

How to Manage a User in PostgreSQL?

written by sysadmin | 8 November 2025

[The previous article](#) explained the basic commands in the PostgreSQL database. This article will explain how to set up a user in PostgreSQL.

Problem

How to manage a user in PostgreSQL?

Solution

By default, PostgreSQL runs MD5 encryption to save PostgreSQL user passwords in the pg_authid table. However, since PostgreSQL version 10 and later, PostgreSQL has added scram-sha-256 encryption, which is more secure compared to MD5 encryption. To use this encryption, you have to go to the **/etc/postgresql/18/main/postgresql.conf** file if you use PostgreSQL 18 and look for the word **scram-sha-256** in the file, and open the fence located to the left of the word so it looks like the one below:

```
password_encryption = scram-sha-256
```

Then restart PostgreSQL using the command:

```
sudo systemctl restart postgresql
```

A. List all users

To see all users in PostgreSQL, type the following command:

```
\du
```

```
postgres=# \du
                                List of roles
Role name | Attributes
-----+-----
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS
postgres=#
```

List all users in PostgreSQL

By default, PostgreSQL uses the postgres user to access all databases in PostgreSQL.

B. Create a new user

To create a new user in PostgreSQL, use the format below on the PostgreSQL server:

```
create user username with encrypted password 'your-password';
```

For example, if you want to create a mydb_user user with the password qwerty, then use the command below:

```
create user mydb_user with encrypted password 'qwerty';
```

```
postgres=# create user mydb_user with encrypted password 'qwerty';
CREATE ROLE
postgres=# \du
                                List of roles
Role name | Attributes
-----+-----
mydb_user | 
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS
postgres=#
```

Create a user in PostgreSQL

C. Change user password

To change the password in PostgreSQL, use the format below:

```
ALTER USER username WITH PASSWORD 'your_password';
```

For example, if you want to change the password for the mydb_user to q1w2e3, then use the command below:

```
ALTER USER mydb_user WITH PASSWORD 'q1w2e3';
```

```
mydb=# ALTER USER mydb_user WITH PASSWORD 'q1w2e3';  
ALTER ROLE  
mydb=#
```



Change the password in PostgreSQL

D. Create a user privilege

To give the user privileges for a database in PostgreSQL using the format below:

```
GRANT permission_type ON db_name TO username;
```

For example, in the previous article, you created a mydb database, so use the command below to grant all permission types:

```
grant all privileges on database mydb to mydb_user;
```

```
postgres=# grant all privileges on database mydb to mydb_user;  
GRANT
```



Grant all to the user in PostgreSQL

If you want a user to only be able to view the mydb database for the sysadmin user, use the command below:

```
CREATE user john with encrypted password '123456';  
GRANT CONNECT ON DATABASE mydb TO john;  
\c mydb  
GRANT USAGE ON SCHEMA public TO john;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO john;
```

```

mydb=# CREATE user john with encrypted password '123456';
CREATE ROLE
mydb=# GRANT CONNECT ON DATABASE mydb TO john;
GRANT
mydb=# \c mydb
You are now connected to database "mydb" as user "postgres".
mydb=# GRANT USAGE ON SCHEMA public TO john;
GRANT
mydb=# GRANT SELECT ON ALL TABLES IN SCHEMA public TO john;
GRANT
mydb=#

```

Grant viewer only in PostgreSQL

To see the grants you have made in PostgreSQL, use the command:

`\l+`

```
mydb=# \l+
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges	Size	Tablespace	Description
mydb	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres postgres=CtC/postgres mydb_user=CtC/postgres john=c/postgres	7742 kB	pg_default	
postgres	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres	7670 kB	pg_default	default administrative connection database
template0	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			postgres=CtC/postgres	7513 kB	pg_default	unmodifiable empty database
template1	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres postgres=CtC/postgres	7742 kB	pg_default	default template for new databases

(4 rows)

```
mydb=#
```

List all grants in PostgreSQL

E. Remove a user privilege

If you want to remove a privilege from a user, you can use the format below:

```
REVOKE permission_type ON table_name FROM user_name;
```

For example, if you want to remove a privilege for mydb_user, use the command below:

```
REVOKE all privileges on database mydb FROM mydb_user;
```

```

mydb=# \l+
          List of databases
  Name | Owner  | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 mydb  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =Tc/postgres + postgres=Ctc/postgres + mydb_user=Ctc/postgres + john=C/postgres | 7742 kB | pg_default |
 postgres | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7670 kB | pg_default | default administrative connection database
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7513 kB | pg_default | unmodifiable empty database
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7742 kB | pg_default | default template for new databases
(4 rows)

mydb=# REVOKE all privileges on database mydb FROM mydb_user;
REVOKE
mydb=#
mydb=# \l+
          List of databases
  Name | Owner  | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 mydb  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =Tc/postgres + postgres=Ctc/postgres + john=C/postgres | 7742 kB | pg_default |
 postgres | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7670 kB | pg_default | default administrative connection database
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7513 kB | pg_default | unmodifiable empty database
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7742 kB | pg_default | default template for new databases
(4 rows)

```

Revoke all grants from a user

Or, use the below command if you want to revoke from user john:

REVOKE CONNECT on database mydb FROM john;

```

mydb=# \l+
          List of databases
  Name | Owner  | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 mydb  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =Tc/postgres + postgres=Ctc/postgres + john=C/postgres | 7742 kB | pg_default |
 postgres | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7670 kB | pg_default | default administrative connection database
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7513 kB | pg_default | unmodifiable empty database
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7742 kB | pg_default | default template for new databases
(4 rows)

mydb=# REVOKE CONNECT on database mydb FROM john;
REVOKE
mydb=# \l+
          List of databases
  Name | Owner  | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 mydb  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =Tc/postgres + postgres=Ctc/postgres | 7742 kB | pg_default |
 postgres | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7670 kB | pg_default | default administrative connection database
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7513 kB | pg_default | unmodifiable empty database
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres + postgres=Ctc/postgres | 7742 kB | pg_default | default template for new databases
(4 rows)

```

Revoke connect only from a user

F. Delete a user

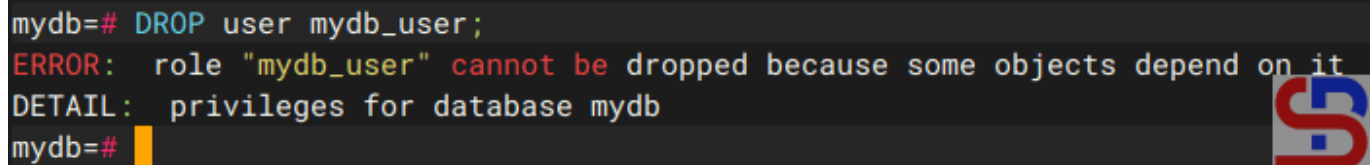
Use the format below to delete a user in PostgreSQL:

DROP USER username;

If you want to delete mydb_user, use the command below:

DROP user mydb_user;

```
mydb=# DROP user mydb_user;  
ERROR:  role "mydb_user" cannot be dropped because some objects depend on it  
DETAIL:  privileges for database mydb  
mydb=#
```

A terminal window with a dark background. The text is in a monospaced font. The first line is a command to drop a user. The second line is an error message in red. The third line is a detail message. The fourth line is the prompt. On the right side, there is a logo consisting of a stylized 'S' in blue and red.

Error when dropping a user in PostgreSQL

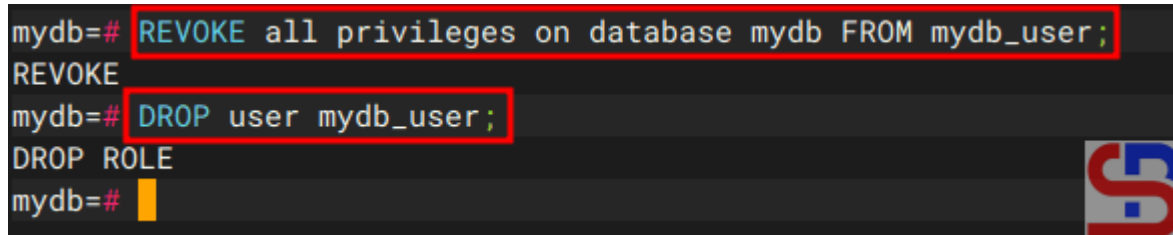
You can see from the image above, there is an error when you want to delete a user. So, you have to remove some objects that depend on it. Use the command below:

```
REVOKE all privileges on database mydb FROM mydb_user;
```

and then delete the user using the below command:

```
DROP user mydb_user;
```

```
mydb=# REVOKE all privileges on database mydb FROM mydb_user;  
REVOKE  
mydb=# DROP user mydb_user;  
DROP ROLE  
mydb=#
```

A terminal window with a dark background. The text is in a monospaced font. The first line is a command to revoke privileges. The second line is the output. The third line is a command to drop a user. The fourth line is the output. The fifth line is the prompt. On the right side, there is a logo consisting of a stylized 'S' in blue and red.

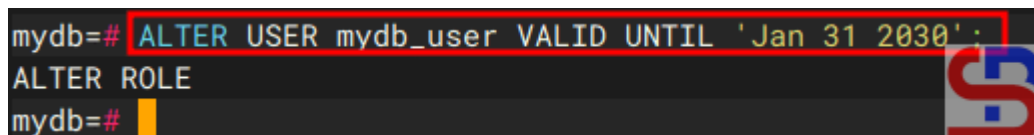
Drop a user in PostgreSQL

Note

If you want a user to only be able to access the PostgreSQL database for a certain time, for example, until January 31, 2030, use the command below:

```
ALTER USER mydb_user VALID UNTIL 'Jan 31 2030';
```

```
mydb=# ALTER USER mydb_user VALID UNTIL 'Jan 31 2030';  
ALTER ROLE  
mydb=#
```

A terminal window with a dark background. The text is in a monospaced font. The first line is a command to alter a user. The second line is the output. The third line is the prompt. On the right side, there is a logo consisting of a stylized 'S' in blue and red.

Provides time constraints for users in PostgreSQL

References

medium.com

datacamp.com

How to Create a Docker Image?

written by sysadmin | 8 November 2025

Previously, you often used a Docker image that you downloaded from [Docker Hub](#). But now, you want to create a Docker image for your own application needs.

Problem

How to create a Docker image?

Solution

To create a Docker image, you must create a Dockerfile.

A. Dockerfile

Dockerfile is a script that contains a set of instructions used to create a Docker image using the format:

```
#comment  
INSTRUCTION arguments
```

You should know that Docker runs Dockerfile files sequentially from top to bottom, and this file does not have a file extension, so just write Dockerfile. To make things easier, it is best to place this Dockerfile in the same location as the files needed to create a Docker image, so that it is easier to create. That way, to create a Docker image, just run the command:

```
docker build -t image_name .
```

B. Instructions

The following are the standard Dockerfile instructions:

1. FROM instruction

This instruction is the first command to perform a build stage in the Dockerfile with the example below:

FROM alpine:3

```
sysadmin@docker:~/image$ cat Dockerfile
FROM alpine:3

sysadmin@docker:~/image$ docker build -t from_ins .
[*] Building 2.6s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 52B
=> [internal] load metadata for docker.io/library/alpine:3
=> [internal] load .dockerignore
=> => transferring context: 63B
=> CACHED [1/1] FROM docker.io/library/alpine:3
=> exporting to image
=> => exporting layers
=> => writing image sha256:706db57fb2063f39f69632c5b5c9c439633fda35110e65587c5d85553fd1cc38
=> => naming to docker.io/library/from_ins
sysadmin@docker:~/image$
```

Using the FROM instruction

2. LABEL instruction

To add metadata to the Docker image you create, where the metadata is additional information, such as the name of the application, creator, website, and so on.

FROM alpine:3

LABEL author="sysadmin"

LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"

```
sysadmin@docker:~/image$ docker image inspect label_ins | grep Labels -A 3
"Labels": {
  "author": "sysadmin",
  "company": "sysadminpedia",
  "website": "https://www.sysadminpedia.com"
}
sysadmin@docker:~/image$
```

Using the LABEL instruction

3. WORKING DIRECTORY instruction

This instruction specifies directories/folders to execute instructions in the container. By default, if there is no working directory, then the container will go to the / folder automatically. If the workdir does not exist, the directory will automatically be created, and then, after we

determine the location of the workdir, the directory will be used as a place to execute the next instruction. If the workdir's location is a relative path, then it will automatically enter the directory of the previous workdir. Workdir can also be used as a path for the first location when it enters the container.

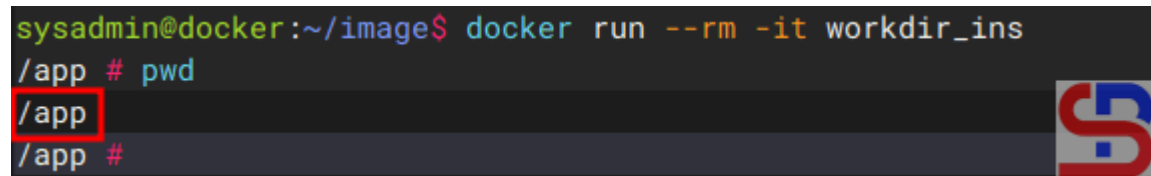
```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
sysadmin@docker:~/image$ docker run --rm -it workdir_ins
/app # pwd
/app
/app #
```



Using the WORKDIR instruction

4. RUN instruction

This instruction is a command in the image during the build stage, where the results of the RUN command will be committed to changes to the image, so that the RUN command will only be executed during the Docker build process, and this command will not be executed again and when you run the docker container from the image the RUN command will not be executed.

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

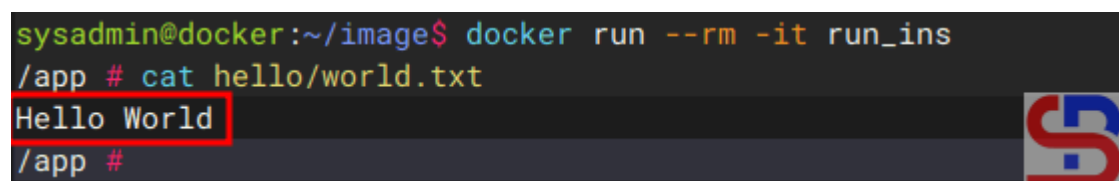
```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > "hello/world.txt"
```

```
sysadmin@docker:~/image$ docker run --rm -it run_ins
/app # cat hello/world.txt
Hello World
/app #
```



Using the RUN instruction

5. USER instruction

To change the user or user group when Docker images are run, because by default, Docker will use the root user, and we can change it by using the user instruction, with the note that the user must be created first.

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

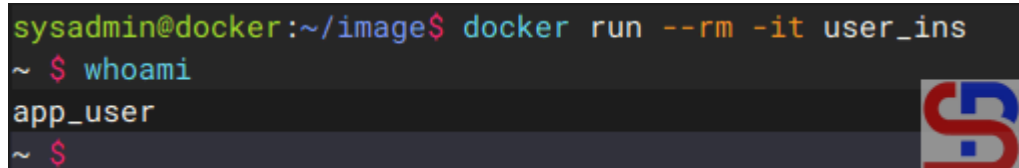
```
RUN echo "Hello World" > "hello/world.txt"
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

```
USER app_user
```

A terminal window with a dark background. The prompt is 'sysadmin@docker:~/image\$'. The command 'docker run --rm -it user_ins' is entered. The prompt changes to '~ \$'. The command 'whoami' is entered. The output 'app_user' is displayed. The prompt changes back to '~ \$'.

```
sysadmin@docker:~/image$ docker run --rm -it user_ins
~ $ whoami
app_user
~ $
```

Using the USER instruction

6. ENTRYPOINT instruction

An instruction in the Dockerfile specifies the main command that is executed when the container is started, and this is the main process of the container.

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

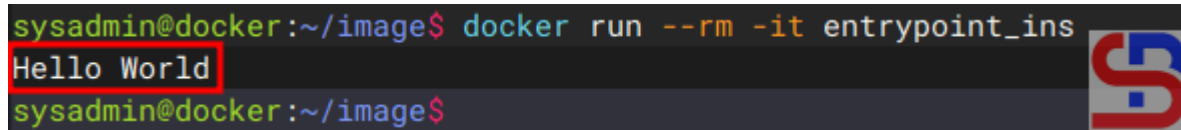
```
RUN mkdir hello
```

```
RUN echo "Hello World" > "hello/world.txt"
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
RUN chown -R app_user:app_group /app
USER app_user
```

```
ENTRYPOINT ["cat", "hello/world.txt"]
```



Using the ENTRYPOINT instruction

7. COMMAND instruction

An instruction that is used when the Docker container is running and will not be executed during the build image process. You cannot add more than one CMD instruction in an image, and if there is more than one instruction in an image, then the last CMD instruction will be executed. If there is an ENTRYPOINT instruction, then the CMD instruction becomes an argument from the ENTRYPOINT instruction. Examples of its use are as below:

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > hello/world.txt
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

```
USER app_user
```

```
ENTRYPOINT ["cat"]
```

```
CMD ["hello/world.txt"]
```

The Dockerfile file above seems to be the command cat hello/world.txt, as in the picture below:

```
sysadmin@docker:~/image$ docker run --rm -it cmd_ins
Hello World
sysadmin@docker:~/image$
```



Using the CMD instruction

If there is no ENTRYPOINT, the CMD itself can be executed directly as a container command, as in the Dockerfile file below:

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > hello/world.txt
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

```
USER app_user
```

```
CMD ["cat", "hello/world.txt"]
```

8. COPY instruction

To add files from the source to the destination folder in a Docker image folder. For example, if you want to copy all files with the extension .txt from a folder on the server, put them in the hello folder in the Docker image.

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > "hello/world.txt"
```

```
RUN addgroup -S app_group
```

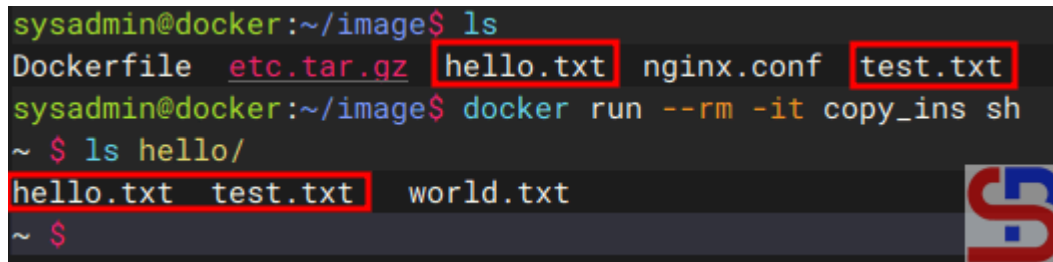
```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

USER app_user

COPY *.txt hello

CMD ["cat", "hello/world.txt"]



```
sysadmin@docker:~/image$ ls
Dockerfile  etc.tar.gz  hello.txt  nginx.conf  test.txt
sysadmin@docker:~/image$ docker run --rm -it copy_ins sh
~ $ ls hello/
hello.txt  test.txt  world.txt
~ $
```

Using the COPY instruction

You can see in the image that in the folder on the server, there is an nginx.conf file and a tar.gz file, but the files that are copied are only files with the extension .txt, such as the COPY command in the Dockerfile file.

9. ADD instruction

To add files from the source to the destination folder in the Docker image, and this command can detect if a source file is a compressed file, such as gzip, and will automatically extract it in the destination folder, and can also support adding multiple files at once. The difference with COPY is that COPY can only copy files, while ADD, in addition to copying, can also download the source from the URL and automatically extract compressed files. The best way to practice is to use COPY as much as possible, but if you really need to extract compressed files, then use the ADD command.

