

PROGRAM 11

Step 1: Verify Docker Installation

```
docker --version
```

Step 2: Pull the Hello World Container

```
docker pull hello-world
```

Step 3: Run the Hello World Container

```
docker run hello-world
```

Step 4: Display the container

```
docker ps
```

Step 5: Pull the Ubuntu Container

```
docker pull ubuntu
```

Step 6: Run a Command Using the Ubuntu Image

```
docker run ubuntu echo "Hello MAD Lab"
```

Step 7: View the Runned Containers

```
docker inspect container_id
```

Step 8: View Downloaded Images

```
docker images
```

Step 9: Start an Interactive Ubuntu Container

```
docker run -it Ubuntu
```

```
ls
```

```
pwd
```

OUTPUT:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MCALAB1>docker --version
Docker version 28.0.4, build b8034c0

C:\Users\MCALAB1>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:c41088499908a59aae84b0a49c70e86f4731e588a737f1637e73c8c09d995654
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

C:\Users\MCALAB1>docker run ubuntu echo "Hello MAD Lab"
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
0622fac788ed: Pull complete
Digest: sha256:6015f66923d7afbc53558d7ccffd325d43b4e249f41a6e93eef074c9505d2233
Status: Downloaded newer image for ubuntu:latest
Hello MAD Lab
```

```
C:\Users\MCALAB1>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    6015f66923d7   7 days ago    117MB
hello-world    latest    c41088499908   3 months ago  20.4kB

C:\Users\MCALAB1>docker run -it ubuntu
root@b66beab77981:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@b66beab77981:/# pwd
/
root@b66beab77981:/# exit
exit
```

PROGRAM 12

Step 1: Install Google Cloud SDK (gcloud CLI)

Step 2: gcloud init

Step 3: gcloud projects create engine-ex12-2025

Step 4: gcloud config set project engine-ex12-2025

Step 5: gcloud app create --region=us-central

Step 6: gcloud app deploy

Step 7: gcloud app browse

Main.py:

```
from flask import Flask, request
```

```
app = Flask(__name__)
```

```
@app.route('/')
def form():
```

```
    return """
        <form action="/add" method="post">
            <input type="text" name="a"> +
            <input type="text" name="b">
            <input type="submit">
        </form>
    """
```

```
@app.route('/add', methods=['POST'])
```

```
def add():
    try:
        a = int(request.form['a'])
        b = int(request.form['b'])
        return f'The sum is: {a + b}'
    except Exception as e:
        return f'Error: {e}'
```

app.yaml:

runtime: python39

```
entrypoint: gunicorn -b :$PORT main:app
```

```
automatic_scaling:
```

```
  max_instances: 1
```

requirements.txt:

```
Flask==2.2.5
```

```
Werkzeug==2.2.3
```

```
gunicorn==20.1.0
```

OUTPUT:

```
Your Cloud Platform project in this session is set to ethereal-mind-459613-d3.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
akashrmvec@cloudshell:~ (ethereal-mind-459613-d3)$ cd gae_hello_addition_app
akashrmvec@cloudshell:~/gae_hello_addition_app (ethereal-mind-459613-d3)$ gcloud app deploy
Services to deploy:




descriptor:          [/home/akashrmvec/gae_hello_addition_app/app.yaml]
source:              [/home/akashrmvec/gae_hello_addition_app]
target project:      [ethereal-mind-459613-d3]
target service:      [default]
target version:      [20250512t134816]
target url:          [https://ethereal-mind-459613-d3.el.r.appspot.com]
target service account: [ethereal-mind-459613-d3@appspot.gserviceaccount.com]



Do you want to continue (Y/n)? y

Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://ethereal-mind-459613-d3.el.r.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse
akashrmvec@cloudshell:~/gae_hello_addition_app (ethereal-mind-459613-d3)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://ethereal-mind-459613-d3.el.r.appspot.com
akashrmvec@cloudshell:~/gae_hello_addition_app (ethereal-mind-459613-d3)$
```

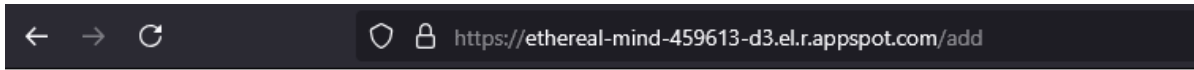
  https://ethereal-mind-459613-d3.el.r.appspot.com

10

+

5

Submit Query



The sum is: 15

PROGRAM 13

Main_activity.java:

```
package com.example.alarmclock;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.ToggleButton;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    private AlarmManager alarmManager;
    private PendingIntent pendingIntent;
    private TimePicker alarmTimePicker;
    private TextView alarmText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        alarmTimePicker = findViewById(R.id.alarmTimePicker);
```

```

        alarmText = findViewById(R.id.alarmText);
        alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    }

    public void onToggleClicked(View view) {
        ToggleButton toggle = (ToggleButton) view;

        if (toggle.isChecked()) {
            int hour = alarmTimePicker.getHour();
            int minute = alarmTimePicker.getMinute();

            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, hour);
            calendar.set(Calendar.MINUTE, minute);
            calendar.set(Calendar.SECOND, 0);

            Intent intent = new Intent(this, AlarmReceiver.class);
            pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
                PendingIntent.FLAG_IMMUTABLE);
            alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
                pendingIntent);

            alarmText.setText("Alarm set for: " + hour + ":" + String.format("%02d", minute));
        } else {
            if (pendingIntent != null) {
                alarmManager.cancel(pendingIntent);
            }
            alarmText.setText("Alarm canceled");
        }
    }
}

```



```
}
```

AlarmService.java:

```
package com.example.alarmclock;

import android.app.IntentService;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import androidx.core.app.NotificationCompat;

public class AlarmService extends IntentService {

    public AlarmService() {
        super("AlarmService");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        sendNotification("Wake Up! Alarm is ringing!");
    }

    private void sendNotification(String msg) {

        NotificationManager manager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);

        Intent intent = new Intent(this, MainActivity.class);

        PendingIntent contentIntent = PendingIntent.getActivity(this, 0, intent,
        PendingIntent.FLAG_IMMUTABLE);
```

```

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "default")
    .setSmallIcon(R.drawable.ic_launcher_foreground)
    .setContentTitle("Alarm Clock")
    .setContentText(msg)
    .setContentIntent(contentIntent)
    .setAutoCancel(true);

manager.notify(1, builder.build());
}
}

```

AlarmReceiver.java:

```

package com.example.alarmclock;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;

public class AlarmReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {

        Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null) {

```

```

        alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    }

    Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
    ringtone.play();

    Intent service = new Intent(context, AlarmService.class);
    context.startService(service);
}
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.WAKE_LOCK" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.AlarmClock"
        tools:targetApi="31">
        <activity

```

```

        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <!-- Register the broadcast receiver for alarm -->
    <receiver android:name=".AlarmReceiver" />
    <!-- Register the service to handle alarm notification -->
    <service android:name=".AlarmService" />
</application>
</manifest>

```

Activity.xml:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TimePicker
        android:id="@+id/alarmTimePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"/>

```

```
<ToggleButton
    android:id="@+id/alarmToggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Alarm On/Off"
    android:onClick="onToggleClicked"
    android:layout_below="@id/alarmTimePicker"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"/>
```

```
<TextView
    android:id="@+id/alarmText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No Alarm Set"
    android:layout_below="@id/alarmToggle"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"/>
```

```
</RelativeLayout>
```

OUTPUT:

