

# Online Bookstore Microservices Project

*Project Progress Report – Checkpoint 3*

**Prepared For:** Prof. Pedram Habibi

**Prepared By:** Group 9

**Group Members:**

Helly Rajeshbhai Patel

Jiyad Mohammed

Arif Shaikh

Nicholas Nwanua Ilechie

Nirajbhai Ranchhodbhai Limbasiya

**Date Submitted:** July 24<sup>th</sup>, 2025

“I declare that the attached assessment is wholly my own work in accordance with Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students.”

## Introduction

This report presents the work completed for Checkpoint 3 of the Service-Oriented Architecture (SOA) course project. We deployed the User and Product Services to a local Kubernetes cluster using Minikube. System monitoring was implemented using Prometheus and Grafana, and a Horizontal Pod Autoscaler (HPA) was configured for the User Service to enhance scalability. Networking between pods was tested to ensure reliable inter-service communication. All resources were organized within the dedicated bookstore namespace, and resource specification issues were resolved to achieve stable operation. This introduction outlines the scope of work and deliverables for Phase 1 of the online bookstore microservices project, including kubectl outputs, monitoring dashboards, and networking validation results.

# Kubernetes Deployment

We deployed the User and Product Services in the bookstore namespace using Kubernetes manifests, including Deployment, Service, ConfigMap, and Secret objects.

## User Service

The User Service, developed using Node.js and Express.js, provides functionalities for user registration and retrieval through the following endpoints: POST /users/register, GET /users, and GET /users/{id}. The service is deployed with two replicas to enhance availability and resilience. Resource requests and limits have been defined to ensure efficient allocation of CPU and memory resources, preventing starvation and maintaining stability.

An excerpt from the deployment configuration (user-service-deployment.yaml) is shown below:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: user-service
  namespace: bookstore
spec:
  replicas: 2
  selector:
    matchLabels:
      app: user-service
  template:
    metadata:
      labels:
        app: user-service
    spec:
      containers:
        - name: user-service
          image: user-service:latest
          ports:
            - containerPort: 3000
          resources:
            limits:
              cpu: "500m"
              memory: "512Mi"
            requests:
              cpu: "200m"
              memory: "256Mi"
```

## Product Service

The Product Service is responsible for managing the book catalog and supports endpoints for catalog retrieval: GET /books and GET /books/{id}. Like the User Service, it is deployed with two replicas and identical resource configurations to ensure consistency, scalability, and stable performance.

## Horizontal Pod Autoscaling

A Horizontal Pod Autoscaler (HPA) was configured for the User Service to ensure scalability under variable workloads. The HPA automatically adjusts the number of replicas between 2 and 5 based on a target CPU utilization of 70%, utilizing the Minikube metrics server for real-time resource monitoring.

An excerpt from the deployment configuration (user-service-hpa.yaml) is shown below:

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: user-service-hpa
  namespace: bookstore
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: user-service
  minReplicas: 2
  maxReplicas: 5
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 70
```

## Monitoring

Prometheus was configured to scrape metrics from both the User Service and Product Service at 15-second intervals, with settings managed via a dedicated ConfigMap.

Grafana was integrated for visualization, providing real-time monitoring through custom dashboards. These dashboards are accessible via NodePort services, enabling performance tracking and health assessment of the deployed services.

An excerpt of the Prometheus configuration (prometheus-config.yaml) is shown below:

```
data:
  prometheus.yml: |
    global:
      scrape_interval: 15s
    scrape_configs:
      - job_name: 'user-service'
        static_configs:
          - targets: ['user-service:80']
      - job_name: 'product-service'
        static_configs:
          - targets: ['product-service:80']
```

## 5. Networking

Inter-service communication was validated using a BusyBox pod, as the node:18-alpine image does not include curl. The following command confirmed that the User Service could be reached from within the cluster:

```
kubectrl run -i --tty test-pod --image=busybox --restart=Never -n bookstore -- sh
wget -q -O- http://user-service:80/users
Output:
[{"id": "1", "username": "johndoe", "email": "john@example.com"}, ...]
```

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: grafana
5   namespace: bookstore
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      app: grafana
11   template:
12     metadata:
13       labels:
14         app: grafana
15     spec:
16       containers:
17       - name: grafana
18         image: grafana/grafana:9.5.2
19         ports:
20         - containerPort: 3000

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:59069
CoreDNS is running at https://127.0.0.1:59069/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get pods -n bookstore
NAME                                READY   STATUS    RESTARTS   AGE
grafana-6f088c4747-r17ns             1/1     Running   0           9h
load-generator                        1/1     Running   0           9h
product-service-5dfbcb586-bfrnz       1/1     Running   0           9h
product-service-5dfbcb586-rzz2g       1/1     Running   0           9h
prometheus-609f08ff-2hwup            1/1     Running   0           9h
user-service-574f55db9-q7ksb         1/1     Running   0           9h
user-service-574f55db9-tfmm          1/1     Running   0           9h
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl logs user-service-574f55db9-q7ksb -n bookstore
> user-service@1.0.0 start
> node index.js

