Server Management Tools

- Install Termix
- Install Nexterm

Install Termix

https://pt.opensourceisawesome.com/videos/embed/532sScp1rxURUcZKMUoYeo

Install Termix - Cross Platform Terminal Emulator

FIRST AND FOREMOST - in the video, I showed changing the permissions **should not** be set to 777, but instead make sure the 'data' folder owner is $USER:docker with: `chown -R $USER:docker ./data`.

Termix is a terminal emulator built to run as a desktop application, mobile application, and / or in your browser on Windows, MacOS, Linux, Android, and iOS / iPadOS.

It’s got a ton of great features like remembering connections, addition of ssh keys, grouping connections, snippets, and so much more. It’s powerful for a whole host of reasons, two of which are it’s incredible accessibility when you are essentially anywhere with an internet connection, and it’s open source!

Setup a Server

Install Updates

Update and upgrade your server's packages using the following commands:

- For Ubuntu/Debian: `sudo apt update && sudo apt upgrade -y`
- For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf update -y`

Create a Non-Root User

Create a non-root user with superuser (sudo) privileges:

1. Add a new user using `adduser <username>`
2. Set the password for this user.

3. Enter the relevant information (optional)
4. Enter 'Y', then press Enter.
5. Add the user to the "sudo" group:
    - For Ubuntu/Debian: `usermod -aG sudo <username>`
    - For RedHat/CentOS/Fedora/Alma/Rocky: `usermod -aG wheel <username>`

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

Install Docker and Docker Compose on your server:

1. Install the `curl` utility:
    - For Ubuntu/Debian: `sudo apt install curl -y`
    - For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf install curl -y`
2. Run the command to install Docker and Docker Compose:
   `curl https://get.docker.com | sh`

Add Your User to the Docker Group

Add your non-root user to the docker group so you can use Docker commands without sudo:

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

Now log out and back into your system, and you are ready to setup your application.

Install Termix with Docker and Docker Compose

First, let’s create our folder structure. We’ll create a parent ‘docker’ folder, with a ‘termix’ folder inside of it.

```
mkdir -p docker/termix
```

Next, we’ll move into our new `termix` folder with

`cd termix`

Now we need to create a configuration file called ‘compose.yaml’.

`nano compose.yaml`

Copy the following code block, and paste it into the new file we just created:

```
services:
  termix:
    image: ghcr.io/lukegus/termix:latest
    container_name: termix
    restart: unless-stopped
    ports:
      - "8080:8080"
    volumes:
      - ./termix-data:/app/data
    environment:
      PORT: "8080"
```

If your host machine (where you are running docker) happens to have port 8080 in use for another service already, then you can change the left side of the port mapping to something less common, such as 8022, or any other unused port number.

The port mapping above should then look like:

```
    ports:
      - "8022:8080"
```

Once, you have any changes needed, save the file with CTRL + O, then press Enter to confirm, and exit the nano text editor with CTRL + X.

Next, we’ll pull our images, and start our container application with the command:

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

The above is really two commands in one, first telling docker compose to pull the images and start the associated container(s), and the second telling docker compose to display the logs for the containers as the system is up and running.

You can stop viewing the logs with CTRL + C.

Once everything is up and running you can navigate to the IP of your host machine on port 8080 (or the port you set on the left side of the port mapping) to see the Web User Interface for Termix.

Here you can create your initial user, which will also be the administrative user of the site. You can add more users, as well as disable registration in the settings. There are also SSO login options for the browser version so you can have your team utilize your SSO IdP if desired.

Setup a Reverse Proxy

First and foremost, you need to own a domain name. I own the domain 'opensourceisawesome.com'. You need to have a domain you own, and can set DNS A and AAAA Recrods for.

You can get a cheap domain at places like NameCheap, or by going to Hover. I use hover, and I like their control panel for managing my domains. If you use my link to sign up, I get a credit, if you don't, then I don't. No big deal, and no pressure, just an option.

Next, you need to have some reverse proxy software setup. As sttated above, I've done videos on setting up NGinX Proxy Manager, and Pangolin, which both have clean user interfaces for setting up your reverse proxy.

Once you have your reverse proxy installed, you can add your subdomain to it.

For instance, you may want to use termix.mygreatdomain.com as your subdomain. In that case you need to have an A record in your domain's DNS settings that points the subdomain 'termix' to the public IP address of your server, or the private network in which your server is setup.

Then, in your reverse proxy, you'll add an entry for 'termix.mygreatdomain.com' and tell it to send the traffic along to the host server and port you setup Termix to run on.

The way this flows is as follows:

A user requests your Termix instance at https://termix.mygreatdomain.com → which asks DNS the IP for the site → which sends you to the public IP address of your host or private network → which forwards you to the reverse proxy → which sends you to the local or private address and port of your Termix instance.

Seems like a lot, but it works, it's reliable, and it's repeatable.

Once complete, you should now be able to get to your Termix instance via it's secure URL, and start accessing your servers and hosts from anywhere you have a browser and internet access.

Support this Channel and Content

Become a Patron at Patreon

Check me out on Patreon

Be me a Coffee or Beer at Paypal

Paypal Support for Awesome Open Source

Get the Awesome Open Source Merch

Great Merch for Open Source Advocates!

