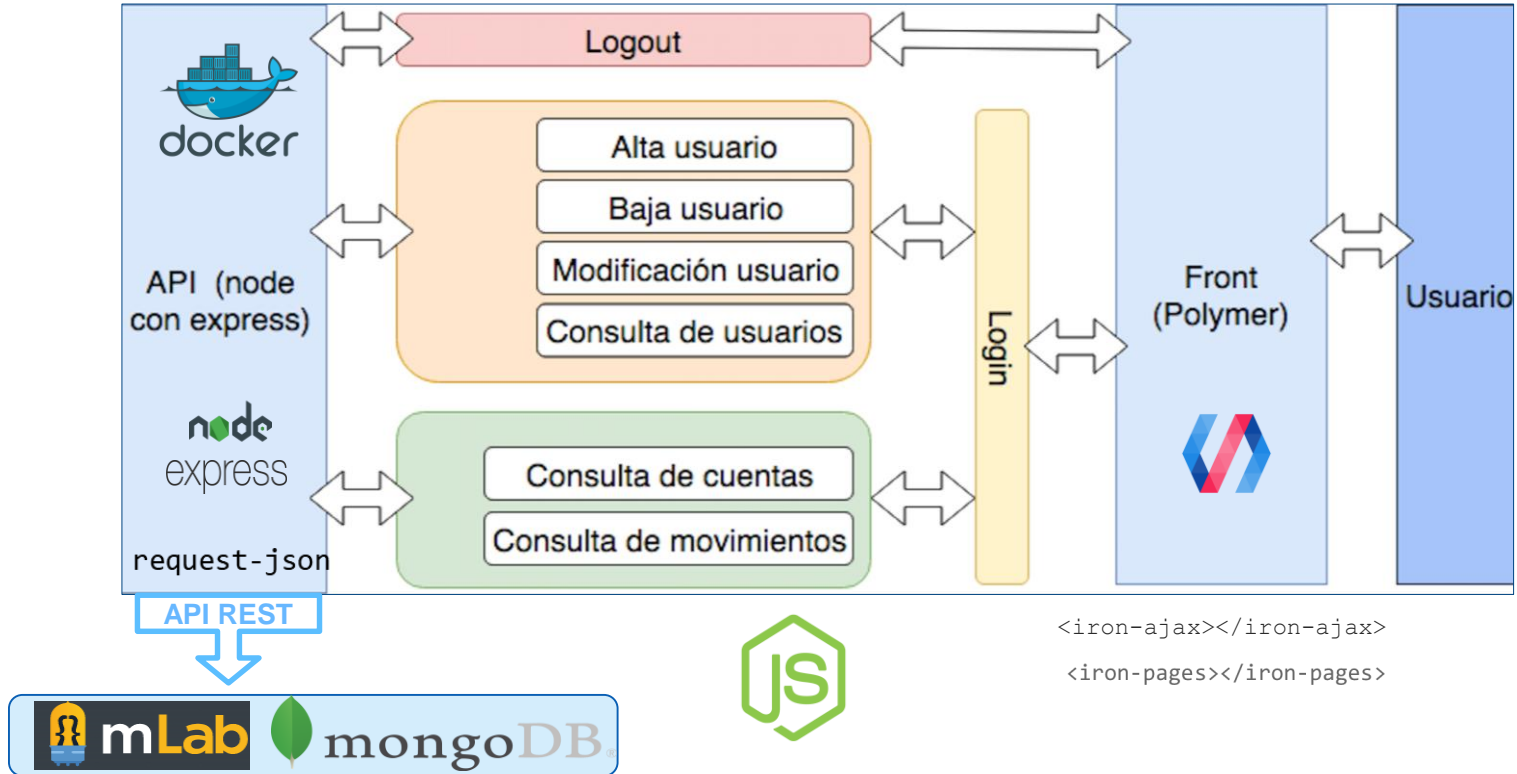




# Arquitectura del Proyecto

## Arquitectura del proyecto



## Descripción del Proyecto

- El proyecto simula una aplicación bancaria con operaciones básicas: login, posición global, consulta de movimientos, transferencias (creación de movimientos en cuenta origen y destino) y logout.
- Como funcionalidad adicional se ha incluido la gestión de cuentas de valores: número de acciones, saldo y gráfica con la serie temporal del valor.

### Back-end

- Servicios web REST implementados en Node JS 6.3.0. Creación del contenedor docker y despliegue en OpenShift.
- Persistencia de la información en MongoDB (mediante llamadas al API REST de MLab)
- Cuentas de valores y serie temporal de acciones: se han utilizado las APIs gratuitas de Quandl (<https://www.quandl.com/>).

### Front-end

- Realizado en Polymer 2.0. a partir de look and feel de Polymer Starter Kit
- Llamadas a servicios REST mediante componente <iron-ajax>



# localStorage y sessionStorage

HTML 5



## localStorage vs sessionStorage

localStorage	sessionStorage
Stores data with <b>no expiration</b> date	Stores data <b>only for a session</b> (until the tab/browser is closed)
Gets cleared through JS or Browser cache / Locally Stored Data	Storage limit is larger than a cookie (at least 5 MB). Max 4KB
Changes available for all current and future visits to the site	Persists over page reloads and restores. Opening a new tab/window will initiate a new session.
	Data is never transferred to the server*
<ul style="list-style-type: none"><li>• Both extend Storage</li><li>• Both can only be read on client-side</li><li>• Web storage is per origin (per domain and protocol)</li><li>• Both allow to storage JS primitive objects (integers, strings, ...) but not arrays neither objects → we must convert them to JSON</li></ul>	



## localStorage vs sessionStorage

Browser Support	version
Google Chrome	4.0
Internet Explorer	8.0
Mozilla Firefox	3.5
Safari	4.0
Opera	11.5



## localStorage methods

Method	Description
<code>setItem()</code>	Add key and value to local storage
<code>getItem()</code>	Retrieve a value by the key
<code>removeItem()</code>	Remove an item by key
<code>clear()</code>	Clear all storage



## HTML Web Storage Objects

Check browser **support** for localStorage and sessionStorage:

```
if (typeof(Storage) !== "undefined") {  
    // Code for localStorage/sessionStorage.  
} else {  
    // Sorry! No Web Storage support...  
}
```





## IndexedDB API

IndexedDB persistently allows to store objects indexed with a "key"

- It's an asynchronous API
- IndexedDB is object-oriented
- IndexedDB databases store key-value pairs
- It's build on a transactional database model
- It does not use SQL



