Configuring and Securing ACR and AKS Report

Yunis Mohamed

MICROSOFT AZURE lab 09

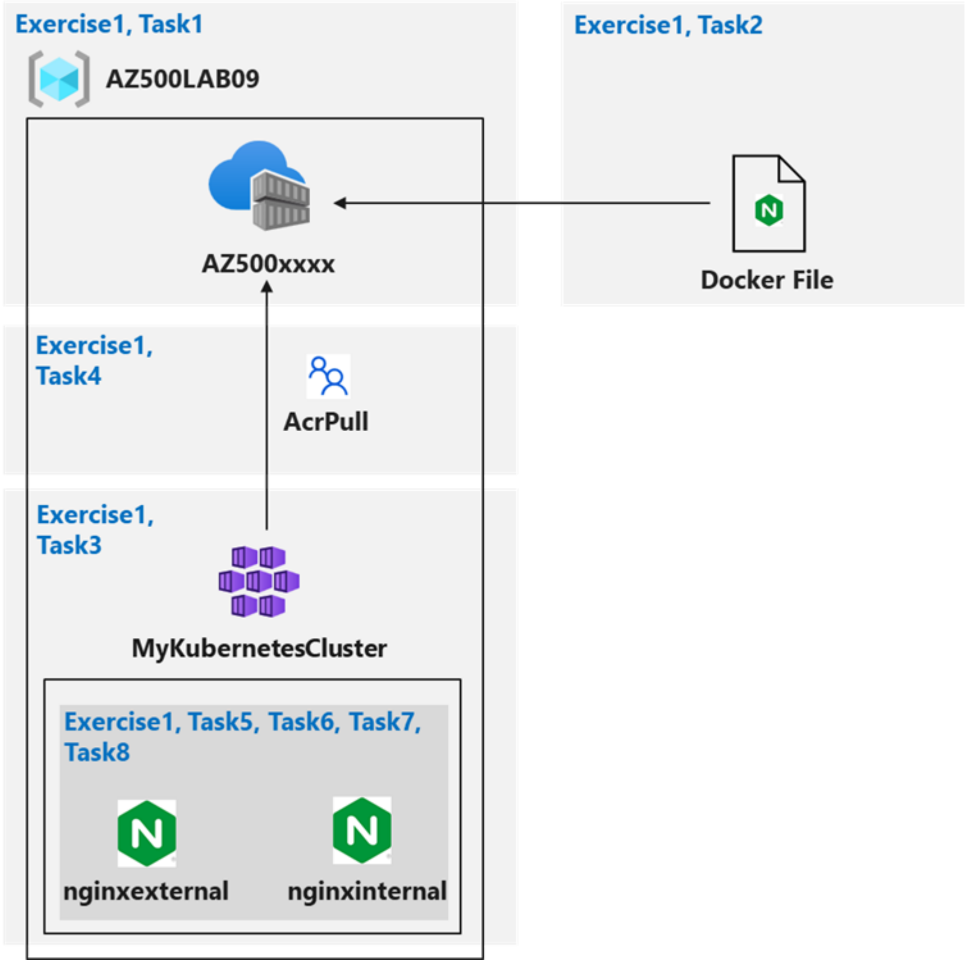
Table of Contents

Introduction

This report summarizes the discoveries and knowledge gained from Microsoft Azure Lab 09, which focused on the configuration and security aspects of Azure Container Registry (ACR) and Azure Kubernetes Service (AKS). ACR serves as a trustworthy and secure platform for the storage and management of container images, while AKS provides a scalable and resilient environment for deploying applications in containers. The report emphasizes the important insights and recommended approaches acquired during the lab, particularly regarding efficient configuration techniques and robust security measures for ACR and AKS. Implementing these practices enables organizations to guarantee the best possible performance and protection for their container-based deployments within the Azure platform.

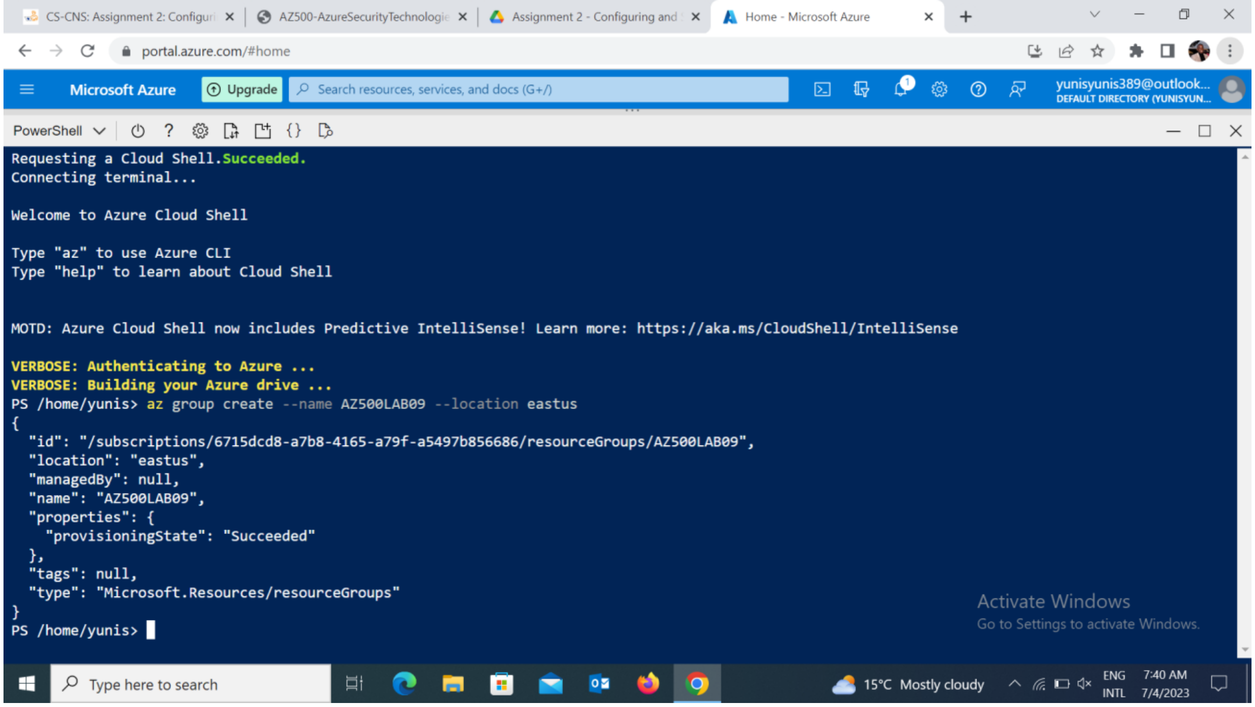
Task 1: Create an Azure Container Registry

Topology



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

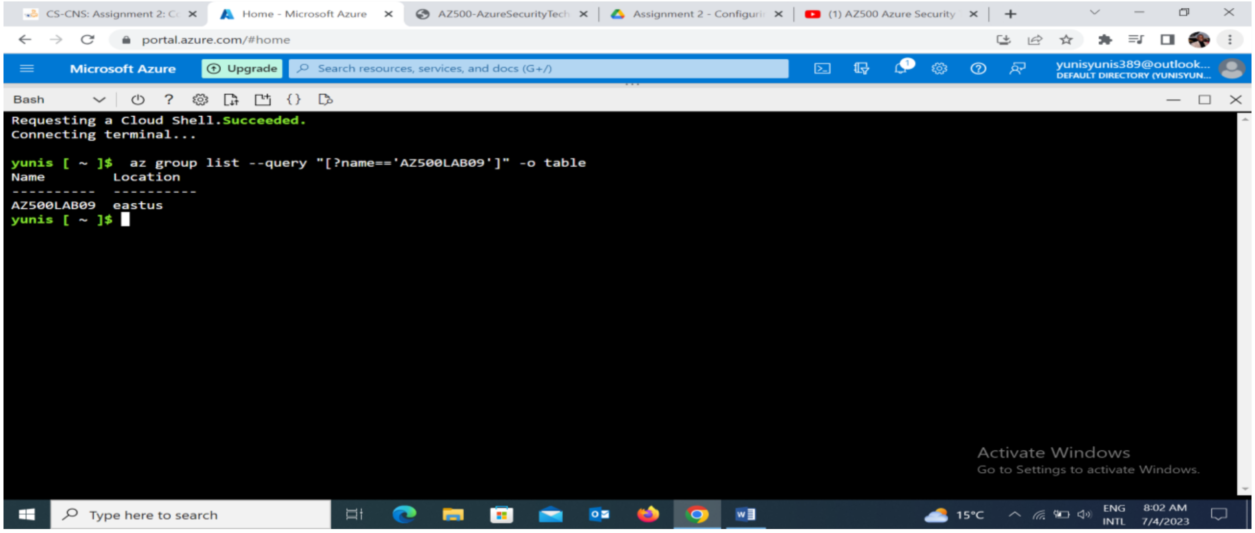


```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell


MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/yunis> az group create --name AZ500LAB09 --location eastus
{
  "id": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB09",
  "location": "eastus",
  "managedBy": null,
  "name": "AZ500LAB09",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
PS /home/yunis> 
```

CS-CNS: Assignment 2: Configur
AZ500-AzureSecurityTechnologie
Assignment 2 - Configuring and
Home - Microsoft Azure
portal.azure.com/#home
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
yunisyunis389@outlook...
DEFAULT DIRECTORY (YUNISYUN...
PowerShell
Activate Windows
Go to Settings to activate Windows.
Type here to search
15°C Mostly cloudy
ENG
INTL
7:40 AM
7/4/2023

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

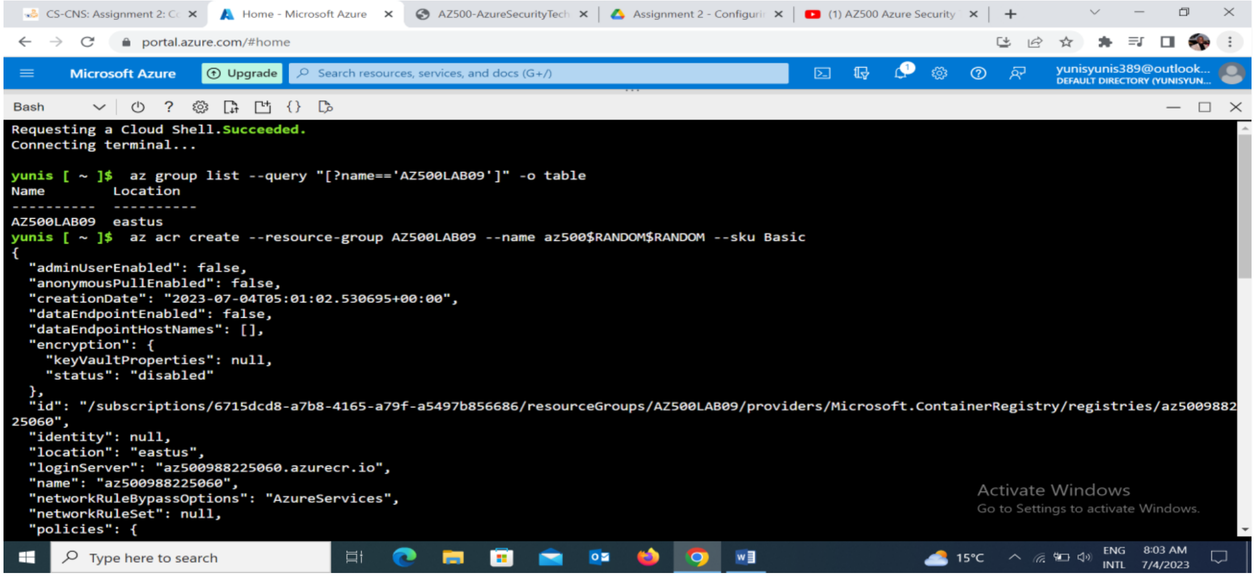
   az group list --query "[?name=='AZ500LAB09']" -o table



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

yunis [ ~ ]$  az group list --query "[?name=='AZ500LAB09']" -o table
Name        Location
----------  ----------
AZ500LAB09  eastus
yunis [ ~ ]$
```

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):

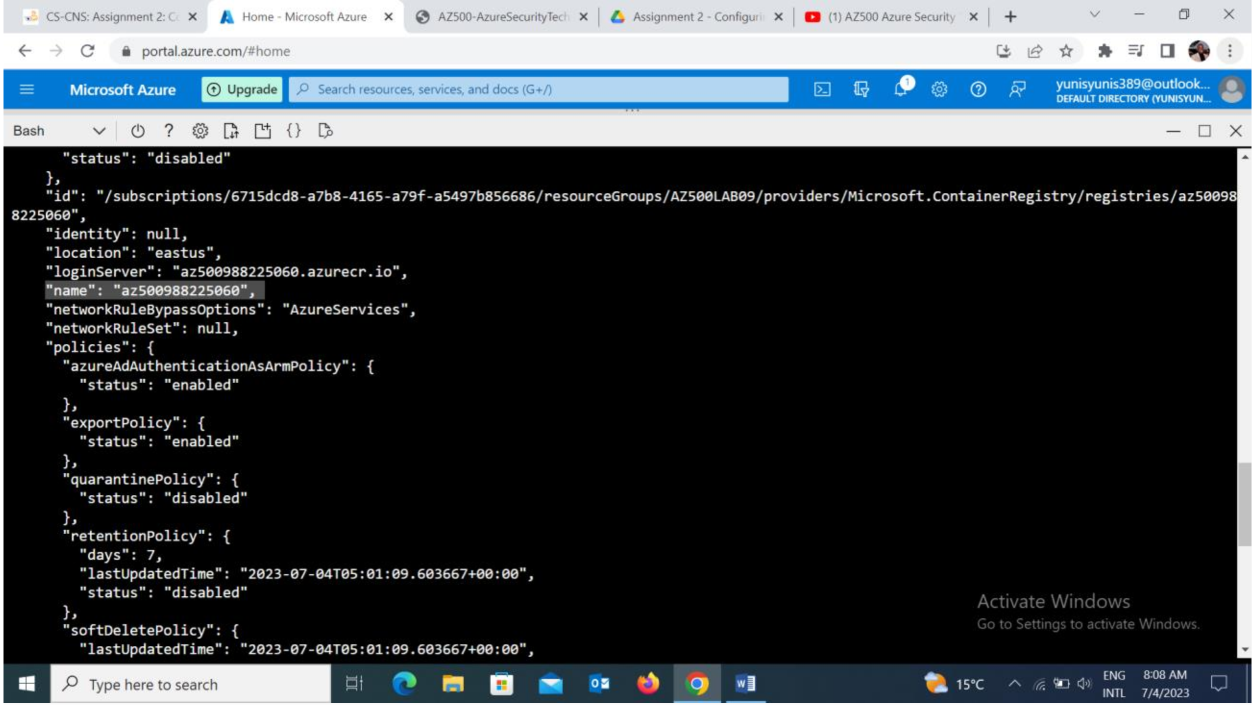
   az acr create --resource-group AZ500LAB09 --name az500$RANDOM$RANDOM --sku Basic



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

yunis [ ~ ]$  az group list --query "[?name=='AZ500LAB09']" -o table
Name        Location
----------  ----------
AZ500LAB09  eastus
yunis [ ~ ]$  az acr create --resource-group AZ500LAB09 --name az500$RANDOM$RANDOM --sku Basic
{
  "adminUserEnabled": false,
  "anonymousPullEnabled": false,
  "creationDate": "2023-07-04T05:01:02.530695+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB09/providers/Microsoft.ContainerRegistry/registries/az5009882
25060",
  "identity": null,
  "location": "eastus",
  "loginServer": "az500988225060.azurecr.io",
  "name": "az500988225060",
  "networkRuleBypassOptions": "AzureServices",
  "networkRuleSet": null,
  "policies": {
```

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

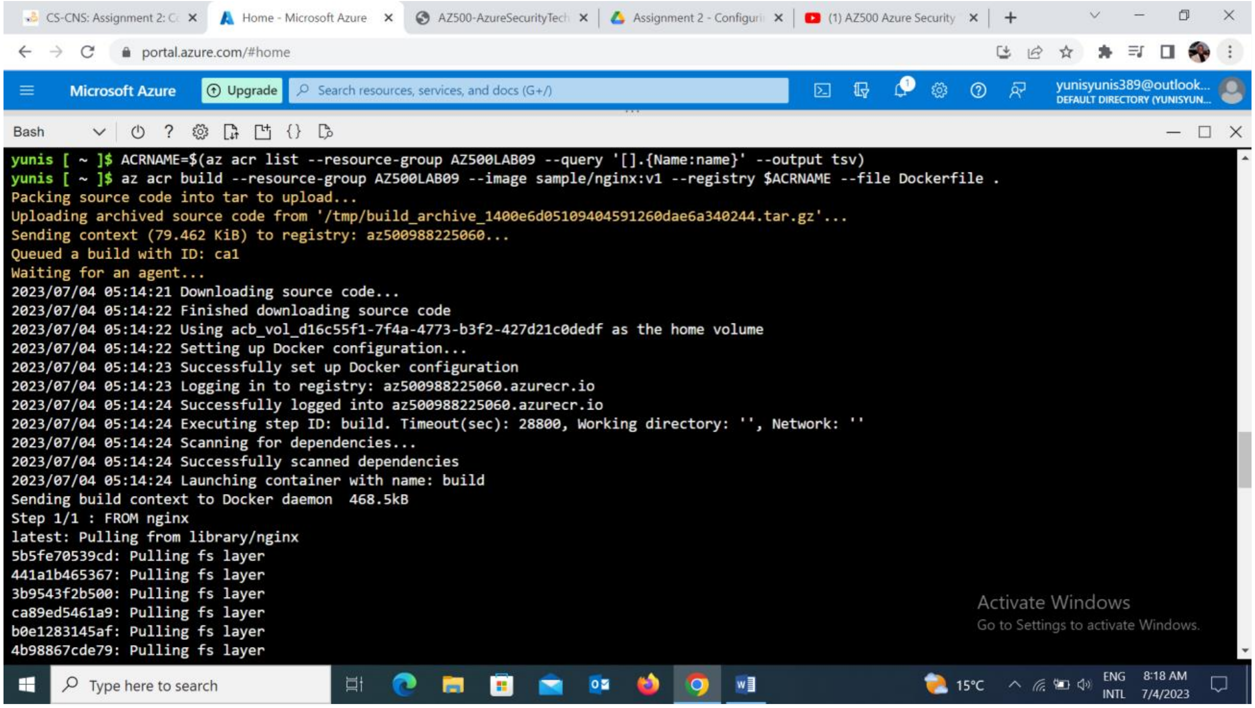


```
      "status": "disabled"
    },
    "id": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB09/providers/Microsoft.ContainerRegistry/registries/az50098
8225060",
    "identity": null,
    "location": "eastus",
    "loginServer": "az500988225060.azurecr.io",
    "name": "az500988225060",
    "networkRuleBypassOptions": "AzureServices",
    "networkRuleSet": null,
    "policies": {
      "azureAdAuthenticationAsArmPolicy": {
        "status": "enabled"
      },
      "exportPolicy": {
        "status": "enabled"
      },
      "quarantinePolicy": {
        "status": "disabled"
      },
      "retentionPolicy": {
        "days": 7,
        "lastUpdatedTime": "2023-07-04T05:01:09.603667+00:00",
        "status": "disabled"
      },
      "softDeletePolicy": {
        "lastUpdatedTime": "2023-07-04T05:01:09.603667+00:00",
```

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

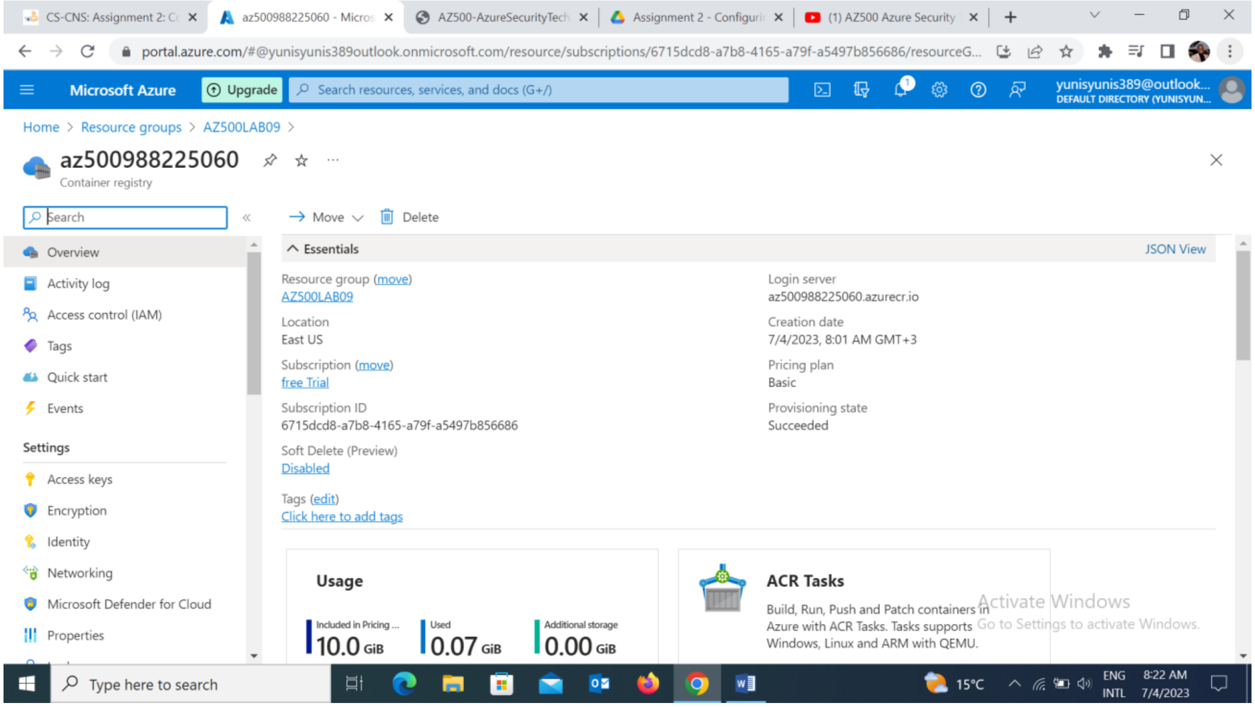
1. In the Bash session within the Cloud Shell pane, run the following to create a Dockerfile to create an Nginx-based image:
   echo FROM nginx > Dockerfile
2. In the Bash session within the Cloud Shell pane, run the following to build an image from the Dockerfile and push the image to the new ACR. ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)

