## Project Overview

MAHESH

This project involves using Terraform to create multiple Spotify playlists for different occasions like morning, evening, party night, etc. Terraform will be used to automate the creation and management of these playlists.

## Prerequisites

1. **Terraform Installed**: Ensure Terraform is installed on your machine.

2. **Docker Installed**: Make sure Docker is installed and running.

3. **Spotify Account**: You need a Spotify account (without premium access)

4. **Spotify Developer Account**: Register and create an application to get the Client ID and Client Secret.

5. **Spotify Provider for Terraform**: Install and configure the Spotify provider for Terraform.

6. **VS Code Editor**: Recommended for editing Terraform files.

## Steps to Complete the Project

### 1. Creating Terraform Code

Start by setting up your Terraform project.

1. Create a new directory for your Terraform project and navigate to it in your terminal.

2. Create a file named `main.tf`.

### 2. Define Provider

In `main.tf`, define the Spotify provider:

```

provider "spotify" {

  api_key = "?"

}

```

### 3. Need API Key

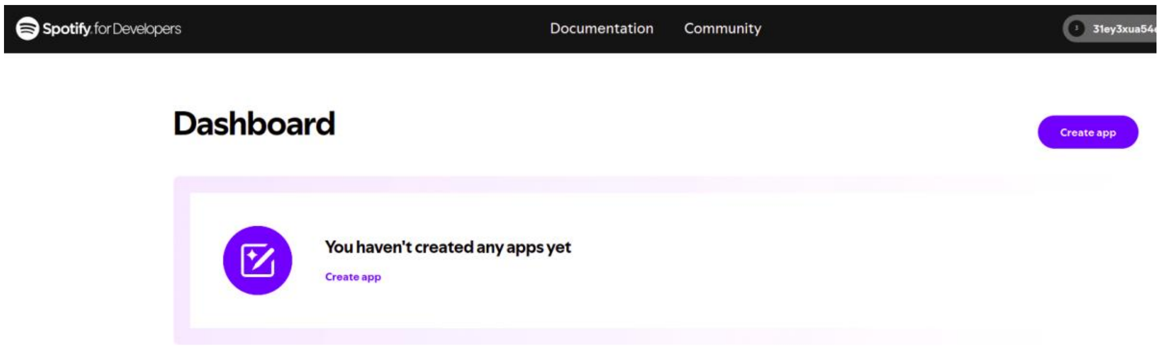
To interact with Spotify's API, you need a Client ID and Client Secret.

### 4. Start with App Creation

1. Go to the [Spotify Developer Dashboard](https://developer.spotify.com/dashboard/).

2. Log in with your Spotify account.

3. Click on "Create an App".



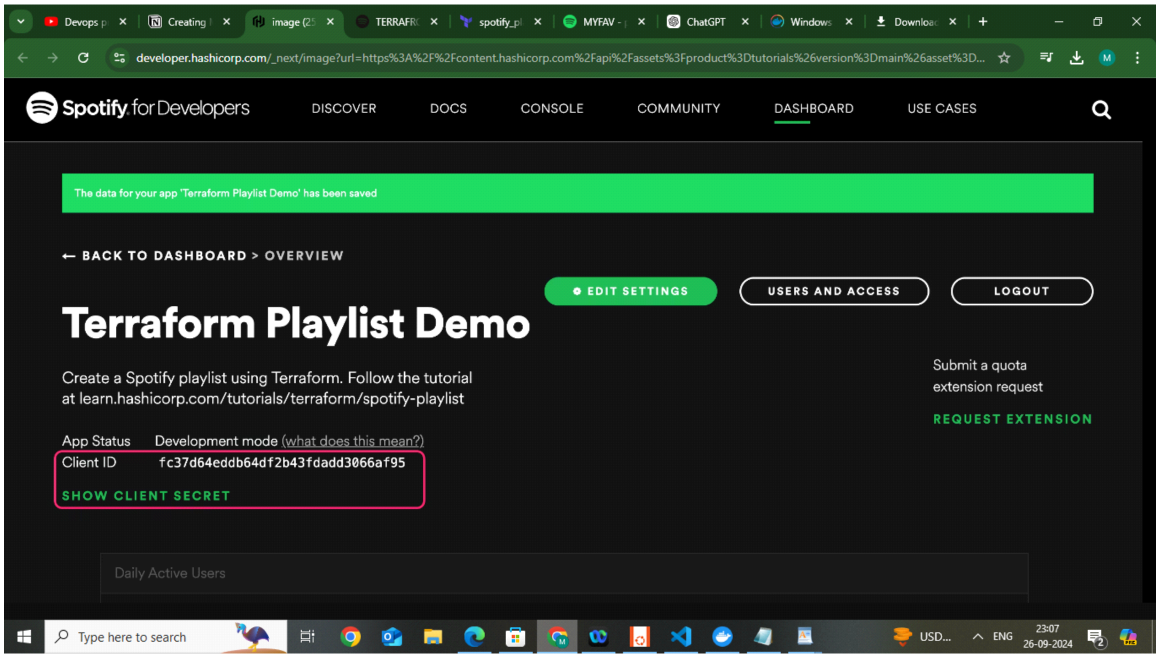
You haven't created any apps yet

4. Fill in the required details and create the app.

| Name | Description |

| My Playlist through Terraform | Create multiple Spotify playlists using Terraform. |

- *Redirect URIs: [http://localhost:27228/spotify_callback](http://localhost:27228/spotify_callback**)



The data for your app 'Terraform Playlist Demo' has been saved

← BACK TO DASHBOARD > OVERVIEW

Terraform Playlist Demo

Create a Spotify playlist using Terraform. Follow the tutorial at learn.hashicorp.com/tutorials/terraform/spotify-playlist

App Status Development mode (what does this mean?)

Client ID fc37d64eddb64df2b43fdadd3066af95

Submit a quota extension request

5. Enter Details

Create a file named .env to store your Spotify application's Client ID and Secret:

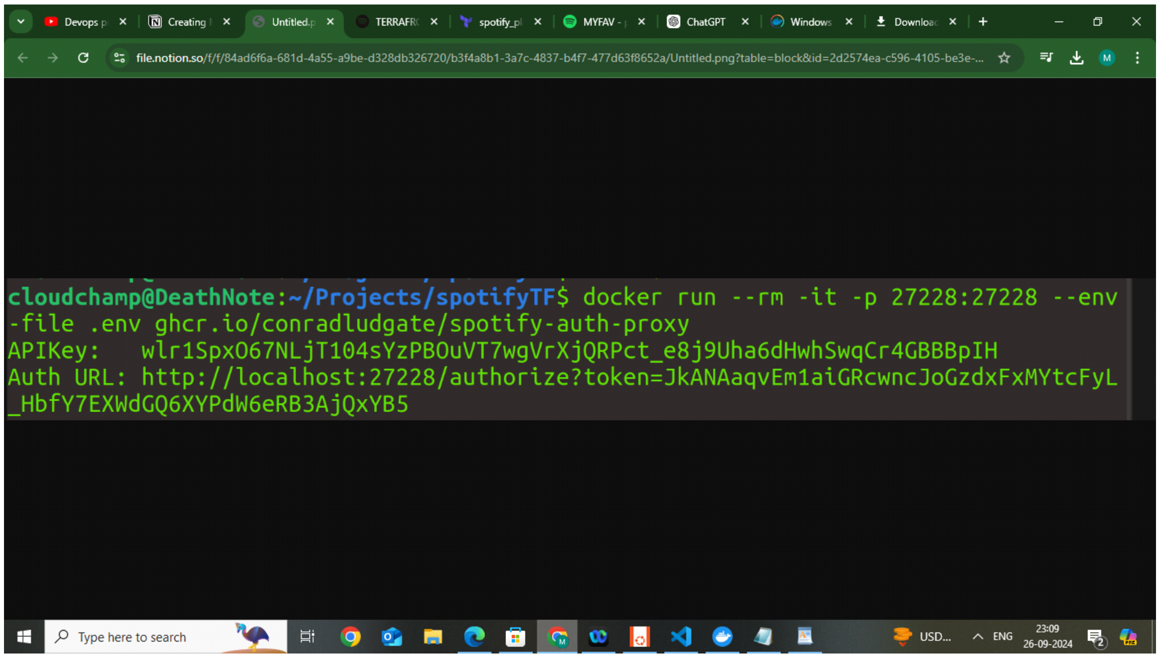
SPOTIFY_CLIENT_ID=<your_spotify_client_id>

SPOTIFY_CLIENT_SECRET=<your_spotify_client_secret>

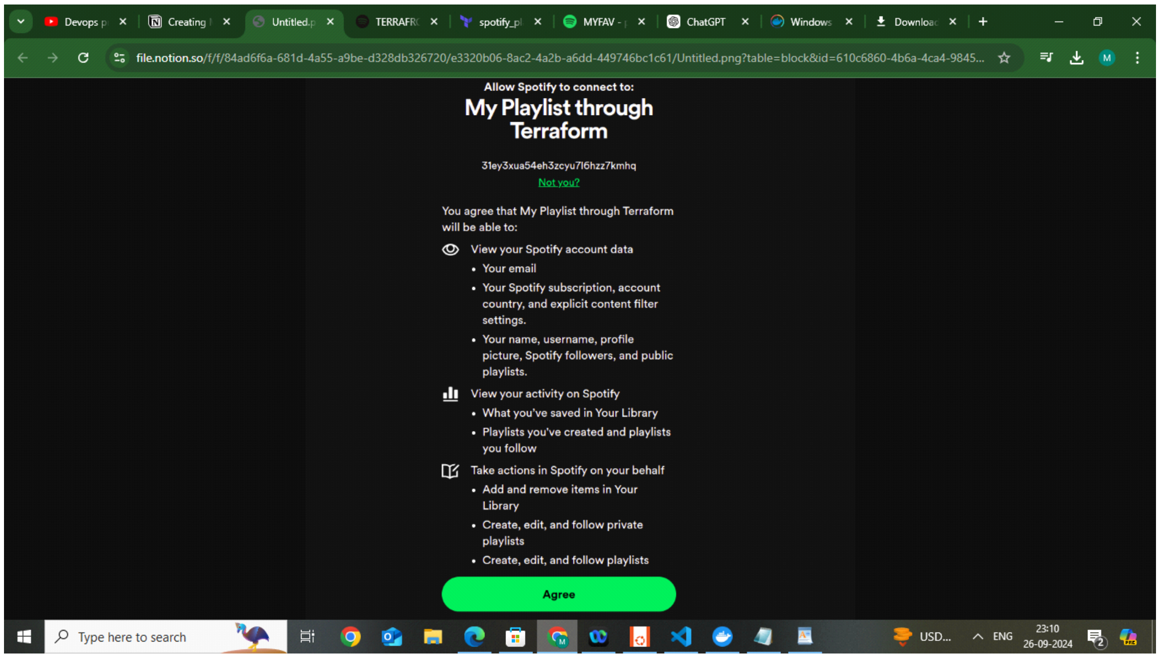
6. Run the Spotify Auth App and Get the API Key

Make sure Docker Desktop is running, and start the authorization proxy server:

docker run --rm -it -p 27228:27228 --env-file ./.env ghcr.io/conradludgate/spotify-auth-proxy



```
cloudchamp@DeathNote:~/Projects/spotifyTF$ docker run --rm -it -p 27228:27228 --env
-file .env ghcr.io/conradludgate/spotify-auth-proxy
APIKey:   wlr1SpxO67NLjT104sYzPBOuVT7wgVrXjQRPct_e8j9Uha6dHwhSwqCr4GBBBpIH
Auth URL: http://localhost:27228/authorize?token=JkANAaqvEm1aiGRcwncJoGzdxFxMYtcFyL
_HbfY7EXWdGQ6XYPdW6eRB3AjQxYB5
```



Allow Spotify to connect to:

My Playlist through Terraform

31ey3xua54eh3zcyu7l6hzz7kmhq

Not you?

You agree that My Playlist through Terraform will be able to:

View your Spotify account data
- Your email
- Your Spotify subscription, account country, and explicit content filter settings.
- Your name, username, profile picture, Spotify followers, and public playlists.

View your activity on Spotify
- What you've saved in Your Library
- Playlists you've created and playlists you follow

Take actions in Spotify on your behalf
- Add and remove items in Your Library
- Create, edit, and follow private playlists
- Create, edit, and follow playlists

### ou should get “Authorization Successful” Message.

8. Continue Creating Terraform Code

----------------------------------Main.tf------------------------

terraform {

  required_providers {

```
    spotify = {
      source = "conradludgate/spotify"
      version = "0.2.7"
    }
  }
}

provider "spotify" {
  # Configuration options
  api_key = var.api_key
}
---------------------------------------------------------------------------
------------------------------------Playlist.tf----------------------------
resource "spotify_playlist" "Mahesh" {
  name = "Mahesh"
  tracks = ["69KUKhBqyGvtO1jtJwZEcv"]
}
data "spotify_search_track" "TheWeeknd" {
    artist = "The Weeknd"
}
resource "spotify_playlist" "MYFAV" {
  name = "MYFAV"
  tracks = [
data.spotify_search_track.TheWeeknd.tracks[1].id,data.spotify_search_track.TheWeeknd.tracks[2].id,dat
a.spotify_search_track.TheWeeknd.tracks[3].id]
}
------------------------------------------------------------------------------------
-------------------------------Varible.tf---------------------------------------------
```

```
variable "api_key" {
  type = string
}
------------------------------------------------------------------------------------------
terraform.tfvars
api_key ="############################################"
---------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------
.env file
SPOTIFY_CLIENT_ID=8817364d43464a249c79553180bf0ff0
SPOTIFY_CLIENT_SECRET=8939c8378e2842fa958184e81
--------------------------------------------------------------------------------------
```

### 9. Initialize and Apply Terraform Configuration

1. Initialize the Terraform configuration:

    terraform init

2. Apply the Terraform configuration:

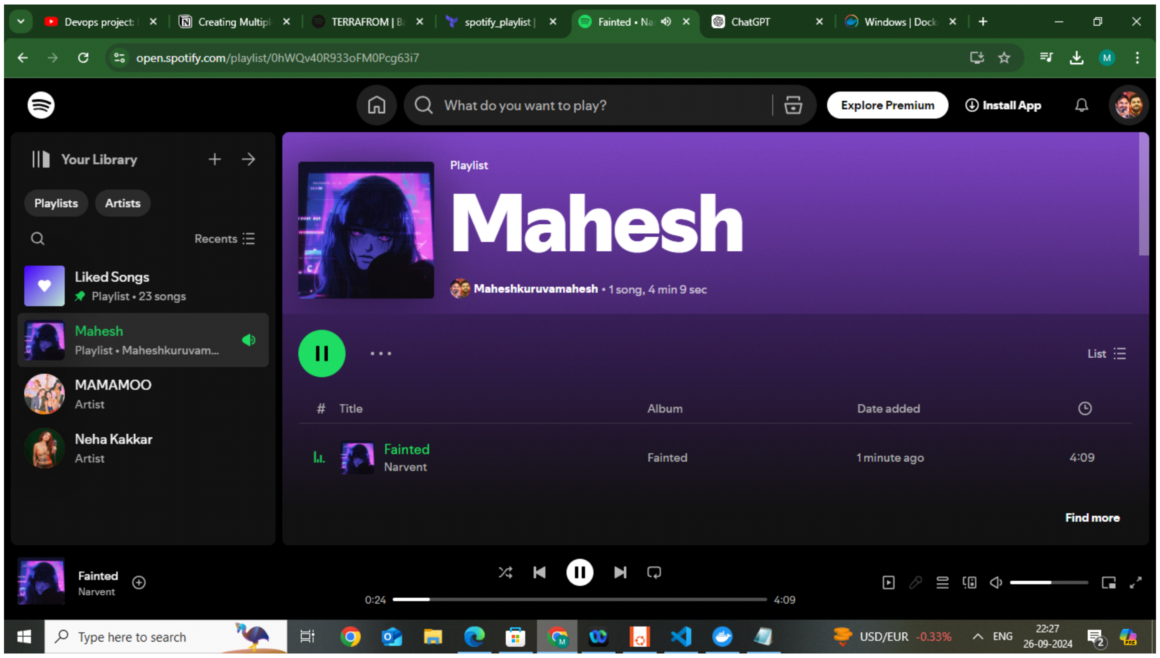
   ```
   terraform apply

   ```

### 11. Verify Playlists on Spotify

After applying the Terraform configuration, log in to your Spotify account and verify that the playlists have been created and populated with the specified tracks.

OUTPUT:

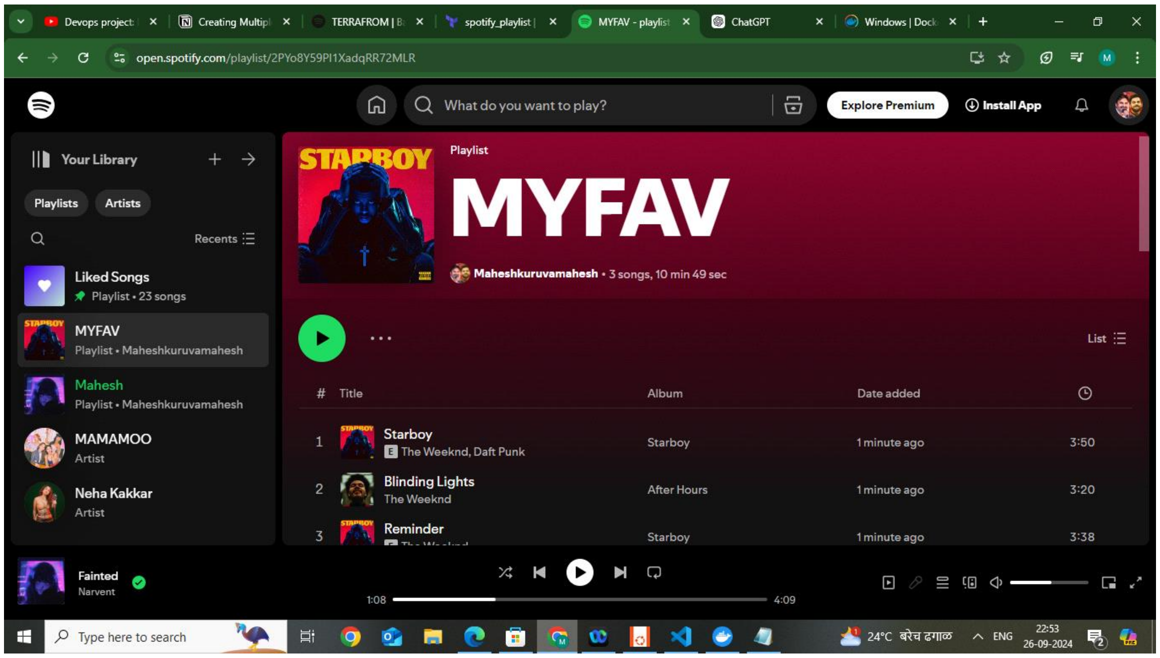


Playlist

Mahesh

Maheshkuruvamahesh • 1 song, 4 min 9 sec

#	Title	Album	Date added	🕒
	Fainted Narvent	Fainted	1 minute ago	4:09



Playlist

MYFAV

Maheshkuruvamahesh • 3 songs, 10 min 49 sec

#	Title	Album	Date added	🕒
1	Starboy E The Weeknd, Daft Punk	Starboy	1 minute ago	3:50
2	Blinding Lights The Weeknd	After Hours	1 minute ago	3:20
3	Reminder The Weeknd	Starboy	1 minute ago	3:38

## Conclusion

By following these steps, you can automate the creation and management of multiple Spotify playlists using Terraform. This approach not only saves time but also ensures consistency across your playlists. Customize the playlists and tracks as per your preference to suit different occasions.