ROLE-BASE ACCESS CONTROL FOR AZURE KUBERNETES CLUSTER

Azure RBAC

BY
TIMOTHY FABELURIN

Role-based access control (RBAC)

Access to cloud resource is a critical function that cloud user need to manage properly. RBAC is an authorization system that maintains the less privilege principle for access management to cloud resource. Azure RBAC was implemented in this work, and it helped to manage who has access to resources and the function they can undertake with those resources and the area the individual has access to. Without a doubt, this security measure curbs many attackers' exploit.

The process includes:

Role definition: collection of permissions known as role. Role indicates the actions that can be perform. Examples are read, write, and delete.

Figure 4.1 Role Definition (Robert Lyon, 2022)

Scope: This is the set of resources that access created is applied to, as the role created are limited by the scope.

Role assignment: this refers to the process of attaching defined role to user or group or service principal within a particular scope for access creation.

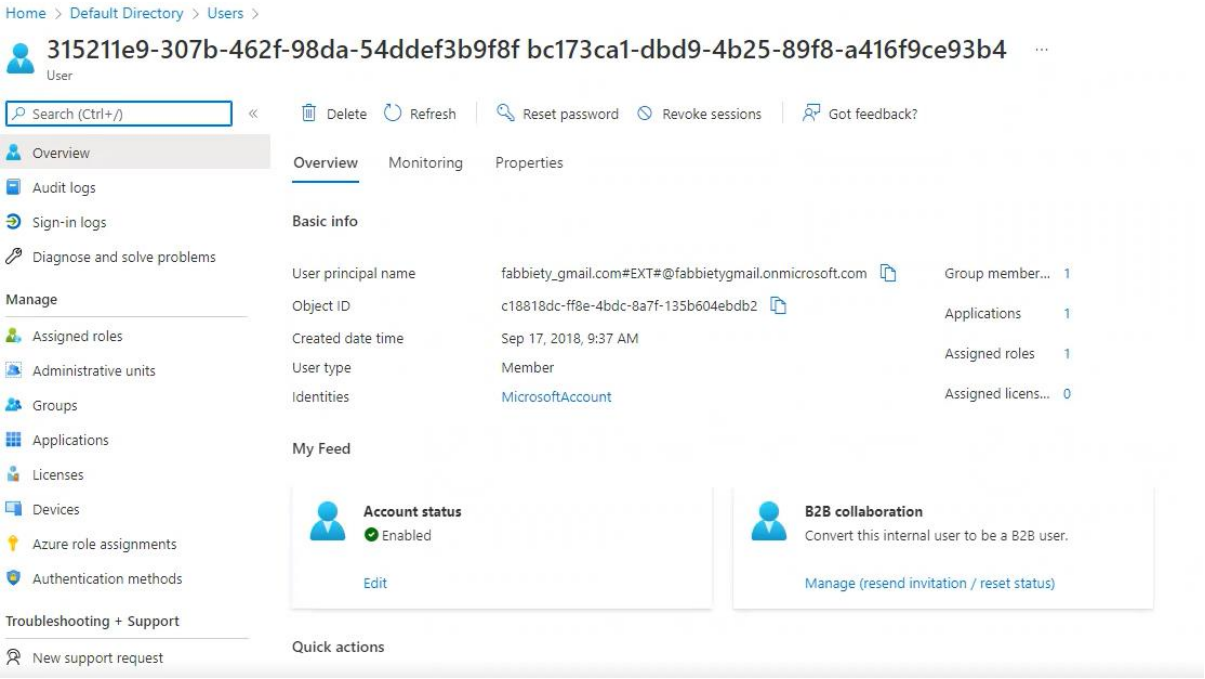
PROCESS

To test this feature, a user on the cloud platform, could successfully log into the portal but could not gain access to the Kubernetes cluster. This user was given the role of "Admin" but because the role assignment was not done, it still could not gain access. This test was simply to indicate that all the implementation stages needed to be carried out before an access can be granted.

To the end, if for by any chance, an attacker gain access into the cloud environment, if the authorization is not explicitly provided to access the cluster, the attacker will not be able to perform any act.


```
PS C:\Users\Fabbiety\Desktop\Project\06-Azure-MySQL-for-AKS-Storage> az role assignment create --role "Azure Kubernetes Service RBAC Admin" --assignee fabbiety_gmail.com#EXT#@fabbietygmail.onmicrosoft.com --scope /subscriptions/191fe5ed-49ee-4667-9fed-a70b6461f460/resourcegroups/terraform-aks-unsecure/providers/Microsoft.ContainerService/managedClusters/terraform-aks-unsecure-cluster
{
  "canDelegate": null,
  "condition": null,
  "conditionVersion": null,
  "description": null,
  "id": "/subscriptions/191fe5ed-49ee-4667-9fed-a70b6461f460/resourcegroups/terraform-aks-unsecure/providers/Microsoft.ContainerService/managedClusters/terraform-aks-unsecure-cluster/providers/Microsoft.Authorization/roleAssignments/ca97ae84-7eb5-4dc5-8ff8-028f7582ef7e",
  "name": "ca97ae84-7eb5-4dc5-8ff8-028f7582ef7e",
  "principalId": "c18818dc-ff8e-4bdc-8a7f-135b604ebdb2",
  "principalType": "User",
  "resourceGroup": "terraform-aks-unsecure",
  "roleDefinitionId": "/subscriptions/191fe5ed-49ee-4667-9fed-a70b6461f460/providers/Microsoft.Authorization/roleDefinitions/3498e952-d568-435e-9b2c-8d77e338d7f7",
  "scope": "/subscriptions/191fe5ed-49ee-4667-9fed-a70b6461f460/resourcegroups/terraform-aks-unsecure/providers/Microsoft.ContainerService/managedClusters/terraform-aks-unsecure-cluster",
  "type": "Microsoft.Authorization/roleAssignments"
}
PS C:\Users\Fabbiety\Desktop\Project\06-Azure-MySQL-for-AKS-Storage>
```

Role was assigned to the user



It can be seen from the capture that the role assignment was successful, and access is now granted to the AKS cluster.