

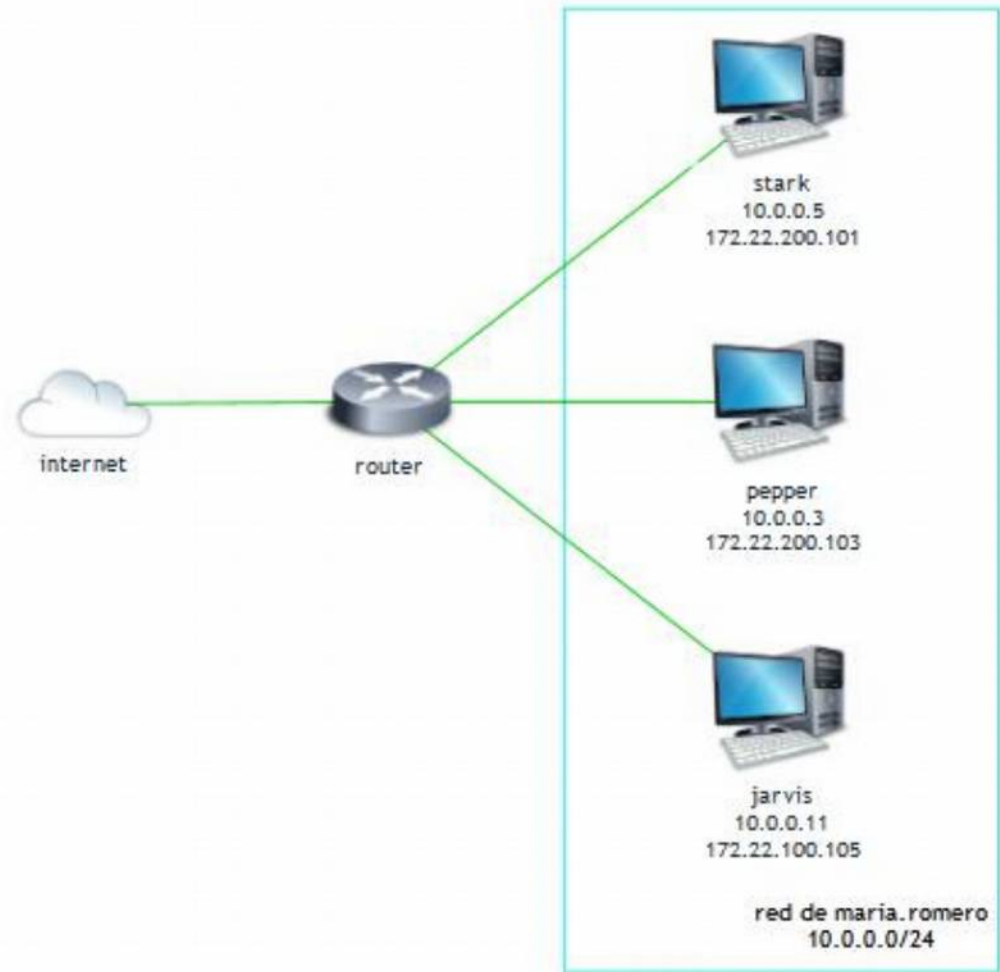
# CouchBase

María Romero Angulo

Proyecto de fin de ciclo

IES Gonzalo Nazareno - 2016/2018

# ENTORNO



# ¿QUÉ ES COUCHBASE?

- Base de datos no relacional
- Tipo mixto: clave-valor y documental
- Distribuida
- Alto rendimiento
- Muy escalable
- De propósito general
- Permite realizar aplicaciones en cualquier lenguaje

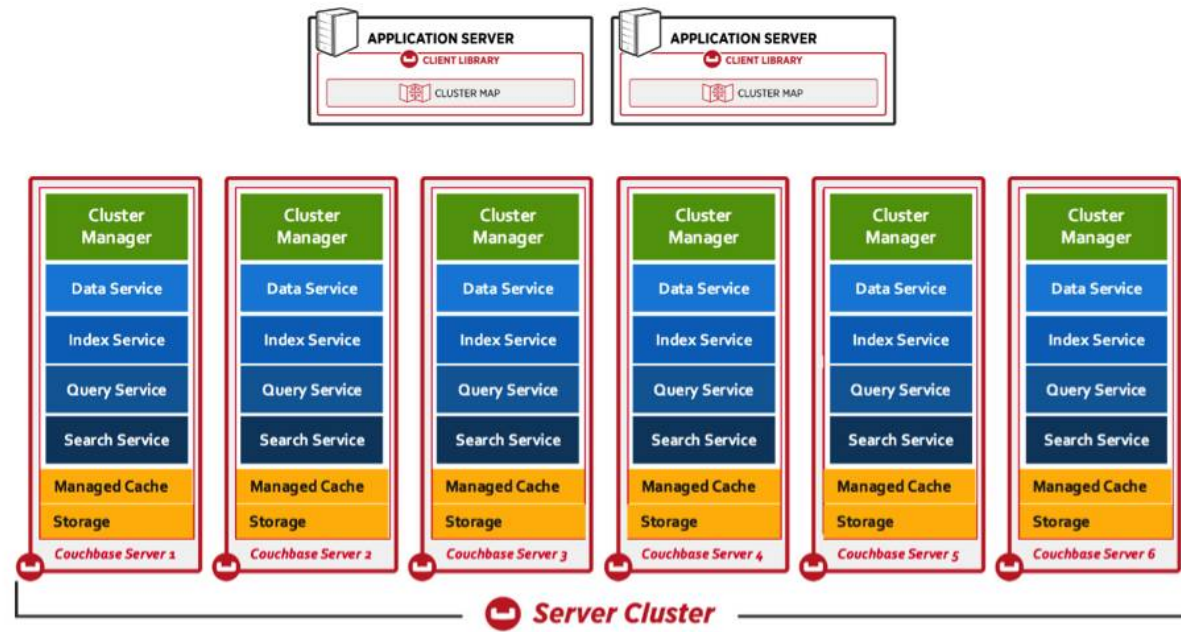
# SQL vs NoSQL

SQL	NoSQL
Integridad referencial	Arquitectura muy diferente
Gran madurez	Falta de soporte comercial
Bastantes estándares de uso	Falta de compatibilidad
Evita la duplicidad de registros	Hay que tener mucho cuidado con la duplicidad
Bajo rendimiento	Alto rendimiento
Baja escalabilidad	Escalabilidad horizontal
Problemas a la hora de manejar grandes cantidades de datos	Pensadas para grandes cantidades de datos

# ¿QUIÉN US A COUCHB ASE?

- **AOL** para publicidad
- **PayPal** para sus estadísticas en tiempo real
- **Tesco** como base de datos de todos sus productos
- **Ryanair** para su página web

# ESTRUCTURA



- Application server: jarvis
- Couchbase server 1: stark
- Couchbase server 2: pepper

- Principales

- connectivity arch
  - autenticación, discovery, conexión
- replication arch
- storage arch
  - couchstore, forestDB
- caching layer arch
  - cuotas en ram
- security arch

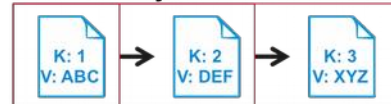
- Secundarios

- cluster manager
- data service
- index service
- query service

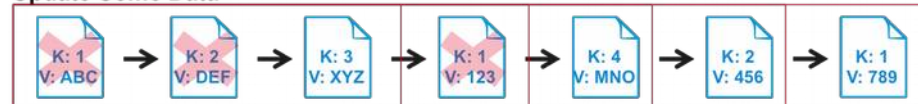
# STORAGE ARCH

Couchstore	ForestDB
Clave valor	Documental
Un documento por bucket	Un documento por índice
Append-only	Append-only y circular reuse
Volumen de datos normal	Grandes cantidades de datos

