**This is three-tier Application**

**Pre-requisites**

**Infrastructure provisioning**

**Using the below script, you can provision infra for this demo**

The following resources will be provisioned:

* A Resource Group
* An AKS Kubernetes Cluster
* An Image container registry
* A SQL Server
* A SQL Database

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

REGION="westus"

RGP="demorg"

CLUSTER\_NAME="demo-cluster"

ACR\_NAME="demoacrnew123"

SQLSERVER="demo-sqlserver"

DB="demodb"

#Create Resource group

az group create --name $RGP --location $REGION

#Deploy AKS

az aks create --resource-group $RGP --name $CLUSTER\_NAME --enable-addons monitoring --generate-ssh-keys --location $REGION

#Deploy ACR

az acr create --resource-group $RGP --name $ACR\_NAME --sku Standard --location $REGION

#Authenticate with ACR to AKS

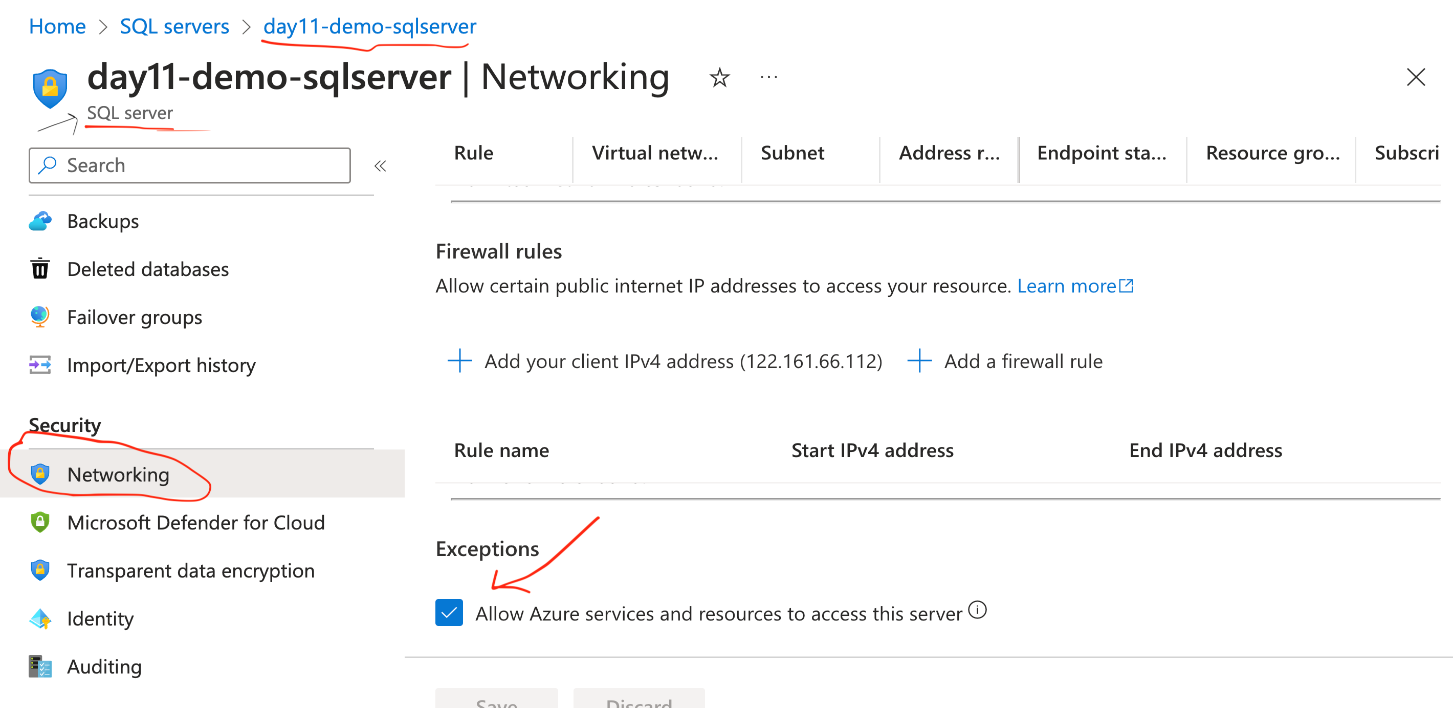
az aks update -n $CLUSTER\_NAME -g $RGP --attach-acr $ACR\_NAME

