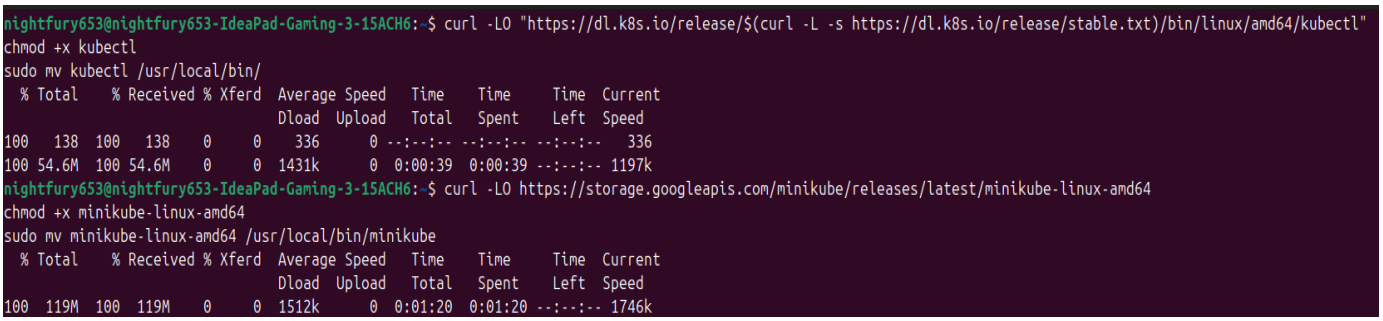
CSE – 322 Cloud Computing

Name: Gaurav Malave
Roll No: 2022BCD0017
Batch: 2

Task 1: Install a Kubernetes Cluster

1] You need to install a Kubernetes cluster using **Minikube**, **kind**, or a cloud provider (**AWS EKS, Azure AKS, GCP GKE**).



2] Start the Kubernetes Cluster using Minikube:

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ sudo usermod -aG docker $USER
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ newgrp docker
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ minikube delete
minikube start --driver=docker
😄  "minikube" profile does not exist, trying anyways.
💀  Removed all traces of the "minikube" cluster.
😄  minikube v1.35.0 on Ubuntu 24.04
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
💾  Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 756.28
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 760.54
🔥  Creating docker container (CPUs=2, Memory=3400MB) ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

3] Verify Installation:

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kubectl config use-context minikube
Switched to context "minikube".
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ 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/ku

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ minikube status
kubectl get nodes
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   2m59s   v1.32.0
```

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kind create cluster --name my-cluster
Creating cluster "my-cluster" ...
 ✓ Ensuring node image (kindest/node:v1.32.2) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
Set kubectl context to "kind-my-cluster"
You can now use your cluster with:

kubectl cluster-info --context kind-my-cluster

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 🙂
```

Task 2: Check the Cluster Details

After the installation, we checked if the cluster was running correctly:

1.Check Node Status

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kubectl get nodes
NAME                       STATUS   ROLES           AGE   VERSION
my-cluster-control-plane   Ready    control-plane   28s   v1.32.2
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kubectl get pods --all-namespaces
NAMESPACE            NAME                                                READY   STATUS    RESTARTS   AGE
kube-system          coredns-668d6bf9bc-pnphb                            1/1     Running   0          31s
kube-system          coredns-668d6bf9bc-zjj5l                            1/1     Running   0          31s
kube-system          etcd-my-cluster-control-plane                       1/1     Running   0          39s
kube-system          kindnet-495x7                                       1/1     Running   0          31s
kube-system          kube-apiserver-my-cluster-control-plane             1/1     Running   0          39s
kube-system          kube-controller-manager-my-cluster-control-plane    1/1     Running   0          39s
kube-system          kube-proxy-f4tvl                                    1/1     Running   0          31s
kube-system          kube-scheduler-my-cluster-control-plane             1/1     Running   0          39s
local-path-storage   local-path-provisioner-7dc846544d-vg7wq             1/1     Running   0          31s
```

2. Check Active Pods

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kubectl get deployments
No resources found in default namespace.
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   68s
```

Task 3: Deploy and Run a Node.js App on Kubernetes

1] A Simple Nodejs Service

2] Dockerfile

3] Build the docker image from the dockerfile

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents$ cd College\ Prep/
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep$ ls
'Cloud Computing'  'Data Warehousing'   Microprocessor   Network_Analysis
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep$ cd Cloud\ Computing/
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ docker build -t nightfury7653/node-app .

2025/03/04 12:40:10 in: []string{}
2025/03/04 12:40:10 Parsed entitlements: []
[+] Building 182.2s (8/8) FINISHED                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 118B                                                                          0.0s
 => [internal] load metadata for docker.io/library/node:16                                                    3.4s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [1/3] FROM docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e424  178.2s
 => => resolve docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c  0.0s
 => => sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c6389705c0b 776B / 776B                    0.0s
 => => sha256:311da6c465ea1576925360eba391bcd32dece9be95960a0bc9ffcb25fe712017 50.50MB / 50.50MB      155.9s
 => => sha256:c94b82f9827cab6e421b350965a9ef11b25b13ffbd1030536203d541f55dcbe2 2.00kB / 2.00kB          0.0s
 => => sha256:1ddc7e4055fdb6f6bf31063b593befda814294f9f904b6ddfc21ab1513bafa8e 7.23kB / 7.23kB          0.0s
 => => sha256:7e9bf114588c05b2df612b083b96582f3b8dbf51647aa6138a50d09d42df2454 17.58MB / 17.58MB        8.5s
 => => sha256:ffd9397e94b74abcb54e514f1430e00f604328d1f895eadbd482f08cc02444e5 51.89MB / 51.89MB      173.7s
 => => sha256:513d779256048c961239af5f500589330546b072775217272e19ffae1635e98e 191.90MB / 191.90MB    104.4s
 => => sha256:ae3b95bbaa61ce24cefdd89e7c74d6fbd7713b2bcae93af47063d06bd7e02172 4.20kB / 4.20kB        105.4s
 => => sha256:0e421f66aff42bb069dffc26af6d132194b22a1082b08c5ef7cd69c627783c04 34.79MB / 34.79MB      127.3s
 => => sha256:ca266fd6192108b67fb57b74753a8c4ca5d8bd458baae3d4df7ce9f42dedcc1d 2.27MB / 2.27MB        129.5s
 => => sha256:ee7d78be1eb92caf6ae84fc3af736b23eca018d5dedc967ae5bdee6d7082403b 450B / 450B            129.9s
 => => extracting sha256:311da6c465ea1576925360eba391bcd32dece9be95960a0bc9ffcb25fe712017              0.9s
 => => extracting sha256:7e9bf114588c05b2df612b083b96582f3b8dbf51647aa6138a50d09d42df2454              0.2s
 => => extracting sha256:ffd9397e94b74abcb54e514f1430e00f604328d1f895eadbd482f08cc02444e5              0.9s
 => => extracting sha256:513d779256048c961239af5f500589330546b072775217272e19ffae1635e98e             2.6s
 => => extracting sha256:ae3b95bbaa61ce24cefdd89e7c74d6fbd7713b2bcae93af47063d06bd7e02172              0.0s
 => => extracting sha256:0e421f66aff42bb069dffc26af6d132194b22a1082b08c5ef7cd69c627783c04              0.5s
 => => extracting sha256:ca266fd6192108b67fb57b74753a8c4ca5d8bd458baae3d4df7ce9f42dedcc1d              0.0s
 => => extracting sha256:ee7d78be1eb92caf6ae84fc3af736b23eca018d5dedc967ae5bdee6d7082403b              0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 305B                                                                             0.0s
 => [2/3] WORKDIR /app                                                                                        0.4s
 => [3/3] COPY server.js .                                                                                    0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:a84e390ed1b40d3b11c8e11eebb8c14dc8a663df07f6f13879ada7ad593cd183                  0.0s
 => => naming to docker.io/nightfury7653/node-app                                                             0.0s
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$
```

4] Push it into the docker hub

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ docker login

USING WEB-BASED LOGIN

i Info → To sign in with credentials on the command line, use 'docker login -u <username>'


Your one-time device confirmation code is: FPMG-PXLZ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…

WARNING! Your credentials are stored unencrypted in '/home/nightfury653/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$
```

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ docker tag 2022bcd0017/node-app 2022bcd0017/node-app:v1
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ docker push 2022bcd0017/node-app:v1
The push refers to repository [docker.io/2022bcd0017/node-app]
8d712d0024ab: Pushed
3deef00f7cc4: Pushed
be322b479aee: Pushed
d41bcd3a037b: Pushed
fe0d845e767b: Pushed
f25ec1d93a58: Pushed
794ce8b1b516: Pushed
3220beed9b06: Pushed
684f82921421: Pushed
9af5f53e8f62: Pushed
v1: digest: sha256:43477a0b9721a6c8d2219a87501951dea0235eb292d54e5a5abc699376d2dd69 size: 2417
```

5] Create a deployment.yaml file

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - name: nodejs-app
        image: 2022bcd0017/node-app:v1
        ports:
        - containerPort: 3000
```

6] Describe and apply node-js deployment

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-deployment configured
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl describe deployment nodejs-deployment
Name:                   nodejs-deployment
Namespace:              default
CreationTimestamp:      Tue, 04 Mar 2025 12:59:26 +0530
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nodejs-app
Replicas:               2 desired | 1 updated | 3 total | 0 available | 3 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nodejs-app
  Containers:
   nodejs-app:
    Image:         2022bcd0017/node-app:v1
    Port:          3000/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      False   MinimumReplicasUnavailable
  Progressing    True    ReplicaSetUpdated
OldReplicaSets:  nodejs-deployment-54d85f495f (2/2 replicas created)
NewReplicaSet:   nodejs-deployment-67dc56ff8 (1/1 replicas created)
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  2m13s  deployment-controller  Scaled up replica set nodejs-deployment-54d85f495f from 0 to 2
  Normal  ScalingReplicaSet  3s     deployment-controller  Scaled up replica set nodejs-deployment-67dc56ff8 from 0 to 1
```

Task 4: Expose the Application Results to the Outside World

1] Create a service.yaml file

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nodejs-service
spec:
  type: NodePort
  selector:
    app: nodejs-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
      nodePort: 30007
```

2] Apply the service.yaml file

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-deployment created
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl apply -f service.yaml
service/nodejs-service unchanged
```

3] Check the deployments,pods and services

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nodejs-deployment   0/2     2            0           115s
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl get pods
NAME                                 READY   STATUS              RESTARTS   AGE
nodejs-deployment-67dc56ff8-627pf    0/1     ContainerCreating   0          118s
nodejs-deployment-67dc56ff8-b48zn    0/1     ContainerCreating   0          118s
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP        3h36m
nodejs-service   NodePort    10.97.163.49    <none>        80:30007/TCP   3m48s
```

4] Get the url where your service is running

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ minikube service nodejs-service --url
http://192.168.49.2:30007
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$
```

5] The web page where your service is running

Task 5: Monitor and Analyze the Pods

1] Get all info of pods and the logs of service

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl get pods -o wide
NAME                                 READY   STATUS    RESTARTS   AGE     IP           NODE       NOMINATED NODE   READINESS GATES
nodejs-deployment-67dc56ff8-627pf    1/1     Running   0          6m47s   10.244.0.4   minikube   <none>           <none>
nodejs-deployment-67dc56ff8-b48zn    1/1     Running   0          6m47s   10.244.0.5   minikube   <none>           <none>
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl logs nodejs-deployment-67dc56ff8-627pf
Server is running on port 3000
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl logs nodejs-deployment-67dc56ff8-b48zn
Server is running on port 3000
```

