

# El Everest es pan comido: Escala fácilmente con Cassandra



PyData

Javier Gómez Santos

Data Alchemist



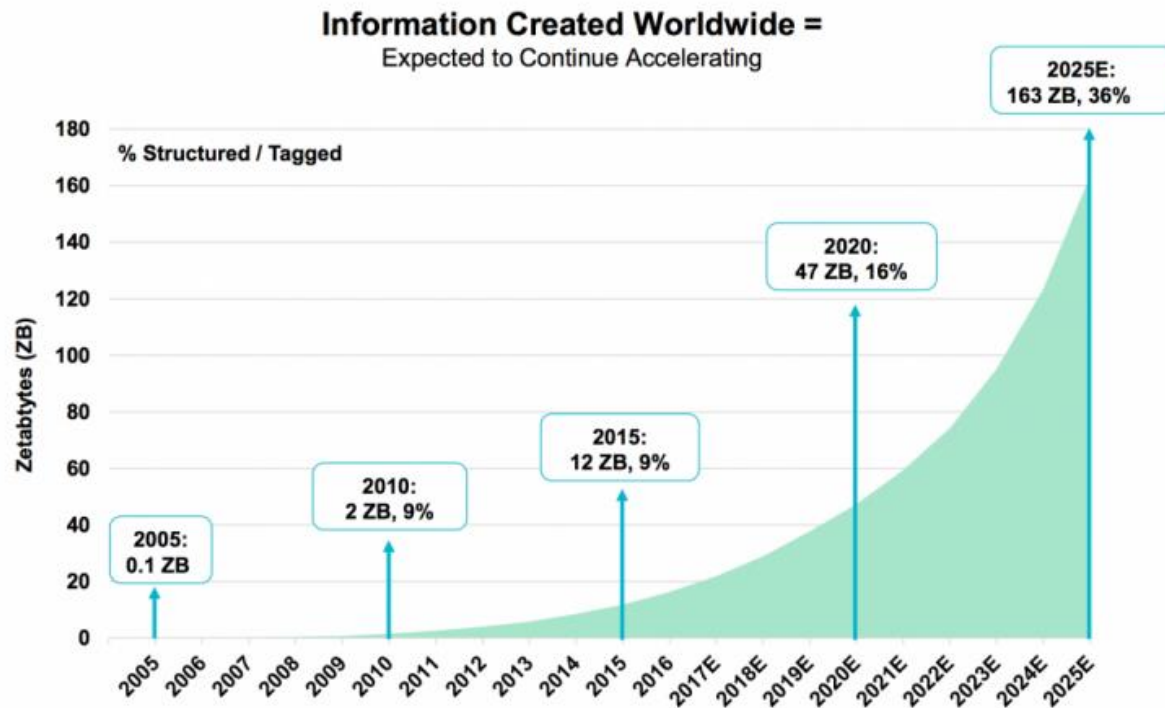
an NTT DATA Company

A low-angle, upward-looking photograph of several modern skyscrapers with glass facades, reaching towards a bright, hazy sky. The perspective creates a sense of height and architectural grandeur.

## Big Data, los orígenes

## Crecimiento de datos en los últimos años

Data Volume Growth Continues @ Rapid Clip...  
% Structured / Tagged (~10%) Rising Fast...

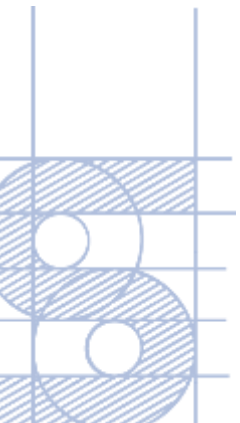


Imposible para gestionar por un sistema tradicional



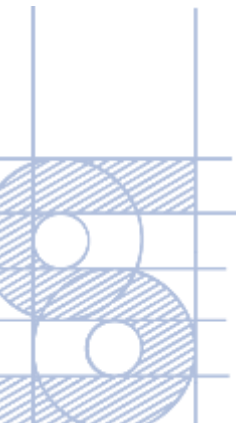
## ¿Qué enfoques aportan las BBDD NoSQL?

- Distribuye la información a través de múltiples nodos
- Es más laxo con la consistencia
- Es más laxo con los esquemas
- Busca optimizar la información para adaptarla a las necesidades actuales



### Hay 4 grandes enfoques a las bases de datos no-relacionales:

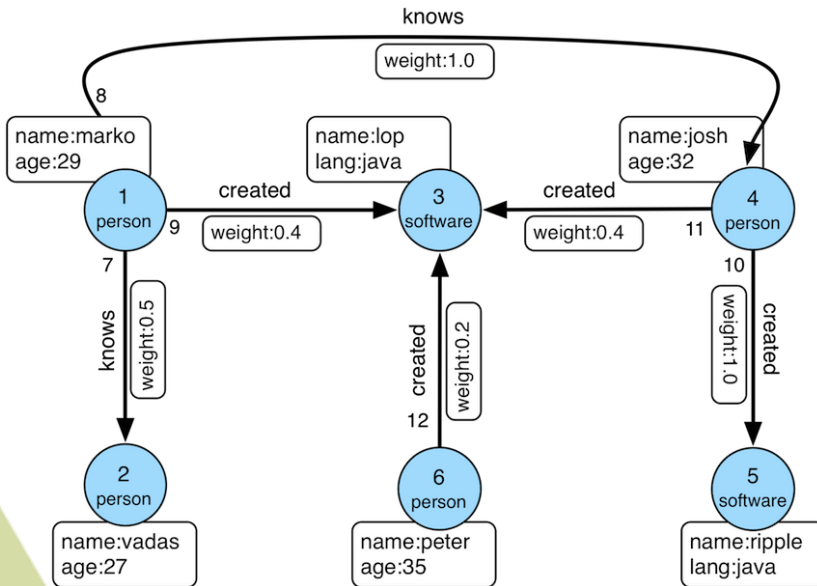
- Grafos
- Clave-Valor
- Documento
- Column Family





## Grafos

Los elementos están distribuidos como vértices de un grafo, siendo las uniones entre vértices las relaciones de ambos (con propiedades).

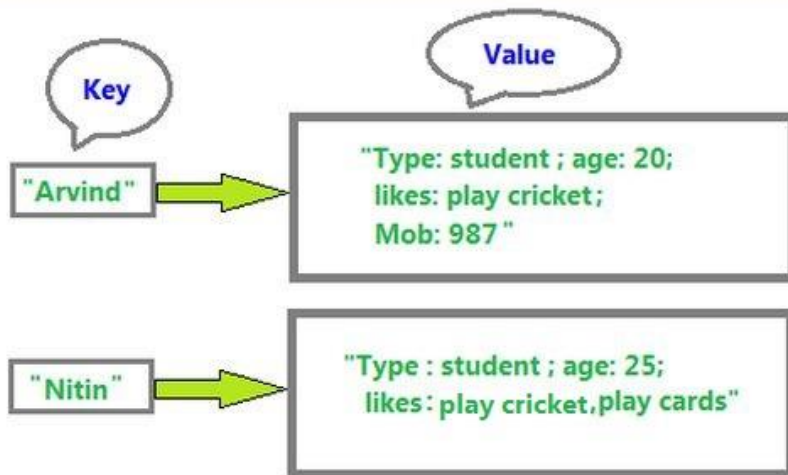


# Gremlin

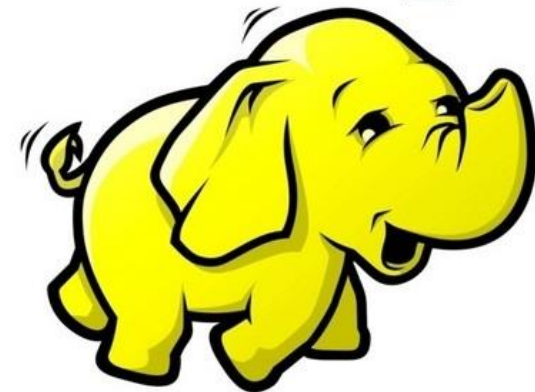
$G = (V, E)$

## Clave-Valor

Existen claves que enlazan a un conjunto de valores asociados a la misma de cualquier tipo



# hadoop





## Documento

Se almacenan conjuntos de documentos (JSON) que pueden ser consultados en su totalidad o en parte.

