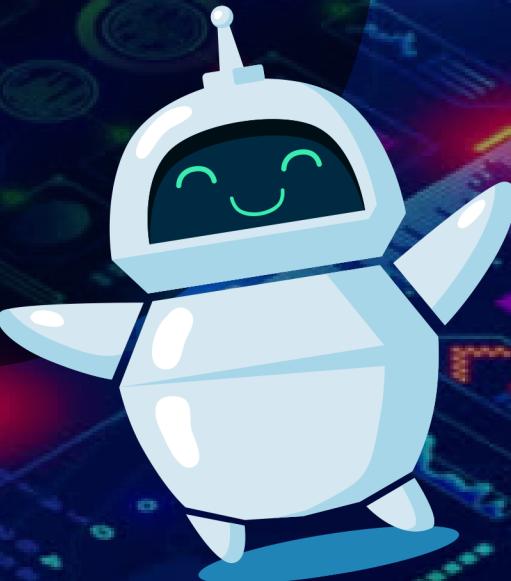




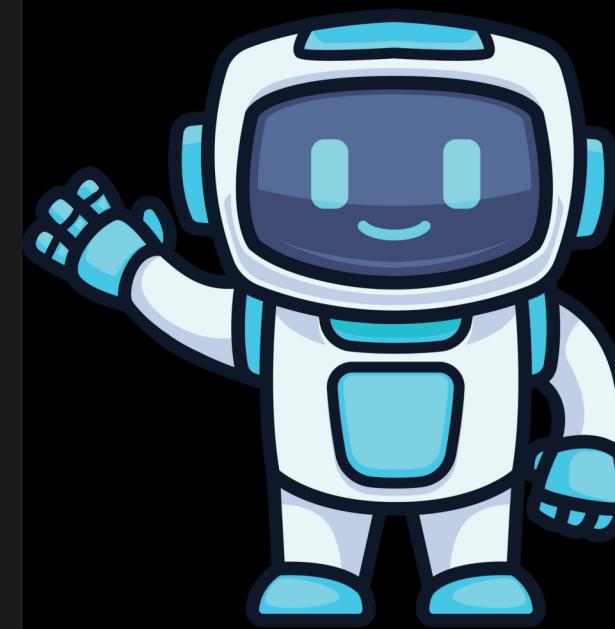
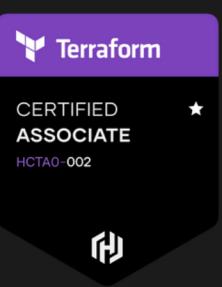
# Sabores de Contenedores en la Nube

## Explorando las Opciones y sus Beneficios



# PRESENTACIÓN

## HASIEL MUÑOZ CLOUD ARCHITECT



Profesional tecnico de redes y comunicaciones, estudiante de ingenieria de sistemas con 4 años de experiencia en el sector tecnologico, entusiasta y miembro de comunidades del mundo tecnologico cloud, devops y seguridad.

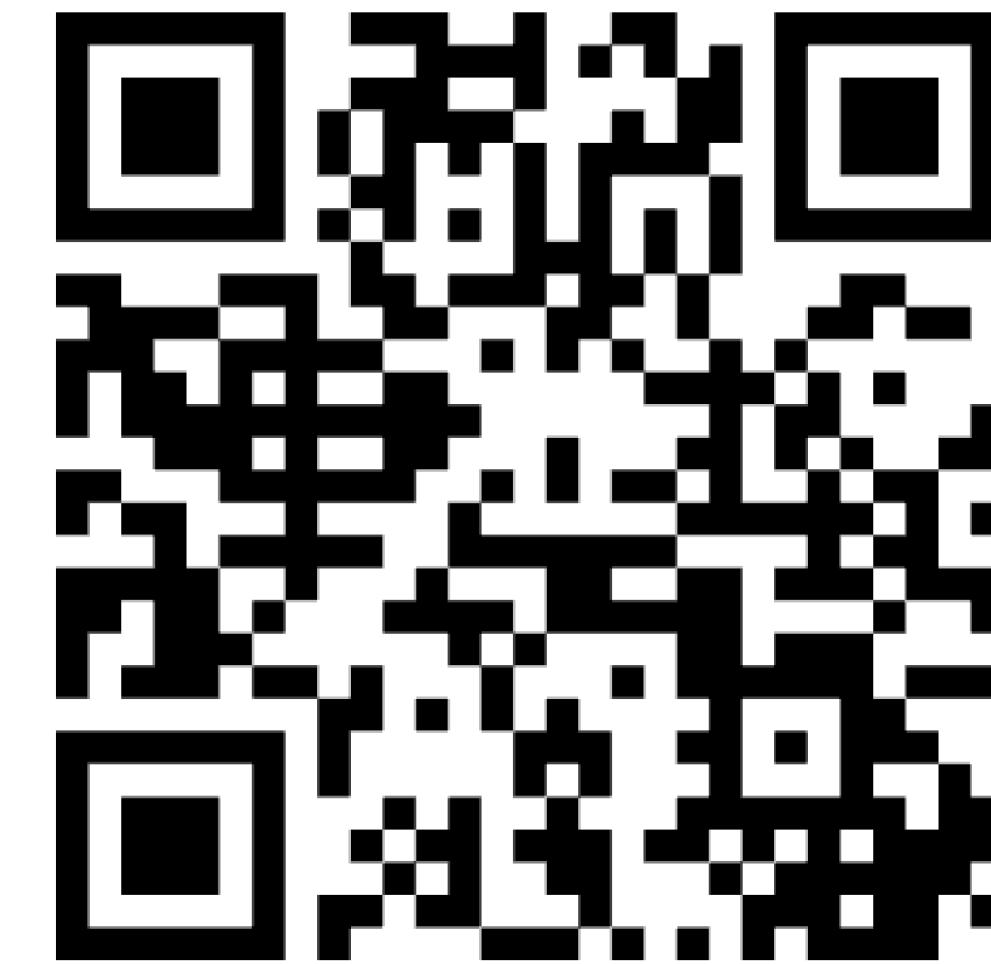
## TERRY QUISPE CLOUD ENGINEER



Bachiller en Ingeniería Electrónica con 2 años de experiencia en el sector TI, apasionado de tecnologias y disciplinas emergentes como cloud computing, ciberseguridad, IoT y devops.

