Abstract

This Deployment Guide gives system administrators and developers information about installing and verifying Red Hat’s Trusted Artifact Signer service running on the OpenShift Container Platform. Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
Table of Contents

PREFACE .......................................................................................................................... 3

CHAPTER 1. INSTALLING RED HAT TRUSTED ARTIFACT SIGNER ON OPENSHEET ................................................. 4

CHAPTER 2. VERIFY THE TRUSTED ARTIFACT SIGNER INSTALLATION ................................................................. 7
  2.1. SIGNING AND VERIFYING COMMITS BY USING GITSIGN FROM THE COMMAND-LINE INTERFACE 7
  2.2. SIGNING AND VERIFYING CONTAINERS BY USING COSIGN FROM THE COMMAND-LINE INTERFACE 8

CHAPTER 3. CONFIGURING GRAFANA FOR THE TRUSTED ARTIFACT SIGNER SERVICE ............................. 12
Welcome to the Red Hat Trusted Artifact Signer Deployment Guide!

These procedures help guide you on deploying the full Trusted Artifact Signer (TAS) software stack on OpenShift, verifying that deployment, and configures a monitoring solution for the service.

IMPORTANT

Red Hat Trusted Artifact Signer service is a Technical Preview release. A Technical Preview release has features that are in early development.

To give feedback or inform our engineering team of any technical issues with Trusted Artifact Signer, email us at tas-support@redhat.com.
CHAPTER 1. INSTALLING RED HAT TRUSTED ARTIFACT SIGNER ON OPENSHIFT

You can install Red Hat’s Trusted Artifact Signer (TAS) service on OpenShift with Keycloak OpenID Connect (OIDC) integration by running a simple script. The script configures the OpenShift environment, and runs a Helm chart to install the full TAS stack on OpenShift.

**IMPORTANT**

The TAS service is a Technical Preview release, and is in early development. For this release, there is no Microsoft Windows support. The install script referenced in this procedure is for Linux operating systems only.

Prerequisites

- A GitHub user account.
- A Red Hat Single-Sign On (SSO) user account.
- A healthy running OpenShift cluster, with the `git` binary, and the `helm` binary installed.
- Access to the OpenShift web console with the `cluster-admin` role.
- Access to the command-line interface on an OpenShift node within the cluster.
- Download your pull secret from Red Hat’s Hybrid Cloud Console.
- Have 1 GB of container storage available for the Keycloak PostgreSQL pod.

Procedure

1. Open a terminal on a node within the OpenShift cluster, and clone the Sigstore OpenShift GitHub repository using SSH:

   **Example**

   ```
   $ git clone --branch release-1.0.beta git@github.com:securesign/sigstore-ocp.git
   ```

   **NOTE**

   Make sure you add the SSH public key to your GitHub account.

2. Change to the `sigstore-ocp` directory:

   **Example**

   ```
   $ cd sigstore-ocp
   ```

3. From a terminal on a node within the OpenShift cluster, log into OpenShift:

   **Syntax**

   ```
   oc login --token=TOKEN --server=SERVER_URL_AND_PORT
   ```
4. Run the `tas-easy-install.sh` script to deploy the TAS service:

```
$ ./tas-easy-install.sh
```

This script sets Red Hat SSO as the default OIDC provider, creates a new realm called `sigstore`. This script creates root certificates for the Fulcio service, and prompts you for four things: an organization name, an email address, to set a certificate password, and to add the absolute path to your pull secret JSON file. Also, this script creates a test user, “jdoe”. Here is the user’s information:

- Password is secure.
- Email address is `jdoe@redhat.com`.

5. After the scripts completes, watch the progress of the various jobs:

```
$ watch oc get jobs -A
```

When the `tuf-system` job completes, the TAS service is ready for use.

```
NAMESPACE  NAME                 COMPLETIONS  DURATION  AGE
----------  ---------------------  ----------  ---------  ----
tuf-system  copy-secrets-job      1/1         2m38s     10m
```

6. Optional. You can create a new Keycloak user by modifying the `KeycloakUser` custom resource in the `example-user.yaml` file, and then applying it:

```
$ oc apply -f keycloak/resources/example-user.yaml --namespace keycloak-system
```

7. To verify if the installation was successful, you can try signing a container image or a Git commit.

Additional resources
• See the Configuring Grafana for the Trusted Artifact Signer service chapter for more details on monitoring the TAS service.

• Updating the global cluster pull secret with the "registry.redhat.io" authentication credentials.
CHAPTER 2. VERIFY THE TRUSTED ARTIFACT SIGNER INSTALLATION

2.1. SIGNING AND VERIFYING COMMITS BY USING GITSIGN FROM THE COMMAND-LINE INTERFACE

The gitsign tool gives you the ability to sign and verify Git repository commits by using Red Hat’s Trusted Artifact Signer (TAS) service.

