Securing Connections to Virtual Networks

Configuring VPNs

Virtual Network Gateways

Provide an endpoint for connections from on-premises locations to the Azure network

Contains two or more VMs in a special gateway subnet

Contain routing tables

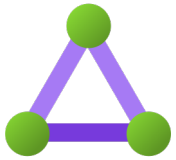
The VMs should not be configured directly

Gateway Types

Point-to-site

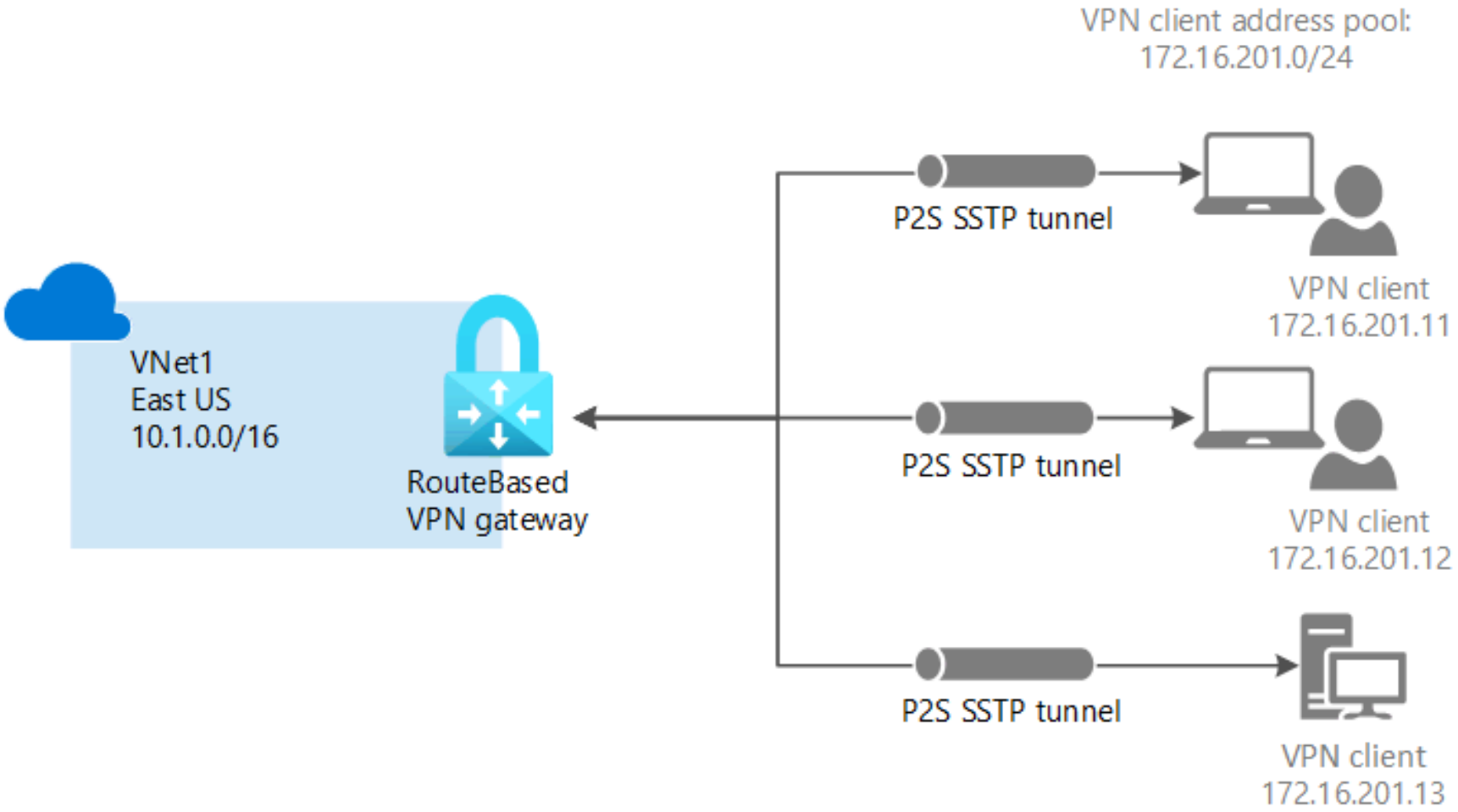
Site-to-site

VNet-to-VNet

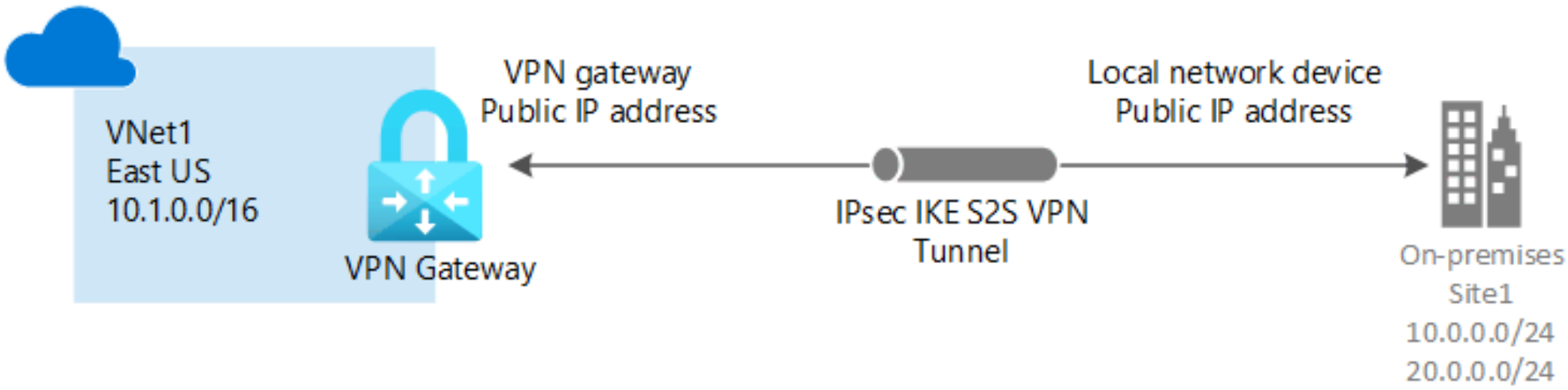