FROM alpine:3

LABEL author="sysadmin"

LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"

WORKDIR /app

RUN mkdir hello

RUN echo "Hello World" > "hello/world.txt"

RUN addgroup -S app_group

RUN adduser -S -D -h /app app_user app_group

```
RUN chown -R app_user:app_group /app
USER app_user
```

```
COPY *.txt hello
```

```
ADD *.gz hello
```

```
CMD ["cat", "hello/world.txt"]
```

```
sysadmin@docker:~/image$ ls
Dockerfile etc.tar.gz hello.txt nginx.conf test.txt
sysadmin@docker:~/image$ docker run --rm -it add_ins sh
~ $ ls hello/
crontab hello.txt hosts test.txt world.txt
~ $
```

Using the ADD instruction

The crontab and hosts files are extracted from the etc.tar.gz file, and this explains that when you use the COPY instruction to copy a compressed file, the file will be extracted automatically in the container.

10. .dockerignore File

To specify which file(s) or folder(s) we ignore when the process of copying or adding file(s) or folder(s) to the Docker image. Because when doing the copy or add process, Docker will read the file named .dockerignore first. Create a **.dockerignore** file and then fill it with example.txt and qwerty.txt. Then, create the files example.txt and qwerty.txt in that folder, and they should not be copied to the folder in the Docker image.

```
sysadmin@docker:~/image$ touch example.txt qwerty.txt
sysadmin@docker:~/image$ ls -a
. .. Dockerfile .dockerignore etc.tar.gz example.txt hello.txt nginx.conf qwerty.txt test.txt
sysadmin@docker:~/image$ cat .dockerignore
example.txt
qwerty.txt
sysadmin@docker:~/image$ docker run --rm -it ignore_ins sh
~ $ ls hello/
crontab hello.txt hosts test.txt world.txt
~ $
```

Using the .dockerignore file

11. EXPOSE instruction

To tell the container which port to listen port on a

specific number and protocol. Actually, EXPOSE will not publish any ports, but is only for documentation to inform the creator of the Docker container that this Docker image will use a specific port when run in a Docker container. For example, as below:

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > "hello/world.txt"
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

```
USER app_user
```

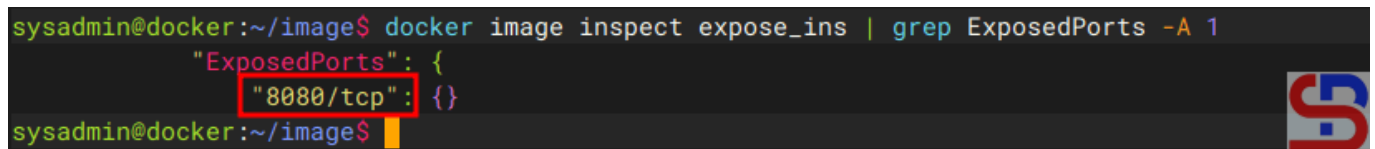
```
ADD *.txt hello
```

```
COPY *.gz hello
```

```
EXPOSE 8080
```

```
CMD ["cat", "hello/world.txt"]
```

To see the open ports on this Docker image, use the command as shown in the image below:



```
sysadmin@docker:~/image$ docker image inspect expose_ins | grep ExposedPorts -A 1
    "ExposedPorts": {
      "8080/tcp": {}
    }
sysadmin@docker:~/image$
```

Using the EXPOSE instruction

To create a container using this Docker image and see the open ports on this container, use the command as shown in the image below:

```
sysadmin@docker:~/image$ docker run -d --name expose_ins -p 8080:80 expose_ins
c57dfefc254a265cc0b557e06f5f239a1a353d25d30719a51ff82b2e6b38b4dc
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker container inspect expose_ins | grep HostPort
    "HostPort": "8080"
sysadmin@docker:~/image$
```

Display the EXPOSE instruction in the container

12. ENVIRONMENT VARIABLE instruction

To change the environment variables either during the build stage or when running in a Docker Container. The ENV defined in the Dockerfile can be reused using the `${ENV_NAME}` syntax.

```
FROM alpine:3
```

```
LABEL author="sysadmin"
```

```
LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"
```

```
WORKDIR /app
```

```
RUN mkdir hello
```

```
RUN echo "Hello World" > "hello/world.txt"
```

```
RUN addgroup -S app_group
```

```
RUN adduser -S -D -h /app app_user app_group
```

```
RUN chown -R app_user:app_group /app
```

```
USER app_user
```

```
COPY *.txt hello
```

```
ADD *.gz hello
```

```
EXPOSE 8080
```

```
ENV APP_PORT=8080
```

```
EXPOSE ${APP_PORT}
```

```
CMD ["cat", "hello/world.txt"]
```

The Environment Variable created using the ENV instruction is stored in the Docker image and can be viewed using the `docker image inspect` command.

```
sysadmin@docker:~/image$ docker image inspect env_ins | grep Env -A 3
    "Env": [
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
      "APP_PORT=8080"
    ],
sysadmin@docker:~/image$
```

Using the ENV instruction

Environment variables can be changed in value when creating a Docker container with the docker command using the **-env-key=value** option. So if you use the Dockerfile above, which uses APP_PORT=8080, but you want to use APP_PORT=9090, then you can use the command:

```
docker container create --name env_ins --env APP_PORT=9090 -p 9090:80 env_ins
```

```
sysadmin@docker:~/image$ docker container create --name env_ins --env APP_PORT=9090 -p 9090:80 env_ins
1d645f103fb2905cb31be5f772349e5d6ca86b7165396b1a115db759d555fecc
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker container inspect env_ins | grep APP_PORT
    "APP_PORT=9090",
sysadmin@docker:~/image$
```

Using the ENV instruction when creating a container

13. VOLUME instruction

To create the volume automatically when creating the container, and all the files contained in the volume are automatically copied to the Docker Volume, even though we didn't create the Docker Volume when creating the Docker Container. It is suitable for cases when the application stores data in a file, so that the data can be automatically and safely stored in the Docker Volume.

FROM alpine:3

LABEL author="sysadmin"

LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"

WORKDIR /app

RUN mkdir hello

RUN echo "Hello World" > "hello/world.txt"

RUN addgroup -S app_group

RUN adduser -S -D -h /app app_user app_group

RUN chown -R app_user:app_group /app

```
USER app_user
```

```
COPY *.txt hello
```

```
ADD *.gz hello
```

```
ENV APP_PORT=8080
```

```
EXPOSE ${APP_PORT}
```

```
ENV APP_DATA=/logs
```

```
VOLUME ${APP_DATA}
```

```
CMD ["cat", "hello/world.txt"]
```

```
sysadmin@docker:~/image$ docker image inspect vol_ins | grep Volume -A2
    "Volumes": {
      "/logs": {}
    },
sysadmin@docker:~/image$
```

Using the VOL instruction

Then, create a container from the Docker image and inspect it with the keyword Mounts. List the volume on the server, and it should be the same as in the image below:

```
sysadmin@docker:~/image$ docker container create --name vol_ins --env APP_PORT=9090 -p 9090:80 vol_ins
33cc9ecac5b30529feccf8f6ddce7bbd0c6edf97b70496976df40100a0de1f48
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker container inspect vol_ins | grep Mounts -A6
    "Mounts": [
      {
        "Type": "volume",
        "Name": "fef469d23766c072f239a44829516d0219ee1b98a2c71ef13a8516cb2f8763a2",
        "Source": "/var/lib/docker/volumes/fef469d23766c072f239a44829516d0219ee1b98a2c71ef13a8516cb2f8763a2/_data",
        "Destination": "/logs",
        "Driver": "local",
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker volume ls
DRIVER    VOLUME NAME
local     a685e9b17ea6c499d743dfd35d2fb799706b2f278d7d09bddfa5f0e377a86229
local     fef469d23766c072f239a44829516d0219ee1b98a2c71ef13a8516cb2f8763a2
sysadmin@docker:~/image$
```

Display the volume

14. ARGUMENT instruction

To define variables that can be used by the user to send when performing a Docker build process, use the **-build-arg key=value** command. This instruction is only used during the build time process, which means that when running in a Docker container, this instruction will not be used any

differently from the ENV used. Accessing variables from ARG is the same as accessing variables from ENV using `${variable_name}`.

FROM alpine:3

LABEL author="sysadmin"

LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"

WORKDIR /app

RUN mkdir hello

RUN echo "Hello World" > "hello/world.txt"

RUN addgroup -S app_group

RUN adduser -S -D -h /app app_user app_group

RUN chown -R app_user:app_group /app

USER app_user

COPY *.txt hello

ADD *.gz hello

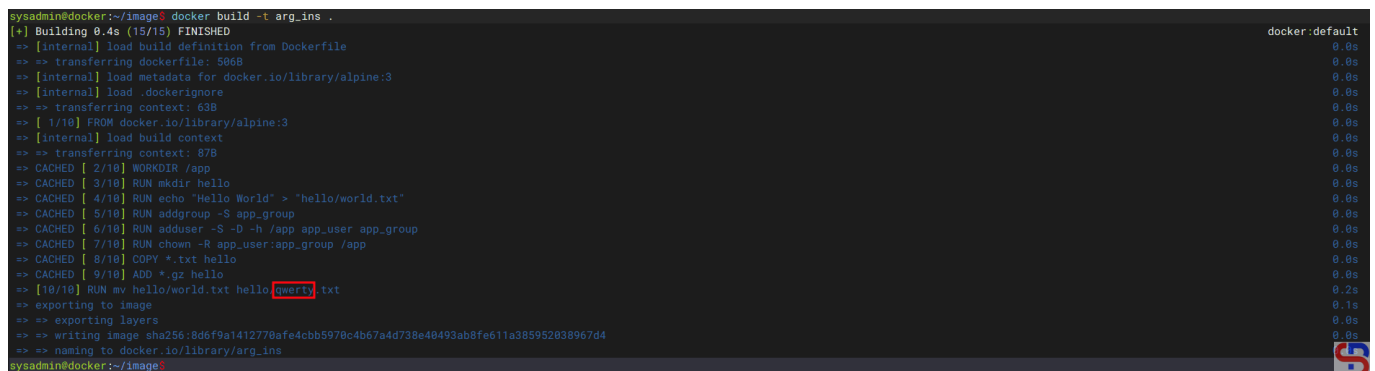
ENV APP_PORT=8080

EXPOSE \${APP_PORT}

ARG app=qwerty

RUN mv hello/world.txt hello/\${app}.txt

CMD ["cat", "hello/\${app}.txt"]



```
sysadmin@docker:~/image$ docker build -t arg.ins .
[+] Building 0.4s (15/15) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 586B
=> [internal] load metadata for docker.io/library/alpine:3
=> [internal] load .dockerignore
=> => transferring context: 63B
=> [1/10] FROM docker.io/library/alpine:3
=> [internal] load build context
=> => transferring context: 87B
=> CACHED [ 2/10] WORKDIR /app
=> CACHED [ 3/10] RUN mkdir hello
=> CACHED [ 4/10] RUN echo "Hello World" > "hello/world.txt"
=> CACHED [ 5/10] RUN addgroup -S app_group
=> CACHED [ 6/10] RUN adduser -S -D -h /app app_user app_group
=> CACHED [ 7/10] RUN chown -R app_user:app_group /app
=> CACHED [ 8/10] COPY *.txt hello
=> CACHED [ 9/10] ADD *.gz hello
=> [10/10] RUN mv hello/world.txt hello qwerty.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:8d6f9a1412770afe4cbb5970c4b67a4d738e40493ab8fe611a385952038967d4
=> => naming to docker.io/library/arg.ins
sysadmin@docker:~/image$
```

