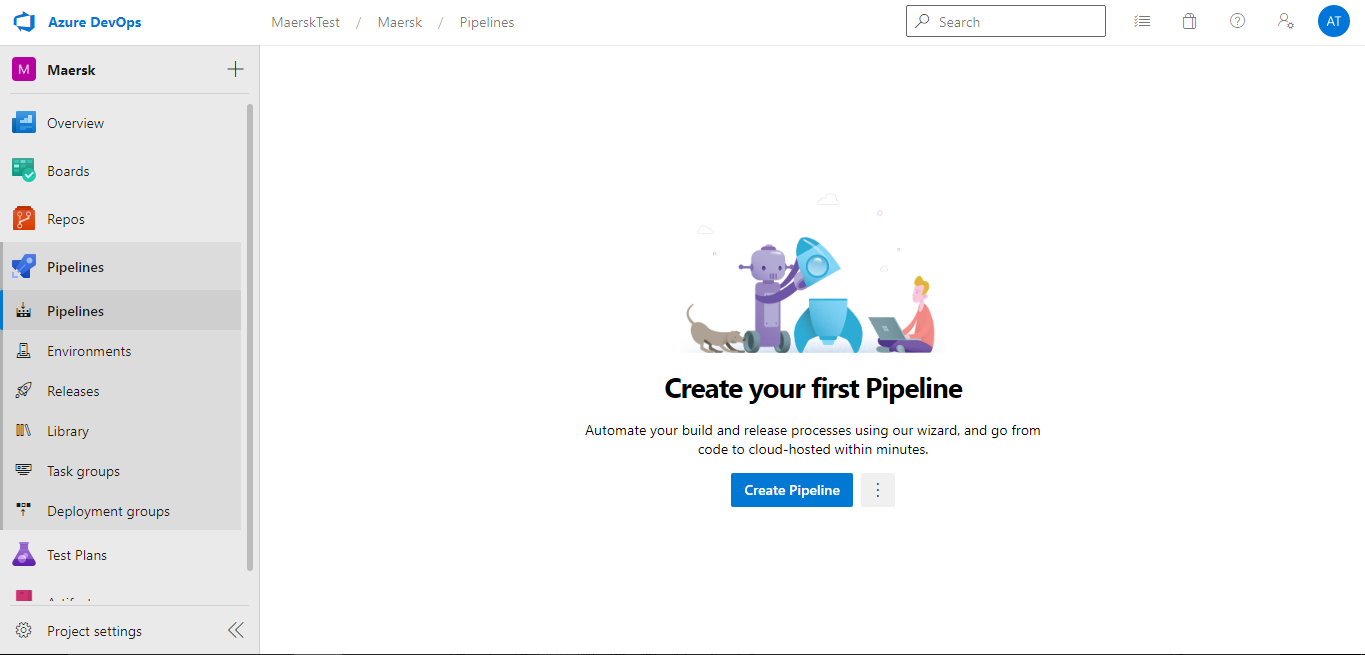
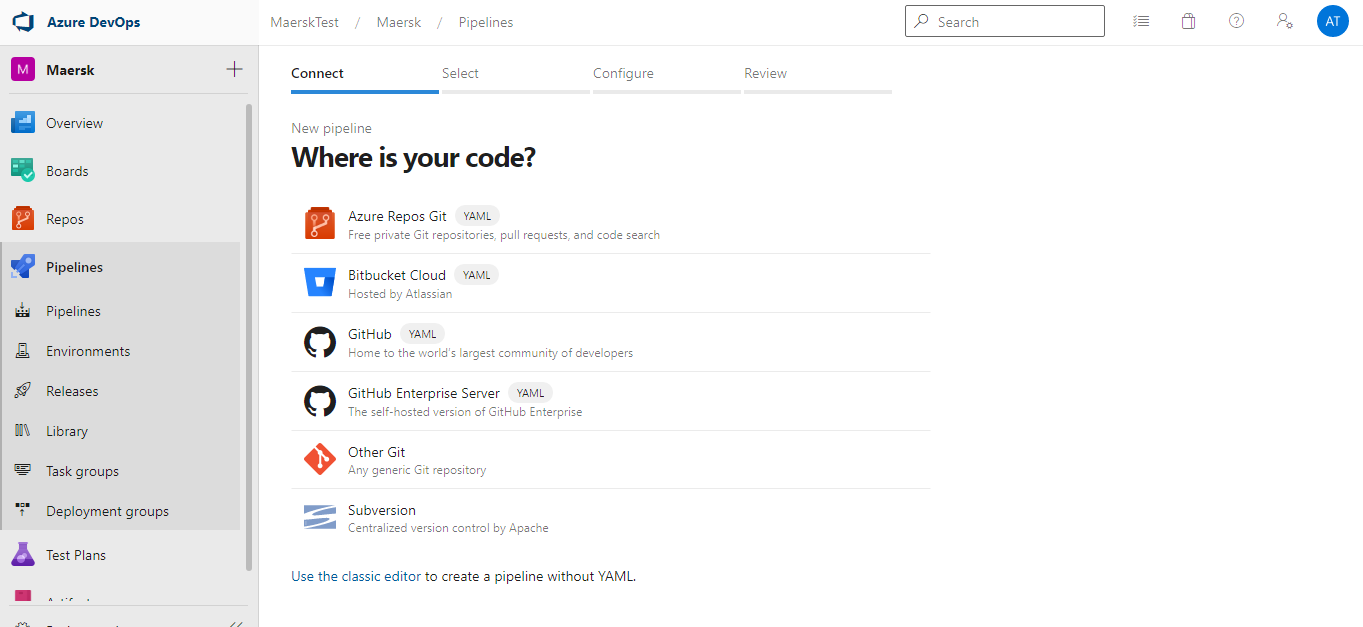
**Procedure to create a new Build Pipeline for AKS Deployment**

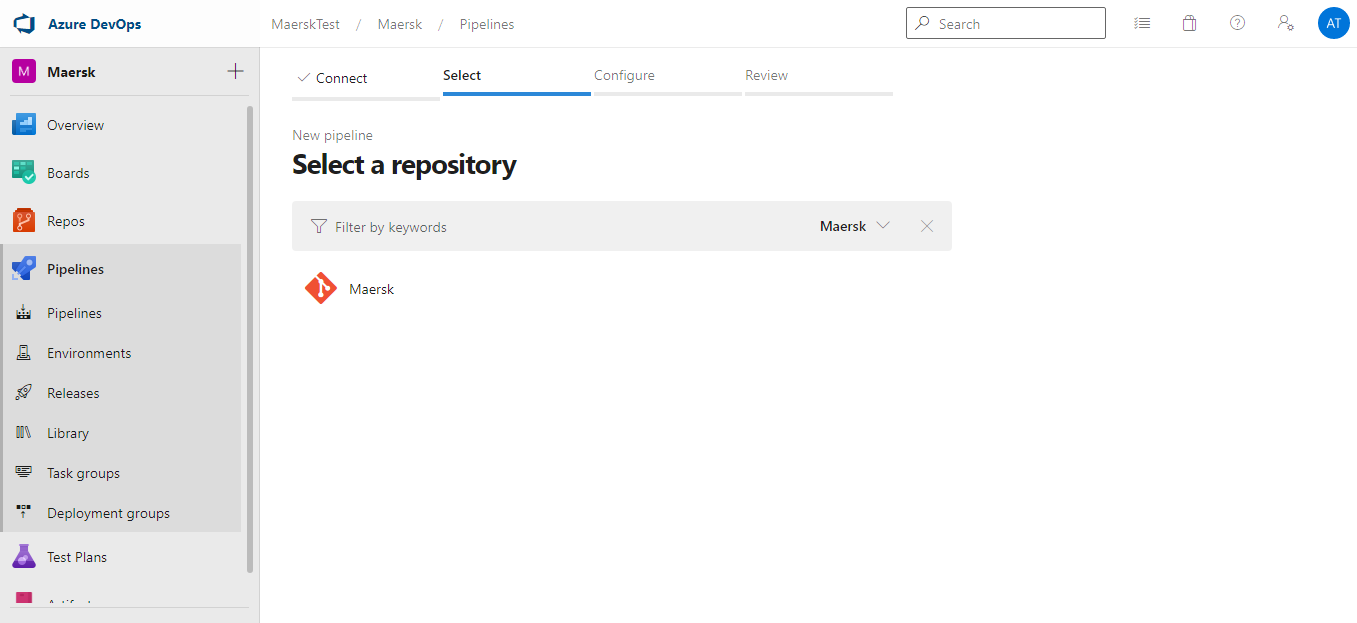
* Select **Pipelines** tab.



