

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Propuesta de solución

Caso práctico 1

URL de repositorio solución de GitHub: <https://github.com/WillyVilloria/todo-list-aws.git>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Caso práctico 1. Apartado A

Una vez se ha determinado la correcta adecuación del aplicativo base dentro del ecosistema de servicios que provee Amazon Web Services a través del uso de [Serverless Framework](#), es tiempo de dotar al caso práctico de una aproximación Full-AWS y Jenkins. ¿Qué significa? En línea con lo aprendido durante el programa, AWS dispone de una suite de servicios orientados a como disponibilizar aplicaciones desde cero a través de Serverless Application Model (**SAM**). Su uso posibilita la construcción de pipelines de integración y entrega continua para automatizar los procesos de compilación de los artefactos software requeridos en el despliegue en el entorno productivo.

A continuación, se van a detallar brevemente cada uno de los servicios necesarios para la elaboración de este apartado:

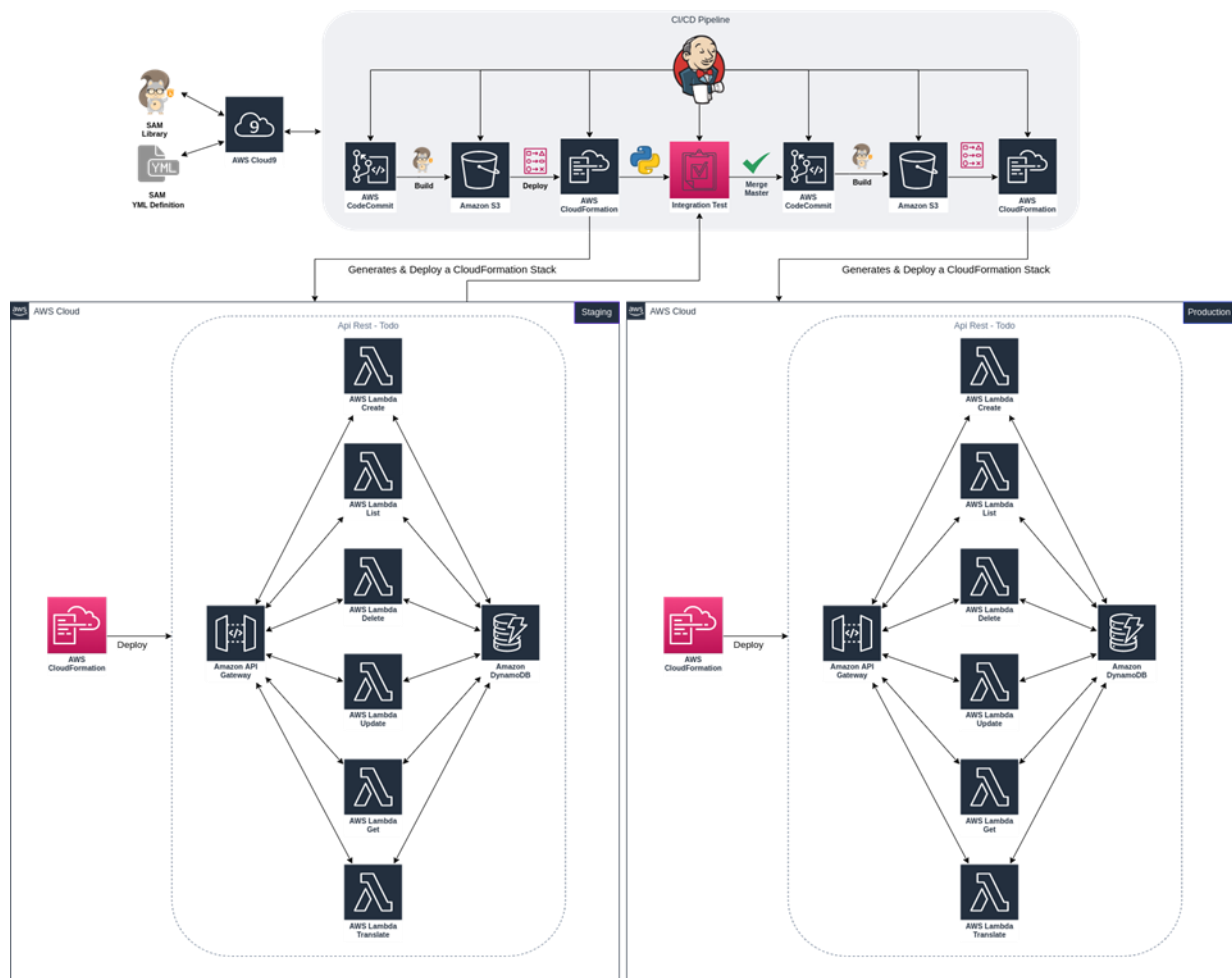
- ▶ [AWS CloudFormation](#): Servicio de diseño, implementación y despliegue automático de infraestructura de aplicaciones Cloud, simplificando su diseño a través de un lenguaje común de modelado.
- ▶ [AWS CodeCommit](#): Servicio autogestionado de control de código fuente, a través de un control del histórico de versiones en repositorios independientes.
- ▶ [Elastic Computer Service \(EC2\)](#): Servicio de computación en la nube de AWS, según la cual el usuario es capaz de disponer al instante, y sin inversión previa en infraestructura hardware física propia, de capacidad informática acorde a las necesidades de las aplicaciones o soluciones digitales. El modelo de negocio y explotación se rige según la demanda a cada instante de la empresa u organización en el uso de dicho servicio en cuestión, optimizando ostensiblemente los costes asociados por ello.
- ▶ [Simple Storage Service \(S3\)](#): Servicio de almacenamiento de objetos con sistema de versionado ante modificaciones en los mismos, alto rendimiento y finalidad multipropósito (Lago de datos y tracking IoT, sitios web, aplicaciones mobile, recurso de backup y archivado, entre otros).

Además de los servicios de AWS, se va a hacer uso de la herramienta de [Jenkins](#), desplegada dentro de una instancia EC2 para construir ahí los diferentes *pipelines* que se van a proponer en este apartado.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Resumen de la solución a implementar. Apartado A

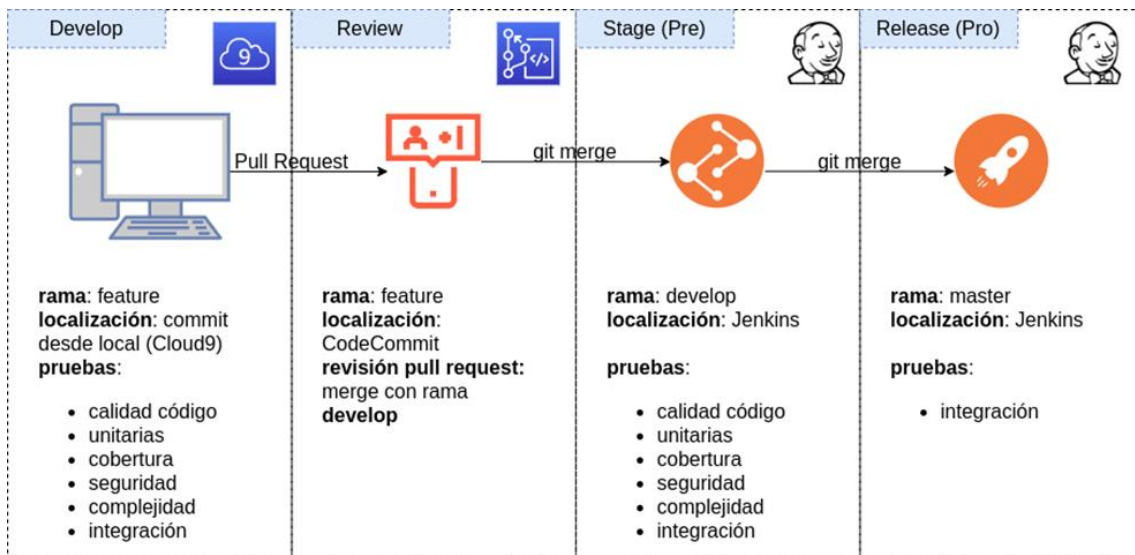
El objetivo global es el de adquirir un conocimiento extenso en el uso de distintos *frameworks* de diseño e implementación de *pipelines* de CI/CD, siendo esta aproximación la correspondiente al ámbito de trabajo del propio proveedor de soluciones en la nube Amazon Web Services, utilizando el servicio más común en los entornos productivos, como es Jenkins y el marco de despliegue de arquitectura de aplicaciones Software Serverless AWS SAM. La visión global que el alumno debe de lograr alcanzar de la solución en esta siguiente ocasión ha de ser similar al siguiente:



Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Para ello se propone un *Pipeline* de CI/CD basado en 2 entornos físicos:

- ▶ Local: donde se desarrollarán y probarán las nuevas features desplegadas de manera local. Requerirá de usar una rama nueva de git denominada **feature**.
- ▶ CI/CD: con dos escenarios de preproducción (*staging*) y producción (*production*), donde se construirá el *Pipeline* de CI/CD. En cada escenario se usará la rama adecuada para cada entorno, siendo **develop** para el entorno de preproducción y **master** para el entorno de producción. En este segundo ejercicio el alumno tiene que implementar las distintas pruebas de sobre el código de manera obligatoria.



