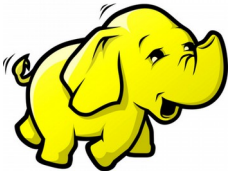


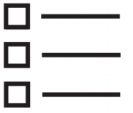
# Cloudera Administrator Apache Hadoop Parte 03-2

## Configuração Avançada e Gerenciamento de Recursos



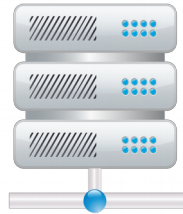
Marco Reis  
<http://marcoreis.net>

# Agenda



- Configuração avançada dos serviços
  - HDFS
  - YARN
  - Hive
  - Spark
  - Impala
- Gerenciamento de recursos

# HDFS



# HDFS - Configuração

- Replicação
- Balancer
- Diretório trash
- Quotas
- HDFS Fuse
- Cache
- Visualização com filtros de status (disponível em todos os serviços)
  - Error
  - Warning
  - Edited
  - Non-default
  - Has Overrides

The screenshot shows the HDFS Configuration page for Cluster 1. The page has a top navigation bar with tabs: Status, Instances, Configuration (selected), Commands, File Browser, Charts Library, Cache Statistics, Audits, NameNode Web UI, and Quick Links. Below the navigation bar is a search bar. The main content area is divided into two columns. The left column contains a 'Filters' section with two expandable categories: 'SCOPE' and 'CATEGORY'. The 'SCOPE' category lists various HDFS services and their counts: HDFS (Service-Wide) (134), Balancer (14), DataNode (79), Gateway (12), HttpFS (53), JournalNode (55), NFS Gateway (50), NameNode (107), SecondaryNameNode (55), and Failover Controller (42). The 'CATEGORY' category lists various configuration categories and their counts: Advanced (91), Checkpointing (2), Cloudera Navigator (4), High Availability (5), Logs (33), Main (38), Monitoring (99), Performance (20), Plugins (2), Ports and Addresses (26), Proxy (22), Replication (5), Resource Management (14), Security (51), and Stacks Collection (5). The right column displays the configuration for several services: ZooKeeper Service (HDFS (Service-Wide) with a radio button selected for 'ZooKeeper' and 'none' as an option), KMS Service (HDFS (Service-Wide) with a radio button selected for 'none'), Object Store Service (HDFS (Service-Wide) with a radio button selected for 'none'), HDFS Block Size (HDFS (Service-Wide) with a text input field containing '128' and a dropdown menu set to 'MiB'), Default Umask (HDFS (Service-Wide) with a text input field containing '022'), Enable WebHDFS (HDFS (Service-Wide) with a checked checkbox), Check HDFS Permissions (HDFS (Service-Wide) with a checked checkbox), and Compression Codecs (HDFS (Service-Wide) with a text input field containing 'org.apache.hadoop.io.compress.DefaultCodec').

# Replicação

- Define a replicação para os novos arquivos gravados no cluster
  - Procure pelo parâmetro dfs.replication
- Altere o fator de replicação para 2, salve e reinicie o serviço
  - A replicação 2 pode ser aplicada em clusters com até 5 datanodes
  - As alterações são propagadas para todos os hosts
- Para alterar a replicação dos arquivos já gravados use o comando
  - `$ sudo -u hdfs hdfs dfs -setrep 2 -r /`

The screenshot displays the HDFS configuration interface. At the top, there are tabs for Status, Instances, Configuration (selected), Commands, File Browser, Charts Library, and Cacti. Below the tabs is a search bar containing 'dfs.replication'. On the left, a 'Filters' section shows a table with columns for 'SCOPE' and values: 'HDFS (Service-Wide)' with value 3, 'Balancer' with value 0, and 'DataNode' with value 0. On the right, a 'General Warning(s)' section shows a warning for 'NameNode and SecondaryNameNode'. Below this, the 'Replication Factor' for 'dfs.replication' is shown as 'HDFS (Service-Wide)' with a value of 3 in a text box.

