

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

Propuesta de solución

Caso práctico 1

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

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

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.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

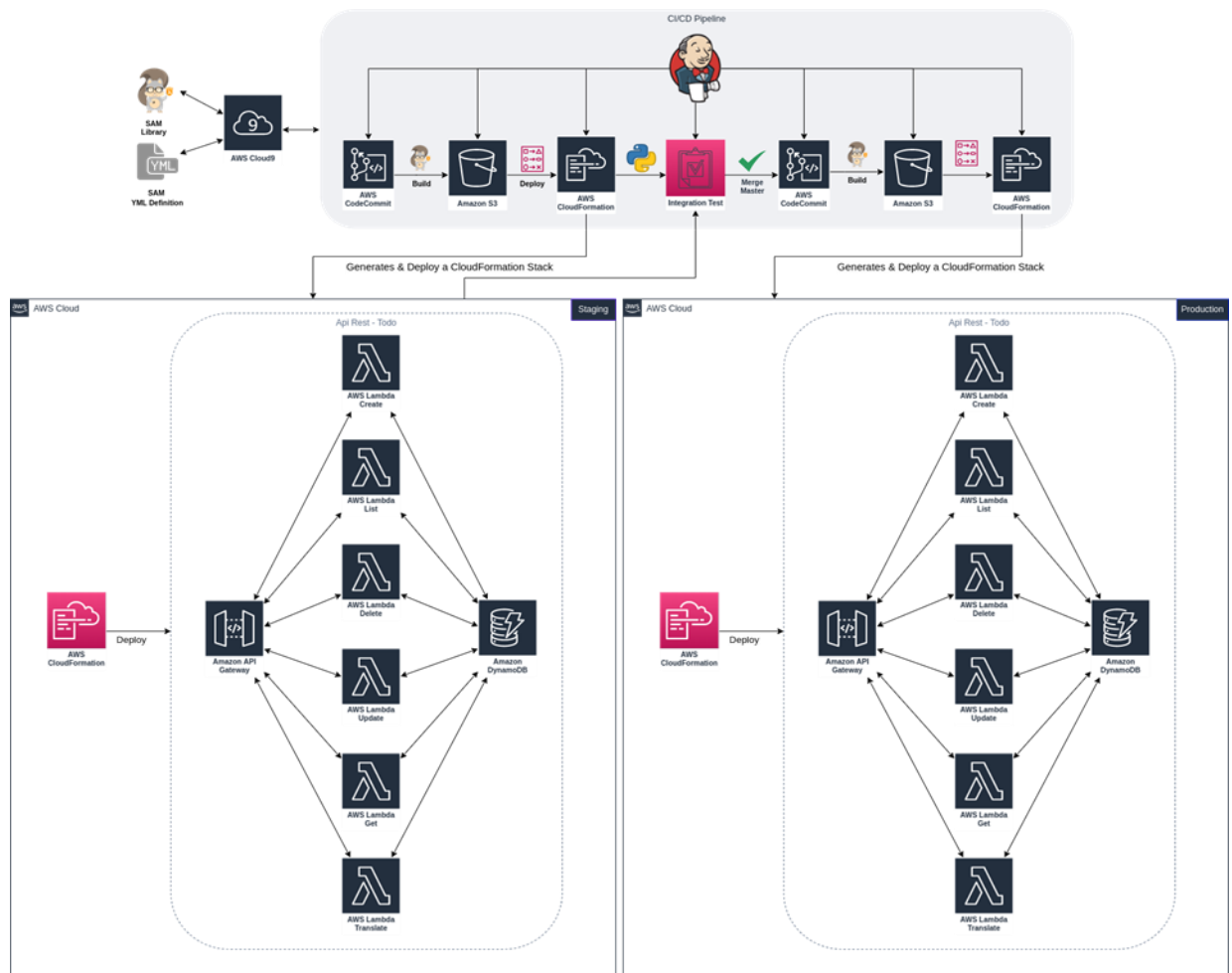
- [Simple Storage Service \(S3\)](#): Servicio de almacenamiento de objetos con sistema de versionado ante modificaciones en los mismos, alto rendimiento y finalidad multipropósito (Lago de datos y tracking IoT, sitios web, aplicaciones mobile, recurso de backup y archivado, entre otros).

