Bharatiya Vidya Bhavans'

Sardar Patel Institute of Technology

Munshinagar, Andheri(W), Mumbai-400058
(Autonomous College Affiliated to University of Mumbai)

Academic Year: 2025 26 **Semester: III** **Class: MCA**
Course Code: MC520 **Course Name: Cloud Computing**

Experiment No.2
Date: 23.09.25

Aim: Ubuntu: Development of an application using Docker and Docker Compose

CO Mapping – OECS1.4

Objective: To understand and implement containerization techniques in Ubuntu using Docker and Docker Compose for developing, deploying, and managing applications efficiently with isolated, reproducible environments.

Concept:
Docker is an open-source platform that automates the deployment, scaling, and management of applications inside lightweight, portable containers.

- A container is an isolated unit that packages an application with all its dependencies, libraries, and configuration files, ensuring it runs the same in any environment.
- Docker uses the Docker Engine to run containers and images (read-only templates) to create them.
- It eliminates the "works on my machine" problem by ensuring environment consistency.

Lab Exercise:

Step 1: Update Ubuntu 22.04 system
sudo apt update

Step 2: Install prerequisites
sudo apt install apt-transport-https ca-certificates curl software-properties-common

Step 3: Add Docker GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Step 4: Add Docker repository
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Step 5: Update package list

```
sudo apt update
```

Step 6: Install Docker

```
sudo apt install docker-ce
```

Step 7: Check Docker status

```
sudo systemctl status docker
```

Step 8: Verify installation

```
sudo docker run hello-world
```

Step 9: Install Docker Compose

```
sudo apt install docker-compose
```

Step 10: Verify Docker Compose

```
docker-compose version
```

Reference: Official Docker Documentation - https://docs.docker.com/engine/install/ubuntu/

Now Create a directory and then make a small project and then create 2 files in same directory Dockerfile and docker-compose.yml

Dockerfile

```
# Build stage
FROM node:18-alpine

WORKDIR /app

# Install dependencies
COPY package*.json ./
RUN npm install –production

# Copy application files
COPY . .

# Expose the application port
EXPOSE 3001

# Start the application
CMD ["node", "index.js"]
```

