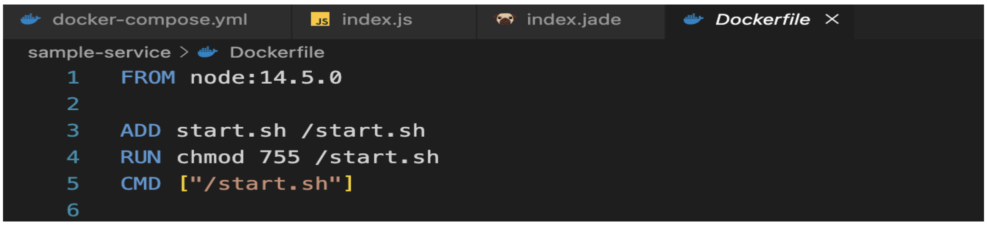
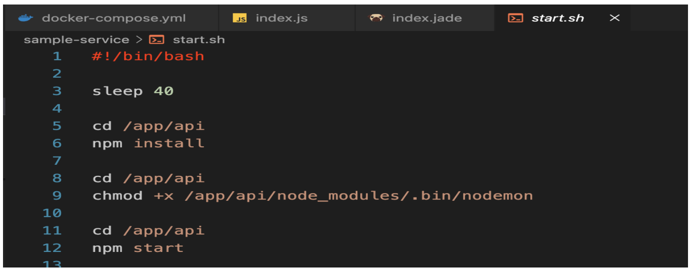
CO515-2024 :Advances in Computer Networks:Selected Topics

E/18/168
KARAN R

1.Create Dockerfiles:

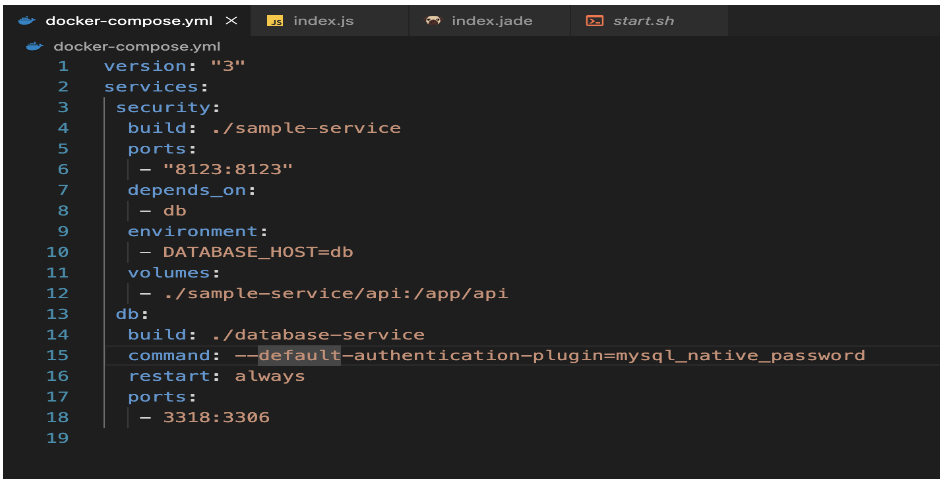


```
FROM node:14.5.0

ADD start.sh /start.sh
RUN chmod 755 /start.sh
CMD ["/start.sh"]

```



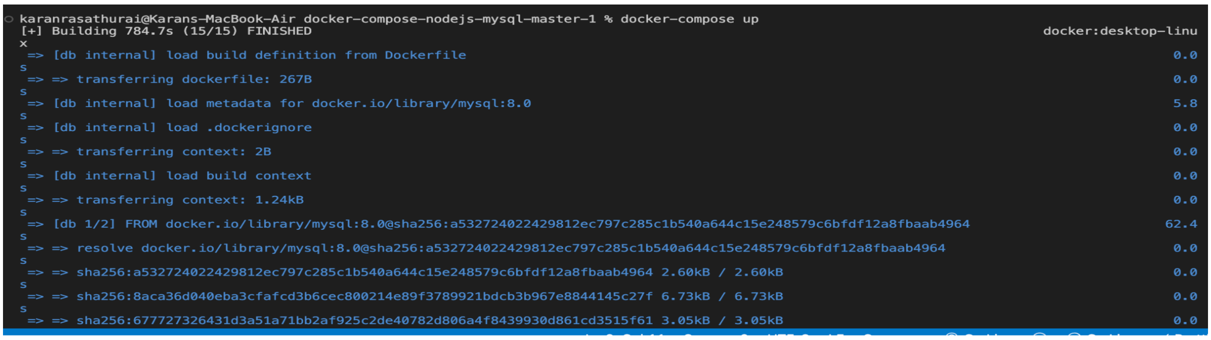
```
#!/bin/bash

sleep 40

cd /app/api
npm install

cd /app/api
chmod +x /app/api/node_modules/.bin/nodemon

cd /app/api
npm start
```

