Single Instance with App-Only Image - Swarm

Overview

In our steps for setting up a single instance of Yellowfin, Yellowfin runs on port 8080 of your Docker Swarm cluster, with 4GB of allocated RAM.

Before deploying a single instance with these defaults, make sure you have already created a repository database and synced it with the same version of Yellowfin that will be used in the Yellowfin container. To do this, download the full application installer for Yellowfin, and install it on your workstation. This will create a Yellowfin repo DB as well as an instance of Yellowfin in a folder which can be deleted after configuring the containers.

For a list of supported database types, see the database information on Install And Deploy Yellowfin.

To deploy a single instance of Yellowfin, follow the steps below.

1. Install the full application installer version of Yellowfin on your workstation (this is temporary to ensure the repo DB is available for the containers to use)
2. Copy the web.xml file from this installation and save it as a backup to your preferred location (this acts as a reference for the Yellowfin credentials required to connect to your Yellowfin repo DB)
3. Ensure Docker is running in swarm mode
4. Copy the following text and paste it into your preferred text editor:

```
version: '3'
services:
  yellowfin-instance:
    ports:
      - "8080:8080" # Maps Yellowfin running on port 8080 to Docker Swarm port 8080
    deploy:
      replicas: 1
    environment:
      # Required environment variables
      - JDBC_CLASS_NAME=INSERT_DATABASE_TYPE_HERE # Database driver class name
      - JDBC_CONN_URL=jdbc:INSERT_JDBC_CONNECTION_STRING_HERE # Database connection string
      - JDBC_CONN_USER=INSERT_DATABASE_USER_HERE # Username to use when accessing the database
      - JDBC_CONN_PASS=INSERT_JDBC_PASSWORD_HERE # Password for the database user
      - JDBC_CONN_ENCRYPTED=true # Flag for indicating if the database user's password supplied is
encrypted or not.
      - APP_MEMORY=4096 # The amount of memory in megabytes to assign to the Yellowfin Application.
    image: "yellowfinbi/yellowfin-app-only:<RELEASE_VERSION_GOES_HERE>" # Path to the app-only image of
Yellowfin
```

5. Read through the above text and replace the environment variable placeholders with your own configuration details (these are located in the web.xml file of the Yellowfin installation); here's an example to connect to a PostgreSQL instance (with a sample IP address for the connection):

```
      # Required environment variables
      - JDBC_CLASS_NAME=org.postgresql.Driver # Database driver class name
      - JDBC_CONN_URL=jdbc:postgresql://192.168.1.50/docker_swarm_yellowfin_single_instance # Database
connection string
      - JDBC_CONN_USER=postgres # Username to use when accessing the database
      - JDBC_CONN_PASS=bXF0oj5gnB1oRB1kZq5 # Password for the database user
      - JDBC_CONN_ENCRYPTED=true # Flag for indicating if the database user's password supplied is
encrypted or not.
      - APP_MEMORY=4096 # The amount of memory in megabytes to assign to the Yellowfin Application.
    image: "yellowfinbi/yellowfin-app-only:9.6.0" # Path to the app-only image of Yellowfin
```

6. Save the text to a YAML file called **yellowfin-single-istance.yml**
7. Run the following command in a terminal to deploy Yellowfin and execute it in the background:
   `docker stack deploy --compose-file yellowfin-single-instance.yml yellowfin`
8. Start Yellowfin by typing your host URL on port 8080
9. Ensure that Yellowfin is running from your container and that you can login (this confirms that your login credentials are correct, so you can safely delete the workstation instance of Yellowfin)
10. Delete the workstation instance of Yellowfin by removing the folder

Section navigation
Current topic - Install in a Container

The page is part of the Install in a Container topic contains the following pages, split by Docker and Kubernetes:

This page is part of the Install And Deploy Yellowfin section of the wiki, which has these topics:

Install on Premises

Install in the Cloud

Install in a container

Deploy Yellowfin

Advanced Deployments

Advanced Deployments

top