

[How to Install Uptime Kuma Application on Ubuntu?](#)

written by sysadmin | 14 January 2026

The previous articles explained how to install the uptime kuma application on Docker, either [using the SQLite database](#) or [using the MariaDB database on Docker](#) or [using the MariaDB database on the host](#). This article will explain how to install the uptime kuma application without using Docker but using packages.

Problem

How to install uptime kuma application on Ubuntu?

Solution

Here are the steps to install uptime kuma application on Ubuntu:

1. Install the packages

Run the commands below to install the required packages:

```
sudo apt update -y  
sudo apt install nginx mariadb-server git -y
```

Then, install nodejs using the command below:

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash - && sudo  
apt install -y nodejs
```

After that, download the uptime kuma application by running the command below:

```
git clone https://github.com/louislam/uptime-kuma.git  
cd uptime-kuma/
```

Next, copy the commands below to install the uptime kuma application:

```
sudo npm run setup
sudo npm install pm2 -g
sudo pm2 install pm2-logrotate
sudo pm2 start server/server.js --name uptime-kuma
sudo pm2 startup
```

2. Configure MariaDB

Access MariaDB and run the queries below:

Akses ke MariaDB dan jalankan query-query di Bawah ini:

```
CREATE DATABASE uptime_kuma;
CREATE USER 'kuma-user'@'%' IDENTIFIED BY 'kumapass123';
GRANT ALL PRIVILEGES ON uptime_kuma.* TO 'kuma-user'@'%';
FLUSH PRIVILEGES;
\q
```

3. Configure 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:

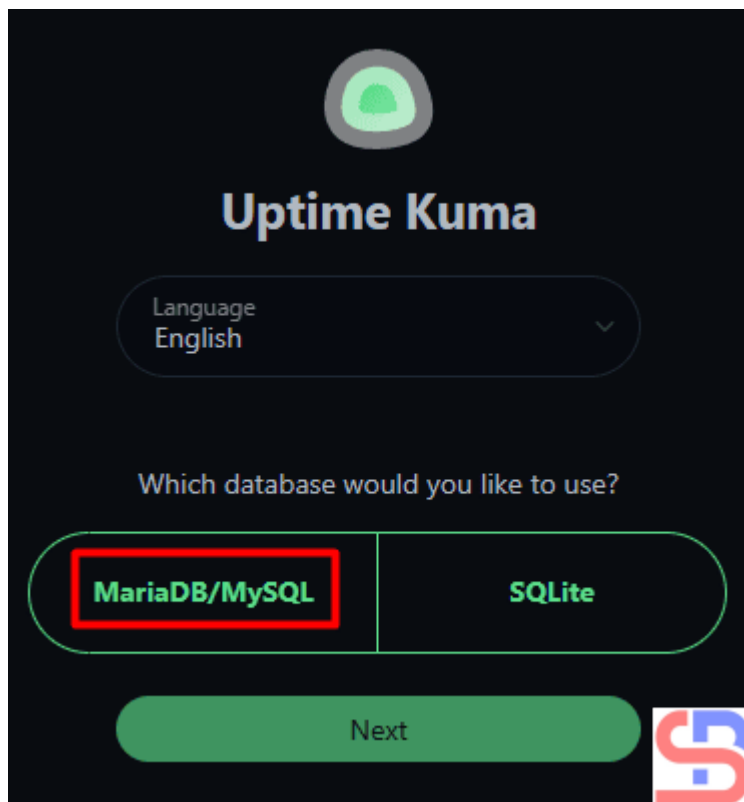
```
sudo nginx -t  
sudo systemctl reload nginx
```

4. Access uptime kuma

Open your browser, and type:

`http://ip_server:3001`

then there will be a display like below:



Click the MariaDB/MySQL button

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

Connect to an external MariaDB database.
You need to set the database connection information.

Hostname


Port
3306

Username

Password

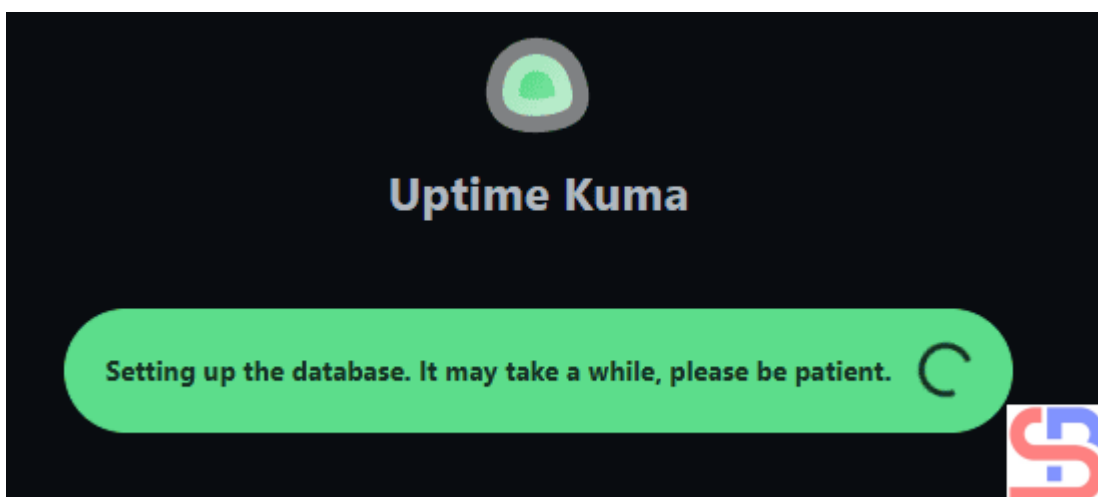
Database Name
kuma

Next



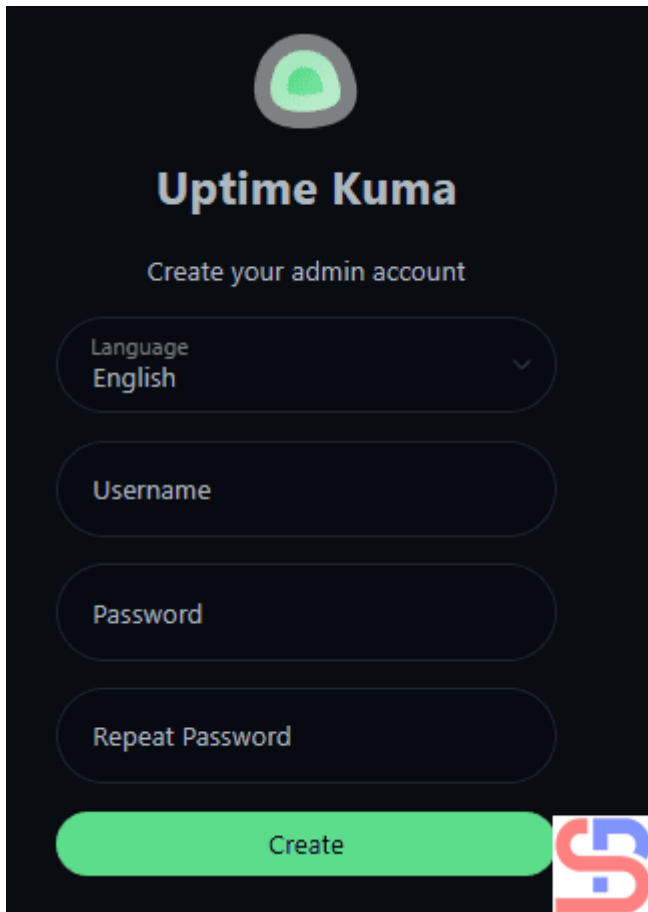
Fill in the columns for the database

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



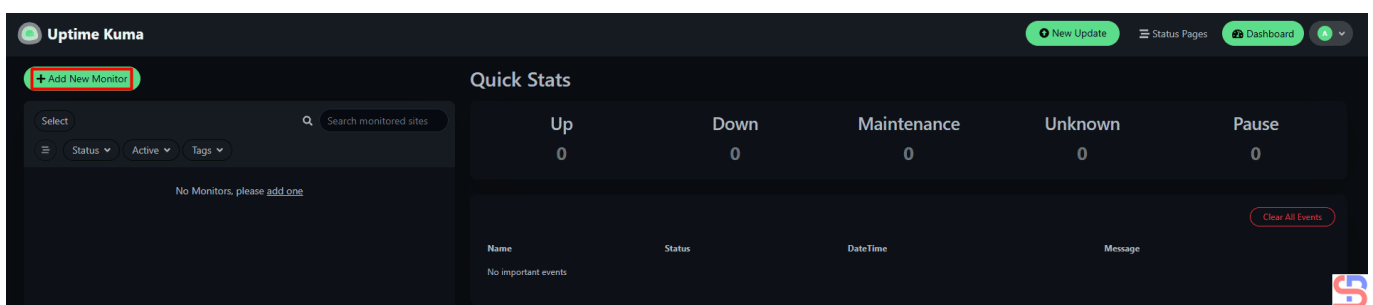
Setting up the database

You have to wait until finish, 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 application

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

Add New Monitor

General

Monitor Type:

Friendly Name:

URL:

Heartbeat Interval (Check every 60 seconds): 1 minute

Retries: Maximum retries before the service is marked as down and a notification is sent

Heartbeat Retry Interval (Retry every 60 seconds):

Request Timeout (Timeout after 48 seconds):

Resend Notification if Down X times consecutively (Resend disabled):

Save

Notifications

Not available, please set up. **Set Up Notification**

Proxy

Not available, please set up. **Set Up Proxy**

HTTP Options

Method:

Body Encoding:

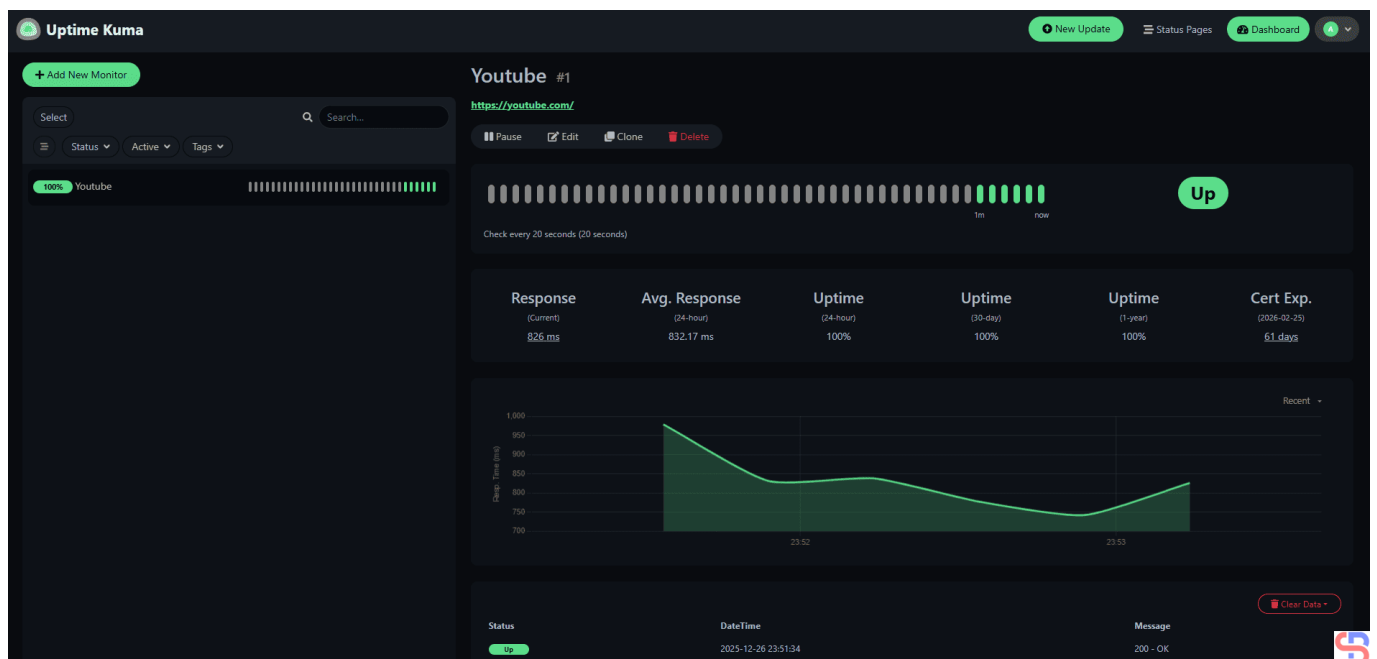
Body:

```
Example:
{
  "key": "value"
}
```

Headers:

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:



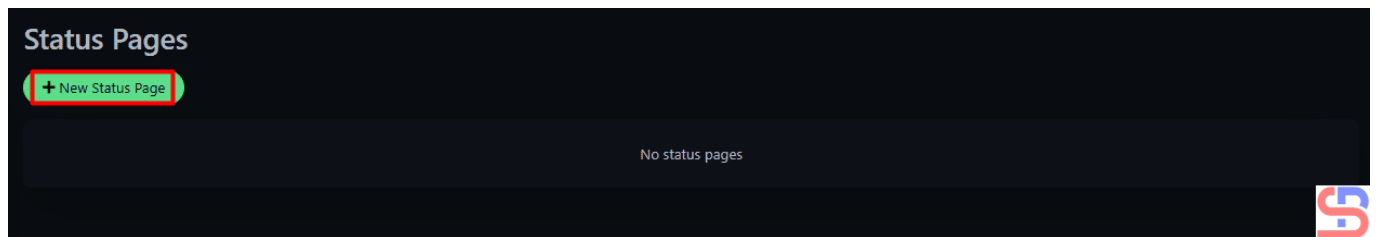
Monitor the host

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:



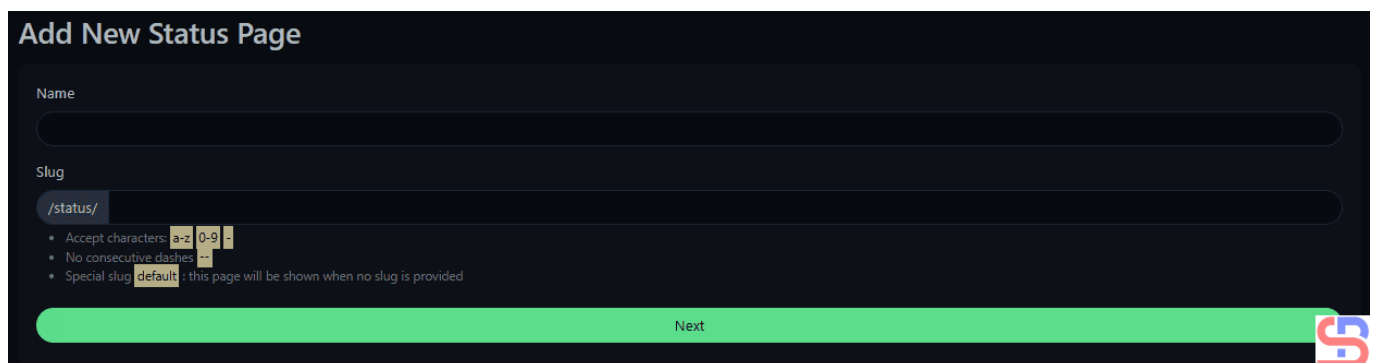
Click the Status Pages button

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



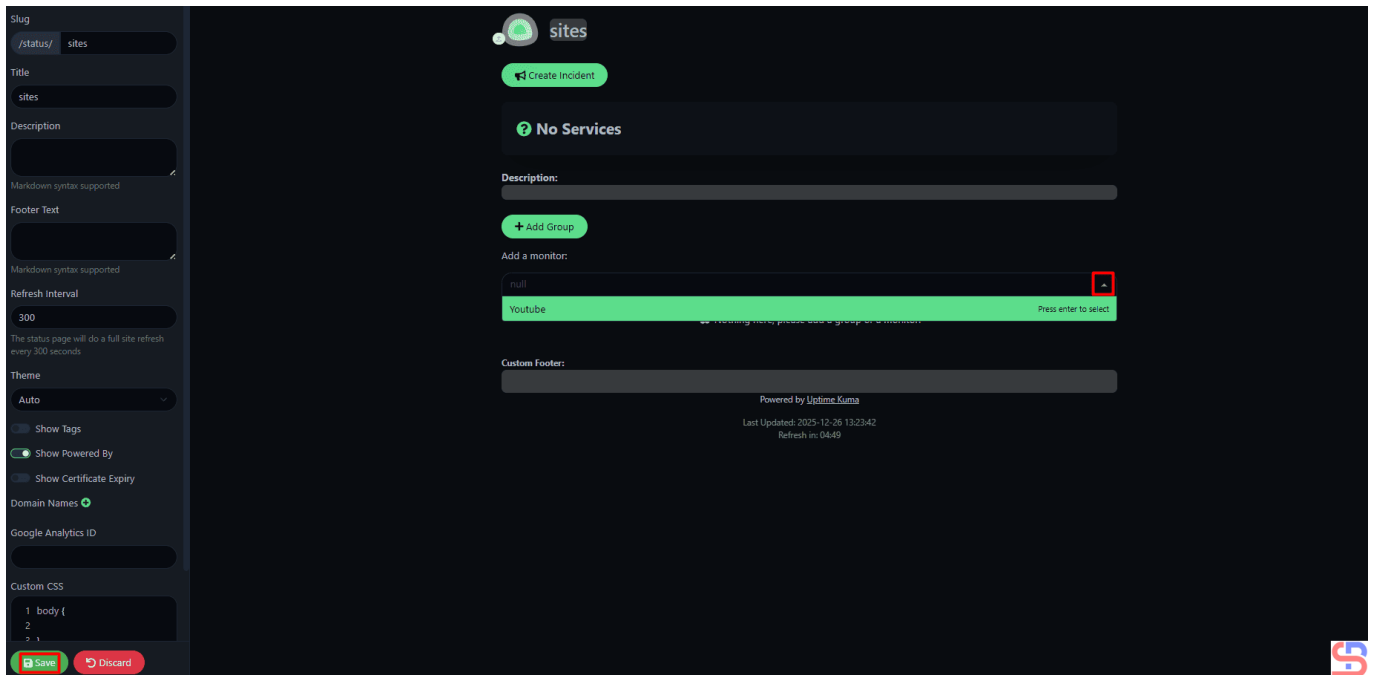
Create the Status Page page

Click the **New Status Page** button, and an image will appear similar to the one shown below:



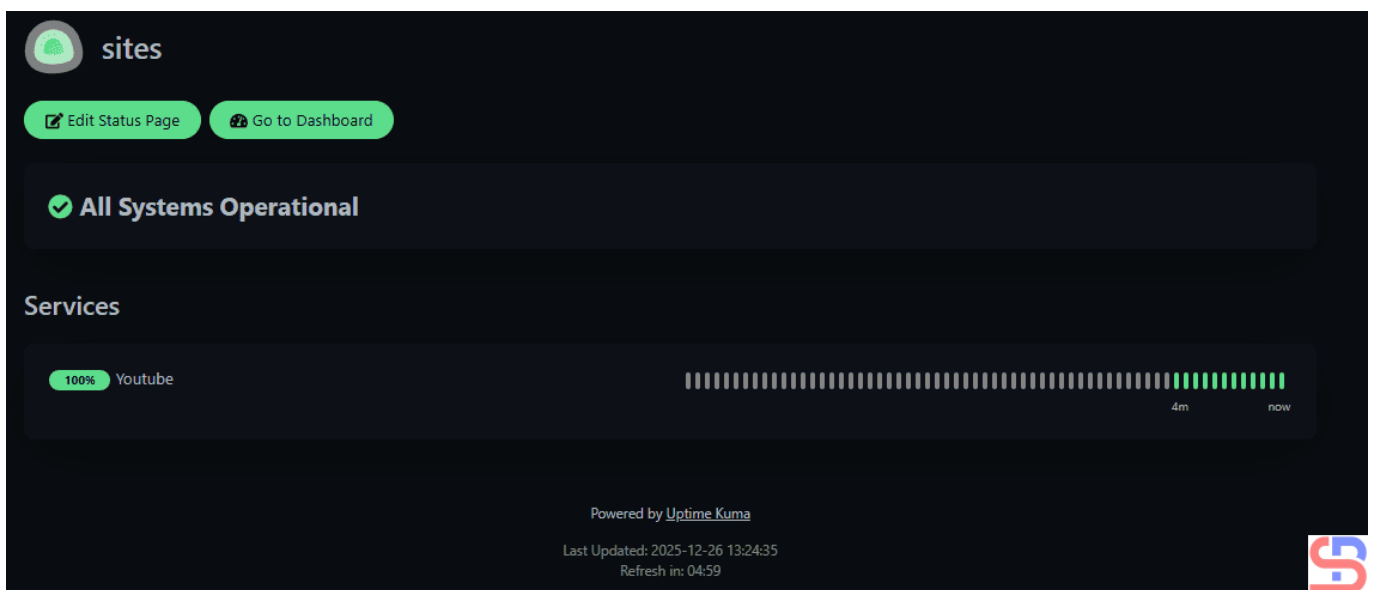
Create the Status Page page

Enter the name and slug you want (I wrote the sites for the name and slug), then press the Next button, 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 check the status of uptime kuma in the server, run the command below :

```
sudo pm2 status server/server.js --name uptime-kuma
```

```
sysadmin@ubuntu24:~/uptime-kuma$ sudo pm2 status server/server.js --name uptime-kuma
```

id	name	namespace	version	mode	pid	uptime	♻	status	cpu	mem	user	watching
1	uptime-kuma	default	2.0.1	fork	25195	75s	0	online	0%	177.2mb	root	disabled

Module

id	module	version	pid	status	♻	cpu	mem	user
0	pm2-logrotate	3.0.0	25165	online	0	0%	73.0mb	root

```
sysadmin@ubuntu24:~/uptime-kuma$
```

Check the status of uptime kuma

But if you want stop uptime kuma in the server, run the command below :

```
sudo pm2 stop server/server.js --name uptime-kuma
```

```
sysadmin@ubuntu24:~/uptime-kuma$ sudo pm2 stop server/server.js --name uptime-kuma
[PM2] Applying action stopProcessId on app [server/server.js](ids: [ 1 ])
[PM2] [uptime-kuma](1) v
```

id	name	namespace	version	mode	pid	uptime	♻	status	cpu	mem	user	watching
1	uptime-kuma	default	2.0.1	fork	0	0	0	stopped	0%	0b	root	disabled

Module

id	module	version	pid	status	♻	cpu	mem	user
0	pm2-logrotate	3.0.0	25165	online	0	0%	65.9mb	root

```
sysadmin@ubuntu24:~/uptime-kuma$
```

Stop status kuma service

References

uptimekuma.org
hostmycode.in
youtube.com

[How to Install Uptime Kuma in Docker](#)

with MariaDB on the Host?

written by sysadmin | 14 January 2026

[The previous article](#) explained how to install the Kuma uptime application using Docker and using the MariaDB database which also runs on Docker. This article will explain about installing the Kuma uptime application using Docker but using MariaDB on the host.

Problem

How to install uptime kuma in docker with MariaDB on the host?

Solution

Here are the steps to install the kuma uptime application using docker but using MariaDB on the host:

1. Configure MariaDB

Make sure you have installed the MariaDB database on your server and then change the file `/etc/mysql/mariadb.conf.d/50-server.cnf` in the **bind-address** section to be as below:

```
bind-address          = 0.0.0.0
```

Next, restart MariaDB using the command:

```
sudo systemctl restart mariadb
```

After that, access to MariaDB and run the command below:

```
CREATE DATABASE uptime_kuma;  
CREATE USER 'kuma-user'@'%' IDENTIFIED BY 'kumapass123';  
GRANT ALL PRIVILEGES ON uptime_kuma.* TO 'kuma-user'@'%';  
FLUSH PRIVILEGES;
```

\q

2. 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 a **docker-compose.yaml** file and copy the script below:

```
services:  
  uptime-kuma:  
    image: louislam/uptime-kuma:2  
    container_name: uptime-kuma  
    restart: unless-stopped  
    ports:  
      - "3001:3001"  
    extra_hosts:  
      - "host.docker.internal:host-gateway"  
    environment:  
      UPTIME_KUMA_DB_TYPE: mariadb  
      UPTIME_KUMA_DB_HOSTNAME: host.docker.internal  
      UPTIME_KUMA_DB_PORT: 3306  
      UPTIME_KUMA_DB_NAME: ${MARIADB_DATABASE}  
      UPTIME_KUMA_DB_USERNAME: ${MARIADB_USER}  
      UPTIME_KUMA_DB_PASSWORD: ${MARIADB_PASSWORD}  
    volumes:  
      - kuma-data:/app/data  
  
volumes:  
  kuma-data:
```

After that, create a **.env** file like the below script (The value must be the same as the value you ran the query in MariaDB):

```
MARIADB_DATABASE=uptime_kuma  
MARIADB_USER=kuma-user  
MARIADB_PASSWORD=kumapass123  
MARIADB_ROOT_PASSWORD=qwerty
```

Run the below command to turn on the 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:

```
sysadmin@docker:~$ docker compose up -d
[+] up 3/3
 ✓ Network sysadmin_default Created 0.5s
 ✓ Volume sysadmin_kuma-data Created 0.0s
 ✓ Container uptime-kuma Created 0.6s
sysadmin@docker:~$
sysadmin@docker:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS              PORTS                               NAMES
4bd48bf06585   louislam/uptime-kuma:2             "/usr/bin/dumb-init _"  7 seconds ago   Up 6 seconds (health: starting)  0.0.0.0:3001->3001/tcp, [::]:3001->3001/tcp  uptime-kuma
```

Check the running container

3. Configure webserver

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
```

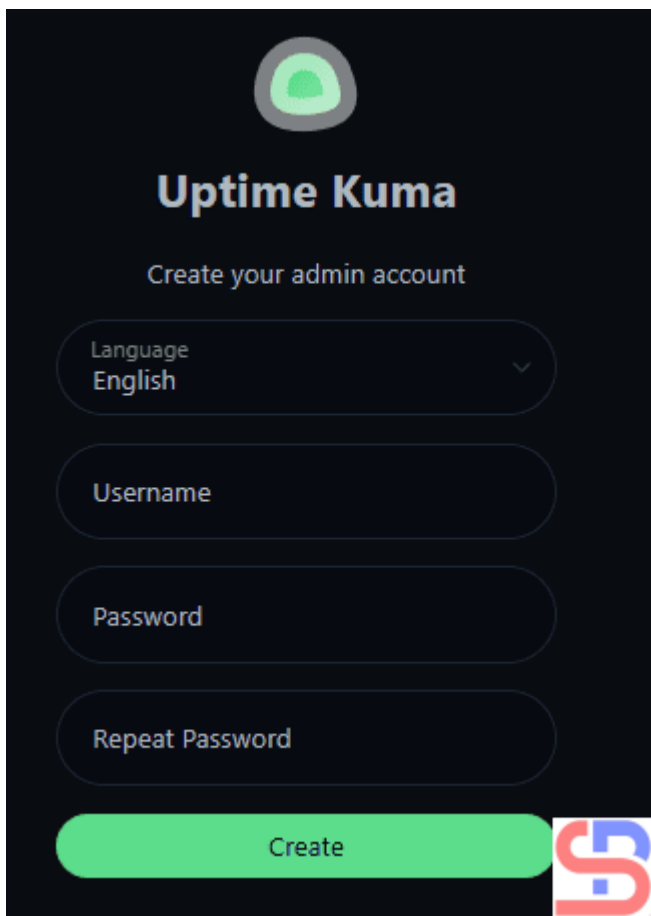
```
sudo systemctl reload nginx
```

4. Access uptime kuma

Open your browser, and type:

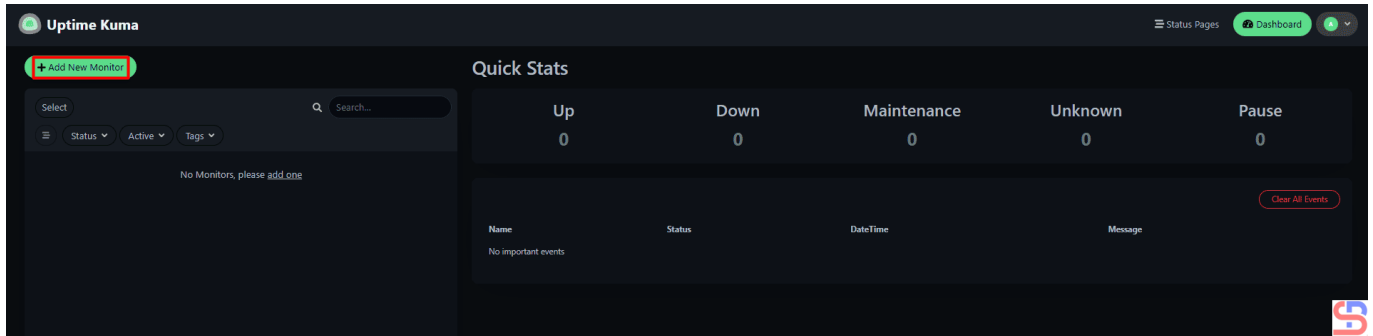
```
http://ip_server:3001
```

then there will be a display like below:

The image shows a dark-themed web interface for creating an admin account. At the top, there is a green circular logo with a white shape inside. Below the logo, the text "Uptime Kuma" is displayed in a bold, white font. Underneath, it says "Create your admin account" in a smaller, lighter font. There are four input fields: "Language" (with a dropdown menu showing "English"), "Username", "Password", and "Repeat Password". At the bottom, there is a prominent green "Create" button. To the right of the button, there is a small logo consisting of a blue and red stylized "S" or "G" shape.

