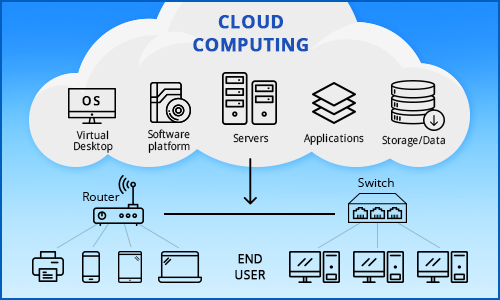
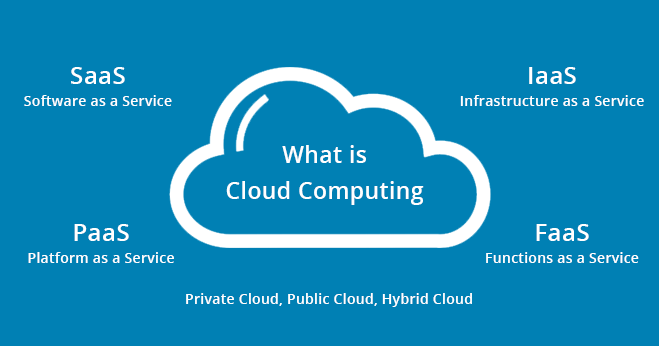
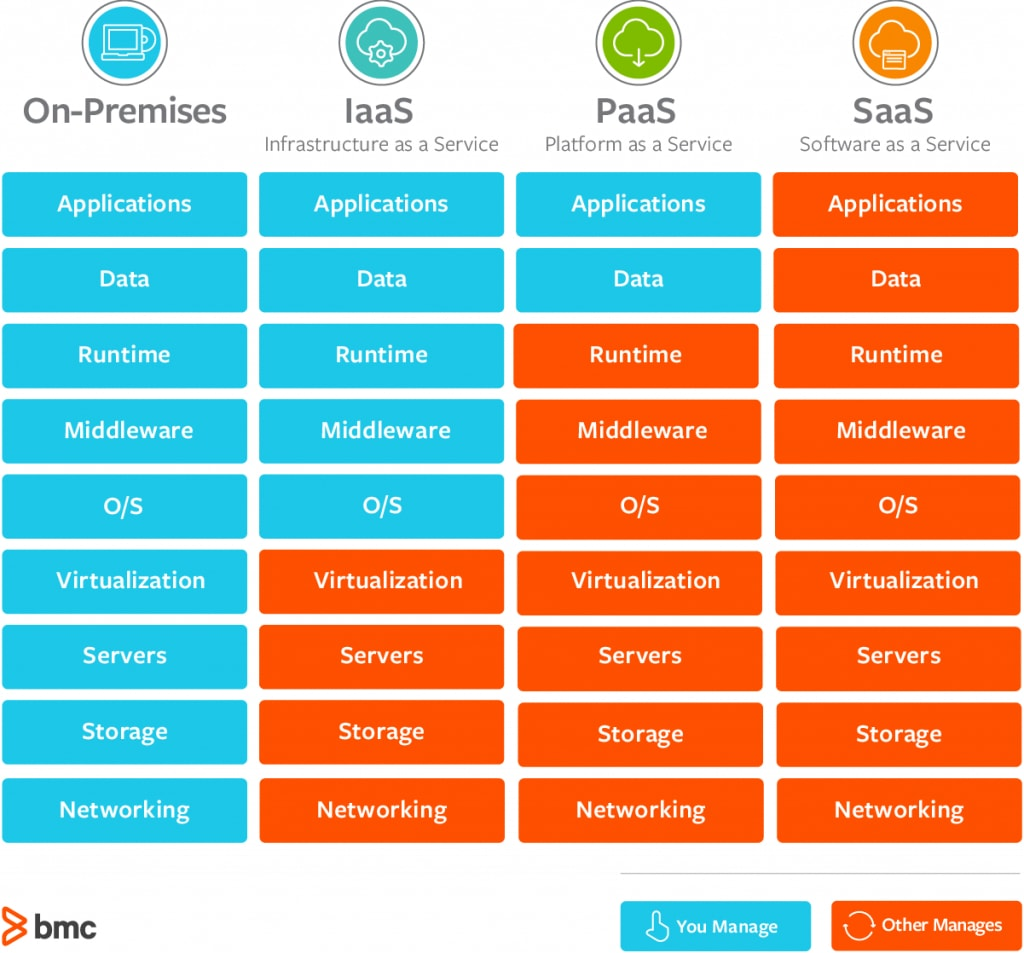
**Cloud Computing:**

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing



Cloud Computing and Its Services:





**Cloud Benefits:**

* Reducing IT costs
* Increasing productivity
* Strengthening cybersecurity
* Enabling flexible/remote working
* Speeds up processes
* Improves the customer experience

**IaaS:** [Infrastructure as a Service](https://azure.microsoft.com/en-gb/overview/what-is-iaas/) (IaaS) allows businesses to put all their IT infrastructure in the cloud. The term ‘infrastructure’ includes everything that powers your IT and makes it run, including storage, hosting, network, and more. Before the cloud, all of this would have been run on a physical server in the office and have required IT staff to look after it. Your IT staff no longer must run, maintain and manage this, meaning they can work on tasks on much higher value. This can work out at large cost savings. IaaS is also more secure and reliable, as you have seasoned experts taking control of your infrastructure

**PaaS:** [Platform as a Service](https://azure.microsoft.com/en-gb/overview/what-is-paas/) (PaaS) gives businesses the tools, infrastructure, and support to build and run apps in the cloud. In 2018, there are [3.8 million](https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/) apps available on Google Play and 2 million apps in Apple’s App store. Mobile apps play such a huge role in how customers and companies interact. Many customers expect companies to have mobile apps – and it can be a key competitive differentiator.

**SaaS:** [Software as a Service](https://azure.microsoft.com/en-gb/overview/what-is-saas/) (SaaS) is software accessed online. In the past, businesses would buy software on disks and install them onto their employees’ computers, with SaaS software is hosted in the cloud and accessed online. Office 365 and Salesforce are good examples of SaaS. If you want to access your email or your Salesforce account, you log in online and can access this anywhere, anytime, on any device.

## Common examples of SaaS, PaaS, & IaaS

| **Platform Type** | **Common Example** |
| --- | --- |
| **SaaS** | Google Workspace,GoToMeeting |
| **PaaS** | AWS Elastic Beanstalk, Google App Engine, OpenShift |
| **IaaS** | Google Compute Engine (GCE) |

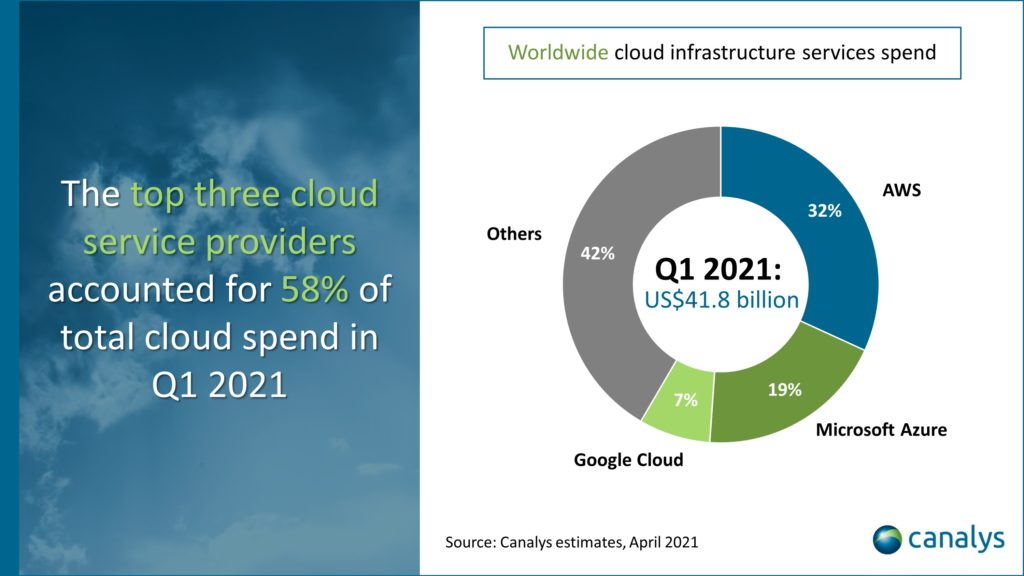
## SaaS vs PaaS vs IaaS:

Each cloud model offers specific features and functionalities, and it is crucial for your organization to understand the differences. Whether you need cloud-based software for storage options, a smooth platform that allows you to create customized applications, or complete control over your entire infrastructure without having to physically maintain it, there is a cloud service for you.

No matter which option you choose, migrating to the cloud is the future of business and technology.

Cloud Computing Platforms:

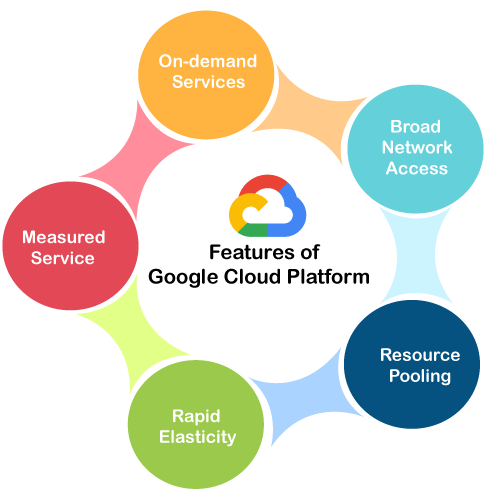
AWS vs Azure vs Google Cloud Market Share 2021:



GCP Cloud :

Google Cloud Platform is a set of Computing, Networking, Storage, Big Data, Machine Learning and Management services provided by Google that runs on the same Cloud infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, Google Photos and YouTube.

Features of GCP Cloud:



**Best Pricing:**

Google Cloud hosting plans are cheaper than other platforms’ hosting plans.

Google Cloud offers to its customers the pay-as-you-go feature where the users only have to pay for the resources they use.

Now Google Cloud Platform is a clear winner when it comes to the cost of Services.

As you can see in the image below a 2 CPU 8 GB RAM instance for GCP is priced **at $50 per month**

whereas AWS instance is priced **at $69 per month**. You save 25% on the same instance per month.

You can also save more as the Billing for AWS is done **on Per-Hour basis**, whereas

Google Cloud Platform provides billing on a **Per-Second basis**

**Regions and Zones Comparison:**

**AWS:** Total of 18 Regions, with more than 3 zones per Region

**GCP:** Total of 15 Regions, with more than 2 zones per Region

* **On-demand services**: Automated environment with web-based tools. Therefore, no human intervention is required to access the resources.
* **Broad network access**: The resources and the information can be accessed from anywhere.
* **Resource pooling**: On-demand availability of a shared pool of computing resources to the users.
* **Rapid elasticity**: The availability of more resources whenever required.
* **Measured service**: Easy-to-pay feature enables users to pay only for consumed services.

**Pay-as-you-go:**

Google Cloud offers the ‘use now, pay later’ policy.

Users have to pay only for the services they use.

**No Termination Fee:** The moment you stop using the services, you stop paying for it.

## Which is better GCP or AWS or Azure?

## Pricing model– the presence of customer-friendly pricing models and discounts makes **GCP** the clear winner.

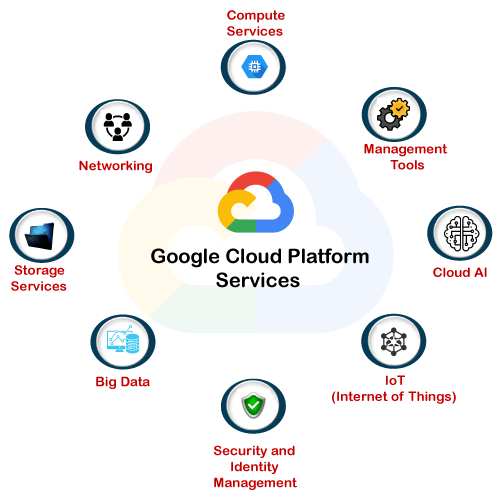
## Services– **AWS** leads the way with the volumes of services provided but with respect to the integration with open-source and on-site platforms **Azure** leads the pack

## 

## **GCP Services:**

Google offers a wide range of Services. Following are the major Google Cloud Services:

* Compute
* Networking
* Storage and Databases
* Big Data
* Machine Learning
* Identity & Security
* Management and Developer Tools



Compute: GCP provides a scalable range of computing options you can tailor to match your needs. It provides highly customizable virtual machines. and the option to deploy your code directly or via containers.

* Google Compute Engine
* Google App Engine
* Google Kubernetes Engine
* Google Cloud Container Registry
* Cloud Functions

Networking: The Storage domain includes services related to networking, it includes the following services

* Google Virtual Private Cloud (VPC)
* Google Cloud Load Balancing
* Content Delivery Network
* Google Cloud Interconnect
* Google Cloud DNS

Storage and Databases: The Storage domain includes services related to data storage, it includes the following services

* Google Cloud Storage
* Cloud SQL
* Cloud Bigtable
* Google Cloud Datastore
* Persistent Disk

Big Data: The Storage domain includes services related to big data, it includes the following services

* Google BigQuery
* Google Cloud Dataproc
* Google Cloud Datalab
* Google Cloud Pub/Sub

Cloud AI: The Storage domain includes services related to machine learning, it includes the following services

* Cloud Machine Learning
* Vision API
* Speech API
* Natural Language API
* Translation API
* Jobs API

Identity & Security: The Storage domain includes services related to security, it includes the following services

* Cloud Resource Manager
* Cloud IAM
* Cloud Security Scanner
* Cloud Platform Security

Management Tools: The Storage domain includes services related to monitoring and management, it includes the following services

* Stackdriver
* Monitoring
* Logging
* Error Reporting
* Trace
* Cloud Console

Developer Tools: The Storage domain includes services related to development, it includes the following services

* + Cloud SDK
  + Deployment Manager
  + Cloud Source Repositories
  + Cloud Test Lab

## **Creating a Free Tier Account on GCP:**

To start using Google Cloud Platform, we are first required to create an account GCP. Here, we will create a free tier account for explaining the upcoming topic of this tutorial. The best thing about free account is that Google provides $300 worth credit to spend over the next 90 days after the date of account creation. Google offers all the core services of GCP with a free account for the next 90 days.

However, users must have a credit card to start a free tier account. Google asks for the credit card details to make sure that it is a genuine human request. Google does not charge automatically even after the 90 days or when we have exhausted the $300 free credit. The amount will only be charged when we will be upgrading our free account to a paid account manually.

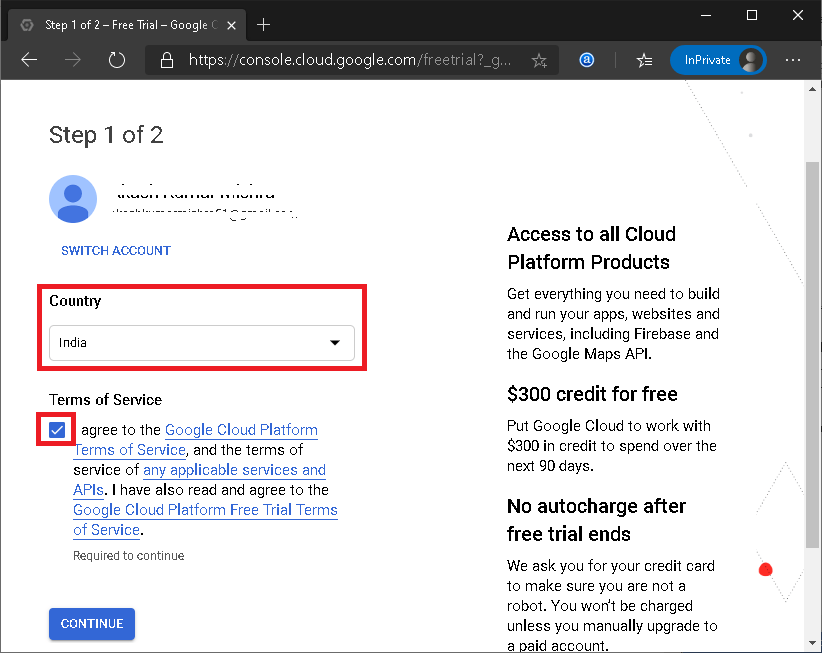
Let's start with the steps of creating a free tier account on Google Cloud Platform:

**Step 1**: First, we are required to navigate to the following link: <https://cloud.google.com/gcp/>

**Step 2**: On the next screen, we need to click on 'Get started for free'

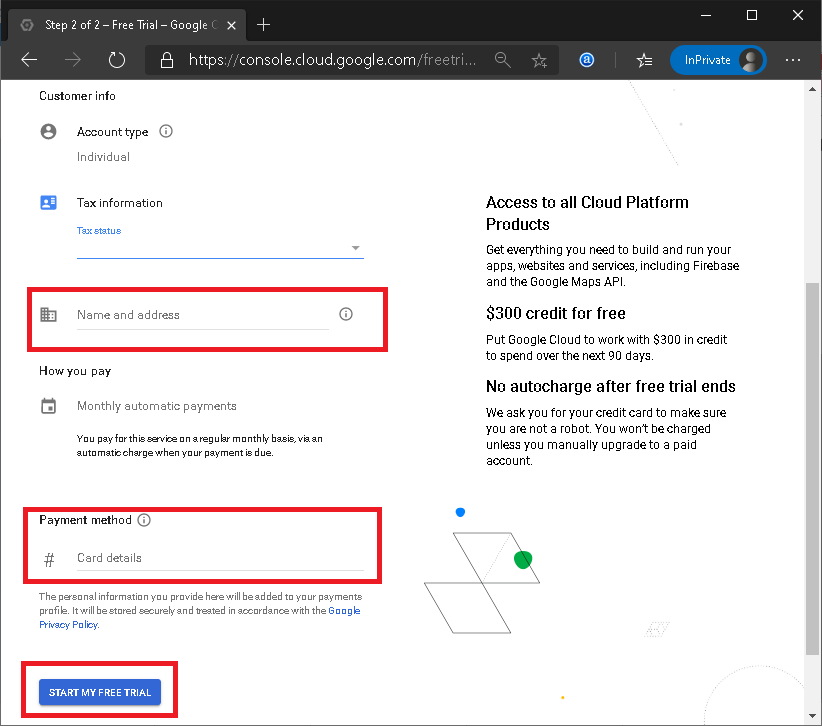
**Step 3**: Next, we are required to login to the Google Account. We can use the 'create an account' button if we don't have an existing Google account.

**Step 4**: Once we have logged in, we will get to the following screen:

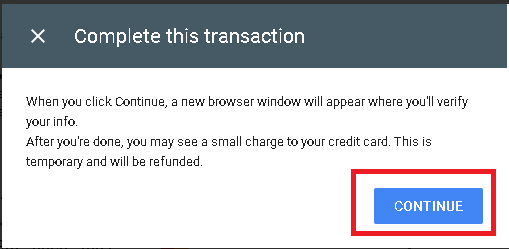


Here, we must select the Country, agree to the Terms of Service, and then click on the 'CONTINUE' button.

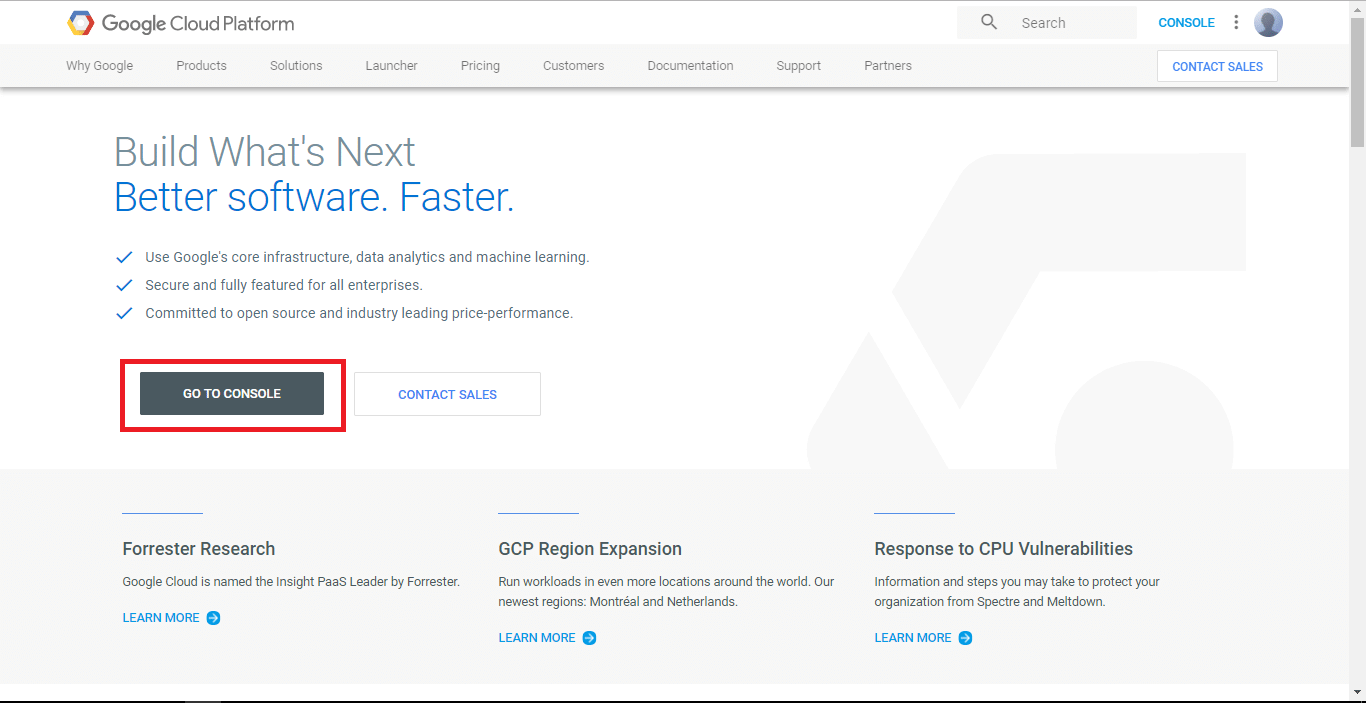
**Step 5**: On the next screen, we have to enter some necessary details such as name and address details. Also, we have to enter payment details like the method of payments and credit card details. After filling all the details, we need to click on the button 'START MY FREE TRIAL' from the bottom of the page:



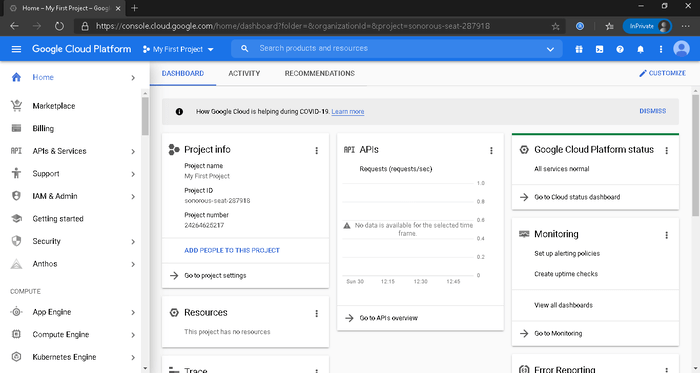
**Step 6**: Google asks for the confirmation to use the credit card for the small deduction to ensure that the card information is correct. However, the amount is refunded back to the same account. Here, we need to click on the 'CONTINUE' button:



**Step 7**: On the next screen, we must click on the 'GO TO CONSOLE' button:



After clicking on the 'GO TO CONSOLE' button, we will be redirected to the Dashboard that includes a summary of GCP services along with projects and other insights. It looks like this:



To be specific, a Dashboards of GCP displays the summarized view of the followings:

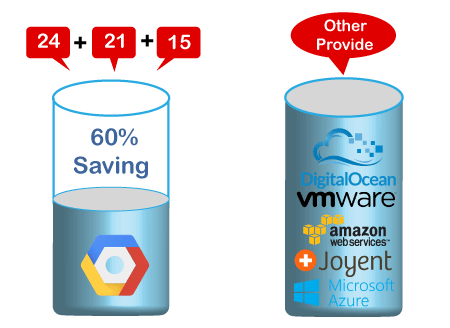
* **Project Info**: contains project details such as project name, ID, and number.
* **Resources**: contains a list of resources being used in the related project.
* **APIs**: contains various API requests running with the project (in request/sec form).
* **Google Cloud Platform Status**: displays an overall summary of services that are part of GCP.
* **Monitoring**: displays alerts, performance stats, Uptime, etc. to ensure that systems are running reliably.
* **Error Reporting**: displays errors occurring in the projects, but it needs to be configured first.
* **Trace**: displays latency data of existing applications across a distributed tracing system.
* **Compute Engine**: displays the insights of CPU usage in percentage (%).
* **Tutorials**: contains Getting Started guides (basic guides) to explain how the GCP features work.
* **News**: displays news and other important updates regarding Google Cloud Platform.
* **Documentation**: contains in-depth guides to teach more about Compute Engine, Cloud Storage, and App Engine.

## **Google Cloud Platform Pricing**

When it comes to pricing, Google Cloud Platform is the cheapest solution in the market. GCP is not only low on price but also offers more features and services than other providers.

When comparing GCP with other leading competitors, it has more benefits over them. Google provides its users a massive 60% savings, including:

* 15% rightsizing recommendation
* 21% list price differences
* 24% of sustained usage discounts



Some of the main benefits of GCP pricing are:

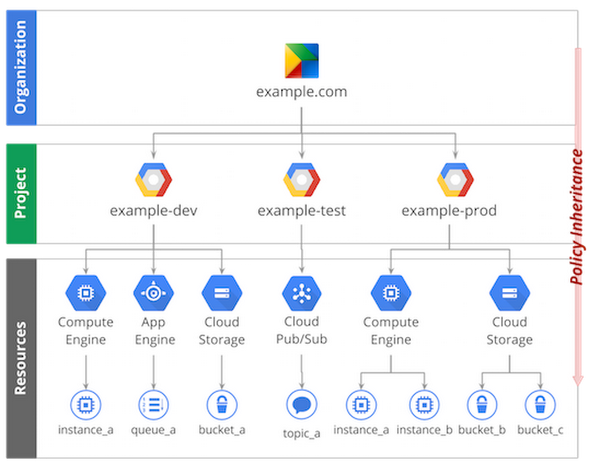
**No Hidden Charges:** There are no hidden charges behind the GCP pricing. Google's pricing structure is straightforward and can be easily understood.

**Pay-as-you-go:** Google offers its customer 'use now, pay later' option. So, users will have to pay only for those services which they want to use or already using.

**No Termination Fee:** Users are free to stop using Google services whenever they want, and there will not have to pay any termination fee. That means the moment users stop using Google services; they stop paying for it.

**GCP Organization Flow diagram:**

The following diagram shows an example of various resources and their hierarchical organization in GCP.

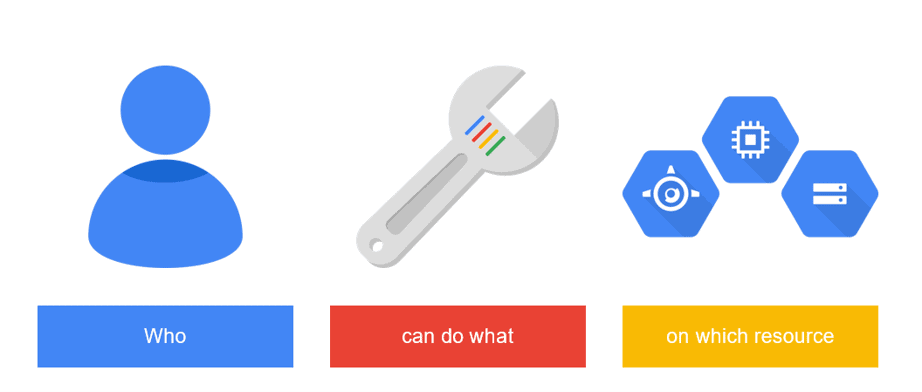


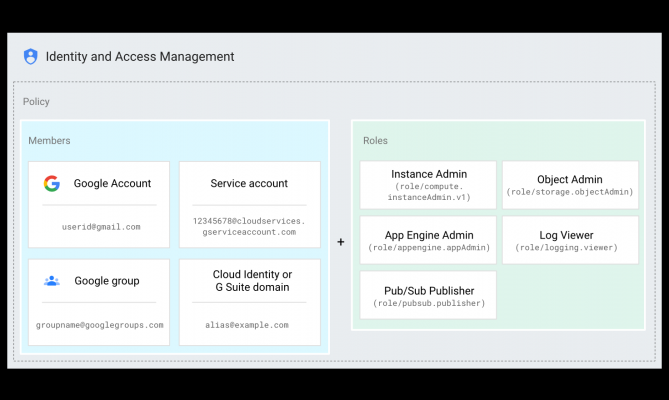
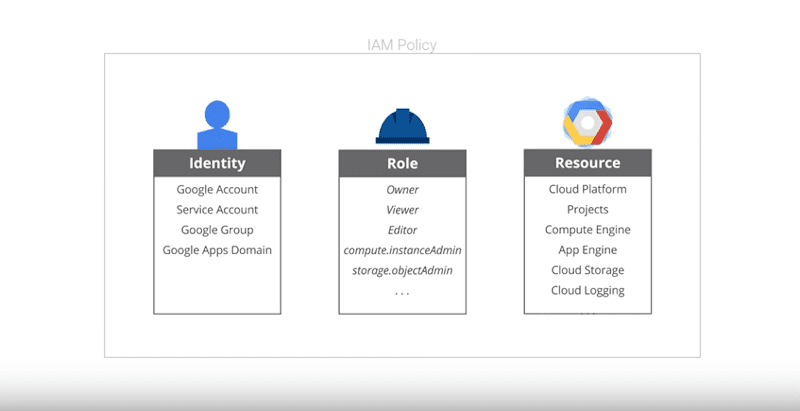
Now we will discuss one of the GCP Service i.e IAM

**IAM: ( Identity and Access Management):**

* Can manage access control by defining
  + who (identity)
  + has what access (role)
  + for which resource.
* permission to access a resource isn’t granted directly to the end user.
* permissions are grouped into roles
* roles are granted to authenticated members.
* IAM policy defines and enforces what roles are granted to which members

**Google Introduces Identity and Access Management for Cloud Platform**

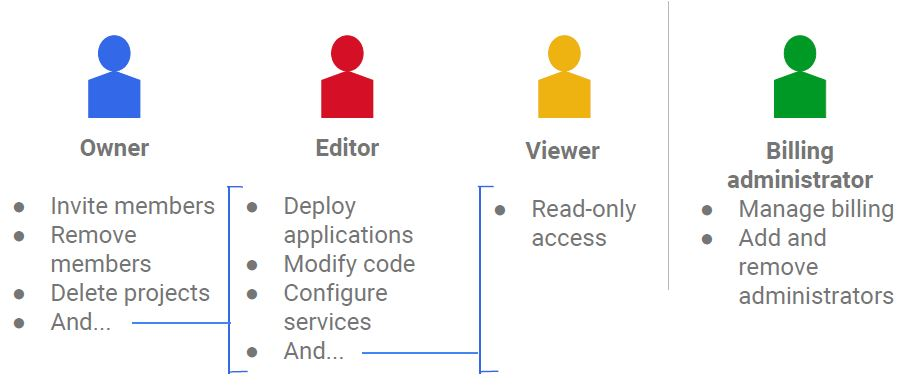




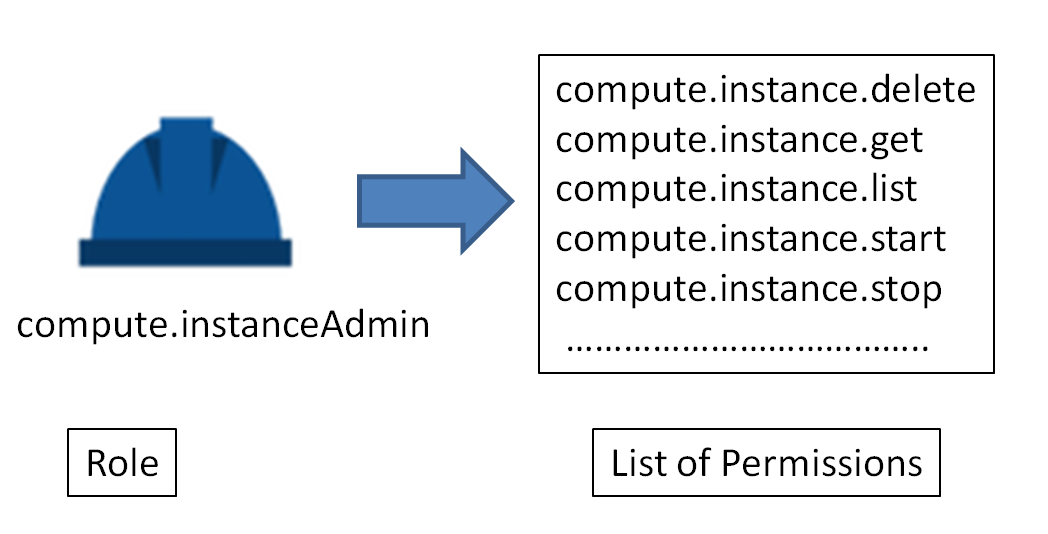
Identity:

* + can be a Google Account (for end users),
  + service account (for apps and virtual machines)
  + Google group
  + G Suite or Cloud Identity domain that can access a resource.
* Role:
  + A collection of permissions.
  + Permissions determine what operations are allowed on a resource.
  + With role allocation, all permissions with the role are granted

GCP Console Basic Access Roles:



Example for Role:



An Identity and Access Management (**IAM**) **policy**, which specifies access controls for Google Cloud resources. A **Policy** is a collection of bindings . A binding binds one or more members to a single **role** . Members can be user accounts, service accounts, Google groups, and domains.

### 

### **Google Account : (Example your flipkart email ID- kiran.kumar@flipkart.com)**

## A Google Account represents a developer, an administrator, or any other person who interacts with Google Cloud. Any email address that's associated with a Google Account can be an identity, including gmail.com or other domains.

### **Service account:**

## A service account is an account for an application instead of an individual end user. When you run code that's hosted on Google Cloud, the code runs as the account you specify. You can create as many service accounts as needed to represent the different logical components of your application.

use the following command to list the service-accounts in the current project.

gcloud iam service-accounts list

## Sample Service Account key file content :

{

"type": "service\_account",

"project\_id": "devops-test-316311",

"private\_key\_id": "955cae7d251131676c0363c1e0dabae9b8431f78",

"private\_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDuiekUuMkNUUdE\ndUO0kes+5eiDGNVpyoD/ZZvILu7exxGjNBfJ8RvwjbxvXcd5drEom3z5kT2Xy5Nu\nqlb+DsVrvNuhmwW6vIN3opw/KCmJR4QBPFsidh6pJCju/w6OTo/dISp0mZ421BJF\ngihyMICkKd5jdldFCagxijcJqBT7HF9wfciExT2XYzVm3Kp2hJCBXNWqj8NeJ7BJ\nWEJVBt/S9inSzR5x6jt7f2zFLT3WAogOFRL/no5SzlSDTMRMFILgOQREqf9gDD4I\nD/rjuJ6aMD+5qh48pk5BaztpSCrYc3LKXArsvxKXwKne91M2ZLzXgd6Q2s4RGFjs\na4rM1PFLAgMBAAECggEAE3xFg3FSzxkwGqn/lV0EG/e45hiduk4V+obo0lHvJliZ\nmrzTIqZk82IIngpJlKT69Me6zkIsh11UVkpo38kCHHppcIqSzb2/eYLZg5k6x2GF\nqXouMaL3fzaad2dzX3GQc+5VIwA3eeAZ+/6VYjqPD5nx9isBFv0idaWGOG44BIkU\n65VRt1gFAEi7YAXFhchzgL6d3MR1O1zMp0ZREijtnzh/i6X/3IY0n1Hb/GtSNKyK\n+sL2pw62iggguOZPFWVhbD4Umov+KHWMCJlYgyHqiArGN8OCq6EDQ0xOd6iQRvKR\nhEpvCnyk4mnn727SBKHlFVLzeQvG8zx8AB3mIRkn5QKBgQD7vPcVx49g3E6aB//c\nIQBoGBmQNkrPV+AOSX/QwdQ5BanMCUIMvUdzdURuoUVD2Hx0AdZbPsoCKrZXmfXi\nhFt7kEhaXSO783YfZQigg3d4tonlVTGodCOIyjR5wXAyxMW278jfdPMPvhENxO2y\nGR3ERl2KASK/Z9wPrugXgb23dQKBgQDyk70mYcryKcVQqjHTizHMr0OsMwE3pkHk\nzJvQ+bLFBDyfAXEUNwkbfdO8iv1YgVNPl5Gj7mJsIUV/6SunepF3g5vKwAVtvxau\nkMqeCorB/zoT+hX6yR+r1/kCW067J5pMCQ63/QlGDrQcq9mY6mbbJc9jpgPBVjl6\nM8l+YzOtvwKBgQDjTfUtTUxBgavXmXGGKqZbV6d9eAiEwCwntyPbC9pMvUY2NX7p\nRDTxxuMrLWdJ8DM2w7z1SHxQsqZjcHXQYq1mFMNRYUUm33jtEFmxIA1WIQKBgQDW\nJolpFhJ1gVzZnrcIO61GN0A2CH09pMAtfljcv2QMPHZMJZivsPqQU6AWplmMxfdH\nn9aKc4xeS20wxCeu3i0H5p8l+imQvo8Go3rM1R+0rl6WfkTjNqWILs19QCCcZxT3\npVNiKLgEKjzlxuz964nMVG/nWAuepkFt2cQ7mNZpvwKBgAfqzjbc4s/ntI/PHARj\nMRIcAHM08hCB1gJx2vzkhfYMFlnmdiveBKdKx6NT07+tquf7XC9+qQ3ZEx5Rp+xq\nO2FR5J5KsuEhjMHUNH2lAYieFgAErB1+A8s8BxJ8BuAfDwHSpLJFBbkqWKo+ZUZY\ngXxTJZum/pO2ClpXzTyM0hOX\n-----END PRIVATE KEY-----\n",

"client\_email": "terraform-infra@devops-test-316311.iam.gserviceaccount.com",

"client\_id": "113979197753561688262",

"auth\_uri": "https://accounts.google.com/o/oauth2/auth",

"token\_uri": "https://oauth2.googleapis.com/token",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/terraform-infra%40devops-test-316311.iam.gserviceaccount.com"

}

## **How to activate Service Account in GCP :**

Gcloud auth list -> it will give auth list (will give user accounts and service accounts details)

kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gcloud auth activate-service-account terraform-infra@devops-test-316311.iam.gserviceaccount.com --key-file=/home/kirankumarrv0990/terraform.json --project=devops-test-316311

Activated service account credentials for: [[terraform-infra@devops-test-316311.iam.gserviceaccount.com](mailto:terraform-infra@devops-test-316311.iam.gserviceaccount.com)]

kirankumarrv0990@cloudshell:~ (devops-test)$ gcloud auth list

Credentialed Accounts

ACTIVE ACCOUNT

kirankumarrv0990@gmail.com

**\* terraform-infra@devops-test-316311.iam.gserviceaccount.com**

To set the active account, run:

$ gcloud config set account `ACCOUNT`

kirankumarrv0990@cloudshell:~ (devops-test)$ gcloud config set account terraform-infra@devops-test-316311.iam.gserviceaccount.com

Updated property [core/account].

kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gcloud compute addresses list

**ERROR: (gcloud.compute.addresses.list) Some requests did not succeed: - Required 'compute.addresses.list' permission for 'projects/devops-test-316311'**

**Note:**

[**terraform-infra@devops-test-316311.iam.gserviceaccount.com**](mailto:terraform-infra@devops-test-316311.iam.gserviceaccount.com) **service account does not have access to get the compute IP address list and we can observe the same from above .**

Now i can try the same command with my account (mean my google account)

**Gcloud auth login ->u can try to click that URL and copy that code and paste it here.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gcloud auth login**

**Go to the following link in your browser:**

**https://accounts.google.com/o/oauth2/auth?response\_type=code&client\_id=32555940559.apps.googleusercontent.com&redirect\_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=Tpzoln4r0Ve7zGFwt8I2s4OT5IGVxp&prompt=consent&access\_type=offline&code\_challenge=tvt-Cd9ouT1x9hwXMwKj686T0fbiGGBCO\_F8c96J1iE&code\_challenge\_method=S256**

**Enter verification code: 4/1AX4XfWhERVhBxHcHFH9hE74huGDoc0sdL17TwnZeqZkEDFck1F1ReLplAMg**

**You are now logged in as [**[**kirankumarrv0990@gmail.com**](mailto:kirankumarrv0990@gmail.com)**].**

**Your current project is [devops-test-316311]. You can change this setting by running:**

**$ gcloud config set project PROJECT\_ID**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gcloud auth list**

**Credentialed Accounts**

**ACTIVE ACCOUNT**

**\* kirankumarrv0990@gmail.com**

**terraform-infra@devops-test-316311.iam.gserviceaccount.com**

**To set the active account, run:**

**$ gcloud config set account `ACCOUNT`**

**Now try to get the Compute Address list with below command.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gcloud compute addresses list Listed 0 items.**

**I can use the same account to stop one running GCE VM as below.**

**gcloud compute instances stop centos-jenkins --zone=us-central1-a**

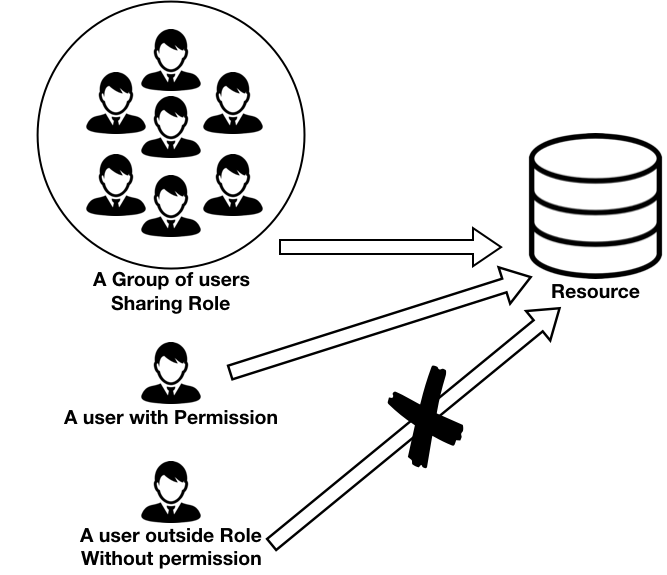
**Stopping instance(s) centos-jenkins...done.**

**Updated [https://compute.googleapis.com/compute/v1/projects/devops-test-316311/zones/us-central1-a/instances/centos-jenkins].**

**Google Group: (Example your flipkart devops team email ID- devops@flipkart.com)**

## A Google group is a named collection of Google Accounts and service accounts. Every Google group has a unique email address that's associated with the group. You can find the email address that's associated with a Google group by clicking **About** on the homepage of any Google group.

Google Groups are a convenient way to apply an access policy to a collection of users. You can grant and change access controls for a whole group at once instead of granting or changing access controls one at a time for individual users or service accounts. You can also easily add members to and remove members from a Google group instead of updating an IAM policy to add or remove users.



## Features of IAM:

## Single access control interface

#### Automated access control recommendations

#### Flexible roles

#### Free of charge

Interview Questions:

How does GCP IAM work?

How **IAM works**. With **IAM**, you manage access control by defining who (identity) has what access (role) for which resource. For example, Compute Engine virtual machine instances, Google Kubernetes Engine (GKE) clusters, and Cloud Storage buckets **are** all Google Cloud resources

How long does it take for IAM changes to take effect?

In general, when you update an **IAM** policy, your **changes take effect** within 60 seconds. In some cases, though, it **can take** up to 7 minutes for **the change** to fully propagate across Google Cloud.

How can I add User to GCP Console :

**Granting access to new Member in GCP**

1. In the Cloud **Console**, go to the IAM page..
2. Click **Add**.
3. Enter an email address. You can **add** individuals, service accounts, or Google Groups as members, but every project must have at least one individual as a member.
4. Select a role. ...
5. Click Save.

How to add new member to specific Group in GCP

To **add** members: Click person **Add** members at the top of the page. Enter the names of the members you want to **add**, choose their Google **Groups** roles, then click **Add** to **add** them to the **group**

Service account:

What is GCP service account?

A **service account** is a special type of **Google account** intended to represent a non-human user that needs to authenticate and be authorized to access data in **Google** APIs. Typically, **service accounts** are used in scenarios such as: Running workloads on virtual machines (VMs).

How do I create a service account in GCP?

1. In the Cloud Console, go to the **Service** accounts page. ...
2. Select a project.
3. Click **Create service account**.
4. Enter a **service account** name to display in the Cloud Console. ...
5. Optional: Enter a description of the **service account**.
6. If you do not want to set access controls now, click Done to finish creating the **service account**.

How do I log into my GCP service account?

1. Create a **service account** with **GCP** console.
2. Download the json key file.
3. Create a role and assign proper required permissions to the role.
4. gcloud config set **account** gcli@someproject.iam.gserviceaccount.com.
5. gcloud auth activate-**service**-**account** — key-file=gcli.json — project=someprojectname

How do I assign roles to Google groups in GCP?

**Using Google Groups**

1. Sign in to Google **Groups**.
2. Click the name of a **group**.
3. On the left, click Members.
4. To **assign** a **role** to a single member—Point to a member and in the **Role** column, select a **role**.
5. To **assign** a **role** to multiple members:

What is the difference between user account and service account?

**User accounts** are used by real **users**, **service accounts** are used by system **services** such as web servers, mail transport agents, databases etc. By convention, and only by convention, **service accounts** have **user** IDs **in the** low range, e.g. < 1000 or so

-----------------------------------------------------------------------------------------------

Now will discuss one more GCP Service called Compute Engine

GOOGLE COMPUTE ENGINE:

**Virtual Machine:**

A VM is a virtualized instance of a computer that can perform almost all of the same functions as a computer, including running applications and operating systems.

**Google Compute Engine VM Creation:**

1.Open GCP Console and Google Compute Engine

2.Select Create Instance

3.Specify a **Name** for your VM.

4.Optional: Change the **Zone** for this VM. Compute Engine randomizes the list of zones within each region to encourage use across multiple zones.

5.Select a **Machine configuration** for your VM.

6.In the **Boot disk** section, click **Change** to configure your boot disk. Unless you explicitly choose a different boot disk, if the name of the new VM matches the name of an existing persistent disk, then the existing persistent disk automatically attaches to the new VM as the boot disk.

7.In the **Public images** tab, choose the following:

* + Operating system
  + OS version
  + Boot disk type
  + Boot disk size

8. Click **Save** to confirm your boot disk options.

9.Select Allow HTTP traffic or Allow HTTPS traffic to permit HTTP or HTTPS traffic to the VM. When you select one of these, Compute Engine adds a network tag to your VM, which associates the firewall rule with the VM. Then, Compute Engine creates the corresponding ingress firewall rule that allows all incoming traffic on tcp:80 (HTTP) or tcp:443 (HTTPS).

10.Optional: If you chose an OS image that supports Shielded VM features, you can modify the Shielded VM settings. To modify shielded VM settings, click the Security tab in the Management, security, disks, networking, sole tenancy section and do the following, as required:

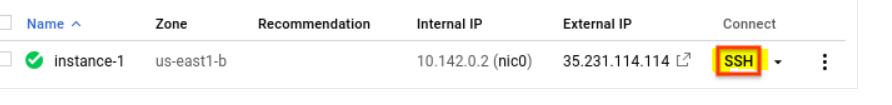
To enable Secure Boot, select Turn on Secure Boot. Secure Boot is disabled by default.

**Click Create to create and start the VM.**

**How to Connect Linux VM:**

**In the GCP Cloud Console, go to the VM instances page**

**In the list of virtual machine instances, click SSH in the row of the instance that you want to connect to**

****

**After you connect, use the terminal to run commands on your Linux instance. When you have finished, disconnect from the instance by using the exit command.**

**Instance Templates:**

**An instance template is a resource that you can use to create virtual machine (VM) instances. Instance templates define the machine type, boot disk image or container image, labels, and other instance properties.**

**Instance templates allow you to predefine a machine configuration and image to start up clones of your servers automatically.**

**How to create Instance Template:**

**To create an instance template through the Google Cloud Platform Console, follow these steps:**

* **In the GCP Console, go to the Instance Templates page.**
* **Click Create instance template.**
* **Fill in the fields you want for your instance template, or accept the default values. The following values are provided by default:**
  + **Machine type: n1-standard-1**
  + **Image: The latest Debian image**
  + **Boot disk: A new standard boot disk named after the instance**
  + **VPC network: The default VPC network**
  + **IP address: An ephemeral external IP address**
* **Optionally, change the instance's Shielded VM settings, if you chose an image that supports Shielded VM.**
* **Optionally, click on the tabs in the Management, Security, Disks, Networking, Sole Tenancy section to further customize your template. For example, you can add up to 15 secondary non-boot disks.**
* **Click Create to create the template.**

**Instance Groups:**

**An instance group is a collection of virtual machine (VM) instances that can be managed as a single entity.**

**Compute Engine offers two kinds of VM instance groups**

**Managed instance groups (MIGs):**

**1.allows app creation with multiple identical VMs.**

**2.workloads can be made scalable and highly available by taking advantage of automated MIG services, including: autoscaling, autohealing, regional (multiple zones) deployment, and automatic updating**

**Unmanaged instance groups:**

**Unmanaged instance groups do not offer autoscaling, autohealing, rolling update support, multi-zone support, or the use of instance templates and are not a good fit for deploying highly available and scalable workloads.**

**Sole tenant nodes:**

**Sole-tenant nodes are physical Compute Engine servers dedicated solely to your workloads. Sole-tenant nodes offer the same machine types and options as regular compute instances, including custom machine shapes and transparent maintenance, but on servers dedicated exclusively to your use.**

**Machine Images and Snapshot:**

**An image is a complete backup of your server including all volumes. A snapshot can be done from a specific volume (for example you have a server with a volume containing the OS and another one containing the application data, and you want to use different snapshot strategies on both volumes).**

# Committed use discounts:

**Commitments provide discounts for VMs and software licensing in return for 1-3 year purchase agreements. Can be a good option if your workloads have predictable resource needs.**

**Disk:**

**Persistent Disk is designed for high durability. It stores data redundantly to ensure data integrity. ... Your storage is located independently from your virtual machine instances, so you can detach or move your disks to keep your data even after you delete your instances.**

**Snapshots:**

**Snapshots are backups of persistent disks. They're commonly used to recover, transfer or make data accessible to other resources in your project.**

**Create snapshot schedules to automatically back up your data from the disk.**

**Gcloud Commands for GCE:**

| **PURPOSE** | **COMMAND** |
| --- | --- |
| **List all instances** | **gcloud compute instances list , gcloud compute instance-templates list** |
| **Show instance info** | **gcloud compute instances describe "<instance-name>" --project "<project-name>" --zone "us-west2-a"** |
| **Stop an instance** | **gcloud compute instances stop instance-name** |
| **Start an instance** | **gcloud compute instances start instance-name** |
| **Create an instance** | **gcloud compute instances create vm1 --image image-1 --tags test --zone "<zone>" --machine-type n1-standard-2** |

**Now we will discuss one more GCP Service called Cloud source repository.**

**Google Cloud Source Repository:**

Google Cloud Source Repositories is a single place for your team to store, manage, and track code. Cloud Source Repositories allows developers to collaborate easily and more securely manage your code on a fully featured, scalable, private Git repository.

Benefits of Using Google Cloud Source Repositories:-

1. **Multiple Repositories** - For a single google cloud project, you can create multiple cloud repositories to organize the code related to your project. It allows developers to create upto 1000 repositories per GCP Project.

2. **Integration with GCP Service**s - Google Cloud Source Repositories can easily be used by connecting with other GCP Services such as Cloud Build, App Engine, Cloud Pub/Sub, Stackdriver.

3. **Code Search** - Cloud Source Repositories utilizes the same search technologies like google search while optimizing the search indexing algorithms.

4. **Source Browser** - It also provides a source browser that can be used to view repository files from within the GCP Console.

5. **Auto Sync with other hosted repositories** - You can connect with other hosted repositories and automatically sync changes to cloud source repositories whenever the changes are pushed to github or bitbucket.

6. **Git Operations** - You can perform push, pull, clone and log(basic git operations) Also you can make changes based on the requirement of the workflow.

**Create Repo from Cloud Shell: gcloud source repos create test**

**Clone GCSR Repo: gcloud source repos clone test --project=my-project**

**How to Setup a New Repository on Google Cloud Source Repositories:-**

Step 1: Select Project within the GCP Console

Step2: Go to Source Repositories

Step 3: create Repo by selecting create Repo

Step 4: Activate Cloud Shell - Click the ‘Activate Cloud Shell’ button at the top of the Google Cloud Console. A ‘Cloud Shell’ session opens inside a new frame at the bottom of the Console and displays a command-line prompt. It can take a few seconds for the session to get initialized.

Step 5: To create a demo repository, type the following-

gcloud source repos create my-demo-repo

Step 6: To create a clone demo repository, type the following-

gcloud source repos clone my-demo-repo

Step 7: create file (vi text.txt)

Step 8: git add file1.txt

Step 9: git commit -m "file1 committed"

Step 10: git config --global user.email"kirankumarrv0990@gmail.com"

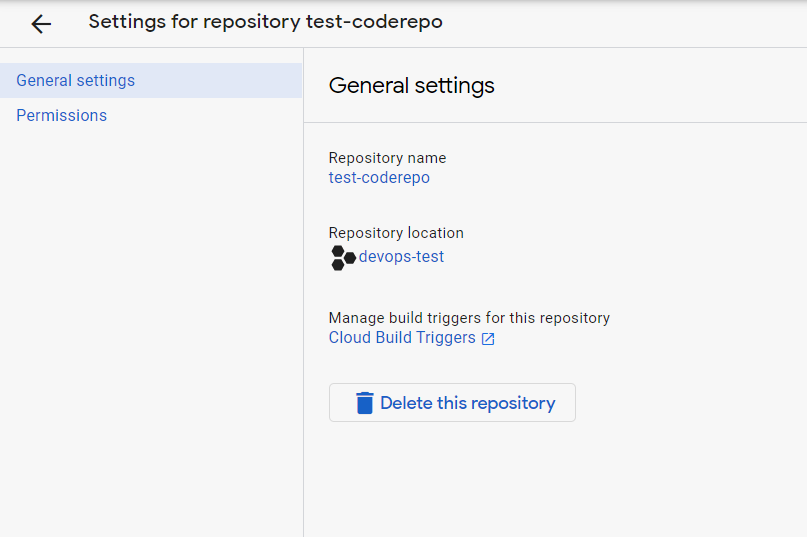
Step 11: git config --global user.name "Kiran Kumar"

Step 12: git commit -m "file1 committed"

Step 13: git push origin master

Step 14: Click on the repo menu and go to the source repository and you can finally find the demo repo and also one can see the content added.

**Delete Repo:**

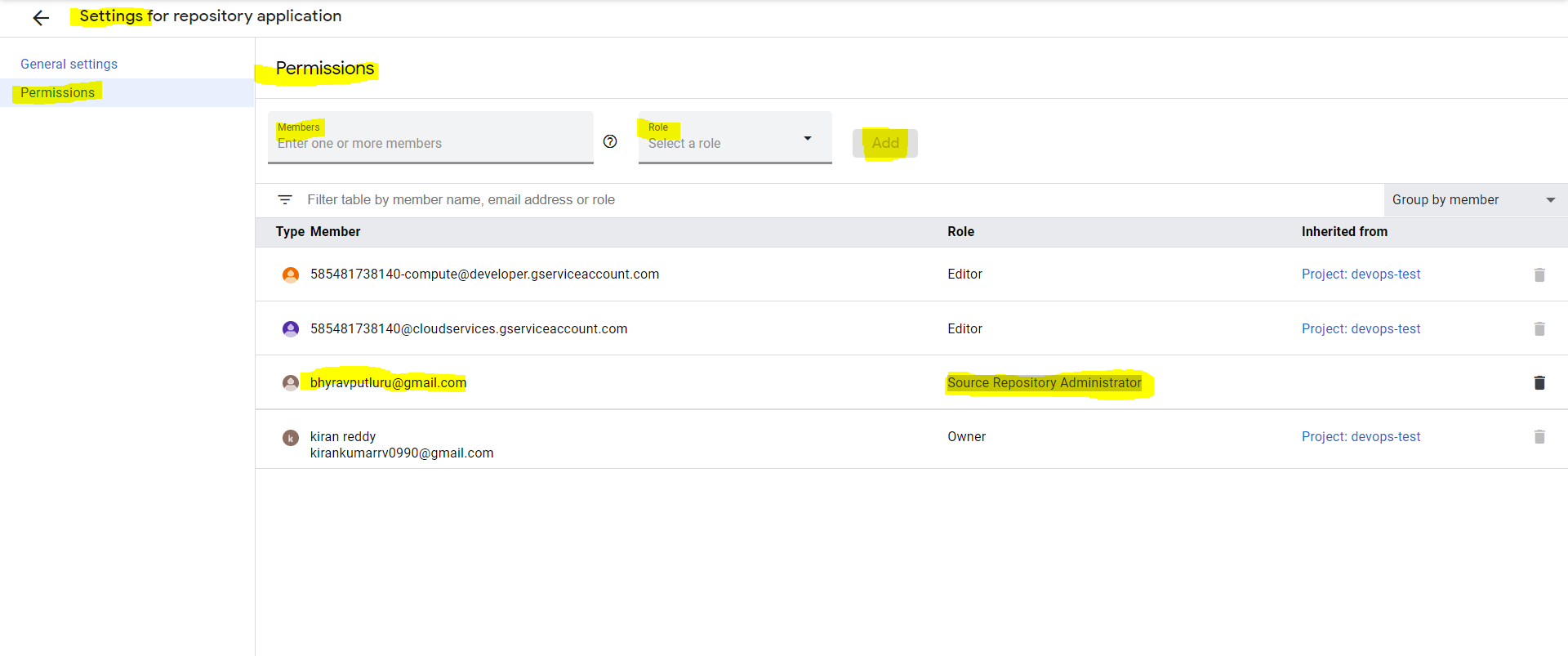
****

**Goto settings - Select Delete this Repository and type the repo name to delete.**

**How to give permissions to Access your Cloud Repo:**

**Go to Repo settings -> permissions -> select Google Account (members) and proper Role -> ADD**

**So respective google users will access that repo from now**

****

Now will discuss one more GCP Service called GCS..

**Google Cloud Storage : (GCS )**

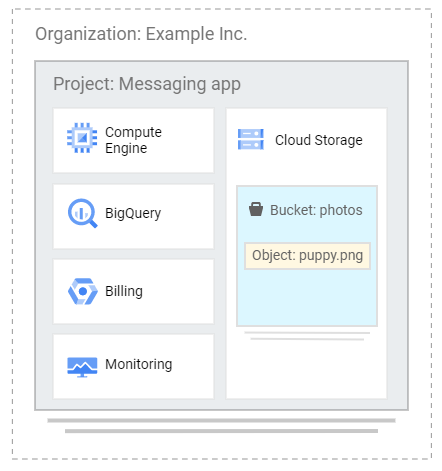
Google Cloud Storage is the object storage service provided by Google Cloud Platform.Google Cloud Storage pricing is also quite low, making this a very popular service for enterprise storage deployments that have vast amounts of data on hand.

Like any cloud-based object storage service, Google Cloud Storage enables close to unlimited storage capacity and out-of-the-box high availability.

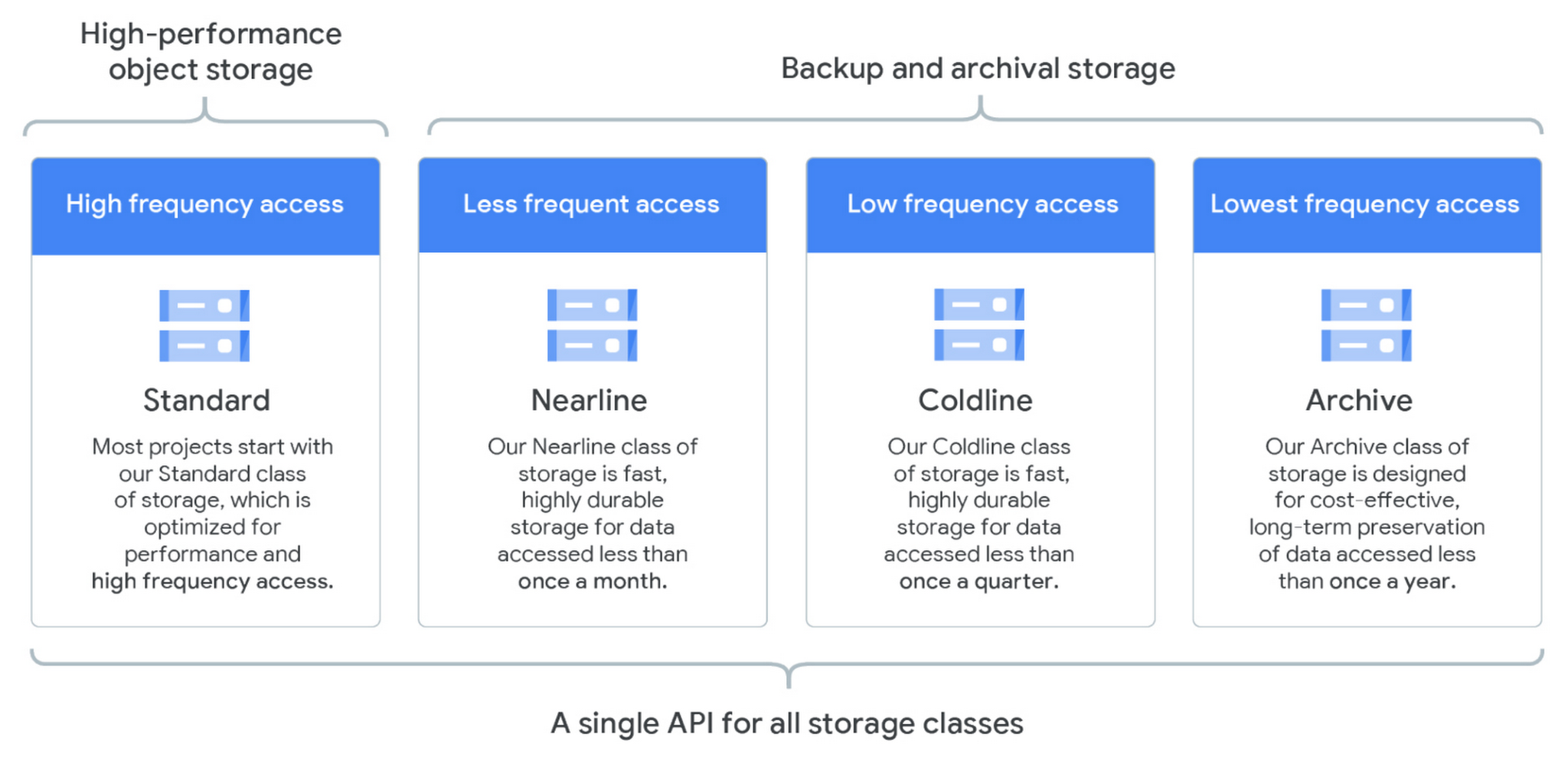
GCS is an object storage service that stores data within buckets.

### **Buckets**

* The data you upload on Cloud Storage are called **objects.**
* An object is an **immutable piece** of data consisting of a **file in any format.**
* You store objects inside containers called **buckets**.
* All buckets belong to a project.
* Each project can have **multiple buckets.**



* Standard Storage
  + Good for hot data that is accessed frequently.
* Nearline Storage
  + Good for use cases that need to store objects for at least 30 days.
  + Ideal for data that you plan to access once per month or less.
* Coldline Storage
  + Is a low-cost storage option for storing infrequently accessed data within 90 days.
* Archive Storage
  + Is the coldest storage among the storage classes.
  + Designed for storing archive data and disaster recovery data that is expected to be accessed once per 365 days or less.



Available storage classes

| **Storage Class** | **Name for APIs and gsutil** | **Minimum storage duration** |
| --- | --- | --- |
| Standard **Storage** | STANDARD | None |
| Nearline **Storage** | NEARLINE | 30 days |
| Coldline **Storage** | COLDLINE | 90 days |
| Archive **Storage** | ARCHIVE | 365 days |

### **Pricing**

* Pricing for Cloud Storage services is based on what you use, including:
  + the amount of data you store,
  + the duration for which you store it,
  + the number of operations you perform on your data,
  + the network resources used when moving or accessing your data.
* For “cold” storage classes meant to store long-term, infrequently accessed data, there are also charges for retrieving data and early deletion of data.
* You can require accessors of your data to include a project ID to bill for network charges, operation charges, and retrieval fees.

### **gsutil :**

* **gsutil** is Google Storage CLI tool. Equivalent to aws s3 but for the Google Cloud Platform, it allows you to access Google Cloud Storage from the command line.
* You can use gsutil to perform bucket and object management tasks like:
  + creating and deleting buckets
  + uploading, downloading, and deleting objects
  + listing buckets and objects
  + moving, copying, and renaming objects

How do I access my GCS bucket?

**To create a bucket:**

1. In the **Google Cloud** Console, go to the Cloud Storage Browser page. Go to Browser.
2. Click Create **bucket** to open the **bucket** creation form.
3. **Enter** your **bucket** information and click Continue to complete each step: **Enter** a unique Name for your **bucket**. ...
4. Click Create.

How do I upload a file to GCS bucket?

**In the Objects tab for the bucket, either:**

1. Drag and drop the desired **files** from your desktop or **file** manager to the main pane in the Cloud Console.
2. Click the **Upload Files** button, select the **files** you want to **upload** in the dialog that appears, and click Open.

Retention policy:

Set a retention policy to specify the minimum duration for which this bucket's objects must be protected from deletion or modification after they're uploaded. You might want to set a policy to address industry-specific retention challenges.

* You can add a *retention policy* to a bucket to specify a *retention period*.
  + If a bucket does not have a retention policy, you can delete or replace objects in the bucket at any time.
  + If a bucket has a retention policy, objects in the bucket can only be deleted or replaced once their age is greater than the retention period.
  + A retention policy retroactively applies to existing objects in the bucket as well as new objects added to the bucket.
* You can *lock* a retention policy to **permanently** set it on the bucket.
  + Once you lock a retention policy, you cannot remove it or reduce the retention period it has.
  + You cannot delete a bucket with a locked retention policy unless every object in the bucket has met the retention period.
  + You can increase the retention period of a locked retention policy.

Retention periods are measured in seconds; however, some tools, like the GCP Console and gsutil allow you to set and view retention periods with other units of time for convenience. The following conversions apply in such cases:

* A day is considered to be 86,400 seconds.
* A month is considered to be 31 days, which is 2,678,400 seconds.
* A year is considered to be 365.25 days, which is 31,557,600 seconds.

How to give access to one folder in gcs

You **cannot** do this in GCS. GCS provides permissions to buckets and permissions to objects. A "folder" is not a GCS concept and does not have any properties or permissions.

Gsutil commands:

gsutil ls ---> lists all your buckets

gsutil mb gs://<bucket\_name> -> creates bucket.

gsutil rb gs://<bucket\_name> -> deletes the bucket.

gsutil cp <filename> gs://<bucket\_name>/ --> copies the local filename into the bucket

gsutil cp <filename> gs://<bucket\_name>/directory/ --> copies the local filename into the directory

gsutil mv <src\_filename> gs://<bucket\_name>/directory/ --> moves the local src\_filename to the bucket directory

gsutil rm gs://<bucket\_name>/file\_or\_dir --> deletes the file\_or\_dir object.

gsutil rb gs://flipkart-qa-login/Test - > Remove objects from the GCS Bucket and then only you have to delete the bucket and delete bucket and its content with below command.

gsutil -m rm -r gs://flipkart-qa-login/ -> it should initially delete the bucket objects and then bucket as we cant delete bucket directly.

gsutil du -h gs://packt-gcp/ -> How much storage are we using (the -h makes it readable)

gsutil cp -r gs://packt-gcp/ -> Copying a whole folder to a bucket

Practice:

**Copy files to Storage Bucket:**

gsutil cp \* gs://<YOUR\_BUCKET\_NAME>

gsutil cp kiran.txt README-cloudshell.txt gs://payment-module/

Copying file://kiran.txt [Content-Type=text/plain]...

Copying file://README-cloudshell.txt [Content-Type=text/plain]...

- [2 files][ 927.0 B/ 927.0 B]

Operation completed over 2 objects/927.0 B.

**List of files in bucket:**

gsutil ls gs://<YOUR\_BUCKET\_NAME>

gsutil ls gs://payment-module/

gs://payment-module/README-cloudshell.txt

gs://payment-module/celcom.txt

gs://payment-module/devops-test-316311-bf60cb3f08d4.json

gs://payment-module/kiran.txt

**Remove files from Google Storage Bucket:**

gsutil rm gs://<YOUR\_BUCKET\_NAME>/\*

gsutil rm gs://payment-module/celcom.txt

Removing gs://payment-module/celcom.txt...

/ [1 objects]

Operation completed over 1 objects.

**Delete Storage Bucket:**

gsutil rb gs://<YOUR\_BUCKET\_NAME>

gsutil rb -r gs://payment-module/

GCS Bucket Life Cycle:

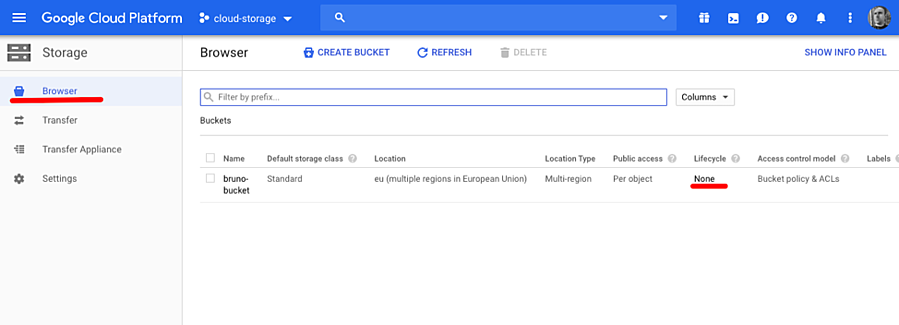
Lifecycle rules provide the ability to manage different aspects of the objects’ lifecycle inside a given bucket. Rules can be triggered based on certain conditions such as age, storage class, date, state, and version. If the defined conditions are met, objects can be either deleted or moved to another of the [Google Cloud Storage classes](https://cloud.netapp.com/blog/google-storage-service-how-to-switch-google-cloud-storage-class).

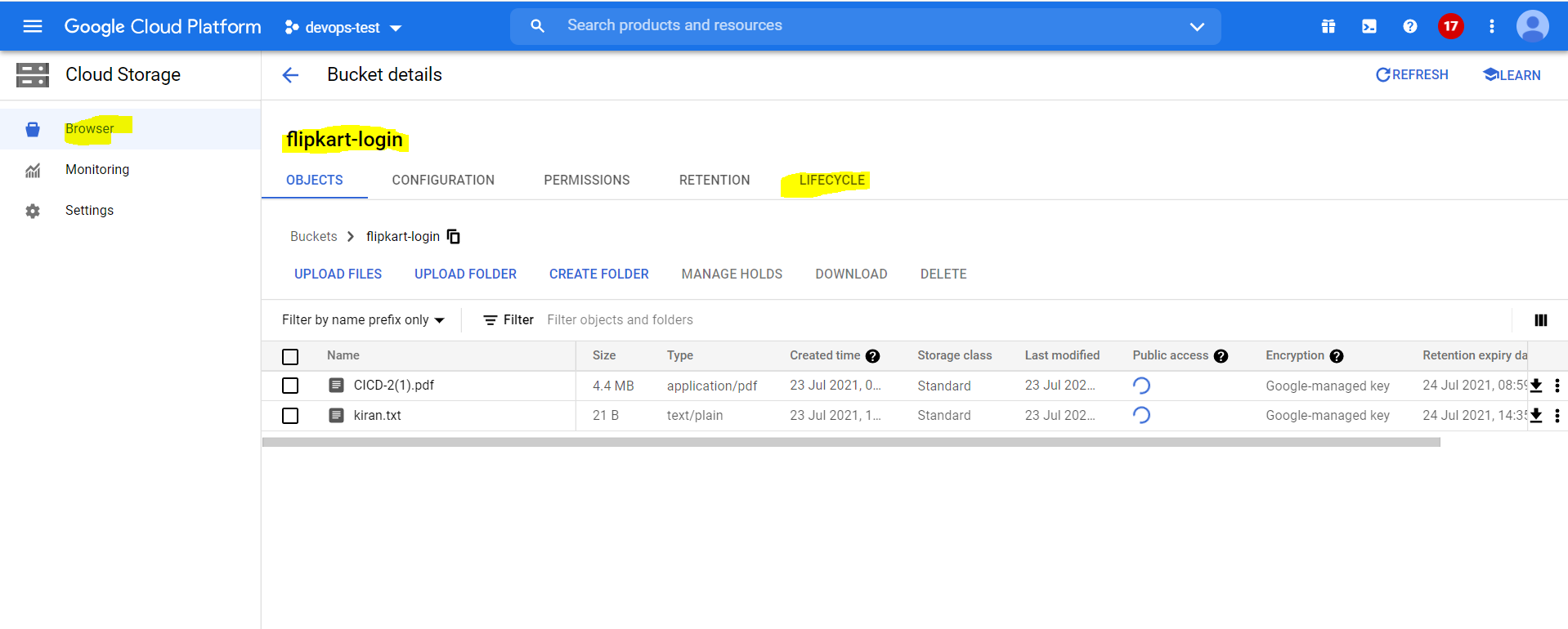
Once your bucket is created and the files are uploaded you can proceed with setting up lifecycle rules to manage them.

### **How to Set Up a Lifecycle Rule to Automatically Delete Data:**

As mentioned above, the built-in lifecycle management feature enables you to automate different aspects of the data lifecycle in a bucket. One possible real-world scenario is to automatically delete data older than a specific number of days.

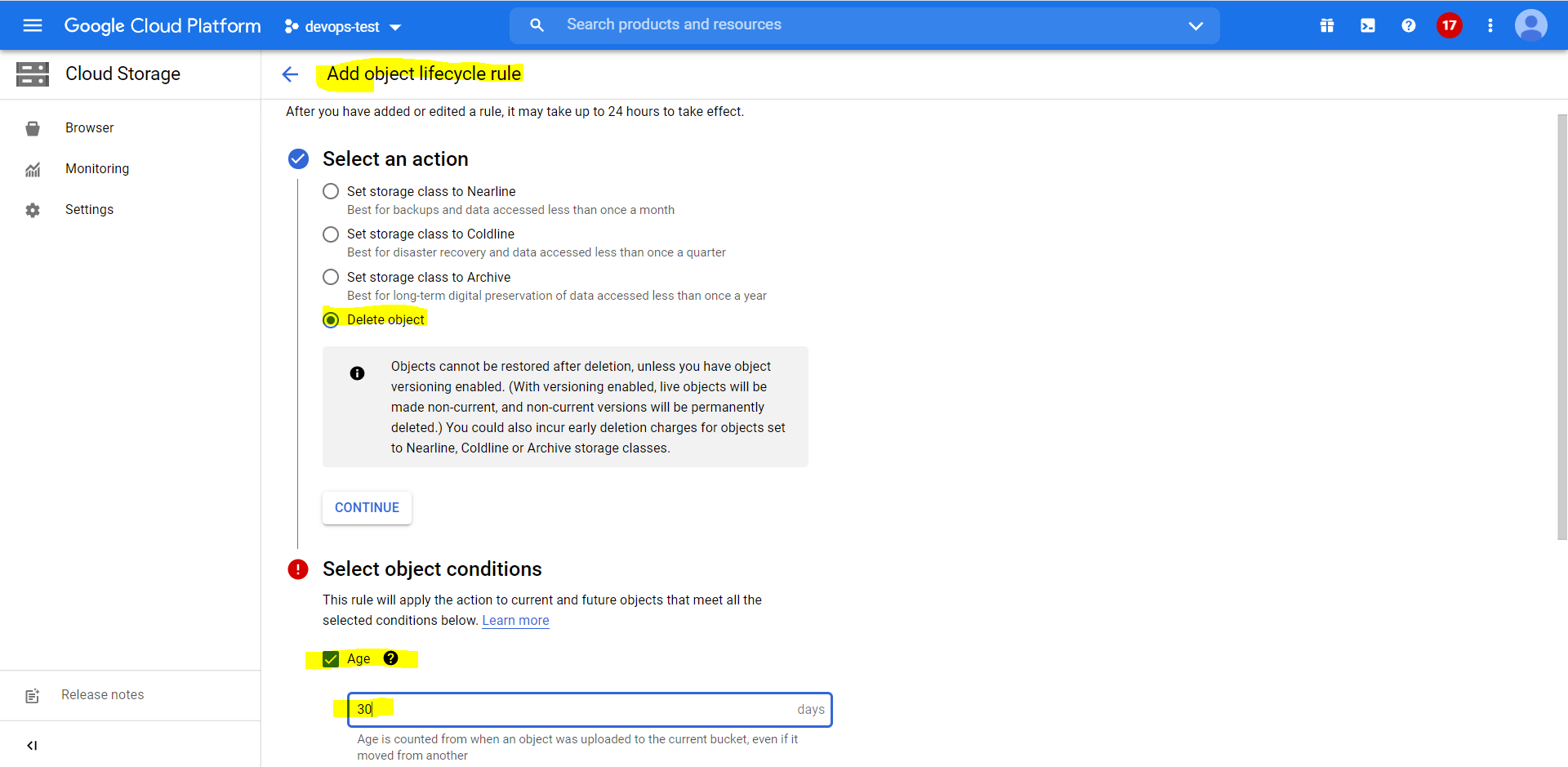
1.Select Bucket from GCP Console - > select Life Cycle ->





Select **Life Cycle** - > Select **Add Rule** -> Select **Delete Object** as an Action

-> Select **Age** as an Object Condition



Once you save the Rule, the rule will immediately become active. Your rule to delete data after a certain period of time (30 days) is now set. This is a powerful option that enables the automation of the object lifecycle without the need for custom code or servers.

**Conclusion:**

**The Object Lifecycle Management feature is a powerful tool that, when understood and used properly, can bring down the overall cost of your object storage usage on Google Cloud Platform.**

**Permissions:**

**Cloud Storage uses access control lists (ACLs) to manage object and bucket access. ACLs are the mechanism you use to share objects with other users and allow other users to access your buckets and objects.**

**Grant the user** [**kirankumarrv0990@gmail.com**](mailto:kirankumarrv0990@gmail.com) **WRITE access to the bucket example-bucket:**

**gsutil acl ch -u kirankumarrv0990@google.com:WRITE gs://example-bucket**

**Grant the group devops@google.com OWNER access to all jpg files in the top level of example-bucket:**

**gsutil acl ch -g devops@google.com:O gs://example-bucket/\*.jpg**

**Remove any current access by** [**kirankumarrv0990@google.com**](mailto:kirankumarrv0990@google.com) **from the bucket example-bucket:**

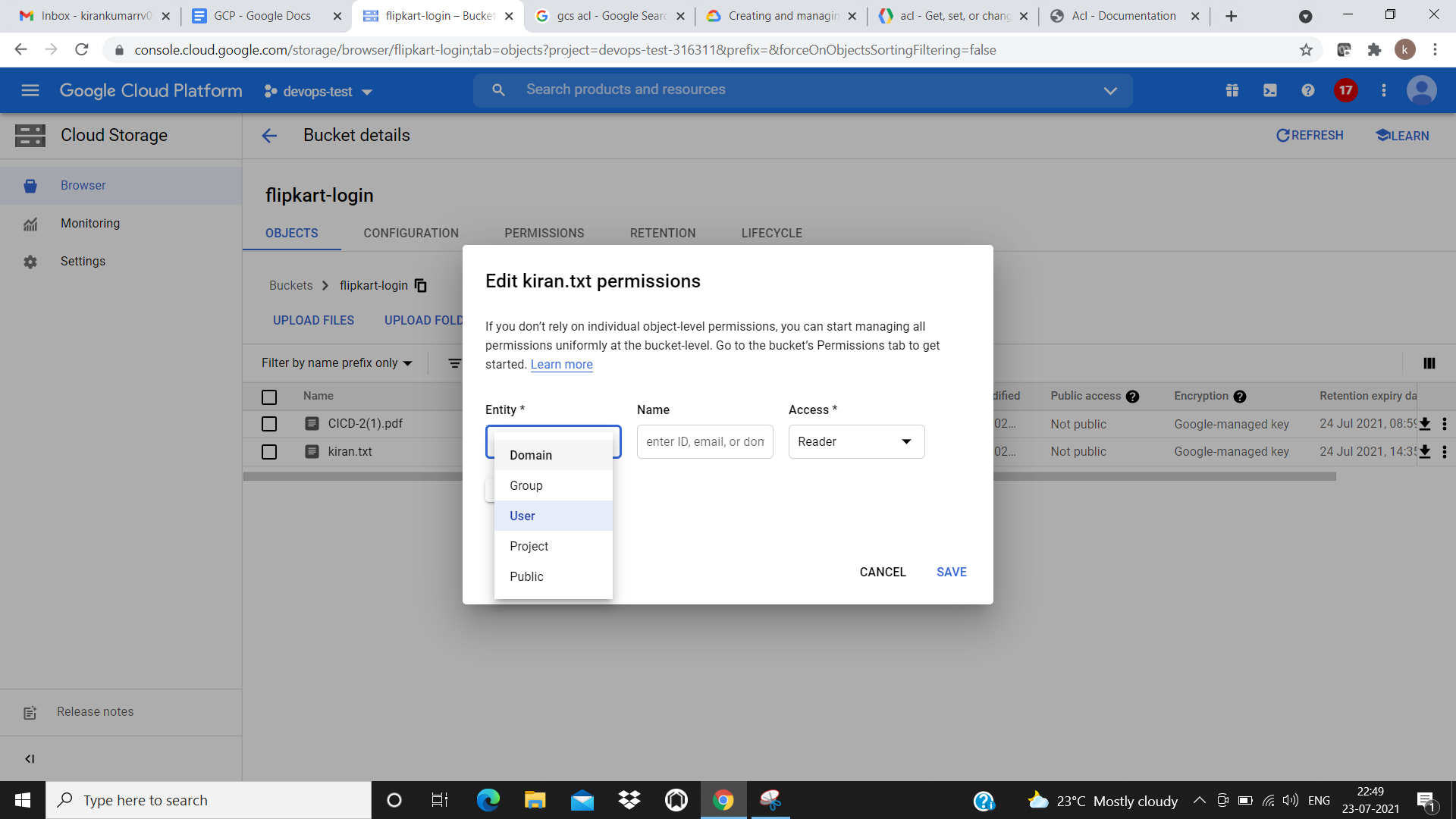
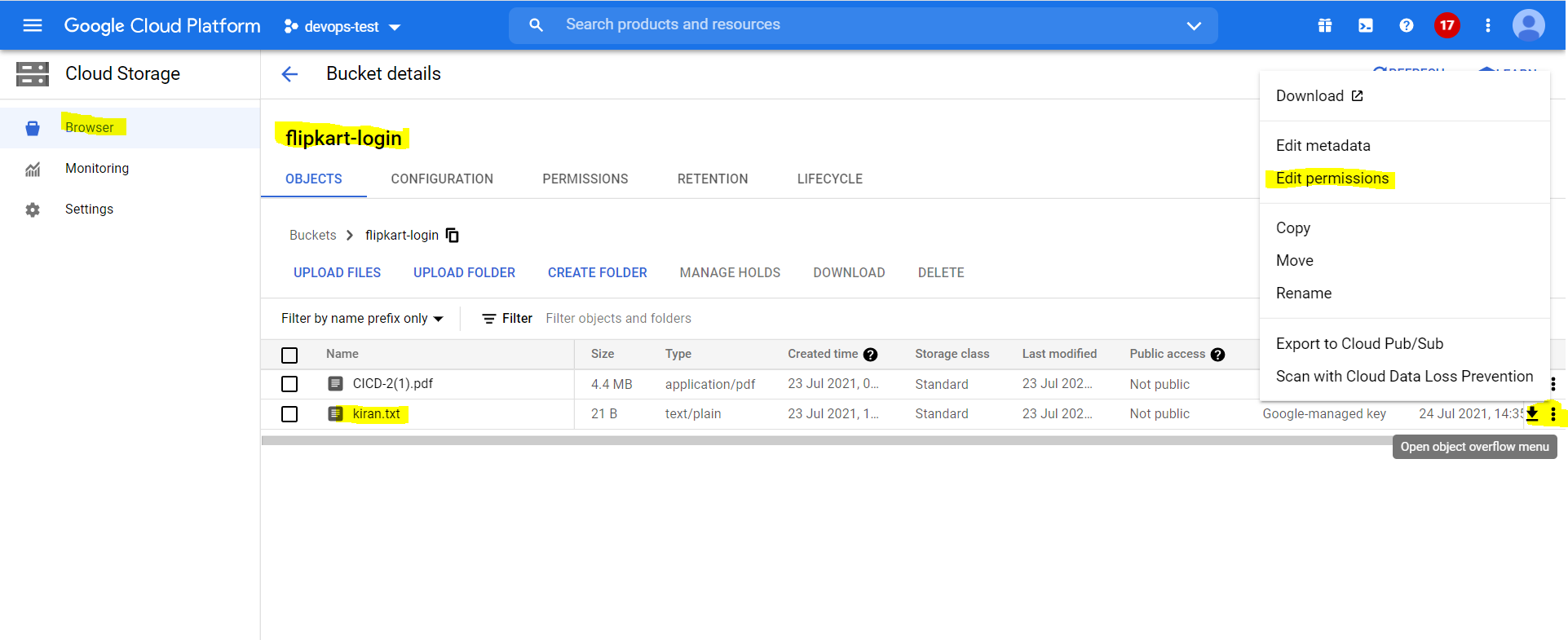
**gsutil acl ch -d** [**kirankumarrv0990@gmail.com**](mailto:kirankumarrv0990@gmail.com) **gs://example-bucket**

**We can do the same stuff from the GCP Console as shown below.**

**Go to the Cloud Storage browser in the Cloud Console**

**Navigate to the object whose ACL you want to modify.**

**Choose Edit Permissions from the more actions menu () associated with the object.**

****

**Entity specifies the type of thing that's getting the permission (for example a user or a group).**

**Name identifies a specific user, group, or other entity type.**

**Access defines the permission that you want to set on the object. (READER/OWNER).**

**Click Save.**

**Practice:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gsutil acl ch -u kirankumarrv0990@gmail.com:O gs://flipkart-login/kiran.txt**

**CommandException: Failed to set acl for gs://flipkart-login/kiran.txt.**

**Please ensure you have OWNER-role access to this resource.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gsutil acl ch -u bhyravputluru@gmail.com:O gs://flipkart-login/kiran.txt**

**Updated ACL on gs://flipkart-login/kiran.txt**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ gsutil acl ch -d bhyravputluru@gmail.com gs://flipkart-login/kiran.txt**

**Updated ACL on gs://flipkart-login/kiran.txt**

**BigQuery:**

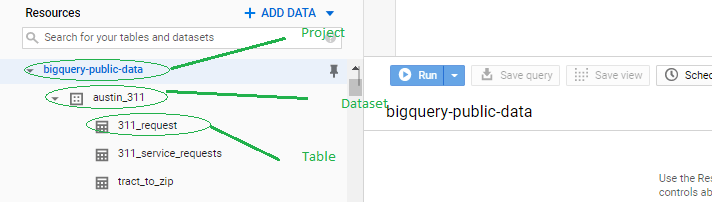
<https://hevodata.com/learn/bigquery-tutorial-a-comprehensive-guide/>

Google BigQuery is a **cloud-based big data analytics web service** for processing very large read-only data sets. BigQuery was designed for analyzing data on the order of billions of rows, using a SQL-like syntax.

Google BigQuery is a serverless, highly scalable data warehouse that comes with a built-in query engine. The query engine is capable of running SQL queries on terabytes of data in a matter of seconds, and petabytes in only minutes.

Google’s BigQuery service follows a four-layer structure. The first layer is known as projects, which act as a top-level container for the data you want to store in the [Google Cloud Platform.](https://cloud.google.com/) Datasets make up the second layer of Google BigQuery. You can have single or multiple datasets in a particular project.

The third layer is known as the tables, which stores your data in the form of rows and columns. Just like datasets, you can have single or multiple tables in a dataset. The final layer of BigQuery is known as jobs and its all about executing SQL queries to fetch, insert and modify data.



BigQuery Project:

BigQuery projects function as the top-level container for your data. Each project has a unique name and id, which makes storing, accessing and removing data from BigQuery, a smooth process

gcloud projects create PROJECT\_ID

### **BigQuery Datasets:**

BigQuery datasets act as the container for your tables and views, with each dataset having multiple tables that store your data. With datasets, you can manage, control and access your data from tables and views.

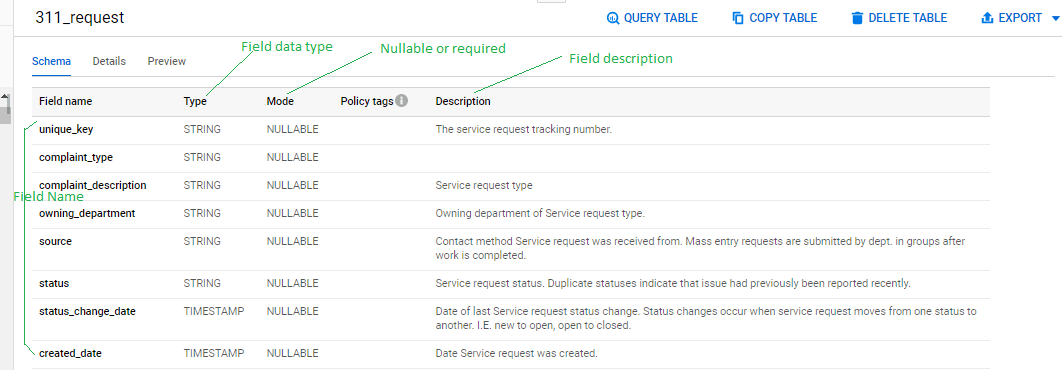
You can create a dataset in BigQuery using the [bq command](https://cloud.google.com/bigquery/docs/bq-command-line-tool) as follows:

bq mk test\_dataset

### **BigQuery Tables:**

BigQuery stores your data in the form of rows and columns in numerous tables

Each BigQuery table follows a particular schema that describes the columns, their name and datatypes.



BiqQuery Jobs:

BigQuery jobs refer to the operations you perform on your data. With BigQuery, you can perform four different operations/tasks, namely, load, query, export and copy on the data you’ve stored in the BigQuery. Every time you execute one of these tasks, it automatically creates a job.

How to use Bigquery:

BigQuery is available in Google Cloud Platform. GCP customers can easily access the service from their familiar web interface console.

##### **Grant Project access:**

To run Load jobs, grant the **BigQuery Data Editor**, **Job User**, and **User** roles to the authenticated user (the user you logged in as) for the relevant BigQuery Dataset Project.

1.Access your Google Cloud Console.

2.Click the **Project name** for which you want to grant access.

3.Open the sidebar menu by clicking the hamburger (3 lines) button in the top left, then choose **IAM & Admin**.

4.In the **IAM** menu click the **ADD** button.

5.Paste your Google Cloud user's email address under NEW MEMBERS. The user must be the owner of the project and dataset or have write permissions on both the project **and** dataset. (If you're using an existing user, you can click the edit button/pencil icon.)

6.Click the drop-down list and choose **BigQuery Data Editor** (or **Viewer** if you don't plan on using Alooma to create datasets) on the secondary drop-down list. Repeat this process to add **Job User** and **User**.

7.Click **Add**. If you're editing an existing user, click **Save**.

##### **Grant Dataset access:**

To enable User to write to a specific dataset, grant **WRITE** permissions to the authenticated user for the relevant BigQuery Dataset:

1.Access your Google Cloud Console.

2.Click the **Project name** for which you want to grant Alooma access.

3.In the sidebar menu, under **Big Data** click **BigQuery**.

4.In the new window, verify the **Project name** (to switch projects, click the down arrow next to the project name, then hover on "Switch to project" and select a different project).

5.On the dataset, then click the down arrow when it appears and click **Share Dataset**.

6.In the **Share Dataset** dialog, make sure the Google Cloud user has the "Is owner" or the "Can edit" permission for the dataset.

**7.**Click **Add**.

8.Click **Save changes**.

## 

## **Accessing BigQuery Data:**

For example, if you want to fetch data from the “bigquery-public-data” table, you can use the select statement as follows:

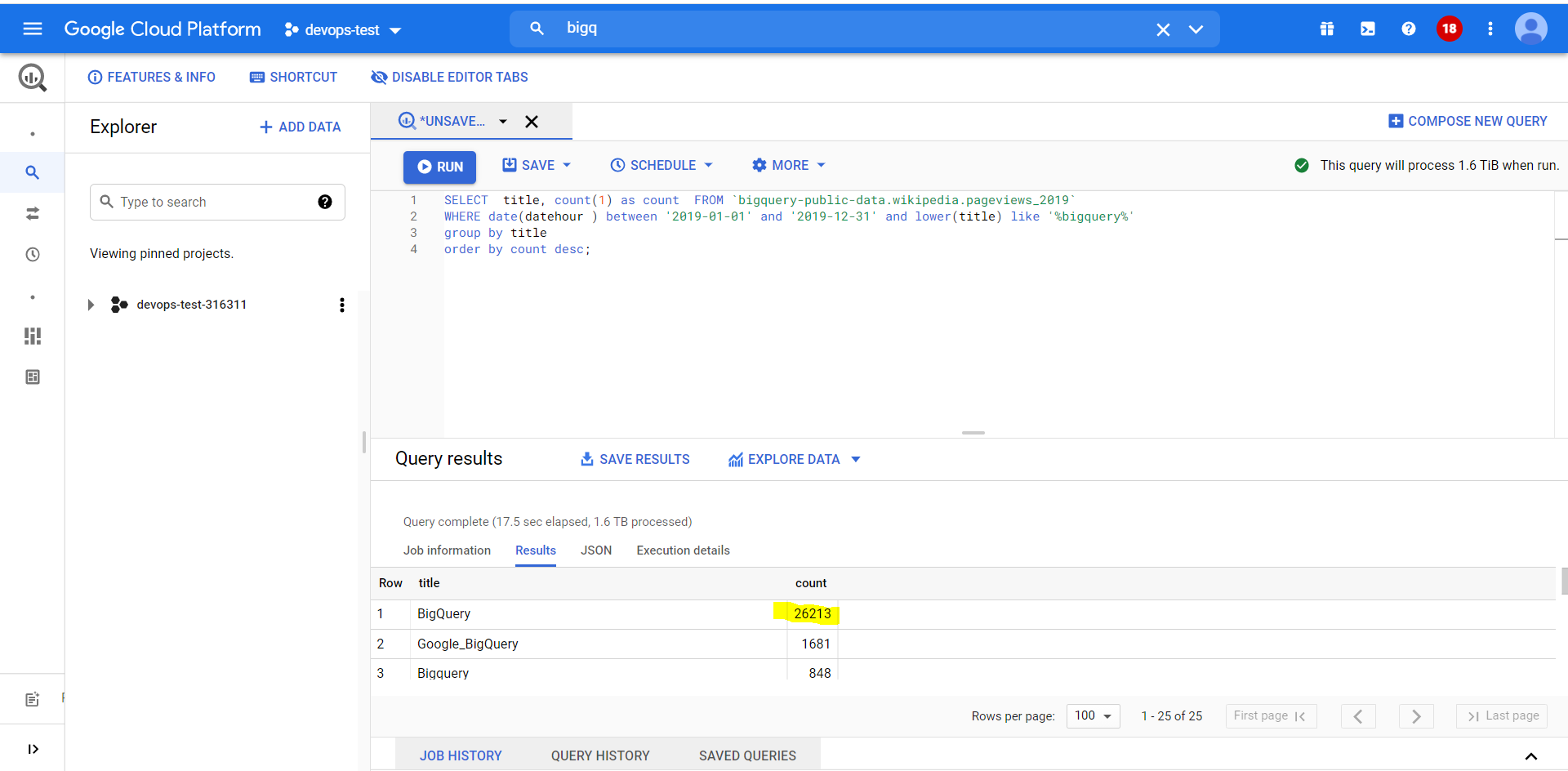
SELECT title, count(1) as count FROM `bigquery-public-data.wikipedia.pageviews\_2019`

WHERE date(datehour ) between '2019-01-01' and '2019-12-31' and lower(title) like '%bigquery%'

group by title

order by count desc;

This query will display the number of times the term “bigquery” featured as the title of a Wikipedia page in 2019, and it will generate the following output:



Practice:

Create Dataset:

kirankumarrv0990@cloudshell:~ (devops-test-316311)$ **bq mk test\_dataset**

Dataset 'devops-test-316311:test\_dataset' successfully created.

Give more details about Dataset:

kirankumarrv0990@cloudshell:~ (devops-test-316311)$ **bq show devops-test-316311:test\_bigquery**

Dataset devops-test-316311:test\_bigquery

Last modified ACLs Labels Type

----------------- ------------------------------- -------- ---------

26 Jul 09:04:21 Owners: DEFAULT

divyadeepu2423@gmail.com,

kirankumarrv0990@gmail.com,

projectOwners

Writers:

projectWriters

Readers:

projectReaders

List out the Datasets:

kirankumarrv0990@cloudshell:~ (devops-test-316311)$ **bq ls**

datasetId

---------------

google\_Data

test\_bigquery

test\_dataset

Why BigQuery is so fast?

Due to the separation between compute and storage layers, **BigQuery** requires an ultra-**fast** network which can deliver terabytes of data in seconds directly from storage into compute for running Dremel jobs. Google's Jupiter network enables **BigQuery** service to utilize 1 Petabit/sec of total bisection bandwidth

Is BigQuery a SQL or NoSQL?

**BigQuery** is a hybrid system that allows you to store data in columns, but it takes into the **NoSQL** world with additional features, like the record type, and the nested feature.

Is BigQuery a memory?

**BigQuery** is a fast petabyte-scale analytics database. To achieve that level of performance, **BigQuery** executes queries completely in **memory**. ... To support large-scale computation, shuffled data is transiently stored on remote servers

How do you run bq commands?

You can **run** the **bq command**-line tool in an interactive shell where you don't need to prefix the **commands** with **bq** . To **start** interactive mode, enter **bq** shell . After launching the shell, the prompt changes to the ID of your default project. To exit interactive mode, enter exit

How do you run a big query?

**Running interactive queries**

1. **In the Cloud Console, open the BigQuery page. Go to BigQuery.**
2. **Click Compose new query.**
3. **Enter a valid BigQuery SQL query in the Query editor text area.**
4. **(Optional) To change the data processing location, click More, then Query settings. ...**
5. **Click Run.**

**How do you run query BQ?**

**To run a query, run the command bq query " SQL\_STATEMENT " . Use a different quotation mark type than the surrounding marks ( " versus ' ).**

**To list objects, use the bq ls command with the following syntax:**

**bq ls [<*project\_id*>:][<*dataset\_id*>]**

### **List Datasets:**

**The syntax differs, depending on whether you have a default project or dataset defined:**

**# List datasets in the default project:**

**bq ls**

**# List datasets another project:**

**bq ls *my\_project\_id*:**

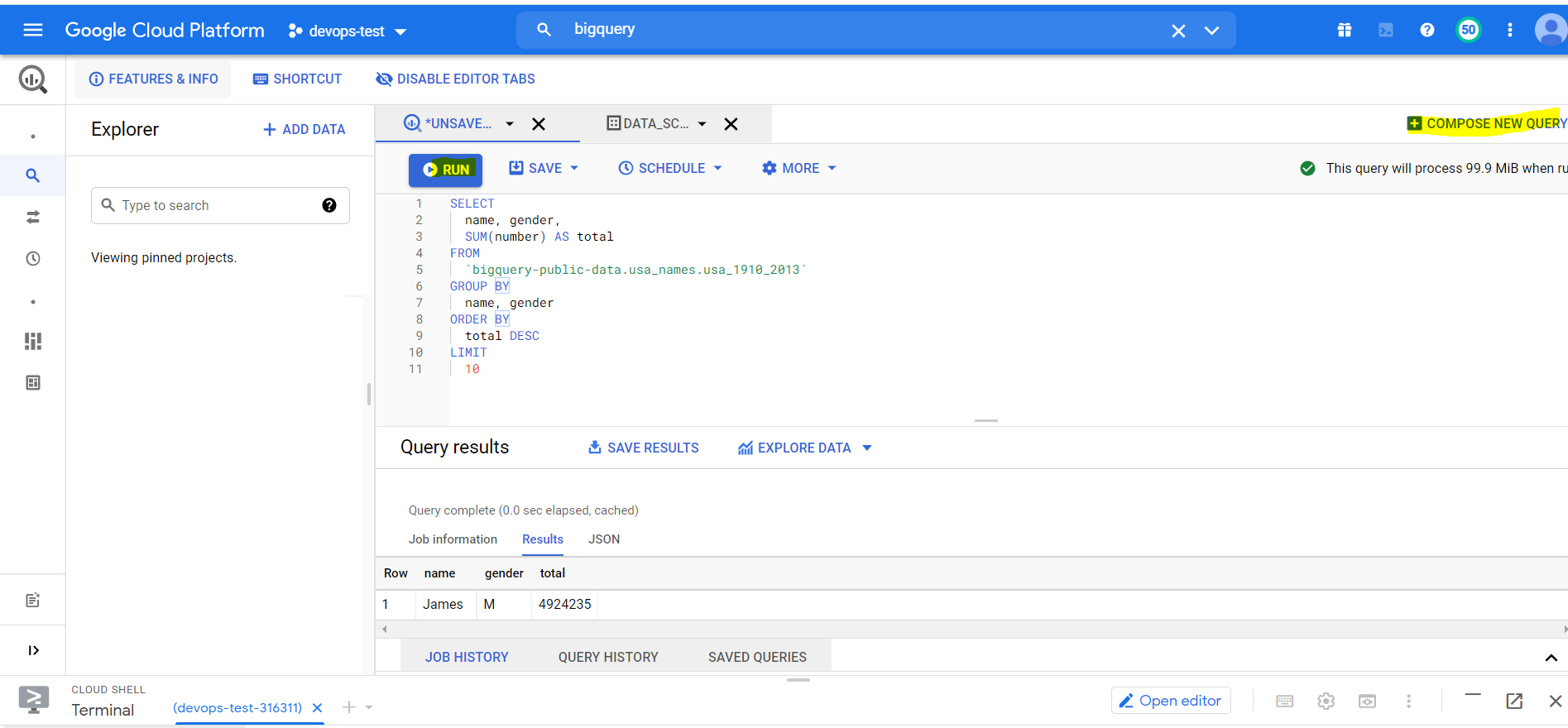
**# List datasets when you have a default dataset defined:**

**bq ls -d**

**OR**

**bq ls :**

**How to Run Bigquery from GCP Bigquery Console:**

****

**Same query Run from the CLI:**

**The following standard SQL query does a word count for the number of times that the substring raisin appears in all of Shakespeare's works.**

**bq query --use\_legacy\_sql=false \**

**'SELECT**

**word,**

**SUM(word\_count) AS count**

**FROM**

**`bigquery-public-data`.samples.shakespeare**

**WHERE**

**word LIKE "%raisin%"**

**GROUP BY**

**word'**

**Delete Dataset:**

**bq ls (will give bigquery dataset list)**

**datasetId**

**--------------**

**data\_sceince**

**bq rm -r devops-test-316311:data\_sceince (delete bq dataset)**

**rm: remove dataset 'devops-test-316311:data\_sceince'? (y/N) y**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ bq ls**

**BigQuery Command Line Tool:**

**How do I access the bq command-line tool?**

To get help for the bq command-line tool, you can enter the following commands:

For the installed version of the bq command-line tool, enter **bq** version .

For a full list of commands, enter **bq help** .

For a list of global flags, enter **bq --help** .

For help with a specific command, enter **bq help COMMAND**

**How to get details about Dataset:**

**bq show devops-test-316311:data\_sceince**

**dataset devops-test-316311:data\_sceince**

**Last modified ACLs Labels Type**

**----------------- ------------------------------- -------- ---------**

**29 Jun 04:19:55 Owners: DEFAULT**

**kirankumarrv0990@gmail.com,**

**projectOwners**

**Writers:**

**projectWriters**

**Readers:**

**projectReaders**

**How to Get the Dataset List:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ bq ls**

**datasetId**

**--------------**

**data\_sceince**

**BigTable:**

[**Google Cloud Bigtable**](https://cloud.google.com/bigtable/) **offers you a fast, fully managed, almost infinitely scalable NoSQL database service that's ideal for web, mobile, and IoT applications requiring terabytes to petabytes of data.**

**Use Cloud Bigtable as the storage engine that grows with you from your first gigabyte to petabyte**

**Cloud Bigtable is one of the best NoSQL available on the market. Google uses Cloud Bigtable for Search, Maps, and Gmail.**

**Bigtable is build upon the Google File System and stored in an immutable datastructure called SSTable**

**Fully managed service that integrates easily with big data tools like** [**Hadoop**](https://hadoop.apache.org/)**,** [**Dataflow**](https://cloud.google.com/dataflow)**, and** [**Dataproc**](https://cloud.google.com/dataproc)**.**

**Is bigtable no SQL?**

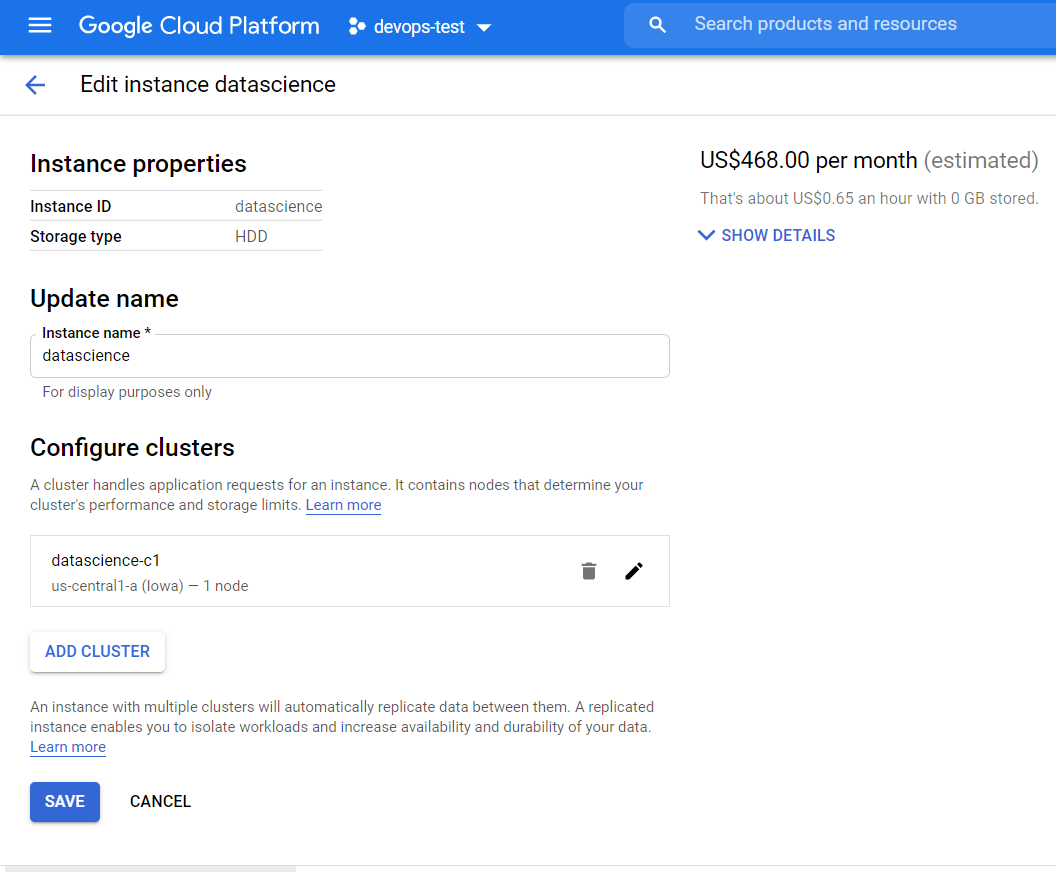
**Cloud Bigtable is a fast, fully managed, massively scalable NoSQL database service.**

**What is bigtable instance?**

**A Bigtable instance is a container for your data. Instances have one or more clusters, located in different zones. Each cluster has at least 1 node. A table belongs to an instance, not to a cluster or node. If you have an instance with more than one cluster, you are using replication.**

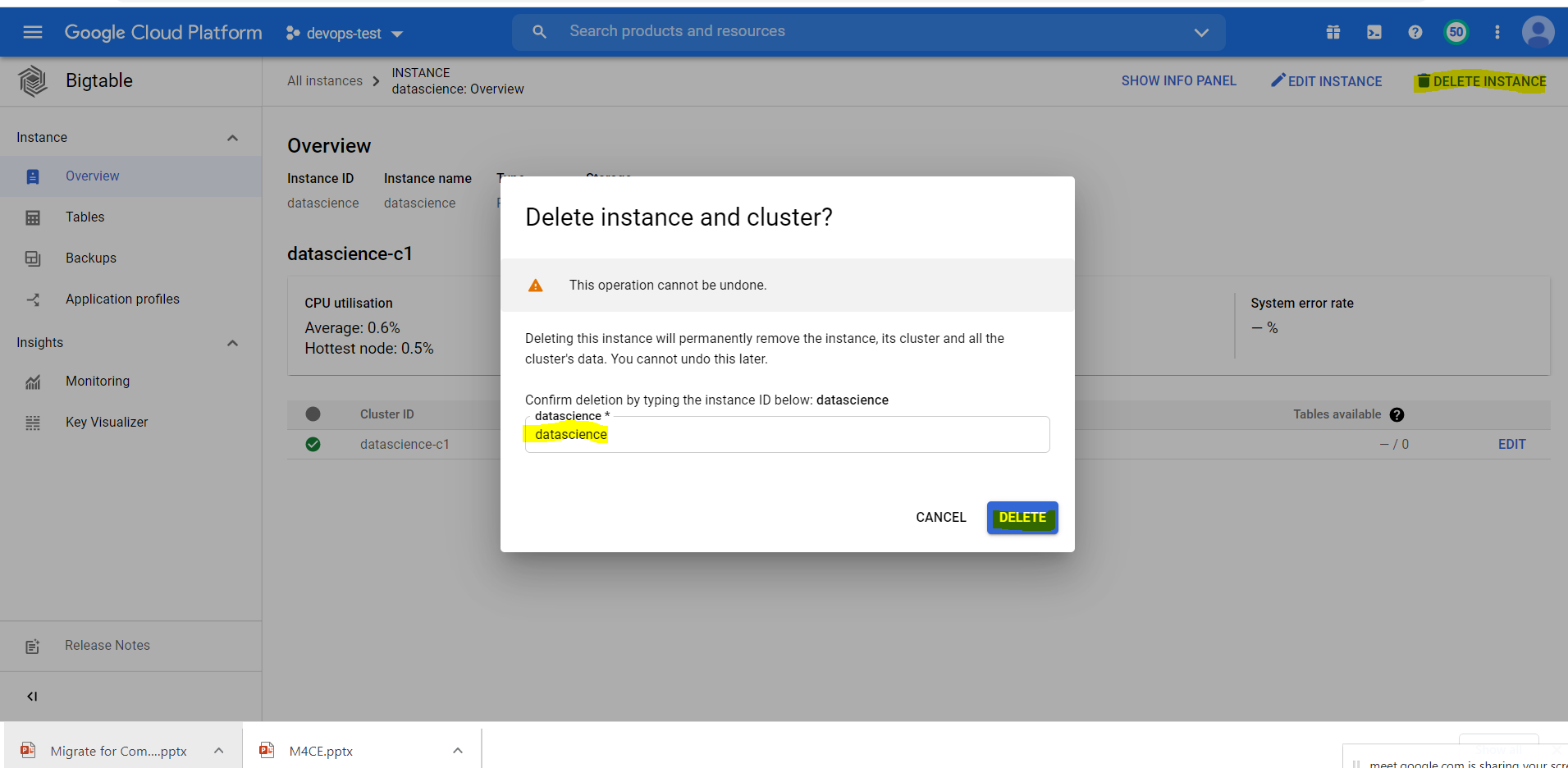
**How to create bigTable Instance from the GCP Console:**

1. **Open the BigTable and Create Instance page in the Cloud Console. ...**
2. **Enter a name for the instance. ...**
3. **Enter an instance ID. ...**
4. **Click Continue.**
5. **Choose whether to use an SSD or HDD disk for your clusters. ...**
6. **Click Continue.**
7. **Enter a cluster ID for the first cluster. ...**
8. **Choose the region and zone where the first cluster will run.**

****

**Delete BigTable Instance:**

1. **Open the list of Bigtable instances in the Cloud Console. Open the instance list.**
2. **Click the instance you want to delete, then click Delete instance. A confirmation dialog appears.**
3. **Follow the instructions in the confirmation dialog, then click Delete. The instance is permanently deleted.**

****

**DevOps role on BigTable:**

**1.BigTable instance creation as per request from Big Data Development / DataScience team (they will provide bigtable instance details like nodes/region/bigtable name ) per client naming convention.**

**2.Increase/Decrease BigTable instance Size during the load testing**

**3.Delete BigTable instance after their Testing**

**4. Provide access to BigTable instances based on the requirement from Dev/QA Teams.**

**Hi DevOps Team,**

**We are going to perform load test on test-gmail bigtable instance (devops-test project) ,please increase node count to 5.**

**Will inform you once we have done the testing and after that you can reduce the node count to 2.**

**We have increased the instance node count to 5 and please check now.**

****

**Please decrease node count to 1.**

**GCR (Google container Registry ):**

When working with Docker images, you’ll often be pushing and pulling them from a container registry, used for storage and quick centralized access

The most popular registry is the [official Docker Hub](https://hub.docker.com/), which is public, but many cloud providers offer private container registries. Google’s Container Registry is one of these, providing your own registry for your GCP account.

What is a container registry in Google cloud?

**Container Registry** is a single place for your team to manage **Docker** images.

**How to Tag a Docker Image to Docker Hub Repo:**

root@docker-ubuntu:~# **docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

kirankumar77/jboss-counterwebapp latest d892d81ad69b 5 weeks ago 730MB

kirankumar77/busybox latest 69593048aa3a 7 weeks ago 1.24MB

busybox latest 69593048aa3a 7 weeks ago 1.24MB

centos latest 300e315adb2f 7 months ago 209MB

jboss/base-jdk 11 a240ec4882cb 10 months ago 471MB

**Tag Image to Docker Gub Repo:**

root@docker-ubuntu:~# docker tag centos:latest kirankumar77/centos:latest

**Push Tagged Docker Image to Docker Hub:**

**docker push kirankumar77/centos:latest** (Will push this image to DockerHub)

We can see tagged image in Docker hub.

root@docker-ubuntu:~# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

kirankumar77/jboss-counterwebapp latest d892d81ad69b 5 weeks ago 730MB

kirankumar77/busybox latest 69593048aa3a 7 weeks ago 1.24MB

busybox latest 69593048aa3a 7 weeks ago 1.24MB

kirankumar77/centos latest 300e315adb2f 7 months ago 209MB

centos latest 300e315adb2f 7 months ago 209MB

jboss/base-jdk 11 a240ec4882cb 10 months ago 471MB

**Above steps are for DOCKER HUB and we can do the same steps for GCR as well.**

**For that we need to add gcr.io as a prefix for docker commands.**

**GCR Steps:**

You’ll need your project ID for the first step; this is visible from the “Select Project” dropdown in the GCP console. Copy the project ID as shown below.



Then, you can tag the image as normal using docker tag, only supplying a custom hostname for GCR. By default, gcr.io stores images in a cloud storage bucket located in the U.S. You can also use eu.gcr.io and asia.gcr.io for those regions.

Before pushing image to Google container registry, you must add the registry name and image name as a tag to the image.

**us.gcr.io hosts your images in the United States.**

**eu.gcr.io hosts your images in the European Union.**

**asia.gcr.io hosts your images in Asia.**

## Add the tag to your image:

*docker tag <user-name>/<sample-image-name> gcr.io/<project-id>/<sample-image-name>:<tag>*

This tags the image as latest, the default for new deployments, but if you want to use another tag, you can append :tag to the image name.

Once it’s tagged, you can upload it to GCR with docker push:

*gcloud docker — push gcr.io/your-project-id/<project-id>/<sample-image-name>:<tag>*

Once uploaded, they’ll be visible in the [Container Registry console](https://console.cloud.google.com/gcr/images/), or by running gcloud container images list-tags, and you’ll be able to use them for your Cloud Run and Kubernetes deployments.

If you want to manually pull the image down, you can do so with docker pull:

docker pull gcr.io/[PROJECT-ID]/[IMAGE]:[TAG]

Docker images Pull command:

docker images

docker pull kirankumar77/jenkins-maven

Tag Docker Image with GCR:

docker tag kirankumar77/jenkins-maven gcr.io/devops-test-316311/kirankumar77/jenkins-maven:latest

Push Tagged Image to GCR:

docker push gcr.io/devops-test-316311/kirankumar77/jenkins-maven:latest

If you want to push busybox image to GCR :

docker pull busybox:latest

docker tag busybox:latest gcr.io/devops-test-316311/busybox

docker push gcr.io/devops-test-316311/busybox

GCR Image Pull:

docker pull gcr.io/devops-test-316311/kirankumar77/jenkins-maven:latest

You can use the command gcloud container images list command to accomplish this task.

You can use gcloud container images [describe](https://cloud.google.com/sdk/gcloud/reference/container/images/describe) command to see the size of the images

gcloud container images describe gcr.io/myproject/myimage:tag

**How to Pull Docker Image:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker pull kirankumar77/centos:latest**

**latest: Pulling from kirankumar77/centos**

**7a0437f04f83: Pull complete**

**Digest: sha256:dbbacecc49b088458781c16f3775f2a2ec7521079034a7ba499c8b0bb7f86875**

**Status: Downloaded newer image for kirankumar77/centos:latest**

**docker.io/kirankumar77/centos:latest**

**After Docker Pull, we can see pulled Docker Image as shown below**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker images**

**REPOSITORY TAG IMAGE ID CREATED SIZE**

**kirankumar77/centos latest 300e315adb2f 7 months ago 209MB**

**How to Tag Docker Image to GCR Repo:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker tag kirankumar77/centos:latest gcr.io/devops-test-316311/kirankumar77/centos:latest**

**How to Push TAGGED Docker Image to GCR:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker push gcr.io/devops-test-316311/kirankumar77/centos:latest**

**The push refers to repository [gcr.io/devops-test-316311/kirankumar77/centos]**

**2653d992f4ef: Layer already exists**

**latest: digest: sha256:dbbacecc49b088458781c16f3775f2a2ec7521079034a7ba499c8b0bb7f86875 size: 529**

**Once push that Docker image to GCR, We can see that image from the GCR.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker images**

**REPOSITORY TAG IMAGE ID CREATED SIZE**

**kirankumar77/centos latest 300e315adb2f 7 months ago 209MB**

**gcr.io/devops-test-316311/kirankumar77/centos latest 300e315adb2f 7 months ago 209MB**

How much does GCR cost?

Google Container Registry (**GCR**) Account Plan

Storage **is** $0.026 per GB-month.

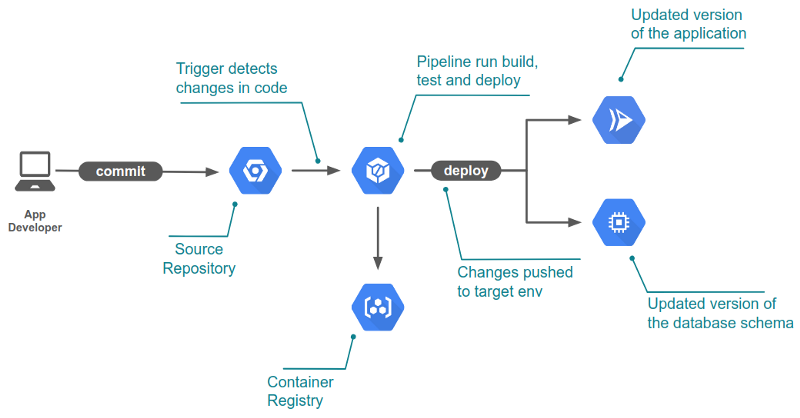
Features:

#### Secure, private Docker registry

#### Native Docker support

#### Fast, high-availability access

**CICD Pipeline flow:**

****

**Git branch watsap\_1.0**

**Git checkout watsap\_1.0**

**Watsap 1.0 version(java) developed by DEV TEAM**

**Once development is done , I need to push that code to Github repo**

**Jenkins will start the build and compile java code with the help of maven -Output is Artifact (watsp\_1.0.war or jar or ear )**

**Jenkins will push watsp\_1.0.war artifact to Nexus (artifact repo)**

**Dockerfile (COPY watsp\_1.0.war /opt/webapps/watsp\_1.0.war)**

**Jenkins will create Docekrimage (watsp\_1.0) and push that to GCR (gcr.io/devops-dev/watsap/watsap\_1.0)**

**Jenkins will pull watsap\_1.0 and create containers from that image and deploy those containers to DEV/QA/PREPROD/PROD Environments.**

**Once Deploy containers to Production , you can see watsap update from Google playstore.**

**Watsap update - > download latest watsap**

**STACKDRIVER:**

Stackdriver monitoring provides you the ability to monitor your applications deployed on GCP. Stackdriver also allows you to monitor your applications deployed in AWS as well. You can monitor your applications for performance and uptime, and all these metrics and events are collected and stored by Stackdriver. Stackdriver monitoring also generates dashboards for easy visual consumption of the data.

[**https://devopspoints.com/google-cloud-platform-stackdriver-monitoring-and-logging.html**](https://devopspoints.com/google-cloud-platform-stackdriver-monitoring-and-logging.html)

****

**STACKDRIVER MONITORING:**

**Monitoring collects metrics, events, and metadata from Google Cloud Platformhosted uptime probes, application instrumentation, and a variety of common application components including Cassandra, Nginx, Apache Web Server, Elasticsearch, and many others. Stackdriver ingests that data and generates insights via dashboards, charts, and alerts.**

**Metrics help you understand how your applications and system services are performing. Stackdriver defines over a thousand metric types that help you monitor GCP, AWS, and third-party software. You can also create your own custom metrics.**

**StackDriver Logging:**

**Stackdriver Logging allows you to store, search, analyze, monitor, and alert on log data and events from Google Cloud Platform and AWS.**

**It includes storage for logs, a user interface called the Logs Viewer, and an API to manage logs programmatically. Logging lets you read and write log entries, search and filter your logs, export your logs, and create logs-based metrics.**

**StackDriver Agent:**

**To monitor VMs, you can use the Stackdriver Monitoring agent which is a collectd-based daemon that collects system and application metrics from virtual machine instances. It then sends these metrics to Monitoring. The Monitoring agent by default collects metrics information on disk, CPU, network and processes.**

**StackDriver Alerting:**

Stackdriver also provides a built-in alerting system, so you can configure alerts on the GCE metrics mentioned above. The free service tier allows you to set up basic email alerts on GCP metrics that cross static thresholds or that are missing data.

gcloud projects create stackdriver-test-123 *--name stackdriver-test-project*

*gcloud projects list*

Project IDs and project numbers are unique. Your IDs and numbers will differ from what you see here.

Now that we have the project created, let’s enable billing for it. Before we do that, let’s make sure that we set our project to stackdriver-test-project:

$ gcloud config list

[core]

account = hidden@gmail.com

disable\_usage\_reporting = True

project = kubecluster-206421

$ gcloud config unset project

Unset property [core/project].

$ gcloud config list

[core]

account = hidden@gmail.com

disable\_usage\_reporting = True

Your active configuration is: [abcd]

$ gcloud config set project stackdriver-test-123

Updated property [core/project].

$ gcloud config list

[core]

account = hidden@gmail.com

disable\_usage\_reporting = True

project = stackdriver-test-123

Your active configuration is: [abcd]

Yum install apache2 php7

curl -sSO https://dl.google.com/cloudagents/add-logging-agent-repo.sh

sudo bash add-logging-agent-repo.sh --also-install

curl -sSO https://dl.google.com/cloudagents/add-monitoring-agent-repo.sh

sudo bash add-monitoring-agent-repo.sh --also-install

How to create the alerts for CPU Utilization:

Goto GCP Console -> Monitoring -> alerting -> create policy

Here’s an example alert for extended periods of prolonged CPU usage:

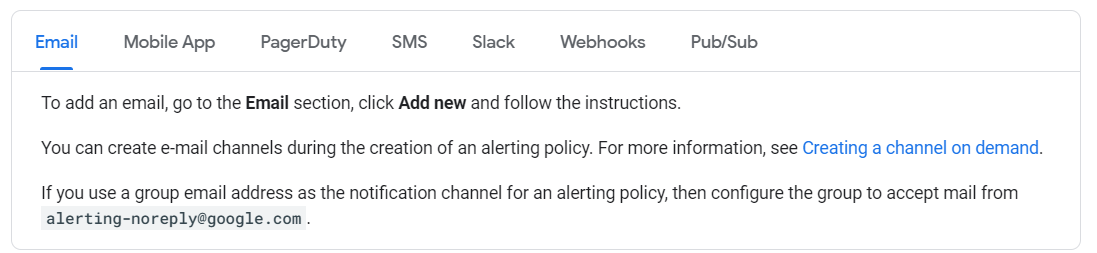


**After this, configure notifications for those alerts**

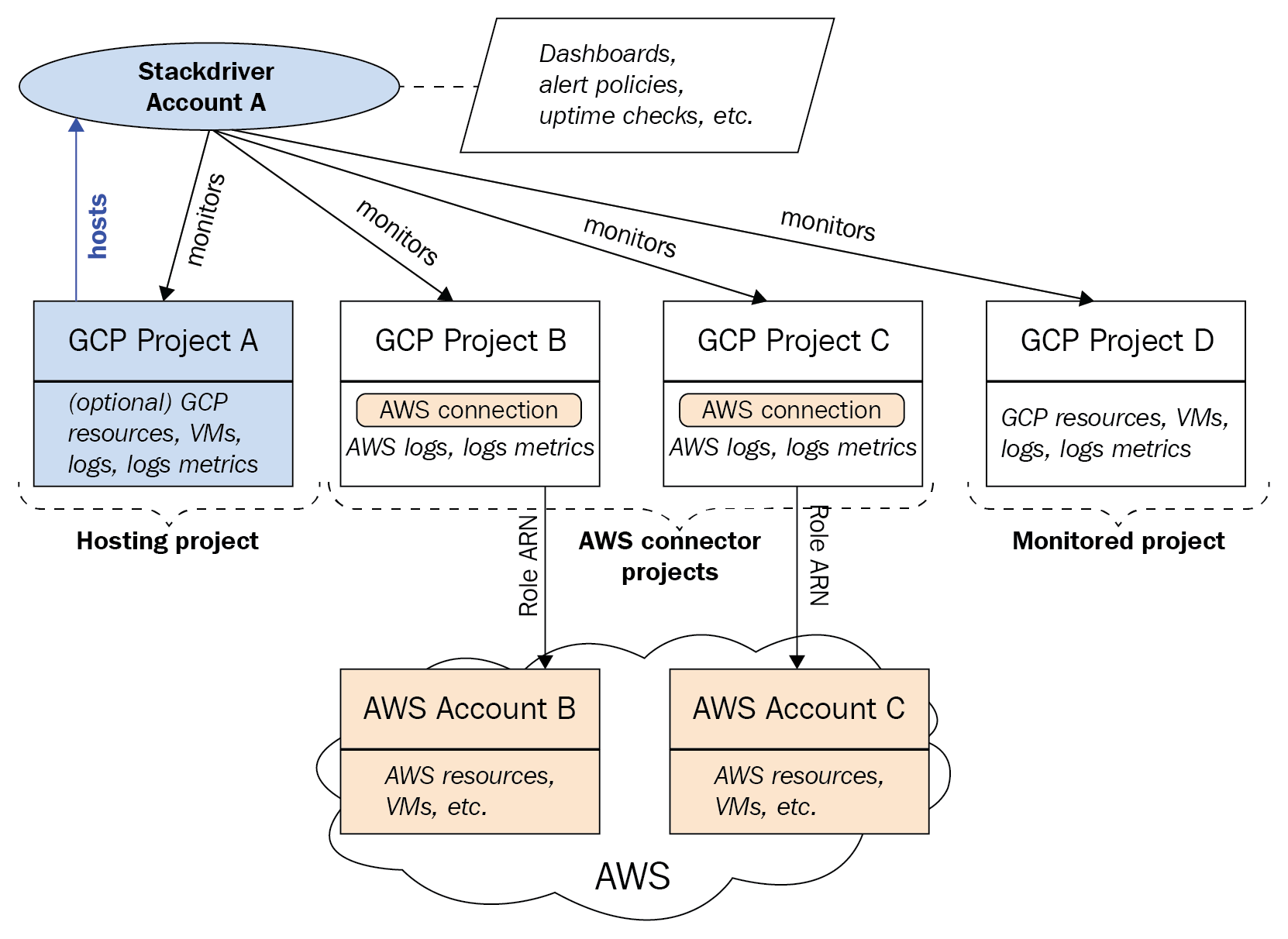
**Click Edit notification channels.**

**To add a new notification channel, locate the channel type,**

**click** Add new**, and then follow the channel-specific instructions contained in the following table:**

****

**Select Email and give your team Distribution list DL ( like devops@flipkart.com so it will notify your entire team) email for alerts notifications.**

****

**Where are Stackdriver logs stored?**

**All logs generated in the project are stored in the \_Required and \_Default logs buckets, which live in the project that the logs are generated in: \_Required: This bucket holds Admin Activity audit logs, System Event audit logs, and Access Transparency logs, and retains them for 400 days**

**Cloud Logging retains the logs in \_Required bucket for 400 days; you can't change this retention period.**

**Cloud Logging automatically routes the following types of logs to the \_Required bucket:**

**Admin Activity audit logs**

**System Event audit logs**

**Access Transparency logs**

**Logs held in the \_Default bucket are retained for 30 days, unless you** [**configure custom retention**](https://cloud.google.com/logging/docs/buckets#custom-retention) **for the bucket.**

**GKE : Google Kubernetes Engine :**

**Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The GKE environment consists of multiple machines (specifically, Compute Engine instances) grouped together to form a cluster.**

**Difference between Kubernetes and GKE:**

**In short, Kubernetes does the orchestration, the rest are services that would run on top of Kubernetes. GKE brings you all these components out-of-the-box, and you don't have to maintain them. They're set up for you, and they're more 'integrated' with the Google portal.**

**POD:**

**Pods are the smallest, most basic deployable objects in Kubernetes. A Pod represents a single instance of a running process in your cluster. Pods contain one or more containers, such as Docker containers. When a Pod runs multiple containers, the containers are managed as a single entity and share the Pod's resources.**

**Difference between POD and Container:**

**A container runs logically in a pod (though it also uses a container runtime); A group of pods, related or unrelated, run on a cluster. A pod is a unit of replication on a cluster; A cluster can contain many pods, related or unrelated [and] grouped under the tight logical borders called namespaces.**

**What is the difference between node and pod in Kubernetes?**

**A Pod always runs on a Node. A Node is a worker machine in Kubernetes and may be either a virtual or a physical machine, depending on the cluster. Each Node is managed by the Master. A Node can have multiple pods, and the Kubernetes master automatically handles scheduling the pods across the Nodes in the cluster.**

**How many containers can run in a pod?**

**No more than 110 pods per node. No more than 5000 nodes. No more than 150000 total pods. No more than 300000 total containers.**

**POD Life cycle/ Does pods have a well defined lifecycle?**

**Pods follow a defined lifecycle, starting in the Pending phase, moving through Running if at least one of its primary containers starts OK, and then through either the Succeeded or Failed phases depending on whether any container in the Pod terminated in failure.**

**POD Out of memory issue / Why do pods restart ?**

**This is one of the common reason of restarting container which happens the resource usage is not configured or application itself behaves unpredictable. If we have allocated 600Mi of memory for a container and it tries to allocate more than this limit, the pod will be killed with OOM.**

**Can you use Kubernetes without Docker?**

**Kubernetes can run without Docker and Docker can function without Kubernetes. But Kubernetes can (and does) benefit greatly from Docker and vice versa. Docker is a standalone software that can be installed on any computer to run containerized applications.**

**What is difference between deployment and StatefulSet?**

**A StatefulSet is another Kubernetes controller that manages pods just like Deployments. But it differs from a Deployment in that it is more suited for stateful apps. A stateful application requires pods with a unique identity (for example, hostname). One pod should be able to reach other pods with well-defined names.**

**Which is minimum recommended number of nodes to have in your Gke cluster?**

**GKE offers a range of options for your cluster's nodes. For example, you can create one or more node pools; node pools are groups of nodes within your cluster that share a common configuration. Your cluster must have at least one node pool, and a node pool called default is created when you create the cluster.**

**What is stateful vs stateless?**

**The key difference between stateful and stateless microservices is that stateless microservices don't store data on the host, whereas stateful microservices require some kind of storage on the host who serves the requests. Keeping the state is critical for a stateful service.**

**What if Kubernetes master goes down?**

**So, if that Master node goes down in Kubernetes, the outcome depends on the architecture you have. That is, Single Master or Multi-Master architecture. If you have a Single-Master architecture, then it becomes a problem, if the master node goes down, and then you cannot create more services and pods, etc.**

**How do I destroy the Kubernetes cluster?**

**Deleting a cluster**

1. **Visit the Google Kubernetes Engine menu in Cloud Console. Visit the Google Kubernetes Engine menu.**
2. **Next to the cluster you want to delete, click more\_vert Actions, then click delete Delete.**
3. **When prompted to confirm, click Delete again.**

**What is Namespace ?**

**Namespaces are Kubernetes objects which partition a single Kubernetes cluster into multiple virtual clusters**

**To create a namespace, use kubectl create command.**

**kubectl create namespace <namespace name>**

**How do I get all pods in namespaces?**

**List all Container images in all namespaces**

**kubectl get pods --all-namespaces**

**Why does Kubernetes have 3 master nodes?**

**Kubernetes Cluster Management. A highly available cluster is composed of at least 3 master nodes, each running a member of the etcd distributed database and all the Kubernetes master components (API, controller manager and scheduler).**

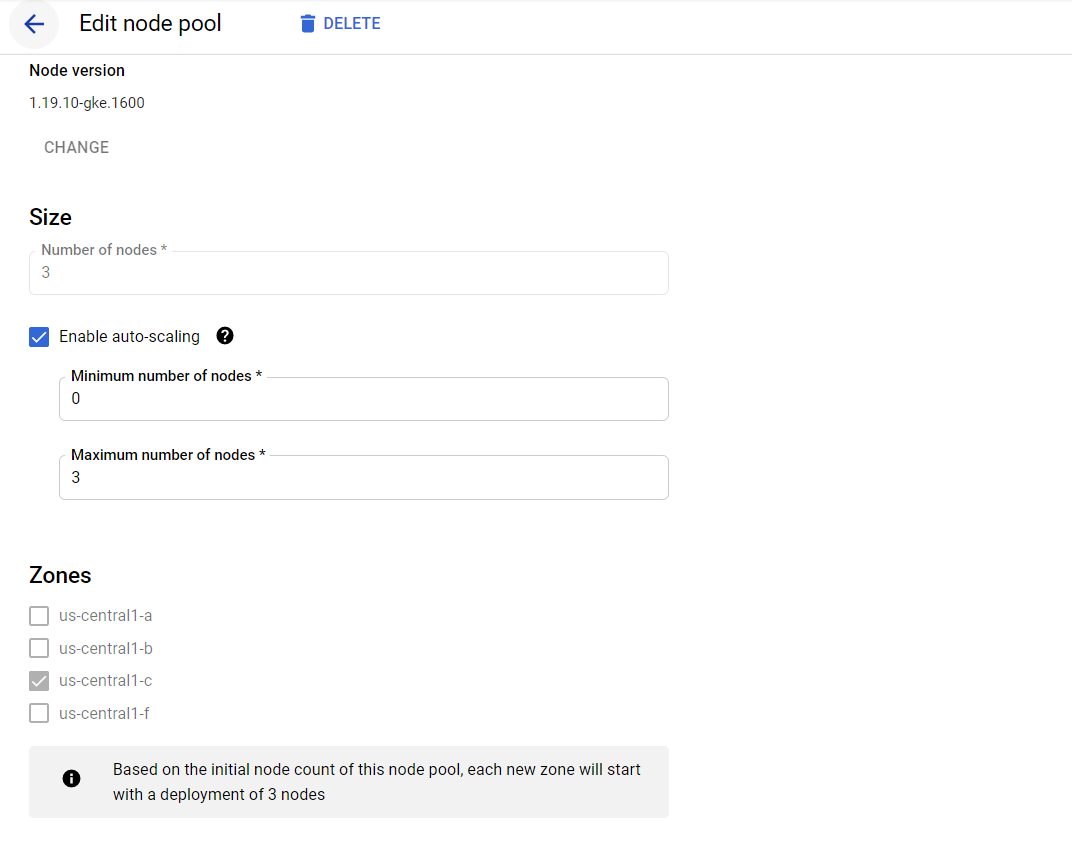
**Kubeconfig file:**

**A kubeconfig file is a file used to configure access to Kubernetes when used in conjunction with the kubectl command line tool (or other clients).**

**Autoscaling in Kubernetes:**

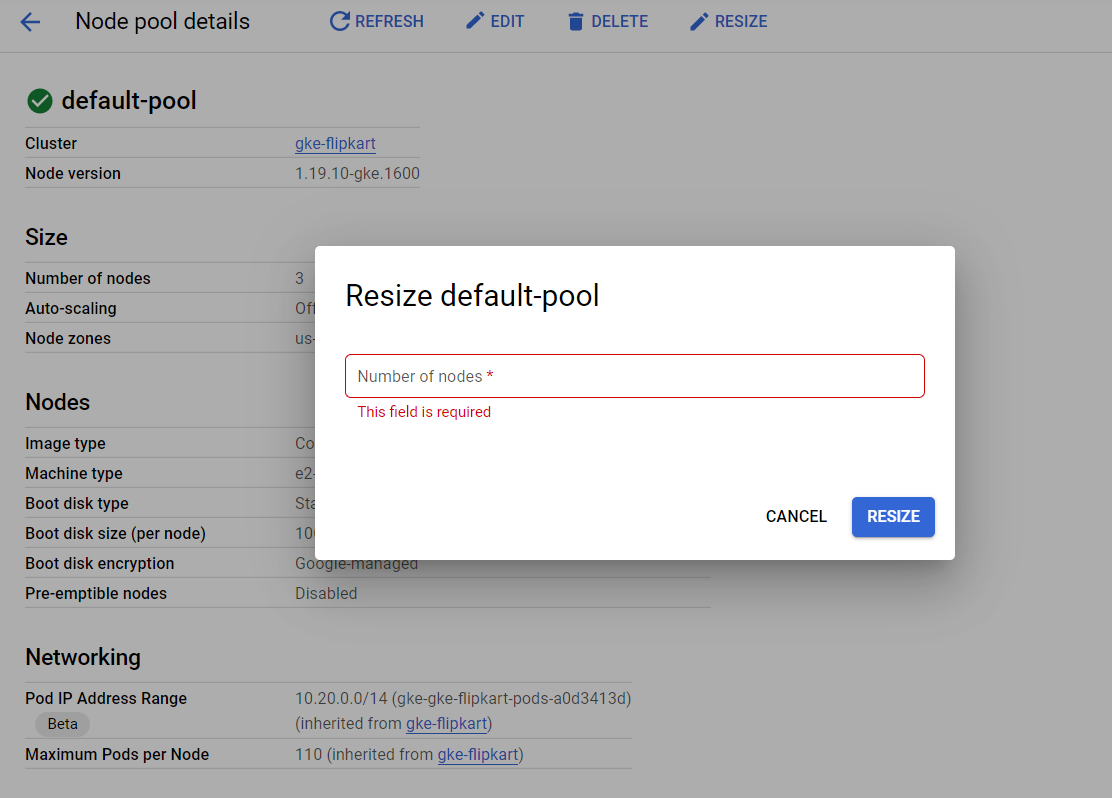
**Select GKE Cluster - Nodes - Nodepool - Edit**

**Here choose Min and Max nodes based on the Dev/QA Teams Requirement**

****

**How to Increase Node Pool size or GKE Cluster Size:**

**Select GKE Cluster - Nodes - Nodepool - Edit - ReSize - Give node count which your customer wants -> Select ReSzie - >will take sometime to apply changes**

****

**Once you are done with GKE Cluster , We can connect**

**You can connect to your cluster via command-line or using a dashboard:**

**Configure** [**kubectl**](http://kubernetes.io/docs/user-guide/kubectl-overview/)  **command line access by running the following command:**

**gcloud container clusters get-credentials gke-flipkart --zone us-central1-c --project devops-test-316311**

**Fetching cluster endpoint and auth data.**

**kubeconfig entry generated for gke-flipkart.**

**kubectl get nodes**

**NAME STATUS ROLES AGE VERSION**

**gke-gke-flipkart-default-pool-9e8c6735-3cg3 Ready <none> 27m v1.19.10-gke.1600**

**gke-gke-flipkart-default-pool-9e8c6735-546v Ready <none> 27m v1.19.10-gke.1600**

**gke-gke-flipkart-default-pool-9e8c6735-t9f6 Ready <none> 27m v1.19.10-gke.1600**

**kubectl cluster-info (It will give GKE Cluster information )**

**Kubernetes control plane is running at https://35.223.92.224**

**GLBCDefaultBackend is running at https://35.223.92.224/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy**

**KubeDNS is running at https://35.223.92.224/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy**

**Metrics-server is running at https://35.223.92.224/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy**

**How to Push/Pull Docker Images to Google Container Registry:**

**If no images on your server , you can pull images from docker hub.**

**Docker search hello-app (it will search for hello-app images in dockerhub)**

**Docker pull chris7444/hello-app (pull image to your server)**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker images (after pull, we can see that images is on your server)**

**REPOSITORY TAG IMAGE ID CREATED SIZE**

**chris7444/hello-app latest c6fcf0661f33 3 months ago 11.8MB**

**Once we have docker image, we can tag that image with GCR Image as below.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker tag chris7444/hello-app:latest gcr.io/devops-test-316311/chris7444/hello-app:latest**

**Once Tagged , push same tagged image to GCR with below command**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ docker push gcr.io/devops-test-316311/chris7444/hello-app:latest**

**The push refers to repository [gcr.io/devops-test-316311/chris7444/hello-app]**

**2b44883ec3f7: Pushed**

**cb381a32b229: Layer already exists**

**latest: digest: sha256:e73b28615760325d9f922bc95db25edba181fb8f57dc4c5d5b874e6850fdb366 size: 739**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**hello-world-556449d948-gw9cp 0/1 ImagePullBackOff 0 13m**

**Once we pushed that image to GCR , That image is available for GKE Cluster Deployments.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl create deployment hello-world --image=gcr.io/${DEVSHELL\_PROJECT\_ID}/chris7444/hello-app:latest**

**deployment.apps/hello-world created ( deployment has been created )**

**({DEVSHELL\_PROJECT\_ID} will be your Project ID and its Env variable )**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl get deployments**

**NAME READY UP-TO-DATE AVAILABLE AGE**

**hello-world 1/1 1 1 6s**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**hello-world-648d99d4cc-vrj2m 1/1 Running 0 16s**

**As of now we have only one POD , if you want to increase the pod count , we can increase replica count as shown below.**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl scale --replicas=5 deployment/hello-world**

**deployment.apps/hello-world**

**Scaled**

**Now pod count has been increased and can see them with below command**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**hello-world-648d99d4cc-5lppn 1/1 Running 0 21s**

**hello-world-648d99d4cc-jdlvq 1/1 Running 0 21s**

**hello-world-648d99d4cc-vmhlr 1/1 Running 0 21s**

**hello-world-648d99d4cc-vrj2m 1/1 Running 0 2m18s**

**hello-world-648d99d4cc-xgckk 1/1 Running 0 21s**

**Since all the pods are “Running,” we know that Kubernetes has distributed the pods to the nodes. Currently, we have five pods running over the three nodes.**

**It’s also possible to check the Kubernetes metrics in GKE with the kubectl top command. Let’s use it to look at the usage of the pods:**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl top pods**

**NAME CPU(cores) MEMORY(bytes)**

**hello-world-648d99d4cc-5lppn 0m 1Mi**

**hello-world-648d99d4cc-jdlvq 0m 1Mi**

**hello-world-648d99d4cc-vmhlr 0m 1Mi**

**hello-world-648d99d4cc-vrj2m 0m 1Mi**

**hello-world-648d99d4cc-xgckk 0m 1Mi**

**Similarly, you can use the kubectl top nodes command to retrieve aggregate data about the nodes/**

**kirankumarrv0990@cloudshell:~ (devops-test-316311)$ kubectl top nodes**

**NAME CPU(cores) CPU% MEMORY(bytes) MEMORY%**

**gke-gke-flipkart-default-pool-9e8c6735-3cg3 48m 5% 503Mi 17%**

**gke-gke-flipkart-default-pool-9e8c6735-546v 120m 12% 537Mi 19%**

**gke-gke-flipkart-default-pool-9e8c6735-t9f6 80m 8% 536Mi 19%**

**Service & Ingress in GKE:**

Now, let’s open our application up to the internet and receive some Hello World responses.

Containers to the Internet:

In addition to container management, Kubernetes provides resources to connect to applications from inside and outside the cluster. With the following command, you expose the deployment to the internet:

**kubectl expose deployment hello-world --type=LoadBalancer --port 80 --target-port 8080**

**This command creates a service resource in Kubernetes, and it provides networking with an IP attached to the application instances.**

**It can be checked from the menu item Kubernetes Engine > Services & Ingress:**

****

**Select Endpoints from the above screenshot , and you will get Application page.**

**The Service details page lists the networking configuration from the point of view of a Kubernetes cluster.**

**In addition, there is an external IP assigned to the service enabling access from the internet.It’s created by GCP TCP Load Balancer by default for zonal and regional clusters.**

**Let’s check the TCP load balancer in Network services > Load Balancer by clicking the load balancer in the previous view:**

**In this screenshot, load balancer instances for all three nodes are listed along with their health status. If you create a multi-region cluster, you will need an ingress controller and global load balancer deployed to your cluster for routing.**

**Check for the external IP with the following command:**

**kubectl get service**

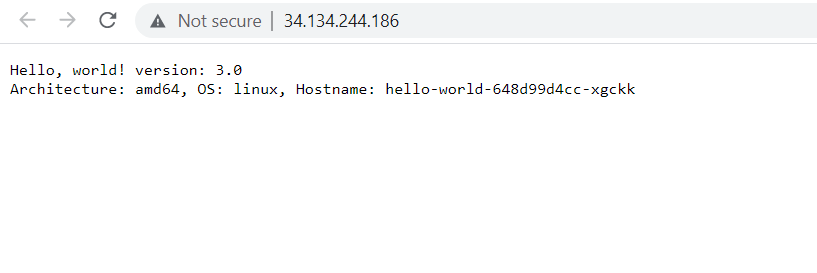
**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**hello-world LoadBalancer 10.0.2.152 35.232.168.243 80:30497/TCP 77s**

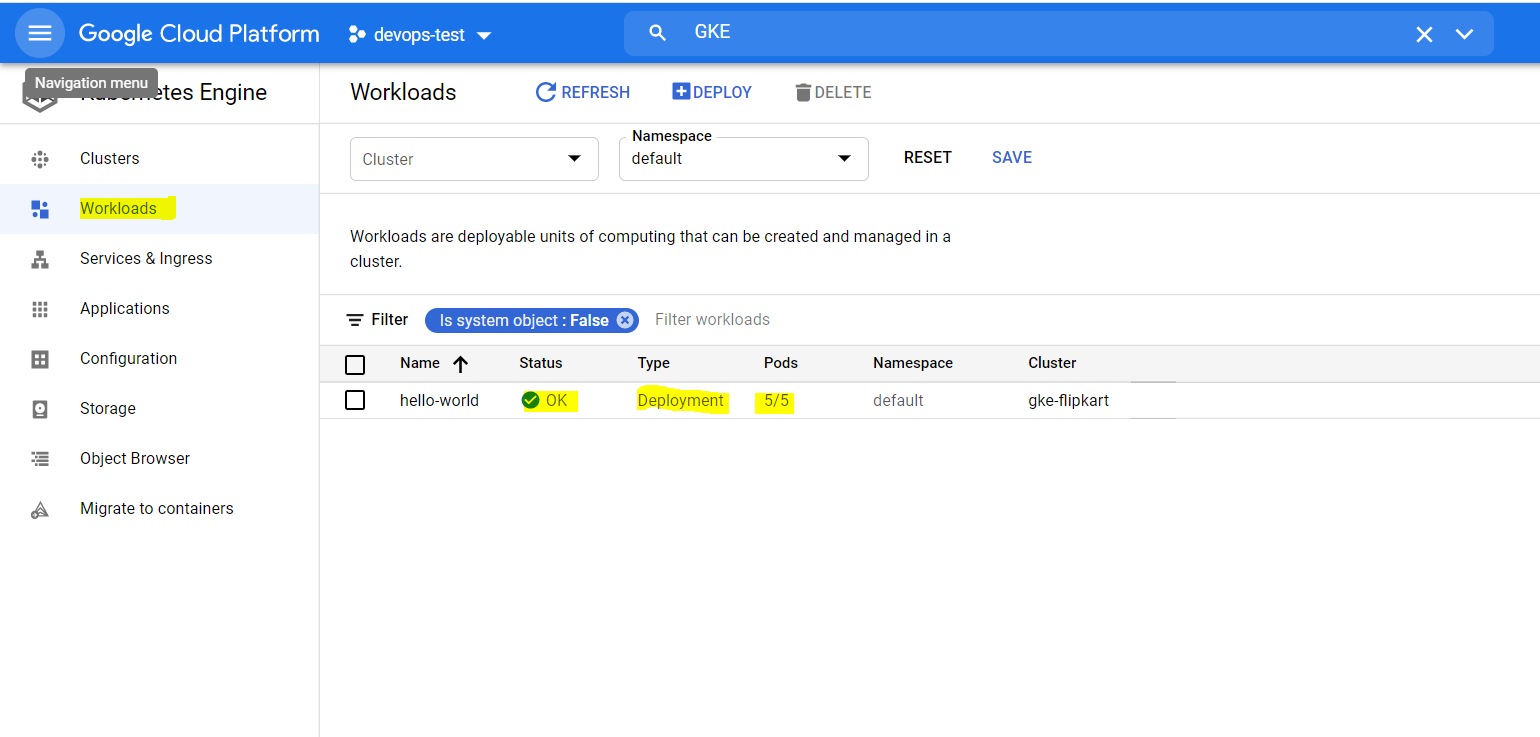
**Now, open the external IP listed above for hello-world in your browser.In the output, the hostname indicates the name of the pod.**

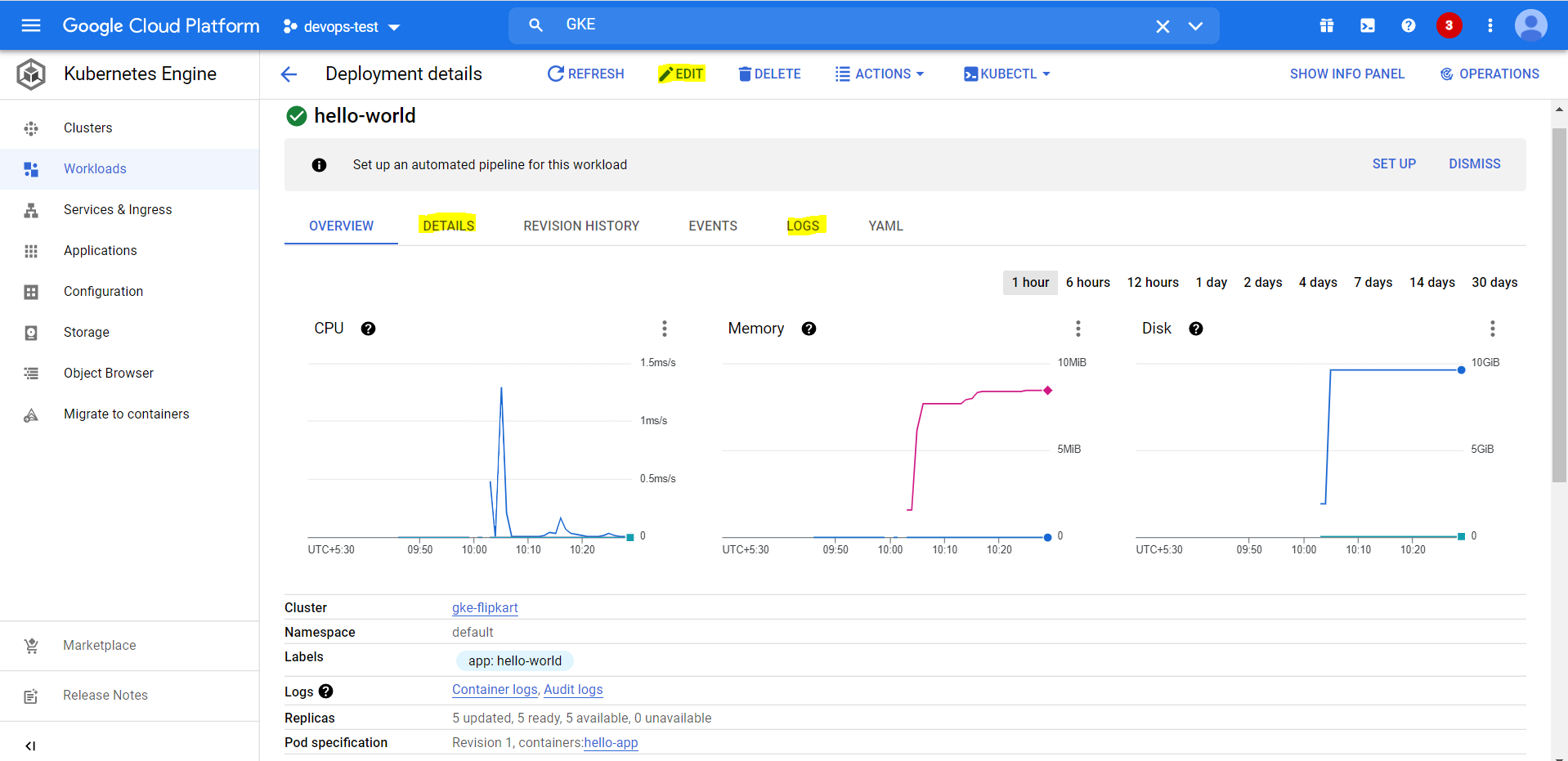
**You can see all of your pod names as hostnames if you reload the browser tab a couple of times.**

**You can expect a change of hostnames with each reload, since we have created a LoadBalancer type of service to expose the application.**

****

**Workloads/ Deployments:**

****

****

**How to Delete GKE Cluster from CLI:**

**gcloud container clusters delete test-gke --zone us-central1-a**

**The following clusters will be deleted.**

**- [test-gke] in [us-central1-a]**

**Do you want to continue (Y/n)? Y**

**Deleting cluster test-gke...done.**

**Deleted [https://container.googleapis.com/v1/projects/devops-test-316311/zones/us-central1-a/clusters/test-gke].**

**How to Delete GKE Cluster from UI :**

**Deleting a cluster**

1. **Visit the Google Kubernetes Engine menu in Cloud Console. Visit the Google Kubernetes Engine menu.**
2. **Next to the cluster you want to delete, click more\_vert Actions, then click delete Delete.**
3. **When prompted to confirm, click Delete again.**

**CONCLUSION:**

**We created a Docker container image which was pushed to the registry for future use in scaling.**

**Then, we created a Kubernetes cluster in GKE and deployed our application into it.**

**We scaled the app with many replicas and checked its status. We reviewed the metrics and logs with potential extensions.**

**Finally, we exposed our application to the internet. With this hands-on knowledge, you should now be able to package, deploy, and**

**manage containerized applications inside a Kubernetes cluster in GKE.**

**What is the difference between liveness and readiness probe in Kubernetes?**

**Kubernetes uses liveness probes to know when to restart a container. Kubernetes uses readiness probes to decide when the container is available for accepting traffic. ... The readiness probe is used to control which pods are used as the backends for a service.**

**ConfigMap:**

**A ConfigMap is an API object that lets you store configuration for other objects to use. Unlike most Kubernetes objects that have a spec , a ConfigMap has data and binaryData fields. These fields accept key-value pairs as their values. ... The keys stored in data must not overlap with the keys in the binaryData field.**

**Secrets in K8S:**

**Secrets can be defined as Kubernetes objects used to store sensitive data such as user name and passwords with encryption.**

**There are multiple ways of creating secrets in Kubernetes.**

* **Creating from txt files.**
* **Creating from yaml file.**

### Creating From Text File

**In order to create secrets from a text file such as user name and password, we first need to store them in a txt file and use the following command.**

**kubectl create secret generic tomcat-passwd –-from-file = ./username.txt –fromfile = ./password.txt**

**Here tomcat-passwd is the secret name and it will have username and password.**

### Creating Secrets from Yaml File:

apiVersion: v1

kind: Secret

Namespace: kittu

metadata:

name: tomcat-pass

type: Opaque

data:

password: <User Password>

username: <User Name>

### Creating the Secret from YAML File:

**$ kubectl create –f Secret.yaml**

**secrets/tomcat-pass**

**Get the Secrets with below command:**

**Kubectl get secrets ( list our the secrets which are in default namespace )**

**If you want to get the secrets for kittu namespace**

**Kubectl get secrets -n kittu ( will give secrets list which are there in kittu namespace )**

**What is the difference between ConfigMap and secret?**

**The major difference is, Secrets store data in base64 format meanwhile ConfigMaps store data in a plain text. If you have some critical data like, keys, passwords, service accounts credentials, db connection string, etc then you should always go for Secrets rather than Configs**

**[root@k8s-master PV]# kubectl create -f pv.yaml**

**persistentvolume/pv-gce created**

**[root@k8s-master PV]# kubectl get pv**

**NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE**

**pv-gce 10Gi RWO Retain Available slow 19s**

**[root@k8s-master PV]# kubectl create -f pvc.yaml**

**persistentvolumeclaim/my-disk-claim created**

**[root@k8s-master PV]# kubectl get pvc**

**NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE**

**my-disk-claim Bound pv-gce 10Gi RWO slow 7s**

**[root@k8s-master PV]# kubectl create -f pvolume.yaml**

**persistentvolume/task-pv-volume created**

**[root@k8s-master PV]# kubectl create -f pvclaim.yaml**

**persistentvolumeclaim/task-pv-claim created**

**[root@k8s-master PV]# kubectl create -f pv-pod.yaml**

**pod/task-pv-pod created**

**[root@k8s-master PV]# kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**task-pv-pod 1/1 Running 0 6s**

**[root@k8s-master PV]# kubectl describe pv task-pv-volume**

**Name: task-pv-volume**

**Labels: type=local**

**Annotations: pv.kubernetes.io/bound-by-controller: yes**

**Finalizers: [kubernetes.io/pv-protection]**

**StorageClass: manual**

**Status: Bound**

**Claim: default/task-pv-claim**

**Reclaim Policy: Retain**

**Access Modes: RWO**

**VolumeMode: Filesystem**

**Capacity: 10Gi**

**Node Affinity: <none>**

**Message:**

**Source:**

**Type: HostPath (bare host directory volume)**

**Path: /mnt/data**

**HostPathType:**

**Events: <none>**

**[root@k8s-master PV]# kubectl get pv**

**NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE**

**task-pv-volume 10Gi RWO Retain Bound default/task-pv-claim manual 114s**

**[root@k8s-master PV]# kubectl get pvc**

**NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE**

**task-pv-claim Bound task-pv-volume 10Gi RWO manual 103s**

**[root@k8s-master PV]# kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**task-pv-pod 1/1 Running 0 2m5s**

**[root@k8s-master PV]# kubectl exec -it task-pv-pod /bin/bash**

**kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.**

**root@task-pv-pod:/# whoami**

**root**

**root@task-pv-pod:/# hostname**

**task-pv-pod**

**root@task-pv-pod:/# exit**

[**https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/**](https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/)

## Difference between Google Cloud Platform, AWS and Azure:

**Like Google cloud platform, AWS and Azure are the other popular cloud-based platforms. However, there are differences amongst them. Some of the main differences between GCP,** [**AWS**](https://www.javatpoint.com/aws-tutorial) **and** [**Azure**](https://www.javatpoint.com/microsoft-azure) **are tabulated below:**

| **Google Cloud** | **AWS** | **Azure** |
| --- | --- | --- |
| **It uses GCE (Google Compute Engine) for computing purposes.** | **AWS EC2 offers core compute services.** | **It uses virtual machines for computation purposes.** |
| **It uses Google Cloud Storage for storage purposes.** | **It uses Amazon S3 for storing the data.** | **It uses a storage block bob that comprises blocks for storing the data.** |
| **It offers the lowest price to the customers to beat other cloud providers.** | **AWS pricing is generally keen to have inscrutable. The overall structure of granular pricing is a bit complex.** | **Like AWS, Azure pricing structure is also difficult to understand unless you have considerable experience.** |
| **It uses Cloud Test labs for App Testing purposes.** | **It uses a device farm for App Testing purposes.** | **It uses DevTest labs for App Testing purposes.** |
| **It uses Subnet as a virtual network.** | **It uses VPC as a virtual network.** | **It uses VNet as a virtual Network.** |
| **It follows the Cloud Load Balancing configuration.** | **It follows the Elastic Load Balancing configuration.** | **It follows the Load-Balancer Application Gateway configuration.** |