Set up EclecticIQ Platform as a virtual appliance

Virtual appliance configuration guide for system administrators

Last generated: October 20, 2017

©2017 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.

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

Table of contents

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

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

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** `(https://www.google.com/chrome/)`.
- DNS name of the host 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).

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#`

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

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**.

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

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 - {user_name}
```

Or:

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

```
$ sudo {command}
```

- Go to the root directory:

```
$ cd /
```

Pre-installation checks

- 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
```

YUM checks if Neo4j exists on the system, and it looks for any available updates.

Install Neo4j

- Install Neo4j 2.3.8 Community:

```
$ yum install -y neo4j-2.3.8
```

Rename the configuration files

- After completing the Neo4j installation, check the original Neo4j configuration files:

```
$ cd /etc/neo4j
$ ls -ltr
```

- The returned list of files should include the following ones:

```
# configuration files installed by Neo4j
-rw-r--r--. 1 root  root  2284 Nov 10  2015 neo4j-wrapper.conf
-rw-r--r--. 1 root  root  3255 Nov 10  2015 neo4j-server.properties
-rw-r--r--. 1 root  root  1796 Nov 10  2015 neo4j.properties

# Platform backup configuration files for Neo4j to work with the platform
lrwxrwxrwx. 1 root  root    49 May 12 10:07 neo4j.properties.rpmsave -> /opt/eclecticiq/etc-
extras/neo4j/neo4j.properties
lrwxrwxrwx. 1 root  root    56 May 12 10:07 neo4j-server.properties.rpmsave ->
/opt/eclecticiq/etc-extras/neo4j/neo4j-server.properties
lrwxrwxrwx. 1 root  root    51 May 12 10:07 neo4j-wrapper.conf.rpmsave -> /opt/eclecticiq/etc-
extras/neo4j/neo4j-wrapper.conf
```

- In the `/etc/neo4j` directory remove the following configuration files installed by Neo4j:

```
$ rm neo4j-wrapper.conf

# Returns the following response. Type 'yes' to confirm deletion.
> rm: remove regular file 'neo4j-wrapper.conf'? yes
```

```
$ rm neo4j-server.properties

# Returns the following response. Type 'yes' to confirm deletion.
> rm: remove regular file 'neo4j-server.properties'? yes
```

```
$ rm neo4j.properties

# Returns the following response. Type 'yes' to confirm deletion.
> rm: remove regular file 'neo4j.properties'? yes
```

- Rename the Neo4j configuration files ending in `.rpmsave` by removing the `.rpmsave` suffix:

```
$ mv neo4j.properties.rpmsave neo4j.properties
$ mv neo4j-server.properties.rpmsave neo4j-server.properties
$ mv neo4j-wrapper.conf.rpmsave neo4j-wrapper.conf
```

- List the content of the `/etc/neo4j` directory to verify that renaming completed successfully:

```
$ ls -l
```

- In the response, the renamed files should still be symlinked to their target files, that is, the Neo4j configuration files provided with the platform:

```
lrwxrwxrwx. 1 root  root    49 May 12 10:07 neo4j.properties ->
/opt/eclecticiq/etc-extras/neo4j/neo4j.properties
lrwxrwxrwx. 1 root  root    56 May 12 10:07 neo4j-server.properties ->
/opt/eclecticiq/etc-extras/neo4j/neo4j-server.properties
lrwxrwxrwx. 1 root  root    51 May 12 10:07 neo4j-wrapper.conf ->
/opt/eclecticiq/etc-extras/neo4j/neo4j-wrapper.conf
```

Check file ownership

- List the content of the `/etc/neo4j` directory to verify that the Neo4j configuration file owners are `root` (user) and `root` (group):

```
$ ls -l /etc/neo4j
```

- In the response, the Neo4j configuration file owners should be `root root`:

```
lrwxrwxrwx. 1 root  root    49 May 12 10:07 neo4j.properties ->
/opt/eclecticiq/etc-extras/neo4j/neo4j.properties
lrwxrwxrwx. 1 root  root    56 May 12 10:07 neo4j-server.properties ->
/opt/eclecticiq/etc-extras/neo4j/neo4j-server.properties
lrwxrwxrwx. 1 root  root    51 May 12 10:07 neo4j-wrapper.conf ->
/opt/eclecticiq/etc-extras/neo4j/neo4j-wrapper.conf
```

- List the content of the `/media/neo4j` directory to verify that the directory owners are `neo4j` (user) and `neo4j` (group):

