



Serverless

"Serverless is a cloud computing execution model where the cloud provider dynamically manages the allocation and provisioning of servers. A serverless application runs in stateless compute containers that are event-triggered, ephemeral (may last for one invocation), and fully managed by the cloud provider."



Benefits

- *Reduced time to market and quicker software release.*
- *Reduces the complexity of software.*
- *Simplifies packaging and deployment and requires no system administration.*
- *Lower operational and development costs. (to some point)*
- *Scalability - you are very flexible with traffic handling*
- *Serverless is managed by a cloud provider Your servers are small, the cloud provider dynamically creates or destroys servers when demand grows or shrinks*
- *Fits with microservices, which can be implemented as functions.*

Providers



Google Clouds



AWS



Azure



Firebase

<https://firebase.google.com/>



Build better apps



Cloud Firestore

Store and sync app data at global scale



Firebase ML ^{BETA}

Machine learning for mobile developers



Cloud Functions

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



Test Lab

Test your app on devices hosted by Google



App Distribution ^{BETA}

Distribute pre-release versions of your app to your trusted testers



Grow your business



In-App Messaging ^{BETA}

Engage active app users with contextual messages



Google Analytics

Get free and unlimited app analytics



Predictions

Smart user segmentation based on predicted behavior



A/B Testing ^{BETA}

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution

End of backenders era? 🦷



Setup

1. ``npm i -g firebase-tools``
2. ``firebase init PROJECT_TYPE`` - to initialise a project
 - a. **Project types:** database, firestore, functions, hosting, storage, or emulators
3. ``firebase init functions``



```
1 const functions = require("firebase-functions");
2 const express = require("express");
3
4 const app = express();
5
6 app.get("/", (req, res) => {
7   res.send("Hello Friend!");
8 });
9
10 exports.app = functions.https.onRequest(app);
11
```



firebase.json

```
1 {  
2   "hosting": {  
3     ...  
4     "rewrites": [  
5       {  
6         "source": "**",  
7         "function": "app"  
8       }  
9     ]  
10  }  
11  ...  
12 }
```

``firebase serve --only functions,hosting``

To run emulation locally



Enable cache

```
1 app.get("/user", (req, res) => {  
2   // public/private - to cache on the server or user's machine  
3   // max-age - how long to store in the user's browser  
4   // s-maxage - how long to store in the CDN cache  
5  
6   res.set("Cache-Control", "public, max-age=300, s-maxage=600");  
7   res.send(`${Date.now()}`);  
8 });
```

Connect firebase DB

```
1 const firebase = require("firebase-admin");  
2  
3 const firebaseApp = firebase.initializeApp(functions.config.firebase);  
4 // Example  
5 const getFriends = async () => {  
6   const ref = firebaseApp.database().ref("friends");  
7   return await ref.once("value").then((snapshot) => snapshot.val());  
8 };
```

Firestore DB docs

Setup

<https://firebase.google.com/docs/database/admin/start#node.js>

Get data

<https://firebase.google.com/docs/database/admin/retrieve-data>

Save data

<https://firebase.google.com/docs/database/admin/save-data>



Firestore Web code lab

<https://codelabs.developers.google.com/codelabs/firebase-web/>



AWS

<https://aws.amazon.com/sdk-for-node-js/>

