Week7: **Assignment 2:- Configuring and Securing ACR and AKS**

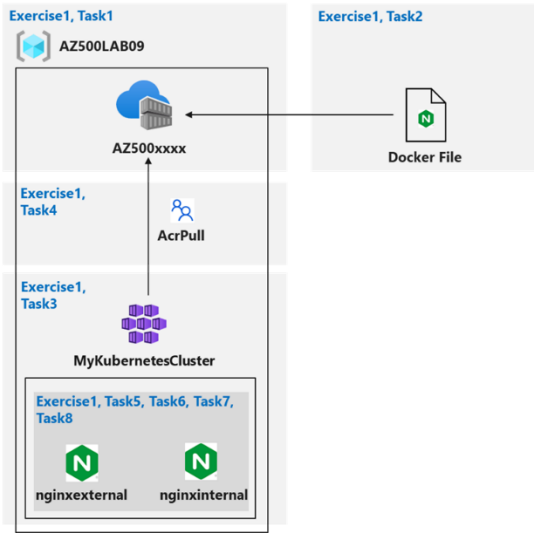
Report by: **Aisha Khalifan, cs-cns04-23014**

Introduction

You have been asked to deploy a proof of concept with Azure Container Registry and Azure Kubernetes Service. Specifically, the proof of concept should demonstrate:

- Using Dockerfile to build an image.
- Using Azure Container Registry to store images.
- Configuring an Azure Kubernetes Service.
- Securing and accessing container applications both internally and externally.Configuring and Securing ACR and AKS diagram

Configuring and Securing ACR and AKS diagram



Intructions

Lab files:

- **\Allfiles\Labs\09\nginxexternal.yaml**
- **\Allfiles\Labs\09\nginxinternal.yaml**

Exercise 1: Configuring and Securing ACR and AKS

Estimated timing: 45 minutes

For all the resources in this lab, we are using the **East (US)** region.

In this exercise, you will complete the following tasks:

- Task 1: Create an Azure Container Registry
- Task 2: Create a Dockerfile, build a container and push it to Azure Container Registry

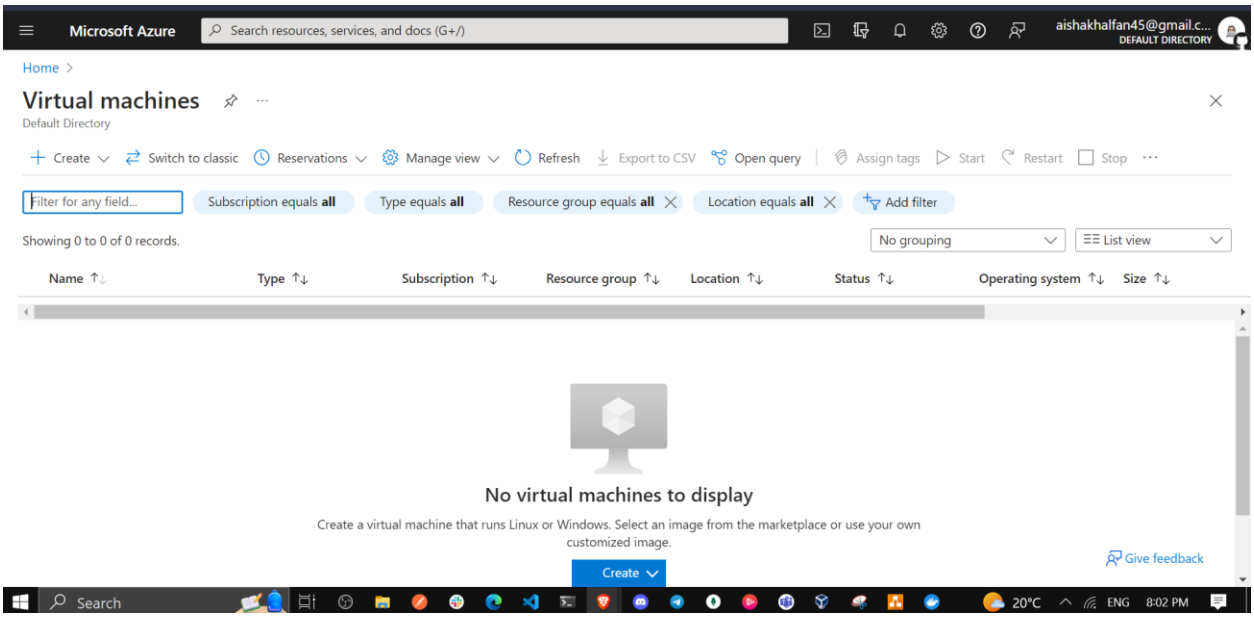
- Task 3: Create an Azure Kubernetes Service cluster
- Task 4: Grant the AKS cluster permissions to access the ACR
- Task 5: Deploy an external service to AKS
- Task 6: Verify the you can access an external AKS-hosted service
- Task 7: Deploy an internal service to AKS
- Task 8: Verify the you can access an internal AKS-hosted service

Task 1: Create an Azure Container Registry

In this task, you will create a resource group for the lab an Azure Container Registry.

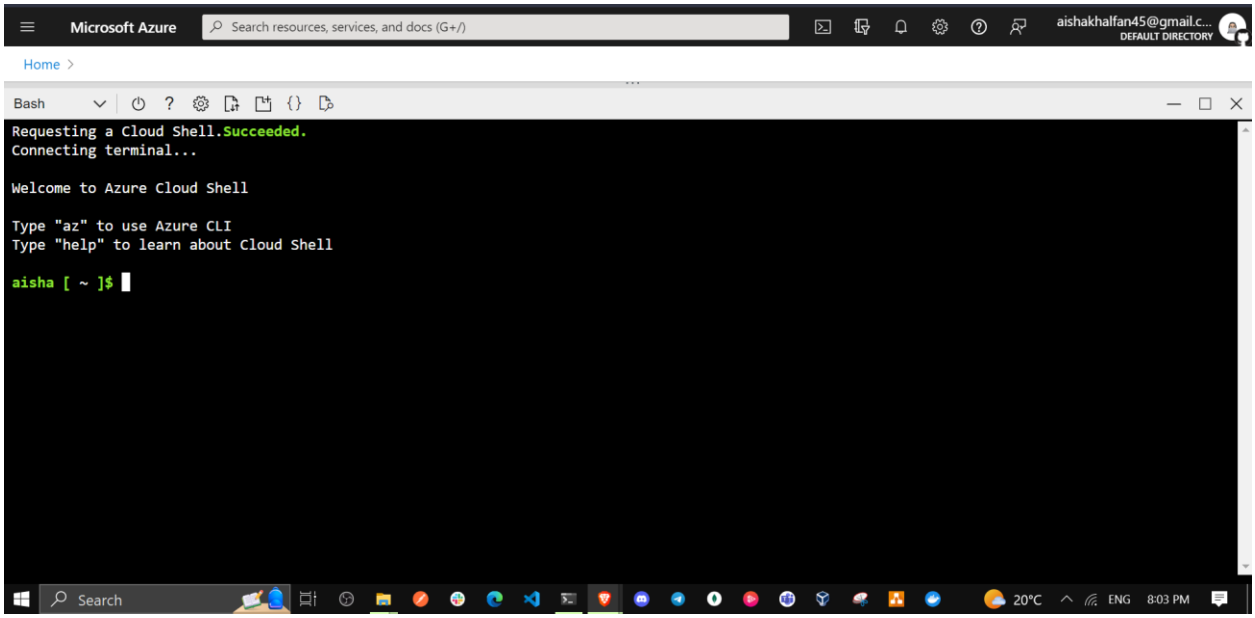
1. Sign-in to the Azure portal **https://portal.azure.com/**.

   Note: Sign in to the Azure portal using an account that has the Owner or Contributor role in the Azure subscription you are using for this lab and the Global Administrator role in the Microsoft Entra tenant associated with that subscription.



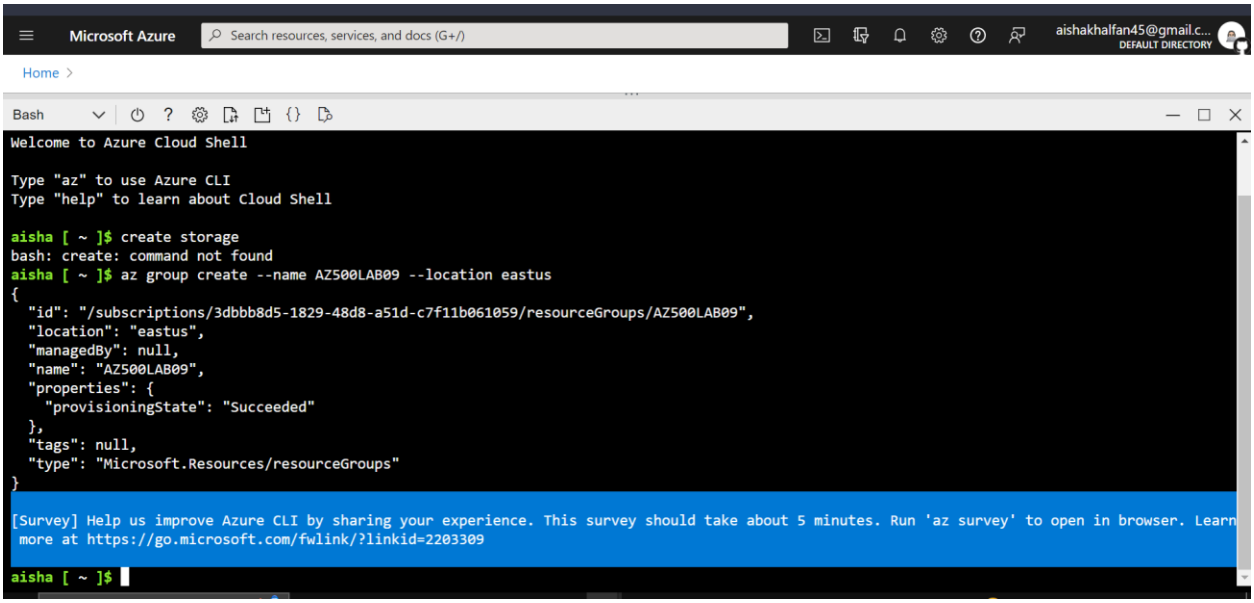
2. In the Azure portal, open the Cloud Shell by clicking the first icon in the top right of the Azure Portal. If prompted, click **Bash** and **Create storage**.

3. Ensure **Bash** is selected in the drop-down menu in the upper-left corner of the Cloud Shell pane.



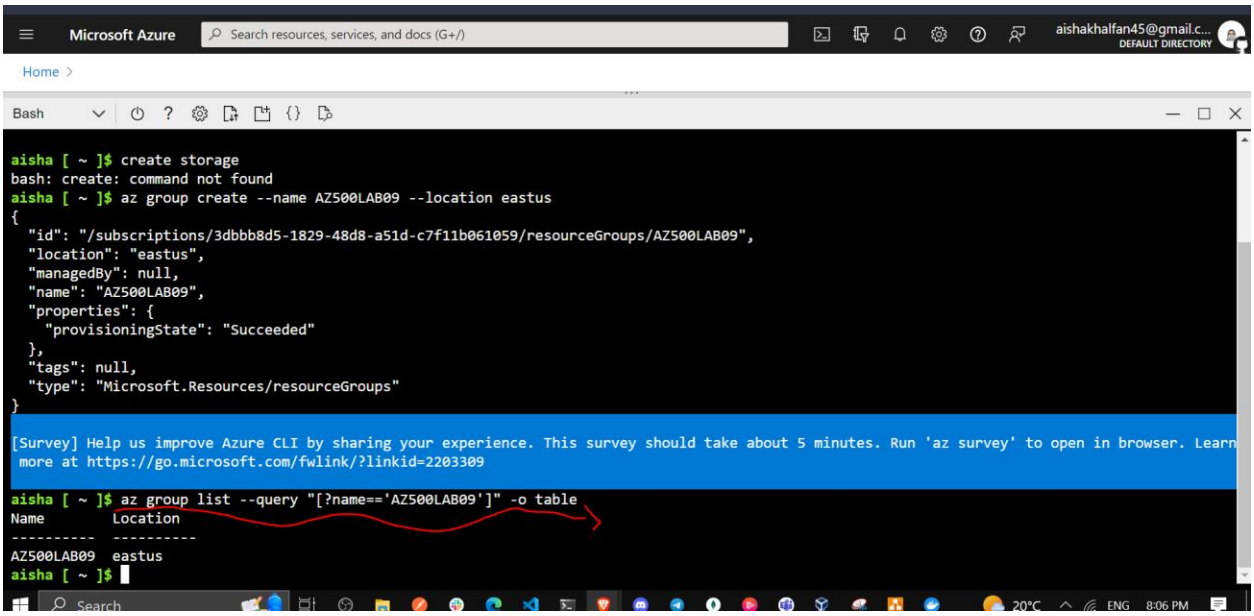
4. In the Bash session within the Cloud Shell pane, run the following to create a new resource group for this lab:

   az group create --name AZ500LAB09 --location eastus



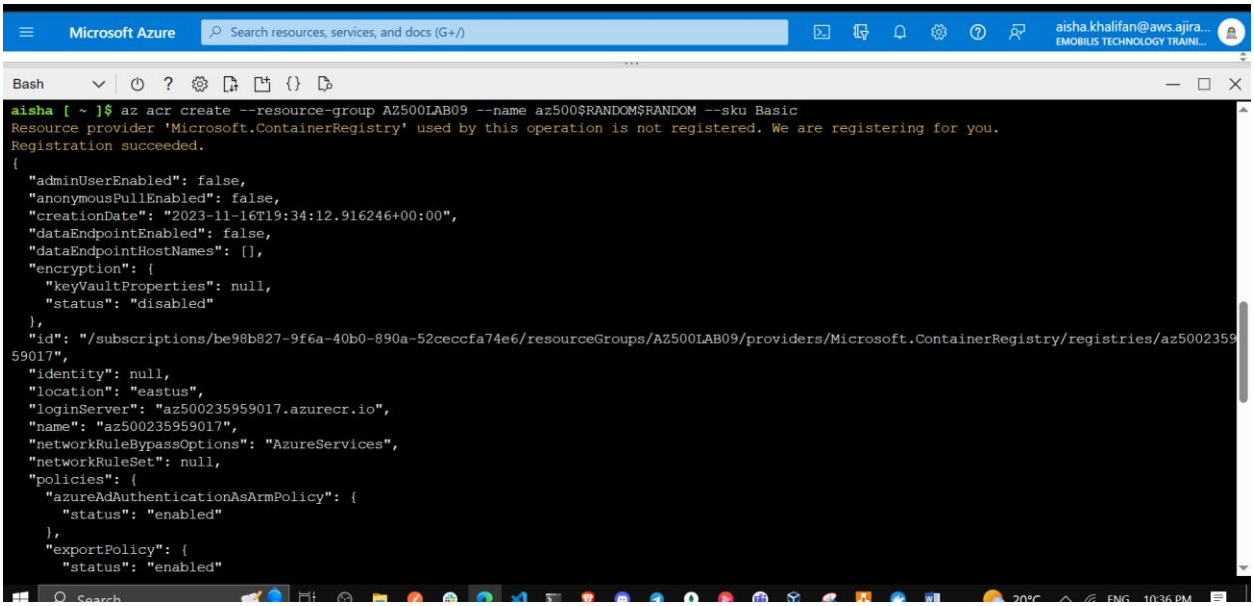
5. In the Bash session within the Cloud Shell pane, run the following to verify the resource group was created:

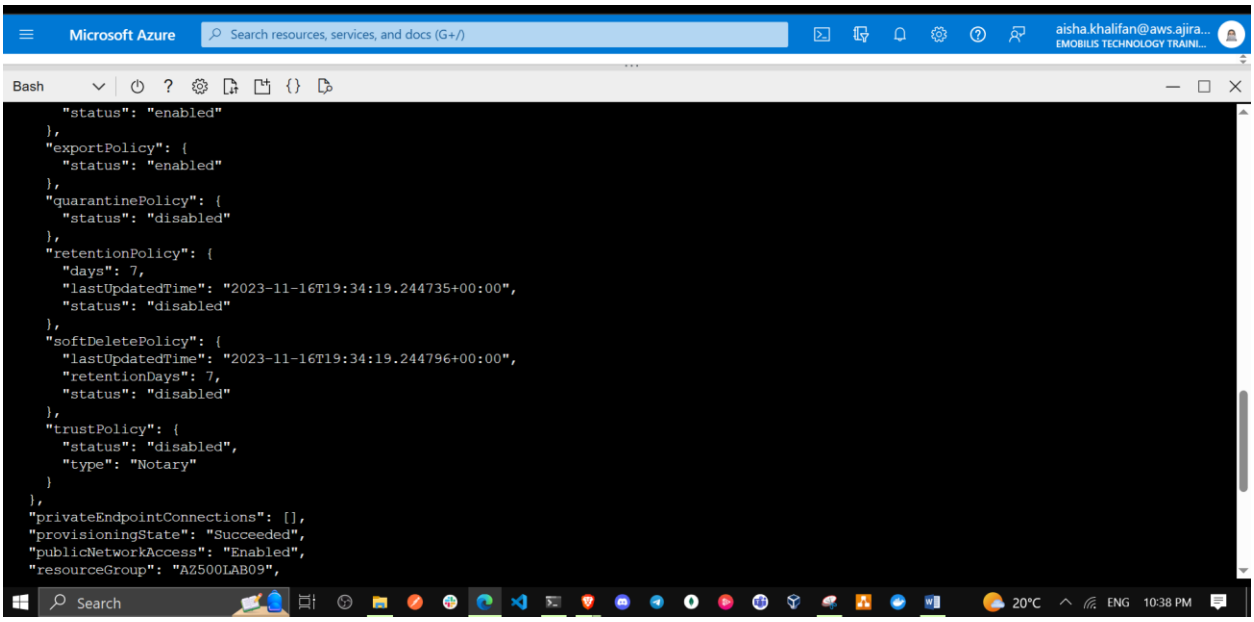
   az group list --query "[?name=='AZ500LAB09']" -o table

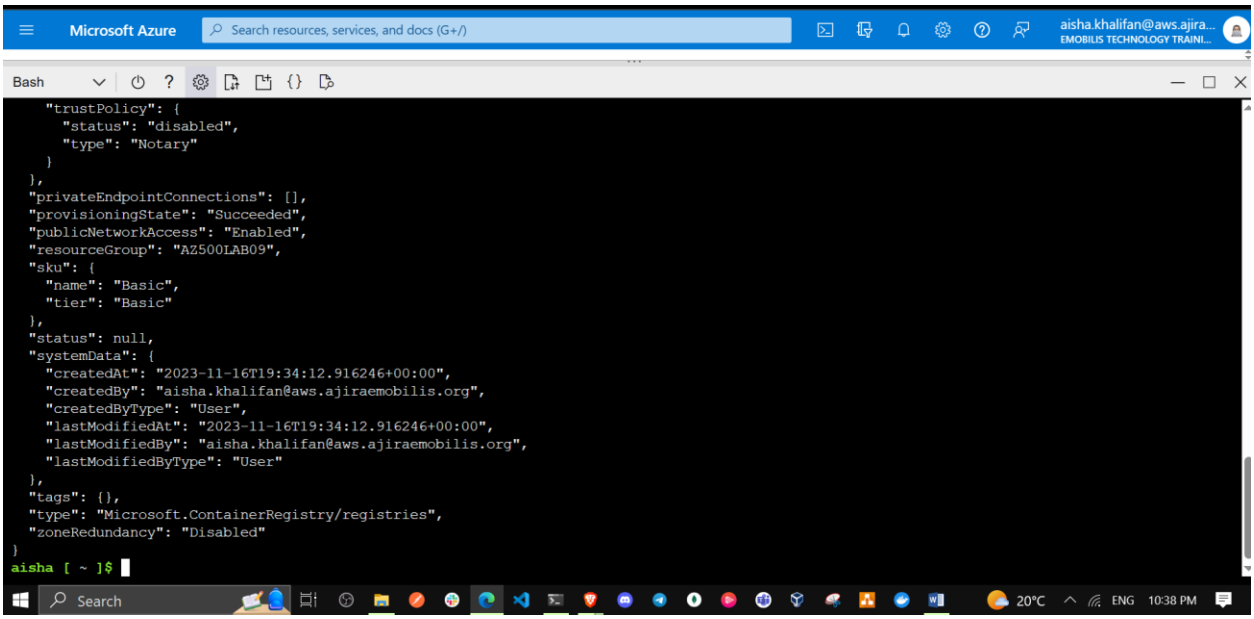


6. In the Bash session within the Cloud Shell pane, run the following to create a new Azure Container Registry (ACR) instance (The name of the ACR must be globally unique):