### Students Document

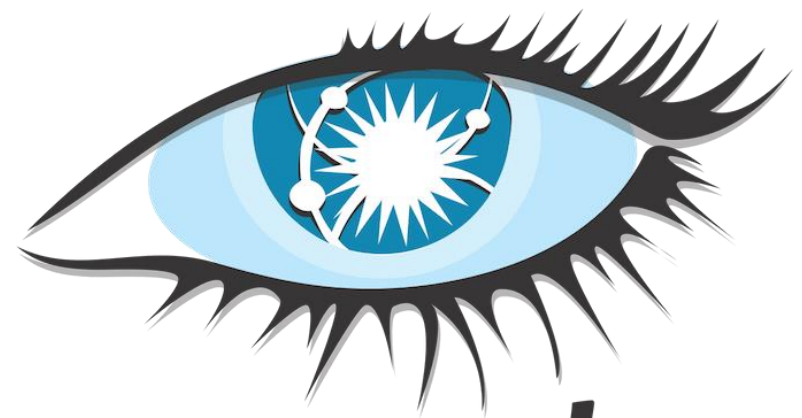
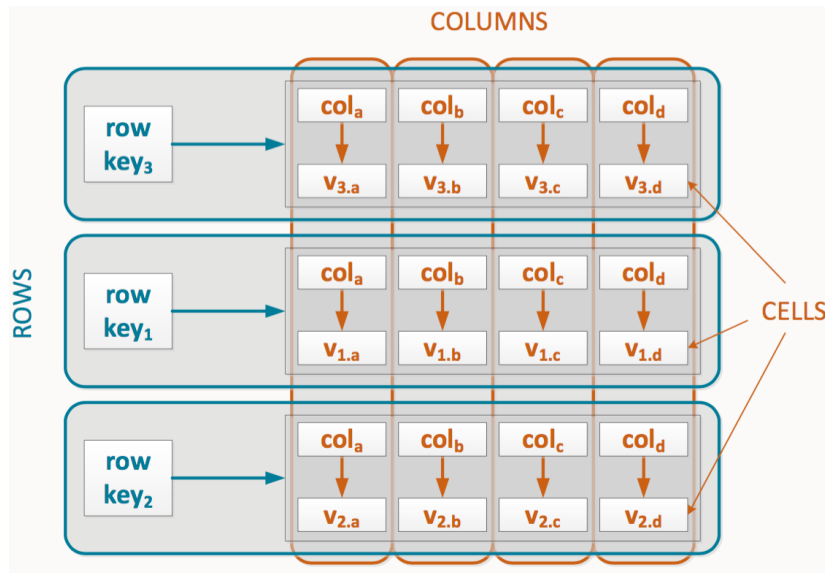
```
{
  _id : "5373aadadcac133aad5b6660",
  name : "kaushal patel",
  email : "kp@example.com",
  courses : {
    course_name : "java",
    fees : "5000",
    duration : "3",
    professor : "g.r."
  }
}
```



# mongoDB®

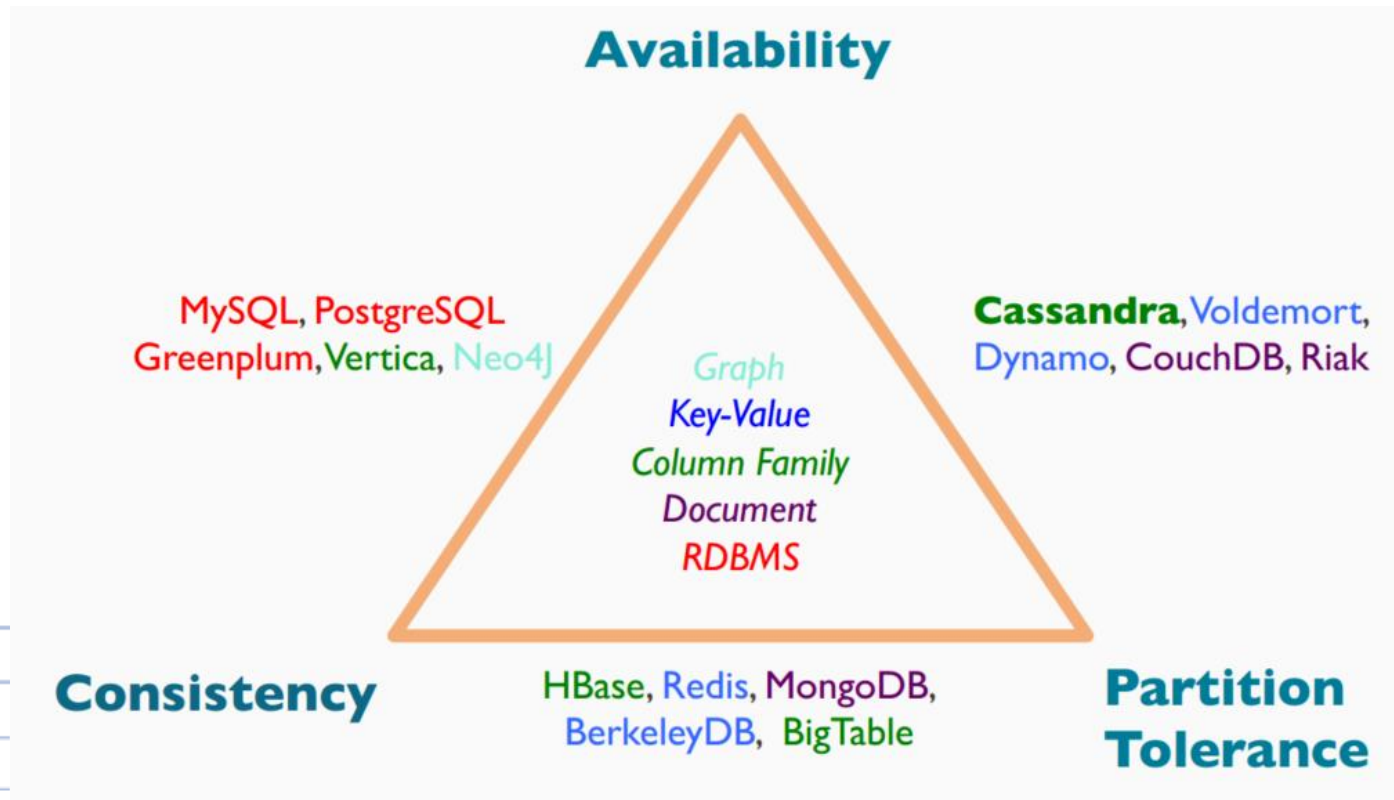
## Column Family

Las claves enlazan con un conjunto de columnas



# cassandra

## El teorema CAP





an NTT DATA Company

The background image shows a person in a dark jacket standing on a beach, looking out at a large rock formation with a natural archway. The scene is overlaid with a green-to-blue gradient and white geometric lines, including circles around the person's feet and a path leading towards the rock formation.

