernetes Components, its working and architecture Answer:

Kubernetes lets you create, deploy, manage, and scale application containers across one or more host clusters. Environments running Kubernetes consist of the following key components:

- Kubernetes control plane—manages Kubernetes clusters and the workloads running on them. Include components like the API Server, Scheduler, and Controller Manager.
- Kubernetes data plane—machines that can run containerized workloads. Each node is managed by the kubelet, an agent that receives commands from the control plane.
- Pods—pods are the smallest unit provided by Kubernetes to manage containerized workloads. A pod typically includes several containers, which together form a functional unit or microservice.
- Persistent storage—local storage on Kubernetes nodes is ephemeral, and is deleted when a pod shuts down. This can make it difficult to run stateful applications. Kubernetes provides the Persistent Volumes (PV) mechanism,

allowing containerized applications to store data beyond the lifetime of a pod or node.

A control plane serves as a nerve centre of each Kubernetes cluster. It includes components that can control your cluster, its state data, and its configuration.

The Kubernetes control plane is responsible for ensuring that the Kubernetes cluster attains a desired state, defined by the user in a declarative manner. The control plane interacts with individual cluster nodes using the kubelet, an agent deployed on each node.

Here are the main components of the control plane:

1. kube-apiserver

Provides an API that serves as the front end of a Kubernetes control plane. It is responsible for handling external and internal requests—determining whether a request is valid and then processing it. The API can be accessed via the kubectl command-line interface or other tools like kubeadm, and via REST calls.

2. kube-scheduler

This component is responsible for scheduling pods on specific nodes according to automated workflows and user defined conditions, which can include resource requests, concerns like affinity and taints or tolerations, priority, persistent volumes (PV), and more.

3. kube-controller-manager

The Kubernetes controller manager is a control loop that monitors and regulates the state of a Kubernetes cluster. It receives information about the current state of the cluster and objects within it, and sends instructions to move the cluster towards the cluster operators desired state.

The controller manager is responsible for several controllers that handle various automated activities at the cluster or pod level, including replication controller, namespace controller, service accounts controller, deployment, statefulset, and daemonset.

4. etcd

A key-value database that contains data about your cluster state and configuration. Etcd is fault tolerant and distributed.

## 5. cloud-controller-manager

This component can embed cloud-specific control logic - for example, it can access the cloud providers load balancer service. It enables you to connect a Kubernetes cluster with the API of a cloud provider. Additionally, it helps decouple the Kubernetes cluster from components that interact with a cloud platform, so that elements inside the cluster do not need to be aware of the implementation specifics of each cloud provider.

This cloud-controller-manager runs only controllers specific to the cloud provider. It is not required for on- premises Kubernetes environments. It uses multiple, yet logically-independent, control loops that are combined into one binary, which can run as a single process. It can be used to add scale a cluster by adding more nodes on cloud VMs, and leverage cloud provider high availability and load balancing capabilities to improve resilience and performance.

## Kubernetes Core Components: Worker Nodes

### 6. Nodes

Nodes are physical or virtual machines that can run pods as part of a Kubernetes cluster. A cluster can scale up to 5000 nodes. To scale a clusters capacity, you can add more nodes.

### 7. Pods

A pod serves as a single application instance, and is considered the smallest unit in the object model of Kubernetes. Each pod consists of one or more tightly coupled containers, and configurations that govern how containers should run. To run stateful applications, you can connect pods to persistent storage, using Kubernetes Persistent Volumes—learn more in the following section.

Learn more in our detailed guide to the Kubernetes pod

### 8. Container Runtime Engine

Each node comes with a container runtime engine, which is responsible for running containers. Docker is a popular container runtime engine, but Kubernetes supports other runtimes that are compliant with Open Container Initiative, including CRI-O and rkt.

## 9. kubelet

Each node contains a kubelet, which is a small application that can communicate with the Kubernetes control plane. The kubelet is responsible for ensuring that containers specified in pod configuration are running on a specific node, and manages their lifecycle. It executes the actions commanded by your control plane.

## 10. kube-proxy

All compute nodes contain kube-proxy, a network proxy that facilitates Kubernetes networking services. It handles all network communications outside and inside the cluster, forwarding traffic or replying on the packet filtering layer of the operating system.

## 11. Container Networking

Container networking enables containers to communicate with hosts or other containers. It is often achieved by using the container networking interface (CNI), which is a joint initiative by Kubernetes, Apache Mesos, Cloud Foundry, Red Hat OpenShift, and others.

CNI offers a standardized, minimal specification for network connectivity in containers. You can use the CNI plugin by passing the kubelet --network-plugin=cni command-line option. The kubelet can then read files from --cni-conf-dir and use the CNI configuration when setting up networking for each pod.

## 4. Difference between POD and node? Answer:

### Kubernetes Pods

When you create a Deployment in Kubernetes creates a Pod to host your application instance. A Pod is a Kubernetes abstraction that represents a group of one or more application containers (such as Docker or rkt), and some shared resources for those containers. Those resources include:

- Shared storage, as Volumes
- Networking, as a unique cluster IP address
- Information about how to run each container, such as the container image version or specific ports to use

A Pod models an application-specific “logical host” and can contain different application containers which are relatively tightly coupled. For example, a Pod might include both the container with your Node.js app as well as a different container that feeds the data to be published by the Node.js webserver. The containers in a Pod share an IP Address and port space, are always co-located and co-scheduled, and run in a shared context on the same Node.

Pods are the atomic unit on the Kubernetes platform. When we create a Deployment on Kubernetes, that Deployment creates Pods with containers inside them (as opposed to creating containers directly). Each Pod is tied to the Node where it is scheduled, and remains there until termination (according to restart policy) or deletion. In case of a Node failure, identical Pods are scheduled on other available Nodes in the cluster.

A Pod is a group of one or more application containers (such as Docker or rkt) and includes shared storage (volumes), IP address and information about how to run them.

## Nodes

A Pod always runs on a Node. A Node is a worker machine in Kubernetes and may be either a virtual or a physical machine, depending on the cluster. Each Node is managed by the Master. A Node can have multiple pods, and the Kubernetes master automatically handles scheduling the pods across the Nodes in the cluster. The Masters automatic scheduling takes into account the available resources on each Node.

Every Kubernetes Node runs at least:

- Kubelet, a process responsible for communication between the Kubernetes Master and the Node; it manages the Pods and the containers running on a machine.
- A container runtime (like Docker, rkt) responsible for pulling the container image from a registry, unpacking the container, and running the application.

Containers should only be scheduled together in a single Pod if they are tightly coupled and need to share resources such as disk.

## Node Overview

### 5. Compare Kubernetes and Docker Swarm

Answer:

Although both orchestration tools offer much of the same functionalities, there are fundamental differences between how the two operate. Below listed are some of the most notable points:

- Application definition

In Kubernetes, an application can be deployed using a combination of pods, deployments, and services (or micro-services).

Whereas, in Docker Swarm, applications can be deployed as services (or micro-services) in a Swarm cluster. YAML files can be used to specify multi-container. Moreover, Docker Compose can deploy the app.

- Installation and set-up

In Kubernetes, installation is manual and it takes serious planning to make Kubernetes up and running. Installation instructions differ from OS to OS and provider to provider. Furthermore, in K8s, you need to know the cluster configuration such as IP addresses of a node or which node takes what role in advance.

Compared to Kubernetes, Docker Swarm is simple to install. With Docker, only one set of tools are required to learn to build upon the environment and configuration. Docker Swarm also provides flexibility by allowing any new node to join an existing cluster as either a manager or a worker.

- Working on two systems

You need to know CLI (Command Line Interface) to run Kubernetes on top of Docker. Knowing Docker CLI helps you navigate within a structure and then

supplemental Kubernetes common language infrastructure to run for those programs.

Since Docker Swarm is a tool of Docker, a common language is used to navigate within a structure. This provides variability and speed to this tool and gives Docker a significant usability edge.

- Logging and monitoring

Kubernetes supports multiple versions of logging and monitoring when the services are deployed within the cluster:

- Elasticsearch/Kibana (ELK) logs within the container
- Heapster/Grafana/ Influx for monitoring in the container
- Sysdig cloud integration

Docker Swarm supports monitoring only with third-party applications. It is recommended to use Docker with Reimann for monitoring, however since Docker Swarm has an open API, it makes it easier to connect with plenty of apps.

- Scalability

Kubernetes is more of an all-in-one framework for distributed systems. It is a complex system as it offers a unified set of APIs and strong guarantees about the cluster state, which slows down container deployment and scaling.

Compared to Kubernetes, Docker Swarm can deploy containers faster; this allows fast reaction times to scale on demand.

- High availability

In Kubernetes, all pods are distributed among nodes and this provides high availability by tolerating application failure. In Kubernetes, load-balancing services detect unhealthy pods and remove them, this supports high availability.

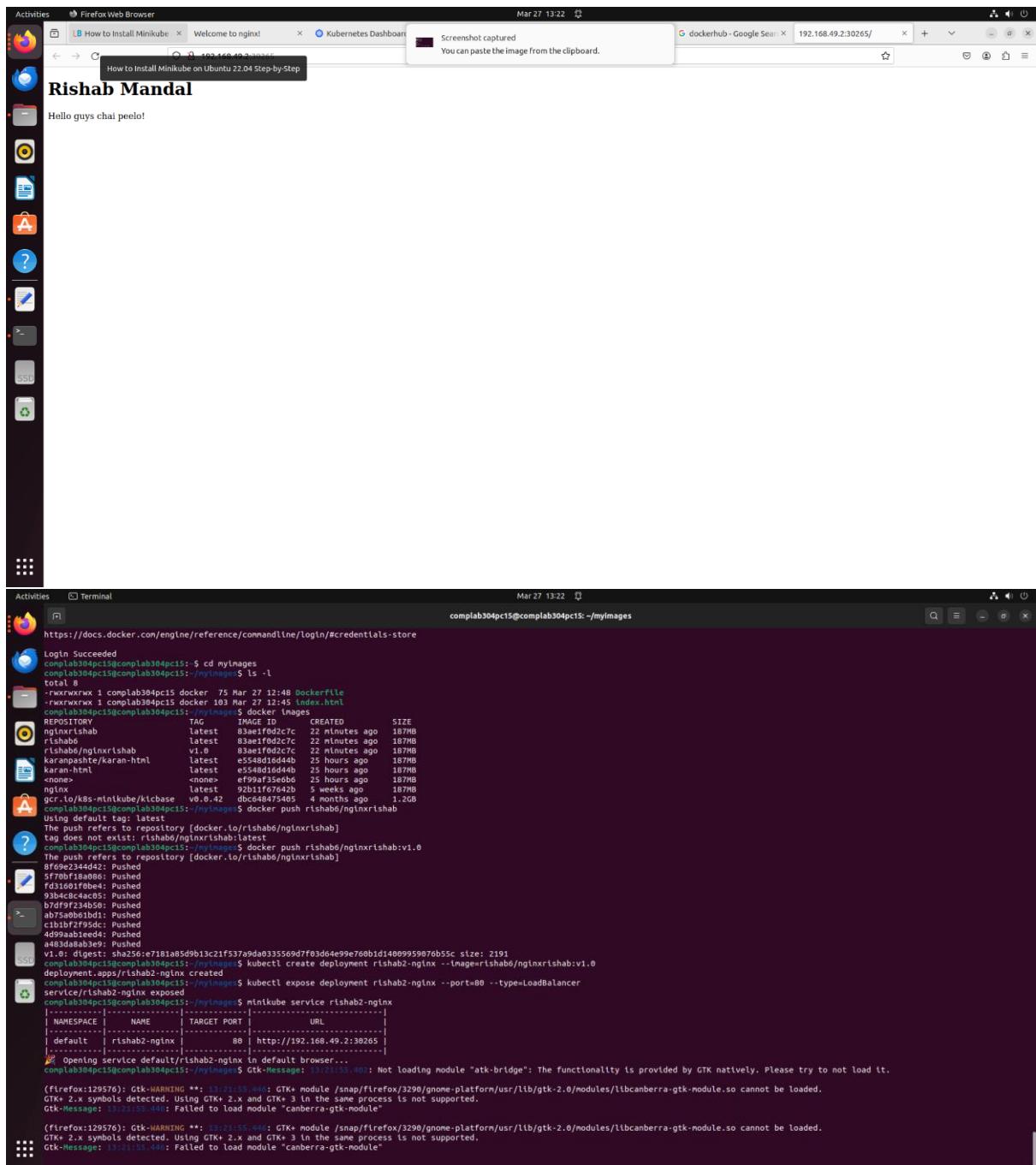
Docker Swarm also provides high availability as services can be replicated in Swarm nodes. In Docker Swarm, Swarm manager nodes are responsible for the whole cluster and manage the resources of worker nodes.

- Networking

Kubernetes network is flat, as it enables all pods to communicate with one another. In Kubernetes, the model requires two CIDRs. The first one requires pods to get an IP address, the other is for services.

In a Docker Swarm, a node joining a cluster creates an overlay network of services that span all of the hosts in the Swarm and a host-only Docker bridge network for containers. In Docker Swarm, users have the option to encrypt container data traffic when creating an overlay network on their own.

### **Screenshots:**



```

Activities Terminal Mar 27 13:20
complab304pc15@complab304pc15: ~/myimages

gcr.io/k8s-minikube/kicbase v0.0.42 dbe648475405 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker image tag 83ae1f0d2c7c rishabh
complab304pc15@complab304pc15:~/myimages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest 83ae1f0d2c7c 10 minutes ago 187MB
rishabh latest 83ae1f0d2c7c 10 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 23 hours ago 187MB
karan-html latest e5548d16d44b 23 hours ago 187MB
nginx <none> ef99af15e6b6 25 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe648475405 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker image tag 83ae1f0d2c7c rishabh6/nginxrishabh:v1.0
complab304pc15@complab304pc15:~/myimages$ 

xterm
complab304pc15@complab304pc15:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/complab304pc15/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
complab304pc15@complab304pc15:~$ cd myimages
complab304pc15@complab304pc15:~/myimages$ ls -l
total 0
complab304pc15@complab304pc15:~/myimages$ docker 75 Mar 27 12:48 Dockerfile
complab304pc15@complab304pc15:~/myimages$ docker 103 Mar 27 12:45 index.html
complab304pc15@complab304pc15:~/myimages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest 83ae1f0d2c7c 10 minutes ago 187MB
rishabh latest 83ae1f0d2c7c 22 minutes ago 187MB
rishabh6/nginxrishabh v1.0 83ae1f0d2c7c 22 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 23 hours ago 187MB
karan-html latest e5548d16d44b 23 hours ago 187MB
nginx <none> ef99af15e6b6 25 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe648475405 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker push rishabh6/nginxrishabh:v1.0
The push refers to repository [docker.io/rishabh6/nginxrishabh]
bf9e9e2344d2: Pushed
3f7a06170be4: Pushed
fd1100170be4: Pushed
934c8c4ac051: Pushed
b7df9f234b50: Pushed
ab75a0b61b1: Pushed
c103f21b2f1: Pushed
4d95aabeed1: Pushed
a483daa9b3: Pushed
V1.0: digest: sha256:e7181a85d9b13cf537a9da033569d7f03d64e99e760b1d140099959076b55c size: 2191
complab304pc15@complab304pc15:~/myimages$ kubectl create deployment rishabh2-nginx --image=rishabh6/nginxrishabh:v1.0
deployment.apps/rishabh2-nginx created
complab304pc15@complab304pc15:~/myimages$ 

Activities Terminal Mar 27 13:15
complab304pc15@complab304pc15: ~/myimages

<none> <none> ef99af15e6b6 25 hours ago 187MB
gcr.io/k8s-minikube/kicbase latest 92b1f67642b 5 weeks ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker image tag 83ae1f0d2c7c rishabh
complab304pc15@complab304pc15:~/myimages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest 83ae1f0d2c7c 10 minutes ago 187MB
rishabh latest 83ae1f0d2c7c 10 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 23 hours ago 187MB
karan-html latest e5548d16d44b 23 hours ago 187MB
nginx <none> ef99af15e6b6 25 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe648475405 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker image tag 83ae1f0d2c7c rishabh6/nginxrishabh:v1.0
complab304pc15@complab304pc15:~/myimages$ 

xterm
complab304pc15@complab304pc15:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/complab304pc15/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
complab304pc15@complab304pc15:~$ cd myimages
complab304pc15@complab304pc15:~/myimages$ ls -l
total 0
complab304pc15@complab304pc15:~/myimages$ docker 75 Mar 27 12:48 Dockerfile
complab304pc15@complab304pc15:~/myimages$ docker 103 Mar 27 12:45 index.html
complab304pc15@complab304pc15:~/myimages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest 83ae1f0d2c7c 22 minutes ago 187MB
rishabh latest 83ae1f0d2c7c 22 minutes ago 187MB
rishabh6/nginxrishabh v1.0 83ae1f0d2c7c 22 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 23 hours ago 187MB
karan-html latest e5548d16d44b 23 hours ago 187MB
nginx <none> ef99af15e6b6 25 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe648475405 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker push rishabh6/nginxrishabh:v1.0
The push refers to repository [docker.io/rishabh6/nginxrishabh]
bf9e9e2344d2: Pushed
3f7a06170be4: Pushed
fd1100170be4: Pushed
934c8c4ac051: Pushed
b7df9f234b50: Pushed
ab75a0b61b1: Pushed
c103f21b2f1: Pushed
4d95aabeed1: Pushed
a483daa9b3: Pushed
V1.0: digest: sha256:e7181a85d9b13cf537a9da033569d7f03d64e99e760b1d140099959076b55c size: 2191
complab304pc15@complab304pc15:~/myimages$ 

