Support the use of docker secrets for the database password #149

Closed **pmarzouk** opened this issue on 30 Mar 2018 · 4 comments

pmarzouk commented on 30 Mar 2018

Hi,

Docker secrets allows to make secrets available in the container as files.
Environment variables are not supported.

Would it be possible to support setting the environment variable for the database password from a file like this is done in the postgres official container in docker-entrypoint.sh

In the postgres container, the POSTGRES_PASSWORD environment variable in the compose file can contain either the password itself or the path to the secrets file (eg: /run/secrets/postgres_password)

2

xlson commented on 3 Apr 2018 • edited Contributor

Sounds like a good idea. We'd have to support some other secrets as well, AWS credentials at the very least. Not sure how well used docker secrets are vs other ways of providing credentials in a more secure manner.

efrecon commented on 4 May 2018 Contributor

Actually, "official" Docker library images make the difference between the "raw" variable and the one coming from a (secret) file. They seem to use the convention of environment variables ending with `_FILE` for reading the value from a file instead.

DidelotK commented on 7 May 2018

This feature would be appreciated. At the moment i have made this workaround, but something more generic would be great.

```
#!/usr/bin/env bash

# Docker secrets support
if [ -f /run/secrets/GRAFANA_USER ]; then
  export GF_SECURITY_ADMIN_USER=$(< /run/secrets/GRAFANA_USER)
fi

if [ -f /run/secrets/GRAFANA_PASSWORD ]; then
  export GF_SECURITY_ADMIN_PASSWORD=$(< /run/secrets/GRAFANA_PASSWORD)
fi

/run.sh
```

dsw9742 commented on 7 May 2018

I also use a workaround, though all credit must go to the authors of this article.

Added generic `env_secrets_expand.sh` script:

```
#!/bin/sh

: ${ENV_SECRETS_DIR:=/run/secrets}

env_secret_debug()
{
    if [ ! -z "$ENV_SECRETS_DEBUG" ]; then
        echo -e "\033[1m$@\033[0m"
    fi
}

# usage: env_secret_expand VAR
#    ie: env_secret_expand 'XYZ_DB_PASSWORD'
# (will check for "$XYZ_DB_PASSWORD" variable value for a placeholder that defines the
#  name of the docker secret to use instead of the original value. For example:
# XYZ_DB_PASSWORD={{DOCKER-SECRET:my-db.secret}}
env_secret_expand() {
    var="$1"
    eval val=\$$var
    if secret_name=$(expr match "$val" "{{DOCKER-SECRET:\([^}]\+\)}}$"); then
        secret="${ENV_SECRETS_DIR}/${secret_name}"
        env_secret_debug "Secret file for $var: $secret"
        if [ -f "$secret" ]; then
            val=$(cat "${secret}")
            export "$var"="$val"
            env_secret_debug "Expanded variable: $var=$val"
        else
            env_secret_debug "Secret file does not exist! $secret"
        fi
    fi
}

env_secrets_expand() {
    for env_var in $(printenv | cut -f1 -d"=")
    do
        env_secret_expand $env_var
    done

    if [ ! -z "$ENV_SECRETS_DEBUG" ]; then
        echo -e "\n\033[1mExpanded environment variables\033[0m"
        printenv
    fi
}

env_secrets_expand
```

Added new `entrypoint.sh` script to wrap the secrets script and the standard grafana `run.sh` script:

```
#!/bin/bash

export GF_SECURITY_ADMIN_USER={{DOCKER-SECRET:"${DOCKER_SECRET_GF_SECURITY_ADMIN_USER}"}}
export GF_SECURITY_ADMIN_PASSWORD={{DOCKER-SECRET:"${DOCKER_SECRET_GF_SECURITY_ADMIN_PASSWORD}"}}

source env_secrets_expand.sh

source run.sh
```

Include additional secret and env flags in `docker service create` command:

```
...
 --env "DOCKER_SECRET_GF_SECURITY_ADMIN_USER=grafana-user"
 --env "DOCKER_SECRET_GF_SECURITY_ADMIN_PASSWORD=grafana-pass"
 --secret grafana-user
 --secret grafana-pass
...
```

1

efrecon mentioned this issue on 21 May 2018

Support for env variables ending with _FILE #166 Merged

xlson closed this in `89b7c50` on 1 Jun 2018

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

5 participants