Apache  
Cassandra

## ADVERTENCIA:

Esto es un taller práctico

Si en algún momento te quedaras atascado, ejecuta los scripts que hay en tu **home**, en la carpeta **SoyMuyVago**



## ¿Qué es Cassandra?

### Características

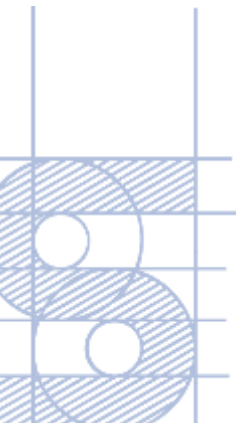
- Una base de datos **NoSQL** basada en **Column Families**.
- Posee un **esquema definido**, pero **flexible**, que permite su alteración.
- Usa un modelo **totalmente distribuido**, sin nodos maestros.



## A trabajar se ha dicho

Vamos a instalar Cassandra y levantar el primer nodo

- Verificamos fichero /etc/hosts y nuestro hostname
- Verificamos JDK
- Descomprimos Cassandra
- Configuramos ficheros “cassandra.yaml” y “cassandra-env.sh”
  - MAX\_HEAP, INITIAL\_HEAP y JMX\_PORT
  - cluster\_name, seed, listen\_address, rpc\_address
- Levantamos Cassandra



## ¿Qué es Cassandra?

Posee CQL, un lenguaje de consultas similar al SQL

```
-- Inserts or updates  
INSERT INTO Standard1 (KEY, col0, col1)  
VALUES (key, value0, value1)
```

vs.

```
-- Inserts or updates  
UPDATE Standard1  
SET col0=value0, col1=value1 WHERE KEY=key
```

## Keyspace

División lógica en la que vamos a trabajar

Crear un Keyspace:

```
CREATE KEYSPACE nombre_keyspace  
WITH replication={'class':'SimpleStrategy', 'replication_factor':3};
```

Asociar un Keyspace a la sesión actual de **cqlsh**

```
USE nombre_keyspace
```

Borrar un Keyspace y todos los datos que este posea

```
DROP KEYSPACE nombre_keyspace
```

## Table

Abstracción de una Column Family

```
CREATE TABLE performer (  
    name VARCHAR PRIMARY KEY,  
    type VARCHAR,  
    country VARCHAR,  
    style VARCHAR,  
    founded INT,  
    born INT,  
    died INT );
```

```
CREATE TABLE tracks_by_album (  
    album_title VARCHAR,  
    year INT,  
    performer VARCHAR STATIC,  
    genre VARCHAR STATIC,  
    number INT,  
    track_title VARCHAR,  
    PRIMARY KEY  
    ((album_title,year),number));
```

## CRUD

### Acceso a los datos a través de la Partition Key

#### Insertar:

Insert into performer (name, type) values ('Paco', 'Cantautor');

#### Actualizar:

Update performer set type = 'Cantante' where name='Paco';

#### Borrar:

Delete performer where name='Paco';

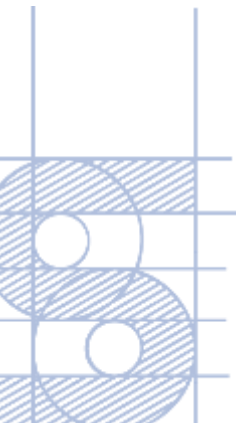
#### Leer:

Select \* from performer where name='Paco';

## A trabajar se ha dicho

Vamos a lanzar nuestras primeras querys

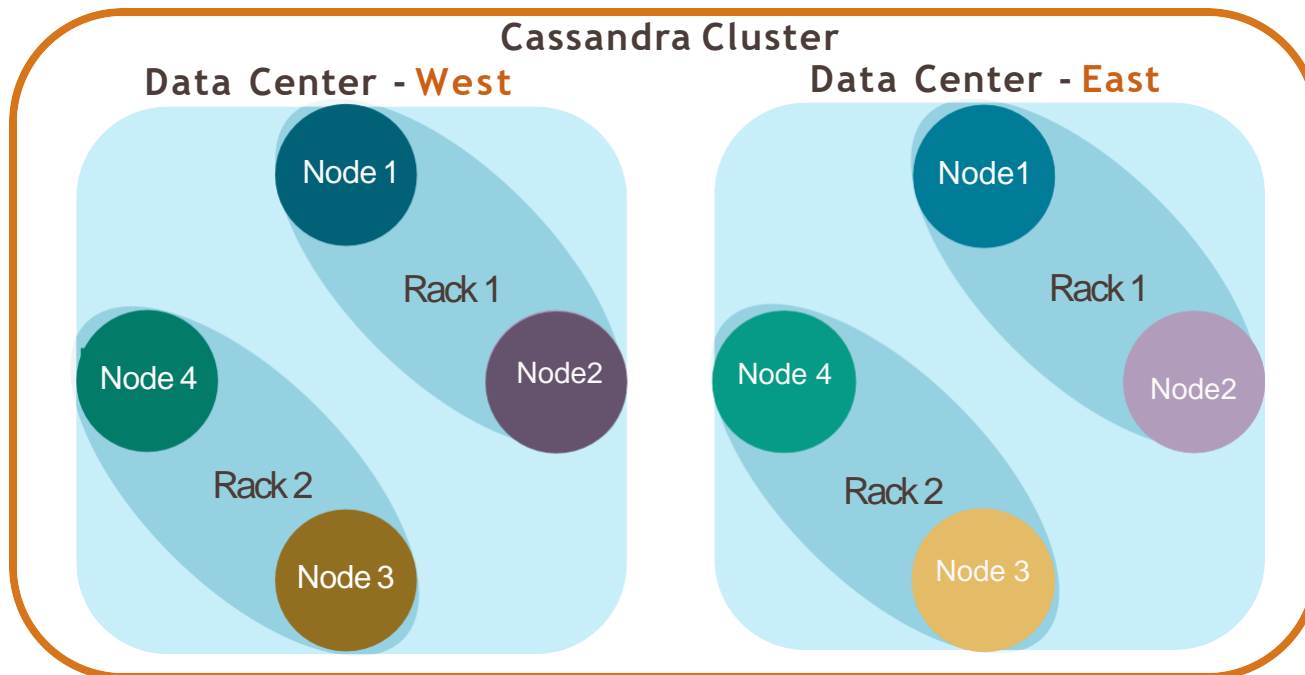
- Asegurémonos de que Cassandra está levantado (Si es necesario, usar “nodetool”)
- Vamos a crear un keyspace
- Vamos a crear una tabla
- Insertaremos 1 dato
- Haremos un import de un CSV





