





Log in

Sign up

257 it is created on container startup.



You can run this script to verify:





```
#!/bin/bash
if [ -f /.dockerenv ]; then
    echo "I'm inside matrix ;(";
else
    echo "I'm living in real world!";
fi
```

MORE: Ubuntu actually has a bash script: /bin/running-in-container and it can return the type of container it has been invoked in. Might be helpful. Don't know about other major distros though.

Share

Improve this answer

Follow

edited May 15, 2022 at 6:09

Archimedes Trajano

40.5k • 24 • 201 • 327

answered Aug 27, 2014 at 3:09



4,846 • 2 • 15 • 10

- 4 On Debian /bin/running-in-container is provided by upstart. With the transition to systemd it might go away. I hope not it sounds useful! Max Murphy Sep 1, 2016 at 9:12
- 9 Others have pointed out that checking .dockerenv is not recommended Dave Sep 25, 2018 at 10:53
- 7 Note: testing for .dockerenv works only if the runtime is docker daemon. If you are using podman or something else this fails. Ben Kircher Sep 7, 2019 at 10:13
- 2 It's Aug 2020 and /.dockerenv is available on ubuntu:20.04 and alpine:3.12, so I'd say that's definitely the best answer Meir Gabay Aug 6, 2020 at 11:12
- 3 Ubuntu 18.0.4 doesn't have /bin/running-in-container . Daniel Griscom Oct 19, 2020 at 12:58

Show 3 more comments



The most reliable way is to check <code>/proc/1/cgroup</code> . It will tell you the control groups of the init process, and when you are *not* in a container, that will be <code>/</code> for all hierarchies. When you are <code>inside</code> a container, you will see the name of the

210





Follow

Log in

Sign up



Share

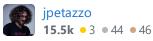
Improve this answer

tshepang

12.4k • 24 • 95 • 139

edited Jul 2, 2019 at 2:44

answered Nov 15, 2013 at 23:36





- 14 docker now uses docker instead of lxc in those paths Andy Nov 21, 2015 at 16:29
- Does not work for lxd/lxc containers, but stackoverflow.com/a/20010626/170230 does.

 Draco Ater Jun 8, 2016 at 6:39
- With later versions of systemd it looks like you can't rely on process 1 using / for all cgroups; on my Debian 9 system (systemd 232) only three of the ten cgroups (3:cpuset, 4:perf_event and 7:freezer) are at root; the rest are under /init.scope. That said, I think that searching that file for :/docker/ is probably the most reliable heuristic at the moment. cjs Feb 28, 2018 at 7:40 /
- 4 Not working for me. Host Ubuntu 19.04, guest Ubuntu 18.04 using LXC privileged container. /proc/1/cgroup does NOT contain the lxc string. Gab Sep 1, 2019 at 11:51
- 11 On my docker image: cat /proc/1/cgroup 0::/ so doesn't work. Brett Sutton Nov 18, 2021 at 10:54

Show 8 more comments



On a new ubuntu 16.04 system, new systemd & lxc 2.0

30

sudo grep -qa container=lxc /proc/1/environ



Share Improve this answer Follow

answered Mar 2, 2016 at 15:12



larss **301** • 3 • 2

43

2 This works for me on Ubuntu focal 20.04. None of the answers above this point did. – Jonathan Hartley Apr 23, 2020 at 15:18

Thanks! it works for lxc! Can you please explain why '-a' is needed? lsn't grep -q
container=lxc /proc/1/environ enough? - Alek Apr 28, 2021 at 16:55 /

/proc/\$\$/environ separates environment variables with null bytes. Without -a, this passage from the man page applies: > By default, TYPE is binary, and grep suppresses output after null input binary data is discovered – Adam Azarchs Sep 22, 2021 at 20:56





Log in

Sign up

Add a comment



A concise way to check for docker/lxc in a bash script is:

21

```
#!/bin/bash
if grep -sq 'docker\|lxc' /proc/1/cgroup; then
   echo "I am running on Docker."
fi
```





