AKS Networking Plugin
Kubenet vs Azure CNI

Houssem Dellai

Network Plugin options in AKS

Kubenet (Basic Networking)

Default mode for AKS
Pod CIDR
Internal network

Azure CNI (Advanced Networking)

Dynamic IP allocation
Overlay mode

Cilium

Bring Your Own (BYO)

Calico CNI

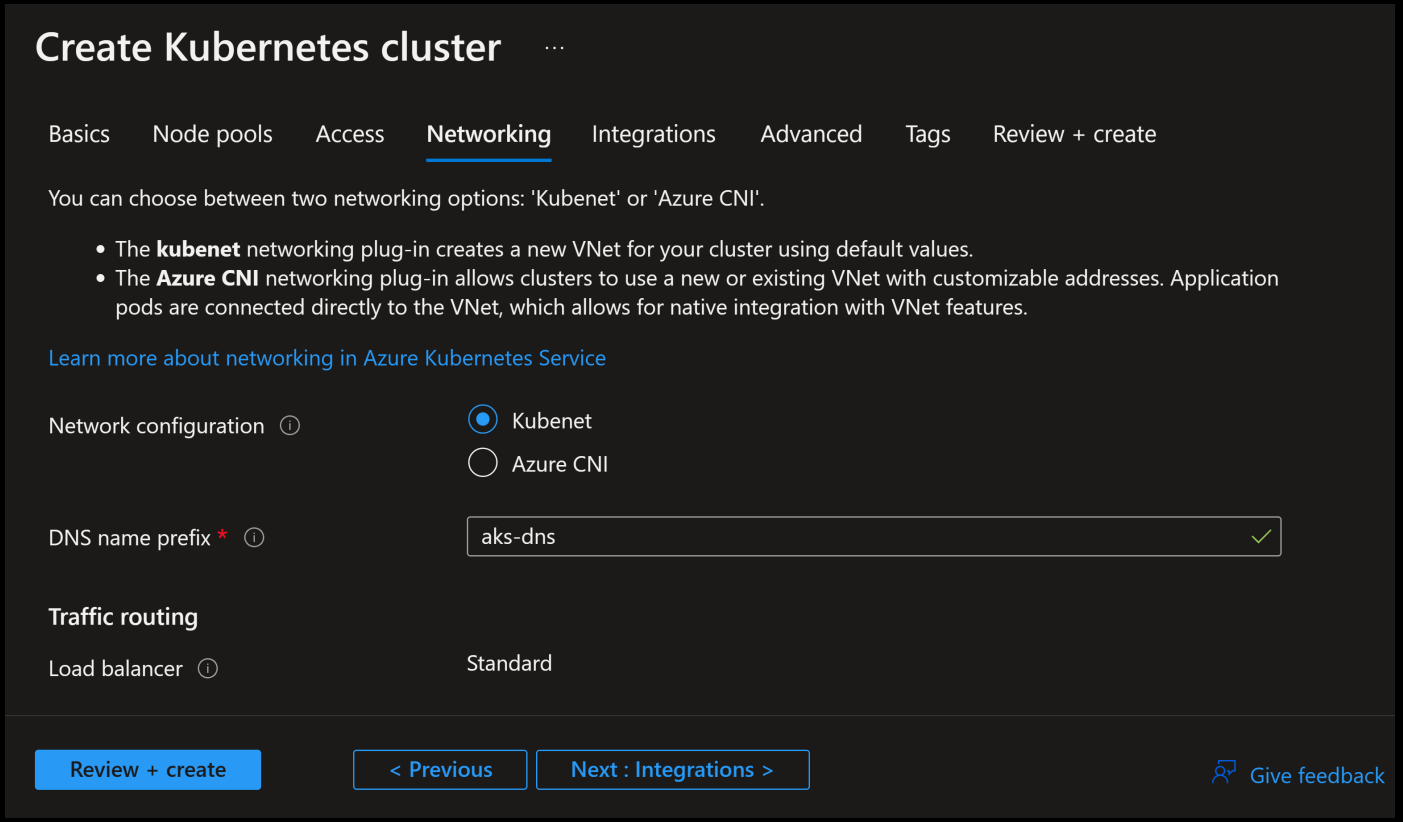
Network Plugin options in AKS

Kubenet (Basic Networking)
Default mode for AKS
Pod CIDR
Internal network

Azure CNI (Advanced Networking)
Dynamic IP allocation
Overlay mode

Cilium

Bring Your Own (BYO)
Calico CNI



AKS Network Plugin: Kubenet

Nodes get an IP address from the Azure virtual network subnet.

Pods receive an IP address from a logically different address space to the Azure virtual network subnet of the nodes.

Network address translation (NAT) is then configured so that the pods can reach resources on the VNET.

The source IP address of the traffic is NAT'd to the node's primary IP address.
This reduces the number of IP addresses that you need to reserve in your network space for pods to use.

Create an AKS cluster with Kubenet

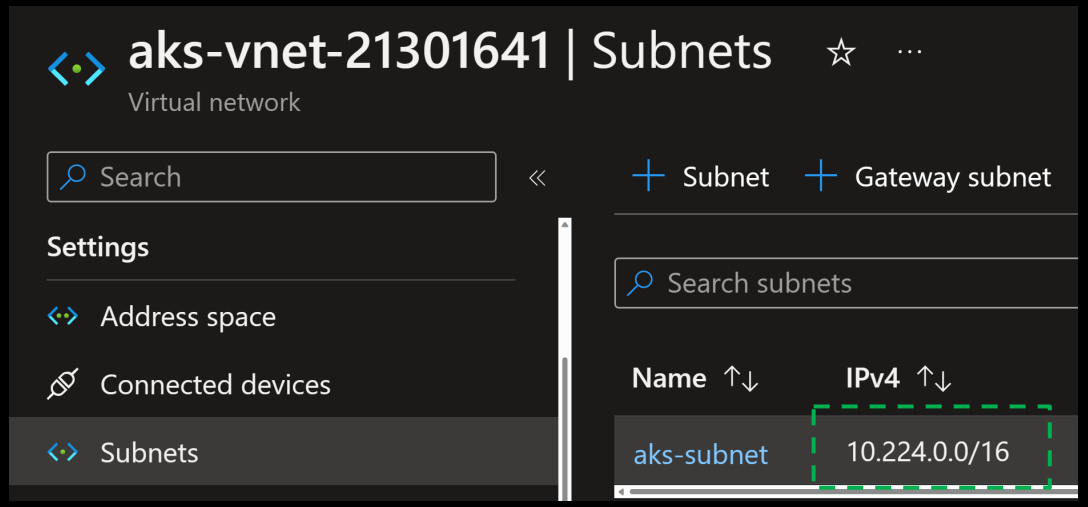
```
az group create --name rg-aks-cni --location westeurope

az aks create -g rg-aks-cni -n aks-cni --network-plugin kubenet
```

Name	Type
45508639-9b7d-468c-9eec-e0602fab461f	Public IP address
aks-agentpool-21301641-nsg	Network security group
aks-agentpool-21301641-routetable	Route table
aks-kubenet-agentpool	Managed Identity
aks-nodepool1-80782964-vmss	Virtual machine scale set
aks-vnet-21301641	Virtual network
kubernetes	Load balancer

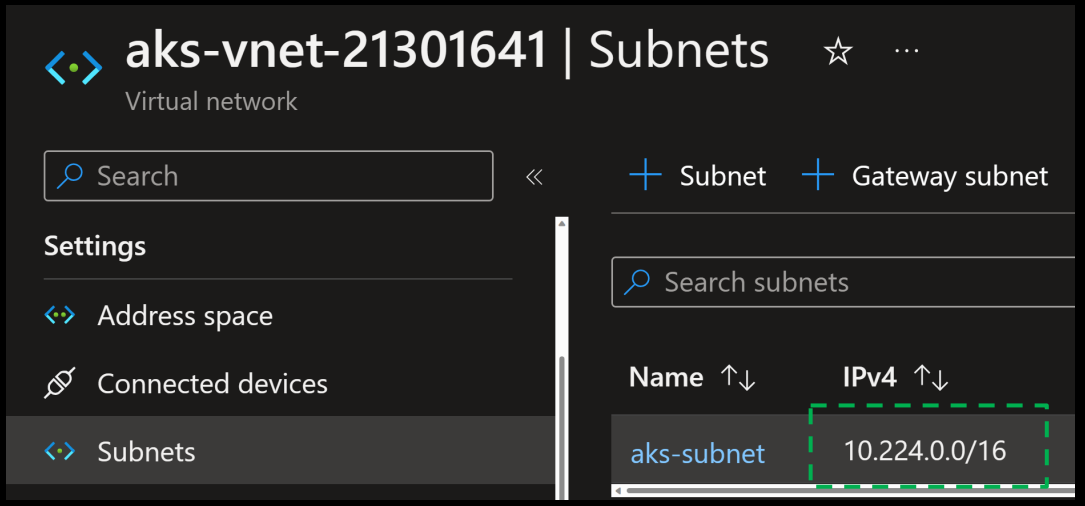
Kubenet: Nodes IPs are from Subnet

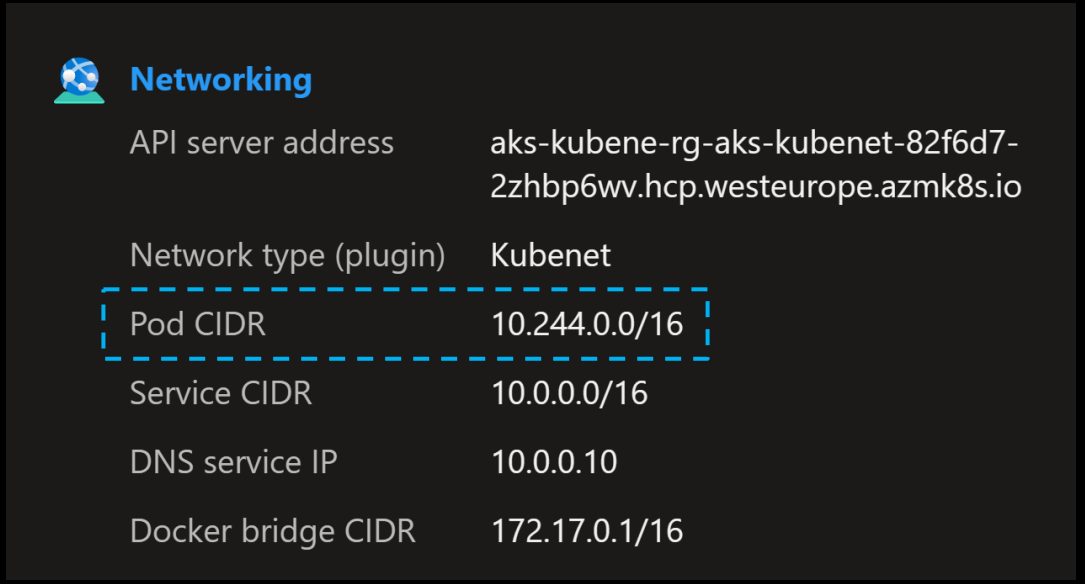
```
$ kubectl get nodes -o wide
NAME                                  INTERNAL-IP
aks-nodepool1-10626751-vmss000000     10.224.0.5
aks-nodepool1-10626751-vmss000001     10.224.0.6
aks-nodepool1-10626751-vmss000002     10.224.0.4
$
```



Kubenet: Pods IPs are from Pod CIDR (except few system Pods)

```
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-10626751-vmss000000   10.224.0.5
aks-nodepool1-10626751-vmss000001   10.224.0.6
aks-nodepool1-10626751-vmss000002   10.224.0.4
$
$ kubectl get pods -A -o wide
NAMESPACE     NAME                                  IP
kube-system   azure-ip-masq-agent-4rh2p             10.224.0.5
kube-system   azure-ip-masq-agent-5w5r8             10.224.0.4
kube-system   azure-ip-masq-agent-sv8m4             10.224.0.6
kube-system   cloud-node-manager-7sblb              10.224.0.4
kube-system   cloud-node-manager-8fstz              10.224.0.5
kube-system   cloud-node-manager-jg826              10.224.0.6
kube-system   coredns-59b6bf8b4f-66hxq              10.244.0.8
kube-system   coredns-59b6bf8b4f-t8wnb              10.244.0.6
kube-system   coredns-autoscaler-5655d66f64-2mvl8   10.244.0.7
kube-system   csi-azuredisk-node-fxbnk              10.224.0.6
kube-system   csi-azuredisk-node-mcvzp              10.224.0.4
kube-system   csi-azuredisk-node-mprpw              10.224.0.5
kube-system   csi-azurefile-node-78npj              10.224.0.6
kube-system   csi-azurefile-node-cf6bl              10.224.0.5
kube-system   csi-azurefile-node-p9xbs              10.224.0.4
kube-system   konnectivity-agent-7cc8ddb5bc-flqkk   10.244.2.2
kube-system   konnectivity-agent-7cc8ddb5bc-wzfmt   10.244.1.2
kube-system   kube-proxy-6768z                      10.224.0.6
kube-system   kube-proxy-hs997                      10.224.0.4
kube-system   kube-proxy-k7kjk                      10.224.0.5
kube-system   metrics-server-5f8d84558d-d5qhg       10.244.0.10
kube-system   metrics-server-5f8d84558d-wvjzf       10.244.0.9
$
```





Kubenet: Pods IPs are from Pod CIDR

```
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-10626751-vmss000000   10.224.0.5
aks-nodepool1-10626751-vmss000001   10.224.0.6
aks-nodepool1-10626751-vmss000002   10.224.0.4
$
$
$ kubectl create deployment nginx --image=nginx --replicas=10
deployment.apps/nginx created
$
$ kubectl get pods -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP
nginx-8f458dc5b-2qp4n   1/1     Running   0          25s   10.244.2.5
nginx-8f458dc5b-6clrc   1/1     Running   0          26s   10.244.0.11
nginx-8f458dc5b-c9zrc   1/1     Running   0          25s   10.244.1.4
nginx-8f458dc5b-f7fdh   1/1     Running   0          25s   10.244.2.3
nginx-8f458dc5b-jgm7s   1/1     Running   0          25s   10.244.1.5
nginx-8f458dc5b-mw5gg   1/1     Running   0          25s   10.244.0.13
nginx-8f458dc5b-p4mn5   1/1     Running   0          25s   10.244.1.6
nginx-8f458dc5b-pnxwg   1/1     Running   0          26s   10.244.2.4
nginx-8f458dc5b-rckqv   1/1     Running   0          25s   10.244.0.12
nginx-8f458dc5b-tlqbp   1/1     Running   0          26s   10.244.1.3
$
$
$ kubectl get node -o jsonpath='{.items[*].spec.podCIDR}'
10.244.2.0/24 10.244.1.0/24 10.244.0.0/24
$
```

Each Node owns /24 CIDR for its Pods

Node pool

Max pods per node
110

aks-vnet-21301641 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

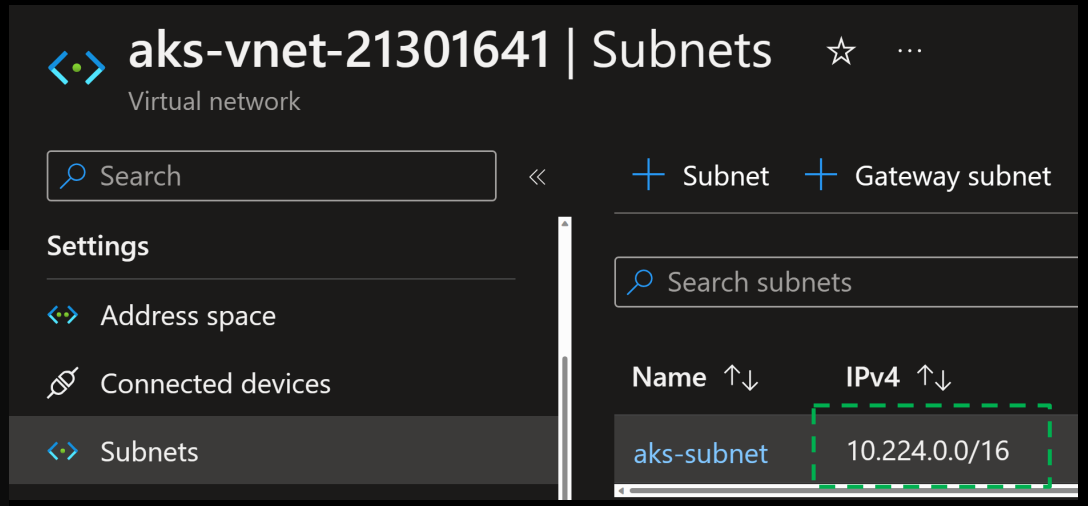
Name	IPv4
aks-subnet	10.224.0.0/16

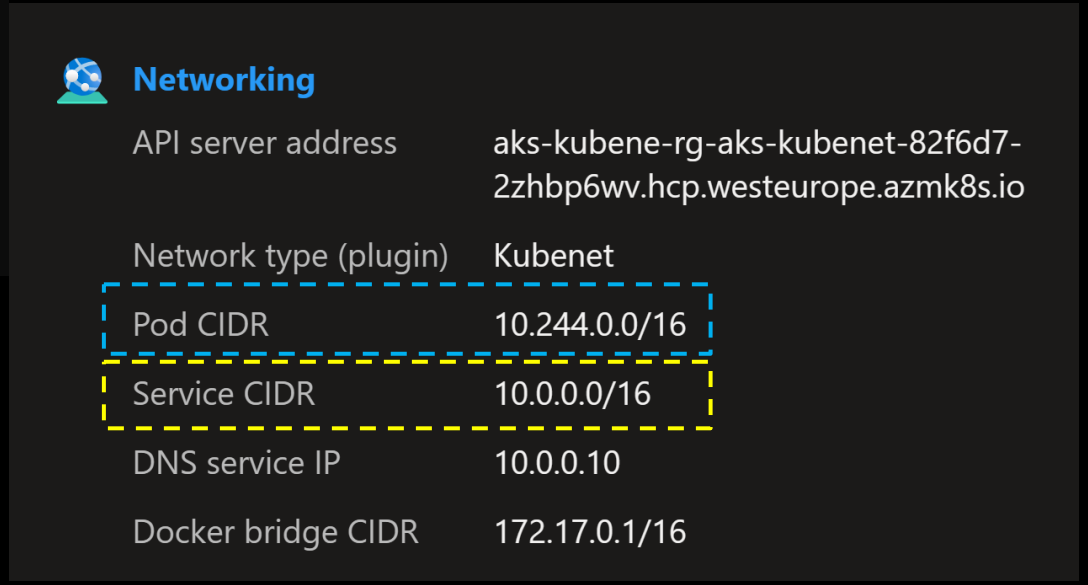
Networking

API server address	aks-kubene-rg-aks-kubenet-82f6d7-2zhbp6wv.hcp.westeurope.azmk8s.io
Network type (plugin)	Kubenet
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

Kubenet: Services IPs are from Service CIDR (like CNI)

```
$ kubectl get nodes -o wide
NAME                                  INTERNAL-IP
aks-nodepool1-10626751-vmss000000     10.224.0.5
aks-nodepool1-10626751-vmss000001     10.224.0.6
aks-nodepool1-10626751-vmss000002     10.224.0.4
$
$
$ kubectl create deployment nginx --image=nginx --replicas=10
deployment.apps/nginx created
$
$ kubectl get pods -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP
nginx-8f458dc5b-2qp4n   1/1     Running   0          25s   10.244.2.5
nginx-8f458dc5b-6clrc   1/1     Running   0          26s   10.244.0.11
nginx-8f458dc5b-c9zrc   1/1     Running   0          25s   10.244.1.4
nginx-8f458dc5b-f7fdh   1/1     Running   0          25s   10.244.2.3
nginx-8f458dc5b-jgm7s   1/1     Running   0          25s   10.244.1.5
nginx-8f458dc5b-mw5gg   1/1     Running   0          25s   10.244.0.13
nginx-8f458dc5b-p4mn5   1/1     Running   0          25s   10.244.1.6
nginx-8f458dc5b-pnxwg   1/1     Running   0          26s   10.244.2.4
nginx-8f458dc5b-rckqv   1/1     Running   0          25s   10.244.0.12
nginx-8f458dc5b-tlqbp   1/1     Running   0          26s   10.244.1.3
$
$
$ kubectl get svc -A
NAMESPACE     NAME             TYPE        CLUSTER-IP
default       kubernetes       ClusterIP   10.0.0.1
default       nginx            ClusterIP   10.0.24.234
kube-system   kube-dns         ClusterIP   10.0.0.10
kube-system   metrics-server   ClusterIP   10.0.228.174
$
```





