Windows Server Deployment Guide on AWS Cloud

Sou Chanrojame, Orn Pheakdey

November 28, 2025

Abstract

This document provides a comprehensive step-by-step guide for deploying various Windows Server roles and services on the AWS Cloud infrastructure, including configuration details for EC2 instances, security groups, and storage optimization.

Contents

1 Prerequisites

1.1 AWS Account Setup

- Active AWS account with appropriate permissions
- VPC configured with public and private subnets
- Security groups properly configured
- Key pairs created for RDP access
- IAM roles for EC2 instances

1.2 General Windows Server Launch Steps

1. Navigate to EC2 Dashboard in AWS Console
2. Click “Launch Instance”
3. Select Windows Server AMI (2019/2022 recommended)
4. Choose instance type based on workload
5. Configure instance details (VPC, subnet, IAM role)
6. Add storage as needed
7. Configure security groups
8. Review and launch with key pair

2 File Server

2.1 AWS Configuration

Instance Type: t3.medium or larger

Storage: EBS volumes with provisioned IOPS for performance

Security Group Ports: 445 (SMB), 139 (NetBIOS), 3389 (RDP)

2.2 Implementation Steps

1. Launch Windows Server EC2 Instance

Select Windows Server 2022 Datacenter and attach additional EBS volumes for file storage.

2. Install File Server Role

```
Install-WindowsFeature -Name FS-FileServer -IncludeManagementTools
Install-WindowsFeature -Name FS-DFS-Namespace, FS-DFS-Replication
```

3. Configure Storage

Initialize and format additional EBS volumes and create shared folders.

```
New-SmbShare -Name "SharedFiles" -Path "D:\Shares" -FullAccess "Domain\Admins" -
    ReadAccess "Domain\Users"
```

4. Enable Shadow Copies

```
Enable-ComputerRestore -Drive "D:\"
vssadmin resize shadowstorage /for=D: /on=D: /maxsize=20%
```

5. Configure AWS Backup

Create a backup plan for EBS volumes and set retention policies.

2.3 Best Practices

- Use AWS Storage Gateway for hybrid scenarios
- Implement Amazon FSx for Windows File Server for a managed solution
- Enable encryption at rest using AWS KMS
- Configure NTFS permissions and share permissions

3 Proxy Server (Caching, Control Access)

3.1 AWS Configuration

Instance Type: t3.medium

Security Group Ports: 8080, 3128 (proxy), 3389 (RDP)

3.2 Implementation Steps

1. **Launch Windows Server Instance**
2. **Install Proxy Server Software**
   - **Option A (WinGate):** Download/install WinGate and configure proxy settings.
   - **Option B (Squid):** Download Squid for Windows and configure `squid.conf`.
3. **Configure Proxy Settings**

```
# Example configuration for basic proxy
netsh winhttp set proxy proxy-server="localhost:8080" bypass-list="*.local"
```

4. **Set Up Caching:** Configure cache directory on separate EBS volume and set policies.
5. **Access Control:** Configure authentication (AD integration), URL filtering, and blacklists.
6. **Configure AWS Security Group:** Allow inbound traffic from specific CIDR blocks only.

4 DNS Server

4.1 AWS Configuration

Instance Type: t3.small

Security Group Ports: 53 (TCP/UDP), 3389 (RDP)

4.2 Implementation Steps

1. Launch Windows Server Instance in a private subnet.
2. **Install DNS Server Role:**

```
Install-WindowsFeature -Name DNS -IncludeManagementTools
```

3. **Configure DNS Zones:**

```
# Create Primary Zone
Add-DnsServerPrimaryZone -Name "yourdomain.local" -ReplicationScope "Forest" -
    PassThru

# Create Reverse Lookup Zone
Add-DnsServerPrimaryZone -NetworkID "10.0.0.0/16" -ReplicationScope "Forest"
```

4. **Configure Forwarders:**

```
# Use AWS DNS or external DNS
Add-DnsServerForwarder -IPAddress "8.8.8.8", "8.8.4.4"
```

5. Integrate with AWS Route 53 Resolver endpoints if needed.

5 DHCP Server

Note: AWS VPC provides DHCP by default; a custom DHCP server is optional.

5.1 AWS Configuration

Instance Type: t3.small

5.2 Implementation Steps

1. **Install DHCP Server Role:**

```
Install-WindowsFeature -Name DHCP -IncludeManagementTools
Add-DhcpServerInDC -DnsName "dhcp.yourdomain.local"
```

2. **Configure DHCP Scope:**

```
Add-DhcpServerv4Scope -Name "Internal Network" -StartRange 10.0.1.100 -EndRange
    10.0.1.200 -SubnetMask 255.255.255.0

Set-DhcpServerv4OptionValue -ScopeId 10.0.1.0 -Router 10.0.1.1
Set-DhcpServerv4OptionValue -ScopeId 10.0.1.0 -DnsServer 10.0.1.10
```

3. **Configure Reservations:**

```
Add-DhcpServerv4Reservation -ScopeId 10.0.1.0 -IPAddress 10.0.1.50 -ClientId "
    00-11-22-33-44-55" -Description "Print Server"
```

4. **Authorize DHCP Server:**

```
Add-DhcpServerInDC -DnsName "dhcp.yourdomain.local" -IPAddress 10.0.1.10
```

6 VPN Server

6.1 AWS Configuration

Instance Type: t3.small to t3.medium

Security Group Ports: 1723 (PPTP), 1701 (L2TP), 500/4500 (IPSec), 443 (SSTP)

Elastic IP: Required

6.2 Implementation Steps

1. Launch Windows Server Instance with Elastic IP.
2. **Install Remote Access Role:**

```
Install-WindowsFeature -Name RemoteAccess -IncludeManagementTools
Install-WindowsFeature -Name DirectAccess-VPN -IncludeManagementTools
Install-WindowsFeature -Name Routing -IncludeManagementTools
```

3. **Configure VPN Server:**

```
Install-RemoteAccess -VpnType Vpn
```

4. Enable SSTP, L2TP/IPSec, or IKEv2 and configure authentication.
5. **Set Up IP Address Assignment:**

```
Set-VpnServerConfiguration -TunnelType SSTP -PassThru
```

6. Configure Routing (NAT and tables).

7 Terminal Server (Thin Clients)

7.1 AWS Configuration

Instance Type: t3.xlarge or larger

Security Group Ports: 3389 (RDP), 3391 (RD Gateway)

7.2 Implementation Steps

1. **Install RDS Roles:**

```
Install-WindowsFeature -Name RDS-RD-Server -IncludeManagementTools
Install-WindowsFeature -Name RDS-Connection-Broker -IncludeManagementTools
Install-WindowsFeature -Name RDS-Web-Access -IncludeManagementTools
Install-WindowsFeature -Name RDS-Gateway -IncludeManagementTools
Install-WindowsFeature -Name RDS-Licensing -IncludeManagementTools
```

2. Configure RDS Deployment via Server Manager.

3. **Configure Session Collections:**

```
New-RDSessionCollection -CollectionName "Production" -SessionHost "rdsh01.yourdomain
    .local" -ConnectionBroker "rdcb.yourdomain.local"
```

4. **Set Up RemoteApp:**

```
New-RDRemoteApp -CollectionName "Production" -DisplayName "Microsoft Word" -FilePath
     "C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE"
```

8 Web Server

8.1 AWS Configuration

Instance Type: t3.medium

Ports: 80, 443, 3389

8.2 Implementation Steps

1. **Install IIS Role:**

```
Install-WindowsFeature -Name Web-Server -IncludeManagementTools
Install-WindowsFeature -Name Web-Asp-Net45, Web-Net-Ext45
Install-WindowsFeature -Name Web-Mgmt-Console
```

