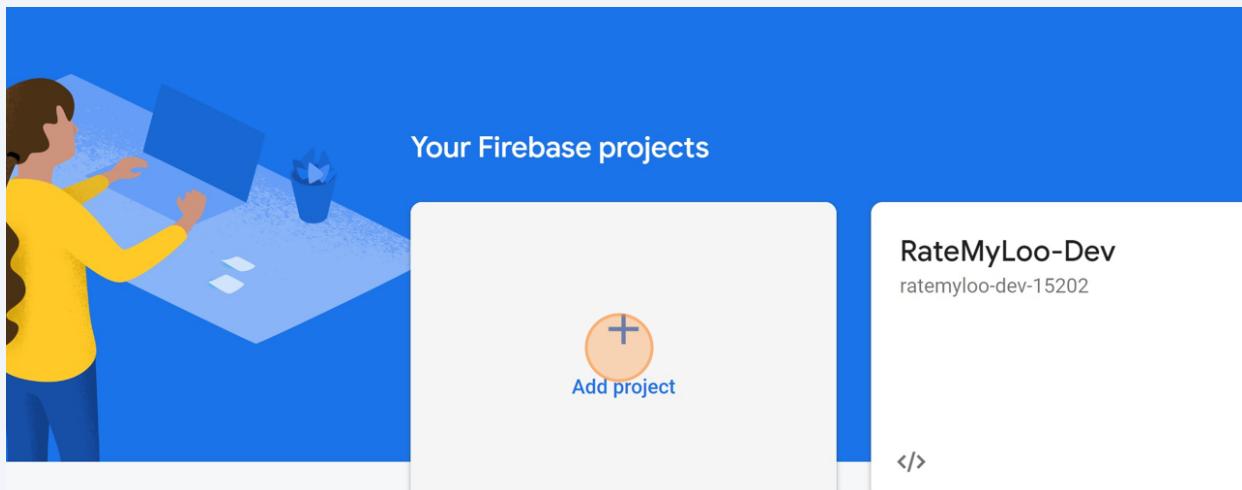


Step-by-Step Guide: Creating a Firebase Project

Scribe 

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

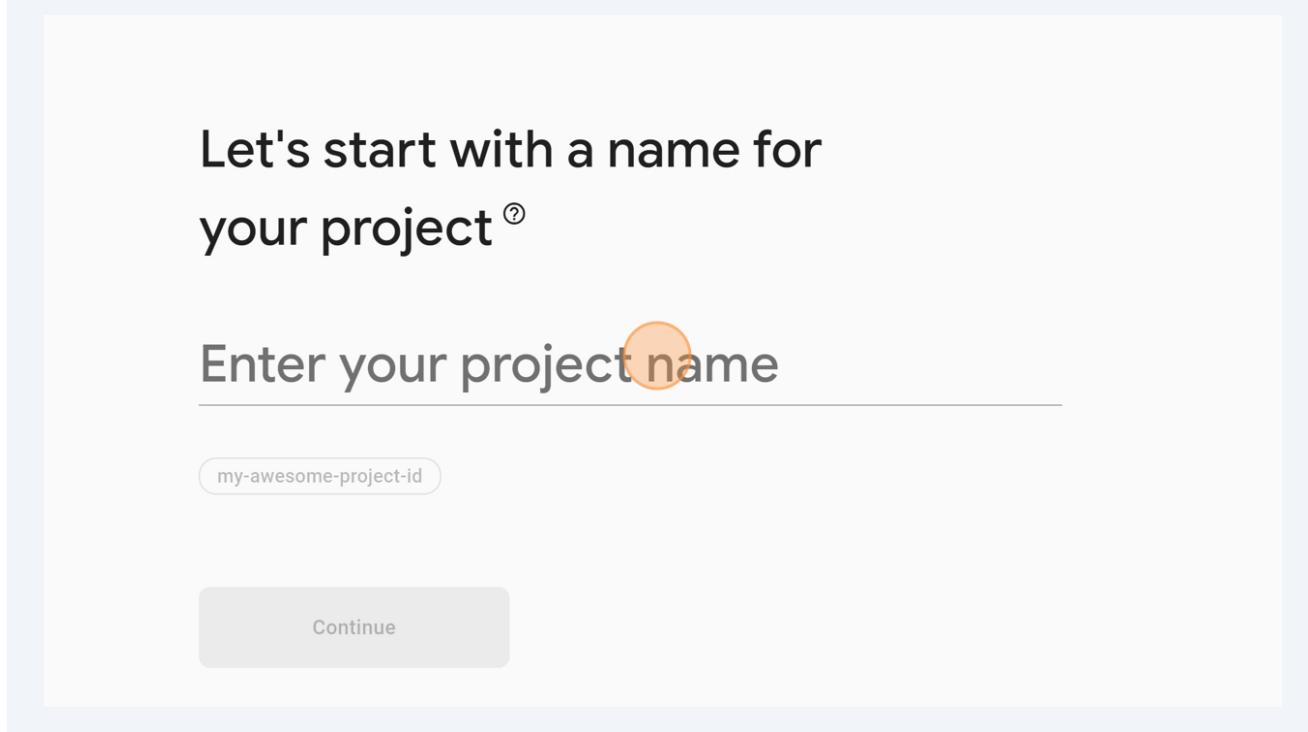
- 2 Click "Add Project"



RateMyLoo 
ratemyloo-c9c7e

Tyler1 Bot DEV
tyler1-bot-dev

- 3 Enter your project name in the field and click "Continue"



- 4 Click "Create Project". Feel free to turn on Google Analytics, but it is not required for this project (if so, click "Continue")

sting, and Cloud Functions.

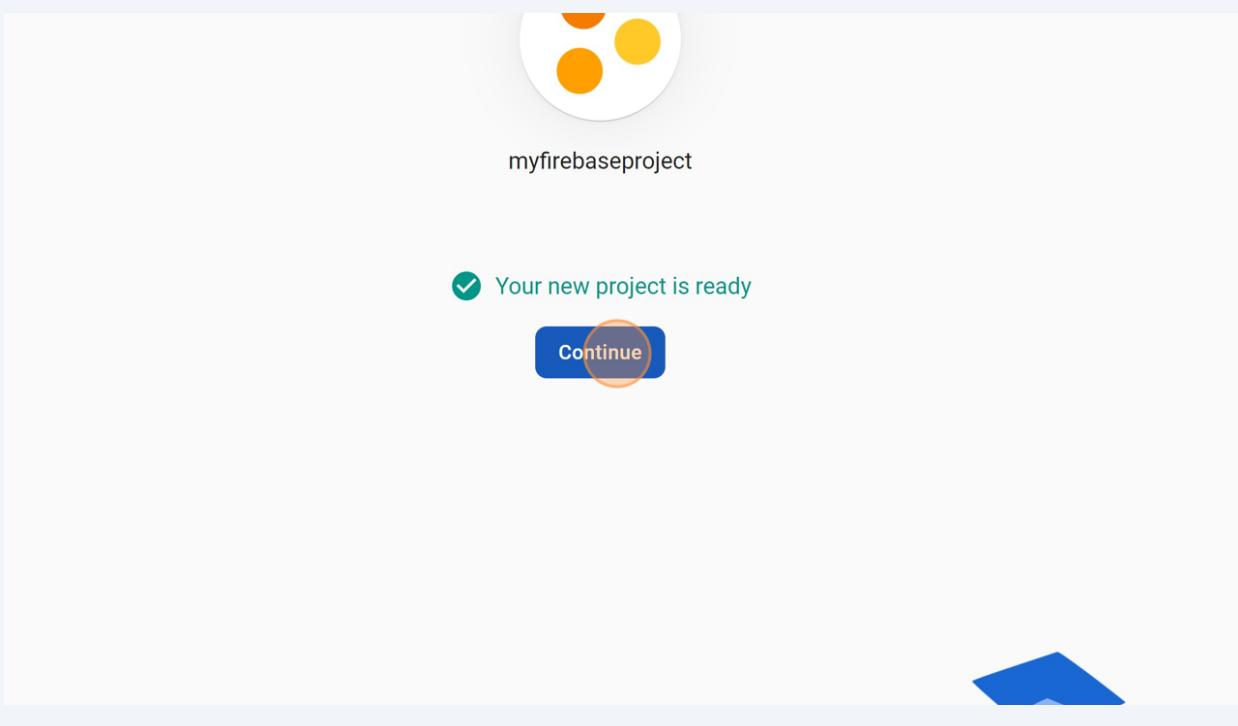
enables:

- [Analytics & targeting across products](#) ?
- [Crash-free users](#) ?
- [Event-based Cloud Functions triggers](#) ?
- [Free unlimited reporting](#) ?

Google Analytics for this project
ended



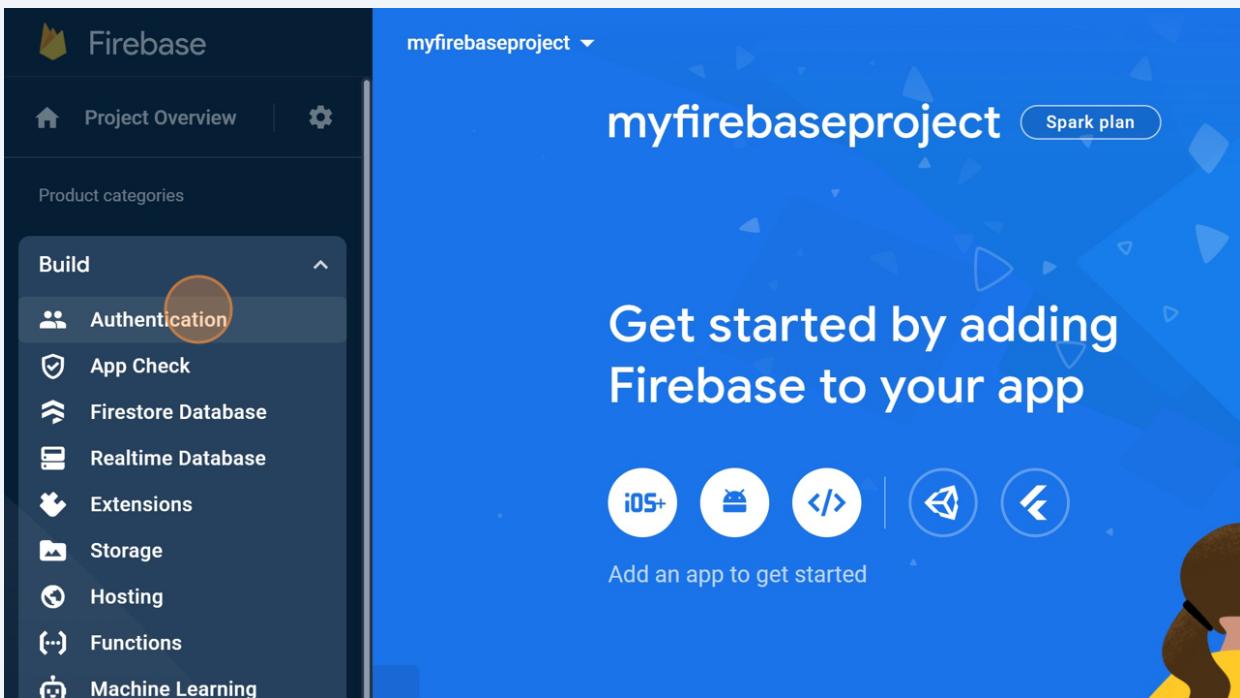
5 Click "Continue"



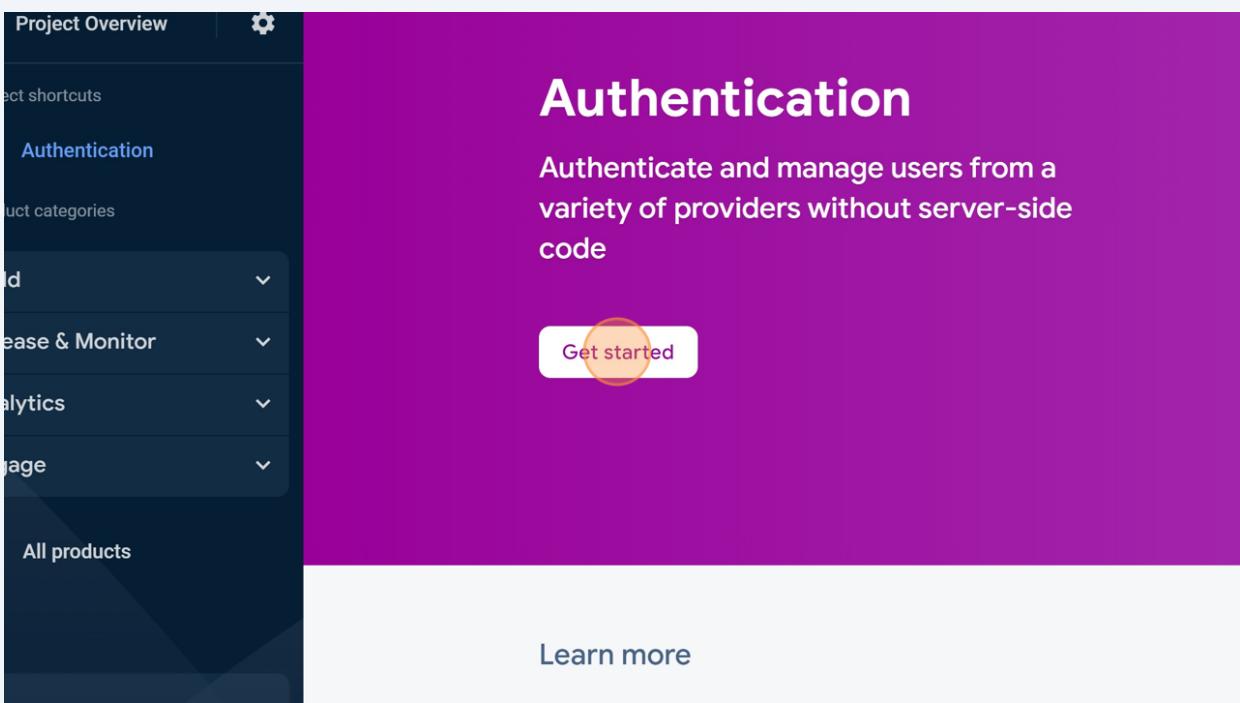
6 Click "Build"

A screenshot of the Firebase console showing the "Project Overview" page for a project named "myfirebaseproject". The sidebar on the left has a dark blue background and contains the following menu items: "Build" (which is highlighted with a red circle), "Release & Monitor", "Analytics", and "Engage". Below the menu is a "All products" section with a grid icon. The main area of the screen has a blue background with white text. It says "myfirebaseproject" at the top and "Spark plan" in a small circle. Below that, it says "Get started by adding Firebase to your app". There are icons for "iOS+", "Android", "HTML", "React Native", and "Flutter". At the bottom, it says "Add an app to get started".

