

# How to use WUD to monitor container updates

WUD monitors running containers, compares them to the latest registry tags, and displays everything in a simple dashboard.

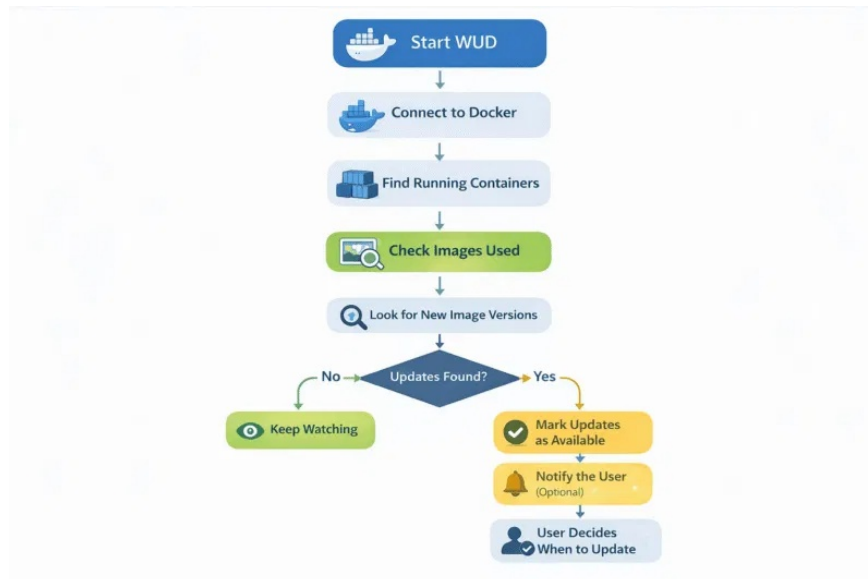
Updating Docker containers is fairly easy when there are only a few services. But as the system grows, things quickly become complicated. Each container has its own tags and release cycle, meaning you need to constantly switch between Docker Hub and GitHub just to see if anything has changed. We need a better way to update containers without constant manual checking.

That's when many people started using What's Up Docker (WUD) . WUD monitors running containers, compares them to the latest registry tags, and displays everything in a simple console. It doesn't force updates; instead, it provides you with the necessary information to decide when to take action.

## How What's Up Docker (WUD) Works

WUD is a lightweight tool that helps monitor your Docker containers without modifying them. It simply monitors the containers you select by adding labels (e.g., `wud.watch=true`) to the 'docker-compose.yml' file. This keeps the console simple and clean.

When the WUD detects a container, it compares the local image to the latest version available in the repository. It understands versioning rules, so you can choose which updates to monitor, such as only minor updates, while ignoring major, error-causing changes. Finally, the web console displays all results and clearly marks containers as updated or in need of updating.



## Setting up WUD is very easy.

Setting up a WUD is very simple. You can run a WUD using a small and simple 'docker-compose.yml' file. To do this, first create a directory to store the WUD data and move it into that directory using the `cd` command.

```
mkdir wud && cd wud
```

Inside this directory, create a file named 'docker-compose.yml' and add the following configuration:

```
services: wud: image: getwud/wud:latest container_name: wud security_opt: - no-
```

**Note :** Keep the settings to a minimum and focus on local monitoring. You can add notifications later using triggers.

Finally, run the following command to start the WUD container in the background:

```
docker compose up -d
```

