Connect to Azure Kubernetes Service (AKS) cluster nodes for maintenance or troubleshooting

Yes, you're correct! To troubleshoot or debug issues with a specific pod on a node in Azure Kubernetes Service (AKS), you would use the methods described above, depending on your specific needs:

In Azure Kubernetes Service (AKS), there are some scenarios you might need to directly access the Nodes and Pods in your cluster, such as inspection, logs or troubleshooting. Since AKS nodes are not exposed to the internet for security reasons.

* There are two methods to access them securely to connect to AKS Nodes and Pods.

1. Using **‘kubectl debug’** (Kubernetes API Access)
2. Using AKS ARM API (Access Node via Private IP)

# Using ‘kubectl debug’ (Kubernetes API Access)

The ‘kubectk debug’ command is a Kubernetes-native method that allows you to run a troubleshooting container directly on the node.

Steps:

* Prerequisites: You need to access the Kubernetes API, you must install **‘kubectl’**.

TO start debug container on a node:

|  |
| --- |
| **kubectl debug node/<node-name> -it --image=<image\_name:tag/>** |

* This command created a pod running on a specified node OS.

Advantages:

* No need direct network access to the Nodes private IP.

# Connect using kubectl debug

1. To list your nodes, use the ‘kubectl get nodes’ command:

**Kubectl get nodes -o wide**

Sample output:

|  |
| --- |
| NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE  aks-nodepool1-37663765-vmss000000 Ready agent 166m v1.25.6 10.224.0.33 <none> Ubuntu 22.04.2 LTS  aks-nodepool1-37663765-vmss000001 Ready agent 166m v1.25.6 10.224.0.4 <none> Ubuntu 22.04.2 LTS  aksnpwin000000 Ready agent 160m v1.25.6 10.224.0.62 <none> Windows Server 2022 |

1. Use the kubectl debug command to start a privileged container on your node and connect to it.

|  |
| --- |
| **kubectl debug node/aks-nodepool1-37663765-vmss000000 -it --image=mcr.microsoft.com/cbl-mariner/busybox:2.0** |

Sample output:

|  |
| --- |
| Creating debugging pod node-debugger-aks-nodepool1-37663765-vmss000000-bkmmx with container debugger on node aks-nodepool1-37663765-vmss000000.  If you don't see a command prompt, try pressing enter.  root@aks-nodepool1-37663765-vmss000000:/# |

You now have access to the node through a privileged container as a debugging pod.

# Exit kubectl debug mode

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-nodepool1-37663765-vmss000000-bkmmx** |