```

```

Activities Terminal Mar 27 12:59
complab304pc15@complab304pc15: ~/myimages
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/a
cces-tokens/
Username: rishabh
Password:
WARNING! Your password will be stored unencrypted in /home/complab304pc15/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
complab304pc15@complab304pc15:~/myimages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest b3ae1f0d2c7c 5 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 25 hours ago 187MB
karan-html latest e5548d16d44b 25 hours ago 187MB
<none> <none> ef99af35e6bb 25 hours ago 187MB
nginx latest 92b116f7642b 5 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe64875465 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myimages$ docker push nginxrishabh
Using default tag: latest
The push refers to repository [docker.io/library/nginxrishabh]
bf9e9de2344d: Preparing
5f76bf1a9a86: Preparing
fd11061f0b8e: Preparing
93b4c8c4ac05: Preparing
b7df9f234b50: Preparing
ab75a0b61b1d: Waiting
c1b1b2f295dc: Waiting
4d99aabeed: Waiting
a493d8ab3e9: Waiting
denied: requested access to the resource is denied
complab304pc15@complab304pc15:~/myimages$ ls -l
total 0
GSD
-rw-rw-r-- 1 complab304pc15 docker 75 Mar 27 12:48 Dockerfile
-rw-rw-r-- 1 complab304pc15 docker 103 Mar 27 12:45 index.html
complab304pc15@complab304pc15:~/myimages$ chmod 777 *
complab304pc15@complab304pc15:~/myimages$ ls -l
total 0
-rwxrwxrwx 1 complab304pc15 docker 75 Mar 27 12:48 Dockerfile
-rwxrwxrwx 1 complab304pc15 docker 103 Mar 27 12:45 index.html
complab304pc15@complab304pc15:~/myimages$ docker push nginxrishabh
Using default tag: latest
The push refers to repository [docker.io/library/nginxrishabh]
bf9e9de2344d: Preparing
5f76bf1a9a86: Preparing
fd11061f0b8e: Preparing
93b4c8c4ac05: Preparing
379a0b61b1d1: Waiting
ab75a0b61b1d1: Waiting
c1b1b2f295dc: Waiting
4d99aabeed: Waiting
a493d8ab3e9: Waiting
denied: requested access to the resource is denied
complab304pc15@complab304pc15:~/myimages$ 

Activities Terminal Mar 27 12:57
complab304pc15@complab304pc15: ~/myImages
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY index.html index.html
complab304pc15@complab304pc15:~/myImages$ docker image build -t nginxrishabh -f Dockerfile .
[+] Building 0.4s (8/8) FINISHED
   = [internal] load build definition from Dockerfile
   = [internal] transfer dockerfile: 11B
   = [internal] load metadata for docker.io/library/nginx:latest
   = [internal] load build context
   = [internal] transfer context: 2B
   = [1/1] FROM docker.io/library/nginx:latest
   = [internal] load build context
   = [internal] transfer context: 1kB
   = [2/2] COPY index.html /usr/share/nginx/html
   = [3/3] COPY index.html index.html
   = exporting to image
   = exporting layers
   = writing manifest to /var/tmp/docker-manifests/05bF930ee412041c0853f2297f36dd6eb0B3
complab304pc15@complab304pc15:~/myImages$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/a
cces-tokens/
Username: rishabh
Password:
WARNING! Your password will be stored unencrypted in /home/complab304pc15/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
complab304pc15@complab304pc15:~/myImages$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginxrishabh latest b3ae1f0d2c7c 5 minutes ago 187MB
karanpashte/karan-html latest e5548d16d44b 25 hours ago 187MB
karan-html latest e5548d16d44b 25 hours ago 187MB
<none> <none> ef99af35e6bb 25 hours ago 187MB
nginx latest 92b116f7642b 5 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbe64875465 4 months ago 1.2GB
complab304pc15@complab304pc15:~/myImages$ docker push nginxrishabh
Using default tag: latest
The push refers to repository [docker.io/library/nginxrishabh]
bf9e9de2344d: Preparing
5f76bf1a9a86: Preparing
fd11061f0b8e: Preparing
93b4c8c4ac05: Preparing
b7df9f234b50: Preparing
ab75a0b61b1d: Waiting
c1b1b2f295dc: Waiting
4d99aabeed: Waiting
a493d8ab3e9: Waiting
denied: requested access to the resource is denied
complab304pc15@complab304pc15:~/myImages$ 

```

Activities Terminal Mar 27 12:56

```

Verifying dashboard health ...
Launching proxy ...
Verifying dashboard health ...
HTTP POST http://[::]:37679/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/snap.firefox/_run/user/1000/doc name bind,rw,x-snapd.ignore-missing 0 0): cannot inspect "/run/user/1000/doc"
c1: lstat /run/user/1000/doc: permission denied
Gtk-Message: 12:49:19.473: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.

(firefox:7474): Gtk-WARNING **: 12:49:19.510: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:49:19.516: Failed to load module "canberra-gtk-module"

(firefox:7474): Gtk-WARNING **: 12:49:19.516: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:49:19.516: Failed to load module "canberra-gtk-module"

complab304pc15@complab304pc15:~/myimages
[*] Building 0.4s (8/8) FINISHED
--> [internal] load .dockerignore
--> [internal] load dockerfile: 13B
--> [internal] load metadata for docker.io/library/nginx:latest
--> [internal] transfer context: 2B
--> [internal] load build context
--> [internal] transfer context: 140B
--> [internal] load metadata for docker.io/library/nginx:html
--> [internal] copy index.html index.html
--> [internal] export layers
--> writing image sha256:03ae1fed2c7c6f952235b901890b05bf990ee412845c8653f2297f36dd6eb983
--> naming to docker.io/library/nginx:html
complab304pc15@complab304pc15:~/myimages$ docker login
Log in to your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: rishab
Password: 
WARNING! Your password will be stored unencrypted in /home/complab304pc15/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
complab304pc15@complab304pc15:~/myimages$
```

Activities Firefox Web Browser Mar 27 12:40

Name	Images	Labels	Pods	Created
rishab1-nginx	nginx:latest	app: rishab1-nginx	1 / 1	5 minutes ago
rishab-nginx	nginx:latest	app: rishab-nginx	1 / 1	8 minutes ago

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
rishab1-nginx-7bbf4ddd-7xg4v	nginx:latest	app: rishab1-nginx	pod-template-hash: 7bbf4ddd-7	minikube	Running	0	-	6 minutes ago
rishab-nginx-6e68b8e8-8tvntwu	nginx:latest	app: rishab-nginx	minikube	Running	0	-	-	8 minutes ago

Activities Firefox Web Browser Mar 27 12:40

How to Install Minikube Welcome to nginx! Kubernetes Dashboard Kubernetes Dashboard

127.0.0.1:44023/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/workloads?namespace=default

Kubernetes Workloads

Deployments

Name	Images	Labels	Pods	Created
rishab1-nginx	nginx:latest	app:rishab1-nginx	2 / 2	5.minutes.ago
rishab-nginx	nginx:latest	app:rishab-nginx	1 / 1	7.minutes.ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
rishab1-nginx-7bbbf4dd7-99ww	nginx:latest	app:rishab1-nginx pod-template-hash:7bbbf4dd7	minikube	Running	0	-	-	a minute ago
rishab1-nginx-7bbbf4dd7-7g4lv	nginx:latest	app:rishab1-nginx pod-template-hash:7bbbf4dd7	minikube	Running	0	-	-	5 minutes ago
rishab-nginx-6c6b8d8c8-tvnfw	nginx:latest	app:rishab-nginx pod-template-hash:6c6b8d8c8	minikube	Running	0	-	-	7 minutes ago

