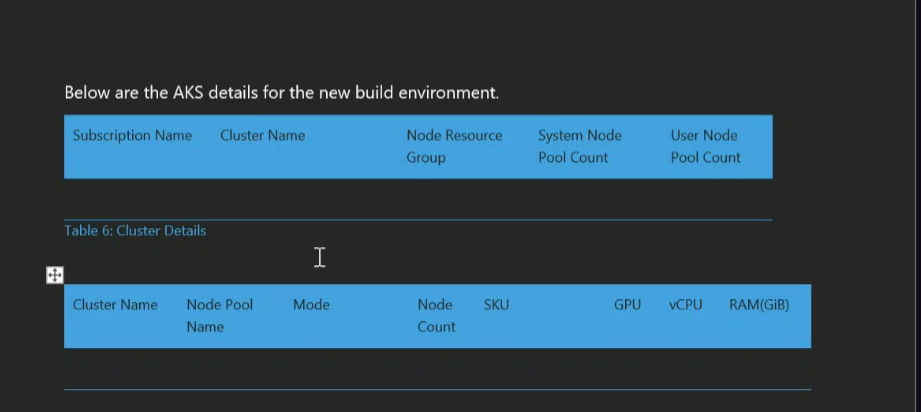
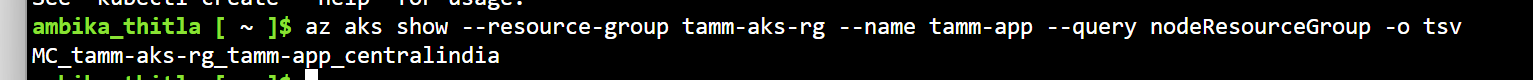
STEPS FOR DISCOVERY



**#1 Node Resource Group** ---resoucegroup name and cluster name keep it handy

Command:

az aks show --resource-group [RESOURCE\_GROUP] --name [CLUSTER\_NAME] --query "nodeResourceGroup" --output tsv



#2 **SystemNode pool count** : to get information about the nodes in your cluster. To get a count of nodes in your cluster, you can use the following command:

az aks nodepool list --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --query "length([])" --output tsv



#3.**user Node pool count :**

az aks nodepool list --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --query "length([])" --output tsv

**#4:Node pool names:**

az aks nodepool list --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --query "[].name" --output tsv

**Output Summary:**

* **Output Format:** A list of node pool names, each on a new line.



**Summary:** Lists the names of all node pools in the specified AKS cluster. Each line represents a different node pool.

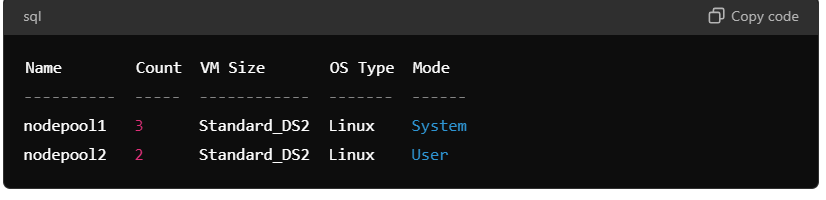


# 5. **Mode**: Indicates whether the node pool is a System pool (used for system components and core Kubernetes functionality) or a User pool (used for application workloads).

**Output Format:** Tabular format that includes details like node pool names, count, VM size, OS type, and mode.

Command:

az aks nodepool list --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --output table



# Node count : list of nodes in a cluster

kubectl get nodes --no-headers | wc -l

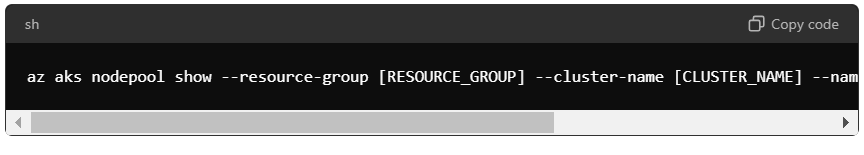




#6.sku**:** This output indicates the SKU of the virtual machines used in the specified node pool. Standard\_DS2\_v2 is a specific VM size SKU in Azure.

Command:

az aks nodepool show --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --name [NODEPOOL\_NAME] --query "vmSize" --output tsv





#7GPU:**:** The VM size used by the node pool, which can be checked against Azure VM sizes with GPU.

az aks nodepool show --resource-group [RESOURCE\_GROUP] --cluster-name [CLUSTER\_NAME] --name [NODEPOOL\_NAME] --query "nodePool.config.vmSize" --output tsv



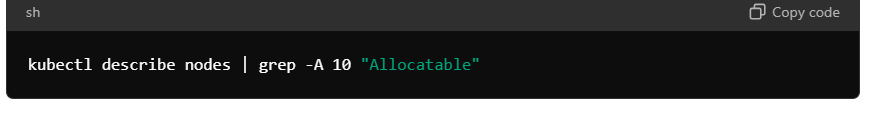
**Summary:** This output indicates the VM size used, such as Standard\_NC6, which is a GPU-enabled VM size in Azure.



#7.vcpu: Detailed output showing allocatable resources on each node, including CPUs.

