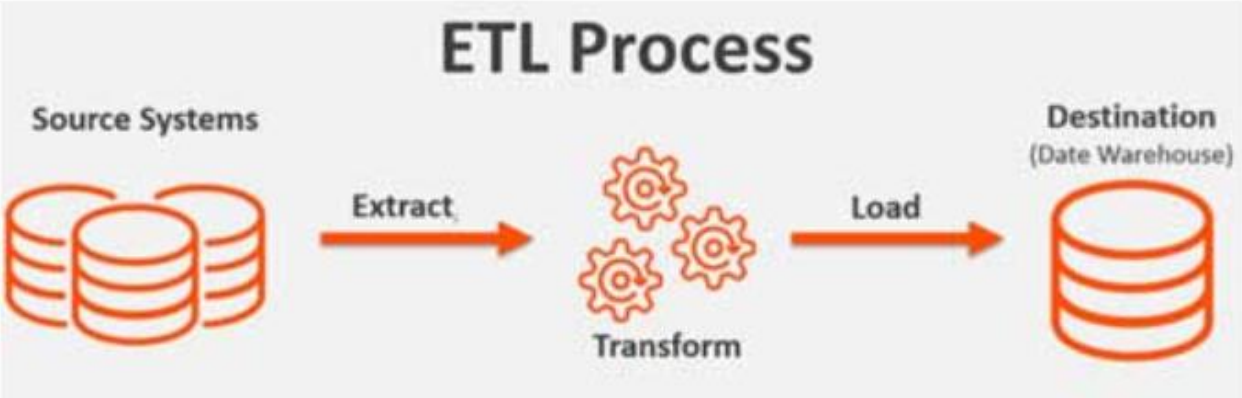
Data Integration

빅데이터

ETL



핵심은 데이터 품질 :

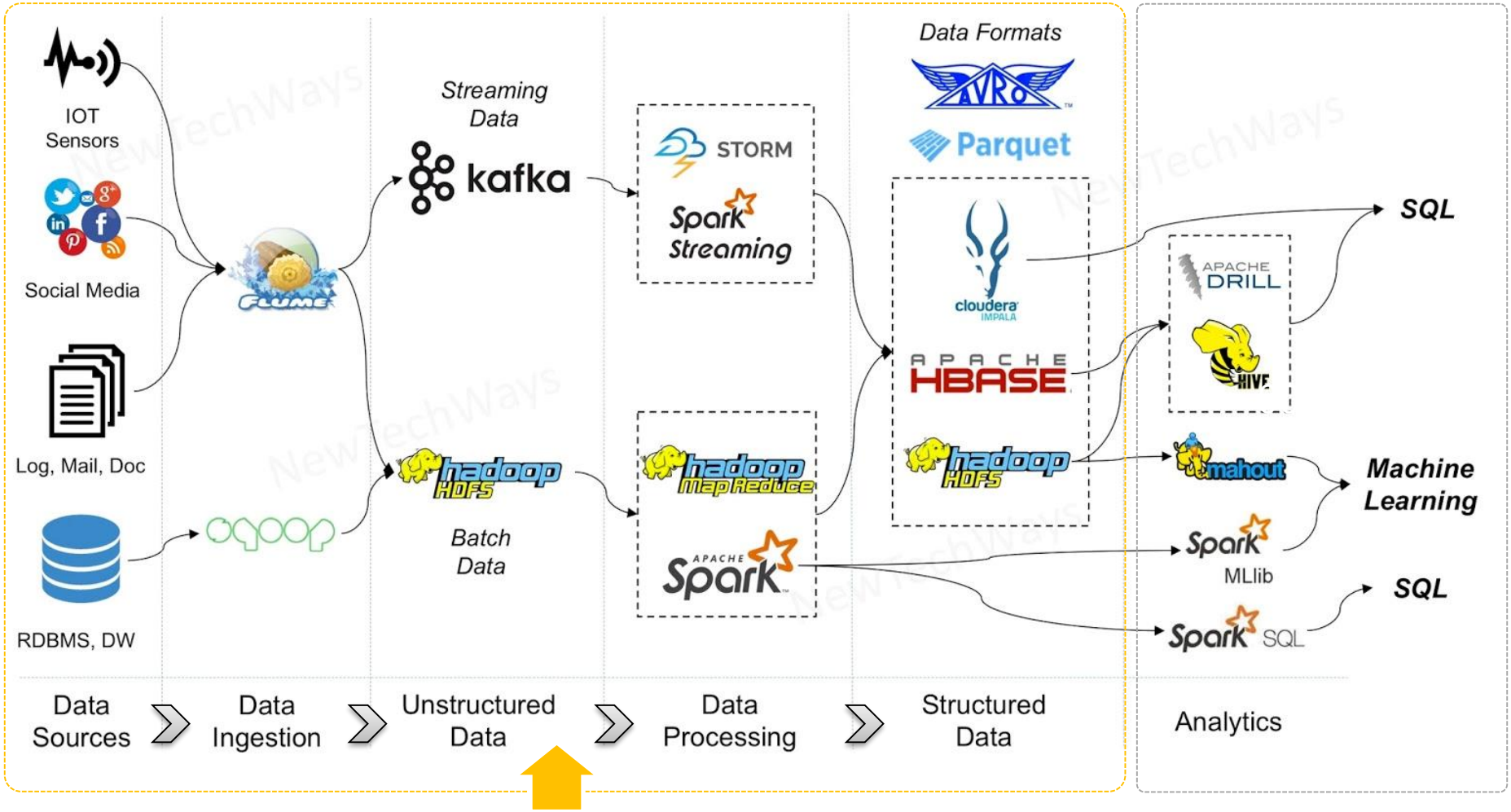
데이터 사이언티스트들이 작업시간의 80% 를 데이터 추출 및 정제 작업에 투자
원천 데이터를 단순히 수집하여 바로 데이터 시스템에 적재해서는 안됨

ETL :

가장 보편적이고 검증된 배치 처리 기술

빅데이터

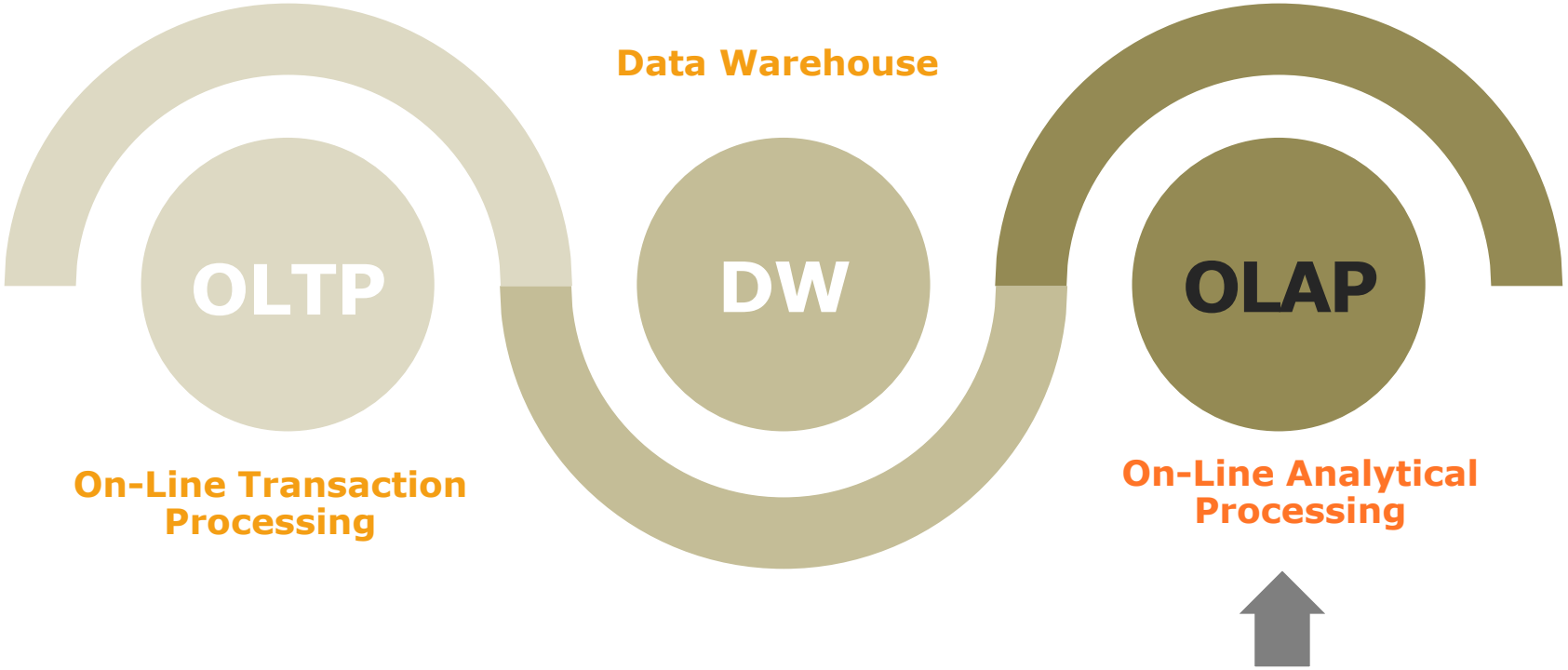
Data Pipeline



Druid

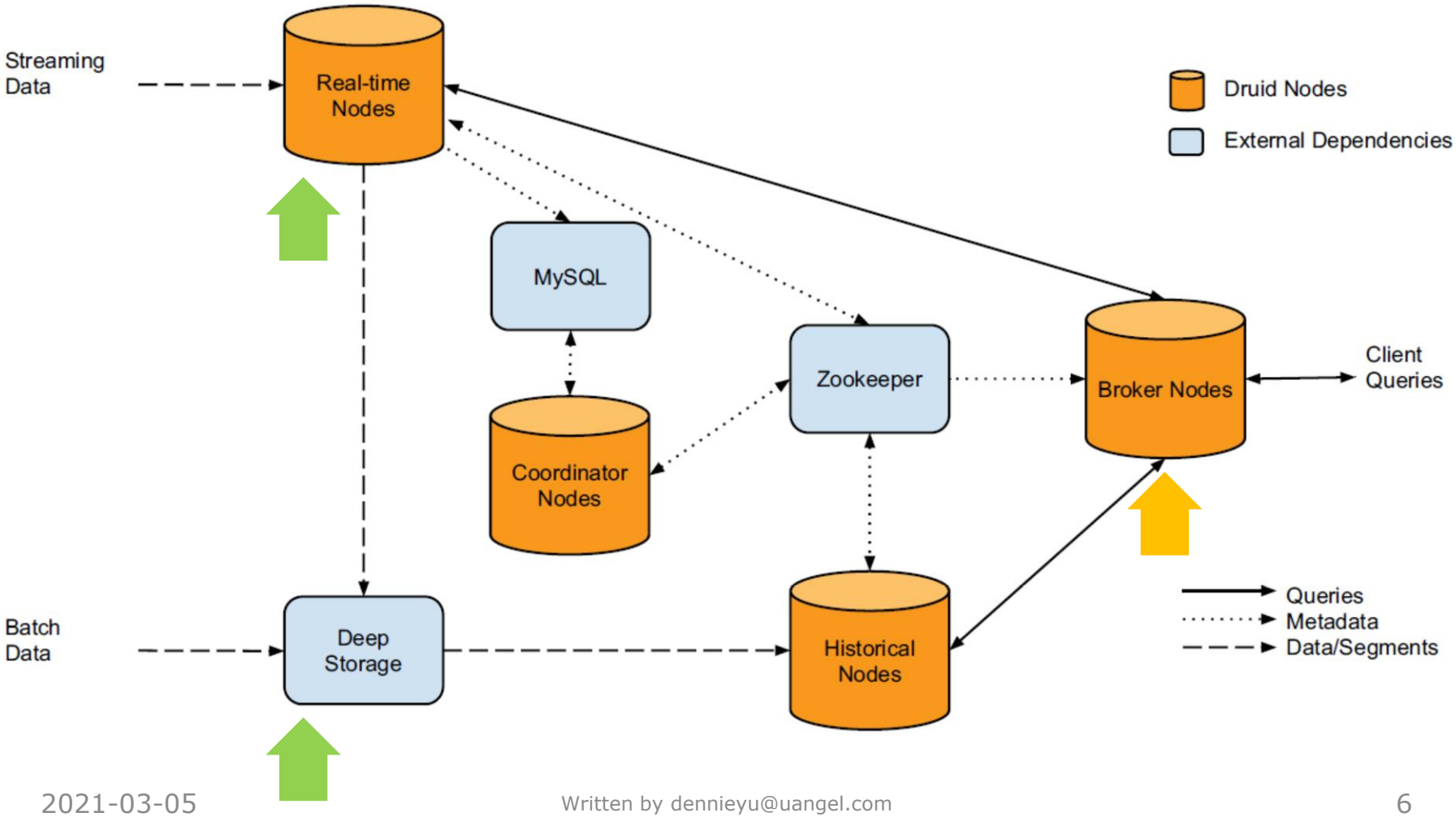
OLAP

druid


Data Warehouse
OLTP
DW
OLAP
On-Line Transaction Processing
On-Line Analytical Processing

Architecture

druid



Overview

druid

Historical Node	***Broker Node***	***Coordinator Node***	***Realtime Node***	***External Dependencies***
• *Deep Storage* 에서 세그먼트 다운로드 • 세그먼트에 대한 *Broker Node* 의 쿼리 실행 후 결과 반환 • *Zookeeper* 를 통해 세그먼트 로드	• 쿼리 인입점 • 쿼리 실행, 결과 수집, 병합을 담당 • *Zookeeper* 를 통해 *Realtime Node*, *Historical Node*가 어떤 것인지 판단	• 세그먼트 관리 • *Historical Node* 에 지시하기 위해 *Zookeeper* 를 사용	• 실시간 데이터 수집, 데이터 인덱싱 및 세그먼트를 *Historical Node* 에 전달	• *Zookeeper* - 클러스터 내부에서의 **서비스 디스커버리** 및 데이터 토폴리지 운영 • *Metadata storage instance* - 세그먼트에 대한 **메타데이터 저장소** • *Deep Storage / File System* - **세그먼트 저장**

druid

배경

빅데이터

PROBLEM DEFINITION

- 리얼 타임이어야 하고
- 멀티 테넌시 하며
- 컬럼 지향
- 쿼리 속도 보장

Hadoop.. 위 **4**가지를 만족시킬 수 없다.

Druid의 메타데이터는 **3**가지 요소로 구성되어 있다.
- **Timestamp**가 있다.
- **Dimension** 컬럼이 많다. 데이터 속성이 많다는 것.
- **Metric** 컬럼이 많다. 수집하고자 하는 지표가 많다는 것.

이러한 **Metric**은 수집되고 **groupby**되고 **aggregate**되는 대상이다.

druid

정의

빅데이터

OLAP(On-Line Analytical Processing) 즉, 온라인 분석 처리는 다차원 데이터 구조를 이용하여 다차원의 복잡한 질의를 고속으로 처리하는 데이터 분석 기술이다. 기업의 분석가, 관리자 및 임원들은 **OLAP** 기술을 통해 필요한 정보에 대해 대화형으로 빠르게 접근 가능하다.

“Druid is NOT time series DB”

즉, **Druid**는 원본을 저장하지 않으며 기존 데이터에서 **OLAP**질의가 가능하도록 **indexing**하는 구조를 가지고 있다.

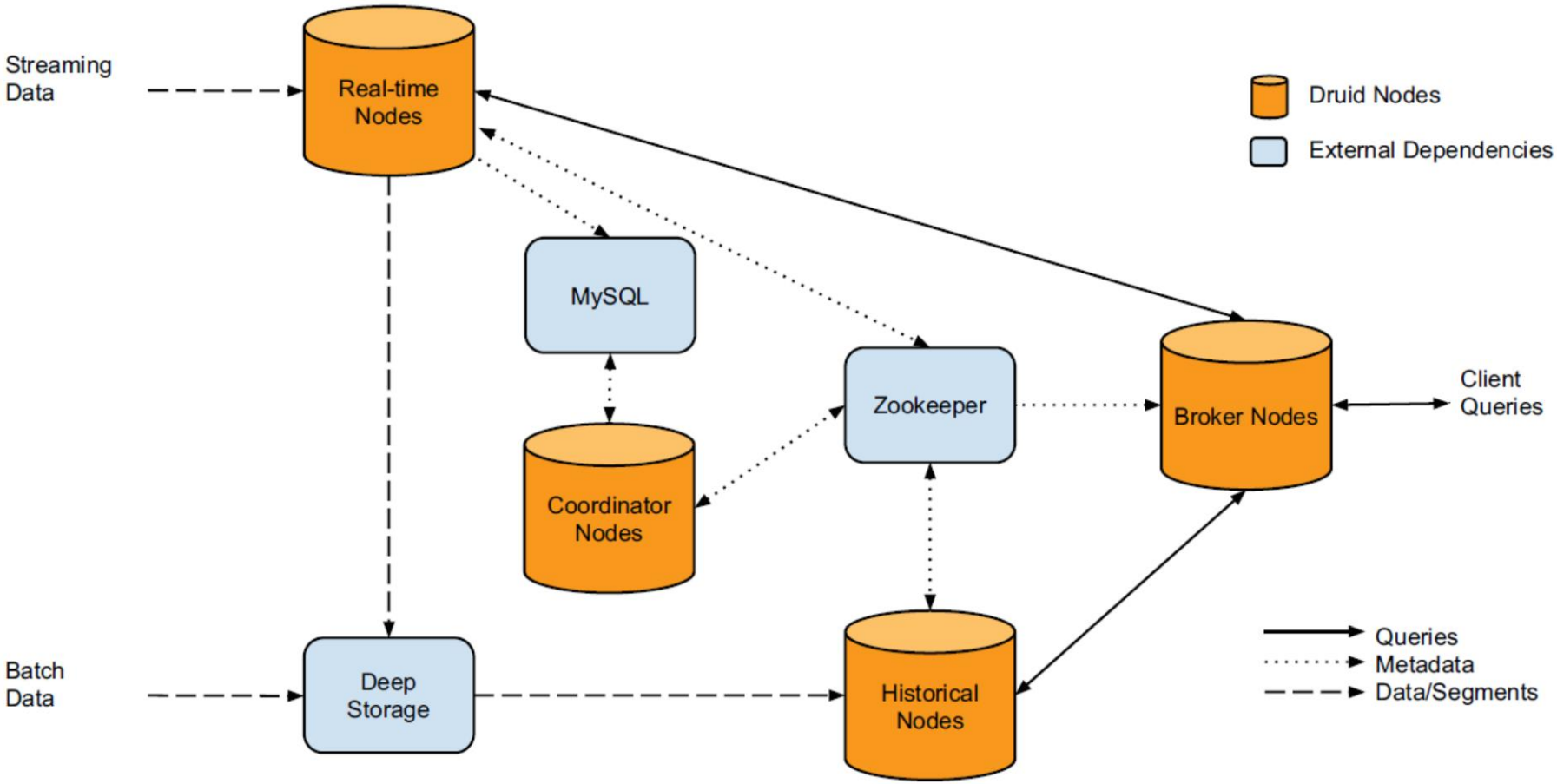
같은 시간 범위 내에 존재하는 데이터를 분할해 저장하는 점은 다른 시계열 **DB (Machbase, OpenTSDB, ..)**에서도 취하는 형태이다.

druid

아키텍처

Diagram

druid



Diagram

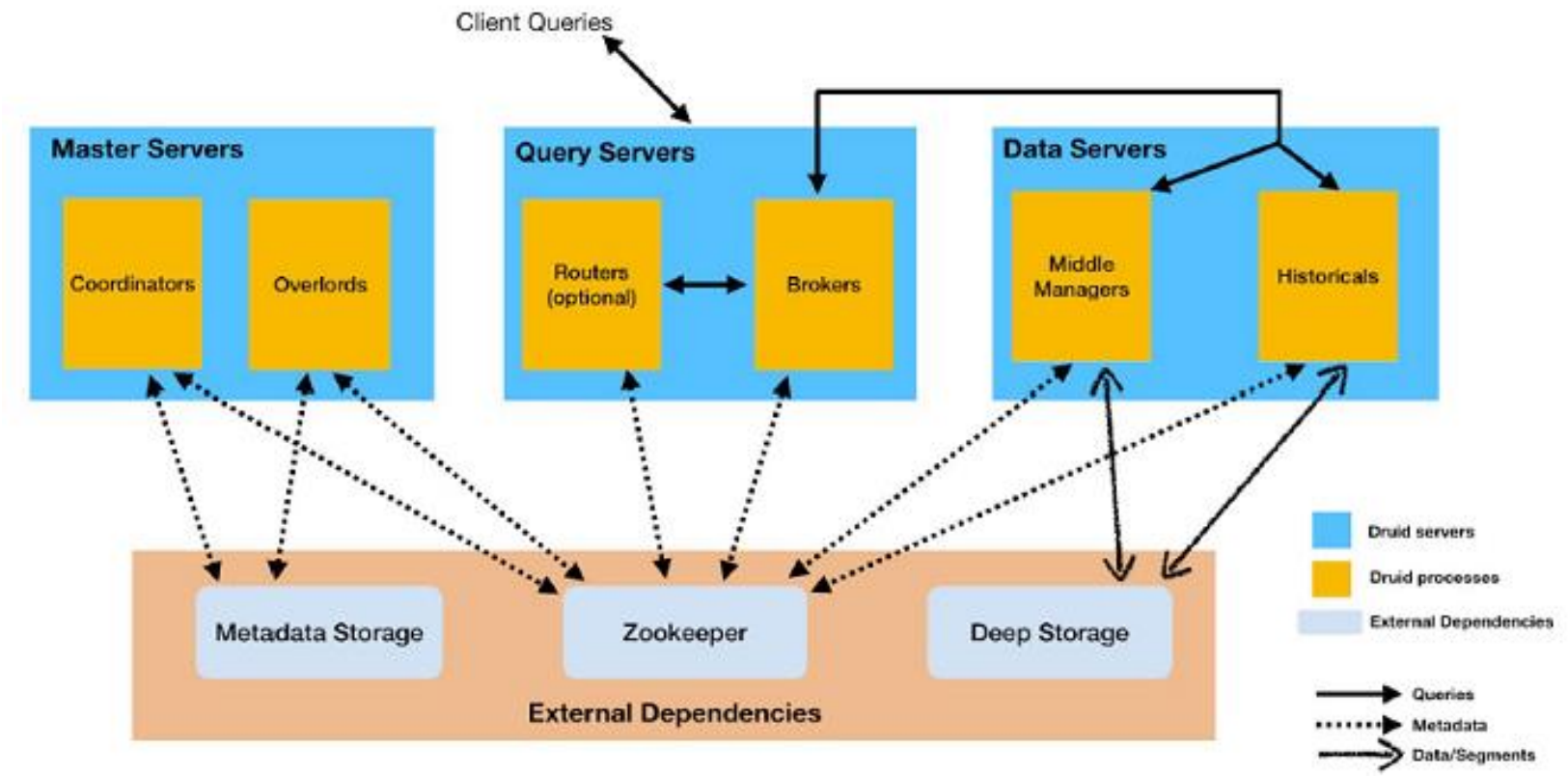
- **Druid** 는 실시간 아키텍처와 배치 아키텍처를 모두 수용한다. 원본 데이터가 실시간인 경우 **Real-time node**에서 색인되며 **segment(time 설정 기준)**으로 **Deep Storage**로 내리게 된다. 배치 데이터는 **Deep Storage**에 저장되며 **Deep Storage**는 **HDFS, S3**등으로 구성될 수 있다.

- 사용자 쿼리는 **Broker node**가 담당한다. **Segment** 정보를 생성하지 않은 **fresh**한 데이터는 **Real-time node**에서 그리고 배치로 색인된 데이터는 **Historical node**에서 가져오게 된다. (람다아키텍처) **Druid** 내부 컴포넌트간 통신은 **zookeeper**로 되어있다.

- **Druid** 관련 메타정보는 **mysql**등 **RDBMS**에 저장할 수있다. **Coordinator**는 메타정보를 통해 **segment life cycle**(복제 갯수 조정 및 **Load/Drop**)을 관리한다.

Diagram

druid



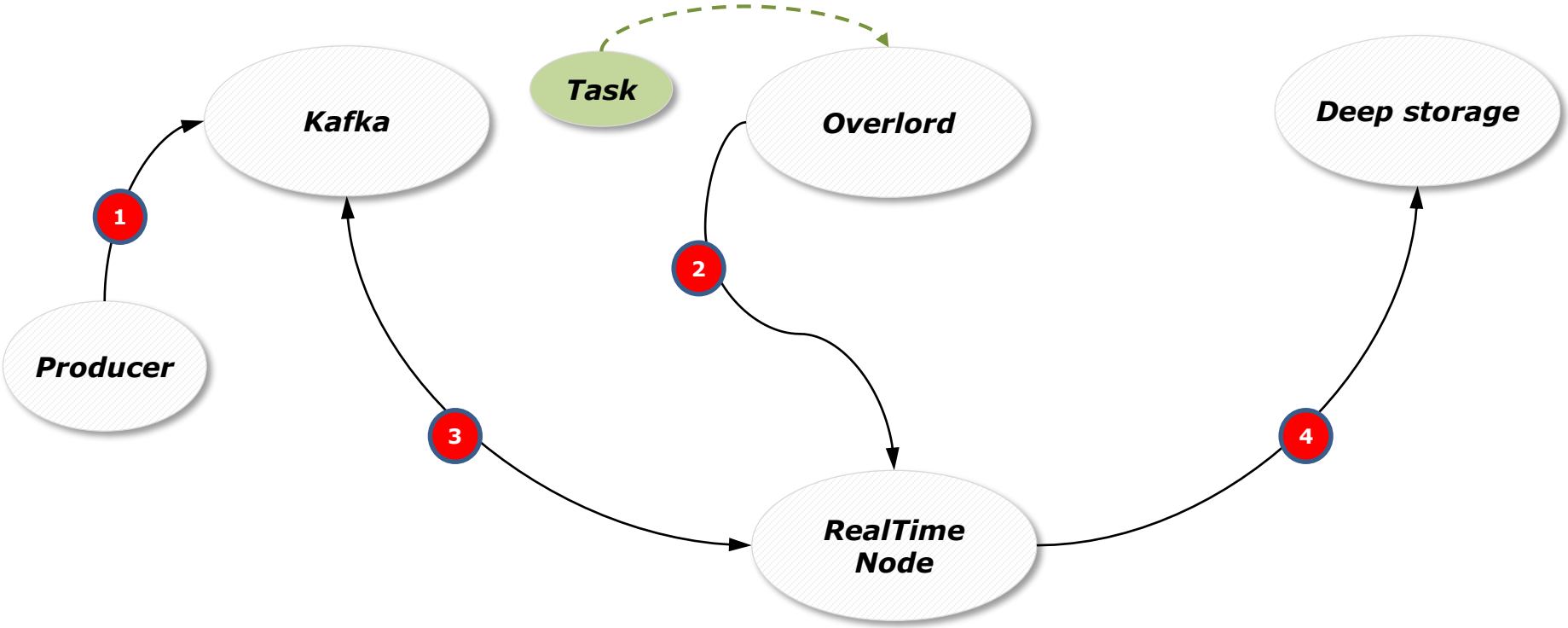
druid

인덱싱

Realtime Ingestion

druid

*http://{**OVERLORD_IP**}:{port}/druid/indexer/v1/supervisor*



Realtime Ingestion

*http://{**OVERLORD_IP**}:{port}/druid/indexer/v1/supervisor*

```
-   {
-     "type": "kafka",
-     "dataSchema": {
-       "dataSource": "___cpoak",
-       "parser": {
-         "type": "string",
-         "parseSpec": {
-           "format": "json",
-           ...
-         }
-       },
-      ...
-     },
-     "tuningConfig": {
-       "type": "kafka",
-       "maxRowsPerSegment": 5000000
-     },
-     "ioConfig": {
-       "type": "kafka",
-       "topic": "example1",
-       "consumerProperties": {
-         "bootstrap.servers": "192.168.7.35:9092"
-       },
-       "taskCount": 1,
-       "replicas": 1,
-       "taskDuration": "PT1H"
-     }
-   }
```

Batch Ingestion

- **Druid는 실시간 처리 외에도 정적 파일 (HDFS/S3)에 대한 ingestion도 지원한다.**

- **HDFS에서 ingestion을 하는 경우 hadoop MR job 으로 수행되며 앞서 언급한 overlord node에 ingestion job을 submit하면 된다.**

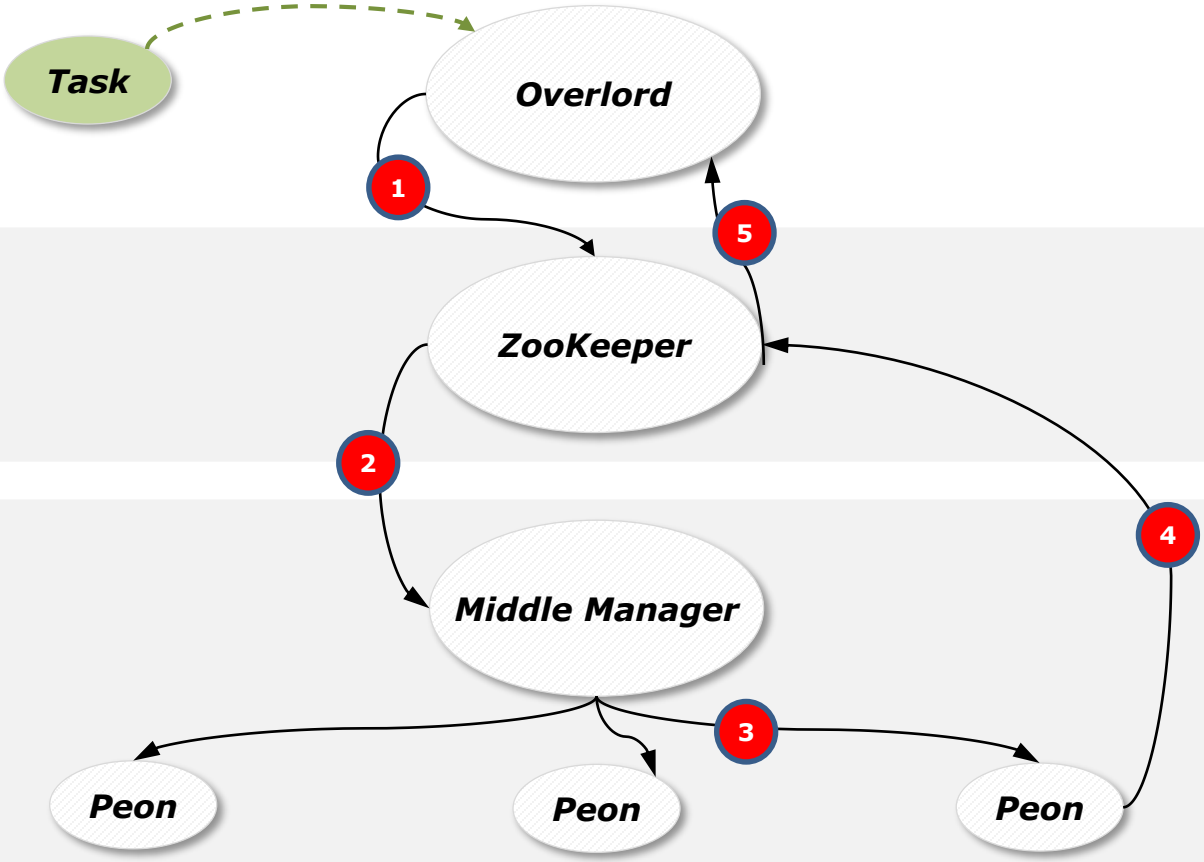
- **batch ingestion의 경우 type은 index-hadoop이다. 실시간과 다른점은 batch의 경우 intervals를 꼭 지정해주어야 한다.**

- **이 후 MR job이 수행 된다. (*1st MR* -determinePartitions, *2nd MR* - IndexGeneration)**

Batch Ingestion

druid

*http://{**OVERLORD_IP**}:{port}/druid/indexer/v1/task*



Batch Ingestion

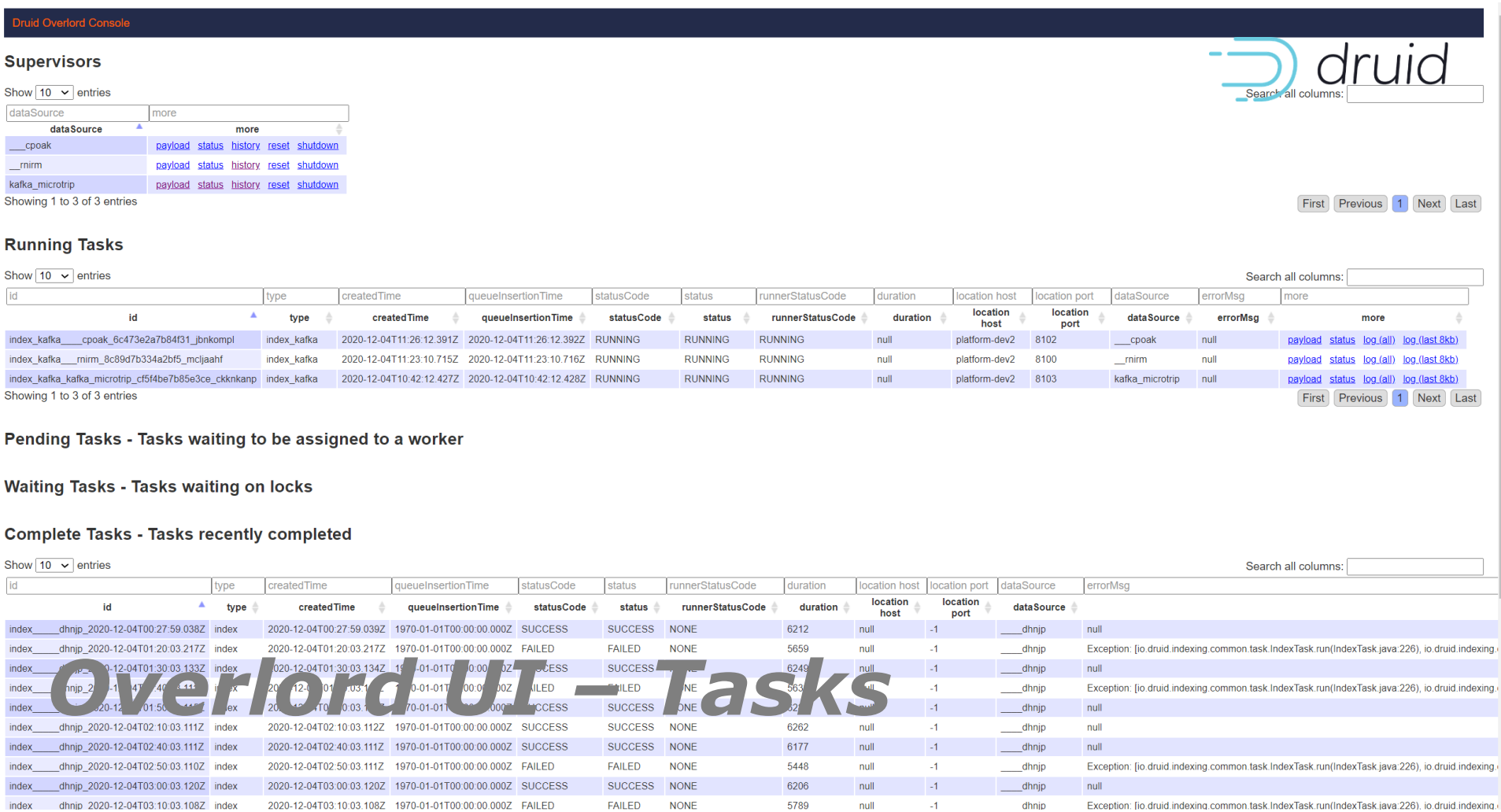
*http://{**OVERLORD_IP**}:{port}/druid/indexer/v1/task*

```
{
  "type": "index",
  "spec": {
    "dataSchema": {
      "dataSource": "____dhnjp",
      "parser": {
        "type": "csv.stream",
        "timestampSpec": {
          "column": "create_time",
          ...
        },
        ...
      },
      "ioConfig": {
        "type": "index",
        "firehose": {
          "type": "local",
          "filter": "____dhnjp_1970-01-01T00:00:00.000Z.csv",
          "baseDir": "/tmp"
        }
      },
      "tuningConfig": {
        "type": "index",
        "maxRowsInMemory": 75000,
        "ignoreInvalidRows": true,
        "buildV9Directly": true
      }
    },
    "context": ...
}
```

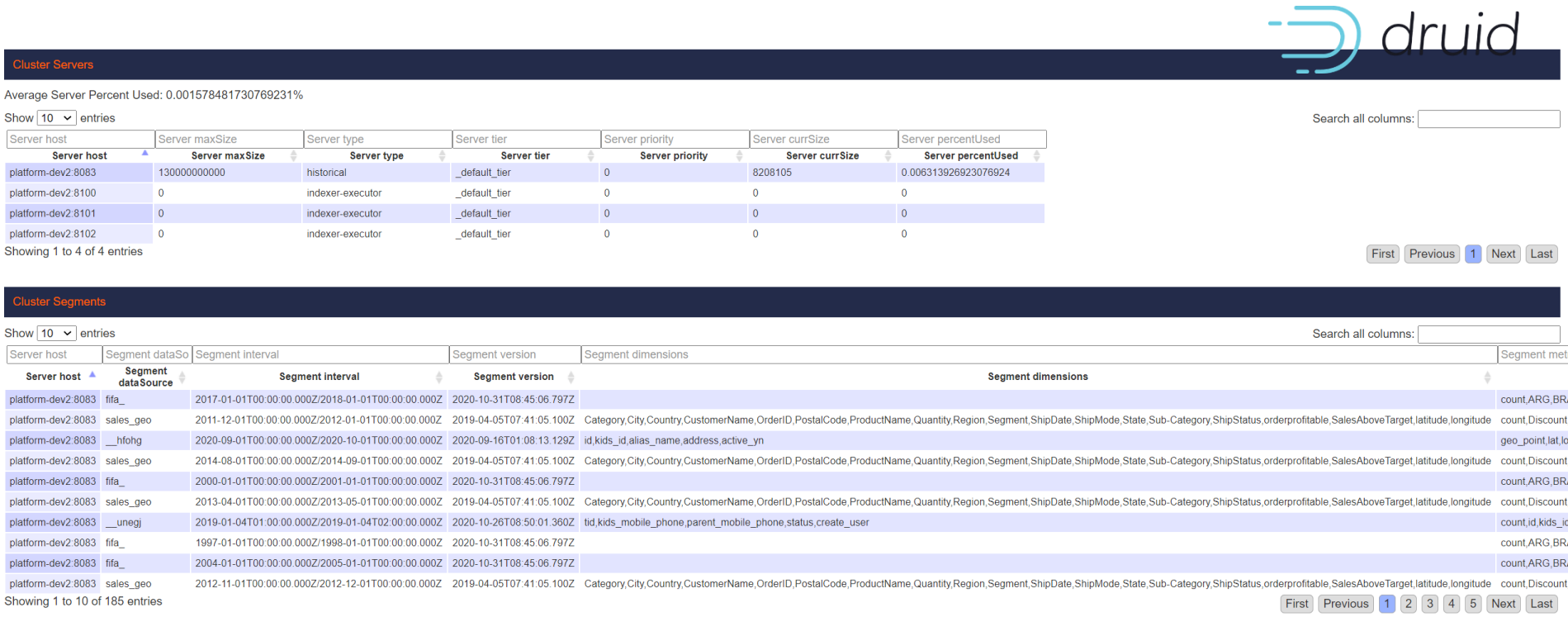
druid

UI

빅데이터



Overlord UI – Tasks



druid

Cluster Servers

Average Server Percent Used: 0.001578481730769231%

Show 10 entries

Search all columns:

Server host	Server maxSize	Server type	Server tier	Server priority	Server currSize	Server percentUsed
platform-dev2:8083	130000000000	historical	_default_tier	0	8208105	0.006313926923076924
platform-dev2:8100	0	indexer-executor	_default_tier	0	0	0
platform-dev2:8101	0	indexer-executor	_default_tier	0	0	0
platform-dev2:8102	0	indexer-executor	_default_tier	0	0	0

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Cluster Segments

Show 10 entries

Search all columns:

Server host	Segment dataSource	Segment interval	Segment version	Segment dimensions	Segment metric
platform-dev2:8083	fifa_	2017-01-01T00:00:00.000Z/2018-01-01T00:00:00.000Z	2020-10-31T08:45:06.797Z		count,ARG,BRA
platform-dev2:8083	sales_geo	2011-12-01T00:00:00.000Z/2012-01-01T00:00:00.000Z	2019-04-05T07:41:05.100Z	Category,City,Country,CustomerName,OrderID,PostalCode,ProductName,Quantity,Region,Segment,ShipDate,ShipMode,State,Sub-Category,ShipStatus,orderprofitable,SalesAboveTarget,latitude,longitude	count,Discount,F
platform-dev2:8083	__hfohg	2020-09-01T00:00:00.000Z/2020-10-01T00:00:00.000Z	2020-09-16T01:08:13.129Z	id,kids_id,alias_name,address,active_yn	geo_point,lat,lon
platform-dev2:8083	sales_geo	2014-08-01T00:00:00.000Z/2014-09-01T00:00:00.000Z	2019-04-05T07:41:05.100Z	Category,City,Country,CustomerName,OrderID,PostalCode,ProductName,Quantity,Region,Segment,ShipDate,ShipMode,State,Sub-Category,ShipStatus,orderprofitable,SalesAboveTarget,latitude,longitude	count,Discount,F
platform-dev2:8083	fifa_	2000-01-01T00:00:00.000Z/2001-01-01T00:00:00.000Z	2020-10-31T08:45:06.797Z		count,ARG,BRA
platform-dev2:8083	sales_geo	2013-04-01T00:00:00.000Z/2013-05-01T00:00:00.000Z	2019-04-05T07:41:05.100Z	Category,City,Country,CustomerName,OrderID,PostalCode,ProductName,Quantity,Region,Segment,ShipDate,ShipMode,State,Sub-Category,ShipStatus,orderprofitable,SalesAboveTarget,latitude,longitude	count,Discount,F
platform-dev2:8083	__unegj	2019-01-04T01:00:00.000Z/2019-01-04T02:00:00.000Z	2020-10-26T08:50:01.360Z	tid,kids_mobile_phone,parent_mobile_phone,status,create_user	count,id,kids_id
platform-dev2:8083	fifa_	1997-01-01T00:00:00.000Z/1998-01-01T00:00:00.000Z	2020-10-31T08:45:06.797Z		count,ARG,BRA
platform-dev2:8083	fifa_	2004-01-01T00:00:00.000Z/2005-01-01T00:00:00.000Z	2020-10-31T08:45:06.797Z		count,ARG,BRA
platform-dev2:8083	sales_geo	2012-11-01T00:00:00.000Z/2012-12-01T00:00:00.000Z	2019-04-05T07:41:05.100Z	Category,City,Country,CustomerName,OrderID,PostalCode,ProductName,Quantity,Region,Segment,ShipDate,ShipMode,State,Sub-Category,ShipStatus,orderprofitable,SalesAboveTarget,latitude,longitude	count,Discount,F

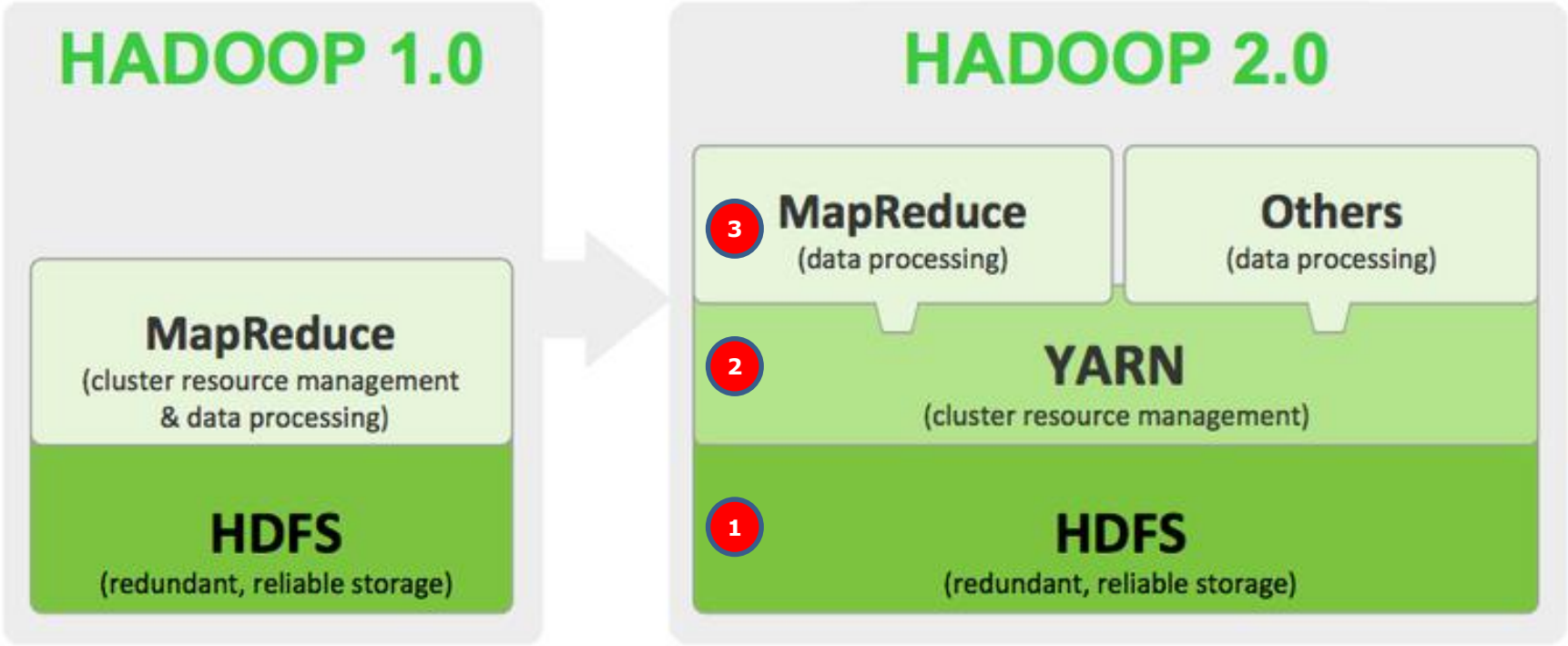
Showing 1 to 10 of 185 entries

First Previous 1 2 3 4 5 Next Last

Coordinater UI – Cluster Servers

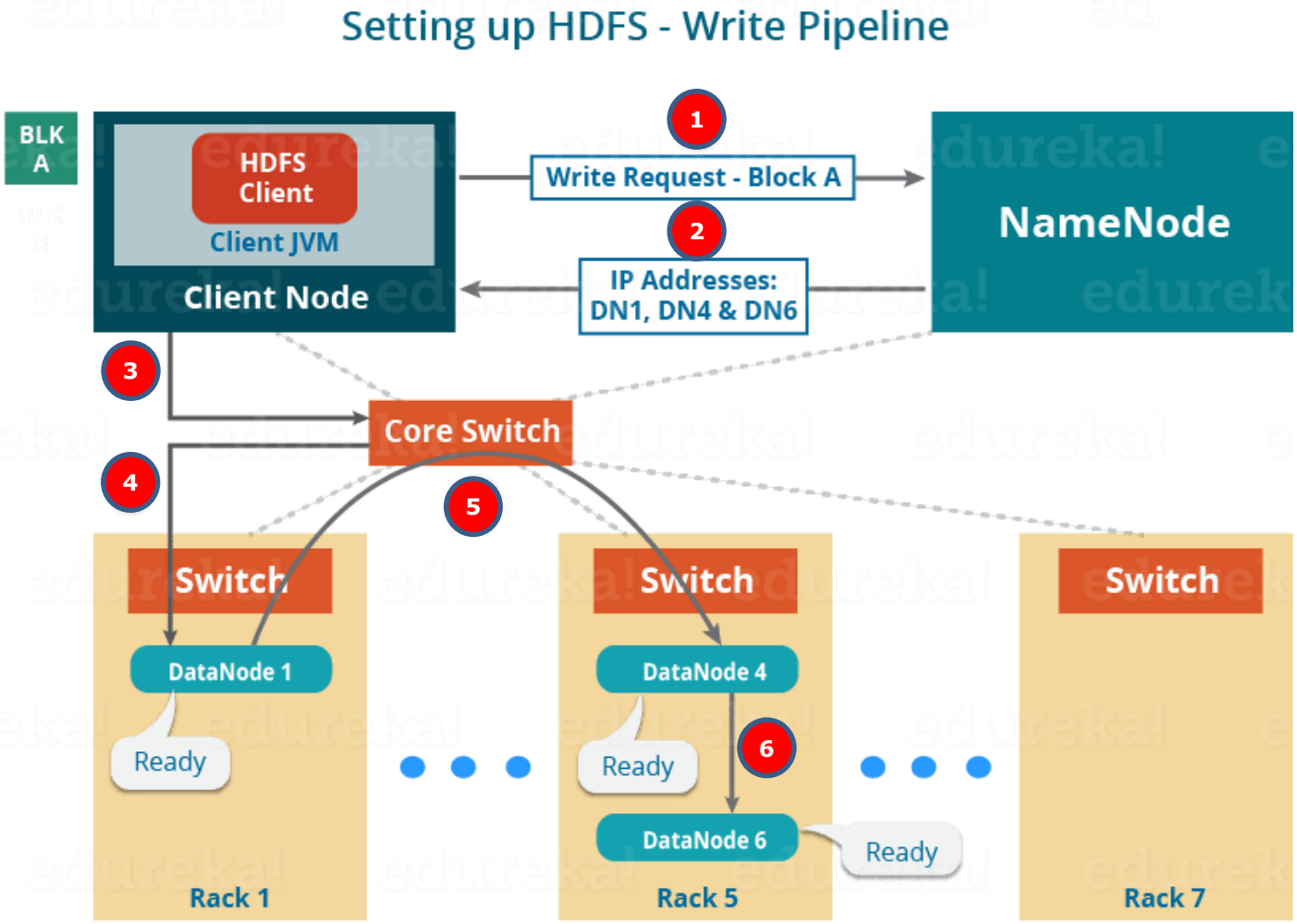
하둡

Architecture

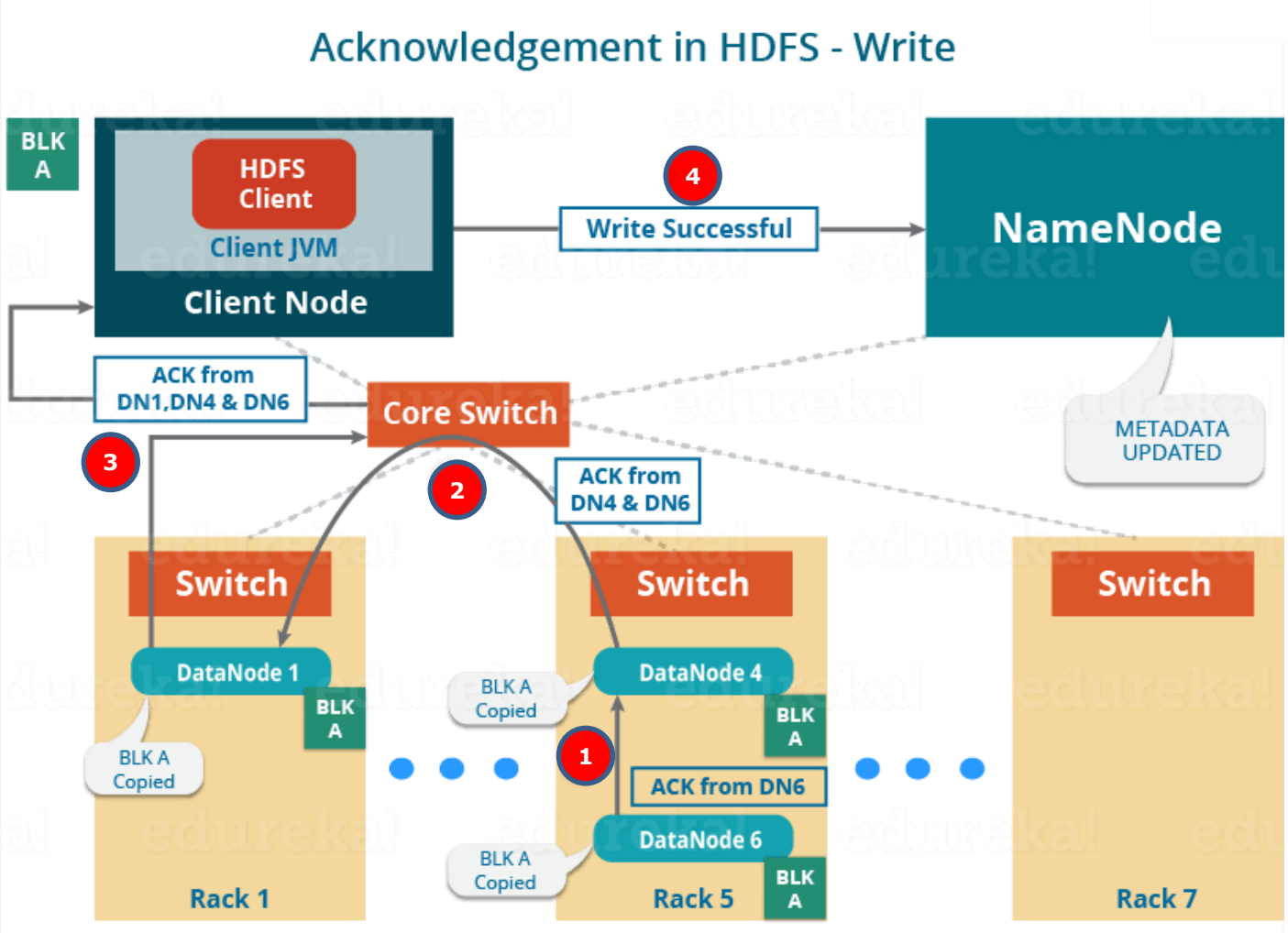


※ 현재 최신버전은 v3 이고, Erasure Coding 을 지원함.

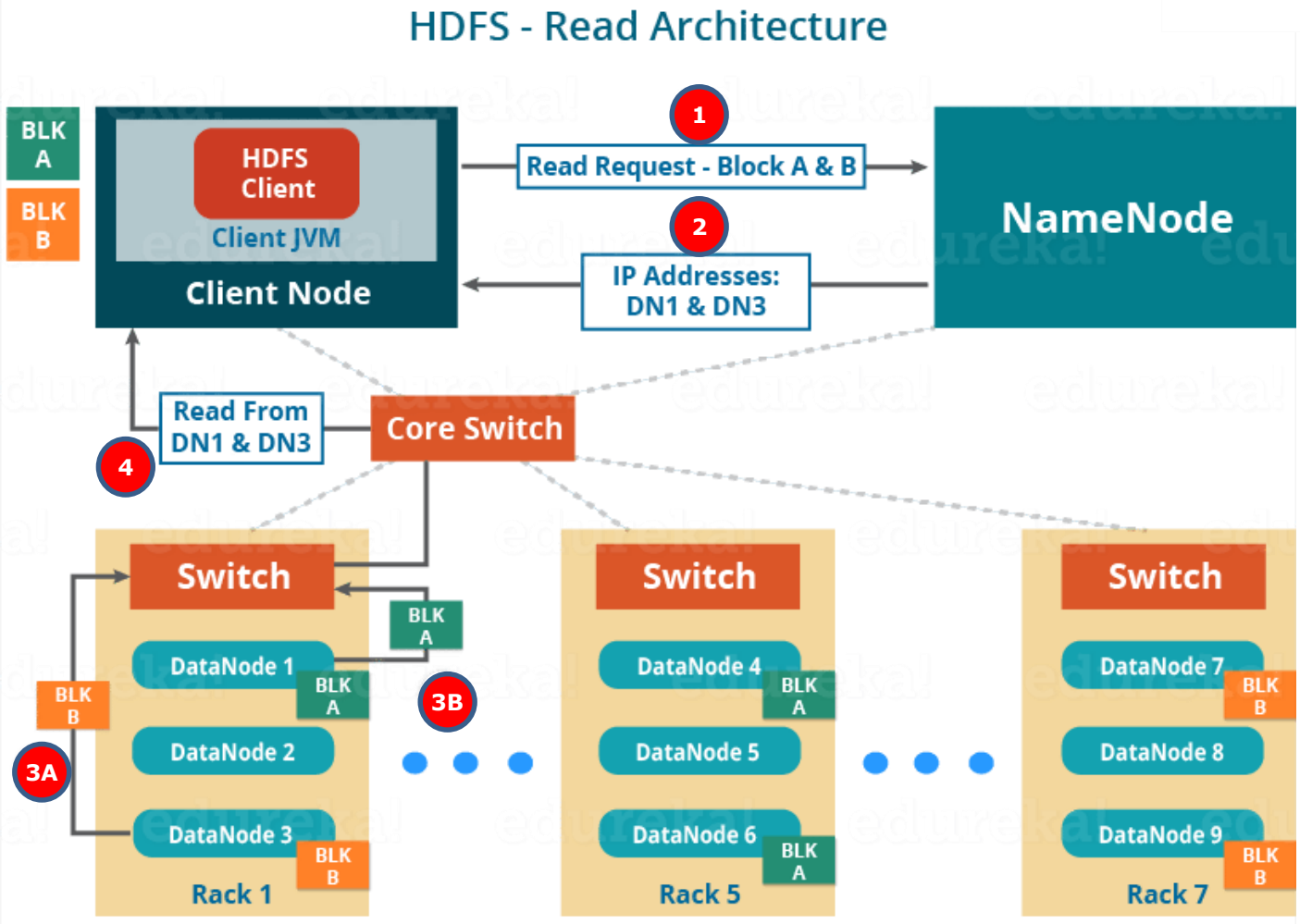
1. HDFS


Setting up HDFS - Write Pipeline
BLK A
HDFS Client
Client JVM
Client Node
1
Write Request - Block A
2
IP Addresses: DN1, DN4 & DN6
NameNode
3
Core Switch
4
5
Switch
DataNode 1
Ready
Rack 1
Switch
DataNode 4
Ready
6
DataNode 6
Ready
Rack 5
Switch
Rack 7

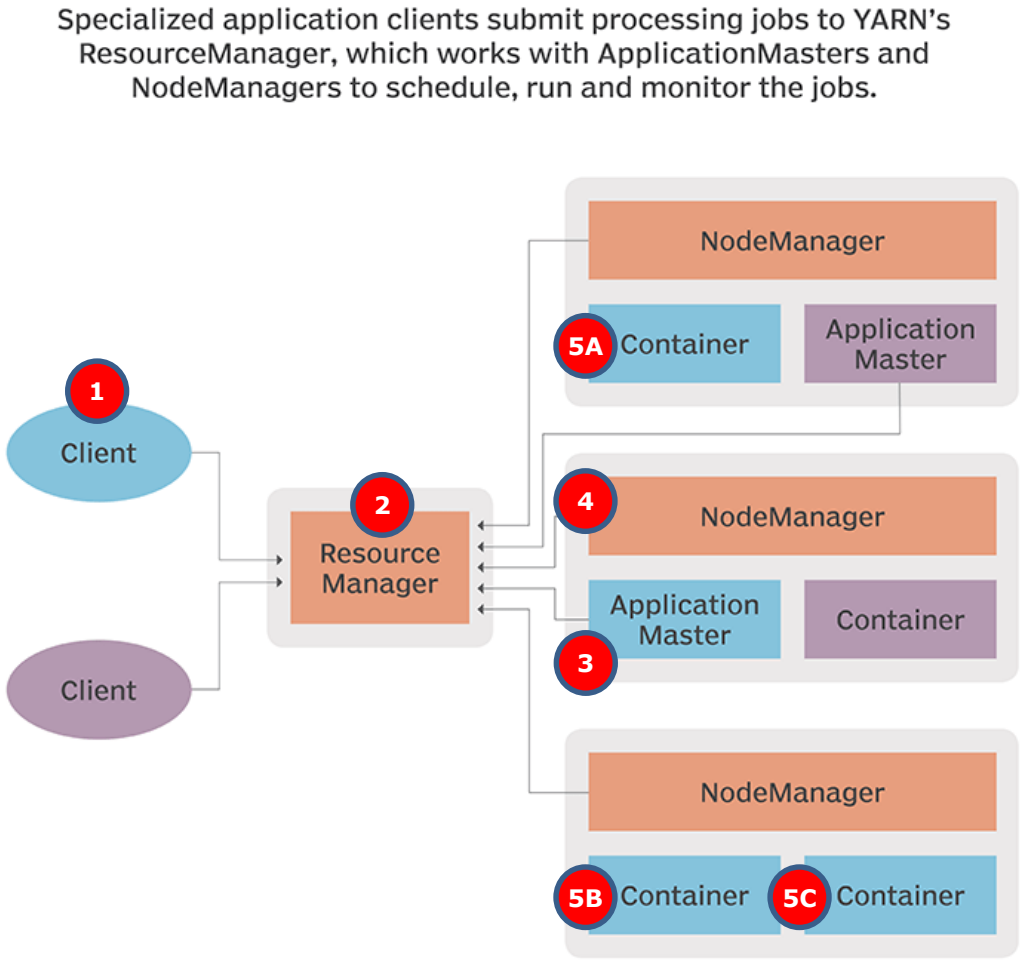
1. HDFS


Acknowledgement in HDFS - Write
BLK A
HDFS Client
Client JVM
Client Node
4
Write Successful
NameNode
ACK from DN1,DN4 & DN6
Core Switch
METADATA UPDATED
3
2
ACK from DN4 & DN6
Switch
Switch
Switch
DataNode 1
BLK A
BLK A Copied
BLK A Copied
DataNode 4
BLK A
1
ACK from DN6
DataNode 6
BLK A Copied
BLK A
Rack 1
Rack 5
Rack 7

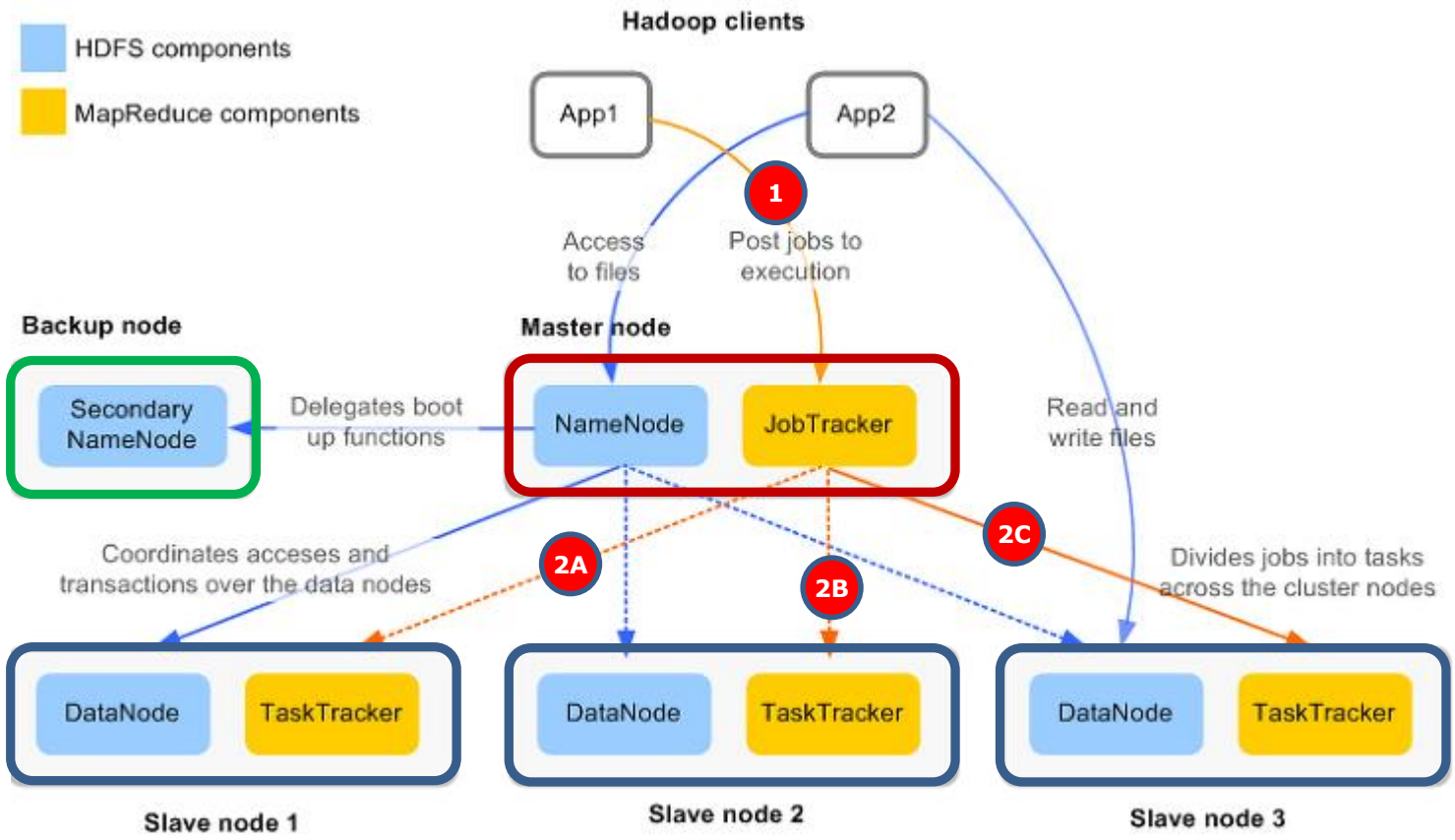
1. HDFS


HDFS - Read Architecture
BLK A
BLK B
HDFS Client
Client JVM
Client Node
1
Read Request - Block A & B
2
IP Addresses: DN1 & DN3
NameNode
Read From DN1 & DN3
4
Core Switch
Switch
BLK A
DataNode 1
BLK A
3B
BLK B
DataNode 2
3A
DataNode 3
BLK B
Rack 1
Switch
DataNode 4
BLK A
DataNode 5
DataNode 6
BLK A
Rack 5
Switch
DataNode 7
BLK B
DataNode 8
DataNode 9
BLK B
Rack 7

2. YARN

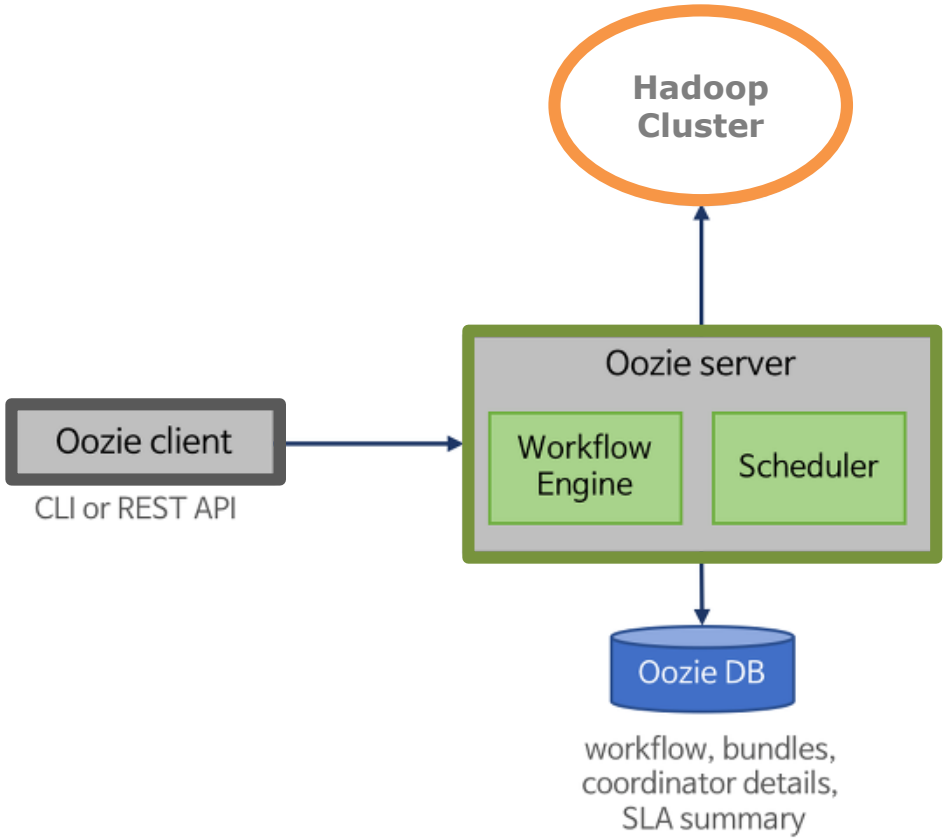

Specialized application clients submit processing jobs to YARN's ResourceManager, which works with ApplicationMasters and NodeManagers to schedule, run and monitor the jobs.
NodeManager
5A
Container
Application Master
1
Client
2
Resource Manager
4
NodeManager
Application Master
3
Container
Client
NodeManager
5B
Container
5C
Container

3. MR (MapReduce)



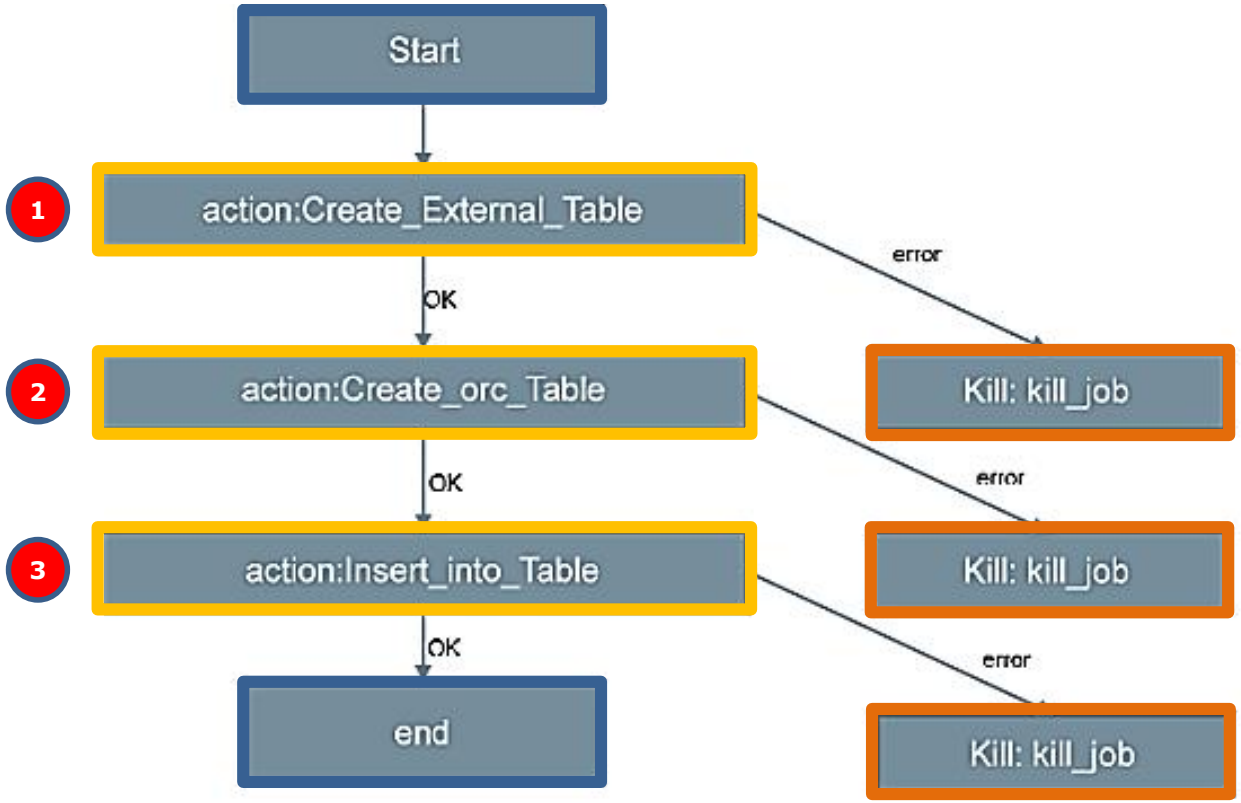
Oozie

Architecture



- Hadoop의 Job을 관리하기 위한 서버 기반의 **Workflow Scheduler System**
- **Java servlet-container**에서 수행되는 서버 기반의 **Workflow Engine**

DAG (**D**irected **A**cyclic **G**raph) 예시



Step 1 - DDL for Hive external table (**external.hive**)


```
Create external table external_table (
   name string,
   age int,
   address string,
   zip int
)
row format delimited
fields terminated by ','
stored as textfile
location '/test/abc';
```

Step 2 - DDL for Hive ORC table (**orc.hive**)


```
Create Table orc_table (
   name string, -- Concate value of first name and last name with space as seperator
   yearofbirth int,
   age int, -- Current year minus year of birth
   address string,
   zip int
)
STORED AS ORC;
```

Step 3 - Hive script to insert data from external table to ORC table (**Copydata.hql**)


```
use ${database_name}; -- input from Oozie
insert into table orc_table
select
concat(first_name,' ',last_name) as name,
yearofbirth,
year(from_unixtime) --yearofbirth as age,
address,
zip
from external_table
;
```

Step 4 - Create a workflow to execute all the above three steps. (**workflow.xml**)

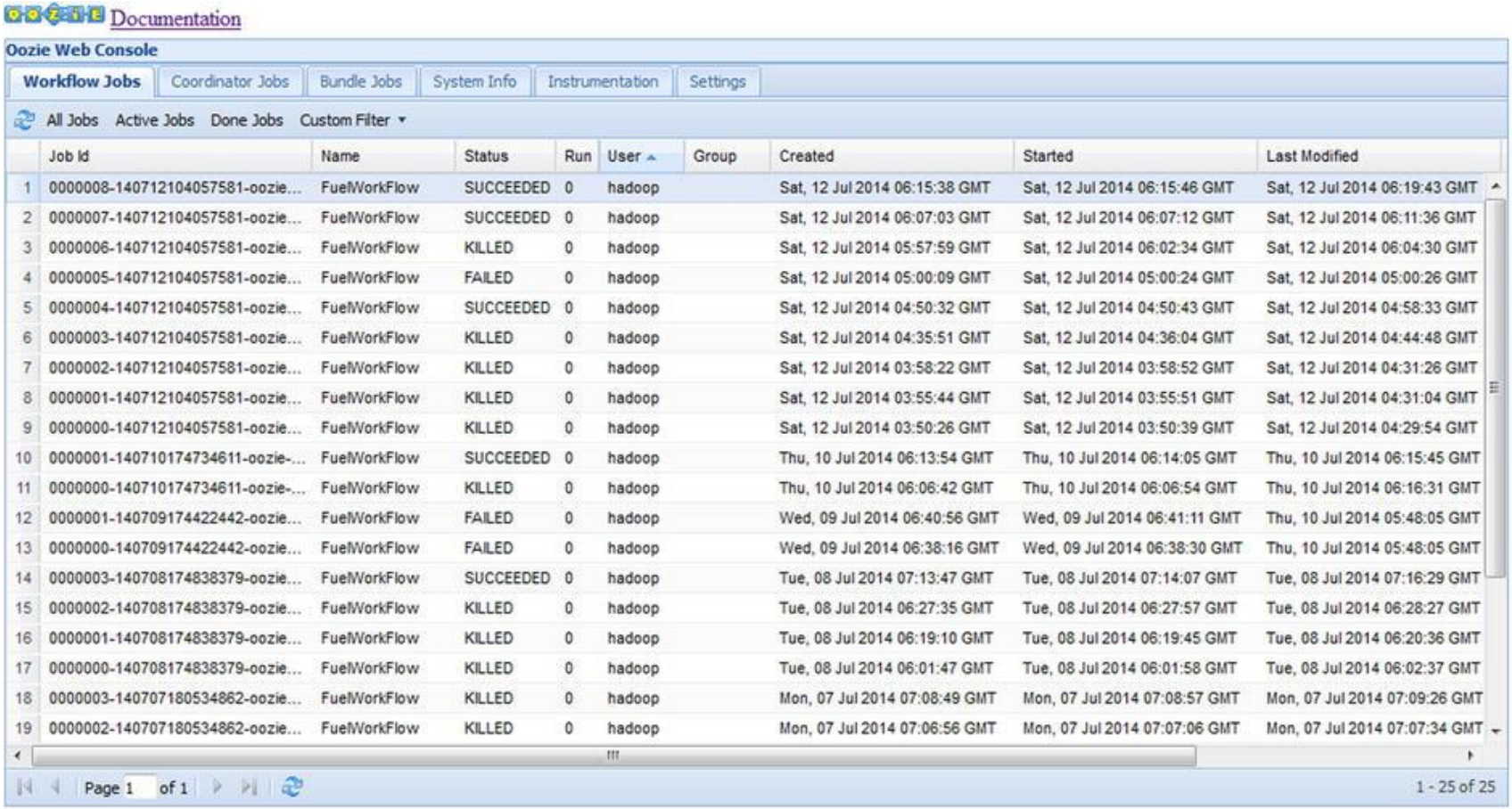
```
<!-- This is a comment -->
<workflow-app xmlns = "uri:oozie:workflow:0.4" name = "simple-Workflow">
  <start to = "Create_External_Table" />

  <!—Step 1 -->

  <action name = "Create_External_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
      <job-tracker>xyz.com:8088</job-tracker>
      <name-node>hdfs://rootname</name-node>
      <script>hdfs_path_of_script/external.hive</script>
    </hive>

    <ok to = "Create_orc_Table" />
    <error to = "kill_job" />
  </action>

  <!—Step 2 -->

  <action name = "Create_orc_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
      <job-tracker>xyz.com:8088</job-tracker>
      <name-node>hdfs://rootname</name-node>
      <script>hdfs_path_of_script/orc.hive</script>
    </hive>

    <ok to = "Insert_into_Table" />
    <error to = "kill_job" />
  </action>
```

Step 4 - Create a workflow to execute all the above three steps. (**workflow.xml**) *Cont'd*

```
  <!—Step 3 -->

  <action name = "Insert_into_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
      <job-tracker>xyz.com:8088</job-tracker>
      <name-node>hdfs://rootname</name-node>
      <script>hdfs_path_of_script/Copydata.hql</script>
      <param>database_name</param>
    </hive>

    <ok to = "end" />
    <error to = "kill_job" />
  </action>

  <kill name = "kill_job">
    <message>Job failed</message>
  </kill>

  <end name = "end" />

</workflow-app>
```

UI

Enable Oozie Web Console



Documentation

Oozie Web Console

Workflow Jobs | Coordinator Jobs | Bundle Jobs | System Info | Instrumentation | Settings

All Jobs | Active Jobs | Done Jobs | Custom Filter

	Job Id	Name	Status	Run	User	Group	Created	Started	Last Modified
1	0000008-140712104057581-oozie...	FuelWorkFlow	SUCCEEDED	0	hadoop		Sat, 12 Jul 2014 06:15:38 GMT	Sat, 12 Jul 2014 06:15:46 GMT	Sat, 12 Jul 2014 06:19:43 GMT
2	0000007-140712104057581-oozie...	FuelWorkFlow	SUCCEEDED	0	hadoop		Sat, 12 Jul 2014 06:07:03 GMT	Sat, 12 Jul 2014 06:07:12 GMT	Sat, 12 Jul 2014 06:11:36 GMT
3	0000006-140712104057581-oozie...	FuelWorkFlow	KILLED	0	hadoop		Sat, 12 Jul 2014 05:57:59 GMT	Sat, 12 Jul 2014 06:02:34 GMT	Sat, 12 Jul 2014 06:04:30 GMT
4	0000005-140712104057581-oozie...	FuelWorkFlow	FAILED	0	hadoop		Sat, 12 Jul 2014 05:00:09 GMT	Sat, 12 Jul 2014 05:00:24 GMT	Sat, 12 Jul 2014 05:00:26 GMT
5	0000004-140712104057581-oozie...	FuelWorkFlow	SUCCEEDED	0	hadoop		Sat, 12 Jul 2014 04:50:32 GMT	Sat, 12 Jul 2014 04:50:43 GMT	Sat, 12 Jul 2014 04:58:33 GMT
6	0000003-140712104057581-oozie...	FuelWorkFlow	KILLED	0	hadoop		Sat, 12 Jul 2014 04:35:51 GMT	Sat, 12 Jul 2014 04:36:04 GMT	Sat, 12 Jul 2014 04:44:48 GMT
7	0000002-140712104057581-oozie...	FuelWorkFlow	KILLED	0	hadoop		Sat, 12 Jul 2014 03:58:22 GMT	Sat, 12 Jul 2014 03:58:52 GMT	Sat, 12 Jul 2014 04:31:26 GMT
8	0000001-140712104057581-oozie...	FuelWorkFlow	KILLED	0	hadoop		Sat, 12 Jul 2014 03:55:44 GMT	Sat, 12 Jul 2014 03:55:51 GMT	Sat, 12 Jul 2014 04:31:04 GMT
9	0000000-140712104057581-oozie...	FuelWorkFlow	KILLED	0	hadoop		Sat, 12 Jul 2014 03:50:26 GMT	Sat, 12 Jul 2014 03:50:39 GMT	Sat, 12 Jul 2014 04:29:54 GMT
10	0000001-140710174734611-oozie-...	FuelWorkFlow	SUCCEEDED	0	hadoop		Thu, 10 Jul 2014 06:13:54 GMT	Thu, 10 Jul 2014 06:14:05 GMT	Thu, 10 Jul 2014 06:15:45 GMT
11	0000000-140710174734611-oozie-...	FuelWorkFlow	KILLED	0	hadoop		Thu, 10 Jul 2014 06:06:42 GMT	Thu, 10 Jul 2014 06:06:54 GMT	Thu, 10 Jul 2014 06:16:31 GMT
12	0000001-140709174422442-oozie...	FuelWorkFlow	FAILED	0	hadoop		Wed, 09 Jul 2014 06:40:56 GMT	Wed, 09 Jul 2014 06:41:11 GMT	Thu, 10 Jul 2014 05:48:05 GMT
13	0000000-140709174422442-oozie...	FuelWorkFlow	FAILED	0	hadoop		Wed, 09 Jul 2014 06:38:16 GMT	Wed, 09 Jul 2014 06:38:30 GMT	Thu, 10 Jul 2014 05:48:05 GMT
14	0000003-140708174838379-oozie...	FuelWorkFlow	SUCCEEDED	0	hadoop		Tue, 08 Jul 2014 07:13:47 GMT	Tue, 08 Jul 2014 07:14:07 GMT	Tue, 08 Jul 2014 07:16:29 GMT
15	0000002-140708174838379-oozie...	FuelWorkFlow	KILLED	0	hadoop		Tue, 08 Jul 2014 06:27:35 GMT	Tue, 08 Jul 2014 06:27:57 GMT	Tue, 08 Jul 2014 06:28:27 GMT
16	0000001-140708174838379-oozie...	FuelWorkFlow	KILLED	0	hadoop		Tue, 08 Jul 2014 06:19:10 GMT	Tue, 08 Jul 2014 06:19:45 GMT	Tue, 08 Jul 2014 06:20:36 GMT
17	0000000-140708174838379-oozie...	FuelWorkFlow	KILLED	0	hadoop		Tue, 08 Jul 2014 06:01:47 GMT	Tue, 08 Jul 2014 06:01:58 GMT	Tue, 08 Jul 2014 06:02:37 GMT
18	0000003-140707180534862-oozie...	FuelWorkFlow	KILLED	0	hadoop		Mon, 07 Jul 2014 07:08:49 GMT	Mon, 07 Jul 2014 07:08:57 GMT	Mon, 07 Jul 2014 07:09:26 GMT
19	0000002-140707180534862-oozie...	FuelWorkFlow	KILLED	0	hadoop		Mon, 07 Jul 2014 07:06:56 GMT	Mon, 07 Jul 2014 07:07:06 GMT	Mon, 07 Jul 2014 07:07:34 GMT

Page 1 of 1

1 - 25 of 25

데이터 파이프라인

빅데이터


Ambari


APACHE
Spark


Zeppelin


R Studio
R studio


jupyter
Jupyter


neo4j
Neo4j


druid


hadoop


HIVE


OOZIE


APACHE
ZooKeeper™


druid

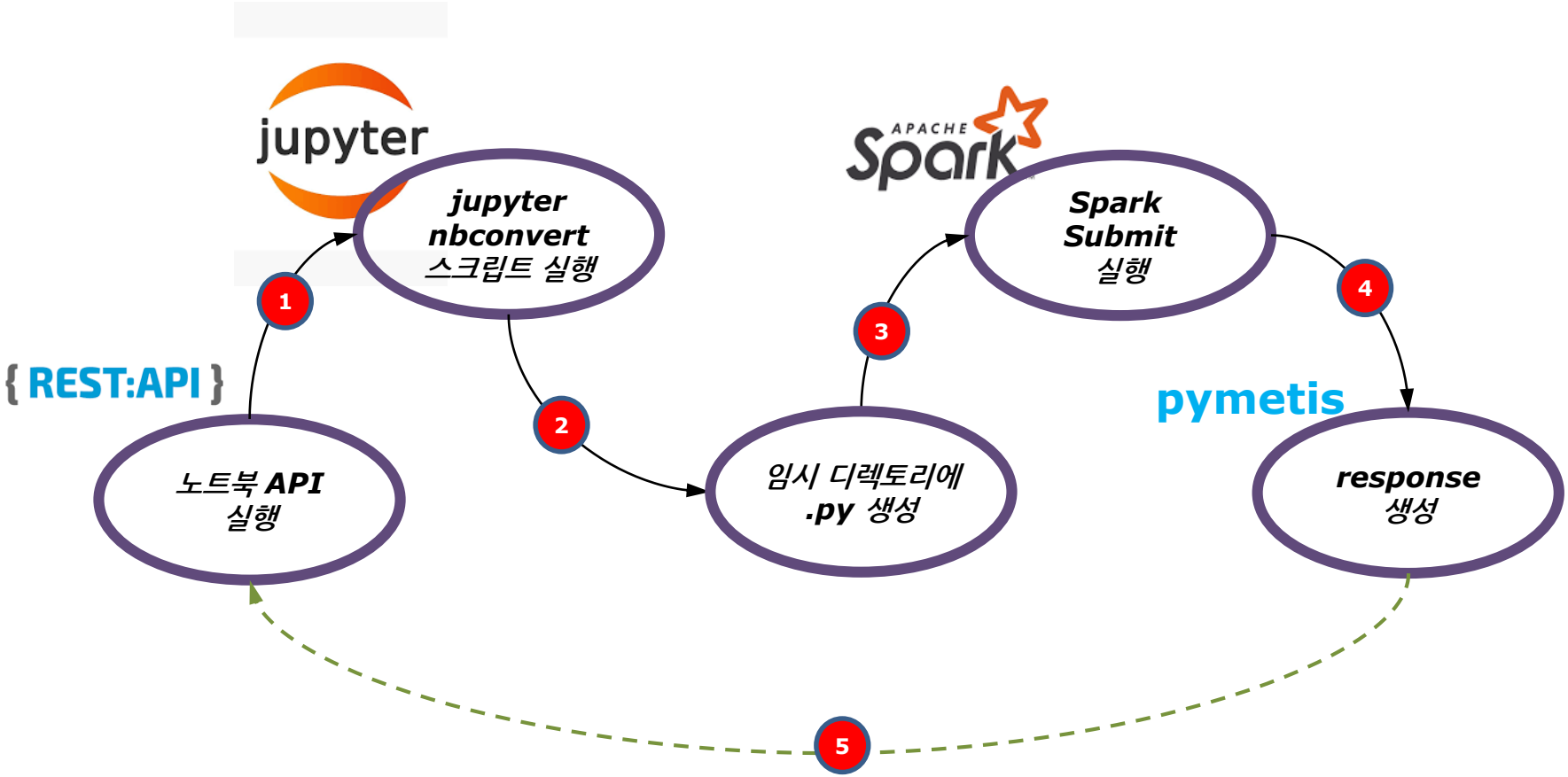

hadoop


APACHE
kafka


APACHE
ZooKeeper™

노트북 활용

Notebook API



국내 Open Platform

통합 데이터 지도

KDX 한국데이터거래소