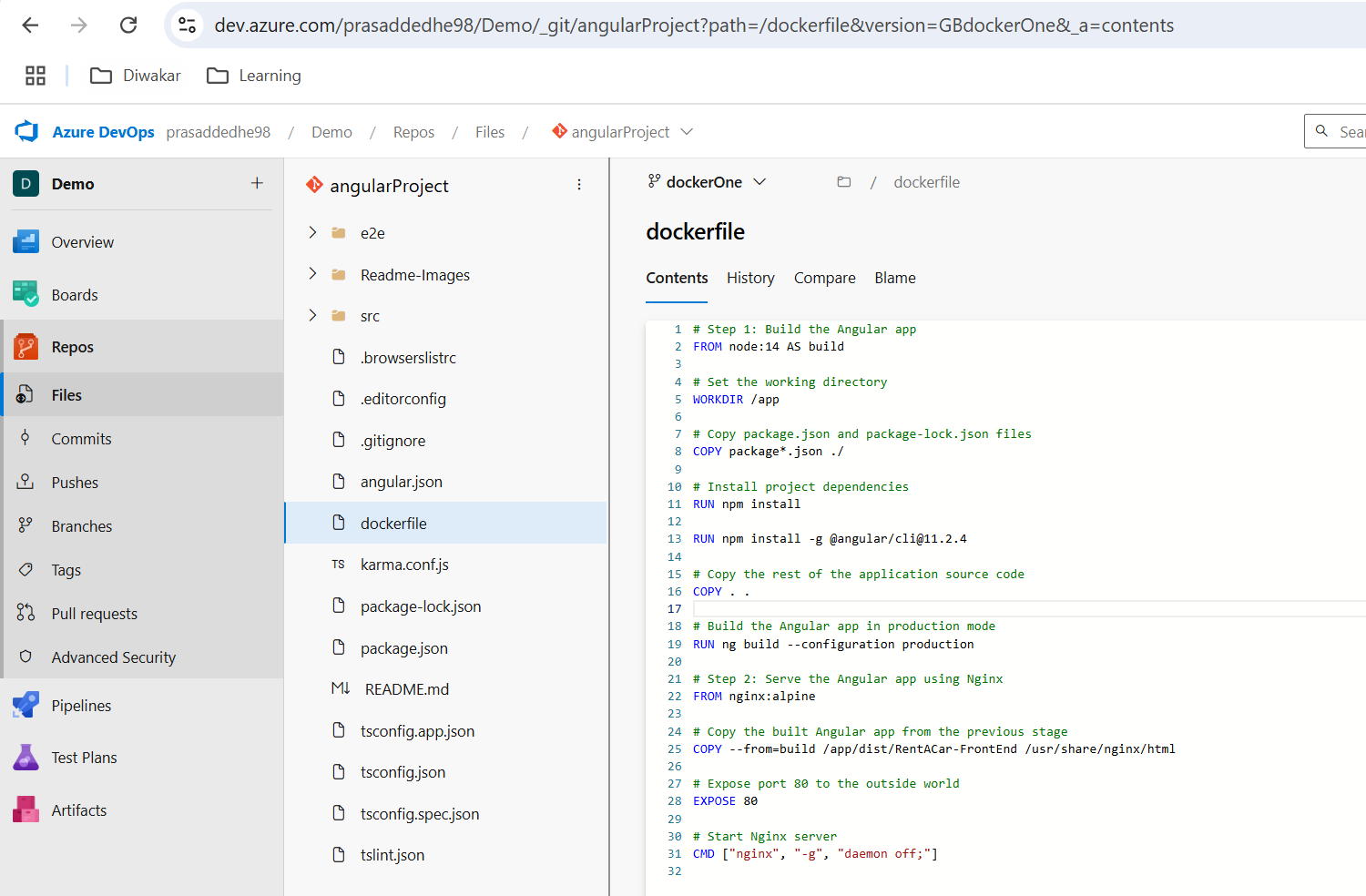
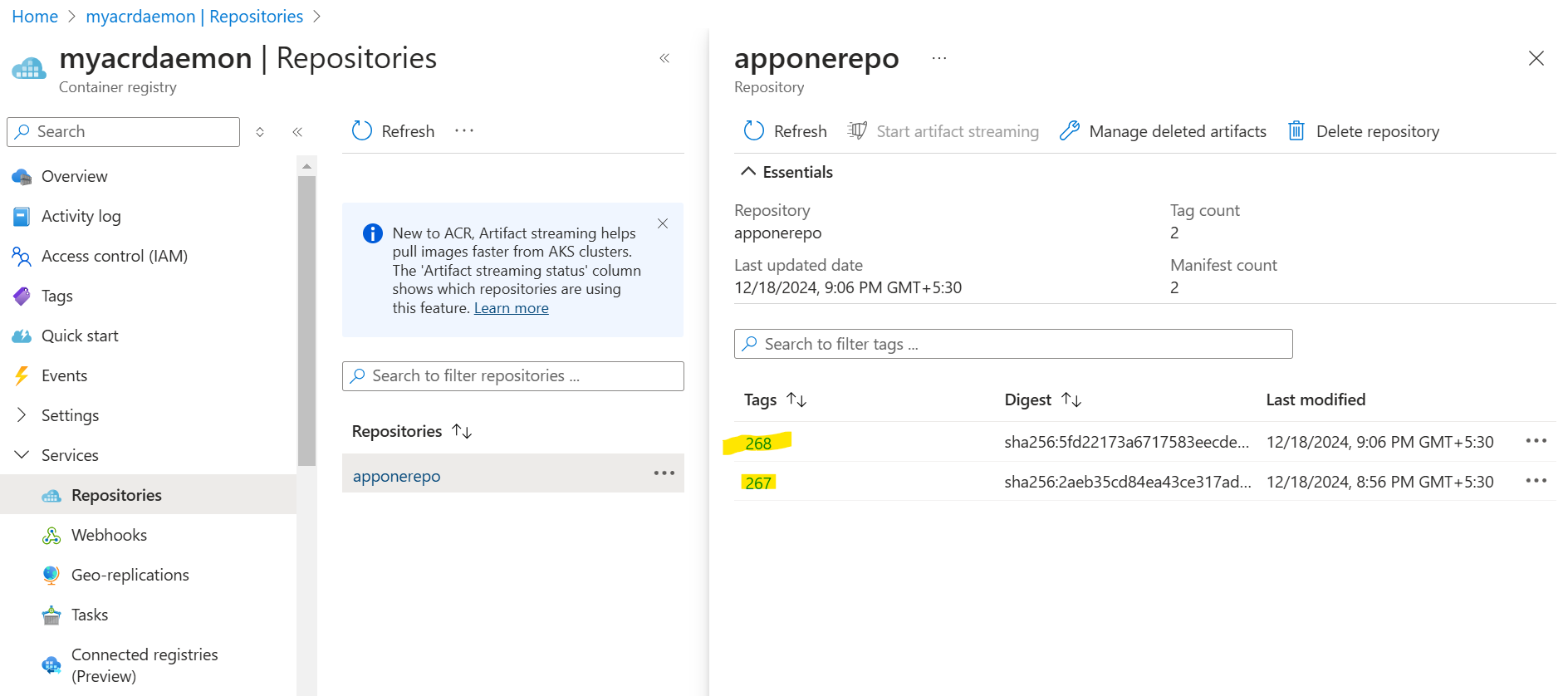
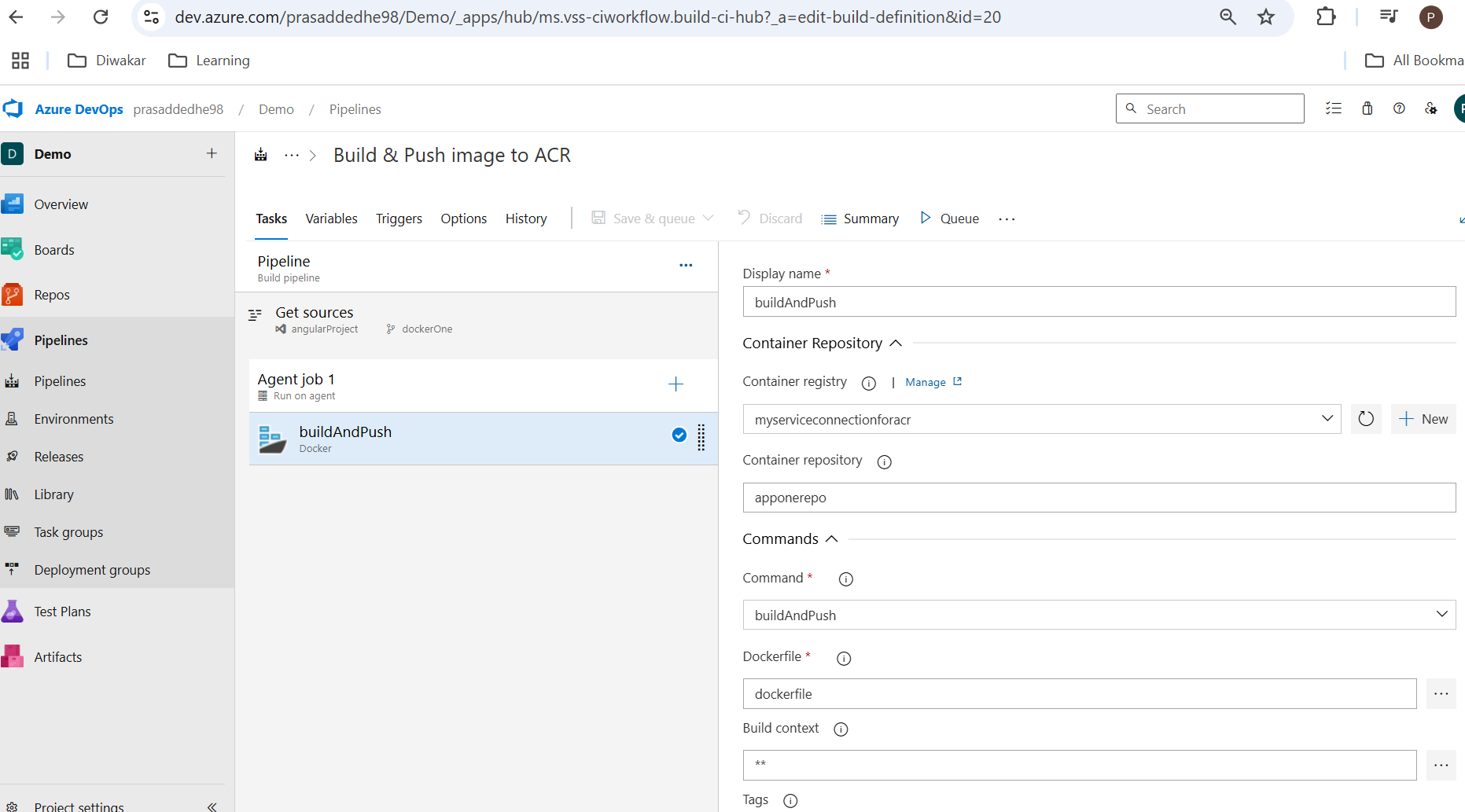
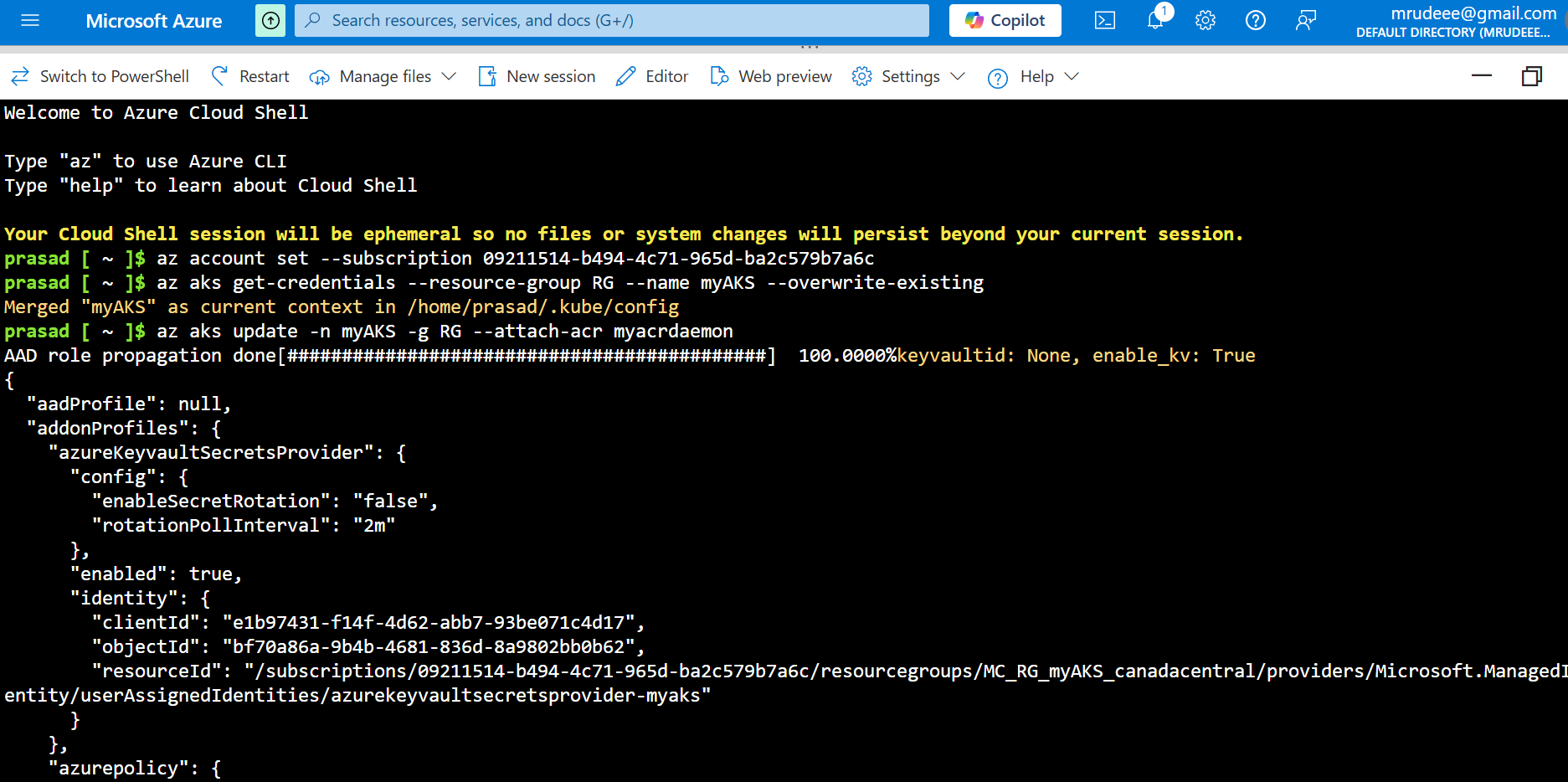
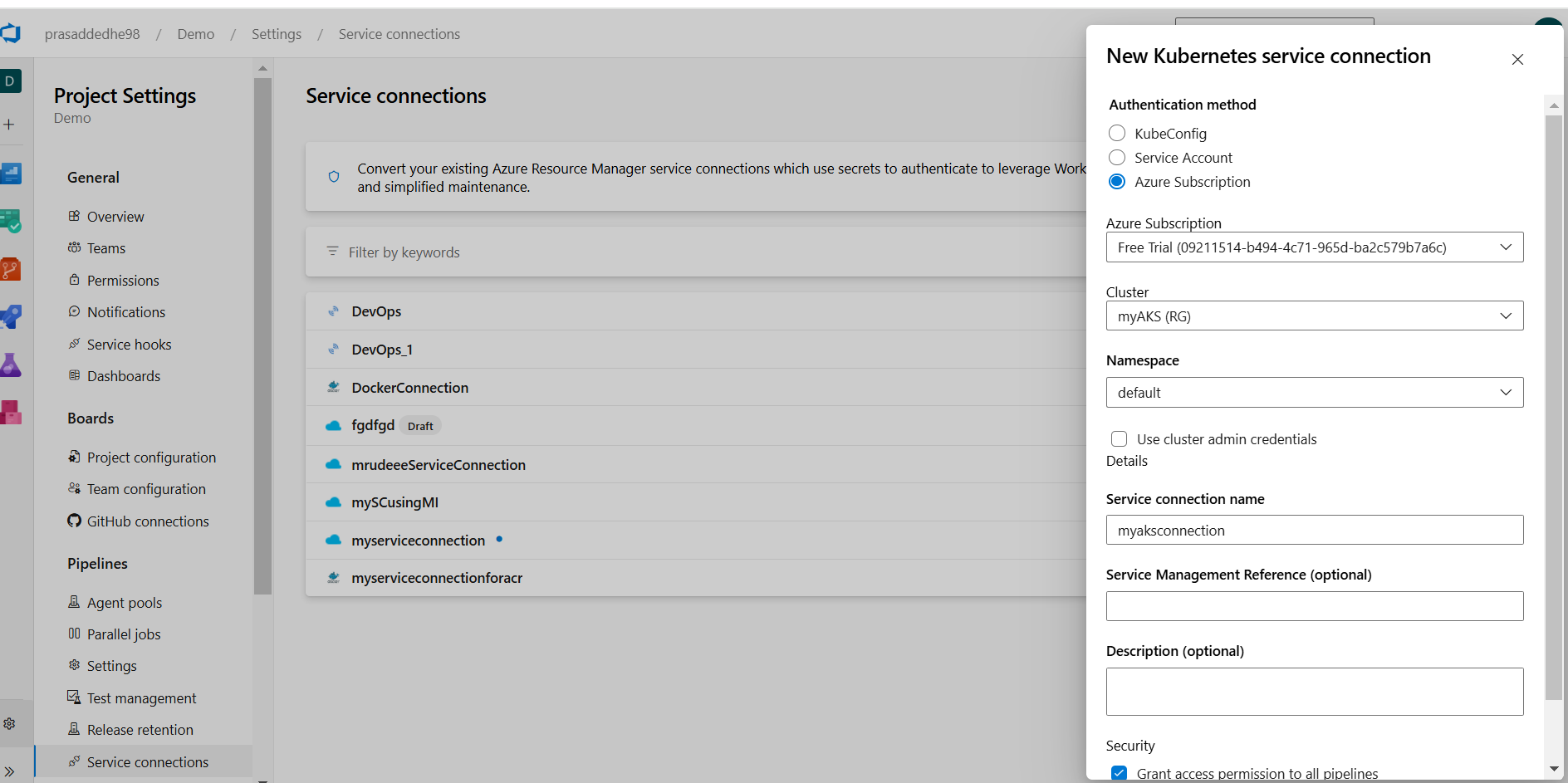
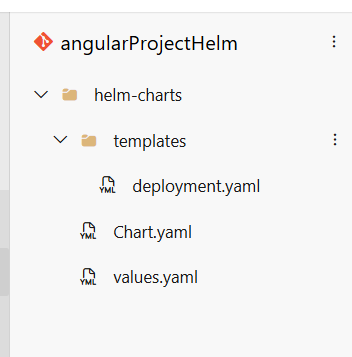