Prerequisites

- Access to the OpenShift web console.
- Access to the command-line interface on an OpenShift node within the cluster.
- A Red Hat Single-Sign On (SSO) user account.
- A healthy running OpenShift cluster with the Red Hat Trusted Artifact Signer stack installed.

Procedure

1. Download the gitsign binary from the OpenShift cluster.
   a. Login to the OpenShift web console. From the home page, click ? icon, click Command line tools, go to the gitsign download section, and click the link for your platform.
   b. Open a terminal, decompress the zip file, and set the execute bit on the gitsign binary:

   \[
   \text{Linux and MacOS Example} \\
   \$ \text{gunzip gitsign.gz} \\
   \$ \text{chmod +x gitsign}
   \]

2. Configure your shell environment.
   a. Open a terminal on the OpenShift cluster where you deployed the TAS service, and run the tas-env-variables.sh script from the sigstore-ocp directory:

   \[
   \text{Example} \\
   \$ \text{cd sigstore-ocp} \\
   \$ \text{source tas-env-variables.sh}
   \]
   b. Or, you can manually set the environment variables:

   \[
   \text{Example} \\
   \$ \text{export OPENSIGHT_APPS_SUBDOMAIN=apps.$(oc get dns cluster -o jsonpath=\{".spec.baseDomain\})} \\
   \$ \text{export OIDC_AUTHENTICATION_REALM=sigstore} \\
   \$ \text{export FULCIO_URL=https://fulcio.$OPENSIGHT_APPS_SUBDOMAIN} \\
   \$ \text{export OIDC_ISSUER_URL=https://keycloak-keycloak-}
   \]


3. Configure the local repository configuration to sign your commits by using the TAS service:

Example

```bash
$ git config --local commit.gpgsign true
$ git config --local tag.gpgsign true
$ git config --local gpg.x509.program gitsign
$ git config --local gpg.format x509
$ git config --local gitsign.fulcio $FULCIO_URL
$ git config --local gitsign.rekor $REKOR_URL
$ git config --local gitsign.issuer $OIDC_ISSUER_URL
```

4. Make a commit to the local repository:

Example

```bash
$ git commit -S -m "Test of a signed commit"
```

A web browser opens allowing you to sign the commit with an email address.

**NOTE**

If you used the `tas-easy-install.sh` script to deploy the TAS software stack, you can use the `jdoe@redhat.com` email address with `secure` as the password to verify the TAS installation.

5. Verify the commit:

**Syntax**

```bash
gitsign verify --certificate-identity=SIGNING_EMAIL --certificate-oidc-issuer=$OIDC_ISSUER_URL HEAD
```

**Example**

```bash
$ gitsign verify --certificate-identity=user@example.com --certificate-oidc-issuer=$OIDC_ISSUER_URL HEAD
```

**Additional resources**

- See the [Installing Red Hat Trusted Artifact Signer on OpenShift](#) chapter for more details on installation.

### 2.2. SIGNING AND VERIFYING CONTAINERS BY USING COSIGN FROM THE COMMAND-LINE INTERFACE

The `cosign` tool gives you the capability to sign and verify Open Container Initiative (OCI) container images, along with other build artifacts by using Red Hat’s Trusted Artifact Signer (TAS) service.
IMPORTANT

Always sign container images by using the image digest, for example, example-operator@sha256:2788a….

Prerequisites

- Access to the OpenShift web console.
- Access to the command-line interface on an OpenShift node within the cluster.
- A Red Hat Single-Sign On (SSO) user account.
- A healthy running OpenShift cluster with the Red Hat Trusted Artifact Signer stack installed.

Procedure

1. Download the cosign binary from the OpenShift cluster.
   a. Login to the OpenShift web console. From the home page, click ? icon, click Command line tools, go to the cosign download section, and click the link for your platform.
   b. Open a terminal, decompress the zip file, and set the execute bit on the cosign binary:

   **Linux and MacOS Example**
   
   ```
   $ gunzip cosign.gz
   $ chmod +x cosign
   ```

2. Configure your shell environment.
   a. Open a terminal on the OpenShift cluster where you deployed the TAS service, and run the tas-env-variables.sh script from the sigstore-ocp directory:

   **Example**
   
   ```
   $ cd sigstore-ocp
   $ source tas-env-variables.sh
   ```
   
   b. Or, you can manually set the environment variables:

   **Example**
   
   ```
   $ export OPENSHIFT_APPS_SUBDOMAIN=apps.$(oc get dns cluster -o jsonpath='{.spec.baseDomain}')
   $ export OIDC_AUTHENTICATION_REALM=sigstore
   $ export FULCIO_URL=https://fulcio.$OPENSHIFT_APPS_SUBDOMAIN
   $ export OIDC_ISSUER_URL=https://keycloak-keycloak-system.$OPENSHIFT_APPS_SUBDOMAIN/auth/realms/$OIDC_AUTHENTICATION_REALM
   $ export REKOR_URL=https://rekor.$OPENSHIFT_APPS_SUBDOMAIN
   $ export TUF_URL=https://tuf.$OPENSHIFT_APPS_SUBDOMAIN
   ```

3. Initialize The Update Framework (TUF) system:
Example

$ cosign initialize --mirror=$TUF_URL --root=$TUF_URL/root.json

4. Sign your container image:

Syntax

```
cosign sign -y --fulcio-url=$FULCIO_URL --rekor-url=$REKOR_URL --oidc-issuer=$OIDC_ISSUER_URL IMAGE_NAME
```

Example

```
$ cosign sign -y --fulcio-url=$FULCIO_URL --rekor-url=$REKOR_URL --oidc-issuer=$OIDC_ISSUER_URL example-hello-world@sha256:2788a47fd0ef1ece30898c1e608050ea71036d3329b9772dbb3d1f69313f745c
```

A web browser opens allowing you to sign the container image with an email address.

**NOTE**

If you used the `tas-easy-install.sh` script to deploy the TAS software stack, you can use the `jdoe@redhat.com` email address with `secure` as the password to verify the TAS installation.

5. Verify a signed container image by using a certificate identity and issuer:

Syntax

```
cosign verify --rekor-url=$REKOR_URL \
--certificate-identity=SIGNING_EMAIL \
--certificate-oidc-issuer=OIDC_ISSUER \
IMAGE_NAME
```

Example

```
$ cosign verify --rekor-url=$REKOR_URL \
--certificate-identity=user@example.com \
--certificate-oidc-issuer=keycloak \
example-hello-world@sha256:2788a47fd0ef1ece30898c1e608050ea71036d3329b9772dbb3d1f69313f745c
```

**NOTE**

You can also use regular expressions for the certificate identity and issuer by using the following options to the `cosign` command, `--certificate-identity-regexp` and `--certificate-oidc-issuer-regexp`.

Additional resources

- See the [Installing Red Hat Trusted Artifact Signer on OpenShift](#) chapter for more details on installation.
• The Update Framework home page.
CHAPTER 3. CONFIGURING GRAFANA FOR THE TRUSTED ARTIFACT SIGNER SERVICE

You can use Grafana, along with Prometheus, to monitor the Trusted Artifact Signer (TAS) service running on OpenShift. Grafana gives you a web-base dashboard for monitoring the TAS service.

IMPORTANT

The TAS service is a Technical Preview release, and is in early development. For this release, there is no Microsoft Windows support. The install script referenced in this procedure is for Linux operating systems only.

IMPORTANT

If you used the `tas-easy-install.sh` script to install the TAS service on OpenShift, you can skip steps 1 and 2 in this procedure.

Prerequisites

- A healthy running OpenShift cluster, with the `git` binary, the `jq` binary, and the `helm` binary installed.
- Access to the OpenShift web console with the `cluster-admin` role.
- Access to the command-line interface on an OpenShift node within the cluster.
- Enable monitoring for user-defined projects.

Procedure

1. Open a terminal on a node within the OpenShift cluster, and log into OpenShift:

   Syntax

   ```
   oc login --token=TOKEN --server=SERVER_URL_AND_PORT
   ```

   Example

   ```
   $ oc login --token=sha256~ZvFDBvoIYAbVECixS4-WmkN4RfnNd8Neh3y1WuiFPXC --server=https://example.com:6443
   ```

   NOTE

   You can find your command-line login token and URL from the OpenShift web console. Log in to the OpenShift web console. Click your user name, and click Copy login command. Offer your user name and password again, if asked, and click Display Token to view the command.

2. From a terminal on a node within the OpenShift cluster, and clone the Sigstore OpenShift GitHub repository using SSH:

   Example
$ git clone --branch release-1.0.beta git@github.com:securesign/sigstore-ocp.git

NOTE
Make sure you add the SSH public key to your GitHub account.

3. Change to the **sigstore-ocp** directory:

   Example

   `$ cd sigstore-ocp`

4. Run the **tas-grafana-install.sh** script to deploy Grafana for the TAS service:

   Example

   `$ ./tas-grafana-install.sh`

   This script checks to see if OpenShift monitoring is enabled, and installs Grafana and the necessary resources. After the script finishes, you get an URL string for the Grafana dashboard. Copy the URL string and paste it into a web browser to view the dashboard for the TAS service.

Additional resources

* See the [Installing Red Hat Trusted Artifact Signer on OpenShift](#) chapter for more details on installation.