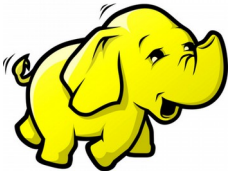


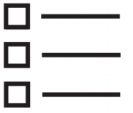
# Cloudera Administrator Apache Hadoop

## Parte 02-1 Clientes Hadoop



Marco Reis  
<http://marcoreis.net>

# Agenda



- Clientes Hadoop
  - Máquina não gerenciada
  - Máquina gateway (EdgeNode)
  - Hue

# Cientes Hadoop

- O Hadoop é operado por meio dos clientes (máquina não gerenciada, máquina de gateway e Hue)
- Um cliente é usado para que o usuário não acesse diretamente um HeadNode
- É uma medida de segurança e performance:
  - Evita acessos indevidos ao HeadNode que possam danificar o cluster
  - Evita que uma operação de I/O degrade o HeadNode, por exemplo, uma cópia de arquivos que exige muito do sistema operacional e rede

# Máquina não gerenciada



- Um computador que não faz parte do cluster pode ser usado para operações no CDH
- A Cloudera disponibiliza um pacote com os programas do Hadoop (hadoop-client)
- A configuração é feita manualmente e as opções são bastante limitadas
  - Suporte apenas para operações do HDFS e YARN
  - Quando a configuração do cluster é alterada, devem ser atualizadas manualmente nos clientes
- Script para instalação:
  - `$ sudo su -`
  - `$ wget 'https://archive.cloudera.com/cdh5/ubuntu/xenial/amd64/cdh/cloudera.list' \`
  - `-O /etc/apt/sources.list.d/cloudera.list`
  - `$ apt-get update`
  - `$ apt-get install hadoop-client hive impala-shell`
- Após a instalação são criados novos usuários, como hdfs, mapred, yarn e zookeeper

# Configuração do cliente HDFS

- Arquivos de configuração:
  - Download Client Configuration (hdfs-clientconfig.zip)
- Descompactar os arquivos e copiar para o computador cliente no diretório /etc/hadoop/conf
  - O computador cliente é o hadoopclient
  - Antes, faça uma cópia do diretório /etc/hadoop/conf, só por precaução
  - Exemplo: `scp * root@hadoopclient:/etc/hadoop/conf/`
- Certificar que os hosts (/etc/hosts) estão configurados no cliente (hadoopclient) e no cluster

The screenshot displays the HDFS (Cluster 1) management interface. On the left, there is a sidebar with sections: Status, Instances, and Configuration. The main area shows the HDFS Summary, including Configured Capacity, Health Tests (with a 'Show 7 Good' link), and a Status Summary table. The Status Summary table lists components and their counts: Balancer (1), DataNode (3), HttpFS (1), NameNode (1), SecondaryNameNode (1), and Hosts (5). On the right, an 'Actions' dropdown menu is open, listing various operations: Start, Stop, Restart, Rolling Restart, Add Role Instances, Rename, Enter Maintenance Mode, Rebalance, Deploy Client Configuration, Upgrade HDFS Metadata, Finalize Rolling Upgrade, Download Client Configuration, Enable High Availability, Roll Edits, and Set up HDFS Data At Rest Encryption.

Component	Count
Balancer	1
DataNode	3
HttpFS	1
NameNode	1
SecondaryNameNode	1
Hosts	5

# Configuração do cliente YARN

- Arquivos de configuração:
  - Download Client Configuration (yarn-clientconfig.zip)
- Copiar os arquivos mapred-site.xml e yarn-site.xml para o hadoopclient com os comandos:
  - `$ scp mapred-site.xml root@hadoopclient:/etc/hadoop/conf/`
  - `$ scp yarn-site.xml root@hadoopclient:/etc/hadoop/conf/`

The screenshot displays the YARN (MR2 Included) cluster management interface for Cluster 1. The 'Actions' dropdown menu is open, showing various operations. The 'Download Client Configuration' option is highlighted.

