# Can't add user to docker group

Asked 5 years, 3 months ago    Modified 1 month ago    Viewed 90k times

▲

**38**

▼

🔖

🕐

I'm trying to set docker up on a new system, and when running docker info I get:

```
docker -v
=> Docker version 18.09.5, build e8ff056

docker info
=> Got permission denied while trying to connect to the Docker daemon
   socket at unix:///var/run/docker.sock: Get
   http://%2Fvar%2Frun%2Fdocker.sock/v1.39/info: dial unix
   /var/run/docker.sock: connect: permission denied
```

Following the docs, I've tried:

```
sudo usermod -a -G docker $USER
```

Which returns no output. When I then run groups:

```
groups
=> mark adm cdrom sudo dip plugdev lpadmin sambashare
```

I can see a docker group exists:

```
less /etc/group | grep docker
=> docker:x:131:mark
```

And can see that it owns a socket running where the error message states:

```
ls -la /var/run/ | grep docker
=>
drwx------  5 root            root            120 May 25 14:54 docker
-rw-r--r--  1 root            root              5 May 25 14:54 docker.pid
srw-rw----  1 root            docker            0 May 25 14:54 docker.sock
```

So why can't I add myself to that group with `sudo usermod -a -G docker $USER` ?

**linux**  **ubuntu**  **usergroups**

Share  Improve this question  Follow

edited Jun 20 at 23:18

**Benjamin Loison**
**5,536** ● 4 ● 18 ● 37

asked May 25, 2019 at 14:13

**Mark**
Mark' **6,336** ● 5 ● 23 ● 50

11    I believe you need to login/restart for the changes to be completed. – Omari Celestine May 25, 2019 at 14:18

1    Your belief is correct, thanks! – Mark May 25, 2019 at 14:34

On Ubuntu 20.04, in my case, "login/restart" literally meant I needed to _reboot_ for the changes to take effect, not merely restart the bash session. Wowzers. – 4Z4T4R Feb 20, 2021 at 23:58

One gotcha to watch for is that you run `groups` to check if you've been added to the docker group whereas you probably need to run `groups $USER`. This is since according to `man groups` "if no USERNAME is specified, [groups will display groups] for the current process (which may differ if the groups database has changed)" – Anil Jan 17, 2023 at 14:04

## 6 Answers

Sorted by:   Highest score (default)   ⇕

You need to reload your shell in order to make the changes take effect. Often you need to reboot your shell process and possibly even restart your computer.

**52**

e.g

```
sudo usermod -aG docker $USER
sudo reboot
```

See @4Z4T4R answer and give a thumbs https://stackoverflow.com/a/66297855/7961500

### Load changes without quitting your shell

To avoid starting a new shell you can run. (Doesn't seem to work for all environments)

```
exec su -l $USER
```

This will create a new subshell with the loaded changes and replace your current shell with it.

### If you need it to work now without restarting your computer

Another way if you just need to get it working now, is to change your primary group. This is only a temp solution as with any new shell you will need to apply it again.

```
export my_group=$(id -gn)
newgrp docker
newgrp $my_group
```

### Documentation

You can also look at the offical documentation here https://docs.docker.com/engine/install/linux-postinstall/

Share   Improve this answer   Follow

edited Jun 20 at 23:18

Benjamin Loison
**5,536** ● 4 ● 18 ● 37

answered Sep 10, 2019 at 9:23

Louis Stekhoven-Smith
**1,119** ● 14 ● 10

---

3   restarting the shell did not work for me, but `newgrp docker` did. Any idea why that would happen?
– reynoldsnlp Aug 11, 2020 at 21:38

2   Same issue here on ubuntu 20.04, `newgrp docker` seemed to work for me. `sudo usermod -a -G docker $USER` seemed to work periodically, but after reboot the group was lost. – monkut Aug 12, 2020 at 8:53 ✎

1   Exactly the same problem on Ubuntu 19.10, for me only `newgrp docker` fixes it, but it's not persistent – Chuckatron Sep 9, 2020 at 20:51

3   The `newgrp` command is *very* different from `usermod -aG GROUP USER` : the latter *adds* group GROUP to USER, without changing the primary group of USER; the newgrp creates a new *shell* and in that shell, the USER's *primary* group changes to GROUP! This is not likely the desired effect: the OP still wants files created by USER to belong to user's primary GROUP, but wants files owned by GROUP to be accessible by USER. Very completely different. – Oliver Feb 6, 2021 at 14:48

That is true, I updated the post to help maintain the same behaviour. It is not the ideal way to solve this issue but is a final option if nothing else is working for you. At this point even the offical docker documentation suggests doing this. I added the link to the post. – Louis Stekhoven-Smith Feb 8, 2021 at 3:32

---

In my case, on Ubuntu 20.04, run `sudo reboot` after this command:

```
$ sudo usermod -aG docker $USER
```

I literally needed to *reboot* my operating system (and machine) for the change to take effect. Restarting/reloading the bash session did not apply the new setting.

Sure, `newgrp docker` does the trick "on the fly" without restart/reboot/re-anything... but once the session terminates, *POOF* you're not in the `docker` group any longer.

Added this as a formal answer bc it genuinely solved the OP's---and my (identical)---problem.

Credit should go to @Omari Celestine for the suggestion, but because I suck at interpretation, I (and maybe you) need the literal disambiguation that this answer provides.

Share   Improve this answer   Follow

answered Feb 21, 2021 at 0:05

4Z4T4R
**2,418** ● 2 ● 27 ● 45

I tried the usermod & reboot & doesn't work for me, the only thing working is `newgrp docker` – kisai May 12, 2022 at 21:08

---

▲

**3**

▼

🔖

🕓

Its a two step process technically. Run

```
sudo usermod -aG docker $USER
```

then,

```
sg docker -c "bash"
```

Share   Improve this answer   Follow

edited Jun 20 at 23:17

Benjamin Loison
**5,536** ● 4 ● 18 ● 37

answered Jun 10, 2022 at 8:28

Saurabh Kumar
**2,589** ● 18 ● 19

---

▲

**0**

▼

🔖

🕓

Change the permissions on the /var/run/docker.sock file and restart docker process.

```
sudo chown jenkins:jenkins /var/run/docker.sock
sudo 644 /var/run/docker.sock
```

Then,

```
sudo service docker restart
```

Share   Improve this answer   Follow

answered Jan 4, 2023 at 4:09

Sri
**9** ● 1

Your answer could be improved with additional supporting information. Please edit to add further details, such as citations or documentation, so that others can confirm that your answer is correct. You can find more information on how to write good answers in the help center. – Kunal Varpe Jan 4, 2023 at 7:41

---

▲

**0**

▼

🔖

Try this

```
sudo groupadd docker
```

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

Restart is required for the changes to take effect.

Share    Improve this answer    Follow

answered Jul 16 at 11:14

subhashis
**4,772**  ● 8   ● 39   ● 55

---

Before running $docker info, Please make sure that the docker service up.

If not pls start the service by running below command.

**-4**

$service docker start

Now you check the $docker info

Share    Improve this answer    Follow

answered Jul 7, 2021 at 16:09

Arrow
1

---