How to Install Uptime Kuma And MariaDB in Docker?

written by sysadmin | 7 January 2026

The previous article explained how to install the Uptime Kuma application using Docker. However, by default, Uptime Kuma uses a SQLite database, and you want to change the database to MariaDB for some reasons.

Problem

How to install Uptime Kuma and MariaDB in Docker?

Solution

Although SQLite serves as a superb embedded database option for numerous scenarios, certain inherent limitations render it inappropriate for particular applications. If your application has a large amount of traffic and uses a lot of write modes simultaneously, and the data growth is very fast, your application is not suitable for using a SQLite database. Likewise, with the Uptime Kuma application. If you monitor many hosts using low intervals, it will cause very fast data growth, so you have to think about another database solution besides the SQLite database.

1. Create a Docker Compose file

Create a compose folder in the **/opt** folder using the command below:

```
sudo mkdir -p /opt/compose/uptime-kuma/
cd /opt/compose/uptime-kuma/
```

After that, create **docker-compose.yaml** file and copy the script below:

```
services:
```

```
  mariadb:
    image: mariadb:11.4
    container_name: mariadb
    restart: unless-stopped
    environment:
      MARIADB_ROOT_PASSWORD: ${MARIADB_ROOT_PASSWORD}
      MARIADB_DATABASE: ${MARIADB_DATABASE}
      MARIADB_USER: ${MARIADB_USER}
      MARIADB_PASSWORD: ${MARIADB_PASSWORD}
    volumes:
      - mariadb-data:/var/lib/mysql
    networks:
      - kuma-net
    healthcheck:
      test: ["CMD", "healthcheck.sh", "--connect", "--innodb_initialized"]
      interval: 10s
      timeout: 5s
      retries: 5

  uptime-kuma:
    image: louislam/uptime-kuma:2
    container_name: uptime-kuma
    restart: unless-stopped
    depends_on:
      mariadb:
        condition: service_healthy
    ports:
      - "3001:3001"
    volumes:
      - kuma-data:/app/data
    networks:
      - kuma-net

volumes:
  mariadb-data:
  kuma-data:

networks:
  kuma-net:
```

After that, create a **.env** file like the below script (Adjust the value of this file to your liking):

```
MARIADB_DATABASE=kuma
MARIADB_USER=kuma-user
MARIADB_PASSWORD=123456
MARIADB_ROOT_PASSWORD=qwerty
```

Run the below command to turn on Docker Compose:

```
docker compose up -d
```

To check if the containers are running or not, use the command below:

```
docker ps
```

After you type the commands, your screen will show up similar to the one below:

Run the Docker Compose

2. Configure the web server

If you use Apache, create a file at **/etc/apache2/sites-available/kuma.conf** and copy the script below to the file:

```
<VirtualHost *:80>
ServerName yourdomain.com
DocumentRoot /var/www/html/

ProxyPass / http://localhost:3001/
RewriteEngine on
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteCond %{HTTP:Connection} upgrade [NC]
RewriteRule ^/?(.*) "ws://localhost:3001/$1" [P,L]

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Then run the command below:

```
sudo a2enmod rewrite
```

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2ensite kuma.conf
```

Check if there is an error in Apache, and if there is no error, reload Apache using the command below:

```
apachectl -t
sudo systemctl reload apache2
```

INFO

If your server is running an nginx webserver, then in the file **/etc/nginx/conf.d/uptime-kuma.conf** insert the script below:

```
server {
    listen 80;
    server_name uptime-kuma.yourdomainname.com;

    location / {
        proxy_pass         http://localhost:3001;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection "upgrade";
        proxy_set_header   Host $host;
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Proto $scheme;

        # Added WebSocket support
        proxy_set_header   Sec-WebSocket-Key $http_sec_websocket_key;
        proxy_set_header   Sec-WebSocket-Version $http_sec_websocket_version;
        proxy_set_header   Sec-WebSocket-Extensions
$http_sec_websocket_extensions;

        # Improve performance of this reverse proxy
        proxy_buffering    off;
    }

    # Redirect HTTP to HTTPS if needed for encryption
    # Uncomment the following lines if you have SSL enabled
    # return 301 https://$host$request_uri;
}
```

Use the command below to check if there is an error in the

nginx configuration and then reload nginx:

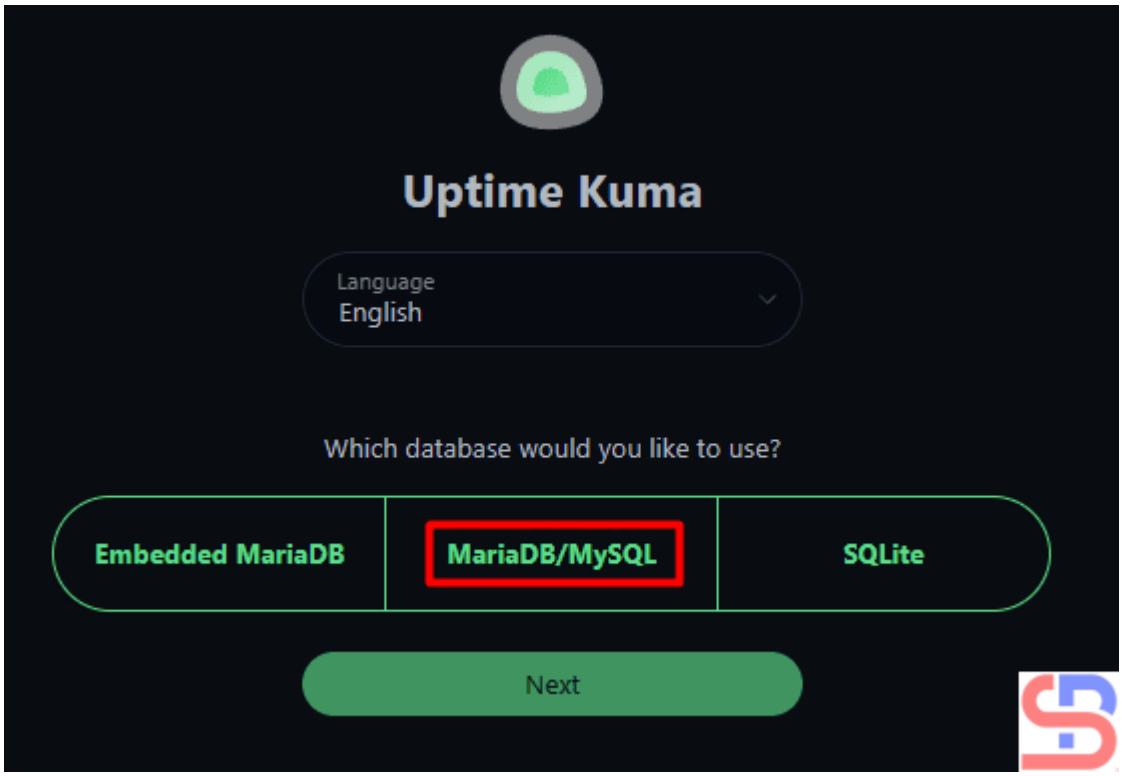
```
sudo nginx -t
sudo systemctl reload nginx
```

3. Access uptime kuma

Open your browser, and type:

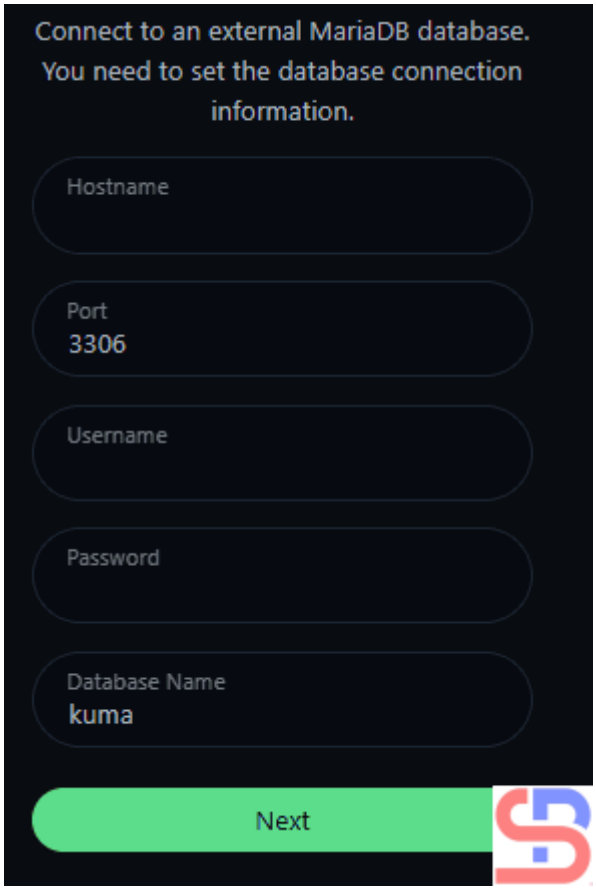
```
http://ip_server:3001
```

Then there will be a display like below:



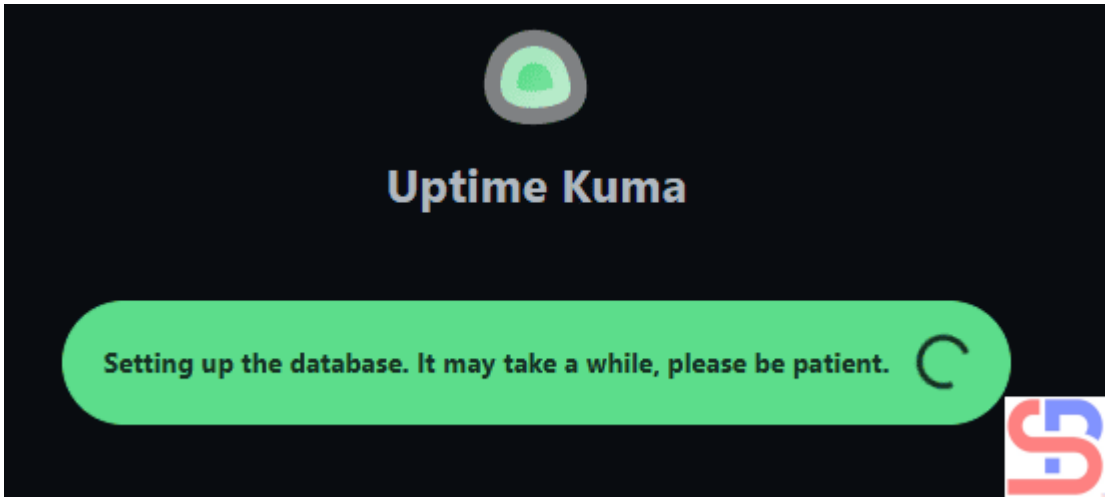
Choose MariaDB/MySQL

Click **MariaDB/MySQL**, and your screen will appear similar to the picture below:



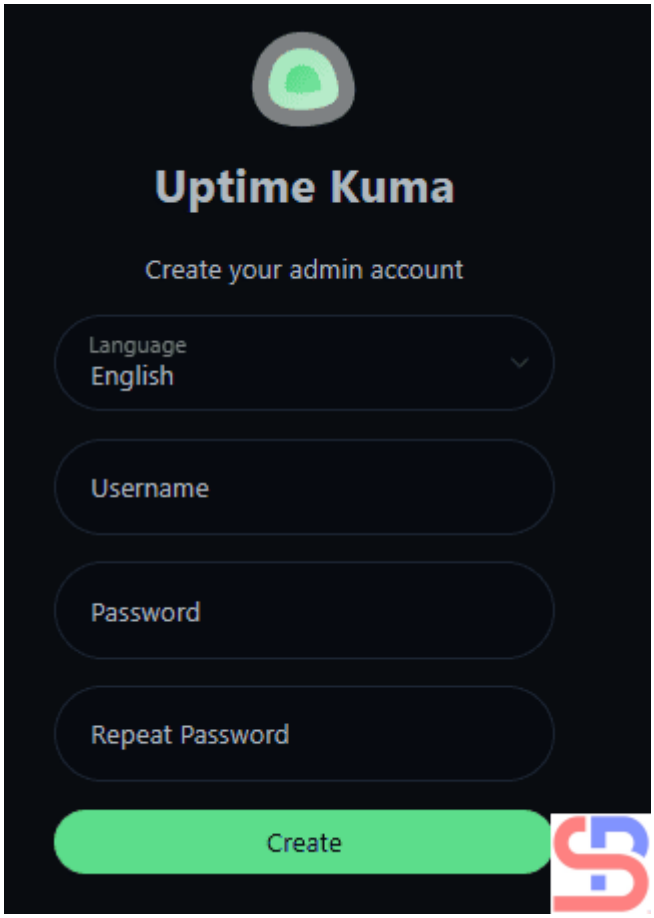
Fill in the columns for the database

Enter in the columns above the values □□that correspond to the **.env** file. Click the **Next** button, and your screen will show up similar to the one below:



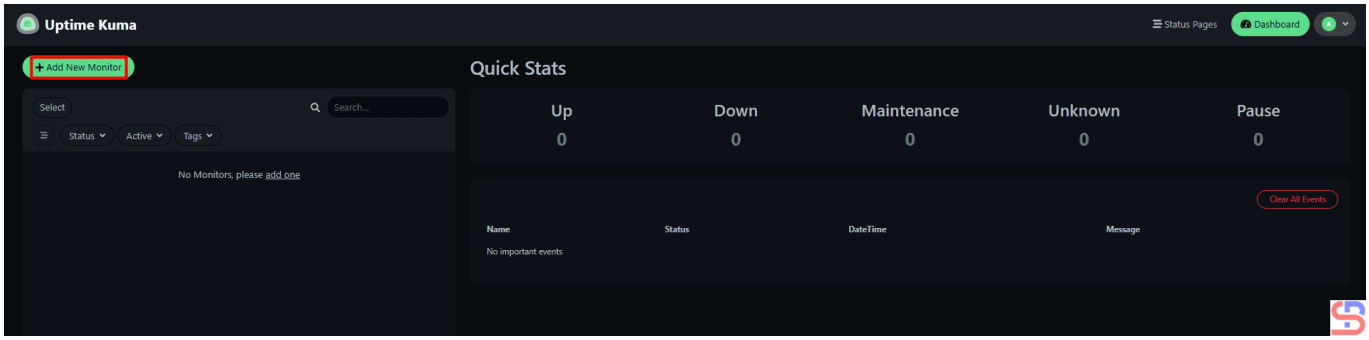
Setting up the database

You have to wait until finishes, and after that, your screen will appear similar to the image shown below:



Fill in the columns for the admin account

Enter in the columns above the value you want and press the **Create** button, then a display will appear similar to the image provided below:



Display of uptime kuma

If you want to make sure Uptime Kuma uses a MariaDB database, run the command below:

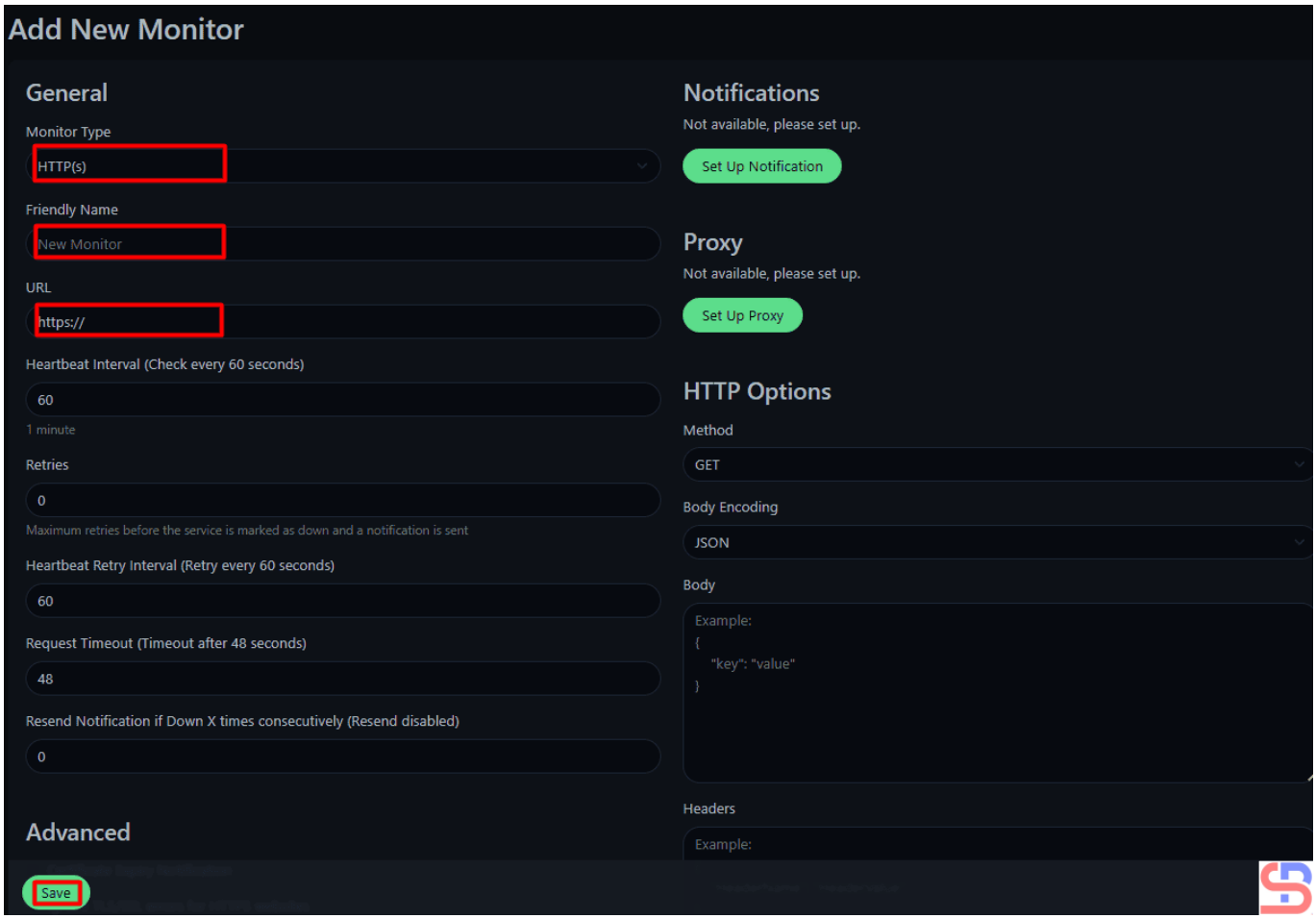
```
docker logs uptime-kuma | grep DB
```

Your screen will appear similar to the picture shown below:

```
sysadmin@docker:~$ docker logs uptime-kuma
Welcome to Uptime Kuma
Your Node.js version: 20.19.5
2025-12-26T03:25:18Z [SERVER] INFO: Env: production
2025-12-26T03:25:28Z [SERVER] INFO: Uptime Kuma Version: 2.0.2
2025-12-26T03:25:28Z [SERVER] INFO: Loading modules
2025-12-26T03:25:36Z [SERVER] INFO: Creating express and socket.io instance
2025-12-26T03:25:36Z [SERVER] INFO: Server Type: HTTP
2025-12-26T03:25:37Z [SERVER] INFO: Data Dir: ./data/
2025-12-26T03:25:37Z [SETUP-DATABASE] INFO: db-config.json is not found or invalid: ENOENT: no such file or directory, open 'data/db-config.json'
2025-12-26T03:25:38Z [SETUP-DATABASE] INFO: Starting Setup Database on 3001
2025-12-26T03:25:38Z [SETUP-DATABASE] INFO: Open http://localhost:3001 in your browser
2025-12-26T03:25:38Z [SETUP-DATABASE] INFO: Waiting for user action...
Request /setup-database-info
2025-12-26T03:32:22Z [SETUP-DATABASE] INFO: Testing database connection...
2025-12-26T03:32:22Z [SETUP-DATABASE] INFO: Database is configured, close the setup-database server and start the main server now.
2025-12-26T03:32:22Z [SETUP-DATABASE] INFO: The setup-database server is closed
2025-12-26T03:32:22Z [DB] INFO: Database Type: mariadb
2025-12-26T03:32:23Z [MARIADB] INFO: Creating basic tables for MariaDB
2025-12-26T03:32:26Z [MARIADB] INFO: Created basic tables for MariaDB
2025-12-26T03:32:26Z [SERVER] INFO: Connected to the database
2025-12-26T03:32:29Z [MIGRATION] INFO: Migration server is running on http://localhost:3001
2025-12-26T03:32:29Z [DB] INFO: Migrating Aggregate Table
2025-12-26T03:32:29Z [DB] INFO: Getting list of unique monitors
2025-12-26T03:32:29Z [DB] INFO: Clearing non-important heartbeats
2025-12-26T03:32:29Z [DB] INFO: No data to migrate
```