Replica Sets

Name	Images	Labels	Pods	Created
------	--------	--------	------	---------

Activities Terminal Mar 27 12:40

```
complab304pc15@complab304pc15:~
```

```
| default | rishab1-nginx | 80 | http://192.168.49.2:31115 |
|-----|-----|-----|-----|
└─ Opening service default/rishab1-nginx in default browser...
complab304pc15@complab304pc15: ~ update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/snap.firefox /run/user/1000/doc none bind,rw,x-snapd.ignore-missing 0 0); cannot inspect "/run/user/1000/doc": lstat /run/user/1000/doc: permission denied
Gtk-Message: 12:34:10.146: Failed to load module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.

(firefox:60144): Gtk-WARNING **: 12:34:10.340: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:34:10.340: Failed to load module "canberra-gtk-module"

(firefox:60144): Gtk-WARNING **: 12:34:10.340: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:34:10.340: Failed to load module "canberra-gtk-module"

complab304pc15@complab304pc15: ~ kubectl scale deployment rishab1-nginx --replicas=2
deployment.apps/rishab1-nginx scaled
complab304pc15@complab304pc15: ~ minikube dashboard
minikube: command not found
complab304pc15@complab304pc15: ~ minikube dashboard
└─ Verifying dashboard health ...
└─ Launching proxy ...
└─ Verifying proxy ...
└─ Opening http://127.0.0.1:44023/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/ in your default browser...
update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/snap.firefox /run/user/1000/doc none bind,rw,x-snapd.ignore-missing 0 0); cannot inspect "/run/user/1000/doc": lstat /run/user/1000/doc: permission denied
Gtk-Message: 12:34:10.446: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.

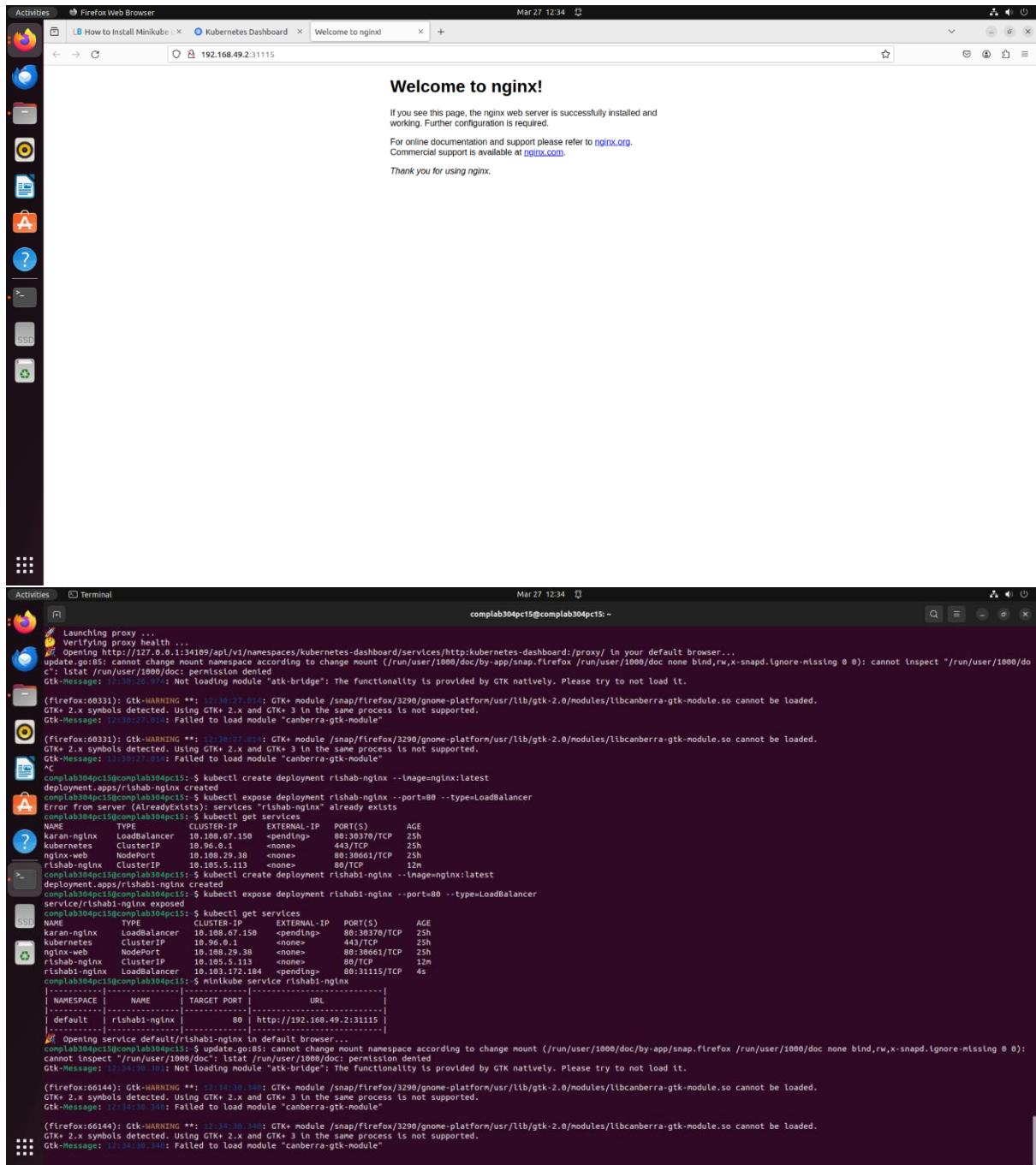
(firefox:73701): Gtk-WARNING **: 12:34:10.500: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:34:10.500: Failed to load module "canberra-gtk-module"

(firefox:73701): Gtk-WARNING **: 12:34:10.500: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:34:10.500: Failed to load module "canberra-gtk-module"

complab304pc15@complab304pc15: ~ kubectl scale deployment rishab1-nginx --replicas=1
deployment.apps/rishab1-nginx scaled
complab304pc15@complab304pc15: ~ minikube dashboard
└─ Verifying dashboard health ...
└─ Launching proxy ...
└─ Verifying proxy health ...
└─ Opening http://127.0.0.1:44023/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/ in your default browser...
update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/snap.firefox /run/user/1000/doc none bind,rw,x-snapd.ignore-missing 0 0); cannot inspect "/run/user/1000/doc": lstat /run/user/1000/doc: permission denied
Gtk-Message: 12:34:10.771: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.

(firefox:74744): Gtk-WARNING **: 12:40:19.510: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:40:19.510: Failed to load module "canberra-gtk-module"

(firefox:74744): Gtk-WARNING **: 12:40:19.510: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:40:19.510: Failed to load module "canberra-gtk-module"
```



```

Activities Terminal Mar 27 12:29 comlab304pc15@comlab304pc15: ~
Gtk-Message: 12:19:00.756: Failed to load module "canberra-gtk-module"
```
comlab304pc15@comlab304pc15: ~$ kubectl expose deployment rishab-nginx --port=80 type=LoadBalancer
Error from server (NotFound): deployments.apps "type=LoadBalancer" not found
comlab304pc15@comlab304pc15: ~$ kubectl expose deployment rishab-nginx --port=80 --type=LoadBalancer
Error from server (AlreadyExists): services "rishab-nginx" already exists
comlab304pc15@comlab304pc15: ~$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
karan-nginx   LoadBalancer 10.108.67.150 <pending>    80:30370/TCP 24h
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   25h
nginx-web     NodePort    10.108.29.38  <none>       80:30661/TCP 25h
rishab-nginx  ClusterIP  10.105.5.113 <none>       80/TCP    49s
comlab304pc15@comlab304pc15: ~$ kubectl get svc
[...]
NAME/SPACE	NAME	TARGET PORT	URL
default	rishab-nginx	No node port	[REDACTED]
-----	-----	-----	-----
? service/default/rishab-nginx has no node port
comlab304pc15@comlab304pc15: ~$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
karan-nginx   LoadBalancer 10.108.67.150 <pending>    80:30370/TCP 24h
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   25h
nginx-web     NodePort    10.108.29.38  <none>       80:30661/TCP 25h
rishab-nginx  ClusterIP  10.105.5.113 <none>       80/TCP    49s
comlab304pc15@comlab304pc15: ~$ kubectl delete deployments karan-nginx
Error from server (NotFound): deployments.apps "karan-nginx" not found
comlab304pc15@comlab304pc15: ~$ kubectl delete deployment nginx-web
deployment.apps "nginx-web" deleted
comlab304pc15@comlab304pc15: ~$ kubectl delete deployment rishab-nginx
deployment.apps "rishab-nginx" deleted
comlab304pc15@comlab304pc15: ~$ kubectl delete deployment karan-nginx
Error from server (NotFound): deployments.apps "karan-nginx" not found
comlab304pc15@comlab304pc15: ~$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
karan-nginx   LoadBalancer 10.108.67.150 <pending>    80:30370/TCP 24h
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   25h
nginx-web     NodePort    10.108.29.38  <none>       80:30661/TCP 25h
rishab-nginx  ClusterIP  10.105.5.113 <none>       80/TCP    49s
comlab304pc15@comlab304pc15: ~$ kubectl get deployments
No resources found in default namespace.
comlab304pc15@comlab304pc15: ~$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
karan-nginx   LoadBalancer 10.108.67.150 <pending>    80:30370/TCP 24h
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   25h
nginx-web     NodePort    10.108.29.38  <none>       80:30661/TCP 25h
rishab-nginx  ClusterIP  10.105.5.113 <none>       80/TCP    49s
comlab304pc15@comlab304pc15: ~$ kubectl get deployment
No resources found in default namespace.
comlab304pc15@comlab304pc15: ~$ kubectl delete deployment karan-nginx
Error from server (NotFound): deployments.apps "karan-nginx" not found
comlab304pc15@comlab304pc15: ~$ ```