User Service running on port 3000
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl logs <user-service-574f55db9-q7ksb> -n bookstore
zsh: no such file or directory: user-service-574f55db9-q7ksb
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get pods -n bookstore
NAME                                READY   STATUS    RESTARTS   AGE
grafana-6f088c4747-r17ns             1/1     Running   0           9h
load-generator                        1/1     Running   0           9h
product-service-5dfbcb586-bfrnz       1/1     Running   0           9h
product-service-5dfbcb586-rzz2g       1/1     Running   0           9h
prometheus-609f08ff-2hwup            1/1     Running   0           9h
user-service-574f55db9-q7ksb         1/1     Running   0           9h
user-service-574f55db9-tfmm          1/1     Running   0           9h
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get services -n bookstore
NAME                                TYPE               CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
grafana                            NodePort           10.100.119.125   <none>            80:31917/TCP     9h
product-service                    ClusterIP          10.100.23.56    <none>            80/TCP           9h
prometheus                         NodePort           10.99.93.197    <none>            80:31667/TCP     9h
user-service                       ClusterIP          10.100.232.237  <none>            80/TCP           9h

```

Figure: kubectl Outputs

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: grafana
5   namespace: bookstore
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      app: grafana
11   template:
12     metadata:
13       labels:
14         app: grafana
15     spec:
16       containers:
17       - name: grafana
18         image: grafana/grafana:9.5.2

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:59069
CoreDNS is running at https://127.0.0.1:59069/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get pods -n bookstore
NAME                                READY   STATUS    RESTARTS   AGE
grafana-6f088c4747-r17ns             1/1     Running   0           9h
load-generator                        1/1     Running   0           9h
product-service-5dfbcb586-bfrnz       1/1     Running   0           9h
product-service-5dfbcb586-rzz2g       1/1     Running   0           9h
prometheus-609f08ff-2hwup            1/1     Running   0           9h
user-service-574f55db9-q7ksb         1/1     Running   0           9h
user-service-574f55db9-tfmm          1/1     Running   0           9h
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl logs user-service-574f55db9-q7ksb -n bookstore
> user-service@1.0.0 start
> node index.js

User Service running on port 3000
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl logs <user-service-574f55db9-q7ksb> -n bookstore
zsh: no such file or directory: user-service-574f55db9-q7ksb
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get pods -n bookstore
NAME                                READY   STATUS    RESTARTS   AGE
grafana-6f088c4747-r17ns             1/1     Running   0           9h
load-generator                        1/1     Running   0           9h
product-service-5dfbcb586-bfrnz       1/1     Running   0           9h
product-service-5dfbcb586-rzz2g       1/1     Running   0           9h
prometheus-609f08ff-2hwup            1/1     Running   0           9h
user-service-574f55db9-q7ksb         1/1     Running   0           9h
user-service-574f55db9-tfmm          1/1     Running   0           9h
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices % kubectl get hpa user-service-hpa -n bookstore
NAME                REFERENCE          TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
user-service-hpa    Deployment/user-service  47%/70%      2         5         2           9h
(base) nicholaslechie@Nicholas-MacBook-Pro online-bookstore-microservices %

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

configmap/user-service-config created
deployment.apps/user-service created
horizontalpodautoscaler.autoscaling/user-service-hpa created
secret/user-service-secret created
service/user-service created
● (base) nicholasilechie@Nicholass-MacBook-Pro online-bookstore-microservices % kubectl get pods -n bookstore
kubectl get services -n bookstore
NAME                                READY   STATUS    RESTARTS   AGE
grafana-6fd88c4747-r17ns            1/1     Running   0           18s
product-service-5dfbcbc586-bfrnz    1/1     Running   0           18s
product-service-5dfbcbc586-rzz2g    1/1     Running   0           18s
prometheus-6b98fd8fff-2hkwp        1/1     Running   0           18s
user-service-574f55db9-q7ksb        1/1     Running   0           18s
user-service-574f55db9-tfmmm        1/1     Running   0           18s
NAME                                TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)          AGE
grafana                             NodePort   10.100.119.125 <none>       80:31917/TCP    18s
product-service                     ClusterIP  10.106.23.56  <none>       80/TCP           18s
prometheus                           NodePort   10.98.93.187  <none>       80:31667/TCP    18s
user-service                         ClusterIP  10.108.232.237 <none>       80/TCP           18s
○ (base) nicholasilechie@Nicholass-MacBook-Pro online-bookstore-microservices %
```

Figure : Kubernetes Cluster



## **Conclusion**

Checkpoint 3 achieved the successful deployment of the User and Product Services to a Kubernetes cluster, complete with monitoring, autoscaling, and networking capabilities. Resource configuration errors were resolved, and all components were organized within the dedicated bookstore namespace to ensure stability and maintainability.

With these foundational services in place, we are now prepared to advance to Phase 2, which will involve the implementation of Order and Notification Services and migration of the deployment to a cloud provider. We welcome constructive feedback to further optimize our implementation.