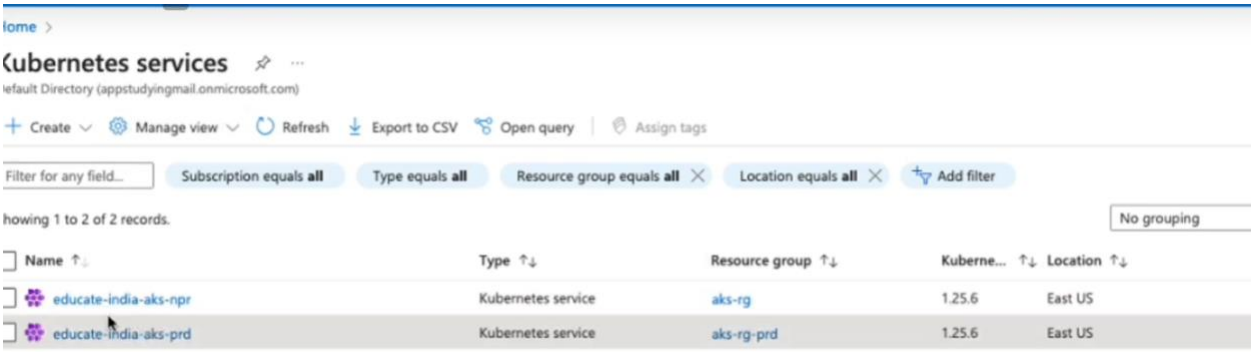
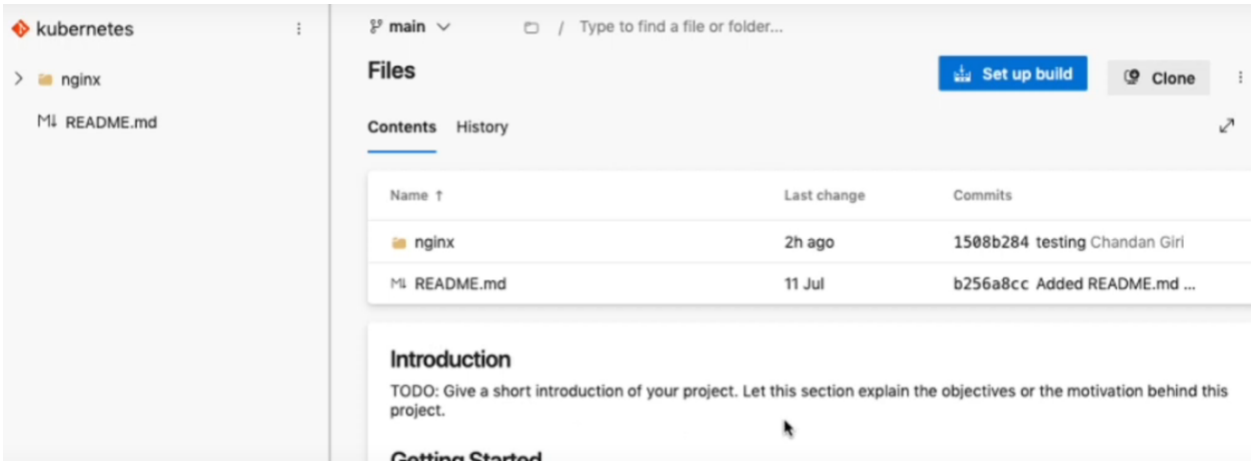
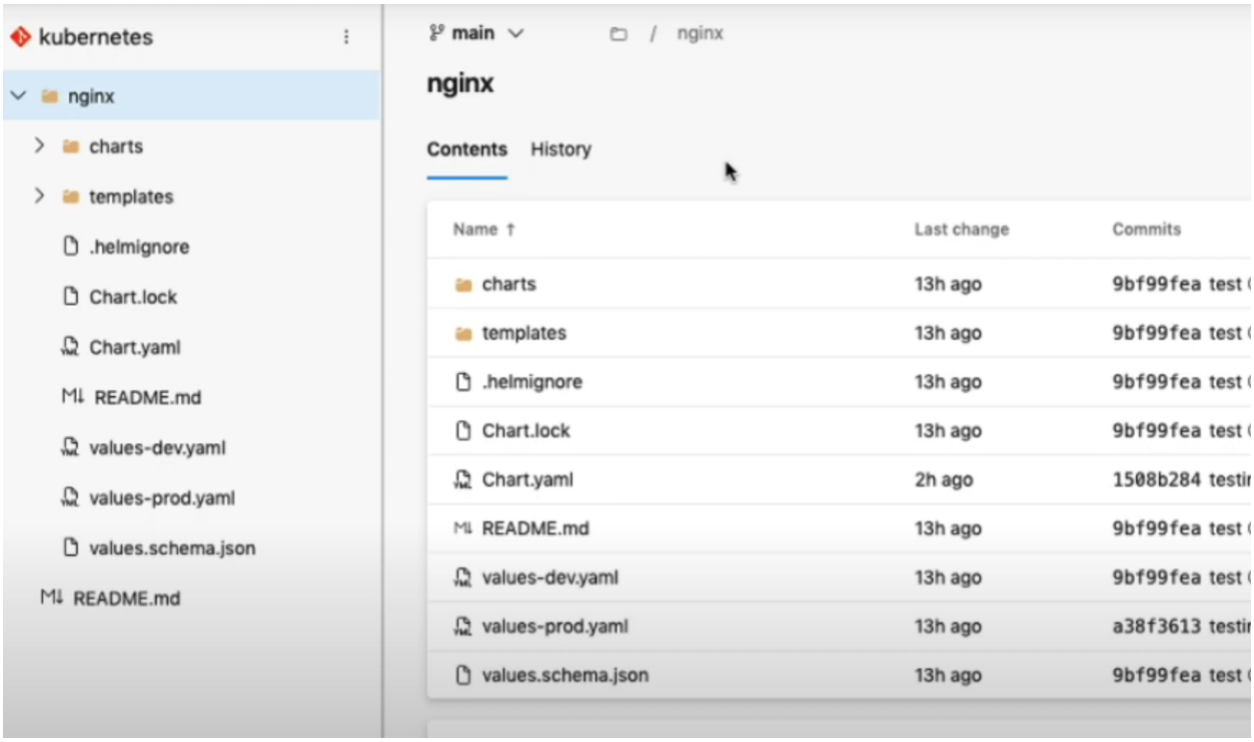
**Helm Deployment using Azure DevOps**

