

Set up EclecticIQ Platform as a virtual appliance

Virtual appliance configuration guide for system administrators

Last generated: March 06, 2018



©2018 EclecticIQ

All rights reserved. No part of this document may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without written permission from the author, except in the case of a reviewer, who may quote brief passages embodied in critical articles or in a review.

Trademarked names appear throughout this book. Rather than use a trademark symbol with every occurrence of a trademarked name, names are used in an editorial fashion, with no intention of infringement of the respective owner's trademark.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author nor the publisher shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this book.

©2018 by EclecticIQ BV. All rights reserved.
Last generated on Mar 6, 2018

Table of contents

| | |
|--|----|
| Table of contents | 2 |
| EclecticIQ Platform as a virtual appliance configuration guide | 4 |
| Scope | 4 |
| Goal | 4 |
| Audience | 4 |
| Feedback | 4 |
| Before you start | 6 |
| Hardware requirements | 6 |
| Single box | 6 |
| Data mount points | 7 |
| Software requirements | 7 |
| Bundled third party software | 8 |
| VM OS login credentials | 11 |
| EclecticIQ Platform login credentials | 12 |
| Install the platform | 13 |
| Install a VMWare Player VM image | 13 |
| Install Neo4j | 14 |
| Pre-installation checks and tasks | 14 |
| Create a YUM repository for the dependencies | 15 |
| Create the YUM repository configuration file | 15 |
| Install Neo4j | 15 |
| Set up Neo4j | 16 |
| Set the Neo4j environment variables | 16 |
| Check that systemd monitors Neo4j | 16 |
| Configure Neo4j | 17 |
| Disable Neo4j remote shell | 17 |
| Enable the service | 17 |
| Configure Neo4j authentication | 18 |
| Create the Neo4j graph schema | 18 |
| Reload Supervisor | 18 |
| Test Neo4j in the UI | 19 |
| Launch the platform | 20 |
| Start the VM | 20 |
| Get the VM IP address | 20 |
| Go to the platform | 20 |
| Upgrade the platform | 22 |
| Before the upgrade | 23 |
| Disable rules | 23 |
| Exit the platform | 23 |
| Back up your data | 24 |
| Shut down the platform | 24 |
| Normal shutdown | 24 |
| Shutdown before a platform upgrade | 24 |
| Check the prerequisites | 26 |
| Check the configuration | 26 |
| Check third-party configurations | 26 |
| After the upgrade | 27 |
| About proxy settings | 27 |
| About Elasticsearch indexes | 27 |
| Run a final check | 28 |
| Check core processes and services | 28 |
| Check search indexing and graph | 29 |
| Check search indexing and graph availability | 29 |
| Re-enable and run the rules | 30 |

| | |
|----------------------------------|----|
| Reload Supervisor configurations | 30 |
| Install extensions | 31 |

EclecticIQ Platform as a virtual appliance configuration guide

This document guides you through the setup and the configuration of EclecticIQ Platform when you choose to install the product as a virtual appliance running in a virtual machine client.

Scope

This document guides you through the steps you need to carry out to complete the following tasks:

- Install — that is, decompress and save to a target location — the virtual machine containing the platform.
- Launch the platform..
- Upgrade the platform to a newer release.

Goal

After completing these tasks, you'll have achieved the following goals:

- The EclecticIQ Platform is set up and configured in the target system as a **virtual appliance** (https://en.wikipedia.org/wiki/virtual_appliance).
- The platform is ready for use as a virtual appliance inside a virtual machine client/player.

Audience

This document targets the following audience:

- DevOps
- System administrators

Feedback

No one reads manuals, ever. We know.

Yet, we strive to give you clear, concise, and complete documentation that helps you get stuff done neatly.

We are committed to crafting good documentation, because life is too short for bad doc.

We appreciate your comments, and we'd love to hear from you: if you have questions or suggestions, drop us a line and share your thoughts with us!

👉 The Product Team

Before you start

Review these system requirements before installing the platform.

This section covers the necessary hardware and software requirements to set up EclecticIQ Platform as a virtual appliance.

Hardware requirements

Hardware requirements for EclecticIQ Platform can vary depending on the target environment you plan to install the platform to. Therefore, the requirements outlined in this section are general guidelines that work in most cases, but they are not tailored to any specific situation.

Single box

Hardware requirement guidelines for EclecticIQ Platform and related dependencies installation on one target machine.

| HW area | Minimum | Recommended | Notes |
|-----------------------|---------------------|----------------------|--|
| Environment | - | VM/Virtual appliance | |
| CPUs | 4 | 8 | Core count includes HT |
| CPU speed | 2.5 GHz | 2.5 GHz or faster | |
| Memory | 32 GB | 64 GB or more | 16 GB is unsuitable for production. A production environment should feature at least 32 GB memory. Consider expanding it to 64 GB when dealing with, for example, large data corpora ingestion or data-intensive graph visualizations. Operations and tasks carried out through the web-based UI may be memory-intensive: the web browser can use ~1 GB or more, occasionally. Monitor system memory usage to determine if your system may need more memory to operate smoothly. |
| Storage | SATA, 100 IOPS | SSD, 200 IOPS | Local attached storage is preferable to SAN or NAS; platform operations are write-intensive. Recommended IOPS range: 200-500 |
| Drives | 5 | 10 | 10 drives to set up 5 sets of mirrored drives (RAID 1) |
| Drive sizes (GB) | 10, 10, 25, 50, 200 | 20, 20, 50, 75, 300 | Each platform database should be allocated to a dedicated drive for data storage |
| Drive allocation (GB) | 10 | 20 | Root (EclecticIQ Platform + Redis) |

| HW area | Minimum | Recommended | Notes |
|---------------------|----------------------|----------------------|---|
| | 10 | 20 | Log data storage |
| | 25 | 50 | Neo4j, graph database |
| | 50 | 75 | Elasticsearch, searching and indexing |
| | 200 | 300 | PostgreSQL, main data storage |
| Network | 2 network interfaces | 2 network interfaces | 1 interface for production, the other for system management |
| Install size | ~240 GB | ~240 GB | Full install, based on VM image size |

Data mount points

When installing and configuring platform components such as PostgreSQL, Elasticsearch, and Neo4j you need to specify dedicated locations where these products store their data.

The table below shows the recommended mount point paths for each data store.

| Component | Mount point | Minimum size (GB) | Recommended size (GB) |
|---------------|----------------------|-------------------|-----------------------|
| Logs | /var/log | 10 | 20 |
| Neo4j | /media/neo4j | 25 | 50 |
| Elasticsearch | /media/elasticsearch | 50 | 75 |
| PostgreSQL | /media/pgsql | 200 | 300 |

Software requirements

- **VMWare Player** (<https://www.vmware.com/products/player>).
- **VirtualBox** (<https://www.virtualbox.org/>) is currently not supported.
- Internet connectivity.
- A web browser with JavaScript enabled.
Supported web browser: Google Chrome (latest)
- DNS name of the host machine you are going to use to access the platform.
Example: \${platform_host}
- SSL certificate and key for the web server.
- EclecticIQ Platform login credentials.

**Warning:**

The EclecticIQ Platform VM image does not include Neo4j.
You need to install Neo4j manually, as described in [Install Neo4j](#).

Bundled third party software

EclecticIQ Platform is bundled with the following third party software.

Each product on the list abides by its own terms and conditions and its own license.

| Dependency | Version | Reference |
|----------------|-----------|--------------|
| alembic | 0.9.6 | Bitbucket |
| amqp | 1.4.9 | GitHub |
| anyjson | 0.3.3 | Bitbucket |
| apispec | 0.4.1 | GitHub |
| appnope | 0.1.0 | GitHub |
| argswargs | 1.0.3 | GitHub |
| asn1crypto | 0.23.0 | GitHub |
| attrs | 17.3.0 | attrs |
| bcrypt | 3.1.4 | GitHub |
| beautifulsoup4 | 4.6.0 | Crummy |
| billiard | 3.3.0.23 | GitHub |
| blinker | 1.4 | PythonHosted |
| boto3 | 1.4.7 | GitHub |
| botocore | 1.7.45 | GitHub |
| cabby | 0.1.18 | GitHub |
| cachetools | 2.0.1 | GitHub |
| celery | 3.1.18 | Celery |
| certifi | 2017.11.5 | Certifi |
| cffi | 1.11.2 | CFFI |
| chardet | 3.0.4 | GitHub |
| click | 6.7 | GitHub |
| colorama | 0.3.9 | GitHub |

| Dependency | Version | Reference |
|-------------------|---------|---------------|
| colorlog | 3.1.0 | GitHub |
| cryptography | 2.1.3 | GitHub |
| datauri | 1.0.0 | GitHub |
| decorator | 4.1.2 | GitHub |
| defusedxml | 0.5.0 | GitHub |
| docutils | 0.14 | Docutils |
| elasticsearch | 1.7.0 | GitHub |
| fancycompleter | 0.8 | Bitbucket |
| feedparser | 5.2.1 | GitHub |
| flamegraph | 0.1 | GitHub |
| Flask | 0.10.1 | GitHub |
| Flask-Classy | 0.6.10 | GitHub |
| Flask-JWT | 0.2.0 | GitHub |
| flask-marshmallow | 0.8.0 | GitHub |
| Flask-Redis | 0.3.0 | GitHub |
| Flask-SQLAlchemy | 2.3.2 | GitHub |
| furl | 1.0.1 | GitHub |
| future | 0.16.0 | Python-Future |
| gunicorn | 19.7.1 | Gunicorn |
| idna | 2.6 | GitHub |
| inflect | 0.2.5 | pyPI |
| ipdb | 0.10.3 | GitHub |
| ipython | 6.2.1 | IPython |
| ipython_genutils | 0.2.0 | IPython |
| itsdangerous | 0.24 | GitHub |
| jedi | 0.11.0 | GitHub |
| Jinja2 | 2.10 | Jinja |
| jmespath | 0.9.3 | GitHub |
| kombu | 3.0.37 | Kombu |
| libtaxii | 1.1.111 | TAXII |