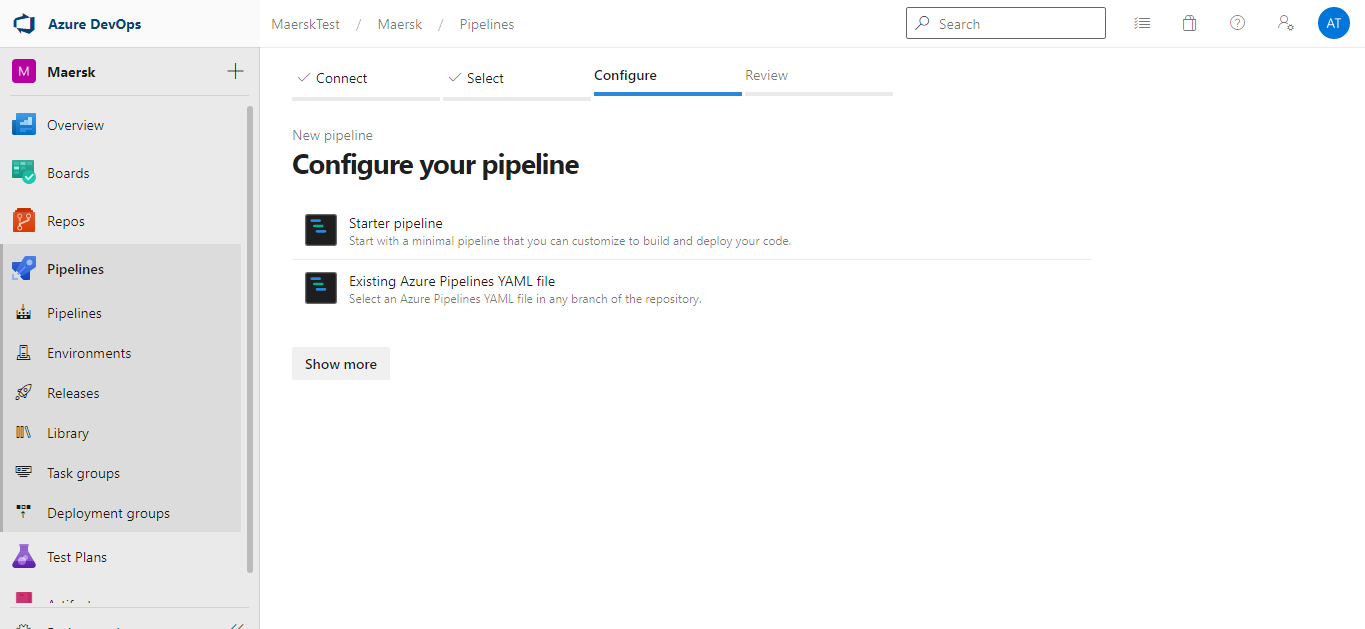
* Under Builds, click on **New Pipeline**
* Select what kind of pipeline need to be developed. In our case we are developing Yaml based Pipeline.



