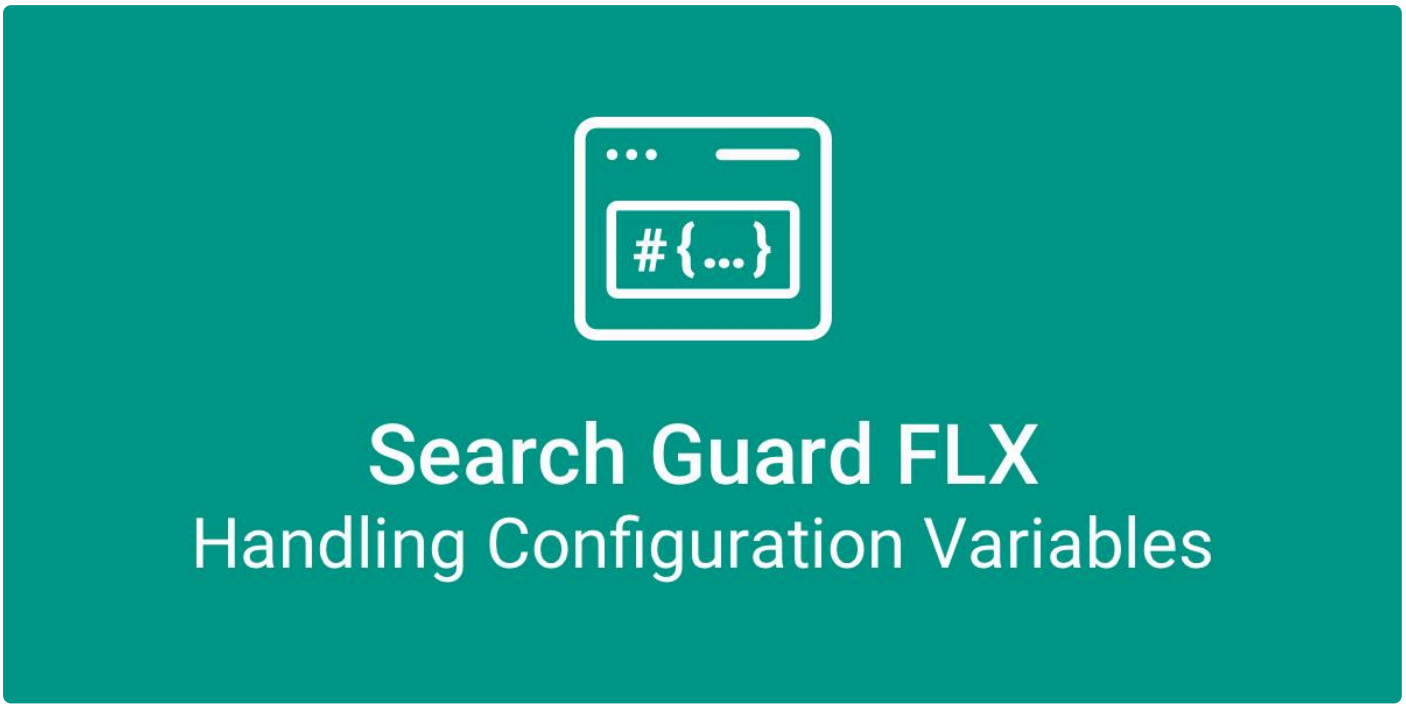
Handling Configuration Variables in Search Guard FLX

Jochen Kressin · 2022-08-09



Configuration Variables Handling so Far

The configuration files for Search Guard would in most cases also contain sensitive information. For example passwords, certificates, system user names and so on. So far the only way to avoid having this data in configuration files in plaintext was to use environment variables.

