

Building the App

Prerequisites

Flutter Installed

Install the Flutter framework found in the dev docs.

<https://docs.flutter.dev/install>

Ensure the flutter is installed by running `flutter` on the terminal/powershell. If help options appear, then flutter is successfully installed.

Cloned Repo

Clone the repo with:

```
git clone https://github.com/Baby-Words-Tracker/Baby_Words.git
```

Android

Building an APK

The Android app is straightforward to build.

First, navigate to the `baby_words_tracker` folder using:

```
cd baby_words_tracker/
```

Then to build, run the command:

```
flutter build apk
```

This will then generate an APK file typically found in:

`baby_words_tracker/build/app/outputs/flutter-apk/app-release.apk`. The location may differ depending on build settings, but will the last output after the build command will also display the file path.

Installing the APK

Connect an Android device with Android Debug (ADB) enabled and in developer mode.

Run the command `flutter install` and follow the prompts to install the app.

You should now be able to run the app on a real Android device

Build, Install, and Debug

You can also live debug the app on real hardware by doing the follow:

1. Connect an Android device and enable ADB

2. List the device name using `flutter devices`
3. Debug the app using `flutter run -d <device-name>`

For more detailed instructions, refer to the dev docs found here: <https://docs.flutter.dev/deployment/android>

iOS

iOS is admittedly difficult to build, due to Apple's strict security and signing protocols.

For more detailed instructions, refer to the dev docs found here: <https://docs.flutter.dev/deployment/ios>

Deploying Admin Console (Web App)

This tutorial is a written derivation from this YouTube guide, with minor tweaks due to several sources being out of date.

<https://www.youtube.com/watch?v=LSt1af5YZcQ>

From Scratch

In the case that the original instance is inaccessible, you will have to deploy the console from scratch.

We will be using GCloud's Compute Engine to deploy our app.

Creating the Virtual Machine

Select the WordBuds project on GCloud and head to <https://console.cloud.google.com/compute>

On the left sidebar go to Virtual machines > VM instances.

If prompted under the Compute Engine API, then enable it and reload the page.

Select **Create instance** on the top ribbon.

Feel free to change the instance name. The only settings you need to change are under **Networking > Firewall**. Tick the boxes that say **Allow HTTP traffic** and **Allow HTTPS traffic**.

Create the instance and wait for it to load.

Setting up the VM

Once the VM is created, simply click on the VM and click the SSH button. This will open a browser window that will access to the terminal.

Install the following:

Git

```
sudo apt install git
```

Docker GPG key

```
sudo apt update

sudo apt install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Add the repo to Apt sources

```
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
```

install docker and docker compose with

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

Docker should now be installed. Test it with the command: `docker compose`

For more information on install Docker see the dev docs here: <https://docs.docker.com/engine/install/debian>

Running the web app

Clone the repo with

```
git clone https://github.com/Baby-Words-Tracker/Baby_Words.git
```

Navigate to `Baby_Words/webapp` and run the container with

```
docker compose up --build -d
```

NPM Error

Sometimes the node packages will be out of date. If this happens just run these commands in the `Baby_Words/webapp` directory:

```
sudo apt install npm  
  
npm i  
  
docker compose up
```

Updating the Web App

Updates to the production web app are done by manually pulling updates from the `main` branch of the repo.

Simply just perform `git pull` in the `Baby_Words/` directory.

Run `docker compose up --build -d` in the `Baby_Words/webapp`

Deploying your Web App to the Wordbuds Domain

The production admin web app can be accessed at <http://wordbuds.info:8080>

This section is a derivative of this YouTube video. <https://www.youtube.com/watch?v=yOVgnNbCneU>

Giving your VM a Static IP

Head to your GCloud VPC Network > IP Address:

<https://console.cloud.google.com/networking/addresses>

You should see your named instance. On the dotted menu on the far right and select **Promote to static IP address**

Your instance should now be accessible on the open web using the static IP address

Connecting the Static IP to the Domain

Your GCloud domains can be found here: <https://console.cloud.google.com/networking/addresses>

Use the instruction found in dev docs to connect your IP to a static IP address.

<https://docs.cloud.google.com/dns/docs/tutorials/create-domain-tutorial#set-up-domain>

Deploying Firebase Node Functions

The Firebase project is found under `Baby_Words/firebase-project/functions`. Most of the currently used firebase functions are in `index.js`. This mainly consists of the push notification function, which is ran server-side.

These Firebase functions can be deployed on any device with the repo pulled.

Ensuring the node modules are installed

Run the commands in the `Baby_Words/firebase-project` directory:

```
npm i
cd functions
npm i
```

These install the node packages inside both folders.

Deploy command

Simply run this command inside `Baby_Words/firebase-project/functions`

```
node index.js
```

Deploying Firebase Python Functions

This section of the code is deprecated, but if you do need to deploy Python functions, here it is.

Installation

Ensure you have python 3.10 installed. Make a virtual environment with Python 3.10

```
cd Baby_Words/firebase-project/functions_python
python3.10 -m venv venv
source venv/bin/activate
cd python_functions
pip install -r requirements.txt
```

Deployment

```
python3.10 main.py
```

