Deploying

Using

By: Sarfaraz Momin (Cloud Engineer)

- Install Docker Compose using the repository.

1. Set up the repository. Find distro-specific instructions in:
   Ubuntu, CentOS, Debian, Fedora, RHEL, SLES
2. Update the package index, and install the latest version of Docker Compose:
   For Ubuntu, run:

```
apt-get update
```

```
rootsources.list.d# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2045 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2425 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [415 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1666 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [234 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1038 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [243 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [812 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [161 kB]
```

```
apt-get install docker-compose-plugin
```

```
rootsources.list.d# apt-get install docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker-compose-plugin is already the newest version (2.15.1-1~ubuntu.20.04~focal).
```

3. Verify that Docker Compose is installed correctly by checking the version.

```
docker compose version
```

```
rootschema# docker compose version
Docker Compose version v2.15.1
```

- Write docker-compose.yaml

```yaml
version: '2.2'

services:
  cube_api:
    restart: always
    image: cubejs/cube:v0.32.7
    ports:
      - 4000:4000
    environment:
      - CUBEJS_DB_TYPE=mysql
      - CUBEJS_DB_HOST=database-1.cnnkopxuqhho.ap-south-1.rds.amazonaws.com
      - CUBEJS_DB_NAME=cubejs
      - CUBEJS_DB_USER=root
      - CUBEJS_DB_PASS=123456789
      - CUBEJS_REDIS_URL=redis://redis:6379
      - CUBEJS_API_SECRET=secret
      - CUBEJS_REFRESH_WORKER=true
      - CUBEJS_DB_EXPORT_BUCKET=cubestore
      - CUBEJS_CUBESTORE_HOST=cubestore_router
      - CUBEJS_DEV_MODE=true
    volumes:
      - .:/cube/conf
    depends_on:
      - cubestore_worker_1
      - cubestore_worker_2
      - cube_refresh_worker
      - redis
  cube_refresh_worker:
    restart: always
    image: cubejs/cube:v0.32.7
    environment:
      - CUBEJS_DB_TYPE=mysql
      - CUBEJS_DB_HOST=database-1.cnnkopxuqhho.ap-south-1.rds.amazonaws.com
      - CUBEJS_DB_NAME=cubejs
      - CUBEJS_DB_USER=root
      - CUBEJS_DB_PASS=123456789
      - CUBEJS_REDIS_URL=redis://redis:6379
      - CUBEJS_API_SECRET=secret
      - CUBEJS_REFRESH_WORKER=true
      - CUBEJS_DB_EXPORT_BUCKET=cubestore
      - CUBEJS_CUBESTORE_HOST=cubestore_router
    volumes:
      - .:/cube/conf
```

```
cubestore_router:
  restart: always
  image: cubejs/cubestore:v0.32.7
  environment:
    - CUBESTORE_WORKERS=cubestore_worker_1:10001,cubestore_worker_2:10002
    - CUBESTORE_REMOTE_DIR=/cube/data
    - CUBESTORE_META_PORT=9999
    - CUBESTORE_SERVER_NAME=cubestore_router:9999
  volumes:
    - .cubestore:/cube/data
cubestore_worker_1:
  restart: always
  image: cubejs/cubestore:v0.32.7
  environment:
    - CUBESTORE_WORKERS=cubestore_worker_1:10001,cubestore_worker_2:10002
    - CUBESTORE_SERVER_NAME=cubestore_worker_1:10001
    - CUBESTORE_WORKER_PORT=10001
    - CUBESTORE_REMOTE_DIR=/cube/data
    - CUBESTORE_META_ADDR=cubestore_router:9999
  volumes:
    - .cubestore:/cube/data
  depends_on:
    - cubestore_router

cubestore_worker_2:
  restart: always
  image: cubejs/cubestore:v0.32.7
  environment:
    - CUBESTORE_WORKERS=cubestore_worker_1:10001,cubestore_worker_2:10002
    - CUBESTORE_SERVER_NAME=cubestore_worker_2:10002
    - CUBESTORE_WORKER_PORT=10002
    - CUBESTORE_REMOTE_DIR=/cube/data
    - CUBESTORE_META_ADDR=cubestore_router:9999
  volumes:
    - .cubestore:/cube/data
  depends_on:
    - cubestore_router

redis:
  image: bitnami/redis:latest
  environment:
    - ALLOW_EMPTY_PASSWORD=yes
  logging:
    driver: none
```