ExpressRoute

P2S VPN Topology



Site-to-site VPN Topology



VPN Gateways

Point-to-site

Clients do not require a VPN device or a public-facing IP address

Supports OpenVPN, SSTP and IPSec

Supports certificate, Azure AD and RADIUS authentication

Supports route-based routing

Supports active-passive connection resiliency

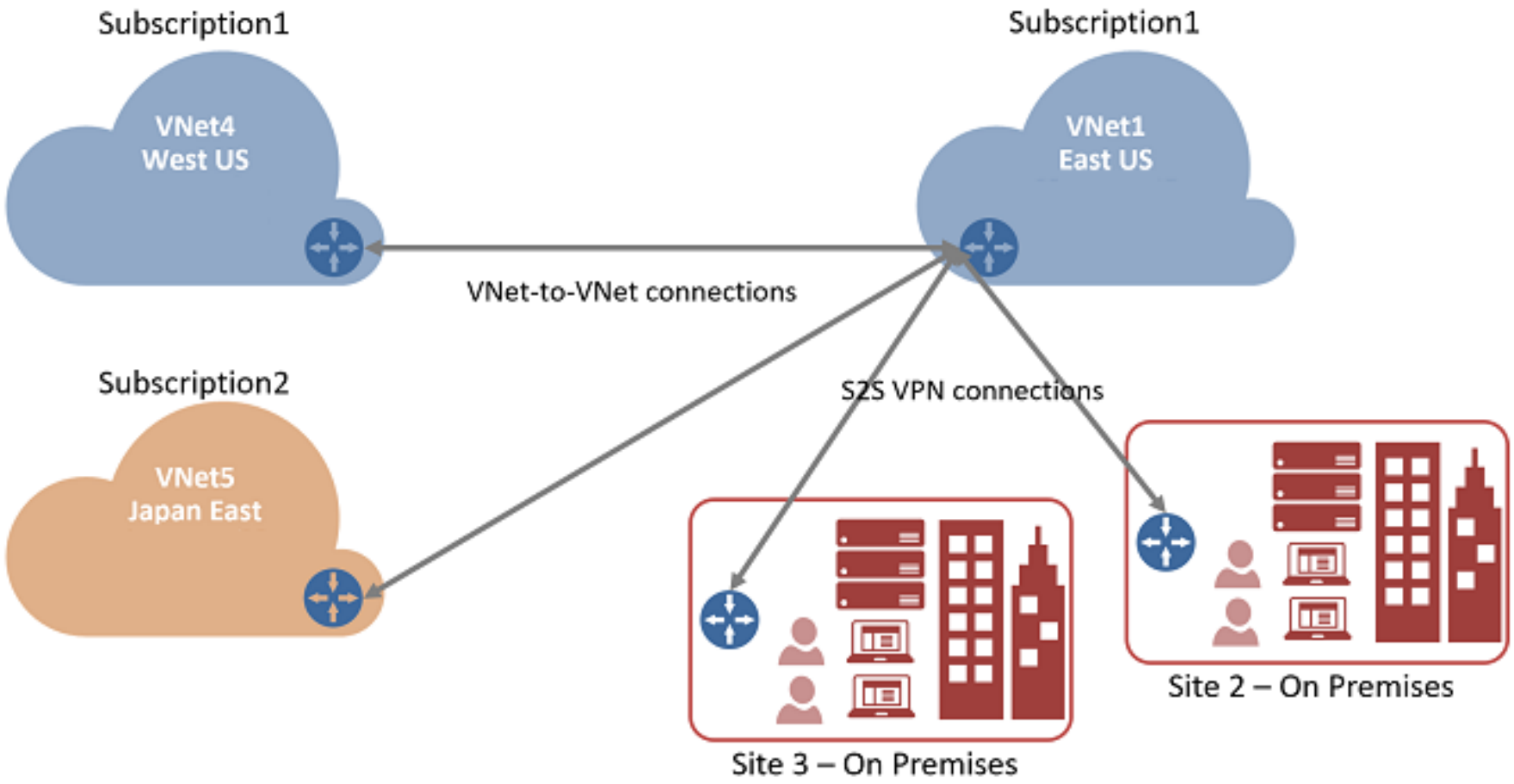
Site-to-site

Supports IPSec

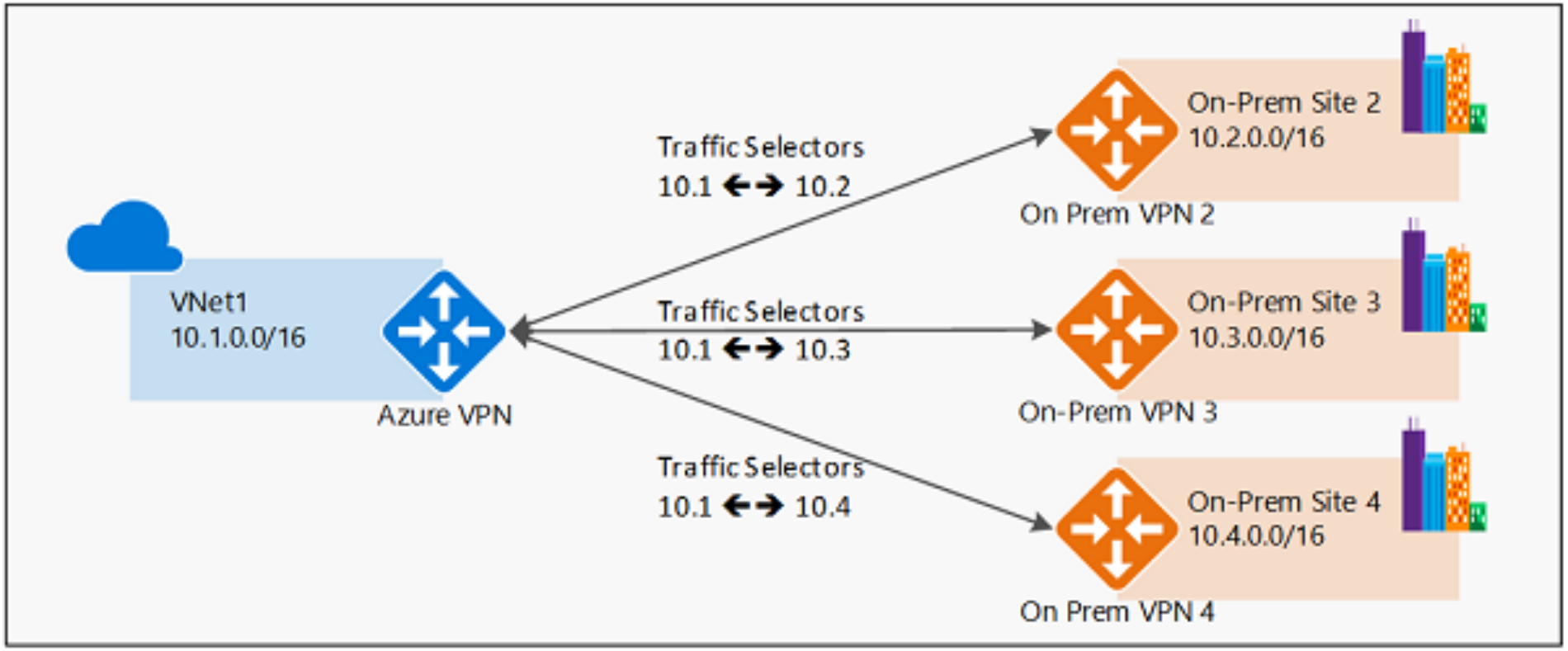
Supports policy-based and route-based routing