2. Write docker-compose.yml:

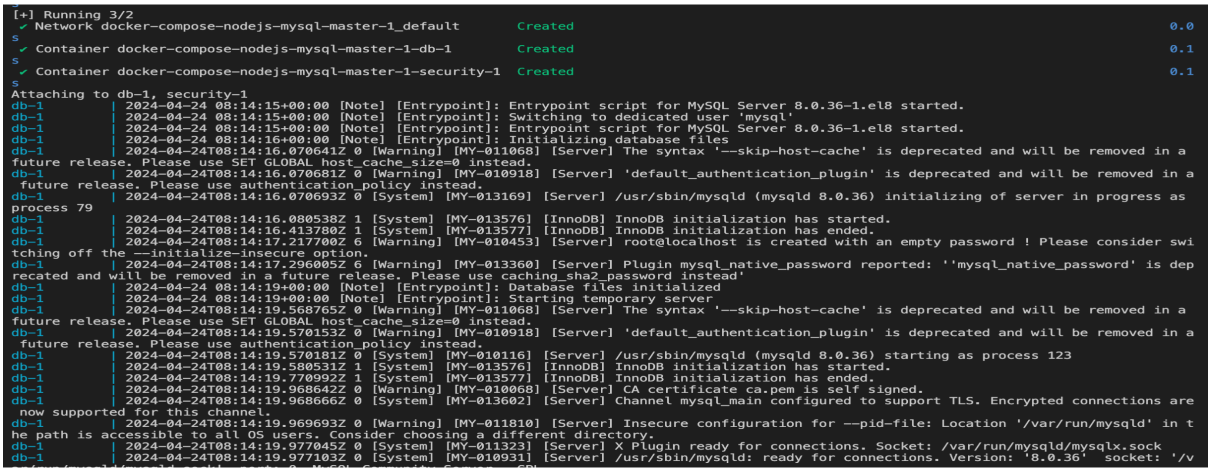


```
version: "3"
services:
  security:
    build: ./sample-service
    ports:
      - "8123:8123"
    depends_on:
      - db
    environment:
      - DATABASE_HOST=db
    volumes:
      - ./sample-service/api:/app/api
  db:
    build: ./database-service
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    ports:
      - 3318:3306

```

3.Build and Run the Application:



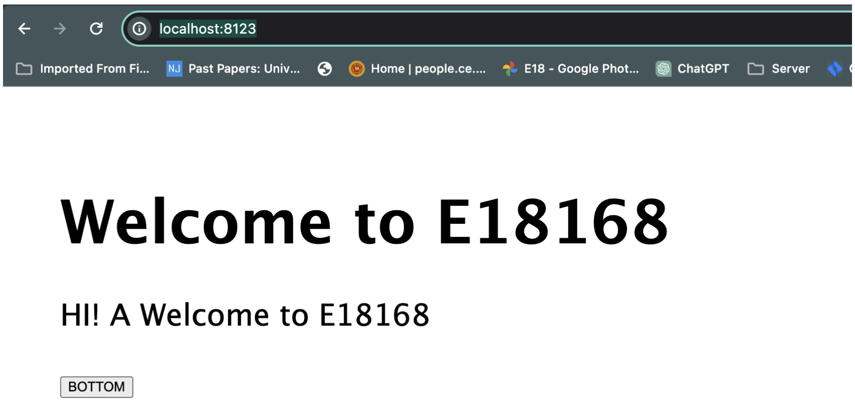
```
○ karanrasathurai@Karans-MacBook-Air docker-compose-nodejs-mysql-master-1 % docker-compose up
 [+] Building 784.7s (15/15) FINISHED                                                docker:desktop-linu
 x
  => [db internal] load build definition from Dockerfile                                               0.0
 s
  => => transferring dockerfile: 267B                                                                  0.0
 s
  => [db internal] load metadata for docker.io/library/mysql:8.0                                       5.8
 s
  => [db internal] load .dockerignore                                                                  0.0
 s
  => => transferring context: 2B                                                                       0.0
 s
  => [db internal] load build context                                                                  0.0
 s
  => => transferring context: 1.24kB                                                                   0.0
 s
  => [db 1/2] FROM docker.io/library/mysql:8.0@sha256:a532724022429812ec797c285c1b540a644c15e248579c6bfdf12a8fbaab4964     62.4
 s
  => => resolve docker.io/library/mysql:8.0@sha256:a532724022429812ec797c285c1b540a644c15e248579c6bfdf12a8fbaab4964        0.0
 s
  => => sha256:a532724022429812ec797c285c1b540a644c15e248579c6bfdf12a8fbaab4964 2.60kB / 2.60kB                           0.0
 s
  => => sha256:8aca36d040eba3cfafcd3b6cec800214e89f3789921bdcb3b967e8844145c27f 6.73kB / 6.73kB                           0.0
 s
  => => sha256:677727326431d3a51a71bb2af925c2de40782d806a4f8439930d861cd3515f61 3.05kB / 3.05kB                           0.0
```



```
[+] Running 3/2
 ✔ Network docker-compose-nodejs-mysql-master-1_default       Created                                  0.0
s
 ✔ Container docker-compose-nodejs-mysql-master-1-db-1        Created                                  0.1
s
 ✔ Container docker-compose-nodejs-mysql-master-1-security-1  Created                                  0.1
s
Attaching to db-1, security-1
db-1          | 2024-04-24 08:14:15+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.36-1.el8 started.
db-1          | 2024-04-24 08:14:15+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db-1          | 2024-04-24 08:14:15+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.36-1.el8 started.
db-1          | 2024-04-24 08:14:16+00:00 [Note] [Entrypoint]: Initializing database files
db-1          | 2024-04-24T08:14:16.070641Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a
future release. Please use SET GLOBAL host_cache_size=0 instead.
db-1          | 2024-04-24T08:14:16.070681Z 0 [Warning] [MY-010918] [Server] 'default_authentication_plugin' is deprecated and will be removed in a
 future release. Please use authentication_policy instead.
db-1          | 2024-04-24T08:14:16.070693Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.36) initializing of server in progress as
process 79
db-1          | 2024-04-24T08:14:16.080538Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db-1          | 2024-04-24T08:14:16.413780Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db-1          | 2024-04-24T08:14:17.217700Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider swi
tching off the --initialize-insecure option.
db-1          | 2024-04-24T08:14:17.296005Z 6 [Warning] [MY-013360] [Server] Plugin mysql_native_password reported: ''mysql_native_password' is dep
recated and will be removed in a future release. Please use caching_sha2_password instead'
db-1          | 2024-04-24 08:14:19+00:00 [Note] [Entrypoint]: Database files initialized
db-1          | 2024-04-24 08:14:19+00:00 [Note] [Entrypoint]: Starting temporary server
db-1          | 2024-04-24T08:14:19.568765Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a
future release. Please use SET GLOBAL host_cache_size=0 instead.
db-1          | 2024-04-24T08:14:19.570153Z 0 [Warning] [MY-010918] [Server] 'default_authentication_plugin' is deprecated and will be removed in a
 future release. Please use authentication_policy instead.
db-1          | 2024-04-24T08:14:19.570181Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.36) starting as process 123
db-1          | 2024-04-24T08:14:19.580531Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db-1          | 2024-04-24T08:14:19.770992Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db-1          | 2024-04-24T08:14:19.968642Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
db-1          | 2024-04-24T08:14:19.968666Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are
 now supported for this channel.
db-1          | 2024-04-24T08:14:19.969693Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in t
he path is accessible to all OS users. Consider choosing a different directory.
db-1          | 2024-04-24T08:14:19.977045Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: /var/run/mysqld/mysqlx.sock
db-1          | 2024-04-24T08:14:19.977103Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.36'  socket: '/v
```

4.Access the Application:

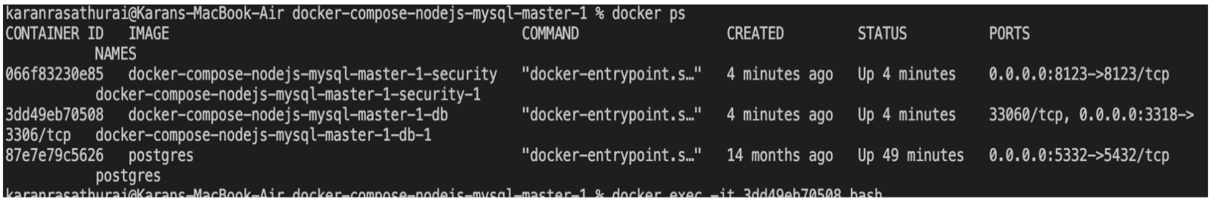
http://localhost:8123/



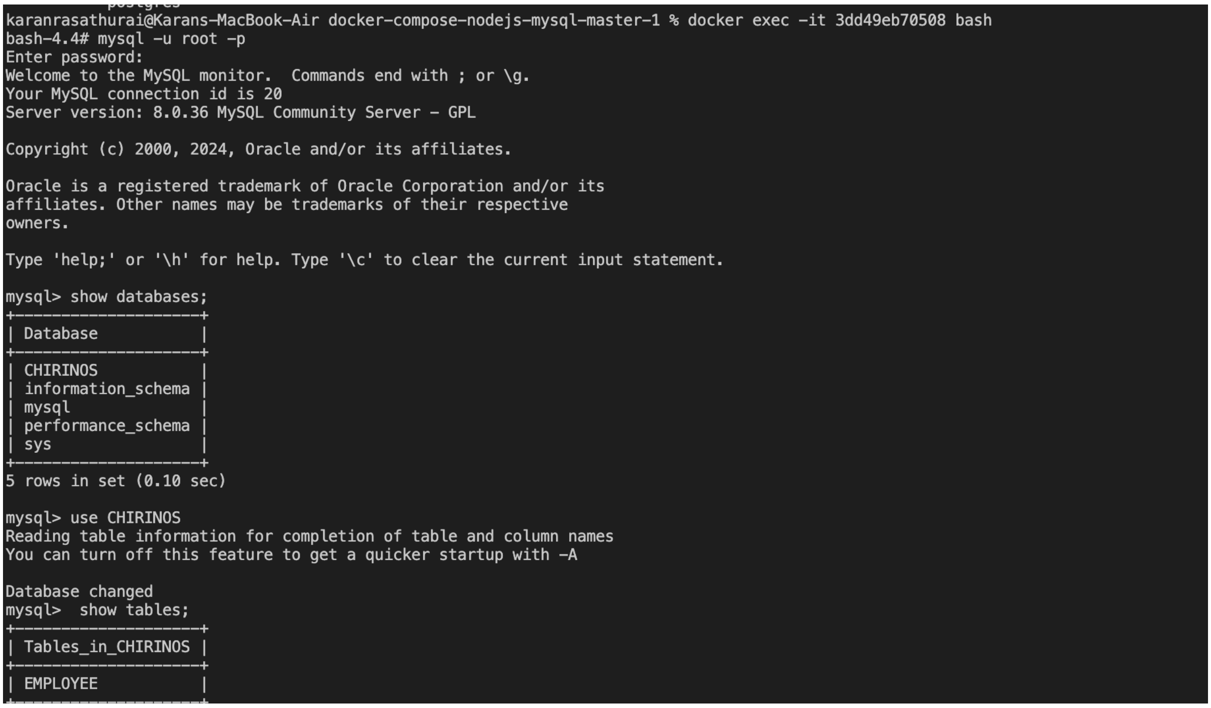
Welcome to E18168

HI! A Welcome to E18168

5.interact with the application to see how it communicates with the database.

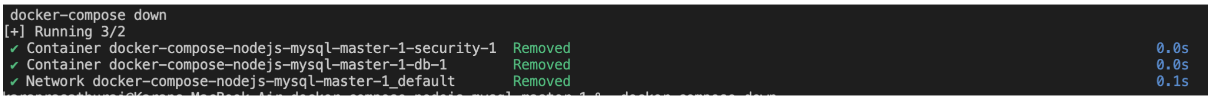


```
karanrasathurai@Karans-MacBook-Air docker-compose-nodejs-mysql-master-1 % docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED          STATUS           PORTS
          NAMES
066f83230e85   docker-compose-nodejs-mysql-master-1-security   "docker-entrypoint.s…"   4 minutes ago    Up 4 minutes     0.0.0.0:8123->8123/tcp
          docker-compose-nodejs-mysql-master-1-security-1
3dd49eb70508   docker-compose-nodejs-mysql-master-1-db         "docker-entrypoint.s…"   4 minutes ago    Up 4 minutes     33060/tcp, 0.0.0.0:3318->
3306/tcp   docker-compose-nodejs-mysql-master-1-db-1
87e7e79c5626   postgres                                       "docker-entrypoint.s…"   14 months ago    Up 49 minutes    0.0.0.0:5332->5432/tcp
          postgres
karanrasathurai@Karans-MacBook-Air docker-compose-nodejs-mysql-master-1 % docker exec -it 3dd49eb70508 bash
```



```
          postgres
karanrasathurai@Karans-MacBook-Air docker-compose-nodejs-mysql-master-1 % docker exec -it 3dd49eb70508 bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| CHIRINOS           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.10 sec)

mysql> use CHIRINOS
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>  show tables;
+--------------------+
| Tables_in_CHIRINOS |
+--------------------+
| EMPLOYEE           |
+--------------------+
```

6.Cleanup:



```
 docker-compose down
[+] Running 3/2
 ✔ Container docker-compose-nodejs-mysql-master-1-security-1  Removed                                                        0.0s
 ✔ Container docker-compose-nodejs-mysql-master-1-db-1        Removed                                                        0.0s
 ✔ Network docker-compose-nodejs-mysql-master-1_default       Removed                                                        0.1s
karanrasathurai@Karans-MacBook-Air docker-compose-nodejs-mysql-master-1 %  docker-compose down
```

Link for Repo folder-

https://github.com/rasathuraikaran/docker-compose-nodejs-mysql-master

Discussion Questions:

1 • What is Docker Compose, and how does it differ from Docker?

Docker Compose is a powerful tool that simplifies the management of multi-container applications. It allows you to define and orchestrate multiple services within a single YAML configuration file. With Compose, you can spin up your entire application stack with a single command or tear it down effortlessly. Here are some key points about Docker Compose:

1. Definition and Sharing: You define your application stack in a YAML file, which can be version controlled and kept at the root of your project repository. This makes it easy for others to contribute to your project by cloning the repository and starting the app using Compose.
2. Multi-Container Coordination: Compose coordinates the deployment and interaction of multiple containers, making it simpler to manage complex applications with interdependent services.
3. Efficient Collaboration: Compose configuration files are easy to share, enabling collaboration among developers, operations teams, and other stakeholders. This leads to smoother workflows and faster issue resolution.
4. Rapid Development: Compose caches container configurations, allowing quick restarts of unchanged services. This accelerates development iterations.
5. Portability: Compose supports variables in the configuration file, making it adaptable to different environments or users.
6. Active Community: Docker Compose benefits from a vibrant community, providing abundant resources, tutorials, and support

2• What are the benefits of using Docker Compose for multi-container applications?

1. Fast and Simple Configuration: Compose leverages YAML scripts and environment variables, allowing easy configuration or modification of application services.
2. Secure Internal Communication: Compose creates a network for services to share, enhancing security by preventing external access to services.

3. Portability and CI/CD Support: Define services once in the Compose file, making it easy to share and deploy across different environments.

3• Explain the purpose of each section in the docker-compose.yml file.

The `docker-compose.yml` file consists of several top-level attributes:

1. Version: Specifies the Compose file format version (e.g., `version: '3.8'`).
2. Services: Defines individual services (containers) and their configurations (images, ports, volumes, environment variables, etc.).
3. Networks: Specifies networks for communication between services.
4. Volumes: Defines data volumes for persistent storage.
5. Configs and Secrets: Used for managing configuration files and secrets.
6. Extensions: Additional features or customizations.

4• What are some real-world scenarios where Docker Compose can be useful?

1. Development Environments: Quickly spin up isolated development environments with all dependencies defined in the Compose file.
2. Automated Testing Environments: Create and destroy isolated testing environments for automated testing suites.
3. Single Host Deployments: Use Compose for development and testing workflows on a single host.
4. Microservices Architecture: Manage multiple microservices within a single application stack.
5. Local Development with Databases and Caches: Set up local development environments with databases, caches, and other services.
6. Application Stacks for Demos or Workshops: Easily share and deploy complete application stacks for demos or workshops.