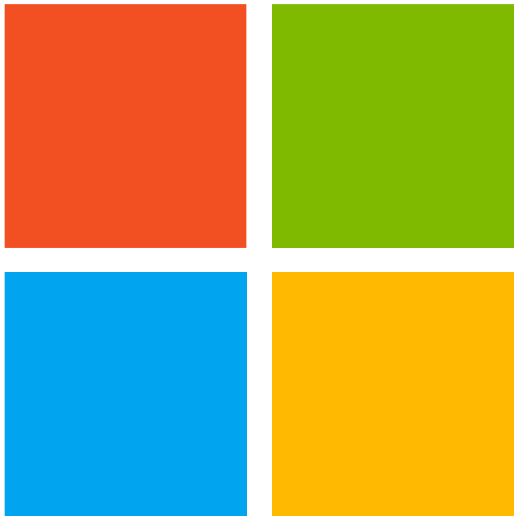






# CLOUDMATE

Azure Expert Group



# Microsoft

# AKS(Azure Kubernetes Service) 네트워크 비교 및 클러스터 구축 시 용량 산정 이야기

<김영대>

# Who Am I?

SPTek (에스피테크놀러지)

- ✓ 김영대 / Zerobig  
(*Azure Solution Architect / DevOps Expert / Cloud 컨설턴트*)
  - Azure Solutions Architect Expert – 2019.10~
  - Azure DevOps Engineer Expert – 2019.12~
- ✓ Kubernetes 문서 한글화팀 공식 Member (2018.12)
- ✓ 슬기로운 Azure 생활 공동 운영자 (2020.1~ )
- ✓ 제로빅 블로그 <https://zerobig-k8s.tistory.com/>
- ✓ 제로빅 페이스북 <https://www.facebook.com/addicted2k8s>



# Agenda

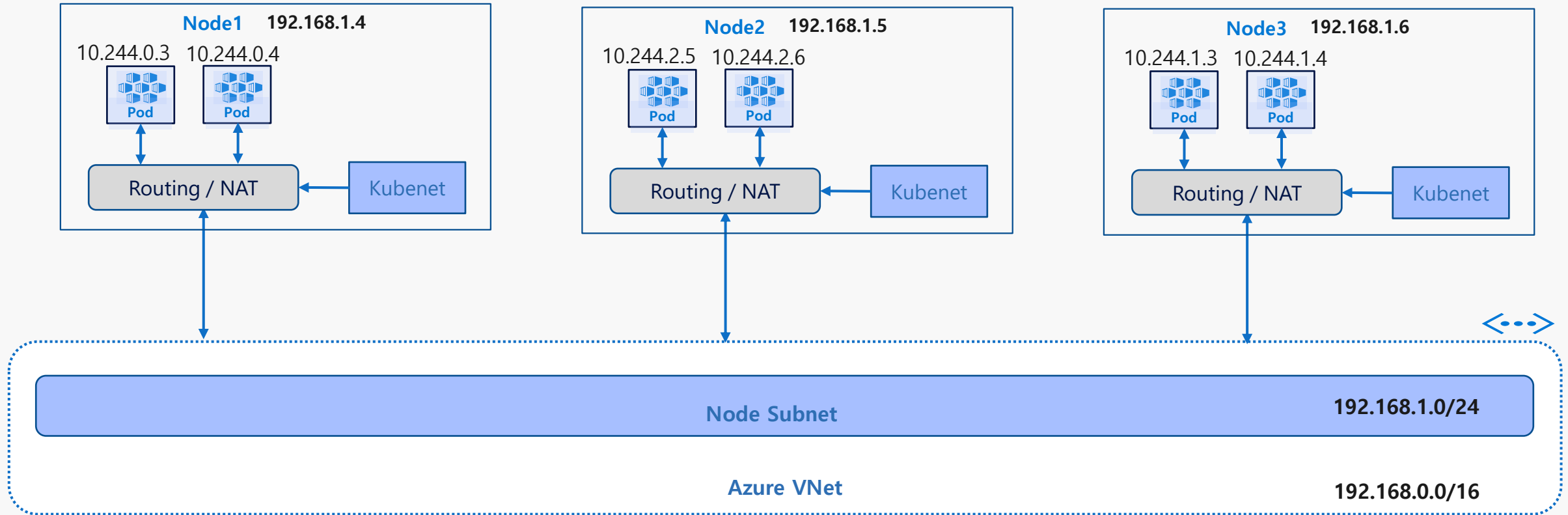
AKS 네트워크 모델 비교

실습 : 아웃바운드 트래픽을 통한 모델 별 비교

AKS 클러스터 구축 시 노드/파드 용량 산정 시  
고려 사항

# AKS 네트워크 모델 비교

# Kubenet : Basic



- 클러스터 서브넷 별도의 Pod CIDR에서 파드 IP 할당 → IP 자원 보존
- 노드 당 110개 파드 IP 할당 지원
- 400개 Route/1개 UDR 지원
- 하나의 AKS 클러스터에 400개 이상의 노드 못 가짐
- AKS Virtual Node, Azure Network Policy 미지원 → Calico Network Policy 지원

# Kubernetes : Basic (계속)

- 노드는 Virtual Network Subnet 주소 공간으로부터 IP 할당

```
zerobig_devops@Azure:~$ kubectl get node -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP
aks-nodepool1-27931838-vmss000000	Ready	agent	42m	v1.18.14	192.168.1.4
aks-nodepool1-27931838-vmss000001	Ready	agent	42m	v1.18.14	192.168.1.5
aks-nodepool1-27931838-vmss000002	Ready	agent	42m	v1.18.14	192.168.1.6

- 모든 파드는 클러스터 생성 단계에 정의된 Pod CIDR(Overlay 네트워크) 주소 공간으로부터 IP 할당

```
zerobig_devops@Azure:~$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP
ds-busybox-2blgs	1/1	Running	0	34m	10.244.0.3
ds-busybox-hxwqp	1/1	Running	0	34m	10.244.1.3
ds-busybox-sqkt8	1/1	Running	0	34m	10.244.2.5

