



INSIGHT



# HADOOP/YARN

Também pode conter um subtítulo logo abaixo

1. Hadoop
2. YARN
3. Aula Prática

# 1. INTRODUÇÃO

# MAPREDUCE V1.0

- Antes 2012.
  - Usuário poderiam escrever programas usando linguagens de script ou de transformação de dados.



**Hadoop v1.0**

**MapReduce**

Data Processing  
& Resource Management

**HDFS**

Distributed File Storage

# MAPREDUCE V1.0

- Antes 2012.
  - Yahoo enfrentou problemas para manusear processamento stream e iterativo de enormes conjuntos de dados na infraestrutura do Hadoop v1.0.



**Hadoop v1.0**

**MapReduce**

Data Processing  
& Resource Management

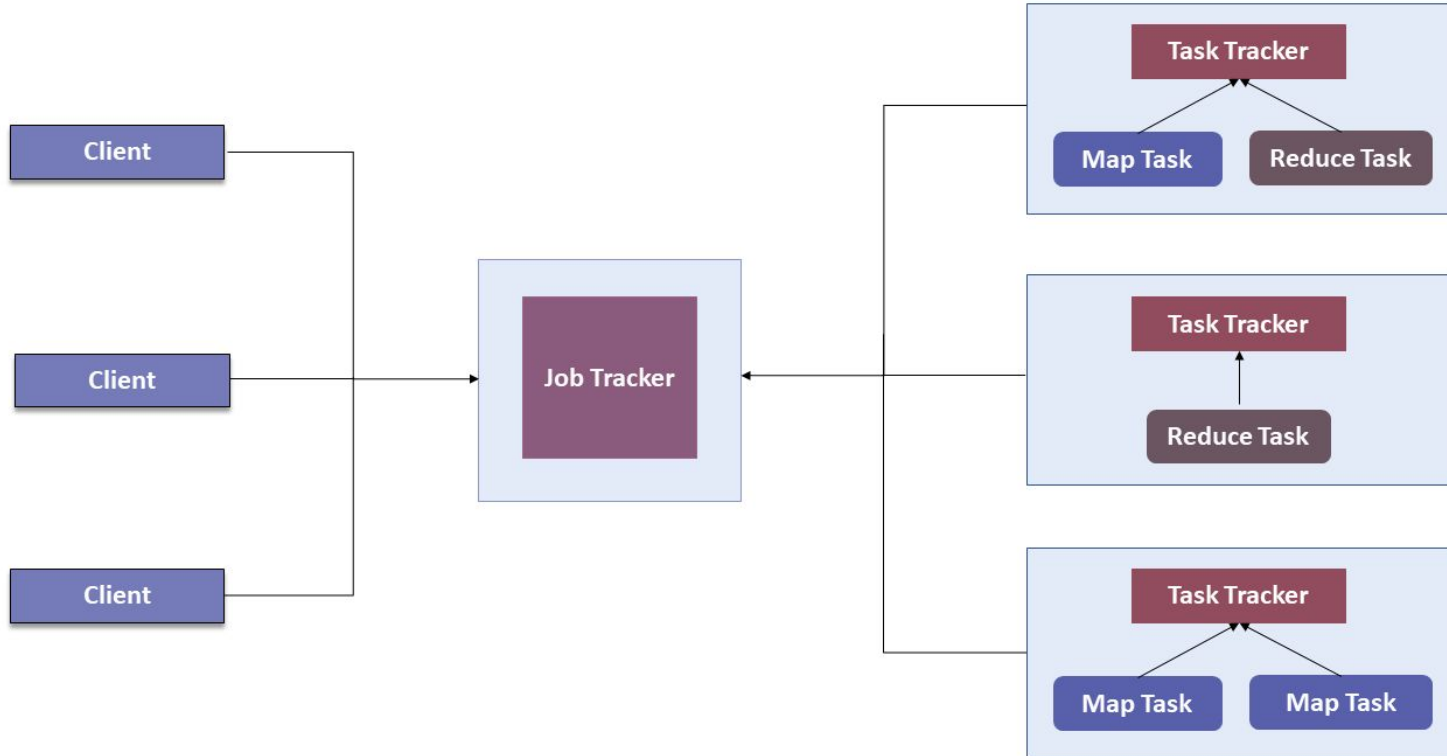
**HDFS**

Distributed File Storage

# MÓDULOS PRINCIPAIS HADOOP V1.0

- HDFS:
  - Namenode;
  - Secondary namenode;
  - Datanode;
- MapReduce:
  - JobTracker:
    - aloca recursos e monitora jobs;
  - TaskTracker:
    - executa e fornece status de MapReduce jobs.

# HADOOP - VERSION 1.0





## LIMITAÇÕES HADOOP V1.0

- Devido ao único JobTracker, a **escalabilidade** se tornou um gargalo;
  - não poderíamos ter um cluster com mais de 4000 nós;
  - não poderíamos executar mais do que 40000 tasks.
- Problemas de **indisponibilidade** pois só existia um único JobTracker;
  - em caso de falha, jobs deveria ser submetidos pelos usuários;

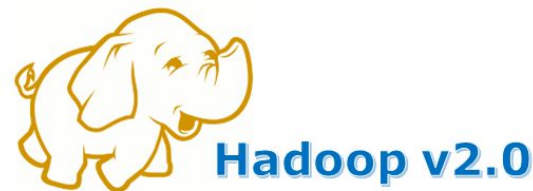
# LIMITAÇÕES HADOOP V1.0

- Problemas para **reutilização de recursos**:
  - slots map e reduce eram pré-definidos para cada TaskTracker;
- Limitações na execução:
  - análise de performance em tempo real;
  - consultas ad-hoc.

## 2. YARN

# YET ANOTHER RESOURCE NEGOTIATOR (YARN)

- É um sistema para gerência de recursos do Hadoop e de outros frameworks;
- YARN surgiu para melhorar a implementação do MapReduce, fornecendo recursos para gerenciar aplicações.



**MapReduce**

**Other Data  
Processing  
Frameworks**

**YARN**

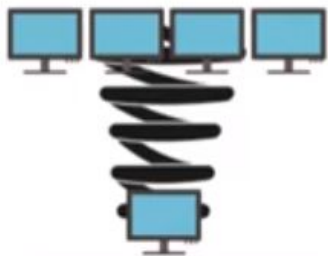
Resource Management

**HDFS**

Distributed File Storage

# YARN

## Escalabilidade



Cluster com mais do que 10.000 nós e mais do que 100.000 tarefas

## Compatibilidade



Aplicações desenvolvidas para hadoop executa YARN sem qualquer perturbação ou problema de indisponibilidade.

## Reutilização de Recursos



Permite a alocação de recursos de cluster para melhorar a utilização de recursos.

## *Multitenancy*

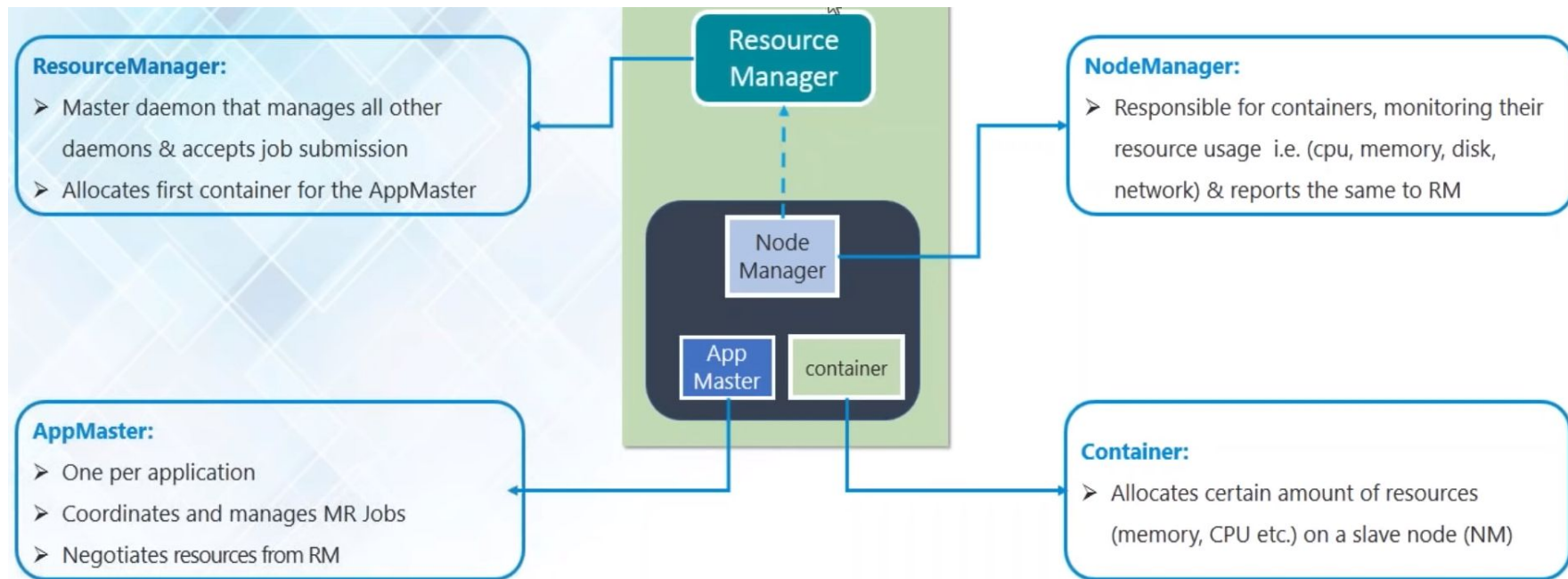


Pode usar propriedades open source para acessar dados e realizar análise de tempo real e consultas ad-hoc.

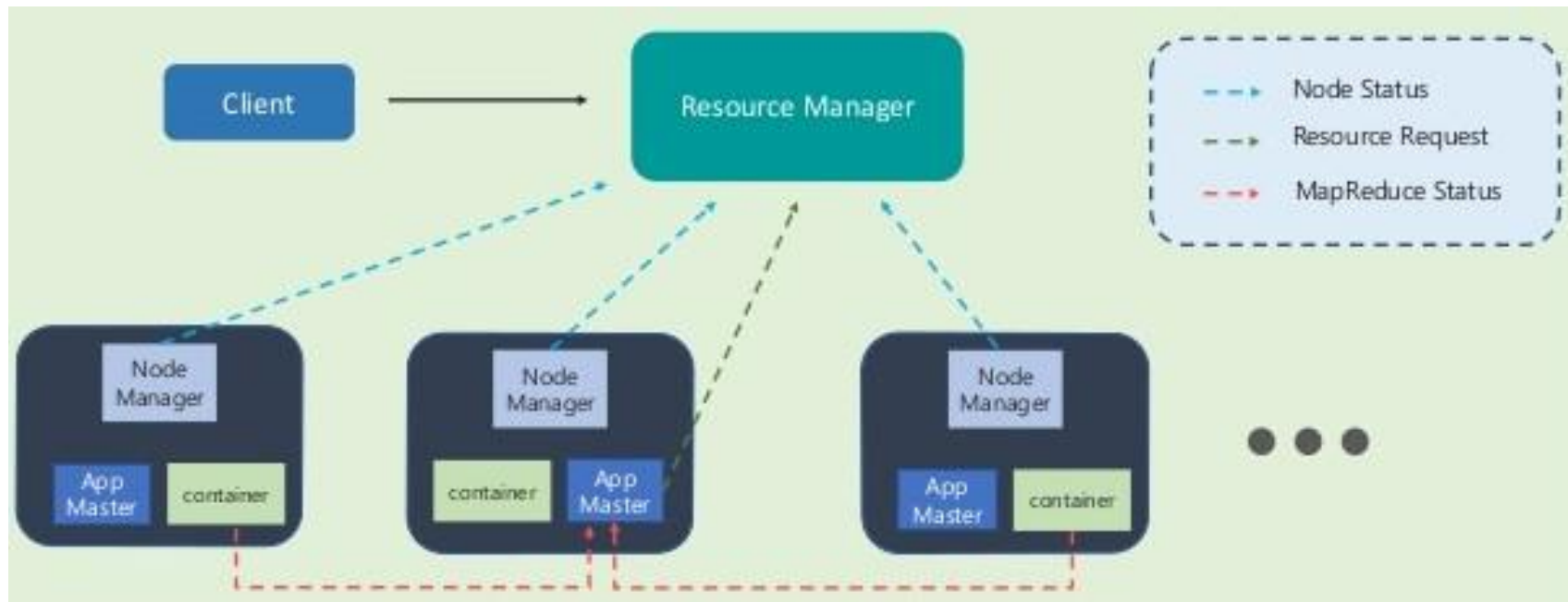
# YARN

- Fornece dois serviços básicos via 2 tipos de daemons:
  - **Resource Manager** (um por cluster), responsável pelo controle de recursos e escalonamento de aplicações.
  - **Node Manager** (um para cada nó), responsável por monitorar os nós e containers, isto é, recursos como CPU, memória, disco e rede.

# YARN - COMPONENTES

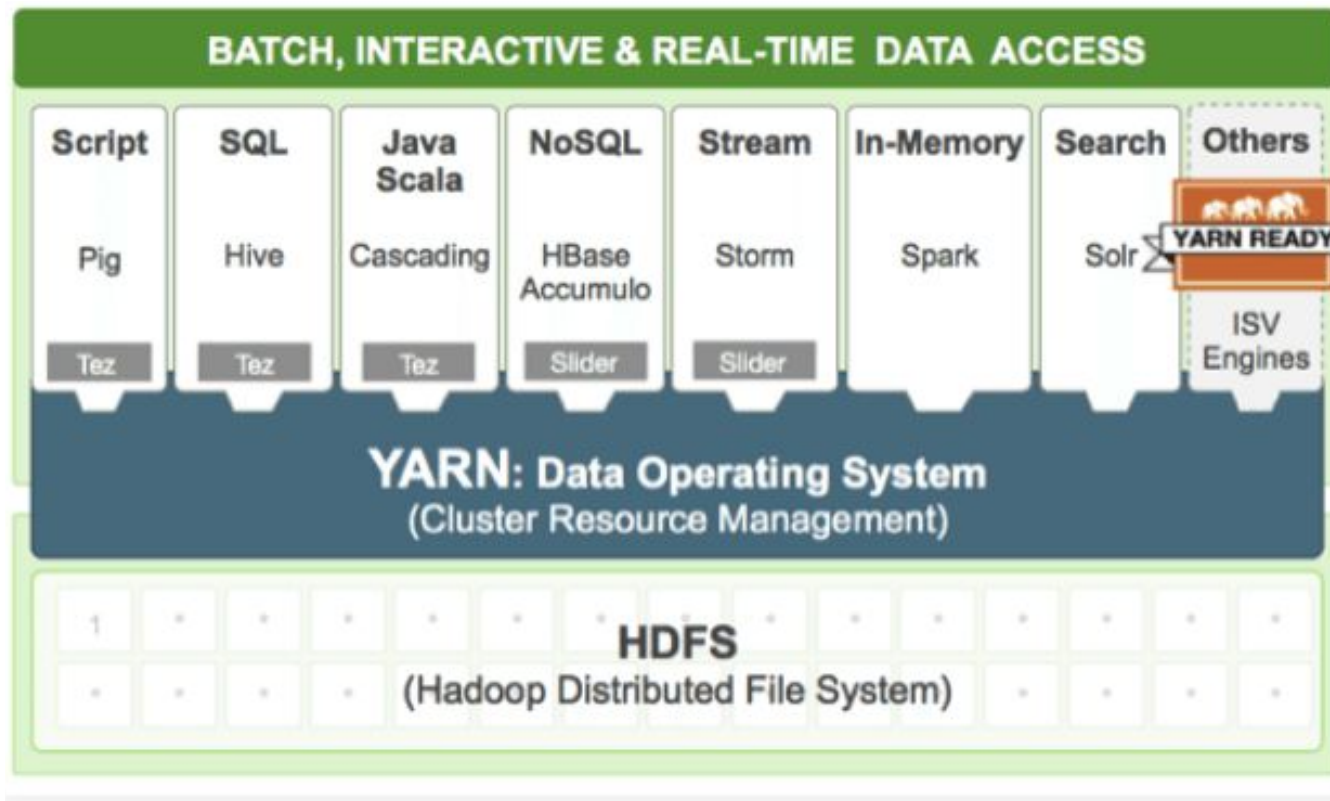


# YARN - COMPONENTES

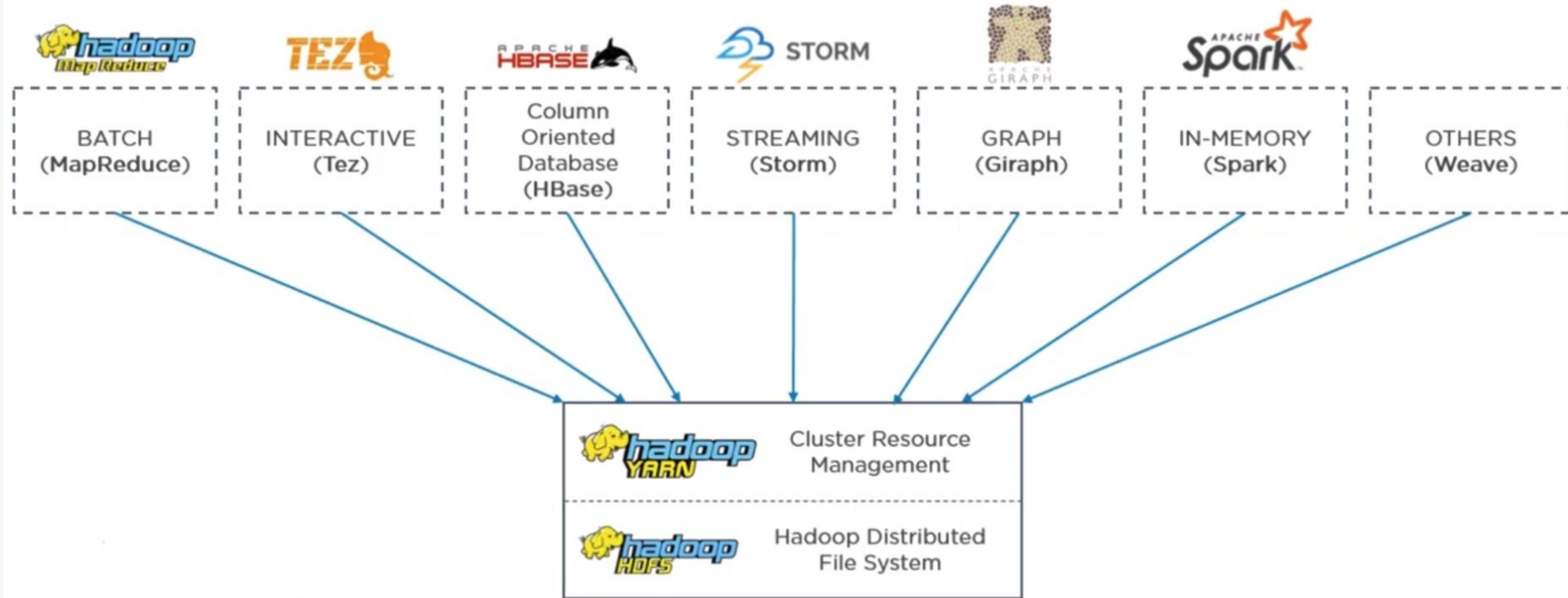




# FRAMEWORKS QUE EXECUTAM SOB O YARN



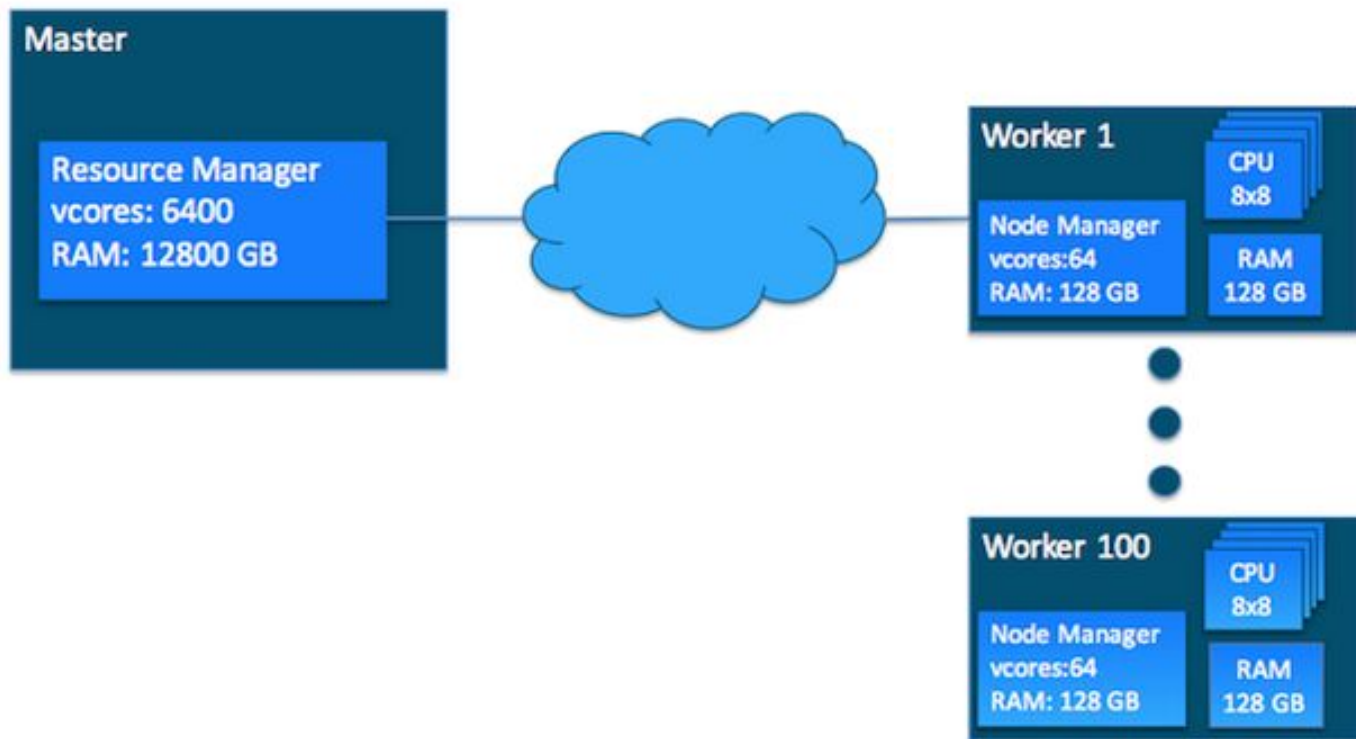
# FRAMEWORKS QUE EXECUTAM SOB O YARN



# HADOOP V1.0 vs HADOOP V2.0

Hadoop 1 (contras)	Hadoop 2 (prós)
<ul style="list-style-type: none"><li>• Um único JobTracker para todas as aplicações (ponto único de falha);</li><li>• Um namenode e um namespaces;</li><li>• Suporta apenas uma versão de MapReduce;</li><li>• Em caso de falha no namenode, a recuperação era manual.</li></ul>	<ul style="list-style-type: none"><li>• Usa o ResourceManager por cluster e executa ApplicationMaster por instância de aplicação;</li><li>• Múltiplos namenodes e namespaces;</li><li>• Suporta diferentes versões do MapReduce em um único cluster;</li><li>• Em caso de falha no namenode, namenode standby é iniciado.</li></ul>

# VISÃO GLOBAL DO RESOURCE MANAGER E NODE MANAGER



# VISÃO GLOBAL DOS CONTAINERS

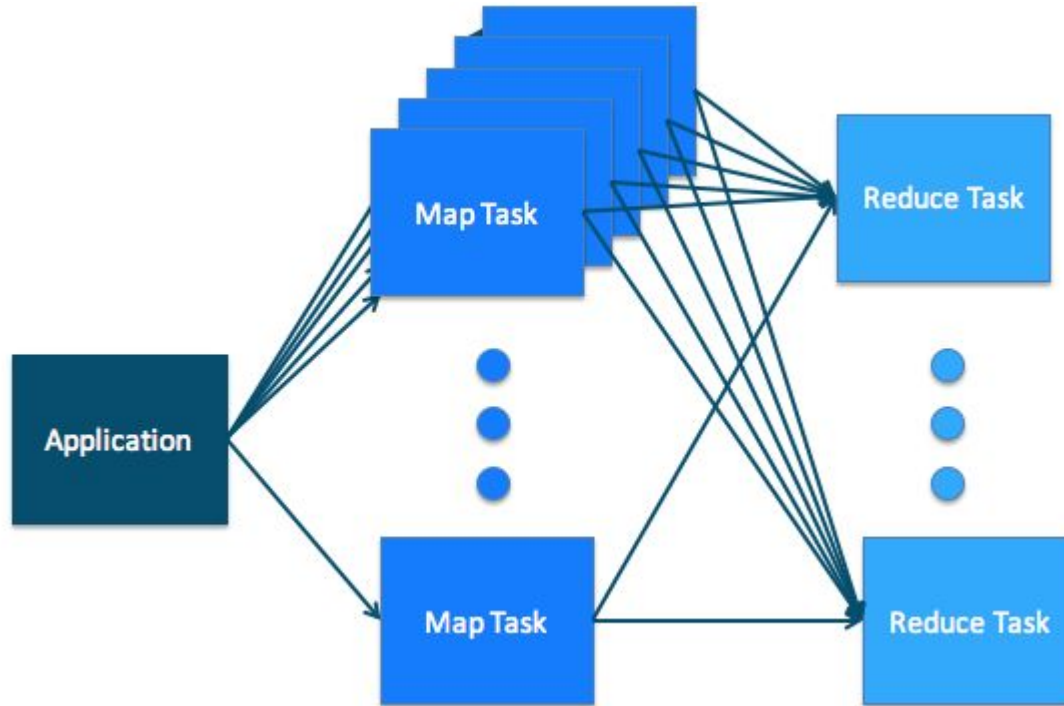
Container  
vcore request: 1  
memory request: 8 GB

Container  
vcore request: 1  
memory request: 8 GB

## PROCESS



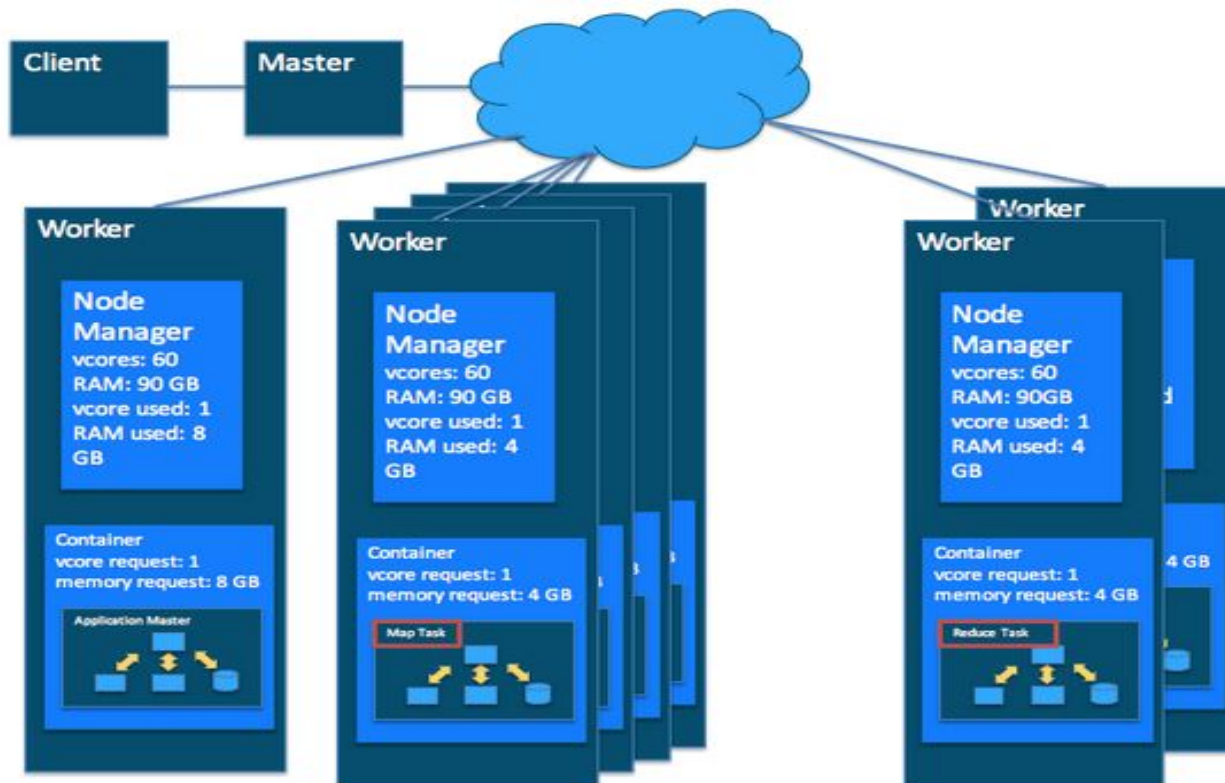
# VISÃO GLOBAL DAS APLICAÇÕES



# VISÃO GLOBAL DOS COMPONENTES

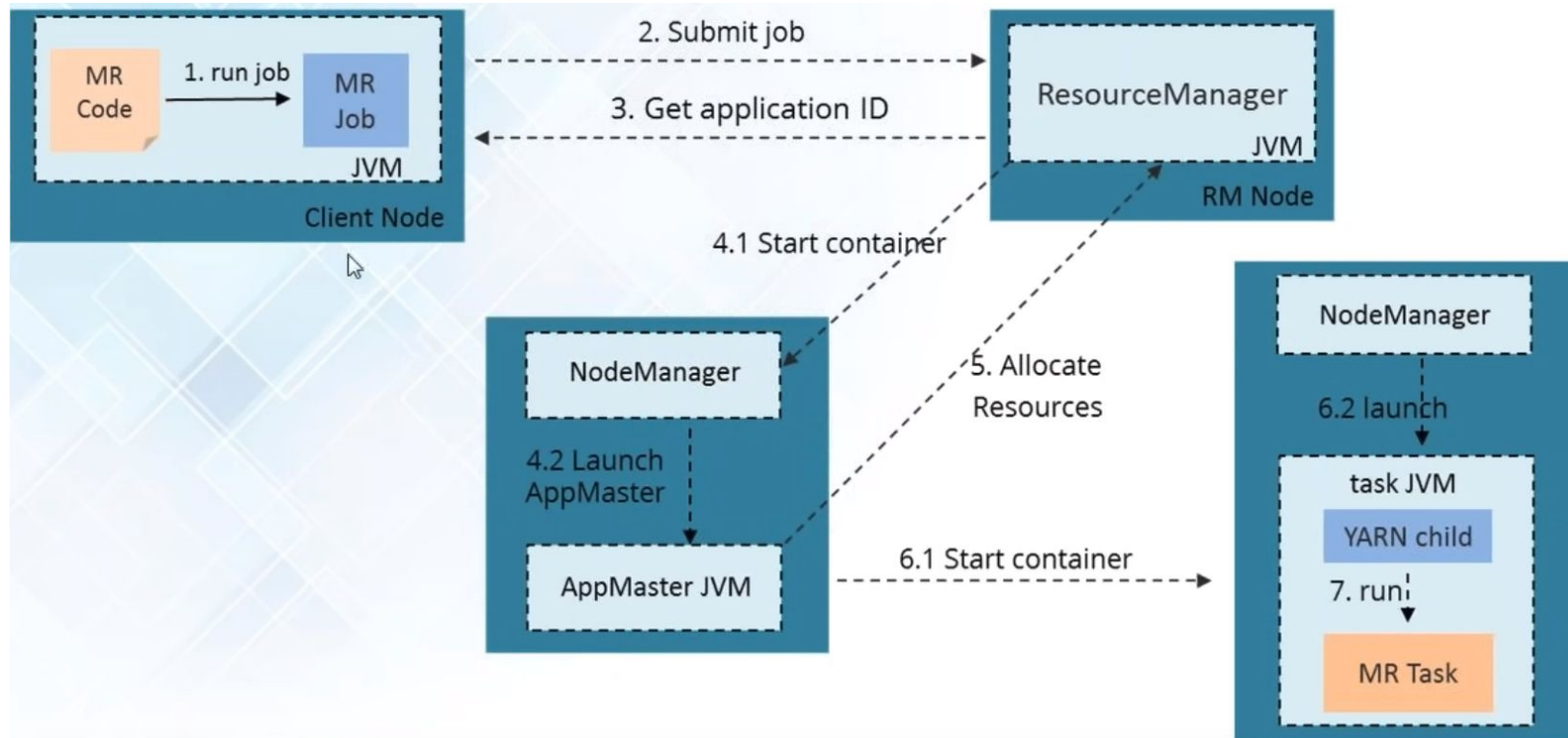


# VISÃO GLOBAL DE TODOS OS COMPONENTES

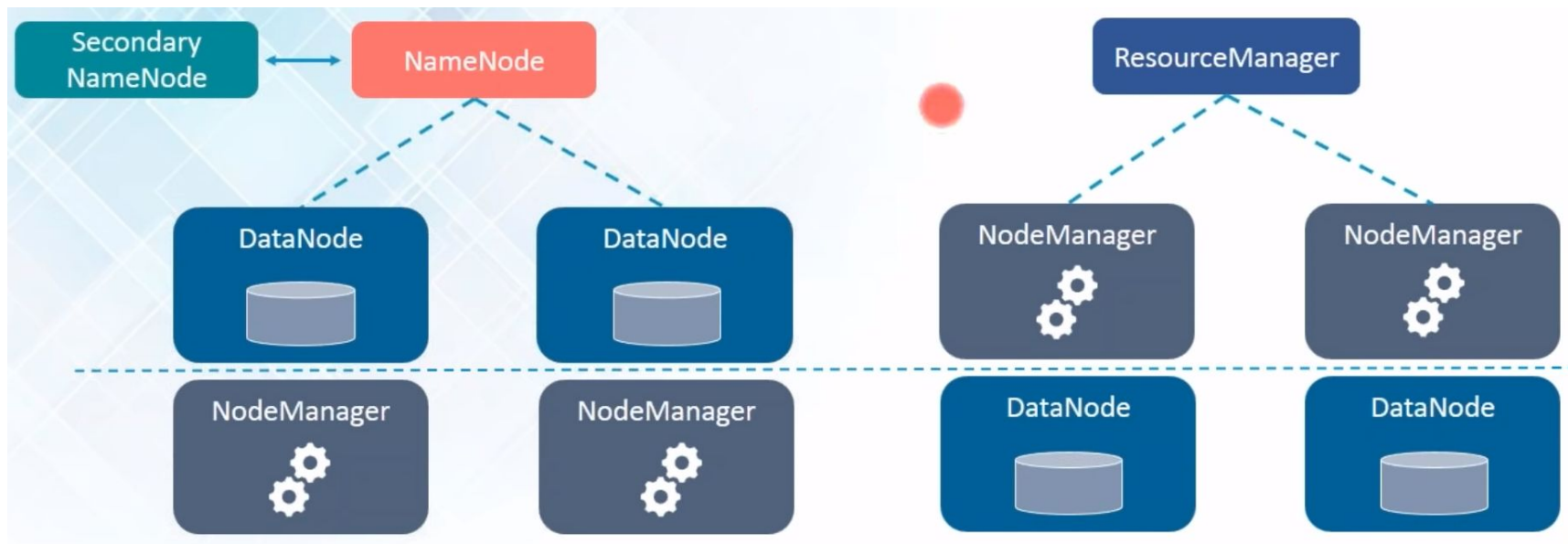




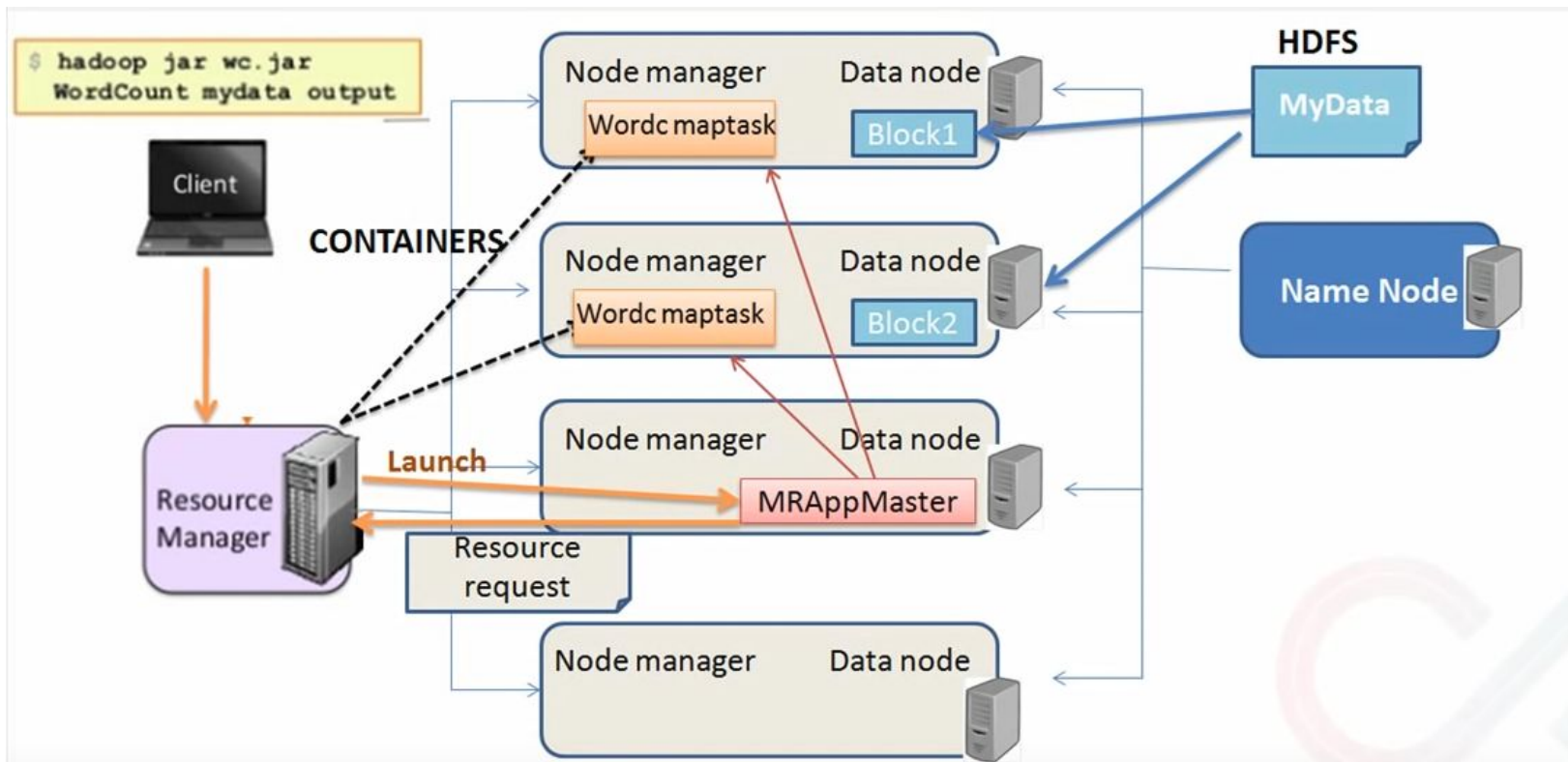
# YARN - FLUXO DE TRABALHO



## HDFS E YARN



## MODO TOTALMENTE DISTRIBUÍDO



# COMO É REALIZADA A CONFIGURAÇÃO DO YARN?

- yarn-site.xml:

```
<property>
<description>
  Number of seconds after an application finishes before the nodemanager's
  DeletionService will delete the application's localized file directory
  and log directory.

  To diagnose Yarn application problems, set this property's value large
  enough (for example, to 600 = 10 minutes) to permit examination of these
  directories. After changing the property's value, you must restart the
  nodemanager in order for it to have an effect.

  The roots of Yarn applications' work directories is configurable with
  the yarn.nodemanager.local-dirs property (see below), and the roots
  of the Yarn applications' log directories is configurable with the
  yarn.nodemanager.log-dirs property (see also below).
</description>
<name>yarn.nodemanager.delete.debug-delay-sec</name>
<value>600</value>
</property>
```

# COMO É REALIZADA A CONFIGURAÇÃO DO YARN

- Propriedades do ResourceManager:

```
<property>  
<name>yarn.resourcemanager.address</name>  
<value>192.168.0.52:8032</value>  
</property>
```

The address of the applications manager interface in the RM.

```
<property>  
<name>yarn.resourcemanager.scheduler.address</name>  
<value>192.168.0.52:8030</value>  
</property>
```

The address of the scheduler interface

```
<property>  
<name>yarn.resourcemanager.resource-tracker.address</name>  
<value>192.168.0.52:8025</value>  
</property>
```

Required by NodeManagers to connect to Resource Manager

# COMO É REALIZADA A CONFIGURAÇÃO DO YARN

- Propriedades do NodeManager:

1. `yarn.nodemanager.resource.memory-mb` = 90000

2. `yarn.nodemanager.resource.vcores` = 60

1. memory:  $50 * 90\text{GB} = 4500\text{GB} = 4.5\text{TB}$

2. vcores:  $50 * 60 \text{ vcores} = 3000 \text{ vcores}$

# COMO É REALIZADA A CONFIGURAÇÃO DO YARN

- Acessando no Browser:



## All Applications

Cluster

- About
- Nodes
- Applications
  - NEW
  - NEW SAVING
  - SUBMITTED
  - ACCEPTED
  - RUNNING
  - FINISHED
  - FAILED
  - KILLED
- Scheduler

Tools

### Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VContainers Used	VContainers Total	VContainers Reserved	Active Nodes
0	0	0	0	0	0 B	4500 GB	0 B	0	3000	0	1

### User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B

Show 20 entries

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalState
-	-	-	-	-	-	-	-	-

No data available in table

Showing 0 to 0 of 0 entries



# COMO É REALIZADA A CONFIGURAÇÃO DO YARN

- Propriedades do Container:
  - YARN Container Memory Sizing
    - Minimum: `yarn.scheduler.minimum-allocation-mb`
    - Maximum: `yarn.scheduler.maximum-allocation-mb`
  - YARN Container VCore Sizing
    - Minimum: `yarn.scheduler.minimum-allocation-vcores`
    - Maximum: `yarn.scheduler.maximum-allocation-vcores`
  - YARN Container Allocation Size Increments
    - Memory Increment: `yarn.scheduler.increment-allocation-mb`
    - VCore Increment: `yarn.scheduler.increment-allocation-vcores`



### 3. AULA PRÁTICA

## TAREFAS 1

1. Crie um container com um determinado nome usando a imagem **sequenceiq/hadoop-docker**;
2. Configure a taxa de replicação dos dados do HDFS para 3;
3. Adicione um arquivo de texto extraído de sua máquina local para o container em execução.

## TAREFAS 2

1. Insira o arquivo de texto no HDFS;

```
-rw-r--r--  3 root supergroup      15429 2019-05-23 10:50 /LICENSE
drwxr-xr-x  - root supergroup         0 2015-07-22 11:17 /user
```

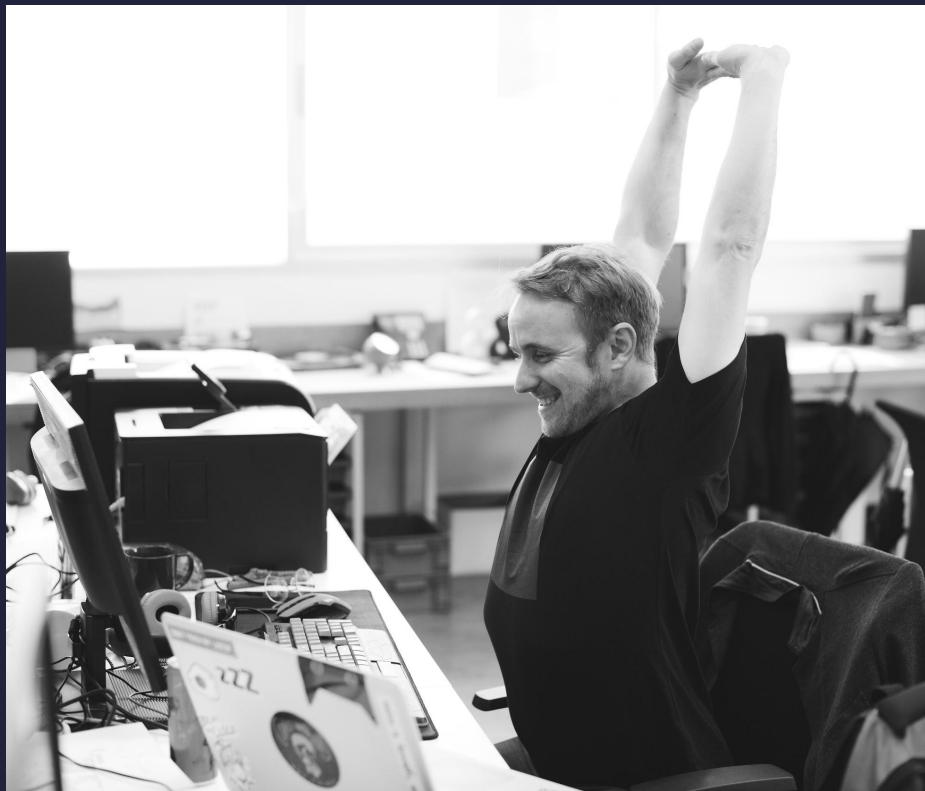
2. Visualize seu arquivo no HDFS;
3. Execute um aplicação wordcount e apresente o resultado de saída.

# 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

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