1. Saat menggunakan linux, terdapat beberapa perintah yang bisa digunakan. Jelaskan dan berikan contoh beberapa perintah dasar yang sering digunakan pada linux berikut ini :   
   A. chmod command untuk menentukan hak akses file/folder

B. mv command untuk memindahkan file

C. rmdir command untuk menghapus folder kosong

D. chown command untuk mengganti kepemilikan/owner dari file atau folder

1. Buatlah sebuah file docker-compose dengan ketentuan sebagai berikut ini :

* Nama services mysql-db
* Image mysql:8.0
* ENV : MYSQL\_USER, MYSQL\_PASSWORD, MYSQL\_DB
* Volume (directory menyesuaikan)
* PORT (bebas)
* Expose (bebas)

1. Berikut ini adalah code konfigurasi saat set up projek dengan firebase console. Terdapat beberapa baris code yang tidak tepat dan kurang. Silakan perbaiki dan dilengkapi :

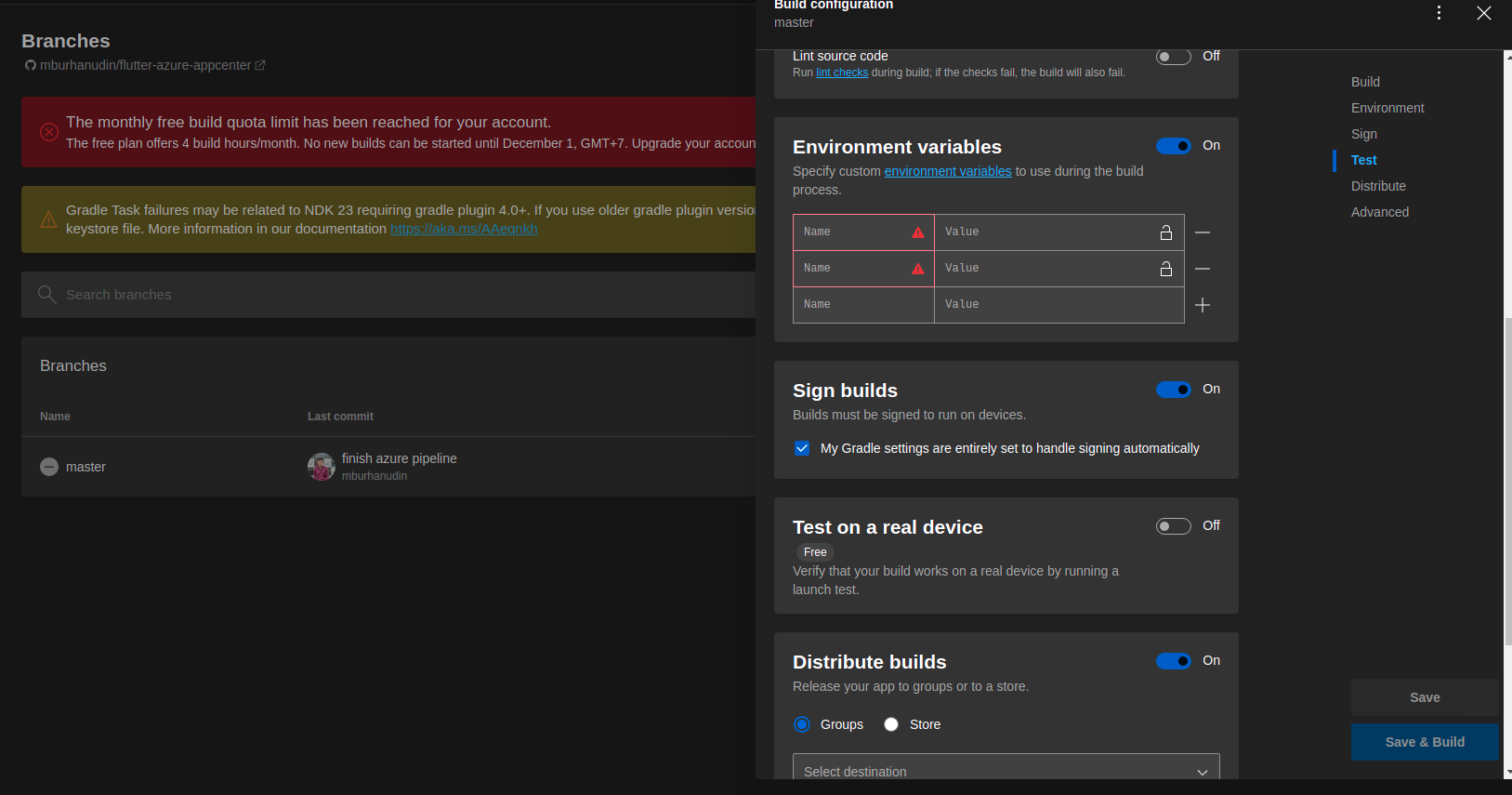
|  |
| --- |
| def localProperties = new Properties()  def localPropertiesFile = rootProject.file('local.properties')  if (localPropertiesFile.exists()) {  localPropertiesFile.withReader('UTF-8') { reader ->  localProperties.load(reader)  }  }  def flutterRoot = localProperties.getProperty('flutter.sdk')  if (flutterRoot == null) {  throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")  }  def flutterVersionCode = localProperties.getProperty('flutter.versionCode')  if (flutterVersionCode == null) {  flutterVersionCode = '1'  }  def flutterVersionName = localProperties.getProperty('flutter.versionName')  if (flutterVersionName == null) {  flutterVersionName = '1.0'  }  apply plugin: 'com.android.application'  apply plugin: 'com.google.gms.google-services'  apply plugin: 'kotlin-android'  apply from: "$flutterRoot/packages/flutter\_tools/gradle/flutter.gradle"  android {  compileSdkVersion 30  compileOptions {  sourceCompatibility JavaVersion.VERSION\_1\_8  targetCompatibility JavaVersion.VERSION\_1\_8  }  kotlinOptions {  jvmTarget = '1.8'  }  sourceSets {  main.java.srcDirs += 'src/main/kotlin'  }  defaultConfig {  // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).  applicationId "com.example.demo\_cicd"  minSdkVersion 21  targetSdkVersion 30  versionCode flutterVersionCode.toInteger()  versionName flutterVersionName  }  buildTypes {  release {  // TODO: Add your own signing config for the release build.  // Signing with the debug keys for now, so `flutter run --release` works.  signingConfig signingConfigs.debug  }  }  }  flutter {  source '../..'  }  dependencies {  implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  implementation platform('com.google.firebase:firebase-bom:29.0.0')  implementation 'com.google.firebase:firebase-analytics-ktx'  def appCenterSdkVersion = '4.3.1'  implementation "com.microsoft.appcenter:appcenter-analytics:${appCenterSdkVersion}"  implementation "com.microsoft.appcenter:appcenter-crashes:${appCenterSdkVersion}"  } |

1. App Center
2. Perhatikan potongan kode berikut saat melakukan kondigurasi pada app center dengan projek flutter

|  |
| --- |
| AppCenter.start(application, "559fc51f-cad9-45d1-bd68-625af966c4de",, Analytics::class.java, Crashes::class.java) |

Potongan kode tersebut masih berformat java, ubah format code tersebut dengan kotlin agar bisa dipakai dalam projek.

1. Perhatikan gambar berikut ini





Apa yang perlu ditambahkan agar potongan code pada build.gradle dapat digunakan pada app center?

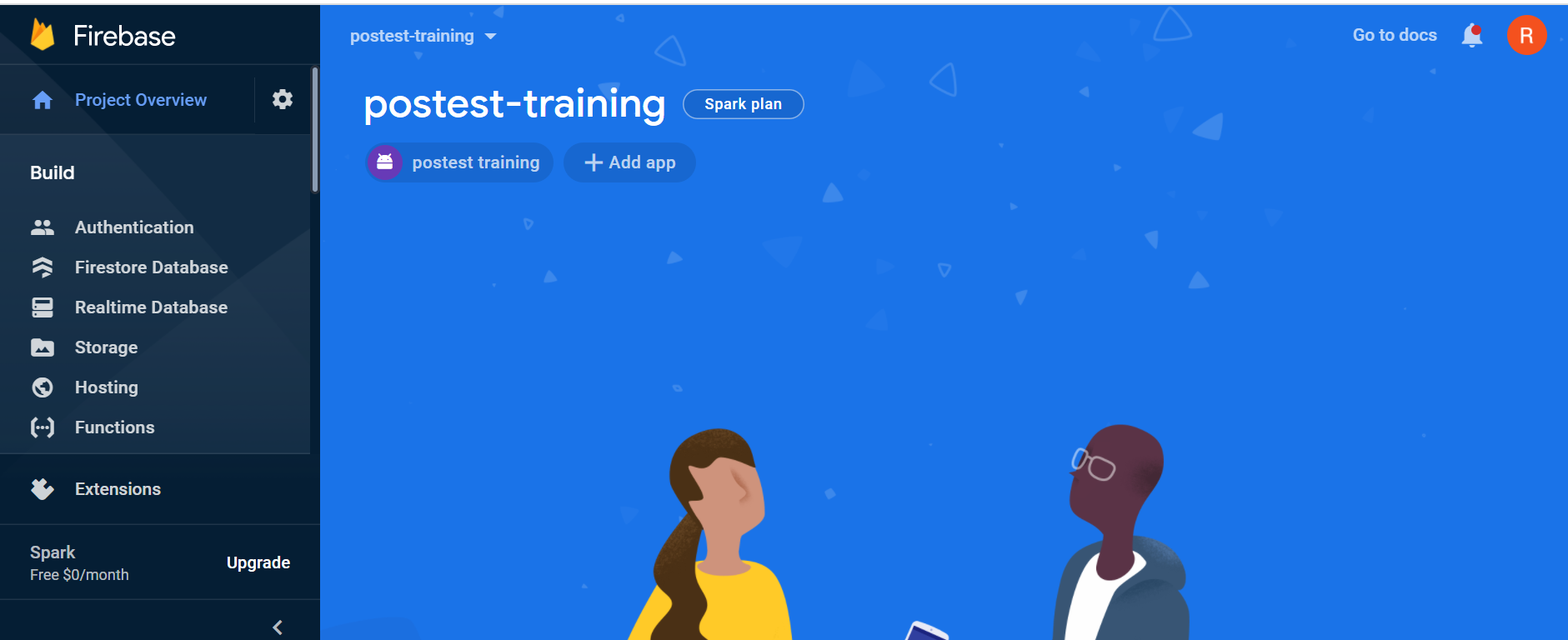
1. Buatlah sebuah CI CD pada golang app menggunakan github action dengan ketentuan sebagai berikut :

* Trigger saat membuat tags baru
* Runs on ubuntu
* Dengan steps checkout code, validates go release config, dan release on github menggunakan docker://goreleaser/goreleaser:latest dengan args release

1. Buat sebuah CI CD Flutter App menggunakan github action dengan ketentuan sebagai berikut :

* Trigger push master
* Run on mac OS
* Beberapa steps yang harus dijalankan adalah get dependencies, anaylize, test, build apk

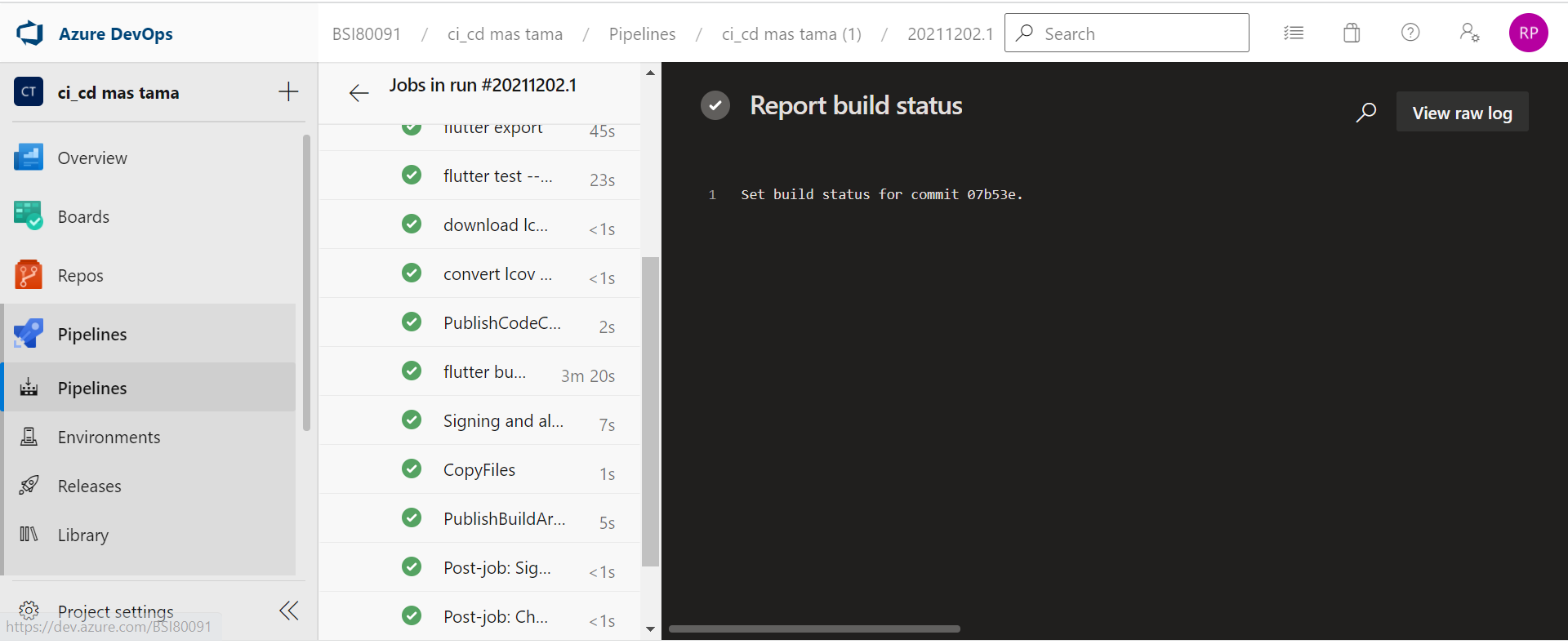
1. Dengan melanjutkan konfigurasi soal nomer 6, lanjutkan continue delivery ke firebase app distribution. Tuliskan code configurasi pada file yml nya dengan menambahkan :

* Upload to artifact
* App secret untuk configurasi dengan firebase
* Melampirkan juga screen shoot hasil konfigurasi firebase dengan projek   
    
  Graphical user interface, text, application

  Description automatically generated  
  Graphical user interface, text

  Description automatically generated
* Source : https://github.com/rizkyari828/postest\_training.git

1. Buatlah sebuah CI CD Flutter App dengan azure pipeline dengan ketentuan sebagai berikut :

* Trigger maste
* Run on ubuntu
* Step yang harus dijalankan adalah flutter install, test, build apk, sign apk, dan copy file to artifact  
    
    
  Graphical user interface, text, application

  Description automatically generated  
    
  - Karena azure saya tidak bisa digunakan untuk create project, saya membuat pipeline baru di project yang sudah ada.

1. Dengan melanjutkan soal nomer 8, lakukanlah configurasi pada release dan hingga deployment ke app center dengan ketentuan berikut :

* Release dijalankan otomatis saat pipeline berhasil dijalankan
* Deploy dijalankan setelah release berhasil
* Pada app center otomatis mengirimkan kepada email tester
* Lampirkan semua hasil konfigurasi masing-masing step

1. Dengan menggunakan projek yang sudah dibuatkan CI CD pada nomer 8 dan 9. Buatlah konfigurasi pada microsoft teams supaya projek bisa terintegrasi. Lampirkan screen shoot setiap langkahnya.