Setting up Oracle 19c in Docker can be done using Oracle's official Docker build scripts, but due to Oracle's licensing, **Oracle does not publish official Docker images on Docker Hub**. Instead, you must **manually download the Oracle 19c installation files** and **build the image locally**.

Here's a full guide with a `Dockerfile` to set up Oracle Database 19c **Enterprise Edition** using Oracle's official scripts.

---

# ✅ Prerequisites

1. **Download the Oracle 19c installation file** manually from:
   👉 https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html
   - File needed (for Linux x86_64):
     `LINUX.X64_193000_db_home.zip`
2. **Install Docker** on your system.

---

## 🔒 Why This Is Required

Oracle does **not allow direct Docker pulls** of its database images due to licensing restrictions. This process ensures you **accept the license** before using the software.

# 🐳 Step-by-step Setup

- **Clone Oracle's Docker GitHub repo**

  ```
  git clone https://github.com/oracle/docker-images.git
  cd docker-images/OracleDatabase/SingleInstance/dockerfiles
  ```

- **Place the ZIP file in the 19.3.0 directory**

  ```
  mv ~/Downloads/LINUX.X64_193000_db_home.zip 19.3.0/
  ```

- **Build the image**

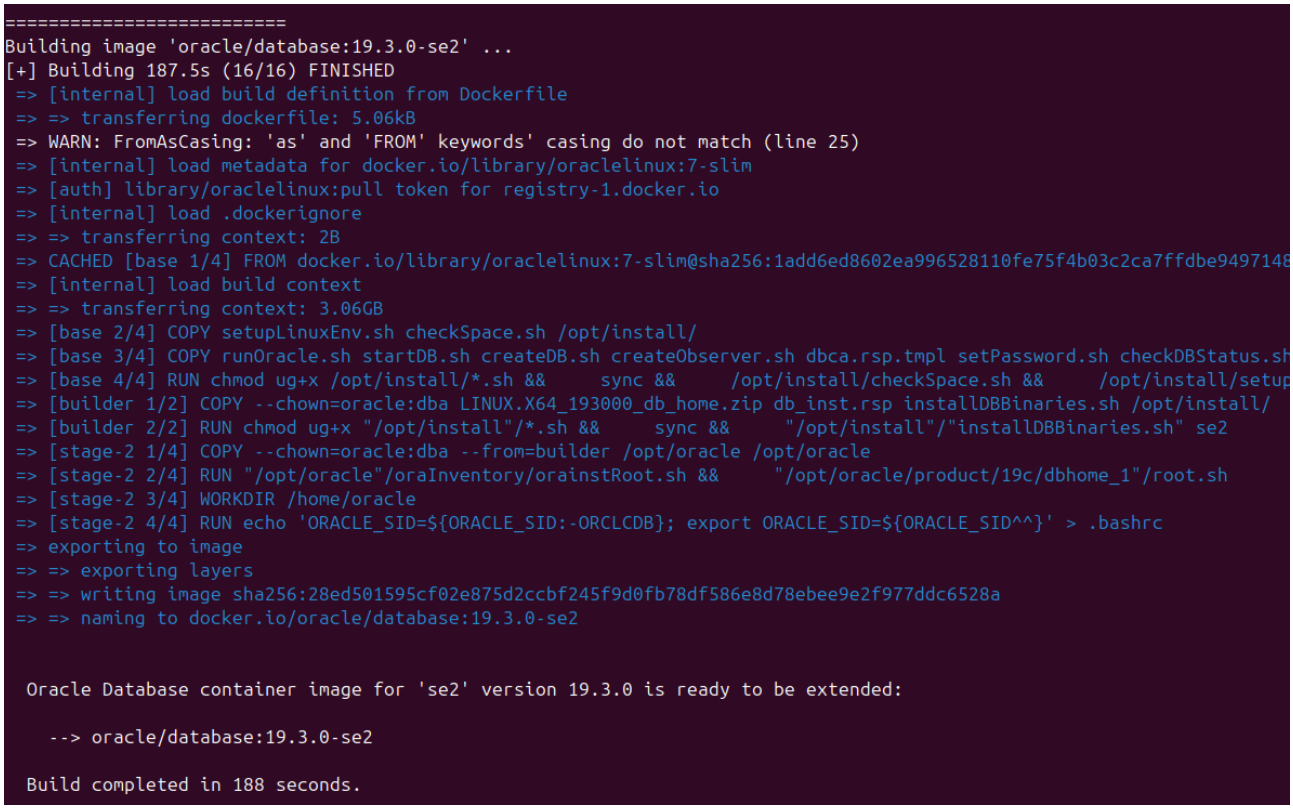
  `./buildContainerImage.sh -v 19.3.0 -s`

