AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

NET208

Integrate Amazon EKS with your networking pattern

Theo Salvo
Senior Solutions Architect – Containers
AWS

Tom Adamski
Principal Solutions Architect – Networking
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Networking foundations recap

Amazon EKS cluster connectivity

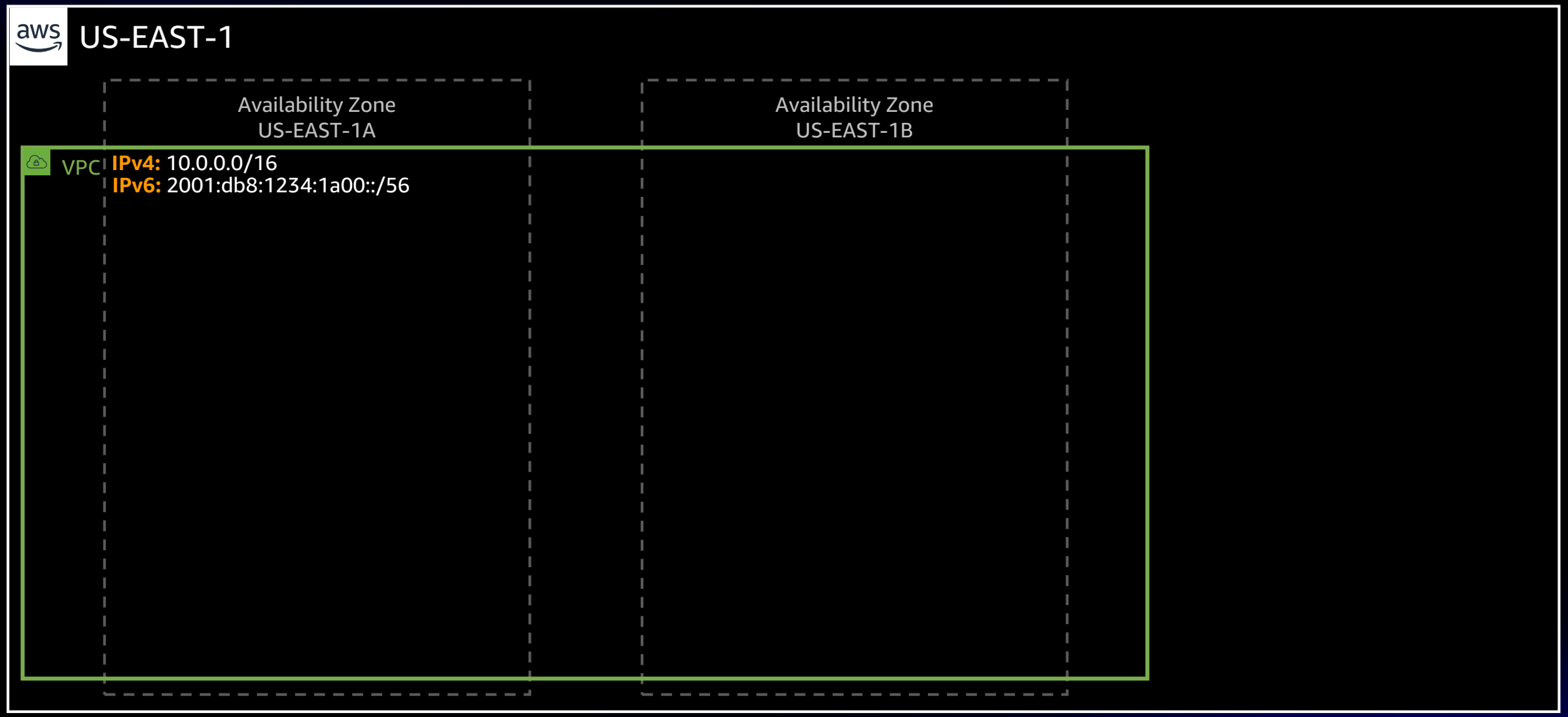
Pod networking

Exposing services

Service discovery and service mesh

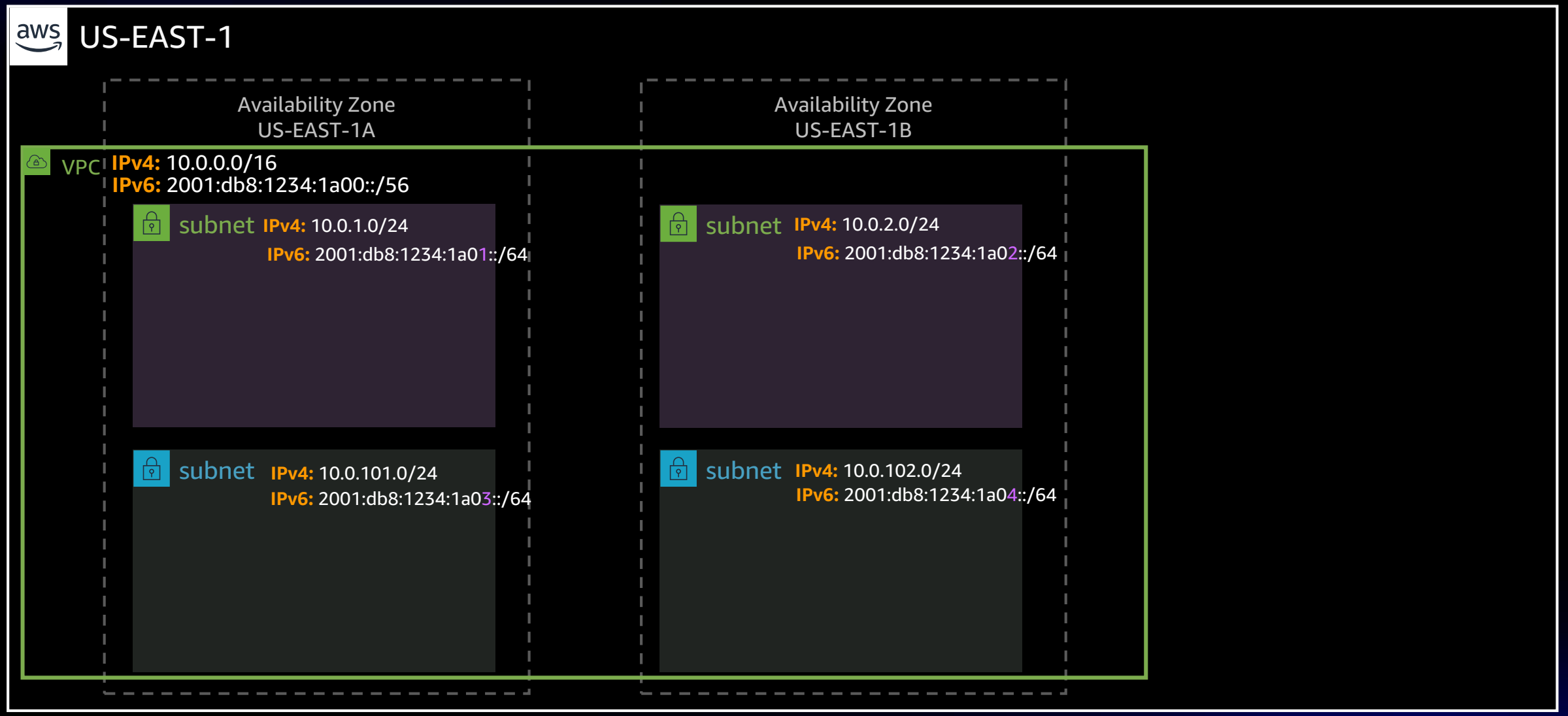
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – IP range (CIDR)



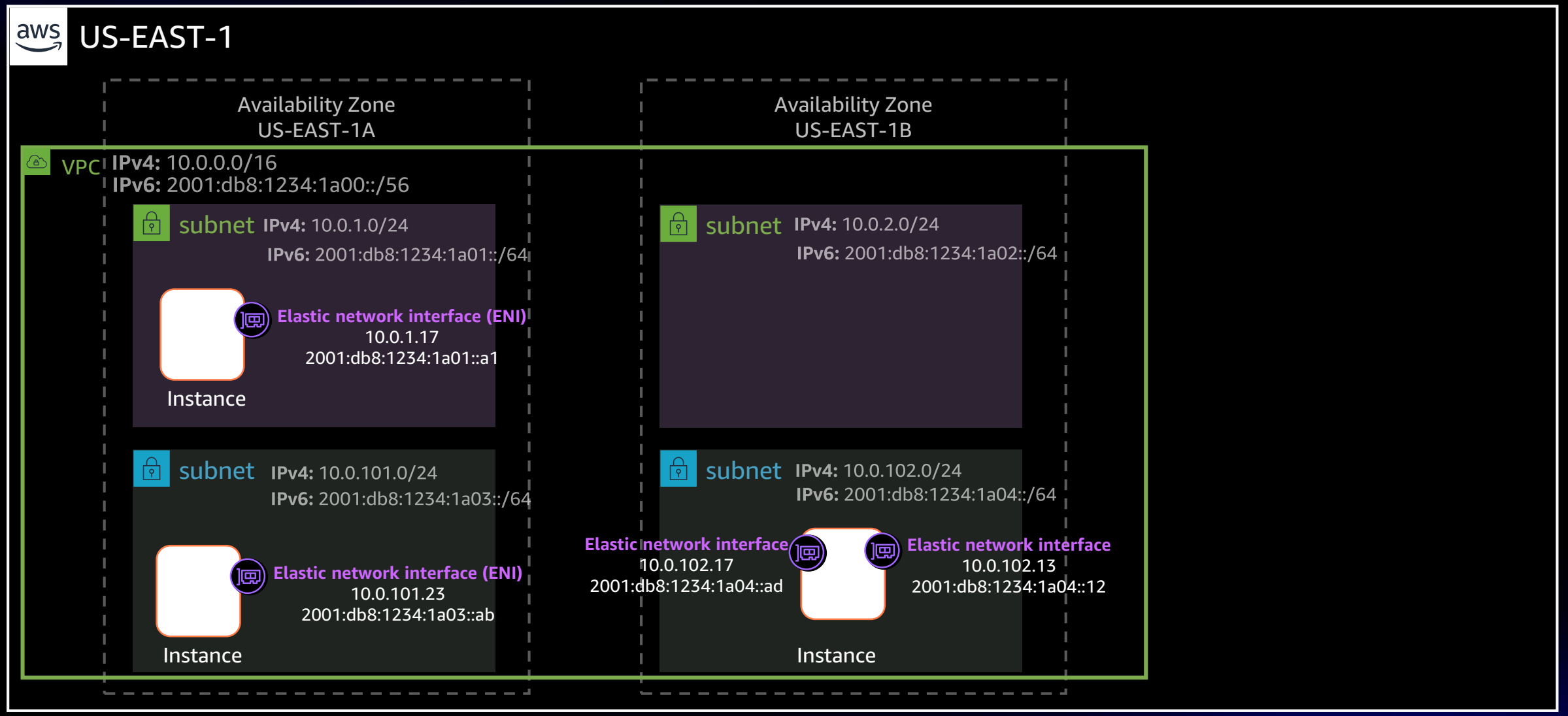
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Subnets



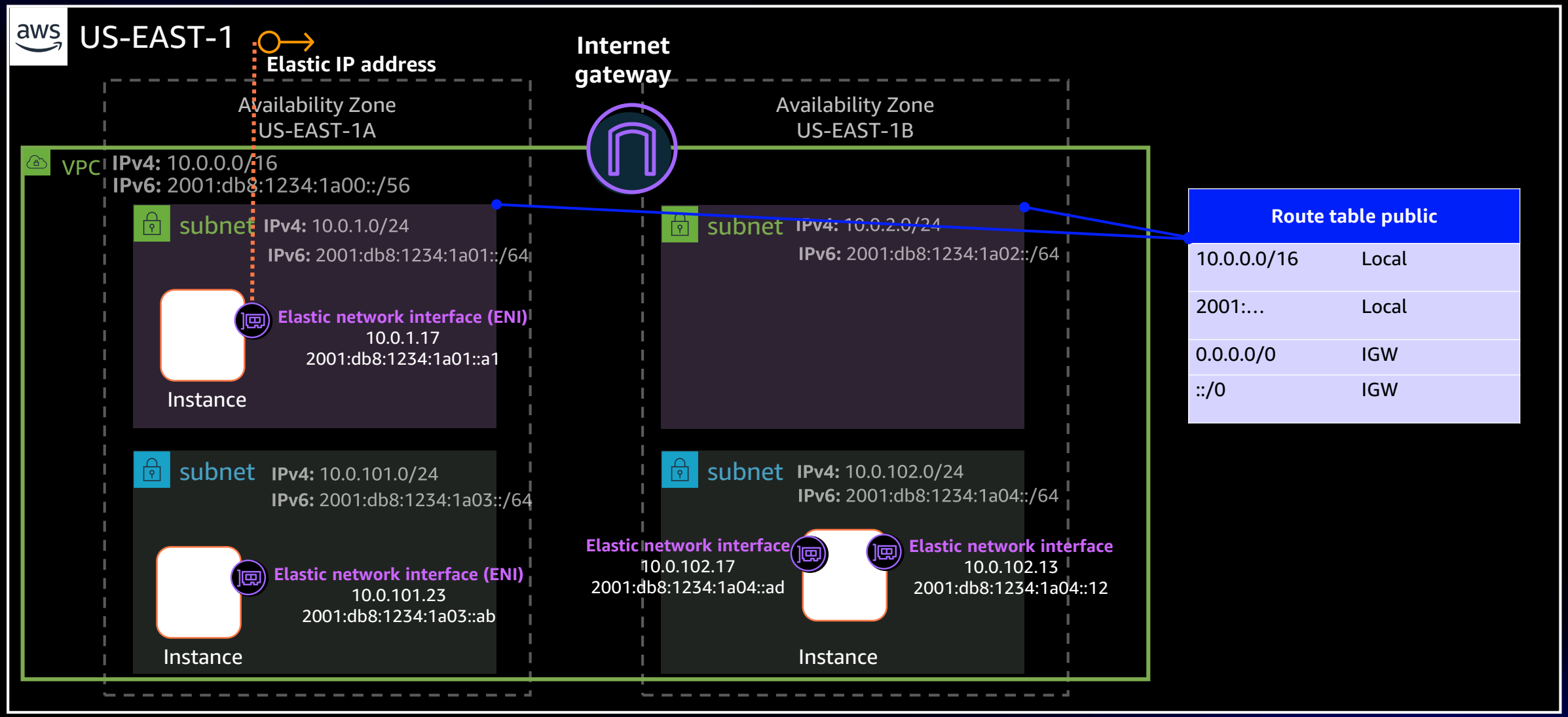
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Interfaces



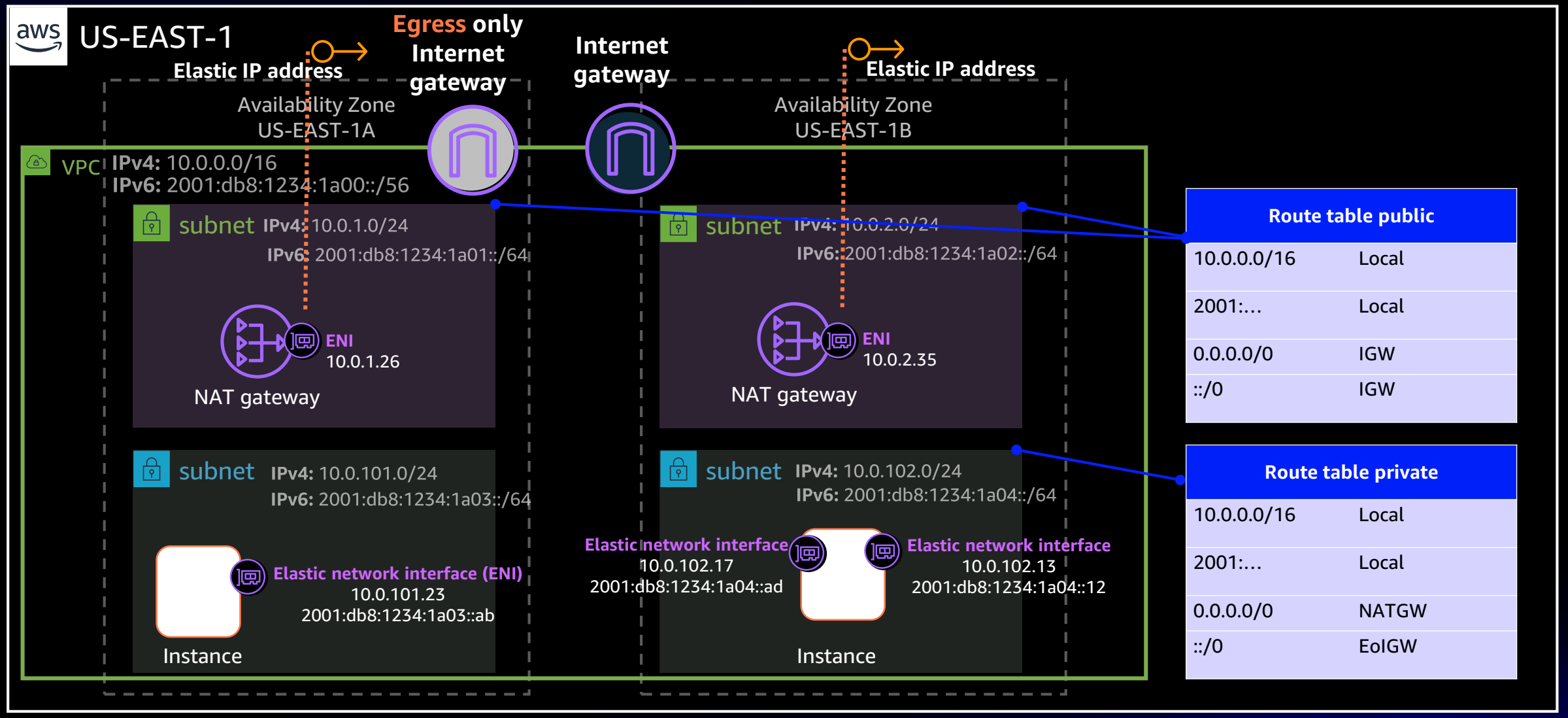
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Connectivity to Internet



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Connectivity to Internet

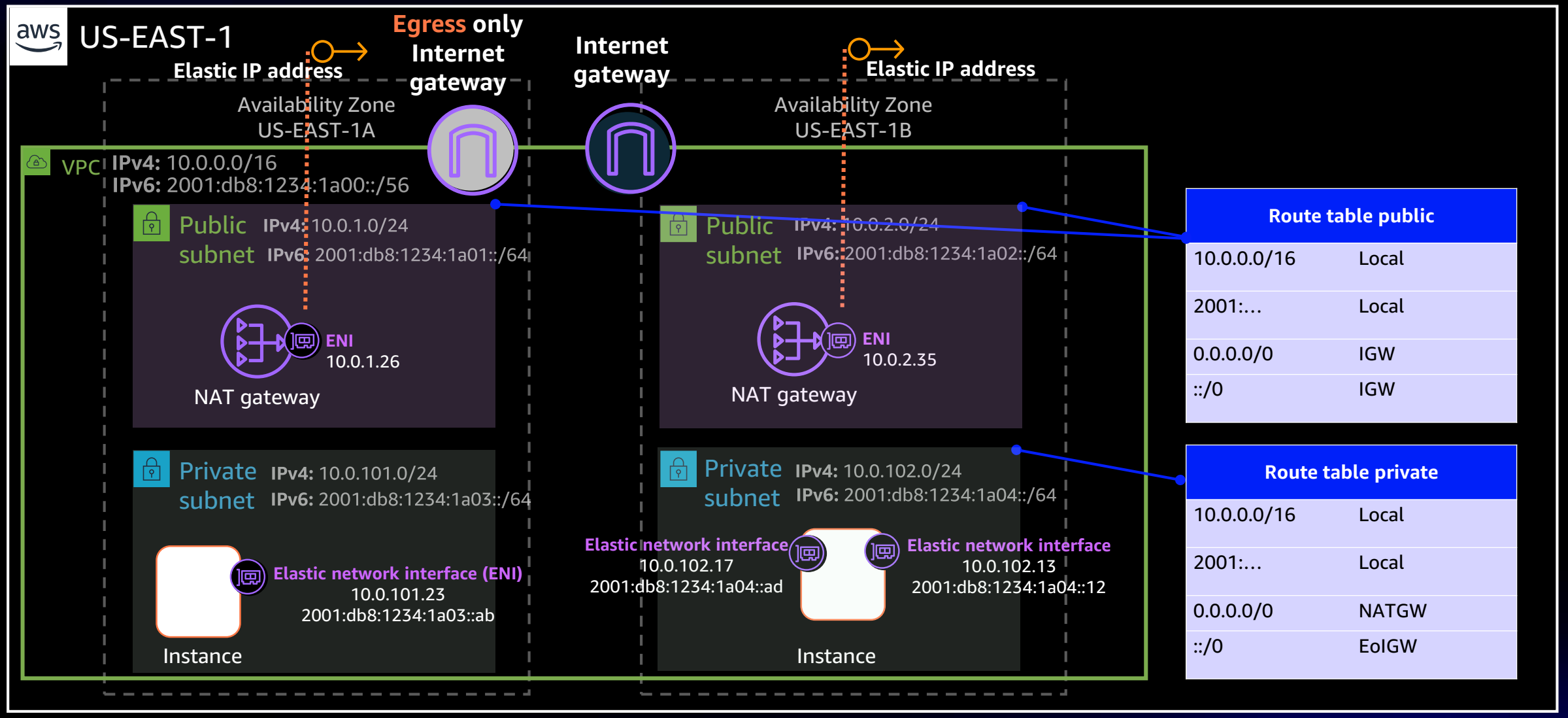


Route table public	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	IGW
::/0	IGW