# INDICACIONES

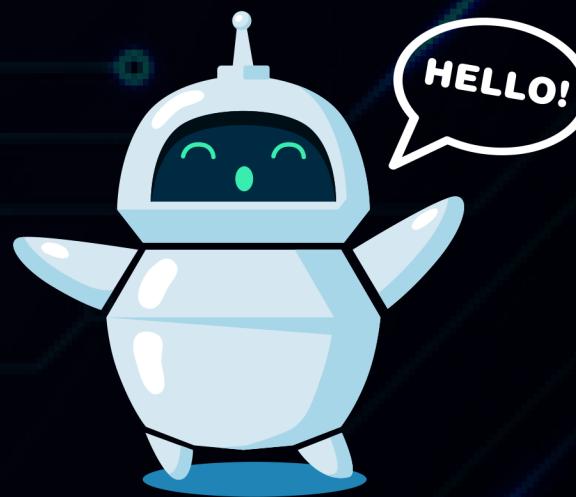
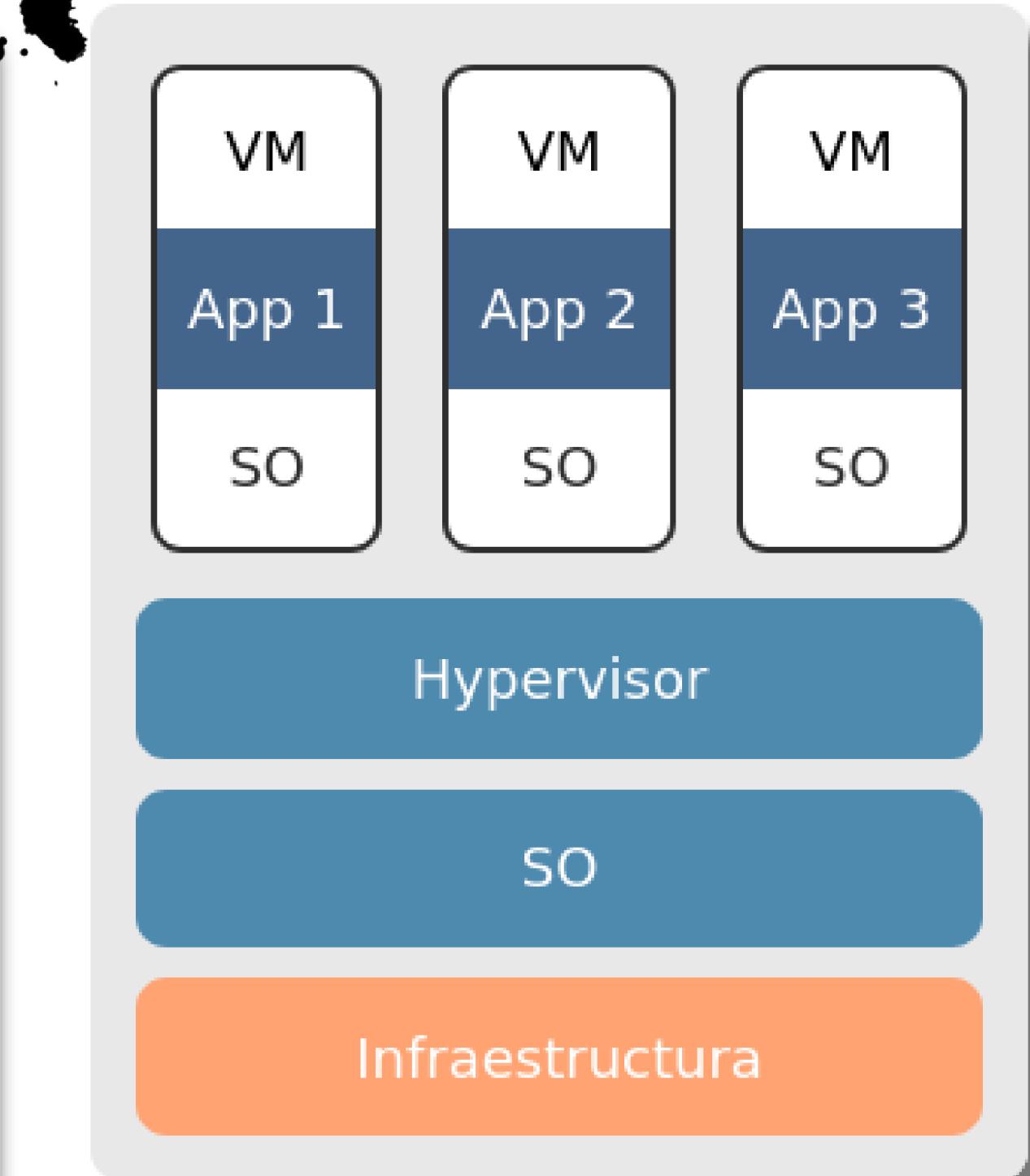
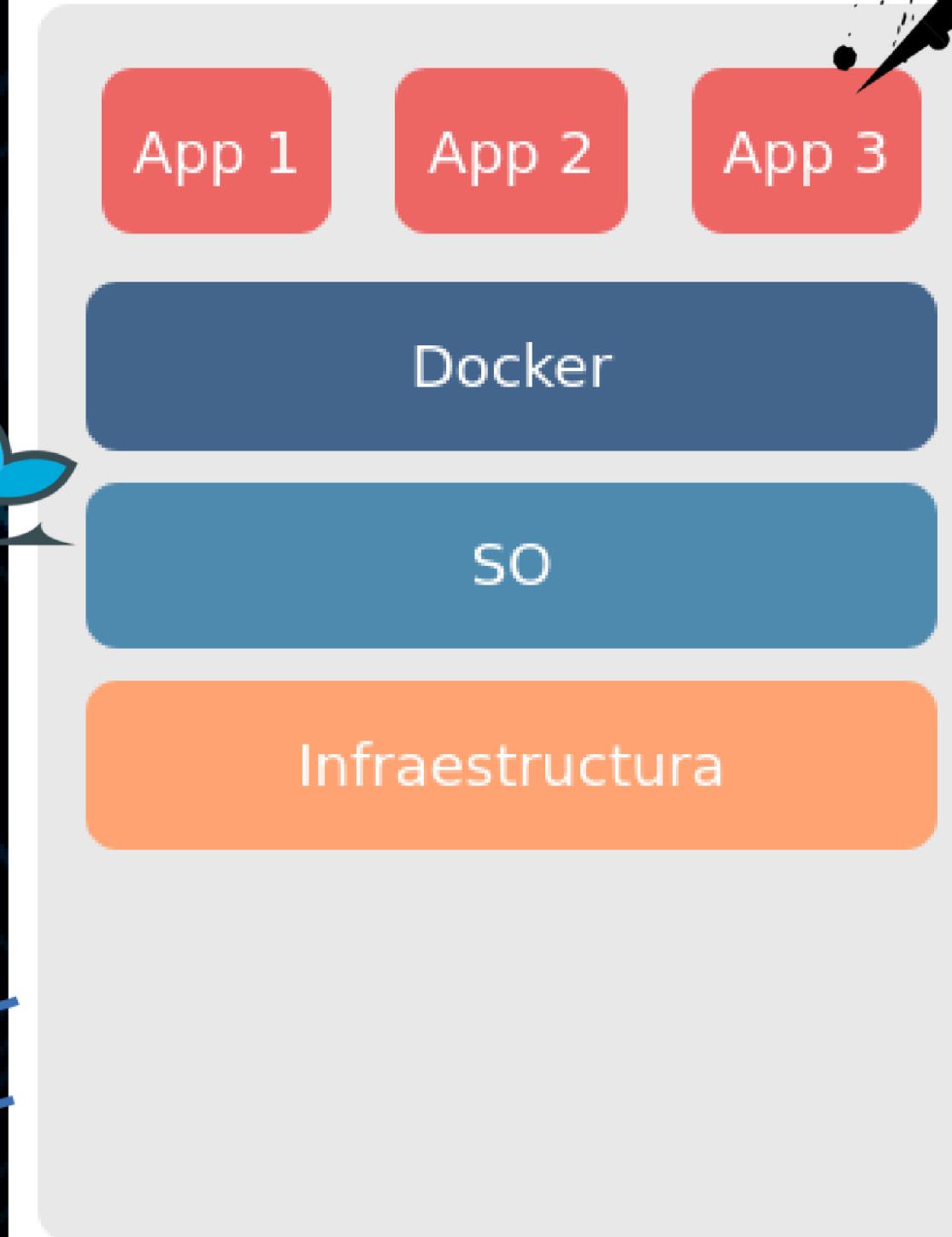
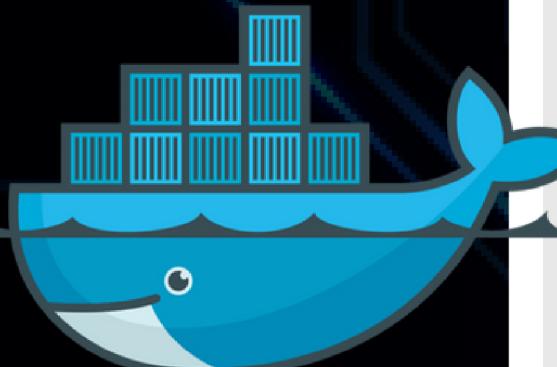
PREGUNTAS  
Y ENCUESTAS

