SELECT value:device, value:geo:city, value:geo:state

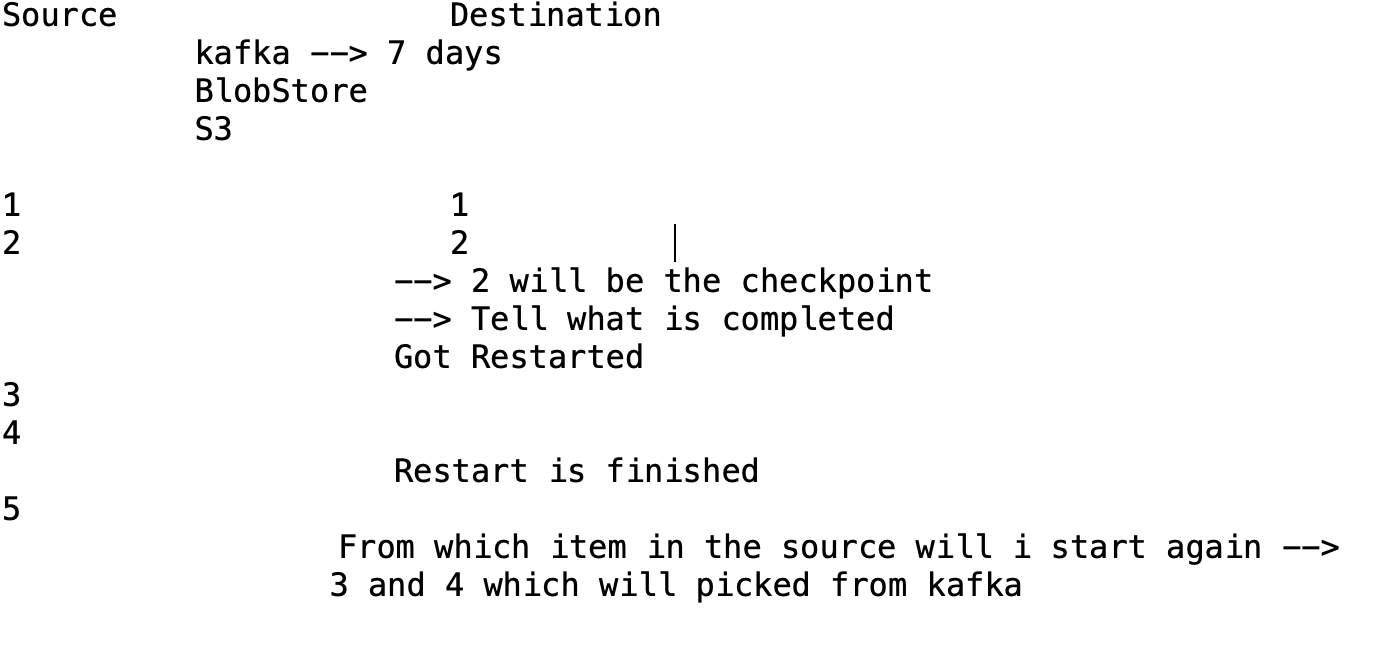
FROM events\_strings

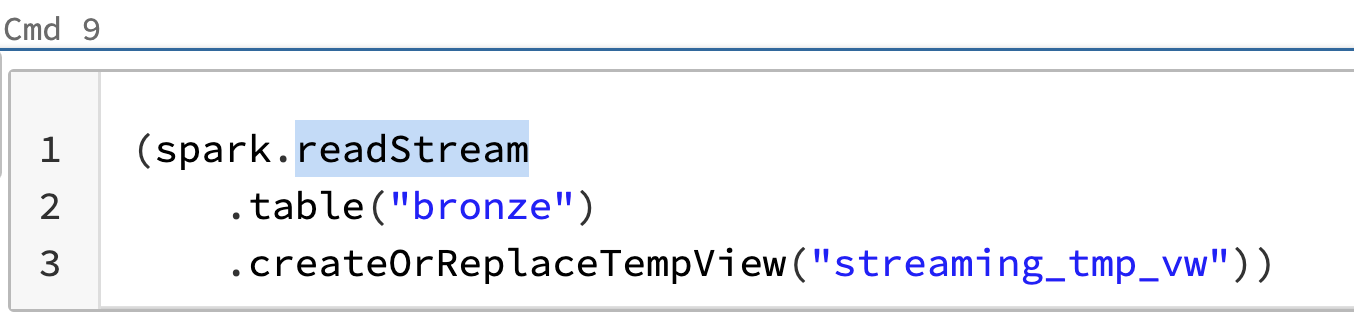
<https://docs.databricks.com/optimizations/higher-order-lambda-functions.html>