# Elastic Block Store

EBS

## Elastic Block Store (EBS)

**Volúmenes de SSD de uso general  
(gp2) de EBS**

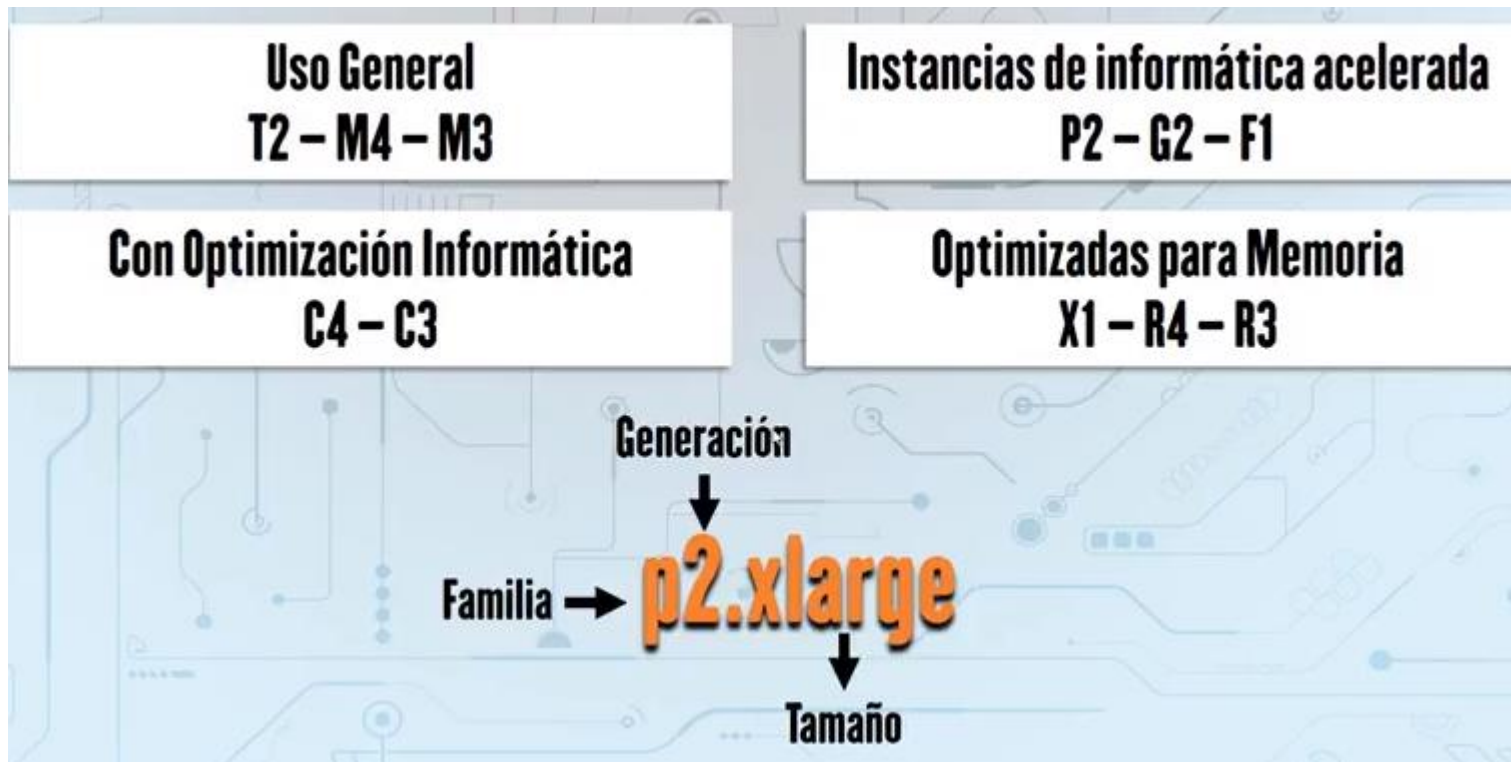
**Volúmenes de SSD de IOPS  
provisionadas (io1) de EBS**

**Volúmenes de HDD optimizados para el  
desempeño (st1) de EBS**

**Volúmenes de HDD frío (sc1) de EBS**

**Snapshots de EBS**

## Tipos de instancias (EC2)



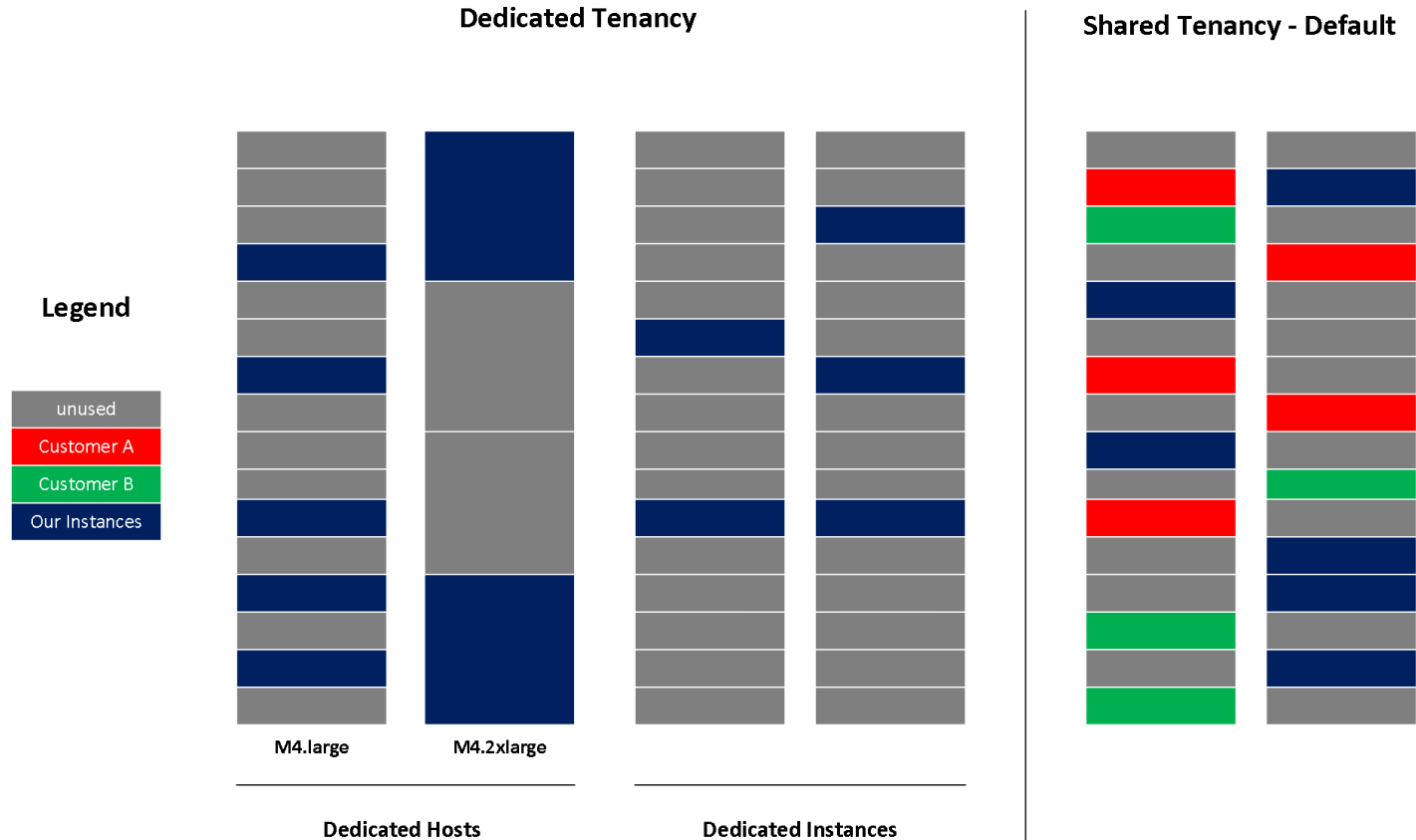


# AWS Tenancy

EC2



## Tenancy - Instancia EC2





# Gestión de Dependencias

NPM y BOWER



## Configuración de dependencias de un proyecto

### NPM - Bower

- Gestores de paquetes/dependencias
- **NPM**, gestor dependencias **back-end**
- **Bower**, gestor dependencias **front-end**
- **package.json** y **bower.json** contienen toda la información necesaria de las dependencias que hay que instalar.





## ¿Qué es el package.json?

### package.json

- Nombre del proyecto.
- Versión.
- Breve descripción.
- Dependencias.
- Repositorio.
- Autores.
- Licencia.
- ...