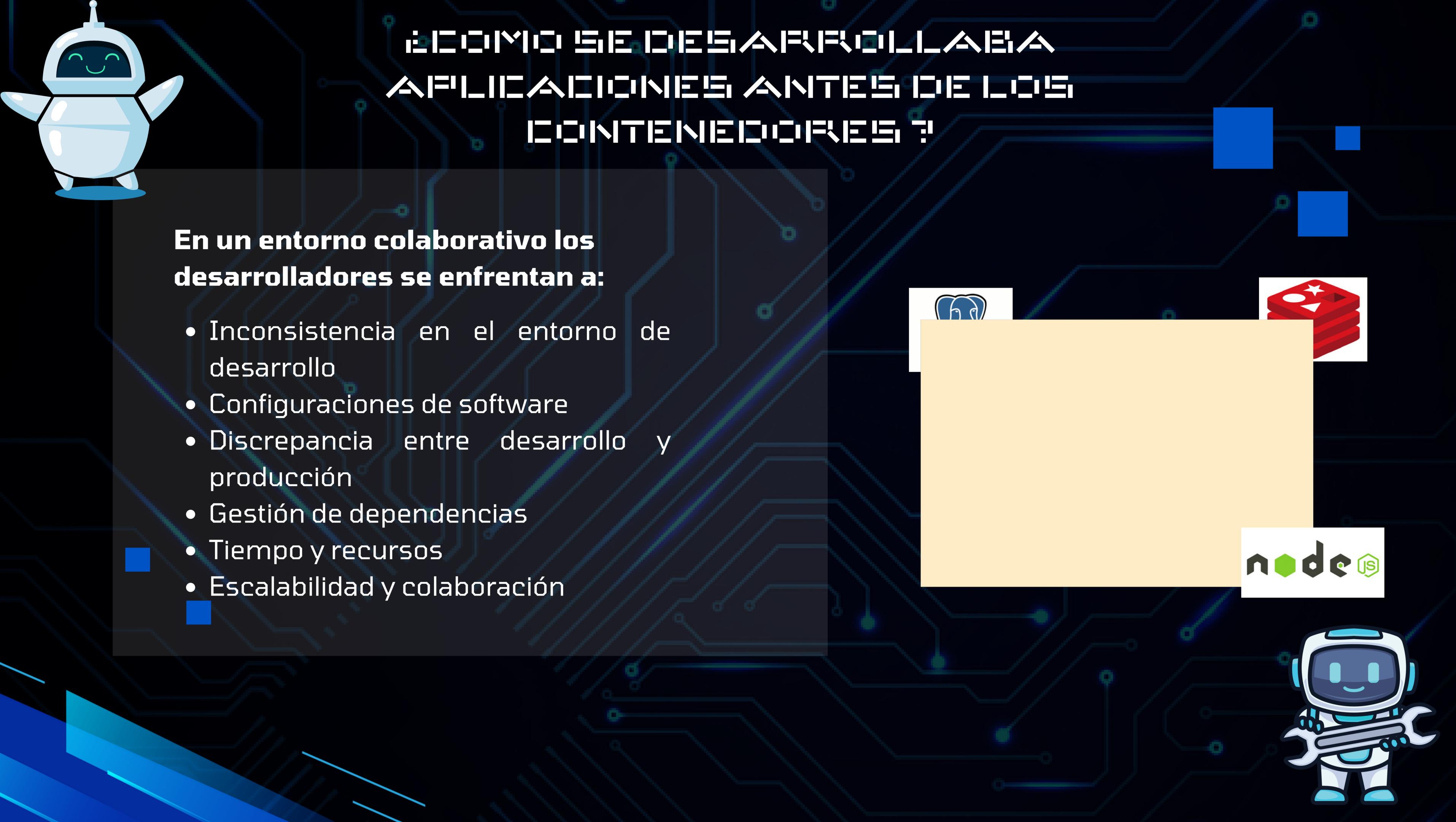


# INTRODUCCIÓN

CONTAINERS

VIRTUAL  
MACHINES

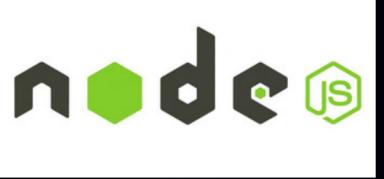
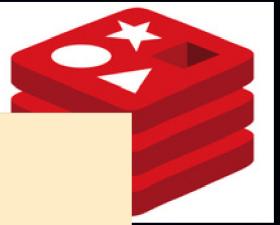




# ¿COMO SE DESARROLLABA APLICACIONES ANTES DE LOS CONTENEDORES?

**En un entorno colaborativo los desarrolladores se enfrentan a:**

- Inconsistencia en el entorno de desarrollo
- Configuraciones de software
- Discrepancia entre desarrollo y producción
- Gestión de dependencias
- Tiempo y recursos
- Escalabilidad y colaboración



