

How to Install Exalate for Zendesk on Docker

Last Modified on 03/18/2024 5:38 am EDT

You can host Exalate for Zendesk on your own server. To do so, you need to install Exalate on Docker.

Note: You need to install Docker. Check the [docker documentation](#) for more details.

How to Install Exalate for Zendesk on Docker

1. Create directory and create docker-compose.yml file

Create a directory to hold the docker-compose file:

```
cd ~  
mkdir exalate-zendesknode
```

Create or download a **docker-compose.yml** file

Note: Click [docker-compose.yml](#) to download the file.

Important: We recommend using the latest version of Exalate for Zendesk. It can be found in the [Release History](#). Enter the latest version in the `image` tag. For example, in `image: idalko/zendesknode:5.6.0`, the version of Exalate for Zendesk is `5.6.0`.

The **docker-compose.yml** file should contain the following information in it:

```

version: '2'

services:
  database:
    restart: unless-stopped
    image: postgres:9.4
    volumes:
      - ./persist/db:/var/lib/postgresql/data
      - ./createdb.sh:/docker-entrypoint-initdb.d/init-user-db.sh
    environment:
      - POSTGRES_PASSWORD=changeme
      - DB_NAME=zendesknode
      - DB_USER=idalko
      - DB_PASS=idalko
    networks:
      - dbnet

  zendesknode:
    restart: unless-stopped
    ports:
      - 9302:9002

    #
    # Change the image tag to the required version
    # Check Release History on docs.exalate.com for an overview
    #
    image: idalko/zendesknode:5.6.0
    depends_on:
      - database #wait for postgres to be started, not for ready
    volumes:
      - ./persist/home:/opt/zendesknode/data
    environment:
      # Add your environment settings here
      - ZENDESKNODE_PG_HOST=database
      - ZENDESKNODE_PG_DB=exalate?gssEncMode=disable
      - ZENDESKNODE_PG_USER=idalko
      - ZENDESKNODE_PG_PWD=idalko
      - ZENDESKNODE_PORT=9302

    #As part of the security improvements, Exalate 5.5.0 validates the origin header
    #that the browser is sending upon every request to Exalate.
    #In order to validate the origin header, Exalate needs to know what is the URL
    #leading to it.
    #When deploying Exalate to Docker one needs to put an environment variable NODE_SELF_URL
    .
    #Example:
    # When you deploy Exalate onto a server, you configure a DNS rule such that
    #whenever people navigate to foo.com, they reach your server's Exalate.
    #You set up SSL so that https://foo.com leads to your Exalate on your server.
    #Now you need to set environment variable

    - NODE_SELF_URL=https://foo.com

    #for your Exalate docker container.

    networks:
      - dbnet
      - default

networks:
  dbnet:
    driver: bridge
  default:
    driver: bridge

```

Connecting to Postgres 10 or Higher

For unencrypted connections from Exalate to a Postgres version 10 or higher, you need to disable `gssEncMode` with the following setting:

```

# exalate is the name of the database on the postgres instance
#
ZENDESKNODE_PG_DB=exalate?gssEncMode=disable

```

2. Ensure that a correct database is setup using a createdb.sh

Create or download a **createdb.sh** file (referenced from docker-compose.yml):

Note: Click [createdb.sh](#)  to download the file.

The file should contain the following information:

```
#!/bin/bash

TEST=`psql -U postgres <<-EOSQL
SELECT 1 FROM pg_database WHERE datname='$DB_NAME';
EOSQL`

echo "*****CREATING DOCKER DATABASE*****"
if [[ $TEST == "1" ]]; then
    # database exists
    # $? is 0
    exit 0
else
    psql -U postgres <<-EOSQL
    CREATE ROLE $DB_USER WITH LOGIN ENCRYPTED PASSWORD '${DB_PASS}' SUPERUSER;
    EOSQL

    psql -U postgres <<-EOSQL
    CREATE DATABASE $DB_NAME WITH OWNER $DB_USER ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE template0
    ;
    EOSQL

    psql -U postgres <<-EOSQL
    GRANT ALL PRIVILEGES ON DATABASE $DB_NAME TO $DB_USER;
    EOSQL
fi

echo ""
echo "*****DOCKER DATABASE CREATED*****"
```

Ensure that the volumes are included in your backup strategy:

- persist

3. Set Environment Variables if necessary

Below, you can find the environment variables used for the app container. All of them are optional, and in the given example, we've overridden ZENDESKNODE_PG_DB, ZENDESKNODE_PG_USER, and ZENDESKNODE_PG_PWD just to show how to pass different credentials to the Exalate application.

