deviantony / **docker-elk**

The ELK stack powered by Docker and Compose.

#docker #elasticsearch #logstash #kibana #elk #docker-compose #searchguard

175 commits | **3** branches | **0** releases | **46** contributors | MIT

Branch: **master** | New pull request | Find File | Clone or download

deviantony Persist Elasticsearch data by default via named volume (#430) ... Latest commit `1e8b471` 8 days ago

.github	Add GitHub issue template (#263)	2 years ago
.travis	Travis: Return exit code of spawned process in password setup script	3 months ago
elasticsearch	Update links to sources of base images	2 months ago
extensions	Make APM server listen on non-loopback interfaces (#414)	2 months ago
kibana	Update links to sources of base images	2 months ago
logstash	Update links to sources of base images	2 months ago
.env	Update to 7.3.1	9 days ago
.gitattributes	Fix line endings issue of shell script files on Windows	3 years ago
.travis.yml	Update Swarm stack to 7.2.1	last month
LICENSE	Added LICENSE.	4 years ago
README.md	Persist Elasticsearch data by default via named volume (#430)	8 days ago
docker-compose.yml	Persist Elasticsearch data by default via named volume (#430)	8 days ago
docker-stack.yml	Update to 7.3.1	9 days ago

README.md

Elastic stack (ELK) on Docker

Run the latest version of the Elastic stack with Docker and Docker Compose.

It gives you the ability to analyze any data set by using the searching/aggregation capabilities of Elasticsearch and the visualization power of Kibana.

> ℹ The Docker images backing this stack include Stack Features (formerly X-Pack) with paid features enabled by default (see How to disable paid features to disable them). The trial license is valid for 30 days.

Based on the official Docker images from Elastic:

- Elasticsearch
- Logstash
- Kibana

Other available stack variants:

- `searchguard` : Search Guard support

Contents

1. Requirements
   - Host setup
   - SELinux
   - Docker for Desktop
     - Windows
     - macOS
2. Usage
   - Bringing up the stack
   - Cleanup
   - Initial setup
     - Setting up user authentication
     - Injecting data
     - Default Kibana index pattern creation
3. Configuration
   - How to configure Elasticsearch
   - How to configure Kibana
   - How to configure Logstash
   - How to disable paid features
   - How to scale out the Elasticsearch cluster
4. Extensibility
   - How to add plugins
   - How to enable the provided extensions
5. JVM tuning
   - How to specify the amount of memory used by a service
   - How to enable a remote JMX connection to a service
6. Going further
   - Using a newer stack version
   - Plugins and integrations
   - Swarm mode

Requirements

Host setup

- Docker Engine version **17.05+**
- Docker Compose version **1.12.0+**
- 1.5 GB of RAM

By default, the stack exposes the following ports:

- 5000: Logstash TCP input
- 9200: Elasticsearch HTTP
- 9300: Elasticsearch TCP transport
- 5601: Kibana

> ⓘ Elasticsearch's bootstrap checks were purposely disabled to facilitate the setup of the Elastic stack in development environments. For production setups, we recommend users to set up their host according to the instructions from the Elasticsearch documentation: Important System Configuration.

SELinux

On distributions which have SELinux enabled out-of-the-box you will need to either re-context the files or set SELinux into Permissive mode in order for docker-elk to start properly. For example on Redhat and CentOS, the following will apply the proper context:

```
$ chcon -R system_u:object_r:admin_home_t:s0 docker-elk/
```

Docker for Desktop

Windows

Ensure the Shared Drives feature is enabled for the `C:` drive.

macOS

The default Docker for Mac configuration allows mounting files from `/Users/`, `/Volumes/`, `/private/`, and `/tmp` exclusively. Make sure the repository is cloned in one of those locations or follow the instructions from the documentation to add more locations.

Usage

Bringing up the stack

Clone this repository, then start the stack using Docker Compose:

```
$ docker-compose up
```

You can also run all services in the background (detached mode) by adding the `-d` flag to the above command.

> ℹ You must run `docker-compose build` first whenever you switch branch or update a base image.

If you are starting the stack for the very first time, please read the section below attentively.

Cleanup

Elasticsearch data is persisted inside a volume by default.

In order to entirely shutdown the stack and remove all persisted data, use the following Docker Compose command:

```
$ docker-compose down -v
```

Initial setup

Setting up user authentication

> ℹ Refer to How to disable paid features to disable authentication.

The stack is pre-configured with the following **privileged** bootstrap user:

- user: *elastic*
- password: *changeme*

Although all stack components work out-of-the-box with this user, we strongly recommend using the unprivileged built-in users instead for increased security. Passwords for these users must be initialized:

```
$ docker-compose exec -T elasticsearch bin/elasticsearch-setup-passwords auto --batch
```

Passwords for all 6 built-in users will be randomly generated. Take note of them and replace the `elastic` username with `kibana` and `logstash_system` inside the Kibana and Logstash configuration files respectively. See the Configuration section below.

> ℹ Do not use the `logstash_system` user inside the Logstash *pipeline* file, it does not have sufficient permissions to create indices. Follow the instructions at Configuring Security in Logstash to create a user with suitable roles.

Restart Kibana and Logstash to apply the passwords you just wrote to the configuration files.

```
$ docker-compose restart kibana logstash
```

> ℹ Learn more about the security of the Elastic stack at Tutorial: Getting started with security.

