

Docker Variables

Last Modified on 12/23/2025 3:54 pm EST

This article describes all Docker variables you can use when installing a self-hosted Exalate application.

All The Docker Variables

PG_HOST

Usage example

```
SNOWNODE_PG_HOST=database
```

The variable will be different based on the work management system:

- `AZURENODE_PG_HOST` for Azure DevOps
- `HPQCNODE_PG_HOST` for HP ALM/QC
- `SNOWNODE_PG_HOST` for ServiceNow

Description

Where the PostgreSQL database is hosted. Exalate will connect to this database.

PG_DB

Usage example

```
SNOWNODE_PG_DB=exalate
```

The variable will be different based on the work management system:

- `AZURENODE_PG_DB` for Azure DevOps
- `HPQCNODE_PG_DB` for HP ALM/QC
- `SNOWNODE_PG_DB` for ServiceNow

Description

Name of the PostgreSQL database that is used for Exalate.

PG_USER

Usage example

```
SNOWNODE_PG_USER=exalate_admin
```

The variable will be different based on the work management system:

- `AZURENODE_PG_USER` for Azure DevOps
- `HPQCNODE_PG_USER` for HP ALM/QC
- `SNOWNODE_PG_USER` for ServiceNow

Description

Name of the PostgreSQL database user. Exalate will perform queries on behalf of that user.

PG_PWD

Usage example

```
SNOWNODE_PG_PWD=secret
```

The variable will be different based on the work management system:

- `AZURENODE_PG_PWD` for Azure DevOps
- `HPQCNODE_PG_PWD` for HP ALM/QC
- `SNOWNODE_PG_PWD` for ServiceNow

Description

Password of the PostgreSQL database user. Exalate will perform queries on behalf of that user.

NODE_SELF_URL

Usage Example

1. You deploy Exalate on a server.
2. You configure a DNS rule such that whenever people navigate to `foo.com`, they reach your server's Exalate.
3. You set up SSL so that <https://foo.com> leads to your Exalate on your server
4. Now you need to set the environment variable `NODE_SELF_URL=https://foo.com` for your Exalate docker container

```
NODE_SELF_URL=https://github-1-0595ec51.vlad-glazov.exalate.st/
```

Description

If a self-hosted node is accessed through any kind of reverse proxy, the `NODE_SELF_URL` variable **MUST** be set to the URL through which the Exalate node is accessed using the following:

```
NODE_SELF_URL=https://github-1-0595ec51.vlad-glazov.exalate.st/
```

Otherwise, the CORS filter will block POST requests to the node and the app won't work.

PORT

Usage example

```
SNOWNODE_PORT=80
```

The variable will be different based on the work management system:

- `AZURENODE_PORT` for Azure DevOps
- `HPQCNODE_PORT` for HP ALM/QC
- `SNOWNODE_PORT` for ServiceNow

Description

Port that is used to start Exalate.

Note: The same port is used in the `exalatesnownode_snownode_1` container. If this variable is changed, for example to `80`, the `ports` value should be updated from:

```
ports:  
- 9000:9000
```

To:

```
ports:  
- 80:80
```

SMTP_HOST_NAME

Usage example

```
SMTP_HOST_NAME=smtp.gmail.com
```

Description

Host name of the SMTP server used to send error notifications.

SMTP_PORT

Usage example

```
SMTP_PORT=587
```

Description

SMTP port (also check the TLS setting).

SMTP_FROM

Usage example

```
SMTP_FROM=my.name@gmail.com
```

Description

Email that is used to send error notifications.

SMTP_LOGIN

Usage example

```
SMTP_LOGIN=my.name
```

Description

Login to the SMTP service.

SMTP_PASS

Usage example

```
SMTP_PASS=secret
```

Description

Password to the SMTP service.

SMTP_TLS

Usage example

```
SMTP_TLS=true
```

Description

This variable controls if the connection will be encrypted with TLS and SSL protocols. If in `true` -- the connection will be encrypted. If in `false` --the connection will not be encrypted.

Product

Note: If `SMTP_TLS=false`, the `SMTP_PORT` variable should be set to the port that accepts non-SSL and non-TLS connections.

[About Us](#)

[Release History](#)

[Glossary](#)

[API Reference](#)

[Security](#)

[Pricing and Licensing](#)

Resources

[Subscribe for a weekly Exalate hack](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)