| Dependency | Version | Reference |
|------------------------|-----------|--------------|
| lxml | 4.1.1 | lxml |
| Mako | 1.0.7 | mako |
| MarkupSafe | 1.0 | GitHub |
| marshmallow | 2.6.1 | GitHub |
| marshmallow-sqlalchemy | 0.13.2 | GitHub |
| newrelic | 2.96.0.80 | New Relic |
| orderedmultidict | 0.7.11 | GitHub |
| paramiko | 2.4.0 | GitHub |
| parso | 0.1.0 | GitHub |
| pdbpp | 0.9.2 | GitHub |
| pexpect | 4.3.0 | Pexpect |
| pickleshare | 0.7.4 | GitHub |
| prompt_toolkit | 1.0.15 | GitHub |
| psycopg2 | 2.7.3.2 | init.d |
| ptyprocess | 0.5.2 | GitHub |
| pyasn1 | 0.3.7 | GitHub |
| pycparser | 2.18 | GitHub |
| Pygments | 2.2.0 | Pygments |
| PyJWT | 1.4.0 | GitHub |
| pyldap | 2.4.25.1 | GitHub |
| pymisp | 2.4.80 | GitHub |
| PyNaCl | 1.2.0 | GitHub |
| pyOpenSSL | 17.3.0 | pyOpenSSL |
| pysaml2 | 4.5.0 | GitHub |
| python-dateutil | 2.4.2 | dateutil |
| python-editor | 1.0.3 | GitHub |
| python-gnupg | 0.4.1 | PythonHosted |
| python-magic | 0.4.13 | GitHub |
| python-slugify | 1.2.4 | GitHub |

| Dependency | Version | Reference |
|------------------|---------|----------------------|
| pytz | 2017.3 | PythonHosted |
| PyYAML | 3.12 | PyYAML |
| rarfile | 3.0.0 | GitHub |
| redis | 2.10.6 | GitHub |
| requests | 2.18.4 | Requests |
| requests-futures | 0.9.7 | GitHub |
| retrying | 1.3.3 | GitHub |
| rfc3986 | 1.1.0 | RFC 3986 |
| s3transfer | 0.1.11 | GitHub |
| simplegeneric | 0.8.1 | CheeseShop |
| six | 1.11.0 | PyPI |
| SQLAlchemy | 1.1.15 | SQLAlchemy |
| statsd | 3.2.1 | GitHub |
| structlog | 16.0.0 | structlog |
| tld | 0.7.9 | GitHub |
| traitlets | 4.3.2 | IPython |
| unrar | 0.3 | GitHub |
| urllib3 | 1.22 | urllib3 |
| wcwidth | 0.1.7 | GitHub |
| Werkzeug | 0.12.2 | The Pallets Projects |
| wmctrl | 0.3 | Bitbucket |

VM OS login credentials

| | |
|--|------------|
| SSH default login credentials for the VM OS | |
| user name | packer |
| password | Packer123! |

EclecticIQ Platform login credentials

| | |
|--|-----------------|
| EclecticIQ Platform default login credentials | |
| user name | admin |
| password | EclecticIQ2015# |

©2018 by EclecticIQ BV. All rights reserved.
Last generated on Mar 6, 2018

Install the platform

Download a VM image, import it into your VM client, install Neo4j, start working on cyber threat analysis.

EclecticIQ Platform is available as a virtual appliance in a downloadable VM image.

- Contact us to obtain a download link to a VM image. We offer VM images for **VMWare Player** (<https://www.vmware.com/products/player>). **VirtualBox** (<https://www.virtualbox.org/>) is currently not supported.
- Save the downloaded archive on your local machine and decompress its content.

