Docker Secrets

by Vladyslav Marchenko

Hello!
Today I tried to create a docker secret for a Docker Swarm stack.
Why?
Because, you can use secrets to manage any sensitive data which a container needs at runtime but you don’t want to store in the image or in source control, such as.

Let's take a closer look!

wordpress-hillel > docker-compose.yaml

```yaml
version: "3.8"
services:
  db:
    image: amd64/mysql:8
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD_FILE: /run/secrets/db_root_password
      MYSQL_DATABASE: wordpress
      MYSQL_USER_FILE: /run/secrets/db_mysql_user
      MYSQL_PASSWORD_FILE: /run/secrets/db_password
    secrets:
      - db_root_password
      - db_password
      - db_mysql_user
  wordpress:
    depends_on:
      - db
    image: amd64/wordpress:6
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER_FILE: /run/secrets/db_mysql_user
      WORDPRESS_DB_PASSWORD_FILE: /run/secrets/db_password
    secrets:
      - db_password
      - db_mysql_user


secrets:
  db_password:
    file: db_password.txt
  db_root_password:
    file: db_root_password.txt
  db_mysql_user:
    file: db_mysql_user.txt
volumes:
  db_data: {}

```

So, first of all , I made a docker-compose config which creates a simple WordPress site using three secrets in a compose file.

Let's break down the above file. Here is what's happening:

- The secrets line under each service defines the Docker secrets you want to inject into the specific container.
- The main secrets segment defines the variables db_password, db_mysql_user and db_root_password and a file that should be used to populate their values.
- The deployment of each container means Docker creates a temporary filesystem mount under /run/secrets/<secret_name> with their specific values.

When deployed, Docker creates these three secrets and populates them with content from the file specified in the build file.

The DB service uses three secrets, while WordPress uses two.

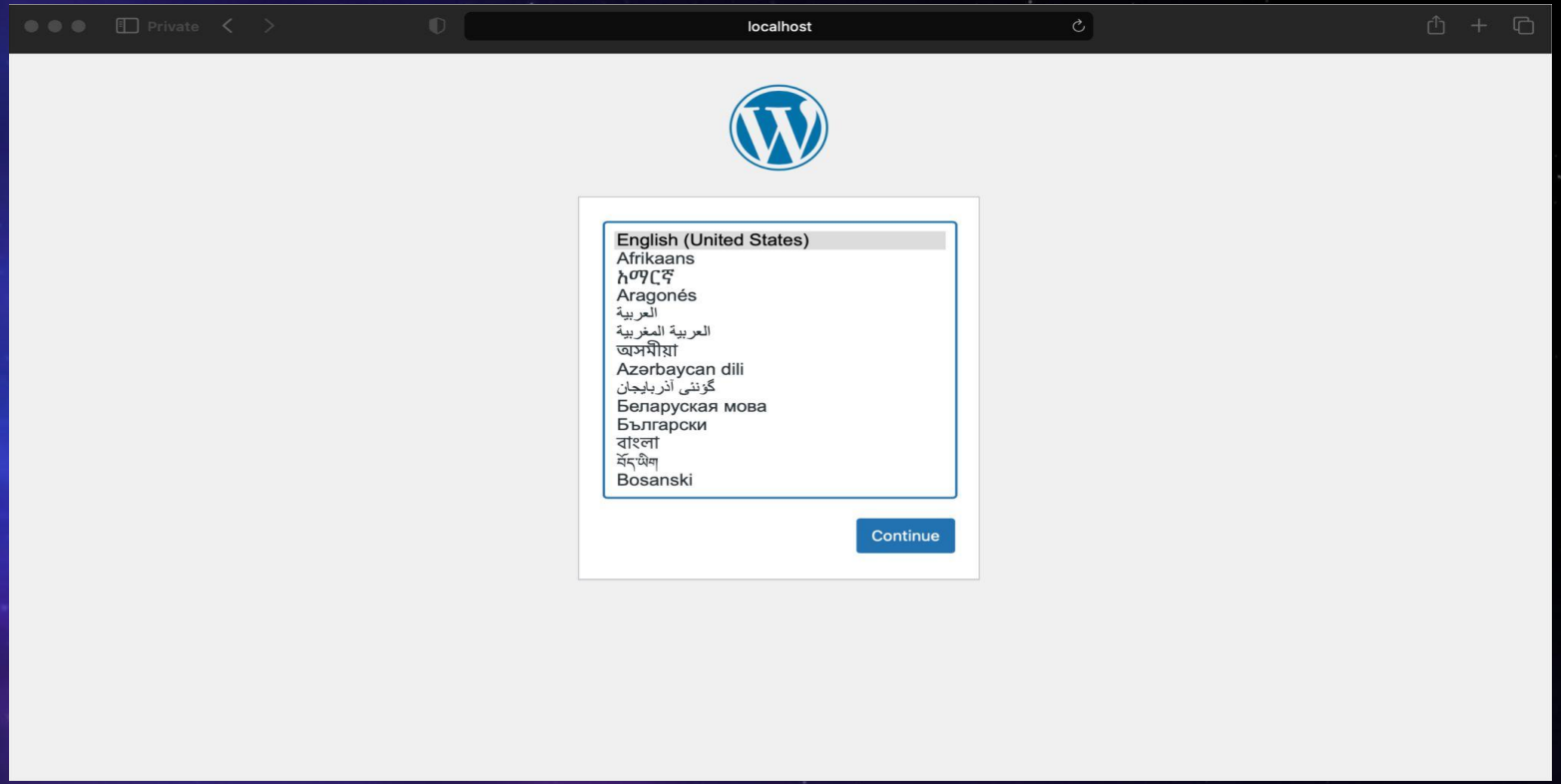
Before deployment, it is important to create three files in the root directory from which the secret will take the password and username, in my example these are the db_password.txt , db_mysql_user.txt and db_root_password.txt files.