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



```
yunis [ ~ ]$ ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)
yunis [ ~ ]$ 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_1400e6d05109404591260dae6a340244.tar.gz'...
Sending context (79.462 KiB) to registry: az500988225060...
Queued a build with ID: ca1
Waiting for an agent...
2023/07/04 05:14:21 Downloading source code...
2023/07/04 05:14:22 Finished downloading source code
2023/07/04 05:14:22 Using acb_vol_d16c55f1-7f4a-4773-b3f2-427d21c0dedf as the home volume
2023/07/04 05:14:22 Setting up Docker configuration...
2023/07/04 05:14:23 Successfully set up Docker configuration
2023/07/04 05:14:23 Logging in to registry: az500988225060.azurecr.io
2023/07/04 05:14:24 Successfully logged into az500988225060.azurecr.io
2023/07/04 05:14:24 Executing step ID: build. Timeout(sec): 28800, Working directory: '', Network: ''
2023/07/04 05:14:24 Scanning for dependencies...
2023/07/04 05:14:24 Successfully scanned dependencies
2023/07/04 05:14:24 Launching container with name: build
Sending build context to Docker daemon  468.5kB
Step 1/1 : FROM nginx
latest: Pulling from library/nginx
5b5fe70539cd: Pulling fs layer
441a1b465367: Pulling fs layer
3b9543f2b500: Pulling fs layer
ca89ed5461a9: Pulling fs layer
b0e1283145af: Pulling fs layer
4b98867cde79: Pulling fs layer
```

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.



az500988225060
Container registry

Resource group (move)
AZ500LAB09

Location
East US

Subscription (move)
free Trial

Subscription ID
6715dcd8-a7b8-4165-a79f-a5497b856686

Soft Delete (Preview)
Disabled

Tags (edit)
Click here to add tags

Login server
az500988225060.azurecr.io

Creation date
7/4/2023, 8:01 AM GMT+3

Pricing plan
Basic

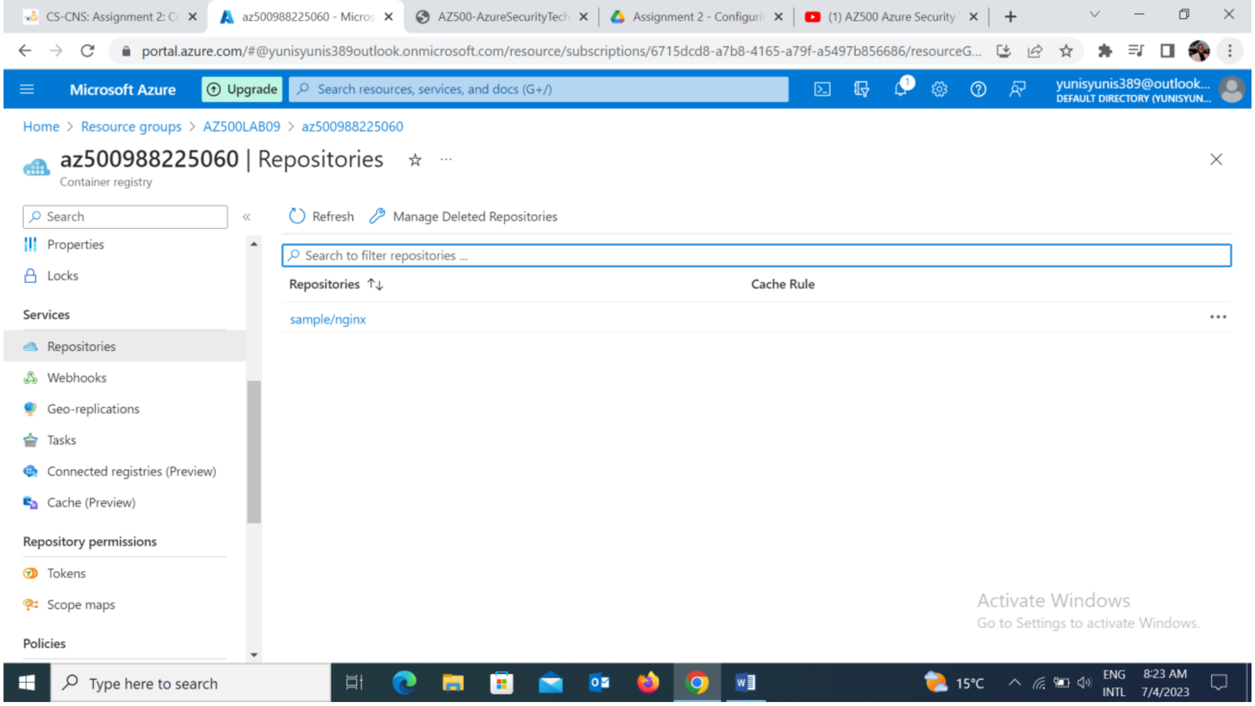
Provisioning state
Succeeded

Usage
Included in Pricing ... 10.0 GiB
Used 0.07 GiB
Additional storage 0.00 GiB

ACR Tasks
Build, Run, Push and Patch containers in Azure with ACR Tasks. Tasks supports Windows, Linux and ARM with QEMU.

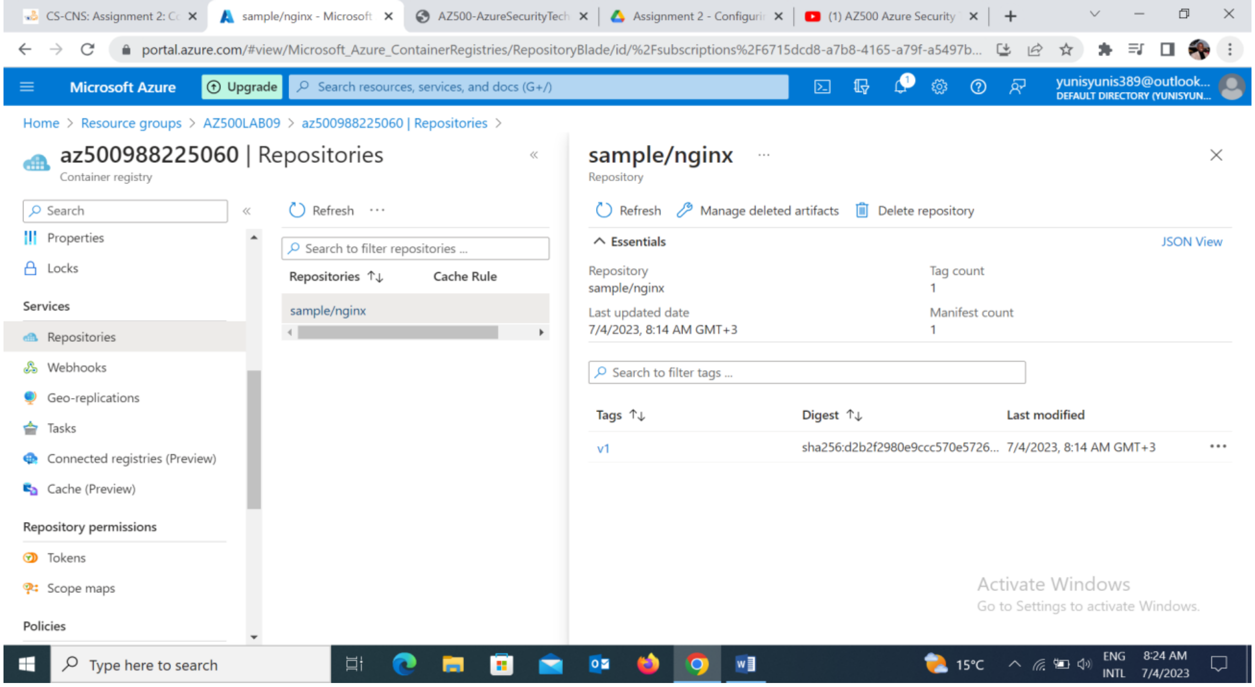
5. On the Container registry blade, in the **Services** section, click **Repositories**.

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


CS-CNS: Assignment 2: C
az500988225060 - Micros
AZ500-AzureSecurityTech
Assignment 2 - Configuri
(1) AZ500 Azure Security
portal.azure.com/#@yunisyunis389outlook.onmicrosoft.com/resource/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceG...
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
yunisyunis389@outlook...
DEFAULT DIRECTORY (YUNISYUN...
Home > Resource groups > AZ500LAB09 > az500988225060
az500988225060 | Repositories
Container registry
Search
Properties
Locks
Services
Repositories
Webhooks
Geo-replications
Tasks
Connected registries (Preview)
Cache (Preview)
Repository permissions
Tokens
Scope maps
Policies
Refresh
Manage Deleted Repositories
Search to filter repositories ...
Activate Windows
Go to Settings to activate Windows.
Type here to search
15°C
ENG INTL
8:23 AM
7/4/2023

Repositories ↑↓	Cache Rule
sample/nginx	

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



Repositories ↑↓	Cache Rule
sample/nginx	

Repository
sample/nginx

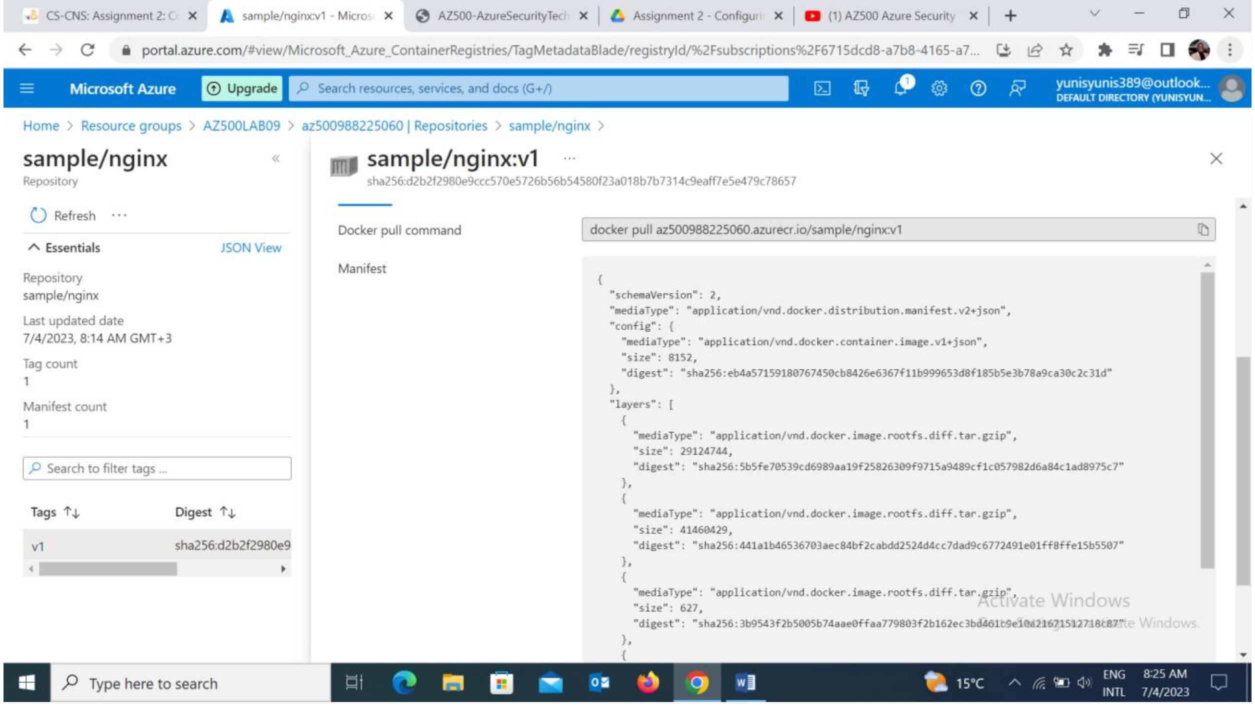
Tag count
1

Last updated date
7/4/2023, 8:14 AM GMT+3

Manifest count
1

Tags ↑↓	Digest ↑↓	Last modified
v1	sha256:d2b2f2980e9ccc570e5726...	7/4/2023, 8:14 AM GMT+3

