# 3 Project Title: Canary Deployment in Kubernetes without using Service Mesh

*Submitted by: Gurleen Kaur Sethi*

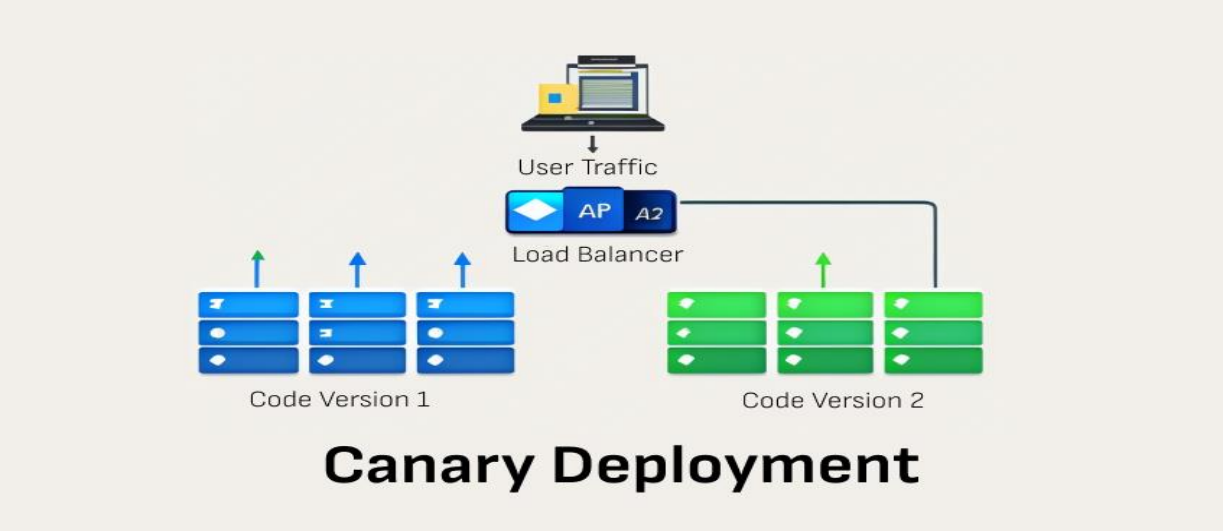
*Program: B.Tech CSE (DevOps)*

*Sap ID: 500093028*

*Github repo link: https://github.com/gurleen131/signup-login-app.git*

**Objective:**

Deploy a new version of an application (canary) alongside the existing version (stable) in a Kubernetes cluster to test its performance and reliability with real traffic before a full rollout.




**Theory:**

A canary deployment is a strategy for gradually releasing a new version of an application to a subset of users or servers. The goal is to minimize the impact of potential issues by limiting the number of users affected during the initial deployment phase

**Prerequisites**

- A Kubernetes cluster (can be set up using Minikube, kind, or a cloud provider like AWS EKS, GKE, or AKS).

```
PS C:\Users\S\Desktop\signup-login-app> minikube version
minikube version: v1.34.0
commit: 210b148df93a80eb872ecbeb7e35281b3c582c61
PS C:\Users\S\Desktop\signup-login-app> minikube start
😄  minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4602 Build 22631.4602
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.45 ...
🔄  Restarting existing docker container for "minikube" ...
❗  Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🐳  Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\S\Desktop\signup-login-app>
```

- Kubectl command-line tool configured to access your cluster.

```
PS C:\Users\S\Desktop\signup-login-app> kubectl version
Client Version: v1.32.0
Kustomize Version: v5.5.0
Server Version: v1.31.0
PS C:\Users\S\Desktop\signup-login-app>
```

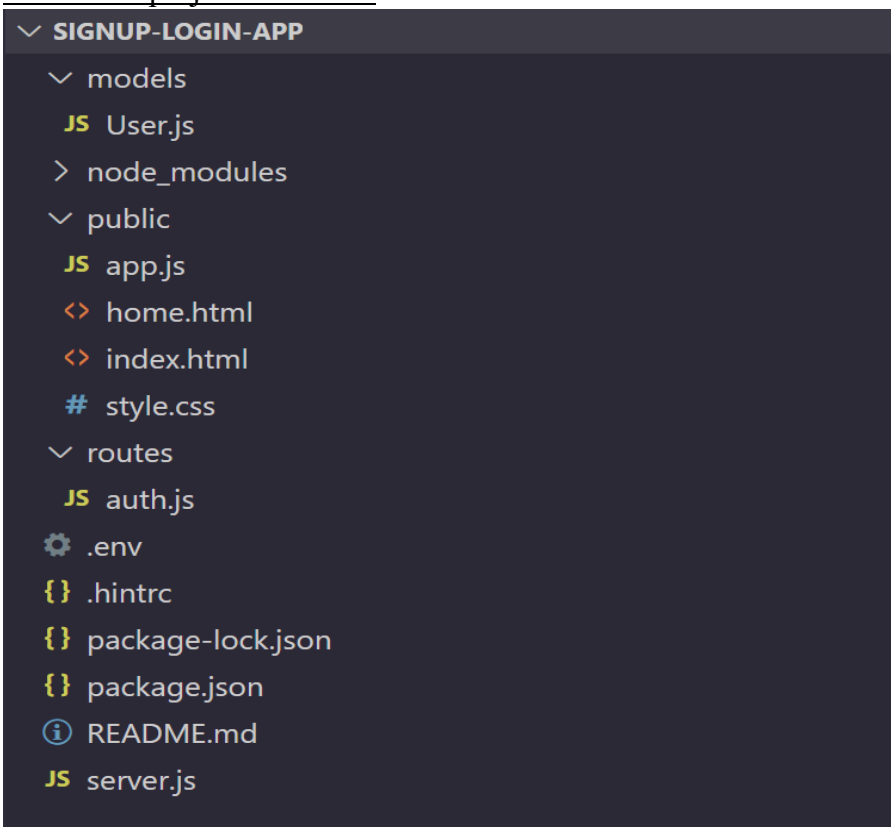
- Docker for building container images.

```
PS C:\Users\S\Desktop\signup-login-app> docker --version
Docker version 27.3.1, build ce12230
PS C:\Users\S\Desktop\signup-login-app>
```

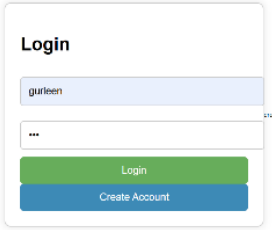
## Project Deliveries:

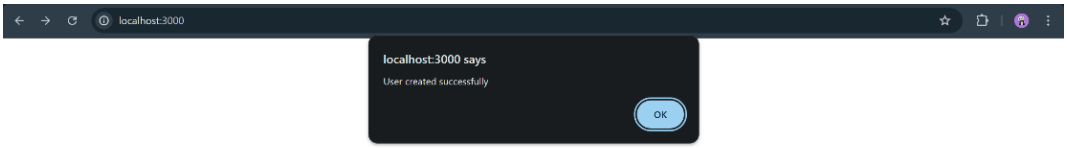
### Part 1- Application Setup

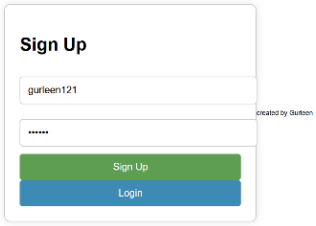
- Create a Sample Application:
  Developed a basic Node.js application as shown below, it is a simple login and signup application connected to MongoDB to serve as the base for deployment.
  Here is the project structure :



## Working of the application on localhost:3000:

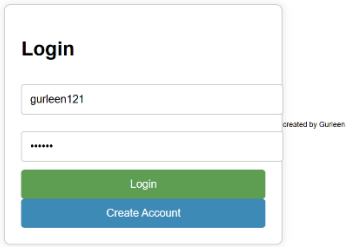



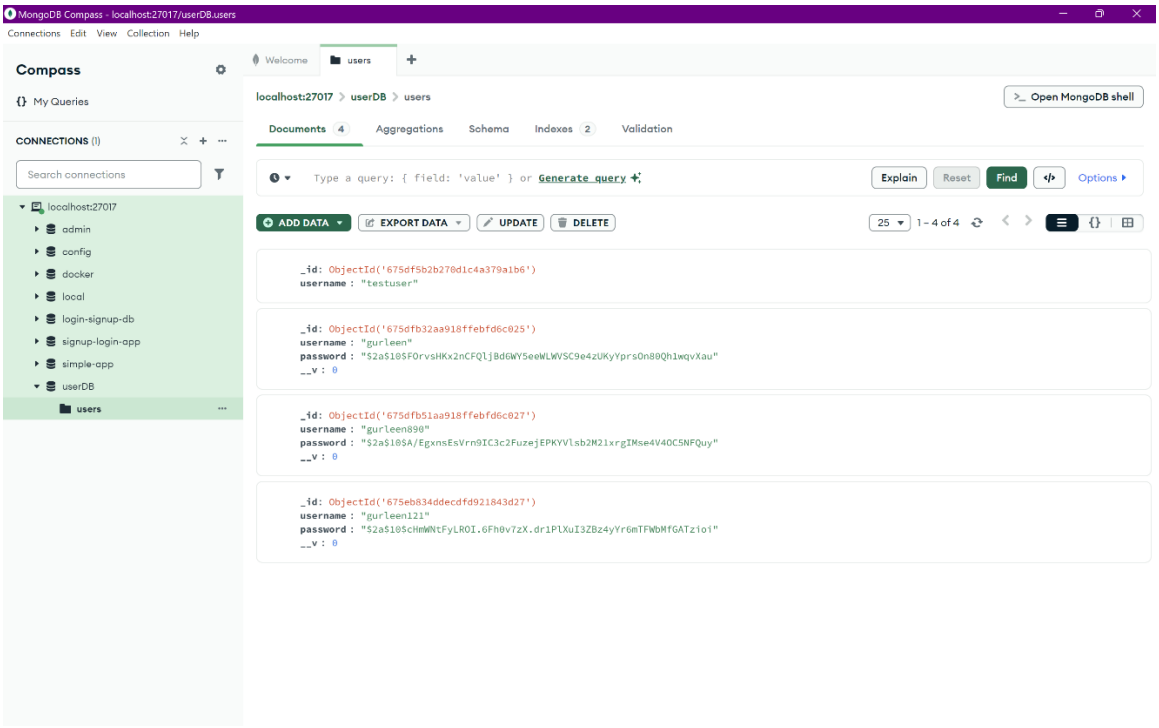




# Welcome to the Home Page

You are successfully logged in!

MongoDB compass to easily see the data retrieval and creation:




- Containerize the Application
  Use Docker to encapsulate the application and its dependencies into a portable container image.
  Here is the Dockerfile for the application:

```
FROM node:18

WORKDIR /app

COPY package.json package-lock.json ./

# Install app dependencies
RUN npm install

#Copy all files to the container
COPY . .

# Expose the port the app runs on (3000 for your app)
EXPOSE 3000

# Step 7: Command to run the application when the container starts
CMD ["node", "server.js"]

```

- Build and Push the Docker Image:
  Docker build -t node-app .

