

Plan para el Despliegue de una Aplicación Web Escalable en AWS con IAC (E-commerce)

Infraestructura III

Andres Bueno Cardona - A00399454

Jeanpaul A. Vidales - A00395404

El objetivo principal de este proyecto es asumir el rol de ingenieros en infraestructura/Devops para así desplegar una aplicación web de comercio electrónico en AWS utilizando CloudFormation (IAC) para garantizar una infraestructura segura, escalable y reproducible.

Se definieron los siguientes objetivos específicos para que guíen la implementación y sirvan como criterios de éxito directo, cumpliendo con los requisitos solicitados por la entrega:

- I. **Usar infraestructura como Código (IaC).** Crear un conjunto de templates de AWS CloudFormation (en formato YAML) que definan y aprovisionan el 100% de la infraestructura de la aplicación, garantizando que el despliegue sea reproducible y automatizado. Estructurando los templates de CloudFormation de manera modular (e.g., separando VPC, RDS y Aplicación) para mejorar la legibilidad y facilitar el mantenimiento.
- II. **Diseño de Arquitectura Escalable y Robusta.** Implementar Escalabilidad Horizontal, mediante la configuración de un Auto Scaling Group para las instancias EC2 de la aplicación, el cual debe escalar automáticamente basándose en métricas de rendimiento (ej. utilización de CPU), y colocarlo detrás de un Application Load Balancer para distribuir el tráfico.
- III. **Seguridad y Cumplimiento.** Aislar Componentes Críticos, se desplegará la base de datos (AWS RDS) y los servidores de aplicación (EC2) exclusivamente en subredes privadas de la VPC, restringiendo el acceso directo desde Internet. Se va a garantizar Acceso Seguro a EC2, configurando un Bastión Host en la subred pública y utilizar AWS Systems Manager Session Manager como el método principal y seguro para acceder a las instancias privadas. Se van a definir y aplicar grupos de Seguridad estrictos que solo permitan la comunicación necesaria entre las capas (e.g., ALB a EC2 en puerto 3000, EC2 a RDS en puerto 5432/3306).
- IV. **Monitoreo y Operaciones.** Configurar Monitoreo de Rendimiento, utilizando AWS CloudWatch para recopilar métricas y crear alarmas automáticas sobre eventos críticos (ej. alta utilización de CPU, fallos del sistema o errores del ALB). Se

implementarán notificaciones para integrar AWS SNS para enviar notificaciones automáticas a un topic específico cuando una alarma de CloudWatch cambie de estado (ej. de OK a ALARMA).

Los anteriores objetivos permiten tener una definición clara del alcance del proyecto, el cual nos permite saber precisamente que está dentro y fuera de los límites de la entrega. El proyecto se centra en el diseño, implementación y automatización (IaC) del entorno de infraestructura necesario para soportar la aplicación de comercio electrónico en AWS.

I. Dentro del alcance:

- A. El desarrollo de la aplicación mínima viable (MVP) con las funcionalidades requeridas (Catálogo, Carrito, Pago Simulado, Registro) que interactúe correctamente con la base de datos RDS.
- B. Todo el aprovisionamiento de recursos de AWS debe realizarse mediante CloudFormation. Esto incluye: VPC, subredes, grupos de seguridad, RDS, ALB, ASG, Launch Templates y Bastión Host.
- C. Despliegue de una arquitectura de alta disponibilidad y escalable de tres capas (web/app, base de datos), con recursos distribuidos en múltiples Zonas de Disponibilidad (AZs), utilizando ALB y ASG.
- D. Implementación de una VPC privada con subredes separadas. Uso estricto de Grupos de Seguridad con el principio de mínimo privilegio. Configuración de Bastión Host para acceso seguro a recursos privados.
- E. Uso de scripts de UserData para la configuración automática de software (Node.js, dependencias, Express.js) en las instancias EC2, permitiendo el escalado rápido.
- F. Configuración de Alarmas de CloudWatch vinculadas a un tema de SNS para notificar sobre métricas críticas del sistema (ej. alta CPU, fallas de salud del ALB).
- G. Generación de la Guía de Despliegue y la Presentación final, incluyendo la explicación del diseño y las lecciones aprendidas.

II. Fuera del alcance:

- A. No se implementarán servicios adicionales no especificados en el requerimiento, tales como: AWS Lambda, ECS/EKS (orquestración de contenedores), servicios de *data warehousing*, o *Web Application Firewall* (WAF).
- B. Debido a las restricciones del entorno de Sandbox/Educativo, la instancia de AWS RDS NO será Multi-AZ (replicación síncrona). Se desplegará en una sola AZ, aunque en subredes privadas.
- C. No se incluye la compra, registro o configuración de un nombre de dominio personalizado a través de Route53. La aplicación será accesible a través del DNS público del Application Load Balancer (ALB).
- D. El código de la aplicación de e-commerce se limitará a las funcionalidades básicas (MVP). No se incluye el desarrollo de características avanzadas (ej. gestión de inventario, reseñas, autenticación avanzada como Cognito).

- E. El proyecto no incluye la implementación de pipelines de Integración Continua/Despliegue Continuo (CI/CD) completas (ej. CodePipeline). El despliegue de la infraestructura se hará manualmente a través de la CLI/Consola de CloudFormation.