Activities Terminal Mar 27 12:22 comlab304pc15@comlab304pc15: ~
```
21m Normal Starting node/minikube Starting kubelet.
21m Normal NodeHasSufficientMemory node/minikube Node minikube status is now: NodeHasSufficientMemory
21m Normal NodeHasNoDiskPressure node/minikube Node minikube status is now: NodeHasNoDiskPressure
21m Normal NodeHasSufficientPID node/minikube Node minikube status is now: NodeHasSufficientPID
21m Normal NodeIsAllocatableEnforced node/minikube Updated Node Allocatable lmt across pods
20m Normal Starting node/minikube
20m Normal RegisteredNode node/minikube Node minikube event: Registered Node minikube in Controller
19m Normal Scheduled pod/nginx-web-5b757f798d-ljft6 Successfully assigned default/nginx-web-5b757f798d-ljft6 to minikube
19m Normal Pulling pod/nginx-web-5b757f798d-ljft6 Pulling image "nginx"
19m Normal Pulled pod/nginx-web-5b757f798d-ljft6 Successfully pulled image "nginx" in 2.014s (2.014s including waiting)
19m Normal Created pod/nginx-web-5b757f798d-ljft6 Created container nginx
19m Normal Started pod/nginx-web-5b757f798d-ljft6 Started container nginx
19m Normal Killing pod/nginx-web-5b757f798d-wf4j6 Stopped container nginx
24h Normal SuccessfulCreate replicaset/nginx-web-5b757f798d-wf4j6 Created pod: nginx-web-5b757f798d-ljft6
19m Normal ScalingReplicaSet deployment/nginx-web-5b757f798d-wf4j6 Successfully assigned default/nginx-web-5b757f798d to 1
22s Normal Scaled pod/nginx-web-5b757f798d-wf4j6 Scaled up replicas set rishab-nginx-6c08bb8dc8-8df8f to 1
22s Normal Scaled pod/nginx-web-5b757f798d-wf4j6 Successfully assigned default/rishab-nginx-6c08bb8dc8-8df8f to 1
22s Normal Pulled pod/rishab-nginx-6c08bb8dc8-8df8f Pulling image "nginx:latest"
22s Normal Created pod/rishab-nginx-6c08bb8dc8-8df8f Created container nginx
22s Normal Started pod/rishab-nginx-6c08bb8dc8-8df8f Started container nginx
22s Normal ScalingReplicaSet deployment/rishab-nginx-6c08bb8dc8-8df8f Created pod: rishab-nginx-6c08bb8dc8-8df8f
22s Normal Scaled up replicas set rishab-nginx-6c08bb8dc8-8df8f Scaled up replicas set rishab-nginx-6c08bb8dc8-8df8f to 1
27s Normal Verifying dashboard health ...
27s Normal Launching proxy ...
27s Normal Verifying proxy ...
27s Normal update /run/user/1000/doc: permission denied
c1: lsat /run/user/1000/doc: permission denied
Gtk-Message: 12:19:00.711: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(GtkFocus@4870): Gtk-WARNING **: 12:19:00.750 : GTK+ module /snap/firefox/3200/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:19:00.750: Failed to load module "canberra-gtk-module"
(GtkFocus@4870): Gtk-WARNING **: 12:19:00.750 : GTK+ module /snap/firefox/3200/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:19:00.750: Failed to load module "canberra-gtk-module"
comlab304pc15@comlab304pc15: ~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-web-5b757f798d-ljft6 1/1   Running   0          19m
rishab-nginx-6c08bb8dc8-8df8f 1/1   Running   0          30s
comlab304pc15@comlab304pc15: ~$ minikube dashboard
? Verifying dashboard health ...
? Launching proxy ...
? Verifying proxy ...
? update /run/user/1000/doc: permission denied
c1: lsat /run/user/1000/doc: permission denied
Gtk-Message: 12:19:00.711: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(GtkFocus@4870): Gtk-WARNING **: 12:19:00.750 : GTK+ module /snap/firefox/3200/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:19:00.750: Failed to load module "canberra-gtk-module"
(GtkFocus@4870): Gtk-WARNING **: 12:19:00.750 : GTK+ module /snap/firefox/3200/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 12:19:00.750: Failed to load module "canberra-gtk-module"
comlab304pc15@comlab304pc15: ~$ kubectl expose deployment rishab-nginx --port=80 type=LoadBalancer
Error from server (NotFound): deployments.apps "type=LoadBalancer" not found
comlab304pc15@comlab304pc15: ~$ kubectl expose deployment rishab-nginx --port=80 --type=LoadBalancer
Error from server (AlreadyExists): services "rishab-nginx" already exists
comlab304pc15@comlab304pc15: ~$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
karan-nginx   LoadBalancer 10.108.67.150 <pending>    80:30370/TCP 24h
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   25h
nginx-web     NodePort    10.108.29.38  <none>       80:30661/TCP 25h
rishab-nginx  ClusterIP  10.105.5.113 <none>       80/TCP    49s
comlab304pc15@comlab304pc15: ~$ ```


```

Activities Firefox Web Browser Mar 27 12:19

How to Install Minikube | Kubernetes Dashboard +

127.0.0.1:45481/api/v1/namespaces/kubernetes-dashboard/services/httpkubernetes-dashboard/proxy/#/deployment/default/rishab-nginx?namespace=default

**kubernetes** default Search +

Workloads > Deployments > rishab-nginx

**Workloads**

- Cron Jobs
- Daemon Sets
- Deployments**
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets
- Service
- Ingresses
- Ingress Classes
- Services
- Config and Storage
- Config Maps
- Persistent Volume Claims
- Secrets
- Storage Classes
- Cluster
- Cluster Role Bindings
- Cluster Roles
- Events
- Namespaces
- Network Policies

**Metadata**

Name: rishab-nginx Namespace: default Created: Mar 27, 2024 Age: a minute ago UID: 209db6d4-57ee-48a7-a4c2-b0cc1e8dca65

Labels: app: rishab-nginx Annotations: deployment.kubernetes.io/revision: 1

**Resource information**

Strategy: RollingUpdate Min ready seconds: 0 Revision history limit: 10

Selector: app: rishab-nginx

**Rolling update strategy**

Max surge: 25% Max unavailable: 25%

**Pods status**

Updated	Total	Available
1	1	1

Activities Firefox Web Browser Mar 27 12:19

How to Install Minikube | Kubernetes Dashboard +

127.0.0.1:45481/api/v1/namespaces/kubernetes-dashboard/services/httpkubernetes-dashboard/proxy/#/workloads?namespace=default

**kubernetes** default Search +

Workloads

**Workloads**

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets
- Service
- Ingresses
- Ingress Classes
- Services
- Config and Storage
- Config Maps
- Persistent Volume Claims
- Secrets
- Storage Classes
- Cluster
- Cluster Role Bindings
- Cluster Roles
- Events
- Namespaces
- Network Policies

**Workload Status**

Running: 2 Deployments

Running: 2 Pods

Running: 2 Replica Sets

**Deployments**

Name	Images	Labels	Pods	Created
rishab-nginx	nginx:latest	app: rishab-nginx	1 / 1	a minute ago
nginx-web	nginx	app: nginx-web	1 / 1	20 minutes ago