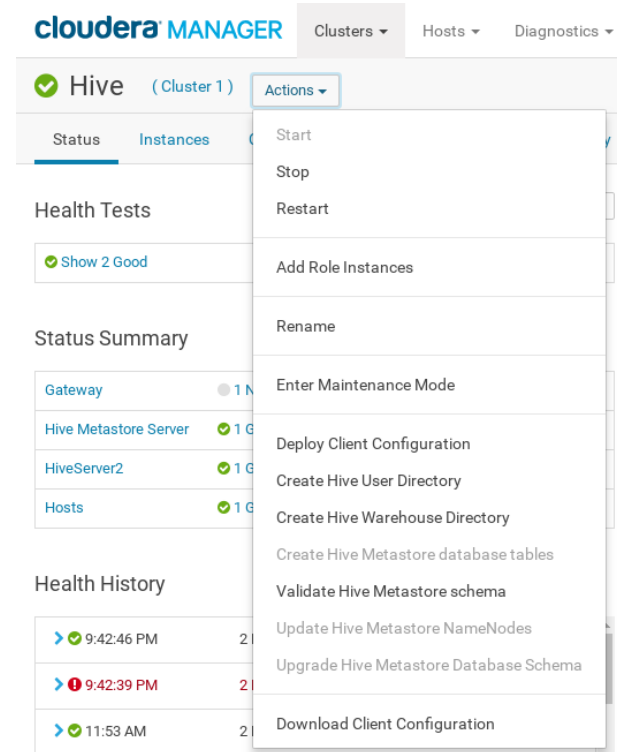
Status	Instances	Configuration	Command
<b>Health Tests</b>			
Show 3 Good			
YARN Container Usage Aggregation This health test is disabled because container usage metric c			
<b>Status Summary</b>			
JobHistory Server	1 Good Health		
NodeManager	3 Good Health		
ResourceManager	1 Good Health		
Hosts	5 Good Health		
<b>Health History</b>			
Aug 15 2:05:25 PM	ResourceManager Health Con		
Aug 15 2:04:35 PM	JobHistory Server Health Con		
Aug 15 1:34 PM	JobHistory Server Health Goo		

**Actions**

- Start
- Stop
- Restart
- Rolling Restart
- Add Role Instances
- Rename
- Enter Maintenance Mode
- Deploy Client Configuration
- Create Job History Dir
- Create NodeManager Remote Application Log Directory
- Import MapReduce Configuration
- Format StateStore
- Create CM Container Usage Metrics Dir
- Download Client Configuration
- Enable High Availability

# Configuração do cliente Hive

- Arquivos de configuração
  - Download Client Configuration (hive-clientconfig.zip)
- Copiar os arquivos hive-site.xml e hive-env.sh para o hadoopclient:
  - `$ scp hive-site.xml root@hadoopclient:/etc/hive/conf`
  - `$ scp Downloads/hive/hive-conf/hive-site.xml root@hadoopclient:/etc/hive/conf`



# Conexão com o HDFS

- Listar os arquivos do cluster

- `sudo -u hdfs hdfs dfs -ls /`

```
Found 2 items
drwxrwxrwt - hdfs supergroup 0 2018-08-14 16:40 /tmp
drwxr-xr-x - hdfs supergroup 0 2018-08-14 16:41 /user
```

- Listar os datanodes disponíveis

- `sudo -u hdfs hdfs dfsadmin -printTopology`

```
Rack: /default
192.168.25.191:50010 (datanode01.lab)
192.168.25.192:50010 (datanode02.lab)
192.168.25.193:50010 (datanode03.lab)
```



# Conexão com o YARN

- Comando no hadoopclient para listar os nós disponíveis
  - `sudo -u yarn yarn node -list`

```
Total Nodes:3
Node-Id      Node-State Node-Http-Address  Number-of-Running-Containers
datanode01.lab:8041  RUNNING datanode01.lab:8042  0
datanode02.lab:8041  RUNNING datanode02.lab:8042  0
datanode03.lab:8041  RUNNING datanode03.lab:8042  0
```

# Conexão com o Hive

- Para conectar no Hive usamos o programa beeline
- Exemplo:
  - \$ beeline -n dataengineer -p dataengineer -u jdbc:hive2://edgenode.lab:10000
- Para mostrar as bases disponíveis:
  - > show databases;

```
dataengineer@hadoopclient:~$ beeline -n dataengineer -p dataengineer -u jdbc:hive2://edgenode.lab:10000
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0
scan complete in 4ms
Connecting to jdbc:hive2://edgenode.lab:10000
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Connected to: Apache Hive (version 1.1.0-cdh5.15.0)
Driver: Hive JDBC (version 1.1.0-cdh5.15.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.1.0-cdh5.15.1 by Apache Hive
0: jdbc:hive2://edgenode.lab:10000> show databases
+-----+
| database_name |
+-----+
| default       |
+-----+
1 row selected (0.34 seconds)
0: jdbc:hive2://edgenode.lab:10000>
```

# Conexão com o Impala

- Para conectar no Impala usamos o programa impala-shell
- Exemplo:
  - `impala-shell -i datanode01.lab`
- Para mostrar as bases disponíveis:
  - `> show databases;`

```
dataengineer@hadoopclient:~$ impala-shell -i datanode01.lab
Starting Impala Shell without Kerberos authentication
Connected to datanode01.lab:21000
Server version: impalad version 2.12.0-cdh5.15.0 RELEASE (build 23f574543323301846b41fa5433690df32efe085)
*****
Welcome to the Impala shell.
(Impala Shell v2.12.0-cdh5.15.1 (64f4e19) built on Thu Aug  9 09:21:02 PDT 2018)

To see a summary of a query's progress that updates in real-time, run 'set
LIVE_PROGRESS=1;'.
*****
[datanode01.lab:21000] > show databases;
Query: show databases
+-----+-----+
| name      | comment                                     |
+-----+-----+
| impala_builtins | System database for Impala builtin functions |
| dbdataengineer | Base de dados de estudos                     |
| default     | Default Hive database                       |
+-----+-----+
Fetched 3 row(s) in 0.01s
[datanode01.lab:21000] >
```

# API de serviços

- O CDH tem uma API de serviços que permite verificar a configuração do servidor por meio de web services
- Listar os clusters disponíveis
  - <http://headnode.lab:7180/api/v18/clusters/>
- Listar os dados do Cluster 1
  - <http://headnode.lab:7180/api/v18/clusters/Cluster%201>
- Listar os serviços do Cluster 1
  - <http://headnode.lab:7180/api/v19/clusters/Cluster%201/services>
- Na imagem vemos o trecho do serviço HDFS

```
{
  "name": "hdfs",
  "type": "HDFS",
  "clusterRef": {
    "clusterName": "cluster"
  },
  "serviceUrl": "http://headnode.lab:7180/cmfserviceRedirect/hdfs",
  "roleInstancesUrl": "http://headnode.lab:7180/cmfserviceRedirect/hdfs/instances",
  "serviceState": "STARTED",
  "healthSummary": "CONCERNING",
  "healthChecks": [
    {
      "name": "HDFS_BLOCKS_WITH_CORRUPT_REPLICAS",
      "summary": "GOOD",
      "suppressed": false
    },
    {
      "name": "HDFS_CANARY_HEALTH",
      "summary": "GOOD",
      "suppressed": false
    }
  ]
}
```

# API de serviços (HDFS)

- Para listar as funcionalidades do HDFS
  - <http://headnode.lab:7180/api/v19/clusters/Cluster%201/services/hdfs/roles>
- Selecione a propriedade name

```
"name": "hdfs-NAMENODE-ab8735db1bc6548069202e72e8a31842",  
"type": "NAMENODE",  
"serviceRef": {  
  "clusterName": "cluster",  
  "serviceName": "hdfs"  
},  
"hostRef": {  
  "hostId": "f10fb064-3af4-435d-9aab-6a5839be83d2"  
},  
"roleUrl": "http://headnode.lab:7180/cm/roleRedirect/hdfs-NAMENODE-ab8735db1bc6548069202e72e8a31842",  
"roleState": "STARTED",  
"healthSummary": "CONCERNING",
```

# API de serviços (HDFS)

- Visualizar as configurações do namenode
  - <http://headnode.lab:7180/api/v19/clusters/Cluster%201/services/hdfs/roles/hdfs-NAMENODE-ab8735db1bc6548069202e72e8a31842/process/>

```
{  
  "configFiles": [  
    "core-site.xml",  
    "dfs_all_hosts.txt",  
    "cloudera_manager_agent_fencer.py",  
    "cloudera-stack-monitor.properties",  
    "navigator.client.properties",  
    "cloudera-monitor.properties",  
    "hdfs-site.xml",  
    "log4j.properties",  
    "redaction-rules.json",  
    "ssl-server.xml",  
    "cloudera_manager_agent_fencer_secret_key.txt",  
    "topology.py",  
    "hdfs.keytab",  
    "http-auth-signature-secret",  
    "event-filter-rules.json",  
    "topology.map",  
    "hadoop-policy.xml",  
    "ssl-client.xml",  
    "hadoop-metrics2.properties"  
  ]  
}
```

# API de serviços (HDFS)

- URL Para baixar um dos arquivos, por exemplo o core-site.xml:
  - <http://headnode.lab:7180/api/v19/clusters/Cluster%201/services/hdfs/roles/hdfs-NAMENODE-ab8735db1bc6548069202e72e8a31842/process/configFiles/core-site.xml>

# Máquina de gateway

- É gerenciado pelo cluster
- Garante a atualização das configurações do CDH automaticamente, não precisa da configuração manual
- Pode ser usado de diversas formas:
  - Cliente para todos os serviços do CDH, como HDFS, YARN, HBase, Hive etc.
  - Área de staging (temporária) para os arquivos que serão copiados no HDFS



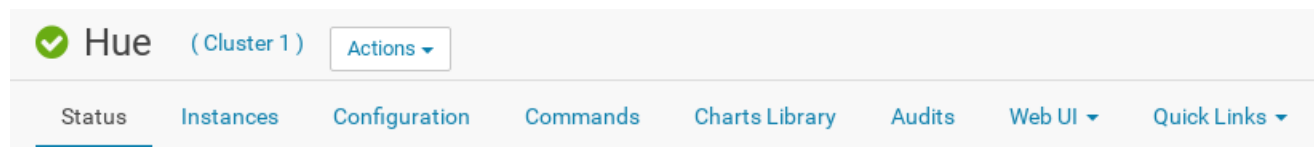
# Acesso ao gateway

- Acesso normal via ssh
  - `ssh root@edgenode`
- Todos os serviços do CDH já estão instalados no gateway
  - Ele é um servidor gerenciado pelo CDH
- Comandos para testar
  - `sudo -u hdfs hdfs dfs -ls /`
  - `sudo -u hdfs hdfs dfsadmin -printTopology`

# Hue

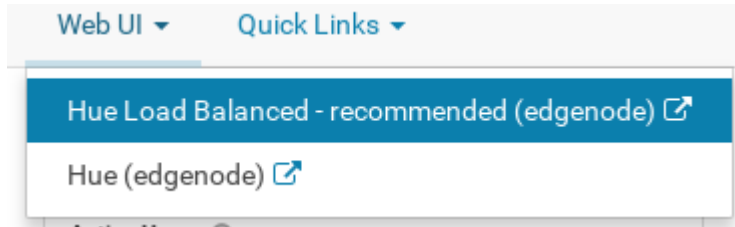


- Originalmente, a operação do Hadoop era descentralizada e complexa, utilizando diversas ferramentas diferentes e sem integração
- O Hue surgiu como uma interface web para operar o Hadoop e realizar análises de dados de forma simples e centralizada
- O foco é principalmente SQL, com suporte a autocompletar
- Acesso por interface gráfica para os serviços do Hadoop:
  - Pig
  - HDFS
  - Hive
  - Impala
  - Oozie



# Acesso pela Web UI

- Recomendado que o acesso seja feito por um EdgeNode para evitar carga excessiva no HeadNode



# Acesso ao Hue

- No primeiro acesso será definido o usuário e a senha do administrador
  - Sugestão para usuário e senha: hive/hive