De cara a la elaboración de apartado B, deberán afrontarse las siguientes fases o etapas desde la cuenta asignada a cada alumno en AWS Academy, de las que se entrará en mayor detalle seguidamente:

1. Clonado repositorio de la práctica y copia en repositorio de alumno

Comando	Resultado
\$ git remote -v	https://github.com/WillyVilloria/todo-list-aws.git

2. Validación SAM CLI (Command Line Interface) y análisis de repositorio

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

2.1. Ejecución del comando SAM funcionando:

Comando	Resultado
\$ sam --version	SAM CLI, version 1.72.0

2.2. Análisis del repositorio: tras llevar a cabo la migración del repositorio entre sistemas de control de versiones de código, realizar la siguiente comprobación de la url de nueva creación de la ubicación destino:

Ficheros	Contenido
src	<p>todolist.py → archivo con varias funciones Python que manipulan datos de la tabla dynamodb</p> <p>create.py → archivo con una función Python para crear un ítem de la tabla de una bd</p> <p>decimalencoder.py → archivo con una clase Python que contiene una función que devuelve un numero entero</p> <p>delete.py → archivo con una función Python que llama a la función delete_item del archivo todoList</p> <p>get.py → archivo con una función Python para que llama a la función get_item de todoList.py</p> <p>list.py → archivo con una función Python que llama a la función get_item del archivo todoList.py</p> <p>update.py → archivo con una función Python para actualizar que fundamentalmente llama a la función update_item del archivo todoList.py</p>
test	<p>todoApiTest.py → archivo con los test de integración. Contiene una clase con varios métodos que simulan y prueban el funcionamiento de las funciones lambda de la api.</p> <p>TestToDo.py → archivo con los test unitarios. Contiene una clase con varios métodos que generan llamadas a las funciones del archivo todoList.py probando el comportamiento de</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	éstos.
pipelines	<p>Pipeline-full.staging:</p> <p>Archivo Jenkinsfile que contiene cinco stages:</p> <ul style="list-style-type: none"> - Setup: hace una llamada al archivo setup.sh que instala los recursos necesarios para la realización de las pruebas - Test: hace llamada a los archivos static_test.sh y unit_test.sh que utiliza los programas randon, flake8, bandit y coverage para testear la aplicación. - Build: Abre el archivo pipelines/common-steps/build.sh - Deploy: Abre el archivo pipelines/common-steps/deploy.sh - Integracion test after deploy: Abre el archivo pipelines/common-steps/integration.sh - Limpia directorios
	<p>Pipeline full production:</p> <p>Archivo Jenkinsfile que contiene 4 stages:</p> <ul style="list-style-type: none"> - Setup: Abre el archivo pipelines/PIPELINE-FULL-PRODUCTION/setup.sh - Build: Abre el archivo pipelines/common-steps/build.sh - Deploy: Abre el archivo pipelines/common-steps/deploy.sh - Integracion test after deploy: Abre el archivo pipelines/common-steps/integration.sh - Limpia directorios
	<p>Pipeline full cd:</p> <p>Archivo Jenkinsfile que contiene 3 stages:</p> <ul style="list-style-type: none"> - Staging: Abre parametros del pipeline full ataging - Merge: realiza un merge de la rama develop a la rama master - Production: Abre parámetros del pipeline full production.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	- Limpia directorios
template.yaml	Plantilla de definición de aplicación sin servidor. Se utiliza para crear las funciones lambda y la tabla de la dynamodb.
samconfig.toml	Archivo de configuración de sam donde, dependiendo del parámetro (default, staging o production), asigna nombre a las pilas de cloudformation, asigna el s3_bucket etc. Este archivo contiene la información que solicita el sam deploy.
localEnvironment.json	Contiene, en formato json, los datos de los endpoints y la dynamodb de las funciones lambda para trabajar con el despliegue en local.

3. Ejecución de proyecto en entorno local (SAM CLI):

3.1. Pasos para levantar el entorno local:

Resultados a mostrar	Salida
Crear red de docker	cac462063698f599e0dd2b87745cc66aec66e64fa534f222e6dbbe431085704
Levantar contenedor de docker	75a46b2c18bdfeaa44335244b528c195bc96e1aa90bfff0ec21abbb9a38ca489
Crear tabla en dynamodb local	{ "TableDescription": { "AttributeDefinitions": [{ "AttributeName": "id", "AttributeType": "S" }], "TableName": "local-TodosDynamoDbTable", "KeySchema": [{ "AttributeName": "id", "KeyType": "HASH" }], "TableStatus": "ACTIVE", "CreationDateTime": 1684657872.38,

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	<pre> "ProvisionedThroughput": { "LastIncreaseDateTime": 0.0, "LastDecreaseDateTime": 0.0, "NumberOfDecreasesToday": 0, "ReadCapacityUnits": 1, "WriteCapacityUnits": 1 }, "TableSizeBytes": 0, "ItemCount": 0, "TableArn": "arn:aws:dynamodb:ddblocal:000000000000:table/lo- cal-TodosDynamoDbTable" } } </pre>
Empaquetar proyecto con SAM	<p>Building codeuri: /home/ubuntu/todo-list-aws/src runtime: python3.7 metadata: {} architecture: x86_64 functions: CreateTodoFunction, ListTodosFunction, GetTodoFunction, UpdateTodoFunction, DeleteTodoFunction Running PythonPipBuilder:ResolveDependencies Running PythonPipBuilder:CopySource</p> <p>Build Succeeded</p> <p>Built Artifacts : .aws-sam/build Built Template : .aws-sam/build/template.yaml</p> <p>Commands you can use next =====</p> <p>[*] Validate SAM template: sam validate [*] Invoke Function: sam local invoke [*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch [*] Deploy: sam deploy --guided s3_bucket = "aws-sam-cli-managed-default-samclisourcebucket-vhgnor3pt2vu"</p>
Levantar la API localmente	<p>Mounting CreateTodoFunction at http://127.0.0.1:8081/todos [POST] Mounting DeleteTodoFunction at http://127.0.0.1:8081/todos/{id} [DELETE] Mounting ListTodosFunction at http://127.0.0.1:8081/todos [GET] Mounting GetTodoFunction at http://127.0.0.1:8081/todos/{id} [GET]</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

```

Mounting UpdateTodoFunction at
http://127.0.0.1:8081/todos/{id} [PUT]
You can now browse to the above endpoints to
invoke your functions. You do not need to
restart/reload SAM CLI while working on your
functions, changes will be reflected
instantly/automatically. If you used sam build before
running local commands, you will need to re-run sam
build for the changes to be picked up. You only need
to restart SAM CLI if you update your AWS SAM
template
2023-05-21 08:44:23 * Running on
http://127.0.0.1:8081/ (Press CTRL+C to quit)

```

Se adjunta el listado de peticiones a los métodos del servicio desplegados localmente con la url o *endpoint* relativa y un campo «resultado» al objeto de completitud por parte del alumno, así como los distintos pasos para conseguir desplegar **SAM** local, como las evidencias de que se ha realizado correctamente:

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

3.2. En entorno local:

Función	In/out	Script
Create	comando	curl -X POST http://127.0.0.1:8081/todos --data '{ "text": "Learn Serverless" }'
	resultado	{ "statusCode": 200, "body": "{ \"id\": \"259947aa-f81c-11ed-8c4a-0242ac110002\", \"text\": \"Learn Serverless\", \"checked\": false, \"createdAt\": \"1684703529.1578696\", \"updatedAt\": \"1684703529.1578696\"}" }
List	comando	curl http://127.0.0.1:8081/todos
	resultado	[{"createdAt": "1684703529.1578696", "checked": true, "id": "259947aa-f81c-11ed-8c4a-0242ac110002", "text": "Learn Serverless", "updatedAt": "1684704049603"}]
Get	comando	curl http://127.0.0.1:8081/todos/<id>
	resultado	{ "createdAt": "1684703529.1578696", "checked": true, "id": "259947aa-f81c-11ed-8c4a-0242ac110002", "text": "Learn Serverless", "updatedAt": "1684704049603" }
Update	comando	curl -X PUT http://127.0.0.1:8081/todos/<id> --data '{ "text": "Learn Serverless", "checked": true }'
	resultado	{ "createdAt": "1684703529.1578696", "checked": true, "id": "259947aa-f81c-11ed-8c4a-0242ac110002", "text": "Learn Serverless", "updatedAt": "1684704049603" }
Delete	comando	curl -X DELETE http://127.0.0.1:8081/todos/<id>
	resultado	Mounting /home/ubuntu/todo-list-aws/.aws-sam/build/DeleteTodoFunction as /var/task:ro,delegated inside runtime container START RequestId: 084f8feb-30ee-46ba-a8f9-faa220adcdd7 Version: \$LATEST URL dynamoDB:http://dynamodb:8000 END RequestId: 084f8feb-30ee-46ba-a8f9-faa220adcdd7 REPORT RequestId: 084f8feb-30ee-46ba-a8f9-faa220adcdd7 Init Duration: 0.36 ms Duration: 795.54 ms Billed Duration: 796 ms Memory Size: 128 MB Max Memory Used: 128 MB Lambda returned empty body! No Content-Type given. Defaulting to 'application/json'. 2023-05-21 21:23:04 127.0.0.1 - - [21/May/2023 21:23:04] "DELETE /todos/259947aa-f81c-11ed-8c4a-0242ac110002 HTTP/1.1" 200 -

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

4. Despliegue manual de aplicación SAM en Amazon Web Services:

Observaciones: Se valorará positivamente la generación de un reporte detallado del proceso de obtención de los distintos hitos: capturas de pantalla acreditando la consecución de los objetivos, diagramas explicativos, etc.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Completa nuevamente el siguiente cuadro con los *endpoints* de cada función:

Función	Endpoint
Create	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos
List	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos
Get	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/<id>
Update	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/<id>
Delete	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/<id>

Adjuntar a continuación los resultados de las validaciones, así como los logs de los distintos comandos ejecutados:

todo

4.1. Invocaciones a los métodos del API

Función	In/out	Script
Create	comando	curl -X POST https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos --data '{ "text": "Learn Serverless" }'
	resultado	{"statusCode": 200, "body": "{\"id\": \"e162b164-f881-11ed-ba4e-8e52a6b81363\", \"text\": \"Learn Serverless\", \"checked\": false, \"createdAt\": \"1684747223.2770526\", \"updatedAt\": \"1684747223.2770526\"}"}
List	comando	curl https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos
	resultado	[{"checked": false, "createdAt": "1684747223.2770526", "text": "Learn Serverless", "id": "e162b164-f881-11ed-ba4e-8e52a6b81363", "updatedAt": "1684747223.2770526"}]
Get	comando	curl https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/1684747223.2770526
	resultado	{"checked": true, "text": " Learn Serverless ", "id": "1684747223.2770526", "updatedAt": 1684747644732}
Update	comando	curl -X PUT https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/1684747223.2770526 --data '{ "text": " Learn Serverless ", "checked": true }'
	resultado	{"checked": true, "text": " Learn Serverless ", "id": "1684747223.2770526", "updatedAt":

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

		1684747644732}
Delete	comando	curl -X DELETE https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/1684747223.2770526
	resultado	

4.2. Logs de SAM

Resultados a mostrar	Salida
Log sam build	<p>Building codeuri: /home/ubuntu/todo-list-aws/src runtime: python3.7 metadata: {} architecture: x86_64 functions: CreateTodoFunction, ListTodosFunction, GetTodoFunction, UpdateTodoFunction, DeleteTodoFunction</p> <p>Running PythonPipBuilder:ResolveDependencies</p> <p>Running PythonPipBuilder:CopySource</p> <p>Build Succeeded</p> <p>Built Artifacts : .aws-sam/build</p> <p>Built Template : .aws-sam/build/template.yaml</p> <p>Commands you can use next</p> <p>=====</p> <p>[*] Validate SAM template: sam validate</p> <p>[*] Invoke Function: sam local invoke</p> <p>[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch</p> <p>[*] Deploy: sam deploy --guided</p>
Log sam deploy	<p>voclabs:~/todo-list-aws (feature) \$ sam deploy template.yaml --config-env default</p> <p>Uploading to todo-list-aws/51a2dbce9d6fe1876a575453320b75e5 11724436 / 11724436 (100.00%)</p> <p>File with same data already exists at todo-list-aws/51a2dbce9d6fe1876a575453320b75e5, skipping upload</p> <p>File with same data already exists at todo-list-aws/51a2dbce9d6fe1876a575453320b75e5, skipping upload</p> <p>File with same data already exists at todo-list-aws/51a2dbce9d6fe1876a575453320b75e5, skipping upload</p> <p>File with same data already exists at todo-list-aws/51a2dbce9d6fe1876a575453320b75e5, skipping upload</p> <p>Deploying with following values</p> <p>=====</p> <p>Stack name : todo-list-aws</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

<p>Region : us-east-1</p> <p>Confirm changeset : True</p> <p>Disable rollback : False</p> <p>Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-vhgnor3pt2vu</p> <p>Capabilities : ["CAPABILITY_IAM"]</p> <p>Parameter overrides : {"Stage": "default"}</p> <p>Signing Profiles : {}</p> <p>Initiating deployment</p> <p>=====</p> <p>Uploading to todo-list-aws/07b6b19b80f8e793cd6b9713c70a6b2a.template 4630 / 4630 (100.00%)</p> <p>Waiting for changeset to be created..</p> <p>CloudFormation stack changeset</p>	
--	--

CloudFormation events from stack operations (refresh every 0.5 seconds)			
ResourceStatus	ResourceType	LogicalResourceId	ResourceStatusReason
CREATE_IN_PROGRESS	AWS::DynamoDB::Table	TodosDynamoDbTable	-
CREATE_IN_PROGRESS	AWS::DynamoDB::Table	TodosDynamoDbTable	Resource creation Initiated
CREATE_COMPLETE	AWS::DynamoDB::Table	TodosDynamoDbTable	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	UpdateTodoFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	GetTodoFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	DeleteTodoFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	CreateTodoFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	ListTodosFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	GetTodoFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Function	UpdateTodoFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Function	DeleteTodoFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Function	CreateTodoFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Function	ListTodosFunction	Resource creation Initiated
CREATE_COMPLETE	AWS::Lambda::Function	GetTodoFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	DeleteTodoFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	UpdateTodoFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	CreateTodoFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	ListTodosFunction	-
CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	ServerlessRestApi	-
CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	ServerlessRestApi	Resource creation Initiated
CREATE_COMPLETE	AWS::ApiGateway::RestApi	ServerlessRestApi	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	DeleteTodoFunctionCreatePermissionProd	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	ListTodosFunctionCreatePermissionProd	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	CreateTodoFunctionCreatePermissionProd	-
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment141b842de6	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	UpdateTodoFunctionCreatePermissionProd	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	GetTodoFunctionCreatePermissionProd	-
CREATE_IN_PROGRESS	AWS::Lambda::Permission	DeleteTodoFunctionCreatePermissionProd	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	CreateTodoFunctionCreatePermissionProd	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	ListTodosFunctionCreatePermissionProd	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	UpdateTodoFunctionCreatePermissionProd	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	GetTodoFunctionCreatePermissionProd	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment141b842de6	Resource creation Initiated
CREATE_COMPLETE	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment141b842de6	-
CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	-
CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Resource creation Initiated
CREATE_COMPLETE	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	-
CREATE_COMPLETE	AWS::Lambda::Permission	DeleteTodoFunctionCreatePermissionProd	-
CREATE_COMPLETE	AWS::Lambda::Permission	CreateTodoFunctionCreatePermissionProd	-