az acr create --resource-group AZ500LAB09 --name az500$RANDOM$RANDOM --sku Basic



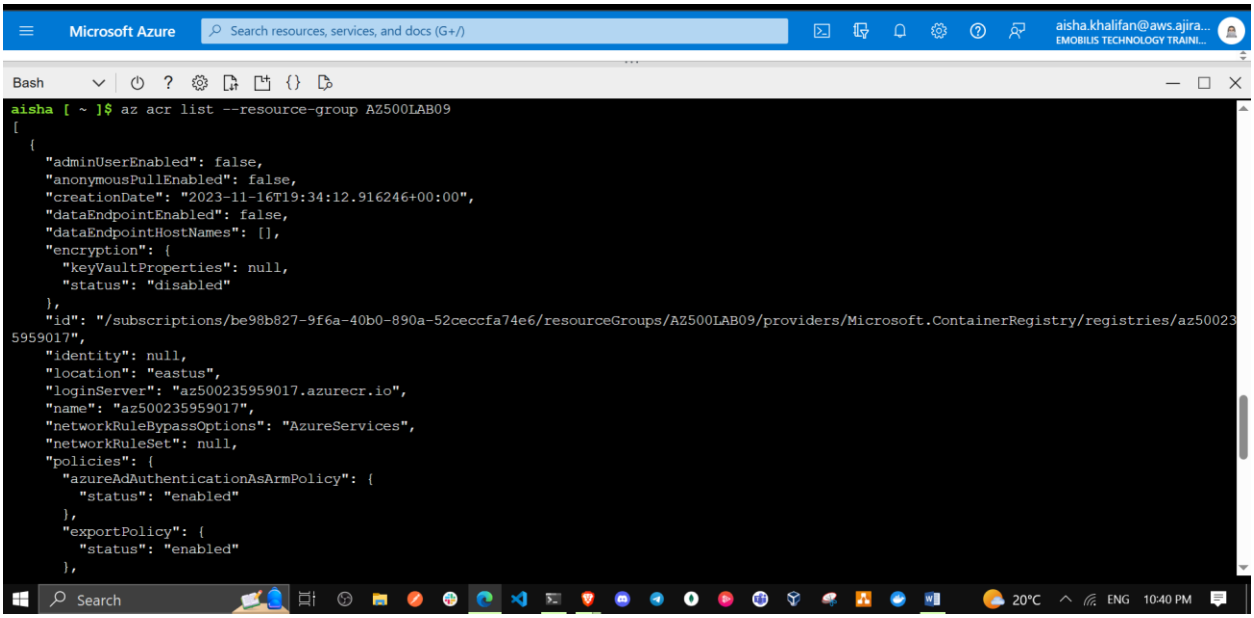


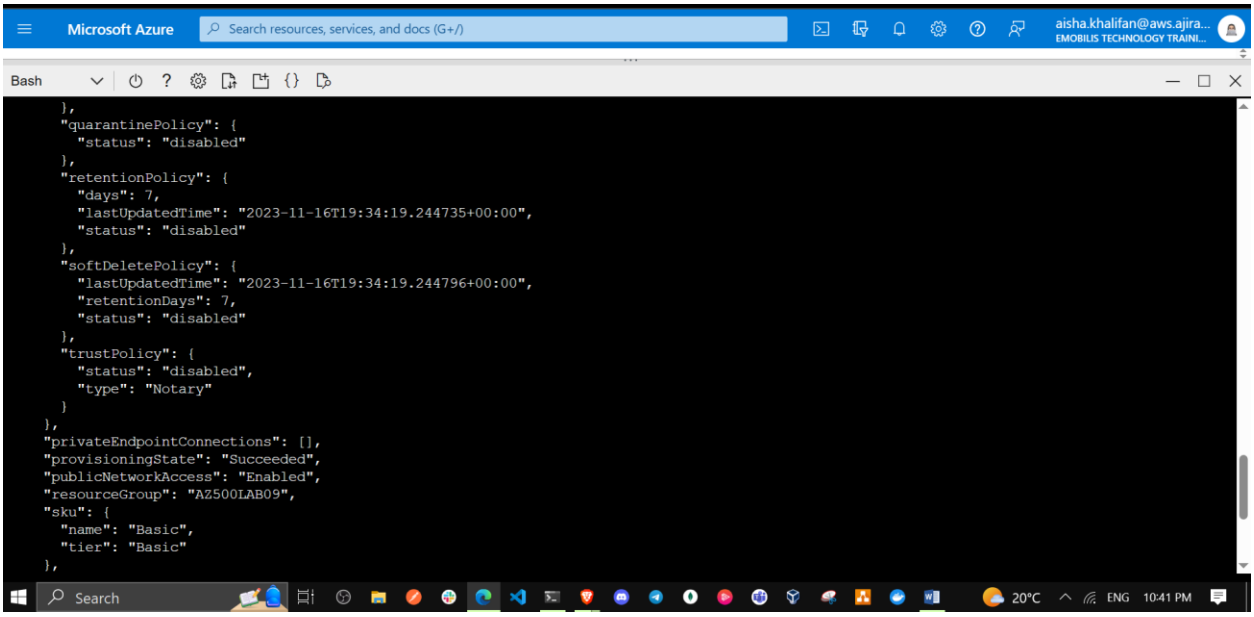


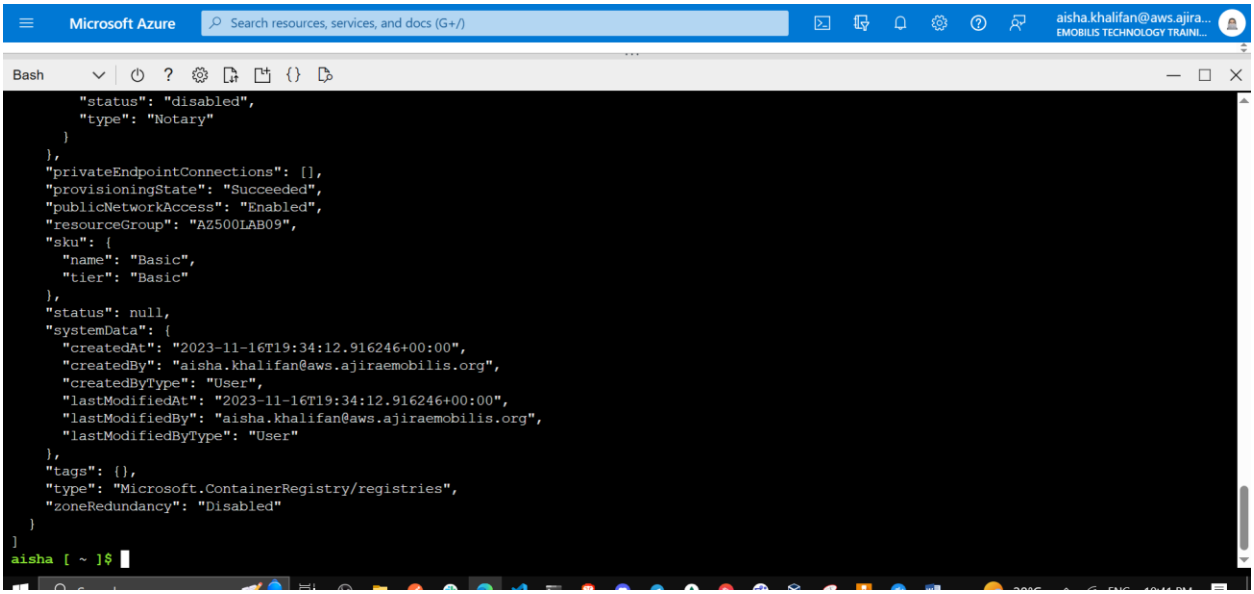
7. In the Bash session within the Cloud Shell pane, run the following to confirm that the new ACR was created:

az acr list --resource-group AZ500LAB09

Note: Record the name of the ACR. You will need it in the next task.


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
aisha [~]$ az acr list --resource-group AZ500LAB09
[
{
"adminUserEnabled": false,
"anonymousPullEnabled": false,
"creationDate": "2023-11-16T19:34:12.916246+00:00",
"dataEndpointEnabled": false,
"dataEndpointHostNames": [],
"encryption": {
"keyVaultProperties": null,
"status": "disabled"
},
"id": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourceGroups/AZ500LAB09/providers/Microsoft.ContainerRegistry/registries/az50023
5959017",
"identity": null,
"location": "eastus",
"loginServer": "az500235959017.azurecr.io",
"name": "az500235959017",
"networkRuleBypassOptions": "AzureServices",
"networkRuleSet": null,
"policies": {
"azureADAuthenticationAsArmPolicy": {
"status": "enabled"
},
"exportPolicy": {
"status": "enabled"
},
Search
20°C
ENG
10:40 PM


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
},
"quarantinePolicy": {
"status": "disabled"
},
"retentionPolicy": {
"days": 7,
"lastUpdatedTime": "2023-11-16T19:34:19.244735+00:00",
"status": "disabled"
},
"softDeletePolicy": {
"lastUpdatedTime": "2023-11-16T19:34:19.244796+00:00",
"retentionDays": 7,
"status": "disabled"
},
"trustPolicy": {
"status": "disabled",
"type": "Notary"
}
},
"privateEndpointConnections": [],
"provisioningState": "Succeeded",
"publicNetworkAccess": "Enabled",
"resourceGroup": "AZ500LAB09",
"sku": {
"name": "Basic",
"tier": "Basic"
},
Search
20°C
ENG
10:41 PM



Task 2: Create a Dockerfile, build a container and push it to Azure Container Registry

In this task, you will create a Dockerfile, build an image from the Dockerfile, and deploy the image to the ACR.

1. In the Bash session within the Cloud Shell pane, run the following to create a Dockerfile to create an Nginx-based image:

   echo FROM nginx > Dockerfile

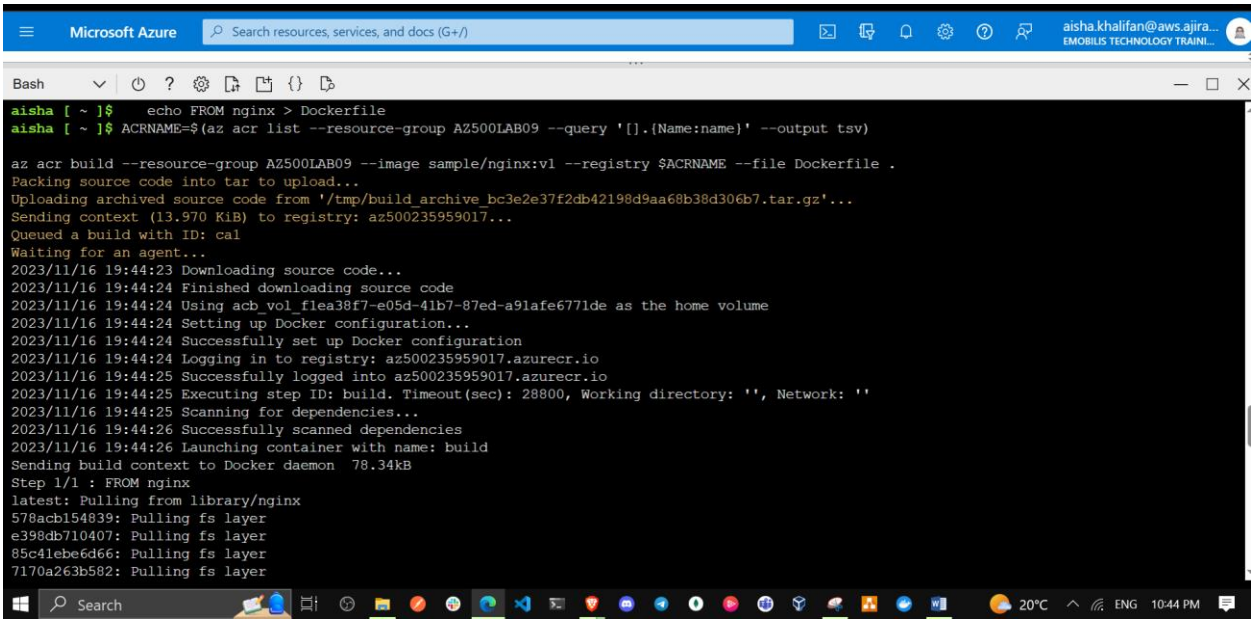
2. In the Bash session within the Cloud Shell pane, run the following to build an image from the Dockerfile and push the image to the new ACR.

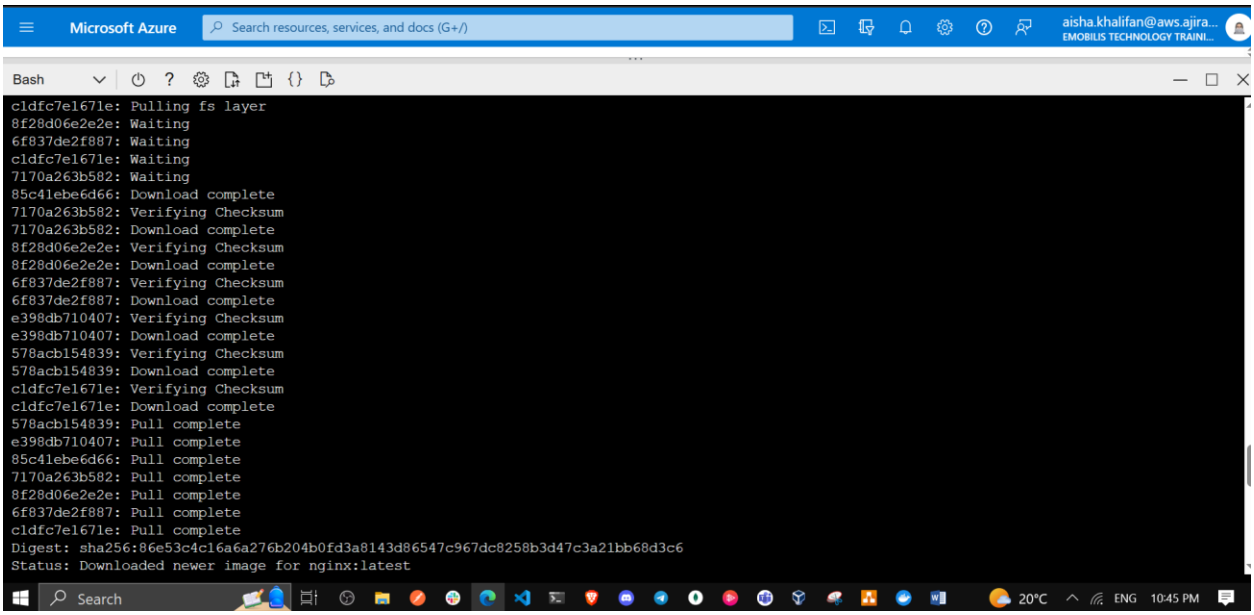
Note: The trailing period at the end of the command line is required. It designates the current directory as the location of Dockerfile.

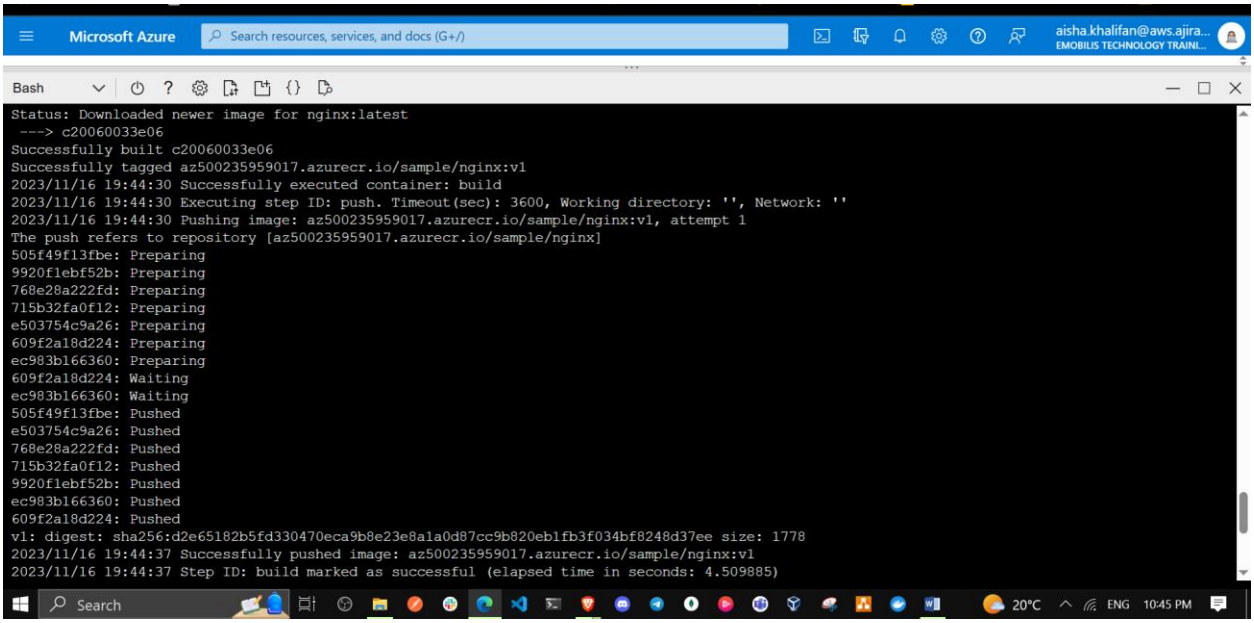
ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)

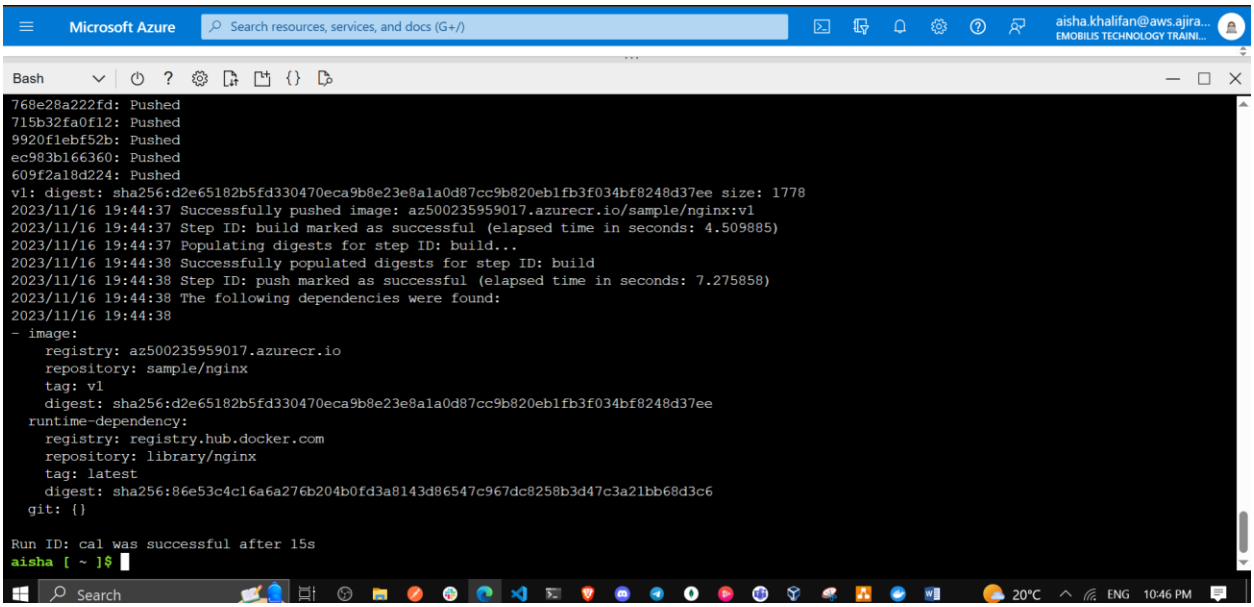
az acr build --resource-group AZ500LAB09 --image sample/nginx:v1 --registry $ACRNAME --file Dockerfile .

Note: Wait for the command to successfully complete. This might take about 2 minutes.


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
aisha [~]$ echo FROM nginx > Dockerfile
aisha [~]$ ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)
az acr build --resource-group AZ500LAB09 --image sample/nginx:v1 --registry $ACRNAME --file Dockerfile .
Packing source code into tar to upload...
Uploading archived source code from '/tmp/build_archive_bc3e2e37f2db42198d9aa68b38d306b7.tar.gz'...
Sending context (13.970 KiB) to registry: az500235959017...
Queued a build with ID: ca1
Waiting for an agent...
2023/11/16 19:44:23 Downloading source code...
2023/11/16 19:44:24 Finished downloading source code
2023/11/16 19:44:24 Using acb_vol_f1ea38f7-e05d-41b7-87ed-a91afe6771de as the home volume
2023/11/16 19:44:24 Setting up Docker configuration...
2023/11/16 19:44:24 Successfully set up Docker configuration
2023/11/16 19:44:24 Logging in to registry: az500235959017.azurecr.io
2023/11/16 19:44:25 Successfully logged into az500235959017.azurecr.io
2023/11/16 19:44:25 Executing step ID: build. Timeout(sec): 28800, Working directory: '', Network: ''
2023/11/16 19:44:25 Scanning for dependencies...
2023/11/16 19:44:26 Successfully scanned dependencies
2023/11/16 19:44:26 Launching container with name: build
Sending build context to Docker daemon 78.34kB
Step 1/1 : FROM nginx
latest: Pulling from library/nginx
578acb154839: Pulling fs layer
e398db710407: Pulling fs layer
85c41ebe6d66: Pulling fs layer
7170a263b582: Pulling fs layer
Search
20°C
ENG
10:44 PM

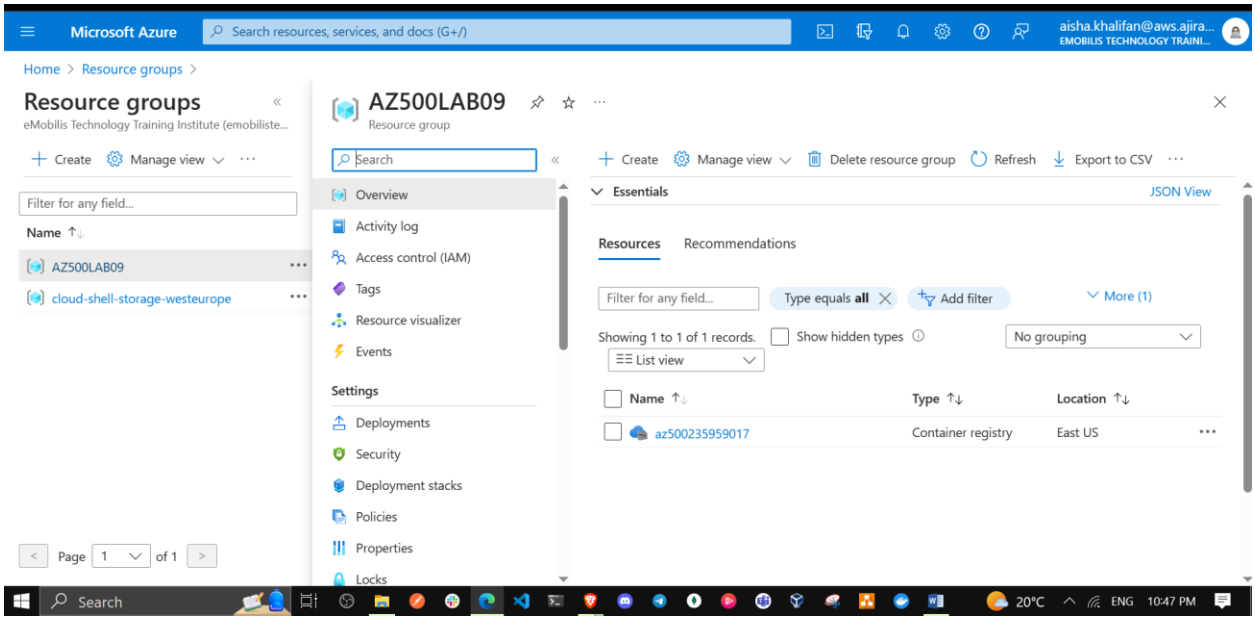

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
c1dfc7e1671e: Pulling fs layer
8f28d06e2e2e: Waiting
6f837de2f887: Waiting
c1dfc7e1671e: Waiting
7170a263b582: Waiting
85c41ebe6d66: Download complete
7170a263b582: Verifying Checksum
7170a263b582: Download complete
8f28d06e2e2e: Verifying Checksum
8f28d06e2e2e: Download complete
6f837de2f887: Verifying Checksum
6f837de2f887: Download complete
e398db710407: Verifying Checksum
e398db710407: Download complete
578acb154839: Verifying Checksum
578acb154839: Download complete
c1dfc7e1671e: Verifying Checksum
c1dfc7e1671e: Download complete
578acb154839: Pull complete
e398db710407: Pull complete
85c41ebe6d66: Pull complete
7170a263b582: Pull complete
8f28d06e2e2e: Pull complete
6f837de2f887: Pull complete
c1dfc7e1671e: Pull complete
Digest: sha256:86e53c4c16a6a276b204b0fd3a8143d86547c967dc8258b3d47c3a21bb68d3c6
Status: Downloaded newer image for nginx:latest
Search
20°C
ENG
10:45 PM