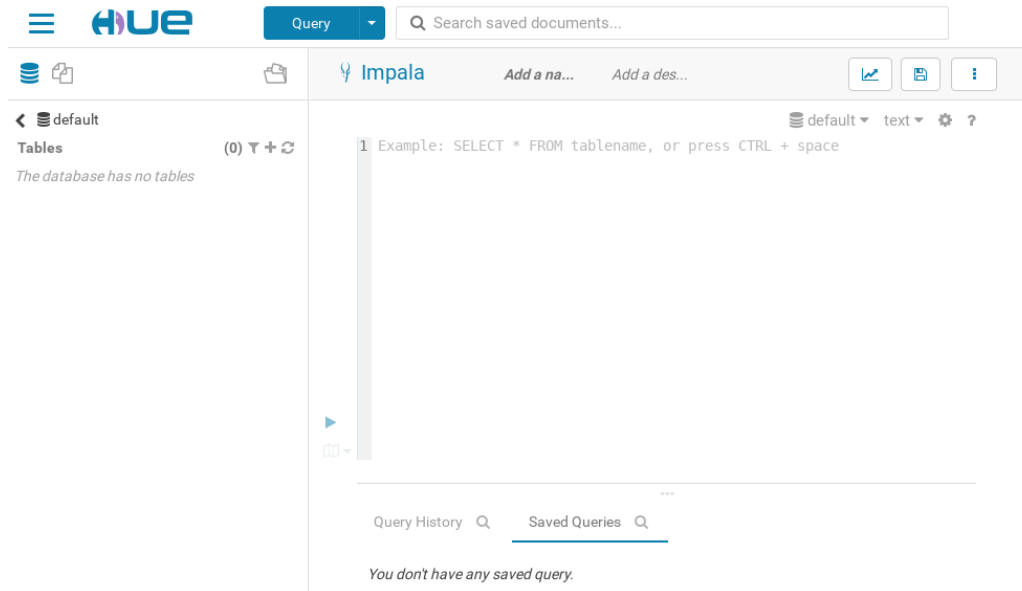


Query. Explore. Repeat.

Sign In

# Tela inicial do Hue

- A tela inicial de consulta do Hue permite a operação dos diversos serviços habilitados
- O Hue é principalmente utilizado para consultas Impala e Hive

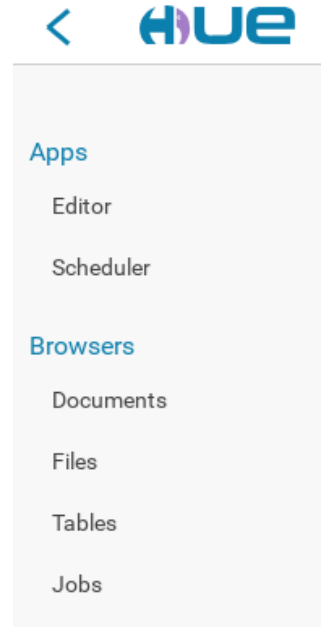


# Tipos de queries

- Menu Query → Editor
  - Impala
  - Hive
  - Pig
  - Java
  - Spark
  - MapReduce
  - Shell
  - Sqoop 1
  - Distcp

# Menu Lateral

- Aplicações
  - Editor: editores de consulta
  - Scheduler: workflows do Oozie
- Browser
  - Documents: scripts gravados
  - Files: navegação no HDFS
  - Tables: administração das tabelas
  - Jobs: execuções do MapReduce, consultas e workflows



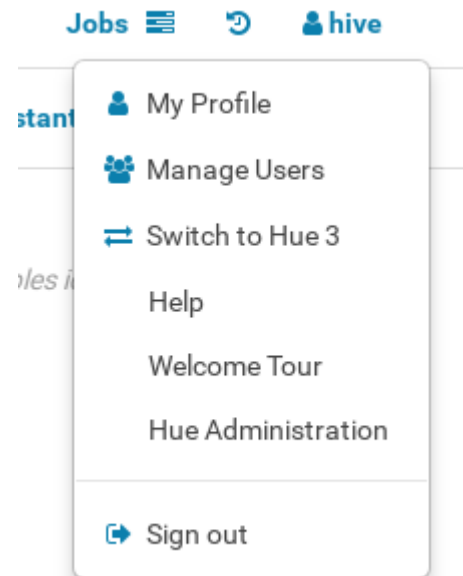
# Gerenciar usuários

- Para listar os usuários vá no menu do usuário (hive) → Manage Users
- Para adicionar um novo usuário clique no botão Add user

Hue Users

 Delete

 Add user 





# Credenciais

- Nos exemplos o usuário será dataengineer
- Cada aluno deve criar um usuário diferente
  - Sugestão: nome e sobrenome, como em marcoreis
- No Step 3: Advanced
  - Checar a opção Superuser status
  - Permite a administração do ambiente
- Clique no botão Add user
- Faça o Sign out do Hue e acesse novamente com o novo usuário

