Advanced Docker Concepts

Nicolas El Khoury

Introduction

Now that we learned the basics of containers and Docker, we will take it to the next level in this demo. We will containerize and deploy the NK-backend application that we previously deployed on an Ubuntu VM. To do so, we will deploy the containerized version of the Arango database, as well as the NK Backend Service. The following steps will be completed:

- Deploy the Containerized version of ArangoDB without data persistence.
- Enable data persistence using Docker volumes.
- Build the backend service using a Dockerfile.
- Deploy the containerized version of the backend service.
- Perform API requests to validate the deployment.

Solution

ArangoDB Deployment

No Data Persistence

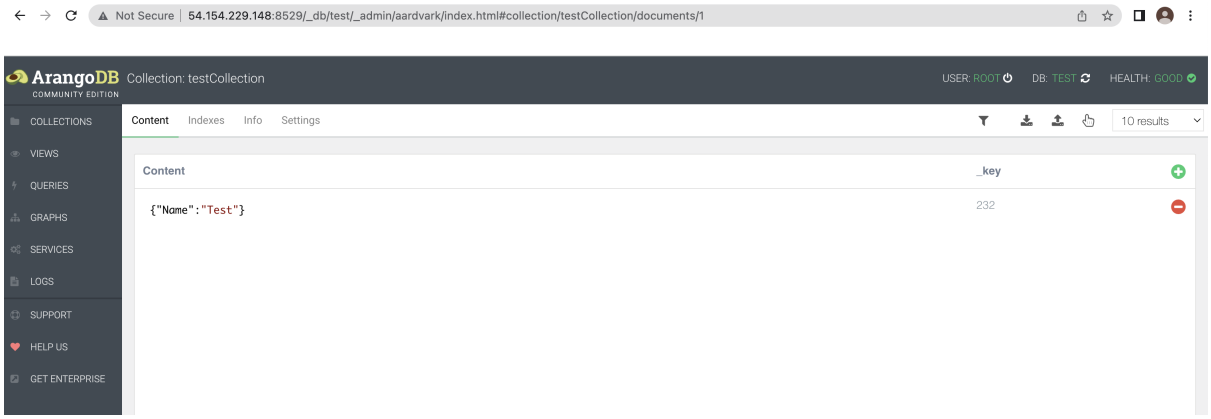
- Create an ArangoDB Docker container:

```
docker run -d --name person-db -p 8529:8529 -e ARANGO_STORAGE_ENGINE=rocksdb -e ARANGO
_ROOT_PASSWORD=rootPassword arangodb/arangodb:3.6.3
```

- Ensure the image is pulled `docker images` and the container is running `docker ps -a`:

```
ubuntu@ip-172-31-39-13:~$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
arangodb/arangodb   3.6.3     151ed6068251   2 years ago   310MB
ubuntu@ip-172-31-39-13:~$ docker ps -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                                                 NAMES
8936407274b7   arangodb/arangodb:3.6.3   "/entrypoint.sh aran…"   24 seconds ago   Up 22 seconds   0.0.0.0:8529->8529/tcp, :::8529->8529/tcp   person-db
ubuntu@ip-172-31-39-13:~$
```

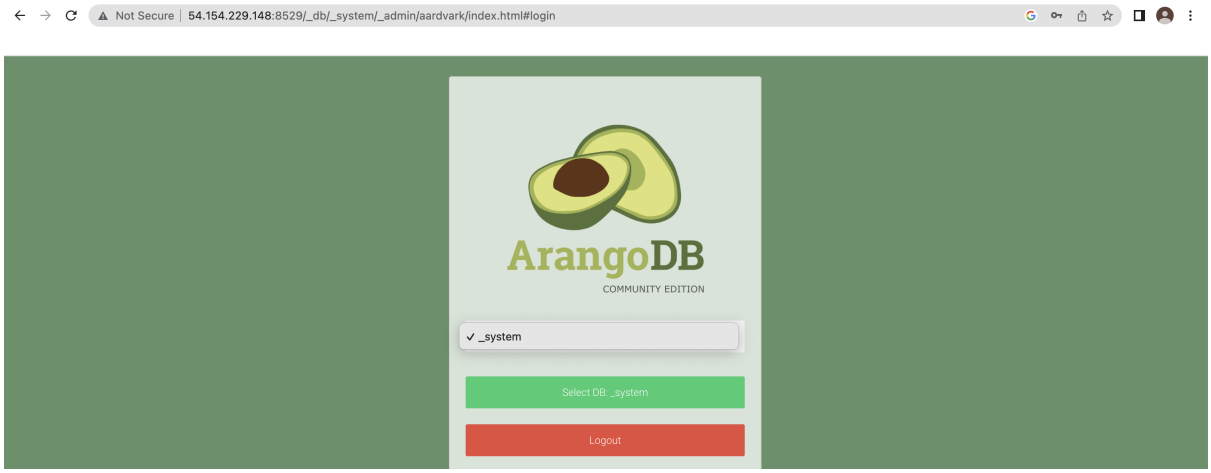
- Login to the database (***username**: root, **password**:* rootPassword), and create some data for testing purposes.



