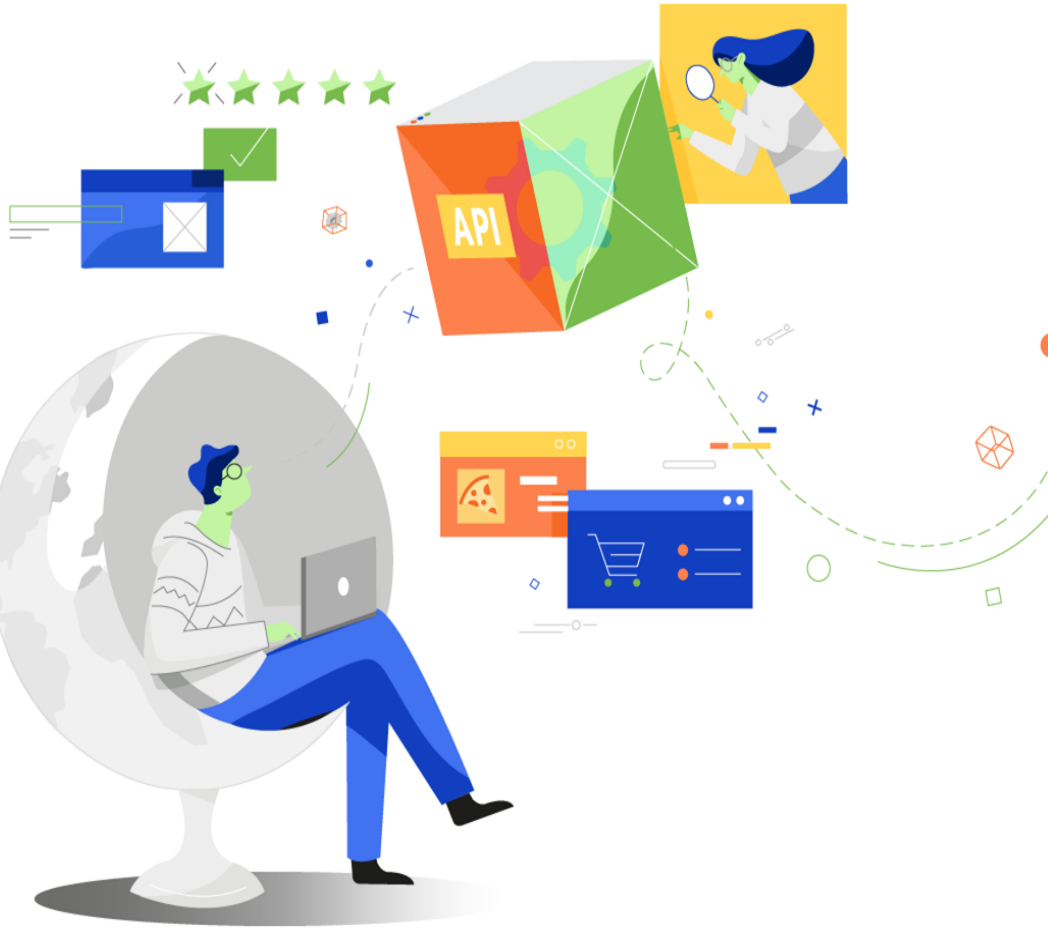




ACADEMIA CLOUD



CURSO ÚNICO EN MÉXICO

**¡Bienvenido al
Curso:**

**DevOps: CI-CD,
Infraestructura en
AWS y
Contenedores**

Trainer: Emilio Moreno
Cloud Architect | DevOps Consultant

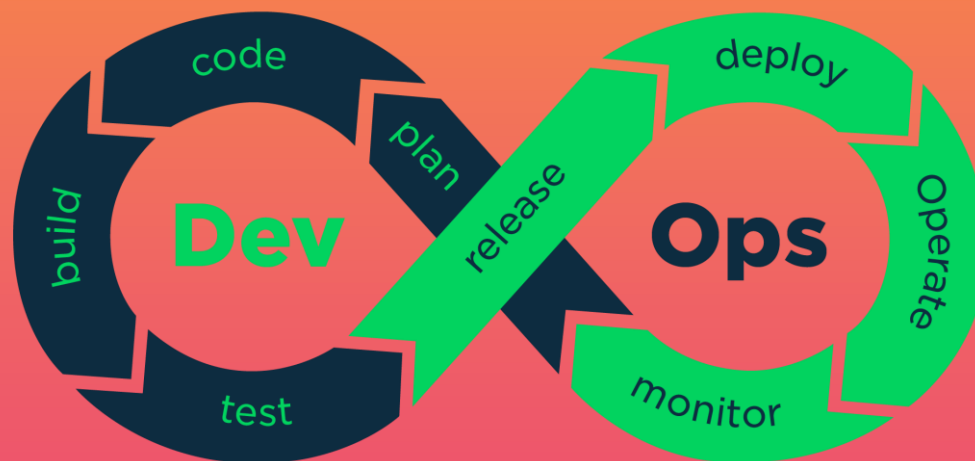
Mail: emilio@touchof.tech
