



INSIGHT



# hadoop

## HADOOP/HDFS

Também pode conter um subtítulo logo abaixo



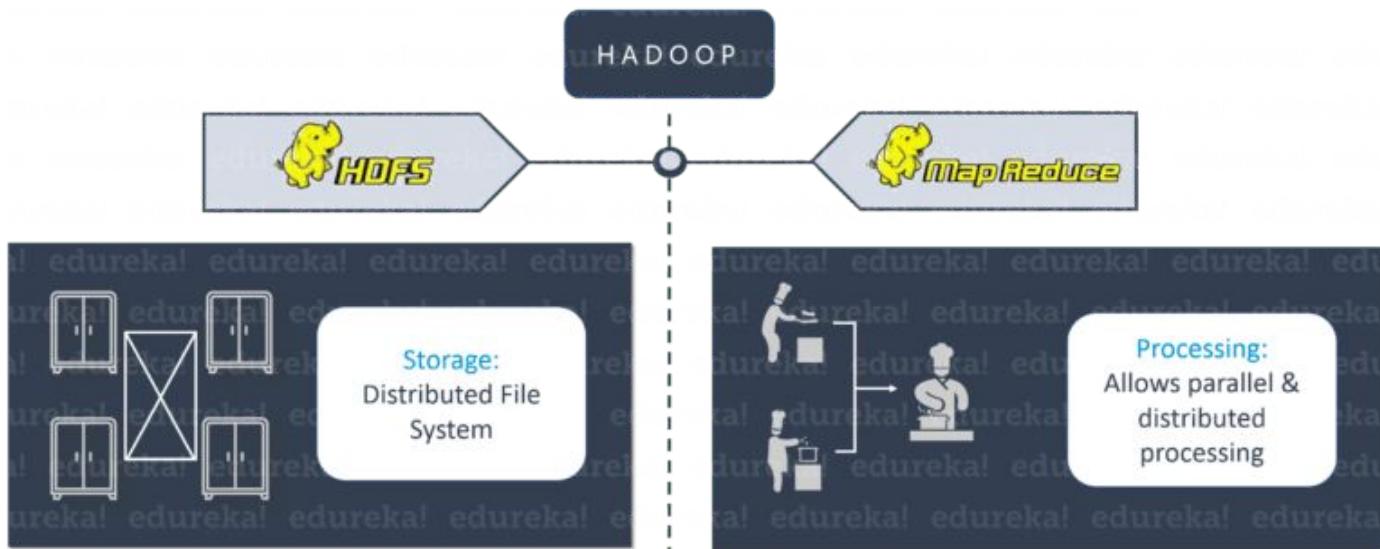
# AGENDA

1. Introdução
2. HDFS
3. Instalação do Hadoop
4. Introdução ao Docker
5. Atividade Prática

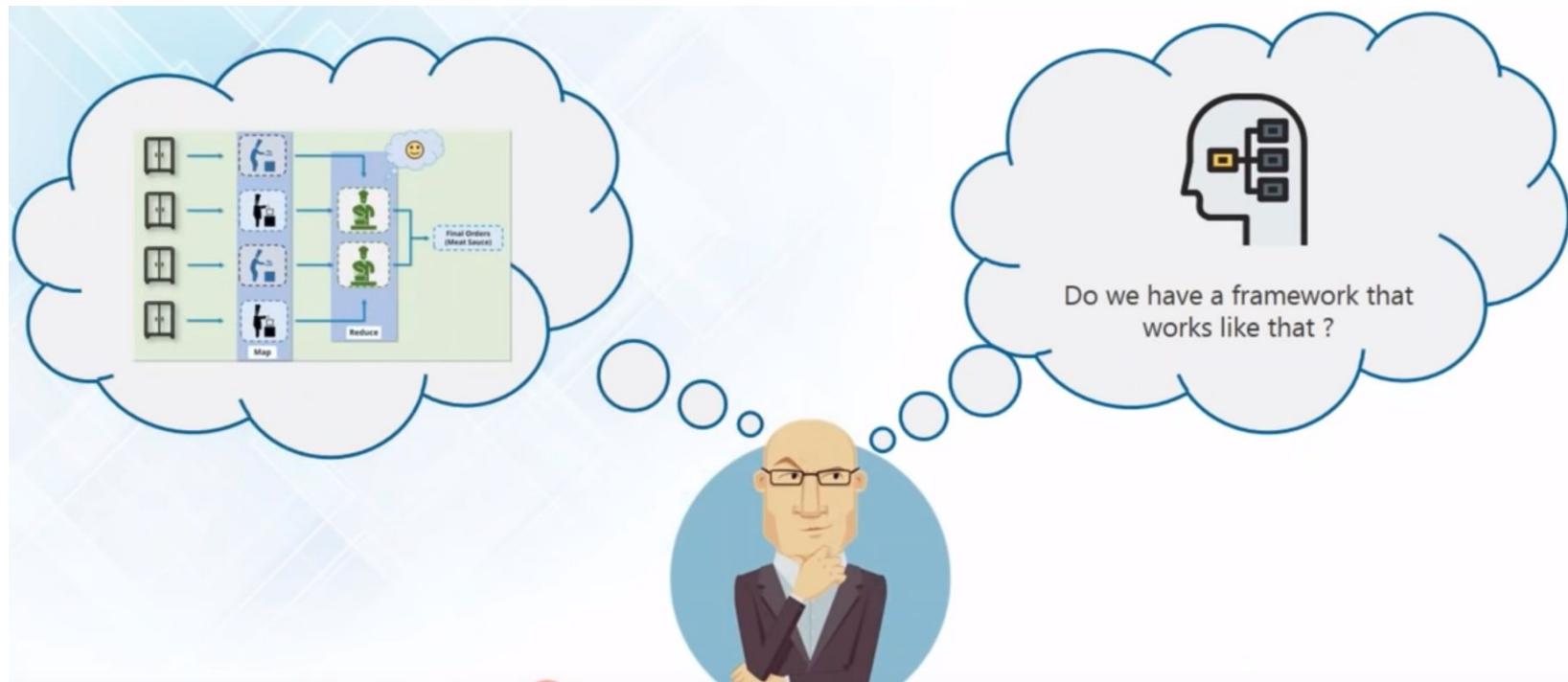
# 1. INTRODUÇÃO

# APACHE HADOOP

- Hadoop é uma plataforma de código aberto que fornece soluções para **armazenar e processar** grandes conjuntos de dados distribuídos, utilizando *clusters* de computadores com **hardware comum**.



# SOLUÇÃO EFETIVA - DISTRIBUIR E PARALELIZAR TAREFAS



# MÓDULOS PRINCIPAIS



**Hadoop v1.0**

**MapReduce**  
Data Processing  
& Resource Management

**HDFS**  
Distributed File Storage



**Hadoop v2.0**

**MapReduce**

**Other Data  
Processing  
Frameworks**

**YARN**

Resource Management

**HDFS**

Distributed File Storage

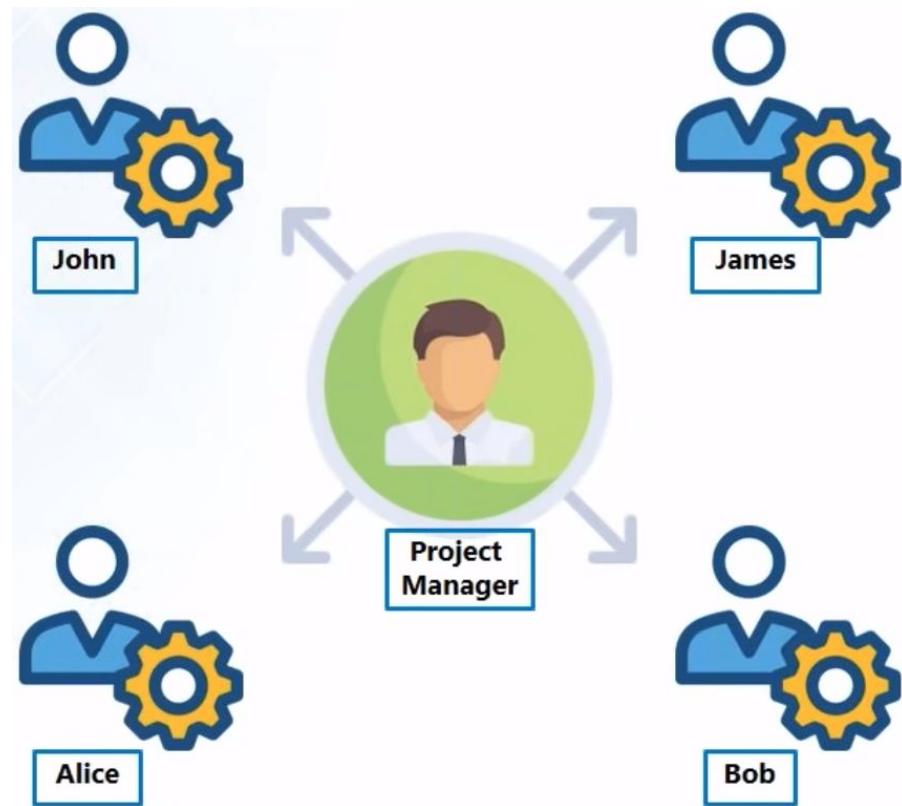
## 2. HDFS

# HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

- HDFS surgiu para dar suporte de armazenamento de grandes arquivos executando em *clusters* de hardware comum;
- Fornece mecanismos de tolerância a falhas;
- Fornece acesso paralelo e de alto rendimento aos dados de uma aplicação;

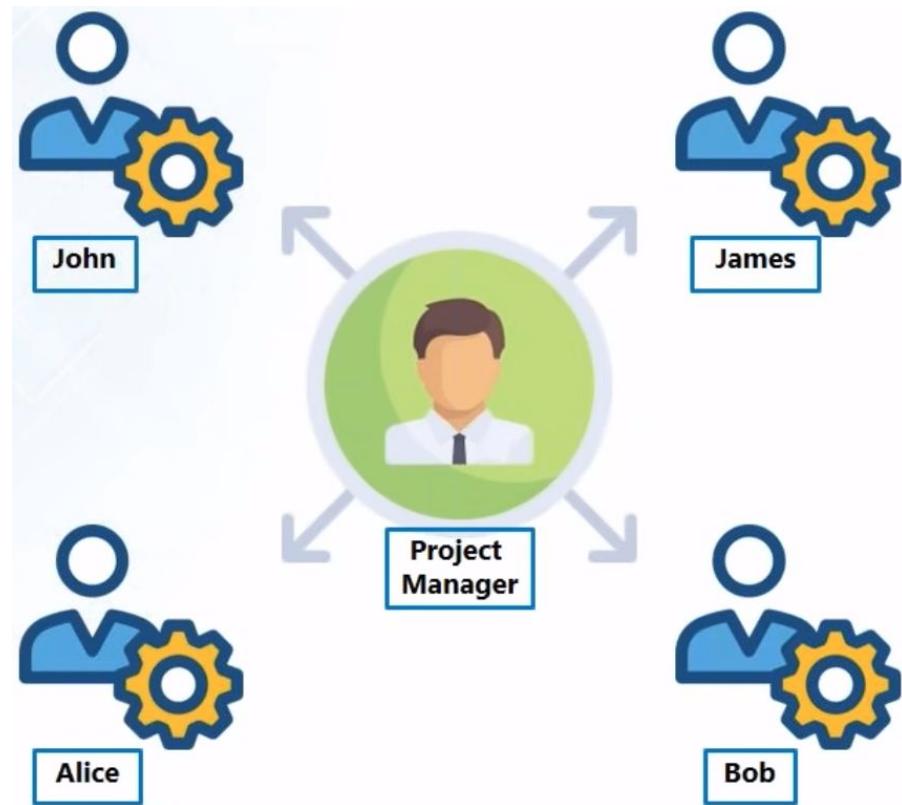
# ARQUITETURA MASTER/SLAVE

- **Cenário:** um gerente de projetos gerencia uma equipe com quatro empregados.