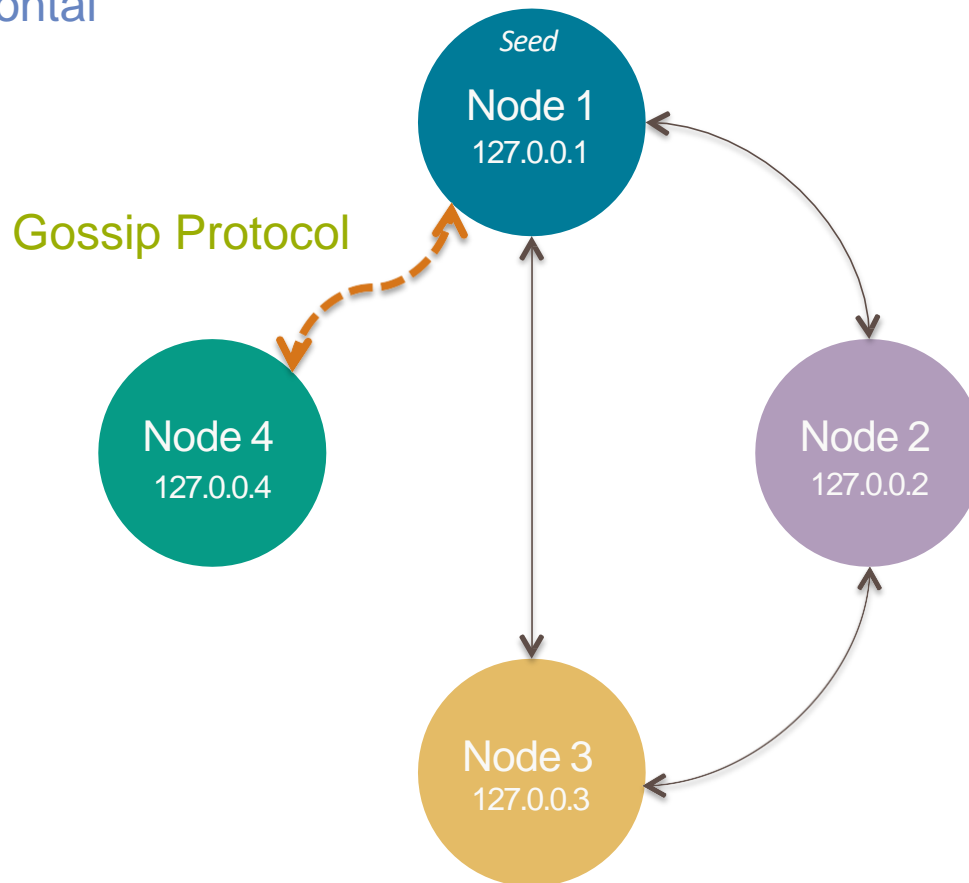
## ¿Qué es Cassandra?

### Clúster Cassandra



## ¿Qué es Cassandra?

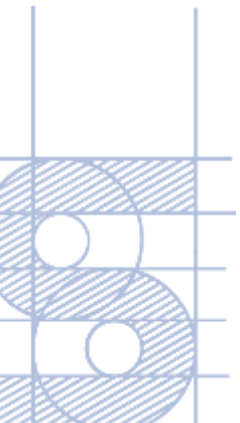
Escalado horizontal



### A trabajar se ha dicho

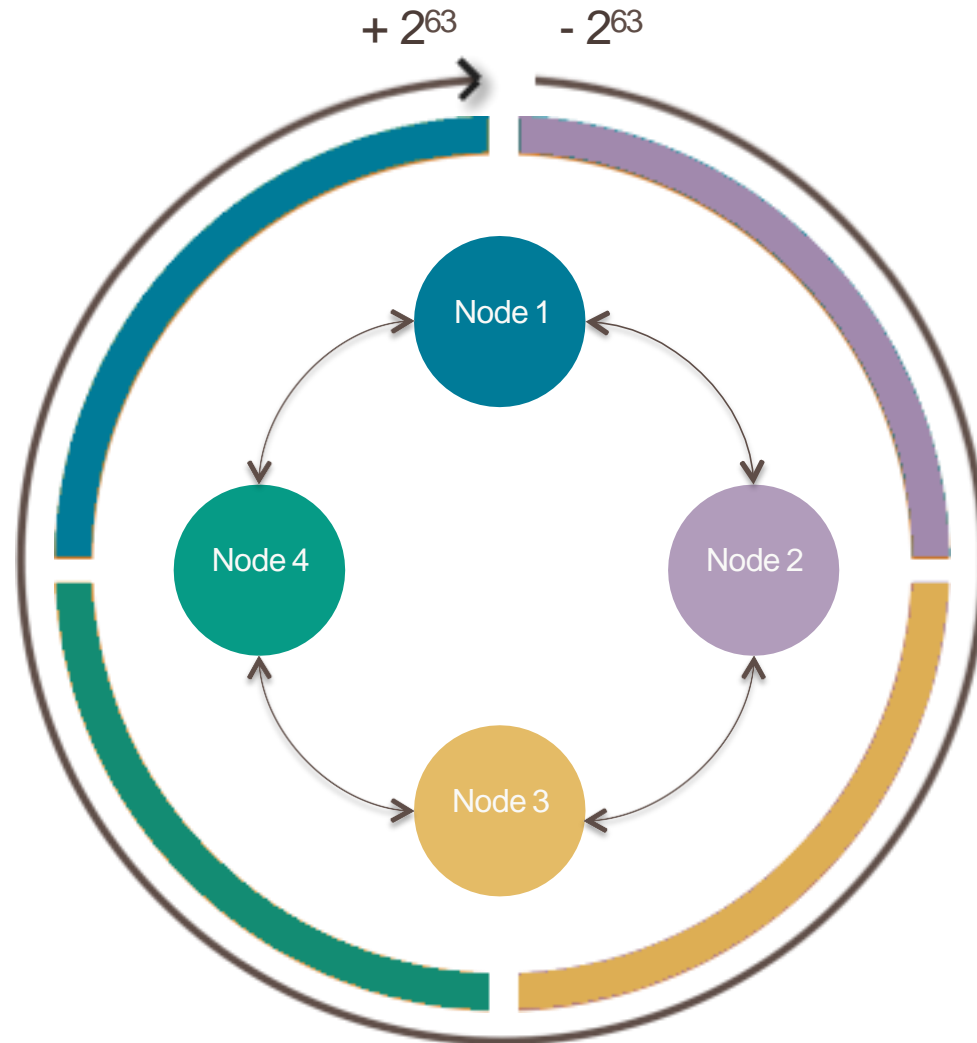
Vamos a instalar un segundo nodo de Cassandra

- Repetimos los pasos de la primera instalación
- Configuramos ficheros “cassandra.yaml” y “cassandra-env.sh”, y nos aseguramos de darle valores distintos
  - MAX\_HEAP, INITIAL\_HEAP y JMX\_PORT
  - cluster\_name, seeds, listen\_address, rpc\_address
- Levantamos Cassandra desde la nueva dirección



