# CYPHER Dashboard Update Guide for Windows Server 2019

## Overview

This guide provides comprehensive instructions for updating your CYPHER Dashboard application after the initial deployment on Windows Server 2019.

## Update Methods Overview

| Method | Use Case | Time Required | Risk Level | Backup Created |
| --- | --- | --- | --- | --- |
| **Quick Update** | Small changes, bug fixes | 2-5 minutes | Low | Manual |
| **Full Script Re-run** | Major updates, new features | 10-15 minutes | Very Low | Automatic |
| **Manual Process** | Custom updates, troubleshooting | 5-10 minutes | Medium | Manual |
| **Git-based** | Continuous deployment | 3-7 minutes | Low | Manual |

## Method 1: Quick Code Updates (Recommended for Small Changes)

### When to Use:

* ✅ Small bug fixes
* ✅ Minor feature changes
* ✅ Single file updates
* ✅ UI/styling changes
* ✅ Configuration tweaks

### Steps:

#### Step 1: Stop Services

# Connect via RDP to your Windows Server  
# Open PowerShell as Administrator  
  
# Stop all CYPHER services  
pm2 stop all  
  
# Verify services are stopped  
pm2 status

#### Step 2: Update Files

# Copy your updated files to the server using one of these methods:  
# - WinSCP file transfer  
# - Remote Desktop copy/paste  
# - Network drive copy  
  
# File locations:  
# API updates: C:\CYPHER-Dashboard\api\src\  
# Client updates: C:\CYPHER-Dashboard\client\src\

#### Step 3: Restart Services

# For API-only updates:  
pm2 restart cypher-api  
  
# For Client-only updates (requires rebuild):  
cd C:\CYPHER-Dashboard\client  
npm run build  
pm2 restart cypher-client  
  
# For both API and Client updates:  
cd C:\CYPHER-Dashboard\client  
npm run build  
pm2 restart all  
  
# Verify services are running  
pm2 status

#### Step 4: Test Application

# Test API health  
Invoke-WebRequest -Uri "http://localhost:3001/health"  
  
# Test Client application  
Start-Process "http://localhost:3000"

## Method 2: Full Script Re-run (Recommended for Major Updates)

### When to Use:

* ✅ Major version updates
* ✅ New npm dependencies
* ✅ Database schema changes
* ✅ Configuration file changes
* ✅ New environment variables
* ✅ When you want automatic backup

### Steps:

#### Step 1: Copy Updated Files

# Copy your updated api and client folders to the server  
# Overwrite existing files in:  
# C:\CYPHER-Dashboard\api\  
# C:\CYPHER-Dashboard\client\

#### Step 2: Run Full Deployment Script

# Navigate to script location  
cd C:\  
  
# Run the deployment script  
.\Deploy-CYPHER-WindowsServer.ps1

### What the Script Does:

1. **Creates automatic backup** of existing installation
2. **Verifies updated files** are present
3. **Installs new dependencies** (if any)
4. **Rebuilds client application**
5. **Updates configuration files**
6. **Restarts all services**
7. **Performs health checks**

## Method 3: Manual Update Process

### When to Use:

* ✅ Custom deployment requirements
* ✅ Troubleshooting issues
* ✅ Selective component updates
* ✅ When you need full control

### Steps:

#### Step 1: Create Backup (Optional but Recommended)

# Create backup of current installation  
$backupName = "CYPHER-Dashboard-Backup-$(Get-Date -Format 'yyyyMMdd-HHmmss')"  
Copy-Item -Path "C:\CYPHER-Dashboard" -Destination "C:\$backupName" -Recurse  
Write-Host "Backup created at C:\$backupName"

#### Step 2: Stop Services

pm2 stop all  
pm2 status

#### Step 3: Update Files

# Copy your updated files to appropriate locations  
# API files: C:\CYPHER-Dashboard\api\src\  
# Client files: C:\CYPHER-Dashboard\client\src\

