Docker Hub pull usage and limits

- Pull definition
- Pull attribution
  - Authentication
    - Docker Desktop
    - Docker Engine
    - Docker Swarm
    - GitHub Actions
    - Kubernetes
    - Third-party platforms
- View monthly pulls and included usage
- View pull rate and limit

Unauthenticated and Docker Personal users are subject to a 6-hour pull rate limit on Docker Hub. In contrast, Docker Pro, Team, and Business users benefit from an unlimited pull rate.

The following pull usage and limits apply based on your subscription, subject to fair use:

User type	Pull rate limit per 6 hours
Business (authenticated)	Unlimited
Team (authenticated)	Unlimited
Pro (authenticated)	Unlimited
Personal (authenticated)	200
Unauthenticated Users	100 per IPv4 address or IPv6 /64 subnet

Pull definition

A pull is defined as the following:

- A Docker pull includes both a version check and any download that occurs as a result of the pull. Depending on the client, a `docker pull` can verify the existence of an image or tag without downloading it by performing a version check.
- Version checks do not count towards usage pricing.
- A pull for a normal image makes one pull for a single manifest.
- A pull for a multi-arch image will count as one pull for each different architecture.

Pull attribution

Pulls from authenticated users can be attributed to either a personal or an organization namespace .

Attribution is based on the following:

- Private pulls: Pulls for private repositories are attributed to the repository's namespace owner.
- Public pulls: When pulling images from a public repository, attribution is determined based on domain affiliation and organization membership.
- Verified domain ownership: When pulling an image from an account linked to a verified domain, the attribution is set to be the owner of that domain.
- Single organization membership:
  - If the owner of the verified domain is a company and the user is part of only one organization within that company, the pull is attributed to that specific organization.
  - If the user is part of only one organization, the pull is attributed to that specific organization.
- Multiple organization memberships: If the user is part of multiple organizations under the company, the pull is attributed to the user's personal namespace.

Authentication

To ensure correct attribution of your pulls, you must authenticate with Docker Hub. The following sections provide information on how to sign in to Docker Hub to authenticate your pulls.

Docker Desktop

If you are using Docker Desktop, you can sign in to Docker Hub from the Docker Desktop menu.

Select **Sign in / Create Docker ID** from the Docker Desktop menu and follow the on-screen instructions to complete the sign-in process.

Docker Engine

If you're using a standalone version of Docker Engine, run the `docker login` command from a terminal to authenticate with Docker Hub. For information on how to use the command, see docker login.

Docker Swarm

If you're running Docker Swarm, you must use the `--with-registry-auth` flag to authenticate with Docker Hub. For more information, see Create a service. If you are using a Docker Compose file to deploy an application stack, see docker stack deploy.

GitHub Actions

If you're using GitHub Actions to build and push Docker images to Docker Hub, see login action. If you are using another Action, you must add your username and access token in a similar way for authentication.

Kubernetes

If you're running Kubernetes, follow the instructions in Pull an Image from a Private Registry for information on authentication.

Third-party platforms

If you're using any third-party platforms, follow your provider's instructions on using registry authentication.

> When pulling images via a third-party platform, the platform may use the same IPv4 address or IPv6 /64 subnet to pull images for multiple users. Even if you are authenticated, pulls attributed to a single IPv4 address or IPv6 /64 subnet may cause abuse rate limiting.

- Artifactory
- AWS CodeBuild
- AWS ECS/Fargate
- Azure Pipelines
- Chipper CI
- CircleCI
- Codefresh

- Drone.io
- GitLab
- LayerCI
- TeamCity

View monthly pulls and included usage

You can view your monthly pulls on the Usage page in Docker Hub.

On that page, you can also send a report to your email that contains a comma separated file with the following detailed information.