Using the ARG instruction

As you can see in the image above, the file name changes to `qwerty.txt`, which corresponds to the arguments you wrote in Docker. If you want to change the argument when creating a Docker image, then you can change it by adding the **-build-arg app=pass** option so that the file becomes `pass.txt`, as

shown in the image below:

```
sysadmin@docker:~/image$ docker run -d --name arg_ins arg_ins
c9fdd5fcf683424c40cb20a63dba8a81a73a737feb8e4ba82c31b175fdc2d344
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker container logs arg_ins
cat: can't open 'hello/${app}.txt': No such file or directory
sysadmin@docker:~/image$
```



Using the ARG when creating a Docker image

Based on the image above, you can see that the file name has changed to pass.txt. However, you have to know that when you run the container and run the log command, it will display as shown in the picture below:

```
sysadmin@docker:~/image$ docker run -d --name arg_ins arg_ins
c9fdd5fcf683424c40cb20a63dba8a81a73a737feb8e4ba82c31b175fdc2d344
sysadmin@docker:~/image$
sysadmin@docker:~/image$ docker container logs arg_ins
cat: can't open 'hello/${app}.txt': No such file or directory
sysadmin@docker:~/image$
```



Error when opening the log

You can see in the image above that there is an error in the container log. This is because ARGs can only be accessed at build time, while CMDs are executed at runtime, so if you want to use ARG at runtime, then you need to insert the ARG into ENV, and your Dockerfile will be as below:

FROM alpine:3

LABEL author="sysadmin"

LABEL company="sysadminpedia" website="https://www.sysadminpedia.com"

WORKDIR /app

RUN mkdir hello

RUN echo "Hello World" > "hello/world.txt"

RUN addgroup -S app_group

RUN adduser -S -D -h /app app_user app_group

RUN chown -R app_user:app_group /app

USER app_user

```
COPY *.txt hello

ADD *.gz hello

ENV APP_PORT=8080
EXPOSE ${APP_PORT}

ARG app=qwerty
RUN mv hello/world.txt hello/${app}.txt

CMD ["cat", "hello/${app}.txt"]
```

15. Health CHECK instruction

To tell Docker if the container is still running properly or not. If there is a HEALTHCHECK, the container will automatically have a health status from the beginning with a starting value. If successful, it has a healthy value, and if it fails, it has an unhealthy value.

```
FROM nginx:alpine

RUN addgroup -S app_group \
    && adduser -S -G app_group -h /app app_user

RUN mkdir -p /app /var/cache/nginx /var/log/nginx /run \
    && chown -R app_user:app_group /app /var/cache/nginx /var/log/nginx /run

RUN echo "OK" > /app/healthz.html

RUN cat <<'EOF' > /etc/nginx/conf.d/default.conf
server {
    listen 80;

    location /healthz {
        root /app;
        try_files /healthz.html =404;
    }

    location / {
        return 200 "Hello from Nginx! Everything works fine.\n";
    }
}
EOF

USER app_user

# Healthcheck
```

```
HEALTHCHECK --interval=30s --timeout=5s --start-period=5s --retries=3 \
  CMD wget --no-verbose --tries=1 --spider http://127.0.0.1/healthz || exit 1
```

```
CMD ["nginx", "-g", "daemon off;"]
```

```
sysadmin@docker:~/image$ docker run -d --name health_ins -p 8080:80 health_ins
d5b4bdc7051b1337a6c288419143559f747b0ba60c02de47d8823f6ac32d417e
sysadmin@docker:~/image$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d5b4bdc7051b	health_ins	"/docker-entrypoint..."	3 seconds ago	Up 2 seconds (health: starting)	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	health_ins

```
sysadmin@docker:~/image$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d5b4bdc7051b	health_ins	"/docker-entrypoint..."	10 seconds ago	Up 9 seconds (healthy)	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	health_ins

```
sysadmin@docker:~/image$
```

Using the HEALTHCHECK instruction

Note

Actually, it was the developers who created the Dockerfile as an image for their applications to run on Docker. However, as a sysadmin, you should also understand the instructions in the Dockerfile so that you can help developers if there is an error in their Docker, or maybe also to create a Docker image for sysadmin purposes.

References

youtube.dimas-maryanto.com
youtube.com
stackify.com
devopscube.com
geeksforgeeks.org

How to Configure Crontab in Linux?

written by sysadmin | 8 November 2025

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

Problem

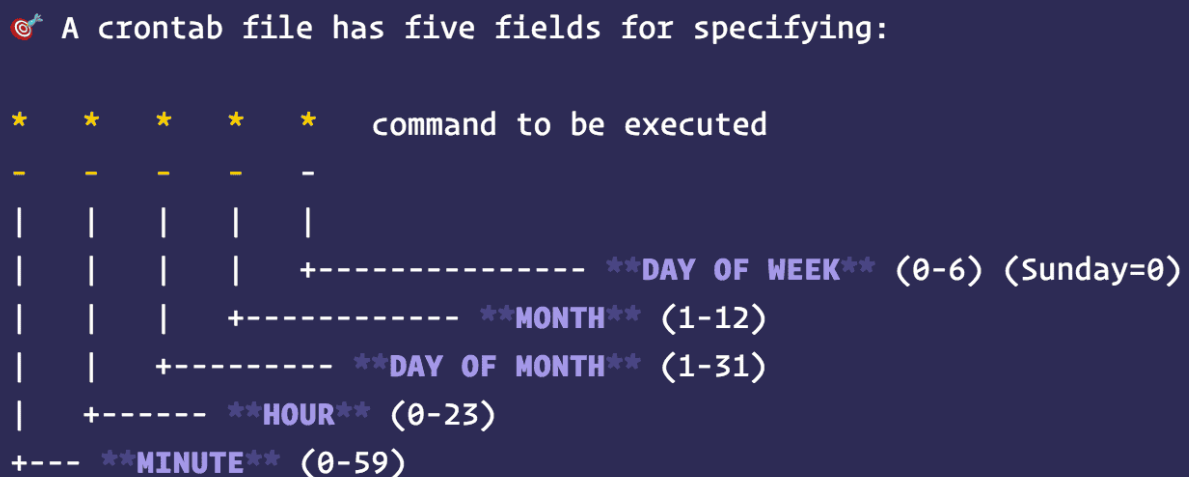
How to configure crontab in Linux?

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 diagram illustrating the syntax of a crontab file. It shows a sequence of five fields, each represented by a yellow star, followed by a space and the text "command to be executed". Below this, a series of vertical lines and horizontal dashes indicate the structure of the time fields. The fields are: DAY OF WEEK (0-6, Sunday=0), MONTH (1-12), DAY OF MONTH (1-31), HOUR (0-23), and MINUTE (0-59). Each field is represented by a vertical line, and the horizontal dashes indicate the range of values for each field.

```
★ ★ ★ ★ ★  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
```

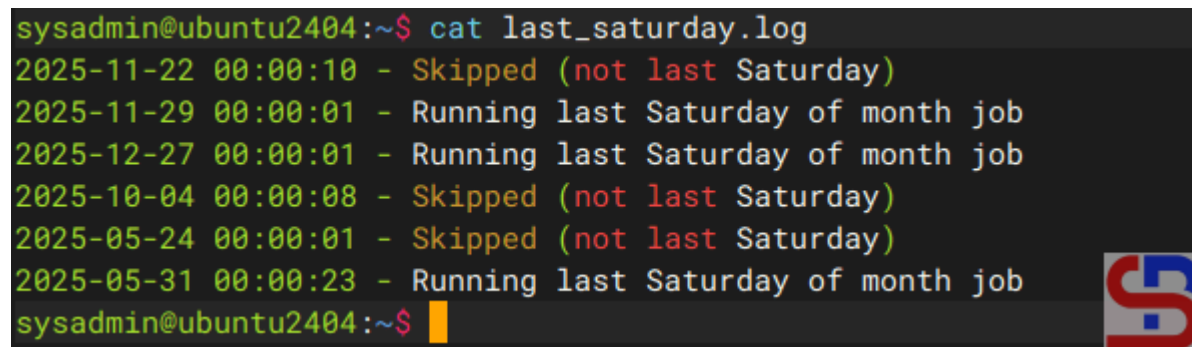
```
TODAY=$(date +%Y-%m-%d)
NEXT_SATURDAY_MONTH=$(date -d "next Saturday" +%m)
CURRENT_MONTH=$(date +%m)
```

```
# 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:~$
```

last_saturday.log

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 Manage a Database and its Table\(s\) in PostgreSQL?](#)

written by sysadmin | 8 November 2025

[The previous article](#) explained how to install a PostgreSQL database on Linux. This article will explain the basics of commands in a PostgreSQL database.

Problem

How to manage a database and its table(s) in PostgreSQL?

Solution

Below are the basic commands of PostgreSQL to manage a database and its table(s):

1. Access to the PostgreSQL database

Use the command below to access PostgreSQL:

```
sudo -u postgres psql
```

```
sysadmin@docker:~$ sudo -u postgres psql
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
Type "help" for help.

postgres=#
```

Access to the PostgreSQL

INFO

The use of capital letters in this article is only to distinguish between original commands from PostgreSQL and data from the user. You don't have to use the capital letters when running these commands, but you can use all lowercase letters.

2. List all databases

Use the command below to list all databases in PostgreSQL:

```
\l
```

```
postgres=# \l

          List of databases
  Name      | Owner   | Encoding | Locale Provider | Collate | Ctype    | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          |
 template0  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres,+
            |          |          |                  |            |            |         |          | postgres=Ctc/postgres
 template1  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |         |          | =c/postgres,+
            |          |          |                  |            |            |         |          | postgres=Ctc/postgres
(3 rows)
```

List all databases

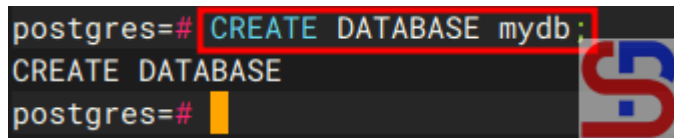
3. Creating a new database

Use the format below to create a new database:

```
CREATE DATABASE database_name;
```

You can see the options for this command [on this page](#). For example, if you want to create a new database called **mydb**, use the command below;

```
CREATE DATABASE mydb;
```

A terminal window with a dark background. The prompt is 'postgres=#'. The command 'CREATE DATABASE mydb;' is entered and highlighted with a red box. Below it, the prompt is 'postgres=#' followed by a yellow cursor bar. To the right of the terminal window is a logo consisting of a stylized 'S' with a blue and red design.

Create a new database

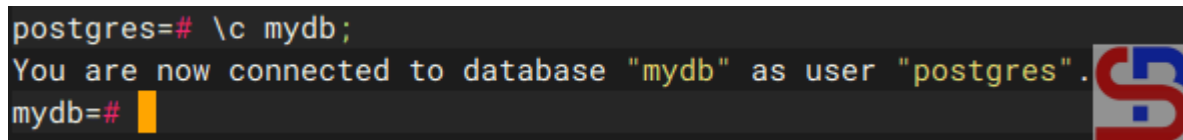
4. Connect to the database

Use the format below to connect to the database:

```
\c db_name;
```

For example, if you want to connect to the mydb database, type the following command:

```
\c mydb;
```

A terminal window with a dark background. The prompt is 'postgres=#'. The command '\c mydb;' is entered. The response is 'You are now connected to database "mydb" as user "postgres".'. Below it, the prompt is 'mydb=#' followed by a yellow cursor bar. To the right of the terminal window is a logo consisting of a stylized 'S' with a blue and red design.

Connect to the database

5. Create a table

Use the format below to create a new table:

```
CREATE TABLE table_name (name_of_column1 column_data_type1, name_of_column2  
column_data_type2, ...);
```

You can see the options for this command [on this page](#). Type the command below to create an employee table:

```
CREATE TABLE employee (name varchar (100), age int);
```

```
mydb=# CREATE TABLE employee (name varchar (100), age int);  
CREATE TABLE  
mydb=#
```

Create a table

6. Display all tables

Use the command below to display all the tables:

```
\dt
```

```
mydb=# \dt  
          List of tables  
Schema |   Name   | Type  | Owner  
-----+-----+-----+-----  
public | employee | table | postgres  
(1 row)  
  
mydb=#
```

Display all table

7. Display the table structure

Use the format below to display the table structure:

```
\d table_name;
```

For example, if you want to display the employee table structure, run the command below:

```
\d employee;
```

```
mydb=# \d employee  
          Table "public.employee"  
Column |          Type          | Collation | Nullable | Default  
-----+-----+-----+-----+-----  
name   | character varying(100) |           |          |  
age    | integer                |           |          |  
  
mydb=#
```

Describe a table

If you want to display more information about the table, type the command below:

```
\d+ employee;
```

```
mydb=# \d+ employee
```

Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description
name	character varying(100)				extended			
age	integer				plain			

Access method: heap

```
mydb=#
```

Display more information in a table

8. Add the column

Use the format below to make a column in the table:

```
ALTER TABLE table_name ADD column column_name type(nnn);
```

You can see the options for this command [on this page](#). For example, if you want to add to the city column in the employee table, use the command below:

```
ALTER TABLE employee ADD column city varchar(100);
```

```
mydb=# \d employee
```

Column	Type	Collation	Nullable	Default
name	character varying(100)			
age	integer			

```
mydb=# ALTER TABLE employee ADD column city varchar(100);
```

```
ALTER TABLE
```

```
mydb=# \d employee
```

Column	Type	Collation	Nullable	Default
name	character varying(100)			
age	integer			
city	character varying(100)			

```
mydb=#
```

Add a column

9. Insert data into the table

Use the format below to enter data in a table:

```
INSERT INTO table_name (Column1, Column2,..., ColumnN) VALUES (Value1,...ValueN),  
(Value1,...ValueN);
```

You can see the options for this command [on this page](#). Type the command below if you want to insert 2 data to the employee table:

```
INSERT INTO employee (name,age,city) VALUES ('bob',21,'New York'),  
('John',22,'Chicago');
```

```
mydb=# INSERT INTO employee (name,age,city) VALUES ('bob',21,'New York'), ('John',22,'Chicago');  
INSERT 0 2  
mydb=#
```

Insert data

INFO

If you want to insert a value in the form of a number, the number does not have to be flanked with an apostrof ('...') sign, whereas if it is a character or a combination of characters and numbers, it must be flanked with an apostrof ('...').

10. Displays data in a table

Use the format below to display all data in a table:

```
SELECT option1 FROM table_name option2;
```

You can see the options for this command [on this page](#). For example, use the command below to display all data in the employee table:

```
SELECT * from employee;
```

```
mydb=# SELECT * from employee;
 name | age | city
-----+-----+-----
  bob  |  21 | New York
  John  |  22 | Chicago
(2 rows)
```



Display the data

11. Update data

Use the format below to update data in a table:

```
UPDATE table_name SET columnX=valueX WHERE columnY=valueY;
```

You can see the options for this command [on this page](#). For example, if you want to update the age of the employee named Bob, use the command below:

```
UPDATE employee SET age=23 WHERE name='bob';
```

```
mydb=# UPDATE employee SET age=23 WHERE name='bob';
UPDATE 1
mydb=#
```



Update the data

12. Delete data

Use the format below to delete one or more rows of a table:

```
DELETE FROM table_name WHERE column=value;
```

You can see the options for this command on [this page](#). For example, if you want to delete the data where the user is in Chicago, use the command below:

```
DELETE FROM employee WHERE city='Chicago';
```

```

mydb=# SELECT * from employee;
 name | age | city
-----+-----+-----
 John |  22 | Chicago
  bob |  23 | New York
(2 rows)

mydb=# DELETE FROM employee WHERE city='Chicago';
DELETE 1
mydb=# SELECT * from employee;
 name | age | city
-----+-----+-----
  bob |  23 | New York
(1 row)

mydb=#

```



Delete the data

13. Delete the column

Use the format below to make changes in the table:

```
ALTER TABLE db_name DROP COLUMN column_name;
```

If you want to delete the column, for example, city, you can use the following command:

```
ALTER TABLE employee DROP COLUMN city;
```

```
mydb=# \d employee
          Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
 city   | character varying(100) |           |          |

mydb=# ALTER TABLE employee DROP column city;
ALTER TABLE
mydb=# \d employee
          Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
```

Delete the column

14. Empty the table

You can delete all data in a table with the format as below:

```
TRUNCATE table_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete all the data in the employee table, type the command below:

```
TRUNCATE employee;
```

```

mydb=# SELECT * from employee;
 name | age
-----+-----
  bob  |  23
(1 row)

mydb=# TRUNCATE employee;
TRUNCATE TABLE
mydb=# SELECT * from employee;
 name | age
-----+-----
(0 rows)

mydb=#

```



Truncate the table

15. Change a table name

You can change the name of the table using the format below:

```
ALTER TABLE old_table_name RENAME TO new_table_name;
```

You can see the options for this command [on this page](#). Use the command below if, for example, you want to change the name of the employee table to employees:

```
ALTER TABLE employee RENAME TO employees;
```

```

mydb=# \dt
 public | employee | table | postgres

mydb=# ALTER TABLE employee RENAME TO employees;
ALTER TABLE
mydb=# \dt
 public | employees | table | postgres

mydb=#

```



Rename the table

16. Delete a table

Use the format below to delete a table:

```
DROP TABLE table_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete the employee table, use the command below:

```
DROP TABLE employees;
```

```
mydb=# \dt
public | employees | table | postgres

mydb=# DROP TABLE employees;
DROP TABLE
mydb=# \dt
Did not find any tables.
mydb=#
```

Delete the table

17. Delete a database

To delete a database, use the format below:

```
DROP DATABASE database_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete the mydb database, type the command below:

```
DROP DATABASE mydb;
```

```
mydb=# DROP DATABASE mydb;
ERROR: cannot drop the currently open database
mydb=# \c postgres
You are now connected to database "postgres" as user "postgres".
postgres=# DROP DATABASE mydb;
DROP DATABASE
postgres=#
```

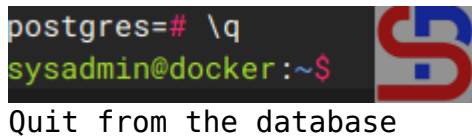
Delete the database

17. Quit from the database

To quit from the database, run the command below:

```
\q
```

```
postgres=# \q
sysadmin@docker:~$
```



Quit from the database

Note

Actually, the basic commands for PostgreSQL and [MariaDB](#) are almost the same, with only there is a slight difference between the two databases.

References

hasura.io
postgresql.org
w3schools.com
geeksforgeeks.org

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

written by sysadmin | 8 November 2025

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
```

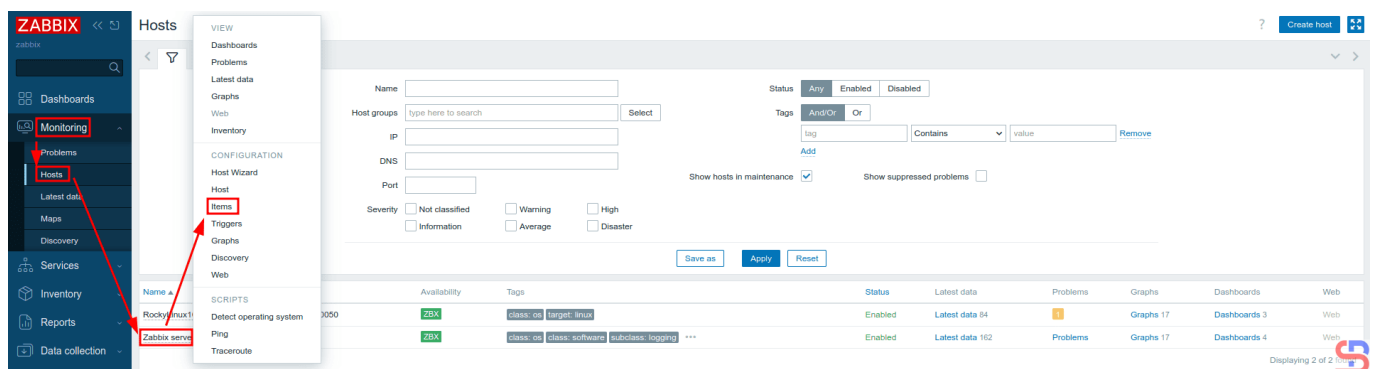
4. Restart zabbix_agent

Restart the Zabbix agent using the following commands:

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

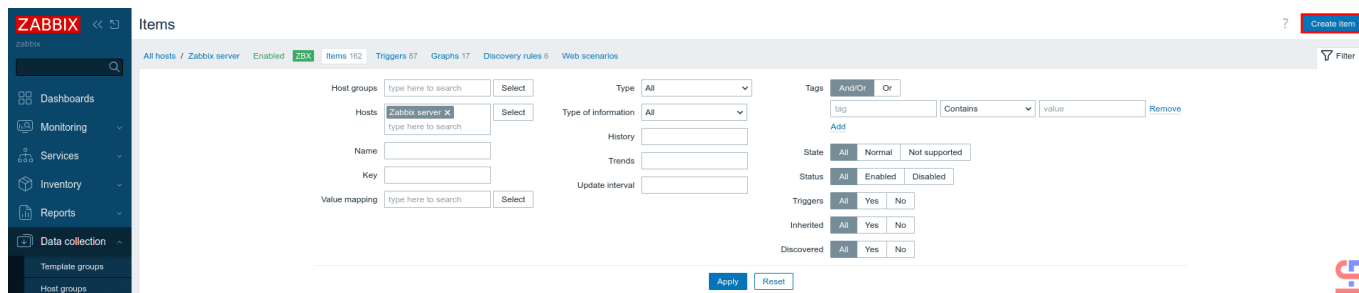
5. Configure Zabbix

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

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

New item

Item Tags Preprocessing

* Name Sysadminpedia site

Type Zabbix agent

* Key check-sysadminpedia-site [Select](#)

Type of information Numeric (unsigned)

* Host interface 127.0.0.1:10050

Units

* Update interval 1m

Custom intervals

Type	Interval	Period
Flexible	Scheduling	50s
		1-7,00:00-24:00

[Add](#) [Remove](#)

* Timeout Global Override 3s [Timeouts](#)

* History Do not store Store up to 31d

* Trends Do not store Store up to 365d

Value mapping type here to search [Select](#)

Populates host inventory field -None-

Description

Enabled ☒

[Add](#) [Test](#) [Cancel](#)

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 127.0.0.1

Port 10050

Test with **Server** Proxy

Get value

Value value

Time now

☐ Not supported

Error error text

Previous value

Prev. time

End of line sequence **LF** CRLF

Get value and test

Cancel

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 127.0.0.1

Port 10050

Test with **Server** Proxy

Get value

Value 1

Time now

☐ Not supported

Error error text

Previous value

Prev. time

End of line sequence **LF** CRLF

Result Result converted to Numeric (unsigned)

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:

New item

? X

Item Tags Preprocessing

* Name Sysadminpedia site

Type Zabbix agent

* Key check-sysadminpedia-site Select

Type of information Numeric (unsigned)

* Host interface 127.0.0.1:10050

Units

* Update interval 1m

Custom intervals

Type	Interval	Period
Flexible	Scheduling	50s
		1-7,00:00-24:00

Add Remove

* Timeout Global Override 3s Timeouts

* History Do not store Store up to 31d

* Trends Do not store Store up to 365d

Value mapping type here to search Select

Populates host inventory field -None-

Description

Enabled ☒

Add Test Cancel

Click the Add button

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

ZABBIX

Items

Item added

All hosts / Zabbix server Enabled 2BX Items 163 Triggers 87 Graphs 17 Discovery rules 0 Web scenarios

Host groups type here to search Select

Hosts Zabbix server X type here to search Select

Name

Key

Value mapping type here to search Select

Type All

Type of information All

History

Trends

Update interval

Tags And/Or Or tag Contains value Remove

Add

State All Normal Not supported

Status All Enabled Disabled

Triggers All Yes No

Inherited All Yes No

Discovered All Yes No

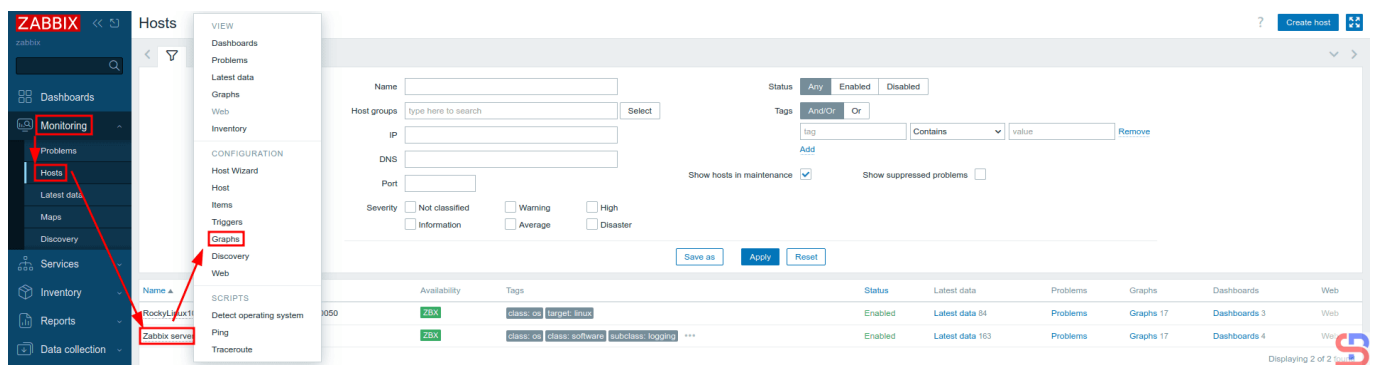
Apply Reset

Succeed in adding an Item

6. Create a graph

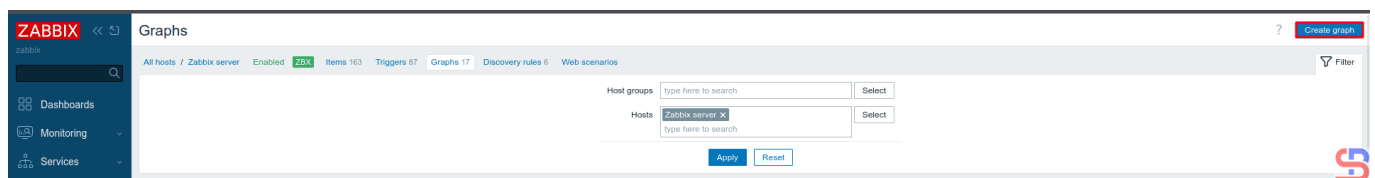
Then, create a graph for the result of the script by selecting the host that will display the result of the bash script. I choose to use the Zabbix server host: **Monitoring >**

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:

New graph

Graph **Preview**

* Name: Sysadminpedia site

* Width: 900

* Height: 200

Graph type: Normal

Show legend: ☒

Show working time: ☒

Show triggers: ☒

Percentile line (left): ☐

Percentile line (right): ☐

Y axis MIN value: Calculated

Y axis MAX value: Calculated

* Items

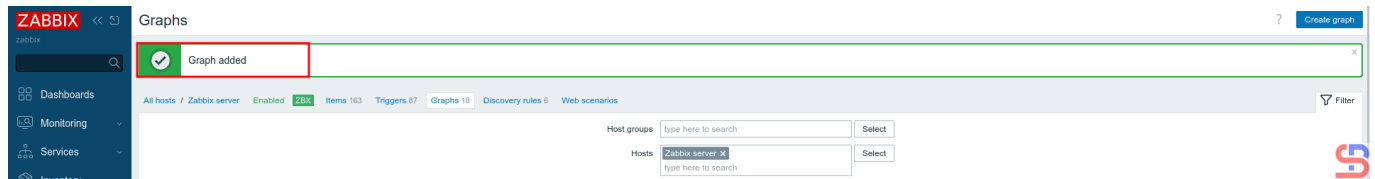
	Name	Function	Draw style	Y axis side	Color	
1:	Zabbix server: Sysadminpedia site	all	Line	Left		Remove

