**Provision Azure cloud resources**

1. **Create resource with terraform:**

[vtoan/sd4418\_azure\_infrastructure (github.com)](https://github.com/vtoan/sd4418_azure_infrastructure)

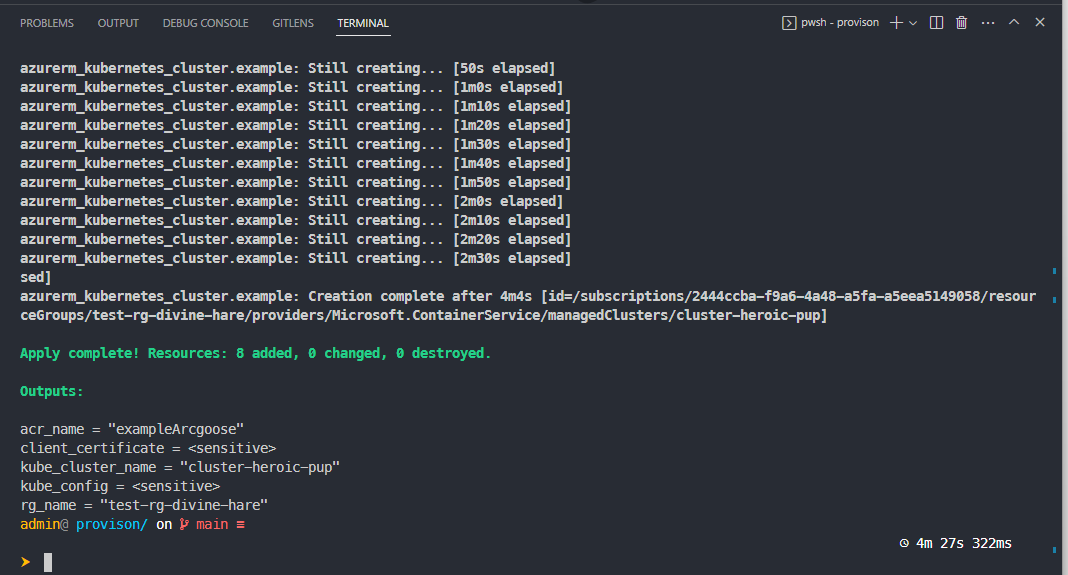
Cloud resource need:

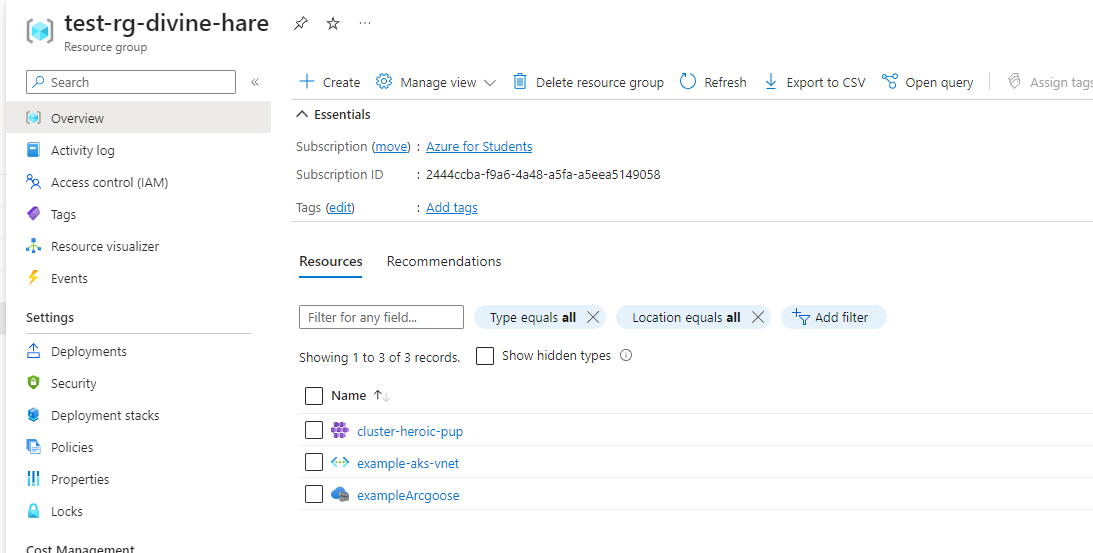
* Virtual Network for resources (VNET)
* Container Registry (ACR)
* Kubernetes Service (AKS)

terraform init

terraform plan -out main.tfplan

terraform apply "main.tfplan"





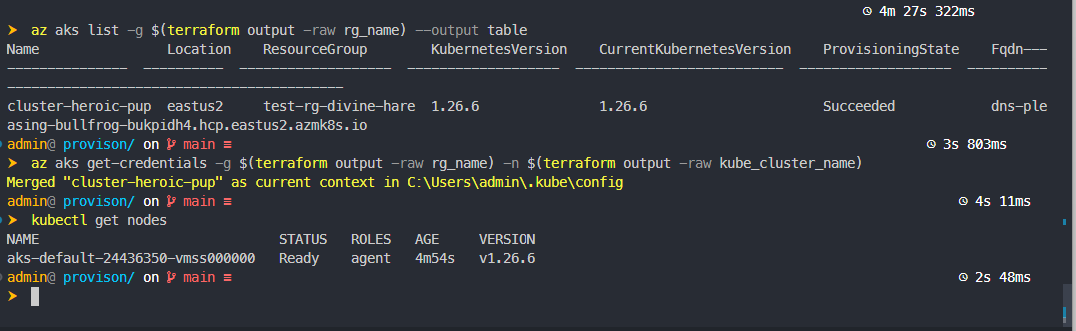
1. **Verify if the AKS has been created**

Merge the configuration of the **kubectl client** with the Azure AKS cluster.

az aks list -g $(terraform output -raw rg\_name) --output table

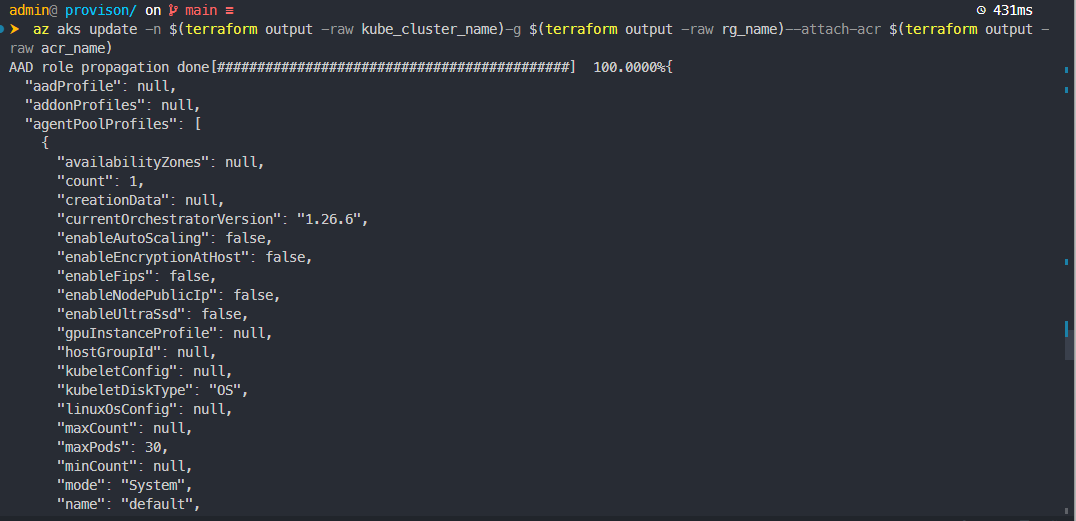
az aks get-credentials -g $(terraform output -raw rg\_name) -n $(terraform output -raw kube\_cluster\_name)

kubectl get nodes



Attach an ACR to an existing AKS cluster.

az aks update -n $(terraform output -raw kube\_cluster\_name)-g $(terraform output -raw rg\_name)--attach-acr $(terraform output -raw acr\_name)



**Setup Azure pipelines**

1. **Setup Agent for pipeline**

* Set up self-hosted agent with VMSS - [Azure Virtual Machine Scale Set agents - Azure Pipelines | Microsoft Learn](https://learn.microsoft.com/en-us/azure/devops/pipelines/agents/scale-set-agents?view=azure-devops)
* Script for cloud-int dependencies of self-hosted VM.

#cloud-config

package\_upgrade: true

runcmd:

- sudo install -m 0755 -d /etc/apt/keyrings

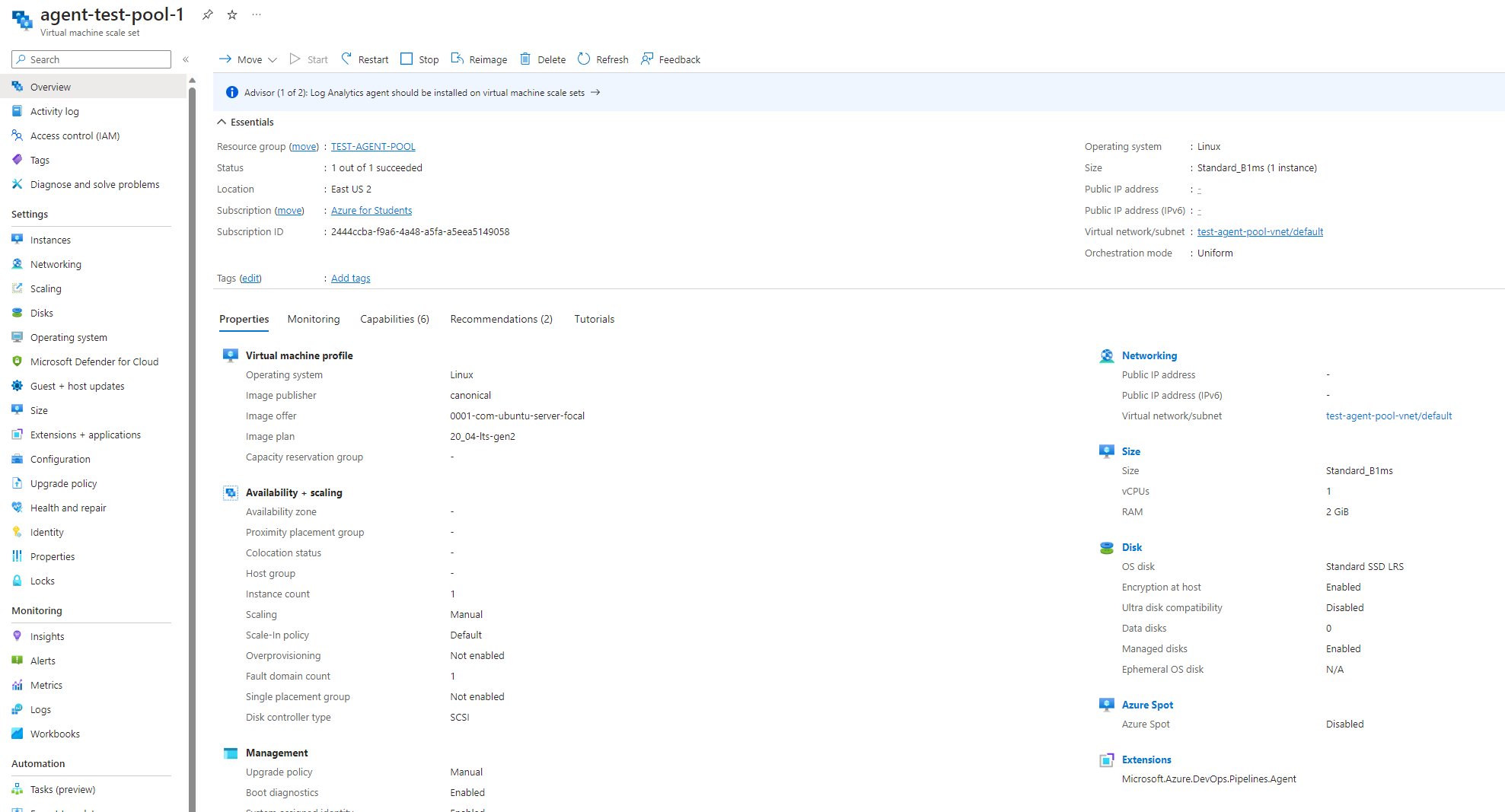
- curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

