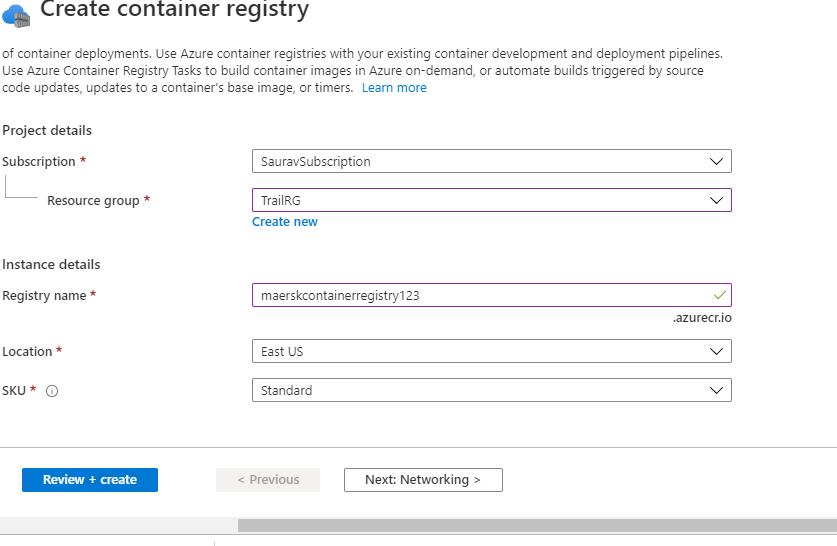
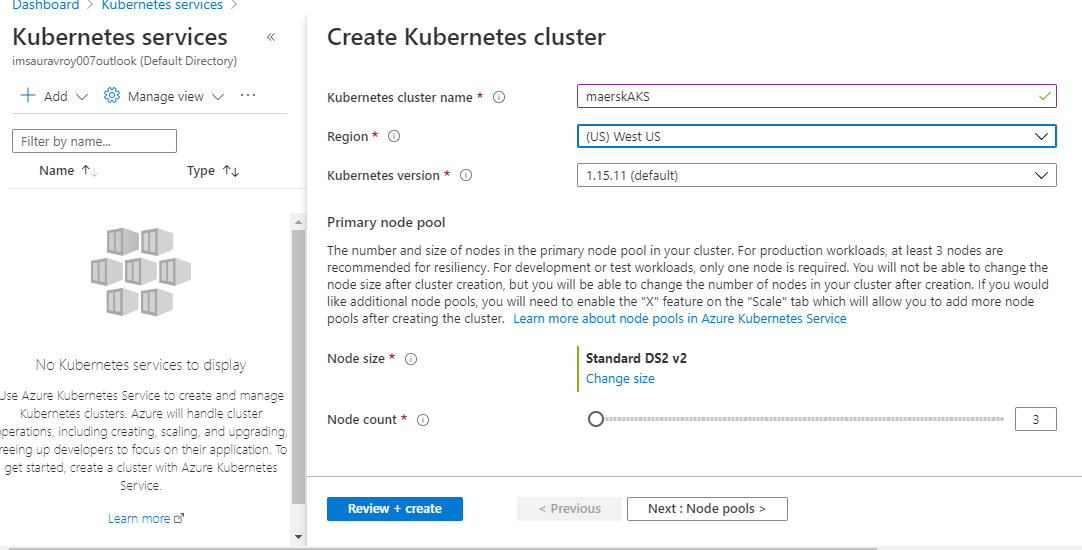
**SCENERIO – 3**

