ONTOTEXT ANSWER:

Security automation is a fairly frequent task and forgetting an important password is an inevitability of life. Fortunately, both problems are really easy to solve using the default GraphDB security module.

Security is controlled by the file **settings.js.**, located under **./work/workbench** in your GraphDB home directory. By default, this would be the root GraphDB installation directory. The file gets created when you first run GraphDB. It is JSON-formatted. Besides the security module, it contains other useful information such as user queries, configured locations, etc. Given how important this file is, we strongly recommend making sure it's backed up. You'll have to do it manually, as GraphDB backups are done per-repository and this is a global per-installation file. If you are deploying with Docker, it's a good idea to keep that file in a volume.

Passwords are hashed using *bcrypt*. So, if you've somehow lost your only administrator password, the easiest way to fix that is to simply set it again.

```
"password" :
"{bcrypt}$2a$10$L6v7FUcxBIE2N.hbdXahk.iA.kGcW5/t.kr0uzvrOyCQY1iRwjMJm"
,
```

This would set your password to the default value, `root`. If you have deleted your administration users, you can simply write up a new one, for example, by copying it from a working default GraphDB installation.

As for automation, everything in the "**settings.js**" file can be configured with an API call. So, it's up to you to do one of the following:

- Store and deploy a **settings.js** file – this options is much easier and faster than the rest but it's an all-or-nothing. It also means that if someone edits your settings with an API call, you may have some unexpected content ingested in your system. Still, you might prefer this option if, for example, you want to store all modifications to the database. In such cases, you may even keep **settings.js** under version control.
- String substitution – the well known and often surprisingly tricky tool can be applied

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

here. String substitution scripts break easily but at least you'll be sure that you are not storing modifications. Additionally, this file doesn't appear before the database has been first initialized, so the method is only good on subsequent runs.
- API calls – this is the "official" way to perform modifications of the workbench settings. You wouldn't be storing modifications here either and this method is safer than string substitution. The drawback is that it must happen on a working database.

For our own deployments, we favor storing and deploying the **settings.js** file, usually as a Docker volume. We sometimes do API calls as well.

Did this help you solve your issue? Your opinion is important not only to us but also to your peers.

Help us get better: review GraphDB on Gartner

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com