



GUÍA COMPLETA DE DESPLIEGUE - COMODÍN IA



ÍNDICE

1. [Preparación del Servidor VPS](#)
 2. [Instalación de Dependencias](#)
 3. [Configuración de Archivos](#)
 4. [Despliegue con Docker](#)
 5. [Configuración de WhatsApp \(Evolution API\)](#)
 6. [Configuración de Dominio y SSL](#)
 7. [Verificación y Pruebas](#)
 8. [Monitoreo y Mantenimiento](#)
-

1. PREPARACIÓN DEL SERVIDOR VPS

1.1 Requisitos Mínimos del Servidor

```
# Especificaciones recomendadas:  
- CPU: 2 vCores mínimo, 4 vCores recomendado  
- RAM: 4GB mínimo, 8GB recomendado  
- Almacenamiento: 50GB mínimo, 100GB recomendado  
- Ancho de banda: Ilimitado  
- Sistema: Ubuntu 20.04 LTS o superior
```

1.2 Acceso al Servidor

```
# Conectarse por SSH  
ssh root@tu-servidor-ip  
  
# O si tienes usuario específico:  
ssh tu-usuario@tu-servidor-ip
```

1.3 Actualizar Sistema

```
# Actualizar paquetes  
sudo apt update && sudo apt upgrade -y  
  
# Instalar herramientas básicas  
sudo apt install -y curl wget git unzip nano htop
```

2. INSTALACIÓN DE DEPENDENCIAS

2.1 Instalar Docker

```
# Desinstalar versiones anteriores (si existen)
sudo apt remove -y docker docker-engine docker.io containerd runc

# Instalar dependencias
sudo apt install -y ca-certificates curl gnupg lsb-release

# Agregar clave GPG de Docker
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/
apt/keyrings/docker.gpg

# Agregar repositorio de Docker
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] ht-
tps://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/
null

# Instalar Docker
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Verificar instalación
sudo docker --version
sudo docker compose version
```

2.2 Configurar Docker

```
# Agregar usuario al grupo docker (opcional)
sudo usermod -aG docker $USER

# Habilitar Docker al inicio
sudo systemctl enable docker
sudo systemctl start docker

# Verificar que Docker esté funcionando
sudo docker run hello-world
```

2.3 Instalar Node.js (para desarrollo local)

```
# Instalar Node.js 18
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs

# Verificar instalación
node --version
npm --version
```

3. CONFIGURACIÓN DE ARCHIVOS

3.1 Métodos para Subir Archivos al Servidor

Opción A: GitHub (RECOMENDADO)

```
# En tu servidor, clonar el repositorio
cd /opt
sudo mkdir comodin-ia
sudo chown $USER:$USER /opt/comodin-ia
cd /opt/comodin-ia

# Si tienes GitHub configurado:
git clone https://github.com/tu-usuario/comodin-ia.git .

# Si no tienes Git configurado, continúa con Opción B o C
```

Opción B: SCP (Transferencia Directa)

```
# Desde tu computadora local, comprimir el proyecto
tar -czf comodin_ia.tar.gz -C /home/ubuntu comodin_ia/

# Subir al servidor
scp comodin_ia.tar.gz root@tu-servidor-ip:/opt/

# En el servidor, extraer
cd /opt
tar -xzf comodin_ia.tar.gz
mv comodin_ia comodin-ia
```

Opción C: SFTP con Cliente Visual

```
# Usar FileZilla, WinSCP, o similar
# Conectar con:
Host: tu-servidor-ip
Usuario: root (o tu usuario)
Puerto: 22
Protocolo: SFTP

# Subir la carpeta completa a /opt/comodin-ia/
```

3.2 Verificar Estructura de Archivos

```
cd /opt/comodin-ia
ls -la

# Deberías ver:
# - app/ (código Next.js)
# - docker-compose.yml
# - docker-compose.evolution.yml
# - nginx.conf
# - production.env
# - scripts/
```

4. DESPLIEGUE CON DOCKER

4.1 Configurar Variables de Entorno

```
cd /opt/comodin-ia

# Copiar y editar variables de entorno
cp production.env .env

# Editar con tu información real
nano .env
```