Share

Improve this answer

Follow

edited Jun 15, 2022 at 17:14



answered Jan 10, 2017 at 1:40



Thanks @DanielGriscom that looks way better. – oNaiPs Mar 29, 2021 at 20:53

This didn't work when my container was running within kubernetes. - erik Jun 22, 2021 at 15:02

FYI the echo command should use double quotes due to the single quote in "I'm" - TommyD Feb 14, 2022 at 14:56

Add a comment



Handy Python function to check if running in Docker:

```
def in docker():
    """ Returns: True if running in a Docker container, else False
    with open('/proc/1/cgroup', 'rt') as ifh:
        return 'docker' in ifh.read()
```





Share

Improve this answer

Follow

```
edited Jan 31 at 21:22
       Josh Correia
```

answered Mar 8, 2017 at 15:09



9.983 • 8 • 50 • 55

Important Note! This does not appear to work when the container is running in kubernetes. Instead, replace the last line with 'kubepod' in place of 'docker'. (Or, put in an "or" statement that checks for both ;)) – JJC Jan 13, 2019 at 3:20

4.279 • 3 • 40 • 61

It's kubepods I guess. – rookie099 Aug 29, 2019 at 7:53





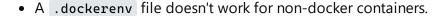
Log in

Sign up



As of 2022, with lxd v4.0+, none of the answers so far work for both docker and lxc.

14





• Checking that all hierarchies in /proc/1/cgroup are / kinda maybe works. However, some hierarchies on non-containers are /init.scope (Ubuntu 20.04 cgroup 0 and 1). So also not entirely reliable.



• Checking for container=lxc in /proc/1/environ works for lxc but not docker. Also, it requires root rights.



2.

The only way I've found so far that works reliably on both CentOS and Ubuntu with Ixc (4.0) containers and Docker, and *also* doesn't require root rights, is to check PID

On all host systems, PID 2 is kthreadd:

In containers, this PID either doesn't exist, or isn't kthreadd. Both docker and lxc show:

```
root@85396f8bce58:/# ps -p 2
PID TTY TIME CMD
root@85396f8bce58:/#
```

The best way seems to be to check /proc/2/status:

```
$ head -n1 /proc/2/status
Name: kthreadd
```

So something like this seems to work:

```
if [ -n "$(grep 'kthreadd' /proc/2/status 2>/dev/null)" ]; then
   echo "Not in container"
else
```





Log in

Sign up

Share Improve this answer Follow

answered May 6, 2022 at 6:08



4 Running docker on macOS; have no pid 2 normally. – Daniel Widdis Jul 6, 2022 at 21:26

With lxd 5.8, no process seen with pid 2 – Prem Anand Dec 6, 2022 at 11:09

With Windows 10, WSL 2 distribution Ubuntu 22.04.1, it does not have process of pid 2. But it certainly is not docker environment. – zerox May 8, 2023 at 4:43

You can simplify the if statement to if ! grep -q kthreadd /proc/2/status 2>/dev/null; then; echo "In container"; fi - Alex Henrie Feb 5 at 22:30

Add a comment



We use the proc's sched (/proc/\$PID/sched) to extract the PID of the process. The process's PID inside the container will differ then it's PID on the host (a non-container system).



12

For example, the output of /proc/1/sched on a container will return:



root@33044d65037c:~# cat /proc/1/sched | head -n 1
bash (5276, #threads: 1)

While on a non-container host:

```
$ cat /proc/1/sched | head -n 1
init (1, #threads: 1)
```

This helps to differentiate if you are in a container or not.

Share Improve this answer Follow

answered May 4, 2016 at 0:00



Depending on the OS, "init" might need to be replaced by "systemd". More information on systemd here. – BrianV Jun 23, 2017 at 1:50 here.



Q

Log in

Sign up

This seems to only work on Docker. In an LXC container It's returning Systemd PID 1
 MillerGeek May 2, 2018 at 14:32

- It's now returning 1 in docker as well. It is usually **sh** and not **init** there, but it may be almost anything in either. Jan Hudec May 6, 2020 at 14:58
- 5 Under docker, this is no longer the case bash-5.0# cat /proc/1/sched bash (1, #threads: 1) shalomb Jun 4, 2020 at 10:57

Add a comment



The easiest way would be to check the environment. If you have the container=lxc variable, you are within a container.