# VENTAJAS DE LA CONTENIERIZACIÓN



Portabilidad



Agilidad



Velocidad



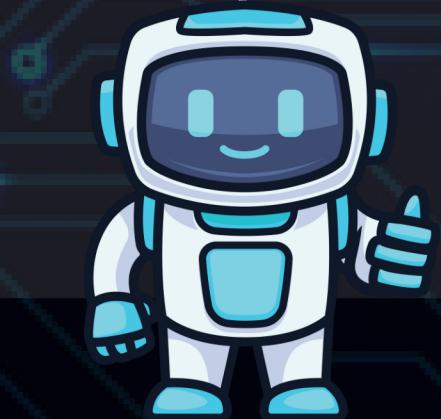
Aislamiento



Seguridad

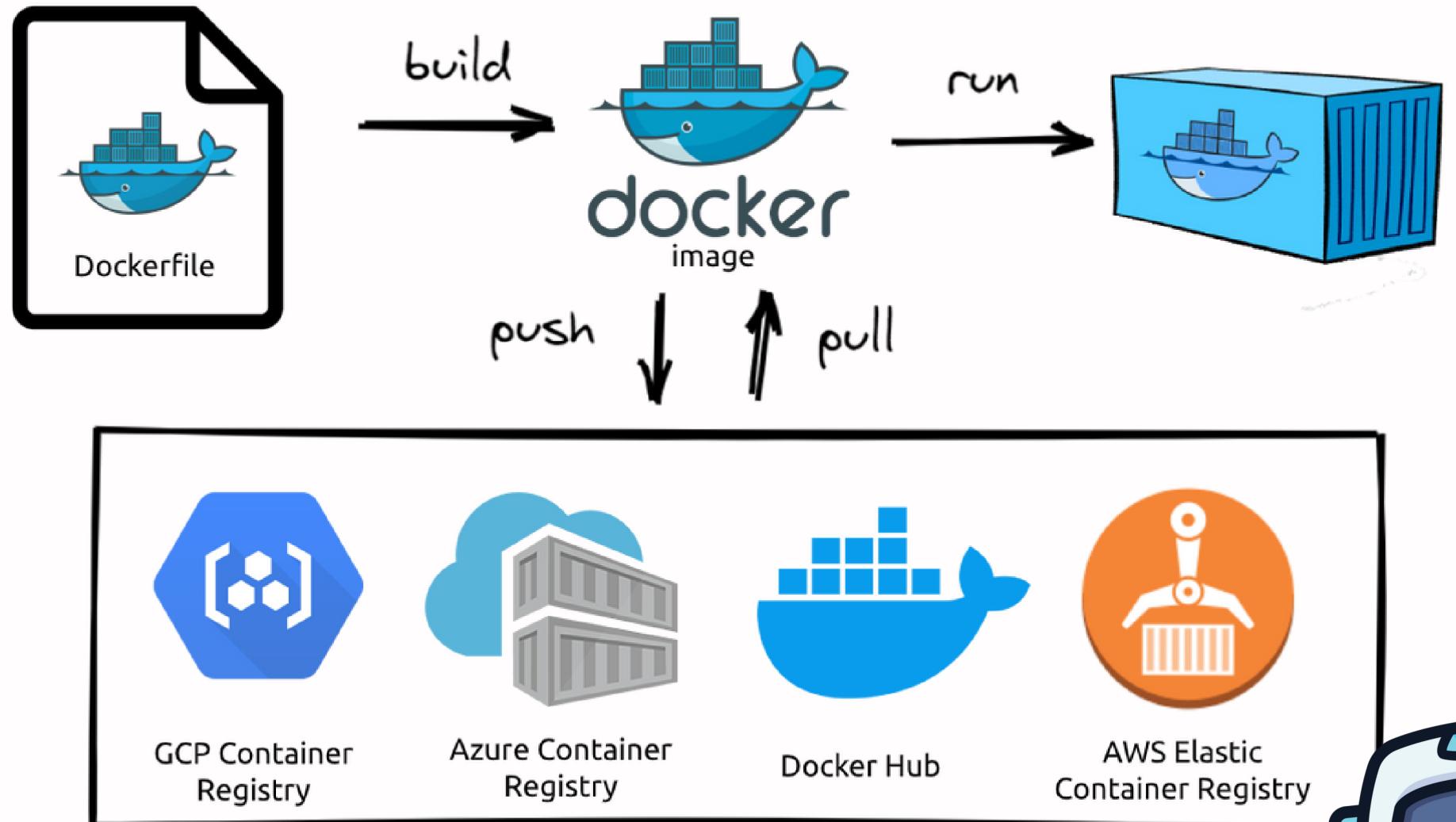


Eficiencia



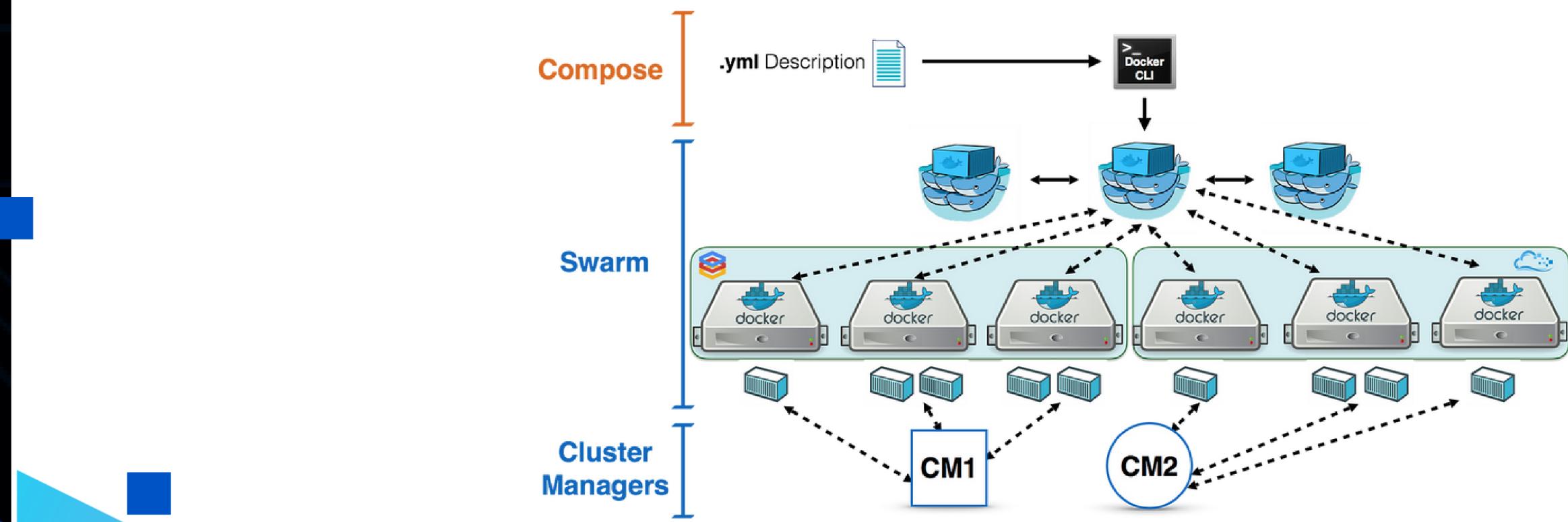
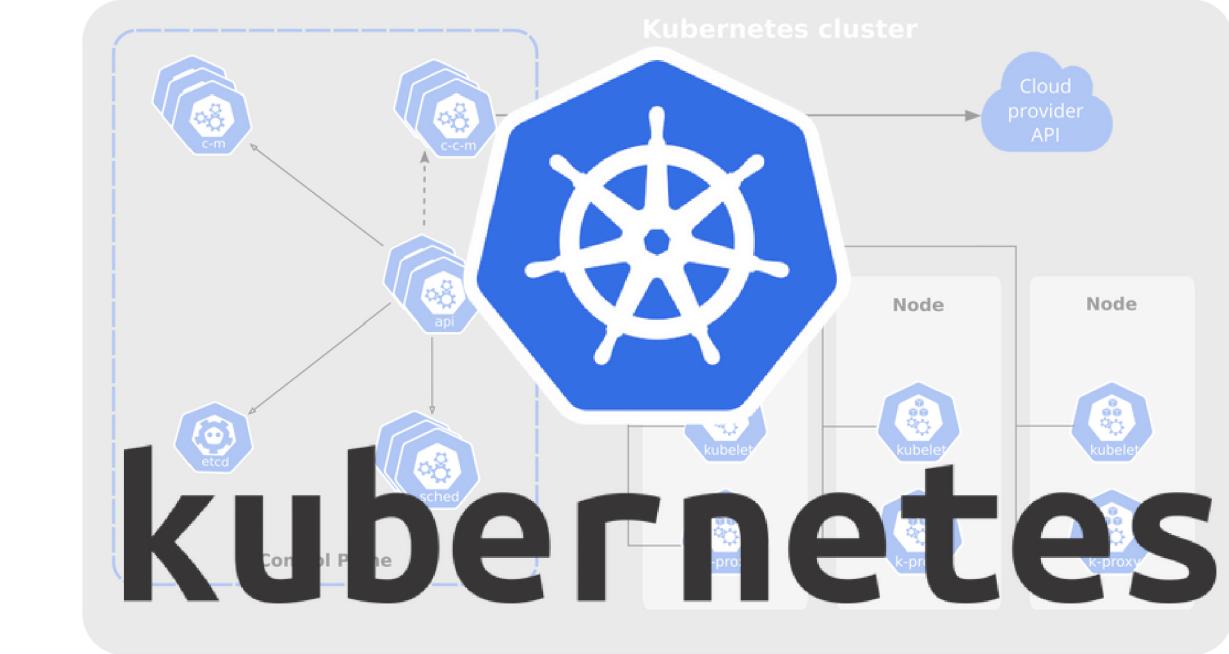
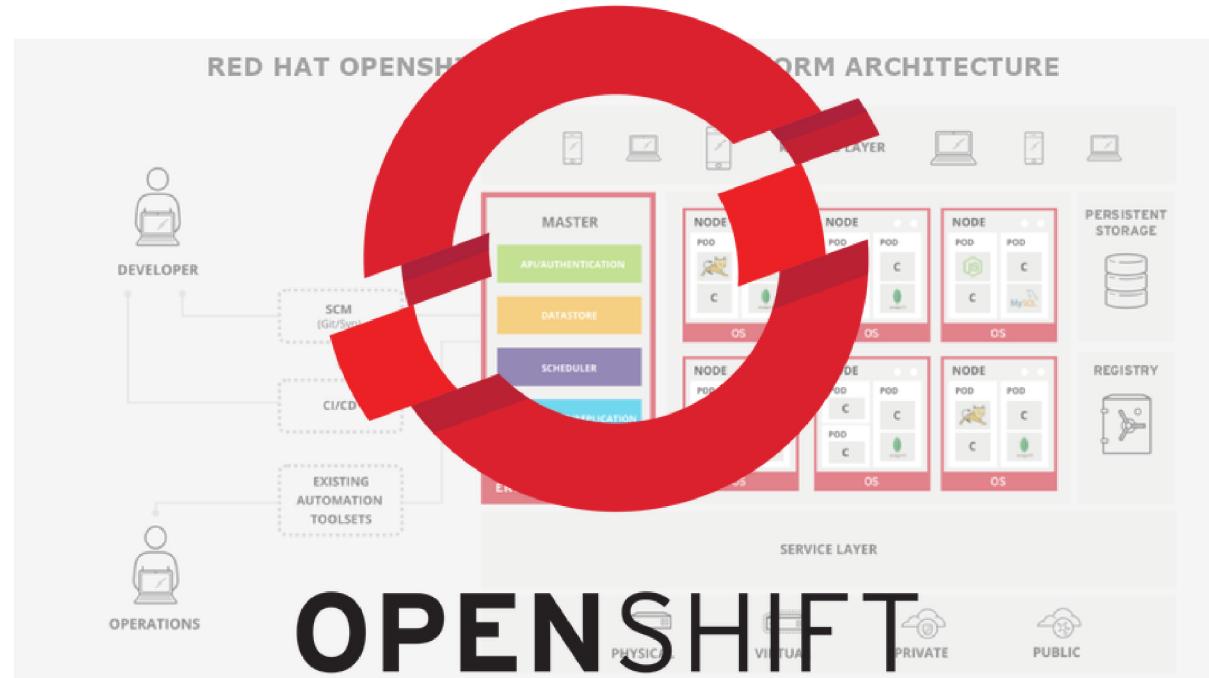
# REPOSITORIOS

```
1 # Utiliza Ubuntu 20.04 como imagen base
2 FROM ubuntu:20.04
3
4 # Establece el directorio de trabajo dentro del contenedor
5 WORKDIR /app
6
7 # Copia los archivos de la aplicación en el contenedor
8 COPY . /app
9
10 # Actualiza la lista de paquetes e instala dependencias
11 RUN apt-get update && apt-get install -y python3 python3-pip
12
13 # Instala las dependencias de Python especificadas en requirements.txt
14 RUN pip3 install --no-cache-dir -r requirements.txt
15
16 # Expone el puerto 5000 para la aplicación
17 EXPOSE 5000
18
19 # Define el comando por defecto para ejecutar la aplicación
20 CMD ["python3", "app.py"]
```

