

Backup Software to Save your Data

- [URBackup](#)
 - [Installing and Configuring URBackup](#)
- [Duplicati](#)
 - [Installing Duplicati to Backup your Data](#)

URBackup

Installing and Configuring URBackup

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

Backups are important for so many reasons. I have had backups save my bacon more than once, and I have also learned the hard way how important backups really are. Data loss is no joke, and in this day and age there is truly no reason to have data loss.

Today, I want to cover a really great backup tool that is open source, and not too difficult to get setup. URBackup (like You Are Back Up! - get it?) is a really terrific tool boasting both a Server and Client installation that then makes your backup server (where you store your data) talk to your client machines (where you are backing up your data from) with some nice auto-discovery features built in.

Now, learn from me. Make sure your firewall has port 55414 and 55415 open so that URBackup's client and server can talk easily. You'll save yourself a lot of time and headache.

What you'll need

- A server with enough drive space to store backups.
- Docker and Docker Compose installed on that server - check out https://github.com/bmcgonag/docker_installs for some easy to use scripts for various versions of Ubuntu, Debian, and CentOS.
- At least one client machine running either Windows, Linux, or both if you like.
- About an hour of time.

Installation

First, if you haven't already done so, please make sure you have both Docker-CE and Docker-Compose installed on your intended server system. This is important as it's how we will install the

server portion of URBackup.

Installing Docker-CE and Docker-Compose

Installation via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

```
chmod +x ./install_docker_nproxyman.sh
```

and then run the script with the command:

```
./install_docker_nproxyman.sh
```

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of `sudo` in front of them.

Installing URBackup Server

Now, we'll grab the docker-compose text from the URoni/URBackup dockerhub page, and setup our space for backups.

1. Setup a space to store your backups.

```
mkdir -p urbackup/{data,storage}
```

This will setup a folder structure with a new folder called "urbackup", and two folders inside of that folder called "data" and "storage".

2. Move into the "urbackup" folder:

```
cd urbackup
```

3. Create a new docker-compose.yml file:

```
nano docker-compose.yml
```

4. Paste the following code into that file:

```
version: '2'

services:
  urbackup:
    image: uroni/urbackup-server:latest
    container_name: urbackup
    restart: unless-stopped
    environment:
      - PUID=1000 # Enter the UID of the user who should own the files here
      - PGID=1000 # Enter the GID of the user who should own the files here
      - TZ=America/Chicago # Enter your timezone
    volumes:
      - ./data:/var/urbackup
      - ./storage:/backups
      # Uncomment the next line if you want to bind-mount the www-folder
      #- /path/to/wwwfolder:/usr/share/urbackup
    network_mode: "host"
    # Activate the following two lines for BTRFS support
    cap_add:
      - SYS_ADMIN
```

Save the file contents with CTRL + O, then press Enter, and exit the nano editor with CTRL + X.

5. Now run the URBackup docker-compose with:

```
docker-compose up -d
```

Once it downloads URBackup, and says "done", give it about 30 seconds, then open a web browser, and go to the ip address of your server on port 55414.

For instance, my server ip is 192.168.7.51, so I went to

```
http://192.168.7.51:55414
```

If all went well, you should now see the URBackup Server Web UI in front of you.

Install a URBackup Client

If you are using Windows, you'll download the client application .exe file from the URBackup site (<https://urbackup.com>). Next install the client normally, and when done, you'll use the Graphical User Interface to decide whether you want to backup only certain folders, or the entire system, etc.

You can go back to the URBackup Server, and refresh the page to see if the Windows machine has been autodiscovered. If not, you might wait a couple of minutes and try again. If it still doesn't show up, then make sure you don't have a firewall on that's blocking ports 55414 and 55415. These are the ports that URBackup uses to communicate between server and client.

Linux Client

If you are running Linux, you'll want to grab the command line command from the URBackup site, and paste it into the terminal.

At the time of writing the command is:

```
TF=$(mktemp) && wget "https://hndl.urbackup.org/Client/2.4.11/UrBackup Client Linux 2.4.11.sh" -O $TF && sudo sh $TF; rm -f $TF
```

But, it's always best to go to the official source, and make sure the command is up to date.

Once pasted into the terminal, run it by pressing Enter, and enter your sudo password when prompted.

You may also have to enter 'Y' at some point, and choose how you want backups / snapshots to be done.

After everything runs, check the Web Interface, and make sure your machine shows up.

Now, you can enter a command to tell URBackup what to backup.

I chose only my Downloads folder, but give it whatever path you want.

```
sudo urbackupclientctl add-backupdir -d /home/brian/Downloads/
```

Run this as many times as you like to add multiple folders for backup.

Support My Efforts on Patreon

<https://www.patreon.com/bePatron?u=234177>

Duplicati

Installing Duplicati to Backup your Data

<https://www.youtube.com/embed/N1NRvg4KaDE>

Duplicati is a really amazing backup solution for your local machine. You can use Duplicati to backup locally, to storage on your own network, or to cloud hosted storage. The front end runs in a browser window, so it essentially just runs in the background. You setup your backup, schedule it, and forget about it. Just let it do it's job!

Today we'll be installing Duplicati in Docker. I'm using it to backup my Docker folder, but it can be mapped to the root level folder of your system, or to your home folder, or really in any way you see fit. Then you can use the browser interface to setup the backup(s) you want.

What You'll Need

- Docker-CE
- Docker-Compose
- A location to Backup to
- About 20 Minutes

Install Docker-CE and Docker-Compose

Installation via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://gitlab.com/bmcgonag/docker\_installs/-/raw/main/install\_docker\_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

```
chmod +x ./install_docker_nproxyman.sh
```

and then run the script with the command:

```
./install_docker_nproxyman.sh
```

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

Installation and Setup

Once you have Docker and Docker Compose installed, you'll want to create a folder inside your main organizational folder. I use a folder called "docker" to keep all of my docker application folders, volume mappings, docker-compose.yml and docker-run.txt files in. This way, I just backup the "docker" folder, and all of my docker compose, docker run commands, volume data, and various application configs are backed up with it. I highly recommend you do something similar.

So inside the "docker" folder, we'll create a new directory called "duplicati":

```
mkdir duplicati
```

Now we move into that directory:

```
cd duplicati
```

and we'll create a new file called "docker-compose.yml" inside this directory.

```
nano docker-compose.yml
```

You'll want to paste the following block of yaml text into that new file, and then we'll go through the spots where you'll need to make adjustments for your server.

```
version: "2.1"
services:
  duplicati:
```

```
image: lscr.io/linuxserver/duplicati
container_name: duplicati
environment:
  - PUID=0
  - PGID=0
  - TZ=<your/timezone>
volumes:
  - /mnt/filesync/duplicati/config:/config
  - /mnt/filesync/duplicati/backups:/backups
  - /home/brian/./source
ports:
  - 8270:8200
restart: unless-stopped
```

In the file above, I have several comments. You'll want to be sure and adjust the following items for your system:

- PGID - might be good to give this the root user's ID, 0, so there are no permissions issues during backup.
- PUID - might be good to give this the root group ID, 0, so there are no permissions issues during backup.
- TZ (time zone) - You'll want to make this your Time Zone. Mine is America/Chicago
- /mnt/filesync/duplicati/config:/config - use this mapping to map or create, then map a location for your backup config. Make sure this location exists before making the mapping. Only change the left side of the colon ":".
- /mnt/filesync/duplicati/backups:/backups - use this mapping to map or create, then map a location for your backup storage. Only change the left side of the colon ":".
- /<your root level folder to choose backup locations from>:/source - this is the mapping of the root level. Only change the left side of the colon ":".
- 8270:8200 - feel free to change the left side of the colon ":" to any open port on your host machine.

Once you've made all of the necessary changes, you'll want to save the file with CTRL + O, then Enter to confirm. Use CTRL + X to exit the nano editor.

Now run the command:

```
docker-compose up -d
```

To pull down the Duplicati image, and start it running.

Once it shows "done" in the terminal for each piece, you'll want to go to your web browser and enter the IP of your host machine, and the port you set on the left side of the colon in the port mapping.

I went to

`http://192.168.10.26:8270`

Your IP will likely be different.

Once you get to the Duplicati web interface, you can start setting up your backups, and adjusting settings available through the Web User Interface.

Make sure to check out the video for more details on using the Web UI to setup and schedule your backups.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

<https://www.patreon.com/bePatron?u=234177>