# Semantic Versioning

SemVer



## Semantic Versioning 2.0.0

Given a version number  
**MAJOR.Minor.patch**, increment the:

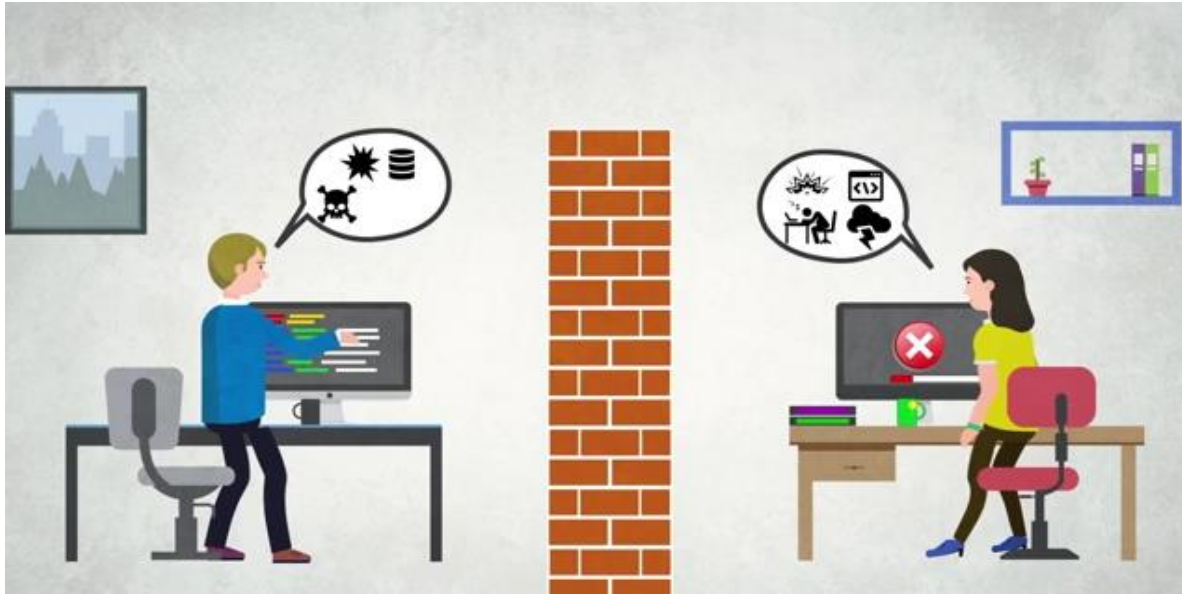
1. MAJOR version when you make incompatible API changes.
2. MINOR version when you add functionality in a backwards-compatible manner.
3. PATCH version when you make backwards-compatible bug fixes.



