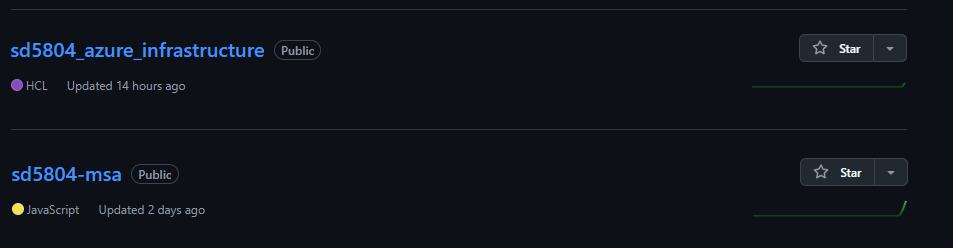
ASSIGNMENT 3: Azure INFRASTRUCTURE

1. Create 2 repositories on github

[ngocdung12112000/sd5804-msa](https://github.com/ngocdung12112000/sd5804-msa/tree/main) - Application code

[ngocdung12112000/sd5804\_azure\_infrastructure](https://github.com/ngocdung12112000/sd5804_azure_infrastructure)- Infrastructure code

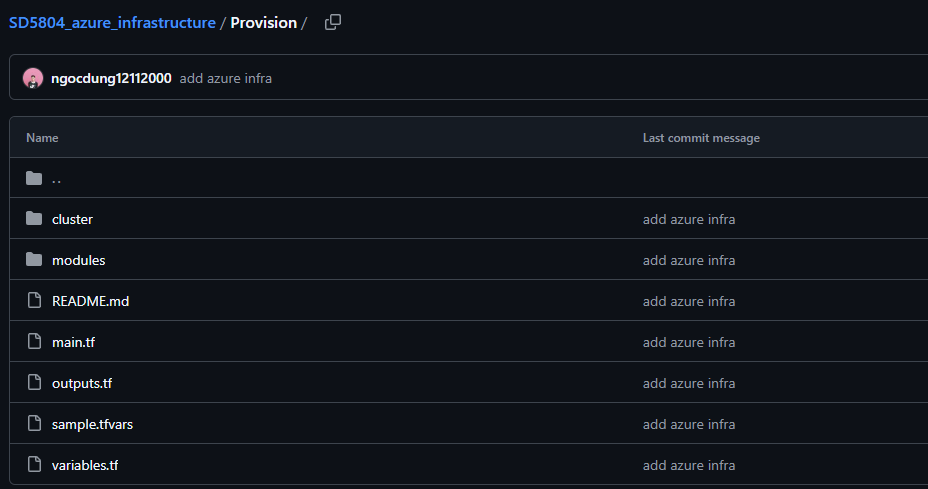


1. Upload code to repo sd5804-msa



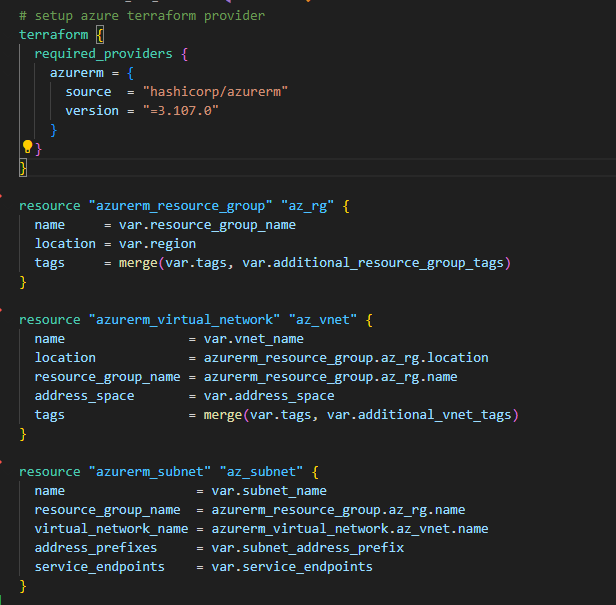
1. In repo sd5804\_azure\_infrastructure

Add folder Provision



Inside folder Provision we have Terraform code to provision VNet, ACR, AKS, Virtual Machine

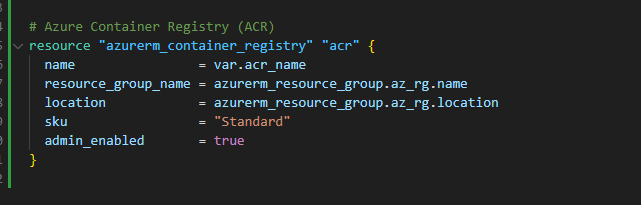
VNet



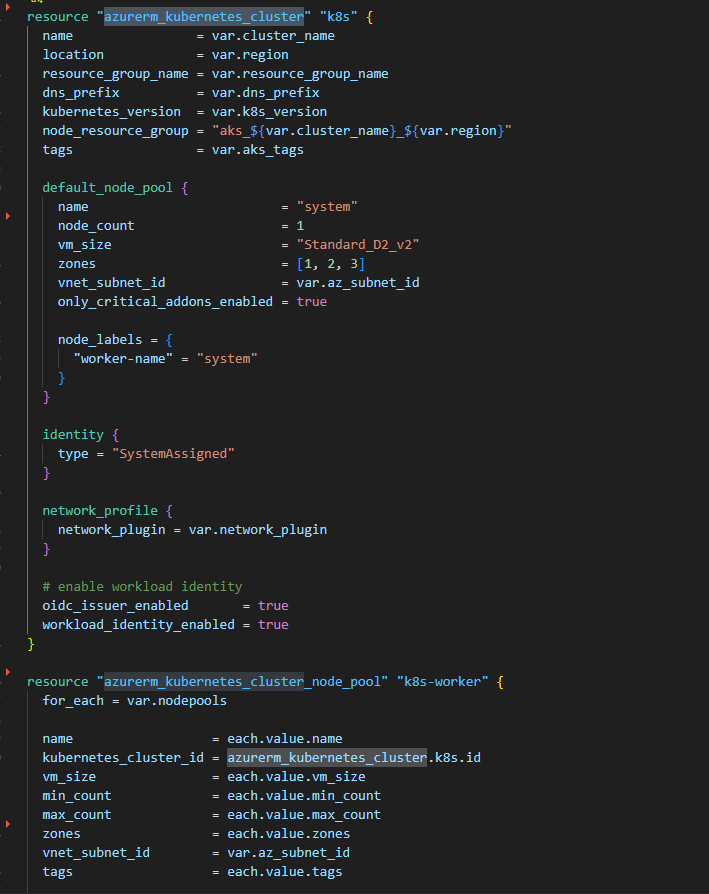
Azure Virtual machine



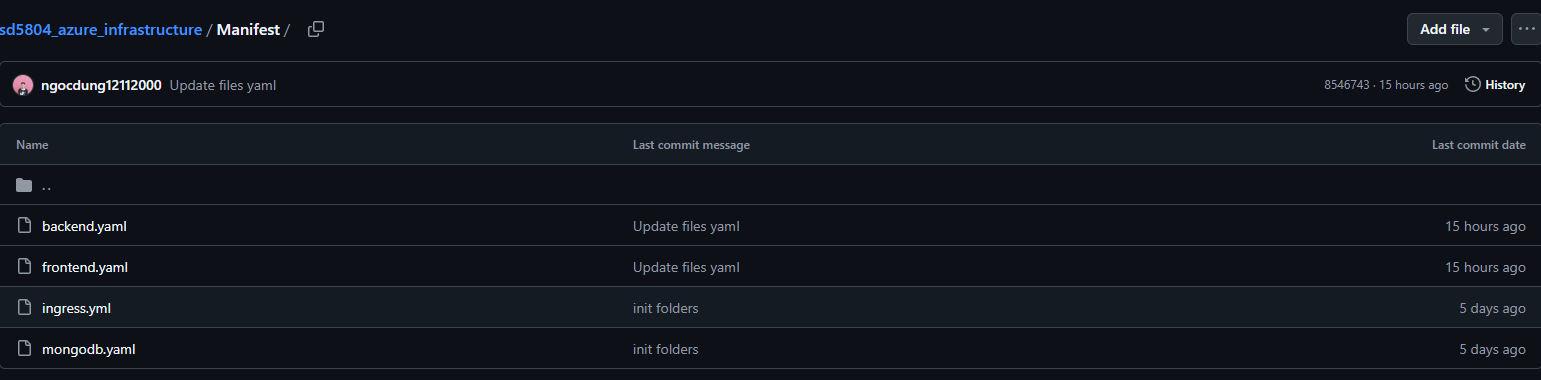
ACR



EKS



Folder Manifest



1. Run Terraform command to create resource in Azure

Initialize:

terraform init

Format and Validate:

terraform fmt

terraform validate

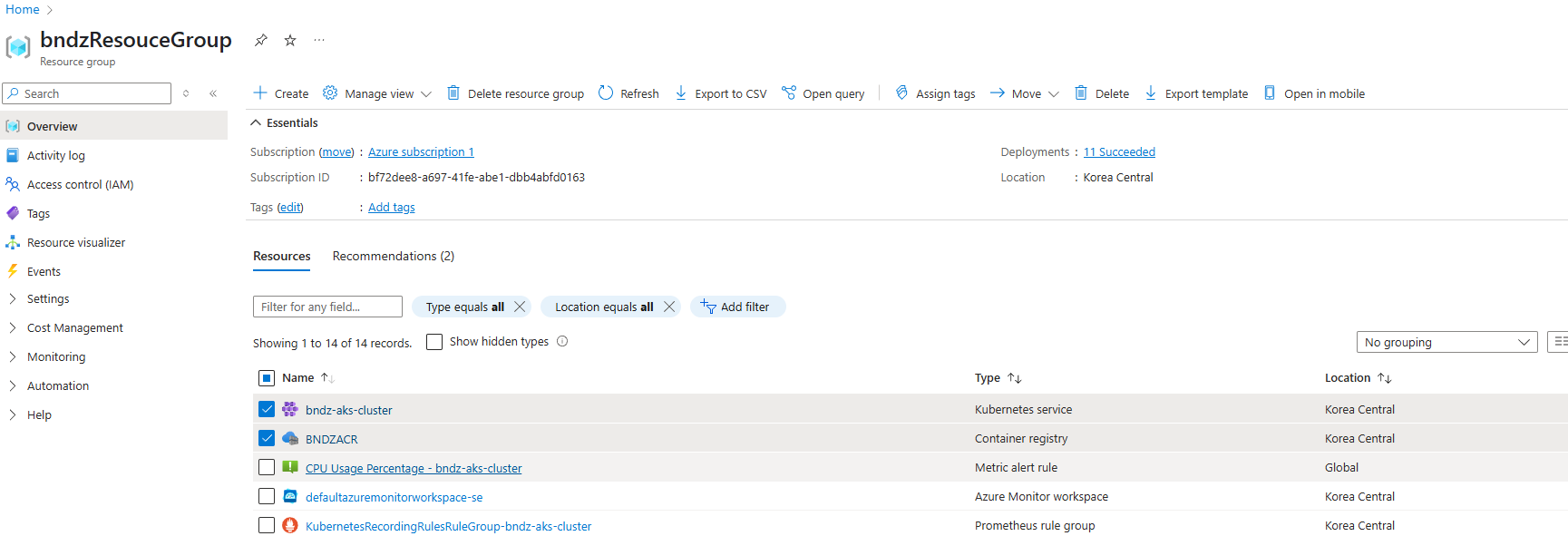
Plan:

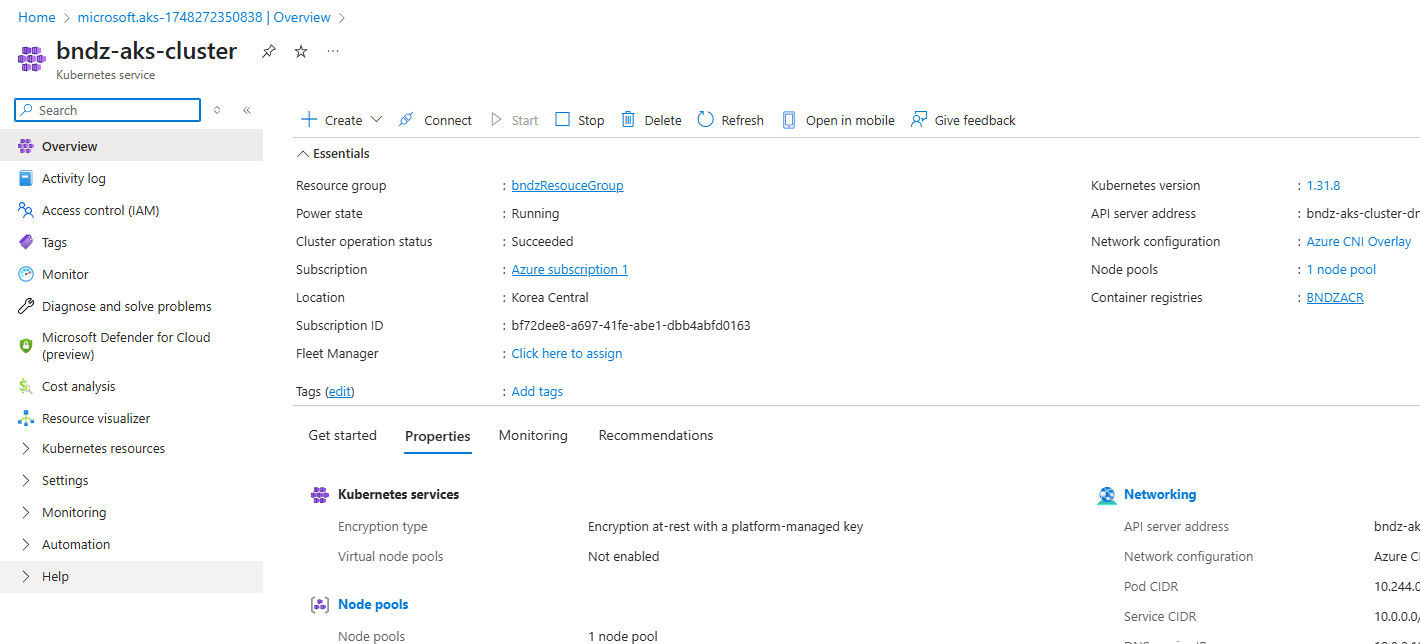
terraform plan -var-file="sample.tfvars"

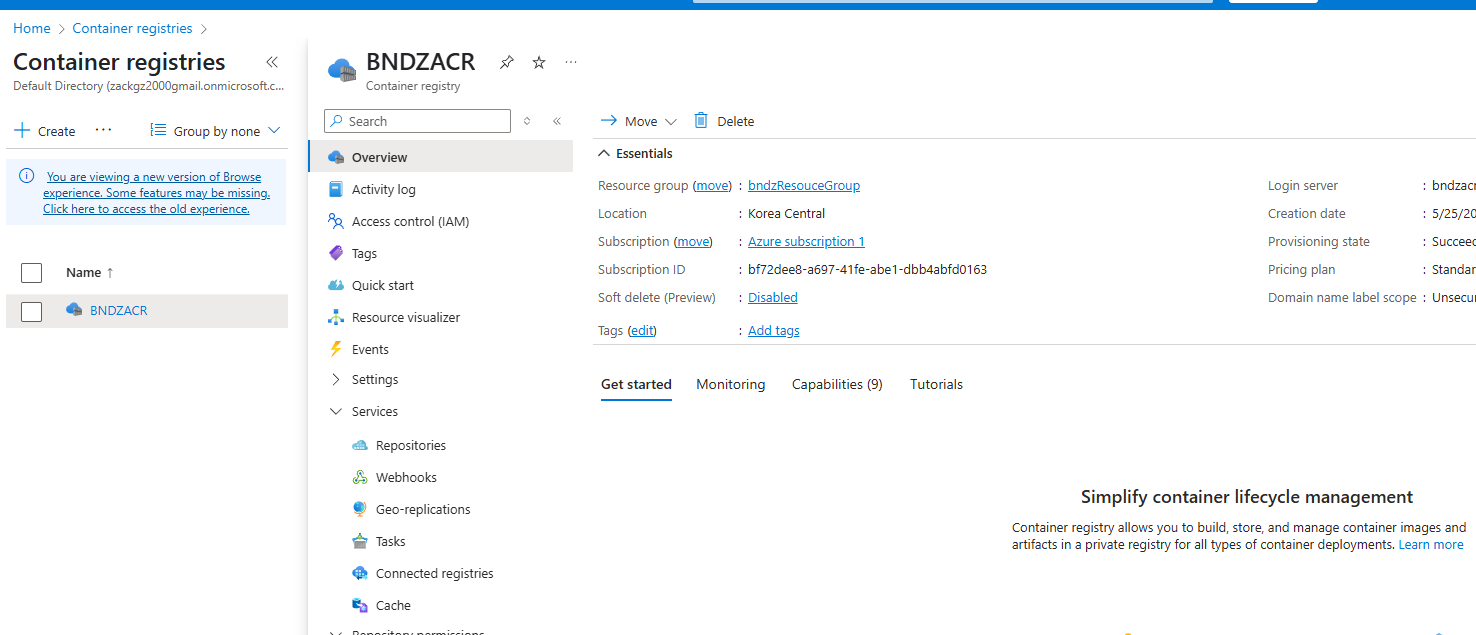
Apply

terraform apply -var-file="sample.tfvars"

After finish we will have these resources



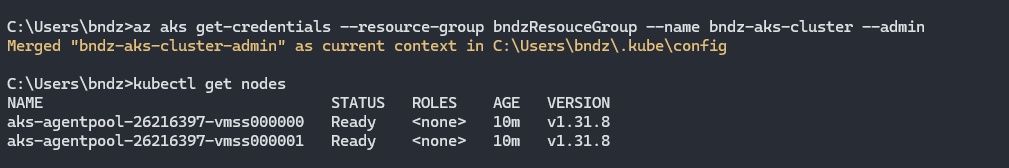




* Run command to connect to AKS

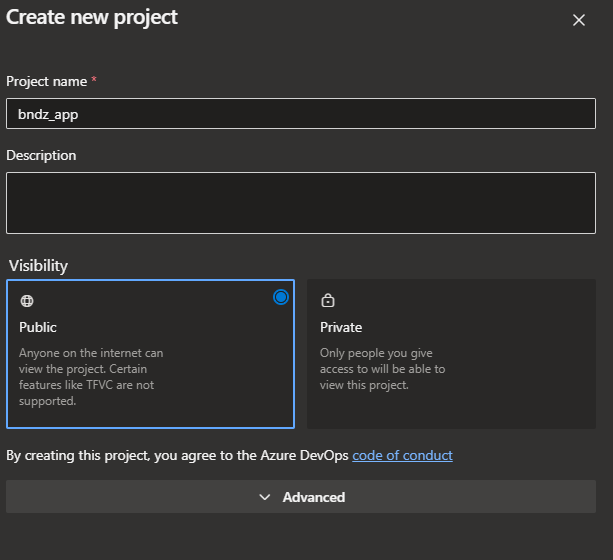
az aks get-credentials --resource-group bndzResouceGroup --name bndz-aks-cluster --admin

kubectl get nodes

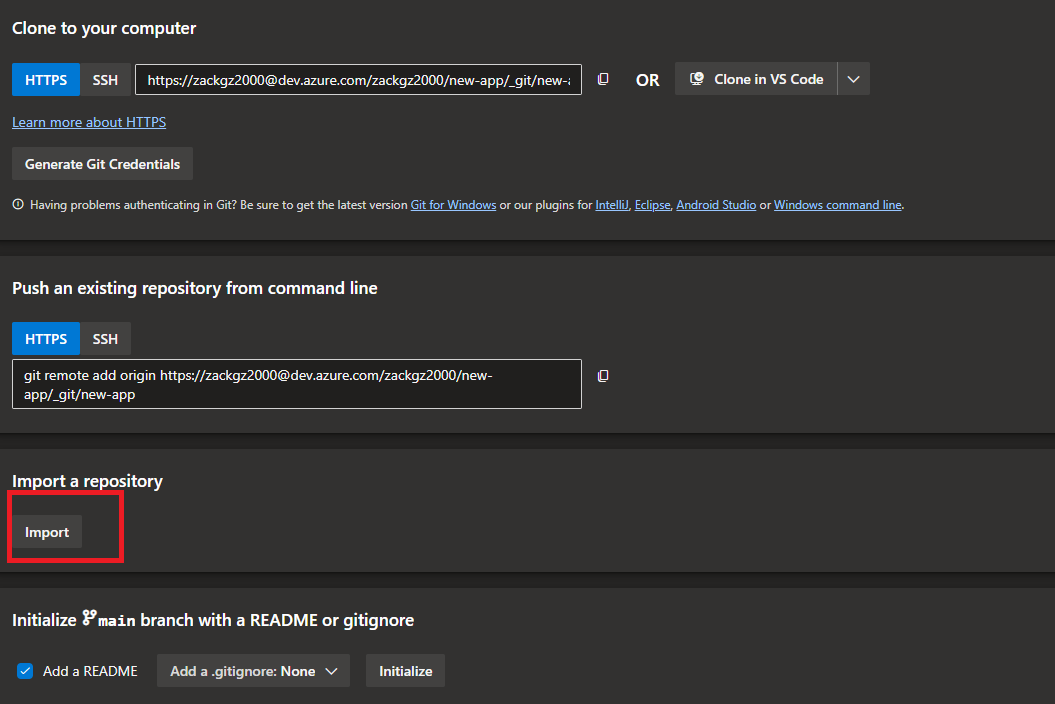


6. Create CI pipeline on Azure Devops

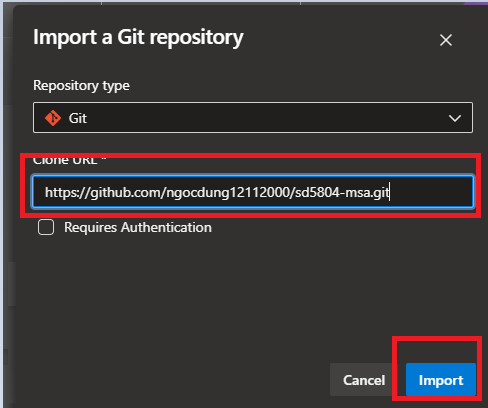
* Go to website <https://dev.azure.com/>
* Click New Project