We'll create a Helm pipeline with Azure DevOps using yaml file and will add Helm related tasks there so our target is to create a multi-stage pipeline that can use a single Azure chart and can deploy to multiple env. We'll target to deploy dev and prod env.

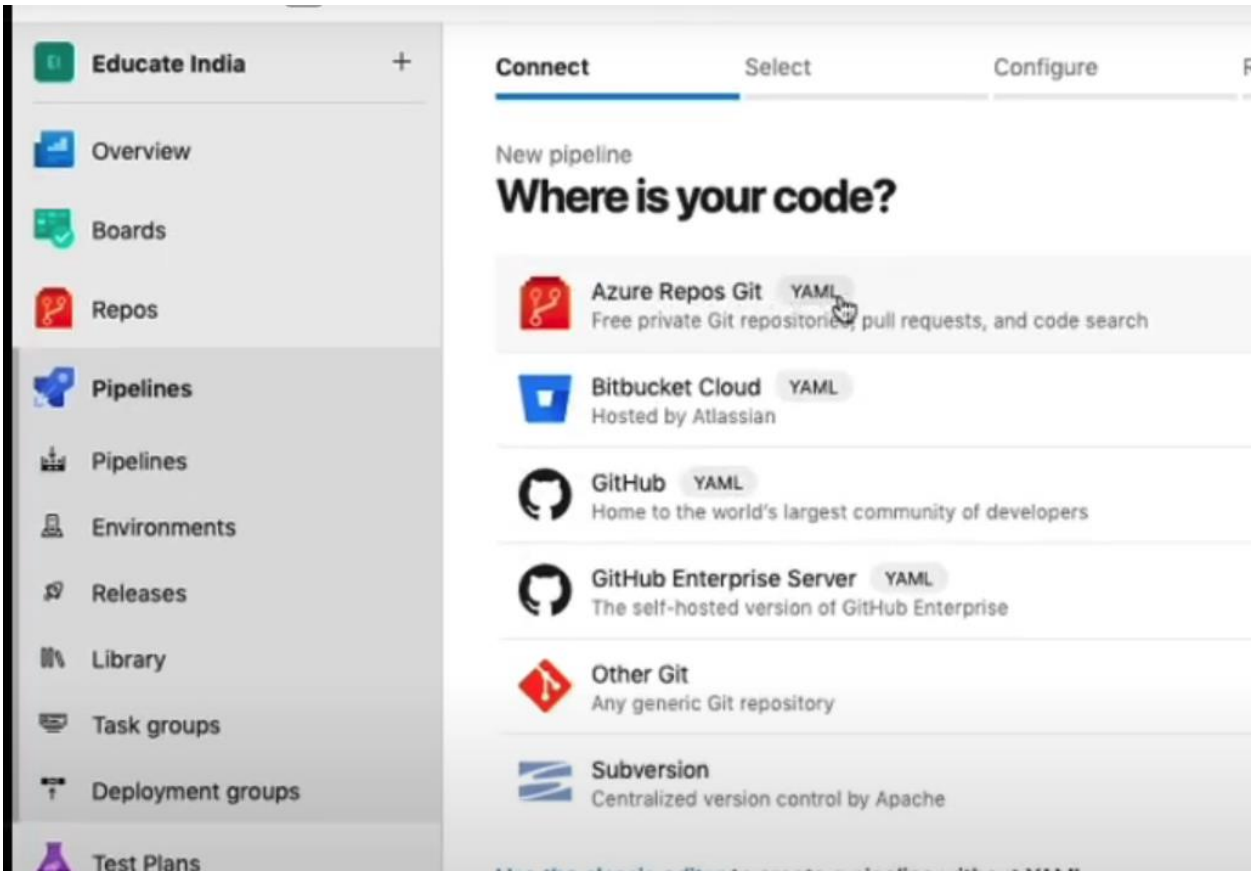
Create 2 clusters and pushed nginx chart Helm chart to Azure devops



