**Activity: Deploy an AKS Cluster Using Azure Portal, Access the Dashboard, and Create Roles for Multiple Users**

**➤ Theory**

**Azure Kubernetes Service (AKS)** is Microsoft’s managed Kubernetes service that simplifies deploying, managing, and operating Kubernetes clusters in Azure. It abstracts away the complexity of setting up Kubernetes and provides scalability, monitoring, and security capabilities out of the box.

In this activity, we will:

* Deploy an AKS cluster using the Azure portal.
* Access the Kubernetes Dashboard.
* Create and assign role-based access control (RBAC) roles to different users.

**➤ Steps to Deploy an AKS Cluster using Azure Portal**

**✅ 1. Login to Azure Portal**

Go to: <https://portal.azure.com>

**✅ 2. Create a Resource Group**

* Search for “Resource Groups”.
* Click **Create** → Enter:
  + Resource Group Name: aks-rg
  + Region: East US (or your preferred region)
* Click **Review + Create** → **Create**.

**✅ 3. Deploy AKS Cluster**

* Search for **Kubernetes Services**.
* Click **Create** → **Create Kubernetes Cluster**.
* Fill in the details:
  + **Cluster Name:** aks-demo
  + **Region:** Same as Resource Group
  + **Node Size:** Standard\_DS2\_v2 (or smaller for testing)
  + **Node Count:** 1 or 2
* Under **Authentication**, enable RBAC.
* Leave other defaults and click **Review + Create**, then **Create**.

✔️ It will take ~5–10 minutes for the cluster to be ready.

**➤ Steps to Access the Kubernetes Dashboard**

**✅ 1. Open Cloud Shell**

* Click on the **Cloud Shell** icon in Azure Portal (top navigation bar).
* Choose **Bash** shell.

**✅ 2. Connect to AKS Cluster**

Run this command to configure kubectl:

bash

CopyEdit

az aks get-credentials --resource-group aks-rg --name aks-demo

**✅ 3. Enable Kubernetes Dashboard Add-on (optional in new AKS versions):**

bash

CopyEdit

az aks enable-addons --addons kube-dashboard --resource-group aks-rg --name aks-demo

**✅ 4. Access the Dashboard**

Run:

bash

CopyEdit

kubectl proxy

Open this URL in your browser:

bash

CopyEdit

http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/

**➤ Steps to Create Roles for Multiple Users**

**✅ 1. Create Role YAML File (role.yaml)**

Example Role to allow listing pods:

yaml

CopyEdit

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

namespace: default

name: pod-reader

rules:

- apiGroups: [""]

resources: ["pods"]

verbs: ["get", "watch", "list"]

Apply the role:

bash

CopyEdit

kubectl apply -f role.yaml

**✅ 2. Create RoleBinding for a User (rolebinding.yaml)**

Example RoleBinding for a user:

yaml

CopyEdit

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: read-pods-binding

namespace: default

subjects:

- kind: User

name: "john@example.com" # Replace with the Azure AD user email

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: Role

name: pod-reader

apiGroup: rbac.authorization.k8s.io

Apply the RoleBinding:

bash

CopyEdit

kubectl apply -f rolebinding.yaml

**✅ 3. Repeat for Multiple Users**

You can create separate RoleBindings for other users like:

* raj@example.com
* emma@example.com

**➤ Conclusion**

✔️ Successfully deployed an AKS cluster using Azure Portal.

✔️ Accessed the Kubernetes dashboard using kubectl proxy.

✔️ Created RBAC roles and role bindings to control user access.

**➤ Additional Useful Commands**

bash

CopyEdit

kubectl get pods --all-namespaces

kubectl get roles --namespace default

kubectl get rolebindings --namespace default