# How to pass environment variable to docker-compose up

Asked 7 years, 5 months ago Modified 8 months ago Viewed 253k times

187

I am trying to run a container. I already have the image uploaded to private Docker registry. I want to write a compose file to download and deploy the image. But I want to pass the TAG name as a variable from the docker-compose run command.My compose file looks like below. How can I pass the value for KB_DB_TAG_VERSION as part of docker-compose up command?

```
version: '3'
services:
   db:
    #build: k-db
    user: "1000:50"
    volumes:
      - /data/mysql:/var/lib/mysql
    container_name: k-db
    environment:
      - MYSQL_ALLOW_EMPTY_PASSWORD=yes
    image:  XX:$KB_DB_TAG_VERSION
    image: k-db
    ports:
      - "3307:3306"
```

docker docker-compose dockerfile environment-variables

Share Improve this question Follow

edited Dec 13, 2024 at 1:44
Jay Prall
5,495 ●5 ●54 ●82

asked Mar 15, 2018 at 7:44
Abhi.G
2,291 ●5 ●24 ●40

## 8 Answers

Sorted by: Highest score (default)

227

✓

You have two options (option `2.` overrides `1.`):

1. Create the .env file as already suggested in another answer.
2. Prepend `KEY=VALUE` pair(s) to your `docker-compose` command, e.g:

   ```
   KB_DB_TAG_VERSION=kb-1.3.20-v1.0.0 docker-compose up
   ```

   Exporting it earlier in a script should also work, e.g.:

   ```
   export KB_DB_TAG_VERSION=kb-1.3.20-v1.0.0
   docker-compose up
   ```

Keep in mind that these options just pass an environment varible to the `docker-compose.yml` file, not to a container. For an environment variable to be actually passed to a container you always need something like this in your `docker-compose.yml`:

```
environment:
  - KB_DB_TAG_VERSION=$KB_DB_TAG_VERSION
```

Share Improve this answer Follow edited Aug 13, 2023 at 12:47 answered Jun 22, 2018 at 15:54

Jakub Kukul
14.9k ● 4 ● 60 ● 60

51 KB_DB_TAG_VERSION=kb-1.3.20-v1.0.0 docker-compose up, Only works if in docker-compose.yml in environment: KB_DB_TAG_VERSION=$KB_DB_TAG_VERSION – Fadi Bakoura Jul 31, 2019 at 6:22

130 How wierd that the `-e` flag is missing. It's there for `docker-compose run`. Well well. – Karl Pokus Sep 24, 2019 at 9:08

3 Prepending a variable seems to override the version given in the .env. (So prepending can be used to do try ad hoc changes without changing the .env) – vlz May 8, 2020 at 12:30

5 This doesn't work on Windows CMD or PowerShell – Slav Oct 6, 2022 at 15:49

1 @KarlPokus "How wierd that the -e flag is missing" - I guess that's because `up` applies to all containers and there could be arbitrary numbers of containers you're passing the same env var to. For an alternative where you can still use the command line, see my answer: stackoverflow.com/a/74069689/28190 – Dan Gravell Oct 14, 2022 at 13:06

71

You can create a `.env` file on the directory where you execute the `docker-compose up` command (and your `docker-compose.yml` file is located) with the following content:

```
KB_DB_TAG_VERSION=kb-1.3.20-v1.0.0
```

Your `docker-compose.yml` file should look like the following (added `{` and `}`):

```
version: '3'
services:
   db:
     user: "1000:50"
     volumes:
       - /data/mysql:/var/lib/mysql
     container_name: k-db
     environment:
       - MYSQL_ALLOW_EMPTY_PASSWORD=yes
     image: XX:${KB_DB_TAG_VERSION}
     image: k-db
     ports:
       - "3307:3306"
```

After making the above changes , check whether the changes are reflected or not using the command `docker-compose config`. The variable will be replaced by the variable value. Please refer to the page here to understand more about variable replacement.

