Propuesta de solución

Caso Práctico 1 – Apartado C

| Asignatura | Datos de los alumnos | Fecha |
| --- | --- | --- |
| Experto Universitario en DevOps & Cloud | Apellidos: Argamenteria Arce | 20240611 |
| Nombre: Daniel |

URL de repositorio solución de GitHub: https://github.com/dargamenteria/actividad1-C

Reto 1 – Creación pipeline CI

En este reto se solicita un único entregable:

* Relación de comandos, junto con sus capturas de pantalla, empleados para llevar a cabo el clonado del repositorio.

La solución se puede encontrar en esta url:

https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_1.md

Reto 2 – Infraestructura AWS

En este reto se solicitan 6 entregables:

* Pasos realizados hasta acceder a la consola de AWS.

Se pueden ver en esta url:

https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md

* Comandos y pasos realizados para la creación de la instancia EC2

La solución de puede observar en la url:

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#reto-2----infraestructura-aws](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "reto-2----infraestructura-aws)

* Comandos y salida obtenida para la comprobación de que podemos acceder a la instancia EC2 con sistema Ubuntu.

Se puede obtener en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#acceso-a-la-instancia](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "acceso-a-la-instancia)

Deben mostrarse los comandos, salidas de conexión SSH, así como los comandos uname, df y otros comandos que el alumno considere para verificar que hemos accedido a la nueva máquina EC2 creada.

Se pueden obtener en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#acceso-a-la-instancia](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "acceso-a-la-instancia)

* Integración con Cloud9

Pasos y comandos empleados para la creación de la instancia Cloud9 corriendo dentro de la máquina EC2.

Adjuntar capturas de pantalla, comandos y sus salidas, de forma cronológica.

Como en casos anteriores, no es necesario adjuntar pantallas que no aporten valor, sino únicamente nombrarlas.

La última pantalla a adjuntar debe ser la del propio Cloud9 corriendo en el navegador del alumno.

Se pueden obtener en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#integraci%C3%B3n-con-cloud9](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "integraci%C3%B3n-con-cloud9)

* Instalación de Jenkins

Pasos y comandos empleados para la instalación de Jenkins, hasta conseguir acceder vía web, desde el PC local a la página principal dentro de Jenkins.

Adjuntar capturas de pantalla, comandos y sus salidas, de forma cronológica.

Como en casos anteriores, no es necesario adjuntar pantallas que no aporten valor, sino únicamente nombrarlas.

Se pueden obener en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#integraci%C3%B3n-con-cloud9](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "integraci%C3%B3n-con-cloud9)

* Creación de IP elástica asociada a la máquina EC2

Pasos y comandos empleados para crear y asociar la IP elástica, indicando capturas de pantalla como en casos anteriores. Las capturas deben incluir una en la que se muestre que la instancia EC2 y Cloud9 están configuradas con la IP elástica.

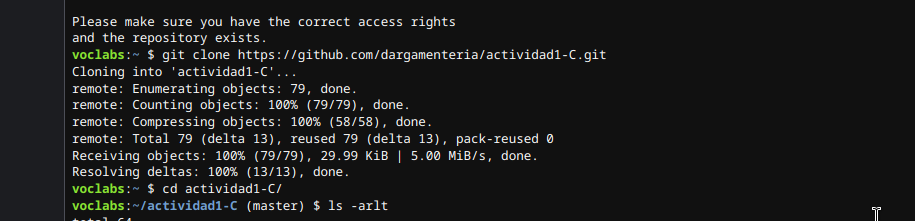
Se puede obtener en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_2.md#integraci%C3%B3n-con-cloud9](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md" \l "integraci%C3%B3n-con-cloud9)

Reto 3 – Stack Serverless

En este reto se solicitan 4 entregables:

* Descarga del repositorio del proyecto todo-list-aws, en la máquina EC2.



* Construcción de la plantilla SAM

Comandos empleados junto con sus salidas

Se pueden encontrar en esta url

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_3.md#reto-3----stack-serverless](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_3.md" \l "reto-3----stack-serverless)

* Despliegue en AWS del stack y revisión de los Outputs

Comandos y parámetros empleados, junto con sus salidas.

