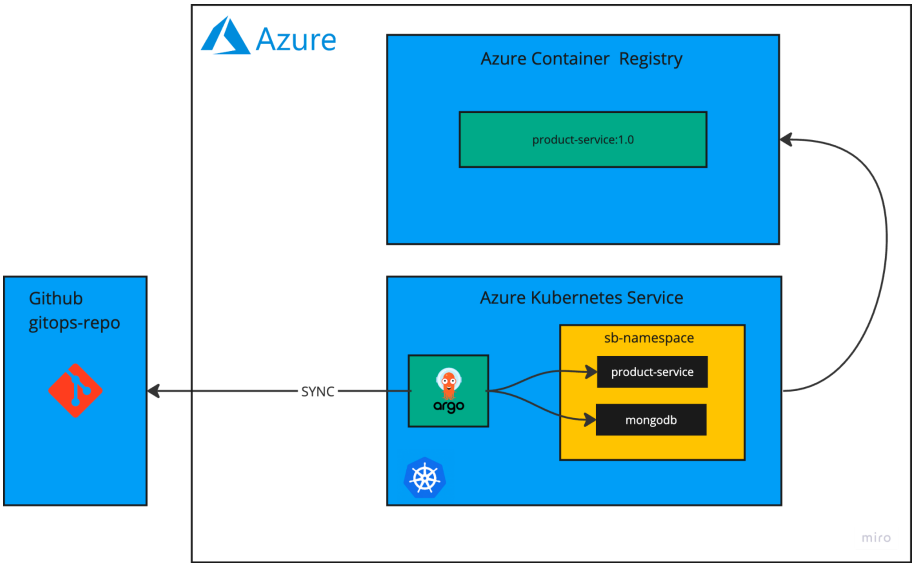
Workshop GitOps

Het doel is om een simpele spring boot applicatie (product-service) te deployen op een Azure Kubernetes cluster via GitOps ArgoCD.



1. Clone GitHub repository

1. Download code van GitHub. Hierin zitten twee projecten. Een simpele spring boot product-service en een gitops map. Idealiter zou je hiervoor twee aparte repositories willen hebben, maar om het voor deze workshop simpel te houden is het in 1 repository gestopt.

   ```
   git clone https://github.com/JeffreyOomen/immuneit-workshop-gitops.git
   ```

2. Login Azure Portal

1. Login op **portal.azure.com** met de e-mail waarop je de uitnodiging hebt gekregen.
2. Maak een resource group aan met jouw eigen naam en achternaam, e.g.: **jeffreyoomen** en region: **(Europe) West Europe**.

3. Maak Azure Resources aan in Azure Portal

1. Maak een Azure Kubernetes Service aan in jouw resource group. Gebruik de volgende gegevens (wat hier niet staat op default laten staan):
   - Resourcegroup: **jeffreyoomen**
   - Cluster preset configuration: Dev/Test
   - Kubernetes cluster naam: **jeffreyoomen**-k8s-cluster
   - Region: (Europe) West Europe
   - Availability Zones: None
2. Login via Azure CLI. Het eerste command redirect naar een browser pagina waar je kunt inloggen.

```
az login

TOKEN=$(az acr login --name immuneitdockerregistry --expose-token
--output tsv --query accessToken)
```

4. Toegang tot het K8s-cluster

1. Zorg ervoor dat jouw kubernetes cluster toegang heeft tot de docker registry met het volgende command (https://argo-cd.readthedocs.io/en/stable/https://learn.microsoft.com/en-us/azure/aks/cluster-container-registry-integration?tabs=azure-cli)

   ```
   az aks update -n jeffreyoomen-k8s-cluster -g jeffreyoomen --attach-acr
   immuneitdockerregistry
   ```

2. Maak verbinding vanaf jouw computer met het kubernetes cluster in Azure. https://learn.microsoft.com/en-us/azure/aks/learn/quick-kubernetes-deploy-cli

   ```
   az aks install-cli

   az aks get-credentials --resource-group jeffreyoomen --name
   jeffreyoomen-k8s-cluster

   kubectl get nodes
   ```

5. Gitops met ArgoCD

1. Installeer ArgoCD in jouw kubernetes cluster. Hoe dit moet, zie: https://argo-cd.readthedocs.io/en/stable/

   ```
   kubectl create namespace argocd

   kubectl apply -n argocd -f
   https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.y
   aml
   ```

2. Zoek de argocd service die je zojuist hebt aangemaakt met:

   ```
   kubectl get svc -n argocd
   ```

3. Achterhaal alvast het gegenereerde wachtwoord om zometeen in te kunnen loggen in het dashboard van ArgoCD:

   ```
   kubectl get secret argocd-initial-admin-secret -n argocd -o yaml

   echo djBhajBHendTN0UwbHF4TQ== | base64 --decode
   ```

   Het wachtwoord is de output van het decoderen, minus het %-teken.

