Use a proxy server with the Docker CLI

- Configure the Docker client
  - Run containers with a proxy configuration
  - Build with a proxy configuration
  - Configure proxy settings per daemon
- Set proxy using the CLI
- Proxy as environment variable for builds

This page describes how to configure the Docker CLI to use proxies via environment variables in containers.

This page doesn't describe how to configure proxies for Docker Desktop. For instructions, see configuring Docker Desktop to use HTTP/HTTPS proxies.

If you're running Docker Engine without Docker Desktop, refer to Configure the Docker daemon to use a proxy to learn how to configure a proxy server for the Docker daemon (`dockerd`) itself.

If your container needs to use an HTTP, HTTPS, or FTP proxy server, you can configure it in different ways:

- Configure the Docker client
- Set proxy using the CLI

> Unfortunately, there's no standard that defines how web clients should handle proxy environment variables, or the format for defining them.
>
> If you're interested in the history of these variables, check out this blog post on the subject, by the GitLab team: We need to talk: Can we standardize NO_PROXY?.

Configure the Docker client

You can add proxy configurations for the Docker client using a JSON configuration file, located in `~/.docker/config.json`. Builds and containers use the configuration specified in this file.

```
{
 "proxies": {
   "default": {
     "httpProxy": "http://proxy.example.com:3128",
     "httpsProxy": "https://proxy.example.com:3129",
     "noProxy": "*.test.example.com,.example.org,127.0.0.0/8"
   }
 }
}
```

> Proxy settings may contain sensitive information. For example, some proxy servers require authentication information to be included in their URL, or their address may expose IP-addresses or hostnames of your company's environment.
>
> Environment variables are stored as plain text in the container's configuration, and as such can be inspected through the remote API or committed to an image when using `docker commit`.

The configuration becomes active after saving the file, you don't need to restart Docker. However, the configuration only applies to new containers and builds, and doesn't affect existing containers.

The following table describes the available configuration parameters.

Property	Description
`httpProxy`	Sets the `HTTP_PROXY` and `http_proxy` environment variables and build arguments.
`httpsProxy`	Sets the `HTTPS_PROXY` and `https_proxy` environment variables and build arguments.
`ftpProxy`	Sets the `FTP_PROXY` and `ftp_proxy` environment variables and build arguments.
`noProxy`	Sets the `NO_PROXY` and `no_proxy` environment variables and build arguments.
`allProxy`	Sets the `ALL_PROXY` and `all_proxy` environment variables and build arguments.

These settings are used to configure proxy environment variables for containers only, and not used as proxy settings for the Docker CLI or the Docker Engine itself. Refer to the environment variables and configure the Docker daemon to use a proxy server sections for configuring proxy settings for the CLI and daemon.

Run containers with a proxy configuration

When you start a container, its proxy-related environment variables are set to reflect your proxy configuration in `~/.docker/config.json`.

For example, assuming a proxy configuration like the example shown in the earlier section, environment variables for containers that you run are set as follows:

```
$ docker run --rm alpine sh -c 'env | grep -i  _PROXY'
https_proxy=http://proxy.example.com:3129
HTTPS_PROXY=http://proxy.example.com:3129
http_proxy=http://proxy.example.com:3128
HTTP_PROXY=http://proxy.example.com:3128
no_proxy=*.test.example.com,.example.org,127.0.0.0/8
NO_PROXY=*.test.example.com,.example.org,127.0.0.0/8
```

Build with a proxy configuration

When you invoke a build, proxy-related build arguments are pre-populated automatically, based on the proxy settings in your Docker client configuration file.

Assuming a proxy configuration like the example shown in the earlier section, environment are set as follows during builds:

```
$ docker build \
  --no-cache \
  --progress=plain \
  - <<EOF
FROM alpine
RUN env | grep -i _PROXY
EOF
```

```
#5 [2/2] RUN env | grep -i _PROXY
#5 0.100 HTTPS_PROXY=https://proxy.example.com:3129
```

```
#5 0.100 no_proxy=*.test.example.com,.example.org,127.0.0.0/8
#5 0.100 NO_PROXY=*.test.example.com,.example.org,127.0.0.0/8
#5 0.100 https_proxy=https://proxy.example.com:3129
#5 0.100 http_proxy=http://proxy.example.com:3128
#5 0.100 HTTP_PROXY=http://proxy.example.com:3128
#5 DONE 0.1s
```

Configure proxy settings per daemon

The `default` key under `proxies` in `~/.docker/config.json` configures the proxy settings for all daemons that the client connects to. To configure the proxies for individual daemons, use the address of the daemon instead of the `default` key.

The following example configures both a default proxy config, and a no-proxy override for the Docker daemon on address `tcp://docker-daemon1.example.com`:

```
{
 "proxies": {
   "default": {
     "httpProxy": "http://proxy.example.com:3128",
     "httpsProxy": "https://proxy.example.com:3129",
     "noProxy": "*.test.example.com,.example.org,127.0.0.0/8"
   },
   "tcp://docker-daemon1.example.com": {
     "noProxy": "*.internal.example.net"
   }
 }
}
```

Set proxy using the CLI

Instead of configuring the Docker client, you can specify proxy configurations on the command-line when you invoke the `docker build` and `docker run` commands.

Proxy configuration on the command-line uses the `--build-arg` flag for builds, and the `--env` flag for when you want to run containers with a proxy.

```
$ docker build --build-arg HTTP_PROXY="http://proxy.example.com:3128" .
$ docker run --env HTTP_PROXY="http://proxy.example.com:3128" redis
```

For a list of all the proxy-related build arguments that you can use with the `docker build` command, see Predefined ARGs. These proxy values are only available in the build container. They're not included in the build output.

Proxy as environment variable for builds

Don't use the `ENV` Dockerfile instruction to specify proxy settings for builds. Use build arguments instead.

Using environment variables for proxies embeds the configuration into the image. If the proxy is an internal proxy, it might not be accessible for containers created from that image.

Embedding proxy settings in images also poses a security risk, as the values may include sensitive information.

□□□ #2
□ □□□ □□□ 2026-10-31 19:21:43 UTC
□ □□□ □□□ 2026-10-31 19:21:44 UTC