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



Home > Resource groups > AZ500LAB09 > az500988225060 | Repositories > sample/nginx >

sample/nginx
Repository

Essentials

Repository
sample/nginx

Last updated date
7/4/2023, 8:14 AM GMT+3

Tag count
1

Manifest count
1

Tags	Digest
v1	sha256:d2b2f2980e9

sample/nginx:v1
sha256:d2b2f2980e9ccc570e5726b56b54580f23a018b7b7314c9eaff7e5e479c78657

Docker pull command: docker pull az500988225060.azurecr.io/sample/nginx:v1

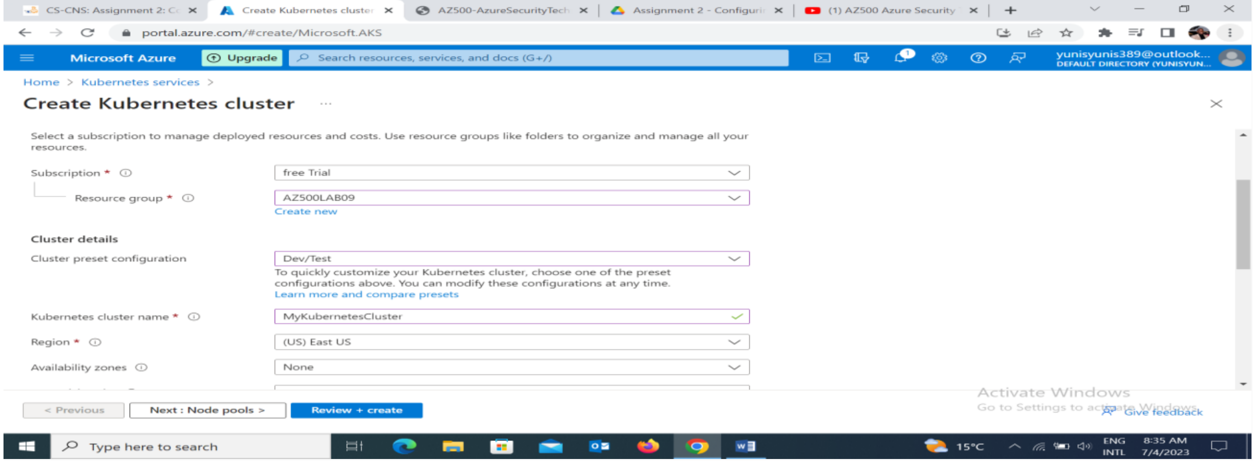
Manifest

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
  "config": {
    "mediaType": "application/vnd.docker.container.image.v1+json",
    "size": 8152,
    "digest": "sha256:eb4a57159180767450cb8426e6367f11b999653d8f185b5e3b78a9ca30c2c31d"
  },
  "layers": [
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 29124744,
      "digest": "sha256:5b5fe70539cd6989aa19f25826309f9715a9489cf1c057982d6a84c1ad8975c7"
    },
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 41460429,
      "digest": "sha256:441a1b46536703aec84bf2cabdd2524d4cc7dad9c6772491e01ff8ffe15b5507"
    },
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 627,
      "digest": "sha256:3b9543f2b5005b74aae0ffaa779803f2b162ec3bd461c9e10a21671512718c87"
    },
    {
```

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):



Home > Kubernetes services >

Create Kubernetes cluster

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: free Trial

Resource group *: AZ500LAB09

Create new

Cluster details

Cluster preset configuration: Dev/Test

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.
Learn more and compare presets

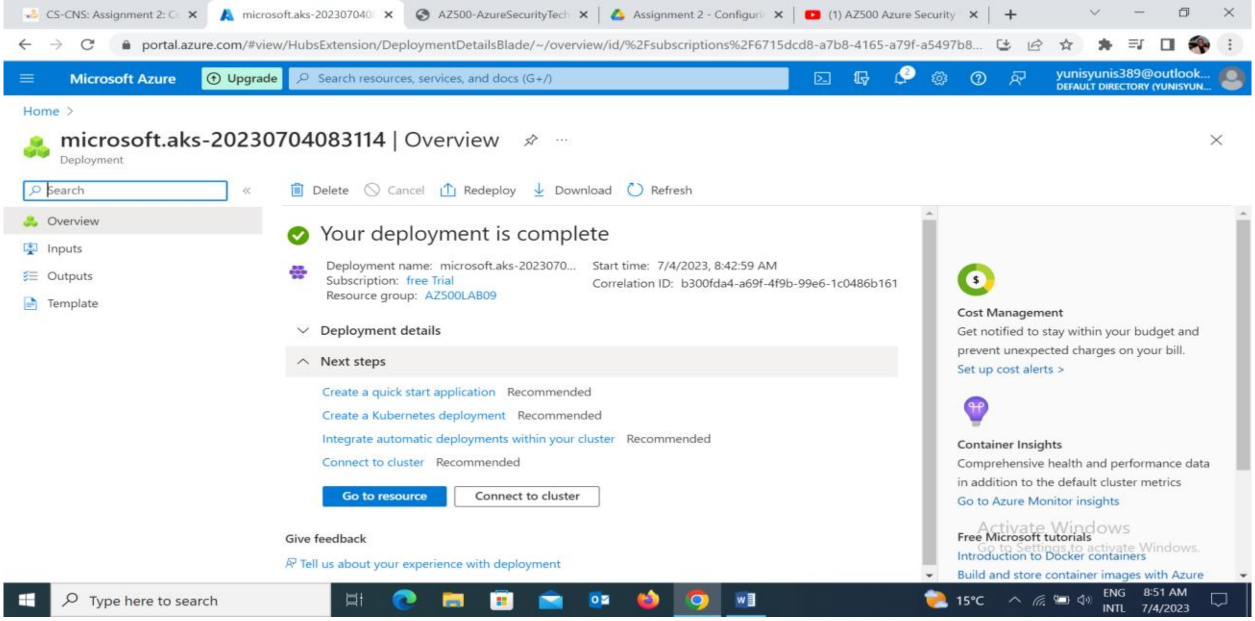
Kubernetes cluster name *: MyKubernetesCluster

Region *: (US) East US

Availability zones: None

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):

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):
7. Click **Next: Integrations >** and, on the **Integrations** tab of the **Create Kubernetes cluster** blade, set **Container monitoring** to **Disabled**.
8. Click **Review + Create** and then click **Create**.



portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F6715dcd8-a7b8-4165-a79f-a5497b8...

microsoft.aks-20230704083114 | Overview

Deployment

Your deployment is complete

Deployment name: microsoft.aks-2023070...
Subscription: free Trial
Resource group: AZ500LAB09

Start time: 7/4/2023, 8:42:59 AM
Correlation ID: b300fda4-a69f-4f9b-99e6-1c0486b161

Deployment details

Next steps

Create a quick start application Recommended
Create a Kubernetes deployment Recommended
Integrate automatic deployments within your cluster Recommended
Connect to cluster Recommended

Give feedback

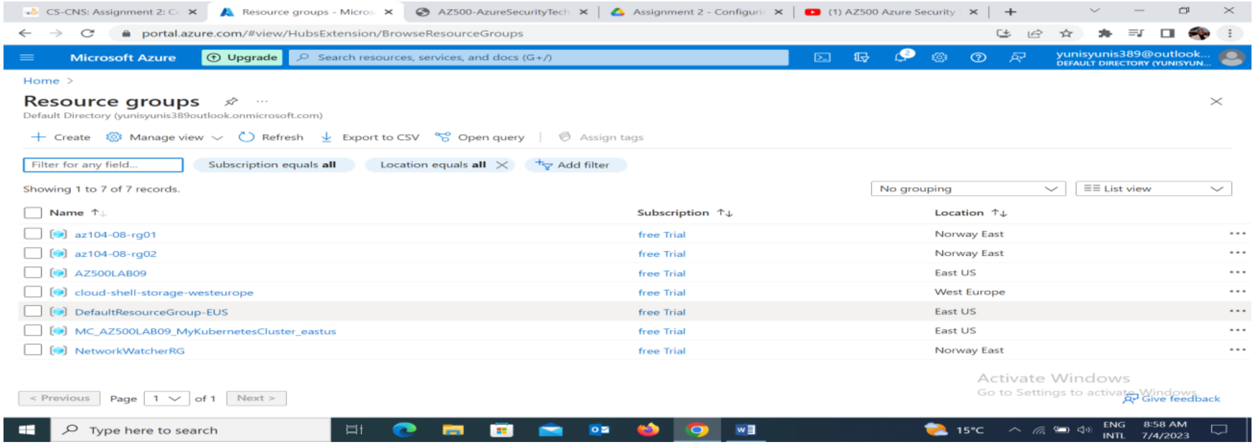
Tell us about your experience with deployment

Cost Management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Container Insights
Comprehensive health and performance data in addition to the default cluster metrics
Go to Azure Monitor insights

Free Microsoft tutorials
Introduction to Docker containers
Build and store container images with Azure

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.



portal.azure.com/#view/HubsExtension/BrowseResourceGroups

Resource groups

Default Directory (yunisyunis389outlook.onmicrosoft.com)

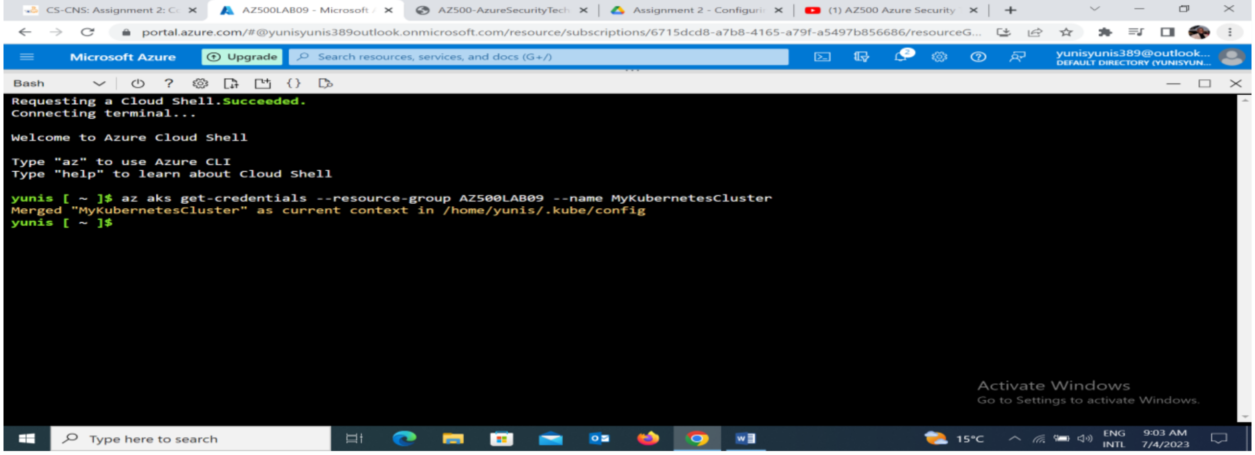
Filter for any field... | Subscription equals **all** | Location equals **all** | Add filter

Showing 1 to 7 of 7 records.

Name	Subscription	Location
az104-08-rg01	free Trial	Norway East
az104-08-rg02	free Trial	Norway East
AZ500LAB09	free Trial	East US
cloud-shell-storage-westeurope	free Trial	West Europe
DefaultResourceGroup-EUS	free Trial	East US
MC_AZ500LAB09_MyKubernetesCluster_eastus	free Trial	East US
NetworkWatcherRG	free Trial	Norway East

11. Navigate back to the **Resource groups** blade and click the **AZ500LAB09** entry.
12. In the Azure portal, open a Bash session in the Cloud Shell.
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

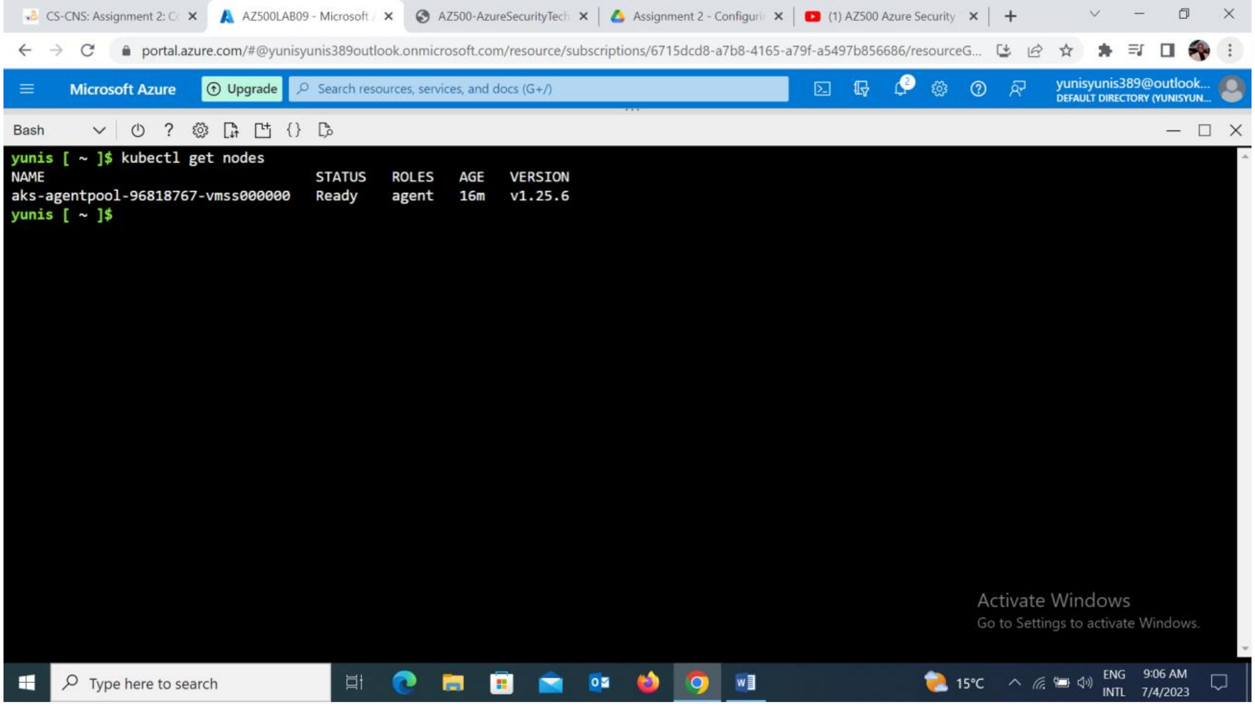


CS-CNS: Assignment 2: C
AZ500LAB09 - Microsoft
AZ500-AzureSecurityTech
Assignment 2 - Configuri
(1) AZ500 Azure Security
portal.azure.com/#@yunisyunis389outlook.onmicrosoft.com/resource/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceG...
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
yunisyunis389@outlook...
DEFAULT DIRECTORY (YUNISYUN...
Bash
Activate Windows
Go to Settings to activate Windows.
Type here to search
15°C
ENG
INTL
9:03 AM
7/4/2023