**These steps answers all the 1 -6 questions**

Step 1 – Create a azure container registry



Step 2 - Create Kubernetes Cluster



Step 3 - Authorize the AKS cluster to connect to ACR using Service Principle

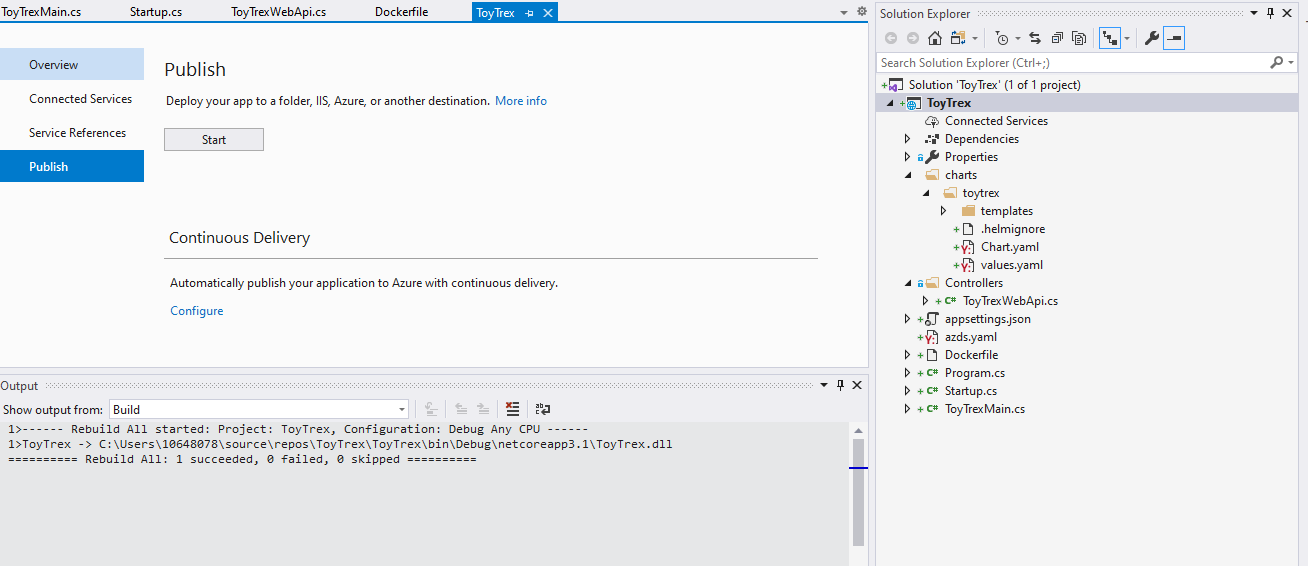
**az command below** -:

CLIENT\_ID=$(az aks show --resource-group TrailRG --name maerskAKS --query "servicePrincipalProfile.clientId" )

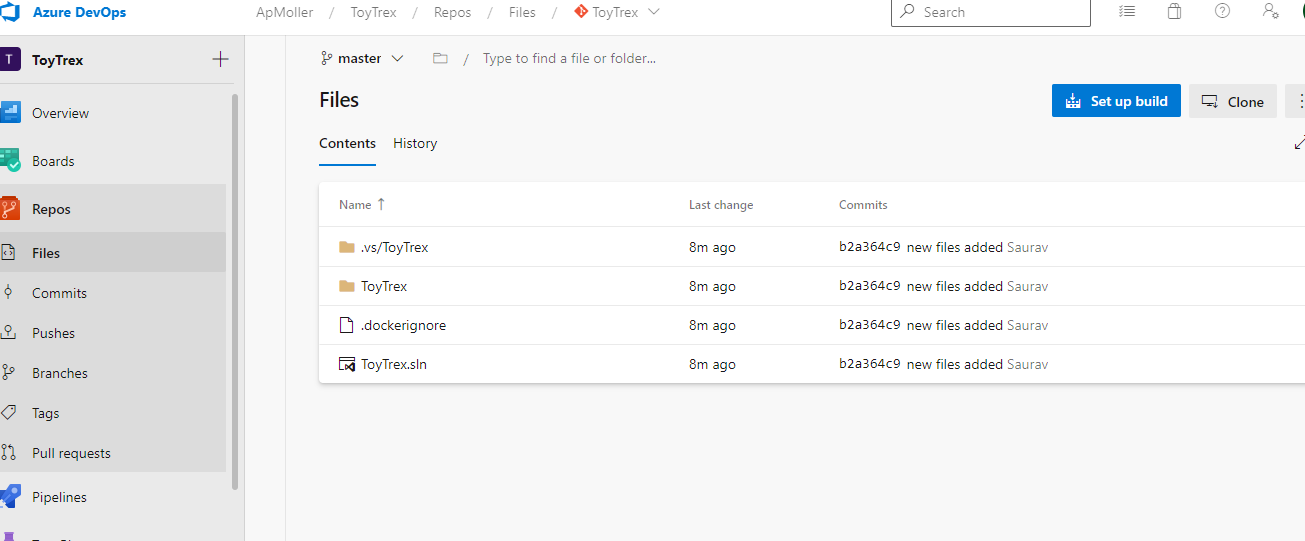
ACR\_ID=$(az acr show --name maerskcontainerregistry123 --resource-group TrailRG --query "id")

az role assignment create --assignee CLIENT\_ID --role acrpull --scope ACR\_ID

Step 4 – Create a Api project in Visual Studio with docker support enabled



Step 5 – Create New Project in AzureDevOps portal with the name TroyTrex, push the code base to the repo