* Select the repository and branch from the available sources

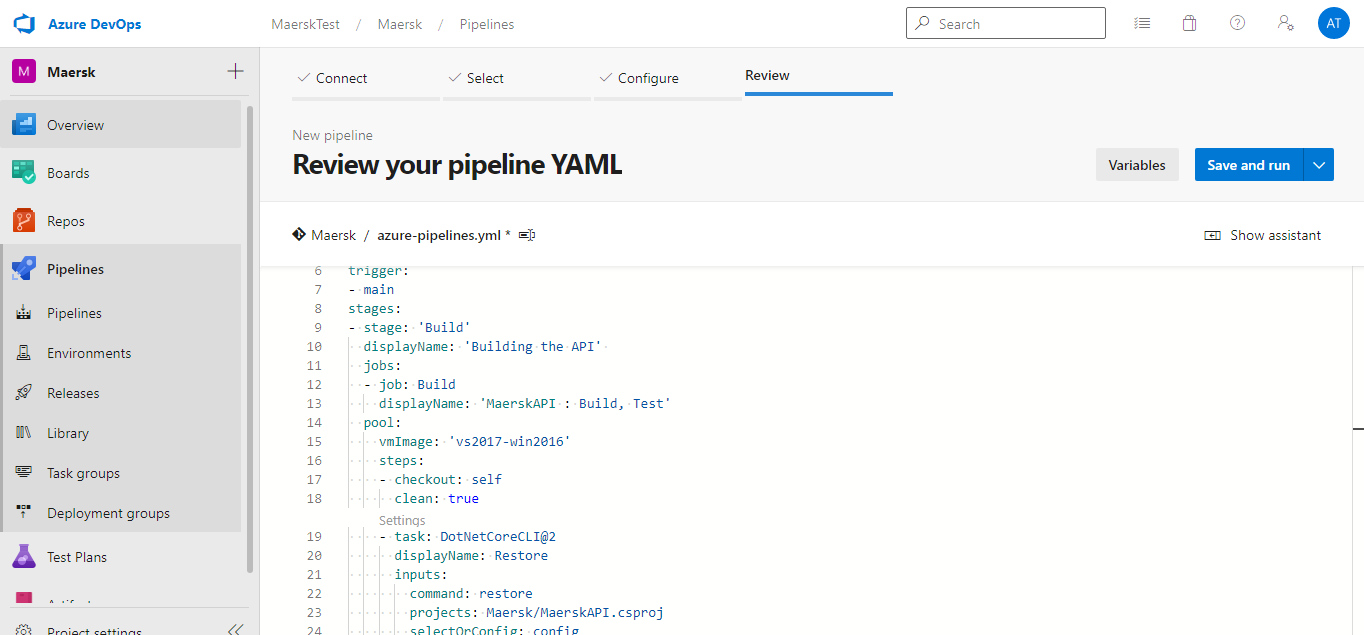


* Select a suitable template for the project

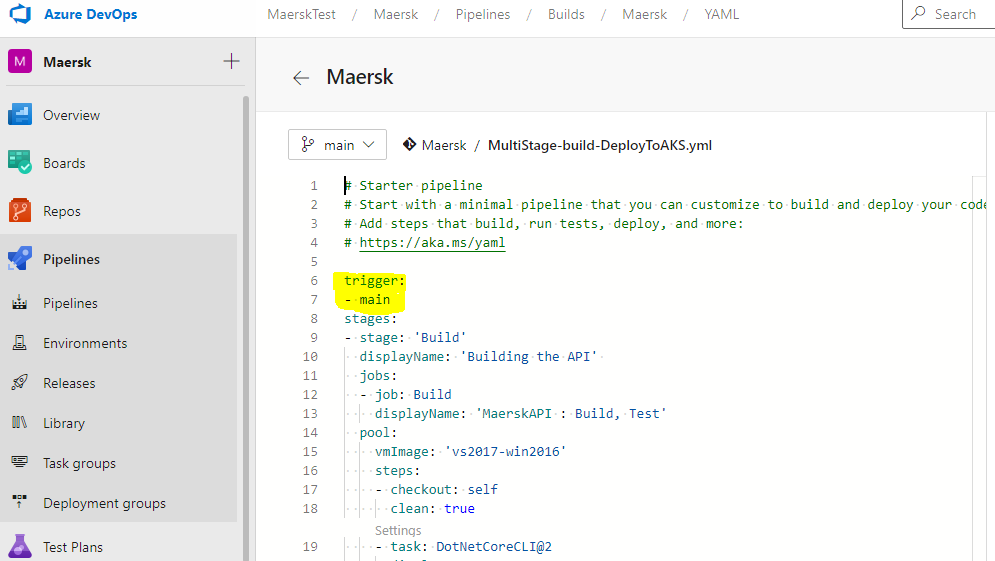


**Stage-1:** Building the Solution check for any error

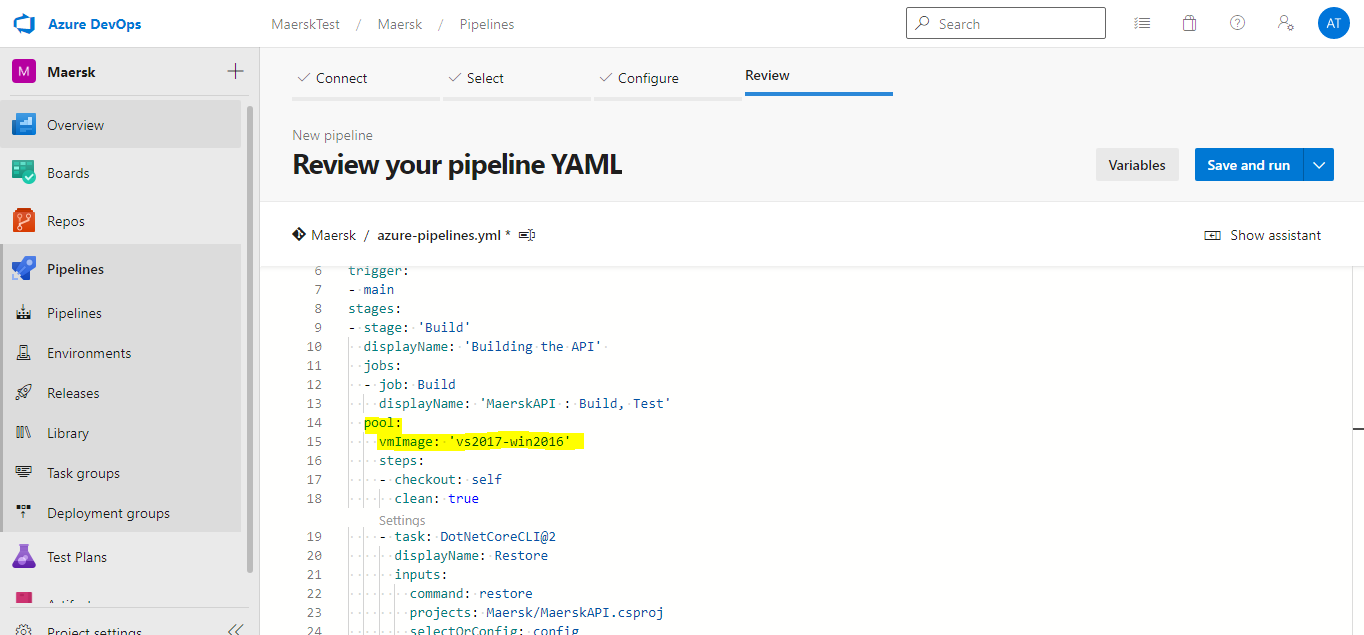
* Build Yaml Pipeline is created and modify according to the requirement



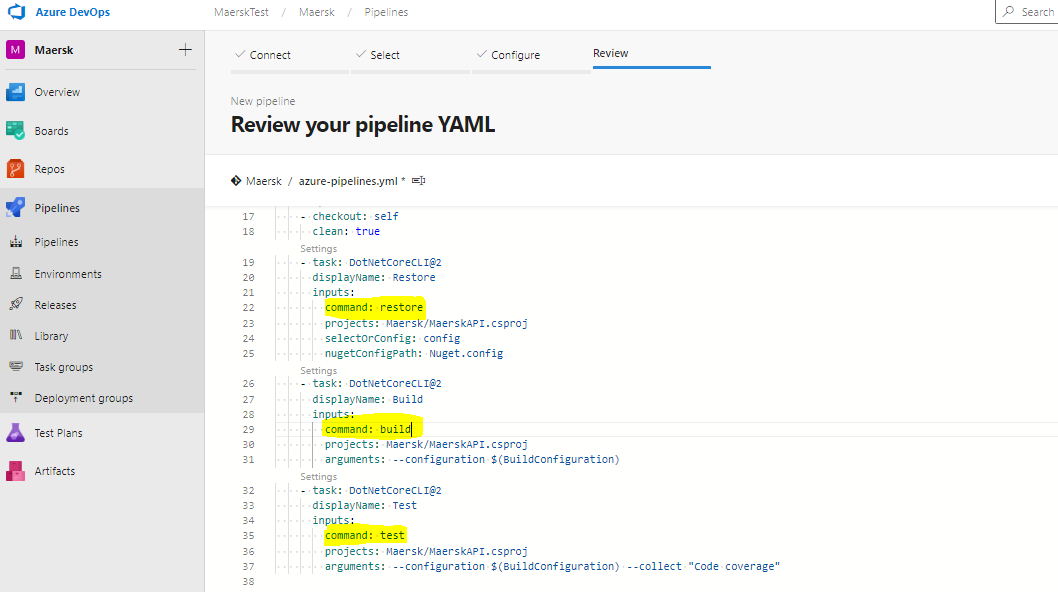
* As we are implementing the CI Pipeline. We have enable CI trigger in Yaml.



* Under the Build Yaml Pipeline tab, Provide the Build agent from the agent pool



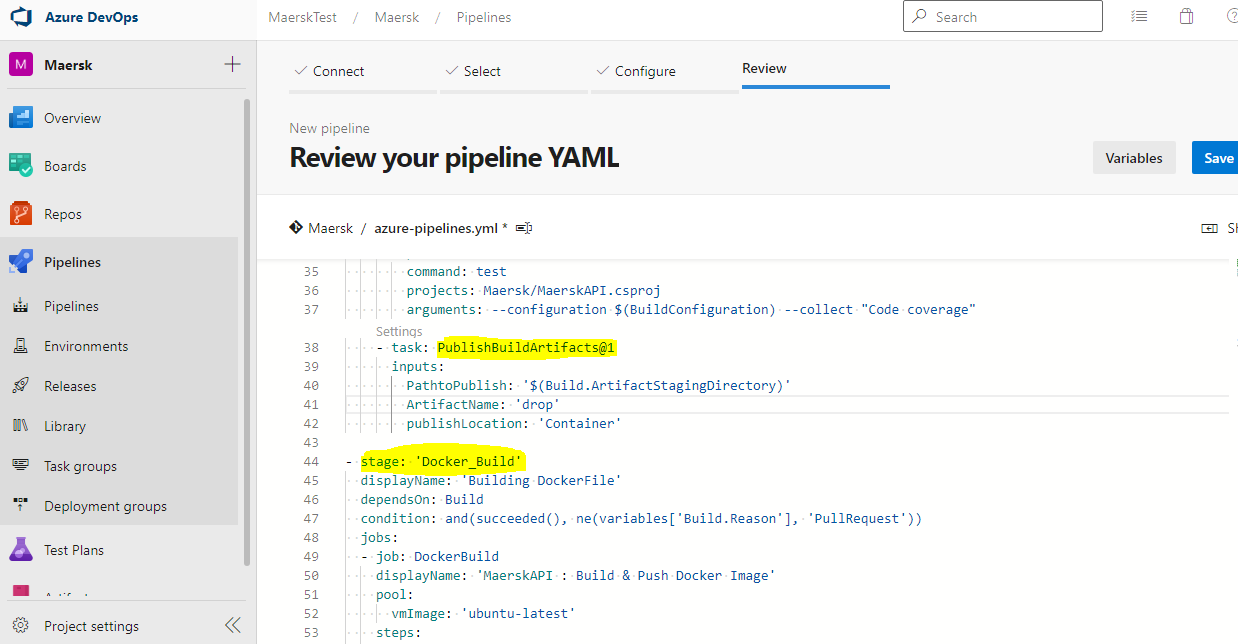
* Specify the path to the **Projects to restore & build** and **test**



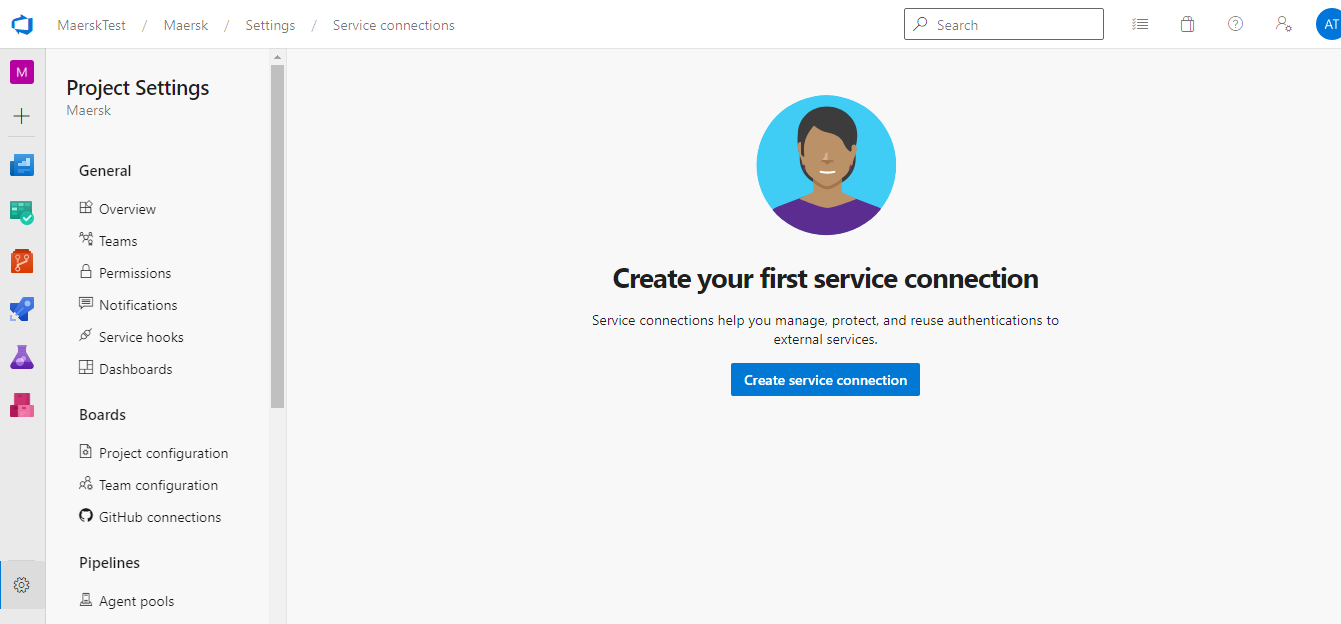
* Here, we have add **Publish** task to publish to Build artifacts

