Project: Name Pronunciation

Team: CheckOps

Release: Version1

Platform: Microsoft Azure Public Cloud

**Pre-Requisite:**

1. Ensure resource group is created on azure.
2. Github repo url for below two components which are to be deployed on azure cloud
   1. CheckOps Python speech APIs
   2. CheckOps React UI Portal

**Installation:**

1. DB Deployment
   1. Connect to Yugabyte

Universe Name: univplat3grp1-1

Database: yugabyte

Username: yugabyte

Password: Hackathon22!

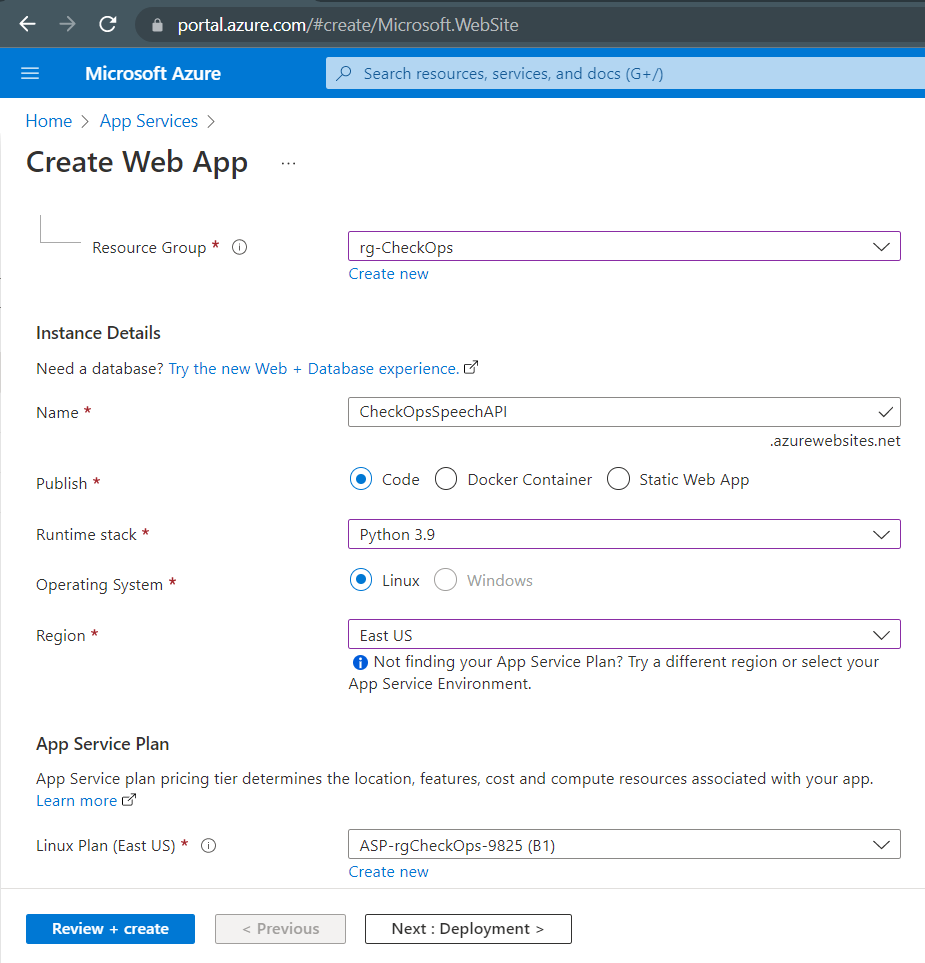
Public IPv4 Node1: 20.127.242.100

* 1. Execute this command

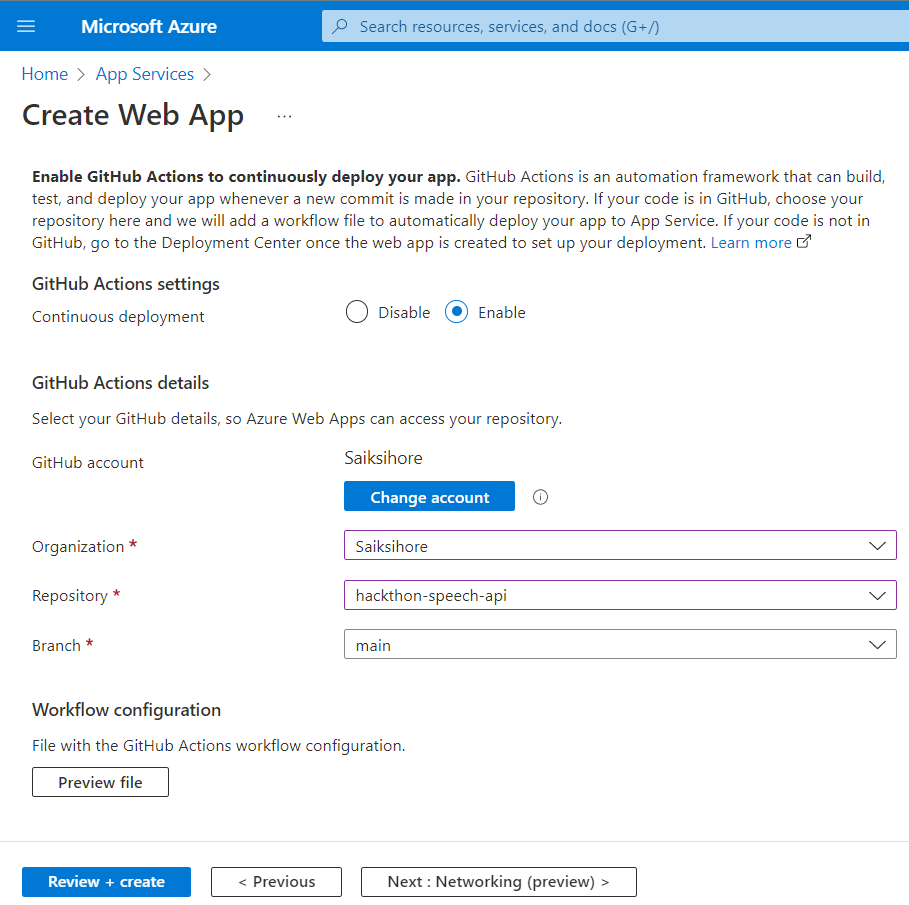
CREATE TABLE NAME\_PRONUNCIATION\_DETAILS(  
USER\_ID TEXT NOT NULL,  
FIRST\_NAME TEXT NOT NULL,  
LAST\_NAME TEXT NOT NULL,  
SHORT\_NAME TEXT NOT NULL,  
VOICE\_PATH text NOT NULL,  
CUSTOM\_VOICE\_PATH text NULL,  
CREATED\_TIMESTAMP timestamp,  
PRIMARY KEY (USER\_ID));

