UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

FINAL PROJECT

Luong Thi Ngoc Diep - 2440056

CLOUD COMPUTING

Automation of Spark Deployment with Ansible and Terraform on GCP

Academic Year: 2024-2026

1. Overview

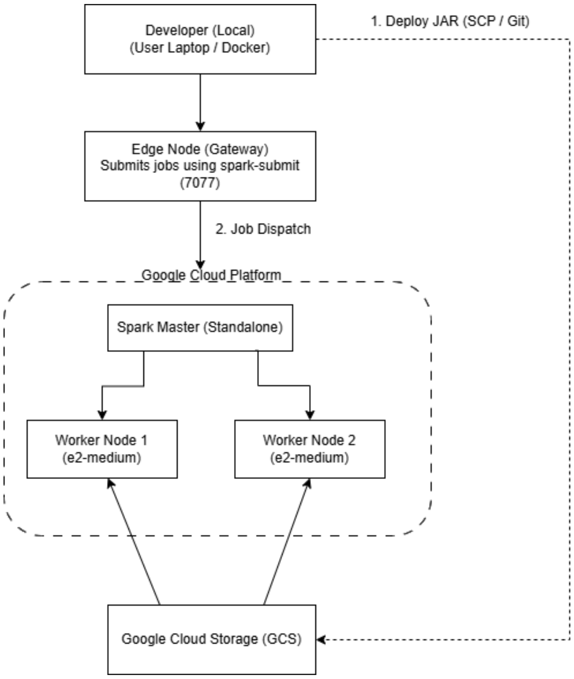

Figure 1.1: Overall architecture

This project focuses on automating the deployment of an Apache Spark cluster on Google Cloud Platform (GCP) using Terraform and Ansible. Terraform handles the provisioning of infrastructure, including virtual machines, networking, and security configurations. Ansible automates the installation and configuration of Spark across the cluster.
The cluster includes:

- Spark Master Node: orchestrates scheduling and resource allocation.
- Spark Worker Nodes: execute parallel tasks.
- Edge Node: submits jobs and manages cluster interactions.
- Optional storage nodes for enhanced data handling.

The deployment is validated by executing a WordCount application and measuring performance across varying numbers of executors.

2. Methodology

2.1 Infrastructure Provisioning

Terraform is used to define the cluster infrastructure as code. Resources include:

- VPC network and subnets for secure communication.
- Compute Engine instances for master, workers, and edge node.
- Firewall rules to allow SSH and Spark communication.
- IAM roles and service accounts to access Google Cloud Storage.

2.2 Cluster Configuration

Ansible automates node configuration:

- Installation of dependencies such as Java and system tools.
- Deployment of Spark binaries and configuration of environment variables.
- Automatic startup of Spark services on each node.
- Security enforcement via SSH key-only authentication and restricted access.

2.3 Validation Procedure

The cluster is validated by running a WordCount job on a sample dataset stored in Google Cloud Storage (GCS). Metrics recorded include job completion time and resource utilization. Multiple executor configurations are tested to assess scaling behavior.

3. Results

The deployment successfully provisioned all nodes and configured Spark services. The WordCount application ran successfully on the cluster, demonstrating parallel processing and correct results.

3.1 Result and Current Status

At this stage, the full WordCount execution on the automated Spark cluster cannot be completed due to an external issue with Google Cloud Storage. During job submission, Spark received the following error:

```
403 Forbidden: The billing account for the owning project is disabled
```

This indicates that the original GCS bucket belongs to a project whose billing has been closed, causing all read operations to fail. The issue is infrastructure-related rather than a problem with the automation scripts orSpark configuration.
I have already created a new bucket under an active billing project and am in the process of re-uploading the dataset and reconfiguring the pipeline. The WordCount job will be rerun as soon as storage access is restored. Although the final output cannot be shown at this moment, the deployment and automation workflow are functioning, and the remaining task is to restore valid GCS access. Work is ongoing and will be completed promptly.

3.2 Conclusion

Although the final WordCount result cannot yet be produced due to the external GCS billing issue, the automation workflow itself functions correctly:

- Terraform provisions the cluster resources reproducibly.
- Ansible performs consistent configuration across all nodes.
- Spark deployment and job submission operate correctly up to the point of data access.
- Work is actively ongoing to re-upload data and rerun the job once valid GCS access is restored.