**Pods**

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
rishab-nginx-6c6b8d8c8-8dfnw	nginx:latest	app: rishab-nginx pod-template-hash: 6c6b8d8c8	minikube	Running	0	-	-	a minute ago
nginx-web-5h757f7984lif6	nginx	app: nginx-web	minikube	Running	0	-	-	20 minutes ago

```
Activities Terminal Mar 27 12:18
complab304pc15@complab304pc15: ~
24h Normal Pulling pod/karan1-nginx-7fcd65db-tszrt Pulling image "karan-nginx"
24h Warning Failed pod/karan1-nginx-7fcd65db-tszrt Failed to pull image "karan-nginx": Error response from daemon: pull access denied for karan-nginx, repository does not exist or may require 'docker login': denied: requested access is denied
24h Normal Backoff pod/karan1-nginx-7fcd65db-tszrt Back-off pulling image "karan-nginx"
24h Warning Failed pod/karan1-nginx-7fcd65db-tszrt Error: ErrImagePull
24h Normal SuccessfulCreate replicaset/karan1-nginx-7fcd65db-tszrt Error: ImagePullBackoff
24h Normal Replicated replicaset/karan1-nginx-7fcd65db-tszrt Created pod: karan1-nginx-7fcd65db-kqj7
24h Normal SuccessfulDelete replicaset/karan1-nginx-7fcd65db-tszrt Created pod: karan1-nginx-7fcd65db-kqj7
24h Normal ScaledDown deployment/karan1-nginx-7fcd65db-tszrt Scaled down replica set karan1-nginx-7fcd65db to 2
24h Normal ScalingReplicaSet deployment/karan1-nginx-7fcd65db-tszrt Scaled down replica set karan1-nginx-7fcd65db to 1 from 2
24h Normal Scheduled pod/karan2-nginx-74fc488bdf-hpqts Successfully assigned default/karan2-nginx-74fc488bdf-hpqts to minikube
24h Normal Pulling pod/karan2-nginx-74fc488bdf-hpqts Pull Image "nginx" in 0.001s (0.001s including waiting)
24h Normal Pulled pod/karan2-nginx-74fc488bdf-hpqts Successfully pulled image "nginx" in 1.982s (1.982s including waiting)
24h Normal Created pod/karan2-nginx-74fc488bdf-hpqts Created container nginx
24h Normal Started pod/karan2-nginx-74fc488bdf-hpqts Started container nginx
24h Normal Killing pod/karan2-nginx-74fc488bdf-hpqts Stopping container nginx
24h Normal Scheduled pod/karan2-nginx-74fc488bdf-wdjlt Successfully assigned default/karan2-nginx-74fc488bdf-wdjlt to minikube
24h Normal Pulled pod/karan2-nginx-74fc488bdf-wdjlt Successfully pulled image "nginx" in 1.983s (3.958s including waiting)
24h Normal Created pod/karan2-nginx-74fc488bdf-wdjlt Created container nginx
24h Normal Started pod/karan2-nginx-74fc488bdf-wdjlt Started container nginx
24h Normal Killing pod/karan2-nginx-74fc488bdf-wdjlt Stopping container nginx
24h Normal SuccessfulCreate replicaset/karan2-nginx-74fc488bdf-wdjlt Created pod: karan2-nginx-74fc488bdf-wdjlt
24h Normal SuccessDelete replicaset/karan2-nginx-74fc488bdf-wdjlt Deleted pod: karan2-nginx-74fc488bdf-wdjlt
24h Normal ScalingReplicaSet deployment/karan2-nginx-7fcd65db-tszrt Scaled up replica set karan2-nginx-74fc488bdf to 2
24h Normal ScalingReplicaSet deployment/karan2-nginx-7fcd65db-tszrt Scaled down replica set karan2-nginx-74fc488bdf to 1 from 2
24h Normal Starting node/minikube Starting Node minikube
21m Normal NodeHasSufficientMemory node/minikube Node minikube status is now: NodeHasSufficientMemory
21m Normal NodeHasNoDiskPressure node/minikube Node minikube status is now: NodeHasNoDiskPressure
21m Normal NodeHasSufficientPID node/minikube Node minikube status is now: NodeHasSufficientPID
21m Normal NodeAllocatableEnforced node/minikube Updated Node Allocatable limit across pods
20m Normal RegisteredNode node/minikube Node minikube event: Registered Node minikube in Controller
20m Normal Scheduled pod/nginx-web-5b757f798d-ljft6 Successfully assigned default/nginx-web-5b757f798d-ljft6 to minikube
19m Normal Pulling pod/nginx-web-5b757f798d-ljft6 Pulling image "nginx"
19m Normal Pulled pod/nginx-web-5b757f798d-ljft6 Successfully pulled image "nginx" in 2.014s (2.014s including waiting)
19m Normal Created pod/nginx-web-5b757f798d-ljft6 Created container nginx
19m Normal Started pod/nginx-web-5b757f798d-wf416 Started container nginx
19m Normal Killing pod/nginx-web-5b757f798d-wf416 Stopping container nginx
19m Normal SuccessfulCreate replicaset/nginx-web-5b757f798d-wf416 Created pod: nginx-web-5b757f798d-wf416
19m Normal ScalingReplicaSet deployment/nginx-web-5b757f798d-wf416 Scaled up replica set nginx-web-5b757f798d-wf416 to 1
27s Normal Scheduled pod/rishab-nginx-6c68bb08c8-8dfrw Successfully assigned default/rishab-nginx-6c68bb08c8-8dfrw to minikube
26s Normal Pulling pod/rishab-nginx-6c68bb08c8-8dfrw Pulling image "nginx:latest"
22s Normal Pulled pod/rishab-nginx-6c68bb08c8-8dfrw Successfully pulled image "nginx:latest" in 4.313s (4.313s including waiting)
22s Normal Created pod/rishab-nginx-6c68bb08c8-8dfrw Created container nginx
22s Normal Started pod/rishab-nginx-6c68bb08c8-8dfrw Started container nginx
27s Normal SuccessfulCreate replicaset/rishab-nginx-6c68bb08c8-8dfrw Created pod: rishab-nginx-6c68bb08c8-8dfrw
27s Normal ScalingReplicaSet deployment/rishab-nginx-6c68bb08c8-8dfrw Scaled up replica set rishab-nginx-6c68bb08c8-8dfrw
complab304pc15@complab304pc15: ~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-web-5b757f798d-ljft6 1/1 Running 0 19m
rishab-nginx-6c68bb08c8-8dfrw 1/1 Running 0 30s
complab304pc15@complab304pc15: ~
```

```
Activities Terminal Mar 27 12:17
complab304pc15@complab304pc15: ~
24h Normal ScalingReplicaSet deployment/karan1-nginx-7fcd65db-tszrt Scaled up replica set karan1-nginx-7fcd65db to 2
24h Normal ScalingReplicaSet deployment/karan1-nginx-74fc488bdf-hpqts Scaled down replica set karan1-nginx-7fcd65db to 1 from 2
24h Normal Scheduled pod/karan2-nginx-74fc488bdf-hpqts Successfully assigned default/karan2-nginx-74fc488bdf-hpqts to minikube
24h Normal Pulling pod/karan2-nginx-74fc488bdf-hpqts Pulling image "nginx"
24h Normal Pulled pod/karan2-nginx-74fc488bdf-hpqts Successfully pulled image "nginx" in 1.982s (1.982s including waiting)
24h Normal Created pod/karan2-nginx-74fc488bdf-hpqts Created container nginx
24h Normal Started pod/karan2-nginx-74fc488bdf-hpqts Started container nginx
24h Normal Killing pod/karan2-nginx-74fc488bdf-hpqts Stopping container nginx
24h Normal Scheduled pod/karan2-nginx-74fc488bdf-wdjlt Successfully assigned default/karan2-nginx-74fc488bdf-wdjlt to minikube
24h Normal Pulling pod/karan2-nginx-74fc488bdf-wdjlt Pulling image "nginx"
24h Normal Pulled pod/karan2-nginx-74fc488bdf-wdjlt Successfully pulled image "nginx" in 1.983s (3.958s including waiting)
24h Normal Created pod/karan2-nginx-74fc488bdf-wdjlt Created container nginx
24h Normal Started pod/karan2-nginx-74fc488bdf-wdjlt Started container nginx
24h Normal Killing pod/karan2-nginx-74fc488bdf-wdjlt Stopping container nginx
24h Normal SuccessfulCreate replicaset/karan2-nginx-74fc488bdf-wdjlt Created pod: karan2-nginx-74fc488bdf-wdjlt
24h Normal SuccessDelete replicaset/karan2-nginx-74fc488bdf-wdjlt Deleted pod: karan2-nginx-74fc488bdf-wdjlt
24h Normal ScalingReplicaSet deployment/karan2-nginx-7fcd65db-tszrt Scaled up replica set karan2-nginx-74fc488bdf to 2
24h Normal ScalingReplicaSet deployment/karan2-nginx-7fcd65db-tszrt Scaled down replica set karan2-nginx-74fc488bdf to 1 from 2
24h Normal Starting node/minikube Starting Node minikube
16m Normal NodeHasSufficientMemory node/minikube Node minikube status is now: NodeHasSufficientMemory
16m Normal NodeHasNoDiskPressure node/minikube Node minikube status is now: NodeHasNoDiskPressure
16m Normal NodeHasSufficientPID node/minikube Node minikube status is now: NodeHasSufficientPID
16m Normal NodeAllocatableEnforced node/minikube Updated Node Allocatable limit across pods
16m Normal RegisteredNode node/minikube Node minikube event: Registered Node minikube in Controller
15m Normal Scheduled pod/nginx-web-5b757f798d-ljft6 Successfully assigned default/nginx-web-5b757f798d-ljft6 to minikube
15m Normal Pulling pod/nginx-web-5b757f798d-ljft6 Pulling image "nginx"
15m Normal Pulled pod/nginx-web-5b757f798d-ljft6 Successfully pulled image "nginx" in 2.014s (2.014s including waiting)
15m Normal Created pod/nginx-web-5b757f798d-ljft6 Created container nginx
15m Normal Started pod/nginx-web-5b757f798d-wx316 Started kubelet
15m Normal Killing pod/nginx-web-5b757f798d-wx316 Stopping container nginx
15m Normal SuccessfulCreate replicaset/nginx-web-5b757f798d-wx316 Created pod: nginx-web-5b757f798d-wx316
15m Normal ScalingReplicaSet deployment/nginx-web-5b757f798d-wx316 Scaled up replica set nginx-web-5b757f798d to 1
complab304pc15@complab304pc15: ~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-web-5b757f798d-ljft6 1/1 Running 0 19m
complab304pc15@complab304pc15: ~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
karanpashte/karan-html latest e554bd1d4b 24 hours ago 187MB
karan-html latest e554bd1d4b 24 hours ago 187MB
node:latest <none> e199af15e0b0 24 hours ago 187MB
nginx latest 92b1bf67e62b 5 weeks ago 187MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbc64875405 4 months ago 1.2GB
complab304pc15@complab304pc15: ~$ docker pull nginx
latest: Pulling from library/nginx
Digest: sha256:6d9391d1c1cf6b3f958b0abf72ea999404f2764febfb0f1f96acd5867ac7ef7a
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
complab304pc15@complab304pc15: ~$ kubectl create deployment rishab-nginx
error: --target-flip-label "not set"
complab304pc15@complab304pc15: ~$ kubectl create deployment rishab-nginx --image=nginx:latest
deployment.apps/rishab-nginx created
complab304pc15@complab304pc15: ~
```

```

Activities Terminal Mar 27 12:13
complab304pc15@complab304pc15: ~ minikube status
minikube Ready control-plane 25h v1.28.3
host: Running
kubelet: Running
apiserver: Running
kubeconfig: configured

complab304pc15@complab304pc15: ~ minikube status
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 25h v1.28.3
complab304pc15@complab304pc15: ~ minikube get events
error: the server doesn't support resource type "deployments"
complab304pc15@complab304pc15: ~ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-web 1/1 1 1 15s
complab304pc15@complab304pc15: ~ kubectl get events
LAST SEEN TYPE REASON OBJECT MESSAGE
24h Normal Scheduled pod/karan-nginx-585f9dbbb-nn6b7 Successfully assigned default/karan-nginx-585f9dbbb-nn6b7 to minikube
24h Normal Pulling pod/karan-nginx-585f9dbbb-nn6b7 Pulling image "nginx:latest"
24h Normal Pulled pod/karan-nginx-585f9dbbb-nn6b7 Successfully pulled image "nginx:latest" in 2.009s (2.009s including waiting)
24h Normal Created pod/karan-nginx-585f9dbbb-nn6b7 Created pod karan-nginx
24h Normal Started pod/karan-nginx-585f9dbbb-nn6b7 Started Container nginx
24h Normal Killing pod/karan-nginx-585f9dbbb-nn6b7 Stopping container nginx
24h Normal SuccessfulCreate replicaset/karan-nginx-585f9dbbb-deployment/karan-nginx-7f5cd65db-k6j7 Created pod: karan-nginx-585f9dbbb-nn6b7
24h Normal ScalingReplicaSet deployment/karan-nginx-7f5cd65db-k6j7 Scaled up replica set karan-nginx-585f9dbbb to 1
24h Normal Scheduled pod/karan-nginx-7f5cd65db-k6j7 Successfully assigned default/karan-nginx-7f5cd65db-k6j7 to minikube
24h Normal Pulling pod/karan-nginx-7f5cd65db-k6j7 Pulling image "karan-nginx"
24h Warning Failed pod/karan-nginx-7f5cd65db-k6j7 Failed to pull image "karan-nginx": Error response from daemon: pull access denied for karan-nginx, repository does not exist or may require docker login: denied: requested access to the resource is denied
24h Warning Failed pod/karan-nginx-7f5cd65db-k6j7 Error: ErrImagePull
24h Normal BackOff pod/karan-nginx-7f5cd65db-k6j7 Back-off pulling image "karan-nginx"
24h Normal Started pod/karan-nginx-7f5cd65db-k6j7 Error: ImagePullBackoff
24h Normal Scheduled pod/karan-nginx-7f5cd65db-ts2rt Successfully assigned default/karani-nginx-7f5cd65db-ts2rt to minikube
24h Normal Pulling pod/karan-nginx-7f5cd65db-ts2rt Pulling image "karan-nginx"
24h Warning Failed pod/karan-nginx-7f5cd65db-ts2rt Failed to pull image "karan-nginx": Error response from daemon: pull access denied for karan-nginx, repository does not exist or may require docker login: denied: requested access to the resource is denied
24h Normal BackOff pod/karan-nginx-7f5cd65db-ts2rt Back-off pulling image "karan-nginx"
24h Warning Failed pod/karan-nginx-7f5cd65db-ts2rt Error: ImagePullBackoff
24h Normal SuccessfulCreate replicaset/karani-nginx-7f5cd65db-ts2rt Created pod: karani-nginx-7f5cd65db-ts2rt
24h Normal SuccessfulCreate replicaset/karani-nginx-7f5cd65db-ts2rt Created pod: karani-nginx-7f5cd65db-ts2rt
24h Normal Normal replicaset/karani-nginx-7f5cd65db-ts2rt Replicating pod: karani-nginx-7f5cd65db-ts2rt
24h Normal ScalingReplicaSet deployment/karani-nginx-7f5cd65db-ts2rt Scaled up replica set karani-nginx-7f5cd65db to 2
24h Normal ScalingReplicaSet deployment/karani-nginx-7f5cd65db-ts2rt Scaled down replica set karani-nginx-7f5cd65db to 1 from 2
24h Normal Scheduled pod/karani-nginx-74fc488bdf-hpqts Successfully assigned default/karanz-nginx-74fc488bdf-hpqts to minikube
24h Normal Pulling pod/karani-nginx-74fc488bdf-hpqts Pulling image "nginx"
24h Normal Normal pod/karani-nginx-74fc488bdf-hpqts Successfully pulled image "nginx" in 1.982s (1.982s including waiting)
24h Normal Created pod/karani-nginx-74fc488bdf-hpqts Created container nginx
24h Normal Started pod/karani-nginx-74fc488bdf-hpqts Started container nginx
24h Normal Killing pod/karani-nginx-74fc488bdf-hpqts Stopping container nginx
24h Normal Scheduled pod/karani-nginx-74fc488bdf-wdjl7 Successfully assigned default/karanz-nginx-74fc488bdf-wdjl7 to minikube
24h Normal Normal pod/karani-nginx-74fc488bdf-wdjl7 Pulling image "nginx"
24h Normal Normal pod/karani-nginx-74fc488bdf-wdjl7 Successfully pulled image "nginx" in 1.98s (3.958s including waiting)
24h Normal Created pod/karani-nginx-74fc488bdf-wdjl7 Created container nginx

Activities Terminal Mar 27 12:01
complab304pc15@complab304pc15: ~ minikube addons enable dashboard
Dashboard is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
■ Using Image docker.io/kubernetes/dashboard:v2.7.8
■ Using Image docker.io/kubernetes/metRICS-scraperv1.0.8
Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server

★ The 'dashboard' addon is enabled
complab304pc15@complab304pc15: ~ minikube dashboard
Verifying dashboard health ...
Verifying proxy health ...
Opening http://127.0.0.1:33107/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
update.go:85: cannot change mount namespace according to change mount '/run/user/1000/doc/by-app/snap.firefox /run/user/1000/doc none bind,rw,x.snapd.ignore-missing 0 0': cannot inspect "/run/user/1000/doc": permission denied
c5: lstat /run/user/1000/doc: permission denied
GTK-Message: 12:01:10.450: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to load it.
(Gtk:20782): Gtk-WARNING **: 12:01:10.495: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
(Gtk:20782): Gtk-WARNING **: 12:01:10.495: GTK+ module /snap/firefox/3290/gnome-platform/usr/lib/gtk-2.0/modules/libcanberra-gtk-module.so cannot be loaded.
GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
(Gtk:20782): Gtk-WARNING **: 12:01:10.495: Failed to load module "canberra-gtk-module"