Install Nexterm

https://pt.opensourceisawesome.com/videos/embed/xhjJgPPWTnAATSHLe5y986

Install Nexterm - Browser based Server Access and Management

Nexterm is an amazing open source project designed to enable secure, powerful server and host management via the web browser. Tools like SSH, SFTP, RDP, VNC, and the use of snippets and code repositories make it a power house for managing your homelab or your business, clients, and cloud infrastructure with ease.

Setup a Server

Install Updates

Update and upgrade your server's packages using the following commands:

- For Ubuntu/Debian: `sudo apt update && sudo apt upgrade -y`
- For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf update -y`

Create a Non-Root User

Create a non-root user with superuser (sudo) privileges:

1. Add a new user using `adduser <username>`
2. Set the password for this user.
3. Enter the relevant information (optional)
4. Enter 'Y', then press Enter.
5. Add the user to the "sudo" group:
   - For Ubuntu/Debian: `usermod -aG sudo <username>`
   - For RedHat/CentOS/Fedora/Alma/Rocky: `usermod -aG wheel <username>`

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

Install Docker and Docker Compose on your server:

1. Install the `curl` utility:
   - For Ubuntu/Debian: `sudo apt install curl -y`
   - For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf install curl -y`
2. Run the command to install Docker and Docker Compose:
   `curl https://get.docker.com | sh`

Add Your User to the Docker Group

Add your non-root user to the docker group so you can use Docker commands without sudo:

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

Now log out and back into your system, and you are ready to setup your application.

Install Nexterm with Docker and Docker Compose

First, let’s create our folder structure. We’ll create a parent ‘docker’ folder, with a ‘termix’ folder inside of it.

```
mkdir -p docker/nexterm
```

Next, we’ll move into our new `nexterm` folder with

```
cd nexterm
```

Now we need to create a configuration file called ‘compose.yaml’.

```
nano compose.yaml
```

Copy the following code block, and paste it into the new file we just created:

```
services:
  nexterm:
    environment:
      ENCRYPTION_KEY: "< openssl rnad -hex 32 >"
    ports:
      - "6989:6989"
    restart: unless-stopped
    volumes:
      - ./nexterm:/app/data
    image: germannewsmaker/nexterm:latest
```

If you prefer not to use port 6989 as the port you access Nexterm through, you can change the left side of the port mapping to any port number which is not in use or reserved on your host system. For instance, if you wanted to use port 8022 as the port you access Nexterm through, then you would setup the port mapping as below.

The port mapping above should then look like:

```
    ports:
      - "8022:6989"
```

Once, you have any changes needed, save the file with CTRL + O, then press Enter to confirm, and exit the nano text editor with CTRL + X.

Next, we'll pull our images, and start our container application with the command:

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

The above is really two commands in one, first telling docker compose to pull the images and start the associated container(s), and the second telling docker compose to display the logs for the containers as the system is up and running.

You can stop viewing the logs with CTRL + C.

Once everything is up and running you can navigate to the IP of your host machine on port 6989 (or the port you set on the left side of the port mapping) to see the Web User Interface for Nexterm.

Here you can create your initial user, which will also be the administrative user of the site. You can add more users, as well as disable registration in the settings. There are also SSO login options for the browser version so you can have your team utilize your SSO IdP if desired.

Setup a Reverse Proxy

First and foremost, you need to own a domain name. I own the domain 'opensourceisawesome.com'. You need to have a domain you own, and can set DNS A and AAAA Recrods for.

You can get a cheap domain at places like NameCheap, or by going to Hover. I use hover, and I like their control panel for managing my domains. If you use my link to sign up, I get a credit, if you don't, then I don't. No big deal, and no pressure, just an option.

Next, you need to have some reverse proxy software setup. As stated above, I've done videos on setting up NGinX Proxy Manager, and Pangolin, which both have clean user interfaces for setting up your reverse proxy.

Once you have your reverse proxy installed, you can add your subdomain to it.

For instance, you may want to use `nexterm.mygreatdomain.com` as your subdomain. In that case you need to have an A record in your domain's DNS settings that points the subdomain 'nexterm' to the public IP address of your server, or the private network in which your server is setup.

Then, in your reverse proxy, you'll add an entry for 'nexterm.mygreatdomain.com' and tell it to send the traffic along to the host server and port you setup Nexterm to run on.

The way this flows is as follows:

A user requests your Nexterm instance at https://nexterm.mygreatdomain.com → which asks DNS the IP for the site → which sends you to the public IP address of your host or private network → which forwards you to the reverse proxy → which sends you to the local or private address and port of your Nexterm instance.

Seems like a lot, but it works, it's reliable, and it's repeatable.

Once complete, you should now be able to get to your Nexterm instance via it's secure URL, and start accessing your servers and hosts from anywhere you have a browser and internet access.

Support this Channel and Content

Become a Patron at Patreon

Check me out on Patreon

Be me a Coffee or Beer at Paypal

Paypal Support for Awesome Open Source

Get the Awesome Open Source Merch

Great Merch for Open Source Advocates!