Para asegurar la completitud de las tareas e ir midiendo el progreso de completitud del proyecto a lo largo del tiempo, se decidió establecer el siguiente cronograma de actividades:

Día	Tarea Principal	Tareas Específicas
Día 1	Planificación y Repositorio	Revisar el plan final, crear la estructura de carpetas, inicializar Git y subir el <code>README.md</code> inicial.
Día 2-3	Diseño de la Arquitectura & VPC (IaC)	Diseñar el diagrama de la arquitectura. Crear el template <code>vpc.yaml</code> de CloudFormation (VPC, Subredes Públicas/Privadas, IGW, NAT Gateway).
Día 4-5	Desarrollo del Back-end (MVP)	Implementar la API REST de Node.js/Express.js para Catálogo y Carrito (sin DB aún). Definir el ORM (Sequelize).
Día 6	Front-end y Scripts Iniciales	Crear la estructura básica del Front-end (HTML/CSS) y el script de instalación (<code>setup_instance.sh</code>) para Node.js.
Día 7	Revisión y Sincronización	Revisión de código (Back-end y <code>vpc.yaml</code>). Merge de ramas y

		preparación para la siguiente semana.
--	--	---------------------------------------

Para la segunda semana:

Día	Tarea Principal	Tareas Específicas
Día 8-9	CloudFormation: RDS	Crear el template <code>database.yaml</code> para AWS RDS (en subredes privadas). Definir los parámetros de seguridad.
Día 10	Integración de Base de Datos	Conectar el Back-end (Node.js/ORM) al RDS. Crear el script SQL de inicialización (<code>db_schema.sql</code>).
Día 11-12	CloudFormation: Aplicación (ASG/ALB)	Crear el template <code>application.yaml</code> (Launch Template, ASG, ALB, Target Groups). Integrar <code>UserData</code> para el despliegue de la aplicación.
Día 13	Seguridad y Acceso	Configurar el Bastión Host (template de CFN o manual, según restricción). Configurar grupos de seguridad (mínimo privilegio).

Día 14	Primer Despliegue y Pruebas	Despliegue completo del Stack de CloudFormation. Pruebas de conectividad entre el ALB, EC2 y RDS. Corrección de errores iniciales.
---------------	-----------------------------	--

Finalmente para la semana 3:

Día	Tarea Principal	Tareas Específicas
Día 15-16	Monitoreo y Notificaciones	Configurar Alarmas de CloudWatch (CPU, errores 5xx del ALB). Configurar SNS Topic y vincularlo a las alarmas.
Día 17	Pruebas de Escalabilidad	Probar el Auto Scaling Group (forzar el aumento de CPU o cambiar los límites del ASG) para verificar que se creen y eliminen instancias correctamente.
Día 18-19	Documentación (Guía de Despliegue)	Redactar la Guía de Despliegue detallada (<code>deployment_guide.md</code>), incluyendo pasos, comandos de CloudFormation y resolución de problemas.

Día 20	Preparación de la Presentación	Crear las diapositivas para la presentación final. Enfocarse en la explicación del diseño, la justificación de la arquitectura (escalabilidad/seguridad) y las lecciones aprendidas.
Día 21	Revisión Final y Entrega	Revisión final de todos los entregables (código, IaC, documentación, presentación). Entrega del Proyecto.

Finalmente se van a definir los recursos del proyecto, se van a enumerar las herramientas y tecnologías necesarias para ejecutar la implementación del proyecto, reconociendo que hay unas limitaciones en el sandbox de AWS:

Categoría	Herramientas Específicas	Propósito en el Proyecto
Nube	AWS (Entorno Sandbox/Educativo)	Plataforma de <i>cloud computing</i> para el despliegue de toda la infraestructura.
Infraestructura como Código (IaC)	AWS CloudFormation (YAML)	Herramienta principal para definir, aprovisionar y gestionar la infraestructura de manera automatizada.
Control de Versiones	Git / GitHub	Control de versiones del código de la aplicación, los <i>templates</i> de CloudFormation y la documentación.

Lenguajes y Frameworks	Node.js, Express.js, JavaScript, HTML/CSS/jQuery, SQL (para RDS)	Tecnologías utilizadas para construir la aplicación.
-------------------------------	--	--

En cuanto a los recursos usados en la nube:

Servicio AWS	Tipo de Recurso Clave	Limitación/Consideración del Sandbox
Computación	EC2 (t2.micro/t2.medium), Auto Scaling Group	Límite máximo de instancias EC2. Se debe utilizar el tipo de instancia más pequeño posible (t2.micro).
Bases de Datos	AWS RDS (db.t3.micro a db.t3.medium)	No se permite el despliegue Multi-AZ (alta disponibilidad con réplica). La instancia debe ser Single-AZ.
Red y Tráfico	VPC, ALB (Application Load Balancer)	La creación de recursos NAT Gateway tiene un costo que debe ser monitoreado; en caso de restricciones de presupuesto, se puede usar un EC2 NAT-Instance.
Identidad y Acceso	IAM, Systems Manager Session Manager	El entorno Sandbox a menudo impone permisos de solo lectura (Read-only) en la cuenta IAM para el usuario. NO se permite crear y registrar dominios en Route53.
Monitoreo	CloudWatch, SNS	Se deben configurar alarmas y notificaciones para no depender de la supervisión constante y cumplir con los requisitos.