- sudo chmod a+r /etc/apt/keyrings/docker.gpg

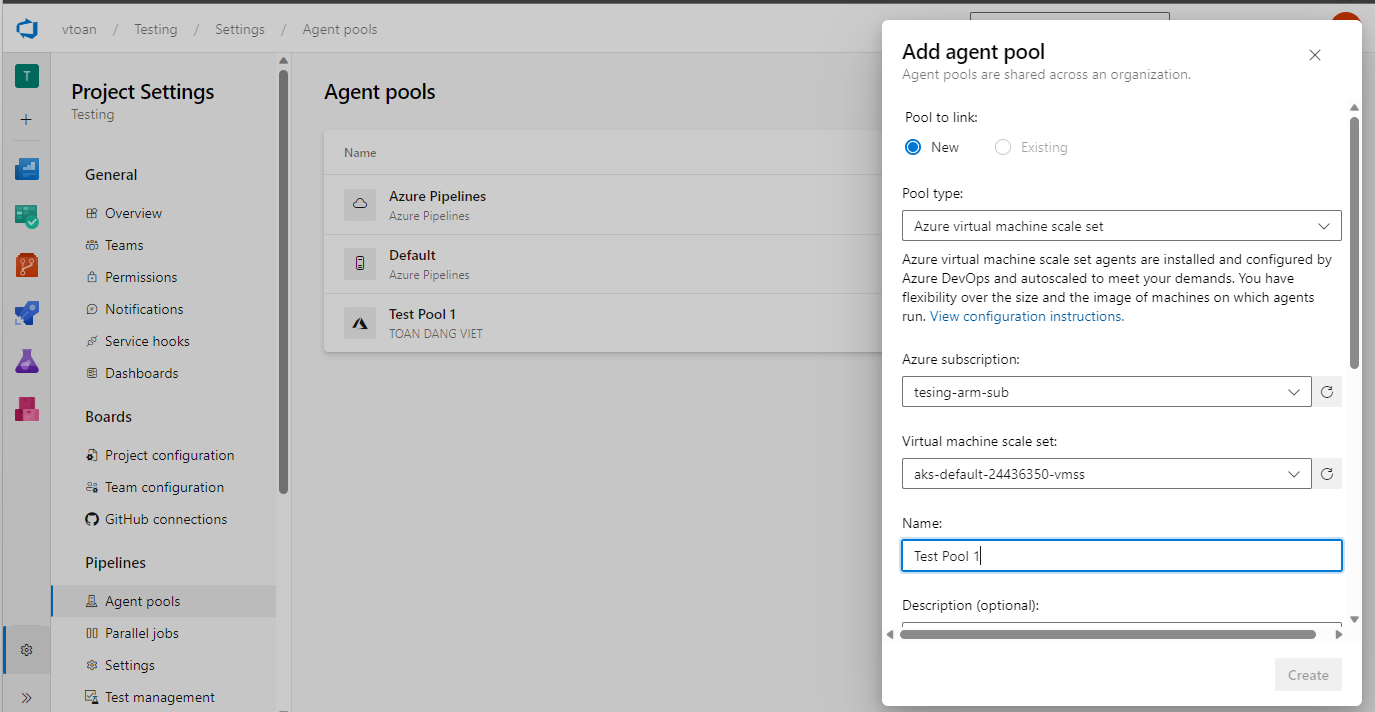
- echo "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] <https://download.docker.com/linux/ubuntu>" $(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

- sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io -y

- sudo chmod 666 /var/run/docker.sock

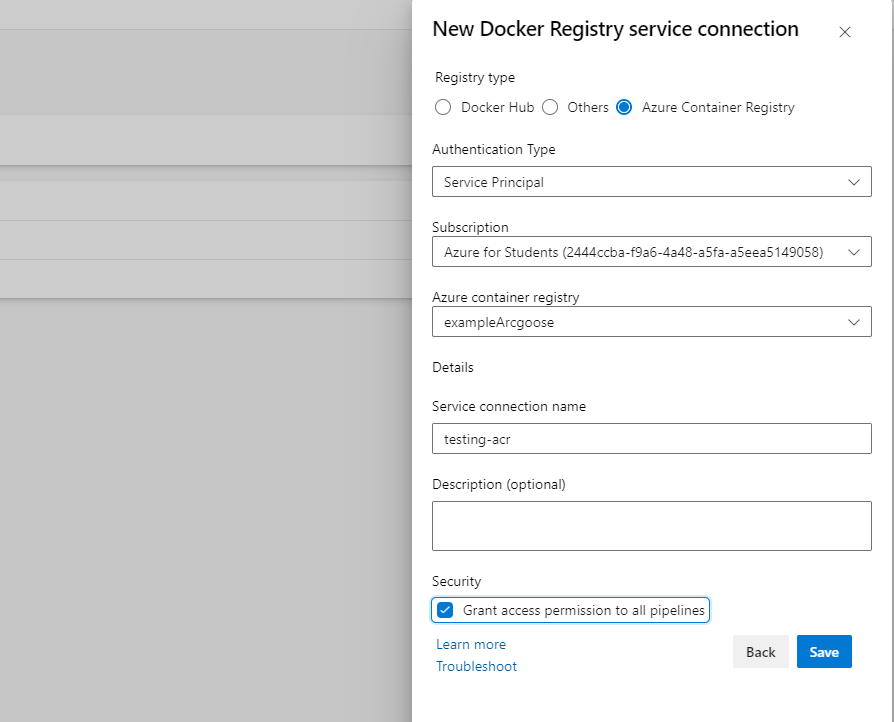


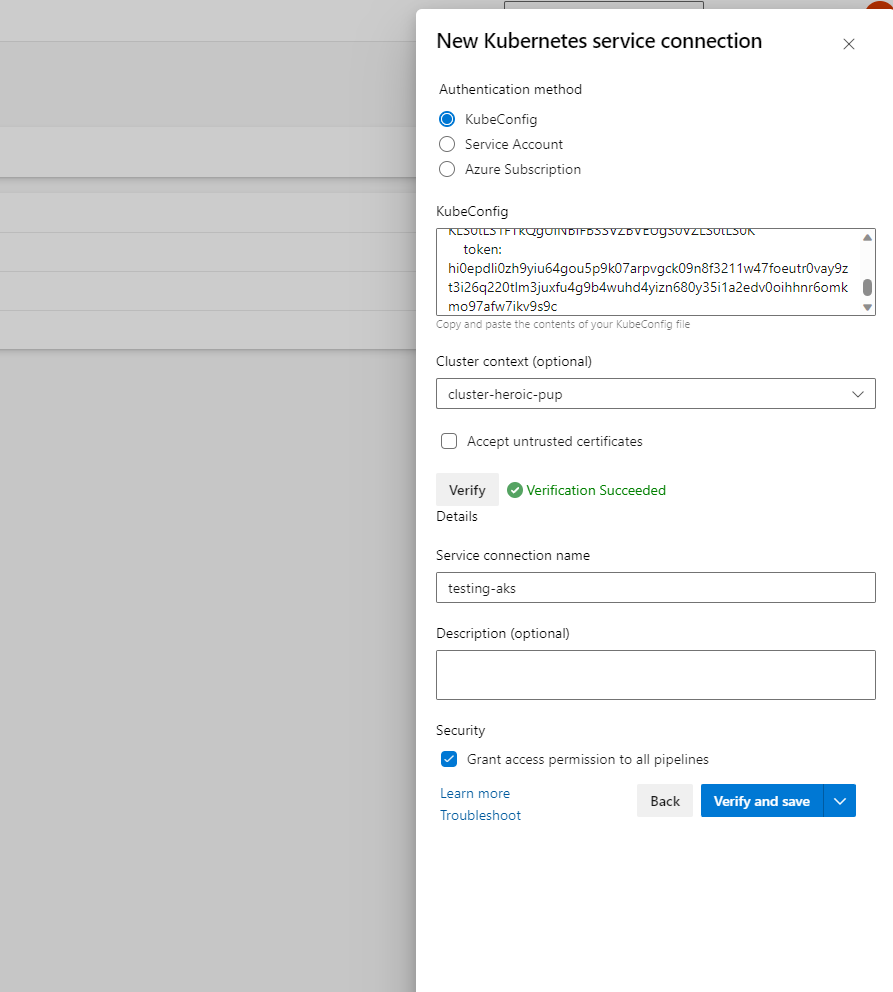
Add Agent Pool to DevOpss

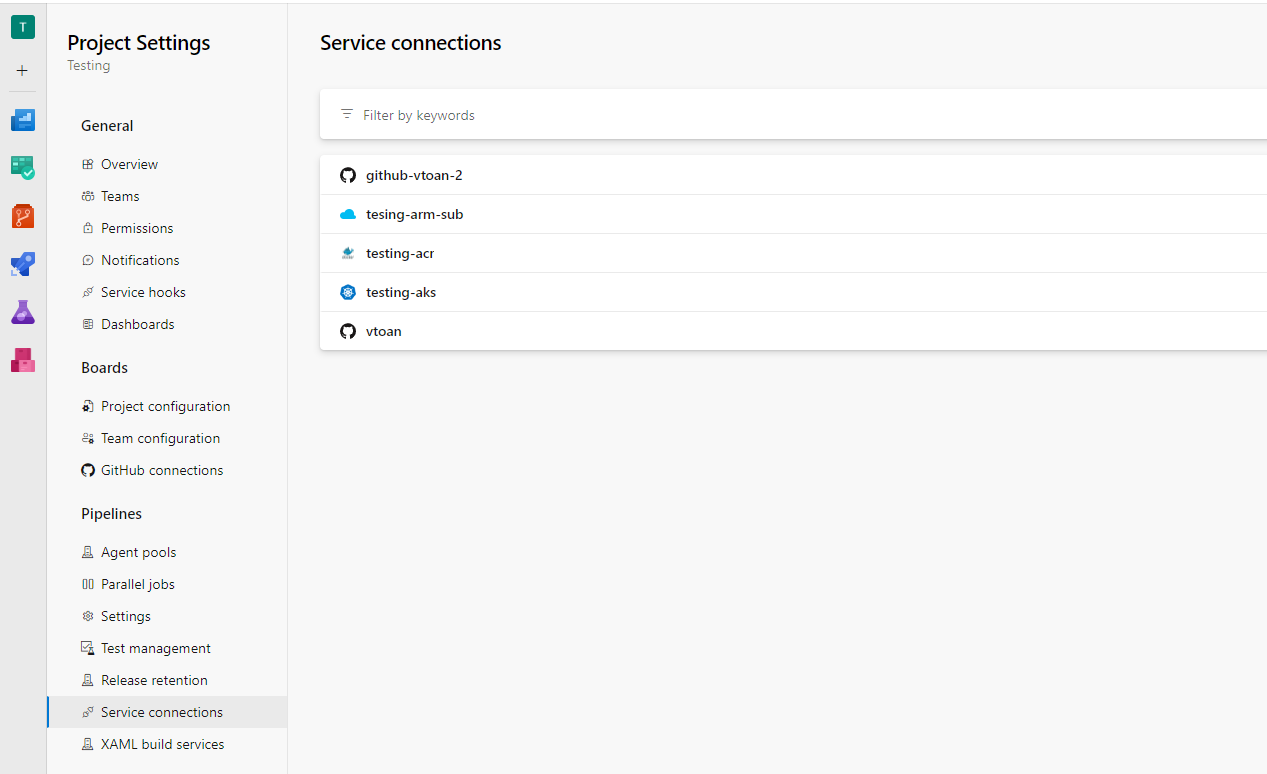


1. **Create Azure Pipelines for CI/CD**

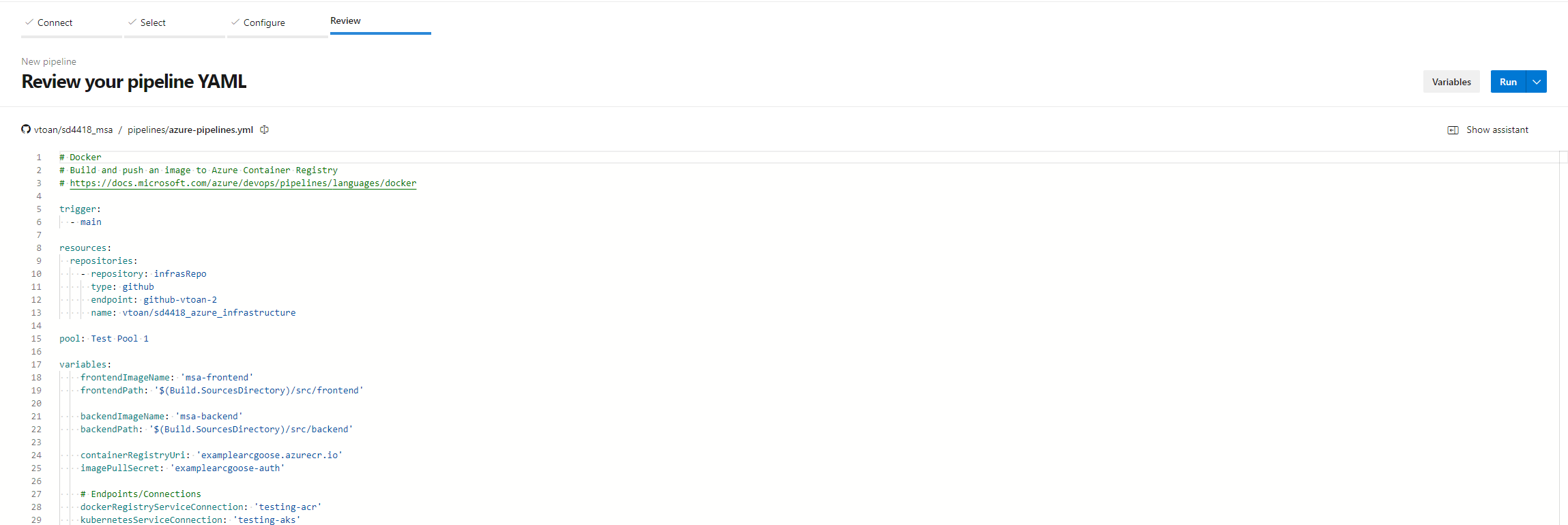
* Create Azure Service Connections /Service Endpoints for:
  + Github - with name “github-vtoan”
  + Container Registry (ACR) - with name “testing-acr”
  + Kubernetes Service (AKS) - with name “testing-aks”