#Create SQL Server and DB

az sql server create -l $REGION -g $RGP -n $SQLSERVER -u sqladmin -p P2ssw0rd1234

az sql db create -g $RGP -s $SQLSERVER -n $DB --service-objective S0

**Change the Firewall settings of the SQL server**

[](https://private-user-images.githubusercontent.com/40286378/294747661-d421dd8b-1a85-447a-ad2d-f0ddb859953d.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..8S_x7y0FRJrg2asGK_w41RabJ9_Be0m_OGNpQ4zKqI8)

**Setup Azure DevOps Project**

**Pre-requisites**

Make sure the below Azure DevOps extensions are installed and enabled in your organization

* Replace Token
* Kubernetes extension

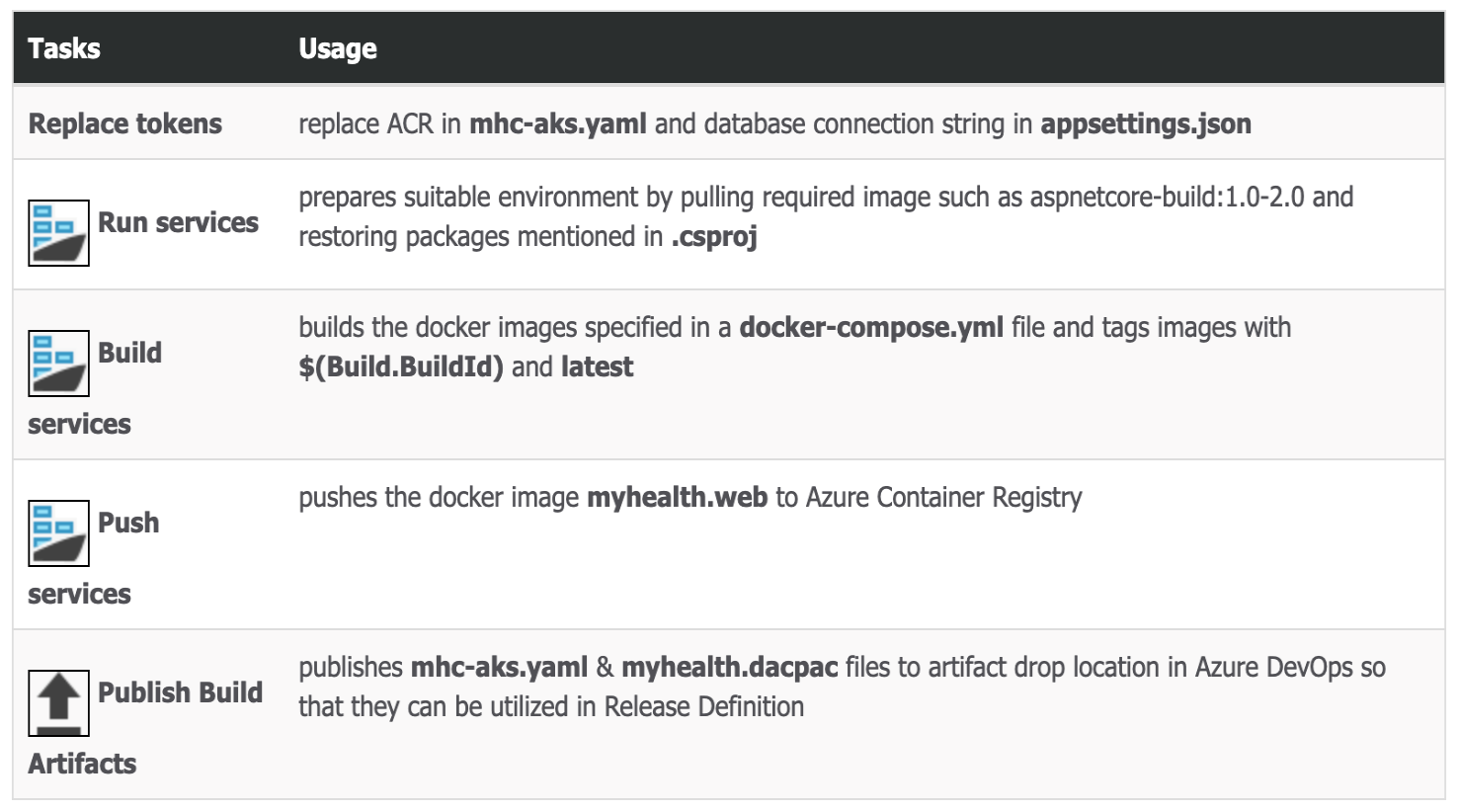
Once the infra is ready, go to dev.azure.com --> Project --> repos and import the below git repo, which has the source code and pipeline code