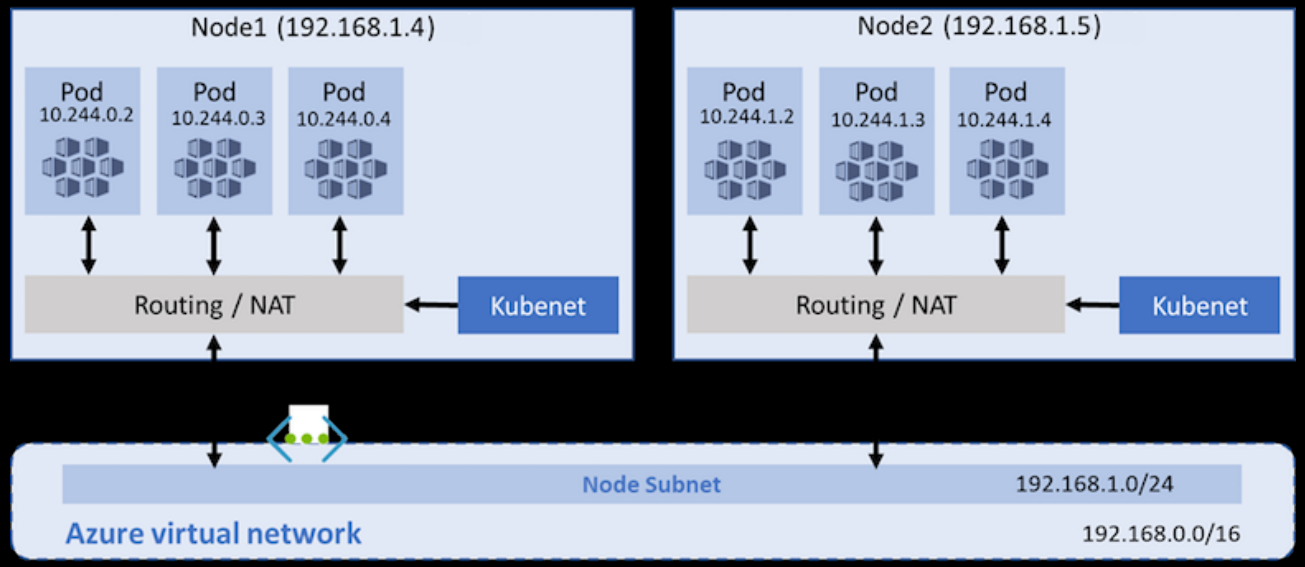
How Kubenet works ?

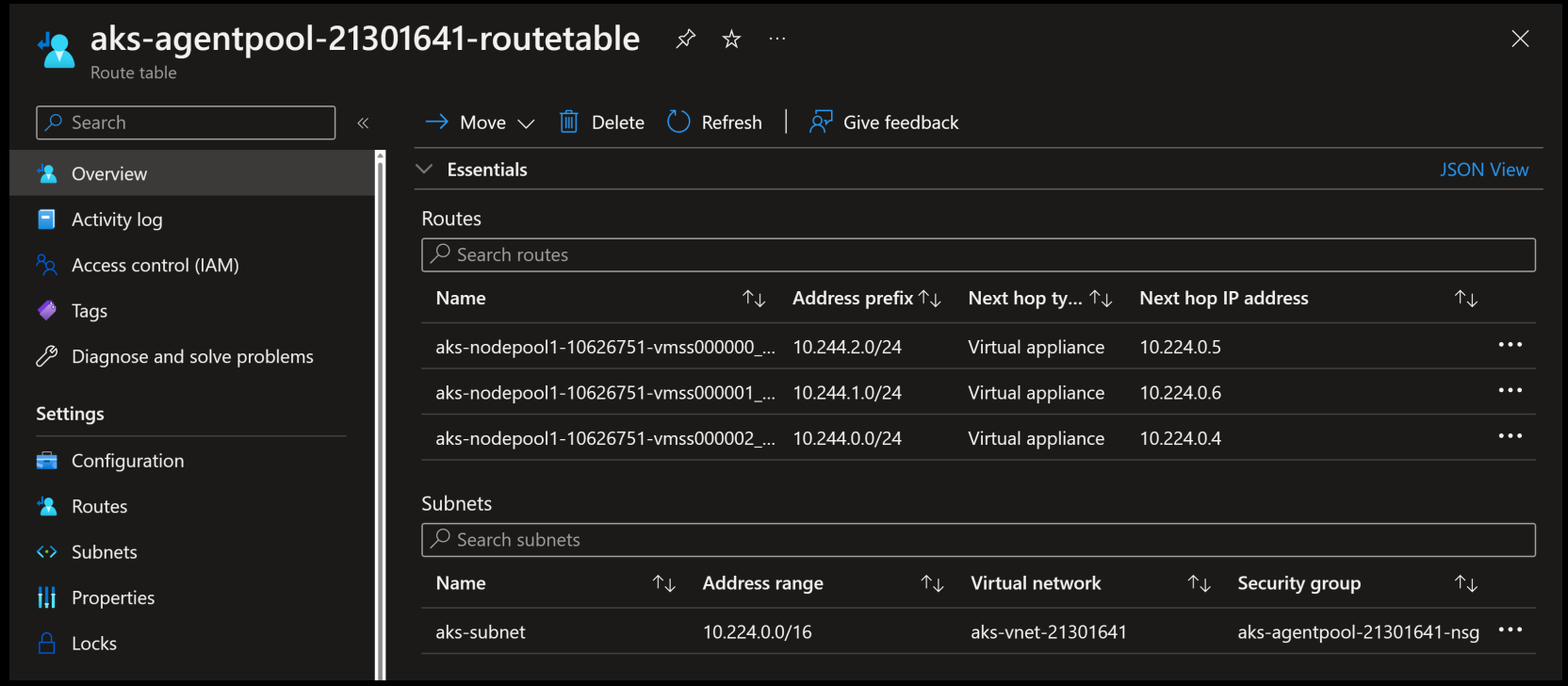
Pods can't communicate directly with each other.