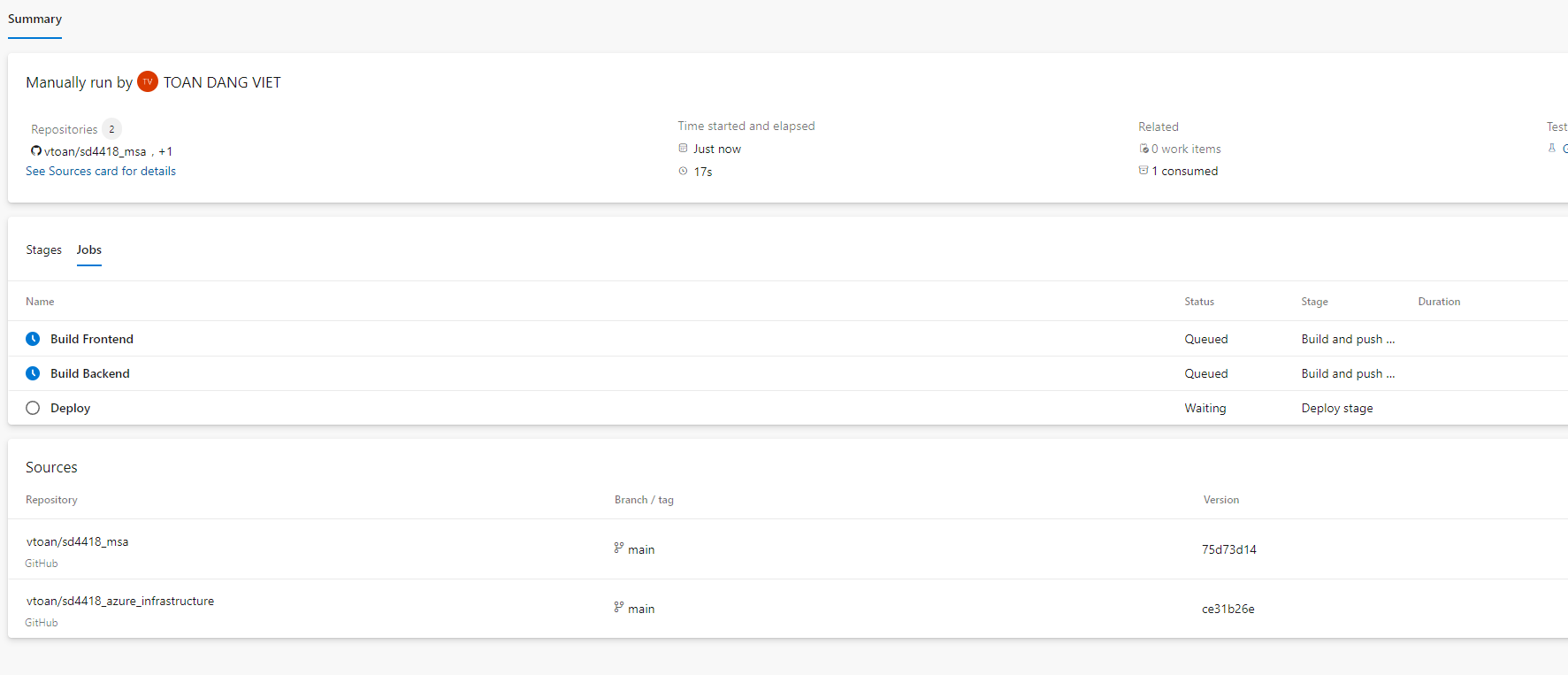


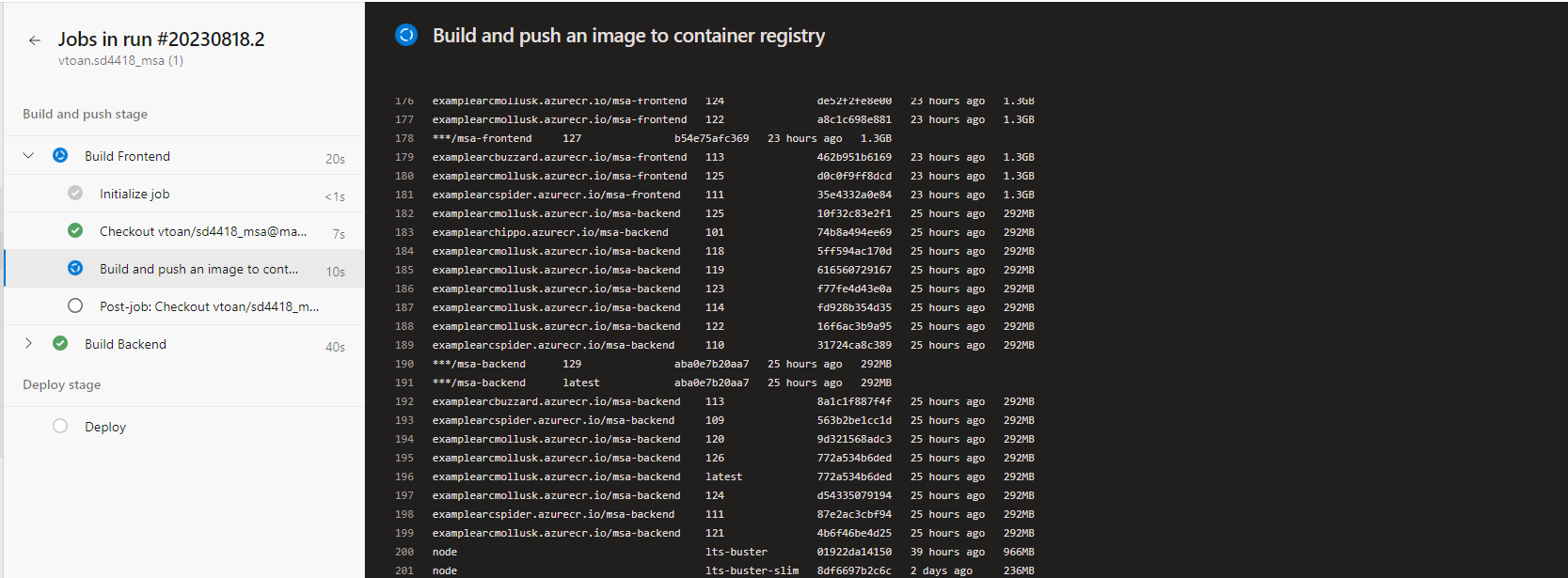


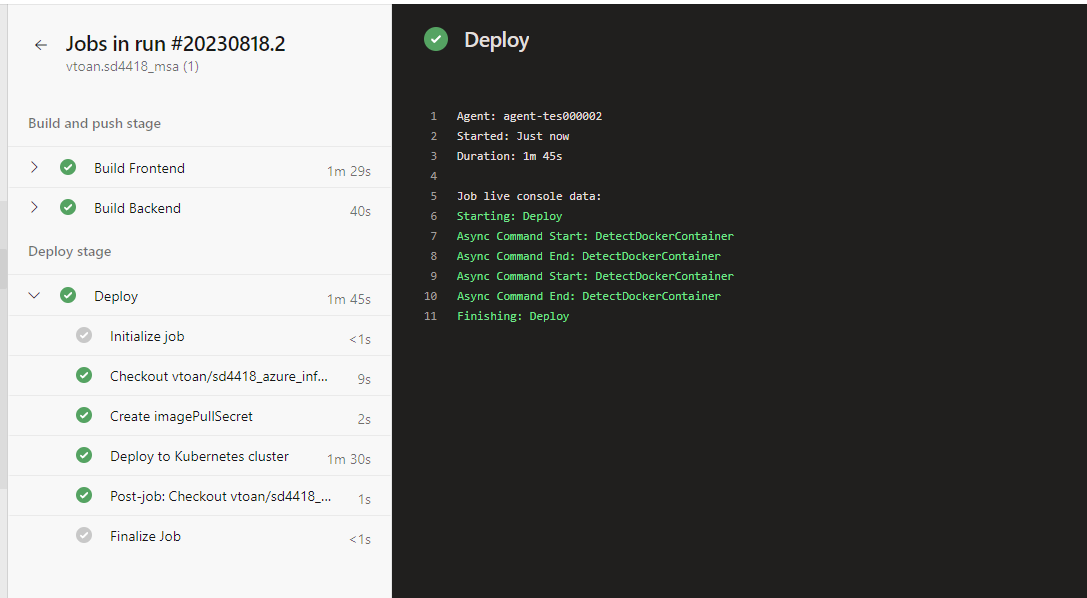


* **Set up Azure CI/CD Pipeline**





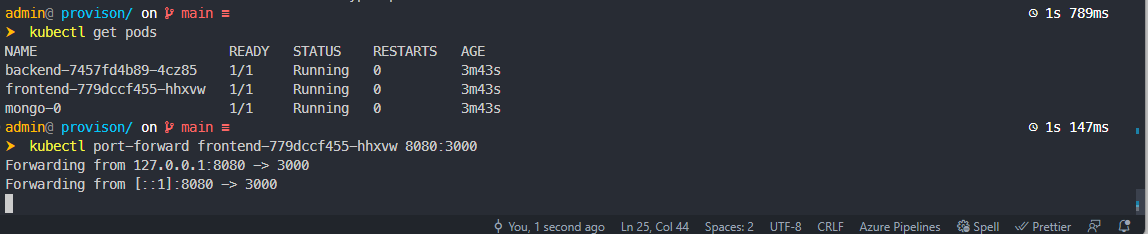




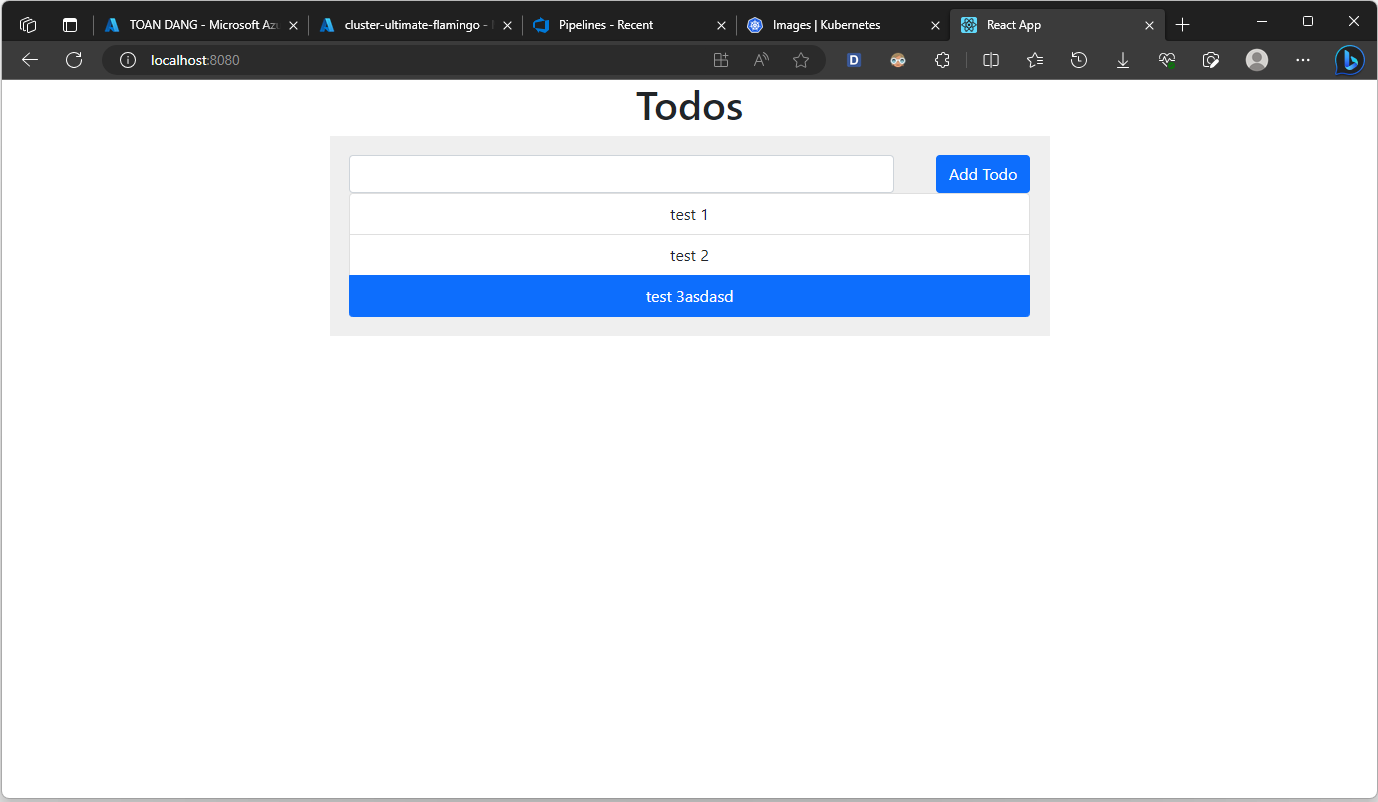