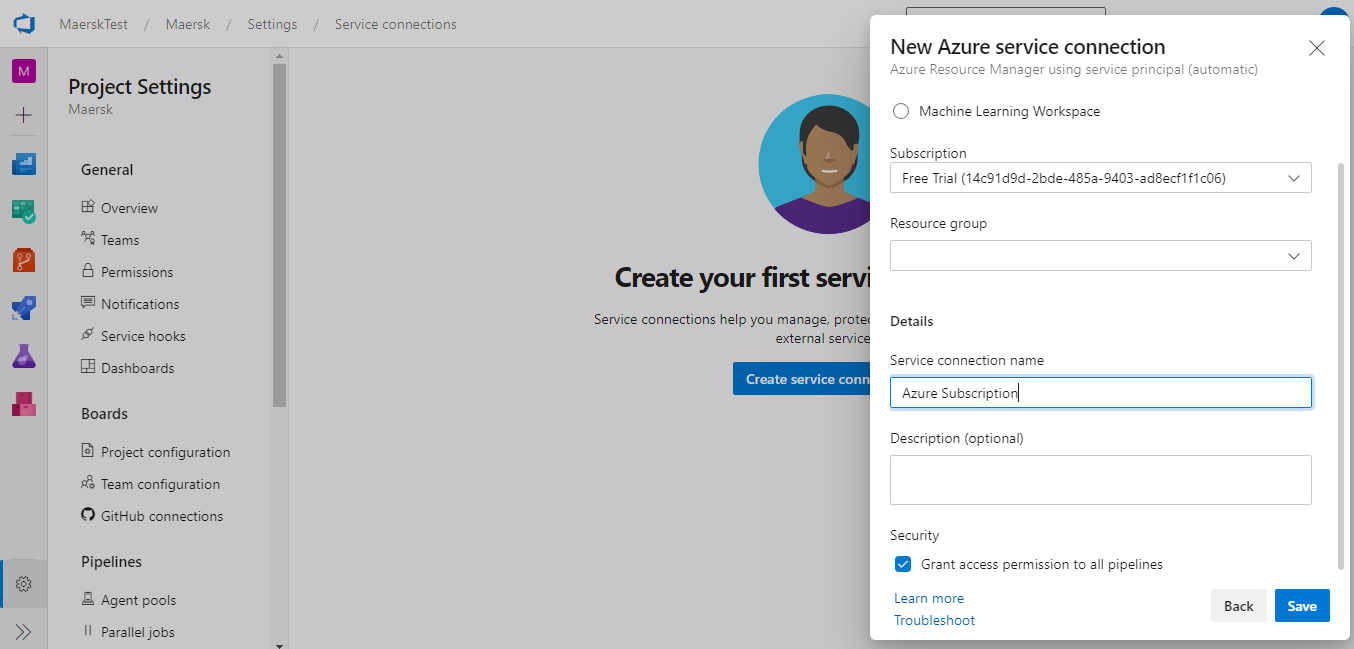
**Stage-2:** Building and pushing the Docker image

* Add the tasks related to **Docker** Container in Next stage.

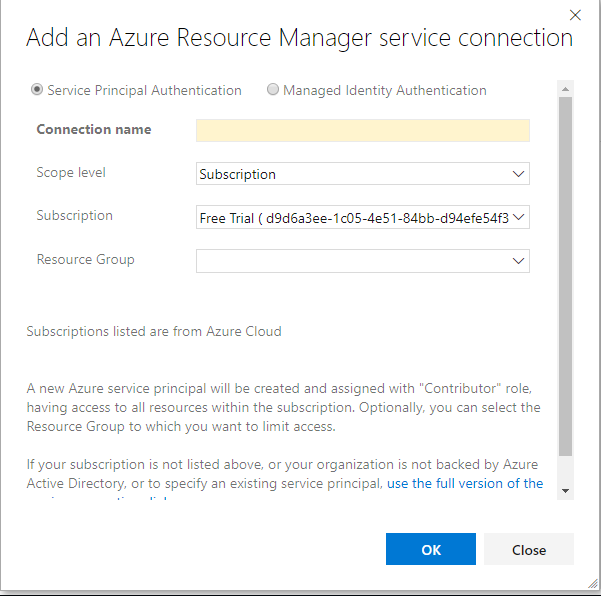


* Go to Project Settings ->Service Connection -> New Service Connection -> Select from the Azure Subscription dropdown.