Cel: +52 331 599 6389

Agenda



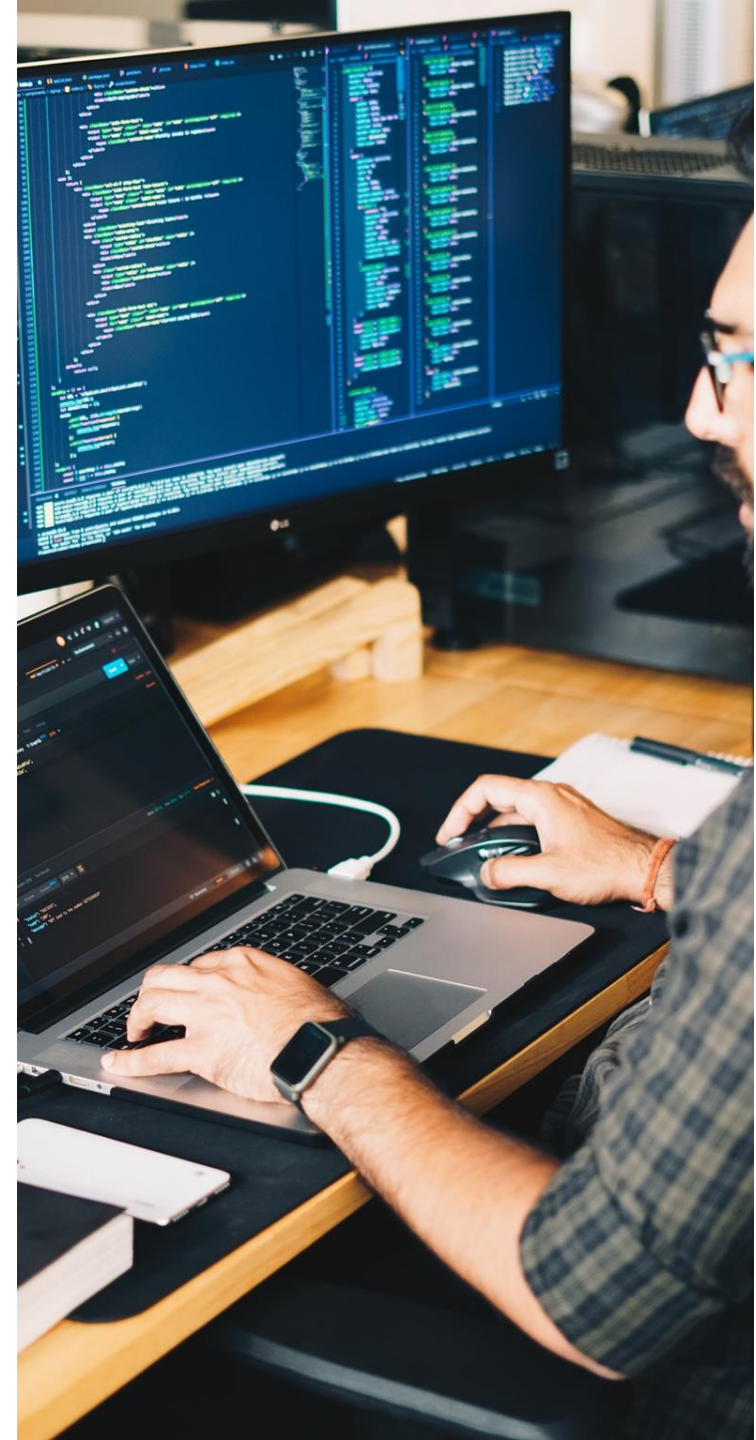
Semana 1

- Introducción a Docker
- ¿Qué es Docker?
- Aplicaciones en contenedores
- Arquitectura de Docker
 - Docker engine
 - Imágenes
 - Contenedores
 - Registros
 - Redes
 - Volúmenes
 - Orquestración
 - Seguridad