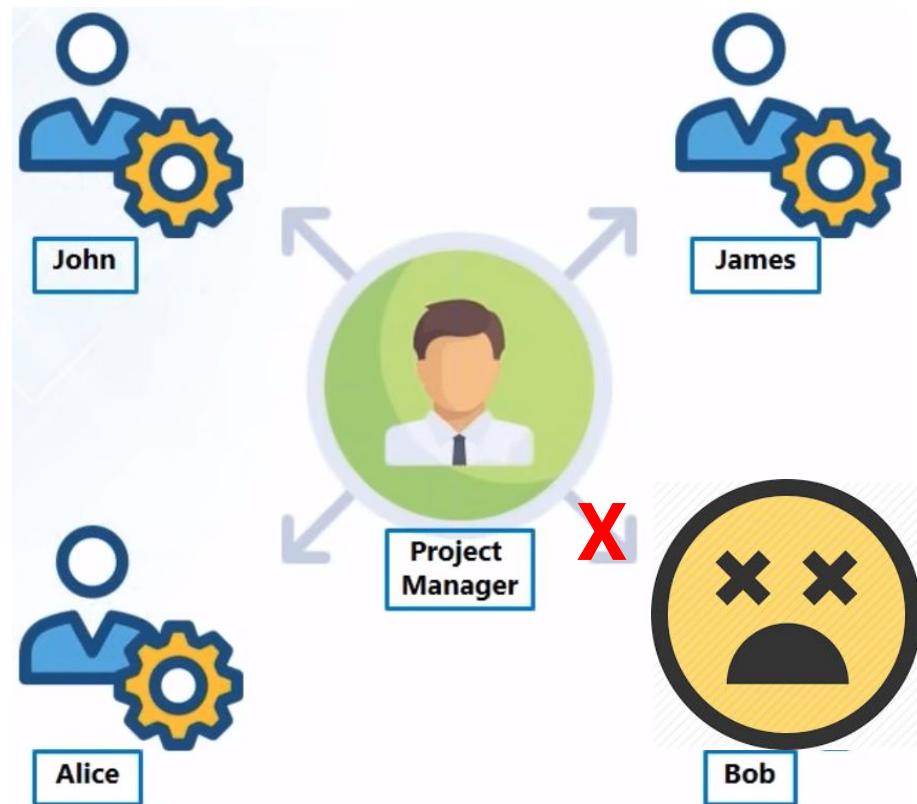
# ARQUITETURA MASTER/SLAVE

- **John** - Projeto A;
- **James** - Projeto B;
- **Bob** - Projeto C;
- **Alice** - Projeto D;
- **Gerente** - conhece os *deadlines* de entrega e possui os *backups* dos dados.



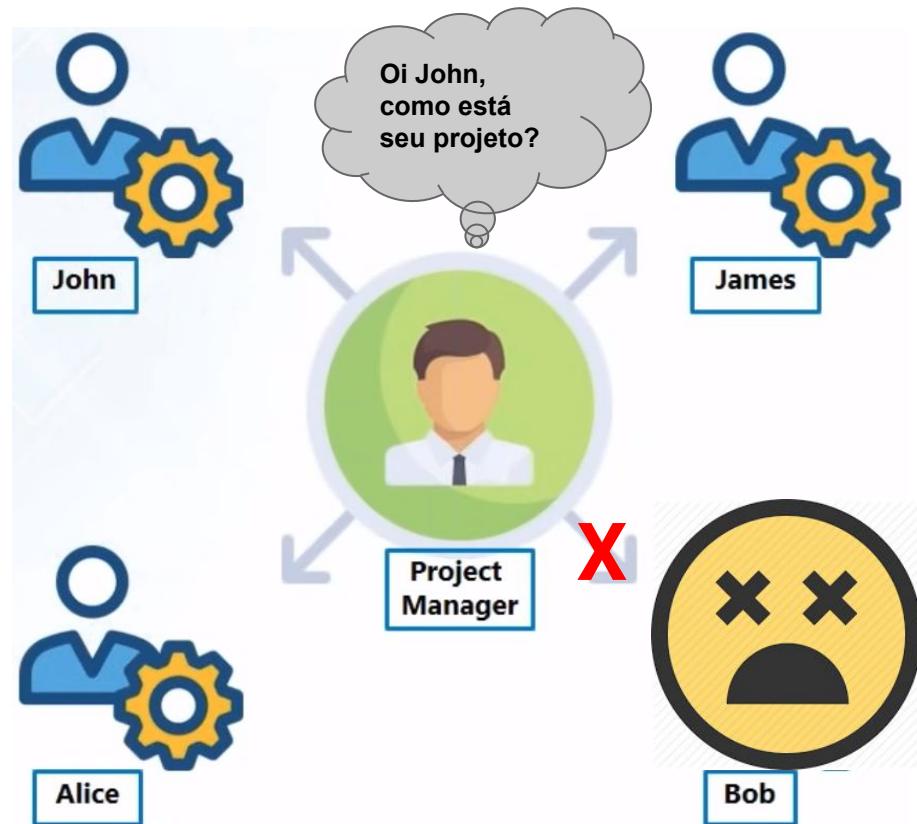
# ARQUITETURA MASTER/SLAVE

- **John** - Projeto A;
- **James** - Projeto B;
- **Bob** - Projeto C;
- **Alice** - Projeto D;
- **Gerente** - conhece os *deadlines* de entrega e possui os *backups* dos dados.



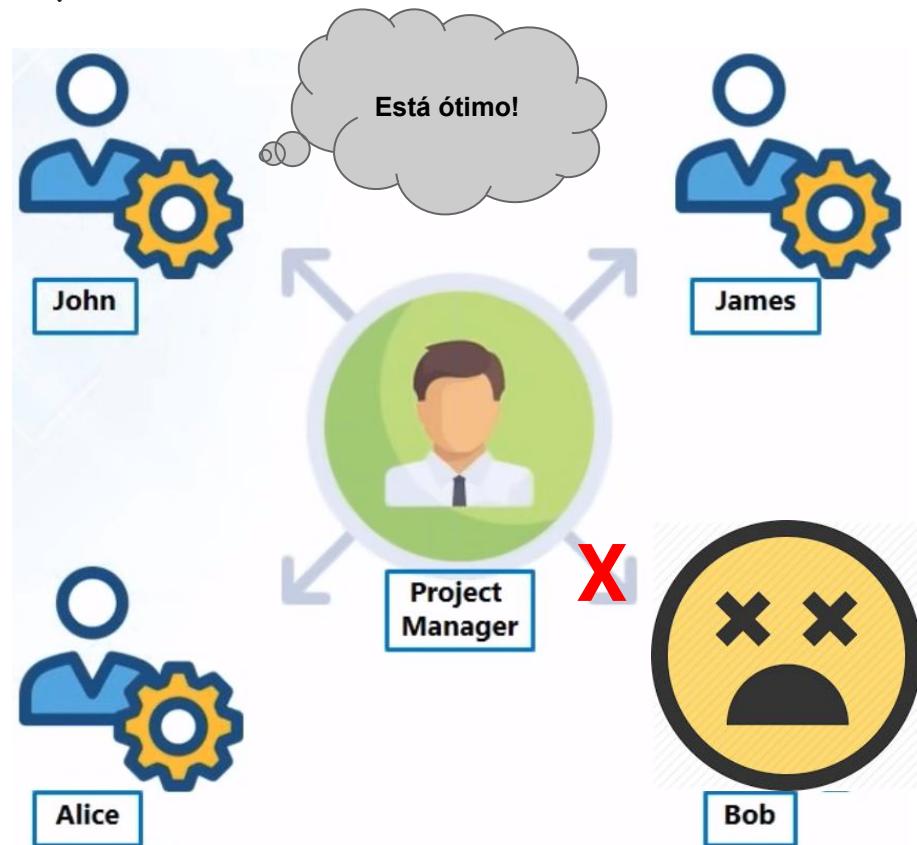
# ARQUITETURA MASTER/SLAVE

- **John** - Projeto A;
- **James** - Projeto B;
- **Bob** - Projeto C;
- **Alice** - Projeto D;
- **Gerente** - conhece os *deadlines* de entrega e possui os *backups* dos dados.



