



# Azure Internet Connectivity: Complete Documentation

☰ Status Not Started

## 1. Azure Internet Access Architecture Overview

Unlike AWS's explicit Internet Gateway, Azure provides internet connectivity through multiple integrated components. Azure's approach is more distributed and service-oriented.

## 2. Core Components for Internet Connectivity

### 2.1 Virtual Network (VNet)

- Azure's equivalent of AWS VPC
- Contains subnets with configurable IP ranges
- Supports both IPv4 and IPv6
- **Default System Routes** automatically provide internet connectivity

### 2.2 Public IP Addresses

#### Types of Public IPs:

##### 1. Basic SKU:

- Static or dynamic assignment
- Open inbound access by default
- Not zone-redundant

##### 2. Standard SKU:

- Static only
- Secure by default (no inbound unless explicitly allowed)
- Zone-redundant capabilities
- Integration with Azure Standard Load Balancer

## 2.3 NAT Gateway

- Managed service for outbound-only internet connectivity
- Provides up to 64,000 concurrent flows per IP address
- Supports up to 16 public IP addresses
- No SLA downtime for maintenance

### Key Features:

- **Static Outbound IPs:** Predictable egress IPs
- **On-demand SNAT Ports:** Dynamic port allocation
- **Idle Timeout Configurable:** 4-120 minutes

## 2.4 Azure Load Balancer

### Types:

1. **Public Load Balancer** (Internet-facing)
  - Frontend with public IP
  - Distributes inbound internet traffic
2. **Internal Load Balancer** (Private)
  - Frontend with private IP
  - Distributes traffic within VNet

# 3. System Routes & Routing Tables

## Default System Routes

Azure automatically creates system routes for:

text

Destination	Next Hop Type
-----	-----
0.0.0.0/0	Internet
10.0.0.0/8	VNetLocal
192.168.0.0/16	VNetLocal
172.16.0.0/12	VNetLocal
VirtualNetwork	VNetLocal

## User-Defined Routes (UDR)

Custom route tables to override system routes:

powershell

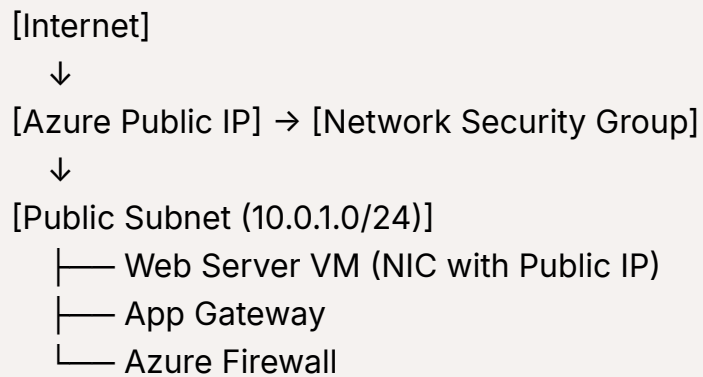
```
# Create route table
$routeTable = New-AzRouteTable `
  -Name "PrivateSubnetRouteTable" `
  -ResourceGroupName "MyRG" `
  -Location "EastUS"

# Add route to NAT Gateway
Add-AzRouteConfig `
  -Name "ToInternetViaNAT" `
  -AddressPrefix "0.0.0.0/0" `
  -NextHopType "VirtualAppliance" `
  -NextHopIpAddress "10.0.1.4" `
  -RouteTable $routeTable
```

## 4. Architecture Patterns

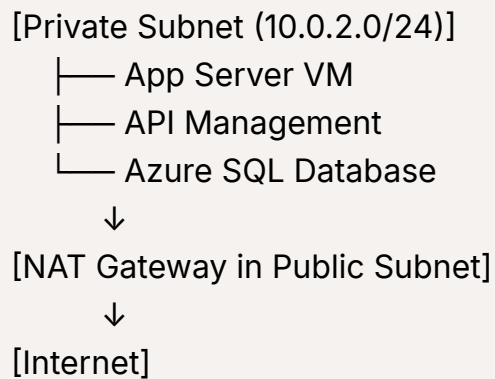
### Pattern 1: Public Subnet with Direct Internet Access

text



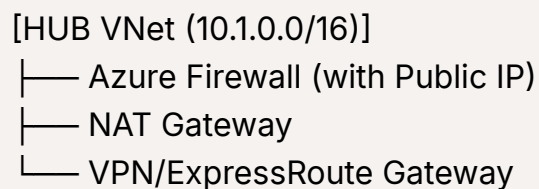
## Pattern 2: Private Subnet with NAT Gateway

text



## Pattern 3: Hub-Spoke with Shared Internet Egress

text



[SPOKE VNet 1 (10.2.0.0/16)]