ii. Cloud Configuration and Application Installation

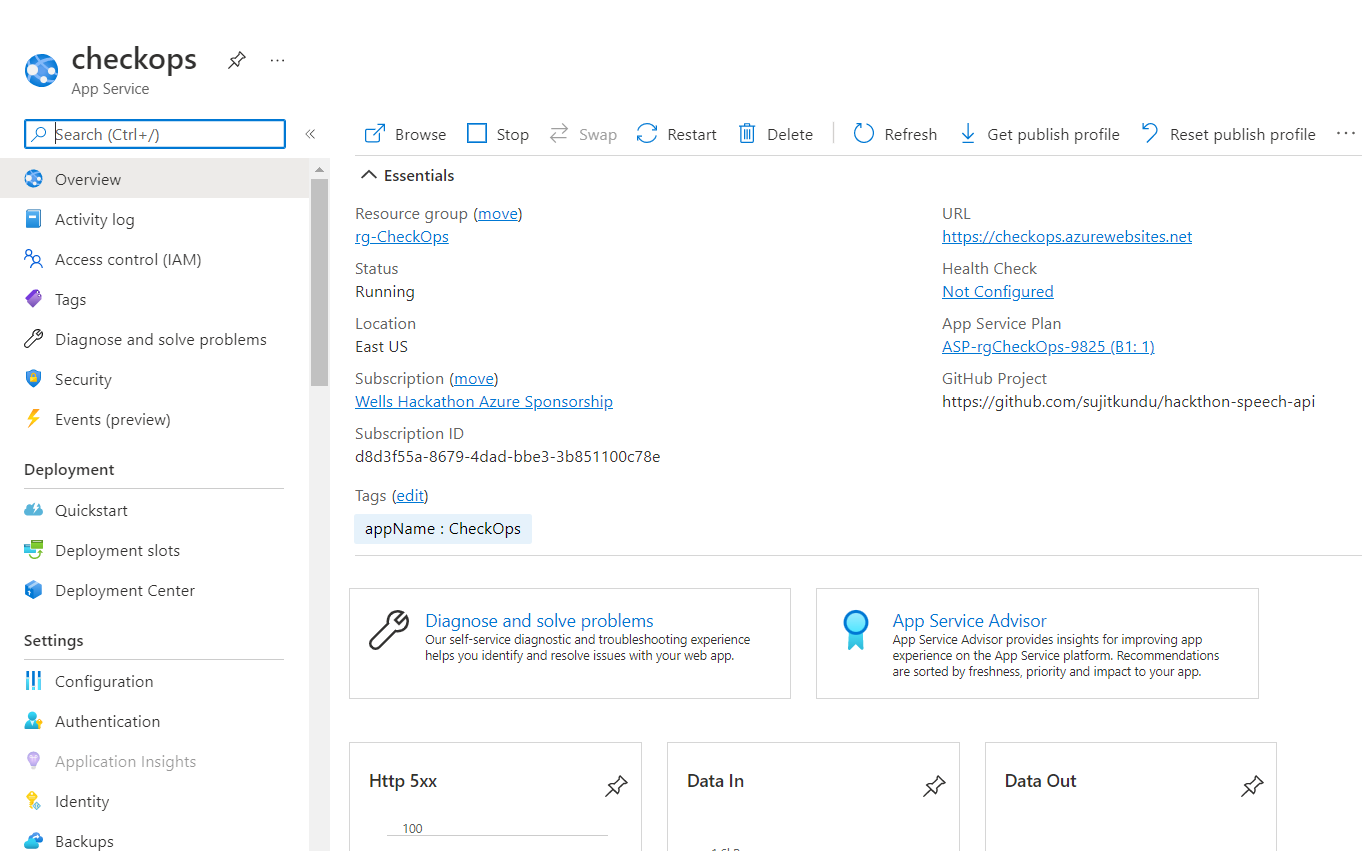
1. Login into <https://portal.azure.com/#home>
2. Click on App Services
3. Click on Create Button
4. Fill the details as given below



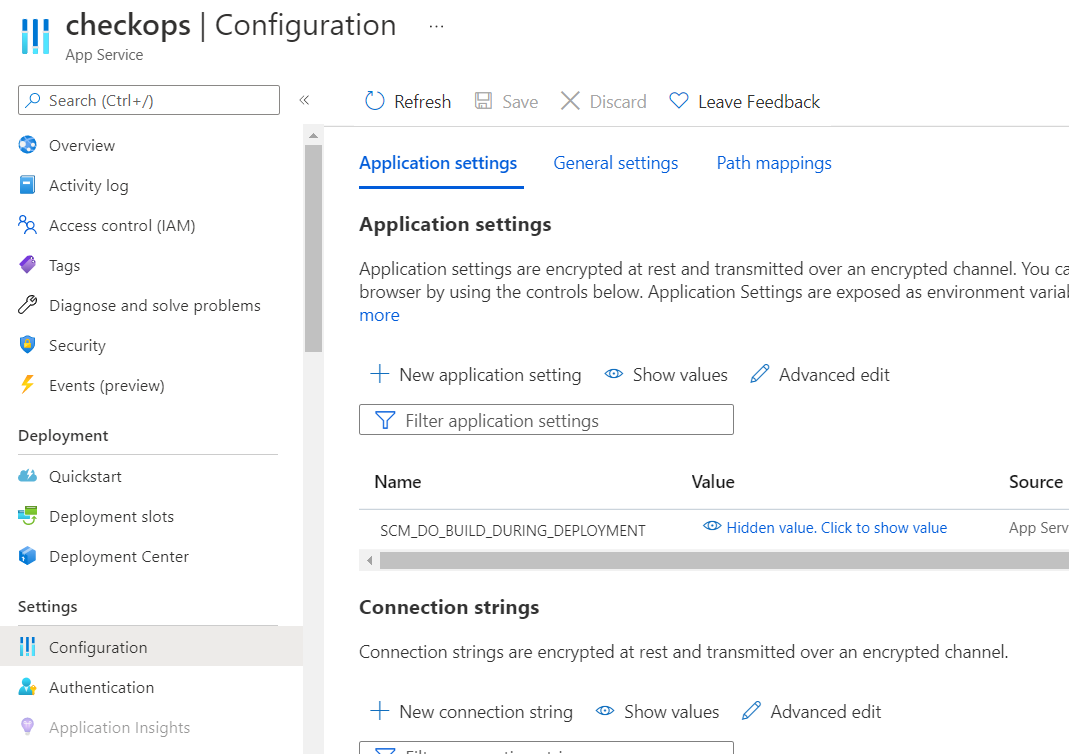
1. Click on Deployment button
2. Set the git configurations as show below



