**Upskill results:**

**Task 1 - Labels & Annotations**

kubectl get nodes --show-labels

kubectl describe node local-cluster-control-plane | grep -A5 "Annotations"

kubectl describe node local-cluster-worker | grep -A5 "Annotations"

kubectl describe node local-cluster-worker2 | grep -A5 "Annotations"

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

I’m starting from scratch

Kind-config.yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

kind create cluster --name lab-cluster --config kind- config.yaml

Namespace-sprint2

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

kubectl apply -f namespace-upskill.yaml

Create Pods with different labels and annotations  
pod1

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

Pod2

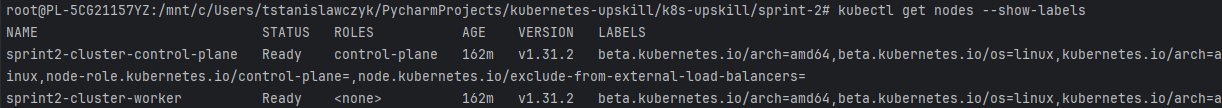
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

kubectl apply -f pod1.yaml

kubectl apply -f pod2.yaml

kubectl get nodes --show-labels



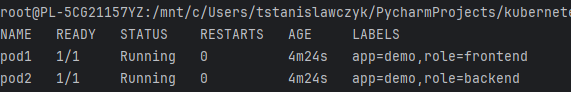
kubectl get namespace upskill --show-labels

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

I wish I could change the namespace name to i.e. “sprint2”, but that’s not possible

kubectl get pods -n upskill --show-labels

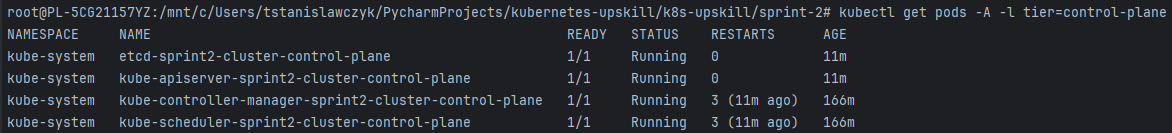


kubectl describe pod pod1 -n upskill | grep Annotations -A 1

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

kubectl get pods -A -l tier=control-plane



kubectl get pods -A

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Task 2 - ConfigMaps & Secrets**

Option 1 – use --from-literal  
kubectl create configmap settings \

--namespace=upskill \

--from-literal=config.yaml="auth\_enabled: false\nserver:\n http\_listen\_port: 3100"

kubectl get configmap settings -n upskill -o yaml

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

kubectl delete configmap settings -n upskill

Option 2 – use --from-file

Configmap.yaml  
Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

kubectl create configmap settings \

--namespace=upskill \

--from-file=config.yaml

kubectl get configmap settings -n upskill -o yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Create a secret**

Option 1 – use --from-literal

openssl rand -base64 16

kubectl create secret generic dbcredentials \

--namespace=upskill \

--from-literal=password= Xi8gVaeGAEtHL1PQHj9nWw==

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

kubectl get secret dbcredentials -n upskill -o yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl get secret dbcredentials -n upskill -o jsonpath="{.data.password}" | base64 --decode > db-password.txt

cat db-password.txt

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

kubectl delete secret dbcredentials -n upskill

Option 2 – use --from-file

echo "Xi8gVaeGAEtHL1PQHj9nWw==" > password.txt

kubectl create secret generic dbcredentials \

--namespace=upskill \

--from-file=password=password.txt

kubectl get secret dbcredentials -n upskill -o yaml

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl get secret dbcredentials -n upskill -o jsonpath="{.data.password}" | base64 --decode > db-password-final.txt

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

I think I need to spend some time understanding the configmap concepts. The documentation says:  
There are four different ways that you can use a ConfigMap to configure a container inside a Pod:

1. Inside a container command and args
2. Environment variables for a container
3. Add a file in read-only volume, for the application to read
4. Write code to run inside the Pod that uses the Kubernetes API to read a ConfigMap

These different methods lend themselves to different ways of modeling the data being consumed. For the first three methods, the [kubelet](https://kubernetes.io/docs/reference/generated/kubelet" \o "" \t "_blank) uses the data from the ConfigMap when it launches container(s) for a Pod.