## Hue Users - Create user

Step 1: Credentials (required)   Step 2: Profile and Groups   Step 3: Advanced

Username ⓘ dataengineer

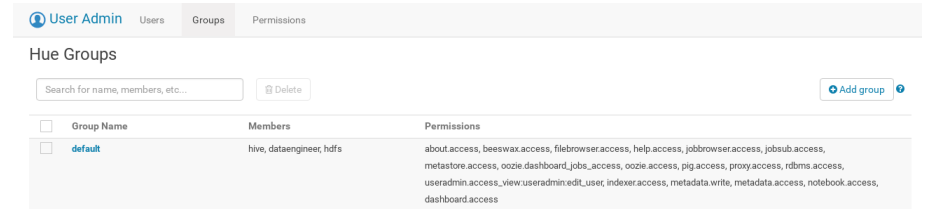
New Password

Password confirmation

Create home directory ⓘ ☒

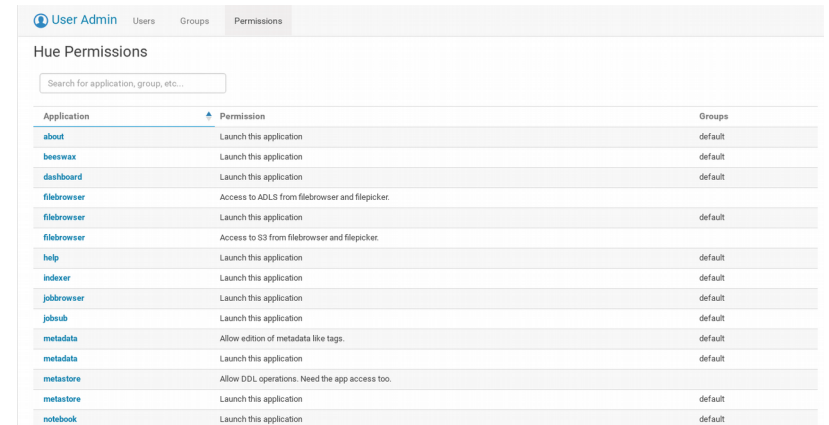
# Grupos e Permissões

- O Hue permite a criação de grupos e a definição de permissões para acesso aos recursos da ferramenta
- As permissões são autoexplicativas e indicam o nome da aplicação e o tipo de ação permitida



The screenshot shows the 'Hue Groups' management page. At the top, there are tabs for 'User Admin', 'Users', 'Groups', and 'Permissions'. Below the tabs, there is a search bar labeled 'Search for name, members, etc...' and a 'Delete' button. On the right, there is an 'Add group' button. The main content area is a table with three columns: 'Group Name', 'Members', and 'Permissions'. There is one row for the 'default' group.

Group Name	Members	Permissions
default	hive, dataengineer, hdfs	about.access, beeswax.access, filebrowser.access, help.access, jobbrowser.access, jobsub.access, metastore.access, oozie.dashboard.jobs.access, oozie.access, pig.access, proxy.access, rdms.access, useradmin.access_view.useradmineditUser, indexer.access, metadata.write, metadata.access, notebook.access, dashboard.access



The screenshot shows the 'Hue Permissions' management page. At the top, there are tabs for 'User Admin', 'Users', 'Groups', and 'Permissions'. Below the tabs, there is a search bar labeled 'Search for application, group, etc...'. The main content area is a table with three columns: 'Application', 'Permission', and 'Groups'. There are 16 rows of permissions.

Application	Permission	Groups
about	Launch this application	default
beeswax	Launch this application	default
dashboard	Launch this application	default
filebrowser	Access to ADLS from filebrowser and flepicker.	
filebrowser	Launch this application	default
filebrowser	Access to S3 from filebrowser and flepicker.	
help	Launch this application	default
indexer	Launch this application	default
jobbrowser	Launch this application	default
jobsub	Launch this application	default
metadata	Allow edition of metadata like tags.	default
metadata	Launch this application	default
metastore	Allow DDL operations. Need the app access too.	
metastore	Launch this application	default
notebook	Launch this application	default






# Inserir dados de exemplo

- Usando o Hue com o usuário individual
  - Copiar um arquivo CSV para o HDFS
  - Criar uma base de dados
  - Criar uma tabela
  - Carregar dados do CSV na tabela


# Copiar para o HDFS

- Menu lateral → Files
  - Mostra o diretório raiz do usuário
- Clique no botão Upload e selecione o arquivo Municipio.csv

[Home](#) / [user](#) / dataengineer


<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	 <a href="#">↑</a>		hdfs	supergroup	drwxr-xr-x	August 24, 2018 11:01 AM
<input type="checkbox"/>	 <a href="#">.</a>		dataengineer	dataengineer	drwxr-xr-x	August 24, 2018 11:09 AM
<input type="checkbox"/>	 <a href="#">Municipio.csv</a>	172.4 KB	dataengineer	dataengineer	-rw-r--r--	August 24, 2018 11:09 AM


Show  of 1 items


Page  of 1 


# Visualizar arquivo


- Clique no arquivo Municipio.csv


 File Browser

 View as binary

 Edit file

 Download

 View file location

 Refresh

Last modified  
08/24/2018  
6:09 PM


User  
dataengineer

Group  
dataengineer

Size  
172.4 KB

Mode  
100644

[Home](#)

Page  to  of 44 



[/ user / dataengineer / Municipio.csv](#)


CodMunicio	UF	NomeMunicípio	PopulacaoResidente
4200051	SC	Abdon Batista	2653
1700251	TO	Abreulândia	2391
3100401	MG	Acaiaca	3920
4300034	RS	Aceguá	4394
5200159	GO	Adelândia	2477
3500204	SP	Adolfo	3557
4200309	SC	Agronômica	4904
3100708	MG	Água Comprida	2025
5200209	GO	Água Limpa	2013
2400406	RN	Água Nova	2900
4300059	RS	Água Santa	3722
3100807	MG	Aguanil	4054
3500600	SP	Águas de São Pedro	2707
4200556	SC	Águas Frias	2424
2900603	BA	Aiquara	4602
3101300	MG	Alagoa	2709
3500758	SP	Alambari	4884
3101409	MG	Albertina	2913
5000252	MS	Alcinópolis	4569

# Nova base de dados

- Menu lateral → Tables
- Clicar em Databases
- Clicar no botão +

Databases

 Drop

<input type="checkbox"/>	Database Name
<input type="checkbox"/>	default

# Nova base de dados

- Database
  - Name: datalake
  - Cada usuário terá seu próprio ambiente, assim, a sugestão é usar o identificador datalake + nome do usuário. Ex: datalakemarcoreis

## DESTINATION

Name

datalake

## PROPERTIES

Description

Description


☒ Default location

# Nova tabela (municipio)

- Navegar até a nova base de dados
- Clicar no botão +
- Clicar na caixa de texto Path para selecionar o arquivo dos dados
- Selecione o Municipio.csv
- O Hue tenta identificar os separadores e mostra um preview dos dados do arquivo

## SOURCE

Type File

Path /user/dataengineer/Municipio.csv .. 

## FORMAT

Field Separator ^Tab ( ) Record Separator New line Quote Character Double Quote

☒ Has Header

## PREVIEW

CodMunicio	UF	NomeMunicpio	PopulacaoResidente
4200051	SC	Abdon Batista	2653
1700251	TO	Abreulândia	2391
3100401	MG	Acaiaca	3920
4300034	RS	Aceguá	4394
5200159	GO	Adelândia	2477



# Create table

- O Hue tenta identificar os tipos de dados dos campos com base em seu conteúdo
- Não é muito confiável...
- Desmascarar a opção Store in Default location

## DESTINATION

Name

## PROPERTIES

Format





☐ Store in Default location

External location

Extras 

Partitions [+ Add partition](#)

## FIELDS

Name	<input type="text" value="CodMunicio"/>	Type	<input type="text" value="bigint"/>		4200051	1700251
Name	<input type="text" value="UF"/>	Type	<input type="text" value="string"/>		SC	TO
Name	<input type="text" value="NomeMunicípio"/>	Type	<input type="text" value="string"/>		Abdon Batista	Abreulândia
Name	<input type="text" value="PopulacaoResidente"/>	Type	<input type="text" value="string"/>		2653	2391

# Tabela criada

- Verifique as estatísticas e informações da tabela criada
- Para consultar os dados clique no botão ►

Databases > datalake > municipio ► ↺ ⬆ ×

No description available

---

Overview Columns (4) Sample Details

**PROPERTIES**

Table  
dataengineer  
Sun Aug 26 09:15:18 BRT 2018  
text External

**STATS**

Location  
08/26/2018 12:15 PM  
0 files  
-1 rows  
0 B

**COLUMNS (4)**

	Name	Type	Description
1	<a href="#">codmunicipio</a>	bigint	Add a description...
2	<a href="#">uf</a>	string	Add a description...
3	<a href="#">nomemunicipio</a>	string	Add a description...
4	<a href="#">populacaoresidente</a>	string	Add a description...