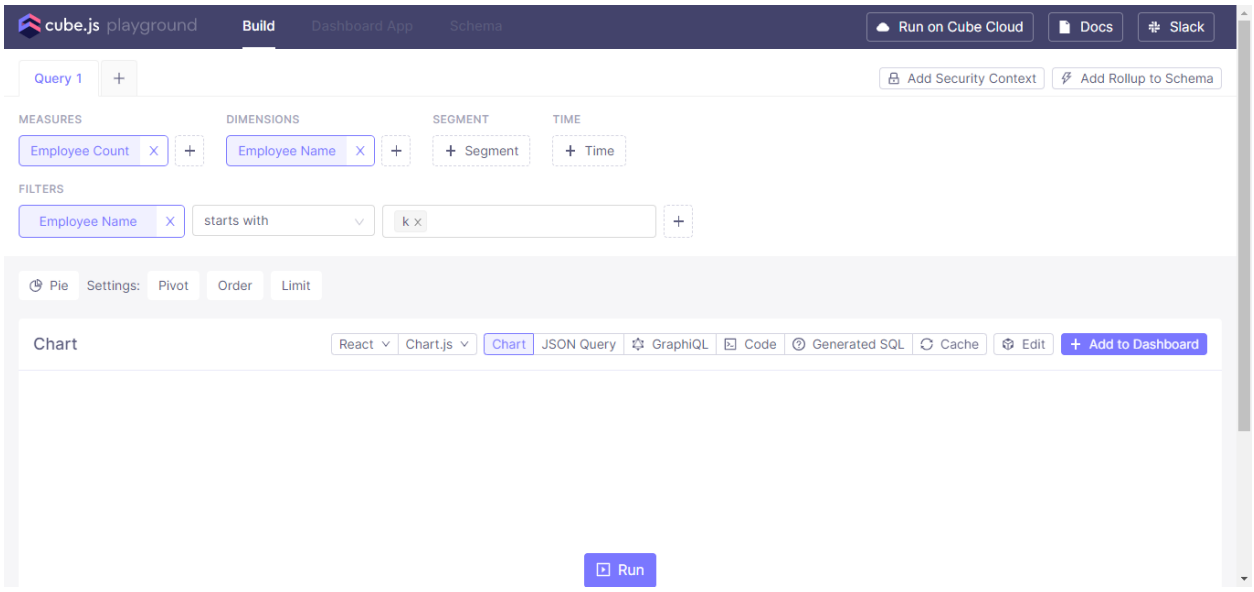
- Run “docker compose up -d” command.

```
root15_CUBE.JS# docker compose up -d
[+] Running 22/22
 ⠿ cube_api Pulled
 ⠿ redis Pulled
   ⠿ 3bcc7c26535e Pull complete
 ⠿ cube_refresh_worker Pulled
   ⠿ 3f9582a2cbe7 Pull complete
   ⠿ 94e5d5746476 Pull complete
   ⠿ 088f8d1aa818 Pull complete
   ⠿ d183209bda9f Pull complete
   ⠿ b4ff7361dea3 Pull complete
   ⠿ e2834840989f Pull complete
   ⠿ b7dfba618c70 Pull complete
   ⠿ 3ade9204f97b Pull complete
   ⠿ 5154d58c0711 Pull complete
   ⠿ 7ac6be5ce026 Pull complete
   ⠿ 09ae9ecee1fa Pull complete
   ⠿ cccb656d2d16 Pull complete
```

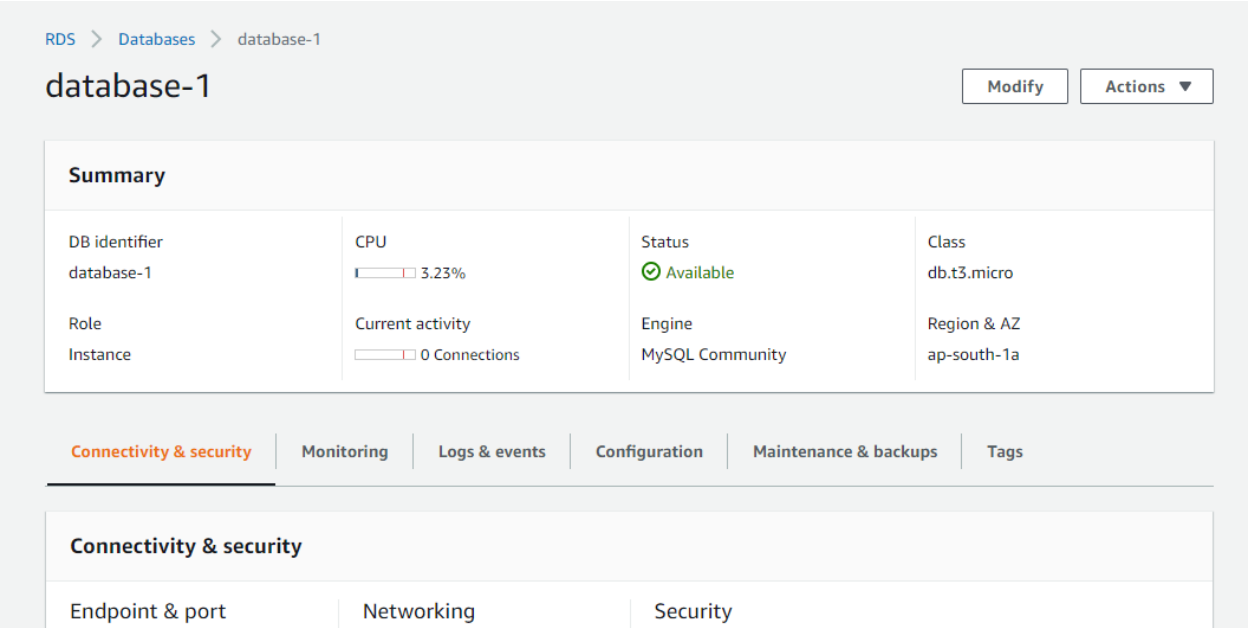
- Run “docker ps” command.

```
root15_CUBE.JS# docker compose ps
NAME                              IMAGE                       COMMAND                  SERVICE             CREATED              STATUS
PORTS
15_cubejs-cube_api-1              cubejs/cube:v0.32.7         "docker-entrypoint.s…"   cube_api            46 seconds ago       Up 44 seconds
0.0.0.0:4000->4000/tcp, :::4000->4000/tcp
15_cubejs-cube_refresh_worker-1   cubejs/cube:v0.32.7         "docker-entrypoint.s…"   cube_refresh_worker 3 minutes ago        Up 3 minutes
4000/tcp
15_cubejs-cubestore_router-1      cubejs/cubestore:v0.32.7    "./cubestored"           cubestore_router    3 minutes ago        Up 3 minutes
3306/tcp
```

Cube_api is running on “port no. 4000”



- To connect Database we need to create one RDS instance on AWS.



- Database Name should be unique.
- Choose a database creation method : Standard Create
- Engine options : MySQL
- Public Access should be ticked on "YES".
- Create a new VPC Group with Inbound Rules of "All Traffic".

- Install "mysql-client" using "apt install mysql-client" Command.

```
root15_CUBE.JS# apt install mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  mysql-client-8.0 mysql-client-core-8.0 mysql-common
The following NEW packages will be installed:
```

- To Connect MySQL to CLI use this command.
- Syntax:

  mysql –u [USER NAME] –h [HOST NAME] –p

```
root15_CUBE.JS# mysql -u root -h database-1.cnnkopxuqhho.ap-south-1.rds.amazonaws.com -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.28 Source distribution
```

- Create a database.

```
mysql> create database cubejs
    -> ;
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| cubejs             |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.03 sec)

mysql> use cubejs;
Database changed
```

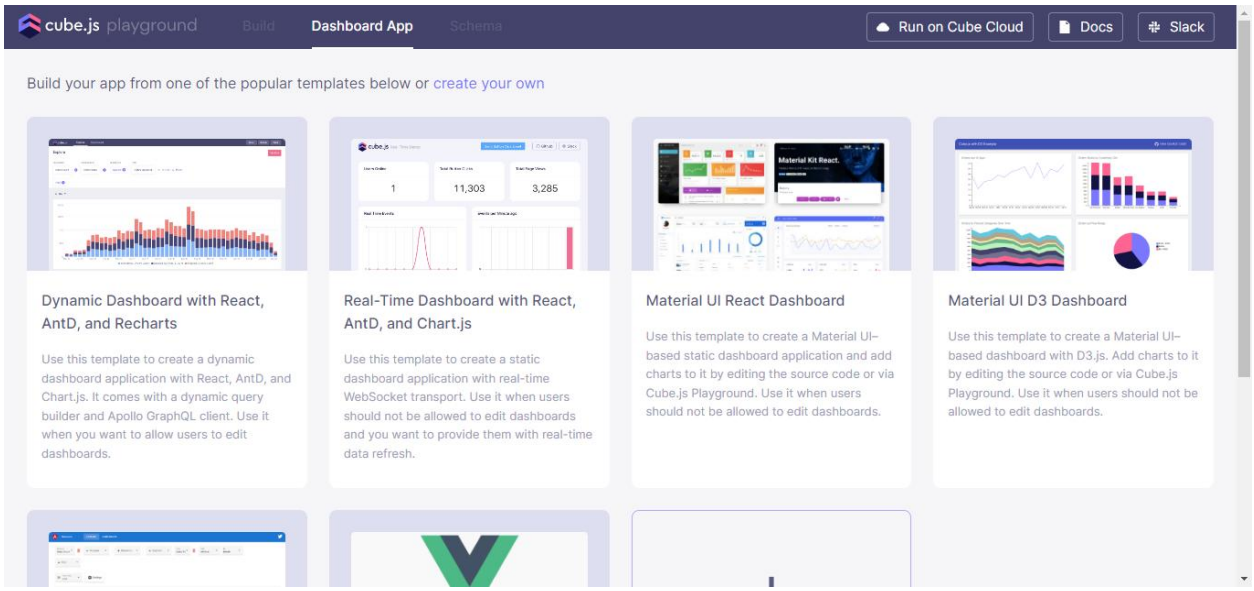
- Create Table.

```
mysql> CREATE TABLE users (
    ->   id INT NOT NULL AUTO_INCREMENT,
    ->   name VARCHAR(50) NOT NULL,
    ->   email VARCHAR(50) NOT NULL,
    ->   PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

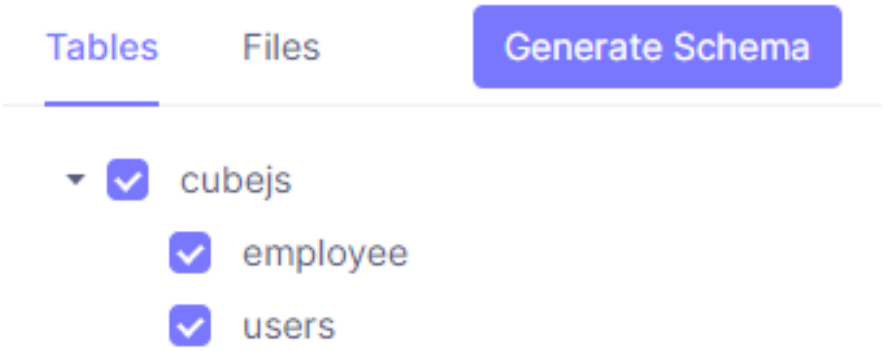
- Add Values

```
mysql> INSERT INTO users (name, email) VALUES ('sarfaraz momin', 'john.smasith@example.com');
Query OK, 1 row affected (0.04 sec)
```

- Open cube.js playground on port no. 4000
  Localhost:4000/



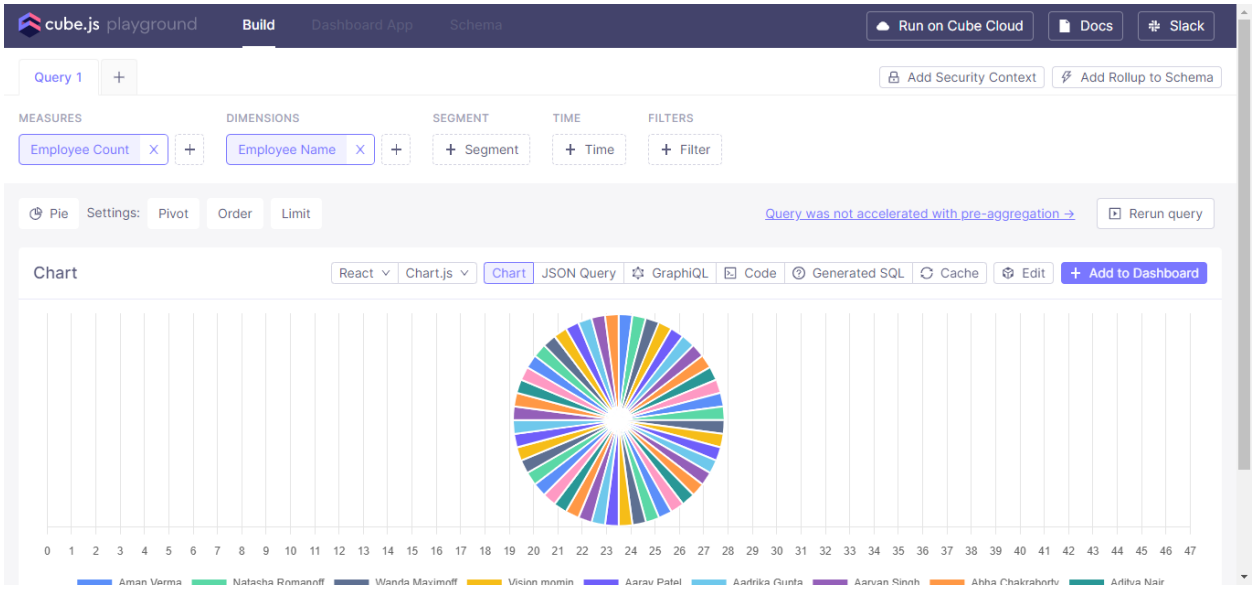
- Click on Schema and Select Database.



- Click on Generate Schema.

```
docker-compose.yaml  node_modules  schema
root15_CUBE.JS# cd schema/
rootschema# ll
total 0
drwxr-xr-x 1 root     root     4096 Mar 16 15:55 ./
drwxrwxrwx 1 sarfaraz sarfaraz 4096 Mar 16 14:24 ../
-rw-r--r-- 1 root     root      561 Mar 16 15:55 Employee.yaml
-rw-r--r-- 1 root     root      555 Mar 16 15:55 Users.yaml
rootschema# cat Users.yaml
cubes:
  - name: Users
    sql: SELECT * FROM cubejs.users
      # preAggregations:
      # Pre-Aggregations definitions go here
      # Learn more here: https://cube.dev/docs/caching/pre-aggregations/getting-started
    joins: []
    measures:
      - name: count
        type: count
        drillMembers: [id, name]
    dimensions:
      - name: email
        sql: email
        type: string
      - name: id
        sql: id
        type: number
        primaryKey: true
      - name: name
        sql: name
        type: string
    dataSource: default
```

- Click on “Build” & Choose Filters.



GIT Repo Link: https://github.com/sarfaraz6677/cloudethix-cube-js.git

-END-