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.

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:

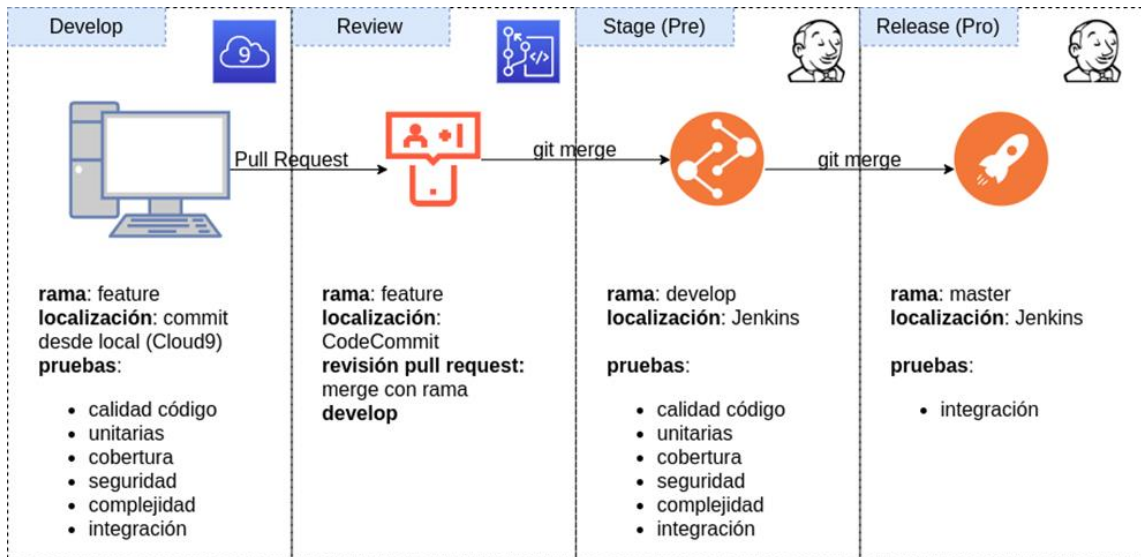
Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	



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.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	



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	origin https://github.com/aremox/todo-list-aws.git (fetch) origin https://github.com/aremox/todo-list-aws.git (push)

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

2.1. Ejecución del comando SAM funcionando:

Comando	Resultado
\$ sam --version	SAM CLI, version 1.33.0

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

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	Código fuente en Python de la aplicación
test	Ficheros de pruebas unitarias
pipelines	Ficheros de workflow de Jenkins
template.yaml	Incluye las definiciones de las funciones lambda y la definición de la BBDD
samconfig.toml	Es el fichero que almacena los parámetros de configuración que queremos para AWS SAM y poder lanzar el sam deploy de forma desatendida. Es similar a los ficheros de instalación desatendida.
localEnvironment.json	Fichero con las variables de entorno local que sobrescribe el endpoint de dynamodb para que apunte contra el docker de dynamo

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	7baca2134d4b150f6b9e65472884051e0df309a6e2d89127e9d711c8645ea77d
Levantar contenedor de docker	Unable to find image 'amazon/dynamodb-local:latest' locally latest: Pulling from amazon/dynamodb-local 3a461b3ae562: Pull complete 14d349bd5978: Pull complete 3e361eec6409: Pull complete Digest:

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	sha256:07e740ad576acdcfdc48676f9a153a93a8e35436ea36942d4c14939caeca8851 Status: Downloaded newer image for amazon/dynamodb-local:latest 1935cac13ec5070d86d12d5a045ac0549322a42f5b62cb26806376ce0359499a
Crear tabla en dynamodb local	<pre>{ "TableDescription": { "AttributeDefinitions": [{ "AttributeName": "id", "AttributeType": "S" }], "TableName": "local-TodosDynamoDbTable", "KeySchema": [{ "AttributeName": "id", "KeyType": "HASH" }], "TableStatus": "ACTIVE", "CreationDateTime": 1652607403.537, "ProvisionedThroughput": { "LastIncreaseDateTime": 0.0, "LastDecreaseDateTime": 0.0, "NumberOfDecreasesToday": 0, "ReadCapacityUnits": 1, "WriteCapacityUnits": 1 }, "TableSizeBytes": 0, "ItemCount": 0, "TableArn": "arn:aws:dynamodb:ddblocal:000000000000:table/local-TodosDynamoDbTable" } }</pre>
Empaquetar proyecto con SAM	Running PythonPipBuilder:ResolveDependencies Running PythonPipBuilder:CopySource Build Succeeded Built Artifacts : .aws-sam/build Built Template : .aws-sam/build/template.yaml

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	<p>Commands you can use next</p> <p>=====</p> <p>[*] Invoke Function: sam local invoke</p> <p>[*] Deploy: sam deploy --guided</p>
Levantar la API localmente	<p>Mounting DeleteTodoFunction at http://127.0.0.1:8081/todos/{id} [DELETE]</p> <p>Mounting UpdateTodoFunction at http://127.0.0.1:8081/todos/{id} [PUT]</p> <p>Mounting ListTodosFunction at http://127.0.0.1:8081/todos [GET]</p> <p>Mounting CreateTodoFunction at http://127.0.0.1:8081/todos [POST]</p> <p>Mounting GetTodoFunction at http://127.0.0.1:8081/todos/{id} [GET]</p> <p>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. You only need to restart SAM CLI if you update your AWS SAM template</p> <p>2022-05-15 09:37:50 * Running on http://127.0.0.1:8081/</p>

NOTA: Se ha creado un script “entorno_local.sh” que automatiza el arranque del entorno local

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: Arenas Morante	23/05/2022
	Nombre: Iván	

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" }' json_pp
	resultado	{ "statusCode" : 200, "body" : "{ \"id\": \"c1e55b74-d434-11ec-8834-0242ac110002\", \"text\": \"Learn Serverless\", \"checked\": false, \"createdAt\": \"1652608359.7930732\", \"updatedAt\": \"1652608359.7930732\" }"
List	comando	curl http://127.0.0.1:8081/todos json_pp
	resultado	[{ "createdAt" : "1652608359.7930732", "text" : "Learn Serverless", "updatedAt" : "1652608359.7930732", "id" : "c1e55b74-d434-11ec-8834-0242ac110002", "checked" : false }]
Get	comando	curl http://127.0.0.1:8081/todos/c1e55b74-d434-11ec-8834-0242ac110002 json_pp
	resultado	[{ "text" : "Learn Serverless", "id" : "c1e55b74-d434-11ec-8834-0242ac110002", "updatedAt" : "1652608359.7930732", "checked" : false, "createdAt" : "1652608359.7930732" }]
Update	comando	curl -X PUT http://127.0.0.1:8081/todos/ c1e55b74-d434-11ec-8834-0242ac110002 --data '{ "text": "Learn Serverless", "checked": true }' json_pp

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	resultado	{ "id" : "c1e55b74-d434-11ec-8834-0242ac110002", "updatedAt" : 1652608511638, "text" : "Learn Serverless", "createdAt" : "1652608359.7930732", "checked" : true }
Delete	comando	curl -X DELETE http://127.0.0.1:8081/todos/c1e55b74-d434-11ec-8834-0242ac110002
	resultado	No da salida, pero devuelve un 200 (ok) 2022-05-15 09:56:03 127.0.0.1 - - [15/May/2022 09:56:03] "DELETE /todos/c1e55b74-d434-11ec-8834-0242ac110002 HTTP/1.1" 200 -

