

[How-To] Web services using Swag (Proxy) in docker - Q&A

SN

- - [Jan 18th 2022](#)
 - [#1](#)

This is a Q&A thread for [\[How-To\] Web services using Swag \(Proxy\) in docker](#)

I took the liberty of creating this, because there is no thread yet and thought it might be helpful for others as well.

- - [Jan 18th 2022](#)
 - [#2](#)

So, I'm starting this off with a few questions myself. First of all, thank you, [chente](#) , for creating your helpful tutorials!

I have two questions regarding the user-defined bridge, which might also be of interest to other users:

1. When using SWAG and Jellyfin (for example), the line **network_mode: swag** should be added to both compose files, right? Otherwise, SWAG would stay in its own default network. At least for me, this wasn't clear when reading the tutorial.
2. To isolate additional apps (if I add Nextcloud to the mix, for example), it should be helpful to create another network to isolate traffic between SWAG and Nextcloud, so Nextcloud and Jellyfin containers don't see each other. I was reading the [docker documentation \(Network Tutorial\)](#) and it seems that having separate networks would be beneficial for isolation of traffic.
--> For example: Create two networks named **swag_jellyfin** and **swag_nextcloud**, where SWAG is part of both, while Jellyfin and Nextcloud are only part of their respective bridge.

I haven't tested this and wanted to confirm before I try anything that doesn't make sense. I assume that it would be necessary to use "networks" instead of "network_mode" to add SWAG to several networks, as described in the [docker documentation \(Networking in Compose\)](#)?

- - [Jan 18th 2022](#)
 - [#3](#)

If you really want to isolate the the networks of your containers, you have to

1. Create an individual network for each container (docker network create <name_of_network>)
2. Attach the new container to that network
3. Attach the swag container to all these networks using (in the docker-compose.yml)

Code

```
some-service:
  networks:
    - some-network
    - other-network
```

See: <https://docs.docker.com/compos...compose-file-v3/#networks>

And define the networks as documented here: <https://docs.docker.com/compos...k-configuration-reference>

But why would you do this?

- - [Jan 18th 2022](#)
 - [#4](#)

Thank you for this!

I was reading the [SWAG documentation](#) and they said it makes sense to use a user-defined bridge network (container names can be used as hostnames). Because I didn't quite understand how this worked, I read the [docker documentation](#) and they use an example to isolate the containers. I thought this might be a sensible approach to protect the other containers if one container is compromised, but also making sure that SWAGs reverse proxy still functions?

My docker-compose.yml for SWAG (I removed a few lines under "environment")

Code

```
---
version: "2.1"
services:
  swag:
    image: lscr.io/linuxserver/swag
    container_name: swag
    network_mode: swag
    cap_add:
      - NET_ADMIN
```

```
environment:
  - PUID=1002
  - PGID=100
volumes:
  - /DockerSpace/Config/SWAG:/config
ports:
  - 443:443
restart: unless-stopped
```

[Display More](#)

My docker-compose.yaml for Jellyfin

Code

```
---
version: "2.1"
services:
  jellyfin:
    image: lscr.io/linuxserver/jellyfin
    container_name: jellyfin
    network_mode: swag
    environment:
      - PUID=1002
      - PGID=100
    volumes:
      - /DockerSpace/Config/Jellyfin:/config
      - /MainStorage/Media:/data
    ports:
      - 8096:8096
    restart: unless-stopped
```

[Display More](#)

So, would it be better to change my docker-compose.yaml (instead of using **network_mode = swag**) to something like this?

Code

```
---
version: "2.1"
services:
  jellyfin:
    image: lscr.io/linuxserver/jellyfin
    container_name: jellyfin
    environment:
      - PUID=1002
```

```
- PGID=100
volumes:
  - /DockerSpace/Config/Jellyfin:/config
  - /MainStorage/Media:/data
ports:
  - 8096:8096
restart: unless-stopped
networks:
  - swag
```

Display More

- - [Jan 18th 2022](#)
 - [#5](#)

I am not quite sure about the difference between `network_mode` and `networks`, only that `network_mode` can only take one network and does not require the `networks` definition in the `docker-compose`.

In the last file, you need to add

Code

```
networks:
  swag:
    external: true
```

after line 17. Mind the indent of 0 spaces.

- - [Jan 18th 2022](#)
 - [#6](#)



[Quote from sNiXx](#)

Thank you for this!

I was reading the [SWAG documentation](#) and they said it makes sense to use a user-defined bridge network (container names can be used as hostnames). Because I didn't quite understand how this worked, I read the [docker documentation](#) and they use an example to isolate the containers. I thought this might be a sensible approach to protect the other containers if one container is compromised, but also making sure that SWAGs reverse proxy still functions?

My docker-compose.yaml for SWAG (I removed a few lines under "environment")

My docker-compose.yaml for Jellyfin

So, would it be better to change my docker-compose.yaml (instead of using **network_mode = swag**) to something like this?

Display More

is really simple:

1 - in portainer create a new network (in my case my-net)

2 - add this after version "x.x" to your dockers.yaml file to deploy:

Code

```
version: "2.1"
networks:
  default:
    external:
      name: my-net
```

Done, all your dockers are attached to my-net network and can be name resolved by swag (swag must have this lines too to run on my-net network)

and example of my airsonic.yaml file (to deploy in portainer stack:

Code

```
version: "2.1"
networks:
  default:
    external:
      name: my-net
services:
  airsonic:
    image: linuxserver/airsonic
    container_name: airsonic
    environment:
      - PUID=1001
      - PGID=1000
```

```

- TZ=Europe/Madrid
#- CONTEXT_PATH=airsonic.raulfg.es #optional
- JAVA_OPTS=-Xmx2g          #optional
- JAVA_OPTS=-Xms2g          #optional
volumes:
- /srv/dev-disk-by-label-DATA/Data/dockers/airsonic:/config
- /srv/dev-disk-by-label-BPool/Musica:/music
- /srv/dev-disk-by-label-DATA/Data/dockers/airsonic:/playlists
- /srv/dev-disk-by-label-BPool/Musica:/podcasts
- /srv/dev-disk-by-label-BPool/Musica:/media #optional
ports:
- 4050:4040
devices:
- /dev/snd:/dev/snd #optional
restart: unless-stopped

```

Display More

- - [Jan 18th 2022](#)
 - [#7](#)



[Quote from sNiXx](#)

So, I'm starting this off with a few questions myself

I think you already have all the answers you needed. I have modified the explanation in the guide, I think that now all the options are there. 😊

- - [Jan 19th 2022](#)
 - [#8](#)

Thanks to everyone's help, it is now working as expected. I can confirm that pinging containers by hostname that are outside of the network does not work.

Something else I figured out: When using user-defined bridge networks, it is not necessary to expose ports (makes sense when I think about it...)

My files for future reference:

docker-compose.yml for Jellyfin

Code

```

---
version: "2.1"
services:
  jellyfin:
    image: lscr.io/linuxserver/jellyfin
    container_name: jellyfin
    networks:
      - swag-jf
    environment:
      - PUID=1002
      - PGID=100
    volumes:
      - /DockerSpace/Config/Jellyfin:/config
      - /MainStorage/Media:/data
    restart: unless-stopped
networks:
  swag-jf:
    external: true

```

Display More

docker-compose.yml for SWAG

Code

```

---
version: "2.1"
services:
  swag:
    image: lscr.io/linuxserver/swag
    container_name: swag
    networks:
      - swag-jf
      - swag-nc
    cap_add:
      - NET_ADMIN
    environment:
      - PUID=1002
      - PGID=100
    volumes:
      - /DockerSpace/Config/SWAG:/config
    ports:
      - 443:443
    restart: unless-stopped
networks:
  swag-jf:

```

```
    external: true
  swag-nc:
    external: true
```

[Display More](#)

docker-compose.yml for Nextcloud

Code

```
---
version: "2.1"
services:
  nextcloud:
    image: lscr.io/linuxserver/nextcloud
    container_name: nextcloud
    networks:
      - swag-nc
    environment:
      - PUID=1002
      - PGID=100
    volumes:
      - /DockerSpace/Config/Nextcloud:/config
      - /DockerSpace/Data/Nextcloud:/data
    depends_on:
      - mariadb
    restart: unless-stopped
  mariadb:
    image: linuxserver/mariadb
    container_name: mariadb
    networks:
      - swag-nc
    environment:
      - PUID=1002
      - PGID=100
      - MYSQL_ROOT_PASSWORD=mariadbpassword
    volumes:
      - /DockerSpace/Config/MariaDB:/config
    restart: unless-stopped
networks:
  swag-nc:
    external: true
```

[Display More](#)

- - [Jan 19th 2022](#)
 - [#9](#)

Hi all. First of all, I apologize for the writing, I use the translator xD I followed @chente's tutorial for swag (thanks). But I have a problem, when I enter my domain I get the swag welcome message but the browser lock makes me not secure. how can it be solved?

- - [Jan 19th 2022](#)
 - [#10](#)



[Quote from moretty](#)

Hi all. First of all, I apologize for the writing, I use the translator xD I followed @chente's tutorial for swag (thanks). But I have a problem, when I enter my domain I get the swag welcome message but the browser lock makes me not secure. how can it be solved?

Is the output of `docker logs -f swag` correct? Do you get the certificates?

- - [Jan 19th 2022](#)
 - [#11](#)

[cont-init.d] 10-adduser: exited 0.

[cont-init.d] 20-config: executing...

[cont-init.d] 20-config: exited 0.

[cont-init.d] 30-keygen: executing...

using keys found in /config/keys

[cont-init.d] 30-keygen: exited 0.

[cont-init.d] 50-config: executing...

Variables set:

PUID=1000

PGID=100

TZ=Europe/Madrid

URL=xxxx.duckdns.org

SUBDOMAINS=ha.xxxx.duckdns.org

EXTRA_DOMAINS=

ONLY_SUBDOMAINS=true

VALIDATION=http

CERTPROVIDER=

DNSPLUGIN=

EMAIL=xxxxxx@hotmail.com

STAGING=

Using Let's Encrypt as the cert provider

SUBDOMAINS entered, processing

SUBDOMAINS entered, processing

Only subdomains, no URL in cert

Sub-domains processed are: -d ha.xxxx.duckdns.org.xxxx.duckdns.org

E-mail address entered: xxxxxx@hotmail.com

http validation is selected

Certificate exists; parameters unchanged; starting nginx

[cont-init.d] 50-config: exited 0.

[cont-init.d] 60-renew: executing...

The cert does not expire within the next day. Letting the cron script handle the renewal attempts overnight (2:08am).

[cont-init.d] 60-renew: exited 0.

[cont-init.d] 70-templates: executing...

[cont-init.d] 70-templates: exited 0.

[cont-init.d] 90-custom-folders: executing...

[cont-init.d] 90-custom-folders: exited 0.

[cont-init.d] 99-custom-files: executing...

[custom-init] no custom files found exiting...

[cont-init.d] 99-custom-files: exited 0.

[cont-init.d] done.

[services.d] starting services

[services.d] done.

Server ready

- - [Jan 19th 2022](#)
 - [#12](#)

What is the service you are trying to access?

- - [Jan 19th 2022](#)
 - [#13](#)

I haven't gotten there yet because of this certificate issue, but my idea is to remove ngm and use homeassistant that I have in a VM (remember from yesterday). My idea is to change a swag to put fail2ban, geoip, etc. everything that your forum tutorial does, it seems safer than ngm.

- - [Jan 19th 2022](#)
 - [#14](#)



[Quote from moretty](#)

SUBDOMAINS=ha.xxxx.duckdns.org

Your YML/stack is wrong.

Post it here but hide sensible data.

- - [Jan 19th 2022](#)
 - [#15](#)

After what [Soma](#) told you, on the road

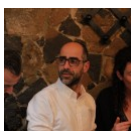
/path_to_your_config_swag/nginx/proxy-confs/

you have a configuration file called homeassistant.subdomain.conf.sample

You must copy this file to another called homeassistant.subdomain.conf and edit it to suit your domain.

- - [Jan 19th 2022](#)
 - [#16](#)

Realemente firefox me indica que tengo un certificado proporcionado por letsencrypt que caduca en abril de 2022 y que la conexion esta cifrada con tls aes



Quote from Soma

Your YAML/stack is wrong.

Post it here but hide sensible data.

you mean the xxxx? I put it on purpose to avoid exposing my information

- - [Jan 19th 2022](#)
 - [#17](#)



Quote from chente

After what [Soma](#) told you, on the road

`/path_to_your_config_swag/nginx/proxy-confs/`

you have a configuration file called `homeassistant.subdomain.conf.sample`

You must copy this file to another called `homeassistant.subdomain.conf` and edit it to suit your domain.

If I understand that and it is very clear in your tutorial, I am not doing it because when I try to enter my domain and I see the swag welcome screen and I see that the browser lock says not secure I do not continue for fear, that is my really ask. that is normal??

I also tell you that I look in firefox and it tells me that I have a certificate obtained by letsencrypt until April 2022 and the connection is encrypted with tls aes.

P.D. right now i made a jellyfin container i add it to the swag network and it works perfectly.

but the same thing happens to me the padlock that goes before https puts not sure that's what you don't put right.

- - [Jan 19th 2022](#)
 - [#18](#)



Quote from moretty

you mean the xxxx? I put it on purpose to avoid exposing my information

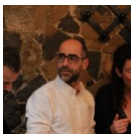
Not that, your swag docker-compose.yml has an error on that line I posted:

Since you didn't post it, I'm guessing:

Code

```
.....
PUID=1000
PGID=100
TZ=Europe/Madrid
URL=xxxx.duckdns.org
SUBDOMAINS=ha.xxxx.duckdns.org # <--- This is wrong. only use the name
you want. For eg:
SUBDOMAINS=www,ha,nextcloud,jellyfin # will make certs for all of
those.
EXTRA_DOMAINS=
ONLY_SUBDOMAINS=true
.....
```

- - [Jan 19th 2022](#)
 - [#19](#)



[Quote from Soma](#)

Not that, your swag docker-compose.yml has an error on that line I posted:

Since you didn't post it, I'm guessing:

Code

```
.....
PUID=1000
PGID=100
TZ=Europe/Madrid
URL=xxxx.duckdns.org
SUBDOMAINS=ha.xxxx.duckdns.org # <--- This is wrong. only use
the name you want. For eg:
SUBDOMAINS=www,ha,nextcloud,jellyfin # will make certs for all
of those.
EXTRA_DOMAINS=
ONLY_SUBDOMAINS=true
.....
```

Oh yeah. what's more I realized before the message because I created jellyfin, but now it's correct and it works but the lock in jellyfin keeps putting not sure

- - [Jan 19th 2022](#)
 - [#20](#)

If you post your docker-compose.yml, it will be easier to make it work for ALL services you need.

You're using duckdns, so it's better to use wildcard.

That way, you'll have subdomain access without much confusion:

nextcloud.xxxxxx.duckdns.org

jellyfin.xxxxx.duckdns.org

ha.xxxxxxxduckdns.org

etc

etc

Participate now!