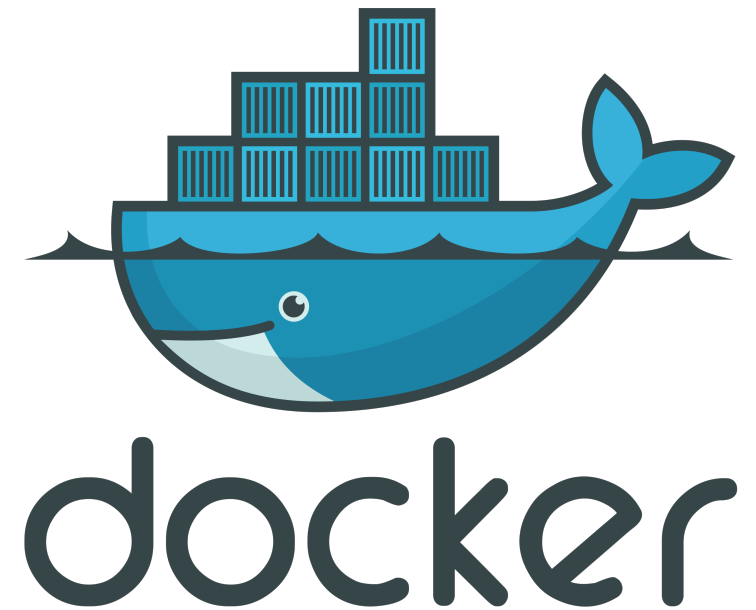


Docker 101 – Módulo 4

GBM Tech Academy

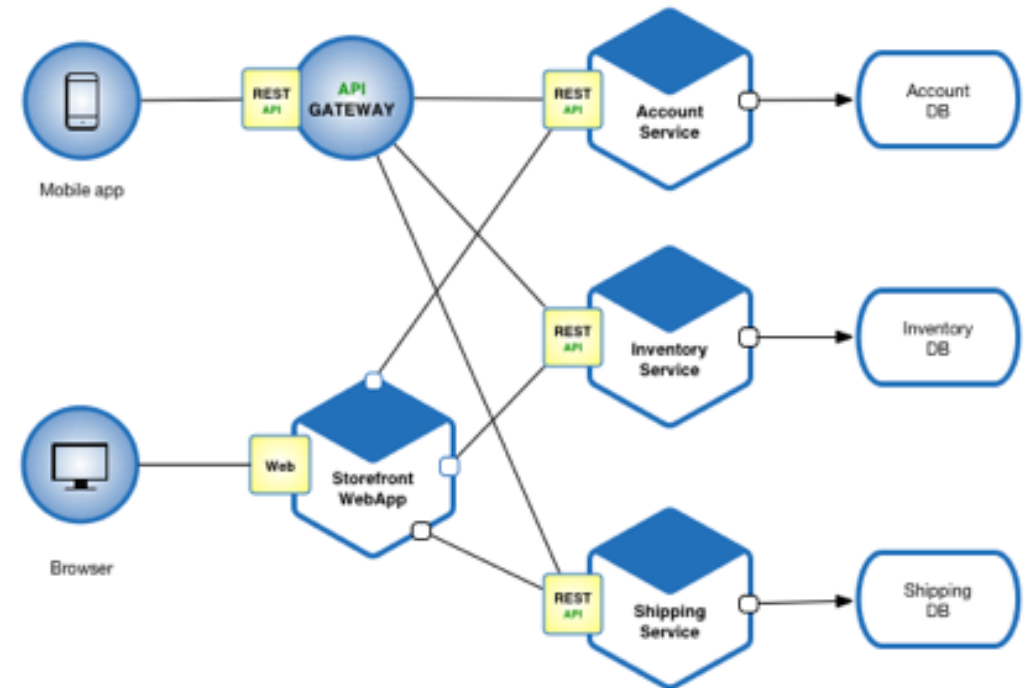
Agenda

- Aplicaciones multi-contenedores
- Composición de servicios
- Manipulación de servicios
- **Extra:** orquestación de contenedores



