

**Lab Manual- Create an Azure Kubernetes Service (AKS) cluster that uses availability zones**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn916

**Contributor:**

Contents

[1. Objective 3](#_Toc159344972)

[1. Limitations and region availability 3](#_Toc159344973)

[2. AKS Cluster with Availability Zone 3](#_Toc159344974)

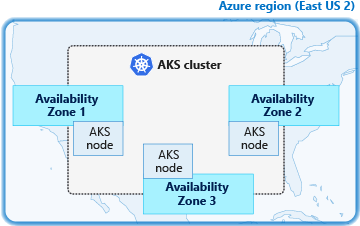
[3. Verify node distribution across zones 5](#_Toc159344975)

[4. Verify pod distribution across zones 5](#_Toc159344976)

# Objective

An Azure Kubernetes Service (AKS) cluster distributes resources such as nodes and storage across logical sections of underlying Azure infrastructure. Using availability zones physically separates nodes from other nodes deployed to different availability zones. AKS clusters deployed with multiple availability zones configured across a cluster provide a higher level of availability to protect against a hardware failure or a planned maintenance event.

By defining node pools in a cluster to span multiple zones, nodes in a given node pool are able to continue operating even if a single zone has gone down. Your applications can continue to be available even if there's a physical failure in a single datacenter if orchestrated to tolerate failure of a subset of nodes.



Kubernetes is aware of Azure availability zones since version 1.12. You can deploy a PersistentVolumeClaim object referencing an Azure Managed Disk in a multi-zone AKS cluster and [Kubernetes takes care of scheduling](https://kubernetes.io/docs/setup/best-practices/multiple-zones/#storage-access-for-zones) any pod that claims this PVC in the correct availability zone.

## Limitations and region availability

AKS clusters can use availability zones in any Azure region that has availability zones.

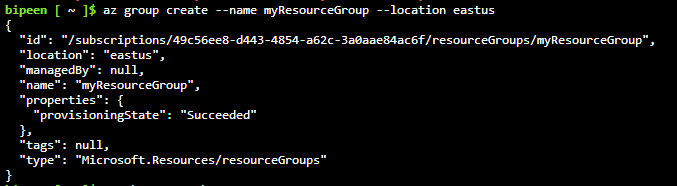
The following limitations apply when you create an AKS cluster using availability zones:

* You can only define availability zones during creation of the cluster or node pool.
* It is not possible to update an existing non-availability zone cluster to use availability zones after creating the cluster.
* The chosen node size (VM SKU) selected must be available across all availability zones selected.
* Clusters with availability zones enabled require using Azure Standard Load Balancers for distribution across zones. You can only define this load balancer type at cluster create time

# AKS Cluster with Availability Zone

creates an AKS cluster named myAKSCluster in the resource group named  myResourceGroup with a total of three nodes. One agent node in zone 1, one in 2, and then one in 3.

az group create --name myResourceGroup --location eastus



az aks create \

--resource-group az800 \

--name bipeenclus \

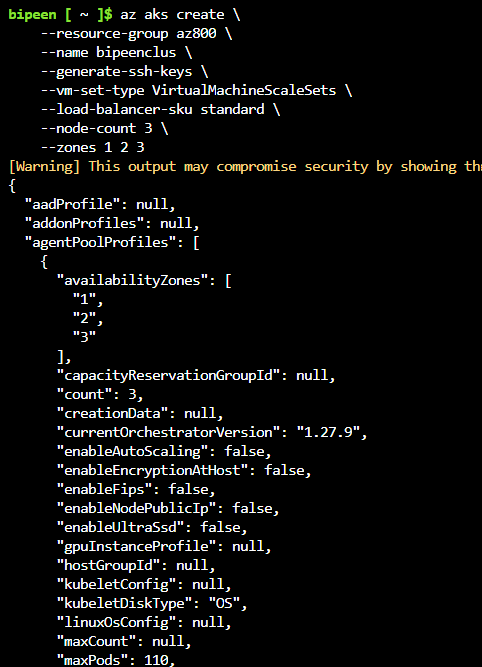
--generate-ssh-keys \

--vm-set-type VirtualMachineScaleSets \

--load-balancer-sku standard \

--node-count 3 \

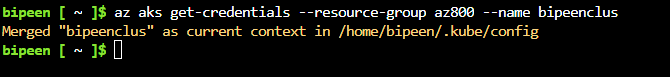
--zones 1 2 3



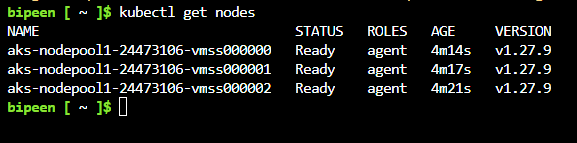
# Verify node distribution across zones

First, get the AKS cluster credentials using the [az aks get-credentials](https://learn.microsoft.com/en-us/cli/azure/aks" \l "az-aks-get-credentials) command:

az aks get-credentials --resource-group az800 --name bipeenclus

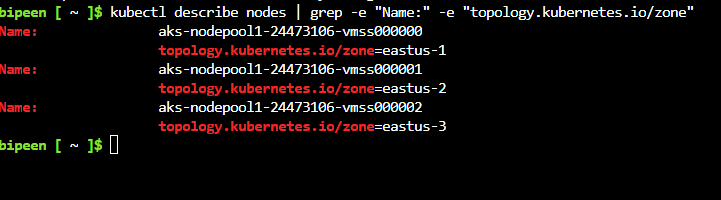


kubectl get nodes

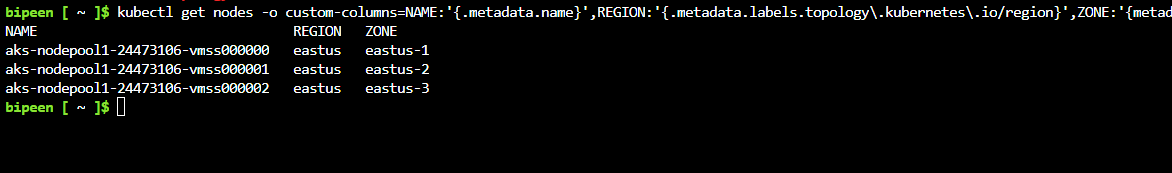


The following example output shows the three nodes distributed across the specified region and availability zones, such as eastus2-1 for the first availability zone and eastus2-2 for the second availability zone:

kubectl describe nodes | grep -e "Name:" -e "topology.kubernetes.io/zone"



kubectl get nodes -o custom-columns=NAME:'{.metadata.name}',REGION:'{.metadata.labels.topology\.kubernetes\.io/region}',ZONE:'{metadata.labels.topology\.kubernetes\.io/zone}'



# Verify pod distribution across zones

Kubernetes uses the **topology.kubernetes.io/zone** label to automatically distribute pods in a replication controller or service across the different zones available. To test the label and scale your cluster from 3 to 5 nodes, run the following command to verify the pod correctly spreads:

az aks scale \

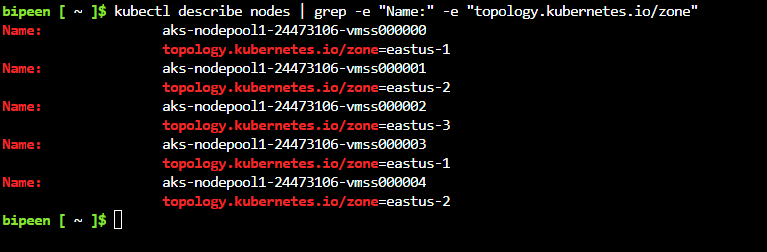
--resource-group az800 \

--name bipeenclus \

--node-count 5

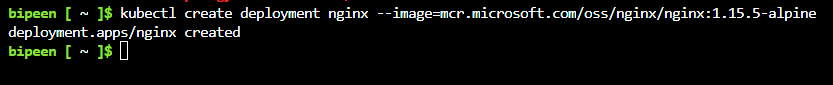


kubectl describe nodes | grep -e "Name:" -e "topology.kubernetes.io/zone"

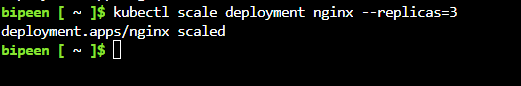


You now have two more nodes in zones 1 and 2. You can deploy an application consisting of three replicas. The following example uses NGINX:

kubectl create deployment nginx --image=mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine

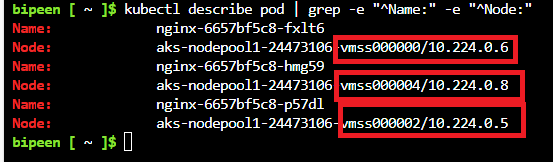


kubectl scale deployment nginx --replicas=3



By viewing nodes where your pods are running, you see pods are running on the nodes corresponding to three different availability zones. For example, with the command kubectl describe pod | grep -e "^Name:" -e "^Node:" in a Bash shell, you see the following example output:

kubectl describe pod | grep -e "^Name:" -e "^Node:"



As you can see from the previous output, the first pod is running on **node 0** located in the availability zone **eastus-1.** The second pod is running on **node 2**, corresponding to **eastus-3,** and the third one in **node 4**, in **eastus-2.** Without any extra configuration, Kubernetes spreads the pods correctly across all three availability zones.