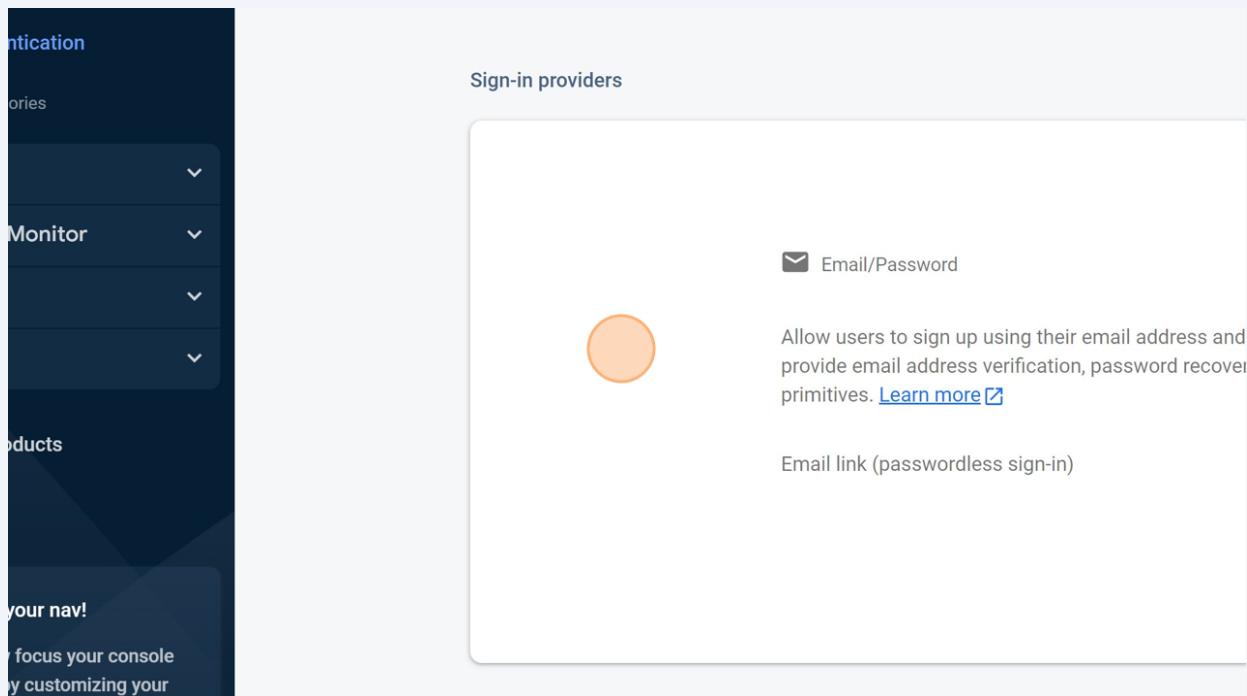
7 Click "Authentication"



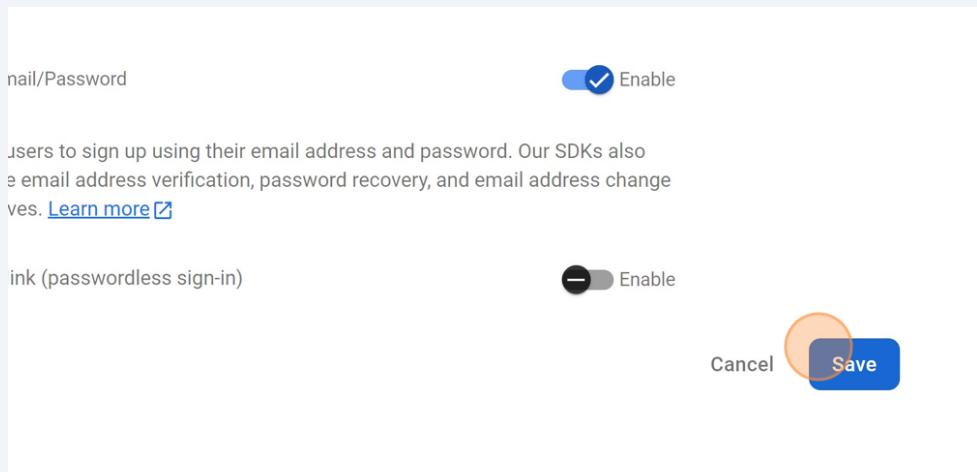
8 Click "Get started"



9 Enable "Email/Password"



10 Click "Save"



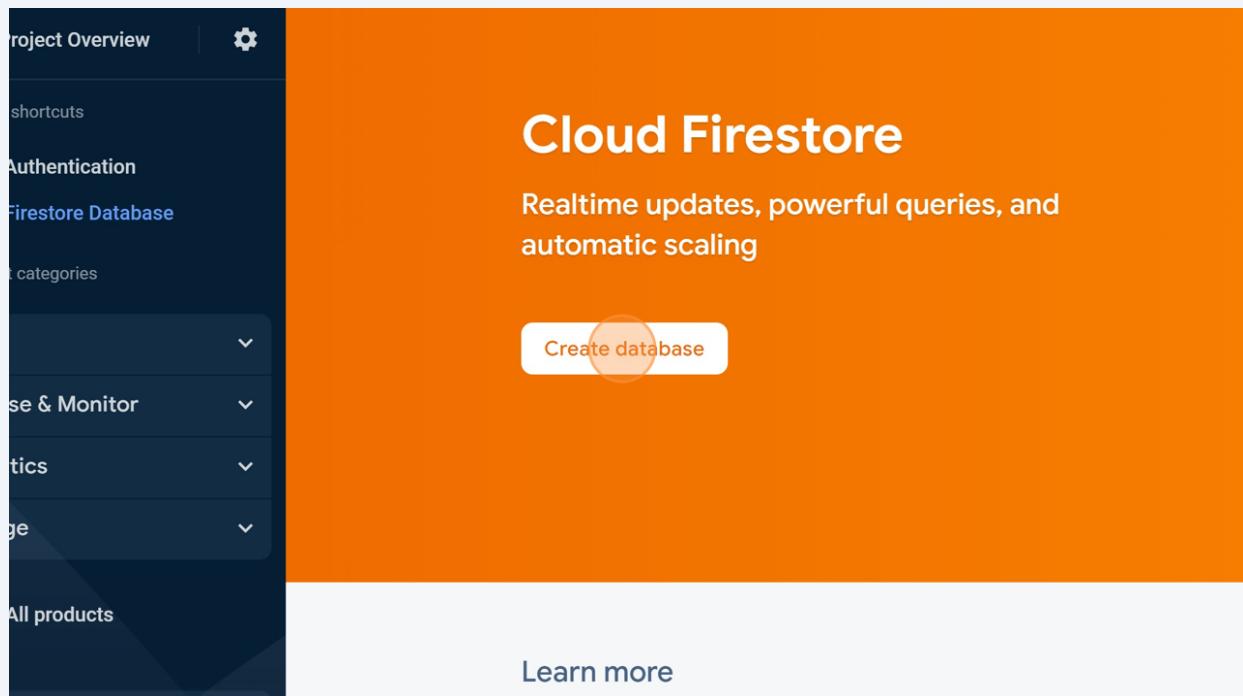
11 Click "Build"

The screenshot shows the Firebase console for a project named "myfirebasoproject". The left sidebar has a "Build" category highlighted with an orange circle. The main content area is titled "Authentication" and shows the "Sign-in method" tab selected. It displays "Sign-in providers" with "Email/Password" listed under the "Provider" section. Below this is an "Advanced" section containing a button for "SMS Multi-factor Authentication".

