

EDH 作为统一的企业级数据中心，往往是一个多租户的应用环境。在该环境中，不同用户会同时使用集群资源。如何保证用户数据不被任意篡改？如何保证任务的权限控制（例如用户 A 不能任性地取消用户 B 的任务）？如何确保用户资源使用不超过他们的配额？

1. 开启 HDFS 权限检查 (默认是开启的)

Category	Property	Value
Service-Wide / Security	Superuser Group dfs.permissions.superusergroup, dfs.permissions.superusergroup	supergroup default value
Service-Wide	Check HDFS Permissions dfs.permissions	<input checked="" type="checkbox"/> default value

“Check HDFS Permissions”中的可选框选中

2. 在集群中创建新用户，以 cloudera-dev 为例

```
# 增加用户组
[root]$ groupadd cloudera-dev
# 增加用户
[root]$ useradd -g cloudera-dev cloudera-dev
# 查看用户 cloudera-dev 所属的所有组
[root]$ groups cloudera-dev

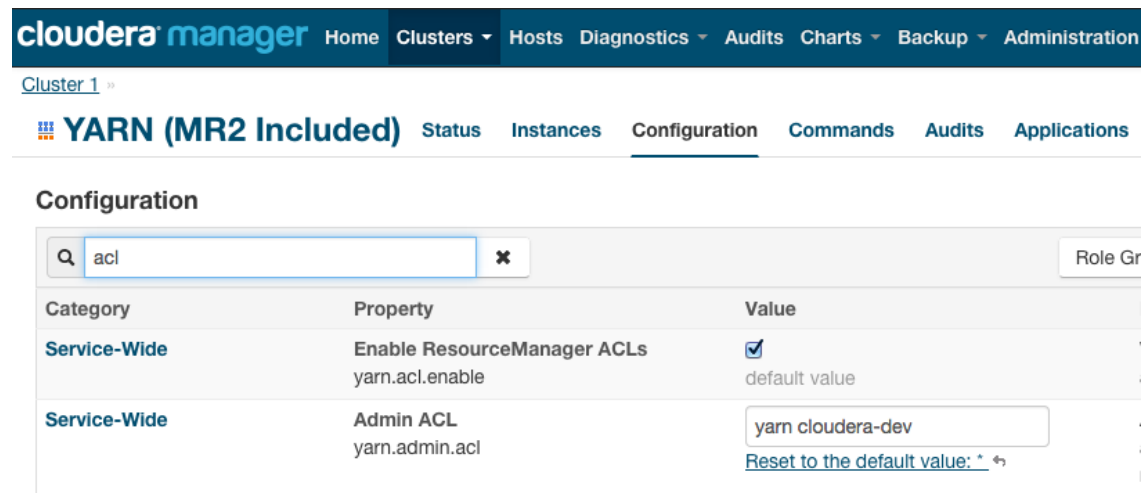
# Hadoop 创建相应的用户
[root]$ sudo -u hdfsadoopfs -mkdir /user/cloudera-dev
[root]$ sudo -u hdfsadoopfs -chown cloudera-dev:cloudera-dev /user/cloudera-dev
```

一般而言，创建新用户会在集群中每台机器上都创建同样的用户。不过理论上，为了运行该案例，只需要在安装资源管理器 (Resource Manager) 的机器上创建相应用户即可。CDH 默认情况下使用 whoami 获取用户的信息、bash -c groups xxx 获取用户与组之间的映射关系，并使用这两份信息进行 ACL 验证。

3. 运行第一个 MapReduce 示例程序“word count”

```
[root]$ sucloudera-dev
[cloudera-dev]$ echo "Hello World Bye World" > file0
[cloudera-dev]$ echo "Hello Hadoop Goodbye Hadoop" > file1
[cloudera-dev]$ hadoopfs -mkdir -p /user/cloudera-dev/wordcount/input
[cloudera-dev]$ hadoop fs -put file* /user/cloudera-dev/wordcount/input
[cloudera-dev]$ hadoop jar /opt/cloudera/parcels/CDH/jars/hadoop-examples.jar wordcountwordcount/input
wordcount/output
[cloudera-dev]$ hadoopfs -getmergewordcount/output output.txt
[cloudera-dev]$ cat output.txt
Bye 1
Hadoop 2
Hello 2
Goodbye 1
World 2
```

4. 开启资源管理器 ACL 并设置相应的管理 ACL (Admin ACL)



Category	Property	Value
Service-Wide	Enable ResourceManager ACLs yarn.acl.enable	<input checked="" type="checkbox"/> default value
Service-Wide	Admin ACL yarn.admin.acl	yarn cloudera-dev Reset to the default value: *

其中 `yarn.acl.enable` 默认值为 `true`。而对于 `yarn.admin.acl` 默认值为 `*`，意味着所有人都可以提交任务、管理已提交（比如取消 `kill`）的任务。格式为“以逗号分隔的用户列表+空格+以逗号分隔的组列表”，例如“`user1,user2 group1,group2`”。如果只有组信息，需要在最前端加入一个空格，例如“`group1,group2`”。另外特别需要注意的是需要将“`yarn`”加入到用户列表中，默认安装 CDH 后，有关 YARN 服务的命令会以 `yarn` 用户的身份进行运行，若 `yarn` 不设置于 `yarn.admin.acl` 中，可能出现权限相关的错误（例如刷新动态资源池）。

在该示例中，`yarn.admin.acl` 列表中包含一个用户 `yarn` 以及一个组 `cloudera-dev`。

5. 关闭未声明资源池的自动生成

cloudera manager Home Clusters Hosts Diagnostics Audits Charts Backup Administration

Cluster 1 »

YARN (MR2 Included) Status Instances Configuration Commands Audits Applications Chart

Configuration

Q undeclare X Role Groups

Category	Property	Value
Service-Wide / Resource Management	Allow Undeclared Pools yarn.scheduler.fair.allow-undeclared-pools	<input checked="" type="checkbox"/> Reset to the default value: true

默认情况下，“Allow Undeclared Pools”可选框是选中的，需要关闭，否则如果用户指定一个尚未声明的资源池，比如 prod，YARN 将会自动生成一个 prod 资源池。配置文件修改后需要重新启动 YARN 服务，重新部署客户端配置。

6. 开启“若用户提交任务不指定特定的 queue，就使用 defaultqueue”

cloudera manager Home Clusters Hosts Diagnostics Audits Charts Backup Administration

Cluster 1 »

YARN (MR2 Included) Status Instances Configuration Commands Audits Applications Cha

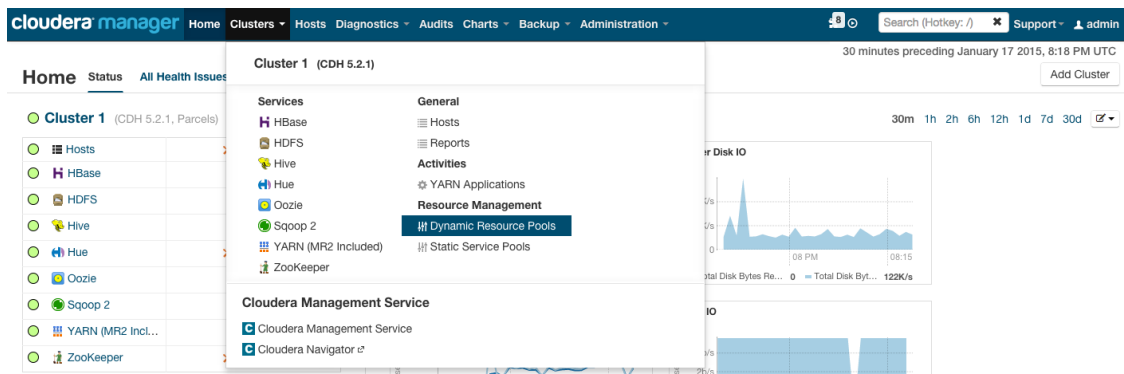
Configuration

Q yarn.scheduler.fair.user-as-default-queue X Role Groups

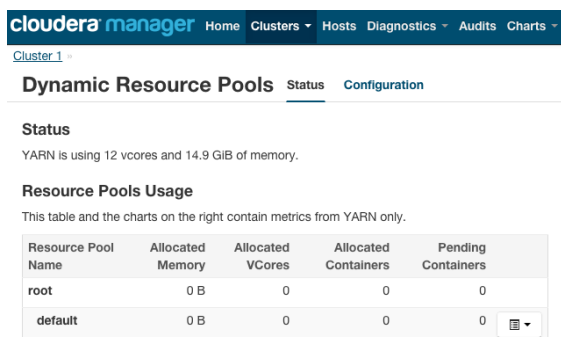
Category	Property	Value
ResourceManager Default Group	Fair Scheduler User As Default Queue yarn.scheduler.fair.user-as-default-queue	<input checked="" type="checkbox"/> Reset to the default value: true