Otherwise, if you are root, you can try to perform mknod or mount operation, if it fails, you are most likely in a container with dropped capabilities.



Share Improve this answer Follow

answered Nov 15, 2013 at 21:09



creack

121k • 12 • 100 • 73

4

This one works not only for docker (I didn't check that), but more importantly for lxd/lxc containers (checked), where /proc/1/cgroup does not allow you to detect that.

– Draco Ater Jun 8, 2016 at 6:37

- 4 I mean...it is weird, usually env variables are in all caps, so looking for some precision here Alexander Mills Dec 20, 2016 at 10:22
- 14 docker run alpine env does not give anything that looks like that variable Archimedes Trajano Mar 10, 2019 at 16:01

Add a comment



This is an old question, but a REALLY good one. :)



I've written some automation scripts that we run on baremetal, VM and in a docker container, with logic branching based on which platform the script is executing on. In my case I have the privilege of creating both the container and the docker image, so this solution will only work if you are in control of the entire stack:





Q

Log in

Sign up

```
FROM ubuntu:18.04

ENV PLATFORM="docker"

RUN apt update; \
...
```

The script can then just check the value of \$PLATFORM for desired outcomes on each platform:

```
#!/bin/bash
# Check for executor specification in environment
case $PLATFORM in
 docker)
    # If running in Docker, do this stuff
   echo "Running containerized, proceeding..."
   ;;
 virtual)
    # If running in a VM, do different stuff
   echo "Running on a VM, loading VM stuff..."
   modprobe some-kernel-module
   i i
   echo "Unknown executor specified! Exiting..."
   exit 1
   i i
esac
```

I've omitted baremetal in the above code to keep it concise.

Share Improve this answer Follow

answered Nov 26, 2020 at 5:17



mainmachine **307** • 5 • 10

Add a comment

import os



Check for all the solutions above in Python:



```
def in_container():
    proc_1 = r'/proc/1/sched'

if os.path.exists(proc_1):
```





Log in

Sign up

```
checks = [
    'docker' in out,
    '/lxc/' in out,
    out.split(' ')[0] not in ('systemd', 'init',),
    os.path.exists('./dockerenv'),
    os.path.exists('/.dockerenv'),
    os.path.exists('/.dockerinit'),
    os.getenv('container') is not None
]
return any(checks)

if __name__ == '__main__':
    print(in_container())
```

Proof of concept:

```
$ docker run --rm -it --mount type=bind,source=${PWD}/incontainer.p
True
```

Share

edited Jun 1, 2023 at 22:39

answered Sep 26, 2017 at 22:54

Improve this answer

Follow



This didn't work for me on Mac based docker container. Returns empty. Docker version 2.1.0.1 (37199). – splintercell Nov 26, 2019 at 18:01

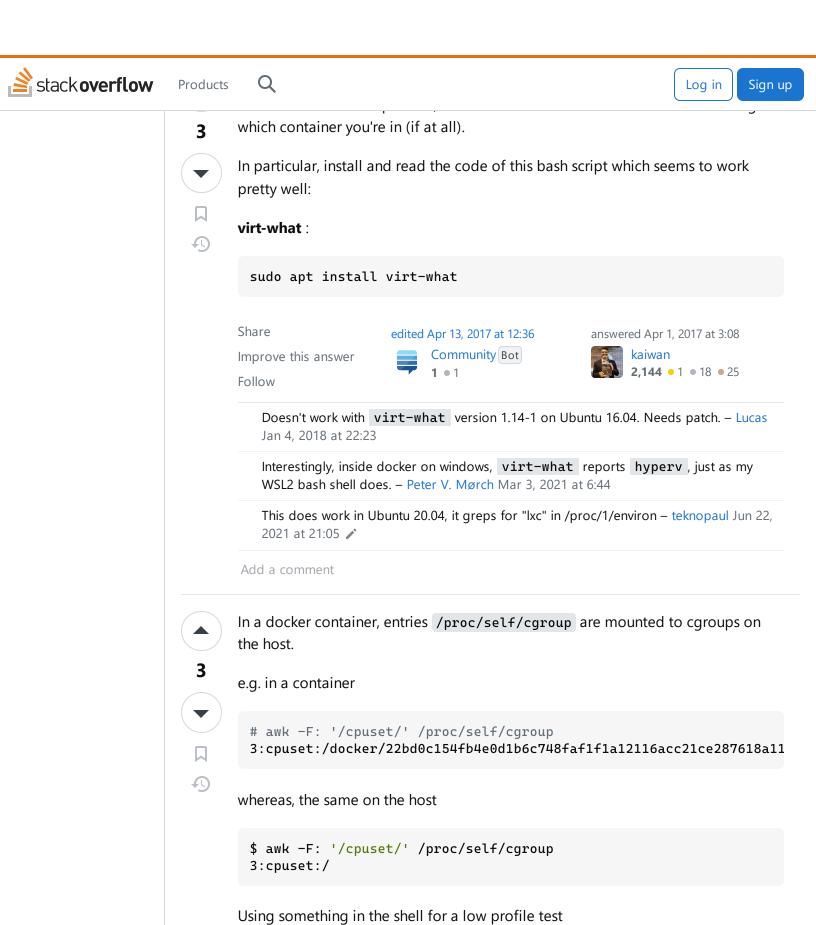
This one did: def is_non_docker(): return
os.path.exists('/proc/1/cgroup') as per the accepted answer here
stackoverflow.com/questions/20010199/... - splintercell Nov 26, 2019 at 18:08

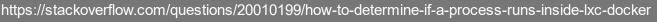
7 You get a Useless Use Of Cat Award. And Useless Use Of Subprocess one. – Jan Hudec May 6, 2020 at 15:00 /

Yeah this is a whole new level of unnecessary cat! Nice one :-D – Timmmm May 7, 2020 at 7:14

You're right, I'll update it the answer even though it's still not all-encompassing. @JanHudec – blakev May 7, 2020 at 17:32

Show 2 more comments







Q

Log in

Sign up

```
if is_running_in_container; then
  echo "Aye!! I'm in a container"
  echo "Nay!! I'm not in a container"
fi
```

Share

edited Oct 7, 2022 at 19:50

answered Jul 15, 2018 at 21:13

Improve this answer

Follow



Returns 1 on both. – sorin Feb 1, 2020 at 14:37

Works! Thanks! @sorin, it actually returns 1 only on host-level, but inside container returns zero. – Dmitry Shevkoplyas Oct 5, 2022 at 23:53

Add a comment



My answer only applies for **Node.js processes** but may be relevant for some visitors who stumble to this question looking for a Node.js specific answer.





I had the same problem and relying on /proc/self/cgroup I created an npm **package** for solely this purpose — to detect whether a Node.js process runs inside a Docker container or not.





The <u>containerized npm module</u> will help you out in Node.js. It is not currently tested in lo.js but may just as well work there too.