Install a VMWare Player VM image

- Launch VMWare Workstation Player.
- The default VMWare Workstation Player start location is **Home**.
- On the right **Home** pane, select **Open a Virtual Machine**.
- On the **Open Virtual Machine** dialog, browse to the location where you saved the decompressed VM files.
- Select the appropriate .ova image file, and then click **Open**.
- The new VM becomes available in the left-hand pane in the player.
- On the right-hand pane, click **Edit virtual machine settings**.
- On the **Virtual Machine Settings** dialog, **Hardware** tab, click the **Add** button.
- On the **Add Hardware Wizard**, **Hardware Type** dialog, select **Network Adapter**, and then click **Next**.
- On the **Add Hardware Wizard**, **Network Adapter Type** dialog, select the **Host-only: A private network shared with the host** option, and then click **Finish**.
- Click **OK**.
- On the right-hand pane, click **Power On**.

Install Neo4j

Install Neo4j before starting working with EclecticIQ Platform.

Neo4j is not included in the EclecticIQ Platform VM image. To install Neo4j manually, do the following:

- Log into the platform OS.
- To successfully execute several commands in the command line or in the terminal, you may need root-level access rights.

To obtain admin rights, run the following command(s):

```
$ sudo su -
```

Alternatively:

- Grant admin rights to a specific user, who can then log in with their password to perform admin tasks:

```
$ su - ${username}
```

Or:

- Prefix `sudo` to the command you want to run:

```
$ sudo ${command}
```

- Go to the root directory:

```
cd /
```

Pre-installation checks and tasks

- Check if Neo4j is installed:

```
$ yum info neo4j
```

- Verify that no Neo4j-related services or processes are running.
If Neo4j is not installed, `neo4j-batching` should either not be included in the returned list of services, or it should not be running if it is included:

```
$ supervisorctl status
```

Create a YUM repository for the dependencies

You can download the core dependencies for the platform from a centralized YUM repository. This repository makes it easy for system administrators to manage platform installation, upgrade procedures, and ensures that the platform dependency versions match the requirements.

Create the YUM repository configuration file

- Create a file with the repository information:

```
# Create and populate the YUM repo file for the platform dependencies
echo "
[platform-centos-deps-2.1]
name=platform-centos-deps-2.1
baseurl=https://downloads.eclecticiq.com/platform-centos-deps/2.1/
gpgcheck=0
repo_gpgcheck=0
enabled=1
username=${eiq_user}
password=${eiq_pass}
gpgkey=https://downloads.eclecticiq.com/public/GPG-KEY-eclecticiq
      https://downloads.eclecticiq.com/public/GPG-KEY-elasticsearch
      https://downloads.eclecticiq.com/public/neo4j.gpg.key
      https://downloads.eclecticiq.com/public/nginx_signing.key
      https://downloads.eclecticiq.com/public/IUS-COMMUNITY-GPG-KEY
      https://downloads.eclecticiq.com/public/RPM-GPG-KEY-EPEL-7
      https://downloads.eclecticiq.com/public/NODESOURCE-GPG-SIGNING-KEY-EL.key
priority=1
" > /etc/yum.repos.d/platform-centos-deps.repo
```

Update and set user name and password as needed, based on your system environment.

- Run the following command to upgrade the YUM repository list, so that it includes only the *platform-centos-deps-2.1* YUM repo:

```
# Upgrade YUM repository
yum -y upgrade --disablerepo=* --enablerepo="platform-centos-deps-2.1,epel,base"
```

Install Neo4j

- Install Neo4j:

```
yum -y --disablerepo=* --enablerepo="platform-centos-deps-2.1,epel,base" install neo4j-3.3.0*
```

- Pin Neo4j, so that it stays on the current version and it is not automatically updated:


```
# Pin Neo4j to installed version
yum versionlock neo4j
```

Set up Neo4j

Set the Neo4j environment variables

- Set up a profile file for Neo4j where you define the following environment variables:

```
$ echo -e 'export NEO4J_HOME=/usr/share/neo4j\nexport PATH=$PATH:$NEO4J_HOME/bin' > /etc/profile.d/neo4j.sh
```

- Reload the Neo4j profile to update it, so that it can pick up the newly defined environment variables:

```
$ source /etc/profile.d/neo4j.sh
```

- Make sure these environment variables are accessible to the `neo4j` user, `neo4j` group profile.

```
$ su -s /bin/bash neo4j -c 'echo $NEO4J_HOME; echo $PATH'

# Expected output
/usr/share/neo4j
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/usr/share/neo4j
```

Check that systemd monitors Neo4j

- Verify that the `/opt/eclecticiq/etc/eclecticiq/platform_settings.py` platform settings configuration file includes `neo4j` in the list of services `systemd` monitors:

```
cat /opt/eclecticiq/etc/eclecticiq/platform_settings.py

# The neo4j service should be included in this section
"SYSTEMD_SERVICES = [
    'elasticsearch',
    'kibana',
    'logstash',
    'neo4j',
    'postfix',
    'postgresql-10',
    'redis',
    'statsd',
]"
```

Configure Neo4j

- Copy the Neo4j configuration file shipped with the platform:

```
# Copy Neo4j config
cp /opt/eclecticiq/etc-extras/neo4j/neo4j.conf /etc/neo4j/
```

- Set the Neo4j graph database location to your data directory:

```
# Set Neo4j database location
mkdir -p ${data_dir}/neo4j/
chown -R neo4j:neo4j ${data_dir}/neo4j
sed -ri 's|^\(s*dbms.directories.data\s*=\) .*|\1'${data_dir}/neo4j'|g' /etc/neo4j/neo4j.conf
```

Disable Neo4j remote shell

- For security reasons, disable remote shell for Neo4j:

```
# Disable Neo4j remote shell
sed -i '/^\s*dbms.shell.enabled\s*=/d' /etc/neo4j/neo4j.conf
echo "dbms.shell.enabled=false" >> /etc/neo4j/neo4j.conf
```

Enable the service

- Enable the Neo4j service to automatically start at system boot:

```
# Enable Neo4j service
systemctl enable neo4j
```

- Restart Neo4j:

```
# Restart Neo4j
systemctl restart neo4j
```

- Check if Neo4j is running:

```
# Check if Neo4j is running
systemctl status neo4j
```

Configure Neo4j authentication

- Set a Neo4j password for the `neo4j:neo4j` user and group:

```
# Set Neo4j password
curl -H "Content-Type: application/json" -X POST -d '{"password":"'${neo4j_pass}'"' -u neo4j:neo4j http://localhost:7474/user/neo4j/password
```

- Update the platform configuration:

```
# Configure Neo4j credentials in platform settings
sed -ri 's/^\s*(NEO4J_USER\s*=\s*) .*/\1\'neo4j\'/g' /opt/eclecticiq/etc/eclecticiq/platform_settings.py
sed -ri 's/^\s*(NEO4J_PASS\s*=\s*) .*/\1\'${neo4j_pass}\'/g' /opt/eclecticiq/etc/eclecticiq/platform_settings.py
```

Create the Neo4j graph schema

- Before creating the graph schema, stop the *graph-ingestion* and any running *intel-ingestion* tasks:

```
# Stop graph-ingestion and intel-ingestion tasks
supervisorctl stop graph-ingestion intel-ingestion:*
```

- Then, run the following command to create the graph schema and to apply the necessary graph database migrations to the Neo4j database:

```
# Create the graph schema / Apply graph database migrations
EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
/opt/eclecticiq/platform/api/bin/eiq-platform graph upgrade
```

Reload Supervisor

At this point Neo4j is installed, and it should use the correct configuration files to work with the platform. When you edit or update Supervisor configurations, run `systemctl restart supervisor` and `supervisorctl reload`, so that Supervisor can pick up and reload any updated configurations to the platform with the latest changes.

- Restart Supervisor to load the changed configuration for Neo4j:

```
$ supervisorctl reload
```

- Verify that `neo4j-batching` is up and running:

```
$ supervisorctl status
```

- The response should include the service, and it should flag it as running:

```
neo4j-batching      RUNNING      pid 30726,      uptime 0:05:23
```

Test Neo4j in the UI

As a final check, sign into the web-based UI to make sure that the graph is up and running, and that it is working correctly. you can add entities to graph.

- Check the system health.
The status of the following processes should be green:
 - *graph-ingestion*
 - *intel-ingestion*
 - *neo4j*
- Add some entities to the graph to make sure that the graph loads them and displays them correctly.
- Create one or two new test entities using the entity editor, and then load them on the graph to make sure it works as expected.

Launch the platform

Start the VM to make the platform available, and then access the platform through a web browser.

Start the VM



To access the VM, you may need to enter valid login credentials. If you do not have these details, contact us.

- Launch VirtualBox or VMWare Player, and then start/play the VM.
- If you are prompted for login credentials at startup, enter the provided user name and password.
- The VM should be up and running.

Get the VM IP address

By default, our VM images run **CentOS Linux 7 (1511)** (<https://lists.centos.org/pipermail/centos-announce/2015-december/021518.html>).

- Run the following command(s):

```
ifconfig
```

- Press **ENTER**.
- Look for the following entry to identify the VM IP address:

```
inet <IP_address>
```

The `inet` IP address is the one you need to use to access the platform.

Example: `inet 10.0.2.148`.

Go to <https://10.0.2.148> to sign in to the platform web-based UI.

Go to the platform

- In your host machine, launch a web browser (recommended: Google Chrome).
- In the address bar, enter the VM IP address.
Example: <https://10.0.2.148>
- The platform login screen is displayed.

- Sign in with the appropriate credentials.



Warning: The browser may display an untrusted connection warning: add it as an exception, and then proceed to the platform.