## Network profile

Type (plugin)	Kubenet
Pod CIDR	10.244.0.0/16



# Kubenet : Basic (계속)

- 파드간 통신은 IP Forwarding 및 UDR을 통해 이루어짐  
→ AKS 생성 시 (없으면) 자동으로 만들어 줌

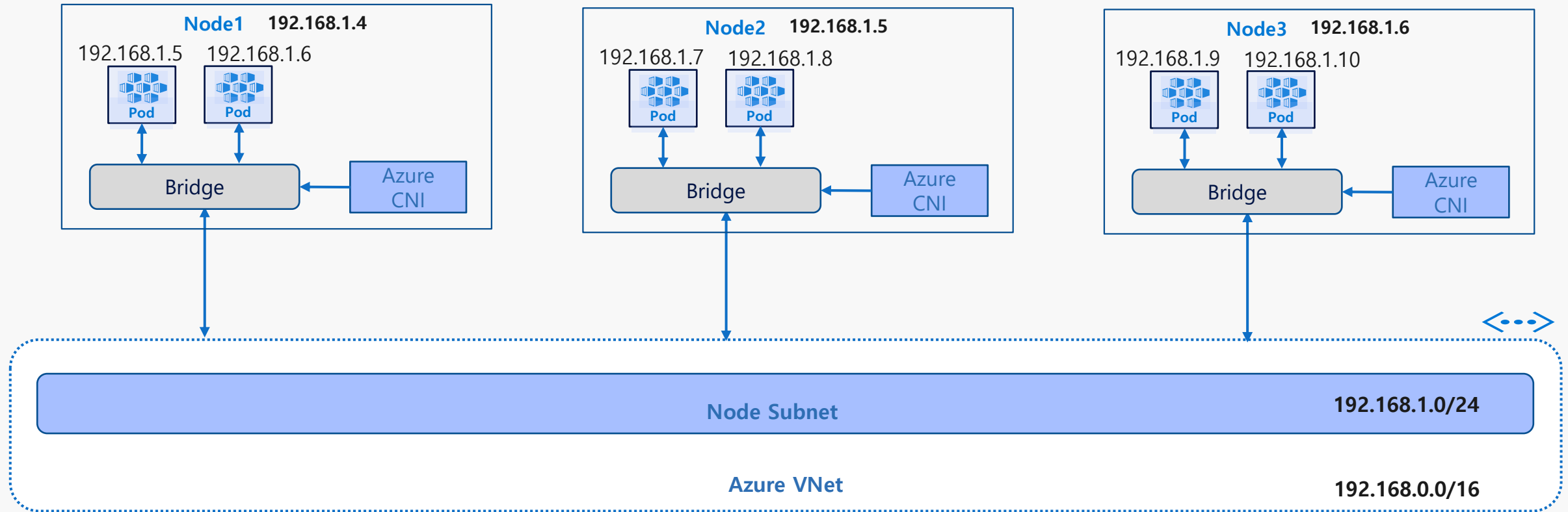
## Routes

 Search routes

Name	↑↓	Address prefix	↑↓	Next hop type	↑↓	Next hop IP address
aks-nodepool1-27931838-vmss000000		10.244.1.0/24		Virtual appliance		192.168.1.4
aks-nodepool1-27931838-vmss000001		10.244.2.0/24		Virtual appliance		192.168.1.5
aks-nodepool1-27931838-vmss000002		10.244.0.0/24		Virtual appliance		192.168.1.6

- 라우팅 규칙의 증가에 따른 네트워크 관리 복잡성 증가 이슈

# Azure CNI : Advanced



- Pod/노드 Node서브넷으로부터 IP 할당. 직접 통신
- Pod CIDR 불필요
- 라우팅 테이블 불필요

<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/> <a href="#">aks-agentpool-15046883-nsg</a>	Network security group	Korea Central
<input type="checkbox"/> <a href="#">aks-nodepool1-15046883-vmss</a>	Virtual machine scale set	Korea Central
<input type="checkbox"/> <a href="#">dcd88fef-f4ff-4a58-8929-eae427584c4e</a>	Public IP address	Korea Central
<input type="checkbox"/> <a href="#">kubernetes</a>	Load balancer	Korea Central

# Azure CNI : Advanced (계속)

- 노드/파드 모두 Virtual Network Subnet 주소 공간으로부터 IP 할당 – No Pod CIDR

## Network profile

Type (plugin)	Azure CNI
Service CIDR	10.0.0.0/16

```
zerobig_devops@Azure:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP
aks-nodepool1-15046883-vmss000000	Ready	agent	3m50s	v1.18.14	192.168.2.4
aks-nodepool1-15046883-vmss000001	Ready	agent	3m53s	v1.18.14	192.168.2.35
aks-nodepool1-15046883-vmss000002	Ready	agent	3m53s	v1.18.14	192.168.2.66

```
zerobig_devops@Azure:~$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP
ds-busybox-6rwj8	1/1	Running	0	23s	192.168.2.80
ds-busybox-94qs7	1/1	Running	0	23s	192.168.2.19
ds-busybox-hwr8d	1/1	Running	0	23s	192.168.2.40

# Azure CNI : Advanced (계속)

- Service, Pod 또는 Cluster VNET 주소 범위 사용 불가 대역

**169.254.0.0/16, 172.30.0.0/16, 172.31.0.0/16 또는  
192.0.2.0/24**

- 클러스터 생성 시 각 노드에 대한 파드용 IP주소 사전 할당

**노드 당 maxPods 기본값 = 30**

(배포 시점 또는 노드 풀 생성시 10~250 중 가변가능)