Filter	Value
HDFS (Service-Wide)	3
Balancer	0
DataNode	0

Warning
NameNode and SecondaryNameNode

Parameter	Value
Replication Factor	3

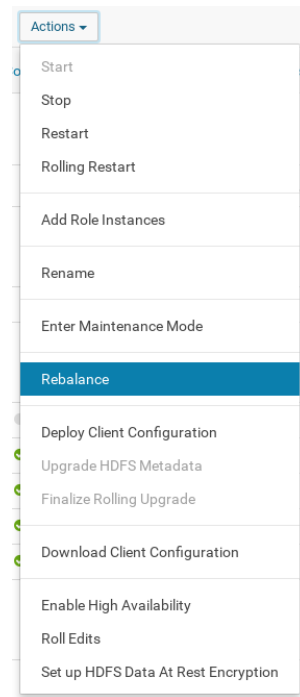
# Balancer

- Durante a operação do cluster é natural haver desbalanceamentos entre as cargas, como na imagem abaixo
- O CDH permite o balanceamento pela interface web (imagem ao lado)
- O HDFS permite pela linha de comando, com o usuário hdfs
  - `$ sudo -u hdfs hdfs balancer`
- A largura de banda para transferência de arquivos do balancer é limitada pelo parâmetro `dfs.balance.bandwidthPerSec`
  - Durante a manutenção dos discos a banda pode ser aumentada para agilizar os procedimentos

Actions for Selected ▾

Columns: 10 Selected ▾

<input type="checkbox"/>	Status	Name	IP	Roles	Commission State	Last Heartbeat	Load Average	Disk Usage	Physical Memory	Swap Space
<input type="checkbox"/>	✓	<a href="#">datanode01.lab</a>	10.4.4.14	5 Role(s)	Commissioned	14.43s ago	0.00 0.00 0.00	28.9 GiB / 48.4 GiB	1.1 GiB / 3.9 GiB	
<input type="checkbox"/>	✓	<a href="#">datanode02.lab</a>	10.4.4.15	5 Role(s)	Commissioned	14.48s ago	0.02 0.02 0.00	28.3 GiB / 38.7 GiB	1.6 GiB / 3.9 GiB	
<input type="checkbox"/>	✓	<a href="#">datanode03.lab</a>	10.4.4.6	6 Role(s)	Commissioned	10.16s ago	0.00 0.00 0.00	7.1 GiB / 38.7 GiB	1.2 GiB / 3.9 GiB	
<input type="checkbox"/>	✓	<a href="#">edgenode.lab</a>	10.4.4.4	16 Role(s)	Commissioned	2.25s ago	0.03 0.23 0.32	6.9 GiB / 19.3 GiB	2.2 GiB / 3.9 GiB	
<input type="checkbox"/>	✓	<a href="#">headnode.lab</a>	10.4.4.5	9 Role(s)	Commissioned	5.21s ago	0.18 0.39 0.34	12.1 GiB / 29 GiB	6.8 GiB / 11.7 GiB	



# Capacidade reservada

- Veja que uma boa quantidade do disco não é utilizada no HDFS porque está reservada para o SO
- Esta reserva é definida no parâmetro `dfs.datanode.du.reserved`
- Altere conforme a imagem, reinicie o serviço e verifique novamente a capacidade do cluster
- Note que essa reserva é importante para o SO se manter operacional, principalmente por causa dos logs
- O valor sugerido pelo CDH é 10 GiB

The screenshot shows the HDFS Configuration page for Cluster 1. The 'Configuration' tab is selected, and the parameter `dfs.datanode.du.reserved` is highlighted in the search bar. Below the search bar, the 'Filters' section shows 'SCOPE'. The 'Reserved Space for Non DFS Use' section displays the parameter `dfs.datanode.du.reserved` with a value of 3 and a unit of GiB. The 'DataNode Default Group' is set to '...and 1 other'.

