

How to Backup And Restore Uptime Kuma Database in Docker?

written by sysadmin | 7 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 \
  mysqldump -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 \
  mysqldump -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 Protect phpMyAdmin Using Nginx?

written by sysadmin | 7 January 2026

[The previous article](#) explained how to install phpMyAdmin with the nginx web server. This article will explain how to protect phpMyAdmin from unauthorized users using nginx.

Problem

How to protect phpMyAdmin using nginx?

Solution

There are several methods to protect phpMyAdmin using nginx:

1. Allowing certain IPs

The phpMyAdmin application can only be accessed by users who have certain IP addresses. For example, you want the IP localhost, and only 192.168.56.1 to be able to access phpMyAdmin. Then add the script below to the **/etc/nginx/sites-available/default** file in the **location /phpmyadmin** section:

```
allow 127.0.0.1;  
allow 192.168.56.1;  
deny all;
```

For more details, take a look at the image below:

```

location /phpmyadmin {
    root /usr/share/;
    index index.php;

    allow 127.0.0.1;
    allow 192.168.56.1;
    deny all;

    location ~ ^/phpmyadmin/(.+\.php)$ {
        try_files $uri =404;
        root /usr/share/;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~* ^/phpmyadmin/(.+\. (css|js|jpg|jpeg|gif|png|ico|html|xml|txt))$ {
        root /usr/share/;
    }
}

```

Allowing certain IPS

After that, use the command below to reload nginx:

```

sudo nginx -t
sudo systemctl reload nginx

```

If any user who uses an IP other than the localhost and 192.168.56.1 wants to access phpMyAdmin, then that user will not be able to access phpMyAdmin, as shown in the image below:

 192.168.56.2/phpmyadmin/

403 Forbidden

nginx/1.24.0 (Ubuntu)



Forbidden access

2. Add a password

To make it safer, phpMyAdmin should be given additional HTTP Auth so that users who want to access the application must enter a password. Use the command below to install HTTP auth:

```
sudo apt install apache2-utils  
sudo htpasswd -c /etc/nginx/.phpmyadmin admin
```

Enter the password that you want, and then in the **/etc/nginx/sites-available/default** file, add the script below:

```
auth_basic 'Restricted';  
auth_basic_user_file /etc/nginx/.phpmyadmin;
```

So the default file will look like the image below:

```
location /phpmyadmin {
    root /usr/share/;
    index index.php;

    allow 127.0.0.1;
    allow 192.168.56.1;
    deny all;

    auth_basic "Restricted";
    auth_basic_user_file /etc/nginx/.phpmyadmin;

    location ~ ^/phpmyadmin/(.+\.php)$ {
        try_files $uri =404;
        root /usr/share/;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~* ^/phpmyadmin/(.+\. (css|js|jpg|jpeg|gif|png|ico|html|xml|txt))$ {
        root /usr/share/;
    }
}
```

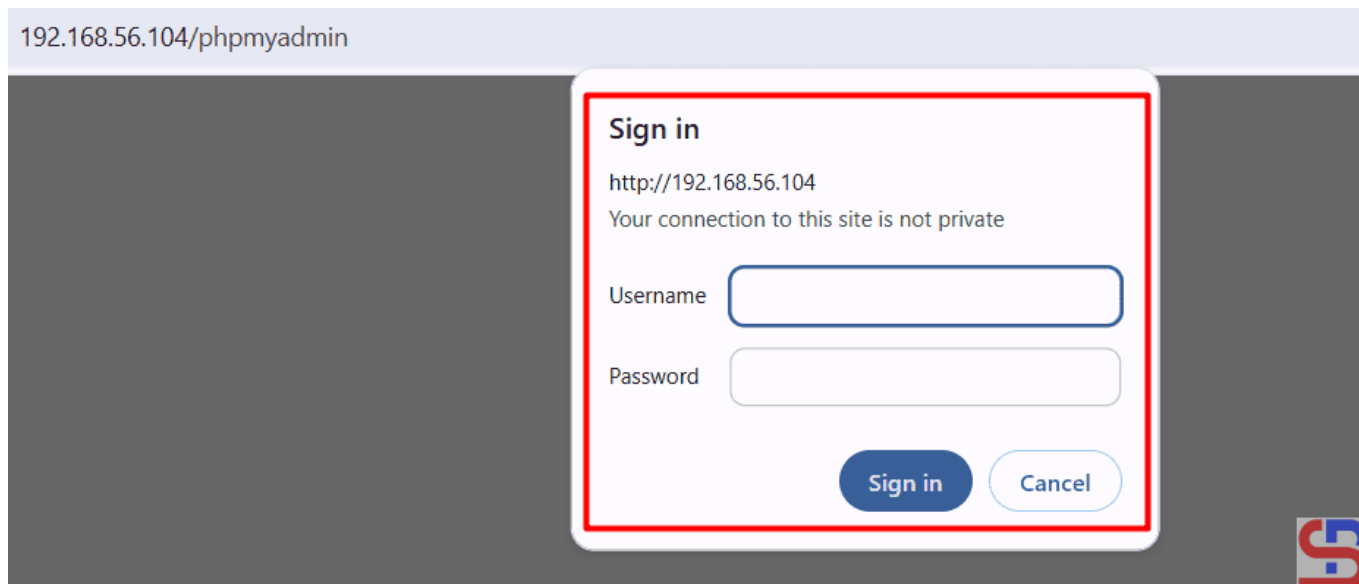


Adding HTTP Auth in Nginx

After that, use the command below to reload nginx:

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

Open the browser, and when you access phpMyAdmin, it should be there should be a display like below:



Enter username and password when accessing phpMyAdmin

Enter the username: **admin** and the password you created previously. If there are no errors, you can access phpMyAdmin.

3. Change the URL

By default, if you want to access phpMyAdmin, then you type the command below:

```
http://ip_server/phpmyadmin
```

However, for security reasons, it is best to replace the word phpMyAdmin with another word, for example, **pma**, so that the site address changes to:

```
http://ip_server/pma
```

Therefore, in the default file, change the file by deleting the **/phpmyadmin** section with the script below:

```
location /pma {
    alias /usr/share/phpmyadmin/;
    index index.php;

    allow 127.0.0.1;
```

```
allow 192.168.56.1;
deny all;

auth_basic "Restricted";
auth_basic_user_file /etc/nginx/.phpmyadmin;

location ~ /\.php$ {
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/run/php/php8.3-fpm.sock;
fastcgi_param SCRIPT_FILENAME $request_filename;
}

location ~* \.(css|js|jpg|jpeg|gif|png|ico|html|xml|txt)$ {
expires 30d;
access_log off;
}
}
```

so that the default file changes to look like the image below:

```

server {
    listen 80;
    server_name _;
    root /var/www/html;
    index index.php index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    location /pma {
        alias /usr/share/phpmyadmin/;
        index index.php;

        allow 127.0.0.1;
        allow 192.168.56.1;
        deny all;

        auth_basic "Restricted";
        auth_basic_user_file /etc/nginx/.phpmyadmin;

        location ~ /\.php$ {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.3-fpm.sock;
            fastcgi_param SCRIPT_FILENAME $request_filename;
        }

        location ~* \.(css|js|jpg|jpeg|gif|png|ico|html|xml|txt)$ {
            expires 30d;
            access_log off;
        }
    }
}

```

Change the URL in Nginx

Use the command below to reload nginx:

```

sudo nginx -t
sudo systemctl reload nginx

```

Open **http://ip_server/pma** in your browser, then you should be able to access phpMyAdmin as in the image below:



phpMyAdmin
Welcome to phpMyAdmin

Language

English ▼

Log in ⓘ

Username:

Password:

Log in



Change the URL

Note

There is one more method so that your phpMyAdmin application can be secure, namely, using SSL. You can use a Let's Encrypt SSL certificate for your phpMyAdmin site because the certificate is free. However, if you want the phpmyadmin application not to be accessed by the public, I think, then there is no need to use SSL.

References

digitalocean.com
serverfault.com
httpd.apache.org

How to Install phpMyAdmin With Nginx on Ubuntu?

written by sysadmin | 7 January 2026

[The previous article](#) has explained how to install the phpMyAdmin application on Linux using the Apache web server. This article explains how to configure phpMyAdmin using nginx on Ubuntu.

Problem

How to install phpMyAdmin with nginx on Ubuntu?

