
# AWS CLOUD NETWORKING

# AWS Global Infrastructure Foundation


**AWS Accounts**:

- Complete resource isolation between accounts
- Separate billing and access controls
- Cross-account networking requires explicit configuration
- Multi-environment strategies (dev/staging/prod)

**Key Concept**: Everything starts with the account boundary

# Geographic Distribution: Regions


**AWS Regions**:

- Physically separated geographic areas
- Independent infrastructure clusters
- 30+ regions globally (us-east-1, eu-west-1, ap-southeast-1)
- Each region contains multiple Availability Zones
- Data sovereignty and compliance boundaries
- Latency optimization through geographic proximity

**Why Multiple Regions?**

- Disaster recovery and high availability
- Regulatory and/or compliance requirements
- Performance optimization for global users

# Fault Tolerance: Availability Zones


**Availability Zones (AZs)**:

- Isolated data centers within a region
- Usually 3-6 AZs per region (a, b, c, d)
- Physically separate buildings with independent power/cooling
- High-speed, low-latency network connections between AZs
- Designed for fault isolation

**Best Practice**:

- Distribute resources across multiple AZs
- Database replicas in different AZs
- Load balancers spanning multiple Azs
- Auto Scaling Groups across AZs

# Virtual Private Cloud (VPC)

**VPC**:

- Isolated network environment within a region
- Private IP address space (RFC1918), optional public IPv6
- Spans all Availability Zones in a region
- Complete control over network configuration
- Default and custom (additional) VPCs available

**VPC Characteristics**:

- CIDR block /16 (or IPv6 /56) defines total IP range
- Cannot change CIDR after creation
- Multiple VPCs per region possible
- VPCs are region-specific, AZ-spanning

# Network Segmentation: Subnets


**Subnets**:

- Segments within your VPC
- Each subnet exists in exactly one Availability Zone
- Public subnets: Direct internet access via Internet Gateway
- Private subnets: No direct internet access, access via NAT Gateway
- CIDR blocks must be subsets of VPC CIDR

**Subnet Planning Example**:

```
VPC: 10.0.0.0/16
├── Public Subnet A: 10.0.1.0/24 (AZ-a)
├── Public Subnet B: 10.0.2.0/24 (AZ-b)
├── Private Subnet A: 10.0.11.0/24 (AZ-a)
    └── Private Subnet B: 10.0.12.0/24 (AZ-b)
```

# Internet Connectivity: Gateways

**Internet Gateway (IGW)**:

- Enables internet access for public subnets
- One IGW per VPC maximum
- Horizontally scalable and highly available
- Routes traffic between VPC and internet
- Not related to Availability Zones (AZ)

**NAT Gateway**:

- Enables outbound internet access for private subnets
- "Anchored" to public subnets
- AZ-specific: at least one per AZ for high availability (redundancy)
- **Traffic Flow**: Private Subnet → NAT Gateway → Internet Gateway → Internet
  - Note: NAT gateway charges for traffic *in* and *out*, on top of normal data transfer charges!
  - Alternatives: EC2 instance for NAT, e.g. https://fck-nat.dev/ or Egress-only gateway for IPv6

# Firewall Rules: Security Groups


**Security Groups**:

- Virtual firewalls for EC2 instances
- **Stateful**: Return traffic automatically allowed
- **Default Deny**: Only explicitly allowed traffic passes
- Applied at instance level, not subnet level
- Can reference other security groups (very handy!)
- Specific to one VPC

**Rules Structure**:

- **Inbound Rules**: Control incoming traffic
- **Outbound Rules**: Control outgoing traffic
- **Protocol**: TCP, UDP, ICMP, or All
- **Port Range**: Specific ports or ranges
- **Source/Destination**: IP ranges or security groups

# Security Group Traffic Patterns


**Ingress Traffic Examples**:

- HTTP: Port 80 from 0.0.0.0/0 (anywhere)
- SSH: Port 22 from specific IP ranges
- Database: Port 3306 from application security group
- Internal: All traffic from same security group

**Egress Traffic Examples**:

- HTTPS: Port 443 to 0.0.0.0/0 (software updates)
- Database: Port 5432 to database security group
- DNS: Port 53 to 0.0.0.0/0