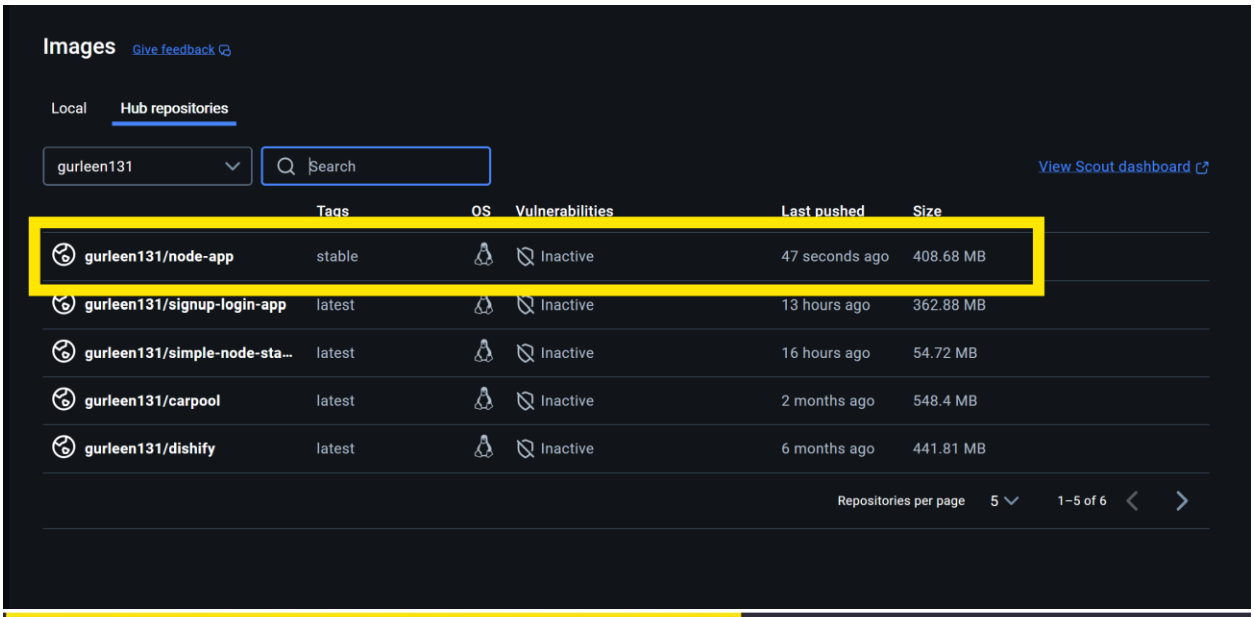
```
PS C:\Users\S\Desktop\signup-login-app> docker images
REPOSITORY                  TAG       IMAGE ID       CREATED        SIZE
gcr.io/k8s-minikube/kicbase   v0.0.45   e7c9bc3bc515   3 months ago   1.81GB
gcr.io/k8s-minikube/kicbase   <none>    81df28859520   3 months ago   1.81GB
PS C:\Users\S\Desktop\signup-login-app> docker build -t node-app .
[+] Building 121.0s (11/11) FINISHED                                                                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                                0.0s
 => => transferring dockerfile: 371B                                                                                                0.0s
 => [internal] load metadata for docker.io/library/node:18                                                                          3.3s
 => [auth] library/node:pull token for registry-1.docker.io                                                                         0.0s
 => [internal] load .dockerignore                                                                                                   0.0s
 => => transferring context: 2B                                                                                                     0.0s
 => [1/5] FROM docker.io/library/node:18@sha256:b57ae84fe7880a23b389f8260d726b784010ed470c2ee26d4e2cbdb955d25b12                  112.4s
 => => resolve docker.io/library/node:18@sha256:b57ae84fe7880a23b389f8260d726b784010ed470c2ee26d4e2cbdb955d25b12                    0.0s
 => => sha256:247468edfd9afcf43bf96caab52a1d979edd5eb13afcaf570c1513f4a35fa43f 446B / 446B                                          0.5s
 => => sha256:2cd8c50fd8ca9ed98f596afc5d92d00b4492b7b069d2d339a6ed8682fc568961 1.25MB / 1.25MB                                      2.2s
 => => sha256:a3c94c84d15dfc1c2c202acca56d7327f541d62c10f9bc1dfb013a618aebd5f1 45.70MB / 45.70MB                                   75.5s
 => => sha256:6399a464889d3eae2913051cb98c35d0b6bfa20ec77d6b3a04617d4a298a2a56 3.32kB / 3.32kB                                     1.0s
 => => sha256:ce82e98d553dd62ca6a12bebfe83992ae9f9ae2748275e74b66a68cc094f868b 211.31MB / 211.31MB                                104.5s
 => => sha256:551df7f94f9c131f2fec0e80631424116 5f0a1c88b935b9fac22be91af227e0 64.39MB / 64.39MB                                   61.5s
 => => sha256:5bd71677db44bb63b94de61b6f1f95d5540b4ba2d6a8a6bc4d19f422b25e0c2b 23.87MB / 23.87MB                                   40.3s
 => => sha256:fdf894e782a221820acf469d425b802be26aedb5e5d26ea80a650ff6a974d488 48.50MB / 48.50MB                                   62.8s
 => => extracting sha256:fdf894e782a221820acf469d425b802be26aedb5e5d26ea80a650ff6a974d488                                            0.9s
 => => extracting sha256:5bd71677db44bb63b94de61b6f1f95d5540b4ba2d6a8a6bc4d19f422b25e0c2b                                            0.4s
 => => extracting sha256:551df7f94f9c131f2fec0e80631424116 5f0a1c88b935b9fac22be91af227e0                                           1.2s
 => => extracting sha256:ce82e98d553dd62ca6a12bebfe83992ae9f9ae2748275e74b66a68cc094f868b                                           3.1s
 => => extracting sha256:6399a464889d3eae2913051cb98c35d0b6bfa20ec77d6b3a04617d4a298a2a56                                           0.0s
 => => extracting sha256:a3c94c84d15dfc1c2c202acca56d7327f541d62c10f9bc1dfb013a618aebd5f1                                           1.0s
 => => extracting sha256:2cd8c50fd8ca9ed98f596afc5d92d00b4492b7b069d2d339a6ed8682fc568961                                           0.0s
 => => extracting sha256:247468edfd9afcf43bf96caab52a1d979edd5eb13afcaf570c1513f4a35fa43f                                           0.0s
 => [internal] load build context                                                                                                   0.5s
 => => transferring context: 472.99kB                                                                                               0.4s
 => [2/5] WORKDIR /app                                                                                                              0.3s
 => [3/5] COPY package.json package-lock.json ./                                                                                    0.1s
 => [4/5] RUN npm install                                                                                                           2.0s
 => [5/5] COPY . .                                                                                                                  0.8s
 => exporting to image                                                                                                              1.8s
 => => exporting layers                                                                                                             0.9s
```

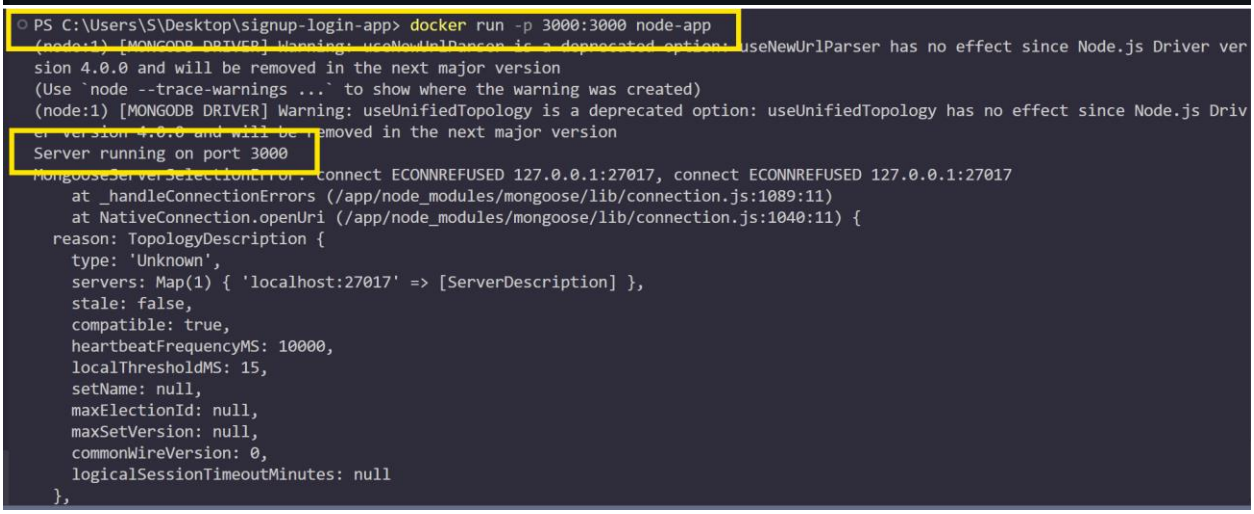
**Docker tag node-app gurleen131/node-app:stable**

