

Infraestructura mediante Código via Terraform en Amazon Web Services



Universidad
Internacional
de Valencia

Miguel Angel Gagliardo
Grado en Ingeniería Informática

17 de Julio de 2025

De:
 Planeta Formación y Universidades

01

Introducción

De lo Físico a lo Declarativo

Entornos Físicos > Virtualizados > Cloud

Escalabilidad + Agilidad + Reducción de Costos

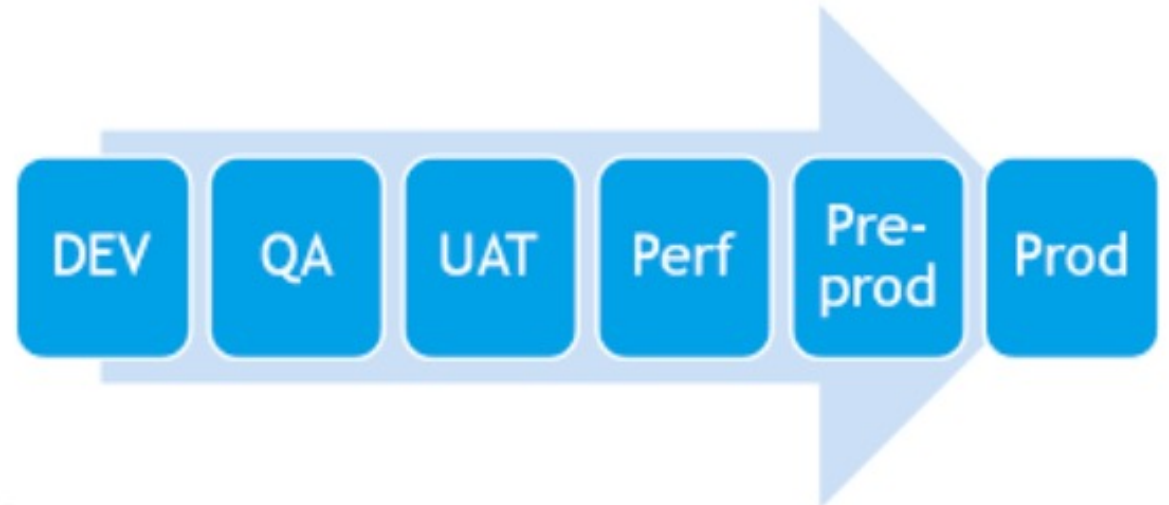
Persisten cuellos de botella por gestión manual