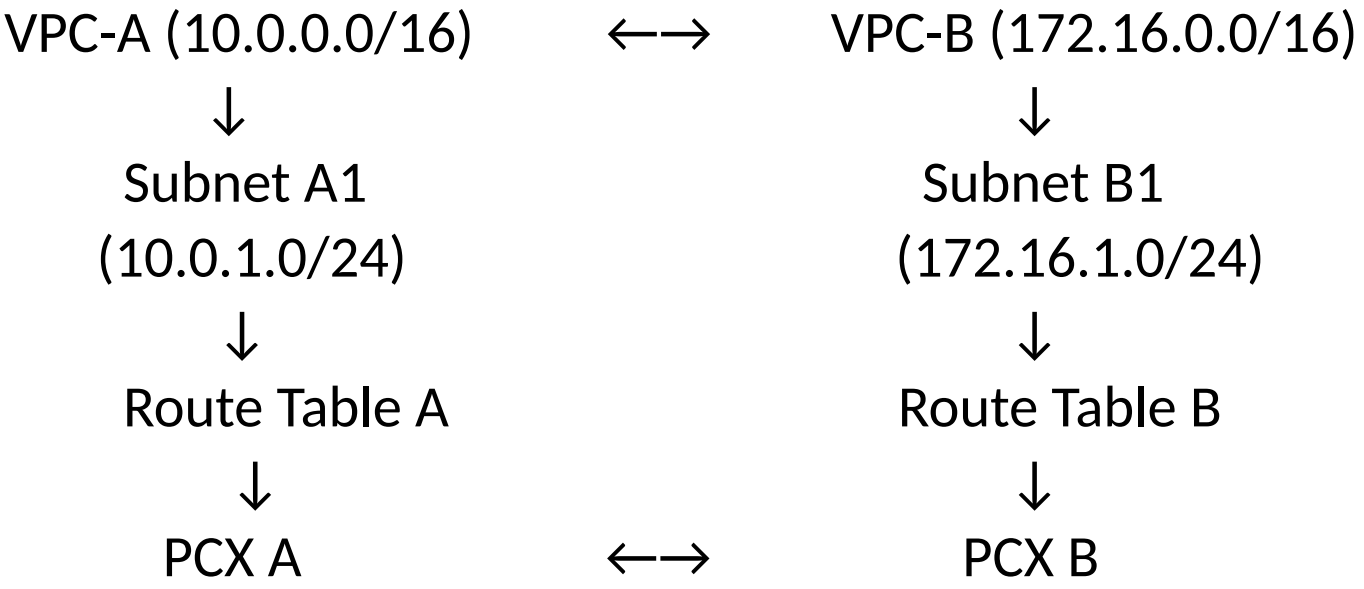
**Best Practice**:

- Least privilege principle
- Only allow necessary traffic
- Monitor and review rules regularly

# VPC Peering: Direct connections


- Direct network connection between two VPCs
- Creates a private communication channel using AWS backbone
- Works within same region or across different regions
- Each peering connection is between exactly two VPCs

# VPC Peering: How does it work?



# VPC Peering Config

**Create Peering Connection**: Establish the peering relationship

**Update Route Tables**: Add routes pointing to peer VPC CIDR blocks

- VPC-A route table: 172.16.0.0/16 → pcx-1
- VPC-B route table: 10.0.0.0/16 → pcx-2
-

**Security Groups**: Allow traffic from peer VPC security groups

**Test Connectivity**: Verify bidirectional communication

**Key Requirements**:

- Non-overlapping CIDR blocks between VPCs
- Mutual acceptance of peering connection
- Route table updates in both VPCs
- Security group rules to allow cross-VPC traffic
-

# Advanced Routing: Transit Gateway


**Transit Gateway**:

- Central hub for VPC connectivity
- Connects multiple VPCs and on-premises networks
- Simplifies complex network topologies
- Route table management for traffic control
- Scales to thousands of VPC attachments

**Use Cases**:

- Hub-and-spoke network architecture
- Centralized connectivity management
- Simplified VPC peering at scale
- Integration with AWS Direct Connect
- AWS IPsec VPN connections to external targets

# Hybrid Connectivity: VPN and Direct Connect


**AWS VPN Tunnels**:

- IPsec VPN connections over public internet
- Customer Gateway on-premises + Virtual Private Gateway in AWS
- A backup connectivity option
- Quick setup, variable bandwidth

**AWS Direct Connect**:

- Dedicated network connection from on-premises to AWS
- Consistent network performance
- Higher bandwidth options (1Gbps to 100Gbps)
- Reduced network costs for high-volume usage
- Private connectivity bypassing public internet

# Network Access Control: Additional Layers


**Network Access Control Lists (NACLs)**:

- Subnet-level firewall (vs Security Group instance-level)
- **Stateless**: Must define both inbound and outbound rules
- Numbered rules processed in order
- Default NACL allows all traffic

**Route Tables**:

- Control traffic routing within VPC
- Each subnet associated with one route table
- Define destinations and targets (IGW, NAT, VPC endpoints)
- Custom route tables for fine-grained control
- BGP available for dynamic routing, e.g. failover between VPN tunnels

# Putting It All Together: Network Architecture


**Typical Multi-Tier Architecture**:

```
Internet
    ↓
Internet Gateway
    ↓
Public Subnets (Load Balancers)
    ↓
Private Subnets (Application Servers)
    ↓
    Private Subnets (Databases)
```