Command:

kubectl describe nodes | grep -A 10 "Allocatable"





#8.RAM (GIB)

A: to get the resource usage of pods. The output shows memory usage in MiB by default.

Command:

kubectl top pods

B. This command lists the pod names along with their memory usage converted from MiB to GiB.

Command:

kubectl top pods --no-headers | awk '{print $1 ": " $3/1024 " GiB"}'

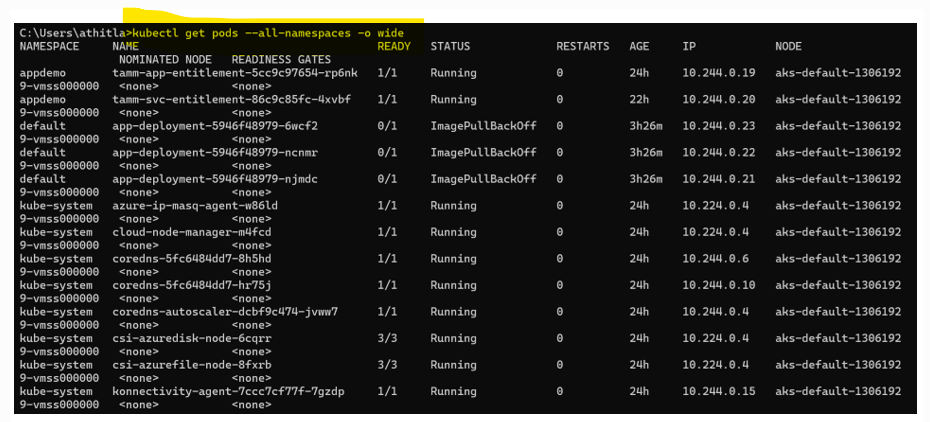
c. To check the memory limits or requests defined for pods, use: This shows the memory limits set for each container in the pods.

Command:

kubectl get pods -o=jsonpath='{range .items[\*]}{.metadata.name}: {.spec.containers[\*].resources.limits.memory}{"\n"}{end}'