# DevOps

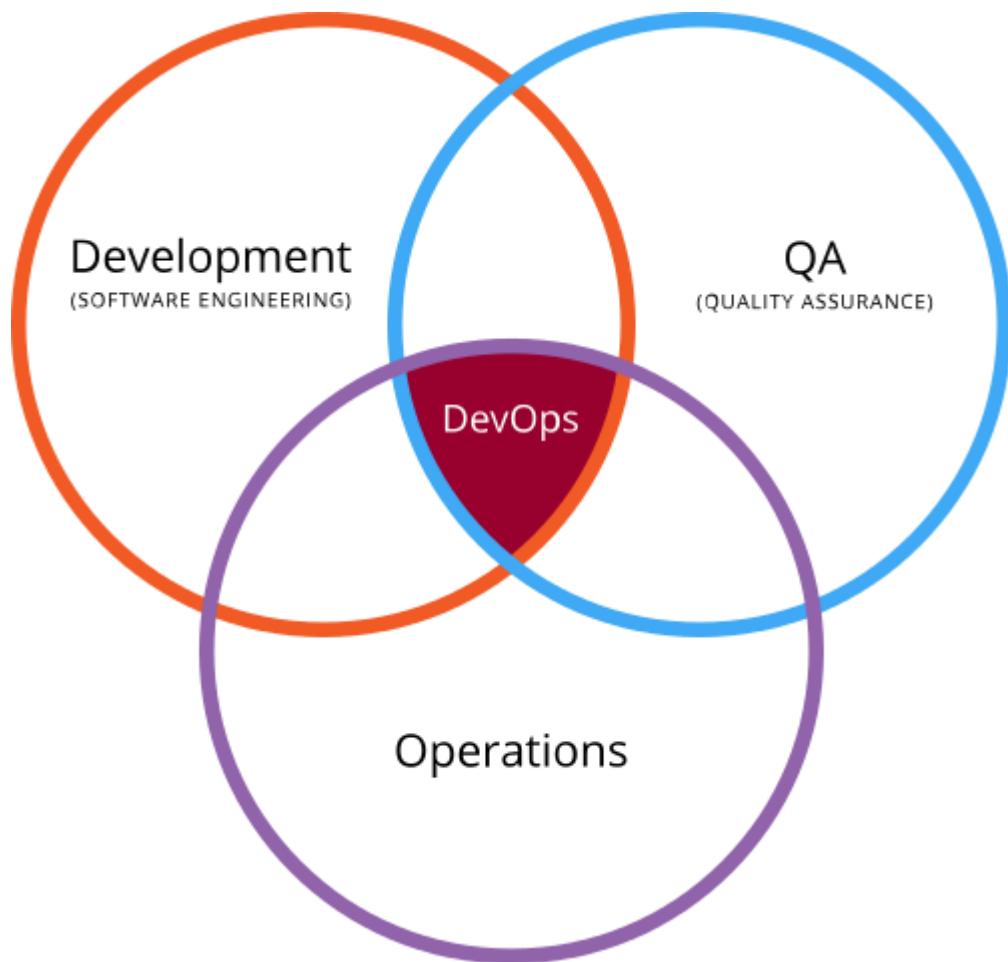
Development + Operations

## Developers vs Operators



# Developers and Operators





# DevOps

## BENEFICIOS



Mayor Flexibilidad

Mejor Calidad

Incremento Productividad

Disminución Costes

Alta Frecuencia de Lanzamientos



## ¿Qué es DevOps exactamente?

### What is DevOps Exactly ?

Phase 1: Automated Testing



Phase 2: Continuous Integration



Phase 3: Continuous Delivery



### Multi-Cloud!



deploy

operate

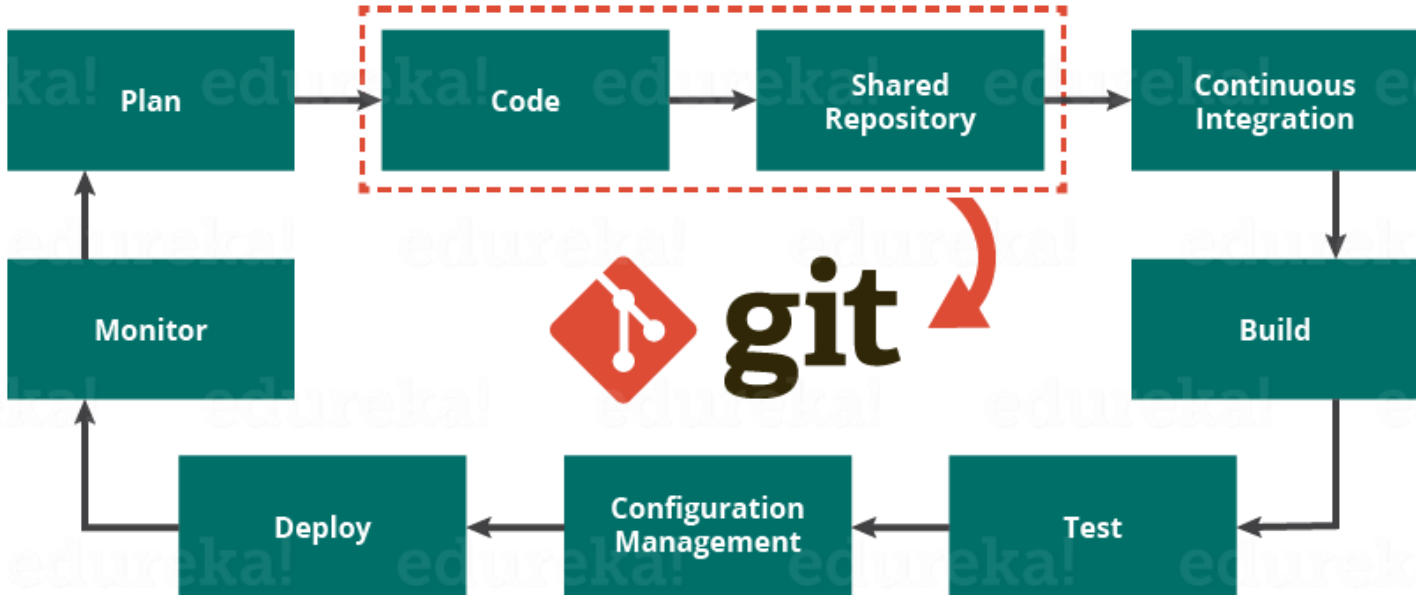




## Herramientas DevOps

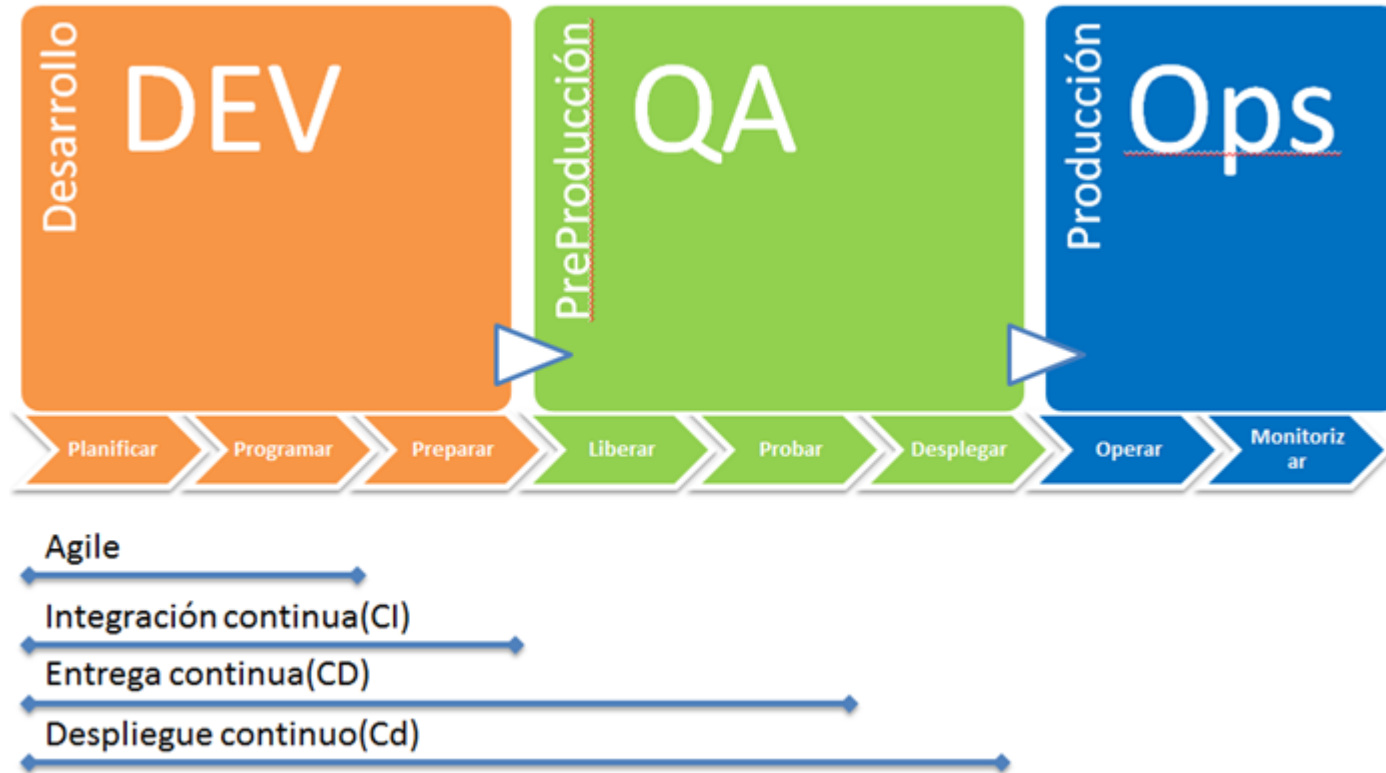
- Sistemas de **Control de Versiones**
- Herramientas de **Empaquetado**
- Orquestación de **Escenarios** (entornos replicables)
- **Automatización** de la Configuración
- **Contenedores**