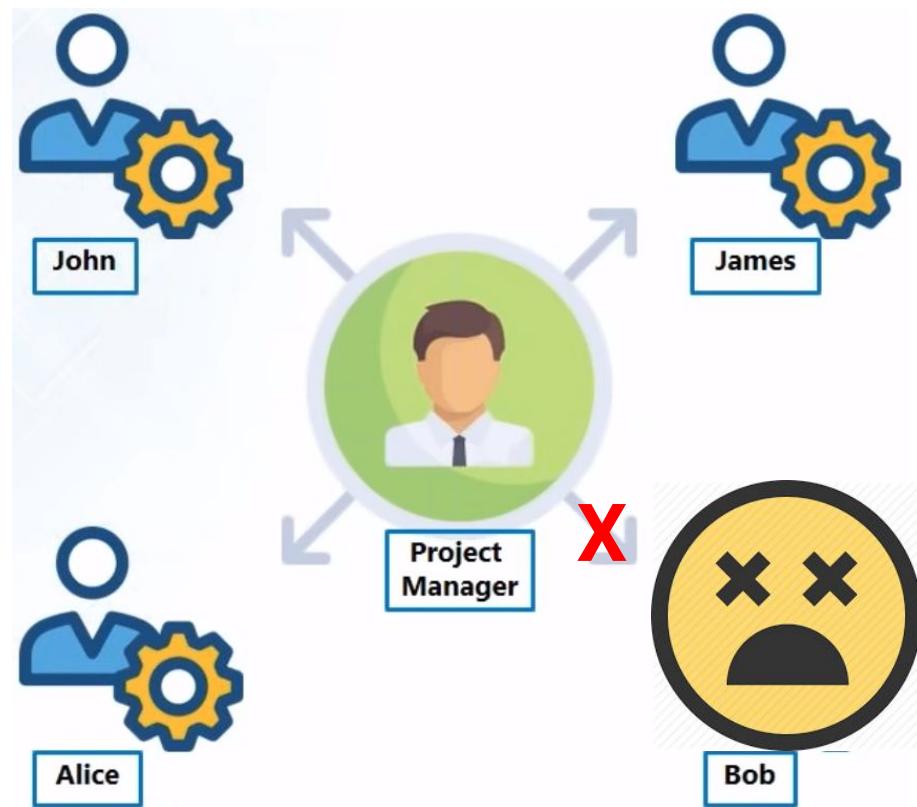
# ARQUITETURA MASTER/SLAVE

- **John** - Projeto A;
- **James** - Projeto B;
- **Bob** - Projeto C;
- **Alice** - Projeto D;
- **Gerente** - conhece os *deadlines* de entrega e possui os *backups* dos dados.

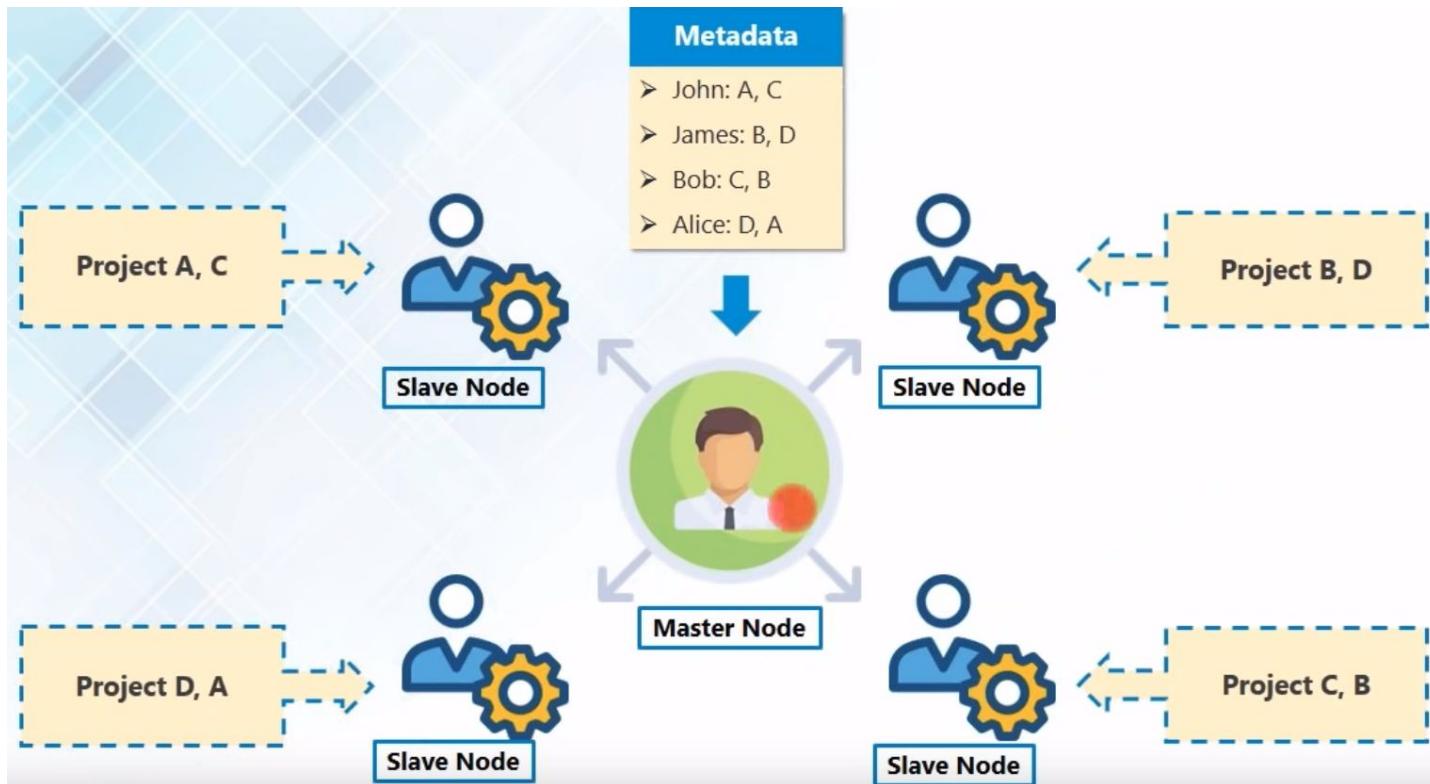


# ARQUITETURA MASTER/SLAVE

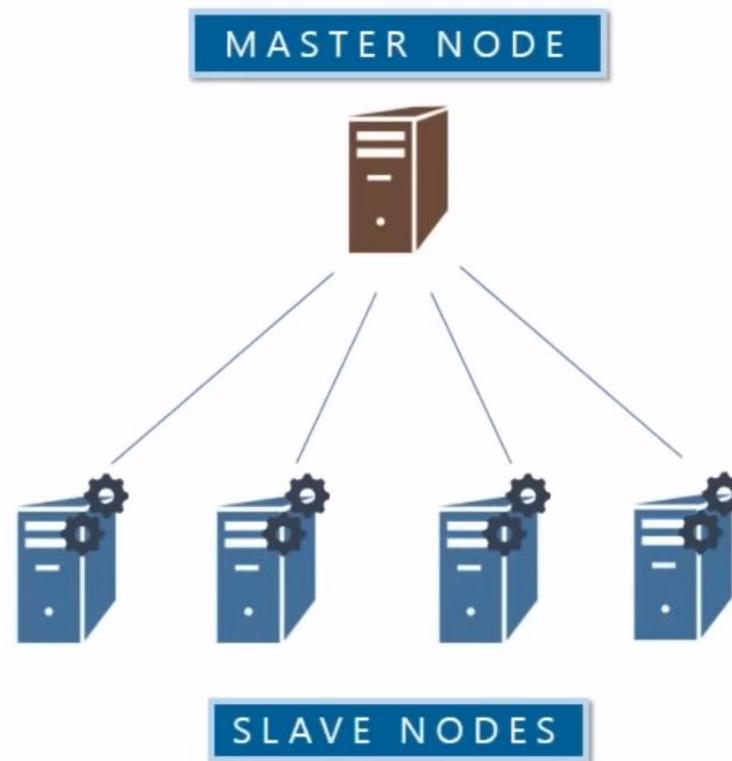
- **John** - Projeto A, C;
- **James** - Projeto B;
- **Alice** - Projeto D;
- **Gerente** - conhece os *deadlines* de entrega e possui os *backups* dos dados.



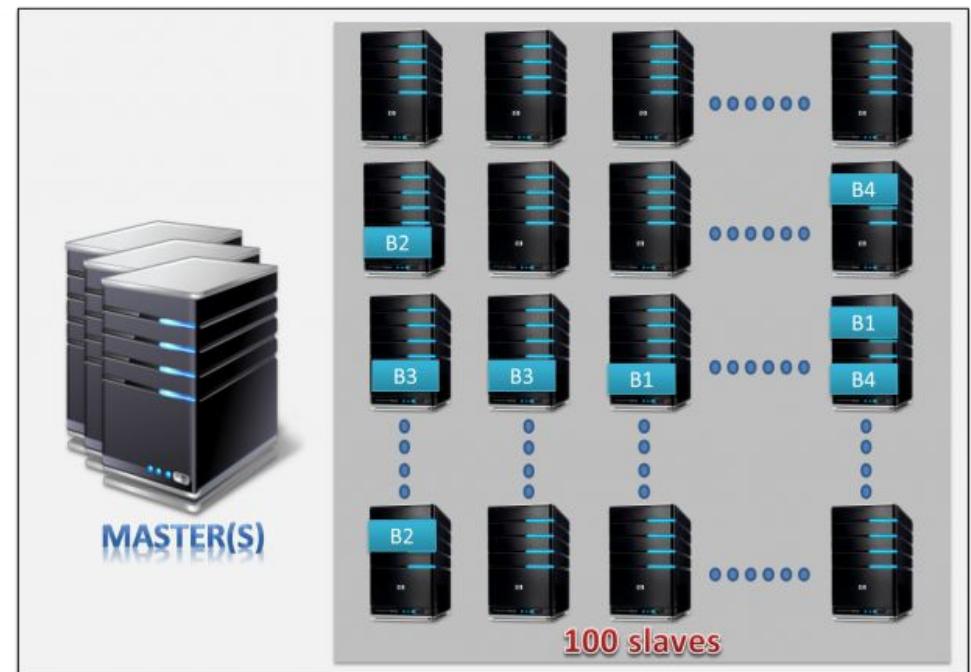
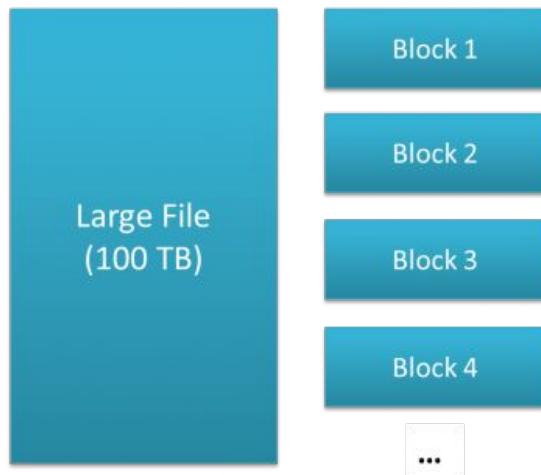
# ARQUITETURA MASTER/SLAVE



