

# - Swarm

YellowfinYellowfin4GBRAMDocker 808080818082Traefik

Docker Swarm [Docker](#)SwarmYellowfin Yellowfin Cookie[Traefik](#)

Yellowfin Yellowfin

Yellowfin Yellowfin YellowfinYellowfinYellowfin Yellowfin

[Yellowfin](#)

3Yellowfin

1. Yellowfin ()
2. web.xml (Yellowfin Yellowfin )
3. DockerSwarmTraefik
- 4.

```
version: '3'
services:
  yellowfin-cluster:
    ports:
      - "8080:8080" # Maps Yellowfin running on port 8080 to Docker Swarm port 8080
      #- "7801:7800" # Maps the Yellowfin cluster port to an external port on the host (Optional)
    hostname: "yellowfin-node-{{.Task.Slot}}" # Optional, sets the hostname to the provided value.
    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.
      - CLUSTER_INTERFACE=match-interface:eth1 #Sets the JGroups "Bind_addr" attribute. Adjust as needed
        if the containers aren't communicating with each other.
      - CLUSTER_PORT=7800 # TCP Port to use for cluster networking, used to connect between containers
      - NODE_BACKGROUND_TASKS=REPORT_BROADCAST_BROADCASTTASK,REPORT_BROADCAST_MIREPORTTASK,FILTER_CACHE,
        SOURCE_FILTER_REFRESH,SOURCE_FILTER_UPDATE_REMINDER,THIRD_PARTY_AUTORUN,ORGREF_CODE_REFRESH,
        ETL_PROCESS_TASK,SIGNALS_DCR_TASK,SIGNALS_ANALYSIS_TASK,SIGNALS_CLEANUP_TASK,COMPOSITE_VIEW_REFRESH,
        SIGNALS_CORRELATION_TASK # Comma separated list of which background Task Types can be run on this node.
      - NODE_PARALLEL_TASKS=4,4,4,4,4,4,4,4,4,4,4,4 # Comma separated list of the number of concurrent
        tasks for each Task Type that can be run on

    deploy:
      replicas: 2 # Total number of instances to deploy - this example deploys two
      labels:
        - "traefik.enable=true" # Tell Traefik to route to these instances, works with exposedbydefault
          param in Traefik
        - "traefik.docker.network=yellowfin-cluster_yf-cluster" # IMPORTANT: This will allow Traefik to
          route to the Yellowfin instances. Format: <%NameOfSwarmStack%>_<%NameOfDockerNetwork%>
        - "traefik.http.routers.yellowfin.rule=Host(`INSERT_DNS_HOSTNAME`)" #IMPORTANT: The URL/DNS Name
          that you want Traefik to use for routing to your Yellowfin instances. Eg: `yellowfin.example.com`
        - "traefik.http.routers.yellowfin.entrypoints=web" # Utilizes Traefik's web entrypoint
        - "traefik.http.services.yellowfin.loadBalancer.server.port=8080" # Traefik to route to the
          Yellowfin application port
        - "traefik.http.services.yellowfin.loadBalancer.sticky.cookie" # Enables sticky sessions support

    networks:
      yf-cluster: # The network to add the Yellowfin instances to
    image: "yellowfinbi/yellowfin-app-only:<RELEASE_VERSION_GOES_HERE>" # Path to the app-only image of
      Yellowfin

  traefik:
    image: traefik # Pulls Traefik from Docker Hub
    command:
      - "--providers.docker.endpoint=unix:///var/run/docker.sock" #Gives Traefik access to the Docker API
      - "--providers.docker.swarmMode=true" # Tells Traefik we're using Docker Swarm
```

```

- "--providers.docker.exposedbydefault=true" # Exposed by default will auto-route to any Docker
containers/services that have the right labels.
- "--providers.docker.network=yf-cluster" # The network Traefik should be monitoring. Must match
the one Yellowfin will be deployed in.
- "--providers.docker.watch=true" # Watch for Docker Events
- "--entrypoints.web.address=:80" # Run Traefik on port 80
- "--api=true" # Enables the Traefik API
ports:
- 80:80 # Runs Traefik on port 80 (HTTP)
- 8090:8080 # Optional - Runs Traefik on port 8090 (if wanting to use the dashboard)
volumes:
# So that Traefik can listen to Docker events
- /var/run/docker.sock:/var/run/docker.sock:ro
networks:
- yf-cluster # The network to add Traefik to.
deploy:
placement:
constraints:
- node.role == manager # IMPORTANT, only allows Traefik to deploy to a manager node. This is
recommended by Traefik.
networks:
yf-cluster: # Creates a network using the overlay driver
driver: overlay

```

5. yellowfin-cluster\_yf-clusterSwarmDocker Stack
6. ('INSERT\_DNS\_HOSTNAME') YellowfinTraefikDNS
7. (Yellowfin web.xml )PostgreSQL

```

# Required environment variables
- JDBC_CLASS_NAME=org.postgresql.Driver # Database driver class name
- JDBC_CONN_URL=jdbc:postgresql://192.168.1.50/docker_yellowfin_cluster # Database connection string
- JDBC_CONN_USER=postgres # Username to use when accessing the database
- JDBC_CONN_PASS=bXF0oj5gnBlorB1kZq5 # 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.
- CLUSTER_INTERFACE=match-interface:eth1 #Sets the JGroups "Bind_addr" attribute. Adjust as needed if
the containers aren't communicating with each other.
- CLUSTER_PORT=7800 # TCP Port to use for cluster networking
- NODE_BACKGROUND_TASKS=REPORT_BROADCAST_BROADCASTTASK,REPORT_BROADCAST_MIREPORTTASK,FILTER_CACHE,
SOURCE_FILTER_REFRESH,SOURCE_FILTER_UPDATE_REMINDER,THIRD_PARTY_AUTORUN,ORGREF_CODE_REFRESH,
ETL_PROCESS_TASK,SIGNALS_DCR_TASK,SIGNALS_ANALYSIS_TASK,SIGNALS_CLEANUP_TASK,COMPOSITE_VIEW_REFRESH,
SIGNALS_CORRELATION_TASK # Comma separated list of which background Task Types can be run on this node.
- NODE_PARALLEL_TASKS=4,4,4,4,4,4,4,4,4,4,4,4 # Comma separated list of the number of concurrent
tasks for each Task Type that can be run on
image: "yellowfinbi/yellowfin-app-only:9.6.0" # Path to the app-only image of Yellowfin

```

8. yellowfin-cluster.ymlYAML
9. Yellowfin
  - docker stack deploy --compose-file yellowfin-cluster.yml yellowfin-cluster
10. URL8080 () Yellowfin
11. Yellowfin (Yellowfin)
12. Yellowfin

---

DockerKubernetes

[Docker](#)

- [SwarmDocker](#)

- Sandbox
  - 
  - 
  -
- SwarmDocker
  - Sandbox - Swarm
  - - Swarm
  - - Swarm
  - - Swarm

## Kubernetes

- Kubernetes
  - Sandbox -
  - -
- Kubernetes
  - 
  -

## Yellowfin

- 
- 

- AWSYellowfin
- AzureYellowfin
- Google Cloud PlatformYellowfin

- Docker
- Kubernetes
- Yellowfin

## Yellowfin

### Yellowfin

- 
- Yellowfin
- 

- 
- Yellowfin
- LinuxYellowfin
- SAML Bridge
-

