

CS2032 - Cloud Computing (Ciclo 2024-2)

Serverless

Semana 11 - Taller 4: UUID. Environment y Resources
en framework serverless

ELABORADO POR: GERALDO COLCHADO

Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. Ejercicio 2: Environment variables
4. Ejercicio 3: Resources (DynamoDB) por stage
5. Ejercicio 4: Eliminar lo desplegado
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

UUID. Environment y Resources

Objetivo del Taller 4

- Aprender a utilizar UUID
- Aprender a utilizar variables de entorno
- Aprender a crear recursos (DynamoDB) por stage
- Aprender a eliminar lo desplegado

Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. **Ejercicio 1: UUID**
3. Ejercicio 2: Environment variables
4. Ejercicio 3: Resources (DynamoDB) por stage
5. Ejercicio 4: Eliminar lo desplegado
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

UUID. Environment y Resources

Ejercicio 1: UUID

¿Qué es un UUID? | Te lo explico en 4 minutos

YouTube · Manuel Zapata · 15 may. 2020

UUID | Es un **número único** que se utiliza como **identificador**



“What is a Version 1 UUID?”

A Version 1 UUID is a universally unique identifier that is generated using a timestamp and the MAC address of the computer on which it was generated.

What is a version 4 UUID?

A Version 4 UUID is a universally unique identifier that is generated using random numbers.”

UUID. Environment y Resources

Ejercicio 1: UUID

- Paso 1: Ingrese a la máquina virtual “MV para serverless”
- Paso 2: Cree un directorio /home/ubuntu/python
- Paso 3: Cree y ejecute el archivo **generar-uuid.py**

```
import uuid
uuidv1 = str(uuid.uuid1())
uuidv4 = str(uuid.uuid4())
print("UUID en V1: ", uuidv1)
print("UUID en V4: ", uuidv4)
```

UUID. Environment y Resources

Ejercicio 1: UUID

- Paso 4: Ejecutar el programa

```
:~/python $ python3 generar-uuid.py  
UUID en V1: 89b69156-68df-11ed-9463-127e3d0cd2c1  
UUID en V4: 4c463431-cf5a-4ca1-80a3-6921e33bb514
```

Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. **Ejercicio 2: Environment variables**
4. Ejercicio 3: Resources (DynamoDB) por stage
5. Ejercicio 4: Eliminar lo desplegado
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

UUID. Environment y Resources

Ejercicio 2: Environment variables

- **Paso 1:** Visualice las variables de entorno de Linux
\$ printenv
- **Paso 2:** Cree las siguientes variables de entorno
\$ NOMBRE_ALUMNO="Geraldo Colchado Ruiz"
\$ NOMBRE_EMPRESA="UTEC"
\$ export NOMBRE_ALUMNO
\$ export NOMBRE_EMPRESA

UUID. Environment y Resources

Ejercicio 2: Environment variables

- **Paso 3:** Verifique que se hayan creado

```
$ printenv
```

```
...
```

```
NOMBRE_ALUMNO=Geraldo Colchado Ruiz
```

```
...
```

```
NOMBRE_EMPRESA=UTEC
```

```
...
```

UUID. Environment y Resources

Ejercicio 2: Environment variables

- **Paso 4:** Cree /home/ubuntu/python/**variables-entorno.py**

```
import os
alumno = os.environ["NOMBRE_ALUMNO"]
empresa = os.environ["NOMBRE_EMPRESA"]
print(alumno)
print(empresa)
```

- **Paso 5:** Ejecute el programa

```
:~/python $ python3 variables-entorno.py
Geraldo Colchado Ruiz
UTEC
```

Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. Ejercicio 2: Environment variables
4. **Ejercicio 3: Resources (DynamoDB) por stage**
5. Ejercicio 4: Eliminar lo desplegado
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

UUID. Environment y Resources

Ejercicio 3: Resources (DynamoDB) por stage

Elabore un lambda que reciba como entrada un `tenant_id` y un texto y lo almacene en una tabla DynamoDB. Considere lo siguiente:

- Publicar como API Rest con Api Gateway
- Debe usar una **tabla con nombre diferente según el stage** (Una para desarrollo, otra para test y otra para producción). Use variables de entorno.
- Debe usar el campo `tenant_id` como clave de partición y el campo `uuid` como clave de ordenamiento (Use UUID V1)
- Debe automatizar el despliegue con framework serverless (La **tabla debe ser creada automáticamente** desde aquí)

UUID. Environment y Resources

Ejercicio 3: Resources (DynamoDB) por stage

- **Paso 1:** En “MV Serverless” actualice el archivo de credenciales
/home/ubuntu/.aws/credentials
- **Paso 2:** Cree el repositorio api-comentario en github con archivos indicados por el docente. Analice el contenido de los archivos. Realice el git clone en
/home/ubuntu/
- **Paso 3:** Realice el despliegue automático en stage de desarrollo
\$ serverless login
\$ serverless deploy

UUID. Environment y Resources

Ejercicio 3: Resources (DynamoDB) por stage

- Paso 4:** Pruebe en postman y verifique la variable de entorno de lambda y tabla

POST ▼ <https://2togxfsrj.execute-api.us-east-1.amazonaws.com/dev/comentario/crear>

Params Authorization Headers (7) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {  
2   .... "tenant_id": "UTEC",  
3   .... "texto": "Este es mi primer comentario"  
4 }
```

Variables de entorno (1)

Las variables de entorno que se muestran a continuación se cifran en reposo con la clave del servicio Lambda predeterminada.

Editar

Clave	Valor
TABLE_NAME	dev-t_comentarios

dev-t_comentarios

tenant_id ▼	uuid ▼	detalle
UTEC	cf32d529-194d-11ee-9f51-41757142260f	{ "texto": { "S": "Este es mi primer comentario" } }

UUID. Environment y Resources

Ejercicio 3: Resources (DynamoDB) por stage

- **Paso 5:** Realice el despliegue automático en stage de test y producción

\$ serverless deploy --stage test

\$ serverless deploy --stage prod

- **Paso 6:** Pruebe en postman y verifique las variables de entorno de lambda y tablas de test y producción

Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. Ejercicio 2: Environment variables
4. Ejercicio 3: Resources (DynamoDB) por stage
5. **Ejercicio 4: Eliminar lo desplegado**
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

UUID. Environment y Resources

Ejercicio 4: Eliminar lo desplegado

- **Paso 1:** Desde el directorio `/home/ubuntu/api-comentario` ejecute estos comandos para eliminar todo lo desplegado

```
$ serverless remove
```

```
$ serverless remove --stage test
```

```
$ serverless remove --stage prod
```

- **Paso 2:** Verifique que se hayan eliminado todos los objetos (Api Gateway, Lambdas y DynamoDB) tanto de desarrollo, test y producción.

Contenido

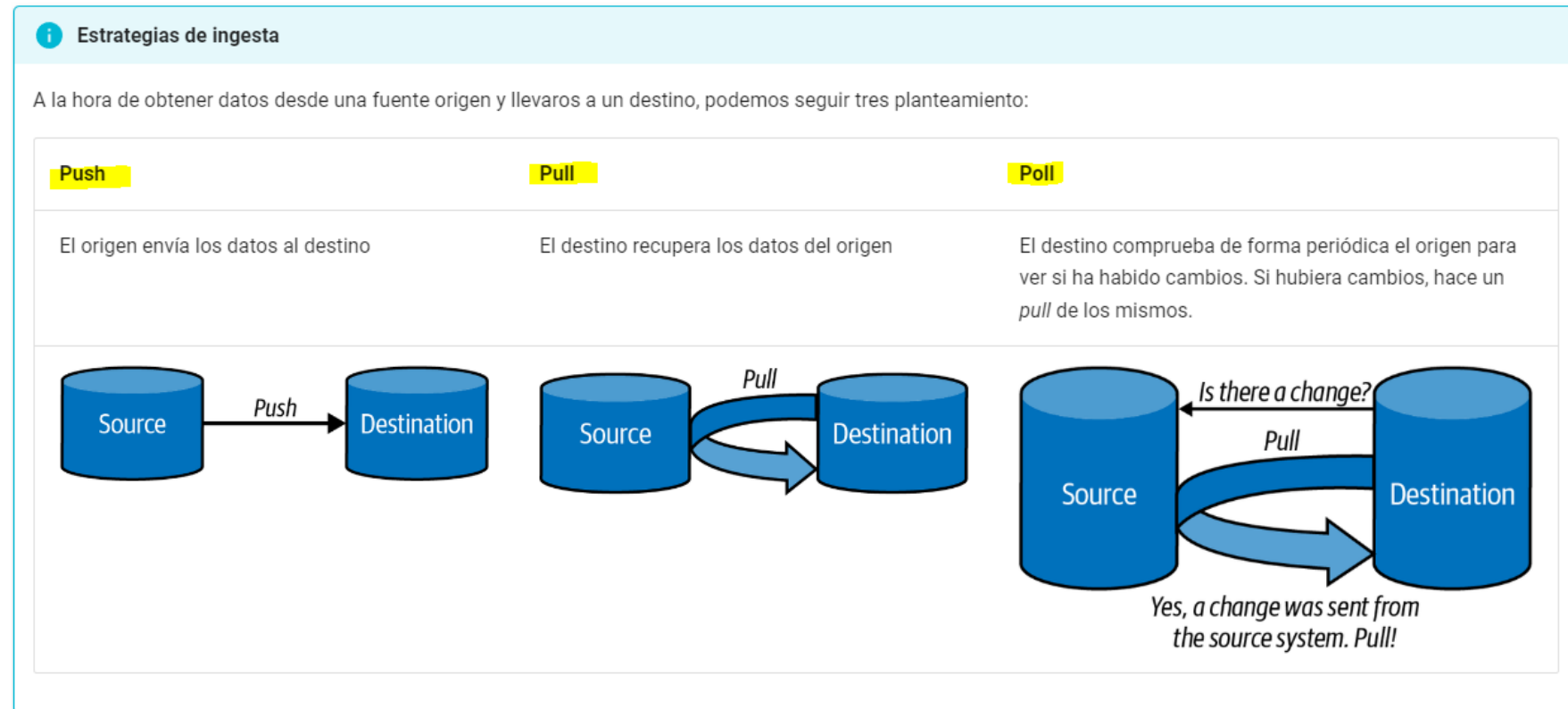
UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. Ejercicio 2: Environment variables
4. Ejercicio 3: Resources (DynamoDB) por stage
5. Ejercicio 4: Eliminar lo desplegado
6. **Ejercicio 5: Ejercicio propuesto**
7. Cierre

UUID. Environment y Resources

Ejercicio 5: Ejercicio propuesto (30 puntos)

- Investigue (*) y modifique el `serverless.yml` para crear automáticamente un **bucket S3 de ingesta** por stage
- Modificar la función `comentario.lambda_handler` para grabar el json del comentario como archivo en el bucket s3 de ingesta que le corresponde según stage (**Estrategia de “Ingesta Push”**)



Contenido

UUID. Environment y
Resources en framework
serverless

1. Objetivo del taller 4
2. Ejercicio 1: UUID
3. Ejercicio 2: Environment variables
4. Ejercicio 3: Resources (DynamoDB) por stage
5. Ejercicio 4: Eliminar lo desplegado
6. Ejercicio 5: Ejercicio propuesto
7. Cierre

Cierre:

UUID. Environment y Resources - Qué aprendimos?

- Utilizar UUID
- Utilizar variables de entorno
- Crear recursos (DynamoDB) por stage
- Eliminar lo desplegado

Gracias

Elaborado por docente: Geraldo Colchado