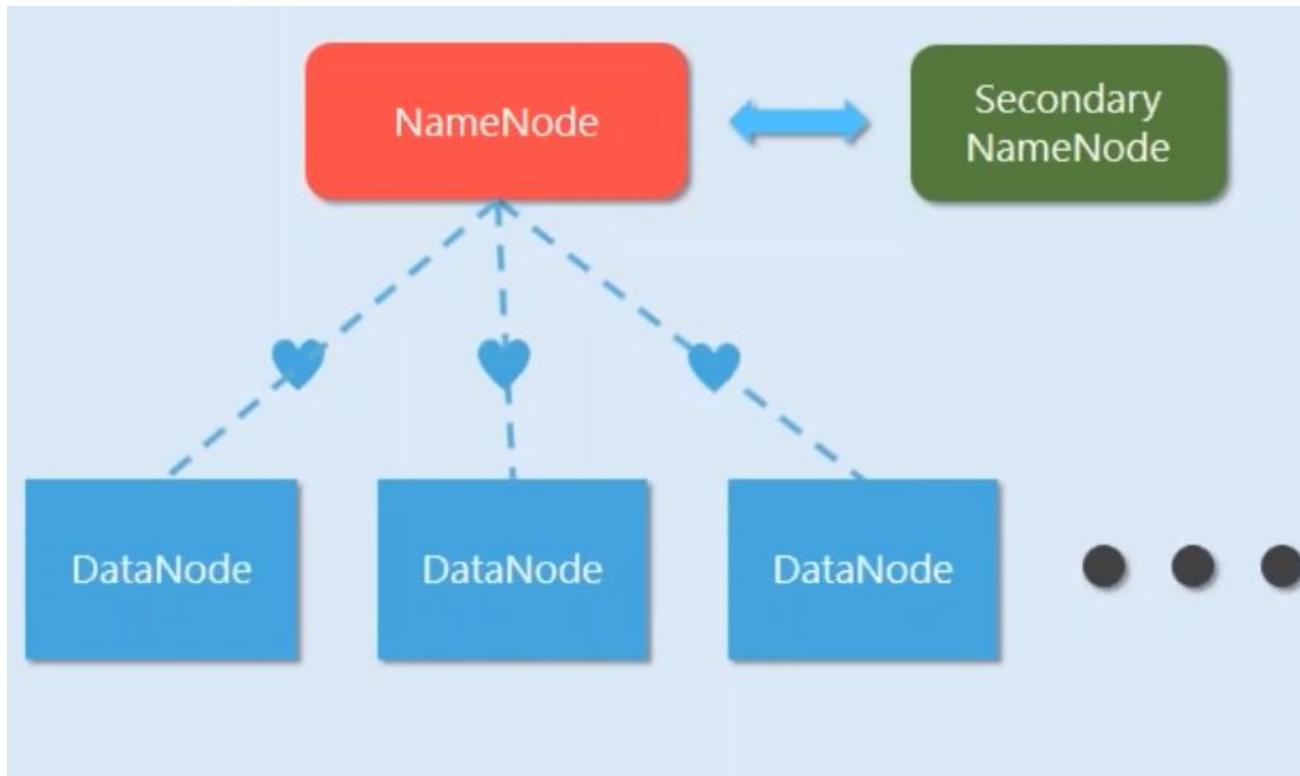
# ARQUITETURA MASTER/SLAVE



# ARQUITETURA MASTER/SLAVE



# HADOOP FILE SYSTEM (HDFS)

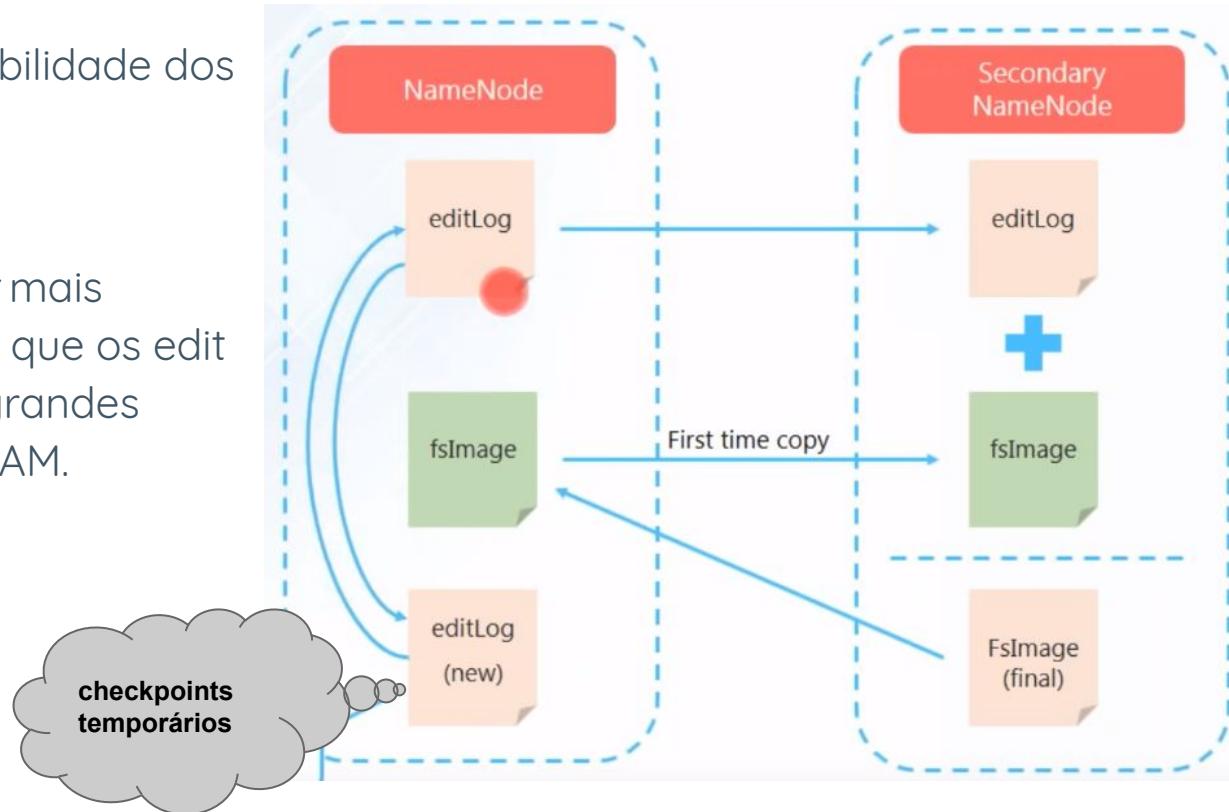


# HDFS - COMPONENTES

- **Namenode:**
  - mantém a árvore do sistema de arquivos e os metadados para todos os arquivos e diretórios (localização dos blocos, permissões, tamanhos, etc);
- **Datanodes:**
  - armazenam e recuperam blocos quando são instruídos por clientes ou *namenode*;
- **Secondary namenode:**
  - não substitui o *namenode*, mas o auxilia. Sua função é obter pontos de verificação dos metadados do sistema de arquivos presentes no *namenode*.

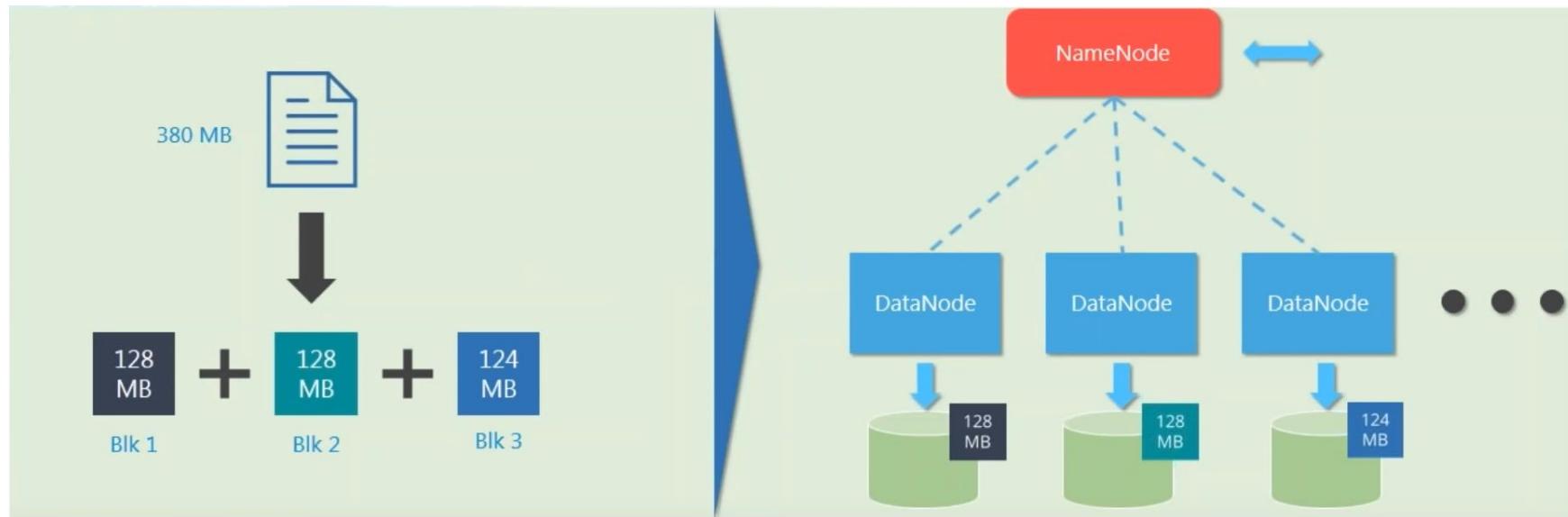
# NAMENODE SECUNDÁRIO

- Assume a responsabilidade dos *checkpoints*;
- Permite um *failover* mais rápido, pois impede que os edit logs fiquem muito grandes consumindo mais RAM.