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 use 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
2025-12-26T08:18:58Z [SERVER] INFO: Env: production
2025-12-26T08:19:01Z [SERVER] INFO: Uptime Kuma Version: 2.0.2
2025-12-26T08:19:01Z [SERVER] INFO: Loading modules
2025-12-26T08:19:05Z [SERVER] INFO: Creating express and socket.io instance
2025-12-26T08:19:05Z [SERVER] INFO: Server Type: HTTP
2025-12-26T08:19:05Z [SERVER] INFO: Data Dir: ./data/
2025-12-26T08:19:05Z [SETUP-DATABASE] INFO: db-config.json is not found or invalid: ENOENT: no such file or directory, open 'data/db-config.json'
2025-12-26T08:19:05Z [DB] INFO: Database Type: mariadb
2025-12-26T08:19:06Z [MARIADB] INFO: Creating basic tables for MariaDB
2025-12-26T08:19:15Z [MARIADB] INFO: Created basic tables for MariaDB
2025-12-26T08:19:15Z [SERVER] INFO: Connected to the database
sysadmin@docker:~$
```

Check the running database

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

Add New Monitor

General

Monitor Type:

Friendly Name:

URL:

Heartbeat Interval (Check every 60 seconds):
1 minute

Retries:
Maximum retries before the service is marked as down and a notification is sent

Heartbeat Retry Interval (Retry every 60 seconds):

Request Timeout (Timeout after 48 seconds):

Resend Notification if Down X times consecutively (Resend disabled):

Advanced

Notifications

Not available, please set up.

Proxy

Not available, please set up.

HTTP Options

Method:

Body Encoding:

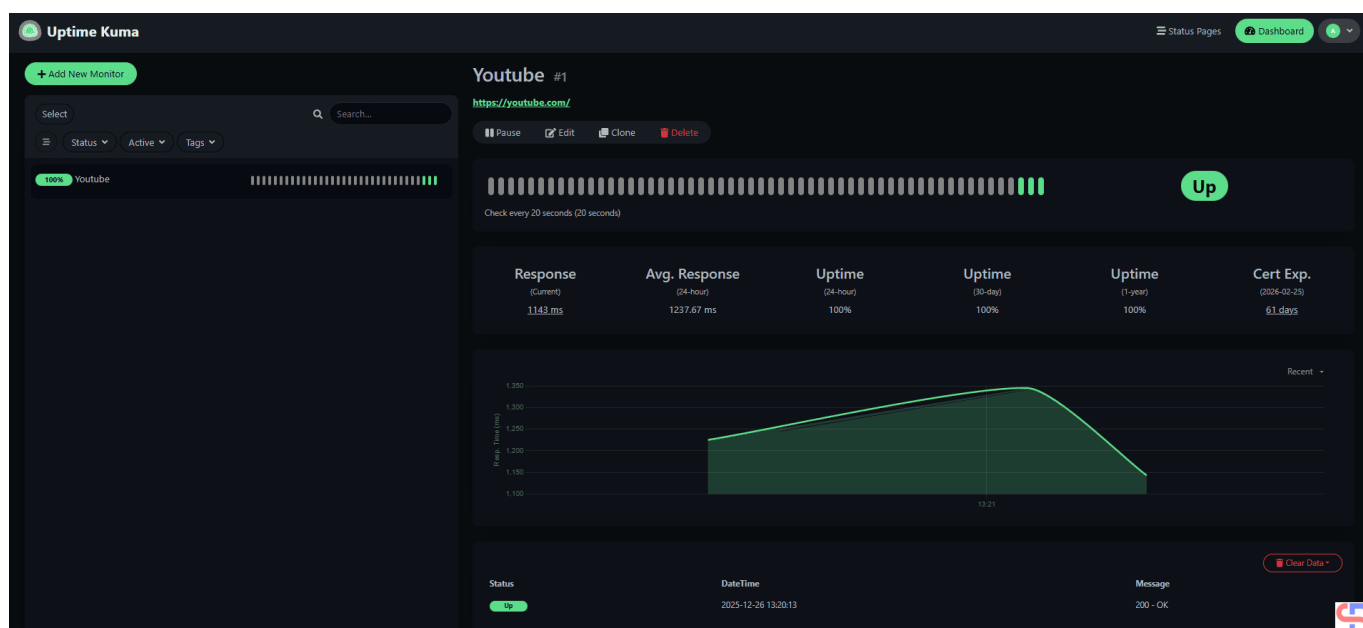
Body:

```
Example:
{
  "key": "value"
}
```

Headers:

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:



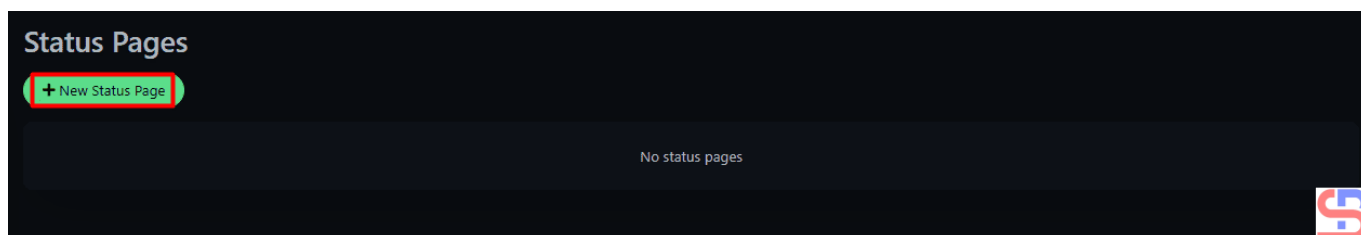
Monitor the host or the website

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:



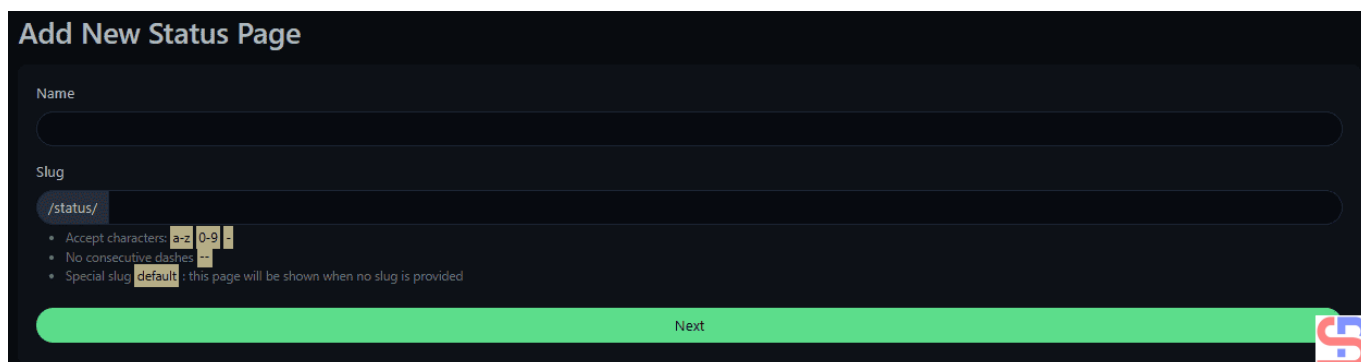
Click the Status Page button

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



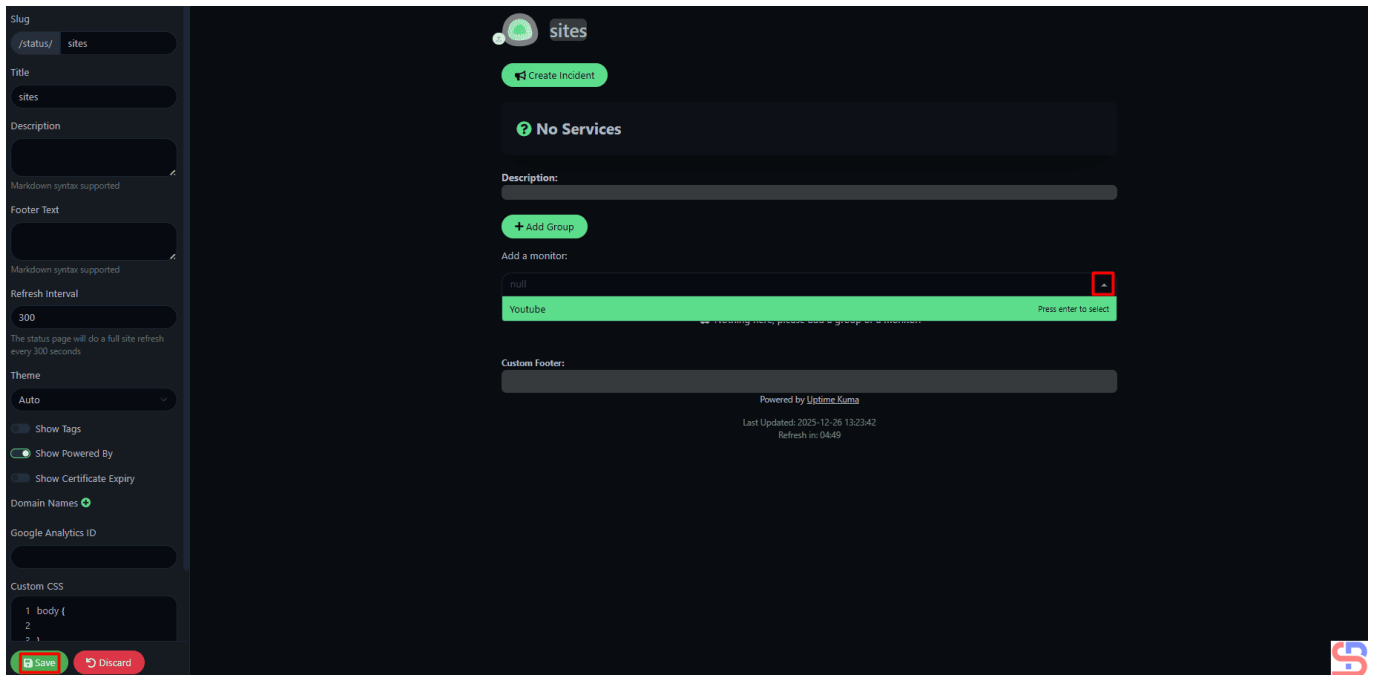
Create the Status Page page

Click the **New Status Page** button, and an image will appear similar to the one shown below:



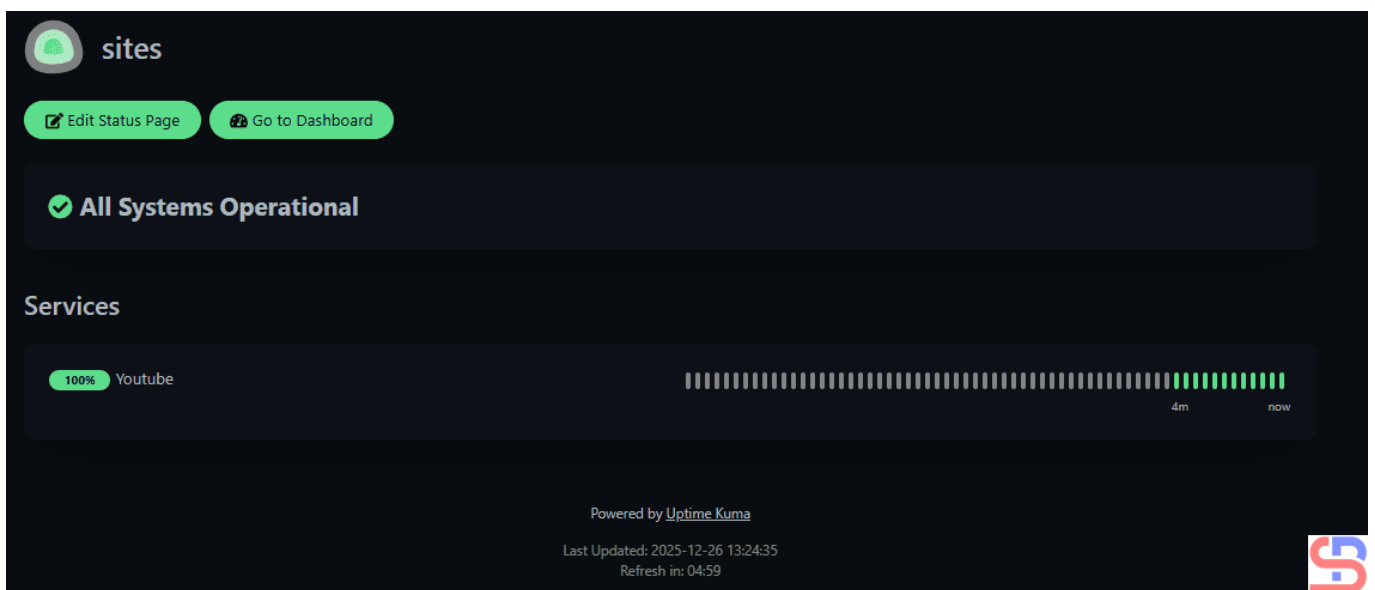
Create the Status Page page

Enter the name and slug you want (I wrote the sites for the name and slug), then press the Next button, 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

Go to [this page](#) if you want to backup the MariaDB database and for how to restore the database, you can go to [this page](#).

References

sysadminpedia.com

magnus919.com

uptimekuma.org

[How to Backup And Restore Uptime Kuma Database in Docker?](#)

written by sysadmin | 14 January 2026

[The previous article](#) explained how to install Uptime Kuma using Docker on Linux. This article will explain how to back up and restore the Kuma uptime database in Docker.

Problem

How to back up and restore the Uptime Kuma database in Docker?

Solution

Below are the steps to back up and restore the Uptime Kuma database in Docker:

A. Database SQLite

Here is the method to back up and restore a SQLite database in Docker:

1. Backup database

If you want to back up the Uptime Kuma database in Docker, you can run the command below to get the Kuma database on your host:

```
docker run --rm
-v uptime-kuma:/data
-v $(pwd):/backup
alpine tar czf /backup/kuma-backup.tar.gz /data
```

After that, look in your current folder; the database should appear like the image below:

```
sysadmin@docker:~$ docker run --rm \
-v uptime-kuma:/data \
-v $(pwd):/backup \
alpine tar czf /backup/kuma-backup.tar.gz /data
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
1074353eec0d: Pull complete
5c1f58ba4e0d: Download complete
644afed44dca: Download complete
Digest: sha256:865b95f46d98cf867a156fe4a135ad3fe50d2056aa3f25ed31662dff6da4eb62
Status: Downloaded newer image for alpine:latest
tar: removing leading '/' from member names
sysadmin@docker:~$ ls
crud  get-docker.sh  kuma-backup.tar.gz  main.zip
```

Back up the uptime Kuma database

However, if you want to back up automatically, then follow the steps below:

a. Create a backup folder on the host

Run the commands below to create a backup folder in the host:

```
sudo mkdir -p /opt/kuma-backup
sudo chown root:root /opt/kuma-backup
sudo chmod 700 /opt/kuma-backup
```

b. Run the backup script

Create a file, for example, backup-kuma.sh, in the root

folder containing the bash script below:

```
#!/bin/bash
set -e

# CONFIG
CONTAINER_NAME="uptime-kuma"
VOLUME_NAME="uptime-kuma"
BACKUP_DIR="/opt/kuma-backup"
RETENTION_DAYS=14
DATE=$(date +"%Y%m%d-%H%M%S")
BACKUP_FILE="$BACKUP_DIR/kuma-backup-$DATE.tar.gz"

# Backup
docker run --rm \
  -v $VOLUME_NAME:/data:ro \
  -v $BACKUP_DIR:/backup \
  alpine \
  tar czf /backup/${basename $BACKUP_FILE} /data

# Cleanup old backups
find $BACKUP_DIR -type f -name "kuma-backup-*.tar.gz" -mtime +$RETENTION_DAYS
-delete
```

Save the file and make it executable using the command:

```
sudo chmod +x /root/backup-kuma.sh
```

If you run the script, the backup database file should be in the **/opt/backup** folder.

c. Set up a Cron Job

If you want to run the script every 2:30 AM, then on the crontab, write the script as shown in the image below

```
0 2 * * * /root/backup-kuma.sh >> /root/kuma-backup.log 2>&1
```

2. Restore database

Before you restore the database, make sure the container has been running first. If you want to restore the Uptime Kuma database that you have previously backed up, then you can run the command below on the host:

```
docker run --rm \
-v uptime-kuma:/data \
-v /opt/kuma-backup:/backup \
alpine \
tar xzf /backup/kuma-backup-YYYYMMDD-HHMMSS.tar.gz -C /
```

If the restore process is complete, the hosts that were monitored in the previous container should be monitored again by the new container.

B. Database MariaDB

Here is the method to back up and restore a MariaDB database in Docker:

1. Backup database

If you want to back up the MariaDB database in Docker, you can run the command below to get the database on your host:

```
docker exec mariadb \
mysql_dump -u root -p kuma > kuma.sql
```

Change the kuma with your database name. Or use the command below if you want to compress the result:

```
docker exec mariadb \
mysql_dump -u root -p kuma | gzip > kuma.sql.gz
```

You can use the script below to back up the database:

```
#!/bin/bash

# =====
# CONFIGURATION
# =====
CONTAINER_NAME="mariadb"
DB_USER="backup_user"
DB_PASSWORD="PASSWORD_DB"
DB_NAME="appdb"

BACKUP_DIR="/opt/backup/mariadb"
RETENTION_DAYS=7
DATE=$(date +"%Y-%m-%d_%H-%M-%S")
```

```

BACKUP_FILE="$BACKUP_DIR/${DB_NAME}_${DATE}.sql.gz"
LOG_FILE="$BACKUP_DIR/backup.log"

# =====
# PREPARATION
# =====
mkdir -p "$BACKUP_DIR"

echo "[$(date)] Backup started" >> "$LOG_FILE"

# =====
# BACKUP DATABASE
# =====
docker exec "$CONTAINER_NAME" \
  mysqldump -u"$DB_USER" -p"$DB_PASSWORD" "$DB_NAME" \
  --single-transaction \
  --quick \
  --routines \
  --triggers \
  --events \
  | gzip > "$BACKUP_FILE"

# =====
# VALIDATION
# =====
if [ $? -eq 0 ]; then
  echo "[$(date)] Backup SUCCESS: $BACKUP_FILE" >> "$LOG_FILE"
else
  echo "[$(date)] Backup FAILED" >> "$LOG_FILE"
  exit 1
fi

# =====
# BACKUP ROTATION (DELETE OLD FILE)
# =====
find "$BACKUP_DIR" -type f -name "*.sql.gz" -mtime +$RETENTION_DAYS -delete

echo "[$(date)] Old backups cleaned (>${RETENTION_DAYS} days)" >> "$LOG_FILE"
echo "[$(date)] Backup finished" >> "$LOG_FILE"

```

And you can insert the script above in the crontab.

2. Restore database

Before you restore the database, make sure the container has been running first. If you want to restore the database that you have previously backed up, then you can run the command below on the host:

```
docker exec -i mariadb \  
mysql -u root -p -e "CREATE DATABASE kuma;"
```

```
docker exec -i mariadb \  
mysql -u root -p kuma < kuma.sql
```

Note

If you have a failure to back up the database, you can see the logs in the `/root/backup-kuma.log` file.

References

fishparts.net

homelab.anita-fred.net

[How to Install Uptime Kuma And MariaDB in Docker?](#)

written by sysadmin | 14 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:

```
sysadmin@docker:~$ docker compose up -d
[+] up 5/5
✓ Network sysadmin_kuma-net Created 0.4s
✓ Volume sysadmin_kuma-data Created 0.0s
✓ Volume sysadmin_mariadb-data Created 0.0s
✓ Container mariadb Healthy 44.8s
✓ Container uptime-kuma Created 0.6s
sysadmin@docker:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
731c6fa09a1e	louislam/uptime-kuma:2	"/usr/bin/dumb-init _"	About a minute ago	Up 17 seconds (health: starting)	0.0.0.0:3001->3001/tcp, [::]:3001->3001/tcp	uptime-kuma
c2b66c4b3ff1	mariadb:11.4	"docker-entrypoint.s_"	About a minute ago	Up About a minute (healthy)	3306/tcp	mariadb

```
sysadmin@docker:~$
```

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

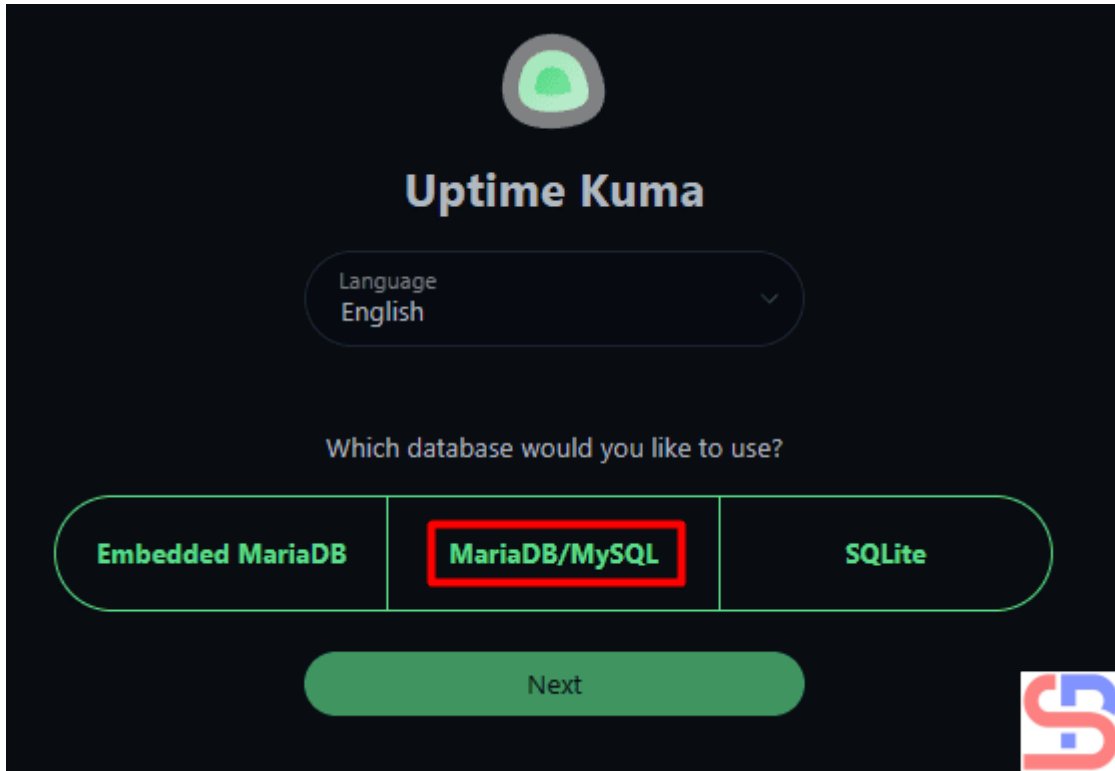
```

3. Access uptime kuma

Open your browser, and type:

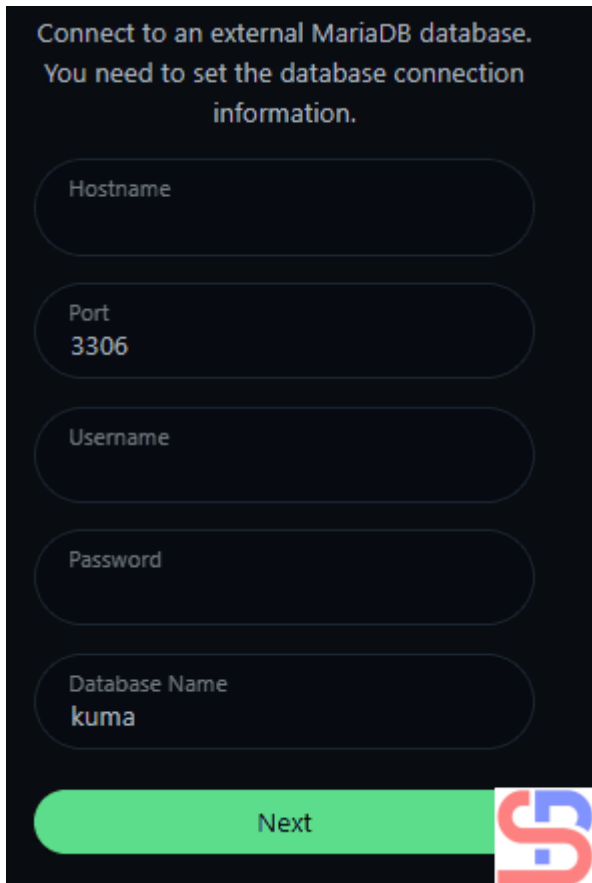
```
http://ip_server:3001
```

Then there will be a display like below:



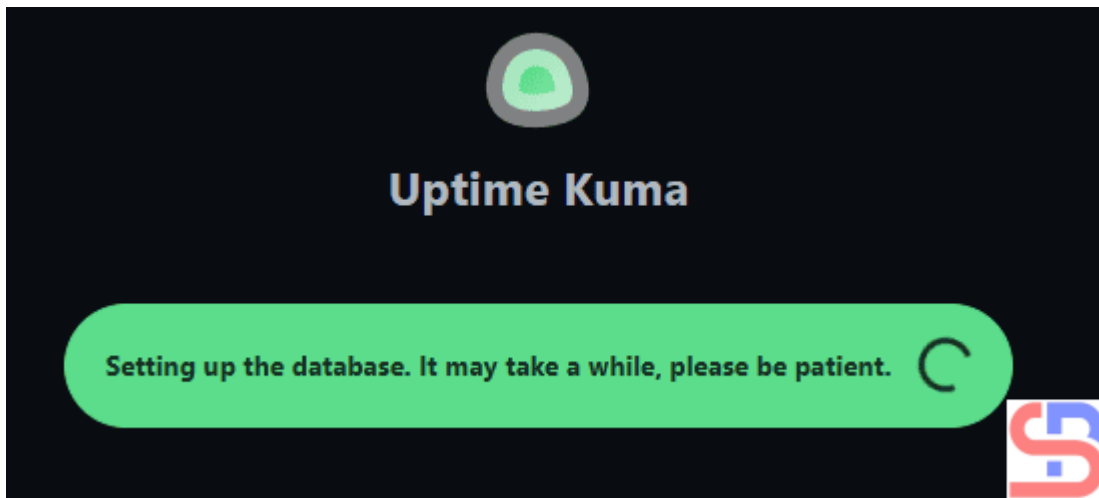
Choose MariaDB/MySQL

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



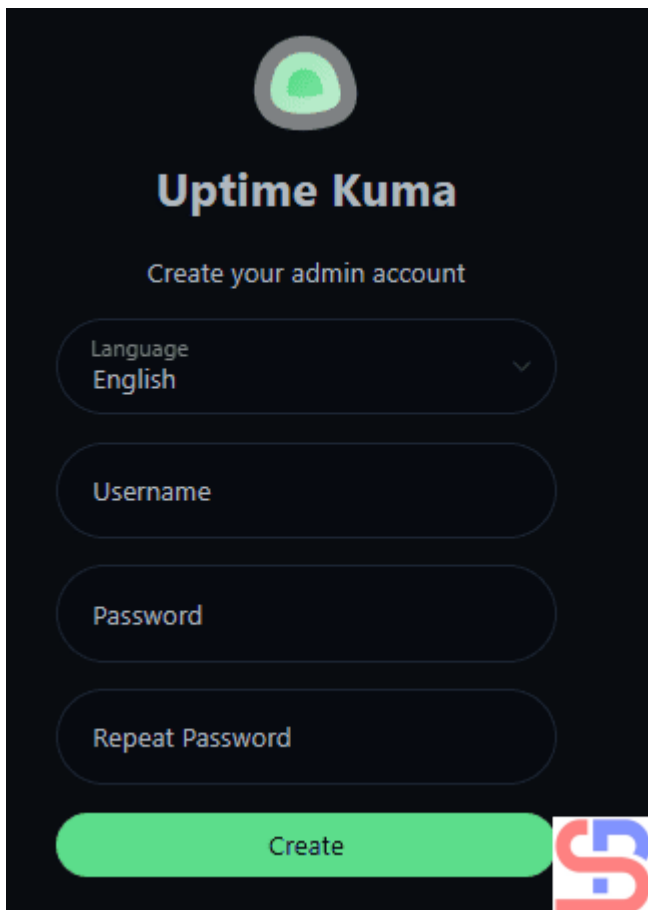
Fill in the columns for the database

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

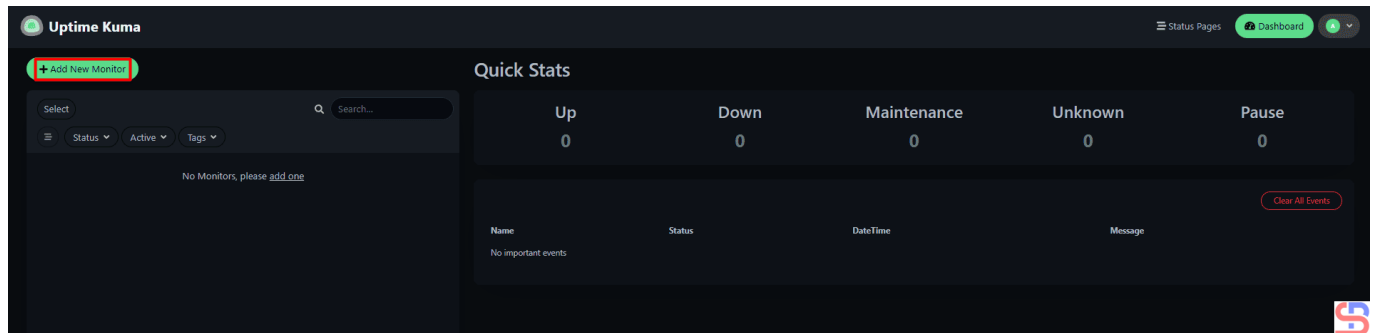
You have to wait until finishes, and after that, your screen will appear similar to the image shown below:



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:

Add New Monitor

General

Monitor Type:

Friendly Name:

URL:

Heartbeat Interval (Check every 60 seconds):
1 minute

Retries:
Maximum retries before the service is marked as down and a notification is sent

Heartbeat Retry Interval (Retry every 60 seconds):

Request Timeout (Timeout after 48 seconds):

Resend Notification if Down X times consecutively (Resend disabled):

Advanced

Notifications

Not available, please set up.

Proxy

Not available, please set up.

HTTP Options

Method:

Body Encoding:

Body:

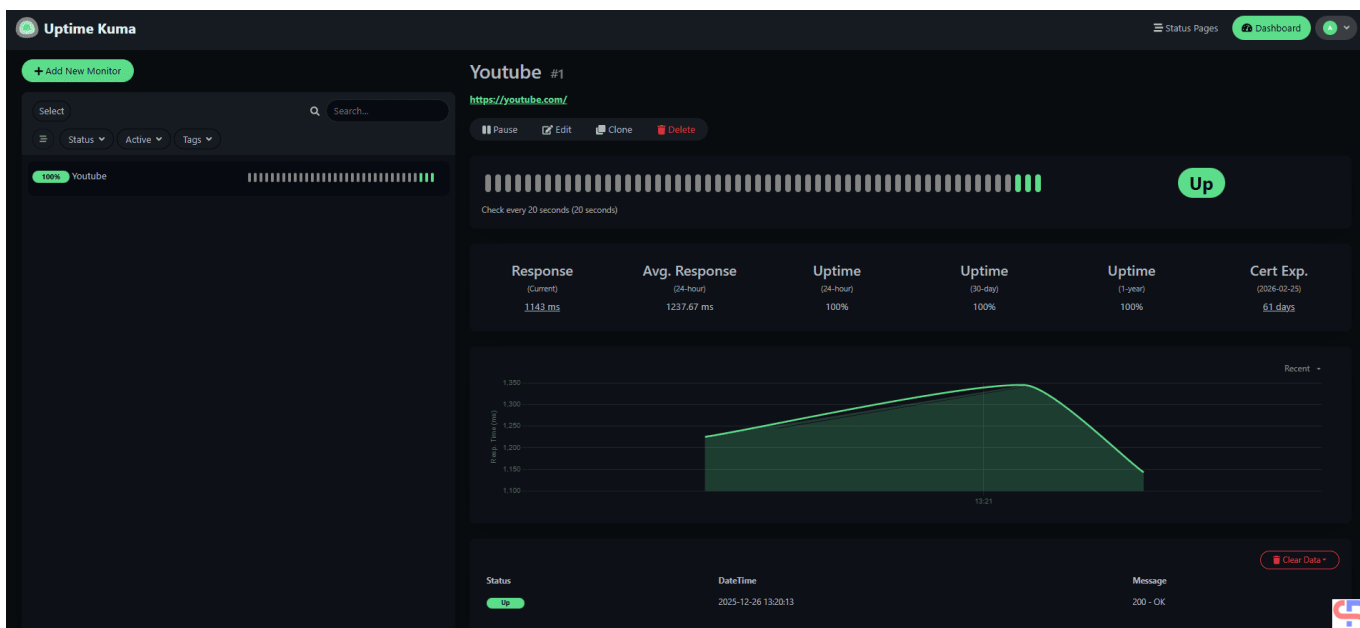
```
Example:
{
  "key": "value"
}
```

Headers:

```
Example:
```

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:



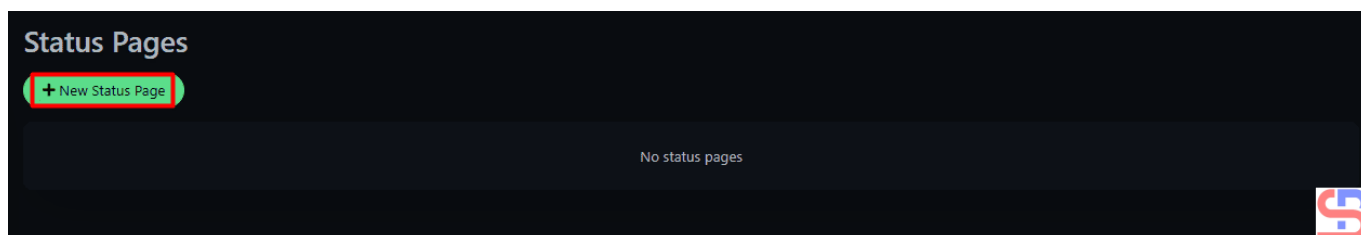
Monitor the host or the website

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:



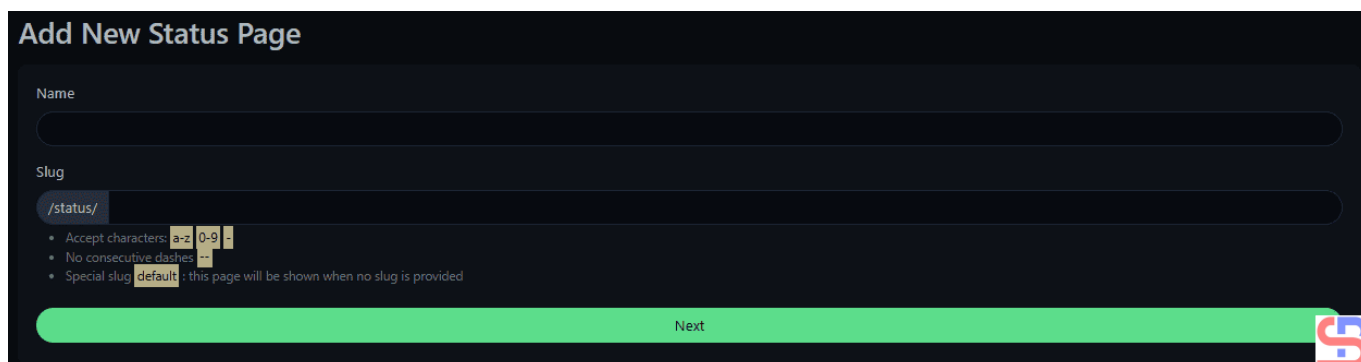
Click the Status Page button

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



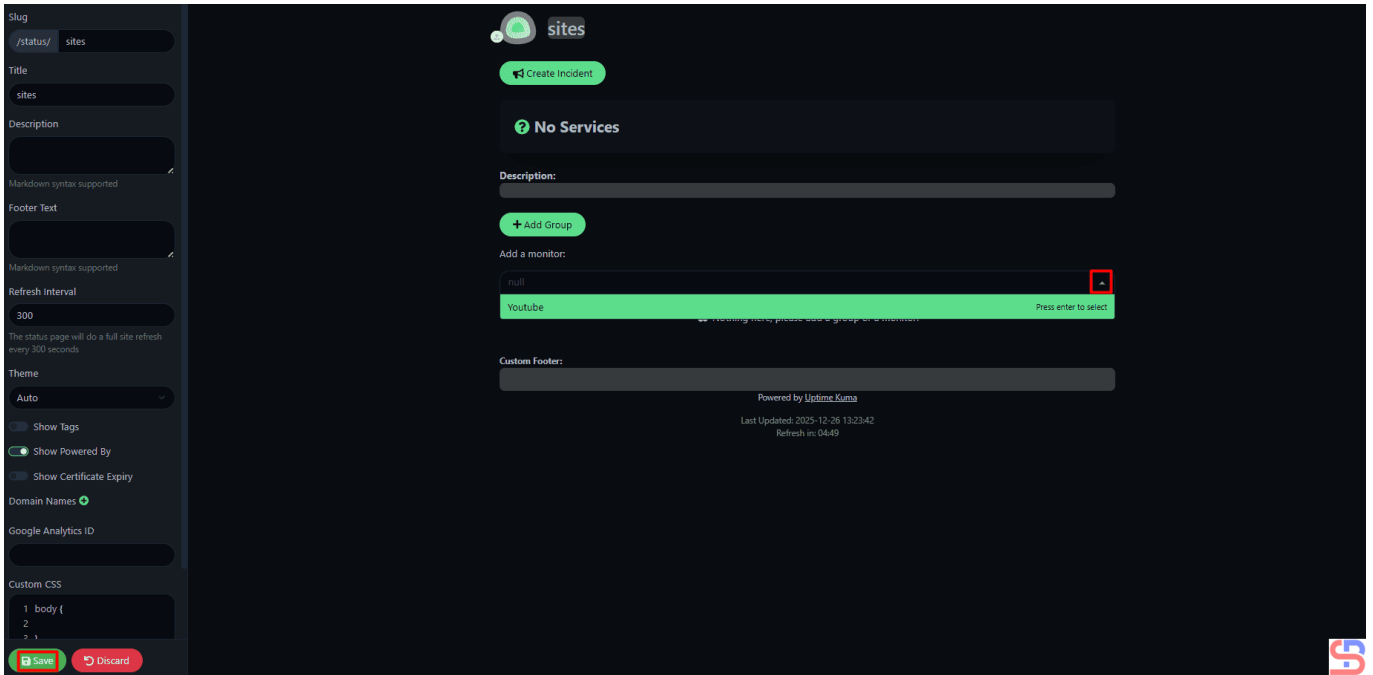
Create the Status Page page

Click the **New Status Page** button, and an image will appear similar to the one shown below:



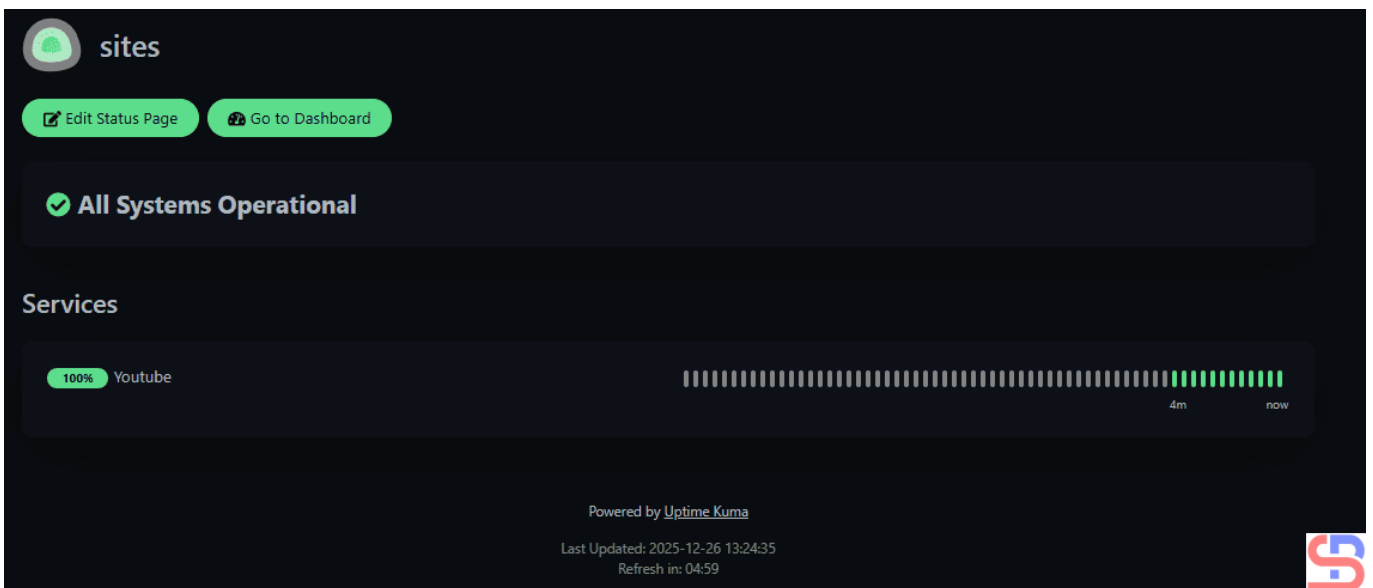
Create the Status Page page

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](#)

[How to Install Uptime Kuma on Linux?](#)

written by sysadmin | 14 January 2026

Uptime Kuma is a self-hosted monitoring solution created to measure the uptime and performance of websites and services. It offers live status updates, flexible alerting choices, and comprehensive metrics to help guarantee that your websites and services stay functional.

Problem

How to install Uptime Kuma on Linux?

Solution

There are 4 methods to install uptime kuma:

1. Using docker.
2. [Using docker compose with database in docker](#).
3. [Using docker compose with database in the host](#).
4. [Using package](#).

This article will explain how to install Kuma using Docker.

1. Install uptime kuma

Make sure you installed Docker in your server and you can see how to install Docker on [this page](#). After that, run the command below to install uptime kuma using docker:

```
docker run -d --restart=always -p 3001:3001 -v uptime-kuma:/app/data --name uptime-kuma louislam/uptime-kuma:1
```

Then check whether the uptime kuma container is running or not using the command:

```
docker ps | grep Kuma
```

```
sysadmin@docker:~$ docker ps | grep kuma
eb2f41d7b0b9   louislam/uptime-kuma:1   "/usr/bin/dumb-init _"   32 seconds ago   Up 30 seconds (healthy)   0.0.0.0:3001->3001/tcp, [::]:3001->3001/tcp   uptime-k
```

Check the uptime kuma container in Docker

2. Configure webserver

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:

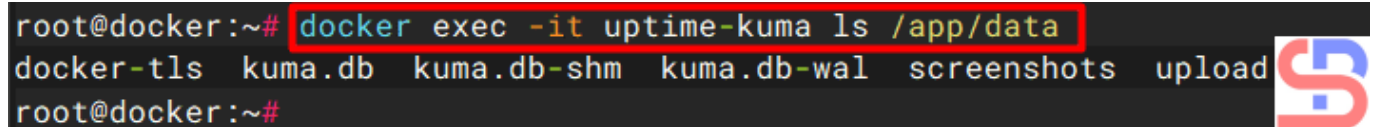
```
nginx -t
sudo systemctl reload nginx
```

3. Configure database

If you install uptime kuma using docker, you **don't need to install the database** because in the docker there is already a SQLite database where you can view it by using the command below:

```
docker exec -it uptime-kuma ls /app/data
```

```
root@docker:~# docker exec -it uptime-kuma ls /app/data
docker-tls  kuma.db  kuma.db-shm  kuma.db-wal  screenshots  upload
root@docker:~#
```



Check the database in the uptime kuma container

4. Access uptime kuma

Open your browser, and type:

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

then there will be a display like below:



Uptime Kuma

Create your admin account

Language
English ▼

Username

Password

Repeat Password

Create



Create username and password for Uptime Kuma

Enter the username and password you want then press the **Create** button, there will be a display as below:

The screenshot shows the Uptime Kuma dashboard. At the top left, there is a header with the Uptime Kuma logo and a '+ Add New Monitor' button. On the right, there are navigation links for 'New Update', 'Status Pages', 'Dashboard', and a user profile dropdown. The main content area is divided into two sections. The left section is a sidebar with a search bar and filters for 'Status', 'Active', and 'Tags'. Below this, it says 'No Monitors, please [add one](#)'. The right section is titled 'Quick Stats' and displays five categories: 'Up' (0), 'Down' (0), 'Maintenance' (0), 'Unknown' (0), and 'Pause' (0). Below the stats is a table with columns for 'Name', 'Status', 'DateTime', and 'Message', which currently shows 'No important events'. A small Uptime Kuma logo is visible in the bottom right corner of the dashboard.

Display of uptime kema application

If you want to monitor a host or a website, click the **Add New Monitor** button like in the below image:

Uptime Kuma

New Update Status Pages Dashboard

+ Add New Monitor

Select Search monitored sites

Status Active Tags

No Monitors, please [add one](#)

Add New Monitor

General

Monitor Type: HTTP(s)

Friendly Name:

URL: https://

Heartbeat Interval (Check every 60 seconds): 60

Retries: 0

Maximum retries before the service is marked as down and a notification is sent

Save (Retry Interval (Retry every 60 seconds))

Notifications

Not available, please setup.

Setup Notification

Proxy

Not available, please setup.

Setup Proxy

HTTP Options

Method: GET

Body Encoding: JSON

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:

Uptime Kuma

New Update Status Pages Dashboard

+ Add New Monitor

Select Search monitored sites

Status Active Tags

100% Youtube

Youtube

https://youtube.com/

Pause Edit Clone Delete

Check every 60 seconds

Up

Response	Avg. Response	Uptime	Uptime	Cert Exp.
(Current) 952 ms	(24-hour) 974 ms	(24-hour) 100%	(30-day) 100%	(2028-02-25) 53 days

Recent

Response Time (ms)

Status: Up

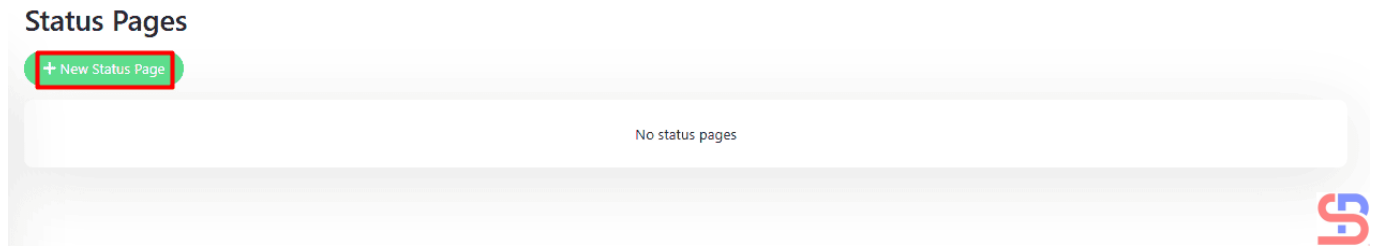
DateTime: 2025-12-24 16:09:18

Message: 200 - OK

Clear Data

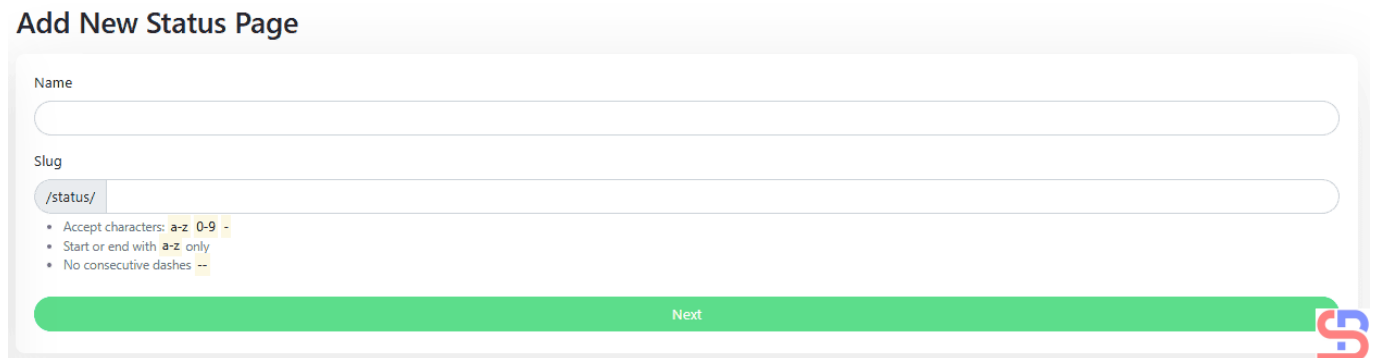
Monitor the host or the website

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 layer then there will be a display like below:



Click the New Status Page button

Click the **New Status** page button, then there will be a display as 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, then there will be a display as below:

Slug
/status/ sites

Title
sites

Description
Markdown syntax supported

Footer Text
Markdown syntax supported

Theme
Auto

Show Tags

Show Powered By

Show Certificate Expiry

Domain Names

Google Analytics ID

Custom CSS

```
1 body {
2
3 }
4
```

[Delete](#)

[Save](#) [Discard](#)

sites

[Create Incident](#)

No Services

Description:

[Add Group](#)

Add a monitor:

Add a monitor
Youtube Press enter to select

Custom Footer:

Powered by [Uptime Kuma](#)
Last Updated: 2025-12-24 16:55:02
Refresh in: 04:07

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:

Not secure 192.168.56.105:3001/status/sites

sites

[Edit Status Page](#) [Go to Dashboard](#)

All Systems Operational

Services

60.51% Youtube 47m ago now

Powered by [Uptime Kuma](#)
Last Updated: 2025-12-24 16:56:34
Refresh in: 04:15

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 backup the uptime kuma database running on docker and how to restore the database, you can go to [this page](#).

References

uptimekuma.org
kb.biznetgio.com

[How to Display the Results of a Script in Zabbix?](#)

written by sysadmin | 14 January 2026

I want to create a monitoring to check if a site is in an error state or not using a script, and the results of this script will be sent to Zabbix for monitoring.

Problem

How to display the results of a script in Zabbix?

Solution

For example, I have a sysadminpedia.com site and want to monitor the site. The way I do monitoring is to look for wordpress writing on the site, and if the wordpress writing is not on the site, it means that the site has an error. I use a bash script to monitor the word on sysadminpedia.com. For the site to be monitored by Zabbix based on the results of the script I created, follow the steps below:

1. Create a script

Log in to the Zabbix server, and you can use any folder to create your script, but I created a special folder for scripts in Zabbix using the command:

```
sudo mkdir -p /etc/zabbix/scripts
```

Then create a bash script in a folder with the file name **check-sysadminpedia-site.sh** and copy the script below:

```
#!/bin/bash

# Fetch the website content
content=$(curl -s https://sysadminpedia.com)

# Check if the word "wordpress" exists (case-insensitive)
if echo "$content" | grep -iq "wordpress"; then
    echo 1
else
    echo 0
fi
```

2. Change the user, group, and permission

Change the user and group on the file using the command below:

```
chown -R zabbix:zabbix /etc/zabbix/scripts/check-sysadminpedia-site.sh
```

After that, type the following command to make the script run:

```
chmod +x /etc/zabbix/scripts/check-sysadminpedia-site.sh
```

3. Configure in the zabbix_agent file

Add the below script in the file **/etc/zabbix/zabbix_agentd.conf**:

```
UserParameter=check-sysadminpedia-site,/etc/zabbix/scripts/check-sysadminpedia-site.sh
```

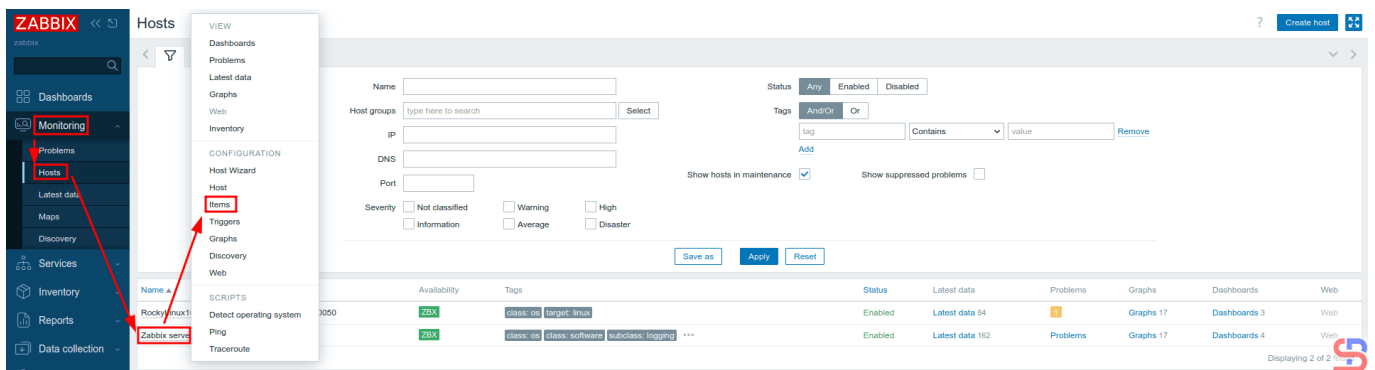
4. Restart zabbix_agent

Restart the Zabbix agent using the following commands:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

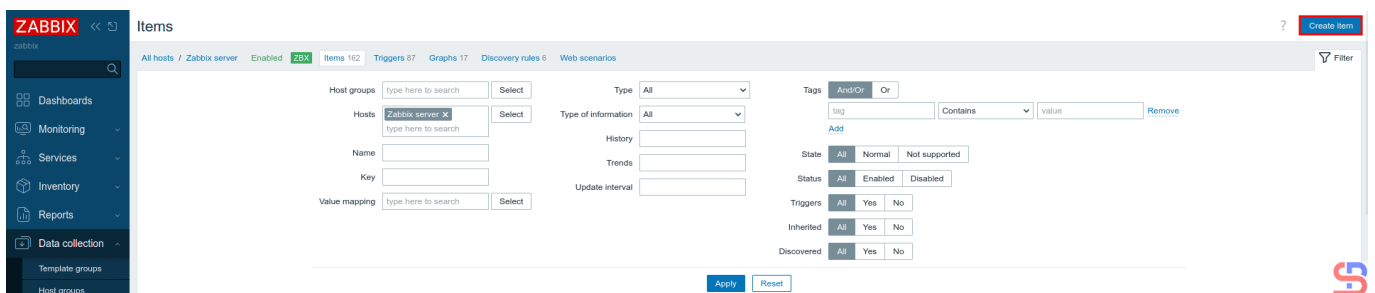
5. Configure Zabbix

Go to your Zabbix application, select the Host you want to enter to display the results of this monitoring in Zabbix. I choose to use the Zabbix server host: **Monitoring > Hosts > Zabbix server > Items** in the **CONFIGURATION** like in the image below:



Click Items in the CONFIGURATION section

And there will be a display like the following:



Click the **Create item** button, and then there will be a display as shown below:

New item

? X

Item Tags Preprocessing

* Name

Type

* Key

Type of information

* Host interface

Units

* Update interval

Custom intervals

Type	Interval	Period
<input type="button" value="Flexible"/> <input type="button" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/> <input type="button" value="Remove"/>

* Timeout

* History

* Trends

Value mapping

Populates host inventory field

Description

Enabled

Click the Test button

I fill in the fields as in the image above, click the **Test** button, and then there will be a display as in the image below:

Test item

? X

Get value from host

* Host address Port

Test with

Value

Time

Not supported Error

Previous value Prev. time

End of line sequence

Click the Get value and test button

Click the **Get value and test** button, and in the **Value** section, there will be a value generated, either it is 1 or 0, according to the value in the bash script, as in the image below:

Test item ? X

Get value from host

* Host address Port

Test with Server Proxy

Get value

Value ↙ Time

Not supported Error ↙

Previous value ↙ Prev. time

End of line sequence LF CRLF

Result 1 📄

Get value and test Cancel

Click the **Cancel** button

You see from the image above, the Value is 1. Click the **Cancel** button, then it will return to the previous view, like the image below:

* Name

Type

* Key

Type of information

* Host interface

Units

* Update interval

Custom intervals

Type	Interval	Period
<input type="button" value="Flexible"/> <input type="button" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/> <input type="button" value="Remove"/>

* Timeout

* History

* Trends

Value mapping

Populates host inventory field

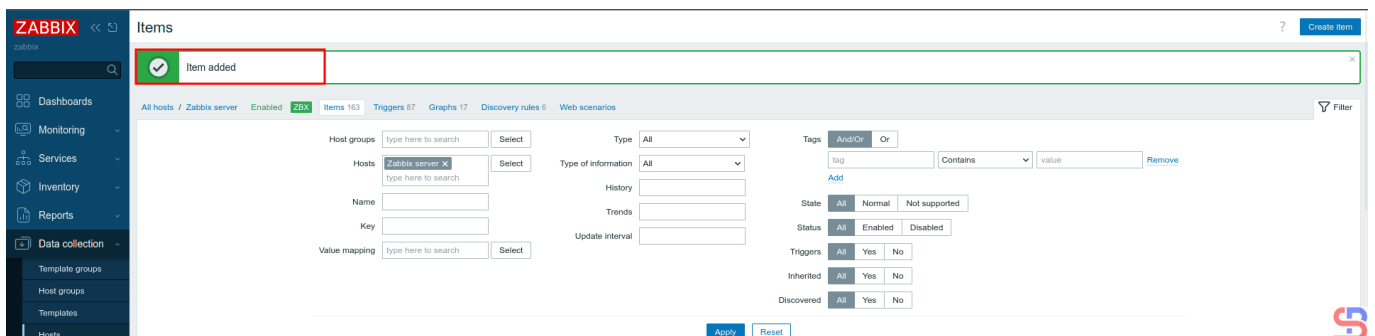
Description

Enabled



Click the Add button

After you press the Add button, there will be the text **Item added** as in the image below:

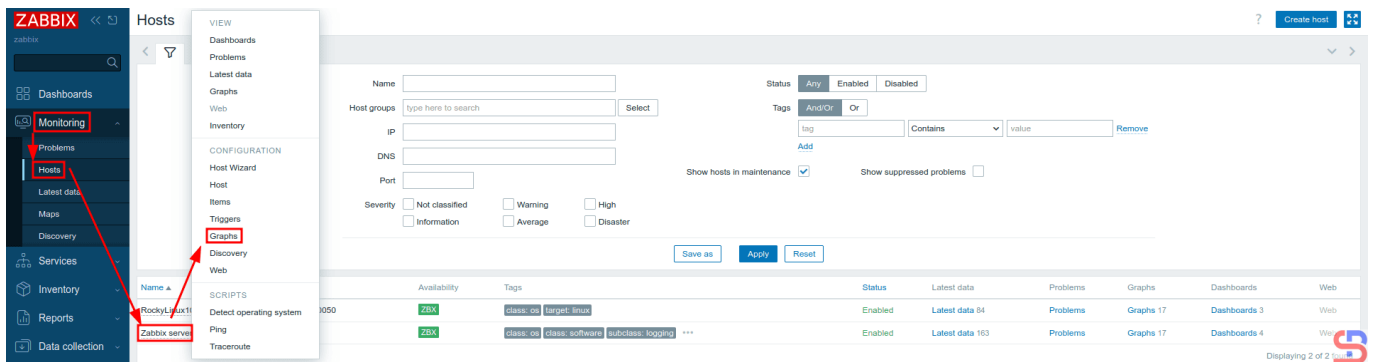


Succeed in adding an Item

6. Create a graph

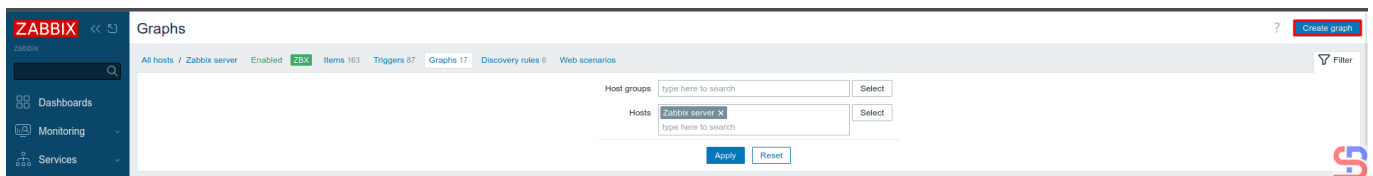
Then, create a graph for the result of the script by selecting the host that will display the result of the bash script. I choose to use the Zabbix server host: **Monitoring >**

Hosts > Zabbix server > Graphs in the **CONFIGURATION** section, like the image below:



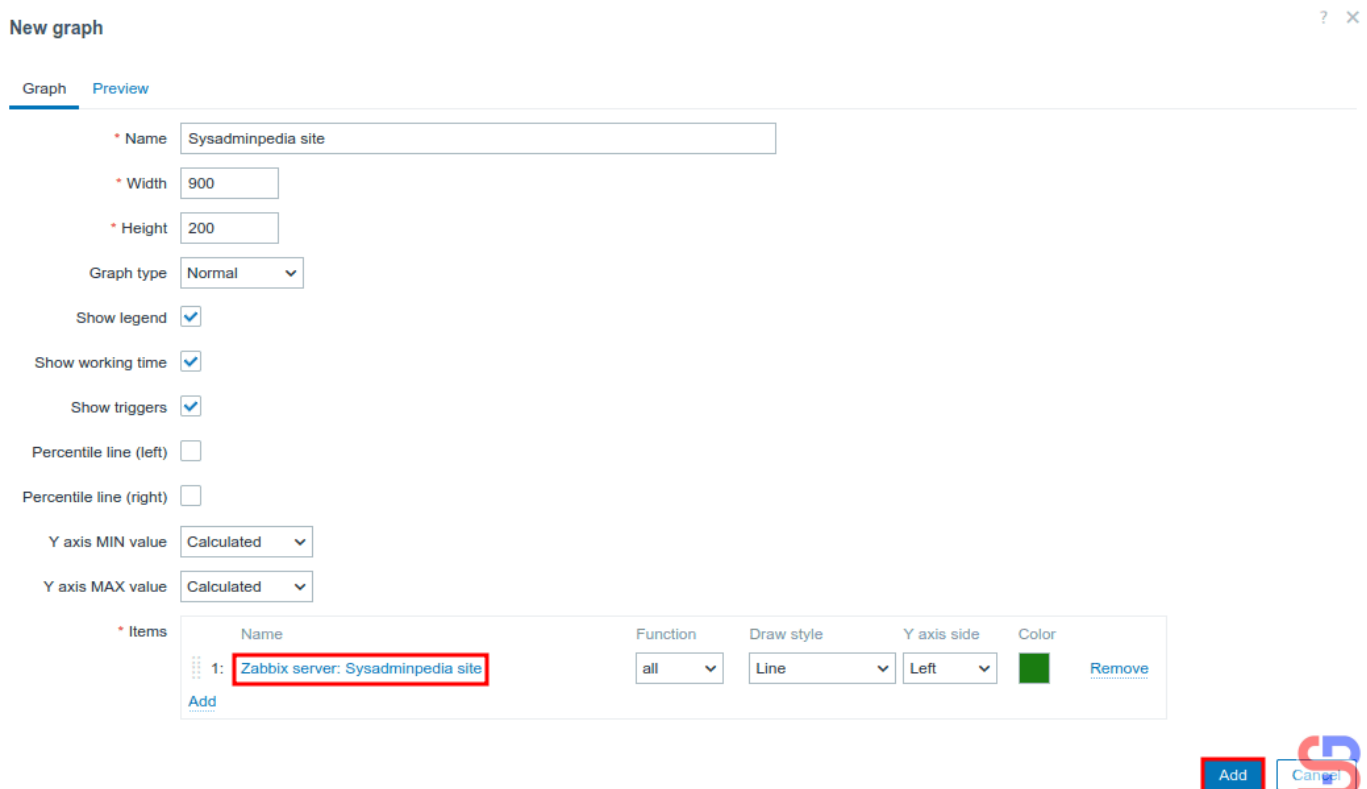
Click the **Graphs** in the **CONFIGURATION** section

And there will be a display as shown in the image below:



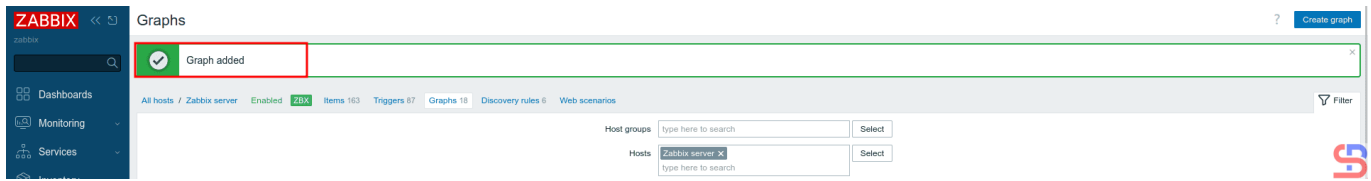
Click the **Create Graph** button

Click **Create Graph**, then there will be a display as below:



Click the **Add** button

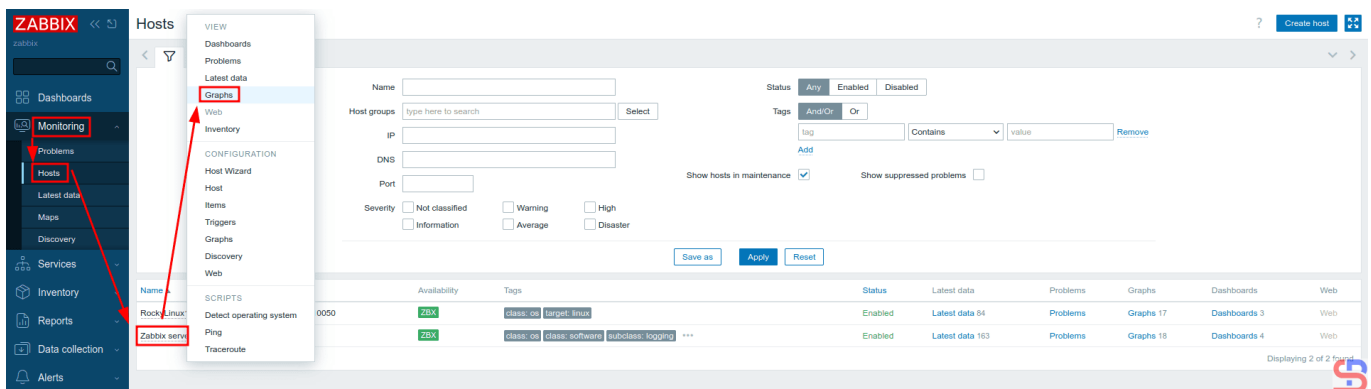
After that, click the **Add** button then there will be the text **Graph Added** as in the image below:



Succeed in adding a Graph

7. Display the graph

Wait a while, and to see the graph, you can go to **Monitoring > Hosts > Zabbix server > Graphs** in the **View** section, as shown in the image below:



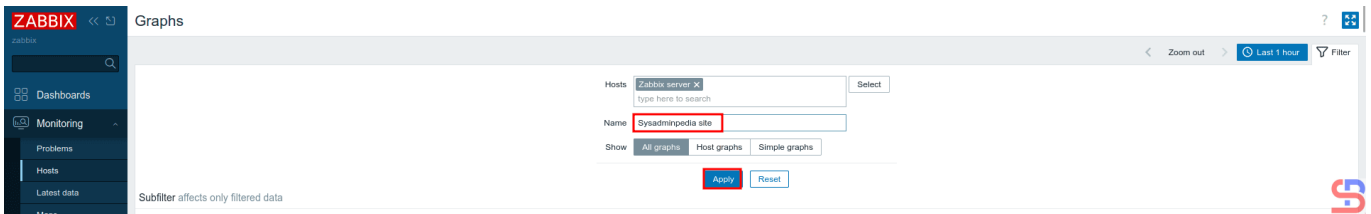
Click the Graphs in the VIEW section

Click the **Filter** button as shown in the image below:



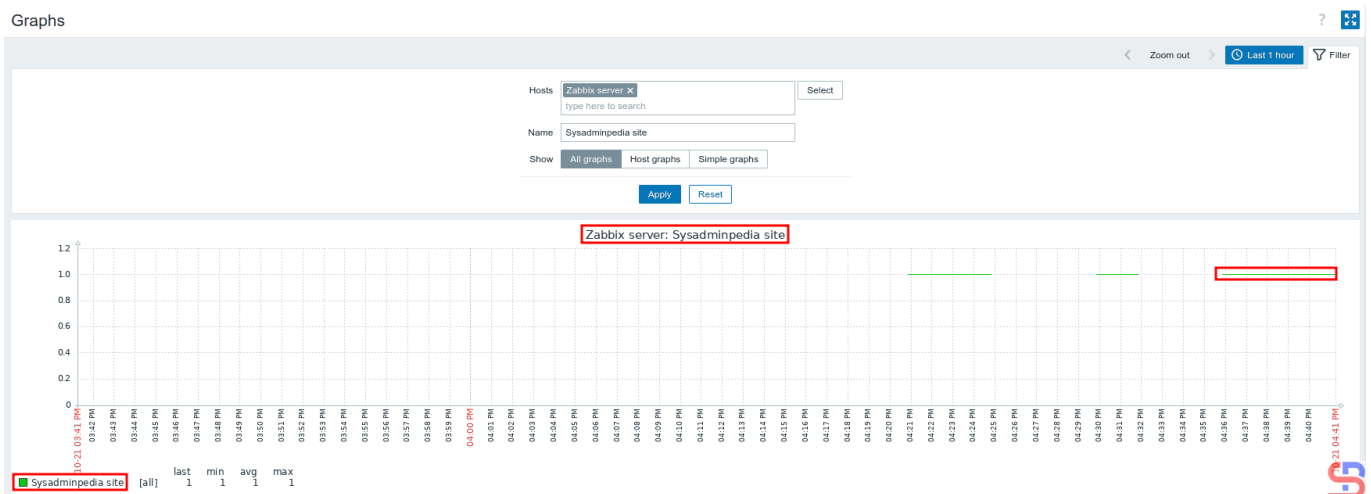
Click the Filter button

Then there will be a display as shown in the image below:



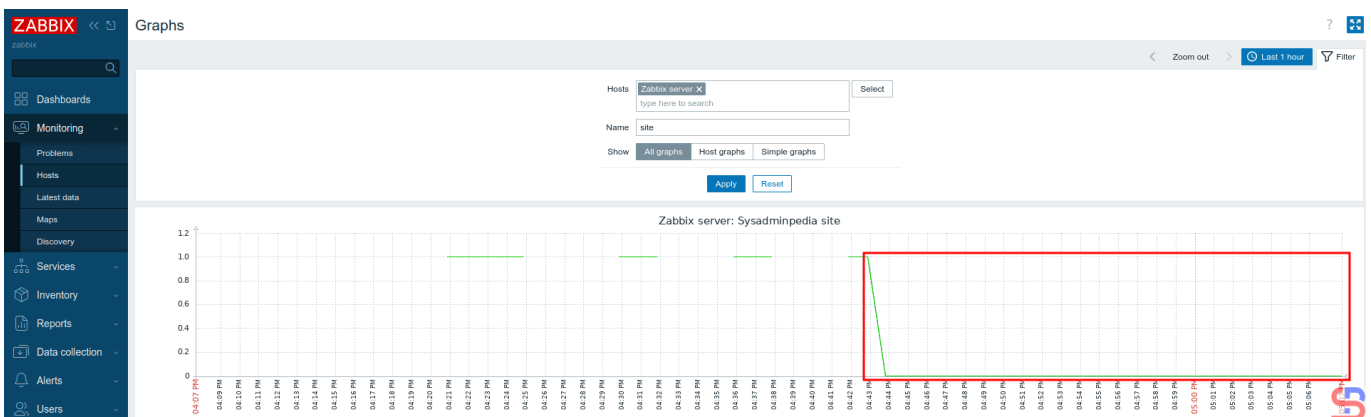
Type the name of the graph and click the Apply button

Type the name of the graph in the **Name** field, then click **Apply**, and then there should be a display below:



The graph from your script

If the site has an error, as known as there is no word wordpress, the graph will look as below:



When your script produces the error

And you successfully created a graph in Zabbix from the result of a script you made yourself.

Note

In this article, I used the Zabbix server to insert a script that monitors the server. However, you can use another host for your script, so you can do points 1 to 4 in the explanation above on another host.

References

blog.zabbix.com
youtube.com
sbcode.net

[How to Add a Linux Host to be Monitored by Zabbix?](#)

written by sysadmin | 14 January 2026

[The previous article](#) explained how to install the Zabbix application on Ubuntu. This article will explain how to add a Linux host to be monitored by Zabbix.

Problem

How to add a Linux host to be monitored by Zabbix?

Solution

This article will add a RockyLinux10 host, which will be monitored by Zabbix with IP 192.168.56.104, while the Zabbix server IP is 192.168.56.101. So that the host can be monitored by Zabbix, you must install the Zabbix-Agent on the host. Here are the steps:

A. On Remote Host

Check whether on your RockyLinux server, you have the file `/etc/yum.repos.d/epel.repo`. Don't worry if your server does not have the `epel.repo` file, but if the file exists on your server, you can add the script below:

```
excludepkgs=zabbix*
```

After that, run the commands below:

```
rpm -Uvh
https://repo.zabbix.com/zabbix/7.4/release/rocky/10/noarch/zabbix-release-latest-7.4.el10.noarch.rpm
dnf clean all
dnf install zabbix-agent -y
```

After you install the zabbix agent, go to copy the file `/etc/zabbix/zabbix_agentd.conf` as a backup:

```
cp /etc/zabbix/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf.ori
```

Then go into the file and change the **Server** section to your Zabbix server IP (which in this article is IP **192.168.56.101**), and in the **Hostname** section, you are free to fill in, and I changed it to `RockyLinux10`, so the file looks like the one below:

```
[root@RockyLinux10 ~]# grep -v "#" /etc/zabbix/zabbix_agentd.conf | grep -v '^$'
PidFile=/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=192.168.56.101
ServerActive=192.168.56.101
Hostname=RockyLinux10
Include=/etc/zabbix/zabbix_agentd.d/*.conf
[root@RockyLinux10 ~]#
```

The `zabbix_agentd.conf` file

If your RockyLinux server has a firewall, open port 10050 using the command:

```
firewall-cmd --permanent --zone=public --add-port=10050/tcp
firewall-cmd --reload
```

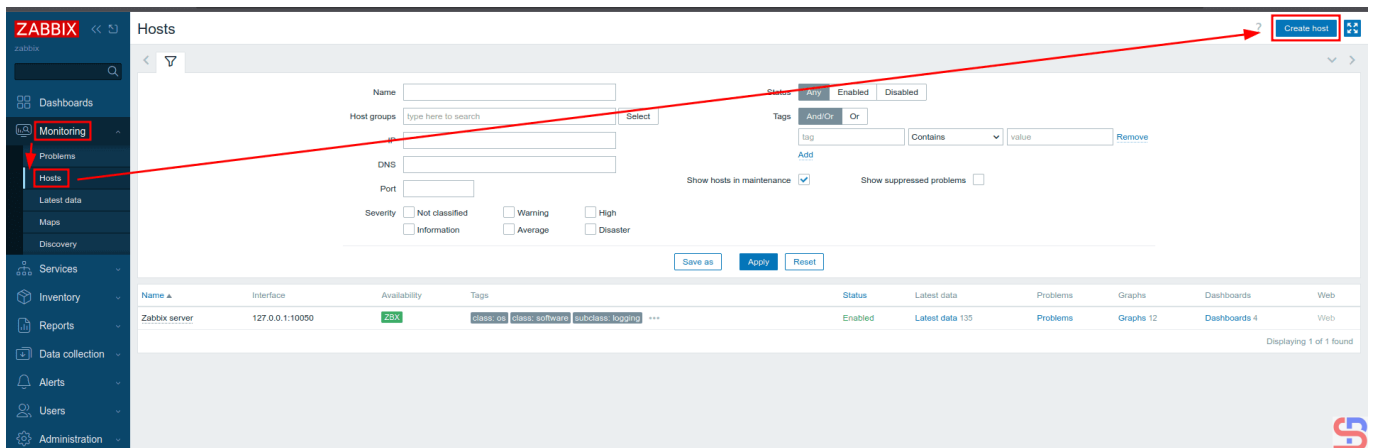
Then run the two commands below:

```
systemctl restart zabbix-agent
systemctl enable zabbix-agent
```

To view the log on zabbix-agent, open the file `/var/log/zabbix/zabbix_agentd.log` on your server.

B. On the Zabbix server

On the Zabbix server, enter the Zabbix application via your browser, then select **Monitoring > Hosts > Create Host** as in the image below:



Add the host to Zabbix

After that, there will be a display like below. You have to fill in the columns according to the host you will monitor. I filled them in as shown in the image below:

Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
Zabbix server	127.0.0.1:10000	ZBX	class: os class: software subclass: logging	Enabled	Latest data 135	Problems	Graphs 12	Dashboards 4	Web

New host

? X

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name

Visible name

Templates
type here to search

* Host groups
type here to search

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
<input type="text" value="Agent"/>	<input type="text" value="192.168.56.104"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="radio" value="IP"/> <input type="radio" value="DNS"/>	<input type="text" value="10050"/>	<input checked="" type="radio" value="Remove"/>

[Add](#)

Description

Monitored by

Enabled

Configure a new host in Zabbix

When finished, click the **Add** button, and you will see a display like the one below:

Hosts ?

Host added

Name

Host groups

IP

DNS

Port

Severity Not classified Warning High
 Information Average Disaster

Status

Tags

Show hosts in maintenance Show suppressed problems

Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
RockyLinux10	192.168.56.104:10050	ZBX	class: os target: linux	Enabled	Latest data 43	Problems	Graphs 8	Dashboards 3	Web
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software subclass: logging ***	Enabled	Latest data 135	Problems	Graphs 12	Dashboards 4	Web

Displaying 2 of 2 items

After the Host added

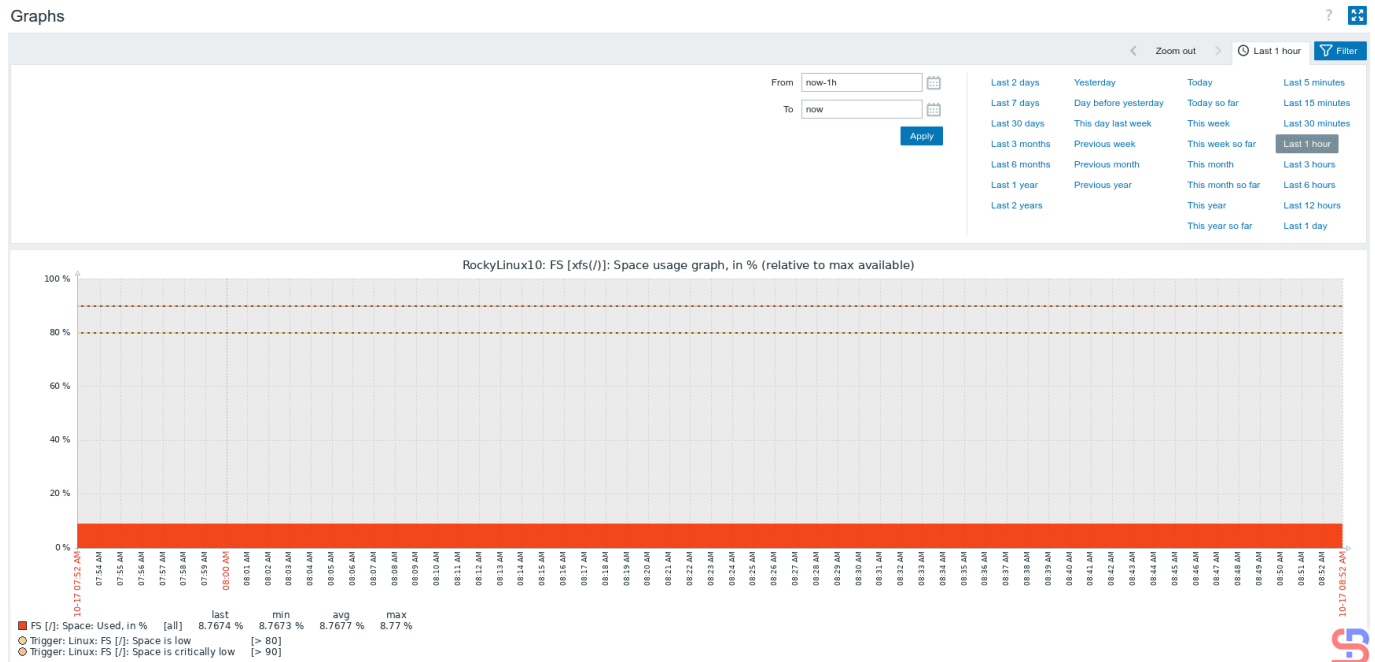
Wait a few moments, and the Zabbix application should be able to monitor your host, which is marked with the word **ZBX** in green, as in the image below:

Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
RockyLinux10	192.168.56.104:10050	ZBX	class: os target: linux	Enabled	Latest data 84	Problems	Graphs 17	Dashboards 3	Web
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software subclass: logging ***	Enabled	Latest data 162	Problems	Graphs 17	Dashboards 4	Web

Displaying 2 of 2 items

Zabbix monitors the host

To see the graph of the host, click on the words **Graphs**, so there will be a display like the one below:



The graphs of the host

And you have successfully added a host to the Zabbix application.

Note

If you want to add a host that Zabbix wants to monitor, you can go to [this address](#) to see the steps to install the Zabbix agent on your server. Make sure the Zabbix version selected is the same as the Zabbix version running on the server. The following is an example image for installing the Zabbix agent on the RockyLinux10 host, which is used as an example in this article:

1

Choose your platform

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
7.4	Alma Linux	10 (amd64, arm64)	Server, Frontend, Agent	---	---
7.2	Amazon Linux	9 (amd64, arm64)	Server, Frontend, Agent 2		
7.0 LTS	CentOS	8 (amd64, arm64)	Proxy		
6.0 LTS	Debian		Agent		
	OpenSUSE Leap		Agent 2		
	Oracle Linux		Java Gateway		
	Raspberry Pi OS		Web Service		
	Red Hat Enterprise Linux				
	Rocky Linux				
	SUSE Linux Enterprise Server				
	Ubuntu				

Release Notes 7.4



Choose the OS host to install Zabbix Agent

And don't forget to open Port 10050 on the host you want to monitor so that the Zabbix application can access that host.

References

tecadmin.net

zabbix.com

bestmonitoringtools.com