Solution

Follow the steps below to install phpMyAdmin with Nginx on Ubuntu:

1. Update repo

Use the command below to update the Ubuntu repo:

```
sudo apt update -y
```

2. Install MariaDB

Next, install the MariaDB database using the command:

```
sudo apt install mariadb-server mariadb-client -y
```

Once finished, use the command below to change the root password in MariaDB:

```
sudo mysql_secure_installation
```

```
Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```



Change the root password

Then check whether the database is up or not using the command below:

```
sudo systemctl status mariadb
```

3. Install PHP

Install PHP by using the command below:

```
sudo apt install php php-fpm php-mysql php-cli php-curl php-gd php-mbstring
php-xml php-zip -y
```

Then check the version of PHP that you just installed by using the command below:

```
php -v
```

```
sysadmin@ubuntu24:~$ php -v
PHP 8.3.6 (cli) (built: Jul 14 2025 18:30:55) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
sysadmin@ubuntu24:~$
```



Check the php version

Usually, when installing PHP, the Apache package will also be installed on the server. Therefore, delete Apache using the command:

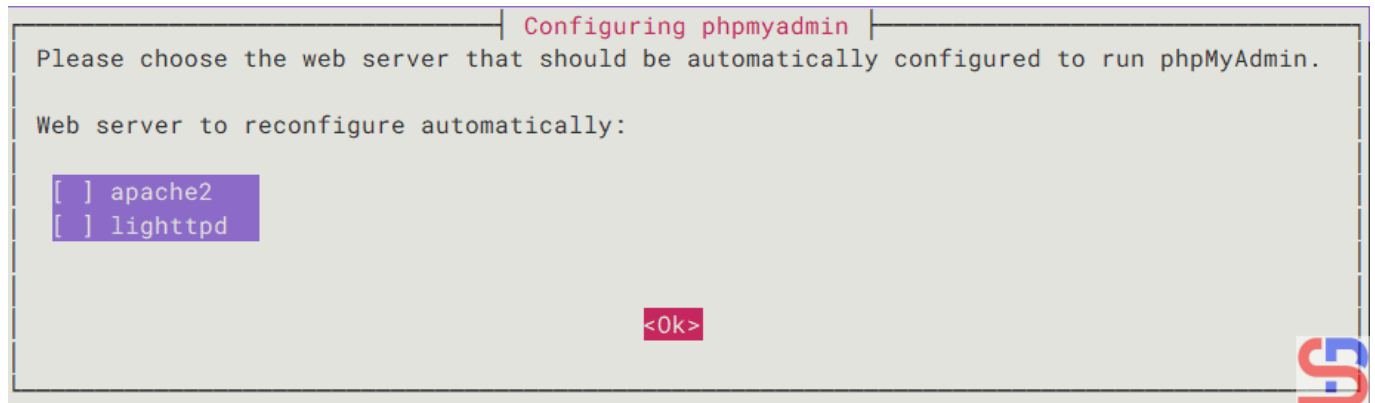
```
sudo apt remove apache2-* -y
```

4. Install phpMyAdmin

Use the command below to install phpMyAdmin:

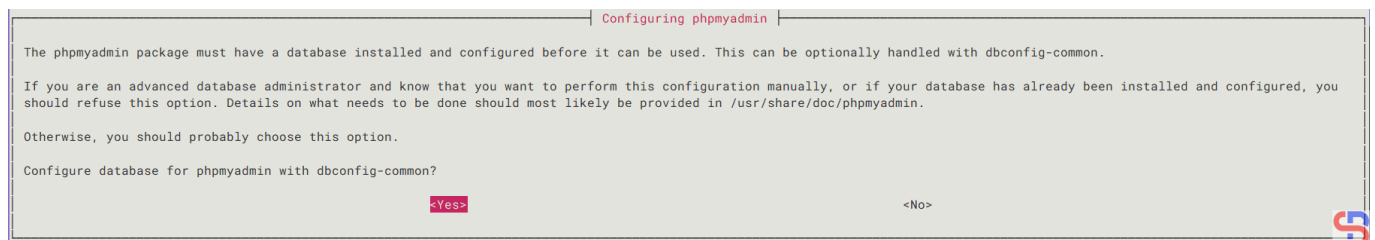
```
sudo apt install phpmyadmin -y
```

At the time of installation, there are several pop-ups that you must answer, such as the selection of the web server you are using, as shown in the image below:



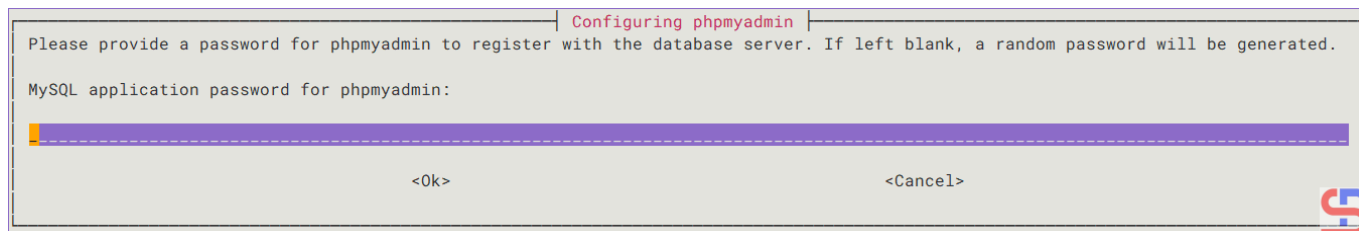
Choose the Ok button

Just select the button **Ok**, then the process of installation will continue. A few seconds later, there will be a pop-up as below to insert the phpMyAdmin in the database:



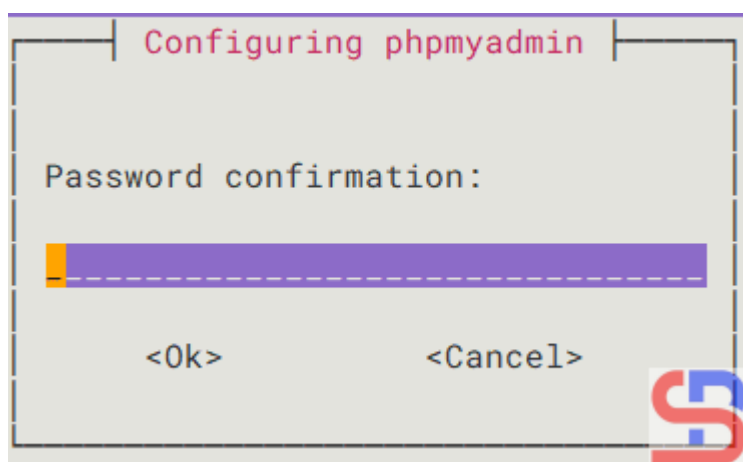
Choose the Yes button

Select the button **Yes**, and there's a pop-up to enter the password for the user phpmyadmin in the database as in the picture below:



Enter the password for the phpmyadmin user

Enter the password you want, select the **OK** button, and there will be another pop-up to confirm the password as in the image below:



Password confirmation

Enter the same password and select the **Ok** button, then the phpMyAdmin installation process will continue until completion.

5. Install nginx

Install nginx by using the command below:

```
sudo apt install nginx -y
```

After that, configure Nginx so that it can be integrated with phpMyAdmin. Copy the default file using the command below:

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/default.ori
```

Then in the default file, copy the script below:

```
server {
    listen 80;
    server_name _;
    root /var/www/html;
    index index.php index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    location /phpmyadmin {
        root /usr/share/;
        index index.php;

        location ~ ^/phpmyadmin/(.+\.php)$ {
            try_files $uri =404;
            root /usr/share/;
            fastcgi_pass unix:/run/php/php8.3-fpm.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            include fastcgi_params;
        }

        location ~*
^/phpmyadmin/(.+\. (css|js|jpg|jpeg|gif|png|ico|html|xml|txt))$ {
            root /usr/share/;
        }
    }
}
```

Warning

You have to be careful when writing the php-fpm version in the **fastcgi_pass_unix** section because there will be an error if the version is different from the one installed on the server. To see the version installed on the server, use the command below:

```
ls -l /run/php/
```

After that, use the command below to check whether the nginx configuration has errors or not:

```
sudo nginx -t
```

If there are no errors, then you can run the command below to reload nginx:

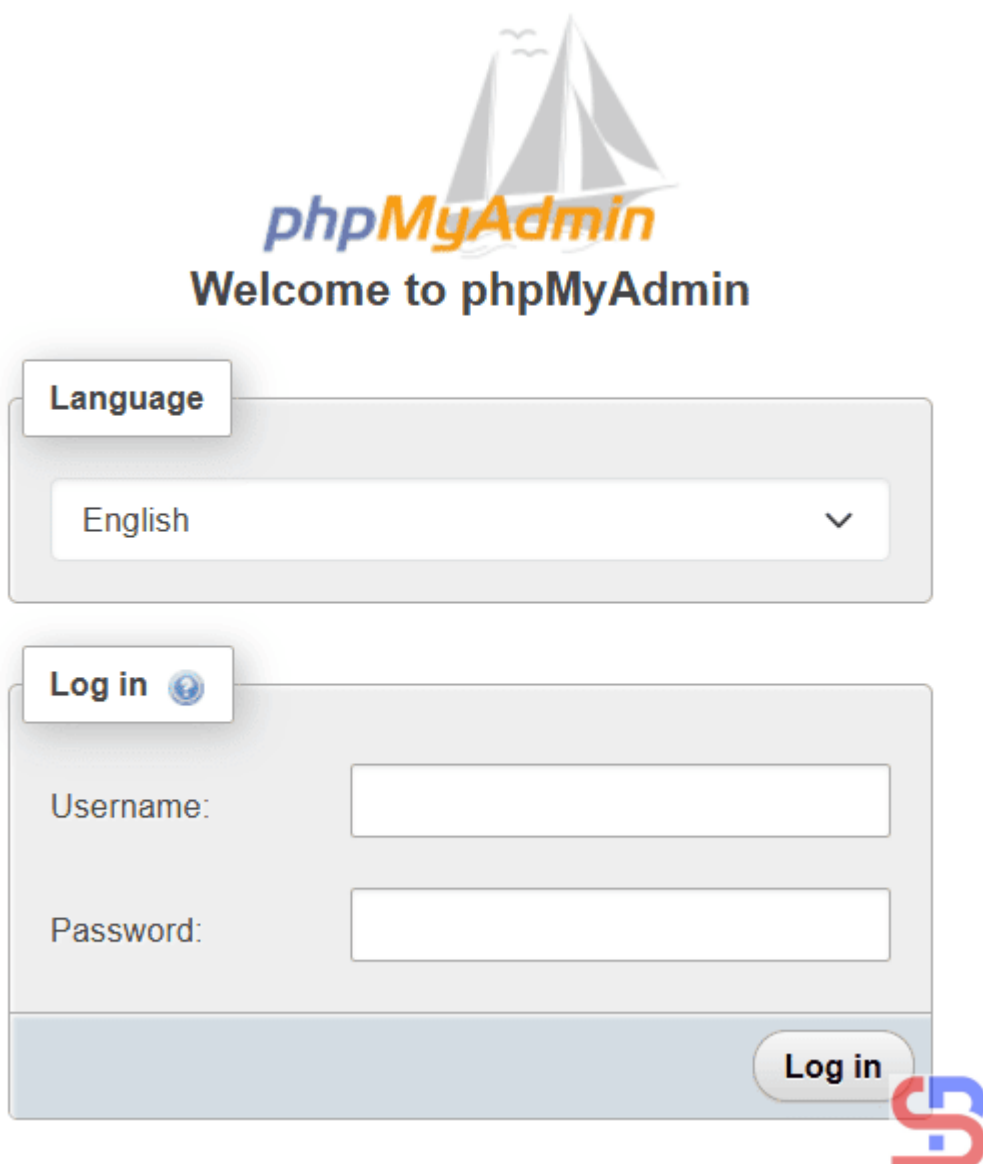
```
sudo systemctl reload nginx
```

6. Open phpMyAdmin

Open your browser and type:

```
http://ip_server/phpmyadmin
```

Then there should be a display like below:



The image shows the phpMyAdmin welcome screen. At the top, there is a logo of a sailboat with the text "phpMyAdmin" in blue and orange, and "Welcome to phpMyAdmin" below it. Below the logo, there is a "Language" section with a dropdown menu showing "English". Below that, there is a "Log in" section with a "Log in" button and a "Log in" link. The "Log in" section contains two input fields: "Username:" and "Password:". A blue and red logo is visible in the bottom right corner.

Language

English

Log in

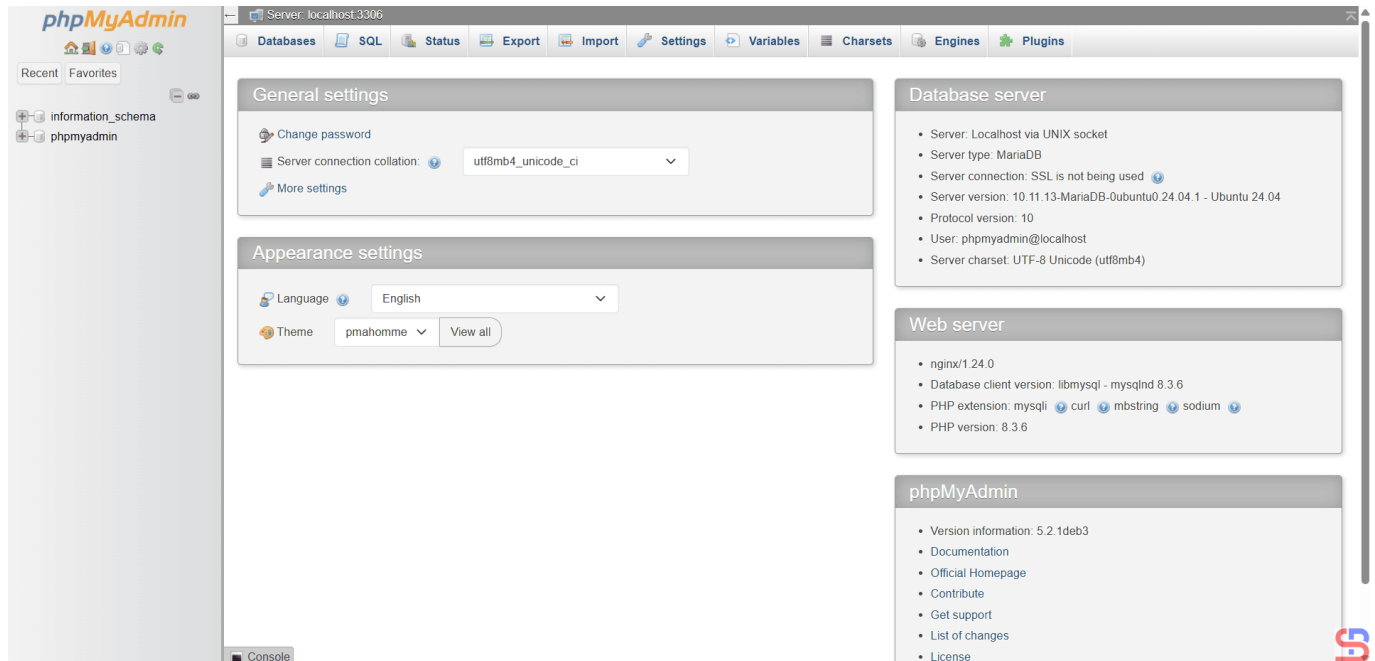
Username:

Password:

Log in

Open phpMyAdmin in the browser

Enter the database username and password, for example, using the user phpmyadmin with the password that you created when installing phpMyAdmin, then there will be a display like the one below:



Display of phpMyAdmin using phpmyadmin user

Note

If you want the phpMyAdmin application to be more secure, you can look at [this page](#).

References

markaicode.com

linuxbabe.com

hostman.com

How to Configure Crontab in Linux?

written by sysadmin | 7 January 2026

As a sysadmin, Crontab is an important tool for running scripts that you want to run at a certain time.

Problem

How to configure crontab in Linux?

Solution

Cron is a command you can use in the shell to set up a task, like a command or a script, to run automatically at certain times, dates, or intervals. It was made by AT&T Bell Laboratories and first came out in May 1975.

A. Format crontab

Cron works based on a crontab file, which tells it what tasks to run and when. Crontab has 2 sections: a time section that has 5 items, where each item has a different parameter, and the command section to be executed. For more details, see the image below:

🎯 A crontab file has five fields for specifying:

```

* * * * *    command to be executed
- - - - -
| | | | |
| | | | +----- **DAY OF WEEK** (0-6) (Sunday=0)
| | | +----- **MONTH** (1-12)
| | +----- **DAY OF MONTH** (1-31)
| +----- **HOUR** (0-23)
+--- **MINUTE** (0-59)
```

The syntax of the crontab file (Credit to blog.marcotorres.pe)

B. Crontab commands

To display the contents of the crontab, use the command below:

```
crontab -l
```

To display the crontab for a user, for example, john, use the command below:

```
crontab -u john -l
```

To create or modify a crontab file, use the command below:

```
crontab -e
```

To delete the crontab file, use the command below:

```
crontab -r
```

C. Crontab examples

Here are examples of Crontab to run time.sh file that contains as below, and don't forget to permit (**chmod +x**) so that the file can be run:

```
#!/bin/bash
#
time=`date +"%Y%m%d-%H:%M:%S"`
echo $time >> time.txt
```

1. Once a week

If you want to run a file once a week, you can use the crontab configuration as below:

```
@weekly          cd /home/sysadmin;./time.sh
```

2. Every time you reboot

Use the crontab configuration below if you want to run a file every time the server reboot is completed:

```
@reboot          cd /home/sysadmin;./time.sh
```

3. Every 5 minutes from 1 to 7

Use the crontab configuration below if you want to run a file every 5 minutes from 1 to 7 (i.e., 01:00, 01:05, 01:10, up until 07:55):

```
*/5 * * * *      cd /home/sysadmin;./time.sh
```

4. Every 10:30 on 1,10,20,30

If you run a file every 10:30 on 1,10,20,30, use the crontab configuration as below:

```
30 10 1,10,20,30 * *      cd /home/sysadmin;./time.sh
```

5. Every first Monday of every month, at 7 a.m.

Use the crontab configuration below if you want to run a file every first Monday of every month, at 7 a.m:

```
0 7 1-7 * 1      cd /home/sysadmin;./time.sh
```

6. Every 15 minutes after rebooting

If you run a file every 15 minutes after rebooting, use the crontab configuration as below:

```
@reboot sleep 900 &&      cd /home/sysadmin;./waktu.sh
```

7. Every last Saturday of every month

If you want to run a file every last Saturday of every month, then you have to create a script file first example, last_saturday.sh, as below:

```
cat sabtu.sh
#!/bin/bash
```

```
TODAY=$(date +%Y-%m-%d)
NEXT_SATURDAY_MONTH=$(date -d "next Saturday" +%m)
CURRENT_MONTH=$(date +%m)

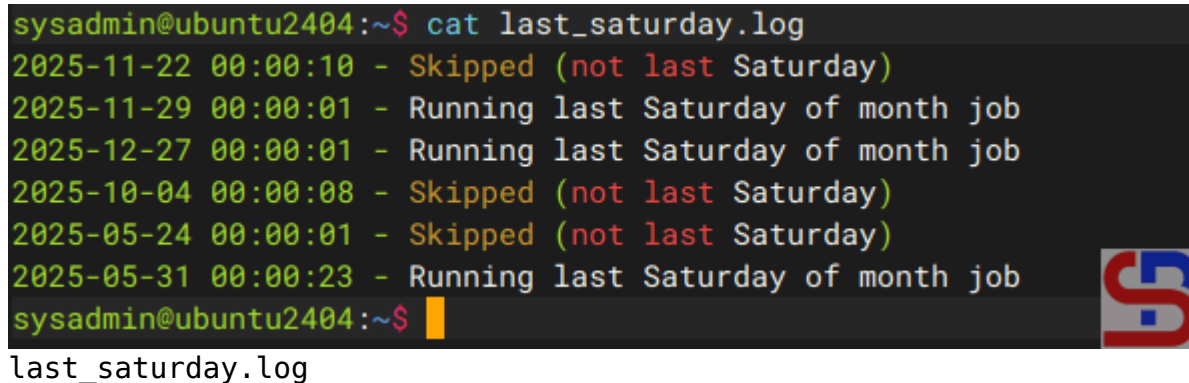
# If the next Saturday is in the next month,
# it means that this Saturday is the last Saturday of the month
if [ "$NEXT_SATURDAY_MONTH" != "$CURRENT_MONTH" ]; then
    echo "$(date '+%Y-%m-%d %H:%M:%S') - Running last Saturday of month job"
    >> /home/sysadmin/last_saturday.log
    /home/sysadmin/time.sh
else
    echo "$(date '+%Y-%m-%d %H:%M:%S') - Skipped (not last Saturday)" >>
/home/sysadmin/last_saturday.log
fi
```

And in the crontab, config like below:

```
0 0 * * 6      cd /home/sysadmin;./last_saturday.sh
```

This script will only work on Saturdays in each month, and if there is a Saturday in the following month, then this script will not run. To see the log for this script, go to /home/sysadmin/last_saturday.log, and here is a sample of the log:

```
sysadmin@ubuntu2404:~$ cat last_saturday.log
2025-11-22 00:00:10 - Skipped (not last Saturday)
2025-11-29 00:00:01 - Running last Saturday of month job
2025-12-27 00:00:01 - Running last Saturday of month job
2025-10-04 00:00:08 - Skipped (not last Saturday)
2025-05-24 00:00:01 - Skipped (not last Saturday)
2025-05-31 00:00:23 - Running last Saturday of month job
sysadmin@ubuntu2404:~$
```



Note

For crontab to run properly, use absolute paths for files and commands. And don't forget to make sure the script can be executed and preferably in the script file, to include logs so that it can be traced if there are errors. To see the log in Linux whether crontab is running or not on Ubuntu/Debian, you can use the command below:

```
sudo grep CRON /var/log/syslog
```

Use the command below if you are using RHEL/RockyLinux/AlmaLinux:

```
sudo grep CRON /var/log/cron
```

If you can't find the cron log in the file, then open the file /etc/rsyslog.d/50-default.conf and search for the word cron. After that, remove the comment mark # behind the cron word. Then restart the service using the command:

```
sudo systemctl restart rsyslog
```

References

en.wikipedia.org
crontab.guru
codepolitan.com
askubuntu.com
medium.com

How to Install Zabbix On Ubuntu?

written by sysadmin | 7 January 2026

Zabbix is an open-source software tool to monitor IT infrastructure such as networks, servers, virtual machines, and cloud services.

Problem

How to install Zabbix in Ubuntu?

Solution

Zabbix was first released in 2001, and as of this writing in October 2025, Zabbix has version 7.4. This article will explain how to install Zabbix on an Ubuntu server by using MariaDB and Apache databases.

A. Install Zabbix

Run the commands below to install Zabbix on Ubuntu:

```
wget  
https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/  
zabbix-release_latest_7.4+ubuntu24.04_all.deb  
sudo dpkg -i zabbix-release_latest_7.4+ubuntu24.04_all.deb  
sudo apt update  
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf  
zabbix-sql-scripts zabbix-agent
```

B. Database Configuration

If your Ubuntu doesn't have a database, then you can use the MariaDB database by using the command:

```
sudo apt install mariadb-server
```

Then, create a password for root in MariaDB using the command:

```
sudo mariadb-secure-installation
```

After that, enter MariaDB using the command:

```
sudo mariadb -uroot -p
```

Run the commands below (change the **password** to what you want):

```
create database zabbix character set utf8mb4 collate utf8mb4_bin;  
create user zabbix@localhost identified by 'password';  
grant all privileges on zabbix.* to zabbix@localhost;  
set global log_bin_trust_function_creators = 1;  
quit;
```

Run the command below to import the initial schema and data, and enter the password you created when you created the Zabbix database in MariaDB:

```
zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix
```

Then log in to MariaDB again using the command:

```
sudo mariadb -uroot -p
```

Run the command below to disable the `log_bin_trust_function_creators` option after importing the

database schema.