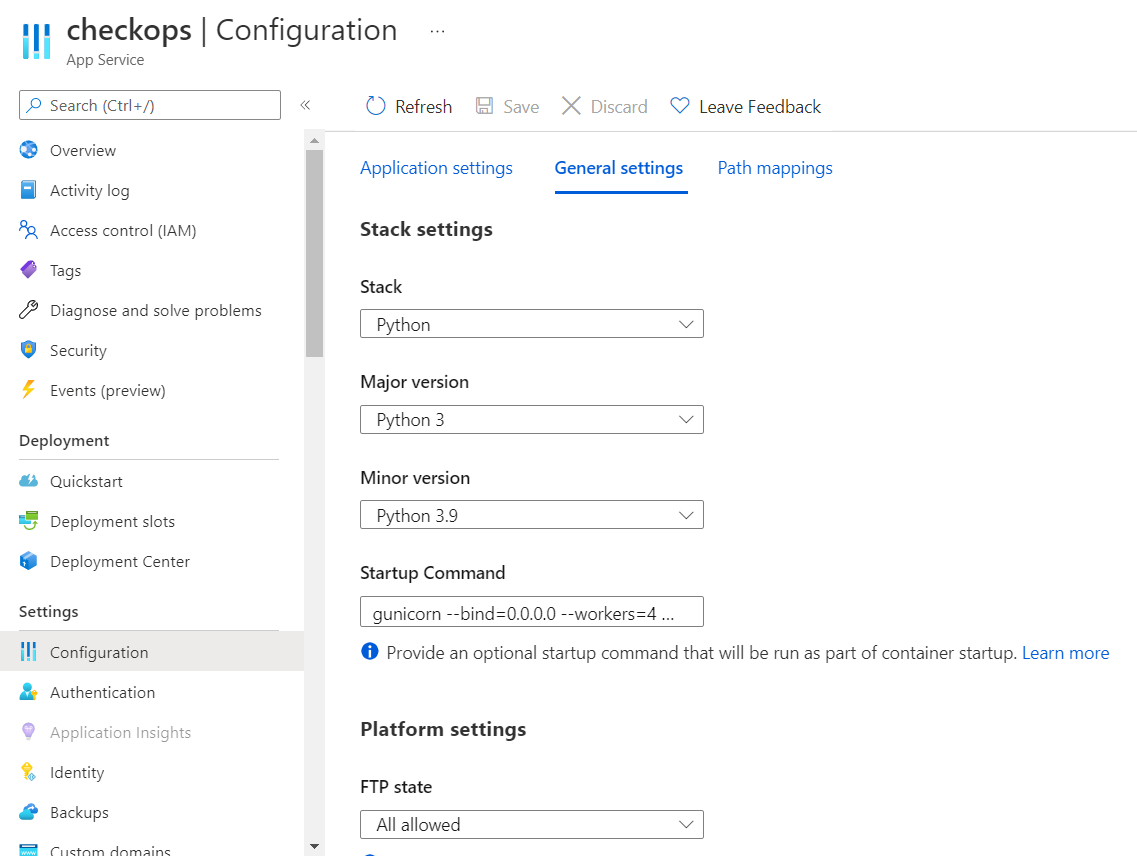
1. Click on Review+Create button
2. Wait for Deployment to complete
3. On successful deployment, below page will be shown



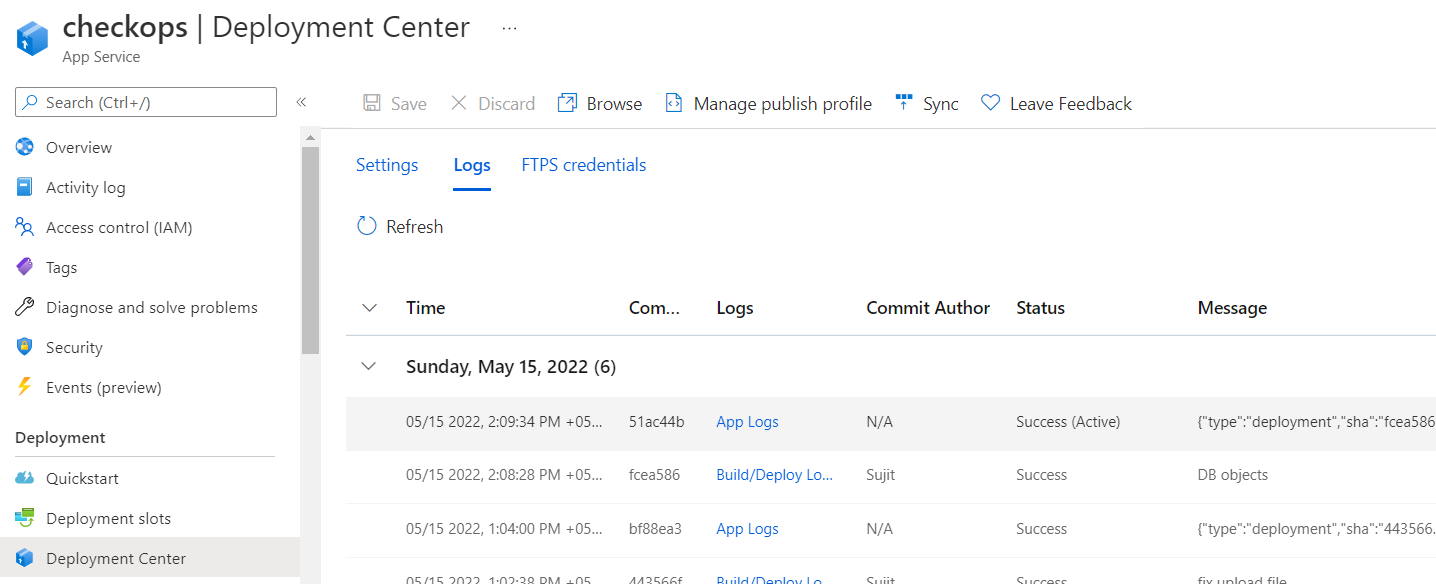
1. Click on Configuration on left pane under settings section



1. On above page, click on General Settings and configure the details as shown below



1. Ensure Startup command is “gunicorn --bind=0.0.0.0 --workers=4 --timeout 600 main:app”
2. Save and refresh.
3. Go to Deployment Center on checkops App Service page as below

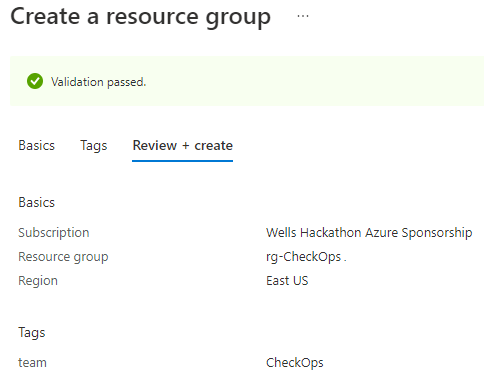


1. Click on App Logs to check the app start up logs
2. CheckOps Speech API Deployment is complete.

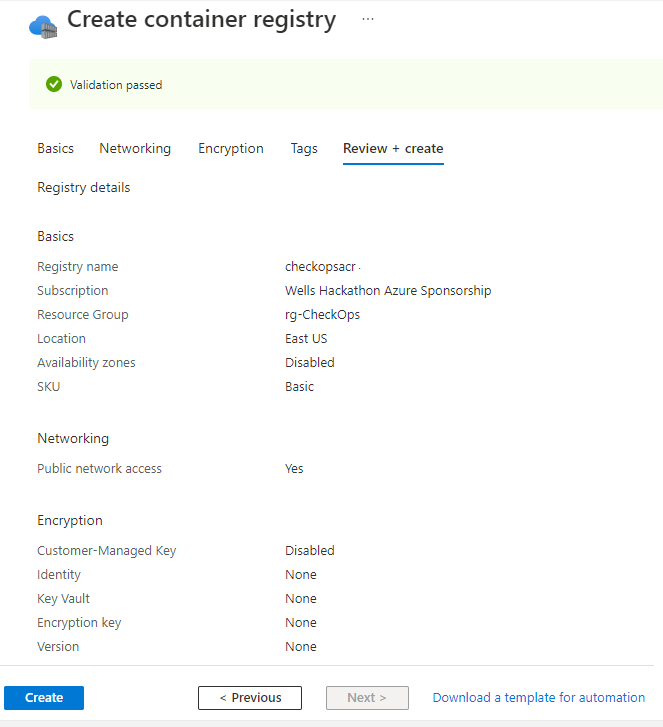
**CheckOpsNamePronunciation Portal Application Deployment**:

Prerequisite:

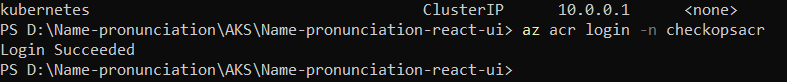
1. Azure CLI must be installed.
2. Kubectl CLI must be installed.
3. Create a Resource Group rg-CheckOps



1. Create Azure Container Registry



1. Login into the Azure Container Registry using Azure CLI

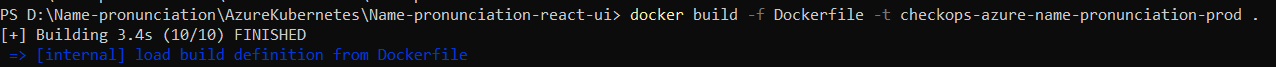


1. Dockerize the application by adding the docker file

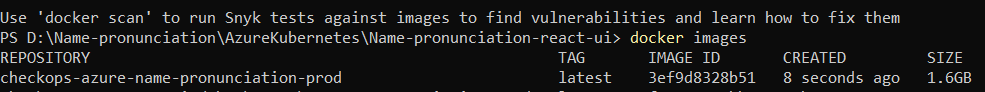
 

1. Create the docker image by running below command in Azure CLI

docker build -f Dockerfile -t checkops-azure-name-pronunciation-prod .



1. Validate the image is created by running below command in Azure CLI

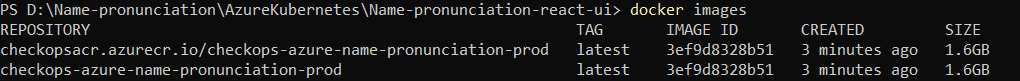


1. Tag the docker image by running below command in Azure CLI

docker tag 3ef9d8328b51 checkopsacr.azurecr.io/checkops-azure-name-pronunciation-prod