默认情况下，“Fair Scheduler User As Default Queue”可选框是选中的，意味着如果用户提交任务时不指定特定的 queue，就使用以用户命名的 queue。

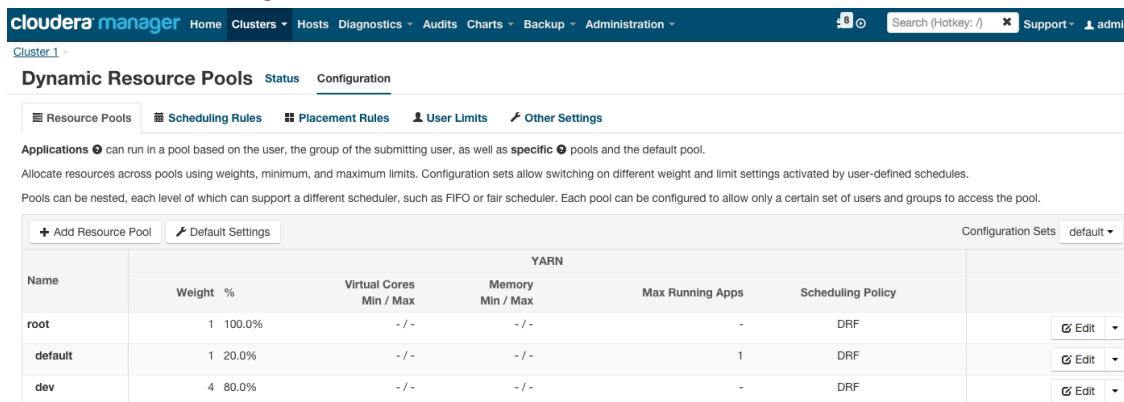
7. 进入动态资源池 (Dynamic Resource Pools) 配置页面



访问动态资源池管理页面



选择配置选项 (Configuration, 右上角)



Resource Pools: 展示了当前资源池的分配情况，默认情况下，只有一个资源池 `root.default`。在该示例中，`dev` 是自定义的资源池。权重 (**weight**) 定义了两个资源池资源分配的比例。示例中，`dev` 设置权重为 4 而 `default` 的权重为 1，那么集群资源的 80% 会被分配给 `dev`。注意，这里提到的资源分配不是一个静态的概念，例如当前资源池 `dev` 中没有任何任务在运行，那么资源池 `default` 是允许使用超过 20% 的集群资源，比如 50%。

8. 为资源池 root 配置资源池 ACL

点击资源池 root 对应的“Edit”按钮

+ Add Resource Pool		Default Settings								Configuration Sets		default ▾
Name	YARN											
	Weight	%	Virtual Cores Min / Max	Memory Min / Max	Max Running Apps	Scheduling Policy						
root	1	100.0%	- / -	- / -	-	DRF						Edit ▾
default	1	20.0%	- / -	- / -	1	DRF						Edit ▾
dev	4	80.0%	- / -	- / -	-	DRF						Edit ▾

在通用 (General) 栏，设置资源池 root 的调度算法，一般使用默认的 DRF

Edit Resource Pool: root

General

YARN

Submission Access Control

Administration Access Control

Scheduling Policy

☒ DRF: Dominant Resource Fairness. Schedules resources fairly based on both CPU and memory. (Recommended)

☐ FAIR: Schedules resources fairly based only on memory.

☐ FIFO: A pool with sub pools cannot be FIFO.

Min Share Preemption Timeout

Seconds

Fairshare Preemption is currently disabled. Enable it here: [Fair Scheduler Preemption](#)

OK

Cancel

在 YARN 栏，设置资源池权重，资源约束等

Edit Resource Pool: root

General

YARN

Submission Access Control

Administration Access Control

Multiple configuration sets allow you to specify different settings based on your schedule.

default

Weight

Share of resources relative to other pools.

Virtual Cores (Min / Max)

/

The minimum and the maximum number of virtual cores available to the pool. These override weight settings. (optional)

Memory (Min / Max)

/

MB

The minimum and the maximum amount of aggregate memory available to the pool. These override weight settings. (optional)

Max Running Apps

A limit on the number of applications simultaneously running in a pool.

