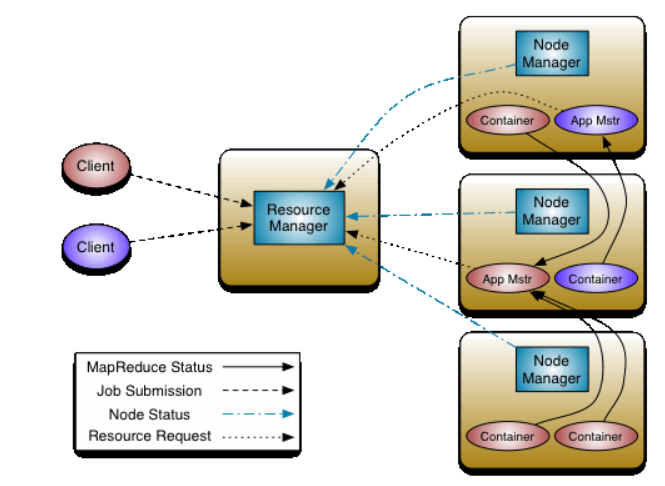
# Deploy do Spark em um cluster Hadoop de Spark

## Apache Hadoop Yarn

Yarn gerencia recursos de um cluster.



(\*) Daemons – processo que roda em segundo plano.

## Parâmetros de configuração do yarn

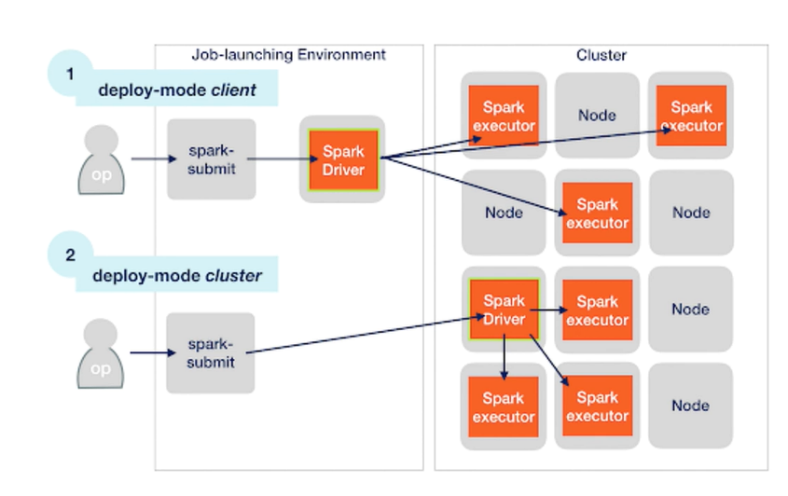
cd /opt/hadoop/etc/hadoop/

gedit yarn-site.xml

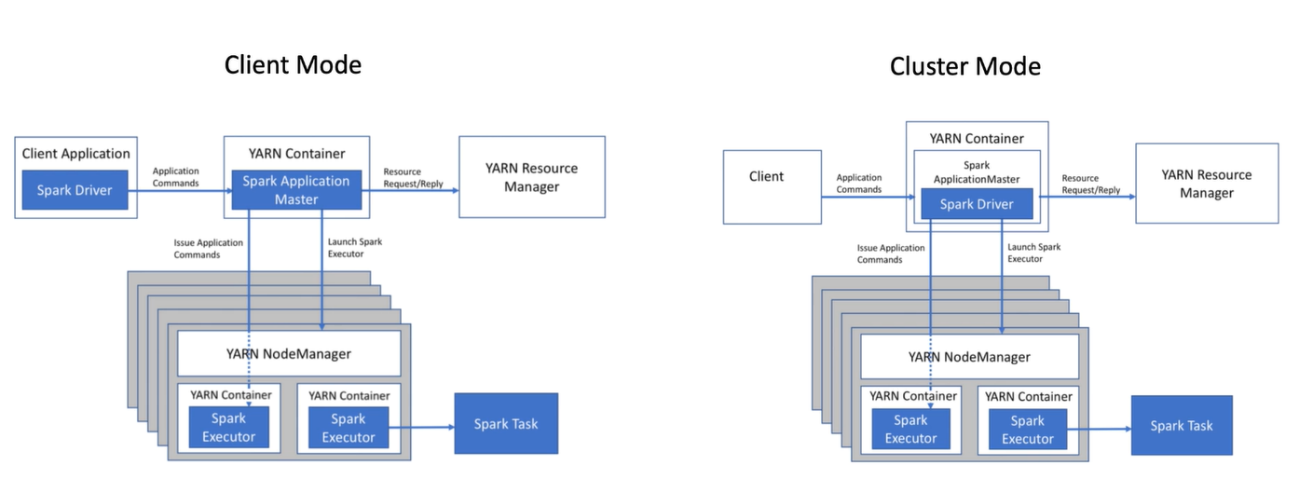
* Para ver os parâmetros do yarn vamos a :<https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html>, no final vai ver [yarn-default.xml](https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-common/yarn-default.xml)
* Nessa lista vai ver o parâmetro: yarn.nodemanager.aux-services.mapreduce\_shuffle.class

## Arquitectura Spark Yarn

O deploy é você pegar sua aplicação e submeter e cluster hadoop. Temos o modo client e cluster. Quando você abre por exemplo o pyspark para que eles funcionem criam um spark context e habilitam o Spark Driver usando a forma de execução client, por outro lado você pode criar um job e agendar para ser executado a noite então você executa de modo cluster. O modo cluster e ideal quando aplicação e muito pesada.



Quando chega um job do Spark o Yarn vai preguntar para o resource manager se tem recursos para executar o job , vai falar divida o job entre dois maquinas, cada tarefa vai rodar em um container dentro de um node manager ou nodes managers que são as maquinas que tem os datanodes.



Como o Spark manda os arquivos de configuração para o yarn:

O Apache Spark pode ser instalado no Master ou remotamente. No Master só bastaria configurar as variáveis de ambiente para o yarn gerenciar os recursos do spark, essa configuração em nosso cluster está em: /opt/hadoop/etc/hadoop/

Mas se o Spark estiver em uma máquina remota, em esse caso o ideal e copiar todo o directorio hadoop e levar para outra máquina remota onde esta o Spark. Lembra não existe instalação e só copiar os arquivos.

Diagrama

Descrição gerada automaticamente

## Configurar o yarn com spark na mesma máquina:

* Uma opção é em opt/spark/conf/ configurar em spark-env.sh a variável

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

* Mas como estamos na mesma máquina podemos configurar uma variável de ambiente, para ver todas as variáveis podemos digitar env:

env

clear

env | grep HADOOP

gedit .bashrc

Agregar em .bashrc

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

Para habilitar salvar e digitar no terminal:

source .bashrc

## Executar spark em modo client

Vamos executar spark nossa máquina virtual, onde o master vai ser o yarn e qual modo e quero em este caso modo client:

spark-shell --master yarn --deploy-mode client

Podemos accesar desde a máquina física do navegador para ver o resource manager na porta 8088 e para ver o node manager na porta 8042, veia que estou apontando ao endereço ip de minha máquina virtual (para saber o endereço é só digitar ifconfig em sua máquina virtual):

http://192.168.0.81:8088/cluster

<http://192.168.0.81:8042/node> (se tiver mais de um node manager você vai ter diferentes ip com a mesma porta 8042)

Em About você tem a página de entrada pode ver que o ResourceManager HÁ Zoopkeeper está desabilitado porque você pode ter dois máquinas com ResourceManager em caso de que uma falha e precisaria configurar qual máquina é a leader elector, por padrão a segurança está desabilitada, se vez ele efetua o login com dr.who, como se fosse um usuário anônimo. A segurança tem que ser configurada por padrão podemos usar o kerberos.

## Executando em modo cluster

Para someter vamos usar spark-summit. Vamos criar um arquivo sh para por exemplo agendar para executar tudo dia a noite, poderia até utilizar o chrome do Linux.

Vamos criar o arquivo em nossa máquina virtual: O driver-memory é o processo spark que vai inicializar o processo da aplicação ele pode rodar no master. Dentro de cada máquina no cluster eu tenho o executor. Os quais vão ser inicializados pelo yarn. Ademais pode definir a quantidade de cores. O número 10 é para ver os comandos de conversa com o yarn.

gedit app.sh

spark-submit --class org.apache.spark.examples.SparkPi \

--master yarn \

--deploy-mode cluster \

--driver-memory 4g \

--executor-memory 2g \

--executor-cores 1 \

/opt/spark/examples/jars/spark-examples\*.jar \

10

Salva o arquivo é para executar o arquivo sh tem que dar o privilégio de execução no terminal da maquina virtual:

chmod 755 app.sh

./app.sh