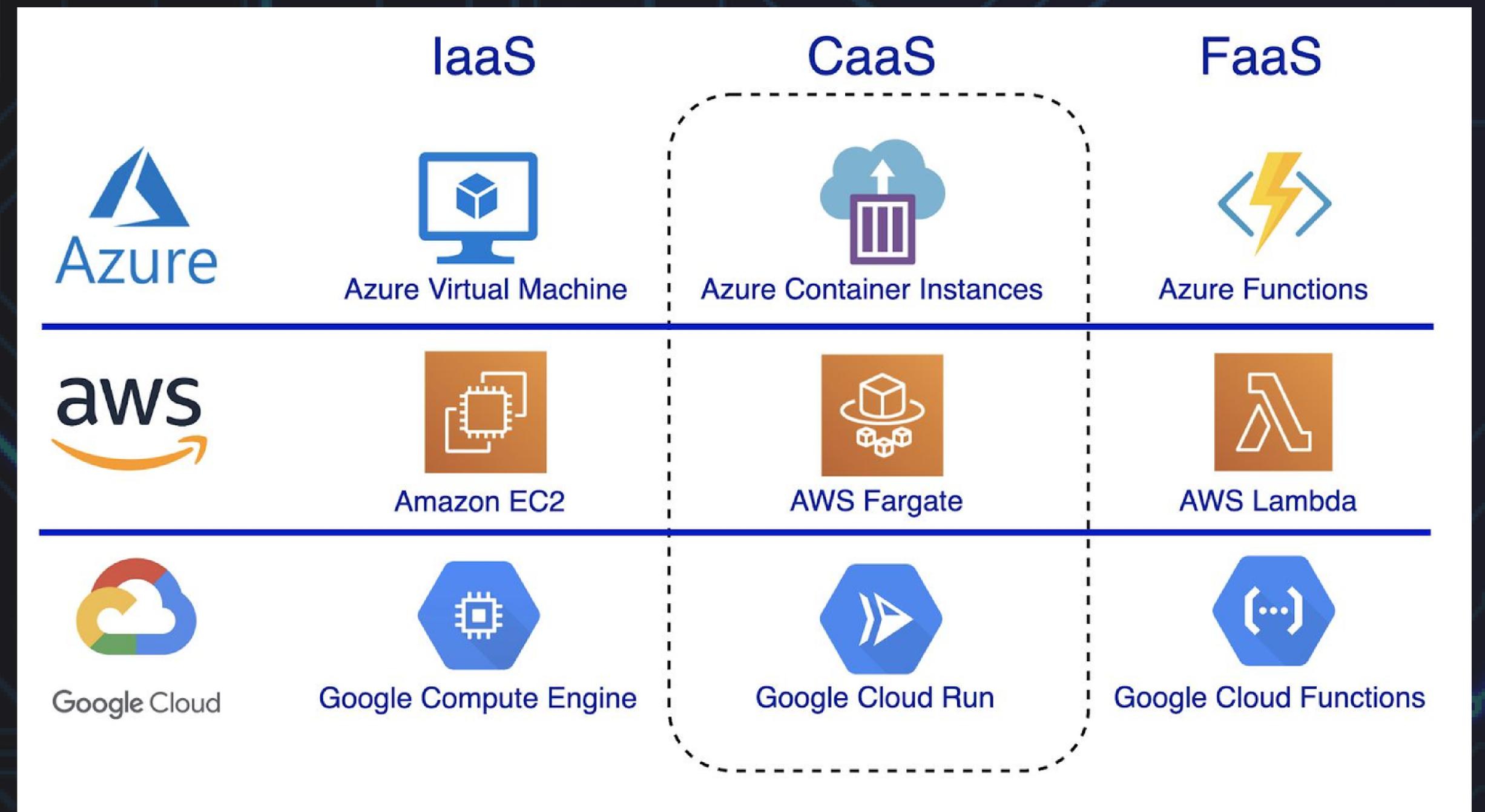




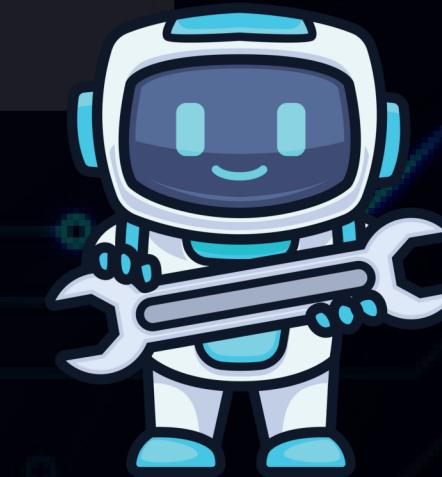
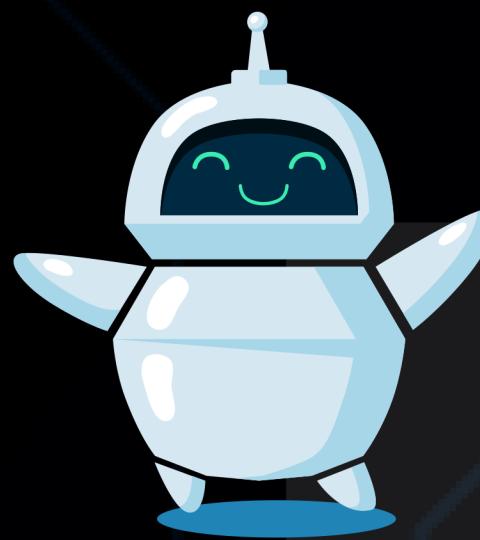
# ORQUESTADORES