**Key Principles**:

- Multiple security layers
- High availability: Resources across multiple AZs
- Scalability: Auto Scaling and load balancing
- Security: Least privilege access controls

# Network Design Best Practices

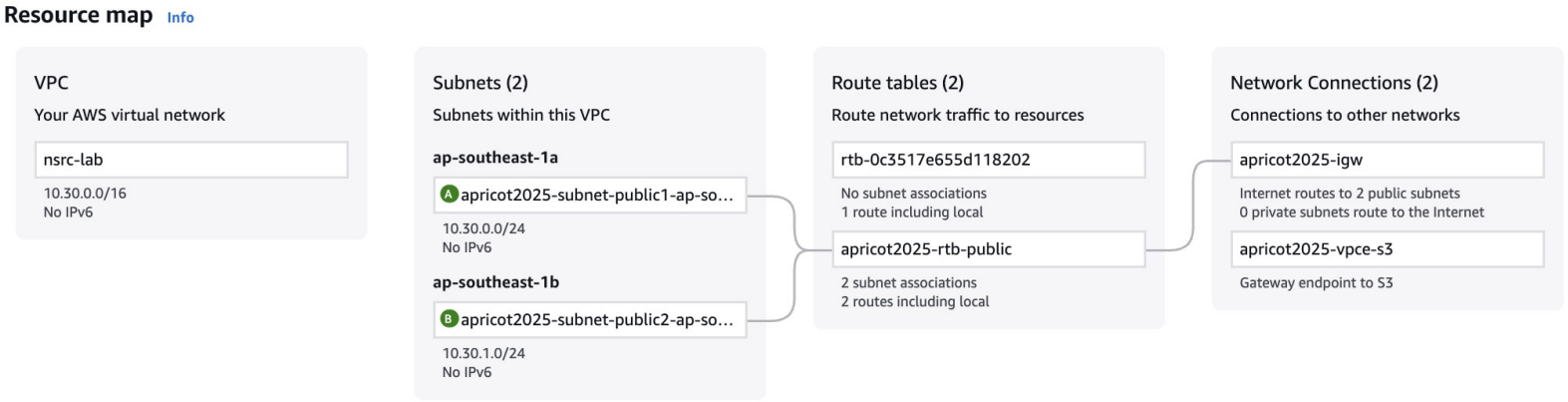

**Planning Considerations**:

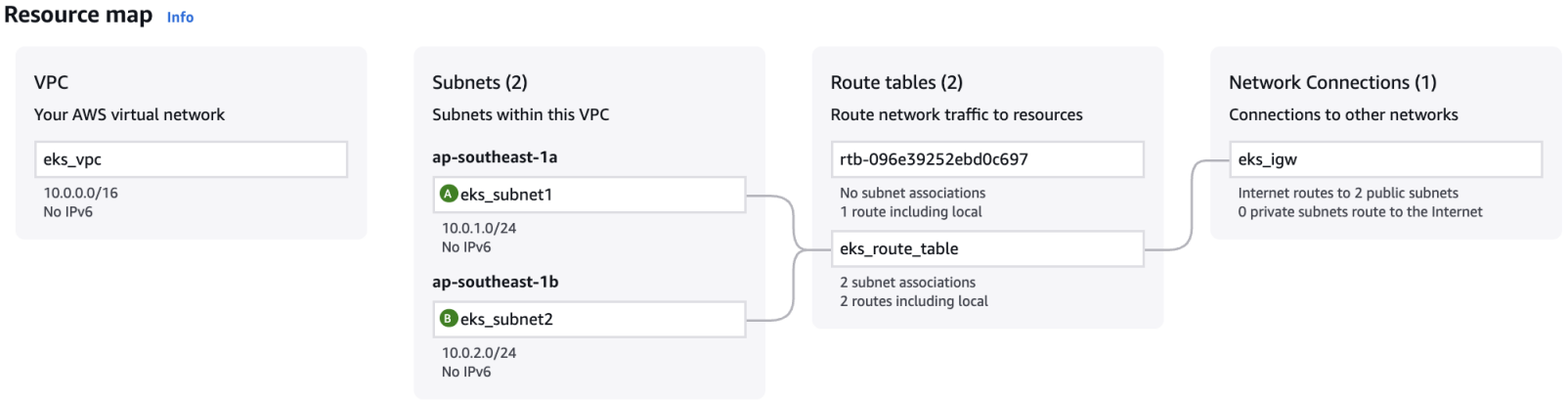
- **CIDR Planning**: Avoid overlapping ranges with on-premises
- **AZ Distribution**: Spread resources for fault tolerance
- **Security Segmentation**: Public/private subnet separation
  **Scalability**: Plan for growth in IP address space

**Operational Excellence**:

- Use Infrastructure as Code (Terraform)
- Document network architecture
- Monitor network performance and costs
- Regular security group audits

# Example VPC network diagrams




# Questions?