In this lab, you will learn how to update your AKS cluster using Azure CLI.
A new AKS cluster is automatically created for you. Use the following command to list all the AKS clusters in your temporary Azure subscription:
az aks list --query "[].{Name: name}"

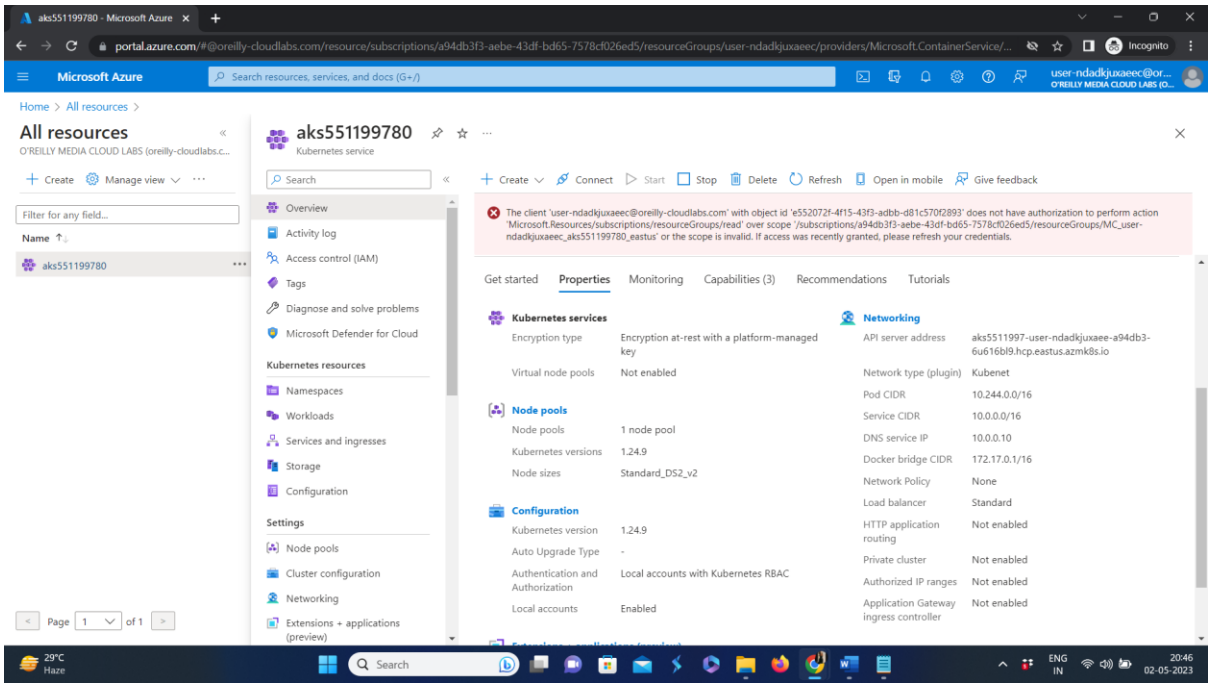
Create aks cluster .
az aks create --resource-group $resource --name $aksName --kubernetes-version 1.24.9

```
$ az aks create --resource-group $resource --name $aksName --kubernetes-version 1.24.9
- Running ..
```

Use the following command to list all the AKS clusters in your temporary Azure subscription:

az aks list --query "[].{Name: name}"

```
$ az aks list --query "[].{Name: name}"
[
  {
    "Name": "aks551199780"
  }
]
```



Understanding Updating an Existing AKS Cluster
A deployed AKS cluster can be managed in the Azure portal or programmatically using tools such as Azure CLI. Big part of managing a cluster is updating its properties.

You can use the az aks update CLI command to update several properties of your AKS cluster. This command has around 50 parameters, and we will list a few of them below.

--api-server-authorized-ip-ranges: Call the API server to perform actions in the AKS cluster. You can limit the clients which are allowed to call the API server to harden your cluster security. Use this parameter to pass allowed IP addresses.
--assign-identity: Attach an existing user-assigned identity to the cluster. This enables the cluster to perform management actions on other Azure resources and cluster dependencies.

--enable-defender: Enable Microsoft Defender for containers. This will protect your cluster from security attacks.
--enable-azure-rbac: Enable role-based access control (RBAC) for you AKS cluster. This allows Azure Active Directory users or groups to manage the cluster based on the permissions or roles they get.
--enable-cluster-autoscaler: Enable AKS cluster autoscale so the number of nodes can increase when needed.
--uptime-sla: For an extra cost, enable a financially backed SLA (service-level agreement) for your cluster, meaning you will get credits if the AKS cluster is down more than the promised SLA.
--windows-admin-password: List the user account password to use on Windows nodes. Note that this will not affect or work on Linux nodes.
Now let's update AKS in the next step.

