How to Install Exalate for ServiceNow on Docker

Last Modified on 06/06/2024 9:27 am EDT

You can host Exalate for Servicenow 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 ServiceNow

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

Create a directory to hold the docker-compose file:

```
cd ~
mkdir exalate-snownode
```

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

> **Important**: We recommend using the latest version of Exalate for ServiceNow. It can be found in the Release History.
>
> Enter the latest version in the `image` tag. For example, in `image: idalko/snownode:5.12.0`, the version of Exalate for ServiceNow is `5.12.0`.

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

```
version: '2'

services:
  database:
    restart: unless-stopped
    image: postgres:12.19
    volumes:
      - ./persist/db:/var/lib/postgresql/data
      - ./createdb.sh:/docker-entrypoint-initdb.d/init-user-db.sh
    environment:
      - POSTGRES_PASSWORD=changeme
      - DB_NAME=snownode
      - DB_USER=idalko
      - DB_PASS=idalko
    networks:
      - dbnet

  snownode:
    restart: unless-stopped
    ports:
      - 9000:9000

    #
    # Change the image tag to the required version
    # Check Release History on docs.exalate.com for an overview
    #
    image: idalko/snownode:5.12.0
    depends_on:
      - database #wait for postgres to be started, not for ready
    volumes:
      - ./persist/home:/opt/snownode/data
    environment:
      # Add your environment settings here
      - SNOWNODE_PG_HOST=database
      - SNOWNODE_PG_DB=snownode?gssEncMode=disable
      - SNOWNODE_PG_USER=idalko
      - SNOWNODE_PG_PWD=idalko
      - SNOWNODE_PORT=9000

      #As part of the security improvements, Exalate 5.6.0 and above 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
#
SNOWNODE_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' TEMPL
ATE 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 snownode_PG_DB, snownode_PG_USER, and snownode_PG_PWD just to show how to pass different credentials to the Exalate application.

Full list of environment variables:

Variable name	Example	Description

Variable name	Example	Description
HTTP_HEADER	HTTP_HEADERS="TestName1: testAddHeader1"	Allows additional information to pass between the clients and the server through the request header.
SNOWNODE_PG_HOST	SNOWNODE_PG_HOST=database	Tells Exalate where is the Postgres database to connect is hosted
SNOWNODE_PG_DB	SNOWNODE_PG_DB=exalate	Tells Exalate what is the Postgres database name for the Exalate application
SNOWNODE_PG_USER	SNOWNODE_PG_USER=exalate	Tells the Exalate application what is the Postgres database username for the Exalate application to perform queries with
SNOWNODE_PG_PWD	SNOWNODE_PG_PWD=secret	Tells the Exalate application what is the Postgres database user's password for the Exalate application to perform queries.

SNOWNODE_PORT	SNOWNODE_PORT=80	Tells what is the port to start the Exalate application on. Note that this is the port within the exalatesnownode_snownode_1 container, thus if this variable is changed (for example to 80), the
		`ports:`
		`- 9000:9000`
		should also be changed to:
		`ports:`
		`- 80:80`

SMTP_HOST_NAME	SMTP_HOST_NAME=smtp.gmail.com	The hostname of the SMTP server used to send error notifications
SMTP_PORT	SMTP_PORT=587	Port (also check the TLS setting)
SMTP_FROM	SMTP_FROM=my.name@gmail.com	Email that is used to send error notifications
SMTP_LOGIN	SMTP_LOGIN=my.name	Login to the SMTP service

Variable name	Example	Description
SMTP_PASS	SMTP_PASS=secret	Password to the SMTP service
SMTP_TLS	SMTP_TLS=true	Can be set to false, but then the snownode_SMTP_PORT should be set to the port, that accepts non-SSL and non-TLS connections
POLL_PAIR_TIME	POLL_PAIR_TIME=90	Tells Exalate to set the polling frequency to 90 seconds for PAIR events
POLL_UPDATE_TIME	POLL_UPDATE_TIME=90	Tells Exalate to set the polling frequency to 90 seconds for UPDATE events

Deprecated

The following fields are not valid anymore starting from version 5.0.28 onward:

Full list of environment variables

Variable name	Example	Description
snownode_SMTP_HOST_NAME	snownode_SMTP_HOST_NAME=smtp.gmail.com	is used to send email notifications about errors blocking synchronization
snownode_SMTP_PORT	snownode_SMTP_PORT=587	is used to send email notifications about errors blocking synchronization
snownode_SMTP_FROM	snownode_SMTP_FROM=my.name@gmail.com	is used to send email notifications about errors blocking synchronization
snownode_SMTP_USER	snownode_SMTP_USER=my.name	is used to send email notifications about errors blocking synchronization
snownode_SMTP_PASS	snownode_SMTP_PASS=secret	is used to send email notifications about errors blocking synchronization

Variable name	Example	Description
snownode_SMTP_TLS	snownode_SMTP_TLS=true	is used to send email notifications about errors blocking synchronization. Can be set to false, but then the snownode_SMTP_PORT should be set to the port, that accepts non-SSL and non-TLS connections

Using a Proxy for Outgoing Connections

Whenever Exalate needs to use a proxy to establish outgoing connections, use the following parameters in the environment (naming should be obvious):

- PROXY_HTTP_HOST
- PROXY_HTTP_PORT
- PROXY_HTTPS_HOST
- PROXY_HTTPS_PORT

4. Start the Application

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

Verify your instance

After starting Exalate for ServiceNow you need to verify your instance. For more information on how to verify your Exalate for ServiceNow, please this article.

How to Manage the Application on Docker

Run Queries to the Application's Database

```
cd ~/exalate-snownode
docker exec -it exalatesnowdnode_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 exalatesnownode_database_1 bash session
```

Inspect the Application's Filesystem

```
cd ~/exalate-snownode
docker exec -it exalatesnownode_snownode_1 bash
```

Remove the Application

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

Remove the Application Data

Warning: Do this only if you wish 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 are ready to handle synchronization errors.

```
cd ~/exalate-snownode
# docker volume ls | grep exalatesnownode_vol |  awk '{ print $2 }' | xargs docker volume rm
docker volume rm exalatesnownode_voldatabase
docker volume rm exalatesnownode_volsnownode
```

System Administration Tasks

With the Exalate for Jira Cloud is running on your environment, you are also 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/snownode
image: idalko/snownode: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 start 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

Problems during the installation of the Exalate server for Snownode

If you have problems during the installation of the Exalate app for Servicenow 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.

Problems while running the Exalate server for Snownode

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

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

Support

Please read our Support options.

ON THIS PAGE