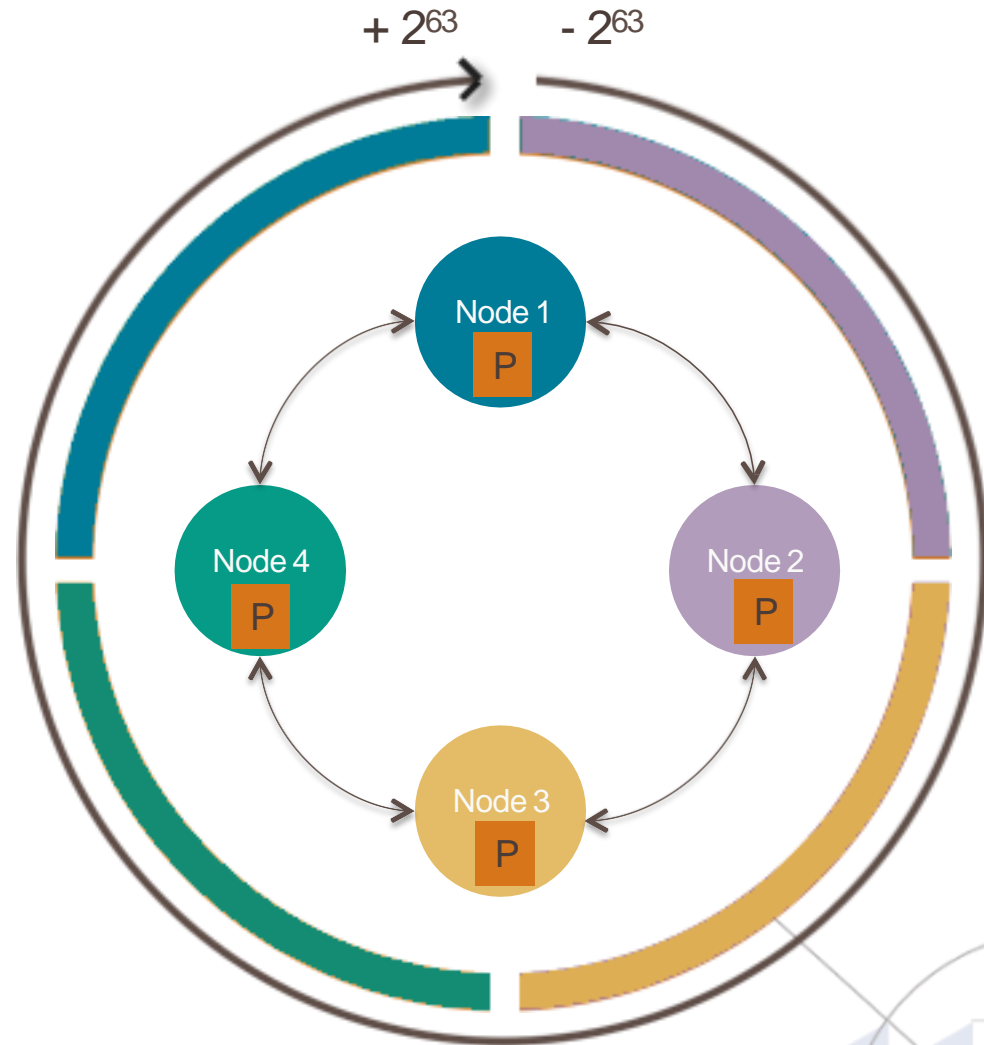
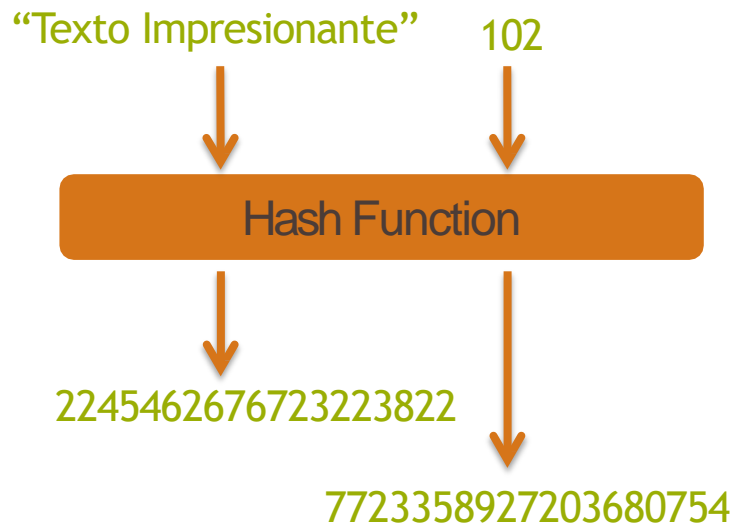
## ¿Qué es Cassandra?

Data sharding



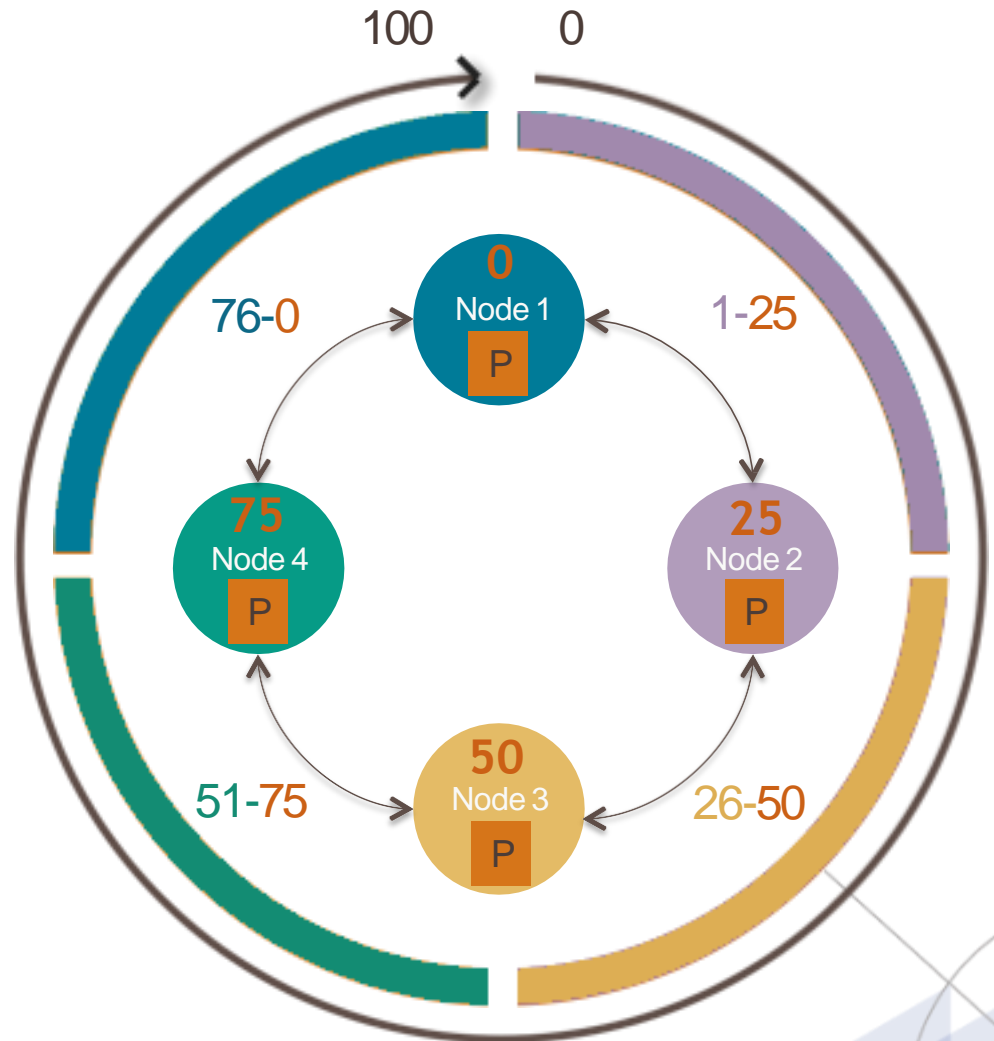
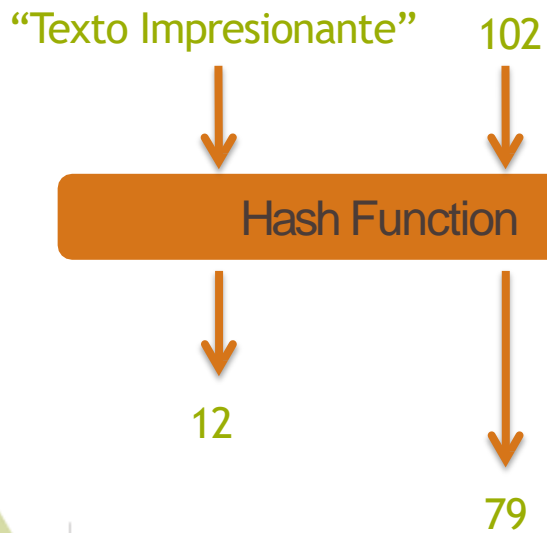
## ¿Qué es Cassandra?