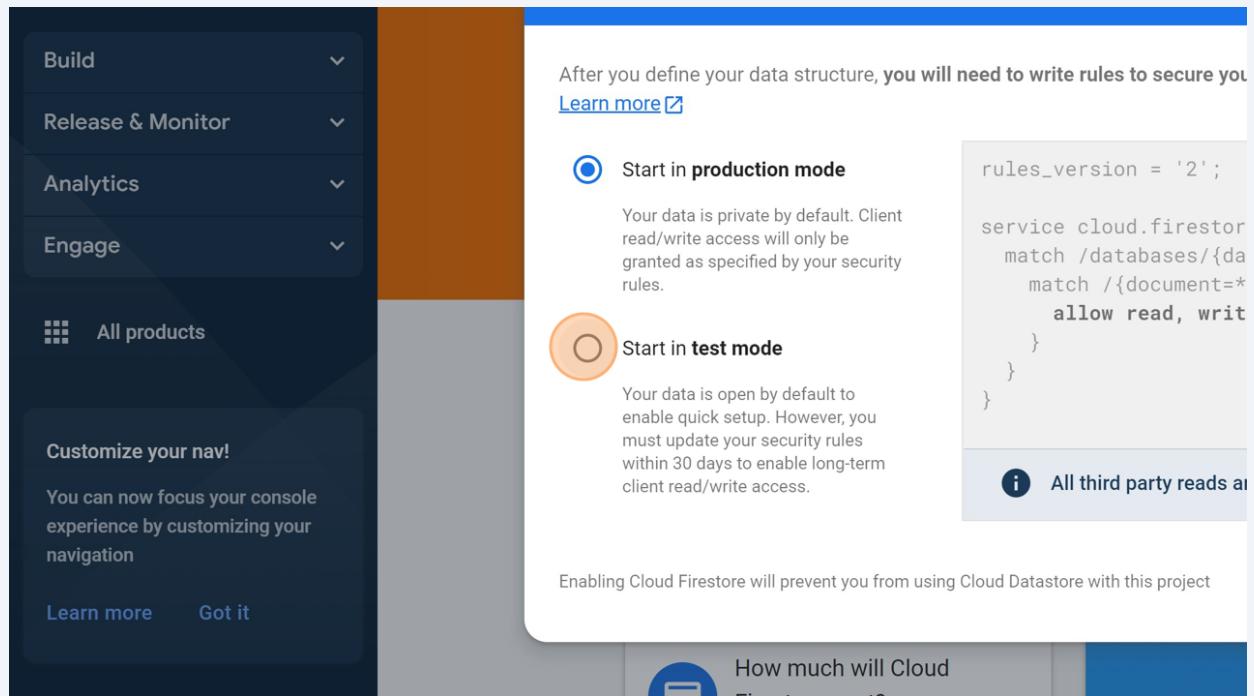
12 Click "Firestore Database"

The screenshot shows the Firebase console for a project named "myfirebasoproject". The left sidebar has a "Firestore Database" category highlighted with an orange circle. The main content area is titled "Authentication" and shows the "Sign-in method" tab selected. It displays "Sign-in providers" with "Email/Password" listed under the "Provider" section. Below this is an "Advanced" section containing a button for "SMS Multi-factor Authentication". A note at the bottom of this section says: "Allow your users to add an extra layer of security to their account. two steps, using SMS. [Learn more](#)".

13 Click "Create database"



14 Click the "Start in test mode" field.



15 Click "Next"

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if  
        request.time < timestamp.date(2023, 11, 16);  
    }  
  }  
}
```

⚠ The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cloud Datastore with this project

Cancel

Next

Watch later Share

do

16 Select the location for your Firestore Database (I usually set it to nam5)

Your location setting is where your Cloud Firestore data will be stored.

⚠ After you set this location, you cannot change it later. Also, this location setting will be the location of the default Cloud Storage bucket.

Learn more

Cloud Firestore location

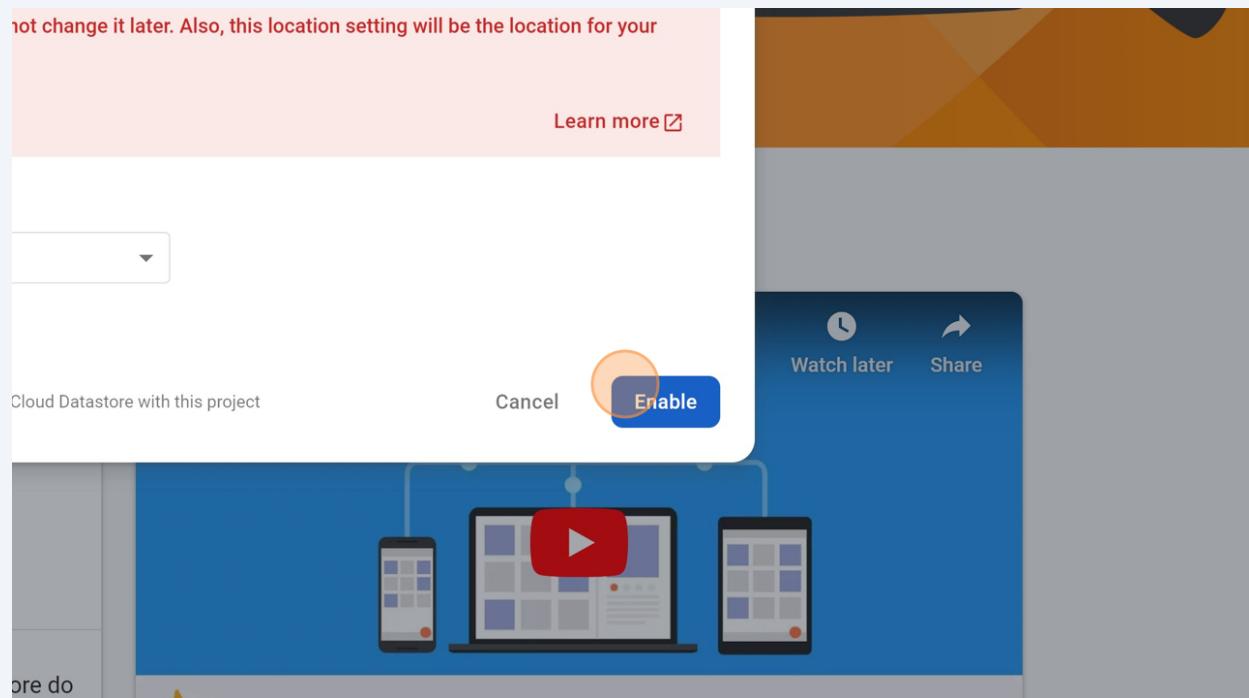
nam5 (United States)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project

Cancel

How much will Cloud Firestore cost?
[View pricing](#)

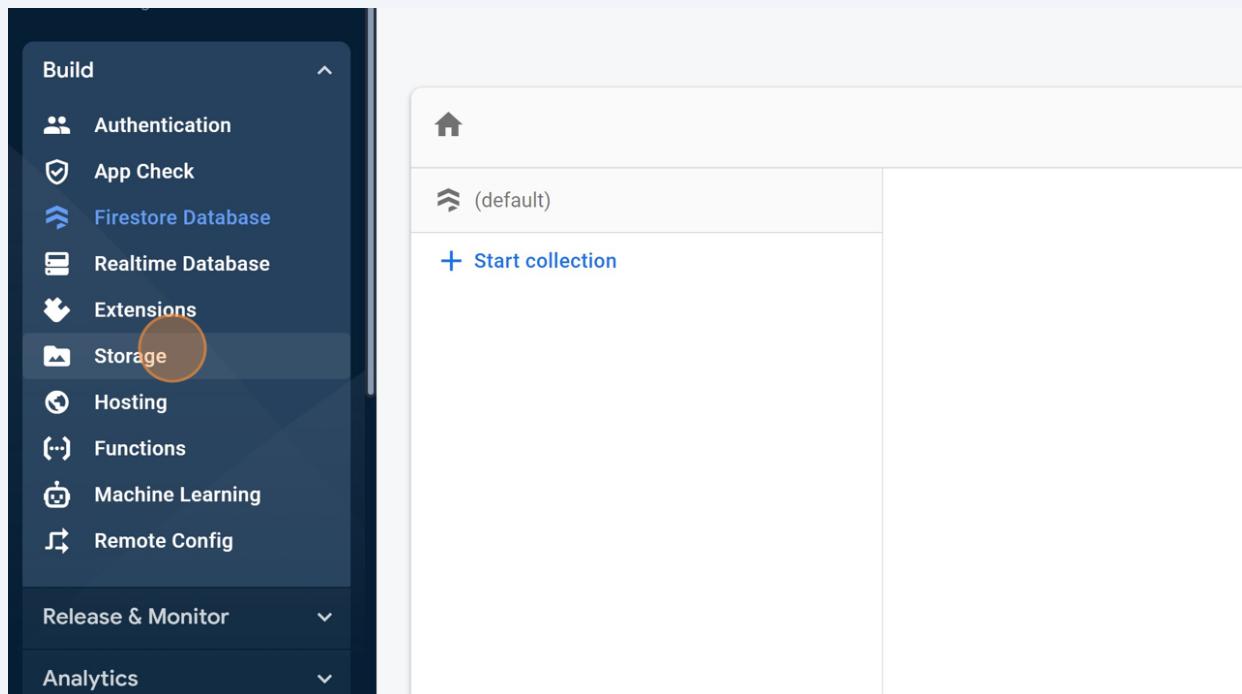
17 Click "Enable"



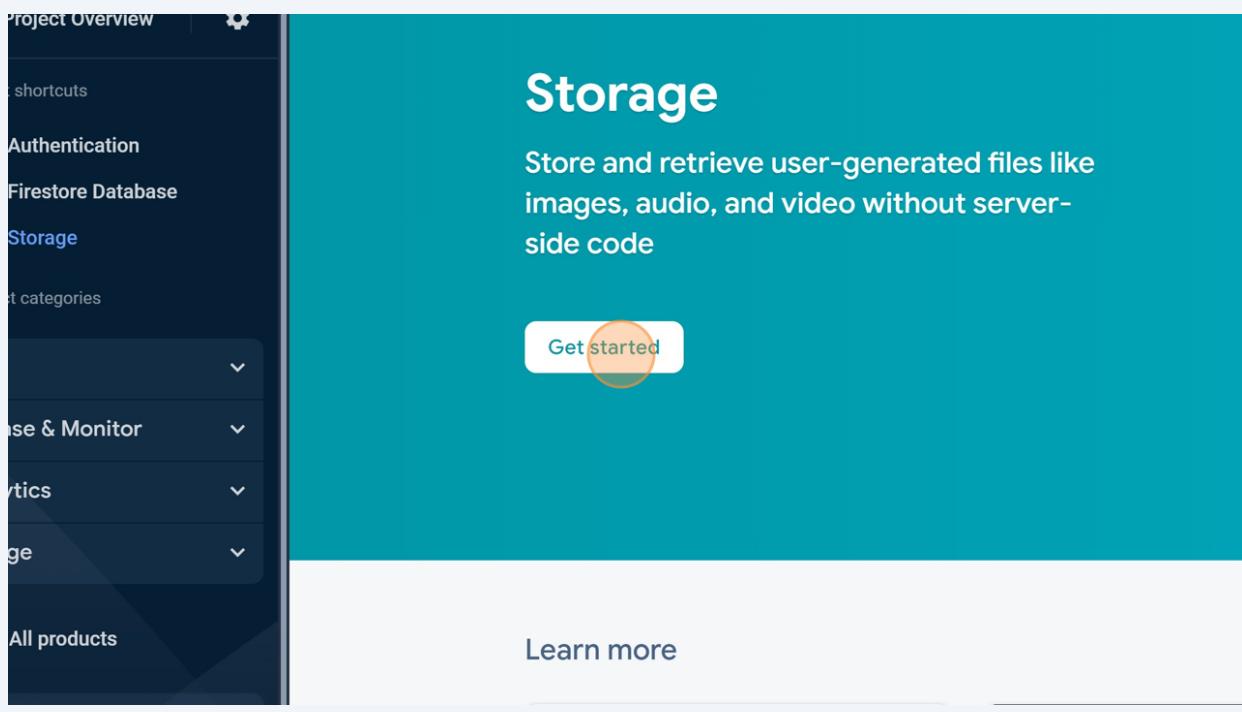
18 Click "Build"