Add

Add Cancel

Click the **Add** button

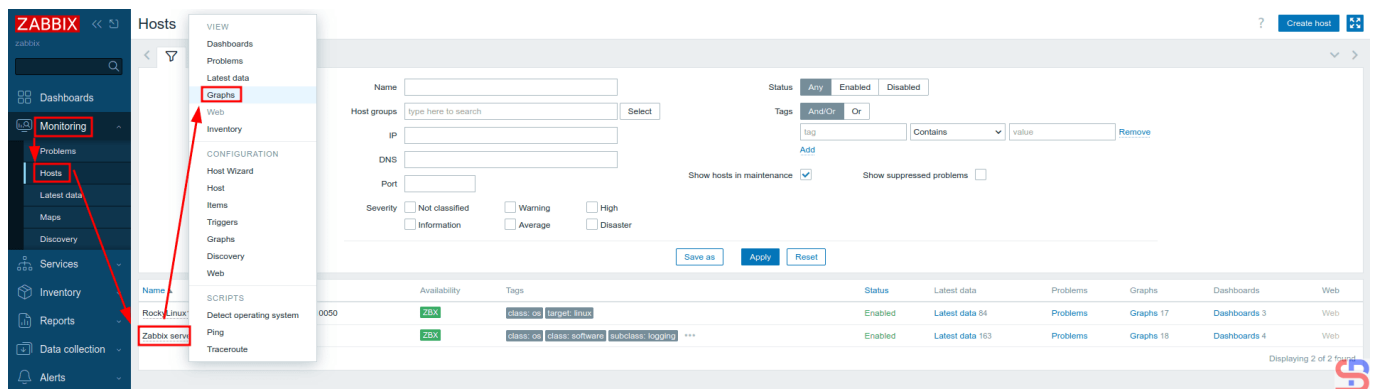
After that, click the **Add** button then there will be the text **Graph Added** as in the image below:



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:



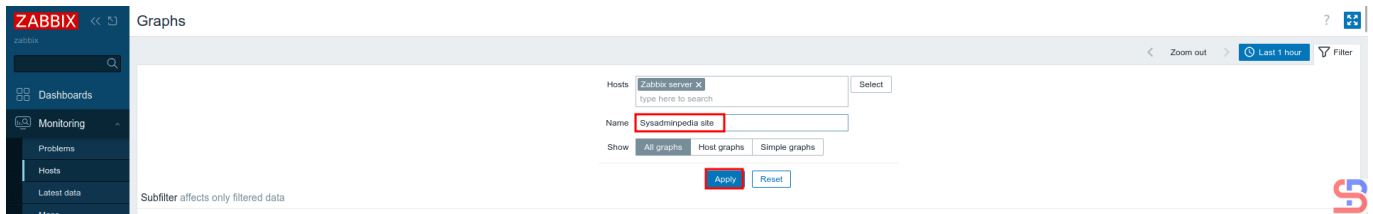
Click the Graphs in the VIEW section

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



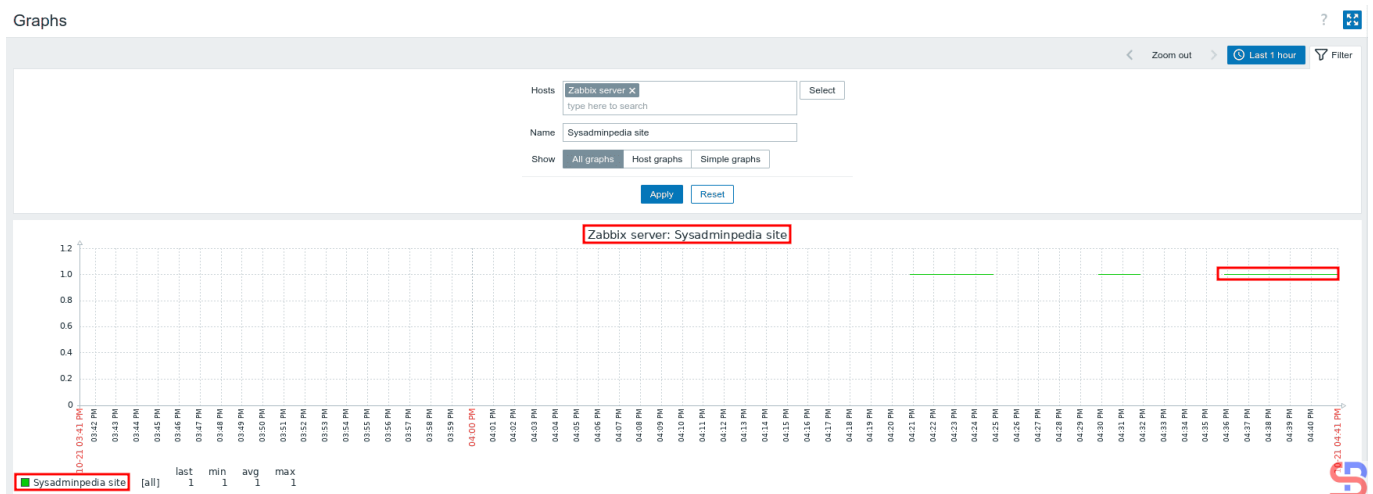
Click the Filter button

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



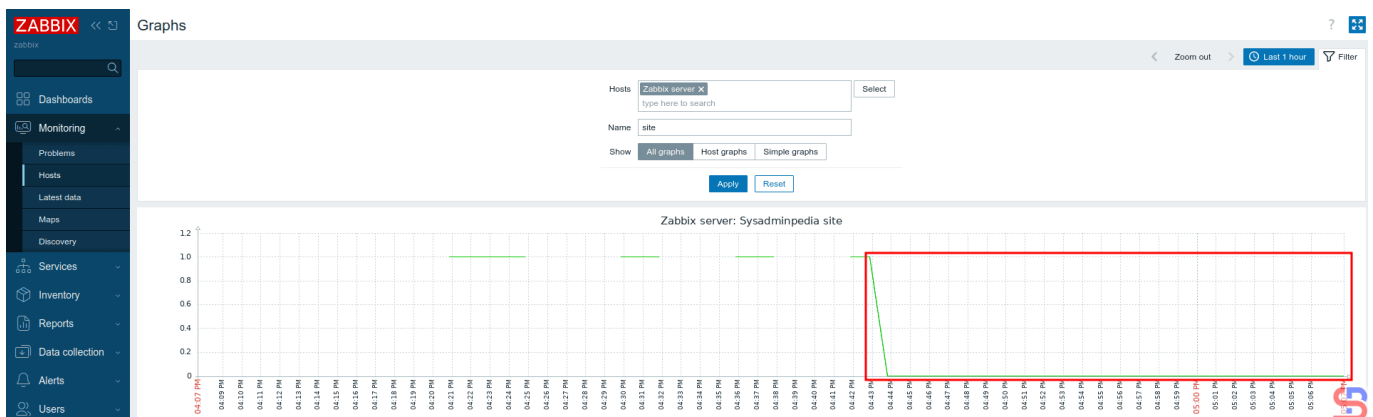
Type the name of the graph and click the Apply button

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 | 8 November 2025

[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:

```
excludepks=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
```

