ILLINOIS TECH

Cloud Computing and Cloud Native Systems

ECE 573

Homework 5

Abhilash Kashyap Balasubramanyam

abalasubramanyam@hawk.iit.edu

A20566944

Fall 2024

Illinois Institute of Technology

Date: November 10, 2024

Table of Contents

List of Tables

Solutions

1 Context

We have gained a lot of knowledge of different kind of cloud services so it is time to investigate what cloud computing platforms can provide beyond computing and storage as well as virtual private servers. Indeed, many of them provide managed services so that developers can work more efficiently by outsourcing deployment and maintenance of mature services like databases and focusing on core application and business logics instead.

Choose a major cloud provider, ideally one from AWS, Azure, and GCP, and investigate one managed service among distributed database (like Cassandra), message queues (like Kafka), and container orchestration (like Kubernetes).

2 Questions

2.1 Question 1

Q: What cloud provider are you using and what is the name of the managed service?

The cloud service that's been chosen to simulate for this homework is the **Google Cloud Platform (GCP)** service and the managed service in discussion will be **Google Kubernetes Engine (GKE)**.

Google Cloud offers a fully managed Kubernetes solution called the GKE. It enables developers to use Kubernetes, an open-source container orchestration technology that was first created by Google, to deploy, manage, and scale containerized applications. With cutting-edge capabilities like autoscaling, multi-cluster support, and integrated security, GKE provides a production-ready environment for Kubernetes cluster operations. By automating processes like cluster provisioning, scaling, and updating, GKE abstracts away a large portion of the complexity involved in operating Kubernetes clusters.

2.2 Question 2

Q: What kind of service is provided? Are the features exactly the same as those from the one we have discussed in the lectures, or what is the major difference?

Containerized applications' lifespan is managed by GKE, a container orchestration service. It uses Kubernetes to automate containerized workload deployment, scaling, and management.

Feature	Traditional Kubernetes	Google Kubernetes Engine (GKE)
Control Plane Management	Requires the control plane to be manually set up and maintained. Patches and updates must be applied by users on a regular basis.	Completely under Google's control, including patching, monitoring, and automated upgrades.
Autoscaling	Although basic horizontal pod autoscaling is provided, cluster size management and node scaling need human involvement.	Supports cluster autoscaling, node auto provisioning, vertical pod autoscaling (VPA), and horizontal pod autoscaling (HPA).
Release Channels	Without any pre-established channels, organizations are responsible for managing their own release schedules and upgrade pathways.	Provides stable, frequent, and fast release channels that enable workload-driven upgrades.
Security Enhancements	Users are responsible for managing security, including any possible upgrades and integration gaps.	Provides automatic security upgrades and integrates with Google Cloud security technologies like Identity Access Management (IAM) and Virtual Private Cloud (VPC).
Multi-Cluster Support	Coordinating the management of several clusters can be difficult and calls for outside tools or scripts.	Integrated support for using Anthos and other tools to manage numerous clusters for smooth operations across environments or locations.

Table 1: Traditional Kubernetes vs Google Kubernetes Engine (GKE).

The table above illustrates how GKE streamlines several parts of Kubernetes cluster management by automating processes that, in a conventional arrangement, would need human interaction.

2.3 Question 3

Q: Is it possible to automatically scale out the service horizontally if demand rises?

The scalability of the service is explained as follows:

1. **Automatic Horizontal Scaling:** GKE's integrated Horizontal Pod Autoscaler (HPA) enables automatic horizontal scaling. Based on measures like CPU utilization, custom-defined parameters like memory use, or external signals (like load balancer traffic), the HPA modifies the number of executing pods. The user may establish target utilization thresholds and minimum and maximum pod numbers by configuring autoscaling rules directly through the Google Cloud Console or by using 'kubectl'.

2. **Cluster Autoscaling:** GKE offers Cluster Autoscaler, which automatically modifies the cluster's size by adding or deleting nodes in response to resource needs, in addition to HPA at the pod level. For instance, Cluster Autoscaler will supply more nodes from the node pool if the cluster is unable to schedule new pods due to resource constraints. On the other hand, Cluster Autoscaler will eliminate any underused nodes that are no longer required because of decreased demand to save money.

1.

3. **Node Auto Provisioning:** This functionality goes beyond simple autoscaling by generating additional node pools on the fly when the workload demands more than the current ones can handle. Even when designated node pools reach capacity, Node Auto Provisioning makes sure the cluster may grow without interruption.

The applications can expand both vertically at the infrastructure level and horizontally at the pod level without the need for human intervention thanks to GKE's four-way autoscaling capabilities.

2.4 Question 4

Q: What is the pricing model?

The edition (Autopilot or Standard), cluster administration costs, node price, and extra resources like persistent storage are some of the variables that affect the GKE

pricing model. A summary of the main elements of GKE pricing, obtained from the Google Cloud-GKE website[1] is provided below:

- **Cluster Management Fee**

1. **Standard Mode:** Regardless of cluster size or configuration, a flat price of $0.10 per hour is assessed. This charge covers all aspects of control plane administration, including patching, monitoring, and updates.

Example Calculation:

$$Cost = 0.10 \times 24 \times 30 = \$72 \; per \; cluster \; per \; month$$

Where, 0.10 is the fixed cost, 24 is the number of hours per day and 30 is the number of days per month.

2. **Autopilot Mode:** In Autopilot mode, customers are billed according to the resources used by their pods (vCPU and memory utilization) rather than a separate cluster administration cost.

- **Node Pricing (Standard Mode)**

The underlying Compute Engine instances (VMs) that act as worker nodes in the cluster are paid for by the users when using Standard mode. The price is determined by variables such as:

- Type of machine (e.g., e2-medium versus n1-standard-4).
- The region in which the nodes are installed.
- These nodes have extra resources like persistent storage or GPUs.

For example:

$$Cost \; per \; VM = vCPU \; price + Memory \; price + Storage \; price$$

- **Autopilot Mode Pricing**

Instead of controlling nodes manually, customers in Autopilot mode are invoiced according to the actual resource utilization of your pods. The expenses are determined using:

1. Hours of virtual CPU: Costs $0.00822 each hour.
2. Hours of memory: $0.001125 per gigabyte per hour.

Because the customer is invoiced based on real resource utilization at a granular level rather than having to worry about provisioning or managing nodes directly, this paradigm makes billing simpler.

[1] (staff, 2024)

- **Persistent Storage Costs**

The user will be billed according to the kind and size of storage utilized if their program needs persistent storage (such as an SSD or HDD). Compute resources and persistent drives are billed individually.

- **Free Tier**

With monthly credits, GCP provides a free tier that may be used with Autopilot or zonal clusters. The free tier consists of monthly credits of $74.40 that can be used to Autopilot or zonal clusters. Regional clusters and other resources, such as persistent disks, are not covered by this credit, nevertheless.

- **Multi-Cluster Ingress Pricing**

Pricing for multi-cluster configurations with Multi Cluster Ingress depends on how many backend pods are included in 'MultiClusterService' resources for each of the project's GKE clusters. About $3 per backend pod per month (730 hours) is the price, which is equivalent to about $0.0041096 per backend pod per hour.

Bibliography

staff. (2024, November 10). *Pricing | Google Kubernetes Engine (GKE) | Google Cloud*. Retrieved from Google Cloud: https://cloud.google.com/kubernetes-engine/pricing