Full list of environment variables:

Variable name	Default value	Example	Description
HTTP_HEADER	n/a	HTTP_HEADERS="TestName1: testAddHeader1"	Allows a header to be added between requests through the Postman application
ZENDESKNODE_PG_HOST	ZENDESKNODE_PG_HOST=database	ZENDESKNODE_PG_HOST=db.acme.com	Tells the application where the Postman database is hosted
ZENDESKNODE_PG_DB	ZENDESKNODE_PG_DB=exalate	ZENDESKNODE_PG_DB=exalate	Tells the application which database the Postman application uses to Exalate
ZENDESKNODE_PG_USER	ZENDESKNODE_PG_USER=idalko	ZENDESKNODE_PG_USER=exalate	Tells the application which user the Postman application uses to Exalate queries
ZENDESKNODE_PG_PWD	ZENDESKNODE_PG_PWD=idalko	ZENDESKNODE_PG_PWD=secret	Tells the application which password the Postman application uses to perform queries


Variable name	Default value	Example	Description
ZENDESKNODE_PORT	ZENDESKNODE_PORT=9000	ZENDESKNODE_PORT=8080	<p>Tells which port the application will listen on.</p> <p><i>Note that the default port (9000) is not available on all operating systems. You should check the port availability on your system.</i></p> <p>port: 9000</p> <p>should be 8080</p> <p>port: 8080</p>
ZENDESKNODE_SMTP_HOST_NAME	ZENDESKNODE_SMTP_HOST_NAME=mail.server.com	ZENDESKNODE_SMTP_HOST_NAME=smtp.gmail.com	is used to connect to the SMTP server about email synchronization
ZENDESKNODE_SMTP_PORT	ZENDESKNODE_SMTP_PORT=465	ZENDESKNODE_SMTP_PORT=587	is used to connect to the SMTP server about email synchronization
ZENDESKNODE_SMTP_FROM	ZENDESKNODE_SMTP_FROM=admin@admin.com	ZENDESKNODE_SMTP_FROM=my.name@gmail.com	is used to set the email address about email synchronization
ZENDESKNODE_SMTP_USER	ZENDESKNODE_SMTP_USER=admin	ZENDESKNODE_SMTP_USER=my.name	is used to set the email user about email synchronization
ZENDESKNODE_SMTP_PASS	ZENDESKNODE_SMTP_PASS=1234567	ZENDESKNODE_SMTP_PASS=secret	is used to set the email password about email synchronization
Using a Proxy for Outgoing Connections <p>Whenever the Exalate node needs to use a proxy to establish outgoing connections, use the following parameters in the environment (naming should be obvious):</p>			
ZENDESKNODE_SMTP_TLS <ul style="list-style-type: none"> • PROXY_HTTP_PORT • PROXY_HTTPS_HOST • PROXY_HTTPS_PORT 	ZENDESKNODE_SMTP_TLS=true	ZENDESKNODE_SMTP_TLS=true	is used to connect to the SMTP server about email synchronization but there is no need to set the ZENDESKNODE_SMTP_PASS to SSL and

4. Start the Application

```
cd ~/exalate-zendesknode
docker-compose up -d
```

Verify the installation

After performing the aforementioned steps and checking that the container is up, you should be able to access the Exalate console via <http://localhost:9302>



Registration

Tracker URL *

Exalate URL *

Save

[Documentation](#)
[EULA](#)
[Support](#)
[Report a bug](#)

Note: You may need to set up [local port forwarding](#) in order to get this to work.