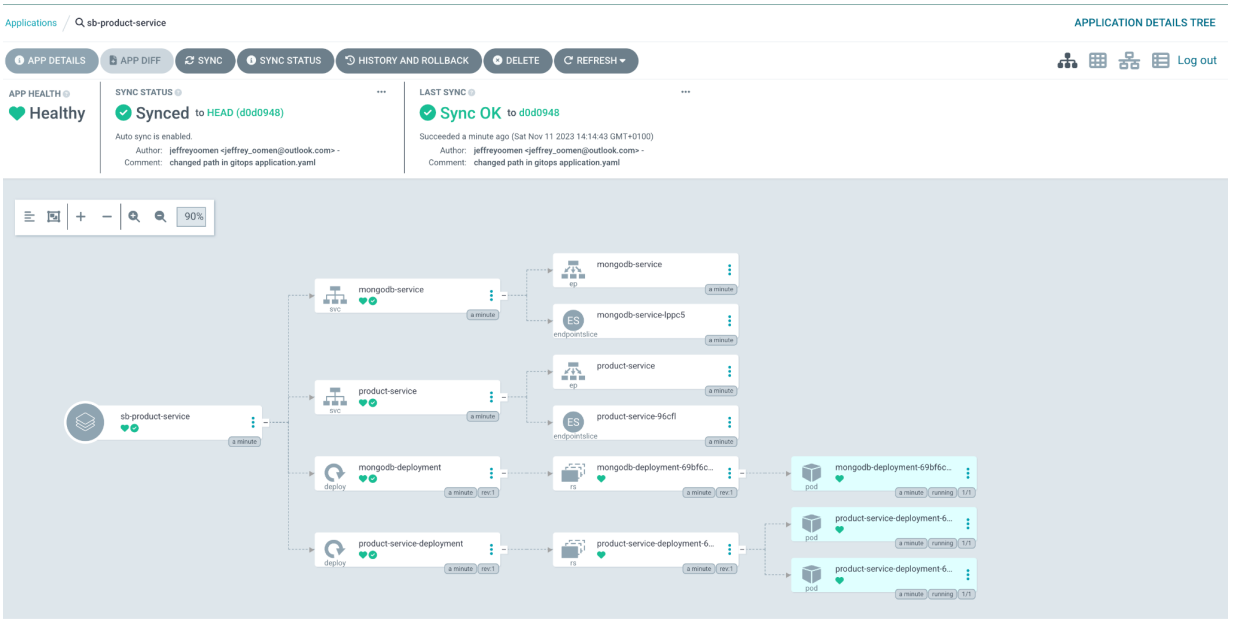
4. Gebruik port-forward zodat je vanaf jouw machine kunt inloggen in ArgoCD:

```
kubectl port-forward -n argocd svc/argocd-server 8080:443
```

5. Navigeer naar `localhost:8080`
6. Login ArgoCD met:
   - **admin**
   - **Het gedecodeerde wachtwoord (zonder %-teken)**
7. Eenmalig toepassen van de application.yaml die alle k8s objecten aanmaakt (in de product-service-gitops map):

   ```
   kubectl apply -f application.yaml
   ```

8. Kijk op `localhost:8080`, er is nu een applicatie automatisch toegevoegd en als je daar op klikt zie je dat er allerlei kubernetes resources zijn aangemaakt.



9. Gebruik port-forward om de product-service die zojuist gedeployed is op een Azure Kubernetes cluster aan te roepen.

   ```
   kubectl get svc -n sb-namespace

   kubectl port-forward product-service 8081:8080

   kubectl port-forward svc/product-service -n sb-namespace 8081:8080
   ```

10. Navigeer naar: http://localhost:8081/api/product. Je kunt een product toevoegen met een POST-request op http://localhost:8081/api/product met bijvoorbeeld de volgende JSON body: `{ "name": "Iphone15", "description": "Iphone15", "price": 1200 }`

Continuous Synchronization in action

Een van de eigenschappen dat GitOps uniek maakt is dat het automatisch synchroniseert met de toestand zoals beschreven in de GitOps repository. Om dit in actie te zien kun je handmatig aanpassingen doen in het cluster.