## ¿Dónde encaja Git en DevOps?



## Ciclo de Vida DevOps



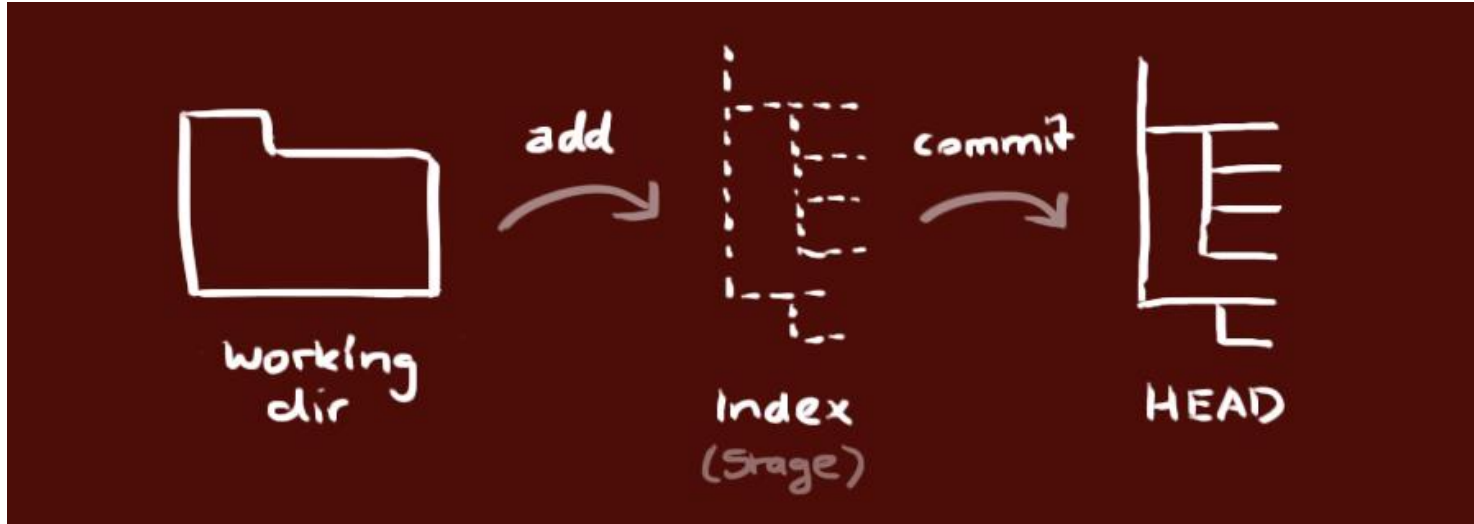




# Git

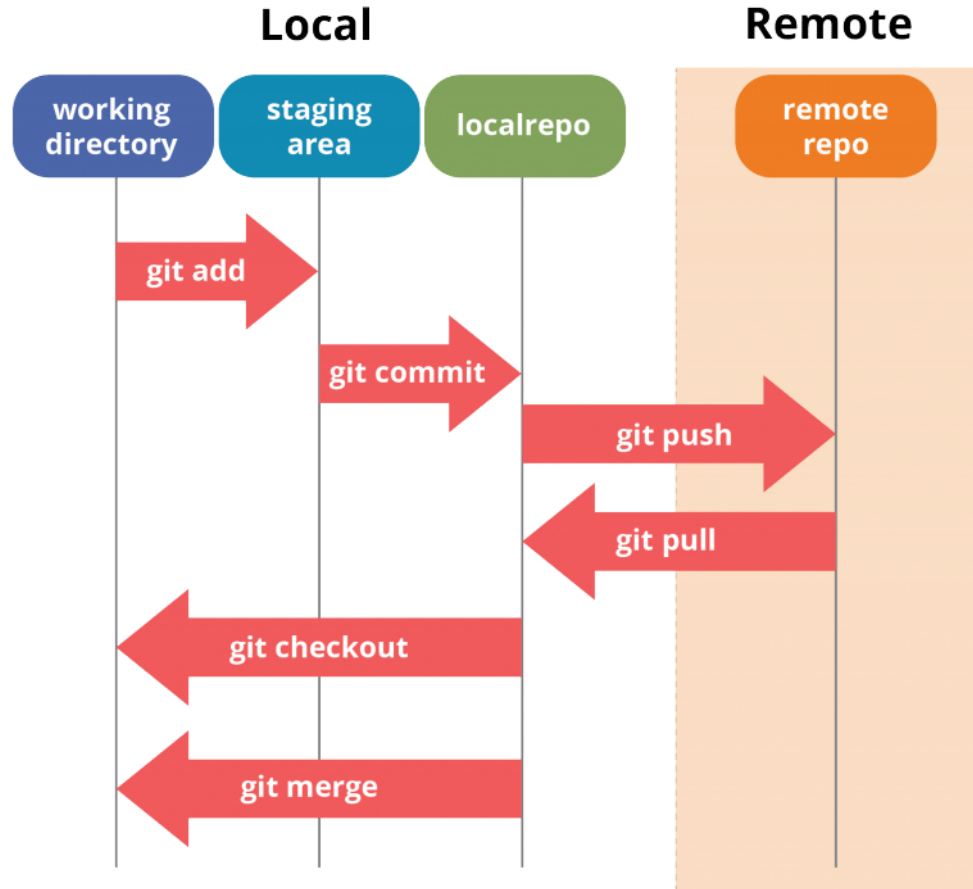
Arquitectura

## Flujo de Trabajo GIT





## Git architecture





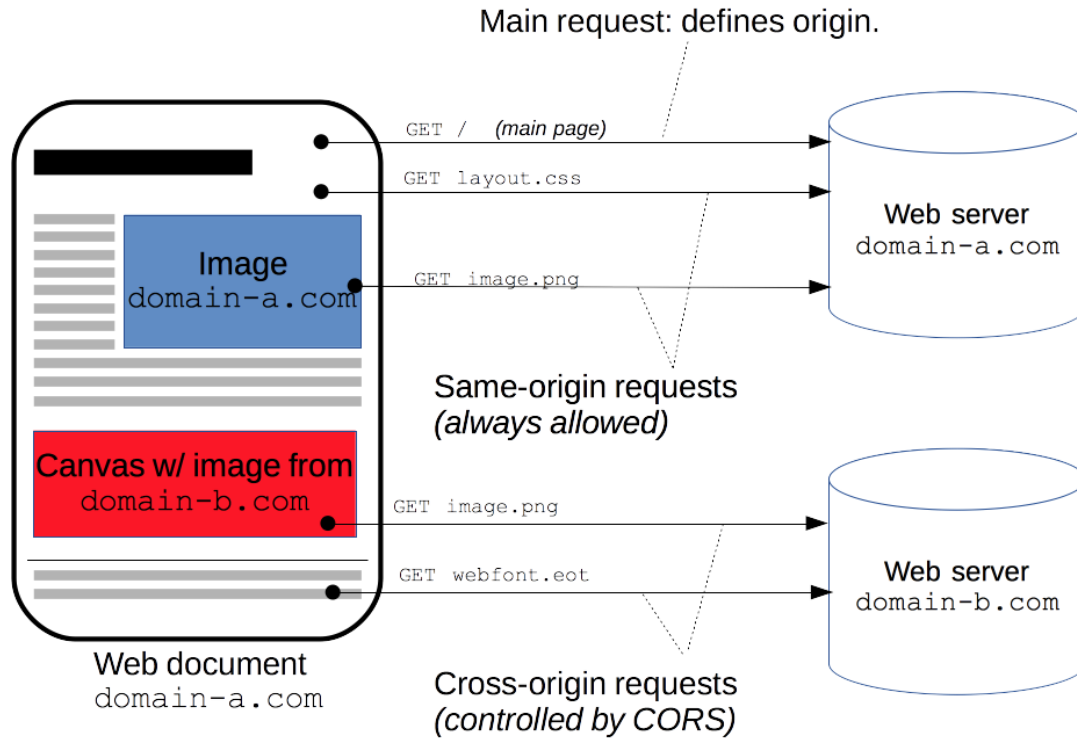


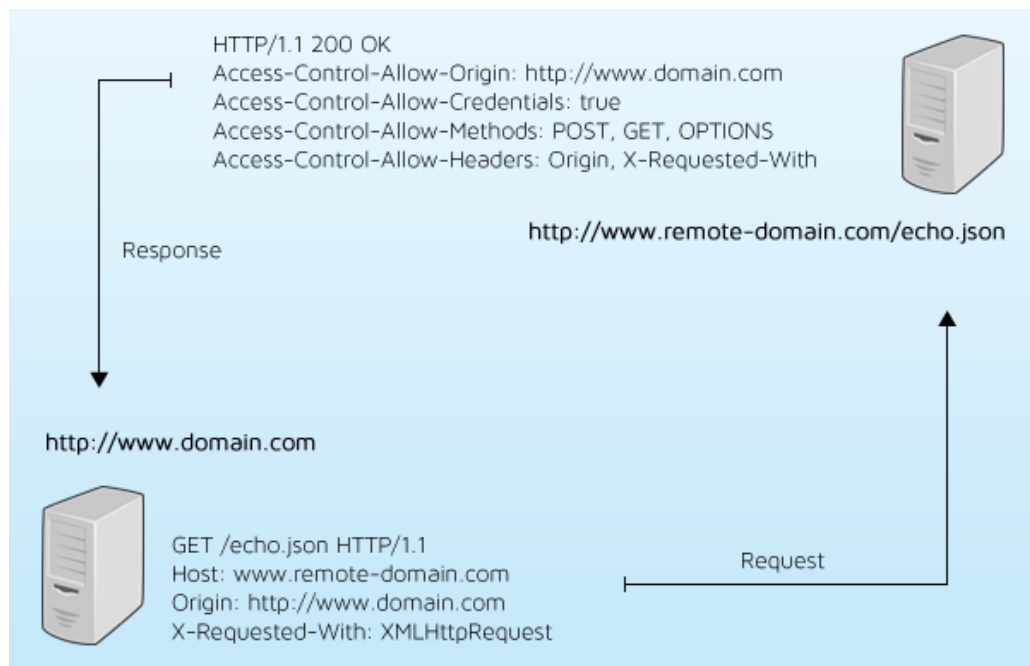
# CORS

Cross-Origin Resource Sharing

## CORS

- El **modelo de seguridad** de las aplicaciones web **no permite** en principio realizar **peticiones** asíncronas **entre dominios**
- Problema: una página de `http://localhost` no puede hacer una petición AJAX a Google
- Solución: CORS añade funcionalidades nuevas a las peticiones AJAX como las **peticiones entre dominios (cross-site)**



