Get started with DHI Select and Enterprise

- Prerequisites
- Step 1: Find an image to use
- Step 2: Mirror the repository
- Step 3: Customize the image
- Step 4: Pull and run your customized image
- Step 5: Remove customization and stop mirroring
- What's next

{{< summary-bar feature_name="Docker Hardened Images" >}}

This guide shows you how to get started with DHI Select and Enterprise subscriptions. Unlike DHI Community, this workflow lets you mirror repositories to your organization namespace on Docker Hub, access compliance variants (FIPS), customize images, and get SLA-backed updates.

Prerequisites

To use this workflow, you need:

- Organization owner access in your Docker Hub namespace.
- One of the following:
  - A DHI Select or Enterprise subscription. Contact Docker sales to purchase DHI Enterprise or learn more about DHI plans.
  - An active DHI trial. Start a free DHI trial.
- Docker Desktop 4.65 or later to use the `docker dhi` CLI.

Each step, when applicable, shows Docker Hub and command line instructions. You can use either interface.

Step 1: Find an image to use

{{< tabs group="interface" >}} {{< tab name="Docker Hub" >}}

1. Go to Docker Hub and sign in.
2. Select your organization in the left sidebar.
3. Navigate to **Hardened Images** > **Catalog**.
4. Use the search bar or filters to find an image (for example, `python`, `node`, or `golang`). For this example, search for `python`.
   To search for an image with a compliance variant (FIPS or STIG), select **Compliance** and select the relevant compliance option.
5. Select the Python repository to view its details.
6. Select **Images** to view available image variants.

{{< /tab >}} {{< tab name="Command line" >}}

1. List available image repositories:

   ```
   $ docker dhi catalog list --type image
   ```

2. To filter by name and FIPS compliance, use the `--filter` and `--fips` flags:

   ```
   $ docker dhi catalog list --filter python --fips
   ```

3. Get image details for the repository:

   ```
   $ docker dhi catalog get python
   ```

{{< /tab >}} {{< /tabs >}}

Continue to the next step to mirror the image. To dive deeper into exploring images see Search and evaluate Docker Hardened Images.

Step 2: Mirror the repository

Mirroring copies a DHI repository into your organization namespace on Docker Hub. This lets you receive SLA-backed Docker security patches for your images and use customization as well as compliance variants. Only organization owners can mirror repositories.

{{< tabs group="interface" >}} {{< tab name="Docker Hub" >}}

1. In the image repository details page you found in the previous step, select **Use this image** > **Mirror repository**. Note that you must be signed in to Docker Hub to perform this action.
2. Select **Mirror**.
3. Wait for images to finish mirroring. This can take a few minutes.
4. Verify the mirrored repository appears in your organization namespace with a `dhi-` prefix (for example, `dhi-python`).

{{< /tab >}} {{< tab name="Command line" >}}

To use the following commands, you must authenticate or configure DHI CLI authentication using your Docker token. For details, see Use the DHI CLI.

1. Start mirroring the repository to your organization namespace. Replace `<your-org>` with your organization name.

   ```
   $ docker dhi mirror start --org <your-org> dhi/python,<your-org>/dhi-python
   ```

2. Wait for images to finish mirroring. This can take a few minutes.
3. Verify the mirrored repository. Replace `<your-org>` with your organization name.

   ```
   $ docker dhi mirror list --org <your-org>
   ```

{{< /tab >}} {{< /tabs >}}

Continue to the next step to customize the image. To dive deeper into mirroring images see Mirror a repository.

Step 3: Customize the image

One of the key benefits of DHI Select and Enterprise is the ability to customize your mirrored images. You can add system packages, configure settings, or make other modifications to meet your organization's specific requirements.

This example shows how to add the `curl` system package to your mirrored Python image.

{{< tabs group="interface" >}} {{< tab name="Docker Hub" >}}

1. Go to your organization namespace on Docker Hub.
2. Navigate to your mirrored repository (for example, `dhi-python`).
3. Select **Customizations**.
4. Select **Create customization**.
5. Search for `3-alpine3.23` and select any one of the images.

6. In **Add packages**, select **curl**.
7. Select **Next: Configure**.
8. In **Customization name**, enter a name for your customization (for example, `curl`).
9. Select **Next: Review customization**.
10. Select **Create customization** to start the build.

It can take a few minutes for the customization to build. Go to the **Customizations** tab of your mirrored repository and view the **Last build** column to monitor the build status.

{{< /tab >}} {{< tab name="Command line" >}}

To use the following commands, you must authenticate or configure DHI CLI authentication using your Docker token. For details, see Use the DHI CLI.

1. Create a customization. Replace `<your-org>` with your organization name. This creates a file called `my-customization.yaml` with the customization details.

   ```
   $ docker dhi customization prepare --org <your-org> python 3-alpine3.23 \
       --destination <your-org>/dhi-python \
       --name "python with curl" \
       > my-customization.yaml
   ```

2. Add the `curl` package to the customization. You can edit the file with any text or code editor. The following commands use `echo` to add the necessary lines to the YAML file:

   ```
   $ echo "contents:" >> my-customization.yaml
   $ echo "  packages:" >> my-customization.yaml
   $ echo "    - curl" >> my-customization.yaml
   ```

3. Apply the customization:

   ```
   $ docker dhi customization create --org <your-org> my-customization.yaml
   ```

4. Verify the customization was created:

   ```
   $ docker dhi customization list --org <your-org>
   ```

It can take a few minutes for the customization to build. To check the build status:

1. Go to your organization namespace on Docker Hub.
2. Navigate to your mirrored repository (for example, `dhi-python`).
3. Select **Customizations**.
4. View the **Last build** column to monitor the build status.

{{< /tab >}} {{< /tabs >}}

To dive deeper into customization, see Customize a Docker Hardened Image.

Step 4: Pull and run your customized image

After the customization build completes, you can pull and run the customized image from your organization namespace on Docker Hub.

1. Sign in to Docker Hub:

   ```
   $ docker login
   ```

2. Pull the customized image from your organization. Replace `<your-org>` with your organization name. The customized tag includes the suffix based on your customization name.

   ```
   $ docker pull <your-org>/dhi-python:3-alpine3.23_python-with-curl
   ```

3. Run the image and test that `curl` is installed:

   ```
   $ docker run --rm <your-org>/dhi-python:3-alpine3.23_python-with-curl curl --version
   ```

   This confirms that the `curl` package was successfully added to the image.

To dive deeper into using images, see:

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

Step 5: Remove customization and stop mirroring

To remove the customization and stop mirroring the repository:

1. Go to your organization namespace on Docker Hub.
2. Navigate to your mirrored repository (for example, `dhi-python`).
3. Select **Customizations**.
4. Find the customization you want to delete (for example, `python with curl`).

5. Select the trash can icon.
6. Select **Delete customization** to confirm the deletion.
7. To stop mirroring, go back to your organization's repositories list.
8. Find the mirrored repository (for example, `dhi-python`).
9. Select **Settings**.
10. Select **Stop mirroring**.
11. Select **Stop mirroring** to confirm.

What's next

You've mirrored, customized, and run a 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.
- 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 19:17:43 UTC
□ □□□ □□□ 2026-10-31 19:17:44 UTC