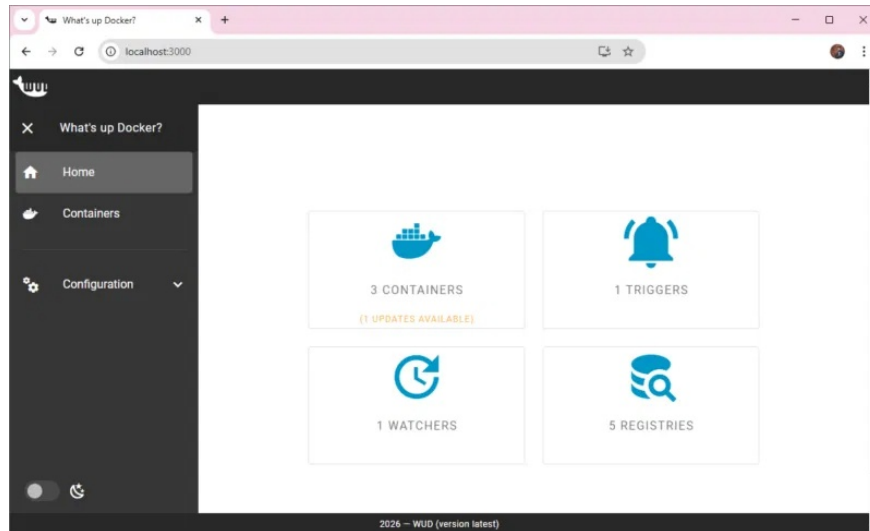
```

C:\Users\HP\wud>docker compose up -d
[*] Running 14/14
wud Pulled
  ✓a4f1e659873b Pull complete 97.6s
  ✓61320b01ae5e Pull complete 92.7s
  ✓0fd11e38a340 Pull complete 64.9s
  ✓d80554801d95 Pull complete 9.8s
  ✓167d9c25e0ff Pull complete 89.8s
  ✓4f4fb700ef54 Pull complete 92.9s
  ✓58d4fec82140 Pull complete 8.8s
  ✓94161915a4b4 Pull complete 86.4s
  ✓a2abe4d18138 Pull complete 93.8s
  ✓dd7774383295 Pull complete 1.6s
  ✓25b64eb3daf5 Pull complete 2.0s
  ✓8e85a9237243 Pull complete 1.9s
  ✓7f2938325747 Pull complete 1.7s
  ✓7f2938325747 Pull complete 86.8s
[*] Running 1/1
  ✓Network wud_default Created 0.2s
  ✓Container wud Started 0.9s
C:\Users\HP\wud>
  
```

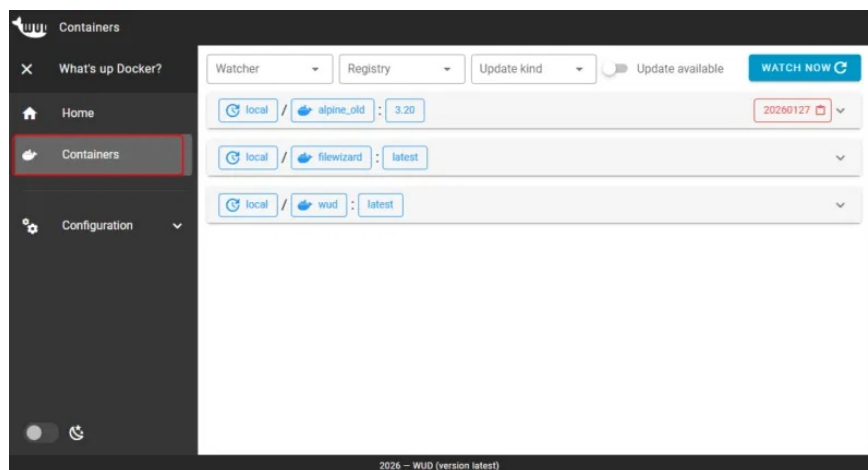
Once the setup is complete and the container is running, you can access the WUD at <http://localhost:3000> in your browser.

## See everything at a glance: Explore the WUD dashboard.

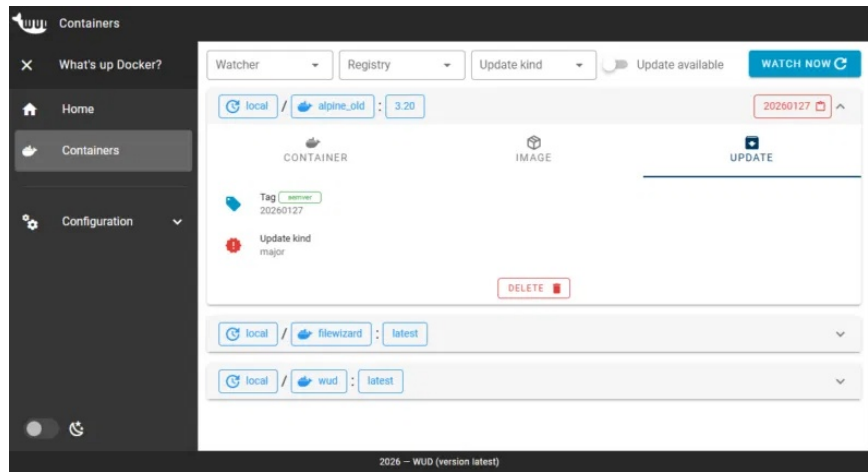
When you access the WUD dashboard, the homepage immediately displays a clear overview of containers, monitors, registry, and triggers, making it easy to understand what the WUD monitors and how everything connects.