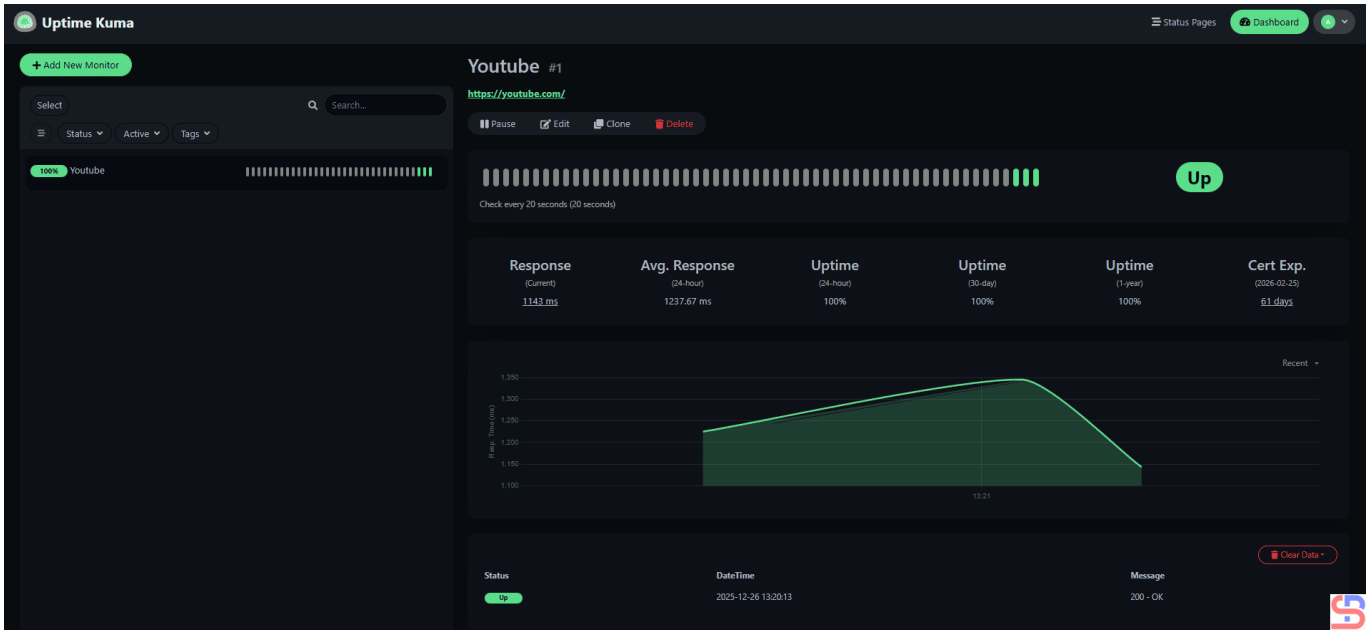
Check the running database

If you want to monitor the website, click the **Add New Monitor** button at the top left of the site, and an image similar to the one shown will appear:



Create a new host or a website to monitor in Uptime Kuma

Fill in the required fields (at least fill in the **Monitor Type**, **Friendly Name**, and **URL** columns) and press the **Save** button, then the host you have filled in will look like in the image below:

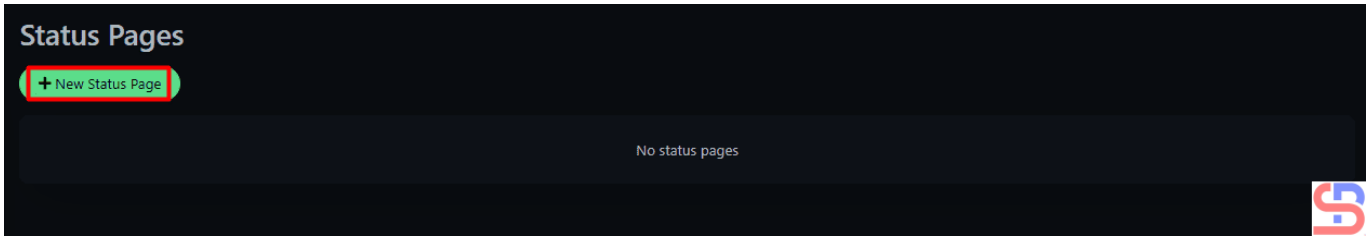


Monitor the host or the website

If you just want to display the status without displaying many attributes, then you can click the **Status Pages** button at the top right of the site, like the image below:

Click the Status Page button

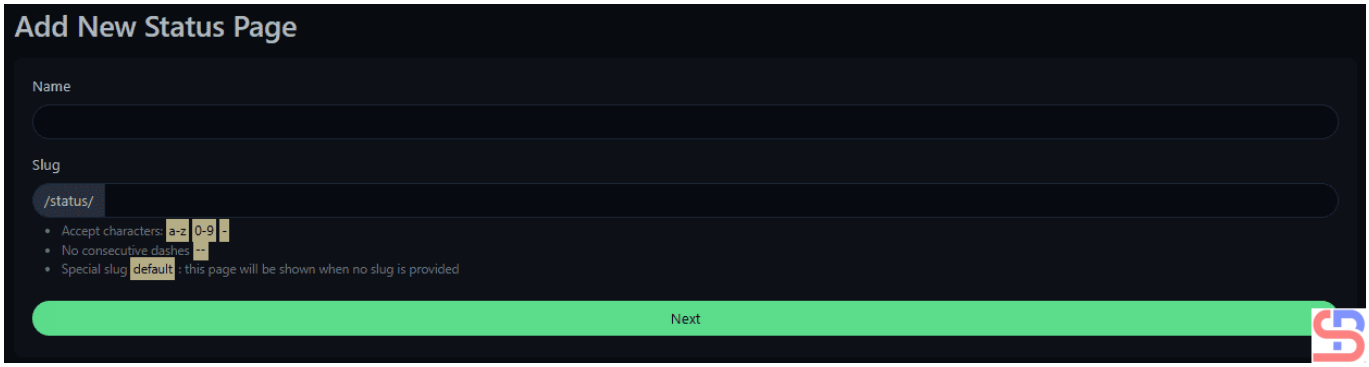
After you press the Status Page button, the following image will appear:



Create the Status Page page

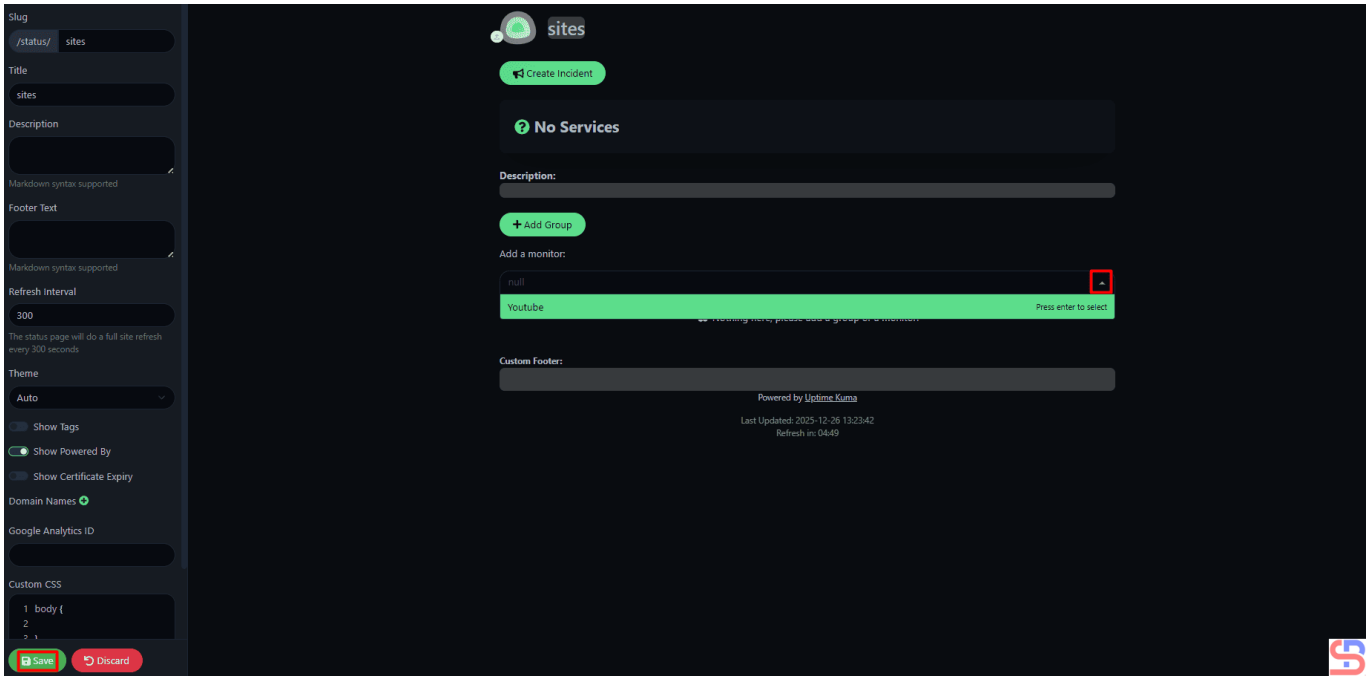
Click the **New Status Page** button, and an image will appear