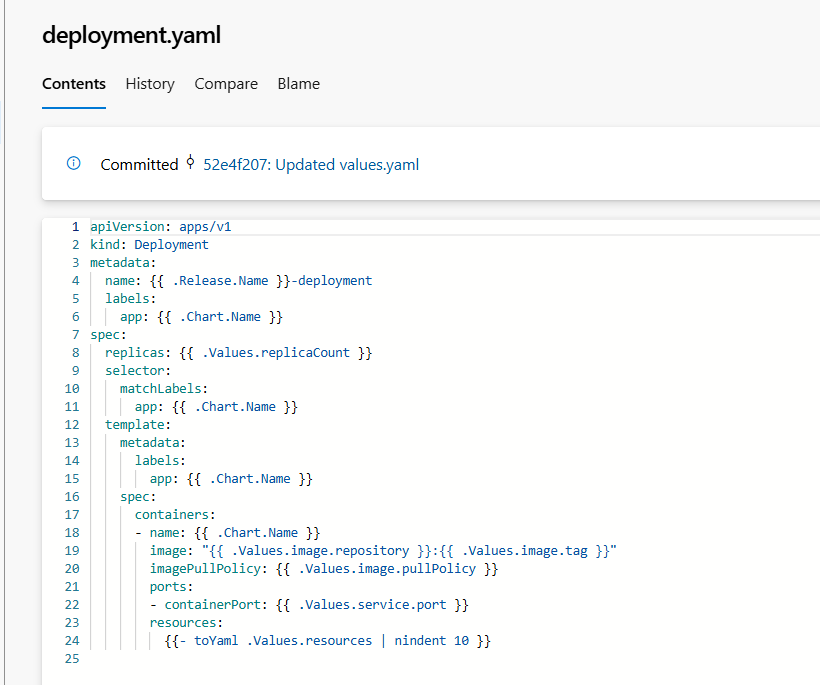
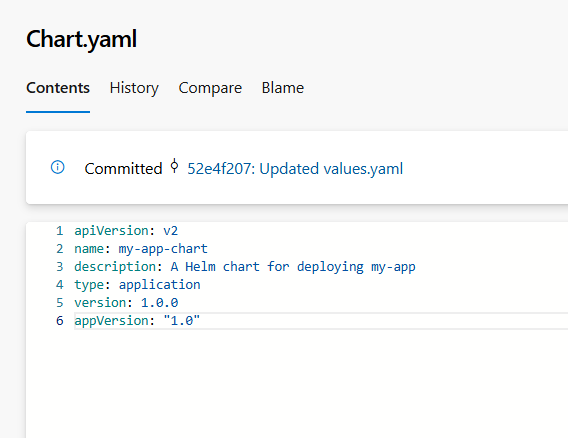
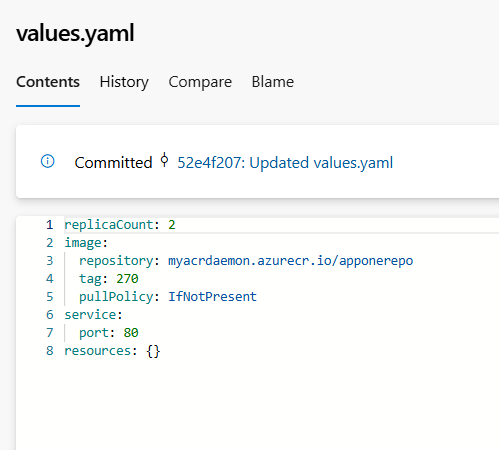
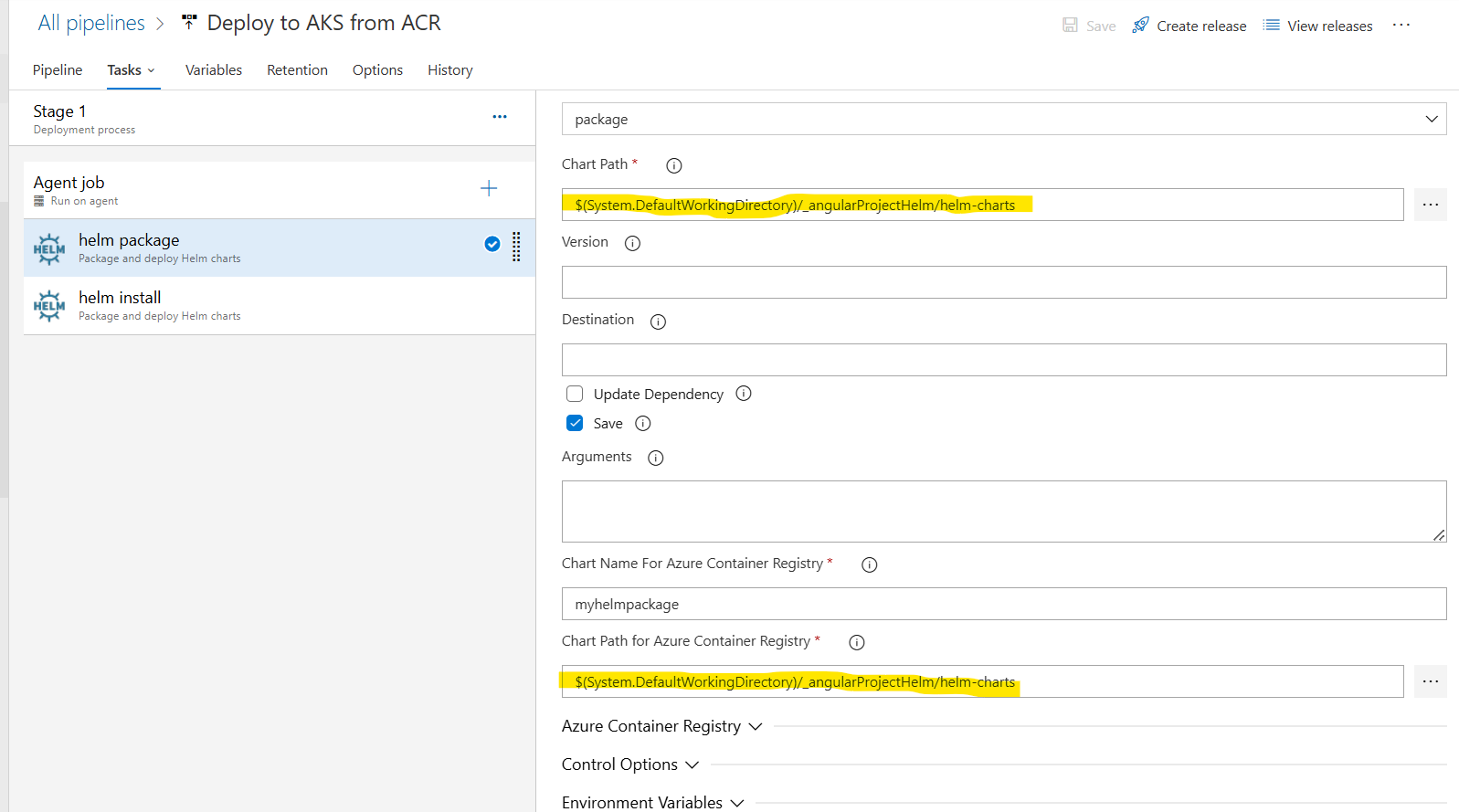
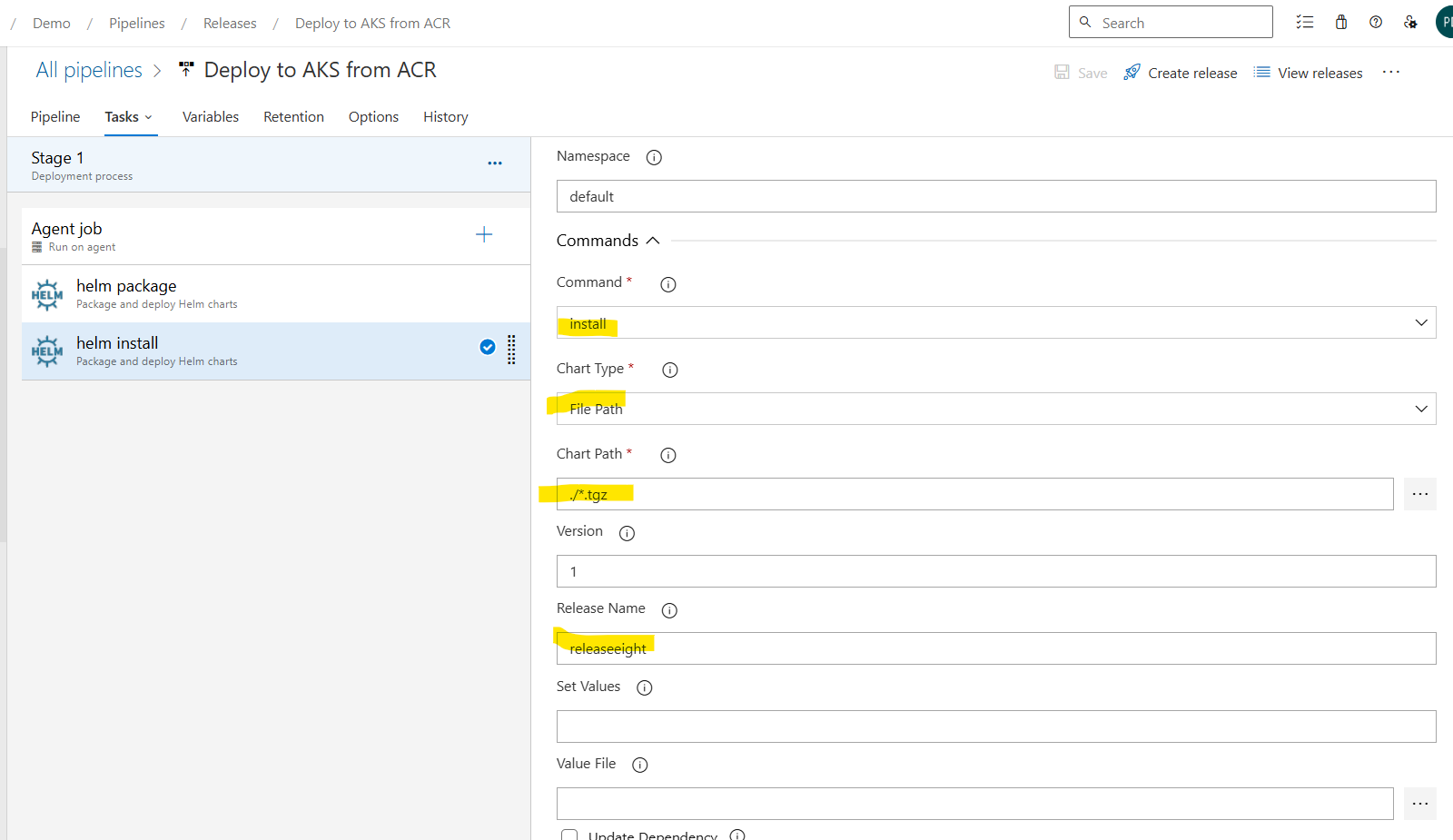
**Deploy on AKS through Azure DevOps**

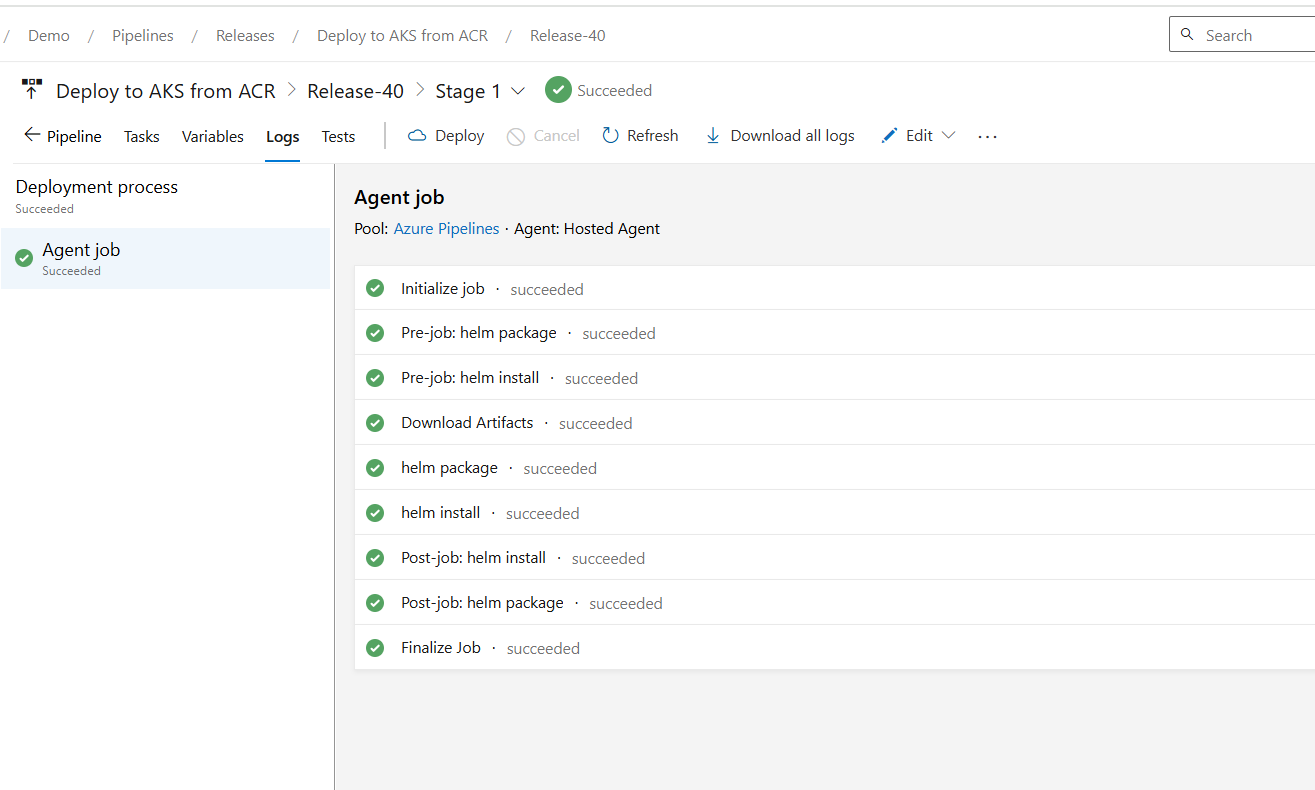