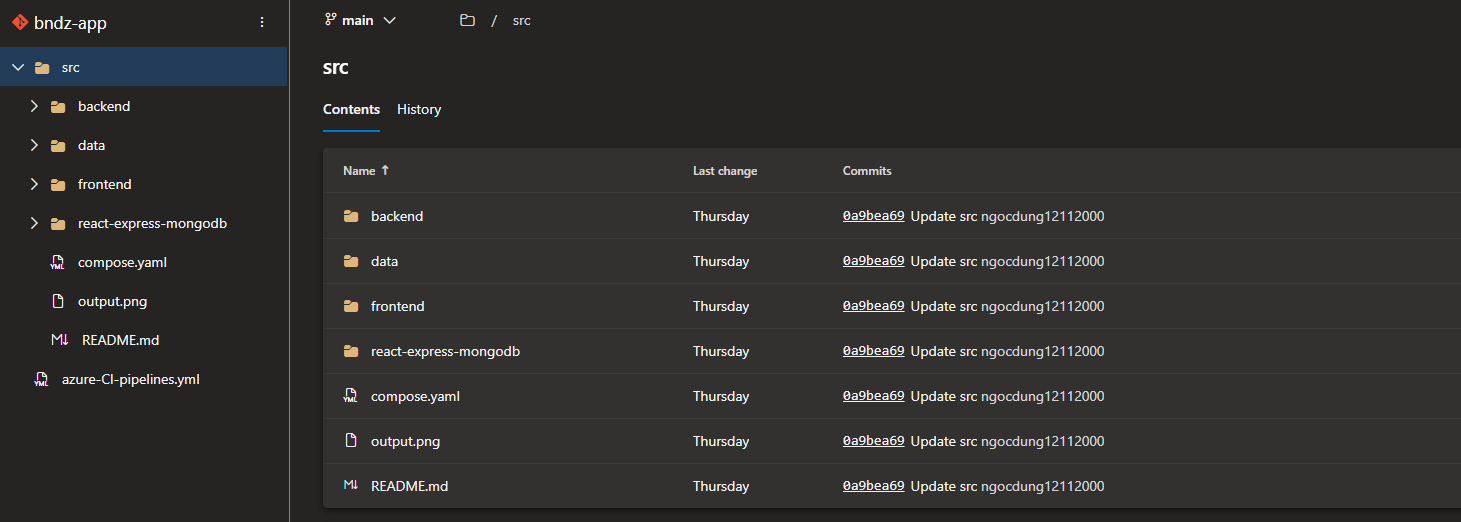


* Add project name -> click Create
* Click Repo tab in left section -> Import a repository

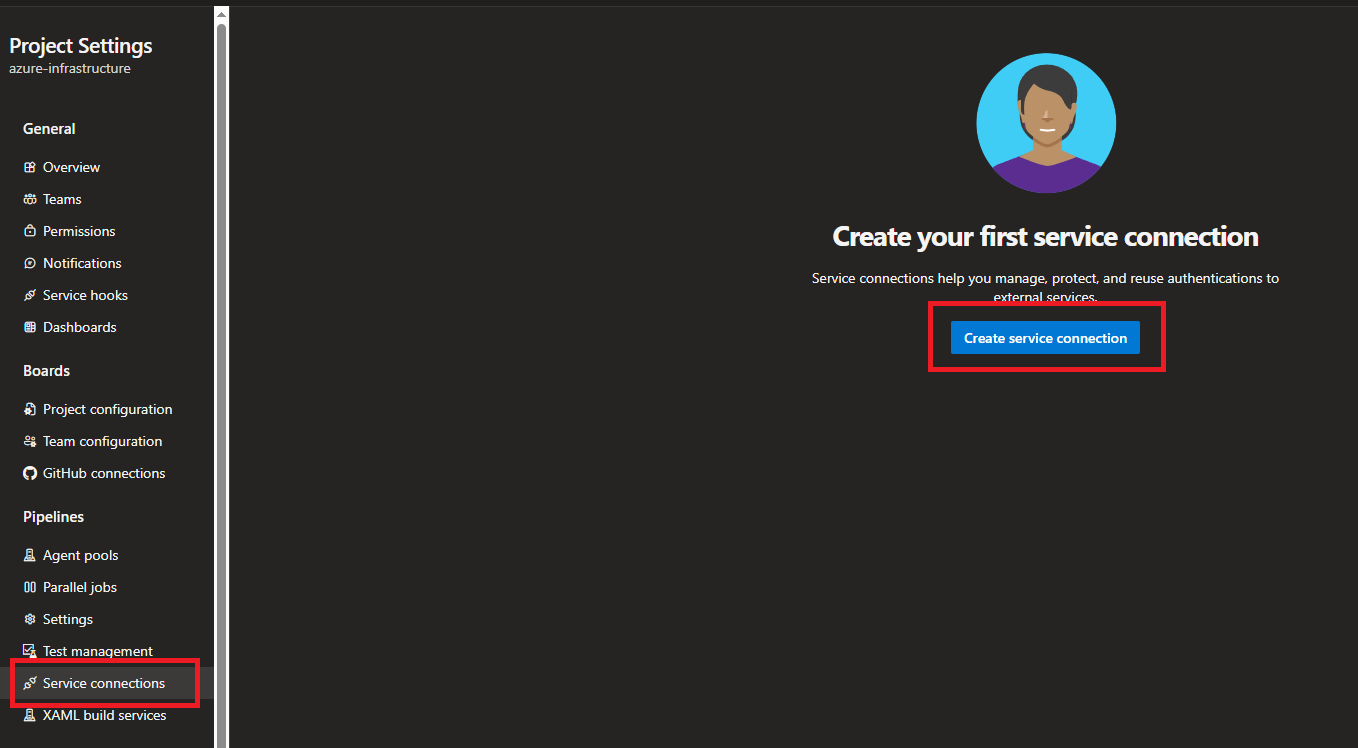


* Fill git url: <https://github.com/ngocdung12112000/sd5804-msa.git> -> Import

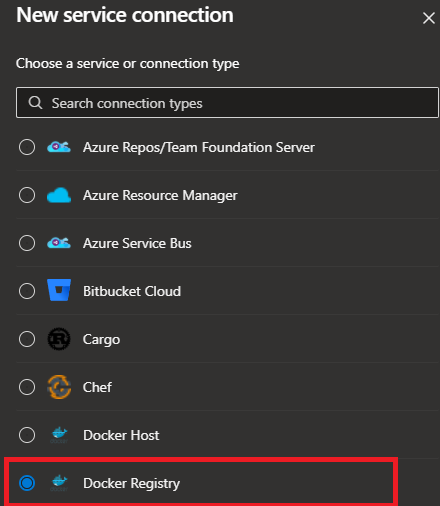




* Go to Project Settings -> Service Connection -> New service connection

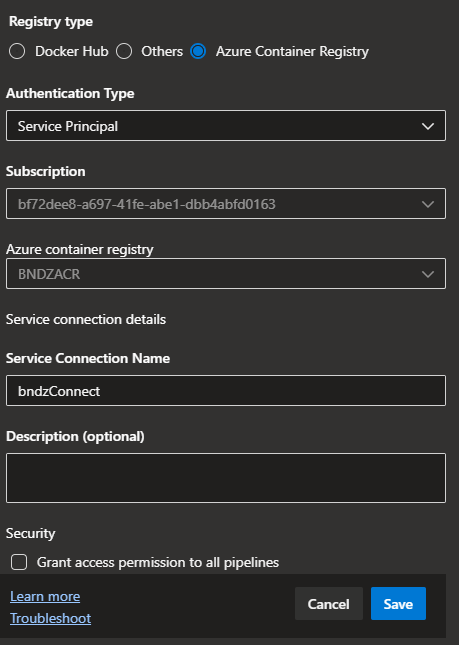


* Select Docker Registry

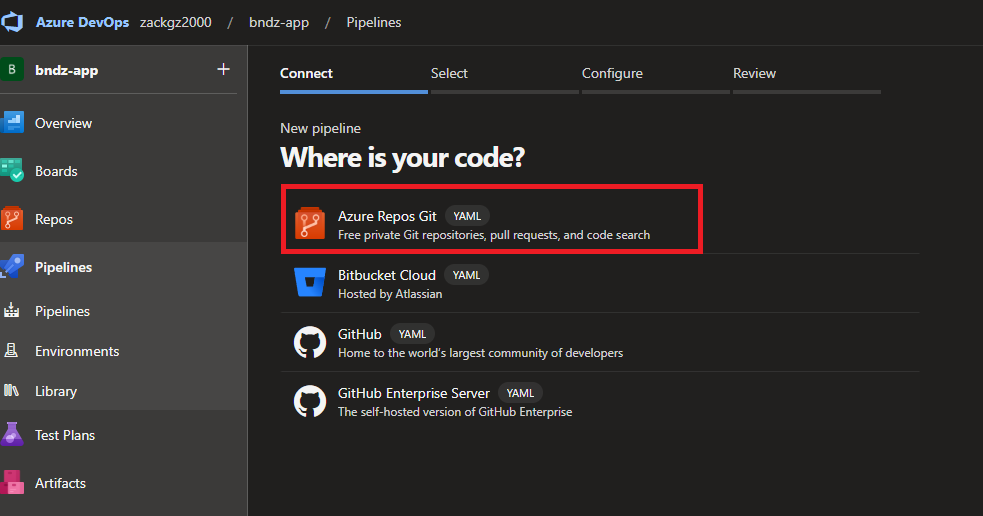


* Check Azure Container Registry
* Choose Azure Container Registry (BNDZACR)
* Add Service Connection Name
* Click Save