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

```
bash - "ip-172-31-94-18" x Immediate (Javascript (bn x) +
-----
CREATE_IN_PROGRESS AWS::Lambda::Permission ListTodosFunctionCreatePermissionProd Resource creation Initiated
CREATE_IN_PROGRESS AWS::Lambda::Permission UpdateTodoFunctionCreatePermissionProd Resource creation Initiated
CREATE_IN_PROGRESS AWS::Lambda::Permission GetTodoFunctionCreatePermissionProd Resource creation Initiated
CREATE_IN_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment141b842de6 Resource creation Initiated
CREATE_COMPLETE AWS::ApiGateway::Stage ServerlessRestApiDeployment141b842de6 -
CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage Resource creation Initiated
CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage -
CREATE_COMPLETE AWS::Lambda::Permission DeleteTodoFunctionCreatePermissionProd -
CREATE_COMPLETE AWS::Lambda::Permission CreateTodoFunctionCreatePermissionProd -
CREATE_COMPLETE AWS::Lambda::Permission ListTodosFunctionCreatePermissionProd -
CREATE_COMPLETE AWS::Lambda::Permission UpdateTodoFunctionCreatePermissionProd -
CREATE_COMPLETE AWS::Lambda::Permission GetTodoFunctionCreatePermissionProd -
CREATE_COMPLETE AWS::CloudFormation::Stack todo-list-aws -
-----
CloudFormation outputs from deployed stack
-----
Outputs
-----
Key BaseUrlApi
Description Base URL of API
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod

Key DeleteTodoApi
Description API Gateway endpoint URL for ${(opt:stage)} stage for Delete TODO
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}

Key ListTodosApi
Description API Gateway endpoint URL for ${(opt:stage)} stage for List TODO
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos

Key UpdateTodoApi
Description API Gateway endpoint URL for ${(opt:stage)} stage for Update TODO
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}

Key GetTodoApi
Description API Gateway endpoint URL for ${(opt:stage)} stage for Get TODO
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}

Key CreateTodoApi
Description API Gateway endpoint URL for ${(opt:stage)} stage for Create TODO
Value https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/
```

Registros función todo-list-aws-GetTodoFunction-ebylKNWkpon

t

CloudWatch Logs Información

Lambda registra todas las solicitudes gestionadas por la función y almacena automáticamente los registros generados por el código a través de Amazon CloudWatch Logs. Para validar el código, debe instrumentarlo con instrucciones de registro personalizadas. En las tablas siguientes se muestran las invocaciones de funciones más recientes y más caras de toda la actividad de las funciones. Para ver los registros correspondientes a un alias o a una versión específicos de una función, visite la sección de **Monitoreo** en ese nivel.

1h 3h 12h 1d 3d 1sem. Personalizado

#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
▶ 1	2023-05-22T09:58:03.728Z	67bc7c95-9d8c-40de-a7f4-d2ff907d2fca	2023/05/22/[LATEST]24f8dfd953994ca298fff68de1e726a1	1.74	2.0	128	53
▶ 2	2023-05-22T09:56:48.526Z	1b1e1eda-e824-43e2-92eb-1725b30cf8e4	2023/05/22/[LATEST]24f8dfd953994ca298fff68de1e726a1	17.15	16.0	128	52
▶ 3	2023-05-22T09:28:126.483Z	9f34a0a7-5033-4a30-a855-4567981363db	2023/05/22/[LATEST]c84da32aa37a4e48b6ef380b05ff3f7b	2039.58	2040.0	128	65

Most expensive invocations in GB-seconds (memory assigned * billed duration)

#	Timestamp	RequestID	LogStream	BilledDurationInMS	MemorySetInMB	BilledDurationInGBSeconds
▶ 1	2023-05-22T09:28:126.483Z	9f34a0a7-5033-4a30-a855-4567981363db	2023/05/22/[LATEST]c84da32aa37a4e48b6ef380b05ff3f7b	2040.0	128	0.255
▶ 2	2023-05-22T09:56:48.526Z	1b1e1eda-e824-43e2-92eb-1725b30cf8e4	2023/05/22/[LATEST]24f8dfd953994ca298fff68de1e726a1	16.0	128	0.00225
▶ 3	2023-05-22T09:58:03.728Z	67bc7c95-9d8c-40de-a7f4-d2ff907d2fca	2023/05/22/[LATEST]24f8dfd953994ca298fff68de1e726a1	2.0	128	0.00025

todo-list-aws-UpdateTodoFunction-JqrnlWodJosh

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

CloudWatch Logs [Información](#)

Lambda registra todas las solicitudes gestionadas por la función y almacena automáticamente los registros generados por el código a través de Amazon CloudWatch Logs. Para validar el código, debe instrumentarlo con instrucciones de registro personalizadas. En las tablas siguientes se muestran las invocaciones de funciones más recientes y más caras de toda la actividad de las funciones. Para ver los registros correspondientes a un alias o una versión específicos de una función, visite la sección de **Monitoreo** en ese nivel.

1h3h12h1d3d1semPersonalizado

Recent invocations

#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
1	2023-05-22T09:58:12.706Z	f7a3d9dd-63f6-4068-b7d7-0178e522056d	2023/05/22/[\$LATEST]b48dff1fa2dc4486bd20966cb4fd2eb	12.69	13.0	128	52
2	2023-05-22T09:51:25.792Z	8f294606-8dcd-4d74-8d6e-f31f2cf79953	2023/05/22/[\$LATEST]e2cf11096dcf40f380a790f889d249ff	301.08	302.0	128	66
3	2023-05-22T09:47:11.430Z	3f64a4c2-0205-409f-bccf-e1124e7e0220	2023/05/22/[\$LATEST]e2cf11096dcf40f380a790f889d249ff	2056.8	2057.0	128	65
4	2023-05-22T09:37:49.950Z	b76dc1a6-86f6-4a44-a00c-b66cf9f6fb94	2023/05/22/[\$LATEST]e5130734a75c474b823fa2d1c4a20fc0	2.37	3.0	128	52
5	2023-05-22T09:37:49.924Z	9b8ec99a-dfa4-43c8-9d6a-02e937e9cbe4	2023/05/22/[\$LATEST]e5130734a75c474b823fa2d1c4a20fc0	4.94	5.0	128	52
6	2023-05-22T09:27:25.053Z	d31e48cd-633e-477b-bb5b-e191b63e7493	2023/05/22/[\$LATEST]c2eb52515cfff44ffa39b3d7d589a7e32	2079.57	2080.0	128	65

Most expensive invocations in GB-seconds (memory assigned * billed duration)

#	Timestamp	RequestID	LogStream	BilledDurationInMS	MemorySetInMB	BilledDurationInGBSeconds
1	2023-05-22T09:27:25.053Z	d31e48cd-633e-477b-bb5b-e191b63e7493	2023/05/22/[\$LATEST]c2eb52515cfff44ffa39b3d7d589a7e32	2080.0	128	0.26
2	2023-05-22T09:47:11.430Z	3f64a4c2-0205-409f-bccf-e1124e7e0220	2023/05/22/[\$LATEST]e2cf11096dcf40f380a790f889d249ff	2057.0	128	0.2571
3	2023-05-22T09:51:25.792Z	8f294606-8dcd-4d74-8d6e-f31f2cf79953	2023/05/22/[\$LATEST]e2cf11096dcf40f380a790f889d249ff	302.0	128	0.03775
4	2023-05-22T09:58:12.706Z	f7a3d9dd-63f6-4068-b7d7-0178e522056d	2023/05/22/[\$LATEST]b48dff1fa2dc4486bd20966cb4fd2eb	13.0	128	0.001625
5	2023-05-22T09:37:49.924Z	9b8ec99a-dfa4-43c8-9d6a-02e937e9cbe4	2023/05/22/[\$LATEST]e5130734a75c474b823fa2d1c4a20fc0	5.0	128	0.000625
6	2023-05-22T09:37:49.950Z	b76dc1a6-86f6-4a44-a00c-b66cf9f6fb94	2023/05/22/[\$LATEST]e5130734a75c474b823fa2d1c4a20fc0	3.0	128	0.000375

todo-list-aws-ListTodosFunction-WFzRM1T586CP

1h3h12h1d3d1semPersonalizado

Recent invocations

#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
1	2023-05-22T09:59:51.493Z	78cfd9ac-0a5a-4577-ac85-8ba31fd202c8	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	272.64	273.0	128	67
2	2023-05-22T09:59:28.293Z	1e95af4a-8925-45ab-bd5c-ac61327357a7	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	312.4	313.0	128	67
3	2023-05-22T09:57:35.193Z	527544f4-928a-4cab-ab3f-5362e8a9fa0c	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	281.75	282.0	128	67
4	2023-05-22T09:54:34.535Z	3125fe9f-4d22-4fa6-a6e2-66e73db3b04e	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	272.22	273.0	128	66
5	2023-05-22T09:53:14.792Z	2bef7601-20fb-44e8-a589-4e28e797d238	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	278.23	279.0	128	66
6	2023-05-22T09:49:51.092Z	05f862eb-54fd-472e-9a83-3cf38558fb13	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	2026.98	2027.0	128	65
7	2023-05-22T09:21:22.976Z	cd326b33-631f-41f6-b55c-6c89948257e8	2023/05/22/[\$LATEST]bc369515aaef42149363687308d33d1f	296.37	299.0	128	66
8	2023-05-22T09:18:20.836Z	3e3e7c02-897d-4fd8-b514-51f7eb7c4c8f	2023/05/22/[\$LATEST]bc369515aaef42149363687308d33d1f	2013.32	2014.0	128	65

