Docker Hardened Images quickstart

- Step 1: Find an image to use
- Step 2: Pull and run the image
- Step 3: Compare with other images
- What's next

This guide shows you how to go from zero to running a Docker Hardened Image (DHI) using a real example. At the end, you'll compare the DHI to a standard Docker image to better understand the differences. While the steps use a specific image as an example, they can be applied to any DHI.

This quickstart uses DHI Community images from `dhi.io`. You sign in with your Docker account, pull and run an image, and compare it with a Docker Official Image.

> If you have a DHI Select or Enterprise subscription, see Get started with DHI Select and Enterprise instead. Select and Enterprise use mirrored repositories in your organization namespace on Docker Hub to enable customization, SLA-backed security updates, and access to compliance variants.

Step 1: Find an image to use

1. Go to the Hardened Images catalog in Docker Hub.
2. Use the search bar or filters to find an image (for example, `python`, `node`, or `golang`). For this example, search for `python`.
3. Select the Python repository to view its details.

Continue to the next step to pull and run the image. To dive deeper into searching and evaluating images, see Search and evaluate Docker Hardened Images.

Step 2: Pull and run the image

You can pull and run a DHI like any other Docker image. Note that Docker Hardened Images are designed to be minimal and secure, so they may not include all the tools or libraries you expect in a typical image. You can view the typical differences in Considerations when adopting DHIs.

The following example demonstrates that you can run the Python image and execute a simple Python command just like you would with any other Docker image:

1. Open a terminal and sign in to the Docker Hardened Images registry using your Docker account credentials.

   ```
   $ docker login dhi.io
   ```

   > If you don't have a Docker account, create a free account to get started.

2. Pull the image:

   ```
   $ docker pull dhi.io/python:3.13
   ```

3. Run the image to confirm everything works:

   ```
   $ docker run --rm dhi.io/python:3.13 python -c "print('Hello from DHI')"
   ```

   This starts a container from the `python:3.13` image and runs a simple Python script that prints `Hello from DHI`.

To dive deeper into using images, see:

- Use a Docker Hardened Image for general usage
- Use a Helm chart for deploying with Helm

Step 3: Compare with other images

You can quickly compare DHIs with other images to see the security improvements and differences. This comparison helps you understand the value of using hardened images.

Run the following command to compare the Docker Hardened Image for Python with the non-hardened Docker Official Image for Python from Docker Hub. Look for the `## Overview` section in

the output for a summary comparison.

```
$ docker scout compare dhi.io/python:3.13 \
    --to python:3.13 \
    --platform linux/amd64 \
    --ignore-unchanged
```

The `## Overview` section of the output looks similar to the following:

```
  ## Overview

                      |                   Analyzed Image                    |
Comparison Image
--------------------+-----------------------------------------------------+--------------------
    Target            |  dhi.io/python:3.13                                 |  python:3.13
      digest          |  c215e9da9f84                                       |  7f48e892134c
      tag             |  3.13                                               |  3.13
      platform        | linux/amd64                                         | linux/amd64
      provenance      | https://github.com/docker-hardened-images/definitions |
https://github.com/docker-library/python.git
                      |  77a629b3d0db035700206c2a4e7ed904e5902ea8           |
3f2d7e4c339ab883455b81a873519f1d0f2cd80a
      vulnerabilities |    0C     0H     0M     0L                          |    0C
1H     5M   141L     2?
                      |           -1     -5   -141     -2                   |
      size            | 35 MB (-377 MB)                                     | 412 MB
      packages        | 80 (-530)                                           | 610
                      |                                                     |
```

> This is example output. Your results may vary depending on newly discovered CVEs and image updates.

Docker maintains near-zero CVEs in Docker Hardened Images. For DHI Select and Enterprise subscriptions, when new CVEs are discovered, the CVEs are remediated within the industry-leading SLA time frame. Learn more about the SLA-backed security features.

This comparison shows that the Docker Hardened Image:

- Removes vulnerabilities: 1 high, 5 medium, 141 low, and 2 unspecified severity CVEs removed
- Reduces size: From 412 MB down to 35 MB (91% reduction)
- Minimizes packages: From 610 packages down to 80 (87% reduction)

To dive deeper into comparing images see Search and evaluate Docker Hardened Images.

What's next

You've pulled and run your first Docker Hardened Image. Here are a few ways to keep going:

- Migrate existing applications to DHIs: Use Gordon to update your Dockerfiles to use Docker Hardened Images as the base.
- Start a trial to explore the benefits of a DHI subscription, such as access to FIPS and STIG variants, customized images, and SLA-backed updates.
- Get started with DHI Select and Enterprise: After subscribing to a DHI subscription or starting a trial, learn how to mirror repositories, customize images, and access compliance variants.
- Verify DHIs: Use tools like Docker Scout or Cosign to inspect and verify signed attestations, like SBOMs and provenance.
- Scan DHIs: Analyze the image with Docker Scout or other scanners to identify known CVEs.

□□□ #2
□ □□□ □□□ 2026-10-31 19:16:46 UTC
□ □□□ □□□ 2026-10-31 19:16:48 UTC