A screenshot of the Cloud Firestore Project Overview page. On the left, a sidebar menu includes "Project Overview", "Authentication", "Firestore Database" (highlighted with a yellow circle), "Build" (highlighted with a brown circle), "Release & Monitor", "Analytics", "Engage", and "All products". The main content area is titled "Cloud Firestore" and shows the "Data" tab selected. It features a "Protect your Cloud Firestore resources from abuse" section with a shield icon and a "Start collection" button. The top navigation bar includes "Project shortcuts", "Data", "Rules", "Indexes", "Usage", and "Extensions".

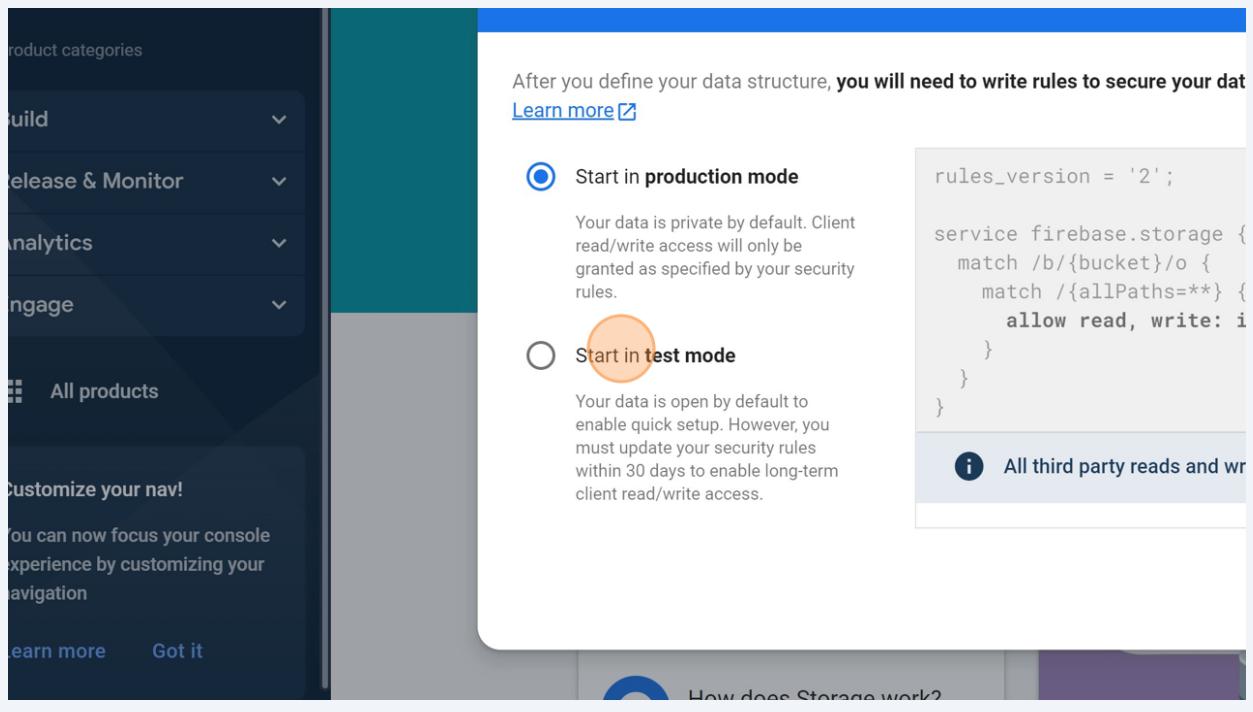
19 Click "Storage"



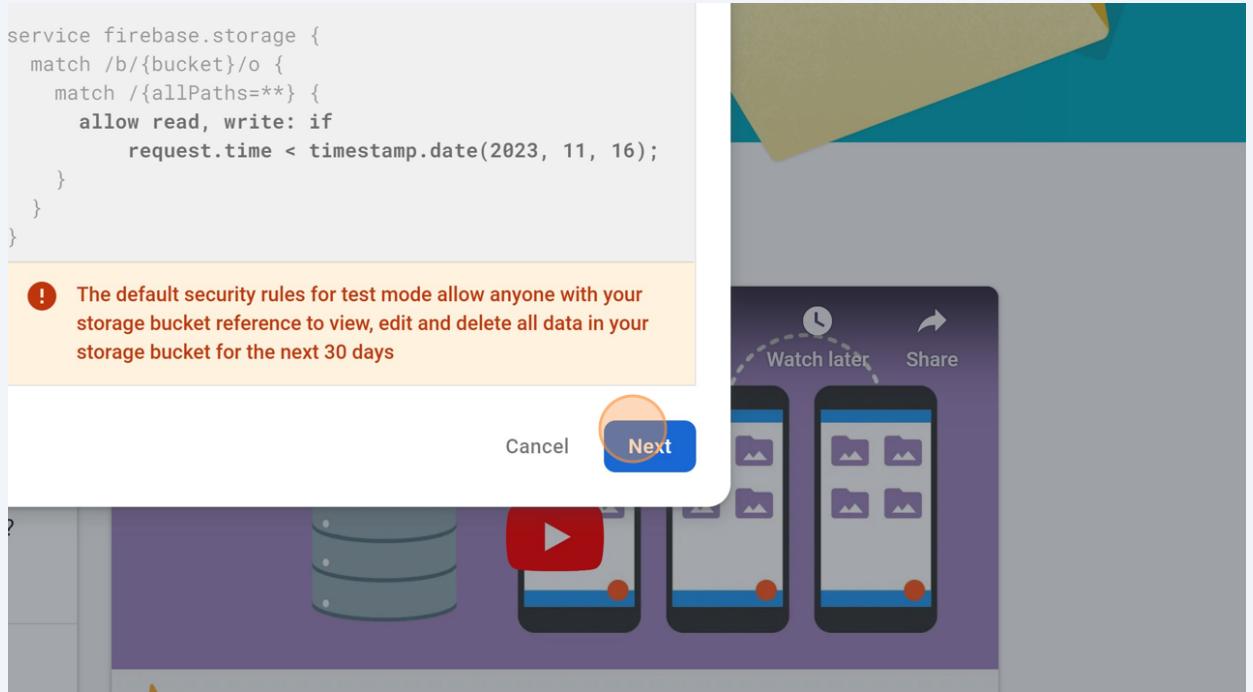
20 Click here.



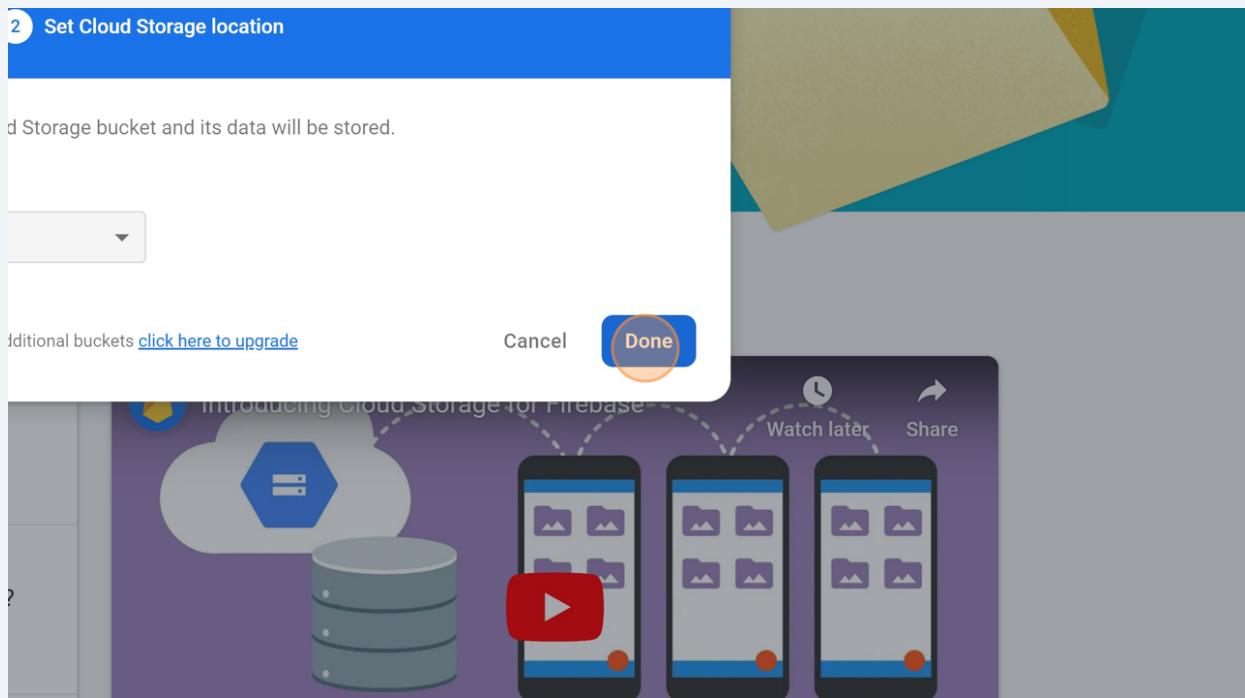
21 Click "Start in test mode"



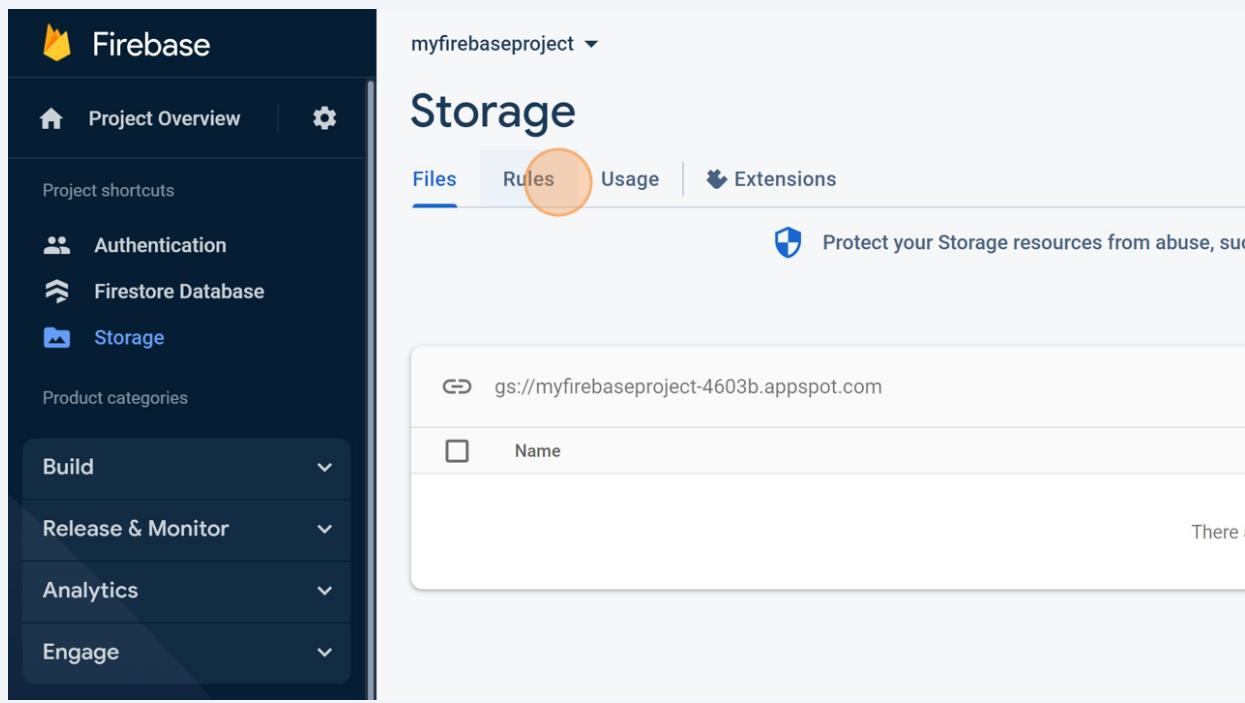
22 Click "Next"



23 Select the location for your Storage bucket, then click "Done"



24 Click here.



25 Remove the condition (it should be: allow read, write;)



data with rules that
is access to it and
is structured
[View the docs](#)

