



SISTEMAS OPERATIVOS 1

CLASE 6

Leonel Aguilar

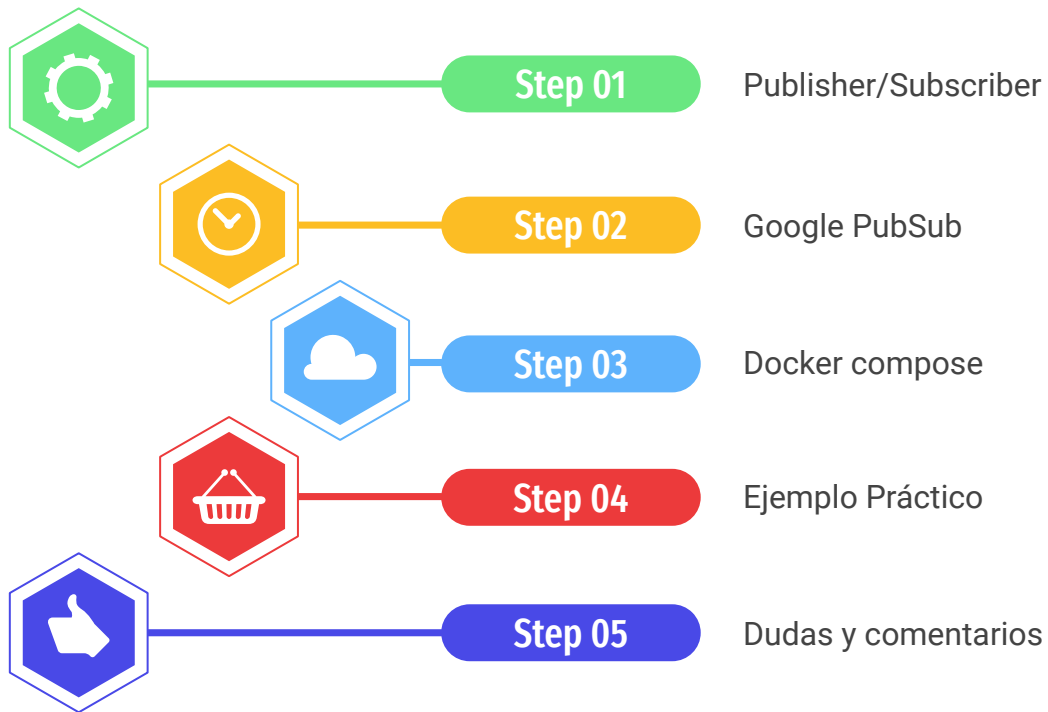
aux.leoagUILar@gmail.com

<https://github.com/leoagUILar97/so1-course/tree/main>

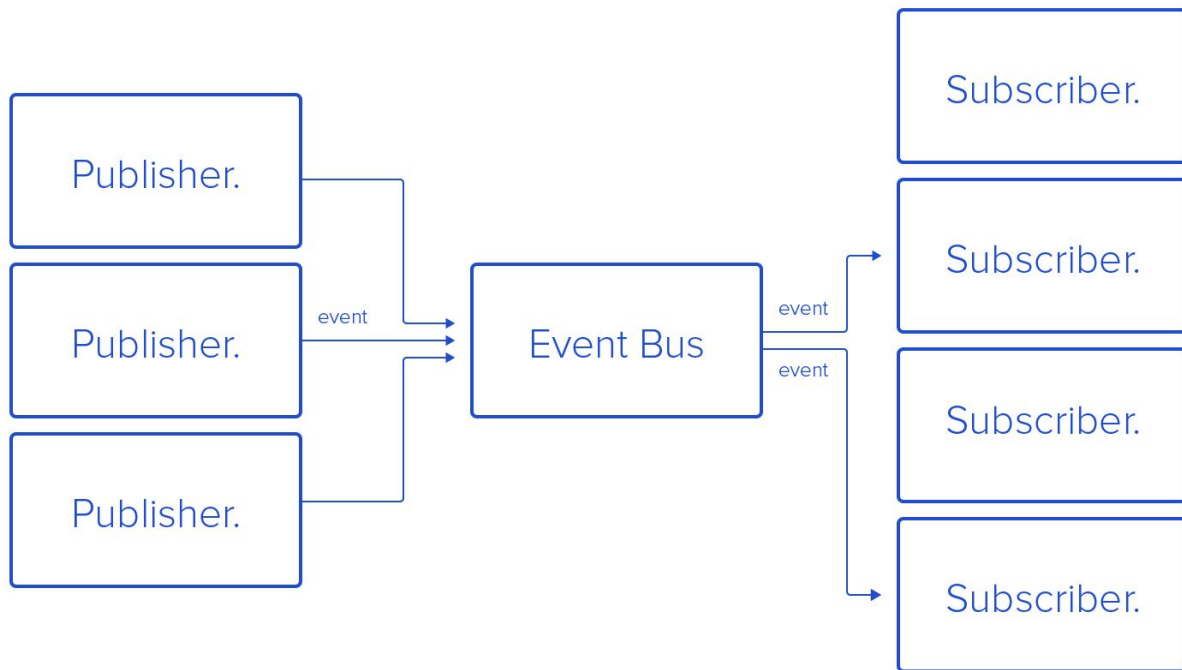
PUBLISH SUBSCRIBE

Google PubSub / Docker Compose

Agenda de hoy

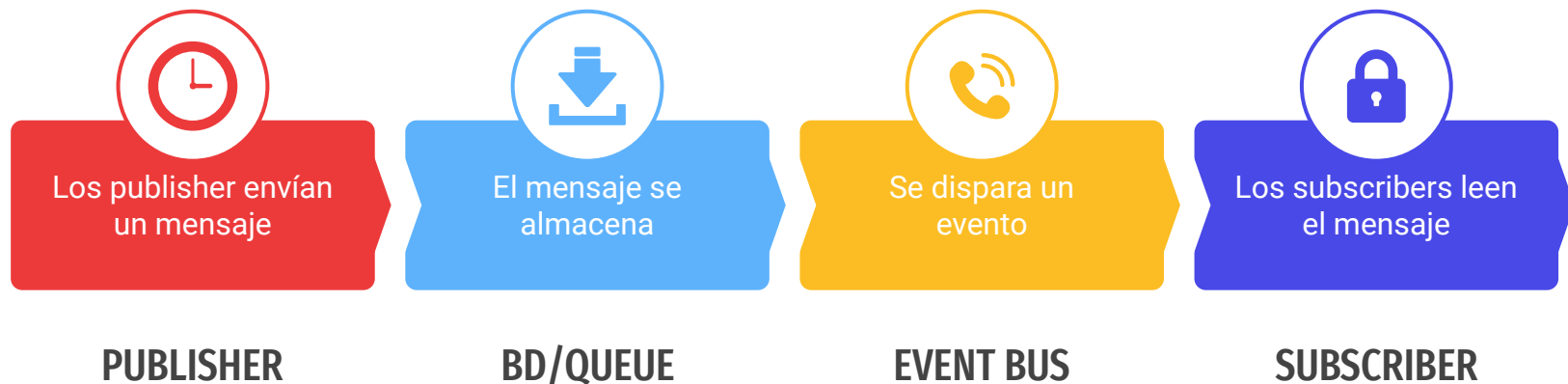


PUBLISHER SUBSCRIBER PATTERN



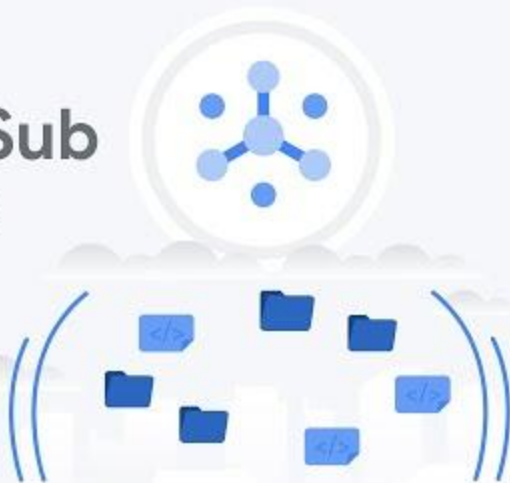
PATRÓN PUBLISHER-SUBSCRIBER PUBSUB

Es un patrón de envío y recepción de mensajería. Los emisores son llamados publishers. Publican su contenido a un canal de eventos que no depende de los que están escuchando. Los receptores están conectados a este canal y leen los mensajes cuando sea que estén disponibles, sin necesidad de saber quién los publicó.