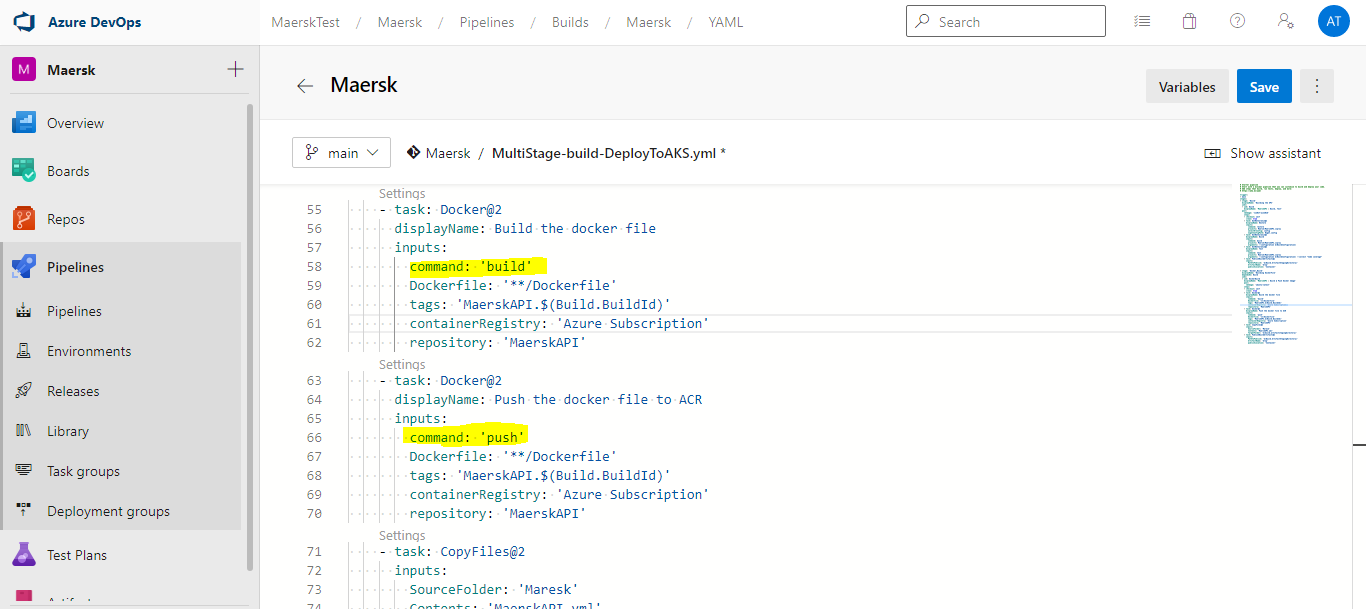




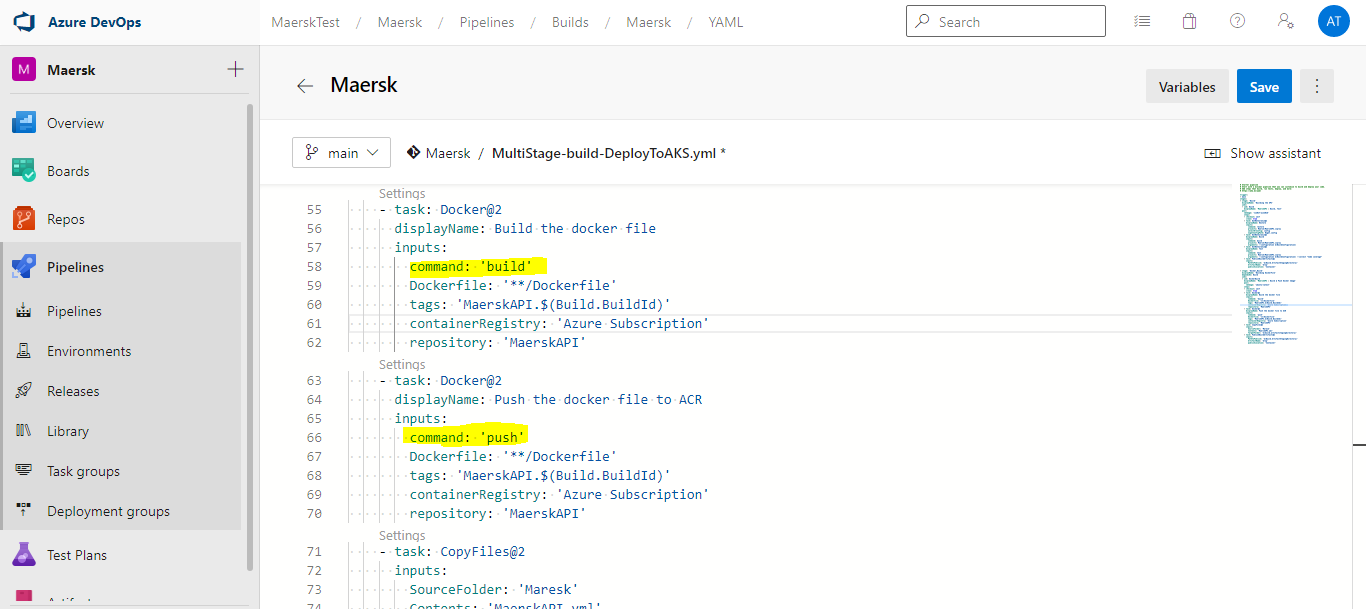
* if you have already created a service connection or click on **Manage** to provide the Azure Subscription details
* Select Azure Resource Manager from the dropdown list and provide the following details and save.



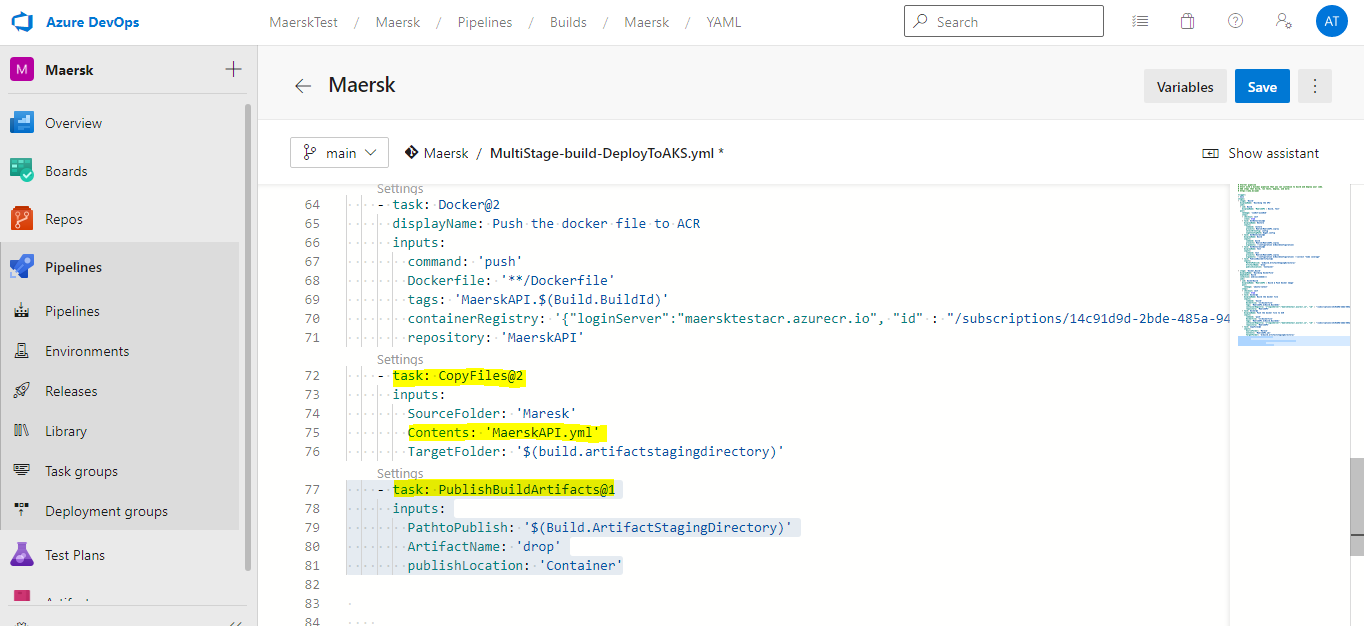
* Add Docker task twice. One for **Build Docker Image** and Other for **Push Docker Image**



* Provide the **Container registry and Container repository** details in Yaml. Provide the Dockerfile details from the source repo. Use **build & Push** command for building & Pushing docker image



* Provide the Image name as uniquely Identified with in ACR repository, also check **include latest tag**
* Copy the MaerskAPI.yml and publish the artifacts, which we will be using for AKS deployments in Release Pipeline.



* Verify Image in the ACR.

**Prerequisite Before deploying to AKS for the first time.**

* As we created the AKS for the first time and Our ACR is a Private registry
* We have to run the below script in Azure Cloud shell for the first time. Which gives the reader permission for AKS.

AKS\_RESOURCE\_GROUP=MaerskRG

AKS\_CLUSTER\_NAME=MaerskAKS

ACR\_RESOURCE\_GROUP=MaerskRG

ACR\_NAME=MaerskTestACR

# Get the id of the service principal configured for AKS

CLIENT\_ID=$(az aks show --resource-group $AKS\_RESOURCE\_GROUP --name $AKS\_CLUSTER\_NAME --query "servicePrincipalProfile.clientId" --output tsv)

# Get the ACR registry resource id

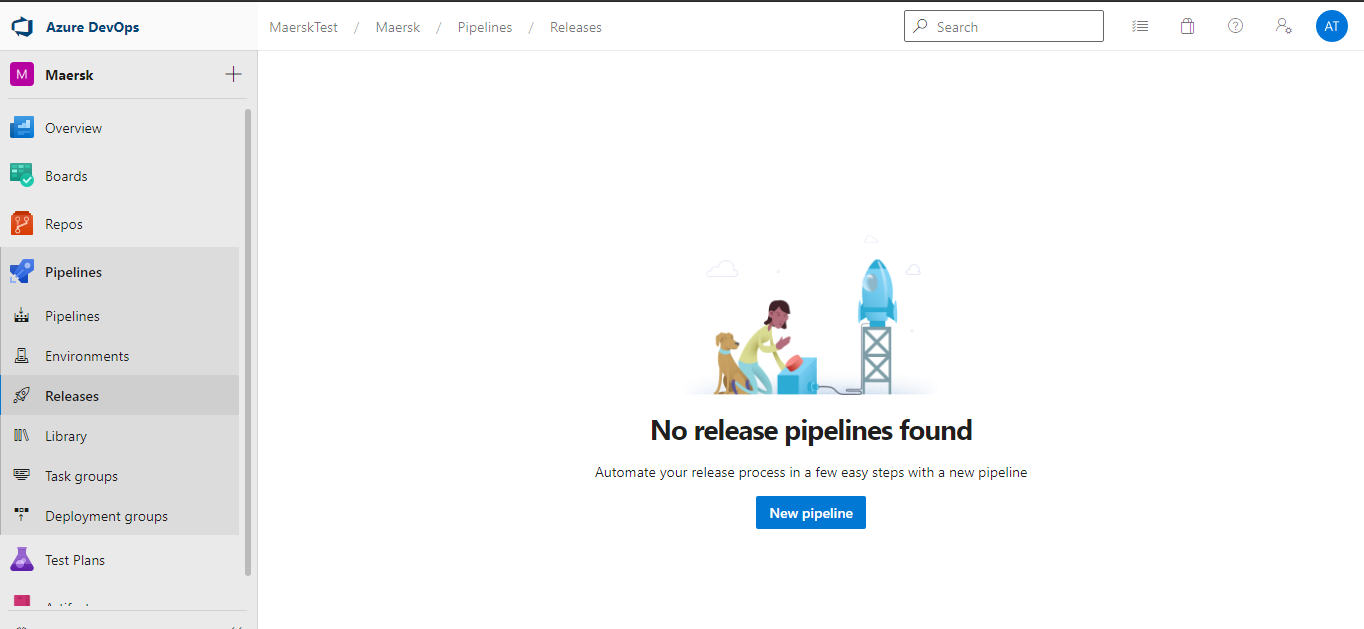
ACR\_ID=$(az acr show --name $ACR\_NAME --resource-group $ACR\_RESOURCE\_GROUP --query "id" --output tsv)