```
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

yunis [ ~ ]$ az aks get-credentials --resource-group AZ500LAB09 --name MyKubernetesCluster
Merged "MyKubernetesCluster" as current context in /home/yunis/.kube/config
yunis [ ~ ]$
```

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

    kubectl get nodes

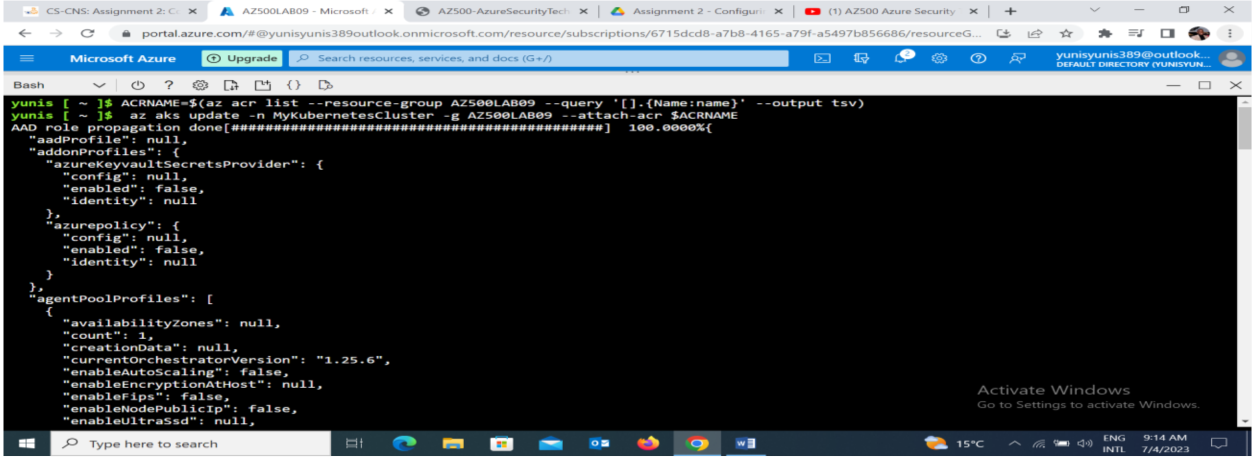


CS-CNS: Assignment 2: C
AZ500LAB09 - Microsoft
AZ500-AzureSecurityTech
Assignment 2 - Configuri
(1) AZ500 Azure Security
portal.azure.com/#@yunisyunis389outlook.onmicrosoft.com/resource/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceG...
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
yunisyunis389@outlook...
DEFAULT DIRECTORY (YUNISYUN...
Bash
Activate Windows
Go to Settings to activate Windows.
Type here to search
15°C
ENG
INTL
9:06 AM
7/4/2023

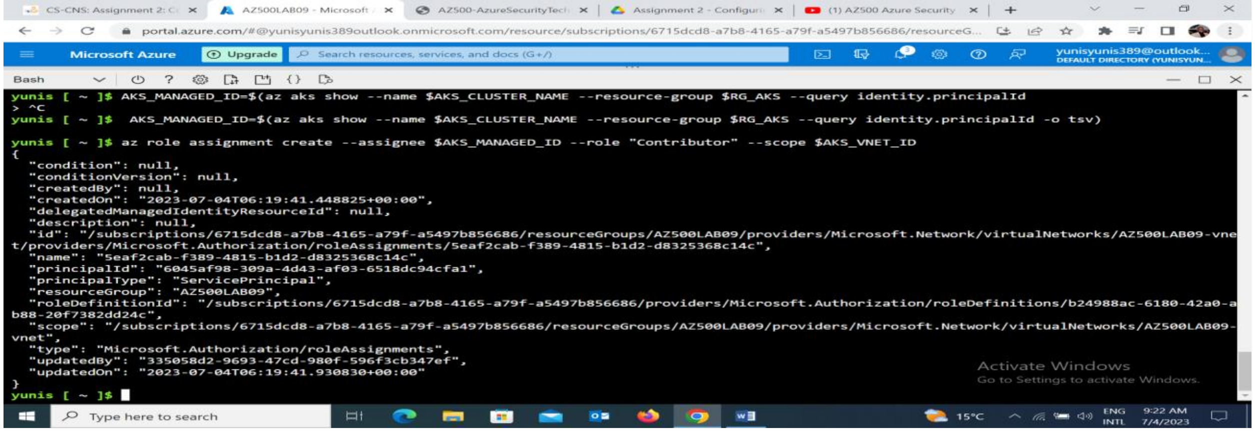
```
yunis [ ~ ]$ kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-96818767-vmss000000   Ready    agent   16m   v1.25.6
yunis [ ~ ]$
```

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
   

```
yunis [ ~ ]$ ACRNAME=$(az acr list --resource-group AZ500LAB09 --query '[].{Name:name}' --output tsv)
yunis [ ~ ]$  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": 1,
      "creationData": null,
      "currentOrchestratorVersion": "1.25.6",
      "enableAutoScaling": false,
      "enableEncryptionAtHost": null,
      "enableFips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": null,
```

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
   

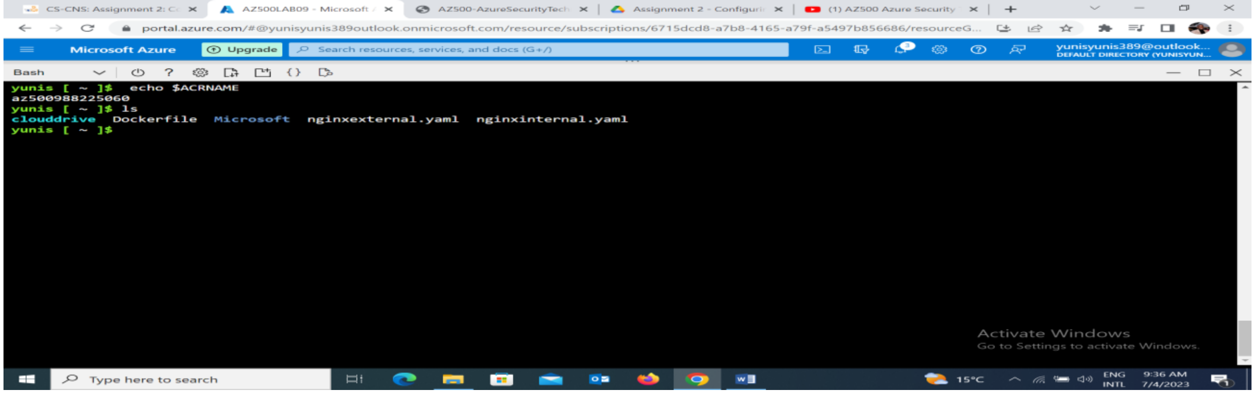
```
yunis [ ~ ]$ AKS_MANAGED_ID=$(az aks show --name $AKS_CLUSTER_NAME --resource-group $RG_AKS --query identity.principalId
> ^C
yunis [ ~ ]$  AKS_MANAGED_ID=$(az aks show --name $AKS_CLUSTER_NAME --resource-group $RG_AKS --query identity.principalId -o tsv)