Supports active-passive and active-active connection resiliency

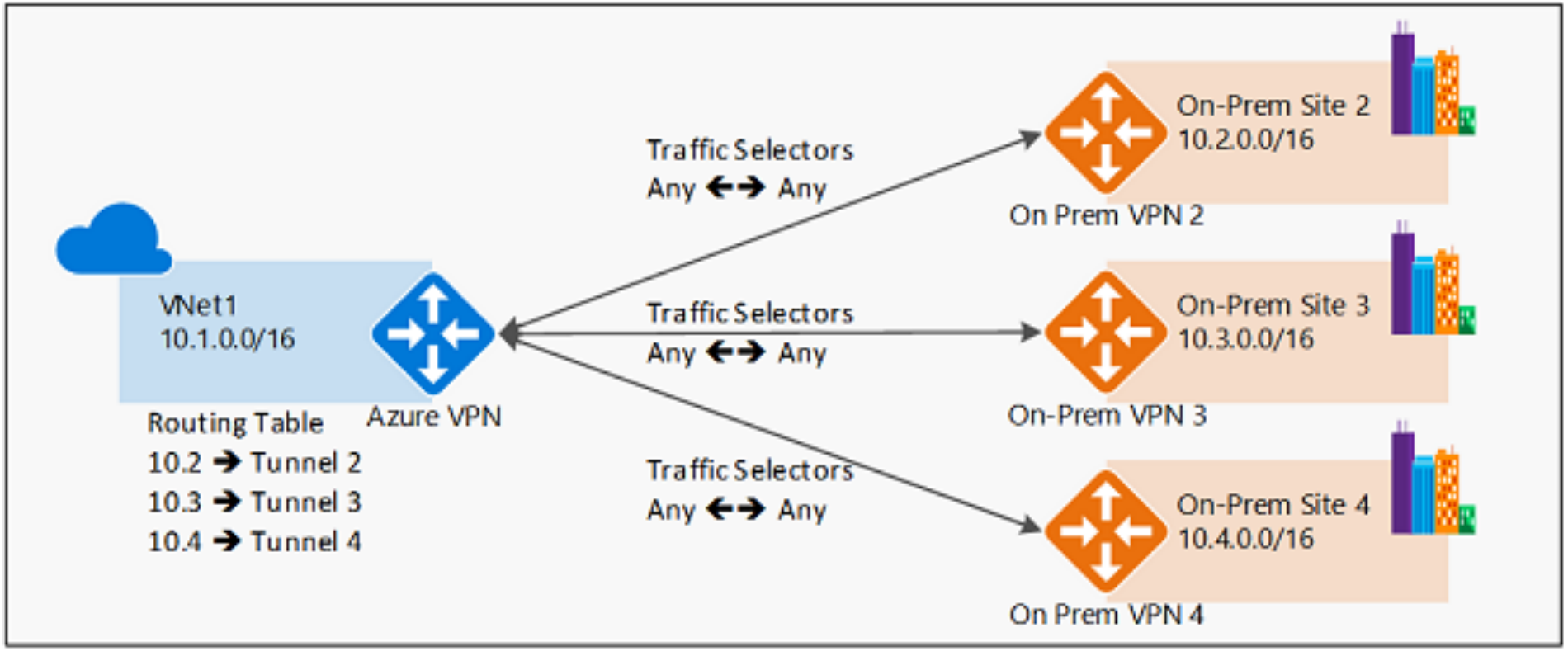
VNet-to-VNet VPN Topology



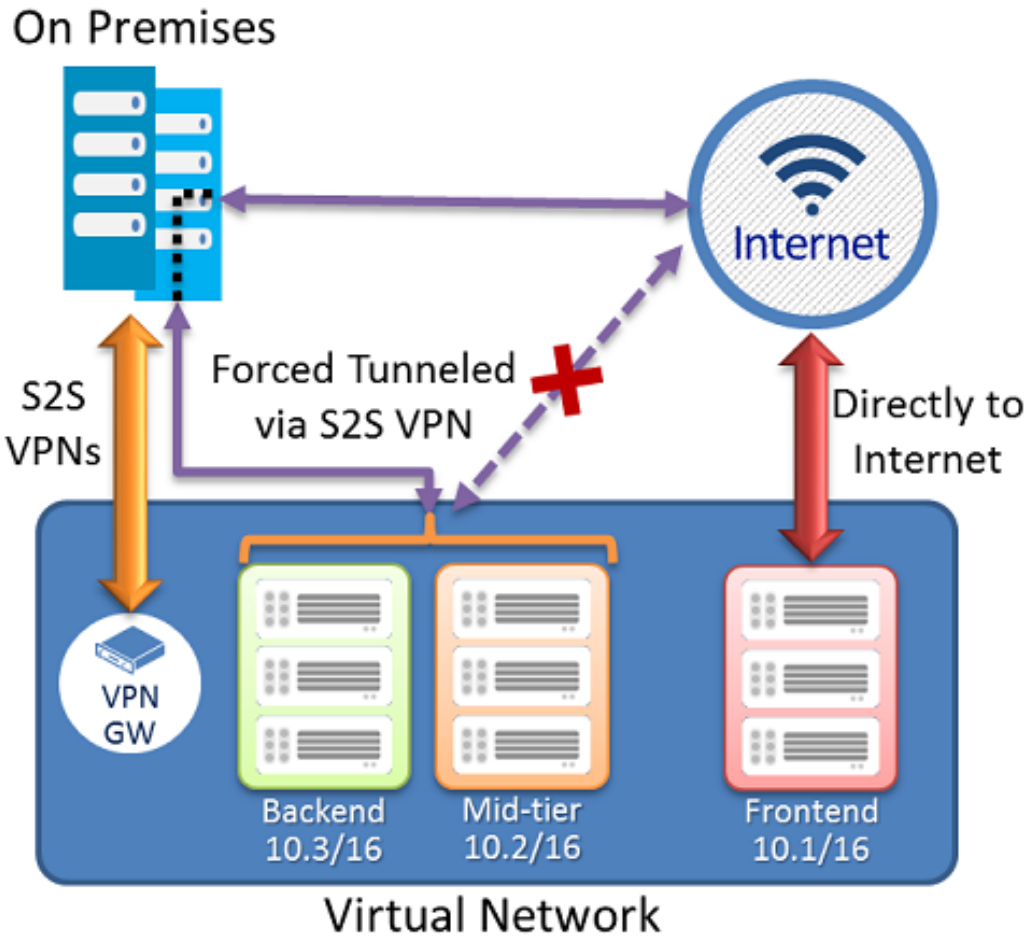
Policy-based Routing Methods



Route-based Routing Methods



Forced Tunneling



VPN Gateway Design Considerations

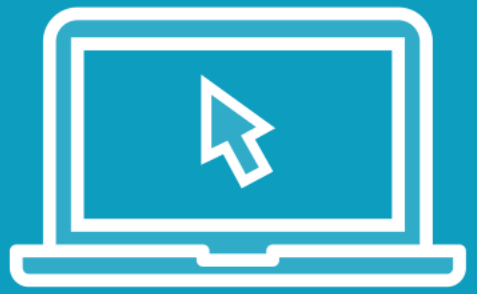
Subnets cannot overlap

IP addresses must be unique

The gateway subnet must be called ‘GatewaySubnet’