# Create role assignment

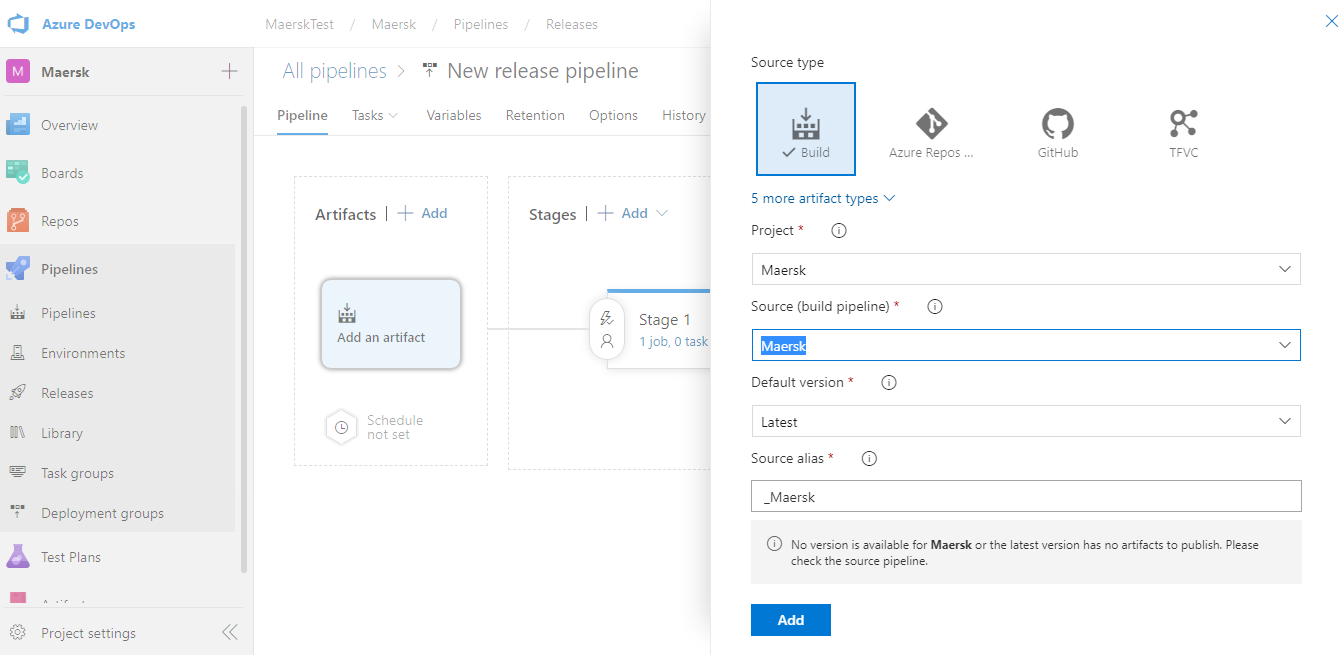
az role assignment create --assignee $CLIENT\_ID --role Reader --scope $ACR\_ID

**Procedure to create a new Release Pipeline for AKS Deployment**

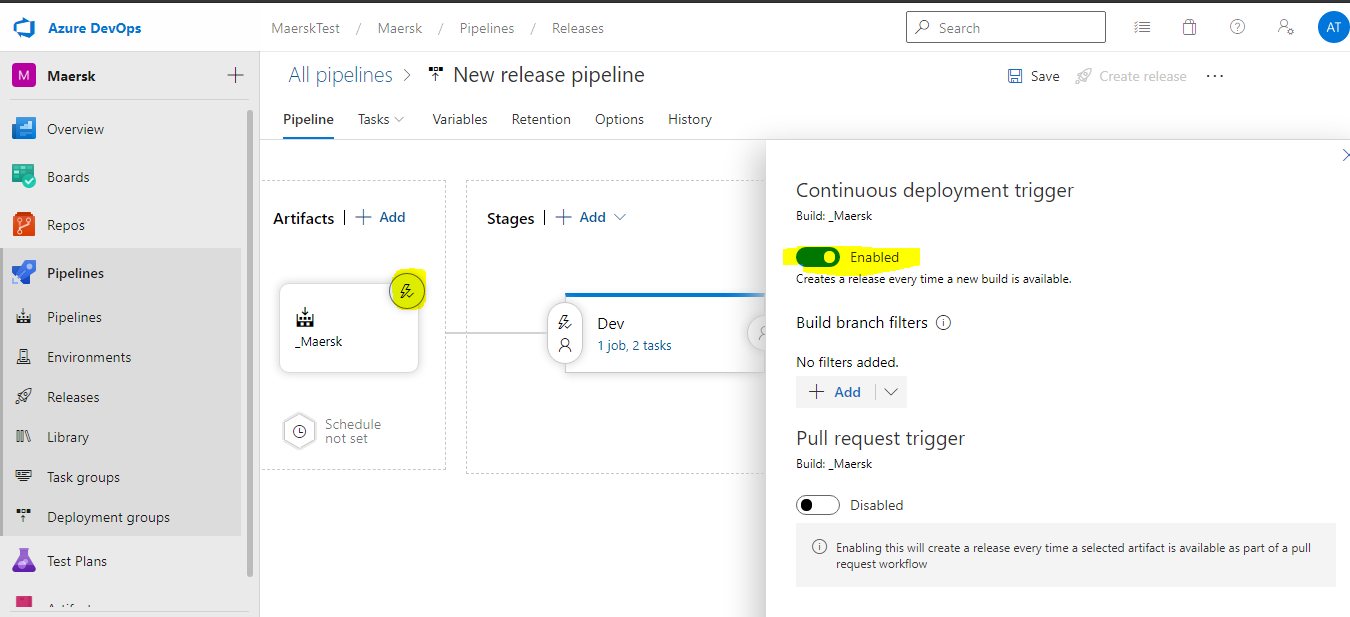
* Select **Release** **Pipelines** tab.



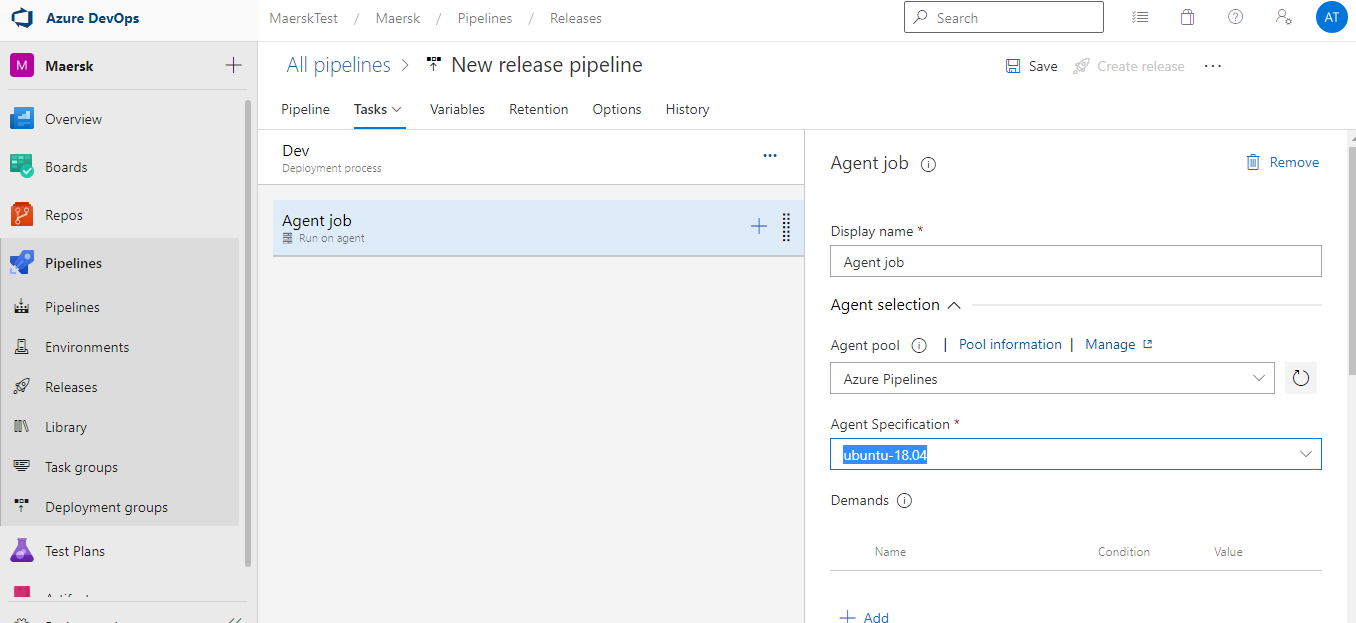
* Under Release, click on **New Pipeline**
* Select **Artifacts**