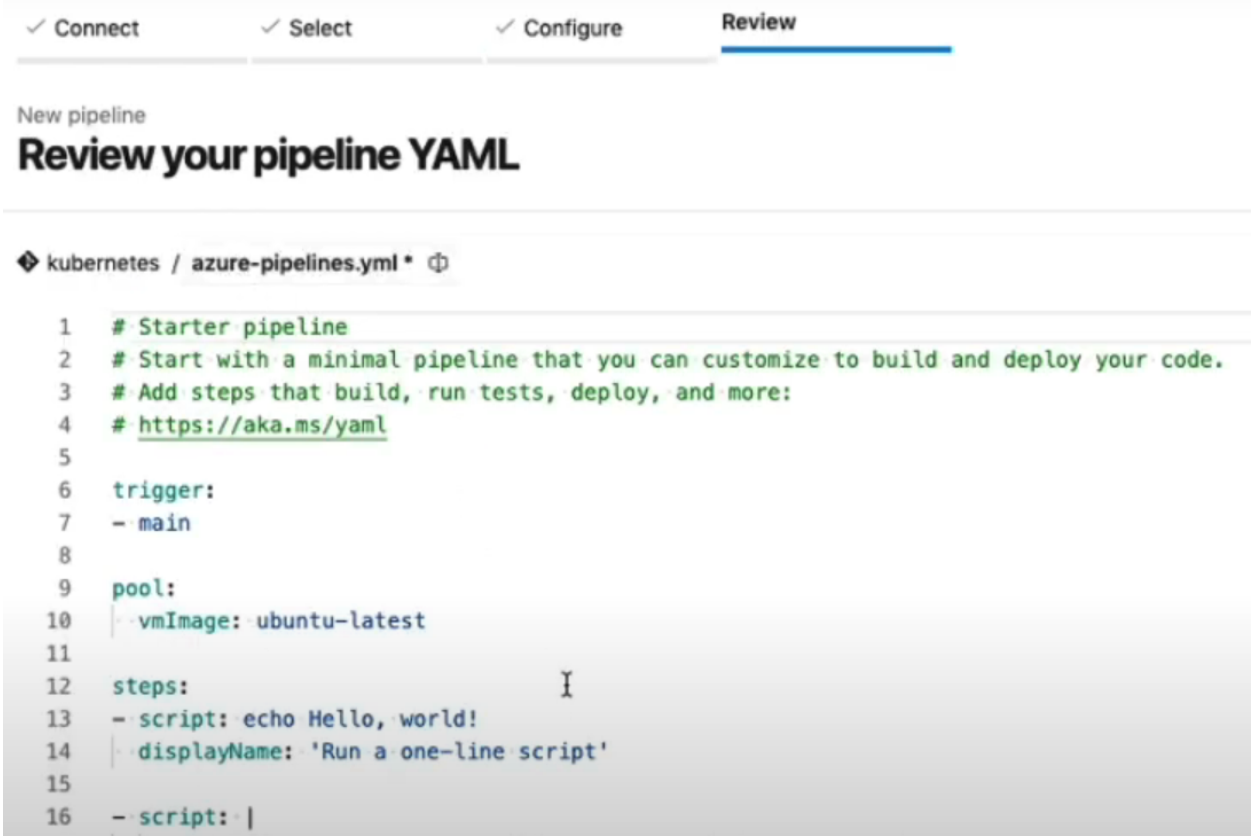





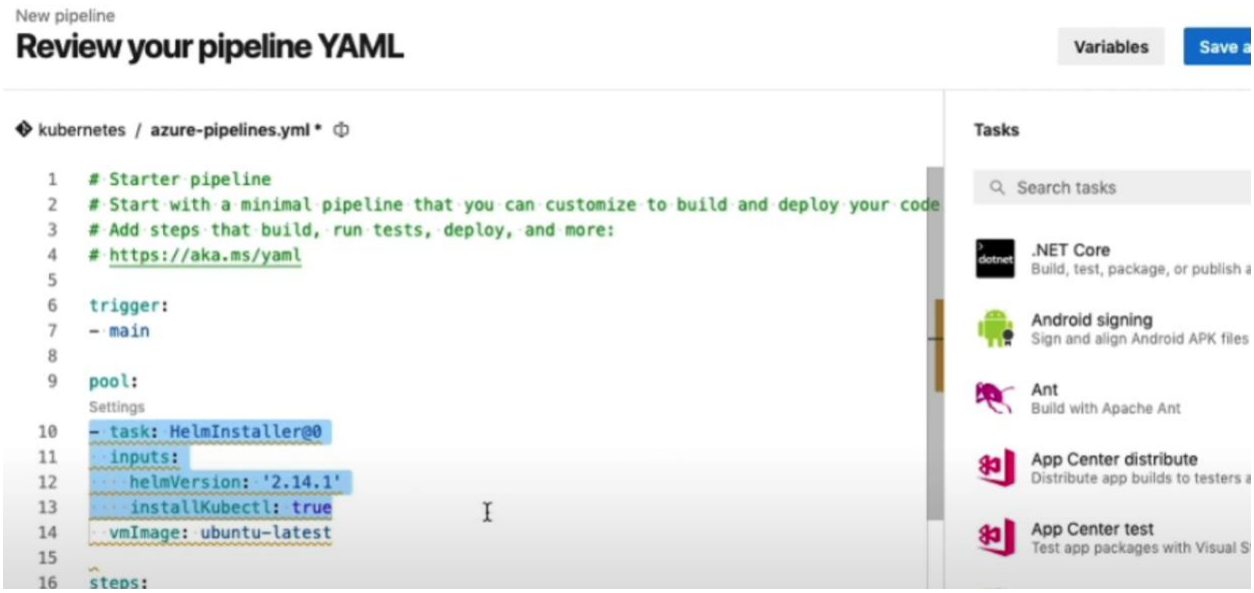

Go to pipelines and go to Azure repos git:



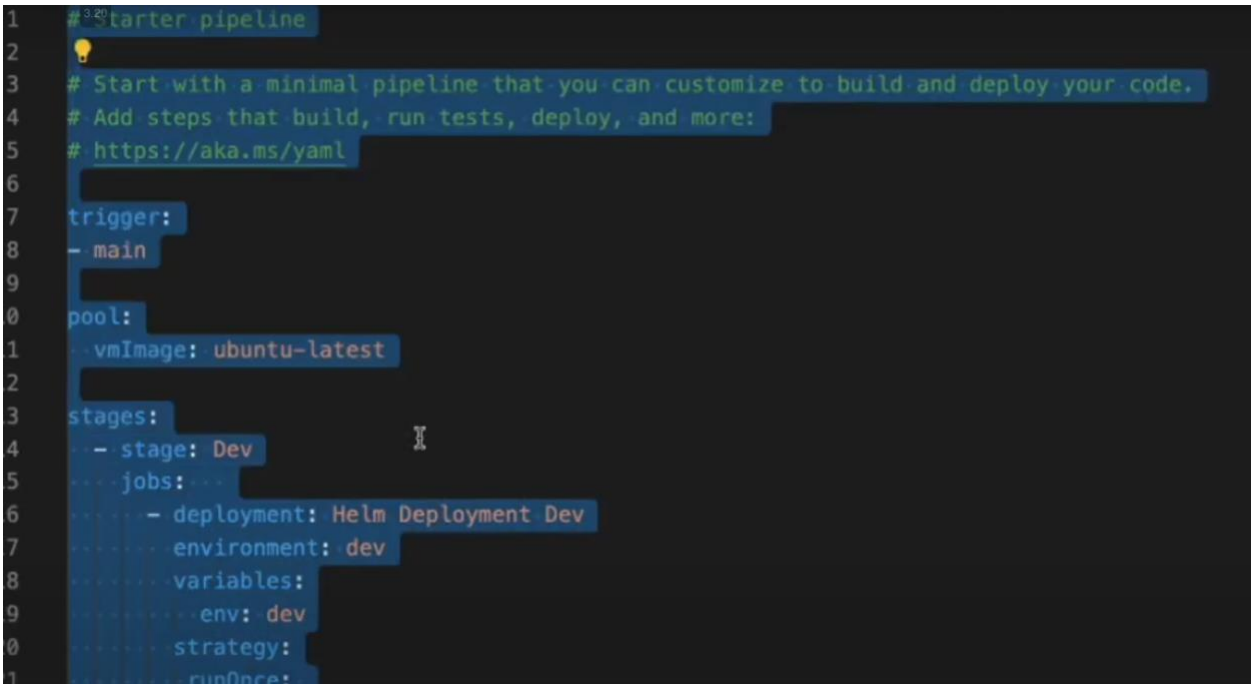

Select Kubernetes repo and starter pipeline




You can add tasks like helm installer:




Copy the code and paste it in the pipeline:



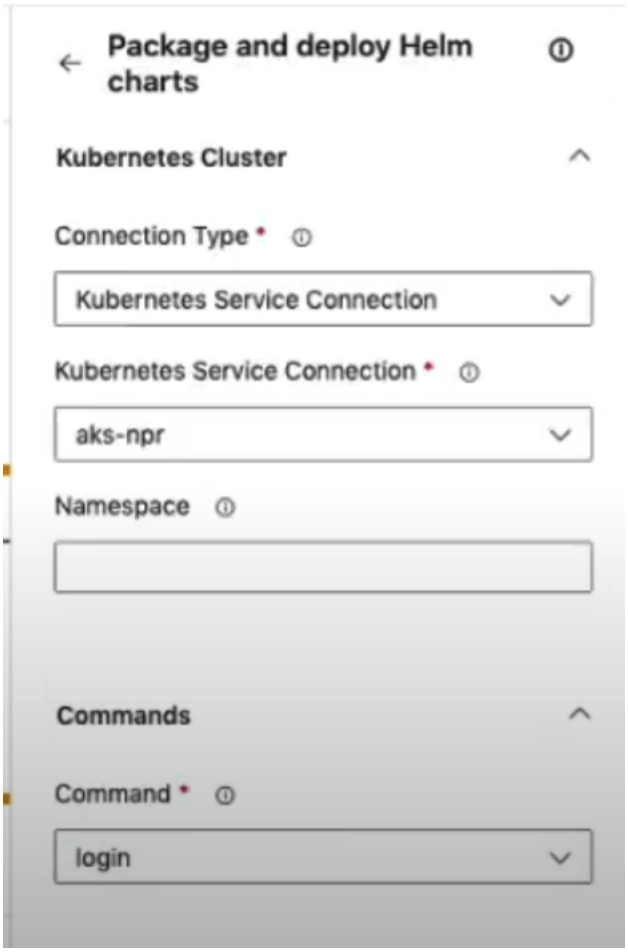
```yaml
# Starter pipeline

# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
- main

pool:
  vmImage: ubuntu-latest

stages:
  - stage: Dev
    jobs:
      - deployment: Helm Deployment Dev
        environment: dev
        variables:
          env: dev
        strategy:
          runOnce:
```

