

Configuring YARN in Informatica Big Data Management[®]

Abstract

You can use YARN in Informatica Big Data Management® to manage how resources are allocated to jobs that run in the Hadoop environment. You can manage resources using YARN schedulers, YARN queues, and node labels. This article describes how you can define and use the schedulers, queues, and node labels.

Supported Versions

- Big Data Management 10.2.1

Table of Contents

Overview.	2
Hadoop Distribution Support.	3
YARN Schedulers.	3
Fair Scheduler.	3
Capacity Scheduler.	5
YARN Queues.	8
Example. Directing Spark Jobs to a Queue.	9
Node Labeling.	9
Example. Queues and Node Labeling on the Blaze Engine.	10
Configuring Schedulers, Queues, and Node Labels.	11
Part 1. Configure the YARN Scheduler.	12
Part 2. Configure YARN Queues.	13
Part 3. Configure Node Labels.	16

Overview

You can use YARN schedulers, YARN queues, and node labels to manage resources on a Hadoop cluster when you run jobs in the Hadoop environment.

Schedulers, queues, and node labels operate in the following ways:

YARN Schedulers

A YARN scheduler allocates resources to YARN queues and determines how the jobs that are submitted to a queue can use the allocated resources. You can use a fair scheduler or a capacity scheduler depending on your organization's policies on sharing resources.

YARN Queues

A YARN queue is an organizing structures for jobs that run on the cluster. You can redirect Blaze, Spark, and Hive jobs to specific queues. The jobs in a queue can access the resources that are allocated to the queue by a YARN scheduler.

Node Labels

Node labels divide a cluster into partitions. Each partition is a collection of nodes that have specific characteristics. You can run jobs in a partition to allow jobs to utilize a node's particular characteristics.

Hadoop Distribution Support

The following table lists the Hadoop distributions, the YARN schedulers that each distribution recommends, and whether each distribution supports node labeling:

Distribution	Recommended Scheduler	Node Labeling Support
Amazon EMR	Capacity scheduler	Yes
Azure HDInsight	Capacity scheduler	Yes
Cloudera CDH	Fair scheduler	No
Hortonworks HDP	Capacity scheduler	Yes
MapR	Capacity scheduler	Yes

Note: Fair and capacity schedulers are supported on all Hadoop distributions, but different Hadoop distributions recommend a different scheduler.

YARN Schedulers

A YARN scheduler determines how resources are allocated on a Hadoop cluster.

When you configure a scheduler, you define the scheduler and the queues that the scheduler manages. The scheduler allocates cluster resources to the queues. The resources that are allocated to a queue determine the resources that are available to jobs that are submitted to the queue.

By integrating queues within schedules, multiple tenants can share a Hadoop cluster. Depending on your organization's policies, you might use a fair scheduler or a capacity scheduler.

For example, the users in an organization might share a single Hadoop cluster. The organization can allocate an equal number of resources to the jobs that users submit. To allocate resources evenly, the organization can employ a fair scheduler.

On the other hand, an association of organizations might collectively fund a Hadoop cluster. Each organization in the association has different computing needs and invests a certain stake in the cluster. Based on an organization's investment, the organization is guaranteed a certain amount of cluster resources. To divide and allocate cluster resources between the organizations in the association, the association can employ a capacity scheduler.

Fair Scheduler

A fair scheduler allows the users in an organization to share resources evenly on a Hadoop cluster.

The scheduler distributes resources using weights. The weights determine the amount of cluster resources that are available to a queue. The available resources in a queue are shared evenly between jobs in the queue.

When you use a fair scheduler, smaller jobs that require less time can access resources that are being used by larger jobs. As a result, users do not have to wait for larger jobs to finish before smaller jobs can run.

Note: You cannot use a fair scheduler with node labeling.

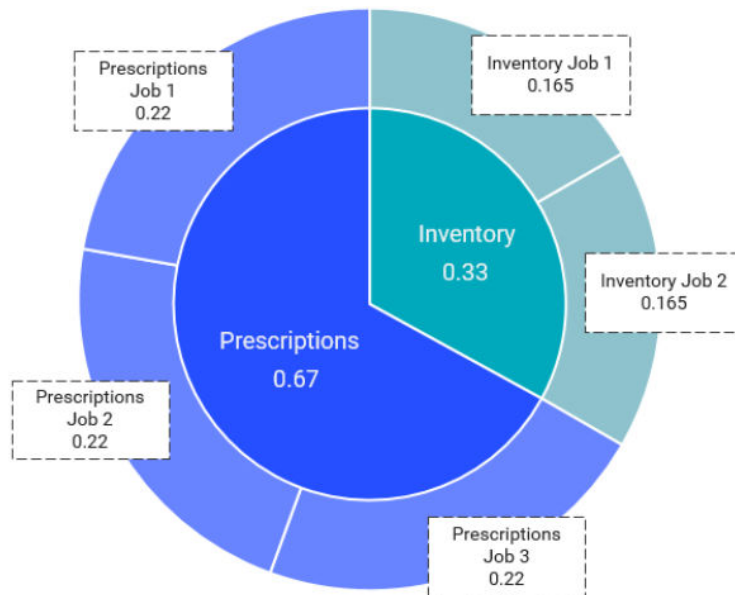
Example. Using a Fair Scheduler

You are a Hadoop administrator for a healthcare company that owns a Hadoop cluster where the company runs different data processes. You want to allocate resources to each of the processes.