Only 1 VPN gateway is allowed per virtual network

Demo

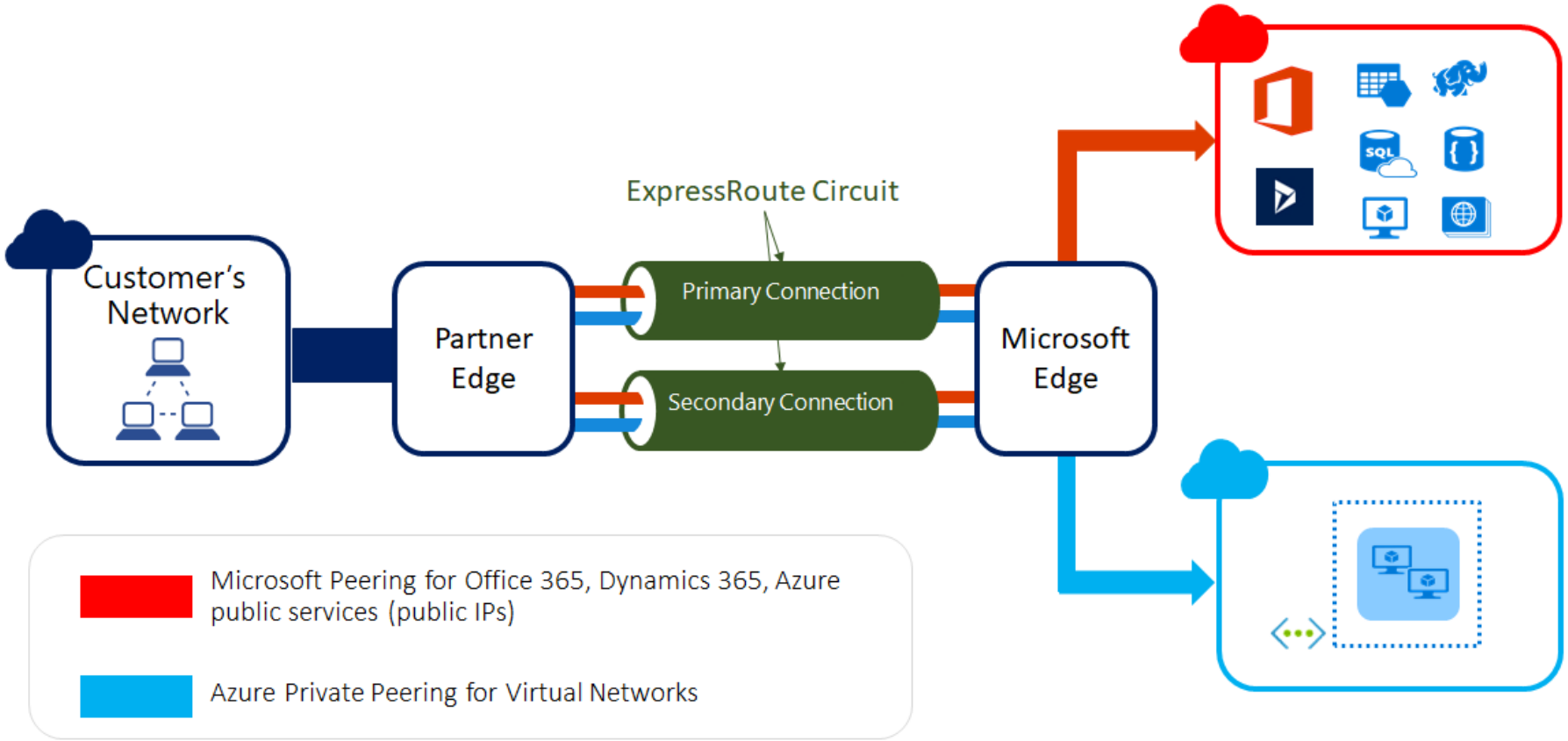


Configure a P2S VPN connection

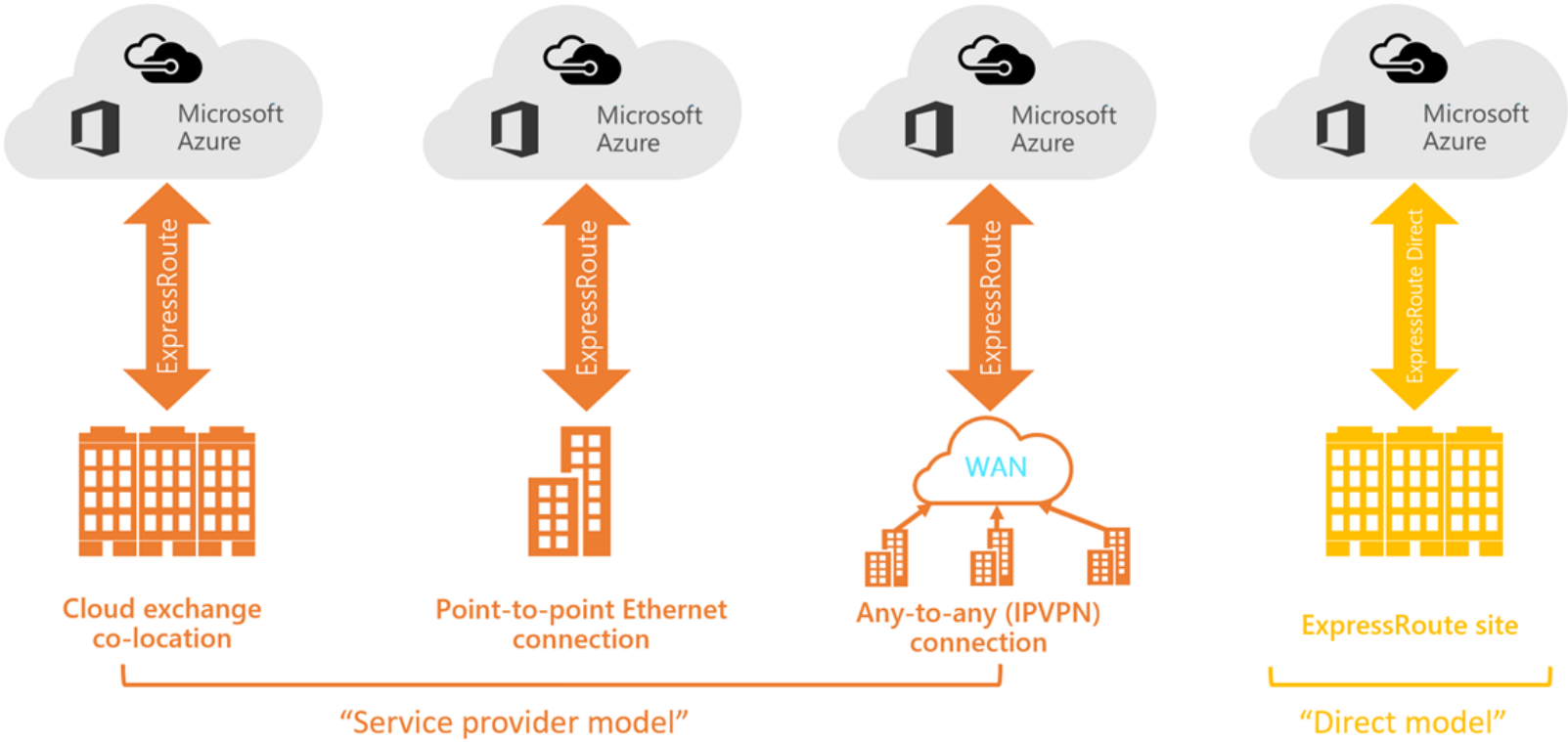
Configure a S2S VPN connection

Azure ExpressRoute

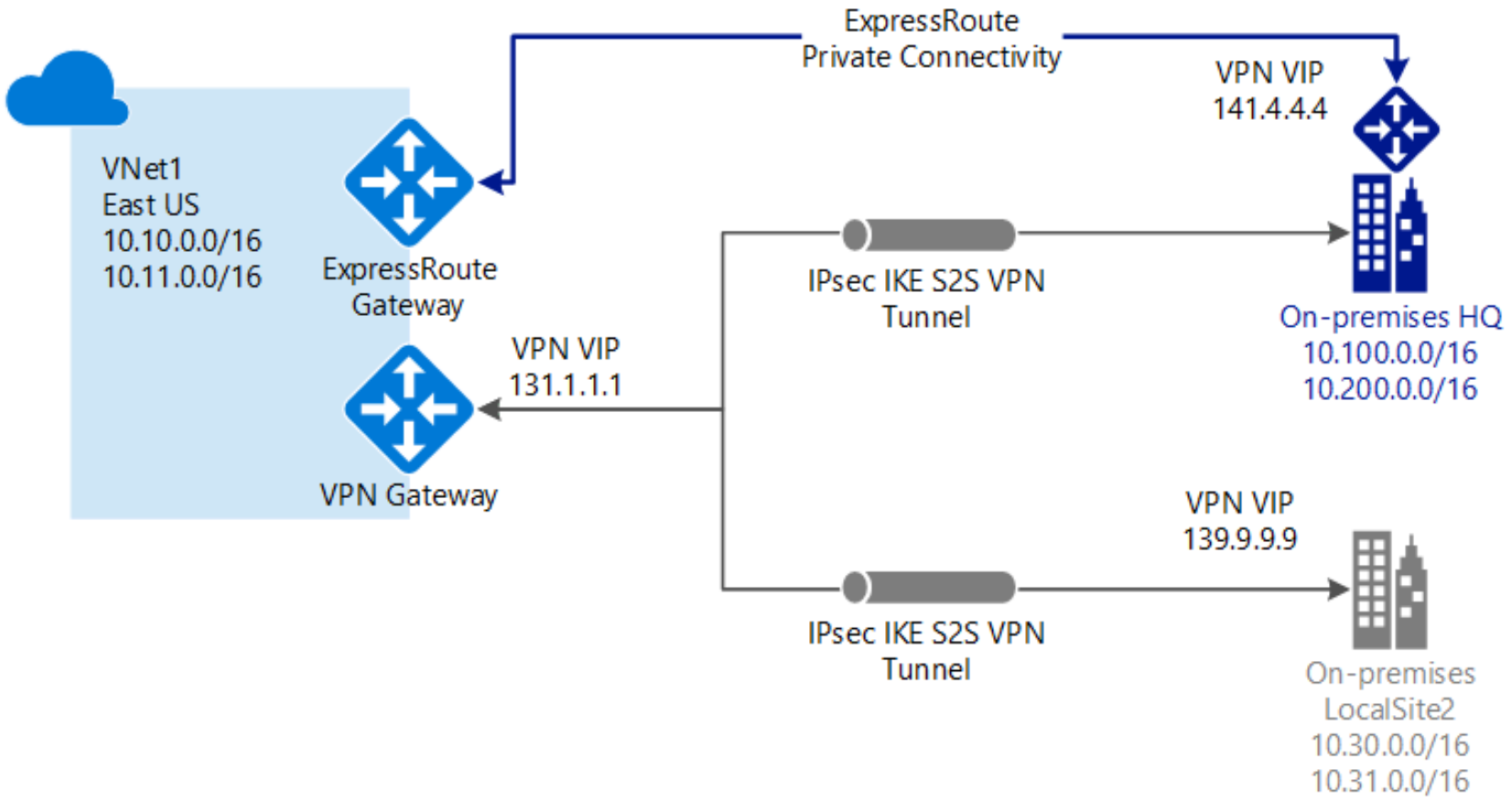