Share Improve this answer Follow

answered Sep 23, 2015 at 23:05



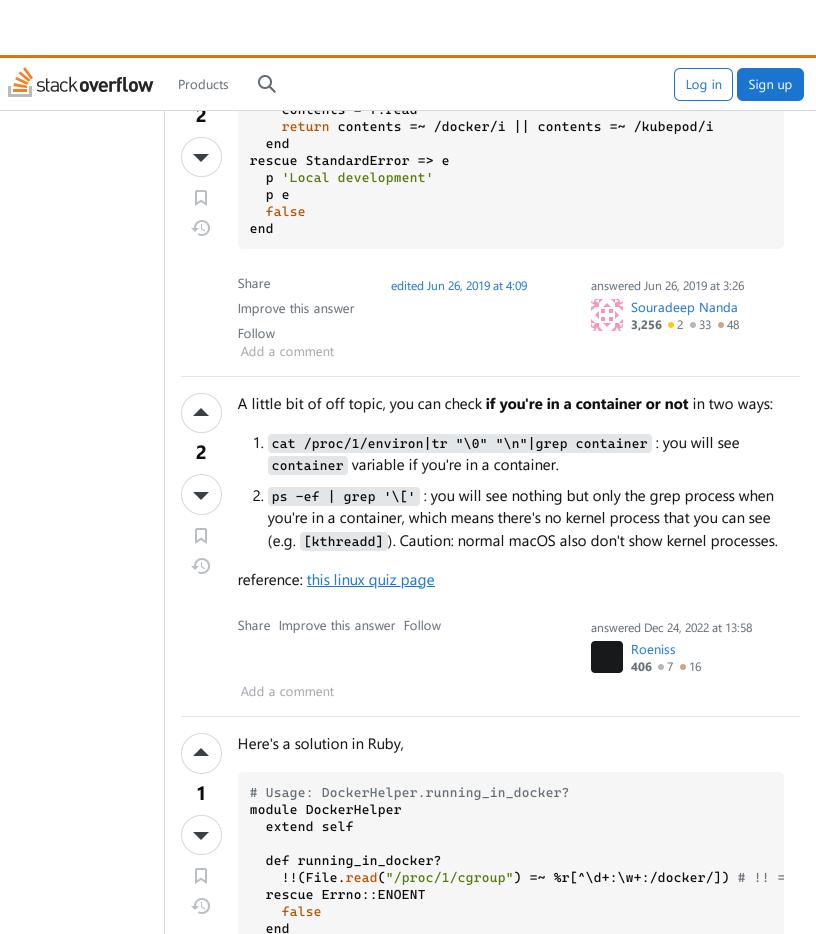
6,140 • 1 • 22 • 15

Thanks for this module, seems to be a couple of open fixes pending - are you still maintaining this? - stevokk Aug 2, 2018 at 8:51

This module is no longer maintained. Additionally, a Node.js module that detects non-Docker container environments is needed. – Jeff Apr 29, 2022 at 14:12

Add a comment

I have translated JJC's answer into ruby



end





Log in

Sign up

Snare improve this answer Follow

answered Sep 24, 2020 at 20:56



Add a comment



The golang code get pid container_id and you can get map container_id get docker image









```
func GetContainerID(pid int32) string {
    cgroupPath := fmt.Sprintf("/proc/%s/cgroup", strconv.Itoa(int(p
    return getContainerID(cgroupPath)
}
func GetImage(containerId string) string {
    if containerId == "" {
        return ""
    image, ok := containerImage[containerId]
    if ok {
        return image
    } else {
        return ""
    }
}
func getContainerID(cgroupPath string) string {
    containerID := ""
    content, err := ioutil.ReadFile(cgroupPath)
    if err != nil {
        return containerID
    lines := strings.Split(string(content), "\n")
    for _, line := range lines {
        field := strings.Split(line, ":")
        if len(field) < 3 {</pre>
            continue
        cgroup_path := field[2]
        if len(cgroup_path) < 64 {</pre>
            continue
        }
        // Non-systemd Docker
        //5:net_prio,net_cls:/docker/de630f22746b9c06c412858f26ca28
        //2.nat cls./buhanade/hurstahla/nad5f200c1a-f0fc-11a8-hf65-
```

Share Improve this answer Follow

answered Dec 15, 2020 at 9:45



Q

Log in

Sign up

Auu a comment



Docker is evolving day by day, so we can't say for sure if they are going to keep .dockerenv .dockerinit in the future.





In most of the Linux flavours <u>init</u> is the first process to start. But in case of containers this is not true.





```
#!/bin/bash
if ps -p1|grep -q init;then
   echo "non-docker"
else
   echo "docker"
fi
```

Share Improve this answer Follow

answered May 11, 2016 at 3:36



- @RomanTrofimov LXC/Docker doesn't either. What a funny comment. abourget Mar 3, 2017 at 17:41
- 2 It does not work in centos 7 as well. When I run in my host machine it says docker. Looks like systemd is running as process id 1 Venkateswara Rao Jan 26, 2018 at 5:02
 - @VenkateswaraRao This must be run inside the container. The intent is to find out if you are inside a docker container or not. Govind Kailas Jan 30, 2018 at 4:43
- 1 @GovindKailas: The problem is that this assumes that the normal PID one is init, which is not true on systemd or launchd based systems... Gert van den Berg Feb 16, 2018 at 14:24
- @SamThomas: launchd, upstart, Solaris SMF, systemd, Sys V style init, BSD style init (these two and some others might call their PID 1 init though), OpenRC, initng, runit.