#### Step 4: Install Dependencies

# Update API dependencies  
cd C:\CYPHER-Dashboard\api  
npm install --production  
  
# Update Client dependencies  
cd C:\CYPHER-Dashboard\client  
npm install

#### Step 5: Rebuild Client

# Build client for production  
cd C:\CYPHER-Dashboard\client  
npm run build

#### Step 6: Update Configuration (if needed)

# Update environment files if needed  
# C:\CYPHER-Dashboard\api\.env  
# C:\CYPHER-Dashboard\client\.env

#### Step 7: Restart Services

# Start services using PM2 configuration  
cd C:\CYPHER-Dashboard  
pm2 start ecosystem.config.js  
  
# Or restart existing processes  
pm2 restart all  
  
# Check status  
pm2 status

## Method 4: Git-based Updates (Advanced)

### Prerequisites:

* Git installed on Windows Server
* Repository configured in C:-Dashboard

### Steps:

# Navigate to application directory  
cd C:\CYPHER-Dashboard  
  
# Stop services  
pm2 stop all  
  
# Pull latest changes  
git pull origin main  
  
# Install any new dependencies  
cd api  
npm install --production  
  
cd ..\client  
npm install  
  
# Rebuild client  
npm run build  
  
# Restart services  
cd ..  
pm2 restart all  
  
# Check status  
pm2 status

## File Locations Reference

### Application Structure:

C:\CYPHER-Dashboard\  
├── api\  
│ ├── src\ # API source code (update here)  
│ ├── package.json # API dependencies  
│ └── .env # API configuration  
├── client\  
│ ├── src\ # Client source code (update here)  
│ ├── dist\ # Built client files (auto-generated)  
│ ├── package.json # Client dependencies  
│ └── .env # Client configuration  
└── ecosystem.config.js # PM2 configuration

### Log Locations:

C:\CYPHER-logs\  
├── cypher-api.log # API application logs  
├── cypher-api-error.log # API error logs  
├── cypher-client.log # Client application logs  
└── cypher-client-error.log # Client error logs

### Backup Locations:

C:\CYPHER-Dashboard-Backup-YYYYMMDD-HHMMSS\ # Automatic backups

## Service Management Commands

### PM2 Commands:

# Check service status  
pm2 status  
  
# View logs (all services)  
pm2 logs  
  
# View logs (specific service)  
pm2 logs cypher-api  
pm2 logs cypher-client  
  
# Restart specific service  
pm2 restart cypher-api  
pm2 restart cypher-client  
  
# Restart all services  
pm2 restart all  
  
# Stop all services  
pm2 stop all  
  
# Start services from config  
pm2 start ecosystem.config.js  
  
# Delete all processes (careful!)  
pm2 delete all  
  
# Monitor services in real-time  
pm2 monit  
  
# Save PM2 configuration  
pm2 save

## Troubleshooting Updates

### Common Issues:

#### 1. Services Won’t Start After Update

# Check PM2 logs for errors  
pm2 logs  
  
# Check if ports are in use  
netstat -ano | findstr :3001  
netstat -ano | findstr :3000  
  
# Restart PM2 daemon  
pm2 kill  
pm2 start ecosystem.config.js

#### 2. Client Build Fails

# Clear npm cache  
cd C:\CYPHER-Dashboard\client  
npm cache clean --force  
  
# Delete node\_modules and reinstall  
Remove-Item -Path "node\_modules" -Recurse -Force  
npm install  
  
# Try build again  
npm run build

#### 3. Database Connection Issues

# Check environment variables  
Get-Content C:\CYPHER-Dashboard\api\.env  
  
# Test database connectivity  
# Verify database credentials and network access

#### 4. Permission Issues

# Run PowerShell as Administrator  
# Check file permissions  
icacls C:\CYPHER-Dashboard /T  
  
# Fix permissions if needed  
icacls C:\CYPHER-Dashboard /grant Everyone:F /T

## Best Practices

### Before Updates:

* ✅ **Test changes** on development environment first
* ✅ **Create backup** of current working version
* ✅ **Document changes** being made
* ✅ **Plan rollback strategy**

### During Updates:

* ✅ **Use maintenance window** for major updates
* ✅ **Monitor logs** during restart
* ✅ **Test functionality** after update
* ✅ **Verify database connectivity**

### After Updates:

* ✅ **Perform health checks**
* ✅ **Monitor application performance**
* ✅ **Check error logs** for issues
* ✅ **Document successful update**

## Rollback Procedure

### If Update Fails:

# Stop current services  
pm2 stop all  
  
# Restore from backup  
$latestBackup = Get-ChildItem C:\ -Name "CYPHER-Dashboard-Backup-\*" | Sort-Object -Descending | Select-Object -First 1  
Remove-Item -Path "C:\CYPHER-Dashboard" -Recurse -Force  
Copy-Item -Path "C:\$latestBackup" -Destination "C:\CYPHER-Dashboard" -Recurse  
  
# Restart services  
cd C:\CYPHER-Dashboard  
pm2 start ecosystem.config.js  
  
# Verify rollback successful  
pm2 status

## Update Checklist

### Pre-Update:

* Backup current installation
* Test changes in development
* Plan maintenance window
* Notify users (if applicable)

### During Update:

* Stop services
* Copy updated files
* Install/update dependencies
* Rebuild client (if needed)
* Update configuration (if needed)
* Restart services

### Post-Update:

* Verify services are running
* Test API endpoints
* Test client application
* Check logs for errors
* Monitor performance
* Document update completion

**Remember**: When in doubt, use **Method 2 (Full Script Re-run)** - it’s the safest option with automatic backups! 🚀