Upgrade the platform

When a new platform release is available, you can upgrade your existing installation to benefit from the latest features and enhancements.

When you download a new VM image containing a newer platform release than the currently installed one on your system, you can upgrade your platform installation to the latest available public version.

The upgrade procedure requires some housekeeping; once you are done, you can access new features, and you can enjoy the improvements we introduce in the product on a regular basis.

To successfully execute several commands in the command line or in the terminal, you may need root-level access rights.

To obtain admin rights, run the following command(s):

```
$ sudo su -
```

Alternatively:

- Grant admin rights to a specific user, who can then log in with their password to perform admin tasks:

```
$ su - ${username}
```

Or:

- Prefix `sudo` to the command you want to run:

```
$ sudo ${command}
```

The upgrade procedure consists of the following steps:

Before the upgrade

- Disable rules
- Exit the platform
- Back up your data
- Check the prerequisites

During the upgrade

- Install the new package

After the upgrade

- Check the configuration
- Check third-party configurations
- Reload Supervisor configurations

Before the upgrade

Disable rules

Disable all platform rules: entity, observable, enrichment, and discovery rules.


You can disable rules in one of the following ways:

In the rule detail pane

- Click **Data configuration > Rules > Observable**, or **Data configuration > Rules > Entity**, or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the rule overview click anywhere on the row corresponding to the rule you want to disable.
- On the rule detail panel:
 - Click **Actions > Disable** to disable the rule.Alternatively:
 - On the **Details** tab click **Disable**.


A notification message is displayed to confirm the change.

In the rule overview

- Click **Data configuration > Rules > Observable**, or **Data configuration > Rules > Entity**, or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the rule overview click the  menu on the row corresponding to the rule you want to disable.
- From the drop-down menu select **Disable** to disable the rule.

A notification message is displayed to confirm the change.

Bulk disable

- Click **Data configuration > Rules > Observable**, or **Data configuration > Rules > Entity**, or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the top-left corner click the quick filter icon () to display the available rule quick filters.
- Click **Show**, select **Enabled**, and then click **OK** to display only enabled rules.
- To select all the rules on the view click the checkbox on the top-left corner of the table.
- To disable all the selected rules in bulk, on the quick filter horizontal bar click **Actions > Disable**.

A notification message is displayed to confirm the change.

Exit the platform

Sign out of the platform:

- Click the active user profile image on the left-hand navigation sidebar, bottom-left corner of the screen.
- From the drop-down menu select **Sign out**.
- You are signed out.

Back up your data

Shut down the platform

To gracefully shut down EclecticlQ Platform, stop all platform-related services and processes.

- A normal/standard platform shutdown does not involve any specific procedure or step sequence.
- If you shut down the platform before performing a platform upgrade, follow the steps described under **Shutdown before a platform upgrade**.
In this case, it is important that you stop platform components, services, and processes gradually to avoid hanging queues, tasks or PIDs.

Normal shutdown

You can stop the platform without following any specific procedure. However, if you want to make sure the platform core services and processes gracefully shut down, manually execute the following commands:

```
$ supervisorctl stop all
```

```
$ systemctl stop postgresql-10 redis neo4j kibana logstash elasticsearch postfix
```

Shutdown before a platform upgrade



If you are shutting down the platform before performing an *upgrade* or a *database backup*, stop platform components in the order described below to make sure no Celery tasks are left over in the queue, and no read/write activity is in progress on Redis.

This prevents hanging tasks in the queue from interfering with the upgrade or backup procedures.

- Stop *platform-api*:

```
$ supervisorctl stop platform-api
```

- Stop the Celery beat:

```
$ supervisorctl stop task:beat
```

- Check Celery queues; they should be empty:

```
# Launch redis-cli
$ redis-cli

$ > llen enrichers

$ > llen integrations

$ > llen priority_enrichers

$ > llen priority_providers

$ > llen priority_utilities

$ > llen providers

$ > llen reindexing

$ > llen utilities
```

- To delete a non-empty Celery queue, run the following command(s):

```
# Launch redis-cli
$ redis-cli

# Delete the entity ingestion queue
$ > del "queue:ingestion:inbound"

# Delete the graph ingestion queue
$ > del "queue:graph:inbound"

# Delete the search indexing queue
$ > del "queue:search:inbound"
```

- Stop the remaining Celery workers:

```
$ supervisorctl stop task:*
```

- Stop Supervisor-managed workers:

- *intel-ingestion*
- *search-ingestion*
- *graph-ingestion*
- *opentaxii*
- *neo4j-batching*

```
$ supervisorctl stop all
```

Check that there are no leftover PID files:

- First, make sure that no platform-related PID is running:

```
$ ps aux | grep beat
```

- If any platform-related PIDs are running, terminate them with the `kill` command.
- Manually remove any leftover PID files with the `rm` command. Usually, PID files are stored in `/var/run`.

