



## ENTREGA 2 - DESPLIEGUE BÁSICO EN LA NUBE

### MIGRACIÓN DE UNA APLICACIÓN WEB A LA NUBE PÚBLICA

#### Objetivos

- Analizar y evaluar los requisitos técnicos y funcionales de una aplicación web desarrollada en un entorno tradicional, con el fin de diseñar una arquitectura óptima para su aprovisionamiento y despliegue a la nube pública de Amazon Web Services (AWS), teniendo en cuenta aspectos rendimiento y seguridad.
- Implementar y parametrizar instancias de cómputo Amazon EC2 en AWS para el despliegue de componentes críticos de la aplicación web, incluyendo el servidor web, el componente worker y el sistema de archivos, asegurando su correcta integración y operación continua.
- Configurar y administrar servicios de bases de datos autogestionados en la nube pública, utilizando Amazon RDS para garantizar la persistencia, disponibilidad y rendimiento de los datos de la aplicación web, con una gestión eficiente de los recursos.
- Garantizar que la arquitectura desplegada en AWS cumpla con los requisitos funcionales y de desempeño definidos para la aplicación web, a través de la monitorización continua, pruebas de carga y ajustes de configuración según sea necesario.

#### Tiempo de Dedicación

La presente entrega, correspondiente a la fase implantación en la nube, está programada para un periodo de **tres semanas**. Durante este tiempo, cada estudiante deberá destinar las horas asignadas en la planificación semanal. Se recuerda la importancia de conformar equipos de trabajo de acuerdo con las directrices establecidas en el curso, como condición clave para el éxito de la actividad.

#### Lecturas Previas

Material de lectura entregado durante el curso, en conjunto con la documentación ofrecida por AWS para los servicios de Amazon EC2 y Amazon RDS.



## Esquema de Evaluación

La calificación de la entrega se distribuye de la siguiente manera:

- Actividades requeridas para la migración inicial de la aplicación: **70%**
- Documento de escenarios y resultados de las pruebas de estrés: **20%**
- Documento de la arquitectura de la aplicación: **10%**

## Recomendaciones y Consideraciones

En esta entrega, se realizará el aprovisionamiento y despliegue en AWS de la aplicación web desarrollada previamente. La aplicación deberá contar, como mínimo, con cuatro componentes esenciales para su ejecución en AWS: servidor web, servidor NFS, worker y base de datos. Se podrá utilizar cualquier sistema de gestión de bases de datos que sea compatible con el servicio Amazon RDS.

En la fase de desarrollo, se sugiere emplear un contenedor docker con el mismo sistema de gestión de bases de datos que se utilizará en Amazon RDS. Cabe resaltar que NO se implementarán mecanismos de escalado automático (autoscaling) en esta entrega, NI almacenamiento en buckets de Amazon S3.

Por otra parte, se recomienda activar las instancias en AWS únicamente durante la ejecución de las pruebas de capacidad y detenerlas cuando no sean necesarias, con el objetivo de optimizar el uso de los créditos asignados. Asimismo, al finalizar y cargar la entrega, se deberá eliminar la base de datos de Amazon RDS, dado que este servicio genera un costo elevado. En caso de requerirse una sustentación síncrona, será obligatorio recrear la base de datos antes del encuentro con los tutores.

Dado que durante el proyecto se gestionarán credenciales de acceso a AWS, se aconseja no almacenarlas en texto plano dentro del código fuente de la aplicación, sino configurar variables de entorno para su manejo seguro. Para mayor protección y control de costos, se recomienda habilitar las alarmas de consumo y los presupuestos, con el fin de monitorear continuamente los gastos asociados a la cuenta de AWS.



## Modelo de Despliegue Básico en la Nube Publica de AWS

La compañía planea mover su aplicación a un proveedor público de IaaS, específicamente AWS. En esta primera migración, se ha decidido aprovechar los siguientes servicios:

- **Amazon EC2:** Ejecución del servidor web, el worker y el almacenamiento de tipo NFS. Por decisión de negocio, se han seleccionado instancias de cómputo con 2 vCPU, 2 GiB de RAM y 30 GiB de almacenamiento.
- **Amazon RDS:** Almacenamiento de la información de la aplicación en una base de datos relacional. Se recomienda establecer el entorno de desarrollo como ajuste predeterminado para minimizar costos. En las fases iniciales de la implementación, se puede utilizar una instancia de Amazon EC2 con un contenedor de base de datos y sustituirla por Amazon RDS únicamente durante las pruebas de estrés.