✓ HDFS (Cluster 1) Actions

Status Instances Configuration Commands File Browser Charts Library Cache Statistics

`dfs.datanode.du.reserved`

Filters

▼ SCOPE

Reserved Space for Non DFS Use

`dfs.datanode.du.reserved`

Edit Individual Values

DataNode Default Group ...and 1 other

3 GiB

# Cache

- O parâmetro `dfs.datanode.max.locked.memory` define a memória usada para cache
- O valor sugerido pela Cloudera é 4 GiB

Maximum Memory Used for DataNode Default Group ...and 1 other ↻  
**Caching**  
dfs.datanode.max.locked.memo  
ry  
[Edit Individual Values](#)

500	MiB ▼
-----	-------



# HDFS Fuse



- O HDFS pode ser usado como um mapeamento no sistema de arquivos do sistema operacional por meio do Fuse
- Instale no host cliente do Hadoop, crie o diretório local para o HDFS e faça o mapeamento
- Exemplo:
  - `$ sudo apt-get install -y hadoop-hdfs-fuse`
  - `$ sudo mkdir /mnt/hdfs`
  - `$ sudo hadoop-fuse-dfs dfs://headnode.lab:8020 /mnt/hdfs`
- Adicione o diretório no `/etc/fstab`
  - `hadoop-fuse-dfs#dfs://headnode.lab:8020 /mnt/hdfs fuse allow_other,usetrash,rw 2 0`

# Maximum File Descriptors

- Você pode alterar o parâmetro para cada um dos serviços
- Na aba Configuration procure pela propriedade rlimit\_fds
- Não existe um valor ideal, devendo ser adaptado para a carga de trabalho e capacidade do cluster
- Caso não preencha, será usado o valor do SO

Maximum Process File  
Descriptors  
[Edit Individual Values](#)

DataNode Default Group [...and 6 others](#)

# HDFS Cache



- Arquivos frequentemente acessados devem ser registrados no cache, uma estratégia que melhora a utilização de memória do NameNode
  - Uma sugestão é aplicar o cache em tabelas do Hive e do Impala
  - O cache está disponível em vários recursos do CDH, inclusive no Spark
- Para listar as opções de cache no HDFS:
  - `$ hdfs cacheadmin`
- O diretório para cache deve ser adicionado a um pool de tamanho adequado, ou seja, o pool deve ser de um tamanho superior ao do diretório
- Exemplo: para criar um pool de 600 MiB:
  - `$ hdfs cacheadmin -addPool dataPool -limit 600000000`
  - `$ hdfs cacheadmin -listPools`
- Exemplo 2: para adicionar um diretório ao pool, com Ttl (time to live) de 1h:
  - `$ hdfs cacheadmin -addDirective -path /user/hive/warehouse/datalake.db/userstackoverflow -pool dataPool -ttl 1h`
  - `$ hdfs cacheadmin -listDirectives -stats`

# Histórico de versões

- As alterações de configuração do cluster podem ser desfeitas a partir do histórico de versões
- Selecione a aba Configuration → History and Rollback
- Verifique a última alteração feita e clique em Details
- Para reverter, clique no botão Revert Configuration Changes
  - Provavelmente será necessário reiniciar o serviço