# CaaS



# SERVICIOS DE ORQUESTACIÓN



**QUE PROVIDER CLOUD  
CONOCES?**

**QUE SERVICIOS DE  
CONTENEDORES  
CONOCES EN CLOUD?**

# CLOUD CONTAINER SERVICES

Docker in  
Cloud



Docker in  
my Machine



EKS



ECS



FARGATE



ELASTIC  
BEANSTALK



LIGHTSAIL



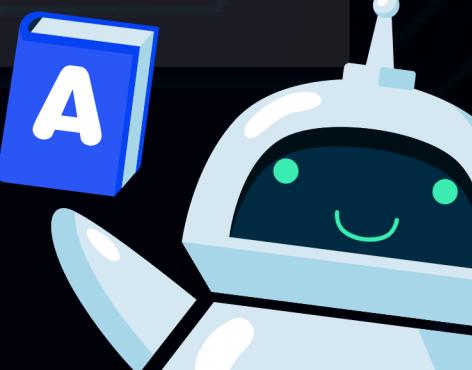
APP RUNNER

# AWS CONTAINER SERVICES

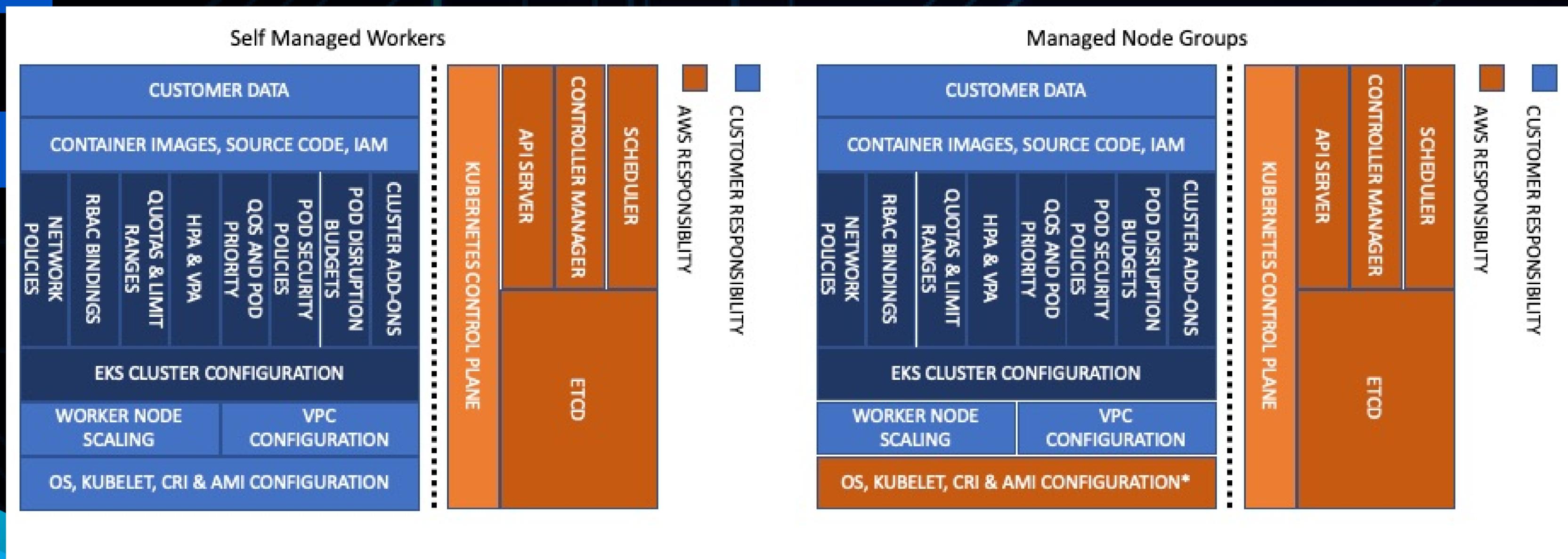


**Amazon EKS** proporciona gestión completa para aplicaciones contenerizadas con Kubernetes, escalabilidad automática y manual, y se integra a fondo con servicios de AWS como IAM y CloudWatch, asegurando redes con Amazon VPC y AWS PrivateLink.

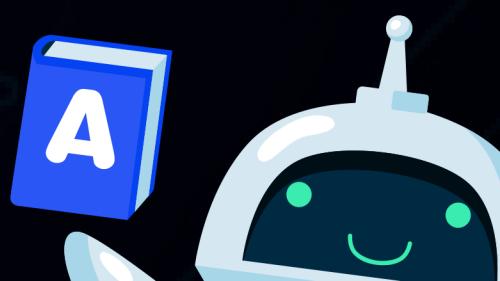
**Amazon ECS** gestiona contenedores Docker en AWS sin necesidad de manejar infraestructura, con orquestación propia, integración total con IAM, VPC, CloudWatch, y soporte para Auto Scaling y balanceo de carga, ideal para aplicaciones escalables en contenedores.