The healthcare company might have two data processes that run simultaneously. One process might handle inventories while another process might handle prescriptions. The following table describes how you might divide the data processes into queues:

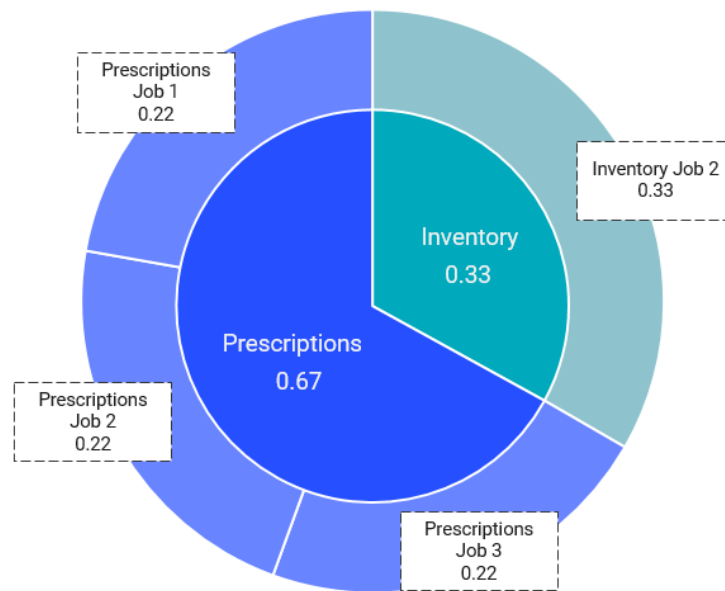
Queue Name	Weight	Description
Inventory	0.33	Queue for jobs that process inventory data from hospital pharmacies.
Prescriptions	0.67	Queue for jobs that process data on the prescriptions that doctors write.

At a point in time, the queue `Inventory` might be running 2 jobs, while the queue `Prescriptions` might be running 3 jobs. The following image shows the proportion of cluster resources that each job can use:



Note that the resources allocated to the queue `Inventory` are divided in half. Each half is allocated to one of the two jobs in the queue. The resources allocated to the queue `Prescriptions` are divided into three parts. Each part is allocated to one of the three jobs in the queue.

The first job in the queue `Inventory` completes. The following image shows the proportion of cluster resources that are available to the remaining jobs:



Capacity Scheduler

A capacity scheduler allows multiple organizations or multiple environments in a single organization to share a large Hadoop cluster.

The scheduler distributes resources using capacities that are allocated to each organization or environment. The capacities determine the percentage of cluster resources that are guaranteed to each organization or environment. The scheduler distributes any excess capacity that is underutilized.

For example, a single organization can use a capacity scheduler to assign resources to test and production environments to guarantee each environment a certain percentage of cluster resources. When the production environment is not fully utilizing its allocated resources, the test environment can use the production environment's excess cluster resources. Similarly, when the test environment is not fully utilizing its allocated resources, the production environment can use the test environment's excess cluster resources. Additionally, the organization does not have to create and maintain different Hadoop clusters for each environment.

Example. Using a Capacity Scheduler

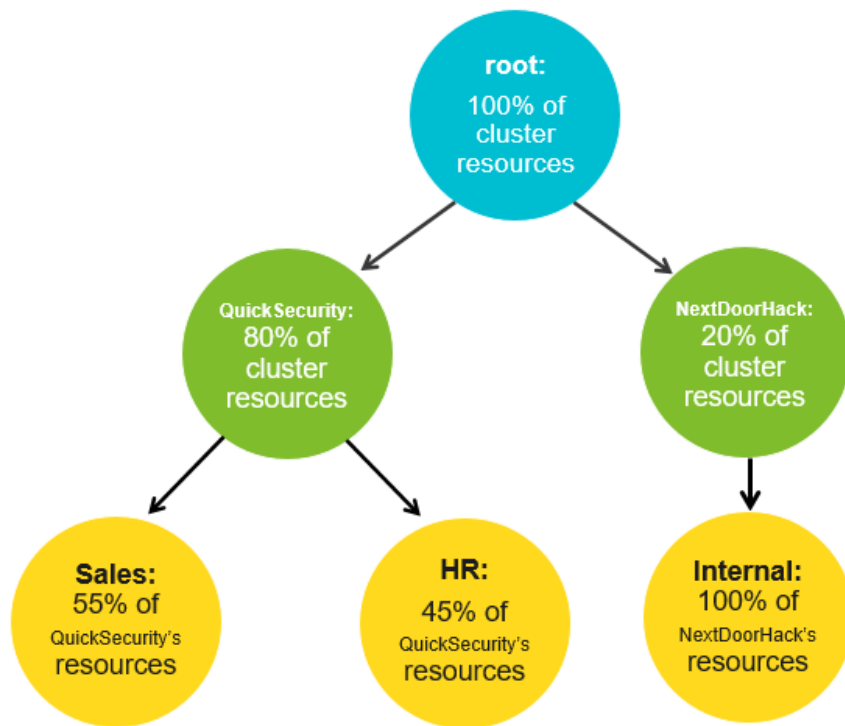
You are the Hadoop administrator for an organization, QuickSecurity. Your organization owns a Hadoop cluster that you share with another organization, NextDoorHack. Due to each organization's processing requirements and stake in the Hadoop cluster, you collaborate and determine that QuickSecurity should be guaranteed 80% of the Hadoop cluster's resources, and NextDoorHack should be guaranteed 20% of the cluster's resources.