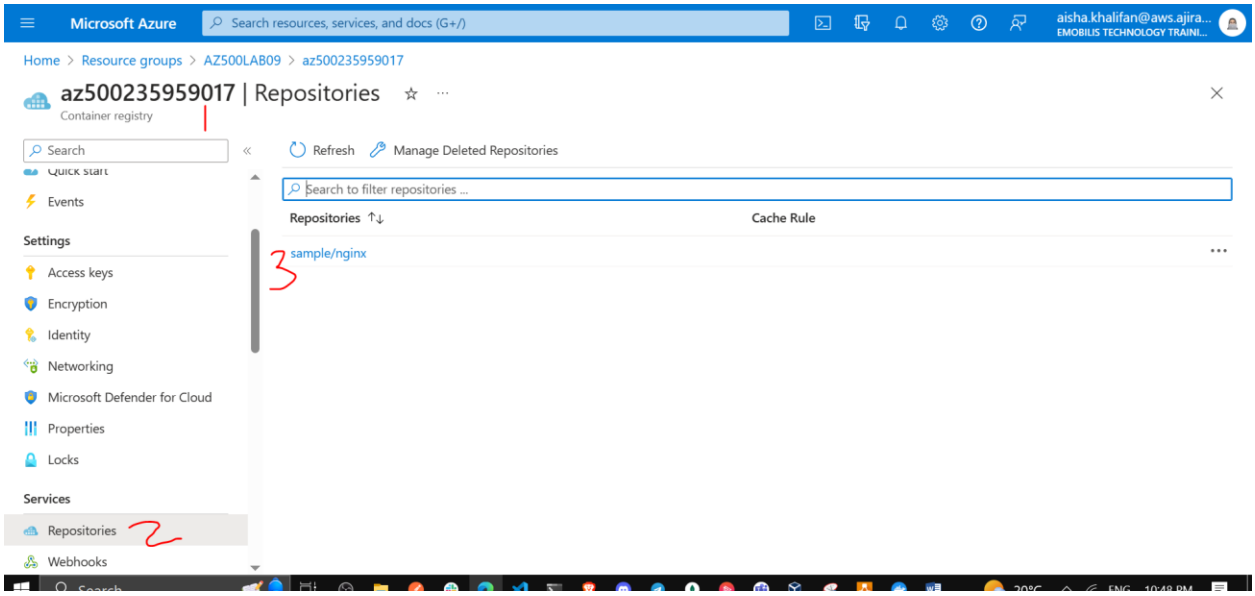


3. Close the Cloud Shell pane.

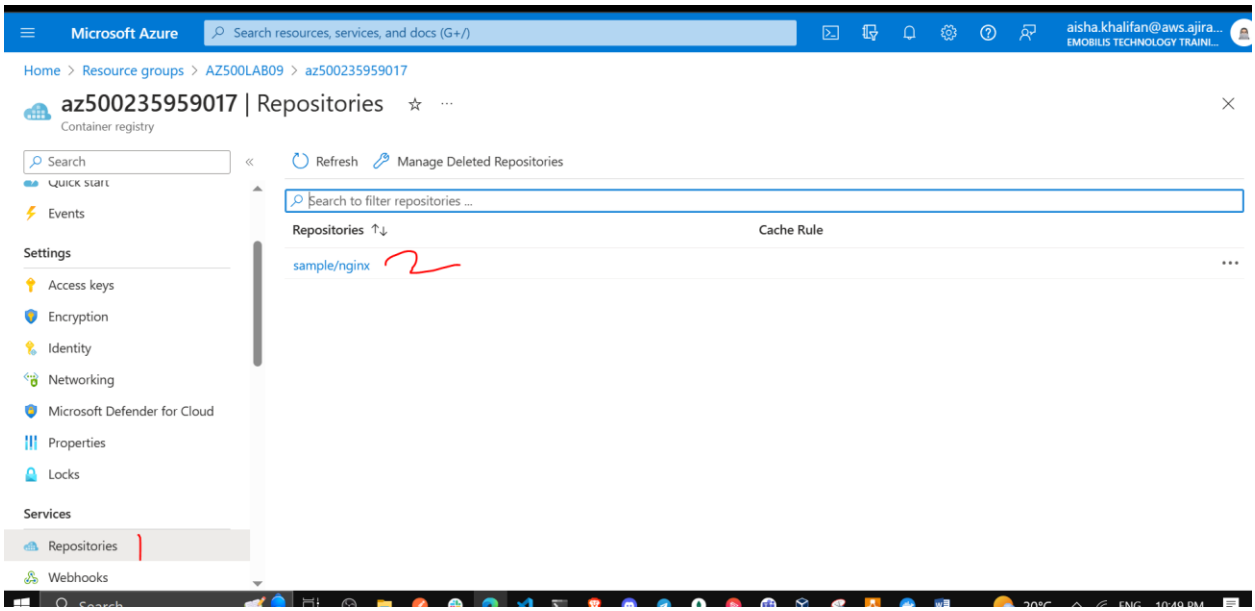
4. In the Azure portal, navigate to the **AZ500Lab09** resource group and, in the list of resources, click the entry representing the Azure Container Registry instance you provisioned in the previous task.



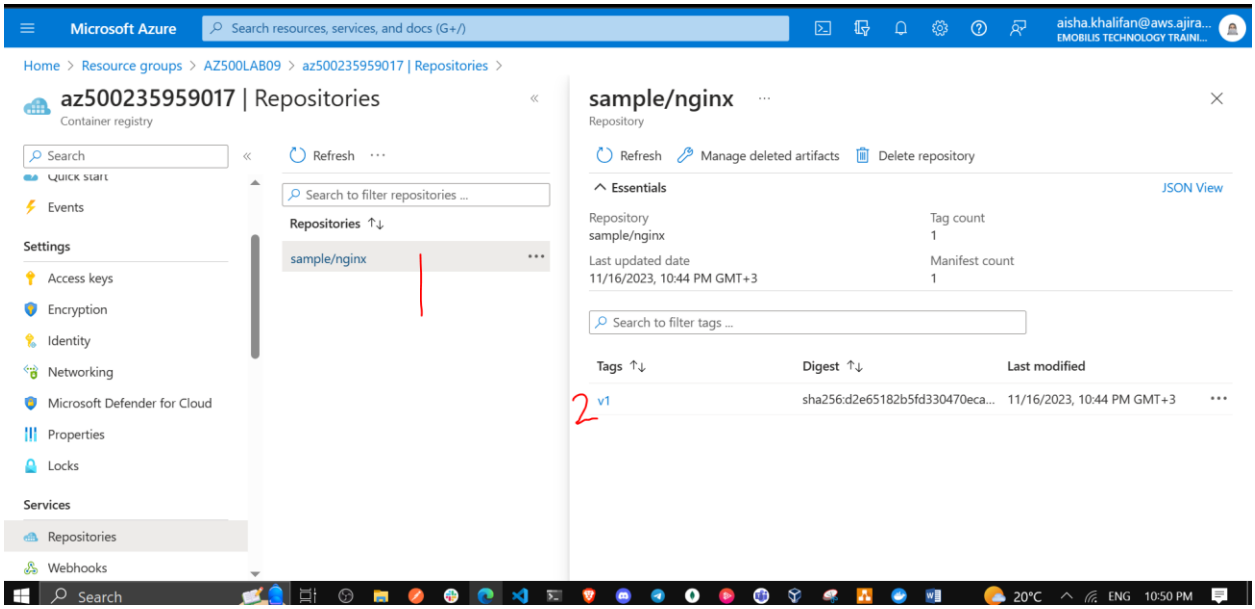
5. On the Container registry blade, in the **Services** section, click **Repositories**.



6. Verify that the list of repositories includes the new container image named **sample/nginx**.

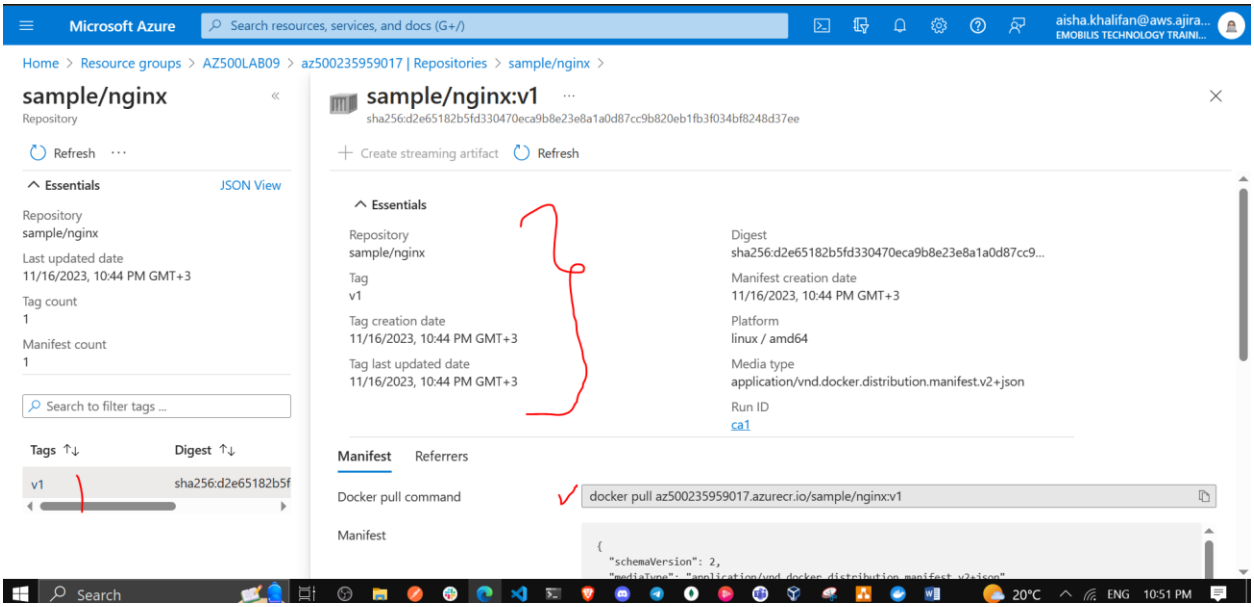


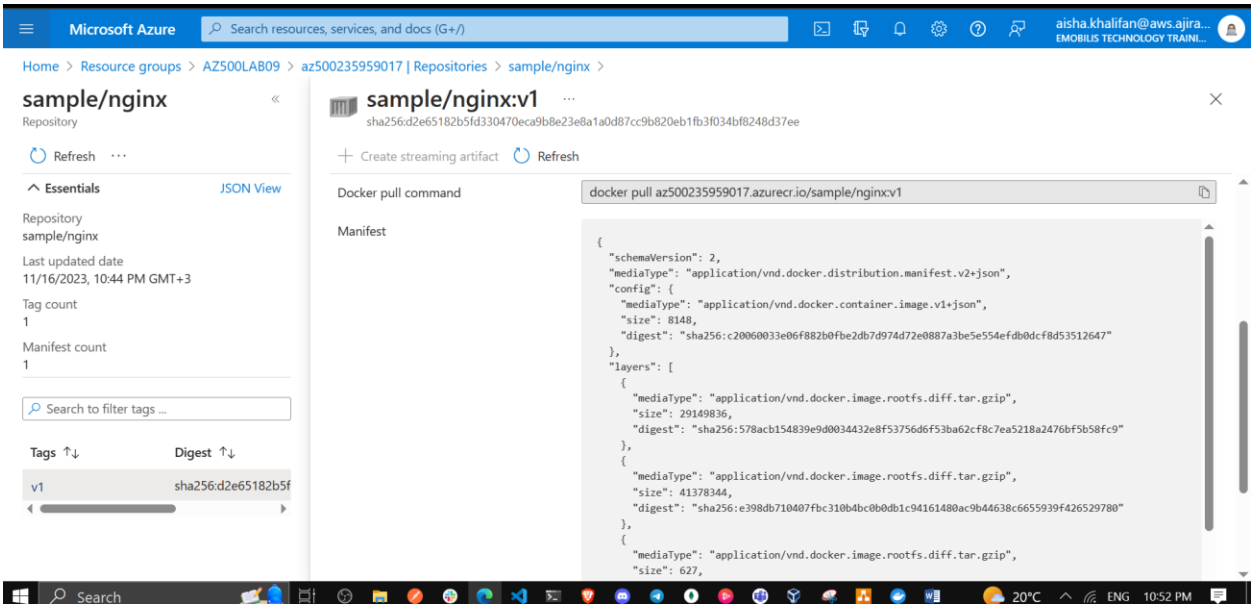
7. Click the **sample/nginx** entry and verify presence of the **v1** tag that identifies the image version.

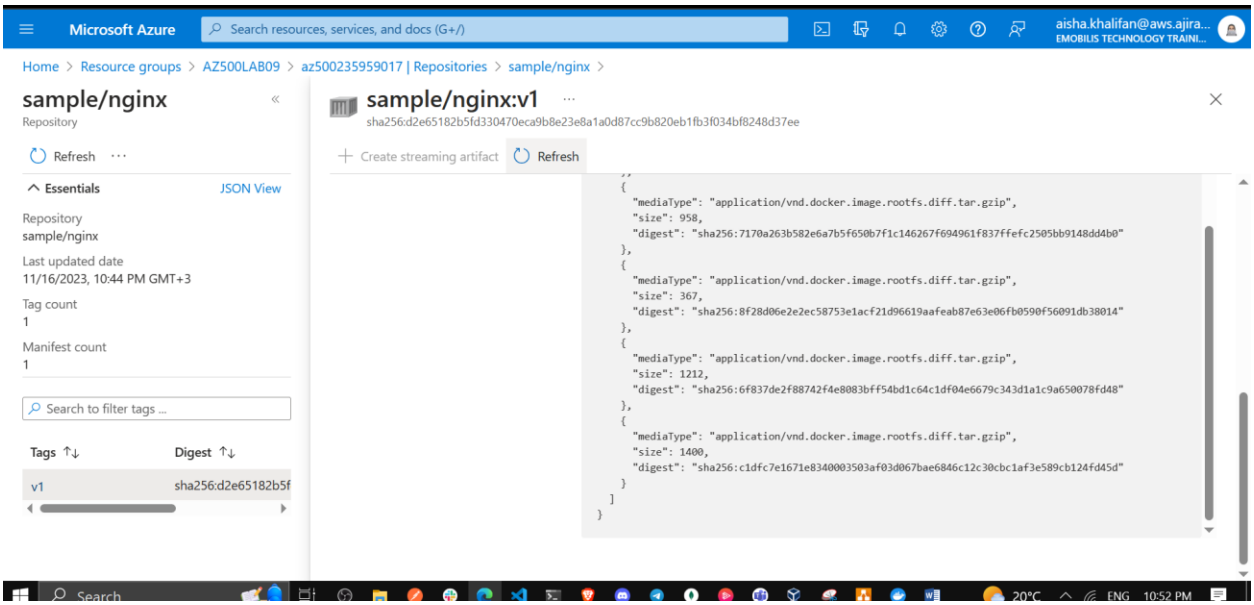


8. Click the **v1** entry to view the image manifest.

   Note: The manifest includes the sha256 digest, manifest creation date, and platform entries.


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups > AZ500LAB09 > az500235959017 | Repositories > sample/nginx >
sample/nginx
Repository
Refresh
Essentials
JSON View
Repository
sample/nginx
Last updated date
11/16/2023, 10:44 PM GMT+3
Tag count
1
Manifest count
1
Search to filter tags ...
Tags
Digest
v1
sha256:d2e65182b5f
sample/nginx:v1
sha256:d2e65182b5fd330470eca9b8e23e8a1a0d87cc9b820eb1fb3f034bf8248d37ee
Create streaming artifact
Refresh
Essentials
Repository
sample/nginx
Tag
v1
Tag creation date
11/16/2023, 10:44 PM GMT+3
Tag last updated date
11/16/2023, 10:44 PM GMT+3
Digest
sha256:d2e65182b5fd330470eca9b8e23e8a1a0d87cc9...
Manifest creation date
11/16/2023, 10:44 PM GMT+3
Platform
linux / amd64
Media type
application/vnd.docker.distribution.manifest.v2+json
Run ID
ca1
Manifest
Referrers
Docker pull command
docker pull az500235959017.azurecr.io/sample/nginx:v1
Manifest
{
"schemaVersion": 2,
Search
20°C
ENG
10:51 PM

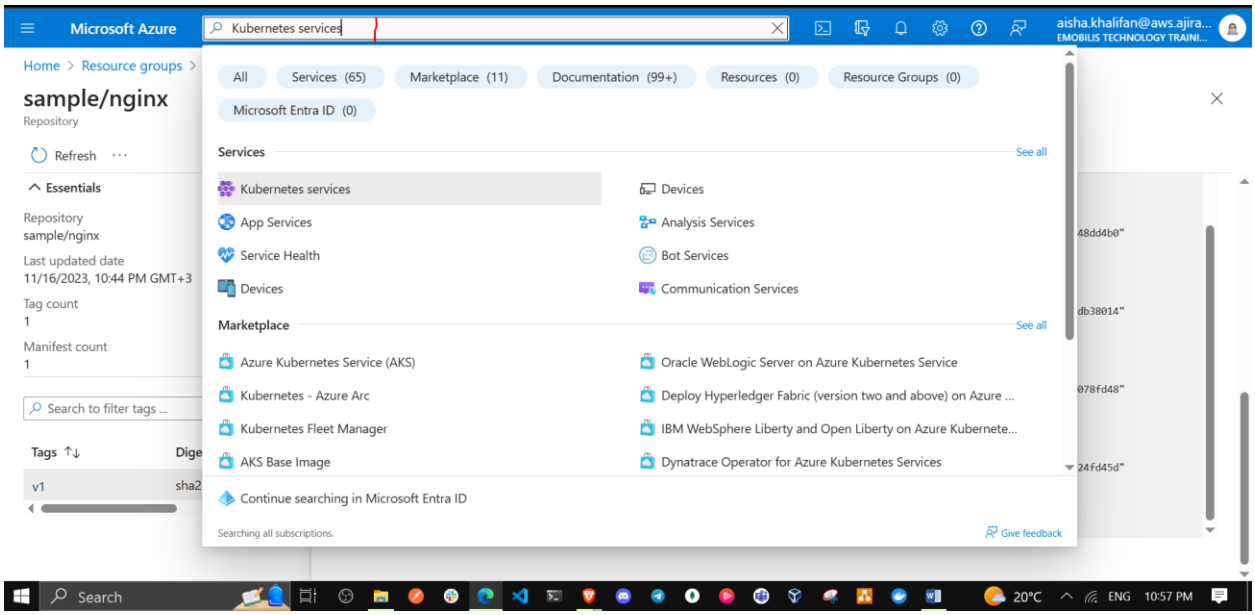

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups > AZ500LAB09 > az500235959017 | Repositories > sample/nginx >
sample/nginx
Repository
Refresh
Essentials
JSON View
Repository
sample/nginx
Last updated date
11/16/2023, 10:44 PM GMT+3
Tag count
1
Manifest count
1
Search to filter tags ...
Tags
Digest
v1
sha256:d2e65182b5f
sample/nginx:v1
sha256:d2e65182b5fd330470eca9b8e23e8a1a0d87cc9b820eb1fb3f034bf8248d37ee
Create streaming artifact
Refresh
Docker pull command
docker pull az500235959017.azurecr.io/sample/nginx:v1
Manifest
{
"schemaVersion": 2,
"mediaType": "application/vnd.docker.distribution.manifest.v2+json",
"config": {
"mediaType": "application/vnd.docker.container.image.v1+json",
"size": 8148,
"digest": "sha256:c20060033e06f882b0fbe2db7d974d72e0887a3be5e554efdb0dcf8d53512647"
},
"layers": [
{
"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
"size": 29149836,
"digest": "sha256:578acb154839e9d0034432e8f53756d6f53ba62cf8c7ea5218a2476bf5b58fc9"
},
{
"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
"size": 41378344,
"digest": "sha256:e398db710407fbc310b4bc0b0db1c94161480ac9b44638c6655939f426529780"
},
{
"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
"size": 627,
Search
20°C
ENG
10:52 PM



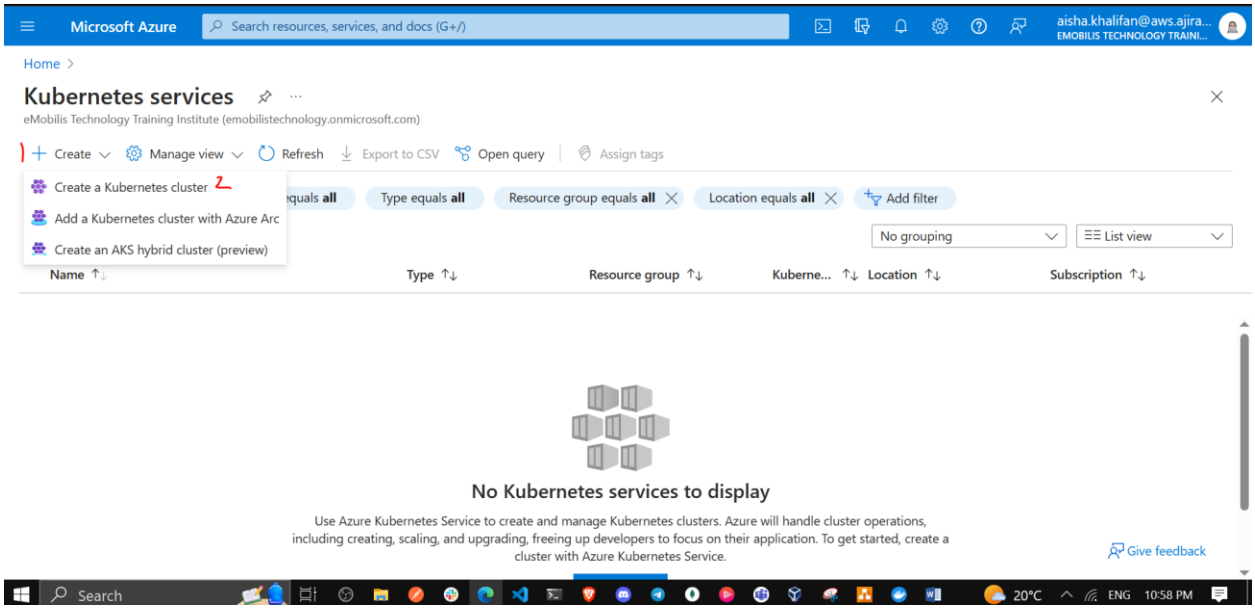
Task 3: Create an Azure Kubernetes Service cluster

In this task, you will create an Azure Kubernetes service and review the deployed resources.

1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **Kubernetes services** and press the **Enter** key.

