

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/dargamentaria/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](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#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#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#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/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md#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 obtener en esta url

https://github.com/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md#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/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_2.md#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.

```
Please make sure you have the correct access rights
and the repository exists.
voclabs:~ $ git clone https://github.com/dargamentaria/actividad1-C.git
Cloning into 'actividad1-C'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 79 (delta 13), reused 79 (delta 13), pack-reused 0
Receiving objects: 100% (79/79), 29.99 KiB | 5.00 MiB/s, done.
Resolving deltas: 100% (13/13), done.
voclabs:~ $ cd actividad1-C/
voclabs:~/actividad1-C (master) $ ls -arlt
```

- ▮ Construcción de la plantilla SAM

Comandos empleados junto con sus salidas

Se pueden encontrar en esta url

https://github.com/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_3.md#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”.

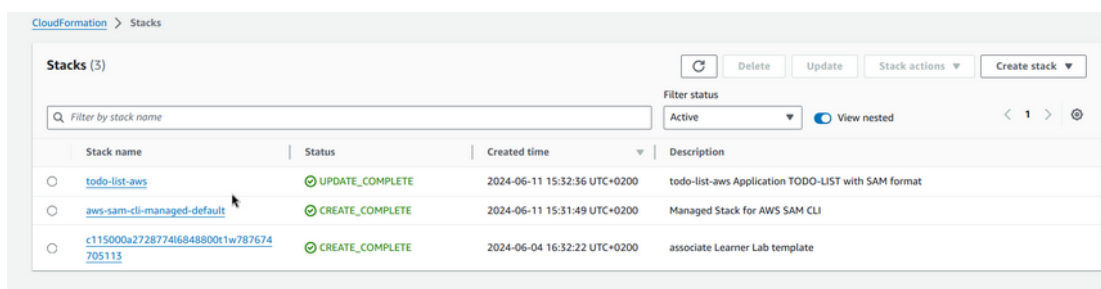
Se pueden encontrar en esta url

[https://github.com/dargamentaria/actividad1-C/blob/master/docs/](https://github.com/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_3.md#reto-3---stack-serverless)

Parte_C_Reto_3.md#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.



The screenshot shows the AWS CloudFormation console with a list of stacks. The 'todo-list-aws' stack is selected, and its details are visible. The stack is in the 'UPDATE_COMPLETE' state, indicating it has been successfully updated.

Stack name	Status	Created time	Description
todo-list-aws	UPDATE_COMPLETE	2024-06-11 15:32:36 UTC+0200	todo-list-aws Application TODO-LIST with SAM format
aws-sam-cli-managed-default	CREATE_COMPLETE	2024-06-11 15:31:49 UTC+0200	Managed Stack for AWS SAM CLI
c115000a2728774i6848800t1w787674705113	CREATE_COMPLETE	2024-06-04 16:32:22 UTC+0200	associate Learner Lab template

Stack info	Events	Resources	Outputs	Parameters	Template	Change sets	Git sync - new
Overview 🔄							
Stack ID arn:aws:cloudformation:us-east-1:787674705113:stack/todo-list-aws/10fb1330-27f7-11ef-9acf-0affe899ceb1				Description todo-list-aws Application TODO-LIST with SAM format			
Status 🟢 UPDATE_COMPLETE				Detailed status -			
Status reason -				Root stack -			
Parent stack -				Created time 2024-06-11 15:32:36 UTC+0200			
				Updated time 2024-06-11 15:38:16 UTC+0200			
Deleted time -				Drift status 🕒 NOT_CHECKED			
Last drift check time -				Termination protection Deactivated			
IAM role -							

Stack info	Events	Resources	Outputs	Parameters	Template	Change sets	Git sync - new
------------	--------	-----------	---------	------------	----------	-------------	----------------

Resources (14)

Search resources

Logical ID	Physical ID	Type	Status
CreateTodoFunction	todo-list-aws-CreateTodoFunction-tevpLaw6lIxE	AWS::Lambda::Function	UPDATE_COMPLETE
CreateTodoFunctionCreatePermissionProd	todo-list-aws-CreateTodoFunctionCreatePermissionProd-KoeBu5t4Zn8c	AWS::Lambda::Permission	CREATE_COMPLETE
DeleteTodoFunction	todo-list-aws-DeleteTodoFunction-XowcpVMxx5tD	AWS::Lambda::Function	UPDATE_COMPLETE
DeleteTodoFunctionCreatePermissionProd	todo-list-aws-DeleteTodoFunctionCreatePermissionProd-MOp8mdtpnULF	AWS::Lambda::Permission	CREATE_COMPLETE
GetTodoFunction	todo-list-aws-GetTodoFunction-w8NQ8fapt51x	AWS::Lambda::Function	UPDATE_COMPLETE
GetTodoFunctionCreatePermissionProd	todo-list-aws-GetTodoFunctionCreatePermissionProd-GaliHU0hb11F	AWS::Lambda::Permission	CREATE_COMPLETE
	todo-list-aws-		UPDATE_COMPLETE

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.

```
Successfully created/updated stack - todo-list-aws in us-east-1
voclabs~/actividad1-C (master) $ curl https://bmvmge3ll.execute-api.us-east-1.amazonaws.com/Prod
{"message": "Missing Authentication Token"}voclabs~/actividad1-C (master) $ curl -X POST https://xxxxxxxx.execute-api.us-east-1.amazonaws.com/Prod/todos -
curl: option -: is unknown
curl: try 'curl --help' or 'curl --manual' for more information
voclabs~/actividad1-C (master) $ curl -X POST https://bmvmge3ll.execute-api.us-east-1.amazonaws.com/Prod/todos --data '{ "text": "Learn Serverless" }'
{"message": "Internal server error"}voclabs~/actividad1-C (master) $ curl -vvv -X POST https://bmvmge3ll.execute-api.us-east-1.amazonaws.com/Prod/todos --data '{ "text": "Learn Serverless" }'
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 52.85.132.185:443...
```

- ▮ 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/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_4.md#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/dargamentaria/actividad1-C/blob/master/docs/
Parte_C_Reto_4.md#reto-4---verificaci%C3%B3n-api](https://github.com/dargamentaria/actividad1-C/blob/master/docs/Parte_C_Reto_4.md#reto-4---verificaci%C3%B3n-api)