1. **ConfigMap-Inside-containers-command-and-args**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

I can’t create a pod before creating a configmap  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

When the namespace and the pod is created, I can call configmap values from environment variable inside a container

kubectl exec -it app-command-args --namespace=upskill -- /bin/sh

env | grep AUTH\_ENABLED

env | grep HTTP\_LISTEN\_PORT  
Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

1. **Use a ConfigMap as Environment Variables**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

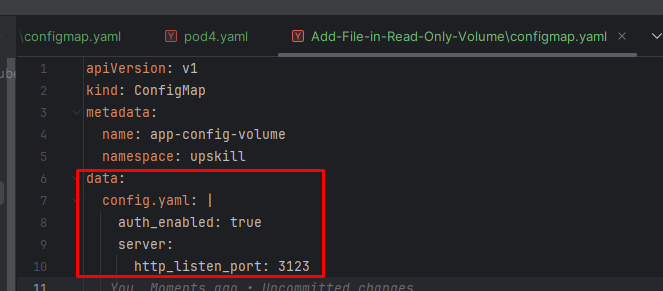
Opis wygenerowany automatycznie

The result seems to be identical, but the way is slightly different

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

1. **Add a file in read-only volume, for the application to read**



Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

**kubectl exec -it app-volume --namespace=upskill -- /bin/sh**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

1. **Write code to run inside the Pod that uses the Kubernetes API to read a ConfigMap**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl exec -it --namespace=upskill python-pod -- pip3 install Kubernetes

kubectl exec -it --namespace=upskill python-pod -- python3 /read\_config.py

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

By default, Kubernetes mounts the required ServiceAccount credentials into pods. However, in some cases (or in custom setups), these mounts might be missing. Explicitly specifying them ensures the Python kubernetes client can authenticate and interact with the API.

The incluster\_config function relies on these files being present at /var/run/secrets/kubernetes.io/serviceaccount.

I need some more understanding of RBAC, but for now let’s test access to the configmap in python app

configmap  
Obraz zawierający tekst, zrzut ekranu, Czcionka, wyświetlacz

Opis wygenerowany automatycznie

reader role  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

reader role binding

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

read\_config.py

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

python-pod (pod7.yaml)

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl exec -it --namespace=upskill python-pod -- pip3 install Kubernetes

kubectl cp read\_config.py python-pod:/read\_config.py --namespace=upskill

kubectl exec -it --namespace=upskill python-pod -- python3 /read\_config.py

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

The configmap is available via API  
This function performs the actual interaction with the Kubernetes API to read a ConfigMap.  
Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

**Task 4 - Commands & Arguments**

In python-pod

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

**Task 5 – Deployments**

kubectl get deployments

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

**READY**: The ratio of available pods to the total number of desired pods. 5/5 means all 5 pods are running and ready.

**UP-TO-DATE**: The number of pods updated to the desired state as per the Deployment. Here, 5 indicates all 5 pods have the latest configuration.

**AVAILABLE**: The number of pods available to handle requests. 5 means all 5 pods are functional and ready to serve.

kubectl rollout status deployment/nginx-deployment

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

kubectl rollout history deployment/nginx-deployment

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Pod names include the Deployment name, a hash to identify the ReplicaSet, and a unique, random suffix.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

kubectl delete pod nginx-deployment-5ff6885787-km2vs --namespace=upskill

Deployments ensure that the specified number of pods are always running. If a pod is deleted, it’s automatically recreated by the ReplicaSet.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Task 6 - Deployments New Image**

kubectl set image deployment/nginx-deployment nginx=nginx:1.21 -n upskill

kubectl rollout status deployment/nginx-deployment -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka, informacja

Opis wygenerowany automatycznie

All pods have been replaced  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl rollout history deployment/nginx-deployment -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

kubectl rollout history deployment/nginx-deployment --revision=2 -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rollback:

kubectl rollout undo deployment/nginx-deployment -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Roll back a Kubernetes Deployment to a specific version **(revision)**

kubectl set image deployment/nginx-deployment nginx=nginx:1.18 -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Check again the revision 2 details (nginx:1.21)

kubectl rollout history deployment/nginx-deployment --revision=2 -n upskill

kubectl rollout undo deployment/nginx-deployment --to-revision=2 -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

it’s better to add --record flag to see the details, but the CLI says it’s been deprecated and will be removed in the future