Instead, User Defined Routing (UDR) and IP forwarding is used for connectivity between pods across nodes.

The source IP address of the traffic is NAT'd to the node's primary IP address.

Pods from different nodes communicates through Node route table.





Kubenet: Limitations

Azure supports a maximum of 400 routes in a UDR, so you can't have an AKS cluster larger than 400 nodes.

An additional hop is required in the design of kubenet, which adds minor latency to pod communication.

Kubenet doesn’t support:

- AKS Virtual Nodes
- Windows Nodepools
- Azure Network Policies
  You can use Calico Network Policies, as they are supported with kubenet.

Recap: AKS with Kubenet

Nodes are assigned IP addresses from Subnet.

Pods are assigned IP addresses from Pod CIDR (internal network in Kubernetes).

Some system Pods are assigned the same IP address of the Node !
Because Pods can request that.

AKS Network Plugin: Azure CNI

Every pod gets an IP address from the subnet and can be accessed directly.

These IP addresses must be unique across the network space and must be planned in advance.

Each node has a configuration parameter for the maximum number of pods that it supports.

The equivalent number of IP addresses per node are then reserved up front for that node.

This approach requires more planning.

Might lead to IP address exhaustion or the need to rebuild clusters in a larger subnet as application demands grow.

Create an AKS cluster with Azure CNI

```
az group create --name rg-aks-cni --location westeurope

az aks create -g rg-aks-cni -n aks-cni --network-plugin azure
```

Name	Type
aks-agentpool-35006646-nsg	Network security group
aks-cni-agentpool	Managed Identity
aks-nodepool1-39057157-vmss	Virtual machine scale set
aks-vnet-35006646	Virtual network
b698b7dc-5b59-4dd0-b071-aae6a6463da3	Public IP address
kubernetes	Load balancer

Azure CNI : Node IPs are from Subnet

```
$
$ kubectl get nodes -o wide
NAME                                   INTERNAL-IP
aks-nodepool1-35527294-vmss000000      10.224.0.62
aks-nodepool1-35527294-vmss000001      10.224.0.33
aks-nodepool1-35527294-vmss000002      10.224.0.4
$
```

Node pool

Max pods per node
30

aks-vnet-21301641 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Azure CNI : Pods IPs are from Subnet

```
 $
 $ kubectl get nodes -o wide
NAME                                  INTERNAL-IP
aks-nodepool1-35527294-vmss000000     10.224.0.62
aks-nodepool1-35527294-vmss000001     10.224.0.33
aks-nodepool1-35527294-vmss000002     10.224.0.4
 $
```

Node pool

Max pods per node
30

```
 $
 $ kubectl get pods -A -o wide
NAMESPACE     NAME                                          IP
kube-system   azure-ip-masq-agent-bw2gh                     10.224.0.4
kube-system   azure-ip-masq-agent-v2z8c                     10.224.0.62
kube-system   azure-ip-masq-agent-xgfkt                     10.224.0.33
kube-system   cloud-node-manager-95mp2                      10.224.0.62
kube-system   cloud-node-manager-mnxwx                      10.224.0.4
kube-system   cloud-node-manager-xjbwd                      10.224.0.33
kube-system   coredns-59b6bf8b4f-49xf2                      10.224.0.67
kube-system   coredns-59b6bf8b4f-bdnkj                      10.224.0.5
kube-system   coredns-autoscaler-5655d66f64-955b5           10.224.0.63
kube-system   csi-azuredisk-node-478kj                      10.224.0.33
kube-system   csi-azuredisk-node-fzqfb                      10.224.0.62
kube-system   csi-azuredisk-node-m8clc                      10.224.0.4
kube-system   csi-azurefile-node-j27tj                      10.224.0.33
kube-system   csi-azurefile-node-nf5d9                      10.224.0.4
kube-system   csi-azurefile-node-w7dqk                      10.224.0.62
kube-system   konnectivity-agent-7d8d9bfc4b-jgpbg           10.224.0.40
kube-system   konnectivity-agent-7d8d9bfc4b-twvvq           10.224.0.23
kube-system   kube-proxy-bn8gj                              10.224.0.4
kube-system   kube-proxy-q4r6t                              10.224.0.33
kube-system   kube-proxy-v9mjp                              10.224.0.62
kube-system   metrics-server-5f8d84558d-csnwl               10.224.0.17
kube-system   metrics-server-5f8d84558d-z9wbm               10.224.0.10
 $
```

aks-vnet-21301641 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Networking

API server address	aks-cni-rg-aks-cni-82f6d7-8oos0rk9.hcp.westeurope.azmk8s.io
Network type (plugin)	Azure CNI
Pod CIDR	-
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

Azure CNI : Services IPs are from Service CIDR (like Kubenet)

```
$
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-35527294-vmss000000   10.224.0.62
aks-nodepool1-35527294-vmss000001   10.224.0.33
aks-nodepool1-35527294-vmss000002   10.224.0.4
$
$
$ kubectl get pods -A -o wide
NAMESPACE     NAME                                      IP
kube-system   azure-ip-masq-agent-bw2gh                 10.224.0.4
kube-system   azure-ip-masq-agent-v2z8c                 10.224.0.62
kube-system   azure-ip-masq-agent-xgfkt                 10.224.0.33
kube-system   cloud-node-manager-95mp2                  10.224.0.62
kube-system   cloud-node-manager-mnxwx                  10.224.0.4
kube-system   cloud-node-manager-xjbwd                  10.224.0.33
kube-system   coredns-59b6bf8b4f-49xf2                  10.224.0.67
kube-system   coredns-59b6bf8b4f-bdnkj                  10.224.0.5
kube-system   coredns-autoscaler-5655d66f64-955b5       10.224.0.63
kube-system   csi-azuredisk-node-478kj                  10.224.0.33
kube-system   csi-azuredisk-node-fzqfb                  10.224.0.62
kube-system   csi-azuredisk-node-m8clc                  10.224.0.4
kube-system   csi-azurefile-node-j27tj                  10.224.0.33
kube-system   csi-azurefile-node-nf5d9                  10.224.0.4
kube-system   csi-azurefile-node-w7dgk                  10.224.0.62
$
$ kubectl get services -A
NAMESPACE     NAME            TYPE        CLUSTER-IP
default       kubernetes      ClusterIP   10.0.0.1
kube-system   kube-dns        ClusterIP   10.0.0.10
kube-system   metrics-server  ClusterIP   10.0.249.103
$
```

Node pool

Max pods per node
30

aks-vnet-21301641 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Networking

API server address	aks-cni-rg-aks-cni-82f6d7-8oos0rk9.hcp.westeurope.azmk8s.io
Network type (plugin)	Azure CNI
Pod CIDR	-
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

IPs utilization in Kubenet vs Azure CNI

How many Nodes and Pods could be deployed into a Subnet with CIDR /24 ?