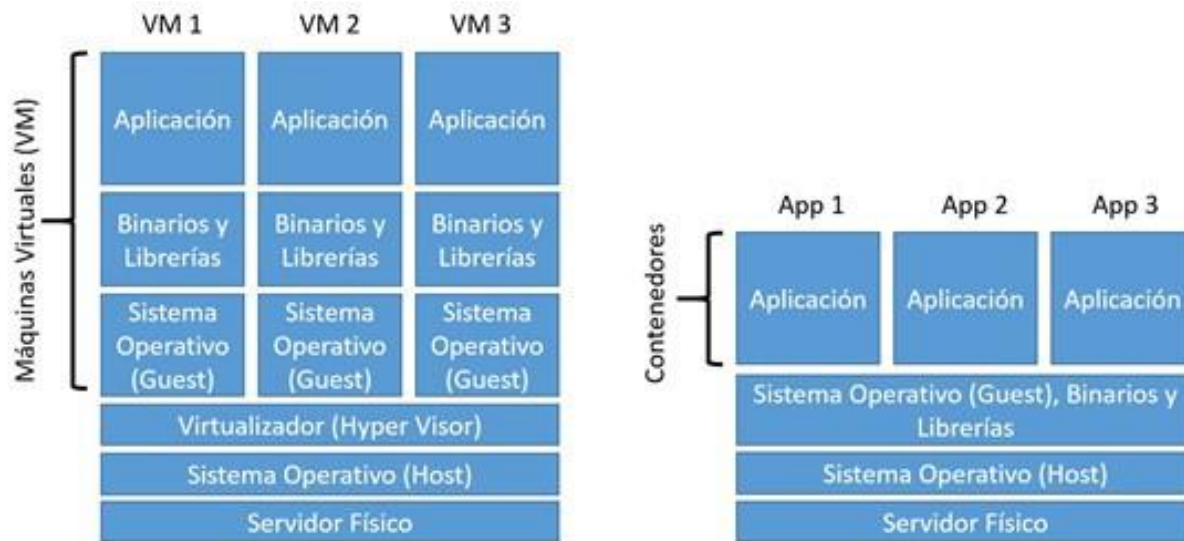




# Contenedores

Docker

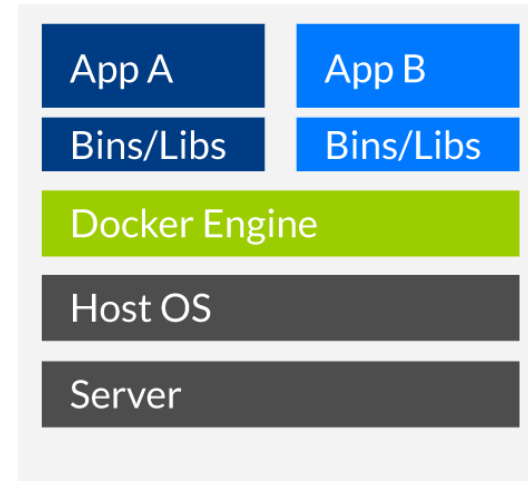
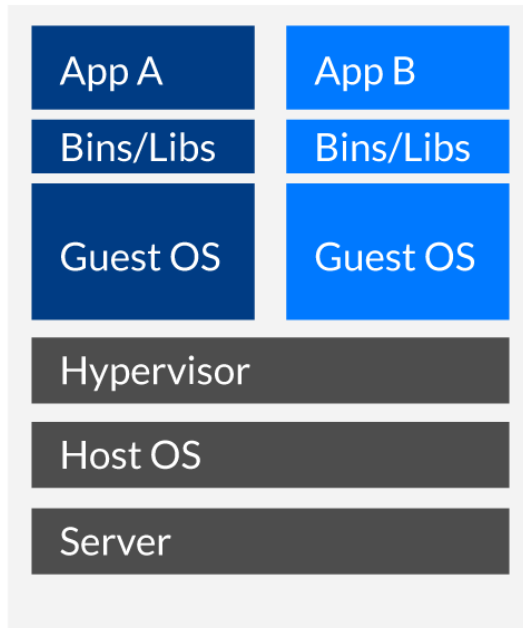
# Contenedores vs Máquinas Virtuales