Motivaciones

IaC esencial en entornos Ágiles

Replicar ambientes

Demanda en Mercado IT

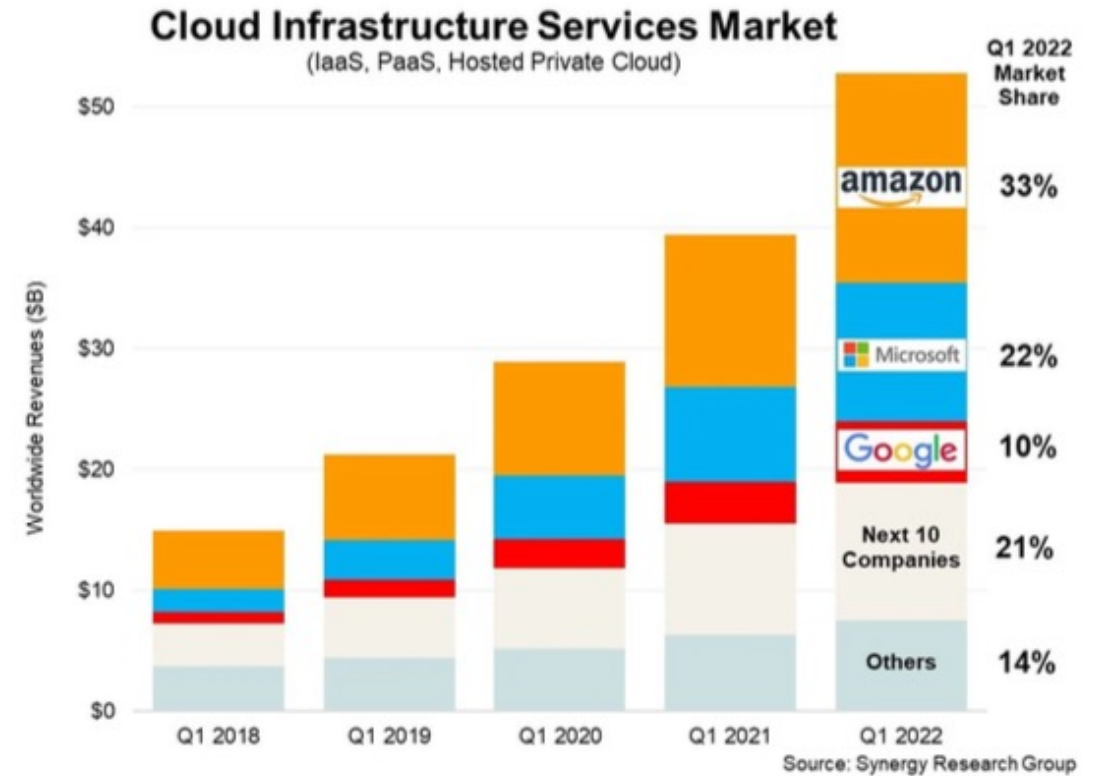


* Fuente: <https://docs.cloudbees.com/docs/cloudbees-cd/latest/plan/server-topology>

02

Tecnologías

Amazon Web Services



Fuente: https://www.theregister.com/2022/05/02/cloud_market_share_q1_2022/

Terraform



Características	Terraform	Pulumi	AWS CDK
Soporte Multi-Cloud	✓	✓	✗
Manejo de Estado	✓	✓	✓
Licencia OpenSource	✓	✓	✗
No require código	✓	✗	✗
Gratuito	✓	✗	✓
Extensible	✓	✗	✗
Lenguaje	HCL (DSL)	Multi-Lang	Multi-Lang
Curva de Aprendizaje	Baja	Alta	Alta

Contenedores



Características	Máquinas Virtuales	Contenedores
Peso	Pesado	Ligero
Uso de recursos OS	Independiente	Compartido
Tiempo de Inicio	Lento	Rápido
Portabilidad	Baja	Alta
Uso de memoria	Alto	Bajo
Instancias por Host	10-100	100-1000
Seguridad	Alta	Moderada

03

Metodología

Project Management

Crear codigo app en python #1

mgagliardo opened on Apr 8 · edited by mgagliardo

En preparación del proyecto de trabajo de fin de grado, se requiere una app simple de tipo de "microblog" para la demostración del trabajo, los casos de uso son:

- Alta, Baja y Modificación (ABM) de usuarios utilizando username/email/password
- Crear un "post" en el blog
- Follow/Unfollow de otro usuario

Adicionalmente (la app tiene que poder funcionar SIN dicha infra de igual manera):

- Background tasks
- Búsquedas indexadas (usando algún motor tipo ElasticSearch)
- Envío de mails

Se va a requerir generar:

- Clases
- Data models
- DB
- APIs
- UI

Las tecnologías a utilizar son:

- Python
- Algún framework de API como:
 - Flask
 - FastAPI
- Algún motor de RDBMS (MySQL o PostgreSQL)
- Algún ORM:
 - SQLAlchemy
 - Flask-SQLAlchemy
 - PycoPG
- Frontend, pueden ser:
 - React
 - O algún SSR como el que viene embebido en flask

mgagliardo added **feature** **Sprint 1** on Apr 12

mgagliardo mentioned this on Apr 12

mgagliardo self-assigned this on Apr 12

mgagliardo mentioned this on Apr 12

Feat: Adding microblog python app #11

mgagliardo on Apr 12

Resolviendo, el código se puede encontrar en: <https://github.com/mgagliardo/viu-84giin-tfg/tree/main/microblog>

[FINALIZADO] Sprint 2 - 22/04/2025 - 05/05/2025

Filter by keyword or by field

Backlog 4

This item hasn't been started

- viu-84giin-tfg #21
Presentación de TFG
- viu-84giin-tfg #8
Obtener cuenta de AWS para TFG
- viu-84giin-tfg #3
Crear codigo de terraform para deployment
- viu-84giin-tfg #4
Trabajo en memoria y puntos importantes

Todo 2

- viu-84giin-tfg #16
Checkpoint meeting #2 con tutor
- viu-84giin-tfg #2
Dockerizar app

In Progress 1

This is actively being worked on

- viu-84giin-tfg #1
Crear codigo app en python

Done 6

This has been completed

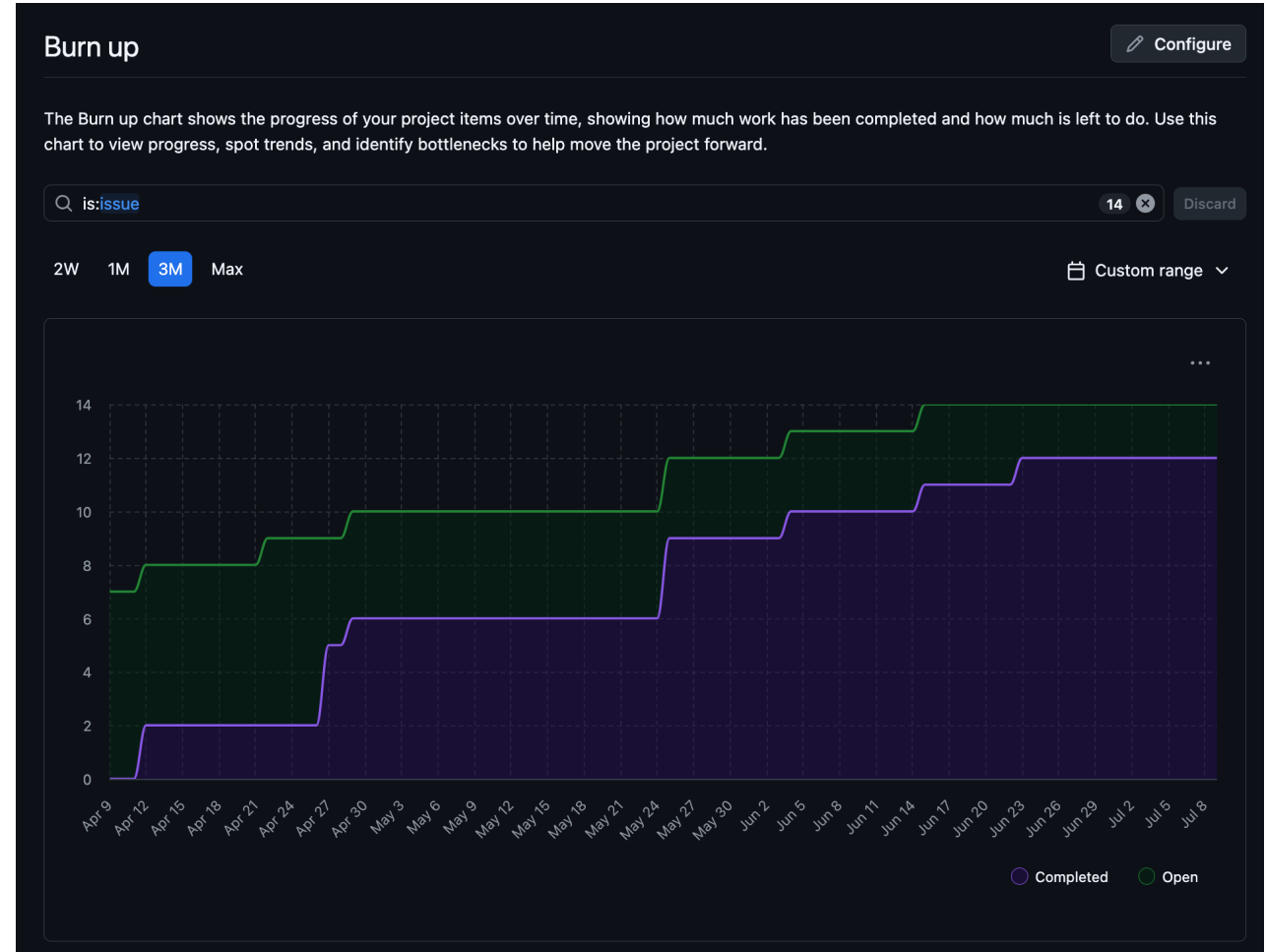
- viu-84giin-tfg #16
Checkpoint meeting #2 con tutor
- viu-84giin-tfg #8
Obtener cuenta de AWS para TFG
- viu-84giin-tfg #2
Dockerizar app
- viu-84giin-tfg #1
Crear codigo app en python
- viu-84giin-tfg #3
Crear codigo de terraform para deployment
- viu-84giin-tfg #7
Dominio de la app. Route53 vs Cloudflare

Link: <https://github.com/mgagliardo/viu-84giin-tfg>

Seguimiento

Burnup charts

Feedback Continuo



04

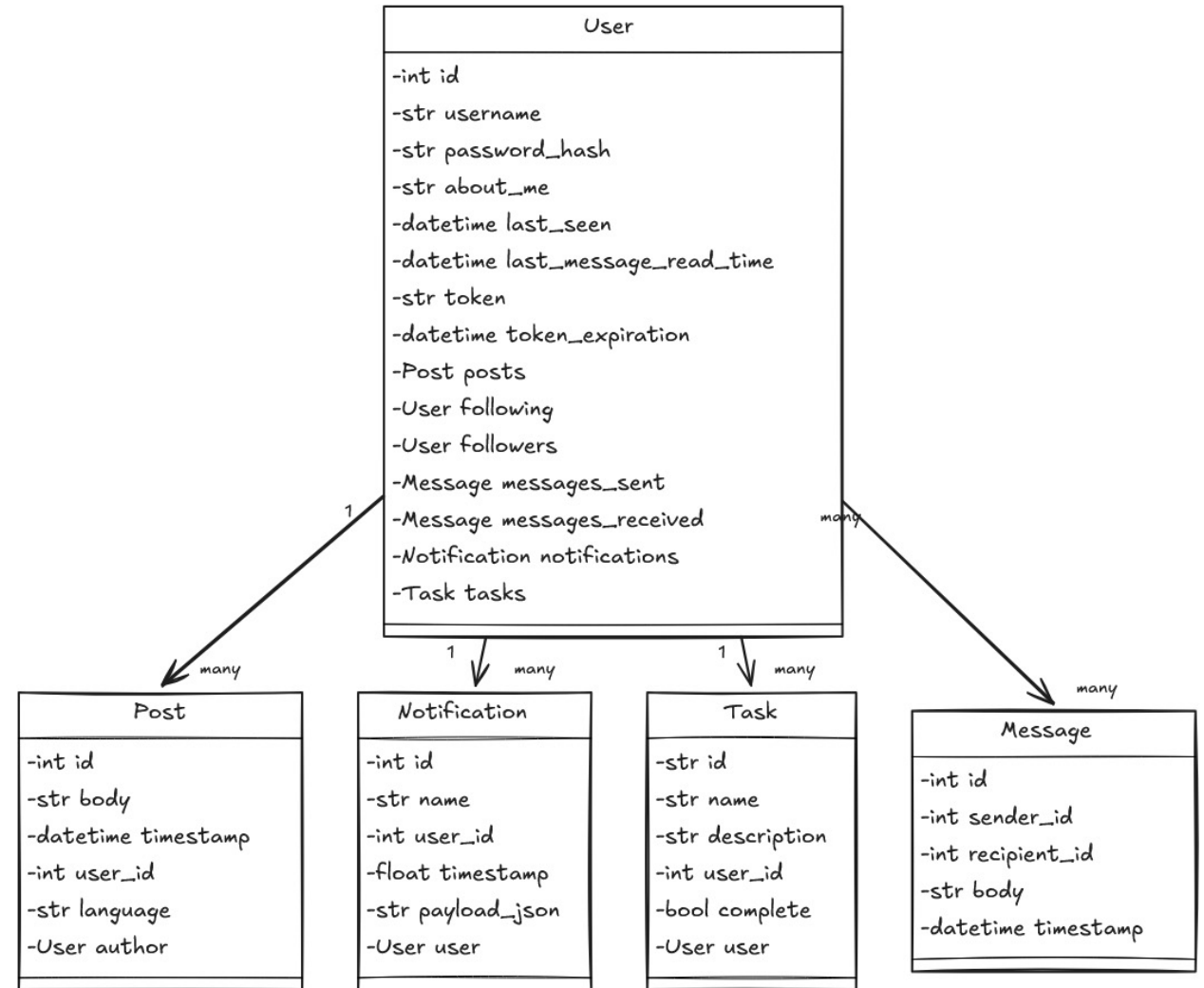
Desarrollo

Aplicación “Microblog”