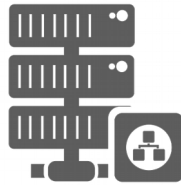
Revision Details		
Message: Service hdfs' config update from API. Sep 6, 2018 5:01:54 PM EDT admin		
Property	Value	Description
<strong>DataNode Group 1 Settings</strong>		
Cgroup CPU Shares	<div><div></div><div>00 -1, 1 +1, 1 00</div><div>1 -1024</div><div>1 +1000</div></div>	Number of CPU shares to assign to this role. The greater the number of shares, the larger the share of the host's CPUs that will be given to this role when the host experiences CPU contention. Must be between 2 and 262144. Defaults to 1024 for processes not managed by Cloudera Manager.
Cgroup I/O Weight		Weight for the read I/O requests issued by this role. The greater the weight, the higher the priority of the requests when the host experiences I/O contention. Must be between 100 and 1000. Defaults to 1000 for processes not managed by Cloudera Manager.
Java Heap Size of DataNode in Bytes	<div><div></div><div>00 -1, 1 +1, 1 00</div><div>-411 M1B</div><div>+1 G1B</div></div>	Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.
Maximum Memory Used for Caching	<div><div></div><div>00 -1, 1 +1, 1 00</div><div>-500 M1B</div><div>+4 G1B</div></div>	The maximum amount of memory a DataNode may use to cache data blocks in memory. Setting it to zero will disable caching.
<strong>dn-group-2-disk Settings</strong>		
Java Heap Size of DataNode in Bytes		Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.
<strong>DataNode Default Group Settings</strong>		
Java Heap Size of DataNode in Bytes	<div><div></div><div>00 -1, 1 +1, 1 00</div><div>-540 M1B</div><div>+1 G1B</div></div>	Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.
<a href="#">Revert Configuration Changes</a> <a href="#">Close</a>		

# Portas

- Os serviços do HDFS (e todos os demais serviços) estão disponíveis em portas HTTP
- Selecione a aba Configuration → Filters → Category → Ports and Address

<b>DataNode Protocol Port</b> dfs.datanode.ipc.address <a href="#">Edit Individual Values</a>	DataNode Default Group ...and 2 others 50020
<b>DataNode Transceiver Port</b> dfs.datanode.address <a href="#">Edit Individual Values</a>	DataNode Default Group ...and 2 others ↗ 1004
<b>DataNode HTTP Web UI Port</b> dfs.datanode.http.address <a href="#">Edit Individual Values</a>	DataNode Default Group ...and 2 others ↗ 1006
<b>Secure DataNode Web UI Port (TLS/SSL)</b> dfs.datanode.https.address <a href="#">Edit Individual Values</a>	DataNode Default Group ...and 2 others 50475
<b>REST Port</b> hdfs.httpfs.http.port	HttpFS Default Group 14000
<b>Administration Port</b> hdfs.httpfs.admin.port	HttpFS Default Group 14001
<b>JournalNode RPC Port</b> dfs.journalnode.rpc.address	JournalNode Default Group 8485
<b>JournalNode HTTP Port</b> dfs.journalnode.http.address	JournalNode Default Group 8480
<b>Secure JournalNode Web UI Port (TLS/SSL)</b> dfs.journalnode.https.address	JournalNode Default Group 8481

# YARN

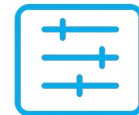


# YARN - Configurações



- O YARN é rico em configurações, especialmente de memória e CPU
  - O Hadoop usa o conceito de vcores no lugar de CPU
- Uma configuração adequada permite uma boa performance no cluster, evitando a ociosidade dos recursos
- As configurações abaixo são recomendações da Cloudera, mas devem ser adaptadas para a carga de trabalho de cada cluster
  - As configurações podem ser feitas a nível de aplicação, ou seja, a aplicação pode enviar as próprias configurações que serão usadas no lugar dos valores definidos no cluster
- A seguir são listas as propriedades e seu valor sugerido:
  - Mínimo de memória para cada container: `yarn.scheduler.minimum-allocation-mb` → 0
  - Máximo de memória para cada container: `yarn.scheduler.maximum-allocation-mb` → memória disponível no host
  - Incremento de memória no container: `yarn.scheduler.increment-allocation-mb` → use um valor ponderado
  - Memória para cada map: `mapreduce.map.memory.mb` → 1 GB
  - Memória para cada reduce: `mapreduce.reduce.memory.mb` → 1 GB
  - Memória do ApplicationMaster: `yarn.app.mapreduce.am.resource.mb` → 1 GB
  - Memória disponível para o YARN no host: `yarn.nodemanager.resource.memory-mb` → total de memória do host menos 1 GB para o SO
  - Número de CPUs disponíveis para o YARN: `yarn.nodemanager.resource.cpu-vcores` → total de processadores do host menos 1 para o SO
  - Número máximo de CPUs para uso no cluster: `yarn.scheduler.maximum-allocation-vcores` → deve ser  $\leq$  `yarn.nodemanager.resource.cpu-vcores`
- Observação: a configuração da memória deve considerar um overhead aproximado de 20% na JVM (Xmx), assim, o Java Heap deve estar entre 75% e 90% da memória
  - É o caso dos parâmetros `mapreduce.map.java.opts` e `mapreduce.reduce.java.opts`