1. **Access sample application is running**

kubectl get pobs

kubectl port-forward frontend-88bd8bf89-cwj96 8080:3000



Open in browser <http://localhost:8080/>



**Set up monitoring with Prometheus and Grafana**

1. **Create cluster namespace for monitoring tools**

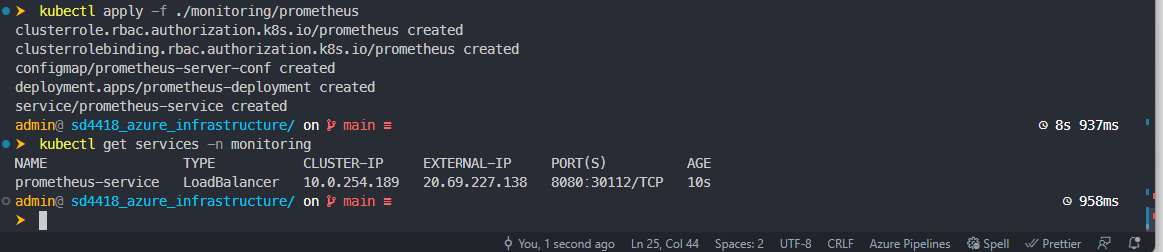
kubectl create namespace monitoring

1. **Deploy Prometheus on Kubernetes**

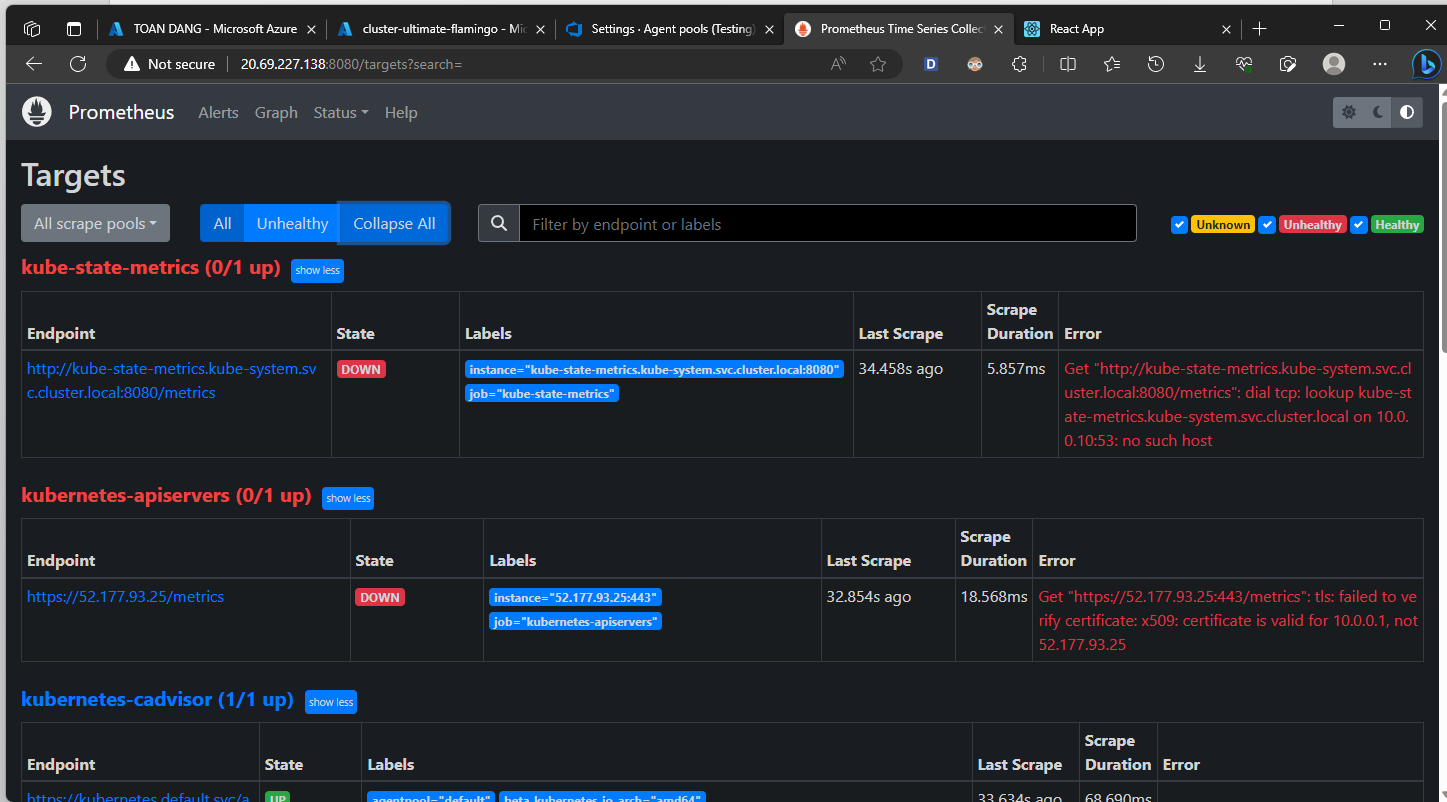
[vtoan/sd4418\_azure\_infrastructure (github.com)](https://github.com/vtoan/sd4418_azure_infrastructure)