- └─ Private Subnet → VNet Peering → Hub Firewall → Internet
- └─ App Services

[SPOKE VNet 2 (10.3.0.0/16)]

- └─ Private Subnet → VNet Peering → Hub Firewall → Internet
- └─ Azure SQL Managed Instance

## 5. Use Cases & Implementation Scenarios

### Use Case 1: E-commerce Application

text

Requirements:

- Public-facing web tier
- Secure backend processing
- PCI-DSS compliance
- High availability

Architecture:

[Public Subnet]

- └─ Application Gateway (WAF enabled)
- └─ Frontend VMs with Public IPs
- └─ CDN Endpoint

[Private Subnet - App Tier]

- └─ App Service Environment
- └─ API Management (Internal)
- └─ Service Fabric Cluster

[Private Subnet - Data Tier]

- └─ Azure SQL (Private Endpoint)
- └─ Redis Cache
- └─ Storage Accounts (Private Endpoint)

## Use Case 2: Hybrid Cloud Connectivity

text

Requirements:

- On-premises to Azure connectivity
- Controlled internet egress
- Centralized security

Architecture:

[On-Premises]



[ExpressRoute/VPN] → [Azure VNet Gateway]



[DMZ Subnet]

|— Azure Firewall

|— NAT Gateway

|— Bastion Host



[Private Subnets]

|— Domain Controllers

|— File Servers

|— Line-of-Business Apps

## Use Case 3: Microservices with AKS

text

Requirements:

- Kubernetes cluster with outbound connectivity
- Ingress controller for inbound
- Service mesh internal traffic

Architecture:

[Public Subnet]

|— AKS Public Load Balancer

- └─ Application Gateway Ingress Controller
- └─ NAT Gateway for Node Pools

[Private Subnet]

- └─ AKS Private Cluster Nodes
- └─ Internal Load Balancer Services
- └─ Azure Container Registry (Private Endpoint)
- └─ Monitoring Services

## 6. Security Considerations

### Network Security Groups (NSG)

#### Public Subnet NSG Example:

json

```
{
  "securityRules": [
    {
      "name": "AllowHTTPInbound",
      "properties": {
        "protocol": "Tcp",
        "sourcePortRange": "*",
        "destinationPortRange": "80",
        "sourceAddressPrefix": "Internet",
        "destinationAddressPrefix": "VirtualNetwork",
        "access": "Allow",
        "priority": 100,
        "direction": "Inbound"
      }
    },
    {
      "name": "DenyAllInbound",
      "properties": {
        "protocol": "*",
```

```
"sourcePortRange": "*",
"destinationPortRange": "*",
"sourceAddressPrefix": "*",
"destinationAddressPrefix": "*",
"access": "Deny",
"priority": 4096,
"direction": "Inbound"
}
}
]
}
```

## Azure Firewall

- Stateful firewall as a service
- Built-in high availability
- Threat intelligence-based filtering
- FQDN filtering in network rules

## Private Endpoints

- Connect privately to PaaS services
- Eliminates public internet exposure
- Uses Private Link service

# 7. Cost Optimization Strategies

## Cost Components:

1. **Public IP Addresses:** Hourly cost + data transfer
2. **NAT Gateway:** Hourly + per GB processed
3. **Load Balancer:** Hourly + rule hours + data processed
4. **Data Transfer:** Egress charges vary by region



## Optimization Tips:

- Use **Basic SKU Public IPs** for dev/test
- Implement **NAT Gateway** for multiple resources
- Use **Azure Front Door** for global HTTP(S) optimization
- Implement **Caching** to reduce egress traffic

## Common Issues & Solutions:

Issue	Root Cause	Solution
<b>No outbound connectivity</b>	Missing NAT Gateway/Public IP	Attach NAT Gateway to subnet
<b>Inbound connections fail</b>	NSG blocking traffic	Check NSG rules and priorities
<b>High latency to internet</b>	Suboptimal routing	Use Azure Front Door or VPN
<b>SNAT port exhaustion</b>	Too many connections	Add more Public IPs to NAT Gateway