# AWS EKS RESPONSIBILITY MODEL

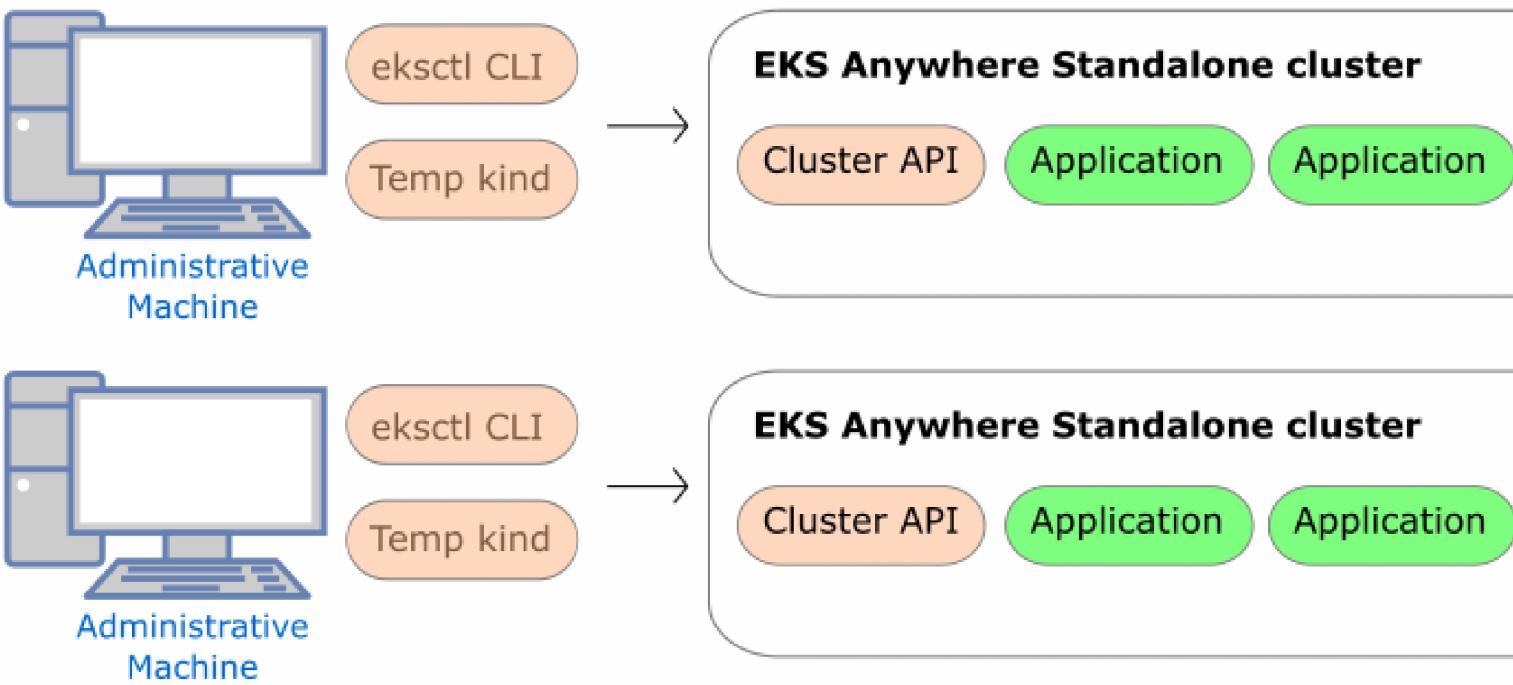


# AWS EKS RESPONSIBILITY MODEL

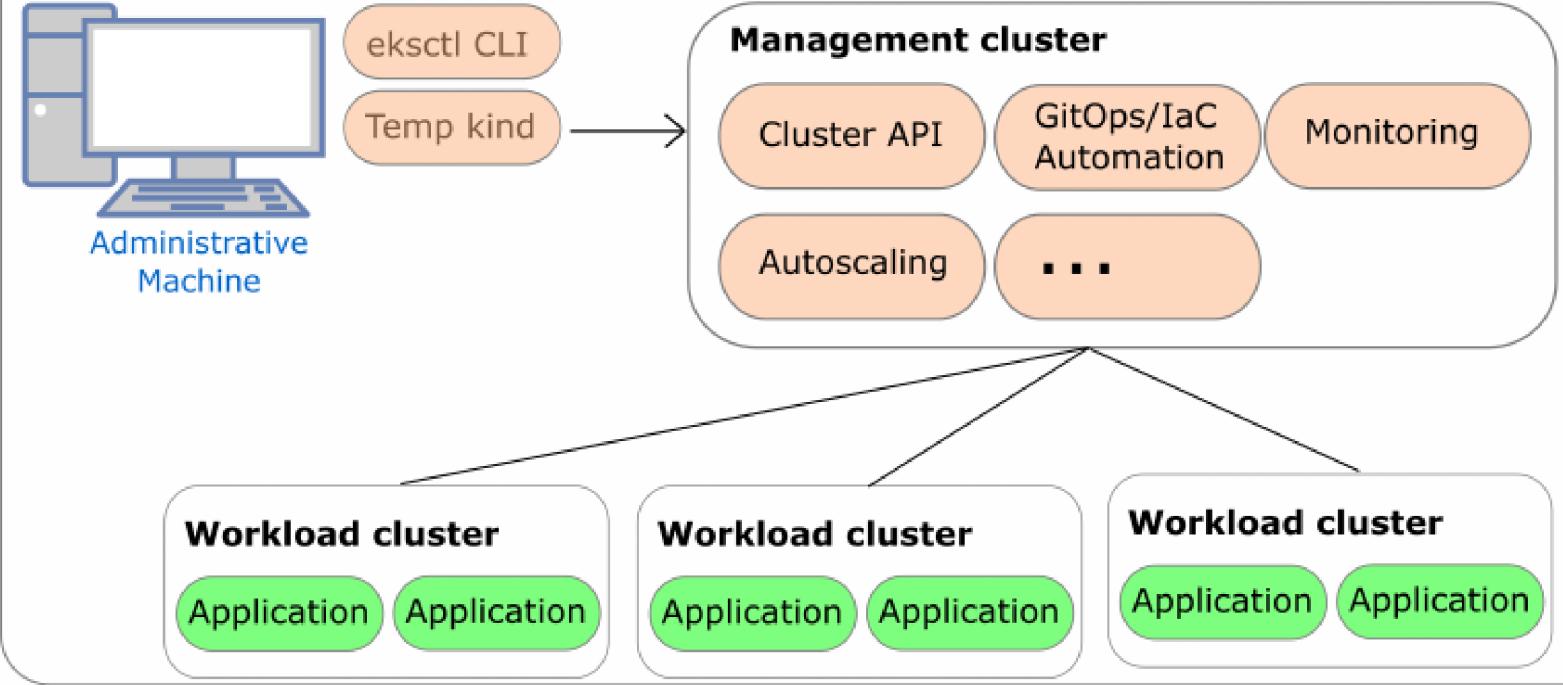


# AWS EKS ANYWHERE

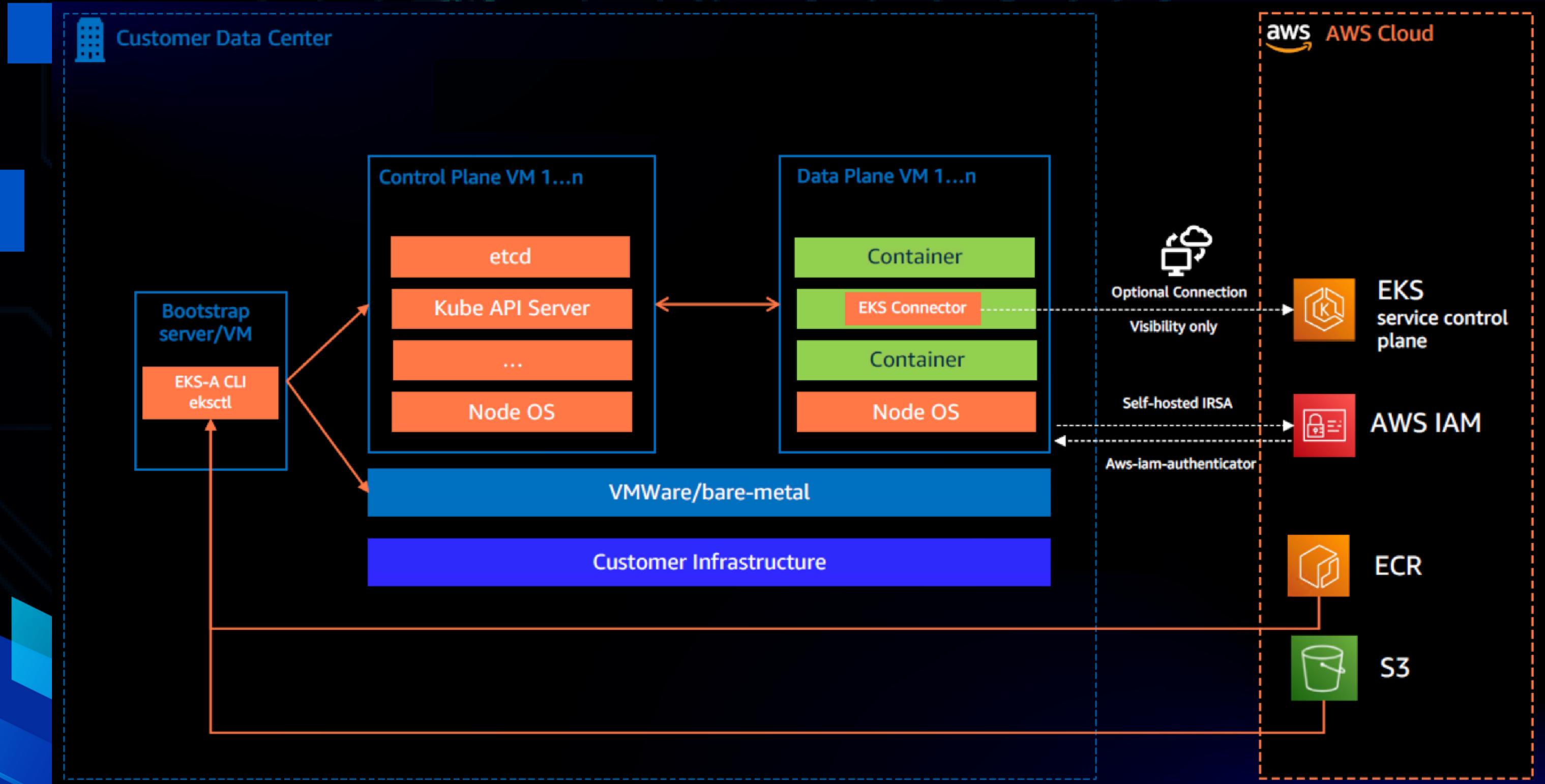
## Standalone cluster



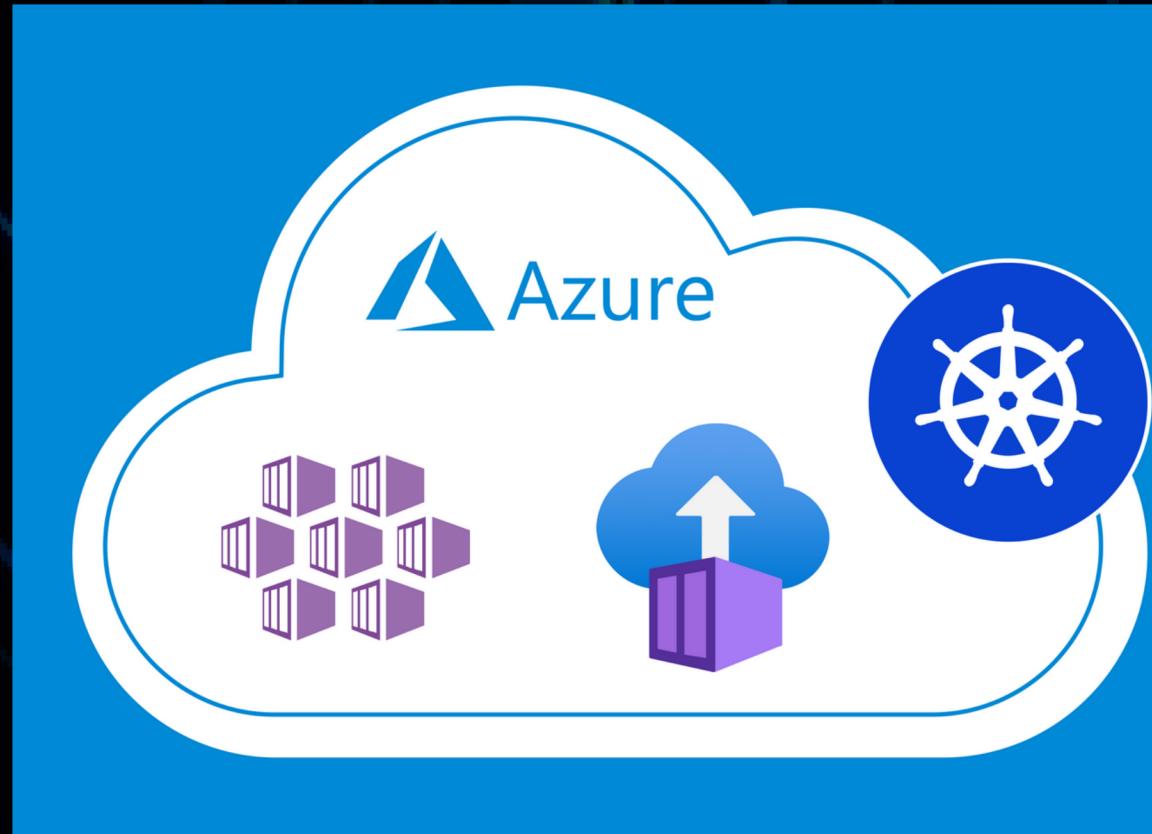
## EKS Anywhere Management / Workload clusters



