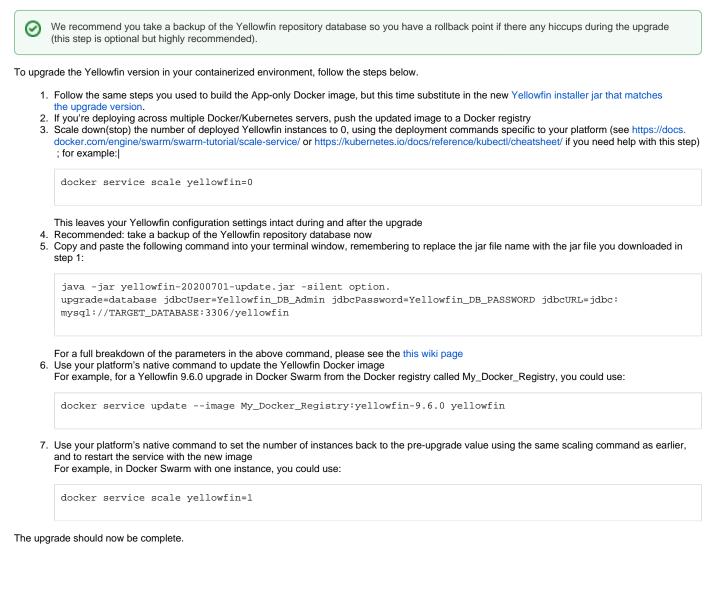# Upgrading Yellowfin Container Deployment

All-In-One Yellowfin image

The All-In-One Yellowfin image is a sandboxed deployment with a built-in database repository, which means any upgrades also upgrade the database, removing any existing content in the process. The All-In-One image is therefore **not suitable for upgrades, and best used for testing purposes only**.

App-Only Yellowfin image

The Yellowfin App-Only image retains content during version upgrades. The upgrade process is similar to a standard Yellowfin cluster upgrade, with a few extra steps to account for a Docker/Kubernetes deployment.

> ✅ We recommend you take a backup of the Yellowfin repository database so you have a rollback point if there any hiccups during the upgrade (this step is optional but highly recommended).

To upgrade the Yellowfin version in your containerized environment, follow the steps below.

1. Follow the same steps you used to build the App-only Docker image, but this time substitute in the new Yellowfin installer jar that matches the upgrade version.
2. If you're deploying across multiple Docker/Kubernetes servers, push the updated image to a Docker registry
3. Scale down(stop) the number of deployed Yellowfin instances to 0, using the deployment commands specific to your platform (see https://docs. docker.com/engine/swarm/swarm-tutorial/scale-service/ or https://kubernetes.io/docs/reference/kubectl/cheatsheet/ if you need help with this step) ; for example:|

   ```
   docker service scale yellowfin=0
   ```

   This leaves your Yellowfin configuration settings intact during and after the upgrade
4. Recommended: take a backup of the Yellowfin repository database now
5. Copy and paste the following command into your terminal window, remembering to replace the jar file name with the jar file you downloaded in step 1:

   ```
   java -jar yellowfin-20200701-update.jar -silent option.
   upgrade=database jdbcUser=Yellowfin_DB_Admin jdbcPassword=Yellowfin_DB_PASSWORD jdbcURL=jdbc:
   mysql://TARGET_DATABASE:3306/yellowfin
   ```

   For a full breakdown of the parameters in the above command, please see the this wiki page
6. Use your platform's native command to update the Yellowfin Docker image
   For example, for a Yellowfin 9.6.0 upgrade in Docker Swarm from the Docker registry called My_Docker_Registry, you could use:

   ```
   docker service update --image My_Docker_Registry:yellowfin-9.6.0 yellowfin
   ```

7. Use your platform's native command to set the number of instances back to the pre-upgrade value using the same scaling command as earlier, and to restart the service with the new image
   For example, in Docker Swarm with one instance, you could use:

   ```
   docker service scale yellowfin=1
   ```

The upgrade should now be complete.

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:

Install on Premises

Install in the Cloud

Install in a container

Deploy Yellowfin

Advanced Deployments

- Standalone
  Configuration Tools