Updating Our AKS Cluster
In this step, we will use Azure CLI to update our existing AKS cluster. We will:

Add a taxonomy tag to the AKS resource.
Enable role-based access control (RBAC) on the cluster.
Note that in order to enable RBAC, Azure Active Directory (AAD) support needs to be enabled too. If not, you will get the following error message when trying to enable RBAC support on your cluster:

Use the following command to perform the update:

az aks update --resource-group $resource --name $aksName --enable-aad --enable-azure-rbac --tags "env=dev"

Wait for the command to finish executing.

```
2388rwn4MnadVdX0
$ az aks update -g $resource --name $aksName --enable-aad --enable-azure-rbac --tags "env=dev"
{
  "aadProfile": {
    "adminGroupObjectIDs": null,
    "adminUsers": null,
    "clientAppId": null,
    "enableAzureRbac": true,
    "managed": true,
    "serverAppId": null,
    "serverAppSecret": null,
    "tenantId": "9a8cd433-6113-49e5-aa7f-42788a01a246"
  },
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "count": 3,
      "creationData": null,
      "currentOrchestratorVersion": "1.24.9",
      "enableAutoScaling": false,
      "enableEncryptionAtHost": false,
      "enableFips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": false,
      "gpuInstanceProfile": null,
      "hostGroupId": null,
      "kubeletConfig": null,
      "kubeletDiskType": "OS",
      "linuxOsConfig": null,
```

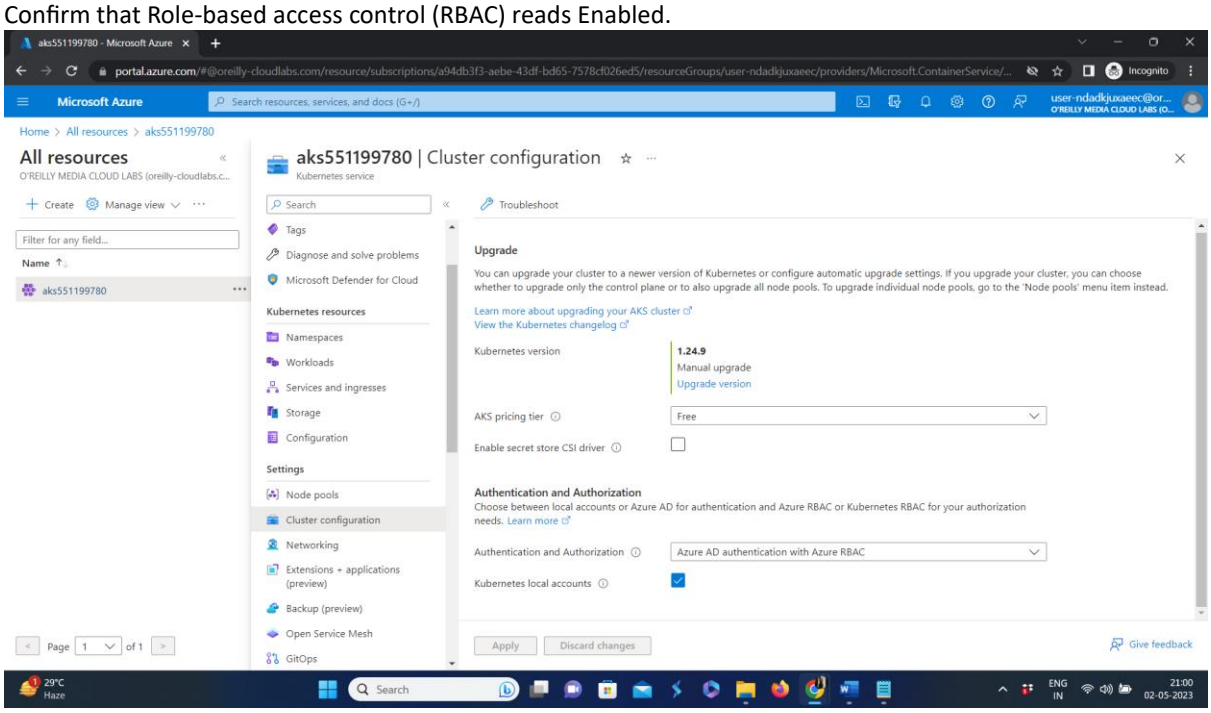
Confirming the AKS Cluster Is Updated
Use the following command to get the AKS cluster details or properties:

az aks show --name $aksName --resource-group $resource --query "{Name: name, EnableRbac: enableRbac, Tags: tags}"

Confirm that EnableRbac reads true and that the env tag is present under the Tags property.

