

Docker container-Creating Docker images

Here's a summary on creating Docker images and an example:

What is a Docker Image?

A Docker image is a template that contains the application code, libraries, dependencies, and configuration required to run a Docker container. It's essentially a snapshot of a filesystem at a particular point in time.

Creating a Docker Image

To create a Docker image, you need to create a `Dockerfile` which is a text file containing instructions on how to build the image. The `Dockerfile` specifies the base image, copies files into the image, sets environment variables, and more.

Here's an example `Dockerfile` for creating an image of a Node.js application:

```
# Use an official Node.js runtime as our base image
FROM node:14

# Set the working directory to /app
WORKDIR /app

# Copy the contents of the current directory into the container at /app
COPY . /app

# Install any dependencies specified in package.json
RUN npm install

# Make port 3000 available to the world outside this container
EXPOSE 3000

# Define environment variable
ENV NAME World

# Run command when image is created
CMD ["node", "server.js"]
```

Explanation of Dockerfile instructions:

1. `FROM node:14`: This line uses an official Node.js runtime as our base image.
2. `WORKDIR /app`: This sets the working directory in the container to `/app`.
3. `COPY . /app`: Copies the contents of the current directory into the container at `/app`.
4. `RUN npm install`: Installs any dependencies specified in `package.json` within the container.
5. `EXPOSE 3000`: Makes port 3000 available to the world outside this container.
6. `ENV NAME World`: Sets an environment variable named `NAME` with value `World`.
7. `CMD ["node", "server.js"]`: This line specifies what command to run when a new container is launched from this image.

Building and Running the Docker Image

Once you have created your `Dockerfile`, save it in a directory along with your project files (in this case, your Node.js application code). Then, navigate to that directory in your terminal and run:

```
docker build -t my-node-app .
```

This will create an image named `my-node-app`. To verify the creation of the image, use:

```
docker images
```

After verifying the existence of your image, you can run a container from it using:

```
docker run -p 3000:3000 my-node-app
```

This will start a new container from `my-node-app` and make port 3000 available on the host machine.