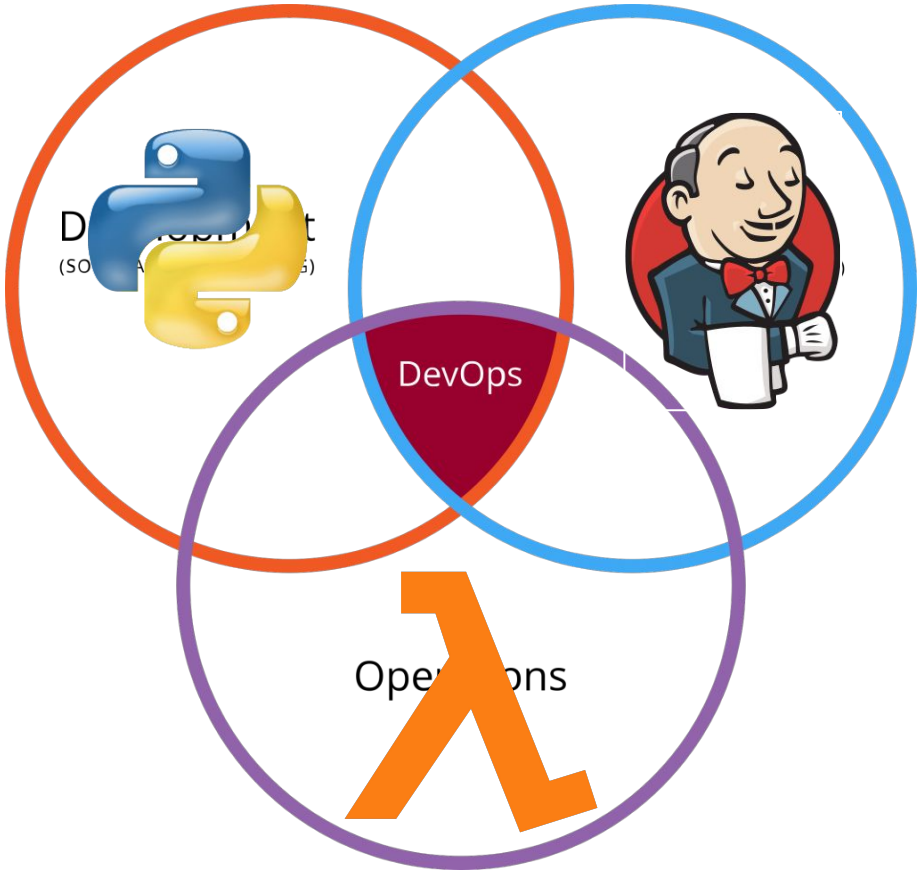
AWS Lambda: CI con Python

¿Quién soy?

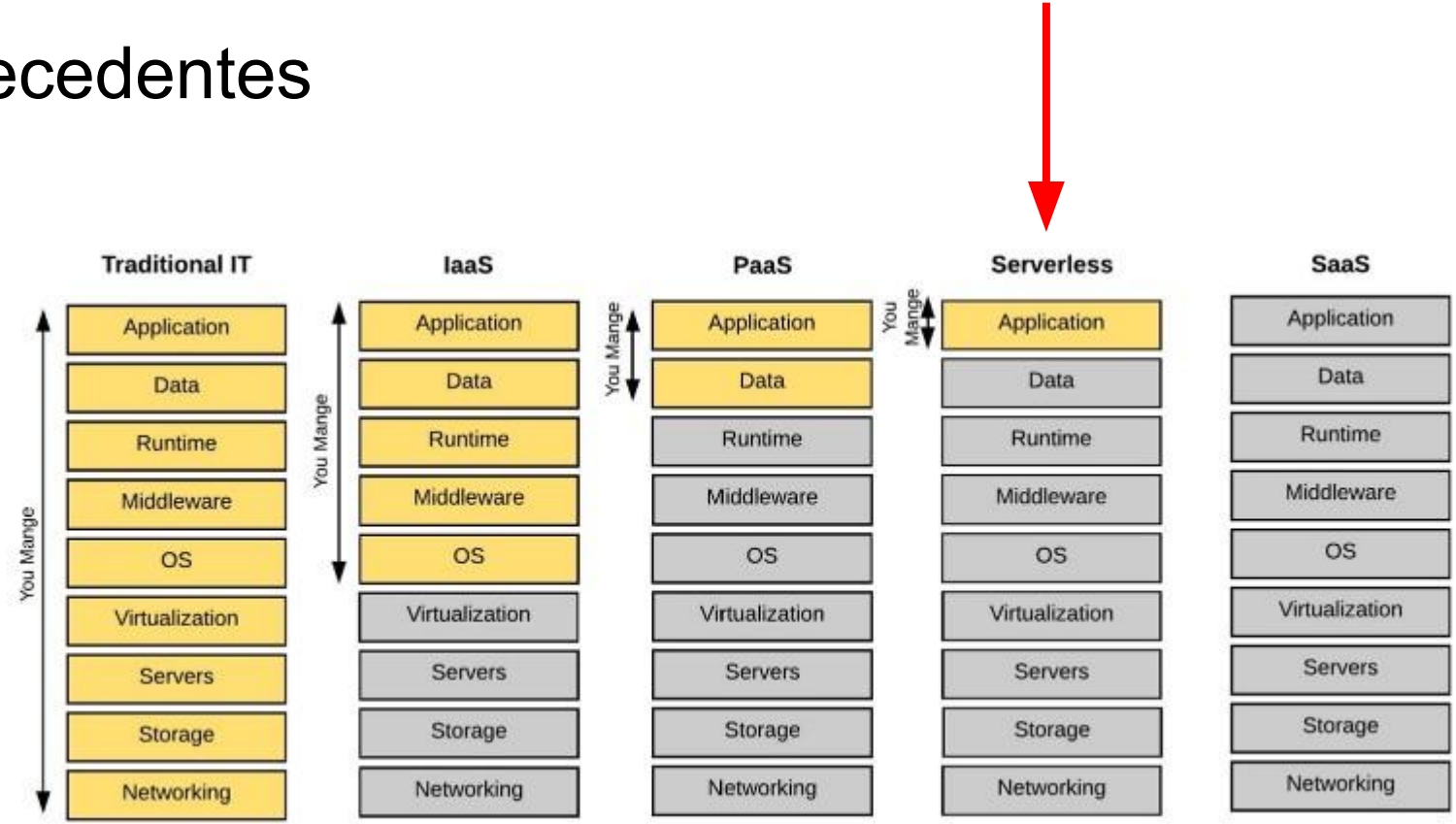
He trabajado con la mayoría de plataformas Cloud: Azure, Google Cloud, AWS, Heroku, OpenShift…

PaaS: Elastik Beanstalk, WebApp, Google App Engine, Azure Websites

IaaS: VPCs/EC2, Compute Engine, Azure


DevOps

Antecedentes



> Un PaaS (Platform as a service) es una plataforma de desarrollo de aplicaciones Cloud. **Facilita** las tareas de automatización del ciclo de vida, configuración, despliegue y escalado de software para que los equipos se centren únicamente en programar las cosas que realmente aportan valor al negocio.
>
> José Ignacio Herranz Roldán

La mayoría de PaaS ofrecen una infraestructura que permite modificarse a través de una GUI o panel web. No hacen “magia” ante pruebas de esfuerzo reales.

Esta definición es “Como se venden los PaaS”, y es la definición REAL de un Serverless

FaaS: Características

Completa abstracción del servidor

Precios en base al consumo y ejecuciones

Servicios basados en eventos e instantáneamente escalables

FaaS: Tipos

Tareas programadas o trabajos

Procesar una solicitud web

Mensajes de cola de proceso

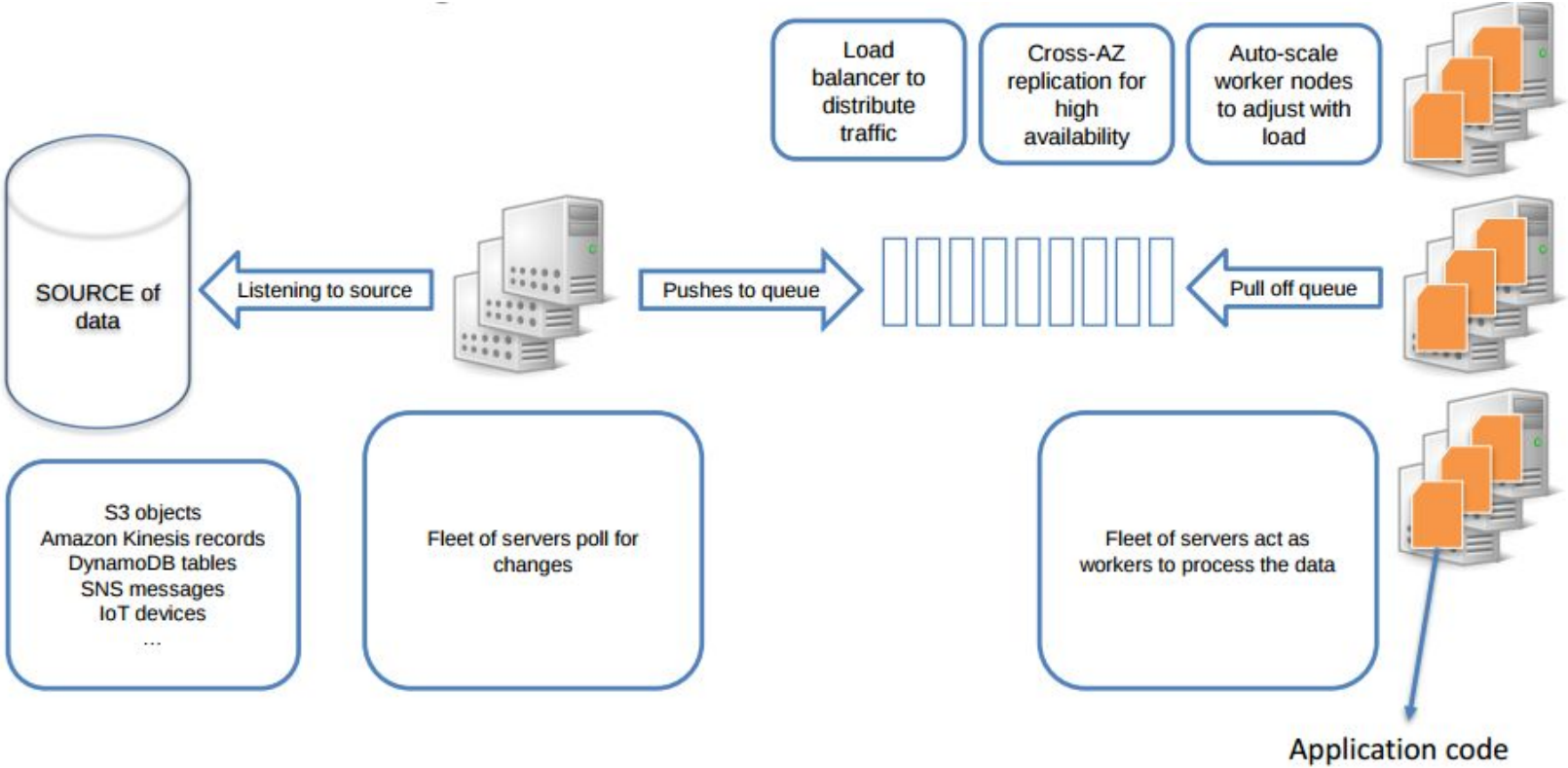
Ejecutar manualmente

¿Qué es AWS Lambda?

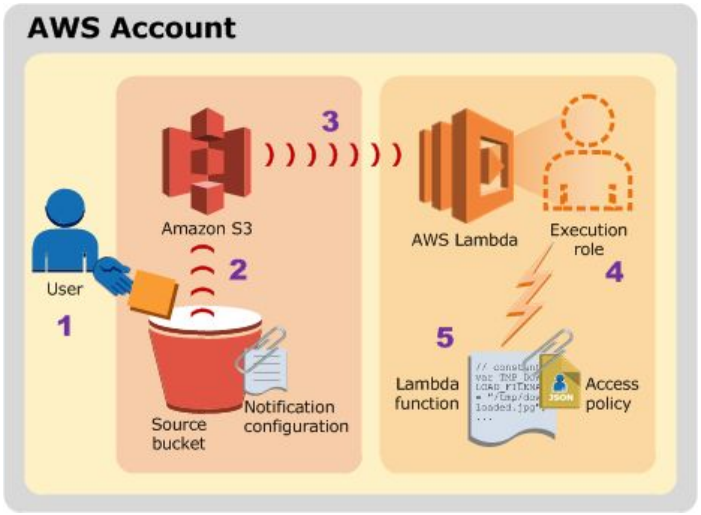
AWS Lambda es un servicio de informática sin servidores que ejecuta el código como respuesta a eventos y administra automáticamente los recursos informáticos subyacentes

Definición AWS

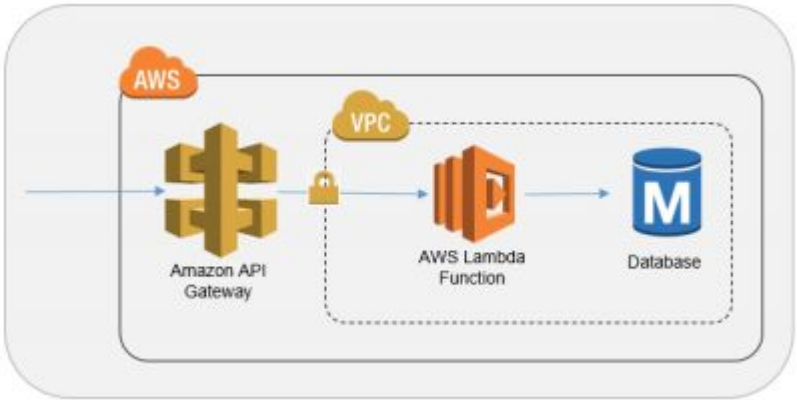
Funcionamiento

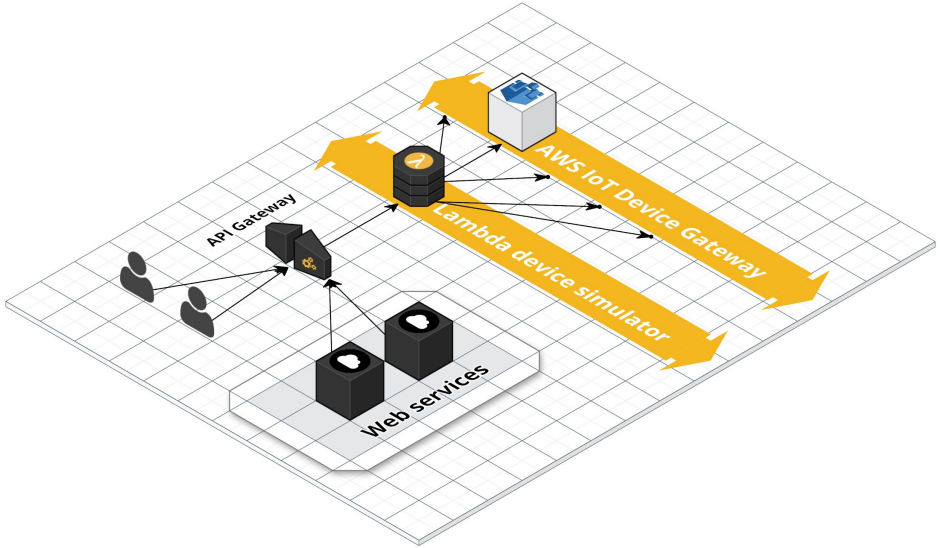


Casos de uso


AWS Account
3
Amazon S3
AWS Lambda
Execution role
User
2
4
1
5
Lambda function
Access policy
Source bucket
Notification configuration

Casos de uso


AWS
VPC
Amazon API Gateway
AWS Lambda Function
Database


API Gateway
AWS IoT Device Gateway
Lambda device simulator
Web services

Otros FaaS

Cloud Functions: https://cloud.google.com/functions/

Iron.io: https://www.iron.io/

Azure Functions: https://azure.microsoft.com/en-us/services/functions/

Webtask: https://webtask.io/

Ventajas

- Como FaaS, olvidarte de la infraestructura, autoescalado,
- Buen enfoque a microservicios con AWS API Gateway.
- **Tareas periódicas con CloudWatch**
- Bajo coste

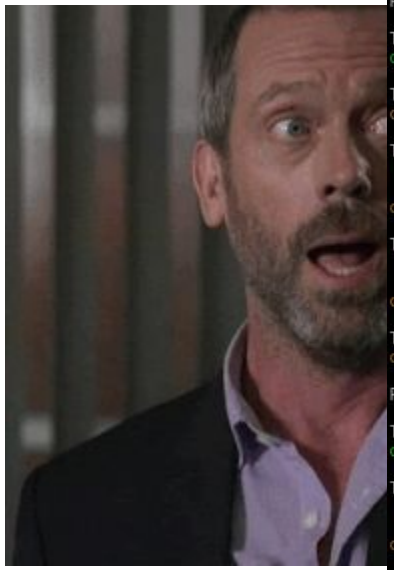
Contras

- Pensado para funcionar con los servicios de AWS
- No está bien pensado para el CI
- Versiones solo por Cliente y API
- Limitación para librerías (“Not all runtimes are available on the Public Amazon Linux AMI version or its yum repositories. You may need to download and install them manually from their respective public sites.” AWS Support team)
- Triggers no configurables desde la API de Lambda diréctamente

Testing

Unitarias: https://github.com/spulec/moto

E2E, Smoke Tests….



```
PLAY [Install basic software] ***************************************************

TASK [Gathering Facts] **********************************************************
ok: [default]

TASK [install Git] **************************************************************
changed: [default] => (item=['git'])

PLAY [Install python dependencies] **********************************************

TASK [Gathering Facts] **********************************************************
ok: [default]

TASK [Download pip] *************************************************************
changed: [default]

TASK [Install pip] **************************************************************
changed: [default]

TASK [Install AWS Cli & virtualenv] *********************************************
changed: [default] => (item=virtualenv)
changed: [default] => (item=awscli)

PLAY [Install jenkins dependencies] *********************************************

TASK [Gathering Facts] **********************************************************
ok: [default]

TASK [install jdk rpm] **********************************************************
changed: [default] => (item=['java-1.8.0-openjdk', 'wget'])

TASK [add Jenkins to yum] *******************************************************
 [WARNING]: Consider using get_url or uri module rather than running wget

changed: [default]

TASK [add jenkins key] **********************************************************
 [WARNING]: Consider using yum, dnf or zypper module rather than running rpm

changed: [default]

TASK [install jenkins rpm] ******************************************************
changed: [default]

PLAY [start Jenkins] ************************************************************

TASK [Gathering Facts] **********************************************************
ok: [default]

TASK [start Jenkins] ************************************************************
 [WARNING]: Consider using service module rather than running service

changed: [default]

PLAY RECAP **********************************************************************
default                    : ok=13   changed=9    unreachable=0    failed=0
```

¿Cómo usar AWS Lambda con Python?

Ejemplos de interés

API Gateway+ Lambda Framework: https://github.com/awslabs/chalice

Lambda toolkit para cualquier Runtime: https://github.com/garnaat/kappa

Lambda toolkit para múltiples funciones: https://github.com/avara1986/ardy

Chatbot con Lambda: https://github.com/awslabs/aws-serverless-chatbot-sample

Contacto

https://github.com/avara1986

a.vara.1986@gmail.com