Continuus TECHNOLOGIES

CASE STUDY

Optimizing Cost Efficiency with Scalable Snowflake Pipelines Using Kubernetes

Continuus implemented Kubernetes to enable real-time scalability for Snowflake pipelines, resulting in reduced idle costs and maximized efficiency that saved the client thousands in AWS EC2 expenses.

AT A GLANCE

ABOUT THE CLIENT

Site Reliability Engineering/DevOps department at an asset management, capital market, and investment management organization

PROBLEM

Client's pipelines were costing money when idle and lacked efficiency during periods of high workload

SOLUTION

Implemented Kubernetes to create scalable pipelines that adjust based on workload, reducing idle costs and enhancing efficiency

OUTCOME

Scalable pipelines saved the client thousands in AWS EC2 costs by efficiently managing resource usage during varying workload periods

FRAMEWORK

- DevOps and CI/CD, Configuration Management
- Snowflake, Kubernetes, Terraform, HashiCorp, AWS, GitLab

PROBLEM

Following our stand up of the AWS infrastructure and Snowpipe integration for the client using Terraform, it was identified that the pipelines were costing the company money when those pipelines sat idle and were not running efficiently during heavy workload periods. We created pipelines that scale by workload using Kubernetes to remove the need to manage a server instance across environments.

SOLUTION

Step 1: Define the Stack in Terraform. We began with the out-of-the-box solution we created previously for the client to stand up their AWS/Snowflake integration. This powerful stack met all of the requirements for the project and simplified the configuration process for us, ultimately speeding up the project for the client.

We defined this stack and stood up the Kubernetes cluster and auto-scaler in Terraform. Starting with the list of 40 manual commands they were previously running to stand up the integration, we cut that down to one “Terraform apply” command.

Step 2: Ongoing DevOps Support and Troubleshooting. Our team provided ongoing support involving troubleshooting any issues that arose during pipeline use and implementing solutions for the issues. With the agility provided by the DevOps framework, we were able to redeploy and solve quickly, not wasting any time for the client.

OUTCOME

Without a real-time scalable pipeline, virtual machines that weren’t actively in use were still costing the organization money while they sat idle waiting for a job to come through. With Kubernetes, the pipelines are able to scale compute resources to meet demand when many developers are pushing code through simultaneously. Another benefit is the ability to quickly revert back to the minimum amount of nodes when activity is low. This saves the company money, resources, and thousands of dollars in AWS EC2 costs.