Single Instance with App-Only Image and Load Balancer

Overview

In our steps for setting up a single instance of Yellowfin, Yellowfin runs on port 80 (standard HTTP port) of your specified DNS hostname or IP address, with 4GB of allocated RAM, with Traefik fronting it. .

When deploying Yellowfin with Traefik fronting it, the Yellowfin Kubernetes service will default to the “ClusterIP” service type in Kubernetes, so it will not expose any ports to the external interface of the Kubernetes cluster.

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 Kubernetes is running and that Traefik has been installed
4. Copy the following text and paste it into your preferred text editor:

```
---
### Yellowfin Standalone Service ###
apiVersion: v1
kind: Service
metadata:
  name: yellowfin-standalone

spec:
  ports:
    - protocol: TCP
      name: web
      port: 8080
  selector:
    app: yellowfin-standalone
---
### Yellowfin Standalone Deployment ###
kind: Deployment
apiVersion: apps/v1
metadata:
  namespace: default
  name: yellowfin-standalone
  labels:
    app: yellowfin-standalone

spec:
  replicas: 1
  selector:
    matchLabels:
      app: yellowfin-standalone
  template:
    metadata:
      labels:
        app: yellowfin-standalone
    spec:
      containers:
        - env:
          - name: APP_MEMORY
            value: "4096"
          - name: JDBC_CLASS_NAME
            value: INSERT_DATABASE_TYPE_HERE
          - name: JDBC_CONN_ENCRYPTED
            value: "true"
          - name: JDBC_CONN_PASS
            value: INSERT_JDBC_PASSWORD_HERE
          - name: JDBC_CONN_URL
```

```
            value: jdbc:INSERT_JDBC_CONNECTION_STRING_HERE
          - name: JDBC_CONN_USER
            value: INSERT_DATABASE_USER_HERE
          name: yellowfin-standalone
          image: yellowfinbi/yellowfin-app-only:<RELEASE_VERSION_GOES_HERE>
          ports:
            - name: web
              containerPort: 8080

---
### Yellowfin Standalone Ingress ###
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: yellowfinstandaloneingressroute
  namespace: default
spec:
  entryPoints:
    - web
  routes:
  - match: Host(`INSERT_DNS_HOSTNAME`)
    kind: Rule
    services:
    - name: yellowfin-standalone
      port: 8080
      sticky:
        cookie:
          httpOnly: true
          name: stickyCookie
```

5. Read through the above text and replace the database connection settings with your own configuration details (these are located in the web.xml file of the Yellowfin installation)
6. In the configuration text, find the section starting with ### Yellowfin Standalone Ingress ### and add your own server details. At a minimum, replace `INSERT_DNS_HOSTNAME` with your own DNS name (or IP address) for Traefik to listen to, for routing requests to the Yellowfin instance. For example:

```
### Yellowfin Standalone Ingress ###
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: yellowfinstandaloneingressroute
  namespace: default
spec:
  entryPoints:
    - web
  routes:
  - match: Host(`yellowfin.example.com`)
    kind: Rule
    services:
    - name: yellowfin-standalone
      port: 8080
      sticky:
        cookie:
          httpOnly: true
          name: stickyCookie
```

7. Save the text to a YAML file called **yellowfin-single-yml**
8. Run the following command in a terminal to deploy Yellowfin and execute it in the background:
   `Kubectl apply –f yellowfin-single-instance.yml`
9. Start Yellowfin by typing your host URL
10. 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
11. Delete the workstation instance of Yellowfin by removing the folder

top

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:

Single Instance with App-Only Image and Load Balancer

- Deploy to Docker without Swarm
  - Sandbox Instance with All-In-One Image
  - Single Instance with App-Only Image
  - Multiple Discrete Instances with App-Only Image
  - A Cluster with App-Only Image
- Deploy to Docker with Swarm
  - Sandbox instance with All-In-One Image - Swarm
  - Single Instance with App-Only Image - Swarm
  - Multiple Discrete Instances with App-Only Image - Swarm
  - A Cluster with App-Only Image - Swarm

Kubernetes

- Deploy to Kubernetes without load balancing
  - Sandbox Instance with All-In-One Image - no Load Balancer
  - Multiple Discrete Instances with App-Only Image - no Load Balancer
- Deploy to Kubernetes with Load Balancing
  - Single Instance with App-Only Image and Load Balancer
  - A Cluster with App-Only Image and Load Balancer

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

Install on Premises

Single Instance with App-Only Image and Load Balancer

- Installation Steps

Install in the Cloud

Install in the Cloud

- Yellowfin for AWS
- Yellowfin for Azure
- Yellowfin for Google Cloud Platform

Install in a container

Install in a Container

- Docker
- Kubernetes
- Upgrading Yellowfin Container Deployment

Deploy Yellowfin

Deploy Yellowfin

- Logs and Logging
- Yellowfin Directory Structure
- User Welcome

Advanced Deployments

Advanced Deployments

- Clustering Guide
- Yellowfin Server Specification
- Automate Yellowfin Deployment on Linux
- SAML Bridge
- Standalone Configuration Tools

top