Voorbeeld:
Onderstaande command toont het aantal pods binnen de namespace sb-namespace. In de Git repository staat deze op 2 pods, dus je zou twee product-service pods moeten zien.

```
kubectl get pods -n sb-namespace
```

Als je nu handmatig het aantal pods op 3 replicas zet, zal je zien dat GitOps meteen in actie komt en het razendsnel weer synchroniseert naar 2 pods zoals in de Git repository staat.

```
kubectl scale --replicas=3 deployment product-service-deployment -n
sb-namespace
```

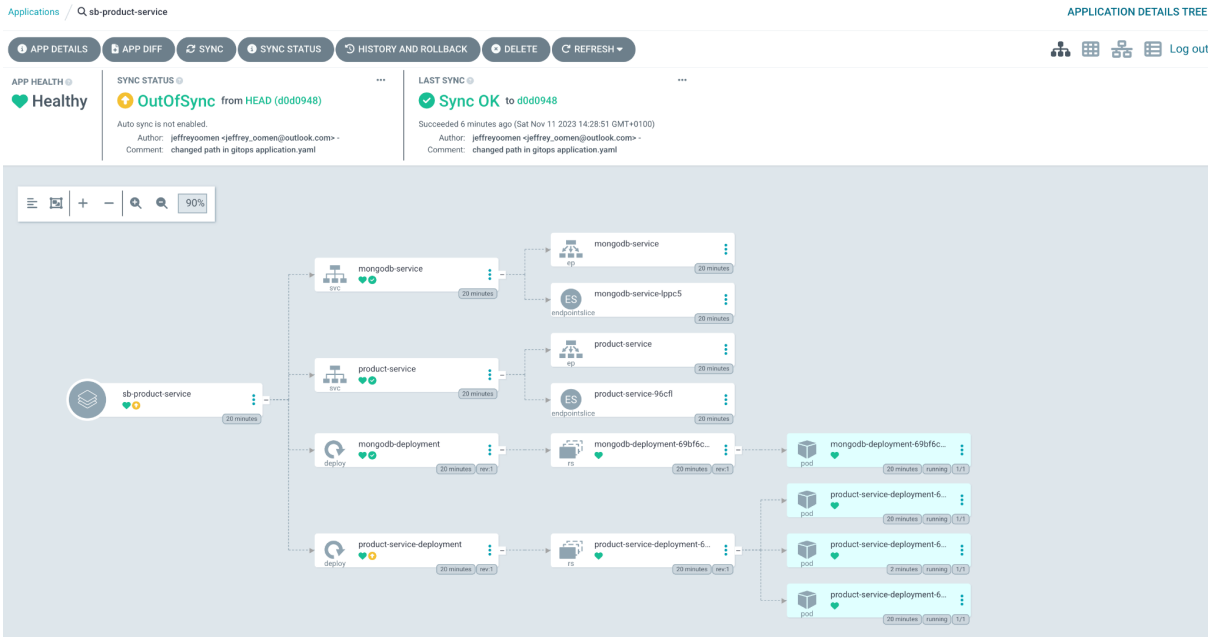
Voer snel onderstaand command uit of kijk snel in het ArgoCD dashboard. Je zal dan heel kort even drie pods zien en dan gaat er 1 weg.

```
kubectl get pods -n sb-namespace
```

Als het te snel gaat kun je ook in het ArgoCD dashboard de synchronisatie even uitzetten, dit kan via: App Details > Disable Auto Sync. Als je dan weer het aantal replicas opschaalt naar drie:

```
kubectl scale --replicas=3 deployment product-service-deployment -n
sb-namespace
```

Zal je dit ook zien in het ArgoCD dashboard. Het dashboard zal ook tonen dat het Out of Sync is (vergeet niet ook Self Heal en Prune aan te zetten). Als je vervolgens Auto Sync weer aanzet zul je zien dat het er weer twee worden.



Troubleshooting

Error (MAC): az aks install-cli

```
1. az aks install-clicurl -LO
   "https://dl.k8s.io/release/v1.27.1/bin/darwin/arm64/kubectl"
2. chmod +x ./kubectl
3. sudo mv ./kubectl /usr/local/bin/kubectl
4. sudo chown root: /usr/local/bin/kubectl
5. kubectl version --client --output=yaml
6. sudo az aks install-cli
```