QuickSecurity uses the Hadoop cluster to run processes for two departments, HR and Sales. HR requires 45% of the organization's resources, while Sales requires 55% of the organization's resources. NextDoorHack uses the Hadoop cluster specifically for internal data processing.

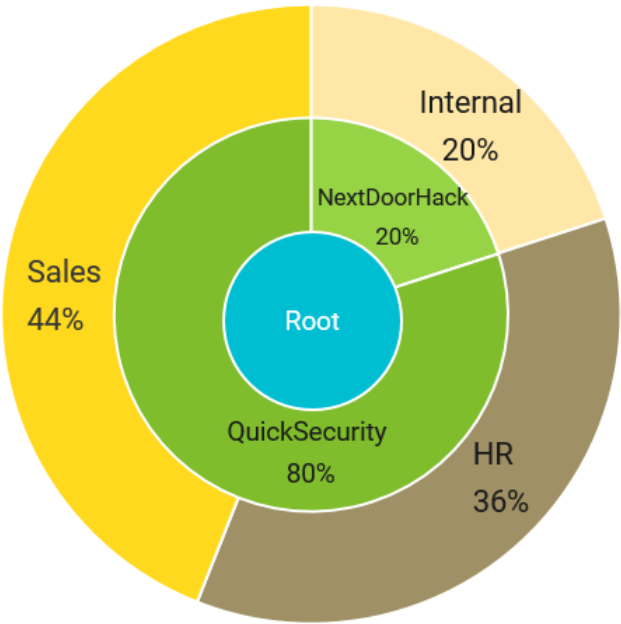
Based on the cluster specification, the Hadoop administrator might design the following queues on the cluster:

Queue Name	Capacity	Description
root	100% of cluster capacity	Root queue.
QuickSecurity	80% of cluster capacity	Queue for the jobs that QuickSecurity runs on the cluster.
NextDoorHack	20% of cluster capacity	Queue for the jobs that NextDoorHack runs on the cluster.
Sales	55% of QuickSecurity's capacity	Queue for the jobs that the Sales department at QuickSecurity runs on the cluster.
HR	45% of QuickSecurity's capacity	Queue for the jobs that the HR department at QuickSecurity runs on the cluster.
Internal	100% of NextDoorHack's capacity	Queue for the jobs that the internal sector at NextDoorHack runs on the cluster.

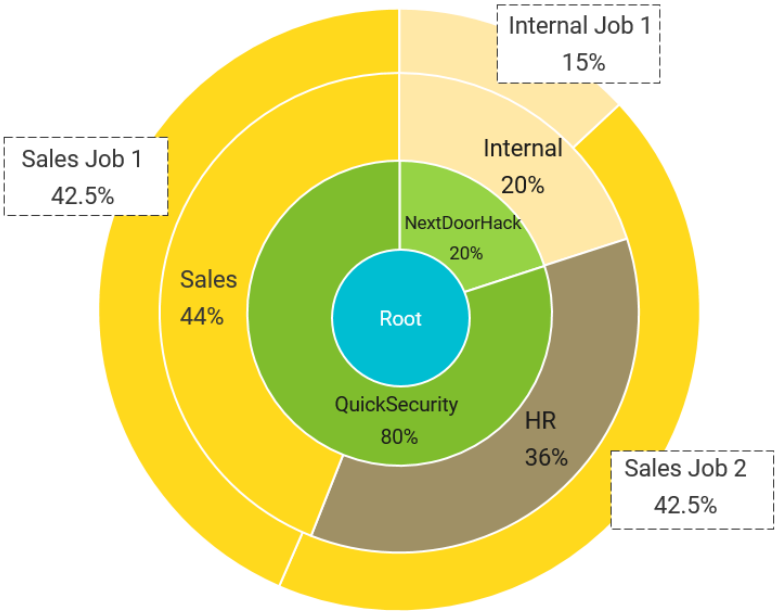
The following image shows the queues and resource allocations:



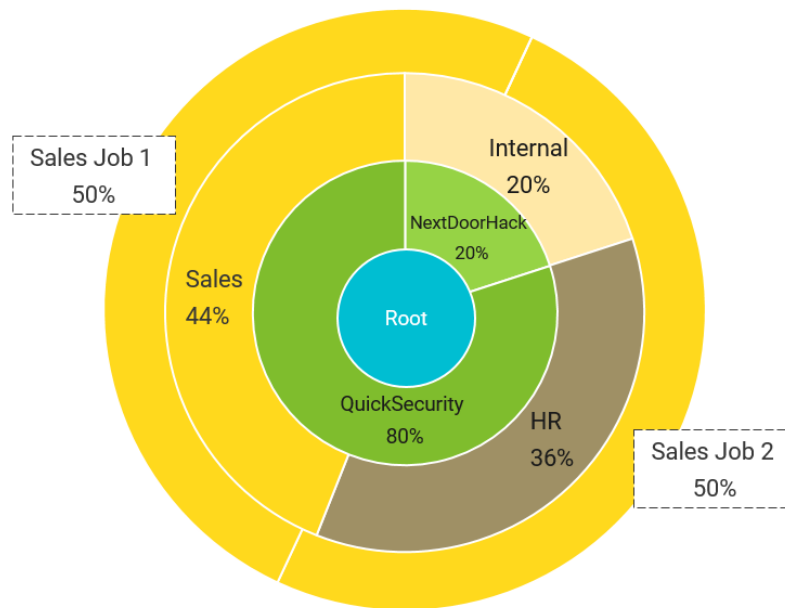
The following image shows the percentage of cluster resources that are guaranteed to jobs in the queues:



The queue `Sales` is running two large jobs and the queue `Internal` is running a small job. The job in the queue `Internal` does not require all of the queue's resources. The following image shows the resources that might be available to each job:



The job in the queue `Internal` completes. The following image shows how the capacity scheduler might reallocate the excess resources to the remaining jobs:



YARN Queues

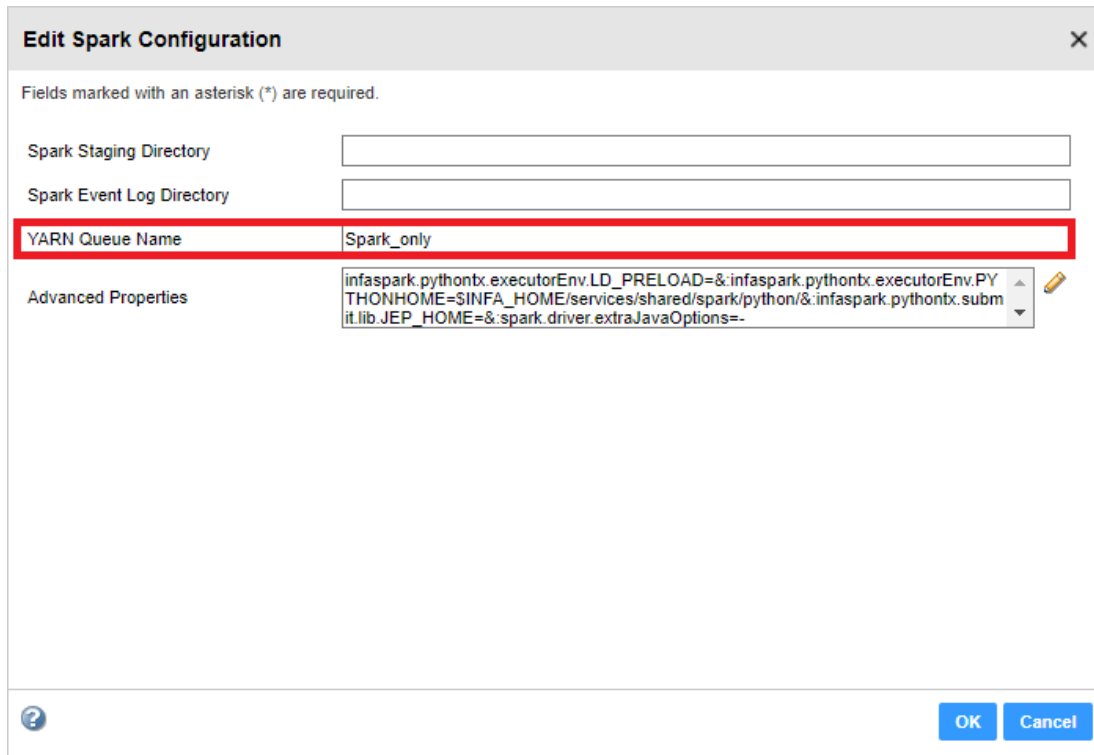
A YARN queue is an organizing structure for jobs that are submitted to the queue.

When you submit a job, a scheduler directs the job to the configured queue. The jobs in the queue might be Blaze, Spark, or Hive jobs. If you do not specify a queue, the jobs are submitted to the default queue.

Example. Directing Spark Jobs to a Queue

You work at an organization that runs a majority of data processing jobs on the Spark engine. To ensure that Spark jobs have access to cluster resources, you direct Spark jobs to the queue `Spark_only`.

To set the YARN queue for Spark jobs, you can configure the following property in the Hadoop connection:



Edit Spark Configuration [X]

Fields marked with an asterisk (*) are required.

Spark Staging Directory

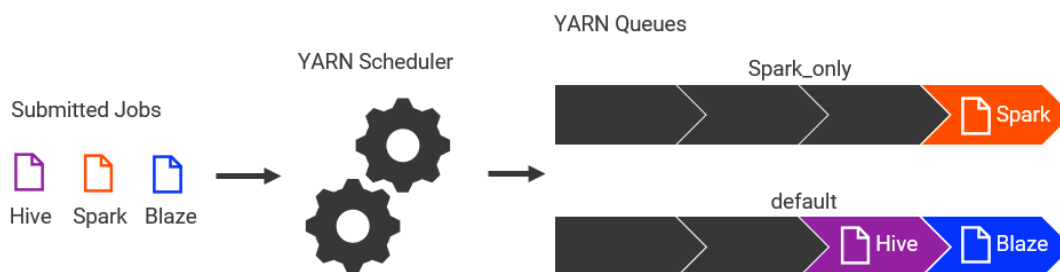
Spark Event Log Directory

YARN Queue Name

Advanced Properties

[?] [OK] [Cancel]

The following image shows how a scheduler directs Spark jobs to the queue that you configured:



The submitted Spark job is directed to the queue `Spark_only`. The submitted Blaze and Hive jobs are directed to the default queue.

