**Step 1: Prerequisites**

Ensure your students have the following set up on their Windows machines:

1. **Install Minikube**  
   Guide: Minikube Installation
2. **Install kubectl**  
   Guide: kubectl Installation
3. **Start Minikube**  
   Run the following in PowerShell or Command Prompt:

bash

CopyEdit

minikube start

**Step 2: Create Namespaces**

Instruct students to create three namespaces using kubectl:

1. Open a terminal and run:

bash

CopyEdit

kubectl create namespace appone

kubectl create namespace elasticsearch

kubectl create namespace dbsql

**Step 3: Write the YAML Configuration Files**

Students should create a file named k8s-namespace-deployment.yml with the following content:

# Namespace: appone (Nginx Service)

apiVersion: v1

kind: PersistentVolume

metadata:

name: nginx-pv

namespace: appone

spec:

capacity:

storage: 1Gi

accessModes:

- ReadWriteMany

hostPath:

path: /Users/janpannees/Documents/nginx\_files

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: nginx-pvc

namespace: appone

spec:

accessModes:

- ReadWriteMany

resources:

requests:

storage: 1Gi

---

apiVersion: v1

kind: Service

metadata:

name: nginx-service

namespace: appone

spec:

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

namespace: appone

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

volumeMounts:

- name: nginx-html

mountPath: /usr/share/nginx/html

- name: nginx-config-volume

mountPath: /etc/nginx/nginx.conf

subPath: nginx.conf

volumes:

- name: nginx-html

persistentVolumeClaim:

claimName: nginx-pvc

- name: nginx-config-volume

configMap:

name: nginx-config

---

apiVersion: v1

kind: ConfigMap

metadata:

name: nginx-config

namespace: appone

data:

nginx.conf: |

http {

server {

listen 80;

location /elasticsearch {

proxy\_pass http://elasticsearch-service.elasticsearch.svc.cluster.local:9200;

}

location /db {

proxy\_pass http://mysql-service.dbsql.svc.cluster.local:3306;

}

location / {

root /usr/share/nginx/html;

index index.html;

}

}

}

# Namespace: elasticsearch

apiVersion: apps/v1

kind: Deployment

metadata:

name: elasticsearch-deployment

namespace: elasticsearch

spec:

replicas: 1

selector:

matchLabels:

app: elasticsearch

template:

metadata:

labels:

app: elasticsearch

spec:

containers:

- name: elasticsearch

image: docker.elastic.co/elasticsearch/elasticsearch:7.17.5

ports:

- containerPort: 9200

- containerPort: 9300

env:

- name: discovery.type

value: single-node

# Namespace: dbsql (MySQL)

apiVersion: apps/v1

kind: Deployment

metadata:

name: mysql-deployment

namespace: dbsql

spec:

replicas: 1

selector:

matchLabels:

app: mysql

template:

metadata:

labels:

app: mysql

spec:

containers:

- name: mysql

image: mysql:5.7

ports:

- containerPort: 3306

env:

- name: MYSQL\_ROOT\_PASSWORD

value: "rootpassword"

---

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: dbsql

spec:

selector:

app: mysql

ports:

- protocol: TCP

port: 3306

targetPort: 3306

type: ClusterIP

**Steps to Set Up**

1. **Create the Local Directory**  
   Ensure the directory /Users/janpannees/Documents/nginx\_files exists on your machine:

bash

CopyEdit

mkdir -p /Users/janpannees/Documents/nginx\_files

1. **Add HTML Files**  
   Place your index.html and any other HTML files in /Users/janpannees/Documents/nginx\_files.
2.  **Start Minikube with Mounting**  
   Start Minikube and mount the directory for Kubernetes to access it:

minikube start

minikube mount /Users/janpannees/Documents/nginx\_files:/Users/janpannees/Documents/nginx\_files

1.  **Access the Nginx Service**  
   Test the Nginx service:

minikube service nginx-service -n appone

**Step 4: Deploy the Configurations**

1. Save the file as k8s-namespace-deployment.yml.
2. Apply it using:

bash

CopyEdit

kubectl apply -f k8s-namespace-deployment.yml

**Step 5: Verify Deployments**

1. Check the namespaces:

bash

CopyEdit

kubectl get namespaces

1. Check resources in each namespace:

bash

CopyEdit

kubectl get all -n appone

kubectl get all -n elasticsearch

kubectl get all -n dbsql

**Bonus: Access Services**

1. To access Nginx or Elasticsearch:

bash

CopyEdit

minikube service nginx-service -n appone

minikube service elasticsearch-deployment -n elasticsearch

1. For MySQL, you can use tools like MySQL Workbench to connect using the mysql-service IP and port.