

Reference:<https://codelabs.developers.google.com/codelabs/automl-vision-edge-in-mlkit/#1>

## A. Execute automl vision and timeseries forecasting models:

## B. end2end deployment of a vision model using automl to mobile device

### 1. Set up and Downloading Training code in firebase console

Train and deploy on-device image classification model with AutoML Vision in ML Kit

⌚ 55 mins remaining Language S

The screenshot shows the 'Setup' step of a codelab. On the left, a vertical sidebar lists steps 1 through 7: 1. Introduction, 2. Setup (which is selected), 3. Prepare training dataset, 4. Train a model, 5. Use the model in mobile apps, 6. (Optional) Use remote model, and 7. Congratulations!. Below the sidebar, there's a 'Report a mistake' button and a 'Back' button. The main content area is titled '2. Setup' and contains instructions to 'Download the code and training dataset'. It includes a 'Download ZIP' button and a note that there are Android and iOS apps. A green callout box provides more details about the download: 'There is an Android app and an iOS app provided in this codelab that have similar functionalities. You can choose either platform that you are more familiar with to proceed.' Another note says 'The flower\_photos.zip file in the archive is the training dataset that you will use in step 3: Prepare training dataset'. Below this, there are sections for 'Create Firebase console Project' (with steps to go to the console and create a new project) and 'Setup the Android app' (with steps to add the app to the Firebase project and download the config file). A 'Next' button is located at the bottom right.

### 2. Creating project

 Create a project (Step 1 of 4)

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

Project name

 **subarna-hw1p2**

[subarna-hw1p2](#)

 [sjsu.edu](#)

[Continue](#)

3. Next step

[X](#) Create a project (Step 2 of 4)

## A few things to remember when adding Firebase to a Google Cloud project

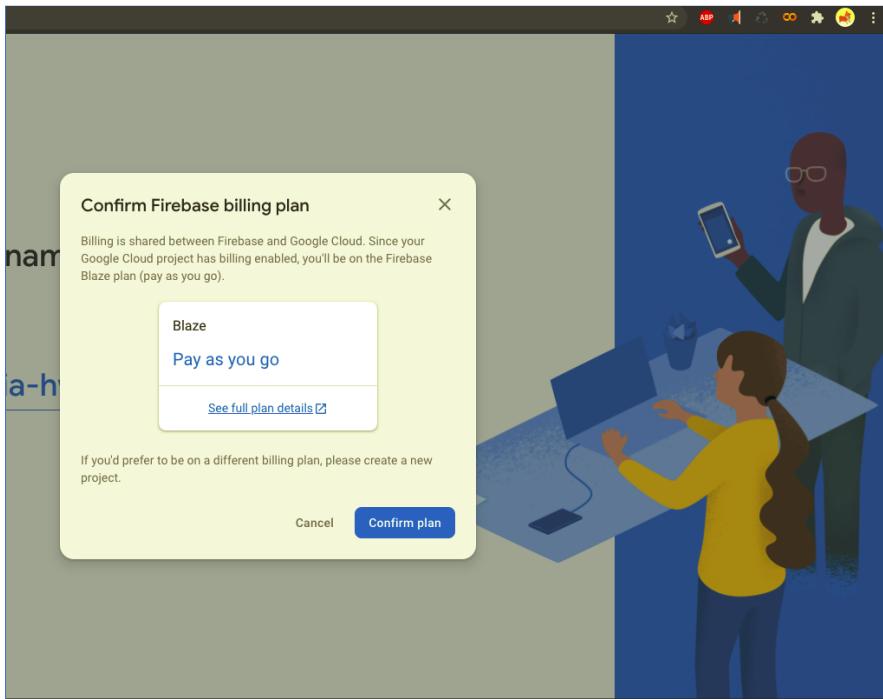


- ">\$ Google Cloud and Firebase billing for your project will be shared [Learn more](#)
- 🔒 User roles and permissions for your project will be shared [Learn more](#)
- ⚠ Deleting a Firebase project deletes the Google Cloud project too, and all contained resources
- ⚠ You won't be able to undo this, though you'll be able to manually disable most Firebase services

Not what you wanted?  
[Go back and create a new project instead](#)

[Continue](#)

### 4. Confirming Bill:



### 5. Created console project

[console.firebaseio.google.com/u/0/?pli=1](#)

X Create a project (Step 2 of 4)

## A few things to remember when adding Firebase to a Google Cloud project



Google Cloud and Firebase billing for your project will be shared [Learn more](#)

Since your project has billing enabled, you'll be on [Firebase's Blaze plan \(pay-as-you-go\)](#), and Firebase line items will appear on your Cloud bill each cycle

User roles and permissions for your project will be shared [Learn more](#)

Deleting a Firebase project deletes the Google Cloud project too, and all contained resources

You won't be able to undo this, though you'll be able to manually disable most Firebase services

Not what you wanted?  
[Go back and create a new project instead](#)

Continue

## 6. Google analytics and project ready :

[console.firebaseio.google.com/u/0/?pli=1](#)

X Create a project (Step 3 of 4)

## Google Analytics for your Firebase project

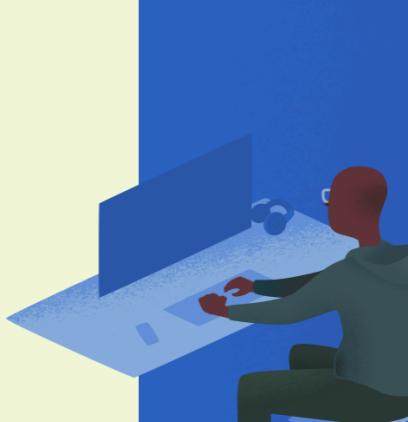
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Predicting user behavior
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project  
Recommended

Previous Continue



Create a project (Step 4 of 4)

## Configure Google Analytics

Analytics location United States

Data sharing settings and Google Analytics terms

Use the default settings for sharing Google Analytics data. [Learn more](#)

- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google to enable Technical Support
- ✓ Share your Analytics data with Google Account Specialists

I accept the [Measurement Controller-Controller Data Protection terms](#) and acknowledge I am subject to the [EU End User Consent Policy](#). This is required when sharing Google Analytics data to improve Google Products and Services. [Learn more](#)

I accept the [Google Analytics terms](#)

Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#).

[Previous](#) [Add Firebase](#)

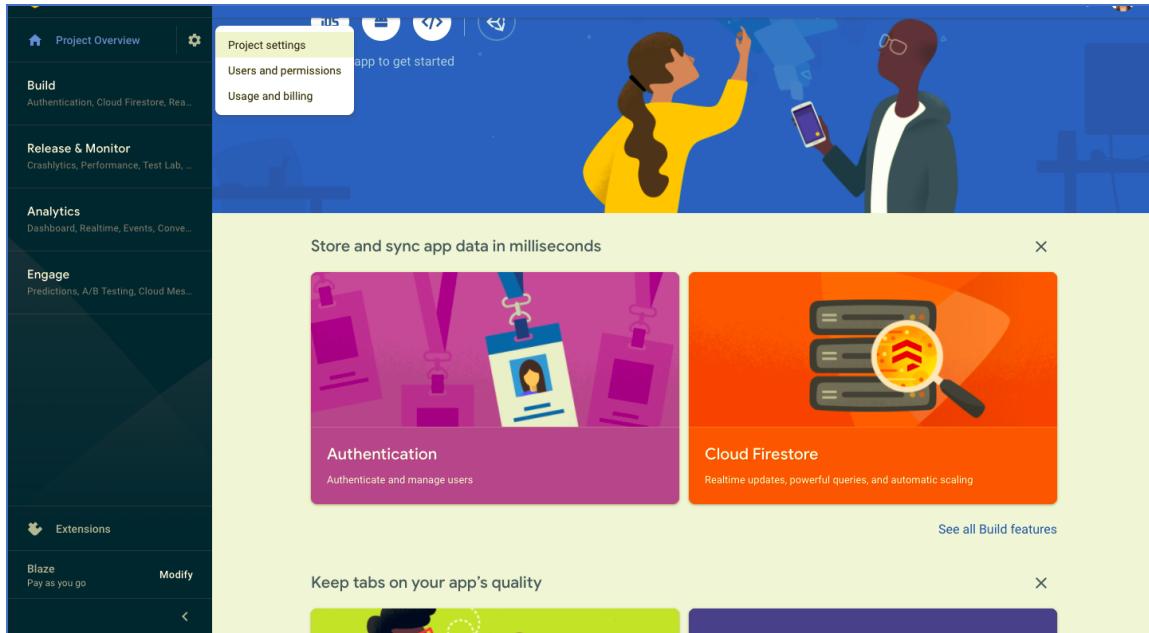


Your new project is ready

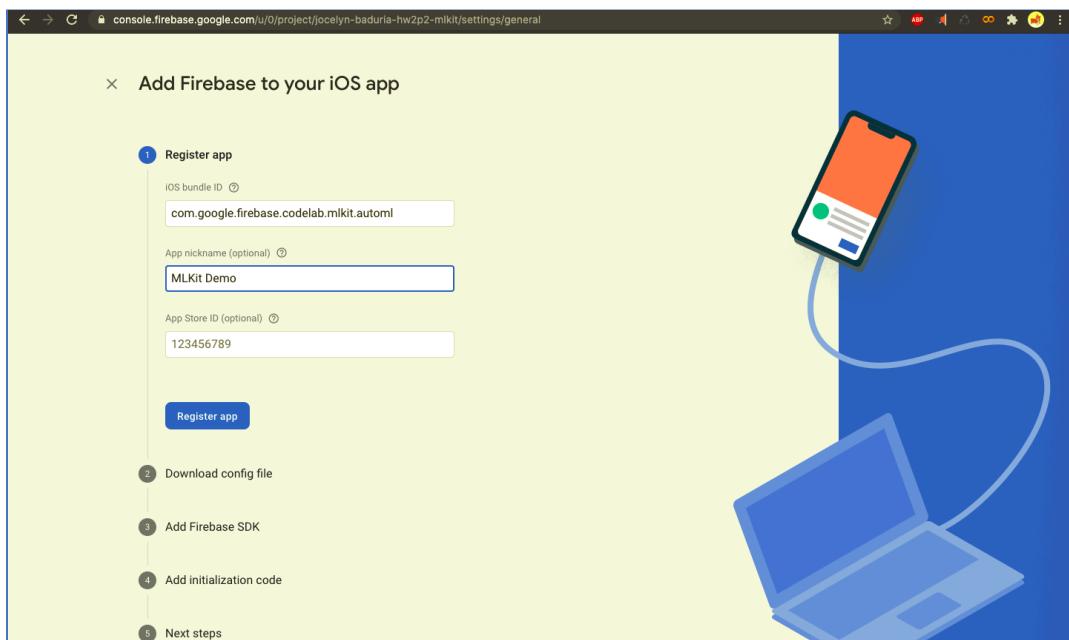
[Continue](#)



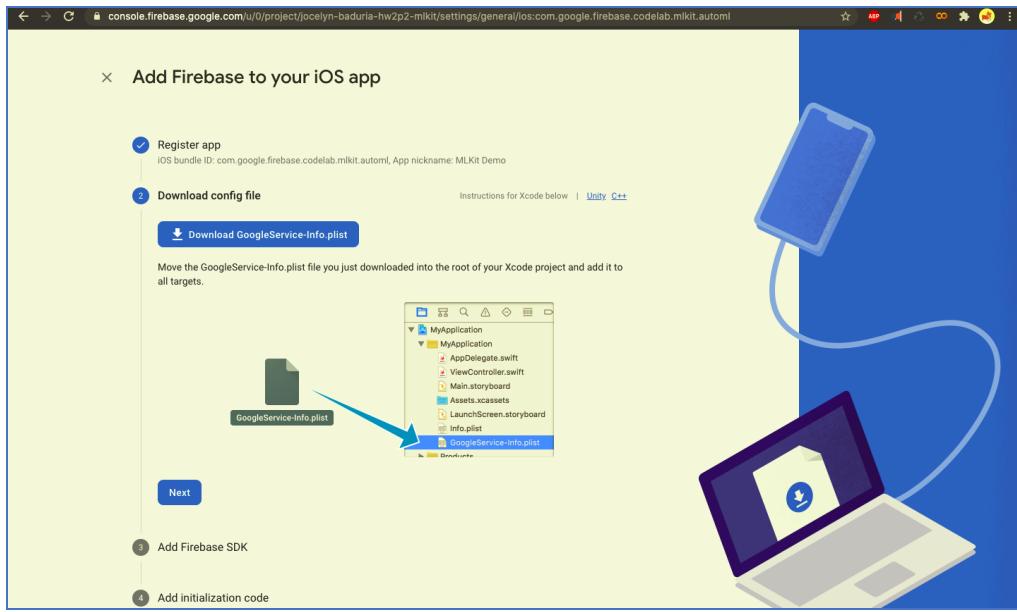
## 8. Setup iOS app:



## 9. Registered app:



## 10. Added firebase to iOS app:



## 11. Setup in IDE

Key	Type	Value
CLIENT_ID	String	115318948861-1g03hldh21h4vhjafg6v070cmpp21uq.apps.googleusercontent.com
REVERSED_CLIENT_ID	String	com.googleusercontent.apps.115318948861-1g03hldh21h4vhjafg6v070cmpp21uq
API_KEY	String	Alza5yAW-H6k9QVOOzg6v1sM2hmdrpY8GleCgg
GCM_SENDER_ID	String	115318948861
PLIST_VERSION	String	1
BUNDLE_ID	String	com.google.firebaseio.codename.mlkit.automa
PROJECT_ID	String	jocelyn-baduria-hw2p2-mlkit
STORAGE_BUCKET	String	jocelyn-baduria-hw2p2-mlkit.appspot.com
IS_IOS_ENABLED	Boolean	0
IS_ANALYTICS_ENABLED	Boolean	0
IS_APPINVITE_ENABLED	Boolean	1
IS_GCM_ENABLED	Boolean	1
IS_SIGNIN_ENABLED	Boolean	1
GOOGLE_APP_ID	String	1:115318948861:ios:88447481c81406dfe04f94

## 12. Added firebase SDK:

**× Add Firebase to your iOS app**



- 1 Register app  
iOS bundle ID: com.google.firebaseio.codelab.mlkit.automl, App nickname: MLKit Demo
- 2 Download config file
- 3 Add Firebase SDK

Instructions for CocoaPods | [Download ZIP](#) [Unity](#) [C++](#)

Google services use [CocoaPods](#) to install and manage dependencies. Open a terminal window and navigate to the location of the Xcode project for your app.

Create a Podfile if you don't have one:

```
$ pod init
```

Open your Podfile and add:

```
# add the Firebase pod for Google Analytics
pod 'Firebase/Analytics'
# add pods for any other desired Firebase products
# https://firebase.google.com/docs/ios/setup#available-pods
```

Save the file and run:

```
$ pod install
```

This creates an .xcworkspace file for your app. Use this file for all future development on your application.

### 13. Installed cocoapods in local and added 'Firebase/Analytics' to POD file:

```
automi-vision-edge-in-mlkit-master — -bash — 99x24
Olens-Macbook-Air:Downloads jocelynbaduria$ cd automl-vision-edge-in-mlkit-master
Olens-Macbook-Air:automl-vision-edge-in-mlkit-master jocelynbaduria$ pod init
-bash: pod: command not found

mlkit-automl — vim Podfile — 107x15
source 'https://github.com/CocoaPods/Specs.git'

platform :ios, '9.0'
use_frameworks!

target 'MLVisionExample' do
  pod 'FirebaseMLVision', '0.16.0'
  pod 'FirebaseMLCommon', '0.16.0'
  pod 'FirebaseMLVisionAutoML', '0.16.0'
  pod 'Firebase/Analytics'
end
~
```

```

mlkit-automl — git · pod install — 107x15
+ update      Update outdated project dependencies and create new Podfile.lock

tions:

--allow-root    Allows CocoaPods to run as root
--silent        Show nothing
--version       Show the version of the tool
--verbose       Show more debugging information
--no-ansi       Show output without ANSI codes
--help          Show help banner of specified command
ens-Macbook-Air:mlkit-automl jocelynbaduria$ vim Podfile
ens-Macbook-Air:mlkit-automl jocelynbaduria$ pod install
alyzing dependencies

```

## 14 Target dependencies and installed:

```

mlkit-automl — vim PodFile — 107x15
Source 'https://github.com/CocoaPods/Specs.git'

platform :ios, '9.0'
use_frameworks!

target 'MLVisionExample' do
  pod 'FirebaseMLVision', '0.16.0'
  pod 'FirebaseMLCommon', '0.16.0'
  pod 'FirebaseMLVisionAutoML', '0.16.0'
  pod 'Firebase/Analytics'
  pod 'Firebase/Auth'
  pod 'Firebase/Firestore'
end
~
"Podfile" 13L, 306C

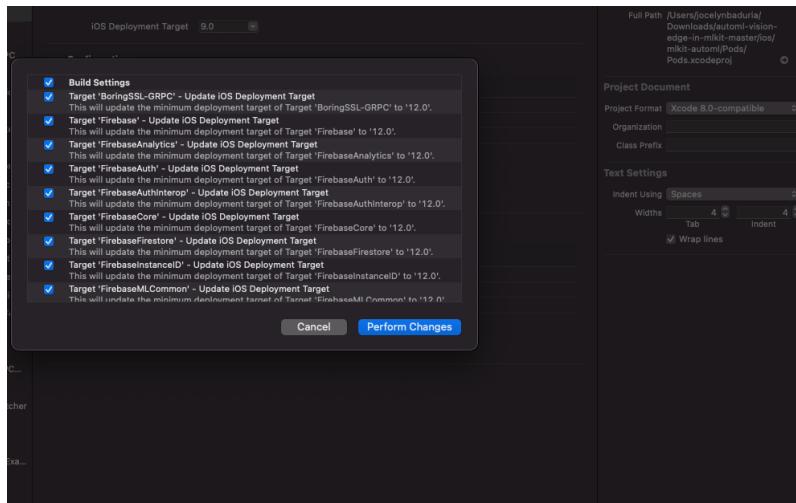
```

```

Analyzing dependencies
Downloading dependencies
Installing BoringSSL-GRPC (0.0.3)
Installing Firebase 6.0.0
Installing FirebaseAuth (6.0.0)
Installing FirebaseAuthInterop (1.1.0)
Installing FirebaseFirestore (1.3.2)
Installing gRPC-C++ (0.0.9)

```

## 15. Added initialization code



**4 Add initialization code**

To connect Firebase when your app starts up, add the initialization code below to your main `AppDelegate` class.

Swift  Objective-C

```
import UIKit
import Firebase

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication,
                    didFinishLaunchingWithOptions launchOptions:
                        [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure()
        return true
    }
}
```

Previous **Next**

**Next steps**

## 16. Next steps continued with Firebase Console:

**× Add Firebase to your iOS app**

- Register app  
iOS bundle ID: com.google.firebaseio.codelab.mlkit.automl, App nickname: MLKit Demo
- Download config file
- Add Firebase SDK
- Add initialization code
- Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

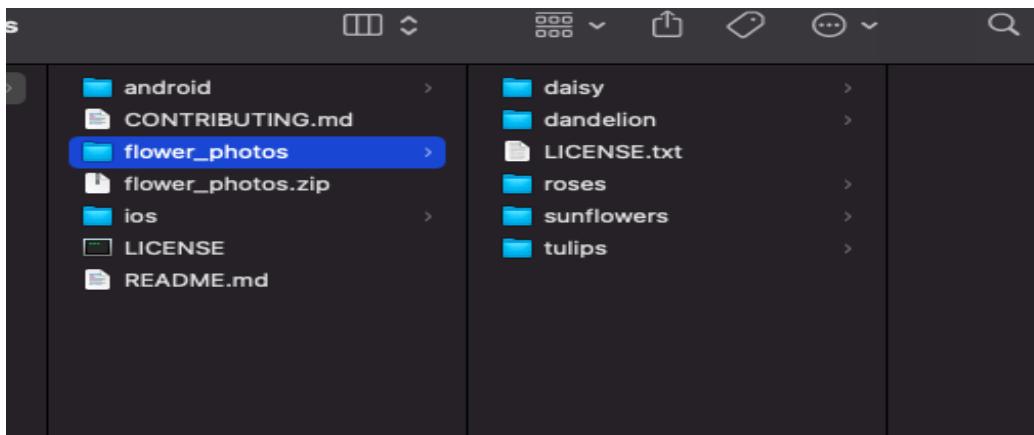
You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous **Continue to console**

The screenshot shows the Firebase console's "SDK setup and configuration" page for an iOS app named "MLKit Demo". The app's package name is listed as com.google.firebaseio.codelab.mlkit.automl. The page includes fields for App ID, Encoded App ID, App nickname, Bundle ID, App Store ID, and Team ID, each with a corresponding edit icon. A "See SDK instructions" button and a "GoogleService-Info.plist" download link are also present.

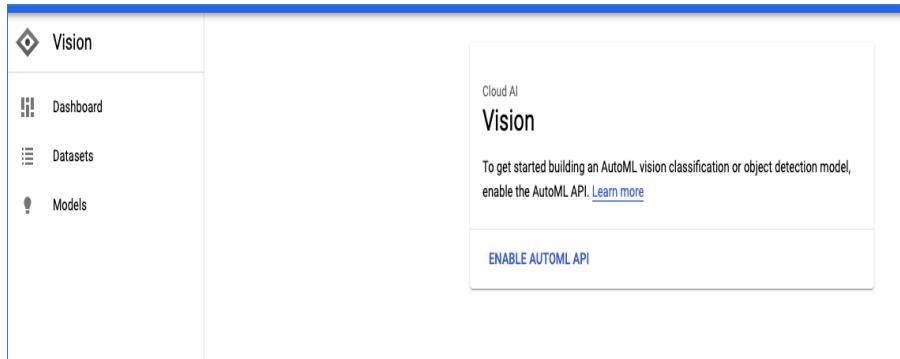
## 17. Training with Flower Data with 6 categories:



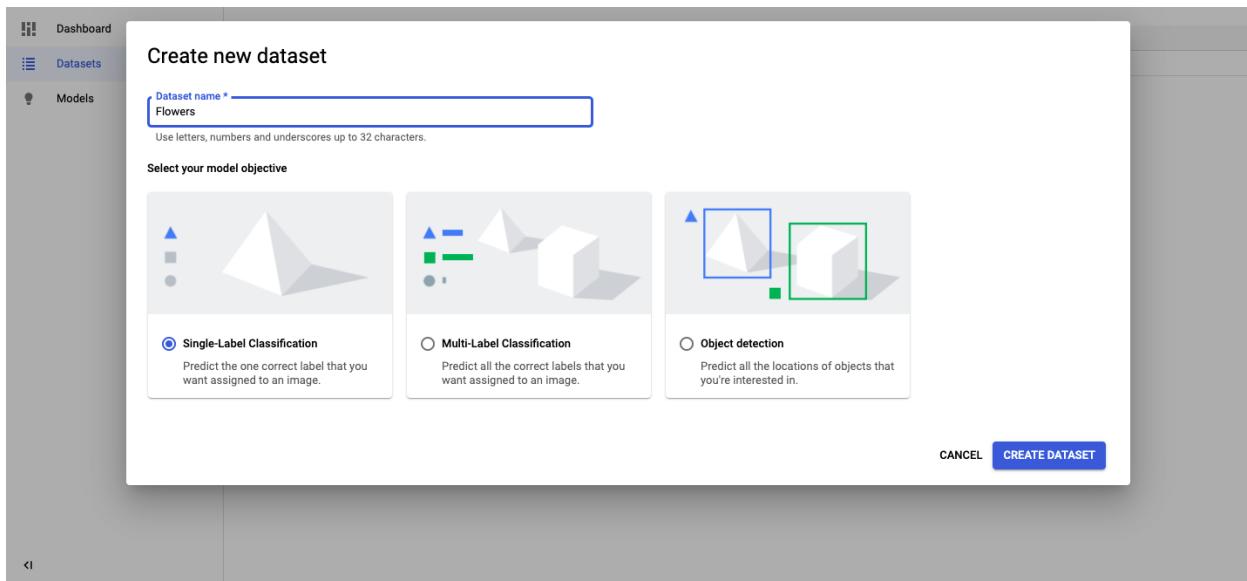
## 18. AutoML Training:

The screenshot shows the Google Cloud Platform console's "Machine Learning" section. The sidebar on the left includes "Project Overview", "Build" (with sub-options like Authentication, Cloud Firestore, etc.), "Release & Monitor", "Analytics", "Engage", and "Extensions". The "Machine Learning" tab is selected under "Build". The main content area displays a "Machine Learning" heading, sub-tabs for "APIs", "Custom", and "AutoML" (which is underlined to indicate it is active). Below this, there is a callout box with an icon of a smartphone displaying a camera viewfinder. The text inside the box reads: "Train high-quality custom machine learning models with minimum effort and machine learning expertise". A note below states: "AutoML Vision Edge model training is now only available in the Google Cloud console." with a "Learn more" link. A "Get started with Cloud AutoML" button is at the bottom.

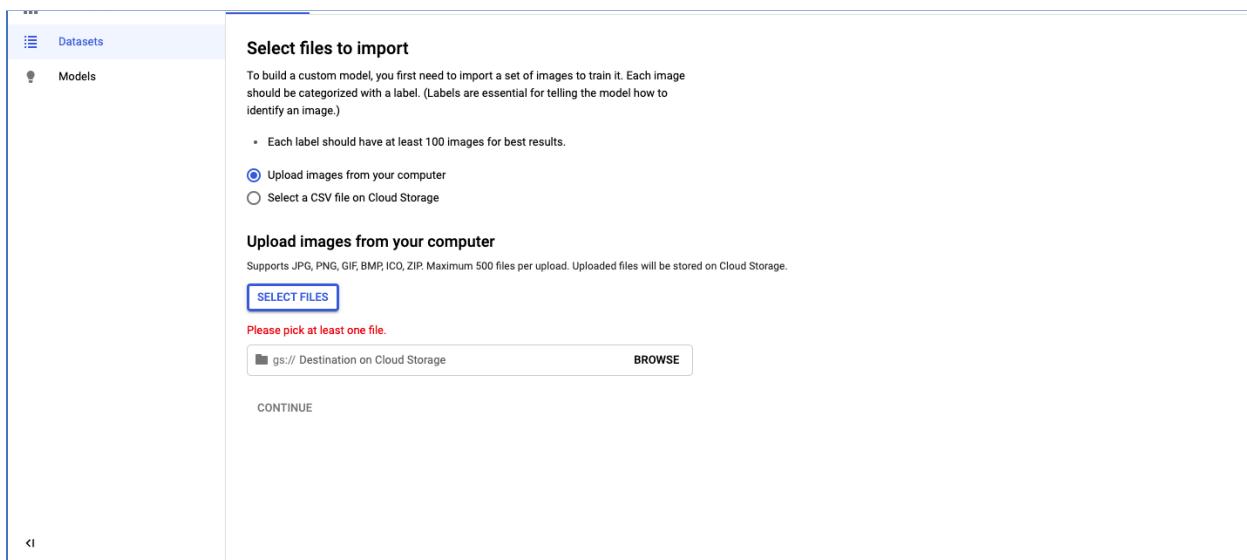
## 19. Enabled AutoML Vision:



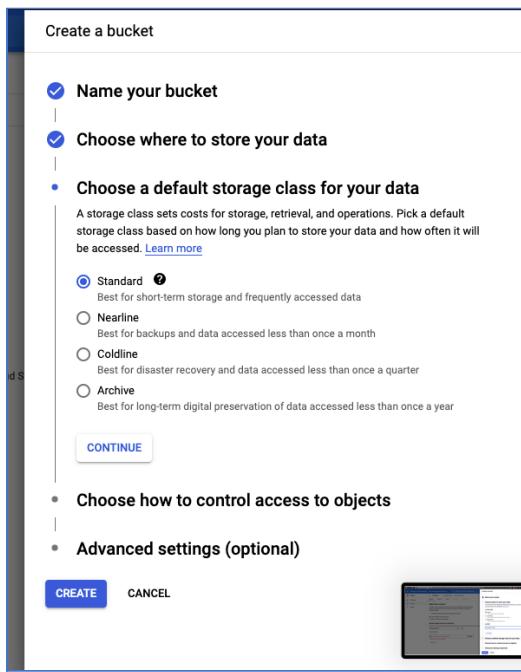
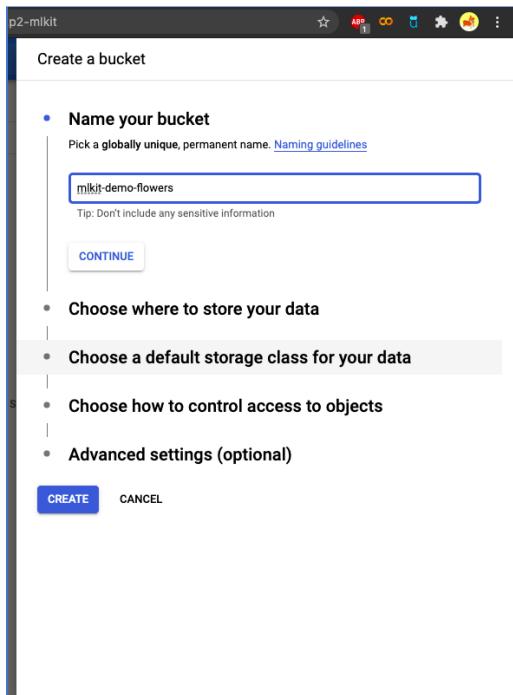
## 20. Added flower dataset by uploading it to AutoML Vision:

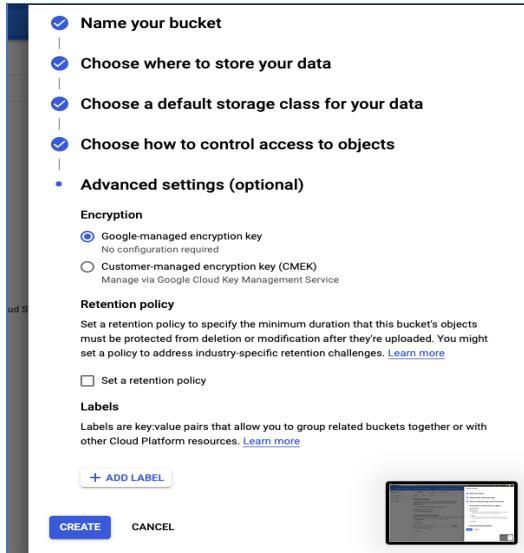


## 21. Imported the data and created bucket:



## 22. Bucket Setup





### 23. Flower Dataset after uploaded :

Label	Count
daisy	200
dandelion	200
roses	200
sunflowers	200
tulips	200

### 24. Training the Model Using AutoML Vision:

**Vision**

**Flowers**

**LABEL STATS** **EXPORT DATA**

IMPORT IMAGES **TRAIN** EVALUATE TEST & USE

You have enough images to start training

Unlabeled images aren't used. Your dataset will be automatically split into Train, Validation, and Test sets.

Ideally, each label should have at least 10 images. Fewer images often result in inaccurate models.

Labels	Images	Train	Validation	Test
daisy	200	160	40	0
dandelion	200	160	40	0
roses	200	160	40	0
sunflowers	200	160	40	0
tulips	200	160	40	0

**CONTINUE**

**Define your model**

Model name \* Flowers\_20210228014559

Cloud hosted Host your model on Google Cloud for online predictions

Edge Download your model for offline/mobile use

**Optimize model for**

**Set a node hour budget**

START TRAINING CANCEL

## 25. Model Optimization:

**Train new model**

**Define your model**

**Optimize model for**

Goal	Package size	Accuracy	Latency for Google Pixel 2
<input type="radio"/> Higher accuracy	6 MB	Higher	360 ms
<input type="radio"/> Best trade-off	3.2 MB	Medium	150 ms
<input checked="" type="radio"/> Faster predictions	0.6 MB	Lower	56 ms

Please note that prediction latency estimates are for guidance only. Actual latency will depend on your network connectivity.

**CONTINUE**

**Set a node hour budget**

START TRAINING CANCEL

## 26. Set the budget for node training hour: 1

Train new model

- Define your model
- Optimize model for
- 3 Set a node hour budget**

Enter the maximum number of node hours you want to spend training your model.

We recommend using [3 node hours](#) for your dataset. However, you can train for as little as 1 node hours. You may also eligible to train with free node hours. [Pricing guide](#)

Set your budget \*  node hours

Estimated completion date: Feb 28, 2021 3 PM  
GMT-8

**START TRAINING**   **CANCEL**

## 27. Trained Model:

Vision	Datasets	NEW DATASET	⋮											
Dashboard														
Datasets	<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Total images</th><th>Labeled images</th><th>Last updated</th><th>Status</th></tr> </thead> <tbody> <tr> <td><span style="color: green;">✓</span> Flowers</td><td>Single-Label Classification</td><td>1,000</td><td>1,000</td><td>Feb 28, 2021, 12:18:48 PM</td><td>Success: Training model</td></tr> </tbody> </table>	Name	Type	Total images	Labeled images	Last updated	Status	<span style="color: green;">✓</span> Flowers	Single-Label Classification	1,000	1,000	Feb 28, 2021, 12:18:48 PM	Success: Training model	
Name	Type	Total images	Labeled images	Last updated	Status									
<span style="color: green;">✓</span> Flowers	Single-Label Classification	1,000	1,000	Feb 28, 2021, 12:18:48 PM	Success: Training model									
Models			⋮											

IMPORT IMAGES TRAIN EVALUATE TEST & USE

**TRAIN NEW MODEL**

**Flowers\_20210228014559**

Average precision 0.971

Precision\* 95.51%  
Recall\* 85%

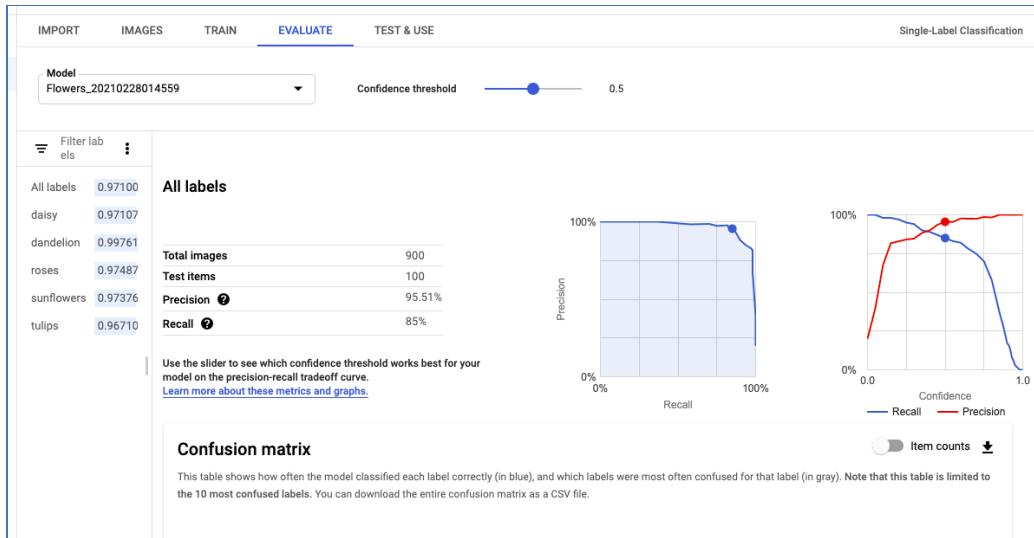
\* Using a score threshold of 0.5

Model ID <span style="color: green;">ICN8459477537364377600</span>	Created <span style="color: green;">Feb 28, 2021, 1:47:54 PM</span>
Base model	None
Data	1,000 images
Model type	Mobile Low Latency
Train cost	0.632 node hours
Deployment state	Not deployed

[SEE FULL EVALUATION](#)

[RESUME TRAINING](#)

## 28. Model Evaluation:



## 29. Confusion Matrix:

Confusion matrix					
True Label	Predicted Label				
	dandelion	tulips	sunflowers	daisy	roses
dandelion	95%	-	5%	-	-
tulips	-	90%	5%	-	5%
sunflowers	-	-	95%	-	5%
daisy	-	-	5%	95%	-
roses	-	10%	5%	10%	75%

## 30. Exporting the TFLight package:

**Export TF Lite package**

The Tensorflow Lite (.tflite) format allows you to run your model on mobile and embedded devices.

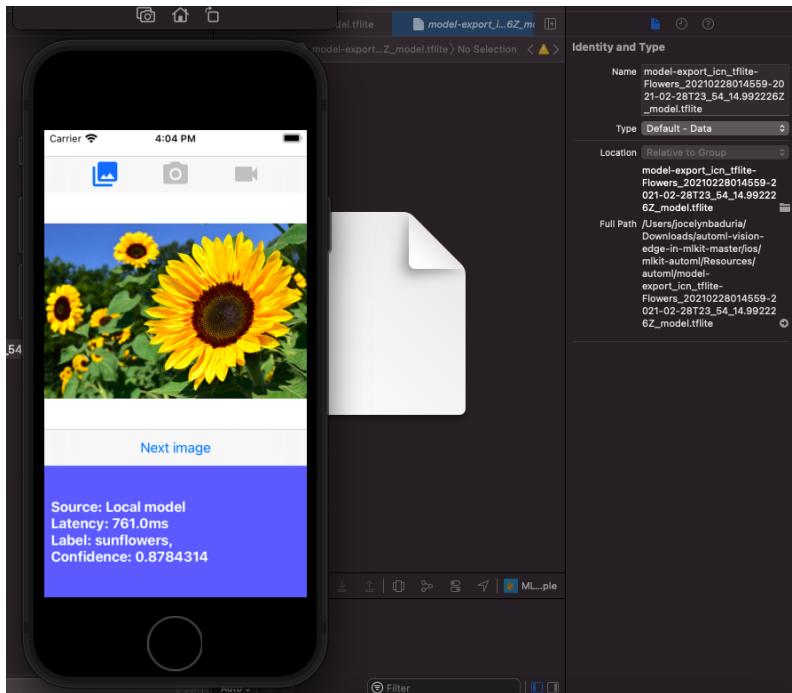
1. Export your model as a TF Lite package.  
Destination folder on Cloud Storage  
 mlkit-demo-flowers [BROWSE](#)

[EXPORT](#) [OPEN IN GCS](#)

2. After your model finishes exporting, you can copy your package to your computer using this command:  
`$ gsutil cp -r gs://mlkit-demo-flowers ./download_dir`

3. Follow the quickstart to learn how to implement your model on your device.  
[Android quickstart](#) [iOS quickstart](#)

### 31. Successfully Running the MLVisionFLEExample.app using iOS App Simulator



### 32. From app:

