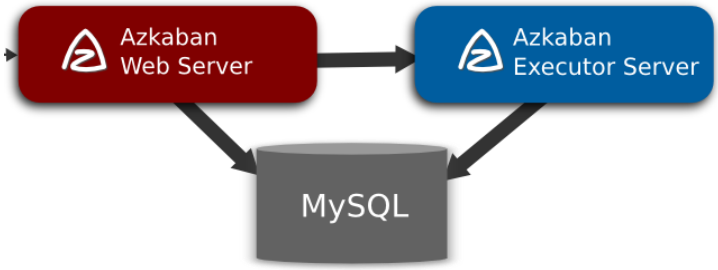
Azkaban Hadoop – A workflow scheduler for Hadoop

Azkaban is simple batch scheduler for constructing and running Hadoop jobs or other offline processes.
A workflow scheduler allows you to string together a group of processes to run in an order that respects the dependencies between the jobs.

Batch jobs need to be scheduled to run periodically. They also typically have intricate dependency chains—for example, dependencies on various data extraction processes or previous steps. Larger processes might have 50 or 60 steps, of which some might run in parallel and others must wait for the output of earlier steps.Azkaban is a workflow scheduler that allows the independent pieces to be declaratively assembled into a single workflow, and for that workflow to be scheduled to run periodically.

Azkaban consists of 3 key components:

- Relational Database (MySQL)
- AzkabanWebServer
- AzkabanExecutorServer



Relational Database (MySQL)
Azkaban uses MySQL to store much of its state. Both the AzkabanWebServer and the AzkabanExecutorServer access the DB.

How does AzkabanWebServer use the DB?

The web server uses the db for the following reasons:

- **Project Management** - The projects, the permissions on the projects as well as the uploaded files.
- **Executing Flow State** - Keep track of executing flows and which Executor is running them.
- **Previous Flow/Jobs** - Search through previous executions of jobs and flows as well as access their log files.
- **Scheduler** - Keeps the state of the scheduled jobs.
- **SLA** - Keeps all the sla rules

How does the AzkabanExecutorServer use the DB?

The executor server uses the db for the following reasons:

- **Access the project** - Retrieves project files from the db.
- **Executing Flows/Jobs** - Retrieves and updates data for flows and that are executing
- **Logs** - Stores the output logs for jobs and flows into the db.
- **Interflow dependency** - If a flow is running on a different executor, it will take state from the DB.

AzkabanWebServer

The AzkabanWebServer is the main manager to all of Azkaban. It handles project management, authentication, scheduler, and monitoring of executions. It also serves as the web user interface.

Using Azkaban is easy. Azkaban uses *.job key-value property files to define individual tasks in a work flow, and the _dependencies_ property to define the dependency chain of the jobs. These job files and associated code can be archived into a *.zip and uploaded through the web server through the Azkaban UI or through curl.

AzkabanExecutorServer

Previous versions of Azkaban had both the AzkabanWebServer and the AzkabanExecutorServer features in a single server. The Executor has since been separated into its own server.

Getting Started

In version 3.0 there are three modes: the stand alone "solo-server" mode, the heavier weight two server mode and distributed multiple-executor mode. The following describes the differences between the two modes.

In **solo server mode**, the DB is embedded H2 and both web server and executor server run in the same process. This should be useful if one just wants to try things out. It can also be used on small scale use cases.

The **two server mode** is for more serious production environment. Its DB should be backed by MySQL instances with master-slave set up. The web server and executor server should run in different processes so that upgrading and maintenance shouldn't affect users.

The **multiple executor mode** is for most serious production environment. Its DB should be backed by MySQL instances with master-slave set up. The web server and executor servers should ideally run in different hosts so that upgrading and maintenance shouldn't affect users. This multiple host setup brings in robust and scalable aspect to Azkaban.