# YARN - Otimização



- Durante a configuração do YARN devem ser observados os valores sugeridos para os outros serviços do cluster
  - Considerando que há outros serviços rodando além do próprio YARN, como Hive, Spark, Impala etc.
- Sugestões:
  - Sistema Operacional – 1 GB a 4 GB / 1 vcore
  - Cloudera Manager agent – 1 GB / 1 vcore
  - Java Heap Size of DataNode – 1 GB / 1 vcore
  - Java Heap Size of NameNode – 1GB / 1 vcore
  - Java Heap Size of NodeManager – 1 GB / 1 vcore
  - Impala daemon – 16 GB por host
- Para otimizações mais específicas, a Cloudera disponibiliza uma planilha que ajuda na configuração no link:
  - <http://tiny.cloudera.com/yarn-tuning-guide>



# Hive





# Hive - Configurações

- A configuração do Hive usa, em grande medida, os parâmetros do HDFS, YARN e Spark (se estiver ativo) para gerenciamento de recursos
- O Spark é executado a partir de executors e vcores, de forma que o mesmo número de vcores deve ser usado em todos os hosts
  - Exemplo VCores: 15 processadores e 5 vcores = 3 executors com 5 vcores cada
- Alguns parâmetros merecem atenção:
  - Spark Driver Maximum Java Heap Size: `spark.driver.memory` → ao menos 1 GB
  - Spark Executor Cores: `spark.executor.cores` → a recomendação é usar por volta de 5 vcores, dependendo do `yarn.nodemanager.resource.cpu-vcores`
    - A fórmula é  $\text{yarn.nodemanager.resource.cpu-vcores} / \text{spark.executor.cores}$  e você deve selecionar o valor que deixe menos vcores ociosos
    - Exemplo: para 15 vcores, selecione `spark.executor.cores=5`. Veja que  $15 / 6 = 2$ , sobrando 3 e  $15 / 4 = 3$ , sobrando 2 vcores, enquanto que  $15 / 5 = 3$  e não sobra nenhum vcore
  - Spark Executor Maximum Java Heap Size: `spark.executor.memory` →  $(\text{memória YARN} / \text{num-executors}) - \text{overhead} = 12 \text{ GB}$
  - `spark.yarn.executor.memoryOverhead` → por volta de 20% da memória em cada executor, ou seja, a equação é  $0.2 * (\text{memória YARN} / \text{num-executors}) = 3 \text{ GB}$

# Impala



# Impala – Configuração

- O Impala não tem configurações de recursos rígidas, apenas valores sugeridos para o Catalog Server (150 MB) e para o StateStore (64 MB)
- Para o Impala Daemon, o limite de memória sugerido é memória do host \* 0.64

# Configuração do Pool de Recursos



# Dynamic Resource Pool Configuration



- É natural que alguns usuários tenham preferência na execução das aplicações
- No CM isso é feito por meio do Dynamic Resource Pool, que é uma política de configuração e agendamento de recursos baseada em usuários
- O agrupamento define os limites de concorrência e prioridade entre aplicações em execução
  - Se não forem usados os recursos ficam disponíveis, assim, um pool pode usar até 100% do cluster
  - O pool é sempre limitado ao Max Resources, ou seja, mesmo com ociosidade, não é possível usar mais do que está definido neste valor
- Para criar novos pools selecione o botão Edit → Create Subpool em root.users. Serão 2 novos pools:
  - root.users.datascientist: 80% Max Resources
  - root.users.dataengineer: 40% Max Resources
- Clique no botão Refresh Dynamic Resource Pools