yunis [ ~ ]$ az role assignment create --assignee $AKS_MANAGED_ID --role "Contributor" --scope $AKS_VNET_ID
{
  "condition": null,
  "conditionVersion": null,
  "createdBy": null,
  "createdOn": "2023-07-04T06:19:41.448825+00:00",
  "delegatedManagedIdentityResourceId": null,
  "description": null,
  "id": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB09/providers/Microsoft.Network/virtualNetworks/AZ500LAB09-vne
t/providers/Microsoft.Authorization/roleAssignments/5eaf2cab-f389-4815-b1d2-d8325368c14c",
  "name": "5eaf2cab-f389-4815-b1d2-d8325368c14c",
  "principalId": "6045af98-309a-4d43-af03-6518dc94cfa1",
  "principalType": "ServicePrincipal",
  "resourceGroup": "AZ500LAB09",
  "roleDefinitionId": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/providers/Microsoft.Authorization/roleDefinitions/b24988ac-6180-42a0-a
b88-20f7382dd24c",
  "scope": "/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB09/providers/Microsoft.Network/virtualNetworks/AZ500LAB09-
vnet",
  "type": "Microsoft.Authorization/roleAssignments",
  "updatedBy": "335058d2-9693-47cd-980f-596f3cb347ef",
  "updatedOn": "2023-07-04T06:19:41.930830+00:00"
}
yunis [ ~ ]$
```

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.

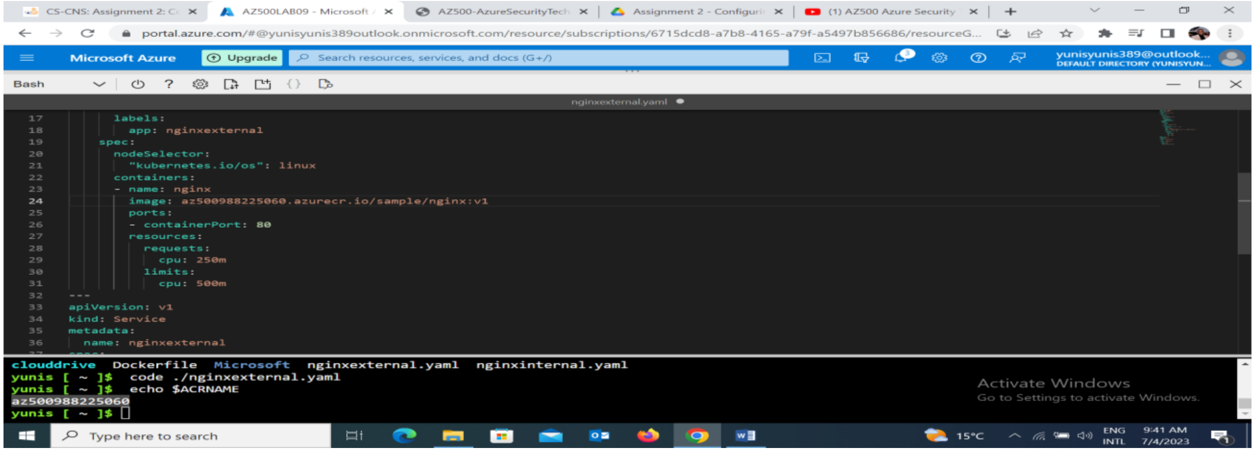
1. In the Bash session within the Cloud Shell pane, click the **Upload/Download files** icon, in the drop-down menu, click **Upload**, in the **Open** dialog box, naviate to the location where you downloaded the lab files, select **\Allfiles\Labs\09\nginxexternal.yaml** click **Open**. Next, select **\Allfiles\Labs\09\nginxinternal.yaml**,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



```
yunis [ ~ ]$  echo $ACRNAME
az500988225060
yunis [ ~ ]$ ls
clouddrive  Dockerfile  Microsoft  nginxexternal.yaml  nginxinternal.yaml
yunis [ ~ ]$
```

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
4. In the editor pane, scroll down to **line 24** and replace the <ACRUniquename> placeholder with the ACR name. **az500988225060**
5. In the editor pane, in the upper right corner, click the **ellipses** icon, click **Save** and then click **Close editor**.



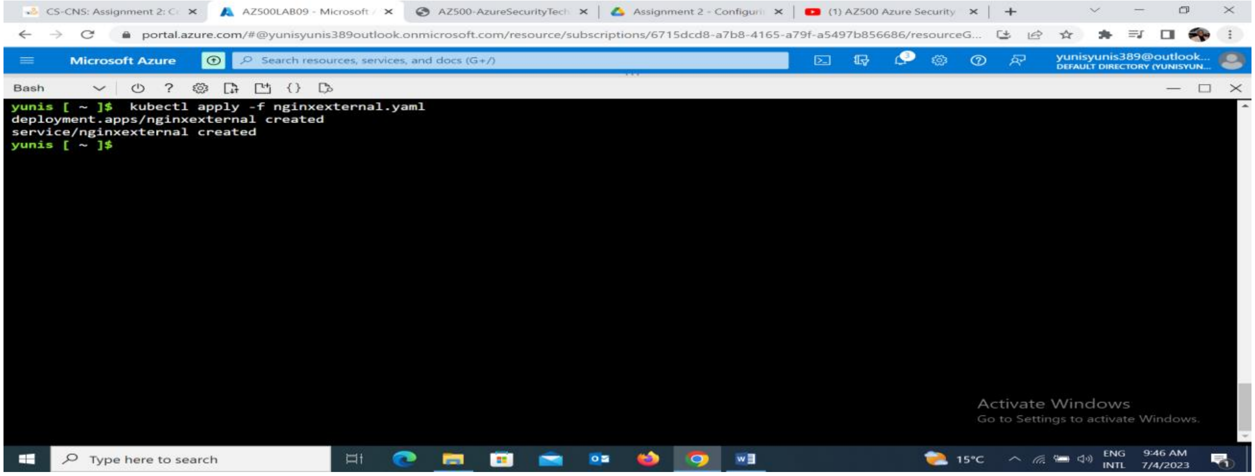
nginxexternal.yaml

```
17        labels:
18          app: nginxexternal
19      spec:
20        nodeSelector:
21          "kubernetes.io/os": linux
22        containers:
23        - name: nginx
24          image: az500988225060.azurecr.io/sample/nginx:v1
25          ports:
26          - containerPort: 80
27          resources:
28            requests:
29              cpu: 250m
30            limits:
31              cpu: 500m
32  ---
33  apiVersion: v1
34  kind: Service
35  metadata:
36    name: nginxexternal
```

```
clouddrive  Dockerfile  Microsoft  nginxexternal.yaml  nginxinternal.yaml
yunis [ ~ ]$  code ./nginxexternal.yaml
yunis [ ~ ]$  echo $ACRNAME
az500988225060
yunis [ ~ ]$ 
```

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



```
yunis [ ~ ]$  kubectl apply -f nginxexternal.yaml
deployment.apps/nginxexternal created
service/nginxexternal created
yunis [ ~ ]$
```

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

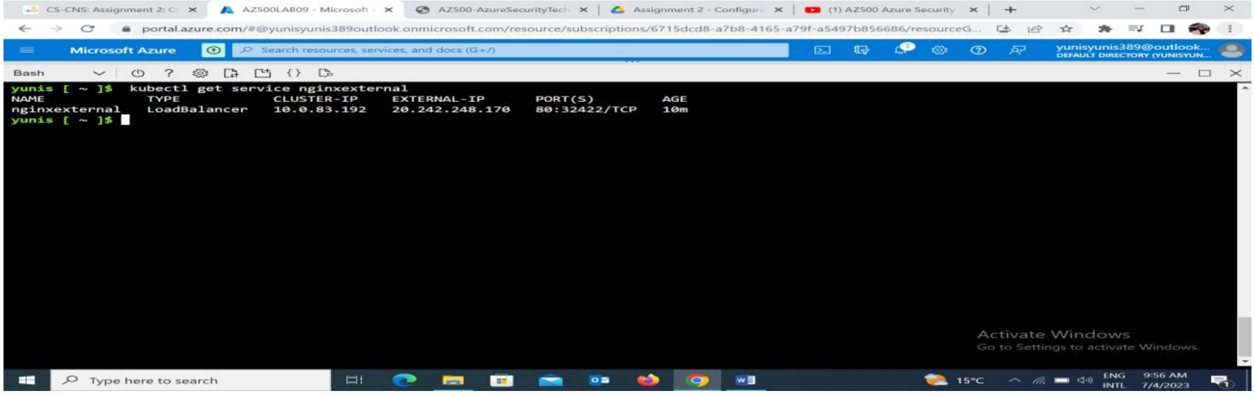
   service/nginxexternal created

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.

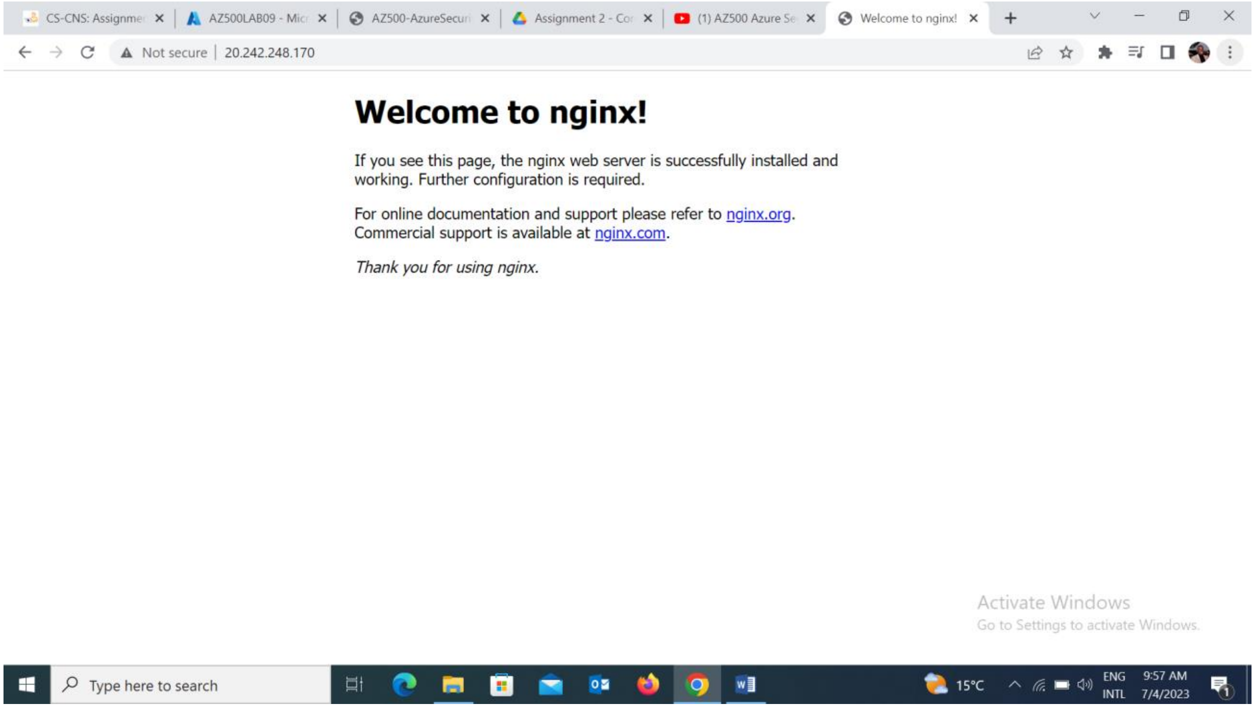
1. In the Bash session within the Cloud Shell pane, run the following to retrieve information about the nginxexternal service including name, type, IP addresses, and ports.

   kubectl get service nginxexternal