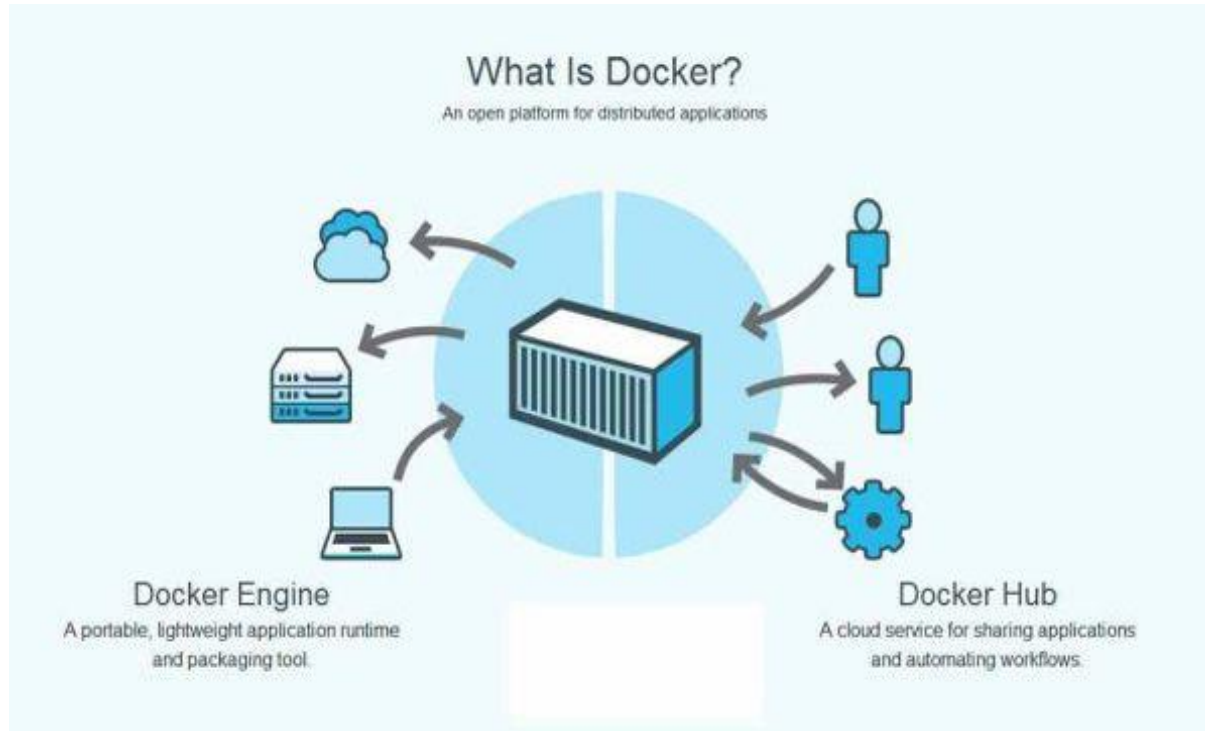


Máquinas Virtuales versus Contenedores

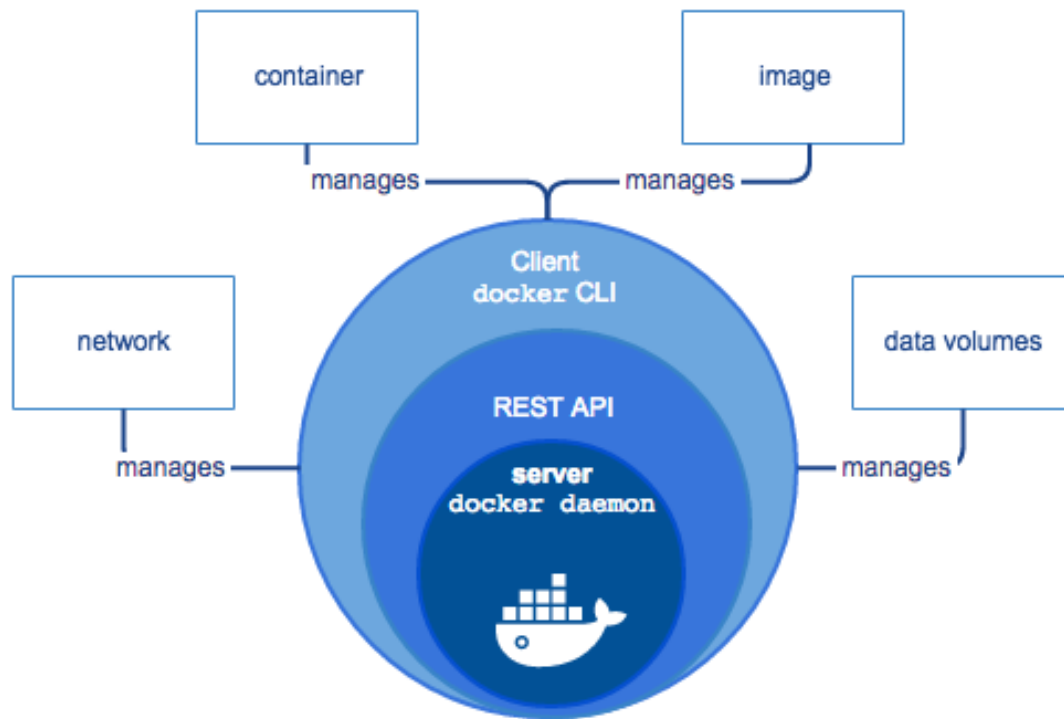


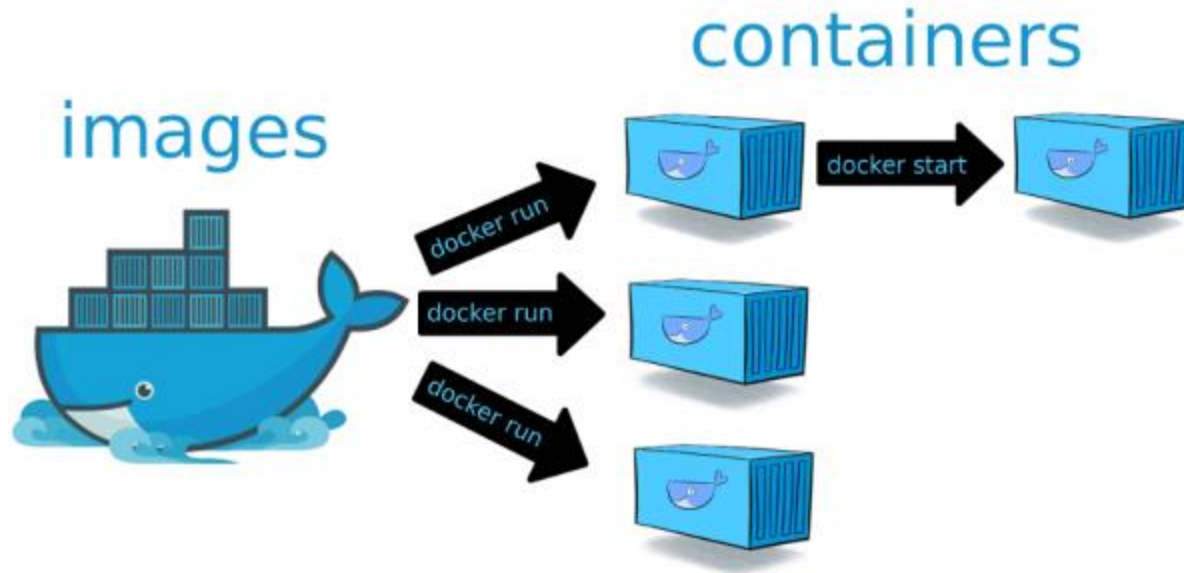
# VMs vs Docker

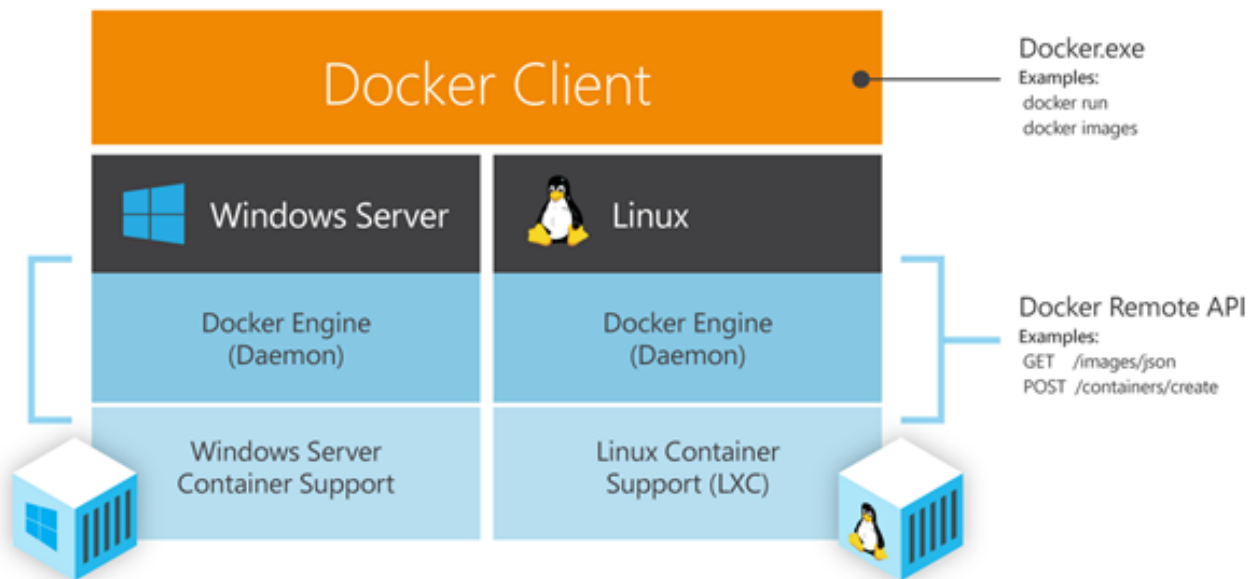














# Principio FIRST

Cómo mejorar la calidad de nuestros tests



## FIRST para tests unitarios

### Características que deben tener Test Unitarios

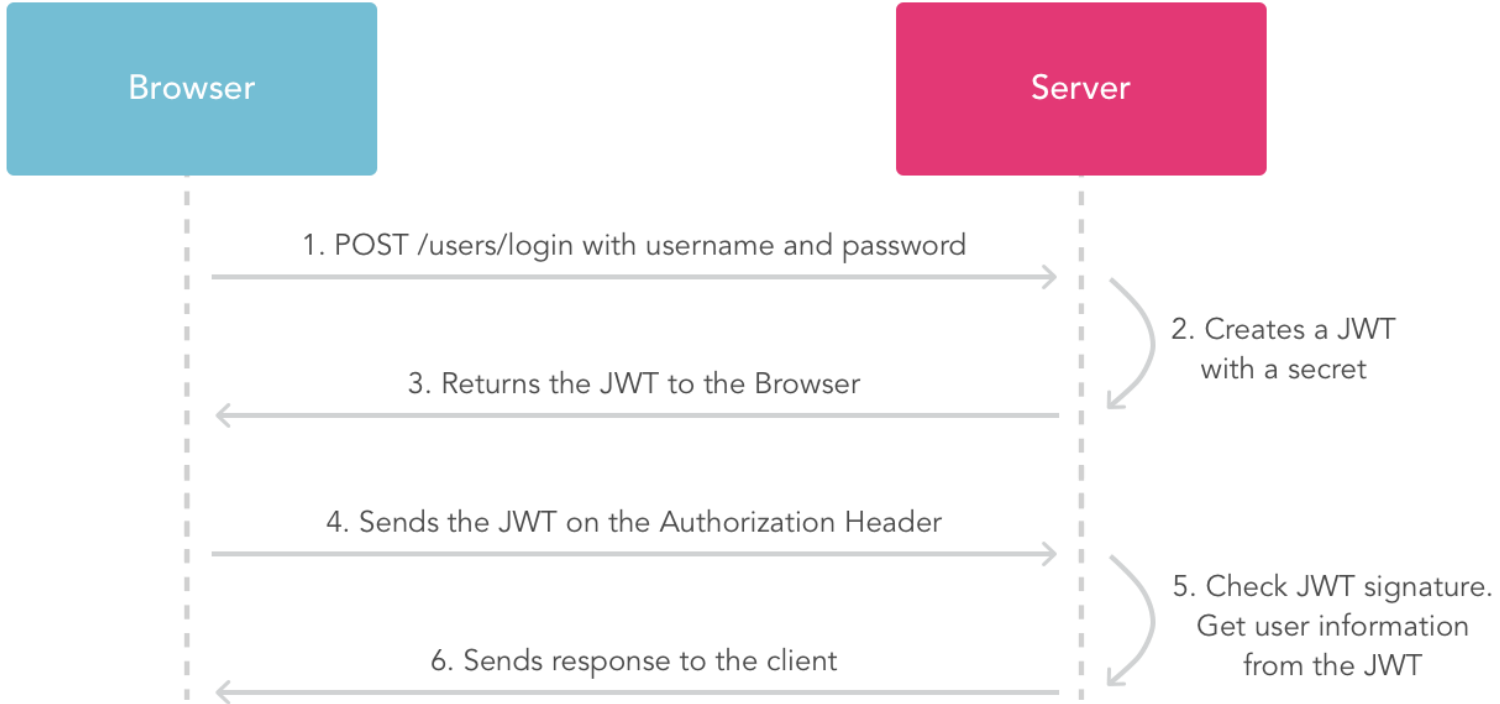
- Fast (rápido)
- Independent (independiente)
- Repeatable (repetible)
- Self-validating (auto evaluable)
- Timely (oportuno)



# JWT

JSON Web Token (RFC 7519)

## JSON Web Tokens





## decoding, verifying, and generating JWTs

ALGORITHM

HS256

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RmHrHdCEfxjoYZgeFONFh7HgQ
```

Decoded

HEADER:

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD:

