Docker Ubuntu Setup

This README explains how to set up and run an Ubuntu container using Docker. The container will set a root password for the system and provide basic Linux utilities.

Files

1. `.env`

This file contains environment variables that will be used in the container setup. Specifically, it defines the `ROOT_PASSWORD` for the root user in the container.

```
ROOT_PASSWORD=abcd1234
```

2. `Dockerfile`

The Dockerfile specifies the base image (Ubuntu) and installs some basic packages (curl, wget, vim, net-tools, etc.). It also includes the `entrypoint.sh` script that sets the root password using the value defined in the `.env` file.

```
FROM ubuntu:latest

# Install basic packages
RUN apt-get update && apt-get install -y \
    sudo \
    curl \
    wget \
    vim \
    net-tools \
    iputils-ping

# Set environment variables
ENV LANG C.UTF-8
ENV LC_ALL C.UTF-8

# Set working directory
WORKDIR /root

# Copy the entrypoint script and make it executable
COPY entrypoint.sh /entrypoint.sh
RUN chmod +x /entrypoint.sh

# Run the entrypoint script
ENTRYPOINT ["/entrypoint.sh"]

# Default command to run
CMD ["/bin/bash"]
```

3. `entrypoint.sh`

This script is executed when the container starts. It sets the root password using the environment variable `ROOT_PASSWORD`.

```
#!/bin/bash

# Check if ROOT_PASSWORD is set
if [ -z "$ROOT_PASSWORD" ]; then
    echo "Error: ROOT_PASSWORD is not set!"
    exit 1
fi

# Set root password
echo "root:$ROOT_PASSWORD" | chpasswd

# Continue with the original command
exec "$@"
```

4. `docker-compose.yml`

This file defines how to build and run the container using Docker Compose. It loads the environment variables from the `.env` file, builds the container from the Dockerfile, and exposes port `8080`.

```
version: '3.8'
services:
  linux:
    container_name: ubuntu-linux
    build:
      context: .
      dockerfile: Dockerfile
    restart: always
    ports:
      - "8080:80"
    networks:
      - network_local_server
    volumes:
      - linux_data:/var/lib/data
      - "/home/ubuntu:/root"
    environment:
      - ROOT_PASSWORD=abcd1234
    healthcheck:
      test: ["CMD-SHELL", "ping -c 1 google.com || exit 1"]
      interval: 30s
      timeout: 10s
      retries: 5
    labels:
      - com.corhuila.group=linux-environment
```

```
volumes:
  linux_data:
    driver: local

networks:
  network_local_server:
    external: true
```

How to Use

Step 1: Build and Run the Container

Make sure you have Docker and Docker Compose installed on your machine. Then, run the following command to build and start the container:

```
docker-compose up --build
```

Step 2: Access the Container

To access the container with the root password defined, run the following command:

```
docker run -it --env ROOT_PASSWORD=abcd1234 ubuntu-linux:latest /bin/bash
```

This will give you access to the bash shell of the running Ubuntu container.

Step 3: Test the Root Password

You can test if the root password has been set by trying to switch to the root user inside the container:

```
su root
```

When prompted for the password, enter the one specified in the `.env` file (e.g., `abcd1234`).

Troubleshooting

- Make sure the `.env` file is correctly formatted and accessible by Docker Compose.
- Ensure that Docker is installed and running on your machine.
- If the container is restarting, check the `docker-compose logs` to see if there are any errors.

Solving the Issue: "REMOTE HOST IDENTIFICATION HAS CHANGED"

If you receive the following message when attempting to connect via SSH:

```
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
...
Host key verification failed.
```

This error occurs because the host key has changed or does not match the one stored in `known_hosts`. Follow these steps to resolve it:

Steps to solve the issue:

1. Run the following command to remove the offending key:

   ```
   ssh-keygen -R [localhost]:2222
   ```

2. Try reconnecting using the following command:

   ```
   ssh root@localhost -p 2222
   ```

Warning about using Docker

If you use the following Docker command:

```
docker run -it --env ROOT_PASSWORD=abcd1234 ubuntu-linux:latest /bin/bash
```

Any changes you make inside the container **will not be saved** once you stop it. This is a significant drawback if you need to persist changes.

Dockerfile network configuration

The network configuration should include the following block to define the driver and network name:

```
networks:
  network_local_server:
    driver: bridge
    name: network_local_server
```