Route table private	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	NATGW
::/0	EoIGW

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Connectivity to Internet

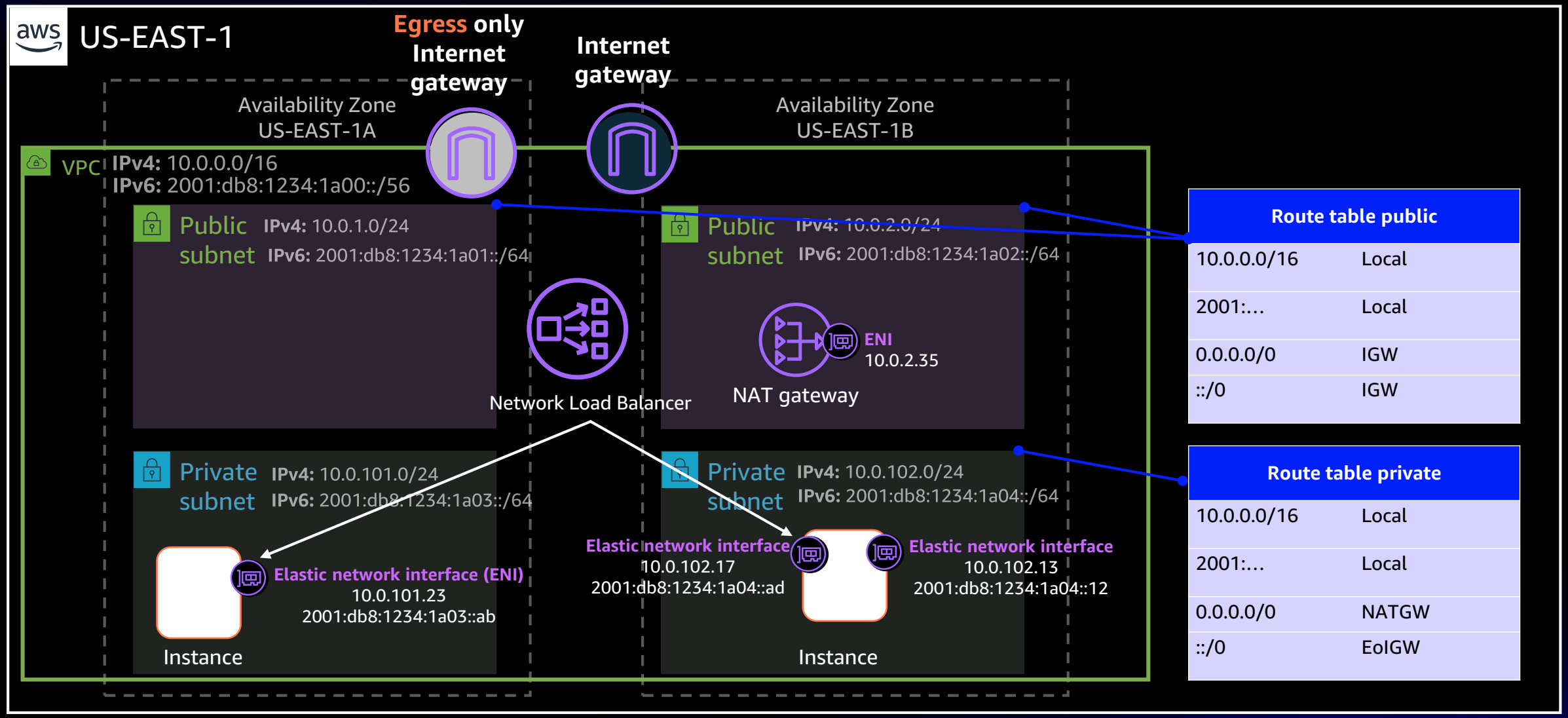


Route table public	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	IGW
::/0	IGW

Route table private	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	NATGW
::/0	EoIGW

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Connectivity to Internet

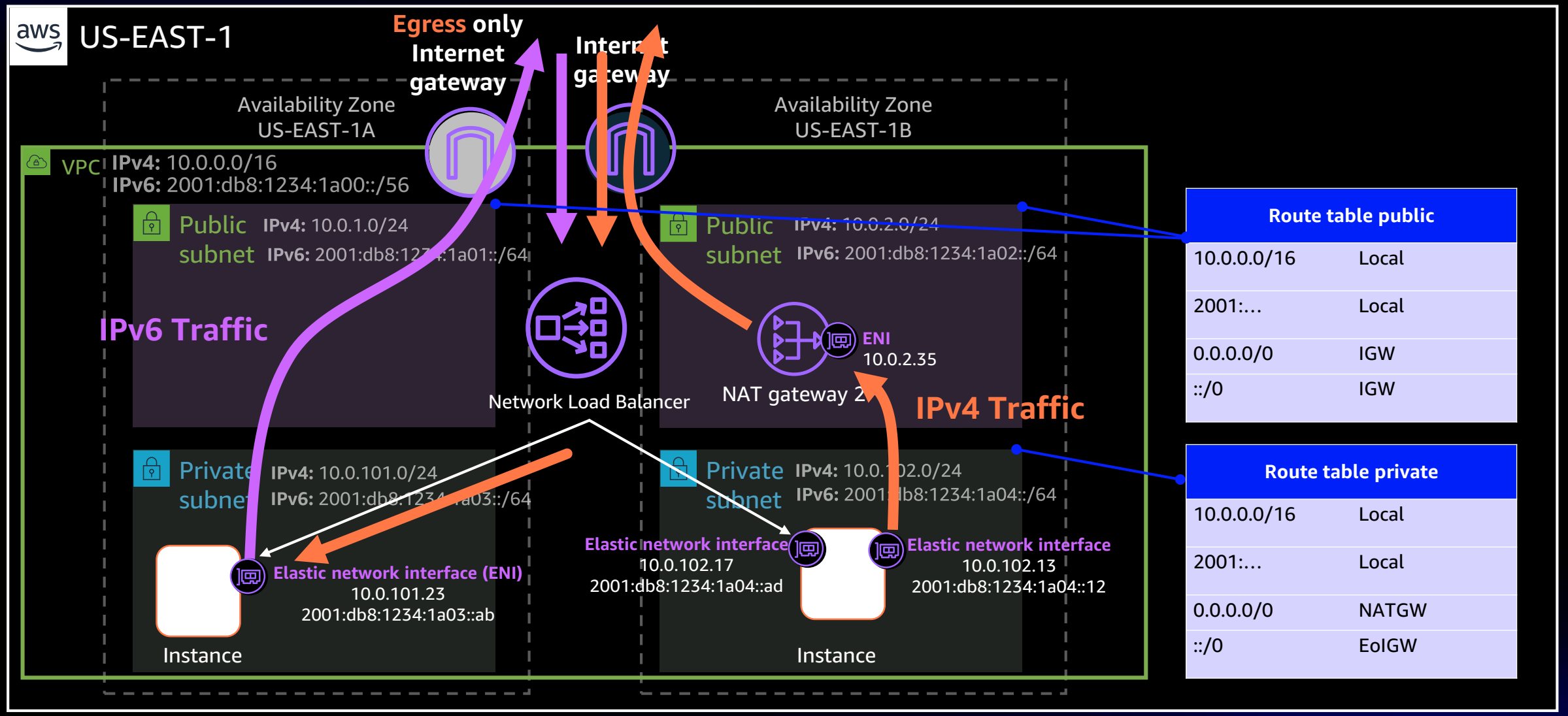


Route table public	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	IGW
::/0	IGW

Route table private	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	NATGW
::/0	EoIGW

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC – Connectivity to Internet

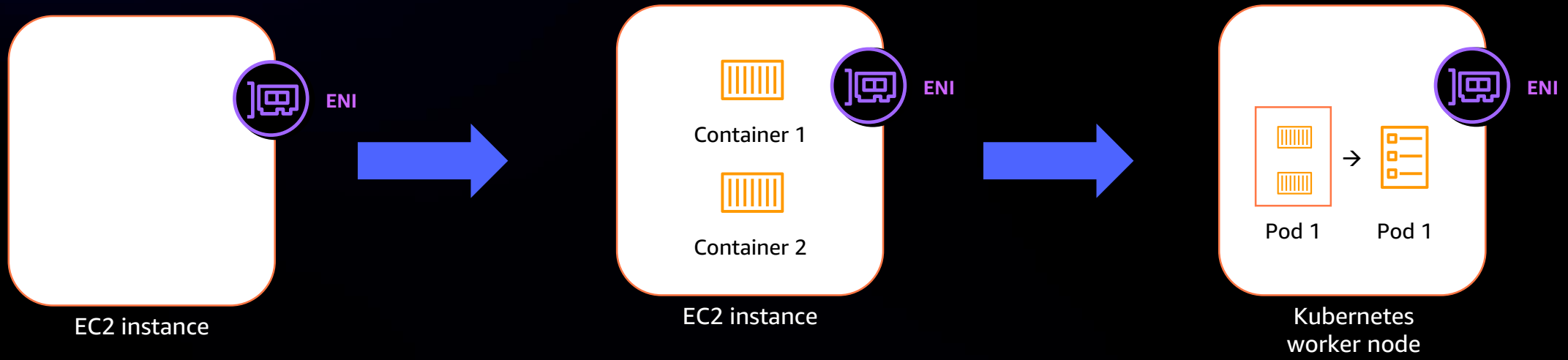


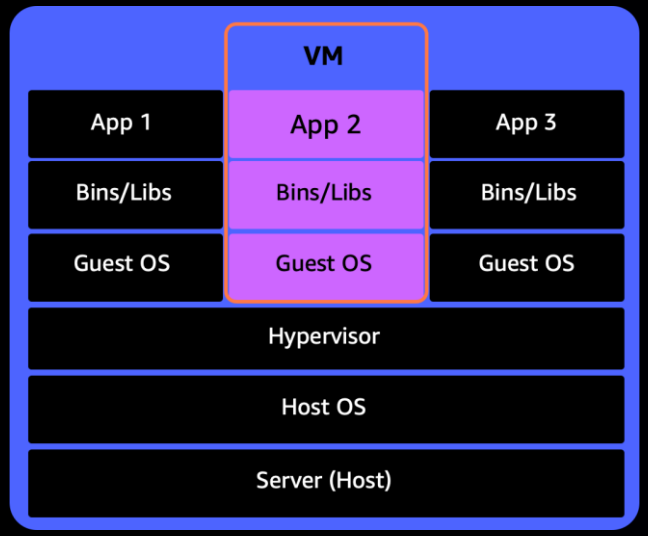
Route table public	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	IGW
::/0	IGW

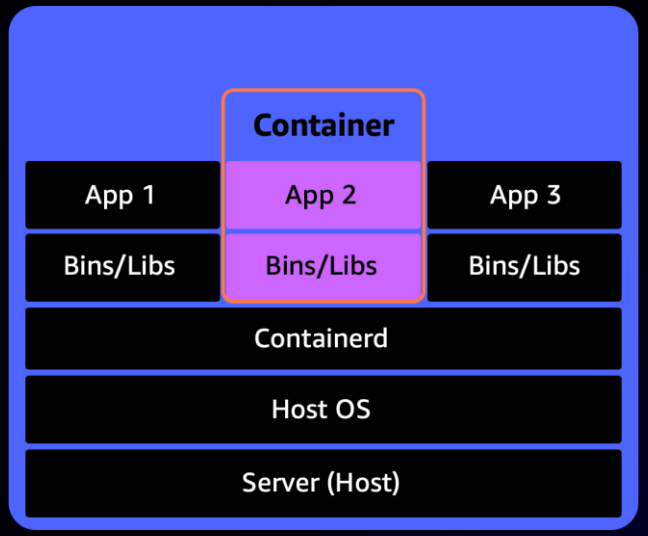
Route table private	
10.0.0.0/16	Local
2001:...	Local
0.0.0.0/0	NATGW
::/0	EoIGW

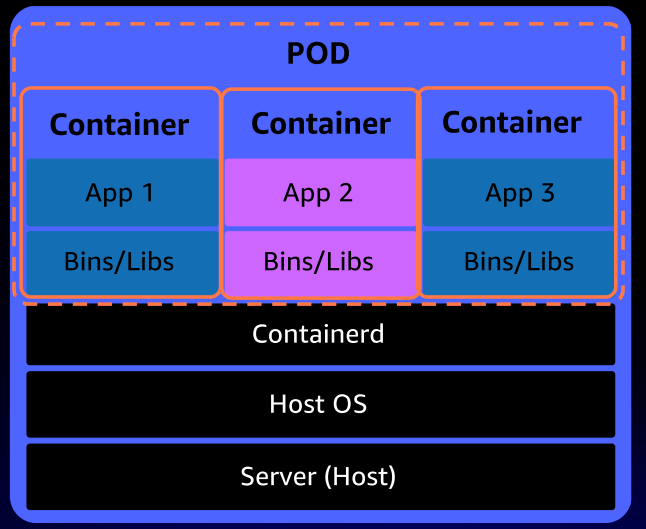
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Moving to containers



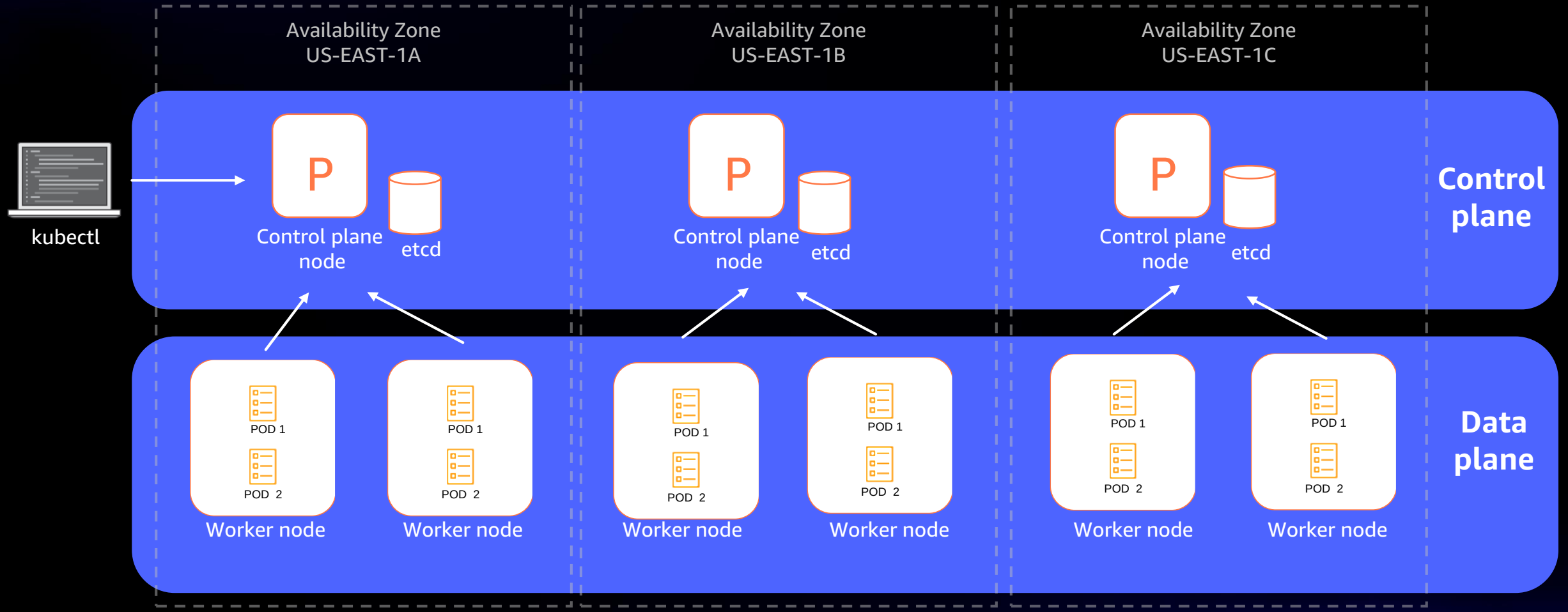






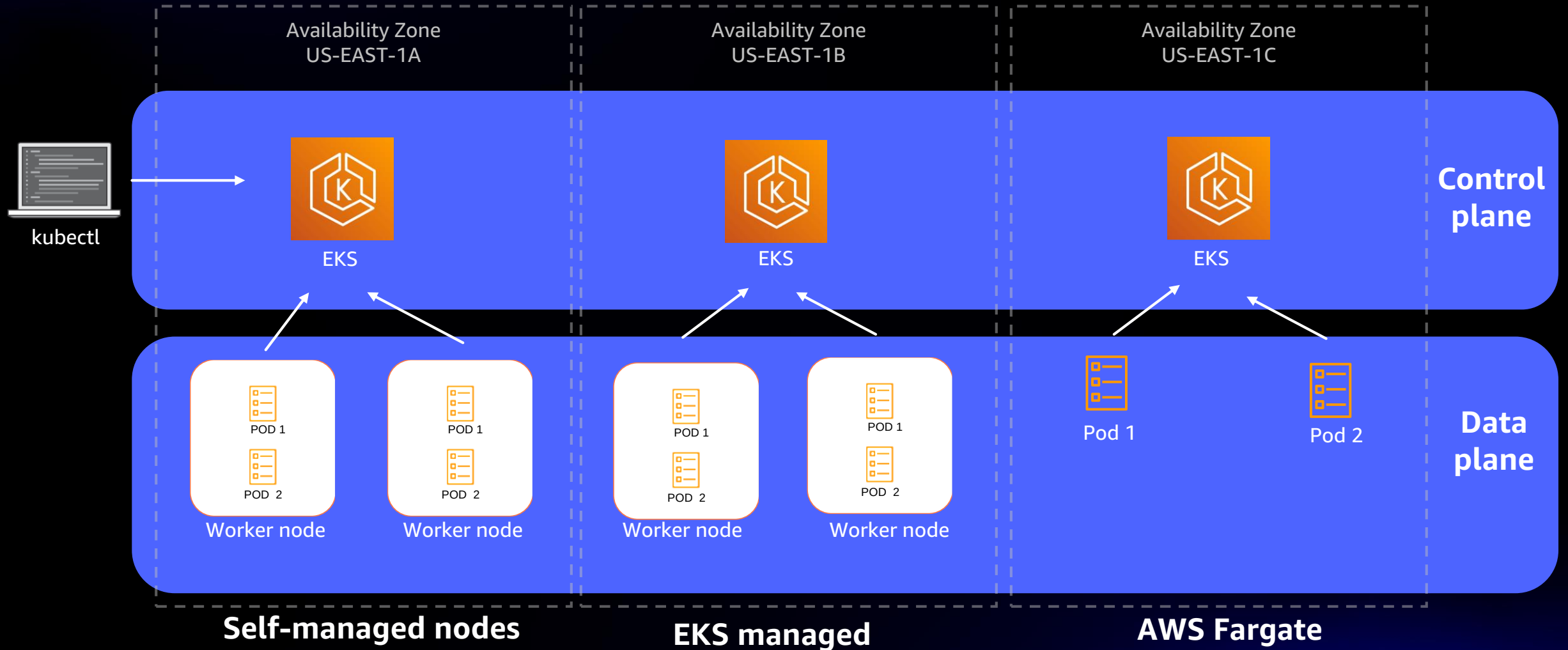
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes orchestration



