How to add users to Docker container?

Asked 6 years, 6 months ago   Active 1 year, 1 month ago   Viewed 413k times

▲

359

▼

⭐

108

🕑

I have a docker container with some processes (uwsgi and celery) running inside. I want to create a celery user and a uwsgi user for these processes as well as a worker group that they will both belong to, in order to assign permissions.

I tried adding `RUN adduser uwsgi` and `RUN adduser celery` to my Dockerfile, but this is causing problems, since these commands prompt for input (I've posted the responses from the build below).

What is the best way to add users to a Docker container so as to set permissions for workers running in the container?

My Docker image is built from the official Ubuntu14.04 base.

Here is the output from the Dockerfile when the adduser commands are run:

```
Adding user `uwsgi' ...
Adding new group `uwsgi' (1000) ...
Adding new user `uwsgi' (1000) with group `uwsgi' ...
Creating home directory `/home/uwsgi' ...
Copying files from `/etc/skel' ...
[91mEnter new UNIX password: Retype new UNIX password: [0m
[91mpasswd: Authentication token manipulation error
passwd: password unchanged
[0m
[91mUse of uninitialized value $answer in chop at /usr/sbin/adduser line 563.
[0m
[91mUse of uninitialized value $answer in pattern match (m//) at /usr/sbin/adduser
line 564.
[0m
Try again? [y/N]
Changing the user information for uwsgi
Enter the new value, or press ENTER for the default
    Full Name []:
Room Number []:    Work Phone []:  Home Phone []:  Other []:
[91mUse of uninitialized value $answer in chop at /usr/sbin/adduser line 589.
[0m
[91mUse of uninitialized value $answer in pattern match (m//) at /usr/sbin/adduser
line 590.
[0m
Is the information correct? [Y/n]
---> 258f2f2f13df
Removing intermediate container 59048863162a
```

```
passwd: password unchanged
[0m
```

linux   ubuntu   dockerfile

Share Improve this question Follow

edited May 26 '20 at 2:55
Manik
452   5   18

asked Dec 30 '14 at 8:26
rfj001
6,138   6   25   43

8 Answers

| Active | Oldest | Votes |

▲

611

▼

✔

🕐

The trick is to use `useradd` instead of its interactive wrapper `adduser`. I usually create users with:

```
RUN useradd -ms /bin/bash newuser
```

which creates a home directory for the user and ensures that bash is the default shell.

You can then add:

```
USER newuser
WORKDIR /home/newuser
```

to your dockerfile. Every command afterwards as well as interactive sessions will be executed as user `newuser`:

```
docker run -t -i image
newuser@131b7ad86360:~$
```

You might have to give `newuser` the permissions to execute the programs you intend to run before invoking the user command.

Using non-privileged users inside containers is a good idea for security reasons. It also has a few drawbacks. Most importantly, people deriving images from your image will have to switch back to root before they can execute commands with superuser privileges.

35   In order to set password you could use chpasswd like: `RUN echo 'newuser:newpassword' |`
      `chpasswd` – iuridiniz Jul 21 '16 at 17:16

5   Note that if you're creating a new user with a large user ID, docker may hang/crash as it tries to
      create *lastlog* - a massive sparse file. Avoid this with the `--no-log-init` option to `useradd` . –
      davidA Aug 24 '16 at 23:24 ✎

14   Nice tip, @iuridiniz! Don't forget to call it before `USER newuser` . If you also need the user to have
      root privileges, you can also include `adduser <username> sudo` . – Yamaneko Sep 1 '16 at 19:47

7   `/bin/sh: useradd: not found` alpine linux – deathangel908 May 27 '18 at 10:12

▲

Ubuntu

111   Try the following lines in `Dockerfile` :

▼
```
RUN useradd -rm -d /home/ubuntu -s /bin/bash -g root -G sudo -u 1001 ubuntu
USER ubuntu
WORKDIR /home/ubuntu
```
🕑

`useradd` options (see: `man useradd`):

- `-r` , `--system` Create a system account. see: Implications creating *system accounts*

- `-m` , `--create-home` Create the user's home directory.

- `-d` , `--home-dir HOME_DIR` Home directory of the new account.

- `-s` , `--shell SHELL` Login shell of the new account.

- `-g` , `--gid GROUP` Name or ID of the primary group.

- `-G` , `--groups GROUPS` List of supplementary groups.

- `-u` , `--uid UID` Specify user ID. see: Understanding how uid and gid work in Docker containers

- `-p` , `--password PASSWORD` Encrypted password of the new account (e.g. `ubuntu`).

Setting default user's password

To set the user password, add `-p "$(openssl passwd -1 ubuntu)"` to `useradd` command.

