

# Readme file for Google Cloud Billing

## H1 Catalog app

## H2 Tables created

The program creates 3 different tables to store information for the Google Cloud Billing Catalog. Following is information about the tables:

### H3 1. Services table

First tables holds the basic information for each service. The information includes the *service id* (which uniquely identifies a google cloud service) and the *service name*.

service_id	service_name
String	String

### H3 2. SKUs table

Second table holds information about each SKU of a google cloud service. The information stored in this table is a *sku id* (which uniquely identifies a google cloud sku), *service id* for which this sku belongs too, *sku description* (which holds a little bit of information regarding what is included in this sku, for example the specification of the compute engine etc.), and the *effective time* (a time since this sku and its services are valid).

sku_id	service_id	description	effective_time
String	String	String	String

### H3 3. Rates table

Third and last table holds the actual information about rates of each sku. This table uses a composite key, which is composed of *sku id* and the *start\_usage\_amount* which, when combined, help uniquely identify a tier rate for a service. The table also includes *units* (main unit of a currency) and the *nanos* (which is equivalent to 1/10000000000 of the main unit of currency). It also includes the *currency code* and the

*formatted\_price* which combines the *units* and *nanos* to display the final price of a service unit.

sku_id	start_usage_amount	units	nanos	currency_code	formatted_price	updated_on
<i>String</i>	<i>Number</i>	<i>Number</i>	<i>Number</i>	<i>String</i>	<i>Decimal</i>	<i>Date</i>

## H2 Charting price for a service

To chart price over time for a service, following steps will be taken:

1. Get the service id for the service from the services table
2. Get all the skus for the service from the skus table
3. For each sku, get all the rates from the rates table, each sku will have multiple tiered rates, depending on the start usage amount. Search can be further narrowed down by specifying the start usage amount too. The date at which the rate was applicable can be deduced from the `updated_on` variable.

## H2 Configurations needed for running

Following are the pre-requisites for running the app:

- Serverless framework (can be installed through npm)
- AWS Credentials configured on the system (easily done through `awscli: aws config`)
- A google cloud API key

In addition to that, you can configure following settings in `serverless.yml` file before deploying app:

- `ServicesTableName`: Name of the table to be created on DynamoDB for services' information
- `SKUsTableName`: Name of the table to be created on DynamoDB for skus' information
- `RatesTableName`: Name of the table to be created on DynamoDB for rates' information
- `S3BucketName`: Name of the bucket to be created to store the raw data for service prices in JSON format
- `GoogleCloudAPIKey`: API Key for the google account, required to call the google billing catalog api

To connect to the serverless.com dashboard, either configure the serverless credentials in the cli tool, or uncomment and set the `app` and `org` values in the `serverless.yml` file.

## H2 Services covered by the app

By default, the app is only monitoring the storage and compute services as mentioned on the google cloud page for respective categories. If more services are to be monitored, a service from following list can be added to existing *storage\_services/compute\_services* list or the *other\_services* list in the handler.py file (list is empty by default). The list of services which can be accessible through the API and can be monitored are:

- Qubole Qubole Data Service
- Web Risk
- Geolocation API
- Cloud Text-to-Speech API
- Cloud Natural Language API
- Prediction
- Mavata Technologies mCart Platform-as-a-Service
- Test Marketplace Procurable OP Full Solution
- Geocoding API
- Cloud Data Fusion
- Bkper
- Maps Elevation API
- StraaS
- Cloud Run
- BigQuery Reservation API
- Cloud Composer
- Shape Security Connect
- Trillo Inc. Trillo Platform Runtime
- NetApp Cloud Volumes
- Redis Labs
- Firebase Realtime Database
- Robin Storage
- Translate
- Cloud Scheduler
- Fivetran Fivetran Data Pipelines
- Kentik Detect
- Transcode API
- NetApp, Inc. Cloud Manager
- Google Everest on BareMetal
- Human API Human API
- SendGrid

- BigQuery
- Firebase Hosting
- Cloud CDN
- Genomics
- Cloud Functions
- k8sBot
- Bell Integrator
- Alcide
- Roads API
- Prodoscore
- Z-Stream
- Aqua Container Security Platform
- Kasten K10 Platform
- Firebase Auth
- Zync
- Confluent Cloud for Apache Kafka
- Elastifile Cloud File System
- Cohesity Backup as a Service
- JFrog Cloud
- Google Maps Mobile SDK
- Portworx
- Stackdriver Logging
- Cloud Dataflow
- Stackdriver Monitoring
- Cloud Memorystore for Redis
- HYCU, Inc HYCU
- Firebase
- GliaStudio
- iKala interactive media inc. StraaS Cloud Media Engine
- Cloud Speech API
- BigQuery BI Engine
- Appranix Site Reliability Automation
- DKube
- zData Cloud Enablement Services
- Cloud IoT Core
- Compute Engine
- Cloud Data Labeling Service
- PingCAP TiDB Operator
- IBM Power Systems for Google Cloud
- Cloud Healthcare

- Cloud Domains
- Firebase Test Lab
- Stackdriver
- CloudHedge Service
- Tackle Upstream
- Cloud Build
- Cloud Video Intelligence API
- Metadot Mojo Helpdesk
- Spotinst
- BigQuery Data Transfer Service
- Cloud Storage
- Cloud SQL
- Twinword Inc. Category Recommendation API
- Confluent, Inc. confluent-cloud-for-apache-kafka-poc
- Stackdriver Trace
- PerimeterX PerimeterX Bot Defender
- Cloud Pub/Sub
- Rollbar
- Global Tax as a Service
- MongoDB Atlas Professional
- MongoDB Atlas GCP Free
- Codefresh
- USAN Dialogflow Enterprise Telephony Gateway
- Product-Market Intelligence - Inventurist AI Lite
- Palo Alto Networks RedLock
- Elasticsearch
- Directions API
- Kafkaesque Technologies Inc. Kafkaesque
- Aerospike Server Enterprise
- Vogsy Service
- Cloud Vision API
- omniX labs omniX View | Analyze video in real time
- Time Zone API
- Cloud Bigtable
- Google Service Control
- Source Repository
- Cloud Spanner
- Kubernetes Engine
- Buzzboard SMB Streams
- Maps Static API

- Cloud Data Loss Prevention
- Galactic Fog Gestalt
- Cloud Talent Solution
- Cloud Filestore
- Identity Platform
- BigQuery Storage API
- Remote Build Execution
- Cloud TPU
- Cloud Machine Learning Engine
- Distance Matrix API
- Harness, Inc. Harness
- Weaveworks
- Cloud Contact Center
- ManagedKube Kubernetes Cost Attribution
- HeadSpin HeadSpin Mobile Performance
- Red Pill Analytics
- Cloud AutoML
- Cloud Key Management Service (KMS)
- App Engine
- superQuery
- Cloud Tasks
- MongoDB Atlas Starter
- Cloud DNS
- Cloud Dialogflow API
- Security Command Center
- NS8 NS8 Protect
- Komprise
- itopia
- Actifio Actifio GO GCP