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS

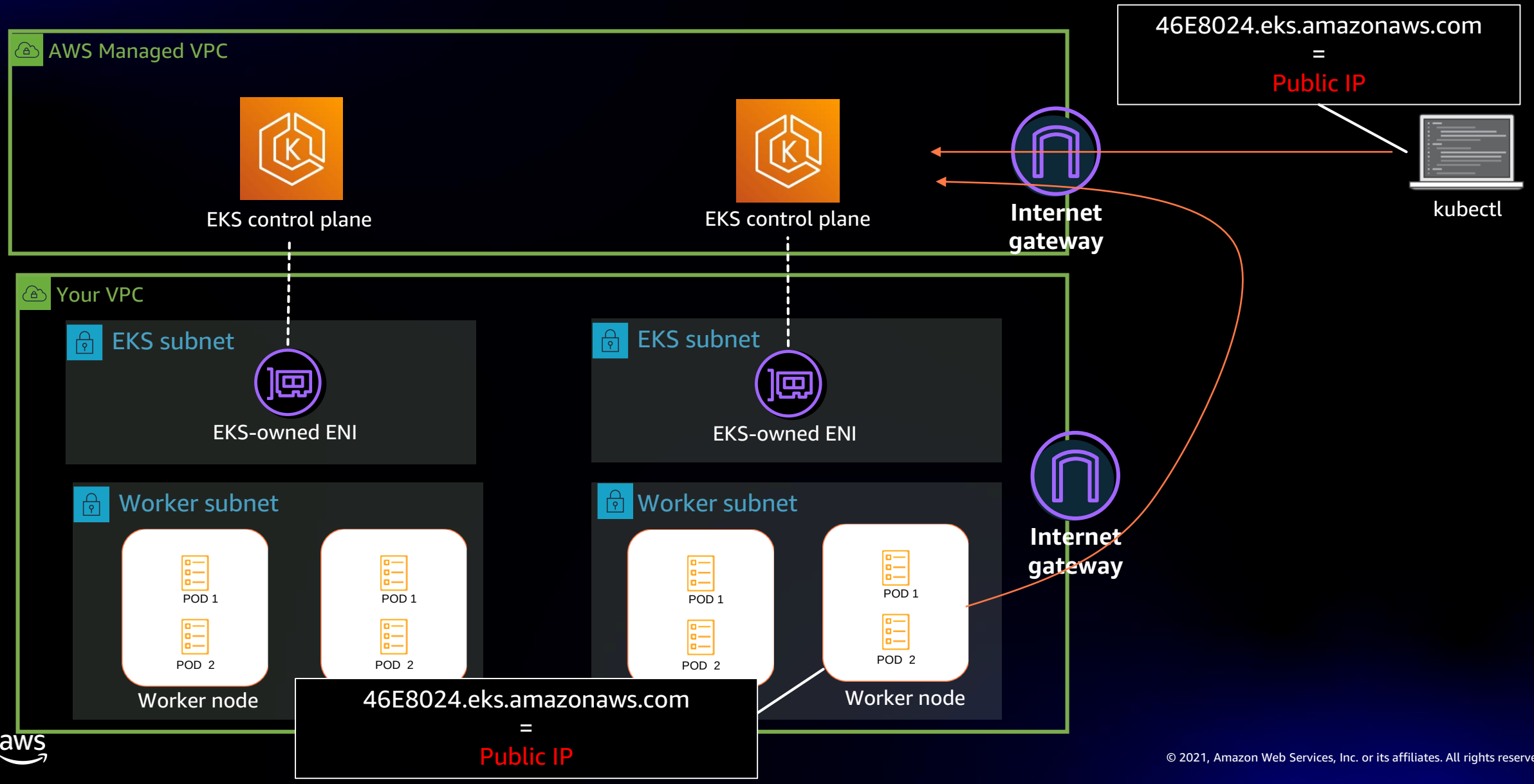


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS cluster endpoint access

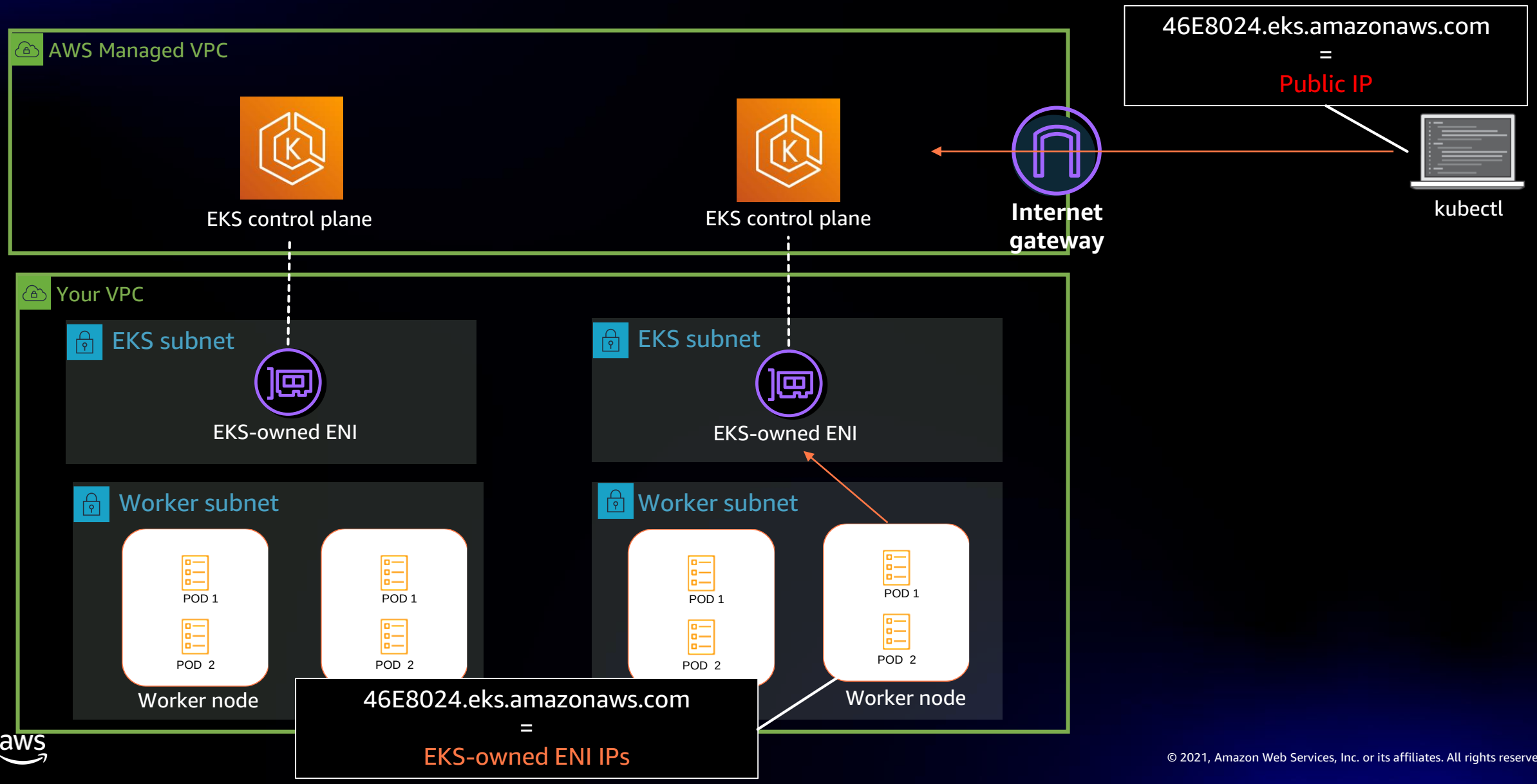
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS cluster endpoint access – Public



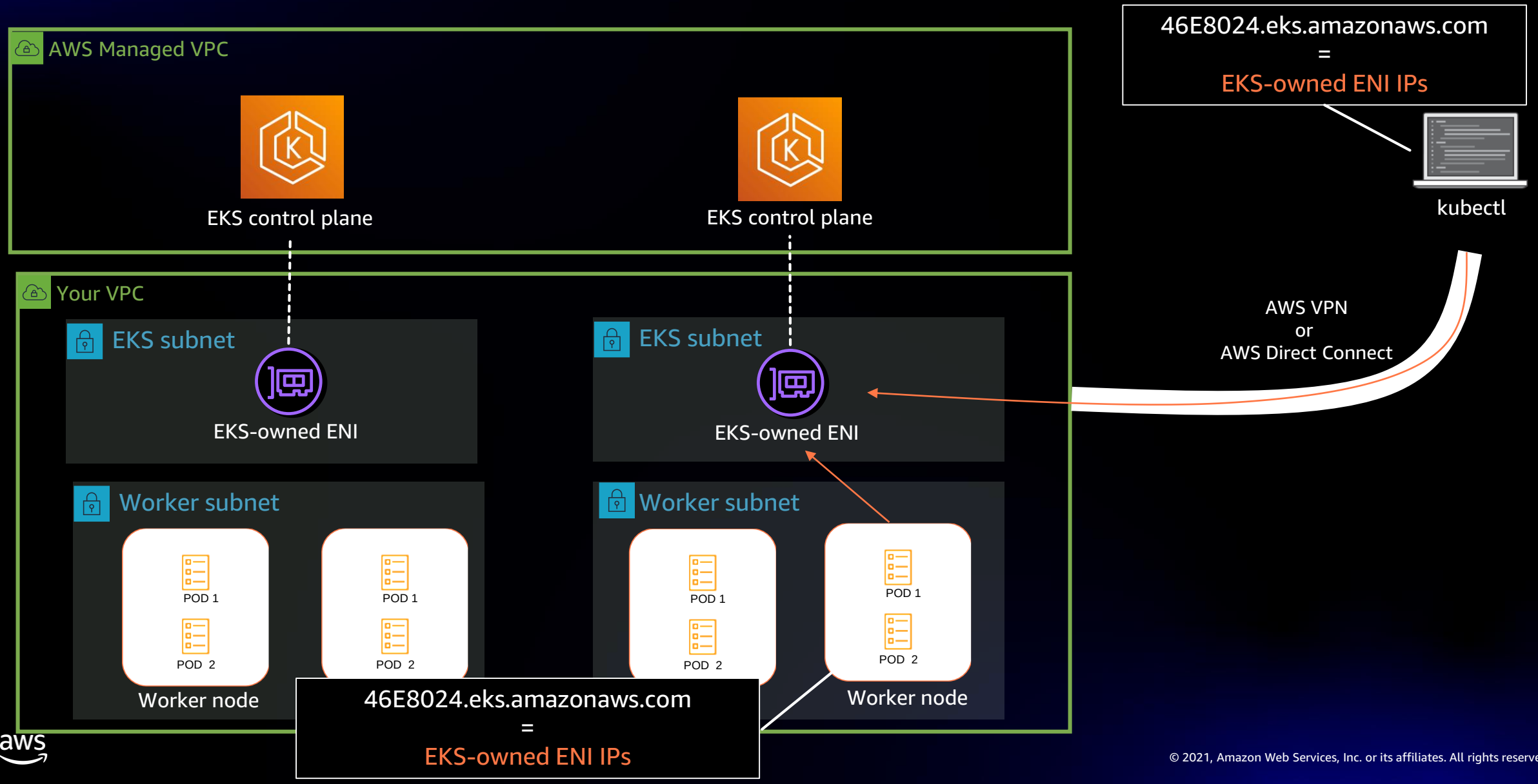
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS cluster endpoint access – Public & private



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS cluster endpoint access – Private only



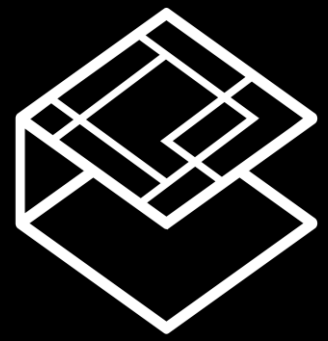
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 network settings

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes networking model

- Every Pod gets its own IP address
- Containers in the same Pod share the network (IP address)
- Pods communicate to other Pods without NAT



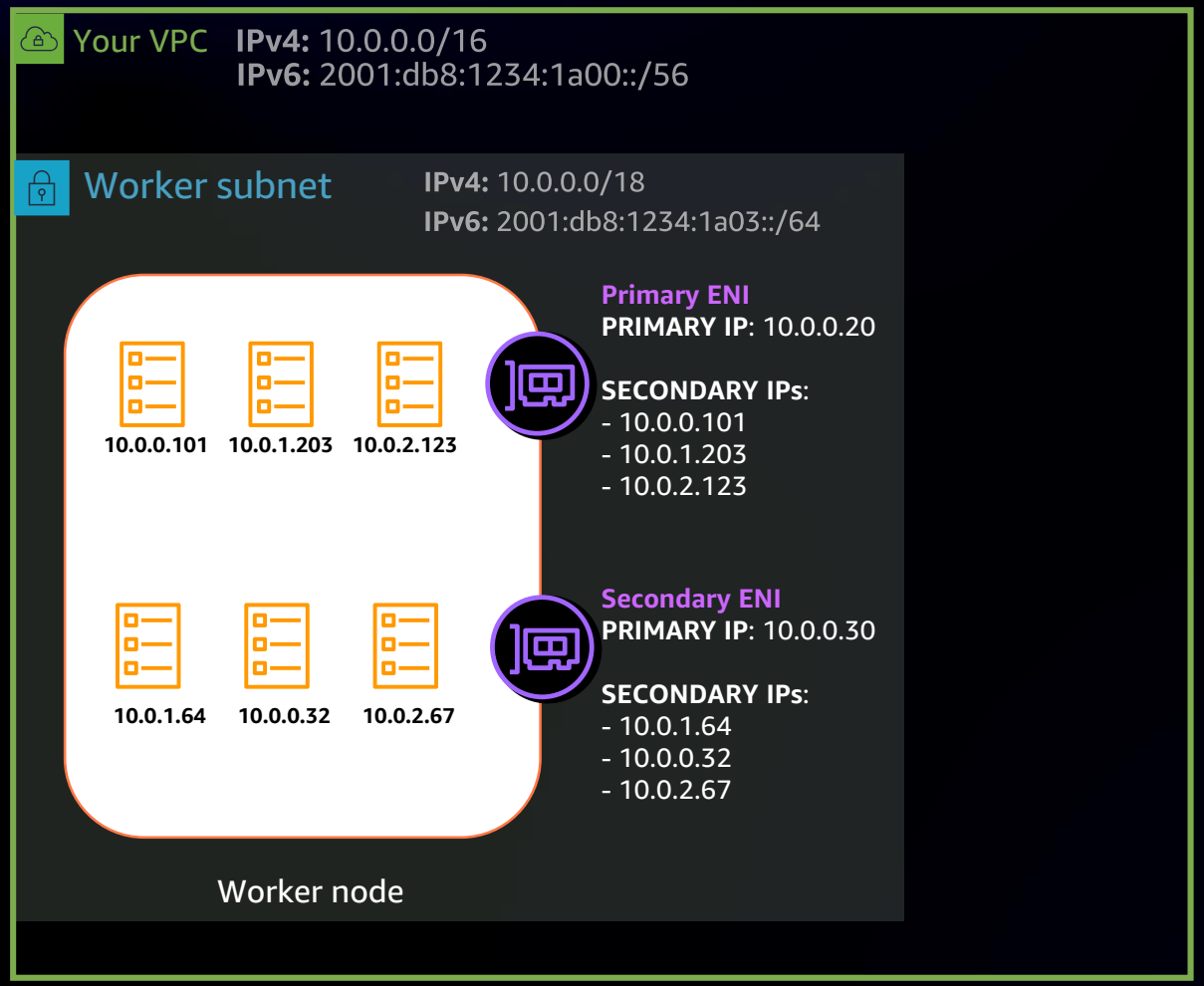
CNI

Container Network Interface plugin

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 networking – CNI plugin



The Amazon VPC Container Network Interface (CNI) plugin is used for

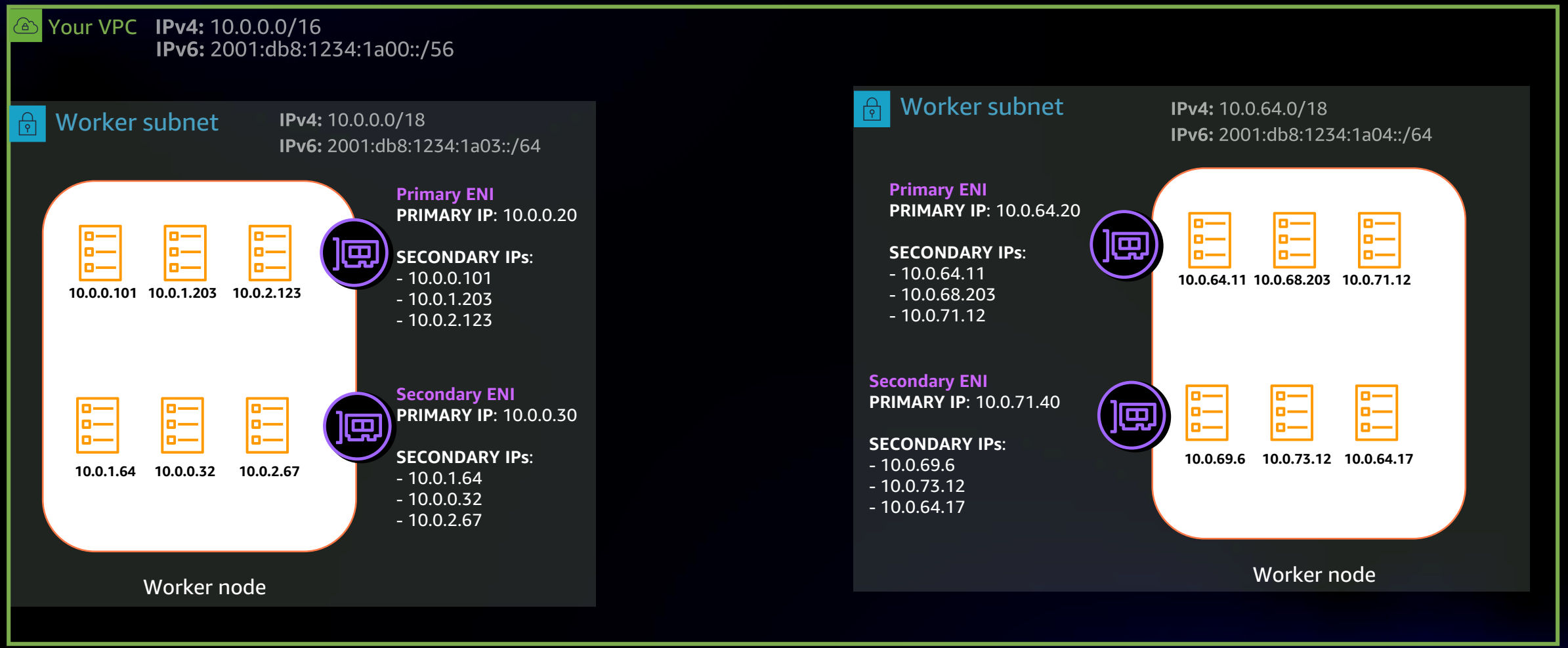
- Creating and attaching ENIs to worker nodes
- Assigning secondary IP addresses for Pods
- Wiring the host network

Alternatives to Amazon VPC CNI

- Calico from Tigera
- Cilium from Isovalent
- Weave Net from Weaveworks
- Antrea from VMware

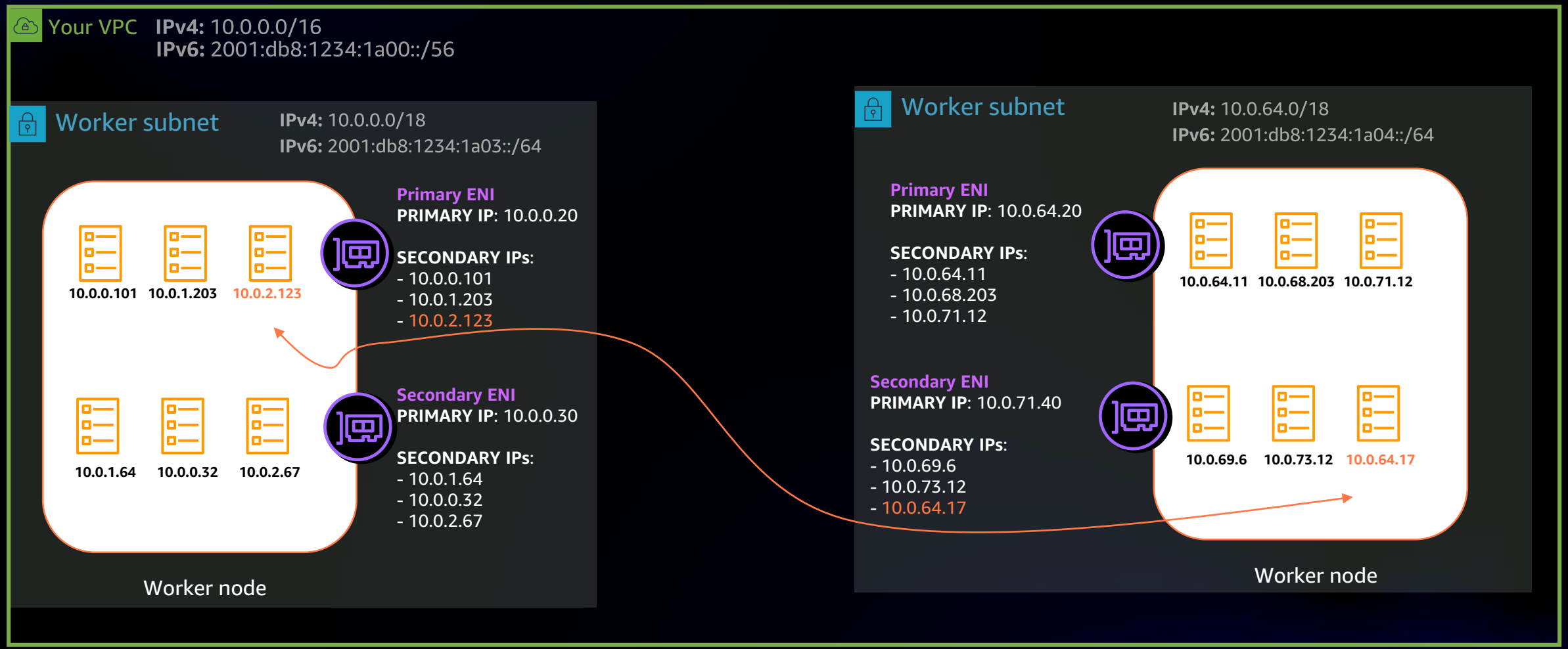
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 networking – Pod to Pod