### Data sharding



## ¿Qué es Cassandra?

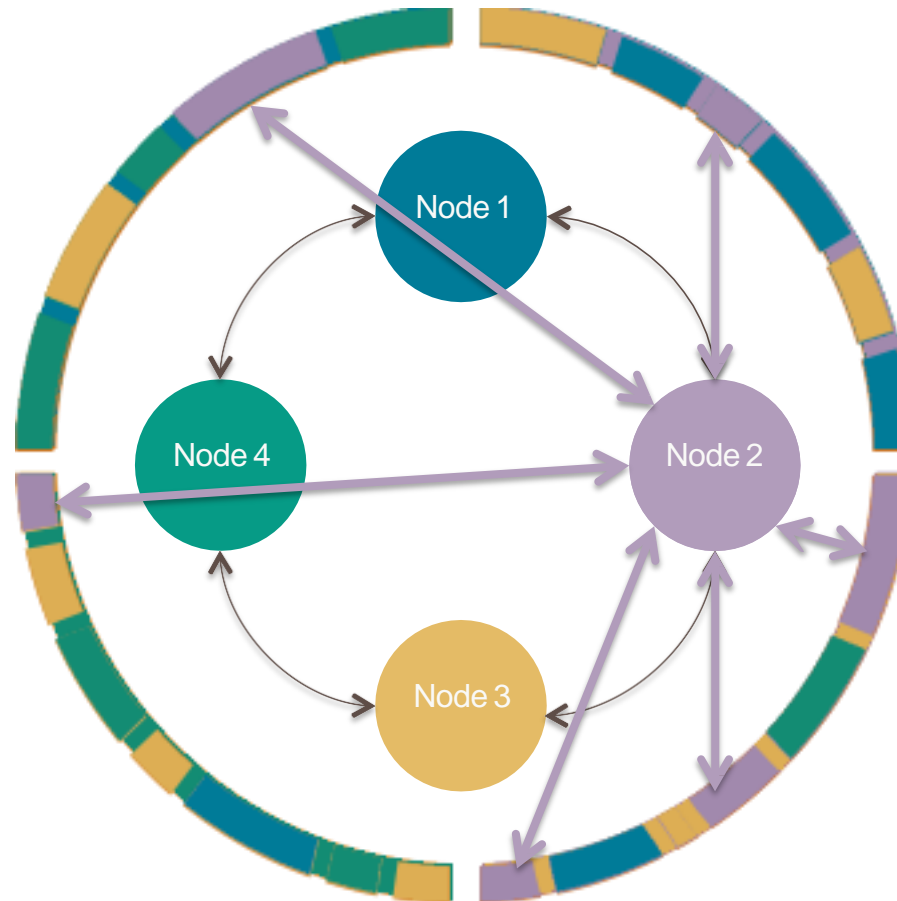
Data sharding





## ¿Qué es Cassandra?

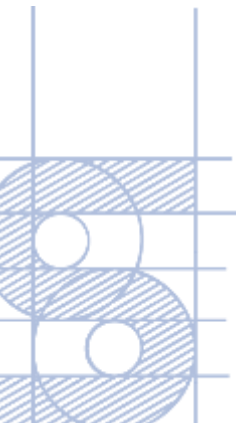
### Data sharding



## A trabajar se ha dicho

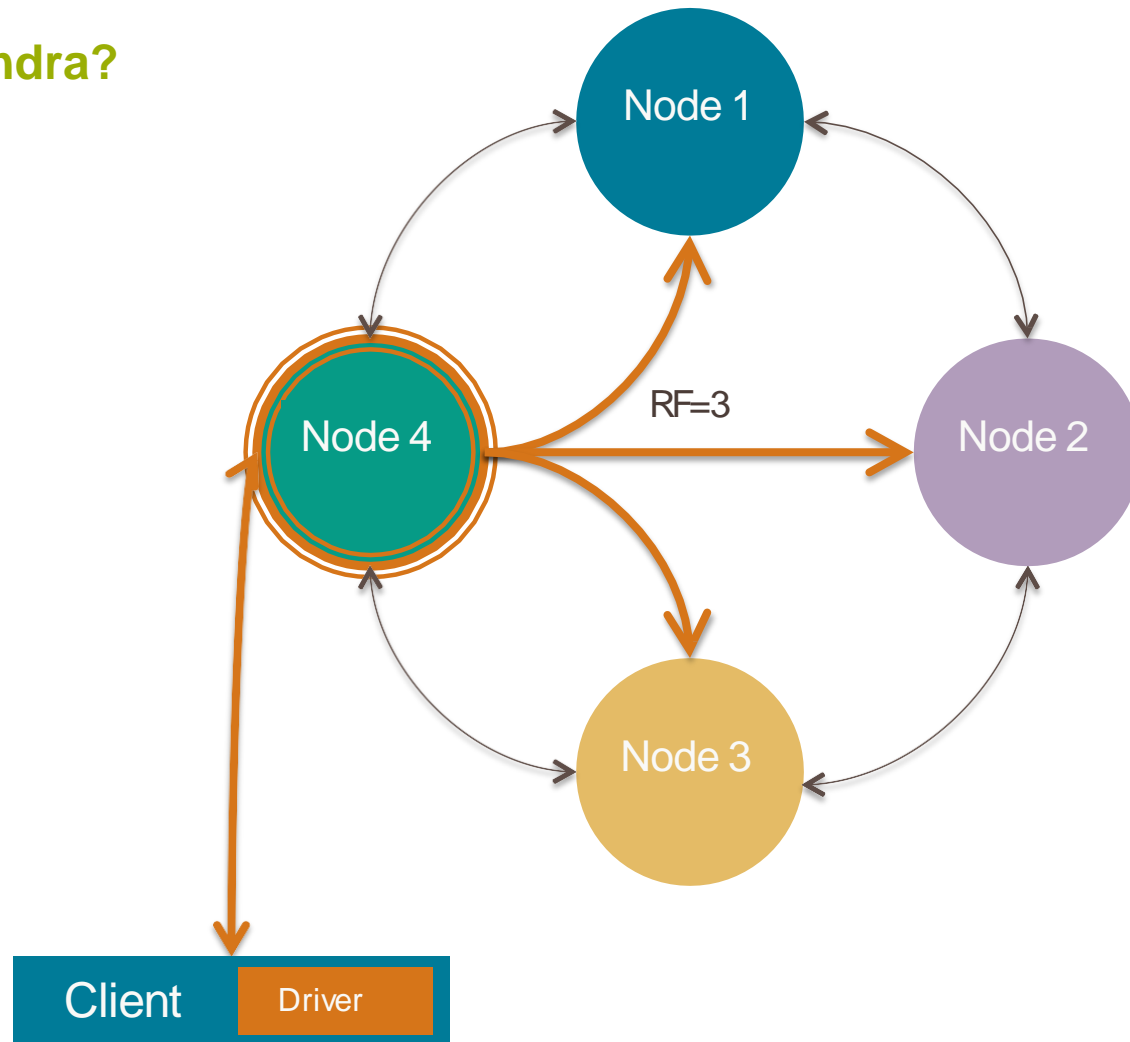
Vamos a probar cómo funciona la distribución de datos

- Usa la herramienta “nodetool getendpoints” para saber dónde se ubican la casa Targaryen, Lannister, Greyjoy y Stark
- Para uno de los nodos con “nodetool stopdaemon -h ip -p puerto”, y confirma que las casas que allí se ubicaban ya no se pueden consultar vía cqlsh
- Arranca de nuevo el nodo y verifica que se pueden consultar



## ¿Qué es Cassandra?

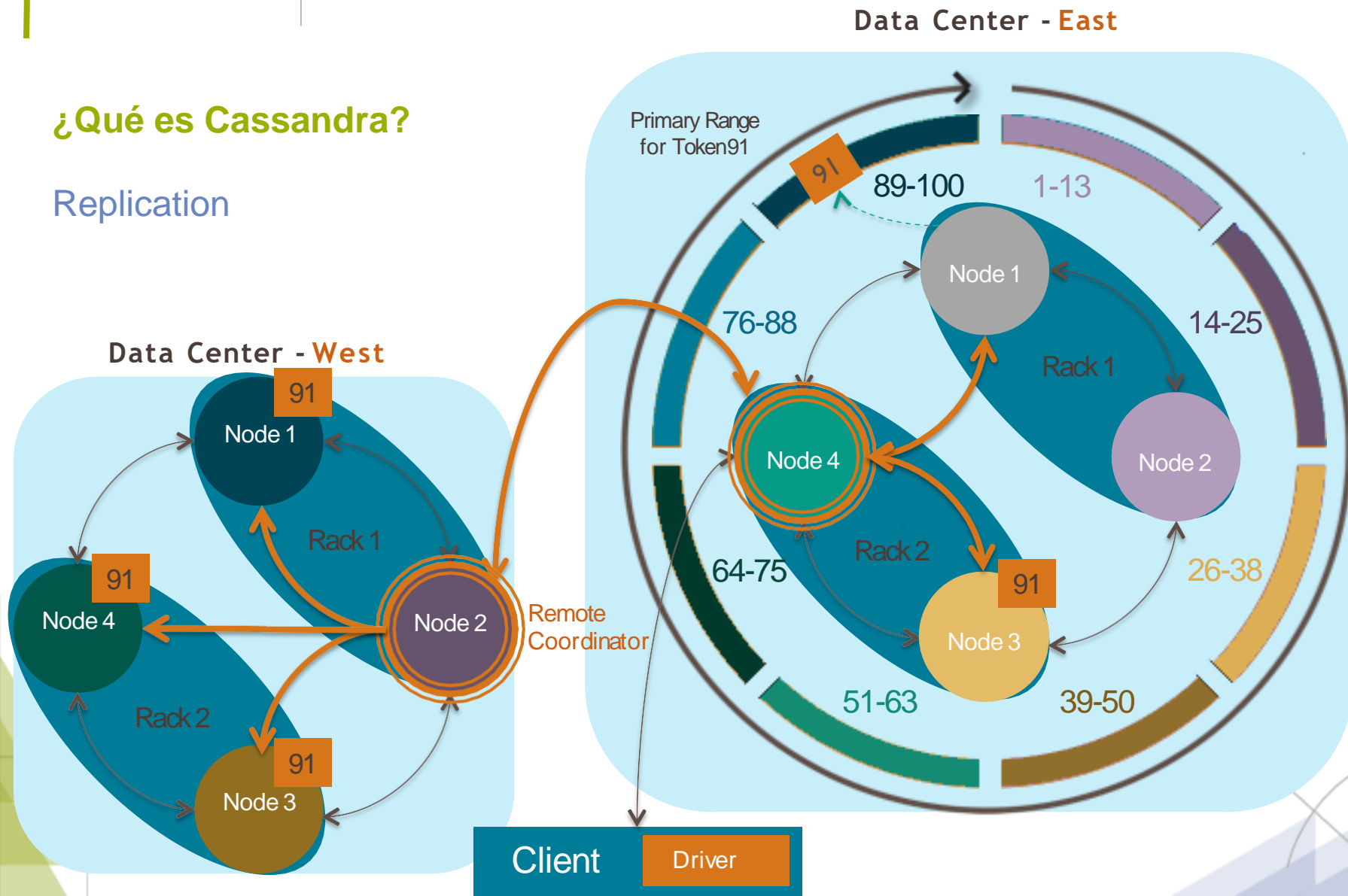
Coordination



# Cassandra

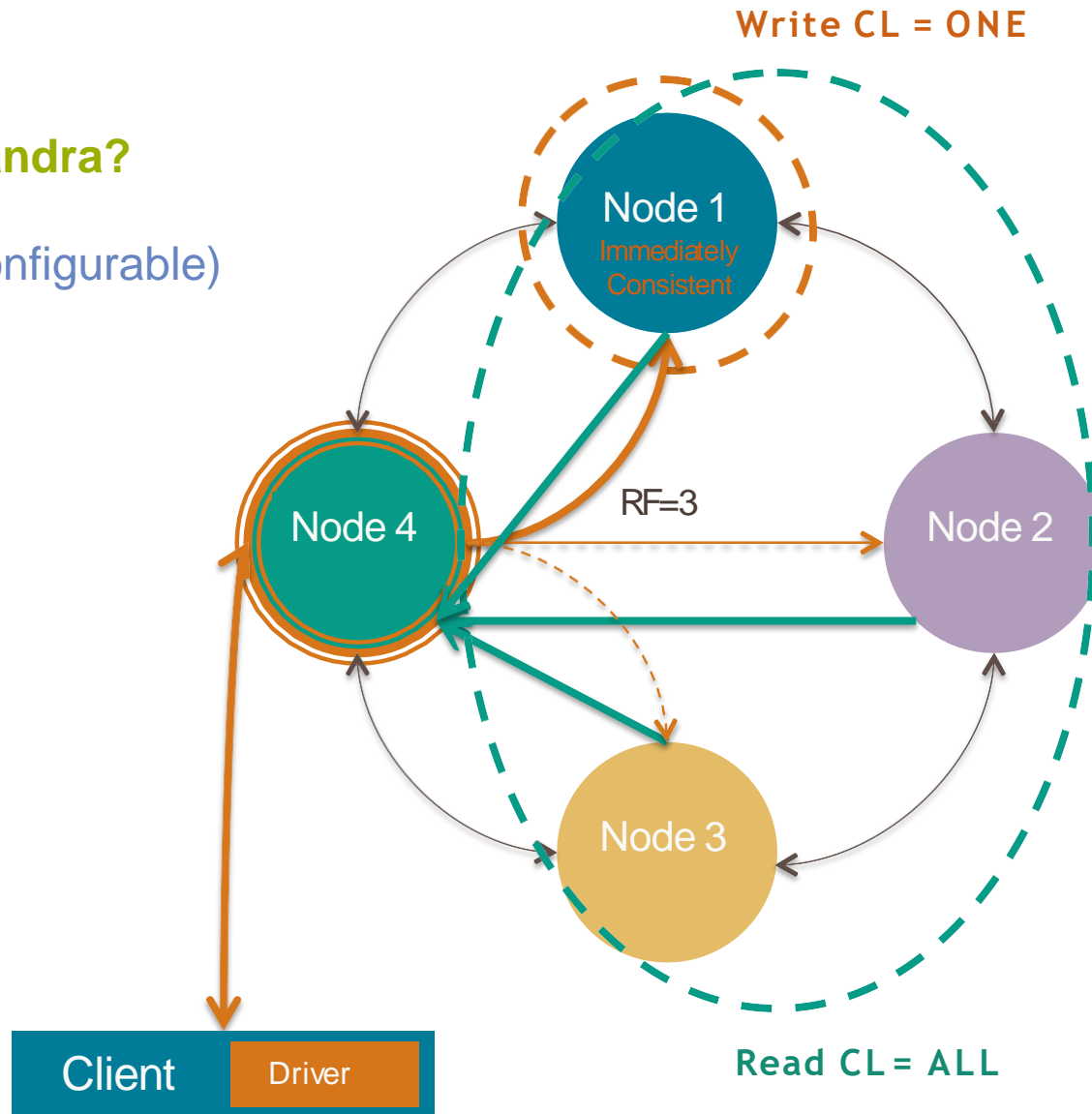
## ¿Qué es Cassandra?