Arquitectura de Microservicios

- Patrón de diseño de aplicaciones
- Su propósito es estructurar aplicaciones como un conjunto de funcionalidades de negocio desacopladas
- Fomenta la exposición rápida y segura de servicios a los clientes.
- Pieza fundamental de la estrategia de **Transformación Digital** que busca mejorar la experiencia de los clientes
- Enfocada en incrementar los ingresos a partir de la entrega de nuevos servicios



Principios y Beneficios de los Microservicios

Principios

One Job

Cada microservicio debe estar optimizado para una sola función

Separate Processes

Comunicación entre microservicios debe estar basada en REST API y Message Brokers

Execution Scope

El foco no está en la interfaz sino en el componente

Separate CI/CD

Cada microservicio debe poder evolucionar a su propio ritmo

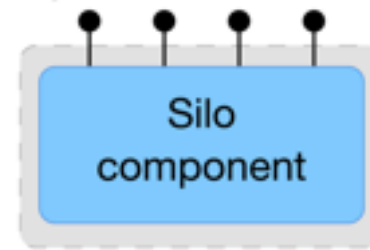
Resiliency

Cada microservicio tiene sus necesidades de alta disponibilidad y recuperación ante fallos

Beneficios

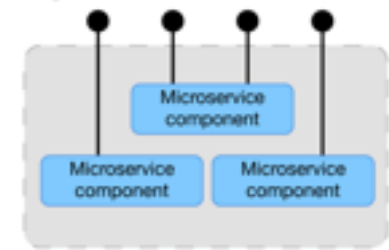
- Efficient Teams
- Simplified Deployment
- Right Tools for the Job
- Improved Application Quality
- Scalability