```
yunis [ ~ ]$  kubectl get service nginxexternal
NAME            TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)        AGE
nginxexternal   LoadBalancer   10.0.83.192   20.242.248.170  80:32422/TCP   10m
yunis [ ~ ]$
```

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.
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.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Task 7: Deploy an internal service to AKS

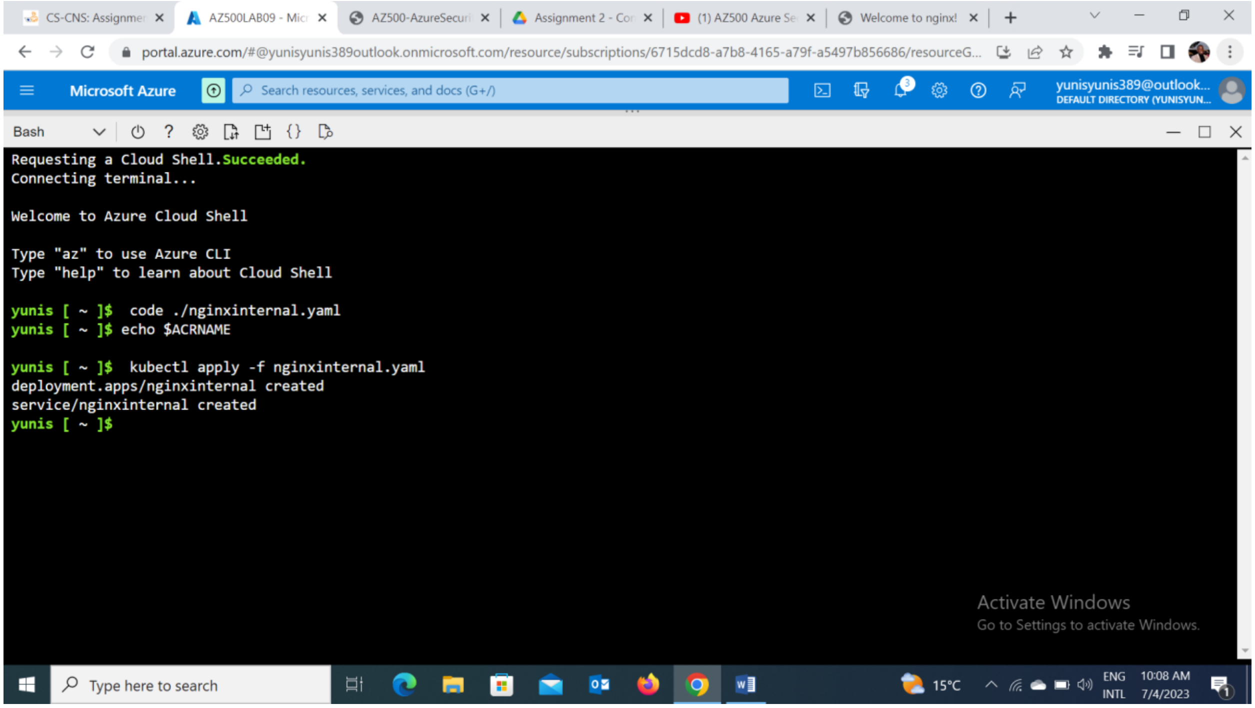
In this task, you will deploy the internal facing service on the AKS.

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
```

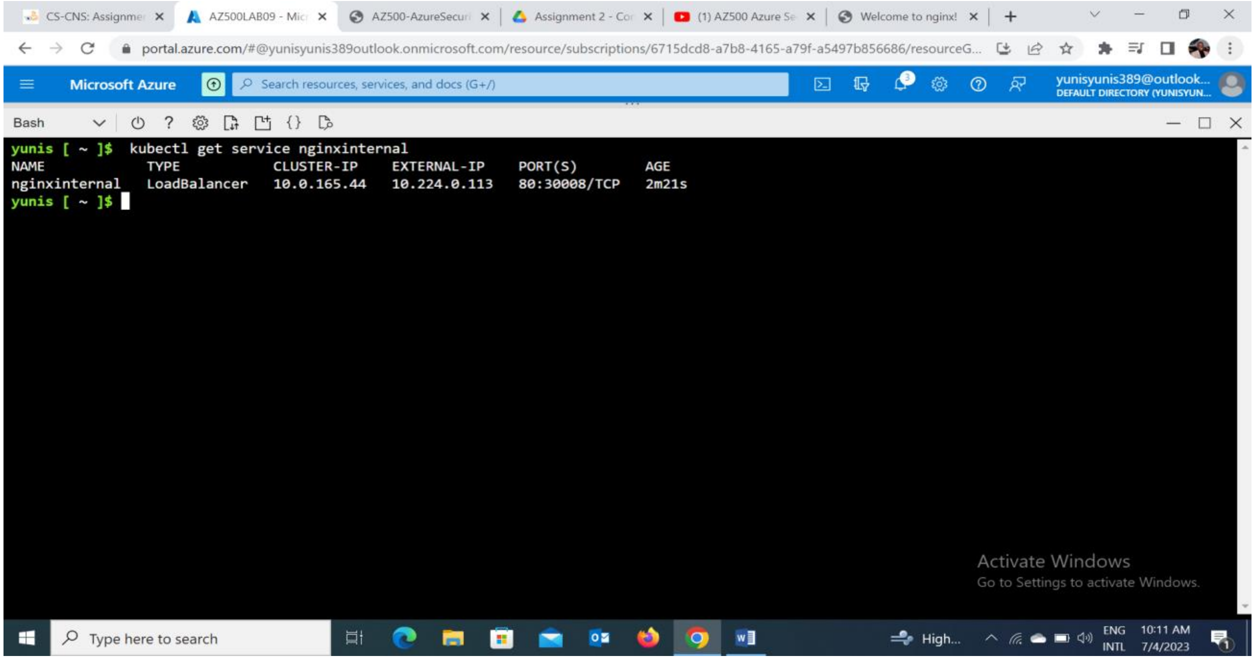
2. In the editor pane, scroll down to the line containing the reference to the container image and replace the **<ACRUniquename>** placeholder with the ACR name.
3. In the editor pane, in the upper right corner, click the **ellipses** icon, click **Save** and then click **Close editor**.
4. In the Bash session within the Cloud Shell pane, run the following to apply the change to the cluster:

```
kubectl apply -f nginxinternal.yaml
```



```
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

yunis [ ~ ]$  code ./nginxinternal.yaml
yunis [ ~ ]$ echo $ACRNAME

yunis [ ~ ]$  kubectl apply -f nginxinternal.yaml
deployment.apps/nginxinternal created
service/nginxinternal created
yunis [ ~ ]$
```

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



```
yunis [ ~ ]$  kubectl get service nginxinternal
NAME           TYPE          CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
nginxinternal  LoadBalancer  10.0.165.44   10.224.0.113  80:30008/TCP   2m21s
yunis [ ~ ]$
```

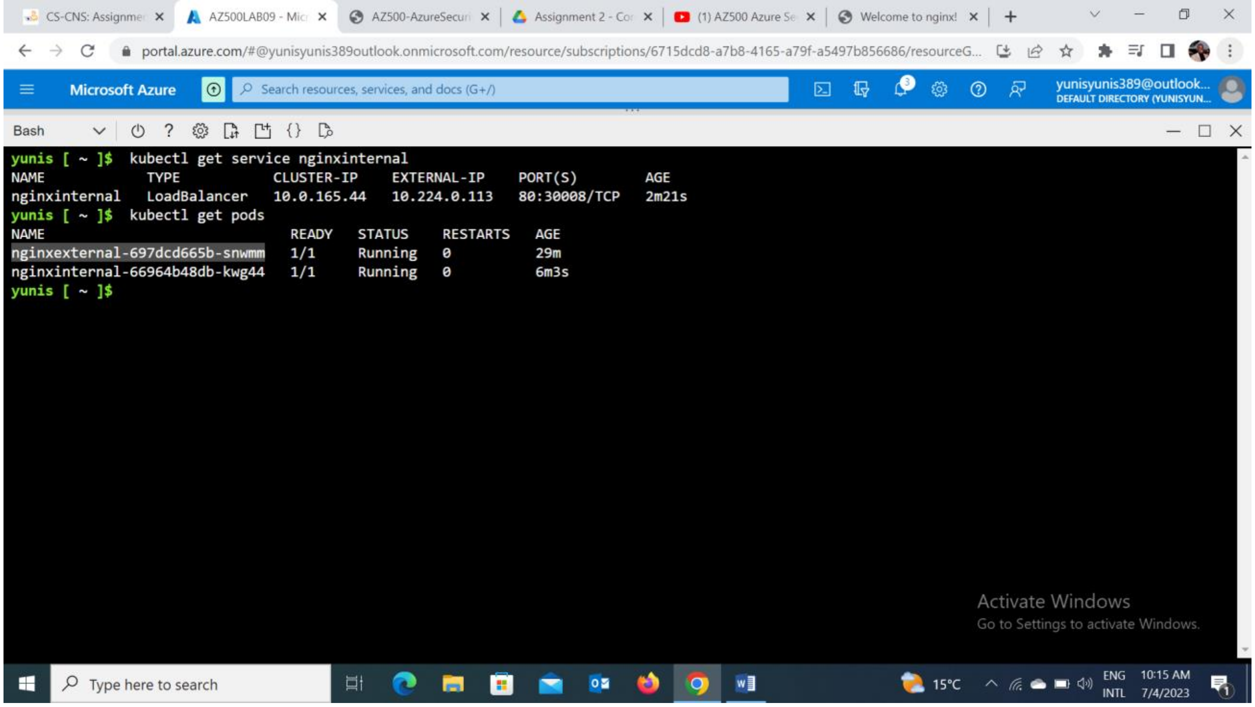
Task 8: Verify the you can access an internal AKS-hosted service

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.