Now let's see how it works in prod!

```
vmarchenko@vmarchenko777-4 wordpress-hillel % docker swarm init
Swarm initialized: current node (ac62ka3saj212ogmfd8latbna) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-4up9y6450nn18eq251yk2sht8zbqg32cz6bxl39gymdlqmbcyc-3iwme33q0xrqczf5zzngmfvqp 192.168.65.4:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

vmarchenko@vmarchenko777-4 wordpress-hillel % docker stack deploy -c docker-compose.yaml wordpress-hillel
Ignoring unsupported options: restart

Creating network wordpress-hillel_default
Creating secret wordpress-hillel_db_password
Creating secret wordpress-hillel_db_root_password
Creating secret wordpress-hillel_db_mysql_user
Creating service wordpress-hillel_wordpress
Creating service wordpress-hillel_db
vmarchenko@vmarchenko777-4 wordpress-hillel % docker secret ls
ID                          NAME                                DRIVER    CREATED          UPDATED
cx3jpj9401l5tid533xi5afzc   wordpress-hillel_db_mysql_user                14 seconds ago   14 seconds ago
smrtkv6rby7vk6wrrvtnargep   wordpress-hillel_db_password                  14 seconds ago   14 seconds ago
66z40frfne4jdq7tkj3omnjlj   wordpress-hillel_db_root_password             14 seconds ago   14 seconds ago
vmarchenko@vmarchenko777-4 wordpress-hillel % docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED         STATUS                  PORTS                  NAMES
ed249b7428dc   amd64/mysql:8      "docker-entrypoint.s…"   1 second ago    Up Less than a second   3306/tcp, 33060/tcp    wordpress-hillel_db.1.d5er6kcq49hbt29xkaaqrbcfo
a0c21cc6096c   amd64/wordpress:6  "docker-entrypoint.s…"   5 seconds ago   Up 3 seconds            80/tcp                 wordpress-hillel_wordpress.1.xthlzsh59vn59ucmy6kocmi3j
vmarchenko@vmarchenko777-4 wordpress-hillel %
```

1. I enabled swarm mode via command docker swarm init. (A Docker Swarm is a group of either physical or virtual machines that are running the Docker application and that have been configured to join together in a cluster).
2. I deployed my docker-compose config.

Than, we see that our stack includes two containers (mysql-server and wordpress), and via the docker secret ls command, we see that docker creates three secrets.



And as a result, we get a working WordPress site!

You can find all files and config in my git repository at the link:
https://github.com/VladMar35/DevOps/tree/main/Docker%20Secrets