Exposed services/APIs



Monolithic application

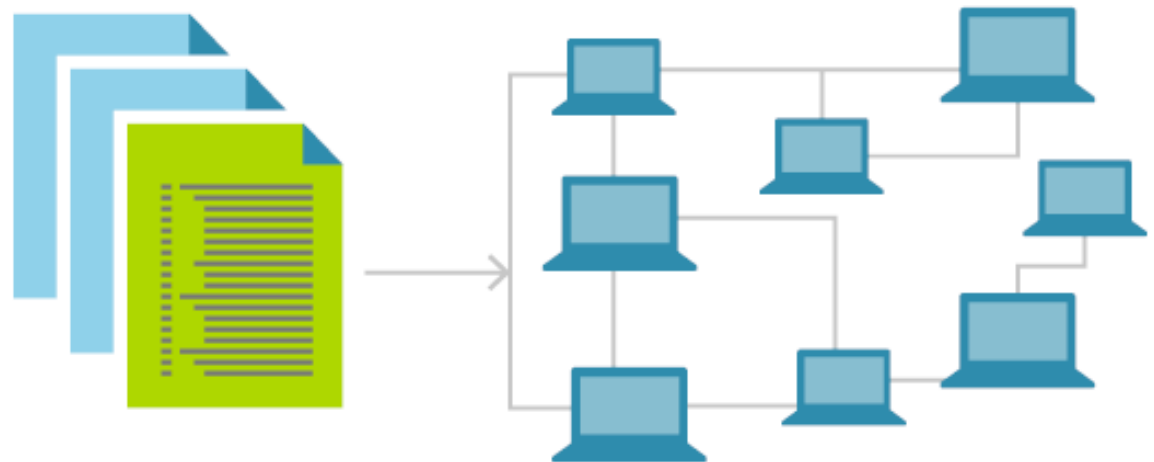
Exposed services/APIs



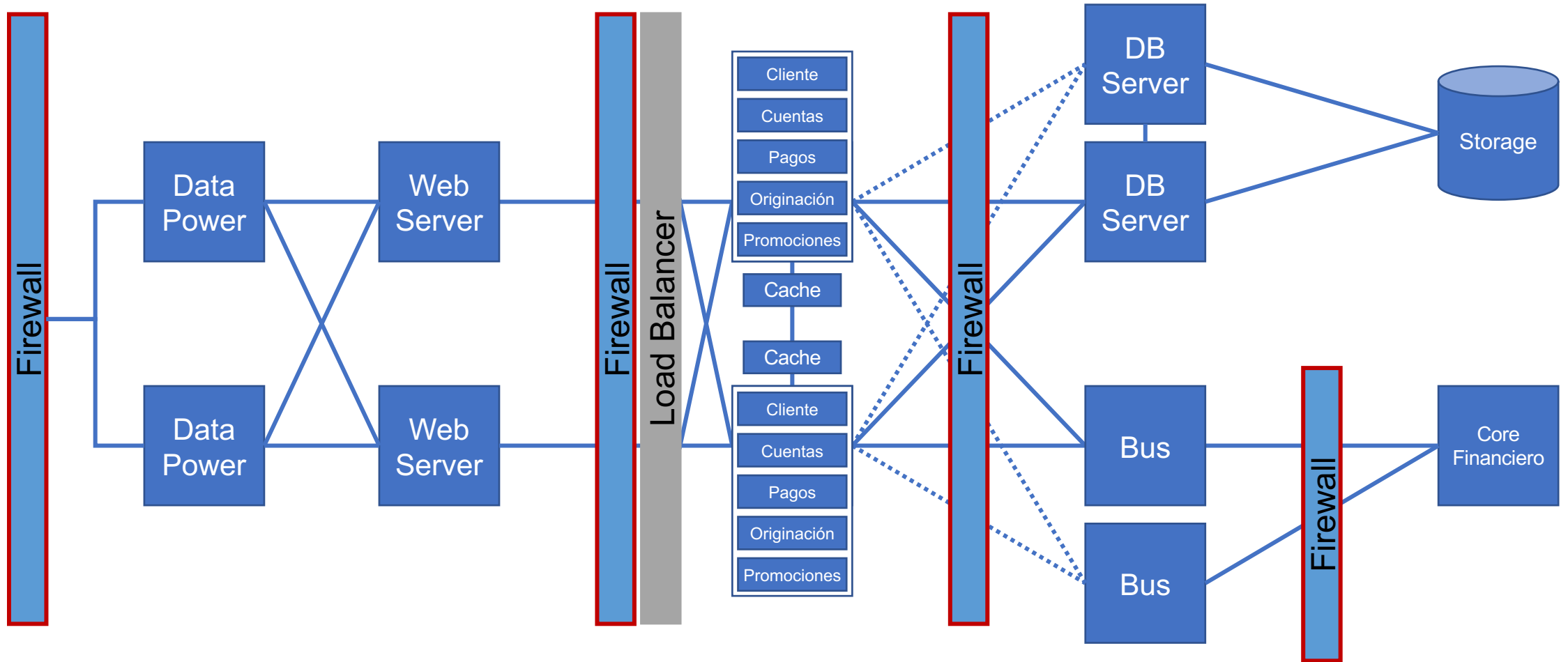
Microservices application

Infrastructure as a Code (IaaS)

Práctica de utilizar **scripts** para **configurar** la infraestructura de una aplicación como máquinas virtuales, **en lugar** de configurar estas máquinas de forma **manual**.



