Manage swarm security with public key infrastructure (PKI)

Estimated reading time: 4 minutes

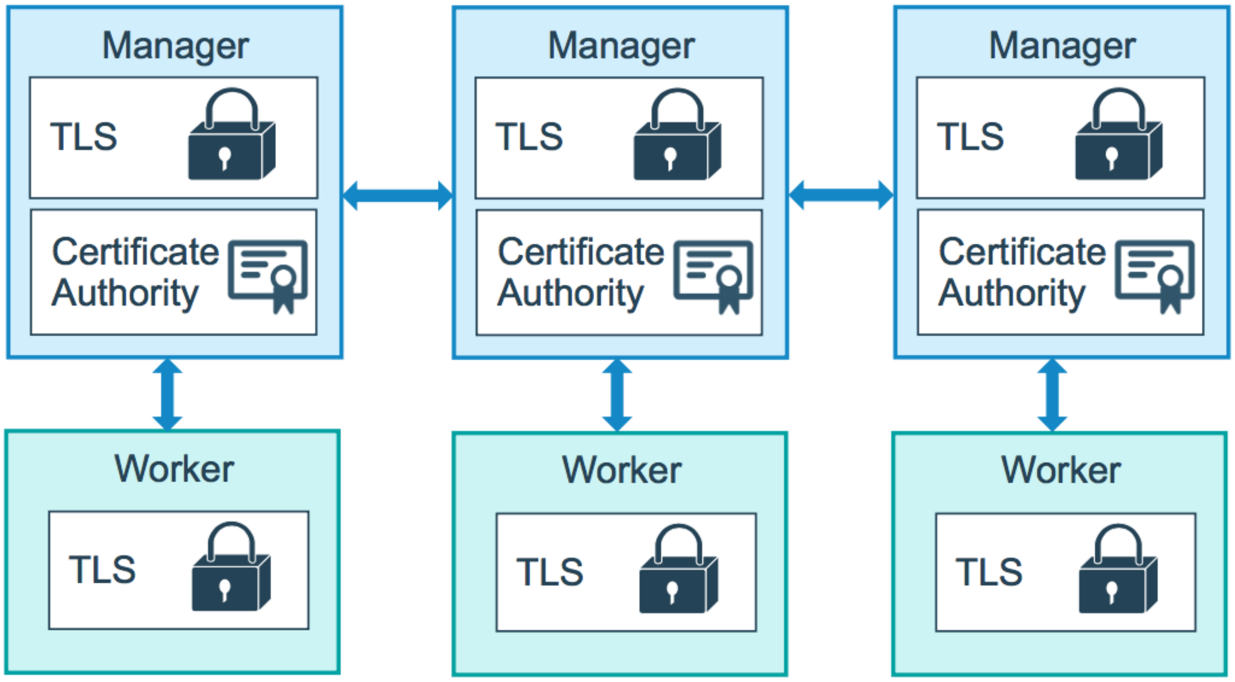
The swarm mode public key infrastructure (PKI) system built into Docker makes it simple to securely deploy a container orchestration system. The nodes in a swarm use mutual Transport Layer Security (TLS) to authenticate, authorize, and encrypt the communications with other nodes in the swarm.

When you create a swarm by running `docker swarm init`, Docker designates itself as a manager node. By default, the manager node generates a new root Certificate Authority (CA) along with a key pair, which are used to secure communications with other nodes that join the swarm. If you prefer, you can specify your own externally-generated root CA, using the `--external-ca` flag of the docker swarm init (https://docs.docker.com/engine/reference/commandline/swarm_init/) command.

The manager node also generates two tokens to use when you join additional nodes to the swarm: one **worker token** and one **manager token**. Each token includes the digest of the root CA's certificate and a randomly generated secret. When a node joins the swarm, the joining node uses the digest to validate the root CA certificate from the remote manager. The remote manager uses the secret to ensure the joining node is an approved node.

Each time a new node joins the swarm, the manager issues a certificate to the node. The certificate contains a randomly generated node ID to identify the node under the certificate common name (CN) and the role under the organizational unit (OU). The node ID serves as the cryptographically secure node identity for the lifetime of the node in the current swarm.

The diagram below illustrates how manager nodes and worker nodes encrypt communications using a minimum of TLS 1.2.



The example below shows the information from a certificate from a worker node:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            3b:1c:06:91:73:fb:16:ff:69:c3:f7:a2:fe:96:c1:73:e2:80:97:3
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN=swarm-ca
        Validity
            Not Before: Aug 30 02:39:00 2016 GMT
            Not After : Nov 28 03:39:00 2016 GMT
        Subject: O=ec2adilxf4ngv7ev8fwsi61i7, OU=swarm-worker, CN=dw02
...snip...
```

By default, each node in the swarm renews its certificate every three months. You can configure this interval by running the `docker swarm update --cert-expiry <TIME PERIOD>` command. The minimum rotation value is 1 hour. Refer to the docker swarm update (https://docs.docker.com/engine/reference/commandline/swarm_update/) CLI reference for details.

Rotating the CA certificate

In the event that a cluster CA key or a manager node is compromised, you can rotate the swarm root CA so that none of the nodes trust certificates signed by the old root CA anymore.

Run `docker swarm ca --rotate` to generate a new CA certificate and key. If you prefer, you can pass the `--ca-cert` and `--external-ca` flags to specify the root certificate and to use a root CA external to the swarm. Alternately, you can pass the `--ca-cert` and `--ca-key` flags to specify the exact certificate and key you would like the swarm to use.

When you issue the `docker swarm ca --rotate` command, the following things happen in sequence:

1. Docker generates a cross-signed certificate. This means that a version of the new root CA certificate is signed with the old root CA certificate. This cross-signed certificate is used as an intermediate certificate for all new node certificates. This ensures that nodes that still trust the old root CA can still validate a certificate signed by the new CA.

2. In Docker 17.06 and higher, Docker also tells all nodes to immediately renew their TLS certificates. This process may take several minutes, depending on the number of nodes in the swarm.

   > **Note: If your swarm has nodes with different Docker versions, the following two things are true:**
   >
   > - Only a manager that is running as the leader **and** running Docker 17.06 or higher tells nodes to renew their TLS certificates.
   > - Only nodes running Docker 17.06 or higher obey this directive.
   >
   > For the most predictable behavior, ensure that all swarm nodes are running Docker 17.06 or higher.

3. After every node in the swarm has a new TLS certificate signed by the new CA, Docker forgets about the old CA certificate and key material, and tells all the nodes to trust the new CA certificate only.

This also causes a change in the swarm’s join tokens. The previous join tokens are no longer valid.

From this point on, all new node certificates issued are signed with the new root CA, and do not contain any intermediates.

Learn More

- Read about how nodes (https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/) work.
- Learn how swarm mode services (https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/) work.

swarm (https://docs.docker.com/glossary/?term=swarm), security (https://docs.docker.com/glossary/?term=security), tls (https://docs.docker.com/glossary/?term=tls), pki (https://docs.docker.com/glossary/?term=pki)