similar to the one shown below:



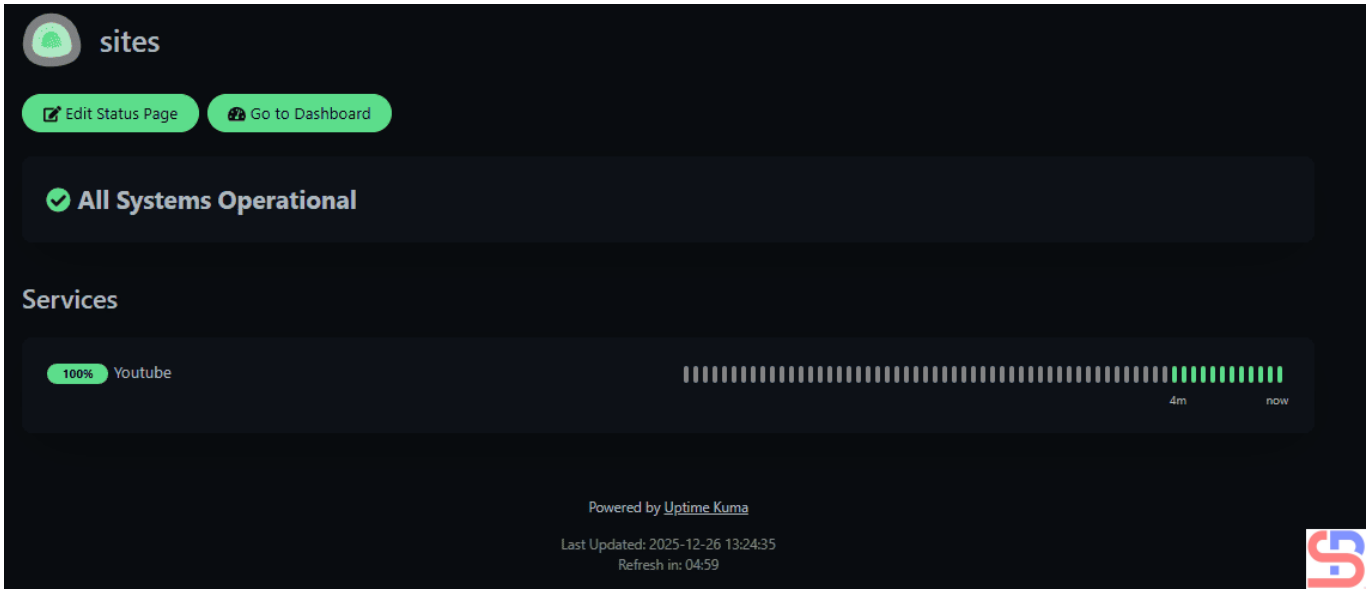
Create the Status Page page

Enter the name and slug you want (I wrote the sites for the name and slug), then press the Next button, and then there will be a display as below:



Insert the host or the monitor in the Status Page

Enter the host you want to display on the Status Page, after that, click the Save button, then there will be a display as below:



Display of Status Page

You can see that the hosts to be monitored look simpler, and you can give the URL to other parties to also monitor these hosts.

Note

If you want to back up the MariaDB database running on Docker and learn how to restore the database, you can go to this page.

References

quora.com
magnus919.com
uptimekuma.org