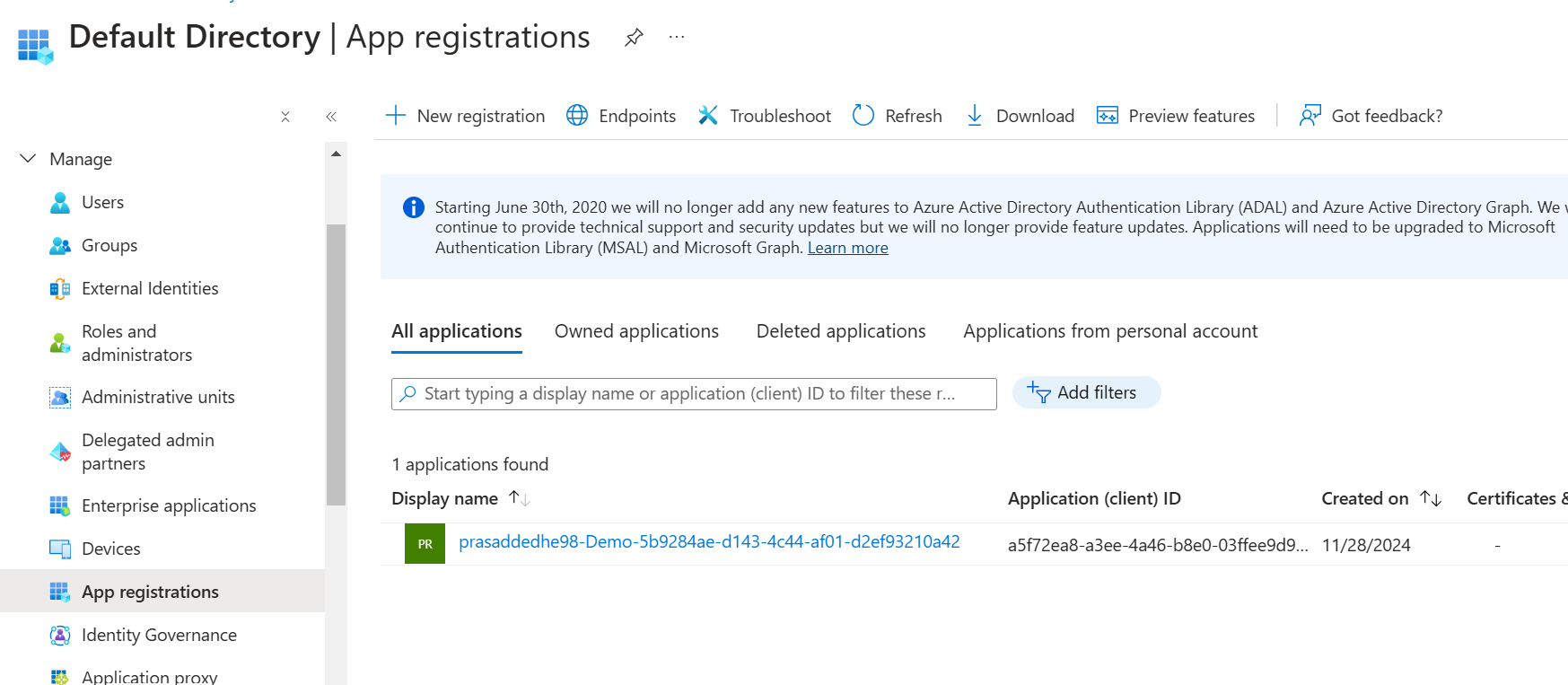
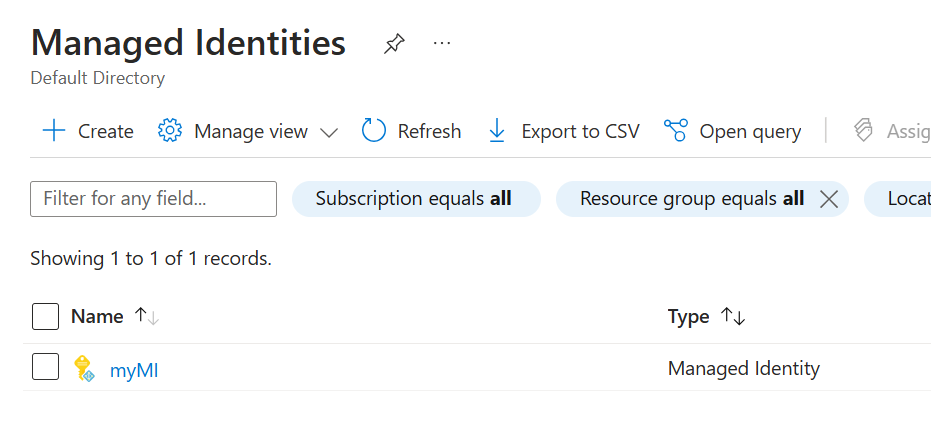
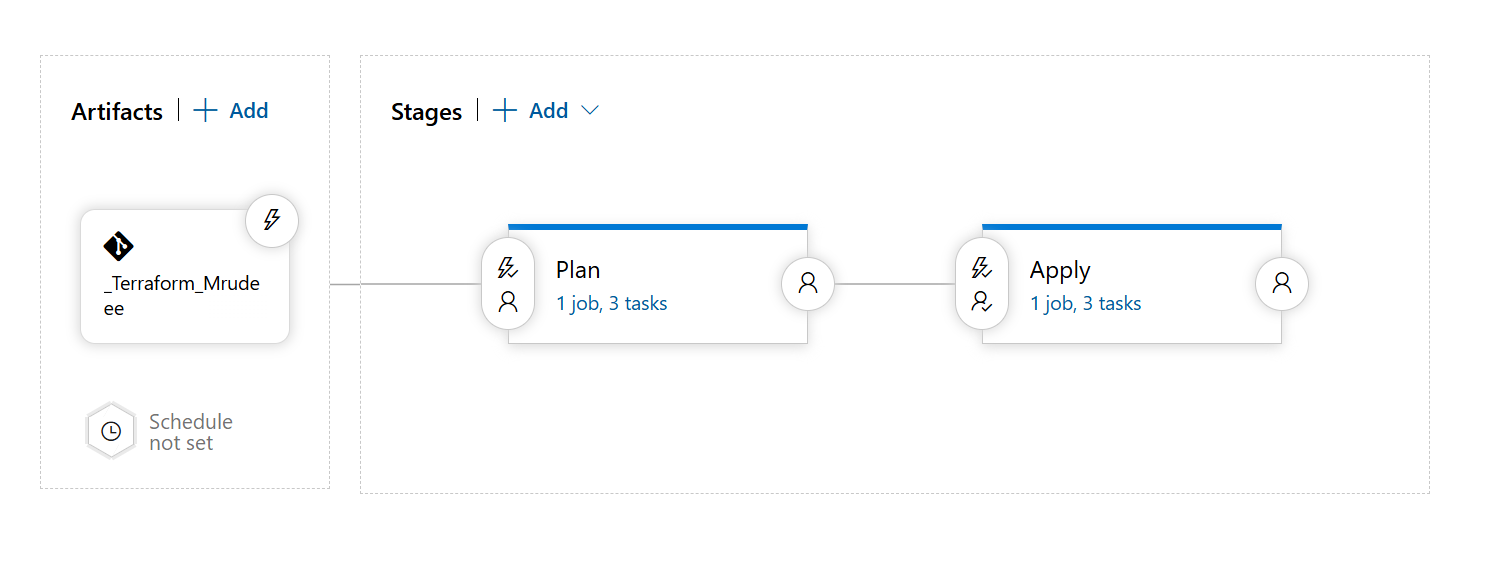
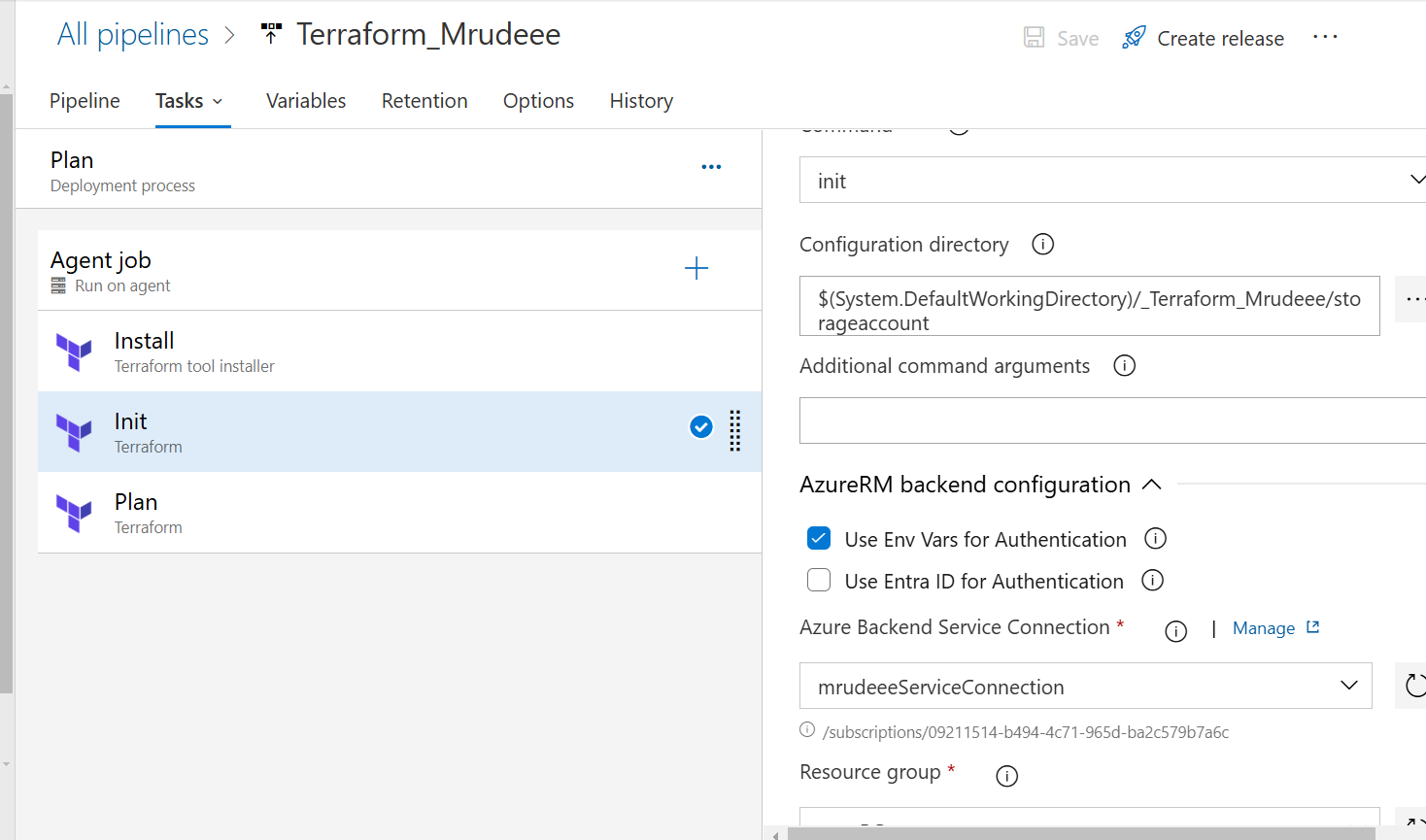
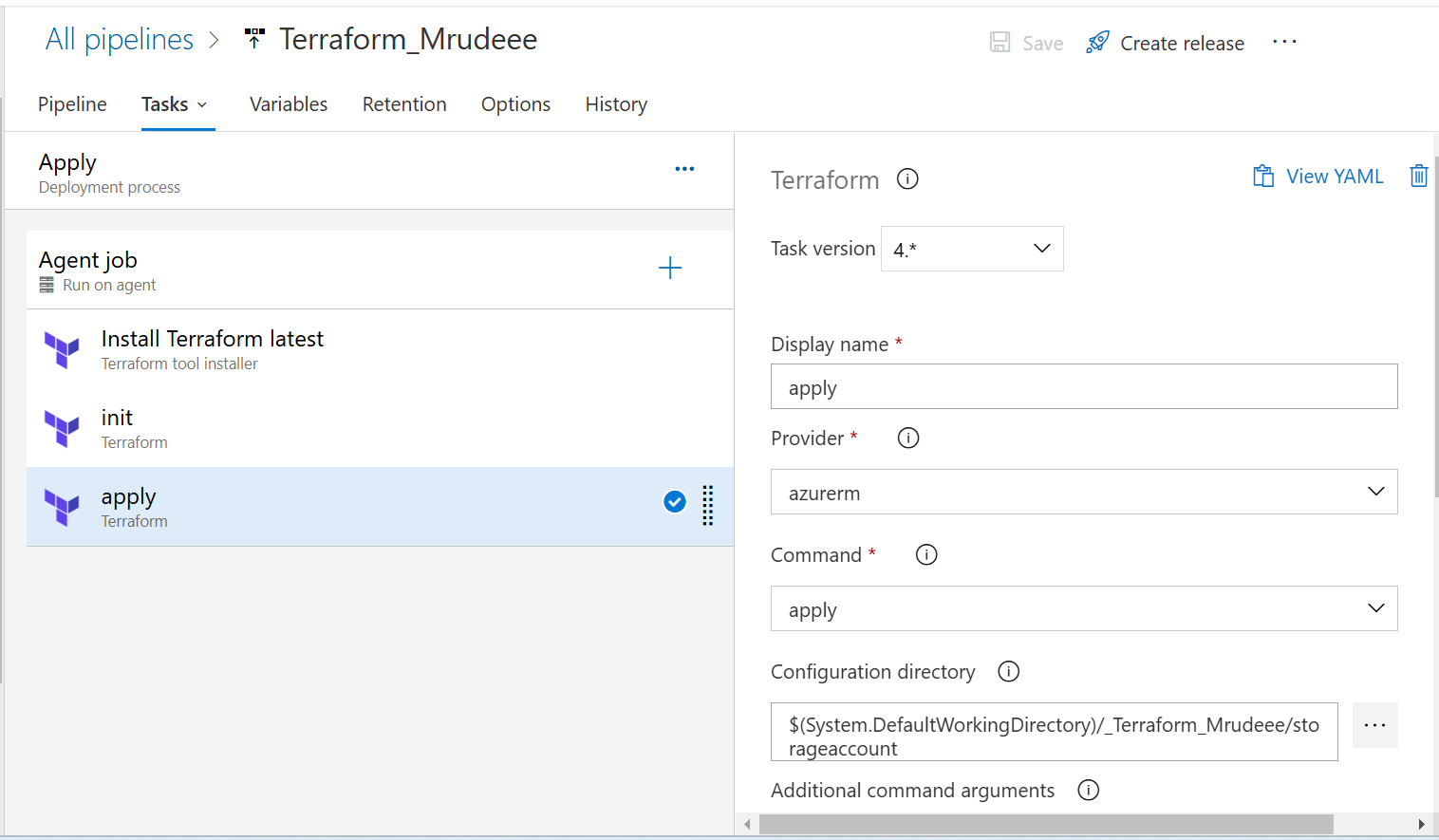
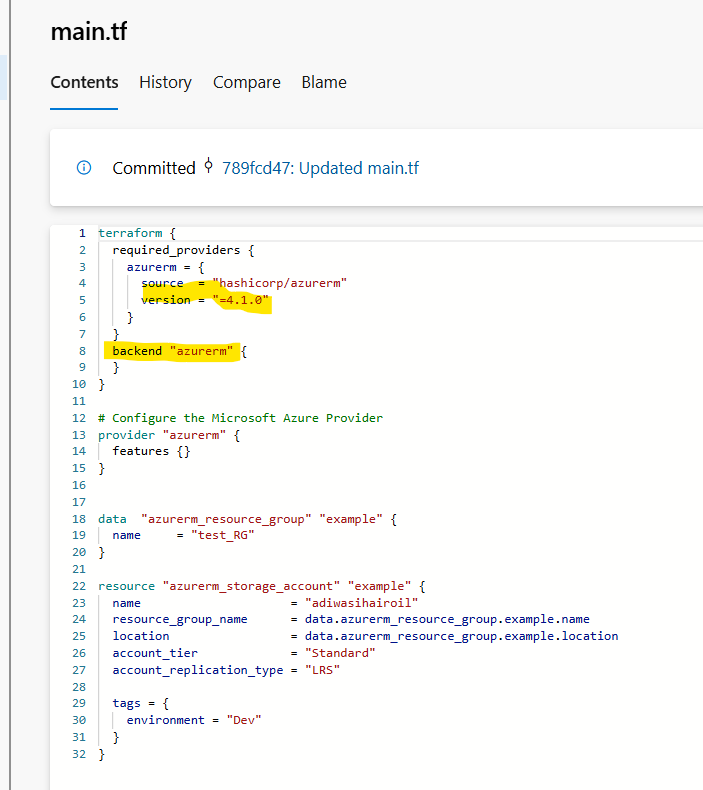
**Terraform using Azure DevOps**

1. Create a service principal in Azure DevOps 🡪 select app registration option 🡪 Azure DevOps will automatically create the app registration and give it the contributor access
2. If you choose managed identity for service connection, then you need to create a managed identity in Azure Portal. The only difference between managed identity and app registration is that manage identity doesn’t use client id and secret, it obtains token every time it needs to login .
3. Create a repo where you create a folder, in which you add main.tf and optionally variable.tf files. Then create a pipeline, with two Jobs one for plan and other for apply. 
4. Plan Job:
   1. In Init task (version 4), Configuration Directory select the folder
   2. In Init task, tick Use Env Vars for Authentication
   3. In Init task, key format 🡪 stfilecontainer/tf/logicapp/terraform.tfstate
   4. In plan task, , Configuration Directory select the folder



1. Apply Job:
   1. Follow the same instructions in init task
   2. Add approval before the apply task



1. The main.tf looks like this. Remember the version in Terraform in very important. My release failed 10 times because older version mentioned. If the release fails, try to change it. Also, do not forget to mention the backend block, else no statefile would be generated. 
2. The successful results. 