Most expensive invocations in GB-seconds (memory assigned * billed duration)





#	Timestamp	RequestID	LogStream	BilledDurationInMS	MemorySetInMB	BilledDurationInGBSeconds
1	2023-05-22T09:49:51.092Z	05f862eb-54fd-472e-9a83-3cf38558fb13	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	2027.0	128	0.2534
2	2023-05-22T09:18:20.836Z	3e3e7c02-897d-4fd8-b514-51f7eb7c4c8f	2023/05/22/[\$LATEST]bc369515aaef42149363687308d33d1f	2014.0	128	0.2517
3	2023-05-22T09:59:28.293Z	1e95af4a-8925-45ab-bd5c-ac61327357a7	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	313.0	128	0.03913
4	2023-05-22T09:21:22.976Z	cd326b33-631f-41f6-b55c-6c89948257e8	2023/05/22/[\$LATEST]bc369515aaef42149363687308d33d1f	299.0	128	0.03737
5	2023-05-22T09:57:35.193Z	527544f4-928a-4cab-ab3f-5362e8a9fa0c	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	282.0	128	0.03525
6	2023-05-22T09:53:14.792Z	2bef7601-20fb-44e8-a589-4e28e797d238	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	279.0	128	0.03488
7	2023-05-22T09:59:51.493Z	78cfd9ac-0a5a-4577-ac85-8ba31fd202c8	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	273.0	128	0.03413
8	2023-05-22T09:54:34.535Z	3125fe9f-4d22-4fa6-a6e2-66e73db3b04e	2023/05/22/[\$LATEST]3446ef68b93b4e1fab3316d7c7590e5	273.0	128	0.03413


todo-list-aws-CreateTodoFunction-p8FH827zD4vb

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

CloudWatch Logs [Información](#)

Lambda registra todas las solicitudes gestionadas por la función y almacena automáticamente los registros generados por el código a través de Amazon CloudWatch Logs. Para validar el código, debe instrumentarlo con instrucciones de registro personalizadas. En las tablas siguientes se muestran las invocaciones de funciones más recientes y más caras de toda la actividad de las funciones. Para ver los registros correspondientes a un alias o una versión específicos de una función, visite la sección de **Monitoreo** en ese nivel.





1h 3h 12h 1d 3d 1sem. Personalizado  								
Recent invocations  								
#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB	
▶ 1	2023-05-22T09:57:45.881Z	526564fc-c23b-4376-8906-45d2b3c1e950	2023/05/22/[LATEST]a5c3f3339f5c48cd901534344c0e91ff	1.73	2.0	128	52	
▶ 2	2023-05-22T09:56:31.144Z	45835427-b432-499e-be42-9ee2720c2ce8	2023/05/22/[LATEST]a5c3f3339f5c48cd901534344c0e91ff	12.61	13.0	128	52	
▶ 3	2023-05-22T09:36:03.960Z	8104bb49-facf-4e67-8d7e-9b0535861472	2023/05/22/[LATEST]c7b4a29fb51f485b90dfbb6f0c103227	2139.19	2140.0	128	66	


Most expensive invocations in GB-seconds (memory assigned * billed duration) 							
#	Timestamp	RequestID	LogStream	BilledDurationInMS	MemorySetInMB	BilledDurationInGBSeconds	
▶ 1	2023-05-22T09:36:03.960Z	8104bb49-facf-4e67-8d7e-9b0535861472	2023/05/22/[LATEST]c7b4a29fb51f485b90dfbb6f0c103227	2140.0	128	0.2675	
▶ 2	2023-05-22T09:56:31.144Z	45835427-b432-499e-be42-9ee2720c2ce8	2023/05/22/[LATEST]a5c3f3339f5c48cd901534344c0e91ff	13.0	128	0.001625	
▶ 3	2023-05-22T09:57:45.881Z	526564fc-c23b-4376-8906-45d2b3c1e950	2023/05/22/[LATEST]a5c3f3339f5c48cd901534344c0e91ff	2.0	128	0.00025	

todo-list-aws-DeleteTodoFunction-JJ9gikPFzALq

CloudWatch Logs [Información](#)

Lambda registra todas las solicitudes gestionadas por la función y almacena automáticamente los registros generados por el código a través de Amazon CloudWatch Logs. Para validar el código, debe instrumentarlo con instrucciones de registro personalizadas. En las tablas siguientes se muestran las invocaciones de funciones más recientes y más caras de toda la actividad de las funciones. Para ver los registros correspondientes a un alias o una versión específicos de una función, visite la sección de **Monitoreo** en ese nivel.

1h 3h 12h 1d 3d 1sem. Personalizado  								
Recent invocations  								
#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB	
▶ 1	2023-05-22T09:57:56.595Z	e1578658-5ab4-4bae-abf6-629d839d1262	2023/05/22/[LATEST]d42f907e942e46199efe20d3a683d742	12.4	13.0	128	52	
▶ 2	2023-05-22T09:44:33.434Z	311e039e-47de-4f48-9418-edb553d32ee5	2023/05/22/[LATEST]deeddbc7513f41739487e06e1121bdbb	1967.71	1968.0	128	65	
▶ 3	2023-05-22T09:29:17.975Z	7a67a9b2-4759-488d-883a-f37db1d96b83	2023/05/22/[LATEST]5388116fa754420494d719c07f668b5d	1994.32	1995.0	128	65	

Most expensive invocations in GB-seconds (memory assigned * billed duration) 							
#	Timestamp	RequestID	LogStream	BilledDurationInMS	MemorySetInMB	BilledDurationInGBSeconds	
▶ 1	2023-05-22T09:29:17.975Z	7a67a9b2-4759-488d-883a-f37db1d96b83	2023/05/22/[LATEST]5388116fa754420494d719c07f668b5d	1995.0	128	0.2494	
▶ 2	2023-05-22T09:44:33.434Z	311e039e-47de-4f48-9418-edb553d32ee5	2023/05/22/[LATEST]deeddbc7513f41739487e06e1121bdbb	1968.0	128	0.246	
▶ 3	2023-05-22T09:57:56.595Z	e1578658-5ab4-4bae-abf6-629d839d1262	2023/05/22/[LATEST]d42f907e942e46199efe20d3a683d742	13.0	128	0.001625	

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Pilas (3)

Activo

Vista anidada

< 1 >

Pilas

todo-list-aws

2023-05-22 11:15:50 UTC+0200

CREATE_COMPLETE

aws-sam-cli-managed-default

2023-05-19 17:36:27 UTC+0200

CREATE_COMPLETE

c82674a175888414084386t1w045641124235

2023-05-10 16:28:05 UTC+0200

CREATE_COMPLETE

Salidas (6)

Clave	Valor	Descripción	N
BaseUrlApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod	Base URL of API	-
CreateTodoApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/	API Gateway endpoint URL for \${opt:stage} stage for Create TODO	-
DeleteTodoApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}	API Gateway endpoint URL for \${opt:stage} stage for Delete TODO	-
GetTodoApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}	API Gateway endpoint URL for \${opt:stage} stage for Get TODO	-
ListTodosApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos	API Gateway endpoint URL for \${opt:stage} stage for List TODO	-
UpdateTodoApi	https://e81nhgej4h.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}	API Gateway endpoint URL for \${opt:stage} stage for Update TODO	-

Registro de aplicación en proceso

CloudFormation

Pilas

StackSets

Exportaciones

Designer

Registro

Extensiones públicas

Extensiones activadas

Editor

Destacados

Comentarios

CloudFormation > Pilas

Pilas (3)

Activo

Vista anidada

< 1 >

Nombre de la pila	Estado	Hora de creación	Descripción
todo-list-aws	CREATE_COMPLETE	2023-05-22 11:15:50 UTC+0200	todo-list-aws Application TODO-LIST with SAM format
aws-sam-cli-managed-default	CREATE_COMPLETE	2023-05-19 17:36:27 UTC+0200	Managed Stack for AWS SAM CLI
c82674a175888414084386t1w045641124235	CREATE_COMPLETE	2023-05-10 16:28:05 UTC+0200	associate Learner Lab template

5. Creación de pipelines de Jenkins para despliegue de arquitectura completa

5.1. Pipeline de staging

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Al completar esta sección, el entorno de *staging* quedaría desplegado y con las pruebas de integración ejecutadas. En caso de fallo del pipeline, se deberá de analizar y corregir los fallos. Al estar en un entorno de *staging* no sería necesario hacer *rollback* de la arquitectura, con el fin de analizar los posibles fallos y corregirlos.