Actualizar estas variables OBLIGATORIAS:

```
# Cambiar a tu dominio
NEXTAUTH_URL=https://tu-dominio.com

# Configurar URLs de producción
EVOLUTION_API_URL=http://evolution-api:8080
WEBHOOK_GLOBAL_URL=https://tu-dominio.com/api/whatsapp/evolution/webhook

# Tu información SMTP real
SMTP_HOST=smtp.gmail.com
SMTP_USER=tu-email@gmail.com
SMTP_PASSWORD=tu-app-password
SMTP_FROM_EMAIL=noreply@tu-dominio.com

# Generar secreto seguro (ejecutar: openssl rand -base64 32)
NEXTAUTH_SECRET=tu-secreto-super-seguro-aqui
ENCRYPTION_SECRET=otro-secreto-para-encryptacion
```

4.2 Construir y Levantar Servicios

```
# Dar permisos a scripts
chmod +x scripts/*.sh

# Levantar Evolution API primero
sudo docker compose -f docker-compose.evolution.yml up -d

# Esperar que Redis esté listo
sleep 30

# Levantar aplicación principal
sudo docker compose up --build -d

# Verificar que todo esté funcionando
sudo docker ps
```

4.3 Verificar Logs

```
# Ver logs de la aplicación
sudo docker logs comodin-ia-app -f

# Ver logs de Evolution API
sudo docker logs evolution-api -f

# Ver logs de Nginx
sudo docker logs comodin-ia-nginx -f
```

5. CONFIGURACIÓN DE WHATSAPP (Evolution API)

5.1 Verificar Evolution API

```
# Comprobar que Evolution API está funcionando
curl http://localhost:8080

# Deberías ver una respuesta JSON
```

5.2 Configurar desde el Panel Web

1. Abrir tu navegador y ir a: `http://tu-servidor-ip:3000`
2. Hacer login como Super Admin
3. Ir a **Configuraciones > WhatsApp**
4. Hacer clic en **“Conectar WhatsApp”**
5. Escanear el código QR con tu WhatsApp Business
6. Verificar que el estado cambie a **“Conectado”**

5.3 Verificación Manual (Opcional)

```
# Probar API directamente
curl -X GET \
  'http://localhost:8080/instance/fetchInstances' \
  -H 'apiKey: comodin_ia_evolution_api_key_2024'

# Deberías ver las instancias creadas
```

6. CONFIGURACIÓN DE DOMINIO Y SSL

6.1 Configurar DNS

En tu proveedor de dominio (Namecheap, GoDaddy, etc.):

```
# Registros A
@ -> tu-servidor-ip
www -> tu-servidor-ip
api -> tu-servidor-ip

# Registro CNAME (opcional)
whatsapp -> tu-dominio.com
```

6.2 Instalar Certificado SSL con Certbot

```
# Instalar Certbot
sudo apt install -y certbot python3-certbot-nginx

# Parar Nginx temporalmente
sudo docker stop comodin-ia-nginx

# Obtener certificado
sudo certbot certonly --standalone -d tu-dominio.com -d www.tu-dominio.com

# Verificar certificados
sudo ls -la /etc/letsencrypt/live/tu-dominio.com/
```

6.3 Actualizar Configuración de Nginx

```
# Editar configuración de Nginx
nano nginx.conf

# Descomentar secciones SSL y actualizar:
server_name tu-dominio.com www.tu-dominio.com;
ssl_certificate /etc/letsencrypt/live/tu-dominio.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/tu-dominio.com/privkey.pem;
```

6.4 Reiniciar con SSL

```
# Actualizar docker-compose.yml para montar certificados
nano docker-compose.yml

# Agregar volumen SSL:
volumes:
  - /etc/letsencrypt:/etc/letsencrypt:ro

# Reiniciar servicios
sudo docker compose up -d --force-recreate
```

6.5 Configurar Renovación Automática de SSL

```
# Agregar cron job
sudo crontab -e

# Agregar línea:
0 12 * * * /usr/bin/certbot renew --quiet && docker restart comodin-ia-nginx
```

7. VERIFICACIÓN Y PRUEBAS

7.1 Verificar Servicios

```
# Comprobar puertos
sudo netstat -tlnp | grep -E ':(80|443|3000|8080|5432) '

# Verificar contenedores
sudo docker ps

# Verificar salud de servicios
sudo docker healthcheck ls
```

