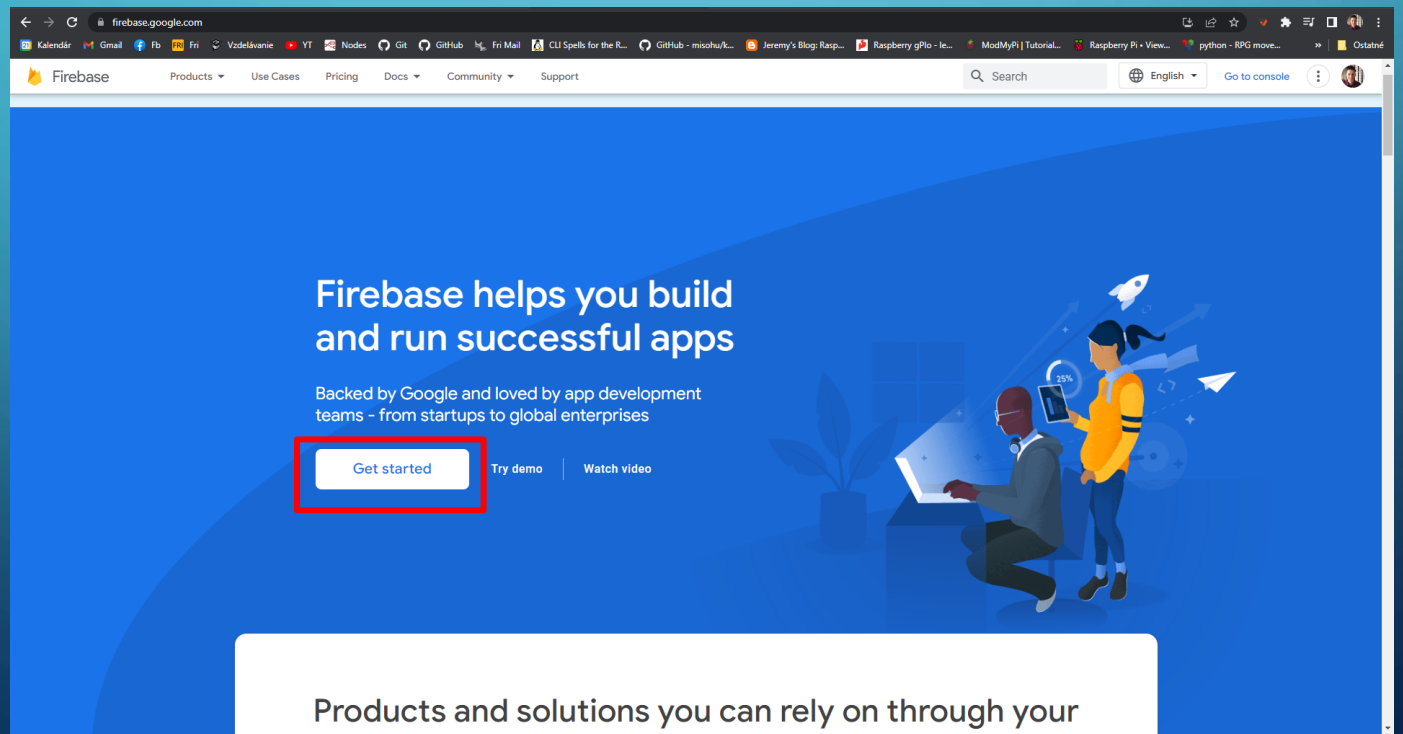
GOOGLE FIREBASE

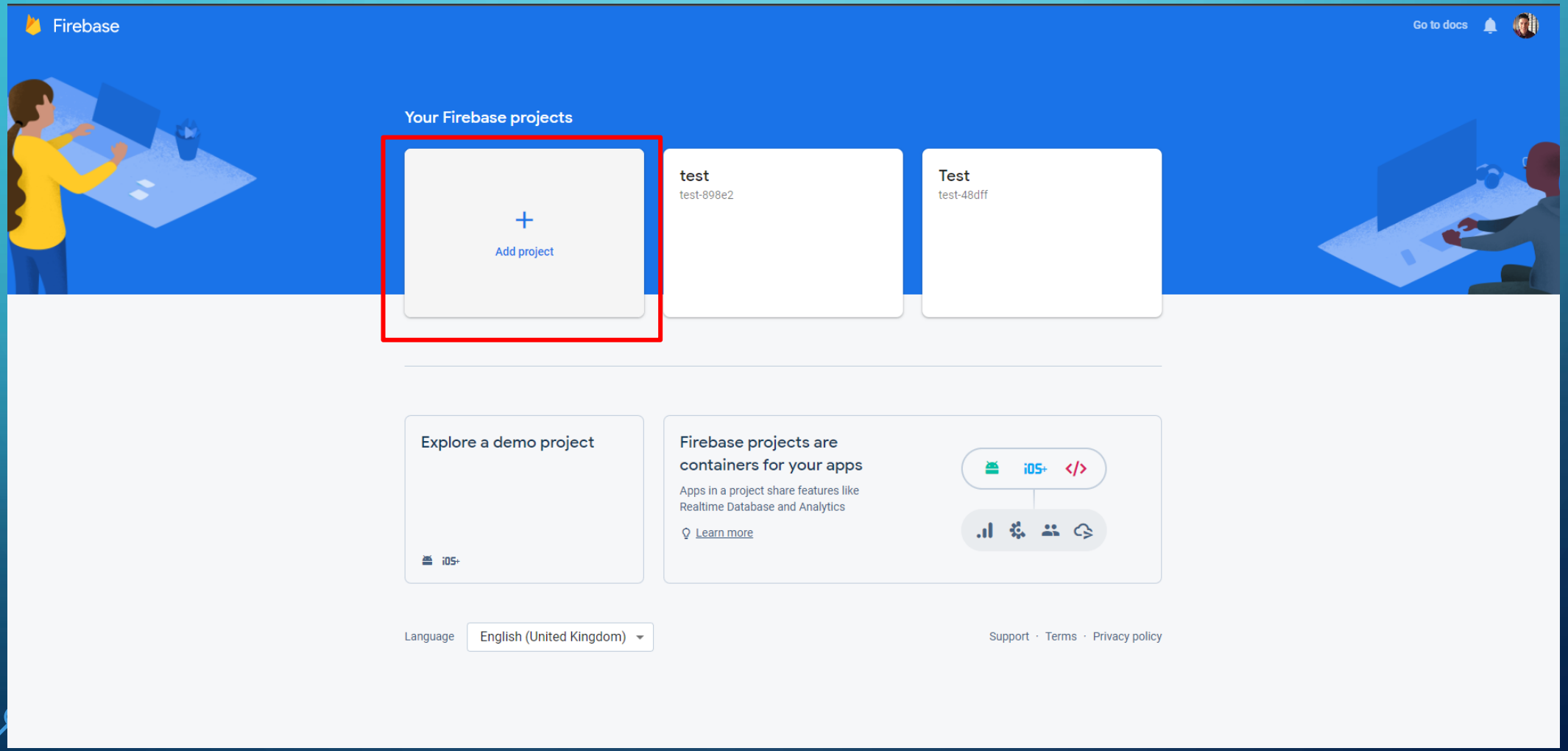
FRI UNIZA

GOOGLE FIREBASE

- https://firebase.google.com/
- Realtime NoSQL database.



PROJECT CREATION



PROJECT CREATION

× Create a project(Step 1 of 3)

Let's start with a name for your project

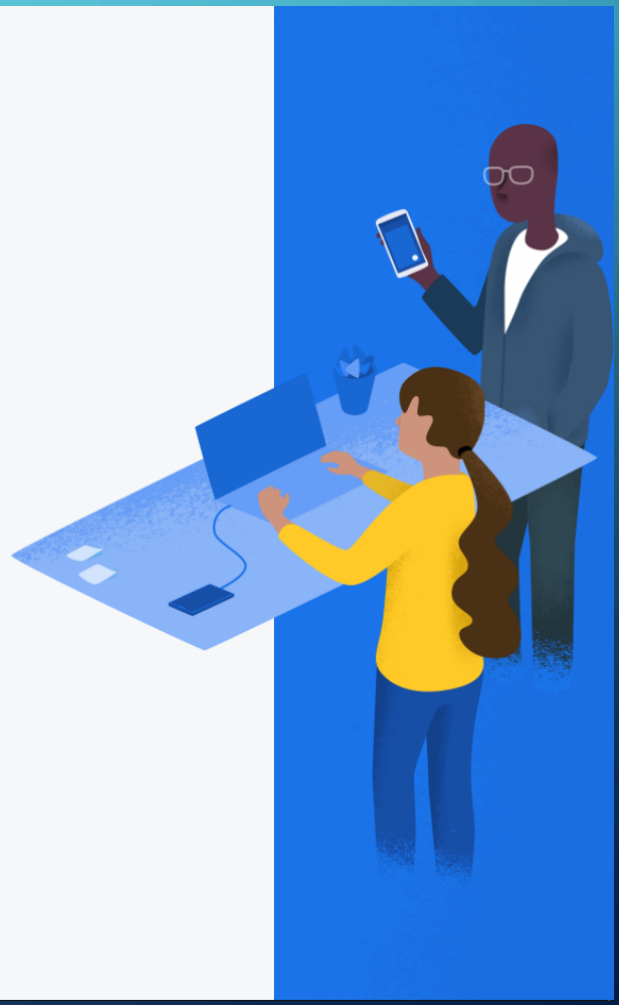
Project name

IoT-ESP32

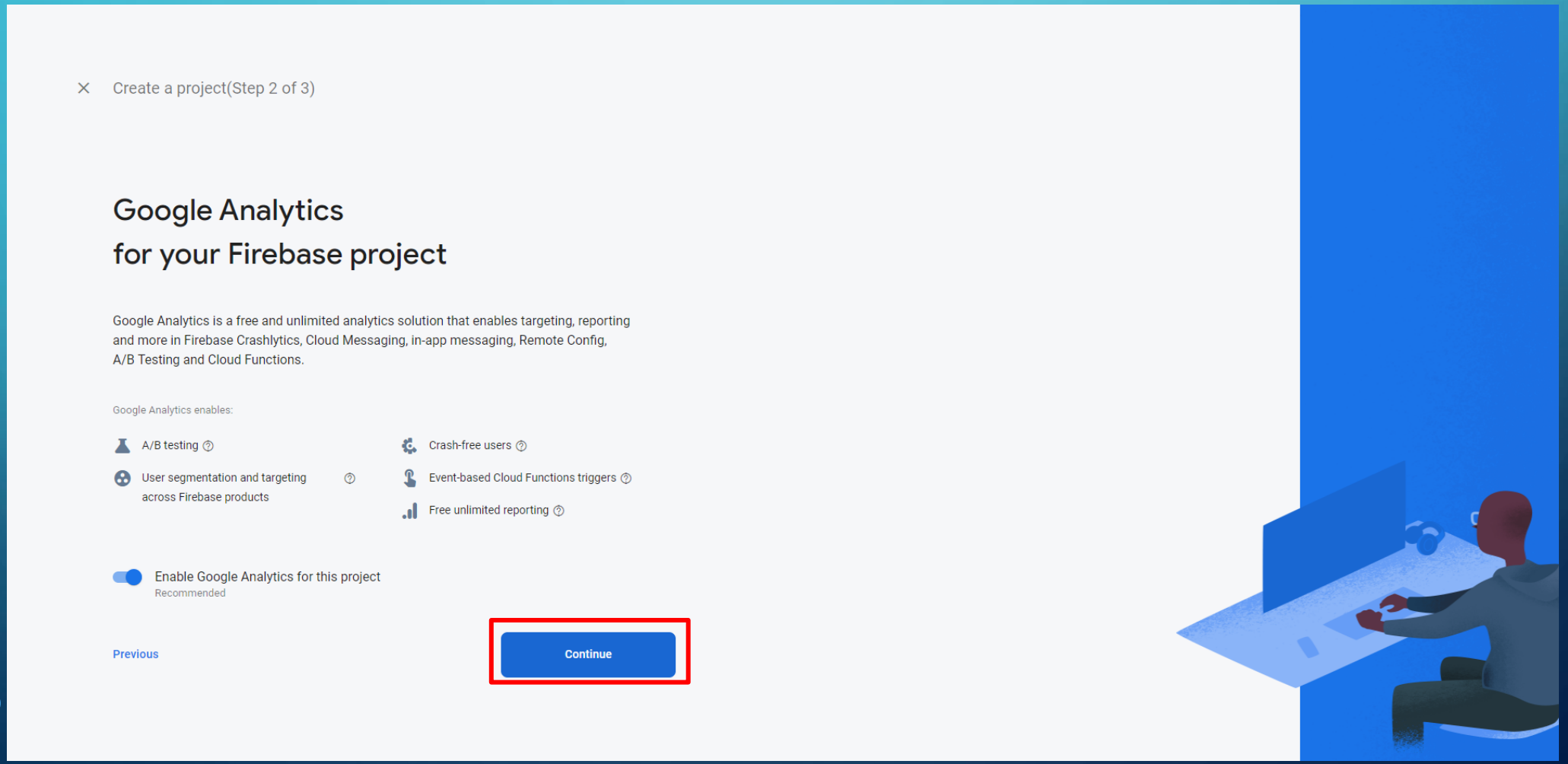
iot-esp32-5c3f0

I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft or profession.

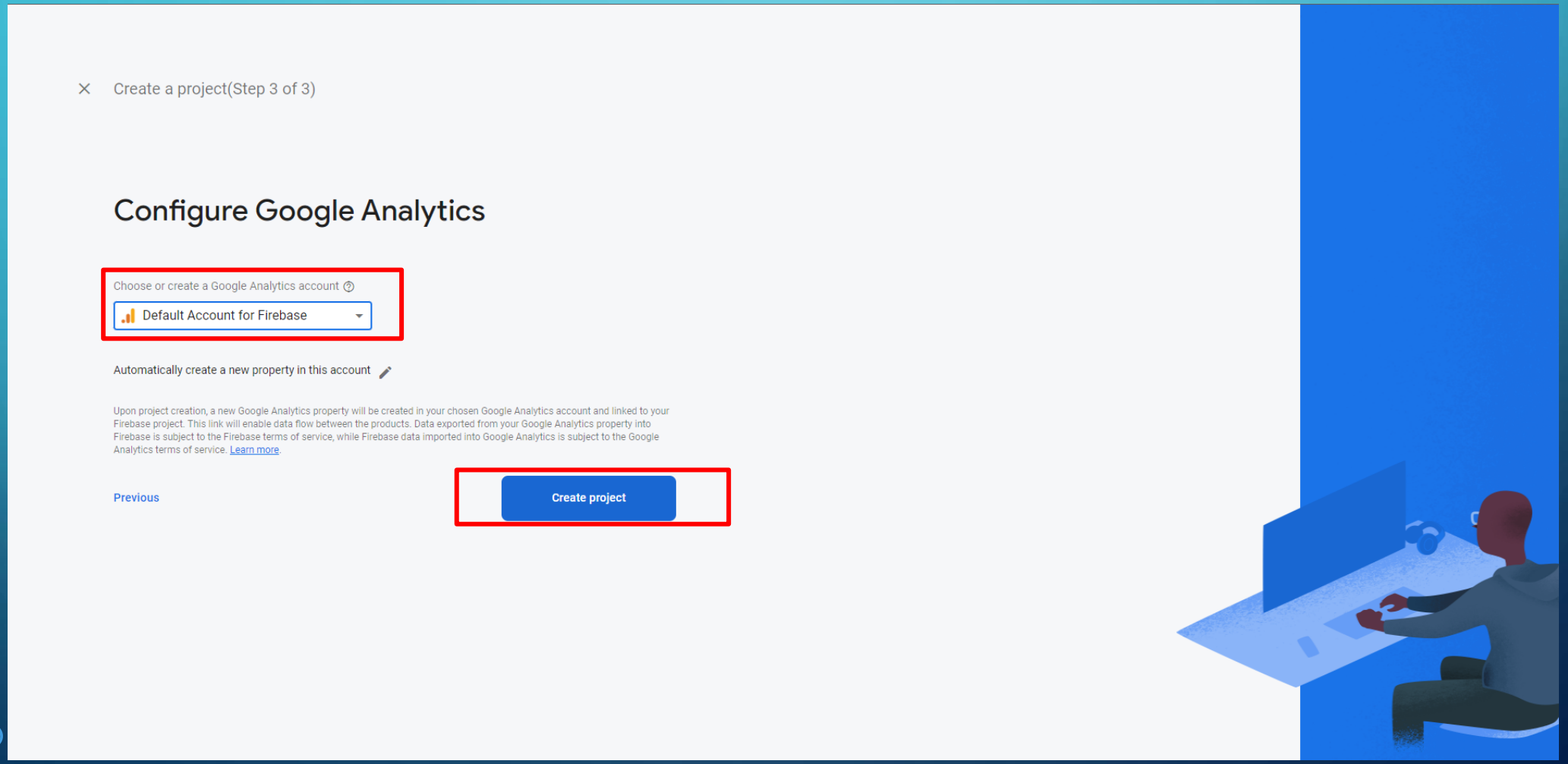
Continue



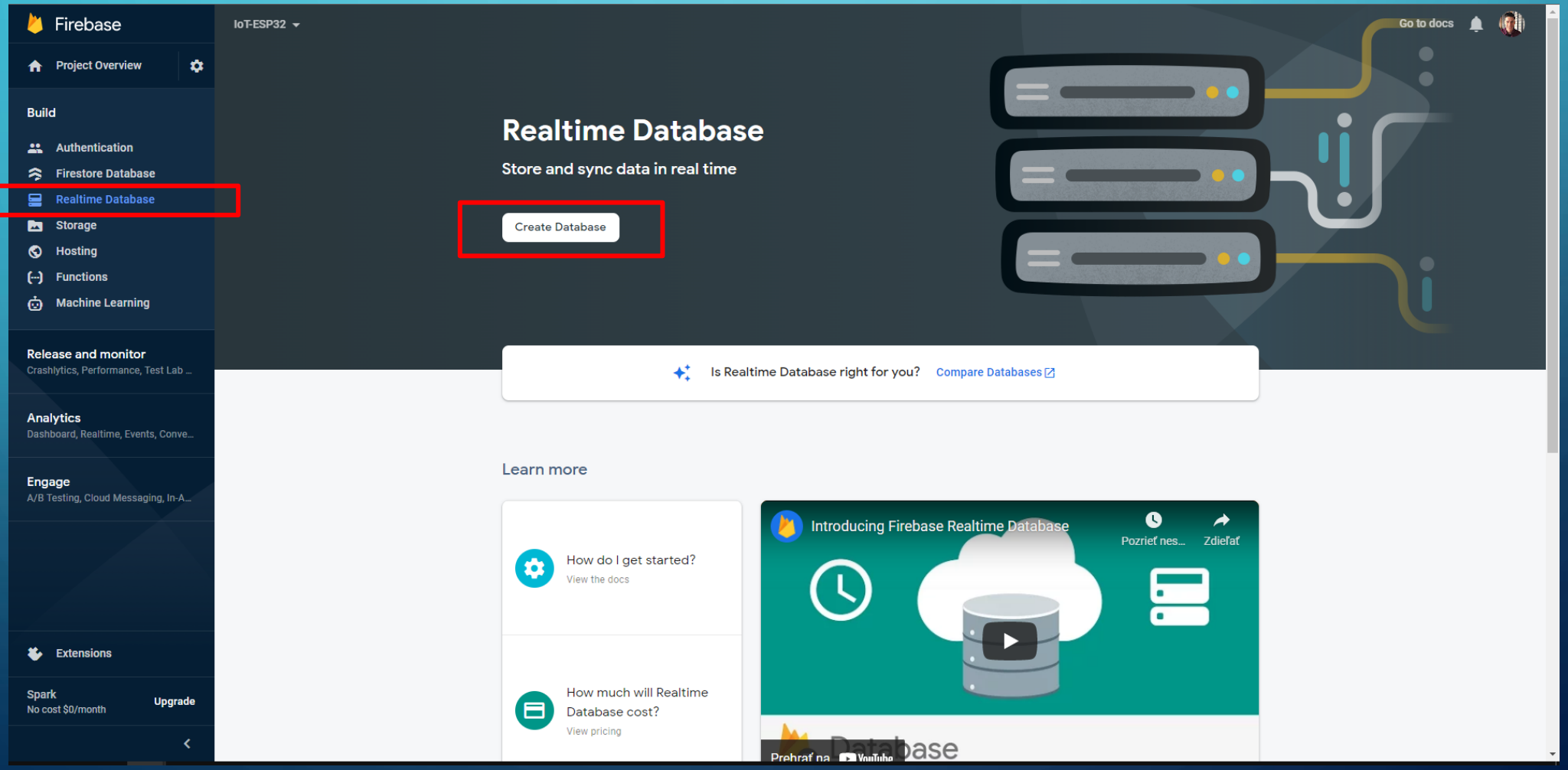
PROJECT CREATION



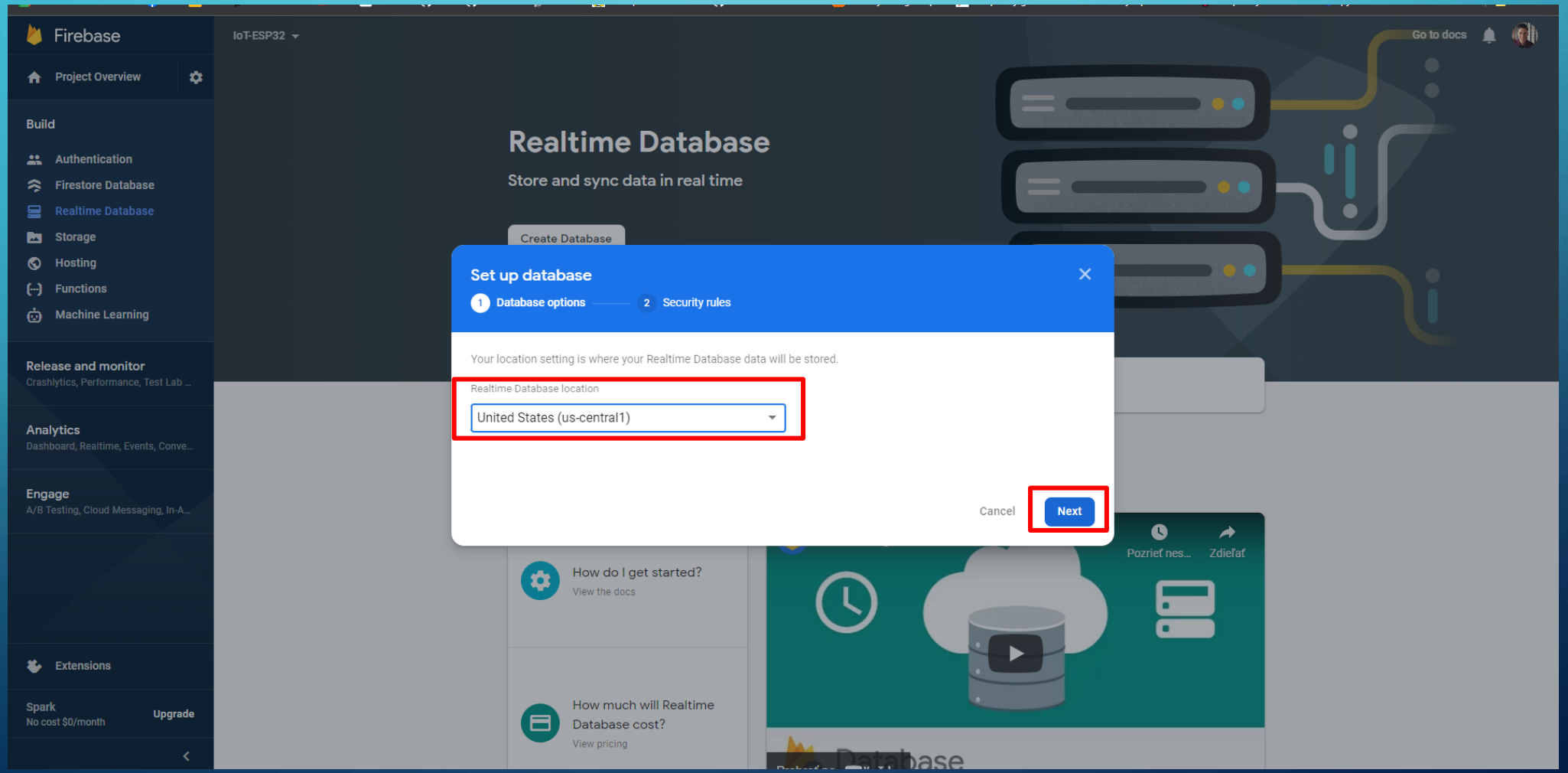
PROJECT CREATION



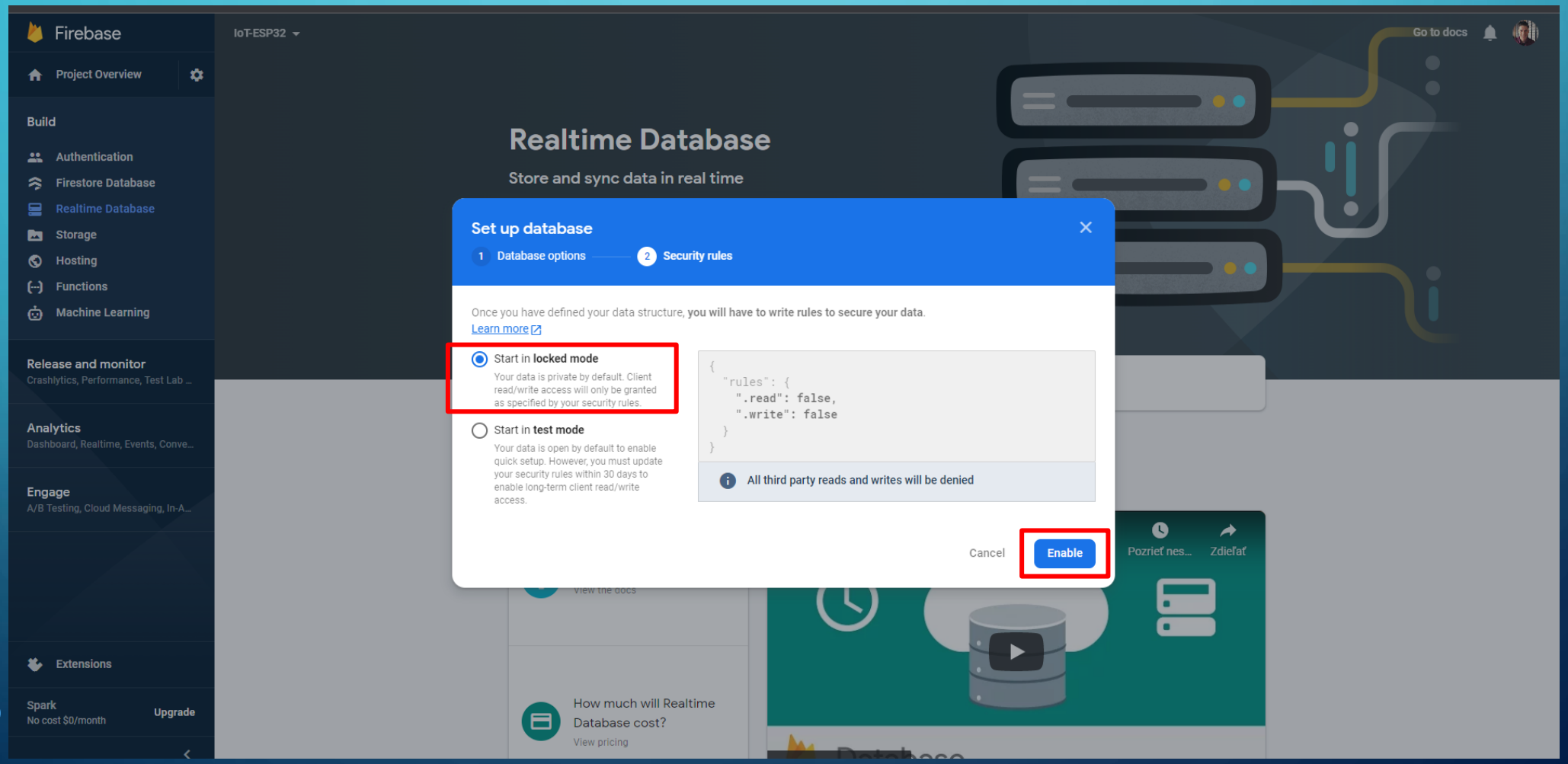
DATABASE CREATION



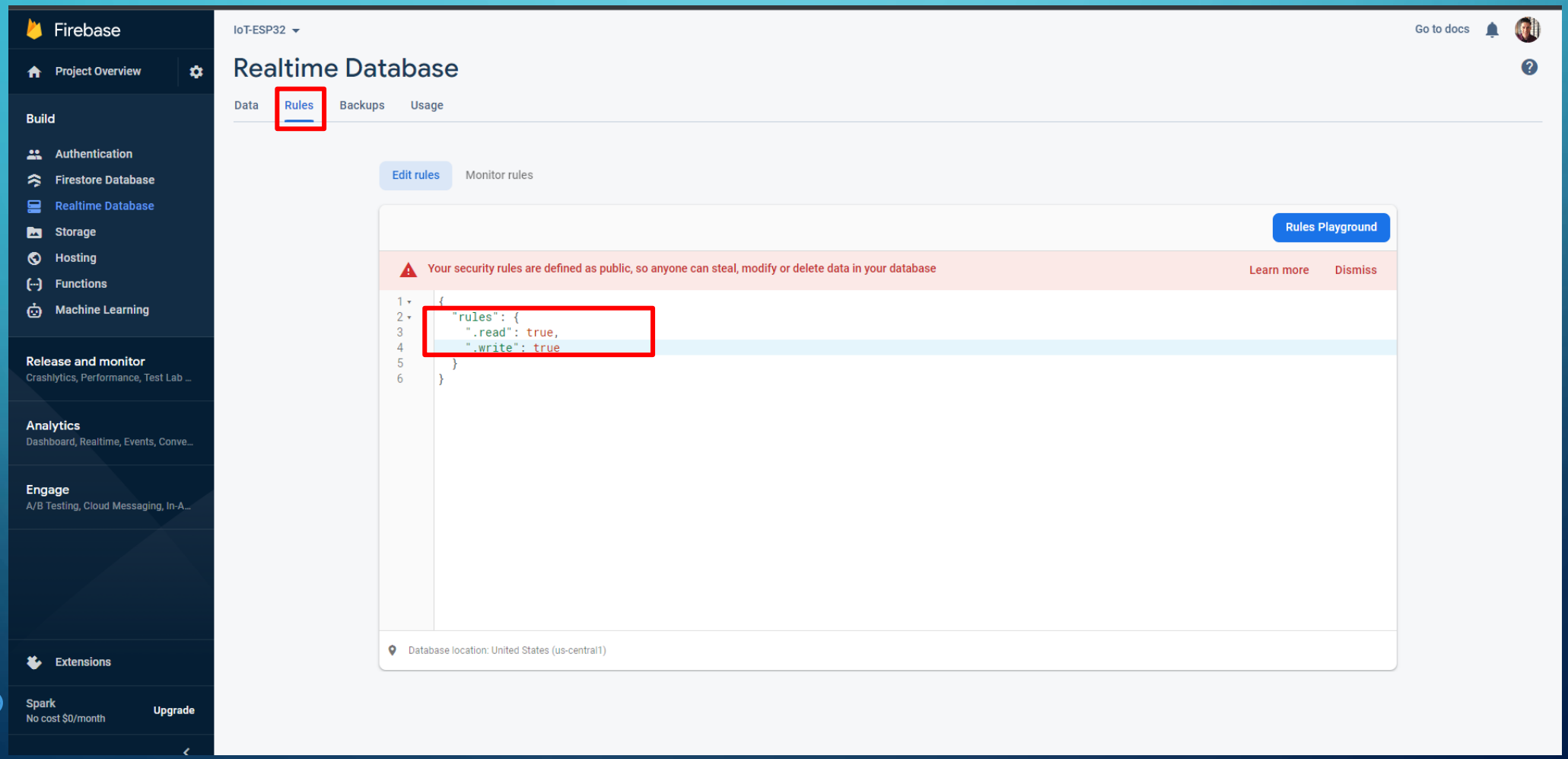
DATABASE CREATION



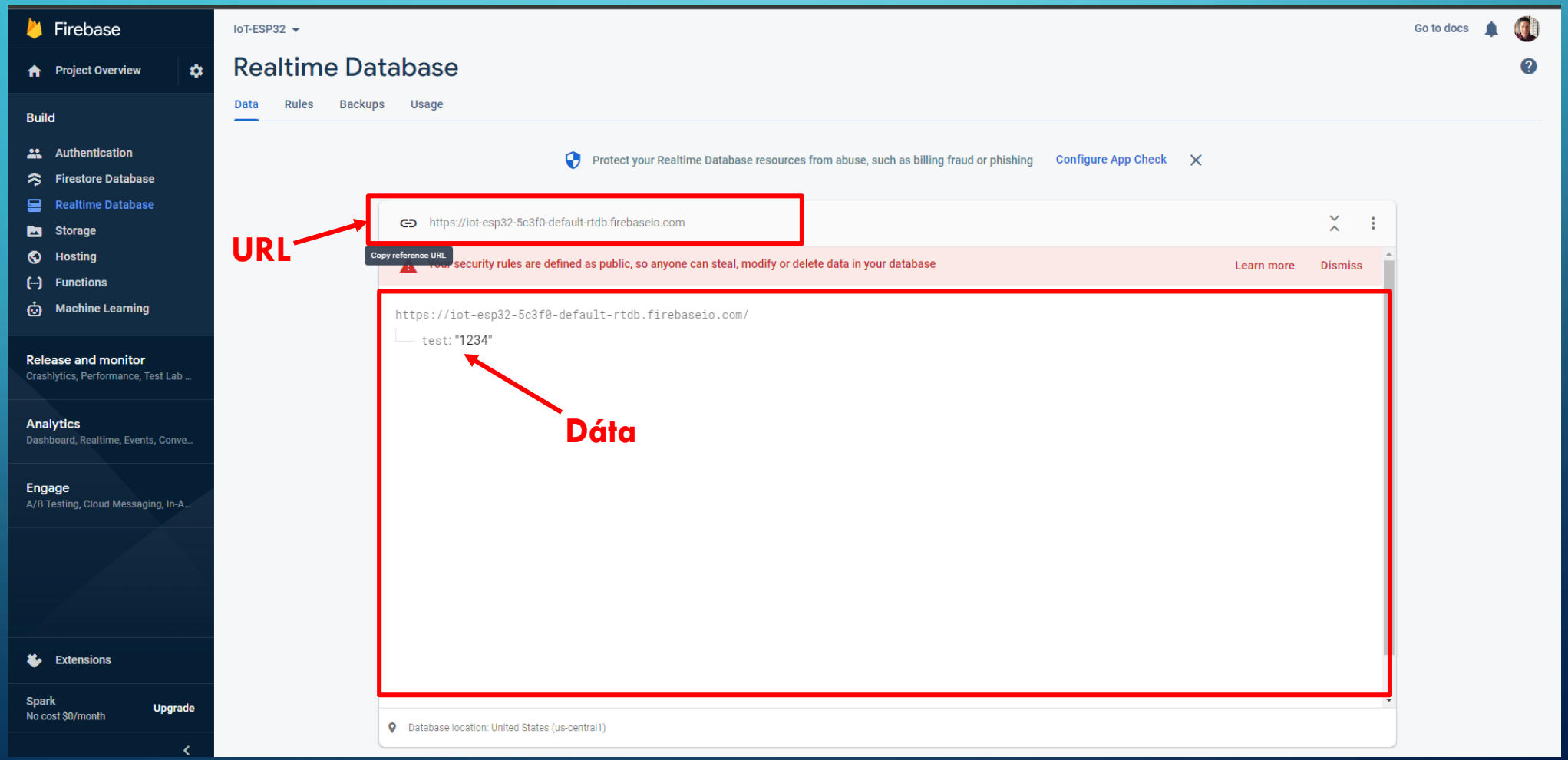
DATABASE CREATION



RULES


Firebase
Project Overview
Build
Authentication
Firestore Database
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release and monitor
Crashlytics, Performance, Test Lab ...
Analytics
Dashboard, Realtime, Events, Conve...
Engage
A/B Testing, Cloud Messaging, In-A...
Extensions
Spark
No cost $0/month
Upgrade
IoT-ESP32
Go to docs
Realtime Database
Data
Rules
Backups
Usage
Edit rules
Monitor rules
Rules Playground
Your security rules are defined as public, so anyone can steal, modify or delete data in your database
Learn more
Dismiss
{
"rules": {
".read": true,
".write": true
}
}
Database location: United States (us-central1)

URL & DATA


Firebase
Project Overview
Build
Authentication
Firestore Database
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release and monitor
Crashlytics, Performance, Test Lab ...
Analytics
Dashboard, Realtime, Events, Conve...
Engage
A/B Testing, Cloud Messaging, In-A...
Extensions
Spark
No cost $0/month
Upgrade
IoT-ESP32
Go to docs
Realtime Database
Data
Rules
Backups
Usage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
URL
https://iot-esp32-5c3f0-default-rtdb.firebaseio.com
Copy reference URL
Learn more
Dismiss
https://iot-esp32-5c3f0-default-rtdb.firebaseio.com/
test: "1234"
Dáta
Database location: United States (us-central1)

MICROPYTHON-FIREBASE LIBRARY

- https://github.com/ckoever/micropython-firebase-realtime-database
- Commands:
  - get,
  - getfile,
  - put,
  - patch,
  - addto,
  - delete.

Download : ufirebase.py and save on device (esp32).

GUIDE

1. Upload library to ESP32.
2. Connect ESP32 to AP with internet connection.
3. Import library: import ufirebase as firebase.
4. Set URL : firebase.setURL(„URL“).
5. Data upload: firebase.put(„test“, „1234“, bg=False).

   firebase.addto („vals“, {„temp“:20, „hum“:50}, bg=False).
6. Data download: firebase.get(„test“, „testVariable“, bg=False).

   print(firebase.testVariable).

Optional run in the background with the keyword bg.

TASKS

1. Create your own firebase database.
2. Upload measured values (temperature & humidity & photoresistor) to database in periodic interval.
3. Upload meassured values to database when button is pressed.
4. Make pairs. If the button on the first board is pressed, the state of the led on the second board will change. Use a database for synchronization.