```

```

Activities Terminal Mar 27 11:59
complab304pc15@complab304pc15: ~

NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/nginx-web 1/1 1 1 20s
pod/nginx-web-5b757f79d-ljfts 1/1 Running 0 26s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/karan-nginx LoadBalancer 10.108.97.150 <pending> 80:30378/TCP 24h
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 24h
service/nginx-web NodePort 10.108.29.38 <none> 80:30661/TCP 24h

complab304pc15@complab304pc15: $ minikube addons list

+-----+-----+-----+-----+
| ADDON NAME | PROFILE | STATUS | MAINTAINER |
+-----+-----+-----+-----+
| ambassador | minikube | disabled | 3rd party (Ambassador) |
| auto-pause | minikube | disabled | minikube |
| cloud-spanner | minikube | disabled | Google |
| csi-hostpath-driver | minikube | disabled | Kubernetes |
| default-storageclass | minikube | enabled | Kubernetes |
| efk | minikube | disabled | 3rd party (Elastic) |
| freshpod | minikube | disabled | Google |
| gcp-auth | minikube | disabled | Google |
| insoker | minikube | disabled | minikube |
| headlamp | minikube | disabled | 3rd party (kinvolk.io) |
| helm-tiller | minikube | disabled | 3rd party (Helm) |
| inacel | minikube | disabled | 3rd party (Inacel) [lnfgoinacel.com] |
| ingress | minikube | enabled | Kubernetes |
| ingress-dns | minikube | disabled | minikube |
| inspektoR-gadget | minikube | disabled | 3rd party |
| istio | minikube | disabled | 3rd party (Istio) |
| istio-provisioner | minikube | disabled | 3rd party (Istio) |
| kong | minikube | disabled | 3rd party (Kong HQ) |
| kubeflow | minikube | disabled | 3rd party |
| kubevirt | minikube | disabled | 3rd party (Kubevirt) |
| logviewer | minikube | disabled | 3rd party (Unknown) |
| metalLB | minikube | disabled | 3rd party (MetalLB) |
| metrics-server | minikube | disabled | Kubernetes |
| nvidia-device-plugin | minikube | disabled | 3rd party (NVIDIA) |
| nvidia-driver-installer | minikube | disabled | 3rd party (Nvidia) |
| nvidia-gpu-device-plugin | minikube | disabled | 3rd party (Nvidia) |
| openebs | minikube | disabled | 3rd party (OpenEBS framework) |
| pod-security-policy | minikube | disabled | 3rd party (Unknown) |
| portainer | minikube | disabled | 3rd party (Portainer.io) |
| registry | minikube | disabled | minikube |
| registry-aliases | minikube | disabled | 3rd party (Unknown) |
| rke2 | minikube | disabled | 3rd party (RKE2 Enterprises) |
| storage-provisioner | minikube | enabled | minikube |
| storage-provisioner-gluster | minikube | disabled | 3rd party (Gluster) |
| storage-provisioner-rancher | minikube | disabled | 3rd party (Rancher) |
| volumesnapshots | minikube | disabled | Kubernetes |
+-----+-----+-----+-----+
complab304pc15@complab304pc15: $

Activities Terminal Mar 27 11:59
Screenshot captured
You can paste the image from the clipboard.

complab304pc15@complab304pc15: $ minikube start --driver=docker
minikube v1.32.0 on Ubuntu 22.04
Using the docker driver based on existing profile
Starting control plane node minikube in cluster minikube
Plugin based provisioning...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
  ▓█████  Using Image registry.k8s.io/ingress-nginx/controller:v1.9.4
  ▓█████  Using Image registry.k8s.io/ingress-nginx/defaultbackend:v2.7.0
  ▓█████  Using Image gcr.io/k8s-minikube/storage-provisioner:v5
  ▓█████  Using Image docker.io/kuberneetsu/metrics-scraper:v1.0.8
  ▓█████  Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabef0
  ▓█████  Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabef0
Some dashboard features require the metrics-server addon. To enable all features please run:
  complab304pc15@complab304pc15: $ minikube addons enable metrics-server

  └─? Enabled addons: storage-provisioner, default-storageclass, dashboard, ingress
Done! kubelet is now configured to use "minikube" cluster and "default" namespace by default
complab304pc15@complab304pc15: $ minikube status
minikube
  ▓█████  Node Control Plane
  ▓█████  host: Running
  ▓█████  kubelet: Running
  ▓█████  apiserver: Running
  ▓█████  kubeconfig: Configured
  ▓█████  coreDNS is running at https://192.168.49.2:8443
  ▓█████  To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
complab304pc15@complab304pc15: $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 24h v1.28.3
complab304pc15@complab304pc15: $ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
coreDNS is running at https://192.168.49.2:8443/api/v3/namespaces/kube-system/services/kube-dns:proxy
complab304pc15@complab304pc15: $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 24h v1.28.3
complab304pc15@complab304pc15: $ kubectl cluster-info dump
complab304pc15@complab304pc15: $ kubectl create deployment nginx-web --image=nginx
deployment/nginx-web created
complab304pc15@complab304pc15: $ kubectl expose deployment nginx-web --type=NodePort --port=80
Error from server (AlreadyExists): services "nginx-web" already exists
complab304pc15@complab304pc15: $ kubectl get deployment,pod,svc
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/nginx-web 1/1 1 1 20s
pod/nginx-web-5b757f79d-ljfts 1/1 Running 0 26s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/karan-nginx LoadBalancer 10.108.67.150 <pending> 80:30378/TCP 24h
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 24h
service/nginx-web NodePort 10.108.29.38 <none> 80:30661/TCP 24h

complab304pc15@complab304pc15: $

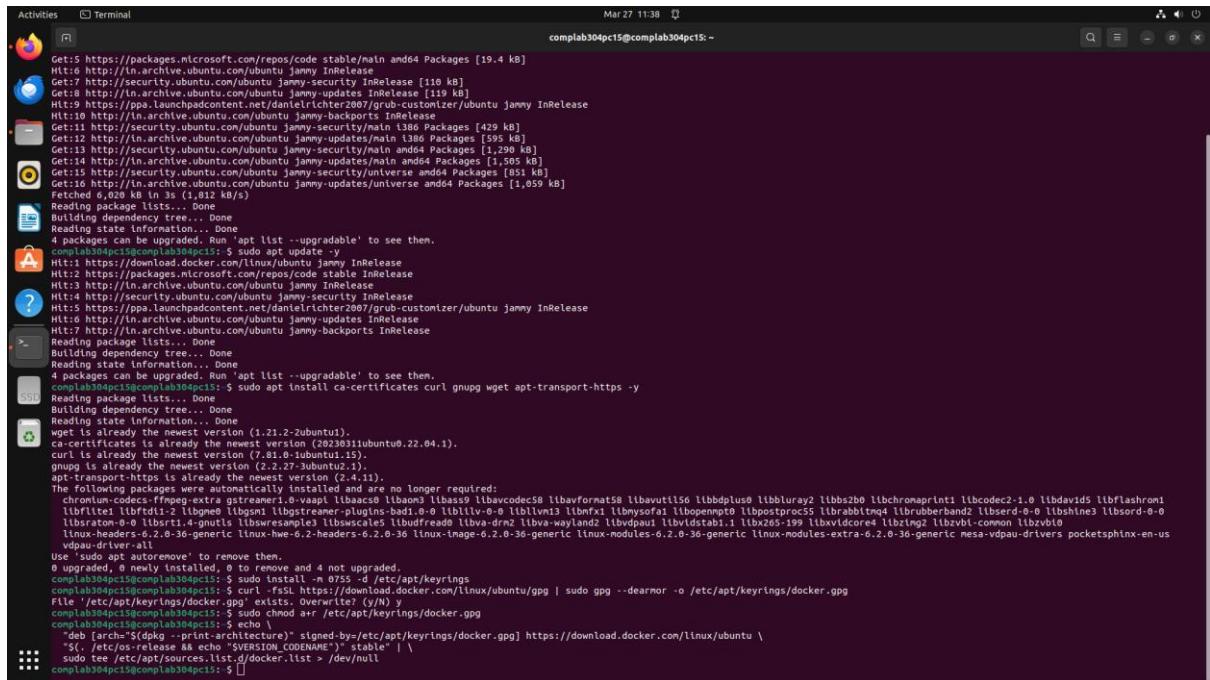
```

```

Activities Terminal Mar 27 11:59
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ minikube start --driver=docker
  minikube v22.0.0 on Ubuntu 22.04
  * Using Docker driver based on existing profile
  * Starting control plane node minikube in cluster minikube
    Pulling base image ...
    Restarting existing docker container for "minikube" ...
    Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
      Configuring bridge CNI (container Networking Interface) ...
      Verifying Kubernetes components ...
        Using Image registry.k8s.io/ingress-nginx/controller:v1.9.4
        Using Image docker.io/kubernetessu/dashboard:v2.7.8
        Using Image gcr.io/k8s-minikube/storage-provisioner:v5
        Using Image registry.k8s.io/kube-webhook-certgen:v20231011-8b53cabe0
        Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabe0
      Verifying ingress addon...
      Some dashboard features require the metrics-server addon. To enable all features please run:
        minikube addons enable metrics-server
      └─ Enabled addons: storage-provisioner, default-storageclass, dashboard, ingress
        Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ minikube status
minikube
  type: Control Plane
  host: Running
  kubelet: Running
  apiserver: Running
  kubeconfig: configured
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane,master 24h v1.28.3
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl create deployment nginx-web --image=nginx
deployment.apps/nginx-web created
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl expose deployment nginx-web --type NodePort --port=80
Error from server (AlreadyExists): services "nginx-web" already exists
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl get deployment,pod,svc
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-web   1/1     1           1           26s
pod/nginx-web-5b757f798d-ljftd   1/1     Running     0           26s
NAME          TYPE    CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/karan-nginx   LoadBalancer   10.108.67.150 <pending>   80:30378/TCP   24h
service/kubernetes   ClusterIP   10.96.0.1    <none>       443/TCP   24h
service/nginx-web   NodePort    10.108.29.38 <none>       80:30661/TCP   24h
complab304pc15@complab304pc15: ~
Activities Terminal Mar 27 11:57
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ minikube version
minikube version: v1.32.0
commit: 8220a6e6b9f6a4d757ff2d7b14cef975f050512d
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ curl -Lo https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
  % Total    % Received   % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 47.4M  100 47.4M    0     0  6569k      0:00:07  0:00:07  -:-:--  959k
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ chmod +x kubectl
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ kubectl version -o yaml
clientVersion:
  buildDate: "2024-03-15T00:08:10Z"
  compiler: gc
  gitCommit: 6813625b7cd706db5bc7388921be03071e1a492d
  gitTreeState: clean
  gitVersion: v1.29.3
  goVersion: go1.21.8
  major: "1"
  minor: "29"
  platform: linux/amd64
kustomizeVersion: V5.0.4-0.20230601165947-6ce0bf390ce3
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ minikube start --driver=docker
  minikube v22.0.0 on Ubuntu 22.04
  * Using Docker driver based on existing profile
  * Starting control plane node minikube in cluster minikube
    Pulling base image ...
    Restarting existing docker container for "minikube" ...
    Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
      Configuring bridge CNI (container Networking Interface) ...
      Verifying Kubernetes components ...
        Using Image registry.k8s.io/ingress-nginx/controller:v1.9.4
        Using Image docker.io/kubernetessu/dashboard:v2.7.8
        Using Image gcr.io/k8s-minikube/storage-provisioner:v5
        Using Image registry.k8s.io/kube-webhook-certgen:v1.8.8
        Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabe0
        Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabe0
      Verifying ingress addon...
      Some dashboard features require the metrics-server addon. To enable all features please run:
        minikube addons enable metrics-server
      └─ Enabled addons: storage-provisioner, default-storageclass, dashboard, ingress
        Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
complab304pc15@complab304pc15: ~
└── complab304pc15@complab304pc15: $ minikube status
minikube
  type: Control Plane
  host: Running
  kubelet: Running
  apiserver: Running
  kubeconfig: configured
complab304pc15@complab304pc15: ~