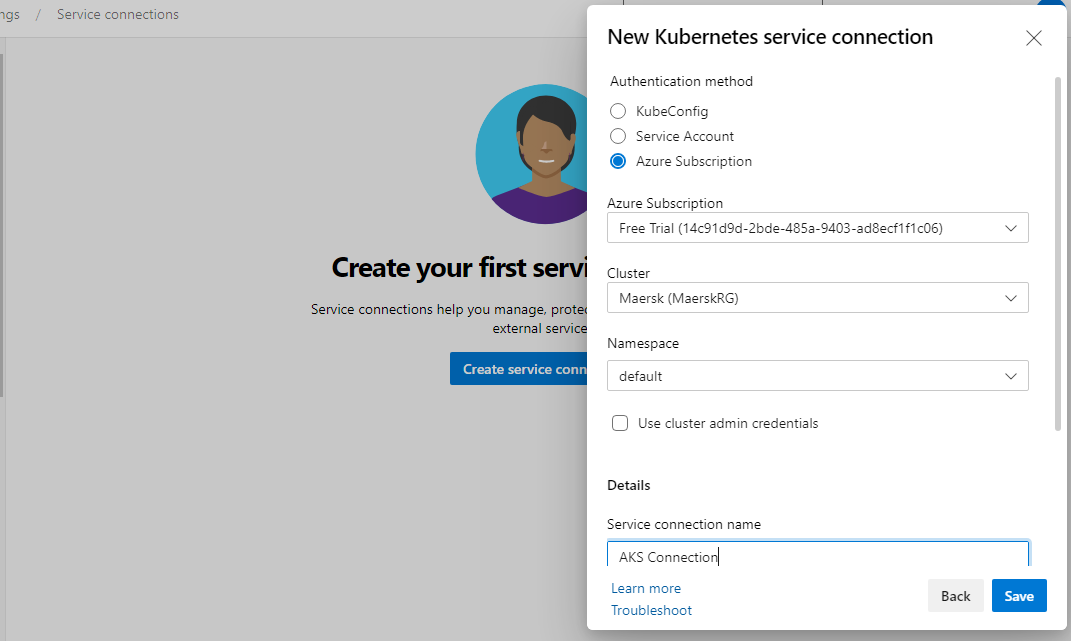
* As we are implementing the CD Pipeline. We have enable CD in the artifacts.



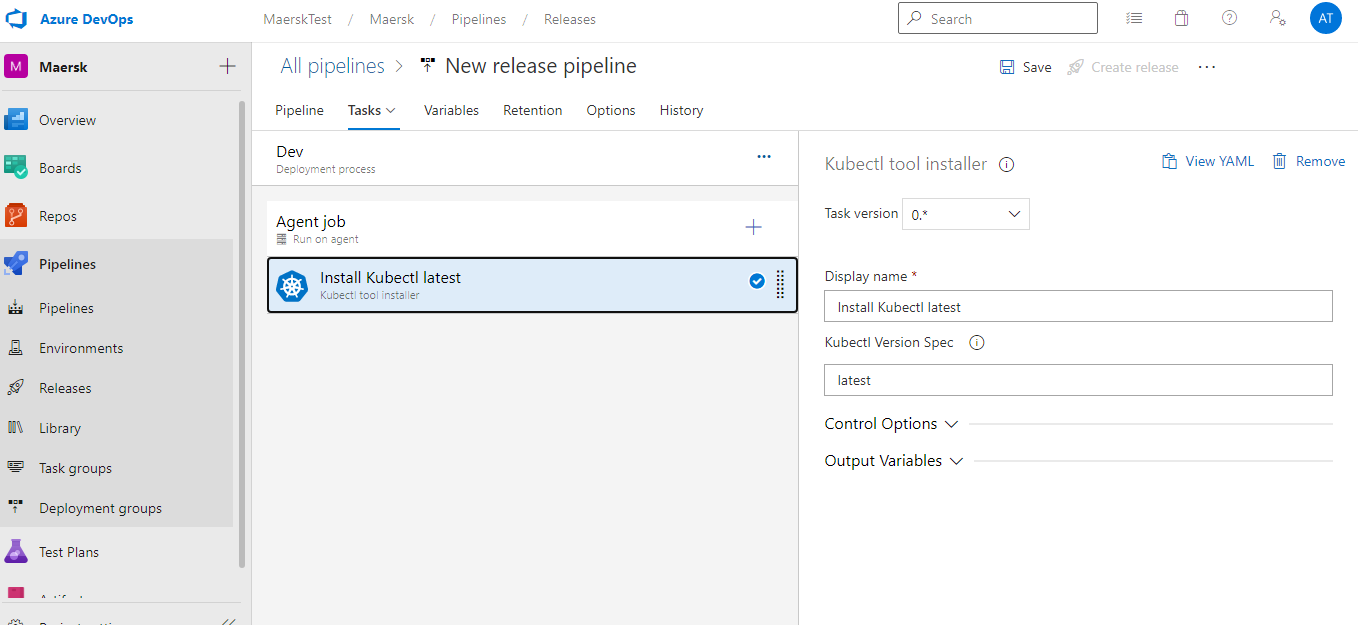
* Select the **Agent Pool**



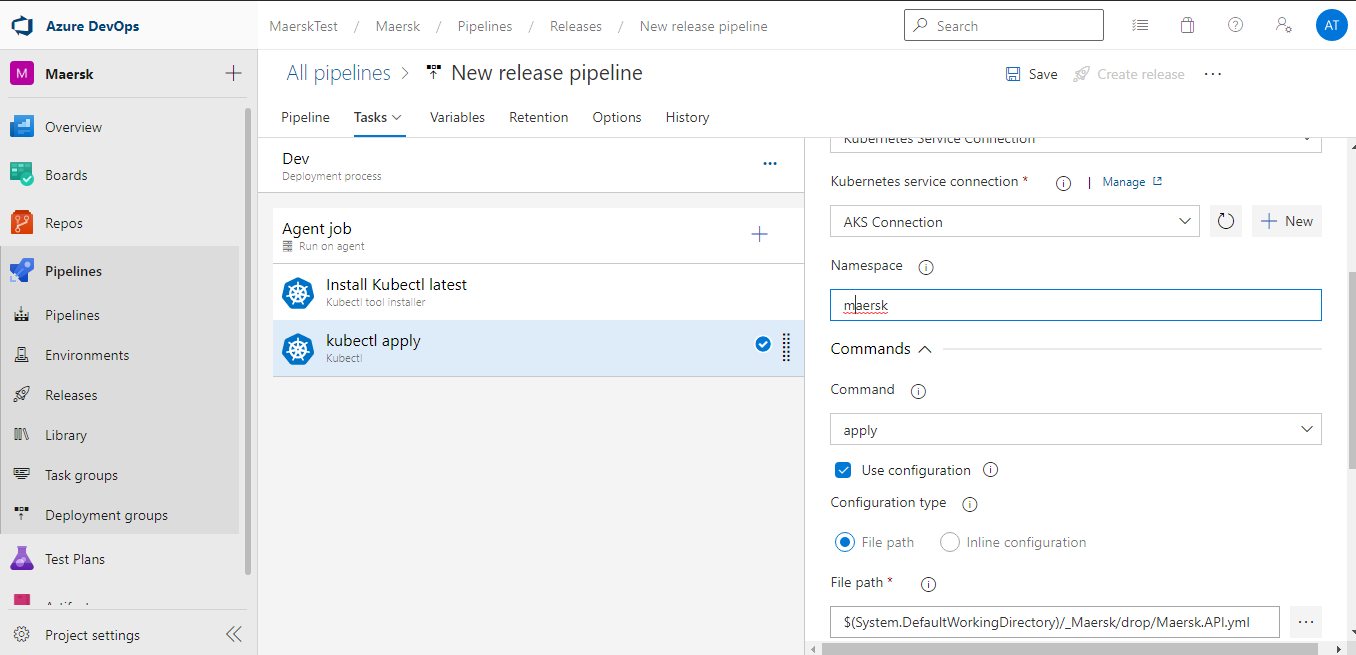
* Go to Project Settings ->Service Connection -> New Service Connection -> Select from the Azure Subscription dropdown.