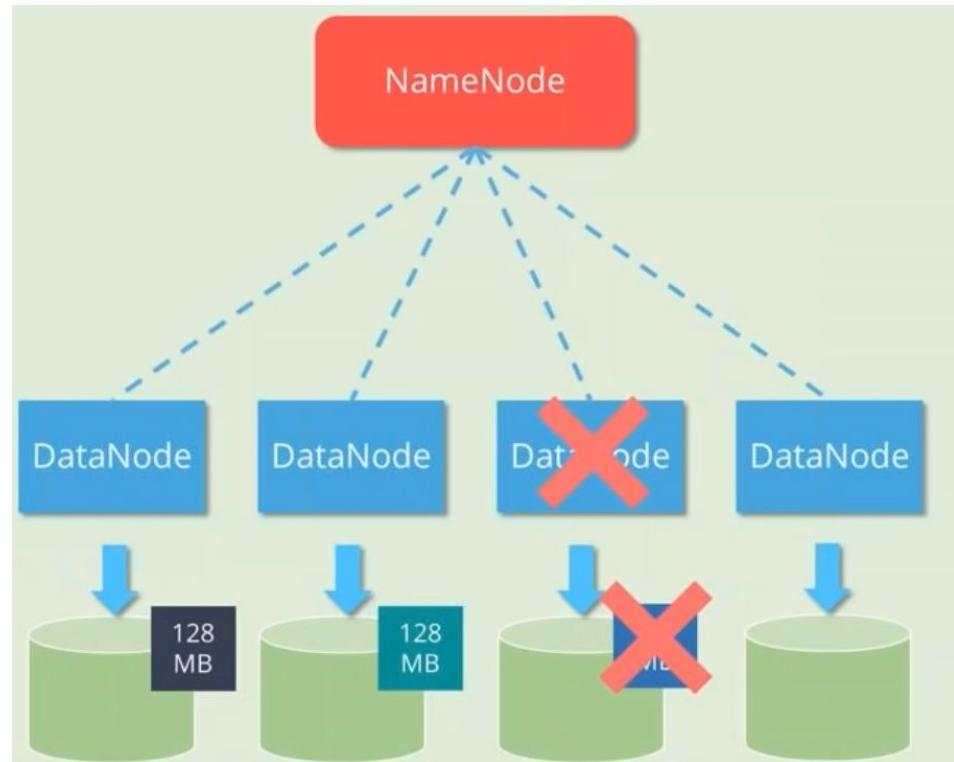
# HDFS - ARMAZENAMENTO NOS DATANODES

- ▶ Arquivos como blocos;
- ▶ O tamanho padrão do bloco na versão 2.x do Hadoop é 128MB;



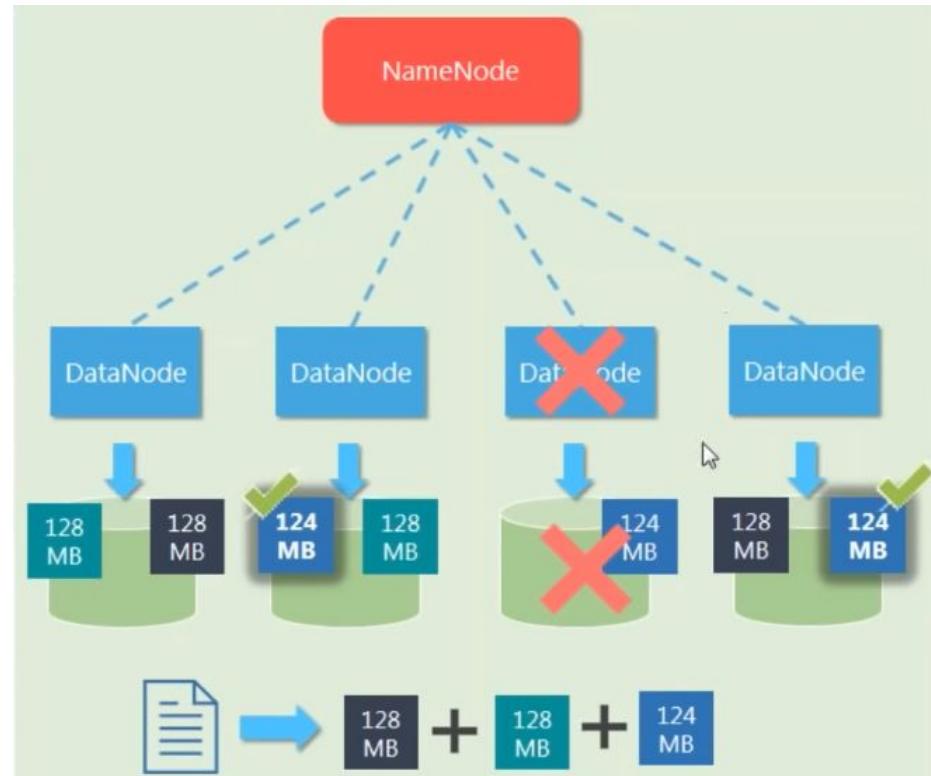
# HDFS - TOLERÂNCIA A FALHAS

- O que ocorre quando um dos datanodes falhar?

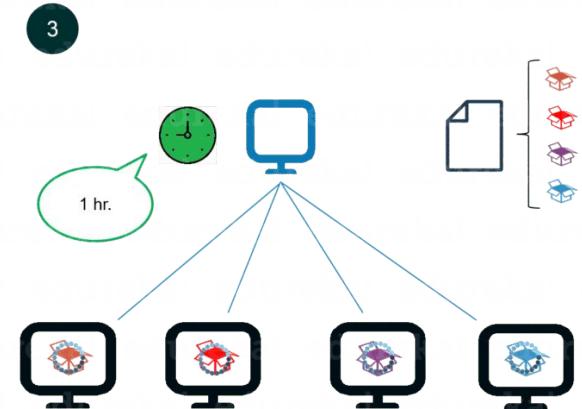
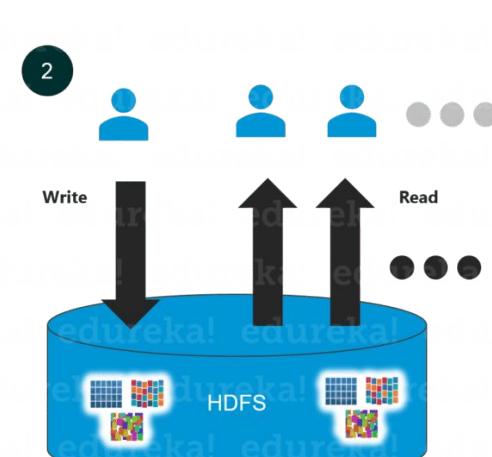
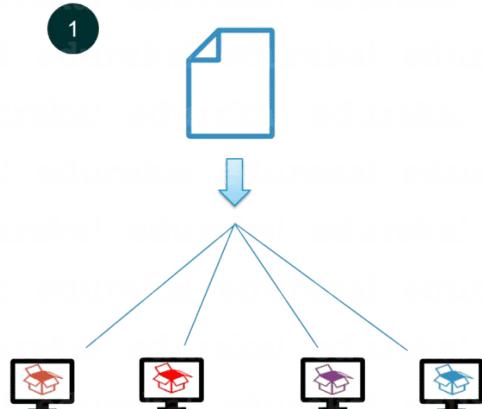


# HDFS - TOLERÂNCIA A FALHAS

- **Solução 1:** replicação dos dados
  - Cada *datanode* é replicado e distribuído em outros *datanodes* (três vezes por padrão)

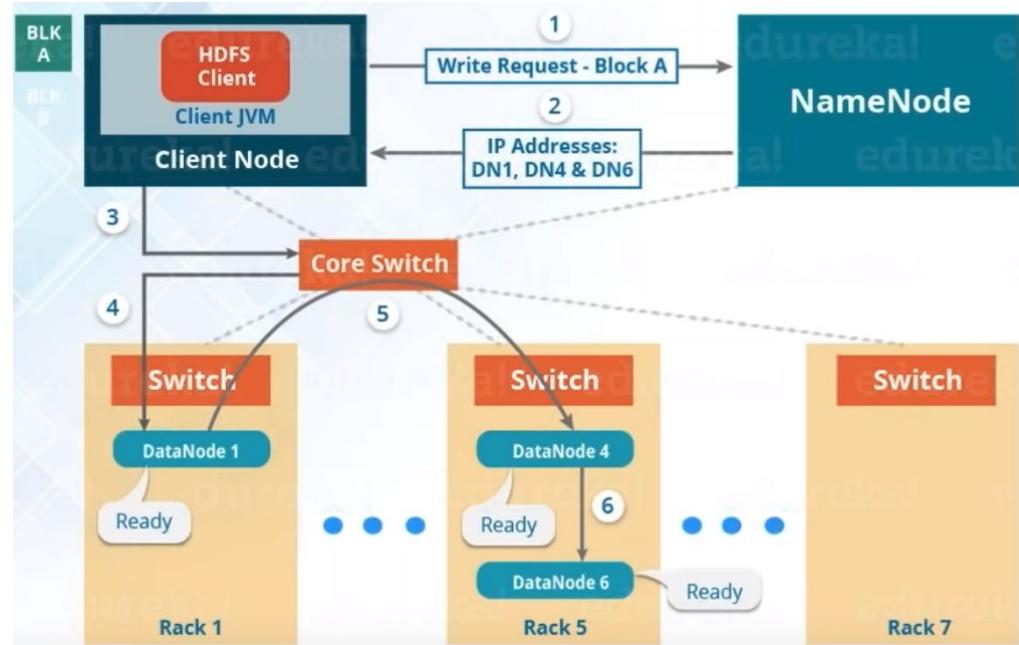


# HADOOP FILE SYSTEM (HDFS)



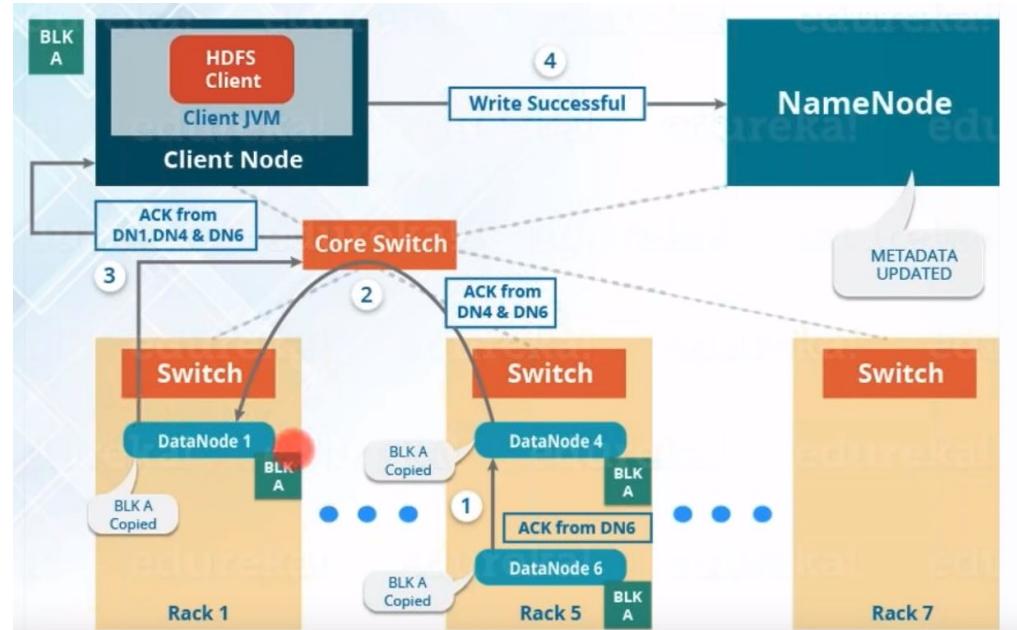
# HDFS - MECANISMO DE ESCRITA

1. Cliente faz requisição;
2. Namenode fornece as informações dos *datanodes*;
3. Cliente solicita a escrita nos *datanodes*;
4. Assim que disponível, o bloco e suas cópias são armazenados.

