Docker Registry Privado

Arturo Silvelo

Try New Roads

¿Qué es un Docker Registry?

- Un **Docker Registry** es un servicio para almacenar y distribuir imágenes de contenedores.
- El más conocido es **Docker Hub** (registro público por defecto de Docker).

Docker Hub (registro público)

- Permite compartir imágenes con la comunidad o equipos de trabajo.
- Muchas imágenes oficiales y de terceros están disponibles públicamente.
- Ejemplo de uso:

```
docker pull nginx
docker push miusuario/miproyecto:latest
```

Registro privado: ¿qué es y para qué sirve?

- Puedes gestionar tu propio registro privado, controlando el acceso y la distribución de imágenes.
- Un registro privado se puede montar fácilmente en un contenedor usando la imagen oficial `registry`.

Ventajas y desventajas de un registro privado

Ventajas:

- Control total sobre tus imágenes y su acceso.
- Sin límites de subida/descarga impuestos por terceros.
- Privacidad y seguridad para imágenes internas o sensibles.
- Integración directa con CI/CD y despliegues internos.

Desventajas:

- Requiere administración y mantenimiento.
- Debes encargarte de la seguridad (TLS, autenticación).
- Consumo de recursos y almacenamiento en tu infraestructura.

Gestión de un Docker Registry privado

¿Qué gestiones se pueden hacer?

- Subir (push) imágenes al registry.
- Descargar (pull) imágenes del registry.
- Eliminar imágenes antiguas o no deseadas.
- Limpiar espacio (garbage collection).
- Consultar el catálogo de imágenes almacenadas.
- Proteger el acceso (autenticación, scopes y TLS).

¿Cómo hacer dichas gestiones?

Subir una imagen:

```
docker tag miimagen:latest localhost:5000/miimagen:latest
docker push localhost:5000/miimagen:latest
```

Descargar una imagen:

```
docker pull localhost:5000/miimagen:latest
```

Eliminar una imagen:

- Habilita el borrado añadiendo `REGISTRY_STORAGE_DELETE_ENABLED=true` al contenedor.
- Borra el manifiesto vía API:

```
curl -X DELETE http://localhost:5000/v2/miimagen/manifests/<digest>
```

- Ejecuta el garbage collector:

```
docker exec registry registry garbage-collect /etc/docker/registry/config.yml
```

Consultar el catálogo de imágenes:

```
curl http://localhost:5000/v2/_catalog
```

Scopes y namespaces en los nombres de imágenes

- Puedes organizar imágenes usando "scopes" o namespaces, que funcionan como carpetas o prefijos.
- El nombre completo de una imagen puede incluir uno o varios niveles de scope:
  - Ejemplo:

```
mi-registry.local:5000/miempresa/app-backend:1.0
mi-registry.local:5000/devops/monitoring/prometheus:latest
```

- Esto permite separar proyectos, equipos o entornos dentro del mismo registry.

Ventajas de usar scopes/namespaces:

- Organización clara de imágenes por proyecto, equipo o entorno.
- Control de acceso más granular (si se combina con autenticación).
- Evita colisiones de nombres entre diferentes aplicaciones o equipos.

Cómo usar scopes al subir imágenes:

```
docker tag miimagen:latest localhost:5000/miempresa/miimagen:latest
docker push localhost:5000/miempresa/miimagen:latest
```

- Puedes anidar tantos niveles como necesites:

  `localhost:5000/area/proyecto/servicio:tag`

Herramientas gráficas para gestión

- **Portus**: Interfaz web avanzada para Docker Registry, con gestión de usuarios, equipos y políticas.
- **Harbor**: Solución completa con UI, autenticación, escaneo de vulnerabilidades y replicación.
- **Docker Registry UI**: Herramienta ligera para explorar y borrar imágenes desde el navegador.