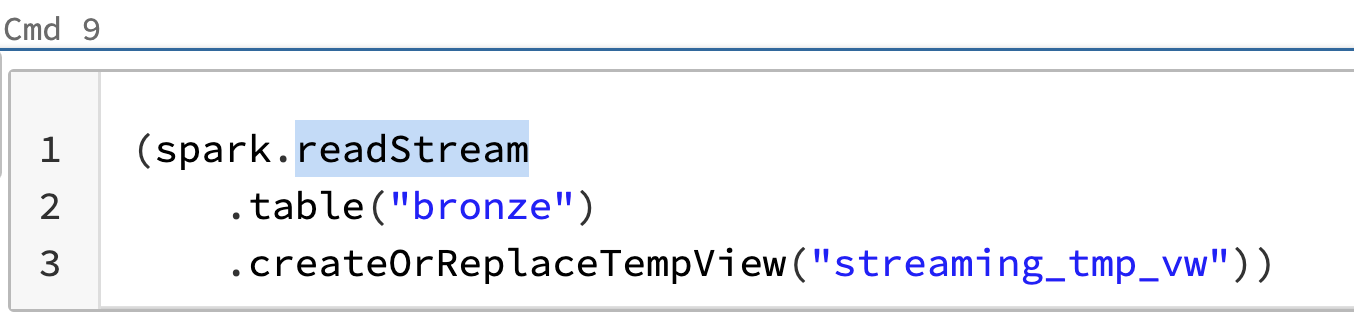
Working with DE 4.8 - SQL UDFs and Control Flow

This notebook will be tried as a Lab after the class:

<https://dbnov7.slack.com/files/U049G3D2XT5/F04AGB98XBK/image.png>







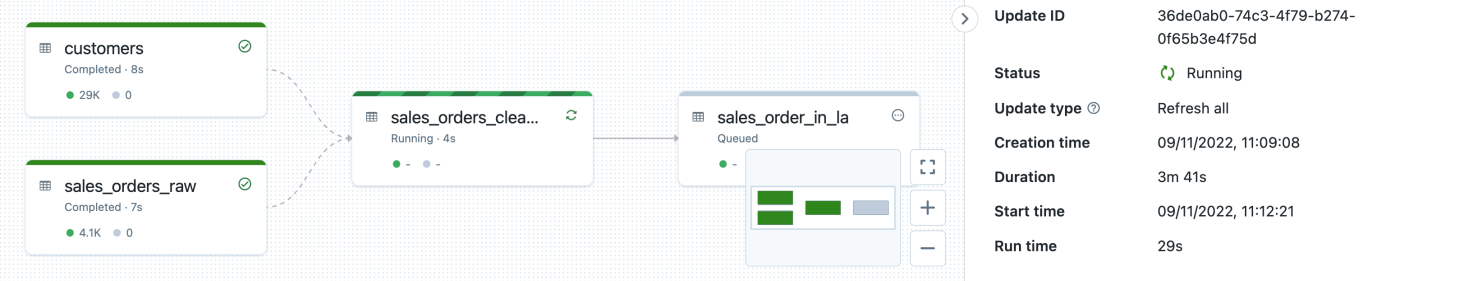
<https://www.databricks.com/product/delta-live-tables>

