GFTS BigData POC Summary

Contents

[Cluster Information 3](#_Toc445465412)

[Nodes 3](#_Toc445465413)

[Servers 3](#_Toc445465414)

[Convenient Commands 4](#_Toc445465415)

[Cluster Architecture 5](#_Toc445465416)

[Hierarchical Architecture 5](#_Toc445465417)

[6](#_Toc445465418)

[Relationship Architecture 6](#_Toc445465419)

[Cluster Brief Introduction 7](#_Toc445465420)

[HDFS 7](#_Toc445465421)

[Reading data from HDFS 7](#_Toc445465422)

[Writing data to HDFS 7](#_Toc445465423)

[Use Linux command line to connect to HDFS (Arbitrary Node) 8](#_Toc445465424)

[Use web browser to connect to HDFS 8](#_Toc445465425)

[Pig 9](#_Toc445465426)

[Pig Execution Modes: 9](#_Toc445465427)

[Configure pig on Hadoop Cluster 9](#_Toc445465428)

[How to run pig in shell 9](#_Toc445465429)

[How to run a pig job in java code: 9](#_Toc445465430)

[YARN 10](#_Toc445465431)

[How YARN runs an application 10](#_Toc445465432)

[Use web browser to check an application on Yarn 11](#_Toc445465433)

[Drill 11](#_Toc445465434)

[How to install distributed mode Drill on the cluster? 11](#_Toc445465435)

[How to start/stop/restart Drillbit and use shell to connect to Drill in distributed mode? 11](#_Toc445465436)

[How to use java to connect to Drill? 11](#_Toc445465437)

[How to enable HDFS and Hive storage plugins? 12](#_Toc445465438)

[Use web browser to execute query 13](#_Toc445465439)

[Use web browser to check query information (profiles) 13](#_Toc445465440)

[References: 13](#_Toc445465441)

[Hive on Tez 14](#_Toc445465442)

[How to start hive? 14](#_Toc445465443)

[How to set hive’s execution engine to tez on beeline? 14](#_Toc445465444)

[How to check hive services’ status on slave03? 14](#_Toc445465445)

[How to use beeline to connect to hive on slave03? 14](#_Toc445465446)

[How to use WebHCat RESTful API to operate Hive? 14](#_Toc445465447)

[How to use java to connect to HiveServer2? 14](#_Toc445465448)

[References: 15](#_Toc445465449)

[[4]. WebHCat RESTful API 15](#_Toc445465450)

[Spark 15](#_Toc445465451)

[Spark cluster information 15](#_Toc445465452)

[Spark configurations 16](#_Toc445465453)

[Spark-jobserver 16](#_Toc445465454)

[Spark history server 16](#_Toc445465455)

[Standalone Mode 16](#_Toc445465456)

[Yarn-client Mode 17](#_Toc445465457)

[Yarn-cluster Mode 17](#_Toc445465458)

[How to use java to connect to Spark cluster 17](#_Toc445465459)

[References: 17](#_Toc445465460)

[ZooKeeper 18](#_Toc445465461)

[Zookeeper install 18](#_Toc445465462)

[Zookeeper management 18](#_Toc445465463)

[Compare 19](#_Toc445465464)

[Cluster Computing Framework: MapReduce, Tez and Spark 19](#_Toc445465465)

[SQL-on-Hadoop: Hive and Drill 19](#_Toc445465466)

[Test Result 19](#_Toc445465467)

[MemSQL 19](#_Toc445465468)

[Use web browser to access MemSQL cluster 20](#_Toc445465469)

[ETL on MemSQL with Spark 20](#_Toc445465470)

# Cluster Information

## Nodes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alias | Node | Host | IP | Mem Avail |
| N-0 | gft.hadoop.**master** | vm-45e5-3412 | 10.116.37.181 | 8G |
| N-1 | gft.hadoop.**slave01** | sd-c8a3-bb0d | 169.172.133.57 | 32G |
| N-2 | gft.hadoop.**slave02** | sd-dfad-52de | 169.172.133.59 | 32G |
| N-3 | gft.hadoop.**slave03** | sd-4261-f1e1 | 169.172.134.169 | 32G |
| N-4 | gft.hadoop.**slave04** | sd-6286-1278 | 169.172.134.168 | 32G |
| N-5 | gft.hadoop.**slave05** | sd-023d-317b | 169.172.134.167 | 32G |
| N-6 | gft.hadoop.**slave06** | sd-4ee9-cecd | 169.172.133.58 | 32G |
| Notes:   1. Node JDK version is 1.8; | | | | |

## Servers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Server | Directory | On Node(s) | Staple Server (port) | How to start | How to stop |
| Hadoop | ~/hadoop | N-0~6 | *HDFS + YARN* | sbin/start-all.sh | sbin/stop-all.sh |
| HDFS | ~/hadoop | N-0~6 | [Web UI (50070)](http://vm-45e5-3412:50070/) | sbin/start-dfs.sh | sbin/stop-dfs.sh |
| YARN | ~/hadoop | N-0~6 | [Web UI (8088)](http://vm-45e5-3412:8088/cluster) | sbin/start-yarn.sh | sbin/stop-yarn.sh |
| Yarn Timeline Server | ~/hadoop | N-0 | [Web UI(8188)](http://vm-45e5-3412:8188/applicationhistory) | yarn-daemon.sh start timelineserver | ./yarn-daemon.sh stop timelineserver |
| MapReduce JobHistory Server | ~/hadoop | N-0 | Port (19888) | mr-jobhistory-daemon.sh start | mr-jobhistory-daemon.sh stop |
| Spark | ~/spark | N-0(Master)  N-4,5,6 (Worker) | Master (7077)  [Standalone Web UI (8888)](http://vm-45e5-3412:8888/) | sbin/start-all.sh | sbin/stop-all.sh |
| Spark History Server | ~/spark | History records stored on HDFS | [History Server (18080)](http://vm-45e5-3412:18080/) | sbin/start-history-server.sh | sbin/stop-history-server.sh |
| [Spark-JobServer](https://github.com/spark-jobserver/spark-jobserver) | ~/spark/job-server | N-0 | [Web UI (8090)](http://vm-45e5-3412:8090/) | server\_start.sh | server\_stop.sh |
| Zookeeper | ~/zookeeper/zookeeper-3.4.6 | N-3,4,5 |  | bin/zkServer start | bin/zkServer stop |
| Hive | ~/hive | N-3 | HiveServer2 (10000)  Metastore Server (9083) | 1. bin/hive --service metastore &  2./bin/hive --service hiveserver2 & | kill -9 pid |
| WebHCat | ~/hive/hcatalog | N-3 | [WebHCat RESTful API](https://cwiki.apache.org/confluence/display/Hive/WebHCat+Reference)  [WebHCat (50111)](http://sd-4261-f1e1:50111/templeton/v1/status?user.name=bigdatagfts) | sbin/webhcat\_server.sh start | sbin/webhcat\_server.sh stop |
| HWI | ~/hive |  | [HWI (9999)](http://sd-4261-f1e1:9999/hwi/) | bin/hive --service hwi | kill -9 pid |
| Tez | ~/tez-0.7.0 | N-0,1,2,3,4,5,6 | As a execution engine | - | - |
| Tez UI | ~/tez-0.7.0 | N-3 | [Web UI (8080)](http://sd-4261-f1e1:8080/tez-ui-0.7.0/#/) | [Tomcat] | [Tomcat] |
| Drill | ~/drill-1.4.0 | N-1,2,3,4,5,6 | [Drillbit Web UI (8047)](http://sd-6286-1278:8047/query) | bin/drillbit.sh start | bin/drillbit.sh stop |
| Pig | ~/pig-0.15.0 | N-1 |  | pig | quit |
| Presto | ~/presto/ presto-server-0.133 | N-3 coordinator  N-2,4,5,6,7 worker |  | bin/launcher start | bin/launcher stop |
| yanagishima | ~/presto/yanagishima | N-6 | [Web UI (9998)](http://sd-4ee9-cecd:9998/) | bin/yanagishima-start.sh | bin/yanagishima-shutdown.sh |
| Zeppelin | ~/zeppelin-0.5.6/ | N-5 | [Web UI](http://sd-023d-317b:8080/zeppelin/#/) | ./ zeppelin-daemon.sh start | ./ zeppelin-daemon.sh stop |
| Kafka | ~/kafka\_2.10-0.9.0.1 | N-3,4,5 |  | [Check chapter Kafka](#_Kafka) | [Check chapter Kafka](#_Kafka) |
| Notes.   1. All nodes’ user is bigdatagfts, so the home directory (~) is /home/bigdatagfts; 2. Tez is an execution engine of hive on cluster; 3. Tez needs create tez session when started by hive; 4. Tez can replace the mapreduce, you need to modify mapred-site.xml to change “mapreduce.framework.name” property from its default value of “yarn” to “yarn-tez” | | | | | |

## Convenient Commands

|  |  |  |  |
| --- | --- | --- | --- |
| Server | Node(s) | Command(s) | Description |
| Presto | N-0 | ~/startpresto  ~/stoppresto  ~/restartpresto | Start/stop/restart presto server; |
| Drill | N-0 | ~/startdrill  ~/stopdrill  ~/restartdrill | Start/stop/restart drill server; |
| Zookeeper | N-0 | ~/startzk  ~/stopzk  ~/restartzk | Start/stop/restart zookeeper server; |
| Hadoop | N-0 | starthadoop  stophadoop | Start/stop Hadoop Cluster (YARN + HDFS) |
| Spark | N-0 | startspark  stopspark | Start/stop Spark Cluster |
|  |  |  |  |

# Cluster Architecture

## Hierarchical Architecture

HDFS

ORACLE

MEMSQL

MACHINE

MACHINE

MACHINE

MACHINE

…

YARN

SPARK STANDLONE

MAPREDUCE

TEZ

SPARK

HIVE

PIG

ZOOKEEPER

SQOOP (ETL)

MACHINE

DRILL

JAVA API

RESTful API

JAVA PROGRAM (e.g. spark-job-server)

SPARK-JOBSERVER

WEB UI

Application

API

Processing Frameworks

Storage

Distributed Computing

Resource Management

Hardware



Execution engine



Query



ETL

Extract data



Store/Fetch data

Store/Fetch metadata



Execution engine

Transform & Load data

Store/Fetch data

Query

Store/Fetch data

Execution engine

Storage

Distributed Computing

SQL-on-Hadoop

Store/Fetch data



Manage Resource

Maintain configuration information

Resource Management

Offer

Offer

API

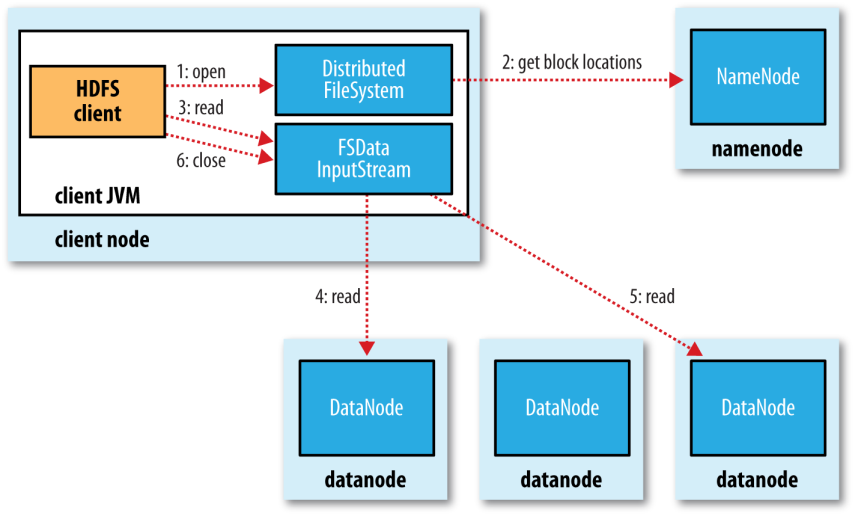
## Relationship Architecture

# Cluster Brief Introduction

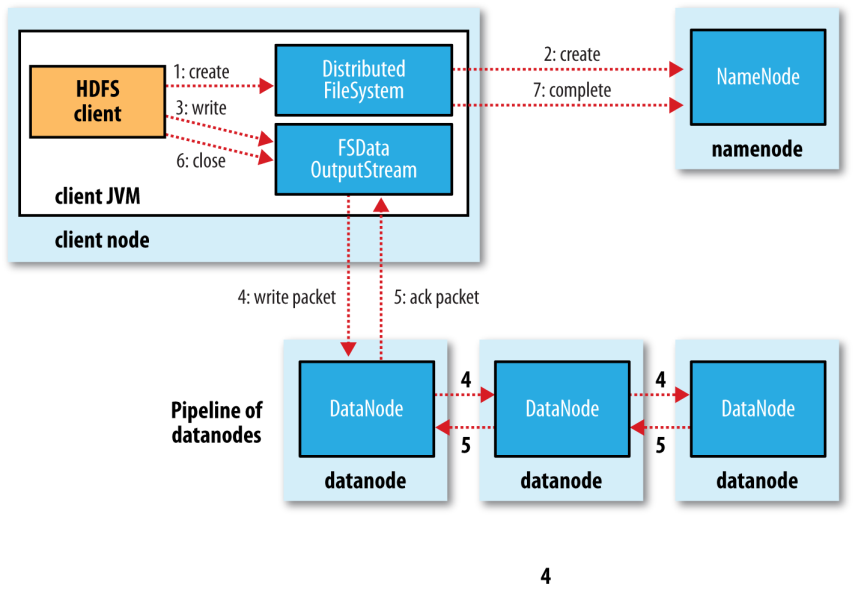
## HDFS

HDFS is the primary distributed storage used by Hadoop applications. A HDFS cluster primarily consists of a **NameNode** that manages the file system metadata and **DataNodes** that store the actual data. HDFS has the concept of a **block—128 MB** by default and it can create **replication** for us automatically.

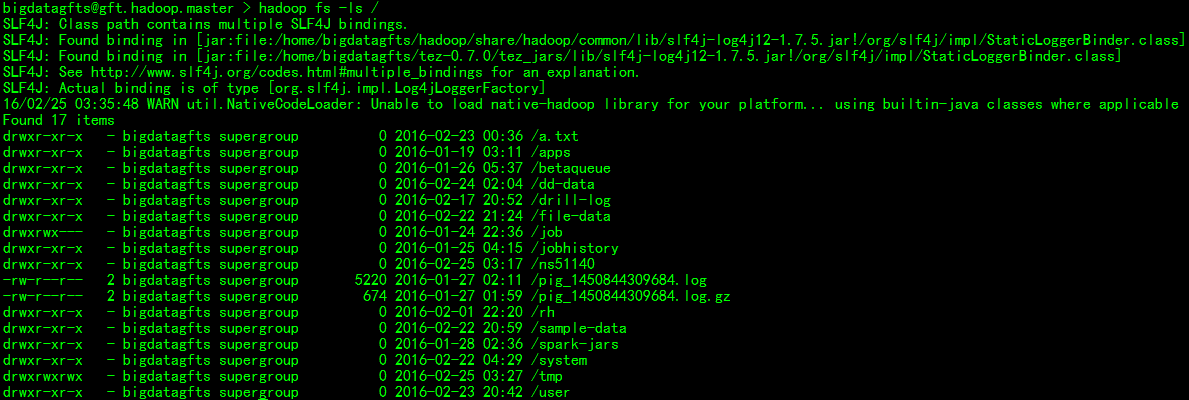
### Reading data from HDFS



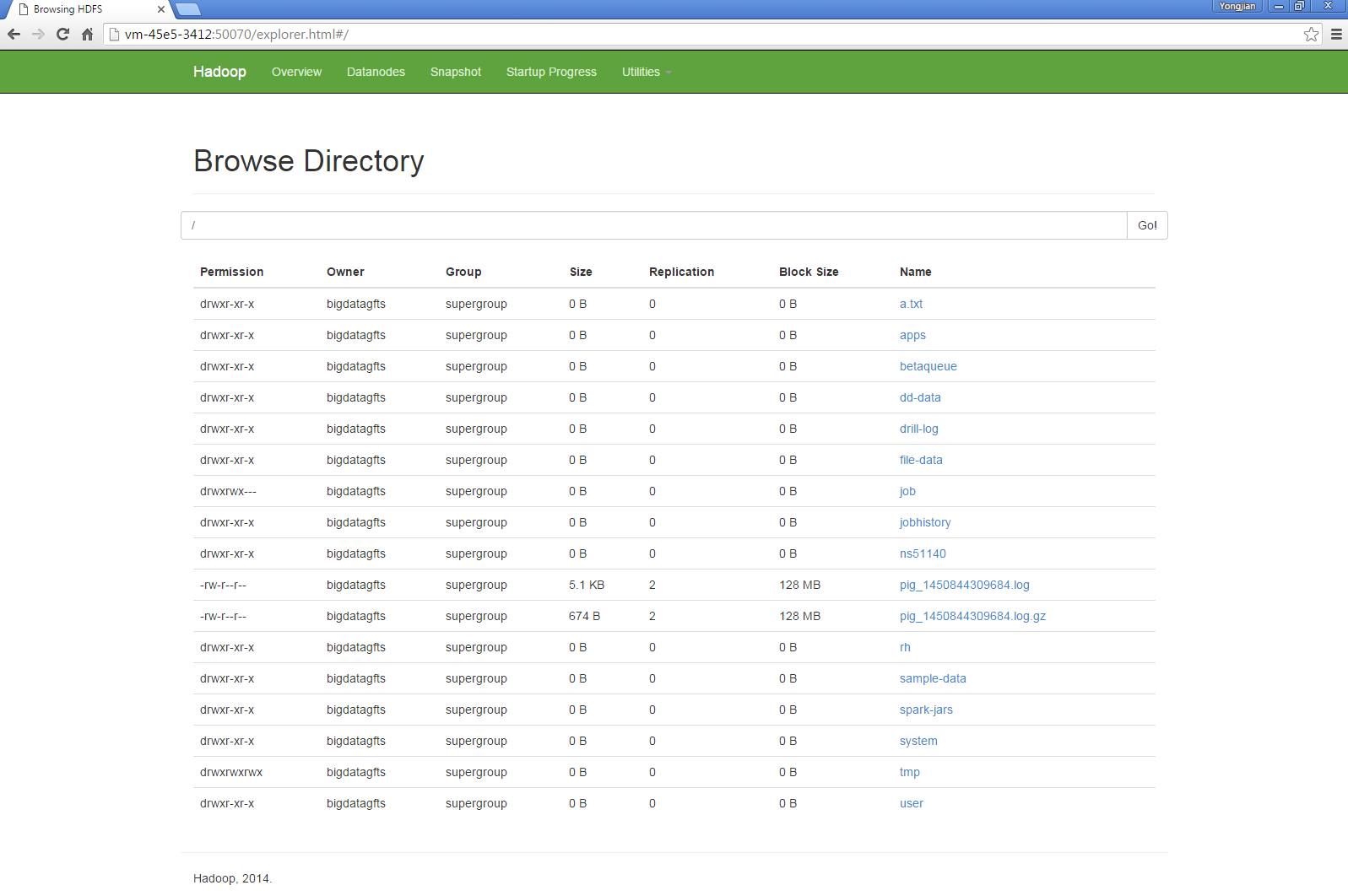
### Writing data to HDFS



### Use Linux command line to connect to HDFS (Arbitrary Node)



### [Use web browser to connect to HDFS](http://vm-45e5-3412:50070/explorer.html#/)



## Pig

**Apache Pig** is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

At the present time, Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig's language layer currently consists of a textual language called Pig Latin, which has the following key properties:

* **Ease of programming.** It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.
* **Optimization opportunities.** The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.
* **Extensibility.** Users can create their own functions to do special-purpose processing.

### Pig Execution Modes:

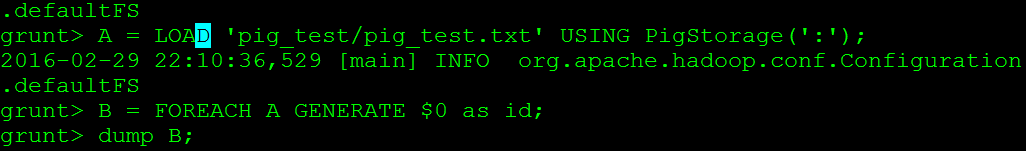
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Local Mode | Tez Local Mode | Mapreduce Mode | Tez Mode |
| Interactive Mode | yes | experimental | yes | yes |
| Batch Mode | yes | experimental | yes | yes |

### Configure pig on Hadoop Cluster

Tell pig the directory destination of hadoop conf by setting the PIG\_CLASSPATH environment variable .Note that this must to the directory that the XML file is in,not the file itself.Pig will read all XML and properties files in that directory.

### How to run pig in shell

choose execution mode pig -x local/tez\_local/mapreduce(default)/tez



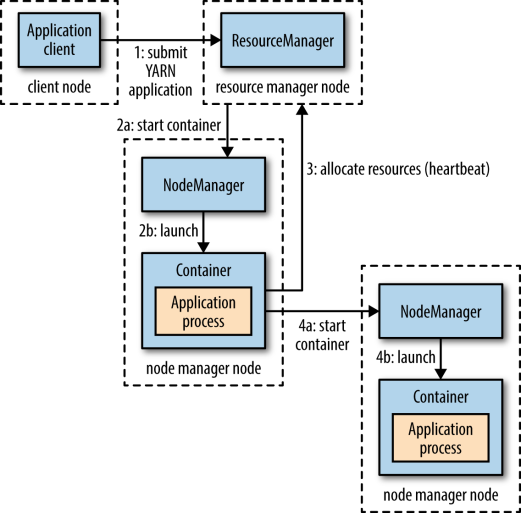
### How to run a pig job in java code:

[run\_pig\_java\_program\_on\_eclipse\_from\_windows\_guide](file:///\\Shavnascsts0001\grp_icggfts1\CIBTech%20II\GFTSFinance\HadoopWork\run_pig_java_program_on_eclipse_from_windows_guide.docx).

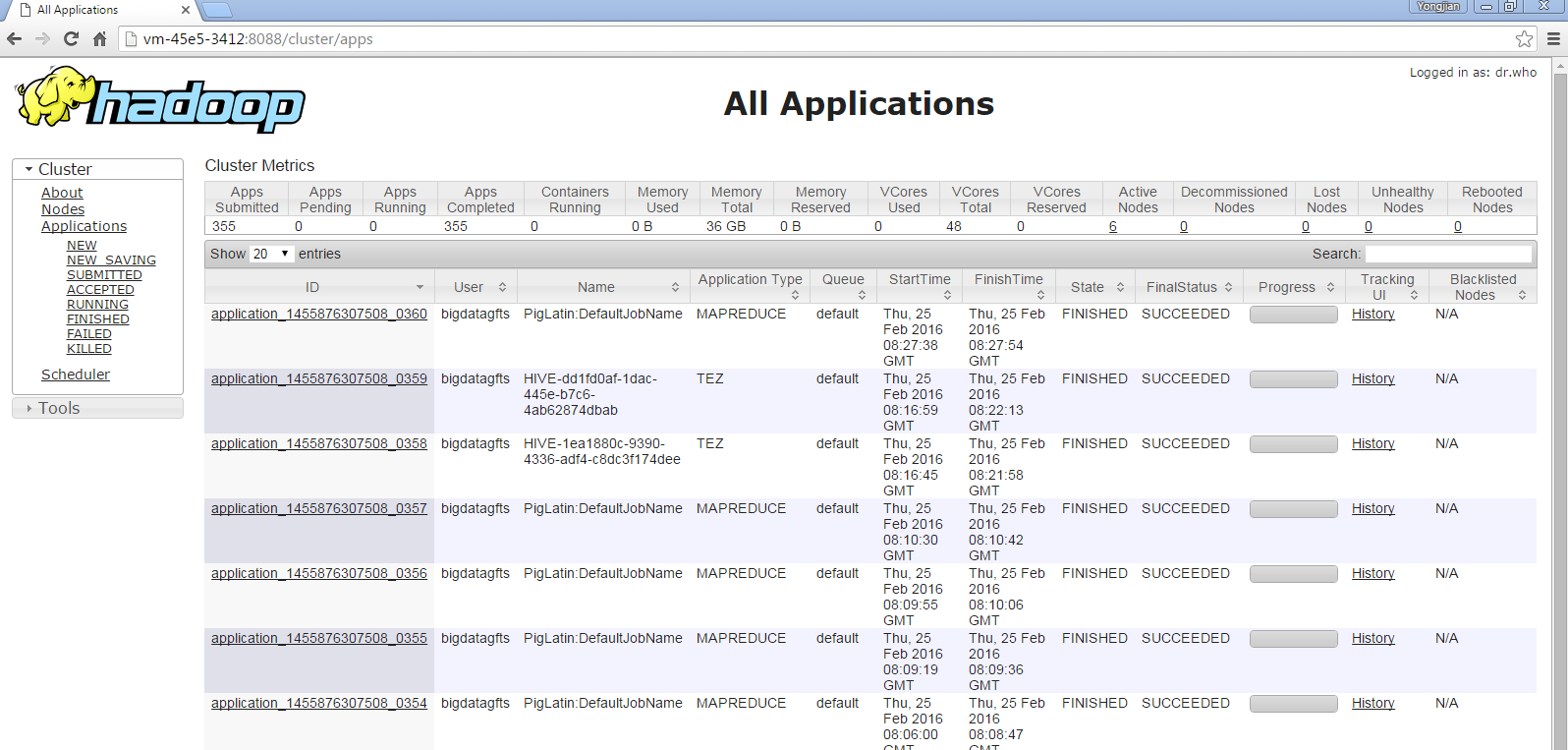
## YARN

YARN is Hadoop’s cluster resource management system. YARN provides its core services via two types of long-running daemon: **a resource manager** (one per cluster) to manage the use of resources across the cluster, and **node managers** running on all the nodes in the cluster to launch and monitor containers. A **container** executes an application-specific process with a constrained set of resources (memory, CPU, and so on).

### How YARN runs an application



### [Use web browser to check an application on Yarn](http://vm-45e5-3412:8088/cluster/apps)



## Drill

Drill is an Apache open-source **SQL query engine** for Big Data exploration. Drill is designed from the ground up to support high-performance analysis on the semi-structured and rapidly evolving data coming from modern Big Data applications, while still providing the familiarity and ecosystem of ANSI SQL, the industry-standard query language. Drill provides plug-and-play integration with existing **Apache Hive** and **Apache HBase** deployments.

### [How to install distributed mode Drill on the cluster?](https://drill.apache.org/docs/installing-drill-on-the-cluster/)

drill-override.conf on our cluster:



sys.store.provider.zk.blobroot: Destination to store drill query profiles.

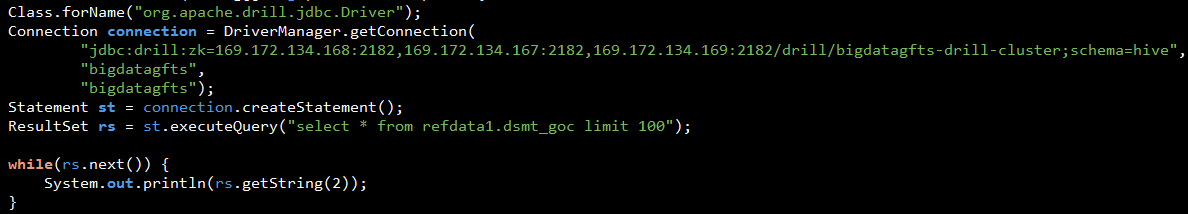
### [How to start/stop/restart Drillbit and use shell to connect to Drill in distributed mode?](https://drill.apache.org/docs/starting-drill-in-distributed-mode/)

### [How to use java to connect to Drill?](https://drill.apache.org/docs/using-the-jdbc-driver/)

Drill JDBC Driver is in: *<drill-installation\_directory>/jars/jdbc-driver/drill-jdbc-all-<version>.jar*

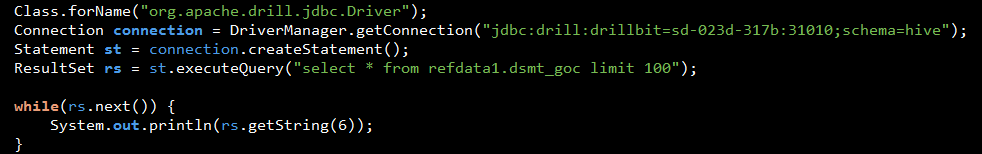
Using ZooKeeper to **randomly** choose a Drillbit in the cluster:

JDBC URL is jdbc:drill:zk=169.172.134.168:2182,169.172.134.167:2182,169.172.134.169:2182/drill/bigdatagfts-drill-cluster



Using the JDBC URL Format for a **Direct** Drillbit Connection:

JDBC URL is jdbc:drill:drillbit=sd-023d-317b:31010



### [How to enable HDFS and Hive storage plugins?](https://drill.apache.org/docs/hive-storage-plugin/)

a. [Use web browser to configure storage plugins](http://sd-6286-1278:8047/storage)

*b. Click Update -> [Update Hive/HDFS configuration] -> Click Update*

Hive storage plugin configuration on our cluster:

{

"type": "hive",

"enabled": true,

"configProps": {

"hive.metastore.uris": "thrift://169.172.134.169:9083",

"javax.jdo.option.ConnectionURL": "jdbc:oracle:thin:@//vm-f221-fe40.nam.nsroot.net:1528/PRIMEDB",

"hive.metastore.warehouse.dir": "/user/hive/warehouse",

"fs.default.name": "hdfs://gft.hadoop.master:9000/",

"hive.metastore.sasl.enabled": "false"

}

}

HDFS storage plugin configuration on our cluster:

{

"type": "file",

"enabled": true,

"connection": "hdfs://gft.hadoop.master:9000/",

"workspaces": {

"root": {

"location": "/",

"writable": true,

"defaultInputFormat": null

},

"dd": {

"location": "/sample-data",

"writable": true,

"defaultInputFormat": null

},

"tmp": {

"location": "/tmp",

"writable": true,

"defaultInputFormat": null

}

}

…

}

### [Use web browser to execute query](http://sd-6286-1278:8047/query)

### [Use web browser to check query information (profiles)](http://sd-6286-1278:8047/profiles)

### References:

[1]. [Drill Document](https://drill.apache.org/docs/drill-introduction/)

[2]. [How to use Drill with BI Tools?](https://drill.apache.org/docs/using-drill-with-bi-tools-introduction/)

## Hive on Tez

Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called **HiveQL**. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.

### How to start hive?

*a. Start hive metastore:*

**bin/hive --service metastore &**

*b. Start hiveserver2*

**bin/hive --service hiveserver2 &**

Note: You need to check the log file in /tmp/bigdatagfts/hive.log to guarantee metastore service and hiveserver2 has started.

### How to set hive’s execution engine to tez on beeline?

**set hive.execution.engine=tez;**

### How to check hive services’ status on slave03?

**netstat -an | grep 9083** # check metastore service status.

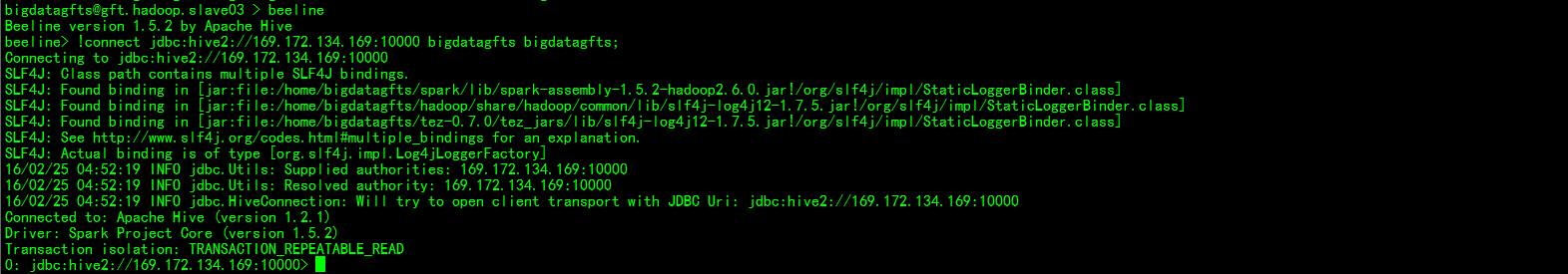
**netstat -nl | grep 10000** # check hiveserver2 service status.

[Click here](http://sd-4261-f1e1:50111/templeton/v1/status?user.name=bigdatagfts) to Check WebHCat status.

### How to use beeline to connect to hive on slave03?

**bigdatagfts@gft.hadoop.slave03 > beeline**

**beeline> !connect jdbc:hive2://169.172.134.169:10000 bigdatagfts bigdatagfts;**

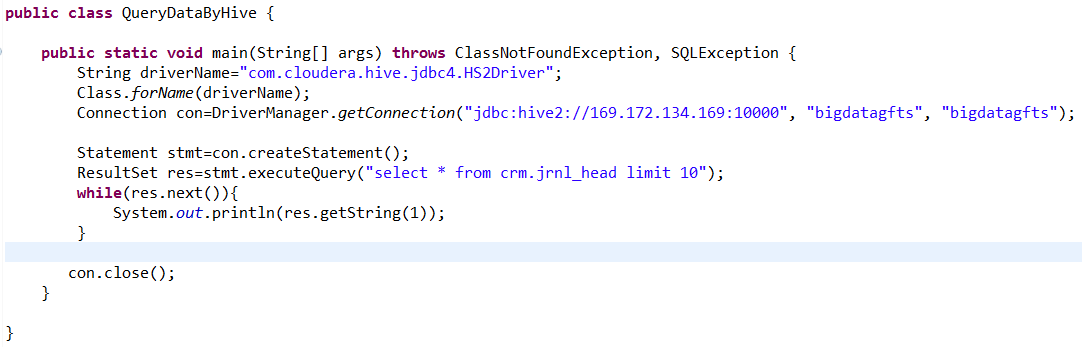


### [How to use WebHCat RESTful API to operate Hive?](https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat)

### How to use java to connect to HiveServer2?

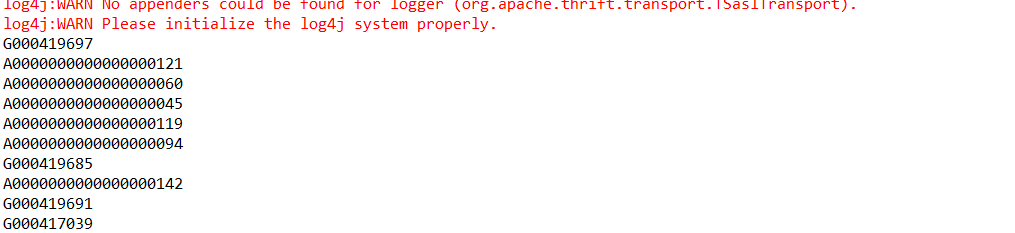
Step1:Import HS2Driver lib

Step2:Code in IDE (below example in Eclipse)



Coding is very similiar to JDBC.

Results printed on console:



### References:

[1]. [Hive on Tez Document](https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez)

[2]. [How to install and deploy Tez?](https://tez.apache.org/install.html)

[3]. [How to install Hive from a Stable Release?](https://cwiki.apache.org/confluence/display/Hive/GettingStarted#GettingStarted-InstallingHivefromaStableRelease)

## [4]. [WebHCat RESTful API](https://cwiki.apache.org/confluence/display/Hive/WebHCat+Reference)

## Spark

Spark is a fast and **general-purpose cluster computing system**. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including **Spark SQL** for SQL and structured data processing, **MLlib** for machine learning, **GraphX** for graph processing, and **Spark Streaming**.

### Spark cluster information

|  |  |  |  |
| --- | --- | --- | --- |
| Server | Address | Cores | Memory |
| Master | spark://10.116.37.181:7077 | - | - |
| Worker01 | 169.172.134.168:54740 | 4 | 1024.0 MB |
| Worker02 | 169.172.134.169:37731 | 4 | 1024.0 MB |
| Worker03 | 10.116.49.45:50755 | 4 | 1024.0 MB |
| Worker04 | 169.172.134.167:34948 | 4 | 1024.0 MB |
| Worker05 | 10.116.37.181:41503 | 4 | 1024.0 MB |
| Worker06 | 10.49.216.44:16580 | 4 | 1024.0 MB |

### Spark configurations

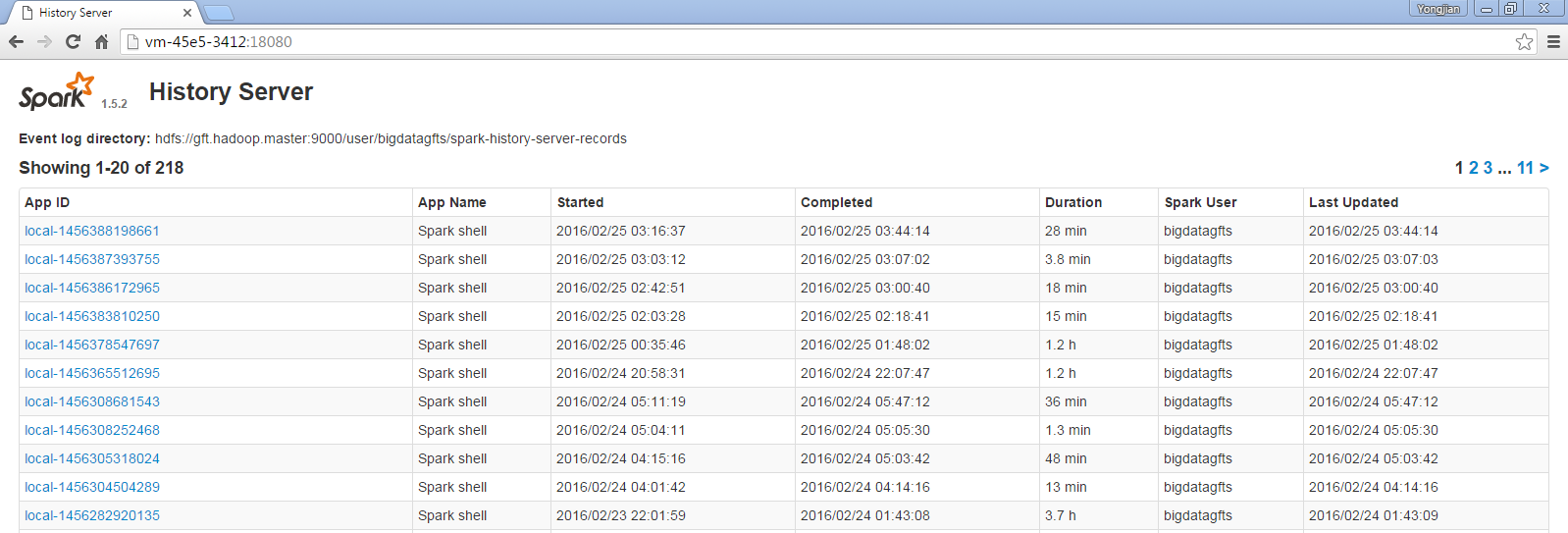
|  |  |
| --- | --- |
| Item | How to set |
| Set spark worker’s memory. | export SPARK\_WORKER\_MEMORY=4g in spark-env.sh |
| Set python 3 for spark. | export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/xenv/python/X/3.4.1l\_64\_RH5/lib  export PATH=$PATH:/xenv/python/X/3.4.1l\_64\_RH5/bin  export PYTHONHASHSEED=0  in ~/.bashrc, don’t forget source it. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Spark-jobserver

Spark-jobserver provides a RESTful interface for submitting and managing Apache Spark jobs, jars, and job contexts. [RESTful API Reference](https://github.com/spark-jobserver/spark-jobserver#api) for more details.

### Spark history server

Spark history server provides a Web UI for completed applications. [Use web browser to access spark history server.](http://vm-45e5-3412:18080/)



### Standalone Mode

Standalone Mode uses Spark standalone cluster.

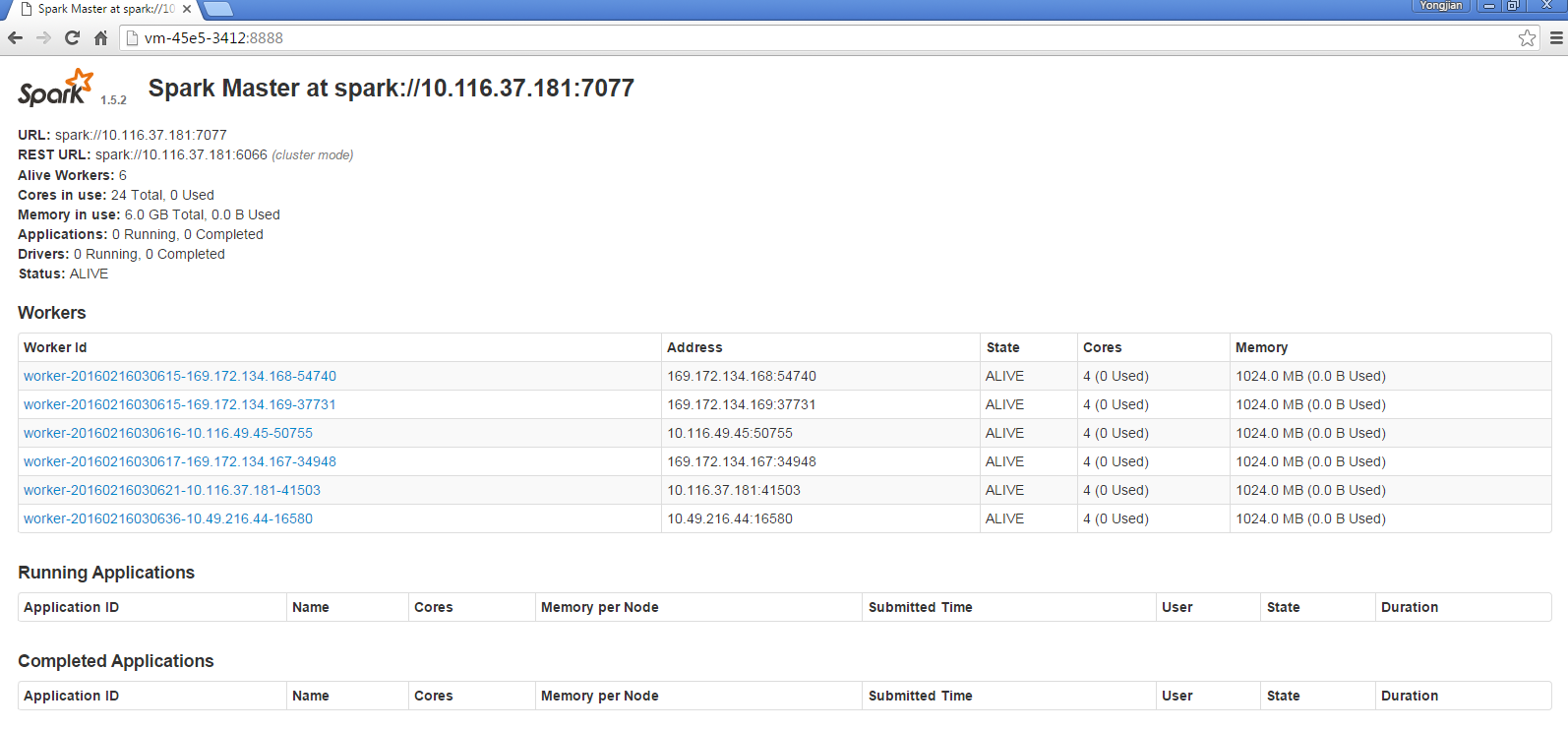
*To run an application on the Spark cluster, simply pass the spark://10.116.37.181:7077 URL of the master as to the SparkContext constructor:*

./bin/spark-submit --master spark://10.116.37.181:7077 <other options and application>

*To run an interactive Spark shell against the cluster, run the following command:*

./bin/spark-shell --master spark://10.116.37.181:7077

[Use web browser to access Standalone Mode Web UI](http://vm-45e5-3412:8888/)



### Yarn-client Mode

Yarn-client Mode uses Yarn as cluster Manager, Spark Driver run in Spark Application Master.

*To launch a Spark application in client mode:*

$ ./bin/spark-submit --class path.to.your.Class --master yarn --deploy-mode client [options] <app jar> [app options]

### Yarn-cluster Mode

Yarn-cluster Mode uses Yarn as cluster Manager, Spark Driver run in Client.

*To launch a Spark application in cluster mode:*

$ ./bin/spark-submit --class path.to.your.Class --master yarn --deploy-mode cluster [options] <app jar> [app options]

*To run spark-shell in client mode:*

$ ./bin/spark-shell --master yarn --deploy-mode client

### How to use java to connect to Spark cluster

[run spark remotely through eclipse. Guide](file:///\\Shavnascsts0001\grp_icggfts1\CIBTech%20II\GFTSFinance\HadoopWork\run%20spark%20%20remotely%20through%20eclipse.%20Guide.docx)

## References:

[1]. [Spark Document](http://spark.apache.org/docs/latest/index.html)

## ZooKeeper

Apache ZooKeeper is an effort to develop and maintain an open-source server which enables highly reliable distributed coordination.

### Zookeeper install

1. get the zookeeper zip file and unzip into the server you want to install on

2.add one record in the the {zk\_home}/zookeeper-XXXX/conf/zoo.cfg like server.{number}={ip}:{communicate\_port}:{election\_port}<:observer>

3.scp the zoo.cfg across all the zookeeper server

4.put the same number in the {zk\_home}/data/myid

5.restart all the zookeepers

### Zookeeper management

|  |  |  |
| --- | --- | --- |
| function | command | Alias in current servers |
| Check status | {zk\_home}/ zookeeper-XXXX/bin/zkServer.sh status | zkServer status |
| Start Server | {zk\_home}/ zookeeper-XXXX/bin/zkServer.sh start | zkStart |
| Stop server | {zk\_home}/ zookeeper-XXXX/bin/zkServer.sh start | zkStop |
| client | {zk\_home}/ zookeeper-XXXX/bin/zkCli.sh –server {ip:port} | zkCli |

# Compare

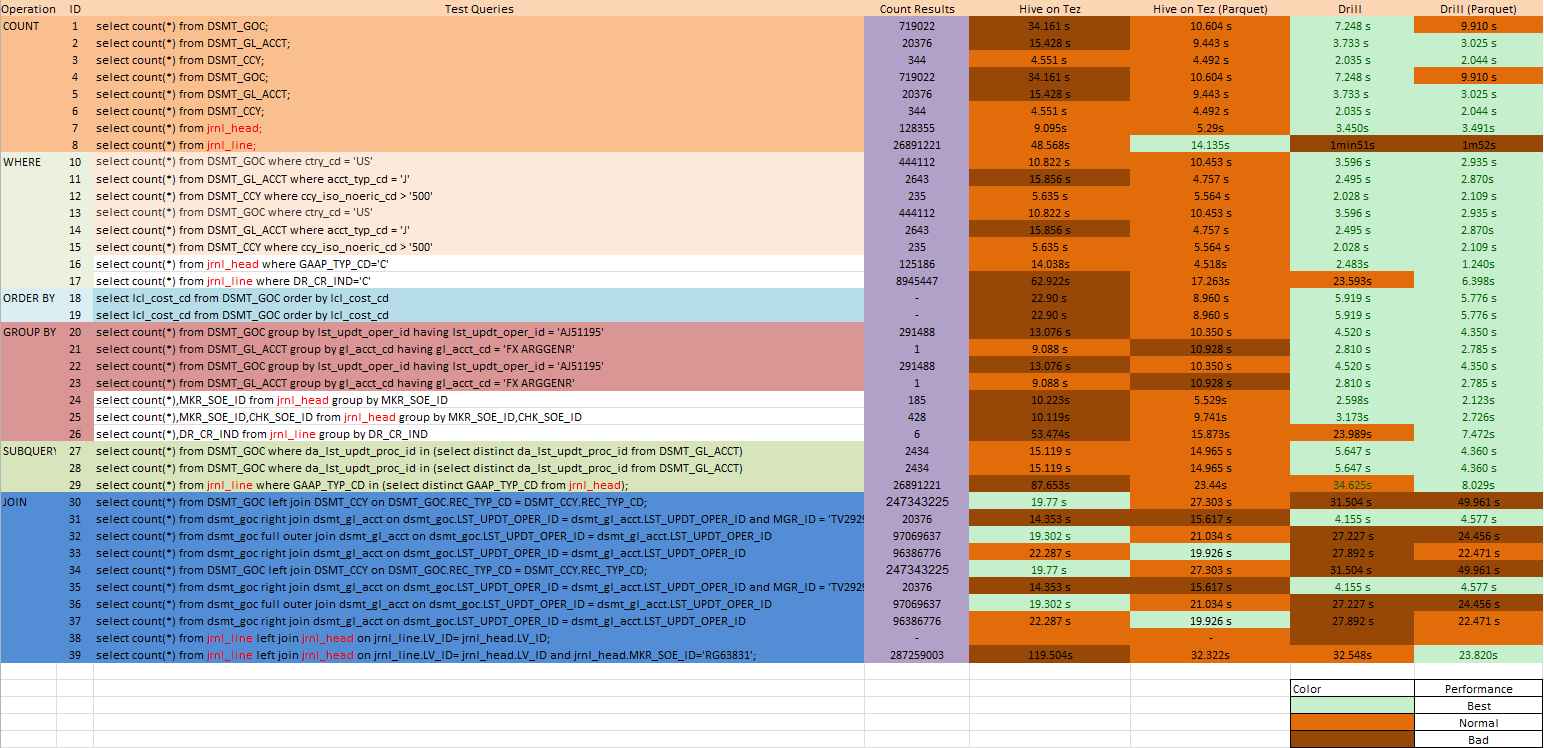
## Cluster Computing Framework: MapReduce, Tez and Spark

For more information:



## SQL-on-Hadoop: Hive and Drill

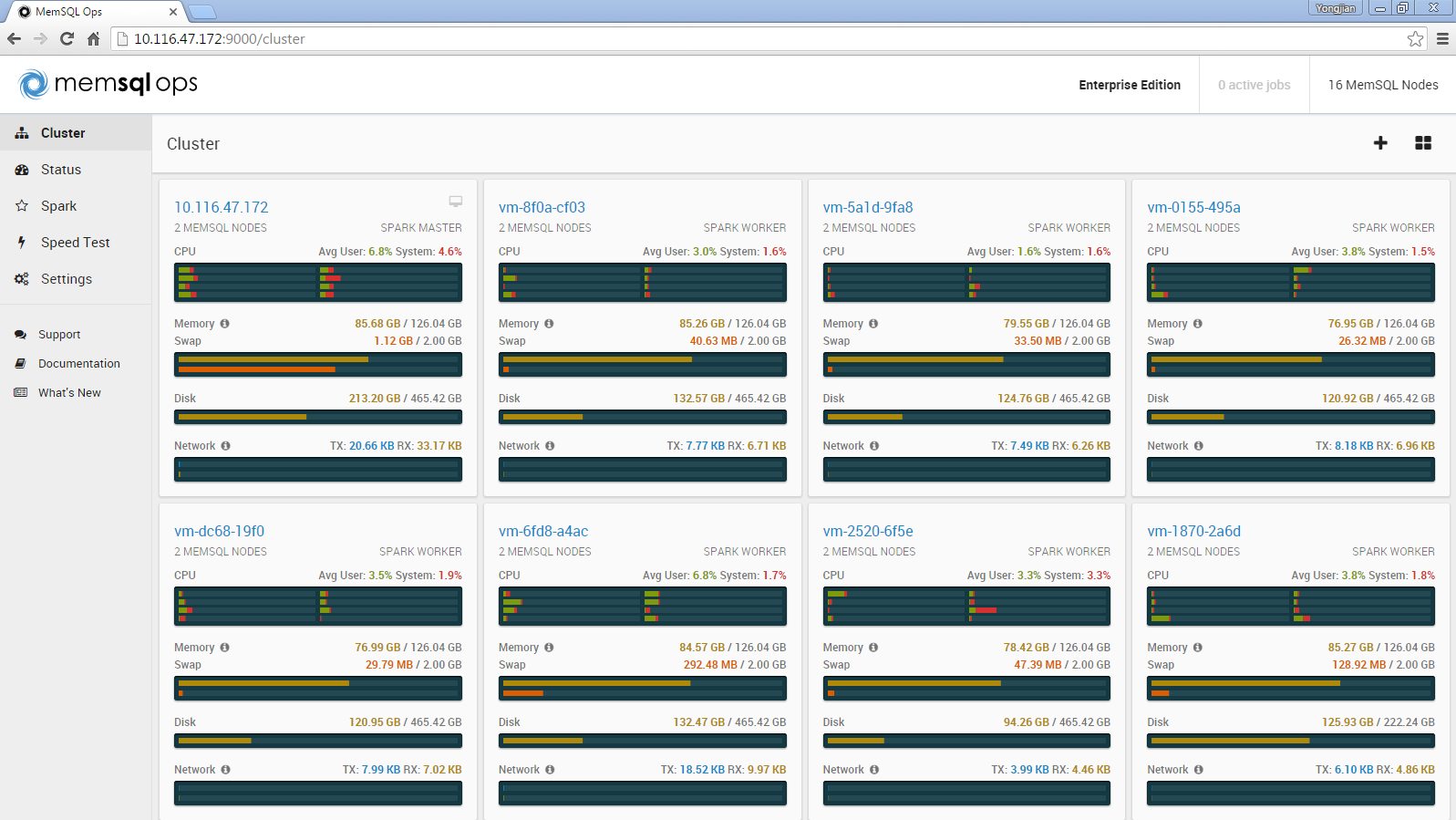
### Test Result



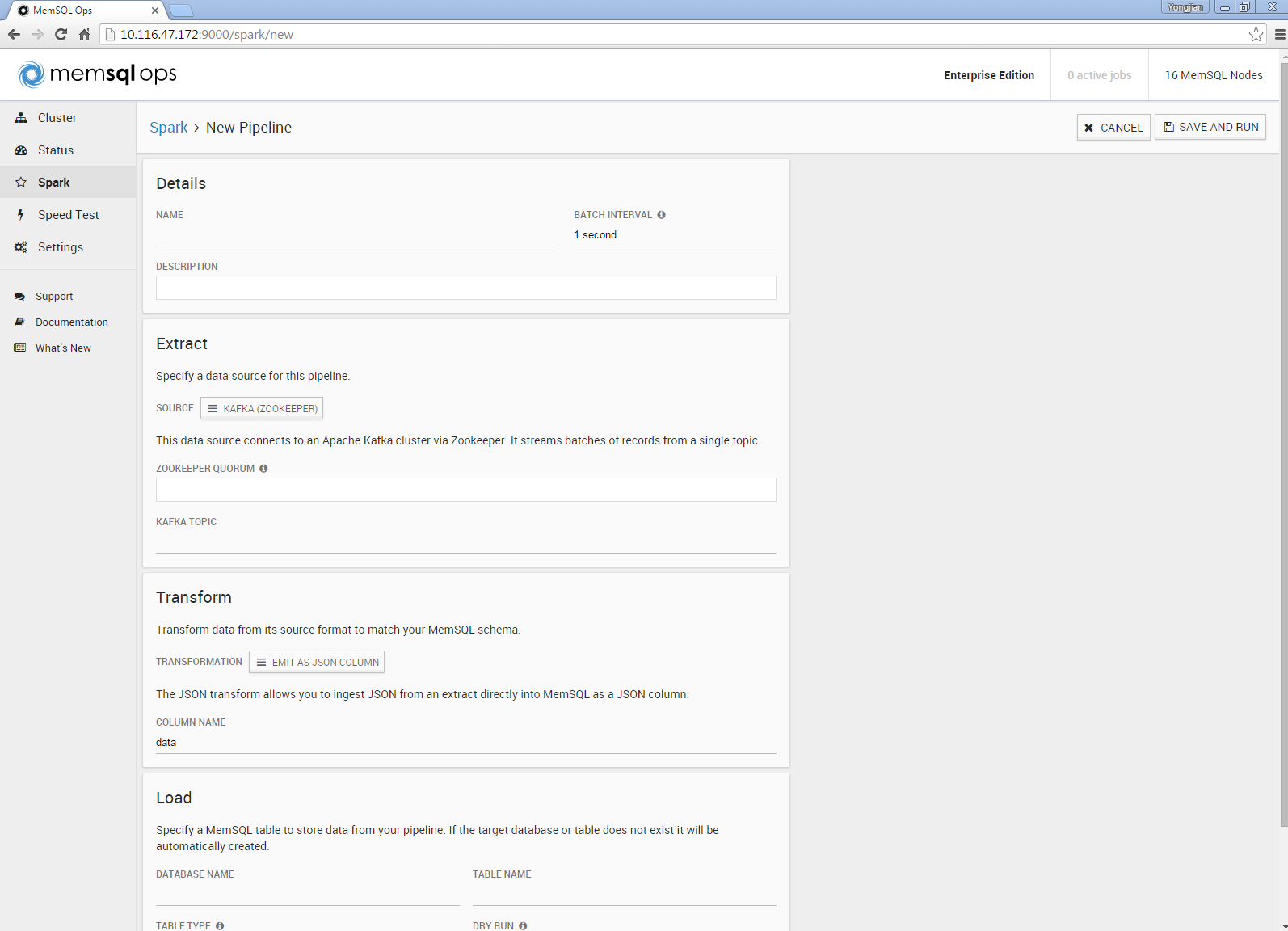
# MemSQL

MemSQL is a high-performance, in-memory database that combines the horizontal scalability of distributed systems with the familiarity of SQL. MemSQL Streamliner allows users to set up real-time data pipelines. It loads data into MemSQL, but first extracts and transforms the data within Apache Spark.

## [Use web browser to access MemSQL cluster](http://10.116.47.172:9000/cluster)



## [ETL on MemSQL with Spark](http://10.116.47.172:9000/spark/new)



# Kafka

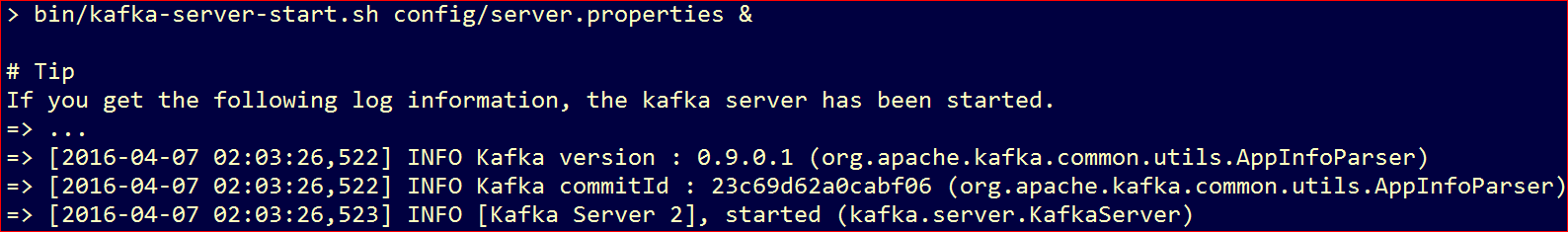
## Kafka server information

|  |  |
| --- | --- |
| Zookeeper Server ID | Address |
| 0 | 169.172.134.168:2182 |
| 1 | 169.172.134.167:2182 |
| 2 | 169.172.134.169:2182 |

|  |  |
| --- | --- |
| Kafka Broker | Address |
| broker 0 | 169.172.134.169:9092 |
| broker 1 | 169.172.134.168:9092 |
| broker 2 | 169.172.134.167:9092 |

## Kafka quick start in our cluster

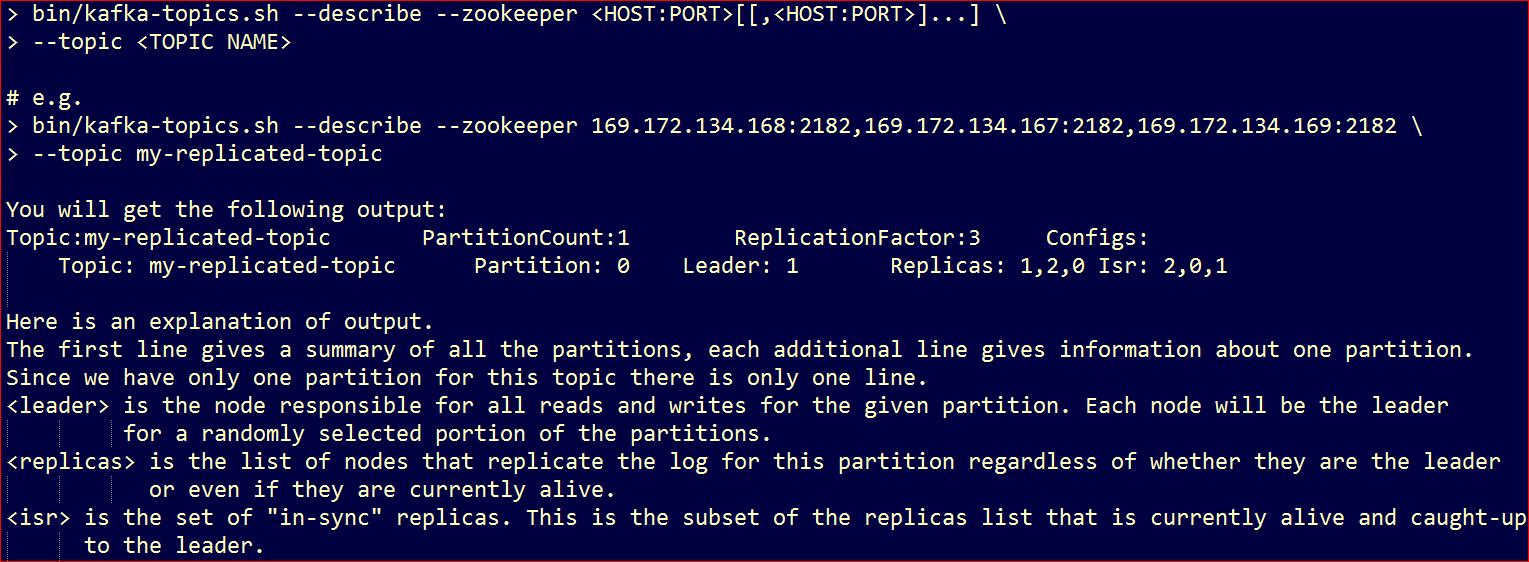
### 1. Start Kafka server



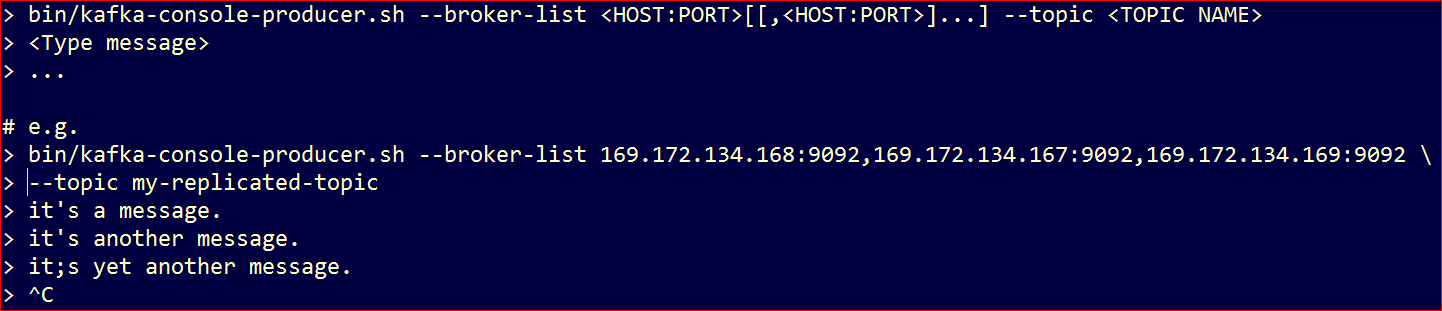
### 2. Create a topic

### 

### 3. Describe a topic



### 4. Start Kafka console producer



### 5. Start Kafka console consumer