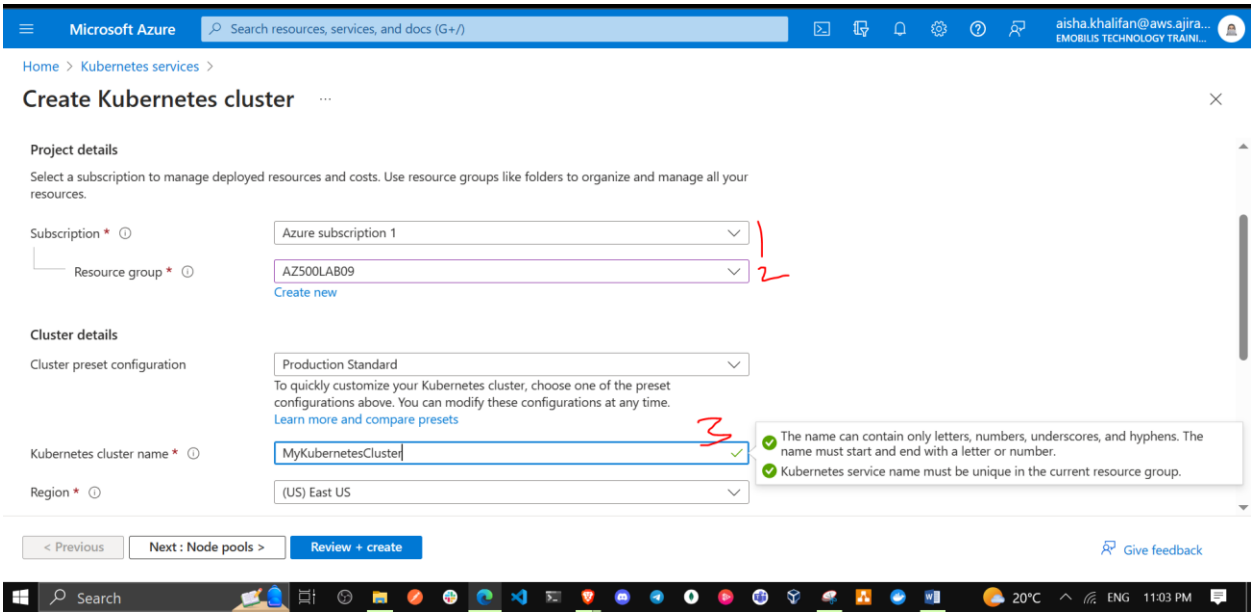


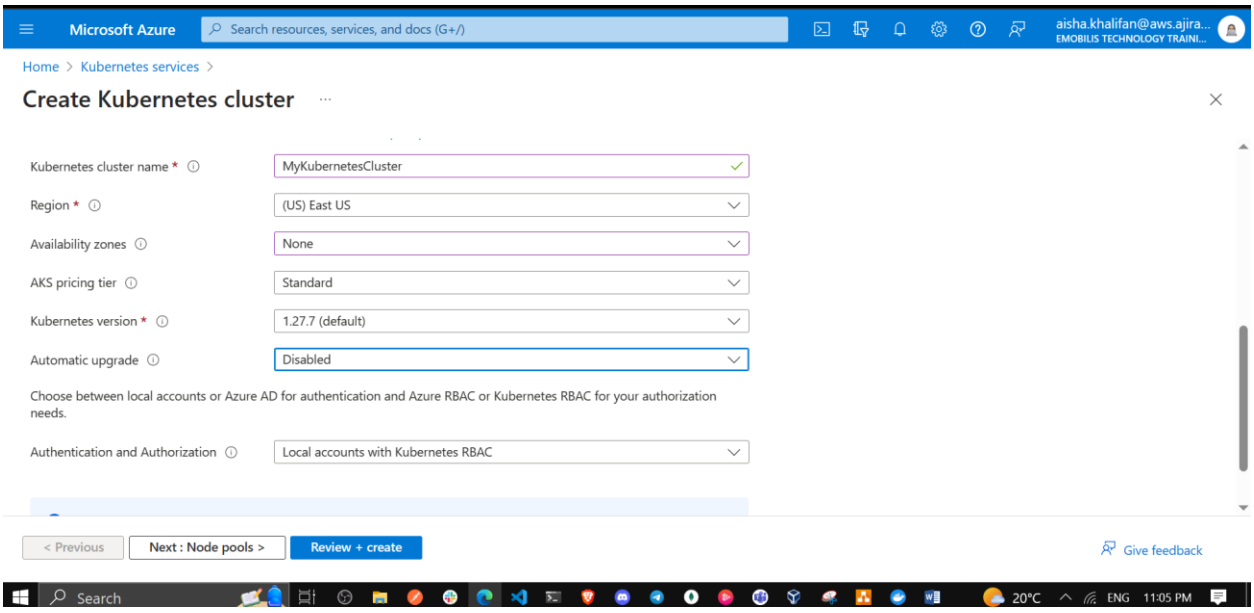
2. On the **Kubernetes services** blade, click **+ Create** and, in the drop-down menu, click **+ Create a Kubernetes cluster**



3. On the **Basics** tab of the **Create Kubernetes cluster** blade, select **Cluster preset configuration**, select **Dev/Test ($)**. Now specify the following settings (leave others with their default values):

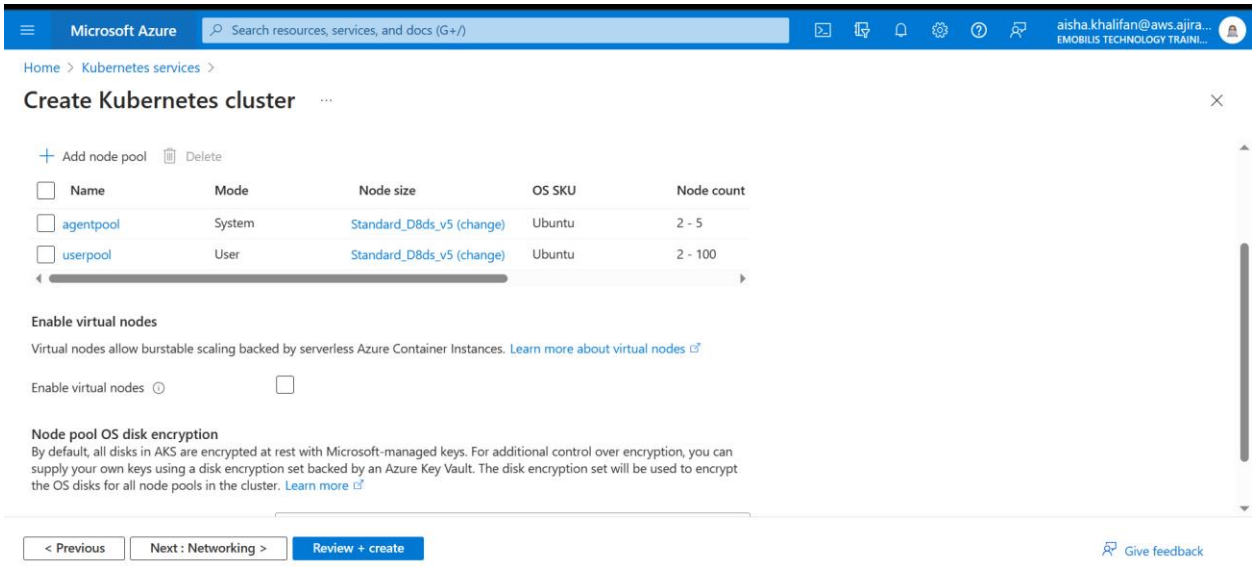
Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**AZ500LAB09**
Kubernetes cluster name	**MyKubernetesCluster**
Region	**(US) East US**
Availability zones	**None**
Scale method	**Manual**
Node count	**1**



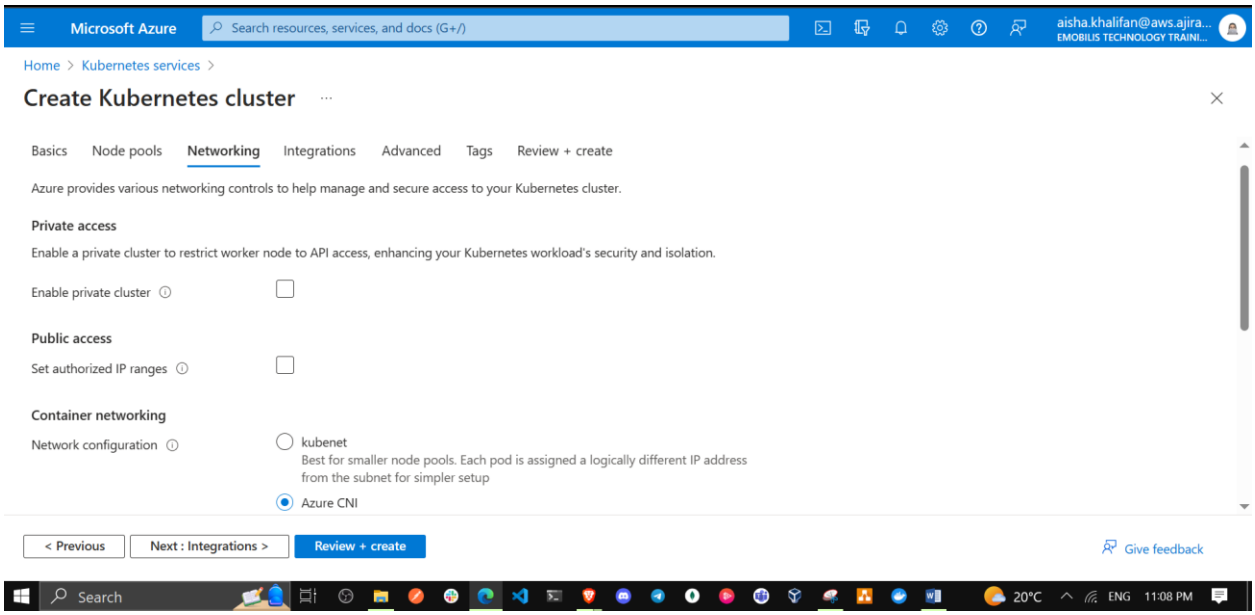


4. Click **Next: Node Pools >** and, on the **Node Pools** tab of the **Create Kubernetes cluster** blade, specify the following settings (leave others with their default values):

Setting	Value
Enable virtual nodes	cleared checkbox



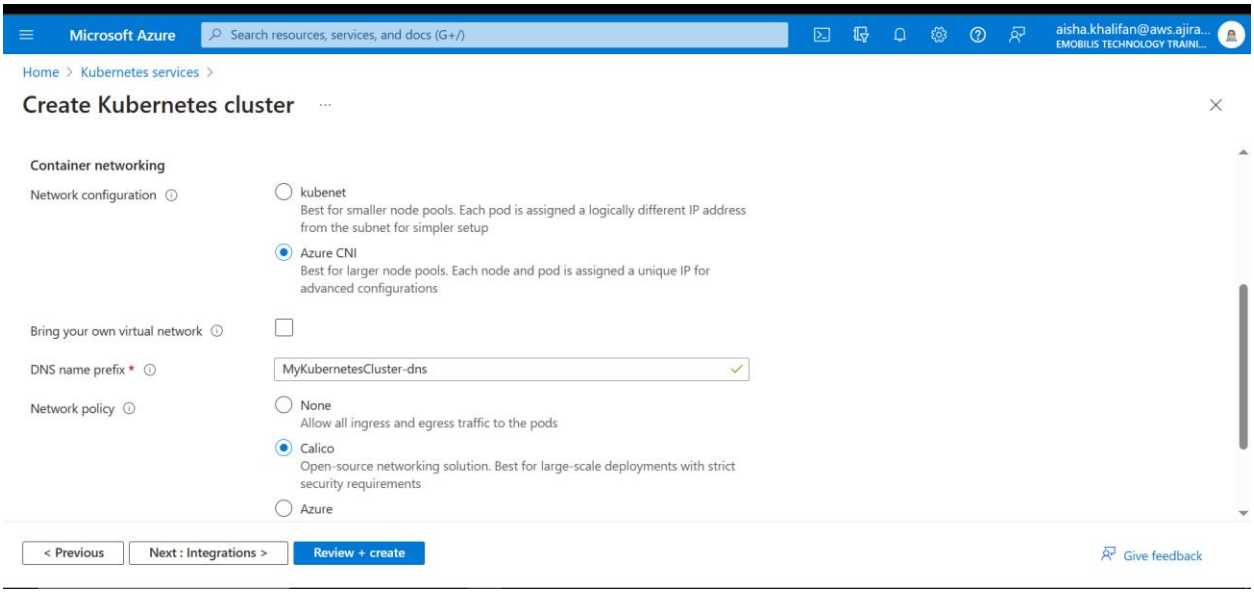
5. Click **Next: Access >**, on the **Access** tab of the **Create Kubernetes cluster** blade, accept the defaults, and click **Next: Networking >**.



6. On the **Networking** tab of the **Create Kubernetes cluster** blade, specify the following settings (leave others with their default values):

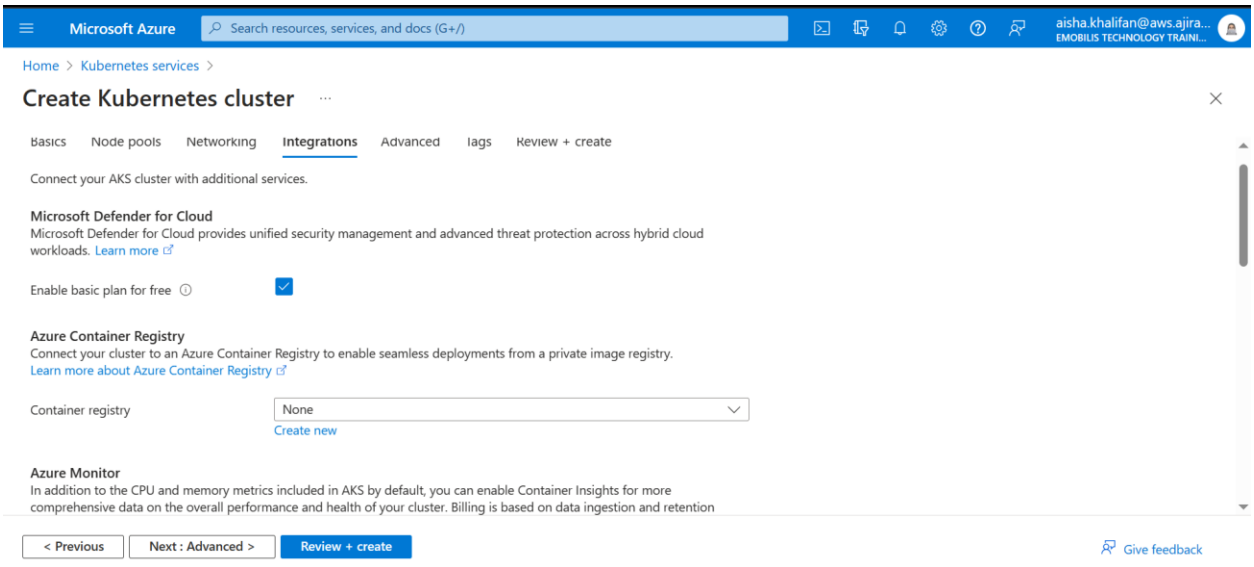
Setting	Value
Network configuration	**Azure CNI**
DNS name prefix	**Leave the default value**

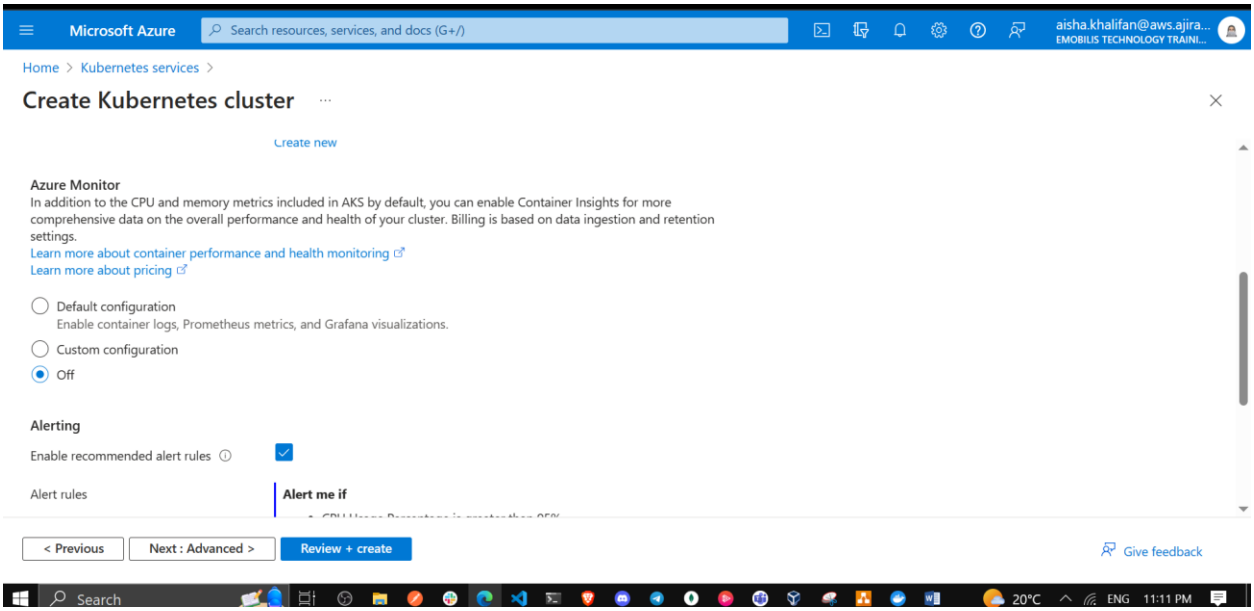
Note: AKS can be configured as a private cluster. This assigns a private IP to the API server to ensure network traffic between your API server and your node pools remains on the private network only. For more information, visit Create a private Azure Kubernetes Service cluster page.



7. Click **Next: Integrations >** and, on the **Integrations** tab of the **Create Kubernetes cluster** blade, set **Container monitoring** to **Disabled**.

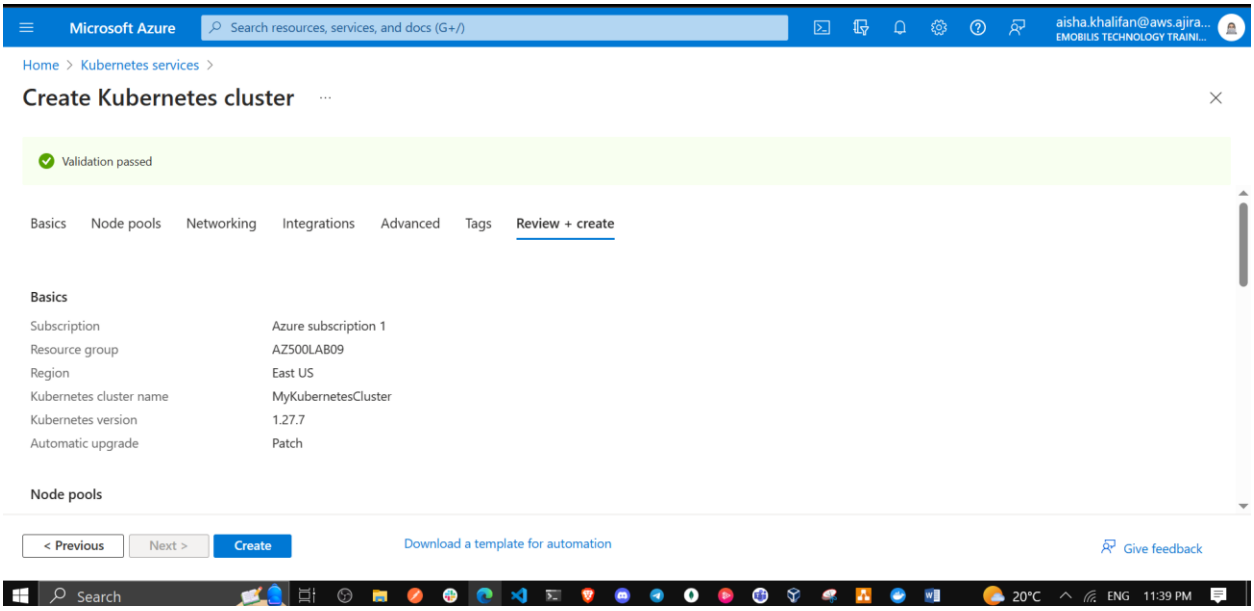
   Note: In production scenarios, you would want to enable monitoring. Monitoring is disabled in this case since it is not covered in the lab.

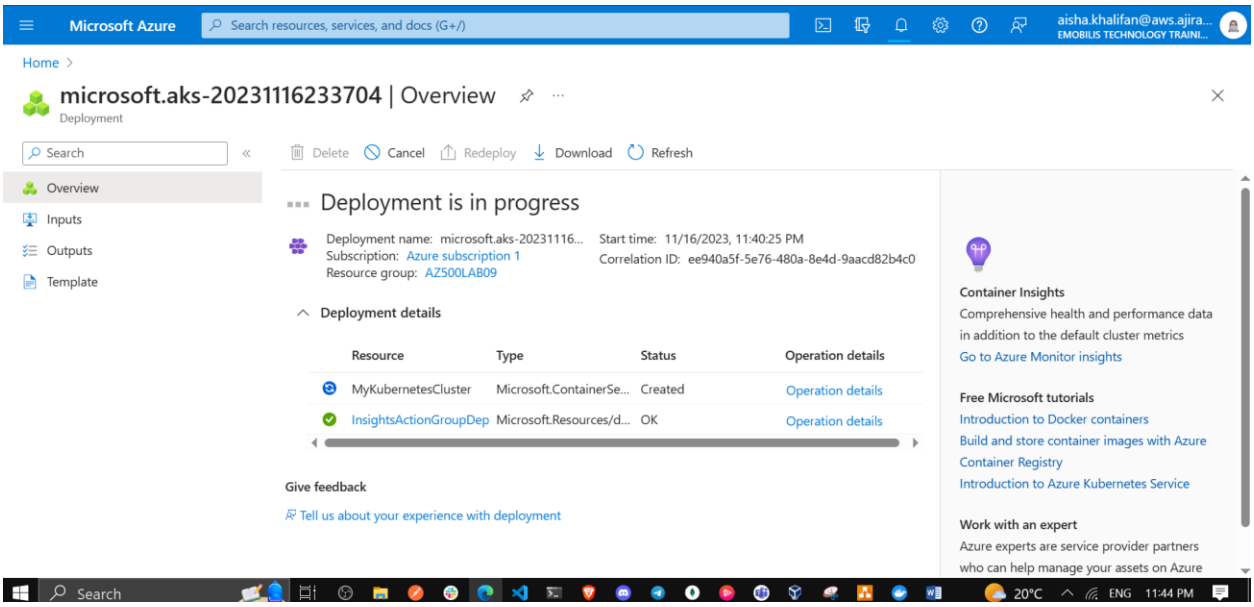




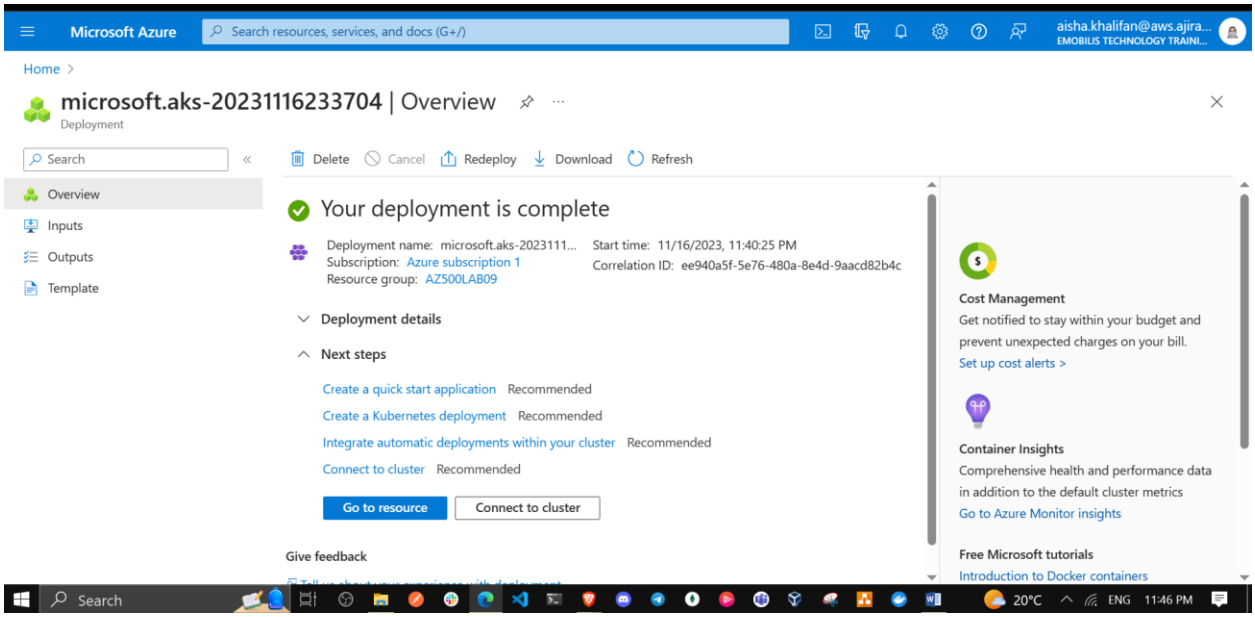
8. Click **Review + Create** and then click **Create**.