Alternatively add the following lines to your `Dockerfile` :

```
SHELL ["/bin/bash", "-o", "pipefail", "-c"]
```

4   Why would the user be in the root group? Isn't the whole point of this is to have a non-root user for security purposes – Novaterata Feb 7 '19 at 15:16

7   @Novaterata Depending on the use. `root` group doesn't indicate they've root access, it just they've more read access to some files (such as logs), which is useful, but it depends on the project. – kenorb Feb 7 '19 at 15:22

I can see that this works, I automatically am logged in as user, but I'm still generating files owned by root. I even just used 'USER user' since my username on the local user and group is 'user'. Still generates root owned files. Is there something else I should be doing? I'm basically making a docker container that compiles our codebase. So it checks out code from svn, sets up variables using bash source. Could bash commands be doing things as root even if I'm never asked for the root password? – JoeManiaci Jan 3 '20 at 22:30

Setting the password interactively with `SHELL ["/bin/bash", "-o", "pipefail", "-c"] RUN echo 'ubuntu:ubuntu' | chpasswd` didn't work for me, the container was just built, didn't ask me for the password. What did I do wrong? – Laryx Decidua Jan 28 at 16:50

▲

106

▼

⟲

To avoid the interactive questions by adduser, you can call it with these parameters:

```
RUN adduser --disabled-password --gecos '' newuser
```

The `--gecos` parameter is used to set the additional information. In this case it is just empty.

On systems with busybox (like Alpine), use

```
RUN adduser -D -g '' newuser
```

See busybox adduser

Share Improve this answer Follow   edited Feb 5 '18 at 15:51   answered Apr 22 '15 at 12:15

                                                                                           Raffael
                                                                                           2,137   1   14   13

---

3   Thanks! It looks like `adduser` high-level solution is generally preferred to using low-level functions like `useradd` . – akhmed Jun 20 '15 at 19:23

1   `adduser: unrecognized option: gecos` This doesn't seem to work on Alpine. – weberc2 Feb 4 '18 at 12:18

22

```
FROM node:10-alpine

# Copy source to container
RUN mkdir -p /usr/app/src

# Copy source code
COPY src /usr/app/src
COPY package.json /usr/app
COPY package-lock.json /usr/app

WORKDIR /usr/app

# Running npm install for production purpose will not run dev dependencies.
RUN npm install -only=production

# Create a user group 'xyzgroup'
RUN addgroup -S xyzgroup

# Create a user 'appuser' under 'xyzgroup'
RUN adduser -S -D -h /usr/app/src appuser xyzgroup

# Chown all the files to the app user.
RUN chown -R appuser:xyzgroup /usr/app

# Switch to 'appuser'
USER appuser

# Open the mapped port
EXPOSE 3000

# Start the process
CMD ["npm", "start"]
```

Above steps is a full example of the copying NodeJS project files, creating a user group and user, assigning permissions to the user for the project folder, switching to the newly created user and running the app under that user.

Share Improve this answer Follow

answered Jan 30 '19 at 17:18

S.Mishra
2,353   20   18

---

2   addgroup failed for me, Step 11/15 : RUN addgroup -S username ---> Running in db9fd22d469d
     Option s is ambiguous (shell, system) adduser [--home DIR] [--shell SHELL] [--no-create-home] [--
     uid ID] [--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID] [--disabled-
     password] [--disabled-login] [--encrypt-home] USER Add a normal user – teoring Mar 19 '20 at 13:35

---

superset: [superset-github](#)

```
RUN useradd --user-group --create-home --no-log-init --shell /bin/bash
    superset
```

I think it's a good way to follow open source.

Share Improve this answer Follow

answered Aug 2 '19 at 6:43

[Ryan Miao](#)
251 3 12

Everyone has their personal favorite, and this is mine:

7

```
RUN useradd --user-group --system --create-home --no-log-init app
USER app
```

Reference: **[man useradd](#)**

The `RUN` line will add the user and group `app`:

```
root@ef3e54b60048:/# id app
uid=999(app) gid=999(app) groups=999(app)
```

Use a more specific name than `app` if the image is to be reused as a base image. As an aside, include `--shell /bin/bash` if you really need.

Partial credit: [answer by Ryan M](#)

Share Improve this answer Follow

edited Dec 17 '19 at 20:41

answered Aug 19 '19 at 21:18

[Acumenus](#)
42.2k 14 116 110

Alternatively you can do like this.

2

```
RUN addgroup demo && adduser -DH -G demo demo
```

```
-H Don't create home directory
```

Share Improve this answer Follow   edited Feb 7 '20 at 9:10   answered Feb 6 '20 at 19:25

Lukasz Dynowski

6,093   2   48   77

P.S This command might slightly vary base on a Linux distribution. – Lukasz Dynowski May 11 at 8:34

Add this line to your Dockerfile (You can run any linux command this way)

0

```
RUN useradd -ms /bin/bash yourNewUserName
```

Share Improve this answer Follow   edited Mar 5 '18 at 14:28   answered Nov 23 '15 at 15:42

user6123723   basickarl

7,856   15   59   97   **26.6k**   44   175   273