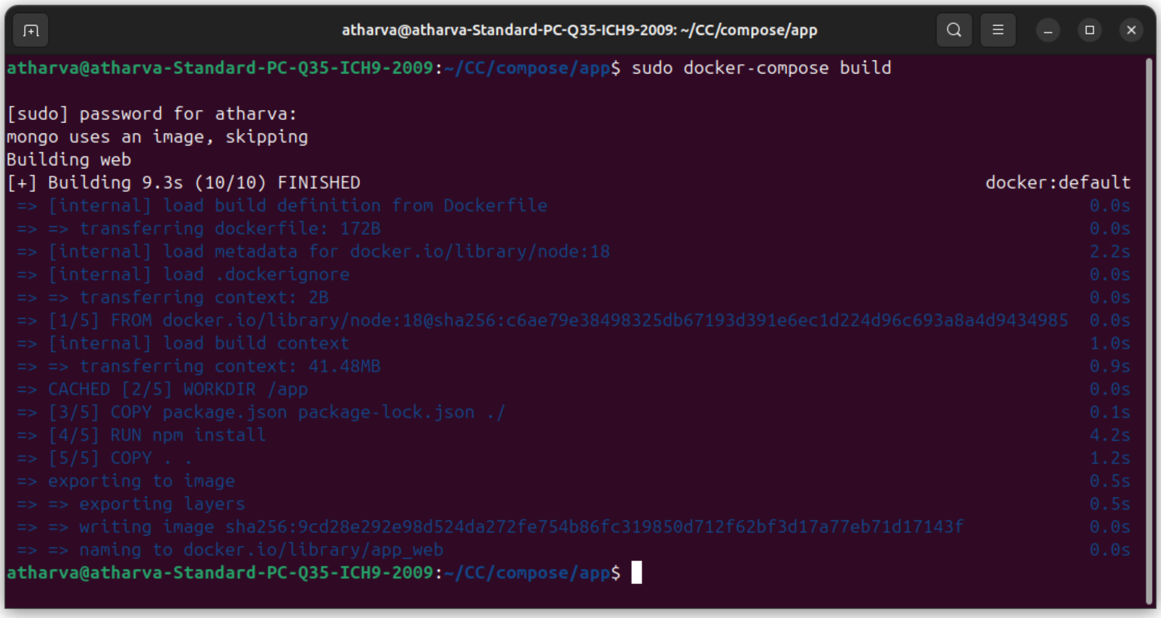
docker-compose.yml

```
version: '1.0'

services:
  web:
    image: todo-app:1.0.0
    build: .
    container_name: web
    ports:
      - "3001:3001"
    environment:
      - NODE_ENV=production
      -
MONGODB_URI=mongodb+srv://angreatharva08_db_user:yYEtVTbasx0uD8Si@cluster0.rhgpdag.mongodb.net/todoapp?retryWrites=true&w=majority
    restart: unless-stopped
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

Then run command sudo docker-compose build

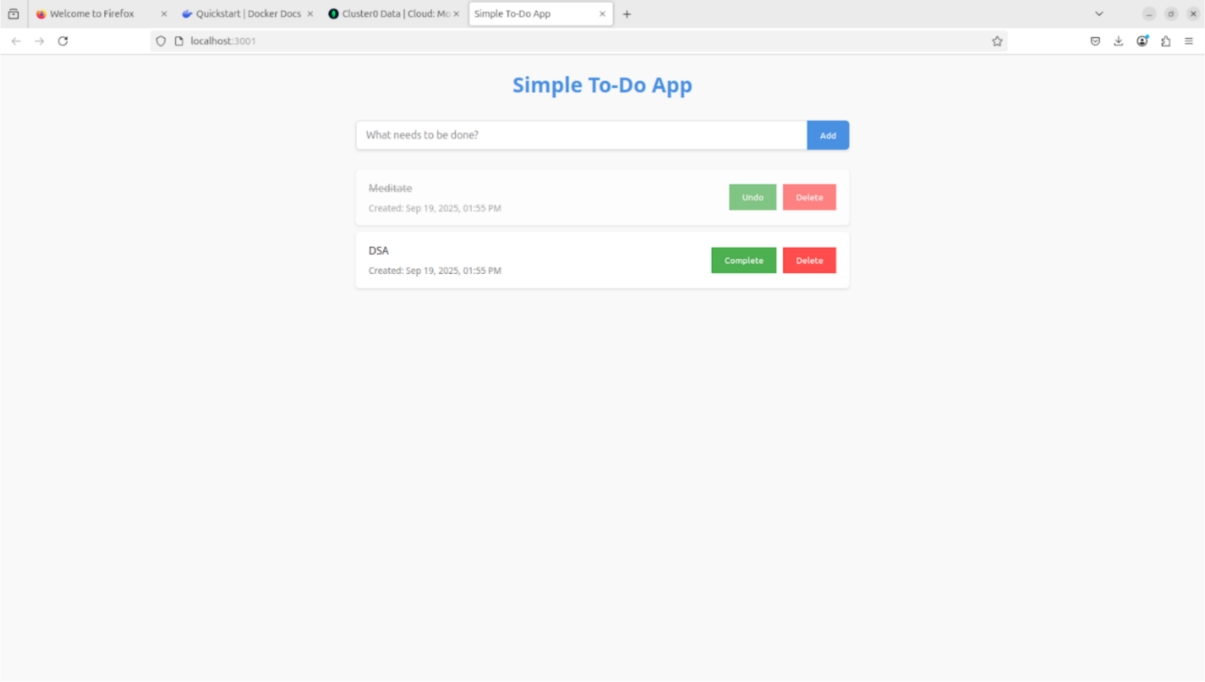


atharva@atharva-Standard-PC-Q35-ICH9-2009: ~/CC/compose/app

```
atharva@atharva-Standard-PC-Q35-ICH9-2009:~/CC/compose/app$ sudo docker-compose build

[sudo] password for atharva:
mongo uses an image, skipping
Building web
[+] Building 9.3s (10/10) FINISHED                                                           docker:default
 => [internal] load build definition from Dockerfile                                                   0.0s
 => => transferring dockerfile: 172B                                                                   0.0s
 => [internal] load metadata for docker.io/library/node:18                                             2.2s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [1/5] FROM docker.io/library/node:18@sha256:c6ae79e38498325db67193d391e6ec1d224d96c693a8a4d9434985  0.0s
 => [internal] load build context                                                                      1.0s
 => => transferring context: 41.48MB                                                                   0.9s
 => CACHED [2/5] WORKDIR /app                                                                          0.0s
 => [3/5] COPY package.json package-lock.json ./                                                       0.1s
 => [4/5] RUN npm install                                                                              4.2s
 => [5/5] COPY . .                                                                                     1.2s
 => exporting to image                                                                                 0.5s
 => => exporting layers                                                                                0.5s
 => => writing image sha256:9cd28e292e98d524da272fe754b86fc319850d712f62bf3d17a77eb71d17143f           0.0s
 => => naming to docker.io/library/app_web                                                             0.0s
atharva@atharva-Standard-PC-Q35-ICH9-2009:~/CC/compose/app$ 
```

Then run command sudo docker-compose up

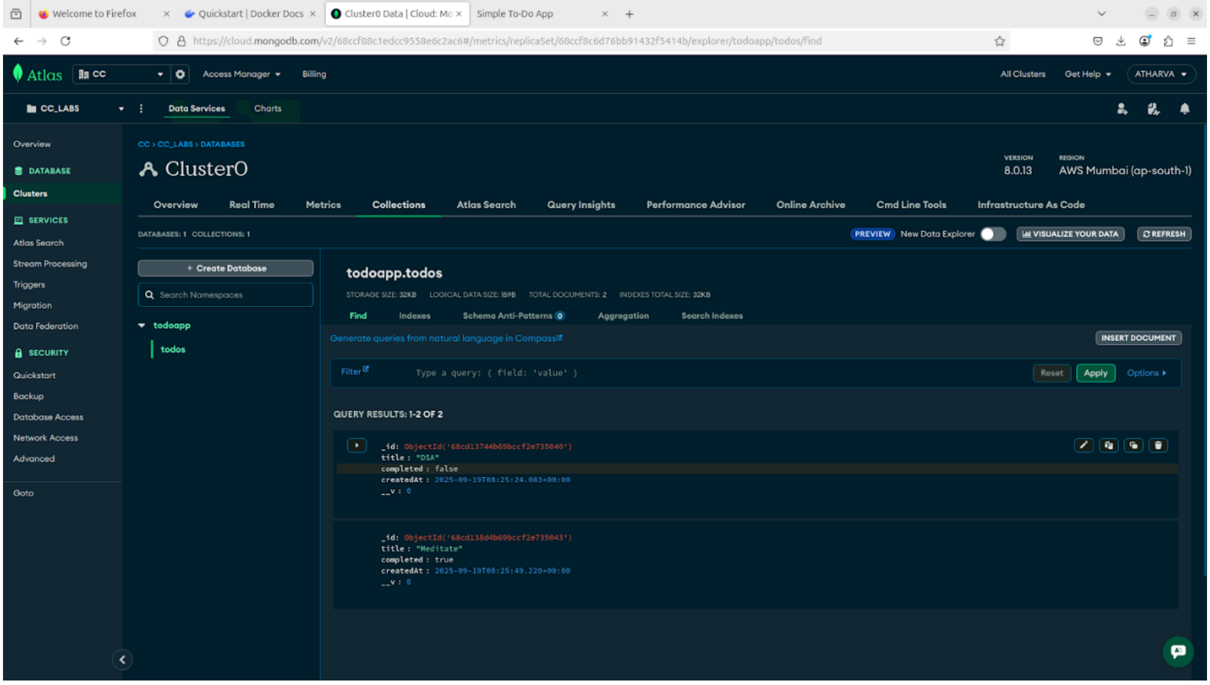


Simple To-Do App

What needs to be done? Add

~~Meditate~~
Created: Sep 19, 2025, 01:55 PM
Undo Delete

DSA
Created: Sep 19, 2025, 01:55 PM
Complete Delete



todoapp.todos

STORAGE SIZE: 32KB LOGICAL DATA SIZE: 159B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 32KB

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('68cd13744b69bccf2e735040')
title : "DSA"
completed : false
createdAt : 2025-09-19T08:25:24.083+00:00
__v : 0

_id: ObjectId('68cd138d4b69bccf2e735043')
title : "Meditate"
completed : true
createdAt : 2025-09-19T08:25:49.220+00:00
__v : 0
```

Observation:

- Docker and Docker compose was successfully installed.
- Running sudo docker-compose build successfully built the Docker image.
- Running sudo docker-compose up started the containers and the application ran without errors.
- The application was accessible at the specified port (e.g., http://localhost:3001 as defined in docker-compose.yml).