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

HMACSHA256(  
base64UrlEncode(header) + "." +  
base64UrlEncode(payload),  

secret

  
) ☐ secret base64 encoded

✔ Signature Verified

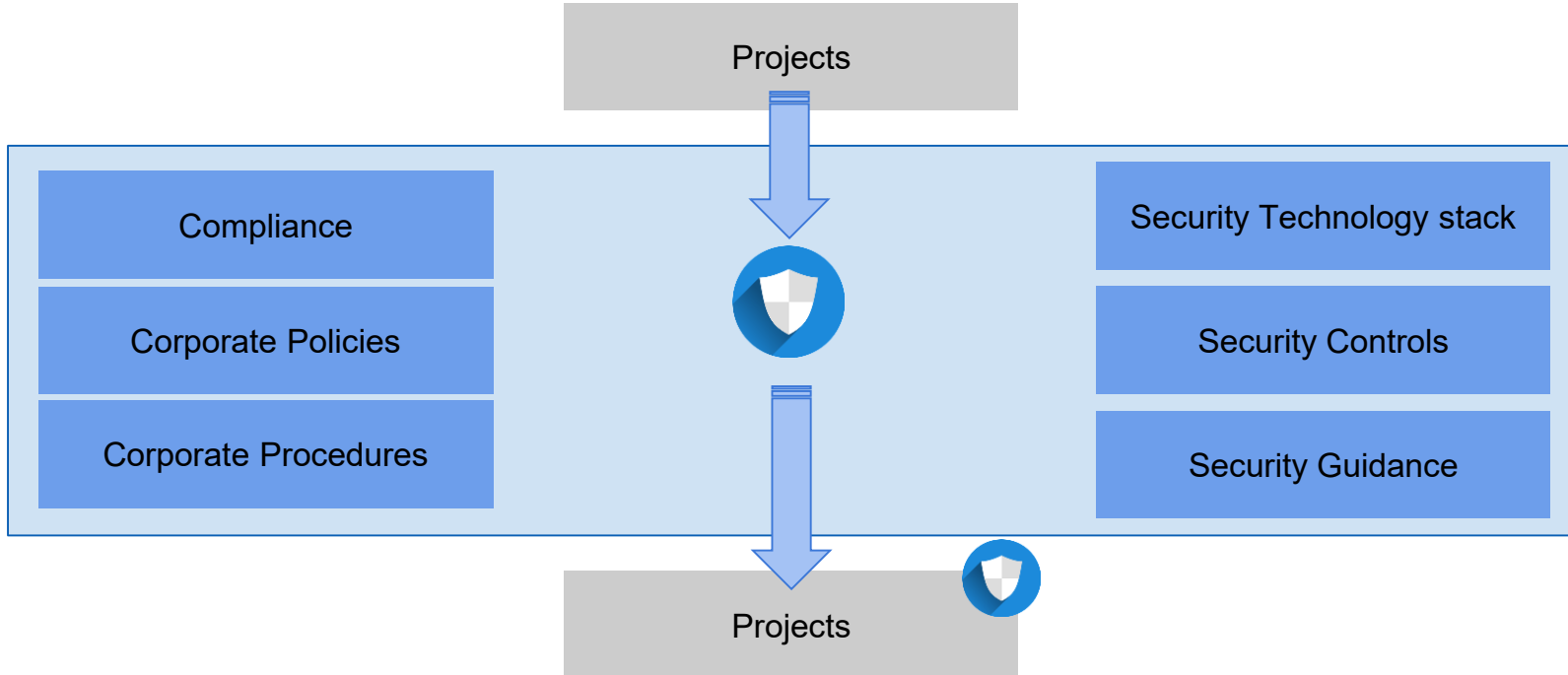




# Seguridad

Desarrollo de aplicaciones seguras

## Security in Projects





# Polymer

Desarrollo de web components



## Polymer 2

### Herramientas de desarrollo

- Bower (sustituído por **Yarn**)
- Polymer CLI

## Qué nos ofrece Polymer 2?

### Funcionalidades del Core

- Renderización del template en el **Shadow DOM**
- Propagación del estado
  - **Binding** de 1 dirección `[[ ]]`
  - **Binding** de 2 direcciones `{{ }}`
- Definición declarativa de manejadores de eventos personalizados
- Sincronización de los atributos (en la etiqueta del componente) a propiedades (dentro del propio componente)