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 networking – Pod to Pod



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CNI – Configuration options

- Source NAT

```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_EXTERNALSNAT=true/false
```

- Prefix delegation

```
kubectl set env daemonset aws-node -n kube-system ENABLE_PREFIX_DELEGATION=true/false
```

- Security groups for Pods

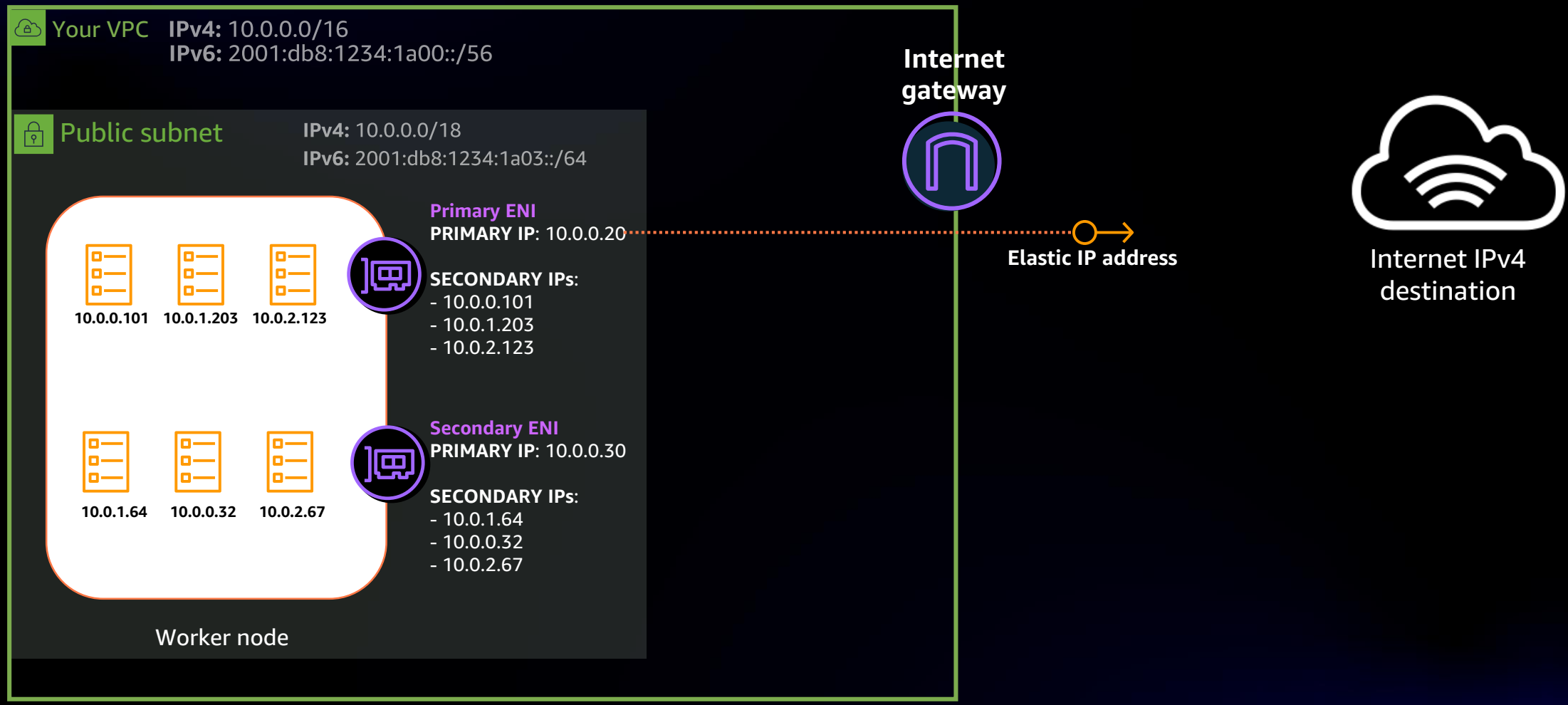
```
kubectl set env daemonset aws-node -n kube-system ENABLE_POD_ENI=true/false
```

- Custom networking

```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true/false
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

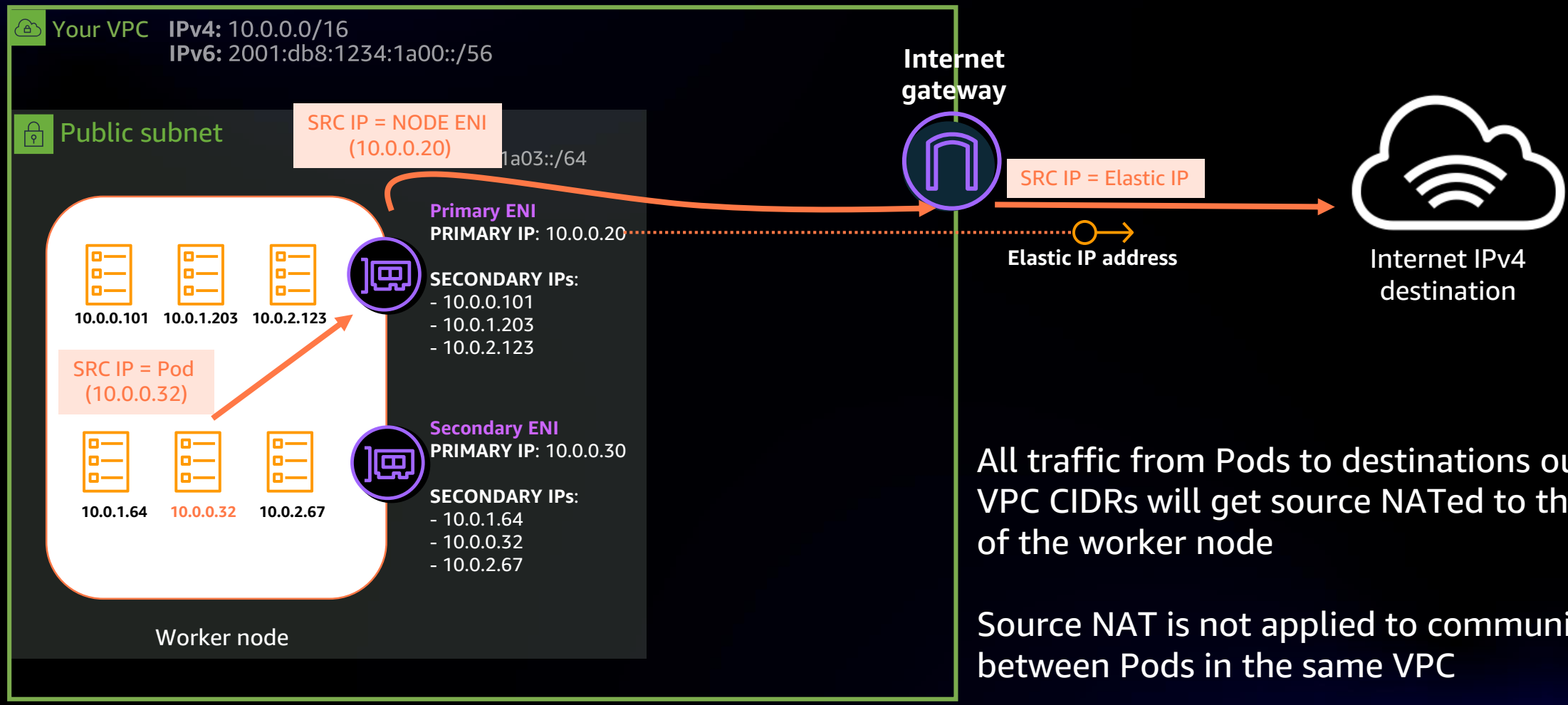
Pod IPv4 networking – Source NAT enabled




```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_EXTERNALSNAT=false
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 networking – Source NAT enabled



All traffic from Pods to destinations outside of the VPC CIDRs will get source NATed to the IP address of the worker node

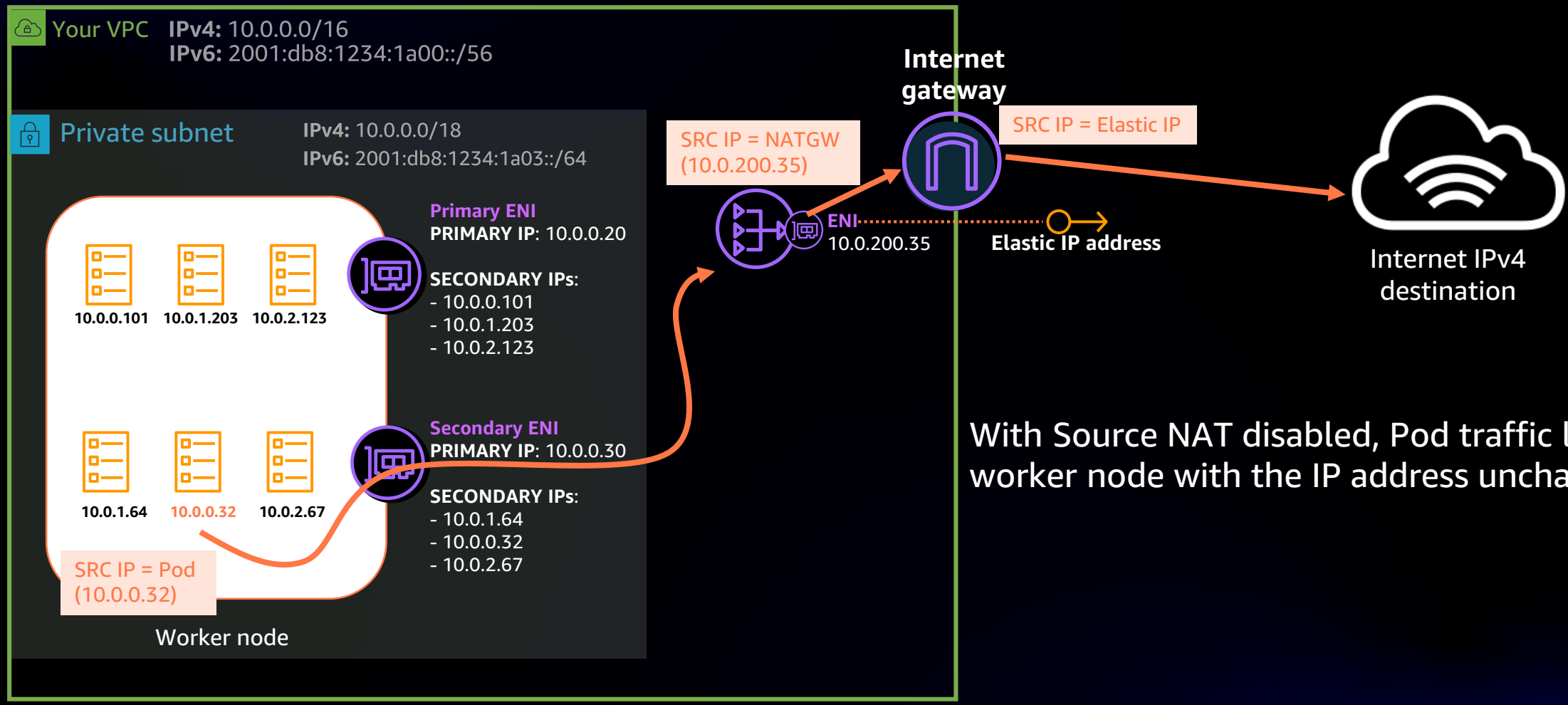
Source NAT is not applied to communication between Pods in the same VPC

```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_EXTERNALSNAT=false
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod networking – Source NAT disabled



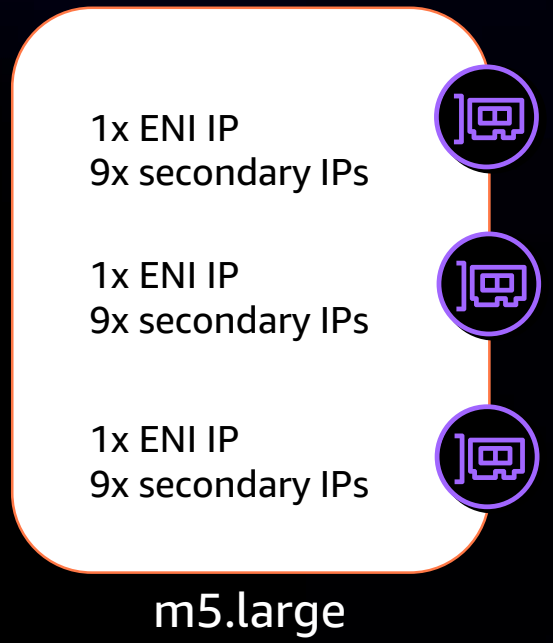
With Source NAT disabled, Pod traffic leaves the worker node with the IP address unchanged

```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_EXTERNALSNAT=true
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Secondary IPs for Pods per EC2 instance



Instance type	Maximum network interfaces	Private IPv4 addresses per interface
m5.large	3	10

MAX PODS = (Number of network interfaces × [the number of IP addresses per network interface – 1]) + 2

29

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon VPC prefix delegation

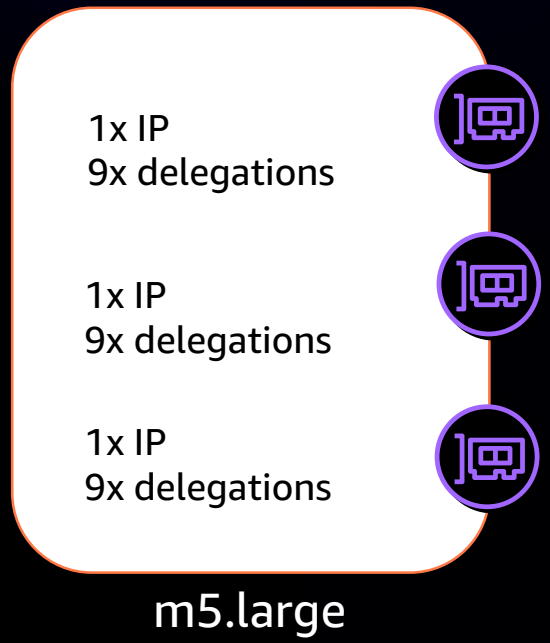
Allows for assigning a prefix to an EC2 ENI

- /28 block for IPv4 (16x IPv4 addresses)
- /80 block for IPv6 (280 trillion IPv6 addresses)

Prefix delegation is only supported on Nitro instances

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prefix delegation



Instance type	Maximum network interfaces	Private IPv4 addresses per interface
m5.large	3	10

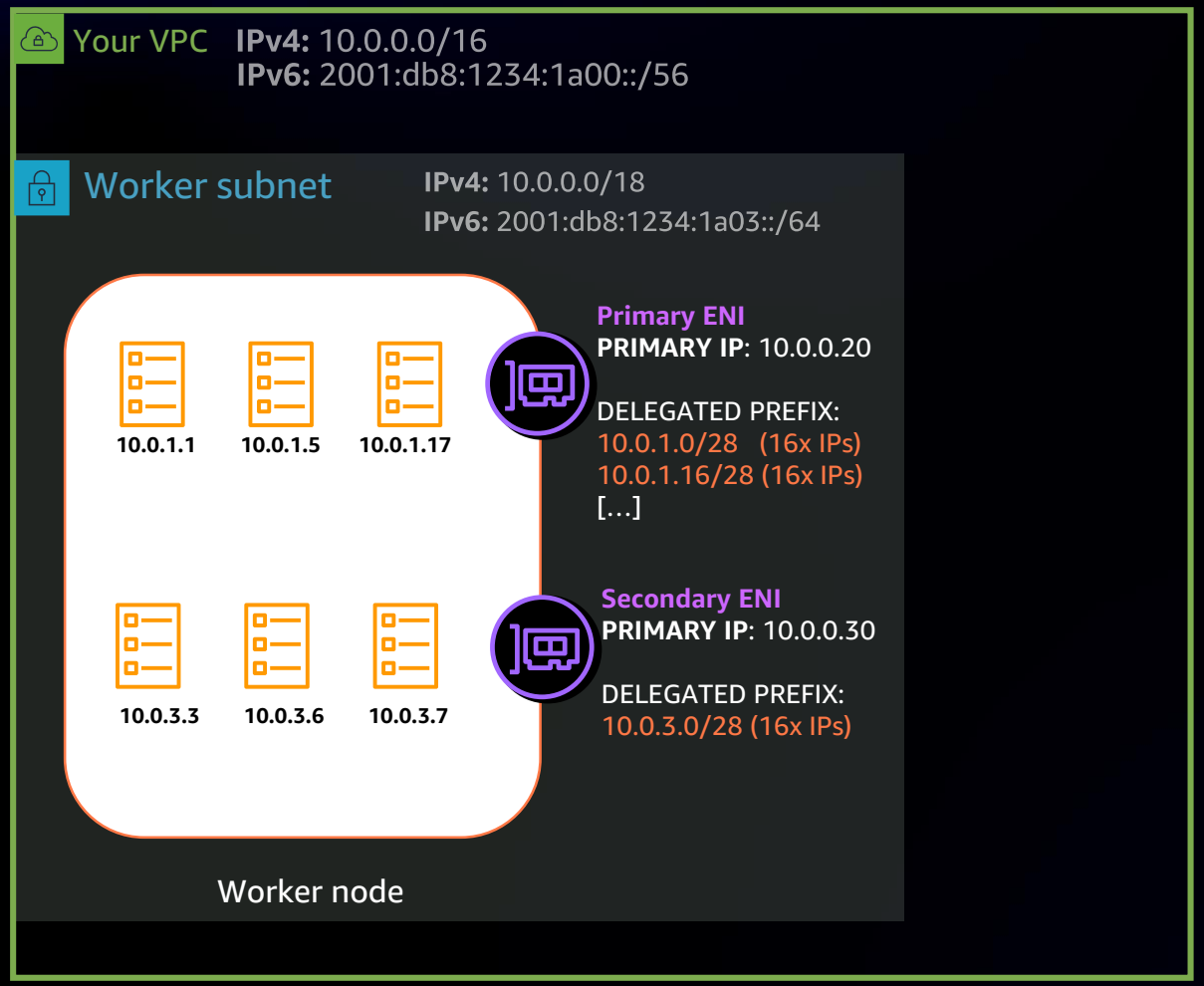
MAX IPs ≠ **MAX PODS**

432 IPv4
Trillions IPv6

110

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod IPv4 networking – Prefix delegation



