**Create node pools for a cluster in Azure Kubernetes Service (AKS)**

https://learn.microsoft.com/en-us/azure/aks/create-node-pools

* Nodes have same configuration in a Node Pool
* When you create an AKS cluster, you define the initial number of nodes and Node size (SKU), which creates a system node pool.

**User Node Pools**: These are additional node pools you can create to support workloads with specific requirements, such as compute-intensive applications needing GPUs or workloads requiring high-performance SSD storage. User node pools allow you to customize the resources according to the needs of your applications without affecting the system node pools.

**Flexibility**: You can configure multiple user node pools to meet various workload demands, ensuring optimal performance and resource allocation.

**Limitations**

The following limitations apply when you create AKS clusters that support multiple node pools:

* See [Quotas, virtual machine size restrictions, and region availability in Azure Kubernetes Service (AKS)](https://learn.microsoft.com/en-us/azure/aks/quotas-skus-regions).
* You can delete the system node pool if you have another system node pool in the AKS cluster. Otherwise, you cannot delete the system node pool.
* System pools must contain at least one node, and user node pools may contain zero or more nodes.
* The AKS cluster must use the Standard SKU load balancer to use multiple node pools. This feature isn't supported with Basic SKU load balancers.
* The AKS cluster must use Virtual Machine Scale Sets for the nodes.
* The name of a node pool may only contain lowercase alphanumeric characters and must begin with a lowercase letter.
  + For Linux node pools, the length must be between 1-12 characters.
  + For Windows node pools, the length must be between 1-6 characters.
* All node pools must reside in the same virtual network.
* When you are creating multiple node pools during AKS cluster creation, the Kubernetes version for all node pools must match with the control plane's version. This is a fundamental requirement for ensuring cluster stability and compatibility.

**Create an AKS cluster**

|  |
| --- |
| **Important**  If you run a single system node pool for your AKS cluster in a production environment, we recommend you use at least three nodes for the node pool. If one node goes down, you lose control plane resources and redundancy is compromised. You can mitigate this risk by having more control plane nodes. |

**OS for Node(VMSS):**

Azure Linux

Ubuntu Linux

**Azure Linux node pools**

In AKS the Azure Linux is a lightweight, open-source Linux distribution designed specifically for running containers on Azure Kubernetes Service (AKS). It’s built to be reliable, secure, and consistent, with only the essential packages needed for container workloads. This makes it faster to boot and enhances performance.

**Node pools with unique subnets**

**Azure Subnet:**

If you’re using Azure CNI (Container Network Interface). When you’re adding a new Subnet to your AKS cluster’s VNET, especially if its outside the original CIDR block, you Should update the cluster first. Using **‘az aks update’**

Scenario:

You’ve created an AKS cluster using Azure VNET that’s has an initial CIDR block is ’10.0.0.0/16’. This VNET contains a subnet with CIDR block of ’10.0.1.0/24’ where your AKS nodes are currently running.

Now, you want to add a new subnet with a CIDR block of ’10.0.2.0/24’ to your existing VNET.

So, now you should update the cluster using Azure CLI.

A black and red background with white text

Description automatically generated

**Kubernetes Networking:**

* **No Subnet Management.**
* **No Cluster Updates for Subnets.**

**Summary:**

* **Kubenet (K8s Native Networking)**: Simpler, no need to manage subnets or update the cluster when making changes to the VNET.
* **Azure CNI**: Requires managing VNET subnets and updating the cluster when adding new subnets, especially if they’re outside the original CIDR block.

**Limitations**

* All subnets assigned to node pools must under same virtual network.
* System pods must have access to all nodes and pods in the cluster to provide critical functionality, such as DNS resolution and tunneling kubectl logs/exec/port-forward proxy.
* Windows nodes SNAT traffic to the new subnets until the node pool is reimaged.
* Internal load balancers default to one of the node pool subnets.