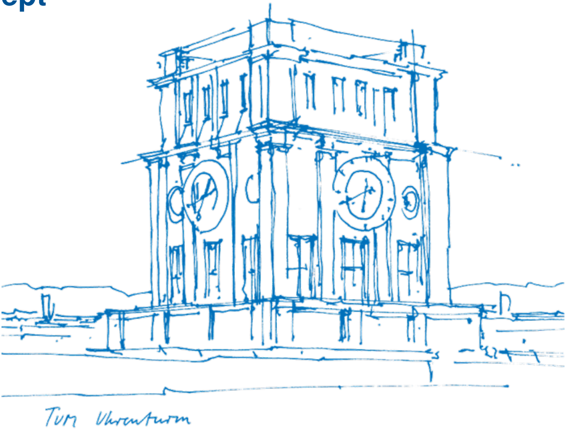
Please adjust group name in tumuser.sty.
Department of Electrical and Computer Engineering
Technical University of Munich

Docker & MongoDB

Develop Containerized MongoDB Concept



Malek Miled

February 3rd, 2024

Outline

1. Task/ Idea:
2. Solution/Idea:

Concept

Using MongoDB in a containerized fashion

- In the context of task76, we will create a concept of hosting a MongoDB in a containerized setting.
- **Goals:**
  - □ Seperate MongoDB intances for different users.
  - □ One main DB, where changes can be seen in real time by all users.
- **Important Notes:** Conflicts that are generated through the simultanious work of several people on the same Database should be avoided.
- **Prerequisites:** In order to run the following instructions, you will need to install Docker Desktop. You can find the installation instructions for your operating system on their website.

Outline

1. Task/ Idea:

2. Solution/Idea:

Running MongoDB as a Docker Container

1. Start Docker.
2. (Optional) Pull the MongoDB image using the terminal: `docker pull mongo:latest`.
3. Create a directory.
4. In the directory, run the image.

```
docker run -d -p 28105:27017 -v ~/<directory_name>:/data/db
        --name <container_name> mongo:latest
```

- `-d`: Run the container in the background (detached mode/return you to the terminal).
- `-p 28105:27017`: Map port 28105 on the host to port 27017 on the container.
- `-v ˜/<directory_name>:/data/db`: Mount the host directory to the container’s `/data/db` directory for persistent data storage.
- `--name <container_name>`: Assign a name to the container for easier identification.
- `mongo:latest`: Specify the Docker image to use, in this case, the latest version.

Playing around with MongoDB

- Accessing the DB from within the container:
  - Execute `docker ps` to see all running containers in detail.
  - Execute bash inside the container: `docker exec -it <container_name> bash`.
  - To directly use mongosh on the DB, just use `mongosh` instead of `bash`.
  - From now on, you can work on your containerized MongoDB like you usually do.
  - Even if you delete the MongoDB container, rerunning the Command n°4 from slide 3 will restore all data that was saved there thanks to the Docker volume we created with the `-v` option.
- Accessing the DB from outside of the container:
  - To access MongoDB remotely, use the following command: (Change the port as needed)

    ```
    mongosh --host <host_IP_address> --port 28105
    ```

  - To obtain the host IP address, run the following command in **bash** and the top address is the one needed: `ifconfig | grep "inet"`

Extras

- This concept can be improved later on by creating a dedicated Dockerfile for more controll over the container.
- Any port can be exposed instead of "28105". Make sure the used port is Free to avoid conflicts.
- On this website, you can check which version of MongoDB you can use instead of "latest".
- For more customization, please refer to the official documentation.

Thank You

Thank you for your attention!