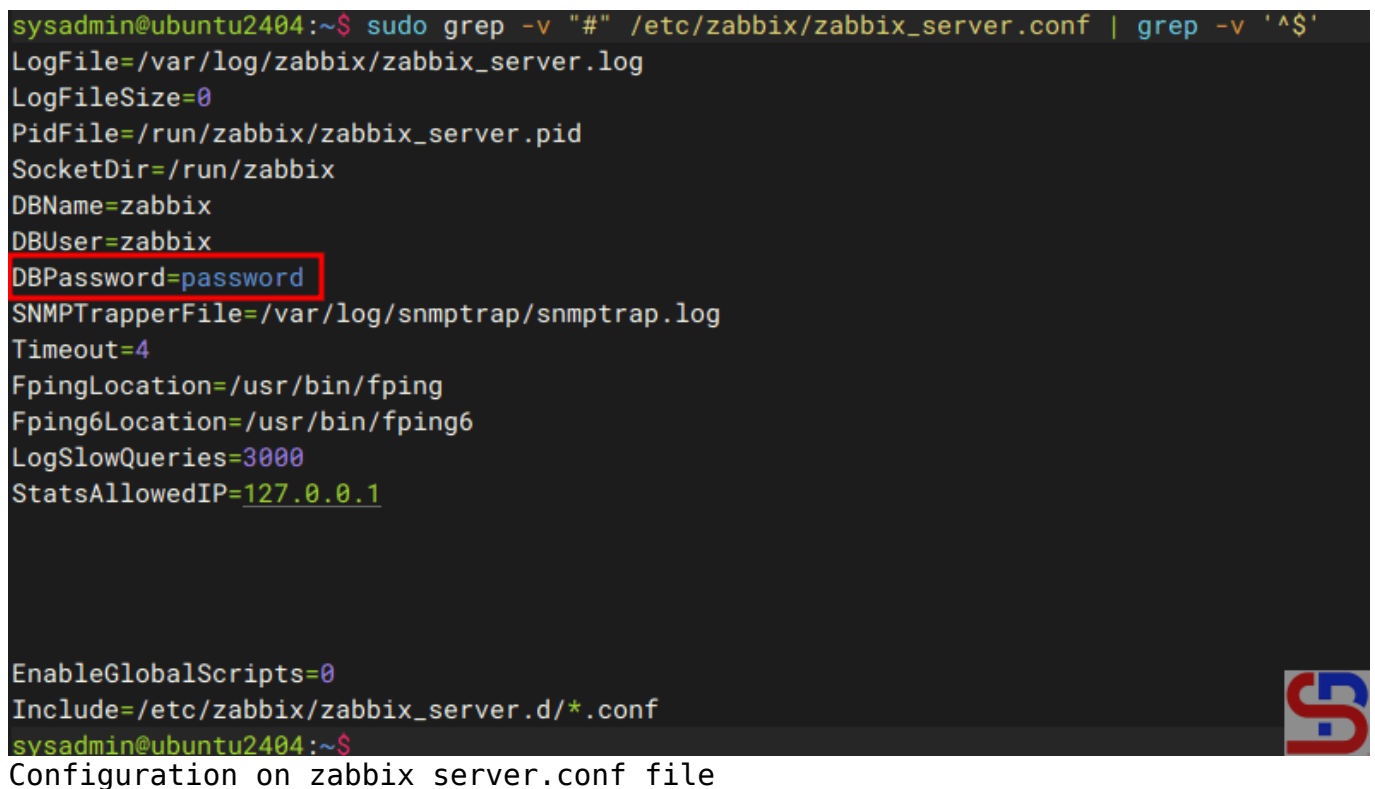
```
set global log_bin_trust_function_creators = 0;
quit;
```

C. Configure the Zabbix file

After that, you will configure the zabbix file located in **/etc/zabbix/zabbix_server.conf**. It's better if you copy the original file as a backup by running the command below:

```
sudo cp /etc/zabbix/zabbix_server.conf /etc/zabbix/zabbix_server.conf.ori
```

Fill in the DBPassword section of the file with the password you created for the Zabbix user, so that it is as follows:



```
sysadmin@ubuntu2404:~$ sudo grep -v "#" /etc/zabbix/zabbix_server.conf | grep -v '^$'
LogFile=/var/log/zabbix/zabbix_server.log
LogFileSize=0
PidFile=/run/zabbix/zabbix_server.pid
SocketDir=/run/zabbix
DBName=zabbix
DBUser=zabbix
DBPassword=password
SNMPTrapperFile=/var/log/snmptrap/snmptrap.log
Timeout=4
FpingLocation=/usr/bin/fping
Fping6Location=/usr/bin/fping6
LogSlowQueries=3000
StatsAllowedIP=127.0.0.1

EnableGlobalScripts=0
Include=/etc/zabbix/zabbix_server.d/*.conf
sysadmin@ubuntu2404:~$
```

Configuration on zabbix_server.conf file

Then run the two commands below:

```
systemctl restart zabbix-server zabbix-agent apache2
systemctl enable zabbix-server zabbix-agent apache2
```

D. Configure Zabbix

Open your browser and type in the URL below:

`http://your_ip_server/zabbix`

Then there will be a display like the image below:



Configure Zabbix using your browser

Click the **Next step** button, and a display similar to the picture below will be present:

Check of pre-requisites

Welcome
Check of pre-requisites
Configure DB connection
Settings
Pre-installation summary
Install

	Current value	Required	
PHP version	8.3.6	8.0.0	OK
PHP option "memory_limit"	128M	128M	OK
PHP option "post_max_size"	16M	16M	OK
PHP option "upload_max_filesize"	2M	2M	OK
PHP option "max_execution_time"	300	300	OK
PHP option "max_input_time"	300	300	OK
PHP databases support	MySQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK
PHP option "mbstring.func_overload"	off	off	OK

[Back](#)[Next step](#)

Checking of pre-requisites

Make sure there is no error like in the image above. After that, click the **Next step** button, and there will be a screen similar to the one below:

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Welcome
Check of pre-requisites
Configure DB connection
Settings
Pre-installation summary
Install

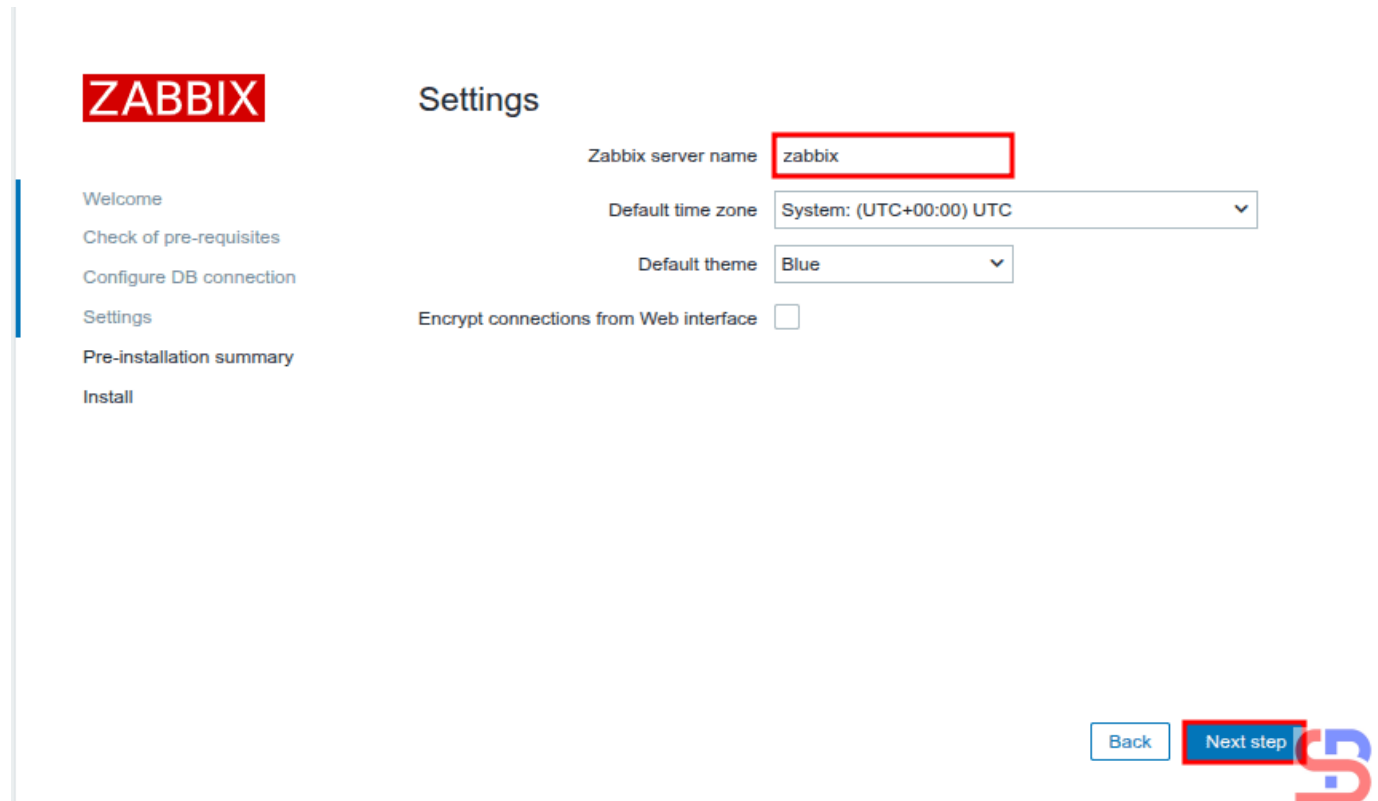
Database type Database host Database port 0 - use default portDatabase name Store credentials in ☒ Plain text ☐ HashiCorp Vault ☐ CyberArk VaultUser Password

Database TLS encryption *Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).*

[Back](#)[Next step](#)

Enter the password of MariaDB

Enter your database password using the Zabbix user, click the **Next step** button, and a screen similar to the one below will be presented:



The screenshot shows the Zabbix installation 'Settings' page. On the left is a sidebar with the ZABBIX logo and a list of steps: Welcome, Check of pre-requisites, Configure DB connection, Settings (highlighted), Pre-installation summary, and Install. The main content area is titled 'Settings' and contains three configuration items: 'Zabbix server name' with a text input field containing 'zabbix' (highlighted with a red box), 'Default time zone' with a dropdown menu showing 'System: (UTC+00:00) UTC', and 'Default theme' with a dropdown menu showing 'Blue'. Below these is a checkbox for 'Encrypt connections from Web interface' which is currently unchecked. At the bottom right, there are two buttons: 'Back' and 'Next step' (highlighted with a red box). A large, stylized red and blue 'S' logo is positioned to the right of the 'Next step' button.

