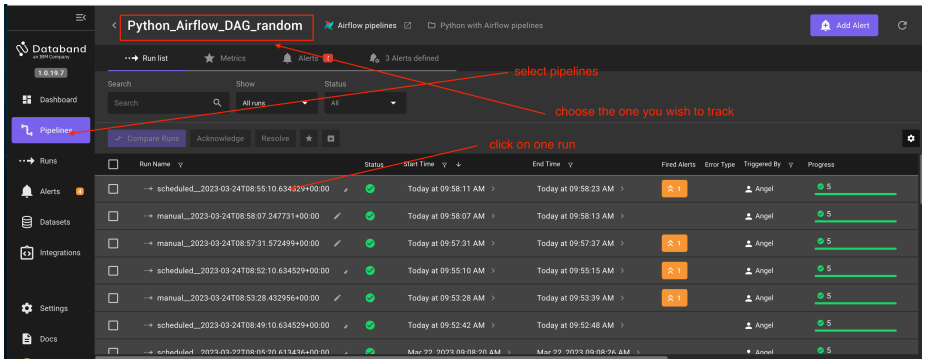
Alerts and exceptions

In this chapter, we will explore the basic ways to handle exceptional situations and how Databand helps find the root cause very quickly.

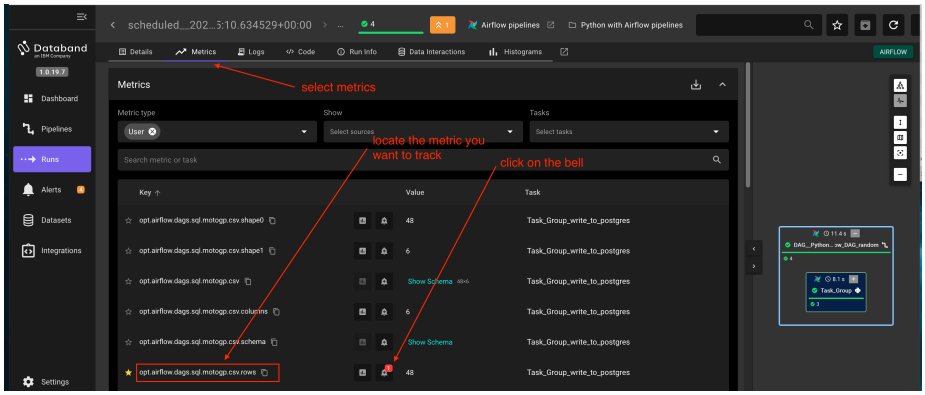
1. Customized alerts and incident management

Apart from the predefined alerts that Databand includes, we will see how to create a customized alert on a specific metric that may be important to track.

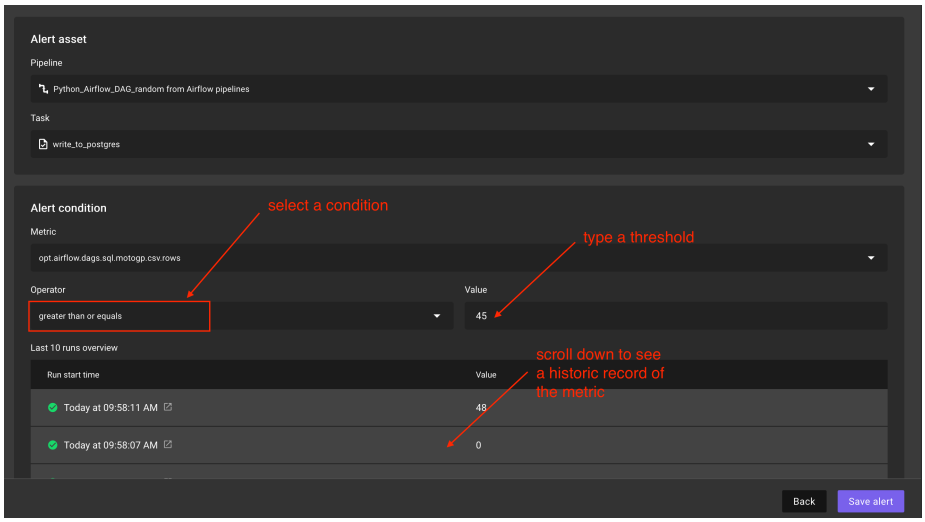
An intuitive way of defining an alert starts by clicking on any run of a pipeline we want to track:



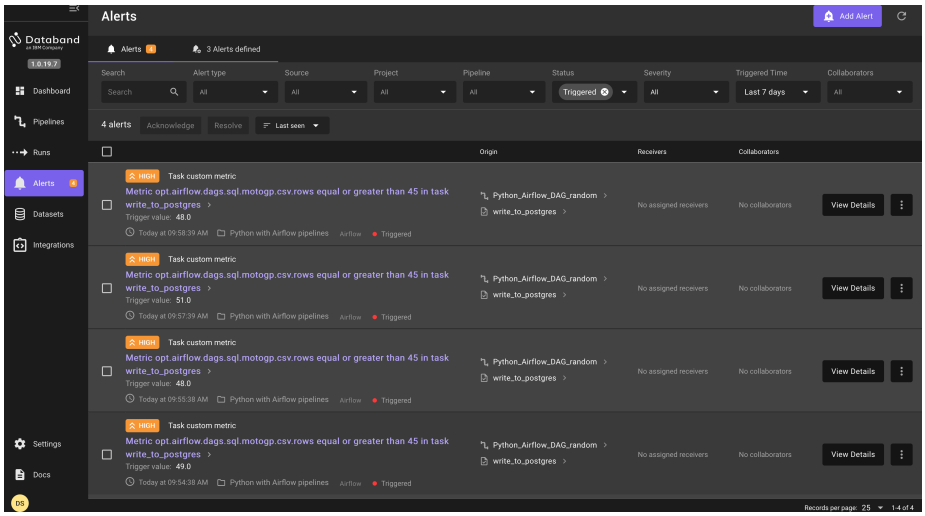
On the metrics track, we locate the specific parameter that will trigger the alterts:



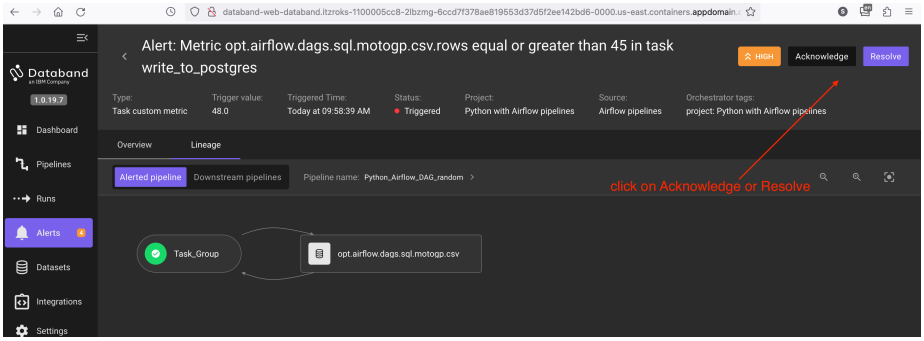
Next, we define the trigger condition and the threshold. In our case, an altert will be triggered if the last step of the pipeline writes more than 45 rows into the destination table.



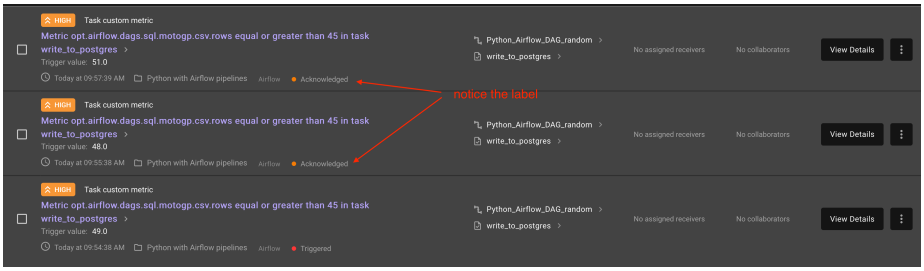
We can let the pipeline run a few times and, if the trigger condition is met, we will start to see new entries in the Alerts section of the Databand GUI:



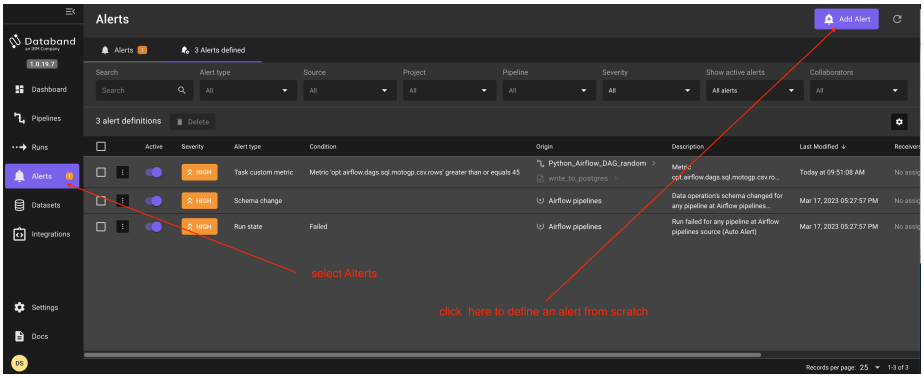
If we click on any of the alerts we will see the normal details of the run but we want to highlight the buttons displayed on the right top corner. They are intende to act as incident management



You can click on the Resolve button to delete the alert entry (not the alert definition) or on Acknowledge to investigate the cause of the alert and let it in the list.



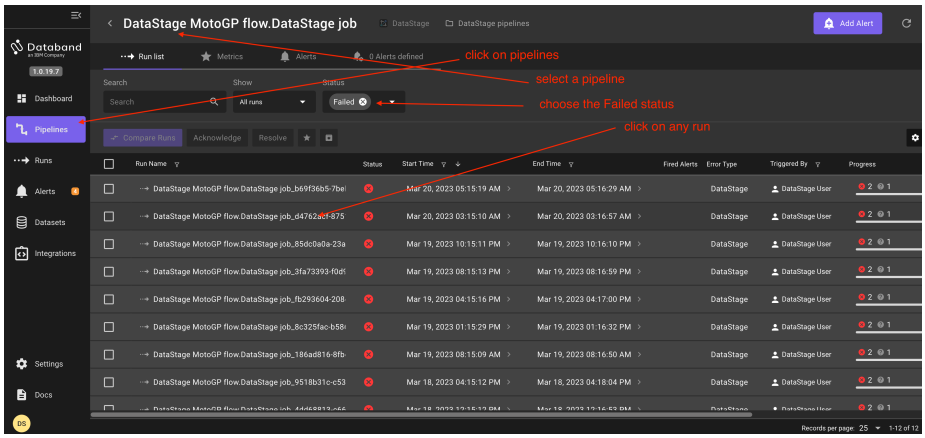
Notice that there is other ways of defining customized alerts. For example, like shown on the next picture:



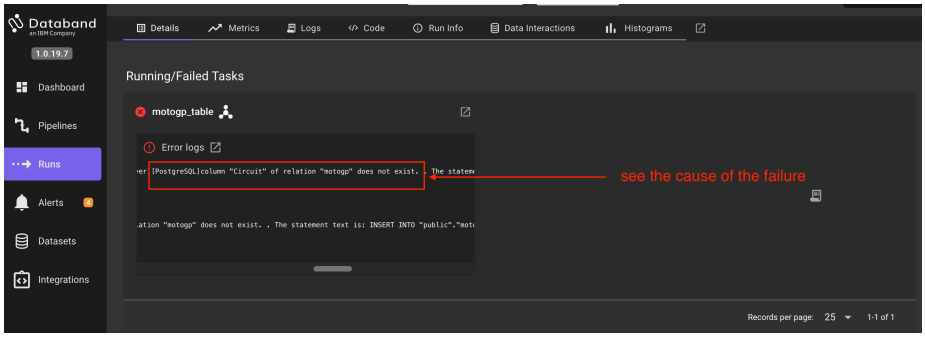
2. Exceptions

In this section you can play with the test pipelines and introduce errors to to how Databand will report them.

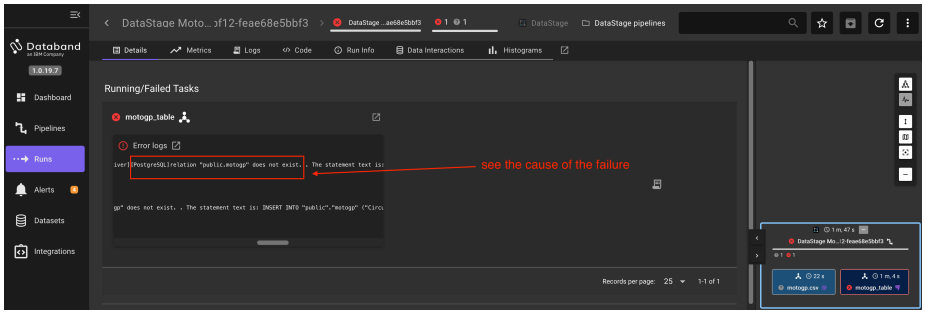
In case of a problem, Databand will mark the runs with the status "Failed":



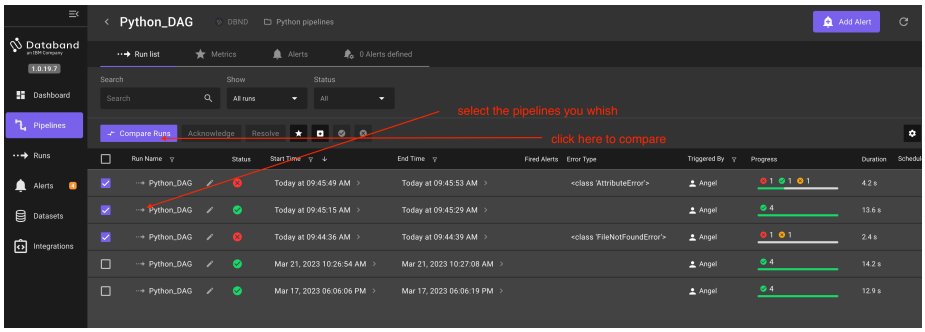
In this case, you the cause of the problem is that a column does not exist, as indicated in the first line of the log.



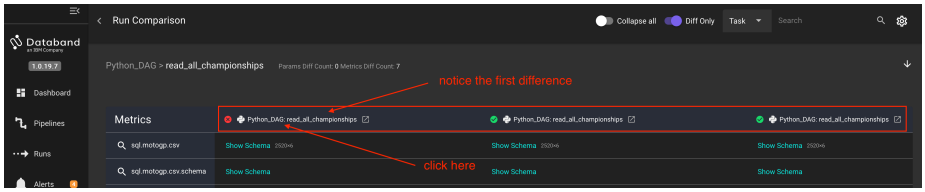
If you choose another failed run, you may see a different error. In that case, that the table does not exist, as displayed in the first line of the log.



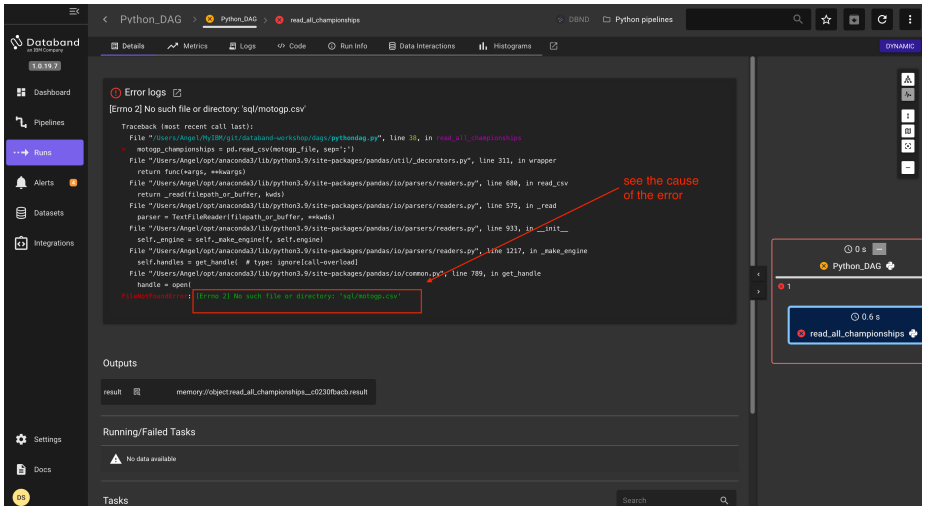
Another great way to determine the cause of an error is to select a few runs and click on "Compare Runs" to see the differences between a good runs and bad ones.



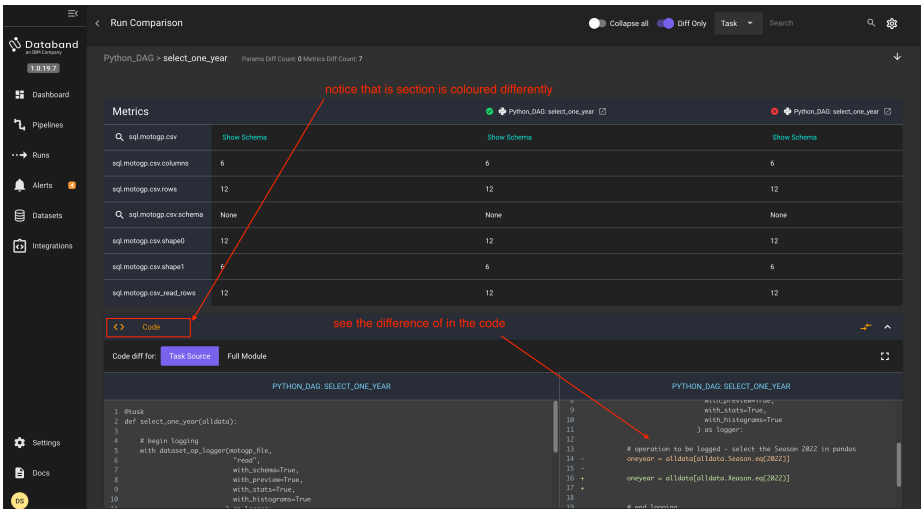
The first difference can be found on one of the tasks, which has failed:



By clicking on the failed task, you can see immediately the cause of the error.



The second difference would be more difficult to find manually but Databand shows the line of the code that causes the error and compares it with the same line of another run that worked well:



Next Section: Customized Metrics
Previous Section: DataStage pipelines

Return to main