   Note: Wait for the deployment to complete. This might take about 10 minutes.

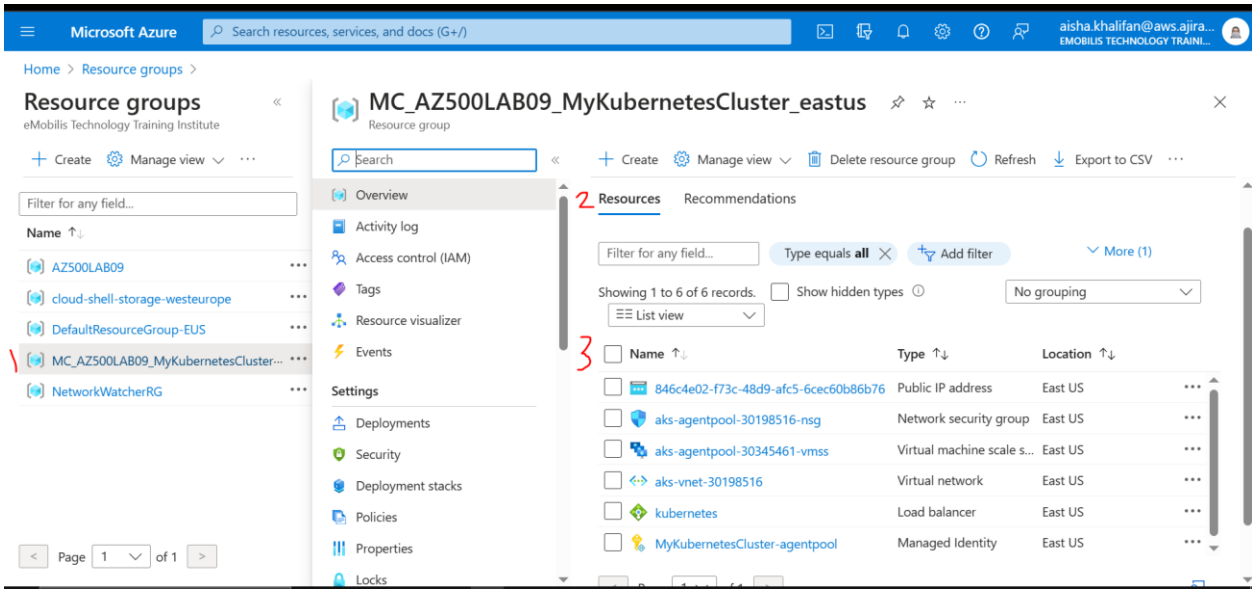




9. Once the deployment completes, in the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **Resource groups** and press the **Enter** key.

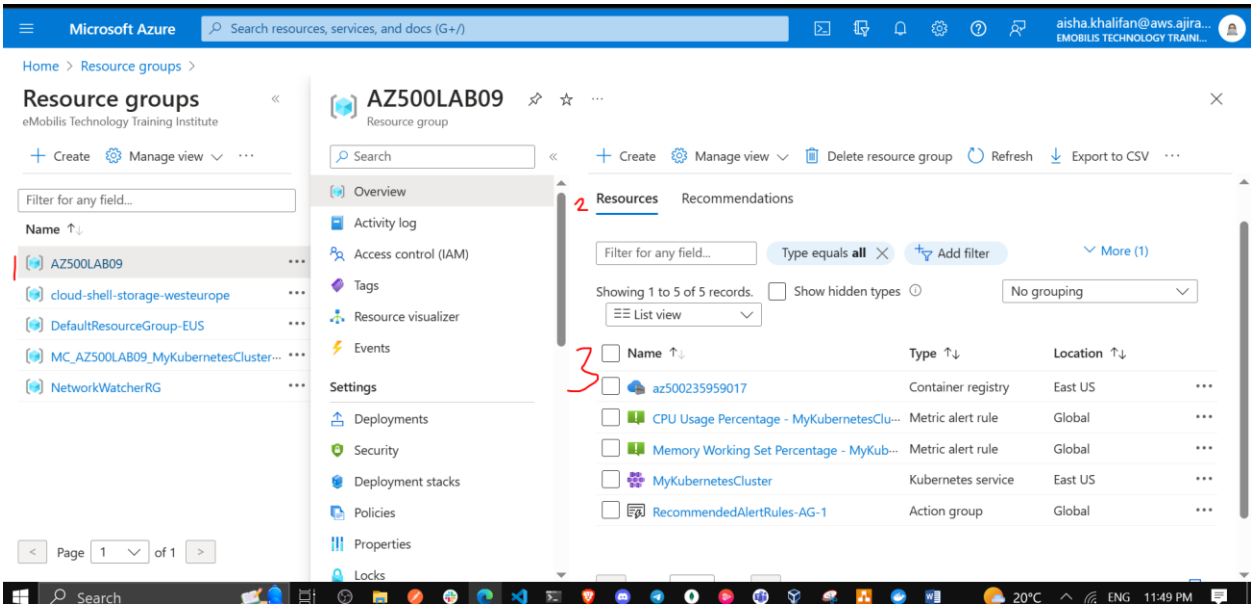


10. On the **Resource groups** blade, in the listing of resource groups, note a new resource group named **MC_AZ500LAB09_MyKubernetesCluster_eastus** that holds components of the AKS Nodes. Review resources in this resource group.



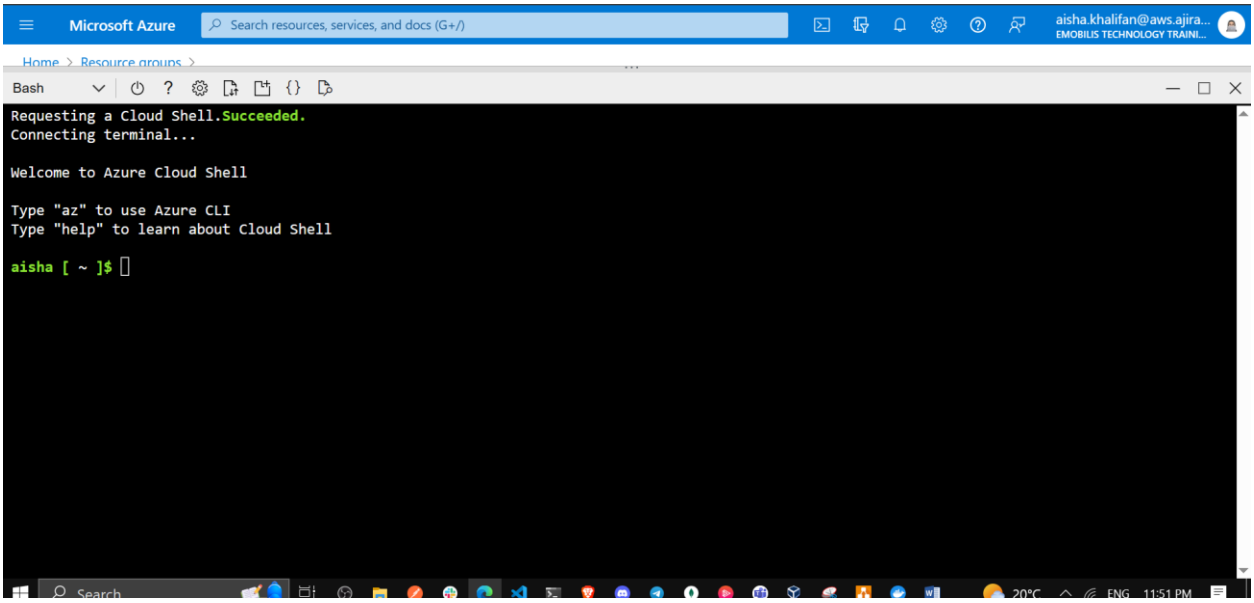
11. Navigate back to the **Resource groups** blade and click the **AZ500LAB09** entry.

    Note: In the list of resources, note the AKS Cluster and the corresponding virtual network.



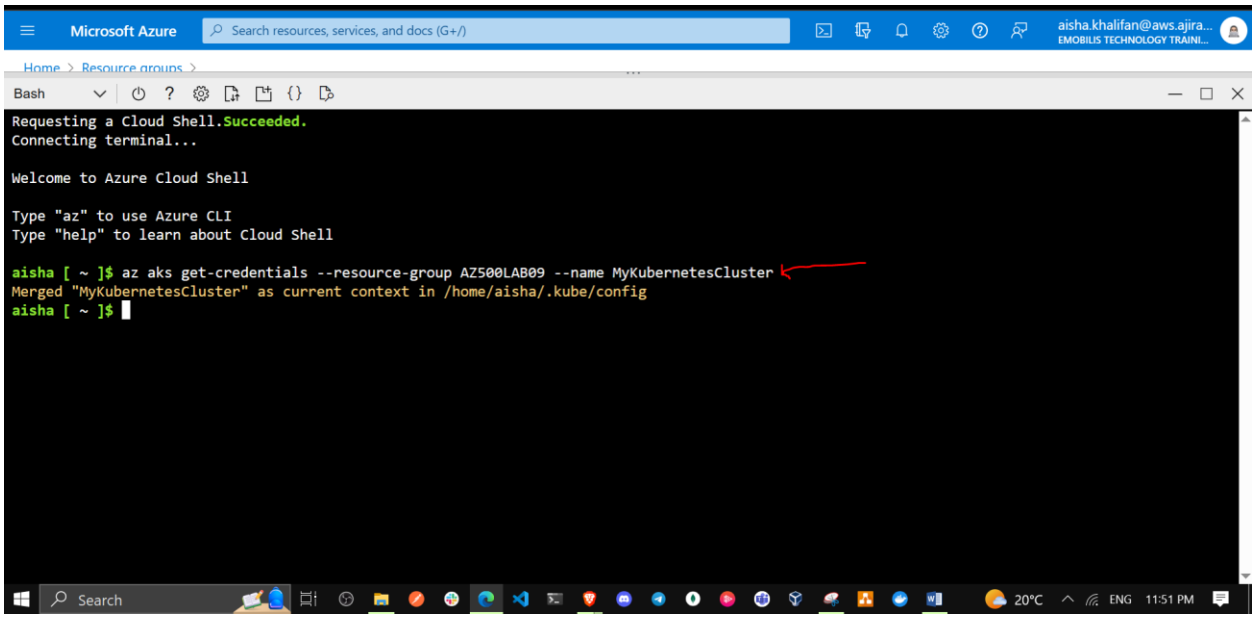
12. In the Azure portal, open a Bash session in the Cloud Shell.

    Note: Ensure **Bash** is selected in the drop-down menu in the upper-left corner of the Cloud Shell pane.



13. In the Bash session within the Cloud Shell pane, run the following to connect to the Kubernetes cluster:

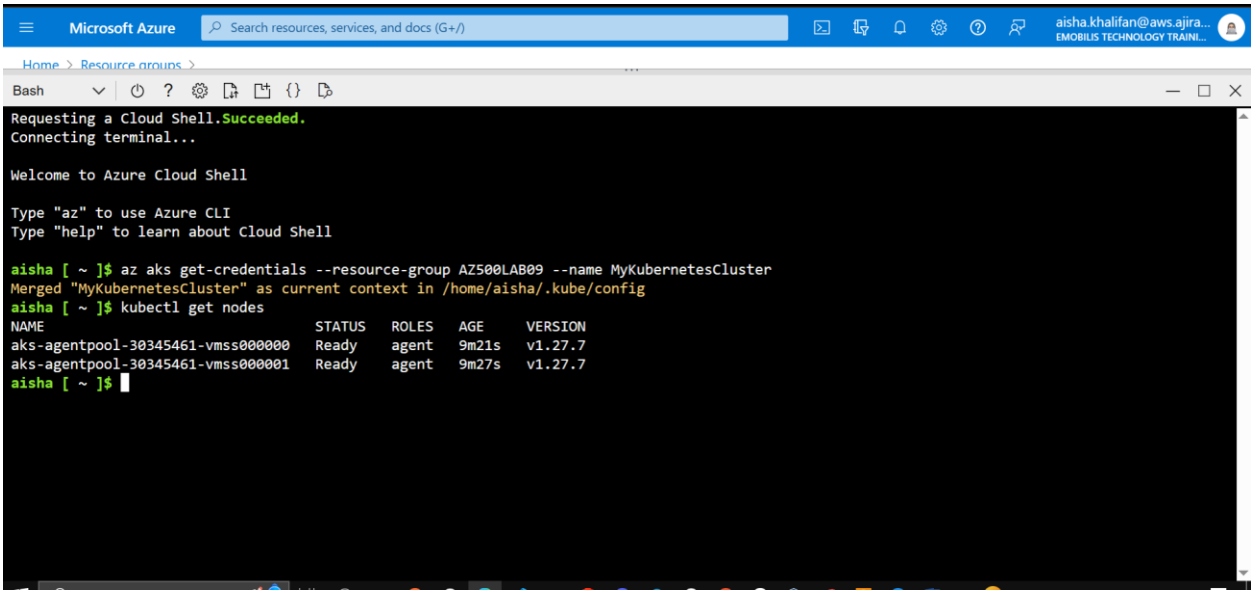
az aks get-credentials --resource-group AZ500LAB09 --name MyKubernetesCluster



14. In the Bash session within the Cloud Shell pane, run the following to list nodes of the Kubenetes cluster:

kubectl get nodes

Note: Verify that the **Status** of the cluster node is listed as **Ready**.



Task 4: Grant the AKS cluster permissions to access the ACR and manage its virtual network

In this task, you will grant the AKS cluster permission to access the ACR and manage its virtual network.

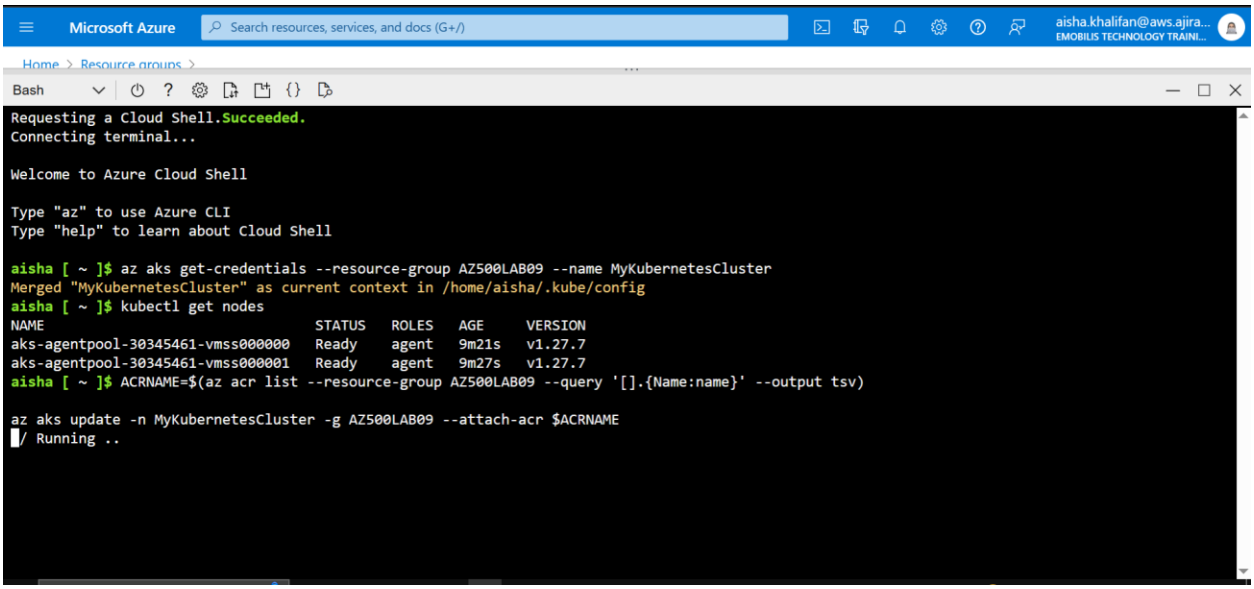
1. In the Bash session within the Cloud Shell pane, run the following to configure the AKS cluster to use the Azure Container Registry instance you created earlier in this lab.

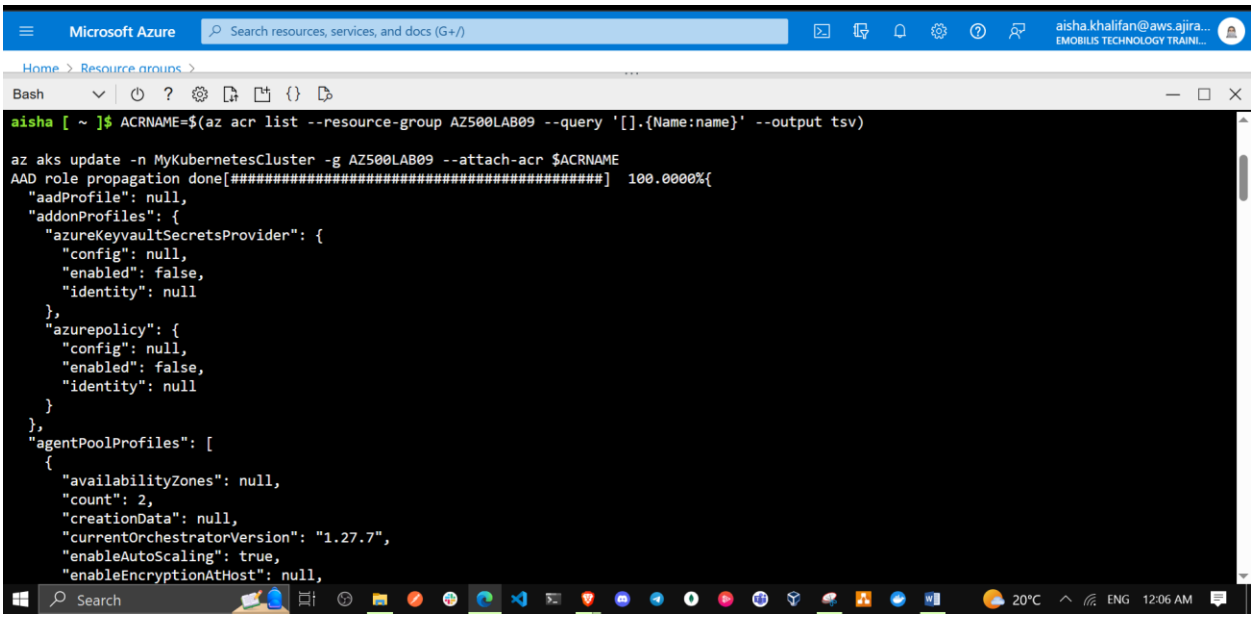
   ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)

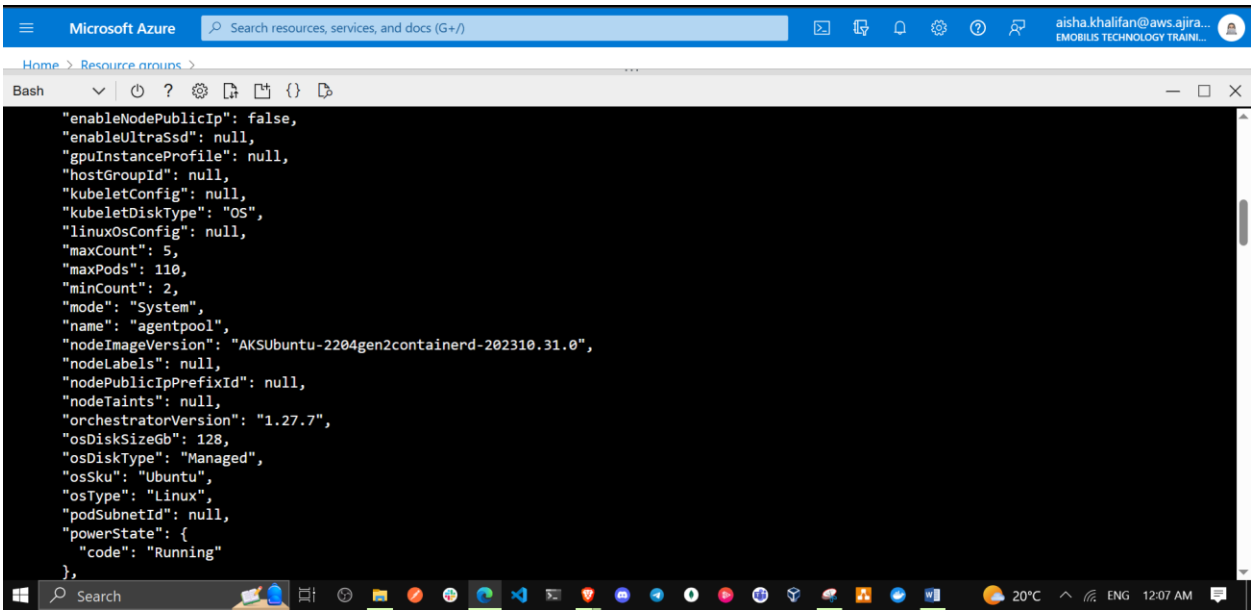
   az aks update -n MyKubernetesCluster -g AZ500LAB09 --attach-acr $ACRNAME

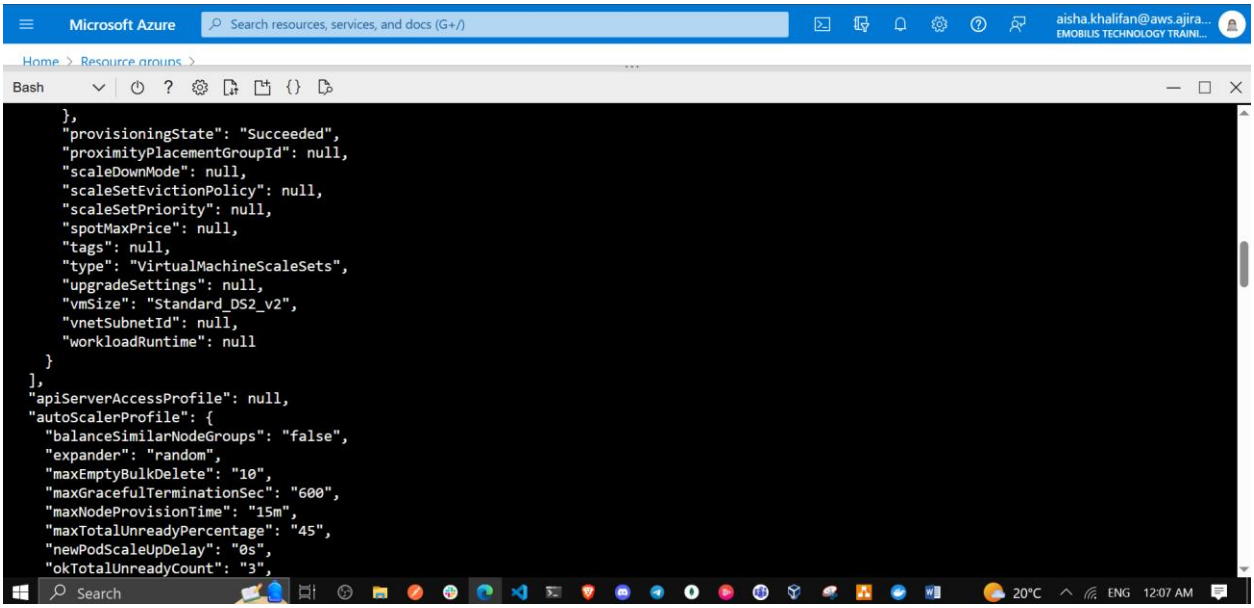
Note: This command grants the ‘acrpull’ role assignment to the ACR.