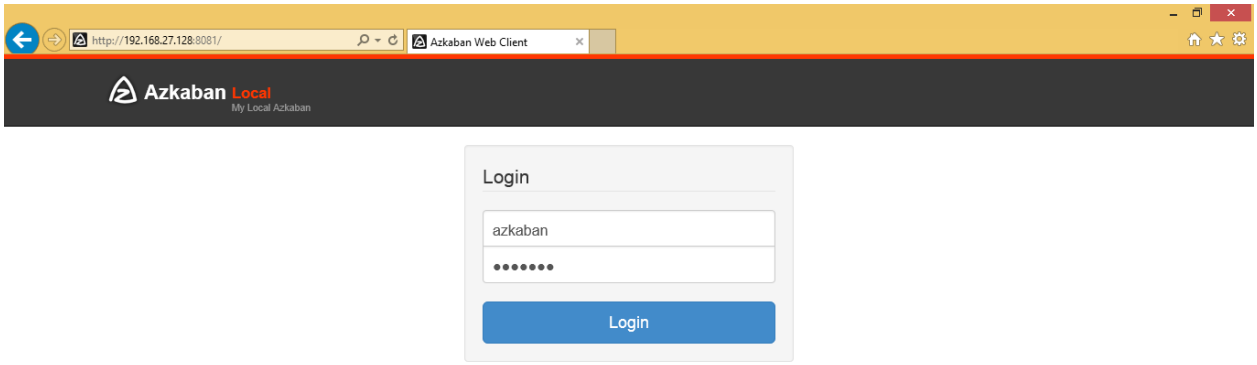
Azkaban Features

- Time based dependency scheduling of workflows for hadoop jobs and compatible with any version of hadoop.
- Provides easy to use web UI and rich set of vitualizations for displaying interactive graph in browser.
- Provides support for e-mail alert on failure and successes of workflow and also supports SLA(Service Level Agreement) alerting via e-mails.
- We can retry running the failed jobs again via browser itself.
- Tracks user actions and supports authentication.
- Provides separate project work spaces for each project for easier future references.