OK

Cancel

在提交访问控制 (Submission Access Control) 栏设置哪些用户或组可以向该资源池提交任务

Edit Resource Pool: root

General

YARN

Submission Access Control

Administration Access Control

This feature is relevant only if **Enable ResourceManager ACLs** is set to **true** and **Admin ACL** is NOT set to * (See Other Settings).

Fair Scheduler Access Control Lists control who can submit applications to pools. For subpools, users who have permission to submit a parent pool automatically inherit the same ability for the child.

☐ Allow anyone to submit to this pool

☒ Allow these users and groups to submit to this pool

Users

Comma separated list of users. Space characters are not allowed.

Groups

cloudera-dev

OK

Cancel

在管理访问控制 (Administration Access Control) 栏设置哪些用户或组可以对资源池中的任务进行管理

Edit Resource Pool: root

General

YARN

Submission Access Control

Administration Access Control

This feature is relevant only if **Enable ResourceManager ACLs** is set to **true** and **Admin ACL** is NOT set to * (See Other Settings).

Fair Scheduler Access Control Lists control who can submit applications to pools. For subpools, users who have permission to submit a parent pool automatically inherit the same ability for the child.

☐ Allow anyone to submit to this pool

☒ Allow these users and groups to submit to this pool

Users

yarn

Groups

cloudera-dev

OK

Cancel

9. 为资源池 dev 配置资源池 ACL

可以使用与 root 相同的 ACL 配置，也可以使用不同的配置，该示例假设使用相同的设置。

10. 测试

测试一：用户 root 向资源池 dev 提交 word count 任务

```
[root]$ hadoop jar wordcount-0.9.0.jar com.cloudera.example.WordCount
```

```
-Dmapreduce.job.queueName=devwordcount/input wordcount/output
...
15/01/17 17:29:28 WARN security.UserGroupInformation: PrivilegedActionExceptionas:root (auth:SIMPLE)
cause:java.io.IOException: Failed to run job : User root cannot submit applications to queue root.dev
Exception in thread "main" java.io.IOException: Failed to run job : User root cannot submit applications to queue root.dev
...
```

注意：这里的 **word count** 是自定义的，与 CDH 自带的 **word count** 示例的唯一区别在于，自定义 **word count** 的 **Mapper** 程序在运行时首先使用 `Thread.sleep(300 * 1000)` 休眠 5 分钟。这主要是为了后续对资源池管理的测试。

测试二：用户 **root** 取消用户 **cloudera-dev** 提交的、运行于资源池 **dev** 中的 **word count** 任务

用户 **cloudera-dev** 向资源池 **dev** 提交 **word count** 任务

```
[cloudera-dev]$ hadoop jar wordcount-0.9.0.jar com.cloudera.example.WordCount
-Dmapreduce.job.queueName=devwordcount/input wordcount/output
...
```

用户 **root** 查询相应任务的 **id**，假设获得任务 **id** 为 **job_1421512955131_0006**

```
[root]$ hadoop job -list
...
```

用户 **root** 取消 (**kill**) 该任务

```
[root]$ hadoop job -kill job_1421512955131_0006
...
Exception in thread "main" java.io.IOException: org.apache.hadoop.yarn.exceptions.YarnException:
java.security.AccessControlException: User root cannot perform operation MODIFY_APP on
application_1421512955131_0006
...
```

测试三：用户 **alex** (属于组 **cloudera-dev**) 取消用户 **cloudera-dev** 提交的、运行于资源池 **dev** 中的 **word count** 任务

增加用户 **alex**，设置所属组 **cloudera-dev**

```
[[root]$ useradd -g cloudera-dev alex
...
```

用户 **cloudera-dev** 向资源池 **dev** 提交 **word count** 任务

```
[cloudera-dev]$ hadoop jar wordcount-0.9.0.jar com.cloudera.example.WordCount
-Dmapreduce.job.queueName=devwordcount/input wordcount/output
...
```

用户 **alex** 查询相应任务的 **id**，假设获得任务 **id** 为 **job_1421512955131_0006**

```
[alex]$ hadoop job -list
...
```

用户 **alex** 取消 (kill) 该任务

```
[alex]$ hadoop job -kill job_1421512955131_0006
```

```
...
```

```
15/01/17 17:44:20 INFO impl.YarnClientImpl: Killed application application_1421512955131_0006
```

```
Killed job job_1421512955131_0006
```

```
...
```