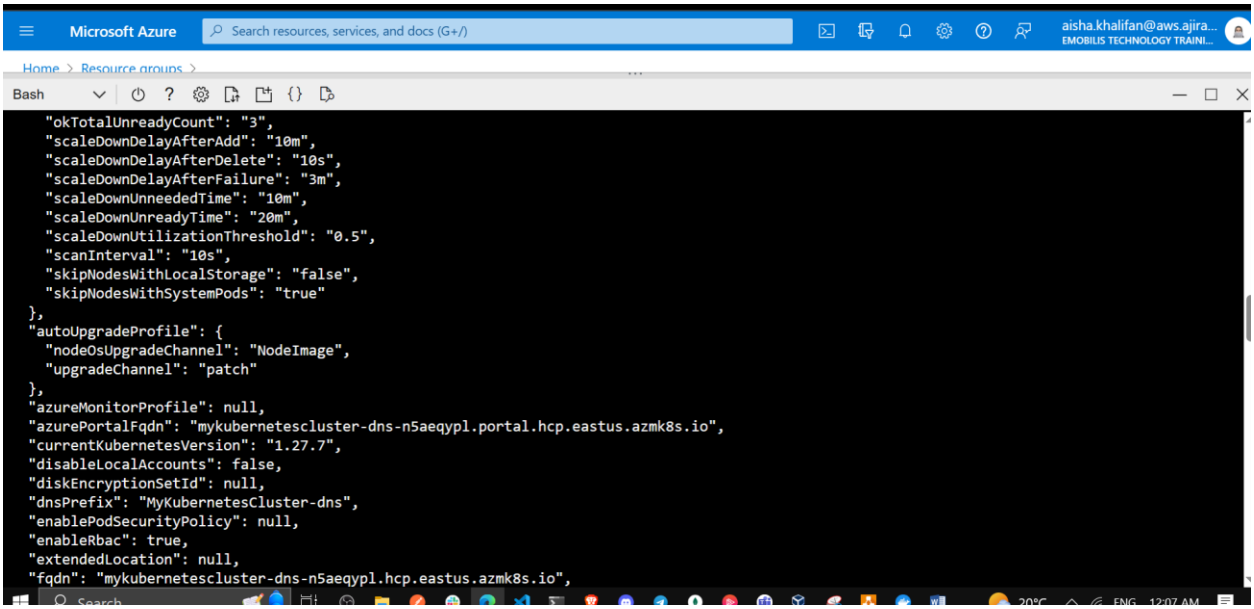
Note: It may take a few minutes for this command to complete.

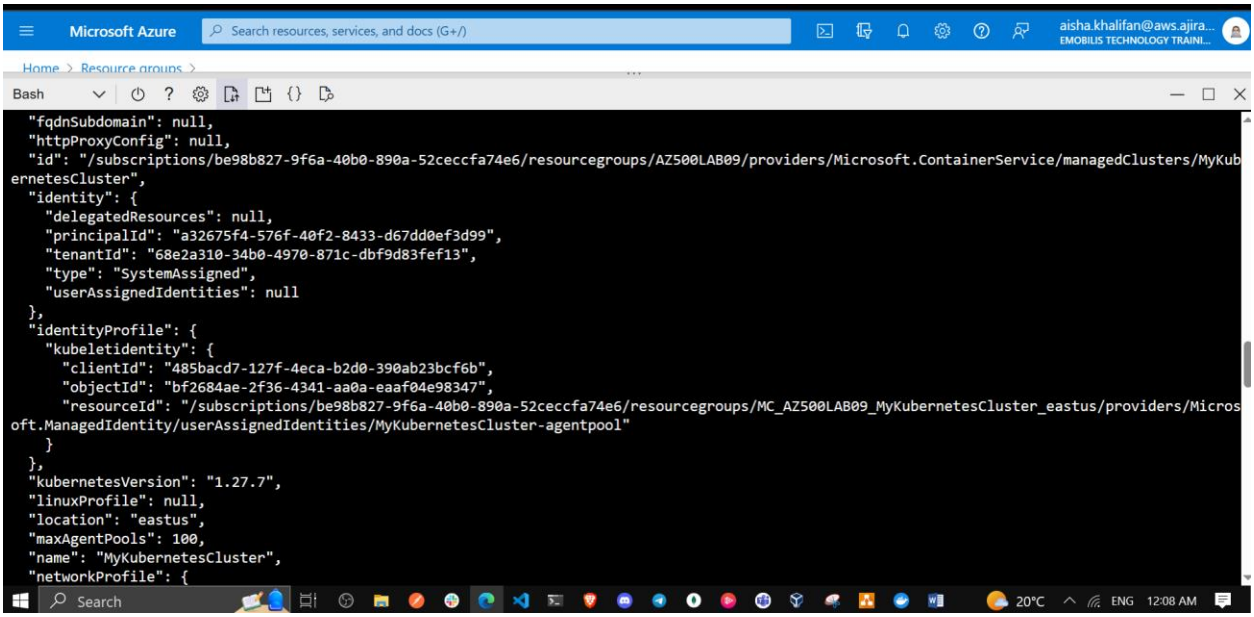

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell
aisha [~]$ az aks get-credentials --resource-group AZ500LAB09 --name MyKubernetesCluster
Merged "MyKubernetesCluster" as current context in /home/aisha/.kube/config
aisha [~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-agentpool-30345461-vmss000000 Ready agent 9m21s v1.27.7
aks-agentpool-30345461-vmss000001 Ready agent 9m27s v1.27.7
aisha [~]$ ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)
az aks update -n MyKubernetesCluster -g AZ500LAB09 --attach-acr $ACRNAME
/ Running ..

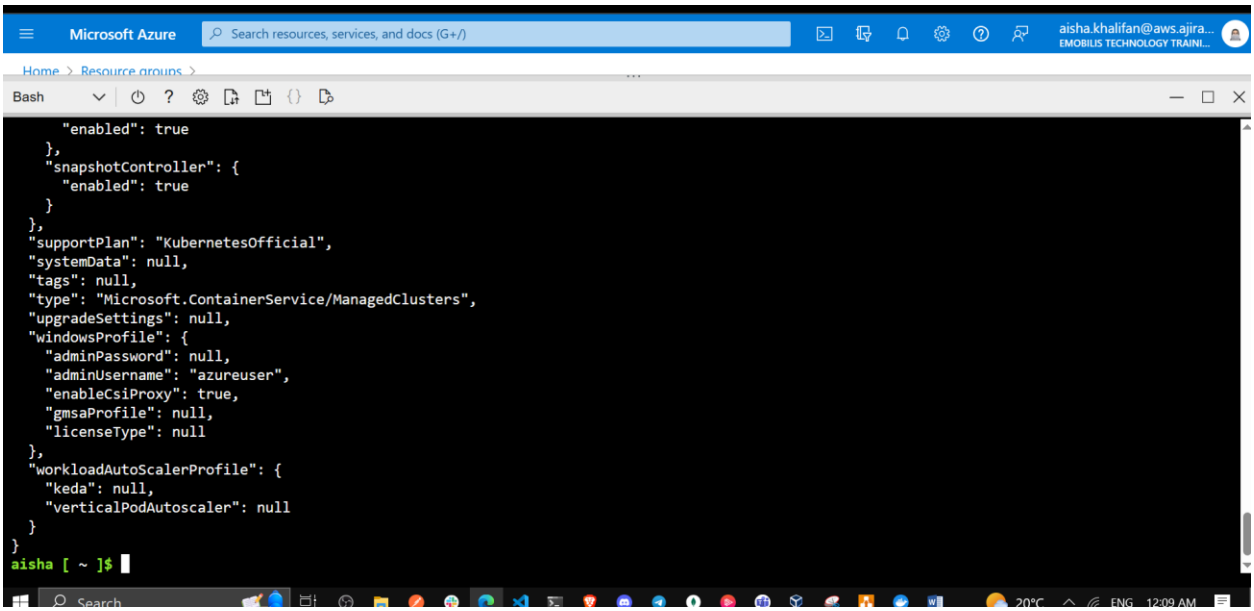

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
aisha [~]$ ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)
az aks update -n MyKubernetesCluster -g AZ500LAB09 --attach-acr $ACRNAME
AAD role propagation done[##] 100.0000%{
"aadProfile": null,
"addonProfiles": {
"azureKeyvaultSecretsProvider": {
"config": null,
"enabled": false,
"identity": null
},
"azurepolicy": {
"config": null,
"enabled": false,
"identity": null
}
},
"agentPoolProfiles": [
{
"availabilityZones": null,
"count": 2,
"creationData": null,
"currentOrchestratorVersion": "1.27.7",
"enableAutoScaling": true,
"enableEncryptionAtHost": null,
Search
20°C
ENG
12:06 AM


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
"enableNodePublicIp": false,
"enableUltraSsd": null,
"gpuInstanceProfile": null,
"hostGroupId": null,
"kubeletConfig": null,
"kubeletDiskType": "OS",
"linuxOsConfig": null,
"maxCount": 5,
"maxPods": 110,
"minCount": 2,
"mode": "System",
"name": "agentpool",
"nodeImageVersion": "AKSUbuntu-2204gen2containerd-202310.31.0",
"nodeLabels": null,
"nodePublicIpPrefixId": null,
"nodeTaints": null,
"orchestratorVersion": "1.27.7",
"osDiskSizeGb": 128,
"osDiskType": "Managed",
"osSku": "Ubuntu",
"osType": "Linux",
"podSubnetId": null,
"powerState": {
"code": "Running"
},
Search
20°C
ENG
12:07 AM


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
},
"provisioningState": "Succeeded",
"proximityPlacementGroupId": null,
"scaleDownMode": null,
"scaleSetEvictionPolicy": null,
"scaleSetPriority": null,
"spotMaxPrice": null,
"tags": null,
"type": "VirtualMachineScaleSets",
"upgradeSettings": null,
"vmSize": "Standard_DS2_v2",
"vnetSubnetId": null,
"workloadRuntime": null
}
],
"apiServerAccessProfile": null,
"autoScalerProfile": {
"balanceSimilarNodeGroups": "false",
"expander": "random",
"maxEmptyBulkDelete": "10",
"maxGracefulTerminationSec": "600",
"maxNodeProvisionTime": "15m",
"maxTotalUnreadyPercentage": "45",
"newPodScaleUpDelay": "0s",
"okTotalUnreadyCount": "3",
Search
20°C
ENG
12:07 AM


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
"okTotalUnreadyCount": "3",
"scaleDownDelayAfterAdd": "10m",
"scaleDownDelayAfterDelete": "10s",
"scaleDownDelayAfterFailure": "3m",
"scaleDownUnneededTime": "10m",
"scaleDownUnreadyTime": "20m",
"scaleDownUtilizationThreshold": "0.5",
"scanInterval": "10s",
"skipNodesWithLocalStorage": "false",
"skipNodesWithSystemPods": "true"
},
"autoUpgradeProfile": {
"nodeOsUpgradeChannel": "NodeImage",
"upgradeChannel": "patch"
},
"azureMonitorProfile": null,
"azurePortalFqdn": "mykubernetescluster-dns-n5aeqypl.portal.hcp.eastus.azmk8s.io",
"currentKubernetesVersion": "1.27.7",
"disableLocalAccounts": false,
"diskEncryptionSetId": null,
"dnsPrefix": "MyKubernetesCluster-dns",
"enablePodSecurityPolicy": null,
"enableRbac": true,
"extendedLocation": null,
"fqdn": "mykubernetescluster-dns-n5aeqypl.hcp.eastus.azmk8s.io",


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Home > Resource groups >
Bash
"fqdnSubdomain": null,
"httpProxyConfig": null,
"id": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourcegroups/AZ500LAB09/providers/Microsoft.ContainerService/managedClusters/MyKubernetesCluster",
"identity": {
"delegatedResources": null,
"principalId": "a32675f4-576f-40f2-8433-d67dd0ef3d99",
"tenantId": "68e2a310-34b0-4970-871c-dbf9d83fef13",
"type": "SystemAssigned",
"userAssignedIdentities": null
},
"identityProfile": {
"kubeletidentity": {
"clientId": "485bacd7-127f-4eca-b2d0-390ab23bcf6b",
"objectId": "bf2684ae-2f36-4341-aa0a-eaaf04e98347",
"resourceId": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourcegroups/MC_AZ500LAB09_MyKubernetesCluster_eastus/providers/Microsoft.ManagedIdentity/userAssignedIdentities/MyKubernetesCluster-agentpool"
}
},
"kubernetesVersion": "1.27.7",
"linuxProfile": null,
"location": "eastus",
"maxAgentPools": 100,
"name": "MyKubernetesCluster",
"networkProfile": {



2. In the Bash session within the Cloud Shell pane, run the following to grant the AKS cluster the Contributor role to its virtual network.

RG_AKS=AZ500LAB09

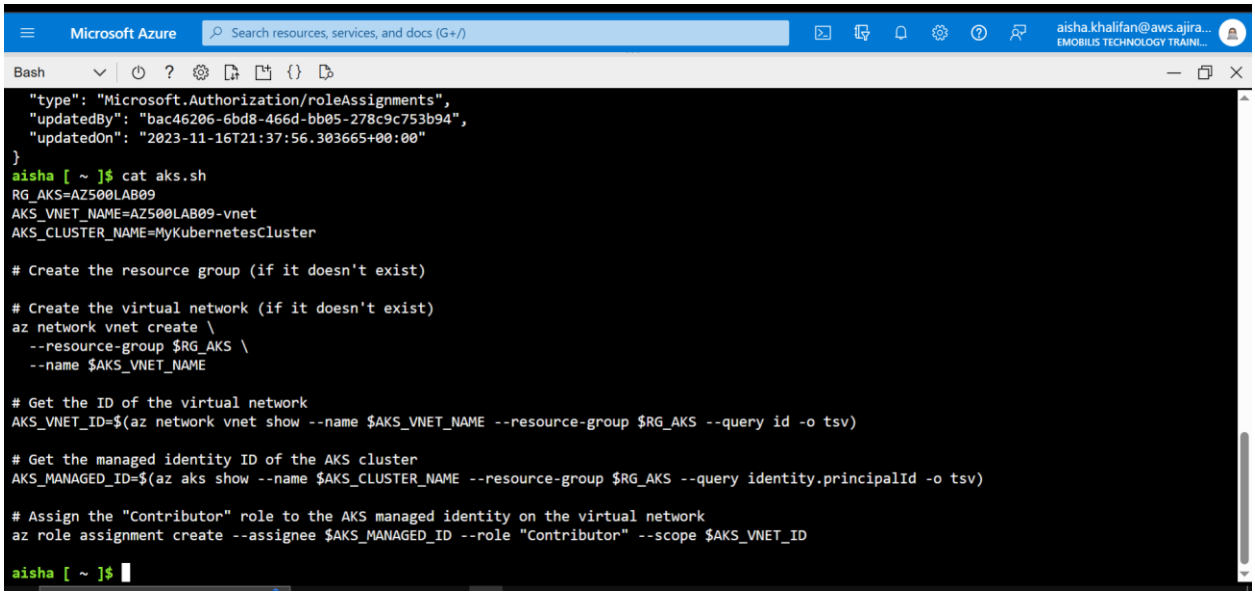
AKS_VNET_NAME=AZ500LAB09-vnet

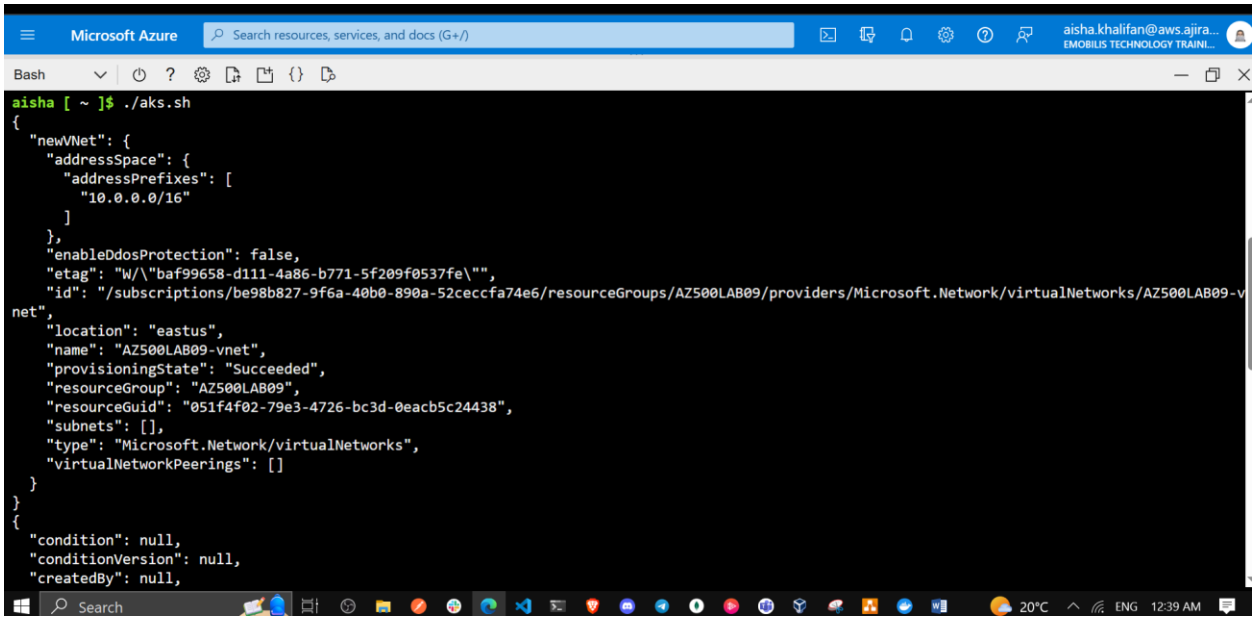
AKS_CLUSTER_NAME=MyKubernetesCluster

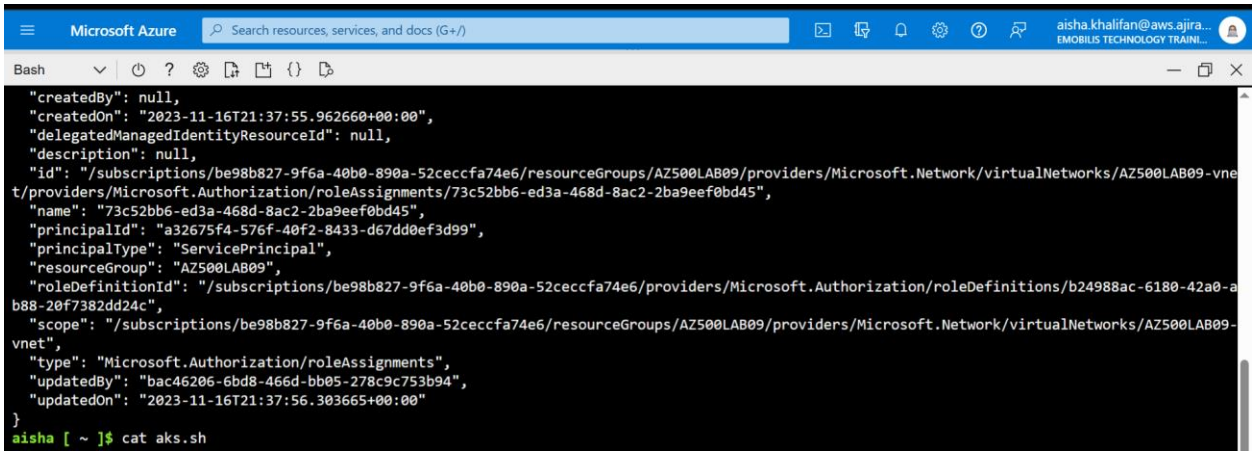
AKS_VNET_ID=$(az network vnet show --name $AKS_VNET_NAME --resource-group $RG_AKS --query id -o tsv)

AKS_MANAGED_ID=$(az aks show --name $AKS_CLUSTER_NAME --resource-group $RG_AKS --query identity.principalId -o tsv)

az role assignment create --assignee $AKS_MANAGED_ID --role "Contributor" --scope $AKS_VNET_ID


Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
"type": "Microsoft.Authorization/roleAssignments",
"updatedBy": "bac46206-6bd8-466d-bb05-278c9c753b94",
"updatedOn": "2023-11-16T21:37:56.303665+00:00"
}
aisha [~]$ cat aks.sh
RG_AKS=AZ500LAB09
AKS_VNET_NAME=AZ500LAB09-vnet
AKS_CLUSTER_NAME=MyKubernetesCluster
# Create the resource group (if it doesn't exist)
# Create the virtual network (if it doesn't exist)
az network vnet create \
--resource-group $RG_AKS \
--name $AKS_VNET_NAME
# Get the ID of the virtual network
AKS_VNET_ID=$(az network vnet show --name $AKS_VNET_NAME --resource-group $RG_AKS --query id -o tsv)
# Get the managed identity ID of the AKS cluster
AKS_MANAGED_ID=$(az aks show --name $AKS_CLUSTER_NAME --resource-group $RG_AKS --query identity.principalId -o tsv)
# Assign the "Contributor" role to the AKS managed identity on the virtual network
az role assignment create --assignee $AKS_MANAGED_ID --role "Contributor" --scope $AKS_VNET_ID
aisha [~]$

