

How to access host port from docker container [duplicate]

Asked 9 years, 1 month ago Modified 6 months ago Viewed 892k times



770



This question already has answers here:

[From inside of a Docker container, how do I connect to the localhost of the machine?](#) (45 answers)

Closed 2 years ago.

I have a docker container running jenkins. As part of the build process, I need to access a web server that is run locally on the host machine. Is there a way the host web server (which can be configured to run on a port) can be exposed to the jenkins container?

I'm running docker natively on a Linux machine.

UPDATE:

In addition to @larsks answer below, to get the IP address of the Host IP from the host machine, I do the following:

```
ip addr show docker0 | grep -Po 'inet \K[\d.]+'
```

docker

docker-container

Share Improve this question Follow

edited Mar 14, 2022 at 9:52

asked Jul 9, 2015 at 18:01



Tri Nguyen

10.8k 9 45 76

149 170



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

... you can generally access it on 172.17.1.78 --
... at 18:16

Failed to connect to 172.17.1.78

... at 18:41

... nning Docker natively on Linux? – larsks Jul

... st.docker.internal also works, I just have
... e URL. Hope it helps someone. – WesternGun

... ably shouldn't have been closed.

17 Answers

Sorted by: Highest score (default) 

▲ For all platforms

989 Docker v 20.10 and above (since December 14th 2020)

▼ Use your internal IP address or connect to the special DNS name `host.docker.internal` which will resolve to the internal IP address used by the host.



This is for development purpose and does not work in a production environment outside of Docker Desktop.

Linux caveats

To enable this in **Docker** on Linux, add `--add-host=host.docker.internal:host-gateway` to your `docker` command to enable the feature.

To enable this in **Docker Compose** on Linux, add the following lines to the container definition:

```
extra_hosts:
  - "host.docker.internal:host-gateway"
```

According to some users the special DNS name only works within the Docker's default `bridge` network, not within custom networks.

For older macOS and Windows versions of Docker



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

DNS name `host.docker.internal` which

[x/issues/264](#)

instead.

Same as above but use `docker.for.mac.localhost` instead.

Docker for Mac 17.05 and below

To access host machine from the docker container you must attach an IP alias to your network interface. You can bind whichever IP you want, just make sure you're not using it to anything else.

```
sudo ifconfig lo0 alias 123.123.123.123/24
```

Then make sure that your server is listening to the IP mentioned above or `0.0.0.0`. If it's listening on localhost `127.0.0.1` it will not accept the connection.

Then just point your docker container to this IP and you can access the host machine!

To test you can run something like `curl -X GET 123.123.123.123:3000` inside the container.

The alias will reset on every reboot so create a start-up script if necessary.

Solution and more documentation here: <https://docs.docker.com/desktop/networking/#use-cases-and-workarounds-for-all-platforms>

Share Improve this answer Follow

edited Feb 26 at 12:01

answered Apr 21, 2017 at 11:36



EsmaeelE

2,619 ● 6 ● 25 ● 33



Janne Annala

28k ● 8 ● 34 ● 43

1 I've successfully tested this. No need to turn off the firewall – [alvaro g](#) May 22, 2017 at 13:49

19 changed to `host.docker.internal` [docs.docker.com/docker-for-mac/networking/...](https://docs.docker.com/docker-for-mac/networking/) – [Snowball](#) Mar 27, 2018 at 12:17



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

untu? After having run `docker run -it`
d:/# `ping host.docker.internal` gives
[non Forsberg](#) Apr 14, 2018 at 16:08

container and config `127.0.0.1`
8 at 11:14 ✎

the WSL change the IP on every reboot :/

s host services using the IP address of
ill be your default route.

```
7: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
```



```
group default
  link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
  inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
    valid_lft forever preferred_lft forever
  inet6 fe80::f4d2:49ff:fedd:28a0/64 scope link
    valid_lft forever preferred_lft forever
```

And inside a container:

```
# ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 src 172.17.0.4
```

It's fairly easy to extract this IP address using a simple shell script:

```
#!/bin/sh

hostip=$(ip route show | awk '/default/ {print $3}')
echo $hostip
```

You may need to modify the `iptables` rules on your host to permit connections from Docker containers. Something like this will do the trick:

```
# iptables -A INPUT -i docker0 -j ACCEPT
```

This would permit access to any ports on the host from Docker containers. Note that:

- `iptables` rules are ordered, and this rule may or may not do the right thing depending on what other rules come before it.



