

Update Elastic Search

The SimplyE Circulation Manager (CM) uses the Elasticsearch search engine to enable keyword searches in the SimplyE app interfaces. We have completed the development required to support the current version (6.x) of Elasticsearch, which provides broader search capabilities and enhances overall service security. The easiest way for everyone to upgrade Elasticsearch is simply to deploy a new Elasticsearch instance and decommission the existing one.

Elasticsearch Deployments Types

- Installation via Docker using the Elastic Docker image (such as the [QuickStart Using Docker](#) demo)
- Deployment using the AWS ES service (such as the demo Ansible playbooks)
- Direct software installation on server

Because the version of Elasticsearch SimplyE currently uses is so far behind, it is not possible to upgrade current indexes. From <https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-upgrade.html>:

Elasticsearch can read indices created in the previous major version. Older indices must be reindexed or deleted. Elasticsearch 6.x can use indices created in Elasticsearch 5.x, but not those created in Elasticsearch 2.x or before. Elasticsearch 5.x can use indices created in Elasticsearch 2.x, but not those created in 1.x or before.

In any case, rebuilding the indexes is not onerous; therefore, the upgrade process simply deletes existing index data in the Postgres database and uses existing code to rebuild the database entries and Elasticsearch indexes. For those who have deployed the standard Docker container supplied by Elastic and those who have deployed using the AWS ES service, this makes the upgrade process a matter of replacing the existing Elasticsearch instance. Example processes for upgrading the Docker container or the AWS ES service are provided below.

For those who installed the Elasticsearch software, and perhaps the Circulation Manager code as well, natively on a host (direct software install), the general process will be similar to the outline we provide below. You will need to follow any specific instructions for your host operating system and hosting environment. Unfortunately, we can't supply specific instructions for your specific environment.

Also Updating the SimplyE Circulation Manager

Using Elasticsearch 6 with the Circulation Manager involves a different set of Python dependencies than version 1. For those deploying with Docker containers, our recommendation is to redeploy the CM containers with the appropriate Elasticsearch version specified. For production systems, performing the Elasticsearch upgrade as part of a normal CM upgrade may be the most convenient approach.

For those who have deployed the CM software directly, be certain to update the Python dependencies by running the appropriate `pip` command as shown in https://github.com/NYPL-Simplified/circulation-docker/blob/master/startup/03_elasticsearch_version.sh.

Development Roadmap for Elasticsearch 6 Support in the CM

As we transition from Elasticsearch version 1 to version 6, we have introduced an Elasticsearch version parameter which must be specified in container deployments as an environment variable (see example below). Setting that environment variable to 6 will enable the container to load the proper Python dependencies we need for Elasticsearch support. Here is a list of current and upcoming CM versions and their support for Elasticsearch:

CM Version < 2.3.2: Only Elasticsearch version 1 is supported.

CM Version 2.3.2+: Elasticsearch version 6 is now supported as an option; no specific version 6 coding is used. Elasticsearch version 1 is the default implementation and is still supported. To use version 6 in an existing CM implementation, CM containers must be redeployed with the version 6 parameter specified.

CM Version 2.3.6: Expected to be the last 2.x version, which supports either Elasticsearch 1 or Elasticsearch 6.

CM Version 3.0: Major CM upgrade where only Elasticsearch 6 is supported. There will be breaking changes. Therefore, Elasticsearch 1 services **must be upgraded** to Elasticsearch 6 prior to upgrading the Circulation Manager to v3.x.

Note: 2.3.x Implementations where a Circulation Manager is configured with Elasticsearch 6 dependencies, but is connecting to an Elasticsearch 1 server, will result in the inability for SimplyE apps to use keyword searches for titles in the CM's hosted collections.

Upgrade Process

The steps below describe the process for upgrading a CM and its Elasticsearch service. Some specific differences when installing CM software directly are noted in parentheses.

1. Create a new Elasticsearch service/instance (direct installs: uninstall the original version if necessary, then install the new v6 package).
2. Create a snapshot of the CM's RDS instance (AWS) or local Postgres container (direct installs: backup the CM's Postgres database).
3. Change the Elasticsearch service URL in the CM's Admin interface to point to the new Elasticsearch service/instance.
4. Deploy new instances of the CM containers (direct installs: checkout the new code and run pip to install the Elasticsearch 6 requirements).
5. Run the search index repair script.

Those who have deployed and maintained the SimplyE Circulation Manager service for a while may not need any further information to perform the upgrade. Post any questions you have on the Slack #devops channel.

On the other hand, if you are just using a demonstration system you wish to upgrade or are fairly new to the system, we've provided more details and some specific instructions at the end of this document as an example.

Q & A

When will the upgrade to Elasticsearch version 6 be required?

We haven't determined a specific target timeframe for implementing the Elasticsearch version 6 requirement. As mentioned above, though, **we have created a versioning roadmap** for the technical requirement: with version 3.0 of the CM, Elasticsearch 1 will no longer function properly. Before performing any upgrades to the system, always refer to the [CM release notes](#). With version 3.0 we will include a highlighted note that Elasticsearch version 6 is the new requirement.

What happens if I upgrade the Circulation Manager to version 3.0 without upgrading Elasticsearch to version 6?

When available, if you upgrade a CM deployment to version 3.0 without previously upgrading to Elasticsearch 6, keyword searches in the SimplyE clients /apps will fail. In addition, other management scripts/functions will fail, since we are moving creation of some feed data to the Elasticsearch engine. Upgrading Elasticsearch is a hard requirement: you **MUST** upgrade to Elasticsearch 6 and reset the Elasticsearch URL in the CM to your new service as described above.

Example Upgrade Instructions

As mentioned earlier, the way your particular CM system is deployed can vary widely from other implementations. The Elasticsearch upgrade process will be fairly straightforward for those who have been deploying SimplyE for a while. For those who are taking over someone else's work or have a simple demonstration system configured as described in the [Quickstart: Deploying SimplyE with Docker](#) guide, we provide an example upgrade with description and instructions below. The system deploys a Circulation Manager as a self-contained service on a single host server. Both Elasticsearch and Postgres are provided using Docker standard containers. This system is not intended for production, but production systems using Docker containers will be upgraded in very similar fashion. (We do include a few notes for those implementing using the AWS ES and RDS services in case it's helpful.)

Step 1. Create a new Elasticsearch service

Deploying as Local Elasticsearch Container

Assuming you have extra disk/storage space available on the host and that you'd like to be able to "roll back" the service if needed, deploying a new Elasticsearch 6 container will take three basic steps:

- stopping the existing CM service
- renaming the existing Elasticsearch container
- deploying the new Elasticsearch container

Assuming you have logged into the host supporting the CM service containers, follow the steps below. Notice that in the `docker run` command, you specifically ask for the Elasticsearch version 6 (latest) container image, as opposed to `elasticsearch:1` as you did according to the [Quickstart](#) guide.

```
sudo docker stop circ-scripts
sudo docker stop circ-webapp
sudo docker stop es
sudo docker rename es es-v1
sudo docker run -d --name es elasticsearch:6
sudo docker inspect es --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}"
```

The first two commands prepare the system for the upgrade. Stopping the CM web application is a good idea so there are no extraneous requests sent to a non-existent Elasticsearch service. Stopping the `es` container prepares it to be preserved in case you wish to roll back. To facilitate that, you need to rename the original container as shown in the third command.

The fourth command creates and runs a new Elasticsearch container on the host.

The last command, as it does in the [Quickstart](#) guide, enables us to see the local IP address assigned to the new container. This address becomes part of the Elasticsearch service URL, which will have the form `http://<ip_address>:9200`. Note this address for use in section 5 below.

NOTE: It is possible to configure specific Docker virtual networking and assign specific IPs to the various service containers. If you originally deployed your containers with custom Docker networking, review your documentation and use the `docker run` command for your specific implementation for the Elasticsearch container, simply substituting the version '6' here for the '1' originally specified.

Deploying using AWS ES Service

If you have deployed your Elasticsearch service using AWS ES, you can just deploy a new service instance. However, there are a couple of things to note.

First, ES instances cannot be stopped/halted/paused as was mentioned in the section above regarding a local `es` container. We *could* leave the service operational as we deploy a new one. But that leaves the possibility of problems after you upgrade the CM containers below. (By the way, leaving the original service operational has a downside: you would be billed for both ES instances after you complete this section, until you choose to remove the previous version service.)

You can, however, create an ES instance snapshot using the `aws` command-line tool (see <https://aws.amazon.com/elasticsearch-service/faqs/>). But you'll need to do some preliminary work configuring an S3 bucket and access rights to store the snapshot file. As we do experimentation, we may enhance this section with some basic instructions. But for now it remains a "homework" assignment as needed to create a snapshot to be used as a backup in case the previous service needs to be restored.

The second note is that the latest Elasticsearch version supported in the ES service is 6.4 at the time of this writing (whereas the container-based service noted above will result in a deployment of v6.7). Check the AWS documentation before you deploy to get the latest version available.

As shown in the section above, we recommend stopping the `circ-webapp` container on its EC2 instance to prevent any problems that may creep in with the Circulation Manager trying to connect/write data to the wrong ES service instance.

There are multiple ways to deploy new AWS services, from manually deploying using Amazon's web-based console to using an automation tool like Ansible to using the `aws` command line tool. Deploying the new ES service is, generally, a matter of just changing the version number supplied in your deployment method and redeploying.

As an example where Ansible is used to deploy a new ES service, your playbook might look like the following:

```
- name: Create Elasticsearch cluster
ec2_elasticsearch:
  name: "{{ es_instance_name }}"
  elasticsearch_version: "6.4"
  region: "{{ aws_region }}"
  instance_type: "t2.small.elasticsearch"
  instance_count: 1
  dedicated_master: False
  zone_awareness: False
  ebs: True
  volume_type: "gp2"
  volume_size: 10
  snapshot_hour: 13
  access_policies: "{{ lookup('template', 'templates/es_cluster_policies.j2', convert_data=False) | from_json }}"
  register: aws_es_service

- set_fact:
  es_endpoint: "{{ aws_es_service.response.DomainStatus.Endpoint }}"

- name: Display the AWS ES service URL
debug:
msg: "ES URL: https://{{ es_endpoint }}:9200"
```

The key data points are the instance `name` and `elasticsearch_version`. You can review your Ansible playbook, and then update version value. The `name` element is important to note only if you want to retrieve/display the endpoint value using the `aws` tool from the command line, as shown below. (You can also see the endpoint in the ES console in the domain's *Overview* tab.)

```
aws es describe-elasticsearch-domain --domain-name <name> --query 'DomainStatus.Endpoints' --output json
```

Remember that the ES URL you need to configure your CM is in the form `https://<es_endpoint>:9200`.

Alternatively, you can add Ansible tasks like the last two in the snippet to display the Elasticsearch endpoint you've created. The example shows capturing an endpoint where ES is configured as a publicly accessible service; change the final `.Endpoint` to `.Endpoints.vpc` if your ES service is configured within a VPC.

Step 2. Backup Your CM Database

Create a backup of your Circulation Manager Postgres database (or snapshot the RDS instance) if you choose to have the capability of "rolling back" to your previous ES instance. Again, there are a number of software clients to use in creating your database backup. Perform your database backup as you normally would, from using a simple `psql` command line approach to creating a snapshot of an AWS RDS database instance.

As an example of performing a full backup of the CM database as configured in the *Quickstart* guide, you could issue the following command (change the Postgres values as needed to match your implementation):

```
pg_dump --format=c --file=datestamp_simplified_circ_db_full.sqlc -U postgres -h 172.17.0.4 simplified_circ_db
```

You can specify a particular path as needed to store the output file. Also, the host IP address is an example here; it could be different depending on the order in which you started the containers. To find the IP address of the `pg` container in your implementation, issue the command:

```
sudo docker inspect pg --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}"
```

Step 3. Configure New Elasticsearch URL

Once the new deployment is complete with no errors, go to the CM's Admin interface in your browser and login. You need to change the Elasticsearch search service URL to point to the new Elasticsearch service URL:

1. Click the *System Configuration* item in the top menu bar.
2. Click the *Search* tab in the left sidebar.
3. Click the *Edit* button for the Elasticsearch configuration.
4. Replace the value in the *URL* field with the new Elasticsearch service endpoint from Section 1 above.
5. Click the *Submit* button to save the change.

Step 4. Deploy New CM Containers

Redeploy the Circulation Manager containers, and perhaps upgrade, using version 2.3.3 or greater. At the time of this writing, version 2.3.6 is the latest CM version; we use that in the example below, since, ultimately, an upgrade to version 2.3.6 is recommended prior to upgrading subsequently to the new 3.x family.) The key element here is adding the new environment variable which specifies the Elasticsearch version to use: `SIMPLIFIED_ELASTICSEARCH_VERSION`. This must be set to the value 6. Be sure to substitute the placeholders with your CM implementation's actual database values.

```
sudo docker run --name circ-webapp \  
-d -p 80:80 \  
-e SIMPLIFIED_ELASTICSEARCH_VERSION="6" \  
-e SIMPLIFIED_PRODUCTION_DATABASE='postgres://[username]:[password]@[ip_address]:[port]/[database_name]' \  
-e SIMPLIFIED_DB_TASK="auto" \  
nypl/circ-webapp:2.3.6  
sudo docker run -d --name circ-scripts \  
-e TZ="US/Central" \  
-e SIMPLIFIED_ELASTICSEARCH_VERSION="6" \  
-e SIMPLIFIED_DB_TASK='auto' \  
-e SIMPLIFIED_PRODUCTION_DATABASE='postgres://[username]:[password]@[ip_address]:[port]/[database_name]' \  
nypl/circ-scripts:2.3.6
```

If you have deployed your Postgres database as an AWS RDS service, substitute your implementation's RDS endpoint for the `[ip_address]` placeholder.

Step 5. Re-create the Search Index

Run the search index repair script to recreate the proper database entries and Elasticsearch index entries to support searching:

```
sudo docker exec -it circ-webapp /bin/bash  
source env/bin/activate  
bin/repair/search_index
```

Step 6. Exit Container

Exit the virtual environment and container. You can also log out of the host if desired.

```
deactivate  
exit
```

At this point, you should have a functional CM using Elasticsearch 6. Test access and searching in your SimplyE apps to verify. If you have questions or issues along the way, post them in the Slack **#devops** channel.