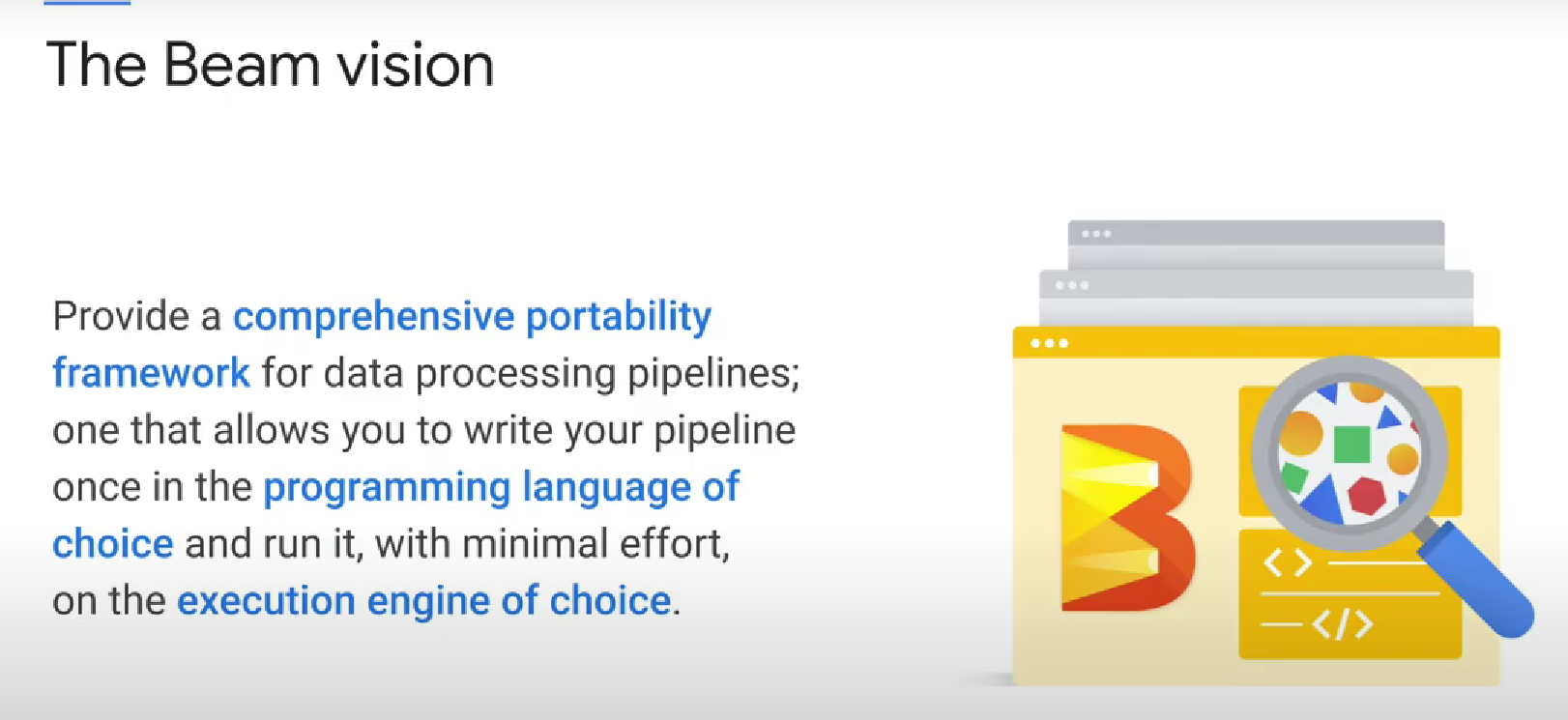
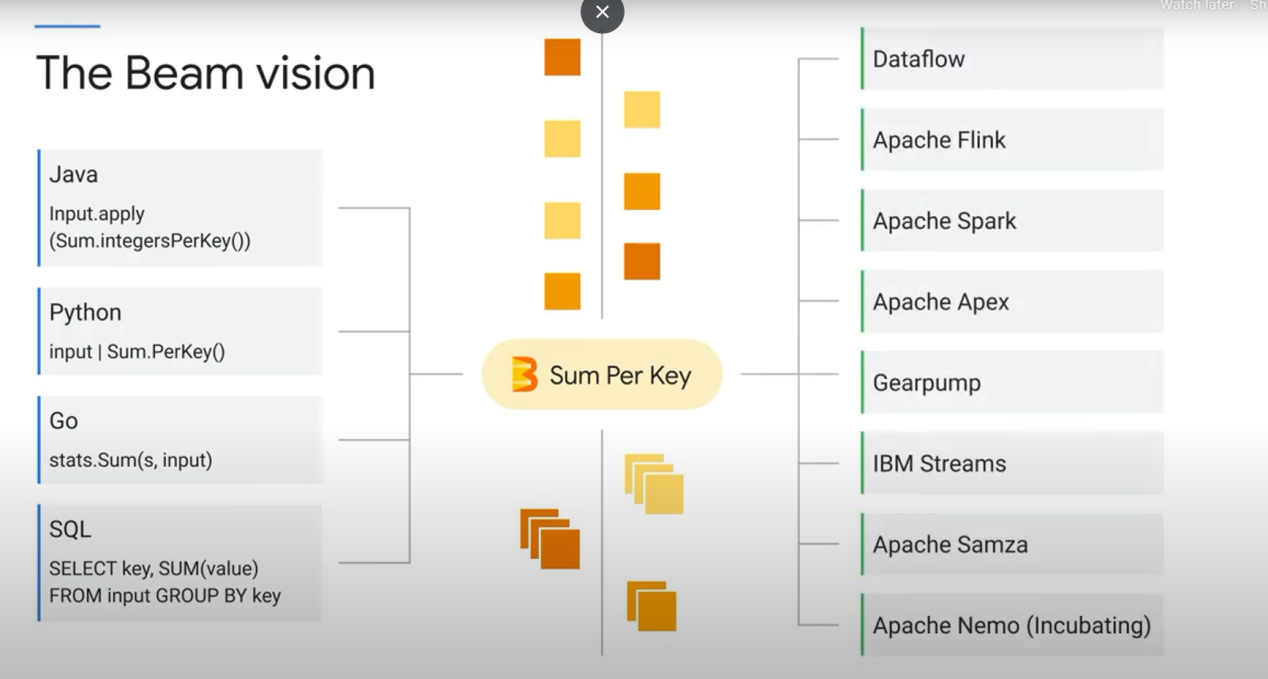
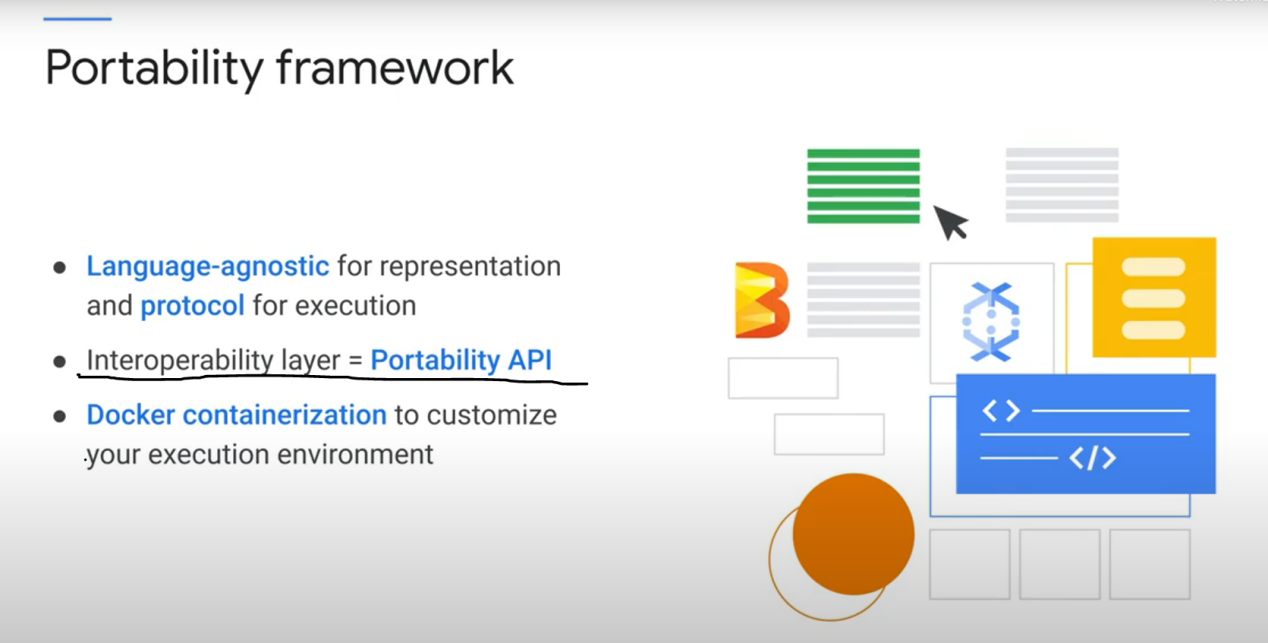