5. Register the Node

To be able to fully use the functionality of your new node, it needs to be registered on the mapper. This mapper acts as a DNS server, mapping tracker URLs to node URLs.

Please raise [a ticket on the support portal](#) providing the following:

- URL of the Zendesk instance.
- URL of the Exalate node which has been deployed on-premise.

How to Manage the Application on Docker

Run Queries to the Application's Database

```
cd ~/exalate-zendesknode
docker exec -it exalate-zendesknode_database_1 bash
su postgres
psql -A $DB_NAME
```

You can find all tables using PSQLs \dt+ command:

```
\dt+
```

All the Postgres SQL queries are permitted

To exit the application's DB:

```
\q
# \q exits the psql
exit
# exits the postgres user session
exit
# exits the exalate-zendesknode_database_1 bash session
```

Inspect the Application's Filesystem

```
cd ~/exalate-zendesknode
docker exec -it exalate-zendesknode_zendesknode_1 bash
```

Remove the application

```
cd ~/exalate-zendesknode
docker-compose rm
```

Remove the application data

Warning: do this only if you want to delete all the synchronization information, including the current synchronizations enqueued to be performed, and synchronization status. Ensure that the remote side you exalate issues with knows that you're stopping synchronization and is ready to handle synchronization errors.

```
cd ~/exalate-zendesknode
# docker volume ls | grep exalate-zendesknode_vol | awk '{ print $2 }' | xargs docker volume r
m
docker volume rm exalate-zendesknode_voldatabase
docker volume rm exalate-zendesknode_volzendesknode
```

System Administration Tasks

With the Exalate for Jira Cloud is running on your environment, you are be required to do the mandatory system administration tasks

- Backup (& restore tests)
- Disaster recovery procedure
- Upgrades whenever needed

Note: Please note that an Exalate version has a lifespan of 2 years. This is to ensure backward compatibility over the whole platform. There are regular new versions deployed which contain bug fixes, security-related improvements, and even new features. Watch the [release notes](#) page for any new versions.

Upgrading Exalate on Docker

If you need to upgrade Exalate on Docker, here are the steps to follow:

1. Edit the YAML File:

Open the `docker-compose.yml` file in a text editor and modify the image tag for the service you wish to upgrade.

```
# use the latest version https://hub.docker.com/r/idalko/zendesknode
image: idalko/zendesknode:latest
depends_on:
  - database #wait for postgres to be started, not for ready
```

Replace `latest` with the latest or desired version tag.

2. Pull the Latest Image:

From the directory containing your `docker-compose.yml` file, pull the latest image.

```
docker-compose pull
```

3. Recreate the Container:

Using Docker Compose, you can easily recreate the container with the new image.

```
docker-compose up -d
```

The `-d` flag runs the containers in detached mode. Docker Compose automatically stops the old container and starts a new one based on the updated image.

4. Post-Upgrade Checks:

After starting the upgraded container, check to make sure everything is running as expected:

- Log into the Exalate interface and verify that all your configurations, connections are intact.
- Test out a few synchronizations to make sure they work as expected.
- Check for any errors in the Docker logs or the Exalate logs.

Troubleshooting

Issues during the installation of the Exalate server for Zendesk

If you have issues during the installation of the Exalate app for Zendesk, you can find logs describing possible problems inside `/tmp`.

The name for the file is generated randomly and automatically by the OS, but you can find the file by the creation date.

Issues while running the Exalate server for Zendesk

Logs are generated under the directory: `/opt/zendesknode/data/logs`.

Refer to these logs to get more information about possible issues and communicate with our support if you need any assistance.

Support

Please read our [Support](#) options.

ON THIS PAGE

[How to Install Exalate for Zendesk on Docker](#)

[How to Manage the Application on Docker](#)

[Product](#)

[System Administration Tasks](#)

[Release History](#)

[Upgrading Exalate on Docker](#)

[Glossary](#)

[API Reference](#)

[Security](#)

[Pricing and Licensing](#)

[Resources](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

[Still need help?](#)

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)

