Dhananjay Kr.

15 Followers About Follow

Solving Docker permission denied while trying to connect to the Docker daemon socket

 Dhananjay Kr. Nov 9, 2019 · 2 min read

Problem:

You are trying to run a docker container or do the docker tutorial, but you only get an error message like this:

```
docker: Got permission denied while trying to connect to the Docker
daemon socket at unix:///var/run/docker.sock: Post
http://%2Fvar%2Frun%2Fdocker.sock/v1.26/containers/create: dial unix
/var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
```

Solution:

The error message tells you that your current user can't access the docker engine, because you're lacking permissions to access the unix socket to communicate with the engine.

As a temporary solution, you can use `sudo` to run the failed command as root (e.g. `sudo docker ps`).

However it is recommended to fix the issue by *adding the current user to the* `docker` *group*:

Run this command in your favourite shell and then **completely log out of your account and log back in** (or exit your SSH session and reconnect, if in doubt, reboot

the computer you are trying to run docker on!):

- sudo usermod -a -G docker $USER

After doing that, you should be able to run the command without any issues. Run `docker run hello-world` as a normal user in order to check if it works. Reboot if the issue still persists.

See What does sudo usermod -a -G group $USER do on Linux? for details on what this command changes on your system and what the parameters mean.

Logging out and logging back in is required because the group change will not have an effect unless your session is closed.

Background information:

On Linux, when you run any `docker` command, the `docker` binary will try to connect to `/var/run/docker.sock`. As indicated by its `.sock` extension, this file is a Unix Domain Socket – basically, a way so multiple processes can communicate on the local computer (also called an IPC mechanism – IPC = "Inter-Process Communication").

In the case of Docker, the main reason for using the socket is that any user belonging to the `docker` group can connect to the socket while the Docker daemon itself can run as `root`. Essentially, it's a convenience feature and allows multiple `docker` client commands to communicate to the same daemon process internally.

Docker Kubernetes DevOps Technology Linux

Get the Medium app