# GCP Batch Processing Infrastructure Overview

## Introduction: Why Serverless for Batch Workloads?

**Serverless** is a cloud‑native model where the provider transparently provisions, scales, and manages compute resources. You pay only for actual usage—no idle servers.  
- **Cost Efficiency**  
- **Pay‑per‑use**: Billed only for actual CPU, memory, and I/O time.  
- **No over‑provisioning**: Automatically scales down to zero when idle.  
- **Operational Simplicity**  
- **No server management**: GCP handles OS updates, patching, and autoscaling.  
- **Faster deployment**: Focus on code, not infrastructure.  
- **Scalable & Resilient**  
- **Automatic scaling** under variable batch loads.  
- **Built‑in retry** and high availability.

For batch workloads with unpredictable spikes, serverless ensures we never pay for idle capacity and can handle bursts without manual intervention.

Canadian Banking has several batch jobs that we need to migrate to cloud and GCP serverless infrastructure is the best fit

I have been working on Terraform development and GCP (personal account) to create a GCP serverless infrastructure as a PoC. And now, after several weeks of testing and engineering, it is ready for use in the bank.

## Introduction: Orchestrating Batch Jobs

The bank has relied on Autosys to ochestrate and manage batch workloads. The jobs are linked in complex ochestration and we cannot unlink the esisting jobs without re-engineering the current orchestration. In order to minimize changes to the way batch jobs are managed, we are actively working with Broadcom autosys team to integrate GCP Cloudbatch functionality with Autosys.

## End‑to‑End Workflow

1. **Shell Script** (run-batch.sh)
   * Invokes the Submitter Function via HTTP.
   * This will probably be used by Autosys (depending on how they implement)
2. **Submitter Cloud Function** (google\_cloudfunctions\_function.batch\_submitter\_http\_function)
   * Validates input, calls Cloud Batch API to launch a job.
3. **Cloud Batch** (google\_batch\_job.batch\_job\_definition)
   * Orchestrates container execution on GCP compute.
   * Emits lifecycle events (start, success/failure).
4. **Pub/Sub Topic** (google\_pubsub\_topic.batch\_status\_topic)
   * Receives job events from Cloud Batch.
5. **Eventarc Trigger** (google\_eventarc\_trigger.batch\_status\_to\_logger\_trigger)
   * Routes Pub/Sub messages to the Logger Function.
6. **Logger Cloud Function** (google\_cloudfunctions\_function.batch\_status\_logger\_function)
   * Logs state transitions for monitoring and audit.

## Main GCP Components & Billing

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Terraform Resource | What It Is | Billing Model |
| **Service Account** | module.iam\_core.google\_service\_account.app\_sa | Identity under which functions & Batch tasks run | Free |
| **Custom IAM Role** | module.iam\_core.google\_project\_iam\_custom\_role.bucket\_rwl\_role | Scoped GCS permissions (list, get, create, delete) | Free |
| **Secret Manager Secrets** | module.iam\_core.google\_secret\_manager\_secret.\* | Stores DB credentials, SSH/PGP keys | $0.06 per secret per month + $0.05 per 10K access operations |
| **Cloud Functions (Submitter & Logger)** | google\_cloudfunctions\_function.batch\_submitter\_http\_functiongoogle\_cloudfunctions\_function.batch\_status\_logger\_function | Event‑driven compute for HTTP & Pub/Sub triggers | $0.40/million invocations + $0.0000025 per GB‑sec execution + networking |
| **Cloud Batch** | google\_batch\_job.batch\_job\_definition | Managed batch scheduling & execution | Charged at underlying VM rates (vCPU, RAM, GPU) only when running |
| **Pub/Sub Topic** | google\_pubsub\_topic.batch\_status\_topic | Asynchronous, at‑least‑once message delivery | $0.40 per million messages + $0.27 per GB data volume |
| **Eventarc Trigger** | google\_eventarc\_trigger.batch\_status\_to\_logger\_trigger | Routes events from Pub/Sub → Cloud Function | $0.10 per million events delivered |
| **GCS Bucket (Function Code)** | google\_storage\_bucket.function\_source\_code\_bucket\_cb | Stores zipped source artifacts for deployment | $0.02 per GB‑month storage + network egress |

## Security Model (Rationale & Details)

* **Least Privilege**
  + **app-sa** only has roles needed by each workload.
  + **Custom Role** (bucket\_rwl\_role) limits GCS access to exactly what Functions require.
* **Separation of Duties**
  + **Ops SA** (used by Terraform) has roles/iam.serviceAccountUser on app-sa but **cannot** read secrets or submit jobs directly.
  + **GCP service agents** (Functions, Batch) impersonate app-sa under strict IAM bindings.
* **Invocation Controls**
  + **Submitter Function**:
    - IAM binding in main.tf:
    - resource "google\_cloudfunctions\_function\_iam\_member" "invoker" {  
       project = var.project\_id  
       region = var.region  
       cloud\_function = google\_cloudfunctions\_function.batch\_submitter\_http\_function.name  
       role = "roles/cloudfunctions.invoker"  
       member = "allUsers" # can be locked down  
      }
    - Can be scoped to specific IP ranges or VPC Egress.
  + **Logger Function**:
    - Only Eventarc service agent can invoke:
    - resource "google\_cloudfunctions\_function\_iam\_member" "eventarc\_invoker" {  
       cloud\_function = google\_cloudfunctions\_function.batch\_status\_logger\_function.name  
       role = "roles/cloudfunctions.invoker"  
       member = "serviceAccount:service-<proj-number>@gcp-sa-eventarc.iam.gserviceaccount.com"  
      }
* **Secrets Management**
  + Defined in modules/iam\_core/main.tf:
  + resource "google\_secret\_manager\_secret" "db\_password" { … }  
    resource "google\_secret\_manager\_secret\_version" "db\_password\_version" { … }
  + Only app-sa has roles/secretmanager.secretAccessor on these secrets.

## Detailed Component Breakdown

### 1. Service Account (module.iam\_core.google\_service\_account.app\_sa)

* **What It Is**: A non‑human identity for running functions & tasks.
* **Roles**:
  + roles/batch.jobsEditor (defined in modules/batch\_deployer/iam.tf)
  + roles/logging.logWriter, roles/monitoring.metricWriter
  + roles/secretmanager.secretAccessor on each secret (modules/iam\_core/secret\_iam.tf)
* **Billing**: No direct charge for using service accounts.

### 2. Custom IAM Role (module.iam\_core.google\_project\_iam\_custom\_role.bucket\_rwl\_role)

* **What It Is**: A scoped set of permissions just for GCS buckets.
* **Permissions**: storage.objects.get, create, delete, list.
* **Billing**: Free; permissions do not incur cost.

### 3. Secret Manager (module.iam\_core.google\_secret\_manager\_secret.\*)

* **What It Is**: Managed vault for secrets.
* **Usage**: Cloud Functions fetch DB credentials at runtime.
* **Billing**: $0.06/secret‐month + $0.05/10K access operations.

### 4. Submitter Cloud Function

* **Resource**: google\_cloudfunctions\_function.batch\_submitter\_http\_function
* **What It Is**:
  + HTTP endpoint that submits Batch jobs.
* **Billing**:
  + Invocations: $0.40/M
  + Execution: $0.0000025/GB‑s
  + Networking: Standard egress rates.
* **Invoke IAM**: roles/cloudfunctions.invoker on allUsers (see above).

### 5. Cloud Batch Job Definition

* **Resource**: google\_batch\_job.batch\_job\_definition
* **What It Is**:
  + Containerized workload specification (image, CPU, memory).
* **Billing**:
  + Billed at underlying Compute rates (vCPU $0.010/hr, RAM $0.0014/GB‑hr) only while running.

### 6. Pub/Sub Topic (google\_pubsub\_topic.batch\_status\_topic)

* **What It Is**: Messaging channel for job events.
* **Billing**: $0.40/M messages + $0.27/GB data.

### 7. Eventarc Trigger (google\_eventarc\_trigger.batch\_status\_to\_logger\_trigger)

* **What It Is**: Routes Pub/Sub events to a Cloud Function.
* **Billing**: $0.10/M events delivered.

### 8. Logger Cloud Function

* **Resource**: google\_cloudfunctions\_function.batch\_status\_logger\_function
* **What It Is**:
  + Logs job state transitions for audit & monitoring.
* **Billing**: Same as Submitter Function but no public invocation.

## Invocation Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Invocation Method | Invoker | Required Invoker Role |
| Submitter Cloud Function | HTTP POST | **User** (script) | roles/cloudfunctions.invoker |
| Logger Cloud Function | Eventarc | **System** | roles/cloudfunctions.invoker (Eventarc service account) |
| Terraform Apply | CLI / CI/CD | **Ops SA** |  |
|  |  |  | roles/iam.serviceAccountUser on app-saroles/resourcemanager.projectIamAdminroles/serviceusage.serviceUsageAdminroles/cloudfunctions.adminroles/batch.adminroles/pubsub.admin |