

A Phoenix Framework Repo

This repo builds a Phoenix Framework docker image.

This image is based on the Elixir Alpine docker image.

The included `prepare` script will create a Phoenix Framework project without database support.

Naming Convention

The naming convention is branched into **Standard** and **Extended** and is based on similar projects based on the Alpine Linux distribution, where `-alpine` is appended to the end of the tag.

The **Standard** branch is based on the latest stable version of Elixir provided by the Elixir docker image.

The **Extended** branch may either be based on the previous stable version of Elixir or the cutting edge version of Elixir.

Standard Naming Convention

```
aviumlabs/phoenix:<version | latest>-alpine
```

where version is either numeric based on the Phoenix version or the literal 'latest'

Extended Naming Convention

```
aviumlabs/phoenix:<version | latest>-elixir<version>-alpine
```

Build

Latest

The image defaults to building the latest version of Phoenix Framework.

```
$ docker build --no-cache -t aviumlabs/phoenix:latest-alpine .
```

Specific Version

To build a specific version of the Phoenix Framework; pass in the Phoenix version you want to build:

```
$ export PHX_VERSION=1.7.6
$ docker build --no-cache -t aviumlabs/phoenix:$PHX_VERSION-alpine \
--build-arg PHX_VERSION=$PHX_VERSION .
```

Template Repo

This repo is a template repo.

GitHub documentation for using a template repository is here:

<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-repository-from-a-template>

Companion Project

There is a companion project to this project for building a Phoenix Framework project integrated with PostgreSQL.

<https://github.com/aviumlabs/phoenix-compose.git>

The aviumlabs/phoenix-compose repo is a template repository.

The services included are:

- PostgreSQL 15.4
- Phoenix Framework 1.7.7 or later

Project Notes

Testing

Testing a new build:

```
$ cd <image/directory>
$ ./prepare -i testapp
```

```
    Initializing Phoenix Framework project...
```

```
    Application container root... /opt
```

```
    Application name..... testapp
```

```
    Running phx.new --install --no-ecto...
```

```
    ... * running mix deps.compile
```

```
    ...
```

```
    We are almost there! The following steps are missing:
```

```
    $ cd testapp
```

```
    Start your Phoenix app with:
```

```
    $ mix phx.server
```

```
    You can also run your app inside IEx (Interactive Elixir) as:
```

```
    $ iex -S mix phx.server
```

```
    The above 3 steps are completed by the prepare script with -f flag:
```

```
$ ./prepare -f
```

```

Compiling 14 files (.ex)
Generated testapp app
[info] Running TestappWeb.Endpoint with cowboy 2.10.0 at
0.0.0.0:4000 (http)
[info] Access TestappWeb.Endpoint at http://localhost:4000
[watch] build finished, watching for changes...
Rebuilding...
Done in 777ms.

```

Prepare -f finalizes the configuration and brings the docker container up in the foreground.

In a separate terminal session, confirm the application is running:

```

$ curl -X 'GET' http://localhost:4000
<!DOCTYPE html>    <html lang="en" class="[scrollbar-gutter:stable]">
<head>            <meta charset="utf-8">            <meta name="viewport"
content="width=device-width, initial-scale=1">            <meta name="csrf-token"
content="GkYEKgwzKR03KGoTB1E_D0h_FRw2FlknEs0bTceKuK2pH70LpGVonAhD">
<title data-suffix=" · Phoenix Framework">    Testapp    ·
Phoenix Framework</title>            <link phx-track-static rel="stylesheet"
href="/assets/app.css">            <script defer phx-track-static
type="text/javascript" src="/assets/app.js">            </script>
</head>            <body class="bg-white antialiased">            ...            </body>
</html>

```

Output from testapp runtime terminal:

```

[info] GET /
[debug] Processing with TestappWeb.PageController.home/2
Parameters: %{}
Pipelines: [:browser]
[info] Sent 200 in 613µs

```

Press ctrl-c a to stop the running testapp

Docker Hub

Internal notes for pushing images to Docker Hub.

```

$ docker push aviumlabs/phoenix:<tagname>-alpine

$ docker push aviumlabs/phoenix:latest-alpine
$ docker push aviumlabs/phoenix:1.7.6-alpine
$ docker push aviumlabs/phoenix:latest-elixir1.14.5-alpine

```