kubectl apply -f ./monitoring/prometheus

kubectl get services -n monitoring



Access to Prometheus Dashboard with **LoadBalancer** IP.

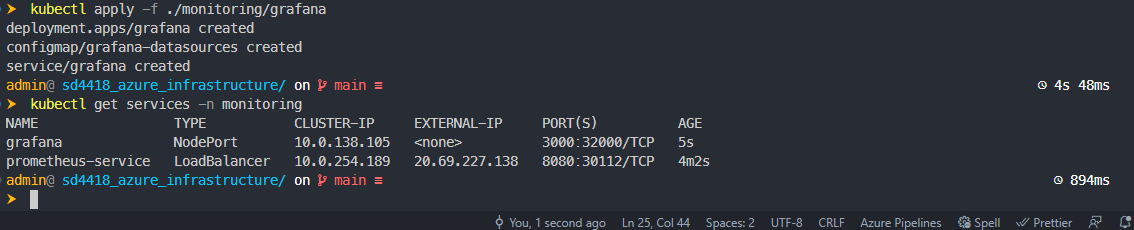


1. **Deploy Grafana on Kubernetes**

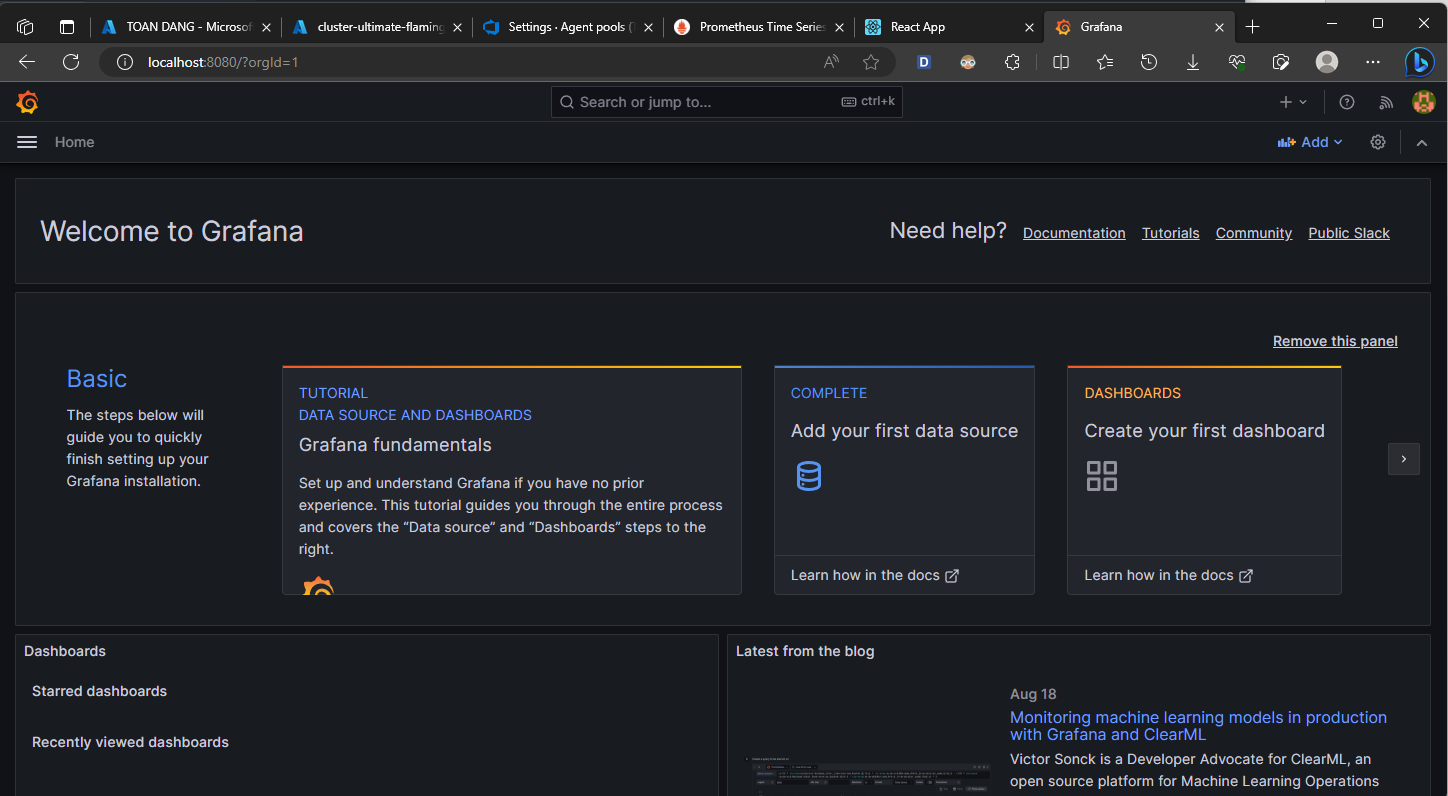
[vtoan/sd4418\_azure\_infrastructure (github.com)](https://github.com/vtoan/sd4418_azure_infrastructure)

kubectl apply -f ./monitoring/grafana

kubectl get services -n monitoring



Access to Grafana with default credentials: admin/admin



Get and import a template for a Grafana dashboard.

* Get template ID: [Kubernetes cluster monitoring (via Prometheus) | Grafana Labs](https://grafana.com/grafana/dashboards/315-kubernetes-cluster-monitoring-via-prometheus/)
* Import template to site.

