## Before you start: Manual CIS Checks

PLEASE NOTE: CIS checks require you to have Security Reader on the azure subscription and

Azure Kubernetes Service RBAC Admin , Azure Kubernetes Service RBAC Reader on the AKS clusters in scope due to privileged checks.

Getting Started:

1. In the Portal, open your AKS resource.
2. Click Connect → Open Cloud Shell.
3. Automatically runs the following commands
   1. Set the cluster subscription
   2. Download cluster credentials ( you will need this to authenticate to the cluster)
4. Run: to List your nodes: **kubectl get nodes -o wide**
   1. Your output will look something like this:

NAME STATUS VERSION OS-IMAGE

aks-nodepool1-xxxx-vmss000000 Ready v1.32.6 Ubuntu 22.04.5 LTS

1. Group the pods by node to understand how many pods there are per node and thus how many times you have to run your CIS checks. You only need to run the checks **once on that node**, and it covers all pods that are within it If 30 pods run on one node, a single set of checks covers all 30:

**kubectl get pods -o wide --all-namespaces | awk '{print $8}' | sort | uniq -c**

To Start running your CIS checks:

CIS checks are **node-level**. This means you run the checks on each **node** (VM) in the cluster, not on every pod. Example: If you have 40 pods spread across 3 nodes, you only need to check **3 nodes** (one debugger per node). the node name you need to enter to create the debugger is the name of the node you got by running **kubectl get nodes -o wide.** To start your CIS checks run the below:

1. Start a debug pod running on the node:

**kubectl debug node/<NODE\_NAME> -it --image=ubuntu --profile=sysadmin -- bash**

1. Enter the host filesystem:

**chroot /host**

1. Run your checks as per the spreadsheet ( as per the audit instructions)
2. To exit the debug node run: exit ( and press enter)

**Clean up:**

That pod existed only while you were inside the session. As soon as you typed exit, kubectl cleaned it up automatically (because that’s the default for interactive debug) unless you ran it with the attach=false flag in which case you would have to delete it.

For example, if I ran:

kubectl debug node/<NODE\_NAME> --image=ubuntu --profile=sysadmin --attach=false

To **list all debug pods** (across all namespaces):

kubectl get pods -A -l debugger=true -o wide

To **delete all debug pods**:

kubectl delete pod -A -l debugger=true

Useful information:

For your information: In AKS, Azure manages the control plane, so you can’t access it. The parts you do control are the worker Nodes (the VMs that run your Pods). Because the kubelet and OS-level settings live on the Nodes, CIS checks that involve system files or kubelet config must be run on the **Nodes**, not directly on the Pods.