# CYPHER Dashboard Deployment Guide

## Overview

This guide provides step-by-step instructions for deploying the CYPHER Dashboard application on your existing RAS DASH EC2 instance (i-04a41343a3f51559a).

## Prerequisites

* AWS CLI configured with appropriate permissions
* Access to the RAS DASH EC2 instance
* PostgreSQL database connection details
* S3 bucket with deployment files

## 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
* **Reverse Proxy**: Nginx (optional, for production)

## Step 1: Connect to EC2 Instance

# Connect to your EC2 instance  
ssh -i your-key.pem ec2-user@your-ec2-public-ip  
  
# Or if using AWS Systems Manager Session Manager  
aws ssm start-session --target i-04a41343a3f51559a

## Step 2: Prepare the Environment

### Install Node.js (if not already installed)

# Install Node.js 20.x  
curl -fsSL https://rpm.nodesource.com/setup\_20.x | sudo bash -  
sudo yum install -y nodejs  
  
# Verify installation  
node --version  
npm --version

### Install AWS CLI (if not already installed)

# Install AWS CLI v2  
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install  
  
# Configure AWS CLI (use your credentials)  
aws configure

## Step 3: Download and Run Deployment Script

# Download the deployment script from S3  
aws s3 cp s3://cypher-deployment-20250806/deploy-cypher.sh /tmp/deploy-cypher.sh  
  
# Make the script executable  
chmod +x /tmp/deploy-cypher.sh  
  
# Run the deployment script as root  
sudo /tmp/deploy-cypher.sh

## Step 4: Verify Deployment

### Check PM2 Status

# Check if services are running  
sudo pm2 status  
  
# View logs  
sudo pm2 logs  
  
# Monitor in real-time  
sudo pm2 monit

### Test API Endpoints

# Test API health endpoint  
curl http://localhost:3001/health  
  
# Test API with sample request  
curl http://localhost:3001/api/status

### Test Client Application

# Test client application  
curl http://localhost:3000  
  
# Or open in browser  
# http://your-ec2-public-ip:3000

## Step 5: Configure Security Groups (if needed)

Ensure your EC2 security group allows inbound traffic on: - Port 3000 (Client) - Port 3001 (API) - Port 22 (SSH) - Port 80/443 (if using Nginx)

# Example AWS CLI commands to update security group  
aws ec2 authorize-security-group-ingress \  
 --group-id sg-your-security-group-id \  
 --protocol tcp \  
 --port 3000 \  
 --cidr 0.0.0.0/0  
  
aws ec2 authorize-security-group-ingress \  
 --group-id sg-your-security-group-id \  
 --protocol tcp \  
 --port 3001 \  
 --cidr 0.0.0.0/0

## Application Configuration

### Environment Variables

The deployment script automatically creates environment files:

**API Environment (/opt/cypher-dashboard/api/.env)**:

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=<auto-generated>  
CORS\_ORIGIN=http://localhost:3000  
LOG\_LEVEL=info

**Client Environment (/opt/cypher-dashboard/client/.env)**:

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

## Service Management

### PM2 Commands

# Start services  
sudo pm2 start ecosystem.config.js  
  
# Stop services  
sudo pm2 stop all  
  
# Restart services  
sudo pm2 restart all  
  
# Delete services  
sudo pm2 delete all  
  
# View logs  
sudo pm2 logs  
  
# Monitor services  
sudo pm2 monit  
  
# Save PM2 configuration  
sudo pm2 save  
  
# Setup PM2 to start on boot  
sudo pm2 startup

### Manual Service Start (if needed)

# Start API manually  
cd /opt/cypher-dashboard/api  
node src/app.js  
  
# Start Client manually (in another terminal)  
cd /opt/cypher-dashboard/client  
npx serve -s dist -l 3000

## Troubleshooting

### Common Issues

1. **Port Already in Use**

* # Check what's using the port  
  sudo netstat -tlnp | grep :3001  
  sudo netstat -tlnp | grep :3000  
    
  # Kill process if needed  
  sudo kill -9 <PID>

1. **Database Connection Issues**

* # Test database connection  
  psql -h rasdash-dev-public.cexgrlslydeh.us-east-1.rds.amazonaws.com \  
   -p 5432 -U rasdashadmin -d rasdashdevo1

1. **Permission Issues**

* # Fix ownership  
  sudo chown -R ec2-user:ec2-user /opt/cypher-dashboard  
    
  # Fix permissions  
  sudo chmod -R 755 /opt/cypher-dashboard

1. **Node.js/npm Issues**

* # Clear npm cache  
  npm cache clean --force  
    
  # Reinstall dependencies  
  cd /opt/cypher-dashboard/api && npm install  
  cd /opt/cypher-dashboard/client && npm install

### Log Locations

* Deployment log: /var/log/cypher-deployment.log
* API logs: /var/log/cypher-api.log
* Client logs: /var/log/cypher-client.log
* PM2 logs: ~/.pm2/logs/

## Accessing the Application

Once deployed successfully: - **Client Application**: http://your-ec2-public-ip:3000 - **API Endpoints**: http://your-ec2-public-ip:3001 - **Health Check**: http://your-ec2-public-ip:3001/health

## Optional: Setup Nginx Reverse Proxy

For production use, consider setting up Nginx as a reverse proxy:

# Install Nginx  
sudo yum install -y nginx  
  
# Configure Nginx (create /etc/nginx/conf.d/cypher.conf)  
sudo tee /etc/nginx/conf.d/cypher.conf > /dev/null <<EOF  
server {  
 listen 80;  
 server\_name your-domain.com;  
  
 location / {  
 proxy\_pass http://localhost:3000;  
 proxy\_http\_version 1.1;  
 proxy\_set\_header Upgrade \$http\_upgrade;  
 proxy\_set\_header Connection 'upgrade';  
 proxy\_set\_header Host \$host;  
 proxy\_cache\_bypass \$http\_upgrade;  
 }  
  
 location /api {  
 proxy\_pass http://localhost:3001;  
 proxy\_http\_version 1.1;  
 proxy\_set\_header Upgrade \$http\_upgrade;  
 proxy\_set\_header Connection 'upgrade';  
 proxy\_set\_header Host \$host;  
 proxy\_cache\_bypass \$http\_upgrade;  
 }  
}  
EOF  
  
# Start and enable Nginx  
sudo systemctl start nginx  
sudo systemctl enable nginx

## Maintenance

### Regular Tasks

* Monitor PM2 processes: sudo pm2 status
* Check logs regularly: sudo pm2 logs
* Update dependencies periodically
* Monitor disk space and memory usage
* Backup application data

### Updates

To update the application: 1. Upload new files to S3 2. Run the deployment script again 3. The script will backup the existing installation automatically

## Support

For issues or questions, refer to: - Application logs in /var/log/ - PM2 logs: sudo pm2 logs - Database connectivity - AWS CloudWatch (if configured)