# 🪟 CYPHER Deployment from Windows Server

## 📋 **Prerequisites**

### **Required Software on Windows Server:**

* ✅ **PowerShell 5.1+** (built-in on Windows Server 2016+)
* ✅ **AWS CLI v2** - [Download here](https://awscli.amazonaws.com/AWSCLIV2.msi)
* ✅ **Git for Windows** - [Download here](https://git-scm.com/download/win)
* ✅ **OpenSSH Client** (Windows 10/Server 2019+ built-in)

### **Optional but Recommended:**

* ✅ **Windows Terminal** - Better terminal experience
* ✅ **VS Code** - For file editing and remote SSH
* ✅ **PuTTY** - Alternative SSH client

## 🚀 **Method 1: PowerShell with AWS Systems Manager (Recommended)**

### **Step 1: Configure AWS CLI**

# Open PowerShell as Administrator  
aws configure  
  
# Enter your credentials:  
# AWS Access Key ID: [Your Access Key]  
# AWS Secret Access Key: [Your Secret Key]  
# Default region name: us-east-1  
# Default output format: json

### **Step 2: Deploy Setup Script via SSM**

# Send the setup script to your EC2 instance  
$instanceId = "i-04a41343a3f51559a"  
$commandId = aws ssm send-command `  
 --instance-ids $instanceId `  
 --document-name "AWS-RunShellScript" `  
 --parameters 'commands=["curl -s https://cypher-deployment.s3.amazonaws.com/complete-cypher-setup.sh | bash"]' `  
 --query 'Command.CommandId' `  
 --output text  
  
Write-Host "Command ID: $commandId"

### **Step 3: Monitor Execution**

# Check command status  
aws ssm get-command-invocation --command-id $commandId --instance-id $instanceId  
  
# Get detailed output  
aws ssm get-command-invocation --command-id $commandId --instance-id $instanceId --query 'StandardOutputContent' --output text

## 🔧 **Method 2: PowerShell with SSH**

### **Step 1: Set Up SSH Connection**

# Create SSH directory if it doesn't exist  
New-Item -ItemType Directory -Force -Path "$env:USERPROFILE\.ssh"  
  
# Copy your EC2 key pair to the SSH directory  
# Place your .pem file in: C:\Users\YourUsername\.ssh\your-key.pem  
  
# Set proper permissions on the key file  
icacls "$env:USERPROFILE\.ssh\your-key.pem" /inheritance:r  
icacls "$env:USERPROFILE\.ssh\your-key.pem" /grant:r "$env:USERNAME:R"

### **Step 2: Connect and Deploy**

# Get your EC2 public IP  
$publicIp = aws ec2 describe-instances --instance-ids i-04a41343a3f51559a --query 'Reservations[0].Instances[0].PublicIpAddress' --output text  
  
# Connect via SSH and run setup  
ssh -i "$env:USERPROFILE\.ssh\your-key.pem" ec2-user@$publicIp "curl -s https://cypher-deployment.s3.amazonaws.com/complete-cypher-setup.sh | bash"

## 🌐 **Method 3: PowerShell Web-Based Deployment**

### **Step 1: Create Deployment Script**

# Create a PowerShell deployment script  
@"  
# CYPHER Windows Server Deployment Script  
Write-Host "🚀 Starting CYPHER Deployment from Windows Server..." -ForegroundColor Green  
  
# Configuration  
`$instanceId = "i-04a41343a3f51559a"  
`$region = "us-east-1"  
  
# Function to check AWS CLI  
function Test-AWSCli {  
 try {  
 aws --version | Out-Null  
 return `$true  
 } catch {  
 Write-Host "❌ AWS CLI not found. Please install AWS CLI v2" -ForegroundColor Red  
 return `$false  
 }  
}  
  
# Function to deploy via SSM  
function Deploy-ViaSSM {  
 Write-Host "📡 Deploying via AWS Systems Manager..." -ForegroundColor Yellow  
   