```





```
Activities Terminal Mar 27 11:38
complab304pc15@complab304pc15: ~
Get:5 https://packages.microsoft.com/repos/code stable/main amd64 Packages [19.4 kB]
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:9 https://ppa.launchpadcontent.net/danielrichter2007/grub-customizer/ubuntu jammy InRelease
Hit:10 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [429 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [1,045 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,299 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,505 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [851 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,059 kB]
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
complab304pc15@complab304pc15: ~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy InRelease
Hit:2 https://packages.microsoft.com/repos/code stable InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://ppa.launchpadcontent.net/danielrichter2007/grub-customizer/ubuntu jammy InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
complab304pc15@complab304pc15: ~$ sudo apt install ca-certificates curl gnupg apt-transport-https -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (1.21.2-2ubuntu1).
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-ubuntu1.15).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.11).
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libbares0 libbares9 libavcodec58 libavutil56 libbdplush libbluray libbs2b0 libchromaprint libcodec2-1.0 libdavids libflashrom
libflite1 libftdi1-2 libgned liblsm1 libgstreamer-plugins-bad1.0-0 libl1v-0-0 libl1v13 libmf1 libmysqfa1 libopenmp10 libpostproc55 librabitmq4 librubberband2 libserd-0-0 libshine3 libsvd0
libratom-0-0 librtt-4-gnult libswresample3 libwscale5 libudfread0 libva-drm2 libvdpau libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzvbi-common libzvbi0
linux-headers-6.2.0-36-generic linux-hwe-6.2.0-36 linux-image-6.2.0-36-generic linux-modules-6.2.0-36-generic linux-modules-extra-6.2.0-36-generic mesa-vdpau-drivers pocketsphinx-en-us
libuv1 libvdpau1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
complab304pc15@complab304pc15: ~$ sudo install -m 0755 -d /etc/apt/keyrings
complab304pc15@complab304pc15: ~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmail -o /etc/apt/keyrings/docker.gpg
TIP: Use 'curl -sSL' to skip SSL verification
complab304pc15@complab304pc15: ~$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
complab304pc15@complab304pc15: ~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
      \"$ . /etc/os-release && echo \"$VERSIOON_CODENAME\"\" stable\" | \
      sudo tee /etc/apt/sources.list.d/docker.list >/dev/null
complab304pc15@complab304pc15: ~$
```

## Conclusion:

In conclusion, we understood how to implement container orchestration using Docker.