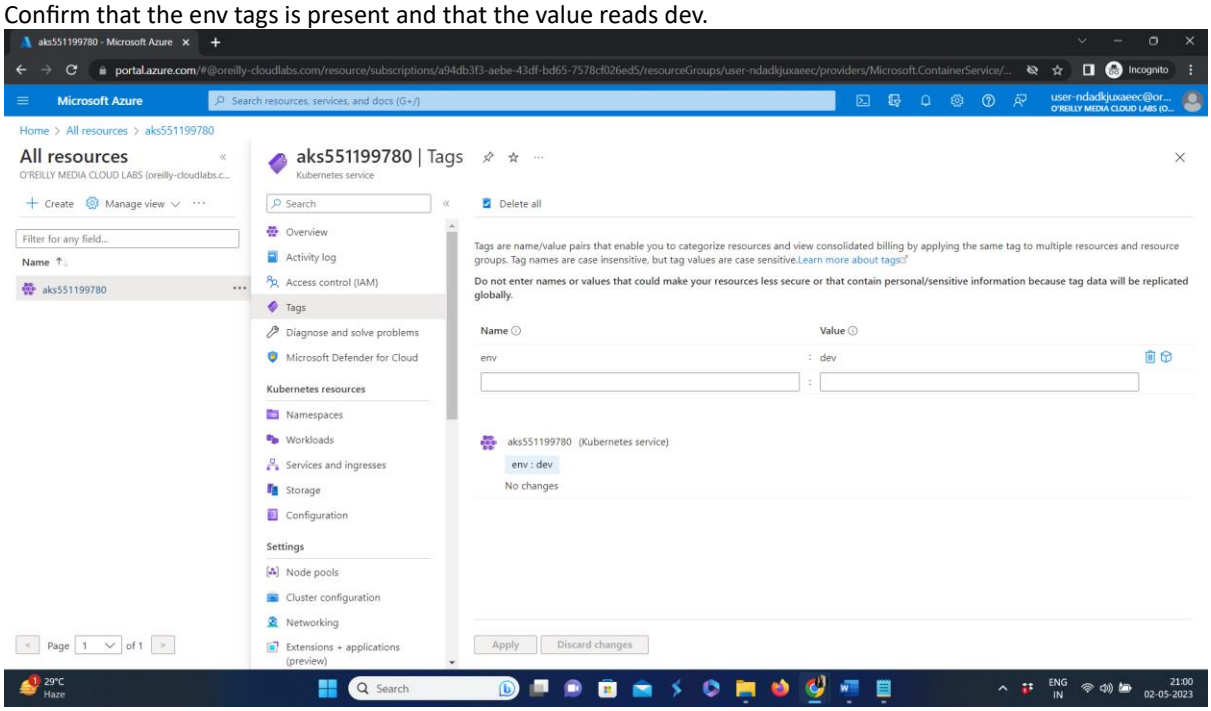
```
$ az aks show --name $aksName --resource-group $resource --query "{Name: name, EnableRbac: enableRbac, Tags: tags}"
{
  "EnableRbac": true,
  "Name": "aks551199780",
  "Tags": {
    "env": "dev"
  }
}
$
```

Click on your new AKS cluster name in the list (you should see only one).
Under Settings, click on Cluster Configuration.
Confirm that Role-based access control (RBAC) reads Enabled.



Confirm that AKS-managed Azure Active Directory reads Enabled.

From the mid-left menubar, click on Tags.
Confirm that the env tags is present and that the value reads dev.



Note: Azure AKS creates a new resource group which includes the AKS dependencies. On the Overview page, you might see a warning which contains "does not have authorization to perform action 'Microsoft.Resources/subscriptions/resourceGroups/read' over scope." This is because your temporary user does not have access to read other resource group details due to security reasons. This error is perfectly fine and does not interfere with our labs

Deleting the AKS Cluster
The following command will clean up the new AKS cluster:

```
az aks delete --name $aksName --resource-group $resource
```