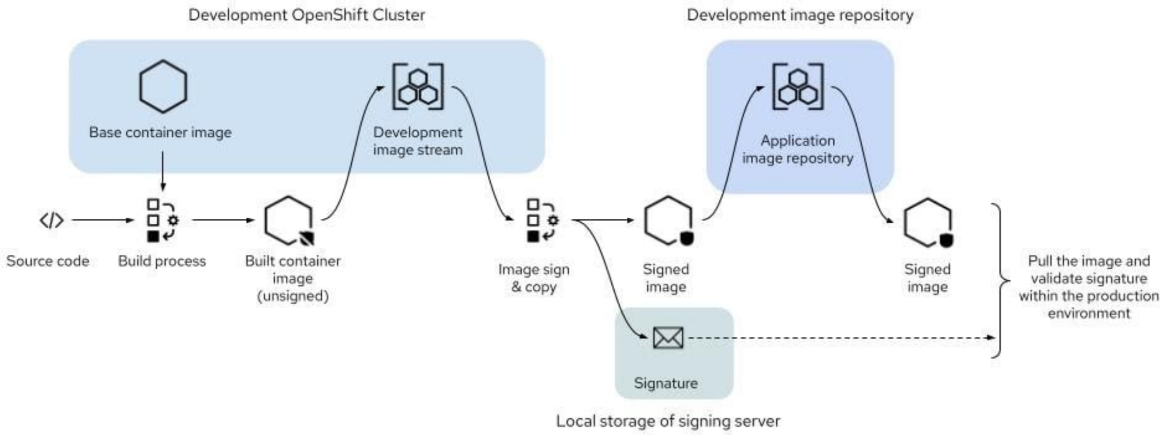
Signing and Verifying Docker Images with Notation and Harbor

This guide will walk you through the process of using Notation for signing and verifying Docker images in Harbor. You will learn how to install Notation, set up keys and certificates, sign Docker images, and verify the signatures.

Explanation of Simple Terms

Here's a concise explanation of the terms for a new developer:

1. **Docker Images**: Pre-configured packages containing everything needed to run a software application, including code, runtime, libraries, and system settings.
2. **Notation**: A tool for signing and verifying the authenticity of Docker images, ensuring their integrity and that they haven't been tampered with.
3. **Harbor**: A cloud-native registry for storing and managing Docker images and Helm charts, with Notation integration for image signing and verification.
4. **Signing Docker Images**: The process of creating a digital signature for a Docker image to prove it's from a trusted source and hasn't been altered.
5. **Verifying Signatures**: Checking the digital signature of a Docker image to confirm its authenticity and integrity.
6. **Keys and Certificates**: Cryptographic elements used for signing (private key) and verifying (public key) Docker images. Certificates confirm the identity of the entity owning the private key.


Development OpenShift Cluster
Development image repository
Base container image
Development image stream
Application image repository
Source code
Build process
Built container image (unsigned)
Image sign & copy
Signed image
Signed image
Pull the image and validate signature within the production environment
Signature
Local storage of signing server

Significance

Signing and verifying Docker images with Notation and Harbor is important because:

1. **Security**: Ensures that Docker images haven't been tampered with and come from a trusted source.
2. **Trust**: Confirms that the image is exactly what the creator intended.
3. **Integrity**: Verifies that the image hasn't been altered since its creation.
4. **Compliance**: Helps meet industry security requirements.
5. **Traceability**: Tracks the origin of an image, aiding in troubleshooting.

What You Need

Before starting this tutorial, make sure you have:

- **Docker Engine**: Software for creating and running containers, required to connect with Harbor.
- **Docker Compose**: A tool for setting up and managing Harbor, a container registry for storing and sharing Docker images.

Prerequisites

Before setting up Harbor, ensure Docker Engine and Docker Compose are installed. Docker Engine is essential for creating and managing containers, while Docker Compose simplifies Harbor setup by defining necessary containers and services in a YAML file, streamlining the orchestration process.

Installing Harbor and Configuring Domain Name

Harbor supports both HTTP and HTTPS domains. For security, this guide focuses on setting up an HTTPS domain.

Step 1: Prepare Your Domain Name

- **Domain Name**: Use a registered domain name. Example: `myharbor-registry.online`.
- **DNS Configuration**: Point a DNS record to the IPv4 address of your Harbor server for proper domain resolution.