Check the prerequisites



When upgrading dependencies and third-party components, refer to their official documentation for detailed instructions on installation and upgrade procedures, and look up their official release notes for any product changes that may impact your environment.

Check the configuration

After installing the platform, browse to `/opt/eclecticiq/etc/eclecticiq/`. Configuration files are stored here. You can find both the new/latest configuration files, as well as the ones belonging to the previous version of the platform you upgraded from.

| Core platform configuration files | |
|-----------------------------------|--|
| <code>platform_settings.py</code> | Contains core platform settings like security key value, authentication bearer token expiration time, URLs pointing to external components Celery-managed tasks, and LDAP configuration. |
| <code>opentaxii.yml</code> | Contains OpenTAXII (https://opentaxii.readthedocs.io/) configuration parameters like URL and port for the service, as well as the designated inbound queue and message broker to use. |

Verify that the platform configuration files reflect the new, upgraded environment.

You may need to carry out this task manually. In this case, you can diff the files with a tool like **Meld** (<http://meldmerge.org/>).

Check third-party configurations

After checking the platform configuration to make sure it correctly describes the upgraded environment, do the same with the configurations of third-party components and dependencies.

You may need to carry out this task manually. In this case, you can diff the files with a tool like **Meld** (<http://meldmerge.org/>).

After the upgrade

About proxy settings

If the platform is configured to be behind a proxy, make sure the platform proxy configuration allows bypassing local hosts *localhost* and *127.0.0.1*.

If you cannot access the platform proxy settings, and if terminal commands acting on platform resources fail to execute correctly, prepend `NO_PROXY=127.0.0.1` to the commands to bypass the proxy server on the fly.

This behavior may affect the following commands:

```
# Start Python shell
$ /opt/eclecticiq/platform/api/bin/eiq-platform shell

# Request current db version
$ /opt/eclecticiq/platform/api/bin/eiq-platform database current-version

# Migrate Elasticsearch indices
$ /opt/eclecticiq/platform/api/bin/eiq-platform search upgrade

# Validate Elasticsearch index migration
$ /opt/eclecticiq/platform/api/bin/eiq-platform search validate

# Reindex Neo4j graph db
$ /opt/eclecticiq/platform/api/bin/eiq-platform graph reindex

# Migrate Neo4j graph db
$ /opt/eclecticiq/platform/api/bin/eiq-platform graph upgrade

# Load fixtures
$ /opt/eclecticiq/platform/api/bin/eiq-platform database load-fixtures

# Registered and enabled extension state validation
$ /opt/eclecticiq/platform/api/bin/eiq-platform extensions validate
```

Example:

```
$ NO_PROXY=127.0.0.1 /opt/eclecticiq/platform/api/bin/eiq-platform graph upgrade
```

About Elasticsearch indexes

If you need to prioritize migrating Elasticsearch indexes, process at least the following ones:

- *stix*: indexes entities
- *extracts*: indexes observables

| Elasticsearch index name | Description |
|--------------------------|---|
| <i>audit</i> | Records audit trail events related to entities, datasets, enrichers, incoming and outgoing feeds, rules and tasks. |
| <i>documents</i> | Records log information related to ingestion, tasks, and task scheduling. |
| <i>draft-entities</i> -* | Indexes draft entity data, that is, entities that are currently saved as drafts, and that have not yet been published to the platform. These entities are not searchable in the platform. |
| <i>extracts</i> -* | Indexes all observable data. |
| <i>logstash</i> -* | Indexes log aggregation and logging information such as host, http request types, http response status codes, platform component, and path to the log where log entries are saved to. |
| <i>statsd</i> -* | Collects metrics about received packets and detected invalid or not well-formed lines in the ingested packets. |
| <i>stix</i> * | Indexes published entity data, that is, entities that are published to the platform. These entities are searchable in the platform. |

Run a final check

As a last step before launching the platform, it is good practice to check the following points:

- Core processes and services
- Search, indexing and graph
- Availability
- To check if a core service is enabled to start at system bootup run the following command(s):

```
systemctl is-enabled ${service_name}
```

- To check if a core service is running run the following command(s):

```
systemctl status ${service_name}
```

- To start a core service run the following command(s):

```
systemctl start ${service_name}
```

Check core processes and services

Nginx

- Verify that Nginx is up and running by checking the web server status:

```
$ systemctl status nginx
```

Supervisor

To check if the *supervisord* daemon is running, run the following command(s):

```
systemctl status supervisord
```