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: Arenas Morante	23/05/2022
	Nombre: Iván	

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

Función	Endpoint
Create	<code>https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos/ '{ "text": "Learn Serverless" }'</code>
List	<code>https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos</code>
Get	<code>https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}</code>
Update	<code>https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos/{id} '{ "text": "Learn Serverless", "checked": true }'</code>
Delete	<code>https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos/{id}</code>

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

4.1. Invocaciones a los métodos del API

Función	In/out	Script
Create	comando	<code>curl -X POST https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos --data '{ "text": "Learn Serverless" }'</code>
	resultado	<pre>{ "body" : "{ \"id\": \"0bda2e82-d442-11ec-9436-52f558e96de6\", \"text\": \"Learn Serverless\", \"checked\": false, \"createdAt\": \"1652614067.3419209\", \"updatedAt\": \"1652614067.3419209\"}", "statusCode" : 200 }</pre>
List	comando	<code>curl https://tz0tn4nqc9.execute-api.us-east-1.amazonaws.com/Prod/todos</code>
	resultado	<pre>[{ "createdAt" : "1652614067.3419209", "checked" : false, "updatedAt" : "1652614067.3419209", "id" : "0bda2e82-d442-11ec-9436-52f558e96de6", "text" : "Learn Serverless" }</pre>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

		}]
Get	comando	curl https:// tz0tn4nqc9 .execute-api.us-east-1.amazonaws.com/Prod/todos/<id>
	resultado	{ "text" : "Learn Serverless", "createdAt" : "1652614067.3419209", "id" : "0bda2e82-d442-11ec-9436-52f558e96de6", "checked" : false, "updatedAt" : "1652614067.3419209" }
Update	comando	curl -X PUT https:// tz0tn4nqc9 .execute-api.us-east-1.amazonaws.com/Prod/todos/<id> --data '{ "text": "Learn python and more", "checked": true }'
	resultado	{ "id" : "0bda2e82-d442-11ec-9436-52f558e96de6", "updatedAt" : 1652614068557, "text" : "Learn Serverless", "createdAt" : "1652614067.3419209", "checked" : true }
Delete	comando	curl -X DELETE https:// tz0tn4nqc9 .execute-api.us-east-1.amazonaws.com/Prod/todos/<id>
	resultado	No da salida, pero devuelve un 200 (ok)

NOTA: Se ha creado un script que automatiza esta comprobación “curl.sh”

La salida de dicho script es la siguiente

```
sh curl.sh manual
CREATE
{
  "body" : "{\"id\": \"0bda2e82-d442-11ec-9436-52f558e96de6\", \"text\": \"Learn Serverless\", \"checked\": false, \"createdAt\": \"1652614067.3419209\", \"updatedAt\": \"1652614067.3419209\"}",
  "statusCode" : 200
}
```

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

```
List
[
  {
    "createdAt" : "1652614067.3419209",
    "checked" : false,
    "updatedAt" : "1652614067.3419209",
    "id" : "0bda2e82-d442-11ec-9436-52f558e96de6",
    "text" : "Learn Serverless"
  }
]
GET
{
  "text" : "Learn Serverless",
  "createdAt" : "1652614067.3419209",
  "id" : "0bda2e82-d442-11ec-9436-52f558e96de6",
  "checked" : false,
  "updatedAt" : "1652614067.3419209"
}
UPDATE
{
  "id" : "0bda2e82-d442-11ec-9436-52f558e96de6",
  "updatedAt" : 1652614068557,
  "text" : "Learn Serverless",
  "createdAt" : "1652614067.3419209",
  "checked" : true
}
DELETE 0bda2e82-d442-11ec-9436-52f558e96de6
BORRAR TODOS
```

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

Listar BBDD

[]

