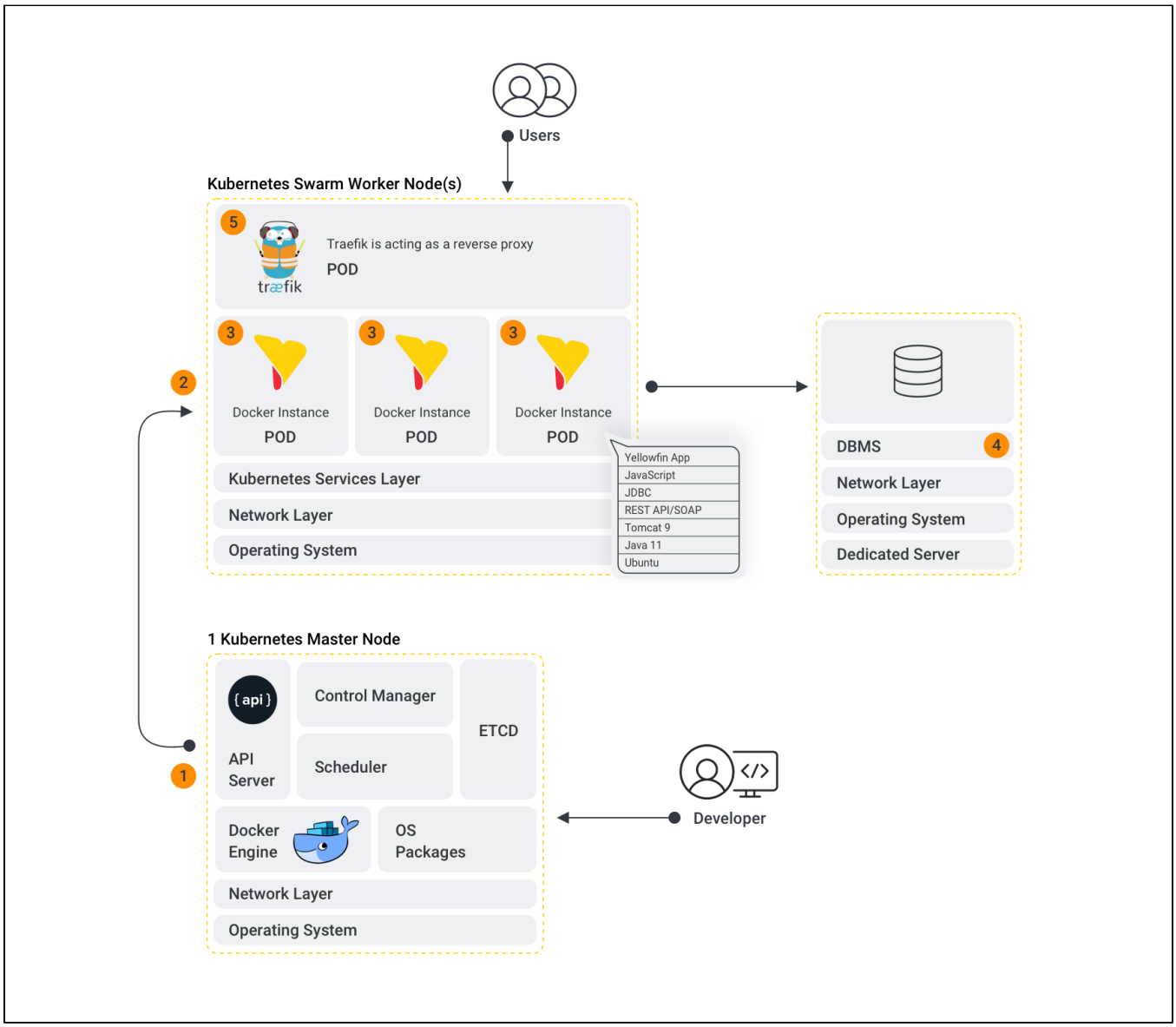
Kubernetes

Kubernetes Architecture

The following diagram shows a typical Yellowfin cluster running in a Kubernetes environment, which typically has the most components of our Kubernetes deployment examples. Note that we've used Traefik in our architecture, but any container-aware proxy that makes use of sticky sessions could be used in its place.



Breaking down the above diagram:

1. Kubernetes Master node:
2. Kubernetes Worker Node(s): Yellowfin containers sit within the worker node or nodes of a Docker Swarm environment.

3. The Yellowfin component: Yellowfin pods that form the Yellowfin cluster. These pods may be placed on multiple Kubernetes worker nodes depending on available resources in the Kubernetes cluster. Depending on the Yellowfin deployment type, the Kubernetes environment will have one or more Yellowfin instances running, with those instances connecting to either the same database (for a clustered Yellowfin deployment) or different database (discrete instance deployment).
4. DBMS: For performance reasons, we currently recommend not running the repository database in a Kubernetes Pod for production workloads. Instead, we recommend a dedicated database server; for example, for AWS, an EC2 instance or using AWS RDS.
5. Traefik: Traefik is a container-aware reverse proxy that runs Kubernetes environment and handles the load balancing and sticky sessions for the Yellowfin instances. We go into more detail about Traefik and Kubernetes later on (link).
   For single instance deployments of Yellowfin on Kubernetes, there will only be a single Yellowfin pod present.

top

Other scenarios

For multiple discrete Yellowfin instances on Kubernetes, deploying Traefik is optional. If it is deployed in this situation, it can route requests to the discrete Yellowfin deployments, instead of exposing ports on the Kubernetes cluster that route directly to the Yellowfin instances. Alternatively, you can avoid Traefik if you have a container-aware proxy that supports sticky sessions (such as NGINX).

In a multiple discrete Yellowfin instances on Kubernetes, each Yellowfin instance will be communicating with a different Yellowfin repository database.

If you're using the Yellowfin All-In-One image, it does not require the external Yellowfin repository database, as the image comes embedded with its own.

top

Preparing for deployment

Before deploying Yellowfin on Kubernetes, make sure you've chosen which type of deployment suits your requirements. In our examples, we've deployed Yellowfin in a Kubernetes environment via Kubectl. We chose Kubectl because it's a free command line tool to interact with the Kubernetes control plane (the master node), which is well documented, but you can choose a different tool if you have a preference.

Although Yellowfin on Kubernetes will run without it, we highly recommend your environment uses a reverse proxy capable of handling load balancing and sticky session support. We've provided deployment steps for environments with proxies (using Traefik) and environments without proxies.

Pre-requisites

Before deploying Yellowfin on Kubernetes, you will need

- a running Kubernetes cluster;
- Kubectl installed on your workstation/deployment system;
- an understanding of Application Server Security; and,
- an understanding of reverse proxies and load balancing in a container environment.

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:

        - 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

Kubernetes

- 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