wsecure access to AKS Nodes via different methods

Securing access to AKS Nodes is essential for secure and controlled Nodes. By default, AKS won’t expose any nodes to public internet. However, still you can set it more securely using SSH methods.

Below are a few best practices for managing SSH access to AKS nodes.

# 1. Using Azure Bastion for Secure SSH Access

Azure Bastion provides security and SSH, RDP access to VM directly in the Azure Portal over SSL. It acts as a jump server without needing the Public IP.

Steps:

1. **Deploy Azure Bastion**:
   * Go to the Azure portal and navigate to your virtual network (VNet) that contains your AKS nodes.
   * Under "Operations," select "Bastion" and click "Create."
   * Follow the prompts to deploy Azure Bastion in the selected VNet.
2. **Connect to AKS Nodes via Bastion**:
   * In the Azure portal, navigate to the Virtual Machine Scale Set (VMSS) that backs your AKS nodes.
   * Select the specific instance (node) you want to access.
   * Click "Connect," then choose "Bastion" and use the SSH option to securely access the node.

**Benefits:**

* **No Public IP Required**: Nodes remain isolated from the public internet.
* **Secure Connection**: Connections are encrypted using SSL, ensuring secure communication.

# 2. Using Private IP with VPN or ExpressRoute

# Create an AKS cluster with SSH key

* + 1. **Configure SSH Access During Cluster Creation**

When creating an AKS cluster, you can configure SSH access to your nodes by specifying an SSH public key. This method is useful if you anticipate needing SSH access to nodes regularly.

Steps:

1. Generate an SSH Key Pair:

* On your local machine, generate an SSH key pair if you don't already have one:

|  |
| --- |
| Bash Copy code  **ssh-keygen -t rsa -b 4096 -C "your\_email@example.com** |

1. Specify the SSH Key During AKS Cluster Creation:

* When creating the AKS cluster via the Azure CLI, provide the SSH public key:

|  |
| --- |
| Bash Copy code  **az aks create \**  **--resource-group <resource-group-name> \**  **--name <aks-cluster-name> \**  **--ssh-key-value ~/.ssh/id\_rsa.pub** |

1. Access the Nodes:

* After the cluster is created, you can SSH into any node using:

|  |
| --- |
| Bash Copy code  **ssh azureuser@<node-private-ip>** |

Benefits:

* **Controlled Access**: SSH access is limited to those with the specified SSH key.
* **Flexibility**: SSH access is baked into the cluster from the start.

# 4. Disable SSH Root Access and Enforce Key-Based Authentication

To further secure SSH access, you should disable root login and ensure that only key-based authentication is allowed.

Steps:

1. Disable Root Login:

* SSH into the AKS node and edit the SSH configuration file:

|  |
| --- |
| Bash Copy code  **sudo nano /etc/ssh/sshd\_config** |

* Set PermitRootLogin to **‘no’**:

|  |
| --- |
| Bash Copy code  **PermitRootLogin no** |

1. Enforce Key-Based Authentication:

* Ensure PasswordAuthentication is set to no:

|  |
| --- |
| Bash Copy code  **PasswordAuthentication no** |

* Restart the SSH service to apply changes:

|  |
| --- |
| Bash Copy code  **Sudo systemctl restart sshd** |

Benefits:

* **Enhanced Security**: Prevents unauthorized access through brute-force password attacks.
* **Best Practices**: Aligns with security best practices for SSH.