spectra

by Fosfor

Point of View

Multi Cloud & Hybrid Deployment

Introduction

Though cloud adoption is growing year on year, an increasingly greater number of organizations and technology leaders are concerned about the significant additional higher cost cloud brings in. This brilliant article by Sarah Wang and Martin Casado explains why it's important to not be married to one cloud and how "cloud repatriation" should be a design decision on day one of any cloud adoption journey.

This document explains how Spectra seamlessly supports multi-cloud and hybrid deployment models that lets the users take advantages of any cloud or on-premises environments without ever being locked in with any of them – and even with Spectra itself.

Key Concepts

Spectra is designed to support true hybrid deployments natively and seamlessly.
It does so by tactically separating the host environment and the execution engines.

Spectra Host Environment

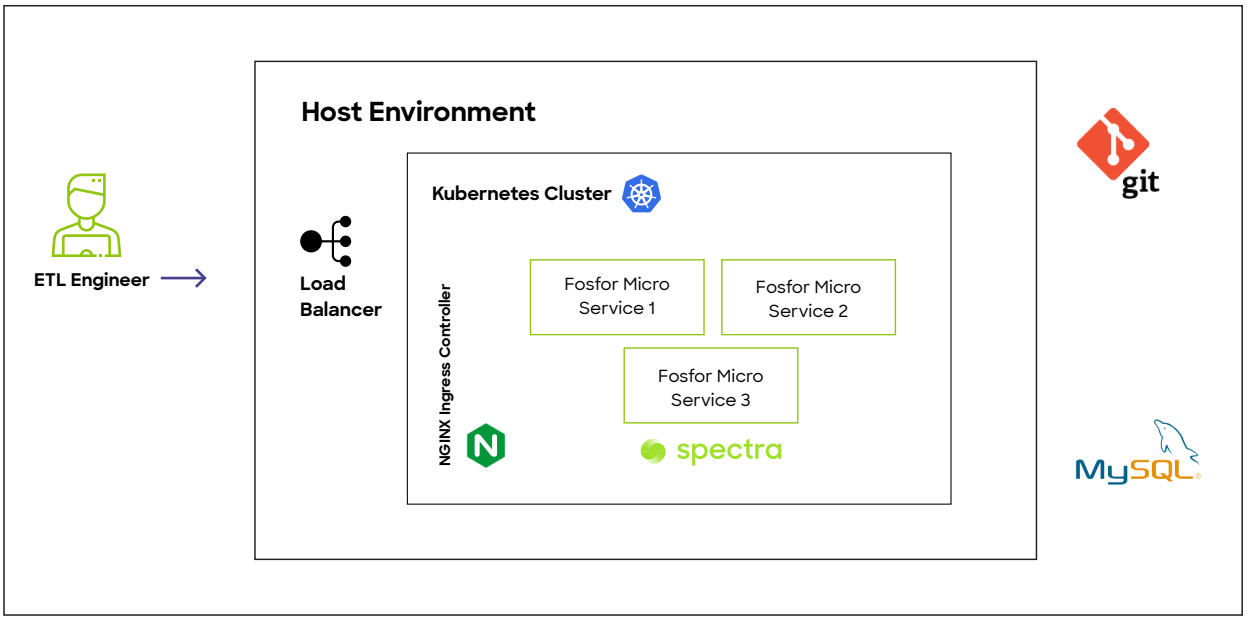


Fig.: Simplified representation of Spectra's host environment deployment

The Spectra's host application can be hosted anywhere - on any cloud or on-premises. There's only one host application per Spectra installation.

All ETL pipelines designed using the Spectra's designer tool are versioned in Git. It can use any popular Git implementation technologies that the organization has adopted, e.g., GitHub, Gitlab, or Bitbucket.

All other metadata including audit trails, RBAC controls, run history, etc. are stored into a centralized meta store DB (typically MySQL or Postgres).

Execution Engine

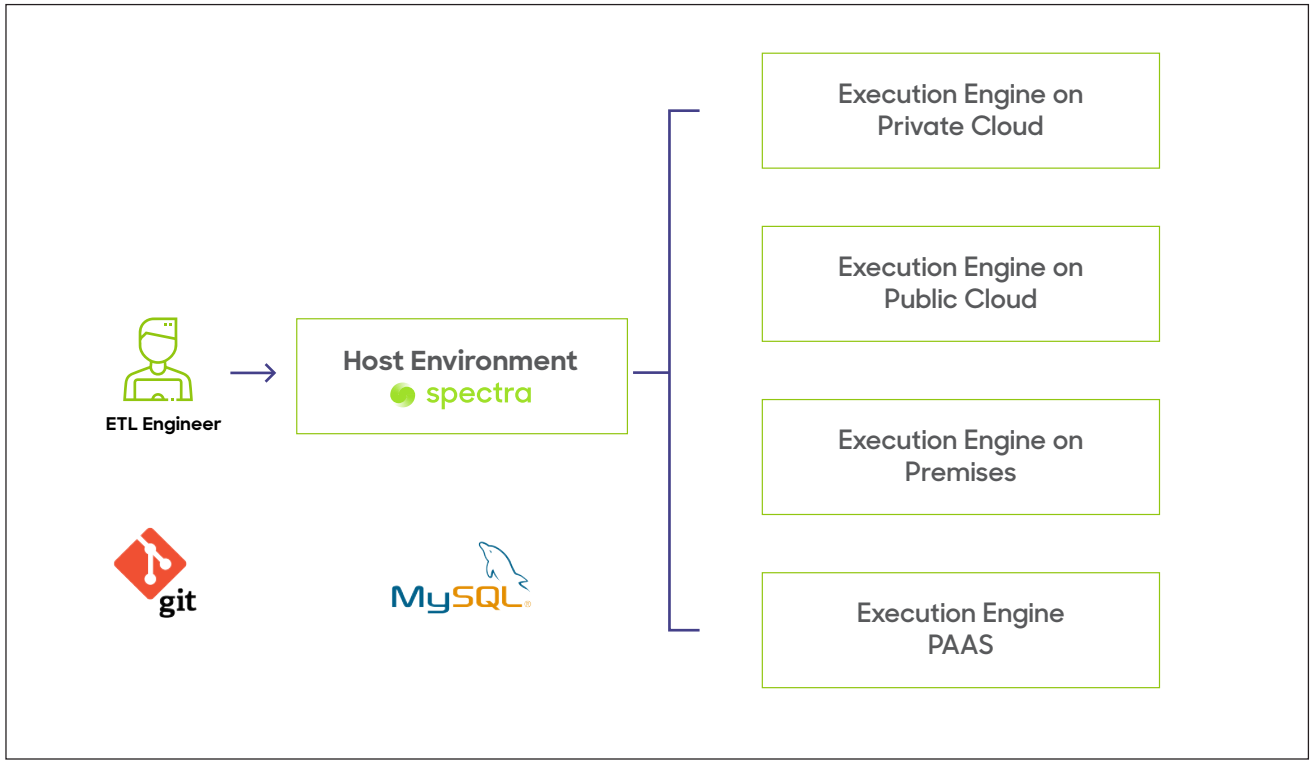


Fig.: Spectra execution engine types

One-to-many execution engines can be used, and they can be hosted anywhere on any cloud or on-premises. The host environment connects and pushes the ETL workload onto them, and no data is persisted in the execution engine. That makes it easy to add and remove them as needed, on the fly.

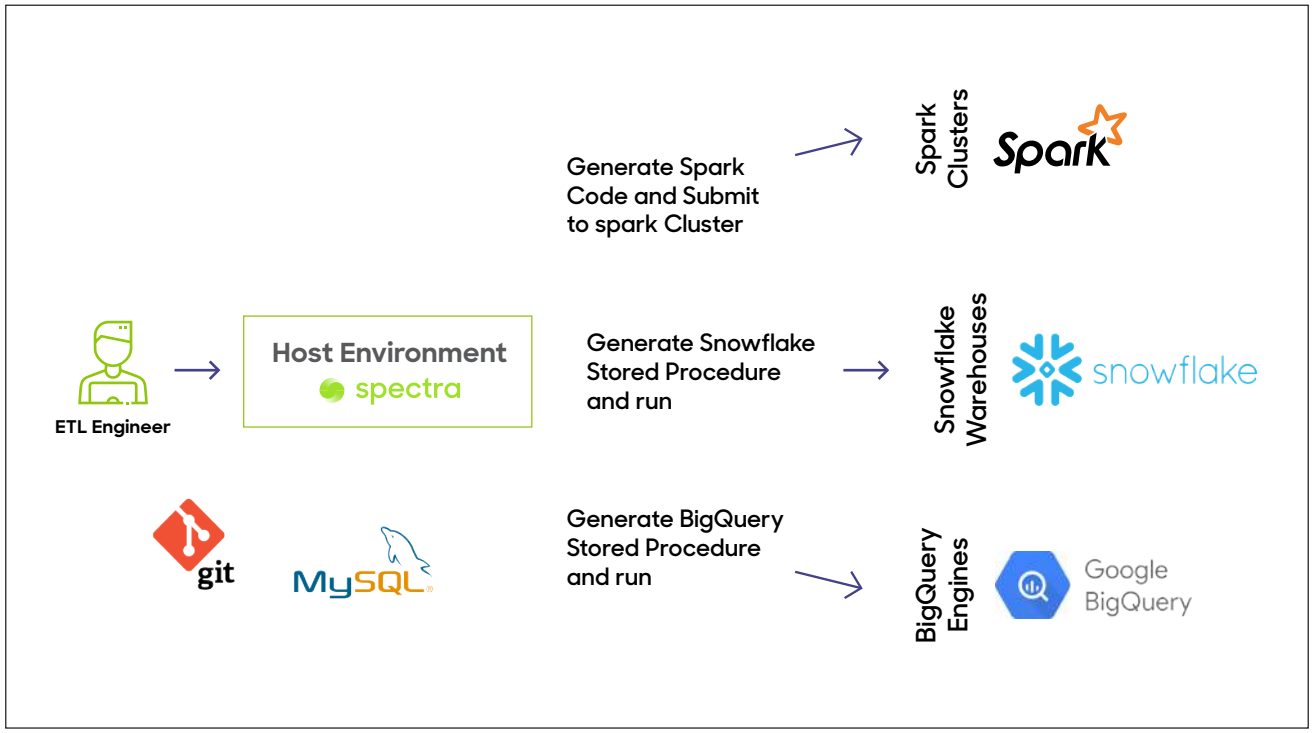


Fig.: Examples of execution engines

Spectra supports different types of execution engines; the host application generates the ETL code based on the type of execution engines. For example, for Spark-based execution engines such as Databricks, EMR, HDInsight, etc., the host application generates Spark code (on Java) and pushes it to the execution engine via Livy end points. Similarly for Snowflake execution engine, the host application generates Snowflake stored procedures and runs it.

Hybrid Implementation Examples

Here are a few sample hybrid implementations which show how a common host environment supports different types of execution engines hosted on different clouds and / or on premises.

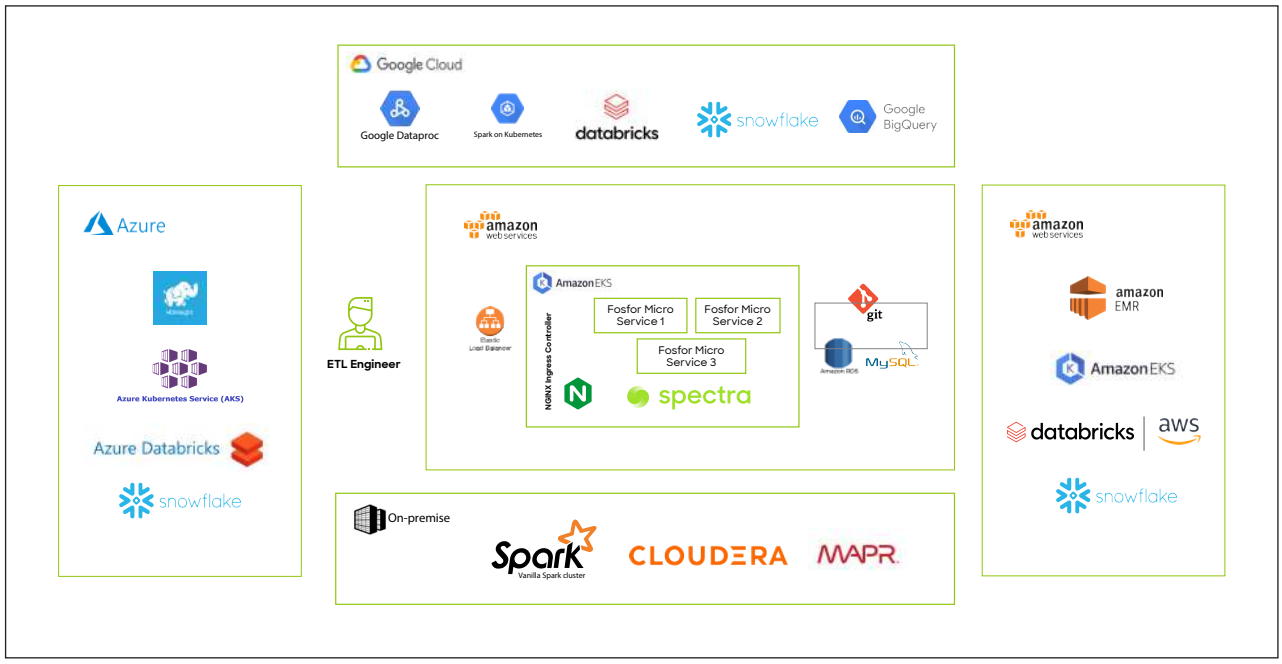


Fig.: Hybrid execution engine example with host environment in AWS

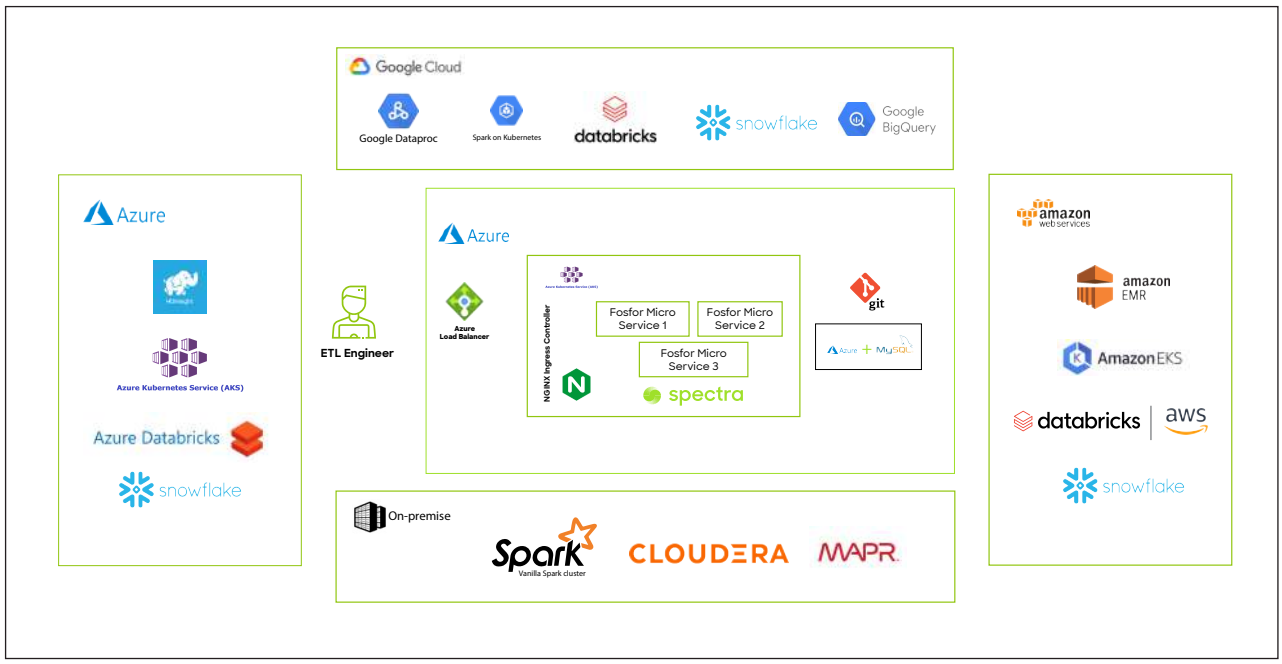


Fig.: Hybrid execution engine example with host environment in Azure

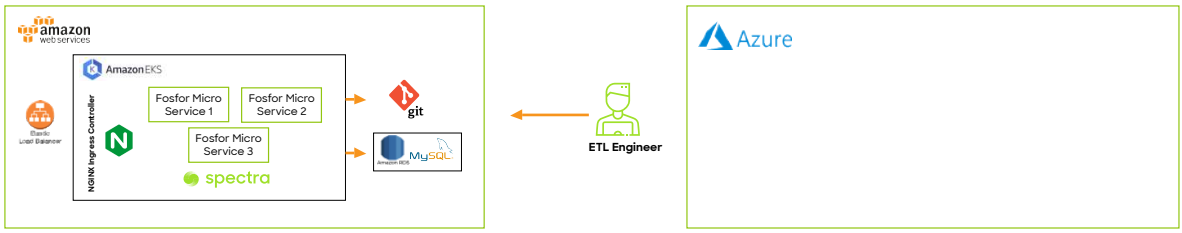
Choosing the Right Host Environment

Though one can choose to deploy the host application anywhere, it's always a better idea to choose your primary cloud as the host environment. It's also important to consider the connectivity from the host environment to all execution engines that the organization is planning to use.

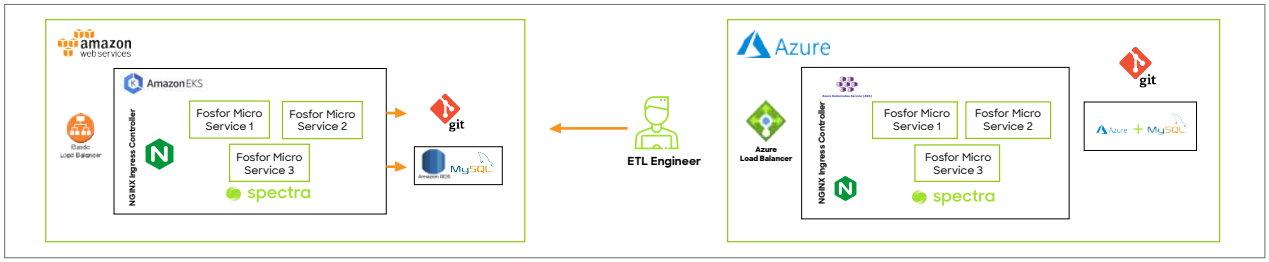
However, it's practically very easy to lift and shift the host application from one cloud to another or to on-premises – thanks to Spectra's metadata-driven design. Here's how that's done.

Illustration of How to Move the Host Application From AWS to Azure

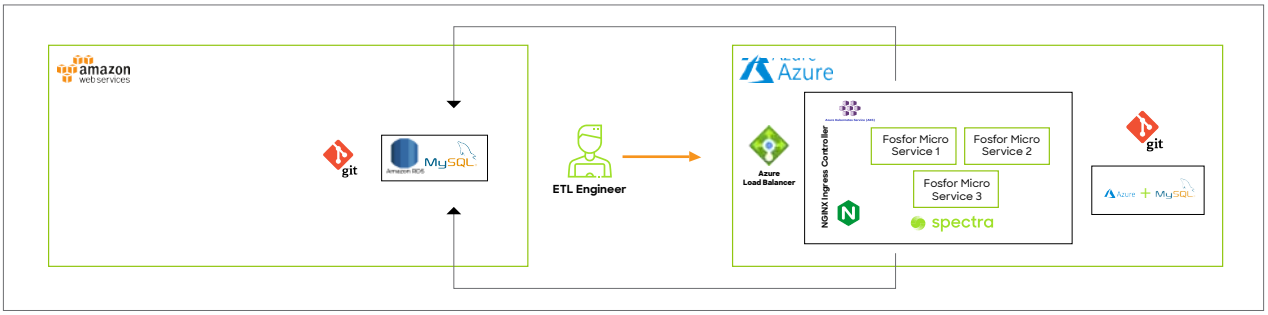
Initial state: User is connected to the host application deployed in AWS



Step 1: Install Spectra host application on Azure



Step 2: Configure the new host application to point to the meta-store and Git hosted in AWS



That's it! The user can now start using the new host environment on Azure.

Now if the user wants to completely migrate off AWS, there's the optional step of one-time movement of metadata and GIT repositories to Azure and configure the host application to point to the new meta-store and git. This way, the dependency on AWS can be completely removed.

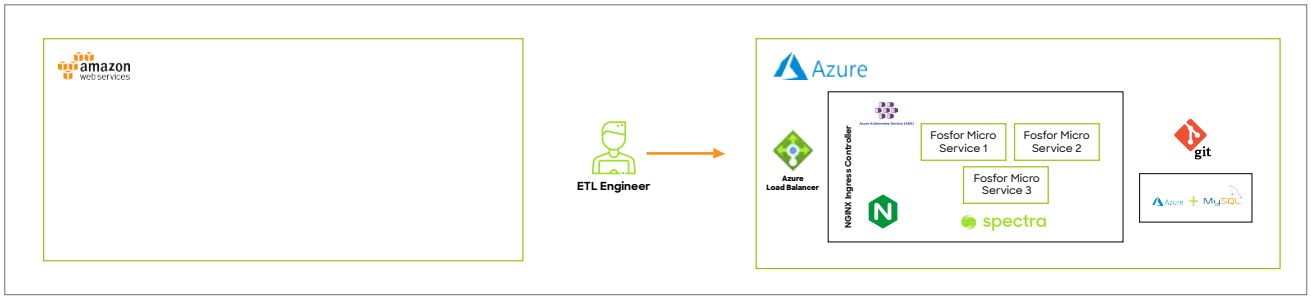


Fig.: Similar steps can be followed to move the host application between any cloud or on-premises environment

Choosing the Right Execution Engines

There are the following three key levers to consider while choosing the right execution engines.

Efficiency: Pick the right tech for the right job

Proximity: Bring the party to your data

Cost: Sometimes cheaper is better

Efficiency

It's always a good idea to choose the best fit technology for the given type of problem. For example, Spark is better suited for processing and transforming huge data from different types of sources that involves complex joins. However, in many cases when the source and target are already the Snowflake tables, it makes sense to push the ETL workload too, to Snowflake.

Proximity

Choosing the execution engine closer to the data often saves a significant data transfer over the network and thus results in better performance. For example, if you need to read a huge dataset from S3, transform and push it to Redshift, it makes sense to select an execution engine from AWS and from the same region.

Cost

Doing a cost comparison on multiple execution engines helps identify the best option in most of the cases. And this may even surprise you too. For example, we have found that Spark on AKS is cheaper than HDInsight and is often faster too.

Switching Between Execution Engines

Switching between execution engines is as easy as clicking a few buttons. The user can configure different execution engines in Fosfor Manager and can select them while running the flow in Spectra.

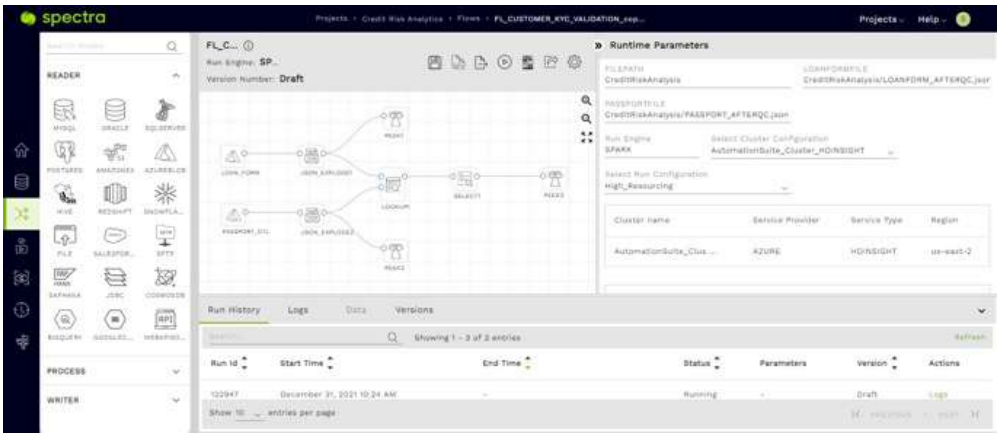


Fig.: While running a flow, it shows the default execution engine

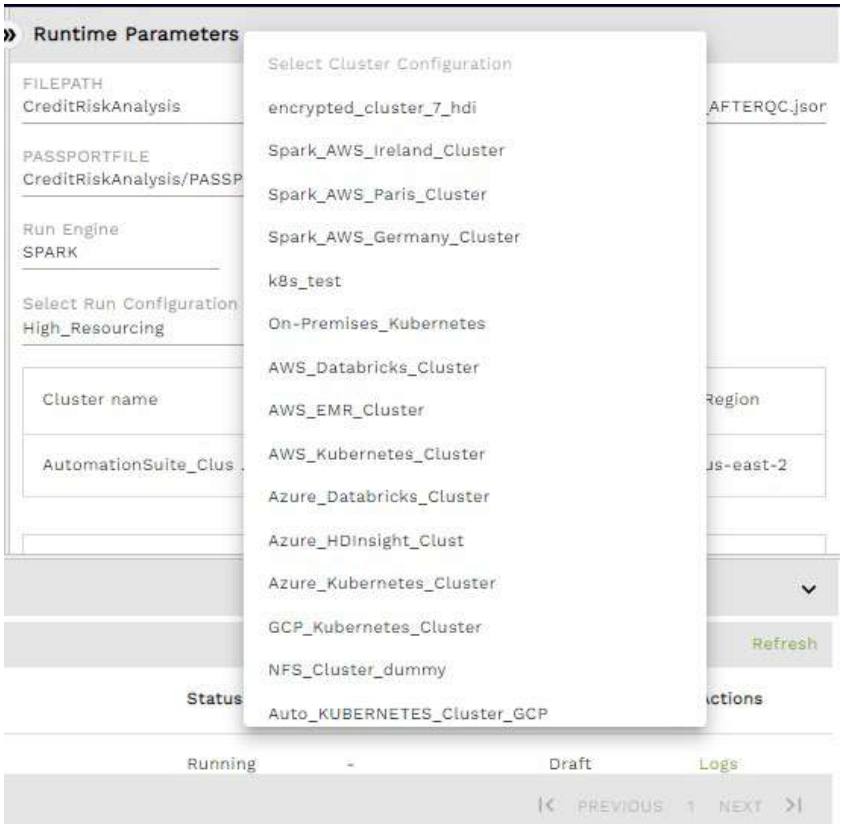


Fig.: The user can change the execution engine by selecting the configured engines from the drop down

Spectra's intuitive user interface and support for on-demand execution engine switching, makes it easy for the user to experiment with different execution engines to find the one that is the best fit for a given job. It also makes it easier for the user to switch to a new engine later if that makes sense for the organization.

This feature is also invaluable in the event that the organization decides to "repatriate" to on-premises hosting due to increasing cloud costs.

Switching DataOps Platforms

While Spectra eliminates the concerns of being locked into any cloud and makes it easy for the user to stay independent of the execution engines, what happens in the unlikely event that a better DataOps platform than Spectra emerges in the future? Spectra addresses this concern by providing the user with the native ETL code upon the event of a contract termination. Spectra provides the execution engine specific native code for every ETL job ever designed and run on Spectra, at a minimal additional cost.

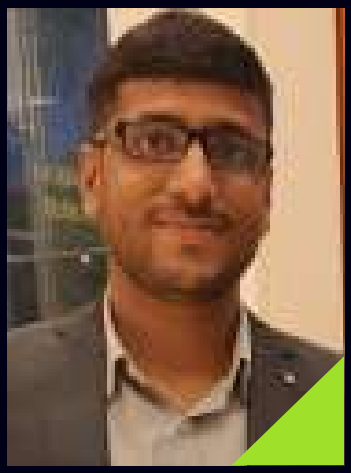
For example, for ETL jobs designed on Spectra and pushed to Spark engines, Spectra will provide the Spark code (on Java) and for the ETL jobs pushed to Snowflake engines, Spectra will provide the Snowflake-stored procedure.

This eliminates the concern of ever being locked into Spectra.

Conclusion

While aggressively adopting new technologies and clouds, modern-day, data-driven enterprises are trying to avoid getting locked into any of them and to keep exploring "the new and the better" in this ever-evolving technology space. Spectra is the perfect tool to make this possible.

By Gireesh Puthumana



Gireesh is an Associate Director, Product Engineering and Innovation leader at Fosfor. He comes with extensive startup experience and has mastered the art of building products and teams from grounds up. He is passionate about open-source technologies and often plays the role of key architect of Fosfor. With his profound product insight and technology expertise, he has spearheaded many successful client adoption journeys. He is a lean startup practitioner who focuses on smaller result-oriented iterations both in product development and in solving enterprise data and analytics problems

Associate Director,
Product Engineering

The Fosfor Product Suite is the only end-to-end suite for optimizing all aspects of the data-to-decisions lifecycle. Fosfor helps you make better decisions, ensuring you have the right data in more hands in the fastest time possible. The Fosfor Product Suite is made up of Spectra, a comprehensive DataOps platform; Optic, a data fabric to facilitate data discovery-to-consumption journeys; Refract, a Data Science and MLOps platform; Aspect, a no-code unstructured data processing platform; and Lumin, an augmented analytics platform. Taken together, the Fosfor suite helps businesses discover the hidden value in their data. The Fosfor Data Products Unit is part of LTI, a global technology consulting and digital solutions company with hundreds of clients and operations in 31 countries. For more information, visit Fosfor.com.

FOSFOR by LTI