Módulo I

Introducción a Docker

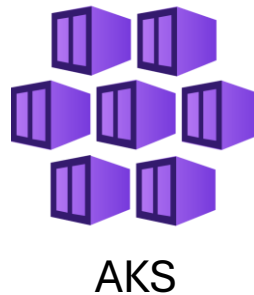
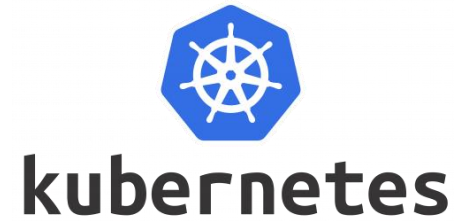
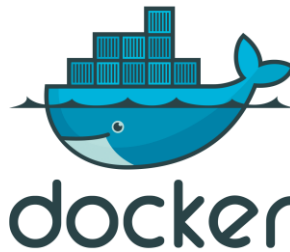




¿Qué es Docker?

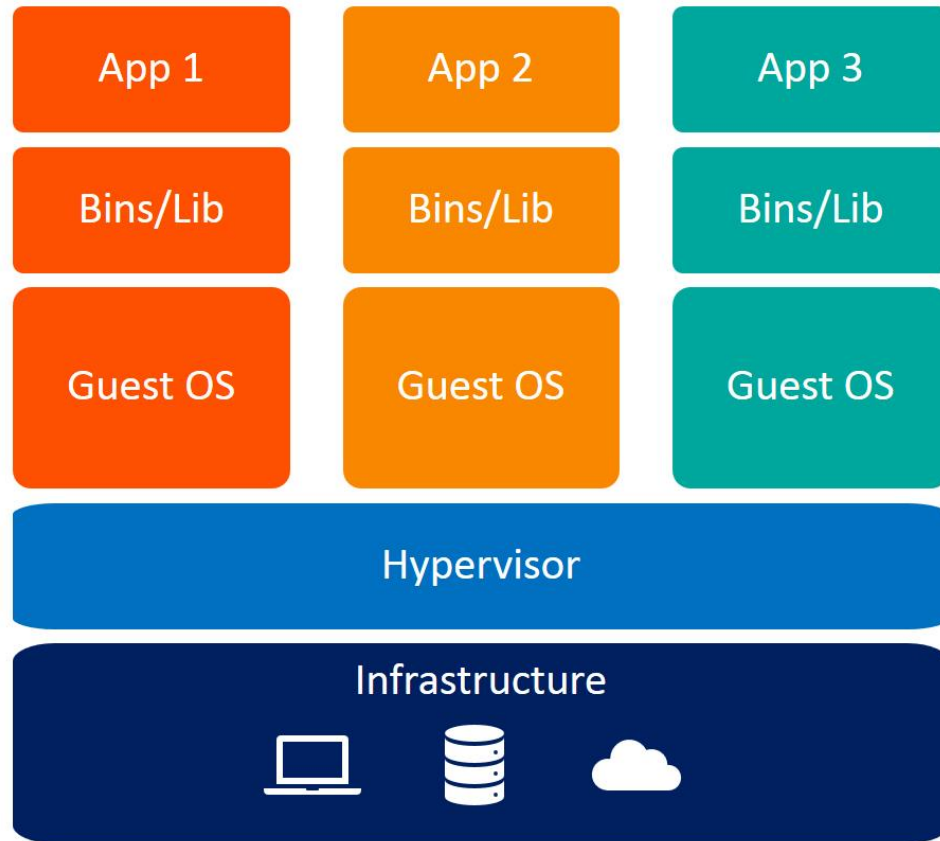
Plataforma de código abierto para crear, desplegar y ejecutar aplicaciones en contenedores