You can replace the plaintext passwords by variables in any configuration file.

```
ldap:
  http_enabled: true
    ...
    config:
      hosts:
        - ${env.LDAP_HOST:-ldapdev.example.com}
      bind_dn: ${env.LDAP_BIND_DN}
      password: ${env.LDAP_BIND_DN_PASSWORD}
      ...
```

Here, the sensitive LDAP connection settings have been replaced with variables. The syntax is:

```
${env.<variable name>}

or

${env.<variable name>:-<default value>}
```

You can substitute any setting in any Search Guard configuration file with variables.

When uploading the files with sgadmin, the variables can be replaced in two ways:

Substitution by sgadmin

Before uploading the files to your Elasticsearch cluster, sgadmin will scan the files for variables, and then substitute them by environment variables. This means the environment variables need to be set on the machine where sgadmin is executed, e.g.:

```
export LDAP_HOST="ldapprod.example.com"
export LDAP_BIND_DN="cn=admin,ou=people,dc=example,dc=com"
export LDAP_BIND_DN_PASSWORD="mypassword"
```

This means you have to configure the variables only on the machine where you run sgadmin. As a downside, if you want to run it from any other machine, you have to make sure the environment variables match. Also, the content of the variables cannot be changed at runtime.

Substitution by Elasticsearch Nodes

You can also configure the environment variables on each Elasticsearch nodes. In this case, sgadmin will upload the configuration files with the variables, and the Search Guard plugin takes care of the substitution before the configuration is applied.

The downside of this approach is that you need to keep the environment variables on all nodes in sync.

For a complete reference please refer to our documentation.

A Better Approach in FLX: Configuration Variables

Search Guard FLX ships with a built-in configuration variables store. With this, you can create, change and delete variables and secrets centrally on your Elasticsearch cluster. The variables are then applied whenever the configuration is modified/reloaded.

Search Guard stores the variables, which can optionally be encrypted, in a protected index. This is much like the Search Guard configuration index. You can only access this index with an admin certificate. Regular users are never able to access this index directly.

Managing secrets is easy, you can use the new sgtcl, a replacement for the older sgadmin, for it.

First, connect to your cluster by specifying the admin TLS certificate:

```
$ ./sgctl.sh connect localhost
  --ca-cart /path/to/root-ca.pem
  --cert /path/to/admin-cert.pem
  --key /path/to/admin-cert-private-key.pem`
```

Next, add a new configuation variable. In our case it is the password for connecting to the LDAP server:

```
./sgctl.sh add-var ldap_bind_dn_password mypassword --encrypt
```

Since we are adding senstive information here, we chose to encrypt the variable. This way, Search Guard never stores the password in cleartext. Even if you create a backup of the configuration, the variable value will be encrypted.

You can change existing variables like:

```
/sgctl.sh update-var ldap_bind_dn_password mynewpassword --encrypt
```

Or, delete a variable completely:

```
/sgctl.sh dlete-var ldap_bind_dn_password
```

You can then reference this variable in any configuration file:

```
auth_domains:
- type: basic/ldap
  ldap.idp.password: '#{var:ldap_bind_dn_password}'
  ...
```

The variable substitution is executed each time the Search Guard configuration index is reloaded. This is the case

- when a node restarts
- when any Search Guard configuration changes
- when a configuration variable is added, changed or deleted

Creating Backups

If you want to create a backup of all configuration variables, you can use the sgctl get-config command.

```
$ ./sgctl.sh get-config
```

The backup of your security configuration also includes the variables in the file sg*config*vars.yml.

Summary

The apporach we used in Search Guard was based on substituting configuration variables with environment variables. This works well, but also had one major drawback: Your security configuration was dependant on your system infrastructure.

With our new Variables Store you can now manage your sensitive configuration variables in a central place: Your Elasticsearch cluster. Search Guard FLX will protect this index from unauthorized access and makes sure any changes are available on all nodes instantly. If you want to rotate any secret or key, just update the configuration variable. Done. Pretty neat, right?

This article was published on the Search Guard blog: https://search-guard.com/blog/handling-configuration-variables-search-guard-flx/