New pipeline

# Review your pipeline YAML

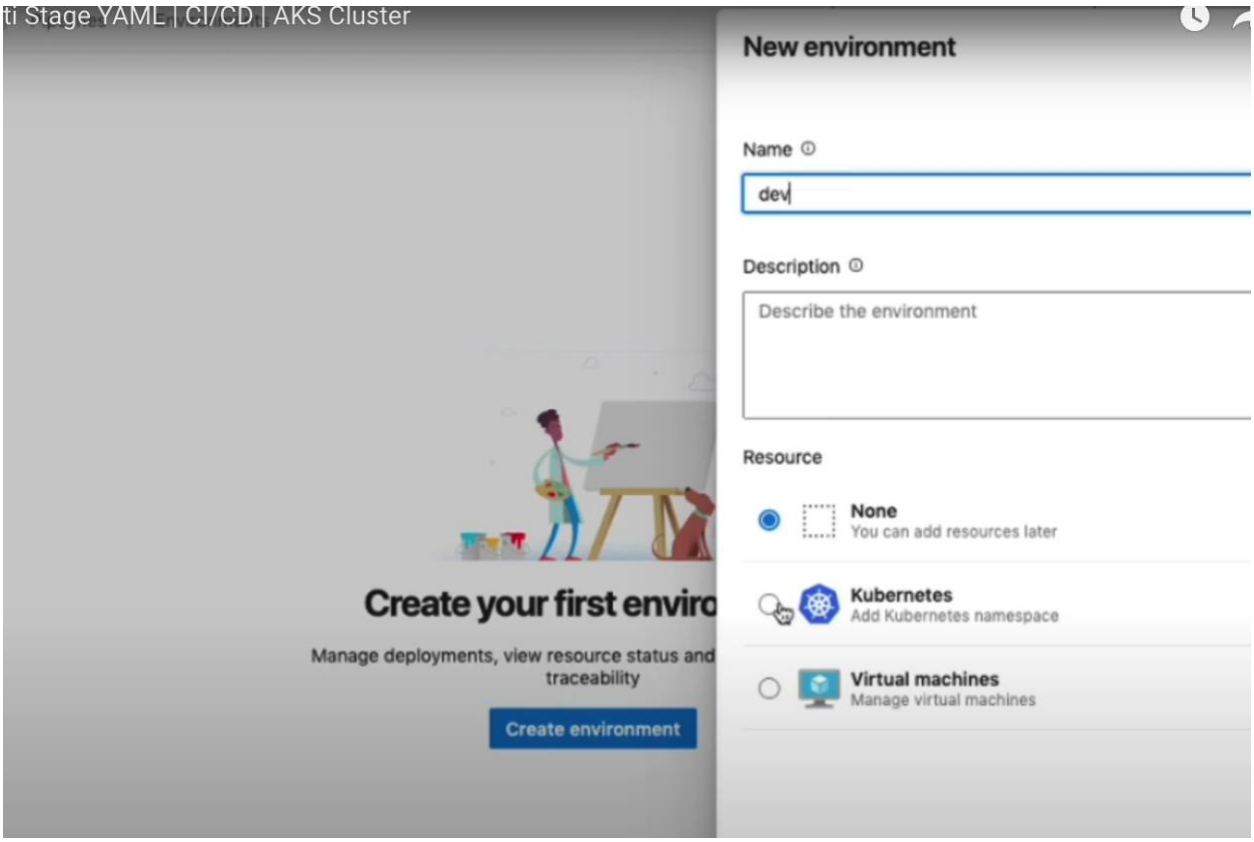
kubernetes / azure-pipelines.yml *

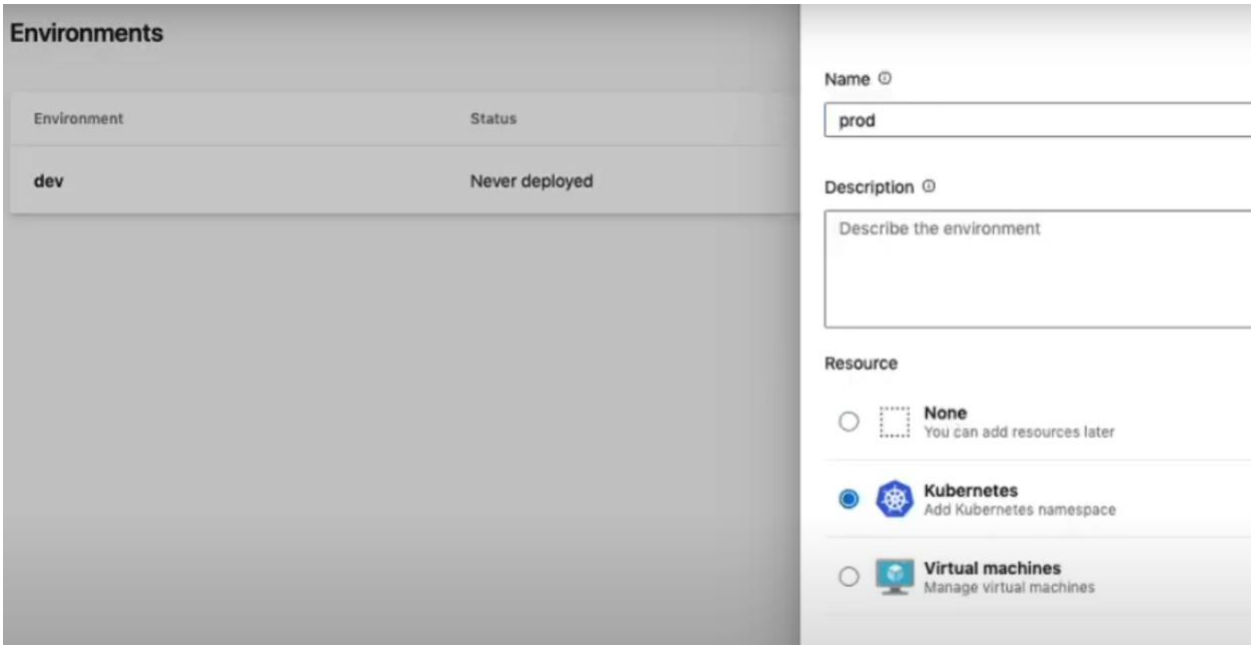
```yaml
trigger:
- main

pool:
  vmImage: ubuntu-latest

stages:
  - stage: Dev
    jobs:
      - deployment: Helm Deployment Dev
        environment: dev
        variables:
          env: dev
        strategy:
          runOnce:
```




```
                script:
- task: HelmDeploy@0
  inputs:
    connectionType: 'Kubernetes Service Connection'
    kubernetesServiceConnection: 'aks-npr'
    command: 'login'
                helm upgrade --install test nginx -f $(Agent.BuildDi
- stage: Prod
  jobs:
```