- Portabilidad
- Aislamiento
- Eficiencia
- Flexibilidad

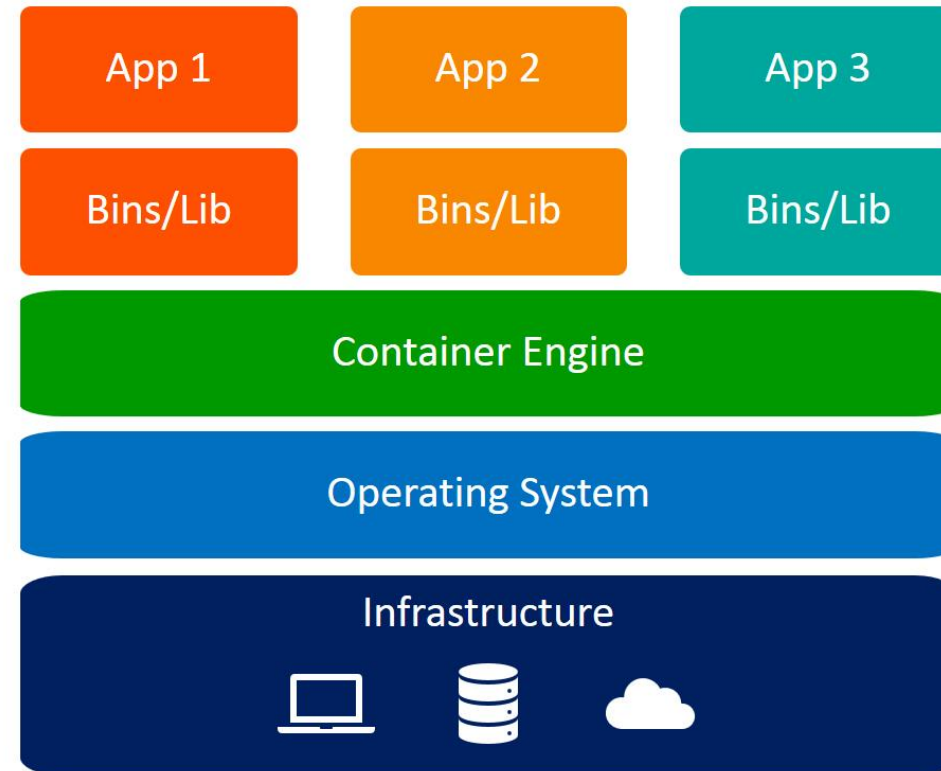




ACADEMIA CLOUD



Virtual Machines



Containers

Módulo II

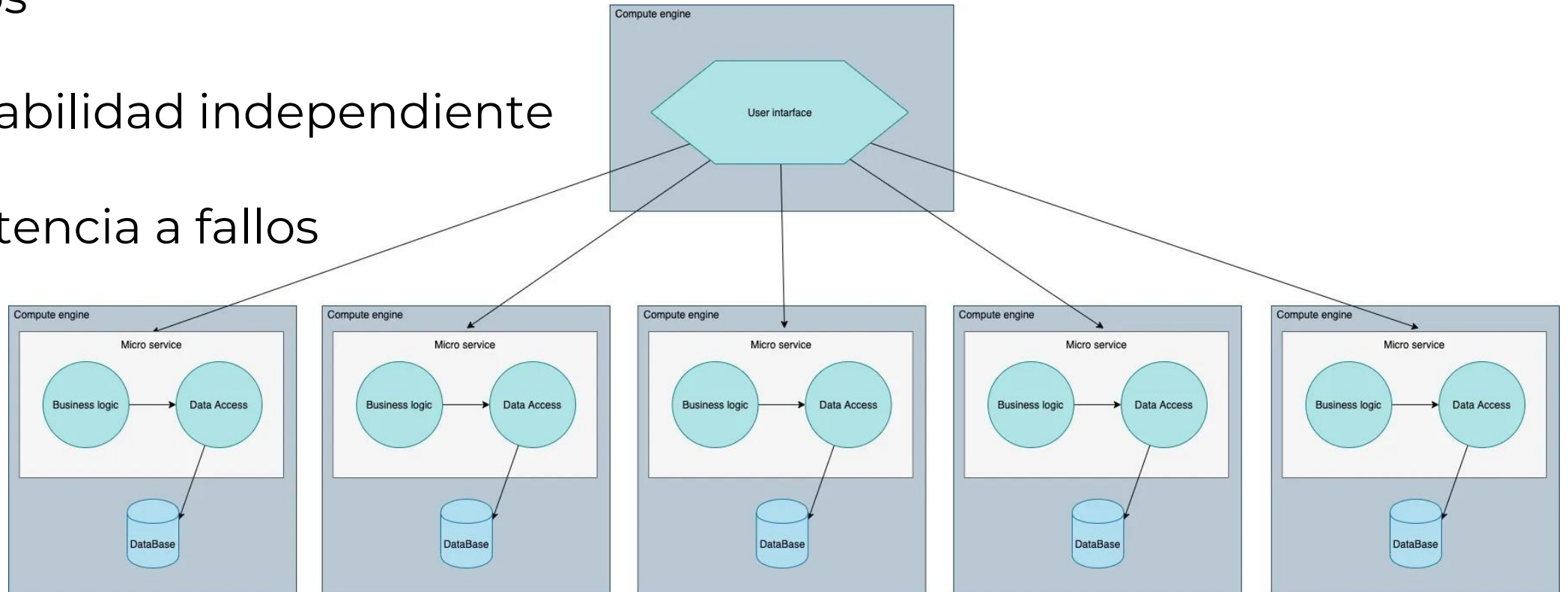