 `$command = 'curl -s https://cypher-deployment.s3.amazonaws.com/complete-cypher-setup.sh | bash'  
   
 `$commandId = aws ssm send-command ``  
 --instance-ids `$instanceId ``  
 --document-name "AWS-RunShellScript" ``  
 --parameters "commands=[`"`$command`"]" ``  
 --query 'Command.CommandId' ``  
 --output text  
   
 if (`$commandId) {  
 Write-Host "✅ Command sent successfully. Command ID: `$commandId" -ForegroundColor Green  
   
 # Monitor execution  
 do {  
 Start-Sleep -Seconds 10  
 `$status = aws ssm get-command-invocation --command-id `$commandId --instance-id `$instanceId --query 'Status' --output text  
 Write-Host "📊 Status: `$status" -ForegroundColor Cyan  
 } while (`$status -eq "InProgress")  
   
 # Get output  
 `$output = aws ssm get-command-invocation --command-id `$commandId --instance-id `$instanceId --query 'StandardOutputContent' --output text  
 Write-Host "📄 Output:" -ForegroundColor Yellow  
 Write-Host `$output  
   
 if (`$status -eq "Success") {  
 Write-Host "🎉 Deployment completed successfully!" -ForegroundColor Green  
 } else {  
 Write-Host "❌ Deployment failed. Check the output above." -ForegroundColor Red  
 }  
 } else {  
 Write-Host "❌ Failed to send command" -ForegroundColor Red  
 }  
}  
  
# Main execution  
if (Test-AWSCli) {  
 Deploy-ViaSSM  
} else {  
 Write-Host "Please install AWS CLI and run 'aws configure' first." -ForegroundColor Red  
}  
"@ | Out-File -FilePath "Deploy-CYPHER.ps1" -Encoding UTF8  
  
Write-Host "✅ Created Deploy-CYPHER.ps1 script" -ForegroundColor Green  
Write-Host "Run: .\Deploy-CYPHER.ps1" -ForegroundColor Yellow

### **Step 2: Execute Deployment**

# Run the deployment script  
.\Deploy-CYPHER.ps1

## 🔐 **Method 4: Windows Server with PuTTY**

### **Step 1: Install PuTTY**

# Download and install PuTTY  
Invoke-WebRequest -Uri "https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.78-installer.msi" -OutFile "putty-installer.msi"  
Start-Process -FilePath "putty-installer.msi" -Wait

### **Step 2: Convert PEM to PPK**

# Use PuTTYgen to convert .pem to .ppk format  
# 1. Open PuTTYgen  
# 2. Load your .pem file  
# 3. Save private key as .ppk file

### **Step 3: Connect and Deploy**

1. **Open PuTTY**
2. **Host Name**: ec2-user@your-ec2-public-ip
3. **Port**: 22
4. **Connection → SSH → Auth**: Browse to your .ppk file
5. **Open connection**
6. **Run**: curl -s https://cypher-deployment.s3.amazonaws.com/complete-cypher-setup.sh | bash

## 📊 **Monitoring and Management from Windows**

### **PowerShell Monitoring Script**

# Create monitoring script  
@"  
# CYPHER Monitoring Script for Windows Server  
param(  
 [string]`$InstanceId = "i-04a41343a3f51559a"  
)  
  
function Get-CypherStatus {  
 Write-Host "🔍 Checking CYPHER Application Status..." -ForegroundColor Cyan  
   
 # Get instance status  
 `$instanceStatus = aws ec2 describe-instances --instance-ids `$InstanceId --query 'Reservations[0].Instances[0].State.Name' --output text  
 Write-Host "EC2 Instance Status: `$instanceStatus" -ForegroundColor Yellow  
   
 # Get public IP  
 `$publicIp = aws ec2 describe-instances --instance-ids `$InstanceId --query 'Reservations[0].Instances[0].PublicIpAddress' --output text  
 Write-Host "Public IP: `$publicIp" -ForegroundColor Yellow  
   
