Registry Access Management

- Supported registries
- Prerequisites
- Configure registry permissions
- Verify restrictions are working
- Registry limits and platform constraints
- Build and deployment restrictions
- Security bypass considerations
- Registry allowlist best practices

{{< summary-bar feature_name="Registry access management" >}}

Registry Access Management (RAM) lets administrators control which container registries developers can access through Docker Desktop. This DNS-level filtering limits pulls and pushes to approved registries.

RAM works with cloud registries, on-premises registries, and registry mirrors. You can allow any hostname or domain, but you must also include redirect domains (for example, `s3.amazonaws.com` for some registries) in your allowlist.

Supported registries

Registry Access Management works with any container registry, including:

- Docker Hub (allowed by default)
- Cloud registries: Amazon ECR, Google Artifact Registry, Azure Container Registry
- Git-based registries: GitHub Container Registry, GitLab Container Registry
- On-premises solutions: Nexus, Artifactory, Harbor
- Registry mirrors: Including Docker Hub mirrors

Prerequisites

Before configuring Registry Access Management, you must:

- Enforce sign-in. Registry Access Management only takes effect when users are signed in to Docker Desktop with organization credentials.
- Use Organization access tokens (OATs) for authentication
- Have a Docker Business subscription

Configure registry permissions

1. Sign in to Docker Home and select your organization from the top-left account drop-down.
2. Select **Docker Desktop**, then **Registry Access**.
3. Use the toggle to turn on registry access. By default, Docker Hub is enabled in the registry list.
4. To add registries, select **Add registry** and provide a **Registry address** and **Registry nickname**.
5. Select **Create**. You can add up to 100 registries.
6. Verify your registry appears in the registry list and select **Save changes**.

> Policy changes can take up to 24 hours to propagate. To apply changes immediately, ask developers to sign out and back in to Docker Desktop.

If a developer belongs to multiple organizations with different RAM policies, only the policy for the first organization in the configuration file is enforced.

> RAM restrictions also apply to Dockerfile `ADD` instructions that fetch content by URL. Include trusted registry domains in your allowlist when using `ADD` with URLs.
>
> RAM is designed for container registries, not general-purpose URLs such as package mirrors or storage services. Adding too many domains may cause errors or hit system limits.

Verify restrictions are working

After users sign in to Docker Desktop with their organization credentials, Registry Access Management takes effect immediately.

When users try to pull from a blocked registry:

```
$ docker pull blocked-registry.com/image:tag
Error response from daemon: registry access to blocked-registry.com is not allowed
```

Allowed registry access works normally:

```
$ docker pull allowed-registry.com/image:tag
# Pull succeeds
```

Registry restrictions apply to all Docker operations including pulls, pushes, and builds that reference external registries.

Registry limits and platform constraints

Registry Access Management has these limits and platform-specific behaviors:

- Maximum allowlist size: 100 registries or domains per organization
- DNS-based filtering: Restrictions work at the hostname level, not IP addresses
- Redirect domains required: Include all domains a registry redirects to (CDN endpoints, storage services)
- Windows containers: Windows image operations aren't restricted by default. Turn on **Use proxy for Windows Docker daemon** in Docker Desktop settings to apply restrictions
- WSL 2 requirements: Requires Linux kernel 5.4 or later. Restrictions apply to all WSL 2 distributions

Build and deployment restrictions

These scenarios aren't restricted by Registry Access Management:

- Docker Buildx with the Kubernetes driver
- Docker Buildx with a custom Docker-container driver
- Some Docker Debug and Kubernetes image pulls (even if Docker Hub is blocked)

- Images previously cached by registry mirrors may still be blocked if the source registry is restricted

Security bypass considerations

Users can potentially bypass Registry Access Management through:

- Local proxies or DNS manipulation
- Signing out of Docker Desktop (unless sign-in is enforced)
- Network-level modifications outside Docker Desktop's control

To maximize security effectiveness:

- Enforce sign-in to prevent bypass through sign-out
- Implement additional network-level controls for complete protection
- Use Registry Access Management as part of a broader security strategy

Registry allowlist best practices

- Include all registry domains. Some registries redirect to multiple domains. For AWS ECR, include:

  ```
  your-account.dkr.ecr.us-west-2.amazonaws.com
  amazonaws.com
  s3.amazonaws.com
  ```

- Maintain the allowlist regularly:
  - Remove unused registries periodically
  - Add newly approved registries as needed
  - Update domain names that may have changed
  - Monitor registry usage through Docker Desktop analytics
- Test configuration changes:
  - Verify registry access after allowlist updates
  - Confirm all necessary redirect domains are included
  - Ensure development workflows aren't disrupted
- Combine with Enhanced Container Isolation for broader Desktop security

---

□□□ #2
□ □□□ □□□ 2026-10-31 19:10:28 UTC
□ □□□ □□□ 2026-10-31 19:10:29 UTC