Aplicaciones en contenedores





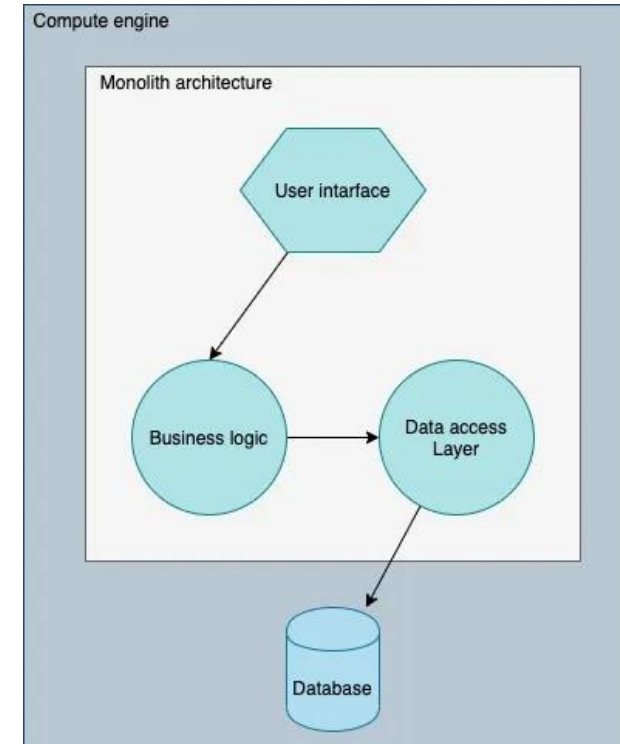
Microservicios

- Servicios independientes y únicos
- Escalabilidad independiente
- Resistencia a fallos