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

- In the response, the directory owners should be `neo4j neo4j`:

```
drwxr-xr-x. 4 neo4j neo4j  4096 May 12 10:08 graph.db
```

- List the content of the `/usr/share/neo4j/` directory to verify that the `/data` subdirectory owners are `neo4j` (user) and `neo4j` (group):

```
$ ls -l /usr/share/neo4j/
```

- In the response, the `/data` subdirectory owners should be `neo4j neo4j`:

```
drwxr-xr-x. 2 neo4j neo4j  16 May 12 10:03 data
```

Enable Neo4j at bootup

- Check if Neo4j is already configured to automatically start during system bootup:

```
$ systemctl is-enabled neo4j; echo $?

# Response if Neo4j is enabled for autostart at bootup:
> enabled
> 0
```

- If Neo4j is not configured to automatically start during system bootup, run the following command(s):

```
$ systemctl enable neo4j

# Positive response:
> neo4j on
```

- Restart Neo4j:

```
$ systemctl start neo4j
```

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

```
$ systemctl status neo4j
```

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.

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

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.

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017

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 - {user_name}
```

Or:

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

```
$ sudo {command}
```

The upgrade procedure consists of the following steps:

Before the upgrade

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

During the upgrade

- Create the YUM repository configuration
- Run the YUM install

After the upgrade

- Check the configuration
- Check third-party configurations
- Migrate the database
- Check component versions
- Run the fixtures
- Reload Supervisor configurations

- Access the platform

Exit the platform

Sign out of the platform:

- Click the active user profile image on the top-right 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. Usually, PID files are stored under *var/run*.
- Stop systemd-managed services:
  - *postfix*
  - *redis*
  - *statsd*
  - *kibana*
  - *neo4j*
  - *elasticsearch*
  - *postgresql-9.5*

```
$ systemctl stop postfix redis statsd kibana neo4j elasticsearch postgresql-9.5
```

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.

Remove deprecated packages

Before installing the new platform release, you may need to remove any deprecated packages.
When applicable, deprecated package notifications are included in this section, as well as in the product release notes for the release they refer to.
To remove deprecated packages, run the following command(s):

```
$ yum remove <package_name>
```

or:

```
$ rpm -e <package_name>
```

In case uninstalling a deprecated package fails because of dependency-related errors, include the `--nodeps` **option** (http://www.rpm.org/max-rpm-snapshot/s1-rpm-erase-additional-options.html#s2-rpm-erase-nodeps-option):

```
$ rpm -e --nodeps <package_name>
```

Create the YUM repository configuration

If you are upgrading or performing a fresh install of EclecticIQ Platform using the **YUM** (http://yum.baseurl.org/) package manager, you need to define your YUM repository configuration for the platform:

- Start the host system and log into it with the appropriate credentials.
- Create a new file, and name it */etc/yum.repos.d/eclecticiq-platform.repo*
- Populate the file with the following content:

```
[eclecticiq-platform]
name=eclecticiq-platform
baseurl=https://downloads.eclecticiq.com/platform
gpgcheck=1
repo_gpgcheck=1
enabled=1
username=<username>
password=<password>
gpgkey=https://downloads.eclecticiq.com/public/GPG-KEY-eclecticiq
```

Run the YUM install

To **upgrade** the EclecticIQ Platform packages, do the following:

- Start the VM and log in to the VM OS with the appropriate credentials
- Run the following command(s):

```
# Displays a list of installed platform packages
$ rpm -qa eclectic*

# Upgrade to the latest version with the YUM package manager
$ yum upgrade eclecticiq*

# Or:
# Upgrade to a specific version number with the YUM package manager
$ yum upgrade eclecticiq-platform-1.15.0
```

- The upgrade process begins.
- When it completes, you may wish to check if the newly installed version is the latest. To do so, Run the following command(s):

```
# Displays a list of installed platform packages
$ rpm -qa eclectic*
```

The package versions in the result list should have a higher version number, corresponding to the latest available public release.

Before starting the installation, a script carries out a version check to detect any previously installed platform versions.

It is possible to upgrade the platform only sequentially, it is not possible to skip versions.
For example, if you want to upgrade from release 1.10.0 to 1.14.1, you need to upgrade step by step by installing all intermediate releases until 1.14.1 included.

Example:

```
# Upgrade from 1.10.0 to 1.10.1
$ yum install eclecticiq-platform-1.10.1

# Upgrade from 1.10.1 to 1.11.0
$ yum install eclecticiq-platform-1.11.0

# Upgrade from 1.11.0 to 1.12.0
$ yum install eclecticiq-platform-1.12.0

# Upgrade from 1.12.0 to 1.13.0
$ yum install eclecticiq-platform-1.13.0

# Upgrade from 1.13.0 to 1.14.0
$ yum install eclecticiq-platform-1.14.0

# Upgrade from 1.14.0 to 1.14.1
$ yum install eclecticiq-platform-1.14.1
```

If the currently installed version details are the same as the corresponding information in the upgrade version, or if the former does not immediately precedes the latter, the upgrade procedure is aborted, and an error message is displayed:

```
You currently have version ${INSTALLED_VERSION}, but you need to have ${PREV_VERSION} installed
in order to upgrade ${PACKAGE_NAME}.

exit 99
```

`${INSTALLED_VERSION}`	The currently installed version on your system. The upgrade fails because this version is too old.
`${PREV_VERSION}`	Before starting the upgrade procedure again, install this platform version on your system. The upgrade procedure works only when the old and the new versions are contiguous in sequence.
`exit 99`	The exit code accompanying the error message.

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/)`.

Migrate the database

Before starting with the steps described in this section, make sure that systems and processes are running, as described in detail in the configuration and more briefly in the bootstrapping sections.

Migrate PostegreSQL

When you upgrade the platform by downloading and decompressing a VM image, you may wish to import your existing database into the new platform version.

If you are upgrading the platform to a newer release, you need to migrate the existing database to the new platform release.
To perform the database migration, run the following script:

- Switch to the `eclecticiq` user by running the following command(s):

```
$ su - eclecticiq
```

```
$ /opt/eclecticiq/migrations/db-migration.sh
```

Reindex Elasticsearch

To reindex Elasticsearch, do the following:

- Make sure that ingestion, indexing, and core platform processes are *not running*.
  If they are running, stop them:

```
$ supervisorctl stop intel-ingestion:* intel-search-indexer platform-api
```

- Specify the platform settings environment variable by exporting it.
- Run `eiq-platform search reindex` to index or to reindex the Elasticsearch database.
- `reindex_elasticsearch` takes one argument: `index-name`.
  Its value is the name of an existing Elasticsearch index.
  Default Elasticsearch indexes for the platform:
  - *audit*
  - *documents*
  - *draft-entities*
  - *extracts*
  - *logstash-**
  - *statsd-**
  - *stix*

`search reindex` copies data from the PostgreSQL database to the Elasticsearch database, which is then indexed.

```
$ su -s /bin/bash elasticsearch -c "export
EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py;
/opt/eclecticiq/platform/api/bin/eiq-platform search reindex --index=<name_of_the_index>"
```

Migrate Elasticsearch indices

To make sure you are applying the latest Elasticsearch schema, migrate Elasticsearch indices. The migration process is idempotent. It sets up and builds the required/specified indices with aliases and mappings, and it updates the index mapping templates, if necessary.

- Make sure that ingestion, indexing, and core platform processes are *not running*.
  If they are running, stop them:

```
$ supervisorctl stop intel-ingestion:* intel-search-indexer platform-api
```

- Specify the platform settings environment variable by exporting it.
- Run `eiq-platform search upgrade` to migrate Elasticsearch indices.
  The command runs in the background. In case of an SSH disconnection, the process should keep running normally.

This upgrade action is idempotent. First, it tries to update the Elasticsearch index mappings in-place. If it is not possible, it proceeds to reindex the existing Elasticsearch indexes.

```
$ su -s /bin/bash elasticsearch -c "export
EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py;
/opt/eclecticiq/platform/api/bin/eiq-platform search upgrade"
```

- Index migration log messages, if any, are printed to the terminal. If no messages are printed to the terminal, the process completed successfully.

Migrate the graph database

> Make sure Neo4j is up and running before running the graph database migrations.

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

```
$ export EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
$ /opt/eclecticiq/platform/api/bin/eiq-platform graph upgrade
```

Check component versions

After migrating the database, check the versions of the upgraded components to verify that they are the correct ones. Authenticate with the platform API to receive a bearer token, then pass it to the `/api/versions` API endpoint to obtain version information:

```
$ curl -X GET
       -v
       --insecure
       -i
       -H "Content-Type: application/json"
       -H "Accept: application/json"
       -H "Authorization: Bearer <token>"
       https://platform.host/api/versions

# copy-paste version:
$ curl -X GET -v --insecure -i -H "Content-Type: application/json" -H "Accept: application/json"
-H "Authorization: Bearer <token>" https://platform.host/api/versions
```

The JSON response contains version information about the core platform components:

```
{
  "data": {

    "platform-api": {
      "branch": "master",
      "source": "github",
      "summary": "release/2.0dev-123-abc1234d",
      "tag": "latest",
      "version": "fb1234c2a62be5a409b38bedf65k273psgf99h54"
    },

    "platform-database": {
      "current": "109d2b29ead",
      "head": "109d2b29ead (head)",
      /*
        Allowed status values:
        - Up to date
        - Outdated
        - Head not available (when no manifest file is found)
      */
      "status": "Up to date"
    },

    "platform-ui": {
      "branch": "master",
      "source": "github",
      "summary": "release/2.0-tp11605-102-g5a04384",
      "tag": "latest",
      "version": "5a04384798f7fa5e9a4a888ss863f77e42abc66d"
    }
  }
}
```

The version information in the API JSON response matches the corresponding details in the manifest files. Manifest files are stored in the following directory:

```
$ cd /opt/eclecticiq/manifests/
```

The directory contains these manifest files:

```
platform-api.mf
platform-database.mf
platform-ui.mf
```

After migrating the database, the current database version needs to match the one in the *platform-database.mf* manifest file.
To perform this check, follow the steps described below.

- Request the current database version:

```
$ export EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
$ cd /opt/eclecticiq/migrations/
$ /opt/eclecticiq/platform/api/bin/eiq-platform database current-version
```

Example:

```
109d2b29ead (head)
```

- Go to the manifest directory:

```
$ cd /opt/eclecticiq/manifests/
```

- Open the *platform-database.mf* manifest file:

```
$ nano platform-database.mf
```

- Verify that the database head hash in the manifest file matches the returned current database version you requested:

```
head: 109d2b29ead (head)
```

Run the fixtures

To generate the default platform fixtures, run the following command(s):

```
$ export EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
$ /opt/eclecticiq/migrations/load-fixtures.sh
```

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
```

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
```

Rewire observables to v.2.0

Rewiring observables to v.2.0 is a one-off task you need to perform only when upgrading EclecticIQ Platform from v.1.14.x to 2.0.

After successfully upgrading EclecticIQ Platform from version 1.14.x to 2.0, you need to run `eiq-platform observable refresh`.

Run this script only after booting up the platform, and after starting all platform components. The platform needs to be fully up and running for the observable refresh script to work correctly.

From v.2.0 observables are powerful tools to drive IOC-centric analysis. Observables ingested and created with previous versions of the platform need to be rewired and upgraded to v.2.0, so that the platform can, among others, index them and make them searchable.

The script reparses existing observables to update their format to platform 2.0-compliant.

Run the script from the terminal or the command line:

```
$ export EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
$ /opt/eclecticiq/platform/api/bin/eiq-platform observable refresh --workers=4
```

`workers`

It is a mandatory parameter.
It takes an integer as a value.
If you do not specify any value, it defaults to `1`.

It defines the number of workers the script should concurrently run in parallel.
The process is resource-intensive, and it takes some time to complete: gauge the number of workers based on your system resources.
`4` workers is a rule-of-thumb value to run the script with a limited impact on normal platform operation.

Logging

- After initializing, the script starts discovering items to process, and outputting log information to the terminal.
- After successfully completing, it notifies you with the following message: `all entities processed`.

Errors

- If the user terminates the script before it completes, the script returns `Aborted!`.
- If an error occurs during execution, the script returns `an unexpected worker error occurred`, along with traceback information.

In case of errors, you may want to check the traceback information, as well as the `first_entity_id` and `last_entity_id` values in the log block preceding the error message: they correspond to the first and last entities in the last successfully processed batch before the error.

Example:

```
{"event": "sending batch to search", "first_entity_id": "00209576-bd04-48be-a4f0-c9a865b2b0c0",
"last_entity_id": "0024eceb-fb14-44ee-8e8d-6dad0ce58849", "level": "info", "logger":
"eiq.platform.scripts.bulk_refresh_extracts", "timestamp": "2017-09-01T16:26:44.685313Z",
"worker_pid": 9332}
```

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.

Access 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.

©2017 by EclecticIQ BV. All rights reserved.
Last generated on Oct 20, 2017