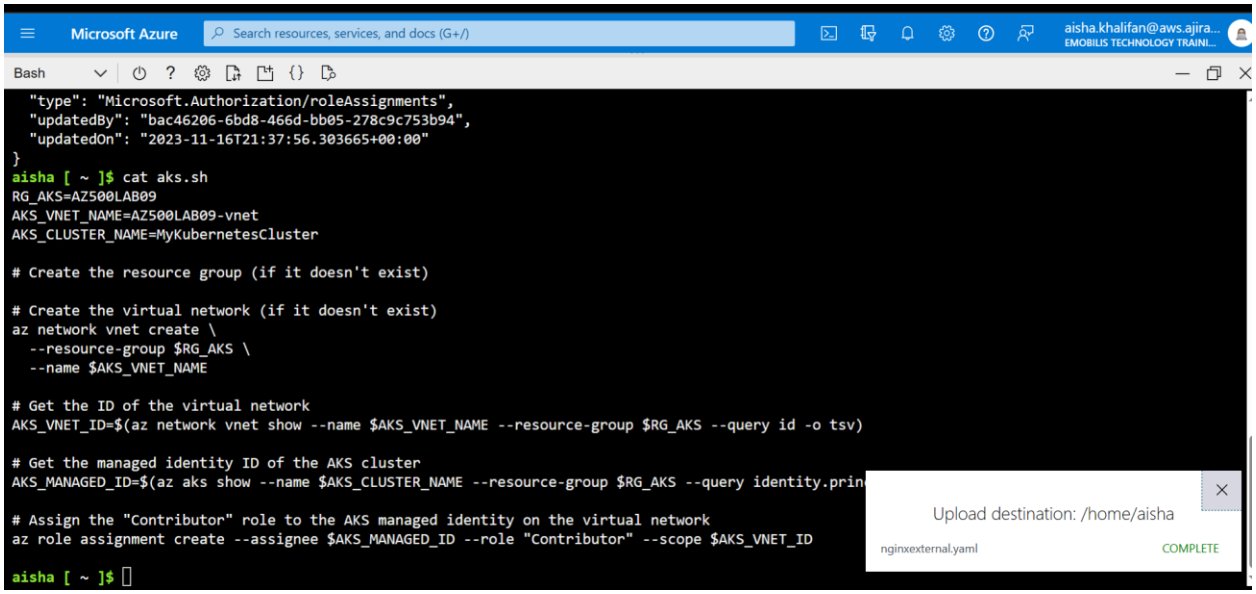

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
aisha [~]$ ./aks.sh
{
"newVNet": {
"addressSpace": {
"addressPrefixes": [
"10.0.0.0/16"
]
},
"enableDdosProtection": false,
"etag": "W/\"baf99658-d111-4a86-b771-5f209f0537fe\"",
"id": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourceGroups/AZ500LAB09/providers/Microsoft.Network/virtualNetworks/AZ500LAB09-v
net",
"location": "eastus",
"name": "AZ500LAB09-vnet",
"provisioningState": "Succeeded",
"resourceGroup": "AZ500LAB09",
"resourceGuid": "051f4f02-79e3-4726-bc3d-0eacb5c24438",
"subnets": [],
"type": "Microsoft.Network/virtualNetworks",
"virtualNetworkPeerings": []
}
}
{
"condition": null,
"conditionVersion": null,
"createdBy": null,
Search
20°C
ENG
12:39 AM

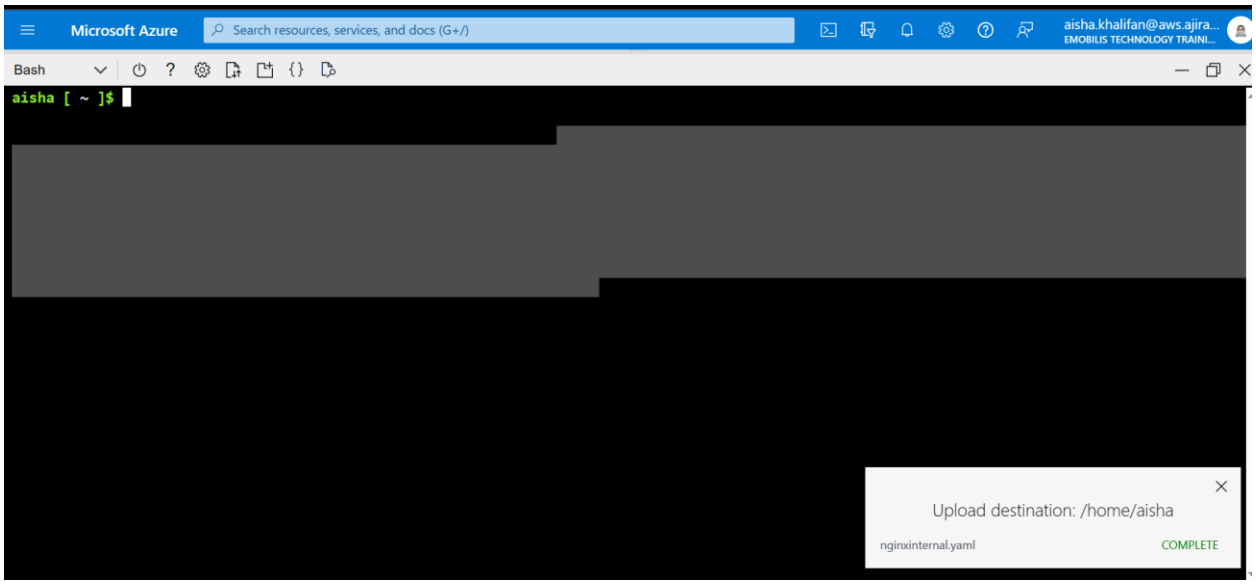

Microsoft Azure
Search resources, services, and docs (G+/)
aisha.khalifan@aws.ajira...
EMOBILIS TECHNOLOGY TRAINI...
Bash
"createdBy": null,
"createdOn": "2023-11-16T21:37:55.962660+00:00",
"delegatedManagedIdentityResourceId": null,
"description": null,
"id": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourceGroups/AZ500LAB09/providers/Microsoft.Network/virtualNetworks/AZ500LAB09-vne
t/providers/Microsoft.Authorization/roleAssignments/73c52bb6-ed3a-468d-8ac2-2ba9eef0bd45",
"name": "73c52bb6-ed3a-468d-8ac2-2ba9eef0bd45",
"principalId": "a32675f4-576f-40f2-8433-d67dd0ef3d99",
"principalType": "ServicePrincipal",
"resourceGroup": "AZ500LAB09",
"roleDefinitionId": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/providers/Microsoft.Authorization/roleDefinitions/b24988ac-6180-42a0-a
b88-20f7382dd24c",
"scope": "/subscriptions/be98b827-9f6a-40b0-890a-52ceccfa74e6/resourceGroups/AZ500LAB09/providers/Microsoft.Network/virtualNetworks/AZ500LAB09-
vnet",
"type": "Microsoft.Authorization/roleAssignments",
"updatedBy": "bac46206-6bd8-466d-bb05-278c9c753b94",
"updatedOn": "2023-11-16T21:37:56.303665+00:00"
}
aisha [~]$ cat aks.sh

Task 5: Deploy an external service to AKS

In this task, you will download the Manifest files, edit the YAML file, and apply your changes to the cluster.

1. In the Bash session within the Cloud Shell pane, click the **Upload/Download files** icon, in the drop-down menu, click **Upload**, in the **Open** dialog box, naviate to the location where you downloaded the lab files, select **\Allfiles\Labs\09\nginxexternal.yaml** click **Open**. Next, select **\Allfiles\Labs\09\nginxinternal.yaml**, and click **Open**.

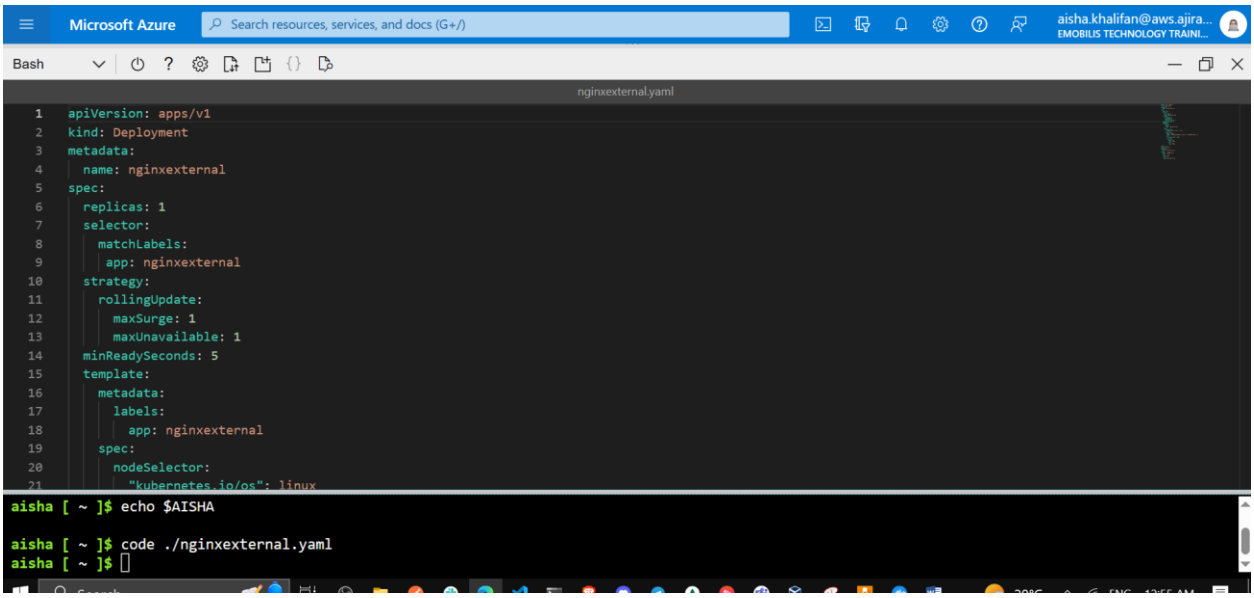




2. In the Bash session within the Cloud Shell pane, run the following to identify the name of the Azure Container Registry instance:

   echo $ACRNAME

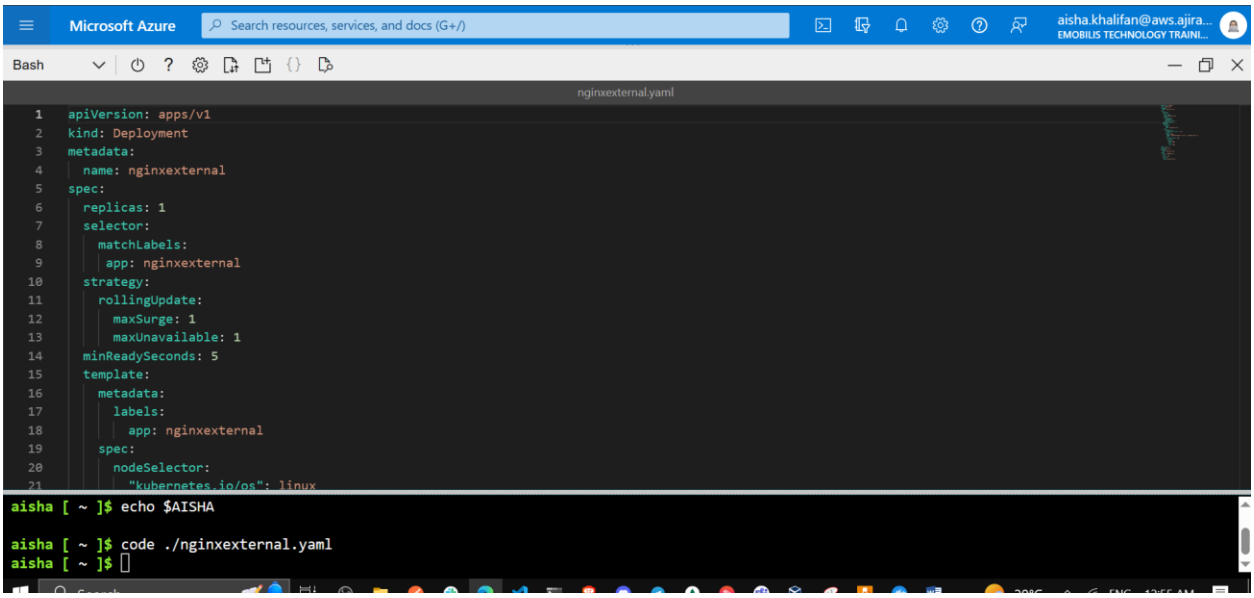
Note: Record the Azure Container Registry instance name. You will need it later in this task.



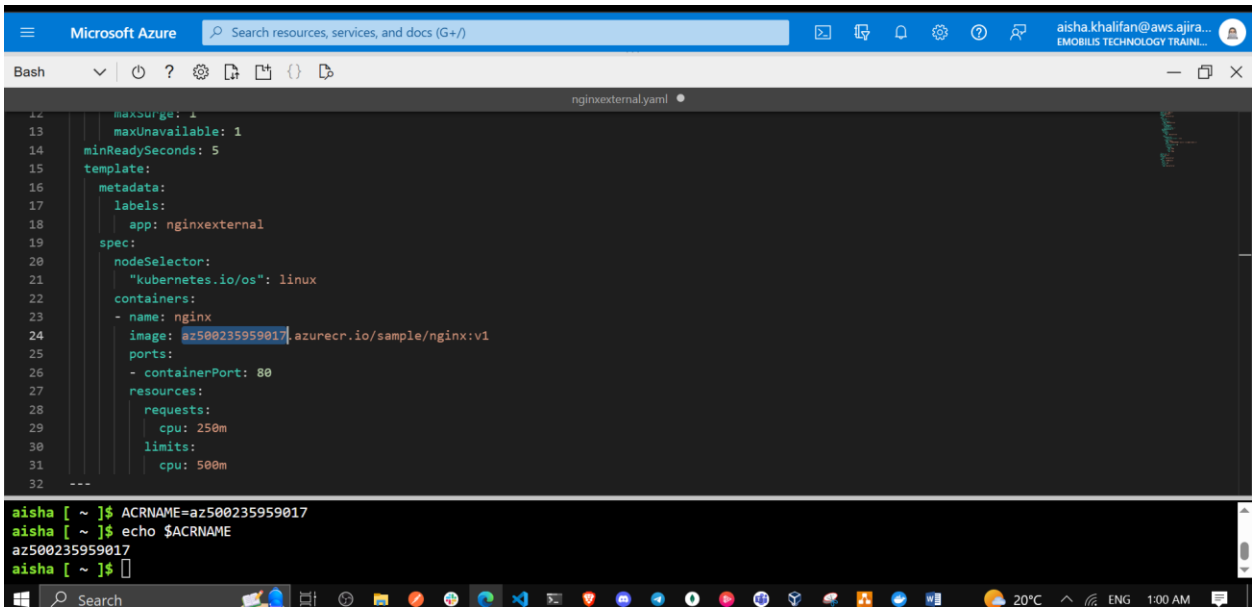
3. In the Bash session within the Cloud Shell pane, run the following to open the nginxexternal.yaml file, so you can edit its content.

   code ./nginxexternal.yaml

Note: This is the *external* yaml file.



4. In the editor pane, scroll down to **line 24** and replace the **<ACRUniquename>** placeholder with the ACR name.



5. In the editor pane, in the upper right corner, click the **ellipses** icon, click **Save** and then click **Close editor**.

6. In the Bash session within the Cloud Shell pane, run the following to apply the change to the cluster:

   kubectl apply -f nginxexternal.yaml

7. In the Bash session within the Cloud Shell pane, review the output of the command you run in the previous task to verify that the deployment and the corresponding service have been created.

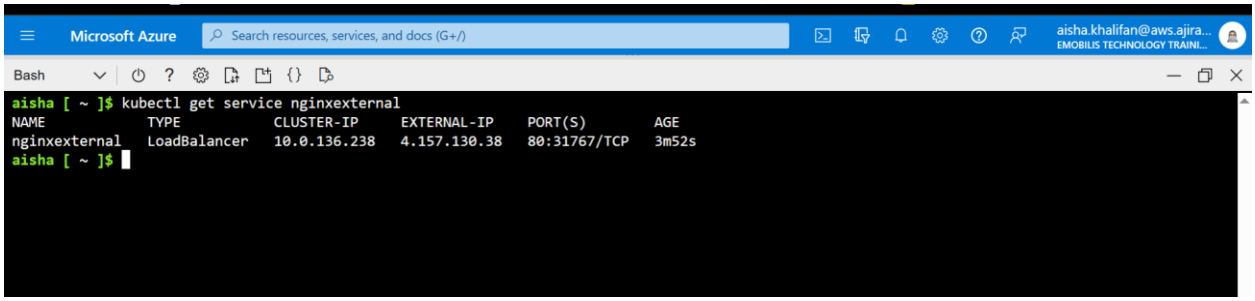
   deployment.apps/nginxexternal created
   service/nginxexternal created

Task 6: Verify the you can access an external AKS-hosted service

In this task, verify the container can be accessed externally using the public IP address.

1. In the Bash session within the Cloud Shell pane, run the following to retrieve information about the nginxexternal service including name, type, IP addresses, and ports.

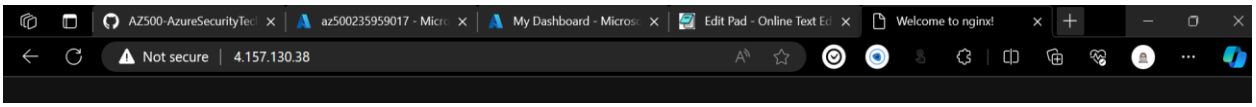
   kubectl get service nginxexternal



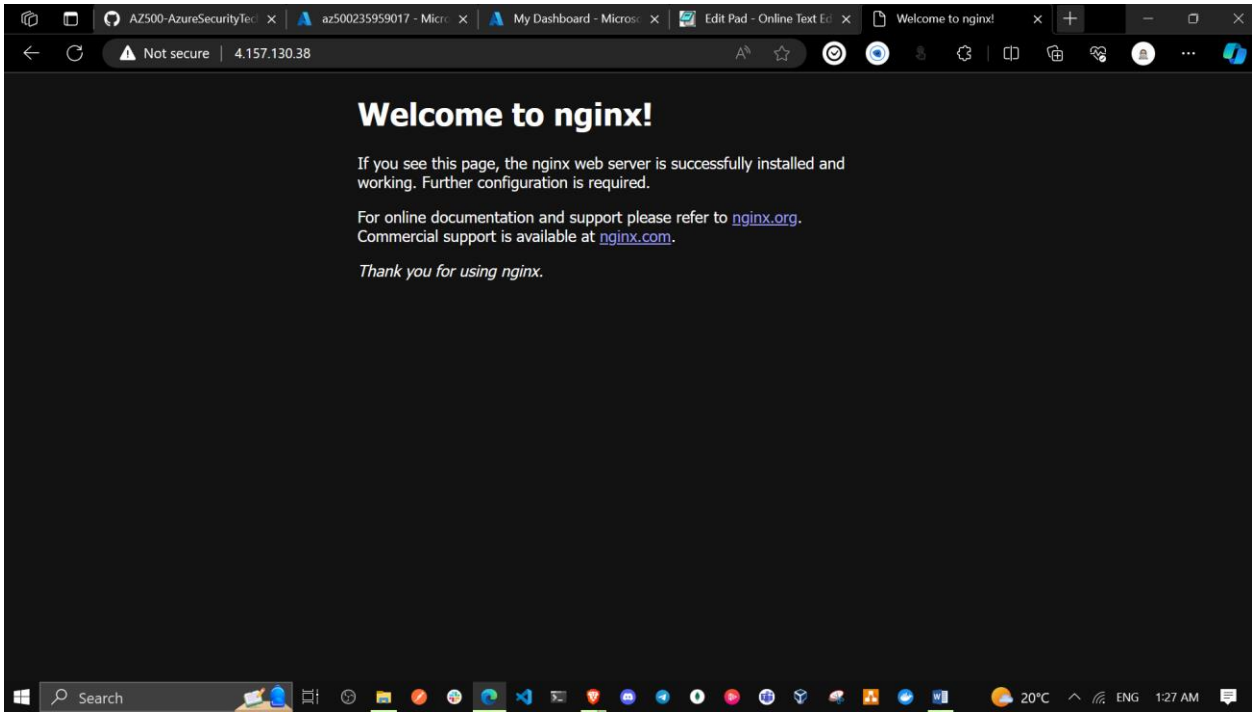
2. In the Bash session within the Cloud Shell pane, review the output and record the value in the External-IP column. You will need it in the next step.

   EXTERNAL IP = 4.157.130.38

3. Open a new browser tab and browse to the IP address you identified in the previous step.



4. Ensure the **Welcome to nginx!** page displays.



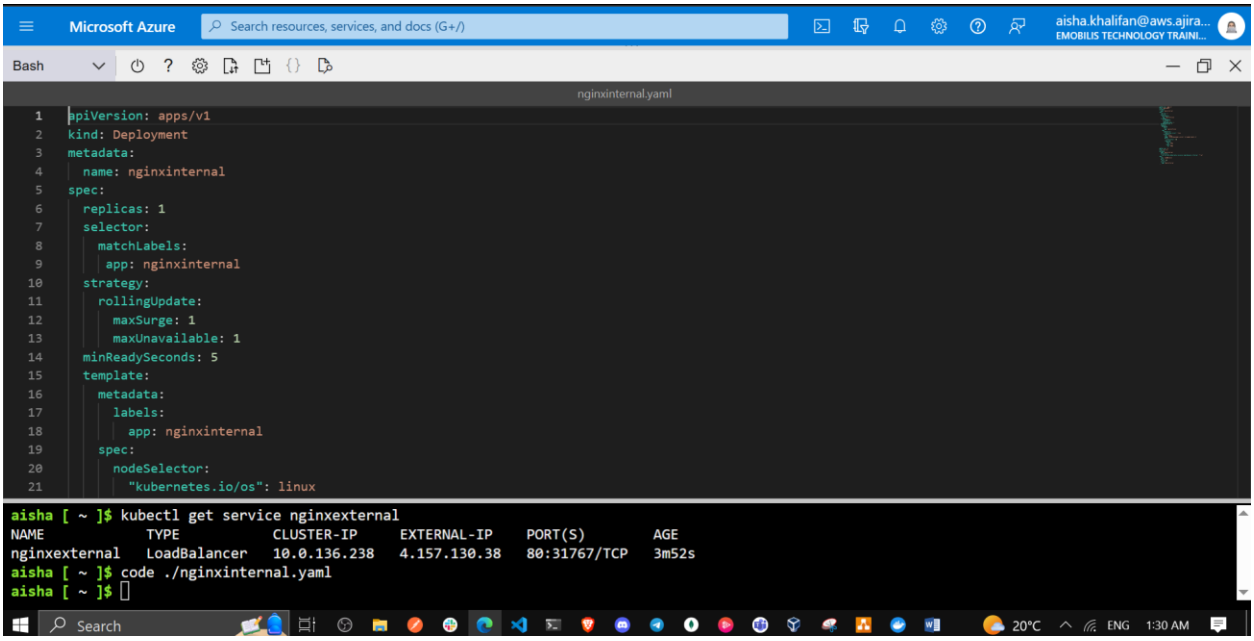
Task 7: Deploy an internal service to AKS

In this task, you will deploy the internal facing service on the AKS.