<https://github.com/piyushsachdeva/MyHealthClinic-AKS>

**Build and Release Pipeline**

* You can create your pipeline by following along the video or editing the existing pipeline. The below details need to be updated in the pipeline:
  + Azure Service connection
  + Token pattern
  + Pipeline variables
  + The Kubectl version should be the latest in the release pipeline
  + Secrets should be updated in the deployment step
  + ACR details in the pipeline should be updated

**Steps in the pipeline**

[](https://private-user-images.githubusercontent.com/40286378/294660529-64907bef-acbf-41f1-b3de-63b46681db3d.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..WdMJtASkNHljqeTULtKFWnkiIW46nfz-EbGD8gAzxT8)

**Destroy the resources at the end of the demo**

#!/bin/bash

# Set environment variables

REGION="westus"

RGP="day11-demo-rg"

CLUSTER\_NAME="day11-demo-cluster"

ACR\_NAME="day11demoacr"

SQLSERVER="day11-demo-sqlserver"

DB="mhcdb"

# Function to handle errors

handle\_error() {

echo "Error: $1"

exit 1

}

# Function to check if the resource exists

resource\_exists() {

az resource show --ids $1 &> /dev/null

}

# Delete Azure Kubernetes Service (AKS)

if resource\_exists $(az aks show --resource-group $RGP --name $CLUSTER\_NAME --query id --output tsv); then

az aks delete --resource-group $RGP --name $CLUSTER\_NAME || handle\_error "Failed to delete AKS."

else

echo "AKS not found. Skipping deletion."

fi

# Delete Azure Container Registry (ACR)

if resource\_exists $(az acr show --name $ACR\_NAME --resource-group $RGP --query id --output tsv); then

az acr delete --name $ACR\_NAME --resource-group $RGP || handle\_error "Failed to delete ACR."

else

echo "ACR not found. Skipping deletion."

fi

# Delete SQL Database

if resource\_exists $(az sql db show --resource-group $RGP --server $SQLSERVER --name $DB --query id --output tsv); then

az sql db delete --resource-group $RGP --server $SQLSERVER --name $DB || handle\_error "Failed to delete SQL Database."

else

echo "SQL Database not found. Skipping deletion."

fi

# Delete SQL Server

if resource\_exists $(az sql server show --resource-group $RGP --name $SQLSERVER --query id --output tsv); then

az sql server delete --resource-group $RGP --name $SQLSERVER || handle\_error "Failed to delete SQL Server."

else

echo "SQL Server not found. Skipping deletion."

fi

# Delete Resource Group

if resource\_exists $(az group show --name $RGP --query id --output tsv); then

az group delete --name $RGP || handle\_error "Failed to delete Resource Group."

else

echo "Resource Group not found. Skipping deletion."

fi

echo "Resources successfully deleted."