

<https://gist.githubusercontent.com/cesarwbr/53e56f4ab6c51a3f81eb796a3cccc206/raw/f6a852366d0fdea355f3767771fe15579d088173/Dockerfile>

FROM elixir:latest

```
RUN apt-get update && \
    apt-get install -y postgresql-client && \
    apt-get install -y inotify-tools && \
    apt-get install -y nodejs && \
    curl -L https://npmjs.org/install.sh | sh && \
    mix local.hex --force && \
    mix archive.install hex phx_new 1.5.3 --force && \
    mix local.rebar --force
```

ENV APP\_HOME /app

RUN mkdir \$APP\_HOME

WORKDIR \$APP\_HOME

CMD ["mix", "phx.server"]

version: "3"

services:

phoenix:

build: .

volumes:

- ./src:/app
- ports:
- "4000:4000"

depends\_on:

- db
- db:
  - image: postgres:9.6
  - environment:
    - POSTGRES\_USER: postgres
    - POSTGRES\_PASSWORD: postgres
    - PGDATA: /var/lib/postgresql/data/pgdata
  - restart: always
  - volumes:
    - ./pgdata:/var/lib/postgresql/data

docker-compose build

Step 3: create the "src" directory and the command "mix" alias.

The “src” directory will contain the Phoenix application.

```
$ mkdir src
```

Create the “mix” alias to run the Elixir commands.

```
$ alias mix="docker-compose run --rm phoenix mix"
```

Step 4: Initialize and configure a new Phoenix application

The following command will create a new Phoenix application called “hello” under the “src/” directory, which is mounted inside the container under “/app” (the default work directory).

```
$ mix phx.new . --app hello
```

Change the database hostname configuration on file src/config/dev.exs to point to the db container.

```
...
```

```
# Configure your database
```

```
...
```

```
hostname: "db",
```

```
...
```

Now you will initialize the database with Ecto.

```
$ cd src
```

```
$ mix ecto.create
```

If you copied an existing application, now would be the time to run your database migrations.

```
$ mix ecto.migrate
```

Step 5: Start the application

Great! Now your application is ready to be run. Starting your application is easy, back to the project directory and start up your application.

```
$ cd ..
```

```
$ docker-compose up
```