Benefits

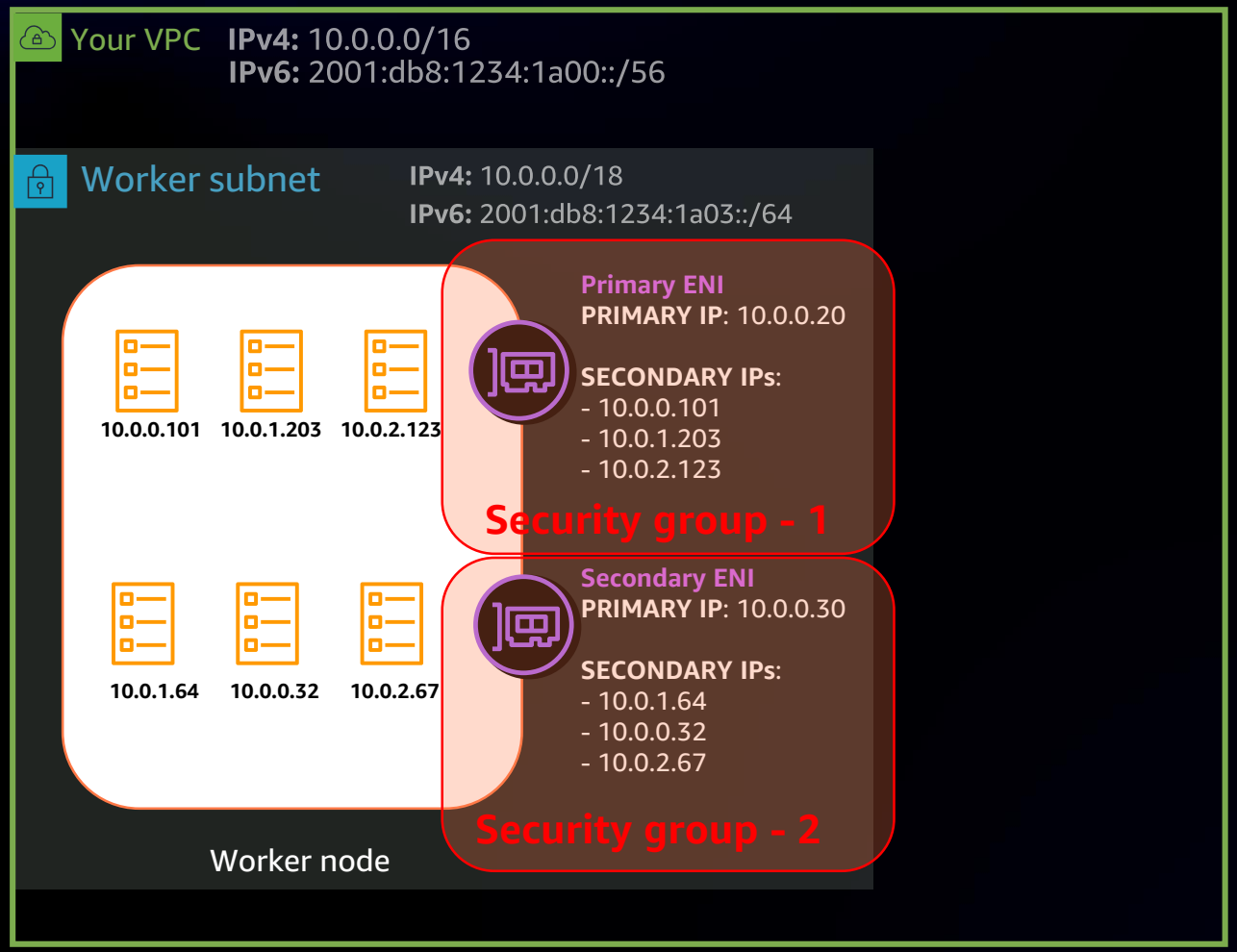
- Increased Pod density
- Fewer API calls required to EC2 control plane

```
kubectl set env daemonset aws-node -n kube-system ENABLE_PREFIX_DELEGATION=true
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

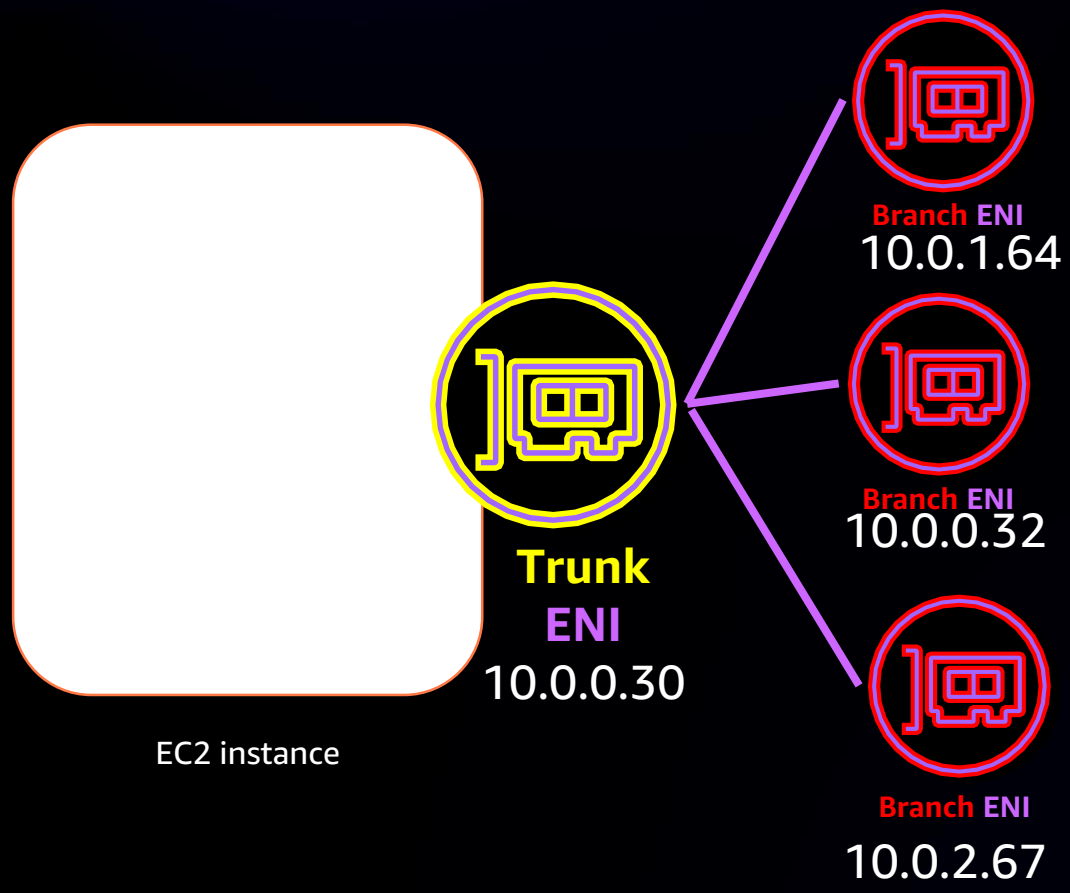
Pod networking – Security groups for Pods



Pods share the security group on their respective ENI

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

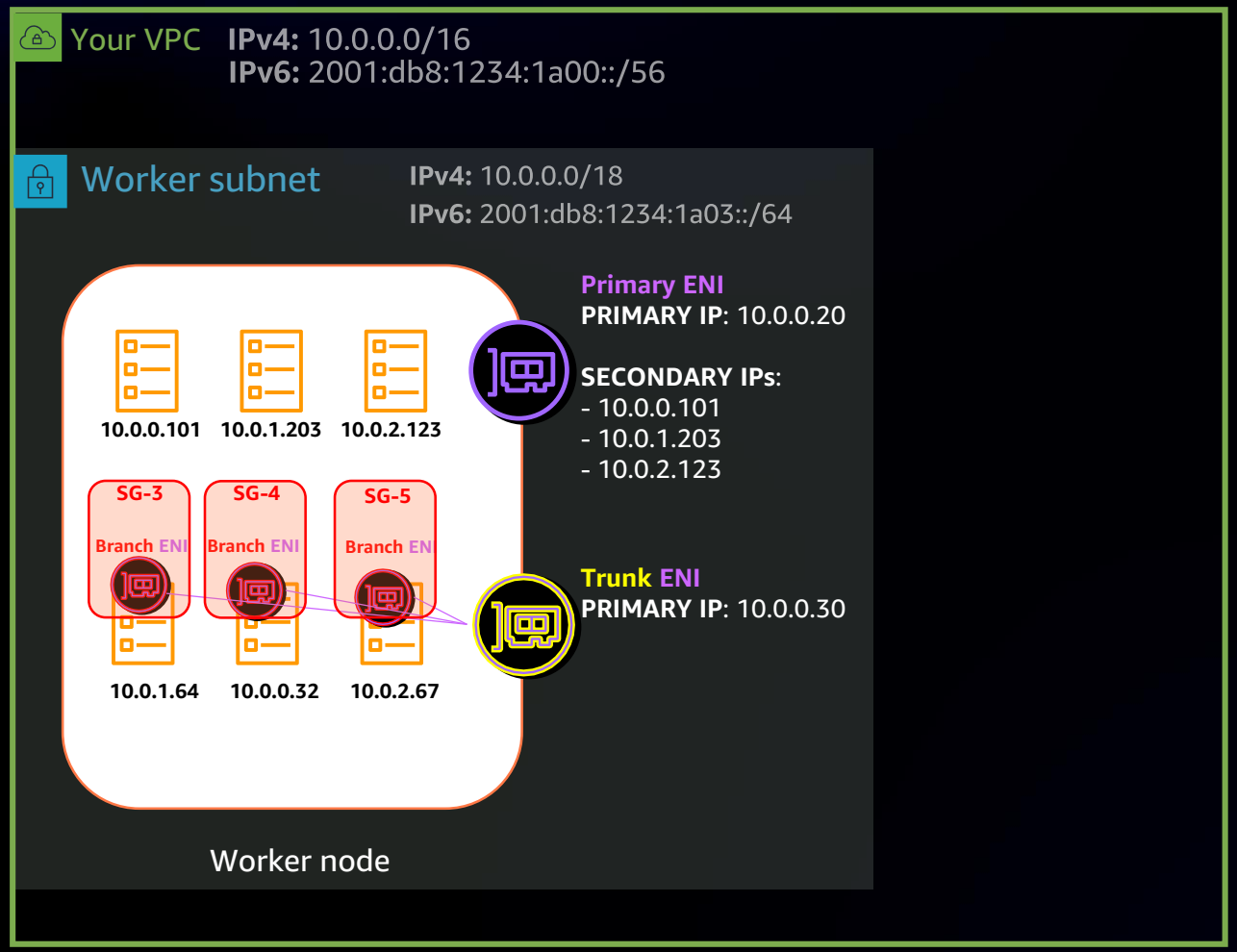
Branch and trunk ENIs



Feature only available through Amazon EKS or Amazon ECS and not directly on Amazon EC2

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod networking – Security groups for Pods



Each Pod gets a dedicated ENI (branch-ENI) mapped to a trunk ENI, allowing for independent security group configurations per Pod

Considerations

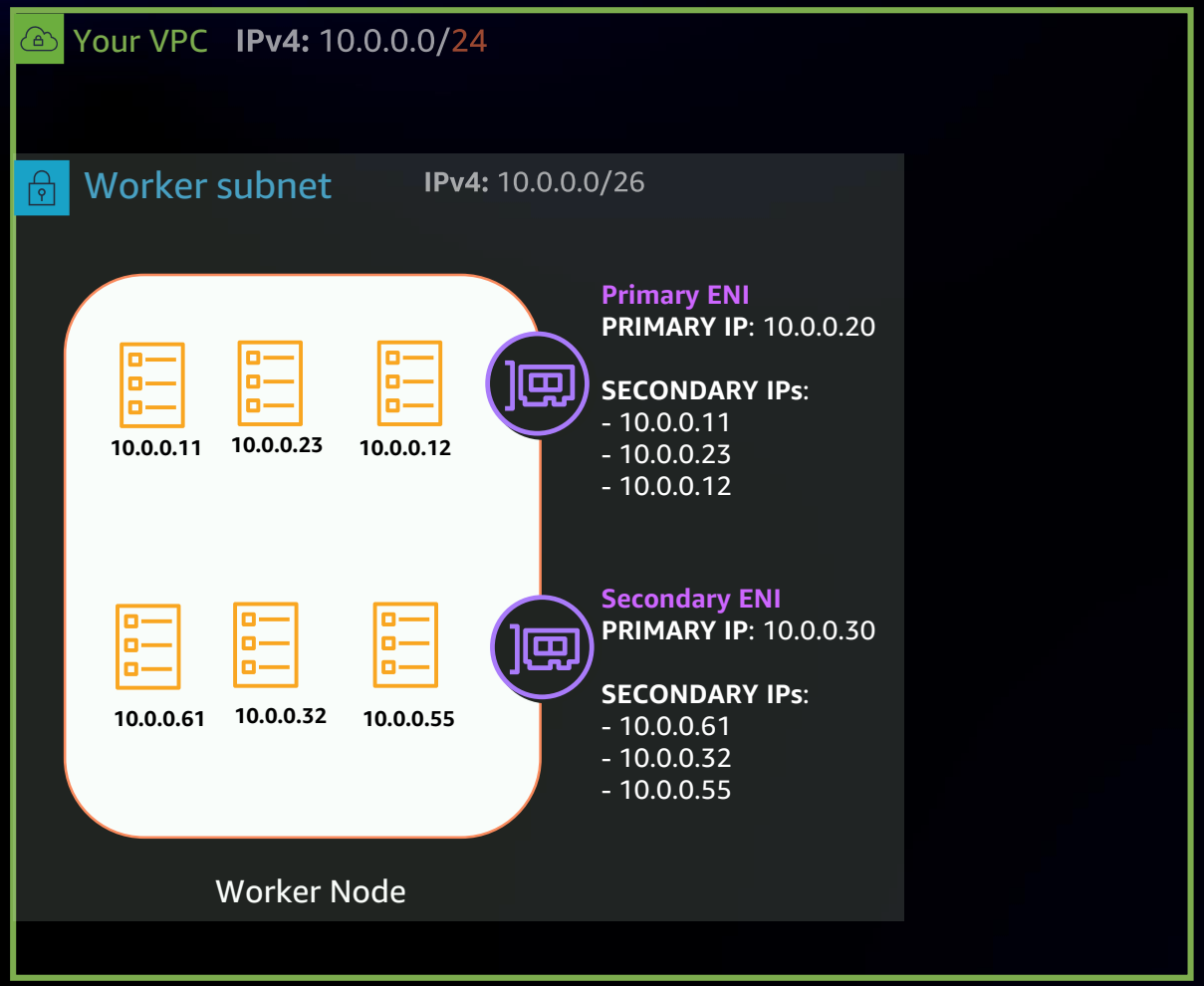
- Supported by most Nitro instances
- SNAT is disabled for Pods using branch ENIs
- Cluster version 1.18+
- Review integration with NodePort and LoadBalancer services

```
kubectl set env daemonset aws-node -n kube-system ENABLE_POD_ENI=true
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod networking – Custom networking

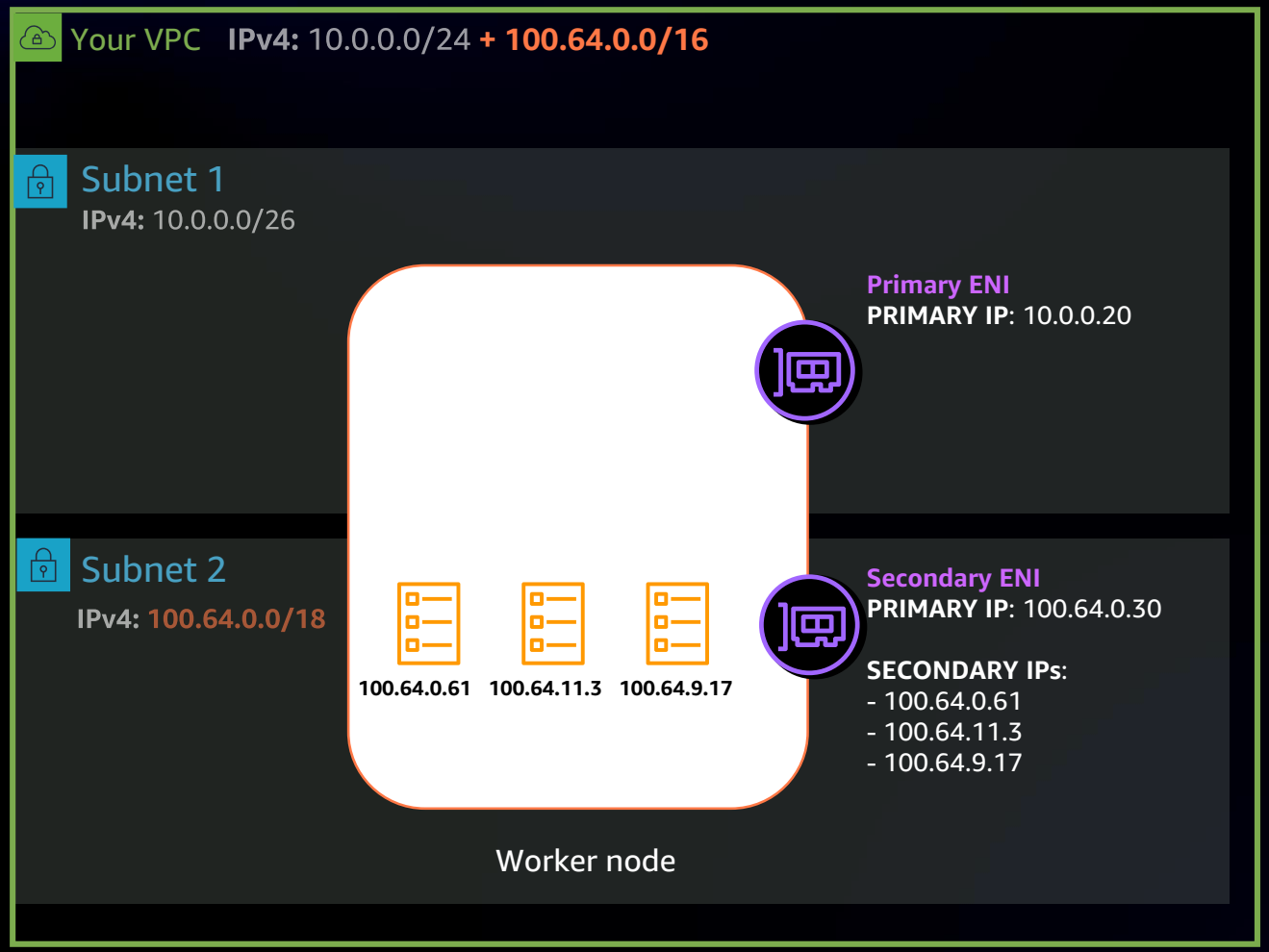


If the private IPv4 space is limited, the number of available IP addresses could constrain the number of Pods

/24 VPC CIDR provides 251 unique IPv4 addresses

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod networking – Custom networking



VPCs can have multiple IPv4 CIDR ranges

100.64.0.0/10 (RFC 6598) can be used in private networks

Worker node primary ENI is not used for Pods

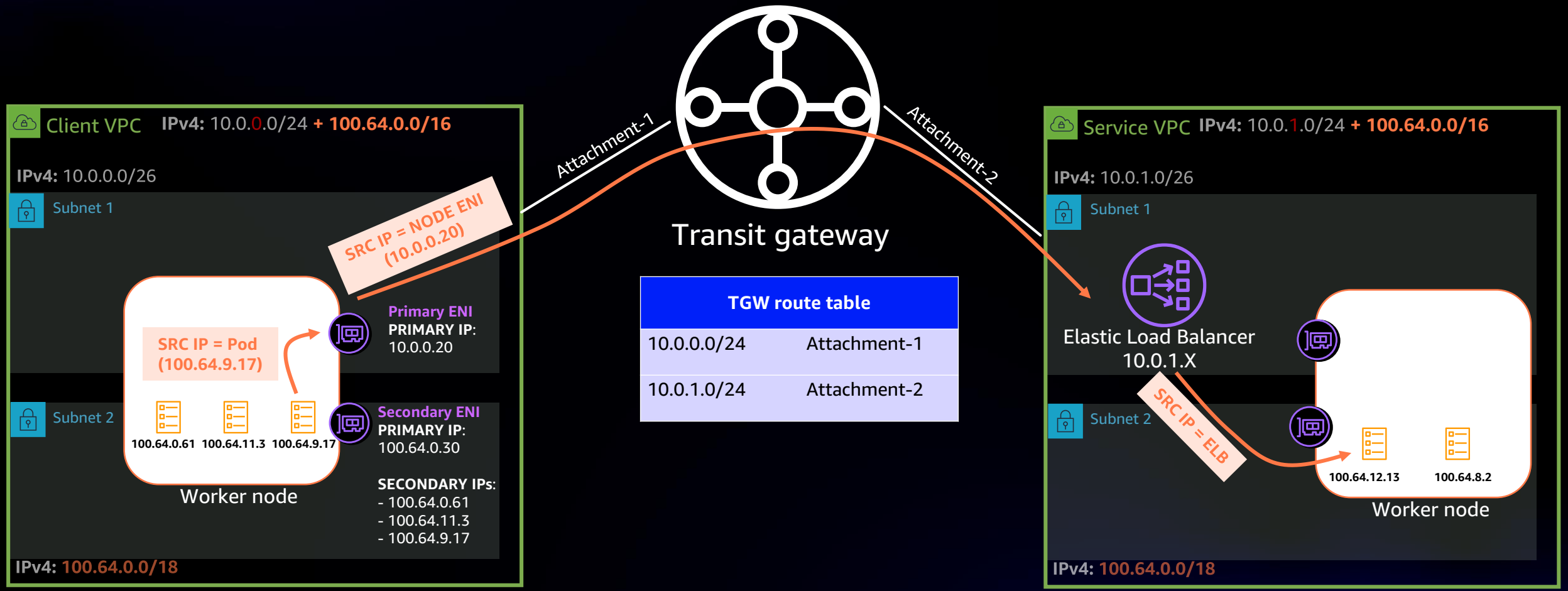
Custom networking can be combined with SNAT and prefix delegation

```
kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true
```

aws

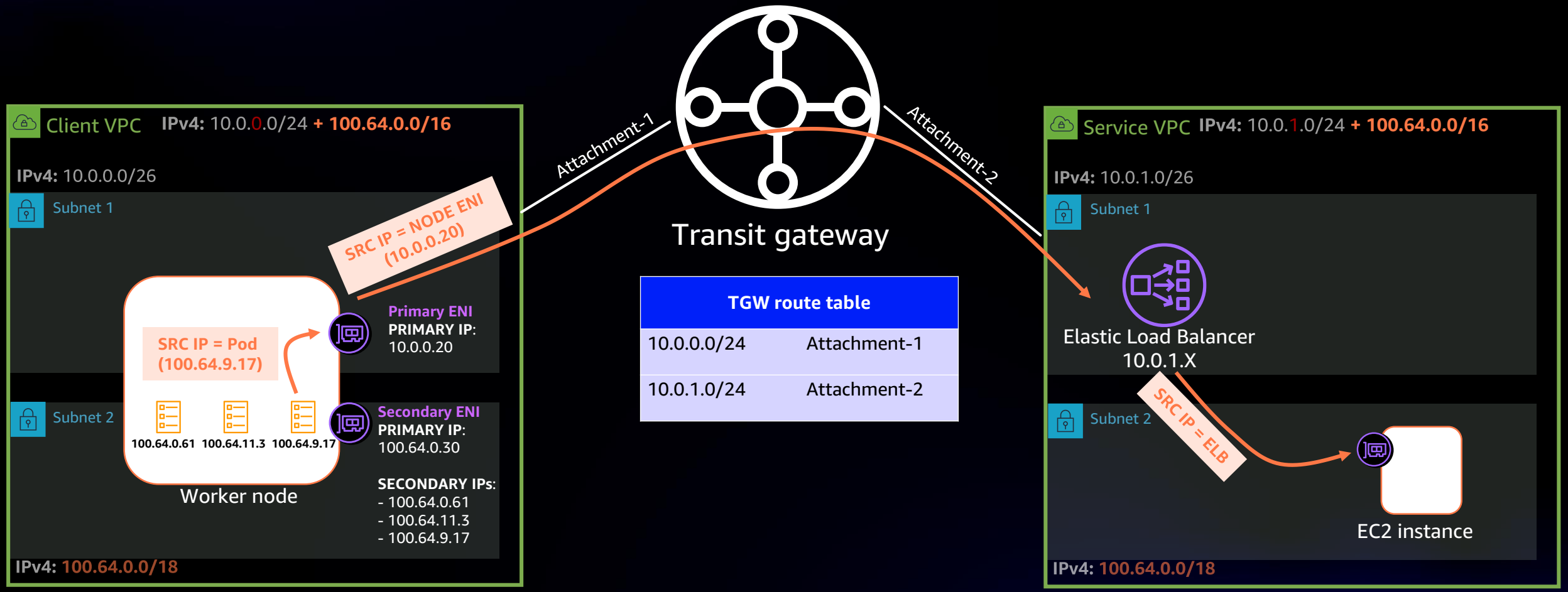
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Custom networking + SNAT + transit gateway



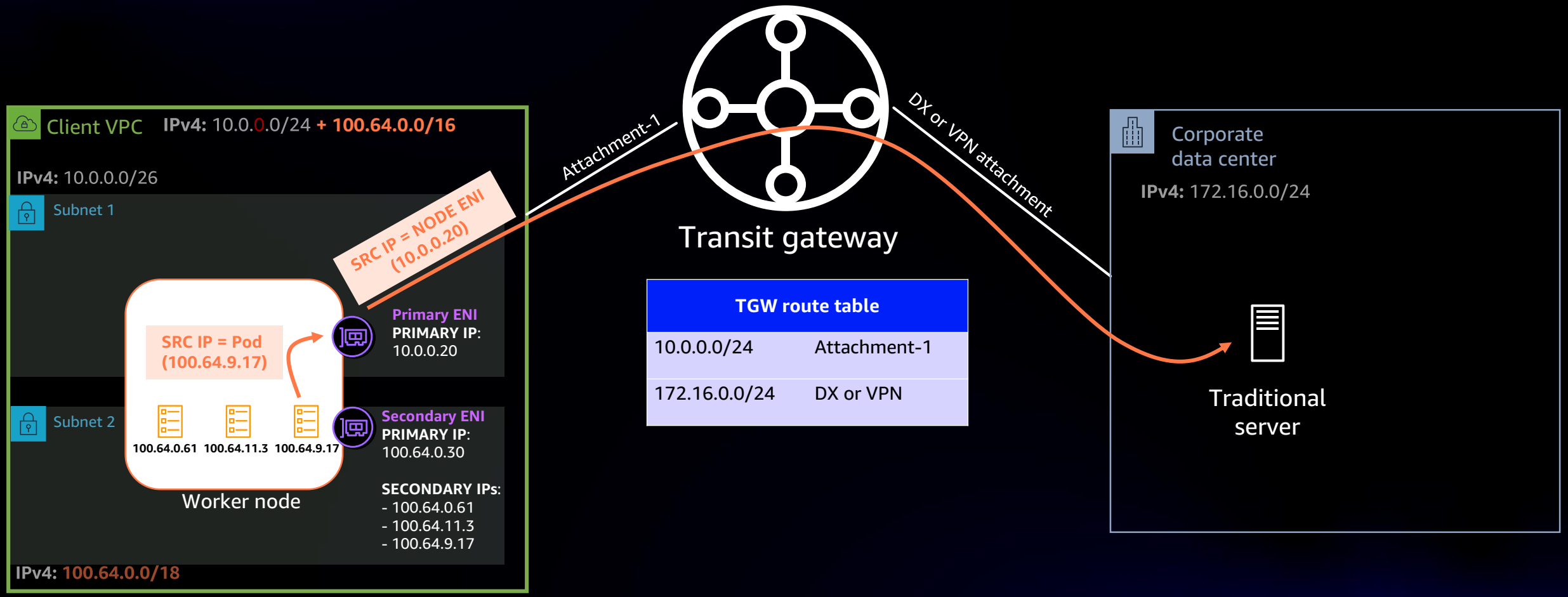
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Custom networking + SNAT + transit gateway



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Custom networking + SNAT + transit gateway



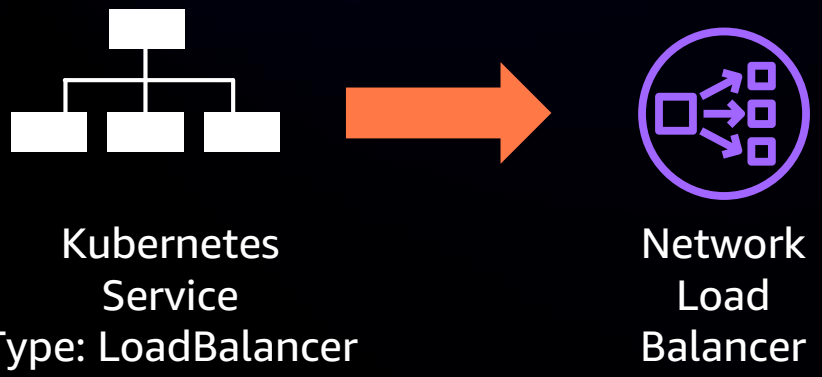
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Exposing applications

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Load Balancer Controller

A KUBERNETES CONTROLLER FOR ELASTIC LOAD BALANCERS



- Manages AWS Elastic Load Balancing for an Amazon EKS or a Kubernetes cluster
- Uses standard Kubernetes resources
  - v1: Service (Type: LoadBalancer)
  - networking.k8s.io/v1: Ingress
  - networking.k8s.io/v1: IngressClass
- Custom resources
  - elbv2.k8s.aws/v1beta1: TargetGroupBinding
- Supports latest ELB features

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Network Load Balancer provisioning in EKS

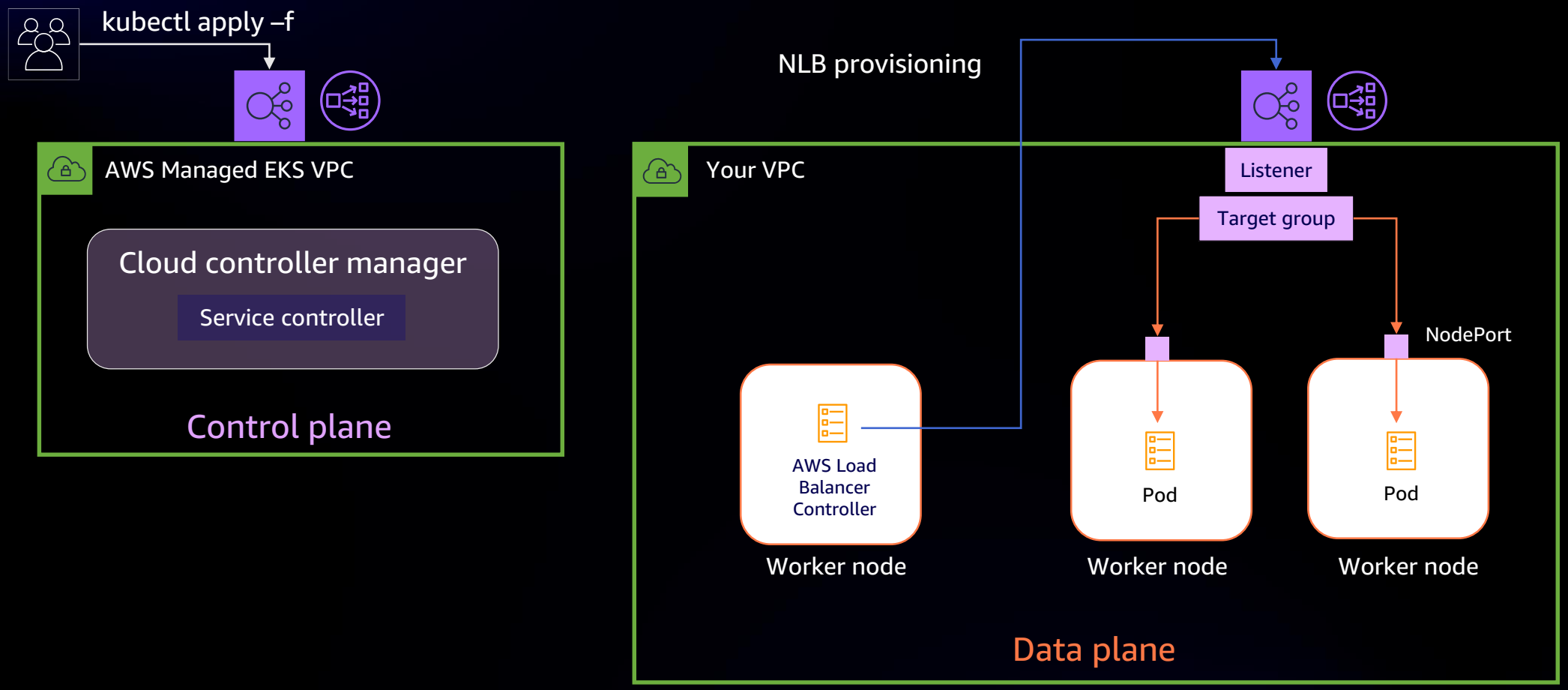
```
kind: Service
apiVersion: v1
metadata:
   name: nginx-service
   namespace: ingress-nginx
   annotations:
      service.beta.kubernetes.io/aws-load-balancer-backend-protocol: tcp
      service.beta.kubernetes.io/aws-load-balancer-cross-zone-load-balancing-enabled: “true”
      service.beta.kubernetes.io/aws-load-balancer-type: "external"
      service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: “instance"
      service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing
spec:
   type: LoadBalancer
   selector:
      app: nginx
   ports:
      - name: http
        protocol: TCP
        port: 8080
        targetPort: 80
```

Using AWS Load Balancer Controller

requires version 2.2.0 or later

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

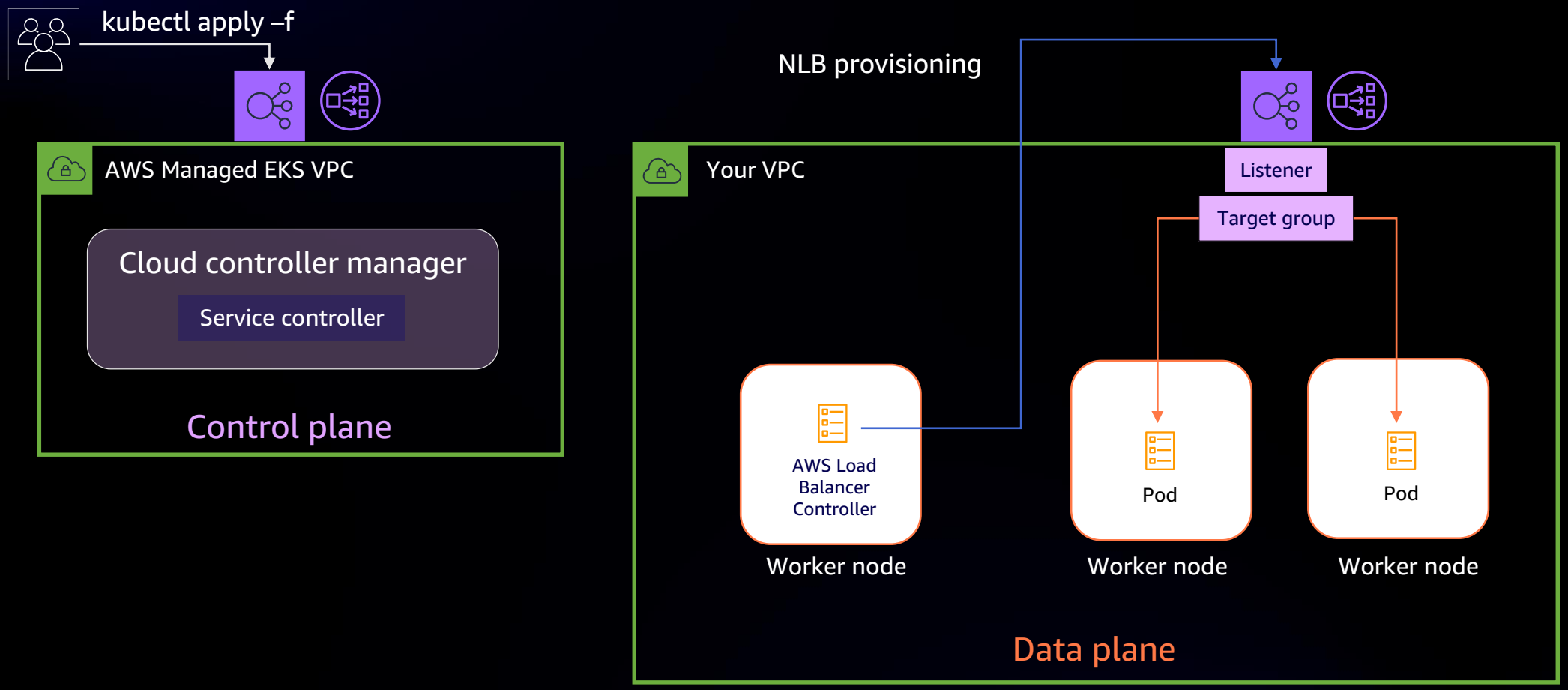
Network Load Balancer provisioning in EKS



service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: instance

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Network Load Balancer provisioning in EKS



service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

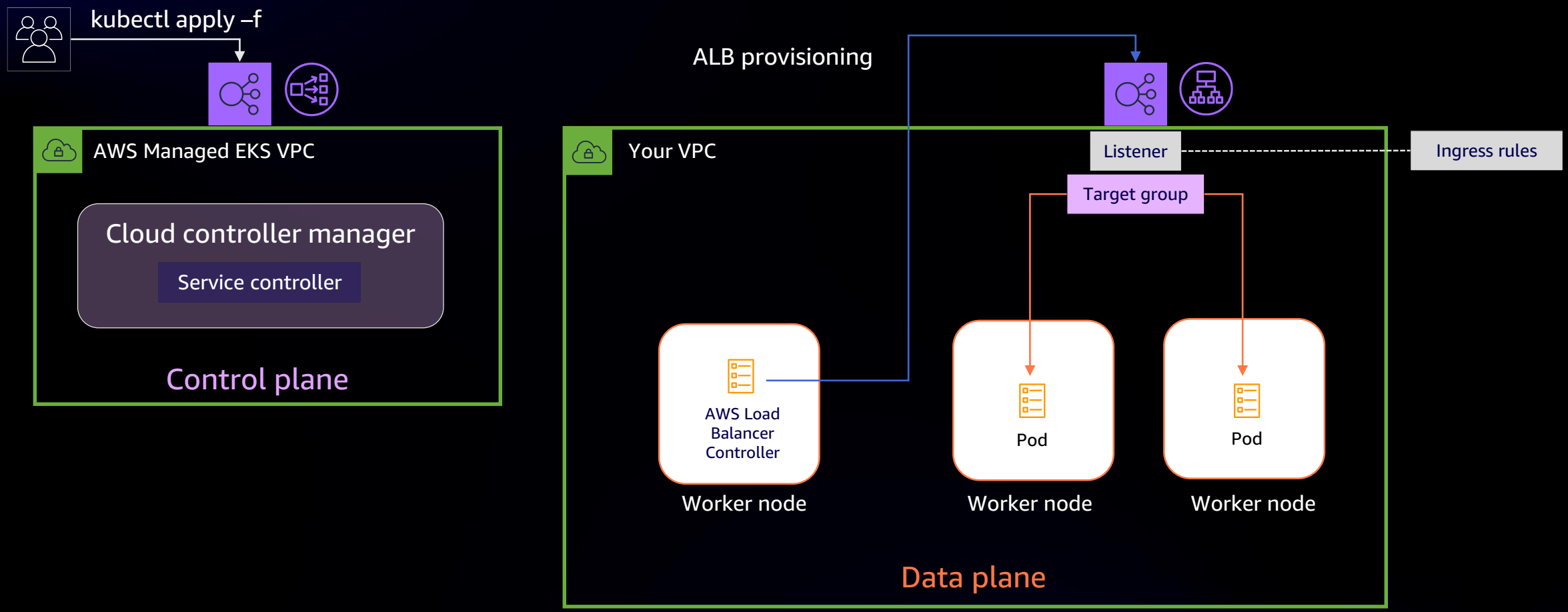
Application Load Balancer provisioning in EKS

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: echoserver-ingress
  namespace: echoserver
  annotations:
    alb.ingress.kubernetes.io/security-groups: sg-010fc3455c73f0a58, sg-049e999c68a291976
    alb.ingress.kubernetes.io/target-type: ip
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/group.name: web-services
    kubernetes.io/ingress.class: alb
spec:
  ingressClassName: alb-ingress-class
  rules:
    - host: echoserver.example.com
      http:
      paths:
        - path: /*
          backend:
            serviceName: echoserver-svc
            servicePort: 80
```

