

SO WHAT HAVE WE DONE SO FAR...

Set up a workflow

Submitted and run it manually

Now let's run workflows based on time
and data triggers

COORDINATORS

COORDINATORS

Manually running a workflow on Oozie works in the very basic use cases

But if that was all that you had to do, you would not be using Oozie at all!

COORDINATORS

Often you want to run a workflow
based on certain triggers

1. At a specified time and frequency
Time trigger

2. When certain data becomes available
Data availability trigger

COORDINATORS

Time trigger

Data availability trigger

Normally you would have both. The job would start at a specified time only if the input data is present to process

COORDINATORS

Time trigger

Data availability trigger

Normally you would have both. The job would start at a specified time only if the input data is present to process

If the data is not available the job **waits** till it is

COORDINATORS

Time trigger

Data availability trigger

Coordinator jobs check for these
and **materialize** a workflow to
run the job, once the triggers are
activated

COORDINATORS

Time trigger

Data availability trigger

As of now the coordinator **only**
supports Oozie workflows and
each coordinator can only
support **one** workflow

COORDINATORS

Time trigger

A coordinator which is triggered purely based on time resembles a Unix **cron** job

COORDINATORS

Time trigger

Start time

Frequency

End time

COORDINATORS

Time trigger

Video 3 on running the cron job

Time trigger

COORDINATORS

job.properties

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/coordinator.xml

start=2016-01-01T00:00Z

end=2017-01-01T01:00Z

workflowAppUri=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/workflow.xml

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000  
jobTracker=localhost:8032  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/cron/coordinator.xml  
start=2016-01-01T00:00Z  
end=2017-01-01T01:00Z  
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/workflow.xml
```

We've seen all of these
properties when we
worked with workflows

Time trigger

COORDINATORS

Instead of a path to the workflow XML specify a path to the coordinator XML

```
nameNode=hdfs://localhost:9000  
jobTracker=localhost:8032  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/cron/coordinator.xml
```

```
start=2016-01-01T00:00Z
```

```
end=2017-01-01T01:00Z
```

```
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/workflow.xml
```

Time trigger

COORDINATORS

The key is different and the XML
it points to is also different

```
nameNode=hdfs://co...  
jobTracker=localot:  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/cron/coordinator.xml
```

```
start=2016-01-01T00:00Z
```

```
end=2017-01-01T01:00Z
```

```
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/workflow.xml
```


Time trigger

COORDINATORS

These are the variables which specify the start time and end time of the coordinator

```
nameNode=hdfs://localhost:9000  
jobTracker=localhost:8032  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/cron/coordinator.xml  
start=2016-01-01T00:00Z  
end=2017-01-01T01:00Z  
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/workflow.xml
```

The first invocation of the workflow will be at the start time and the last invocation be just before the end time

Time trigger

COORDINATORS

This variable holds the
workflow XML which has to be
specified in the coordinator.xml

```
nameNode=hdfs://namenode:8020  
jobTracker=localhost:8032  
queueName=default  
oozieRoot=oozie  
oozie.system.lispatch=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/cron/coordinator.xml  
start=2016-01-01T00:00Z  
end=2017-01-01T01:00Z
```

```
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/workflow.xml
```

Time trigger

COORDINATORS

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/coordinator.xml

start=2016-01-01T00:00Z

end=2017-01-01T01:00Z

workflowAppUri=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/workflow.xml

Time trigger

COORDINATORS

coordinator.xml

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
    start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
    <configuration>
      <property>
        <name>jobTracker</name>
        <value>${jobTracker}</value>
      </property>
      <property>
        <name>nameNode</name>
        <value>${nameNode}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>
```

Time trigger

COORDINATORS

A coordinator handles
one workflow i.e.
coordinates one
workflow, that is the
action for a
coordinator

```
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
    <configuration>
      <property>
        <name>jobTracker</name>
        <value>${jobTracker}</value>
      </property>
      <property>
        <name>nameNode</name>
        <value>${nameNode}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
    </configuration>
  </workflow>
</action>
```

```
</coordinator-app>
```


Time trigger

COORDINATORS

The app-path specifies
the path to the
workflow.xml

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"  
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">  
<action>  
  <workflow>  
    <app-path>${workflowAppUri}</app-path>  
    <configuration>  
      <property>  
        <name>jobTracker</name>  
        <value>${jobTracker}</value>  
      </property>  
      <property>  
        <name>nameNode</name>  
        <value>${nameNode}</value>  
      </property>  
      <property>  
        <name>queueName</name>  
        <value>${queueName}</value>  
      </property>  
    </configuration>  
  </workflow>  
</action>  
</coordinator-app>
```

For a standalone
workflow we would have
specified this in the
job.properties

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
  start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
    <configuration>
      <property>
        <name>jobTracker</name>
        <value>${jobTracker}</value>
      </property>
      <property>
        <name>nameNode</name>
        <value>${nameNode}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
    </configuration>
  </workflow>
</action>
```

```
</coordinator-app>
```

And here is the
configuration for the
workflow

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"  
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>  
  <workflow>  
    <app-path>${workflowAppUri}</app-path>  
    <configuration>  
      <property>  
        <name>jobTracker</name>  
        <value>${jobTracker}</value>  
      </property>  
      <property>  
        <name>name</name>  
        <value>${name}</value>  
      </property>  
      <property>  
        <name>queueName</name>  
        <value>${queueName}</value>  
      </property>  
    </configuration>  
  </workflow>  
</action>
```

```
</coordinator-app>
```

The interesting stuff is here

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"  
  start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

The time-triggered coordinator launches
the workflow at the start time

```
</coordinator-app>
```

Time trigger

COORDINATORS

The time-triggered coordinator launches the workflow at the start time

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"  
  start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>  
  <workflow>  
    <app-path>${workflowAppUri}</app-path>  
    <configuration>  
      <property>  
        <name>jobName</name>  
        <value>${jobName}</value>  
      </property>  
      <property>  
        <name>name</name>  
        <value>${name}</value>  
      </property>  
      <property>  
        <name>queueName</name>  
        <value>${queueName}</value>  
      </property>  
    </configuration>  
  </workflow>  
</action>  
</coordinator-app>
```

It continuously launches one at a predefined interval till the end time is reached

Time trigger

COORDINATORS

The time-triggered coordinator launches the workflow at the start time

It continuously launches one at a predefined interval till the end time is reached

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

And this is the frequency of materializing the coordinator action i.e. the workflow

```
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
    <configuration>
      <property>
        <name>jdbcDriverName</name>
        <value>${jdbcDriverName}</value>
      </property>
      <property>
        <name>jdbcUrl</name>
        <value>${jdbcUrl}</value>
      </property>
      <property>
        <name>jdbcUser</name>
        <value>${jdbcUser}</value>
      </property>
      <property>
        <name>jdbcPassword</name>
        <value>${jdbcPassword}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>
```

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"  
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>  
  <workflow>  
    <app-path>${workflowAppUri}</app-path>  
    <configuration>  
      <property>  
        <name>jobTracker</name>  
        <value>${jobTracker}</value>  
      </property>  
      <property>  
        <name>name</name>  
        <value>${name}</value>  
      </property>  
      <property>  
        <name>queueName</name>  
        <value>${queueName}</value>  
      </property>  
    </configuration>  
  </workflow>  
</action>  
</coordinator-app>
```

This is an EL function to
specify the frequency in a
convenient manner

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
```

`${coord:days(1)}` is preferred for a frequency of 1 day, it's better than hardcoding 1440 minutes

```
<configuration>
  <property>
    <name>jobTracker</name>
    <value>${jobTracker}</value>
  </property>
  <property>
    <name>mapred.job.tracker</name>
    <value>${mapred.job.tracker}</value>
  </property>
  <property>
    <name>mapred.framework.name</name>
    <value>${mapred.framework.name}</value>
  </property>
</configuration>
</workflow>
</action>
</coordinator-app>
```

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>
  <workflow>
    <app-path>${workflowAppUri}</app-path>
    <configuration>
      <property>
        <name>jcr:key</name>
        <value>{cron-coord}/value</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>
```

The workflow will be launched
every 5 minutes starting at
2016-01-01T00:00Z

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>
```

```
<workflow>
```

```
<app-path>${workflowAppUri}</app-path>
```

```
<configuration>
```

```
<property>
```

```
<name>jobTrackerName</name>
```

```
<value>${jobTracker}</value>
```

```
</property>
```

```
</configuration>
```

```
<property>
```

```
<name>nameNode</name>
```

```
<value>${nameNode}</value>
```

```
</property>
```

```
</configuration>
```

```
</workflow>
```

```
</action>
```

```
</coordinator-app>
```

At 2016-01-01T00:00Z, 2016-01-01T00:05Z,
2016-01-01T00:10Z, 2016-01-01T00:15Z,
2016-01-01T00:20Z, 2016-01-01T00:25Z etc.

Time trigger

COORDINATORS

At 2016-01-01T00:00Z, 2016-01-01T00:05Z,
2016-01-01T00:10Z, 2016-01-01T00:15Z,
2016-01-01T00:20Z, 2016-01-01T00:25Z etc.

```
<coordinator-app name="cron-coord" frequency="{coord:minutes(5)}"  
start="{start}" end="{end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<action>  
  <workflow>  
    <app-path>${workflowAppUri}</app-path>  
    <configuration>  
      <property>  
        <name>jobTracker</name>  
        <value>${jobTracker}</value>  
      </property>  
      <property>  
        <name>nameNode</name>  
        <value>${nameNode}</value>  
      </property>  
      <property>  
        <name>queueName</name>  
        <value>${queueName}</value>  
      </property>  
    </configuration>  
  </workflow>  
</action>  
</coordinator-app>
```

Each of these time instances is
called the **nominal** time of the action

Time trigger

COORDINATORS

workflow.xml

```
<workflow-app xmlns="uri:oozie:workflow:0.5" name="one-op-wf">
<start to="action"/>
<action name="action">
  <shell xmlns="uri:oozie:shell-action:0.2">
    <job-tracker>${jobTracker}</job-tracker>
    <name-node>${nameNode}</name-node>
    <configuration>
      <property>
        <name>mapred.job.queue.name</name>
        <value>${queueName}</value>
      </property>
    </configuration>
    <exec>echo</exec>
    <argument>my_output=Hello Oozie</argument>
    <capture-output/>
  </shell>
  <ok to="end"/>
  <error to="end"/>
</action>
<end name="end"/>
</workflow-app>
```

Time trigger

COORDINATORS

```
<workflow-app xmlns="uri:oozie:workflow:0.5" name="one-op-wf">
<start to="action"/>
<action name="action">
  <shell xmlns="uri:oozie:shell-action:0.2">
    <job-tracker>${jobTracker}</job-tracker>
    <name-node>${nameNode}</name-node>
    <configuration>
      <property>
        <name>mapred.job.queue.name</name>
        <value>${queueName}</value>
      </property>
    </configuration>
    <exec>echo</exec>
    <argument>my_output=Hello Oozie</argument>
    <capture-output/>
  </shell>
  <ok to="end"/>
  <error to="end"/>
</action>
<end name="end"/>
</workflow-app>
```

The workflow is
a simple shell
action

Time trigger

COORDINATORS

It's possible that when the coordinator materializes an action that there is a great backlog of jobs to complete

Time trigger

COORDINATORS

This would mean that all the
backlogged jobs would try and start
at the same time and overwhelm
Hadoop system resources

Time trigger

COORDINATORS

Oozie offers a way to control the jobs spun off by a coordinator

Time trigger

COORDINATORS

throttle

The maximum number of jobs
that can be in the WAITING
state at any time

Time trigger

COORDINATORS

throttle

The maximum number of jobs
that can be in the WAITING
state at any time

The system default is 12

Time trigger
throttle

COORDINATORS

timeout

The maximum time for which
a coordinator action can be in
the WAITING state

Time trigger
throttle

COORDINATORS

timeout

The maximum time for which
a coordinator action can be in
the **WAITING** state

The system default is **7** days

Time trigger

throttle

timeout

COORDINATORS

execution order

When multiple actions are ready, this determines which one is **chosen first** to execute

Time trigger

throttle

timeout

COORDINATORS

execution order

When multiple actions are ready, this determines which one is **chosen first** to execute

FIFO, LIFO and LAST_ONLY - only
FIFO is stable and fully tested

Time trigger

throttle

timeout

execution order

COORDINATORS

concurrency

This dictates how many
coordinator actions can run
simultaneously

Time trigger

throttle

timeout

execution order

COORDINATORS

concurrency

This dictates how many
coordinator actions can run
simultaneously

The default value is 1 and -1
means infinity

Time trigger

COORDINATORS

throttle

timeout

execution order

concurrency

Time trigger

COORDINATORS

job.properties

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

my_timeout=2

my_concurrency=2

my_execution=FIFO

my_throttle=5

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/coordinator.xml

start=2016-01-01T00:00Z

end=2017-01-01T01:00Z

workflowAppUri=\${nameNode}/user/\${user.name}/\${oozieRoot}/cron/workflow.xml

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000  
jobTracker=localhost:8032  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
my_timeout=2  
my_concurrency=2  
my_execution=FIFO  
my_throttle=5
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${  
{oozieRoot}/cron/coordinator.xml  
start=2016-01-01T00:00Z  
end=2017-01-01T01:00Z  
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/  
workflow.xml
```

Set up values for each of the execution controls

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000
jobTracker=localhost:8032
queueName=default
oozieRoot=oozie
oozie.system.libpath=true
```

```
my_timeout=2
```

```
my_concurrency=2
```

```
my_execution=FIFO
```

```
my_throttle=5
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${
{oozieRoot}/cron/coordinator.xml
start=2016-01-01T00:00Z
end=2017-01-01T01:00Z
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/
workflow.xml
```

Set the timeout for jobs in
the **WAITING** state to **2**
minutes

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000
jobTracker=localhost:8032
queueName=default
oozieRoot=oozie
oozie.system.libpath=true
```

```
my_timeout=2
```

```
my_concurrency=2
```

```
my_execution=FIFO
```

```
my_throttle=5
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${
{oozieRoot}/cron/coordinator.xml
start=2016-01-01T00:00Z
end=2017-01-01T01:00Z
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/
workflow.xml
```

Only 2 actions can
run at a time

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000
jobTracker=localhost:8032
queueName=default
oozieRoot=oozie
oozie.system.libpath=true
```

```
my_timeout=2
my_concurrency=2
my_execution=FIFO
my_throttle=5
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${
{oozieRoot}/cron/coordinator.xml
start=2016-01-01T00:00Z
end=2017-01-01T01:00Z
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/
workflow.xml
```

Execute the actions
in first in first out
order

Time trigger

COORDINATORS

```
nameNode=hdfs://localhost:9000
jobTracker=localhost:8032
queueName=default
oozieRoot=oozie
oozie.system.libpath=true
```

```
my_timeout=2
my_concurrency=2
my_execution=FIFO
my_throttle=5
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${
{oozieRoot}/cron/coordinator.xml
start=2016-01-01T00:00Z
end=2017-01-01T01:00Z
workflowAppUri=${nameNode}/user/${user.name}/${oozieRoot}/cron/
workflow.xml
```

Allow only 5 jobs to
be in the **WAITING**
state at a time

Time trigger

COORDINATORS

coordinator.xml

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
                 start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
  <controls>
    <timeout>${my_timeout}</timeout>
    <concurrency>${my_concurrency}</concurrency>
    <execution>${my_execution}</execution>
    <throttle>${my_throttle}</throttle>
  </controls>
  <action>
    <workflow>
      <app-path>${workflowAppUri}</app-path>
      <configuration>
        <property>
          <name>jobTracker</name>
          <value>${jobTracker}</value>
        </property>
        <property>
          <name>nameNode</name>
          <value>${nameNode}</value>
        </property>
        <property>
          <name>queueName</name>
          <value>${queueName}</value>
        </property>
      </configuration>
    </workflow>
  </action>
</coordinator-app>
```

Time trigger

COORDINATORS

```
<coordinator-app name="cron-coord" frequency="${coord:minutes(5)}"
  start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<controls>
```

```
  <timeout>${my_timeout}</timeout>
```

```
  <concurrency>${my_concurrency}</concurrency>
```

```
  <execution>${my_execution}</execution>
```

```
  <throttle>${my_throttle}</throttle>
```

```
</controls>
```

```
<action>
```

```
  <workflow>
```

```
    <app-path>${workflowAppUri}</app-path>
```

```
    <configuration>
```

```
      <property>
```

```
        <name>job-tracker</name>
```

```
        <value>${job_tracker}</value>
```

```
      </property>
```

```
      <property>
```

```
        <name>hadoop.job.name</name>
```

```
        <value>${hadoop.job.name}</value>
```

```
      </property>
```

```
      <property>
```

```
        <name>hadoop.job.queue</name>
```

```
        <value>${hadoop.job.queue}</value>
```

```
      </property>
```

```
    </configuration>
```

```
  </workflow>
```

```
</action>
```

```
</coordinator-app>
```

These controls apply to the coordinator actions which are materialized by this coordinator

COORDINATORS

Same video 3, show the number of running jobs

COORDINATORS

Often you want to run a workflow
based on certain triggers

1. At a specified time and frequency
Time trigger

2. When certain data becomes available
Data availability trigger

Data availability trigger COORDINATORS

At some nominal time for a coordinator action, if the data for the job is not available for the job, it will not run

Data availability trigger COORDINATORS

Oozie supports **directory** based data triggers as well as **metadata** based data triggers

Directory triggers are more common and we'll discuss those here

Data availability trigger COORDINATORS

The input data for the job can be produced at **fixed** intervals or **adhoc**

Oozie only supports data sets which are generated at fixed intervals i.e.
synchronous data sets

Data availability trigger COORDINATORS

Now, data which is generated at fixed intervals are usually stored in a pretty standard manner - with some date based structure

Data availability trigger COORDINATORS

YEAR

Folder Structure

MONTH 1

DAY 1

DAY 2

HOURL 1

MINUTE 30

Data availability trigger

COORDINATORS

2016

Folder Structure

01

01

02

00

30

COORDINATORS

Video 4, folder structure as well as running the aggregator

Data availability trigger **COORDINATORS**

How do we define what data is
required by the workflow?

Data availability trigger COORDINATORS

How do we define what data is
required by the workflow?

dataset


input-events

output-events

Data availability trigger **COORDINATORS**

output-events

This defines the data input



dataset
input-events

Data availability trigger COORDINATORS dataset

This is a set of data which
exists and which the workflow
jobs need to run

Data availability trigger **COORDINATORS**

This is a set of data which **dataset**
exists and which the workflow
jobs need to run

Oozie supports **synchronous**
data produced at well defined
intervals

Data availability trigger COORDINATORS

This is a set of data which exists and which the workflow jobs need to run **dataset** Oozie supports **synchronous** data produced at well defined intervals

This is the dataset that the job is interested in

We now need to specify the **actual instances** of data in this dataset

Data availability trigger **COORDINATORS**

We now need to specify the **actual**
instances of data in this dataset

input-events

The **specific** data in the dataset
which we're interested in

Data availability trigger **COORDINATORS**

output-events

**This is the data that is produced
as output by the workflow**

Data availability trigger **COORDINATORS**

output-events

This is the data that is produces
as output by the workflow

The workflow does not wait
for this data, it just outputs it
after processing is complete

COORDINATORS

Same video 4, folder structure of input data

Data availability trigger COORDINATORS

job.properties

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/aggregator/
coordinator.xml

start=2016-01-01T01:00Z

end=2017-01-01T03:00Z

Data availability trigger COORDINATORS

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/aggregator/
coordinator.xml

start=2016-01-01T01:00Z

end=2017-01-01T01:00Z

We know this part really well

Data availability trigger COORDINATORS

We point to the coordinator.xml in the application path and specify start and end time for this coordinator

```
nameNode=hdfs://localhost:9000  
jobTracker=localhost:8020  
queueName=default  
oozieRoot=oozie  
oozie.system.libpath=true
```

```
oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieRoot}/aggregator/  
coordinator.xml
```

```
start=2016-01-01T01:00Z
```

```
end=2017-01-01T03:00Z
```

Data availability trigger COORDINATORS

job.properties

nameNode=hdfs://localhost:9000

jobTracker=localhost:8032

queueName=default

oozieRoot=oozie

oozie.system.libpath=true

oozie.coord.application.path=\${nameNode}/user/\${user.name}/\${oozieRoot}/aggregator/
coordinator.xml

start=2016-01-01T01:00Z

end=2017-01-01T03:00Z

Data availability trigger COORDINATORS

coordinator.xml

```
<coordinator-app name="aggregator-coord" frequency="${coord:hours(1)}" start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
<controls>
  <concurrency>1</concurrency>
</controls>

<datasets>
  <dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z" timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>
  </dataset>
  <dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z" timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/aggregatedLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}</uri-template>
  </dataset>
</datasets>

<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>

<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>

<action>
  <workflow>
    <app-path>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator</app-path>
    <configuration>
      <property>
        <name>jobTracker</name>
        <value>${jobTracker}</value>
      </property>
      <property>
        <name>nameNode</name>
        <value>${nameNode}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
      <property>
        <name>inputData</name>
        <value>${coord:dataIn('input')}</value>
      </property>
      <property>
        <name>outputData</name>
        <value>${coord:dataOut('output')}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>
```

Data availability trigger COORDINATORS

```
<action>
  <workflow>
    <app-path>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator</app-path>
    <configuration>
      <property>
        <name>jobTracker</name>
        <value>${jobTracker}</value>
      </property>
      <property>
        <name>nameNode</name>
        <value>${nameNode}</value>
      </property>
      <property>
        <name>queueName</name>
        <value>${queueName}</value>
      </property>
      <property>
        <name>inputData</name>
        <value>${coord:dataIn('input')}</value>
      </property>
      <property>
        <name>outputData</name>
        <value>${coord:dataOut('output')}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>
```

This is the
workflow,
exactly like we've
seen before

Data availability trigger COORDINATORS

```
<coordinator-app name="aggregator-coord" frequency="${coord:hours(1)}"
start="${start}" end="${end}" timezone="UTC"
xmlns="uri:oozie:coordinator:0.2">
```

```
<controls>
```

```
<concurrency>1</concurrency>
```

```
</controls>
```

```
<datasets>
```

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z" timezone="UTC">
```

```
<uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>
```

```
</dataset>
```

```
<dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z" timezone="UTC">
```

```
<uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/aggregatedLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}</uri-template>
```

```
</dataset>
```

```
</datasets>
```

```
<input-events>
```

```
<data-in name="input" dataset="raw-logs">
```

```
<start-instance>${coord:current(2)}</start-instance>
```

```
<end-instance>${coord:current(0)}</end-instance>
```

```
</data-in>
```

```
</input-events>
```

```
<output-events>
```

```
<data-out name="output" dataset="aggregated-logs">
```

```
<instance>${coord:current(0)}</instance>
```

```
</data-out>
```

```
</output-events>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

```
</coordinator-app>
```

This job runs every hour between the start and end time

```
</coordinator-app>
```


Data availability trigger COORDINATORS

```
<coordinator-app name="aggregator-coord" frequency="${coord:hours(1)}" start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
```

```
<controls>
  <concurrency>1</concurrency>
</controls>
```

```
<datasets>
  <dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z" timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>
  </dataset>
  <dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z" timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/aggregatedLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}</uri-template>
  </dataset>
</datasets>
```

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>

<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>
```

```
<action name="aggregator" class="org.apache.oozie.aggregator.Aggregator" location="aggregator.xml" start="true">
  <input>
    <dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z" timezone="UTC">
      <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>
    </dataset>
  </input>
  <output>
    <dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z" timezone="UTC">
      <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/aggregatedLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}</uri-template>
    </dataset>
  </output>
</action>
</coordinator-app>
```

We want just one job to run at any time

Data availability trigger COORDINATORS

```
<coordinator-app name="aggregator-coord" frequency="${coord:hours(1)}" start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
<controls>
  <concurrency>1</concurrency>
</controls>
```

```
<datasets>
  <dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z"
  timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${
    {YEAR}}/${MONTH}}/${DAY}}/${HOUR}}/${MINUTE}</uri-template>
  </dataset>
  <dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z"
  timezone="UTC">
    <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/
    aggregatedLogs/${YEAR}}/${MONTH}}/${DAY}}/${HOUR}</uri-template>
  </dataset>
</datasets>
```

These are the datasets that this workflow is interested in

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current}</end-instance>
  </data-in>
</input-events>

<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>
```

```
</coordinator-app>
```

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

We'll focus on the raw-logs data
set which is the input of this
aggregator job

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

The data set used by the workflow managed by this coordinator is generated every 20 minutes

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

We only care about the data set
generated after this time instance
- ignore anything before this

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

This specifies how the data is
available, the directory structure
where the input exists for every 20
minutes

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

All the input data is in the rawLogs
file in this HDFS directory

Data availability trigger COORDINATORS

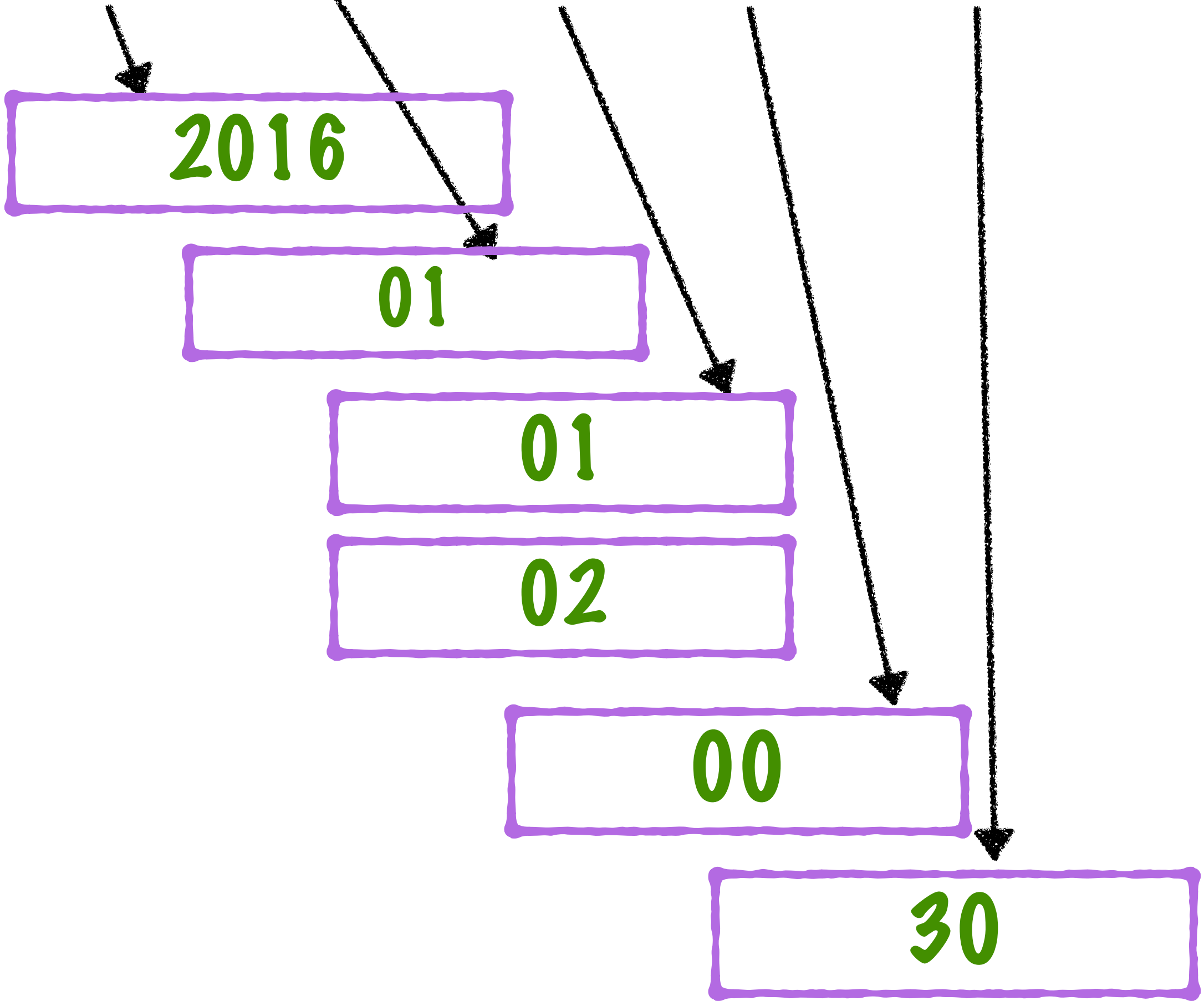
```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```

This specifies how the data is
available, the directory structure
where the input is stored

Remember we have raw logs for
every 20 minutes

Data availability trigger COORDINATORS

```
<dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-  
instance="2016-01-01T00:00Z" timezone="UTC">  
  <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/  
input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>  
</dataset>
```



Data availability trigger COORDINATORS

```
<coordinator-app name="aggregator-coord" frequency="${coord:hours(1)}" start="${start}" end="${end}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
  <controls>
    <concurrency>1</concurrency>
  </controls>
  <datasets>
    <dataset name="raw-logs" frequency="${coord:minutes(20)}" initial-instance="2016-01-01T00:00Z" timezone="UTC">
      <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/input-data/rawLogs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MINUTE}</uri-template>
    </dataset>
    <dataset name="aggregated-logs" frequency="${coord:hours(1)}" initial-instance="2016-01-01T01:00Z"
    timezone="UTC">
      <uri-template>${nameNode}/user/${coord:user()}/${oozieRoot}/aggregator/output-data/aggregatedLogs/${
{YEAR}}/${MONTH}}/${DAY}}/${HOUR}</uri-template>
    </dataset>
  </datasets>
  <input-events>
    <data-in name="input" dataset="raw-logs">
      <start-instance>${coord:current(-2)}</start-instance>
      <end-instance>${coord:current(0)}</end-instance>
    </data-in>
  </input-events>
  <output-events>
    <data-out name="output" dataset="aggregated-logs">
      <instance>${coord:current(0)}</instance>
    </data-out>
  </output-events>
</coordinator-app>
```

This is the data set that is produced as the **output** of this workflow - how we specify which datasets are input and output, we'll see in a bit

Data availability trigger COORDINATORS

```
<data-set>  
  <done-flag>_trigger<done-flag>  
</data-set>
```

The done-flag is an additional node
within the data-set node

Data availability trigger COORDINATORS

```
<data-set>  
  <done-flag>_trigger<done-flag>  
</data-set>
```

This specifies **what file** the Oozie should look for in the directory to know that the data is indeed available

Data availability trigger COORDINATORS

```
<data-set>  
  <done-flag>_trigger<done-flag>  
</data-set>
```

The default value for this is the
_SUCCESS file which Hadoop jobs add
to their output directory

Data availability trigger COORDINATORS

```
<data-set>  
  <done-flag><done-flag>  
</data-set>
```

If the element is specified but is **empty**
then the **presence of the directory itself**
is a signal that the data is available

Data availability trigger

<input-events>

```
<data-in name="input" dataset="raw-logs">
```

<start-instance> $\{coord:current(-2)\}$ **</start-instance>**

<end-instance> $\{coord:current(0)\}$ **</end-instance>**

</data-in>

`</input-events>`

```
<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>
```

[illegible]

This tells us which specific instances of the data set we're interested in

Data availability trigger

data-in specifies the raw-logs dataset as the required input data

<input-events>

```
<data-in name="input" dataset="raw-logs">
```

```
<start-instance>${coord:current(-2)}</start-instance>
```

<end-instance> $\{coord:current(0)\}$ **</end-instance>**

</data-in>

</input-events>

<output-events>

```
<data-out name="output" dataset="aggregated-logs">
```

```
<instance>${coord:current(0)}</instance>
```

</data-out>

</output-events>

</coordinator-app>

Data availability trigger COORDINATORS

And these tell us the exact instances of the data needed to run a job

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>
```

```
<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>
```

```
</coordinator-app>
```

Use `<instance>` for one instance
and `<start-instance>` and `<end-
instance>` for a range of instances

Data availability trigger COORDINATORS

These EL functions allow us to specify data instances relative to the time the job was run

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>
```

```
<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>
```

```
</coordinator-app>
```

current(0) is the last data set available just before the time the workflow runs

Data availability trigger COORDINATORS

coord:current(0)

2016/01/01/01/00

2016/01/01/01/20

2016/01/01/01/40

2016/01/01/02/00

2016/01/01/02/20

2016/01/01/02/40

**data generated for 1st
January, 2016
between 1am and 3am**

**at 20 minute
intervals**

Data availability trigger COORDINATORS

coord: current(0)

2016/01/01/01/00

2016/01/01/01/20

2016/01/01/01/40

job runs at 1:55am on 1st Jan, 2016

2016/01/01/02/00

2016/01/01/02/20

2016/01/01/02/40

Data availability trigger COORDINATORS

coord: current(0)

2016/01/01/01/00

2016/01/01/01/20

2016/01/01/01/40

2016/01/01/02/00

job runs at 2:12am on 1st Jan, 2016

2016/01/01/02/20

2016/01/01/02/40

Data availability trigger **COORDINATORS**

coord: current(0)

2016/01/01/01/00

job runs at 1:00am on 1st Jan, 2016

2016/01/01/01/20

2016/01/01/01/40

2016/01/01/02/00

2016/01/01/02/20

2016/01/01/02/40

Data availability trigger

current(-2) is 2 before the latest data

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>

<output-events>
  <data-out name="output" dataset="aggregated-logs">
    <instance>${coord:current(0)}</instance>
  </data-out>
</output-events>

</coordinator-app>
```

This entire data range needs to be available before the job can run

Data availability trigger COORDINATORS

2016/01/01/01/00

2016/01/01/01/20

coord: current(-2)

2016/01/01/01/40

2016/01/01/02/00

coord: current(0)

job runs at 2:00am on 1st Jan, 2016

2016/01/01/02/20

2016/01/01/02/40

Data availability trigger

This is the output data, just

once instance for every run

```
<start-instance name="aggr-coord-start">
  <coord-start><coord-start(0)>
    <start-instance>{{start}}</start-instance>
  </coord-start>
</start-instance>

<end-instance name="aggr-coord-end">
  <coord-end><coord-end(0)>
    <end-instance>{{end}}</end-instance>
  </coord-end>
</end-instance>

<dataset>
```

One

```
<input-events>
  <data-in name="input" dataset="raw-logs">
    <start-instance>${coord:current(-2)}</start-instance>
    <end-instance>${coord:current(0)}</end-instance>
  </data-in>
</input-events>
```

<output-events>

```
<data-out name="output" dataset="aggregated-logs">
  <instance>${coord:current(0)}</instance>
</data-out>
```

</output-events>

[illegible]

Specify one data-out

COORDINATORS

Video 4 to explain the use of the right data set instances

Data availability trigger COORDINATORS

There is no change in the
workflow.xml

COORDINATORS

Often you want to run a workflow
based on certain triggers

1. At a specified time and frequency
Time trigger

2. When certain data becomes available
Data availability trigger