**Docker push gurleen131/node-app:stable**

```
PS C:\Users\S\Desktop\signup-login-app> docker tag node-app gurleen131/node-app:stable
PS C:\Users\S\Desktop\signup-login-app> docker push gurleen131/node-app:stable
The push refers to repository [docker.io/gurleen131/node-app]
2d0a41aea58a: Pushed
ce82e98d553d: Pushed
5564e2087ec5: Pushed
91379ea53bbd: Pushed
551df7f94f9c: Pushed
2cd8c50fd8ca: Pushed
e4bc6e880673: Pushed
247468edfd9a: Pushed
1e160d1b6ace: Pushed
fdf894e782a2: Pushed
5bd71677db44: Pushed
6399a464889d: Pushed
a3c94c84d15d: Pushed
stable: digest: sha256:57beebd1d5b0d57241c6bd13787794a7ea8a72ef703a8160932ca981657058c8 size: 856
PS C:\Users\S\Desktop\signup-login-app>
```

Here is the dockerHub registry:




## Part 2- Create Kubernetes Deployment and Service

- Deploy the Stable Version
  Deploy the existing, tested version of the application in the Kubernetes cluster as the stable baseline.

1. **stable-deployment.yaml file:**

```
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
  name: node-app-stable
  labels:
    app: node-app
    version: stable
spec:
  replicas: 3  # Number of pod replicas
  selector:
    matchLabels:
      app: node-app
      version: stable
  template:
    metadata:
      labels:
        app: node-app
        version: stable
    spec:
      containers:
      - name: node-app
        image: gurleen131/node-app:stable  # Docker image for stable version
        ports:
        - containerPort: 3000

```

- Create a Service
  Expose the stable application using a Kubernetes Service for consistent network access
  Service.yaml file:

```
io.k8s.api.core.v1.Service (v1@service.json)
apiVersion: v1
kind: Service
metadata:
  name: node-app-service
spec:
  selector:
    app: node-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
  type: LoadBalancer

```

- Apply the Deployments and Service
  Kubectl apply -f stable-deployment.yaml
  Kubectl apply -f service.yaml

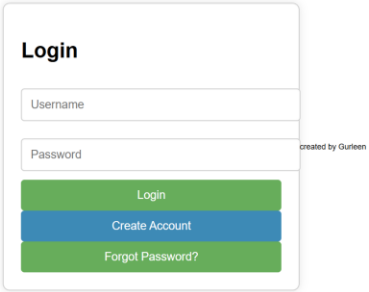
```
PS C:\Users\S\Desktop\signup-login-app> kubectl apply -f stable-deployment.yaml
deployment.apps/node-app-stable created
PS C:\Users\S\Desktop\signup-login-app> kubectl apply -f service.yaml
service/node-app-service created
PS C:\Users\S\Desktop\signup-login-app>
```

```
PS C:\Users\S\Desktop\signup-login-app> kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
mongodb           1/1     1            1           14h
node-app-stable   3/3     3            3           8m37s
PS C:\Users\S\Desktop\signup-login-app>
```

**Part 3- Create the Canary Deployment**

Deploy the updated version of the application alongside the stable one as a separate deployment.

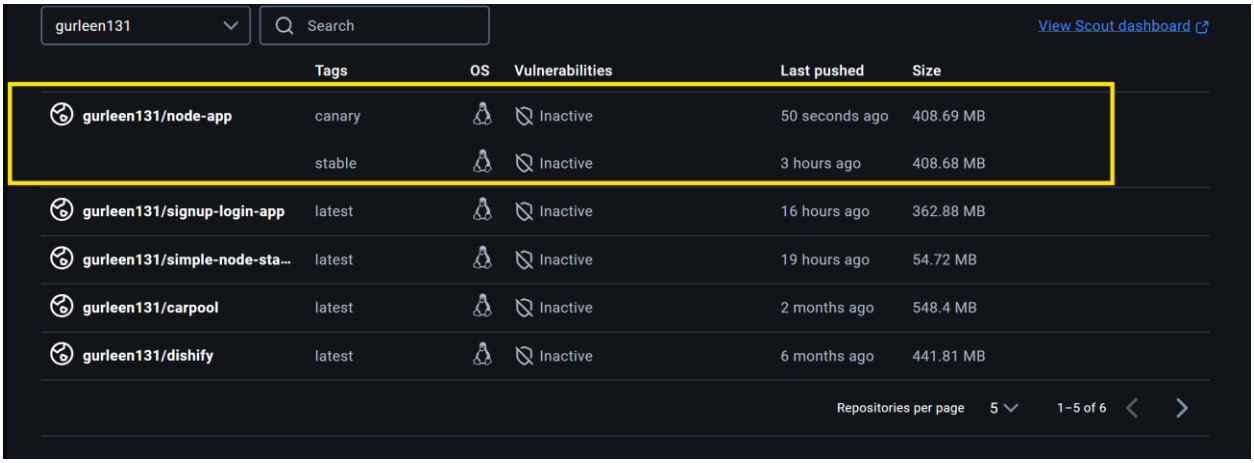
New feature added(forgot password button):