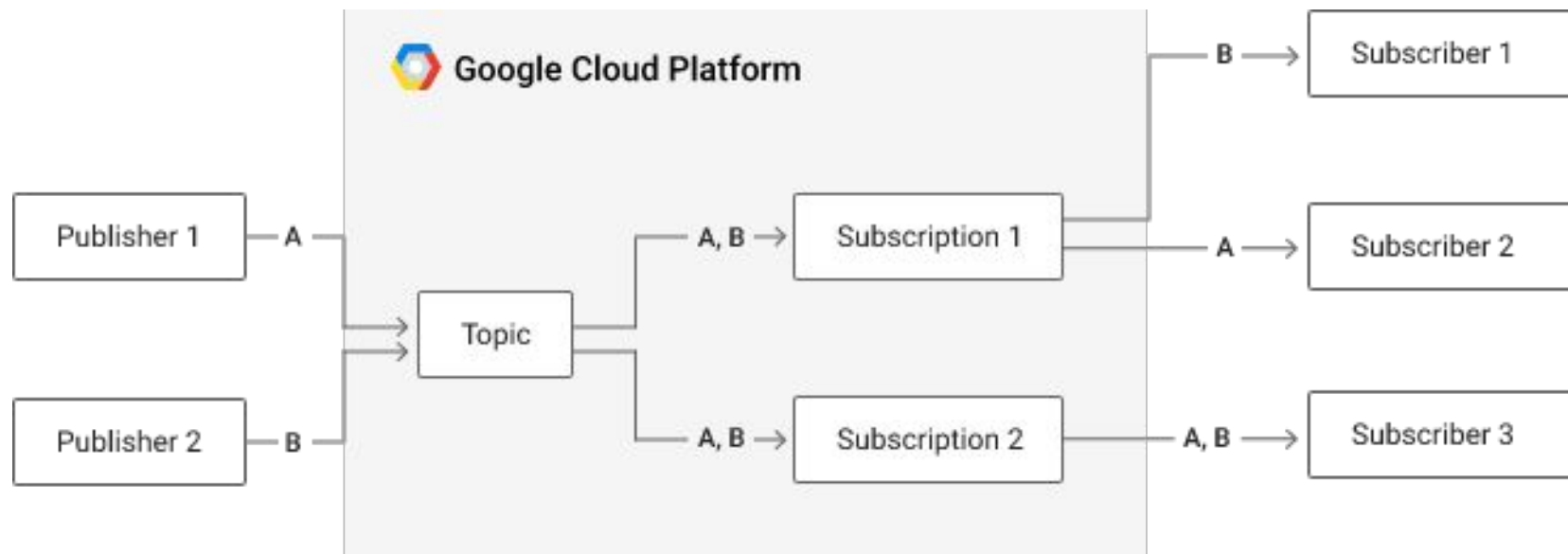


Cloud PubSub in a minute

Cloud Bytes



GOOGLE PUBSUB

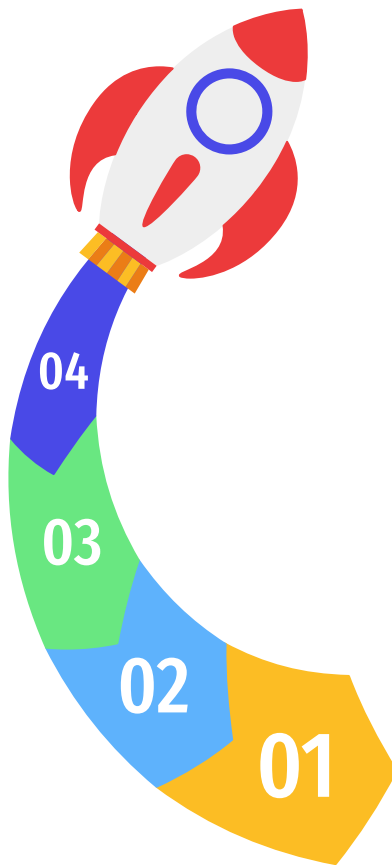




Google PubSub

- **Topic:** Recurso a donde enviaremos los mensajes.
- **Subscription:** Recurso de donde obtendremos los mensajes.
- **Message:** Información que enviaremos (bytes)

Google PubSub



KEY

Obtener una llave .json

SERVICE

Crear un usuario de servicio

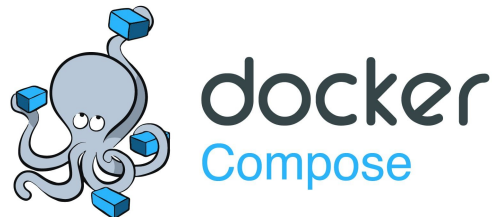
SUBSC.

Crear una subscripción en GCP

TOPIC

Crear un topic en GCP

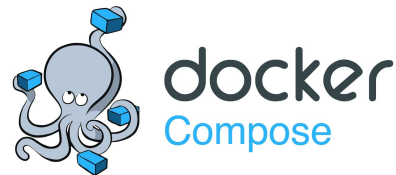
DOCKER COMPOSE



```
version: "3.9"
services:
networks:
volumes:
```

- **Simplificar** la vida.
- Crear distintos contenedores a partir de builds.
