

# How to Set Up Firebase for a Node.js Project

Scribe

- 1 Navigate to <https://console.firebaseio.google.com/u/0/>

- 2 Click the "Enter your Project name" field.

Let's start with a name for  
your project <sup>?</sup>

Enter your project name

my-awesome-project-id

Continue

- 3 Type "gpt-node-firebase"

4 Click here.

## Let's start with a name for your project<sup>②</sup>

Project name

gpt-node-firebase

 gpt-node-firebase

Continue

5 Click here.

ting, and Cloud Functions.

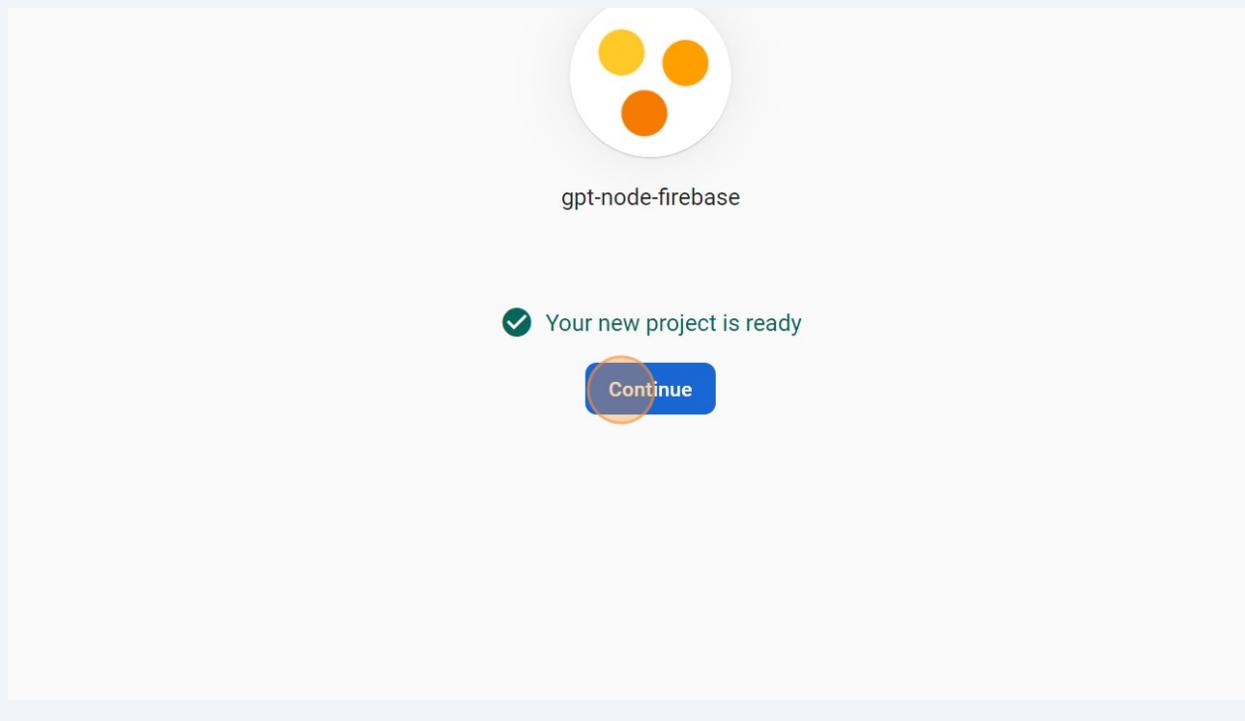
nables:

-   Crash-free users <sup>②</sup>
-   Event-based Cloud Functions triggers <sup>②</sup>
-   Free unlimited reporting <sup>②</sup>

Google Analytics for this project  
nded

Create project

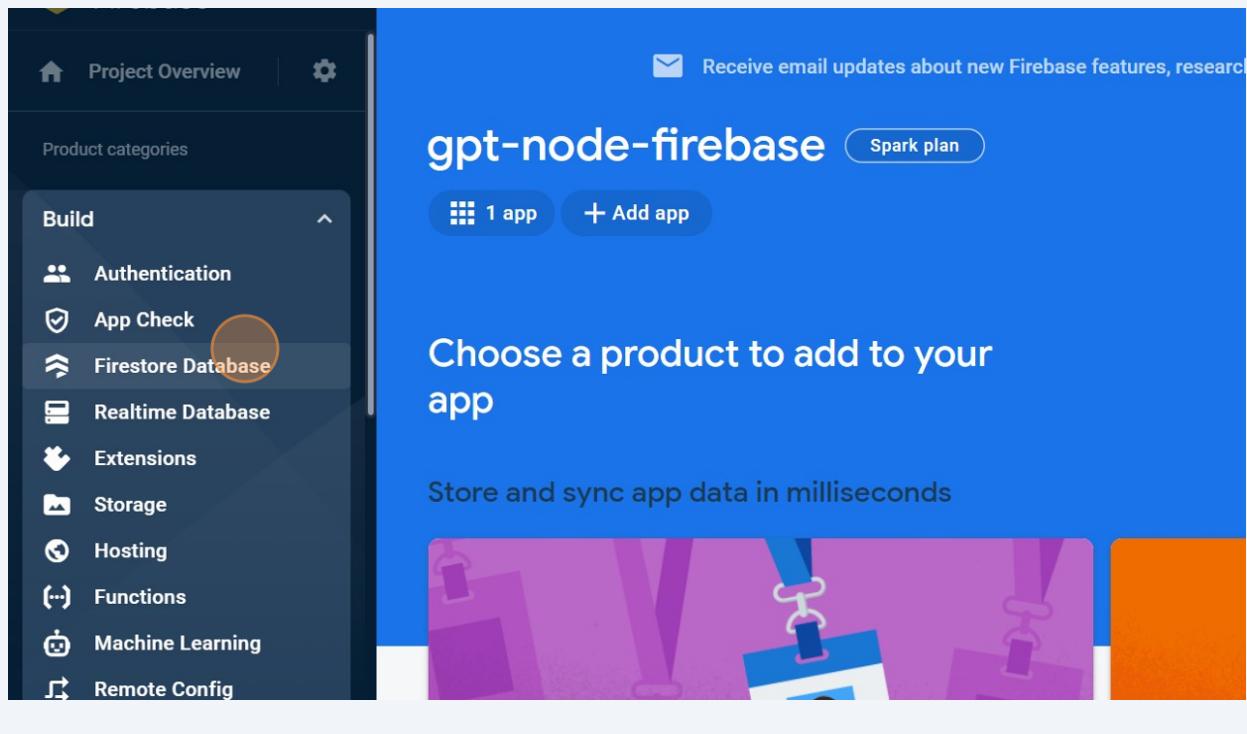
6 Click "Continue"



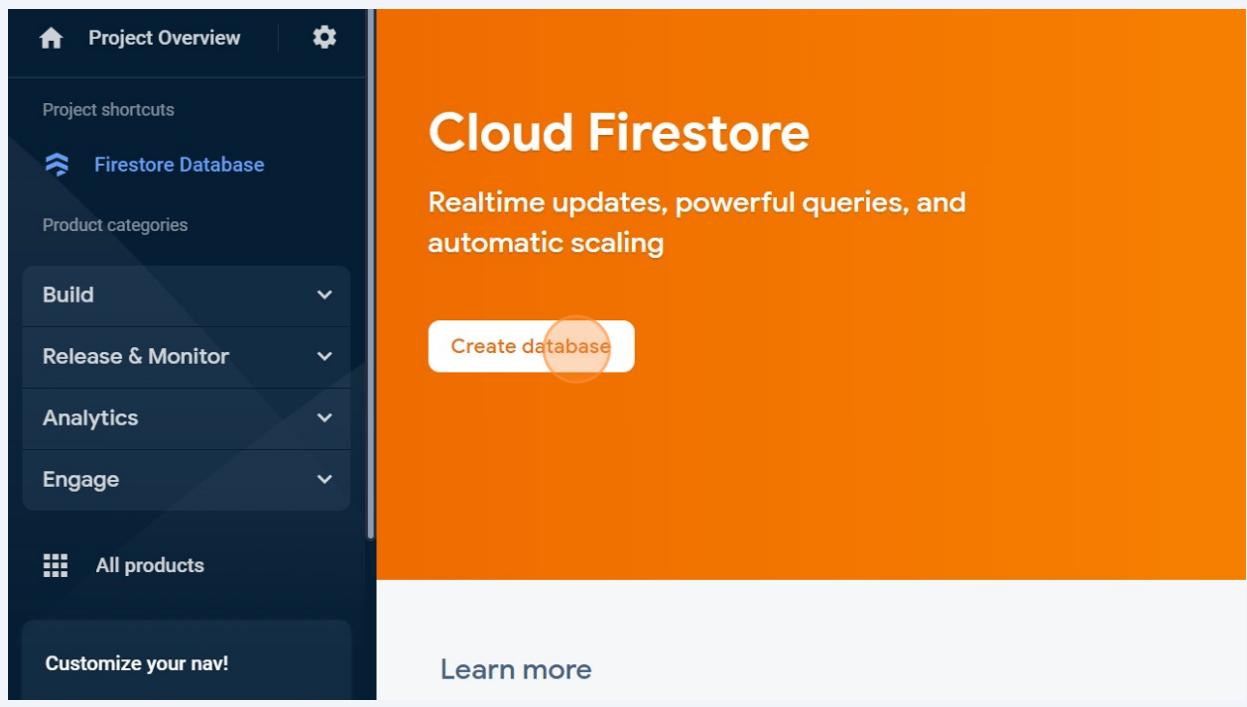
7 Click "Build"

A screenshot of the Firebase console. On the left, a dark sidebar shows navigation options: 'Project Overview' (selected), 'Build' (highlighted with a red circle), 'Release &amp; Monitor', 'Analytics', 'Engage', 'All products', and 'Customize your nav!'. To the right, the main area is titled 'gpt-node-firebase' and shows a summary: '1 app' (with a plus icon to add more). It also features a 'Spark plan' button and a message to 'Receive email updates about new Firebase features, research, and best practices'. Below this, a large call-to-action button says 'Choose a product to add to your app' and 'Store and sync app data in milliseconds'. The bottom of the sidebar has a message: 'You can now focus your console'.

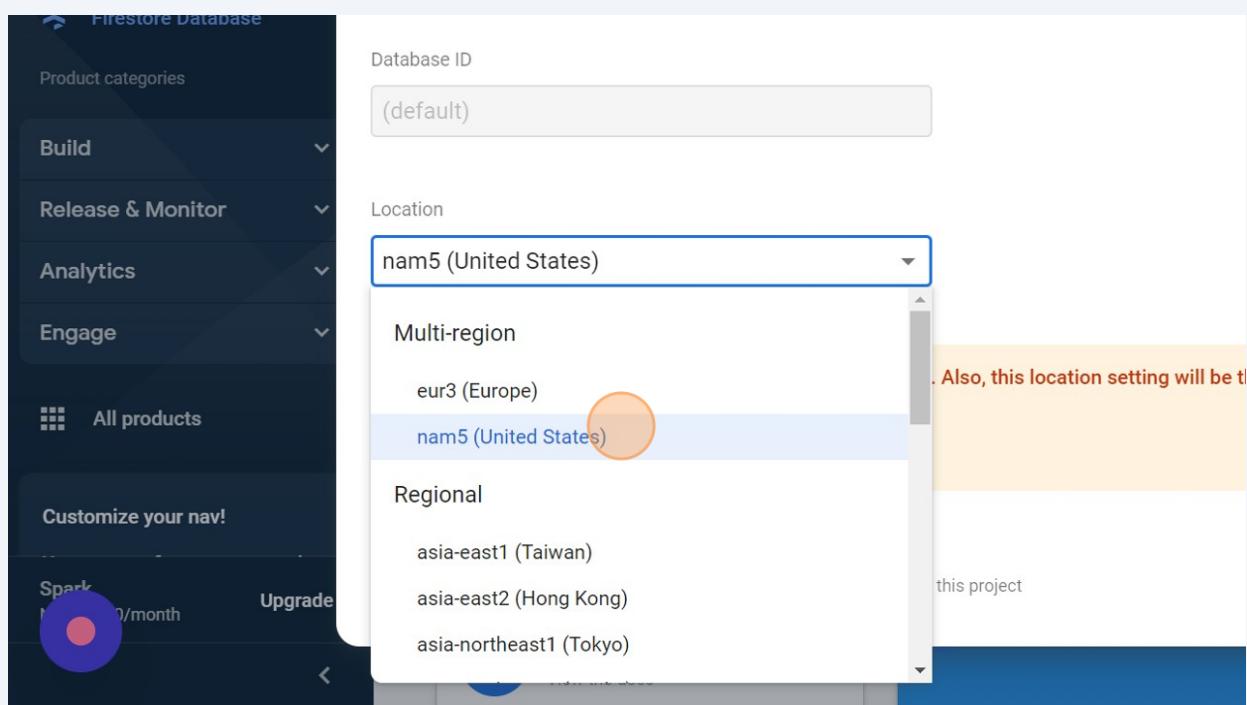
8 Click "Firestore Database"



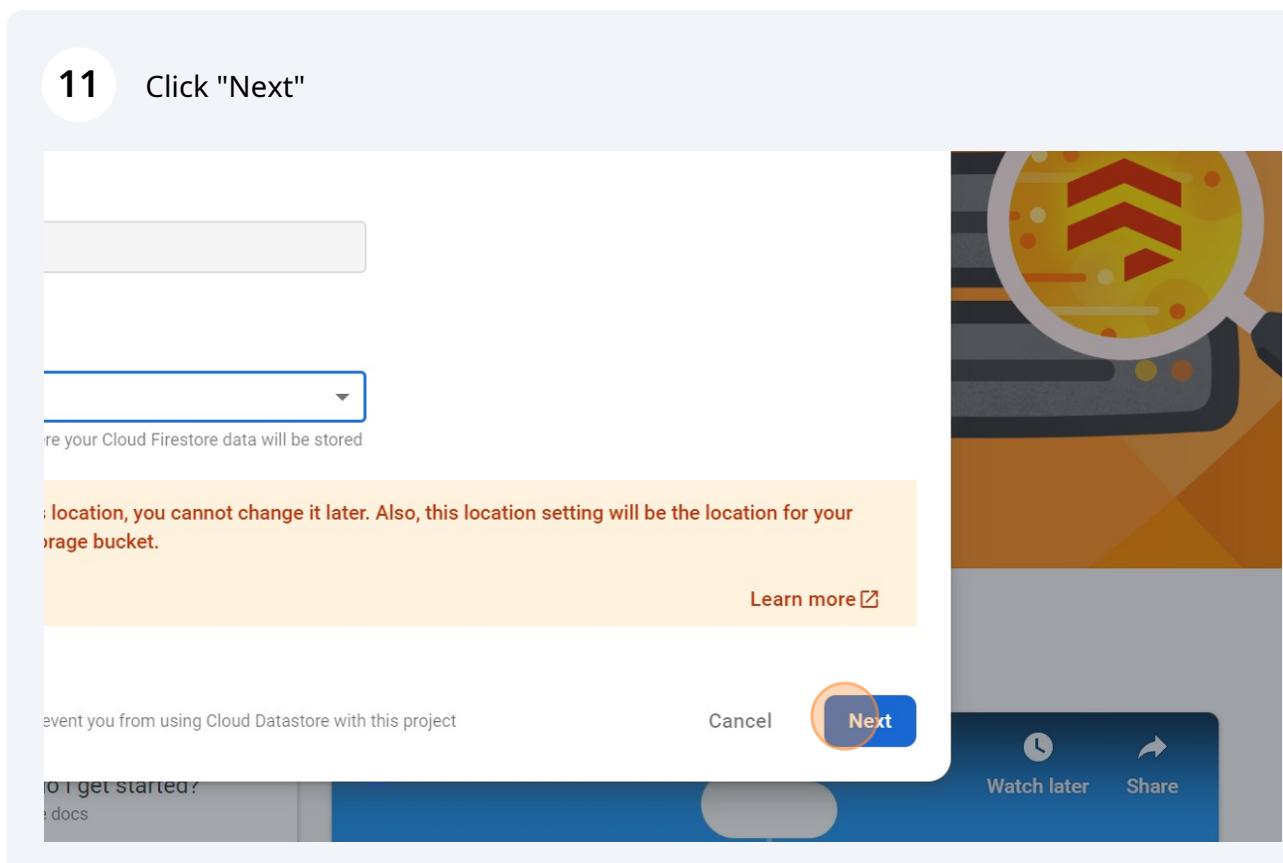
9 Click "Create database"



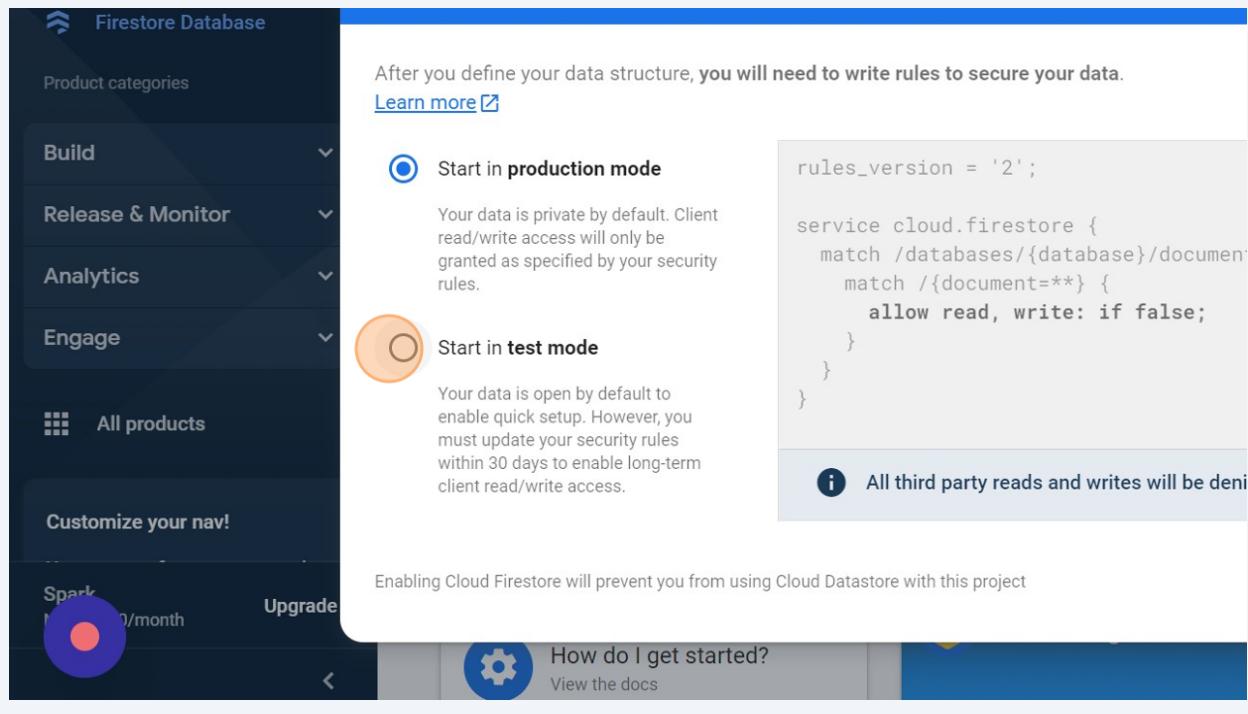
## 10 Select a location



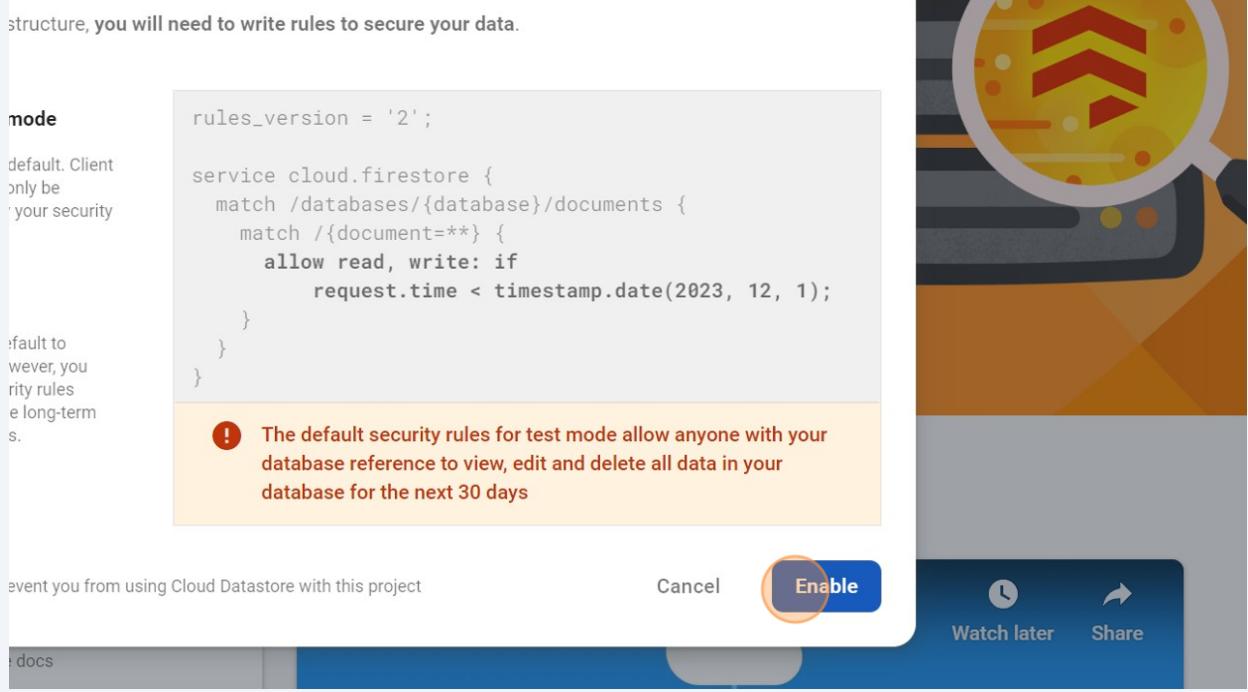
## 11 Click "Next"



## 12 Click the "Start in test mode" field.



## 13 Click "Enable"



14 Click "Project settings"

The screenshot shows the Firebase Project Overview interface. On the left, there's a sidebar with 'Project Overview', 'Firestore Database', and other build-related sections like Release & Monitor, Analytics, and Engage. The main area shows a collection named '(default)' with a '+ Start collection' button. At the top right, there's a dropdown for the project name 'gpt-node-firebase'. A context menu is open over the 'Project settings' link in the top navigation bar, with options like 'Project settings', 'Users and permissions', and 'Usage and billing'. The 'Project settings' option is circled in orange.

15 Click "Service accounts"

The screenshot shows the 'Project settings' page. At the top, there are tabs for General, Cloud Messaging, Integrations, Service accounts (which is highlighted with an orange circle), Data privacy, and Users and permissions. Below the tabs, there's a section titled 'Your project' containing various project details: Project name (gpt-node-firebase), Project ID (gpt-node-firebase), Project number (988406647234), Default GCP resource location (Not yet selected), and Web API Key (No Web API Key for this project).

**16** Scroll down to see the Admin SDK configuration setup for Nodejs

Firebase Admin SDK

[Manage service account permissions](#)

service accounts

**Firebase Admin SDK**

Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. [Learn more](#)

Admin SDK configuration snippet

Node.js  Java  Python  Go

```
var admin = require("firebase-admin");
```

**17** We need to generate a new private key in order to allow our backend app to interact with the firebase database.

service accounts

[6 service accounts](#)

Admin SDK configuration snippet

Node.js  Java  Python  Go

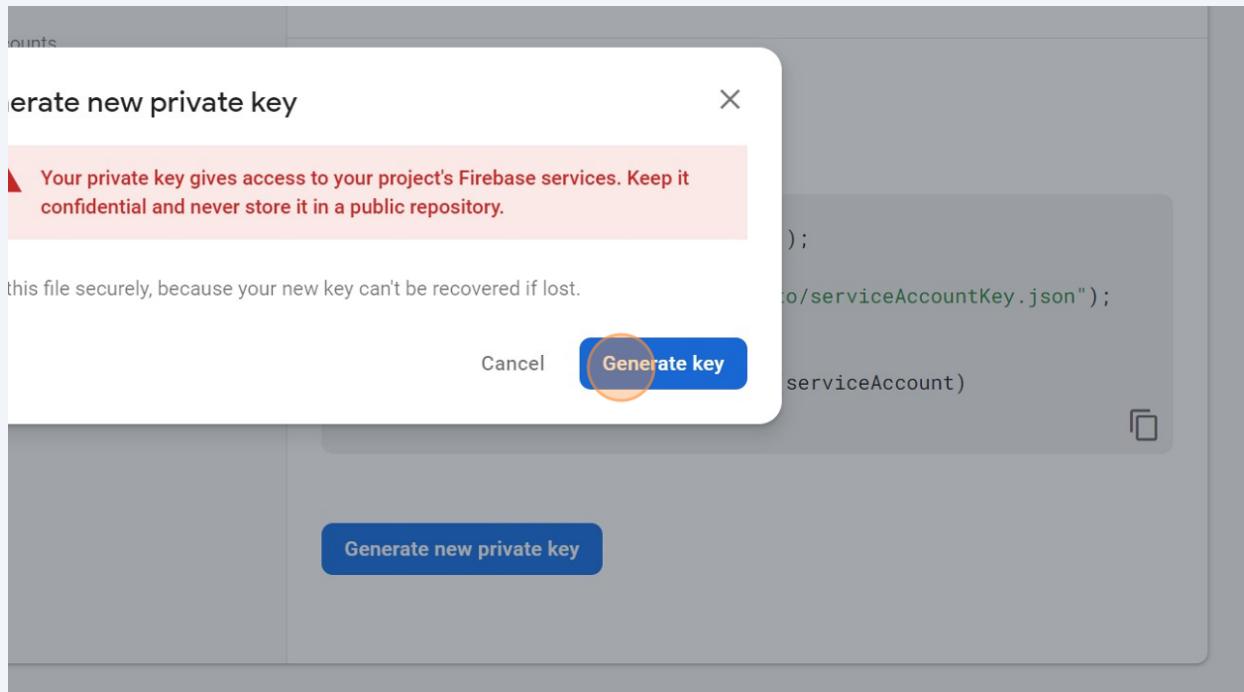
```
var admin = require("firebase-admin");

var serviceAccount = require("path/to/serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});
```

Generate new private key

**18** Click "Generate key"



**19** Copy the configuration snippet and use it in the backend application along with the private key.

