# AWS VPC Security Architecture

Complete Documentation - Week 1: AWS Security Services

* Overview
* VPC Layout
* IAM Roles
* Security Considerations
* Lessons Learned

## 🎯 Architecture Overview

This document covers the complete implementation of a secure AWS VPC infrastructure built as part of a cybersecurity learning journey. The architecture demonstrates **"defense in depth"** principles through multiple layers of security controls.

#### 🏗️ Architecture Goals

* Network isolation through custom VPC design
* Multi-layered security with NACLs and Security Groups
* Least privilege access through IAM roles
* Secure authentication using SSH keys and MFA
* No hardcoded credentials anywhere in the infrastructure

## 🌐 VPC Network Layout

### Core Network Design

**VPC CIDR Block**

**10.0.0.0/16**

Provides 65,536 IP addresses following enterprise standards and avoiding corporate network conflicts.

**Region Selection**

**us-east-1 (N. Virginia)**

Chosen for cost optimization, service availability, and compliance with data residency requirements.

### Subnet Architecture

| **Subnet Name** | **CIDR Block** | **Availability Zone** | **Type** | **Internet Access** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| public-subnet | 10.0.1.0/24 | us-east-1a | Public | ✅ Direct via IGW | Web servers, Load balancers, Bastion hosts |
| private-subnet | 10.0.2.0/24 | us-east-1b | Private | ❌ No direct access | Database servers, Internal applications |

### Network Routing Configuration

**Public Route Table (public-rt)**

* **Local route:** 10.0.0.0/16 → Local
* **Internet route:** 0.0.0.0/0 → Internet Gateway
* **Associated with:** public-subnet

**Private Route Table (Default)**

* **Local route:** 10.0.0.0/16 → Local
* **No internet route** (intentional isolation)
* **Future enhancement:** NAT Gateway for outbound-only access

### CIDR Block Calculation & Planning

#### 📊 CIDR Spreadsheet Analysis

Understanding CIDR notation is critical for proper network planning. Here's the mathematical breakdown of our chosen addressing scheme:

#### VPC CIDR Block: 10.0.0.0/16

| **Component** | **Binary Representation** | **Decimal Value** | **Explanation** |
| --- | --- | --- | --- |
| Network Address | 00001010.00000000.00000000.00000000 | 10.0.0.0 | Base network identifier |
| Subnet Mask (/16) | 11111111.11111111.00000000.00000000 | 255.255.0.0 | First 16 bits are network, last 16 are host |
| Available IPs | 2^16 = 65,536 | 65,536 addresses | Including network and broadcast |
| Usable IPs | 65,536 - 5 (AWS reserved) | 65,531 addresses | AWS reserves 5 IPs per subnet |

#### Subnet CIDR Breakdown

| **Subnet** | **CIDR** | **Address Range** | **Available IPs** | **Usable IPs** | **AWS Reserved IPs** |
| --- | --- | --- | --- | --- | --- |
| public-subnet | 10.0.1.0/24 | 10.0.1.0 - 10.0.1.255 | 256 | 251 | 10.0.1.0, .1, .2, .3, .255 |
| private-subnet | 10.0.2.0/24 | 10.0.2.0 - 10.0.2.255 | 256 | 251 | 10.0.2.0, .1, .2, .3, .255 |

#### AWS Reserved IP Addresses (Per Subnet)

| **Reserved IP** | **Public Subnet** | **Private Subnet** | **Purpose** |
| --- | --- | --- | --- |
| Network Address | 10.0.1.0 | 10.0.2.0 | Network identifier (cannot be assigned) |
| VPC Router | 10.0.1.1 | 10.0.2.1 | VPC router interface |
| DNS Server | 10.0.1.2 | 10.0.2.2 | Amazon-provided DNS |
| Future Use | 10.0.1.3 | 10.0.2.3 | Reserved for future AWS use |
| Broadcast | 10.0.1.255 | 10.0.2.255 | Network broadcast address |

#### CIDR Planning Considerations

**Growth Planning**

**Current Utilization:** 2 subnets using only 512 addresses (0.78% of VPC)

**Future Expansion:** Can create 254 additional /24 subnets

**Scalability:** Supports enterprise growth without renumbering

**Network Conflicts**

**Avoided Ranges:**

* 172.16.0.0/16 (common corporate networks)
* 192.168.0.0/16 (home networks)
* 169.254.0.0/16 (AWS link-local)

**Subnet Sizing Strategy**

**/24 subnets chosen because:**

* 251 usable IPs per subnet (adequate for most use cases)
* Easy to remember and calculate
* Aligns with traditional networking practices
* Allows for 256 subnets total

**Alternative Subnet Sizes**

**Comparison of options:**

* /20 = 4,091 usable IPs (oversized for demos)
* /24 = 251 usable IPs (chosen - right-sized)
* /26 = 59 usable IPs (too small for growth)
* /28 = 11 usable IPs (minimal, specialized use)

# CIDR Calculation Examples

# VPC: 10.0.0.0/16

# Binary: 00001010.00000000.xxxxxxxx.xxxxxxxx

# Network bits: 16, Host bits: 16

# Total addresses: 2^16 = 65,536

# Subnet: 10.0.1.0/24

# Binary: 00001010.00000000.00000001.xxxxxxxx

# Network bits: 24, Host bits: 8

# Total addresses: 2^8 = 256

# AWS Calculation for usable IPs:

# Total IPs - AWS Reserved = Usable

# 256 - 5 = 251 usable IP addresses per /24 subnet

**⚠️ CIDR Planning Mistakes to Avoid**

**Overlapping Subnets:** Ensure subnet CIDRs don't overlap (e.g., 10.0.1.0/24 and 10.0.1.128/25)

**Too Large VPC:** /8 networks can cause routing issues and are unnecessarily large

**Too Small Subnets:** /28 or smaller subnets may not provide enough IPs for auto-scaling

**Conflict with On-Premises:** Always check existing corporate network ranges before choosing CIDR

#### 🔒 Network Security Design Decisions

**Why separate availability zones?** Provides high availability and demonstrates proper subnet distribution for production environments.