Además, el modelo de la aplicación definido por la empresa contempla las siguientes actividades:

**(50%) Despliegue del componente web, el worker y el sistema de almacenamiento de archivos en tres instancias de cómputo (máquinas virtuales) distintas.** Para ello, se requiere:

- a. Configurar todas las dependencias (servidor de aplicaciones, librerías, etc.) y herramientas necesarias (reglas de firewall, llaves de acceso, etc.) para exponer la aplicación web en una instancia denominada **Web Server**.
- b. Configurar el componente **Worker** que procesa los archivos en una segunda instancia de Amazon EC2.
- c. Configurar un sistema de archivos de red (NFS) en la tercera instancia de Amazon EC2. La instancia se denominará **File Server**, y tanto el componente web como el worker deberán almacenar y acceder a todos los archivos originales y procesados en este servidor. No utilice el servicio Amazon EFS.

**(10%) Configuración del servicio Amazon RDS** para administrar la base de datos de la aplicación web. No es necesario definir ningún mecanismo de replicación ni alta disponibilidad.



**(10%) La aplicación web debe cumplir con los requisitos funcionales previamente establecidos y detallados en el enunciado del proyecto.** Valide el correcto funcionamiento de cada uno de los endpoints definidos en la primera entrega.

**Nota:** Es importante configurar las políticas de red VCP y una arquitectura de subredes básica para garantizar el correcto aislamiento de los servicios.

## Documentación – Arquitectura de la Aplicación

Se deberá entregar un documento que describa la arquitectura ajustada, incluyendo una breve explicación de las tecnologías y servicios incorporados, así como una indicación clara de los cambios realizados con respecto a la entrega anterior. Es fundamental presentar tanto el modelo de despliegue como el modelo de componentes; no obstante, se pueden incluir otros recursos visuales que se consideren pertinentes para complementar la documentación.

Toda la documentación deberá estar alojada en el repositorio de Github, dentro del directorio dedicado (**/docs/entrega2**), y deberá estar referenciada en el archivo **README.md** para facilitar su acceso. Se recomienda utilizar esta estructura para las entregas posteriores.

## Documentación – Análisis de Capacidad

Lea el documento “**Entrega - Análisis de capacidad**” para conocer toda la especificación del entregable solicitado. Los porcentajes de esta actividad se dividen en los siguientes ítems de calificación:

- **(10%)** Pruebas de estrés, análisis y documentación - Escenario 1.
- **(10%)** Pruebas de estrés, análisis y documentación - Escenario 2.

Además, incorpore las conclusiones derivadas de las pruebas de estrés ejecutadas y las consideraciones adicionales para continuar escalando la aplicación web y así atender a los cientos de usuarios finales que van a subir archivos o consumir recursos del API REST de manera concurrente. En otras palabras, incluya las modificaciones, en cuanto a las características de los recursos empleados, por ejemplo, para que la siguiente versión del proyecto pueda admitir más clientes y tareas procesadas.



El **de análisis de capacidad** debe ser organizado y entregado dentro del repositorio del proyecto, dentro de la carpeta **/capacity-planning**, donde se almacenará el documento correspondiente.

El reporte debe almacenarse en un archivo llamado **pruebas\_de\_carga\_entrega2.md**, el cual debe incluir el análisis detallado de capacidad de la aplicación, los resultados de los escenarios de carga planteados y las recomendaciones para escalar la solución. Esta estructura debe mantenerse de forma consistente en las futuras entregas del proyecto.

## Entregables

- a. Aplicación desplegada y en ejecución sobre AWS.
- b. Documento con la descripción de la arquitectura de solución planteada.
- c. Documento con el plan de pruebas de carga refinado.
- d. Documento de análisis de las pruebas de carga.

Por lo cual, se solicita:

- a. Crear un release del código fuente en el repositorio del grupo en Github.
- b. Entregar toda la documentación vía Github.

Adicionalmente, cada grupo debe preparar un video de sustentación de máximo 20 minutos, donde explore la solución propuesta y efectúe pruebas sobre cada uno de los endpoints, empleando distintos parámetros de entrada. En especial, emplee vídeos con características (duración y relación de aspecto) variadas que permitan evidenciar en profundidad el comportamiento de la aplicación. A petición de los tutores del curso, podría programarse una sustentación síncrona, para la cual es indispensable que la aplicación se encuentre desplegada y en ejecución. Además, debe tener configurado el acceso a la infraestructura y los recursos necesarios para poder realizar una prueba de desempeño si así lo solicita el tutor.