```
apiVersion: networking.k8s.io/v1beta1
kind: IngressClass
metadata:
  name: alb-ingress-class
spec:
  controller: ingress.k8s.aws/alb
```

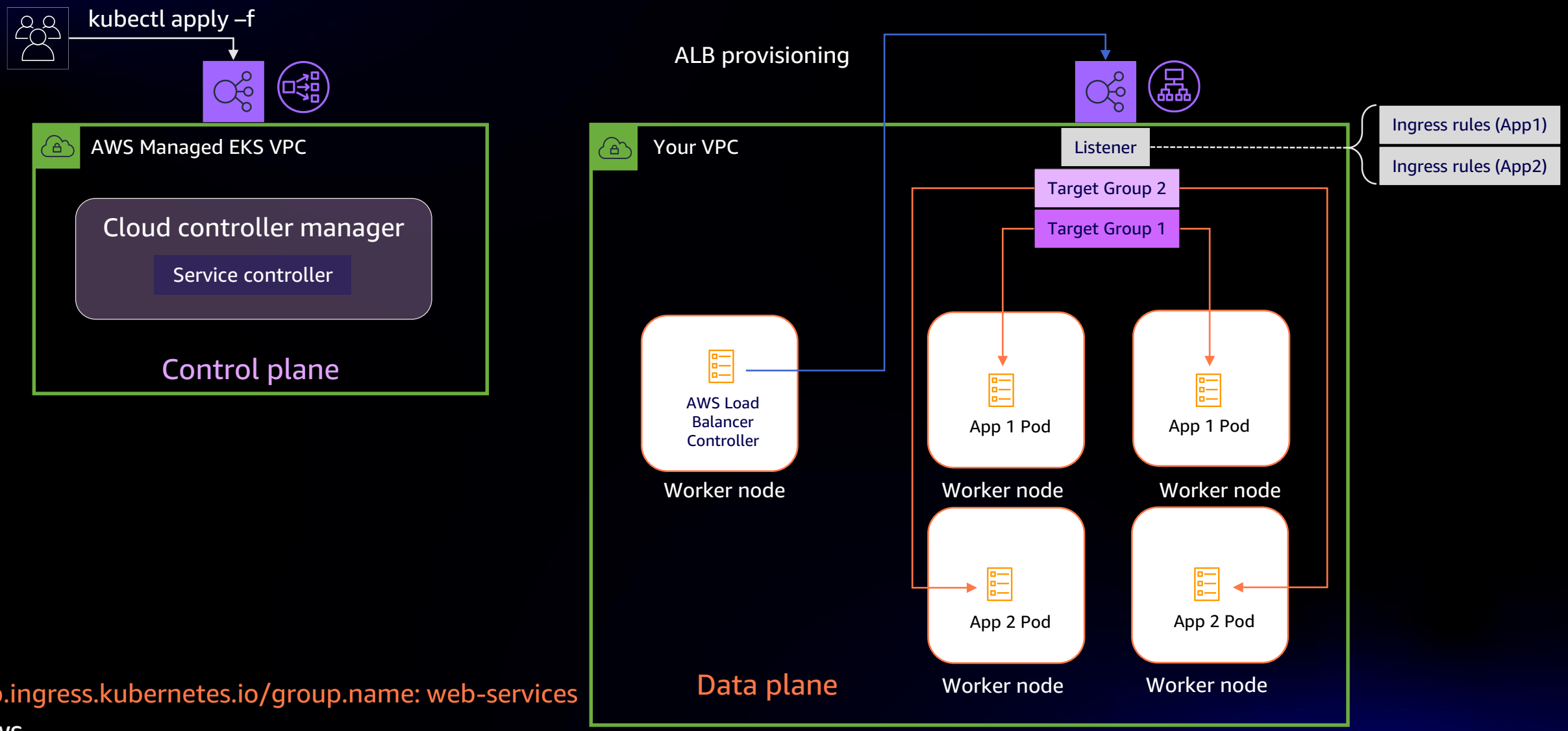
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application Load Balancer provisioning in EKS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

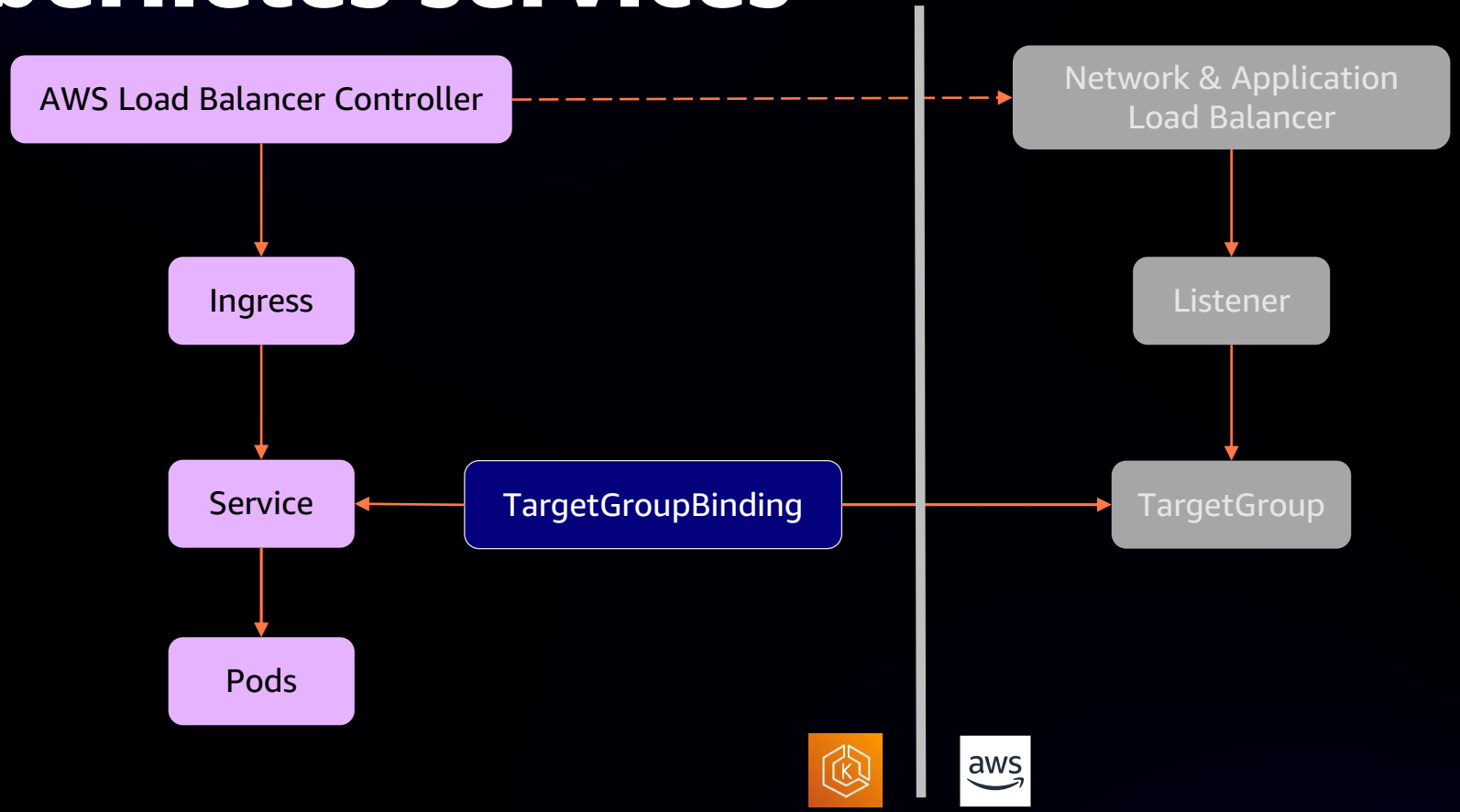
Application Load Balancer provisioning in EKS



alb.ingress.kubernetes.io/group.name: web-services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Binding AWS target groups and Kubernetes services



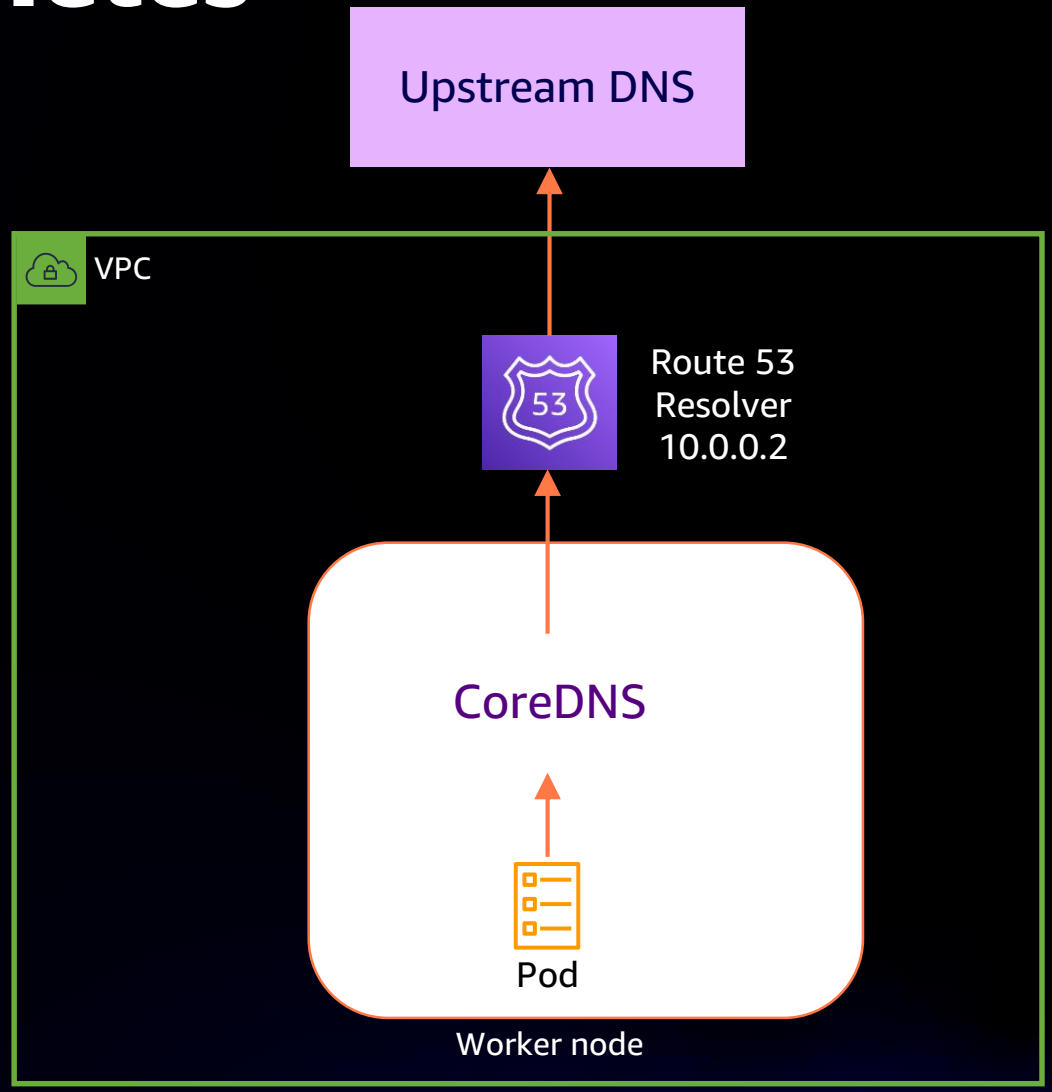
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service discovery

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DNS resolution in Kubernetes

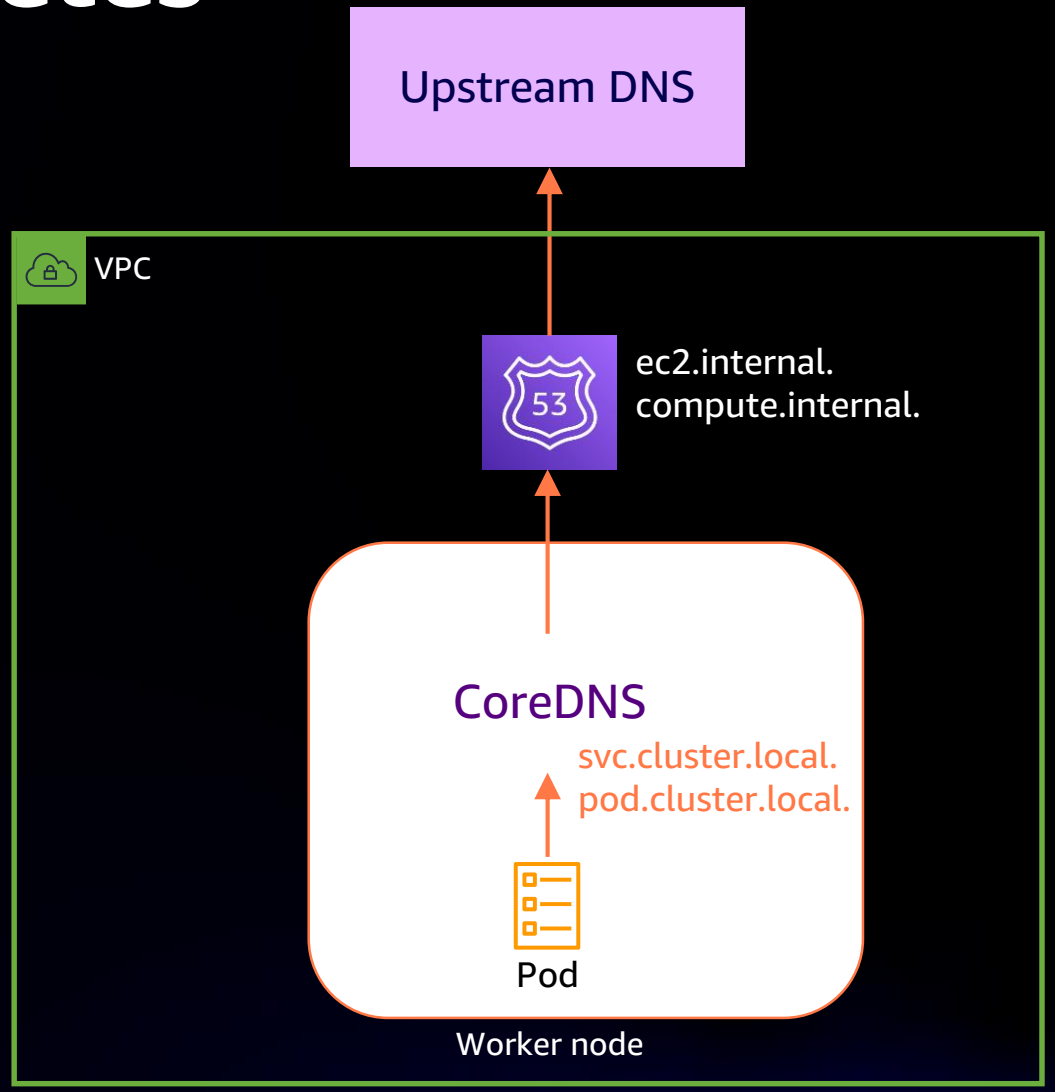
- CoreDNS add-on resolves DNS names of resources within the EKS cluster
- Default Route 53 Resolver for the VPC resolves resources within the VPC but outside the cluster
- Upstream DNS name server for resources outside the VPC



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

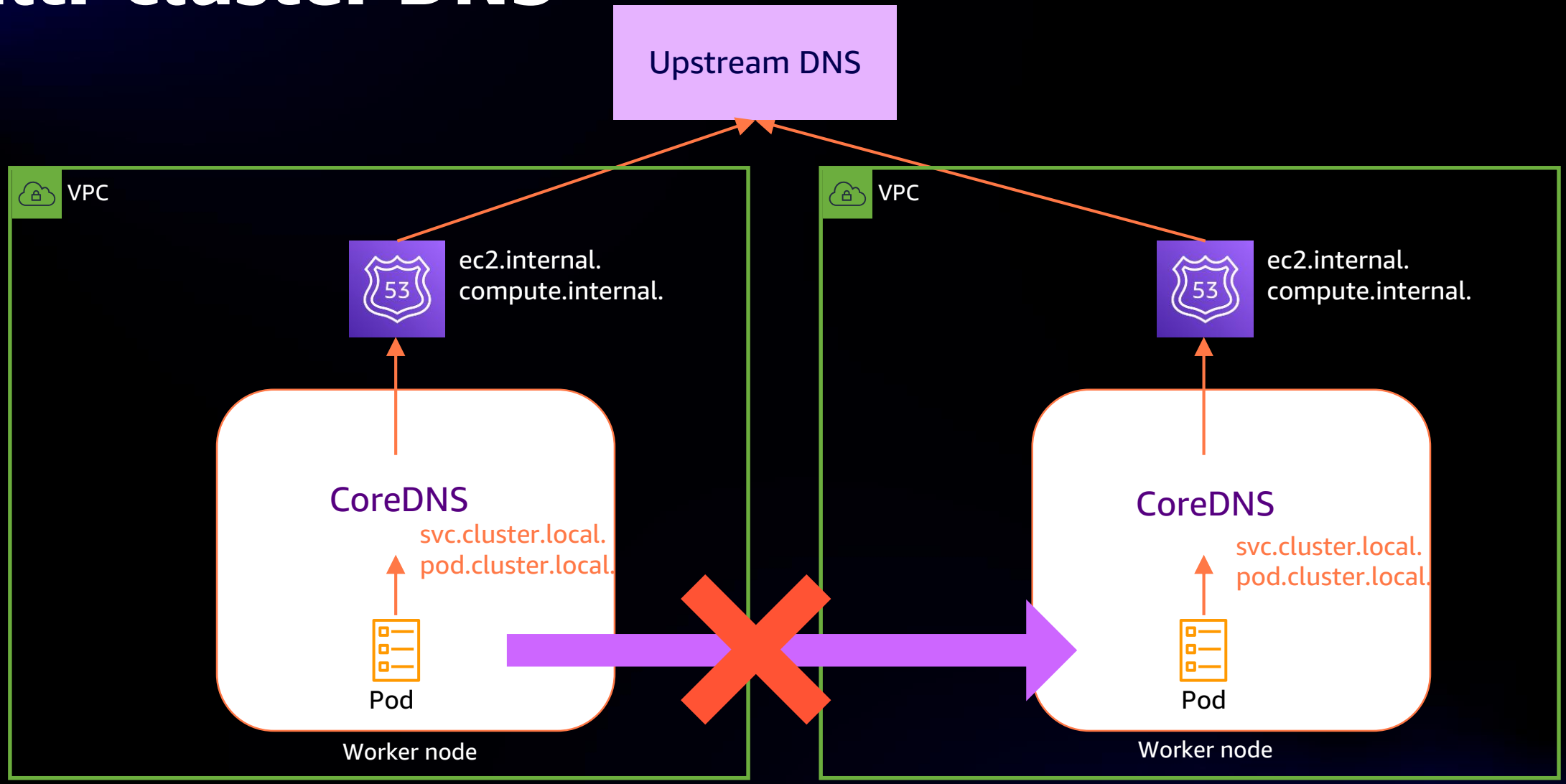
DNS authority in Kubernetes

- CoreDNS
  - svc.cluster.local.
  - pod.cluster.local.
- Kubernetes creates a subdomain per namespace
- Route 53 Resolver
  - ec2.internal. (us-east-1)
  - compute.internal. (other regions)



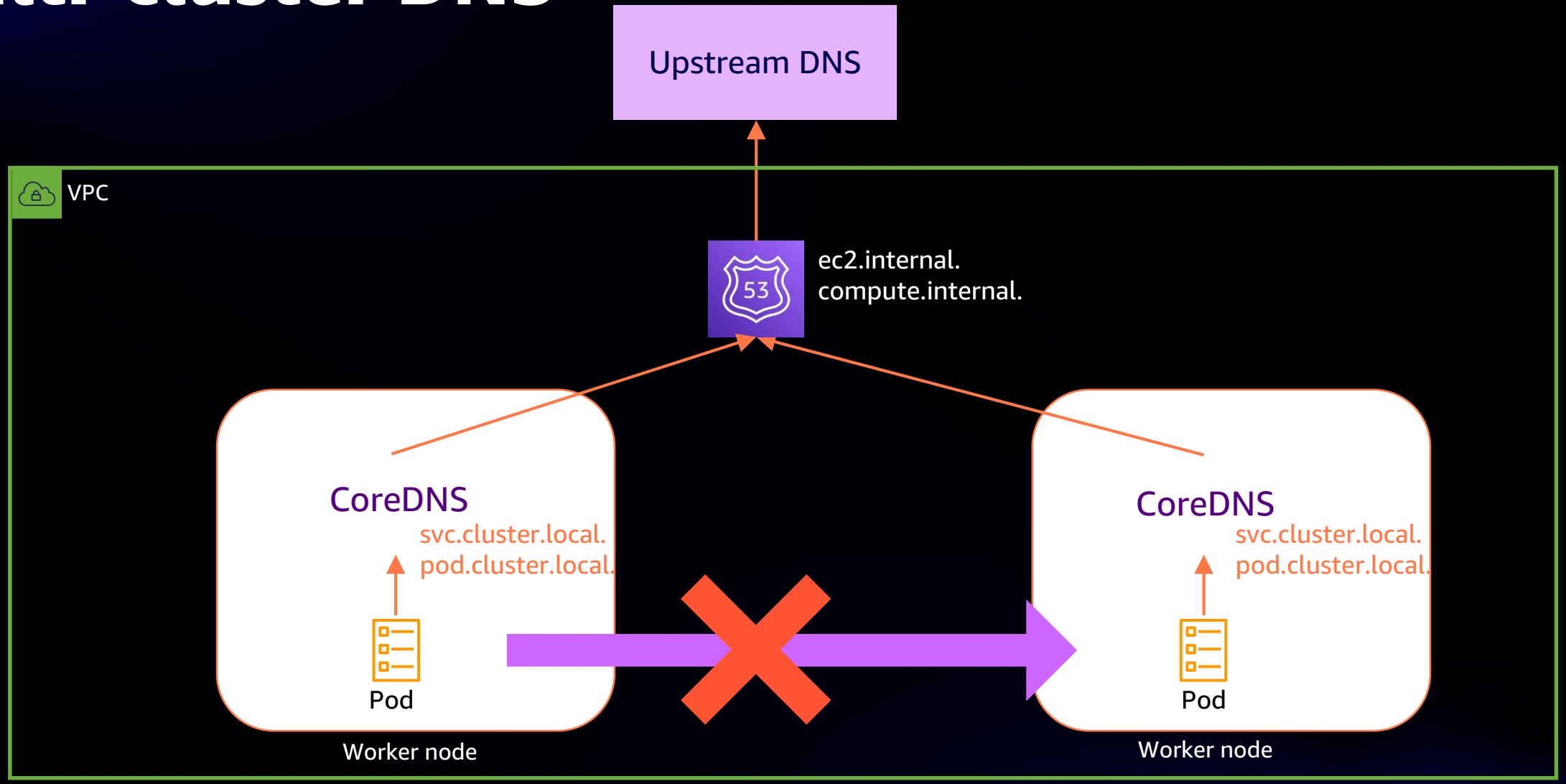
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-cluster DNS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-cluster DNS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Cloud Map registry

- Cloud map namespace ≈ DNS domain
  - For example, services.company.internal.
- Connect each EKS/kubernetes service to Cloud Map service using external-dns
- Works with multiple namespaces
- Works with multiple EKS/Kubernetes clusters
- Works with Route 53 Resolvers