Enter the server name of Zabbix

Enter the name of the Zabbix server you want, click the **Next step** button, and there will be a display like the image below:



Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Database type **MySQL**

Database server **localhost**

Database port **default**

Database name **zabbix**

Database user **zabbix**

Database password *********

Database TLS encryption **false**

Zabbix server name **zabbix**

Encrypt connections from Web interface **false**

Back

Next step



Pre-installation summary

Click the **Next step** button, and there will be a display similar to the image shown below.



Install

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "conf/zabbix.conf.php" created.

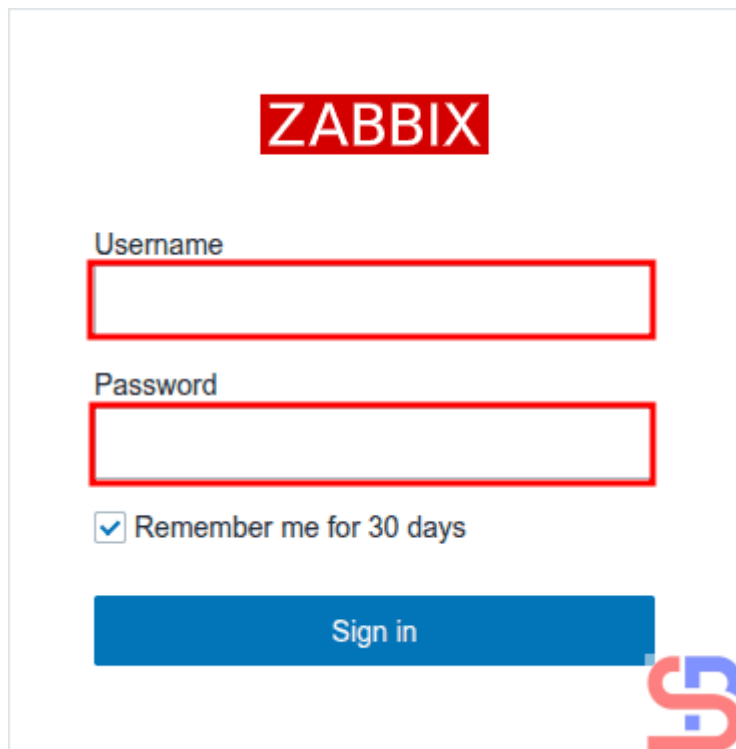
Back

Finish



Finish installation

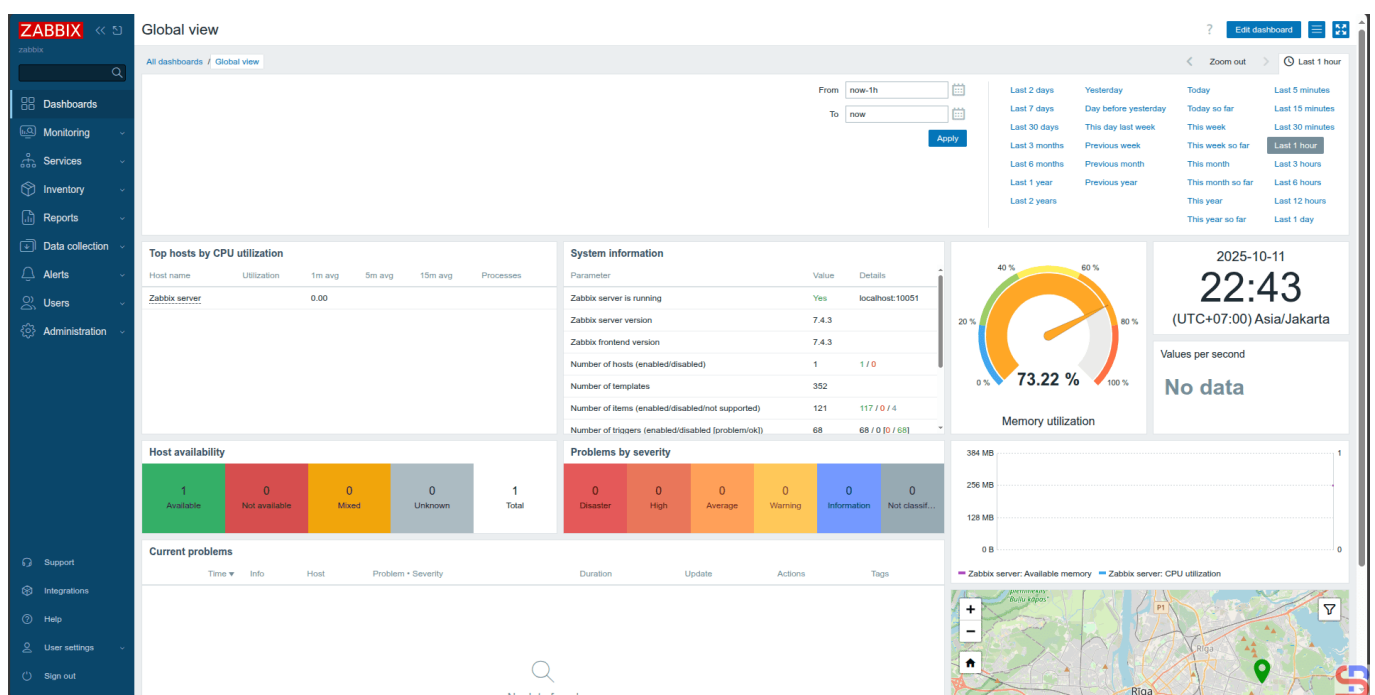
Click the **Finish** button, and a screen like the one shown below will appear.



The image shows the Zabbix login interface. At the top is the ZABBIX logo in a red box. Below it are two input fields: 'Username' and 'Password', both outlined in red. Under the password field is a checkbox labeled 'Remember me for 30 days' which is checked. At the bottom is a blue 'Sign in' button. A small red and blue logo is visible in the bottom right corner of the login area.

Enter the username and password of Zabbix

For your information, the initial username for Zabbix is **Admin** and the initial password is **zabbix**. After you enter the username and password, click the Sign in button, and there will be a display like the image below:



The initial display of Zabbix

You have successfully installed the Zabbix application on your Ubuntu server.

Note

To install Zabbix on a different operating system, you can go to [this page](#) to see how to install Zabbix on your server.

References

en.wikipedia.org

zabbix.com

medium.com

How to Stop Linux From Erasing the File(s) or Folder(s)?

written by sysadmin | 7 January 2026

I want to prevent a specific file or folder from being deleted, even with the root user.

Problem

How to stop Linux from erasing the file(s) or folder(s)?

Solution

In Linux, there are two commands you can use to prevent unauthorized changes, protect the critical file(s) or folder(s), and ensure the integrity of the system: the **lsattr** and **chattr** commands.

A. The lsattr command

To see the properties of files or directories on a file system that supports extended attributes, use the `lsattr` command. So, `lsattr` command displays special attributes that are not visible with the `ls -l` command. Run the command below to see the list of attributes:

`lsattr`

```
root@ubuntu2404:/data# ls
docs  test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# lsattr
-----e----- ./test.txt
-----e----- ./docs
root@ubuntu2404:/data#
```

List the attribute(s)

By default, if your server uses the ext4 format, a file or folder on that Linux server will have the `e` attribute, or extent format, which is a more efficient file storage method than the traditional block method. Below is a brief explanation of the various attributes:

File/Folder Attributes

Attribute	Explanation
-	Attribute not set
a	Append-only – file can only be opened for appending without modifying existing data on a File, not overwritten or truncated
A	No atime updates – access time is not updated when the file is read
c	Compressed – file is stored compressed on disk (kernel support required).
C	No Copy-on-Write (CoW) – disables CoW for Btrfs files.
d	No dump – file is ignored by the dump backup program.

D	Synchronous directory updates – directory changes are written immediately to disk.
e	Extents – file uses extents to map blocks (default on ext4).
i	Immutable – file cannot be modified, deleted, renamed, or linked (even by root).
j	Data journaling – file data is journaled as well as metadata.
s	Secure deletion – blocks are zeroed when file is deleted (if supported).
S	Synchronous updates – file changes are written immediately to disk.
t	No tail-merging – prevents tail-packing (used in ReiserFS).
T	Top of directory hierarchy – marks directory as top-level for block allocator.
u	Undelete – when deleted, file content can be recovered (if supported).

B. The `chattr` command

With Linux, users can modify the properties of files and directories with the ‘`chattr`’ (change attribute) command. Using this command, you can protect a file or directory from deletion or addition, which is very useful for protecting critical files or folders. To use this command, you can follow the format below:

```
chattr [operator][attribute] file(s)/folder(s)
```

You can see the attributes in the table above, while the table below shows the operators you can use:

The Operators in attribute `file(s)/folder(s)`

Operator	Explanation
----------	-------------

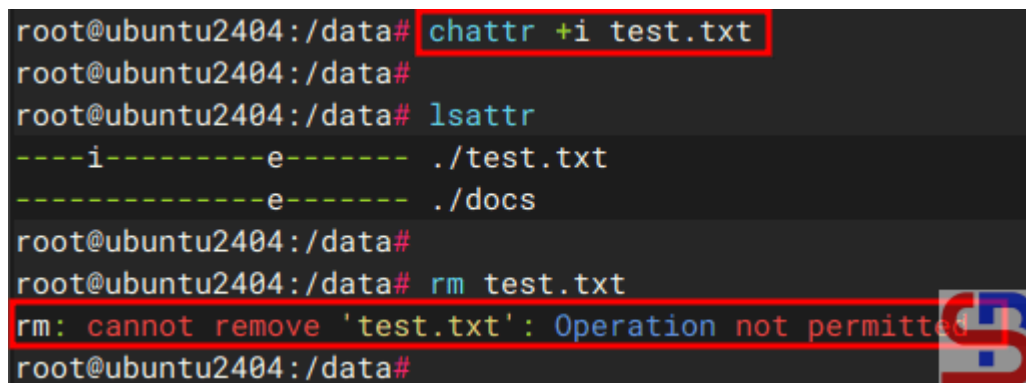
+	Add the specified attribute(s) to the file/directory (keep existing ones).
-	Remove the specified attribute(s) from the file/directory.
=	Set the attribute(s) exactly as specified (replace all existing ones).

1. Making a file undeletable

Use the command below to make a file undeletable in a file, for example, test.txt:

```
chattr +i test.txt
```

Then, try deleting the file. It should be undeletable even with the root user, as shown in the image below:



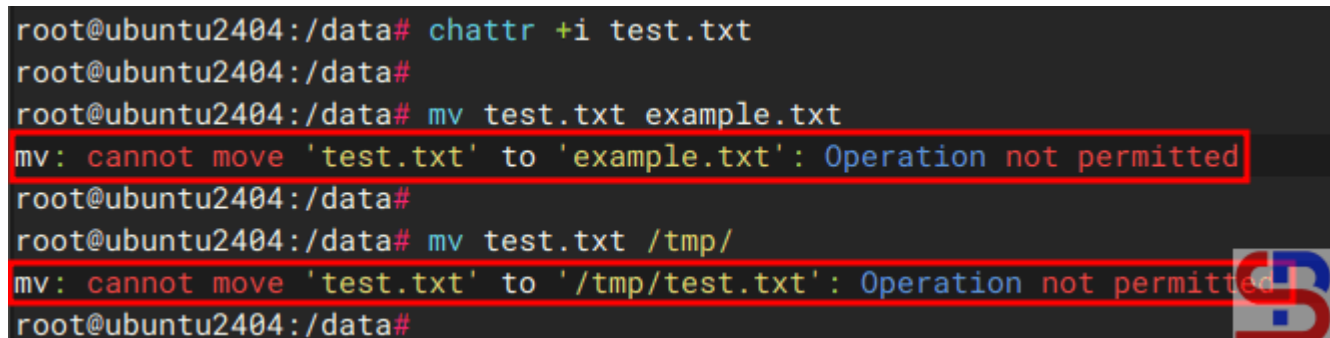
```

root@ubuntu2404:/data# chattr +i test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# lsattr
----i-----e----- ./test.txt
-----e----- ./docs
root@ubuntu2404:/data#
root@ubuntu2404:/data# rm test.txt
rm: cannot remove 'test.txt': Operation not permitted
root@ubuntu2404:/data#

```

Making a file undeletable

You cannot even rename the file or move it to another folder, as shown in the image below:



```

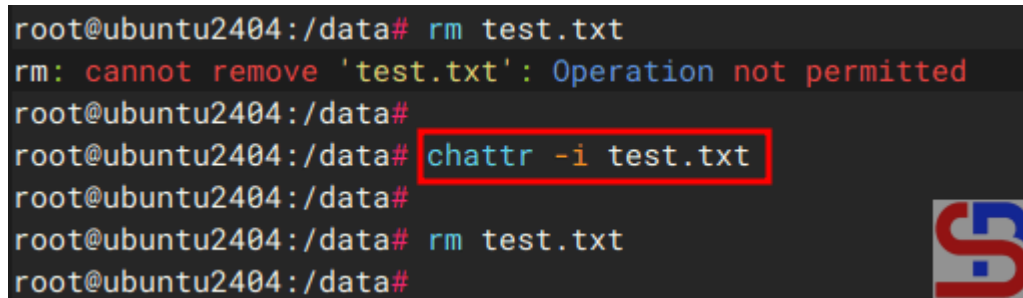
root@ubuntu2404:/data# chattr +i test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# mv test.txt example.txt
mv: cannot move 'test.txt' to 'example.txt': Operation not permitted
root@ubuntu2404:/data#
root@ubuntu2404:/data# mv test.txt /tmp/
mv: cannot move 'test.txt' to '/tmp/test.txt': Operation not permitted
root@ubuntu2404:/data#

```

Can not rename or move a file

If you want to really delete a file, you have to run the command below, and you can delete the file like in the image below:

```
root@ubuntu2404:/data# rm test.txt
rm: cannot remove 'test.txt': Operation not permitted
root@ubuntu2404:/data#
root@ubuntu2404:/data# chattr -i test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# rm test.txt
root@ubuntu2404:/data#
```

A terminal window screenshot showing a sequence of commands. The user attempts to delete 'test.txt' with 'rm', which fails with 'Operation not permitted'. Then, they run 'chattr -i test.txt', which is highlighted with a red box. Finally, they run 'rm test.txt' again, and the prompt returns without an error message. A small logo is visible in the bottom right corner of the terminal window.

The file can be deleted

2. Append data without modifying existing data on a File

If you want the file to be able to add content without deleting the content that is already in the test.txt file, use the command below

```
chattr +a test.txt
```

Then try running the two commands below:

```
echo "Add test" > test.txt
echo "Just test" >> test.txt
```

And only the second command should be able to be executed, as shown in the image below:

```
root@ubuntu2404:/data# cat test.txt
This is first line
root@ubuntu2404:/data#
root@ubuntu2404:/data# chattr +a test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# lsattr
-----a-----e----- ./test.txt
-----e----- ./docs
root@ubuntu2404:/data#
root@ubuntu2404:/data# echo "Just test" > test.txt
-bash: test.txt: Operation not permitted
root@ubuntu2404:/data#
root@ubuntu2404:/data# echo "Just test" >> test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# cat test.txt
This is first line
Just test
root@ubuntu2404:/data#
```

Append a file

To get the file back to “normal”, use the command below:

```
chattr -a test.txt
```

3. Making a folder secure

Use the command below if you want your folder to be undeleted, for example, the docs folder:

```
chattr -R +i docs/
```

Now, try to delete the folder, and it should not be deletable as shown in the image below:

```
root@ubuntu2404:/data# ls docs/
todo.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# chattr -R +i docs/
root@ubuntu2404:/data#
root@ubuntu2404:/data# rm -rf docs/
rm: cannot remove 'docs/todo.txt': Operation not permitted
root@ubuntu2404:/data#
```

Can not delete the folder

Even you can't delete the files in the folder, as shown in the image below:

```
root@ubuntu2404:/data# rm -rf docs/todo.txt
rm: cannot remove 'docs/todo.txt': Operation not permitted
root@ubuntu2404:/data#
```