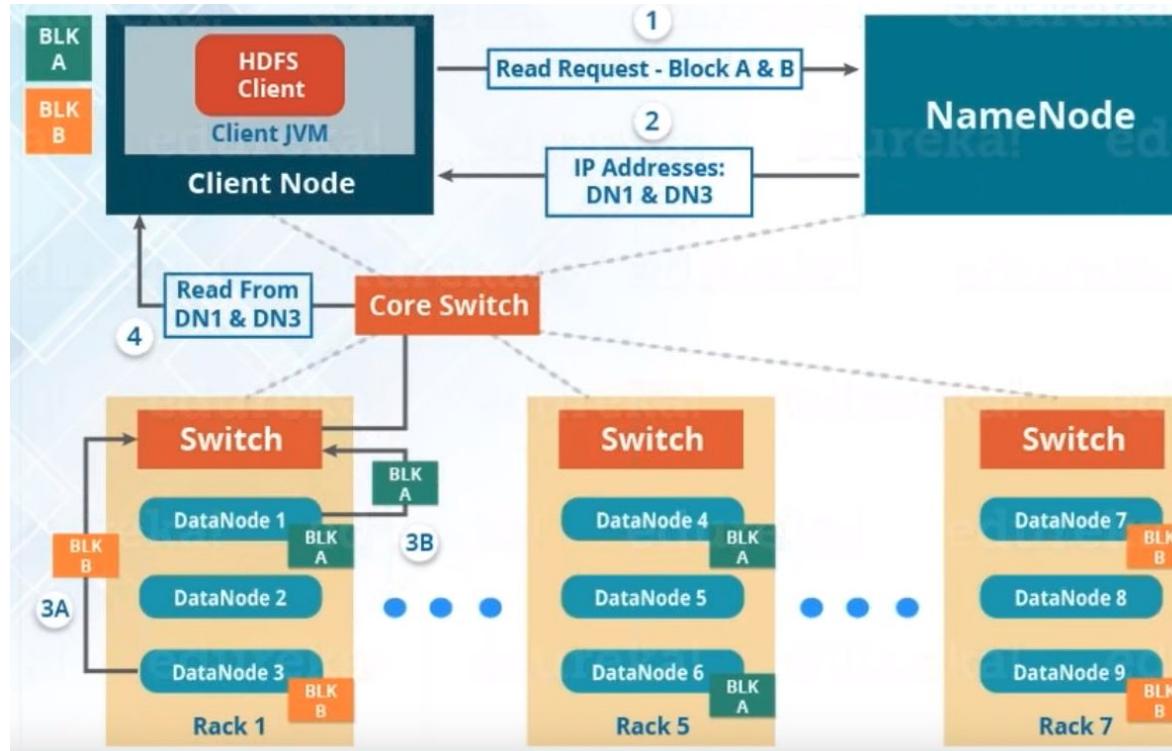


# HDFS - MECANISMO DE ESCRITA

1. O bloco é armazenado nos *datanodes*;
2. As informações de *logs* são enviadas;
3. Cliente recebe a confirmação;
4. *Namenode* atualiza os metadados.



# HDFS - MECANISMO DE LEITURA



## HDFS - OVERVIEW

- Capacidade de armazenamento distribuído de grandes arquivos (petabytes, ex.);
- Suporte a dados de *streaming*;
- Acesso a dados com baixa latência, mesmo em um hardware razoável;
- Escalável (aumentando a capacidade ou a quantidade de nós de cluster)

### 3. INSTALAÇÃO HADOOP

# HADOOP INSTALAÇÃO

- JAVA 6 ou superior (sun JDK);
- SSH deve ser instalada e SSHD deve estar executando para usar scripts hadoop;
- Suporta sistemas Unix e Windows (requer cygwin).

# CONFIGURAÇÃO HADOOP

- Na prática, a configuração é realizada através das propriedades informadas nos seguintes arquivos:
  - core-site.xml
  - hdfs-site.xml
  - mapred-site.xml
  - yarn-site.xml
- Tutoriais:
  - <https://www.devmedia.com.br/hadoop-fundamentos-e-instalacao/29466>
  - <https://www.guru99.com/how-to-install-hadoop.html>
  - <https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster>

# MODOS SUPORTADOS

- **Modo local (standalone):**
  - Sem daemons, tudo executado em uma única JVM;
  - Versão útil para executar programas MapReduce durante o desenvolvimento;
- **Modo pseudo-distribuído:**
  - Hadoop daemons completo onde cada instância hadoop executa em uma JVM;
- **Modo totalmente distribuído:**
  - Hadoop daemons completo é configurado em cluster com máquinas físicas ou virtualizadas, cada qual com um endereço IP.

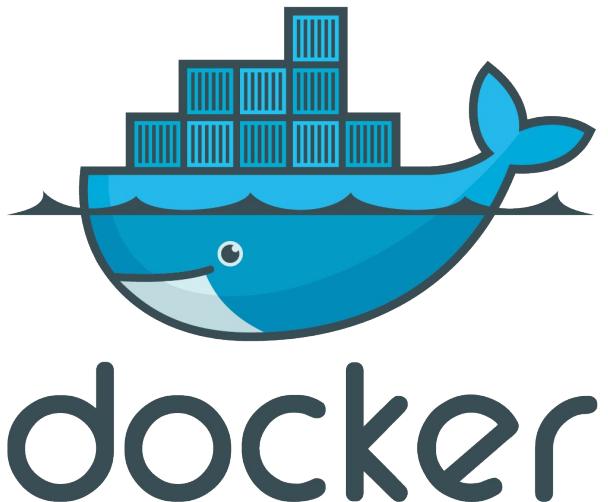
# COMO EXECUTAR APLICAÇÃO HADOOP?

- **Instalação Tradicional:**
  - <<https://hadoop.apache.org>>;
- **Máquina Virtual:**
  - <<https://www.virtualbox.org>>
- **Hortonworks:**
  - <https://br.hortonworks.com>
- **Docker:**
  - <<https://www.docker.com>>

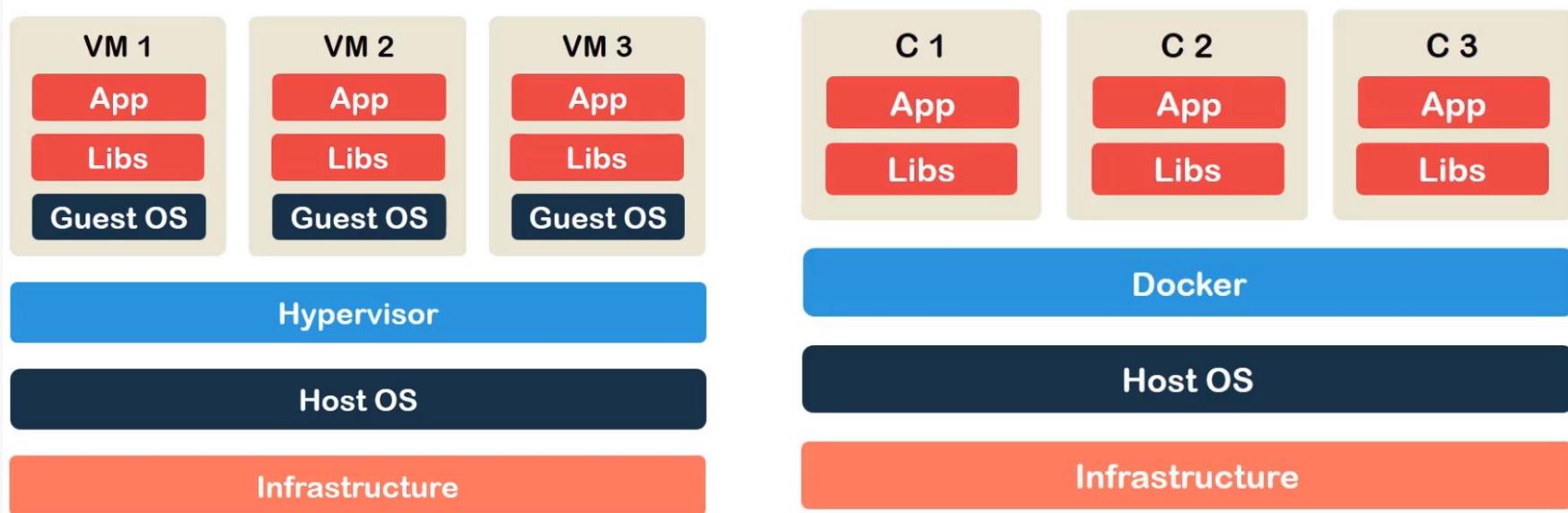
## 4. INTRODUÇÃO AO DOCKER

# INTRODUÇÃO

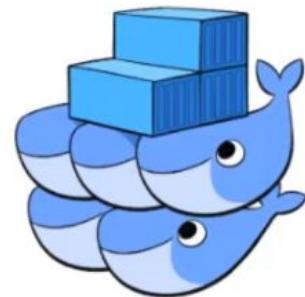
- É uma plataforma aberta e mais popular para usuários construir, preparar e executar aplicações em *containers*;