PostgreSQL

To check if PostgreSQL is running, run the following command(s):

```
systemctl status postgresql-10
```

or:

```
systemctl list-units | grep -i postgre
```

Check search indexing and graph

Elasticsearch

To check if Elasticsearch is running, run the following command(s):

```
systemctl status elasticsearch
```

Neo4j

To check if Neo4j is running, run the following command(s):

```
$ systemctl status neo4j
```

Check search indexing and graph availability

Make sure Elasticsearch and Neo4j are available by making cURL calls to the corresponding endpoints:

```
# Check Elasticsearch availability
$ curl localhost:9200

# Check Neo4j availability
# HTTP port: 7474; HTTPS port: 7473
$ curl localhost:7474
```

Re-enable and run the rules

Before starting Supervisor-managed ingestion processes, enable again the rules you previously disabled.

Run the re-enabled rules after completing the data migration, so that they can filter out any observables marked to be ignored.

Enable platform rules: entity, observable, enrichment, and discovery rules.

You can enable rules in one of the following ways:

In the rule detail pane


- Click **Data configuration > Rules > Observable** , or **Data configuration > Rules > Entity** , or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the rule overview click anywhere on the row corresponding to the rule you want to enable.
- On the rule detail panel:
 - Click **Actions > Enable** to enable the rule.

Alternatively:

- On the **Details** tab click **Enable**.


A notification message is displayed to confirm the change.

In the rule overview

- Click **Data configuration > Rules > Observable** , or **Data configuration > Rules > Entity** , or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the rule overview click the  menu on the row corresponding to the rule you want to enable.
- From the drop-down menu select **Enable** to enable the rule.

A notification message is displayed to confirm the change.

Bulk enable

- Click **Data configuration > Rules > Observable** , or **Data configuration > Rules > Entity** , or **Data configuration > Rules > Enrichment**, or **Data configuration > Rules > Discovery** to display the observable, entity, enrichment, or discovery rule overview.
- On the top-left corner click the quick filter icon () to display the available rule quick filters.
- Click **Show**, select **Disabled**, and then click **OK** to display only disabled rules.
- To select all the rules on the view click the checkbox on the top-left corner of the table.
- To enable all the selected rules in bulk, on the quick filter horizontal bar click **Actions > Enable**.

A notification message is displayed to confirm the change.

Reload Supervisor configurations

To reload the Supervisor configuration and to restart all Supervisor-managed processes run the following command(s):

```
$ supervisorctl reload
```

To check the statuses of the tasks managed by Supervisor, run the following command(s):

```
$ supervisorctl status
```

The response should return `RUNNING` for all relevant tasks to confirm that all Supervisor tasks are being executed normally.

The following example serves as a guideline:

```
graph-ingestion          RUNNING    pid 19527, uptime 0:00:03
intel-ingestion:0        RUNNING    pid 19071, uptime 0:00:51
intel-ingestion:1        RUNNING    pid 19070, uptime 0:00:51
intel-ingestion:2        RUNNING    pid 19073, uptime 0:00:51
intel-ingestion:3        RUNNING    pid 19072, uptime 0:00:51
neo4j-batching           RUNNING    pid 19268, uptime 0:00:43
opentaxii                RUNNING    pid 19330, uptime 0:00:36
platform-api             RUNNING    pid 19077, uptime 0:00:51
search-ingestion         RUNNING    pid 19075, uptime 0:00:51
task:beat                RUNNING    pid 19061, uptime 0:00:51
task:discovery           RUNNING    pid 19068, uptime 0:00:51
task:discovery-priority  RUNNING    pid 19065, uptime 0:00:51
task:enrichers           RUNNING    pid 19056, uptime 0:00:51
task:enrichers-priority  RUNNING    pid 19062, uptime 0:00:51
task:entity-rules-priority RUNNING    pid 19063, uptime 0:00:51
task:extract-rules-priority RUNNING    pid 19055, uptime 0:00:51
task:incoming-transports RUNNING    pid 19053, uptime 0:00:51
task:incoming-transports-priority RUNNING    pid 19054, uptime 0:00:51
task:outgoing-feeds      RUNNING    pid 19066, uptime 0:00:51
task:outgoing-feeds-priority RUNNING    pid 19057, uptime 0:00:51
task:outgoing-transports RUNNING    pid 19060, uptime 0:00:51
task:outgoing-transports-priority RUNNING    pid 19058, uptime 0:00:51
task:reindexing          RUNNING    pid 19064, uptime 0:00:51
task:utilities           RUNNING    pid 19059, uptime 0:00:51
task:utilities-priority  RUNNING    pid 19067, uptime 0:00:51
```

Install extensions

After successfully completing the platform upgrade, you can proceed to install extensions as necessary to expand platform functionality, and to add support for a broad range of transport types and content types for incoming and outgoing feeds, as well as many enrichers.