Steps to convert the output of Kubernetes into .CSV file

#1. To Get Pod Details

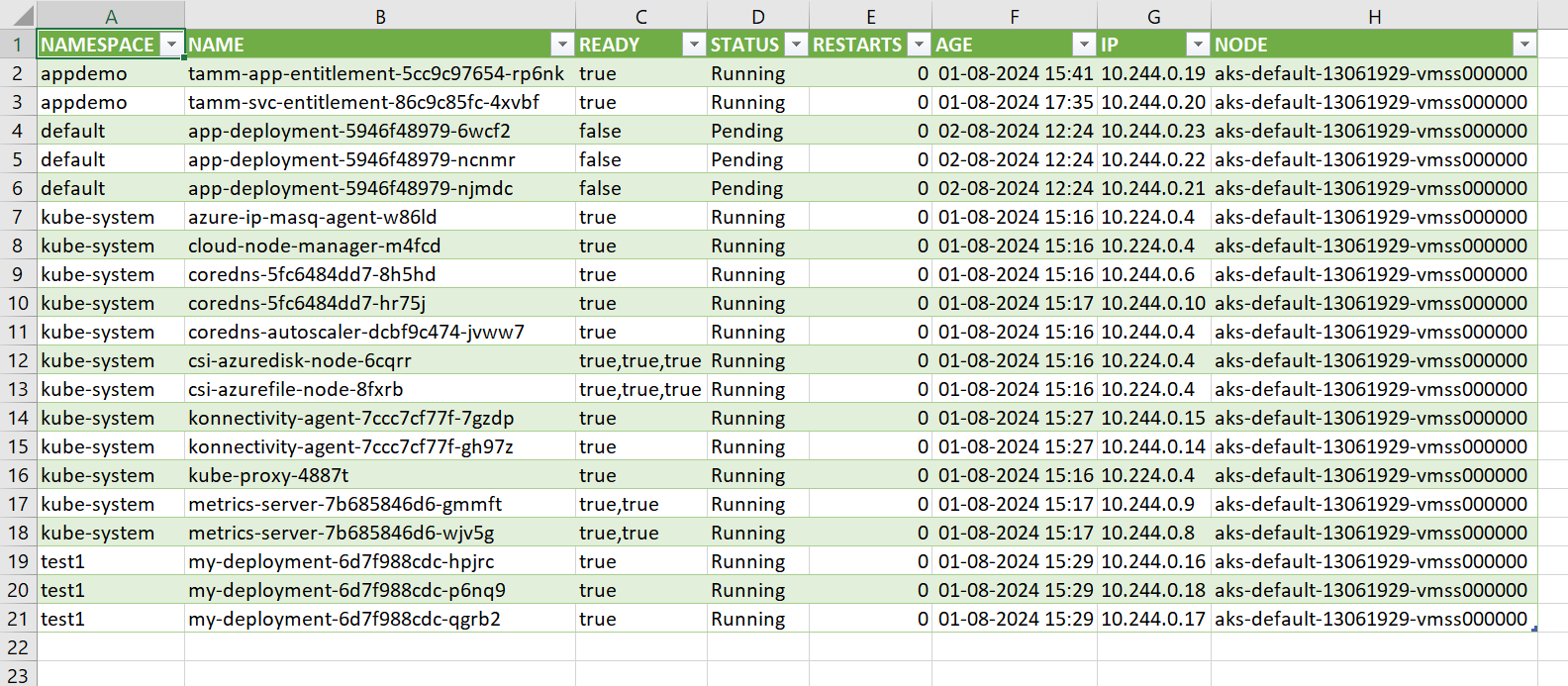
$ kubectl get pods --all-namespaces 

# to download as a output into .txt files

$kubectl get pods --all-namespaces > pods\_data2.txt

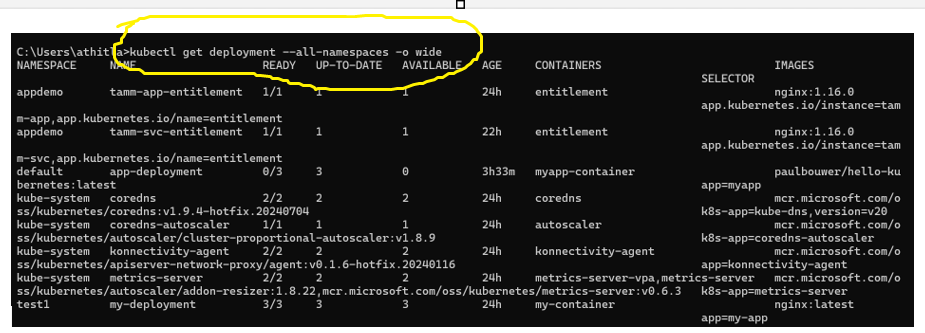
# to convert the output into .csv files

$kubectl get pods --all-namespaces -o=custom-columns=NAMESPACE:.metadata.namespace,NAME:.metadata.name,READY:.status.containerStatuses[\*].ready,STATUS:.status.phase,RESTARTS:.status.containerStatuses[\*].restartCount,AGE:.metadata.creationTimestamp,IP:.status.podIP,NODE:.spec.nodeName > /Users/athitla/test/pods\_data3.csv



#2. To get Deployment details.

$kubectl get deployment --all-namespaces -o wide



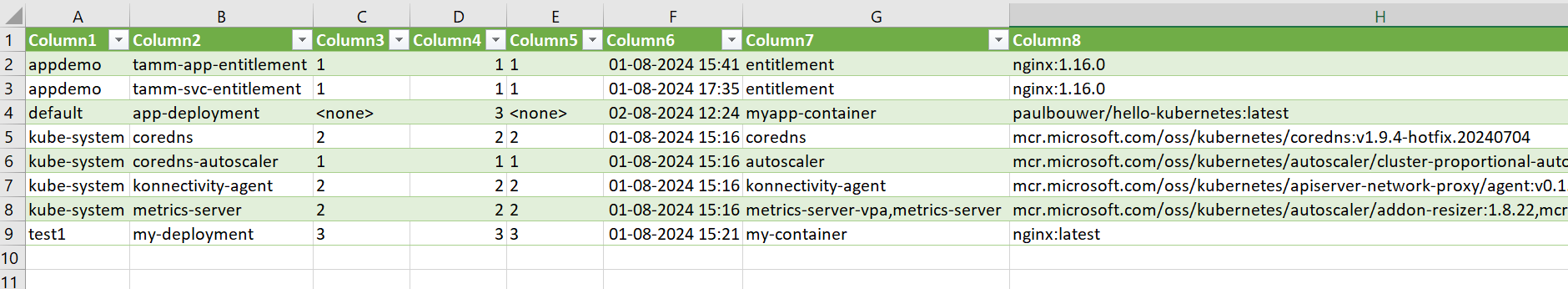
# to covert the output into .txt files

$kubectl get deployments --all-namespaces > pods\_data2.txt

# to convert the output into .csv files

$kubectl get deployments --all-namespaces -o=custom-columns=NAMESPACE:.metadata.namespace,NAME:.metadata.name,READY:.status.readyReplicas,UP-TO-DATE:.status.updatedReplicas,AVAILABLE:.status.availableReplicas,AGE:.metadata.creationTimestamp,CONTAINERS:.spec.template.spec.containers[\*].name,IMAGES:.spec.template.spec.containers[\*].image > /Users/athitla/test/pods\_data56.csv

To get deployment



import subprocess

import csv

import os

# Define the output CSV file path

output\_csv = "kubectl\_get\_all\_output.csv"

# Run the 'kubectl get all' command

kubectl\_command = "kubectl get all -o wide"

process = subprocess.Popen(kubectl\_command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

stdout, stderr = process.communicate()

if process.returncode != 0:

    print(f"Error running kubectl command: {stderr.decode()}")

else:

    # Decode the output and split by lines

    lines = stdout.decode().splitlines()

    # Open the CSV file for writing

    with open(output\_csv, mode='w', newline='') as file:

        writer = csv.writer(file)

        # Write each line to the CSV file

        for line in lines:

            writer.writerow(line.split())

    print(f"Output saved to {output\_csv}")