2. **Configure IIS:**

```
# Create new website
New-Website -Name "MyWebsite" -Port 80 -PhysicalPath "C:\inetpub\MyWebsite" -
    ApplicationPool "DefaultAppPool"

# Create application pool
New-WebAppPool -Name "MyAppPool"
Set-ItemProperty IIS:\AppPools\MyAppPool -name "managedRuntimeVersion" -value "v4.0"
```

3. **Install SSL Certificate:**

```
New-WebBinding -Name "MyWebsite" -Protocol "https" -Port 443 -SslFlags 0
```

9 Mail Server

9.1 AWS Configuration

Instance Type: t3.medium

Ports: 25 (SMTP), 110, 143, 587, 993, 995

Elastic IP: Required

- **Crucial:** AWS blocks port 25 by default. You must request removal via AWS Support.
- **Software Options:** hMailServer (Free) or Microsoft Exchange Server.
- **Alternative:** Use Amazon SES for better deliverability.

10 Database Server

10.1 AWS Configuration

Instance Type: r5.large or larger (Memory Optimized)

Storage: Provisioned IOPS or io2

10.2 Implementation Steps

SQL Server

```
# Silent installation example
Setup.exe /Q /ACTION=Install /FEATURES=SQLEngine /INSTANCENAME=MSSQLSERVER /
    SQLSYSADMINACCOUNTS="DOMAIN\SQLAdmins" /AGTSVCACCOUNT="NT AUTHORITY\SYSTEM" /
    SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"
```

```
-- Enable remote connections
EXEC sys.sp_configure 'remote access', 1;
RECONFIGURE;

-- Configure max memory
EXEC sys.sp_configure 'max server memory (MB)', 8192;
RECONFIGURE;

-- Backup to S3
BACKUP DATABASE [MyDB] TO URL = 's3://my-bucket/backups/MyDB.bak'
```

PostgreSQL

Edit `postgresql.conf` and `pg_hba.conf`:

```
# postgresql.conf
listen_addresses = '*'
max_connections = 100
shared_buffers = 2GB

# pg_hba.conf
host all all 0.0.0.0/0 md5
```

MongoDB

```
# mongod.cfg
net:
  port: 27017
  bindIp: 0.0.0.0
security:
  authorization: enabled
storage:
  dbPath: D:\MongoDB\data
```

11 Backup Server

11.1 AWS Configuration

- **Instance:** t3.medium
- **Role:** IAM permissions for S3 and EBS snapshots.

11.2 Implementation Steps

1. **Install Windows Server Backup:**

```
Install-WindowsFeature -Name Windows-Server-Backup -IncludeManagementTools
```

2. **Configure Backup to S3 (Example):**

```
$Policy = New-WBPolicy
$Target = New-WBBackupTarget -VolumePath "D:"
Add-WBBackupTarget -Policy $Policy -Target $Target
Add-WBVolume -Policy $Policy -Volume (Get-WBVolume -VolumePath "C:")
Set-WBSchedule -Policy $Policy -Schedule 02:00
Set-WBPolicy -Policy $Policy
```

3. Use AWS Backup for centralized management and S3 Glacier for archiving.

12 Load Balancing

12.1 Implementation Steps

1. Launch multiple identical servers in different availability zones.
2. **Create Target Group:** Protocol HTTP/HTTPS, Health Check Path `/health`.
3. **Create Application Load Balancer (ALB):** Add listener rules and register target group.
4. **Session Persistence:** Enable sticky sessions if required.

13 Failover Cluster

13.1 AWS Configuration

Instance Type: r5.xlarge or larger

Storage: Shared storage via FSx for Windows or EBS Multi-Attach (io2).