  - `-v`: version
  - `-s`: software-only (optional, for INSTALL_DB_SWONLY)

- **Use the image locally**

  Once built, Docker will tag it as:

  `oracle/database:19.3.0-se2`




You can now use that in your Dockerfiles:

**`FROM oracle/database:19.3.0-se2`**

---

# ✅ Use the created docker image

Great! You can fully integrate this into the `Dockerfile` **by using a** `.env` **file in combination with a** `docker-compose.yml` **file**—because a Dockerfile alone can't run `docker run` or define container runtime configuration like ports and environment variables.

Here's how to achieve your goal **step by step**:

---

# 🐳 Step 1: `.env` File

Create a file named `.env` in the same directory:

```
# .env
ORACLE_SID=ORCLCDB
ORACLE_PDB=tordb1
ORACLE_PWD=pwdtrainer
ORACLE_CHARACTERSET=AL32UTF8
```

---

## 🐳 Step 2: `Create init-user.sql`

```
-- init-user.sql

-- Creates user and grants privileges on first run

ALTER SESSION SET "_ORACLE_SCRIPT"=true;


CREATE USER zeus IDENTIFIED BY pwdtrainer;

GRANT CONNECT, RESOURCE, DBA TO zeus;
```

---

## 🐳 Step 3: Dockerfile (already correct)

Your Dockerfile should already include the following to copy your init SQL:

```
# Base image from Oracle
FROM oracle/database:19.3.0-se2

# Switch to root to fix permissions
USER root

# Ensure required directories exist with correct permissions
RUN mkdir -p /opt/oracle/product/19c/dbhome_1/network/log && \
    mkdir -p /opt/oracle/oradata && \
    mkdir -p /opt/oracle/cfgtoollogs && \
    chown -R oracle:oinstall /opt/oracle && \
    chmod -R 777 /opt/oracle

# Copy the startup SQL script to create user and grant privileges
COPY init-user.sql /opt/oracle/scripts/startup/

# Set correct permissions for startup script
RUN chown oracle:oinstall /opt/oracle/scripts/startup/init-user.sql && \
    chmod 775 /opt/oracle/scripts/startup/init-user.sql

# Expose ports for Oracle DB and EM Express
EXPOSE 1521 5500

# Define mountable volume
VOLUME ["/opt/oracle/oradata"]

# Switch back to oracle user
USER oracle

# Default command (will auto-run DBCA at first startup if DB not initialized)
CMD ["/bin/bash", "-c", "/opt/oracle/runOracle.sh"]
```

---

## 🐳 Step 4: `docker-compose.yml`

```
version: '3.8'

services:
  oracle-db:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: tordb1-container
    ports:
      - "1521:1521"
      - "5500:5500"
    env_file:
      - .env
    environment:
      ORACLE_SID: ${ORACLE_SID}
      ORACLE_PDB: ${ORACLE_PDB}
      ORACLE_PWD: ${ORACLE_PWD}
      ORACLE_CHARACTERSET: ${ORACLE_CHARACTERSET}
    volumes:
      - oracle-data:/opt/oracle/oradata

volumes:
  oracle-data:
```

---

## 🐳 Step 5: Run Everything

Now run the following in the same folder as the Dockerfile and `.env`:

```
# adjust permissions

sudo chown -R 54321:54321 ./oracle-data

sudo chmod -R 775 ./oracle-data
```

```
#rum the docker image

docker compose up --build -d
```

---

# ✅ Result

- Your database container (`tordb1-container`) will be started
- Environment variables will be taken from `.env`
- The database will include the PDB `tordb1`
- The user `zeus` with password `pwdtrainer` will be created