Subnet reserves the first three IPs for management operations.
Default maximum of 110 pods per node with kubenet.
Default maximum of 30 pods per node with Azure CNI.

Subnet CIDR /24	Kubenet	Azure CNI
#Nodes	251 nodes	8 nodes
#Pods	27,610 (251*110) pods	240 (8*30) pods

⇒ Problem of IP exhaustion in Azure CNI.
⇒ Could be resolved using Azure CNI Overlay.

Kubenet or Azure CNI ?

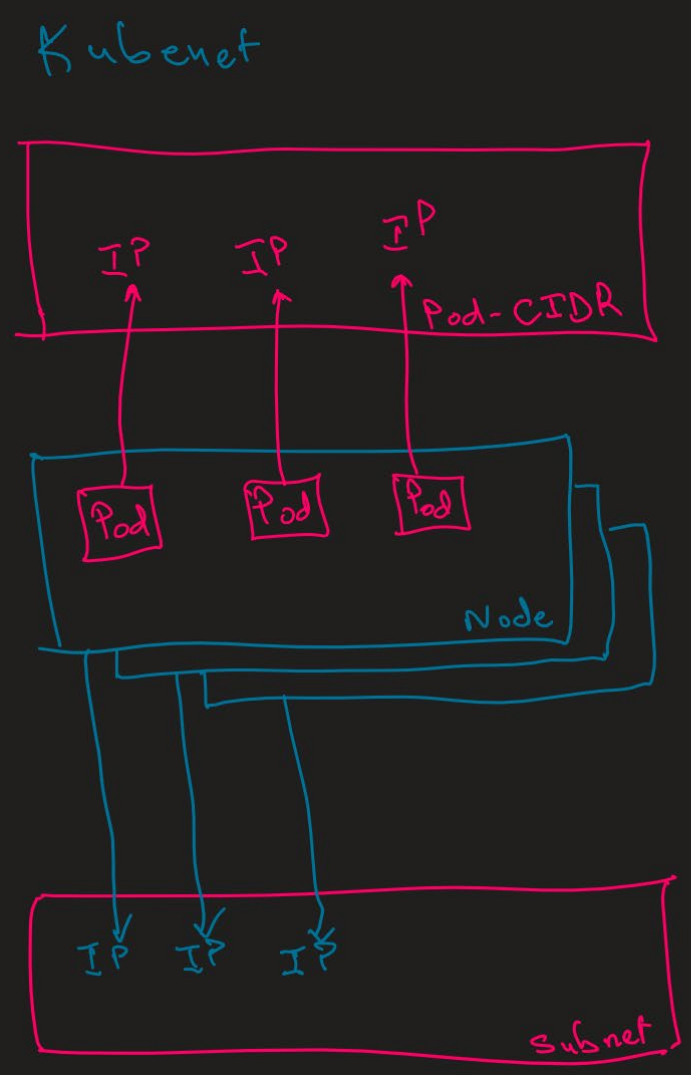
Use kubenet when:

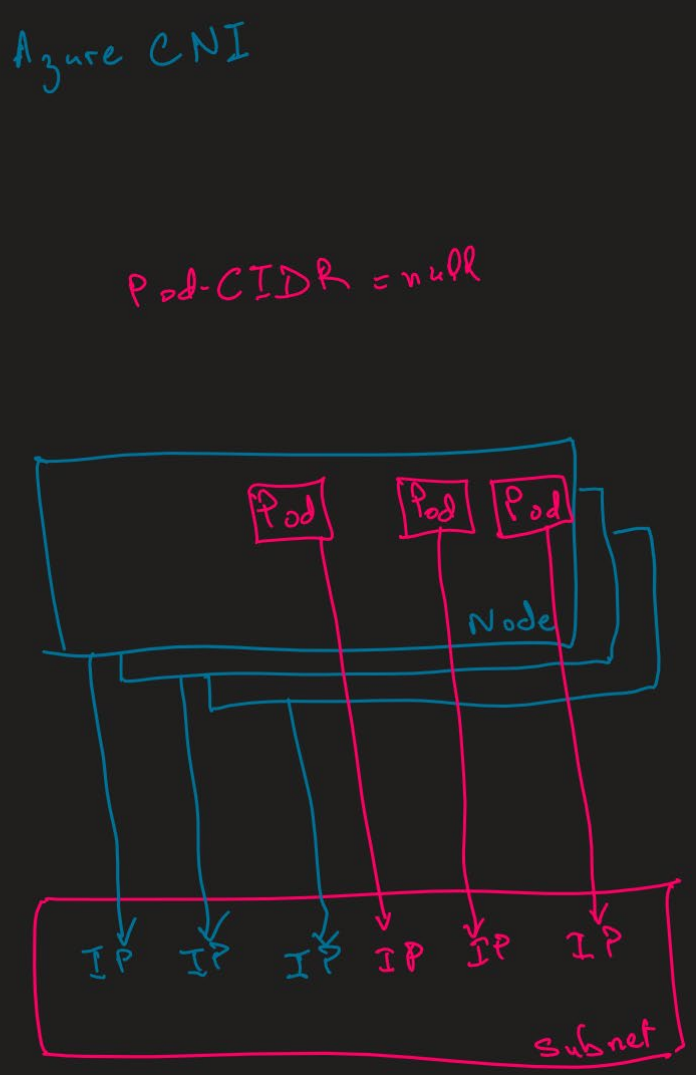
- You have limited IP address space.
- Most of the pod communication is within the cluster.
- You don't need advanced AKS features such as virtual nodes or Azure Network Policy.

Use Azure CNI when:

- You have available IP address space.
- Most of the pod communication is to resources outside of the cluster.
- You don't want to manage user defined routes for pod connectivity.
- You need AKS advanced features such as virtual nodes or Azure Network Policy.

Kubenet vs Azure CNI





Azure CNI Overlay overview

In overlay networking, only the Kubernetes cluster nodes are assigned IPs from a subnet.

Pods receive IPs from a private CIDR that is provided at the time of cluster creation.