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Cloud Map discovery methods

DNS query

- A, AAAA, or SRV records
- Up to 8 healthy records in answer



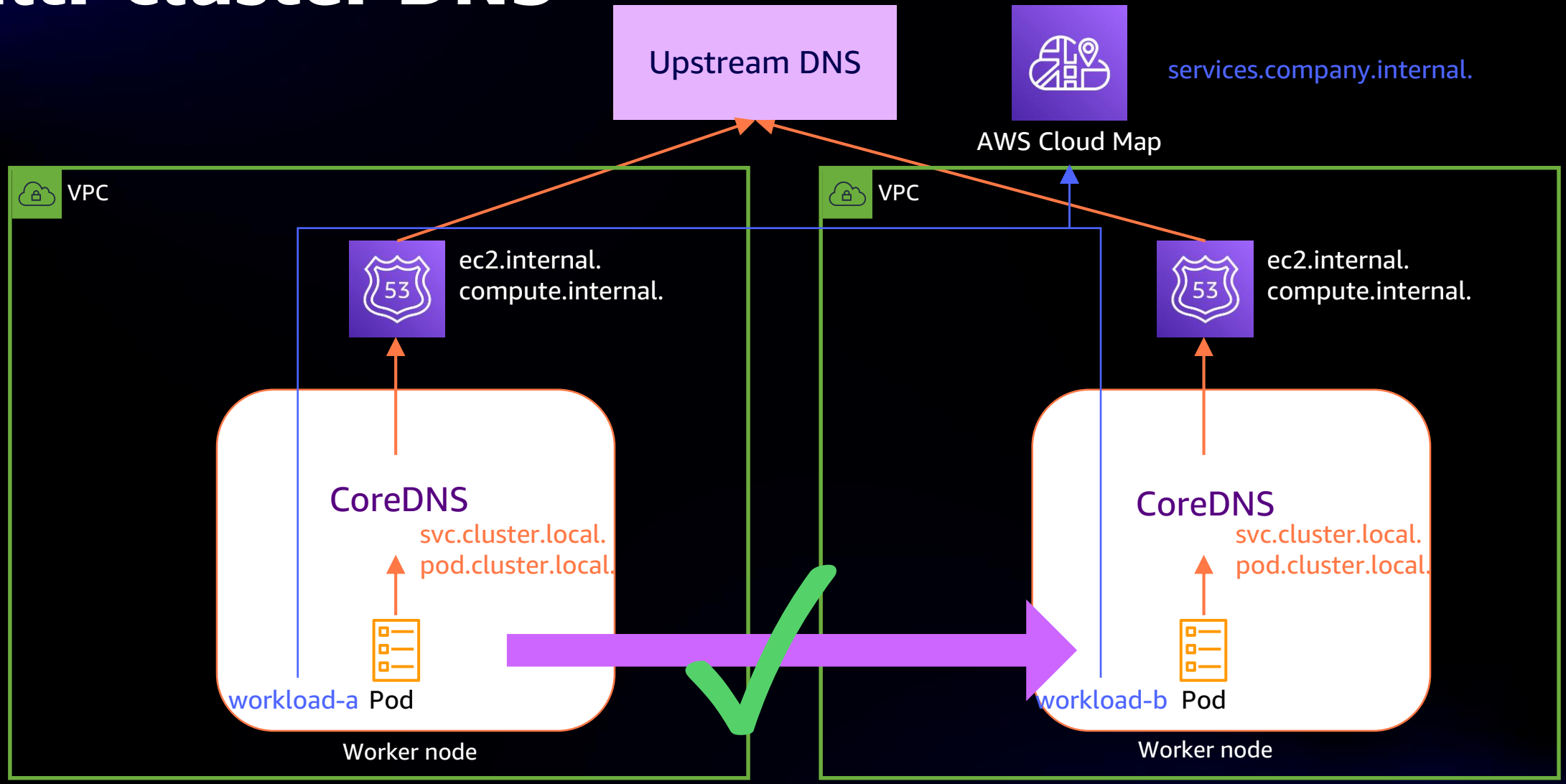
DiscoverInstances API

- Supports custom attributes
- Up to 100 records in answer
  - Can query for health status



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-cluster DNS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Cloud Map MCS Controller



- Multi-Cluster Service API from the Kubernetes community
  - Still in alpha (Proposal: KEP-1645 [https://bit.ly/3jxfWcA])
- Adds new zone to CoreDNS
  - svc.clusterset.local.
- Can reference services in another cluster using Kubernetes
  - ServiceExport
  - ServiceImport → creates EndpointSlices
- Not involved in network

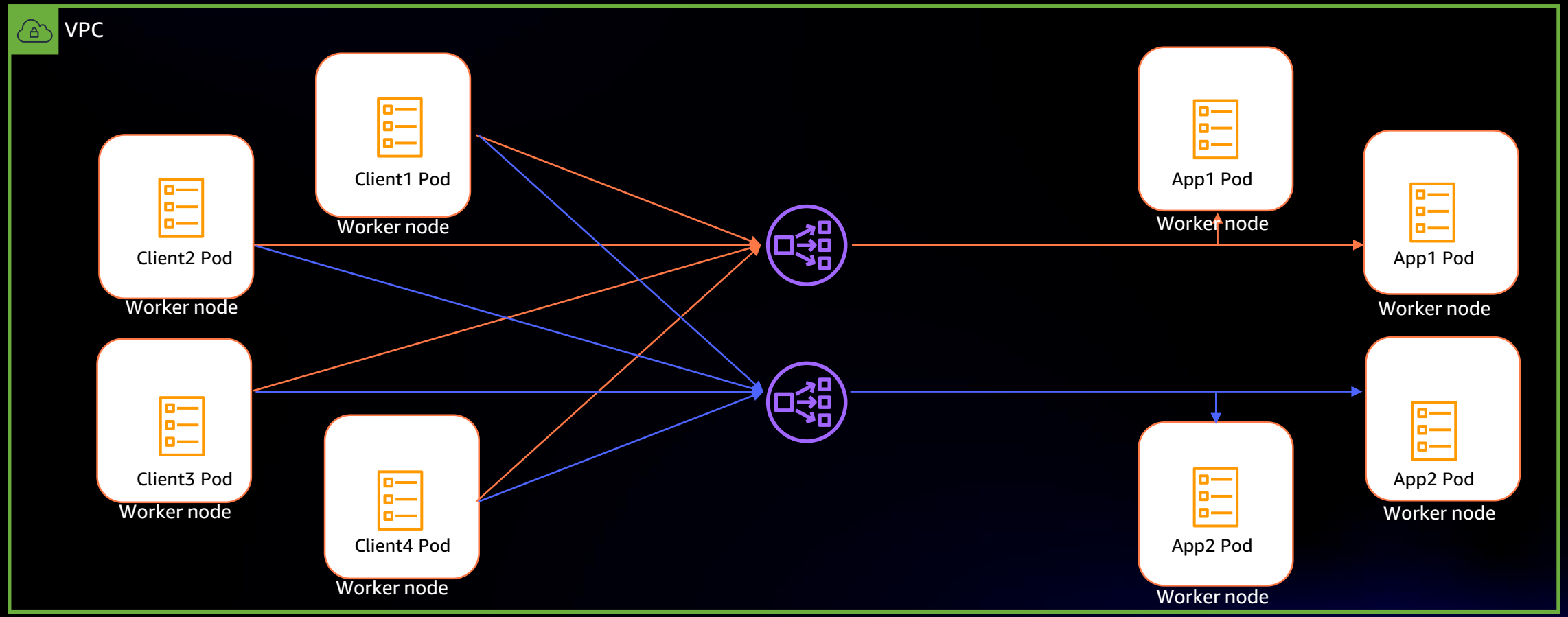
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service mesh

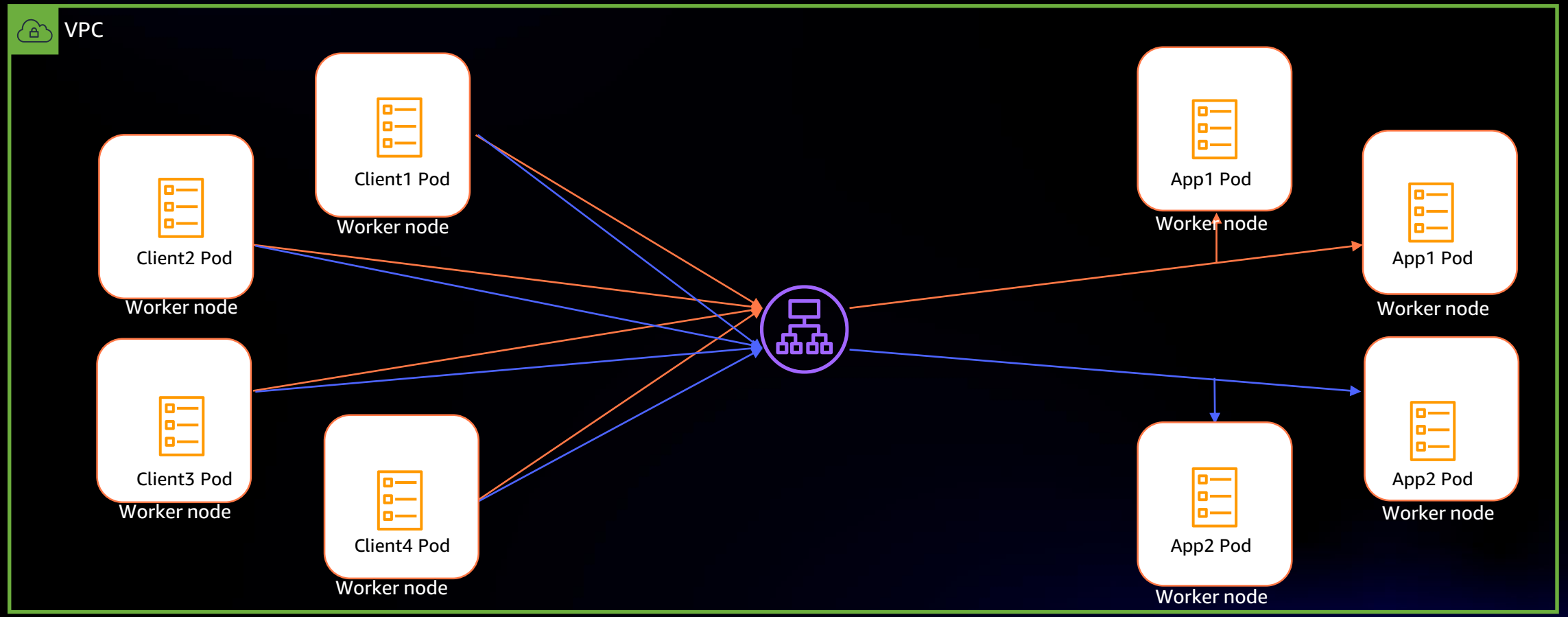
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Network Load Balancer traffic flow in EKS



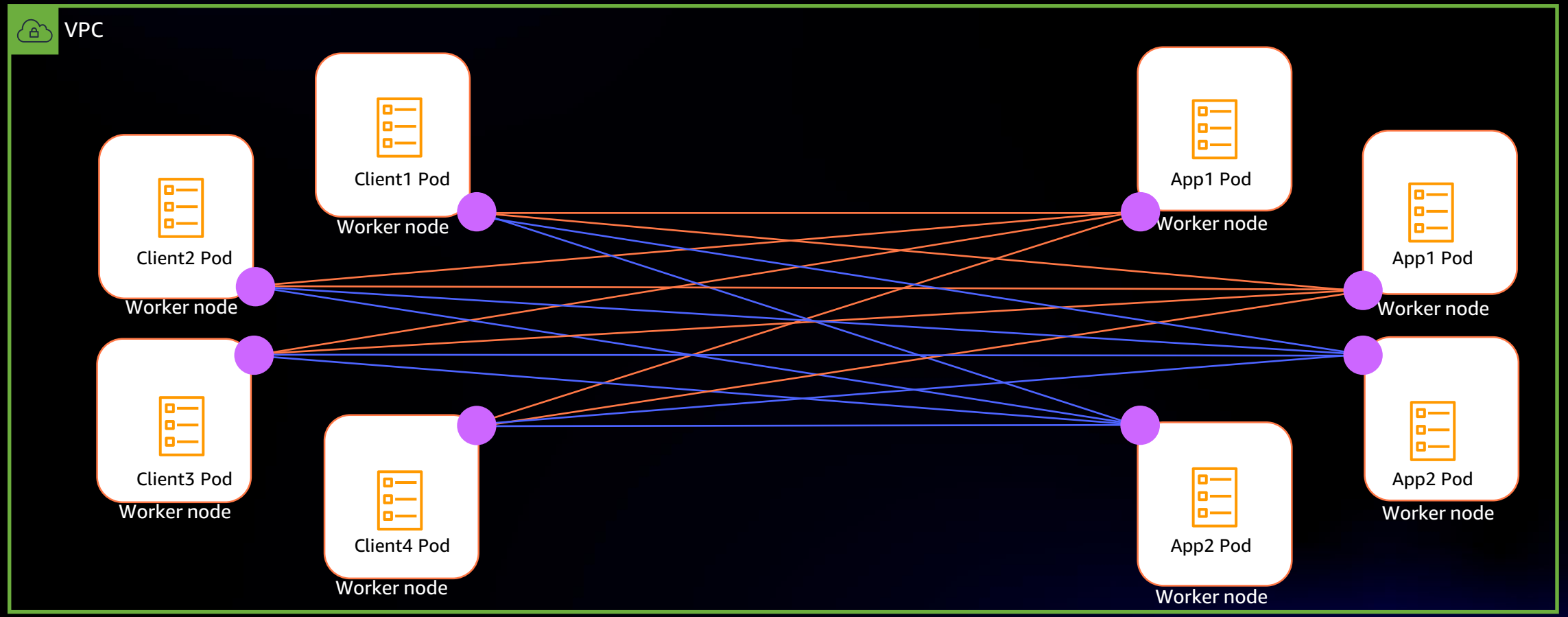
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application Load Balancer traffic flow in EKS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

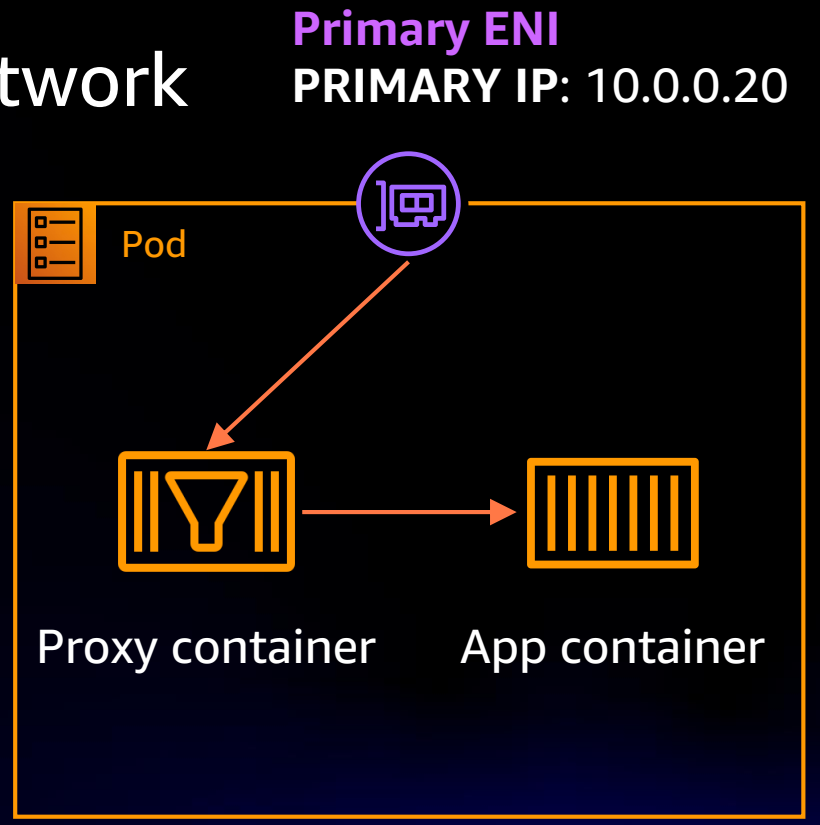
Service mesh traffic flow in EKS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service mesh uses a proxy

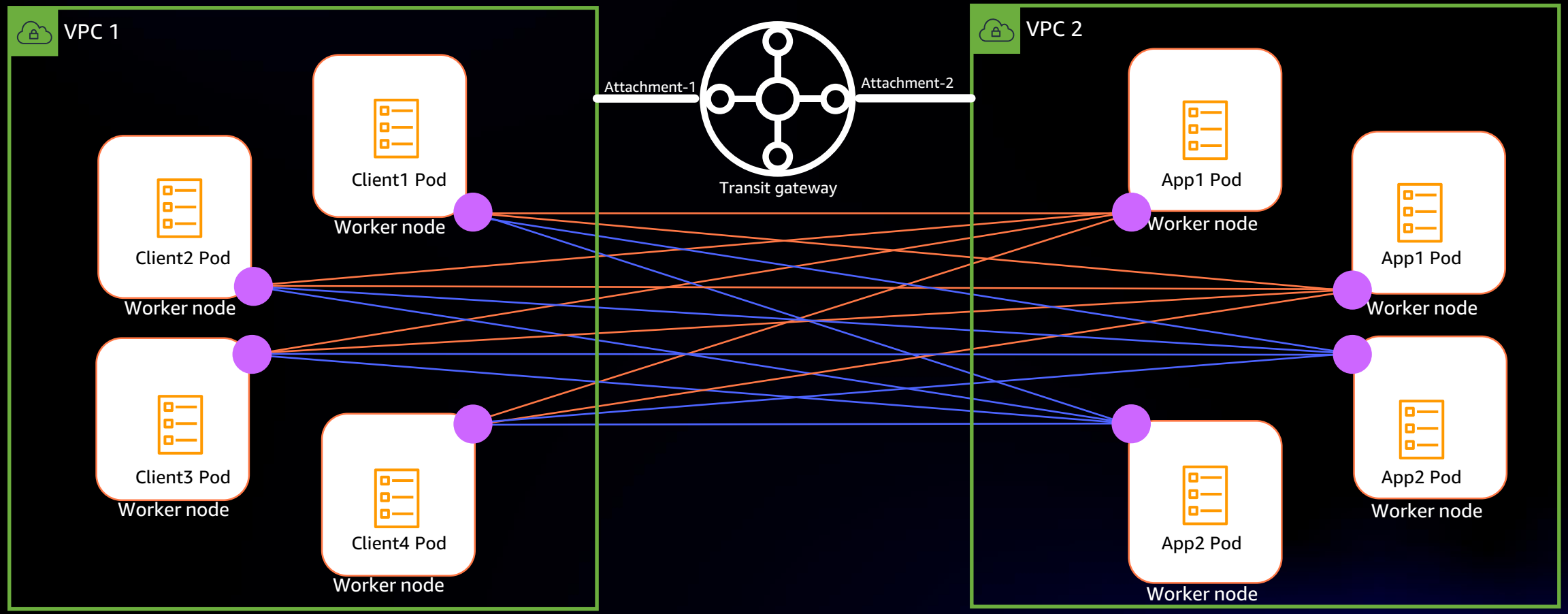
- Distributed proxy
  - 1 proxy per workload instance (pod)
- Proxy container and app container share network
  - Traffic routed to proxy first (via iptables)
- Proxy injected into pod
  - No changes to original spec
- Container-to-container communication over localhost
- Not involved in underlying network



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service mesh traffic flow in EKS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.