Azure ExpressRoute



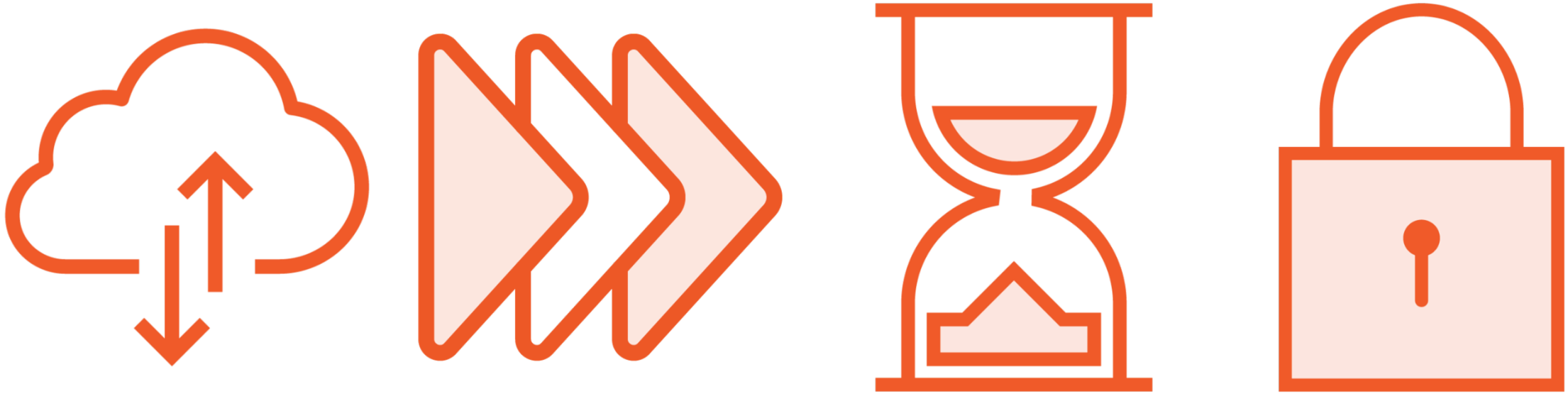
Connectivity Models



ExpressRoute & Site-to-site VPNs



Benefits of Azure ExpressRoute



More reliable

Faster

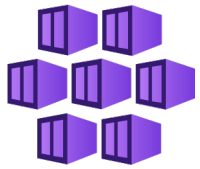