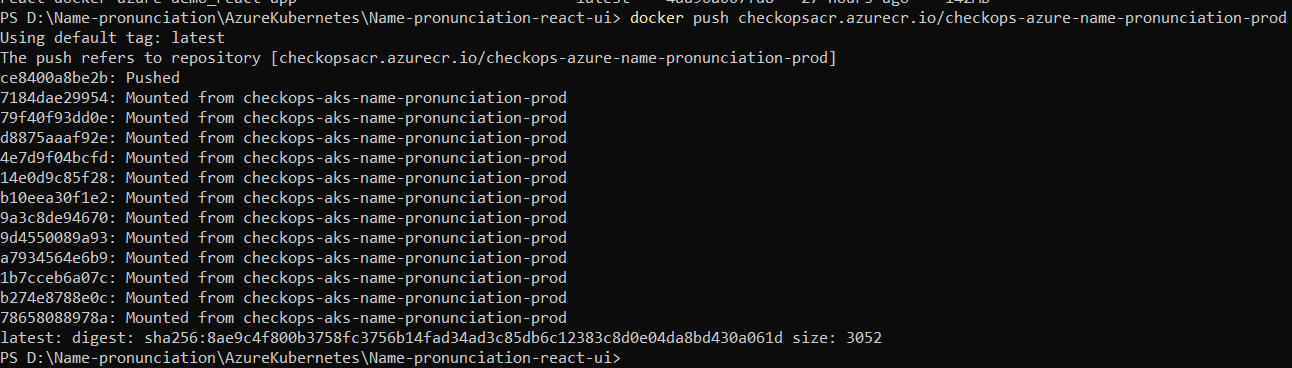


1. Validate the Tag by running below command in Azure CLI

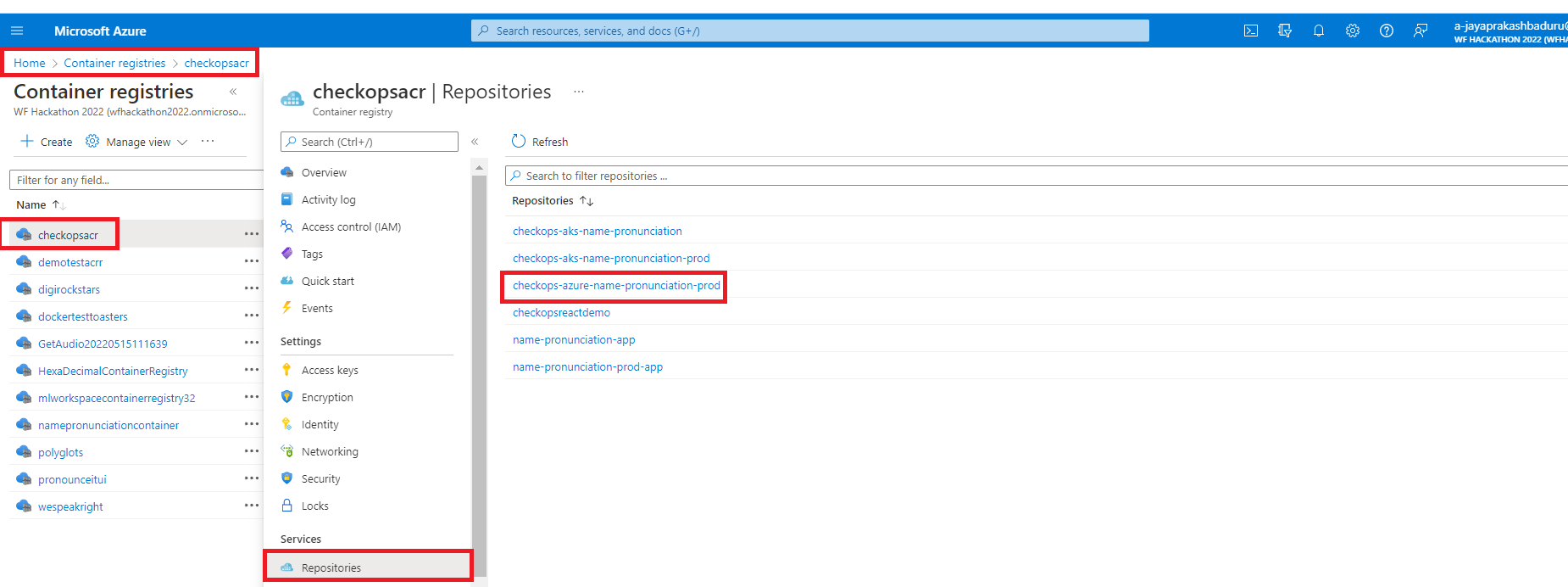


1. Push the image by running below command in Azure CLI

docker push checkopsacr.azurecr.io/checkops-azure-name-pronunciation-prod



1. Validate the image in the Azure Portal

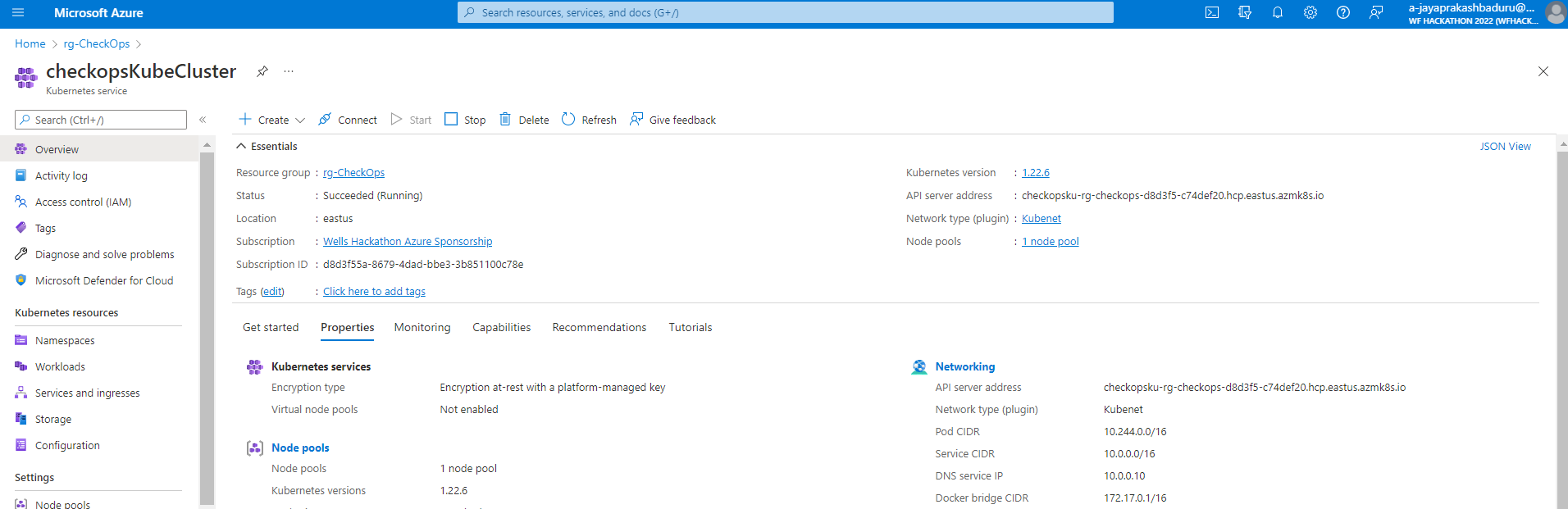


1. Create AKS Cluster by running below command in Azure CLI

az aks create --resource-group rg-CheckOps --name checkopsKubeCluster --node-count 2 --generate-ssh-keys --attach-acr checkopsacr

