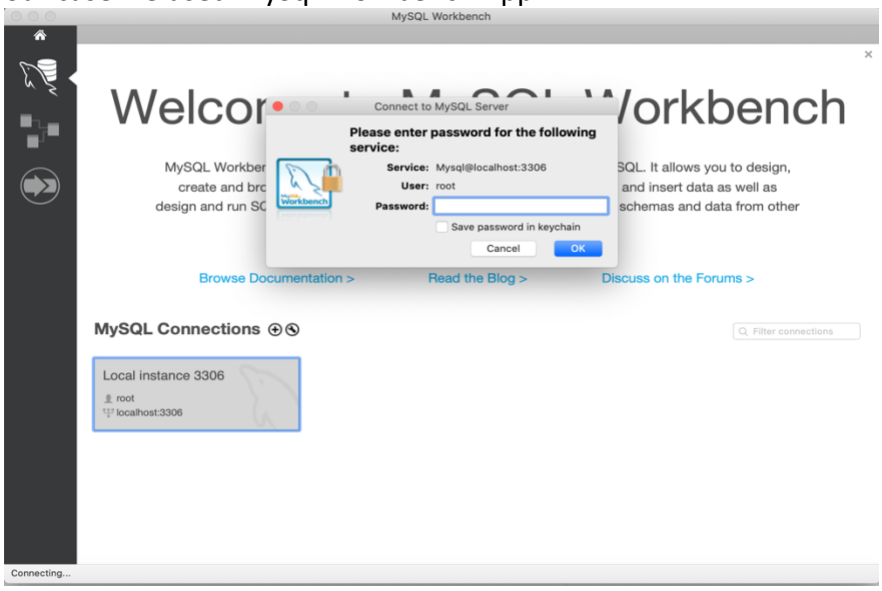
Miners Team
Loans Repayment Use Case
This is a “how to” guide to install, configure and run the use case correctly

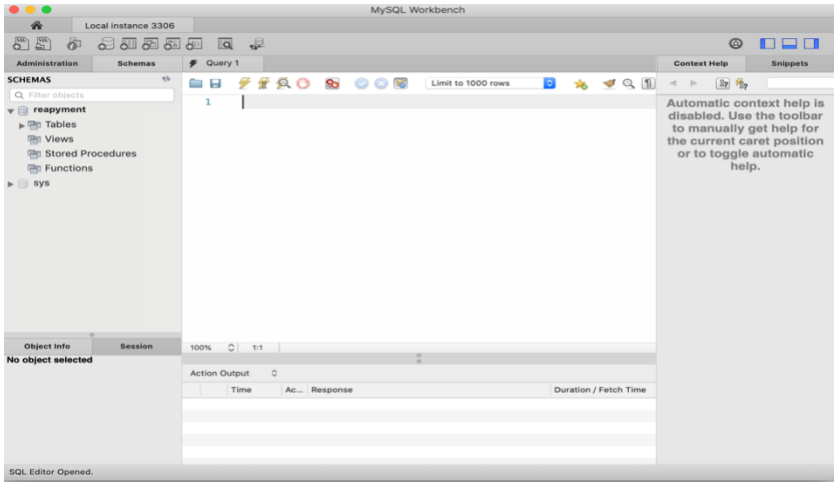
Important tools

1- Install Python and make sure its environment variable is set correctly.
2- Install Java and be sure to set JAVA_PATH environment variable
3- install Hadoop on your machine and HDFS and configure them correctly and start hdfs
    a. We followed the link below for installation and configuration of our machine
        i. https://gist.github.com/viecode09/ad56b09bea4da59b4240d45b666321cf
    b. Create hdfs directory with name “repayment” using this command “hdfs dfs -mkdir /repayment”

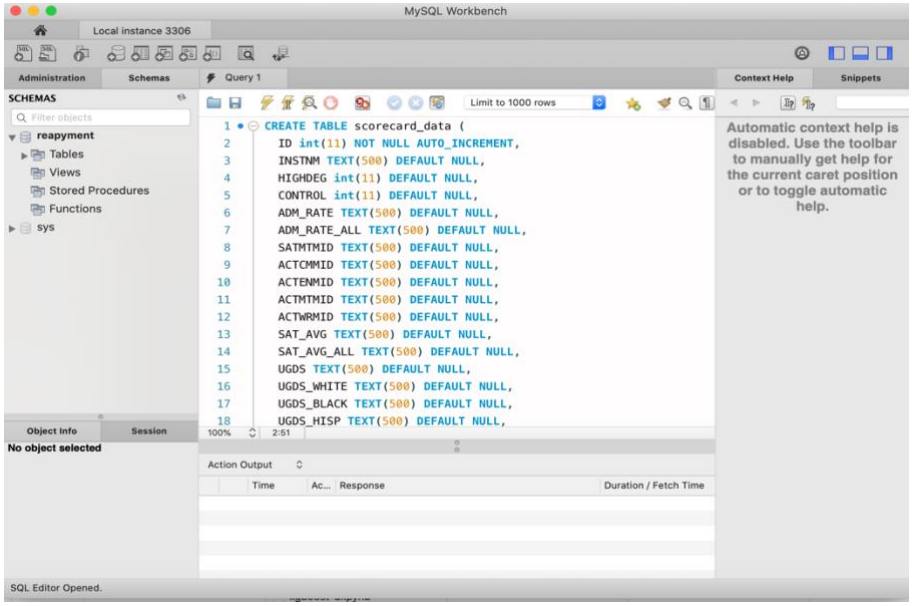
4- Install Spark
    a. We followed the link below for installation and configuration of our machine
        i. https://www.tutorialkart.com/apache-spark/how-to-install-spark-on-mac-os/
5- install MySql (just download the binary and install it)
6- Configure database
    - First we used ETL tool to import excel files to SQL engine. To make our work reproducible, we generated the needed queries to get exact image of our database. Please follow the steps below to get exact image from our database:
        1- Open MySql from any client application and connect to database server. In our case we used MySql Workbench App



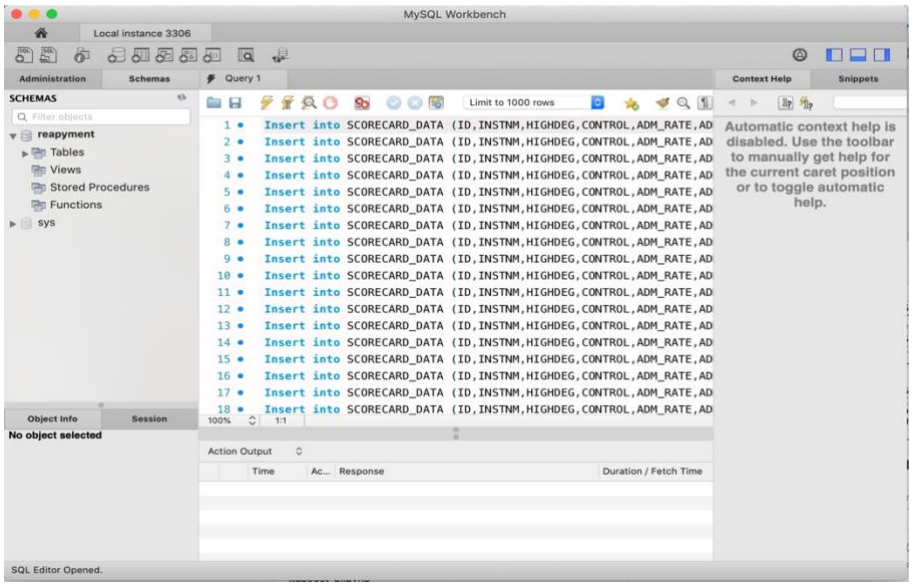
        2- create a database with name “repayment”



3- Use the sql script that is in create_mysql_table_script.txt file to create table with the same name we use in our workflow



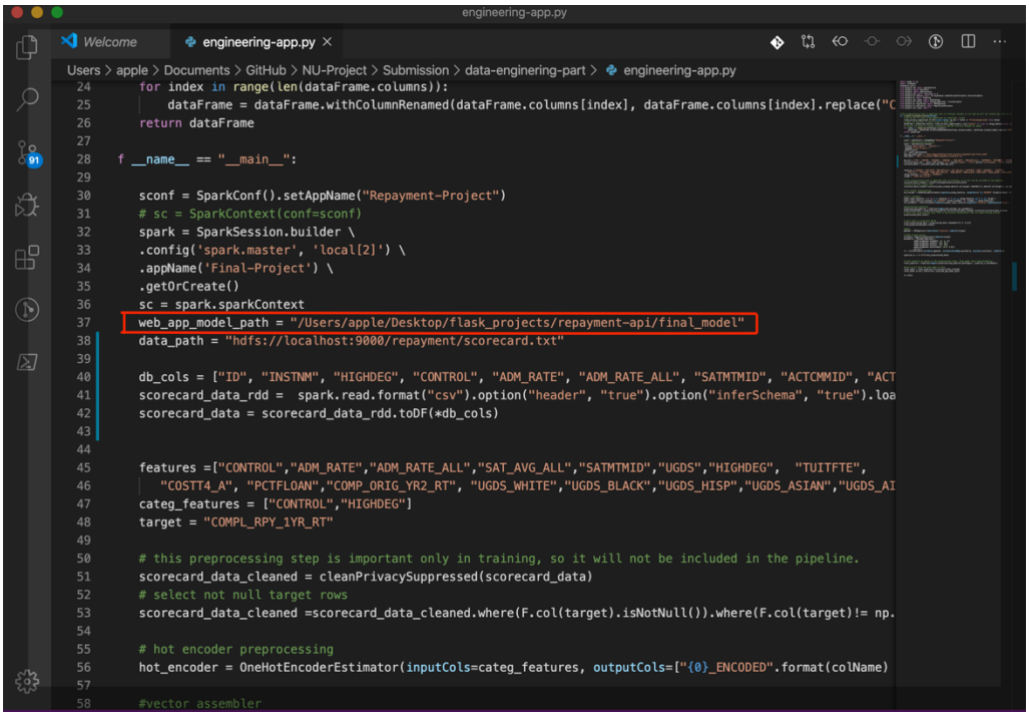
4- Use the sql script that is in scorecard_data_insertion.txt to import data the data to this table



7- Install Flask Server

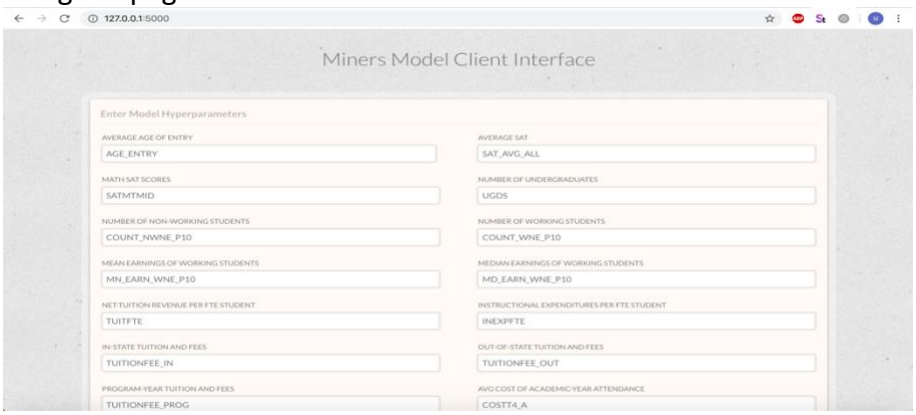
- We followed this link https://gist.github.com/dineshviswanath/af72af0ae2031cd9949f
- Then copy all files from end-user-app/ repayment-api to your installation directory. This is how our installation directory looks like after copying

- Some files point to this directory as part of their work. Flask listens on final_model subdirectory to detect changes automatically and redeploy the new model. Also the training script save the trained model to this directory (final_model), so you have to change them to your new directory.
- Open data-enginering-part/engineering-app.py file, and change the directory to your new directory

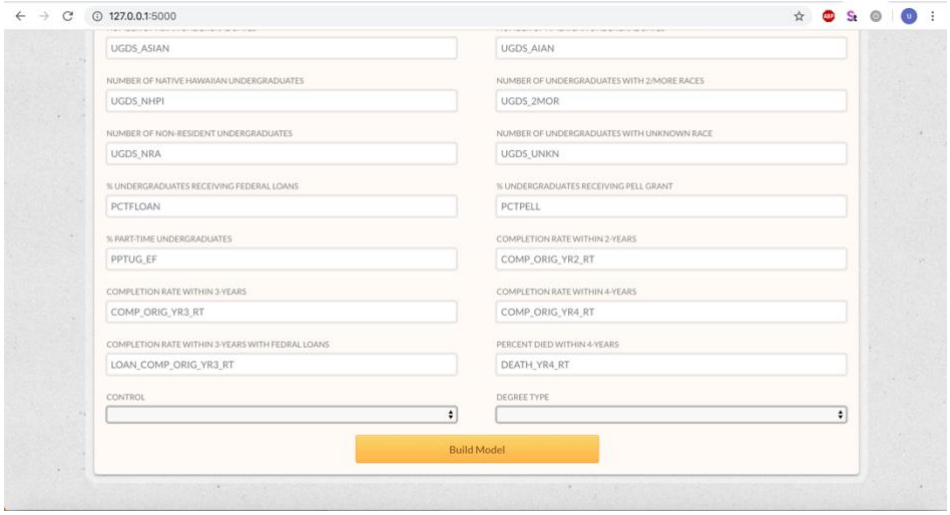


- Also don't forget to do the same step above for Open data-enginering-part/engineering-notebook.py if you want to check the engineering steps using jupyter notebook file.

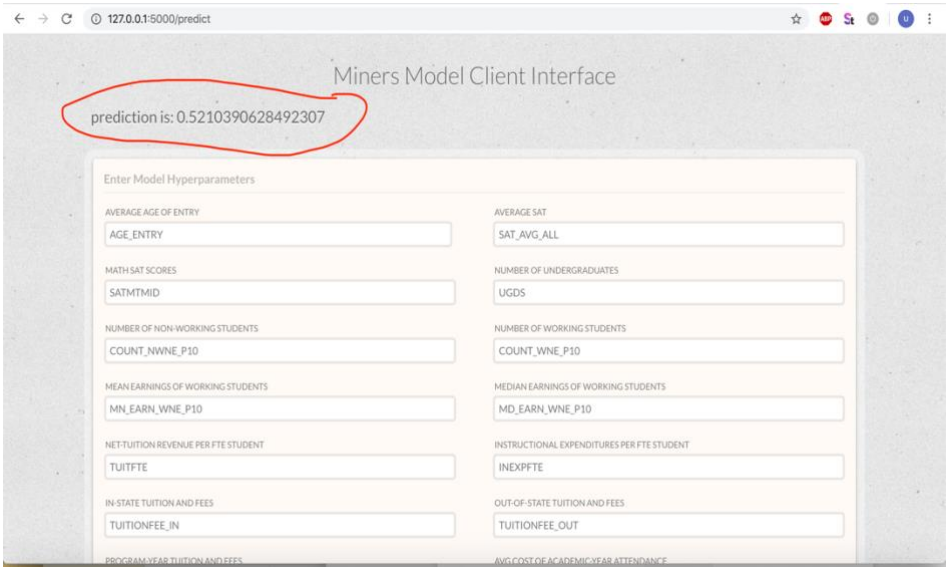
8- Start Flask Server
   a. From cmd, goto installation path of your flask project (the path where you put end-user-app/repayment-api files in)
   b. Type "source bin/activate". Then "python flaskApp.py"
   c. Now, open the browser and type http://127.0.0.1:5000/
   d. You should get a page like this



e. Fill the form and press “Build Model”



f. The, you get a prediction for the institute with the provided information



9- Install Azkaban Scheduler

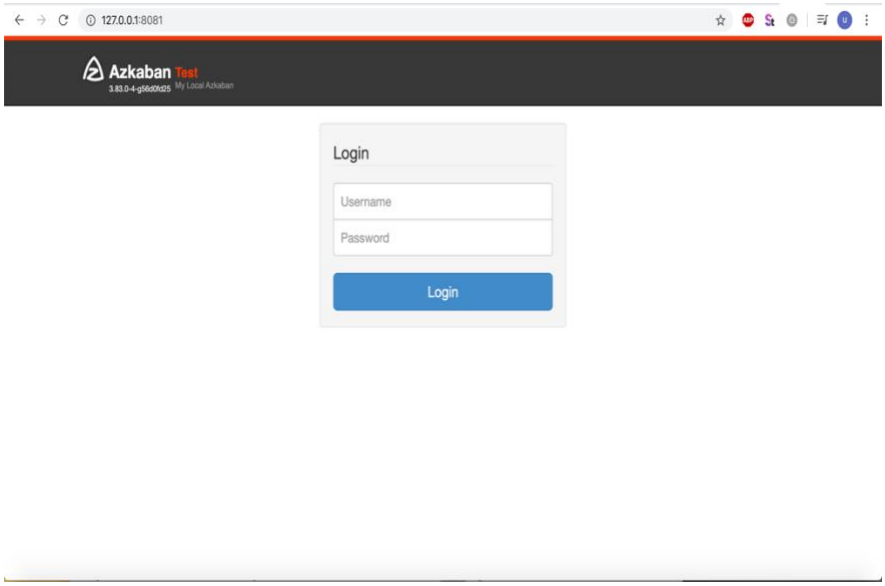
a. Clone the githube repository of this link

i. https://github.com/azkaban/azkaban

ii. From terminal, go to the directory of the cloned github project above and type ./gradlew build

iii. Start Azkaban server by going to azkaban-solo-server/build/install/azkaban-solo-server from cmd and type bin/start-solo.sh

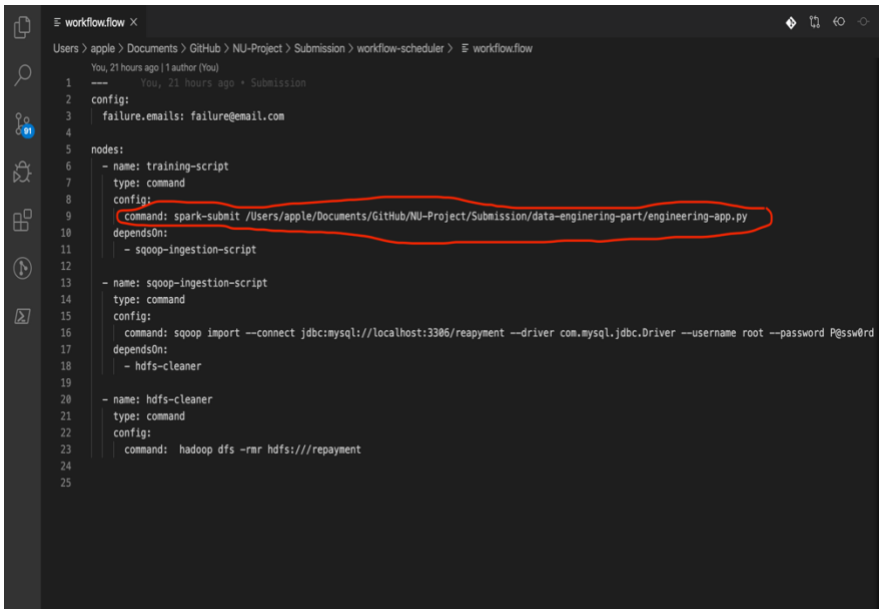
iv. Now you can start the GUI by typing in the web browser http://127.0.0.1/8081



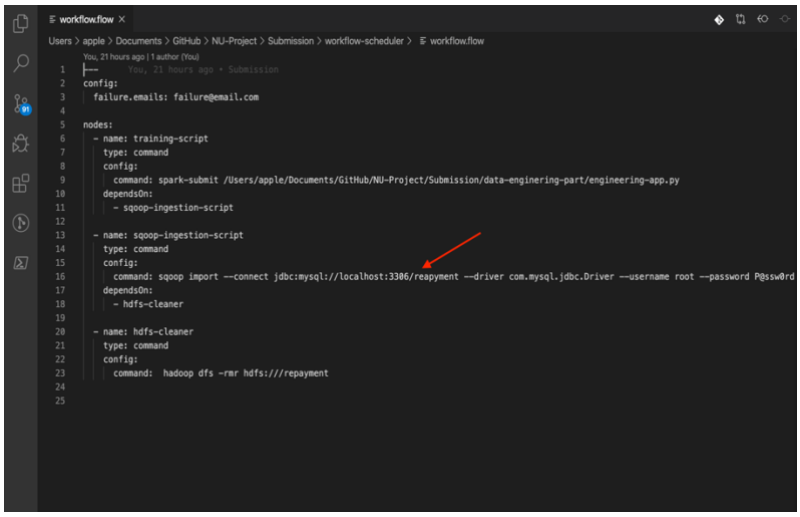
v. The default username and password are “azkaban”

vi. Open workflow-scheduler/workflow.flow and edit the following

1. Edit the path of data-enginering-part/engineering-app.py to your path on your machine

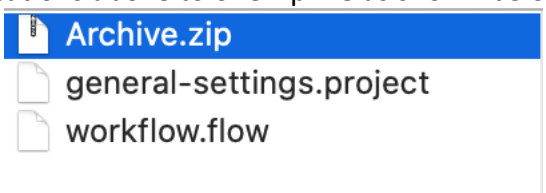


2. Configure sqoop script in the same file with your system configuration

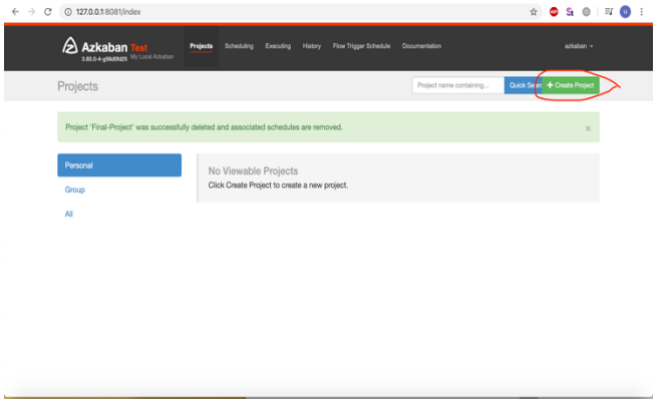
   a. 

   b. The changes you must do to the script are
      i. Change “import --connect jdbc:mysql://localhost:3306/reapyment” to your Mysql port instead of 3306
      ii. Change username and password of database to yours “--username root --password P@ssw0rd”
      iii. Download SQL connector jar file from internet and put it in sqoop lib folder and mention this folder in “—bindir your-sqoop-lib-path-folder” option instead of current configuration “--bindir /usr/local/Cellar/sqoop/1.4.6_1/libexec/lib”

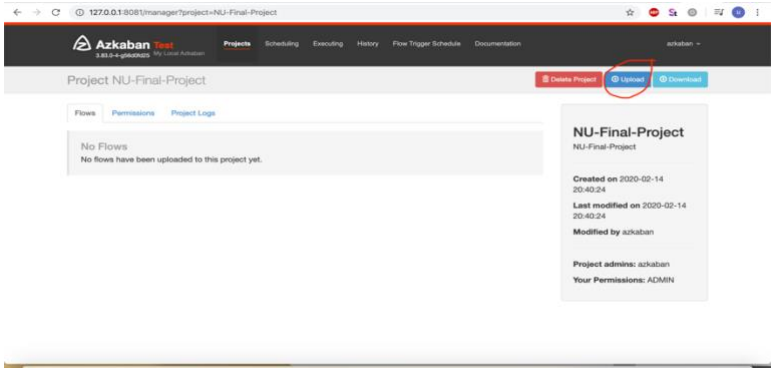
3. Compress/Archive workflow-scheduler/general-settings.project file and workflow-scheduler/workflow.flow file after modifications above to one zip file as shown below

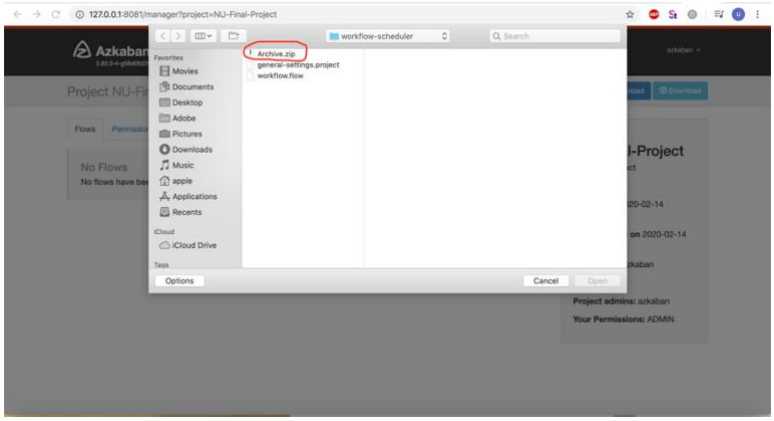


4. Create New Project from GUI

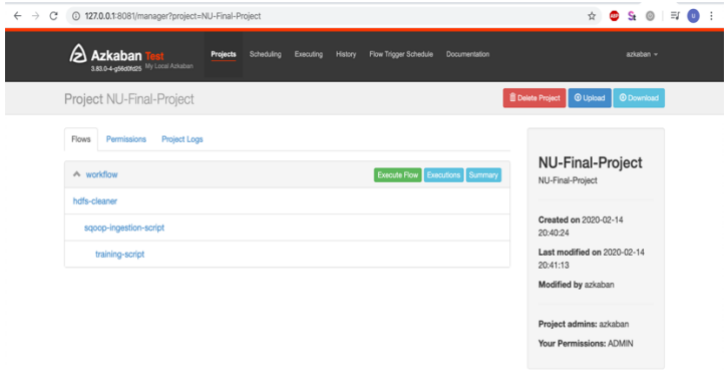


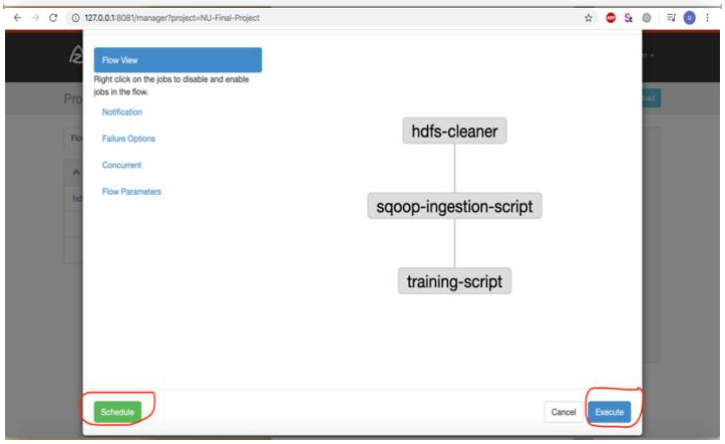
5. Upload archived file to Azkaban server





6. Execute/Schedule workflow from “Execute Flow” button





7. Monitor Working Jobs