# Kubenet Vs Azure CNI

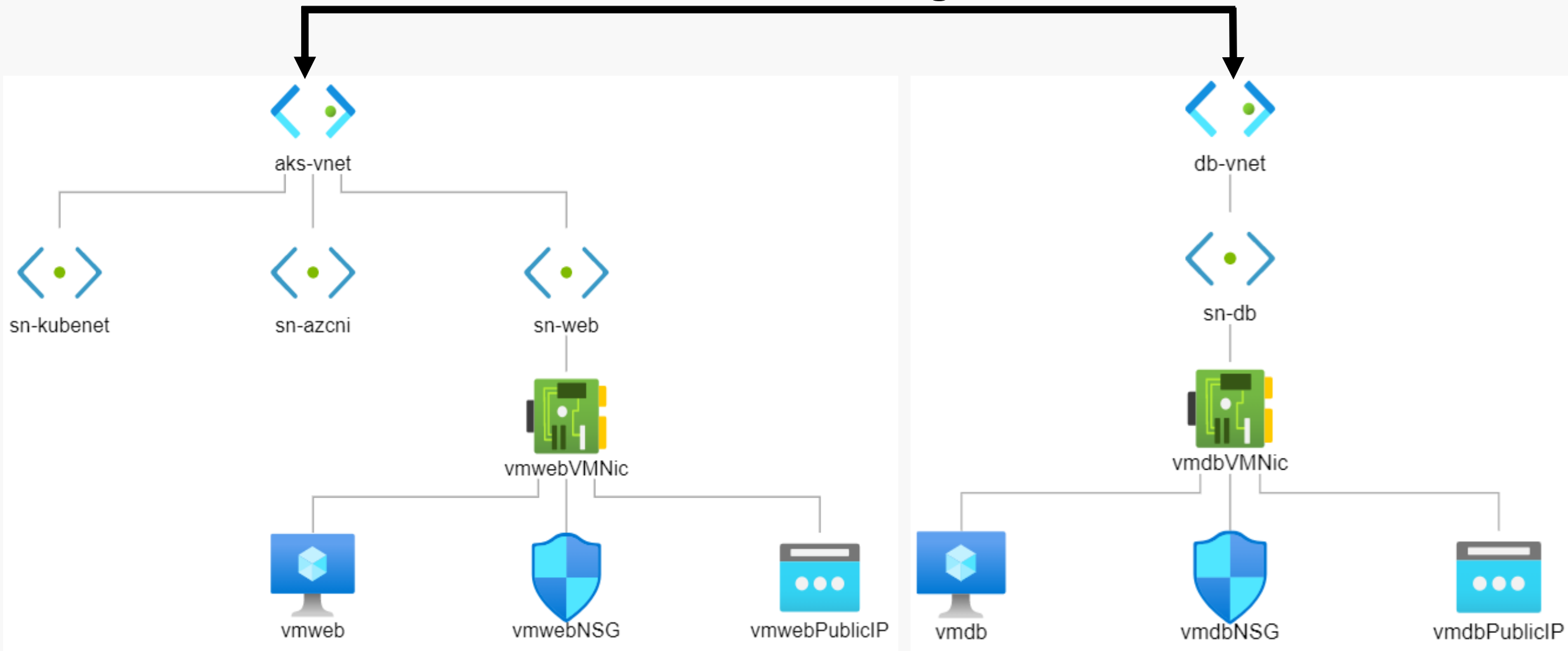
Kubenet	Azure CNI
VNET IP 주소 공간 절약	노드/파드 동일 VNET내 지정된 Subnet 대역에서 IP 할당 (기본 30개 사전 할당 구성) → <b>IP 부족 이슈</b>
대부분 클러스터 내부 파드 통신 시나리오	대부분 클러스터 외부 파드 통신 시나리오
가상 노드, Azure 네트워크 정책, Windows 컨테이너 지원 등 고급 기능 불필요 시	가상 노드, Azure 네트워크 정책, Windows 컨테이너 지원 등 고급 기능 필요 시
노드 당 110개 파드 지원	노드 당 250개 파드 지원
네트워크 트래픽 NAT 및 추가 라우팅 요구 사항으로 인해 관리 복잡성 증가 → 성능 저하 초래	VNET 엔드포인트를 사용하여 다른 Azure 리소스와 안전하고 프라이빗하게 연결  전체 파드 트래픽 높고 온 프레미스 등 연결 필요 시 → <b>더 낮은 성능</b>