ArangoDB is now deployed as a Docker container, and is able to serve requests on port **8529**. However, if the container fails, the data will disappear. To verify, completely remove the container, and then start a new one, simulating a total failure:

```
# Remove the container
docker rm -f person-db
# Ensure that the container is totally removed
docker ps -a
# Recreate the container with the same command used
docker run -d --name person-db -p 8529:8529 -e ARANGO_STORAGE_ENGINE=rocksdb -e ARANGO
_ROOT_PASSWORD=rootPassword arangodb/arangodb:3.6.3
```

- Logging back into the management console clearly shows that all the databases, collections, and data entered are now missing.



Data Persistence

- Create an ArangoDB container with a named volume:

```
# Delete the container
docker rm -f person-db
# Make sure the container is deleted
docker ps -a
# Recreate the container with a named volume
docker run -d --name person-db -p 8529:8529 -v arango-volume:/var/lib/arangodb3 -e ARA
NGO_STORAGE_ENGINE=rocksdb -e ARANGO_ROOT_PASSWORD=rootPassword arangodb/arangodb:3.6.
3
```

- Examine the Volume:

```
# Check if the container is successfully up
docker ps -a
# List all the available volumes. The picture clearly shows the creation of the arango
-volume volume
```

```
docker volume ls
# Navigate inside the volume directory (sudo permissions are needed). Clearly, the vol
ume is created inside the directory.
sudo ls -lah /var/lib/docker/volumes
# Navigate inside the volume directory. Each volume is created in such a format: /var/
lib/docker/volumes/<volume>/_data/
sudo ls -lah /var/lib/docker/volumes/arango-volume
# Inspect what's inside the _data directory. Evidently, it is Arango's Data
sudo ls -lah /var/lib/docker/volumes/arango-volume/_data
# Exec into the container's data directory. The same data that was found on the host v
olume, is present in the container
docker exec -it person-db sh
# Once inside the container, list the content of the data directory
ls -lah /var/lib/arangodb3
```

- Simulate a Failure, and ensure data persistence:
  - Create temporary data (e.g., database, collection, data)
  - Remove the container: `docker rm -f person-db`
  - Recreate the container: `docker run -d --name person-db -p 8529:8529 -v arango-volume:/var/lib/arangodb3 -e ARANGO_STORAGE_ENGINE=rocksdb -e ARANGO_ROOT_PASSWORD=rootPassword arangodb/arangodb:3.6.3`
  - Log back into the management console. unlike the previous command, the data created still exists.

NK-backend Service Deployment

- Clone the repository:

```
# Clone the repository
git clone https://github.com/devops-beyond-limits/nk-backend-service.git
# Navigate to the root directory
cd nk-backend-service
```

- Modify the Dockerfile: `nano Dockerfile`
  - Modify the `ARANGODB_HOST` variable to include the machine's public IP.
  - Modify the `ARANGODB_PASSWORD` variable to match that specified on the ArangoDB container.
- Build the Docker image: `docker build -t backend-service:v-Dockerfile -f Dockerfile .`
- List all the available images on the server: `docker images`

```
ubuntu@ip-172-31-39-13:~/nk-backend-service$ docker images
REPOSITORY           TAG                 IMAGE ID       CREATED          SIZE
backend-service      latest              f6c92ae9dc09   18 seconds ago   245MB
node                 14.7.0-alpine3.12   419d77aad1ff   2 years ago      117MB
arangodb/arangodb    3.6.3               151ed6068251   2 years ago      310MB
```

- Run a container from the backend service image: `docker run -d --name backend -p 80:1337 backend-service:v-Dockerfile`
- Ensure that the application connected to the database through the logs: `docker logs backend`

```
ubuntu@ip-172-31-39-13:~$ docker run -d --name backend -p 1337:1337 --restart on-failure:3 -e ARANGODB_HOST="54.154.229.148" backend-service
a813985d2de0014f09f5a97ed32de558005a87779b0c1de3c9322685f76cc780
ubuntu@ip-172-31-39-13:~$ docker logs backend
(node:1) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
 info: /bootstrap.js: Attempting to initialize the required database and collection
 info: /bootstrap.js: fetching the available databases
 info: /bootstrap.js: The database does not have a database named 'persons'. Creating the database...
 info: /bootstrap.js: Successfully created the database:
 info: /bootstrap.js: {"error":false,"code":201,"result":true}
 info: /bootstrap.js: Attempting to create the 'persons' collection
 info: /bootstrap.js: Successfully created the 'persons' collection
 info:
 info:               .-..-.
 info:
 info:    Sails              <|    .-..-.
 info:    v1.2.4              |\
 info:                       /|.\
 info:                      / || \
 info:                    ,'  |'  \
 info:                 .-'.-==|/_--'
 info:                 `--'-------'
 info:    __---___--___---___--___---___--___
 info:  ____---___--___---___--___---___--___-__
 info:
 info: Server lifted in `/app`
 info: To shut down Sails, press <CTRL> + C at any time.
 info: Read more at https://sailsjs.com/support.

debug: -------------------------------------------------------
debug: :: Mon Nov 14 2022 16:37:56 GMT+0000 (Coordinated Universal Time)

debug: Environment : development
debug: Port        : 1337
debug: -------------------------------------------------------
```