**Dataflow allows you to separate storage ressources from computing ressources.**

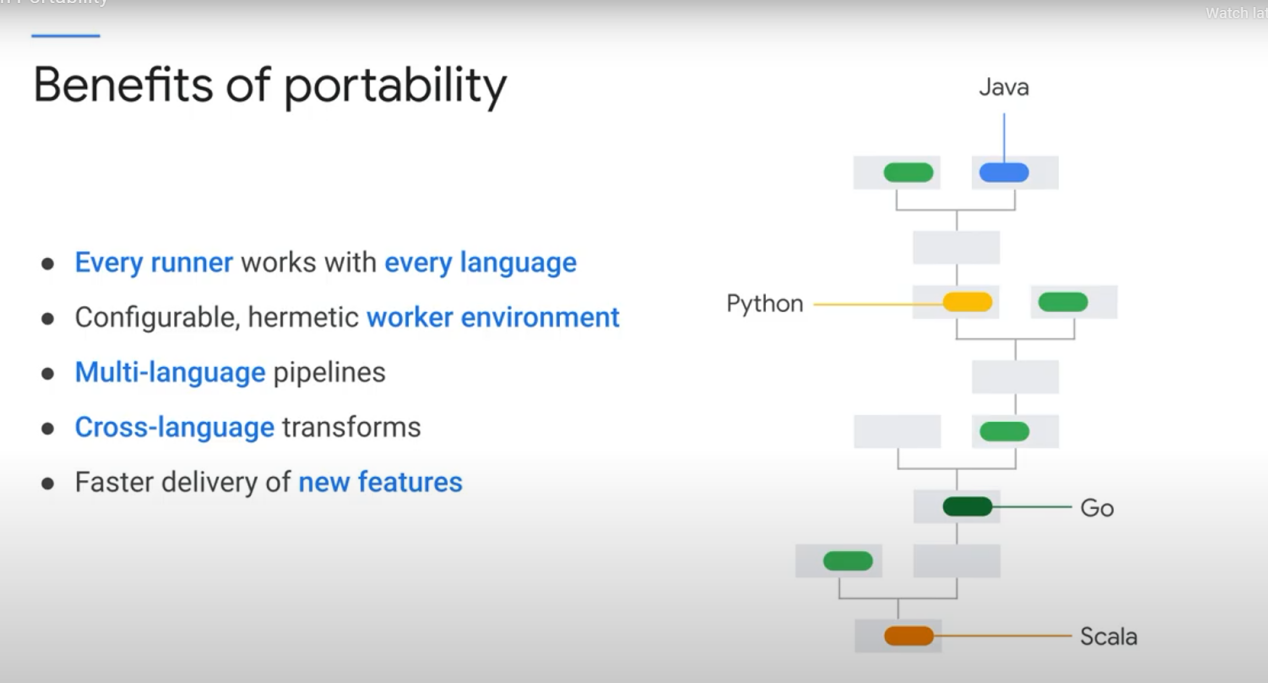




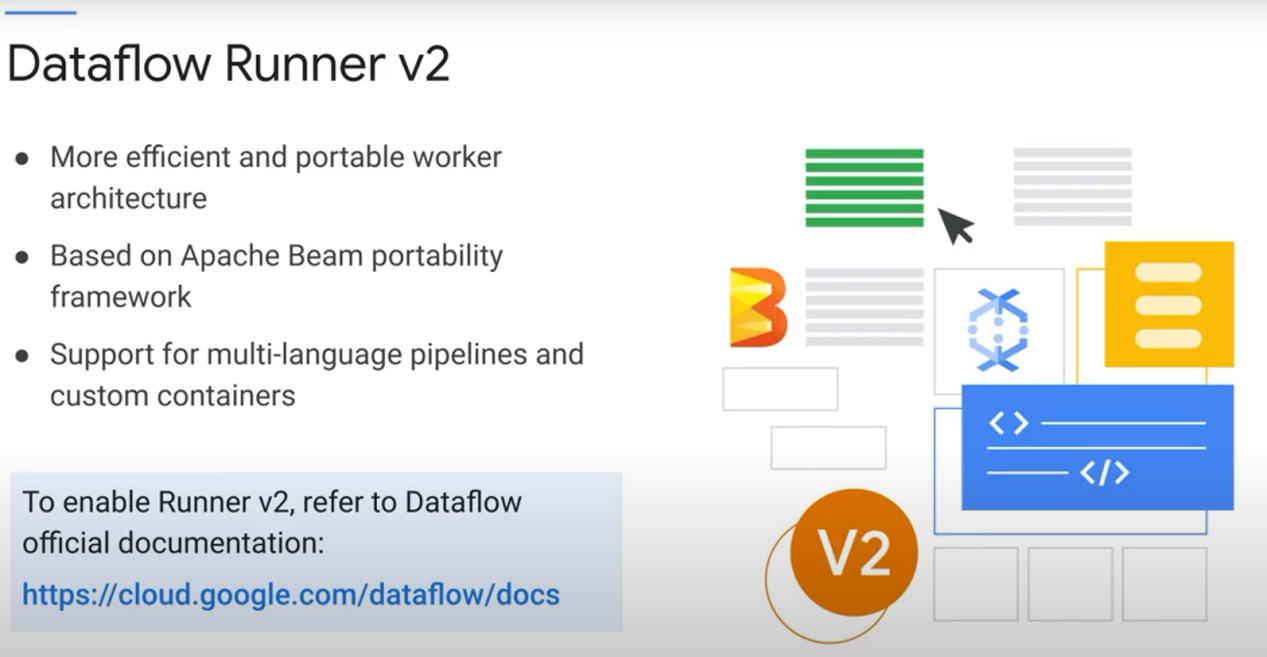
You can move your beam pipeline from your local machine (coded in python, java, or...), to google cloud or any other cloud.



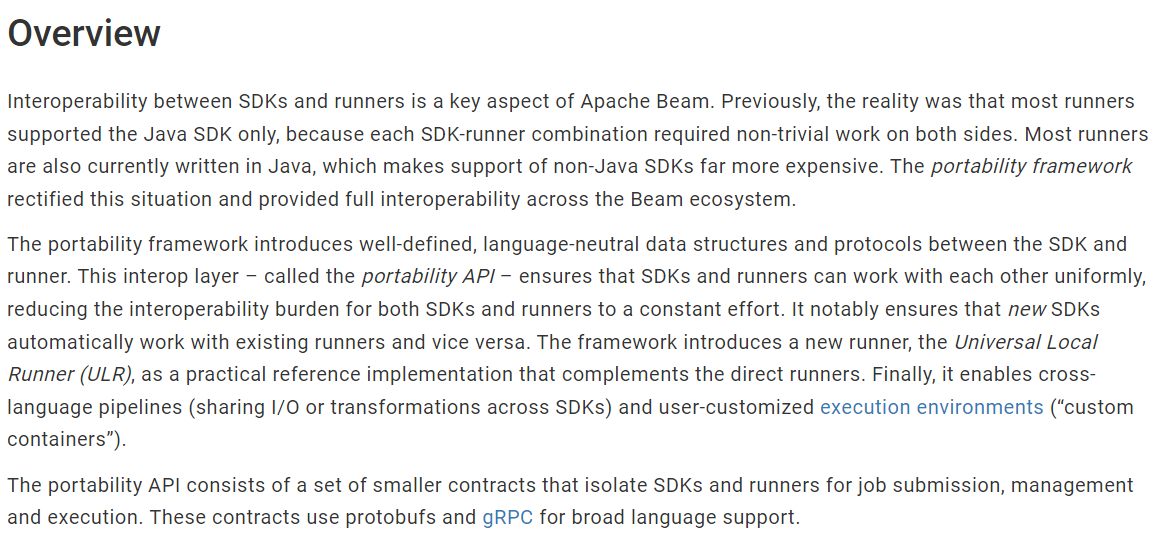
This means you can use any programming language you want to create the pipeline, and then execute it with any runner you want (including google cloud dataflow, etc...)



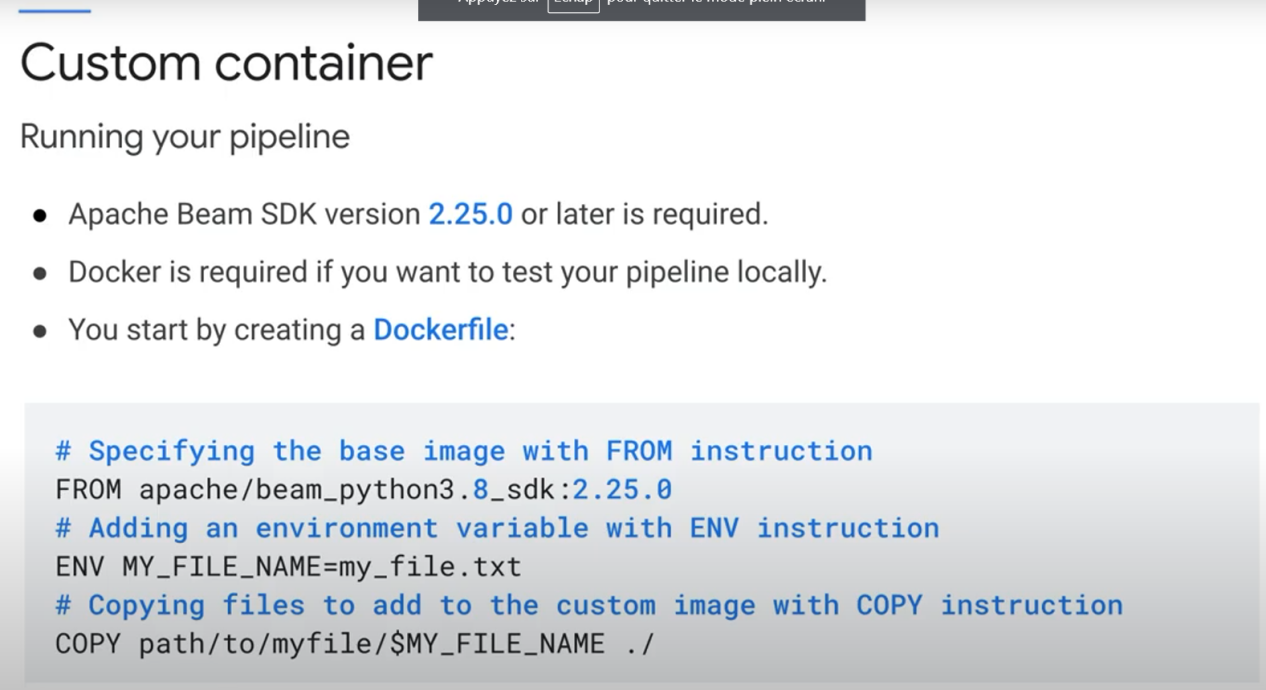
MANY PROGRAMMING LANGUAGES IN ONE PIPELINE!

