# CYPHER Dashboard Manual Deployment Guide for Windows Server 2019

## Overview

This guide provides step-by-step manual instructions for deploying the CYPHER Dashboard application on Windows Server 2019 without using PowerShell scripts. This approach gives you complete control over each step of the deployment process.

## Prerequisites

* Windows Server 2019 EC2 instance
* Administrator access to the server
* Internet connectivity
* Your CYPHER application source code (api and client folders)
* Remote Desktop (RDP) access to the server

## Deployment Architecture

* **API Server**: Node.js/Express running on port 3001
* **Client**: React/Vite application running on port 3000
* **Database**: PostgreSQL (existing RAS DASH database)
* **Process Manager**: PM2 for service management
* **Static File Server**: serve package for client files

## Step 1: Install Prerequisites

### Install Node.js

1. **Connect to your Windows Server** via RDP
2. **Open a web browser** (Internet Explorer/Edge)
3. **Navigate to** https://nodejs.org
4. **Download** the LTS version (recommended)
5. **Run the installer** and follow the setup wizard
6. **Accept all default settings**
7. **Verify installation** by opening Command Prompt and running:

* node --version  
  npm --version
* You should see version numbers for both commands.

### Install PM2 Process Manager

1. **Open Command Prompt as Administrator**
2. **Install PM2 globally:**

* npm install -g pm2  
  npm install -g pm2-windows-startup

1. **Verify PM2 installation:**

* pm2 --version

### Install serve Package (for serving client files)

npm install -g serve

## Step 2: Copy Application Files

### Create Directory Structure

1. **Create the main application directory:**

* mkdir C:\CYPHER-Dashboard

### Copy Your Source Code

Using your preferred method (WinSCP, RDP copy/paste, network share):

1. **Copy your api folder** to C:\CYPHER-Dashboard\api\
2. **Copy your client folder** to C:\CYPHER-Dashboard\client\

### Verify File Structure

Your directory should look like this:

C:\CYPHER-Dashboard\  
├── api\  
│ ├── src\  
│ ├── package.json  
│ └── [other API files]  
└── client\  
 ├── src\  
 ├── package.json  
 └── [other client files]

## Step 3: Install Dependencies

### Install API Dependencies

1. **Open Command Prompt**
2. **Navigate to API directory:**

* cd C:\CYPHER-Dashboard\api

1. **Install dependencies:**

* npm install --production

1. **Wait for installation to complete** (may take several minutes)

### Install Client Dependencies

1. **Navigate to Client directory:**

* cd C:\CYPHER-Dashboard\client

1. **Install dependencies:**

* npm install

1. **Wait for installation to complete** (may take several minutes)

## Step 4: Create Environment Configuration Files

### Create API Environment File

1. **Navigate to** C:\CYPHER-Dashboard\api\
2. **Create a new file** named .env
3. **Add the following content:**

* NODE\_ENV=production  
  PORT=3001  
  DB\_HOST=rasdash-dev-public.cexgrlslydeh.us-east-1.rds.amazonaws.com  
  DB\_PORT=5432  
  DB\_NAME=rasdashdevo1  
  DB\_USER=rasdashadmin  
  DB\_PASSWORD=RasDash2025$  
  JWT\_SECRET=your-random-jwt-secret-here-replace-with-actual-secret  
  CORS\_ORIGIN=http://localhost:3000  
  LOG\_LEVEL=info

**Important**: Replace your-random-jwt-secret-here-replace-with-actual-secret with a secure random string.

### Create Client Environment File

1. **Navigate to** C:\CYPHER-Dashboard\client\
2. **Create a new file** named .env
3. **Add the following content:**

* VITE\_API\_URL=http://localhost:3001  
  VITE\_APP\_NAME=CYPHER Dashboard  
  NODE\_ENV=production

## Step 5: Build Client Application

1. **Open Command Prompt**
2. **Navigate to client directory:**

* cd C:\CYPHER-Dashboard\client

1. **Build the application:**

* npm run build

1. **Wait for build to complete**
2. **Verify** that a dist folder was created in the client directory

## Step 6: Create PM2 Configuration

### Create PM2 Ecosystem File

1. **Navigate to** C:\CYPHER-Dashboard\
2. **Create a new file** named ecosystem.config.js
3. **Add the following content:**

* module.exports = {  
   apps: [  
   {  
   name: 'cypher-api',  
   script: './api/src/app.js',  
   cwd: 'C:/CYPHER-Dashboard',  
   instances: 1,  
   exec\_mode: 'fork',  
   env: {  
   NODE\_ENV: 'production',  
   PORT: 3001  
   },  
   error\_file: 'C:/CYPHER-logs/cypher-api-error.log',  
   out\_file: 'C:/CYPHER-logs/cypher-api-out.log',  
   log\_file: 'C:/CYPHER-logs/cypher-api.log',  
   time: true,  
   autorestart: true,  
   max\_restarts: 10,  
   min\_uptime: '10s',  
   max\_memory\_restart: '1G'  
   },  
   {  
   name: 'cypher-client',  
   script: 'npx',  
   args: 'serve -s dist -l 3000',  
   cwd: 'C:/CYPHER-Dashboard/client',  
   instances: 1,  
   exec\_mode: 'fork',  
   env: {  
   NODE\_ENV: 'production'  
   },  
   error\_file: 'C:/CYPHER-logs/cypher-client-error.log',  
   out\_file: 'C:/CYPHER-logs/cypher-client-out.log',  
   log\_file: 'C:/CYPHER-logs/cypher-client.log',  
   time: true,  
   autorestart: true,  
   max\_restarts: 10,  
   min\_uptime: '10s',  
   max\_memory\_restart: '512M'  
   }  
   ]  
  };

### Create Logs Directory

mkdir C:\CYPHER-logs

## Step 7: Configure Windows Firewall

### Method A: Using Windows Firewall GUI

1. **Open** “Windows Defender Firewall with Advanced Security”
2. **Click** “Inbound Rules” in the left panel
3. **Click** “New Rule…” in the right panel
4. **Select** “Port” → Next
5. **Select** “TCP” and “Specific Local Ports”
6. **Enter** 3000,3001 → Next
7. **Select** “Allow the connection” → Next
8. **Check all profiles** (Domain, Private, Public) → Next
9. **Name** the rule “CYPHER Dashboard” → Finish

### Method B: Using Command Line (as Administrator)

netsh advfirewall firewall add rule name="CYPHER API" dir=in action=allow protocol=TCP localport=3001  
netsh advfirewall firewall add rule name="CYPHER Client" dir=in action=allow protocol=TCP localport=3000

## Step 8: Start Services

### Start PM2 Services

1. **Open Command Prompt as Administrator**
2. **Navigate to application directory:**

* cd C:\CYPHER-Dashboard

1. **Start services:**

* pm2 start ecosystem.config.js

1. **Save PM2 configuration:**

* pm2 save

1. **Configure auto-start on Windows boot:**

* pm2-startup install

## Step 9: Verify Deployment

### Check PM2 Status

pm2 status

You should see both cypher-api and cypher-client with status “online”.

### Test API Endpoint

1. **Open web browser**
2. **Navigate to** http://localhost:3001/health
3. **Verify** you get a response (may be JSON or simple text)

### Test Client Application

1. **Open web browser**
2. **Navigate to** http://localhost:3000
3. **Verify** the CYPHER Dashboard loads

### Check Logs

pm2 logs

This will show real-time logs from both services.

## Step 10: Access Your Application

### Local Access (from the server)

* **CYPHER Dashboard**: http://localhost:3000
* **API**: http://localhost:3001

### Remote Access (from other computers)

* **CYPHER Dashboard**: http://your-server-ip:3000
* **API**: http://your-server-ip:3001

**Note**: Replace your-server-ip with your actual Windows Server IP address.

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

## File Locations Reference

### Application Files

C:\CYPHER-Dashboard\  
├── api\  
│ ├── src\ # API source code  
│ ├── node\_modules\ # API dependencies  
│ ├── package.json # API configuration  
│ └── .env # API environment variables  
├── client\  
│ ├── src\ # Client source code  
│ ├── dist\ # Built client files (served to users)  
│ ├── node\_modules\ # Client dependencies  
│ ├── package.json # Client configuration  
│ └── .env # Client environment variables  
└── ecosystem.config.js # PM2 configuration

### Log Files

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

## Troubleshooting

### Common Issues

#### 1. Services Won’t Start

# 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. Cannot Access from Remote Computers

* **Check Windows Firewall** rules are created
* **Verify EC2 Security Group** allows inbound traffic on ports 3000 and 3001
* **Test locally first** to ensure services are running

#### 3. Client Build Fails

# Clear npm cache  
cd C:\CYPHER-Dashboard\client  
npm cache clean --force  
  
# Delete node\_modules and reinstall  
rmdir /s node\_modules  
npm install  
  
# Try build again  
npm run build

#### 4. Database Connection Issues

* **Check environment variables** in C:\CYPHER-Dashboard\api\.env
* **Verify database credentials** and network connectivity
* **Check API logs** for specific error messages

#### 5. Permission Issues

* **Run Command Prompt as Administrator**
* **Check file permissions** on application directories
* **Ensure PM2 has proper permissions**

### Log Analysis

# View recent logs  
pm2 logs --lines 50  
  
# View specific service logs  
pm2 logs cypher-api --lines 20  
  
# Monitor logs in real-time  
pm2 logs --follow

## Updating Your Application

### To Update Code:

1. **Stop services**: pm2 stop all
2. **Copy new files** to replace existing ones
3. **Install new dependencies** (if any): npm install
4. **Rebuild client**: npm run build (in client directory)
5. **Restart services**: pm2 restart all

### To Update Configuration:

1. **Edit .env files** as needed
2. **Restart services**: pm2 restart all

## Security Considerations

### Production Recommendations:

* **Change default JWT secret** to a secure random string
* **Configure HTTPS** with SSL certificates
* **Restrict database access** to specific IP addresses
* **Regular security updates** for Node.js and dependencies
* **Monitor logs** for suspicious activity
* **Backup configuration files** regularly

## Support and Maintenance

### Regular Tasks:

* **Monitor PM2 processes**: pm2 status
* **Check logs regularly**: pm2 logs
* **Monitor disk space** and memory usage
* **Update Node.js and npm** periodically
* **Backup application data** and configuration

### Backup Strategy:

* **Application files**: C:\CYPHER-Dashboard\
* **Configuration files**: .env files and ecosystem.config.js
* **Database**: Regular PostgreSQL backups
* **Logs**: Archive old log files periodically

**Deployment completed successfully!** 🚀

Your CYPHER Dashboard should now be running and accessible. For any issues, refer to the troubleshooting section or check the PM2 logs for detailed error information.