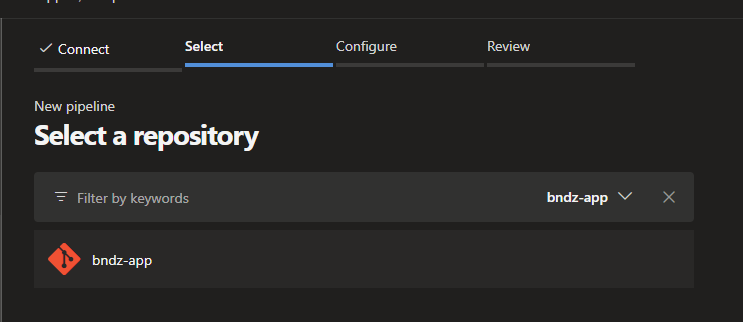
This is connection will be used for CI pipeline



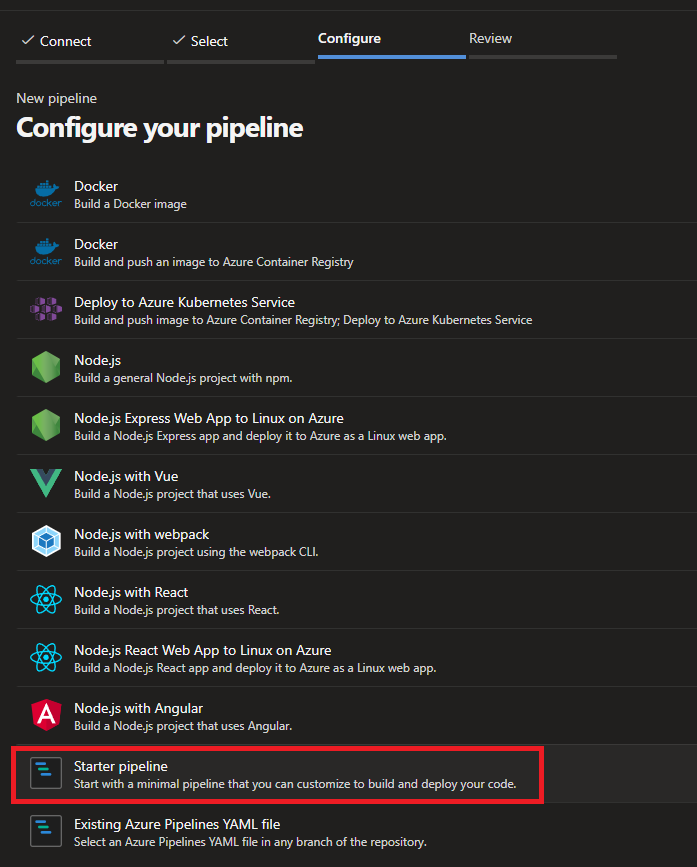
* In left section -> Click Pipeline -> New Pipeline
* Choose Azure Repos git