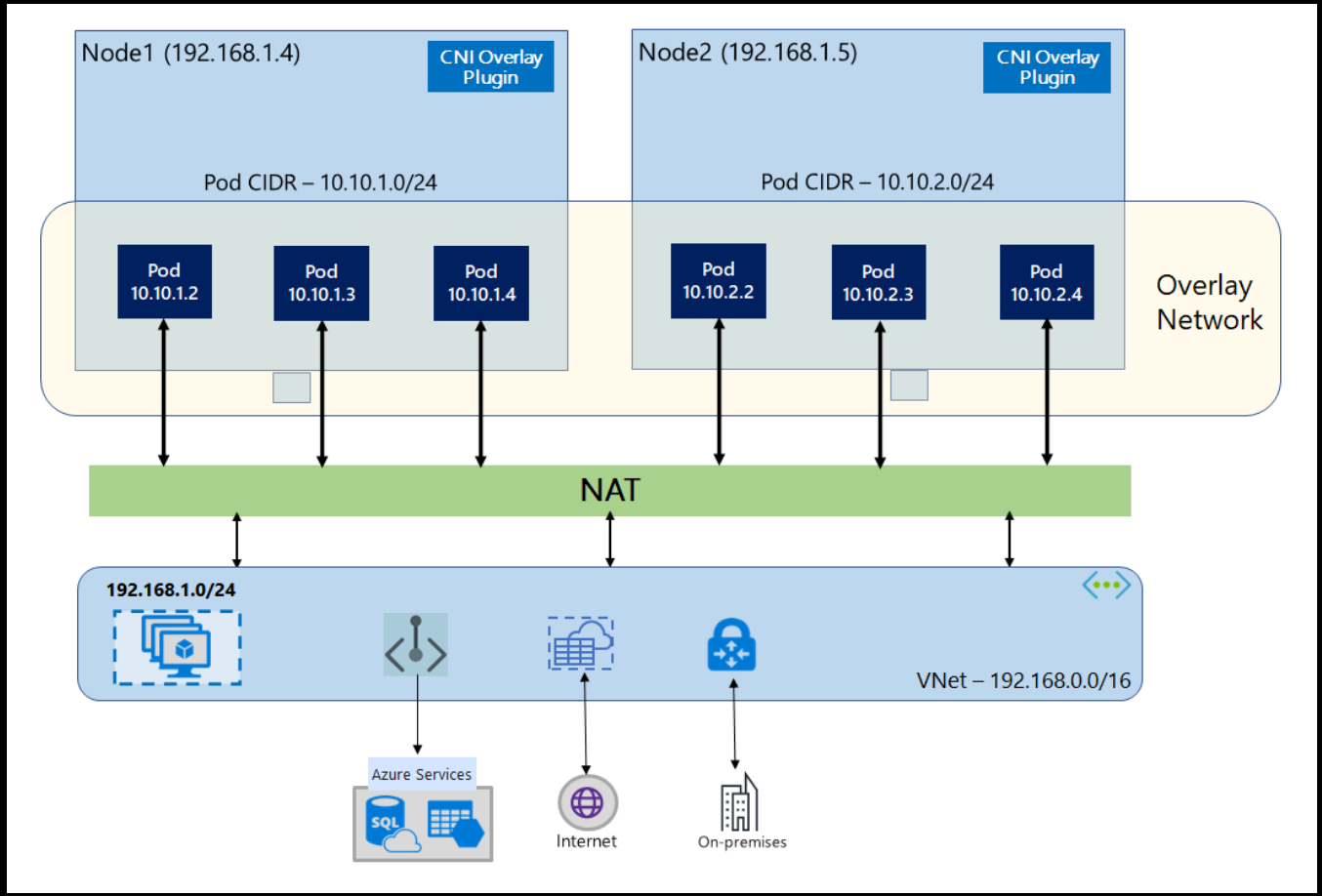
Each node is assigned:

- An IP address from Subnet
- /24 from Pod-CIDR to assign to its Pods.

There is no need to provision custom routes on the cluster subnet or use an encapsulation method to tunnel traffic between pods.

Promises connectivity performance between pods.

Endpoints outside the cluster can't connect to a pod directly.



Create cluster with Azure CNI Overlay

```
$ az group create -n rg-aks-cni-overlay -l westeurope

$ az aks create -n aks-cni-overlay -g rg-aks-cni-overlay `
                --network-plugin azure `
                --network-plugin-mode overlay `
                --pod-cidr 192.168.0.0/16
```

Name	Type
aks-agentpool-51020892-nsg	Network security group
aks-cni-overlay-agentpool	Managed Identity
aks-nodepool1-14046306-vmss	Virtual machine scale set
aks-vnet-51020892	Virtual network
e1d2d241-1296-4c89-90ba-b300c5a5405d	Public IP address
kubernetes	Load balancer

Azure CNI Overlay: Node IPs are from Subnet

```
$
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-14046306-vmss000000   10.224.0.4
aks-nodepool1-14046306-vmss000001   10.224.0.5
aks-nodepool1-14046306-vmss000002   10.224.0.6
$
$
$ kubectl get pods -A -o wide
NAMESPACE     NAME                                      IP
kube-system   azure-cns-8p4pq                           10.224.0.6
kube-system   azure-cns-l76pf                           10.224.0.4
kube-system   azure-cns-mjz6p                           10.224.0.5
kube-system   azure-ip-masq-agent-9h2dw                 10.224.0.5
kube-system   azure-ip-masq-agent-ldwc8                 10.224.0.6
kube-system   azure-ip-masq-agent-lr64z                 10.224.0.4
kube-system   cloud-node-manager-d7p55                  10.224.0.6
kube-system   cloud-node-manager-fhkmr                  10.224.0.5
kube-system   cloud-node-manager-pcqdn                  10.224.0.4
kube-system   coredns-59b6bf8b4f-2nz4d                  192.168.1.33
kube-system   coredns-59b6bf8b4f-shk6w                  192.168.0.165
kube-system   coredns-autoscaler-5655d66f64-wwfdg       192.168.1.142
kube-system   csi-azuredisk-node-6wpqt                  10.224.0.5
kube-system   csi-azuredisk-node-d992s                  10.224.0.4
kube-system   csi-azuredisk-node-fxgn2                  10.224.0.6
kube-system   csi-azurefile-node-74vcl                  10.224.0.6
kube-system   csi-azurefile-node-f7q72                  10.224.0.4
kube-system   csi-azurefile-node-zchd4                  10.224.0.5
kube-system   konnectivity-agent-79bc94d486-d4t9j       192.168.1.53
kube-system   konnectivity-agent-79bc94d486-wb4rk       192.168.1.62
kube-system   kube-proxy-p28kk                          10.224.0.6
kube-system   kube-proxy-p2k5c                          10.224.0.4
kube-system   kube-proxy-q6d4b                          10.224.0.5
kube-system   metrics-server-5f8d84558d-dt95p           192.168.0.233
kube-system   metrics-server-5f8d84558d-jv46s           192.168.0.151
$
```

Node pool

Max pods per node
250

aks-vnet-51020892 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Networking