Resultados a mostrar

Salida

Log pipeline

[Log pipeline staging](#)

Captura de pantalla del pipeline

Status

Changes

Build with Parameters

Configurar

Borrar Pipeline

Full Stage View

Rename

Coverage Report

Pipeline Syntax

Historia de tareas

Tendencia

Filter builds...

#35

1 jun. 2023 14:06

#34

27 may. 2023 23:02

#33

27 may. 2023 22:47

#32

27 may. 2023 22:41

#25

25 may. 2023 21:51

#12

25 may. 2023 19:52

#9

25 may. 2023 11:28

#8

25 may. 2023 11:22

#7

25 may. 2023 9:45

#6

23 may. 2023 16:58

#5

23 may. 2023 16:17

#4

23 may. 2023 16:03

#3

23 may. 2023 15:43

Atom feed Para Todos

Atom feed para los errores

Pipeline PIPELINE-FULL-STAGING


Stage View

Average stage times:

Average full run time: ~ 1min 59s

	Declarative: Checkout SCM	SetUp	Test	Build	Deploy	Integration Test after deploy	Declarative: Post Actions
Jun01 1606	2s	11s	13s	13s	1min 0s	33s	169ms
May28 01:02	2s	11s	8s	9s	56s	19s	128ms
May28 00:47	2s	11s	10s	12s	57s	20s	199ms
May28 00:41	3s	11s	1s	61ms	58ms	62ms	102ms
May25 23:51	483ms	20s	6s	10s	34s	7s	105ms
May25 20:52	704ms	23s	11s	13s	30s	21s	226ms
May25 13:28	468ms	21s	1s	62ms	52ms	54ms	127ms
May25 13:00	505ms	21s	1s	88ms	73ms	58ms	129ms

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	
Comentarios adicionales	<p>Para llegar a los datos solicitados</p> <ul style="list-style-type: none">- he modificado el archivo Jenkinsfile para aumentar el porcentaje mínimo de cobertura al 80%- He modificado el archivo TestTodo.py para que las pruebas de error generasen excepciones- He modificado el archivo todoList.py para que las funciones con try incluyeran dentro de estos mas líneas para que capturasen las excepciones que he forzado en el archivo de test. <p>Debido a que inicialmente no me di cuenta de que podía generar los tests en la consola en el despliegue "default", he hecho varios commits del código en la rama develop del repositorio remoto para después probarlo en Jenkins. Evidentemente esto me ha supuesto lanzar varias veces el pipeline de stage para hacer las pruebas desde el repositorio de github en vez de hacer (después sí que lo he hecho así) sólo los cambios y test en el repositorio de cloud9 y después de terminar hacer un único commit y lanzar el pipeline.</p>

5.2. Pipeline de Production

Indicar las salidas solicitadas del *pipeline*, bien copiando los logs, o adjuntando en la entrega final los ficheros de logs aparte.

Resultados a mostrar	Salida
----------------------	--------

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Log pipeline	Log pipeline production																																																								
Captura de pantalla del pipeline	<div><div><div>Status</div><div></> Changes</div><div>▷ Build with Parameters</div><div>⚙️ Configurar</div><div>🗑️ Borrar Pipeline</div><div>🔍 Full Stage View</div><div>✎️ Rename</div><div>🔗 Pipeline Syntax</div></div><div><div>☀️ Historia de tareas</div><div>Tendencia ▾</div><div>🔍 Filter builds... /</div><div><div>▷ #6 1 jun. 2023 14:08</div><div>▷ #5 1 jun. 2023 13:59</div><div>▷ #4 1 jun. 2023 13:50</div><div>▷ #3 23 may. 2023 17:00</div><div>▷ #2 23 may. 2023 16:54</div><div>▷ #1 23 may. 2023 16:47</div><div>📡 Atom feed Para Todos</div><div>📡 Atom feed para los errores</div></div></div><div><div><div>Pipeline PIPELINE-FULL-PRODUCTION</div><div>Stage View</div><div><div>Average stage times: (Average full run time: ~3min 50s)</div><table><thead><tr><th></th><th>Declarative: Checkout SCM</th><th>SetUp</th><th>Build</th><th>Deploy</th><th>Integration Test after deploy</th><th>Declarative: Post Actions</th></tr></thead><tbody><tr><td></td><td>1s</td><td>3min 45s</td><td>6s</td><td>37s</td><td>12s</td><td>411ms</td></tr><tr><td><div>#5 Jun01 16:08 31 commits</div></td><td>2s</td><td>5min 54s</td><td>10s</td><td>1min 0s</td><td>22s</td><td>634ms</td></tr><tr><td><div>#5 Jun01 15:59 No Changes</div></td><td>2s</td><td>1min 23s</td><td>9s</td><td>55s</td><td>22s</td><td>468ms</td></tr><tr><td><div>#4 Jun01 15:50 No Changes</div></td><td>2s</td><td>7min 16s</td><td>186ms</td><td>109ms</td><td>114ms</td><td>314ms</td></tr><tr><td><div>#3 May23 19:00 No Changes</div></td><td>949ms</td><td>1min 4s</td><td>8s</td><td>34s</td><td>7s</td><td>225ms</td></tr><tr><td><div>#3 May23 18:54 No Changes</div></td><td>620ms</td><td>1min 0s</td><td>9s</td><td>1min 16s</td><td>23s</td><td>476ms</td></tr><tr><td><div>#2 May23 18:47 No Changes</div></td><td>423ms</td><td>5min 55s</td><td>162ms</td><td>89ms</td><td>82ms</td><td>353ms</td></tr></tbody></table></div><div>Enlaces permanentes</div></div></div></div>		Declarative: Checkout SCM	SetUp	Build	Deploy	Integration Test after deploy	Declarative: Post Actions		1s	3min 45s	6s	37s	12s	411ms	<div>#5 Jun01 16:08 31 commits</div>	2s	5min 54s	10s	1min 0s	22s	634ms	<div>#5 Jun01 15:59 No Changes</div>	2s	1min 23s	9s	55s	22s	468ms	<div>#4 Jun01 15:50 No Changes</div>	2s	7min 16s	186ms	109ms	114ms	314ms	<div>#3 May23 19:00 No Changes</div>	949ms	1min 4s	8s	34s	7s	225ms	<div>#3 May23 18:54 No Changes</div>	620ms	1min 0s	9s	1min 16s	23s	476ms	<div>#2 May23 18:47 No Changes</div>	423ms	5min 55s	162ms	89ms	82ms	353ms
	Declarative: Checkout SCM	SetUp	Build	Deploy	Integration Test after deploy	Declarative: Post Actions																																																			
	1s	3min 45s	6s	37s	12s	411ms																																																			
<div>#5 Jun01 16:08 31 commits</div>	2s	5min 54s	10s	1min 0s	22s	634ms																																																			
<div>#5 Jun01 15:59 No Changes</div>	2s	1min 23s	9s	55s	22s	468ms																																																			
<div>#4 Jun01 15:50 No Changes</div>	2s	7min 16s	186ms	109ms	114ms	314ms																																																			
<div>#3 May23 19:00 No Changes</div>	949ms	1min 4s	8s	34s	7s	225ms																																																			
<div>#3 May23 18:54 No Changes</div>	620ms	1min 0s	9s	1min 16s	23s	476ms																																																			
<div>#2 May23 18:47 No Changes</div>	423ms	5min 55s	162ms	89ms	82ms	353ms																																																			
Explicación teórica de rollback	<p>Al hacer el SAM deploy se especifica si se desea revertir la pila de AWS CloudFormation si se produce un error durante una implementación. De forma predeterminada, si se produce un error durante una implementación, la pila AWS CloudFormation vuelve al último estado estable. Si durante la ejecución del SAM deploy se especifica--disable-rollback y se produce un error durante una implementación, los recursos que se crearon o actualizaron antes de que se produjera el error no se revertirán.</p> <p>Están disponibles las siguientes opciones:</p> <p>ROLLBACK— Hace retroceder la pila a un estado válido previo conocido.</p> <p>DELETE— Hace retroceder la pila a un estado válido previo conocido, si existe. De lo contrario, elimina la pila.</p> <p>DO_NOTHING— No revierte ni elimina la pila. El efecto es el mismo que el de--disable-rollback.</p> <p>El comportamiento predeterminado es ROLLBACK.</p>																																																								
Comentarios adicionales	<p>Al haber realizado las tres primeras construcciones antes de realizar los cambios en develop he tenido que hacer más construcciones con los cambios introducidos en las pruebas unitarias</p> <p>A parte de los cambios en las pruebas unitarias y las modificaciones del código en esta fase no he tenido que realizar ningún cambio más.</p>																																																								

5.3. /CD completo

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Indicar las salidas solicitadas del *pipeline*, bien copiando los logs, o adjuntando en la entrega final los ficheros de logs aparte.

