

AKS Security Simplified for Developers

Wolfgang Ofner
Senior Cloud Architect

Agenda

Authentication and
Authorization

Entra Workload ID

Private AKS Cluster

Azure Key Vault Provider for
Secrets

Wolfgang Ofner

Freelance Cloud Architect, Toronto, Canada

Focus on Azure, Kubernetes, and DevOps



programmingwithwolfgang.com



[wolfgangofner](#)



[@programmingwithwolfgang](#)



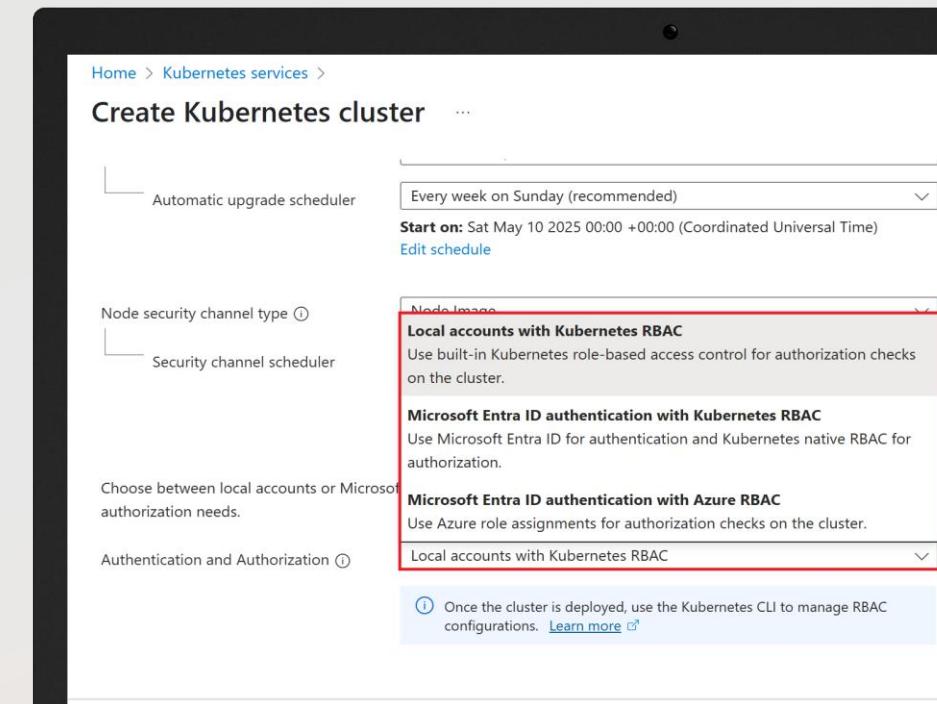
Authentication and Authorization

Authentication

Local Accounts with
Kubernetes RBAC

Entra ID Authentication
with Kubernetes RBAC

Entra ID Authentication
with Azure RBAC



Create Kubernetes cluster



Automatic upgrade scheduler

Every week on Sunday (recommended)



Start on: Sat May 10 2025 00:00 +00:00 (Coordinated Universal Time)

[Edit schedule](#)

Node security channel type ⓘ

Security channel scheduler

Choose between local accounts or Microsoft Entra ID for authentication and authorization needs.

Authentication and Authorization ⓘ

Node Image



Local accounts with Kubernetes RBAC

Use built-in Kubernetes role-based access control for authorization checks on the cluster.

Microsoft Entra ID authentication with Kubernetes RBAC

Use Microsoft Entra ID for authentication and Kubernetes native RBAC for authorization.

Microsoft Entra ID authentication with Azure RBAC

Use Azure role assignments for authorization checks on the cluster.

Local accounts with Kubernetes RBAC



Once the cluster is deployed, use the Kubernetes CLI to manage RBAC configurations. [Learn more](#) ↗

[Previous](#)

[Next](#)

[Review + create](#)

Give feedback

Local Accounts with Kubernetes RBAC

Local Accounts K8s RBAC

Default authentication mode for AKS

No link between Microsoft Entra and AKS

Use K8s build-in authentication

Token is stored unencrypted in .kube config file

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUU2RENDQXRDZ0I
    server: https://azurecloudnative-aks-dns-537mjg6w.hcp.canadacentral.azurek8s.io:443
  name: AzureCloudNative-aks
contexts:
- context:
  cluster: AzureCloudNative-aks
  user: clusterUser_AzureCloudNative-rg_AzureCloudNative-aks
  name: AzureCloudNative-aks
current-context: AzureCloudNative-aks
kind: Config
preferences: {}
users:
- name: clusterUser_AzureCloudNative-rg_AzureCloudNative-aks
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZIakNDQXdhZ0F3SU
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkJURSBLRVktLS0tLQpNSU1KSndJQkFBS0NBZ0VBN
```

Local Accounts K8s RBAC

Only recommended when none of the users are in Entra

User management can become very challenging

Local accounts should be disabled for better security

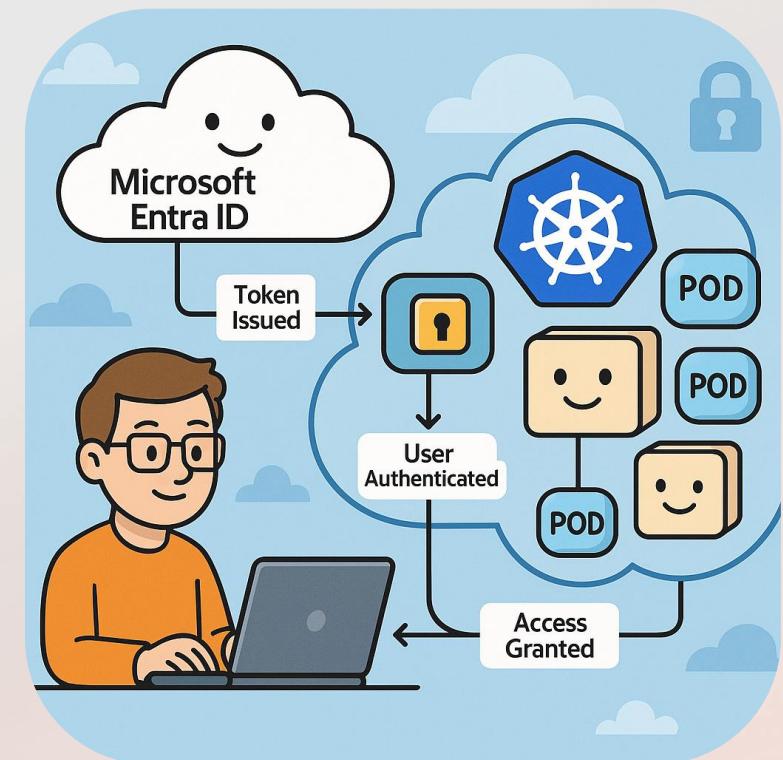
```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJ
  server: https://azurecloudnative-aks-
  name: AzureCloudNative-aks
contexts:
- context:
  cluster: AzureCloudNative-aks
  user: clusterUser_AzureCloudNative-rg
  name: AzureCloudNative-aks
current-context: AzureCloudNative-aks
kind: Config
preferences: {}
users:
- name: clusterUser_AzureCloudNative-rg_A
  user:
    client-certificate-data: LS0tLS1CRUdJ
    client-key-data: LS0tLS1CRUdJTiBSU0Eg
```

Entra ID Authentication with Kubernetes RBAC

Entra ID with K8s RBAC

Authentication via Microsoft Entra

Authorization via K8s RBAC



Entra ID with K8s RBAC

Admin creates role bindings between K8s role and Entra user or group

Entra user or group needs “Azure Kubernetes Cluster User Role” role to download credentials

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader
  namespace: read
rules:
- apiGroups: [""]
  resources: ["pods", "services", "endpoints", "persistentvolumeclaims", "persistentvolumeclaims/status"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources: ["deployments", "daemonsets", "replicasets", "statefulsets"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["batch"]
  resources: ["jobs", "cronjobs"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
  resources: ["ingresses"]
```

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader-user-binding
  namespace: read
subjects:
- kind: Group
  name: 24975d09-19e9-47a5-aa3b-e952c693c016 # Entra ID
  namespace: read
roleRef:
  kind: Role # or ClusterRole
  name: reader
  apiGroup: rbac.authorization.k8s.io
```

Entra ID with K8s RBAC

```
PS C:\Demo> kubectl get all -n read
NAME        READY   STATUS    RESTARTS   AGE
pod/nginx   1/1     Running   0          4m32s
```

```
Error from server (Forbidden): replicationcontrollers is forbidden
: User "demo.user@programmingwithwolfgang.com" cannot list resource
"replicationcontrollers" in API group "" in the namespace "read"
```

```
Error from server (Forbidden): horizontalpodautoscalers.autoscaling is forbidden: User "demo.user@programmingwithwolfgang.com" cannot list resource "horizontalpodautoscalers" in API group "autoscaling" in the namespace "read"
```

Entra ID with K8s RBAC

Easier user management
than local accounts

Use for Arc-enabled K8s
clusters

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader
  namespace: read
rules:
- apiGroups: [""]
  resources: ["pods", "services", "endpoints", "persistentvolumeclaims"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources: ["deployments", "daemonsets", "replicasets", "statefulsets"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["batch"]
  resources: ["jobs", "cronjobs"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
  resources: ["ingresses"]
```



