Monday, 3 February 2025

Docker image signature

🔐 Signing Your Own Docker Images for Security

✅ **Signing images** ensures they come from a trusted source and haven't been modified. We’ll use **Cosign** (part of Sigstore) to sign and verify a Docker image.

🔷 Step 1: Install Cosign

Install **Cosign**, a tool for signing and verifying container images.

For Linux/macOS:

```
curl -LO https://github.com/sigstore/cosign/releases/latest/download/cosign-linux-amd64
chmod +x cosign-linux-amd64
sudo mv cosign-linux-amd64 /usr/local/bin/cosign
```

For Windows (PowerShell):

```
iwr -useb https://github.com/sigstore/cosign/releases/latest/download/cosign-windows-amd64.exe -OutFile cosign.exe
```

Verify installation:

```
cosign version
```

🔷 Step 2: Build and Push a Docker Image

Let’s create and push an image to **Docker Hub** or a private registry.

1️⃣ Build the Image

```
docker build -t myusername/myapp:latest .
```

2️⃣ Push the Image

```
docker push myusername/myapp:latest
```

🔷 Step 3: Generate a Signing Key

Before signing, generate a **private/public key pair**:

```
cosign generate-key-pair
```

🔷 This creates:

- **cosign.key** (Private key)
- **cosign.pub** (Public key)

✅ **Securely store cosign.key**—never share it!

🔷 Step 4: Sign the Docker Image

```
cosign sign --key cosign.key myusername/myapp:latest
```

📌 **If using a public registry**, you may be prompted for **Docker Hub authentication**.

🔷 Step 5: Verify the Signed Image

Once signed, verify the signature before pulling the image:

```
cosign verify --key cosign.pub myusername/myapp:latest
```

✅ If the signature is valid, you'll see output confirming the **image was signed** and **not tampered with**.

🔷 Optional: Use Keyless Signing (No Keys Needed!)

Instead of managing keys, you can sign using **OIDC authentication (GitHub Actions, Google, etc.)**:

```
cosign sign myusername/myapp:latest
```

🔷 This verifies **your identity** with **GitHub, Google, or another OIDC provider** before signing.

🔷 Summary of Key Commands

Command	Purpose
`cosign generate-key-pair`	Creates signing keys.
`cosign sign --key cosign.key image-name`	Signs the image.
`cosign verify --key cosign.pub image-name`	Verifies the signed image.
`cosign sign image-name`	Keyless signing (OIDC).