

Project 01

In this project, you will develop a simple Node.js application, deploy it on a local Kubernetes cluster using Minikube, and configure various Kubernetes features. The project includes Git version control practices, creating and managing branches, and performing rebases. Additionally, you will work with ConfigMaps, Secrets, environment variables, and set up vertical and horizontal pod autoscaling.

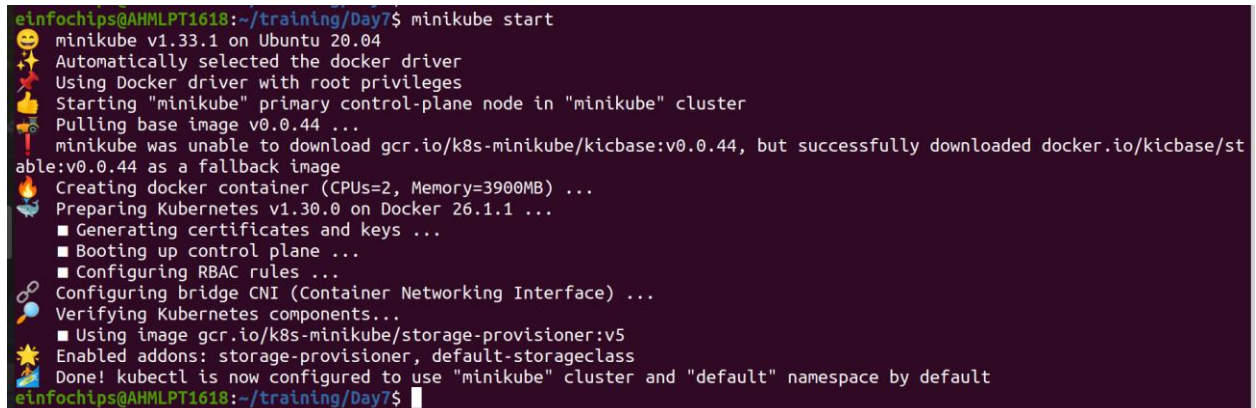
Project 01

Project Steps

1. Setup Minikube and Git Repository

Start Minikube:

```
minikube start
```



```
einfochips@AHMLPT1618:~/training/Day7$ minikube start
🐹 minikube v1.33.1 on Ubuntu 20.04
🌟 Automatically selected the docker driver
👉 Using Docker driver with root privileges
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📡 Pulling base image v0.0.44 ...
🔧 minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.44, but successfully downloaded docker.io/kicbase/stable:v0.0.44 as a fallback image
🔥 Creating docker container (CPUs=2, Memory=3900MB) ...
🚀 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
   ▪ 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
einfochips@AHMLPT1618:~/training/Day7$
```

1.2 Set Up Git Repository

Create a new directory for your project:

```
mkdir nodejs-k8s-project
cd nodejs-k8s-project
```

```
einfochips@AHMLPT1618:~/training/Day7$ mkdir nodejs-k8s-project
einfochips@AHMLPT1618:~/training/Day7$ ls
nodejs-k8s-project
einfochips@AHMLPT1618:~/training/Day7$ cd nodejs-k8s-project
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Initialize Git repository:

```
git init
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git init
Initialized empty Git repository in /home/einfochips/training/Day7/nodejs-k8s-project/.git/
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Create a **.gitignore** file:

```
node_modules/
.env
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ nano .gitignore
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat .gitignore
node_modules/
.env
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ll
total 16
drwxrwxr-x 3 einfochips einfochips 4096 Jul 20 14:13 ./
drwxrwxr-x 3 einfochips einfochips 4096 Jul 20 14:11 ../
drwxrwxr-x 7 einfochips einfochips 4096 Jul 20 14:12 .git/
-rw-rw-r-- 1 einfochips einfochips  19 Jul 20 14:13 .gitignore
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Add and commit initial changes:

```
git add .
git commit -m "Initial commit"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git commit -m "Initial commit"
[master (root-commit) 8ebf43e] Initial commit
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

2. Develop a Node.js Application

2.1 Create the Node.js App

Initialize the Node.js project:

```
npm init -y
```

Install necessary packages:

```
npm install express body-parser
```

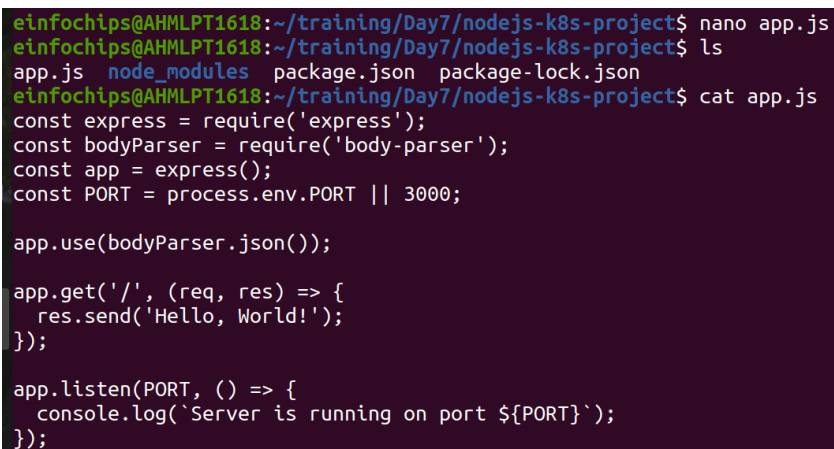
Create **app.js**:

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const PORT = process.env.PORT || 3000;

app.use(bodyParser.json());

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

A terminal window with a dark background and light green text. It shows the following commands and their outputs: 1. 'nano app.js' is run, creating the file. 2. 'ls' is run, showing 'app.js', 'node_modules', 'package.json', and 'package-lock.json'. 3. 'cat app.js' is run, displaying the contents of the app.js file, which matches the code shown in the previous block.

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ nano app.js
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ls
app.js  node_modules  package.json  package-lock.json
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat app.js
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const PORT = process.env.PORT || 3000;

app.use(bodyParser.json());

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

Update **package.json** to include a start script:

```
"scripts": {  
  "start": "node app.js"  
}
```

2.2 Commit the Node.js Application

Add and commit changes:

git add .

git commit -m "Add Node.js application code"

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat package.json  
{  
  "name": "nodejs-k8s-project",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
    "start": "node app.js"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "body-parser": "^1.20.2",  
    "express": "^4.19.2"  
  }  
}
```

einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project\$

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git add .  
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git commit -m "Add Node.js application code"  
[master 0dcc888] Add Node.js application code  
2 files changed, 2 insertions(+), 2 deletions(-)  
create mode 100644 .dockerignore  
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

3. Create Dockerfile and Docker Compose

3.1 Create a Dockerfile

Add **Dockerfile**:

```
# Use official Node.js image
```

```
FROM node:18

# Set the working directory
WORKDIR /usr/src/app

# Copy package.json and package-lock.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code
COPY . .

# Expose the port on which the app runs
EXPOSE 3000

# Command to run the application
CMD [ "npm", "start" ]
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ nano Dockerfile
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ls
app.js  Dockerfile  node_modules  package.json  package-lock.json
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat Dockerfile
# Use official Node.js image
FROM node:18

# Set the working directory
WORKDIR /usr/src/app

# Copy package.json and package-lock.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code
COPY . .

# Expose the port on which the app runs
EXPOSE 3000

# Command to run the application
CMD [ "npm", "start" ]
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Create a **.dockerignore** file:

```
node_modules
```

.npm

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ nano .dockerignore
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ll
total 60
drwxrwxr-x 4 einfochips einfochips 4096 Jul 20 14:51 ./
drwxrwxr-x 3 einfochips einfochips 4096 Jul 20 14:11 ../
-rw-rw-r-- 1 einfochips einfochips 316 Jul 20 14:37 app.js
-rw-rw-r-- 1 einfochips einfochips 357 Jul 20 14:50 Dockerfile
-rw-rw-r-- 1 einfochips einfochips 18 Jul 20 14:51 .dockerignore
drwxrwxr-x 8 einfochips einfochips 4096 Jul 20 14:45 .git/
-rw-rw-r-- 1 einfochips einfochips 19 Jul 20 14:13 .gitignore
drwxrwxr-x 66 einfochips einfochips 4096 Jul 20 14:36 node_modules/
-rw-rw-r-- 1 einfochips einfochips 361 Jul 20 14:42 package.json
-rw-rw-r-- 1 einfochips einfochips 20563 Jul 20 14:36 package-lock.json
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ls
app.js Dockerfile node_modules package.json package-lock.json
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

3.2 Create `docker-compose.yml` (optional for local testing)

Add `docker-compose.yml`:

```
version: '3'
services:
  app:
    build: .
    ports:
      - "3000:3000"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ nano docker-compose.yml
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ ls
app.js docker-compose.yml Dockerfile node_modules package.json package-lock.json
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat docker-compose.yml
version: '3'
services:
  app:
    build: .
    ports:
      - "3000:3000"
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Add and commit changes:

```
git add Dockerfile docker-compose.yml
git commit -m "Add Dockerfile and Docker Compose configuration"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git add Dockerfile docker-compose.yml
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git commit -m "Add Dockerfile and Docker Compose configuration"
[master eaac9c6] Add Dockerfile and Docker Compose configuration
2 files changed, 26 insertions(+)
create mode 100644 Dockerfile
create mode 100644 docker-compose.yml
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

4. Build and Push Docker Image

4.1 Build Docker Image

Build the Docker image:

```
docker build -t nodejs-app:latest .
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ docker build -t nodejs-app:latest .
[+] Building 5.6s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 396B                                0.0s
=> [internal] load metadata for docker.io/library/node:18         0.9s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 58B                                     0.0s
=> [1/5] FROM docker.io/library/node:18@sha256:aabbaf118c7c0a6e9a3bda69bd2a94b0f6c4150bf80c501ef1c87ffcb5687365 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 7.23kB                                  0.0s
=> CACHED [2/5] WORKDIR /usr/src/app                             0.0s
=> [3/5] COPY package*.json ./                                    0.0s
=> [4/5] RUN npm install                                          4.4s
=> [5/5] COPY . .                                                0.0s
=> exporting to image                                             0.1s
=> => exporting layers                                             0.1s
=> => writing image sha256:d8943c6e4029210655e4988d0e06f0cc1007b67f89e6f05a913463085255d32f 0.0s
=> => naming to docker.io/library/nodejs-app:latest              0.0s
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

4.2 Push Docker Image to Docker Hub

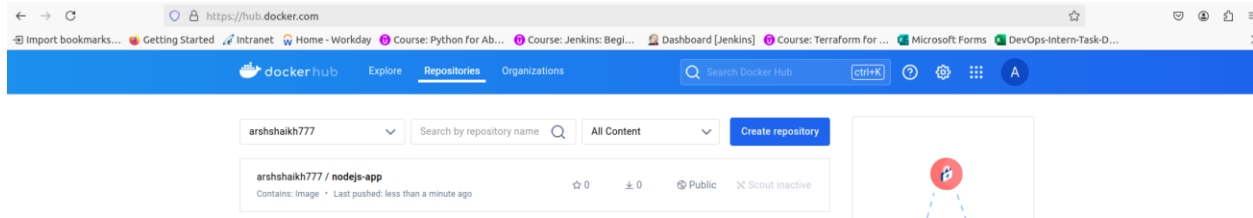
Tag and push the image:

```
docker tag nodejs-app:latest your-dockerhub-username/nodejs-app:latest
```

```
docker push your-dockerhub-username/nodejs-app:latest
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ docker tag nodejs-app:latest arshshaikh777/nodejs-app:latest
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ docker push arshshaikh777/nodejs-app:latest
The push refers to repository [docker.io/arshshaikh777/nodejs-app]
2dee376ad5f0: Pushed
3668dfece38c: Pushed
d61b0ca87c06: Pushed
9989ba107fb9: Pushed
0970e1a837f7: Mounted from library/node
d4061df7c236: Mounted from library/node
9487e6e19e60: Mounted from library/node
6ef00066aa6f: Mounted from library/node
b11bb163e263: Mounted from library/node
b779a72428fa: Mounted from library/node
8ada682d3780: Mounted from library/node
15bb10f9bb3a: Mounted from library/node
latest: digest: sha256:741926f057df13a53407f403c5e800921d97a979aff7298fe030c358807f1f4b size: 2839
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```



