**Upskill results:**

**Task 1 - Volume HostPath**

A hostPath volume maps a directory or file from the host node's filesystem into a pod. I’m mounting /etc/kubernetes/ from the node into the pod at /host/kubernetes/.

Obraz zawierający tekst, zrzut ekranu, Czcionka, menu

Opis wygenerowany automatycznie

Key Fields:

hostPath.path: /etc/kubernetes (directory on the host node).

volumeMounts.mountPath: /host/kubernetes (where it will be accessible inside the pod).

type: Specifies the type of the host path (Directory ensures the path exists and is a directory).

kubectl exec -it hostpath-pod -n upskill – sh

ls -l /host/kubernetes/

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

The contents of /host/kubernetes/ are files and directories from the host node's /etc/kubernetes directory. Typically, this directory contains configuration files for the Kubernetes cluster running on the host. These may include:

**kubeadm-config.yaml**: Configuration for kubeadm.

**Certificates**: Used by Kubernetes components to authenticate securely.

**manifests/**: Static pod manifests for control plane components (e.g., kube-apiserver, kube-scheduler).

Using hostPath can be a security risk:

1. **Host Dependency**: Tightly couples your pod to the node's filesystem, reducing portability.
2. **Access Control**: Pods may inadvertently gain elevated access to sensitive host files.
3. **Cluster Misconfigurations**: Malicious containers can modify critical host files.

**Task 2 - Volume EmptyDir**

This task explores using an emptyDir volume to temporarily persist data during container restarts but not across pod recreations.

kubectl exec -it nginx-deployment-689b9b776b-hpfcm -n upskill -- sh

nginx-deployment-689b9b776b-hpfcm

Obraz zawierający zrzut ekranu, Czcionka, linia, tekst

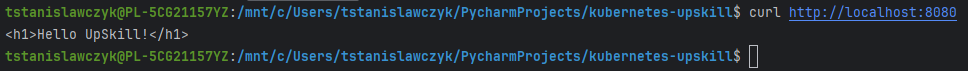
Opis wygenerowany automatycznie

kubectl port-forward service/nginx-service 8080:80 -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

curl <http://localhost:8080>



kubectl rollout restart deployment nginx-deployment -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznieThe change to the welcome page does not persist because it was written to the container’s ephemeral storage.

Update nginx-deployment.yaml to include the emptyDir volume:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl exec -it <nginx-pod-name> -n upskill -- sh

echo '<h1>Hello UpSkill with emptyDir!</h1>' > /usr/share/nginx/html/index.html

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

kubectl rollout restart deployment nginx-deployment -n upskill

curl localhost…

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Data stored in a Volume of type emptyDir will survive a container restart, but will not survive the removal of the Pod.

**Task 3 – PersistentVolumeClaim**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

**accessModes**: ReadWriteOnce allows a single pod to read/write.

**resources.requests.storage**: Requests 100Mi of storage.

**storageClassName**: Uses the standard StorageClass (provided by rancher.io/local-path in your Kind cluster).

kubectl get pvc -n upskill



Deployment update

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl exec -it <nginx-pod-name> -n upskill – sh

echo '<h1>Hello UpSkill with PersistentVolume!</h1>' > /usr/share/nginx/html/index.html

kubectl port-forward service/nginx-service 8080:80 -n upskill

curl <http://localhost:8080>

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Even though pod and pvc were created correctly, pvc was still stuck in pending state. I tried to use different methods, restarts and etc., probably I didn’t add namespace in the deployment.

kubectl get pvc -n upskill



kubectl exec -it <pod-name> -n upskill -- sh

echo '<h1>Persistent Volume Test</h1>' > /usr/share/nginx/html/index.html

kubectl rollout restart deployment nginx-deployment -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

I can find the file in a brand new pod.

**Task 4 – StatefulSet**

Create the headless service

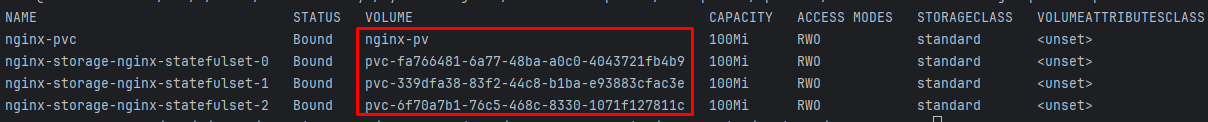
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

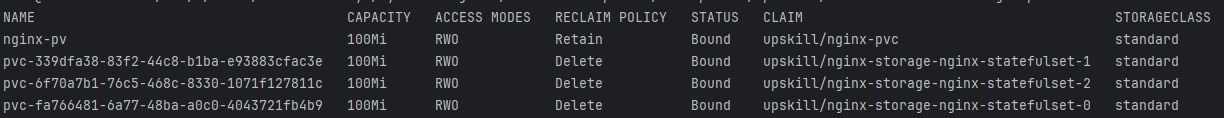
kubectl get statefulset nginx-statefulset -n upskill



kubectl get pvc -n upskill



kubectl get pv



Each PVC have a corresponding PV, bound and available.

kubectl delete pod nginx-statefulset-1 -n upskill

kubectl get pods -n upskill  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

The pod was recreated with the same name (nginx-statefulset-1) and reattached to its volume.

DNS Test  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

kubectl exec -it netshoot -n upskill – sh

nslookup nginx-statefulset-0.nginx-service

nslookup nginx-statefulset-1.nginx-service

nslookup nginx-statefulset-2.nginx-service

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Each command resolved to the IP of the respective pod.

**Task 5 - Init Container**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

InitContainer runs before the main Nginx container starts. It writes a custom HTML file (index.html) to the /usr/share/nginx/html directory.

Volume (emptyDir) is used between the InitContainer and the Nginx container. The InitContainer writes the HTML file to the volume and the main container reads the file from the same volume.

Main Nginx Container finally serves the custom HTML file modified by the InitContainer.

kubectl describe pod nginx-init -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl port-forward pod/nginx-init 8080:80 -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka, Grafika

Opis wygenerowany automatycznie

Key Benefits of Using InitContainers

* One-Time Setup: Useful for tasks like downloading files, initializing configuration, or setting up dependencies.
* Isolation: Keeps initialization logic separate from the main application container.
* Error Handling: If the InitContainer fails, Kubernetes prevents the main container from starting.

**Task 6 - Multiple Containers**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get pods -n upskill



kubectl exec -it multi-container-deployment-9fc9ccf8f-kq9mh -n upskill -c logs – sh

wget -qO- http://localhost:80

Obraz zawierający tekst, Czcionka, zrzut ekranu, design

Opis wygenerowany automatycznie

Both containers share the same network namespace within the pod. The logs container can communicate with the app container using localhost and the port exposed by the app container.

The app container runs Nginx, while the logs container provides tools to debug or interact with the application. If needed, both containers can share data using a shared volume.

**Task 7 – non-parallel job**

A Job in Kubernetes is designed to run a specific task to completion.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

kubectl get jobs -n upskill

Obraz zawierający zrzut ekranu, tekst, Czcionka, linia

Opis wygenerowany automatycznie

kubectl get pods -n upskill --selector=job-name=print-hostname-job   
(or without selector)



kubectl logs print-hostname-job-gt7zv -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

The Job ran a single instance of the busybox container, executed the hostname command, and completed successfully. Logs confirmed that the hostname was output by the pod.

**Task 8 – parallel job**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get jobs -n upskill





kubectl logs print-hostname-parallel-job-9-stw8c -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

**Task 9 – Cron Job**

A CronJob in Kubernetes is a scheduled task that runs periodically based on a cron expression. This task requires modifying a previous Job to be executed every 5 minutes and learning how to suspend and resume the CronJob.

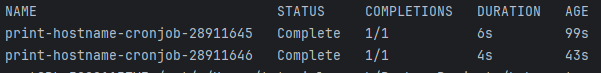
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get cronjob -n upskill



kubectl get jobs -n upskill



Suspend:

kubectl patch cronjob print-hostname-cronjob -n upskill -p '{"spec": {"suspend": true}}'



Resume:

kubectl patch cronjob print-hostname-cronjob -n upskill -p '{"spec": {"suspend": false}}'

**Task 10 - Service Account**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

This task involves creating a Service Account, using it in a Deployment, and exploring how to manage the token authentication to the API Server.

kubectl get serviceaccounts -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get pods -n upskill



kubectl exec -it nginx-deployment-sa-dfb4f9cbb-h7k6v -n upskill – sh

Inside the pod, inspect the token file (mounted automatically):

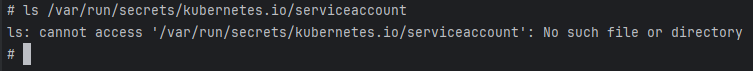
cat /var/run/secrets/kubernetes.io/serviceaccount/token

The output is the token that the pod uses to authenticate with the Kubernetes API.

disable automatic token mounting for the Service AccountObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

kubectl rollout restart deployment nginx-deployment-sa -n upskill



The token is no longer available.

Service Account is Kubernetes resource used to provide pod-level authentication to the API Server. Each pod running under a Service Account inherits its permissions.

A token is automatically generated for the Service Account and mounted in the pod. Disabling token mounting improves security by preventing unnecessary API access.

Pods will no longer have direct access to the Kubernetes API unless I explicitly provide authentication credentials.