Monolítico

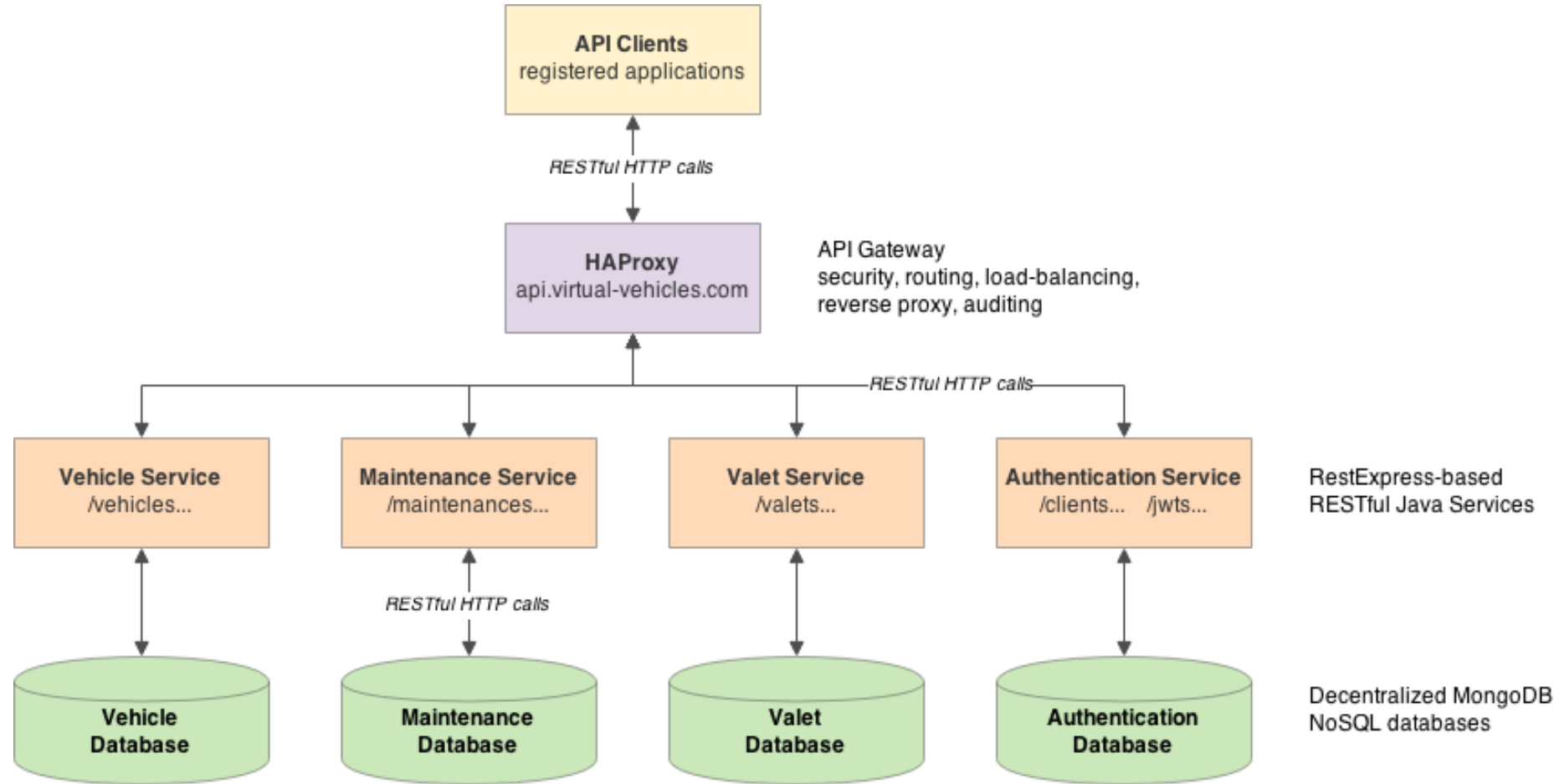
- Una sola unidad de implementación
- Dependencia en los componentes
- Escalabilidad limitada