- Deploy the Canary Version:

```
PS C:\Users\S\Desktop\signup-login-app> docker build -t node-app:v2 .
[+] Building 6.1s (11/11) FINISHED                                                                    docker:desktop-
inux
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 371B
0.0s
 => [internal] load metadata for docker.io/library/node:18
2.6s
 => [auth] library/node:pull token for registry-1.docker.io
0.0s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/5] FROM docker.io/library/node:18@sha256:b57ae84fe7880a23b389f8260d726b784010ed470c2ee26d4e2cbdb955d25b12
0.0s
 => => resolve docker.io/library/node:18@sha256:b57ae84fe7880a23b389f8260d726b784010ed470c2ee26d4e2cbdb955d25b12
```

```
PS C:\Users\S\Desktop\signup-login-app> docker push gurleen131/node-app:canary
The push refers to repository [docker.io/gurleen131/node-app]
247468edfd9a: Layer already exists
5bd71677db44: Layer already exists
ce82e98d553d: Layer already exists
2cd8c50fd8ca: Layer already exists
fdf894e782a2: Layer already exists
2d0a41aea58a: Layer already exists
6399a464889d: Layer already exists
2c4f77247eef: Pushed
a3c94c84d15d: Layer already exists
5564e2087ec5: Layer already exists
551df7f94f9c: Layer already exists
e4bc6e880673: Layer already exists
1016d93bf402: Pushed
canary: digest: sha256:e724ee6213dd6cea34d406e837b2075691dd924e024e15980f1c43340f2e926c size: 856
```




- Apply the Canary Deployment
  Create Canary-deployment.yaml file:

```
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
  name: node-app-canary
  labels:
    app: node-app
    version: canary
spec:
  replicas: 3  # Number of pod replicas
  selector:
    matchLabels:
      app: node-app
      version: canary
  template:
    metadata:
      labels:
        app: node-app
        version: canary
    spec:
      containers:
      - name: node-app
        image: gurleen131/node-app:canary
        ports:
        - containerPort: 3000

```

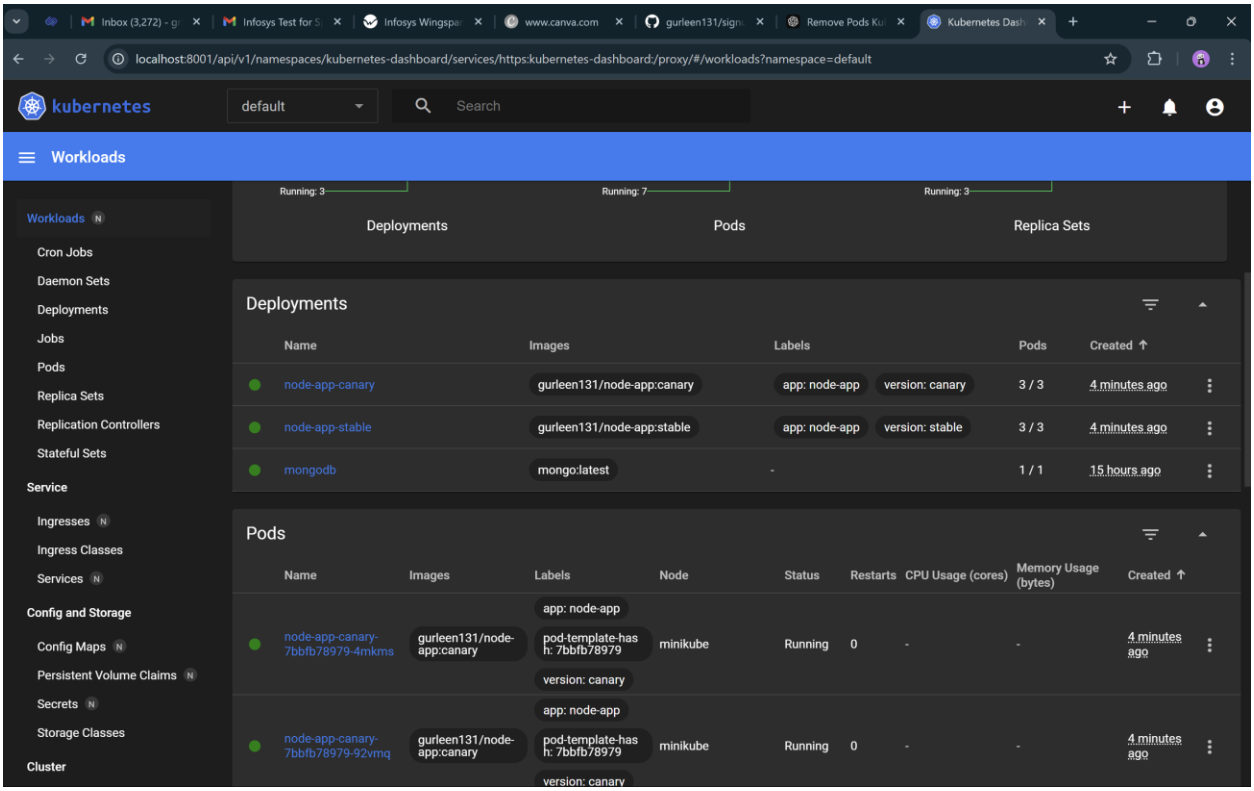
Kubectl apply -f canary-deployment.yaml

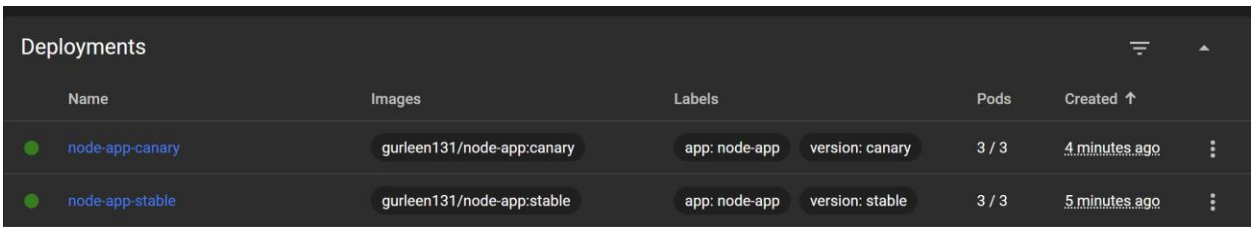
```
PS C:\Users\S\Desktop\signup-login-app> kubectl apply -f canary-deployment.yaml
deployment.apps/node-app-canary created
PS C:\Users\S\Desktop\signup-login-app>
```

```
PS C:\Users\S\Desktop\signup-login-app> kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
mongodb-57b8f5699b-pscfm            1/1     Running   3 (12m ago)   15h
node-app-canary-7bbfb78979-4mkms    1/1     Running   0             17s
node-app-canary-7bbfb78979-92vmq    1/1     Running   0             17s
node-app-canary-7bbfb78979-94sk2    1/1     Running   0             17s
node-app-stable-54cc45f8f5-9bv5s    1/1     Running   0             48s
node-app-stable-54cc45f8f5-bhkmx    1/1     Running   0             48s
node-app-stable-54cc45f8f5-wjd6g    1/1     Running   0             48s
PS C:\Users\S\Desktop\signup-login-app>
```

