

CA6C1 – DevOps - National Institute of Technology, Trichy

Assignment 2 – Setting Up DOCKER - Workshop

Roll No. 205224005

Q1 (A): Create a Container with PostgresDB or mySQL database installed.

Objective:

To create a Docker container with PostgreSQL database installed, perform basic database operations, and document the process.

Environment Details:

- Docker Version: 3.8
- Operating System: Windows 10
- PostgreSQL Version: 13

Docker Compose Configuration:

The following **docker-compose.yml** file was created to set up the PostgreSQL container:

```
👉 docker-compose.yml
1  version: '3.8'
2
3  services:
4    postgres:
5      image: postgres:13
6      container_name: postgres_container
7      restart: always
8      environment:
9        POSTGRES_USER: admin
10       POSTGRES_PASSWORD: admin123
11       POSTGRES_DB: vreqstdb
12     ports:
13       - "5432:5432"
14     volumes:
15       - pgdata:/var/lib/postgresql/data
16
17   volumes:
18     pgdata:
```

Commands Executed:

- Starting the container: **docker-compose up -d**

```
D:\Project\DevOps\Final Assignment\docker-postgres-db>docker-compose up -d
time="2025-04-20T12:09:55+05:30" level=warning msg="D:\\Project\\DevOps\\Final Assignment\\docker-postgres-db\\docker-compose.yml
id potential confusion"
[+] Running 14/14
  postgres Pulled
    f8afe3b22640 Pull complete
    420af9c31ddb Pull complete
    e45b05d88be2 Pull complete
    ada8823e5b6f Pull complete
    2d9287dc0c9b Pull complete
    623da1635329 Pull complete
    8f010006cabb Pull complete
    fc4323444c9b Pull complete
    51c504225859 Pull complete
    c030864720fa Pull complete
    e222bc95278a Pull complete
    6aaf5665e758 Pull complete
    8687d4c2b8df Pull complete
[+] Running 3/3
  Network docker-postgres-db_default Created
  Volume "docker-postgres-db_pgdata" Created
  Container postgres_container Started
D:\Project\DevOps\Final Assignment\docker-postgres-db>
```

- To check running containers: **docker ps**

```
D:\Project\DevOps\Final Assignment\docker-postgres-db>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
6a1936b0d9e3   postgres:13 "docker-entrypoint.s..." 8 minutes ago  Up 8 minutes  0.0.0.0:5432->5432/tcp             postgres_container
```

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage
0.02% / 800% (8 CPUs available)

Container memory usage
20.28MB / 7.5GB

[Show charts](#)

Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	my_postgres_db	9853700b5cf8	postgres	5432:5432	0%	13 hours ago	▶ ⋮ 🗑
<input type="checkbox"/>	sleepy_perلمان	449f9789ee6d	hello-world		0%	13 hours ago	▶ ⋮ 🗑
<input type="checkbox"/>	jovial_hermann	d204bd6e5d65	hello-world		0%	13 hours ago	▶ ⋮ 🗑
<input type="checkbox"/>	yreqst	-	-	-	0%	12 hours ago	▶ ⋮ 🗑
<input type="checkbox"/>	docker-postgres-db	-	-	-	0.03%	2 minutes ago	▶ ⋮ 🗑

- To connect to PostgreSQL container:
docker exec -it postgres_container psql -U admin -d vreqstdb

PostgreSQL CLI:

- Table creation

```
D:\Project\DevOps\Final Assignment\docker-postgres-db>docker exec -it postgres_container psql -U admin -d vreqstdb
psql (13.20 (Debian 13.20-1.pgdg120+1))
Type "help" for help.

vreqstdb=# CREATE TABLE test_table (
vreqstdb(#       id SERIAL PRIMARY KEY,
vreqstdb(#       name TEXT
vreqstdb(# );
CREATE TABLE
```

- Data Insertion

```
vreqstdb=# INSERT INTO test_table (name) VALUES ('Jenil Prajapati'), ('Het Patel'), ('Mayank');
INSERT 0 3
```

- Select query showing inserted rows

```
vreqstdb=# SELECT * FROM test_table;
 id |      name
-----+-----
  1 | Jenil Prajapati
  2 | Het Patel
  3 | Mayank
(3 rows)
```

- Stopping the container: **docker-compose down**

```
D:\Project\DevOps\Final Assignment\docker-postgres-db>docker-compose down
[+] Running 2/2
  Container postgres_container      Removed
  Network docker-postgres-db_default Removed
```

Summary:

- A PostgreSQL container was created and configured successfully utilizing Docker Compose.
- Fundamental database activities such as creating tables, inserting data, and querying data were done.
- Environment variables were utilized in order to set up the PostgreSQL username, password, and database.
- A Docker volume was setup to retain data.
- Screenshots were taken in order to capture the running container and SQL command execution.

Conclusion: This task demonstrated the setup and basic use of a PostgreSQL database within a Docker container environment, providing hands-on experience with containerized database deployment and management.

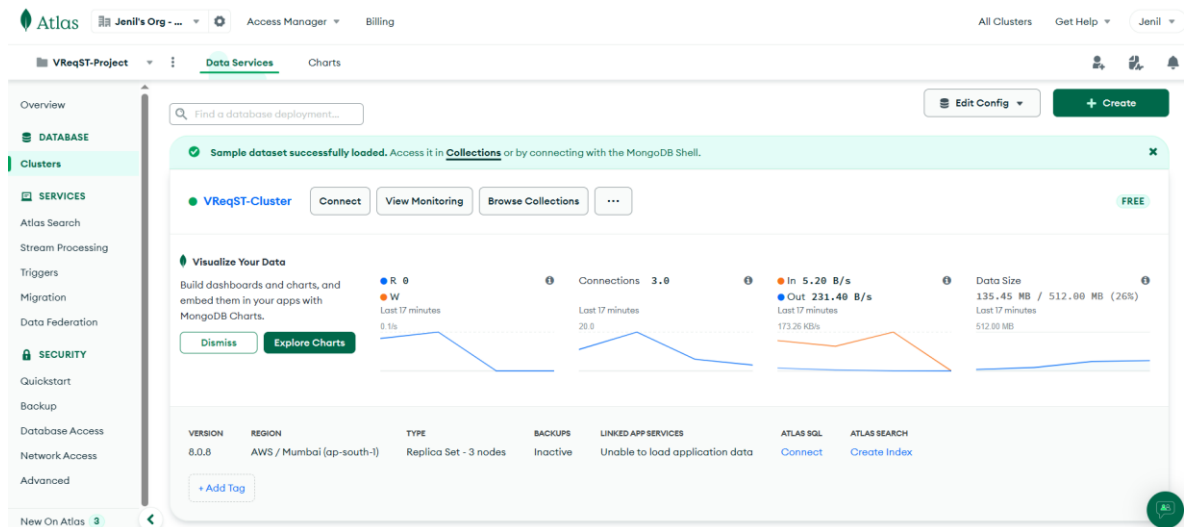
Q1 (B): Deploy VReqST – A requirement specification tool in a container.

Objective:

To deploy the VReqST (Virtual Reality Requirement Specification Tool) application using Docker, containerizing both the application and the associated MongoDB database service, ensuring proper database configuration and connectivity.

Setting up MongoDB Database:

- Create a new Project within MongoDB Atlas.
- Inside the project, create a Database Cluster (free-tier is sufficient for this assignment).



- Once the cluster is created, define four collections within a new database using the following names:
 - customrules
 - jsons
 - projects
 - users

The screenshot shows the MongoDB Atlas console for the 'vreqstadb' database. The 'Create Database' button is visible at the top. The database is currently empty, with a search bar for namespaces. The 'vreqstadb' database is selected, and a table lists four collections: 'customrules', 'jsons', 'projects', and 'users'. Each collection has 0 documents, 0B logical data size, 0B avg document size, 4KB storage size, 1 index, 4KB index size, and 4KB avg index size. A 'CREATE COLLECTION' button is located at the top right of the table.

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
customrules	0	0B	0B	4KB	1	4KB	4KB
jsons	0	0B	0B	4KB	1	4KB	4KB
projects	0	0B	0B	4KB	1	4KB	4KB
users	0	0B	0B	4KB	1	4KB	4KB

- Update the application's server code to replace any local MongoDB connection string (e.g. `mongodb://localhost:27017/vreqst`) with the above cloud-hosted MongoDB Atlas connection string.

Typically, this connection string is found in either:

- `backend/server.js`
- `backend/app.js`
- or a configuration file such as `backend/config.js` or `backend/config/db.js`

Writing the Dockerfile and docker-compose.yml File:

Create a **Dockerfile** file in the project root directory with the following code:

```
Dockerfile
1 FROM node:14
2
3 WORKDIR /app
4
5 COPY . .
6
7 WORKDIR /app/backend
8 RUN npm install
9
10 WORKDIR /app/validation_server
11 RUN npm install
12
13 WORKDIR /app/frontend
14 RUN npm install
15 RUN npm run client-install
16
17 EXPOSE 3000 5001 5002
18
19 CMD ["bash", "-c", "cd /app/backend && npx nodemon index.js & cd /app/validation_server && npx nodemon index.js & cd /app/frontend && npm run dev"]
```

Create a **docker-compose.yml** file in the project root directory with the following configuration:

```
🐳 docker-compose.yml
1  services:
2      vreqst-app:
3          build: .
4          ports:
5              - "3000:3000"
6              - "5001:5001"
7              - "5002:5002"
8          depends_on:
9              - mongo
10     mongo:
11         image: mongo
12         ports:
13             - "27017:27017"
14         volumes:
15             - mongo-data:/data/db
16
17     volumes:
18         mongo-data:
```

Explanation:

- The vreqst-app service builds the VReqST application from the Dockerfile in the current directory and maps necessary ports.
- The mongo service pulls the official MongoDB image and binds port 27017.
- mongo-data volume ensures data persistence for MongoDB.

Building and Running the Docker Containers:

Building and running: **docker-compose up --build**

This command will:

- Build the Docker images.
- Start both vreqst-app and mongo services.
- Map ports as defined in docker-compose.yml

```
D:\Project\DevOps\Final Assignment\Vreqst\docker-compose up --build
time="2025-04-20T18:35:41+05:30" level=warning msg="D:\Project\DevOps\Final Assignment\Vreqst\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 360.4s (26/28) FINISHED
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 161B
=> [validation_server internal] load build definition from Dockerfile
=> => transferring dockerfile: 167B
=> [frontend internal] load build definition from Dockerfile
=> => transferring dockerfile: 163B
=> [backend internal] load metadata for docker.io/library/node:14
=> [validation_server auth] library/node/pull token for registry-1.docker.io
=> [validation_server internal] load .dockerignore
=> => transferring context: 2B
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [validation_server 1/5] FROM docker.io/library/node:14@sha256:a158d3b9bdc3fa813fa6c8c590b8f0a860e015adde59b8ce5744d2f6fd8461aa
=> resolve docker.io/library/node:14@sha256:a158d3b9bdc3fa813fa6c8c590b8f0a860e015adde59b8ce5744d2f6fd8461aa
=> sha256:5f2eddc3f22b6dd4df571e8083277355a2baef5203c4f665f0837ba590 35.24MB / 35.24MB
=> sha256:0d27a8e861329007574c676fba946d68e20d2ce8e64e873de352603f22c4ceb 459B / 459B
=> sha256:0c8cc2f24a4dc6d6e02e086fc944600a541e8acd9ad72d2e90df3ba22f15b3 2.29MB / 2.29MB
=> sha256:6f51ee005deac0d99898d4188ce08bf250ebela31a0b03f613aecbbbc983d8 4.19kB / 4.19kB
=> sha256:d940df5904311ce280e92925797b8cde8093c29ed7244ffba823d1b1569 191.80MB / 191.80MB
=> sha256:1de76e268b103d05fa8960e077951ff54b912b63429c34f5d6adfa09f5f9ee2 51.38MB / 51.88MB
=> sha256:3d2201bd995ccc12851a50820de03d34a17011dcbb9ac9fd3a50c952cbb131 10.00MB / 10.00MB
=> sha256:b253aeafaa7e0671bb6008df01de101a38a045ff7bc656e3b0bf7c7c05cca5 7.80MB / 7.80MB
=> sha256:2ff1df741c74a25258bfa6f08ad0ba727f04518f55f05ca845abc747976c408 50.45MB / 50.45MB
=> extracting sha256:2ff1df741c74a25258bfa6f08ad0ba727f04518f55f05ca845abc747976c408 6.4s
=> extracting sha256:b253aeafaa7e0671bb6008df01de101a38a045ff7bc656e3b0bf7c7c05cca5 1.1s
=> extracting sha256:3d2201bd995ccc12851a50820de03d34a17011dcbb9ac9fd3a50c952cbb131 0.7s
=> extracting sha256:1de76e268b103d05fa8960e077951ff54b912b63429c34f5d6adfa09f5f9ee2 5.4s
=> extracting sha256:d940df5904311ce280e92925797b8cde8093c29ed7244ffba823d1b1569 12.7s
=> extracting sha256:6f51ee005deac0d99898d4188ce08bf250ebela31a0b03f613aecbbbc983d8 0.2s
=> extracting sha256:3f2eddc3f22b6dd4df571e8083277355a2baef5203c4f665f0837ba590 5.3s
=> extracting sha256:0c8cc2f24a4dc6d6e02e086fc944600a541e8acd9ad72d2e90df3ba22f15b3 0.2s
=> extracting sha256:0d27a8e861329007574c676fba946d68e20d2ce8e64e873de352603f22c4ceb 0.1s
=> [validation_server internal] load build context
=> => transferring context: 31.69MB
=> [frontend internal] load build context
=> => transferring context: 662.32MB
=> [frontend 2/5] WORKDIR /app
=> [validation_server 3/5] COPY package*.json ./
=> [backend 3/5] COPY package*.json ./
=> [validation_server 4/5] RUN npm install
=> [backend 4/5] RUN npm install
=> [validation_server 5/5] COPY . .
=> [validation_server] exporting to image
=> => exporting layers
=> => exporting manifest sha256:17577a70433431d7aed18a83f246d6d744806d095823deea4771631318a1128b
=> => exporting config sha256:bb97a2a88771fac7f68cdcccd47a76c10221775960bd644b51ee2273967be56
=> => exporting attestation manifest sha256:0ebcb3b773a2d8323920b6c8a79ebad94e34d0b9f13eeceF4982095b049127f71
=> => exporting manifest list sha256:142e50201da08a76b6eaa09e75f1ac9e8a8f788aa30280daf7660fd2c6d1a5d
=> => naming to docker.io/library/vreqst-validation_server:latest
=> => unpacking to docker.io/library/vreqst-validation_server:latest
=> [validation_server] resolving provenance for metadata file
=> [backend 5/5] COPY . .
=> [backend] exporting to image
=> => exporting layers
=> => exporting manifest sha256:7533ba38e0bed396f982b10180223592b93e9991316b0d35840f00a65766da2
=> => exporting config sha256:1be18f4003472becd5fb50e0913cc0882e2703ea93a855af1f9f698e528a779
=> => exporting attestation manifest sha256:0b300f15fa958b28ec7caa0262caf155f1c1c4f8132e3533d090a8e5ccc482
=> => exporting manifest list sha256:a0f4c2f13abaf040bcf68398b7420a52d1e922de958b50da8712c012elfc1b5
=> => naming to docker.io/library/vreqst-backend:latest
=> => unpacking to docker.io/library/vreqst-backend:latest
=> [frontend 3/5] COPY package*.json ./
=> [frontend 4/5] RUN npm install
=> [backend] resolving provenance for metadata file
=> [frontend 5/5] COPY . .
=> [frontend] exporting to image
=> => exporting layers
=> => exporting manifest sha256:Bce45161450f5b000ef11f351360b295bef7e1ab302d4b39b445753f685ab974
=> => exporting config sha256:9f6197f47626fc2d3a0314dcb3c443600a90cd4732e8a2f14cd4a004f7035c
=> => exporting attestation manifest sha256:aad192b9dcb31c516332c0c56eeaa6c276dd5459ab9181ca187b2e430297154
=> => exporting manifest list sha256:f1a0b06c7342710398a8e42d1f80f4d1caaa0a825f6f36de59683f30aadbfee
=> => naming to docker.io/library/vreqst-frontend:latest
=> => unpacking to docker.io/library/vreqst-frontend:latest
=> [frontend] resolving provenance for metadata file
[+] Running 3/3
  backend          Built
  vreqst-frontend  Built
  validation_server Created
  Container vreqst-backend-1 Created
  Container vreqst-validation_server-1 Created
  Container vreqst-frontend-1 Created
Attaching to backend-1, frontend-1, validation_server-1
```

Running Container

Containers

View all your running containers and applications. [Learn more](#)

Container CPU usage

0.79% / 800% (8 CPUs available)

Container memory usage

794.1MB / 7.5GB

Show charts

Q Search

Only show running containers

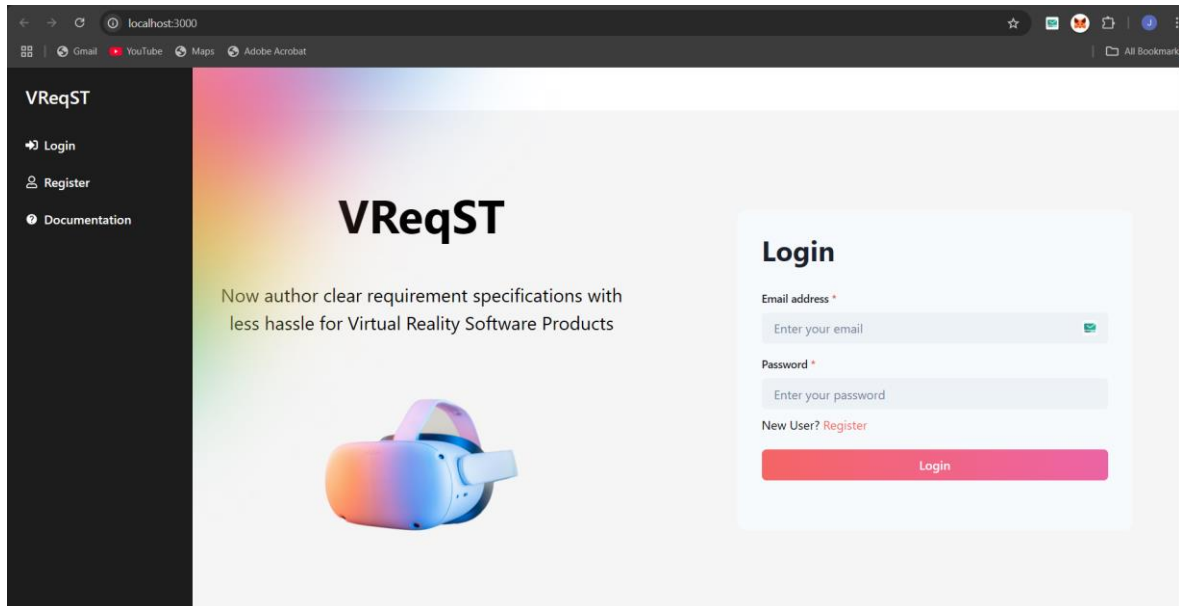
<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input checked="" type="checkbox"/>	vreqst	-	-	-	0.79%	7 minutes ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	mongo-1	66566cd758e9	mongo	27017:27017	0.79%	8 minutes ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	vreqst-app-1	112f79e996d3	vreqst-vreqst-app	3000:3000	0%	7 minutes ago	<div><div></div><div></div><div></div></div>

Verifying Application and Database

Application Access:

Once the containers are running, access the application through:

- <http://localhost:3000>



MongoDB Access assumption :

If connecting to a locally running Mongo container, MongoDB would be accessible on `mongodb://localhost:27017`.

However, as per our configuration, we are using **MongoDB Atlas**, so no local connection is necessary after linking the Atlas connection string in the server code.

Outcome:

At the end of this task:

- The VReqST application runs inside a Docker container.
- It is connected to a cloud-hosted MongoDB Atlas database instance.
- Application services are accessible via defined ports.