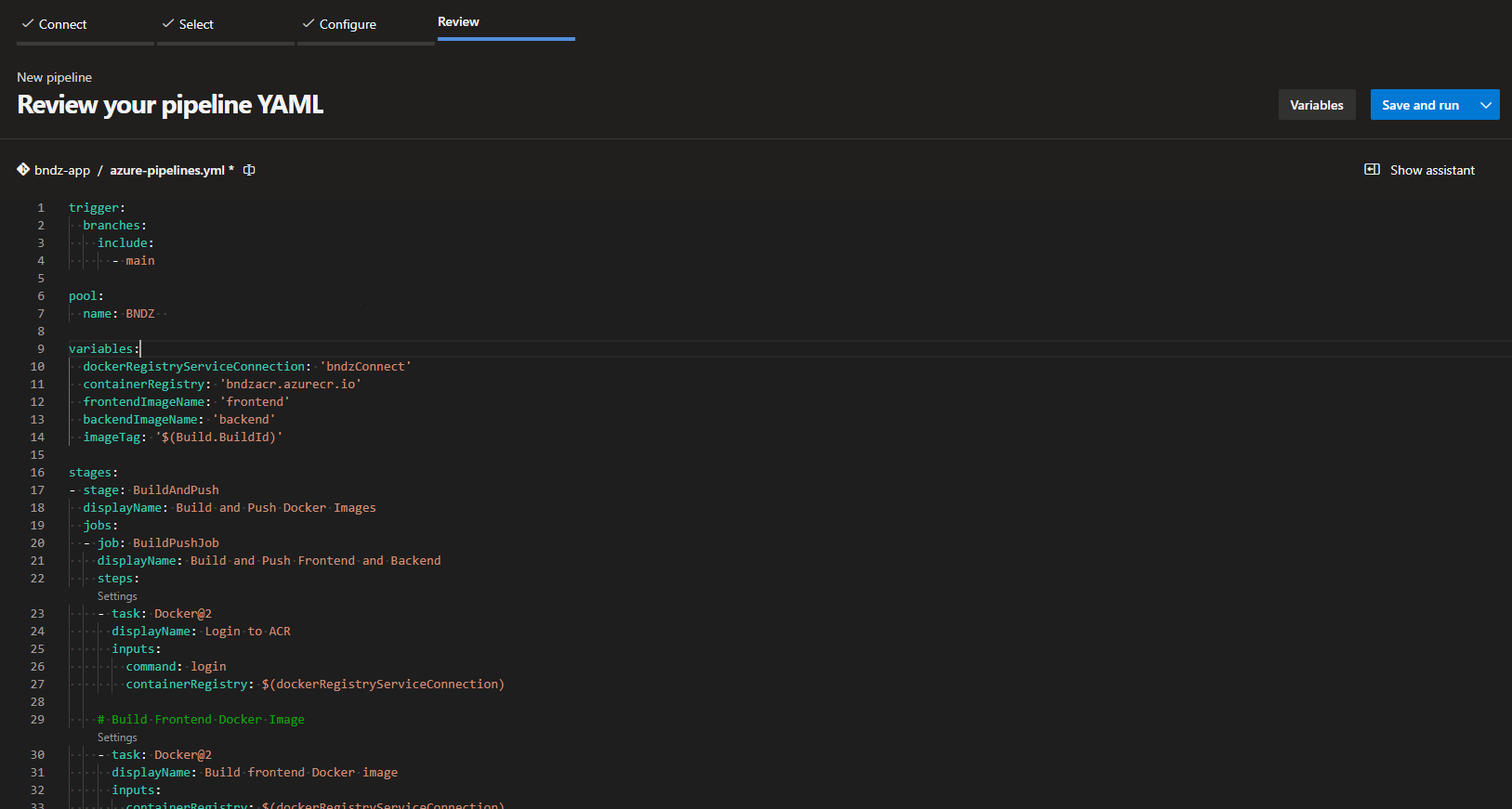
* Select repository

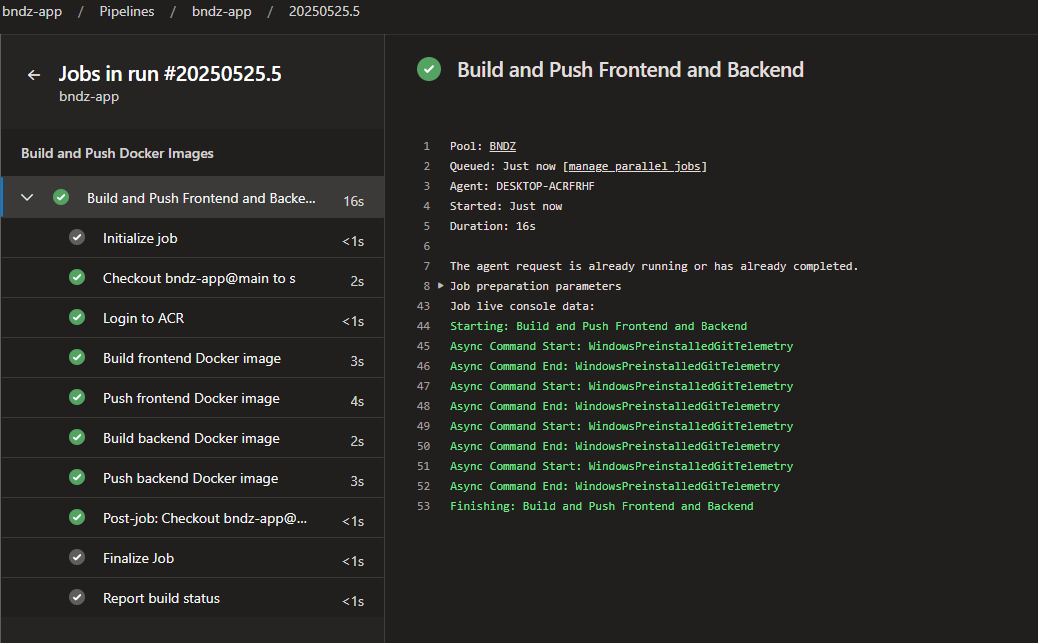


* Select Starter Pipeline



* Add pipeline script and name as azure-CI-pipeline.yml
* Click Save and Run

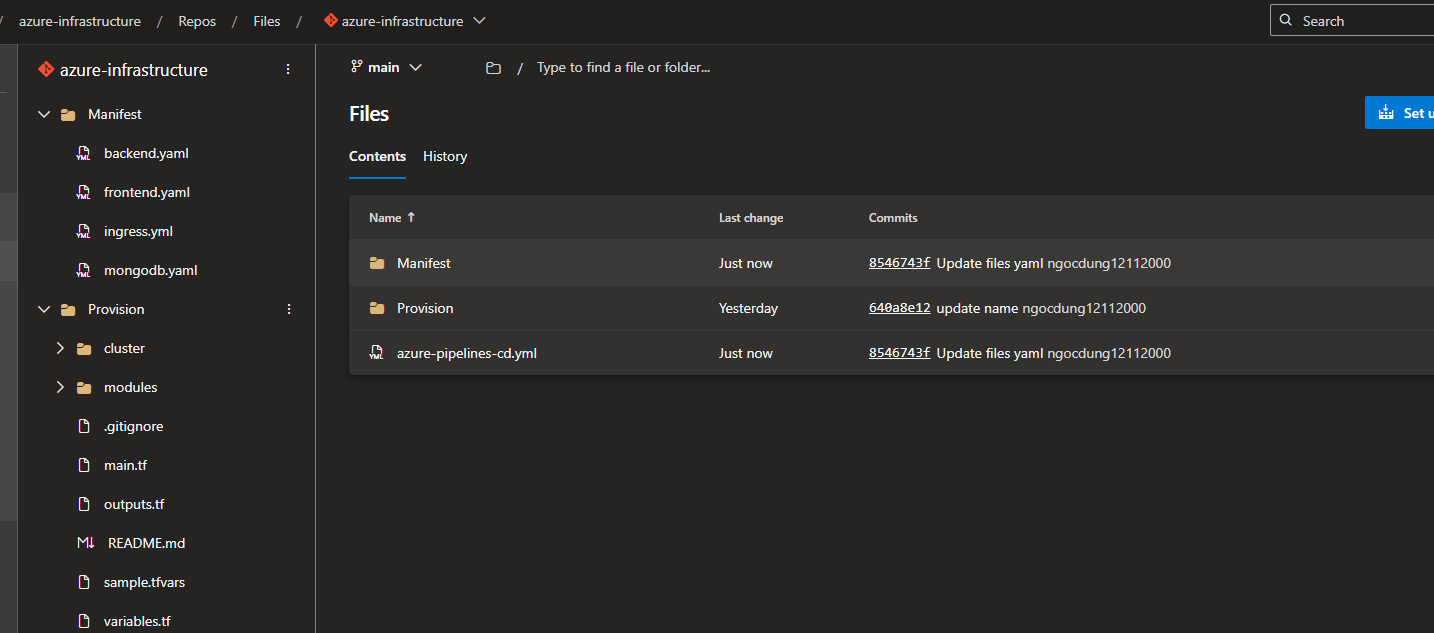




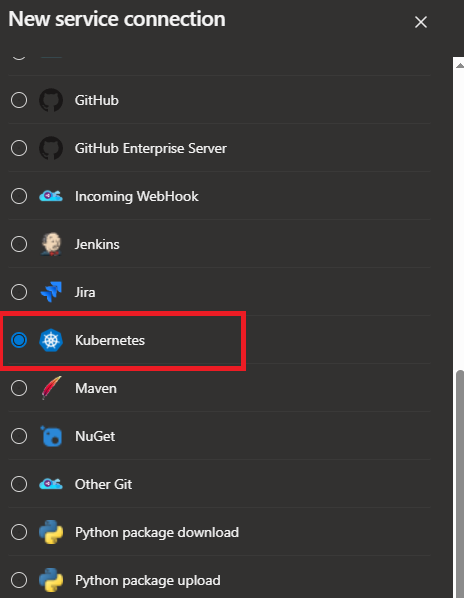
* After pipeline run success
* Go to Container Registry (BNDZACR)
* Click Repositories -> will see 2 images backend and frontend

7. Create CD pipeline on Azure Devops

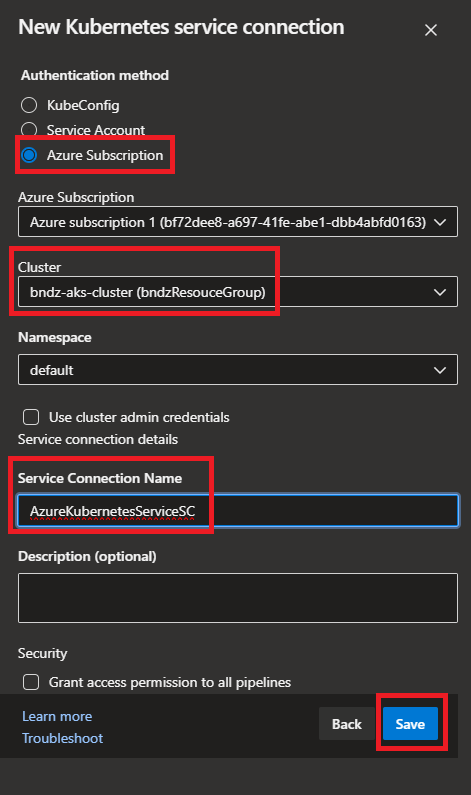
* Create a project named azure infrastructure
* Import repo https://github.com/ngocdung12112000/sd5804\_azure\_infrastructure.git



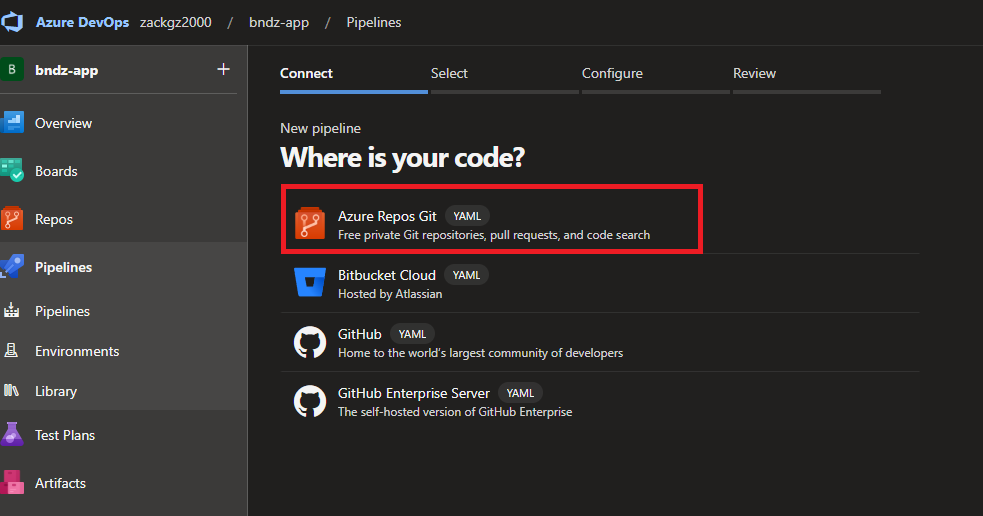
* Go to Project Settings -> Service Connection -> New service connection
* Select Kubernetes



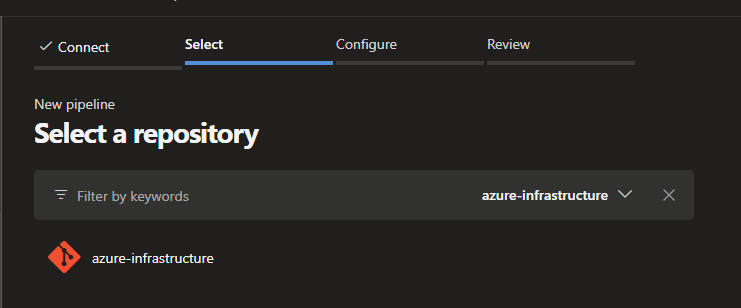
* Choose Azure Subscription
* Select Cluster -> Add Service Connection Name -> Save



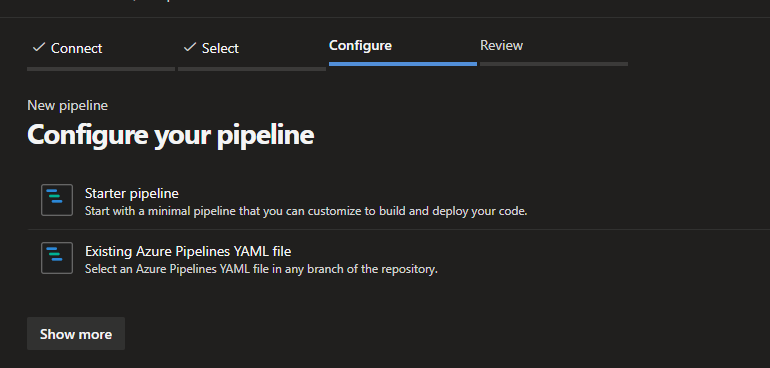
* In left section -> Click Pipeline -> New Pipeline
* Choose Azure Repos git



