

Step-by-Step GitHub Releases Workaround: dockur example

When you can't use docker pull or the github workspace is blocked by your IT department.

Step-by-Step GitHub Releases Workaround

1. Create the Action Workflow File

1. Open your regular web browser on your Windows host.
2. Go to your GitHub profile and create a **New Repository**. Make it **Private** (e.g., `my-image-fetcher`).
3. Inside your repo, click on the **Actions** tab at the top.
4. Click the link that says "**set up a workflow yourself**".
5. Delete all default text in the editor and paste this configuration exactly:

yaml

```
name: Export Docker Image to Releases
on: [workflow_dispatch]

permissions:
  contents: write

jobs:
  bundle_image:
    runs-on: ubuntu-latest
    steps:
      - name: Pull and Save Image
        run: |
          docker pull ghcr.io/dockur/windows:latest
          docker save -o windows_image.tar ghcr.io/dockur/windows:latest

      - name: Create Private Release Asset
        uses: softprops/action-gh-release@v2
```

with:

tag_name: v1.0.0

name: "Windows Container Download"

files: windows_image.tar

env:

GITHUB_TOKEN: \${{ secrets.GITHUB_TOKEN }}

-
6. Click **Commit changes...** in the top right, then confirm it.

2. Trigger the Automated Build

1. Go back to the **Actions** tab.
2. On the left side, click "**Export Docker Image to Releases**".
3. On the right side, click the **Run workflow** dropdown, and then click the green **Run workflow** button.
4. Wait about **3 to 5 minutes**. GitHub's remote cloud servers will download the image and package it into a single `.tar` archive asset. A green checkmark will appear when it finishes.

3. Download the Tar via the Browser

1. Click the main logo of your repository to go back to its home page.
2. Look at the right-hand sidebar. Under the **Releases** section, you will see a fresh `v1.0.0` tag. Click on it.
3. Under the "Assets" heading, you will see `windows_image.tar`.
4. Left-click it. Your web browser will download it as a standard static file asset, bypassing the terminal blocks entirely.

4. Import the File into local Docker

Once your browser finishes downloading the archive to your Windows machine:

1. Drag or cut-paste `windows_image.tar` out of your Downloads folder and into `C:\PROJECTS\CobolSim\`.
2. Open your local WSL terminal and run:
bash

```
cd /mnt/c/PROJECTS/CobolSim
docker load -i windows_image.tar
```

-
3. Verify it is ready by typing `docker images`.
-
-

Revision #2

Created 25 June 2026 12:45:26 by Tim

Updated 25 June 2026 12:51:26 by Tim