

LAB: Experiment Management

Prerequisites

- LAB: Spark and Data Prep

Recommended

- LAB: Working with Signals
- LAB: Building a Query Intent Classifier

Getting Started

This lab assumes that you are using an AWS virtual machine provided by Lucidworks Training. If this is not the case, your filepaths and IP addresses will vary significantly from those shown.

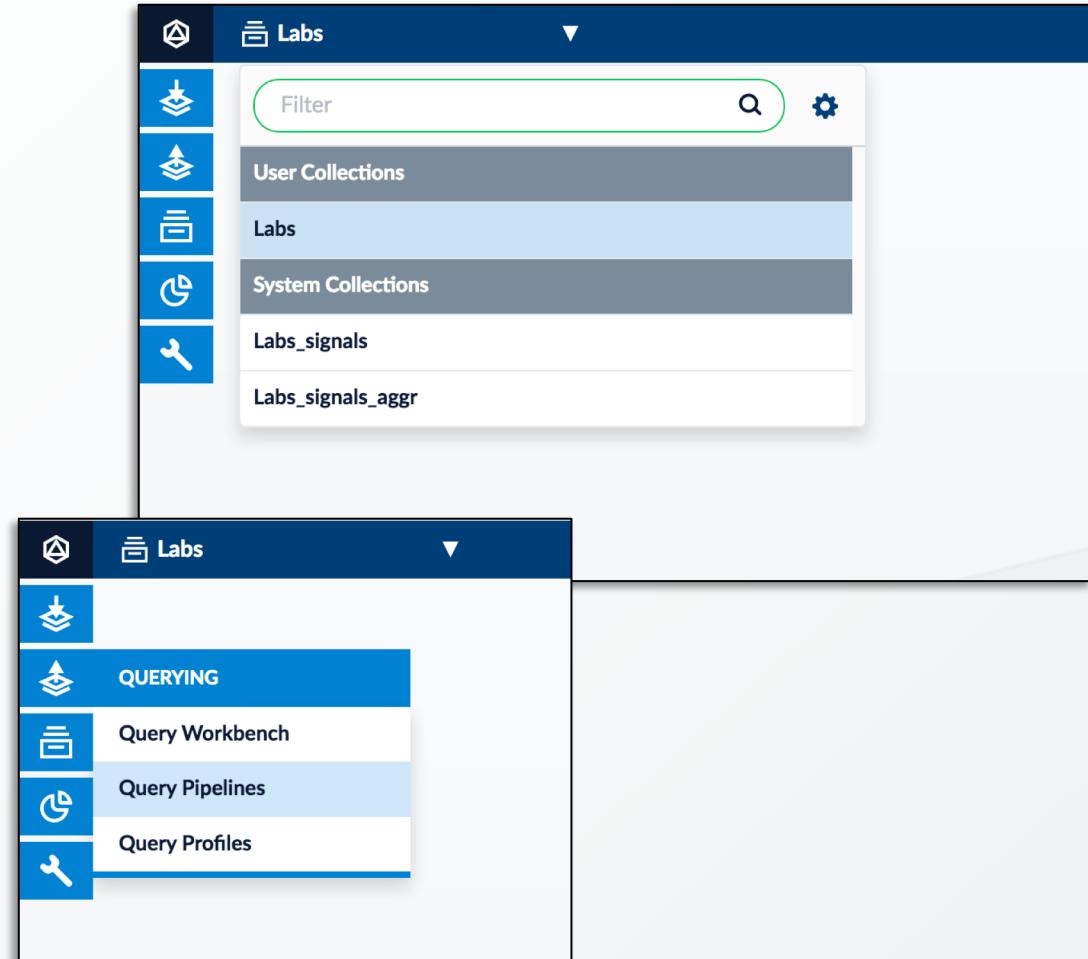
- In a bash/shell terminal, start Fusion

```
./fusion/4.0.1/bin/fusion start
```

- In a web browser, open Fusion Admin

```
<your-vm-ip>:8764
```

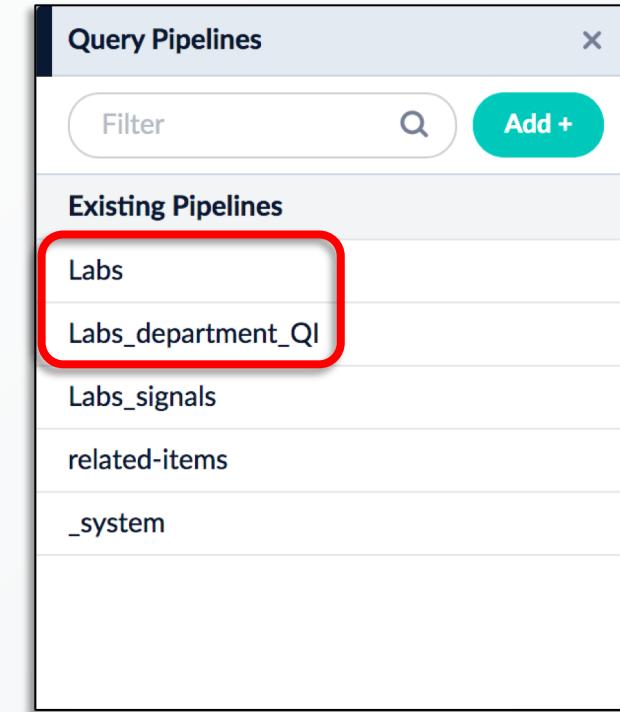
- Enter your username and password
*The default is **admin** and **Lucidworks1***
- Click into the **Labs** Fusion App
- In the top left dropdown, change to the **Labs** collection
- In the left side menu, go to **QUERYING > Query Pipelines**



If you completed the lab **Building a Query Intent Classifier**, you will have two main query pipelines available: **Labs** and **Labs_department_QI**. In this lab, we will experimentally compare the clickthrough rate for three different pipelines: the base pipeline, the base pipeline plus signal boosting, and the **Query Intent Classifier** pipeline.

If you did not complete the lab **Building a Query Intent Classifier**, you will see only the pipeline **Labs**. In this case, we will compare only two pipelines: the base pipeline, and the base pipeline plus signal boosting.

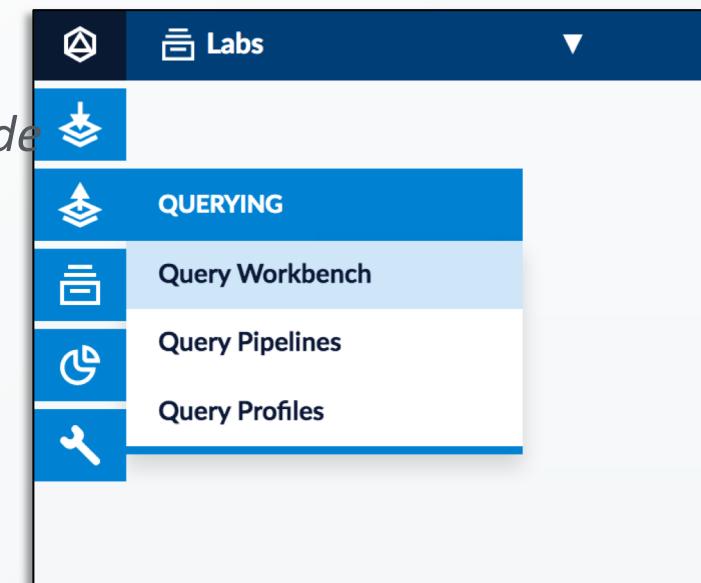
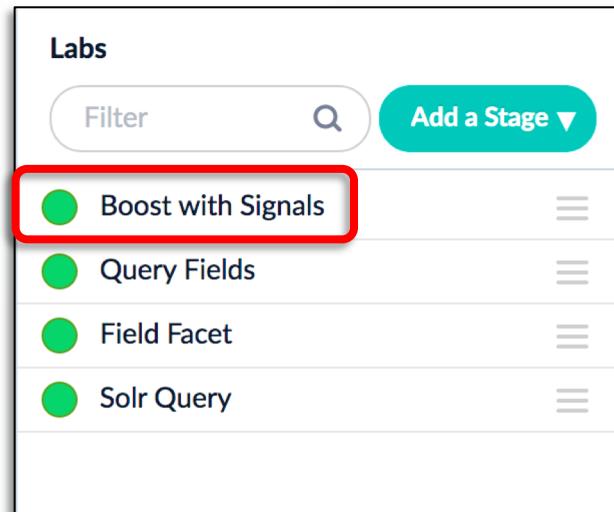
First, we need to set up all of the pipelines to be tested.



Creating the Pipeline Labs_no_signals

- In the left side menu, go to **QUERYING > Query Workbench**

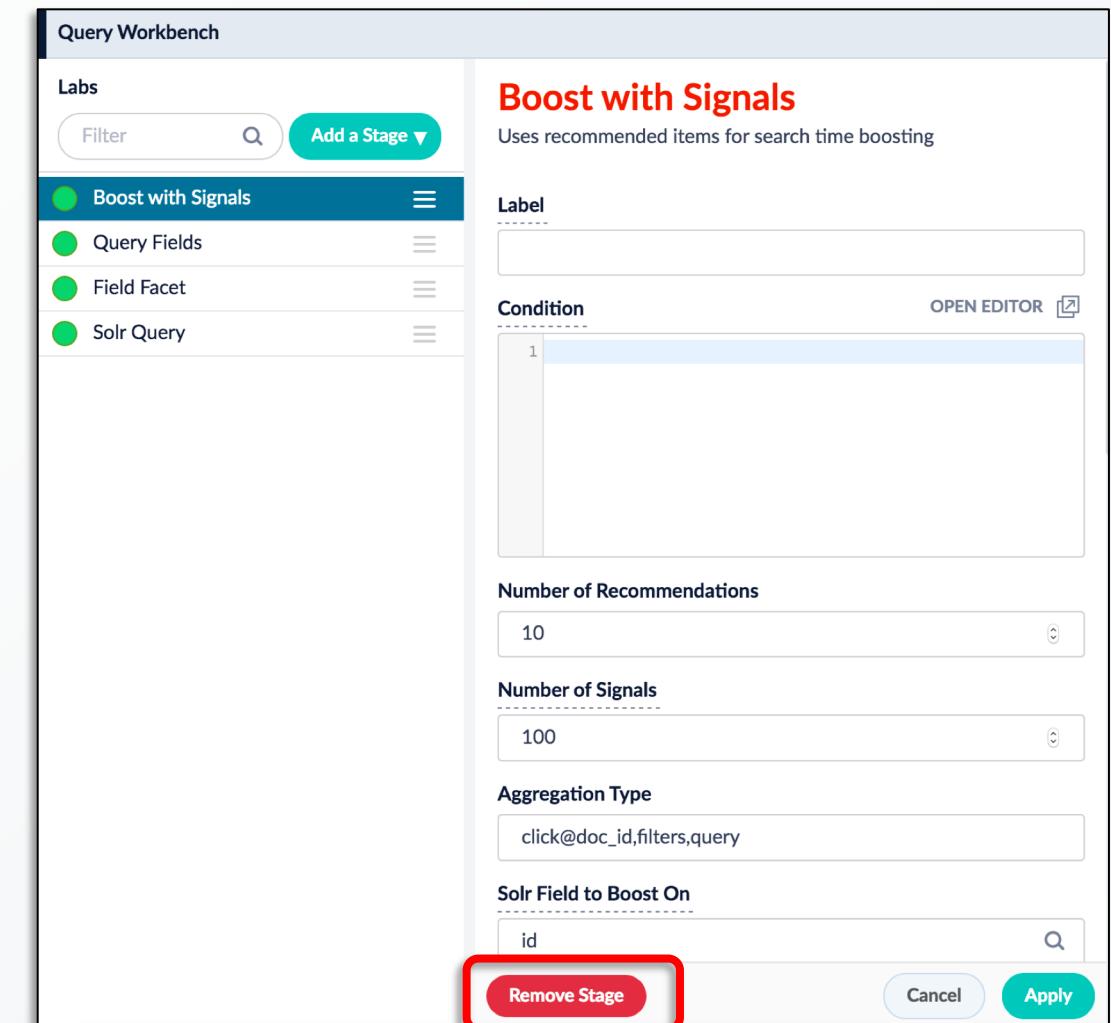
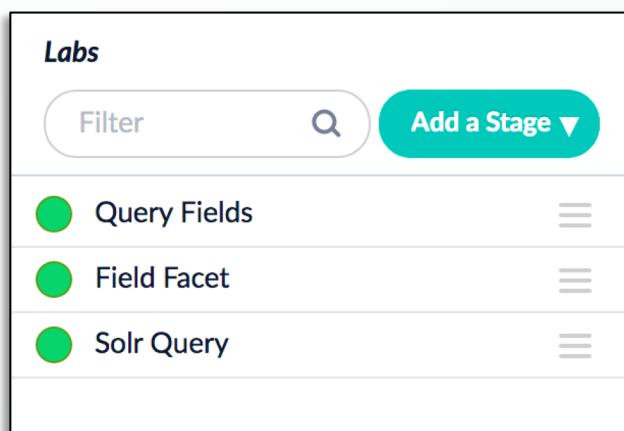
Note that the default Labs pipeline already has signal boosting enabled. We will create a copy of this pipeline that does not include signal boosting.



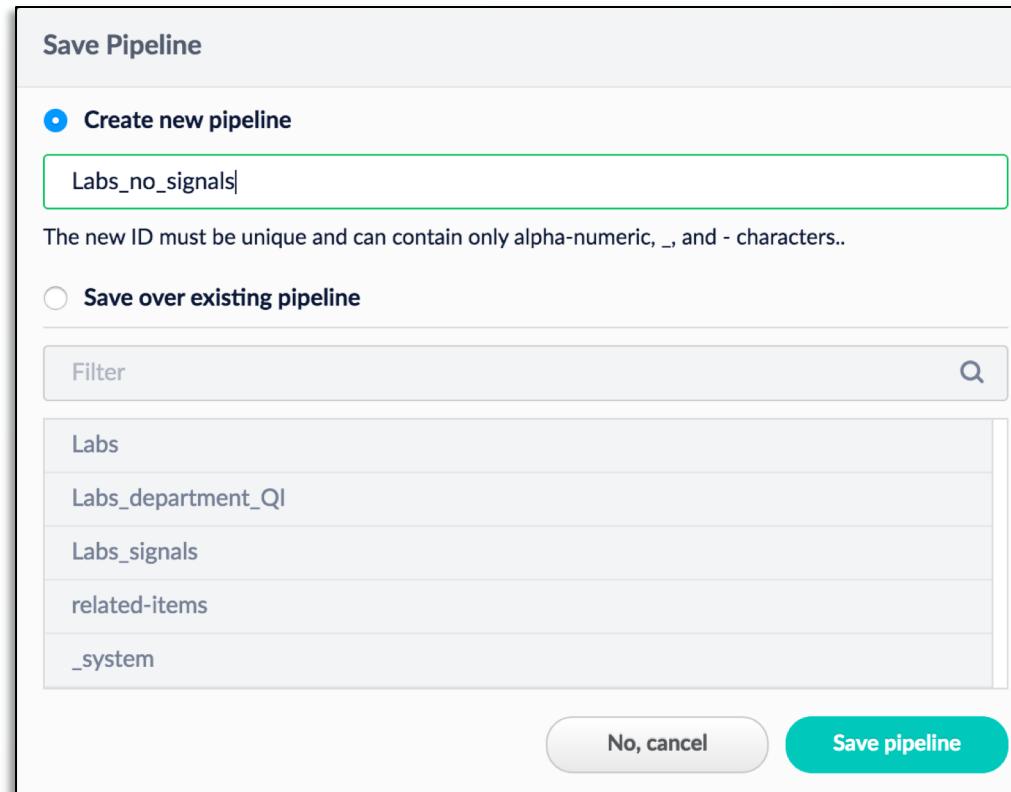
Creating the Pipeline Labs_no_signals

- Select the **Boost with Signals** stage
- At the bottom of the Stage Editor pane, click **Remove Stage**

The finished pipeline should have three stages



- At the top right of the Query Workbench panel, click **Save**



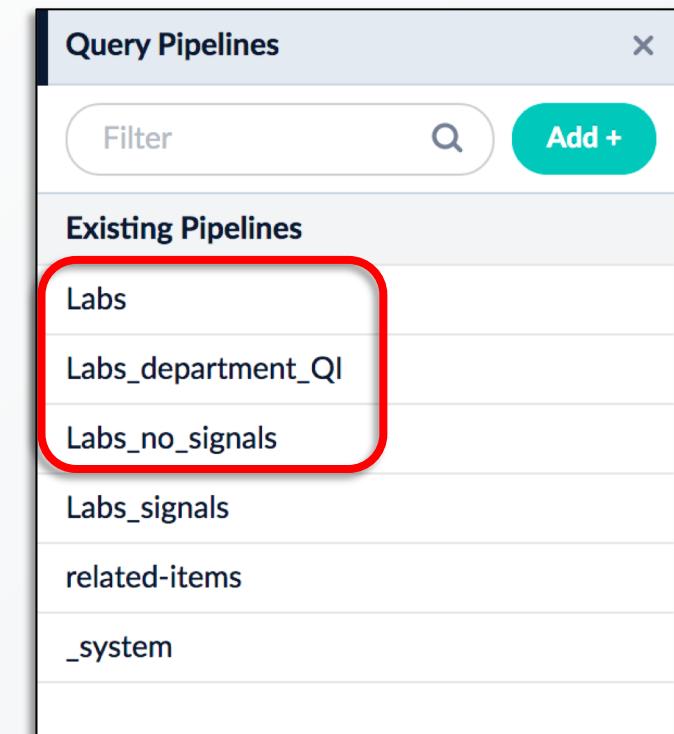
- Select **Create new pipeline**
- Name the pipeline **Labs_no_signals**
- Click **Save pipeline**

- In the left side menu, go to **QUERYING > Query Pipeline**

You should now see both *Labs* and *Labs_no_signals* in your pipeline set.

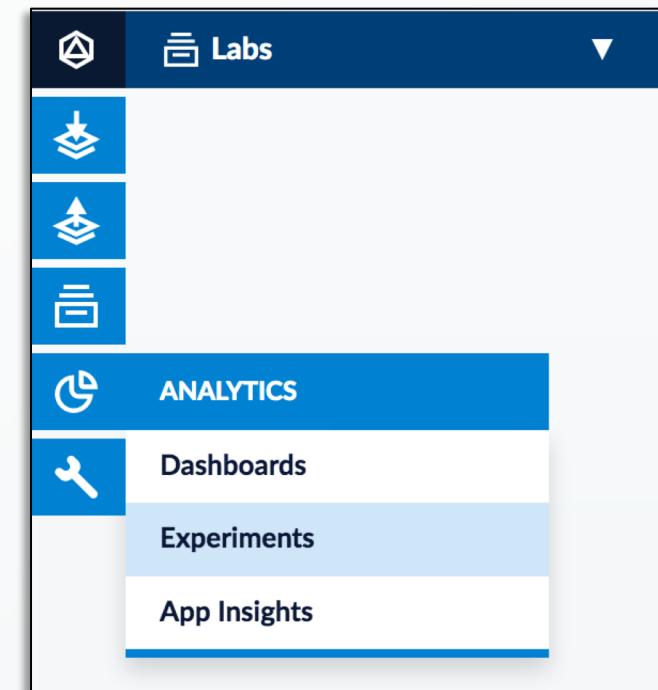
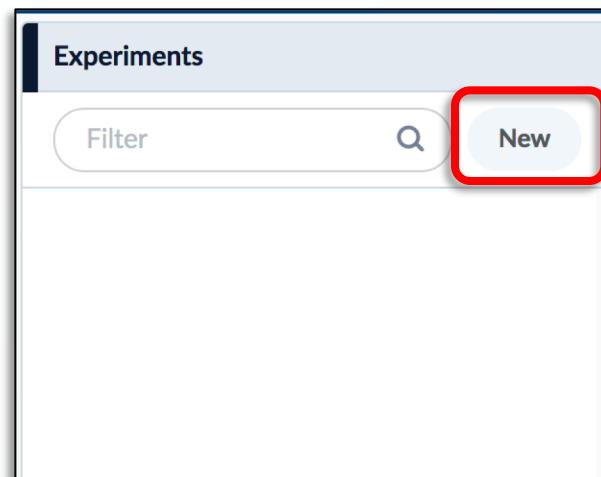
Labs_department_QI was created in the lab **Building a Query Intent Classifier**. It is recommended to do that lab first, but not strictly necessary.

We will now set up an experiment that compares the "success" rates of these two (or three) pipelines.



Setting Up an Experiment

- In the left side menu of Fusion Admin, go to **ANALYTICS > Experiments**
- Click **New**



- Fill out the experiment parameters according to the following table:

Parameter	Value	Explanation
ID	Baseline_vs_Signals_vs_QI	<i>Unique name for this experiment.</i>
Unique ID parameter	userId	<i>Field in the query object that will be used to direct traffic within the experiment. Routing based on userId guarantees that the same user will get the same experience every time.</i>
Base collection for signals	Labs	<i>All signals generated by the experiment will be sent to the signals collection associated with the Labs collection—Labs_signals</i>

*Every experiment has some number of **variants**, representing the different cases to be tested. In this case, we are testing three (or two) pipelines, and so will have three (or two) variants.*

- At the bottom of the page, click **Add Variant** until you have three variants

VARIANTS
Specify what varies in this variant, and optionally change the traffic weight

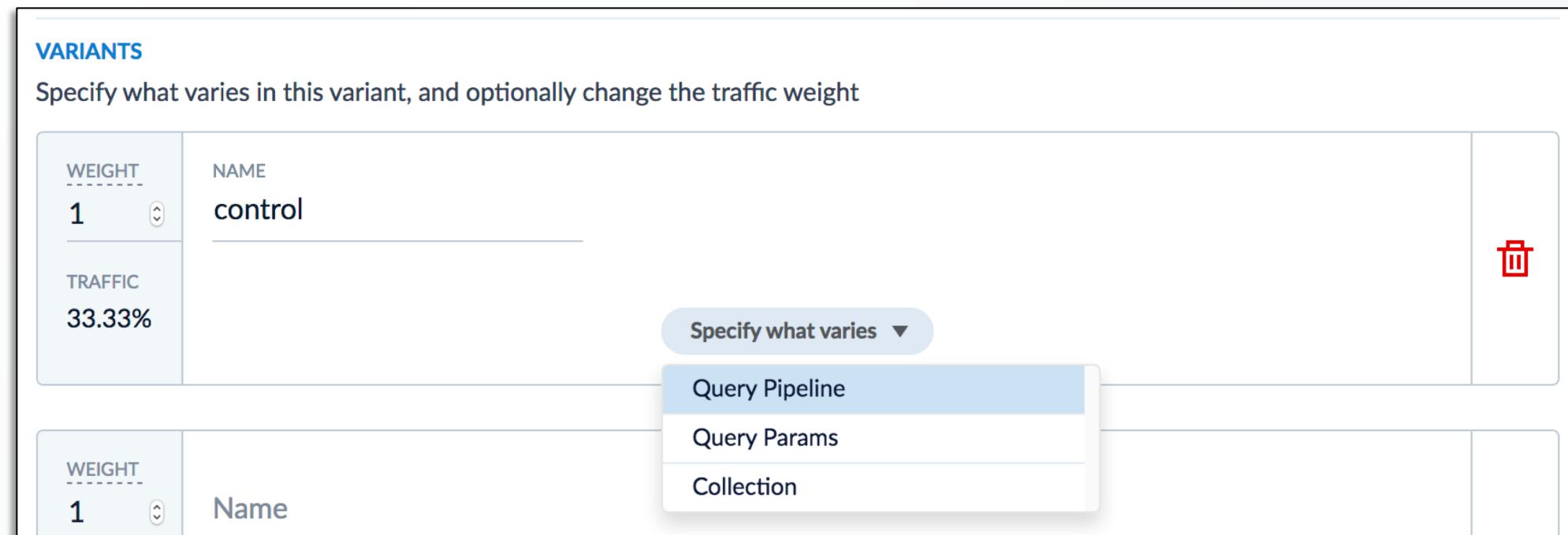
WEIGHT 1	NAME control
TRAFFIC 33.33%	Specify what varies ▾

WEIGHT 1	Name
TRAFFIC 33.33%	Specify what varies ▾

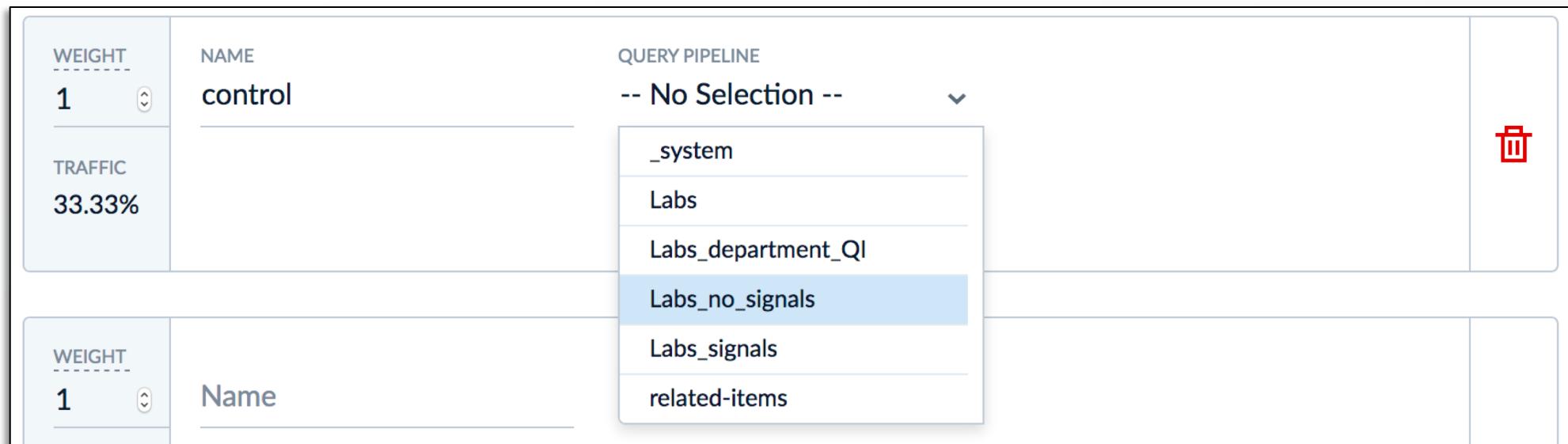
WEIGHT 1	Name
TRAFFIC 33.33%	

[Add Variant](#)

- In the first variant (**control**) click **Specify what varies**
- Choose **Query Pipeline**



- Set the Query Pipeline to **Labs_no_signals**



This is our control variant—we are using it to set a baseline for comparison for Signals and Query Intent. Hence, the pipeline with no signal boosting.

- Name the second variant **control_plus_signals**
- Click **Select what varies > Query Pipeline**
- Set the query pipeline to **Labs**

WEIGHT 1	NAME control_plus_signals	QUERY PIPELINE Labs	
TRAFFIC 33.33%		Specify what varies ▾	

*Recall that the default Labs pipeline includes the stage **Boost with Signals**.*

- Name the third variant **control_plus_qi**
- Click **Select what varies > Query Pipeline**
- Set the query pipeline to **Labs_department_QI**

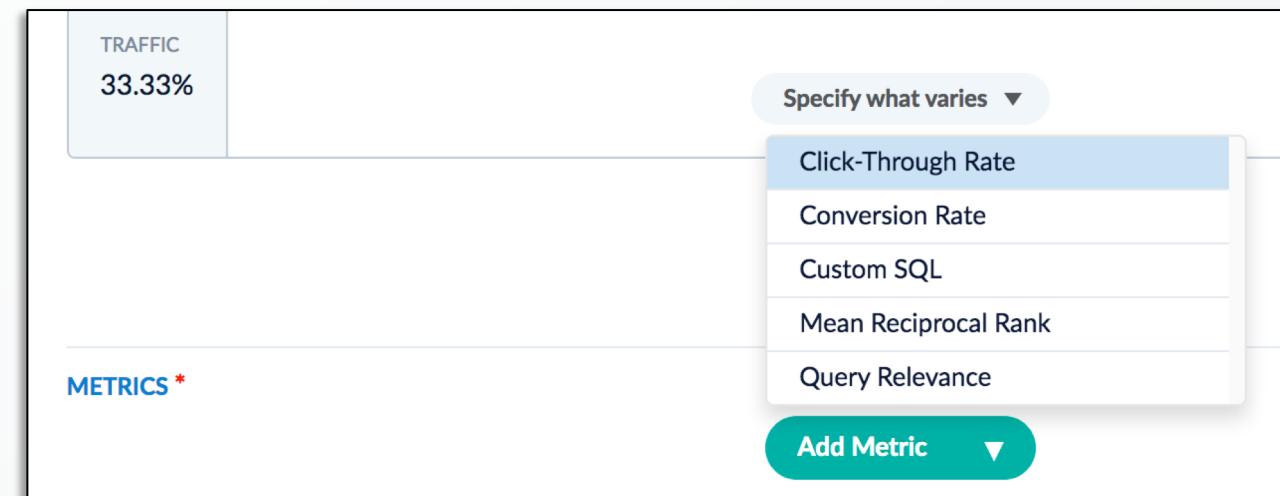
WEIGHT 1	NAME control_plus_qi	QUERY PIPELINE Labs_department_QI	
TRAFFIC 33.33%	Specify what varies ▾		

*If you did not complete the lab **Building a Query Intent Classifier**, skip this variant. The first two are enough to demonstrate the experiments framework.*

*In addition to variants, we need to specify how Fusion will measure the “success” of each path. We do this by adding **metrics**.*

- At the bottom of the page, click **Add Metric > Click-Through Rate**

This metric tracks how frequently a user query is followed by a click on a document. The assumption is that if the user clicked, it means that Fusion showed them something they wanted—the simplest definition of “success” in the context of search.



- Name the metric `clickthrough_rate`
- Click the **primary** checkbox

METRICS *

PRIMARY	<input checked="" type="checkbox"/> Click-Through Rate Measures the number of clicks per query since the beginning of the experiment	
NAME *	<code>clickthrough_rate</code>	Description <hr/>

Setting Up an Experiment

The finished experiment should look like this:

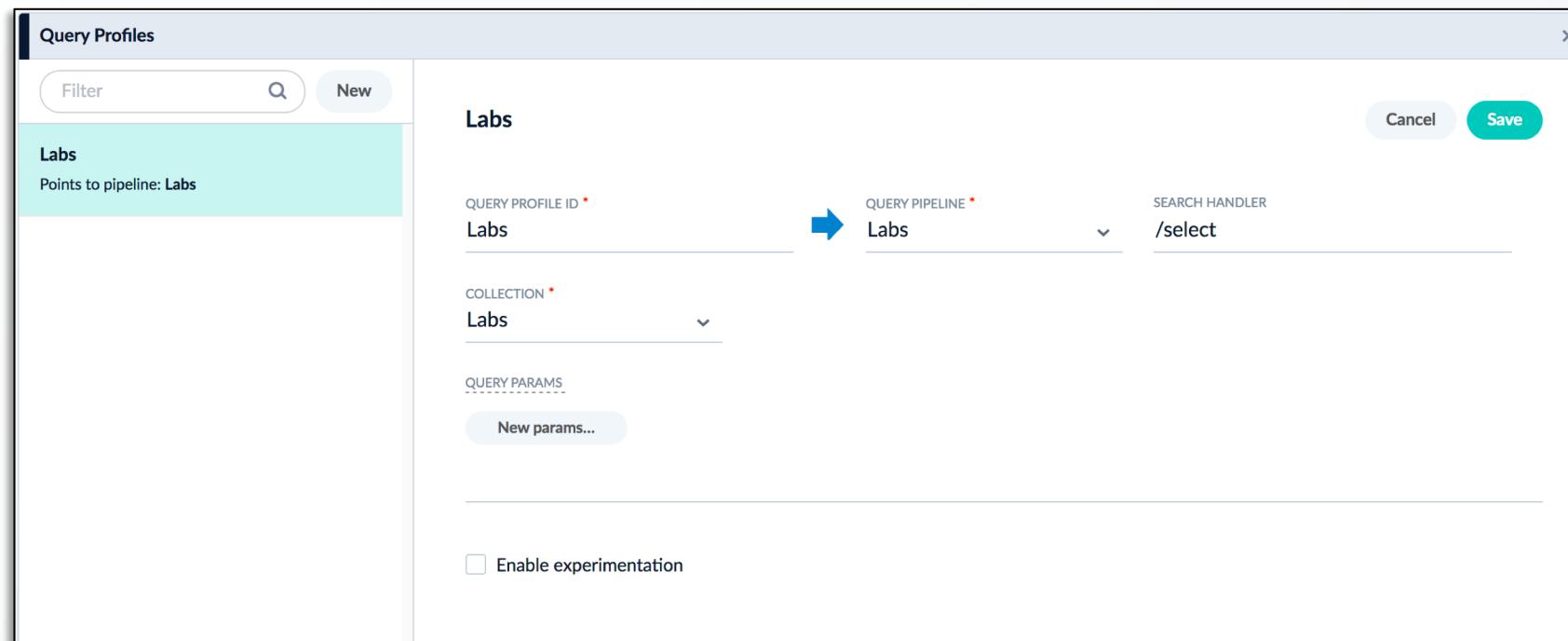
- At the top of the page, click **Save**

ID *	Baseline_vs_Signals_vs_QI		UNIQUE ID PARAMETER	userId	BASE COLLECTION FOR SIGNALS *	Labs
Description						
VARIANTS						
Specify what varies in this variant, and optionally change the traffic weight						
WEIGHT	NAME	QUERY PIPELINE				
1	control	Labs_no_signals				
TRAFFIC						
33.33%						
Specify what varies ▾						
WEIGHT	NAME	QUERY PIPELINE				
1	control_plus_signals	Labs				
TRAFFIC						
33.33%						
Specify what varies ▾						
WEIGHT	NAME	QUERY PIPELINE				
1	control_plus_qi	Labs_department_QI				
TRAFFIC						
33.33%						
Specify what varies ▾						
METRICS *						
PRIMARY	Click-Through Rate					
<input checked="" type="checkbox"/>	Measures the number of clicks per query since the beginning of the experiment					
NAME *	clickthrough_rate					
Description						

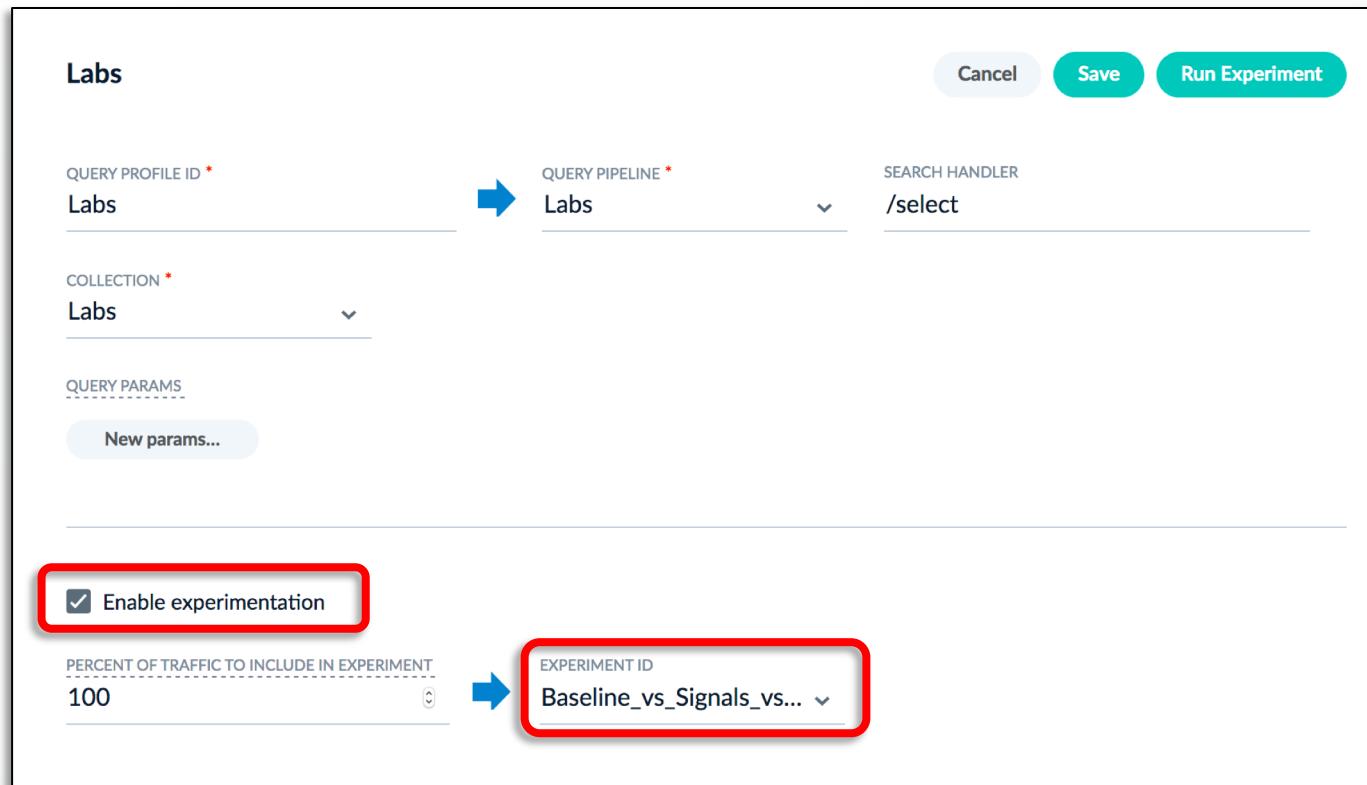
Running an Experiment

Before an experiment can be run, it first must be attached to a Query Profile.

- In the left side menu, go to **QUERYING > Query Profiles**
- Select the **Labs** profile



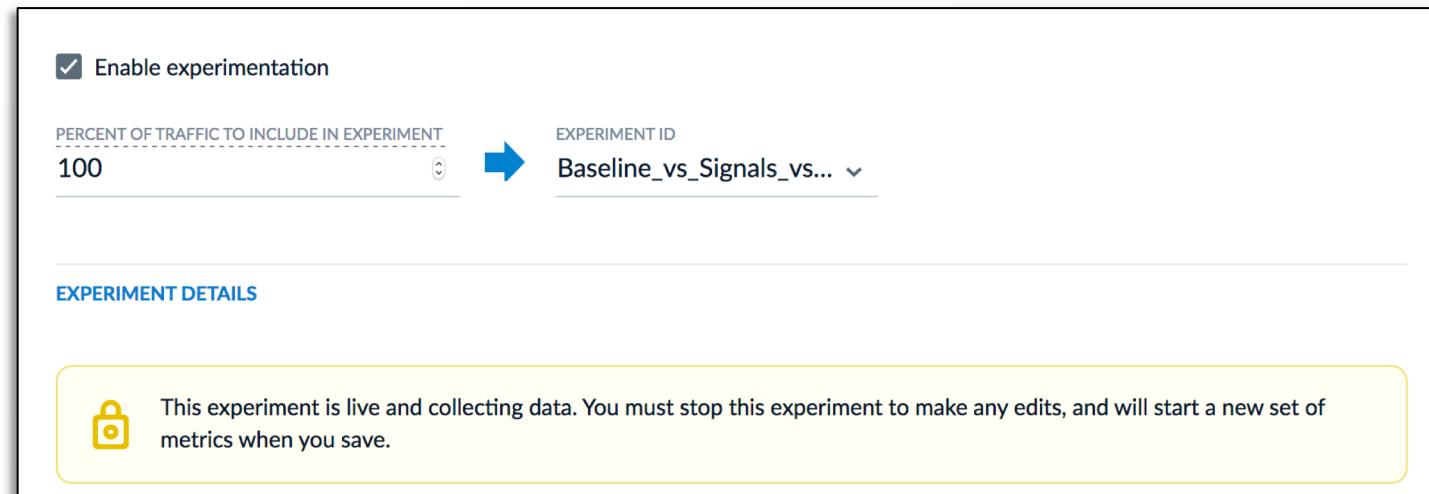
- Click the **Enable Experimentation** checkbox
- In the **Experiment ID** dropdown, select **Baseline_vs_Signals_vs_QI**



The screenshot shows the 'Labs' experiment configuration screen. At the top right are three buttons: 'Cancel', 'Save' (in a teal box), and 'Run Experiment'. Below these are three input fields: 'QUERY PROFILE ID *' (set to 'Labs'), 'QUERY PIPELINE *' (set to 'Labs'), and 'SEARCH HANDLER' (set to '/select'). Underneath are 'COLLECTION *' (set to 'Labs') and 'QUERY PARAMS' (with a 'New params...' button). At the bottom left is a checked checkbox labeled 'Enable experimentation' (surrounded by a red box). To its right is a dropdown menu labeled 'EXPERIMENT ID' with the value 'Baseline_vs_Signals_vs...'. The entire interface has a clean, modern design with a white background and light gray borders for the input fields.

- At the top of the page, click **Save**
- Click **Run Experiment**

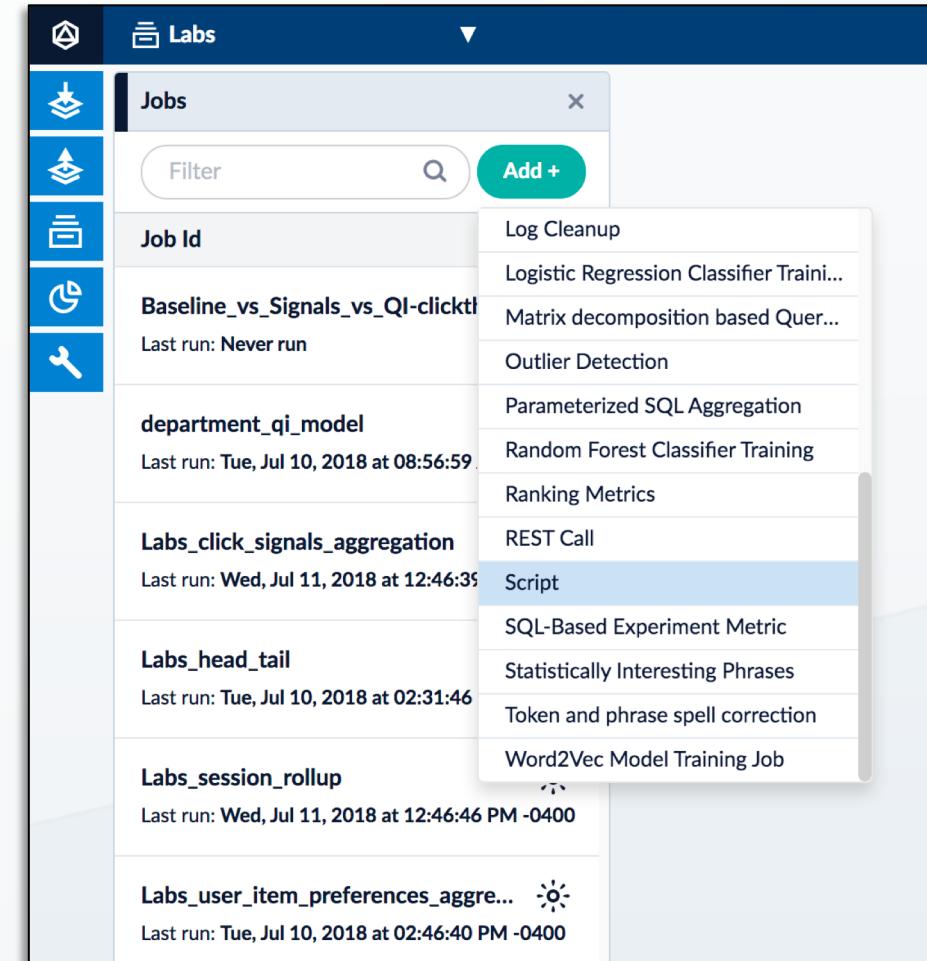
The experiment will be edit-locked until stopped by an administrator.



However, note that Fusion does not currently enforce edit-locks on any of the underlying components—pipelines, collections, landing pages, etc. Therefore, the architect must take care not to disrupt experimental integrity by modifying any of the involved components.

Typically, an experiment will run for a number of days, and capture signals generated by large numbers of users. We have neither the time nor the users to do this, so we will simulate user activity with a script.

- In the left side menu of Fusion Admin, go to **COLLECTIONS > Jobs**
- Click **Add**
- Select **Script**



- Name the job **Generate_clicks**
- In the **Scala Script** pane, copy and paste the code from this link:
[Generate clicks script](#)

* Spark Job ID
Generate_clicks

* Scala Script

OPEN EDITOR 

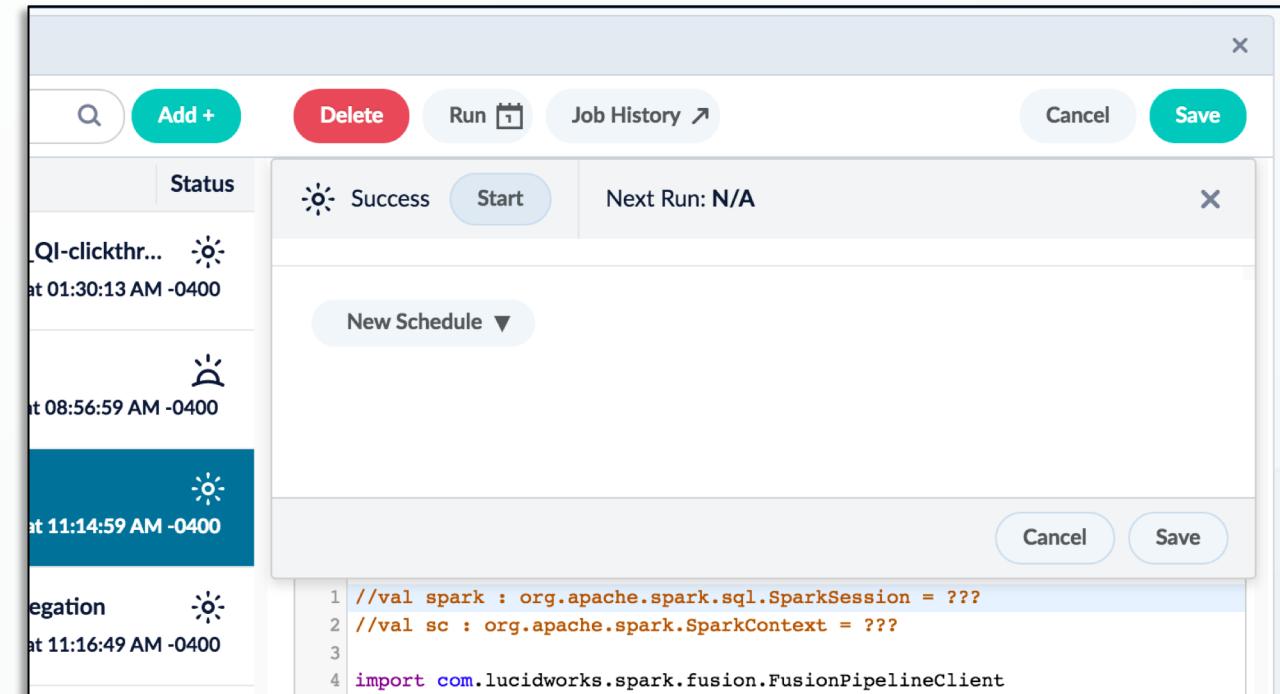
```
1 //val spark : org.apache.spark.sql.SparkSession = ???  
2 //val sc : org.apache.spark.SparkContext = ???  
3  
4 import com.lucidworks.spark.fusion.FusionPipelineClient  
5 import shaded.com.fasterxml.jackson.databind.JsonNode  
6 //import com.fasterxml.jackson.databind.JsonNode  
7 import java.time.format.DateTimeFormatter  
8 import java.util.{Collections, Date}  
9
```

*This script pulls signals from the **Labs_signals** collection, and sends them as queries to the **Labs** catalog.*

If the clicked document from the signal is in the results set of the query, this script will generate a new click signal and send it to Fusion.

- Click Save
- Click Run > Start

If, for any reason, you want to delete these generated clicks from the collection, execute the following command in your terminal:



```
curl -X POST 'http://localhost:8983/solr/Labs_signals/update?commit=true'  
-d '<delete><query>app_id:("scala_traffic_gen")</query></delete>'
```

- Wait for the job status to shows **Success**

This should take ~5 minutes

- In the top left menu, change to the **Labs_signals** collection
- In the left side menu, go to **QUERYING > Query Workbench**
- **Add a field facet for app_id**

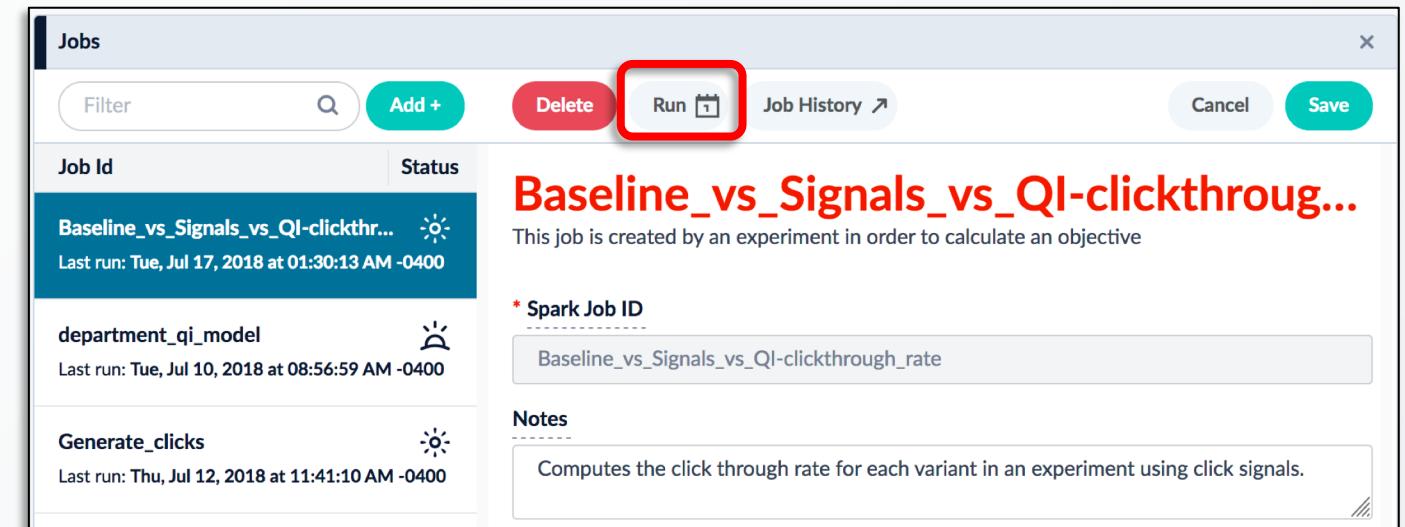
*If you do not see **app_id** in the field list, add it via the Field Facet stage instead. If **app_id** is not available in the Field Facet stage, the **Generate_clicks** job did not succeed, and troubleshooting is needed.*

Evaluating the Experiment

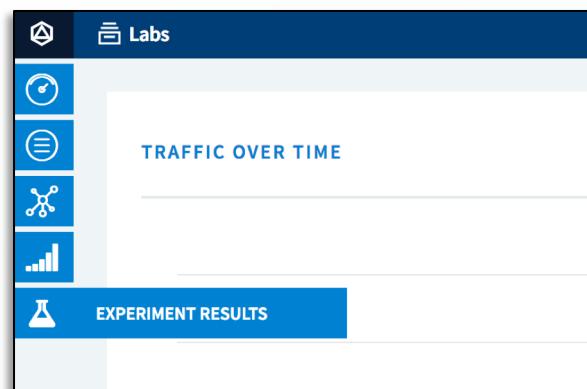
By default, experiment metrics every 12 hours. We don't want to wait that long, so we'll kick off the metrics analysis job manually.

- In the left side menu of Fusion Admin, go to **COLLECTIONS > Jobs**
- Select **Baseline_vs_Signals_vs_QI-clickthrough_rate**
- Click **Run > Start**

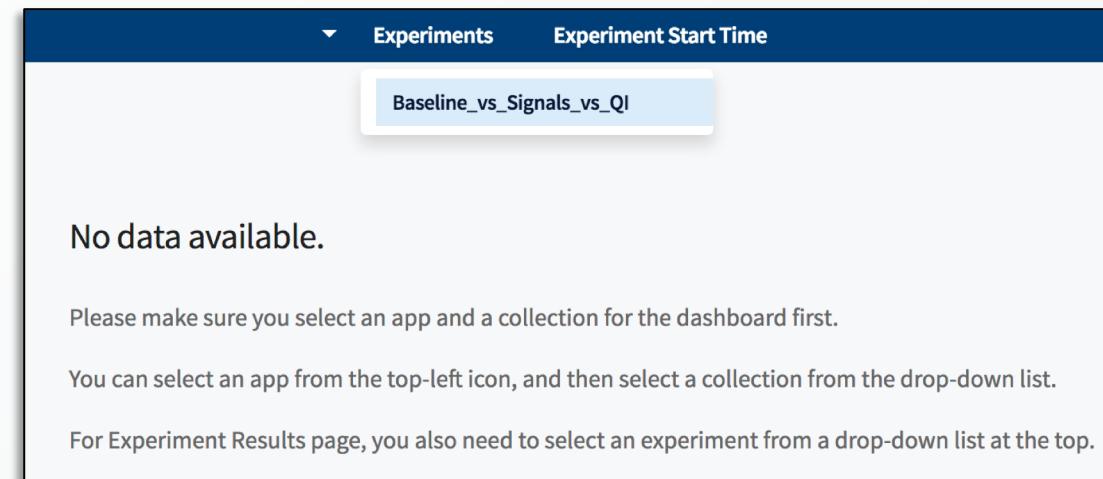
The job should finish in ~30 seconds



- In the left side menu, go to **ANALYTICS > App Insights**
- In the left side menu of **Insights**, go to **Experiment Results**



- In the top dropdown, select **Baseline_vs_Signals_vs_QI**



No data available.

Please make sure you select an app and a collection for the dashboard first.

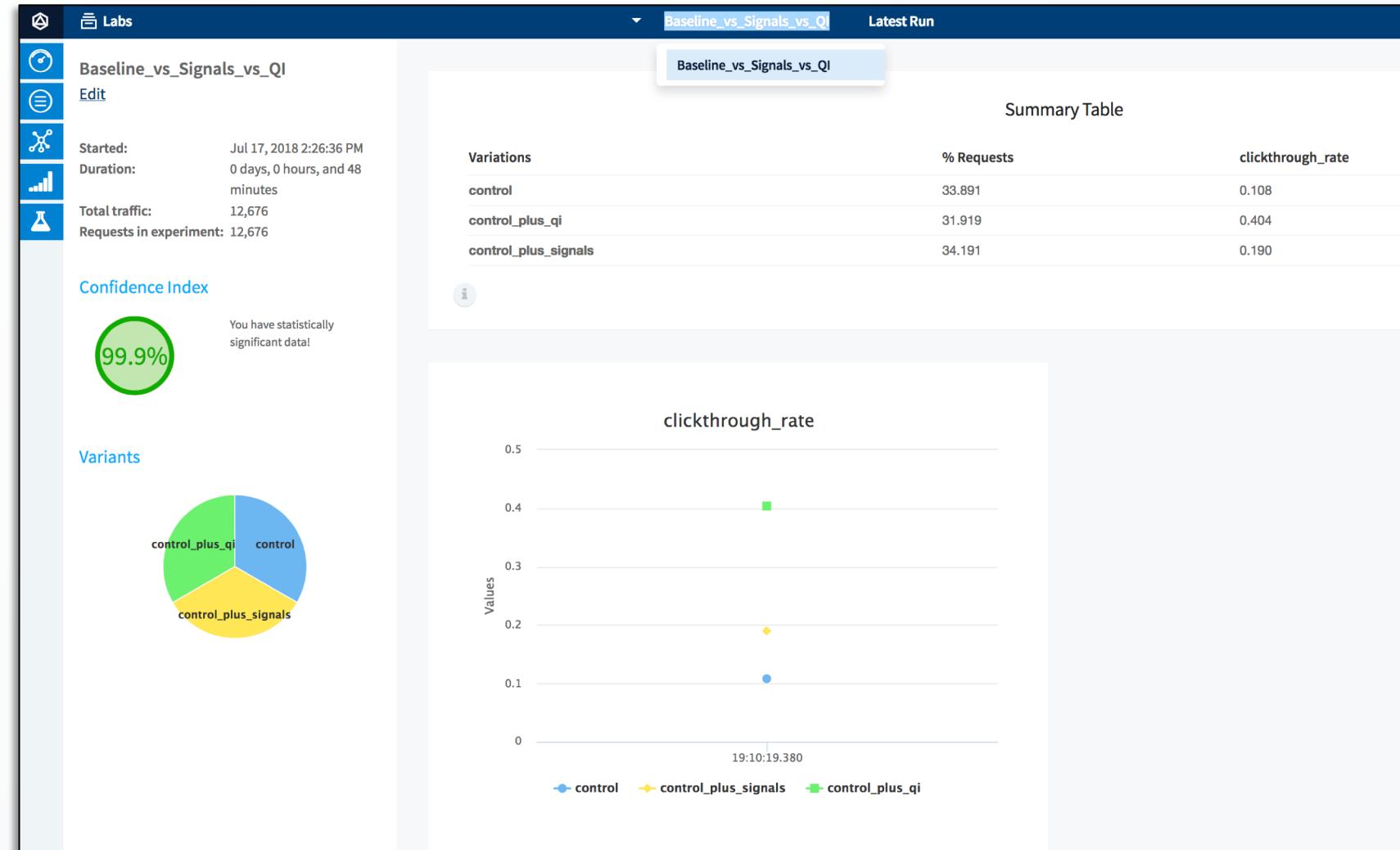
You can select an app from the top-left icon, and then select a collection from the drop-down list.

For Experiment Results page, you also need to select an experiment from a drop-down list at the top.

Evaluating the experiment

The results dashboard should look like this. Your exact results may vary; The **Generate_clicks** job is stochastic and not fully predictable.

However, **Signals** should always outperform **Control**. **Query Intent** may outperform both (if performing a Boost) or may underperform both (if performing a Filter).

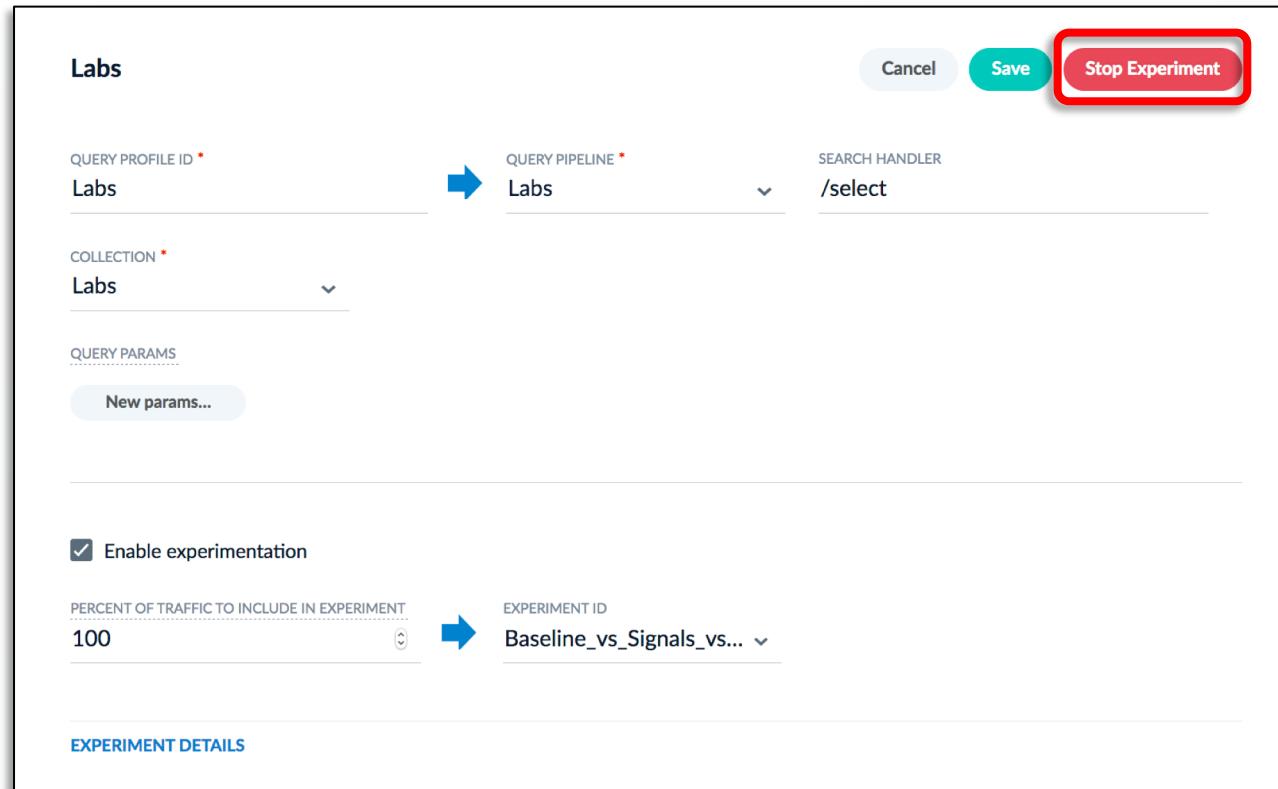


Stopping the Experiment

Once the experiment has run for a sufficient length of time and has gathered statistically significant results, you should shut down the experiment and act upon the results.

- In a web browser, open Fusion Admin
<your-vm-ip>:8764
- In the left side menu, go to **QUERYING > Query Profiles**
- Select **Labs**

- At the top of the page, click **Stop Experiment**



The screenshot shows the 'Labs' experiment configuration page. At the top right, there are three buttons: 'Cancel', 'Save' (in green), and 'Stop Experiment' (in red, which is highlighted with a red box). Below these buttons, there are several input fields:

- QUERY PROFILE ID ***: A dropdown menu showing 'Labs'.
- QUERY PIPELINE ***: A dropdown menu showing 'Labs'.
- SEARCH HANDLER**: A text input field containing '/select'.
- COLLECTION ***: A dropdown menu showing 'Labs'.
- QUERY PARAMS**: A section with a 'New params...' button.
- Enable experimentation**: A checked checkbox.
- PERCENT OF TRAFFIC TO INCLUDE IN EXPERIMENT**: A text input field showing '100'.
- EXPERIMENT ID**: A dropdown menu showing 'Baseline_vs_Signals_vs...'.

At this point, you would likely implement the most successful of the experiment branches.

*In this case, that would entail setting the **Query Pipeline** parameter in the Profile to either **Labs** or **Labs_department_QI**, whichever had the best clickthrough rate.*

End of Lab