Node Labeling

Node labels divide a cluster into partitions. Each partition is a collection of nodes that have specific characteristics.

For example, you label the nodes in the cluster that process data faster compared to other nodes. You might use a label such as `fast`. The nodes that have the label `fast` represent a partition of the Hadoop cluster. Nodes that are not labeled belong to the default partition.

After you label the nodes in the cluster, you can assign the node labels to YARN queues. The jobs in queues run on the nodes that have the node label.

You can also use the node labels to configure the Blaze engine. When you use node labels to configure the Blaze engine, you specify the nodes on the Hadoop cluster where the Blaze engine starts.

Example. Queues and Node Labeling on the Blaze Engine

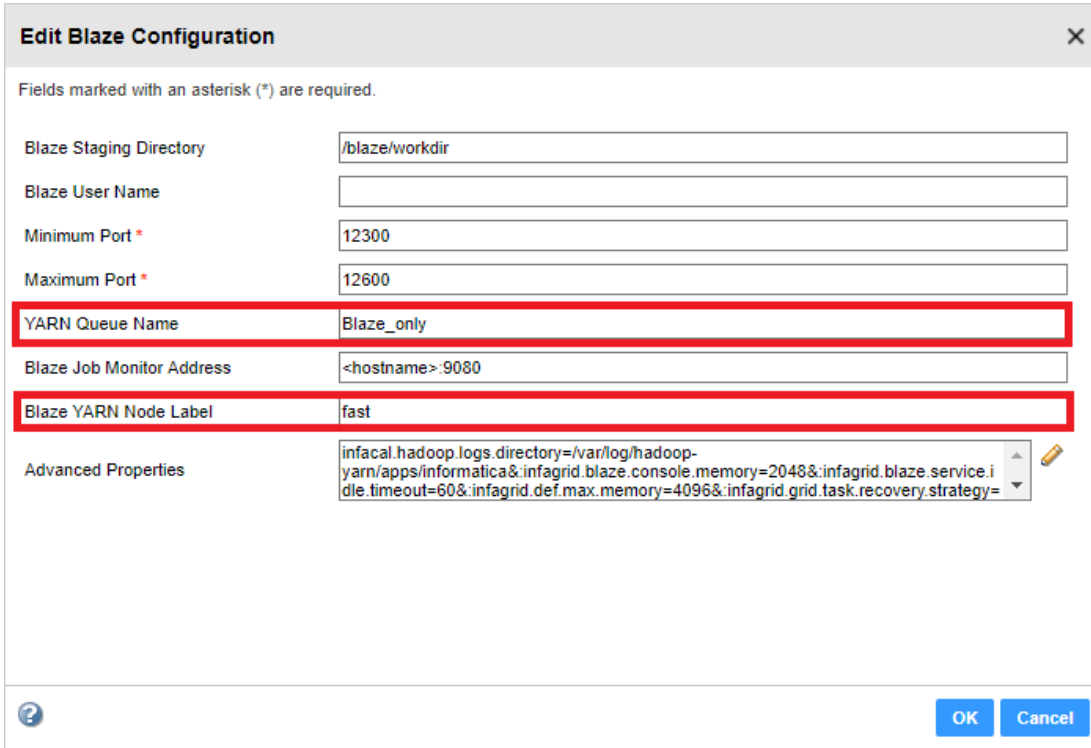
You can configure the Blaze engine to submit Blaze jobs to a YARN queue that is configured with a set of node labels. You can simultaneously configure the Blaze engine using a node label to start the engine on the cluster nodes that have the node label. The node label that you configure for the Blaze engine must be accessible by the YARN queue that receives submitted Blaze jobs.

For example, the Hadoop cluster might be divided into the following partitions:

- memory. Partition of nodes that have higher memory.
- fast. Partition of nodes that have higher CPU.
- default. Default partition.

You configure a YARN queue `Blaze_only` using the node labels `fast` and `default`. When you configure the Blaze engine, you specify the queue `Blaze_only` and the node label `fast`. Note that the YARN queue can access the node label `fast`.

The following image shows the configurations for the Blaze engine in the Hadoop connection:



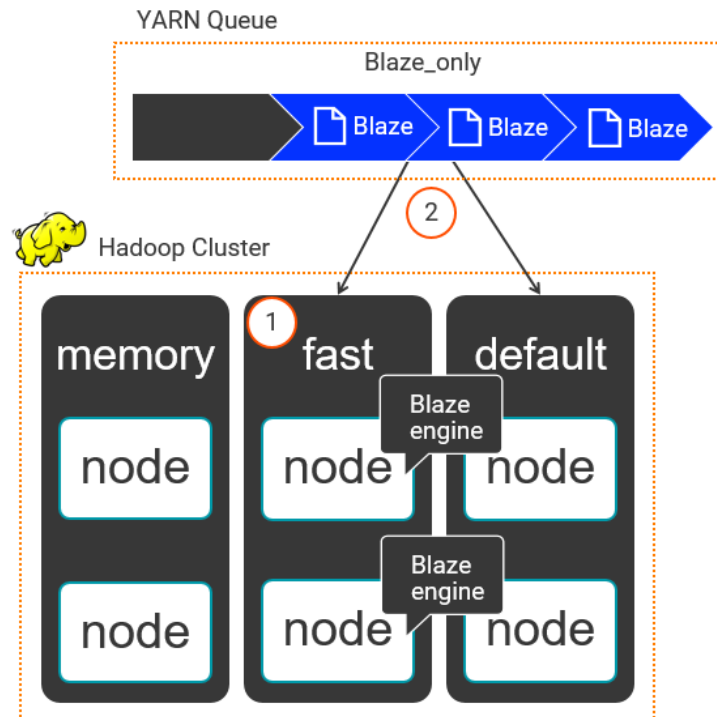
Edit Blaze Configuration [X]

Fields marked with an asterisk (*) are required.

Blaze Staging Directory	/blaze/workdir
Blaze User Name	
Minimum Port *	12300
Maximum Port *	12600
YARN Queue Name	Blaze_only
Blaze Job Monitor Address	<hostname>:9080
Blaze YARN Node Label	fast
Advanced Properties	infacal.hadoop.logs.directory=/var/log/hadoop-yarn/apps/informatica&infagrid.blaze.console.memory=2048&infagrid.blaze.service.idle.timeout=60&infagrid.def.max.memory=4096&infagrid.grid.task.recovery.strategy=

[?] [OK] [Cancel]

The following image shows the nodes where the Blaze engine starts and the nodes that are accessible to the queue:

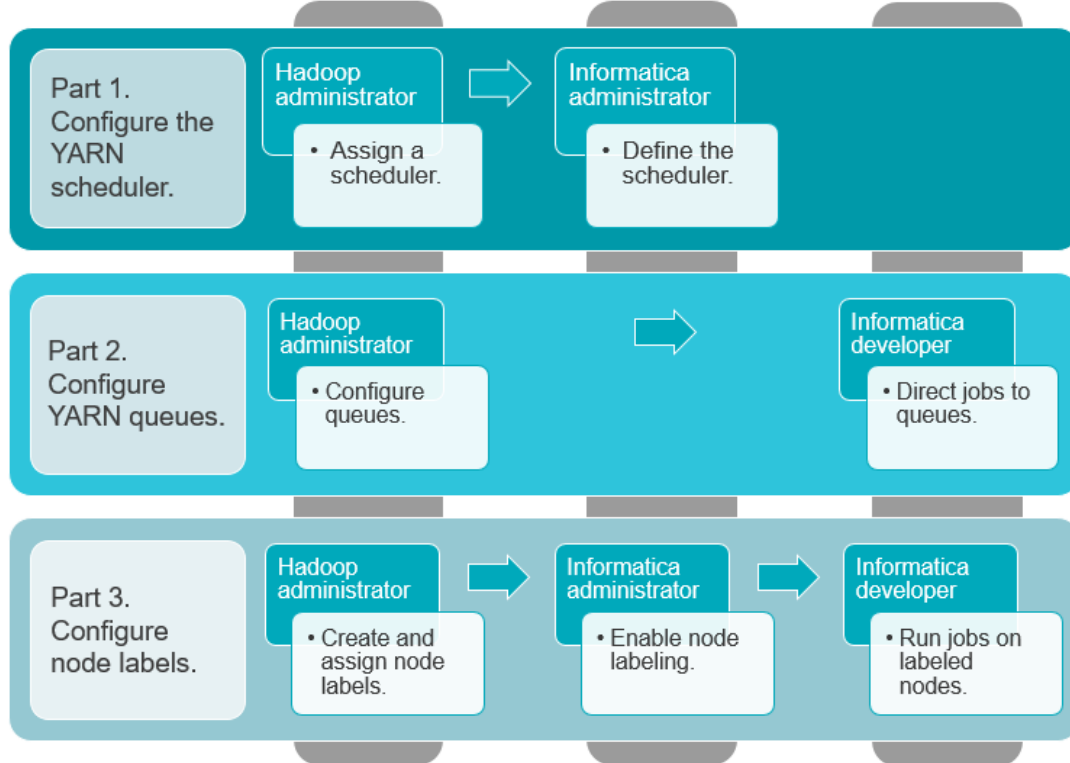


1. The Blaze engine starts on the nodes in the *fast* partition.
2. The queue can access the *fast* and the *default* partitions. The jobs in the queue run on nodes in the *fast* and the *default* partitions.

Configuring Schedulers, Queues, and Node Labels

Configuring a scheduler, queues, and node labels requires Hadoop administrator, Informatica administrator, and Informatica developer tasks.