 See here. Most modern Linux-based systems would use systemd, some older ones, upstart.... All modern OS X systems would use launchd Gert van den Berg May 23, 2018 at 8:16

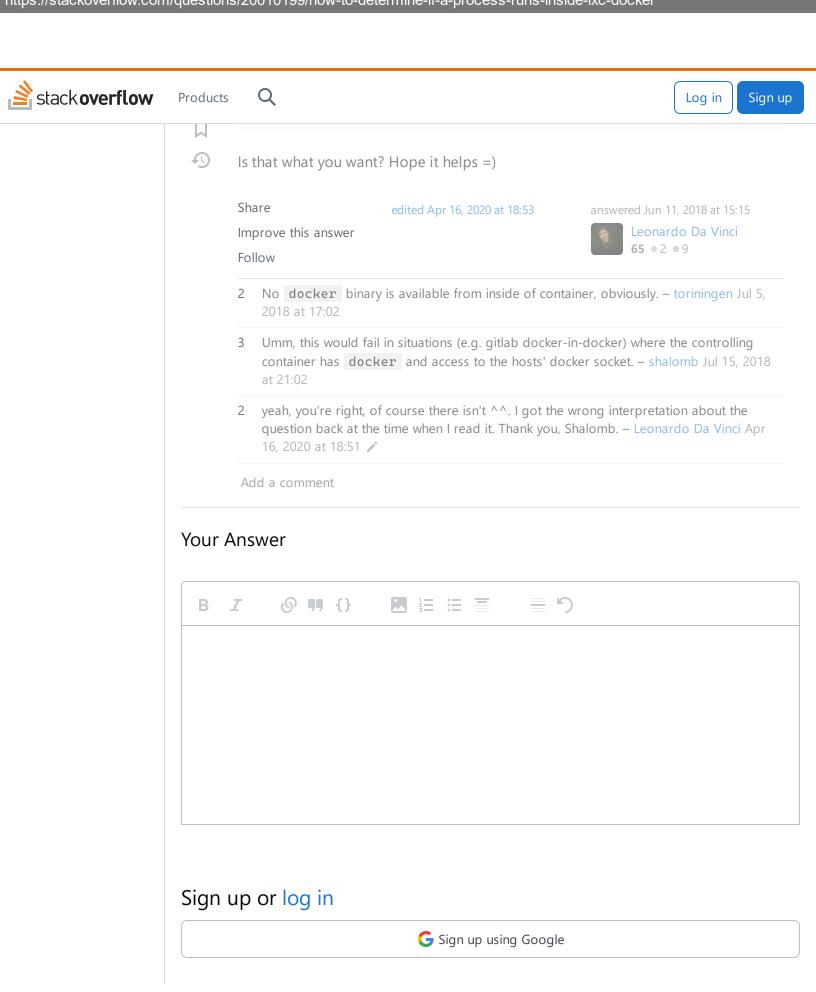
Show 1 more comment

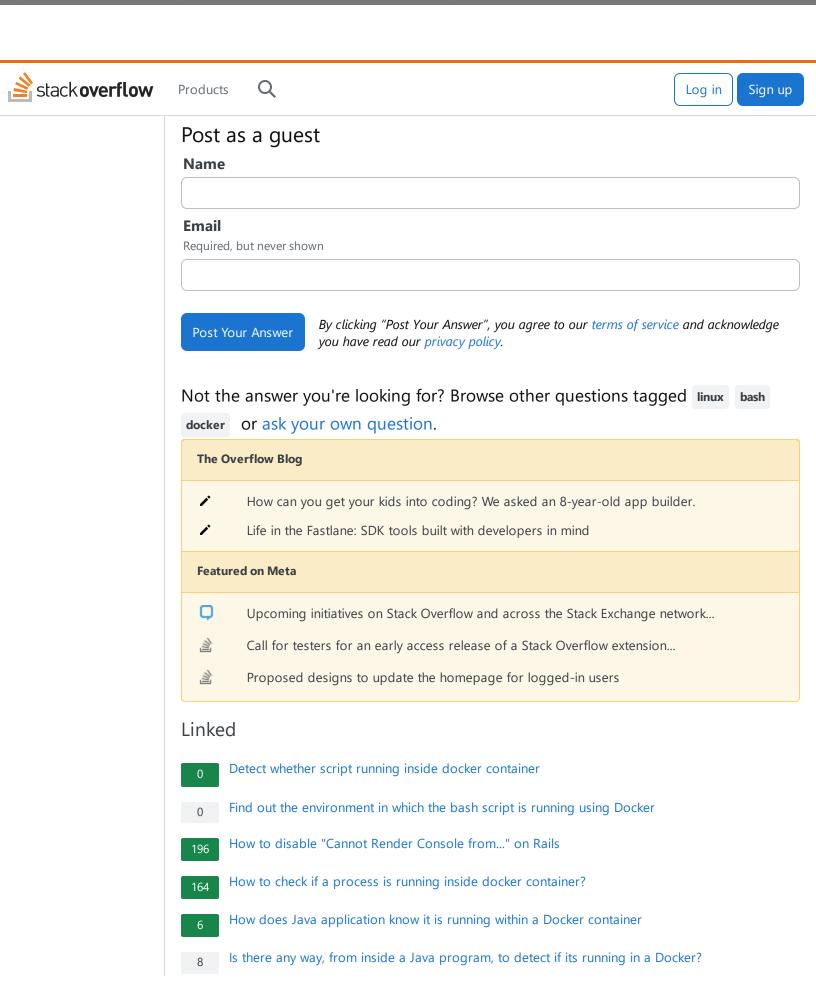


Maybe this do the trick:

-7

```
if [ -z $(docker ps -q) ]; then
  echo "There is not process currently running"
```









Log in

Sign up

- Agnostic bash check that detects is running inside containers or not
- While return si_addr with offset from sigwaitinfo function causing segmentation fault

See more linked questions

Related

- How to check if a process is running inside docker container?
- making sure a given docker container is running
- A way to test if processes running inside docker are on the same host
- How to know a process is running under docker?
- How does one detect if one is running within a docker container within Python?
- How to check if docker daemon is running?
- detect a process running on the host into a docker container
- How to check if any application is running in docker container?
- How to detect if the current script is running in a docker build?
- How can a script check if a particular Docker container is running?

Hot Network Questions

- Undead Uniforms
- Simultaneous clang, hesitation and drop when I pedal with any weight
- Why did James not defend Paul? Does James failing to defend Paul mean that James' faith is one without works?
- As someone that has been laid off how do I respond when an interviewer asks me why am I looking for a job?