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>[*] Invoke Function: sam local invoke</p> <p>[*] Deploy: sam deploy --guided</p>
Log sam deploy	<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>[*] Invoke Function: sam local invoke</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	<pre>[*] Deploy: sam deploy --guided ~/todo-list-aws (feature-apartado_A) \$ sam deploy File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload File with same data already exists at todo-list-aws/e0bdedd6a86b55249711b9315a662165, skipping upload Deploying with following values ===== Stack name : todo-list-aws Region : us-east-1 Confirm changeset : True Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-12cqvx8xu951le Capabilities : ["CAPABILITY_IAM"] Parameter overrides : {"Stage": "default"} Signing Profiles : {} Initiating deployment ===== File with same data already exists at todo-list-aws/b6fc14abf474b14d3b91f6df2b12aee2.template, skipping upload Waiting for changeset to be created.. Error: No changes to deploy. Stack todo-list-aws is up to date</pre>
--	---

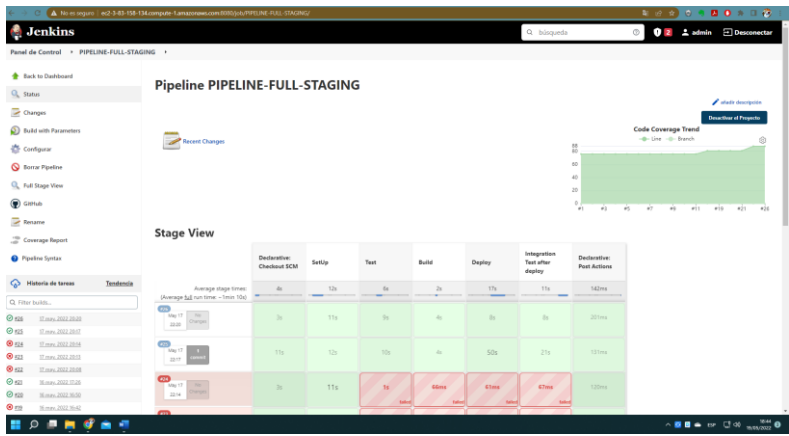
5. Creación de pipelines de Jenkins para despliegue de arquitectura completa

5.1. Pipeline de staging

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

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

hacer *rollback* de la arquitectura, con el fin de analizar los posibles fallos y corregirlos.

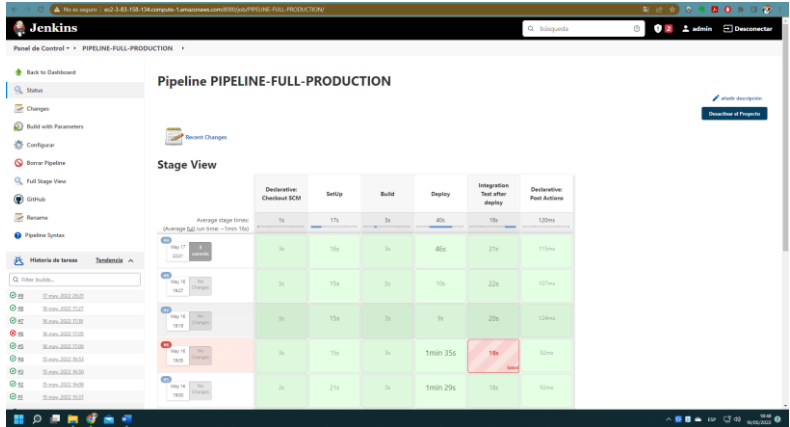
Resultados a mostrar	Salida
Log pipeline	https://alumnosunir-my.sharepoint.com/:f:/g/personal/ivan_arenas534_comunidadunir_n_et/Es71g5HFEUJPvVPPTt0pKCCBsZ3sGuE4KBNTDiWo-2ZUGg?e=yGcYmi
Captura de pantalla del pipeline	
Comentarios adicionales	Se sube la cobertura al 89%

5.2. Pipeline de Producción

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	https://alumnosunir-my.sharepoint.com/:f:/g/personal/ivan_arenas534_comunidadunir_n_et/Es71g5HFEUJPvVPPTt0pKCCBsZ3sGuE4KBNTDiWo-2ZUGg?e=yGcYmi