Ejemplo

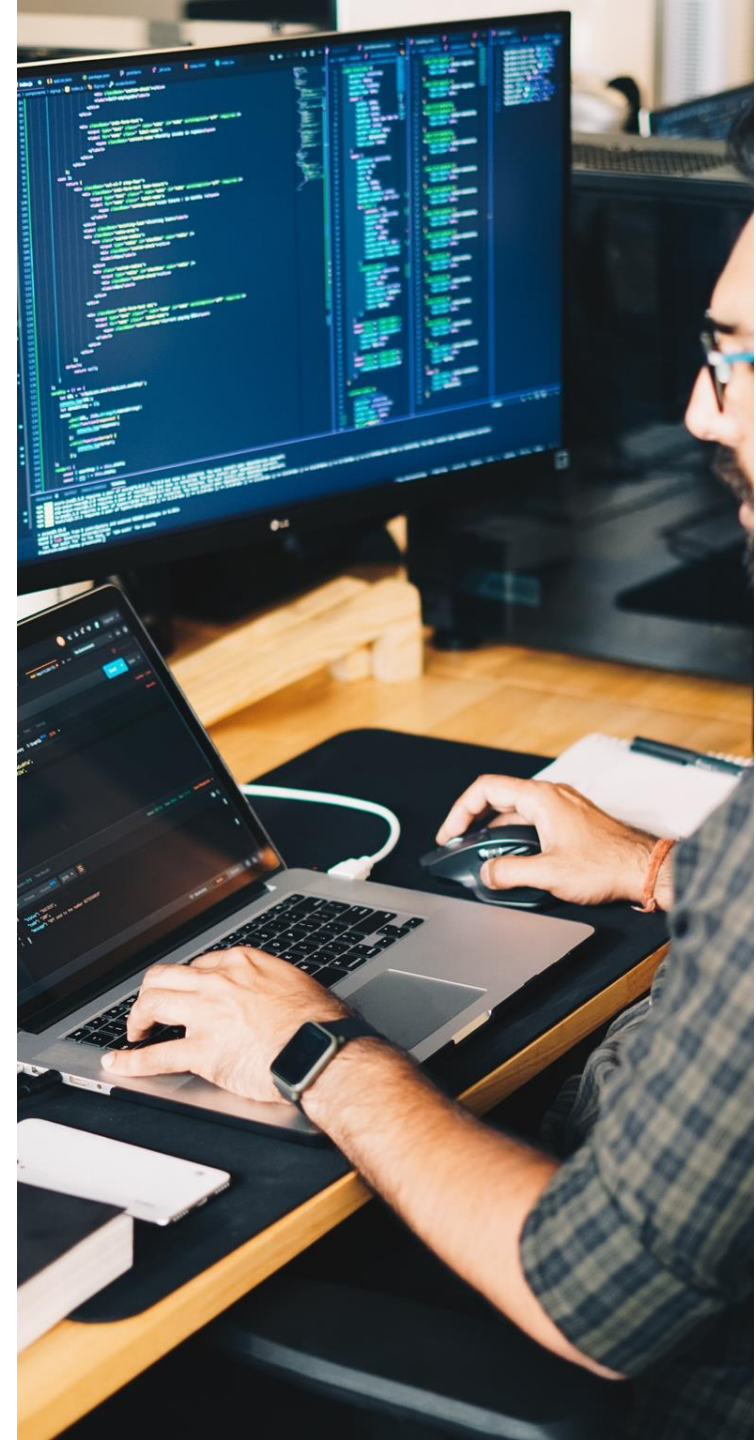


ACADEMIA CLOUD



Módulo III

Arquitectura de Docker

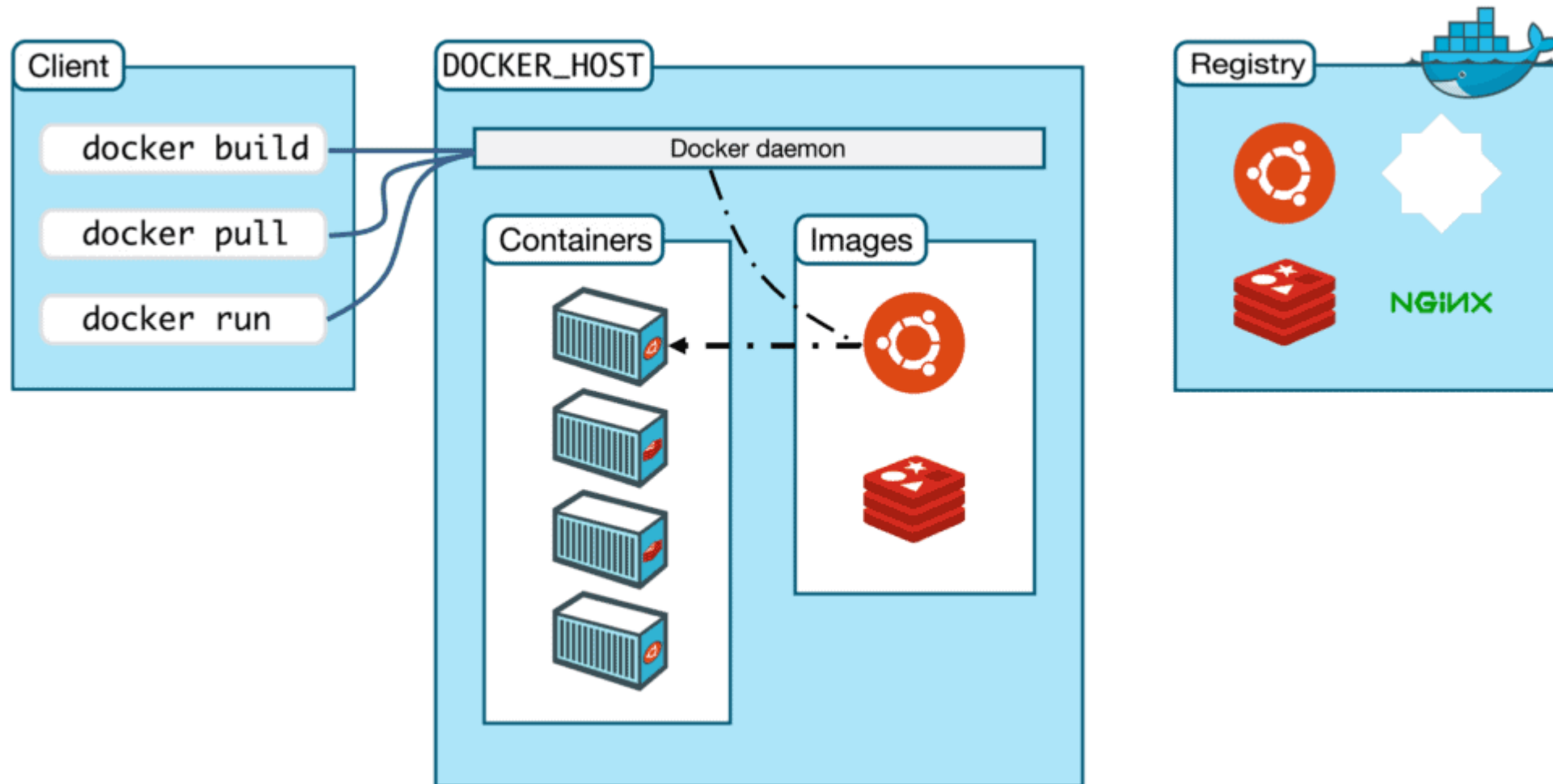




Arquitectura

- Docker engine
- Imágenes
- Contenedores
- Registros
- Redes
- Volúmenes
- Orquestación
- Seguridad

Ejemplo





Recursos

Cheatsheet: https://docs.docker.com/get-started/docker_cheatsheet.pdf

Docs: <https://docs.docker.com/reference/cli/docker/>

Dockerfile: <https://docs.docker.com/reference/dockerfile/>

Q&A





Ejercicio

```
docker pull nginx:1.25.5
```

```
docker run -d -p 80:80 nginx:1.25.5
```

```
docker ps
```

```
docker stop <CONTAINER_ID>
```

```
docker run -d --name mongo-academia -e  
MONGO_INITDB_ROOT_USERNAME=apix -e  
MONGO_INITDB_ROOT_PASSWORD=apix -e MONGO_INITDB_DATABASE=apix -p  
27017:27017 mongo:latest
```

Ejercicio



```
git clone https://github.com/lemiliomoreno/demo-devops-aws.git
```

```
cd src
```

```
docker run -d --name mongo-apix -e MONGO_INITDB_ROOT_USERNAME=apix -e MONGO_INITDB_ROOT_PASSWORD=apix -e  
MONGO_INITDB_DATABASE=apix -p 27017:27017 mongo:latest
```

```
pip install venv
```

```
python -m venv .venv
```

```
pip install poetry
```

```
poetry install
```

```
poetry run uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

```
Visitar localhost:8000/docs
```

```
Mac/Linux: source .venv/bin/activate
```

```
Windows: .venv/Scripts/activate
```

```
docker stop <CONTAINER_ID>
```



Ejercicio

```
docker run -d --name mongo-apix -e MONGO_INITDB_ROOT_USERNAME=apix -e  
MONGO_INITDB_ROOT_PASSWORD=apix -e MONGO_INITDB_DATABASE=apix -p 27017:27017  
mongo:latest
```

```
docker inspect <CONTAINER_ID> # obtener IP del contenedor
```

```
docker build -t apix-demo:1.0.0 .
```

```
docker run -d -e MONGODB_URI='mongodb://apix:apix@<CONTAINER_IP>:27017' -p 80:80 apix-  
demo:1.0.0
```