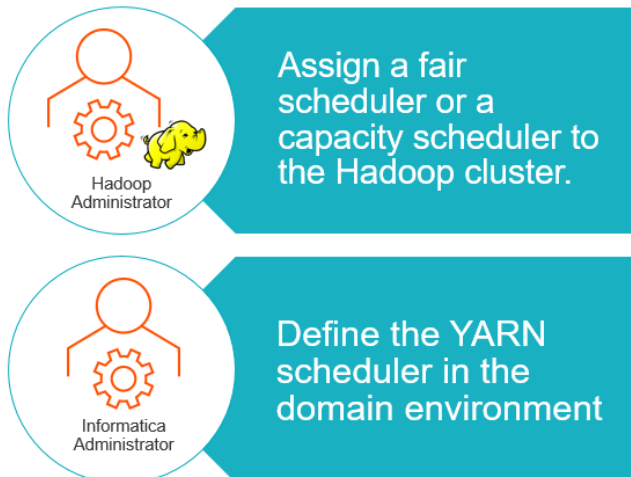
The following image shows the tasks that each role must complete:



Part 1. Configure the YARN Scheduler

Configure the YARN scheduler that you want to use to allocate cluster resources.

The following image shows the tasks that each role must complete:



Hadoop Administrator Tasks

Before an Informatica administrator can define the YARN scheduler in the domain environment, the Hadoop administrator must assign a scheduler to the Hadoop cluster.

The Hadoop administrator must complete the following task:

Assign a scheduler to the Hadoop cluster.

Depending on organizational policies, assign a fair scheduler or a capacity scheduler to determine how cluster resources are assigned.

Informatica Administrator Tasks

After the Hadoop administrator assigns a scheduler to the Hadoop cluster, the Informatica administrator must define the scheduler in the domain environment.

The Informatica administrator must complete the following task:

Define the YARN scheduler.

Define the YARN scheduler to assign resources to jobs that run on the cluster.

In the yarn-site.xml file, configure the following property:

yarn.resourcemanager.scheduler.class

If the Hadoop cluster employs a fair scheduler, use the value

```
org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler
```

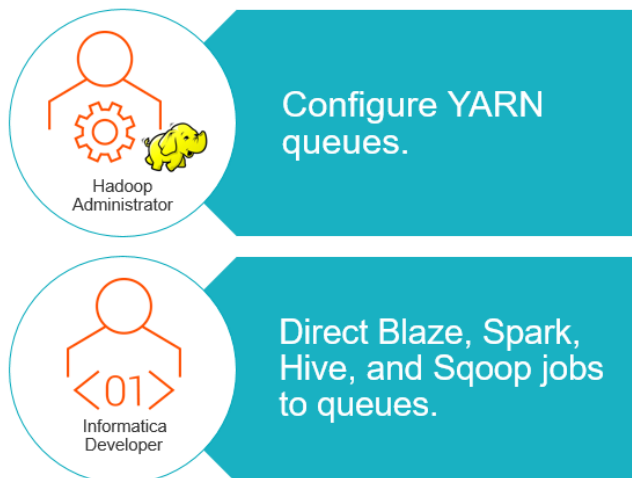
If the Hadoop cluster employs a capacity scheduler, use the value

```
org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler
```

Part 2. Configure YARN Queues

Configure the YARN queues that you want to use and direct the jobs that you run to a specific queue.

The following image shows the tasks that each role must complete:



Hadoop Administrator Tasks

Before an Informatica developer can direct Blaze, Spark, and Hive jobs to YARN queues, the Hadoop administrator must configure the queues on the Hadoop cluster.

The Hadoop administrator must complete the following task:

Configure YARN queues.

Depending on organizational policies, configure YARN queues and allocate cluster resources to the queues that you configure.

Informatica Developer Tasks

After the Hadoop administrator configures YARN queues on the Hadoop cluster, the Informatica developer can direct Blaze, Spark, and Hive jobs to run on specific queues. If the developer does not direct jobs to specific queues, jobs run on the default queue.

Optionally, the Informatica developer can complete the following tasks:

Direct Blaze jobs to a queue.

To direct Blaze jobs to a specific queue, configure the following Blaze configuration property in the Hadoop connection:

Property	Description
YARN Queue Name	The YARN scheduler queue name used by the Blaze engine that specifies available resources on a cluster.

Direct Spark jobs to a queue.

To direct Spark jobs to a specific queue, configure the following Spark configuration property in the Hadoop connection:

Property	Description
YARN Queue Name	The YARN scheduler queue name used by the Spark engine that specifies available resources on a cluster. The name is case sensitive.

Direct Hive jobs to a queue.

To direct MapReduce or Tez jobs on the Hive engine to a specific queue, configure the following Hive connection property:

Property	Description
Environment SQL	SQL commands to set the Hadoop environment. Use the following format: - <code>MapReduce.mapred.job.queue.name=<YARN queue name></code> - <code>Tez.tez.queue.name=<YARN queue name></code> For example, <code>mapred.job.queue.name=root.test</code>

Direct Sqoop jobs on the Spark engine to a queue.

To direct Sqoop jobs for mappings that run on the Spark engine to a specific queue, configure the following JDBC connection property:

Property	Description
Sqoop Arguments	The Sqoop connection-level argument to direct a MapReduce job for a Sqoop mapping to a specific YARN queue. Use the following format: <code>-Dmapred.job.queue.name=<YARN queue name></code>

Direct SQL override mappings on the Blaze engine to a queue.

