

Chhoto URL

- [Install Chhoto URL Shortener](#)

Install Chhoto URL Shortener

<https://www.youtube.com/embed/ZUOmlHRaBow>

Setup

Chhoto URL Shortener is a really lightweight, fast, and pretty simple to setup URL shortener. The developer says, on the GitHub README, that 'Chhoto' means 'small', and thus 'Chhoto URL' is 'Small URL'.

That said, it is a rust based web application that does exactly what it claims. It gives shortened URLs. You can run it as a service just for you, on your local network, or as a fully internet capable shortener. If you've ever tried Bit.ly, or any of the other shorteners out there, then you'll understand just what this does. It's a site to take a really long URL like "https://wiki.opensourceisawesome.com/install-chhoto-url-in-docker" and make it more like "https://short.url/rYubNk". These applications save you space and characters when sharing links through social media, chat, and dozens of other applications.

Setup a Server

In most self hosted applications these days, you'll need a machine to act as a server. This can be a machine in your home / business (such as an old laptop or desktop, a Raspberry PI or Single Board Computer (SBC), or even the computer you're reading this article from), a VM / Container hosted on one of your machines, or a VPS (Virtual Private Server) hosted by companies like RackNerd, Digital Ocean, Linode, Vultr, and so many more. Regardless of which option you choose, you'll want to do a few things to get the server setup properly.

Install updates to our Server

Ubuntu / Debian

```
sudo apt update && sudo apt upgrade -y
```

RedHat / CentOS / Fedora / Alma / Rocky

```
sudo dnf update -y
```

Add a non-root / sudo user on the server

Generally, when you setup a new server, the VPS (Virtual Private Server) service sets up a default "root" user for you. It's considered unsafe to do everything as "root", so let's setup a non-root user

who has super user (sudo) privileges.

`adduser <username>` You'll be prompted to enter and confirm a password for this user. You'll also be asked for some user information like Name, etc, but this is not required information. At the end, confirm the entries, and you'll have your new user.

Next, we need to add the user to the super user group.

Ubuntu / Debian

```
usermod -aG sudo <username>
```

RedHat / CentOS / Fedora / Alma / Rocky

```
usermod -aG wheel <username>
```

Now, you can log out of the system, and log back in as your new non-root super user.

Install Docker and Docker Compose

Fortunately, there is a single line command that will install both Docker and Docker Compose for us on most Linux based distributions.

We need the 'curl' utility to get this to work, so if you don't have it, you'll want to install it first with

Ubuntu / Debian

```
sudo apt install curl -y
```

RedHat / CentOS / Fedora / Alma / Rocky

```
sudo dnf install curl -y
```

Next, we'll run the command to install Docker and Docker Compose:

```
curl https://get.docker.com | sh
```

You may be prompted to enter your super user password, so be ready for it. Once you do, the install should proceed.

Once complete, we want to add our user to the 'docker' group so we can do `docker` and `docker compose` commands without having to type in `sudo` each time.

```
sudo usermod -aG docker <username>
```

Now we'll logout / exit the session, and log right back in so the updated group will take effect.

Install Chhoto URL

First, let's create a folder structure that will allow us to easily backup Chhoto URL and any other docker virtual machine applications we want to run. I like to do this by creating a parent 'docker' folder, then creating a folder inside that for each application or service I run.

```
mkdir -p docker/chhoto
```

Now we'll move into that directory and create our configuration file for the application (called a docker-compose file).

```
nano compose.yaml
```

Next, copy the code block below, and paste it into the editor.

```
---
services:
  chhoto-url:
    image: sintan1729/chhoto-url:latest
    restart: unless-stopped
    container_name: chhoto-url
    tty: true
    ports:
      - <4567>:4567
    environment:
      - db_url=/db/urls.sqlite
      - site_url=<https://short.url>
      # - hash_algorithm=Argon2
      # - port=4567
      - password=<ASup3rS3cre7Pa5sw0rdForCh0ot0Ur1>
      # - api_key=SECURE_API_KEY # if you intend to use the API, uncomment this line, and
create a secure key
      # - redirect_method=TEMPORARY
      - slug_style=UID # Pair or UID
      - slug_length=6 # default is 8
      - try_longer_slug=True # default is False
      - allow_capital_letters=True # default is False
      # - public_mode=Disable # set Enable if you want anyone to be able to use your
site
      # - public_mode_expiry_delay=3600 # uncomment if you want to auto expire URLs in 1
hour on your public site
```

```
# - disable_frontend=False
# - custom_landing_directory=/custom/dir/location
# - cache_control_header=no-cache, private
volumes:
- ./data/db:/db
```

Once copied, note the lines with the less than '<' and greater than '>' surrounding any values. These are the values you'll need to change to match your needs and setup. Let's go through them.

First, the port mapping: You'll want to remove the less than and greater than around the left side port. This port is probably fine, but I like to use higher ports than this, so I made mine 34567 on the left. Don't change the port on the right.

Next, the URL you want the shortener to use when creating short URLs.

1. you need to own the domain (e.g. example.com) and have the ability to set it's DNS to point to either the public IP of the server where you're running Chhoto URL, or to the Reverse Proxy you'll be using for Chhoto URL.
2. You need to setup your Reverse Proxy after you start the containers, or you may get certificate errors.

change the value

```
- site_url=<https://short.url>
```

to use the proper url you want, e.g.

```
- site_url=https://small.io <-- remember you need to own small.io.
```

Finally, a password with which to access your Chhoto site. You can use any long, strong password here, so make it a good one, and store it in a password manager.

```
- password=<ASup3rS3cre7Pa5sw0rdForCh0ot0Ur1>
```

There are other settings, you may want to change. Check the comments (anything to the right of a #), and determine if you want to change any values.

Use CTRL + O to save your changes, press Enter to confirm, and exit the nano editor with CTRL + X.

Let's pull down the Chhoto image with

```
docker compose pull
```

If you get no errors, you're on the right track. Now let's start our instance with

```
docker compose up -d && docker compose logs -f
```

Once the application comes up and begins to run you'll see log output in the terminal. It should be fairly brief, with no errors. You can leave this running. If you are not using a reverse proxy, you may need to get some valid certificates for your site. This container **does not** utilize certbot for LetsEncrypt as far as I can tell.

I will be using a reverse proxy, so let's go through the basic setup of it.

NOTE: You can use a reverse proxy even if your server has its own public IP address.

Reverse Proxy

Most reverse proxies work pretty much the same, though configuring them may be slightly different. Some have a web user interface (I prefer these), and others are containers all their own that you run alongside your application. Regardless of what you use, make sure to setup the following items:

1. The domain name for your shortener (e.g. small.io)
2. The IP address relative to your reverse proxy for the server where your shortener runs. I'm running mine inside a private network on a separate server so I'll use the private IP address of the server. If you run your proxy on the same machine, you **may** be able to use `localhost` as the address, but it will depend on your setup.
3. The port you setup in the compose.yaml file for your application. Recall I changed mine from 4567 to 34567, so I'll set 34567 in my reverse proxy.
4. Set your proxy to request valid SSL certs (generally these are from LetsEncrypt, but you may be advanced and creating your own).
5. Save your proxy, and test it out.

Testing Our Shortener

If all has gone well, you'll be greeted with the Web UI slightly blurred, and a pop-up box center screen requesting your password. You will want to grab that long, strong password you created in your compose.yaml file, and put it in here, then login.

Now all you need to do is enter a URL you want to make shorter. If you have done well, when you shorten it, you'll be able to click on it, and be directed to the proper page.

Well done!

Making Your Shortener Public

Chhoto URL gives you the power and choice to make your site public. If you choose to do this, please monitor the URLs and redirects people are putting in. As much as these tools are incredibly useful, they can also be used maliciously if not monitored well. The one thing I'd hate to happen to anyone, is they be held responsible for the actions of a dishonest person.