

Towards A to Z guide on how to deploy a Node.js todo app with Docker Compose

March 31, 2023



Deploy a Node.js todo app with Docker Compose



Introduction

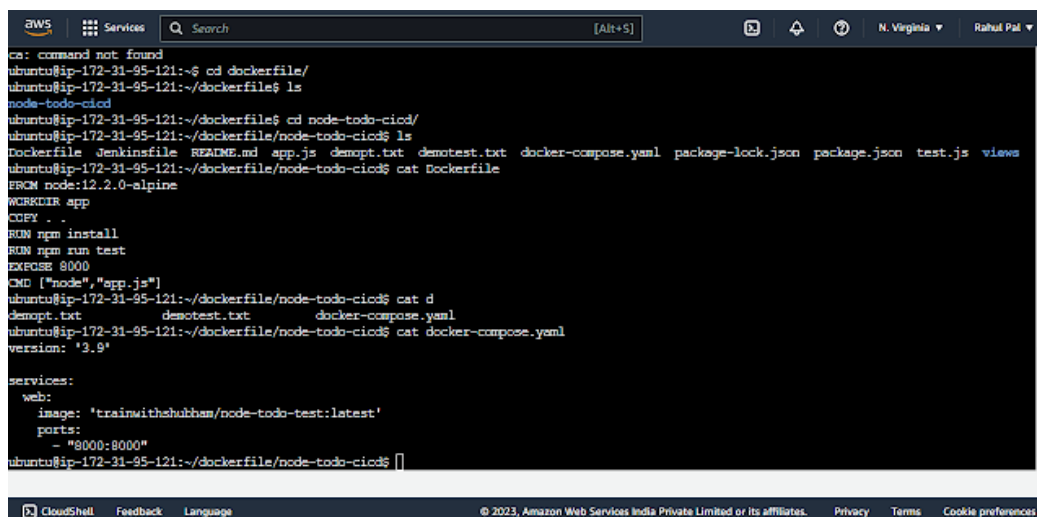
Deploying a Node.js app can be a complex process, especially when it comes to managing dependencies, environment variables, and infrastructure. Docker Compose is a tool that makes it easy to define and run multi-container Docker applications. It simplifies the process of deploying and managing complex applications by providing a simple way to define the environment and dependencies for each service in your app. In this tutorial, we'll walk through the process of deploying a Node.js todo app with Docker Compose. We'll start by creating a Dockerfile for our app, then we'll define the services that

Prerequisites

- Basic knowledge of Docker and Docker Compose
- Docker installed on your local machine
- A Node.js todo app (you can use any existing app or create a new one)

Step 1: Create a Dockerfile

The first step is to create a Dockerfile for your Node.js app. This file will define the environment for your app and how to build the container image. Here's an example Dockerfile:



```
aws
Services
Search
[Alt+S]
N. Virginia
Rahul Pat

ca: command not found
ubuntu@ip-172-31-95-121:~$ cd dockerfile/
ubuntu@ip-172-31-95-121:~/dockerfile$ ls
node-todo-cid
ubuntu@ip-172-31-95-121:~/dockerfile$ cd node-todo-cid/
ubuntu@ip-172-31-95-121:~/dockerfile/node-todo-cid$ ls
Dockerfile Jenkinsfile README.md app.js demotest.txt docker-compose.yml package-lock.json package.json test.js views
ubuntu@ip-172-31-95-121:~/dockerfile/node-todo-cid$ cat Dockerfile
FROM node:12.2.0-alpine
WORKDIR app
COPY . .
RUN npm install
RUN npm run test
EXPOSE 8000
CMD ["node", "app.js"]
ubuntu@ip-172-31-95-121:~/dockerfile/node-todo-cid$ cat d
demotest.txt docker-compose.yml
ubuntu@ip-172-31-95-121:~/dockerfile/node-todo-cid$ cat docker-compose.yml
version: '3.9'

services:
  web:
    image: 'trainwithshubham/node-todo-test:latest'
    ports:
      - "8000:8000"
ubuntu@ip-172-31-95-121:~/dockerfile/node-todo-cid$
```

This Dockerfile uses the official Node.js 14 Alpine image as its base, sets the working directory to /app, copies the package.json and package-lock.json files to the working directory, installs the dependencies, copies the rest of the app files, exposes port 8000, and starts the app.

Step 2: Create a docker-compose.yml file

The next step is to create a docker-compose.yml file to define the services that make up your app. Example docker-compose.yml file

Step 3: Build and run the app

To build and run the app with Docker Compose, navigate to the directory that contains the docker-compose.yml file and run the following command:

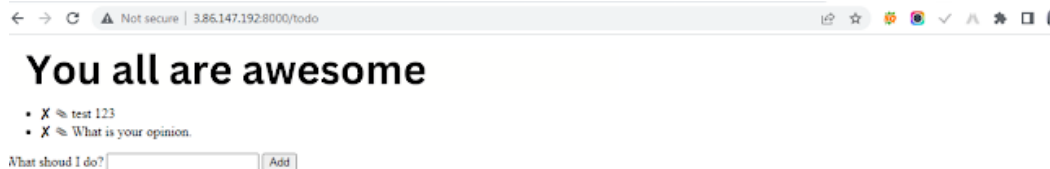
Copy code

```
docker-compose up
```

This command will build the images and start the containers for the web and db services. You should see the output from your app in the console.

Step 4: Test the app

To test the app, open a web browser and navigate to <http://localhost:3000>. You should see your todo app running.



Conclusion

In this tutorial, you learned how to deploy a Node.js todo app with Docker Compose. You created a Dockerfile to define the environment

Brand2Cloud - Cloud DevOps & Branding Culture

and tested the app. This approach makes it easy to deploy your app in any environment, and ensures that your app is consistent across all environments.

DEVOPS



Enter comment