Add and commit changes:

```
git add .
git commit -m "Build and push Docker image"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git commit -m "Build and push Docker image"
On branch master
nothing to commit, working tree clean
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

5. Create Kubernetes Configurations

5.1 Create Kubernetes Deployment

Create **kubernetes/deployment.yaml**:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
        - name: nodejs-app
          image: your-dockerhub-username/nodejs-app:latest
          ports:
```


- containerPort: 3000
- env:
- name: PORT
valueFrom:
configMapKeyRef:
name: app-config
key: PORT
 - name: NODE_ENV
valueFrom:
secretKeyRef:
name: app-secrets
key: NODE_ENV

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
        - name: nodejs-app
          image: your-dockerhub-username/nodejs-app:latest
          ports:
            - containerPort: 3000
          env:
            - name: PORT
              valueFrom:
                configMapKeyRef:
                  name: app-config
                  key: PORT
            - name: NODE_ENV
              valueFrom:
                secretKeyRef:
                  name: app-secrets
                  key: NODE_ENV
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$
```

5.2 Create ConfigMap and Secret

Create `kubernetes/configmap.yaml`:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$ cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$
```

Create `kubernetes/secret.yaml`:

```
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbmFs # Base64 encoded value for "production"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbmFs # Base64 encoded value for "production"
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$
```

Add and commit Kubernetes configurations:

```
git add kubernetes/
```

```
git commit -m "Add Kubernetes deployment, configmap, and secret"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git add kubernetes/
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ git commit -m "Add Kubernetes deployment, configmap, and secret"
[master 9a19f28] Add Kubernetes deployment, configmap, and secret
3 files changed, 43 insertions(+)
create mode 100644 kubernetes/configmap.yaml
create mode 100644 kubernetes/deployment.yaml
create mode 100644 kubernetes/secret.yaml
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

5.3 Apply Kubernetes Configurations

Apply the ConfigMap and Secret:

```
kubectl apply -f kubernetes/configmap.yaml
kubectl apply -f kubernetes/secret.yaml
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ kubectl apply -f kubernetes/configmap.yaml
configmap/app-config created
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ kubectl apply -f kubernetes/secret.yaml
secret/app-secrets created
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Apply the Deployment:

```
kubectl apply -f kubernetes/deployment.yaml
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ kubectl apply -f kubernetes/deployment.yaml
deployment.apps/nodejs-app-deployment created
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

6. Implement Autoscaling

6.1 Create Horizontal Pod Autoscaler

Create **kubernetes/hpa.yaml**:

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
```

```

    name: nodejs-app-deployment
minReplicas: 2
maxReplicas: 5
metrics:
- type: Resource
  resource:
    name: cpu
    target:
      type: Utilization
      averageUtilization: 50

```

```

einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$ cat hpa.yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-app-deployment
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project/kubernetes$

```

Apply the HPA:

```
kubectl apply -f kubernetes/hpa.yaml
```

```

einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-app-hpa created
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$

```

6.2 Create Vertical Pod Autoscaler

Create `kubernetes/vpa.yaml`:

```
apiVersion: autoscaling.k8s.io/v1beta2
```

```
kind: VerticalPodAutoscaler
metadata:
  name: nodejs-app-vpa
spec:
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-app-deployment
  updatePolicy:
    updateMode: "Auto"
```

```
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$ cat kubernetes/vpa.yaml
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: nodejs-app-vpa
spec:
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-app-deployment
  updatePolicy:
    updateMode: "Auto"
einfochips@AHMLPT1618:~/training/Day7/nodejs-k8s-project$
```

Apply the VPA:

```
kubectl apply -f kubernetes/vpa.yaml
```