# VIRTUAL MACHINES vs CONTAINERS



# DOCKER ECOSSISTEMA

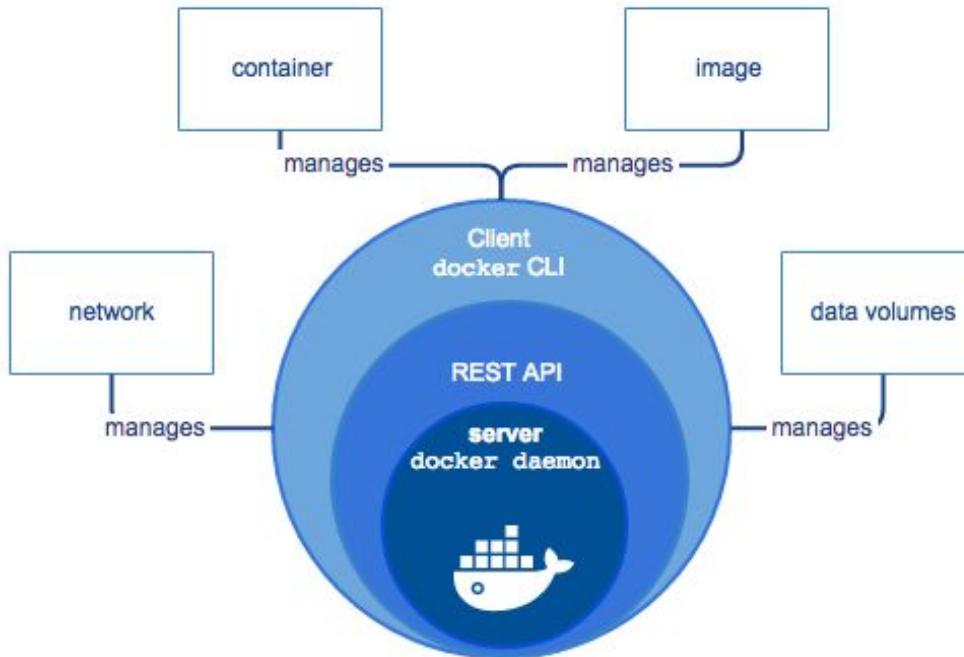


▷ Docker Engine

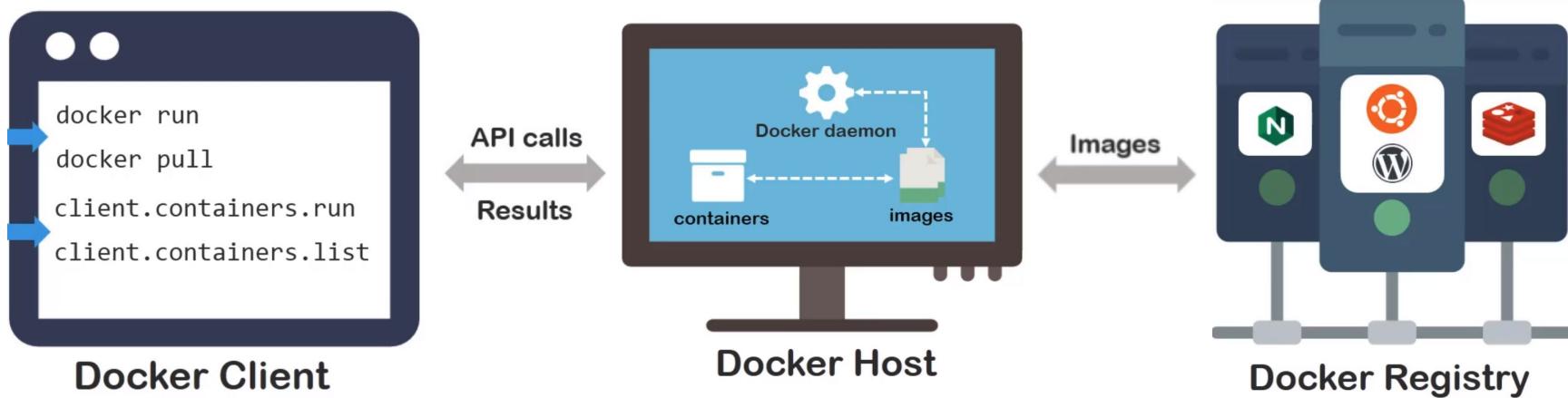
▷ Docker Compose

▷ Docker swarm

# DOCKER ENGINE: ARQUITETURA

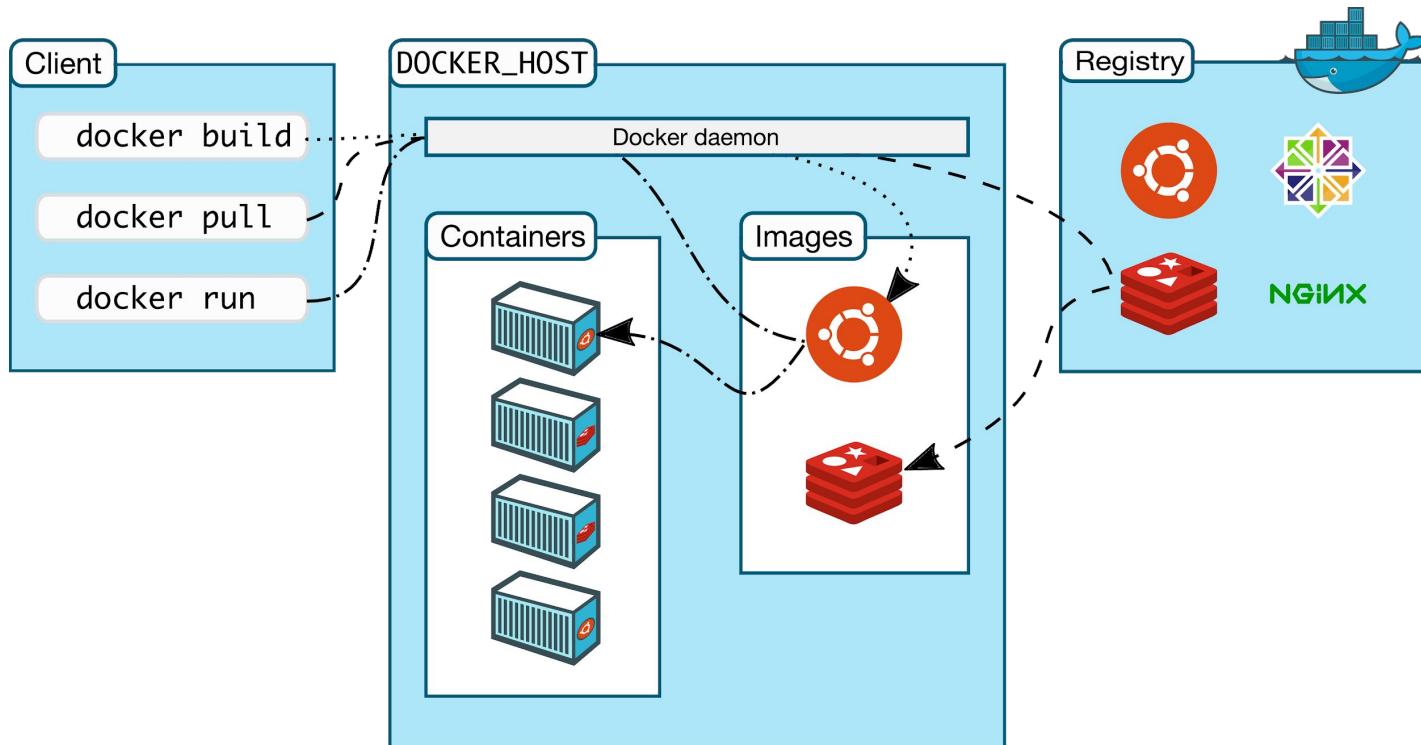


# DOCKER ENGINE: COMPONENTES



- ▷ Docker CLI
- ▷ Docker API
- ▷ Docker daemon
- ▷ Containers
- ▷ Images

# DOCKER ENGINE: FUNCIONAMENTO



# FASES PARA CONTEINERIZAÇÃO



Dockerfile  
(Build)

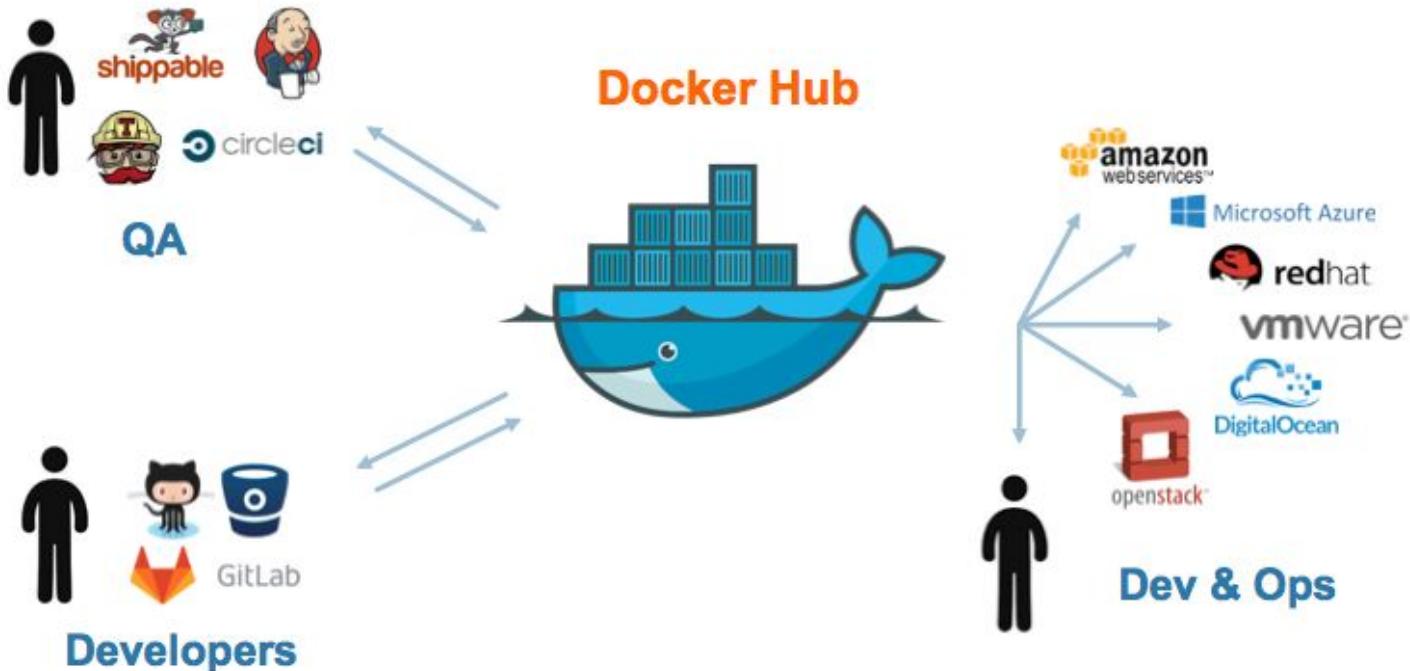


Docker Image  
(Ship)