 # Test application health  
 try {  
 `$response = Invoke-WebRequest -Uri "http://`$publicIp/health" -TimeoutSec 10  
 if (`$response.StatusCode -eq 200) {  
 Write-Host "✅ Application is healthy!" -ForegroundColor Green  
 }  
 } catch {  
 Write-Host "❌ Application health check failed" -ForegroundColor Red  
 }  
   
 # Test main application  
 try {  
 `$response = Invoke-WebRequest -Uri "http://`$publicIp" -TimeoutSec 10  
 if (`$response.StatusCode -eq 200) {  
 Write-Host "✅ Main application is accessible!" -ForegroundColor Green  
 Write-Host "🌐 Application URL: http://`$publicIp" -ForegroundColor Cyan  
 }  
 } catch {  
 Write-Host "❌ Main application not accessible" -ForegroundColor Red  
 }  
}  
  
Get-CypherStatus  
"@ | Out-File -FilePath "Monitor-CYPHER.ps1" -Encoding UTF8  
  
Write-Host "✅ Created Monitor-CYPHER.ps1 script" -ForegroundColor Green

### **Automated Deployment Check**

# Check if deployment completed successfully  
function Test-DeploymentComplete {  
 $instanceId = "i-04a41343a3f51559a"  
   
 # Check if PM2 is running  
 $command = "pm2 status"  
 $commandId = aws ssm send-command --instance-ids $instanceId --document-name "AWS-RunShellScript" --parameters "commands=[`"$command`"]" --query 'Command.CommandId' --output text  
   
 Start-Sleep -Seconds 5  
 $output = aws ssm get-command-invocation --command-id $commandId --instance-id $instanceId --query 'StandardOutputContent' --output text  
   
 if ($output -match "cypher-api") {  
 Write-Host "✅ CYPHER application is running!" -ForegroundColor Green  
 return $true  
 } else {  
 Write-Host "❌ CYPHER application not detected" -ForegroundColor Red  
 return $false  
 }  
}

## 🎯 **Post-Deployment Steps**

### **Step 1: Configure Application**

# Send AWS credentials configuration to EC2  
$instanceId = "i-04a41343a3f51559a"  
$configCommand = @"  
aws configure set aws\_access\_key\_id YOUR\_ACCESS\_KEY  
aws configure set aws\_secret\_access\_key YOUR\_SECRET\_KEY  
aws configure set default.region us-east-1  
aws configure set default.output json  
"@  
  
aws ssm send-command --instance-ids $instanceId --document-name "AWS-RunShellScript" --parameters "commands=[`"$configCommand`"]"

### **Step 2: Deploy Application**

# Deploy the CYPHER application  
$deployCommand = "./deploy-cypher.sh"  
aws ssm send-command --instance-ids $instanceId --document-name "AWS-RunShellScript" --parameters "commands=[`"$deployCommand`"]"

### **Step 3: Verify Deployment**

# Run monitoring script  
.\Monitor-CYPHER.ps1

## 🚨 **Troubleshooting from Windows**

### **Common Issues and Solutions**

#### **AWS CLI Not Found**

# Install AWS CLI v2  
Invoke-WebRequest -Uri "https://awscli.amazonaws.com/AWSCLIV2.msi" -OutFile "AWSCLIV2.msi"  
Start-Process -FilePath "AWSCLIV2.msi" -Wait

#### **SSH Connection Issues**

# Test SSH connectivity  
Test-NetConnection -ComputerName $publicIp -Port 22

#### **SSM Not Working**

# Check if instance is SSM-managed  
aws ssm describe-instance-information --filters "Key=InstanceIds,Values=i-04a41343a3f51559a"

## 🎉 **Quick Start Summary**

**For immediate deployment from Windows Server:**

1. **Install AWS CLI**: Download and install AWS CLI v2
2. **Configure credentials**: Run aws configure
3. **Deploy**: Run the PowerShell deployment script
4. **Monitor**: Use the monitoring script to check status
5. **Access**: Open browser to http://your-ec2-ip

**Total deployment time: ~15-20 minutes** 🚀