- My client keeps changing password and denying me access before the project is fully complete. How do I politely ask him not to do so?
- Does "He is the image of the invisible God" Imply that Jesus is not God?





Log in

Sign up

- Why was Adam considered unique as a talking creature when the snake could speak as well?
- φ Does Pascal's Wager refute Apatheism?
- Colour the vertices of a polygon
- Short science fiction story about "Full Contact Football"
- Positioning the Sefer Torah poles
- How can I use a pattern of every nth row reference from another sheet in Excel?
- Margarian Photodiode reverse light current question
- How to position the node with TikZ.positioning relative to specific angle?
- A play where a corpse keeps growing and growing from one scene to the next one
- Short-stay Schengen Visa for Germany: Which purpose?
- (Why) Is TMS really needed to reference NMR spectra?
- What is the character [esc][cc][esc]?
- Is it possible for a voter to be rendered ineligible in both their prior and new states while in the process of moving?
- what are Philosophical ways of ending unproductive debates?
- Precise Electronic Measurements
- **Question feed**



STACK OVERFLOW

Questions Help Chat

PRODUCTS

Teams Advertising Talent

COMPANY

About Press Work Here Legal Privacy Policy Terms of Service Contact Us Cookie Settings Cookie Policy

STACK EXCHANGE NETWORK

Technology Culture & recreation Life & arts Science Professional Business API Data



Site design / logo © 2024 Stack Exchange Inc; user contributions licensed under $\underline{\text{CC BY-SA}}$. rev 2024.10.29.17670