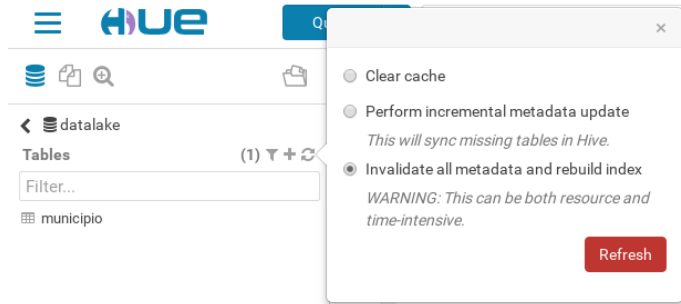
**SAMPLE**

	municipio.codmunicipio	municipio.uf	municipio.nomemunicipio	municipio.populacaoresidente
1	4200051	SC	Abdon Batista	2653
2	1700251	TO	Abreulândia	2391
3	3100401	MG	Acaiaca	3920

[View more...](#)

# Consultando no Impala

- A consulta pode ser executada no Impala ou Hive
- Tabelas criadas no Hive podem não aparecer no Impala. Neste caso, atualize os metadados
  - Invalidate all metadata



The screenshot shows the Impala query interface. At the top, there's a header with the Impala logo and options to 'Add a name...' and 'Add a description...'. Below the header, a query is entered in a text area: `1 SELECT * FROM `datalake`.`municipio` LIMIT 100;`. The query is executed, and the results are displayed in a table below. The table has four columns: 'codmunicio', 'uf', 'nomemunicípio', and 'populacaoresidente'. The results show 10 rows of data.

	codmunicio	uf	nomemunicípio	populacaoresidente
1	4200051	SC	Abdon Batista	2653
2	1700251	TO	Abreulândia	2391
3	3100401	MG	Acaiaca	3920
4	4300034	RS	Aceguá	4394
5	5200159	GO	Adelândia	2477
6	3500204	SP	Adolfo	3557
7	4200309	SC	Agronômica	4904
8	3100708	MG	Água Comprida	2025
9	5200209	GO	Água Limpa	2013
10	2400406	RN	Água Nova	2980

# Consultando no Hive

- A consulta pelo Hive deve retornar exatamente os mesmos registros
- O editor padrão é o do Impala
- Para alterá-lo, clique na estrela ao lado no nome do Hive:
  - Set default application

The screenshot shows the Hive Impala editor interface. At the top, there's a header with the Hive logo, a refresh button, and fields for 'Add a name...' and 'Add a description...'. Below the header, the SQL query editor contains the following query:

```
1 | SELECT * FROM `datalake`.`municipio` LIMIT 100;
```

Below the query editor, there's a section for 'Query History', 'Saved Queries', and 'Results (100+)'. The 'Results (100+)' section is active, displaying a table with 5 columns: an index, 'municipio.codmunicio', 'municipio.uf', 'municipio.nomemunicio', and 'municipio.populacao'. The table contains 13 rows of data.

	municipio.codmunicio	municipio.uf	municipio.nomemunicio	municipio.populacao
1	4200051	SC	Abdon Batista	2
2	1700251	TO	Abreulândia	2
3	3100401	MG	Acaiaca	3
4	4300034	RS	Aceguá	4
5	5200159	GO	Adelândia	2
6	3500204	SP	Adolfo	3
7	4200309	SC	Agronômica	4
8	3100708	MG	Água Comprida	2
9	5200209	GO	Água Limpa	2
10	2400406	RN	Água Nova	2
11	4300059	RS	Água Santa	3
12	3100807	MG	Aguanil	4
13	3500600	SP	Águas de São Pedro	2

# Dúvidas?

Marco Reis  
<http://marcoreis.net>