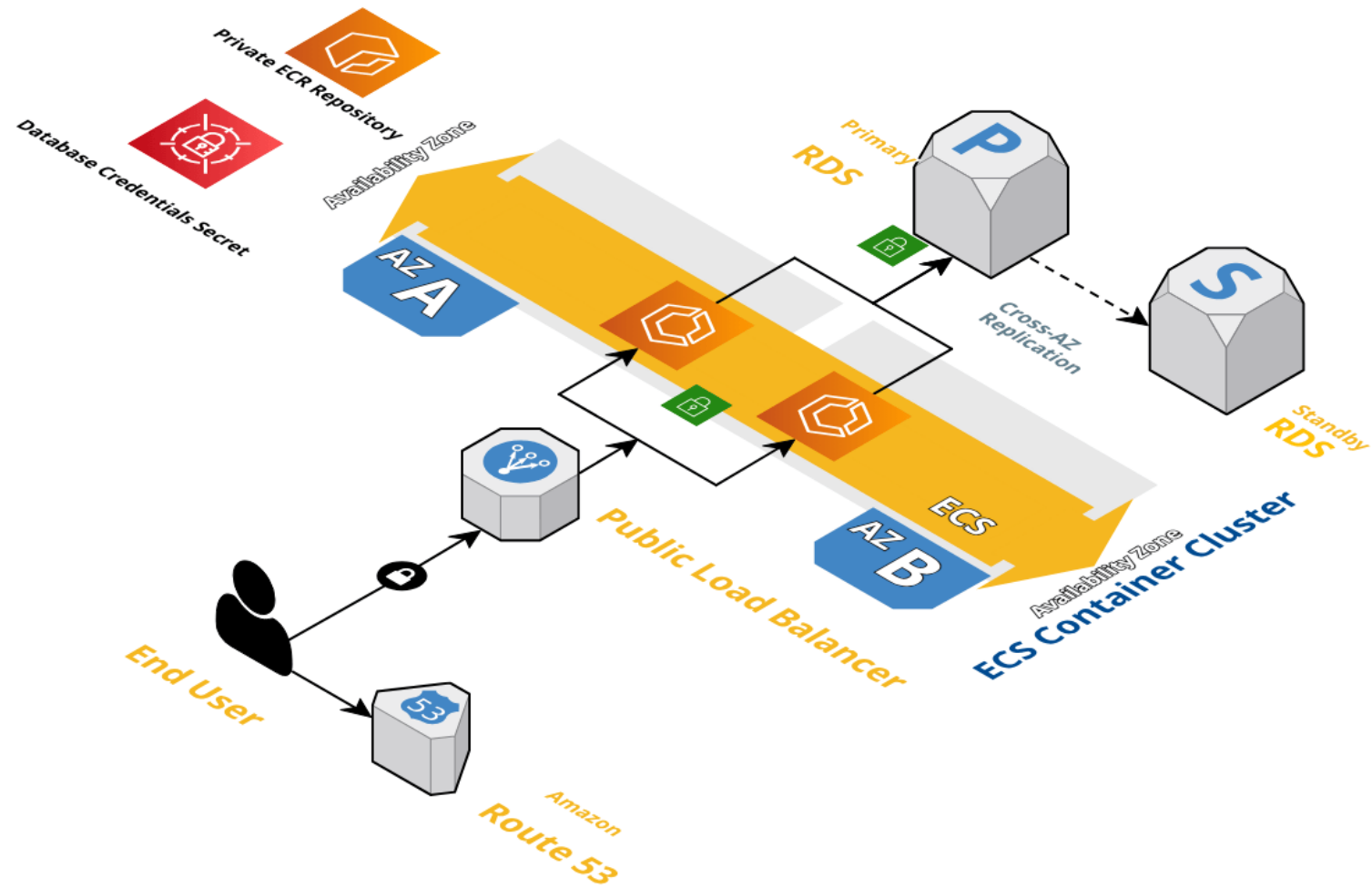
Python Webapp

PostgreSQL DB

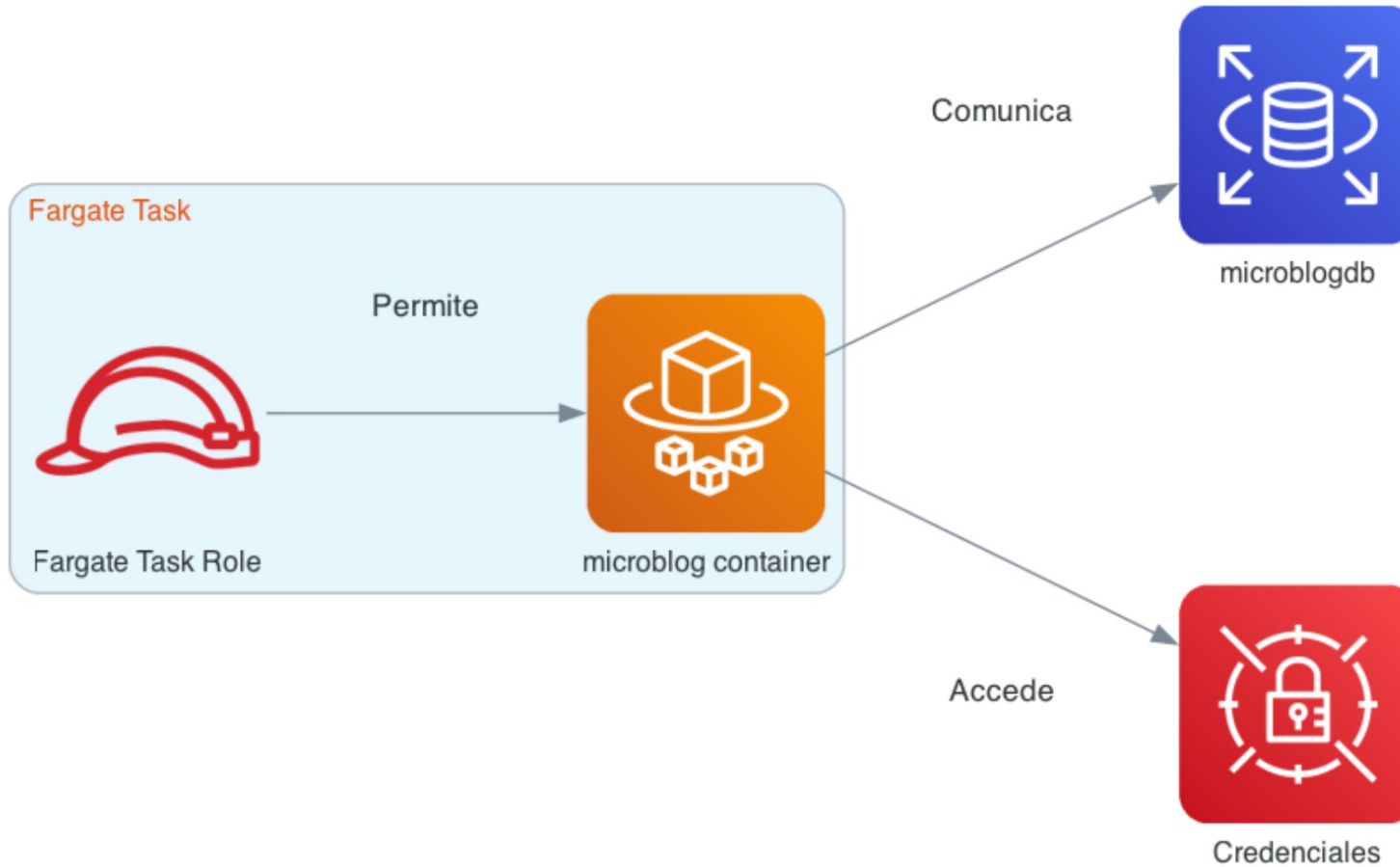
Rapid Development



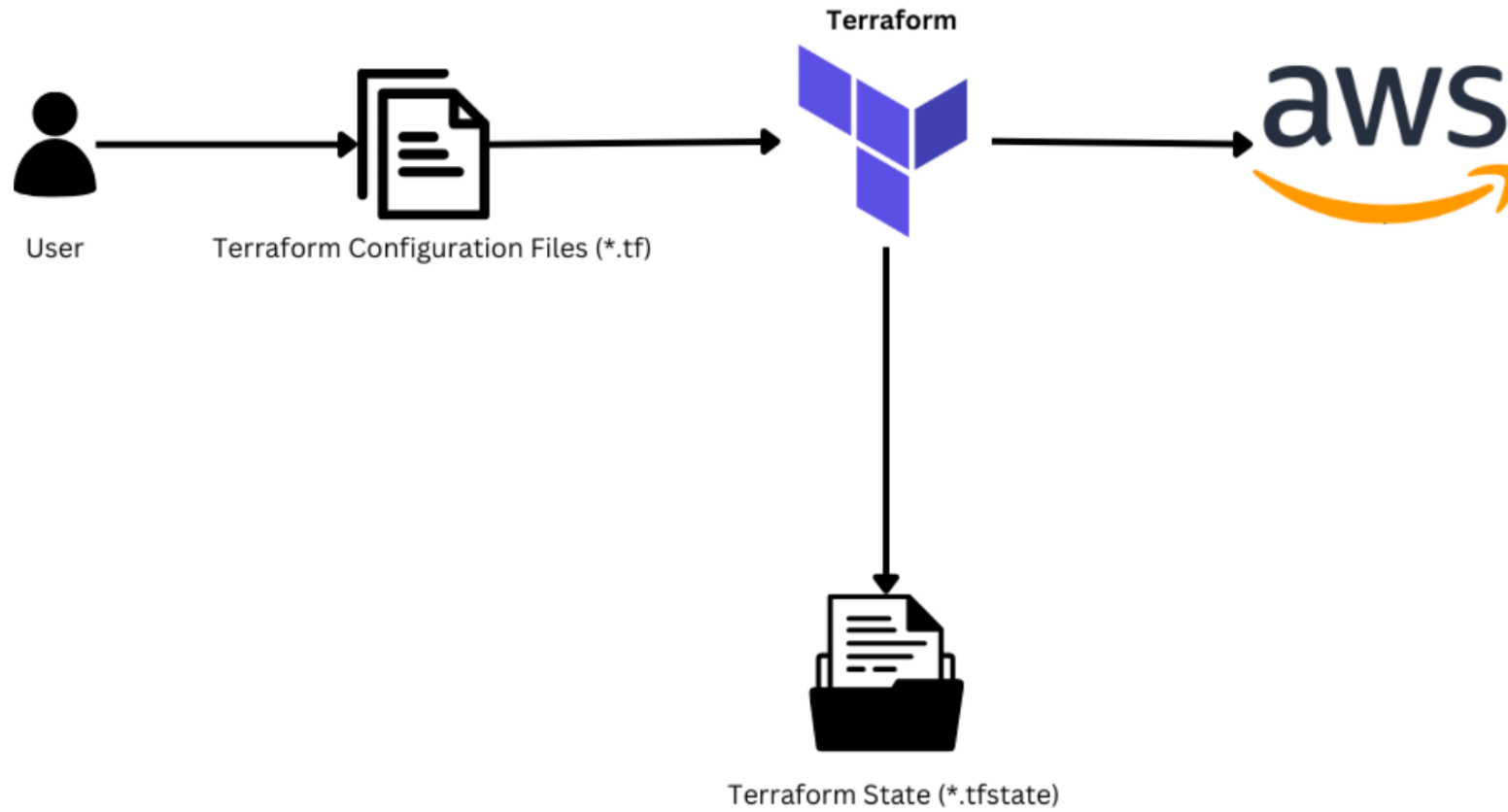
Infraestructura



Autenticación Dinámica



Terraform y el manejo de Estado



05

Resultados

Ventajas Obtenidas

Código Reutilizable y Auditable

Entorno de Desarrollo Ágil

Sencillez para replicar entornos

Mejor colaboración en equipos de trabajo

Despliegue Manual vs IaC

Características	Despliegue Manual	Despliegue con IaC
Tiempo de Despliegue	210 minutos	15 minutos
Tiempo de Destrucción	90 minutos	17 minutos
Replicable	No	Si
Factores de Error Humano	Si	No
Uso de memoria	Alto	Bajo
Requiere Conocimientos Previos de AWS	Si	No
Documentación Auto-generada	No	Sí
Auditable	No	Sí
Versionable	No	Sí

Árbol de Dependencia



06

Conclusiones

Impacto en Operaciones

Automatización y Eficiencia Operativa

Consistencia y Reducción de Errores

Escalabilidad, Auditabilidad, Mantenimiento Simplificado

Áreas de Mejora y Trabajo Futuro

CI/CD

Pruebas automatizadas

Smart Monitoring

Auto escalado dinámico

Reflexión Final



DEMO

GRACIAS!



Universidad
Internacional
de Valencia