Test an insecure registry

Estimated reading time: 3 minutes

While it's highly recommended to secure your registry using a TLS certificate issued by a known CA, you may alternatively decide to use self-signed certificates, or even use your registry over plain http.

You have to understand the downsides in doing so, and the extra burden in configuration.

Deploying a plain HTTP registry

> **Warning**: it's not possible to use an insecure registry with basic authentication.

This basically tells Docker to entirely disregard security for your registry. While this is relatively easy to configure the daemon in this way, it is **very** insecure. It does expose your registry to trivial MITM. Only use this solution for isolated testing or in a tightly controlled, air-gapped environment.

1. Open the `/etc/default/docker` file or `/etc/sysconfig/docker` for editing.

   Depending on your operating system, your Engine daemon start options.

2. Edit (or add) the `DOCKER_OPTS` line and add the `--insecure-registry` flag.

   This flag takes the URL of your registry, for example.

   ```
   DOCKER_OPTS="--insecure-registry myregistrydomain.com:5000"
   ```

3. Close and save the configuration file.

4. Restart your Docker daemon

   The command you use to restart the daemon depends on your operating system. For example, on Ubuntu, this is usually the `service docker stop` and `service docker start` command.

5. Repeat this configuration on every Engine host that wants to access your registry.

Using self-signed certificates

> **Warning**: using this along with basic authentication requires to **also** trust the certificate into the OS cert store for some versions of docker (see below)

This is more secure than the insecure registry solution. You must configure every docker daemon that wants to access your registry

1. Generate your own certificate:

   ```
   mkdir -p certs && openssl req \
     -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key \
     -x509 -days 365 -out certs/domain.crt
   ```

2. Be sure to use the name `myregistrydomain.com` as a CN.

3. Use the result to start your registry with TLS enabled

4. Instruct every docker daemon to trust that certificate.

   This is done by copying the `domain.crt` file to `/etc/docker/certs.d/myregistrydomain.com:5000/ca.crt`.

5. Don't forget to restart the Engine daemon.

Troubleshooting insecure registry

This sections lists some common failures and how to recover from them.

Failing...

Failing to configure the Engine daemon and trying to pull from a registry that is not using TLS will results in the following message:

```
FATA[0000] Error response from daemon: v1 ping attempt failed with err
Get https://myregistrydomain.com:5000/v1/_ping: tls: oversized record
If this private registry supports only HTTP or HTTPS with an unknown C
`--insecure-registry myregistrydomain.com:5000` to the daemon's argume
In the case of HTTPS, if you have access to the registry's CA certific
simply place the CA certificate at /etc/docker/certs.d/myregistrydomai
```

Docker still complains about the certificate when using authentication?

When using authentication, some versions of docker also require you to trust the certificate at the OS level. Usually, on Ubuntu this is done with:

```
$ cp certs/domain.crt /usr/local/share/ca-certificates/myregistrydomai
update-ca-certificates
```

... and on Red Hat (and its derivatives) with:

```
cp certs/domain.crt /etc/pki/ca-trust/source/anchors/myregistrydomain.
update-ca-trust
```

... On some distributions, e.g. Oracle Linux 6, the Shared System Certificates feature needs to be manually enabled:

```
$ update-ca-trust enable
```

Now restart docker (`service docker stop && service docker start`, or any other way you use to restart docker).

Feedback? Suggestions? Can't find something in the docs?
Edit this page ● Request docs changes ● Get support
Rate this page: