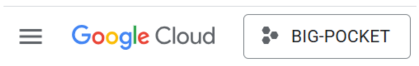


📌 *Overview*

*This documentation covers the creation and configuration of a Google Cloud project titled **"BIG-POCKET"** using the Google Cloud Console. It includes steps to manage **API credentials**, integrate **Firebase**, and work with **OAuth 2.0**, **API keys**, and **service accounts**. Additionally, it introduces Google Cloud fundamentals, key terminologies, use cases, and comparisons with peer cloud platforms.*

What is Google Cloud?

***Google Cloud Platform (GCP)** is a suite of cloud computing services provided by Google. It allows individuals and businesses to build, deploy, and scale applications, websites, and services on the same infrastructure that Google uses internally.*

🧠 *Key Concepts:*

- ***Compute:** Virtual Machines (Compute Engine), Kubernetes (GKE)*
- ***Storage:** Cloud Storage, Databases (Firestore, BigQuery, etc.)*
- ***AI/ML:** Vertex AI, AutoML, NLP APIs*
- ***Networking:** VPCs, Cloud Load Balancing*
- ***Security:** IAM, KMS, Cloud Armor*

🎯 *Why Use Google Cloud?*

- ***Scalability:** Scale resources automatically based on traffic*
- ***Security:** Enterprise-grade security and compliance*
- ***Global Infrastructure:** Reliable and fast services across the globe*
- ***Integrated Services:** Native integration with Firebase, BigQuery, etc.*
- ***Free Tier & Trial:** $300 credits to try services risk-free*

🔍 Peer Comparisons

Feature / Provider	**Google Cloud**	**AWS**	**Microsoft Azure**
Strength	AI/ML, Big Data, Firebase	Market Share, Services	Hybrid Cloud, Enterprise
Integrated with	Firebase, Android, YouTube	Alexa, Prime Video	Microsoft Office, GitHub
Market Adoption	Growing	Leading	Strong in enterprises
Machine Learning	Vertex AI, TPUs	SageMaker	Azure ML
Serverless	Cloud Functions	Lambda	Azure Functions
Developer Experience	Firebase Integration	Amplify (limited)	Visual Studio focused
Big Data & Analytics	BigQuery	Redshift	Synapse
Free Tier	$300/90 days + always-free	12-month free + tier	12-month free + tier

Firebase and GCP Integration:

🔑 Step-by-Step: Setting Up Firebase with Google Cloud Console

1. Project Creation

- Project: BIG-POCKET
- Firebase auto-integrates with Google Cloud Console when createdFirebase

2. Navigating to API & Services

- Go to console.cloud.google.com
- Access **"APIs and Services" → "Credentials"**

3. Credentials Setup

You created the following credentials:

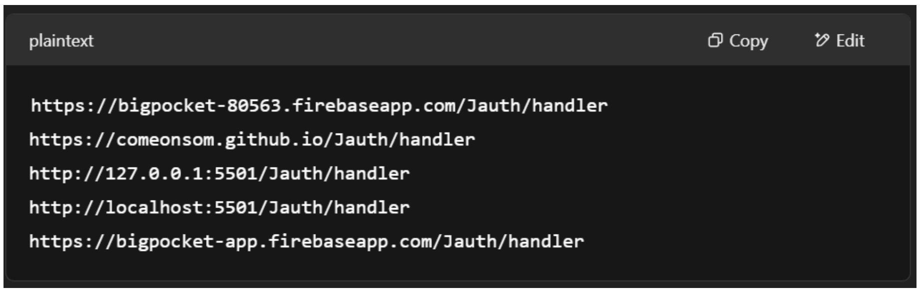
- **API Keys**
    - Example: AIzaSy… (Unrestricted — recommended to apply restrictions)
    - Used in frontend apps for accessing public APIs
- **OAuth 2.0 Client ID**
    - Type: Web application
    - Used for user authentication and authorization
    - Auto-created by Firebase or by manual setup
- **Service Account**
    - Example: firebase-adminsdk-fbsvc@big-pocket-80563.iam.gserviceaccount.com
    - Used for backend authentication and Firebase Admin SDK

4. OAuth Consent Screen

- Required for authenticating users via Google
- Set up branding, scopes, and redirect URIs

5. Authorized Redirect URIs (for OAuth Web Clients)

Includes:



```plaintext
https://bigpocket-80563.firebaseapp.com/Jauth/handler
https://comeonsom.github.io/Jauth/handler
http://127.0.0.1:5501/Jauth/handler
http://localhost:5501/Jauth/handler
https://bigpocket-app.firebaseapp.com/Jauth/handler
```

📘 Key Terminologies Explained

Term	Description
API Key	A simple encrypted string used to authenticate apps
OAuth 2.0	Protocol for user authorization
Service Account	Special account used by apps/services for backend communication
Firebase	Google's mobile platform with tools for building apps quickly
IAM	Identity and Access Management — controls who can do what
Redirect URI	The URL where users are redirected after authentication
Google Auth Platform	Interface for managing OAuth clients and settings
Client ID	Unique identifier for OAuth applications

🚀 What Can You Do With Google Cloud + Firebase?

- **Authentication:** Secure login with Google, Facebook, email/password
- **Real-time Database:** Sync data across users instantly
- **Hosting:** Deploy static and dynamic websites
- **Cloud Functions:** Run backend code in response to events
- **Cloud Firestore:** Scalable NoSQL database
- **Cloud Storage:** Store images, videos, and files securely
- **Analytics:** Monitor user engagement and performance

🧩 Security Best Practices

- Secure your credentials: Restrict API keys, rotate keys periodically, and store secrets in Secret Manager.
- Use IAM roles: Grant least privilege to service accounts and users.
- Monitor usage: Enable billing alerts and view quotas in the APIs Dashboard.
- Leverage SDKs: Use official client libraries (e.g., Node.js, Python) to simplify integration.
- Automate with CLI and Terraform: Manage resources declaratively using gcloud CLI or Infrastructure as Code tools.

What Is Terraform?

Terraform is a tool for automating infrastructure on cloud platforms like GCP using simple code. It:

- Is **declarative:** you declare *what* you want, Terraform figures out *how* to do it.
- Uses **HCL (HashiCorp Configuration Language)**
- Supports **GCP, AWS, Azure**, and more

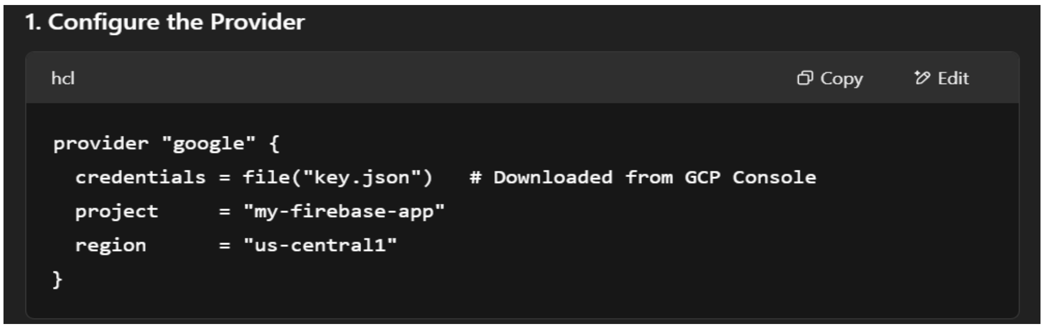
Why Use Terraform with GCP?

- Infrastructure as code: version your infrastructure like app code
- Reusability: write once, deploy many times
- Automation: no need to click through cloud consoles
- Team collaboration: share .tf files like source code

Terraform Workflow Overview

1. **Write Code:** Define resources in .tf files
2. **Initialize:** terraform init
3. **Plan:** See what Terraform will do with terraform plan
4. **Apply:** Run terraform apply to create resources
5. **Destroy:** Run terraform destroy to remove everything

Sample Terraform Configuration for GCP + Firebase

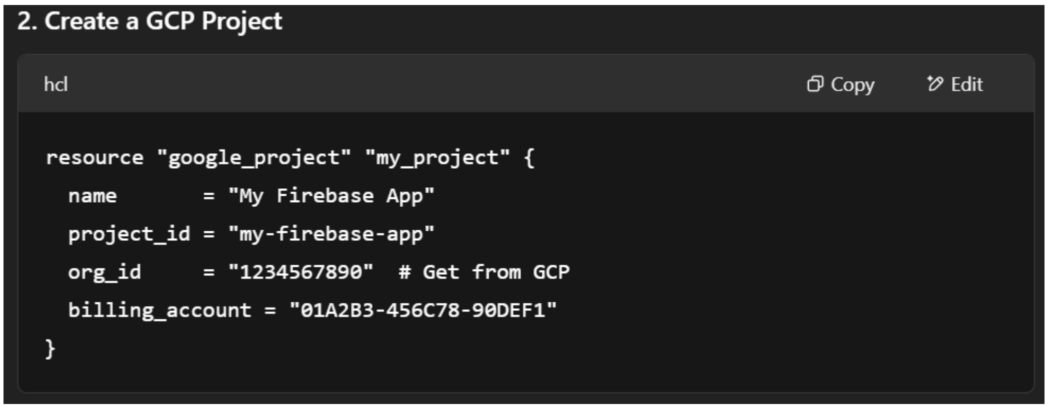


1. Configure the Provider

hcl

Copy Edit

```hcl
provider "google" {
  credentials = file("key.json")   # Downloaded from GCP Console
  project     = "my-firebase-app"
  region      = "us-central1"
}
```

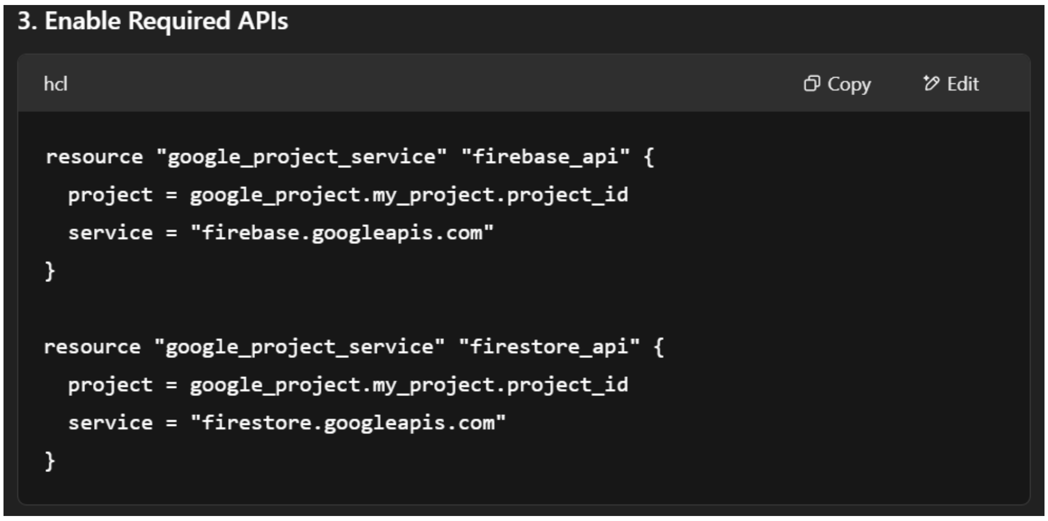


2. Create a GCP Project

hcl

Copy Edit

```hcl
resource "google_project" "my_project" {
  name       = "My Firebase App"
  project_id = "my-firebase-app"
  org_id     = "1234567890"  # Get from GCP
  billing_account = "01A2B3-456C78-90DEF1"
}
```



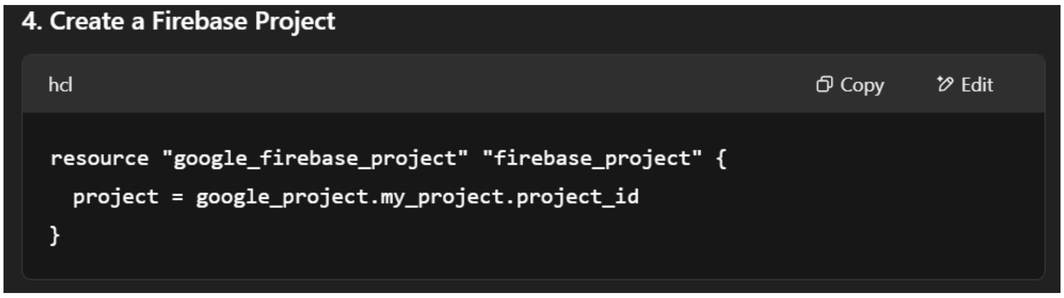
3. Enable Required APIs

hcl

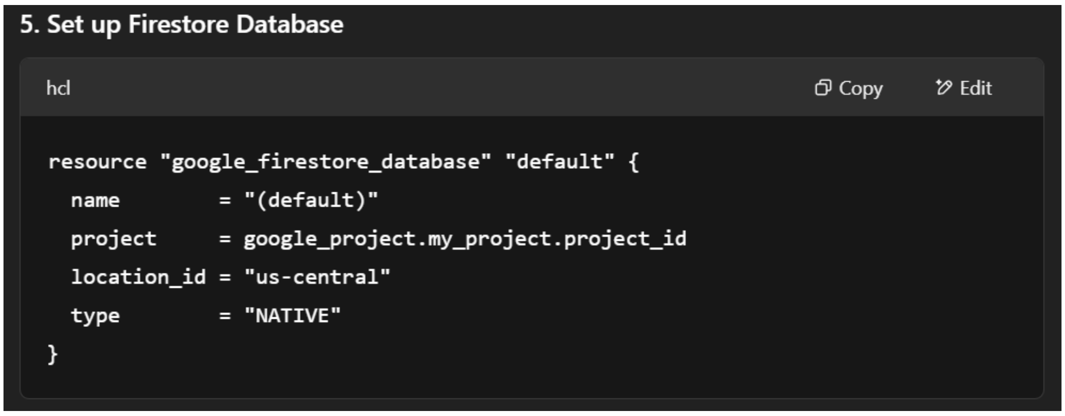
Copy Edit

```hcl
resource "google_project_service" "firebase_api" {
  project = google_project.my_project.project_id
  service = "firebase.googleapis.com"
}

resource "google_project_service" "firestore_api" {
  project = google_project.my_project.project_id
  service = "firestore.googleapis.com"
}
```


4. Create a Firebase Project

```hcl
resource "google_firebase_project" "firebase_project" {
  project = google_project.my_project.project_id
}
```


5. Set up Firestore Database

```hcl
resource "google_firestore_database" "default" {
  name        = "(default)"
  project     = google_project.my_project.project_id
  location_id = "us-central"
  type        = "NATIVE"
}
```

Once upon a time, in a land where ideas floated on fluffy white clouds, there was a bustling kingdom called Cloudoria. In Cloudoria lived two best friends: Mia, a curious inventor, and Leo, an aspiring artist. Together they dreamed of sharing their creations with the whole world—but they needed a way to store, protect, and deliver their work.

The Great Invitation

One morning, a royal messenger arrived bearing a golden scroll:

"You are cordially invited to display your masterpiece at the Annual Sky Festival in the capital city, Nimbus. Only those who harness the power of the Cloud Keepers' magic may enter."

Mia and Leo exchanged worried glances. "How will we secure our work, make it instantly available, and share it far and wide?" Mia wondered.

Meeting the Cloud Keepers

They journeyed to the highest tower of Cloudoria, where six wise keepers awaited:

1. Guardian Aria, the Gatekeeper who wore a shining badge and welcomed only those with the proper pass. She introduced herself with a warm smile: "I am Authentication (Firebase Authentication). I make sure everyone who enters is who they say they are—no impostors allowed."
2. Scribe Rion, who carried quills and scrolls that updated themselves the instant someone wrote new words in them. He giggled, "I'm Realtime, the living chronicle (Realtime Database). Whenever you add a sentence, all scholars see it at once—no delays!"
3. Archivist Flora, the gentle keeper of endless hallways filled with labeled tomes. "I'm Firestore (Cloud Firestore)," she said. "You can ask any question—'Show me every painting with a rainbow'—and I'll fetch the exact scroll in a heartbeat."
4. Chestmaster Orion, who guarded shining vaults deep beneath the tower. "I'm Storage (Cloud Storage). Bring me your sculptures, music boxes, or canvases—I'll keep them safe, and share them only with those you choose."
5. Wizard Coda, a merry fellow with a twinkling hat. "I'm Functions (Cloud Functions)," he declared with a flourish. "When something special happens—like a new song uploaded—I spring into action and perform little spells: resizing images, sending messages, you name it!"
6. Herald Nova, who arranged grand spectacles on the floating stage above the tower. "I'm Hosting (Firebase Hosting). Whatever website or show you prepare, I'll broadcast it on the winds so every cloud-dweller can watch."

Preparing for the Festival

Armed with their new friends' powers, Mia and Leo set to work:

- ***Mia built a secure workshop door, calling on Authentication to let only invited guests inside. (Firebase Authentication)***
- ***Leo sketched a living mural, trusting Realtime to update every stroke instantly for visitors watching from distant islands. (Realtime Database)***
- ***Together they catalogued each fairy-tale illustration in Firestore, so curious onlookers could filter by color, theme, or character with ease. (Cloud Firestore)***
- ***They stored their music tracks and high-definition videos in Storage, handing out magic keys so friends could stream them without worry. (Cloud Storage)***
- ***Whenever Leo uploaded a new melody to their gallery, Functions automatically created a short teaser clip and sent it to their subscribers' mailboxes. (Cloud Functions)***
- ***Finally, Hosting displayed their interactive gallery and streaming stage atop a cloud platform, ensuring festival-goers from every kingdom could join the fun. (Firebase Hosting)***

The Spectacular Sky Festival

On the day of the festival, the skies above Nimbus glittered with flying lanterns. Thanks to the Cloud Keepers:

- Mia's guestbook only let genuine fans sign and leave messages. (Authentication)
- Leo's mural changed color live as remote visitors added their own drawings from afar. (Realtime Database)
- A curious child in a faraway village searched for "dragon pictures" and found exactly the panels Leo had painted—thanks to Firestore. (Cloud Firestore)
- Streams of their songs played without buffering because Storage delivered content from the nearest cloud tower. (Cloud Storage)
- When someone clicked "Play Preview," Functions had already prepared the short clip in advance—no waiting! (Cloud Functions)
- And everyone, whether on a floating cloudboat or a sky-train, enjoyed the festival website loaded in a flash via Hosting. (Firebase Hosting)

As the sun dipped behind cotton-candy clouds, Mia and Leo took a bow. They had shared their creations seamlessly, securely, and at lightning speed—all thanks to the magic of Firebase and Google Cloud.

The Moral of the Tale

Whenever you need to build something secure (Authentication), real-time (Realtime Database), searchable (Firestore), safe (Storage), automated (Functions), and globally delivered (Hosting), remember the Cloud Keepers of Cloudoria. Their teamwork turned a simple festival invitation into a legendary, unforgettable spectacle—just like Firebase and Google Cloud can transform your apps in the real world!