Title: How to handle Docker-Secrets in application.properties files

Post Body:

How do you inject Docker secrets (files/data from /run/secrets) into the application.properties files? Is it safe to use environment variables?

Accepted Answer:

First of all, usage of environment variables for secret data for the `application.properties` isn't safe.

You have mainly two options when talking about Secrets.

> If you are using Docker Secrets **without Docker Swarm** then you can directly load the whole `application.properties` in a secret, mount it under `/run/secrets` and refer to it as configuration file with the Spring flags.
>
> If you are using Docker Secrets **with Docker Swarm** then you can just store as secret the concrete fields that you're interested in and relate to them using the Configuration Templates of Swarm.

Example:

```
echo -n "myUser" | docker secret create db_user - echo -n "myPass" | docker secret create db_password - echo -n "jdbc://..." |
```

application.properties.tmpl

```
spring.datasource.url={{ secret "db_url" }} spring.datasource.user={{ secret "db_user" }} spring.datasource.password={{ secret
```

docker-compose.yml

```
version: '3.9' services:   api:     image: yourapp:1.0.0   configs:     - source: application.properties       target: /usr/ap
```

When you deploy with `docker stack deploy -c docker-compose.yml myapp`, it will automatically populate the configuration with the contents of the secrets and it will mount it in the destination path.

Highest Rated Answer:

The best way to use docker secrets in application.properties seems to be using configuration trees.

If your docker secrets get mounted to /run/secrets (which is the default) then you just need to put `spring.config.import=optional:configtree:/run/secrets/` at the top of application.properties. Then you can use for example the docker secret `db_password` as `${db_password}`.

See https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#features.external-config.files.configtree