Entra ID with K8s RBAC

Auditing access to the cluster can be cumbersome

Management with Entra IDs

Access can be given to Entra users and groups

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader-user-binding
  namespace: read
subjects:
- kind: Group
  name: 24975d09-19e9-47a5-aa3b-e952c693c016 # Entra ID Group
  namespace: read
roleRef:
  kind: Role # or ClusterRole
  name: reader
  apiGroup: rbac.authorization.k8s.io
```

Entra ID Authentication with Azure RBAC

Entra Authentication with Azure RBAC

Manage access to the cluster
with Azure only

Use Azure RBAC roles to
manage permissions inside AKS

Recommended way to manage
AKS cluster

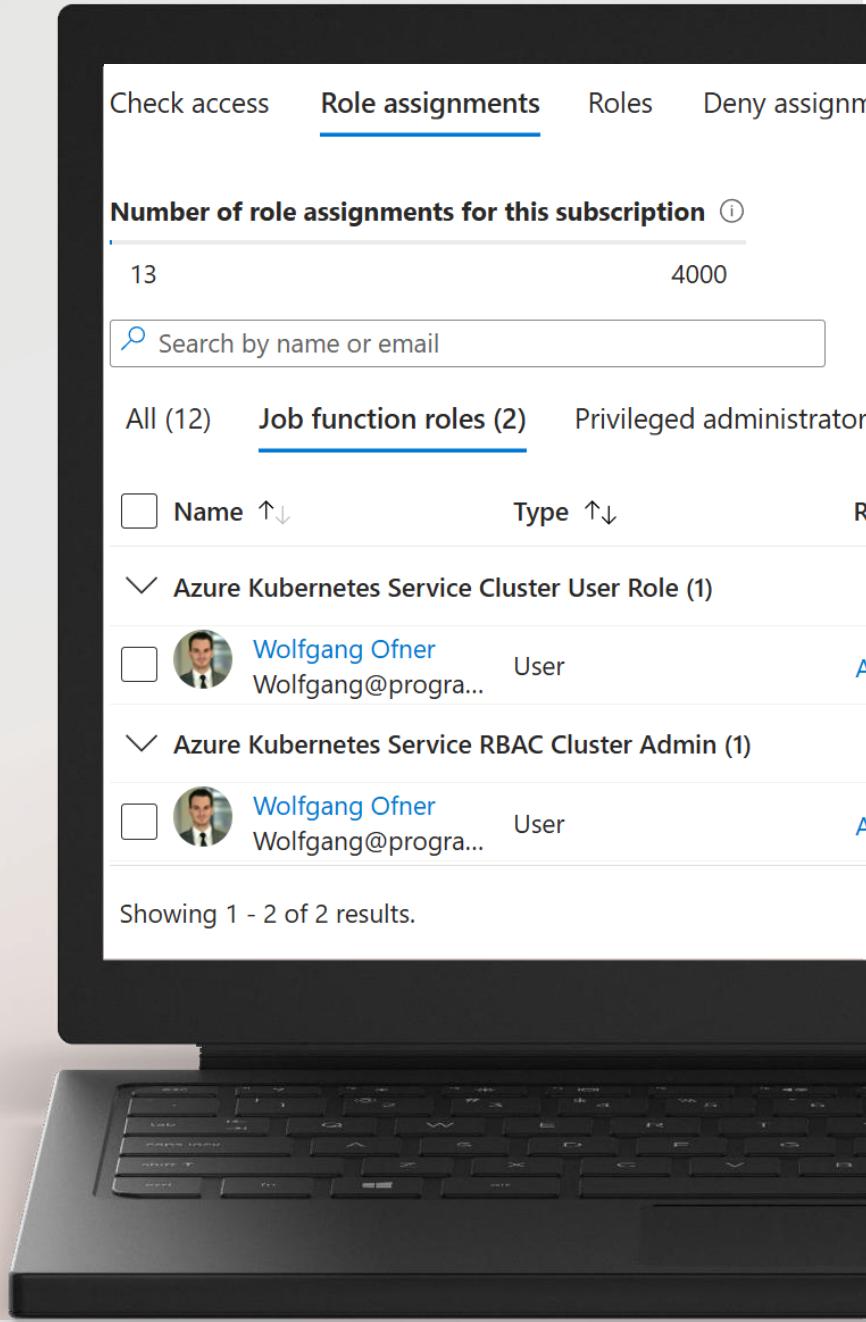


Entra Authentication with Azure RBAC

“Azure Kubernetes Service Cluster User Role” to download credentials

Assign built-in or custom roles

Namespace specific permissions via Azure CLI only



Entra
Workload Identity

