Lab: [Optional] Setting Up MongoDB Docker

Estimated time needed: **15** minutes

Objectives

After completing this lab, you will be able to:

- Pull the MongoDB Atlas image to your local machine
- Run the Atlas database using the Atlas Command Line Interface (CLI)

Running the lab

This is an instructional lab and does not require a Skills Network lab environment. You will follow it on your local machine.

1. Pull MongoDB image from DockerHub

MongoDB provides an Atlas image on DockerHub. You can pull this image by using the command:

```
docker pull mongodb/atlas
```

Depending on your machine's network speed and resources, this activity may take a few minutes to complete. After pulling the image, you can verify the image by running `docker images`.

```
$ docker images
REPOSITORY               TAG         IMAGE ID      CREATED       SIZE
docker.io/mongodb/atlas  latest      e8c3a080a6bc  23 hours ago  722 MB
```

2. Run the MongoDB Atlas image

Next, you will use the `docker run` command to run the image exposing port 27017 to a local port of the same number, which allows you to communicate with MongoDB Docker image using `localhost:27017`. Notice you are also running the image as a privileged container. Using the privileged container to run the image is normally not encouraged. However, you will need to use the privileged account to install and run the Atlas CLI in this case.

```
docker run -p 27017:27017 -d --privileged --name mongodb-atlas mongodb/atlas
```

You can refer to the running container using the name `mongodb-atlas` as specified in the `docker run` command above. Let's check the status of this container using its name and the `docker inspect` command.

```
$ docker inspect mongodb-atlas
[
     {
          "Id": "40031588d0390f1a7c868e975306ad885eba9af583f03c5f03cc6e4969c05ef6",
          "Created": "2024-05-20T17:20:02.057270461-07:00",
          "Path": "tail",
          "Args": [
               "-f",
               "/dev/null"
          ],
          "State": {
               "OciVersion": "1.1.0+dev",
               "Status": "running",
               "Running": true,
               "Paused": false,
               "Restarting": false,
               "OOMKilled": false,
               "Dead": false,
               "Pid": 236942,
               "ConmonPid": 236940,
               "ExitCode": 0,
               "Error": "",
               "StartedAt": "2024-05-20T17:20:02.149188599-07:00",
               "FinishedAt": "0001-01-01T00:00:00Z",
               "Health": {
                    "Status": "",
                    "FailingStreak": 0,
                    "Log": null
               },
               "CgroupPath": "/user.slice/user-502.slice/user@502.service/user.slice/libpod-40031588d0390f1a7c868e975306ad885eba9af583f03c5f03cc6e4969c05ef6.scope",
               "CheckpointedAt": "0001-01-01T00:00:00Z",
               "RestoredAt": "0001-01-01T00:00:00Z"
          },
          "Image": "e8c3a080a6bc4d99b96008a0df187153640a386b9ea04c945cc4b0030fe2f512",
```

```
"ImageDigest": "sha256:1c4b233a3da10bd1c2febf662c55bc366354043a504d5ae3d13ed6cb37396381",
"ImageName": "docker.io/mongodb/atlas:latest",
```

The result provides the ImageName, such as `docker.io/mongodb/atlas:latest`. The result also provides other information about the running container.

3. Exec into the container

Exec into the Docker container using the `exec` command.

```
docker exec -it mongodb-atlas bash
```

4. Validate Atlas CLI

The `exec` command should take you inside a Mongo container where you can use the following command to validate that `atlas cli` is available in this Docker container.

```
atlas --version
```

You should see a version number similar to this output:

```
[root@3a5e271a89fa /]# atlas --version
atlascli version: 1.19.0
git version: aff1de1dbdfb031ee5ff9efdcc07632a493b9ffc
Go version: go1.22.1
   os: linux
   arch: arm64
```

```
compiler: gc
```

The command output shows that Atlas CLI version `1.19.0` is available.

5. Deploy local database

You will need to set up an `atlas deployment` using the following command:

```
atlas deployments setup --bindIpAll --username root --password root --type local --force
```

This step downloads a local version of Atlas and might take 5-10 minutes to complete. Also, note that you will never add the username or password in plain text to the CLI in a production environment. You are adding the username or password in plain text since this is a lab environment, which will help simplify the step. You can choose your username and password. **Note your username and password**, as you will use them later in the lab.

The output should look something like this:

```
[root@40031588d039 /]# atlas deployments setup --bindIpAll --username root --password root --type local --force
[Default Settings]
Deployment Name   local9684
MongoDB Version   7.0
Port              27017
Creating your cluster local9684 [this might take several minutes]
1/3: Starting your local environment...
2/3: Downloading the MongoDB binaries to your local environment...
3/3: Creating your deployment local9684...
Deployment created!
Connection string: mongodb://root:root@localhost:27017/?directConnection=true
connection skipped
```

Congratulations! You successfully installed MongoDB locally on your machine using Docker.

Authors

UL