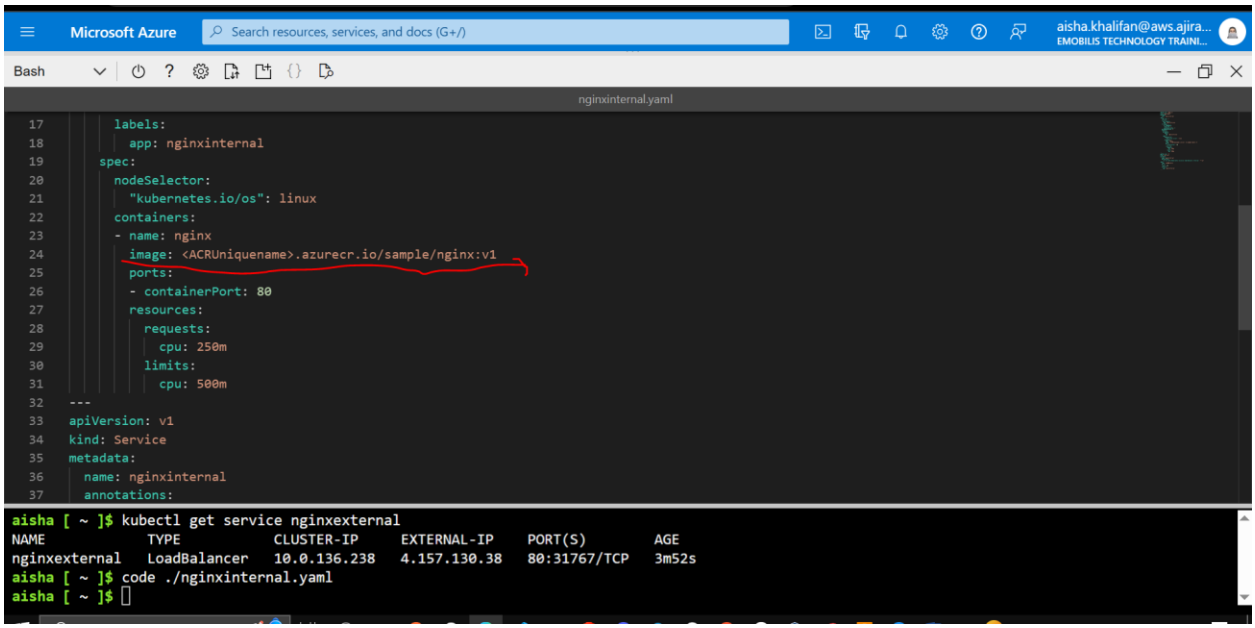
1. In the Bash session within the Cloud Shell pane, run the following to open the nginxintenal.yaml file, so you can edit its content.

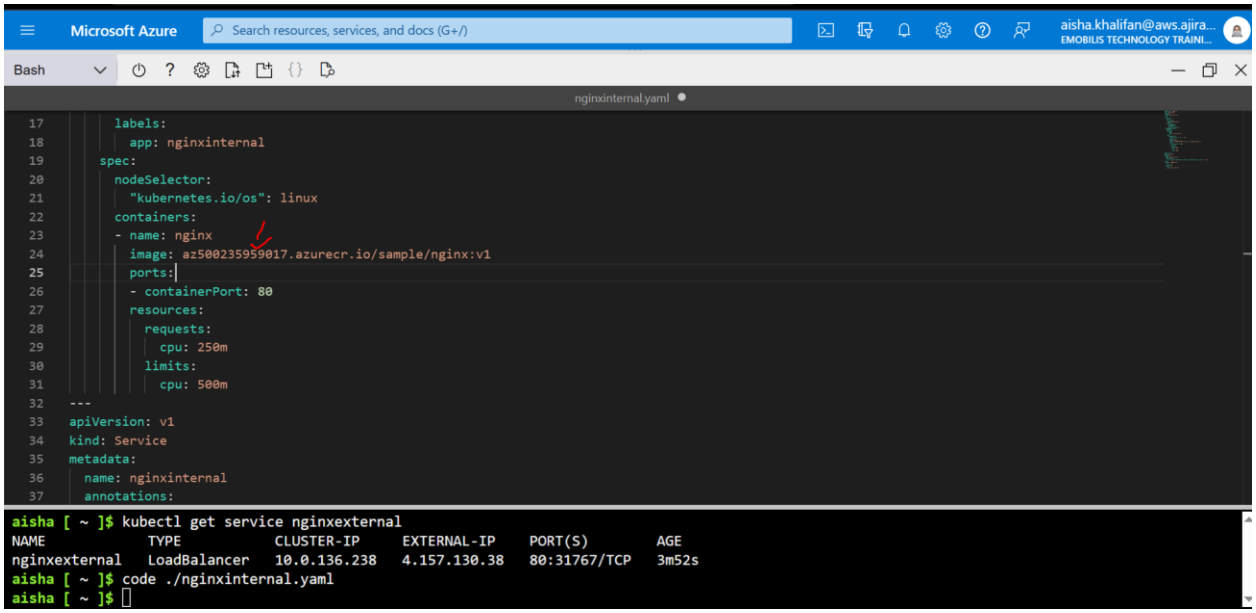
   code ./nginxinternal.yaml

Note: This is the *internal* yaml file.

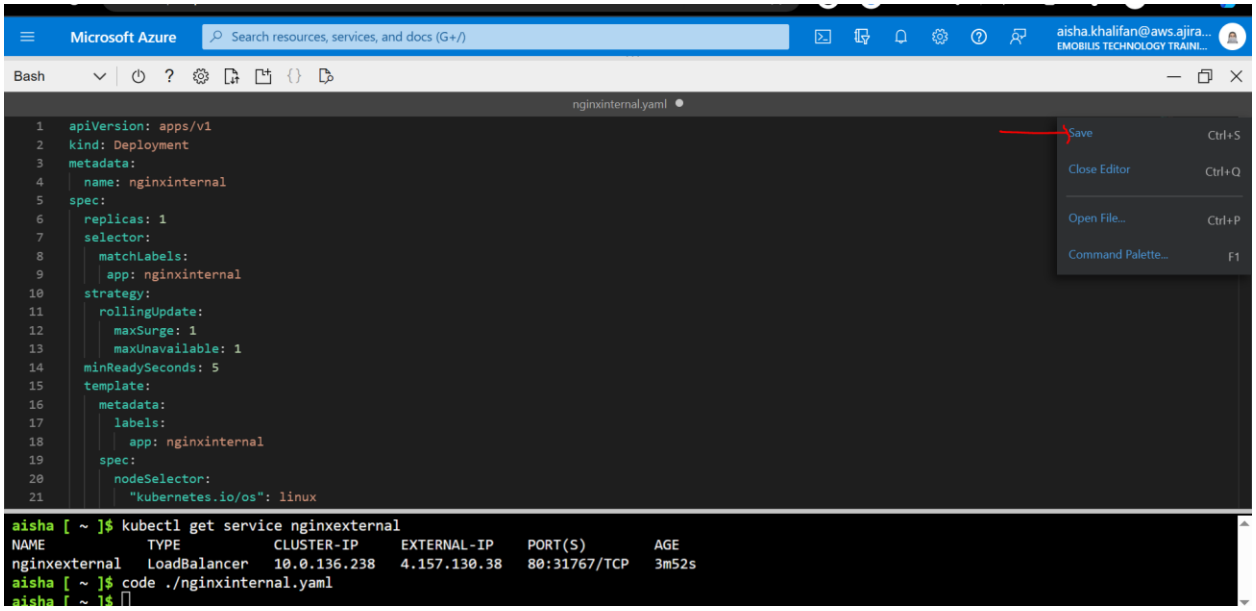


2. In the editor pane, scroll down to the line containing the reference to the container image and replace the **<ACRUniquename>** placeholder with the ACR name.



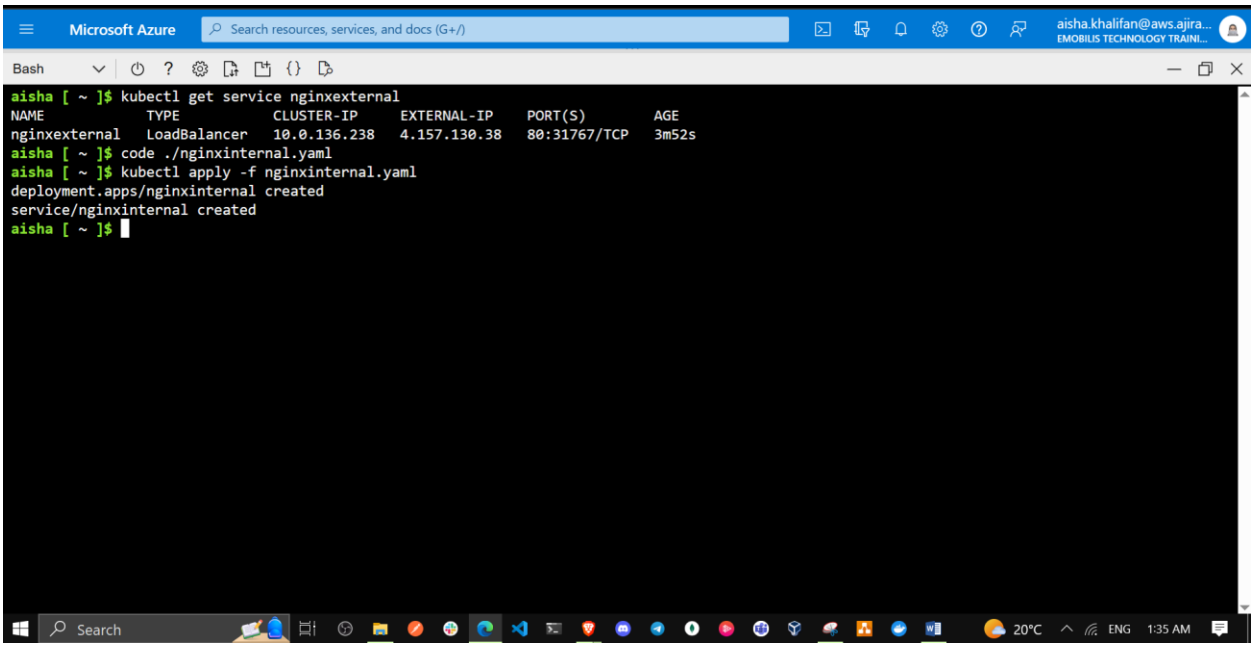


3. In the editor pane, in the upper right corner, click the **ellipses** icon, click **Save** and then click **Close editor**.



4. In the Bash session within the Cloud Shell pane, run the following to apply the change to the cluster:

   kubectl apply -f nginxinternal.yaml

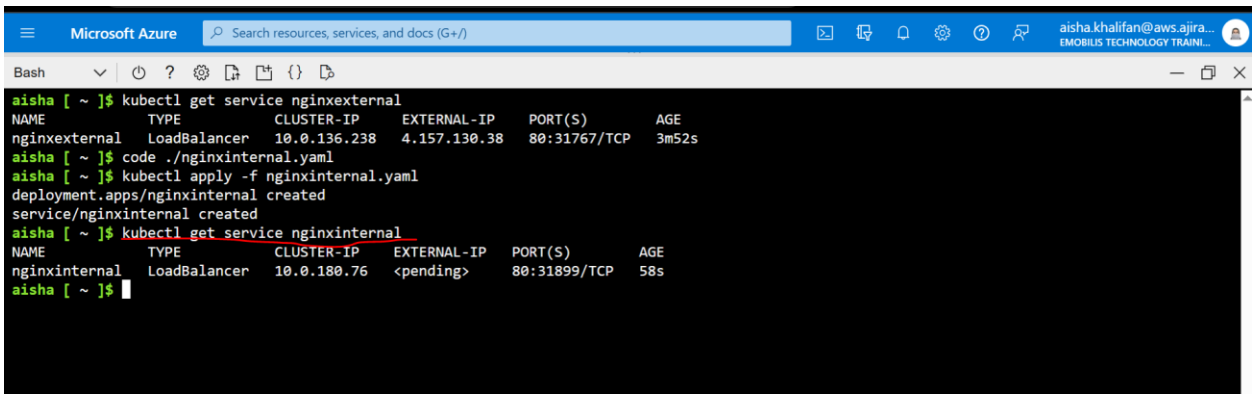


5. In the Bash session within the Cloud Shell pane, review the output to verify your deployment and the service have been created:

   deployment.apps/nginxinternal created
   service/nginxinternal created

6. In the Bash session within the Cloud Shell pane, run the following to retrieve information about the nginxinternal service including name, type, IP addresses, and ports.

   kubectl get service nginxinternal

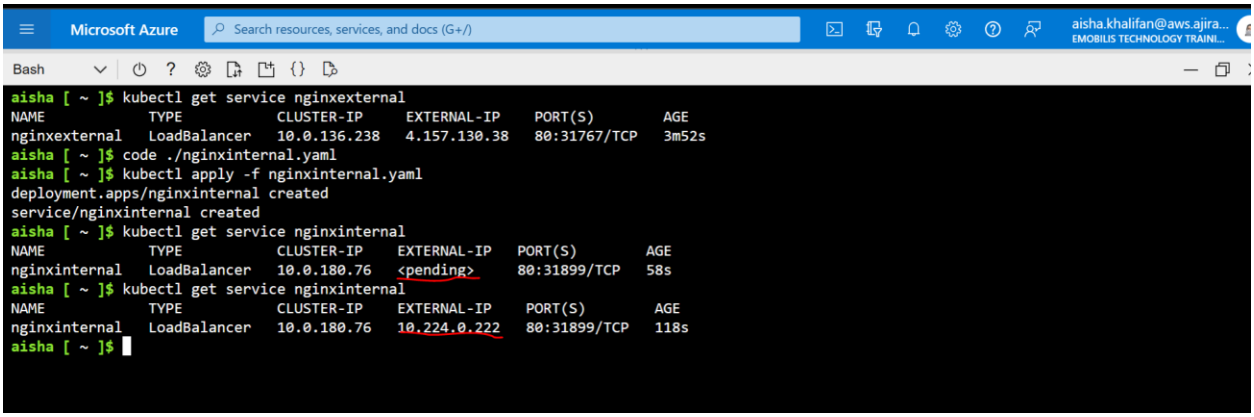


7. In the Bash session within the Cloud Shell pane, review the output. The External-IP is, in this case, a private IP address. If it is in a **Pending** state then run the previous command again.

   Note: Record this IP address. You will need it in the next task.

   Note: To access the internal service endpoint, you will connect interactively to one of the pods running in the cluster.

Note: Alternatively you could use the CLUSTER-IP address.

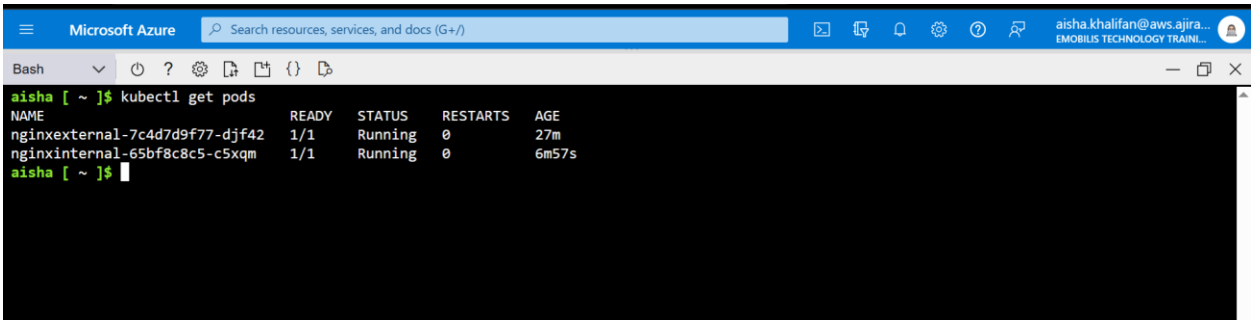


Task 8: Verify the you can access an internal AKS-hosted service

In this task, you will use one of the pods running on the AKS cluster to access the internal service.

1. In the Bash session within the Cloud Shell pane, run the following to list the pods in the default namespace on the AKS cluster:

kubectl get pods



2. In the listing of the pods, copy the first entry in the **NAME** column.

Note: This is the pod you will use in the subsequent steps.

nginxexternal-7c4d7d9f77-djf42

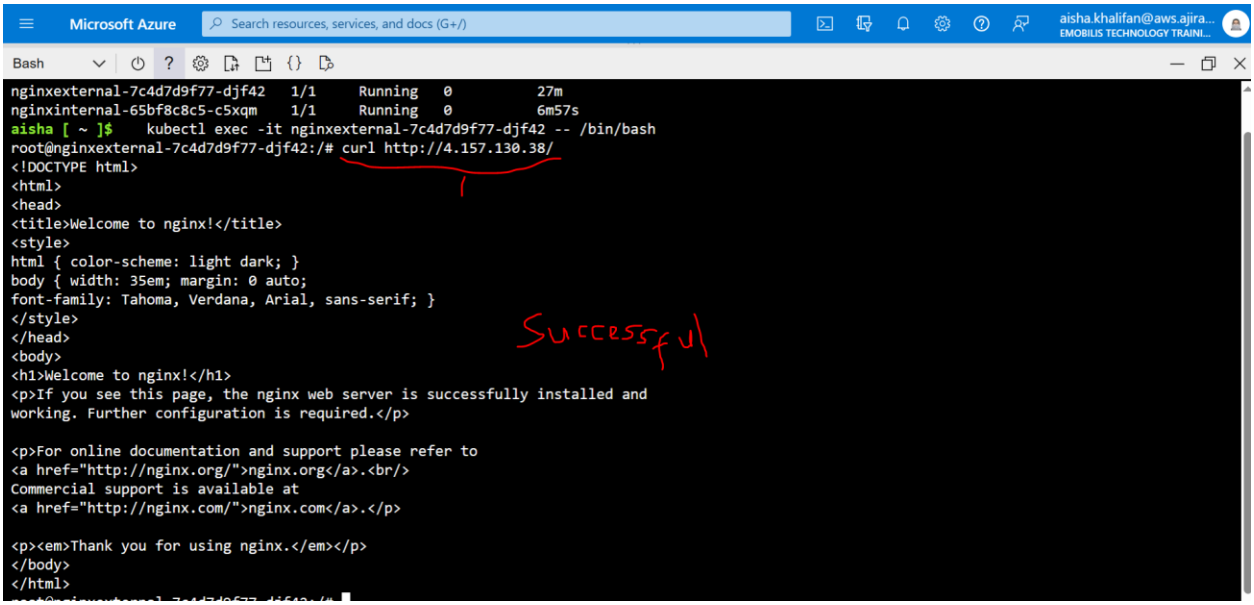
3. In the Bash session within the Cloud Shell pane, run the following to connect interactively to the first pod (replace the <pod_name> placeholder with the name you copied in the previous step):

   kubectl exec -it <pod_name> -- /bin/bash

4. In the Bash session within the Cloud Shell pane, run the following to verify that the nginx web site is available via the private IP address of the service (replace the <internal_IP> placeholder with the IP address you recorded in the previous task):

   curl http://<internal_IP>

This was my Internal IP = 4.157.130.38/



5. Close the Cloud Shell pane.

   Result: You have configured and secured ACR and AKS.

Clean up resources

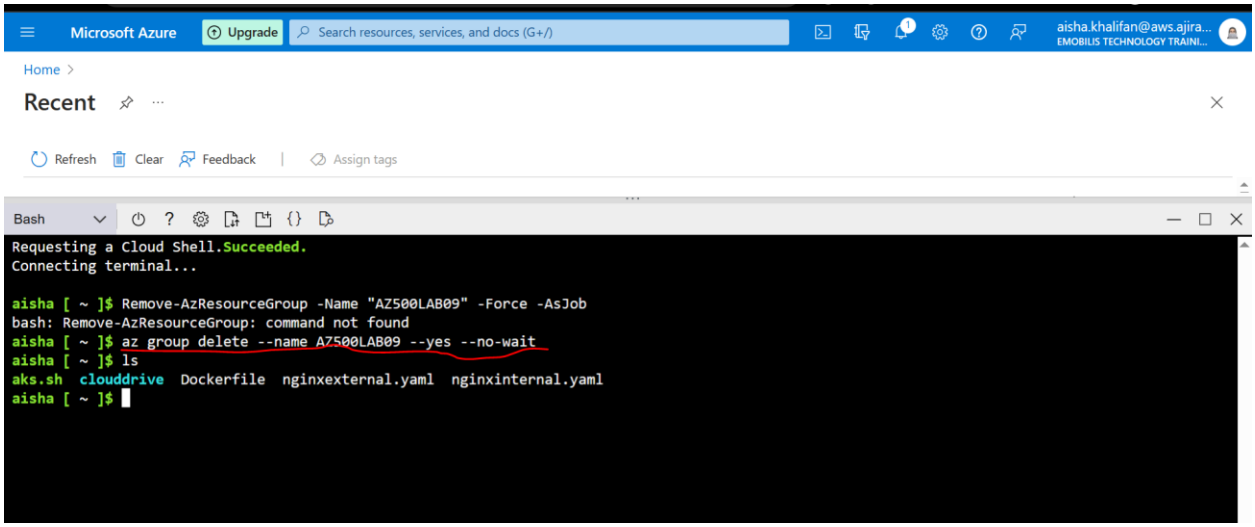
Remember to remove any newly created Azure resources that you no longer use. Removing unused resources ensures you will not incur unexpected costs.

1. In the Azure portal, open the Cloud Shell by clicking the first icon in the top right of the Azure Portal.

2. In the upper-left drop-down menu of the Cloud Shell pane, select **PowerShell** and, when prompted, click **Confirm**.

3. In the PowerShell session within the Cloud Shell pane, run the following to remove the resource groups you created in this lab:

   Remove-AzResourceGroup -Name "AZ500LAB09" -Force –AsJob



4. Close the **Cloud Shell** pane.

Conclusion

In the lab, I had a hands-on journey with Azure tech. Learned to make apps work in containers using Doc ker. Azure Container Registry helped keep the app stuff safe and organized. Then, we jumped into Azure Kubernetes Service, setting up a smart system for our apps that can grow easily. Made sure apps were saf e to use, whether we were working together or letting others see them. This all happened in the East US p art of Azure, giving me a cool experience with putting apps on the cloud.