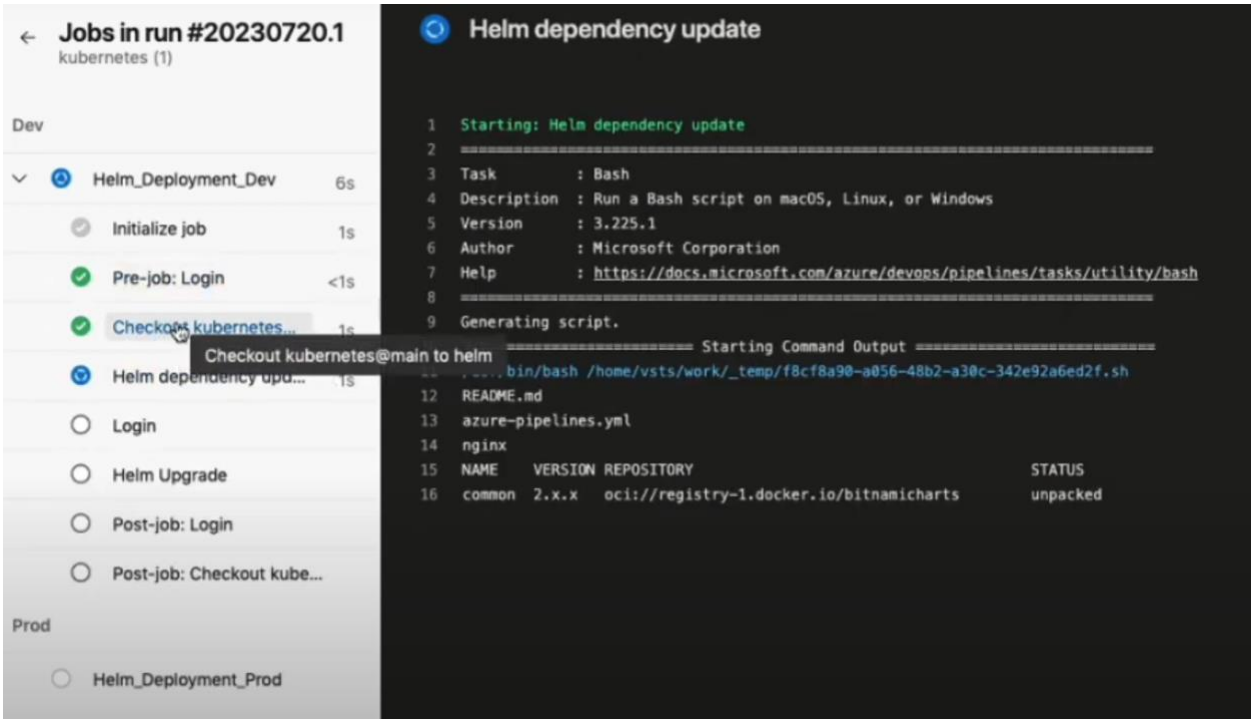
Create env dev and prod:


ti Stage YAML | CI/CD | AKS Cluster
New environment
Name
dev
Description
Describe the environment
Resource
None
You can add resources later
Kubernetes
Add Kubernetes namespace
Virtual machines
Manage virtual machines
Create your first enviro
Manage deployments, view resource status and traceability
Create environment