Entra Workload Identity

Azure resource can have identities

Always use identities over username
and password

AKS identity is not assigned to
the pods



Entra Workload Identity

Entra Workload Identity gives a pod an identity

- Pods can access Azure resources with this identity
- OIDC Issuer must be enabled for the AKS cluster
- Label pod and annotate service account



Kubelet



AKS workload



Microsoft Entra ID



OpenID Discovery Document



Azure resources

Projects service account token
to the workload at a
configurable file path

Sends projected, signed service
account token and requests
Microsoft Entra access token

Checks trust on the app and
validates using incoming token

Issues Microsoft Entra
access token

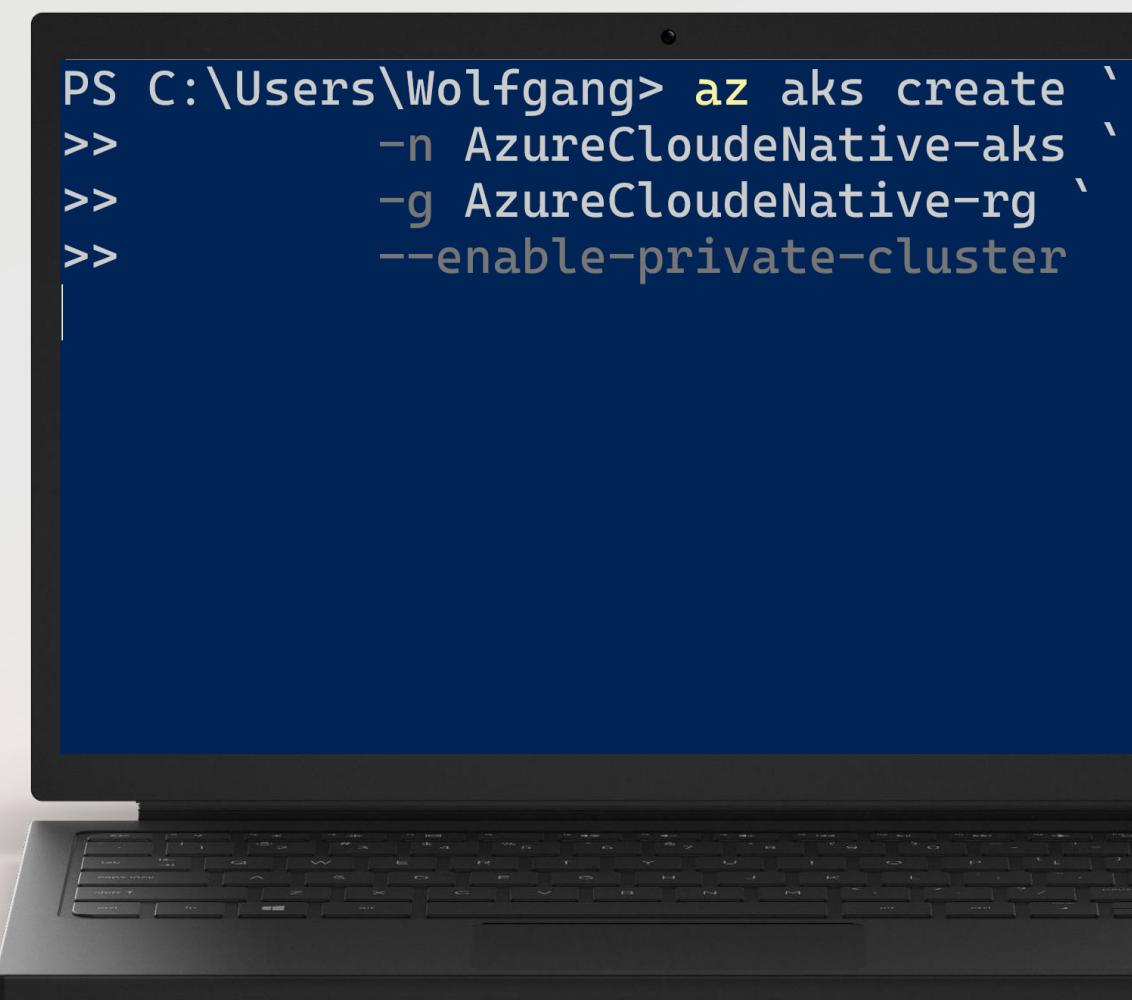
Access resources using
Microsoft Entra access token

Private AKS Cluster

Private AKS Cluster

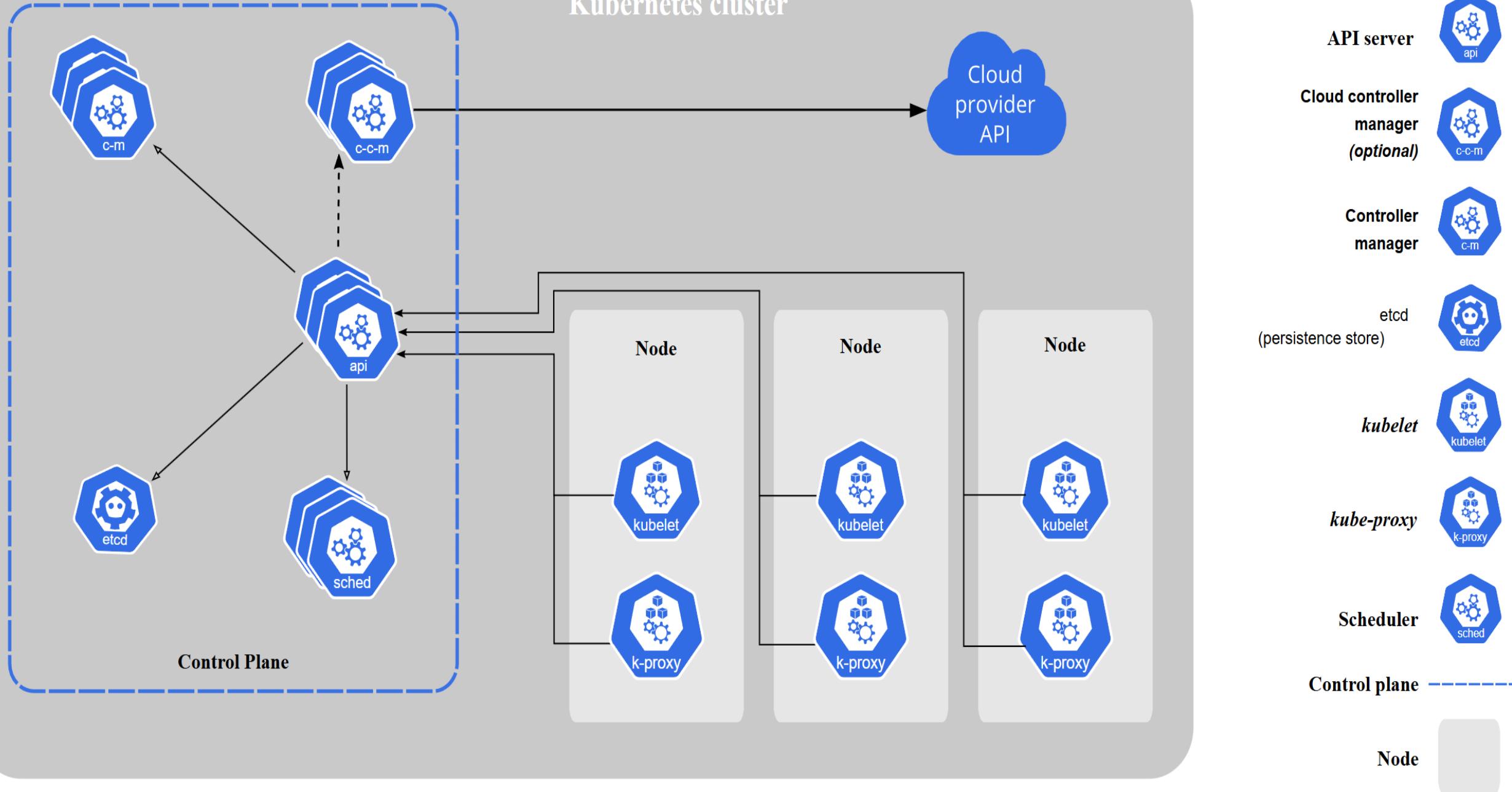
Disable public access to API

