Lab 4 — Circuit Breaker with NGINX Plus on Kubernetes

Module

TSM_CloudSys — Cloud Services and Systems (2025–2026)

---

1. Introduction

This laboratory session introduces the **Circuit Breaker pattern** and its practical implementation using **NGINX Plus** deployed on **Kubernetes**.

The objective is to protect a backend service from overload or failure by:

- Detecting unhealthy states using **active health checks**
- Stopping traffic to failing services (**Open state**)
- Redirecting traffic to a **fallback service**

A mock backend service implemented in **Node.js** is used to simulate normal operation and overload conditions.

---

2. Learning Objectives

At the end of this lab, you will be able to:

- Explain the Circuit Breaker pattern and its states
- Deploy services on Kubernetes using Deployments and Services
- Configure **NGINX Plus** as a Circuit Breaker
- Use Kubernetes **ConfigMaps** and **Secrets**
- Observe and analyze failure handling and traffic redirection

---

3. Architecture Overview

Components

Component	Description
Mock Webserver	Simulated backend service with overload mode
Fallback Service	Replacement service during failures

Component	Description
NGINX Plus	Circuit Breaker and reverse proxy
Kubernetes	Orchestration platform

Mock Webserver Endpoints

Endpoint	Purpose	Normal Mode	Overload Mode
`/`	Main service	200 OK	30s delay + 503
`/ready`	Circuit Breaker health	200 OK	503
`/alive`	Kubernetes liveness	200 OK	200 OK
`/fakeerrormodeon`	Enable overload	—	—
`/fakeerrormodeoff`	Disable overload	—	—

---

4. Prerequisites

Required Tools

- Kubernetes cluster (Minikube recommended)
- `kubectl`
- Web browser
- `curl`

Provided Files

- `circuitbreaker.zip`
- `mock-webserver-deployment.yaml`
- `circuitbreaker.yaml`
- `nginx-configmap.yaml`
- `docker-token`

---

5. Task 1 — Kubernetes Cluster Setup

5.1 Start Minikube

```
minikube start
```

5.2 Configure kubectl

```
alias kubectl="minikube kubectl --"
```

5.3 Verify Cluster Status

```
minikube status
kubectl get nodes
```

Expected node status: `Ready`

5.4 Start Dashboard (Optional)

```
minikube dashboard
```

---

6. Task 2 — Deploy Services and Circuit Breaker

6.1 Deploy the Mock Webserver

```
kubectl create -f mock-webserver-deployment.yaml
```

Verify pod:

```
kubectl get pods -o wide
```

6.2 Expose the Service

```
kubectl expose deployment mock-webserver-deployment
  --port=80 --type=LoadBalancer --name mock-service
```

For Minikube:

```
minikube tunnel
```

Retrieve external IP:

```
kubectl get services --watch
```

6.3 Test Endpoints

```
curl -v http://<external-ip>/
curl -v http://<external-ip>/alive
curl -v http://<external-ip>/ready
```

6.4 Activate Overload Mode

```
curl --data "" --dump-header - http://<external-ip>/fakeerrormodeon
```

Re-test all endpoints and observe:

- `/` → delayed 503
- `/ready` → 503
- `/alive` → 200

Disable overload:

```
curl --data "" http://<external-ip>/fakeerrormodeoff
```

Deliverable: Document endpoint behavior in normal vs overload mode.

---

7. Deploy the Fallback Service

Create a new deployment file:

`fallback-webserver-deployment.yaml`

Changes from original:

- Deployment name → `fallback-webserver-deployment`
- App label → `fallback-webserver`
- Container name → `fallback-webserver`

Deploy and expose:

```
kubectl create -f fallback-webserver-deployment.yaml
kubectl expose deployment fallback-webserver-deployment
  --port=80 --type=LoadBalancer --name fallback-service
```

Verify accessibility.

Deliverable: Include `fallback-webserver-deployment.yaml` in the report.

---

8. Deploy the Circuit Breaker (NGINX Plus)

8.1 NGINX Circuit Breaker Configuration

Key features:

- Active health checks every 3 seconds on `/ready`
- Immediate failover to fallback service
- Two states: Closed and Open

8.2 Create ConfigMap

```
kubectl create -f nginx-configmap.yaml
```

8.3 Create Docker Registry Secret

```
kubectl create secret generic regcred
  --from-file=.dockerconfigjson=docker-token
  --type=kubernetes.io/dockerconfigjson
```

Verify:

```
kubectl get secret regcred -o yaml
```

8.4 Deploy NGINX Circuit Breaker

```
kubectl create -f circuitbreaker.yaml
```

Retrieve external IP:

```
kubectl get services --watch
```

Test routing:

```
curl -v http://<circuitbreaker-ip>/
```

8.5 Inspect Logs

```
kubectl logs pod/circuitbreaker-<id>
```

Deliverable: Include Circuit Breaker logs in the report.

---

9. Task 3 — Exercising the Circuit Breaker

Steps

1. Enable overload on mock service
2. Wait 3 seconds
3. Send request to Circuit Breaker
4. Observe response source
5. Inspect logs
6. Disable overload and repeat

Expected Behavior

Backend State	Traffic Routed To
Healthy	mock-service
Overloaded	fallback-service

Deliverable: Document observations and state transitions.

---

10. Cleanup

Delete Resources

```
kubectl delete service/circuitbreaker-service
kubectl delete service/fallback-service
kubectl delete service/mock-service
kubectl delete deployment/circuitbreaker-deployment
kubectl delete deployment/fallback-webserver-deployment
kubectl delete deployment/mock-webserver-deployment
```

```
kubectl delete configmap/nginx-configuration
kubectl delete secret/regcred
```

Stop and Delete Cluster

```
minikube stop
minikube delete
```

---

11. Notes for Report

Include:

- Architecture explanation
- Endpoint behavior tables
- YAML configurations
- Logs and observations
- Short conclusion on Circuit Breaker effectiveness

End of Lab 4