

# Aceleração Global Dev #4 everis

## Zookeeper e Sqoop em um ambiente clusterizado Hadoop

---

Rodrigo Garcia  
Big Data Projects Team Lead

# Objetivos da Aula

1. Entender o papel do Zookeeper em um cluster Hadoop
2. Entender o funcionamento do Sqoop para ingestão de bancos SGBD no HDFS
3. Realizar uma ingestão com Sqoop

# Requisitos Básicos

- ✓ Linux básico
- ✓ Noções de Shellsript
- ✓ Noções de processamento clusterizado
- ✓ Ter acompanhado a live de ontem

# Parte 1: Zookeeper

Zookeeper e Sqoop em  
um ambiente  
clusterizado Hadoop

# Zookeeper

- Serviço de **coordenação** distribuído;
- Gerenciamento de um grande conjunto de hosts (nós);
- Arquitetura simples e API;
- Assim como o Hadoop, vem **simplificar** o processo do desenvolvedor;
- Fornece as **rotas** necessárias para as peças do cluster.

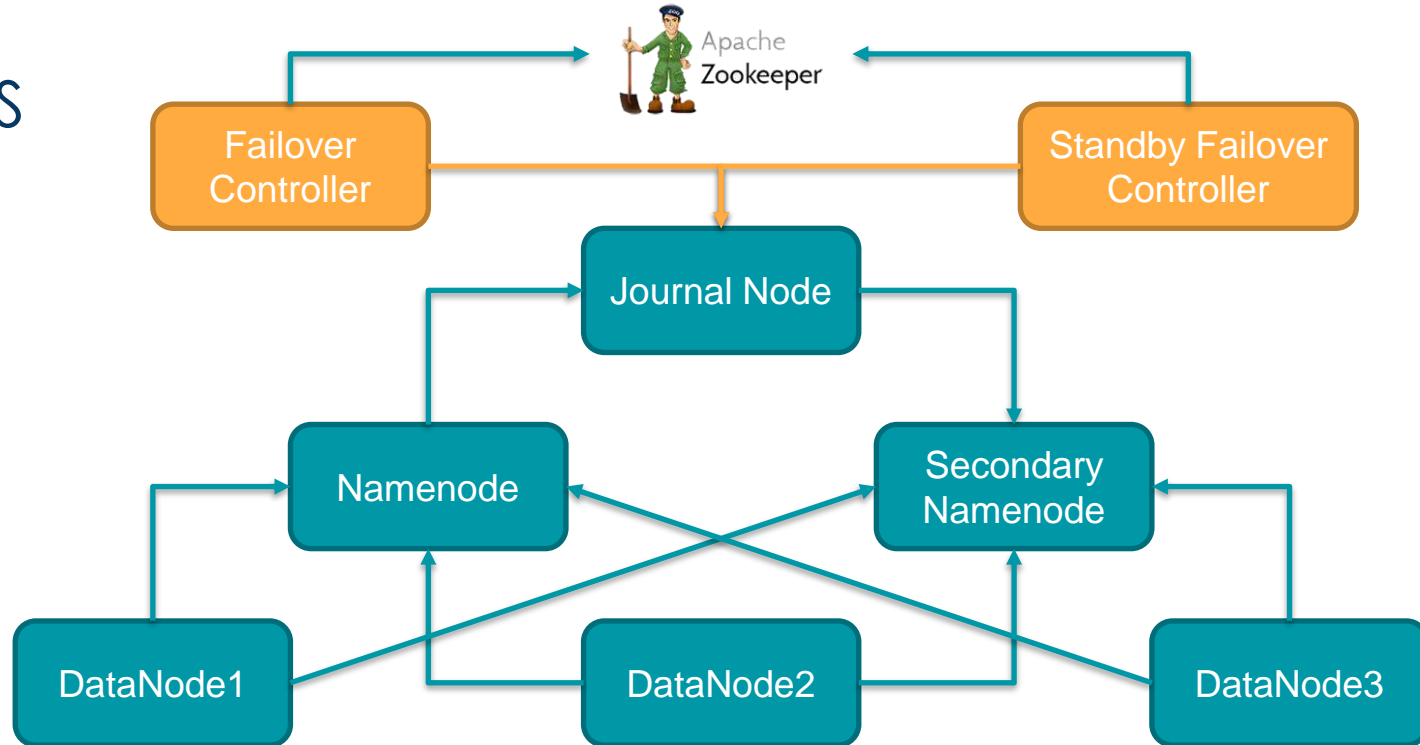
# Zookeeper

- **Identifica** os nós por nomes (DNS like);
- Gerencia e coordena as **configurações**;
- Funciona com esquema de **eleição** de líder (usa-se sempre pelo menos três Zookeeper);
- Pode **indisponibilizar** o dado enquanto está sendo **modificado**;
- Ajuda na **recuperação automática de falhas** (HBase, por exemplo).