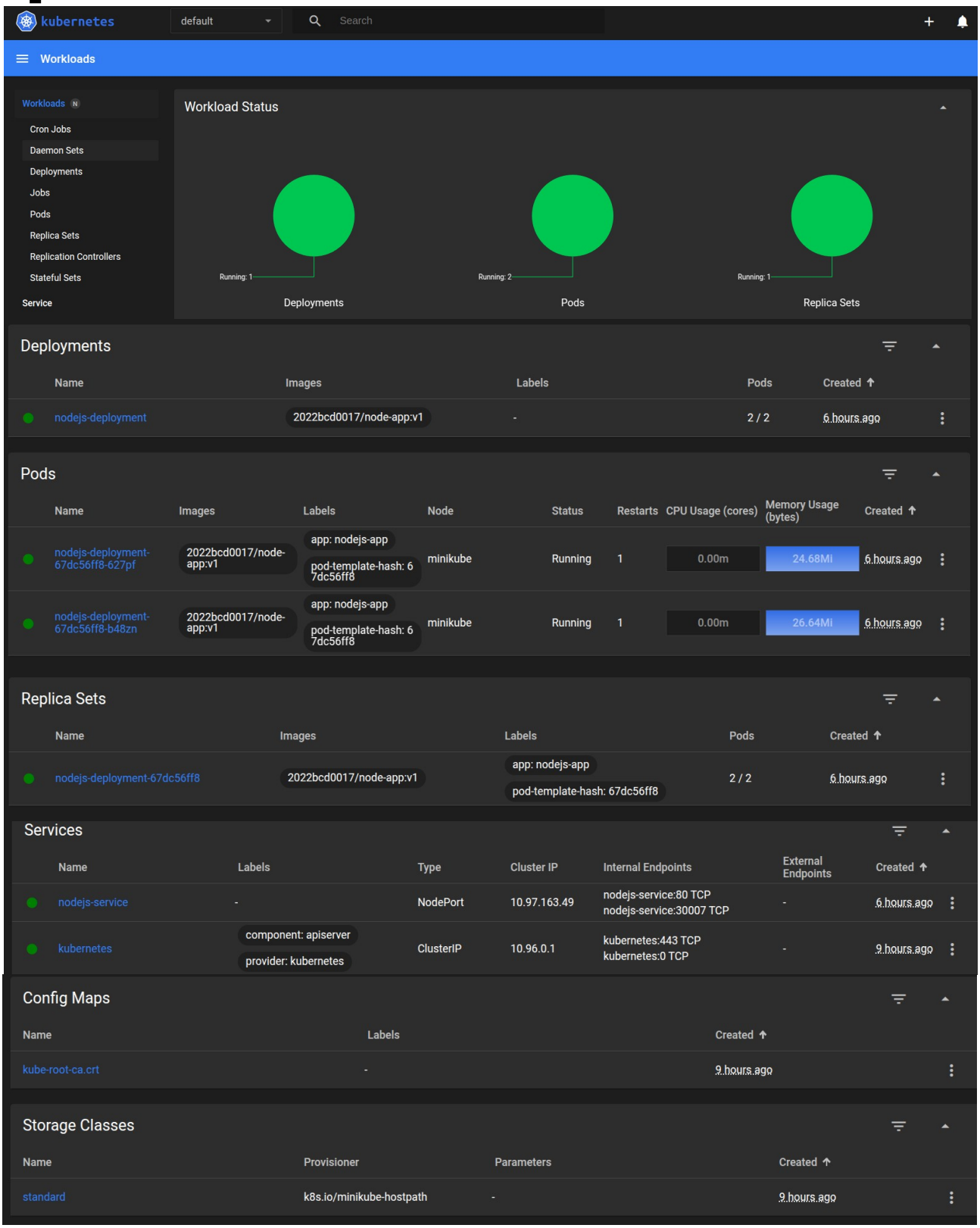
2] Description of nodejs-deployment

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl describe pod nodejs-deployment-67dc56ff8-627pf
Name:             nodejs-deployment-67dc56ff8-627pf
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Tue, 04 Mar 2025 15:52:27 +0530
Labels:           app=nodejs-app
                  pod-template-hash=67dc56ff8
Annotations:      <none>
Status:           Running
IP:               10.244.0.4
IPs:
  IP:           10.244.0.4
Controlled By:  ReplicaSet/nodejs-deployment-67dc56ff8
Containers:
  nodejs-app:
    Container ID:   docker://f524e085f14d6a29581889a063803874bfa7325a230c5e32fd77de35665efea8
    Image:          2022bcd0017/node-app:v1
    Image ID:       docker-pullable://2022bcd0017/node-app@sha256:43477a0b9721a6c8d2219a87501951dea0235eb292d54e5a5abc699376d2dd69
    Port:           3000/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 04 Mar 2025 15:54:29 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xfdn9 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-xfdn9:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  7m54s  default-scheduler  Successfully assigned default/nodejs-deployment-67dc56ff8-627pf to minikube
  Normal  Pulling    7m53s  kubelet            Pulling image "2022bcd0017/node-app:v1"
  Normal  Pulled     5m52s  kubelet            Successfully pulled image "2022bcd0017/node-app:v1" in 3.708s (2m1.319s including waiting). Image size: 908820932 bytes.
  Normal  Created    5m52s  kubelet            Created container: nodejs-app
  Normal  Started    5m52s  kubelet            Started container nodejs-app
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$
```

3] Activate minikube dashboard

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:34353/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
Opening in existing browser session.
^C
```

4] Workload Status



Task 6: Expose the Application to the External World

1] Enable ingress on minikube addons

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~$ minikube addons enable ingress
💡  ingress 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 registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
    ▪ Using image registry.k8s.io/ingress-nginx/controller:v1.11.3
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
```

2] Create a ingress.yaml file

ingress.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx
  rules:
  - host: myapp.local
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nodejs-service
            port:
              number: 80
```

3] Apply the ingress.yaml file on kubernetes

```
nightfury653@nightfury653-IdeaPad-Gaming-3-15ACH6:~/Documents/College Prep/Cloud Computing$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/my-ingress created
```

4] Add the myapp.local ip address to the /etc/hosts

```
  GNU nano 7.2
127.0.0.1 localhost
127.0.1.1 nightfury653-IdeaPad-Gaming-3-15ACH6
127.0.0.1 myapp.local

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.49.2 myapp.local
```

5] This myapp.local can access our service on the port specified on the /etc/hosts