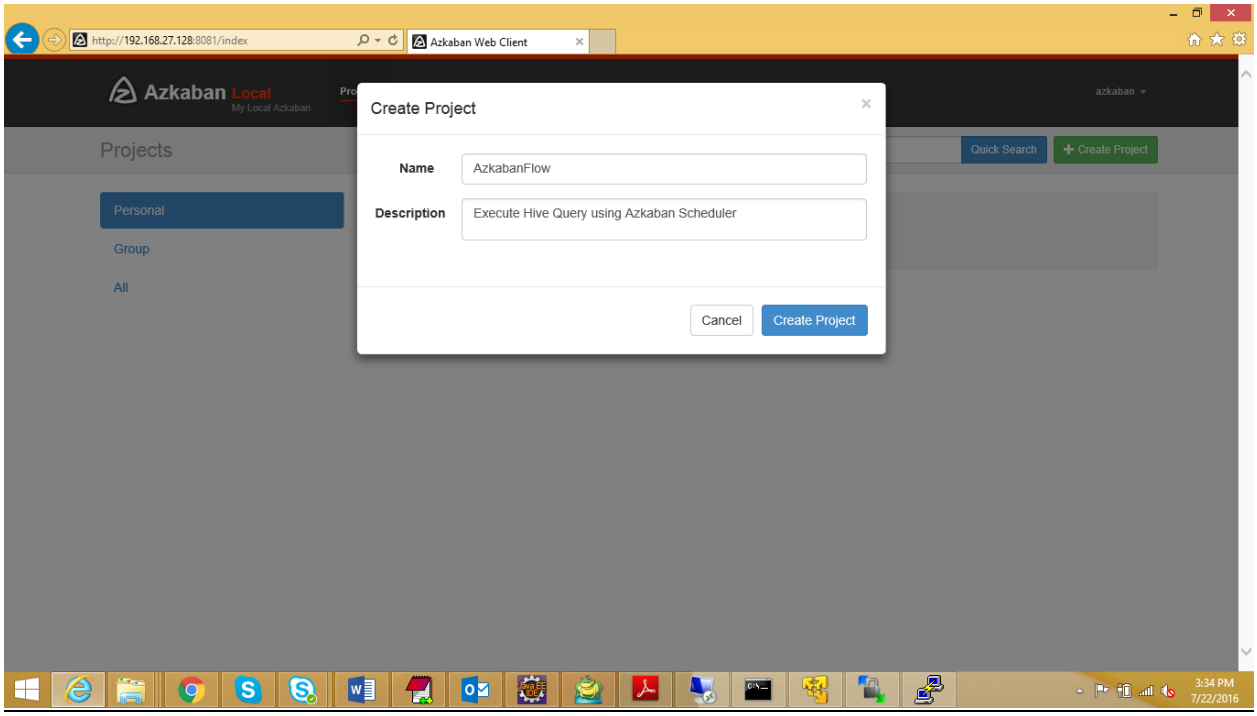
Creating Workflows in Azkaban

Hit Url: http://192.168.27.128:8081/manager?project=test

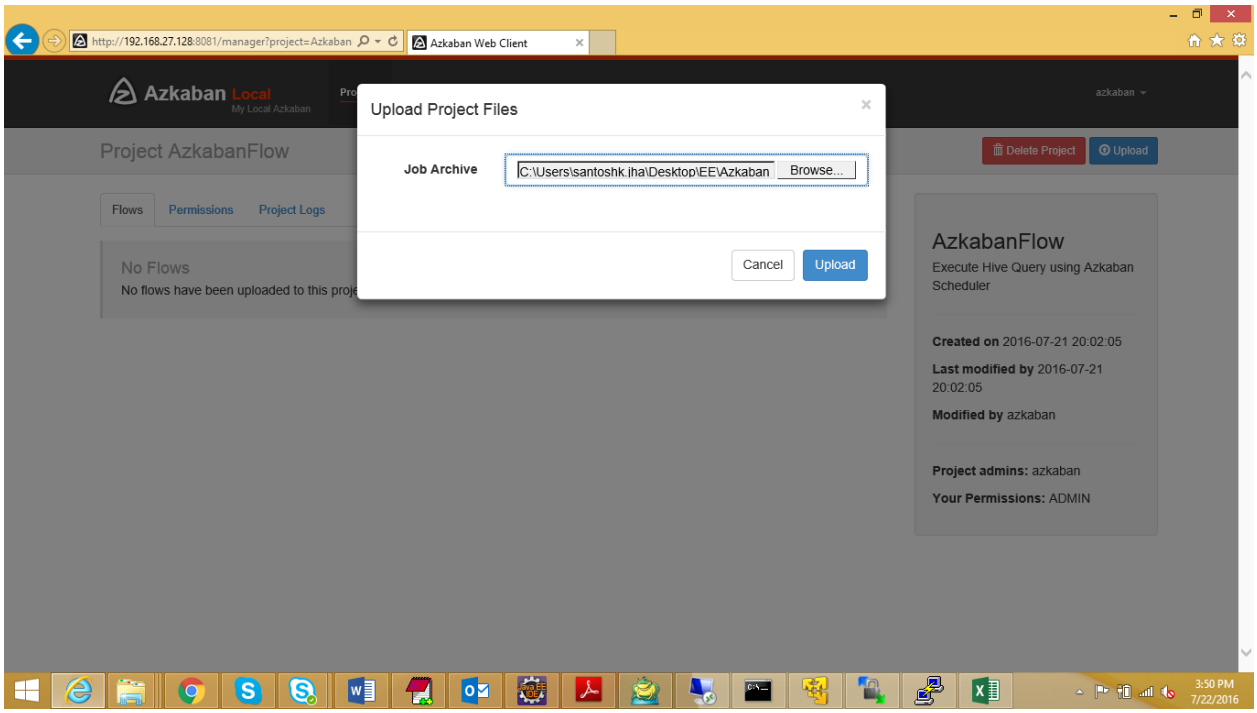
Login credential : azkaban/Azkaban



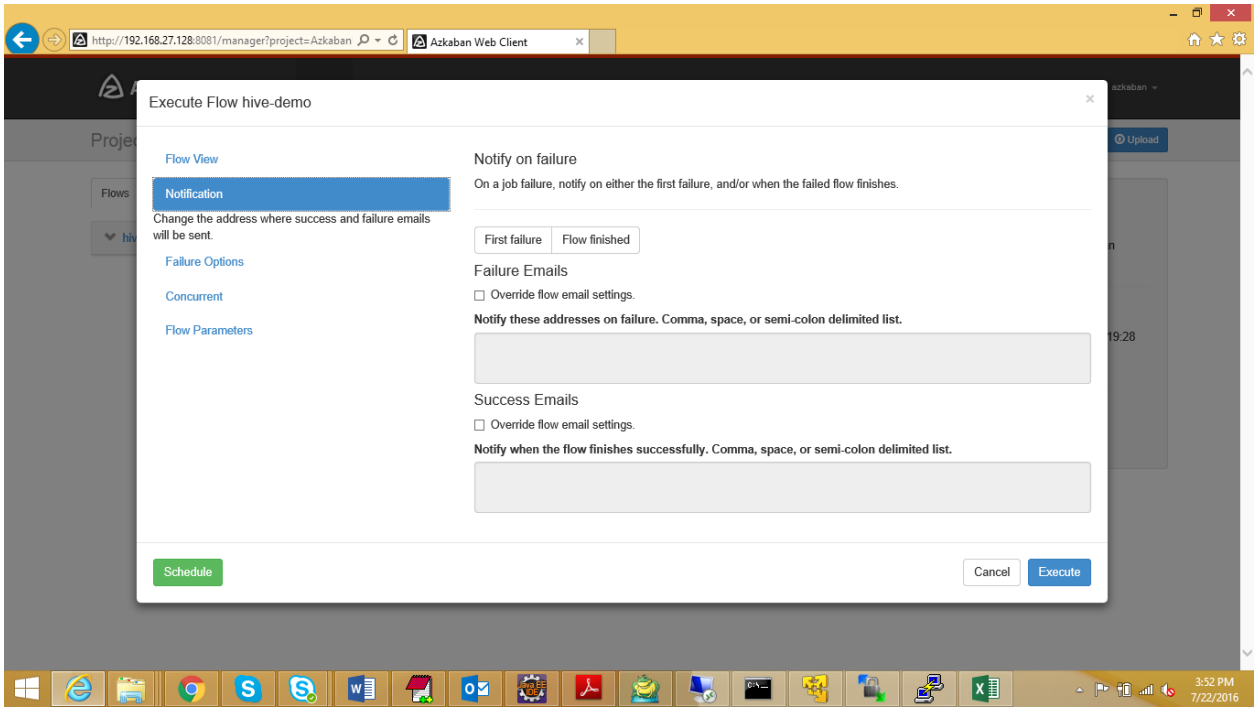
Create Project :

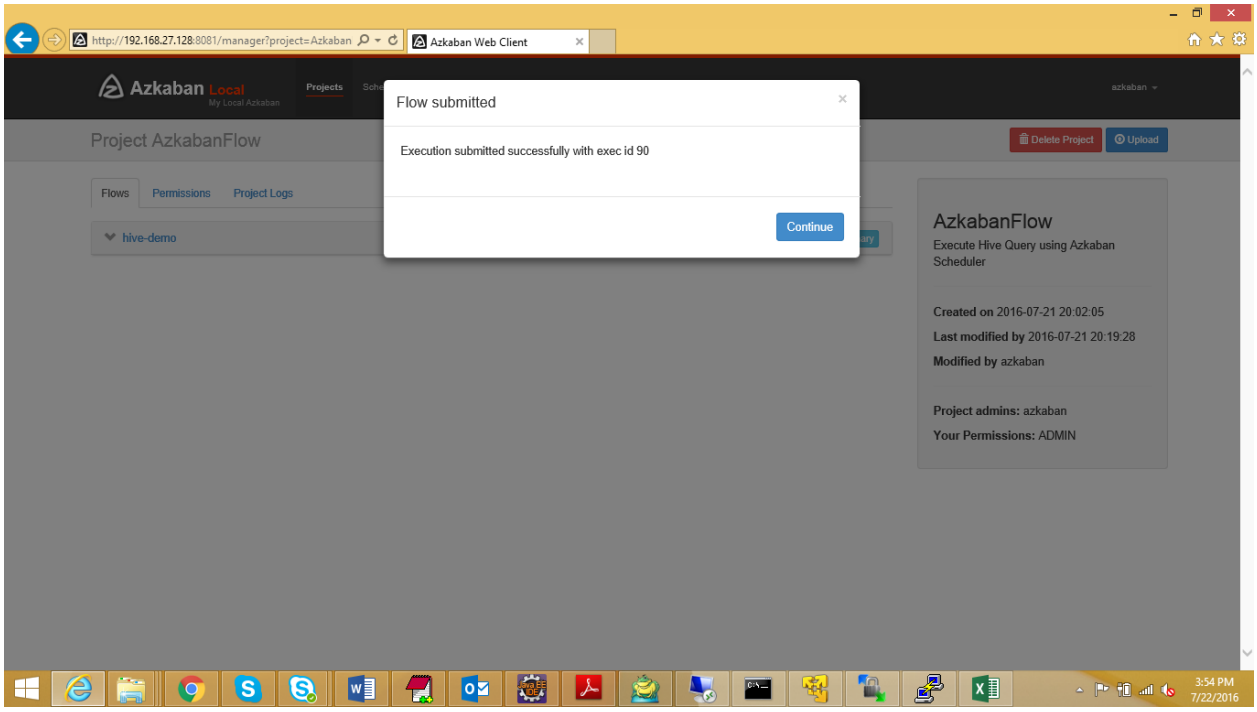


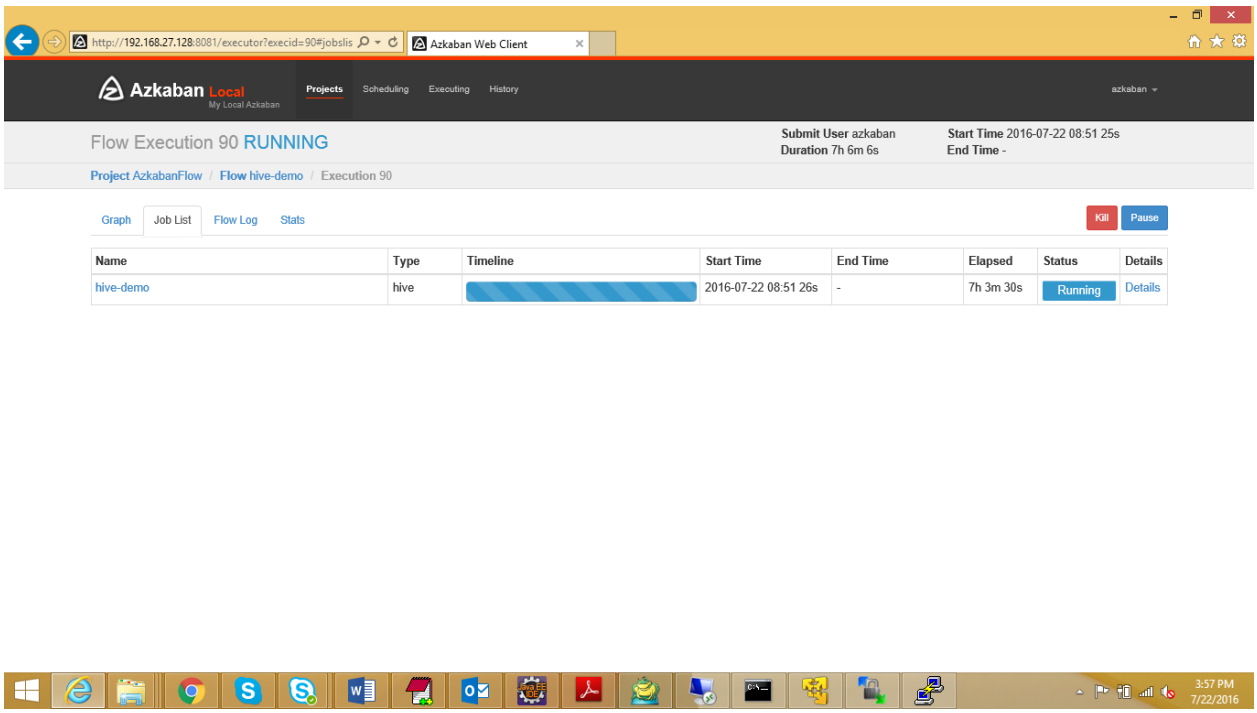
Upload Job file including hive query in zip format :

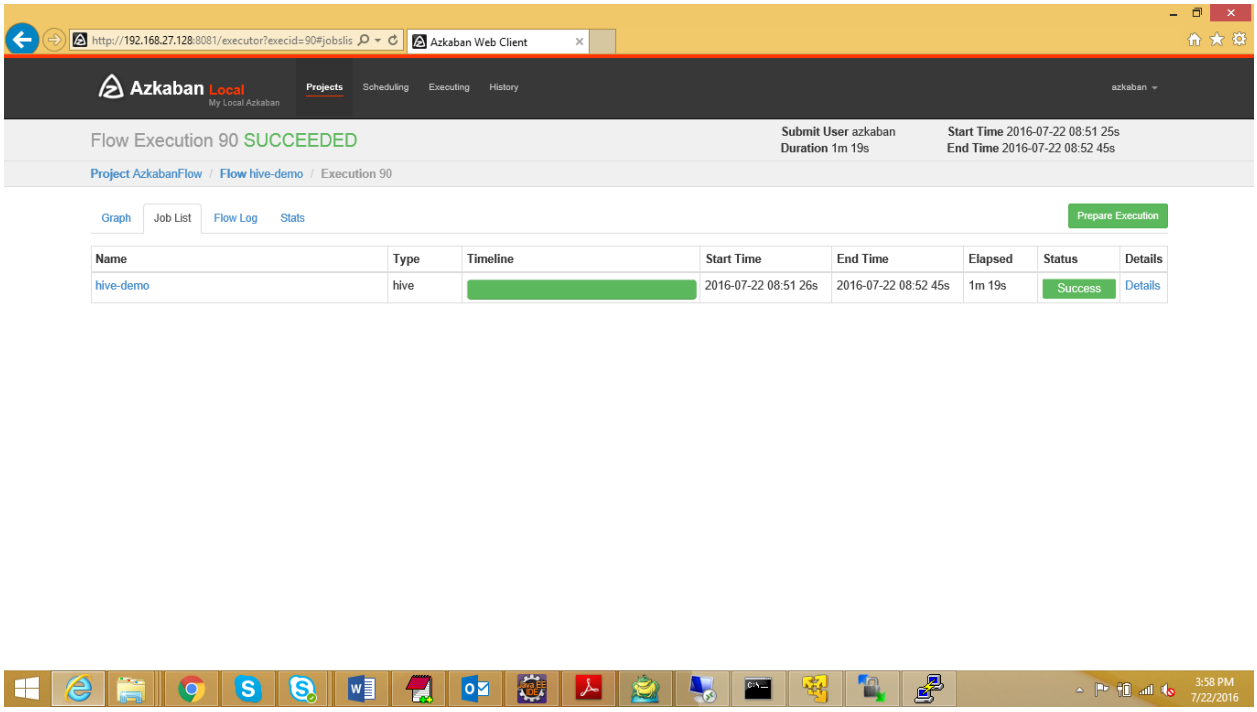


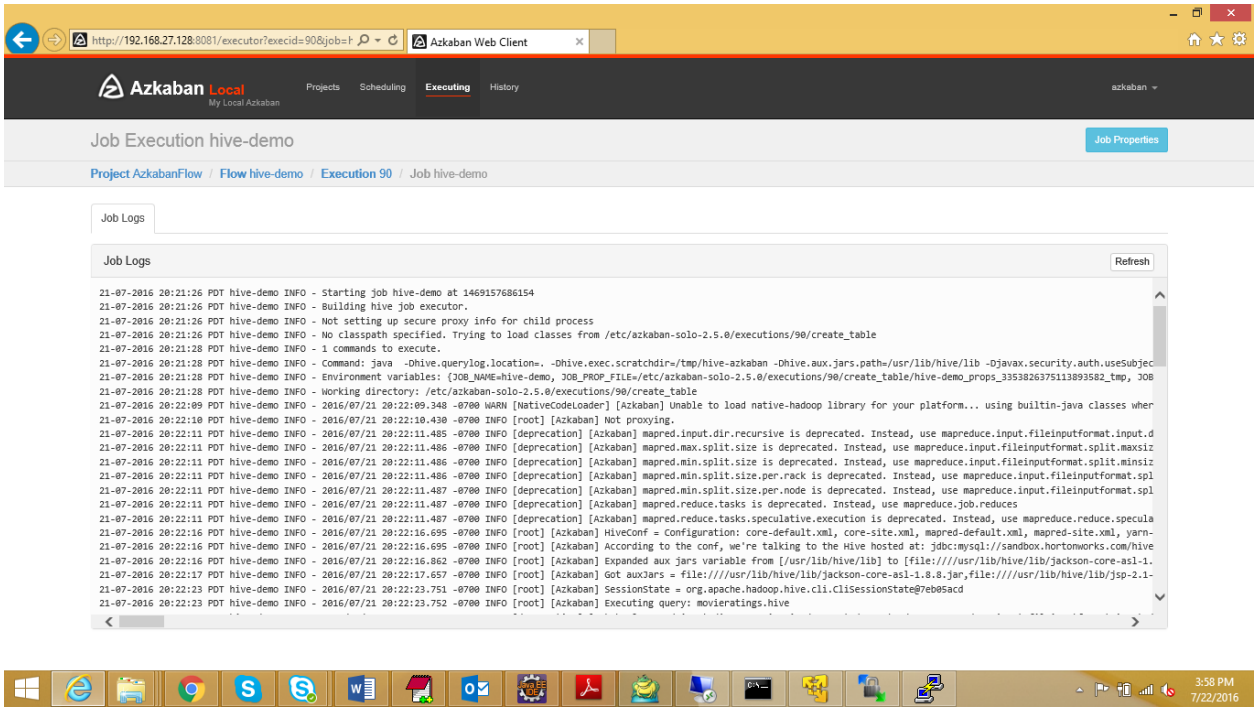
Execute or Schedule Job :






http://192.168.27.128:8081/executor?execid=90#jobslis
Azkaban Web Client
Azkaban Local
My Local Azkaban
Projects
Scheduling
Executing
History
azkaban
Flow Execution 90 RUNNING
Submit User azkaban
Duration 7h 6m 6s
Start Time 2016-07-22 08:51 25s
End Time -
Project AzkabanFlow / Flow hive-demo / Execution 90
Graph
Job List
Flow Log
Stats
Kill
Pause
Name
Type
Timeline
Start Time
End Time
Elapsed
Status
Details
hive-demo
hive
2016-07-22 08:51 26s
-
7h 3m 30s
Running
Details
3:57 PM
7/22/2016


http://192.168.27.128:8081/executor?execid=90#jobslis
Azkaban Web Client
Azkaban Local
My Local Azkaban
Projects
Scheduling
Executing
History
azkaban
Flow Execution 90 SUCCEEDED
Submit User azkaban
Duration 1m 19s
Start Time 2016-07-22 08:51 25s
End Time 2016-07-22 08:52 45s
Project AzkabanFlow / Flow hive-demo / Execution 90
Graph
Job List
Flow Log
Stats
Prepare Execution
Name
Type
Timeline
Start Time
End Time
Elapsed
Status
Details
hive-demo
hive
2016-07-22 08:51 26s
2016-07-22 08:52 45s
1m 19s
Success
Details
3:58 PM
7/22/2016


http://192.168.27.128:8081/executor?execid=90&job=h
Azkaban Web Client
Azkaban Local
My Local Azkaban
Projects
Scheduling
Executing
History
azkaban
Job Execution hive-demo
Job Properties
Project AzkabanFlow / Flow hive-demo / Execution 90 / Job hive-demo
Job Logs
Job Logs
Refresh
21-07-2016 20:21:26 PDT hive-demo INFO - Starting job hive-demo at 1469157686154
21-07-2016 20:21:26 PDT hive-demo INFO - Building hive job executor.
21-07-2016 20:21:26 PDT hive-demo INFO - Not setting up secure proxy info for child process
21-07-2016 20:21:26 PDT hive-demo INFO - No classpath specified. Trying to load classes from /etc/azkaban-solo-2.5.0/executions/90/create_table
21-07-2016 20:21:28 PDT hive-demo INFO - 1 commands to execute.
21-07-2016 20:21:28 PDT hive-demo INFO - Command: java -Dhive.querylog.location=. -Dhive.exec.scratchdir=/tmp/hive-azkaban -Dhive.aux.jars.path=/usr/lib/hive/lib -Djavax.security.auth.useSubjec
21-07-2016 20:21:28 PDT hive-demo INFO - Environment variables: {JOB_NAME=hive-demo, JOB_PROP_FILE=/etc/azkaban-solo-2.5.0/executions/90/create_table/hive-demo_props_3353826375113893582_tmp, JOB
21-07-2016 20:21:28 PDT hive-demo INFO - Working directory: /etc/azkaban-solo-2.5.0/executions/90/create_table
21-07-2016 20:22:09 PDT hive-demo INFO - 2016/07/21 20:22:09.348 -0700 WARN [NativeCodeLoader] [Azkaban] Unable to load native-hadoop library for your platform... using builtin-java classes wher
21-07-2016 20:22:10 PDT hive-demo INFO - 2016/07/21 20:22:10.430 -0700 INFO [root] [Azkaban] Not proxying.
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.485 -0700 INFO [deprecation] [Azkaban] mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.d
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.486 -0700 INFO [deprecation] [Azkaban] mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsiz
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.486 -0700 INFO [deprecation] [Azkaban] mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsiz
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.486 -0700 INFO [deprecation] [Azkaban] mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.spl
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.487 -0700 INFO [deprecation] [Azkaban] mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.spl
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.487 -0700 INFO [deprecation] [Azkaban] mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
21-07-2016 20:22:11 PDT hive-demo INFO - 2016/07/21 20:22:11.487 -0700 INFO [deprecation] [Azkaban] mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.specula
21-07-2016 20:22:16 PDT hive-demo INFO - 2016/07/21 20:22:16.695 -0700 INFO [root] [Azkaban] HiveConf = Configuration: core-default.xml, core-site.xml, mapred-default.xml, mapred-site.xml, yarn-
21-07-2016 20:22:16 PDT hive-demo INFO - 2016/07/21 20:22:16.695 -0700 INFO [root] [Azkaban] According to the conf, we're talking to the Hive hosted at: jdbc:mysql://sandbox.hortonworks.com/hive
21-07-2016 20:22:16 PDT hive-demo INFO - 2016/07/21 20:22:16.862 -0700 INFO [root] [Azkaban] Expanded aux jars variable from [/usr/lib/hive/lib] to [file:////usr/lib/hive/lib/jackson-core-asl-1.
21-07-2016 20:22:17 PDT hive-demo INFO - 2016/07/21 20:22:17.657 -0700 INFO [root] [Azkaban] Got auxJars = file:////usr/lib/hive/lib/jackson-core-asl-1.8.8.jar,file:////usr/lib/hive/lib/jsp-2.1-
21-07-2016 20:22:23 PDT hive-demo INFO - 2016/07/21 20:22:23.751 -0700 INFO [root] [Azkaban] SessionState = org.apache.hadoop.hive.cli.CliSessionState@7eb05acd
21-07-2016 20:22:23 PDT hive-demo INFO - 2016/07/21 20:22:23.752 -0700 INFO [root] [Azkaban] Executing query: movieratings.hive
3:58 PM
7/22/2016