Step 2: Install Certbot for SSL Certificates

Certbot, a tool from Let’s Encrypt, is used to obtain an SSL certificate:

- **Install Certbot**:

  ```
  sudo apt install --classic certbot
  ```

- **Obtain SSL Certificate**:

  ```
  sudo certbot certonly --standalone -d myharbor-registry.online
  ```

  Certbot provides the file paths for the certificate and key files. Note these paths for future configuration.

Downloading and Installing Harbor

Install Harbor on an Ubuntu instance using Docker and Docker-Compose:

Step 1: Download the Harbor Release Package

```
wget https://github.com/goharbor/harbor/releases/download/v2.9.1/harbor-online-
installer-v2.9.1.tgz
tar -xvf harbor-online-installer-v2.9.1.tgz
cd harbor
```

Step 2: Configure Harbor Settings

1. **Copy the Template Configuration**:

   ```
   cp harbor.yml.tmpl harbor.yml
   ```

2. **Update Harbor Domain**:
   - Open `harbor.yml` and set the hostname to your domain:

     ```
     hostname: myharbor-registry.online
     ```

3. **Configure HTTPS with SSL Certificates**:
   - Update the paths to the SSL certificate and key files:

     ```
     https:
       certificate: /path/to/certificate.crt
       private_key: /path/to/private.key
     ```

Step 3: Run the Installation Script

Execute the installation script:

```
sudo ./install.sh
```

Replace `myharbor-registry.online` in the configuration file with your domain name. Update the paths for the SSL certificate and private key under the HTTPS configuration. After execution, you can access Harbor at `https://myharbor-registry.online`.

Install Notation CLI

Install the Notation CLI on Linux:

- **For Brew**:

  ```
  brew install notation
  ```

- **For x86_64 Processors**:

  ```
  export NOTATION_VERSION=1.0.1
  curl -LO
  https://github.com/notaryproject/notation/releases/download/v$NOTATION_VER
  SION/notation_$NOTATION_VERSION\_linux_amd64.tar.gz
  curl -LO
  https://github.com/notaryproject/notation/releases/download/v$NOTATION_VER
  SION/notation_$NOTATION_VERSION\_checksums.txt
  shasum --check notation_$NOTATION_VERSION\_checksums.txt
  sudo tar xvzf notation_$NOTATION_VERSION\_linux_amd64.tar.gz -C /usr/bin/
  ```

Build and Push an Image to Harbor

Step 1: Build the Image

```
docker build -t myharbor-registry.online/notation-project/net-monitor:v1
https://github.com/wabbit-networks/net-monitor.git#main
```

Step 2: Push the Image to Harbor

```
docker login myharbor-registry.online
docker push myharbor-registry.online/notation-project/net-monitor:v1
```

After pushing, the “Signed” section in Harbor will indicate if the artifact is unsigned.

Signing and Verifying Docker Images

Step 1: Generate a Test Key and Self-Signed Certificate

```
notation cert generate-test --default "wabbit-networks.io"
notation key ls
notation cert ls
```

Step 2: Sign the Image

```
notation sign $IMAGE
```

Set the image digest as an environment variable:

```
export IMAGE=myharbor-registry.online/notation-project/net-
monitor@sha256:<digest>
```

Once signed, the status in Harbor will show a green tick.

Step 3: Create a Trust Policy

Save the following JSON as `trustpolicy.json`:

```
{
    "version": "1.0",
    "trustPolicies": [
        {
            "name": "wabbit-networks-images",
            "registryScopes": [ "*" ],
            "signatureVerification": {
                "level" : "strict"
            },
            "trustStores": [ "ca:wabbit-networks.io" ],
            "trustedIdentities": [
                "*"
            ]
        }
    ]
}
```

Import the trust policy:

```
notation policy import ./trustpolicy.json
```

Step 4: Verify the Image

```
notation verify $IMAGE
```

Inspect the signature and certificate information:

```
notation inspect $IMAGE
```

Conclusion

This guide covered:

1. Setting up SSL certificates for Harbor.
2. Installing Notation CLI.
3. Signing and verifying container images.
4. Creating a trust policy to ensure image integrity and authenticity in your software supply chain.