Setup kubernetes dash board (kubectl proxy)

Check kubernetes cluster (localhost:8081) to see the deployments and services



**Part 4- Traffic Routing**

- Update the Service to Route Traffic

  create two separate services, like stable-service and canary-service, each linked to its own deployment. Then, you can manually control traffic flow by adjusting your app or DNS settings.
  **node-app-stable-service.yaml to ensure it only targets the stable version:**

```
io.k8s.api.core.v1.Service (v1@service.json)
apiVersion: v1
kind: Service
metadata:
  name: node-app-stable-service
spec:
  selector:
    app: node-app
    version: stable
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

**Similarly, created node-app-canary-service.yaml to only target the canary version:**

io.k8s.api.core.v1.Service (v1@service.json)

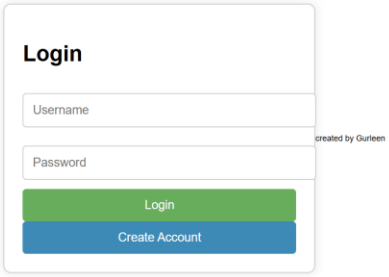
```yaml
apiVersion: v1
kind: Service
metadata:
  name: node-app-canary-service
spec:
  selector:
    app: node-app
    version: canary
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

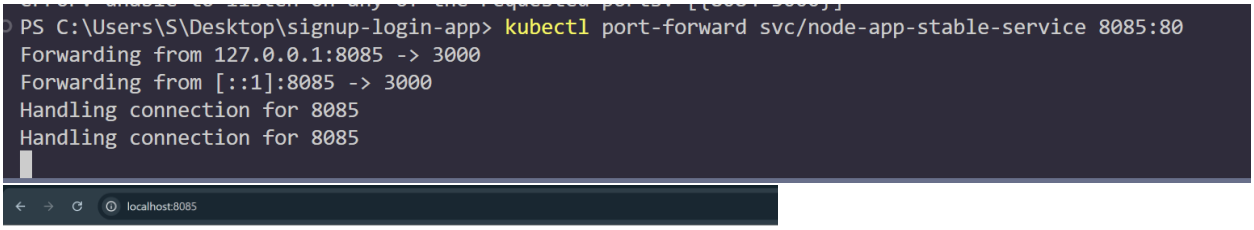
- Manual Traffic Splitting:

1. Can be done via port forwarding: this way app will be redirected to the desired service (stable or canary)

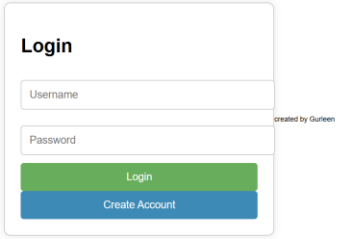
```
PS C:\Users\S\Desktop\signup-login-app> kubectl port-forward svc/node-app-stable-service 8082:80
Forwarding from [::1]:8082 -> 3000
```

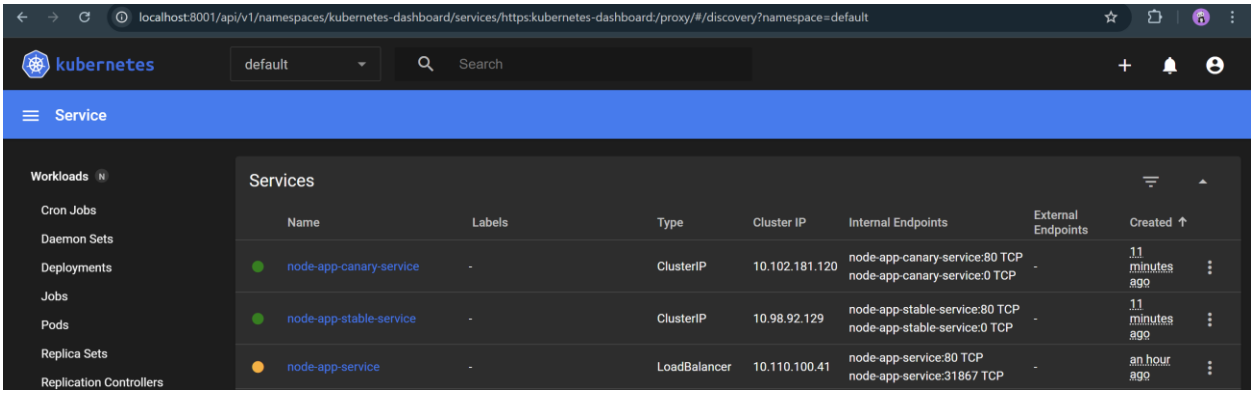

localhost:8082


Login
Username
Password
Login
Create Account
created by Gurleen


PS C:\Users\S\Desktop\signup-login-app> kubectl port-forward svc/node-app-stable-service 8085:80
Forwarding from 127.0.0.1:8085 -> 3000
Forwarding from [::1]:8085 -> 3000
Handling connection for 8085
Handling connection for 8085