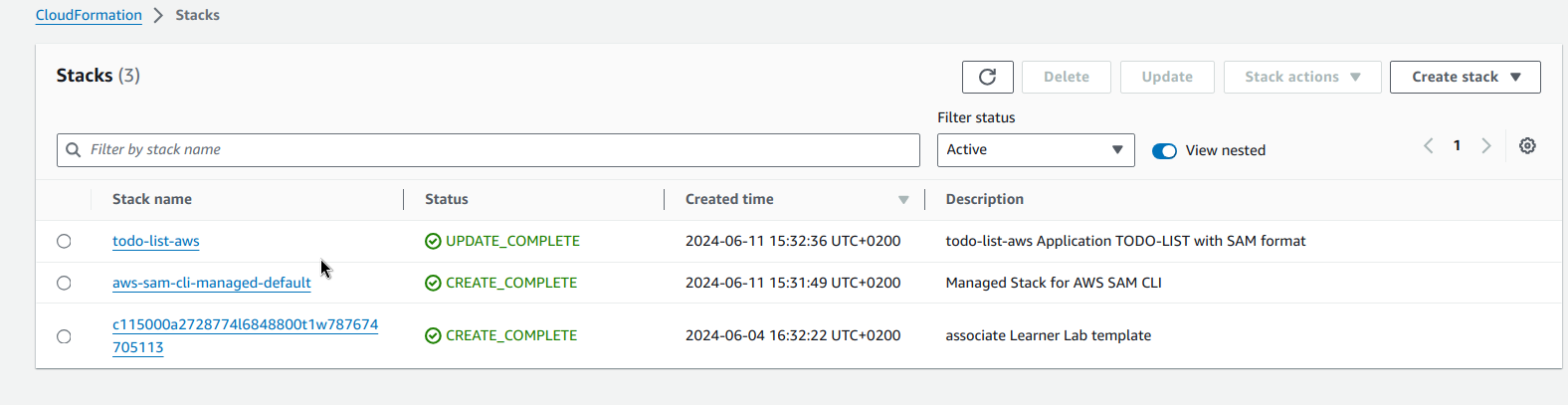
Explicación de qué nos indican las salidas “Outputs”.

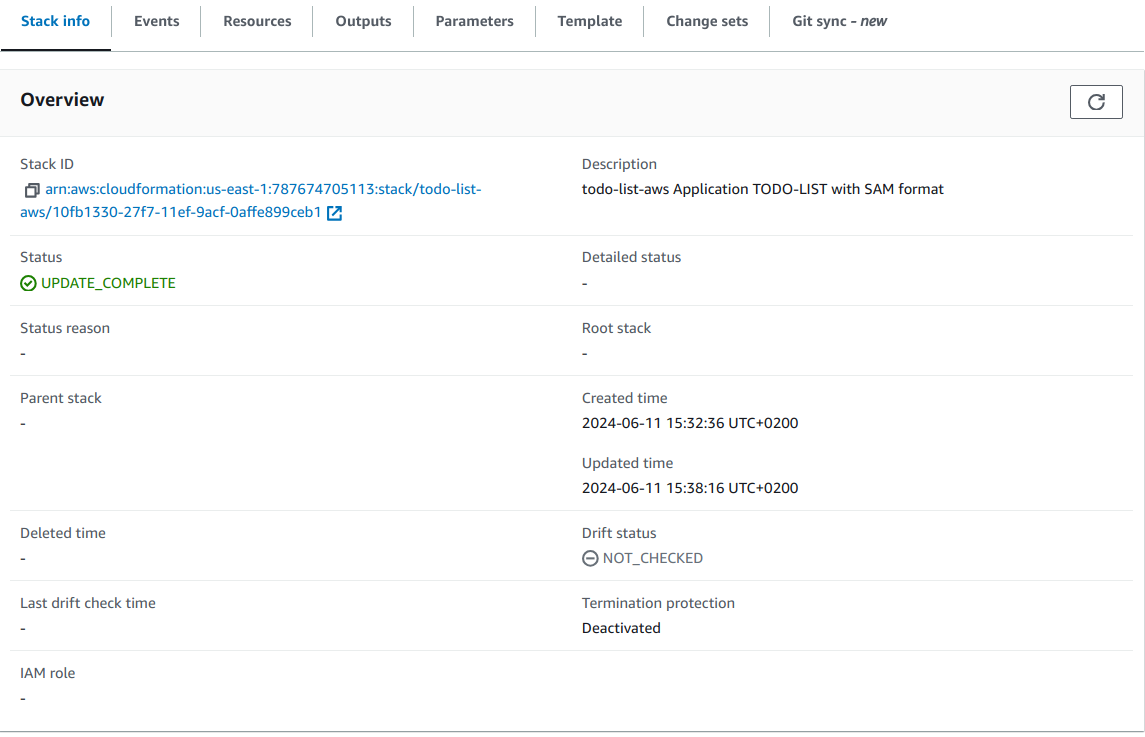
Se pueden encontrar en esta url

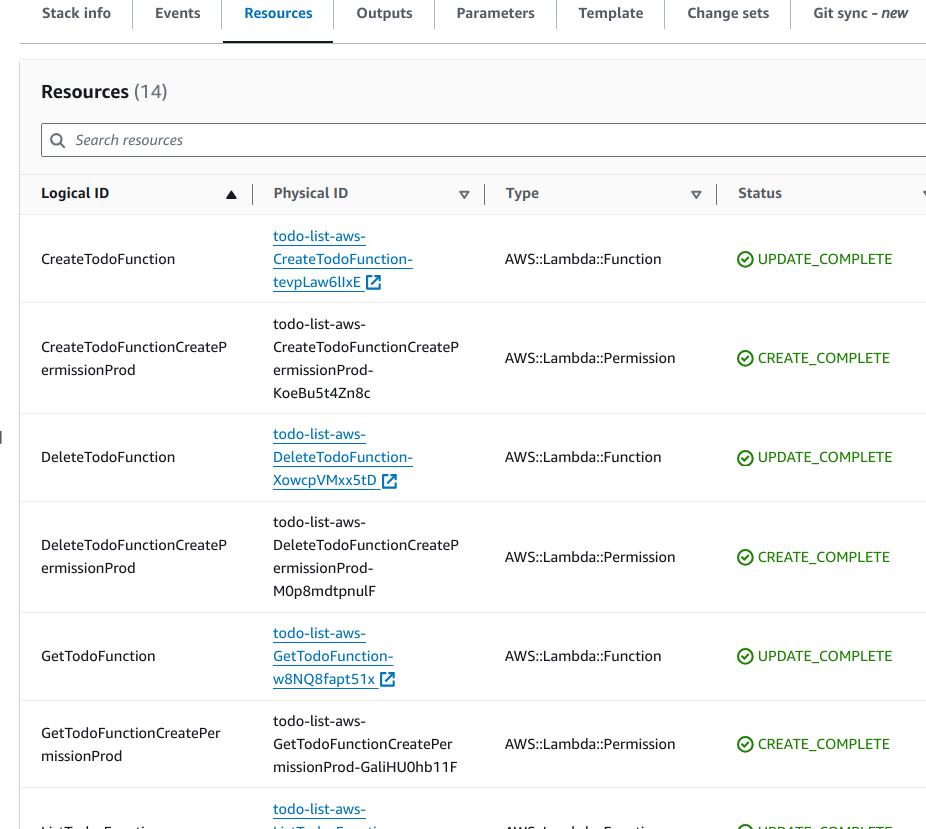
[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_3.md#reto-3----stack-serverless](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_3.md" \l "reto-3----stack-serverless)

Se observa en las salidas la creación y actualización de las distintas funciones de nuestro stack serverless, así como los endpoints que utilizará.

* Comprobación, en CloudFormation, de que el Stack se ha creado correctamente.







Captura de pantalla donde se muestren todos los recursos AWS generados tras el despliegue, y englobados en un stack/pila.

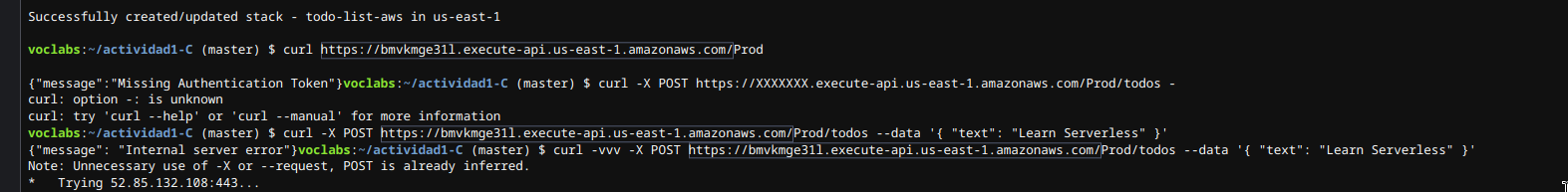
Reto 3 – Verificación API

En este reto se solicitan 4 entregables:

* Ejecución de los comandos *curl*, con la API recién desplegada.

Estos comandos deben mostrar un error.

Adjuntar salida de todos los comandos.



* Ejecución manual de la función lambda del listado de To-Do’s, para obtener el detalle del error producido en la invocación a esta función.

Adjuntar captura de pantalla donde se muestre el error completo

No se tienen ya que se corrigió sin utilizar la ejecución manual.

* Corrección del código fuente y explicación del motivo del fallo

Explicación breve del fallo y cómo se ha corregido

Se puede obtener en este enlace

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_4.md#error](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_4.md" \l "error)

* Ejecución de todos los comandos *curl*, que ahora sí deben mostrar datos de la lista de tareas. El orden a seguir para la ejecución de los comandos será:
  + Alta de un elemento To-Do
  + Alta de otro elemento To-Do
  + Lista de To-Do’s
  + Modificación del primer To-Do creado
  + Obtención del primer To-Do creado
  + Borrado del segundo To-Do
  + Listado de To-Do’s

Deben mostrarse los comandos y salidas correspondientes.

Todo esto puede hacerse mediante curl, o bien con Postman, a elección del alumno.

Se pueden observar en este enlace

[https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte\_C\_Reto\_4.md#reto-4----verificaci%C3%B3n-api](https://github.com/dargamenteria/actividad1-C/blob/master/docs/Parte_C_Reto_4.md" \l "reto-4----verificaci%C3%B3n-api)