The screenshot shows the 'Edit Resource Pool' interface for a pool named 'root.users.datascientist'. The 'Resource Limits' tab is active, showing various configuration options:

- Resource Pool Name:** root.users. datascientist
- ☐ Parent Pool
- Resource Limits:** Weight: 1 (Share of resources relative to other pools.)
- Min Resources:** Virtual Cores (The minimum number of CPU and memory available to this pool. This takes priority over the weight based share. (optional))
- Max Resources:** 60 % CPU & Memory [Edit separately](#) (The maximum amount of CPU and memory available to this pool. This takes priority over the weight based share. (optional))
- [Switch to use absolute values](#)
- Max Running Apps:** (A limit on the number of applications simultaneously running in a pool.)
- Max Application Master Share:** (Limit the fraction of the resource pool's fair share that can be used to run ApplicationMasters. For example, if set to 1.0, then ApplicationMasters in the leaf pool can take up to 100% of both the memory and CPU fair share. A value of -1.0 disables monitoring of the ApplicationMaster share. The default value is 0.5.)

# Dynamic Pool em ação

- Rode novamente as aplicações e perceba que os recursos estão limitados ao Max Resources
- O dimensionamento dos recursos é calculado a partir da CPU e memória utilizada pelo algoritmo DRF (Dominant Resource Fairness)

<a href="#">application_1536409474975_0016</a>	datascientist	Total por município	MAPREDUCE	root.users.scientist	Sat Sep 8 10:50:45 -0300 2018	N/A	RUNNING	UNDEFINED	4	4	4096	0	0		<a href="#">ApplicationMaster</a>
<a href="#">application_1536409474975_0015</a>	dataengineer	Total por município	MAPREDUCE	root.users.dataengineer	Sat Sep 8 10:50:44 -0300 2018	N/A	RUNNING	UNDEFINED	2	2	2048	0	0		<a href="#">ApplicationMaster</a>

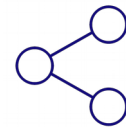
# Seleção do pool na aplicação



- O YARN permite a passagem de parâmetros para personalizar da aplicação, como por exemplo memória, vcores etc.
- Para mudar o pool padrão da aplicação use o parâmetro `mapreduce.job.queue.name`.  
Exemplo:
  - `$ yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi -Dmapreduce.job.queue.name=root.users.datascientist 100 1000`
  - `$ yarn jar analisador-hadoop.jar net.marcoreis.hadoop.mapreduce.parte2.TotalPorMunicipioDriver -Dmapreduce.job.queue.name=root.users.dataengineer /user/dataengineer/dados/bolsafamilia/saida/bolsafamilia/`
- Atenção para a combinação entre usuário do Kerberos e permissão de acesso no HDFS
  - O usuário que submete a aplicação precisa gravar em seu diretório HDFS



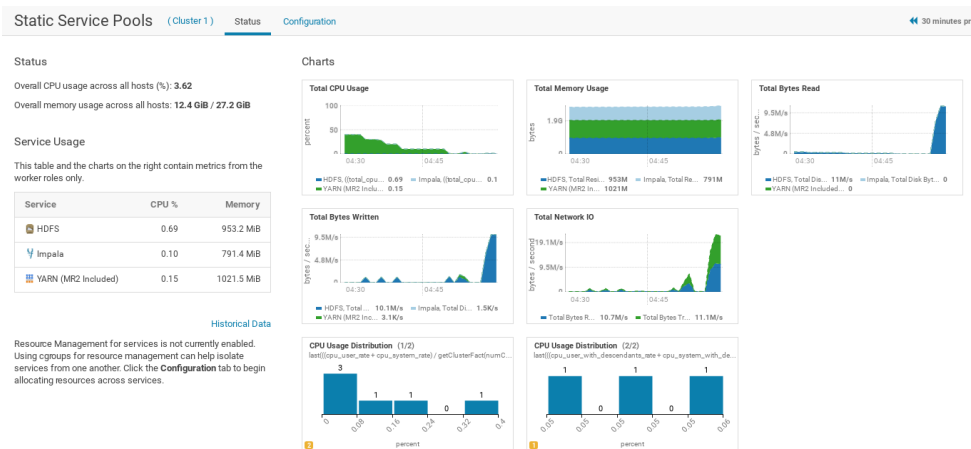
# Pool para produção e desenvolvimento



- Uma sugestão de configuração para um cluster seria usar um pool para produção e outro para desenvolvimento
  - Pode ser uma alternativa a ter de usar 2 clusters separados
  - Um pool para root.production (90%) e outro root.development (20%)

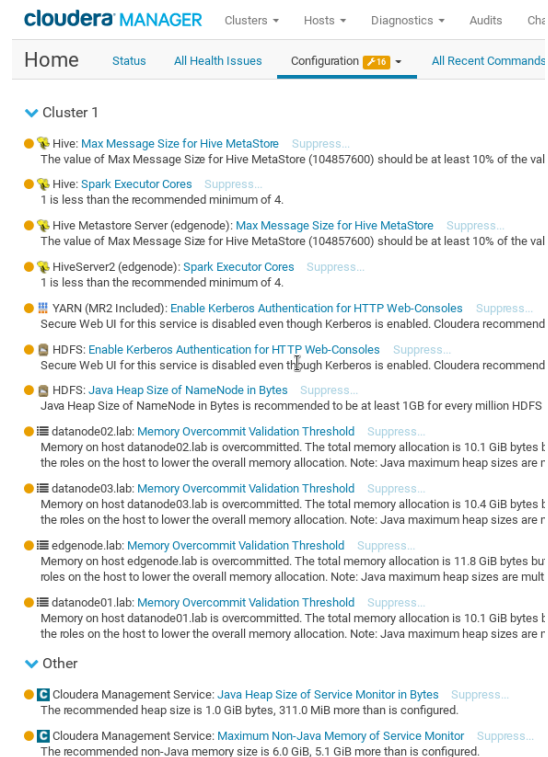
# Static Service Pools

- Permite limitar os recursos do cluster (CPU, memória e IO) para cada serviço (HDFS, YARN e Impala), de forma que a sobrecarga de um serviço não impacta nos demais
- O administrador define um limite percentual para cada serviço e o CM sugere alterações nas configurações dos workernodes
- Essas alterações são estáticas, uma vez que são feitas nos arquivos de configuração



# Configuration Issues

- O Cloudera Manager dispõe de uma facilidade para configuração, na qual são mostrados alertas referentes aos valores que estão fora do padrão
- A funcionalidade está disponível na página inicial do Cloudera Manager → aba Configuration → Configuration Issues
- Os alertas são referentes aos serviços do CDH e de próprio CM
- Para eliminar o alerta de uma configuração, clique na opção Suppress e escreva a justificativa da anulação



# All Health Issues

- Problemas críticos no cluster são listados na aba All Health Issues

cloudera MANAGER Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾

Home Status **All Health Issues 15** Configuration 15 All Recent Commands

Organize By Entity Organize By Health Test

▼ Cloudera Management Services

- Activity Monitor (headnode) Log Files ▾
  - Process Status Suppress...

▼ Cluster 1

- ZooKeeper
  - ZooKeeper Canary Suppress...
- Server (datanode01) Log Files ▾
  - Quorum Membership Suppress...
  - Process Status Suppress...
- Server (edgenode) Log Files ▾
  - Quorum Membership Suppress...
  - Process Status Suppress...
- Server (headnode) Log Files ▾
  - Quorum Membership Suppress...
  - Process Status Suppress...

Dúvidas?

Marco Reis  
<http://marcoreis.net>