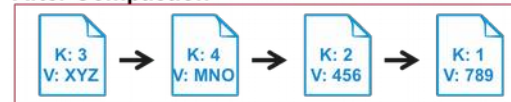
Initial File Layout



Update Some Data

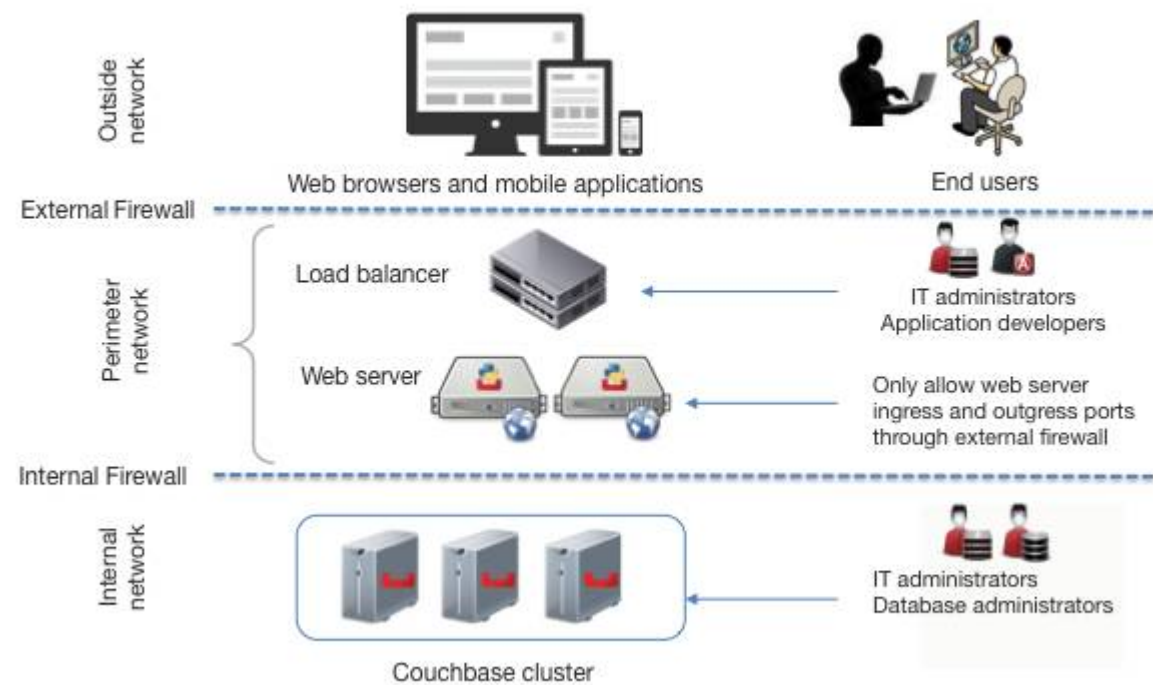


After Compaction





# SECURITY ARCH



# DATA SERVICE

- Los datos se agrupan en buckets
- El acceso desde varias aplicaciones es simultáneo dada su baja latencia y la eficiencia de la cache
- Detecta qué datos son los más consultados y los almacena en RAM, dejando los otros en disco
- Siempre se accede al bucket principal, salvo catástrofe que se consultará a las réplicas
- Almacenado en binario y JSON (más utilizado)

## EJEMPLO DE CONEXIÓN A SERVIDOR

```
root@pepper:/opt/couchbase/bin# ./cbq -u Administrator  
-engine=http://172.22.200.103:8091/
```

N1QL

- Al igual que en SQL, tenemos dos grupos: DDL Y DML
- Se pueden incluir expresiones con DISTINCT, ORDER BY, EXISTS y expresiones de cadena de SQL (INITCAP, LTRIM, SUBSTR,...)
- Para obtener los elementos con cierto campo vacío, se usa IS MISSING
  - ```
SELECT name, brewery_id FROM `beer-sample` WHERE brewery_id IS MISSING;
```
- Para borrar un registro concreto
  - ```
DELETE FROM `beer-test` WHERE name = "563 Stout";
```
- Modificar una tabla existente
  - ```
UPDATE `beer-sample` SET name = 'deleted' WHERE brewery_id = (SELECT brewery_id FROM `beer-sample` WHERE brewery_id IS MISSING);
```
- Inserción de datos
  - ```
INSERT INTO `travel-sample` ( KEY, VALUE ) VALUES ( "k001", { "id": "01", "type": "airline"} ) RETURNING META().id as docid, *;
```

# IMPORTACIÓN Y EXPORTACIÓN

- Se pueden utilizar varios formatos de datos, pero el más común y cómodo es JSON.
- Exportación: `cbexport json`
- Importación: `cbimport json`
- Formato de datos: líneas, lista, muestra (es como un zip solo compatible con `cbimport`)

```
{"key": "mykey1", "value": "myvalue1"}  
{"key": "mykey2", "value": "myvalue2"}  
{"key": "mykey3", "value": "myvalue3"}  
{"key": "mykey4", "value": "myvalue4"}
```

```
[  
  {"key": "mykey1", "value": "myvalue1"},  
  {"key": "mykey2", "value": "myvalue2"},  
]
```

# RÉPLICAS

- En un cluster hay dos tipos de datos: activos (escritos directamente en ese nodo) y de réplica.
- En caso de saturación: mensaje backoff
- Hay dos tipos de réplica: la "normal" y XDCR.

<b>Normal</b>	<b>XDCR</b>
Si un nodo cae, se mantiene la información, pero no se pueden hacer consultas	Si se cae un nodo, la consultas se hacen contra el siguiente nodo activo de la lista
Para configurarla, hay que instalar el paquete y agregar un servidor principal	Se configura como un servidor independiente y en el primario se añade como XDCR
Para entornos de prueba	Para entornos reales
Pensado para red local	Permite servidores por todo el mundo



## Couchbase > Join Cluster

### Cluster Host Name/IP Address

### Cluster Admin Username

### Cluster Admin Password

### ▼ Configure Services & Settings For This Node

- Data Service
- Index Service
- Search Service
- Query Service

### This Node: Host Name/IP Address

Usually localhost or similar

### Data Disk Path

Path cannot be changed after setup

Free: 17 GB

### Indexes Disk Path

Path cannot be changed after setup

Free: 17 GB

< Back

Join With Default Configuration

### Add Remote Cluster ✕

**Cluster Name**

**IP/Hostname** ⓘ

**Username for Remote Cluster**

**Password**

Enable TLS Encryption ⓘ

### Add Replication ✕

**Replicate From Bucket**

**Remote Cluster**

**Remote Bucket**

Enable advanced filtering

▶ Show Advanced Settings

ncel

shield > XDCR Replications ☰

**Remote Clusters**

name	IP/hostname	
universe	10.0.0.3:8091	Delete Edit

**Ongoing Replications**

bucket	protocol	from	to	filtered	status	when
beer-sample	Version 2	this cluster	bucket "beer-test" on cluster "universe"	No	Replicating <span>▬▬</span>	Delete Edit



# PROBLEMA S

- Falsos positivos por recursos de los servidores
- Hay tareas que se podrían realizar de forma sencilla pero lo han reservado para la versión de pago
- Hay errores catastróficos que solo pueden ser recuperables con herramientas de la versión de pago

# APLICACIÓN

Mientras se muestra el funcionamiento de la aplicación web se explicará cómo funcionan las copias de seguridad.

FIN

「(ツ)」