localhost:8085


Login
Username
Password
Login
Create Account
created by Gurleen

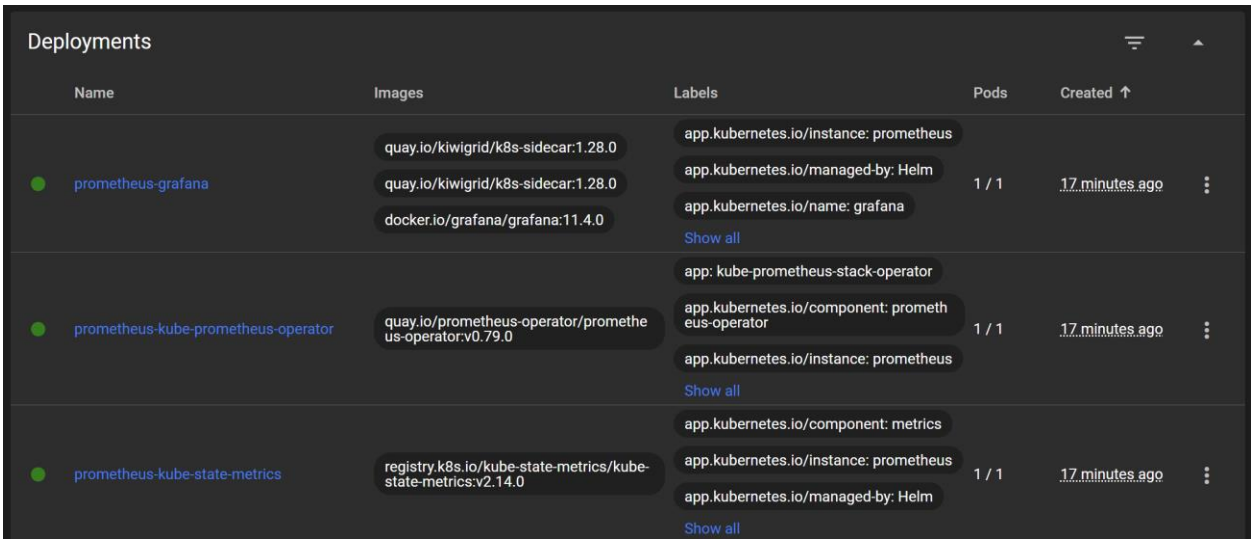


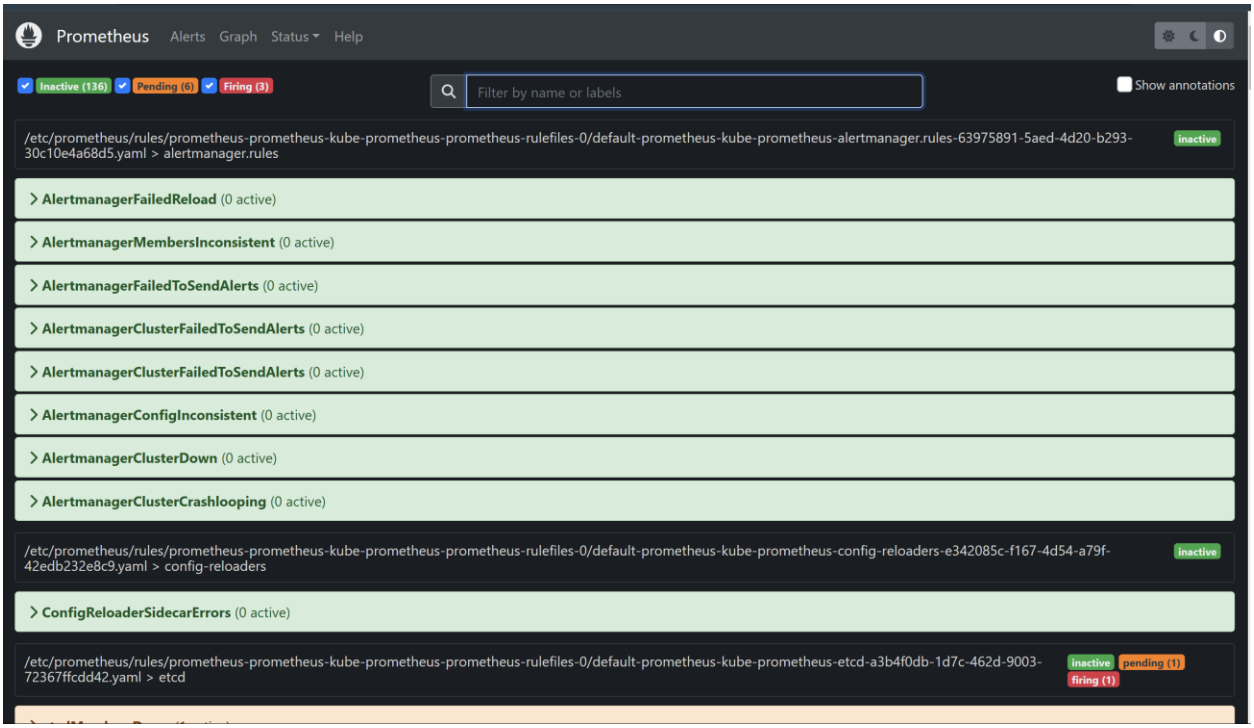

## Part 5- Monitoring and Rollback

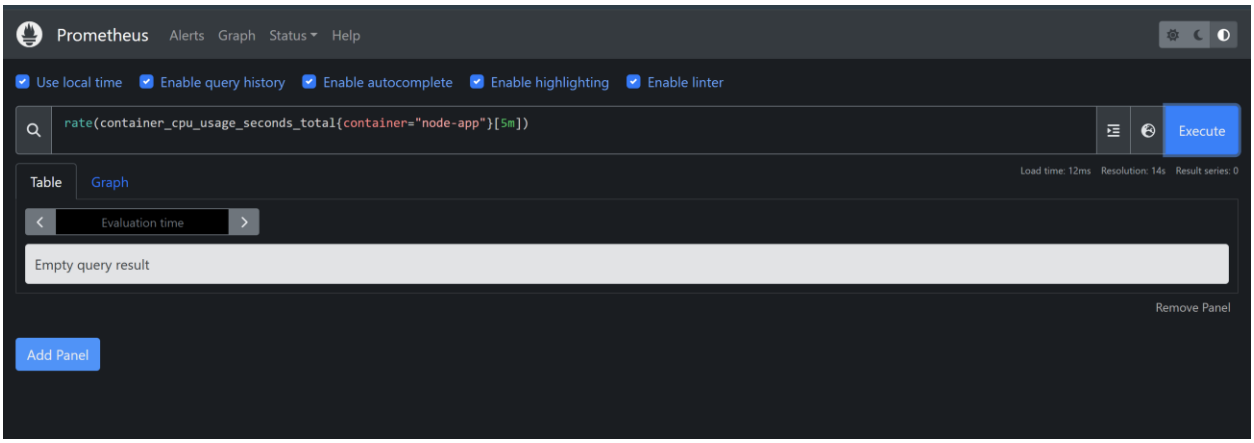
- Monitor Performance

Use monitoring tools (e.g., Kubernetes Dashboard, Prometheus) to observe metrics like response times and error rates.

```
PS C:\Users\S\Desktop\signup-login-app> kubectl port-forward svc/prometheus-operated 9090:9090 -n default
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
```




Prometheus
Alerts
Graph
Status
Help
Inactive (136)
Pending (6)
Firing (3)
Filter by name or labels
Show annotations
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/default-prometheus-kube-prometheus-alertmanager.rules-63975891-5aed-4d20-b293-30c10e4a68d5.yaml > alertmanager.rules
inactive
AlertmanagerFailedReload (0 active)
AlertmanagerMembersInconsistent (0 active)
AlertmanagerFailedToSendAlerts (0 active)
AlertmanagerClusterFailedToSendAlerts (0 active)
AlertmanagerClusterFailedToSendAlerts (0 active)
AlertmanagerConfigInconsistent (0 active)
AlertmanagerClusterDown (0 active)
AlertmanagerClusterCrashlooping (0 active)
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/default-prometheus-kube-prometheus-config-reloaders-e342085c-f167-4d54-a79f-42edb232e8c9.yaml > config-reloaders
inactive
ConfigReloaderSidecarErrors (0 active)
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/default-prometheus-kube-prometheus-etcd-a3b4f0db-1d7c-462d-9003-72367ffcdd42.yaml > etcd
inactive
pending (1)
firing (1)


Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
rate(container_cpu_usage_seconds_total{container="node-app"}[5m])
Execute
Table
Graph
Load time: 12ms
Resolution: 14s
Result series: 0
Evaluation time
Empty query result
Remove Panel
Add Panel

Or use kubectl logs <pod_name> to analyse the logs of the pod

```
PS C:\Users\S\Desktop\signup-login-app> kubectl logs node-app-stable-54cc45f8f5-9bv5s
(node:1) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect sin
ce Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:1) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effe
ct since Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 3000
MongooseServerSelectionError: connect ECONNREFUSED ::1:27017, connect ECONNREFUSED 127.0.0.1:27017
    at _handleConnectionErrors (/app/node_modules/mongoose/lib/connection.js:1089:11)
    at NativeConnection.openUri (/app/node_modules/mongoose/lib/connection.js:1040:11) {
  reason: TopologyDescription {
    type: 'Unknown',
    servers: Map(1) { 'localhost:27017' => [ServerDescription] },
    stale: false,
    compatible: true,
    heartbeatFrequencyMS: 10000,
    localThresholdMS: 15,
    setName: null,
    maxElectionId: null,
    maxSetVersion: null,
    commonWireVersion: 0,
    logicalSessionTimeoutMinutes: null
  },
  code: undefined
}
```

- Rollback if Necessary

Revert to the stable version if performance or reliability issues are detected in the canary version.

kubectl apply -f stable-deployment.yaml

This ensures that the canary traffic is replaced by the stable version again.

```
PS C:\Users\S\Desktop\signup-login-app> kubectl apply -f stable-deployment.yaml
deployment.apps/node-app-stable configured
PS C:\Users\S\Desktop\signup-login-app> kubectl get pods
NAME                                                READY   STATUS    RESTARTS      AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0   2/2     Running   0             22m
mongodb-57b8f5699b-pscfm                            1/1     Running   3 (73m ago)   16h
node-app-canary-7bbfb78979-4mkms                    1/1     Running   0             61m
node-app-canary-7bbfb78979-92vmq                    1/1     Running   0             61m
node-app-canary-7bbfb78979-94sk2                    1/1     Running   0             61m
node-app-stable-54cc45f8f5-pbpqn                    1/1     Running   0             3s
node-app-stable-54cc45f8f5-smlwn                    1/1     Running   0             3s
node-app-stable-54cc45f8f5-wg7kr                    1/1     Running   0             3s
prometheus-grafana-75b6db5f48-7mwqj                 3/3     Running   0             22m
prometheus-kube-prometheus-operator-b46dc5ccc-c64pf      1/1     Running   0             22m
prometheus-kube-state-metrics-6489887dc-dpzcp       1/1     Running   0             22m
prometheus-prometheus-kube-prometheus-prometheus-0       2/2     Running   0             22m
prometheus-prometheus-node-exporter-v8vzc           1/1     Running   0             22m
PS C:\Users\S\Desktop\signup-login-app>
```

### 3. Monitor the Rollback

You can check the status of the rollback by monitoring the rollout status:

kubectl rollout status deployment/node-app-stable

```
PS C:\Users\S\Desktop\signup-login-app> kubectl rollout status deployment/node-app-stable
deployment "node-app-stable" successfully rolled out
```

## Part 6- Full Rollout

- Promote the Canary

Once the canary proves stable, update the deployment to replace the stable version with the canary as the primary application

**Scale the Canary Deployment:**

- Gradually increase the number of replicas for the canary deployment. For example, if you started with 1 replica, you can increase it to 2 or more, depending on your desired traffic distribution.

kubectl scale deployment canary-app --replicas=3

```
PS C:\Users\S\Desktop\signup-login-app> kubectl scale deployment node-app-canary --replicas=3  # Adjust replica count as needed
deployment.apps/node-app-canary scaled
```

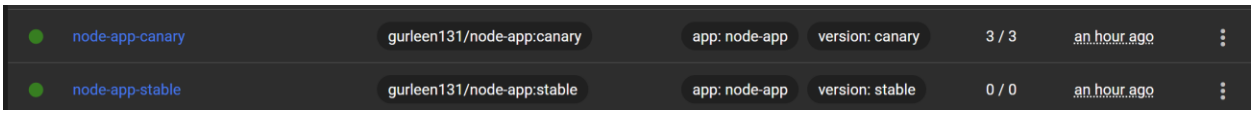
**Scale Down the Stable Deployment:**

- At the same time, decrease the replicas for the stable version. This ensures that as the canary gets more traffic, the stable version's traffic is reduced.

kubectl scale deployment stable-app --replicas=0

```
PS C:\Users\S\Desktop\signup-login-app> kubectl scale deployment node-app-stable --replicas=0  # Scale down the stable version
deployment.apps/node-app-stable scaled
```

```
PS C:\Users\S\Desktop\signup-login-app> kubectl get pods
NAME                                                     READY   STATUS        RESTARTS      AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0   2/2     Running       0             19m
mongodb-57b8f5699b-pscfm                                 1/1     Running       3 (71m ago)   16h
node-app-canary-7bbfb78979-4mkms                         1/1     Running       0             59m
node-app-canary-7bbfb78979-92vmq                         1/1     Running       0             59m
node-app-canary-7bbfb78979-94sk2                         1/1     Running       0             59m
node-app-stable-54cc45f8f5-9bv5s                         1/1     Terminating   0             60m
node-app-stable-54cc45f8f5-bhkmx                         1/1     Terminating   0             60m
node-app-stable-54cc45f8f5-wjd6g                         1/1     Terminating   0             60m
prometheus-grafana-75b6db5f48-7mwqj                      3/3     Running       0             20m
prometheus-kube-prometheus-operator-b46dc5ccc-c64pf      1/1     Running       0             20m
prometheus-kube-state-metrics-6489887dc-dpzcp            1/1     Running       0             20m
prometheus-prometheus-kube-prometheus-prometheus-0       2/2     Running       0             19m
prometheus-prometheus-node-exporter-v8vzc                1/1     Running       0             20m
```




After full rollout :

**Update the Service:**

- Update the Kubernetes Service if necessary to ensure that all traffic is routed to the canary (now the stable version).

kubectl apply -f service.yaml