### **GCP Cost Optimization Report**

#### **Overview**

This report focuses on optimizing the costs associated with **Google Cloud Composer2**, **Compute Engine** and **Storage Bucket** resources within the current GCP environment. The insights below are based on discussions with relevant teams and observations from ongoing load testing and resource usage.

### **1. Composer2 Cost Optimization**

#### **Current Status:**

* **Load Testing**: Ongoing load testing for Composer2 in the UAT environment started in mid-September and is expected to continue until the end of October. This involves running 150 to 500 sessions daily.
* **Production Plans**: The same load testing patterns are expected to continue when Composer2 moves to the production environment after October.

#### **Key Findings:**

* **Resource Consumption**: The resource provisioning currently used in Composer2 UAT is likely over-provisioned and may not be fully optimized for the production environment.
* **Machine Configuration**: Optimization opportunities exist at the machine configuration level, which will directly affect the efficiency and cost when moving Composer2 to production.

#### **Recommendations:**

* **Collaborate for Machine Configuration**: Partner with Liang to ensure Composer2 is provisioned with the right machine configuration based on actual workload requirements. This will minimize over-provisioning and ensure cost-efficiency.
* **Scale Efficiently**: As Composer2 transitions to production, review the expected load and adjust the scale of resources accordingly. Avoid reserving resources for peak loads unless absolutely necessary to further reduce costs.

### 

### 

### **2. Compute Engine Cost Optimization**

#### **Current Status:**

* **Idle VMs**: Several VMs have been identified as idle and no longer in use, representing unnecessary cost.
* **Underutilized VMs**: Some VMs are not handling high-load tasks and are running on over-provisioned configurations, increasing costs without justified usage.
* **GKE Cluster Nodes**: A majority of the identified VMs are GKE cluster nodes that require further investigation to understand their configuration and usage patterns.

#### **Key Findings:**

* **Idle Resources**: Unused VMs contribute to avoidable costs and should be decommissioned.
* **Over-Provisioned Resources**: Over-provisioning on low-load VMs creates inefficiency in the cloud spend.

#### **Recommendations:**

* **Remove Idle VMs**: Decommission the idle VMs to stop incurring costs for unused resources.
* **Downgrade Underutilized VMs**: For VMs handling low-load tasks, downgrade their configurations to match the current workload. This will reduce the overall compute costs.
* **GKE Cluster Review**: Investigate the configuration of GKE cluster nodes further. This will ensure the clusters are using appropriate VM types and sizes based on workload requirements, preventing overspending on compute resources.

### **3. Storage Bucket Cost Optimization**

#### **Current Status:**

* **Object Deletion**: There are 8 buckets in the UAT environment, and a Python script has been created to automate the object deletion process for cost savings.

#### **Python Script Details:**

1. **Keyword Detection**: The script identifies specific folders in GCP buckets based on provided keywords.
2. **Date Filtering**: It filters folders older than 7 days.
3. **Deletion Process**: Deletes the filtered folders.
4. **Logging**: Logs the deleted objects, including their creation time.

#### **Progress:**

* **Completed**: Object deletion process for 1 bucket.
* **Ongoing**: Object deletion for 2 more buckets, currently running on the GCPOPS VM.

#### **Recommendations:**

* **Continue Automation**: Complete the deletion process for the remaining buckets to achieve full cost savings.
* **Review Bucket Retention Policies**: Ensure appropriate lifecycle policies are in place to automatically delete or archive old objects, reducing manual intervention in the future.

### **Next Steps:**

1. **Composer2**:
   * Work with Liang to refine machine configuration for Composer2’s production environment.
   * Perform a cost analysis of different machine types and configurations to find an optimal setup for production.
2. **Compute Engine**:
   * Initiate the decommissioning process for idle VMs.
   * Evaluate the workload of underutilized VMs and adjust their configurations.
   * Perform a deeper analysis of the GKE cluster nodes to identify optimization opportunities.
3. **Storage Buckets**:
   * Complete the object deletion process for all UAT buckets and implement lifecycle policies for long-term management.

#### **Conclusion:**

By optimizing resource allocation, removing unused assets, and automating object management, substantial cost reductions can be achieved across the Composer2, Compute Engine, and Storage Bucket resources. This will help streamline cloud spend and ensure efficient resource usage in the GCP environment.