
Unbound DNS Server Docker Image

Supported tags and respective Dockerfile links

- 1.18.0, [latest](#) (1.18.0/Dockerfile)
- 1.17.1, (1.17.1/Dockerfile)
- 1.17.0, (1.17.0/Dockerfile)
- 1.16.3, (1.16.3/Dockerfile)
- 1.16.2, (1.16.2/Dockerfile)
- 1.16.1, (1.16.1/Dockerfile)
- 1.16.0, (1.16.0/Dockerfile)
- 1.15.0, (1.15.0/Dockerfile)
- 1.14.0, (1.14.0/Dockerfile)

What is Unbound?

Unbound is a validating, recursive, and caching DNS resolver. > [unbound.net](#)

How to use this image

Standard usage

Run this container with the following command:

```
1 docker run \  
2 --name=my-unbound \  
3 --detach=true \  
4 --publish=53:53/tcp \  
5 --publish=53:53/udp \  
6 --restart=unless-stopped \  
7 mvance/unbound:latest
```

By default, this image forwards queries Cloudflare DNS server over TLS. In other words, it does not act as a recursive server. The unbound.sh file provides the configuration unless it is overridden as described below.

Note: The example unbound.conf file is different from the one set by unbound.sh file. The example is provided to help you re-configure this as a recursive server.

Override default forward

By default, forwarders are configured to use Cloudflare DNS. You can retrieve the configuration in the `forward-records.conf` file.

You can create your own configuration file and override the one placed in `/opt/unbound/etc/unbound/forward-records.conf` in the container. This is useful if you prefer to use something other than Cloudflare DNS but do not want to provide a custom `unbound.conf` file.

Example `forward-records.conf`:

```
1 forward-zone:
2   # Forward all queries (except those in cache and local zone) to
3   # upstream recursive servers
4   name: "."
5
6   # my DNS
7   forward-addr: 192.168.0.1@53#home.local
```

Another example `forward-records.conf`:

```
1 forward-zone:
2   # Forward all queries (except those in cache and local zone) to
3   # upstream recursive servers
4   name: "."
5   # Queries to this forward zone use TLS
6   forward-tls-upstream: yes
7
8   ## CleanBrowsing Family Filter
9   forward-addr: 185.228.168.168@853#family-filter-dns.cleanbrowsing.
10  org
11  forward-addr: 185.228.169.168@853#family-filter-dns.cleanbrowsing.
12  org
```

Once the file has your entries in it, mount your version of the file as a volume when starting the container:

```
1 docker run \
2   --name my-unbound \
3   --detach=true \
4   --publish=53:53/tcp \
5   --publish=53:53/udp \
6   --restart=unless-stopped \
7   --volume $(pwd)/forward-records.conf:/opt/unbound/etc/unbound/forward-
8   records.conf:ro \
9   mvance/unbound:latest
```

Use a customized Unbound configuration

Instead of using this image's default configuration for Unbound, you may supply your own configuration. If your customized configuration is located at `/my-directory/unbound/unbound.conf`, pass `/my-directory/unbound` as a volume when creating your container:

```
1 docker run --name=my-unbound \  
2 --detach=true \  
3 --publish=53:53/tcp \  
4 --publish=53:53/udp \  
5 --restart=unless-stopped \  
6 --volume=/my-directory/unbound:/opt/unbound/etc/unbound/ \  
7 mvance/unbound:latest
```

This will expose all files in `/my-directory/unbound/` to the container. As an alternate way to serve custom DNS records for any local zones, either place them directly in your `unbound.conf`, or place the local zones in a separate file and use Unbound's include directive within your `unbound.conf`:

```
1 include: /opt/unbound/etc/unbound/local-zone-unbound.conf
```

Your volume's contents might eventually look something like this:

```
1 /my-directory/unbound/  
2 -- unbound.conf  
3 -- local-zone-unbound.conf  
4 -- secret-zone.conf  
5 -- some-other.conf
```

Overall, this approach is very similar to the `a-records.conf` approach described below. However, by passing your unbound directory rather than a single file, you have more options for customizing and segmenting your Unbound configuration.