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Rollout History Limit**:

* By default, Kubernetes keeps a history of the last 10 revisions of a Deployment.
* You can configure this limit by adding the revisionHistoryLimit field to the Deployment spec:

spec:

revisionHistoryLimit: 20

**Task 7 - Deployments Scale**

Scale Using kubectl scale (Imperative Method)

kubectl scale deployment nginx-deployment --replicas=10 -n upskill



Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl scale deployment nginx-deployment --replicas=2 -n upskill



kubectl get deployments -n upskill



Scale Using Manifest Changes (Declarative Method)

Obraz zawierający tekst, Czcionka, zrzut ekranu, design

Opis wygenerowany automatycznie

kubectl apply -f nginx-deployment.yaml

Obraz zawierający zrzut ekranu, tekst, Czcionka, linia

Opis wygenerowany automatycznie

**Task 8 - Deployments Rollout Strategy**

Step 1: Deployment Rollout Strategies

Kubernetes supports two main rollout strategies:

1. Recreate:
   * Deletes all existing pods before creating new ones.
   * Causes downtime because there are no pods running during the transition.
2. RollingUpdate:
   * Replaces pods gradually, ensuring some pods remain available during the update.
   * Default strategy for Deployments.
   * Controlled by maxSurge and maxUnavailable parameters.

Kubectl get pods -w --namespace=upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

maxSurge controls how many extra pods are created during the rollout, maxUnavailable controls how many pods can be unavailable during the rollout. I.e. changing nginx version causes container replacement, maxSurge and maxUnavailable with a big value makes the deployment faster, but it may cause interruptions, so for more safety, I can set lower values.

With Recreate all old pods are terminated before the new pods are created.

**Task 9 – Service**

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

<https://kubernetes.io/docs/concepts/services-networking/service/>

Services providing a stable, consistent way to access a group of Pods. Pods are ephemeral, so that they can be created, destroyed, or rescheduled dynamically, resulting in changing IP addresses. Services solve this problem by providing a permanent, unchanging endpoint to access Pods, even as the Pods' underlying IPs change.

Types:  
ClusterIP (Default) exposes the service only within the cluster and is available via a cluster-internal IP. Used for communication between internal pods/microservices

NodePort exposes the service on a specific port of each node and is accessible externally using NodeIP:NodePort. Useful when direct access to nodes is required or for testing.

LoadBalancer – creates an external load balancer (i.e. AWS ELB) to route traffic to the service. Useful for exposing applications to external users.

ExternalName maps a service to an external DNS name. It doesn’t create endpoints or forward traffic, it resolves the DNS name instead. Useful for proxying external services (i.e. databases or APIs)

**Service vs Pod-to-Pod Communication**

| **Aspect** | **Without Service** | **With Service** |
| --- | --- | --- |
| **Access Method** | Use Pod IPs directly (dynamic and ephemeral). | Use a stable Service IP or DNS name. |
| **Load Balancing** | Manual traffic distribution. | Automatic load balancing. |
| **Resilience** | IPs change when Pods are rescheduled. | IPs/DNS remain constant regardless of Pod changes. |

kubectl expose deployment nginx-deployment --type=ClusterIP --port=80 --name=nginx-service -n upskill

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Nginx-service.yaml

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get services -n upskill



kubectl describe service nginx-service -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl get endpoints nginx-service -n upskill



kubectl get endpointslices -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

kubectl get deployments -n upskill

kubectl get pods -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

kubectl get endpoints nginx-service -n upskill

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Port forwarding in Kubernetes is operation meant for local development or troubleshooting. Kubernetes doesn’t natively support port forwarding as a declarative resource because it’s a client-side operation handled by kubectl. For permanent access I could use a NodePort, LoadBalancer, or Ingress.

**Task 10 – Service DNS**

kubectl run netshoot-pod --image=nicolaka/netshoot --restart=Never -n upskill -- sleep infinity



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

wget -O- <http://nginx-service>

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

wget output confirms the service is accessible using its DNS name and verifies that the service correctly routes traffic to underlying pods

nslookup nginx-service

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

nslookup Output confirms the Service has a DNS record within the cluster.

nginx-service.upskill.svc.cluster.local is the FQDN (fully qualified domain name) for the service