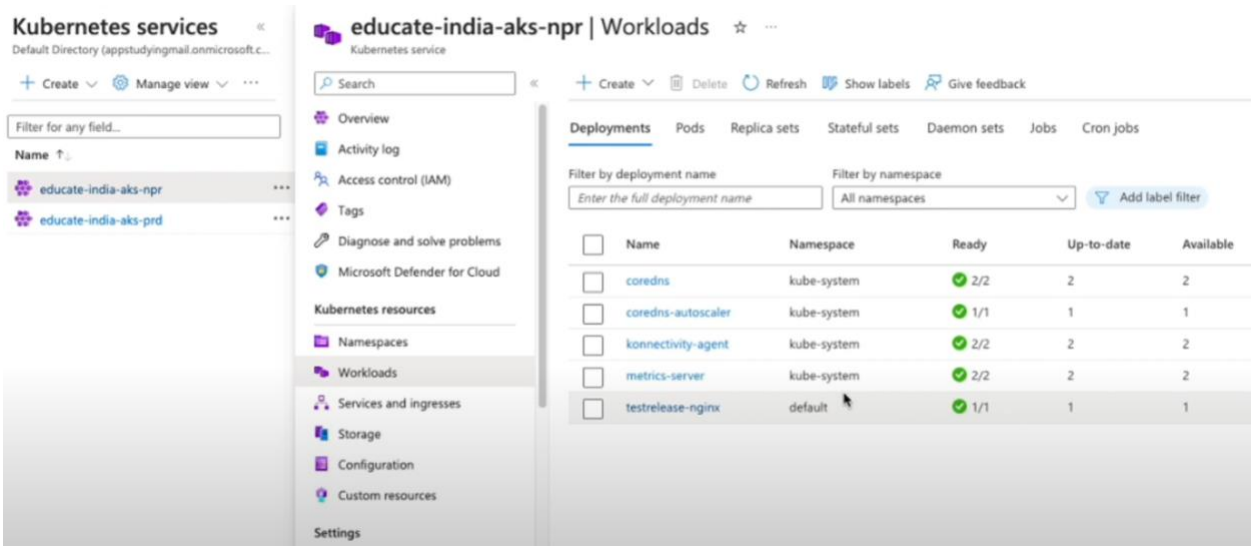

Environments
Environment
Status
dev
Never deployed
Name
prod
Description
Describe the environment
Resource
None
You can add resources later
Kubernetes
Add Kubernetes namespace
Virtual machines
Manage virtual machines

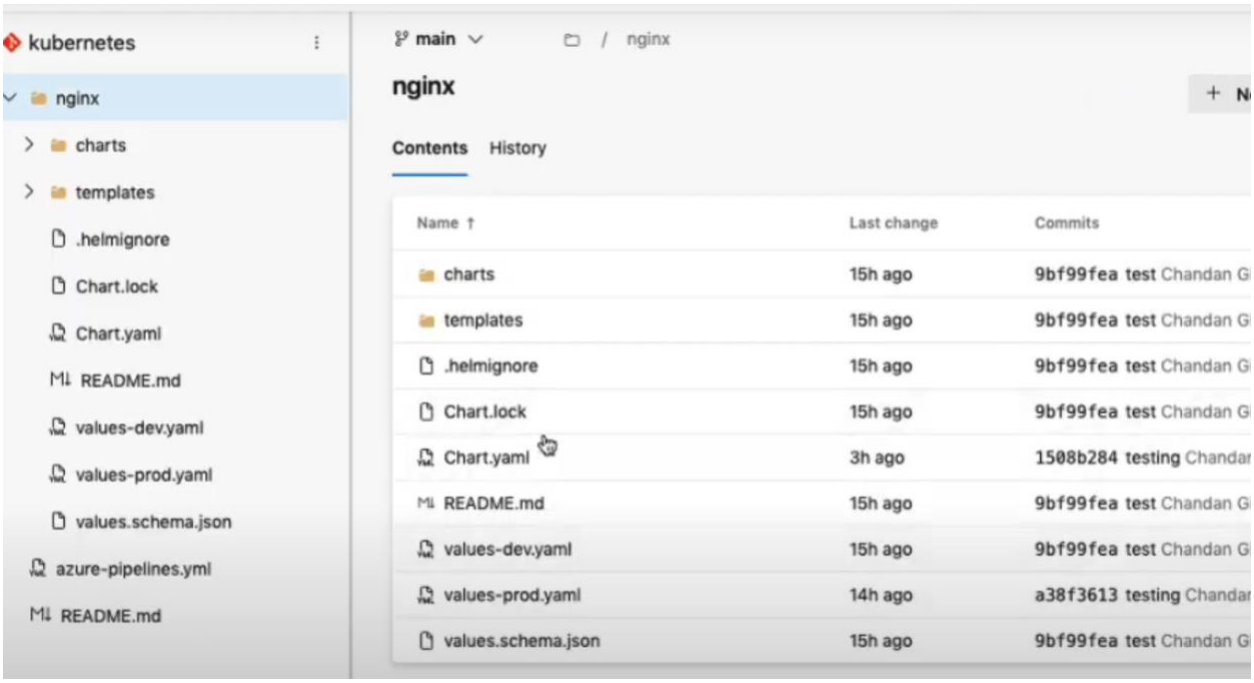
```
            helm upgrade --install test nginx -f $(Agent
- stage: Prod
  jobs:
    - deployment: Helm_Deployment_Prod
      environment: prod
      variables:
        env: prod
      strategy:
        runOnce:
          deploy:
            steps:
              - checkout: self
                clean: true
                submodules: true | recursive
                path: helm
                displayName: Checkout
```

Settings




Tesrelease is here:




So that's how we can manage the Helm chart within Azure repos and create a simple yaml that will fetch the chart and it will create the release and later manage that particular env by creating different values for that different stages