- ✓ 대부분의 새로운 AKS 기능은 처음에는 Azure CNI와 함께 개발되고
- ✓ 기술적으로 호환될 때 Kubenet에 맞게 조정

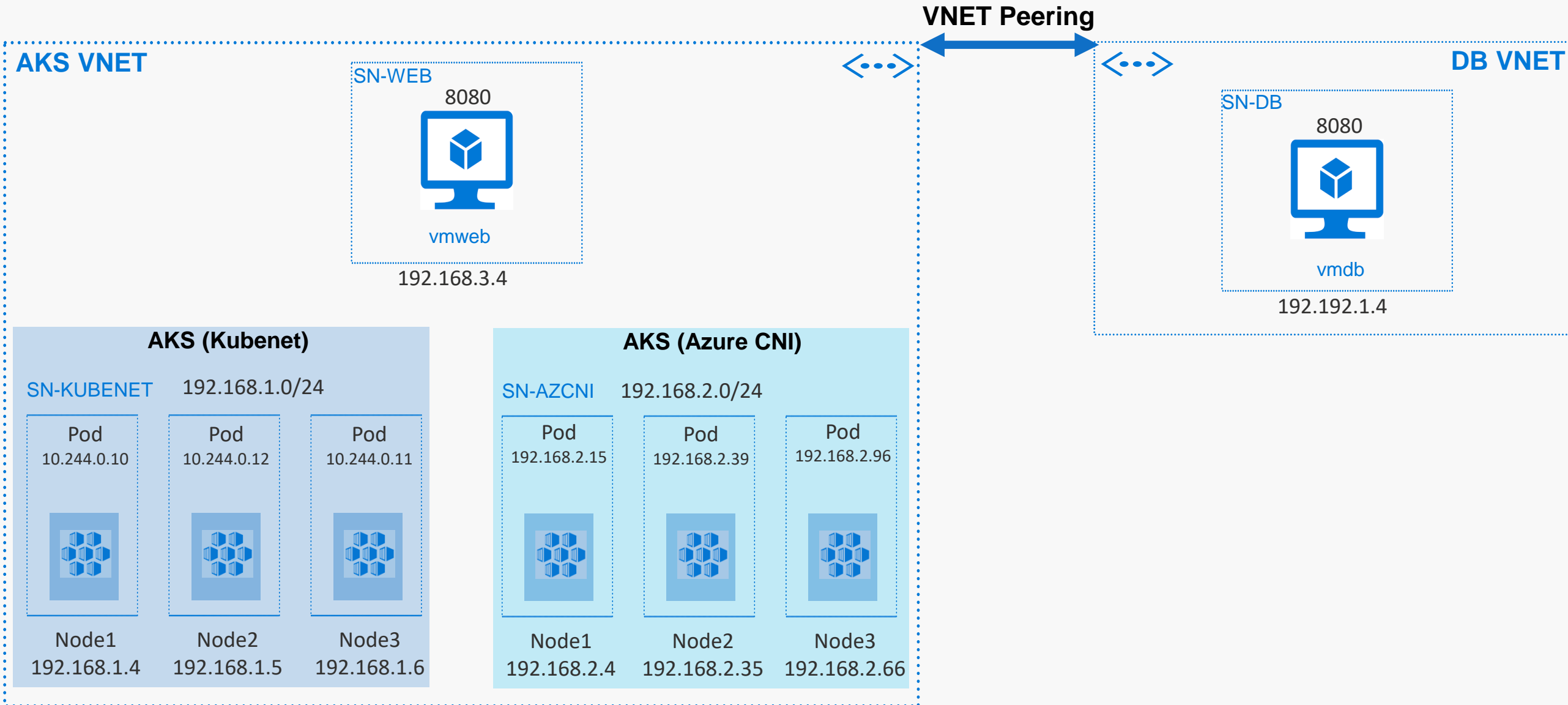
# 실습 : 네트워크 모델 별 비교

# 실습 환경 구성

## VNET Peering



# 실습 환경 구성





# AKS 클러스터 구축 시 노드/파드 용량 산정 시 고려 사항

# AKS 클러스터 용량 산정

## 고려 사항

- ✓ 네트워크 모델
- ✓ 서브넷 크기
- ✓ 클러스터 크기
- ✓ 파드 용량

# AKS 클러스터 용량 산정

## AKS 서브넷 크기

기본 요구 총 IP 주소 : **36개**

서브넷 크기 : **/26**

- 각 서브넷 내에서 Azure VNet에서 예약한 **5 개의 IP**

(<https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-faq>)

- ✓ x.x.x.0 : 네트워크 주소
- ✓ x.x.x.1 : Azure에서 기본 게이트웨이 용으로 예약 됨
- ✓ x.x.x.2, x.x.x.3 : Azure DNS IP를 VNet 공간에 매핑하기 위해 Azure에서 예약
- ✓ x.x.x.255 : 네트워크 브로드 캐스트 주소

- 클러스터에 있는 1 노드의 네트워크 인터페이스에 할당된 **1 개의 IP**

- ✓ x.x.x.x.4

- 노드 당 최대 파드에 해당하는 Azure CNI에서 예약한 **30 개의 IP**

- ✓ x.x.x.x.5 – x.x.x.35 (노드 1 개)

# AKS 클러스터 용량 산정

## AKS 서브넷 크기 (업그레이드 케이스 고려)

요구되는 총 IP 주소: **67개**,

서브넷 크기: **/25**

- 기본적으로 요구되는 총 **36개의 IP**
- 클러스터에 있는 1 노드의 네트워크 인터페이스에 할당된 **1 개의 IP**  
✓ x.x.x.x.4
- 노드 당 최대 파드에 해당하는 Azure CNI에서 예약한 **30 개의 IP**  
✓ x.x.x.x.5 – x.x.x.35 (노드 1 개)

