Az script to Provision the CDN and Storage account:

#Create the Resource Group

az group create --subscription ${{env.subscription}} -l ${{env.location}} --name ${{env.ResourceGroup}}

# Create the Storage account with geo-replication enabled

 az storage account create --subscription ${{env.subscription}} -n ${{env.storageaccount}} -g ${{env.ResourceGroup}} -l ${{env.location}} --sku Standard\_GRS

## Create Statis Website

az storage blob service-properties update --account-name ${{env.storageaccount}} --static-website --404-document 404.html --index-document index.html

# Create the CDN profile.

az cdn profile create --subscription ${{env.subscription}} --name "${{env.CDN\_Profile\_Name}}" --resource-group "${{env.ResourceGroup}}" --location global --sku ${{env.CDN\_sku}}

# Get the Static Website name

$Storage\_Website=$(az storage account show --subscription ${{env.subscription}} --name ${{env.storageaccount}} --resource-group ${{env.ResourceGroup}} --query "primaryEndpoints.web" --output tsv)

 # Get the Hostname from website

  $Origin\_Hostname=([System.Uri]"$Storage\_Website").Host

 # Create the CDN endpoint

az cdn endpoint create --subscription ${{env.subscription}} --resource-group ${{env.ResourceGroup}} --name ${{env.CDN\_Endpoint\_Name}} --profile-name ${{env.CDN\_Profile\_Name}} --origin $Origin\_Hostname --origin-host-header $Origin\_Hostname --location global --enable-compression

Az Script to Provision the Virtual network with Subnet

# Create Resource Group

az group create --subscription ${{env.subscription}} -l ${{env.location}} --name ${{env.RG}}

# Create a Virtual Network for AKS cluster

az network vnet create --subscription ${{env.subscription}} -g ${{env.RG}} --location ${{env.location}} -n ${{env.Vnet}} --address-prefix "${{env.Vnet\_addressPrefix}}"

# Create Subnet with required service endpoints

az network vnet subnet create --subscription ${{env.subscription}} --resource-group ${{env.RG}}  --vnet-name ${{env.Vnet}} --name ${{env.Subnet1}} --address-prefix "${{env.Subnet1\_addressPrefix}}" --service-endpoints Microsoft.Storage Microsoft.Sql Microsoft.AzureCosmosDB Microsoft.KeyVault Microsoft.ContainerRegistry

Az Script to Provision the Cosmos DB

# Create the Cosmos DB instance

az cosmosdb create --name ${{env.ServerName}} --resource-group ${{env.ResourceGroup}} --subscription ${{env.subscription}} --kind ${{env.kind}} --locations regionName=${{env.Primaryregion}} failoverPriority=${{env.Primaryregion\_FP}} isZoneRedundant=${{env.Primaryregion\_ZR}} --locations regionName=${{env.Secondaryregion}} failoverPriority=${{env.Secondaryregion\_FP}} isZoneRedundant=${{env.Secondaryregion\_ZR}} --backup-policy-type ${{env.Backup\_Policy}} --enable-automatic-failover ${{env.Automatic\_failover}} --server-version ${{env.serverversion}} --tags "${{env.tags}}" --enable-virtual-network ${{env.Enable\_virtual\_network}} --enable-public-network ${{env.Enable\_public\_network}}

# Create the Database

az cosmosdb mongodb database create --account-name ${{env.ServerName}}  --resource-group ${{env.ResourceGroup}} --name ${{env.DBName}}

# Create the Collections

az cosmosdb mongodb collection create -g ${{env.ResourceGroup}} -a ${{env.ServerName}} -d ${{env.DBName}} -n ${{env.collectionName}} --shard "${{env.shardkey}}" --idx "@ Index.json"

#Add service endpoint security

 az cosmosdb network-rule add -n ${{env.ServerName}} --subscription ${{env.subscription}} -g ${{env.ResourceGroup}} --virtual-network ${{env.Vnet}} --subnet $SubnetID --ignore-missing-vnet-service-endpoint false

Az script to Provision the SQL server and Database

# Create Primary Resource Group for Primary Region

 az group create --subscription ${{env.subscription}} -l ${{env.Primary\_location}} --name ${{env.Primary\_RG}}

# Create the Primary SQL server

 az sql server create --subscription ${{env.subscription}} -l ${{env.Primary\_location}} -g ${{env.Primary\_RG}} -n ${{env.Primary\_SQLServerName}} -u ${{env.SQL\_User}} -p ${{secrets.SQL\_PASSWORD}} --minimal-tls-version 1.2

 # Create Secondary Resource Group for secondary region

az group create --subscription ${{env.subscription}} -l ${{env.Secondary\_location}} --name ${{env.Secondary\_RG}}