Resultados a mostrar	Salida																				
Log pipeline	Los pipeline full cd																				
Captura de pantalla del pipeline	<div><div><div>Status</div><div></> Changes</div><div>▶ Construir ahora</div><div>⚙ Configurar</div><div>🗑 Borrar Pipeline</div><div>🔍 Full Stage View</div><div>✎ Rename</div><div>🔗 Pipeline Syntax</div></div><div><div>Historia de tareasTendencia</div><div>Filter builds.../</div><div><div>#21 jun_2023 14:06</div><div>#123 may_2023 16:58</div><div>Atom feed Para Todos</div><div>Atom feed para los errores</div></div></div><div><div>Pipeline PIPELINE-FULL-CD</div><div><div>Stage View</div><div><div>Average stage times: (Average full run time: ~ 7min 8s)</div><div><div>Jun01 16:06No Changes</div><div>May23 18:58No Changes</div></div><table><thead><tr><th>Declarative: Checkout SCM</th><th>Staging</th><th>Merge</th><th>Production</th><th>Declarative: Post Actions</th></tr></thead><tbody><tr><td>1s</td><td>2min 6s</td><td>4s</td><td>4min 54s</td><td>198ms</td></tr><tr><td>1s</td><td>2min 24s</td><td>6s</td><td>7min 40s</td><td>253ms</td></tr><tr><td>571ms</td><td>1min 48s</td><td>1s</td><td>2min 7s</td><td>143ms</td></tr></tbody></table></div></div><div><div>Enlaces permanentes</div><div><ul style="list-style-type: none">• "Última ejecución (#2) hace 52 Min"• "Última ejecución estable (#2) hace 52 Min"• "Última ejecución correcta (#2) hace 52 Min"• "Last completed build (#2) hace 52 Min"</div></div></div></div>	Declarative: Checkout SCM	Staging	Merge	Production	Declarative: Post Actions	1s	2min 6s	4s	4min 54s	198ms	1s	2min 24s	6s	7min 40s	253ms	571ms	1min 48s	1s	2min 7s	143ms
Declarative: Checkout SCM	Staging	Merge	Production	Declarative: Post Actions																	
1s	2min 6s	4s	4min 54s	198ms																	
1s	2min 24s	6s	7min 40s	253ms																	
571ms	1min 48s	1s	2min 7s	143ms																	
Comentarios adicionales	Se han realizado, antes de cambios en las pruebas y después de los cambios, sin problemas todas las fases del pipeline recogiendo todos los cambios realizados en las fases anteriores.																				

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Caso práctico 1. Apartado B

En este apartado se han de desarrollar una serie de conclusiones la aproximación realizada durante el apartado A.

Además de las conclusiones, el alumno debe de analizar alternativas a esta solución de despliegues realizados mediante SAM y mediante un entorno Jenkins. Para ello se propone que analice el uso completo del stack de AWS (CodeCommit, CodeBuild, CodeDeploy y CodePipeline) para identificar pros y contras y qué funcionalidades se podrían mejorar del ciclo de integración y despliegue continuos si se utilizaran y que planteen un *pipeline* teórico, identificando cada una de las etapas y qué servicios usarían para mejorar los que se han desarrollado durante la práctica. También sería interesante valorar alternativas Serverless como Serverless Framework, uso de herramientas como Sonarqube, etcétera.

Resultados a mostrar	Salida
Conclusiones finales	<p>La infraestructura planteada en el ejercicio está basado en Amazon AWS SAM CLI + Jenkins con los siguientes servicios:</p> <p>AWS SAM: es un framework de trabajo de código abierto para construir aplicaciones sin servidor. Proporciona una sintaxis abreviada para expresar funciones, APIs, bases de datos y mapeos de fuentes de eventos. Con solo unas pocas líneas por recurso, se puede definir la aplicación que se desee y modelarla utilizando YAML. Durante el despliegue, SAM transforma y expande la sintaxis de SAM en la sintaxis de AWS CloudFormation, lo que te permite construir aplicaciones sin servidor más rápidamente.</p> <p>Para comenzar a construir aplicaciones basadas en SAM, utiliza AWS SAM CLI. SAM CLI proporciona un entorno de ejecución similar a Lambda que te permite construir, probar y depurar aplicaciones localmente definidas mediante plantillas SAM o a través del AWS Cloud Development Kit (CDK). También puedes utilizar SAM CLI para desplegar tus aplicaciones en AWS o crear tuberías de integración y despliegue continuo (CI/CD) seguras que sigan las mejores prácticas e integren con los sistemas de CI/CD nativos y de terceros de AWS.</p> <p>Github: Servicio de control de versiones basado en git.</p> <p>Jenkins: Aplicación orquestadora.</p>
Pros	<p>Esta arquitectura es del tipo IaaS (Infraestructura como Servicio). EC2 es un servicio no gestionado.</p> <p>Modelo basado en que un proveedor se encarga de la distribución de la infraestructura necesaria: Externaliza el servidor, proporciona escalabilidad, elasticidad, disponibilidad, seguridad,</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	<p>automatización. El pago es por configuración y uso. Se tiene un control más preciso sobre cómo una solución maneja los cambios de carga, errores y situaciones en las que los recursos no están disponibles, pudiendo ajustar mejor los recursos a las necesidades reales de proyecto.</p> <p>Tanto Github como Jenkins son dos aplicaciones de uso generalizado y compatibles con AWS.</p>
Contras	<p>Es un servicio no gestionado que requiere de un alto conocimiento por parte del usuario en labores de administración para responder en caso de cambios en la carga, errores y disponibilidad de los recursos</p> <p>Ejemplo: si el servidor web no escala apropiadamente para dar servicio ante una mayor carga de tráfico o reemplazar las instancias en mal estado por otras en buen estado; en ese caso, debido a que Amazon EC2 es una solución gestionada por el usuario, este debiera de configurar manualmente una solución de escalado como Auto Scaling, gestión de la que un proveedor Cloud no se responsabiliza</p>
Arquitectura alternativa explicación	<p>La arquitectura que se podría plantear sería:</p> <p>AWS CodeBuild: es el servicio autogestionado de AWS responsable de compilar código fuente, ejecutar pruebas y producir software empaquetado que esté listo para ser desplegado. La mayor particularidad es la de no ser necesario administrar servidores dedicados para dicha labor de compilación y empaquetado.</p> <ul style="list-style-type: none"> - Descubre errores prematuramente del proceso de desarrollo, identificados en las fases iniciales, lo cual permite la corrección con una mayor facilidad. - Inclusión en procesos de integración continua y entrega continua (CI/CD) from scratch o ya existentes, a partir de las fases de origen del código («Source»), en comandos de construcción (en fichero buildspec.yml) o a través de la integración con Jenkins. - Como integrante de los servicios para la práctica CI/CD, viabiliza la operabilidad con AWS CodePipeline, el cual automatiza la compilación y las pruebas de código en CodeBuild cada vez que se confirma un cambio en el repositorio de origen (CodeCommit). <p>AWS CodeDeploy: automatiza las implementaciones de código en cualquier instancia, gestionando la complejidad de actualizar sus aplicaciones ante nuevas revisiones del mismo. Este proceso evita el tiempo de inactividad durante el despliegue de una aplicación. Cuenta con flexibilidad en el tipo de instancias sobre las que llevar a cabo el despliegue del artefacto software (Amazon EC2 o en servidores locales), soporte multi idioma y multi sistema operativo.</p> <p>AWS CodeCommit: permite almacenar y administrar de forma privada los activos de una empresa (documentos, código fuente y archivos binarios, entre otros) en la nube. AWS CodeCommit es un servicio de control de fuente administrado, seguro y altamente escalable que aloja repositorios Git privados, permitiendo un histórico de control de versiones con las modificaciones realizadas en el tiempo por cada uno de los miembros del equipo, área o departamento técnico. Las características más destacadas son las siguientes:</p> <ul style="list-style-type: none"> - Cifrado: permite transferir archivos desde y hacia AWS CodeCommit a través de los protocolos HTTPS y SSH. Los repositorios también se cifran automáticamente en reposo con AWS Key Management Service (AWS KMS) mediante el uso de