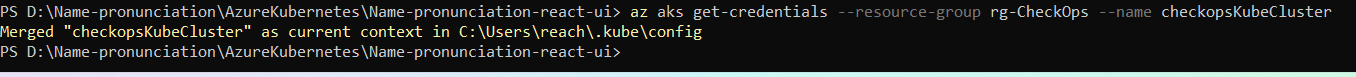


1. Validate the Kubernetes cluster



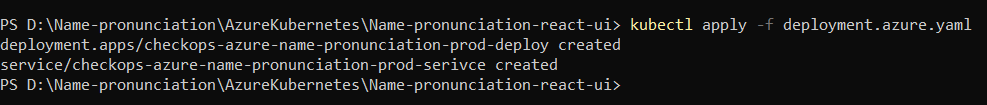
1. Get AKS cluster credentials by running below command

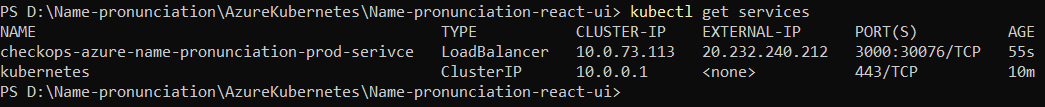
az aks get-credentials --resource-group rg-CheckOps --name checkopsKubeCluster

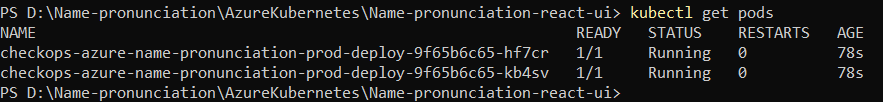


1. Create Pods and Services

kubectl apply -f deployment.azure.yaml







1. Validate the service by accessing