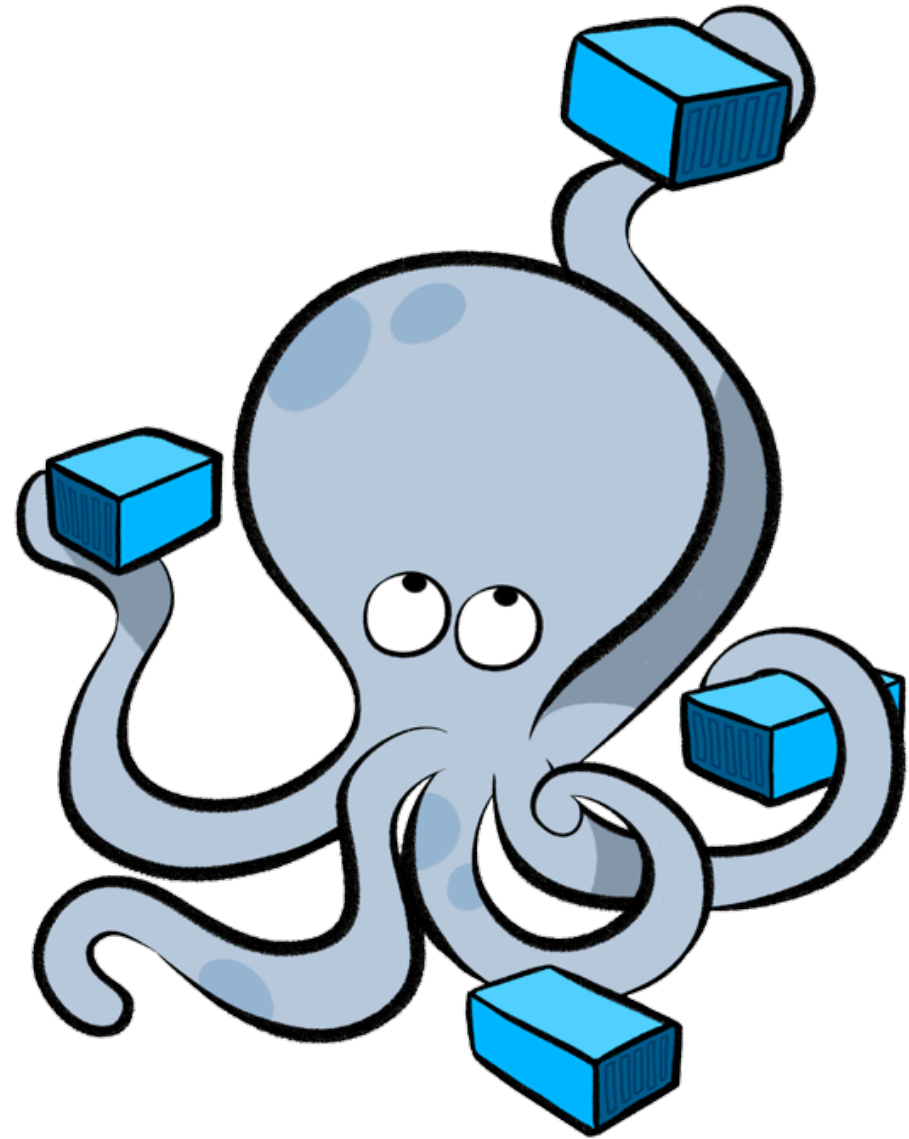
Aplicaciones Multicontenedor



Docker Compose

Es una **herramienta** de Docker que permite definir y ejecutar aplicaciones multi-contenedor.

- `docker-compose build`
- `docker-compose up`
- `docker-compose down`
- `docker-compose logs`



Archivo docker-compose.yml

Define los elementos de una aplicación multi-contenedor.