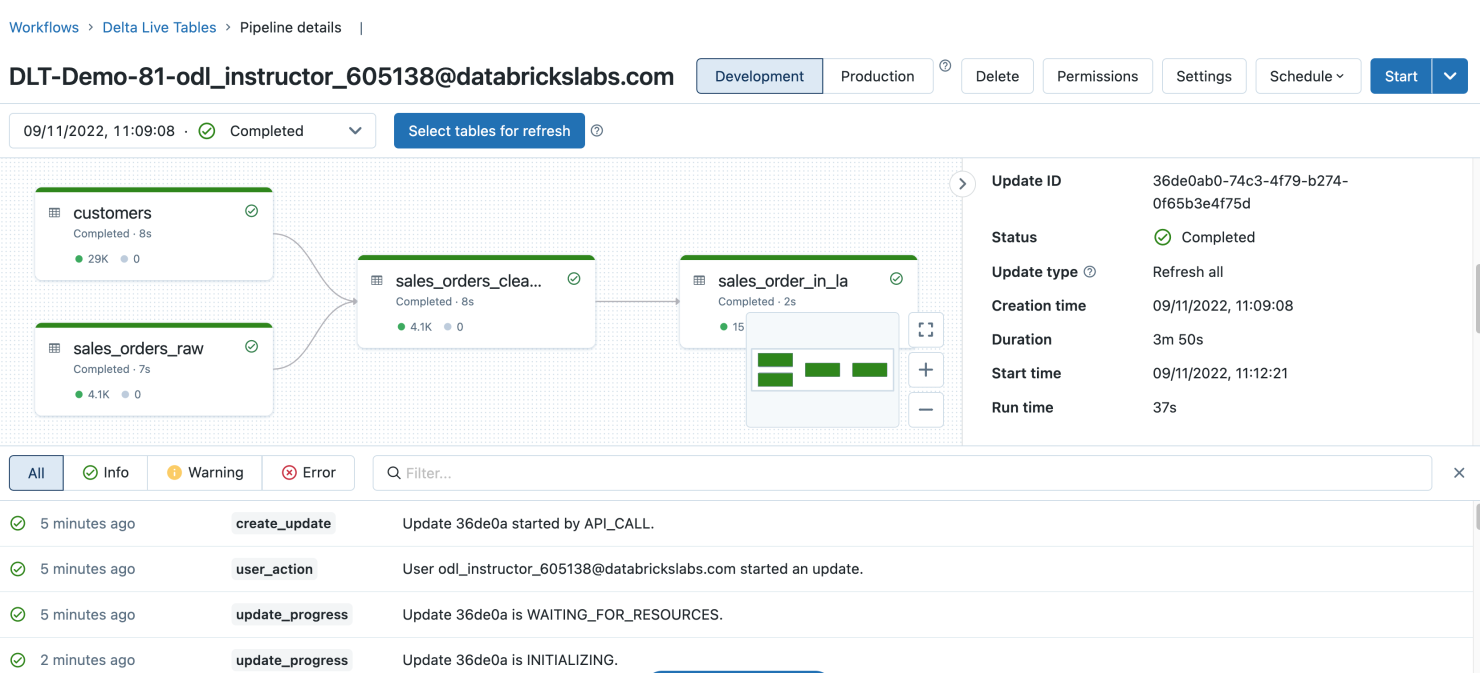
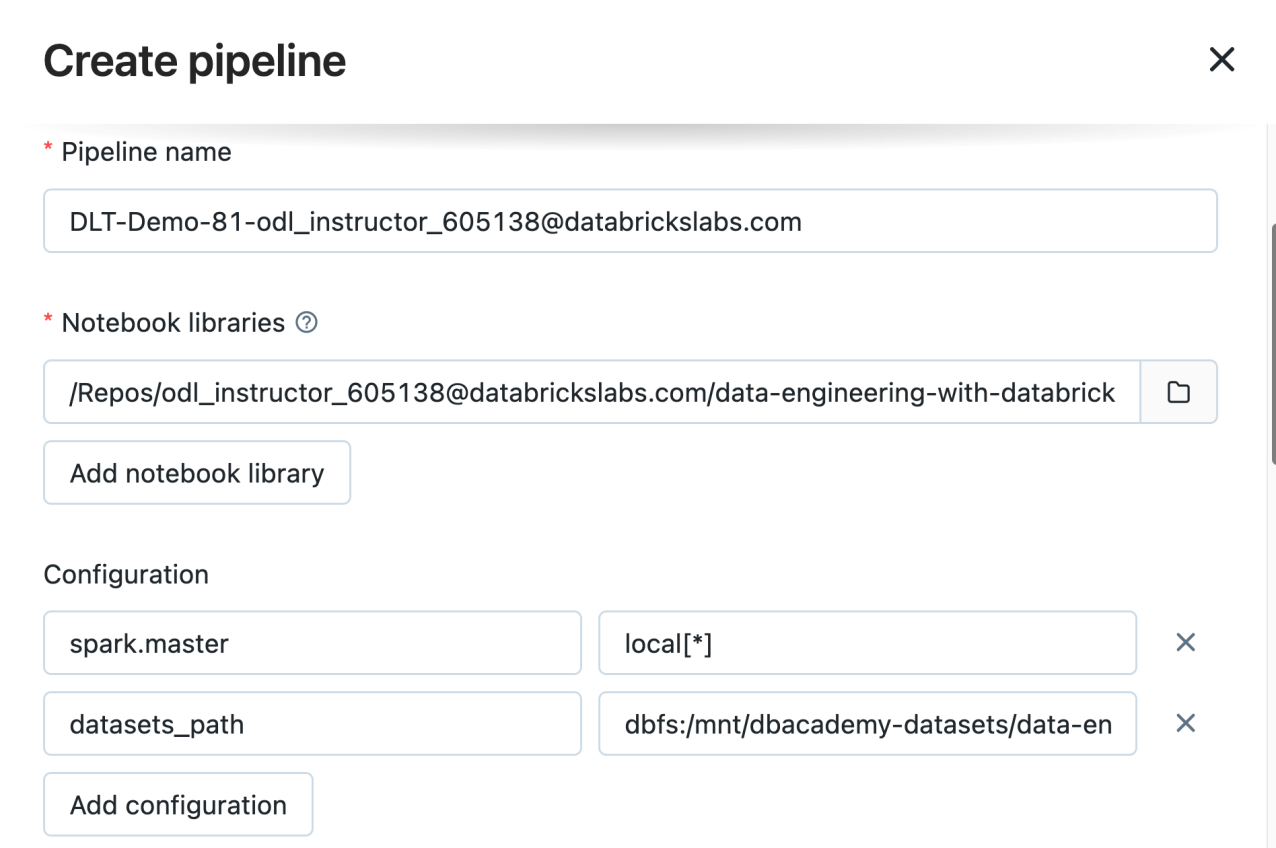
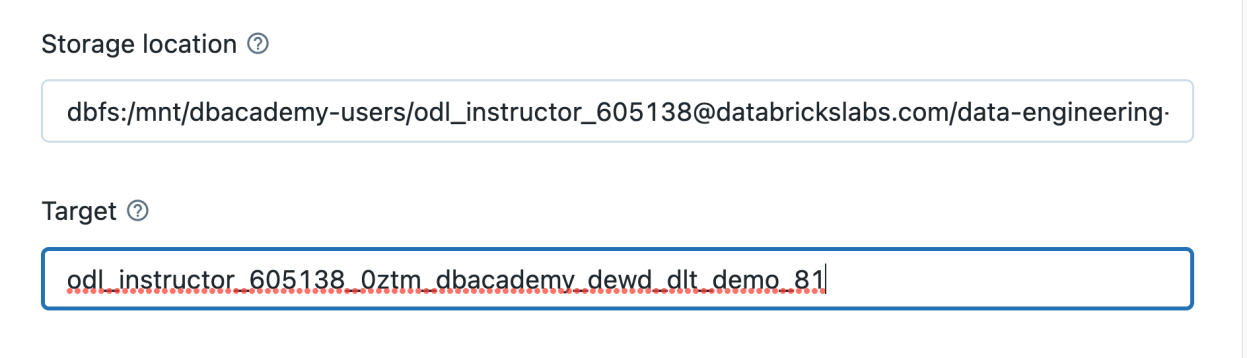
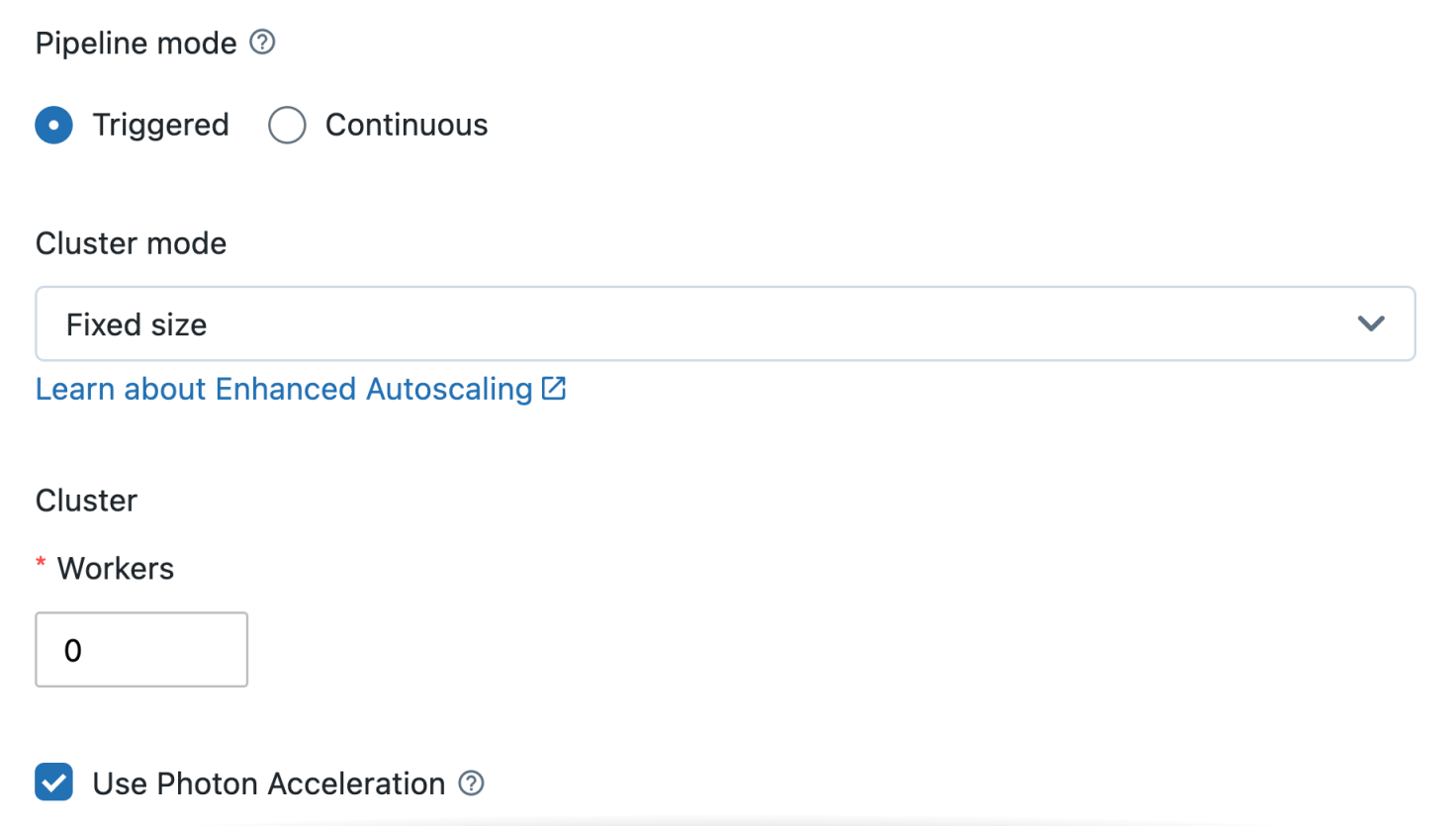
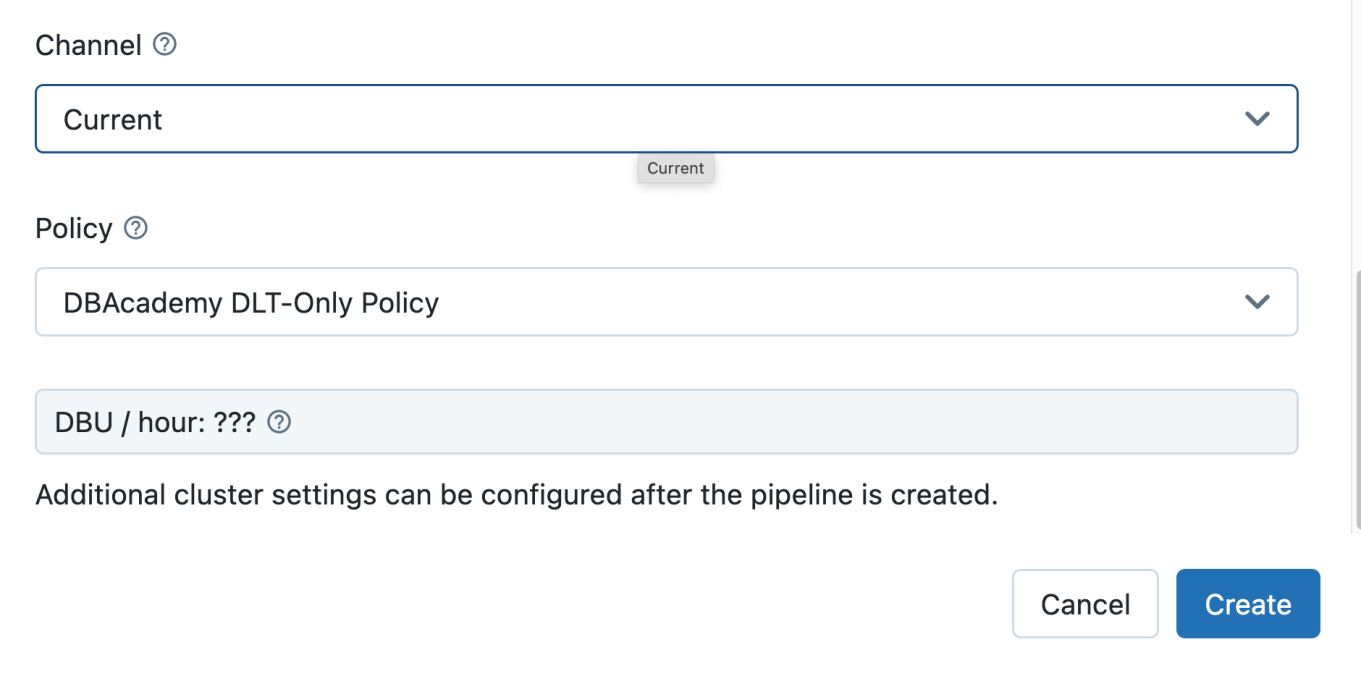
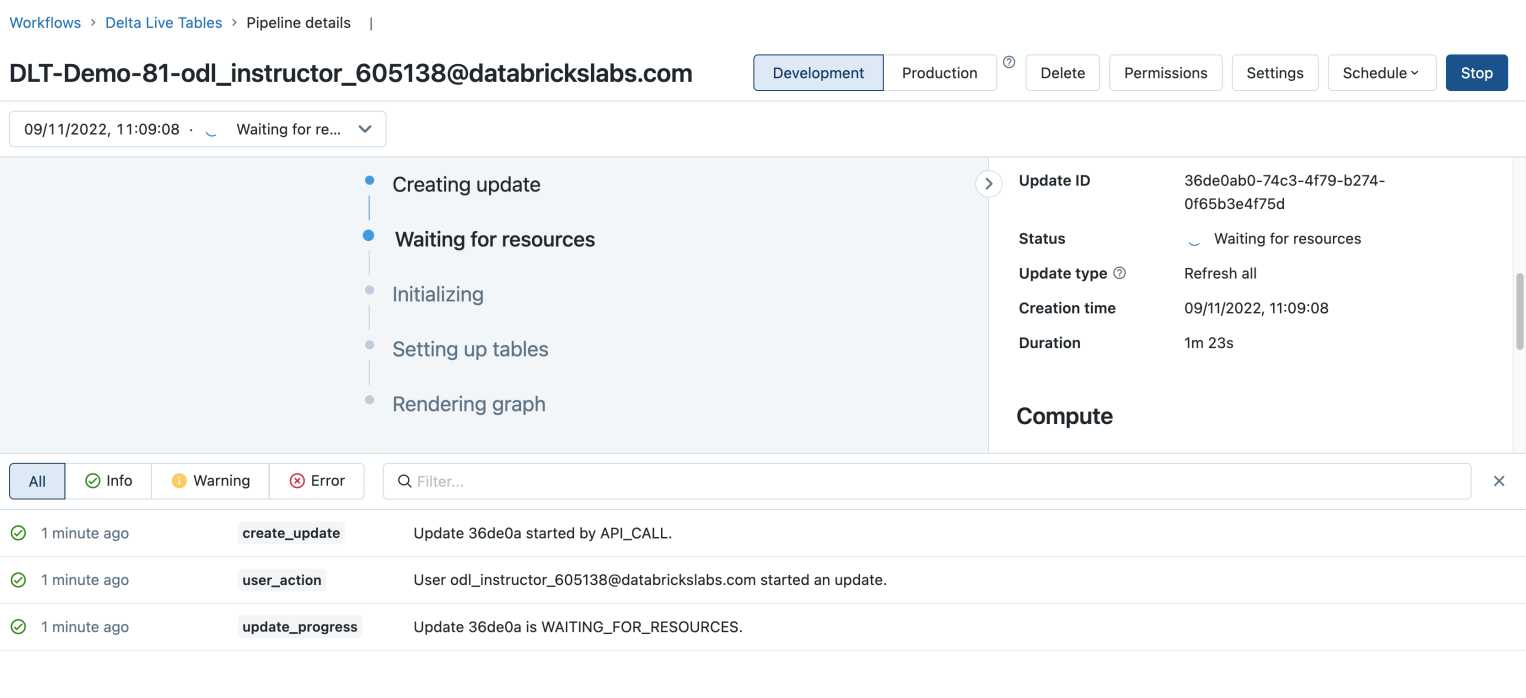
<https://www.databricks.com/discover/pages/getting-started-with-delta-live-tables>

<https://docs.databricks.com/workflows/delta-live-tables/index.html>

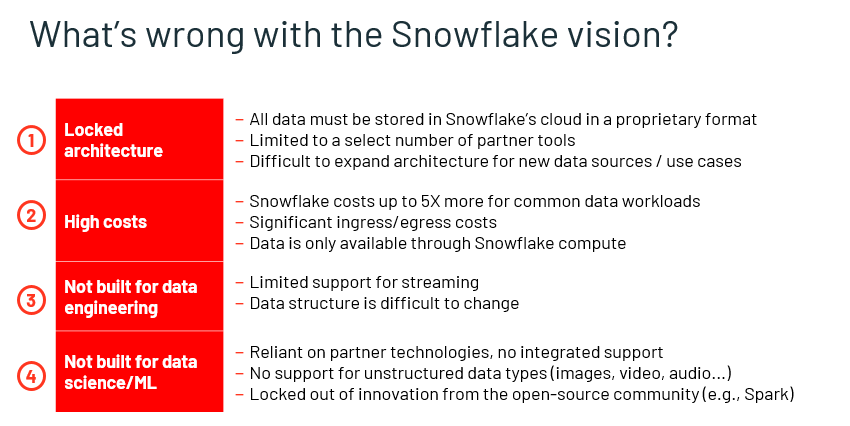
Advanced analytics with Delta Live Tables:

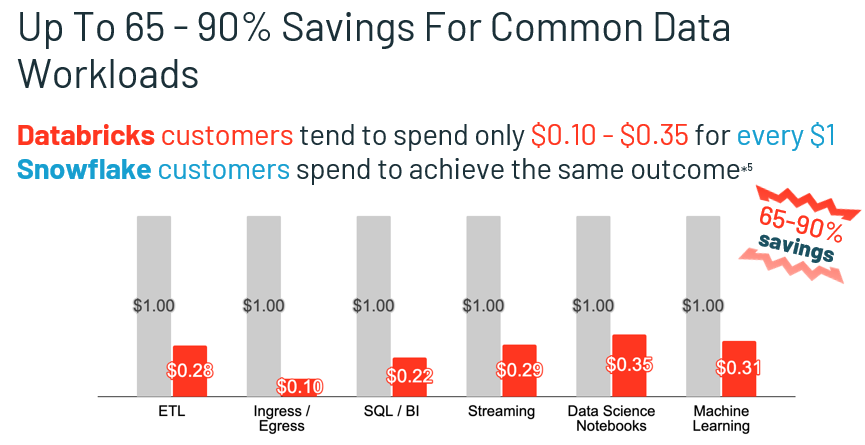
<https://www.youtube.com/watch?v=MZd2MgM5JFY>





SNOWFLAKE:





1 Ganglia Images for the load distribution what i was mentioning in the class. Click on the image below to check the status

[**Z-order curve**](https://en.wikipedia.org/wiki/Z-order_curve)

In mathematical analysis and computer science, functions which are Z-order, Lebesgue curve, Morton space-filling curve, Morton order or Morton code map multidimensional data to one dimension while preserving locality of the data points. It is named in France after Henri Lebesgue, who studied it in 1904, and named in US after Guy Macdonald Morton, who first applied the order to file sequencing in 1966. The z-value of a point in multidimensions is simply calculated by interleaving the binary representations of its coordinate values. Once the data are sorted into this ordering, any one-dimensional data structure can be used such as binary search trees, B-trees, skip lists or (with low significa… Show more

Use this link <https://en.wikipedia.org/wiki/Z-order_curve>

The check the results of the 8.2 pipeline

files = dbutils.fs.ls(DA.paths.storage\_location)

display(files)

For the result 2 of the excel file:

files = dbutils.fs.ls(f"{DA.paths.storage\_location}/system/events")

display(files)

Photon accelerates modern Apache Spark workloads, reducing your total cost per workload.

DBU/Hour is driven by the number of instances and whether Photon is enabled. There is a different DBU cost per DLT Edition (Core, Pro or Advanced). A system-generated job for maintaining Delta Live Tables will also run daily, incurring additional costs at Jobs Compute rate.

java.lang.RuntimeException: Failed to execute python command for notebook '/Users/odl\_user\_783173@databrickslabs.com/data-engineering-with-databricks/Solutions/09 - Task Orchestration with Jobs/DE 9.1 - Scheduling Tasks with the Jobs UI/DE 9.1.1 - Task Orchestration with Databricks Jobs' with id RunnableCommandId(6161955419850073805) and error AnsiResult(---------------------------------------------------------------------------

NameError Traceback (most recent call last)

<command--1> in <cell line: 1>()

----> 1 DA.print\_pipeline\_config()

2

NameError: name 'DA' is not defined,Map(),Map(),List(),List(),Map())

at com.databricks.pipelines.execution.core.languages.PythonRepl.runCmd(PythonRepl.scala:376)

at com.databricks.pipelines.execution.service.PipelineRunnable$.$anonfun$loadPythonGraph$8(PipelineRunnable.scala:463)

at com.databricks.pipelines.execution.service.PipelineRunnable$.$anonfun$loadPythonGraph$8$adapted(PipelineRunnable.scala:459)

at scala.collection.IndexedSeqOptimized.foreach(IndexedSeqOptimized.scala:36)

at scala.collection.IndexedSeqOptimized.foreach$(IndexedSeqOptimized.scala:33)

at scala.collection.mutable.ArrayOps$ofRef.foreach(ArrayOps.scala:198)

at com.databricks.pipelines.execution.service.PipelineRunnable$.$anonfun$loadPythonGraph$7(PipelineRunnable.scala:459)

at com.databricks.pipelines.execution.service.PipelineRunnable$.$anonfun$loadPythonGraph$7$adapted(PipelineRunnable.scala:450)

at scala.collection.immutable.Map$Map1.foreach(Map.scala:193)

at com.databricks.pipelines.execution.service.PipelineRunnable$.loadPythonGraph(PipelineRunnable.scala:450)

at com.databricks.pipelines.execution.service.PipelineRunnable.loadGraph(PipelineRunnable.scala:255)

at com.databricks.pipelines.execution.service.PipelineRunnable.loadGraph(PipelineRunnable.scala:177)

at com.databricks.pipelines.execution.service.UpdateExecution.$anonfun$initializeAndloadGraph$1(UpdateExecution.scala:400)

at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)

at com.databricks.pipelines.execution.core.monitoring.DeltaPipelinesUsageLogging.$anonfun$recordPipelinesOperation$2(DeltaPipelinesUsageLogging.scala:112)

at com.databricks.pipelines.common.monitoring.OperationStatusReporter.executeWithPeriodicReporting(OperationStatusReporter.scala:120)

at com.databricks.pipelines.common.monitoring.OperationStatusReporter$.executeWithPeriodicReporting(OperationStatusReporter.scala:160)

at com.databricks.pipelines.execution.core.monitoring.DeltaPipelinesUsageLogging.$anonfun$recordPipelinesOperation$5(DeltaPipelinesUsageLogging.scala:131)

at com.databricks.logging.UsageLogging.$anonfun$recordOperation$1(UsageLogging.scala:413)

at com.databricks.logging.UsageLogging.executeThunkAndCaptureResultTags$1(UsageLogging.scala:507)

at com.databricks.logging.UsageLogging.$anonfun$recordOperationWithResultTags$4(UsageLogging.scala:528)

at com.databricks.logging.Log4jUsageLoggingShim$.$anonfun$withAttributionContext$1(Log4jUsageLoggingShim.scala:32)

at scala.util.DynamicVariable.withValue(DynamicVariable.scala:62)

at com.databricks.logging.AttributionContext$.withValue(AttributionContext.scala:94)

at com.databricks.logging.Log4jUsageLoggingShim$.withAttributionContext(Log4jUsageLoggingShim.scala:30)

at com.databricks.logging.UsageLogging.withAttributionContext(UsageLogging.scala:283)

at com.databricks.logging.UsageLogging.withAttributionContext$(UsageLogging.scala:282)

at com.databricks.pipelines.execution.core.monitoring.PublicLogging.withAttributionContext(DeltaPipelinesUsageLogging.scala:22)

at com.databricks.logging.UsageLogging.withAttributionTags(UsageLogging.scala:318)

at com.databricks.logging.UsageLogging.withAttributionTags$(UsageLogging.scala:303)

at com.databricks.pipelines.execution.core.monitoring.PublicLogging.withAttributionTags(DeltaPipelinesUsageLogging.scala:22)

at com.databricks.logging.UsageLogging.recordOperationWithResultTags(UsageLogging.scala:502)

at com.databricks.logging.UsageLogging.recordOperationWithResultTags$(UsageLogging.scala:422)

at com.databricks.pipelines.execution.core.monitoring.PublicLogging.recordOperationWithResultTags(DeltaPipelinesUsageLogging.scala:22)

at com.databricks.logging.UsageLogging.recordOperation(UsageLogging.scala:413)

at com.databricks.logging.UsageLogging.recordOperation$(UsageLogging.scala:385)

at com.databricks.pipelines.execution.core.monitoring.PublicLogging.recordOperation(DeltaPipelinesUsageLogging.scala:22)

at com.databricks.pipelines.execution.core.monitoring.PublicLogging.recordOperation0(DeltaPipelinesUsageLogging.scala:57)

at com.databricks.pipelines.execution.core.monitoring.DeltaPipelinesUsageLogging.recordPipelinesOperation(DeltaPipelinesUsageLogging.scala:124)

at com.databricks.pipelines.execution.core.monitoring.DeltaPipelinesUsageLogging.recordPipelinesOperation$(DeltaPipelinesUsageLogging.scala:96)

at com.databricks.pipelines.execution.service.PipelineRunnable.recordPipelinesOperation(PipelineRunnable.scala:48)

at com.databricks.pipelines.execution.service.UpdateExecution.executeStage(UpdateExecution.scala:292)

at com.databricks.pipelines.execution.service.UpdateExecution.initializeAndloadGraph(UpdateExecution.scala:363)

at com.databricks.pipelines.execution.service.UpdateExecution.executeUpdate(UpdateExecution.scala:353)

at com.databricks.pipelines.execution.service.UpdateExecution.startImpl(UpdateExecution.scala:151)

at com.databricks.pipelines.execution.service.UpdateExecution.start(UpdateExecution.scala:175)

at com.databricks.pipelines.execution.service.ExecutionBackend$$anon$1.run(ExecutionBackend.scala:351)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)

at java.util.concurrent.FutureTask.run(FutureTask.java:266)

at org.apache.spark.util.threads.SparkThreadLocalCapturingRunnable.$anonfun$run$1(SparkThreadLocalForwardingThreadPoolExecutor.scala:104)

at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)

at org.apache.spark.util.threads.SparkThreadLocalCapturingHelper.runWithCaptured(SparkThreadLocalForwardingThreadPoolExecutor.scala:68)

at org.apache.spark.util.threads.SparkThreadLocalCapturingHelper.runWithCaptured$(SparkThreadLocalForwardingThreadPoolExecutor.scala:54)

at org.apache.spark.util.threads.SparkThreadLocalCapturingRunnable.runWithCaptured(SparkThreadLocalForwardingThreadPoolExecutor.scala:101)

at org.apache.spark.util.threads.SparkThreadLocalCapturingRunnable.run(SparkThreadLocalForwardingThreadPoolExecutor.scala:104)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