**Why different CIDR blocks?** Clear network segmentation enables easy firewall rules and traffic monitoring.

**Why no internet access for private subnet?** Follows zero-trust principles and reduces attack surface for backend services.

**Why /24 subnets?** Provides optimal balance between address conservation and growth capacity while maintaining simple network calculations.

## 👤 IAM Security Implementation

### EC2 Instance Profile: ec2-s3-profile

Created a custom IAM role for EC2 instances to eliminate the security risk of hardcoded credentials while following the principle of least privilege.

#### EC2 Instance Permissions (Read-Only S3 Access)

{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "s3:GetObject", "s3:GetObjectVersion", "s3:ListBucket", "s3:ListBucketVersions", "s3:GetBucketLocation", "s3:GetBucketVersioning" ], "Resource": [ "arn:aws:s3:::\*", "arn:aws:s3:::\*/\*" ] } ] }

#### Permission Justification

**s3:GetObject & s3:GetObjectVersion**

Allows downloading files and specific versions from S3 buckets for application data and configuration files.

**s3:ListBucket & s3:ListBucketVersions**

Enables listing bucket contents and object versions for inventory and monitoring purposes.

**s3:GetBucketLocation & s3:GetBucketVersioning**

Provides metadata access for proper SDK functionality and regional optimization.

**Resource: "arn:aws:s3:::\*"**

Allows access to all buckets for flexibility without requiring policy updates for new resources.

#### ✅ Security Benefits of IAM Roles

* **Temporary credentials:** Automatically rotate every 6 hours
* **No hardcoded keys:** Eliminates credential storage on instances
* **Least privilege:** Only S3 read operations allowed
* **Audit trail:** All actions logged through CloudTrail
* **Revocable access:** Can be instantly disabled at role level

### Access Pattern Comparison

| **Access Method** | **Use Case** | **Permissions Scope** | **Credential Type** | **Security Level** |
| --- | --- | --- | --- | --- |
| EC2 Instance Role | Automated systems | Minimal (S3 read-only) | Temporary (6h rotation) | 🟢 High |
| IAM User (Admin) | Human administrators | Broader (testing/admin) | Long-term keys | 🟡 Medium |
| Root Account | Emergency only | Full account access | Password + MFA | 🔴 Critical |

## 🛡️ Security Considerations Applied

### Multi-Layered Security Controls (Defense in Depth)

#### Layer 1: Network ACLs (Subnet Level)

**NACL Name:** demo-nacl (Applied to public-subnet only)

**Type:** Stateless firewall - Must configure both inbound and outbound rules

#### Layer 2: Security Groups (Instance Level)

**SG Name:** demo-sg-restrictive

**Type:** Stateful firewall - Automatically allows return traffic

#### Layer 3: SSH Key Authentication

**Key Type:** RSA .pem format with 400 permissions

**Security:** Eliminates password-based attacks

### Port Security Configuration

#### Security Group Rules (Stateful)

| **Direction** | **Protocol** | **Port** | **Source/Destination** | **Purpose** | **Security Level** |
| --- | --- | --- | --- | --- | --- |
| Inbound | TCP | 22 | My.IP.Address/32 | SSH admin access | 🟢 Secure |
| Outbound | All | All | 0.0.0.0/0 | Return traffic (automatic) | 🟢 Secure |

**⚠️ Critical Security Issue Identified**

**NACL Rule 200:** Originally configured to allow inbound traffic on ports 1024-65535 from 0.0.0.0/0

**Risk:** Opens over 64,000 ports to the entire internet, creating massive attack surface

**Impact:** Port scanning vulnerability, potential for unintended service exposure

**Lesson Learned:** Security Groups are typically better for instance-level security due to their stateful nature

### Authentication & Access Control

#### 🔐 MFA Implementation

* Root account protected with virtual MFA device
* Administrative users require MFA for console access
* Breakglass accounts configured with mandatory MFA
* MFA enforced through IAM policies that deny actions without multi-factor authentication

#### SSH Key Security Practices

# Immediately set proper permissions after key creation chmod 400 demo-key.pem

# Store in secure location (not in project repositories) mv demo-key.pem ~/.ssh/

# Use full path for SSH connections ssh -i ~/.ssh/demo-key.pem ec2-user@PUBLIC-IP

# Verify key fingerprint before first use ssh-keygen -l -f ~/.ssh/demo-key.pem

### Regional Security Considerations

**Region: us-east-1 (N. Virginia)**

**Advantages:**

* Lowest cost for most AWS services
* Highest service availability
* Best performance for US East Coast users

**Security Implications**

**Considerations:**

* Data sovereignty compliance
* Latency impact on security controls
* Regional service dependencies

### Network Security Best Practices Applied

#### ✅ Security Controls Implemented

* **Network Segmentation:** Public/private subnet isolation
* **Minimal Open Ports:** Only SSH (22) from admin IP
* **Stateful Firewalls:** Security Groups for automatic return traffic handling
* **DNS Security:** Enabled DNS hostnames for proper service discovery
* **IP Whitelisting:** Restricted access to single administrative IP address

#### 🚨 Security Lessons Learned

* **NACL Complexity:** Stateless rules require careful planning for both directions
* **Return Traffic:** Must explicitly allow high-port ranges for stateless firewalls
* **Security Group Preference:** Stateful firewalls are typically more secure and manageable
* **Least Privilege:** Start with no access and add permissions as needed
* **Regular Review:** Security configurations must be regularly audited