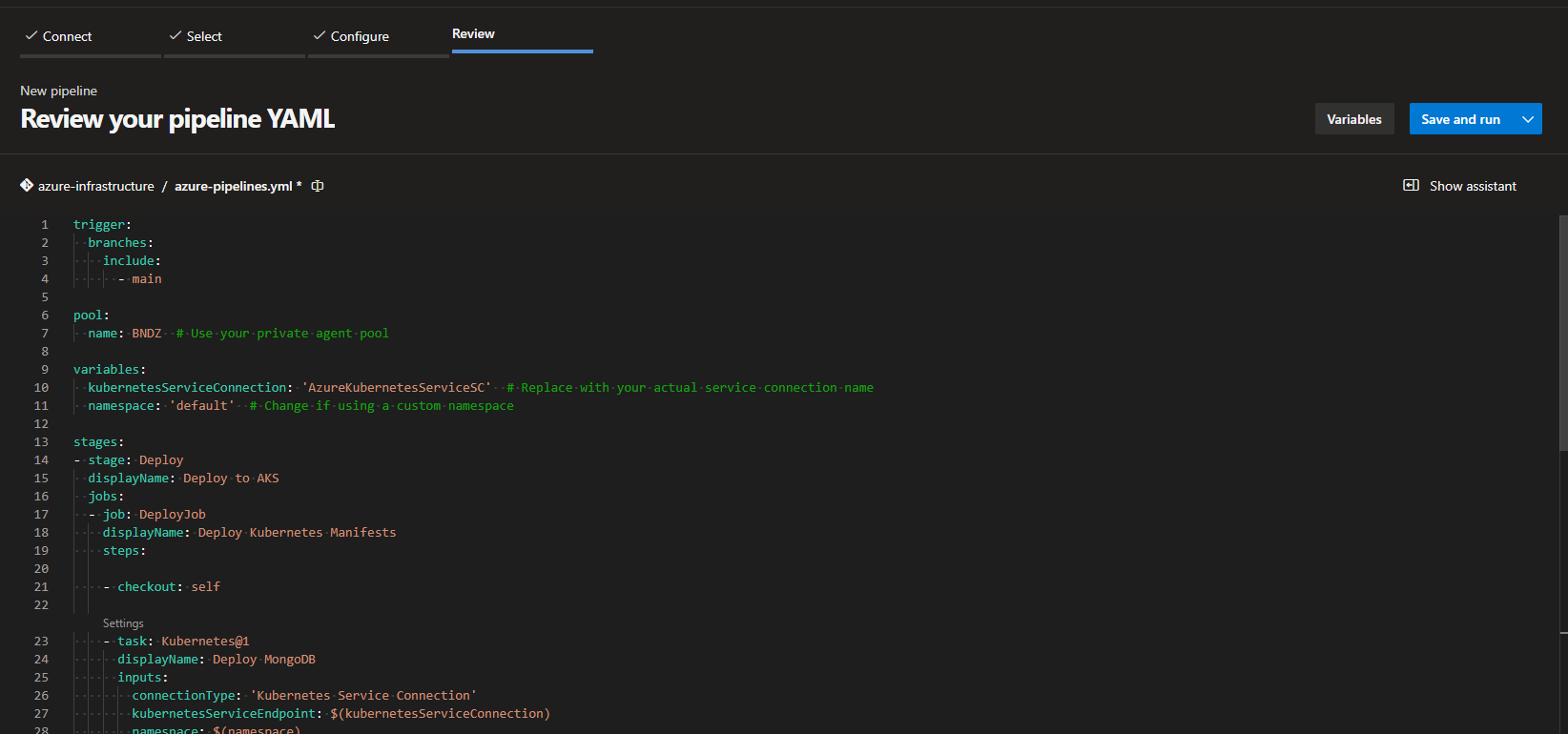
* Select repository



* Select Starter Pipeline

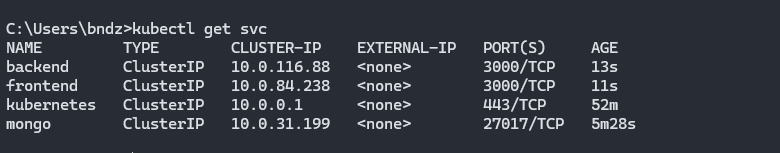


* Add pipeline script and name as azure-CD-pipeline.yml
* Click Save and Run

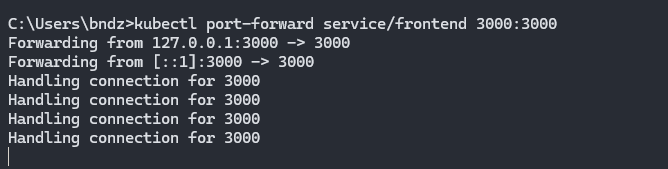


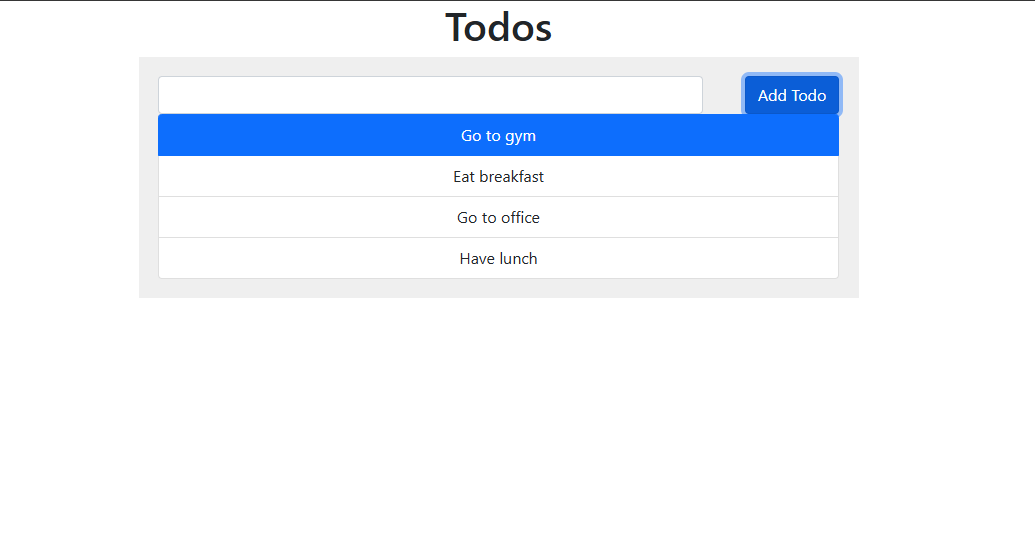
* After Pipeline run success
* Run command

kubectl get nodes



* Run kubectl port-forward service/frontend 3000:3000





8. Set up Prometheus and Grafana

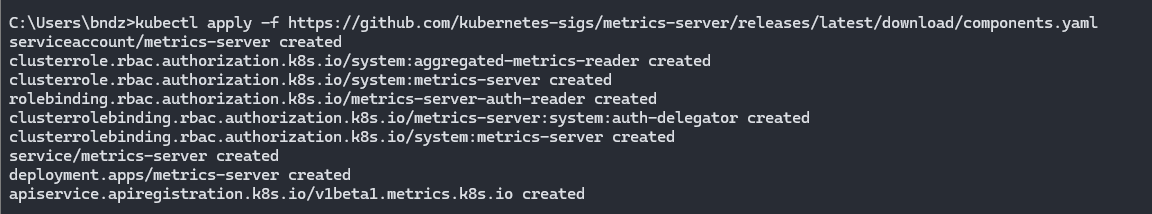
Run command to connect AKS cluster

az aks get-credentials --resource-group bndzResouceGroup --name bndz-aks-cluster --admin

* Set up Metrics Server

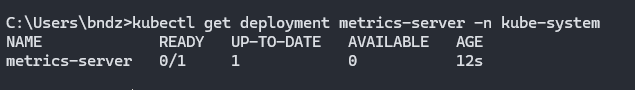
Deploy the Metrics Server with the following command:

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml



Verify that the metric-server

kubectl get deployment metrics-server -n kube-system



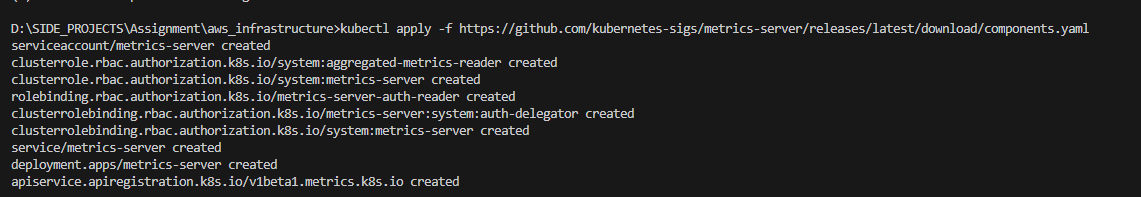
- Deploy Prometheus using Helm

Create a Prometheus namespace.

kubectl create namespace prometheus

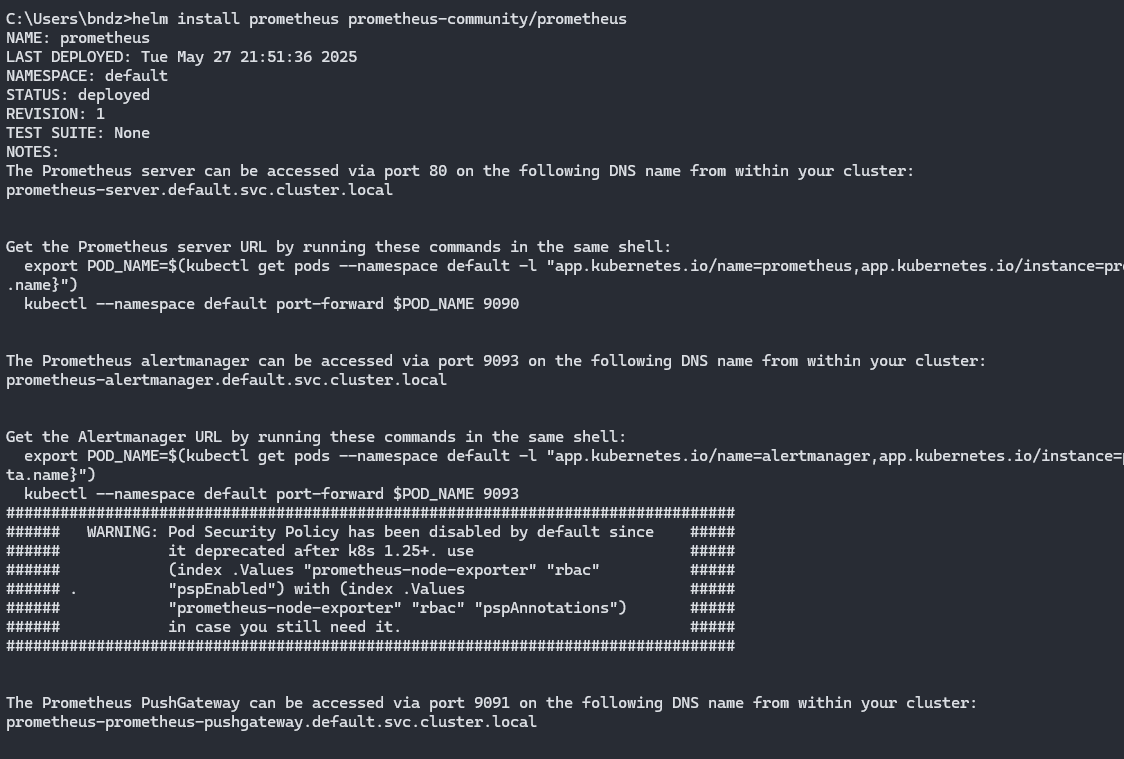
Add the prometheus-community chart repository.

helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>

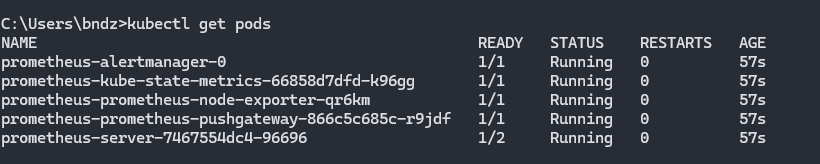


Install prometheus

helm install prometheus prometheus-community/prometheus

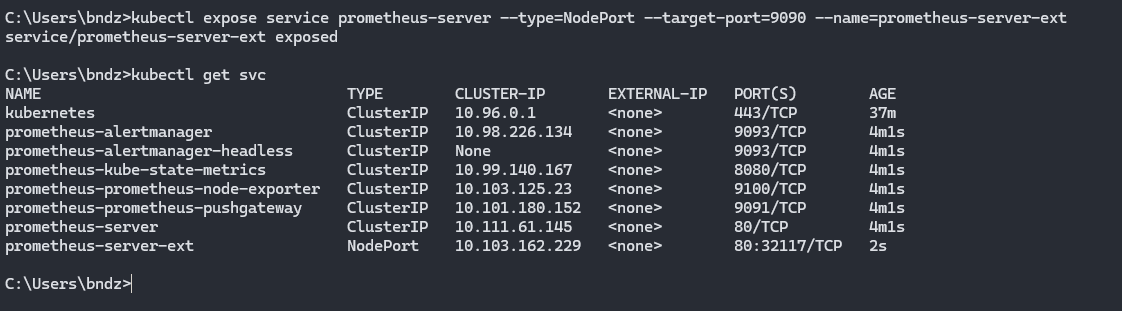


kubectl get pods

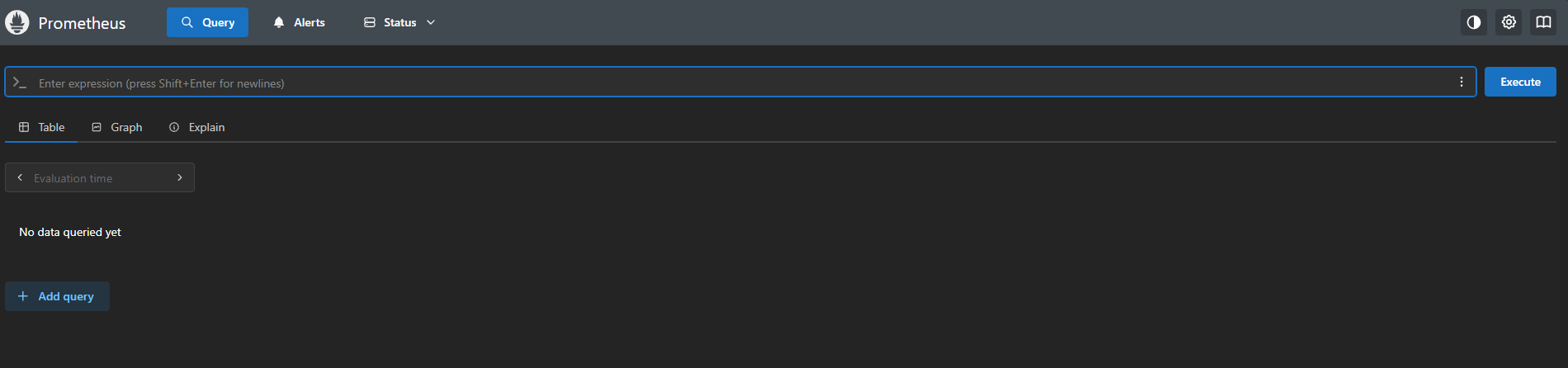


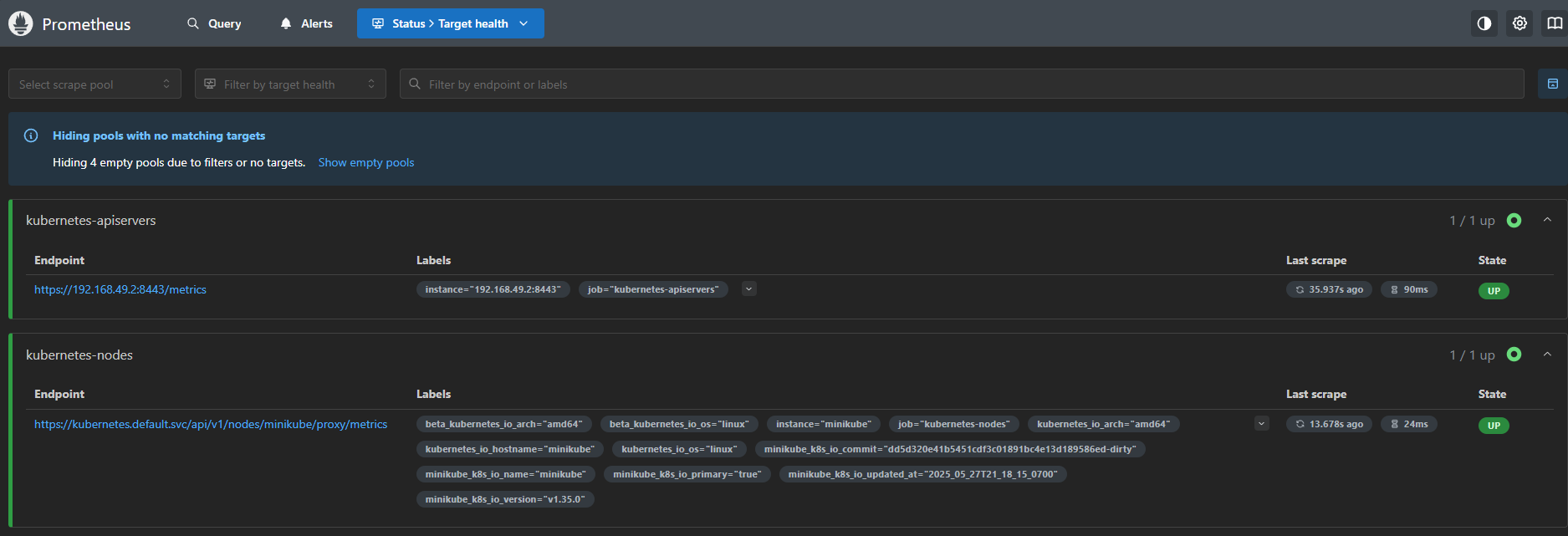
Expose service prometheus

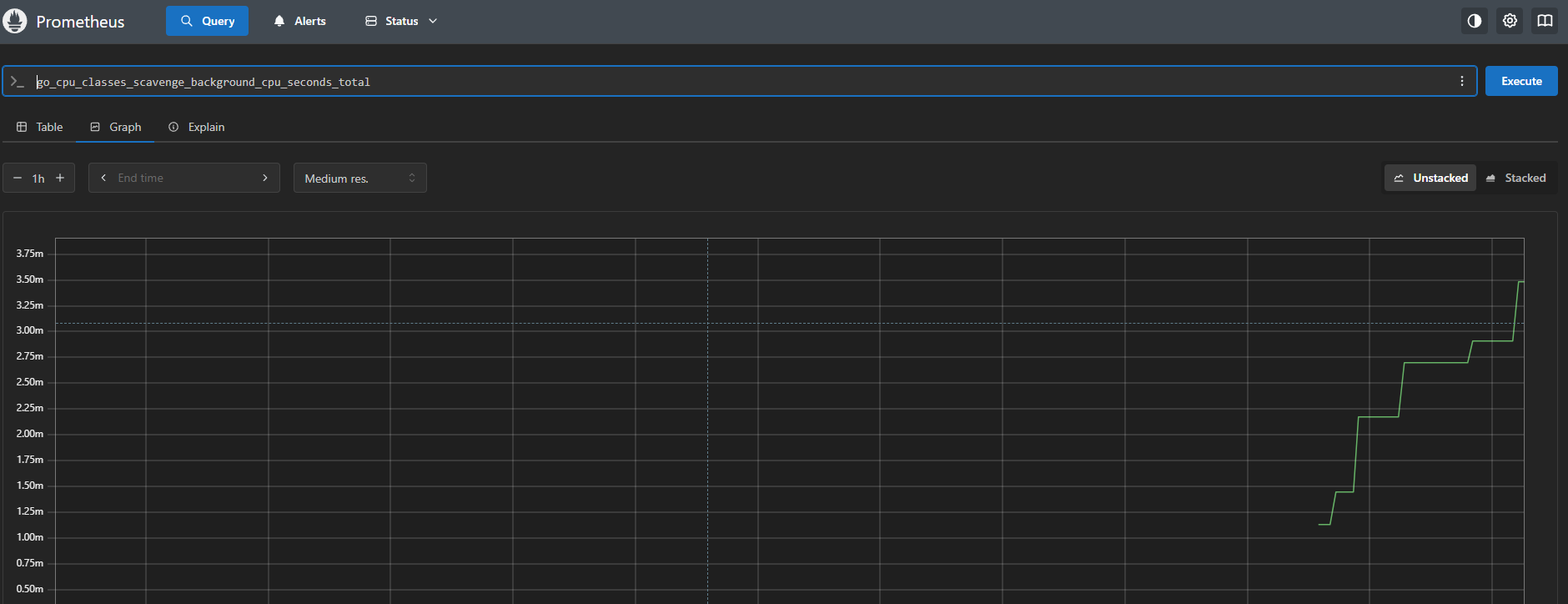
kubectl expose service prometheus --type=NodePort --target-port=9090 --name=prometheus-server-ext



Open in browser





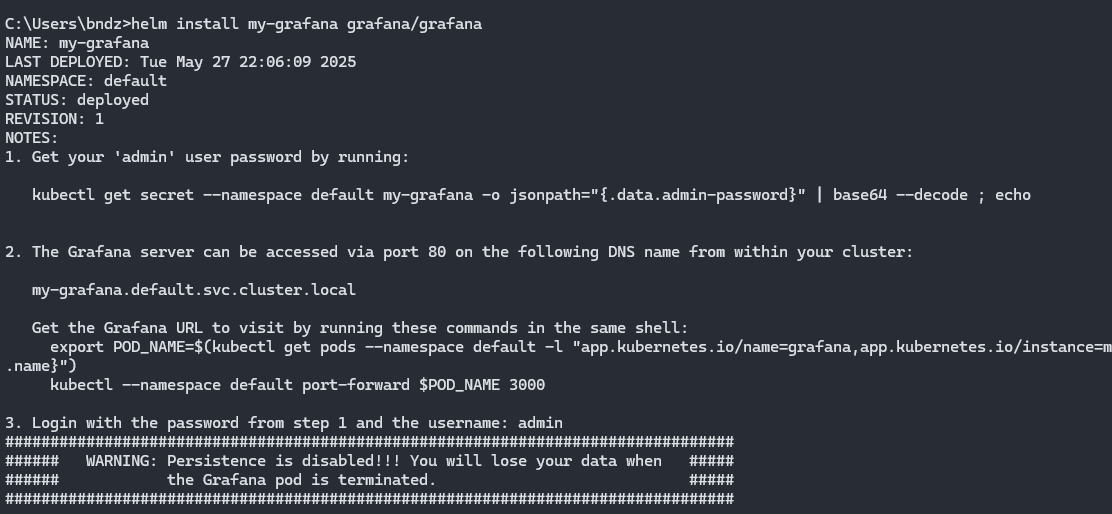


Install grafana

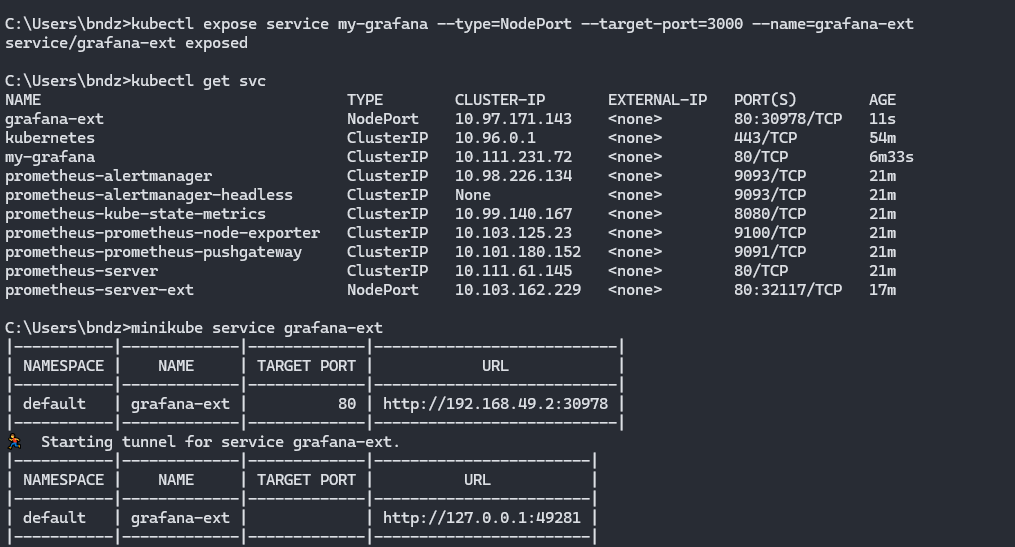
Add repo

helm repo add grafana https://grafana.github.io/helm-charts

Install

helm install my-grafana grafana/grafana

Expose grafana service

kubectl expose service my-grafana --type=NodePort --target-port=3000 --name=grafana-ext

