M. Valentina Castro, David Molta

# Guía de Despliegue de la Infraestructura

## Archivos Utilizados

La infraestructura está dividida en 5 stacks independientes, cada uno representado por un archivo .yml, y un archivo de parámetros general:

| Archivo | Descripción breve |
|---|---|
| network.yml | Define toda la infraestructura de red base |
| security.yml | Crea los grupos de seguridad necesarios |
| database.yml | Despliega la base de datos MySQL |
| compute.yml | Configura instancias, ALB y Auto Scaling |
| monitoring.yml | Añade monitoreo, alertas y auditoría |
| params.json | Contiene parámetros reutilizables como tipo de instancia, ambiente, credenciales, etc. |
| deploy-all.sh | Script que ejecuta todos los stacks en orden, mostrando el progreso |

## Descripción General de Cada Stack

### 1. network.yml

Este stack crea toda la red base del sistema, incluyendo:

- VPC principal
- Subredes públicas, privadas y de base de datos
- Internet Gateway, NAT Gateway
- Tablas de enrutamiento y asociaciones

Exporta valores como el ID de la VPC y los IDs de subred que serán utilizados por los demás stacks.

Debe desplegarse primero.

### 2. security.yml

Define los grupos de seguridad para controlar el acceso:

- ALB (HTTP/HTTPS)

- Web Servers (Node.js + Nginx)
- Bastion Host (SSH)
- Base de datos MySQL

Utiliza !ImportValue para obtener la VPC definida en network.yml.

Se despliega después del stack de red.

**3. database.yml**

Despliega la instancia MySQL usando Amazon RDS:

- Instancia privada
- DB Subnet Group
- Backup, seguridad y configuración

Depende de las subredes creadas por el stack de red y del grupo de seguridad definido en security.yml.

Se despliega como tercer paso.

**4. compute.yml**

Despliega los recursos de cómputo:

- Bastion Host (EC2 t2.micro)
- Launch Template con UserData que:
    - Instala Node.js y Nginx
    - Clona los repositorios del frontend y backend
    - Levanta los servicios automáticamente
- Auto Scaling Group
- Application Load Balancer (ALB)
- Target Group para distribuir el tráfico a las instancias

Depende de los valores exportados por network.yml, security.yml y el endpoint de la base de datos del stack database.yml.

**5. monitoring.yml**

Agrega capacidades de monitoreo y auditoría:

- Alarmas de CloudWatch (CPU alta o baja)
- Políticas de escalado automático
- SNS Topic para enviar alertas por correo
- CloudTrail para auditoría de eventos, con bucket S3 privado

Este stack depende del Auto Scaling Group del stack compute.yml.

Es el último en desplegarse.

**Flujo de Despliegue (deploy-all.sh)**

Este script automatiza el despliegue completo ejecutando los stacks en el siguiente orden:

1. ecommerce-network
2. ecommerce-security
3. ecommerce-database
4. ecommerce-compute
5. ecommerce-monitoring

El script utiliza aws cloudformation deploy con los parámetros definidos en params.json, y espera que cada stack finalice antes de continuar con el siguiente.

**Ventajas de la Estructura Modular**

- Reutilizable: permite modificar un stack sin afectar el resto.
- Escalable: permite agregar nuevos stacks fácilmente (por ejemplo, S3, Lambda).
- Mantenible: si ocurre un error, se puede identificar fácilmente en qué stack está el problema.
- Eficiente: evita duplicación de lógica entre recursos.

**Recomendaciones**

- Comenzar siempre con el stack de red (network.yml).
- Verificar los valores exportados con aws cloudformation list-exports.

- Asegurarse de que los nombres de los Export y Import coincidan exactamente entre stacks.
- Utilizar nombres de stacks únicos por entorno (por ejemplo: ecommerce-network-dev, ecommerce-network-prod).

**Resultado Esperado**

Al finalizar el despliegue, se espera que:

- La aplicación e-commerce esté completamente operativa tras el Application Load Balancer.
- El backend esté conectado a la base de datos RDS.
- El frontend responda correctamente.
- Se encuentren activadas las alarmas de CPU y el monitoreo con CloudWatch.
- CloudTrail esté registrando los eventos en el bucket de S3.
- Las notificaciones lleguen correctamente al correo configurado en SNS.

Ejemplo CloudWatch: