# Set up EclecticIQ Platform as a virtual appliance

## Virtual appliance configuration guide for system administrators

Last generated: January 12, 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
Software requirements 7
Bundled third party software 7
VM OS login credentials 11
EclecticIQ Platform login credentials 11
Install the platform 12
Install a VMWare Player VM image 12
Install Neo4j 13
Pre-installation checks and tasks 13
Create a YUM repository for the dependencies 14
Create the YUM repository configuration file 14
Install Neo4j 15
Set the Neo4j environment variables 15
Configure Neo4j 15
Set Neo4j JVM memory limits 16
Configure page cache size and remote shell 16
Check permissions 17
Configure HTTPS and authentication in Neo4j 17
Enable secure access 17
Enable the service 19
Create the Neo4j graph schema 19
Reload Supervisor 19
Test Neo4j in the UI 20
Launch the platform 21
Start the VM 21
Get the VM IP address 21
Go to the platform 21
Upgrade the platform 23
Before the upgrade 24
Disable rules 24
Exit the platform 24
Back up your data 25
Shut down the platform 25
Normal shutdown 25
Shutdown before a platform upgrade 25
Check the prerequisites 27
Check the configuration 27
Check third-party configurations 27
After the upgrade 28
About proxy settings 28
About Elasticsearch indexes 28
Run a final check 29
Check core processes and services 29
Check search indexing and graph 30
Check search indexing and graph availability 30
Re-enable and run the rules 31

Reload Supervisor configurations 31
Install extensions 32

# 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 |

# 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.
  - Recommended: Google Chrome (latest)
  - Supported: Mozilla Firefox, Microsoft Edge and Internet Explorer (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](../_vm-install/vm_neo4j_install.html).

## Bundled third party software

EclecticIQ Platform is bundled with the following third party software: Note: 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 |

| Dependency | Version | Reference |
|---|---|---|
| apispec | 0.4.1 | GitHub |
| appnope | 0.1.0 | GitHub |
| argskwargs | 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 |
| 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 |

| Dependency | Version | Reference |
|---|---|---|
| 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 |
| 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 |

| Dependency | Version | Reference |
|---|---|---|
| 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 |
| 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 |

| Dependency | Version | Reference |
|---|---|---|
| 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#` |

# 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
```

- Verify that the `neo4j` user and the `neo4j` group own the `/media/neo4j/` directory, and therefore the `graph.db` graph database file it stores:

```
$ ls -l /media/neo4j
```

## 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.

This YUM repository provides the following platform dependencies:

- Nginx
- Redis
- PostgreSQL
- Node.js, including npm
- Elasticsearch, including plugins
- Logstash
- Kibana
- Neo4j
- StatsD
- Supervisor
- Postfix

## Create the YUM repository configuration file

- Create a new file with the repository information:

```
# Create and populate yum repo file for Platform dependencies
echo "
[eclecticiq-platform-deps-2.0]
name=eclecticiq-platform-deps-2.0
baseurl=https://downloads.eclecticiq.com/platform-rpm-deps/2.0/
gpgcheck=1
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/eclecticiq-platform-deps.repo
```

Make sure you update and set user name and password as needed.

- Run the following command to upgrade the YUM repository list, so that it includes only the eclecticiq-platform-deps-2.0 YUM repo:

```
# Upgrade YUM repository
yum -y upgrade --disablerepo=* --enablerepo="eclecticiq-platform-deps-2.0,epel,base"
```

# Install Neo4j

- Install the following core dependencies:

```
yum -y --disablerepo=* --enablerepo="eclecticiq-platform-deps-2.0,epel,base" install neo4j-2.3.8
```

## Set the Neo4j environment variables

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

```
# Export profile for Neo4j
echo 'export NEO4J_HOME=/usr/share/neo4j
export PATH=$PATH:$NEO4J_HOME/bin' > /etc/profile.d/neo4j.sh
```

- Make sure these environment variables are accessible to the `neo4j` user, `neo4j` group profile by checking the output of `su -s /bin/bash neo4j -c 'echo $NEO4J_HOME; echo $PATH`which should look like this:

```
/usr/share/neo4j
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/usr/share/neo4j
```

# Configure Neo4j

- Neo4j uses the following configuration files:
  - `/etc/neo4j/neo4j-server.properties`
  - `/etc/neo4j/neo4j-wrapper.conf`
  - `/etc/neo4j/neo4j.properties`
- Set the database location to your data directory:

```
# Set Neo4j database location
mkdir -p ${data_dir}/neo4j/
chown -R neo4j:neo4j ${data_dir}/neo4j
sed -i '/^\s*org.neo4j.server.database.location\s*=/d' /etc/neo4j/neo4j-server.properties
echo "org.neo4j.server.database.location=${data_dir}/neo4j/graph.db" >> /etc/neo4j/neo4j-
server.properties
```

- Disable security authentication:

```
# Disable Neo4j security auth
sed -i '/^\s*dbms.security.auth_enabled\s*=/d' /etc/neo4j/neo4j-server.properties
echo "dbms.security.auth_enabled=false" >> /etc/neo4j/neo4j-server.properties
```

- Disable HTTPS:

```
# Disable Neo4j https
sed -i '/^\s*org.neo4j.server.webserver.https.enabled\s*=/d' /etc/neo4j/neo4j-server.properties
echo "org.neo4j.server.webserver.https.enabled=false" >> /etc/neo4j/neo4j-server.properties
```

- Set the database tuning properties location:

```
# Set Neo4j tuning properties location
sed -i '/^\s*org.neo4j.server.db.tuning.properties\s*=/d' /etc/neo4j/neo4j-server.properties
echo "org.neo4j.server.db.tuning.properties=/etc/neo4j/neo4j.properties" >> /etc/neo4j/neo4j-
server.properties
```

## Set Neo4j JVM memory limits

Set memory values in `/etc/neo4j/neo4j-wrapper.conf`:

- The recommended value for `wrapper.java.initmemory` is at least 4096 MB or more, if possible.
- The recommended value for `wrapper.java.maxmemory` is at least 4096 MB or more, if possible.
  This value should never be lower than `wrapper.java.initmemory`.

```
# Set Neo4j memory limits
sed -i '/^\s*wrapper.java.initmemory\s*=/d' /etc/neo4j/neo4j-wrapper.conf
echo "wrapper.java.initmemory=4096" >> /etc/neo4j/neo4j-wrapper.conf
sed -i '/^\s*wrapper.java.maxmemory\s*=/d' /etc/neo4j/neo4j-wrapper.conf
echo "wrapper.java.maxmemory=8192" >> /etc/neo4j/neo4j-wrapper.conf
```

## Configure page cache size and remote shell

- Disable remote shell for Neo4j:

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

- Set the size of the page cache to `5G`:

```
# Set Neo4j page cache size
sed -i '/^\s*dbms.pagecache.memory\s*=/d' /etc/neo4j/neo4j.properties
echo "dbms.pagecache.memory=5G" >> /etc/neo4j/neo4j.properties
```

## Check permissions

Make sure the target directory to save application data to exists, and that it is accessible to read and write data.

```
# Create Neo4j data dir and set file permissions
mkdir -p ${data_dir}/neo4j
chown -R neo4j:neo4j /var/log/neo4j ${data_dir}/neo4j/
```

## Configure HTTPS and authentication in Neo4j

If Neo4j is hosted on a server that has network access, it is a good idea to enable secure access and user access control through HTTPS/SSL and an authentication mechanism.

## Enable secure access

To enable access to Neo4j through a secure connection, you need to:

- Set the transport protocol to HTTPS.
- Set a secure port.
- Provide a valid SSL key and certificate pair.
  On first-time start, Neo4j automatically generates a self-signed SSL certificate and a private key. For production environments, replace these files with your own SSL key and certificate.

These tasks require editing the following configuration files:

- */etc/neo4j/neo4j-server.properties*: for detailed instructions on the properties and values you need to edit to enable SSL, refer to the **Neo4j official documentation** `(https://neo4j.com/docs/2.3.8/security-server.html#security-server-https)`.

  By default, neo4j-server.properties port for HTTPS access is `7473`. Set it to a different port.

  The relevant properties that control secure access are:

```
# All values in this example are dummy

# Certificate location (auto-generated if the file does not exist)
dbms.security.tls_certificate_file=/etc/neo4j/ssl/snakeoil.cert

# Private key location (auto-generated if the file does not exist)
dbms.security.tls_key_file=/etc/neo4j/ssl/snakeoil.key

# Enable/Disable HTTPS
org.neo4j.server.webserver.https.enabled=true

# HTTPS port (for all data, administrative, and UI access)
org.neo4j.server.webserver.https.port=7473
```

- */opt/eclecticiq/etc/eclecticiq/platform_settings.py*: add or edit the following parameters, so that the platform is aware that communication with Neo4j flows through HTTP and SSL:

```
# Specify HTTPS and secure port for Neo4j
NEO4J_URL = "http://127.0.0.1:7473"

# Specify SSL cert for Neo4j
NEO4J_SSL_VERIFY = "/etc/neo4j/ssl/snakeoil.cert"
```

The port number you set in `org.neo4j.server.webserver.https.port` should be the same as the port number you assign to `NEO4J_URL`.

The SSL certificate name and path in `dbms.security.tls_certificate_file` should be the same as the value you assign to `NEO4J_SSL_VERIFY`.

To enable user authentication for Neo4j, you need to:

- Enable authentication in Neo4j.
- Set the correct Neo4j login user name and password in the platform settings file.

These tasks require editing the following configuration files:

- */etc/neo4j/neo4j-server.properties*: for detailed descriptions of Neo4j authentication and authorization, refer to the Neo4j official documentation:
  - **Server authentication and authorization** `(https://neo4j.com/docs/2.3.8/security-server.html#security-server-auth)`
  - **REST API Authentication and Authorization** `(https://neo4j.com/docs/2.3.8/rest-api-security.html#rest-api-security-authenticating)`

  In *neo4j-server.properties* set the following property to `true`:

```
# Enable authorization
dbms.security.auth_enabled=true
```

- */opt/eclecticiq/etc/eclecticiq/platform_settings.py*: add or edit the following parameters, so that the platform is aware that it needs to log in to Neo4j with the specified credentials:

```
# User name to log in to Neo4j
NEO4J_USER = "neo4j"

# Password to log in to Neo4j
NEO4J_PASSWORD = "changeme"
```

Change any default values for user name and password as appropriate

If you do not need to enable authentication, remove the `NEO4J_USER` parameter, or set it to `NEO4J_USER = ""`.

### 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
```

## 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/manage do_graph_migrations
```

## 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 graciously shut down EclecticIQ 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 graciously shut down, manually execute the following commands:

```
$ supervisorctl stop all
```

```
$ systemctl stop postgresql-9.5 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 emtpy:

```
# 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** | |
|---|---|
| *platform_settings.py* | Contains core platform settings like security key value, authentication bearer token expiration time, URLs ponting to external components Celery-managed tasks, and LDAP configuration. |
| *opentaxii.yml* | 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 |
|---|---|
| *audit* | Records audit trail events related to entities, datasets, enrichers, incoming and outgoing feeds, rules and tasks. |
| *documents* | Records log information related to ingestion, tasks, and task scheduling. |
| *draft-entities-** | 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. |
| *extracts-** | Indexes all observable data. |
| *logstash-** | 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 entires are saved to. |
| *statsd-** | Collects metrics about received packets and detected invalid or not well-formed lines in the ingested packets. |
| *stix** | 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-9.5
```

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.