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

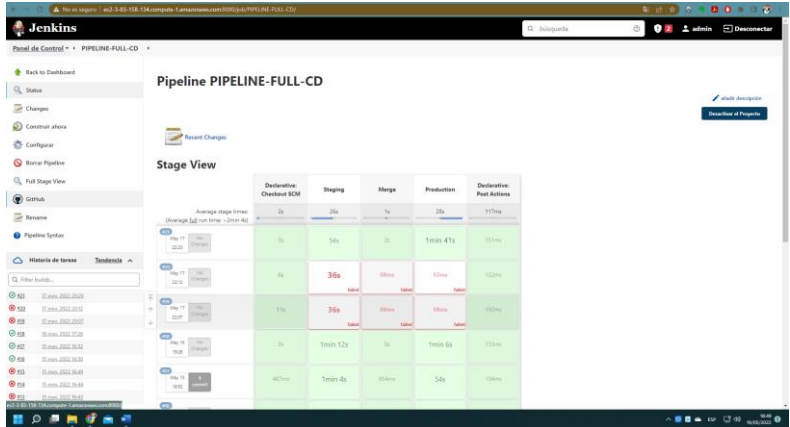
Captura de pantalla del pipeline	
Explicación teórica de <i>rollback</i>	<p>Tendríamos varias opciones:</p> <ul style="list-style-type: none"> - Descargar una versión previa del código de la rama Master y repetir el ciclo con ella - Tener una versión del artefacto previo desplegado y volver a desplegarlo - Sistema de “switcheo” entre dos versiones desplegadas. Se mantiene la anterior y se borra cuando se comprueba que funciona la nueva, si la nueva falla se deja la existente. <p>Esta claro que la mas interesante por temas de tiempo es la última.</p>
Comentarios adicionales	

5.3. CI/CD completo

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	https://alumnosunir-my.sharepoint.com/:f:/g/personal/ivan_arenas534_comunidadunir_net/Es71g5HFEUJPvVPPTt0pKCcBsZ3sGuE4KBNTDiWo-2ZUGg?e=yGcYmi

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

Captura de pantalla del pipeline	
Comentarios adicionales	Los pipelines que han fallado han sido por temas de configuración del id de credenciales ssh

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

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>Como conclusiones podemos indicar que la cantidad de alternativas en el mercado es amplia y que puede adaptarse a las necesidades de cada usuario u organización tanto en costes como en facilidad de uso. Por tanto, no es necesario la utilización de los productos indicados en el caso práctico para cubrir las necesidades de un proyecto pudiéndose sustituir por las indicadas a continuación:</p> <p>GitHub -> CodeCommit o gitlab Jenkins -> AWS CodePipeline o CoodeBuild Pruebas de seguridad -> owasp zap Pruebas de calidad del software -> sonarQube Pruebas de integración y rendimiento -> jMeter Lambda -> Serverless Framework</p>
Pros	<p>Las indicadas para el ejercicio son fáciles de integrar y la curva de aprendizaje es rápida.</p> <p>De la solución propuesta, algunos productos al estar implementadas directamente en AWS son más sencillas de integrar.</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	<p>Productos como sonarQube te permiten gestionar de forma centralizada las pruebas de calidad del software</p>
Contras	<p>De la solución propuesta, algunos productos al ser más potentes, tienen una curva de aprendizaje más complicada. Esto mismo pasa con los problemas y estabilidad que se puede ver afectada por la envergadura de las soluciones.</p>
Arquitectura alternativa explicación	<p>El funcionamiento es similar a la solución propuesta en el caso práctico, sustituyendo unos productos por otros.</p> <p>Comenzamos con la subida del código a CodeCommit. Como sistema de orquestación usamos AWS CodePipeline que descarga y compila el código para desplegarlo en preproducción. Antes de desplegar, ejecutamos las pruebas de calidad del software en sonarQube y pasamos un análisis de seguridad con owasp zap. Como pruebas de integración podemos ejecutar unos test grabados en jmeter y obtener también unas pruebas de rendimiento</p> <p>Para desplegar en producción solo descargaremos el código de preproducción y lo subiremos a producción donde realizaremos el despliegue.</p> <p>Como las pruebas ya están hechas en preproducción, solo repetiremos las de integración con jmeter sin alterar los datos de la BBDD.</p>
Arquitectura alternativa diagramas	