Get secret

kubectl get secret --namespace monitoring my-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

-> Password: CUnxmrxBuKFcWUm8HERWxZbqgRQiB2a2ruOr1VVn

Open browser

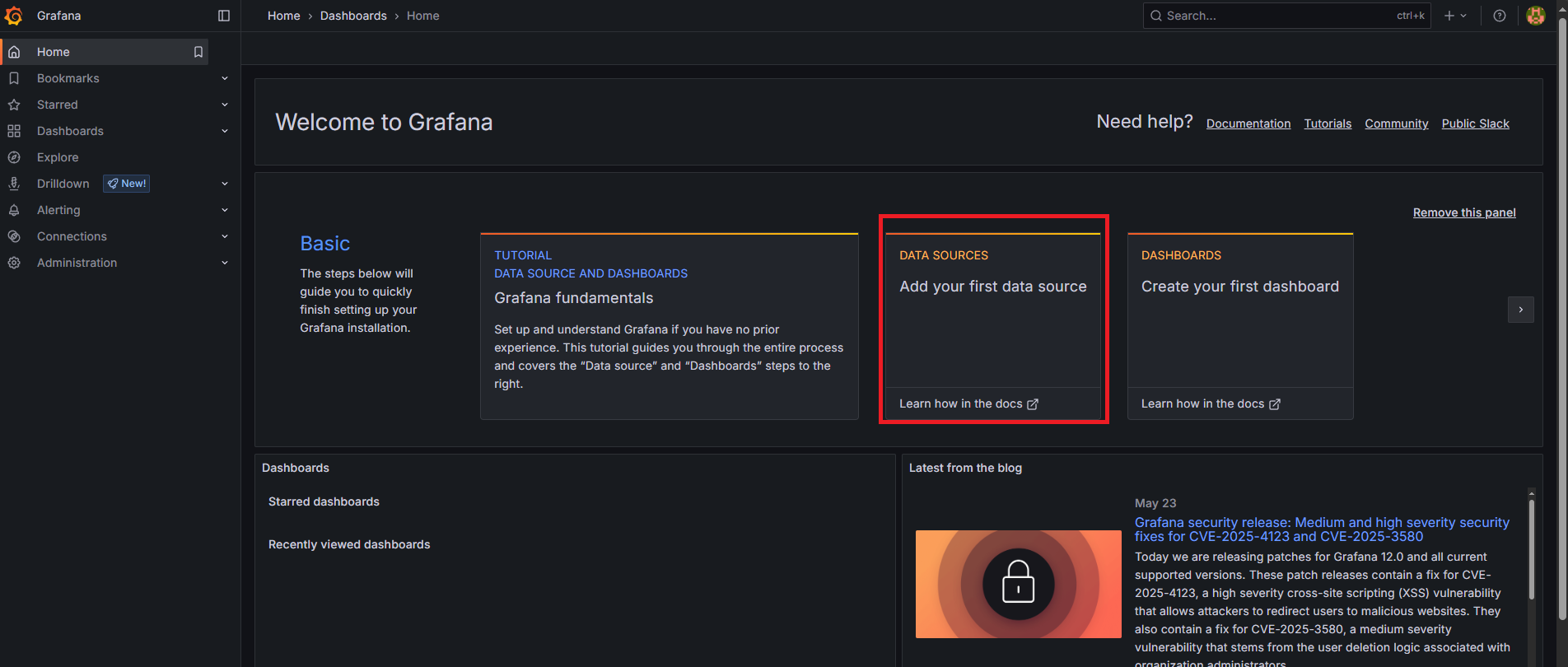
Login with

username: admin

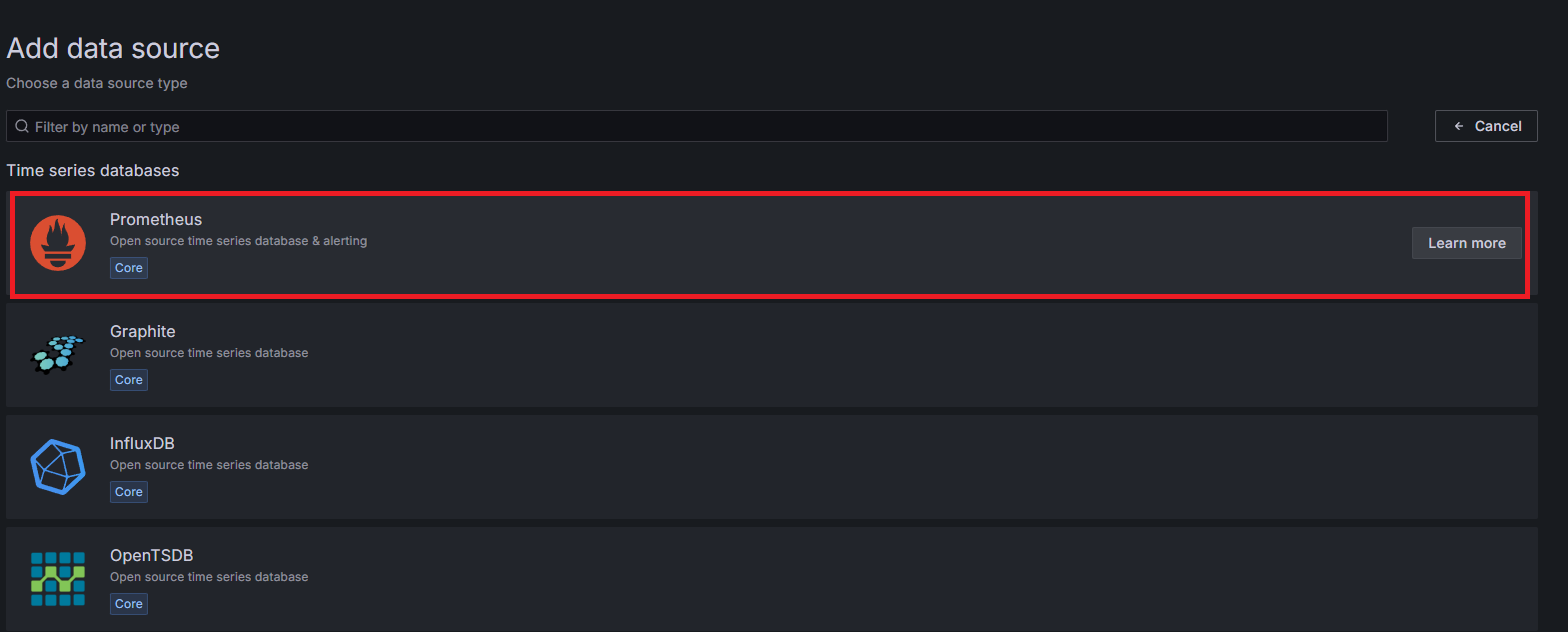
password: CUvqpLm2oXYy4DWKMz1fRNgTjU5e3bn7qAXFbQGm

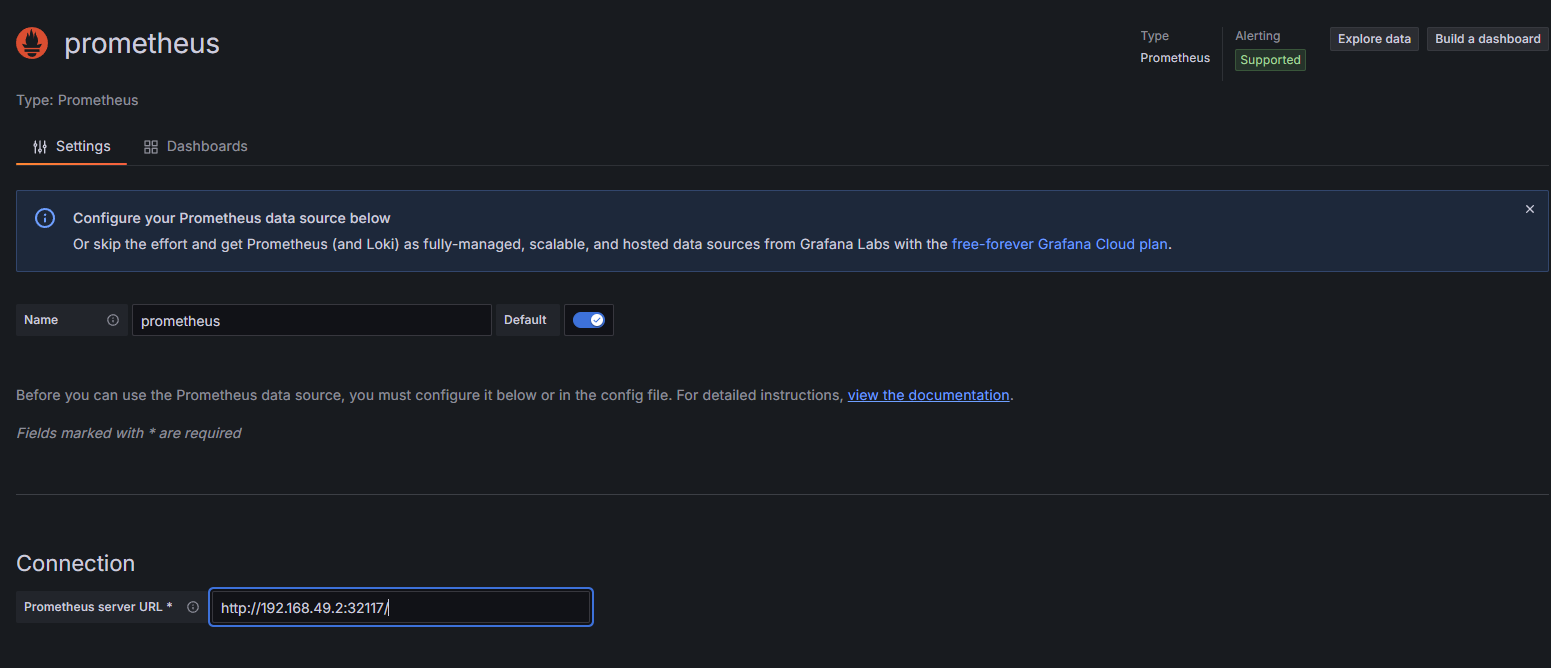


Add first data



Select Prometheus





Import dashboard

