**Connect the AKS cluster from Same VNet**

**CONTAINERD**

**containerd** is AKS default container runtime, replacing docker which was used in the past.

Kubectl get nodes -o wide ( to find container runtime version)

A black background with white text

AI-generated content may be incorrect.

The crictl ps command is used to list **running containers** in a Kubernetes cluster when you're using a container runtime interface (CRI)-compatible runtime (like containerd or CRI-O).

<https://kubernetes.io/docs/tasks/debug/debug-cluster/crictl/>

crictl pods

crictl ps

crictl logs <id>

This will show a list of currently **running** containers.

**Connect to AKS Cluster nodes for maintenance or troubleshooting**

Throughout the lifecycle of your Azure Kubernetes Service (AKS) cluster, you eventually need to directly access an AKS node. This access could be for maintenance, log collection, or troubleshooting operations

A screenshot of a computer

AI-generated content may be incorrect.

By default , external IP is not created for worker nodes.

**Access nodes using the Kubernetes API**

This method requires usage of kubectl debug command.

This guide shows you how to create a connection to an AKS node and update the SSH key of your AKS cluster.

**kubectl get nodes -o wide**

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

aks-agentpool-47159840-vmss000000 Ready <none> 30h v1.31.8 10.224.0.6 <none> Ubuntu 22.04.5 LTS 5.15.0-1090-azure containerd://1.7.27-1

aks-workerpool-47159840-vmss000000 Ready <none> 30h v1.31.8 10.224.0.5 <none> Ubuntu 22.04.5 LTS 5.15.0-1090-azure containerd://1.7.27-1

aks-workerpool-47159840-vmss000001 Ready <none> 30h v1.31.8 10.224.0.4 <none> Ubuntu 22.04.5 LTS 5.15.0-1090-azure containerd://1.7.27-1

kubectl debug node/aks-workerpool-47159840-vmss000000 -it --image=mcr.microsoft.com/cbl-mariner/busybox:2.0

Creating debugging pod node-debugger-aks-workerpool-47159840-vmss000000-wrn89 with container debugger on node aks-workerpool-47159840-vmss000000.

If you don't see a command prompt, try pressing enter.

chroot/host

**You can interact with the node session by running chroot /host from the privileged container.**

When you're done with your node, enter the exit command to end the interactive shell session. After the interactive container session closes, delete the debugging pod used with kubectl delete pod.

**kubectl delete pod node-debugger-aks-workerpool-47159840-vmss000000-wrn89**

kubectl delete pod node-debugger-aks-workerpool-47159840-vmss000000-wrn89

pod "node-debugger-aks-workerpool-47159840-vmss000000-wrn89" deleted

**Use 3rd Party Tool**

<https://github.com/kvaps/kubectl-node-shell>