Authentication in Azure Kubernets Service (AKS) explained

Agenda

1. Local account with Kubernetes RBAC
2. Microsoft Entra ID authentication with Kubernetes RBAC
3. Microsoft Entra ID authentication with Azure RBAC

Recap and Conclusion

Local Accounts with Kubernetes RBAC

No link between Microsoft Entra and Kubernetes

Use K8s built-in authentication

- Kubernetes Authenticating

Only recommended when none of the users are in Entra

User management can become very challenging

Local accounts should be disabled for better security

Token is stored unencrypted in .kube config

Entra ID Authentication with Kubernetes RBAC

Authentication via Microsoft Entra

Authorization via Kubernetes RBAC

Admin creates a role binding between K8s role and Entra user or group

Entra user or group needs Azure Kubernetes Service Cluster User role to be able to download the .kube config

Entra ID Authentication with Kubernetes RBAC

Easier user management than local accounts

Choose this option to have a “portable” cluster

- Cluster contains all roles and role bindings definitions

Auditing access to the cluster is not easy

- Access can be given to Entra groups
- Groups and users are managed with their IDs, not name

Entra ID Authentication with Azure RBAC

Manage access to the cluster with Azure only

Use Azure RBAC roles to manage permissions inside the cluster

Recommended way to manage AKS clusters

User or group needs the Azure Kubernetes Service Cluster User role to download the .kube config file

Use the Azure CLI to give namespace specific permission