Injecting data

Give Kibana about a minute to initialize, then access the Kibana web UI by hitting http://localhost:5601 with a web browser and use the following default credentials to log in:

- user: *elastic*
- password: *<your generated elastic password>*

Now that the stack is running, you can go ahead and inject some log entries. The shipped Logstash configuration allows you to send content via TCP:

```
$ nc localhost 5000 < /path/to/logfile.log
```

You can also load the sample data provided by your Kibana installation.

Default Kibana index pattern creation

When Kibana launches for the first time, it is not configured with any index pattern.

Via the Kibana web UI

> ℹ You need to inject data into Logstash before being able to configure a Logstash index pattern via the Kibana web UI. Then all you have to do is hit the *Create* button.

Refer to Connect Kibana with Elasticsearch for detailed instructions about the index pattern configuration.

On the command line

Create an index pattern via the Kibana API:

```
$ curl -XPOST -D- 'http://localhost:5601/api/saved_objects/index-pattern' \
    -H 'Content-Type: application/json' \
    -H 'kbn-version: 7.3.1' \
    -u elastic:<your generated elastic password> \
    -d '{"attributes":{"title":"logstash-*","timeFieldName":"@timestamp"}}'
```

The created pattern will automatically be marked as the default index pattern as soon as the Kibana UI is opened for the first time.

Configuration

> ℹ Configuration is not dynamically reloaded, you will need to restart individual components after any configuration change.

How to configure Elasticsearch

The Elasticsearch configuration is stored in `elasticsearch/config/elasticsearch.yml`.

You can also specify the options you want to override by setting environment variables inside the Compose file:

```
elasticsearch:

  environment:
    network.host: _non_loopback_
    cluster.name: my-cluster
```

Please refer to the following documentation page for more details about how to configure Elasticsearch inside Docker containers: Install Elasticsearch with Docker.

How to configure Kibana

The Kibana default configuration is stored in `kibana/config/kibana.yml`.

It is also possible to map the entire `config` directory instead of a single file.

Please refer to the following documentation page for more details about how to configure Kibana inside Docker containers: Running Kibana on Docker.

How to configure Logstash

The Logstash configuration is stored in `logstash/config/logstash.yml`.

It is also possible to map the entire `config` directory instead of a single file, however you must be aware that Logstash will be expecting a `log4j2.properties` file for its own logging.

Please refer to the following documentation page for more details about how to configure Logstash inside Docker containers: Configuring Logstash for Docker.

How to disable paid features

Switch the value of Elasticsearch's `xpack.license.self_generated.type` option from `trial` to `basic` (see License settings).

How to scale out the Elasticsearch cluster

Follow the instructions from the Wiki: Scaling out Elasticsearch

Extensibility

How to add plugins

To add plugins to any ELK component you have to:

1. Add a `RUN` statement to the corresponding `Dockerfile` (eg. `RUN logstash-plugin install logstash-filter-json`)
2. Add the associated plugin code configuration to the service configuration (eg. Logstash input/output)
3. Rebuild the images using the `docker-compose build` command

How to enable the provided extensions

A few extensions are available inside the `extensions` directory. These extensions provide features which are not part of the standard Elastic stack, but can be used to enrich it with extra integrations.

The documentation for these extensions is provided inside each individual subdirectory, on a per-extension basis. Some of them require manual changes to the default ELK configuration.

JVM tuning

How to specify the amount of memory used by a service

By default, both Elasticsearch and Logstash start with 1/4 of the total host memory allocated to the JVM Heap Size.

The startup scripts for Elasticsearch and Logstash can append extra JVM options from the value of an environment variable, allowing the user to adjust the amount of memory that can be used by each component:

Service	Environment variable
Elasticsearch	ES_JAVA_OPTS
Logstash	LS_JAVA_OPTS

To accomodate environments where memory is scarce (Docker for Mac has only 2 GB available by default), the Heap Size allocation is capped by default to 256MB per service in the `docker-compose.yml` file. If you want to override the default JVM configuration, edit the matching environment variable(s) in the `docker-compose.yml` file.

For example, to increase the maximum JVM Heap Size for Logstash:

```
  logstash:

    environment:
      LS_JAVA_OPTS: -Xmx1g -Xms1g
```

How to enable a remote JMX connection to a service

As for the Java Heap memory (see above), you can specify JVM options to enable JMX and map the JMX port on the Docker host.

Update the `{ES,LS}_JAVA_OPTS` environment variable with the following content (I've mapped the JMX service on the port 18080, you can change that). Do not forget to update the `-Djava.rmi.server.hostname` option with the IP address of your Docker host (replace **DOCKER_HOST_IP**):

```
  logstash:

    environment:
      LS_JAVA_OPTS: -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxre
```

Going further

Using a newer stack version

To use a different Elastic Stack version than the one currently available in the repository, simply change the version number inside the `.env` file, and rebuild the stack with:

```
$ docker-compose build
$ docker-compose up
```

> ℹ Always pay attention to the upgrade instructions for each individual component before performing a stack upgrade.

Plugins and integrations

See the following Wiki pages:

- External applications
- Popular integrations

Swarm mode

Experimental support for Docker Swarm mode is provided in the form of a `docker-stack.yml` file, which can be deployed in an existing Swarm cluster using the following command:

```
$ docker stack deploy -c docker-stack.yml elk
```

If all components get deployed without any error, the following command will show 3 running services:

```
$ docker stack services elk
```

> ℹ To scale Elasticsearch in Swarm mode, configure *zen* to use the DNS name `tasks.elasticsearch` instead of `elasticsearch`.