

# **ARQUITECTURAS SERVERLESS EN AWS**

---

**JUAN MARQUINHO VILCA CASTRO**

# ¿QUÉ ES SERVERLESS?

---

- Para responder esta pregunta, es mejor comenzar por decir que no es serverless.

# ¿QUÉ ES SERVERLESS?

---

- No administrar servidores o procesos que corren en ellos.
- Autoescalable y auto provisionable basado en la carga.
- Costo basado en el uso.
- Capacidades de performance no están definidas en función al tamaño o número de hosts.
- Es altamente disponible (Resiliency).

# ¿QUÉ NO ES SERVERLESS?

---

- No significa que los servidores ya no formen parte de la infraestructura de la aplicación.
- No es simplemente FAAS (AWS Lambda).

# ¿QUÉ ES SERVERLESS?

---

- Cualquier producto de software que abstraee el uso de servidores.
- Lo más importante es: “Gastar menos tiempo en administrar y más tiempo en construir cosas”.

# TODO ES ABSTRACCIÓN

---

**Cloud code?**

**Application code**

**Assembly code**

**Machine code**

# TODO ES ABSTRACCIÓN

---

**Platform as a service**

**Serverless (FAAS, BAAS, Messaging)**

**CoAAS**

**Managed DBs**

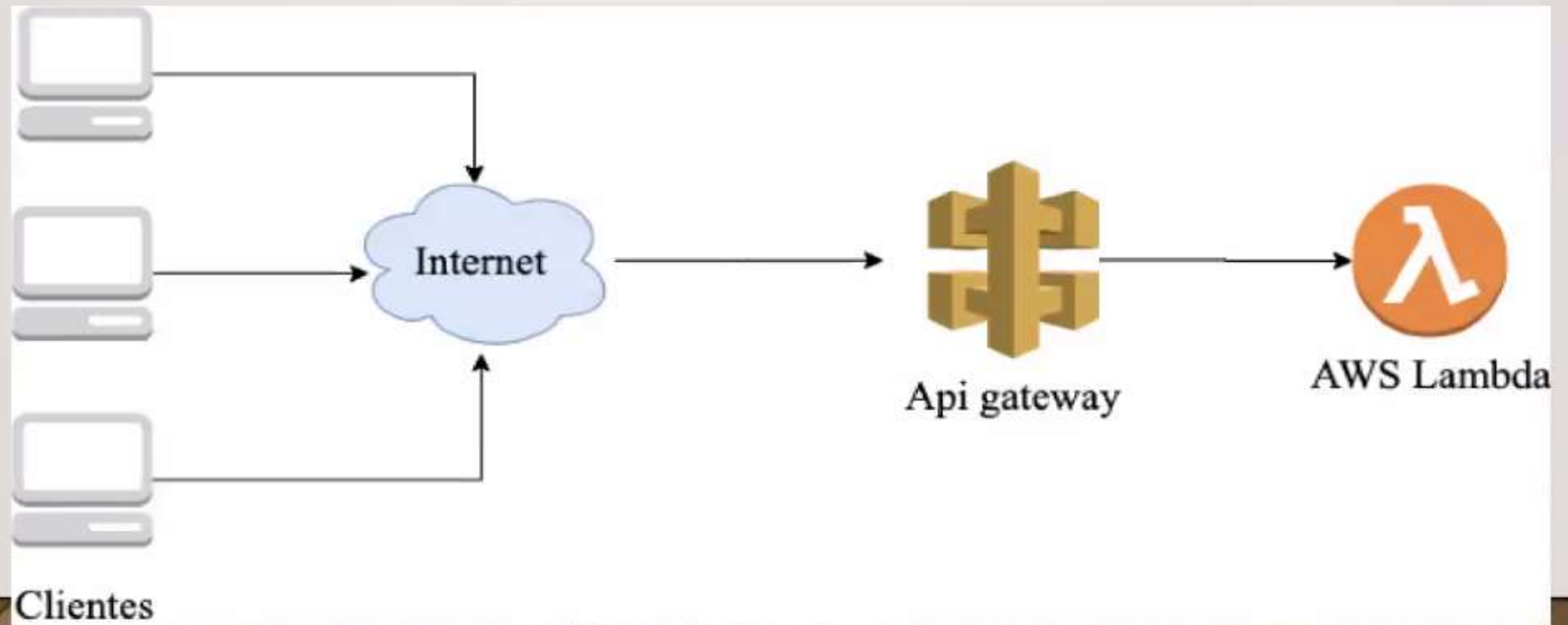
**Virtualised infrastructure (Compute, Network, Storage)**

**Physical Infrastructure setup**



# FAAS (FUNCTION AS A SERVICE)

Usando FAAS:





# BAAS (BACKEND AS A SERVICE)

---

- AWS -> DynamoDB, RDS
  - Pagar por lo que se usa.
  - Escalamiento transparente.
  - Administración muy reducida.
  - Backup simplificado.

# SERVERLESS – BUILDING BLOCKS

---

- FAAS (Lambda, ECS)
- BAAS (DynamoDB, Aurora)
- Storage (S3)
- Messaging (SNS, SQS, Kinesis)

# FAAS (FUNCTION AS A SERVICE)

---

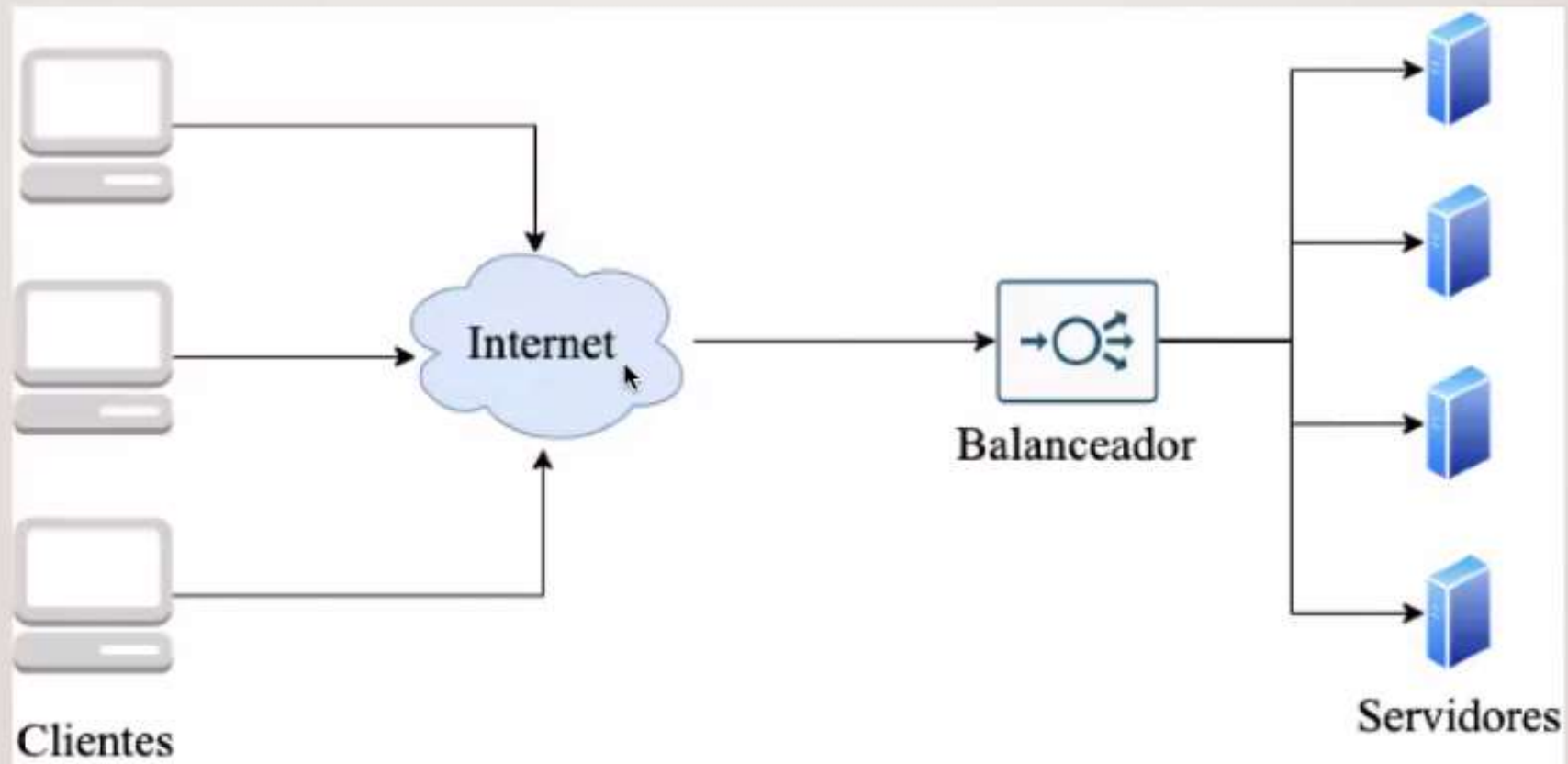
Sin FAAS:

- Configuración de los recursos de computación (Sistema operativo, dependencias, software base).
- Escalar.
- Pagar por los recursos de computo aun si no se están usando.
- Configuración de los recursos de software por cada Host.

# FAAS (FUNCTION AS A SERVICE)

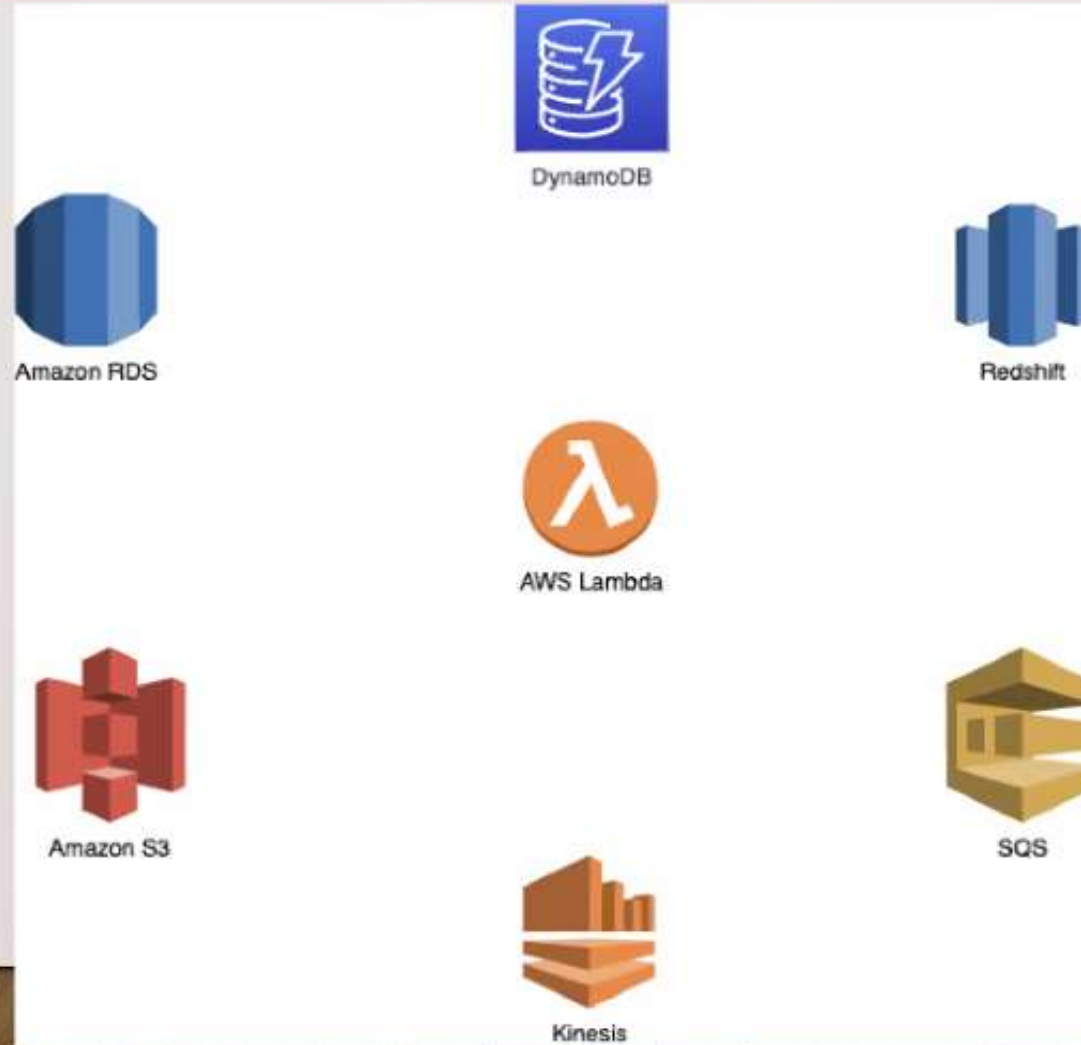
---

Sin FAAS:



# FAAS (FUNCTION AS A SERVICE)

---



# FAAS (FUNCTION AS A SERVICE)

---

- ¿AWS tiene restricciones?
- <https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>
- ¿Solución? Fat Lambdas (Lambdas gordas).

# SERVERLESS CONCERNS

---

- Security
- Local executions
- Troubleshooting
- Performance.
- Deployment.



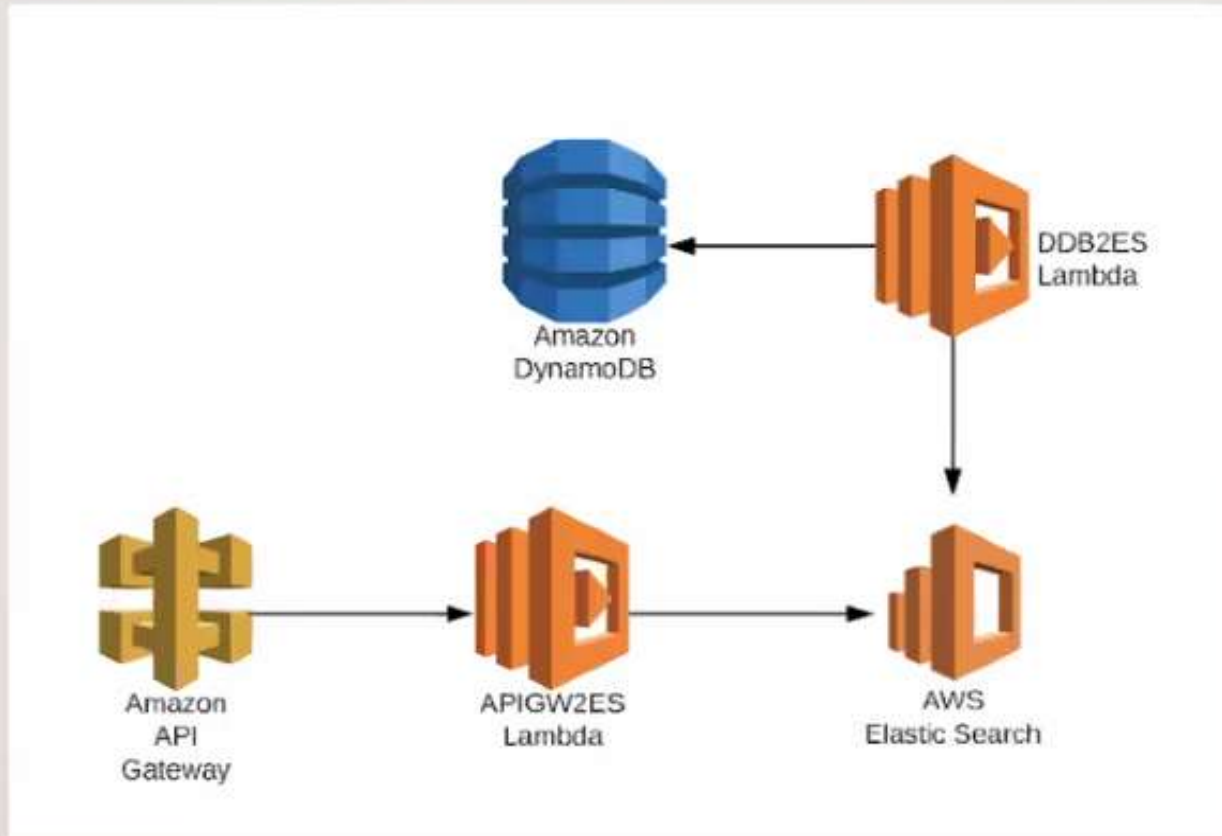
# ARCHITECTURE: **EXAMPLE 1**



<https://dimensionless.in/methods-to-manage-amazon-spark-application-memory-on-amazon-emr/>

# ARCHITECTURE: **EXAMPLE 2**

---



<http://techincalpostings.blogspot.com/2018/04/aws-elasticsearch.html>

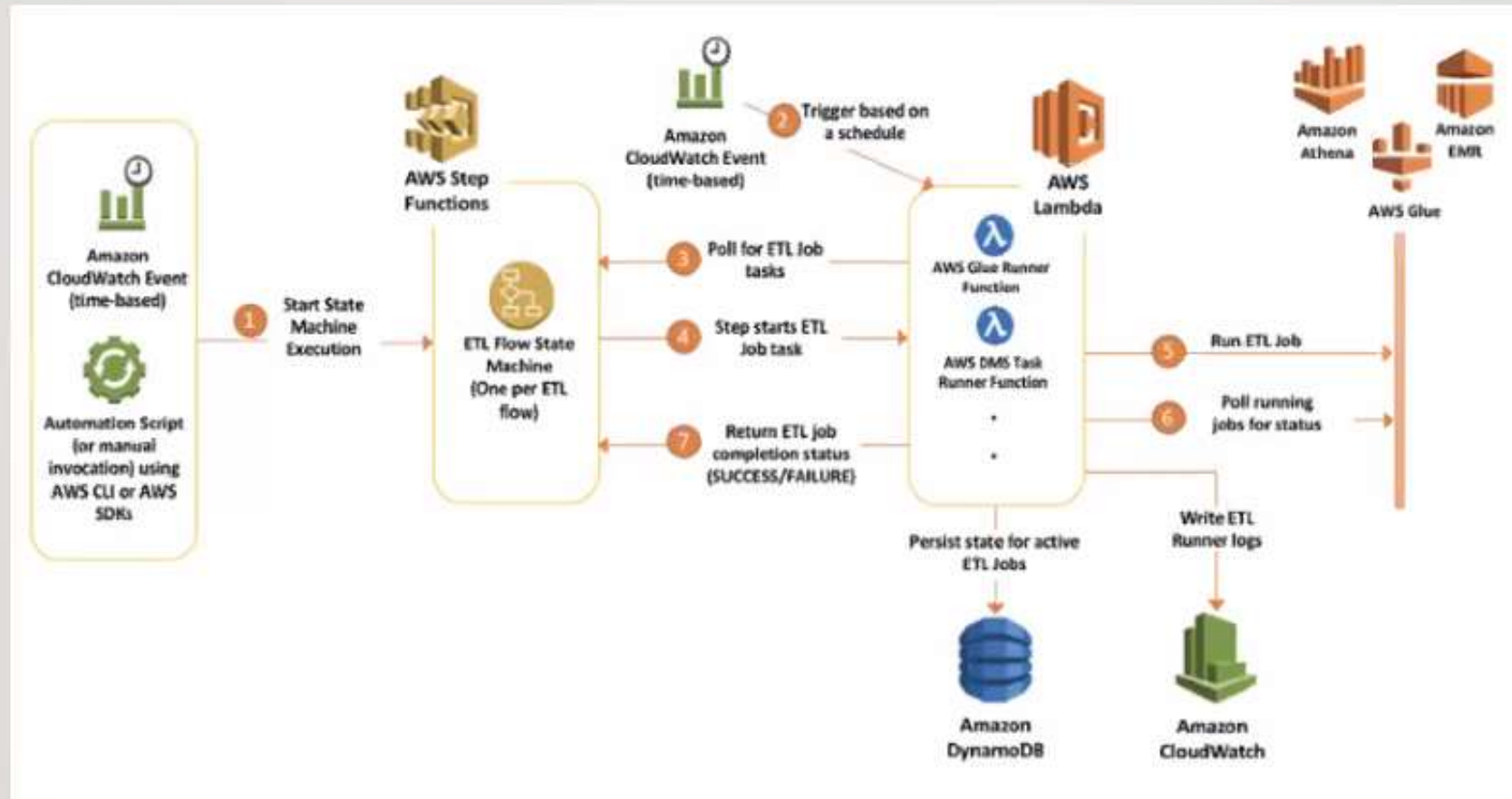
# ARCHITECTURE: **EXAMPLE 3**

---



<https://aws.amazon.com/blogs/big-data/orchestrate-multiple-etl-jobs-using-aws-step-functions-and-aws-lambda/>

# ARCHITECTURE: **EXAMPLE 3**



<https://aws.amazon.com/blogs/big-data/orchestrate-multiple-etl-jobs-using-aws-step-functions-and-aws-lambda/>