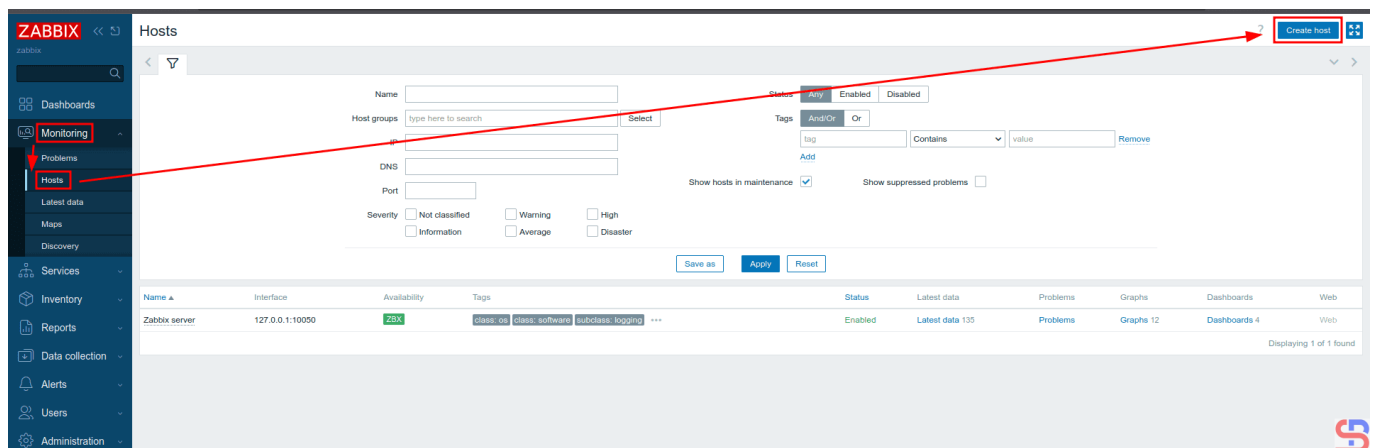
Then run the two commands below:

```
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:

New host

? X

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name **RockyLinux10**

Visible name RockyLinux10

Templates **Linux by Zabbix agent**
type here to search

* Host groups **Linux servers**
type here to search

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
	Agent	192.168.56.104		☒ IP ☐ DNS	10050	☒ Remove

[Add](#)

Description Monitor host RockyLinux10

Monitored by ☒ Server ☐ Proxy ☐ Proxy group

Enabled ☒



Configure a new host in Zabbix

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

Hosts ? Create host

Host added

Name Status

Host groups type here to search

Tags tag Contains value

Show hosts in maintenance ☒ Show suppressed problems ☐

Severity ☐ Not classified ☐ Warning ☐ High
☐ Information ☐ Average ☐ Disaster

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

After the Host added

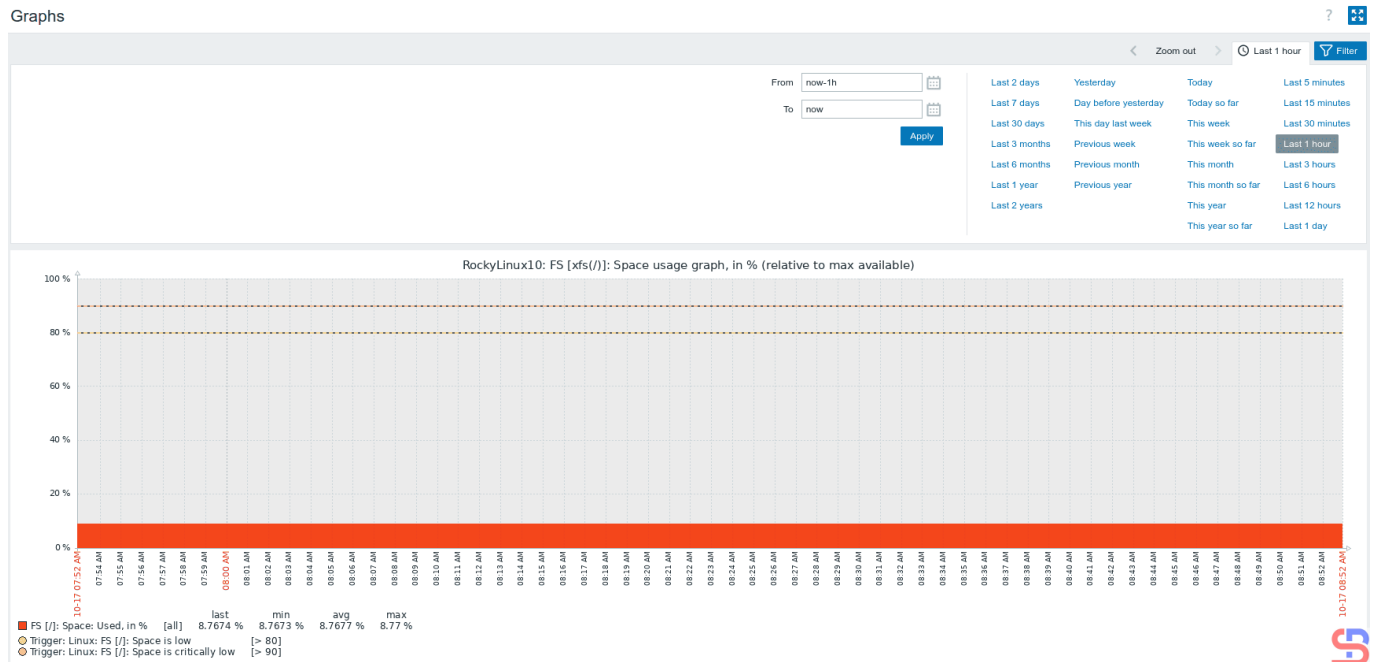
Wait a few moments, and the Zabbix application should be able to monitor your host, which is marked with the word **ZBX** in green, as in the image below:

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 54	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

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

How to Install Zabbix On Ubuntu?

written by sysadmin | 8 November 2025

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:

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Welcome to

Zabbix 7.4

Default language

Back

Next step

Configure Zabbix using your browser

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

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Check of pre-requisites

	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.

Database type

MySQL

Database host

localhost

Database port

0

0 - use default port

Database name

zabbix

Store credentials in

Plain text

HashiCorp Vault

CyberArk Vault

User

zabbix

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:



Settings

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Zabbix server name

zabbix

Default time zone

System: (UTC+00:00) UTC

Default theme

Blue

Encrypt connections from Web interface

☐

Back

Next step



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

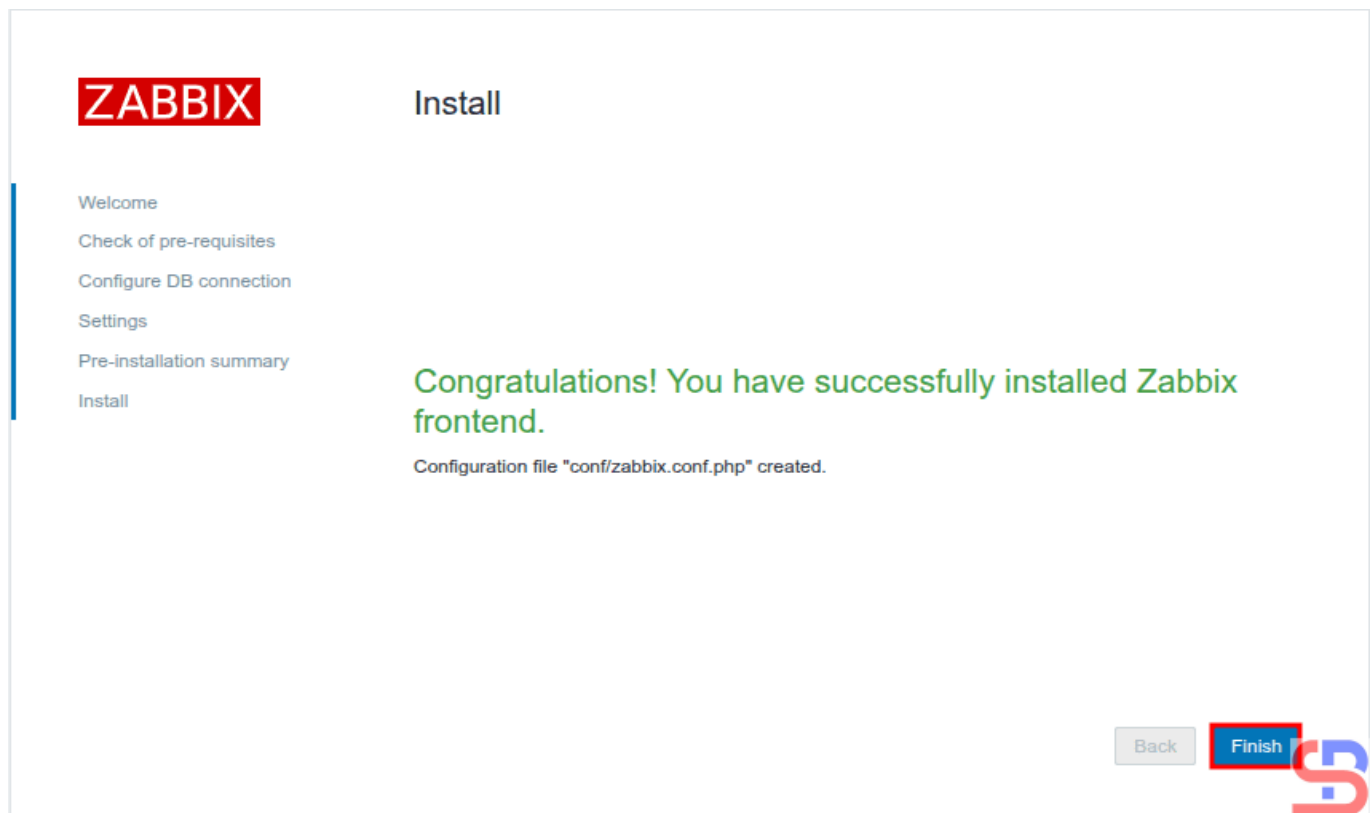
Back

Next step



Pre-installation summary

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



Finish installation

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

ZABBIX

Username

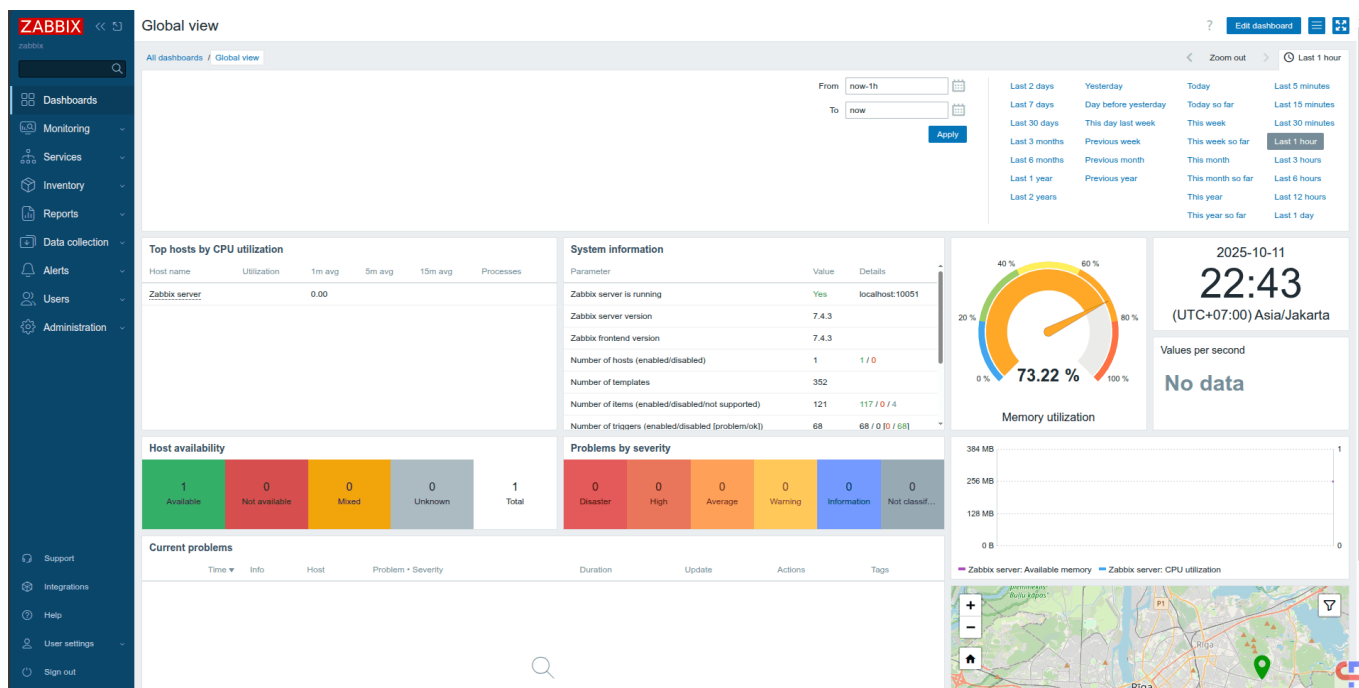
Password

☒ Remember me for 30 days

Sign in

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