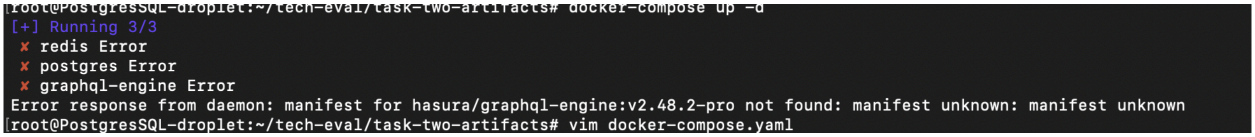
Task two

Environment Setup.

To setup the environment I used an Ubuntu droplet.Upload the shared zip file to s3 bucket and move the files on droplet by downloading them there. Navigated to the folder task-two-artifacts and used the below command to start the container.

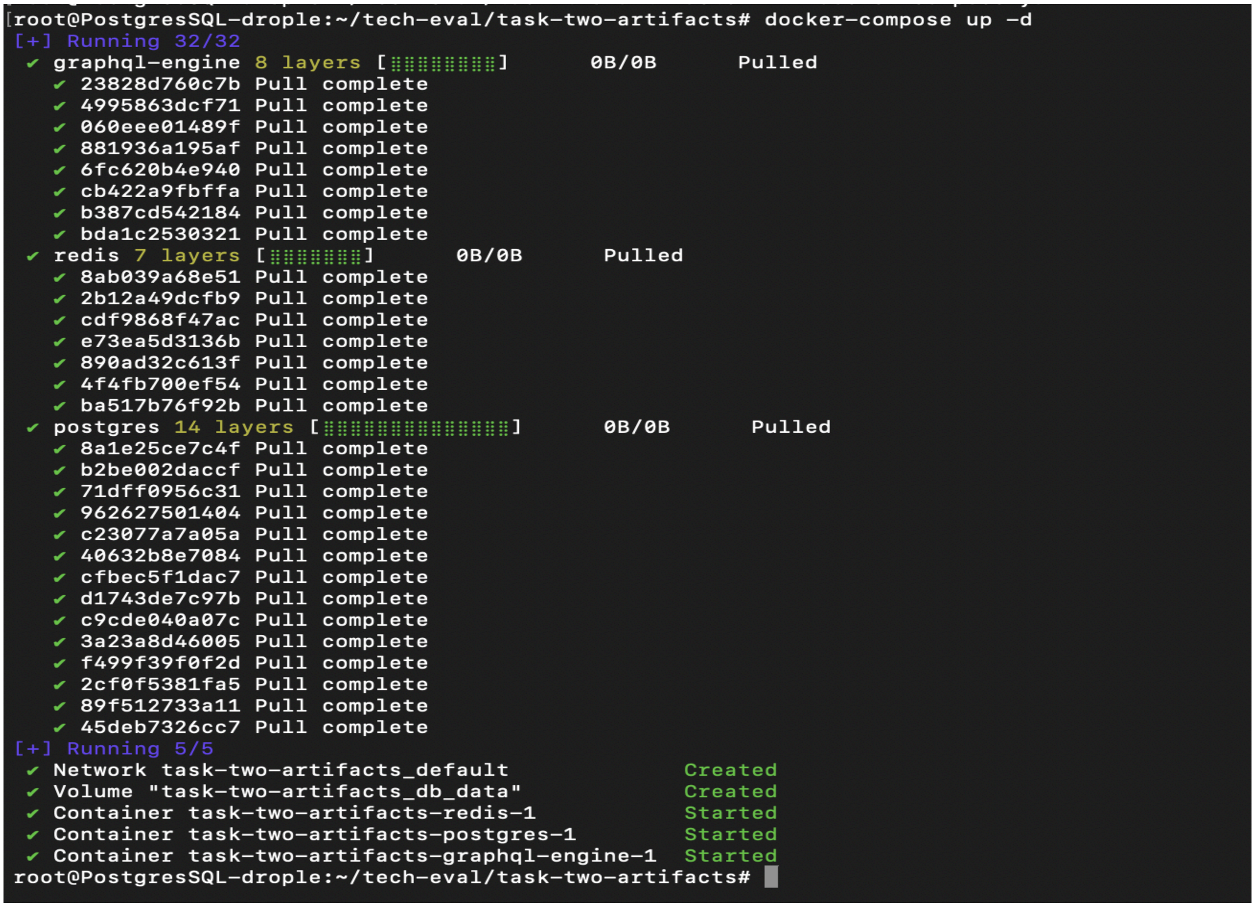
```
docker-compose up -d
```

However, I got the below error :



```
[root@PostgresSQL-droplet:~/tech-eval/task-two-artifacts# docker-compose up -d
[+] Running 3/3
 ✘ redis Error
 ✘ postgres Error
 ✘ graphql-engine Error
Error response from daemon: manifest for hasura/graphql-engine:v2.48.2-pro not found: manifest unknown: manifest unknown
[root@PostgresSQL-droplet:~/tech-eval/task-two-artifacts# vim docker-compose.yaml
```

