

Lunalytics

- [Install Lunalytics Site Monitor](#)

Install Lunalytics Site Monitor

<https://www.youtube.com/embed/gVlq3C0gbo0>

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.

Step 2: 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>
```

Install Lunalytics in Docker

First we'll create a new folder to hold our applications and services files.

```
mkdir -p docker/lunalytics
```

Now we'll move into that folder.

```
cd docker/lunalytics
```

Now we'll make our docker compose configuration file, called 'compose.yaml':

```
nano compose.yaml
```

Copy the configuration file contents below, and paste them into your new file.

```
---
services:
  lunalytics:
    image: ksjaay/lunalytics:latest
    container_name: lunalytics
    ports:
      - '2308:2308'
    volumes:
      - <./path/to/data>:/app/data
      - <./path/to/logs>:/app/logs
```

In the above file, you'll want to replace `<./path/to/data>` and `<./path/to/logs>` with something a bit more sensible. I'll be using `./data` and `./logs`.

Now we'll pull down the image for Lunalytics with:

```
docker compose pull
```

Once it pulls and extracts, you'll be back at the terminal prompt. Now we'll start our application, and the first time we start it up, we'll show the log output as well. We can use a one liner,

concatenated command for this:

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

The first command, before the '&&', tells docker compose to bring up the containers from the image we just pulled down. The next part, after the '&&', tells docker compose to show us the live logs as everything is starting up. You can stop following the logs with the hotkey combination CTRL + c whenever you feel like it.

Setup your Reverse Proxy

For an application like Lunalytics, you really want to setup a reverse proxy or have this running on a VPS with a public IP address that you can point an A record at.

Each Reverse proxy is a little different. I have videos on NginX Proxy Manager and Pangolin. If you are interested in those having a Web GUI for easier setup, then definitely check those out.

I'll be using Pangolin for my reverse proxy, but feel free to use whichever if your favorite. The basic idea behind a reverse proxy is that you make a request to a domain or subdomain, that request is handled by DNS and you will be sent to the A / AAAA record associated to your reverse proxy. The reverse proxy then takes the request for that site name and checks to see if it has an entry for that site, and if so sends you along to it, if not, will display a 404 page or some other error page. Reverse proxies allow you to send all traffic to one place, and then deal with the requests as they head into your network for instance.

What do I need in my reverse proxy?

1. Site domain and subdomain (e.g. subdomain = luna, domain = mygreatdomain.com)
2. Private IP of the host machine where Lunalytics is running (e.g. 192.168.1.12)
3. Port on which Lunalytics is running on the host (left side in port mapping in compose.yaml file - 2308).

Once you've set those up, you can specify if you want SSL (hint: you do!), and even get your reverse proxy to make automated requests from LetsEncrypt to keep your certificates up to date.

Initial Login / Registration

To login, you'll want to go to your site and add the path `/setup` at the end. If you created 'luna.mygreatdomain.com'.

<https://luna.mygreatdomain.com/setup>

This will take you through a quick setup wizard, and once complete you'll be ready to start adding sites to check with the Lunalytics tools. Don't forget to setup a notification service such as Email, Slack, Apprise, HomeAssistant, etc. so you'll know when something may be wrong with your site.

Congratulations, you've setup a site monitor, and now you're ready to keep those sites and services up and running.

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!](#)