Note: Care has been taken in the image's default configuration to enable security options so it is recommended to use it as a guide.

Run on different port

If you want to run Unbound on a different port such as 5353, modify the publish flags:

```
1 sudo docker run \  
2 --name=my-unbound \  
3 --publish=5353:53/tcp \  
4 --publish=5353:53/udp \  
5 --detach=true \  
6 --restart=unless-stopped \  

```

```
7 --volume=$(pwd)/my-directory/forward-records.conf:/opt/unbound/etc/
  unbound/forward-records.conf:ro \
8 --volume=$(pwd)/my-directory/a-records.conf:/opt/unbound/etc/unbound/a-
  records.conf:ro \
9 mvance/unbound:latest
```

Serve Custom DNS Records for Local Network

While Unbound is not a full authoritative name server, it supports resolving custom entries on a small, private LAN. In other words, you can use Unbound to resolve fake names such as `your-computer.local` within your LAN.

To support such custom entries using this image, you need to provide an `a-records.conf` or `srv-records.conf` file. This configuration file is where you will define your custom entries for forward and reverse resolution.

A records The `a-records.conf` file should use the following format:

```
1 # A Record
2 #local-data: "somecomputer.local. A 192.168.1.1"
3 local-data: "laptop.local. A 192.168.1.2"
4
5 # PTR Record
6 #local-data-ptr: "192.168.1.1 somecomputer.local."
7 local-data-ptr: "192.168.1.2 laptop.local."
```

Once the file has your entries in it, mount your version of the file as a volume when starting the container:

```
1 docker run \
2 --name my-unbound \
3 --detach=true \
4 --publish=53:53/tcp \
5 --publish=53:53/udp \
6 --restart=unless-stopped \
7 --volume $(pwd)/a-records.conf:/opt/unbound/etc/unbound/a-records.conf:
  ro \
8 mvance/unbound:latest
```

SRV records The `srv-records.conf` file should use the following format:

```
1 # SRV records
2 # _service._proto.name. | TTL | class | SRV | priority | weight | port
  | target.
```

3	_etcd-server-ssl._tcp.domain.local.	86400	IN	SRV	0	10
	2380 etcd-0.domain.local.					
4	_etcd-server-ssl._tcp.domain.local.	86400	IN	SRV	0	10
	2380 etcd-1.domain.local.					
5	_etcd-server-ssl._tcp.domain.local.	86400	IN	SRV	0	10
	2380 etcd-2.domain.local.					

Run a container that use this SRV config file:

```
1 docker run \  
2 --name my-unbound \  
3 --detach=true \  
4 --publish=53:53/tcp \  
5 --publish=53:53/udp \  
6 --restart=unless-stopped \  
7 --volume $(pwd)/srv-records.conf:/opt/unbound/etc/unbound/srv-records.  
  conf:ro \  
8 mvance/unbound:latest
```

Docker Compose

The following `docker-compose.yml` file is a starting point. The provided example shows how to override default forward and serve custom DNS records for your LAN. It requires `forward-records.conf` and `a-records.conf` files be provided at the `./my_conf/`.

```
1 version: '3'  
2 services:  
3   unbound:  
4     container_name: unbound  
5     image: "mvance/unbound:latest"  
6     expose:  
7       - "53"  
8     networks:  
9       - dns  
10    ports:  
11      - "53:53/tcp"  
12      - "53:53/udp"  
13    volumes:  
14      - "/data/unbound/my_conf/forward-records.conf:/opt/unbound/etc/  
    unbound/forward-records.conf"  
15      - "/data/unbound/my_conf/a-records.conf:/opt/unbound/etc/unbound/  
    a-records.conf"  
16    restart: unless-stopped  
17    networks:  
18      dns:
```

If you would rather provide a fully custom `unbound.conf` file, you will need to provide an `unbound.conf` file and mount it as a volume:

```
1     volumes:
2       - type: bind
3         read_only: true
4         source: ./my_conf/unbound.conf
5         target: /opt/unbound/etc/unbound/unbound.conf
```

Kubernetes usage

The method described here is basic and not recommended for larger environments. While this example is provided, support for Kubernetes related issues is outside the scope of this project.