API server address	aks-cni-ov-rg-aks-cni-overl-82f6d7-iznmpgpa.hcp.westeurope.azmk8s.io
Network type (plugin)	Azure CNI
Pod CIDR	-
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

```
"networkProfile": {
    "networkPlugin": "azure",
    "networkPluginMode": "overlay",
    "networkDataplane": "azure",
    "loadBalancerSku": "Standard",
    "loadBalancerProfile": {…
    },
    "podCidr": "192.168.0.0/16",
    "serviceCidr": "10.0.0.0/16",
    "dnsServiceIP": "10.0.0.10",
    "dockerBridgeCidr": "172.17.0.1/16",
```

Azure CNI Overlay: Pod IPs are from Pod-CIDR

```
$
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-14046306-vmss000000   10.224.0.4
aks-nodepool1-14046306-vmss000001   10.224.0.5
aks-nodepool1-14046306-vmss000002   10.224.0.6
$
```

Node pool

Max pods per node
250

```
$
$ kubectl get pods -o wide
NAME                    READY   IP
nginx-8f458dc5b-4ddwt   1/1     192.168.1.143
nginx-8f458dc5b-56qgw   1/1     192.168.2.41
nginx-8f458dc5b-6n799   1/1     192.168.0.97
nginx-8f458dc5b-d6dpx   1/1     192.168.1.174
nginx-8f458dc5b-dcr5w   1/1     192.168.2.158
nginx-8f458dc5b-fm7bj   1/1     192.168.2.190
nginx-8f458dc5b-k26fp   1/1     192.168.0.220
nginx-8f458dc5b-lm9wn   1/1     192.168.1.29
nginx-8f458dc5b-mwfr2   1/1     192.168.0.171
nginx-8f458dc5b-tvxgq   1/1     192.168.2.79
$
```

aks-vnet-51020892 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Networking

API server address	aks-cni-ov-rg-aks-cni-overl-82f6d7-iznmpgpa.hcp.westeurope.azmk8s.io
Network type (plugin)	Azure CNI
Pod CIDR	-
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

```
"networkProfile": {
    "networkPlugin": "azure",
    "networkPluginMode": "overlay",
    "networkDataplane": "azure",
    "loadBalancerSku": "Standard",
    "loadBalancerProfile": {…
    },
    "podCidr": "192.168.0.0/16",
    "serviceCidr": "10.0.0.0/16",
    "dnsServiceIP": "10.0.0.10",
    "dockerBridgeCidr": "172.17.0.1/16",
```

Azure CNI : Service IPs are from Service-CIDR

```
$
$ kubectl get nodes -o wide
NAME                                INTERNAL-IP
aks-nodepool1-14046306-vmss000000   10.224.0.4
aks-nodepool1-14046306-vmss000001   10.224.0.5
aks-nodepool1-14046306-vmss000002   10.224.0.6
$
```

Node pool

Max pods per node
250

```
$
$ kubectl get pods -o wide
NAME                    READY   IP
nginx-8f458dc5b-4ddwt   1/1     192.168.1.143
nginx-8f458dc5b-56qgw   1/1     192.168.2.41
nginx-8f458dc5b-6n799   1/1     192.168.0.97
nginx-8f458dc5b-d6dpx   1/1     192.168.1.174
nginx-8f458dc5b-dcr5w   1/1     192.168.2.158
nginx-8f458dc5b-fm7bj   1/1     192.168.2.190
nginx-8f458dc5b-k26fp   1/1     192.168.0.220
nginx-8f458dc5b-lm9wn   1/1     192.168.1.29
nginx-8f458dc5b-mwfr2   1/1     192.168.0.171
nginx-8f458dc5b-tvxgq   1/1     192.168.2.79
$
```

```
$
$ kubectl get services -A
NAMESPACE     NAME            TYPE        CLUSTER-IP
default       kubernetes      ClusterIP   10.0.0.1
kube-system   kube-dns        ClusterIP   10.0.0.10
kube-system   metrics-server  ClusterIP   10.0.67.197
$
```

aks-vnet-51020892 | Subnets

Virtual network

Search

Settings

- Address space
- Connected devices
- Subnets

+ Subnet + Gateway subnet

Search subnets

Name	IPv4
aks-subnet	10.224.0.0/16

Networking

API server address	aks-cni-ov-rg-aks-cni-overl-82f6d7-iznmpgpa.hcp.westeurope.azmk8s.io
Network type (plugin)	Azure CNI
Pod CIDR	-
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Docker bridge CIDR	172.17.0.1/16

```
"networkProfile": {
    "networkPlugin": "azure",
    "networkPluginMode": "overlay",
    "networkDataplane": "azure",
    "loadBalancerSku": "Standard",
    "loadBalancerProfile": {…
    },
    "podCidr": "192.168.0.0/16",
    "serviceCidr": "10.0.0.0/16",
    "dnsServiceIP": "10.0.0.10",
    "dockerBridgeCidr": "172.17.0.1/16",
```

Kubenet vs Azure CNI Overlay

Area	Azure CNI Overlay	Kubenet
Cluster scale	1000 nodes and 250 pods/node	400 nodes and 250 pods/node
Network configuration	Simple - no additional configuration required for pod networking	Complex - requires route tables and UDRs on cluster subnet for pod networking
Pod connectivity performance	Performance on par with VMs in a VNet	Additional hop adds minor latency
Kubernetes Network Policies	Azure Network Policies, Calico, Cilium	Calico
OS platforms supported	Linux and Windows Server 2022	Linux only

Limitations of Azure CNI Overlay

Azure CNI Overlay has the following limitations:

- You can't use Application Gateway as an Ingress Controller (AGIC) for an overlay cluster.
- Windows Server 2019 node pools are not supported for overlay.

Bring your own CNI

Cilium (OSS & Enterprise)

Calico CNI

Canal

Flannel

Weave

CIDR Subnet, Pod, Service & Docker Bridge overlapping

Subnet, Pod, Service & Docker Bridge should have different CIDR ranges. No overlap.
Pod, Service and Docker Bridge CIDRs could be the same for different clusters.

These IP address ranges should be an address space that isn't in use elsewhere in:

- Your network environment
  - including any on-premises network ranges connected, or plan to connect.
- Your Azure virtual networks using Express Route or a Site-to-Site VPN connection.

AKS clusters may not use:

- 169.254.0.0/16
- 172.30.0.0/16
- 172.31.0.0/16
- 192.0.2.0/24

for the :

- Service-CIDR
- Pod-CIDR
- Cluster virtual network

Can't be updated after cluster creation.

```
az aks create \
    --resource-group myResourceGroup \
    --name myAKSCluster \
    --network-plugin kubenet \
    --service-cidr 10.0.0.0/16 \
    --dns-service-ip 10.0.0.10 \
    --pod-cidr 10.244.0.0/16 \
    --docker-bridge-address 172.17.0.1/16 \
    --vnet-subnet-id $SUBNET_ID
```