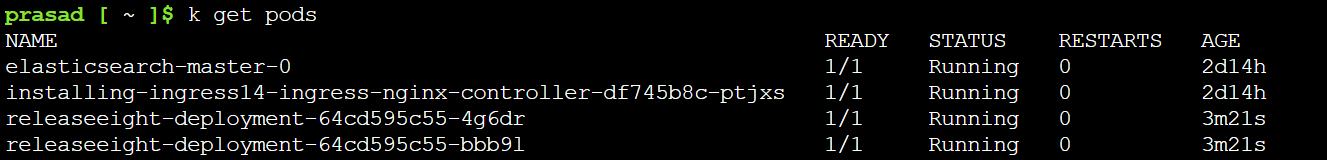
**Part – 1: Build and Push Image to ACR (Build pipeline)**

1. Import the repo in azure devops. 
2. Create a Dockerfile in it.
3. User DockerBuildandPush task. Create Docker service connection with acr information and then run the pipeline, it will create a repo and push the image with tag. 

**Part – 2: Deploy the image from ACR to AKS (release)**

1. Attach ACR to AKS, from shell,
   1. az aks update -n myAKS -g RG --attach-acr myacrdaemon. Also give acrPull role to AKS in acr through IAM.
2. Create “Kubernetes” service connection. 
3. Create a repo which has following structure.    
4. Then create a release pipeline. (a) These paths are very important. Package task is required because this task will convert your charts in .tgz (zip) package and helm can only deploy a zip package through pipeline. From Azure CLI you can directly deploy without packaging.

(b) The Install task or helm install command is the one which deploys the chart to AKS. These settings are very important, as I stumbled a lot of times before getting them right. Also, the release name, should be unique each time, else the release will fail. It’s advisable to use variable in it.

5. Once the pipeline is successful the pods will come up.  

6. There is no need of other tasks, like kubernetes or helm tool installer. These two tasks are enough. Further more I included lb service in chart and then the site was properly hosted on the ip. This took me almost 3 days, so don’t get dishearten if you are stuck somewhere. Keep it up! 