More Consistent Latencies

More Secure

Azure Kubernetes Service

Prerequisite Knowledge

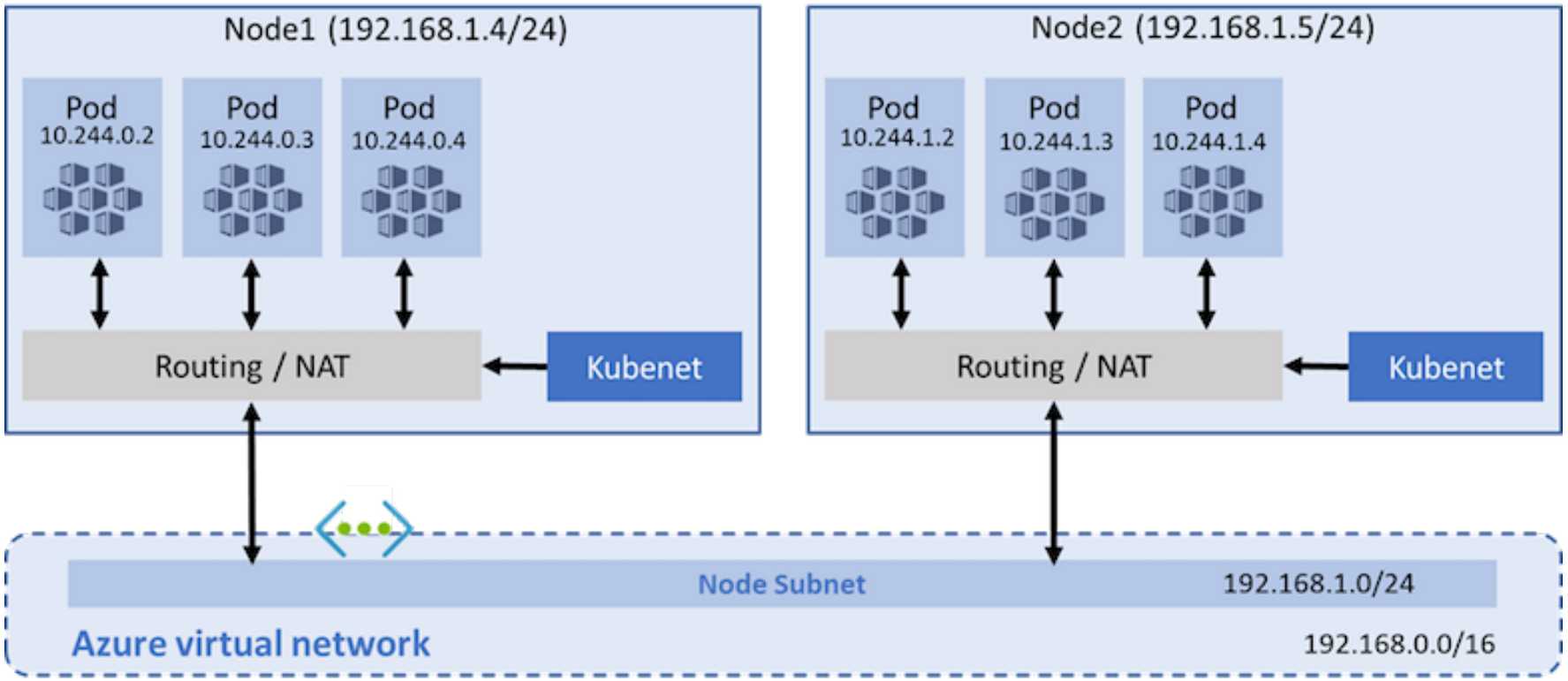
https://bit.ly/kubernetes-general-intro



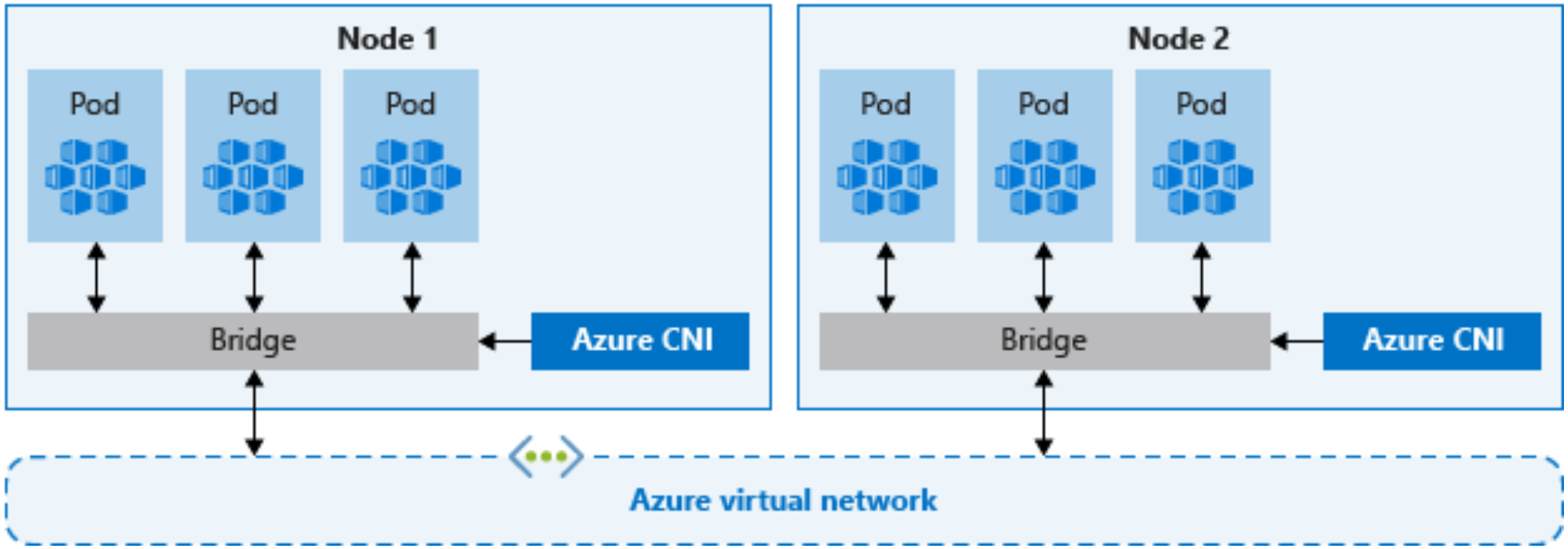
http://bit.ly/microsoft-kubernetes-basics

http://bit.ly/core-concepts-aks

Kubenet


Node1 (192.168.1.4/24)
Pod
10.244.0.2
Pod
10.244.0.3
Pod
10.244.0.4
Routing / NAT
Kubenet
Node2 (192.168.1.5/24)
Pod
10.244.1.2
Pod
10.244.1.3
Pod
10.244.1.4
Routing / NAT
Kubenet
Node Subnet
192.168.1.0/24
Azure virtual network
192.168.0.0/16

Azure CNI


Node 1
Pod
Pod
Pod
Bridge
Azure CNI
Node 2
Pod
Pod
Pod
Bridge
Azure CNI
Azure virtual network

Network Models Comparison

Kubenet

Pods cannot be accessed by default

Pods can be accessed using load balancers

AKS master manages network resources

Conserves IP address space

Azure CNI

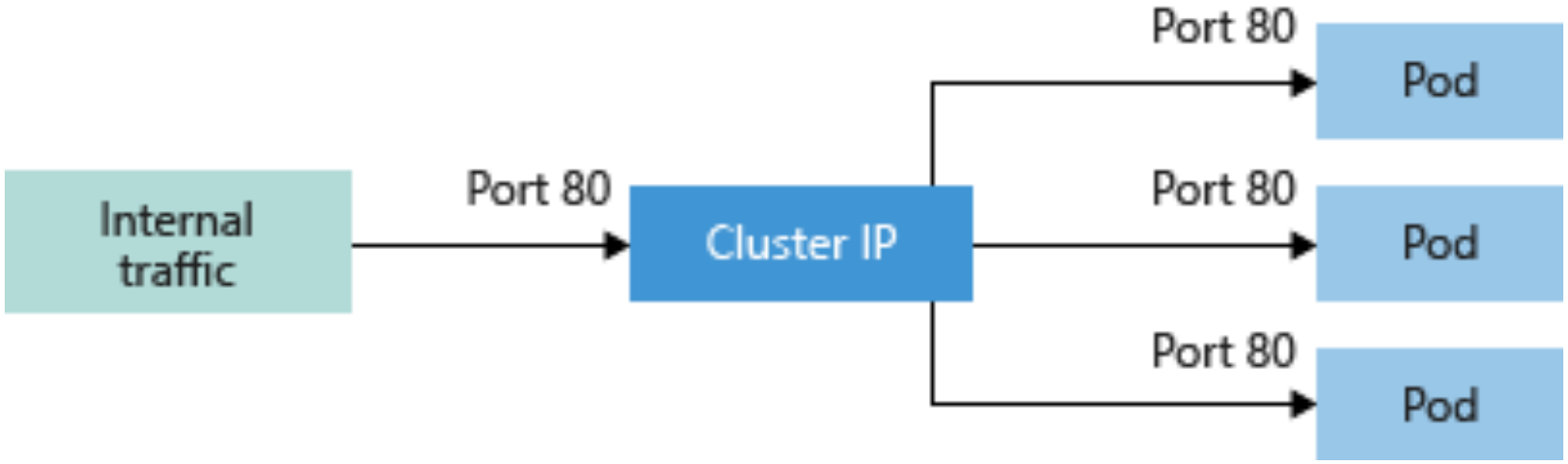
Pods get full virtual network connectivity

Pods can be reached via their private IP address from connected networks

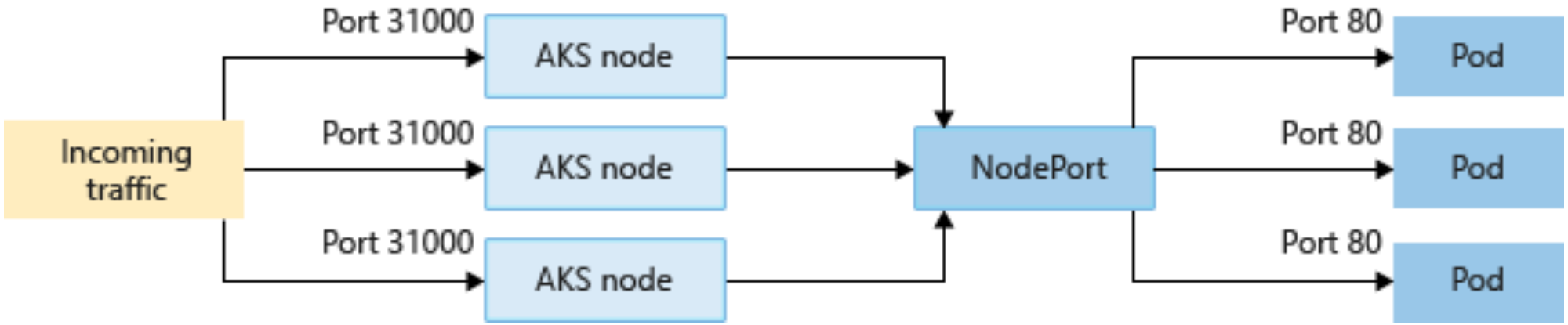
Network resources managed independently

Requires more IP address space

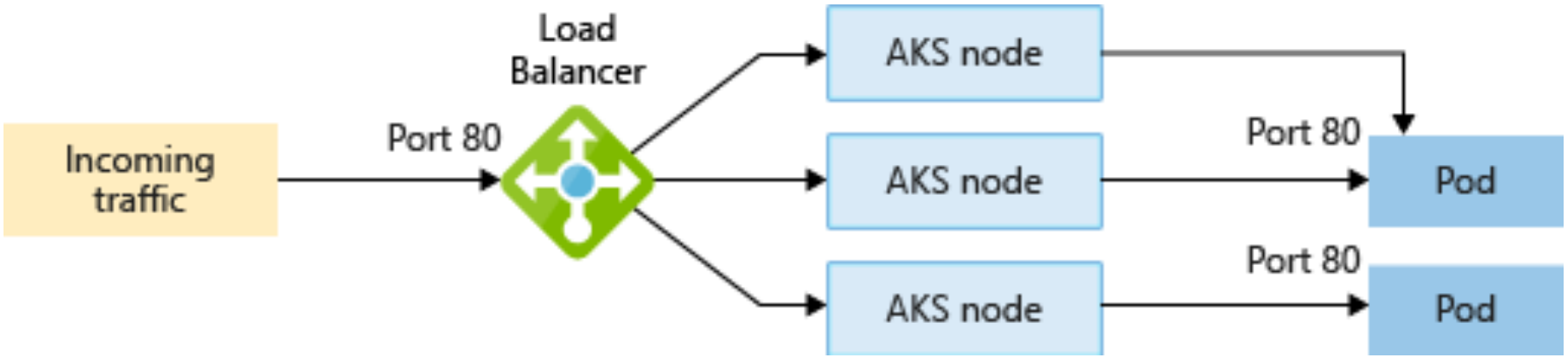
Cluster IP


Internal traffic
Port 80
Cluster IP
Port 80
Pod
Port 80
Pod
Port 80
Pod

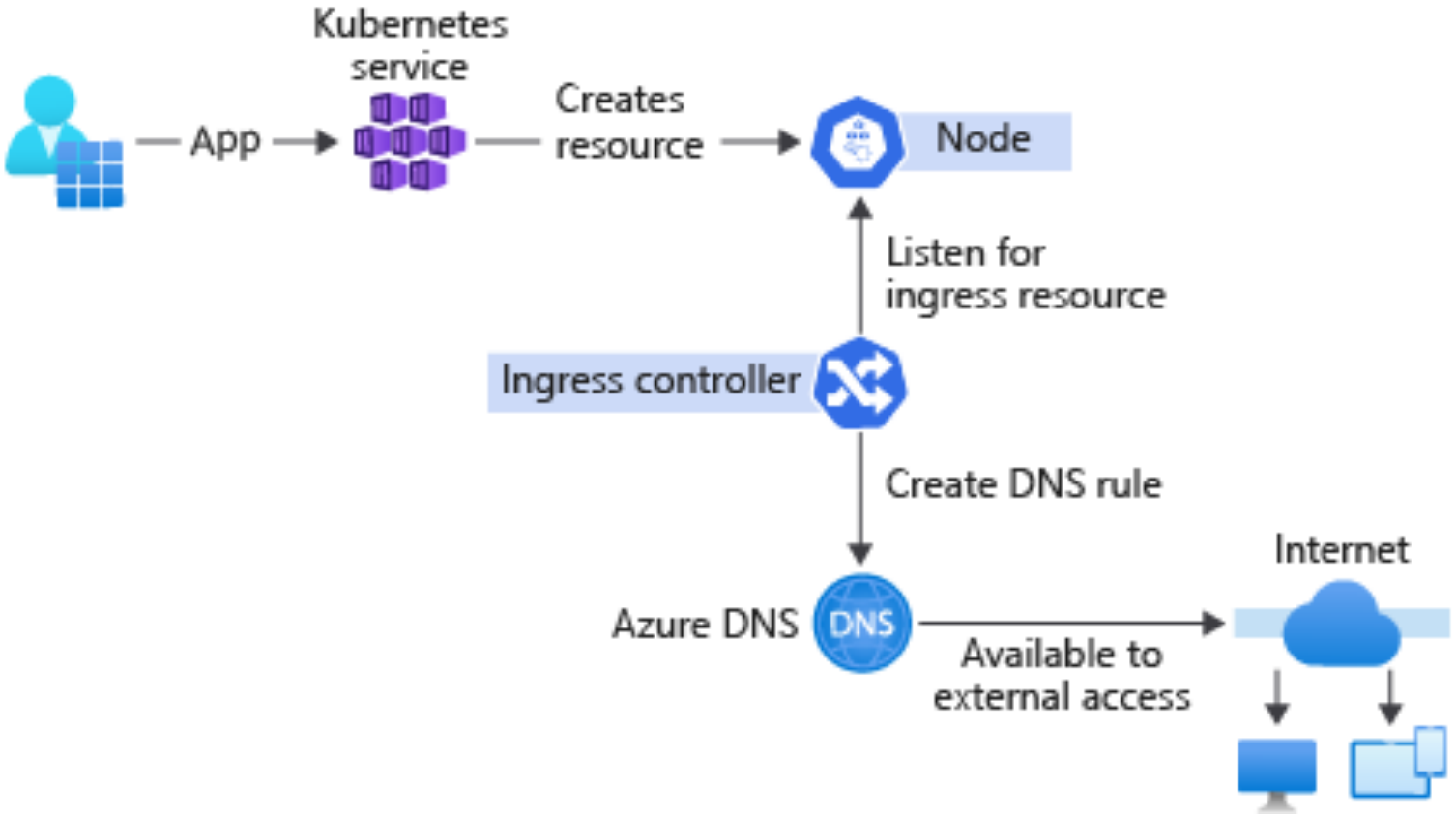
Node Port


Incoming traffic
Port 31000
AKS node
Port 31000
AKS node
Port 31000
AKS node
NodePort
Port 80
Pod
Port 80
Pod
Port 80
Pod

Load Balancer



Ingress Controllers



Ingress Resources

Ingress routes are defined in an ingress deployment manifest file

YAML

```
rules:
  - host: example.com
        http:
      paths: - path: /admin
          backend:
                      serviceName: admin-site
                      servicePort: 80
```

Network Policies

YAML

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: backend-policy
  namespace: development
spec:
  podSelector:
    matchLabels:
      app: webapp
      role: backend
  ingress: []
```

Allow inbound traffic based on a pod label

YAML

```yaml
ingress:
  - from:
    - namespaceSelector: {}
      podSelector:
        matchLabels:
          app: webapp
          role: frontend
```

Allow inbound traffic based on a pod label and namespace

YAML

```
ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          purpose: development
      podSelector:
        matchLabels:
          app: webapp
          role: frontend
```

Course Summary

Securing connections between VMs and Azure services

Securing web applications hosted in Azure

Securing connections to virtual networks

Microsoft Azure Security Engineer: Implement Advanced Network Security

Gabriel McNeilly
CLOUD & QA ENGINEER

@gmcneilly software-tester.io