13.2 Implementation Steps

1. **Install Failover Clustering:**

```
Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools
```

2. **Create Failover Cluster:**

```
# Validate
Test-Cluster -Node "Node1", "Node2"

# Create
New-Cluster -Name "MyCluster" -Node "Node1", "Node2" -StaticAddress "10.0.1.100" -
    NoStorage
```

3. **Configure Quorum:**

```
Set-ClusterQuorum -NodeAndFileShareMajority "\\FSx\Witness"
```

4. **Add Clustered Role (e.g., SQL):**

```
Add-ClusterServerRole -Name "SQL-Cluster" -Storage "Cluster Disk 1"
```

14 FTP Server

14.1 AWS Configuration

Ports: 21, 20, 990, Passive Range (50000-50100)

Elastic IP: Required

14.2 Implementation Steps

1. **Install Role:**

```
Install-WindowsFeature -Name Web-Ftp-Server -IncludeManagementTools
Install-WindowsFeature -Name Web-Ftp-Service
```

2. **Configure Site & Passive Mode:**

```
New-WebFtpSite -Name "FTP Site" -Port 21 -PhysicalPath "D:\FTP"

# Passive Ports
Set-WebConfigurationProperty -Filter /system.ftpServer/firewallSupport -PSPath IIS:\
     -Name lowDataChannelPort -Value 50000
Set-WebConfigurationProperty -Filter /system.ftpServer/firewallSupport -PSPath IIS:\
     -Name highDataChannelPort -Value 50100
```

3. **Enable FTPS (SSL):**

```
$cert = New-SelfSignedCertificate -DnsName "ftp.yourdomain.com" -CertStoreLocation
    cert:\LocalMachine\My
Set-WebConfigurationProperty -Filter /system.ftpServer/security/ssl -PSPath IIS:\ -
    Location "FTP Site" -Name serverCertHash -Value $cert.Thumbprint
Set-WebConfigurationProperty -Filter /system.ftpServer/security/ssl -PSPath IIS:\ -
    Location "FTP Site" -Name ssl128 -Value $true
```

15 Container (Docker)

15.1 AWS Configuration

OS: Windows Server 2019/2022 with Containers

15.2 Implementation Steps

1. **Install Docker:**

```
Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
Install-Package -Name docker -ProviderName DockerMsftProvider -Force
Restart-Computer -Force
```

2. **Create Dockerfile:**

```
FROM mcr.microsoft.com/dotnet/framework/aspnet:4.8
WORKDIR /inetpub/wwwroot
COPY ./app .
EXPOSE 80
```

3. **Build and Run:**

```
docker build -t mywebapp:v1 .
docker run -d -p 80:80 --name webapp mywebapp:v1
```

4. **Push to ECR:**

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --
    password-stdin ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com
docker push ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/mywebapp:v1
```

16 Domain Controller

16.1 AWS Configuration

Instance Type: t3.medium or larger

Ports: 53, 88, 135, 139, 445, 389, 636, 3268, 3269, 49152-65535

Storage: Minimum 50GB SSD

16.2 Implementation Steps

1. Initial Configuration

Set static IP and rename the computer.

```
New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 10.0.1.10 -PrefixLength 24 -
    DefaultGateway 10.0.1.1
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 127.0.0.1,8.8.8.8
Rename-Computer -NewName "DC01" -Restart
```

2. Install AD DS and Promote to DC

```
Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools
Import-Module ADDSDeployment

Install-ADDSForest `
    -DomainName "company.local" `
    -DomainNetbiosName "COMPANY" `
    -ForestMode "WinThreshold" `
    -DomainMode "WinThreshold" `
    -InstallDns:$true `
    -Force:$true
```

3. Post-Installation Config

```
# Check status
Get-Service ADWS
Get-ADDomainController

# Create OUs
New-ADOrganizationalUnit -Name "Users" -Path "DC=company,DC=local"
New-ADOrganizationalUnit -Name "Computers" -Path "DC=company,DC=local"

# Enable Recycle Bin
Enable-ADOptionalFeature -Identity 'Recycle Bin Feature' -Scope ForestOrConfigurationSet
    -Target 'company.local' -Confirm:$false
```

4. Time Sync (PDC Emulator)

```
w32tm /config /manualpeerlist:"time.windows.com,0x8" /syncfromflags:manual /reliable:yes
    /update
Restart-Service W32Time
```

16.3 Security Best Practices

- Implement least privilege access.
- Use separate administrative accounts.
- Enable and monitor security logs.
- Regularly patch and update.
- Use strong password policies.