# Exemplo Arquitetura

HDFS

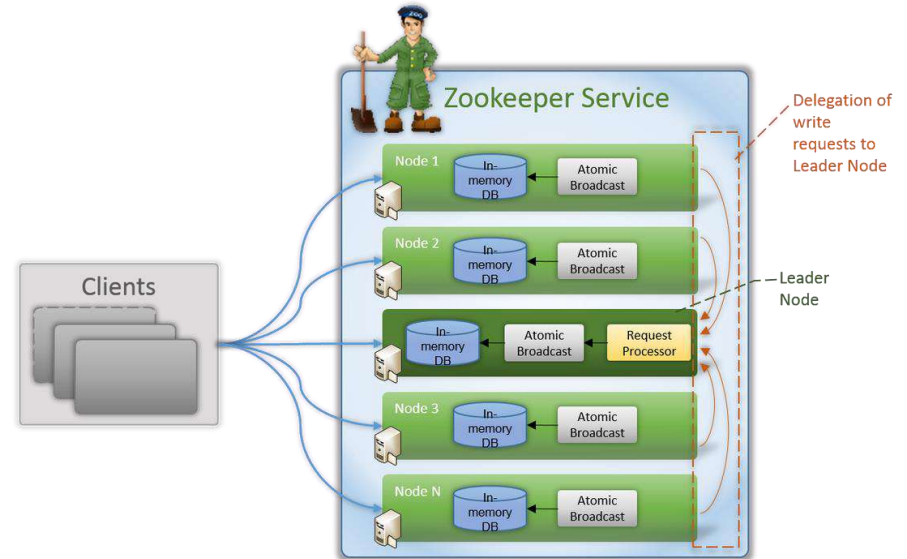




# Arquitetura Zookeeper

**Leader:** responsável pelo processamento de requests de escrita. Eleito internamente.

**Followers:** recebem as requests de leitura.







# Parte 2: Conceito Sqoop

Zookeeper e Sqoop em  
um ambiente  
clusterizado Hadoop

# Sqoop

- Originalmente desenvolvido pela **Cloudera**;
- Movimenta dados entre **banco de dados relacional** e **HDFS**;
- Pode-se **importar** todas as tabelas, apenas uma tabela ou parte de uma tabela para o HDFS;
- Também permite **exportar** de dados do HDFS para um banco de dados;
- Permite **automação** do processo de ingestão.

# Como funciona?

- Realiza a **leitura linha por linha** da tabela para escrever o arquivo no HDFS;
- O resultado do import é um conjunto de arquivos contendo a **cópia** dos dados da tabela importada;
- Under the hood, gera classes **Java**, permitindo que o usuário possa interagir com o dado importado;
- Pode importar dados e metadados de bancos de **dados SQL** direto para o **Hive**;

# Como funciona?

- Utiliza **MapReduce** para realizar import/ export dos dados, provendo um processamento paralelo e tolerante a falha.
- Permite especificar o **intervalo** e quais **colunas** serão importadas;
- Possibilita a especificação de **delimitadores** e **formatos** de arquivos;

# Como funciona?

- Realiza conexões com bancos de dados em paralelo, **executando comandos** de Select(import) e Insert/Update(export);
- Aceita conexão com **diversos** plug-ins: MySQL, PostgreSQL, Oracle, Teradata, Netezza, Vertica, DB2 e SQL Server;
- O formato padrão do arquivo importado no HDFS é **CSV**.



# Exemplo

Origem: Tabela Cities no MySQL

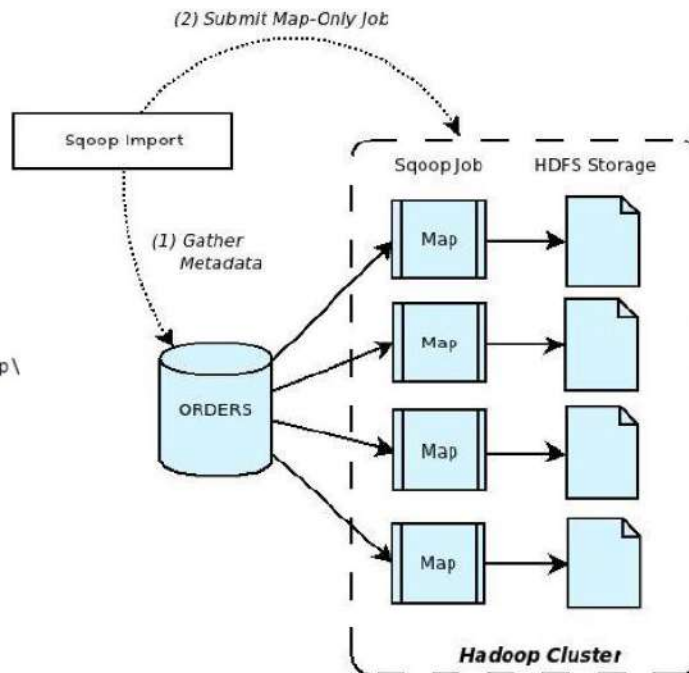
id	country	city
1	USA	Palo Alto
2	Czech Republic	Brno
3	USA	Sunnyvale