```
3 // Craft rules based on data in your Firestore database
4 // allow write: if firestore.get(
5 //   /databases/(default)/documents/users/$(request.auth.uid)).data.isAdm
6 service firebase.storage {
7   match /b/{bucket}/o {
8
9     // This rule allows anyone with your Storage bucket reference to view,
10    // and delete all data in your Storage bucket. It is useful for getting
11    // started, but it is configured to expire after 30 days because it
12    // leaves your app open to attackers. At that time, all client
13    // requests to your Storage bucket will be denied.
14    //
15    // Make sure to write security rules for your app before that time, or
16    // all client requests to your Storage bucket will be denied until you
17    // your rules
18    match /{allPaths=**} {
19      allow read, write: if request.time < timestamp.date(2023, 11, 16);
20    }
21  }
22}
```

und
ore with



26 Click "Publish"



Usage | Extensions



Write Security Rules that control access to Storage based on the contents of your Firestore Database.

[Learn more \[](#)



Looking for the Rules monitoring chart? It has been moved to the [Usage](#) tab for consolidated metric viewing.

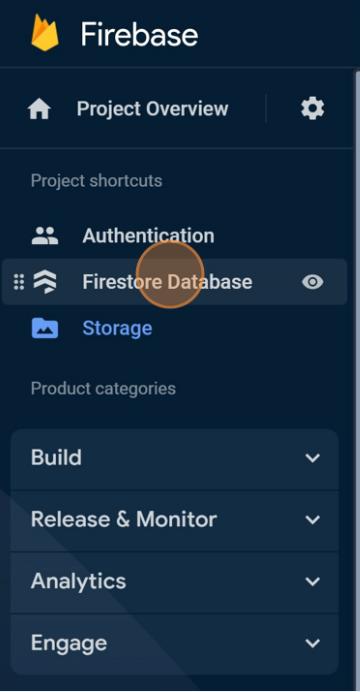


ur data with rules that
o has access to it and
it is structured
[View the docs](#)

unpublished changes Publish Discard

```
1 rules_version = '2';
2
3 // Craft rules based on data in your Firestore database
4 // allow write: if firestore.get(
5 //   /databases/(default)/documents/users/$(request.auth.uid)).data.isAdm
6 service firebase.storage {
7   match /b/{bucket}/o {
8
9     // This rule allows anyone with your Storage bucket reference to view,
10    // and delete all data in your Storage bucket. It is useful for getting
11    // started, but it is configured to expire after 30 days because it
12    // leaves your app open to attackers. At that time, all client
```

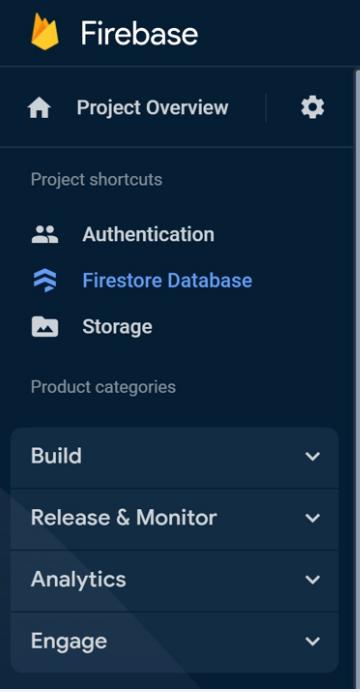
27 Click "Firestore Database"



The screenshot shows the Firebase console interface. On the left, the navigation sidebar includes 'Project Overview', 'Authentication', 'Firestore Database' (which is highlighted with an orange circle), and 'Storage'. The main content area is titled 'myfirebaseproject' and shows the 'Storage' tab. At the top, there are tabs for 'Files', 'Rules', 'Usage', and 'Extensions'. A prominent message says 'Write Security Rules that control access to Storage based'. Below it, another message says 'Looking for the Rules monitoring chart? It has been moved to the Rules tab'. On the right, there's a code editor window displaying a security rules file for Cloud Storage:

```
rules_version = '2';
// Craft rules based on data
// allow write: if firebase
//   /databases/(default)
service firebase.storage
match /b/{bucket}/o {
  // This rule allows all
}
```

28 Click "Rules"



The screenshot shows the Firebase console interface. On the left, the navigation sidebar includes 'Project Overview', 'Authentication', 'Firestore Database' (which is highlighted with an orange circle), and 'Storage'. The main content area is titled 'myfirebaseproject' and shows the 'Cloud Firestore' section. At the top, there are tabs for 'Data', 'Rules' (which is highlighted with an orange circle), 'Indexes', 'Usage', and 'Extensions'. A message says 'Protect your Cloud Firestore resources from abuse, set security rules'. Below, there's a collection list with '(default)' and a '+ Start collection' button.

29 Remove the condition (it should be: allow read, write;)



data with rules that
is access to it and
structured
[w the docs](#)

und
ore with

```
1  rules_version = '2';

2  service cloud.firestore {
3      match /databases/{database}/documents {
4
5          // This rule allows anyone with your Firestore database reference to vi
6          // and delete all data in your Firestore database. It is useful for get
7          // started, but it is configured to expire after 30 days because it
8          // leaves your app open to attackers. At that time, all client
9          // requests to your Firestore database will be denied.
10         //
11         // Make sure to write security rules for your app before that time, or
12         // all client requests to your Firestore database will be denied until
13         // your rules
14         match /{document=**} {
15             allow read, write: if request.time < timestamp.date(2023, 11, 16);
16         }
17     }
18 }
19 }
```

30 Click "Publish"

Indexes Usage Extensions

Looking for the Rules monitoring chart? It has been moved to the [Usage](#) tab for consolidated metric viewing. [Discard](#)

W
hed changes

10:34 AM

unpublished changes

Publish

Discard

```
1  rules_version = '2';

2  service cloud.firestore {
3      match /databases/{database}/documents {
4
5          // This rule allows anyone with your Firestore database reference to vi
6          // and delete all data in your Firestore database. It is useful for get
7          // started, but it is configured to expire after 30 days because it
8          // leaves your app open to attackers. At that time, all client
9          // requests to your Firestore database will be denied.
10         //
11         // Make sure to write security rules for your app before that time, or
12         // all client requests to your Firestore database will be denied until
13         // your rules
14         match /{document=**} {
15             allow read, write: if request.time < timestamp.date(2023, 11, 16);
16         }
17     }
18 }
19 }
```

31 Click here.

The screenshot shows the Firebase Cloud Firestore Rules page for a project named "myfirebaseproject". The "Rules" tab is selected. A message at the top right says, "Looking for the Rules monitoring chart? It has been moved to the Rules Overview page." Below this, there is a log entry from "Today · 10:35 AM" showing a star icon. To the right of the log is a code editor displaying the following Firestore security rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database} {
    match / {
      allow read, write: if true;
    }
  }
}
// This rule allows all reads and writes.
```

32 Click "Project settings"

The screenshot shows the Firebase Project Overview page for the same project "myfirebaseproject". The "Project settings" link in the top navigation bar is highlighted with a red circle. A dropdown menu is open over the "Project settings" link, showing three options: "Project settings", "Users and permissions", and "Usage and billing". The main content area displays the Cloud Firestore Rules page, identical to the one in the previous screenshot.

33 Click here to integrate Firebase into our project

The screenshot shows the Firebase console interface. At the top, there's a header with the text "Unspecified" and a pencil icon. Below the header, there's a section titled "in your project". Underneath this, there are several circular icons representing different platforms: "iOS+", "TV", "HTML", "Android", and "Web". The "Web" icon, which contains a code symbol "</>", is highlighted with a blue border and has a yellow circle around it. Below these icons, there's a "Delete project" button. The overall background is white with some light gray shadows.

34 Enter the name of your web app (can be anything) and click "Register App"

The screenshot shows a registration form titled "Add Firebase to your web app". The first step, "1 Register app", is selected. In the "App nickname" field, the text "My web app" is entered and is highlighted with a yellow circle. There is also a yellow circle around the "Register app" button at the bottom. Below the nickname field, there is a checkbox for "Also set up Firebase Hosting for this app." followed by a note: "Hosting can also be set up later. There is no cost to get started anytime." At the very bottom, there is another "Register app" button. The second step, "2 Add Firebase SDK", is partially visible at the bottom.

35

Save the information provided in firebaseConfig somewhere safe in your computer.

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyBg2TxDPcQZGQkv3keiKdl_upZDJxL9fms",
  authDomain: "myfirebaseproject-4603b.firebaseio.com",
  projectId: "myfirebaseproject-4603b",
  storageBucket: "myfirebaseproject-4603b.appspot.com",
  messagingSenderId: "768085886583",
  appId: "1:768085886583:web:f71466a9889aba30412872",
  measurementId: "G-HMM7CGJ6E8"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

36

Click "Service accounts"

The screenshot shows the Google Cloud Platform Project settings interface. On the left is a dark sidebar with a gear icon. The main area has a header 'myfirebaseproject ▾' and 'Project settings'. Below the header, there are tabs: General (selected), Cloud Messaging, Integrations, Service accounts (circled in orange), Data privacy, and Users and permissions. Under the 'Your project' section, there is a table with the following data:

Project name	myfirebaseproject
Project ID	myfirebaseproject-4603b
Project number	768085886583
Default GCP resource location	nam5 (us-central)
Web API Key	AlzaSyBg2TxDPcQZGQkv3keiKdl_upZDJxL9fms

At the bottom of the table, there is a section labeled 'Environment'.

37 Click "Generate new private key"

The screenshot shows the Firebase console interface. On the left, there's a sidebar with 'ice accounts' and a link to '3 service accounts'. The main area is titled 'Admin SDK configuration snippet' and shows code snippets for Node.js, Java, Python, and Go. The Node.js snippet is displayed:

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

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

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

A large blue button at the bottom of the snippet area is labeled 'Generate new private key', which is circled in orange.

38 Click "Generate key". Save the information somewhere safe in your computer.

A modal dialog box is centered on the screen. It has a title 'Create new private key' and a close button 'X'. Inside the dialog, a red warning message reads: 'Your private key gives access to your project's Firebase services. Keep it confidential and never store it in a public repository.' Below the message, a note says: 'Save the file securely, because your new key can't be recovered if lost.' At the bottom of the dialog are two buttons: 'Cancel' and a large blue 'Generate key' button, which is also circled in orange. In the background, the same Admin SDK configuration snippet from the previous step is visible.

39 You're done setting up Firebase!