Share Improve this answer Follow

edited Dec 3, 2019 at 11:22
Shubhanshu Rastogi
2,433 ●2 ●22 ●33

answered Mar 15, 2018 at 8:00
Sebastian Brosch
43.7k ●15 ●78 ●89

When I try this option I see that additional "+" is being added. XX:+kb-1.3.20-v1.0.0 – Abhi.G Mar 15, 2018 at 9:36

You don't want to set the additional `+`? Which value should be set? – Sebastian Brosch Mar 15, 2018 at 10:02

I want it to be XX:kb-1.3.20-v1.0.0 not XX:+kb-1.3.20-v1.0.0. I dont want the "+". – Abhi.G Mar 20, 2018 at 5:04

Quick hint, in my case I had the .env file named .env.dev and referenced inside the env_file attribute in docker-compose.yml and could not use ${variable} option inside docker-compose. Renaming my .env.dev file to just .env as stated here did the trick. – Alfredo Rodriguez Sep 5, 2022 at 17:35

This is really This is really weird, `${XXX}` doesn't work under my .env, `$XXX` does. Wait ...... It works now, if it doesn't, restart your shell ...... – Jay Nov 30, 2022 at 5:42

24

Just to supplement what has been outlined by others, in particular by @JakubKukul

For security purposes you probably wouldn't want to keep vulnerable information such as username/password in your docker-compose files if they're under version control. You can map environment variables that you have on your host to environment variables inside container as well. In this case it could be something like the following:

```
version: '3'
services:
   db:
    #build: k-db
    user: "1000:50"
    volumes:
      - /data/mysql:/var/lib/mysql
    container_name: k-db
    environment:
      - MYSQL_ALLOW_EMPTY_PASSWORD=yes
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
    image:  XX:$KB_DB_TAG_VERSION
    image: k-db
    ports:
      - "3307:3306"
```

where `MYSQL_PASSWORD` would be both:

1. An environment variable on your host (maybe just in the current shell session)
2. An environment variable inside the containers from the `db` service

Share Improve this answer Follow

answered May 18, 2021 at 11:48
vasigorc
992 ● 11 ● 27

14

It's possible to pass environment variables to containers on the command line *without* specifying the values in files. Add an `environment` key with the variable's name *only* (no value or assignment operator) on the container's service definition in the docker-compose file:

```
db:
    ...
    environment:
        - KB_DB_TAG_VERSION
```

Used in that way, with no assignment, means that docker-compose will look up the environment variable in the current environment/shell:

```
KB_DB_TAG_VERSION=mytagversion docker-compose up
```

Ref: https://docs.docker.com/compose/environment-variables/#pass-environment-variables-to-containers

Share Improve this answer Follow

answered Oct 14, 2022 at 13:04
Dan Gravell
8,345 ● 7 ● 46 ● 72

14

```
docker-compose  --env-file .\.env up
```

https://docs.docker.com/compose/environment-variables/

Share Improve this answer Follow

edited Jan 30, 2023 at 9:51
Adriaan
18.2k ● 7 ● 47 ● 88

answered Jan 30, 2023 at 9:50
riyaj9993
149 ● 1 ● 2

2

The doc that you can declare the env file in compose -

```
web:
  env_file:
    - web-variables.env
```

Share Improve this answer Follow

answered Nov 21, 2023 at 10:17

ashraf minhaj
1,197 ● 8 ● 26

---

1

For Windows, instead of export, use:

```
$env.KB_DB_TAG_VERSION = "1.3.20-v1.0.0"
docker-compose up
```

Share Improve this answer Follow

answered Dec 22, 2022 at 8:23

Matan Dobrushin
305 ● 3 ● 11

---

-4

In your docker-compose.yml file add

```
env_file:
  - .env_file
```

to your `db` service where .env_file is your .env file (change its name accordingly).

```
version: '3'
services:
   db:
    #build: k-db
    user: "1000:50"
    volumes:
      - /data/mysql:/var/lib/mysql
    container_name: k-db
    env_file:
      - .env_file
    environment:
      - MYSQL_ALLOW_EMPTY_PASSWORD=yes
    image:  XX:$KB_DB_TAG_VERSION
    image: k-db
    ports:
      - "3307:3306"
```

Share Improve this answer Follow

answered May 29, 2019 at 9:56

user1734227

23 THIS IS WRONG! Read the docs. env_file: defines *runtime* envars, not build-time! By default, docker-compose will parse a .env file (HAS to be named .env this is not parametrable apparently) into build-time environment. In that case however, the fact that you declared it under env_file: .env or not is completely irrelevant - docker-compose looks for it anyways. So it will SEEM like your answer works, but in fact

doesn't. Those looking for built-time envars should read the other answer. – logicOnAbstractions Mar 17, 2021 at 15:38

**Protected question**. To answer this question, you need to have at least 10 reputation on this site (not counting the association bonus). The reputation requirement helps protect this question from spam and non-answer activity.

### Start asking to get answers

Find the answer to your question by asking.

Ask question

### Explore related questions

**docker** **docker-compose** **dockerfile** **environment-variables**

See similar questions with these tags.