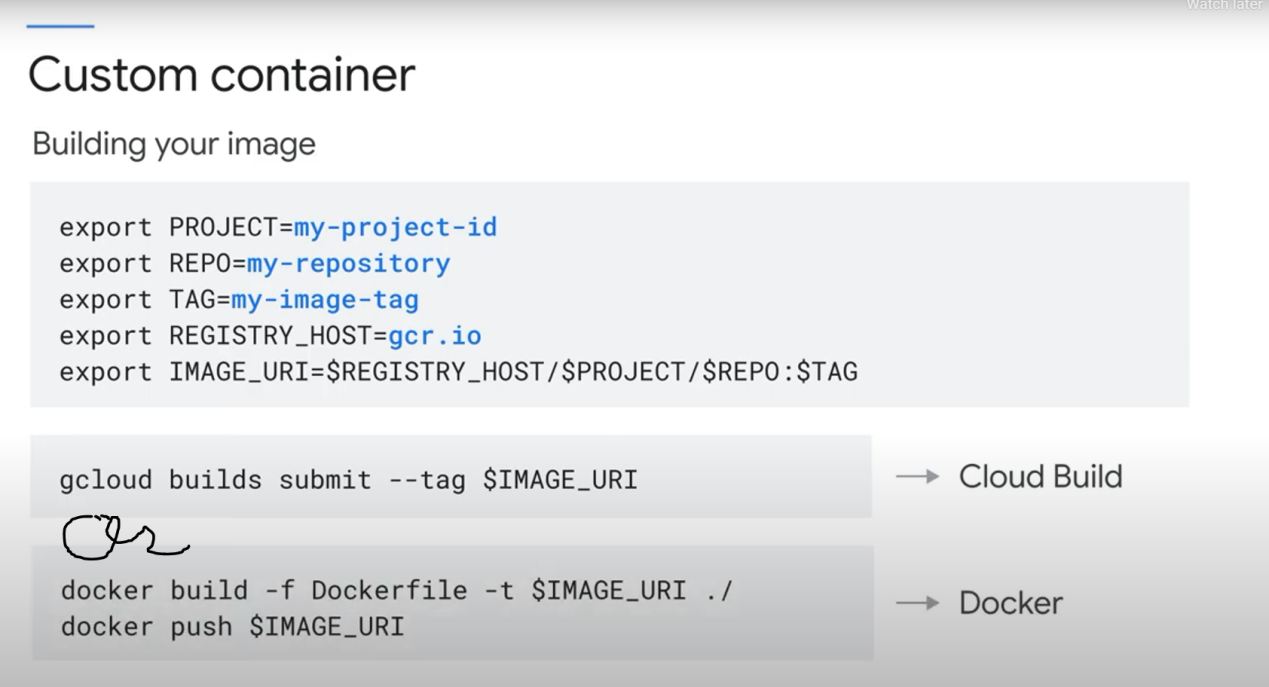


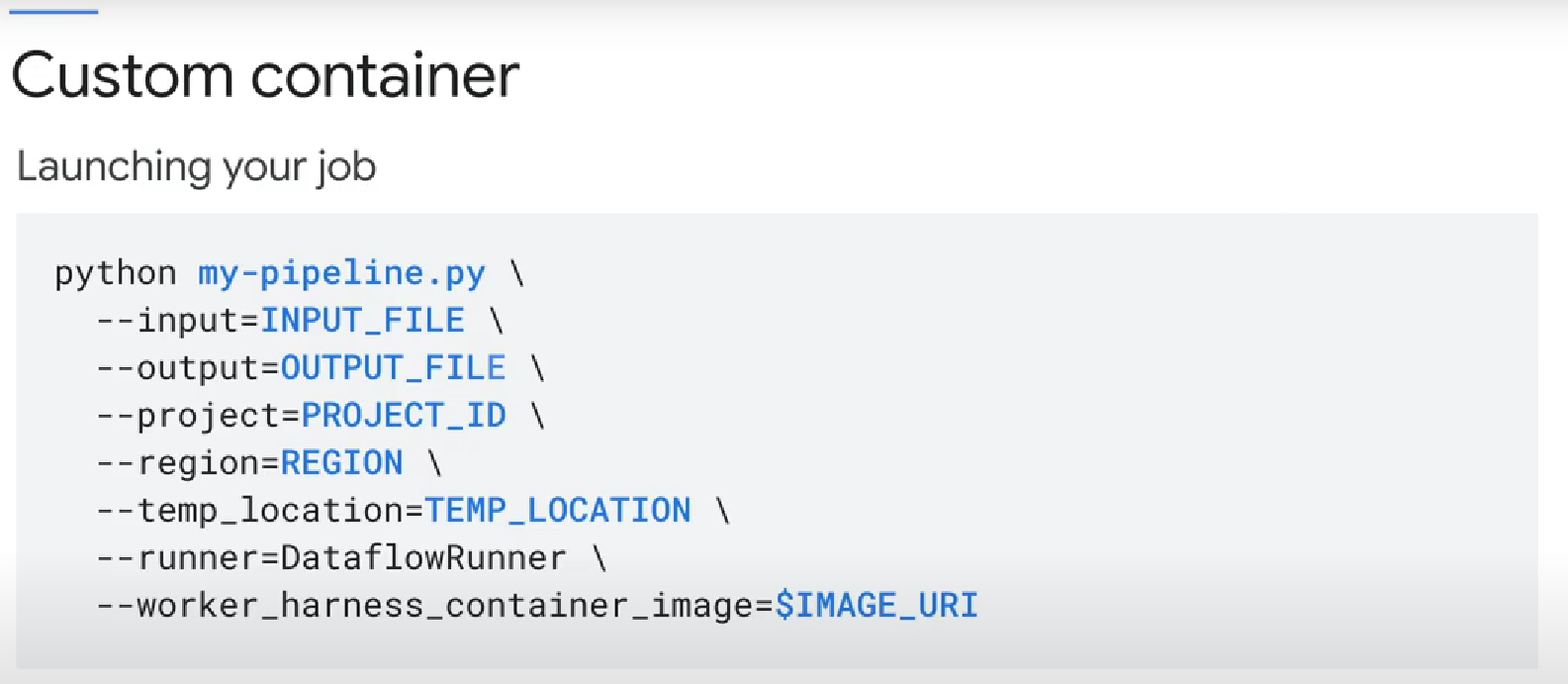
Bonus:



The **Beam SDK runtime** **environment** can be containerized with **Docker** to isolate it from other runtime systems. Each user operation has an associated environment in which to execute. Typically, supported SDKs provide a default environment that you can further customize. Because of containerization, you can benefit from ahead-of-time installation. You can include arbitrary dependencies, and even further customization is possible.



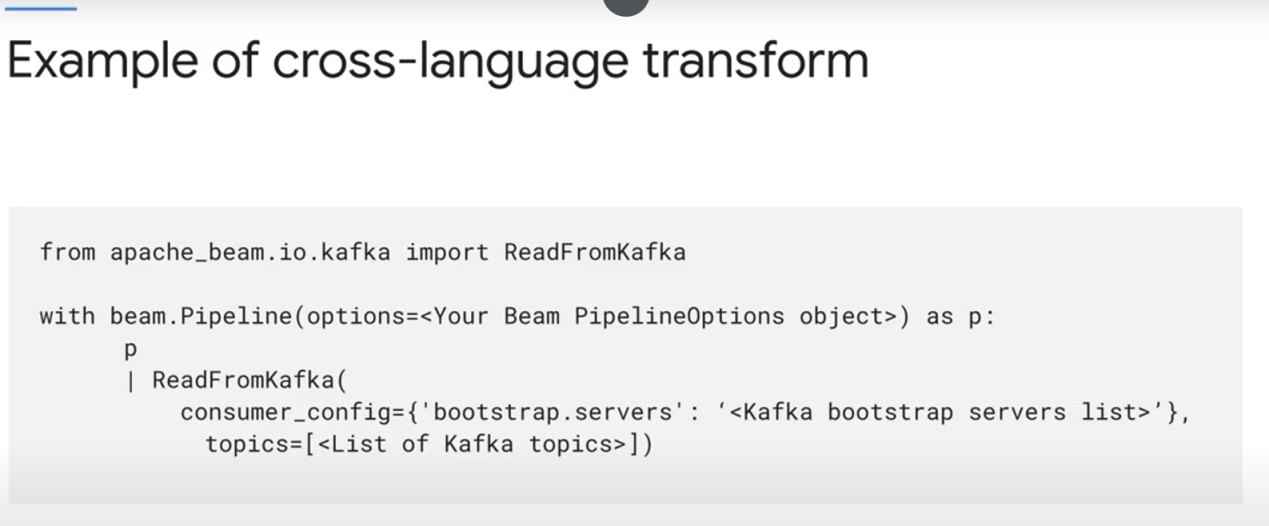


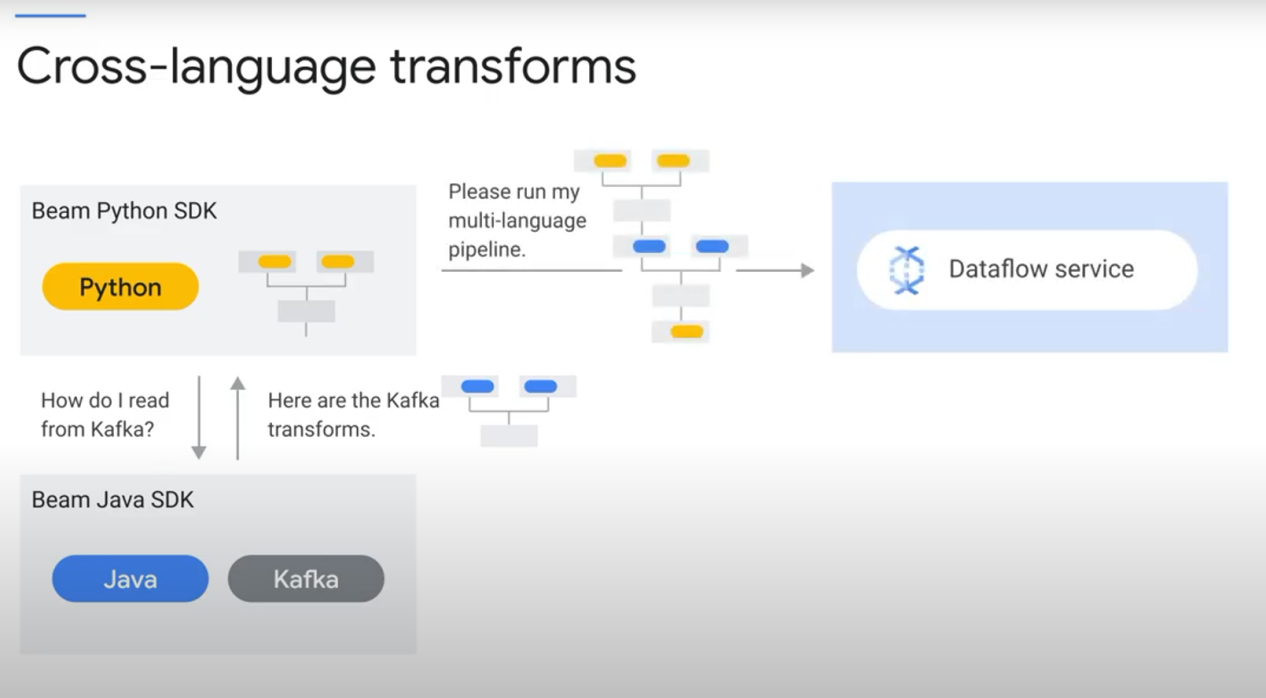


You can use a library that is only available in a language X, while coding your pipeline in language Y.

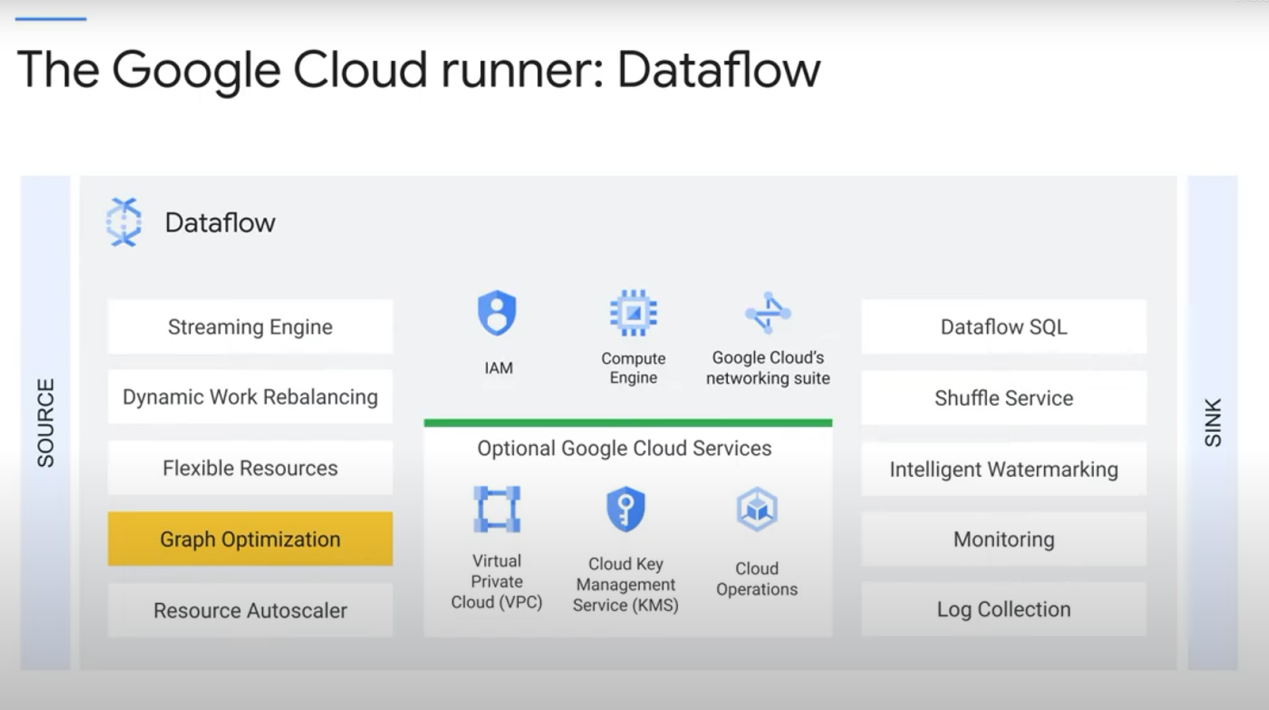
For instance, using tensorflow in a Java-defined pipeline:

E.g (python code, Kafka is a java transform)

How it works:

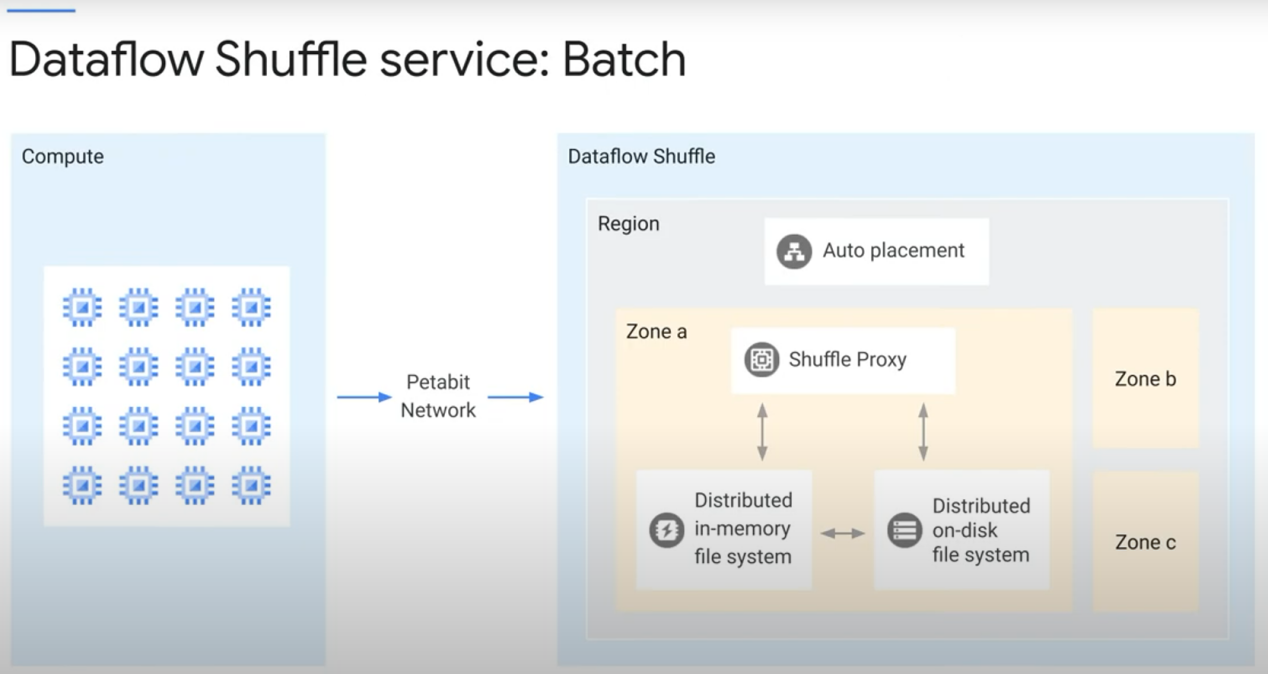


Why choose dataflow as a runner?

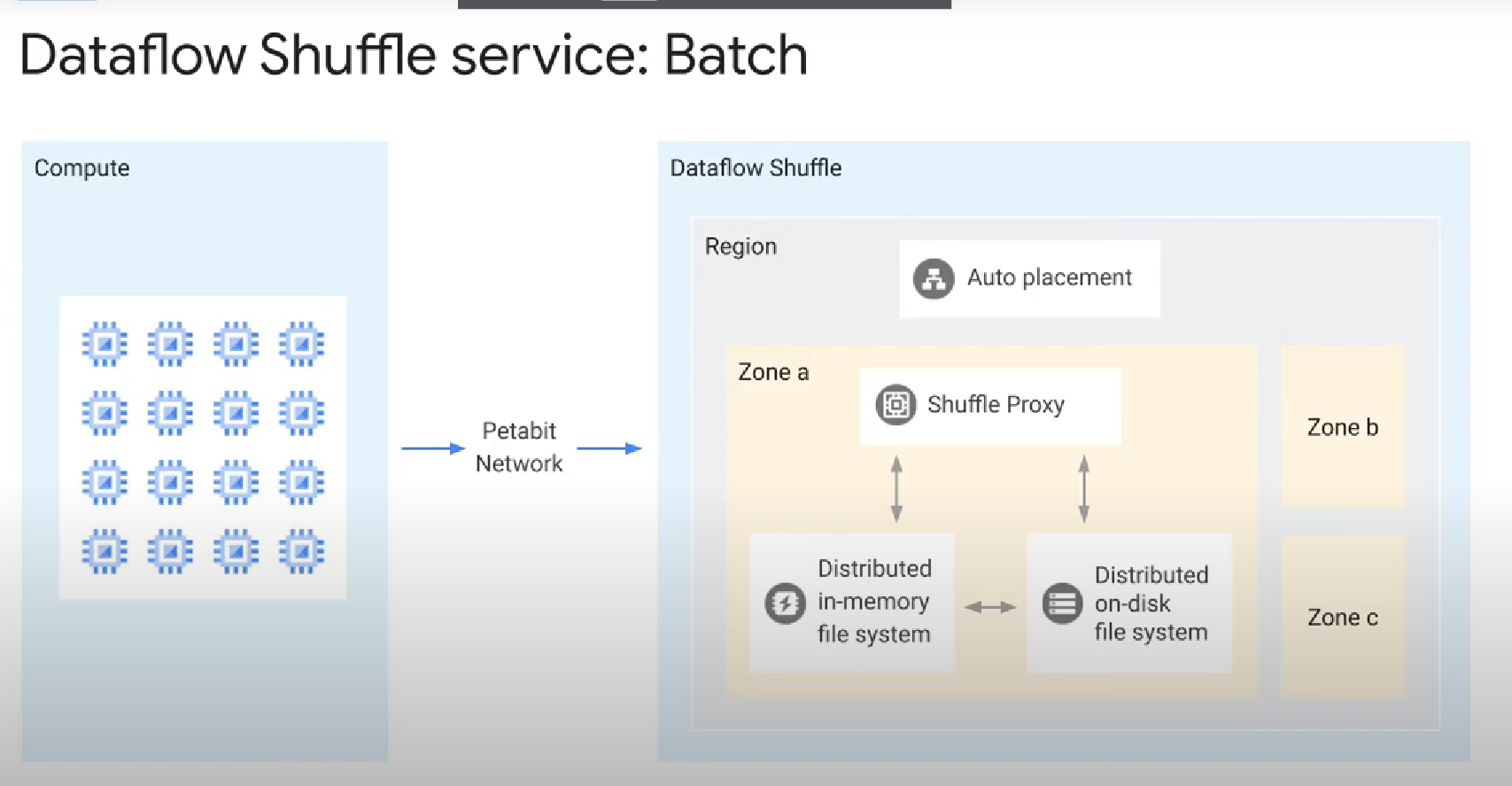


Dataflow optimizes the graph and automatically chooses which steps of the graph to run in parallel.

Dataflow shuffle service:



A shuffle is a Dataflow-based operation behind transforms such as GroupByKey, CoGroupByKey, and Combine.

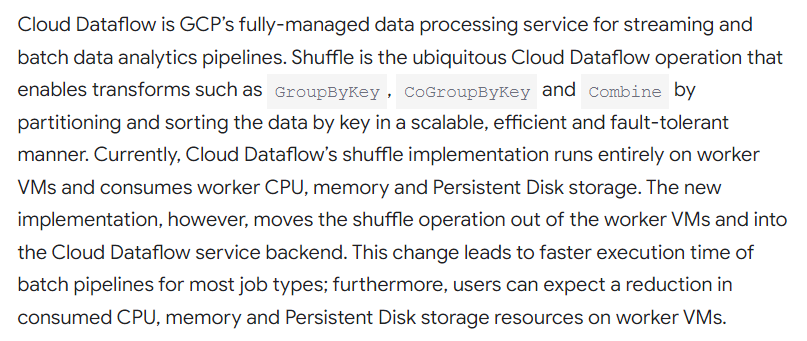


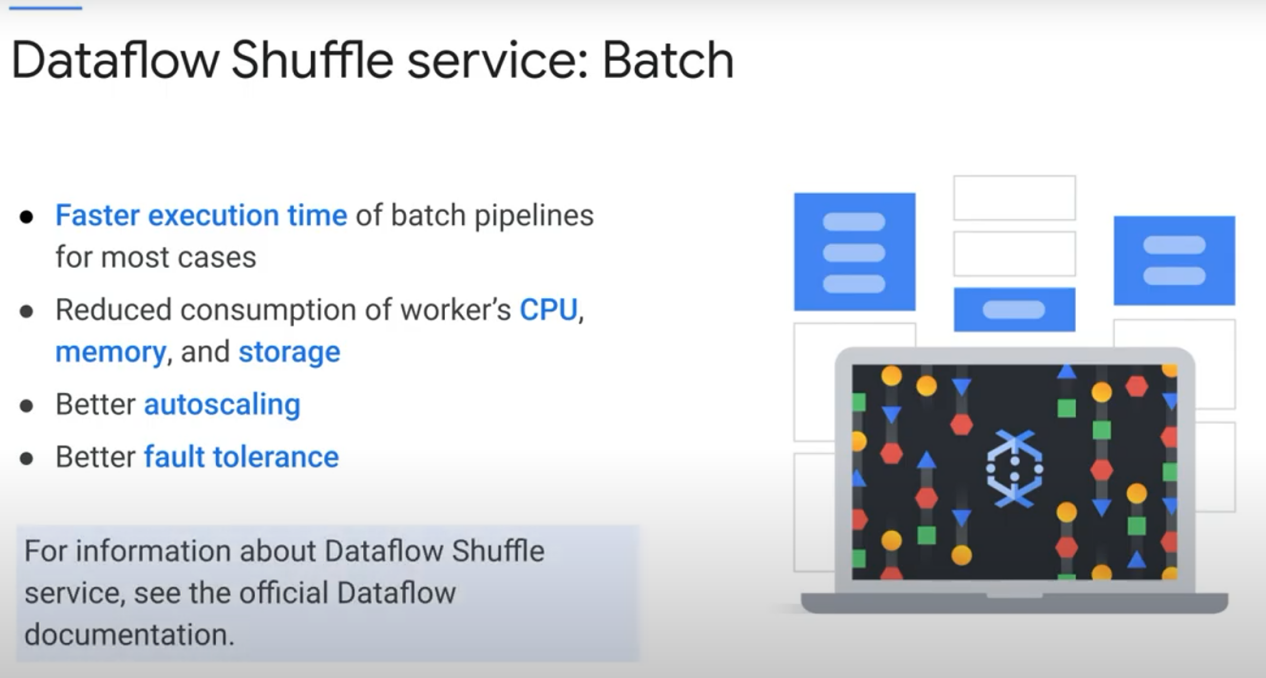
Currently, Dataflow uses a shuffle implementation that runs entirely on worker virtual machines

The service-based Dataflow Shuffle feature (or Dataflow service) is only available for batch pipelines only moves the shuffle operations out of the worker VMs and into the Dataflow service backend. With the Dataflow Shuffle service, you will have faster execution time of batch pipelines

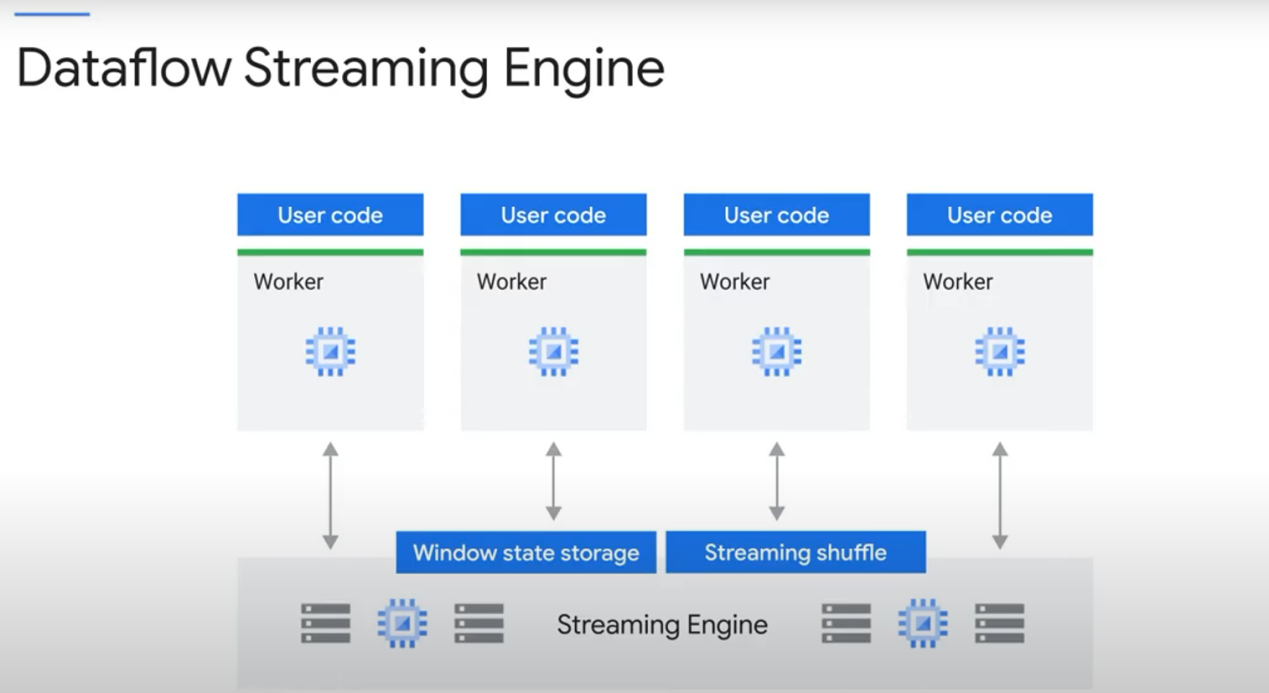
Note: I think the normal dataflow shuffle works in a VM for both batch and streaming pipelines. But the service-based dataflow shuffle is only for batch pipelines

More details:



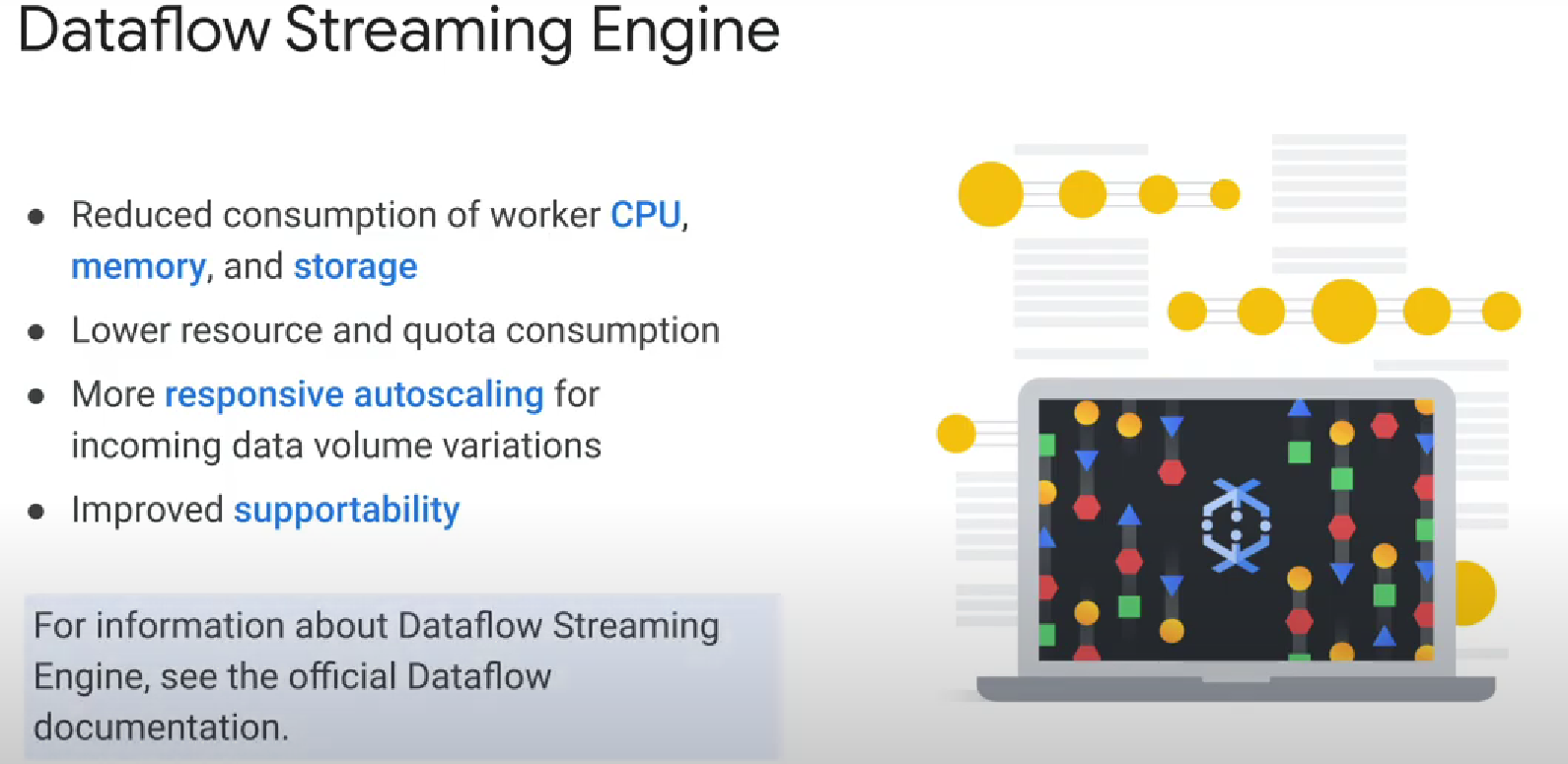


Streaming engine:

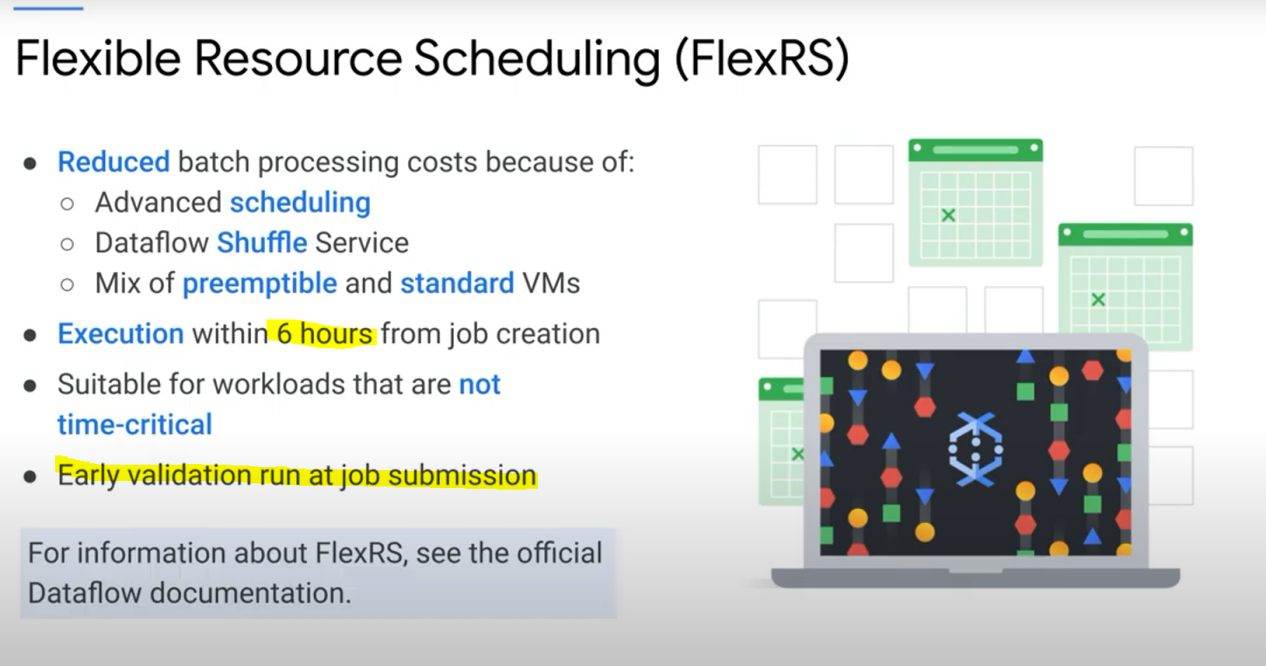


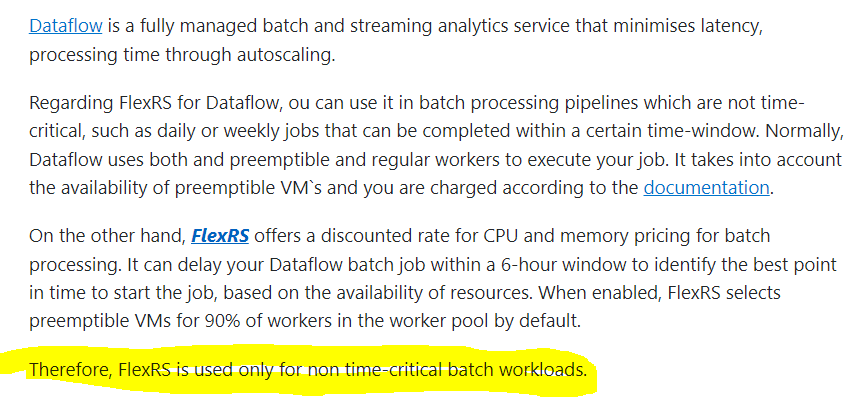
 the streaming engine offloads the window state storage from the persistent disks attached to worker VMs to a back-end service. It also implements an efficient shuffle for streaming cases.

Note: **Streaming engine works best with smaller worker machine types like n1-standard-2, and does not require persistent disks beyond a smaller worker boot disk.**

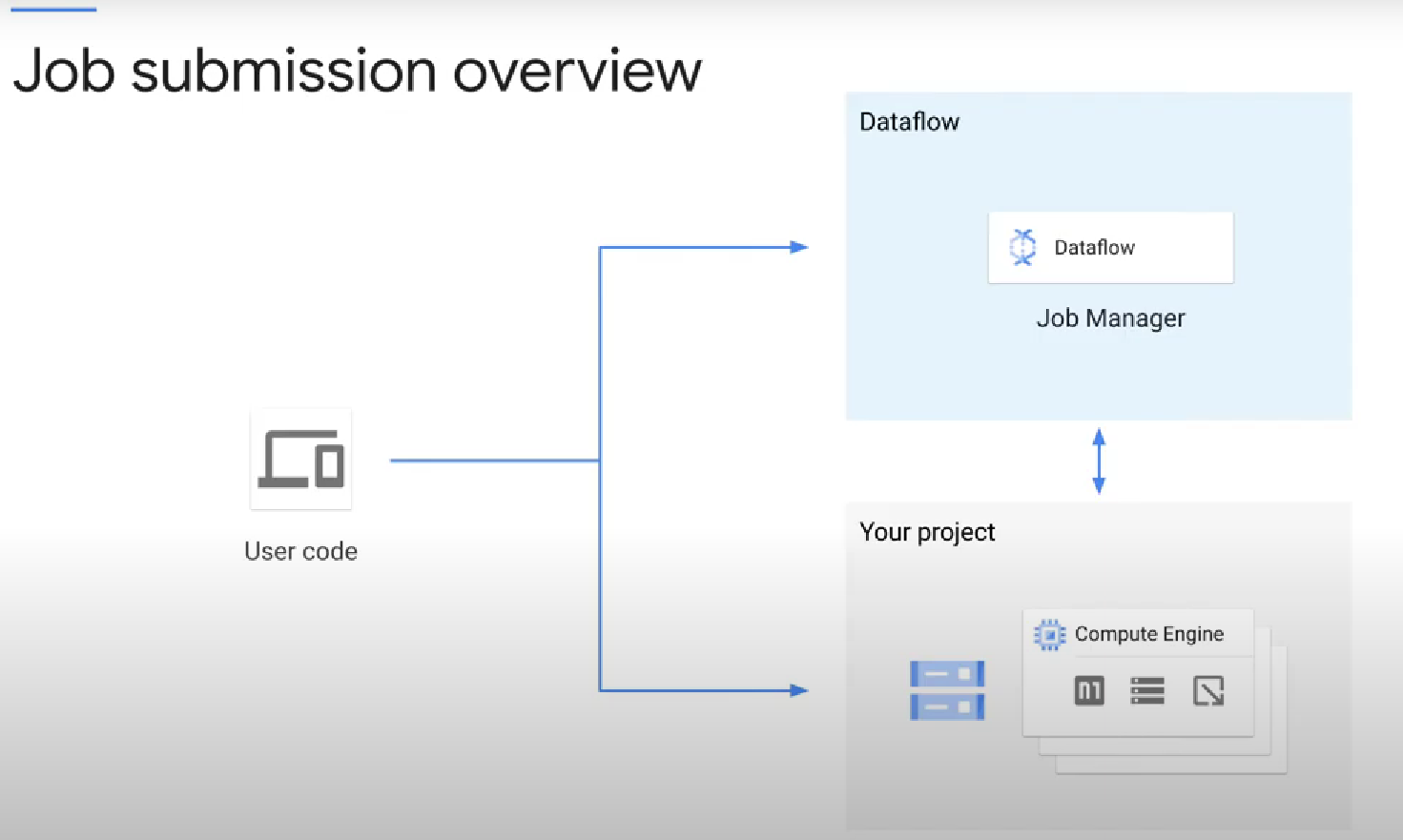


If your job is not time-critical AND IS A BATCH JOB, use **FlexRS.** FlexRS helps you reduce the cost of your batch processing pipelines because you can use advanced scheduling techniques in the Dataflow Shuffle Service and leverage a mix of preemptible and normal virtual machines.  When you submit a FlexRS job, the Dataflow service places the job into a queue; The execution of the job will happen within 6 hours of the job’s submission.

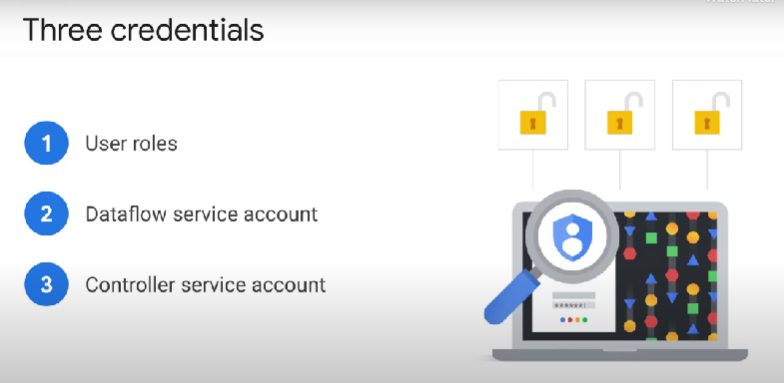


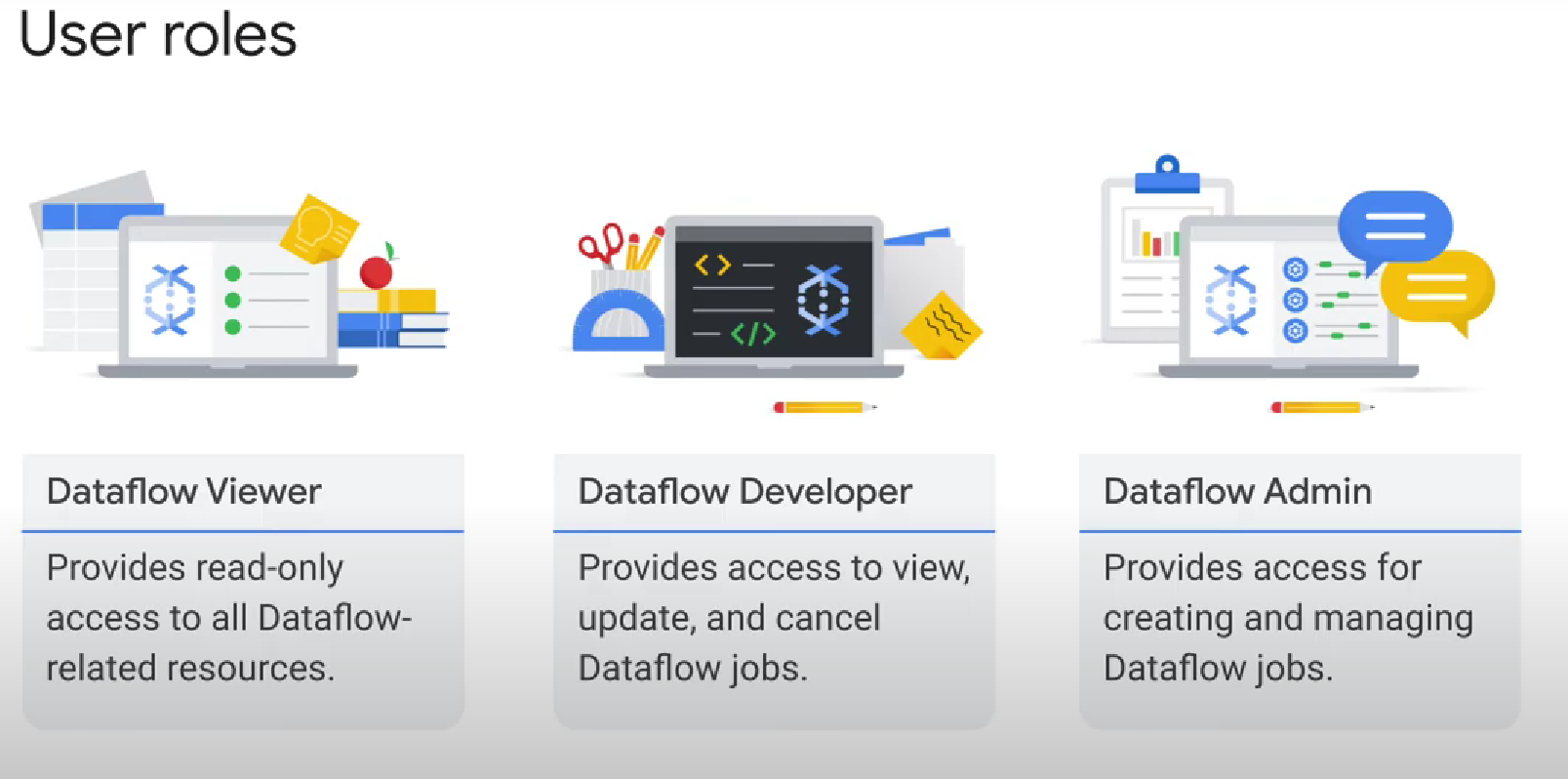


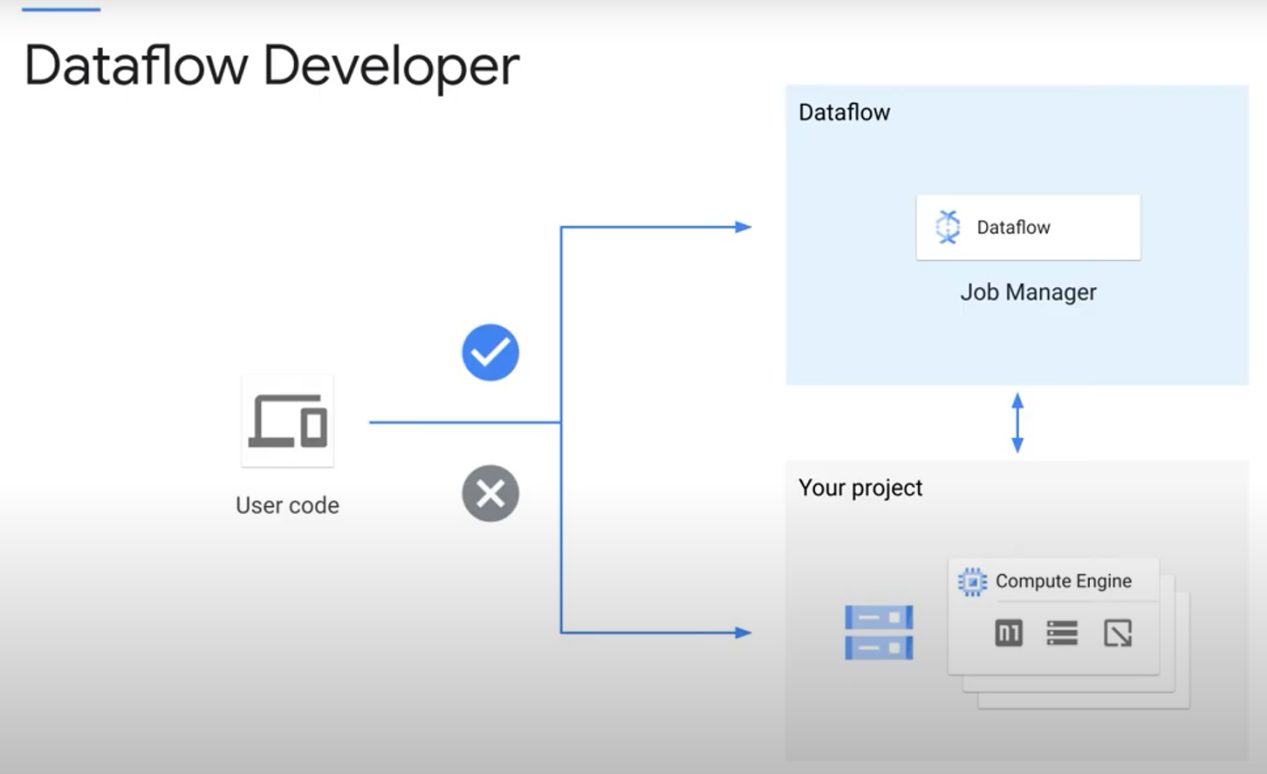
IAM:



Three user roles can be assigned to each user or group.





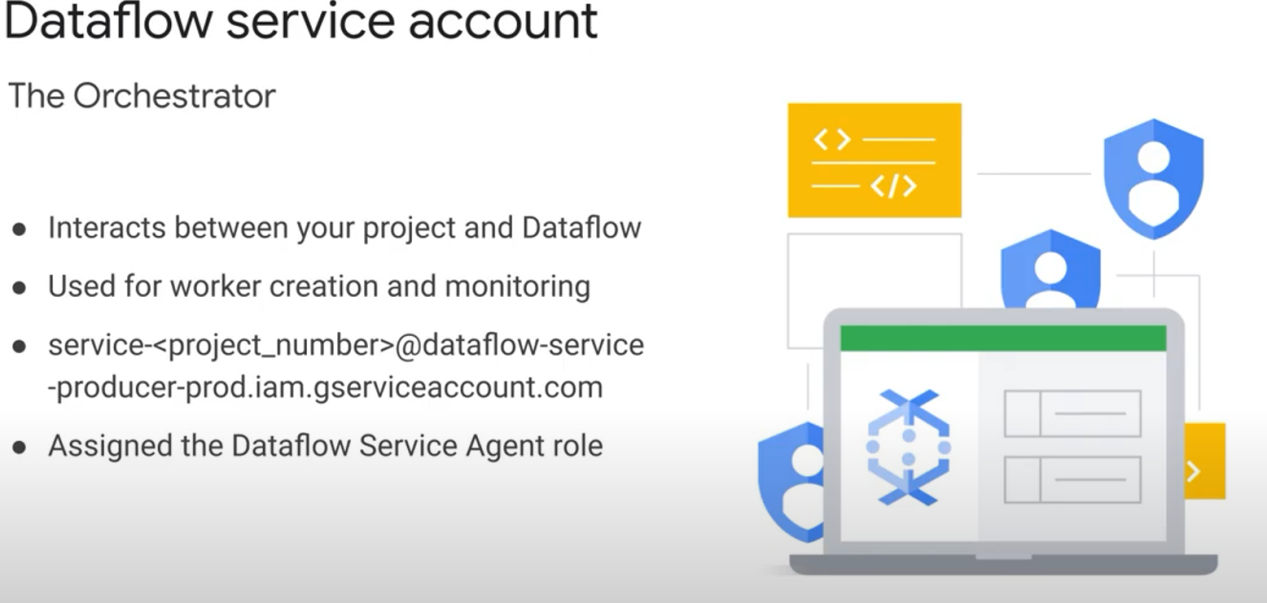


**You can also create a custom role.**

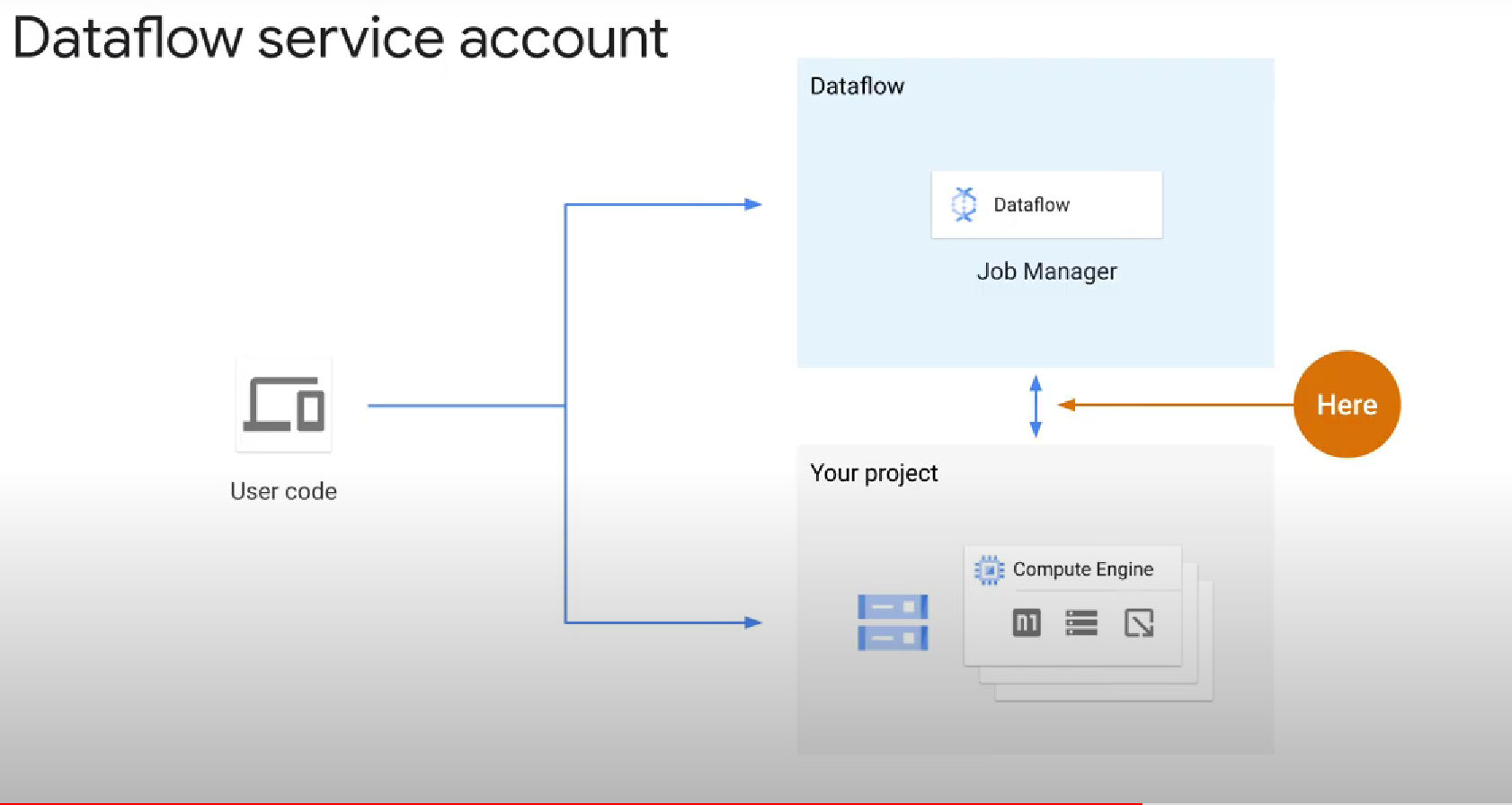
Dataflow developer’s role doesn’t allow you to create new jobs, nor to view the compute engine quota. Doesn’t have access to stage files.

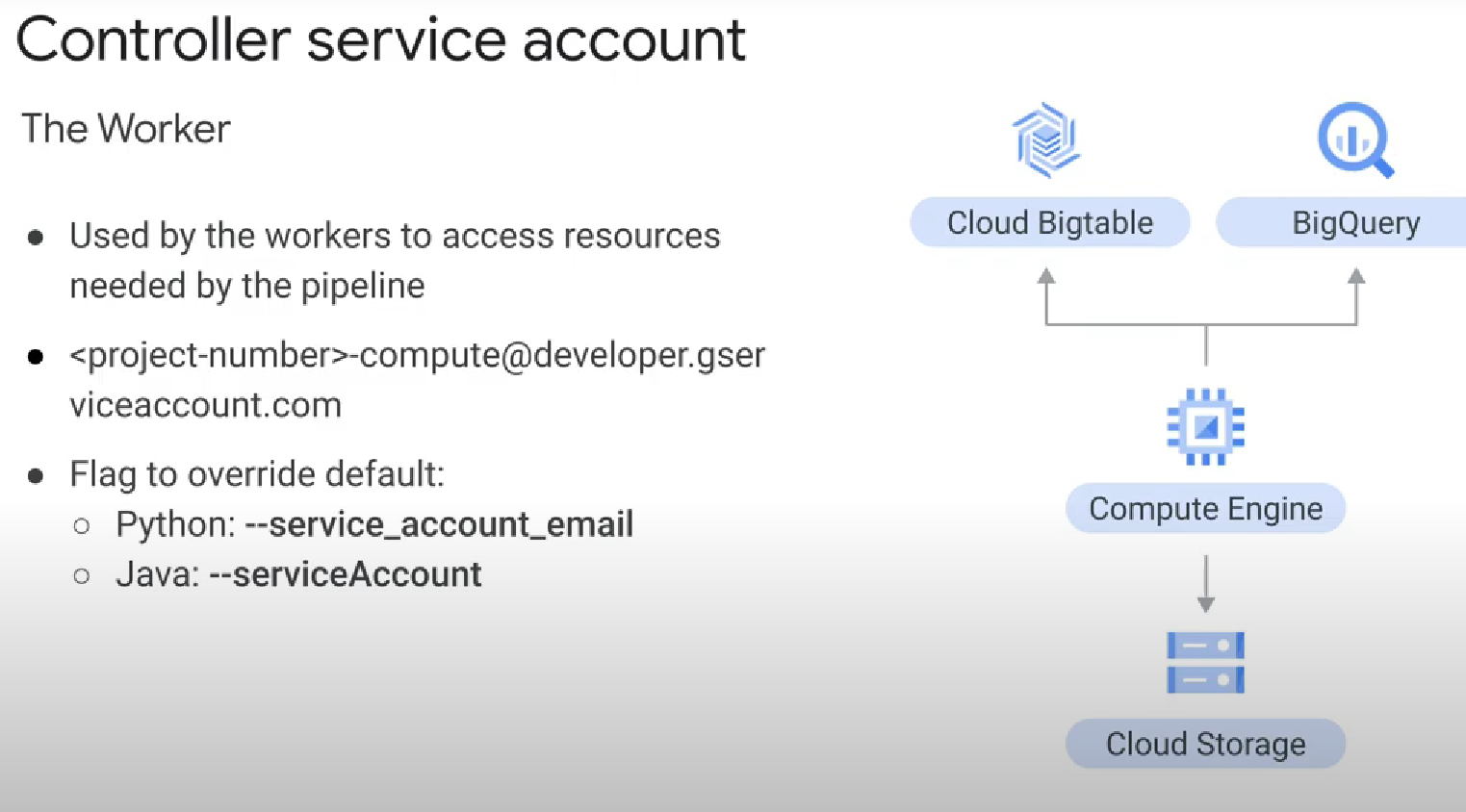


The second credential Dataflow uses is the Dataflow service account. Dataflow uses the Dataflow service account to interact between your project and Dataflow.



This account is automatically created when the Dataflow API is enabled. It is assigned the Dataflow service agent role and has the necessary permissions to run a Dataflow job in your project.





Finally, the third type of credentials is the controller service account:

This service account, <project-number>-compute @developer.gservices.com, is automatically created when you enable the Compute Engine API for your project

 The Compute Engine default service account has broad access to your project's resources, which makes it easy to get started with Dataflow. However, for production workloads, we recommend that you create a new service account with only the roles and permissions that you need.

At a minimum, your service account must have the Dataflow worker role and can be used by adding the service account email flag when launching a Dataflow pipeline.

Where does the controller service account work?:

