Docker

Introduction

Dockerfile.jammy can be used to run Playwright scripts in Docker environment. These image includes all the dependencies needed to run browsers in a Docker container, and also include the browsers themselves.

Usage

This Docker image is published to Microsoft Artifact Registry.

> **INFO**
> This Docker image is intended to be used for testing and development purposes only. It is not recommended to use this Docker image to visit untrusted websites.

Pull the image

```
docker pull mcr.microsoft.com/playwright/python:v1.38.0-jammy
```

Run the image

By default, the Docker image will use the `root` user to run the browsers. This will disable the Chromium sandbox which is not available with root. If you run trusted code (e.g. End-to-end tests) and want to avoid the hassle of managing separate user then the root user may be fine. For web scraping or crawling, we recommend to create a separate user inside the Docker container and use the seccomp profile.

End-to-end tests

On trusted websites, you can avoid creating a separate user and use root for it since you trust the code which will run on the browsers.

```
docker run -it --rm --ipc=host mcr.microsoft.com/playwright/python:v1.38.0-jammy
/bin/bash
```

Crawling and scraping

On untrusted websites, it's recommended to use a separate user for launching the browsers in combination with the seccomp profile. Inside the container or if you are using the Docker image as a base image you have to use `adduser` for it.

```
docker run -it --rm --ipc=host --user pwuser --security-opt
seccomp=seccomp_profile.json mcr.microsoft.com/playwright/python:v1.38.0-jammy
/bin/bash
```

`seccomp_profile.json` is needed to run Chromium with sandbox. This is a default Docker seccomp profile with extra user namespace cloning permissions:

```
{
  "comment": "Allow create user namespaces",
  "names": [
    "clone",
    "setns",
    "unshare"
  ],
  "action": "SCMP_ACT_ALLOW",
  "args": [],
  "includes": {},
  "excludes": {}
}
```

> **NOTE**
>
> Using `--ipc=host` is recommended when using Chrome (Docker docs). Chrome can run out of memory without this flag.

Using on CI

See our Continuous Integration guides for sample configs.

Image tags

See all available image tags.

Docker images are published automatically by GitHub Actions. We currently publish images with the following tags (`v1.33.0` in this case is an example:):

- `:next` - tip-of-tree image version based on Ubuntu 22.04 LTS (Jammy Jellyfish).
- `:next-jammy` - tip-of-tree image version based on Ubuntu 22.04 LTS (Jammy Jellyfish).
- `:v1.33.0` - Playwright v1.33.0 release docker image based on Ubuntu 22.04 LTS (Jammy Jellyfish).
- `:v1.33.0-jammy` - Playwright v1.33.0 release docker image based on Ubuntu 22.04 LTS (Jammy Jellyfish).
- `:v1.33.0-focal` - Playwright v1.33.0 release docker image based on Ubuntu 20.04 LTS (Focal Fossa).
- `:sha-XXXXXXX` - docker image for every commit that changed docker files or browsers, marked with a short sha (first 7 digits of the SHA commit).

> **NOTE**
>
> It is recommended to always pin your Docker image to a specific version if possible. If the Playwright version in your Docker image does not match the version in your project/tests, Playwright will be unable to locate browser executables.

Base images

We currently publish images based on the following Ubuntu versions:

- **Ubuntu 22.04 LTS** (Jammy Jellyfish), image tags include `jammy`
- **Ubuntu 20.04 LTS** (Focal Fossa), image tags include `focal`

Alpine

Browser builds for Firefox and WebKit are built for the glibc library. Alpine Linux and other distributions that are based on the musl standard library are not supported.