Tor Project | Set up Your Onion Service

7-9 minutes

This guide shows you how to set up an Onion Service for your website. For the technical details of how the Onion Service protocol works, see our Onion Service protocol page.

Step 0: Get a working Tor

As part of this guide, we will assume you have a functional Tor in your machine. To set up Tor, please follow the Tor installation guide. Tor should be up and running correctly for this guide to work. You should also know where Tor's configuration files are.

Step 1: Get a web server working

As a first step, you should set up a web server locally, like Nginx, Apache, or your favorite web server. Setting up a web server can be complex. If you get stuck or want to do more, find a friend who can help you or join our tor-onions mailing list to speak with other operators.

As an example, we will cover how to set up an onionsite with Nginx and Apache on Debian. We recommend you install a new separate web server for your Onion Service, since even if you already have one installed, you may be using it (or want to use it later) for a regular website.

On this page, the commands to manage the web server are based on Debian-like operating systems and may differ from other systems. Check your web server and operating system documentation.

Apache

Apache is available in the main repository of multiple Linux and *BSD distributions. To install `apache2` package:

```
$ sudo apt install apache2
```

Nginx

Nginx is available in the main repository of multiple Linux and *BSD distributions. To install `nginx` package:

```
$ sudo apt install nginx
```

By default, the web server will be running on `localhost:80` at the end of the installation. If you get an error message, something has gone wrong and you cannot continue until you've figured out why this didn't work.

Once your web server is set up, make sure it works: open your browser and go to http://localhost/. Then try putting a file in the main HTML directory, and make sure it shows up when you access the site.

Step 2: Configure your Tor Onion Service

The next step is opening the config file of Tor (torrc) and doing the appropriate configurations to setup an Onion Service. Depending on your operating system and setup, your Tor configuration file can be at a different location or look different.

You will need to add the following two lines to your `torrc` file:

```
HiddenServiceDir /var/lib/tor/my_website/
HiddenServicePort 80 127.0.0.1:80
```

The `HiddenServiceDir` line specifies the directory which should contain information and cryptographic keys for your Onion Service. You will want to change the `HiddenServiceDir` line, so that it points to an actual directory that is readable/writeable by the user that will be running Tor.

The `HiddenServicePort` line specifies a *virtual port* (that is, the port that people visiting your Onion Service will be using), and in the above case it says that any traffic incoming to port 80 of your Onion Service should be redirected to `127.0.0.1:80` (which is where the web server from step 1 is listening).

Tip: A good practice to avoid leaking an Onion Service to a local network is to run Onion Services over Unix sockets instead of a TCP socket. You will need to add the following two lines to your `torrc` file:

```
HiddenServiceDir /var/lib/tor/my-website/
HiddenServicePort 80 unix:/var/run/tor/my-website.sock
```

Step 3: Restart Tor and check that it worked

Now save your `torrc` and restart Tor.

```
$ sudo systemctl restart tor
```

If Tor starts up again, great. Otherwise, something is wrong. First look at your logfiles for hints. It will print some warnings or error messages. That should give you an idea of what went wrong. Typically, there are typos in the `torrc` or wrong directory permissions (See the logging FAQ entry if you don't know how to enable or find your log file.)

When Tor starts, it will automatically create the `HiddenServiceDir` that you specified (if necessary). Make sure this is the case.

Step 4: Test that your Onion Service works

Now to get your Onion Service address, go to your `HiddenServiceDir` directory, and find a file named `hostname`. The `hostname` file in your Onion Service configuration directory contains the hostname for your new onion v3 service. The other files are your Onion Service keys, so it is imperative that these are kept private. If your keys leak, other people can impersonate your Onion Service, deeming it compromised, useless, and dangerous to visit.

Now you can connect to your Onion Service using Tor Browser, and you should get the HTML page you set up back in **Step 1**. If it doesn't work, look in your logs for some hints, and keep playing with it until it works.

It is important to note that an Onion Service configured like this will be readable by anybody who knows or discovers the address. You can make Onion Services require authentication, and only users with a private key will access the service. Read more about Client authorization documentation.

(Optional) Step 5: Running multiple Onion Services

If you want to forward multiple virtual ports for a single Onion Service, just add more `HiddenServicePort` lines. If you want to run multiple Onion Services from the same Tor client, just add another `HiddenServiceDir` line. All the following `HiddenServicePort` lines refer to this `HiddenServiceDir` line, until you add another `HiddenServiceDir` line:

```
HiddenServiceDir /var/lib/tor/onion_service/
HiddenServicePort 80 127.0.0.1:80

HiddenServiceDir /var/lib/tor/other_onion_service/
HiddenServicePort 6667 127.0.0.1:6667
HiddenServicePort 22 127.0.0.1:22
```

If you're running multiple onionsites on the same web server, remember to edit your web server virtual host file and add the onion address for each website. For example, in Nginx and using Tor with Unix sockets, the configuration would look like this:

```
server {
        listen unix:/var/run/tor/my-website.sock;
        server_name <your-onion-address>.onion;
        access_log /var/log/nginx/my-website.log;
        index index.html;
        root /path/to/htdocs;
}
```

Or in Apache with Tor service listening on port 80:

```
     <VirtualHost *:80>
       ServerName <your-onion-address.onion>
       DocumentRoot /path/to/htdocs
       ErrorLog ${APACHE_LOG_DIR}/my-website.log
     </VirtualHost>
```

Step 6: Security advice and more tips

The default version of Onion Services is version 3 and its address is 56 characters long, without the `http://` and `.onion` parts. Onion services version 2 is deprecated and is no longer supported since the 0.4.6.1-alpha Tor release, in 2021. Please read the blog post Onion Service version deprecation timeline for more information.

Some onionsite operators may not want to disclose their Onion Service location. Therefore, you need to configure your web server so it doesn't give away any information about you, your computer, or your location. That is not an easy task, and these resources will help on how to make this possible:

- Operational Security.
- Onion services best practices by Riseup Collective.
- OnionScan is a tool to check if your onionsite is leaking information that could compromise your anonymity like your server IP address.

Finally, if you plan to keep your service available for a long time, you might want to make a backup copy of the `private_key` file somewhere.

Now that you have an onionsite working, you may want to deploy Onion-Location, or use tools like Docker, Heroku, Terraform, Ansible or stem to automate the management of your Onion Services. If you have a static website, but never installed Nginx or Apache, another project to try is OnionShare, where running an onionsite will be easier: guided with a graphic interface and with minimal configuration.