CSV column	Definition	Usage guidance
`datehour`	The date and hour (`yyyy/mm/dd/hh`) of the pull that resulted in the data transfer.	This helps in identifying peak usage times and patterns.
`user_name`	The Docker ID of the user that pulled the image	This lets organization owners track data consumption per user and manage resources effectively.
`repository`	The name of the repository of the image that was pulled.	This lets you identify which repositories are most frequently accessed and consume most of the data transfer.
`access_token_name`	Name of the access token that was used for authentication with Docker CLI. `generated` tokens are automatically generated by the Docker client when a user signs in.	Personal access tokens are usually used to authenticate automated tools (Docker Desktop, CI/CD tools, etc.). This is useful for identifying which automated system issued the pull.
`ips`	The IP address that was used to pull the image. This field is aggregated, so more than one IP address may appear, representing all the IPs used to pull an image within the same date and hour.	This helps you understand the origin of the data transfer, which is useful for diagnosing and identifying patterns in automated or manual pulls.
`repository_privacy`	The privacy state of the image repository that was pulled. This can either be `public` or `private`.	This distinguishes between public and private repositories to identify which data transfer threshold the pull impacts.

CSV column	Definition	Usage guidance
`tag`	The tag for the image. The tag is only available if the pull included a tag.	This helps in identifying the image. Tags are often used to identify specific versions or variants of an image.
`digest`	The unique image digest for the image.	This helps in identifying the image.
`version_checks`	The number of version checks accumulated for the date and hour of each image repository. Depending on the client, a pull can do a version check to verify the existence of an image or tag without downloading it.	This helps identify the frequency of version checks, which you can use to analyze usage trends and potential unexpected behaviors.
`pulls`	The number of pulls accumulated for the date and hour of each image repository.	This helps identify the frequency of repository pulls, which you can use to analyze usage trends and potential unexpected behaviors.

View pull rate and limit

The pull rate limit is calculated on a 6-hour basis. There is no pull rate limit for users or automated systems with a paid subscription. Unauthenticated and Docker Personal users using Docker Hub will experience rate limits on image pulls.

When you issue a pull and you are over the limit, Docker Hub returns a `429` response code with the following body when the manifest is requested:

```
You have reached your pull rate limit. You may increase the limit by authenticating and upgrading: https://www.docker.com/increase-rate-limits
```

This error message appears in the Docker CLI or in the Docker Engine logs.

To view your current pull rate and limit:

> To check your limits, you need `curl` and `jq` installed.

1. Get a token.
   - To get a token anonymously, if you are pulling anonymously:

     ```
     $ TOKEN=$(curl "https://auth.docker.io/token?service=registry.docker.io&scope=repository:ratelimitpreview/test:pull" | jq -r .token)
     ```

- To get a token with a user account, if you are authenticated, insert your username and password in the following command:

  ```
  $ TOKEN=$(curl --user 'username:password'
  "https://auth.docker.io/token?service=registry.docker.io&scope=repository:rateli
  mitpreview/test:pull" | jq -r .token)
  ```

2. Get the headers that contain your limits. These headers are returned on both GET and HEAD requests. Using GET emulates a real pull and counts towards the limit. Using HEAD won't.

   ```
   $ curl --head -H "Authorization: Bearer $TOKEN" https://registry-
   1.docker.io/v2/ratelimitpreview/test/manifests/latest
   ```

3. Examine the headers. You should see the following headers.

   ```
   ratelimit-limit: 100;w=21600
   ratelimit-remaining: 20;w=21600
   docker-ratelimit-source: 192.0.2.1
   ```

   In the previous example, the pull limit is 100 pulls per 21600 seconds (6 hours), and there are 20 pulls remaining.
   If you don't see any `ratelimit` header, it could be because the image or your IP is unlimited in partnership with a publisher, provider, or an open source organization. It could also mean that the user you are pulling as is part of a paid Docker subscription. Pulling that image won't count toward pull rate limits if you don't see these headers.

□□□ #2
□ □□□ □□□ 2026-10-31 19:21:25 UTC
□ □□□ □□□ 2026-10-31 19:21:27 UTC