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

... either (a) listening on `INADDR_ANY` (aka `0.0.0.0` interface).

... you can connect to the magic hostname

... host network namespace by setting `--` ...
... me as `localhost` inside the container, so
... vices and will be accessible without any

... t 2:24

answered Jul 9, 2015 at 20:54



larsks

304k ● 41 ● 442 ● 451

3 Relevant link: [adding entries to container's /etc/hosts](#) – [chronos](#) Mar 22, 2016 at 10:25

11 How about Docker for MAC? AFAIK there is no docker0 network available for "Docker for MAC ". In that case how can I connect to host from container? – [Vijay](#) Jul 27, 2016 at 19:41

This works - but if you're trying to connect to a service like redis, you need to add the ip address in the `ip addr show docker0` command to the bind in redis.conf and restart redis. – [James O'Brien](#) Apr 1, 2018 at 19:35

1 I have used hostname of my host instead of getting IP address (hostname command on host) – [Marek F](#) Apr 9, 2019 at 12:30

1 After installing commands "ip" "dig" "ping" etc into the container (service) for testing, I have found... Ping is always successful, but attempted to access the published ports of another service results in "No Route to Host". However turning off iptables, and this changes to "Could not resolve host" and using the IP directly for the connection works! So I reenabled the iptables and added the appropriate ACCEPT rules, but still not having much luck. Work continues. – [anthony](#) Apr 28, 2021 at 23:27



Use `--net="host"` in your `docker run` command, then `localhost` in your docker container will point to your docker host.

199

Share Improve this answer Follow

answered Feb 15, 2018 at 12:08



[samthebest](#)

31.4k ● 25 ● 104 ● 147



42 Won't work for those using Docker for Windows / Docker for Mac as far as I understand due to the fact that containers run in the virtualized environments: Hyper-V (Windows) / xhyve (Mac) – [ninjaboy](#) May 9, 2018 at 13:06

1:17

de: "host" – [jbarros](#) Feb 9, 2019 at 18:12

h between two dockers containers and as i

– [bormat](#) Dec 11, 2019 at 9:14

makes Docker use [host networking](#): "network (shares the host's networking namespace)". Which

– [Raketenolli](#) Jul 29, 2020 at 8:56

http://host.docker.internal.



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

why:

 [Source in the docs of Docker.](#)

 My google search brought me to here, and after digging in the comments I found it's a duplicate of [From inside of a Docker container, how do I connect to the localhost of the machine?](#). I voted for closing this one as a duplicate, but since people (including myself!) often scroll down on the answers rather than reading the comments carefully, here is a short answer.

Share Improve this answer Follow

edited Mar 16, 2023 at 10:31

answered Apr 26, 2021 at 19:01

 **paulie4**
487 ● 3 ● 12

 **Aidin**
29k ● 10 ● 79 ● 80

- 2 This is a perfect solution unless you dont have to make changes locally in your code base – [Deekshith Anand](#) Jan 10, 2022 at 12:21 
- 2 Yup thats it. @DeekshithAnand , generally I'd advise structuring your code so that URLs and the like belong in environment variables (ie something like a .env file w/ appropriate loader). That way you can do stuff like this without compromising production repeatability – [Shayne](#) Jun 27, 2022 at 1:19 
- 10 This won't work automatically, but you need to provide the following run flag: `--add-host=host.docker.internal:host-gateway` – [Eugen Konkov](#) Aug 6, 2022 at 12:24
- 1 dial tcp: lookup host.docker.internal on 127.0.0.11:53: no such host – [Alexey Sh.](#) Jan 19, 2023 at 12:21
- 2 You have no idea how many hours I wasted on this just to find the concise answer, thank you. – [kennysliding](#) Aug 2, 2023 at 10:11 

 For linux systems, you can – starting from major version `20.04` of the docker engine – now also communicate with the host via `host.docker.internal`. This won't work *automatically*, but you need to provide the following run flag:



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

[comment-578729356](#)

[comment-598864064](#)

answered Apr 25, 2020 at 10:56

 **Samuel**
5,523 ● 5 ● 31 ● 35

it in a similar manner using `--add-host`

`ep -Po 'inet \K[\\d.]+')` – [conny](#) Sep 1,

- 2 is `host-gateway` a keyword or does it have to be the explicit gateway of the host? hm – [enrm](#) Nov 9, 2020 at 7:50

- 7 Docker Engine 20.10 was just released and with it, `host-gateway` should finally be available. – [Cellane](#) Dec 14, 2020 at 4:24

- 1 [@SzczepanHolyszewski](#) Just tried it and YES. – [zaf](#) Jul 6, 2021 at 11:55



Solution with docker-compose: For accessing to host-based service, you can use `network_mode` parameter https://docs.docker.com/compose/compose-file/#network_mode

55



```
version: '3'
services:
  jenkins:
    network_mode: host
```



EDIT 2020-04-27: recommended for use only in local development environment.

EDIT 2021-09-21: IHaveHandedInMyResignation wrote it does not work for Mac and Windows. Option is supported only for Linux

Share Improve this answer Follow

edited Sep 21, 2021 at 14:09

answered Jun 14, 2018 at 21:11



[vovan](#)
1,510 ● 12 ● 22

- 1 Then how to access jenkins? Seems the port forwarding is not working if use host network mode – [Jeff Tian](#) Mar 24, 2020 at 10:35

- 8 it's a very risky solution and not at all recommended. we should NOT open our host network to containers unless explicitly needed – [FatemeH Maid](#) Apr 24, 2020 at 7:49

ws. Option is supported only for Linux

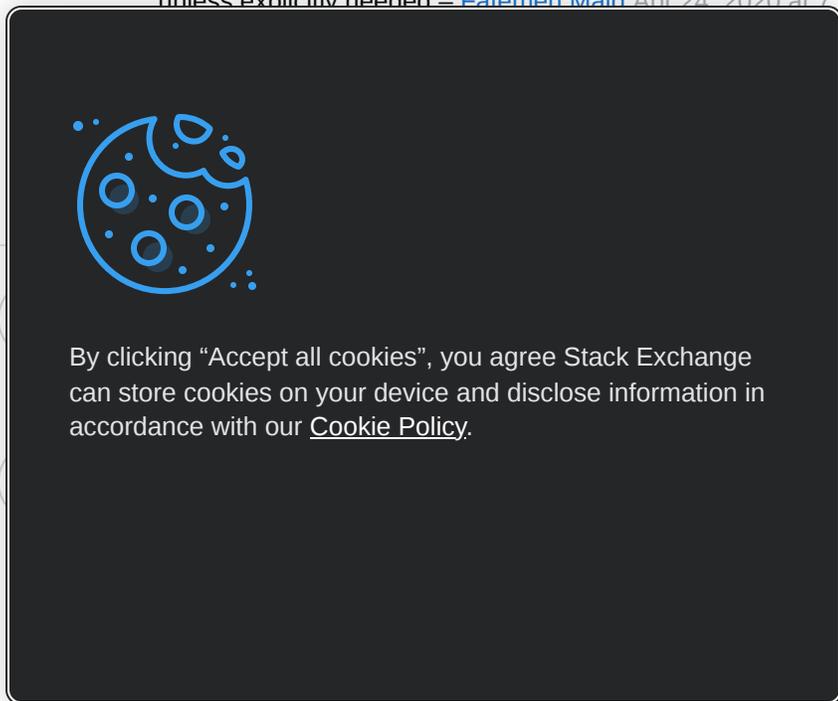
<https://github.com/qoomon/docker-host>

host system e.g. `curl`

answered Aug 6, 2018 at 8:36



[qoomon](#)
5,138 ● 1 ● 23 ● 31



- 1 That's a clever solution. Are you aware of anything using this with a lot of traffic? There might be overhead to proxying all the traffic through this container. – [bcoughlan](#) Sep 4, 2018 at 15:31
- 4 Yes it works quite nice barely no overhead at all because it just works over loopback-device – [qoomon](#) Sep 6, 2018 at 13:46
- 2 Super clean solution, Thanks! – [silencej](#) Jul 21, 2020 at 2:20



26

Currently the easiest way to do this on Mac and Windows is using host `host.docker.internal`, that resolves to host machine's IP address. Unfortunately it [does not work on linux yet](#) (as of April 2018).



Share Improve this answer Follow

answered Apr 20, 2018 at 19:47



[Tomáš Fejfar](#)

11.2k ● 8 ● 57 ● 84



This solution worked with Docker version 19.03.1. Many other solutions given here do not work. This is documented at [docs.docker.com/docker-for-mac/networking/...](#) – [clay](#) Aug 12, 2019 at 16:31

- 1 this is great. `ping host.docker.internal` revealed the host machine's ip and I was able to connect to it. thank you! – [Tamas Kalman](#) Apr 28, 2021 at 2:08



17

I've explored the various solution and I find this the least hacky solution:

1. Define a static IP address for the bridge gateway IP.
2. Add the gateway IP as an extra entry in the `extra_hosts` directive.

projects doing this, you have to ensure that



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

```
- subnet: 172.20.0.0/16
  gateway: 172.20.0.1
```

You can then access ports on the host from inside the container using the hostname "dockerhost".

Share Improve this answer Follow

edited Nov 6, 2020 at 0:12

answered Sep 6, 2018 at 14:09



Pang

9,996 ● 146 ● 85 ● 124



bcoughlan

26.5k ● 20 ● 92 ● 147



16

We found that a simpler solution to all this networking junk is to just use the domain socket for the service. If you're trying to connect to the host anyway, just mount the socket as a volume, and you're on your way. For postgresql, this was as simple as:



```
docker run -v /var/run/postgresql:/var/run/postgresql
```



Then we just set up our database connection to use the socket instead of network. Literally that easy.

Share Improve this answer Follow

answered May 7, 2019 at 17:19



mlissner

18k ● 18 ● 118 ● 174

5 FYI, we ran into a big issue with this: Docker for Mac doesn't support sockets as mounted volumes. This went swimmingly until a Mac person tried it. :(– mlissner Sep 12, 2019 at 22:22



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

a private network between containers, the
the egress interface from the containers
interface id, such as:

```
MULTICAST> mtu 1500
broadcast 192.168.47.255
```

change each time compose has to
n to this is to create the private network in
range:

```

services:
  mongodb:
    image: mongo:4.2.2
    networks:
      - mynet
    # rest of service config and other services removed for clarity

networks:
  mynet:
    name: mynet
    ipam:
      driver: default
      config:
        - subnet: "192.168.32.0/20"

```

You can change the subnet if your environment requires it. I arbitrarily selected `192.168.32.0/20` by using `docker network inspect` to see what was being created by default.

Configure `iptables` on the host to permit the private subnet as a source:

```
$ iptables -I INPUT 1 -s 192.168.32.0/20 -j ACCEPT
```

This is the simplest possible `iptables` rule. You may wish to add other restrictions, for example by destination port. Don't forget to persist your iptables rules when you're happy they're working.

This approach has the advantage of being repeatable and therefore automatable. I use ansible's `template` module to deploy my compose file with variable substitution and then use the `iptables` and `shell` modules to configure and persist the firewall rules, respectively.

Share Improve this answer Follow

edited Nov 6, 2020 at 0:13

answered Jan 1, 2020 at 16:37



Pang

996 ● 146 ● 85 ● 124



Andy Brown

12.8k ● 2 ● 51 ● 66



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

i `docker0 -j ACCEPT`, but that didn't help
`.0/20 -j ACCEPT` suggested here solved my

7, 2021 at 16:28

`192.168.32.0/20`. That persists restarts.

e of those fit well enough to my context.
 tain any of the networking tools
 e container.



Also, using the `--net="host"` approach is a very rough approach that is not applicable when one wants to have well isolated network configuration with several containers.



So, my approach is to extract the host's address at the host's side, and then pass it to the container with `--add-host` parameter:

```
$ docker run --add-host=docker-host:`ip addr show docker0 | grep -Po 'inet \K[\d.]+'` image_name
```

or, save the host's IP address in an environment variable and use the variable later:

```
$ DOCKERIP=`ip addr show docker0 | grep -Po 'inet \K[\d.]+'`
$ docker run --add-host=docker-host:$DOCKERIP image_name
```

And then the `docker-host` is added to the container's hosts file, and you can use it in your database connection strings or API URLs.

Share Improve this answer Follow

answered Mar 24, 2020 at 17:59



Passiday

7,955 ● 10 ● 44 ● 64

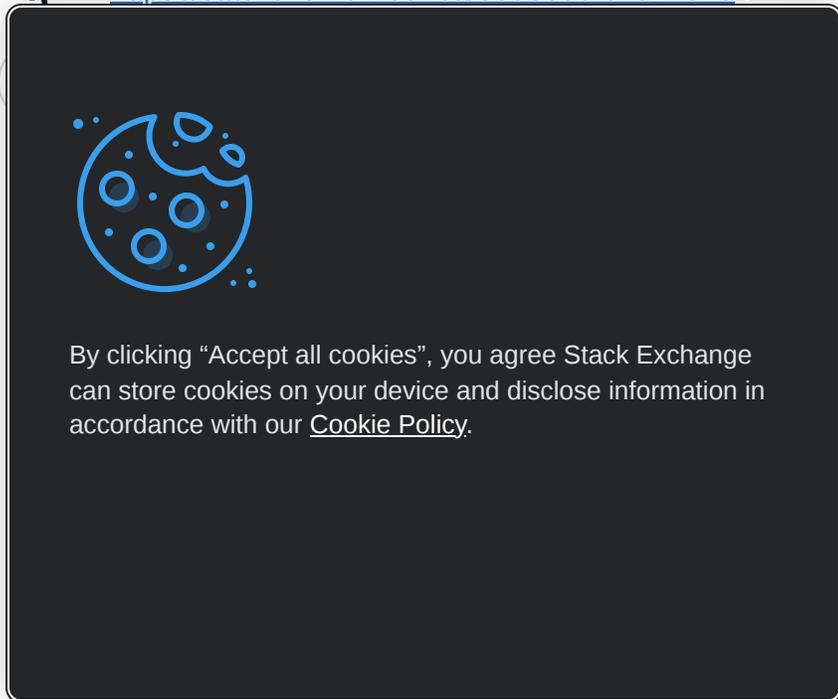
This also allows you to add a specific host, in case you are talking to Apache and you want to speak to a very specific host. `--add-host local.acme.com:$DOCKERIP` – Bruno Bossola Aug 31, 2021 at 13:57



For me (Windows 10, Docker Engine v19.03.8) it was a mix of

<https://stackoverflow.com/a/43541732/7924573> and

<https://stackoverflow.com/a/50866007/7924573>.



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

[085/log"](#)

maintain the port forwarding; if not use

using docker-compose (similar to

ly be used in a local development

environment.

For more information read: https://docs.docker.com/compose/compose-file/#network_mode and <https://docs.docker.com/docker-for-windows/networking/#use-cases-and-workarounds>

Share Improve this answer Follow

answered Jun 2, 2020 at 8:27

 **tschomacker**
774 ● 10 ● 20

host.docker.internal solved the issue for me. Allowed me to access a dev postgres from another container that did not have container access to the other container.

DATABASE_URL=jdbc:postgresql://host.docker.internal:5432/mirthdb – Michael Hobbs Aug 16, 2020 at 16:21

▲
4
▼
🔖
🕒

You can access the local webserver which is running in your host machine in two ways.

1. Approach 1 with public IP

Use host machine public IP address to access webserver in Jenkins docker container.

2. Approach 2 with the host network

Use "--net host" to add the Jenkins docker container on the host's network stack. Containers which are deployed on host's stack have entire access to the host interface. You can access local webserver in docker container with a private IP address of the host machine.

NETWORK ID	NAME	DRIVER	SCOPE
b3554ea51ca3	bridge	bridge	local
2f0d6d6fdd88	host	host	local
b9c2a4bc23b2	none	null	local

Start a container with the host network `Eq: docker run --net host -it ubuntu` and run

h are reachable from docker container.

d I am able to access the nginx website

CREATED	STATUS
22 seconds ago	Up 20

machine) from Ubuntu docker container



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

```

root@linuxkit-025000000001:/# curl 192.168.x.x -I
HTTP/1.1 200 OK
Server: nginx/1.15.10
Date: Tue, 09 Apr 2019 05:12:12 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 26 Mar 2019 14:04:38 GMT
Connection: keep-alive
ETag: "5c9a3176-264"
Accept-Ranges: bytes

```

Share Improve this answer Follow

answered Apr 9, 2019 at 5:47



Sreenivasa Reddy

304 ● 1 ● 7



In almost 7 years the question was asked, it is either docker has changed, or no one tried this way. So I will include my own answer.

3



I have found all answers use complex methods. Today, I have needed this, and found 2 very simple ways:



- use `ipconfig` or `ifconfig` on your host and make note of all IP addresses. At least two of them can be used by the container.
 - I have a fixed local network address on WiFi LAN Adapter: `192.168.1.101`. This could be `10.0.1.101`. the result will change depending on your router
 - I use WSL on windows, and it has its own `vEthernet` address: `172.19.192.1`
- use `host.docker.internal`. Most answers have this or another form of it depending on OS. The name suggests it is now globally used by docker.

in other words IP given by the service and requires routing and firewall settings.

posted this answer there, I first found answer.

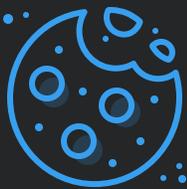
answered Jan 26, 2022 at 21:23



Yilmaz Durmaz

Yilma 2,828 ● 15 ● 32

IP address of my machine on the local



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).



e.g



ifconfig

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether f0:18:98:08:74:d4
    inet 192.168.178.63 netmask 0xfffff00 broadcast 192.168.178.255
    media: autoselect
    status: active
```

and then used the inet address. This worked for me to connect any ports on my machine.

Share Improve this answer Follow

answered Apr 22, 2021 at 15:41



Amrut Prabhu

1,449 ● 13 ● 12



When you have two docker images "already" created and you want to put two containers to communicate with one-another.

-4

For that, you can conveniently run each container with its own --name and use the --link flag to enable communication between them. You do not get this during docker build though.



When you are in a scenario like myself, and it is your



```
docker build -t "centos7/someApp" someApp/
```

That breaks when you try to



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

```
download.tar.gz > dump.tar.gz
```

ost".

by default is banning communication nning on your host. This was quite system of docker machines running on a hout too much hurdle.

wing documentation.

[g.html](#)

ing by lowering down the network

The simplest alternative is just to turn the firewall off - or allow all. This means running the necessary command, which could be `systemctl stop firewalld`, `iptables -F` or equivalent.

Share Improve this answer Follow

edited Mar 14, 2022 at 9:52

answered Nov 16, 2016 at 21:47



desertnaut

59.8k ● 29 ● 149 ● 170



99Sono

3,667 ● 30 ● 41

4 Just as a note, `--link` is now deprecated – [Mr.Budris](#) Sep 22, 2017 at 19:49



By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).