**Oozie Workflow:**

The following are the Oozie workflow nodes:

Control Flow

• Start/end/kill

• Decision

• Fork/join

Actions

• Map-reduce

• Pig

• Hdfs

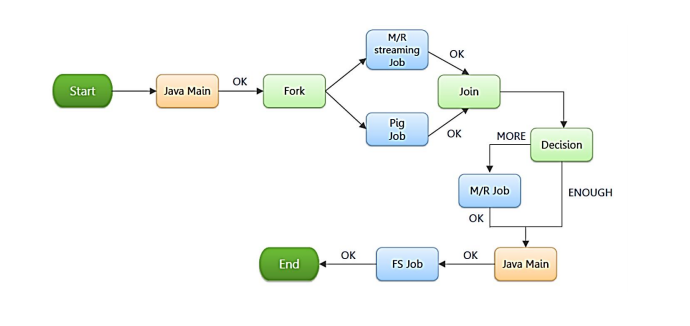
• Sub-workflow

• Java-run custom java code

To run Oozie workflows, two files are needed.

1. workflow.xml (stored in HDFS) - It contains the structure of workflow.

2. job.properties (stored in local) - It contains the configuration properties.



<workflow-app name="**Name of Workflow**" xmlns="uri:oozie:workflow:0.1">

<start to="**Start Node**"/>

<action name="**Name of Node**">

<hive xmlns="uri:oozie:hive-action:0.2">

<job-tracker>**Job Tracker Address**</job-tracker>

<name-node>**Name Node Address**</name-node>

<job-xml>**Path of hive-site.xml in HDFS**</job-xml>

<configuration>

<property>

<name>**Configuration Name**</name>

<value>**Configuration Value**</value>

</property>

</configuration>

<script>**Hive Query File**</script>

</hive>

<ok to="end"/>

<error to="end"/>

</action>

The benefits of Oozie Workflow are:

1. When new data arrives that requires the same processing,  
   we already have the workflow defined and ready to run.
2. As we build up a set of useful action definitions over time, creating new workflows becomes quicker and quicker.

**Sqoop Workflow**:

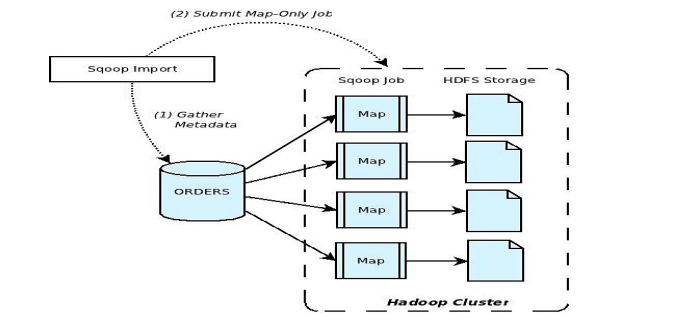
Sqoop allows easy import and export of data from structured data stores such as relational databases, enterprise data warehouses, and NoSQL systems. Using Sqoop, we can provision the data from external system on to HDFS, and populate tables in Hive and HBase. Sqoop integrates with Oozie, allowing you to schedule and automate import and export tasks.

The import is done in two steps:

• In the first Step Sqoop introspects the database to gather the necessary metadata for the data being imported.

• The second step is a map-only Hadoop job that Sqoop submits to the cluster.

It is this job that does the actual data transfer using the metadata captured in the previous step.



**Benefits:**

1. Efficiently transfer bulk data between Hadoop and external structured datastores, such as RDBMS and data warehouses, because databases are not easily accessible by Hadoop.
2. Ease of Use :  Configured server-side. This means that connectors will be configured in one place, managed by the Admin role and run by the Operator role. Likewise, JDBC drivers will be in one place and database connectivity will only be needed on the server.
3. Ease of Extension : Connectors are not restricted to the JDBC model, but can rather define their own vocabulary.
4. Security : Operate as a server based application with support for securing access to external systems via role-based access to Connection objects.