Communication between
Worker and Master nodes
over a private connection

A photograph of a dark-colored laptop computer. The screen displays a black terminal window with white text. The text shows a PowerShell command being typed, starting with 'PS C:\Users\Wolfgang> az aks create <'. The command includes several parameters: '-n AzureCloudNative-aks', '-g AzureCloudNative-rg', and '--enable-private-cluster'. The keyboard of the laptop is visible at the bottom of the frame.

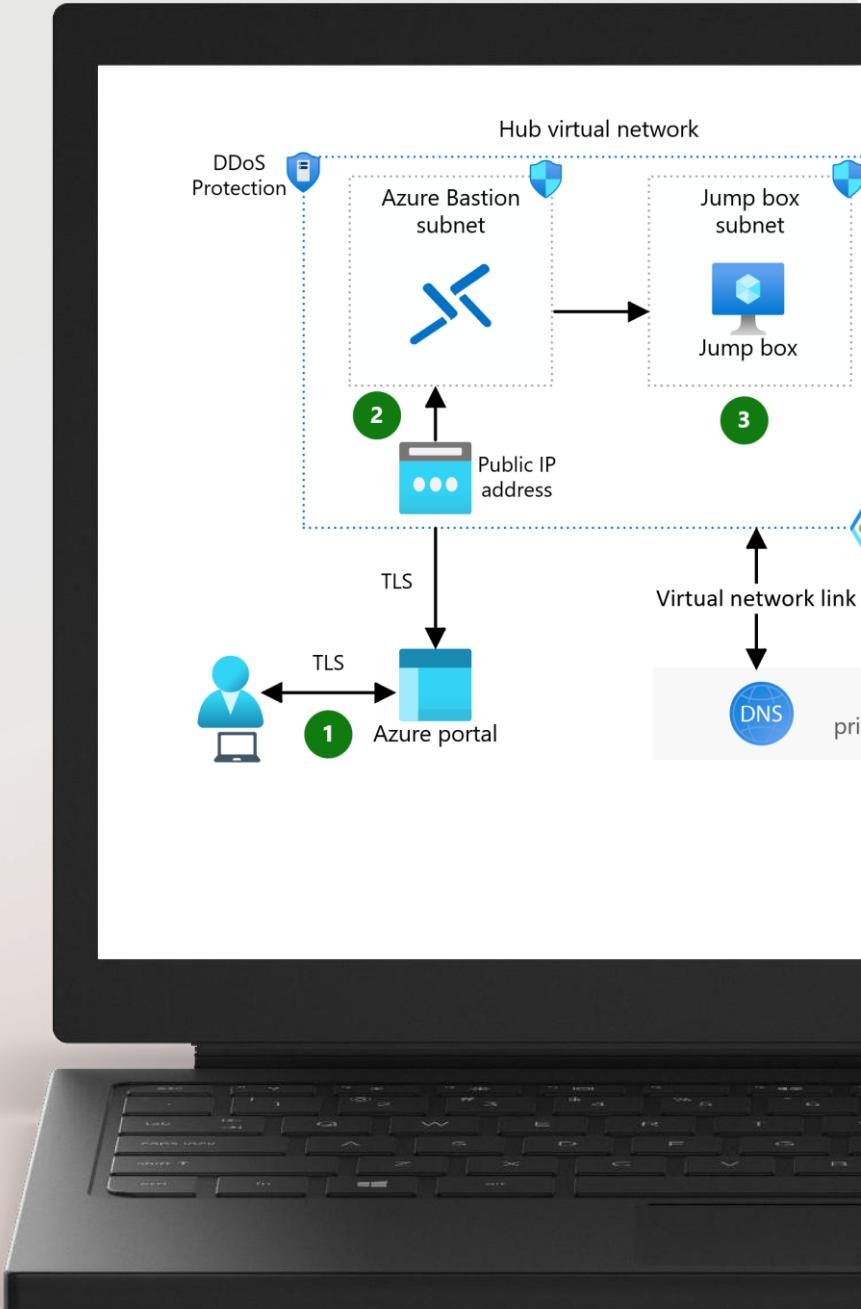
```
PS C:\Users\Wolfgang> az aks create `<
>> -n AzureCloudNative-aks `<
>> -g AzureCloudNative-rg `<
>> --enable-private-cluster`<
```

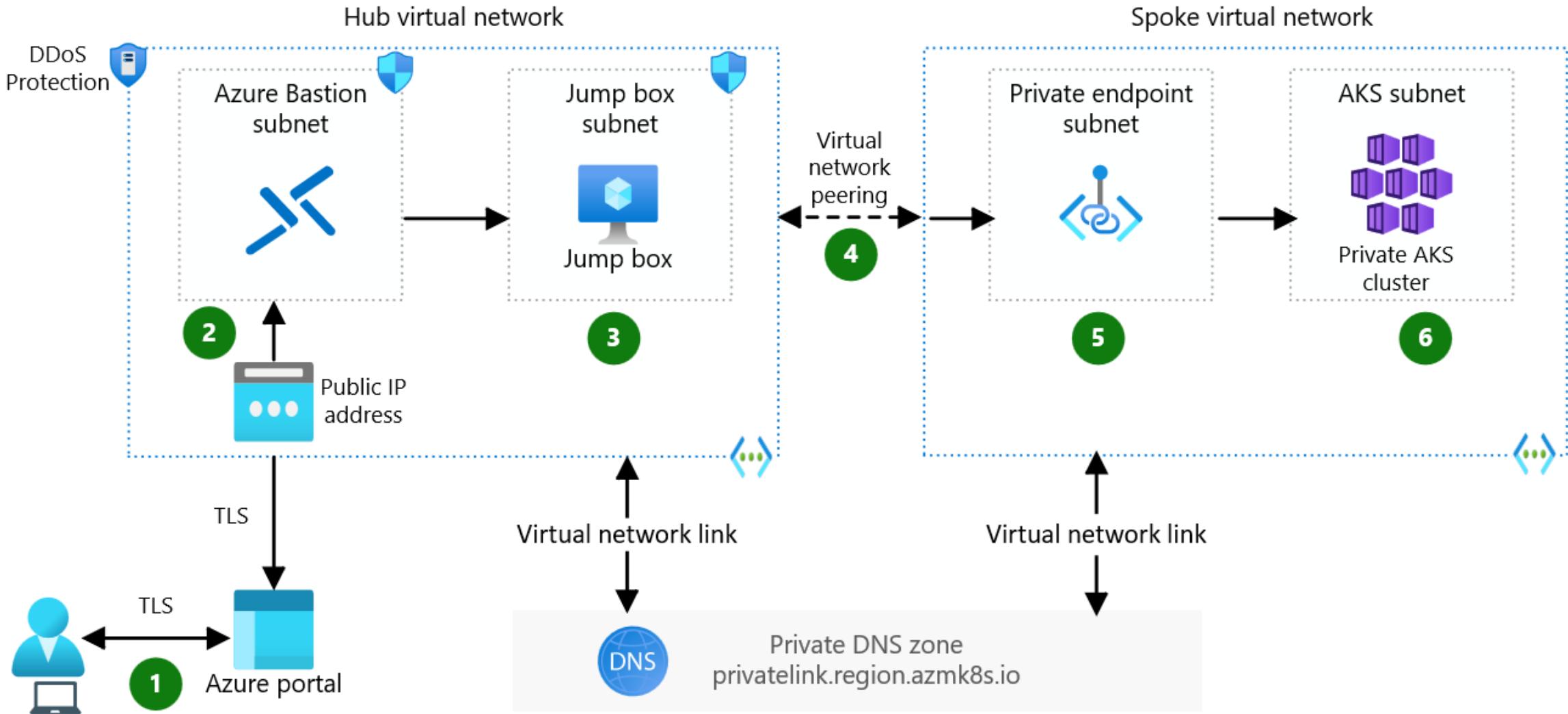
Kubernetes cluster



Private AKS Cluster

Use Azure Bastion and a Jump box

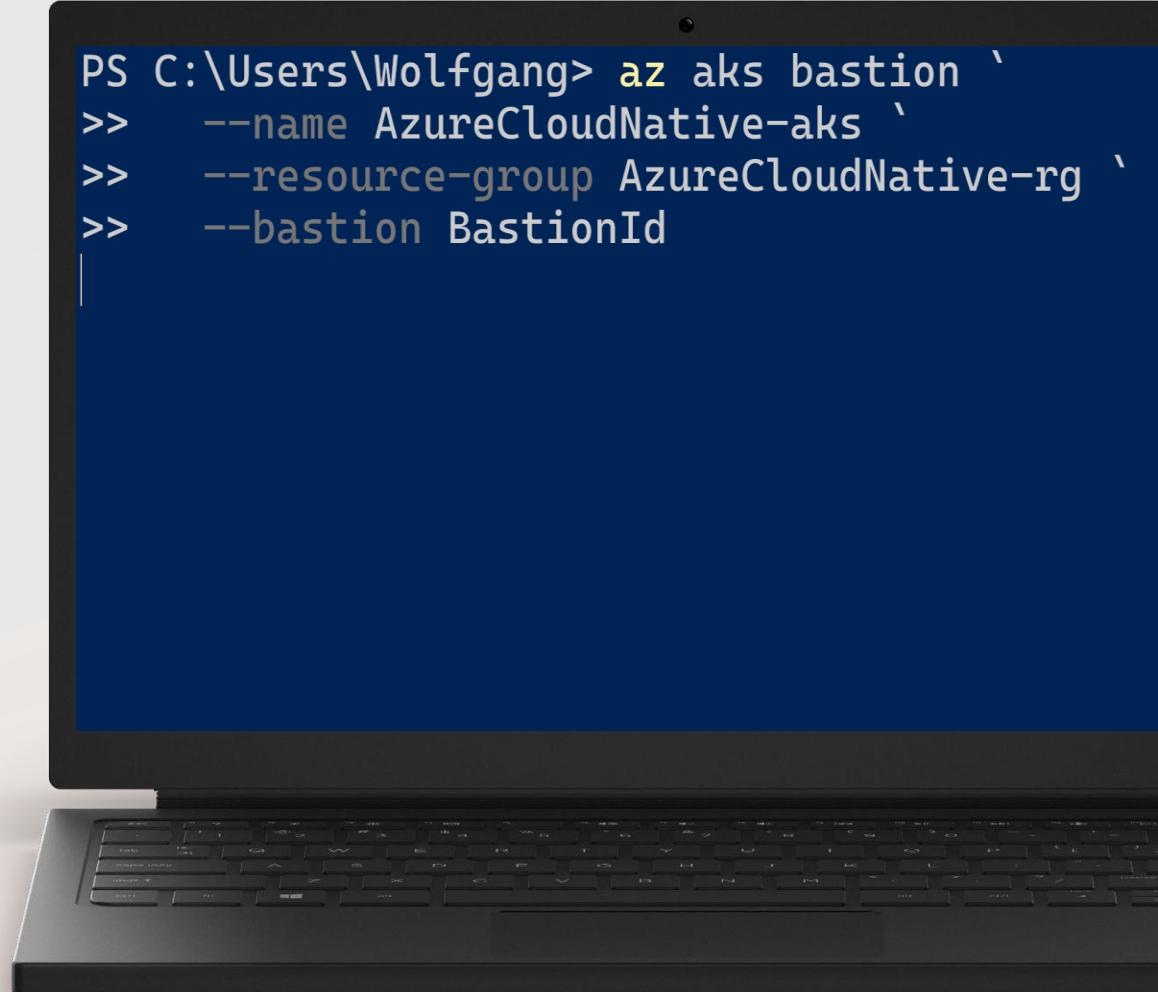


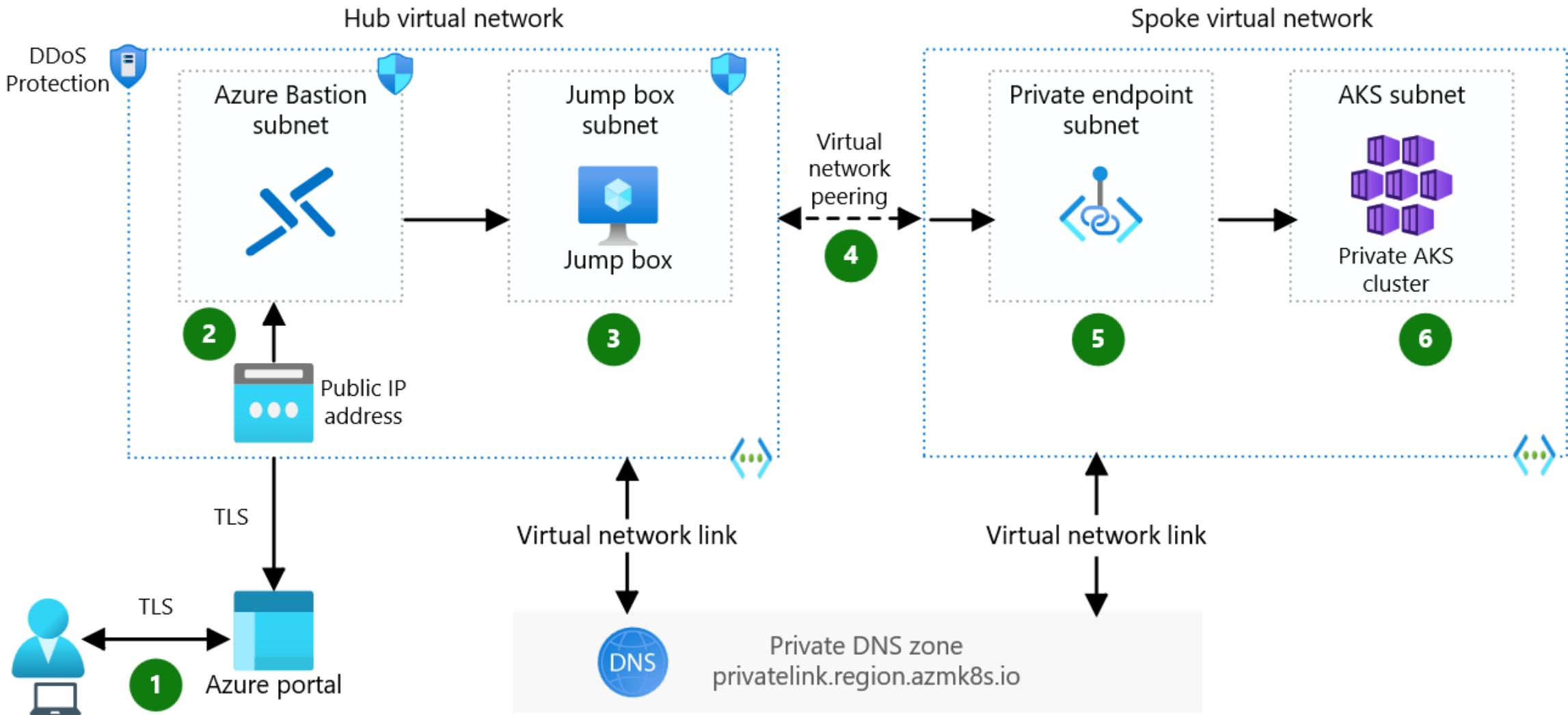


Private AKS Cluster

New feature to access AKS through Azure Bastion

