To Orchestration and Beyond

Your CTO was impressed with your ability to show that AKS can easily support your application using a test deployment. However, you agreed that this deployment would not pass muster with your internal security team or meet audit requirements. Going forward, you'll need to configure a cluster that will ultimately become part of Humongous Insurance's existing cloud infrastructure.

Challenge

Your team's goal in this challenge is to create and configure a Kubernetes cluster on Azure with the appropriate security measures in place. Your company deals with sensitive information, so it is imperative that you address security when configuring your cluster. You need to integrate with your company's Azure Active Directory (AAD) tenant to implement **Role-Based Access Control (RBAC)** for cluster authentication, protect your resources by using a dedicated **VNet** and protect the most critical part of your Kubernetes cluster, the **Kubernetes API Server**.

> Keep in mind these are just the first steps of securing your cluster. You will be asked to further improve your security in later challenges.

Configuration

As you configure your cluster, your CTO would like you to consider the following:

1. Due to the size of Humongous Insurance, many of the private IP address spaces are being used. You were lucky enough to get your networking team to give you an IP range for running applications within Azure. There is an existing VNet in your subscription that represents the IP range that has been allocated for your team.
2. Users of TripInsights expect their data to be accurate and up-to-date at all times. It's important to consider the availability of the application to inform your decision on the number of nodes in your cluster.
3. Only your team members should be able to access the Kubernetes API server.
4. Pods on your cluster should be able to directly communicate with other resources on the VNET via private IP addresses.

Use of RBAC (Role-Based Access Control)

RBAC is used to assign **Roles** (a group of permissions to resources) to **Users** (any entity that accesses a resource interactively) or **Service accounts** (any entity that accesses a resource non-interactively and independent of a User).

Using these constructs allows you to separate permissions between different users and engage in the **Principle of Least Privilege**. This principle suggests that any **User** or **Service account** should be assigned **Role(s)** with the minimum privilege necessary to access the resources that they require to complete their operational role against the cluster and for each application.

Protecting Resources with a VNet

As with many other Azure services, you can protect your Kubernetes nodes by placing them into a VNet. The use of a VNet prevents unauthorized external connections, and can increase the security of corresponding managed services.

Success Criteria

- **Your team** successfully created an RBAC enabled AKS cluster within the address space allocated to you by the network team
- **Your team** must demonstrate that you are prompted on cluster access to authenticate with AAD
- **Your team** must demonstrate connectivity to and from your cluster by being able to reach the `internal-vm` (already deployed)

References

Access and Identity for AKS

- Access and identity options (https://docs.microsoft.com/en-us/azure/aks/concepts-identity)
- Azure AD Integration with AKS (https://docs.microsoft.com/en-us/azure/aks/azure-ad-integration-cli)
- Control Kubeconfig Access (https://docs.microsoft.com/en-us/azure/aks/control-kubeconfig-access)
- Azure CLI: az ad (https://docs.microsoft.com/en-us/cli/azure/ad?view=azure-cli-latest)

Networking for AKS

- Configuring Azure CNI with AKS (https://docs.microsoft.com/en-us/azure/aks/configure-azure-cni)