To spin the deployment up use:

```
1 kubectl apply -f unbound-main-conf.yml -f other-files.yml ...
```

When taking it down, remember to use the reverse order in which you spun the deployment up.

Restarting:

```
1 kubectl rollout restart deployment dns
```

An example deployment can be viewed [here](#). It is not ready since you need to fill it with your records and the main unbound configuration file.

A fair warning: The example is not using a Service but a hostPort, thus this is only a mock-up. One should not use hostPort in a production cluster.

Additional warning: As per this document the default secrets configuration is unencrypted per default. You are responsible to harden this yourself and should do so!

Notes

Recursive config

The default config forwards forwards DNS queries to another DNS server over TLS. If you would rather this work as a recursive DNS server, you must use a customized Unbound configuration. An example unbound.conf file to configure unbound as a recursive server is available as a [guide](#).

Performance

For a DNS server with lots of short-lived connections, you may wish to consider adding `--net=host` to the run command for performance reasons. However, it is not required and some shared container hosting services may not allow it. You should also be aware that using `--net=host` can be a security risk in some situations. The Center for Internet Security Docker 1.6 Benchmark recommends against this mode since it essentially tells Docker to not containerize the container's networking, thereby giving it full access to the host machine's network interfaces. It also mentions this option could cause the container to do unexpected things such as shutting down the Docker host as referenced in Docker Issue #6401. For the most secure deployment, unrelated services with confidential data should not be run on the same host or VPS. In such cases, using `--net=host` should have limited impact on security.

Logging

Logging is very limited in the default config created by `unbound.sh`. If using the default config as an example starting point, a placeholder for a logfile (`unbound.log`) has been provided with the correct file ownership at the path `/opt/unbound/etc/unbound/` in case you want to increase logging and send to a file.

Healthcheck

By default, this image includes a healthcheck that performs a query for `cloudflare.com` on localhost at a regular interval.

To disable the healthcheck, add the `--no-healthcheck` flag to your Dockerfile. If using `docker-compose`, you can configure the healthcheck differently as explained in the Docker docs.

Known issues

The following message may appear in the logs about IPv6 Address Assignment:

```
[1644625926] libunbound[24:0] error: udp connect failed: Cannot  
assign requested address for 2001:xxx:xx::x port 53
```

While annoying, the container works despite the error. Search this issues in this repo for “udp connect” to see more discussion.

User feedback

Documentation

Documentation for this image is stored right here in the [README .md](#).

Documentation for Unbound is available on the project's website.

Issues

If you have any problems with or questions about this image, please contact me through a GitHub issue.

Contributing

You are invited to contribute new features, fixes, or updates, large or small. I imagine the upstream projects would be equally pleased to receive your contributions.

Please familiarize yourself with the repository's [README .md](#) file before attempting a pull request.

Before you start to code, I recommend discussing your plans through a GitHub issue, especially for more ambitious contributions. This gives other contributors a chance to point you in the right direction, give you feedback on your design, and help you find out if someone else is working on the same thing.

Acknowledgments

The code in this image is heavily influenced by DNSCrypt server Docker image, though the upstream projects most certainly also deserve credit for making this all possible. - Docker - DNSCrypt server Docker image - OpenSSL - Unbound

Licenses

License

Unless otherwise specified, all code is released under the MIT License (MIT). See the repository's [LICENSE](#) file for details.

Licenses for other components

- Docker: Apache 2.0
- DNSCrypt server Docker image: ISC License
- LibreSSL: Various
- OpenSSL: Apache-style license
- Unbound: BSD License