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	<p>claves específicas propias autogeneradas por los usuarios.</p> <ul style="list-style-type: none"> - Control de acceso: uso de AWS Identity and Access Management (IAM) para el control y monitorización del acceso a los datos relativos al código fuente. Monitorización de repositorios disponibles a través de AWS CloudTrail y AWS CloudWatch. - Alta disponibilidad y durabilidad: Backup automático para garantizar redundancia en varios centros de datos, apoyándose en otros servicios como Amazon S3 y Amazon DynamoDB para ello. - Repositorios ilimitados: sin limitación en la creación de repositorios de código independientes, permitiendo habilitar el almacenamiento de recursos de todo tipo y el histórico de cambios asociado. - Acceso e integración sencillos: como en el resto de servicios del catálogo de AWS, existen múltiples vías de conexión entre servicios: interfaz de línea de comandos, SDK de AWS y la consola de administración de AWS. <p>AWS CodePipeline: Aplicación orquestadora.</p> <p>Aplicaciones de una manera rápida y confiable desde el origen hasta su despliegue en el entorno productivo. Permite modelar visualmente el proceso de revisión del software, incorporando a través de una interfaz visual las subetapas de construcción, testing y despliegue del artefacto resultante de la etapa de construcción ante un cambio en el código fuente. Por último hay que destacar, al igual que el resto de servicios CI/CD incluidos en el portfolio de AWS, que se integra con herramientas de terceros y AWS.</p> <p>A continuación, vamos a estudiar un caso práctico (procedente de la propia documentación oficial de Amazon Web Services) a modo de ejemplo que permita al alumno comprender en esencia un diagrama de flujo de canalización («pipeline») en el que se apoyan todas las etapas o servicios de un ciclo de DevOps (AWS CodeCommit, AWS CodeBuild y AWS CodeDeploy) sobre este servicio padre.</p> <p>Lambda: servicio de computación que permite ejecutar código en la nube sin para ello requerir del aprovisionamiento y gestión de servidores de computación. Únicamente se paga por el tiempo de computación que la organización o particular consume, no existiendo cargo económico en caso de que el código deje de ejecutarse</p> <p>API Gateway: permite crear, publicar, mantener, monitorizar y securizar APIs, así como funciones Lambda</p> <p>Es un servicio completamente administrado que facilita la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala. Las API actúan como la "puerta de entrada" para que las aplicaciones accedan a los datos, la lógica empresarial o la funcionalidad de sus servicios de backend. Con API Gateway, se puede crear API RESTful y API WebSocket que permiten aplicaciones de comunicación bidireccional en tiempo real. API Gateway admite cargas de trabajo en contenedores y sin servidor, así como aplicaciones web.</p> <p>API Gateway gestiona todas las tareas implicadas en la aceptación y el procesamiento de hasta cientos de miles de llamadas a API simultáneas, entre ellas, la administración del tráfico, compatibilidad con CORS, el control de autorizaciones y acceso, la limitación controlada, el monitoreo y la administración de versiones de API. API Gateway no requiere pagos mínimos ni costos iniciales. Se paga por las llamadas a las API que se reciben y por la cantidad de datos</p>
--	---

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

	<p>salientes transferidos; además, con el modelo de precios por niveles de API Gateway, puede reducir sus costos a medida que cambie la escala de uso de las API.</p> <p>Cloud9: servicio que hace las veces de entorno integrado de desarrollo o IDE, contexto de uso recurrente en el ámbito de la programación</p> <p>Este servicio resuelve los siguientes problemas desde el punto de vista del desarrollador:</p> <ul style="list-style-type: none"> - Uso del entorno local para la implementación de aplicaciones diseñadas para operar en la nube. - Vía de implementación colaborativa de código. - Requerimiento de realizar múltiples tareas. - Dificultad de trabajar desde múltiples ubicaciones (áreas, departamentos, equipos portátiles). <p>S3: almacenamiento de objetos altamente seguro, duradero y escalable.</p> <p>Es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Clientes de todos los tamaños y sectores pueden almacenar y proteger cualquier cantidad de datos para prácticamente cualquier caso de uso, como los lagos de datos, las aplicaciones nativas en la nube y las aplicaciones móviles. Gracias a las clases de almacenamiento rentables y a las características de administración fáciles de usar, es posible optimizar los costos, organizar los datos y configurar controles de acceso detallados para cumplir con requisitos empresariales, organizacionales y de conformidad específicos.</p> <p>Ejemplos de uso:</p> <p>Copias de seguridad, archivos de datos, ejecución de aplicaciones nativas en la nube, etc.</p> <p>DynamoDB: servicio gestionado de bases de datos no relacionales para toda aplicación que necesite de un espacio persistencia consistente y con una latencia mínima casi inexistente y que soporta tanto documentos como modelo de almacenamiento clave-valor.</p> <p>CloudWatch: se utiliza para obtener visibilidad de todo el sistema sobre la utilización de recursos, el rendimiento de las aplicaciones y el estado operativo</p> <p>CloudWatch recopila y visualiza los registros, las métricas y los datos de evento en tiempo real en paneles automatizados para simplificar la infraestructura y el mantenimiento de aplicaciones.</p> <p>Se utiliza para visualizar datos de rendimiento, crear alarmas, analizar métricas, registros, estadísticas de registros.</p> <p>Se utiliza para automatizar respuestas a cambios con cloudwatch events .</p> <p>CloudFormation: Crea y administra recursos con plantilla.</p>
Arquitectura alternativa diagramas	Diagrama arquitectura

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Caso práctico 1. Apartado C

Como se indicaba en el enunciado de la práctica, a continuación, se ha de desarrollar una nueva función lambda desde cero, partiendo de todo el conocimiento adquirido durante esta primera parte de la práctica. Recordad que esta función lambda debe devolver una entrada de la ToDo list, traducida al idioma que se solicite a través del API. Para ello se recomienda el uso de las API's de los servicios de [Comprehend](#) y [Translate de AWS](#) si fueran necesarios. Una vez desarrollada la nueva función lambda, se ha de integrar con el resto de las componentes del API. Para ello habrá que incluir en el fichero **template.yml** la definición de la nueva lambda, el código fuente de esta función en el sitio adecuado de la estructura de directorios e integrarlo dentro del *pipeline* de CI/CD que se ha definido previamente, para ver cómo se propagan todos los cambios. El nuevo método de la API debe tener una estructura de este tipo:

- ▶ Método: GET
- ▶ PATH: /todos/<id>/<language>

A continuación, se ha de adjuntar la respuesta de tres invocaciones a la API, una en el idioma original del registro y dos con dos idiomas distintos (**Nota:** debe de funcionar con cualquier idioma que soporte la API de Translate de AWS y que contenga alfabeto latino):

Método	Resultado
/todos/<id>	
/todos/<id>/en	
/todos/<id>/fr	

Adicionalmente también se han de mostrar el código fuente desarrollado en la siguiente tabla, las inclusiones efectuadas en el fichero **template.yaml**, con el fin de tener la evidencia del trabajo realizado, así como las pruebas:

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Pruebas	Detalle a entregar
template.yaml	
Bloque código pruebas unitarias	
Bloque código pruebas integración	
Bloque código pruebas calidad	
Bloque código pruebas unitarias complejidad	

Pruebas

Se deberán de incluir dentro de las pruebas unitarias y de integración la nueva función *translate*, y validarlo dentro del pipeline de CI/CD diseñado en el apartado A.

Material para entregar

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Lorenzo Villoria	12/06/2023
	Nombre: Miguel Angel	

Rellenar completamente el documento que lleva por nombre **plantilla_solucion_CP1.docx**, con las secciones requeridas para los supuestos de los apartados A y B, de los cuales se piden evidencias (tablas, capturas de pantalla, logs, fragmentos de código, etc.) que reflejen el correcto progreso del alumno en el despliegue de los *pipelines* de CI/CD para ambas aproximaciones o *frameworks* de operativización.

Nota: para su entrega, dicho documento de plantilla se ha de exportar como PDF.

En la plantilla de la solución se ha de incorporar el enlace al repositorio de código del alumno con el código fuente como propuesta de la solución (en la portada de este presente documento).