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

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>	<pre>{ "checked" : false, "createdAt" : "1652697378.8178983", "text" : "Hacer apartado C", "updatedAt" : "1652697378.8178983", "id" : "055c90fa-d504-11ec-91cf-ee7acd9cd62c" }</pre>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

/todos/<id>/en	<pre>{ "id" : "055c90fa-d504-11ec-91cf-ee7acd9cd62c", "updatedAt" : "1652697378.8178983", "checked" : false, "text" : "Make section C", "createdAt" : "1652697378.8178983" }</pre>
/todos/<id>/fr	<pre>{ "checked" : false, "updatedAt" : "1652697378.8178983", "id" : "055c90fa-d504-11ec-91cf-ee7acd9cd62c", "text" : " Créer la section C", "createdAt" : "1652697378.8178983" }</pre>

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:

Pruebas	Detalle a entregar
template.yaml	<pre>[...] TranslateTodoFunction: Type: AWS::Serverless::Function Properties: CodeUri: src/ Role: !Sub "arn:aws:iam::\${AWS::AccountId}:role/LabRole" Handler: translate.translate Runtime: python3.7 Events: Create: Type: Api Properties: Path: /todos/{id}/{idioma} Method: get [...]</pre> <p>TranslateTodoApi: Description: "API Gateway endpoint URL for \${opt:stage} stage for Get TODO"</p>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	Value: !Sub https://\$\${ServerlessRestApi}.execute-api.\$\${AWS::Region}.amazonaws.com/Prod/todos/{id}/{idiotoma} [...] [...]
Bloque código pruebas unitarias	[...] <pre>def test_translate_todo(self): print ('-----') print ('Start: test_translate_todo') from src.todoList import get_translate translation = get_translate(self.text, "en", self.dynamodb) print ('Response translate en:' + str(translation)) self.assertEqual(None, translation) translation = get_translate(self.text, "fr", self.dynamodb) print ('Response translate fr:' + str(translation)) self.assertEqual(None, translation) print ('End: test_delete_todo') [...]</pre>
Bloque código pruebas integración	<pre>def test_api_translatetodo(self): print('-----') print('Starting - integration test translate TODO') #Add TODO url = BASE_URL+"/todos" data = { "text": "Integration text example - translate" } response = requests.post(url, data=json.dumps(data)) json_response = response.json() print('Response Add Todo: ' + str(json_response)) jsonbody= json.loads(json_response['body']) ID_TODO = jsonbody['id'] print ('ID todo:'+ID_TODO)</pre>

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

	<pre> self.assertEqual(response.status_code, 200, "Error en la petición API a {url}") self.assertEqual(jsonbody['text'], "Integration text example - translate", "Error en la petición API a {url}") #Test translate TODO url = BASE_URL+"/todos/"+ID_TODO+"/en" response = requests.get(url) json_response = response.json() print('Response translate Todo: '+ str(json_response)) self.assertEqual(response.status_code, 200, "Error en la petición API a {url}") self.assertEqual(json_response['text'], "Integration text example - translate", "Error en la petición API a {url}") #Delete TODO to restore state url = BASE_URL+"/todos/"+ID_TODO response = requests.delete(url) self.assertEqual(response.status_code, 200, "Error en la petición API a {url}") print('End - integration test translate TODO') </pre>
Bloque código pruebas calidad	No cambia
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.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps & Cloud	Apellidos: Arenas Morante	23/05/2022
	Nombre: Iván	

Material para entregar

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).
