# 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

#### 🔒 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.

## 👤 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