# AKS 클러스터 용량 산정

## AKS 클러스터 크기

노드 IP :  $n + 1$

파드 IP :  $(n+1)*30$

총 소요 IP :  $(n+1) + ((n+1)*30) + 5$

- ✓  $n$  : AKS 클러스터 내 총 노드 수
- ✓ 1 : AKS 노드 풀 업그레이드 중 추가 노드

- ✓  $x.x.x.0$  : 네트워크 주소
- ✓  $x.x.x.1$  : Azure에서 기본 게이트웨이 용으로 예약 됨
- ✓  $x.x.x.2, x.x.x.3$  : Azure DNS IP를 VNet 공간에 매핑하기 위해 Azure에서 예약
- ✓  $x.x.x.255$  : 네트워크 브로드 캐스트 주소

Ex) 3개 노드 기준

$$5 + (3+1) + ((3+1)*30) = 129개 > /25 (128개)$$

→ 노드 당 최대 파드 수를 29 적용

$$5 + (3+1) + ((3+1)*29) = 125개 > /25 (128개)$$

# AKS 클러스터 용량 산정

## AKS 파드 용량 (기본 30Pod, 3 Node 기준)

SYSTEM 파드 수 : 14개

따라서

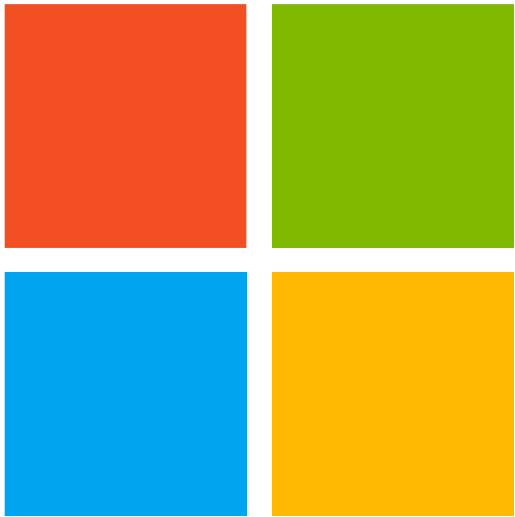
$$3 * 30 - 14 = 76 \text{ 개}$$

```
zerobig_devops@Azure:~$ kubectl get pods -o wide -A
NAMESPACE      NAME                                                    READY   STATUS    RESTARTS   AGE   IP
kube-system    azure-cni-networkmonitor-5xvdj                        1/1     Running   0           7m43s 192.168.2.4
kube-system    azure-cni-networkmonitor-8f9n5                        1/1     Running   0           7m46s 192.168.2.66
kube-system    azure-cni-networkmonitor-zl4cj                        1/1     Running   0           7m46s 192.168.2.35
kube-system    azure-ip-masq-agent-6zxxg                             1/1     Running   0           7m46s 192.168.2.66
kube-system    azure-ip-masq-agent-jzt5x                             1/1     Running   0           7m46s 192.168.2.35
kube-system    azure-ip-masq-agent-xnlsv                             1/1     Running   0           7m43s 192.168.2.4
kube-system    coredns-748cdb7bf4-5p7xg                             1/1     Running   0           7m23s 192.168.2.22
kube-system    coredns-748cdb7bf4-zs6zl                             1/1     Running   0           9m21s 192.168.2.89
kube-system    coredns-autoscaler-868b684fd4-l7rbd                  1/1     Running   0           9m19s 192.168.2.14
kube-system    kube-proxy-dnc9d                                       1/1     Running   0           7m46s 192.168.2.35
kube-system    kube-proxy-g2vr6                                       1/1     Running   0           7m43s 192.168.2.4
kube-system    kube-proxy-grzn5                                       1/1     Running   0           7m46s 192.168.2.66
kube-system    metrics-server-58fdc875d5-wpntf                      1/1     Running   0           9m21s 192.168.2.13
kube-system    tunnelfront-7587c547b6-hlcv                             1/1     Running   0           9m18s 192.168.2.60
zerobig_devops@Azure:~$ kubectl get pods -o wide -A | wc -l
15
```



**CLOUDMATE**

Azure Expert Group



Microsoft

