Managing Kubernetes with Azure Kubernetes Service (AKS), Creating and managing AKS clusters, Scaling and upgrading AKS clusters

1. Creating an AKS Cluster

We can create Azure Kubernetes Service in multiple methods such as Azure Portal and using Azure CLI (terminal)

AKS cluster using the Azure Portal:

1. Sign in to the Azure Portal:
2. First create a resource
3. After creating a resource, we will create a Kubernetes cluster.
4. For this, I'll select my resource group and enter my service namespace
5. Clicking on next tap, creating my node pool and in which choosing Node image and System and Operating system (Linux).

Node pools: Configuring multiple node pools with different VM sizes and configurations.
Autoscaling: Enable cluster autoscaling to automatically scale the number of worker nodes based on resource utilization.
Tags: Addingtags to our cluster for organization and cost tracking.

6. Review + Create:

This will take some time and our AKS will get deployed

2. Managing AKS Clusters

Once a cluster is created, we can manage it using the Azure Portal, Azure CLI, or Azure PowerShell.

Using Azure Portal

Viewing Cluster Status: Checking the status of the cluster (e.g., running, starting, stopped).

Monitoring Cluster Health: Monitoring the health of the nodes, pods, and deployments.

Managing Nodes: Add, remove, or update worker nodes.

Managing Networking: Configure network policies, ingress controllers, and load balancers.

Managing Storage: Create and manage Persistent Volumes and Persistent Volume Claims.

Managing Security: Configure RBAC (Role-Based Access Control), network policies, and pod security policies.

3. Scaling and Upgrading AKS Clusters

Scaling:

Horizontal Pod Autoscaling (HPA): Automatically scales the number of Pods in a deployment based on CPU utilization, memory utilization, or custom metrics.

Cluster Autoscaler: Automatically scales the number of worker nodes in a node pool based on resource requests from Pods. This ensures that there are enough nodes to run the applications.

Manual Scaling: We can manually scale the number of nodes in a node pool using the Azure Portal, Azure CLI, or Azure PowerShell.

Upgrading

Upgrading an AKS cluster involves updating the Kubernetes version.

1. Check for Available Versions: Determine the available Kubernetes versions for our cluster.

2. Plan the Upgrade: Review the upgrade documentation and identify any potential compatibility issues.

3. Upgrade the Control Plane: Upgrade the AKS control plane to the desired Kubernetes version. This is typically done using the Azure Portal, Azure CLI, or Azure PowerShell.

4. Upgrade the Node Pools: Upgrade the worker nodes in each node pool to the desired Kubernetes version. This is typically done in a rolling fashion to minimize downtime.