7.2 Pruebas de Funcionalidad

```
# Probar aplicación web
curl -I https://tu-dominio.com

# Probar API
curl https://tu-dominio.com/api/health

# Probar Evolution API
curl http://localhost:8080/instance/fetchInstances \
-H "apiKey: comodin_ia_evolution_api_key_2024"
```

7.3 Verificar Base de Datos

```
# Conectar a la base de datos (si es necesario)
sudo docker exec -it postgres psql -U usuario -d comodin_ia

# Verificar tablas principales
\d

# Salir
\q
```

8. MONITOREO Y MANTENIMIENTO

8.1 Configurar Monitoreo Básico

```
# Crear script de monitoreo
sudo nano /opt/monitor.sh
```

```
#!/bin/bash
# Script de monitoreo básico

echo "=== ESTADO DE SERVICIOS $(date) ===" >> /var/log/comodin-monitor.log

# Verificar contenedores Docker
docker ps --format "table {{.Names}}\t{{.Status}}" >> /var/log/comodin-monitor.log

# Verificar uso de disco
df -h >> /var/log/comodin-monitor.log

# Verificar memoria
free -h >> /var/log/comodin-monitor.log

echo "===== " >> /var/log/comodin-monitor.log
```

```
# Hacer ejecutable
sudo chmod +x /opt/monitor.sh

# Agregar a cron
sudo crontab -e

# Ejecutar cada hora
0 * * * * /opt/monitor.sh
```

8.2 Configurar Backups

```
# Script de backup
sudo nano /opt/backup.sh
```

```
#!/bin/bash
BACKUP_DIR="/opt/backups"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# Backup de base de datos
docker exec postgres pg_dump -U usuario comodin_ia > $BACKUP_DIR/db_$DATE.sql

# Backup de configuraciones
tar -czf $BACKUP_DIR/config_$DATE.tar.gz /opt/comodin-ia/.env /opt/comodin-ia/docker-
compose.yml

# Limpiar backups antiguos (>7 días)
find $BACKUP_DIR -name "*.sql" -mtime +7 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +7 -delete
```

```
# Hacer ejecutable
sudo chmod +x /opt/backup.sh

# Backup diario a las 2 AM
0 2 * * * /opt/backup.sh
```


8.3 Comandos Útiles de Mantenimiento

```
# Reiniciar todos los servicios
sudo docker compose restart

# Ver logs en tiempo real
sudo docker compose logs -f

# Actualizar aplicación (si hay cambios)
cd /opt/comodin-ia
git pull
sudo docker compose up --build -d

# Limpiar Docker (liberar espacio)
sudo docker system prune -f

# Verificar espacio en disco
df -h

# Verificar memoria
free -h

# Ver procesos de Docker
sudo docker stats
```

RESUMEN RÁPIDO PARA DESPLEGAR

Comandos Rápidos (Solo para Expertos)

```
# 1. Preparar servidor
sudo apt update && sudo apt upgrade -y
sudo apt install -y docker.io docker-compose-plugin

# 2. Subir archivos
scp -r comodin_ia/ root@servidor:/opt/comodin-ia/

# 3. Configurar
cd /opt/comodin-ia
cp production.env .env
nano .env # Actualizar variables

# 4. Desplegar
sudo docker compose -f docker-compose.evolution.yml up -d
sleep 30
sudo docker compose up --build -d

# 5. Verificar
sudo docker ps
curl http://localhost:3000
```

SOLUCIÓN DE PROBLEMAS

Problemas Comunes

Error: “Port already in use”

```
sudo netstat -tlnp | grep :3000
sudo kill -9 <PID>
sudo docker compose up -d
```

Error: “Permission denied”

```
sudo chown -R $USER:$USER /opt/comodin-ia
chmod +x scripts/*.sh
```

WhatsApp no conecta

```
# Verificar Evolution API
curl http://localhost:8080
sudo docker logs evolution-api -f

# Verificar webhook
curl -I https://tu-dominio.com/api/whatsapp/evolution/webhook
```

Error de SSL

```
sudo certbot certificates
sudo nginx -t
sudo docker restart comodin-ia-nginx
```

SOPORTE

Si tienes problemas con el despliegue:

1. **Revisa los logs:** `sudo docker compose logs -f`
 2. **Verifica la configuración:** Asegúrate de que todas las variables de entorno estén correctas
 3. **Contacta soporte:** Proporciona los logs específicos del error
-

¡Tu COMODÍN IA está listo para funcionar! 🚀