Deploy an AKS cluster using the portal. Access the dashboard and create roles for multiple users

To deploy an Azure Kubernetes Service (AKS) cluster using the Azure portal first we will have to deploy an AKS cluster Azure AD Integration using follow steps:

1) Sign in to the Azure Portal.
2) Create an AKS Cluster :
   - Navigate to Create a resource > Kubernetes Service.
   - Subscription, Resource Group, Cluster name, Region.
    - Select Kubernetes version (default is fine).
   - Configure **Primary Node Pool**:
     - Node size, count (e.g., 2 nodes), and availability zones (optional).
   - Authentication:
     - Enable Azure AD integration
     - Leave default options unless custom Azure AD app registration is required.
- Click Review + Create, then Create.

Now our cluster has been deployed.

Now we will connect our Azure cluster to terminal as follow:

1. Install Azure CLI and kubectl:

```
$  curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

To Install kubectl

```
$ az aks install-cli
```

2. Get Cluster Credentials:

```
$ az login
$ az aks get-credentials --resource-group <RESOURCE_GROUP> --name <CLUSTER_NAME>
```

This updates `~/.kube/config` with Azure AD authentication.

3. Verify Connection:

```
$kubectl get nodes
```

Access the Kubernetes

To Access the Kubernetes Dashboard, first

1. we will have to install Dashboard
(if not enabled by default)

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```

2. Create a Service Account for the Dashboard:

```
# dashboard-admin.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: dashboard-admin
  namespace: kubernetes-dashboard
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: dashboard-admin
subjects:
  - kind: ServiceAccount
    name: dashboard-admin
    namespace: kubernetes-dashboard
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

To apply it:

```
$ kubectl apply -f dashboard-admin.yaml
```

3. To access the Dashboard:

```
$ kubectl proxy
```

Create roles

To create Roles and Bind to Azure AD Users
1. Define a Role

```
# pod-reader-role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""]
  resources: ["pods"]
```

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

Apply it :

~~~
$  kubectl apply -f pod-reader-role.yaml
  ```
~~~

2. Bind the Role to an Azure AD User:

```
  # pod-reader-binding.yaml
  apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
  metadata:
    name: pod-reader-binding
    namespace: default
  subjects:
  - kind: User
    name: user1@domain.com  # Azure AD UPN
    apiGroup: rbac.authorization.k8s.io
  roleRef:
    kind: Role
    name: pod-reader
    apiGroup: rbac.authorization.k8s.io
```

Apply:

```
$ kubectl apply -f pod-reader-binding.yaml
```

Now we have deployed Azure Kubernetes cluster and created the multiple role for users