Containers  
(Run)

# DOCKER HUBs



## 5. ATIVIDADE PRÁTICA

# DOCKER INSTALAÇÃO

- Instalação do docker (Ubuntu);
  - \$bash: sudo apt update
  - \$bash: sudo apt install docker-ce
  - \$bash: docker version



```
Client:          18.09.6
Version:        18.09.6
API version:   1.39
Go version:    go1.10.8
Git commit:    481bc77
Built:          Sat May  4 02:35:57 2019
OS/Arch:        linux/amd64
Experimental:  false
```

- Instalação do docker (Windows e Mac);
  - [https://docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/)
  - <https://docs.docker.com/v17.12/docker-for-mac/install/>

# HADOOP INSTALAÇÃO

- Buscar imagem hadoop:
  - `$bash:docker search hadoop`

NAME	DESCRIPTION	STARS
sequenceiq/hadoop-docker	An easy way to try Hadoop	601
uhopper/hadoop	Base Hadoop image with dynamic configuration...	94
harisekhon/hadoop	Apache Hadoop (HDFS + Yarn, tags 2.2 - 2.8)	50
bde2020/hadoop-namenode	Hadoop namenode of a hadoop cluster	20
izone/hadoop	Hadoop 2.8.5 Ecosystem fully distributed, Ju...	14
bde2020/hadoop-datanode	Hadoop datanode of a hadoop cluster	9
uhopper/hadoop-namenode	Hadoop namenode	9
ibmcom/iop-hadoop	IBM Open Platform with Apache Hadoop	9

- Carregar uma imagem do Docker Hub em sua máquina local.
  - `$bash:docker pull sequenceiq/hadoop-docker`

# HADOOP INSTALAÇÃO

- Carregar arquivo de imagem na máquina local:
  - \$bash: sudo apt update
  - \$bash: docker load < hadoop.tar
  - \$bash: docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
sequenceiq/hadoop-docker	latest	5c3cc170c6bc	3 years ago	1.77GB

- Criar, iniciar e executar comando em um container:
  - \$bash: docker run -it --name ct-hadoop sequenceiq/hadoop-docker /bin/bash
  - \$bash: docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9d08373cc79	sequenceiq/hadoop-docker	"bash"	3 minutes ago	Up 5 seconds	2122/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp	focused_habit

# INTERAGINDO COM HADOOP

- Iniciar daemons do hadoop via script:

- \$bash: /etc/bootstrap.sh

- Verificar a daemons:

- \$bash: jps



```
1415 Jps
339 ResourceManager
1026 SecondaryNameNode
1286 NodeManager
862 DataNode
740 NameNode
```

- Acesse no browser
  - **Namenode** → <http://172.17.0.2:50070>
  - **Datanode** → <http://172.17.0.2:50075>
  - **Namenode secondary** → <http://172.17.0.2:50090>
  - **Resource Manager** → <http://172.17.0.2:8042>
  - **All applications** → <http://172.17.0.2:8088/cluster>

# INTERAGINDO COM HADOOP

- Acesse o diretório raiz da instalação do hadoop:

- `$bash: cd $HADOOP_PREFIX && ls`

```
LICENSE.txt  NOTICE.txt  README.txt  bin  etc  include  input  lib  libexec  logs  sbin  share
```

- Abram todos os arquivos de texto no diretório `$HADOOP_PREFIX`:

- `$bash: cat *.txt`

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

# VISUALIZANDO ARQUIVOS DE CONFIGURAÇÃO

- core-site.xml:
  - \$bash: vi \$HADOOP\_PREFIX/etc/hadoop/core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://ad795e0b131b:9000</value>
  </property>
</configuration>
```

- hdfs-site.xml:
  - \$bash: vi \$HADOOP\_PREFIX/etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

# VISUALIZANDO ARQUIVOS DE CONFIGURAÇÃO

- mapred-site.xml:
  - \$bash: vi \$HADOOP\_PREFIX/etc/hadoop/mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- yarn-site.xml:
  - \$bash: vi \$HADOOP\_PREFIX/etc/hadoop/yarn-site.xml

# INTERAGINDO COM HADOOP

- Acesse o diretório “/bin”:
  - \$bash: cd \$HADOOP\_PREFIX/bin

```
container-executor  hadoop.cmd  hdfs.cmd  mapred.cmd  test-container-executor  yarn.cmd  
hadoop           hdfs        mapred      rcc          yarn
```

- Listando diretórios do HDFS;
  - \$bash: \$HADOOP\_PREFIX/bin/hadoop fs -ls /

```
Found 1 items  
drwxr-xr-x  - root supergroup          0 2015-07-22 11:17 /user
```

# INSERINDO ARQUIVOS NO HDFS

- Copie os arquivos locais para container:
  - `obs: abra um novo terminal (fora do bash do container);`
  - `$bash: docker cp -a data ct-hadoop:/data`
- Crie um novo diretório no HDFS:
  - `volte ao terminal no container;`
  - `$bash: $HADOOP_PREFIX/bin/hadoop fs -mkdir -p /user/word_count/input`
- Adicione diretórios do container para o HDFS:
  - `$bash: $HADOOP_PREFIX/bin/hadoop fs -put /data /user/word_count`
  - `&bash: $HADOOP_PREFIX/bin/hadoop fs -ls /user/word_count/data`

# INSERINDO ARQUIVOS NO HDFS

- Copie os arquivos locais para container:
  - `obs: abra um novo terminal (fora do bash do container);`
  - `$bash: docker cp -a data ct-hadoop:/data`
- Crie um novo diretório no HDFS:
  - `volte ao terminal no container`
  - `$bash: ls /`

```
bin boot data dev etc home lib lib64 media mnt opt
```

# INSERINDO ARQUIVOS NO HDFS

- Crie um diretório no HDFS:

- \$bash: \$HADOOP\_PREFIX/bin/hadoop fs -mkdir -p /user/word\_count/
  - \$bash: \$HADOOP\_PREFIX/bin/hadoop fs -ls /user/

```
drwxr-xr-x  - root supergroup      0 2015-07-22 11:17 /user/root
drwxr-xr-x  - root supergroup      0 2019-05-21 09:50 /user/word_count
```

- Adicione arquivos do container para o HDFS:

- \$bash: \$HADOOP\_PREFIX/bin/hadoop fs -put /data /user/word\_count
  - &bash: \$HADOOP\_PREFIX/bin/hadoop fs -ls /user/word\_count/data

```
Found 2 items
-rw-r--r--  1 root supergroup    2137 2019-05-21 09:54 /user/word_count/data/data_science.txt
-rw-r--r--  1 root supergroup    5195 2019-05-21 09:54 /user/word_count/data/mercado_ti.txt
```

# EXECUTANDO MAPREDUCE

- Usando grep :

- \$bash: \$HADOOP\_PREFIX/bin/hadoop jar \$HADOOP\_PREFIX/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.0.jar grep input output 'dfs[a-z.]+'
- \$HADOOP\_PREFIX/bin/hadoop fs -cat output/part-r-00000

```
bash-4.1# $HADOOP_PREFIX/bin/hadoop fs -cat output/part-r-00000
6      dfs.audit.logger
4      dfs.class
3      dfs.server.namenode.
2      dfs.period
2      dfs.audit.log.maxfilesize
2      dfs.audit.log.maxbackupindex
1      dfsmetrics.log
1      dfsadmin
1      dfs.servers
1      dfs.replication
1      dfs.file
```

# EXECUTANDO MAPREDUCE

- Em arquivos:

- \$HADOOP\_PREFIX/bin/hadoop  
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.0.jar  
/user/word\_count/data/ /user/word\_count/data/output;  
○ \$HADOOP\_PREFIX/bin/hadoop fs -ls /user/word\_count/data/output

jar

wordcount

```
0 2019-05-21 10:49 /user/word_count/data/output/_SUCCESS
5873 2019-05-21 10:49 /user/word_count/data/output/part-r-00000
```

- Analise o resultado:

- \$HADOOP\_PREFIX/bin/hadoop fs -cat /user/word\_count/data/output/part-r-00000

```
"Acreditávamos 1
"In-memory 1
"Isso 1
"Nossa 1
"O 4
"o 1
```

# EXECUTANDO MAPREDUCE

- Acesse o diretório das aplicações mapreduce do hadoop:
  - \$bash: \$HADOOP\_PREFIX/shared/hadoop/mapreduce

```
hadoop-mapreduce-client-hs-plugins-2.7.0.jar      hadoop-mapreduce-examples-2.7.0.jar
hadoop-mapreduce-client-jobclient-2.7.0-tests.jar   lib
hadoop-mapreduce-client-jobclient-2.7.0.jar        lib-examples
hadoop-mapreduce-client-shuffle-2.7.0.jar          sources
```

# REFERÊNCIAS PRINCIPAIS

- WHITE, Tom. **Hadoop: The definitive guide.** " O'Reilly Media, Inc.", 2012.
- RADTKA, Zachary; MINER, Donald. **Hadoop with Python.** O'Reilly Media, 2015.
- CHU, Cheng-Tao et al. Map-reduce for machine learning on multicore. In: **Advances in neural information processing systems.** 2007. p. 281-288.
- DEAN, Jeffrey; GHEMAWAT, Sanjay. MapReduce: a flexible data processing tool. **Communications of the ACM**, v. 53, n. 1, p. 72-77, 2010.

# REFERÊNCIAS COMPLEMENTARES

- **Curso Edureka.** Hadoop Tutorial: All you need to know about Hadoop! Acesso em <<https://www.edureka.co/blog/hadoop-tutorial>>;
- **Curso Guru.** Big Data Hadoop Tutorial for Beginners: Learn in 7 Days!. acesso em <<https://www.guru99.com/bigdata-tutorials.html>>;
- **Curso Cetax.** Apache Hadoop Essentials. acesso disponível em <<https://www.cetax.com.br/curso-de-apache-hadoop-essentials/>>



# OBRIGADO!

## Dúvidas?

Você pode me encontrar em

- ▶ [nickssonarais@gmail.com](mailto:nickssonarais@gmail.com)
- ▶ [\(84\) 9 9990-4373](tel:(84)9990-4373)