```
PS C:\Users\Wolfgang> az aks bastion `>> --name AzureCloudNative-aks `>> --resource-group AzureCloudNative-rg `>> --bastion BastionId`
```





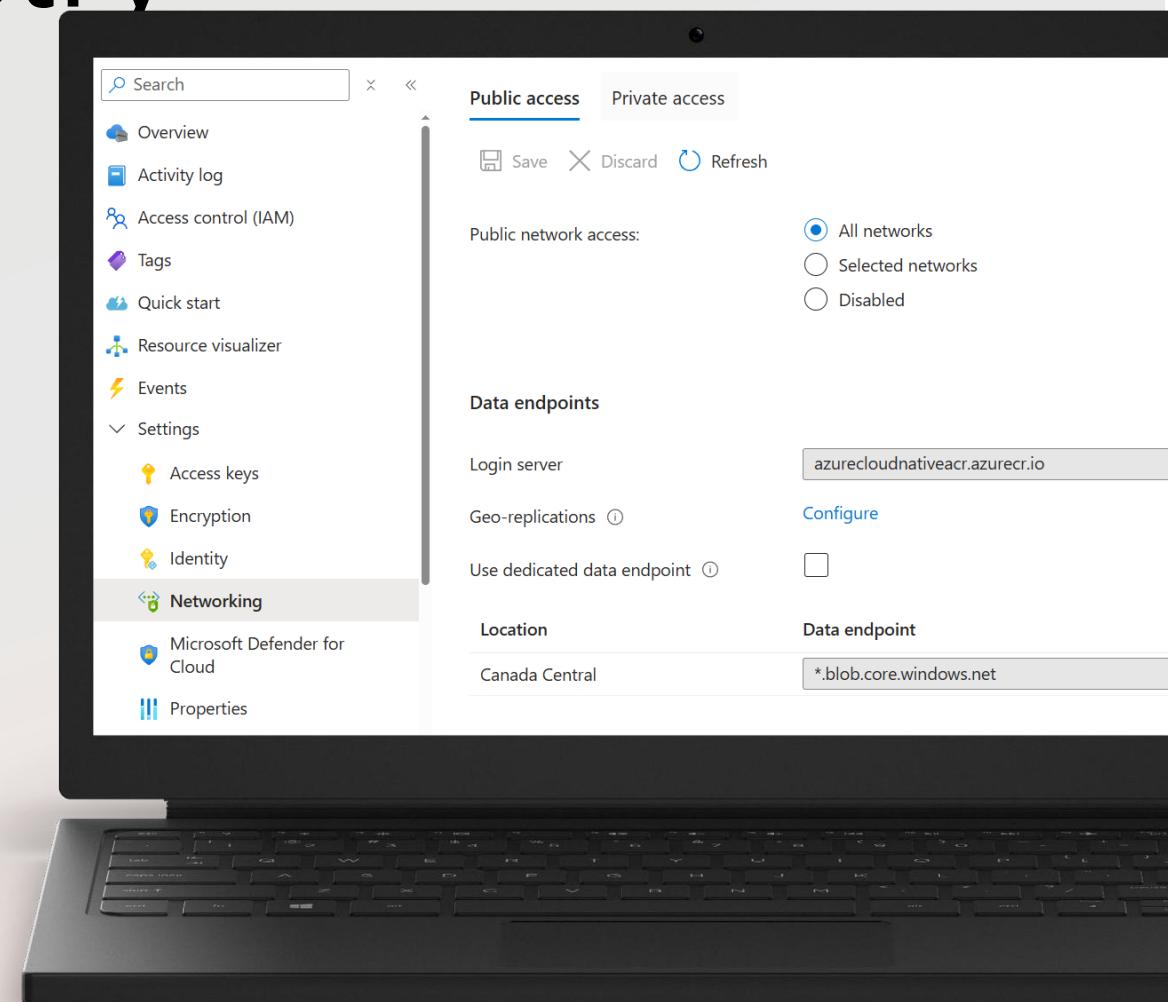
Private Azure Container Registry

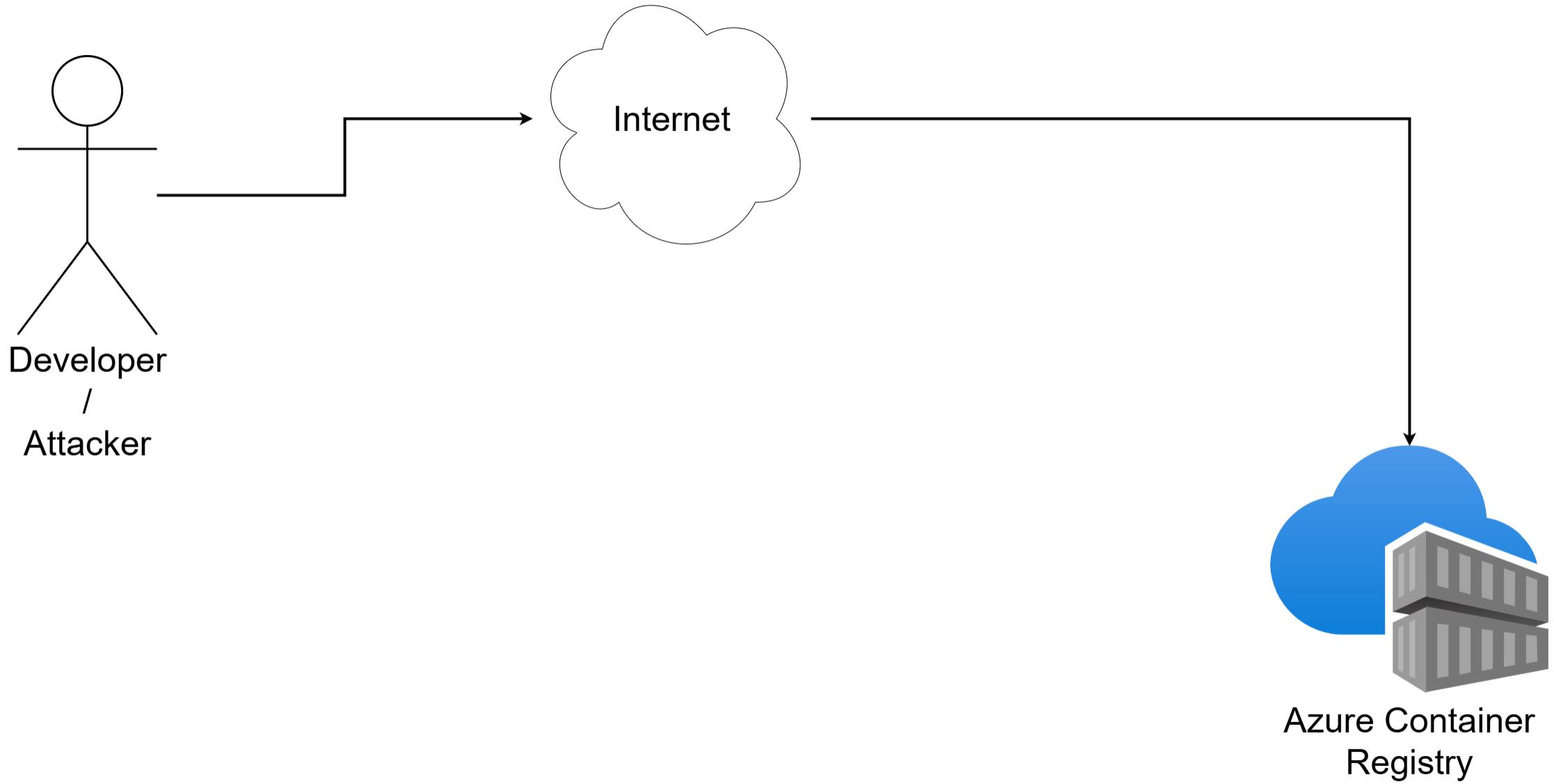
Azure Container Registry

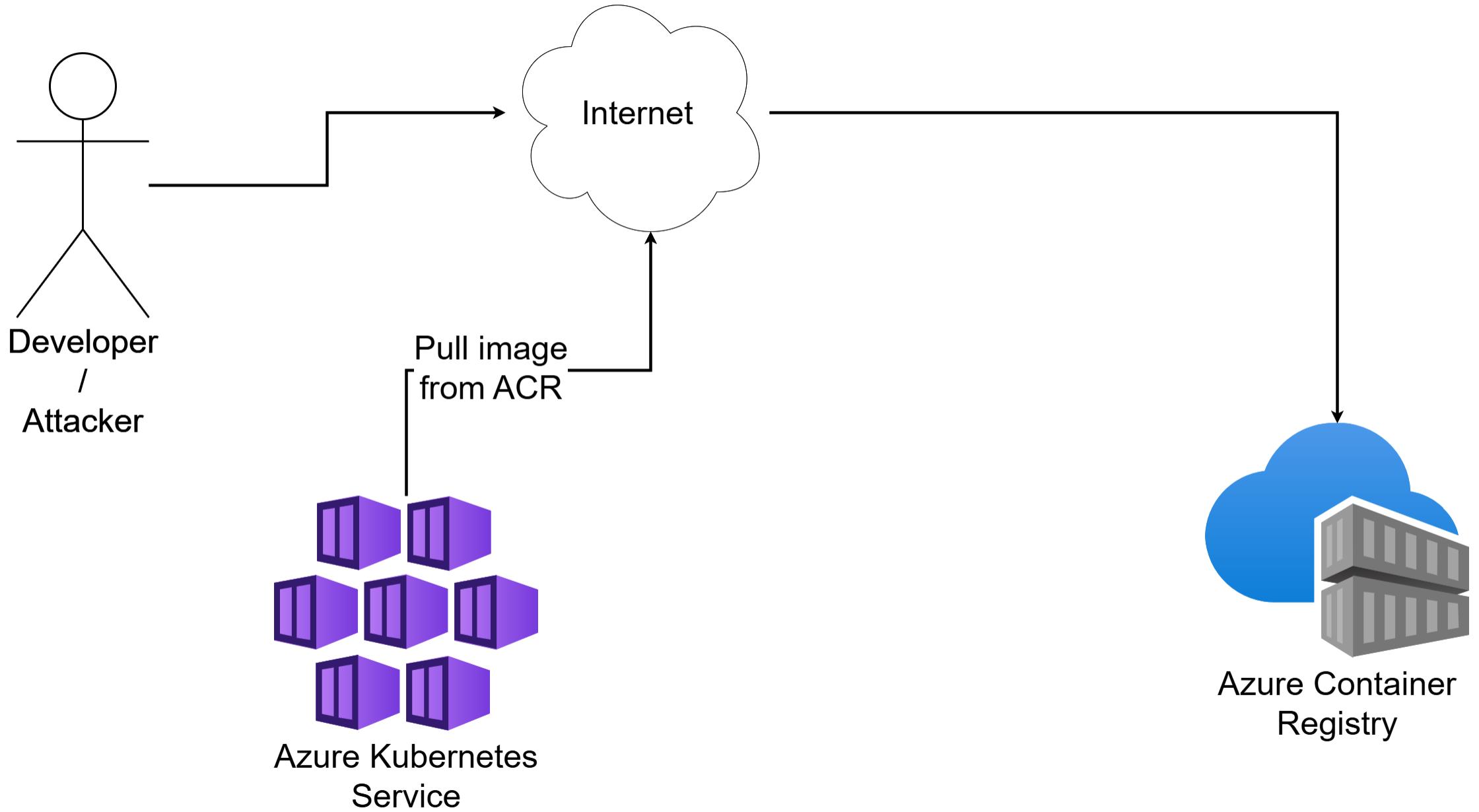
Store images in private registry

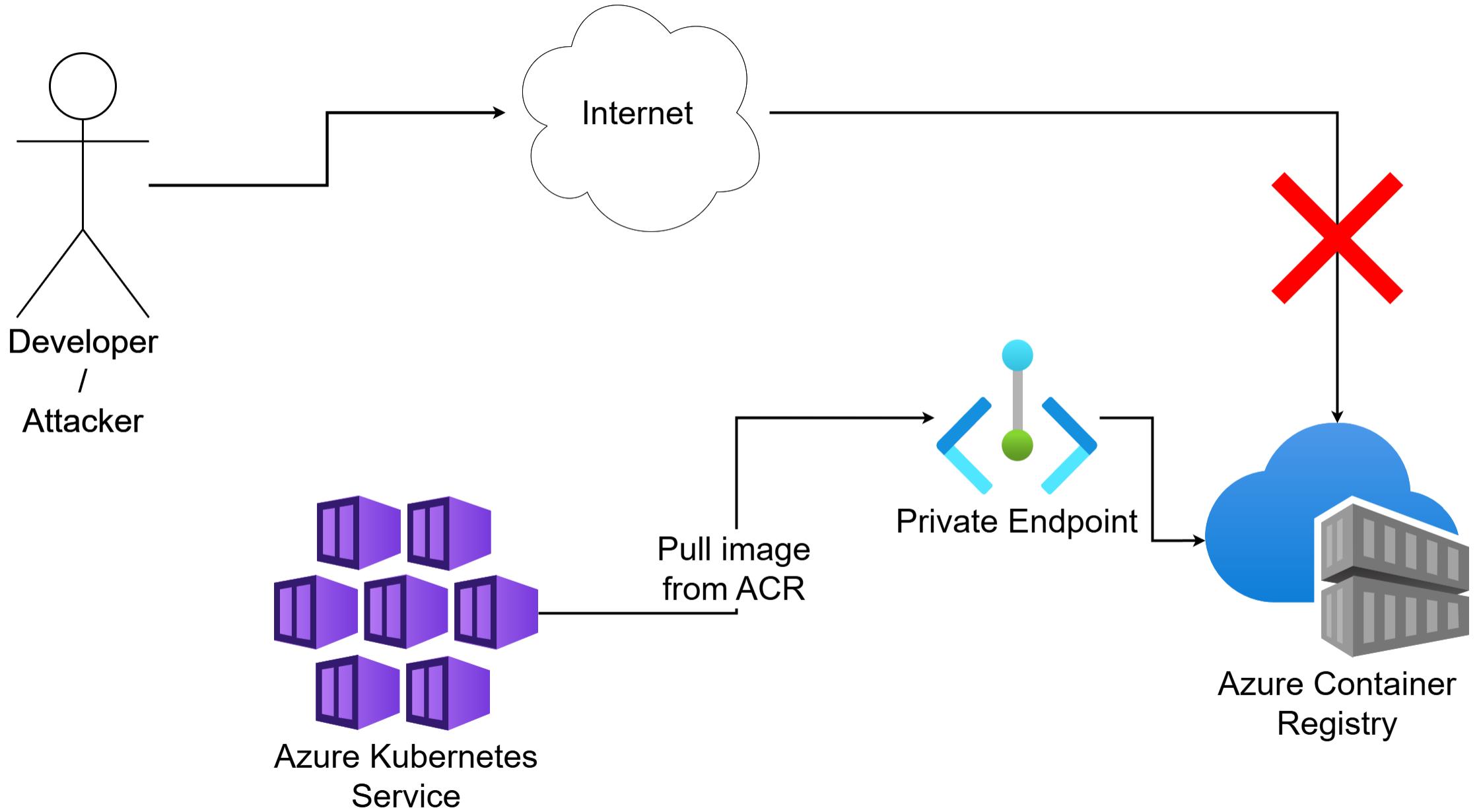
Keep everything private

AKS should download images over private connection