Linha de comando do Sqoop

```
sqoop import \  
--connect jdbc:mysql://mysql.example.com/sqoop \  
--username sqoop \  
--password sqoop \  
--table cities
```

Destino: Arquivo CSV no HDFS  
(home do usuário)

```
1,USA,Palo Alto  
2,Czech Republic,Brno  
3,USA,Sunnyvale
```





# Exemplo

```
sqoop import \  
--connect jdbc:mysql://mysql.example.com/sqoop \  
--username sqoop \  
--password sqoop \  
--table cities \  
--warehouse-dir /etl/input/ Permite especificar um diretório no HDFS como destino. \  
--where "country = 'Brazil'" Para importar apenas um subconjunto de registros de uma tabela. \  
-P ou --password-file my-sqoop-password \  
--as-sequencefile ou --as-avrodatafile Para escrever o arquivo no HDFS em formato binário (Sequence ou Avro). \  
--compress Comprime os blocos antes de gravar no HDFS em formato gzip por padrão. \  
--compression-codec Utilizar outros codecs de compressão, exemplo: org.apache.hadoop.io.compress.BZip2Codec. \  
--direct Realiza import direto por meio das funcionalidades nativas do banco de dados para melhorar a performance, exemplo: mysqldump ou pg_dump. \  
--map-column-java c1=String Especificar o tipo do campo. \  
--num-mappers 10 Especificar a quantidade de paralelismo para controlar o workload. \  
--null-string "\\N" \  
--null-non-string "\\N" \  
--incremental append ou lastmodified Funcionalidade para incrementar os dados. \  
--check-column id ou last_update_date Identifica a coluna que será verificada para incrementar novos dados. \  
--last-value 1 ou "2013-05-22 01:01:01" Para especificar o último valor importado no Hadoop.
```

Splittable	Not Splittable
BZip2, LZO	GZip, Snappy



# Exemplo

Import da tabela accounts

```
$ sqoop import --table accounts \  
--connect jdbc:mysql://dbhost/loudacre \  
--username dbuser --password pw
```

Import da tabela accounts utilizando um delimitador

```
$ sqoop import --table accounts \  
--connect jdbc:mysql://dbhost/loudacre \  
--username dbuser --password pw \  
--fields-terminated-by "\t"
```

Import da tabela accounts limitando os resultados

```
$ sqoop import --table accounts \  
--connect jdbc:mysql://dbhost/loudacre \  
--username dbuser --password pw \  
--where "state='CA'"
```





# Exemplo

Import incremental baseado em um timestamp.

Deve certificar-se de que esta coluna é atualizada quando os registros são atualizados ou adicionados

```
$ sqoop import --table invoices \  
--connect jdbc:mysql://dbhost/loudacre \  
--username dbuser --password pw \  
--incremental lastmodified \  
--check-column mod_dt \  
--last-value '2015-09-30 16:00:00'
```

Import baseado no último valor de uma coluna específica

```
$ sqoop import --table invoices \  
--connect jdbc:mysql://dbhost/loudacre \  
--username dbuser --password pw \  
--incremental append \  
--check-column id \  
--last-value 9478306
```

# Parte 3: Realizar uma ingestão com Sqoop

Zookeeper e Sqoop em um ambiente clusterizado Hadoop

# Instalando o Sqoop

```
sudo yum install --assumeyes sqoop
cd /tmp
wget
http://www.java2s.com/Code/JarDownload/java-
json/java-json.jar.zip
unzip /tmp/java-json.jar.zip
sudo mv /tmp/java-json.jar /usr/lib/sqoop/lib/
sudo chown root: /usr/lib/sqoop/lib/java-
json.jar
```

# Instalando o Sqoop

sqoop-version

21/01/20 17:00:47 INFO sqoop.Sqoop: Running

**Sqoop version: 1.4.6-cdh5.16.2**

Sqoop 1.4.6-cdh5.16.2

git commit id

Compiled by jenkins on Mon Jun 3 03:34:57 PDT  
2019



DIGITAL  
INNOVATION  
ONE

UNIVERSITY



# Live Demo

## HANDS-ON!





# Live Demo

## Arquivos necessários:

- ✓ `install_sqoop.sh`
- ✓ `pokemon.sql`
- ✓ `sqoop_import.sh`



# Live Demo

## Comandos:

```
sh install_sqoop.sh
```

```
mysql -u root -h localhost -pEveris@2021 < pokemon.sql
```

```
sh sqoop_import.sh
```

```
hdfs dfs -text /user/everis-bigdata/pokemon/*.gz  
| more
```

**Import subsets com Sqoop seguindo as seguintes premissas:**

1. Todos os Pokémon lendários;
2. Todos os Pokémon de **apenas** um tipo;
3. Os top 10 Pokémon mais rápidos;
4. Os top 50 Pokémon com menos HP;
5. Os top 100 Pokémon com maiores atributos;



# Dúvidas?

Zookeeper e Sqoop em  
um ambiente  
clusterizado Hadoop