Documento de Arquitectura – Célula 2

Proyecto: Scheduling Serverless para Generación Automática de Órdenes de Compra

Empresa ficticia: *Acme Inc.*
Célula: 2
Integrantes: Alejandra Roa, Ulises Hernández
Fecha: 2025
Versión: 1.0

Índice

10. Estimación de Costos (Guía)
11. Control de Cambios

---

1. Introducción

Este documento describe la solución asignada a la **Célula 2**, siguiendo los lineamientos del Documento Maestro del proyecto “Arquitecturas Serverless y Solutions Architect en AWS”.

La arquitectura se basa en un sistema de **scheduler serverless**, capaz de ejecutar tareas automáticas en horarios definidos, usar Python, desplegar por IaC en múltiples ambientes, y cumplir con las políticas corporativas de seguridad, incluyendo KMS, VPC Endpoints, Cognito, zona de inspección y CI/CD.

---

2. Escenario Acme Inc.

La empresa ficticia **Acme Inc.**, dedicada a la venta global de gadgets y dispositivos innovadores, ha determinado que necesita un sistema automatizado para **generar órdenes de compra internas**.

Actualmente, la reposición de gadgets se realiza manualmente cada cierto tiempo por el equipo de operaciones. Acme Inc. desea automatizar este proceso mediante un servicio que:

- Programe tareas en horarios específicos
- Genere automáticamente órdenes de compra según reglas internas
- Registre dichas órdenes en una base de datos
- Permita consultas seguras de órdenes programadas y ejecutadas
- Exponga endpoints protegidos para administración del scheduler

El sistema debe ser completamente **serverless**, altamente seguro, fácil de desplegar y con pipelines automáticos.

---

3. Objetivo de la Arquitectura

Construir un servicio que:

- Permita programar tareas mediante una API
- Ejecute tareas en horarios definidos usando **Amazon EventBridge Scheduler**
- Cree órdenes de compra de gadgets de manera automática
- Registre resultados en DynamoDB
- Cifre y proteja toda la información con KMS y Cognito
- Se despliegue en sandbox, pre-producción y producción por CI/CD
- Exponga pruebas mediante curl

---

4. Alcance

Incluye

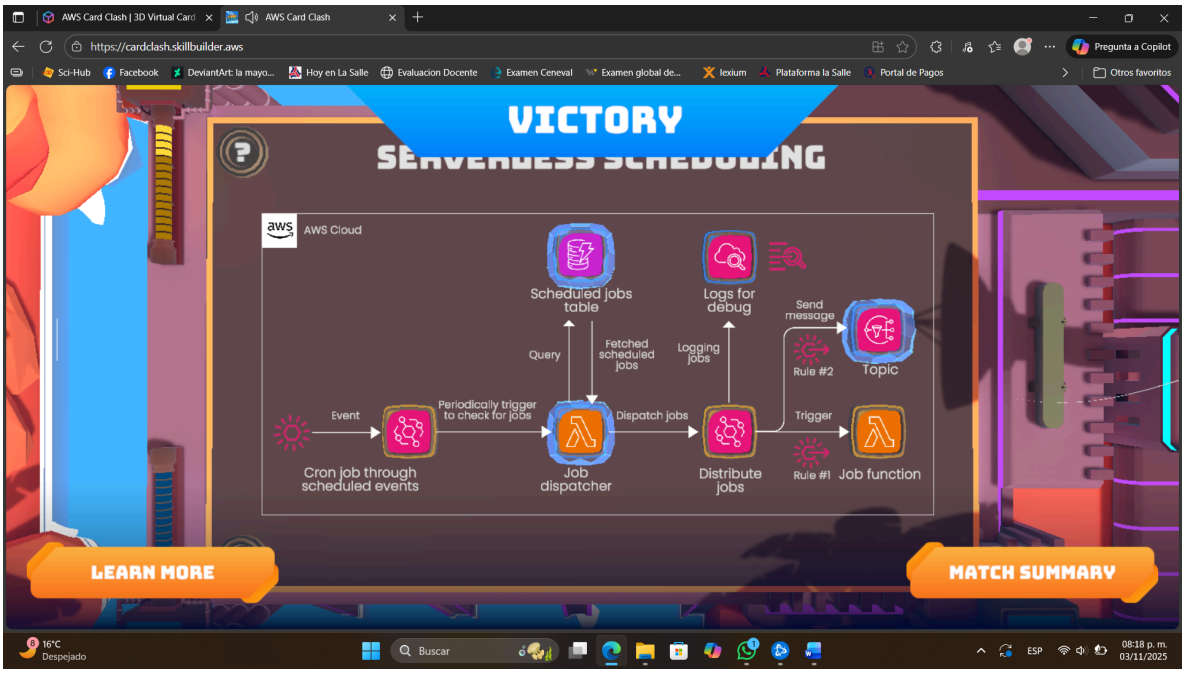
- Programación de tareas (scheduling) vía EventBridge
- Funciones Lambda para generar órdenes de compra
- Exposición de APIs para crear, consultar y cancelar schedules
- Persistencia de órdenes en DynamoDB
- Generación de datos de prueba con GenAI
- Pipeline completo CI/CD
- Seguridad con KMS, Cognito, VPC Endpoints y zona de inspección
- Diagrama draw.io

No incluye

- Integración con inventario real
- Flujos complejos entre microservicios
- Interfaces gráficas completas (solo pruebas con curl)

---

5. Arquitectura Propuesta



5.1 Componentes Principales

- **Amazon API Gateway**
    - Expone APIs:
        - Crear nuevo schedule
        - Consultar schedules
        - Consultar órdenes generadas
        - Cancelar schedule
- **Amazon Cognito**

    - Autenticación de usuarios (JWT)
    - Uso obligatorio en todos los endpoints
- **AWS Lambda (Python)**
    - Servicio de administración de schedules
    - Servicio ejecutor de órdenes de compra
    - Lógica de negocio basada en reglas
- **Amazon EventBridge Scheduler**
    - Programa tareas recurrentes o puntuales
    - Apunta a una Lambda ejecutora
- **Amazon DynamoDB**
    - Tablas para:
        - Órdenes generadas
        - Definiciones de schedules (si se replica internamente)
- **S3**
    - Almacenamiento de logs o snapshots si la célula lo requiere
    - Cifrado SSE-KMS
- **AWS KMS**
    - Cifrado para:
        - DynamoDB
        - S3
        - Variables de entorno
        - Logs (si aplica)
        - Secrets Manager
- **VPC + Endpoints + Zona de Inspección**

    - Lambdas en subred privada
    - Endpoints para DynamoDB, S3, Logs, Secrets Manager
- **CodePipeline / CodeBuild / CloudFormation**
    - Pipeline de despliegue automático

5.2 Flujo de Datos

1. Usuario autenticado solicita crear un schedule mediante API Gateway.
2. Lambda registra parámetros y crea una regla de EventBridge Scheduler.
3. En el horario programado, EventBridge invoca una Lambda ejecutora.
4. Lambda genera automáticamente la orden de compra siguiendo reglas definidas.
5. El resultado de la orden se almacena en DynamoDB.
6. Usuario consulta órdenes mediante un endpoint protegido por Cognito.

---

6. Lineamientos Técnicos Obligatorios

6.1 Infraestructura como Código

- Toda la arquitectura se define usando CloudFormation.
- No se permiten cambios manuales en ambientes objetivo ni producción.

6.2 Python y Datos GenAI

- Lógica en Python.
- Datos sintéticos de órdenes de gadgets (mín. 50) generados con GenAI.

6.3 API Gateway + Cognito

- Endpoints protegidos.
- Cognito administra roles y permisos.

6.4 Seguridad

- Lambdas en subredes privadas.
- VPC Endpoints obligatorios.
- Cifrado con AWS KMS.
- Zona de inspección para tráfico saliente.

6.5 Pipeline CI/CD

- GitHub → CodePipeline → CodeBuild → CloudFormation
- Despliegue automático sandbox → pre-producción → producción

---

7. Entregables

1. Repositorio GitHub completo
2. Pipeline CI/CD funcional
3. Despliegues validados en sandbox y pre-producción
4. Diagrama de arquitectura (draw.io)
5. Datos generados con GenAI
6. Backlog e historias de usuario
7. Costos estimados
8. Pruebas curl

---

8. Pruebas con Curl

1. Obtener JWT

```
curl -X POST https://<cognito-domain>/oauth2/token \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=password&client_id=<id>&username=<user>&password=<pass>"
```

2. Crear un schedule

```
curl -X POST https://<api>/prod/schedule \
  -H "Authorization: Bearer <JWT>" \
  -d '{"frequency":"rate(1 hour)","gadgetType":"Rocket Shoes"}'
```

3. Consultar órdenes generadas

```
curl -H "Authorization: Bearer <JWT>" https://<api>/prod/orders
```

4. Cancelar schedule

```
curl -X DELETE https://<api>/prod/schedule/<id> \
  -H "Authorization: Bearer <JWT>"
```

---

9. Historias de Usuario

HU-01 – Crear programación de órdenes

Como gerente de operaciones
Quiero crear un schedule automático
Para generar órdenes de compra sin intervención manual.

HU-02 – Consultar órdenes generadas

Como usuario autenticado
Quiero consultar las órdenes generadas por el scheduler
Para validar su correcta ejecución.

HU-03 – Cancelar un schedule

Como usuario autorizado
Quiero pausar o cancelar un schedule
Para evitar órdenes innecesarias.

---

10. Estimación de Costos (Guía)

Considerar:

- EventBridge Scheduler
- Lambda (administración y ejecución)
- DynamoDB (lecturas/escrituras)
- API Gateway
- KMS
- CodePipeline / CodeBuild
- VPC Endpoints

---

11. Control de Cambios

Versión	Fecha	Cambios	Autor
1.0	2025	Primera versión formal del Documento de Célula 2	Arquitecto del Proyecto

---