Azure Container Registry

No changes for pull operation necessary

Private DNS-Zone resolves
public FQDN

Build agent needs to push
images over private endpoint



Azure Key Vault Provider for Secrets Store CSI Driver

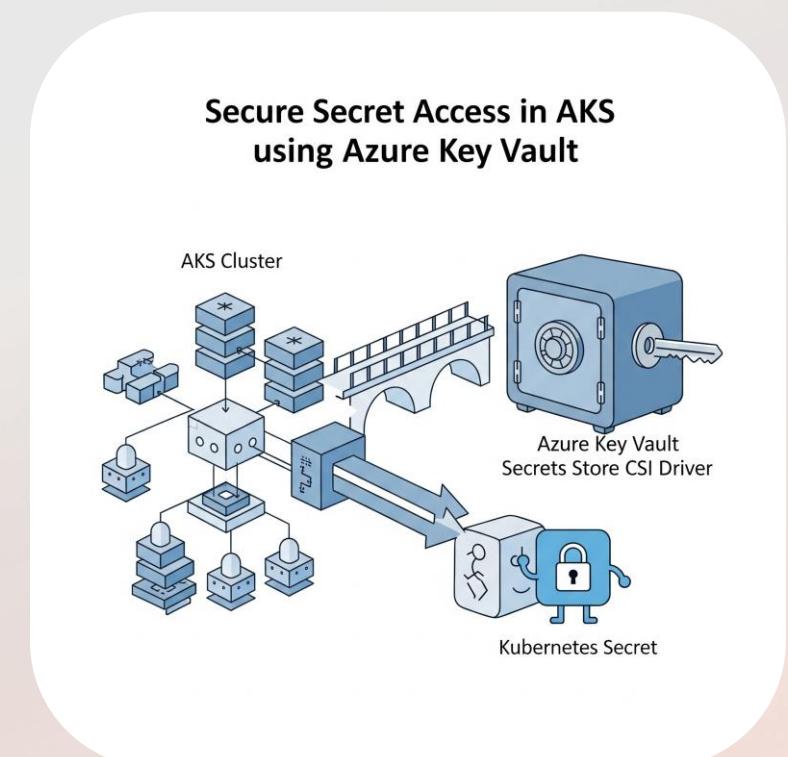
Key Vault Provider for Secrets Store

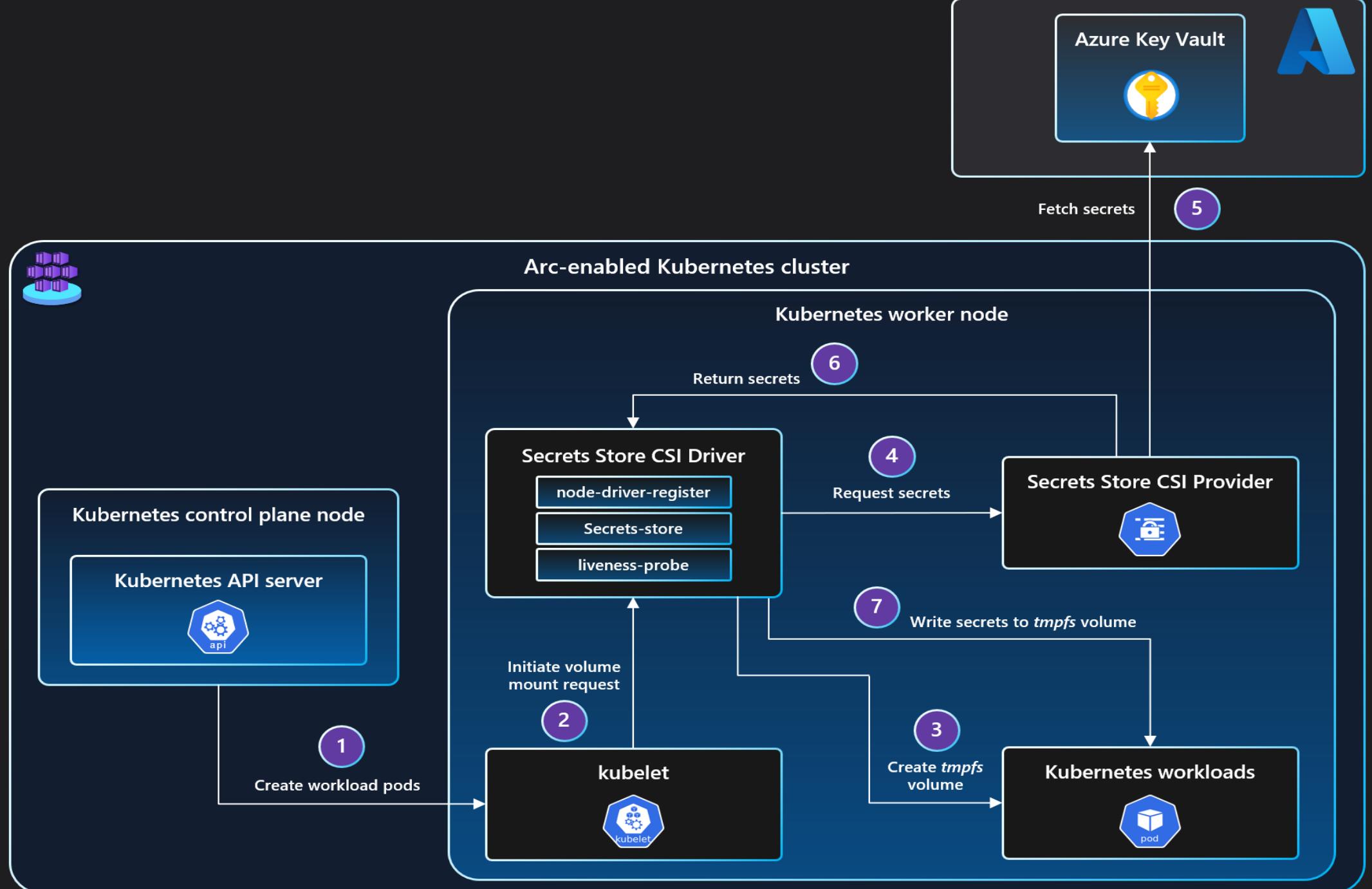
Mount secrets, keys, and certificates to pods

Auto-rotate secrets

Sync Azure Key Vault with Kubernetes secrets

Separation of concerns





Further Security Topics

Further Security Topics

Use Azure Linux as your node OS

Disable SSH access

Disable local accounts

Install the Azure Policy addon

Microsoft Defender for Containers

Setup your cluster using AKS Automatic

Further Security Topics

Only run signed images

Validate image integrity

Limit the pod privileges

Set the security context

Reduce the pod capabilities

Configure seccom (secure computing)

AKS Security Simplified for Developers

Wolfgang Ofner
Senior Cloud Architect