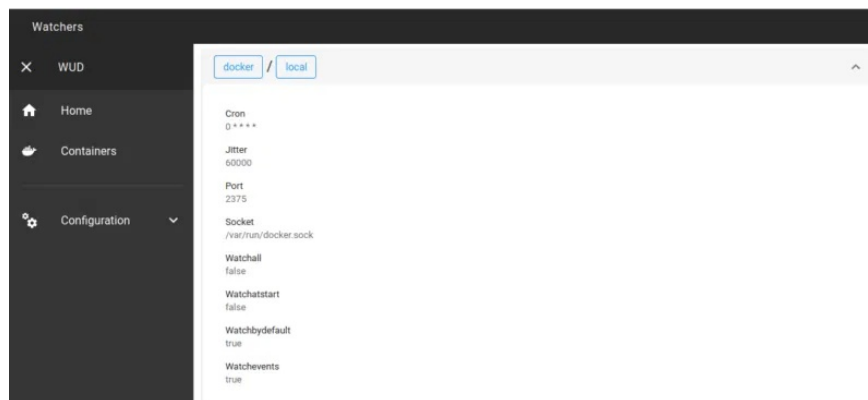
Containers are the heart of WUD. This is where you can see all your running Docker containers, along with their current image versions and update status.



The WUD clearly shows whether a container has been updated or if there's a newer image available, so you don't need to manually check tags or change logs.



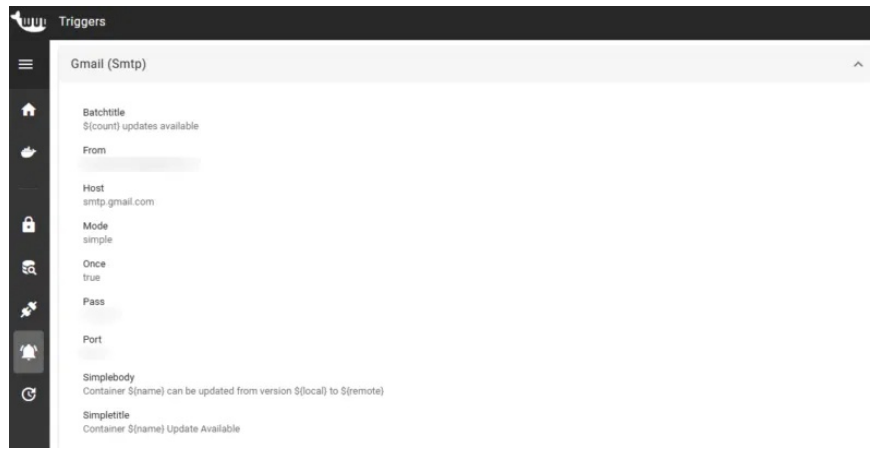
**Watchers** represent the Docker hosts that the WUD is monitoring. In the example setup, a watcher defines how the WUD connects to Docker, including the Docker socket location, the frequency of update checks, and optional cron schedules. This is useful when you are monitoring multiple servers or environments, as each watcher can be configured independently.



**Registries** define the origin of container images. These are upstream image sources, such as Docker Hub, GitHub Container Registry, or GitLab. By configuring repositories, you can control how the WUD authenticates and retrieves version information, which is especially useful when working with private images.



Triggers determine what happens when an update is detected. Instead of automatically updating the container, WUD gives you control.



You can configure triggers to send emails, activate webhooks, run scripts, or send notifications to services like Telegram, Discord, Slack, or Pushover.

In this way, WUD helps you consider, decide, and act precisely as you want when managing updates in a real-world production or self-hosted environment, instead of blindly updating containers.

You finished reading the article "**How to use WUD to monitor container updates**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.