- Servicios
- Volúmenes
- Redes
- Secretos
- Build & Deploy
- v2 vs. V3

<https://docs.docker.com/compose/compose-file/>

```
version: '3'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```


Demo docker-compose

Comandos de Docker Compose

Tarea	Comando base(*)
Compilar aplicación	<code>docker-compose [-f filename] build</code>
Ejecutar aplicación	<code>docker-compose [-f filename] up</code>
Detener aplicación	<code>docker-compose [-f filename] down</code>
Eliminar aplicación	<code>docker-compose [-f filename] rm</code>
Visualizar logs	<code>docker-compose [-f filename] logs [service]</code>

* Se muestran las opciones y parámetros mas comunes, documentación completa se encuentra en el sitio de docker



Pero hay que estar
preparados para
recibir la agilidad



Pasando de esto a

¡A esto!



Orquestación de Contenedores

Funcionalidades Básicas

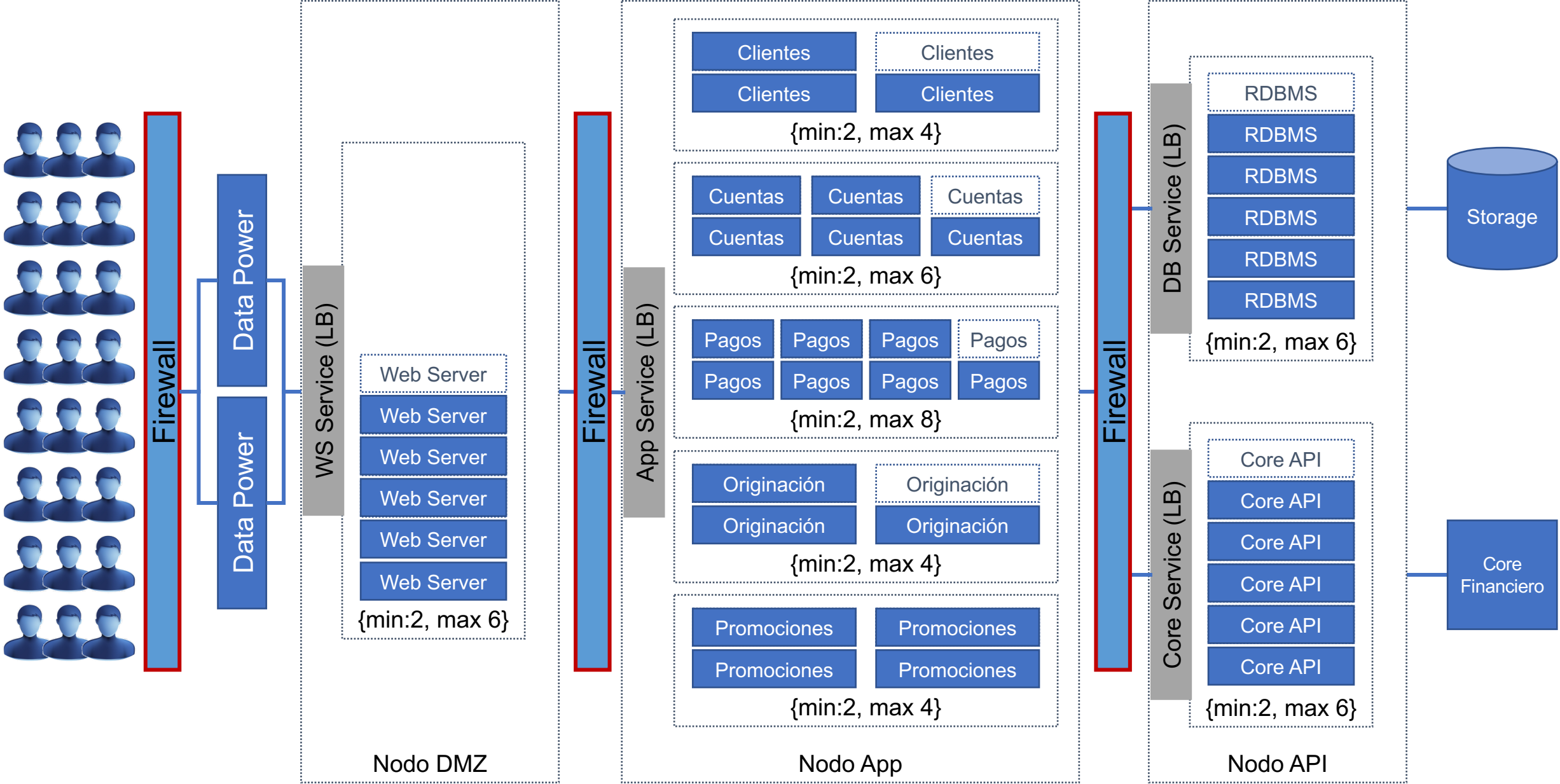
- Distribuir contenedores en un conjunto de máquinas (scheduling)
- Empaquetar aplicaciones
- Replicación
- Manejo de fallas en equipos
- Descubrimiento de servicios en contenedores
- Monitoreo

Funcionalidades Avanzadas

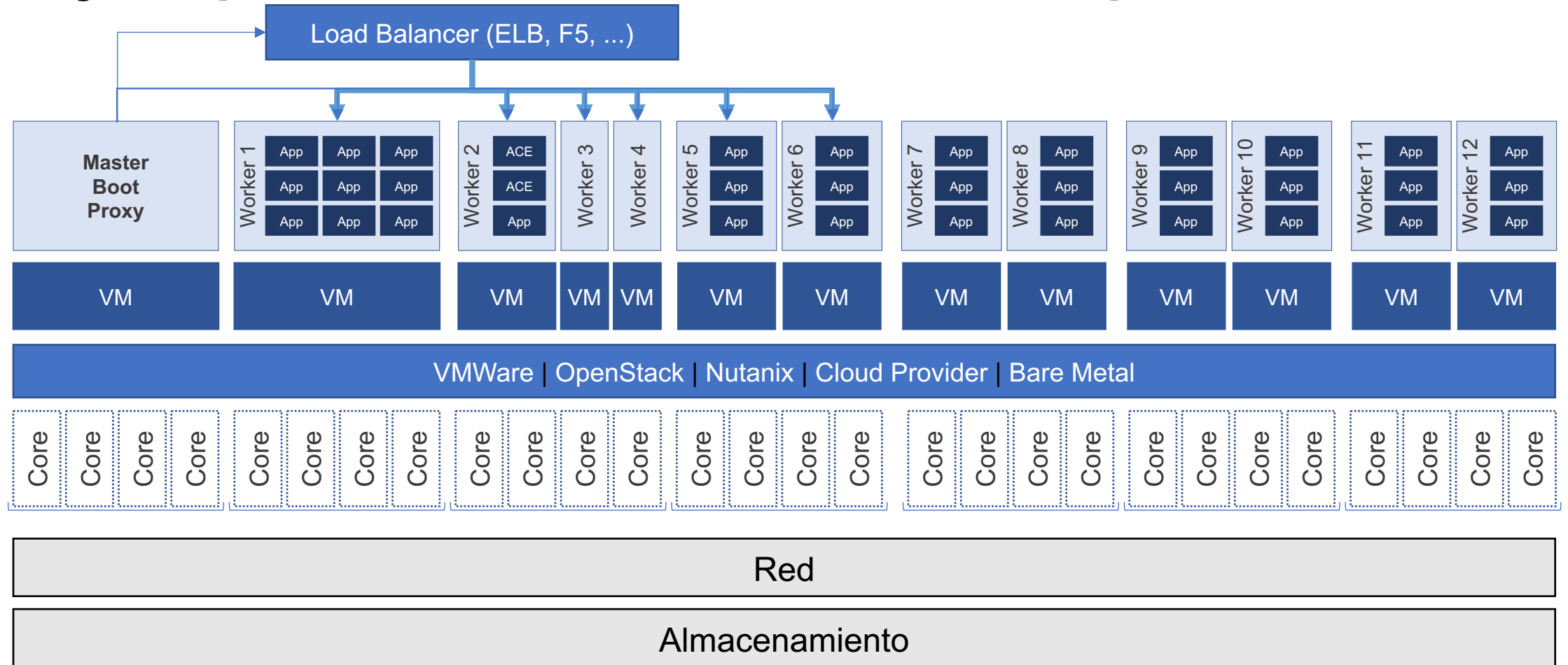
- Balanceo de carga
- Updates automáticos
- Cluster auto-scaling
- App auto-scaling
- Aprovisionamiento del almacenamiento
- Late-binding Configuration



Auto Escalamiento



Ejemplo de instalación de Orquestador



The End