- Crear distintos contenedores a partir de imágenes.
- Crear redes de docker.
- Crear volúmenes.
- Conectar todo.

DOCKER COMPOSE



Levantar un servicio de docker

`docker compose up`

Crear un servicio de docker

`docker compose down`

Botar un servicio de docker

`docker compose build`

EJEMPLO PRÁCTICO

Configurar nuestro ambiente

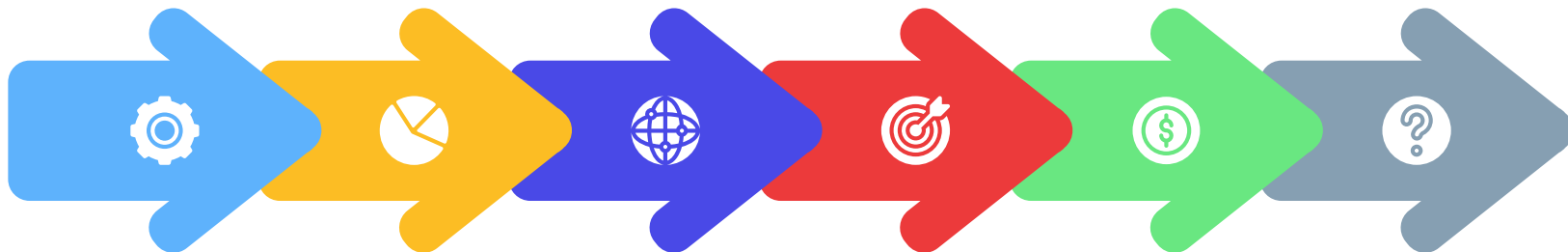
Configurar folders, iniciar modulus, etc...

Subscriber

Creamos nuestro Subscriber utilizando Node.

¡Containerizar!

Creamos los Dockerimage y docker-compose.yml



CONFIGURAR GCP

Configurar topics, subscriptions, etc.

Descargar librerías

Descargamos las librerías necesitadas

Publisher

Programamos nuestro publisher utilizando Go.