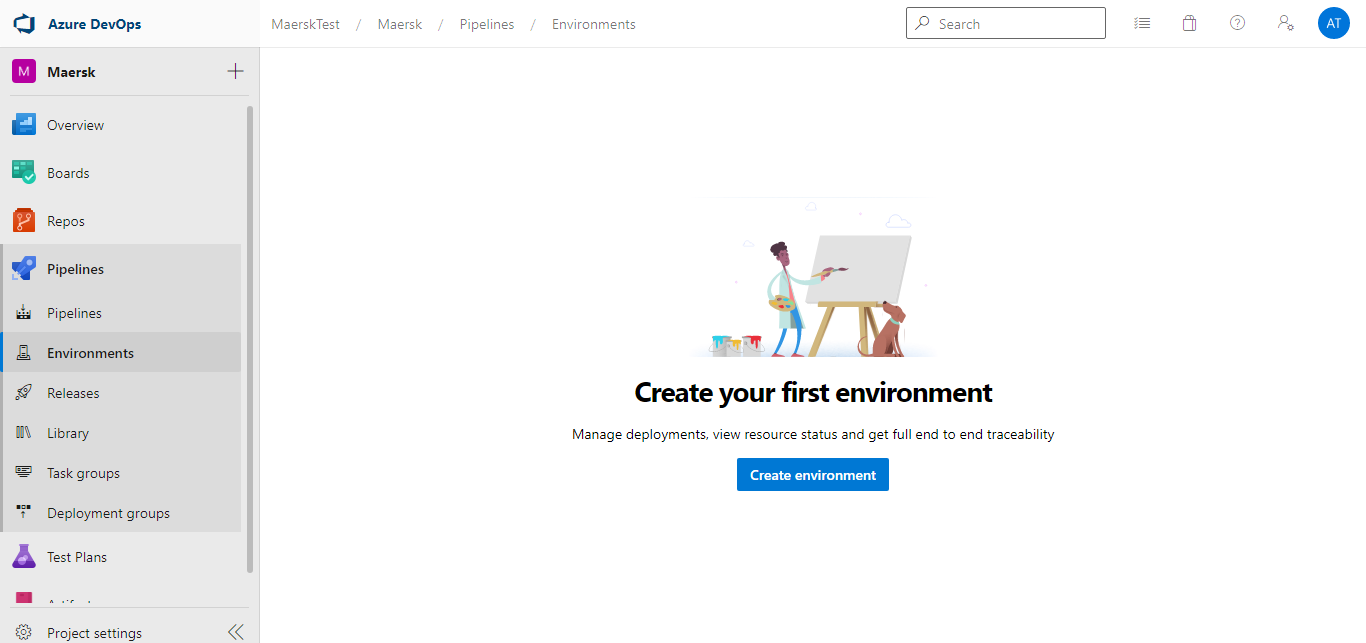
* Install Kubectl Latest Version in the Agent Machine.



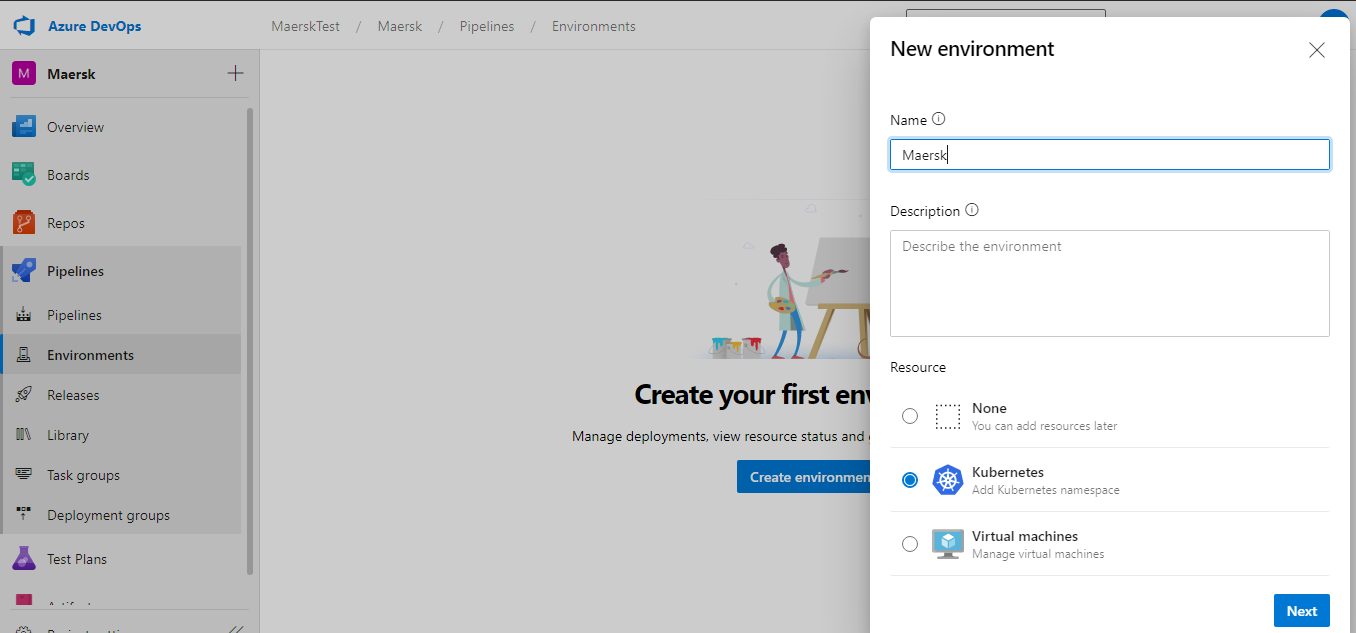
* Add kubectl task to deploy the MaerskAPI.yml to deploy to AKS.



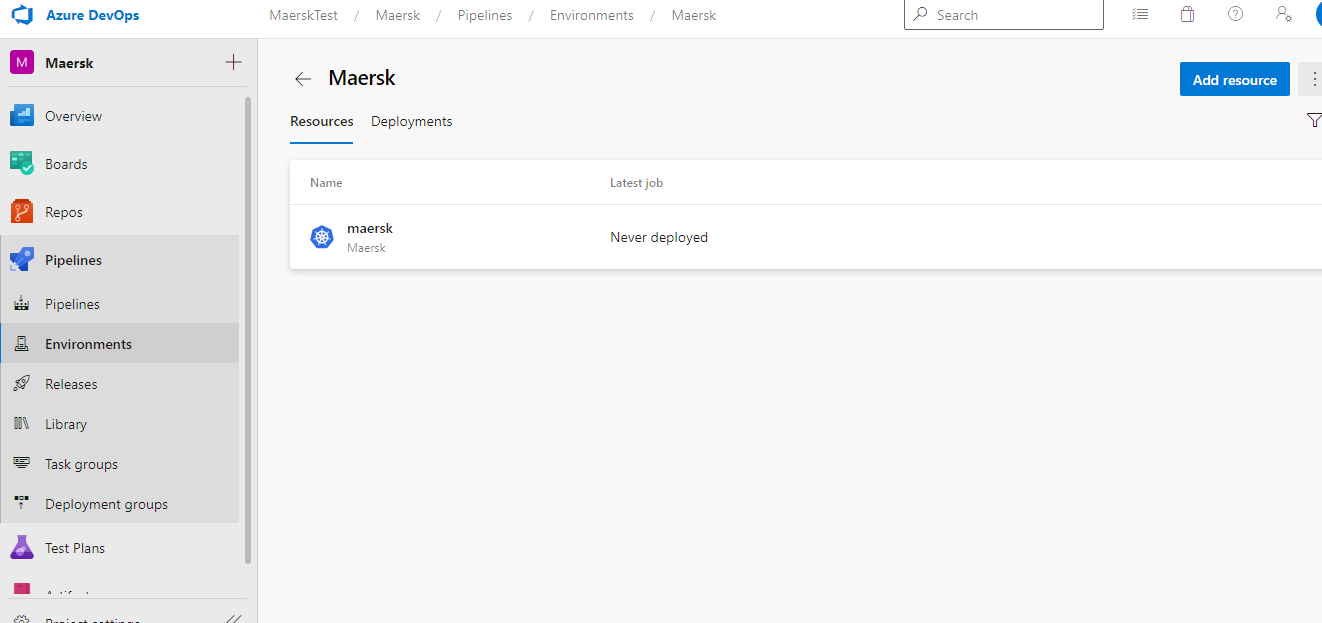
* Save the Release pipeline and run the Pipeline to deploy to the AKS.
* To verify the deployment/Pods status you can configure the **environment** tab.



* Select Create **environment**



* Select Resource to see all resource



* Select Deployment to see the Deployment status of the Pods.