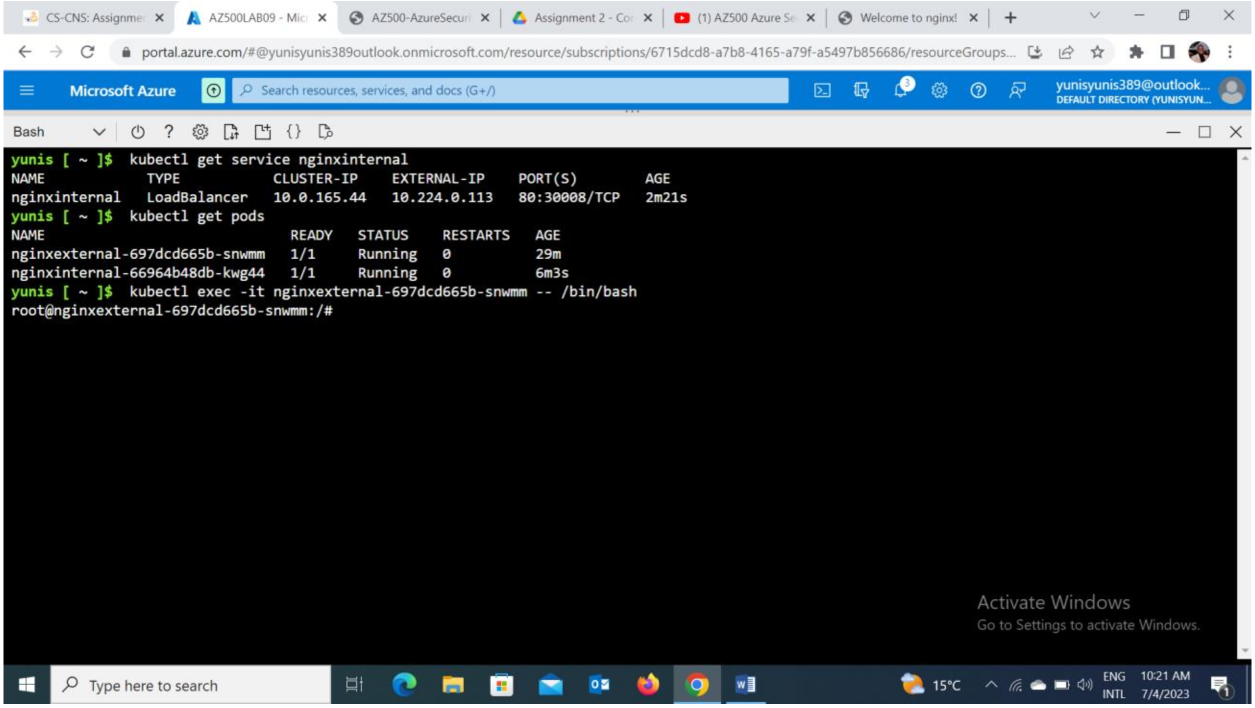


```
yunis [ ~ ]$  kubectl get service nginxinternal
NAME            TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)        AGE
nginxinternal   LoadBalancer   10.0.165.44   10.224.0.113   80:30008/TCP   2m21s
yunis [ ~ ]$  kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
nginxexternal-697dcd665b-snwmm   1/1     Running   0          29m
nginxinternal-66964b48db-kwg44   1/1     Running   0          6m3s
yunis [ ~ ]$
```

portal.azure.com/#@yunisyunis389outlook.onmicrosoft.com/resource/subscriptions/6715dcd8-a7b8-4165-a79f-a5497b856686/resourceG...
Microsoft Azure
Search resources, services, and docs (G+/)
yunisyunis389@outlook...
DEFAULT DIRECTORY (YUNISYUN...
Bash
Activate Windows
Go to Settings to activate Windows.
Type here to search
15°C
ENG INTL
10:15 AM
7/4/2023

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):

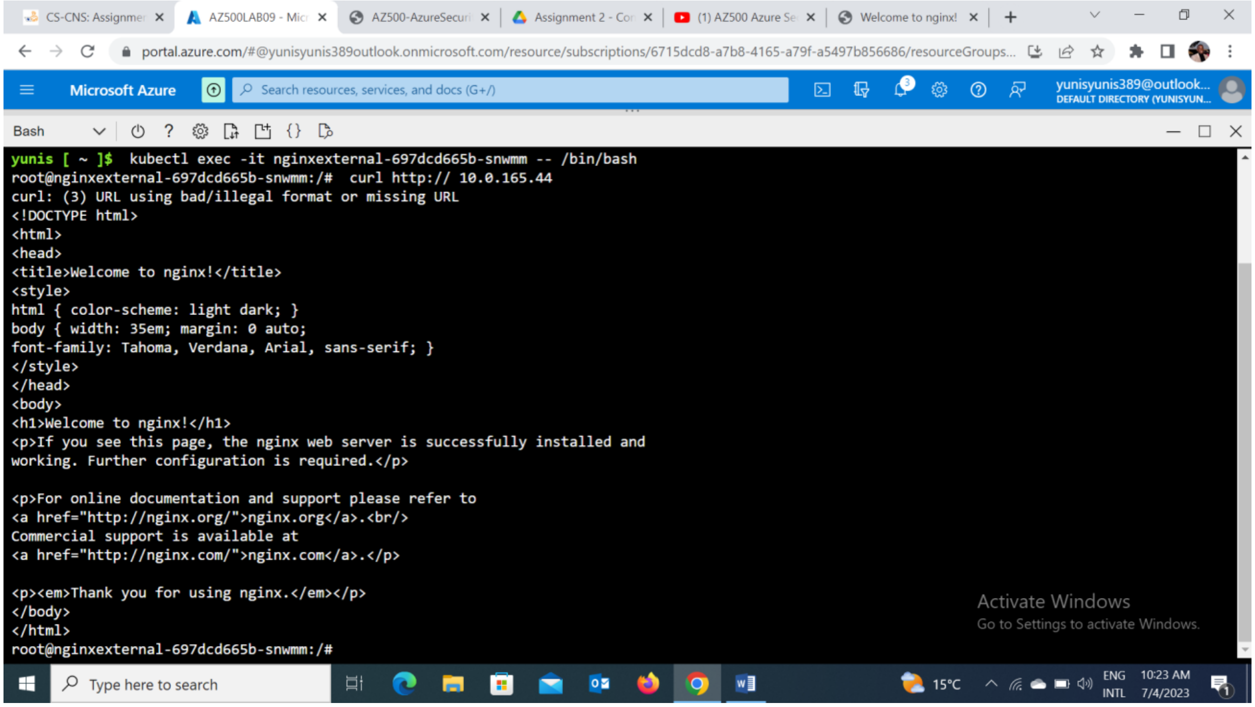
   kubectl exec -it <pod_name> -- /bin/bash



```
yunis [ ~ ]$  kubectl get service nginxinternal
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
nginxinternal  LoadBalancer   10.0.165.44    10.224.0.113   80:30008/TCP   2m21s
yunis [ ~ ]$  kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
nginxexternal-697dcd665b-snwmm    1/1     Running   0          29m
nginxinternal-66964b48db-kwg44    1/1     Running   0          6m3s
yunis [ ~ ]$  kubectl exec -it nginxexternal-697dcd665b-snwmm -- /bin/bash
root@nginxexternal-697dcd665b-snwmm:/#
```

4. In the Bash session within the Cloud Shell pane, run the following to verify that the nginx website 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>



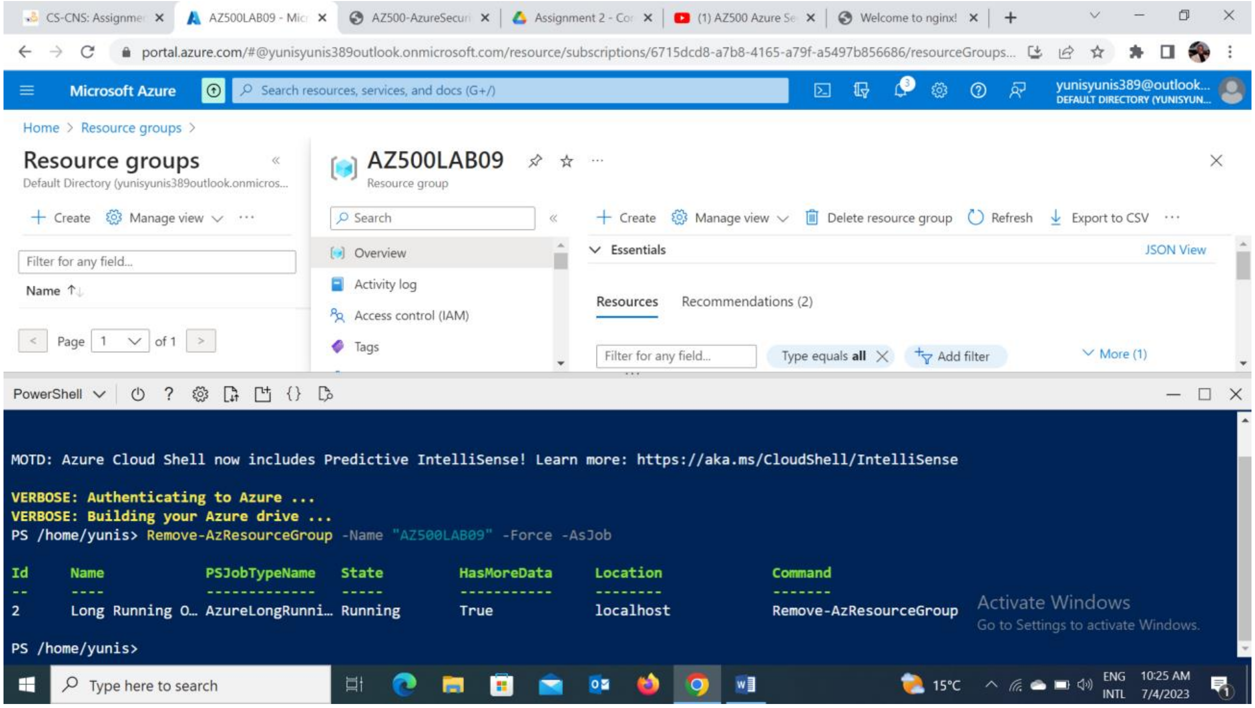
```
yunis [ ~ ]$  kubectl exec -it nginxexternal-697dcd665b-snwmm -- /bin/bash
root@nginxexternal-697dcd665b-snwmm:/#  curl http:// 10.0.165.44
curl: (3) URL using bad/illegal format or missing URL
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@nginxexternal-697dcd665b-snwmm:/#
```

5. Close the Cloud Shell pane.

Clean up resources

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



```
MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/yunis> Remove-AzResourceGroup -Name "AZ500LAB09" -Force -AsJob

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
2      Long Running O… AzureLongRunni… Running       True            localhost            Remove-AzResourceGroup

PS /home/yunis>
```

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

Conclusion

In conclusion, Microsoft Azure Lab 09 has provided me with valuable insights into the configuration and security aspects of Azure Container Registry (ACR) and Azure Kubernetes Service (AKS). By implementing the best practices learned during the lab, organizations can optimize the performance and enhance the security of their container-based deployments in Azure. Configuring ACR effectively ensures the reliable storage and management of container images, while securing AKS safeguards the environment for deploying and managing containerized applications. By prioritizing effective configuration and robust security measures, organizations can mitigate potential risks and vulnerabilities, enabling them to fully leverage the benefits of ACR and AKS. This lab has equipped me with the necessary knowledge to establish a solid foundation for deploying and securing container workloads in Azure, empowering organizations to drive innovation and achieve success in their cloud-based initiatives.