

EscalaFin Multi-Instance Deployment Guide

Casos de Uso

Multi-Tenant por Cliente

- `cliente1.escalafin.com` → Puerto 3001
- `cliente2.escalafin.com` → Puerto 3002
- `cliente3.escalafin.com` → Puerto 3003

Multi-Región

- `mx.escalafin.com` → México
- `co.escalafin.com` → Colombia
- `pe.escalafin.com` → Perú

Entornos Separados

- `dev.escalafin.com` → Desarrollo
 - `staging.escalafin.com` → Staging
 - `escalafin.com` → Producción
-

Uso Rápido

Crear Instancia Individual

```
cd /home/ubuntu/escalafin_mvp
./coolify-multi-instance.sh create
```

Crear Múltiples Instancias

```
./coolify-multi-instance.sh multi
# Te preguntará: cuántas, prefijo, dominio base, puerto inicial
```

Ver Estado

```
./coolify-multi-instance.sh status
```

Proceso Automático

1. Preparación por Instancia

-  Variables de entorno únicas

- ☒ Base de datos separada
- ☒ Redis independiente
- ☒ Archivos S3 en carpetas separadas
- ☒ Configuración WhatsApp única

2. Despliegue en Coolify

- ☒ Proyecto independiente por instancia
- ☒ Dominio SSL automático
- ☒ Monitoreo separado
- ☒ Logs independientes

3. Aislamiento Total

- ☒ Base de datos: `escalafin_cliente1`, `escalafin_cliente2` ...
- ☒ Redis: Passwords únicos por instancia
- ☒ S3: Carpetas `cliente1/`, `cliente2/` ...
- ☒ WhatsApp: Instancias separadas



Estructura Generada

```
/opt/coolify/instances/
├── cliente1/
│   ├── docker-compose.cliente1.yml
│   ├── .env.cliente1
│   └── [archivos app]
├── cliente2/
│   ├── docker-compose.cliente2.yml
│   ├── .env.cliente2
│   └── [archivos app]
└── cliente3/
    ├── docker-compose.cliente3.yml
    ├── .env.cliente3
    └── [archivos app]
```



Configuración por Instancia

Variables Únicas Automáticas:

```
# Base
NEXTAUTH_SECRET=<único_por_instancia>
DATABASE_URL=postgresql://escalafin_cliente1:...@db_cliente1:5432/escalafin_cliente1

# Servicios
AWS_BUCKET_NAME=cliente1-files
AWS_FOLDER_PREFIX=cliente1/
EVOLUTION_INSTANCE_NAME=cliente1
```

Servicios Separados:

- **App:** `escalafin_app_cliente1` (Puerto 3001)
- **DB:** `escalafin_db_cliente1`
- **Redis:** `escalafin_redis_cliente1`

URLs por Instancia

Instancia	URL	Admin
cliente1	<code>https://cliente1.escalafin.com</code>	<code>https://cliente1.escalafin.com/admin</code>
cliente2	<code>https://cliente2.escalafin.com</code>	<code>https://cliente2.escalafin.com/admin</code>
cliente3	<code>https://cliente3.escalafin.com</code>	<code>https://cliente3.escalafin.com/admin</code>

Monitoreo en Coolify

Por Instancia:

- Logs independientes
- Métricas separadas
- Health checks individuales
- Alertas configurables

Dashboard:

- Ver todas las instancias desde un panel
- Estado de cada servicio
- Uso de recursos por instancia

Operaciones

Crear Nueva Instancia:

```
./coolify-multi-instance.sh create
# Nombre: cliente4
# Dominio: cliente4.escalafin.com
# Puerto: 3004
```

Actualizar Todas:

```
# El script puede actualizar en lote
for instance in cliente1 cliente2 cliente3; do
    ssh root@adm.escalafin.com "cd /opt/coolify/instances/$instance && docker-compose
pull && docker-compose up -d"
done
```

Backup por Instancia:

```
# Cada instancia tiene su propia BD
ssh root@adm.escalafin.com "docker exec escalafin_db_cliente1 pg_dump -U escal-
afin_cliente1 escalafin_cliente1 > backup_cliente1_$(date +%Y%m%d).sql"
```

Ventajas del Multi-Instance

Aislamiento Total

- Cada cliente tiene su propia instancia completa
- Falla de una instancia no afecta otras
- Datos completamente separados

Escalabilidad

- Agregar nuevos clientes es automático
- Recursos dedicados por instancia
- Fácil migración individual

Personalización

- Variables específicas por cliente
- Configuraciones únicas
- Dominios personalizados

Mantenimiento

- Updates por instancia
- Rollbacks individuales
- Testing independiente

Solución de Problemas

Ver Logs de Instancia:

```
ssh root@adm.escalafin.com "cd /opt/coolify/instances/cliente1 && docker-compose logs
-f"
```

Reiniciar Instancia:

```
ssh root@adm.escalafin.com "cd /opt/coolify/instances/clientel && docker-compose re-start"
```

Estado de Todas:

```
./coolify-multi-instance.sh status
```

 **¡Listo para Multi-Instance con Coolify!**

Tu sistema ahora es completamente escalable para múltiples clientes o regiones.