### Replication



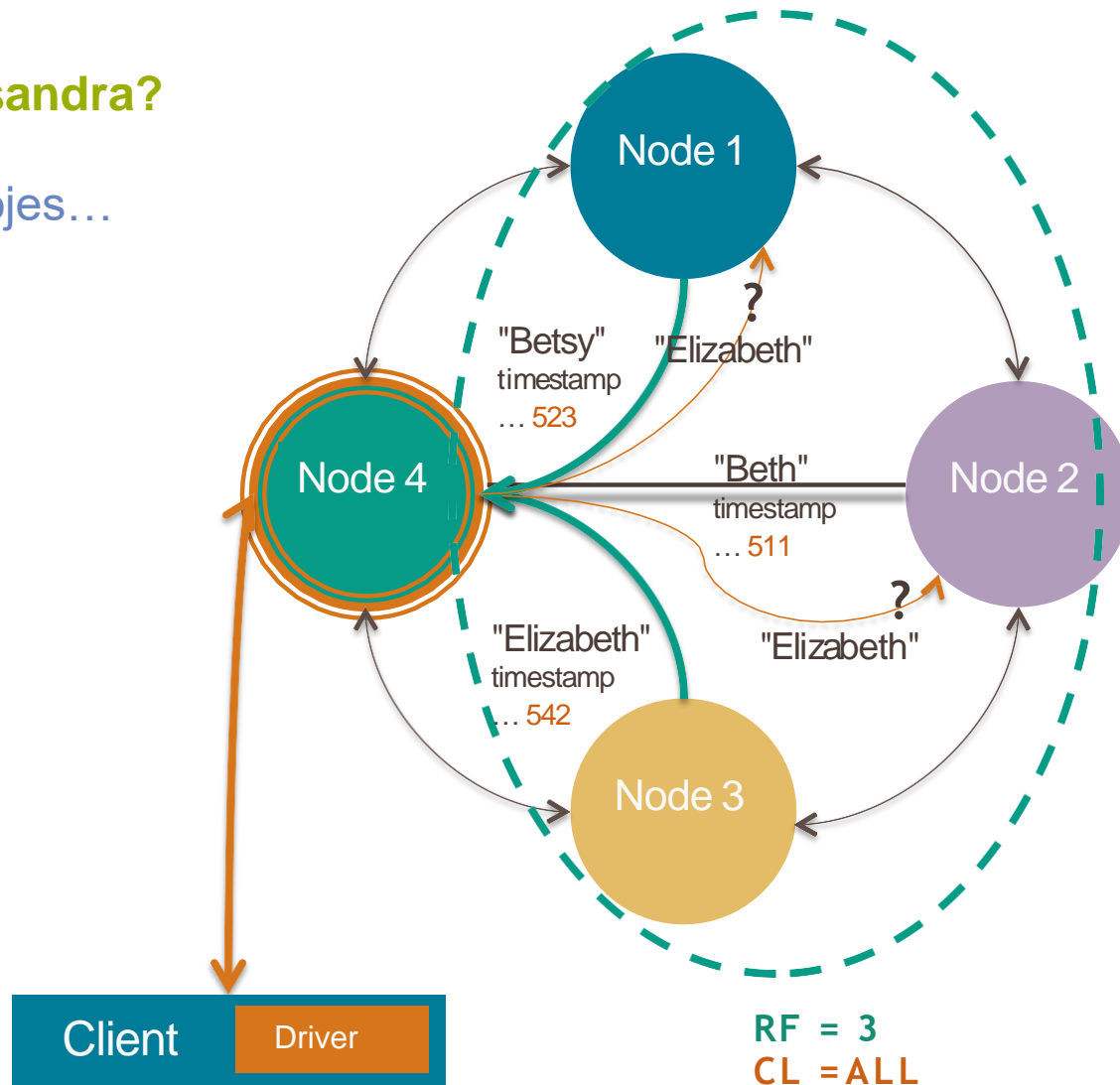
## ¿Qué es Cassandra?

## Consistenza (configurable)



## ¿Qué es Cassandra?

Si tienes 2 relojes...

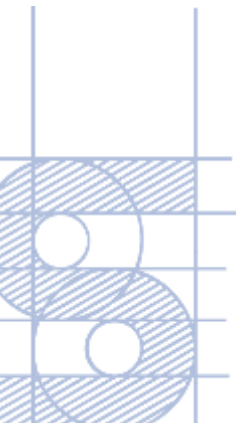




### A trabajar se ha dicho

Vamos a instalar un segundo nodo de Cassandra

- Borramos el keyspace westeros
- Volvemos a crearlo, pero esta vez con  $RF=2$
- Cargamos todos los datos, como hicimos la primera vez
- Tiramos un nodo y verificamos que todo funciona igual
- Arrancamos nuevamente el nodo para dejarlo todo bien



## Drivers Python

Mediante la herramienta pip de Python podemos instalar los drivers Cassandra

```
pip install cassandra-driver
```

Si la instalación falla, es necesario instalar cython (para los módulos en Python escritos en C y C++)

```
pip install cython
```

Si estamos ejecutando la VM, ya tenemos el driver instalado.

Si vamos a conectarnos desde otra máquina, desactivar antes el firewall. Asegurarnos de que las IPs de los hosts son las públicas (si no, configurar forwarding)

Para que funcione, en `cassandra.yaml`, tenemos que configurar `rpc_address` y `listen_address` con la ip externa

## Cassandra

En primer lugar, vamos a añadir los imports que vamos a usar

```
from cassandra.cluster import Cluster
from cassandra.policies import (TokenAwarePolicy,
DCAwareRoundRobinPolicy, RetryPolicy)
from cassandra.query import (PreparedStatement, BoundStatement)
```

Ahora tenemos que conectarnos al clúster (se dan una serie de nodos, el resto se obtiene por gossip).

```
cluster = Cluster(contact_points=['127.0.0.1'],
load_balancing_policy=
TokenAwarePolicy(DCAwareRoundRobinPolicy(local_dc='US-West')),
default_retry_policy = RetryPolicy() ,protocol_version=3 )
```

Como trabajamos en un clúster formado por varios nodos, vamos a especificar que las peticiones van a ir al nodo responsable de dicho token (primary range) con TokenAwarePolicy, y le indicamos cuál es el Data-Center que tiene que interpretar como local para realizar las queries. También especificamos la política de reintentos si la operación no ha finalizado.

Una vez que estamos conectados al clúster, especificamos con qué keyspace vamos a trabajar

```
session = cluster.connect('keyspace1')
```

A partir de ahora, podemos lanzar queries sobre dicha sesión

# Insertar

```
prepared_stmt = session.prepare ( "INSERT INTO ejemplo_python  
(col1,col2,col3) VALUES (?, ?, ?)" )  
bound_stmt = prepared_stmt.bind(['Jones','bob@example.com', 33])  
stmt = session.execute(bound_stmt)
```

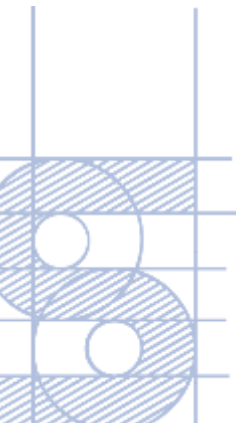
# select que devuelve nuestros datos

```
prepared_stmt = session.prepare ( "SELECT * FROM ejemplo_python  
WHERE (col1 = ?)" )  
bound_stmt = prepared_stmt.bind(['Jones'])  
stmt = session.execute(bound_stmt)  
for x in stmt: print x.col2, x.col3
```

### A trabajar se ha dicho

Vamos a instalar un segundo nodo de Cassandra

- Hacer un programa que inserte datos en Cassandra en una nueva tabla y los muestre de alguna forma



FIN

Gracias por venir

## El Everest es pan comido: Escala fácilmente con Cassandra



Javier Gómez Santos



[javier.gomez.santos@ucm.es](mailto:javier.gomez.santos@ucm.es)



@Javi\_Pronoide



<https://www.linkedin.com/in/javier-g%C3%B3mez-santos-7b182939/>



<https://github.com/gatchan00>