Step 6 – Create Deployment.yml under manifest Folder of the project root like below

apiVersion: apps/v1beta1

kind: Deployment

metadata:

name: Toytrex-deployment

labels:

app: Toytrex

spec:

replicas: 1

selector:

matchLabels:

app: Toytrex

template:

metadata:

labels:

app: Toytrex

spec:

containers:

- name: k8s-Toytrex

image: maerskcontainerregistry123.azurecr.io/Toytrex:latest

ports:

- containerPort: 80

imagePullSecrets:

- name: maerskcontainerregistry123

---

apiVersion: v1

kind: Service

metadata:

name: Toytrex

labels:

app: Toytrex

spec:

ports:

- name: http-port

port: 80

targetPort: 80

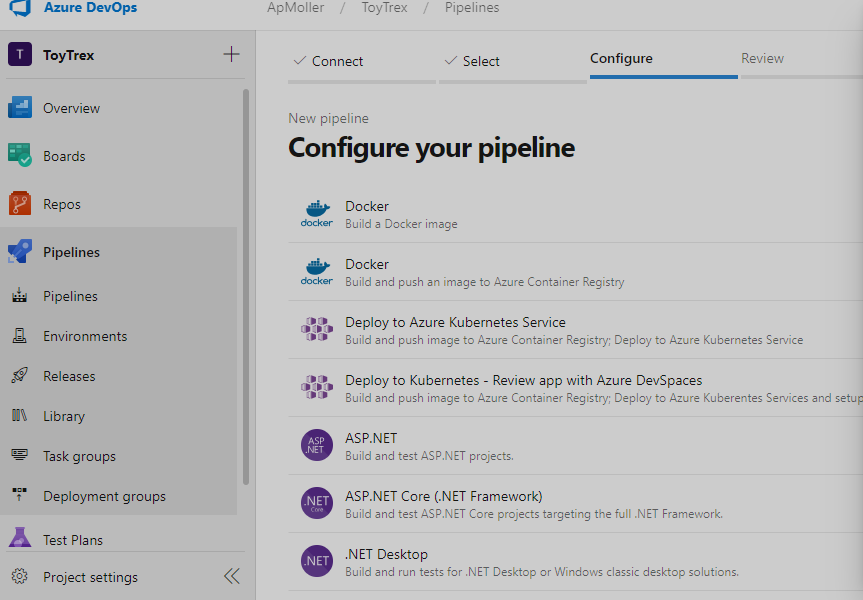
selector:

app: Toytrex

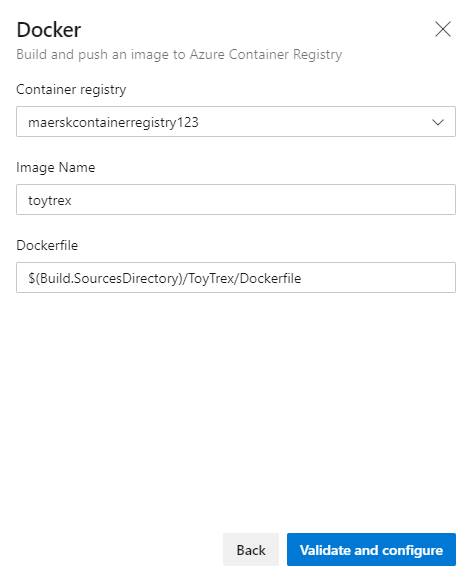
type: LoadBalancer

Step 7 - Click on SetUp build button on AzureDevOps Portal

Step 8- Select Docker Template (build and push to container registry)

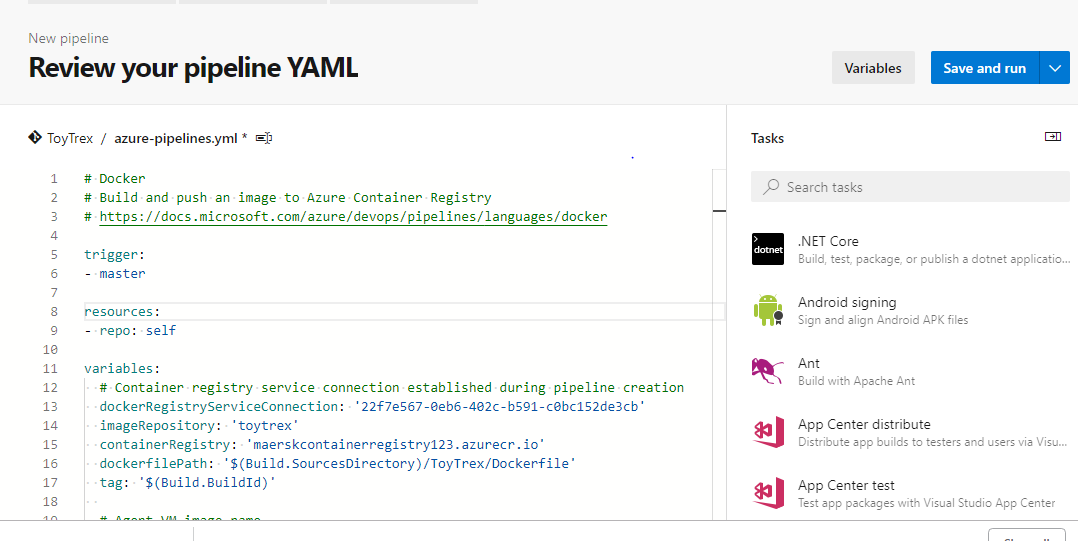


Step 9 – Enter Subscription details and Select the container registry

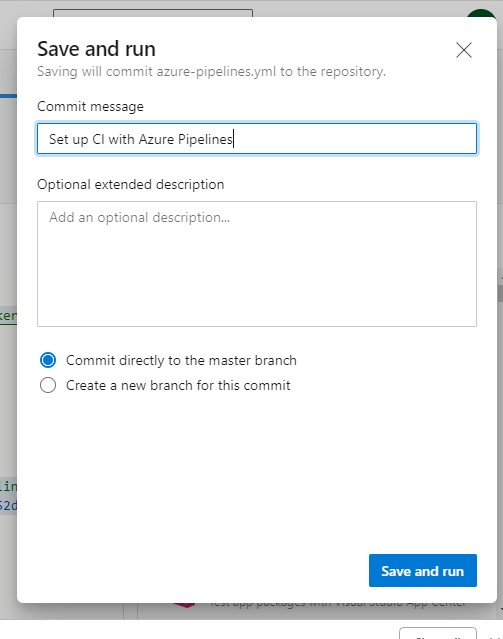


Step 10 – Click validate and configure button

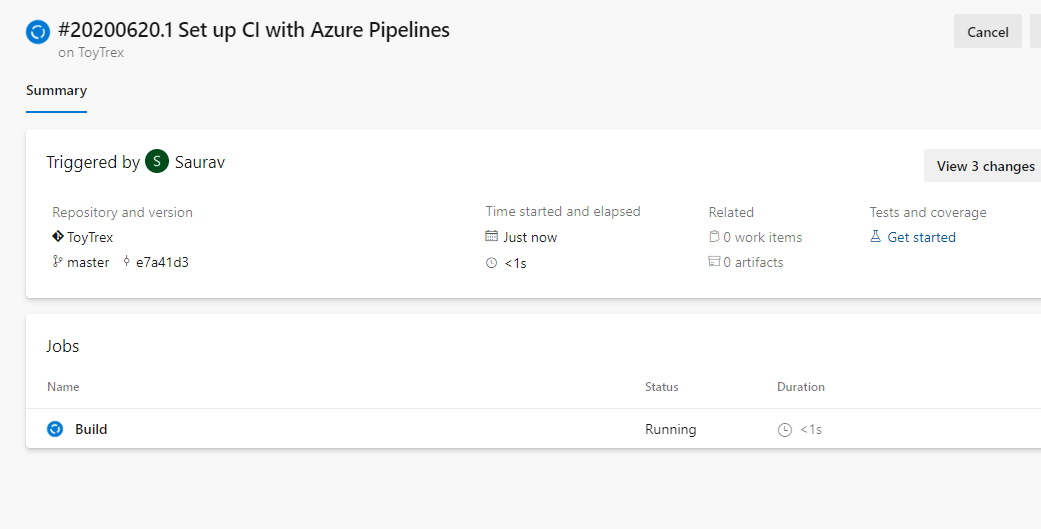
Step 11 – Review your build pipeline



Step 12 – Click Save and Run



Step 13 – Check the status of the build



Step 14 :

In addition to, include the latest tag and give buildId to image name with ToyTrex:$(Build.BuildId)