 # Create the Secondary SQL server for Uk west location

az sql server create --subscription ${{env.subscription}} -l ${{env.Secondary\_location}} -g ${{env.Secondary\_RG}} -n ${{env.Secondary\_SQLServerName}} -u ${{env.SQL\_User}} -p ${{secrets.SQL\_PASSWORD}} --minimal-tls-version 1.2

# Create the Failover Group

az sql failover-group create --subscription ${{env.subscription}} --name "${{env.FGName}}" --resource-group "${{env.Primary\_RG}}" --server "${{env.Primary\_SQLServerName}}" --partner-resource-group "${{env.Secondary\_RG}}" --partner-server "${{env.Secondary\_SQLServerName}}" --failover-policy Automatic --grace-period 1

 #Create SQL Database for app

 az sql db create --subscription ${{env.subscription}} -g ${{env.Primary\_RG}} -s ${{env.Primary\_SQLServerName}} -n ${{env.DatabaseName}} -e ${{env.Edition}} -f ${{env.ComputeFamily}} --capacity ${{env.maxVcore}} --max-size ${{env.DataStorageSize}} --compute-model ${{env.ComputeModel}}

# Add Database to failover group

az sql failover-group update --subscription ${{env.subscription}} -g ${{env.Primary\_RG}} --name "${{env.FGName}}" --server "${{env.Primary\_SQLServerName}}" --add-db "${{env.DatabaseName }}"

# Service Endpoint security

az sql server vnet-rule create --subscription ${{env.subscription}} --server ${{env.Primary\_SQLServerName}} --name ${{env.SqlVnetRule1}} -g ${{env.Primary\_RG}} --subnet ${{env.SubnetID}} --ignore-missing-endpoint false

Az script to Provision the Keyvault

az keyvault create --subscription ${{env.subscription}} --location ${{env.location}} --name ${{env.AzureKeyVaultName}} --resource-group ${{env.ResourceGroup}} --sku ${{env.sku}} --network-acls-vnets $SubnetID --enable-soft-delete ${{env.Soft\_delete}} --retention-days ${{env.Retention\_period}} --tags ${{env.tags}}

Az script to Provision the Container registry

az acr create --subscription ${{env.subscription}} --name ${{env.RegistryName}} --resource-group ${{env.ResourceGroup}} --location ${{env.location}} --sku ${{env.sku}} --admin-enabled ${{env.admin-enabled}}

Az Script to Provision the ALS cluster with Application Gateway

 az aks create --subscription ${{env.subscription}} --resource-group ${{env.ResourceGroup}} --name ${{env.AKSName}} --kubernetes-version ${{env.K8SVersion}} --network-plugin ${{env.NetworkPlugin}} --vnet-subnet-id "$SubnetID" --service-cidr ${{env.service-cidr}} --dns-service-ip ${{env.dns-service-ip}} --docker-bridge-address ${{env.docker-bridge-address}} --zones ${{env.Availability\_zones}} --enable-managed-identity --nodepool-name ${{env.nodepool-name}} --node-vm-size ${{env.node-vm-size}} --enable-cluster-autoscaler --node-count ${{env.node-count}} --min-count ${{env.min-count}} --max-count ${{env.max-count}} -a ingress-appgw --appgw-name ${{env.appgw-name}} --appgw-subnet-cidr "${{env.appgw-subnet-prefix}}" --generate-ssh-keys –yes

Az Script to Provision Traffic manager (Used Priority Routing method)

# Create a Traffic manager profile

  az network traffic-manager profile create --name ${{env.AzureTrafficManager}} --resource-group ${{env.ResourceGroup}} --subscription ${{env.subscription}} --routing-method Priority --path /  --protocol https  --unique-dns-name ${{env.Failover\_uniqueDnsName}} --ttl 30 --port 443 --max-failures 2 --interval 30 --timeout 10

# Create a Traffic Manager Primary Endpoint

 az network traffic-manager endpoint create --name ${{env.Primary\_Endpoint}} --profile-name ${{env.AzureTrafficManager}} --resource-group ${{env.ResourceGroup}} --subscription ${{env.subscription}} --type ${{env.Primary\_Endpointtype}} --endpoint-status ${{env.Primary\_Endpoint\_Status}} --priority ${{env.Primary\_Endpoint\_Priority}} --target ${{env.Primary\_Endpoint\_FQDN}}

#  # Create a Traffic Manager Failover Endpoint

az network traffic-manager endpoint create --name ${{env.Failover\_Endpoint}} --profile-name ${{env.AzureTrafficManager}} --resource-group ${{env.ResourceGroup}} --subscription ${{env.subscription}} --type ${{env.Failover\_Endpointtype}} --endpoint-status ${{env.Failover\_Endpoint\_Status}} --priority ${{env.Failover\_Endpoint\_Priority}} --target ${{env.Failover\_Endpoint\_FQDN}}