Which indicated the Hasura version mentioned in the docker compose file was not found. While I checked the graph-engine releases I found out there was no version 2.48 available yet. I changed the version to 2.38.1 and the containers were created successfully.



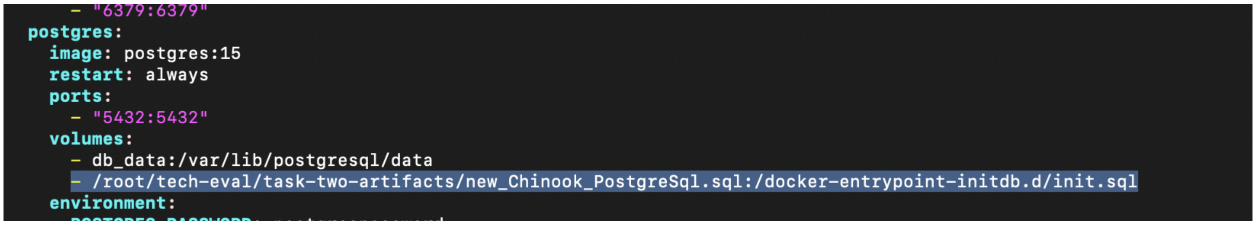
```
[root@PostgresSQL-drople:~/tech-eval/task-two-artifacts# docker-compose up -d
[+] Running 32/32
 ✔ graphql-engine 8 layers [⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ 23828d760c7b Pull complete
   ✔ 4995863dcf71 Pull complete
   ✔ 060eee01489f Pull complete
   ✔ 881936a195af Pull complete
   ✔ 6fc620b4e940 Pull complete
   ✔ cb422a9fbffa Pull complete
   ✔ b387cd542184 Pull complete
   ✔ bda1c2530321 Pull complete
 ✔ redis 7 layers [⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ 8ab039a68e51 Pull complete
   ✔ 2b12a49dcfb9 Pull complete
   ✔ cdf9868f47ac Pull complete
   ✔ e73ea5d3136b Pull complete
   ✔ 890ad32c613f Pull complete
   ✔ 4f4fb700ef54 Pull complete
   ✔ ba517b76f92b Pull complete
 ✔ postgres 14 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ 8a1e25ce7c4f Pull complete
   ✔ b2be002daccf Pull complete
   ✔ 71dff0956c31 Pull complete
   ✔ 962627501404 Pull complete
   ✔ c23077a7a05a Pull complete
   ✔ 40632b8e7084 Pull complete
   ✔ cfbec5f1dac7 Pull complete
   ✔ d1743de7c97b Pull complete
   ✔ c9cde040a07c Pull complete
   ✔ 3a23a8d46005 Pull complete
   ✔ f499f39f0f2d Pull complete
   ✔ 2cf0f5381fa5 Pull complete
   ✔ 89f512733a11 Pull complete
   ✔ 45deb7326cc7 Pull complete
[+] Running 5/5
 ✔ Network task-two-artifacts_default              Created
 ✔ Volume "task-two-artifacts_db_data"             Created
 ✔ Container task-two-artifacts-redis-1            Started
 ✔ Container task-two-artifacts-postgres-1         Started
 ✔ Container task-two-artifacts-graphql-engine-1   Started
root@PostgresSQL-drople:~/tech-eval/task-two-artifacts# 
```

Then I ran docker ps command to check the running container. However, the graphQL engine was not running as intended. When checking the container logs find out the postgres was not reachable. Below was the error in the logs:

```
{"error":"connection error","path":"$","code":"postgres-error","internal":"could not translate host name \"postgresdb\" to address: Temporary failure in name resolution\n"}
```

So I went back to the docker compose file and found the HASURA_GRAPHQL_METADATA_DATABASE_URL : postgres://postgres:somepassword@postgresdb:5432/postgres was set incorrectly. Changed the string to postgres://postgres:postgrespassword@postgres:5432/postgres and rerun the docker-compose command again to re-create the graphQl engine container. After recreation the container was running successfully.

Now moving the chinook data set to postgres running on the container. I'd like you or the hint you shared. I added the chinook.sql file to my droplet on which the container is hosted and added the path using init.sql to volume under postgres container "- /root/tech-eval/task-two-artifacts/new_Chinook_PostgreSql.sql:/docker-entrypoint-initdb.d/init.sql "

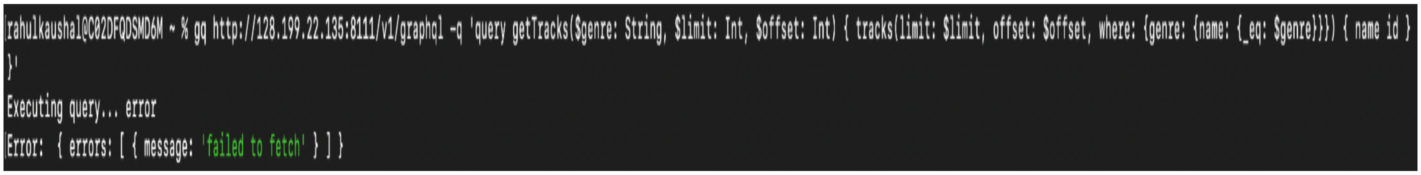


```
    - "6379:6379"
  postgres:
    image: postgres:15
    restart: always
    ports:
      - "5432:5432"
    volumes:
      - db_data:/var/lib/postgresql/data
      - /root/tech-eval/task-two-artifacts/new_Chinook_PostgreSql.sql:/docker-entrypoint-initdb.d/init.sql
    environment:
      POSTGRES_PASSWORD: postgrespassword
```

However, when I re-created the container the database was getting created. So I use a exec command to bash into the container and check if the file was copied successfully. However the folder /docker-entrypoint-initdb.d/ only had one file init.sql and the chinook.sql file was not there.

So as a workaround I used docker cp /root/tech-eval/task-two-artifacts/new_Chinook_PostgreSql.sql task-two-artifacts-postgres-1:/docker-entrypoint-initdb.d/ to copy the file manually to the container and the logged in to psql and used the \i command to initiate the data migration.

However, after data migration I was still not able to connect.Below was the error I was receiving while trying to access the endpoint.



```
rahulkaushal@C02DFQDSMD6M ~ % gq http://128.199.22.135:8111/v1/graphql -q 'query getTracks($genre: String, $limit: Int, $offset: Int) { tracks(limit: $limit, offset: $offset, where: {genre: {name: {_eq: $genre}}}) { name id }
}'
Executing query... error
Error:  { errors: [ { message: 'failed to fetch' } ] }
```

Upn checking the postgres container logs. I didn't find any entry for the request I made. Which made me think the request never reached the container. While checking the Hasura docker document. I found out that by default the connection to the graphql engine uses port 8080. However, we were using the target port as 80 and 8111 as the entry port. So I changed the port to 8080 and re-created the container. After which the request was coming in however, the query call was giving an error. Below is the error I found in container logs.

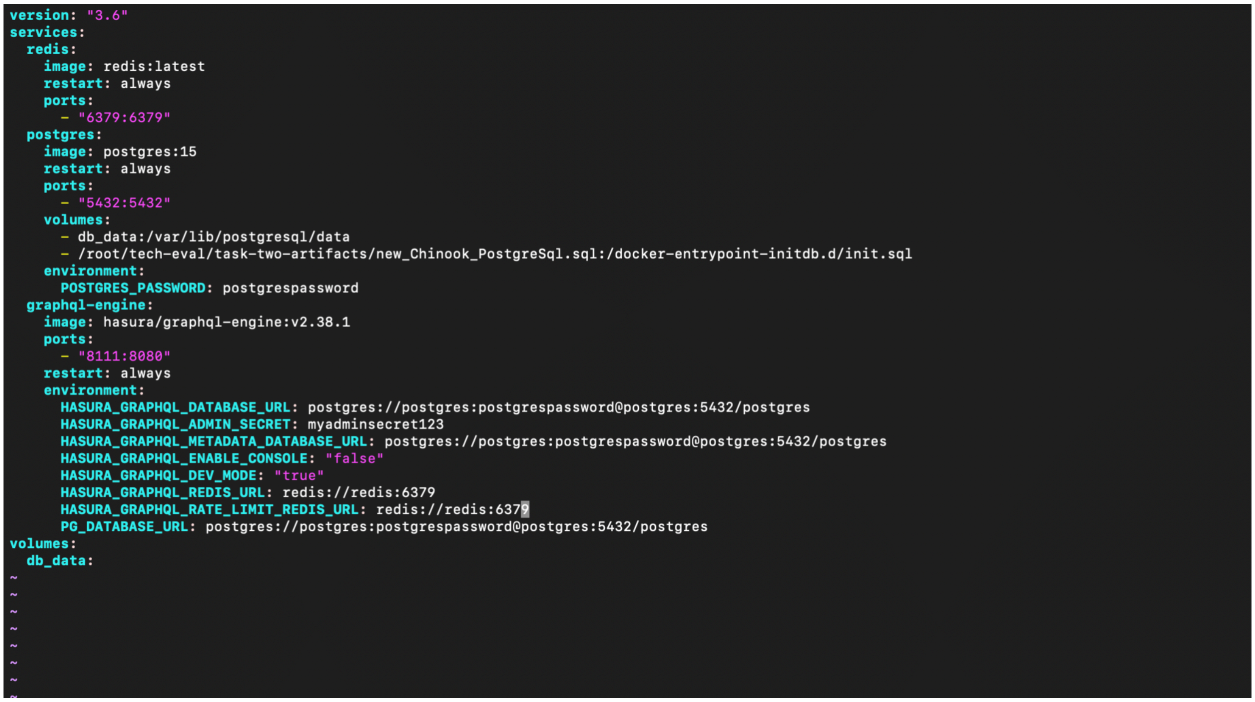
```
{"detail":{"http_info":{"content_encoding":null,"http_version":"HTTP/1.1","ip":"223.178.210.211","method":"POST","status":200,"url":"/v1/graphql"},"operation":{"error":{"code":"validation-failed","error":"field 'artist' not found in type: 'query_root'","path":"$.selectionSet.artist"},"query":{"type":null},"request_id":"7983cd8d-209b-49fd-824f-db32879e66dc","request_mode":"error","response_size":116,"uncompressed_response_size":116,"user_vars":{"x-hasura-role":"admin"}},"request_id":"7983cd8d-209b-49fd-824f-db32879e66dc"},"level":"error","span_id":"02a29cb9ae6320e4","timestamp":"2024-04-05T05:38:35.583+0000","trace_id":"57e303f64930ee1b4dd7a52be152670d","type":"http-log"}
```

Tried multiple ways to fix it by looking at the many articles but could not find a solution. Seems like an issue with the schema not created from the database. The reason I feel this is because when I run the command to introspect schema I get no response. Just a output stating "Executing query... done"

I have hit the 4 hours mark so I stopped troubleshooting this further. However, I'll need more to troubleshoot and find out what might be wrong. I'll be keeping my investigation ongoing to learn more about graphQL and Hasura.

Unfortunately I was not able to run the queries you shared due to the graphql engine not working. However, I think if it was working I would be able to as I had already added the admin secret to the docker compose file and by creating a user role and header x-hasura-artist-id i would have been able to run those queries against my graphql engine.

Below is the docker yaml file with all the changes I made:



```yaml
version: "3.6"
services:
  redis:
    image: redis:latest
    restart: always
    ports:
      - "6379:6379"
  postgres:
    image: postgres:15
    restart: always
    ports:
      - "5432:5432"
    volumes:
      - db_data:/var/lib/postgresql/data
      - /root/tech-eval/task-two-artifacts/new_Chinook_PostgreSql.sql:/docker-entrypoint-initdb.d/init.sql
    environment:
      POSTGRES_PASSWORD: postgrespassword
  graphql-engine:
    image: hasura/graphql-engine:v2.38.1
    ports:
      - "8111:8080"
    restart: always
    environment:
      HASURA_GRAPHQL_DATABASE_URL: postgres://postgres:postgrespassword@postgres:5432/postgres
      HASURA_GRAPHQL_ADMIN_SECRET: myadminsecret123
      HASURA_GRAPHQL_METADATA_DATABASE_URL: postgres://postgres:postgrespassword@postgres:5432/postgres
      HASURA_GRAPHQL_ENABLE_CONSOLE: "false"
      HASURA_GRAPHQL_DEV_MODE: "true"
      HASURA_GRAPHQL_REDIS_URL: redis://redis:6379
      HASURA_GRAPHQL_RATE_LIMIT_REDIS_URL: redis://redis:6379
      PG_DATABASE_URL: postgres://postgres:postgrespassword@postgres:5432/postgres
volumes:
  db_data:
~
~
~
~
~
~
~
~
```

I Hope I was able to complete the assignment 😢.