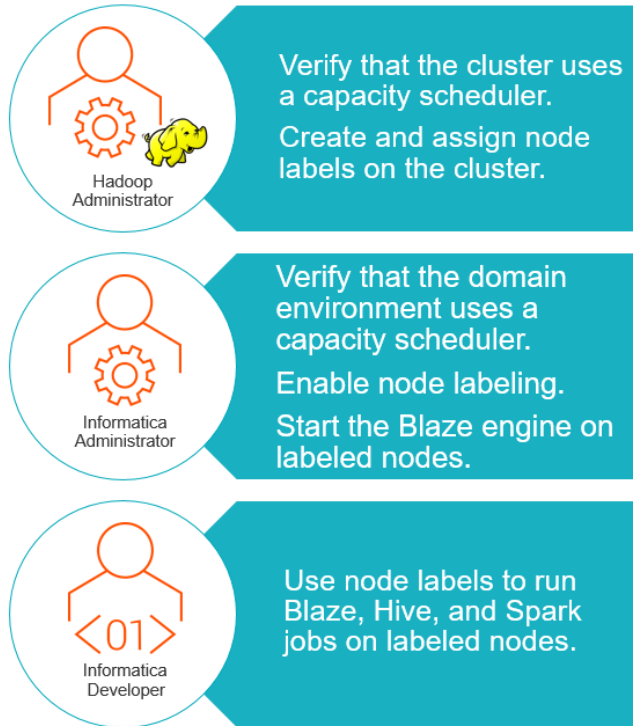
To direct SQL override mappings that run on the Blaze engine to a specific queue, configure the following property in the Hive connection:

Property	Description
Data Access Connection String	The Hive connection string to specify the queue name for Hive SQL override mappings on the Blaze engine. Use the following format: <ul style="list-style-type: none">- <code>MapReduce.mapred.job.queue.name=<YARN queue name></code>- <code>Tez.tez.queue.name=<YARN queue name></code> For example, <code>jdbc:hive2://business.com:10000/default;principal=hive/_HOST@INFAKRB?mapred.job.queue.name=root.test</code>

Part 3. Configure Node Labels

Configure node labels to run jobs on nodes that have specific characteristics.

The following image shows the tasks that each role must complete:



Hadoop Administrator Tasks

Before the Informatica administrator and the Informatica developer can use node labels, the Hadoop administrator must configure the node labels on the Hadoop cluster.

The Hadoop administrator must complete the following tasks:

Verify the scheduler on the cluster.

Ensure that the Hadoop cluster uses a capacity scheduler to allocate resources. If the cluster does not use a capacity scheduler, you cannot configure node labels.

Create and assign node labels to cluster nodes.

Create the node labels that you want to use. Assign the node labels to nodes in the cluster.

Configure YARN queues to use node labels.

Associate the node labels on the cluster with YARN queues, and specify the capacity for each node label. The node labels that a queue can access must be accessible by the queue's parent queue.

Informatica Administrator Tasks

After the Hadoop administrator configures node labels on the Hadoop cluster, the Informatica administrator must enable node labeling in the domain environment.

The Informatica administrator must complete the following tasks:

Verify the scheduler in the domain environment.

Ensure that the domain environment uses a capacity scheduler to allocate resources to jobs that run on the cluster.

Enable node labeling in the domain environment.

To enable node labeling, configure the following property in the yarn-site.xml file:

yarn.node-labels.enabled

```
<property>
  <name>yarn.node-labels.enabled</name>
  <value>TRUE</value>
</property>
```

Create a location to store node labels.

To configure the HDFS directory, configure the following property in the yarn-site.xml file:

yarn.node-labels.fs-store.root-dir

```
<property>
  <name>yarn.node-labels.fs-store.root-dir</name>
  <value>hdfs://[Node name]:[Port]/[Path to store]/[Node labels]</value>
</property>
```

The ResourceManager must be able to access the directory.

To store node labels on a local file system of the ResourceManager instead of HDFS, you can configure a path that is similar to the following path:

```
file:///home/yarn/node-label
```

Optionally, the Informatica administrator can complete the following task:

Start the Blaze engine using a node label.

Use node labels to start the Blaze engine on nodes that have the node label. To start the Blaze engine on nodes with specific labels, configure the following Blaze configuration property in the Hadoop connection:

Property	Description
Blaze YARN Node Label	Node label that determines the node on the Hadoop cluster where the Blaze engine runs. If you do not specify a node label, the Blaze engine runs on the nodes in the default partition. If the Hadoop cluster supports logical operators for node labels, you can specify a list of node labels. To list the node labels, use the operators && (AND), (OR), and ! (NOT).

Note: When the Blaze engine uses node labels, Blaze components might be redundant on the labeled nodes. If a node contains multiple labels and you specify the labels in different Hadoop connections, multiple Grid Manager, Orchestrator, or Job Monitor instances might run on the same node.

Informatica Developer Tasks

After the Hadoop administrators configures the node labels and the Informatica administrator enables node labeling in the domain environment, the Informatica developer can use node labels to run Blaze, Spark, and Hive jobs on nodes that have the node label.

Optionally, the Informatica developer can complete the following task:

Use node labels to run jobs.

To use node labels to run Blaze, Spark, and Hive jobs, the Hadoop administrator must have assigned the node labels to a YARN queue. You must direct the jobs to the queue that is assigned the particular label.

When you run jobs, the jobs are submitted to the configured queue. The node labels that are assigned to the queue determine the nodes where the jobs run.

Author

Margarita Pelyushenko