

Laboratorio 2

Desafío:

Infraestructura de Alta Disponibilidad con ECS y ALB

Fecha de entrega:

Jueves 10/10 XX hs

Descripción:

Crear una infraestructura de alta disponibilidad para hostear un sitio web estático utilizando ECS y un Application Load Balancer (ALB). El sitio web deberá renderizar una página PHP que muestre un mensaje "Hola mundo" y la instancia del web server que lo está sirviendo.

Requerimientos:

1. Application Load Balancer (ALB):

- Un ALB que distribuya el tráfico entre las tareas de ECS.
- Configuración de HTTPS utilizando un certificado TLS gestionado por AWS Certificate Manager (ACM), reutilizando el certificado y dominio del Laboratorio 1.

2. ECS Cluster:

- Un ECS Cluster.
- Definir un servicio en ECS con al menos dos tareas (mínimo 2 running tasks).

3. Task Definition

- Crear una task definition con un contenedor que ejecute un servidor web con PHP.
- La página PHP debe mostrar un mensaje “Hola Mundo” junto con el ID de la instancia que está respondiendo la solicitud.

Requerimientos:

4. ECR (Elastic Container Registry):

- Construir una imagen Docker con el código de la aplicación.
- Subir la imagen a ECR utilizando tags que identifiquen la versión del código.

5. Acceso TLS:

- Crear un registro en Route53 para asociar un nombre de dominio personalizado al ALB.
- El sitio web deberá ser accesible a través de un FQDN (Fully Qualified Domain Name) usando TLS.

6. Pipeline CI/CD:

- Crear un pipeline utilizando CodePipeline, CodeBuild y CodeDeploy para automatizar la creación de la imagen Docker, el push a ECR y el deployment al ECS.
- El pipeline deberá ejecutarse automáticamente al hacer push en la rama main.

Requerimientos:

7. Organización del Código:

- El código fuente debe organizarse en carpetas utilizando el ID del ticket de trabajo.

Definición de "Done" (DoD):

- El sitio debe ser accesible a través del nombre de dominio asociado al ALB.
- Al hacer una solicitud HTTP, el ALB debe redireccionar la solicitud al listener HTTPS, y el servidor debe responder con un mensaje del tipo: “web server #”, donde # identifica la instancia que está respondiendo.
- El código debe estar organizado por carpetas según el ticket de trabajo.
- Se debe proveer la URL del ALB para verificar la disponibilidad del sitio web.
- El pipeline de CI/CD debe estar configurado para que se ejecute automáticamente cuando se haga push en la rama main.

Opcionales:

- Configurar notificaciones por correo para informar el estado de finalización del pipeline.
- Usar variables de entorno para la configuración del registry ECR.

Entregables:

1. Acceso a la Infraestructura:

- Proveer la URL pública del ALB y el nombre de dominio asociado al ALB.

2. Pipeline Funcional:

- Pipeline en funcionamiento para la construcción y despliegue automático de la aplicación.

3. Documentación:

- Incluir un diagrama de la solución propuesta, mostrando la arquitectura de la infraestructura (ALB, ECS, ECR, etc.).
- Guía paso a paso de cómo se configuró la infraestructura, incluyendo comandos usados y decisiones técnicas.

Evaluación:

- Entrega y presentación por meet en la fecha establecida (**Jueves 10/10 a las XX hs**).
- Documentación clara, organizada y comprensible por terceros.
- Cumplimiento de todos los requisitos funcionales.
- La infraestructura debe funcionar sin errores y estar accesible por las URLs correspondientes.

Clientes:

- Nicolas Cortes / Leonel Barragan y Santiago Bratti
- Ariana Vidal / Nicolas Rossi y Martin Carletti
- Tomas Kohler / Ezequiel Haiatzanian
- Ramiro Cerrato / Lourdes Dorado
- Silvio Depetri / Martin Osella
- Valentino Gabrieloni / Ignacio Rubio

***In the Cloud,
together.***



www.teracloud.io