Cannot delete the file in the folder

For the folder to be deleted, use the command below:

```
chattr -R +i docs/
```

Note

You can run more than one option to change the attributes of a file in a command. For example, you want the file to be undeletable and appended without deleting the content that was previously present in the test.txt file, then use the command below:

```
chattr +ia test.txt
```

```
root@ubuntu2404:/data# lsattr
-----e----- ./test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# chattr +ia test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# lsattr
---ia-----e----- ./test.txt
root@ubuntu2404:/data#
```

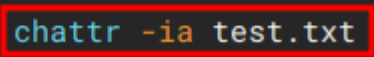


Give more than one attribute for one file

Likewise, you can delete more than one attribute for the test.txt file, then use the command below:

```
chattr -ia test.txt
```

```
root@ubuntu2404:/data# lsattr
---ia-----e----- ./test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# chattr -ia test.txt
root@ubuntu2404:/data#
root@ubuntu2404:/data# lsattr
---e-----e----- ./test.txt
root@ubuntu2404:/data#
```



Delete more than one attribute in one file

You can also run and change the attribute in more than one file or folder. For example, you want to change the attributes for test.txt and ok.txt, use the following command:

```
chattr +ia test.txt ok.txt
```

References

[geeksforgeeks.org](https://www.geeksforgeeks.org)

[tecmint.com](https://www.tecmint.com)

[howtoforge.com](https://www.howtoforge.com)

[How to Reset the Password in Ubuntu?](#)

written by sysadmin | 7 January 2026

I want to access the user on the Ubuntu server that has the privilege of root using the sudo command, but I forgot my user password.

Problem

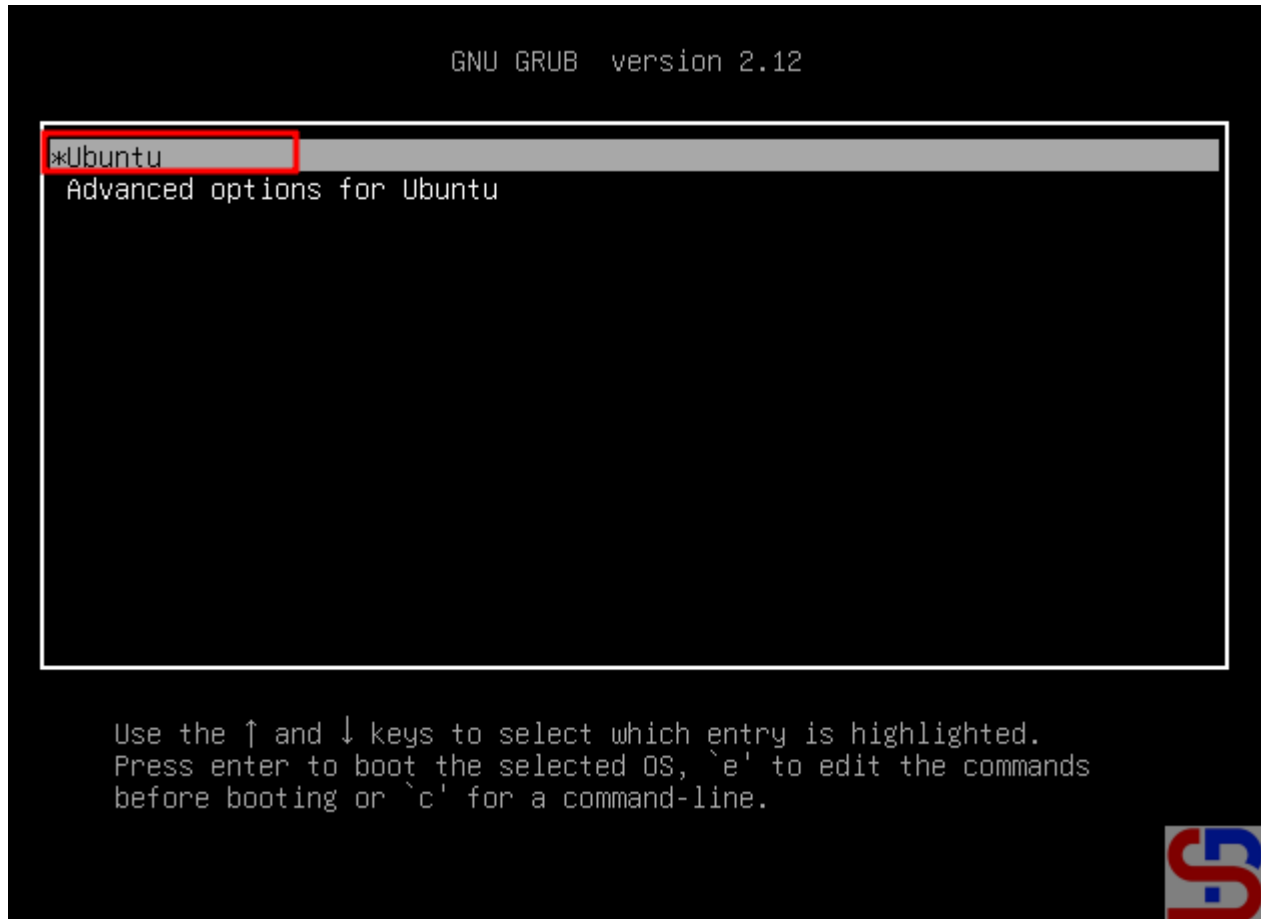
How to reset the password in Ubuntu?

Solution

Here are the steps to reset the password in Ubuntu:

1. Reboot the server

Reboot the server and press the **Esc** key or Shift key, and there should be a display like below:



Choose the Ubuntu

2. Click the first option

To enter recovery mode, select the top part of the image above and push the **e** button, so that there will be a display like the image below:

GNU GRUB version 2.12

```
setparams 'Ubuntu'

    recordfail
    load_video
    gfxmode $linux_gfx_mode
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; \
fi
    insmod part_gpt
    insmod ext2
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 -\
-hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2 c59b0229-fcf2-4f2f-a6c7-\
e183c8ca6093
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.



The GRUB options

Find the line starting with **linux**, similar to the picture below:

GNU GRUB version 2.12

```
insmod part_gpt
insmod ext2
set root='hd0,gpt2'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 -\
-hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2  c59b0229-fcf2-4f2f-a6c7-\
e183c8ca6093
else
    search --no-floppy --fs-uuid --set=root c59b0229-fcf2-4f2f-a6c\
7-e183c8ca6093
fi
linux      /vmlinuz-6.8.0-84-generic root=/dev/mapper/ubuntu--\
vg-ubuntu--lv ro
initrd     /initrd.img-6.8.0-84-generic
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.



Find the line starting with linux

Remove everything from **ro** and append **rw init=/bin/bash** to the end of this line, like the picture below:

GNU GRUB version 2.12

```
insmod part_gpt
insmod ext2
set root='hd0,gpt2'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 -\
-hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2  c59b0229-fcf2-4f2f-a6c7-\
e183c8ca6093
else
  search --no-floppy --fs-uuid --set=root c59b0229-fcf2-4f2f-a6c\
7-e183c8ca6093
fi
linux /vmlinuz-6.8.0-84-generic root=/dev/mapper/ubuntu--\
vg-ubuntu--lv rw init=/bin/bash
initrd /initrd.img-6.8.0-84-generic
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.



Change the script

After you change the script, press **F10** or **Ctrl+x** to boot these parameters.

3. Run the commands

In the recovery mode, run the command below:

```
mount | grep -w /
```

After that, execute the command below to change the password:

```
passwd
```

After you change the password, run the commands below:

```
mount -o remount,ro /
exec /sbin/init
```

```
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/# mount | grep -w /
/dev/mapper/ubuntu--vg-ubuntu--lv on / type ext4 (rw,relatime)
root@(none):/#
root@(none):/# passwd sysadmin
New password:
Retype new password:
passwd: password updated successfully
root@(none):/#
root@(none):/# mount -o remount,ro /
[ 119.578818] EXT4-fs (dm-0): re-mounted 0eef966e-11fc-40fc-a390-e4418282042c r
o. Quota mode: none.
root@(none):/#
root@(none):/# exec /sbin/init
```

Run the commands

The Linux server will reboot, and after that, try to log in with the new password that you set before.

Note

By default, you cannot log in directly as root on Ubuntu, so you can't change your password to root because to be root on Ubuntu, you only need to use your sudo command and enter your user password.

References

[tecmint.com](https://www.tecmint.com)

askubuntu.com

[infotechys.com](https://www.infotechys.com)