# AWS EKS ANYWHERE ARCHITECTURE



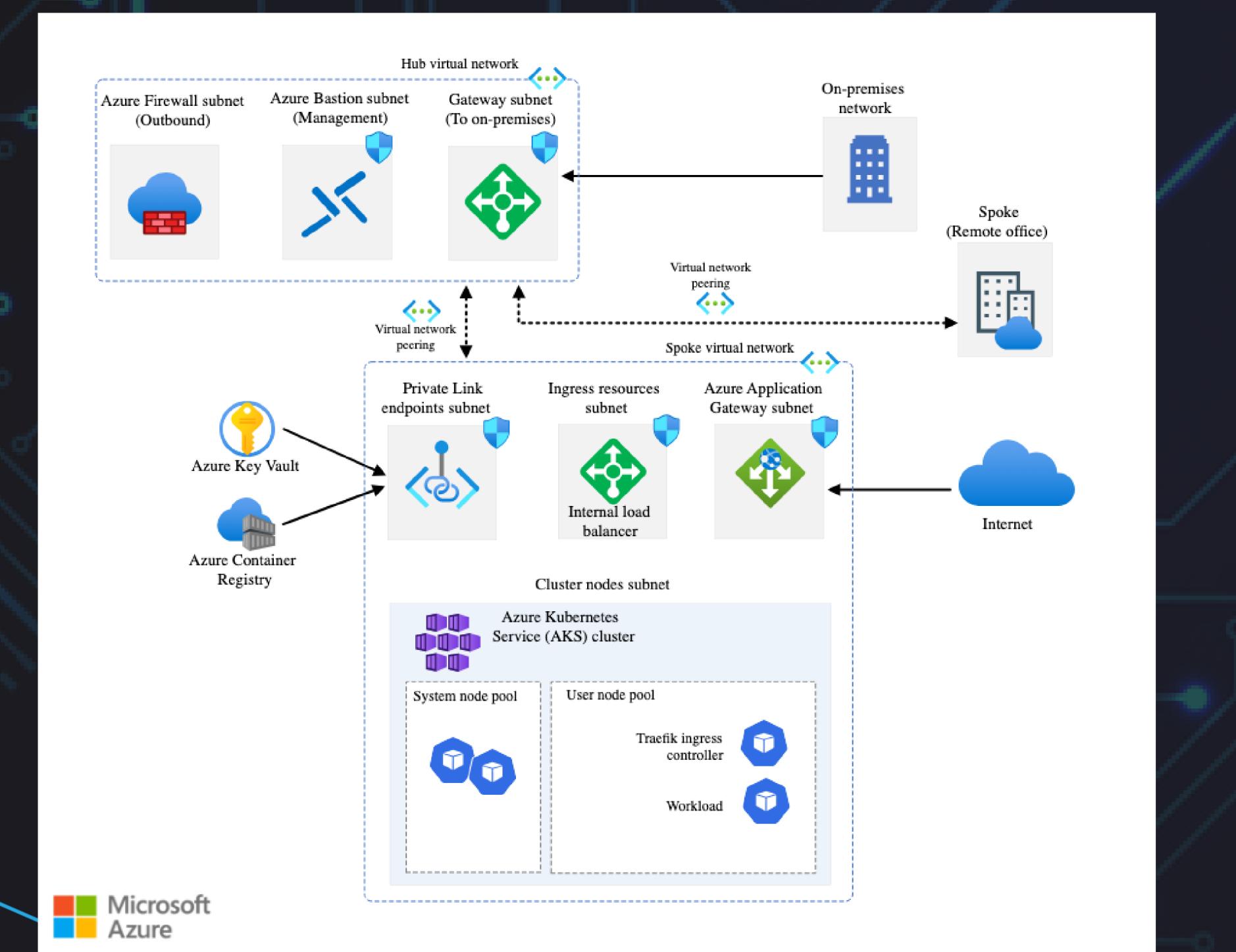
# AZURE CONTAINER SERVICES



**Azure AKS** ofrece gestión completa de aplicaciones contenerizadas con Kubernetes, integración nativa con servicios como Azure Active Directory y Azure Monitor, seguridad mejorada con Azure Security Center, y networking integrado con Azure Virtual Network para controlar el tráfico de red.

**Azure Container Instances (ACI)** permite ejecutar contenedores sin servidor, ideal para cargas de trabajo con escalado rápido y bajo demanda, con total integración en el ecosistema de Azure incluyendo Azure Monitor y Azure Active Directory, y ofrece simplicidad al eliminar la necesidad de gestionar infraestructura subyacente.

# AZURE ARCHITECTURE CONTAINER SERVICES



# AZURE CLUSTER CI/CD



# CONTACTANOS

HASSIEL MUÑOZ



LINKEDIN:

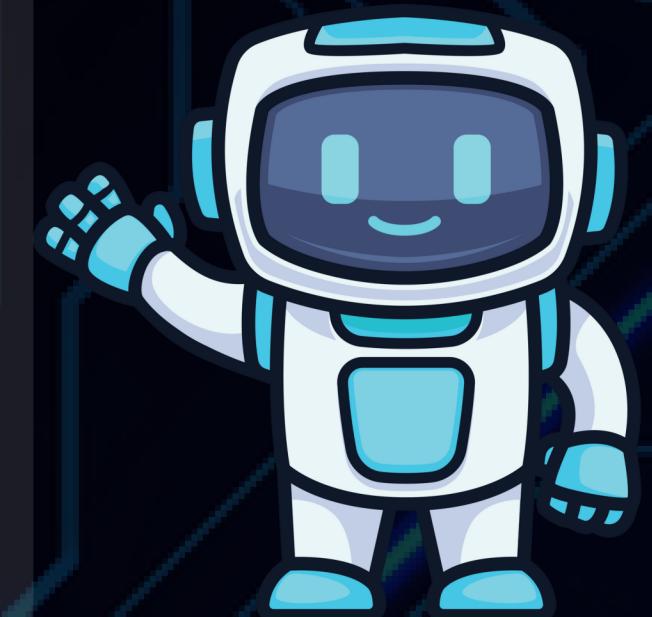
<https://www.linkedin.com/in/munozhassel/>

GITHUB:

<https://github.com/8infinitecloud>

CORREO:

[munozhassel@outlook.com](mailto:munozhassel@outlook.com)



TERRY QUISPE



LINKEDIN:

[www.linkedin.com/in/terry-martin-quispe-paniagua](https://www.linkedin.com/in/terry-martin-quispe-paniagua)

CORREO:

[terryfiee2017@gmail.com](mailto:terryfiee2017@gmail.com)