Red Hat Trusted Profile Analyzer 1

Deployment Guide

Installing and configuring the Trusted Profile Analyzer service

Last Updated: 2024-05-01

Red Hat Trusted Profile Analyzer 1 Deployment Guide

Installing and configuring the Trusted Profile Analyzer service

Legal Notice

Abstract

This Deployment Guide gives system administrators information about installing Red Hat's Trusted Profile Analyzer service on Red Hat's 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

Welcome to the Red Hat Trusted Profile Analyzer Deployment Guide!

This guide helps you with deploying Red Hat's Trusted Profile Analyzer (RHTPA) software stack on the Red Hat OpenShift Container Platform.

CHAPTER 1. SELECT YOUR INSTALLATION PLATFORM

To install Red Hat Trusted Profile Analyzer (RHTPA), you can select two different installation paths based on your choice of service providers. You can use Amazon Web Services (AWS), or use a variety of service providers that meet certain criteria for installing Trusted Profile Analyzer running on Red Hat's OpenShift Container Platform.

1.1. PREREQUISITES

- Red Hat OpenShift Container Platform version 4.14, or 4.15.

Select your installation path:

- Amazon Web Services
- Other services

1.2. INSTALLING TRUSTED PROFILE ANALYZER BY USING HELM WITH AMAZON WEB SERVICES

You can install Red Hat's Trusted Profile Analyzer (RHTPA) service on OpenShift by using a Helm chart from Red Hat. This procedure guides you on integrating Amazon Web Services (AWS) with RHTPA by using a customized values file for Helm.

IMPORTANT

If the secret values change after the installation, OpenShift redeploys RHTPA.

Prerequisites

- A Red Hat OpenShift Container Platform cluster running version 4.14 or later.
  - Support for the Ingress resource to serve publicly trusted certificates that use HTTPS.
- An AWS account with access to the following services:
  - Simple Storage Service (S3)
  - Simple Queue Service (SQS)
  - Relational Database Service (RDS) using a PostgreSQL database instance.
  - Cognito with an existing Cognito domain.
- Have the following S3 bucket names created:
  - **bombastic-default**
  - **vexination-default**
  - **v11y-default**
- Have the following standard SQS queue names created:
  - **bombastic-failed-default**

- **bombastic-indexed-default**
- **bombastic-stored-default**
- **vexination-failed-default**
- **vexination-indexed-default**
- **vexination-stored-default**
- **v11y-failed-default**
- **v11y-indexed-default**
- **v11y-stored-default**

- Access to the OpenShift web console with the **cluster-admin** role.
- A workstation with the **oc**, and the **helm** binaries installed.

Procedure

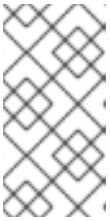
1. On your workstation, open a terminal, and log in to OpenShift by using the command-line interface:

   #### 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 login token and URL from the OpenShift web console to use on the command line. Log in to the OpenShift web console. Click your user name, and click Copy login command. Offer your user name and password again, and click Display Token to view the command.

2. Create a new project for the RHTPA deployment:

   #### Syntax

   ```
   oc new-project PROJECT_NAME
   ```

   #### Example

   ```
   $ oc new-project trusted-profile-analyzer
   ```

3. Open a new file for editing:

Example

```
$ vi values-rhtpa-aws.yaml
```

4. Copy and paste the RHTPA values file template into the new **values-rhtpa-aws.yaml** file.
5. Update the **values-rhtpa-aws.yaml** file with your relevant AWS information.
   a. Replace *REGIONAL_ENDPOINT* with your Amazon S3 storage, and Amazon SQS endpoint URLs.
   b. Replace *COGNITO_DOMAIN_URL* with your Amazon Cognito URL. You can find this information in the AWS Cognito Console, under the **App Integration** tab.
   c. Replace *REGION*, *USER_POOL_ID*, and *FRONTEND_CLIENT_ID* and *WALKER_CLIENT_ID* with your relevant Amazon Cognito information. You can find this information in the AWS Cognito Console, in the **User pool overview** section, and in the **App clients and analytics** section under the **App Integration** tab.
   d. Save the file, and quit the editor.
6. Create the S3 storage secret object by using your AWS credentials:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: storage-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  aws_access_key_id: AWS_ACCESS_KEY
  aws_secret_access_key: AWS_SECRET_KEY
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: storage-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  aws_access_key_id: RHTPASTORAGE1EXAMPLE
  aws_secret_access_key: xBalrKUtnFEMI/K7RDENG/aPxRfzCYEXAMPLEKEY
```

7. Create the SQS event bus secret object by using your AWS credentials:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
```

```
  name: event-bus-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  aws_access_key_id: AWS_ACCESS_KEY
  aws_secret_access_key: AWS_SECRET_KEY
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: event-bus-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  aws_access_key_id: RHTPAEVENTBS1EXAMPLE
  aws_secret_access_key: mBaliKUtnFEMI/K6RDENG/aPxRfzCYEXAMPLEKEY
```

8. Create two PostgreSQL database secret objects by using your Amazon RDS credentials.

   a. A PostgreSQL standard user secret object:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  db.host: DB_HOST
  db.name: DB_NAME
  db.user: USERNAME
  db.password: PASSWORD
  db.port: PORT
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  data:
  db.host: rds.us-east-1.amazonaws.com
  db.name: rhtpadb
  db.user: jdoe
  db.password: example1234
  db.port: 5432
```

b. A PostgreSQL administrator secret object:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-admin-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  db.host: DB_HOST
  db.name: DB_NAME
  db.user: USERNAME
  db.password: PASSWORD
  db.port: PORT
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-admin-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  data:
  db.host: rds.us-east-1.amazonaws.com
  db.name: rhtpadb
  db.user: admin
  db.password: example1234
  db.port: 5432
```

9. Set up your shell environment:

Syntax

```
export NAMESPACE=PROJECT_NAME
export APP_DOMAIN_URL=-$NAMESPACE.$(oc -n openshift-ingress-operator get
ingresscontrollers.operator.openshift.io default -o jsonpath='{.status.domain}')
```

Example

```
$ export NAMESPACE=trusted-profile-analyzer
$ export APP_DOMAIN_URL=-$NAMESPACE.$(oc -n openshift-ingress-operator get
ingresscontrollers.operator.openshift.io default -o jsonpath='{.status.domain}')
```

10. Add the OpenShift Helm chart repository:

Example

```
$ helm repo add openshift-helm-charts https://charts.openshift.io/
```

11. Get the latest chart information from the Helm chart repositories:

    Example

    ```
    $ helm repo update
    ```

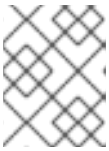
12. Run the Helm chart:

    Syntax

    ```
    helm install redhat-trusted-profile-analyzer openshift-helm-charts/redhat-trusted-profile-analyzer -n $NAMESPACE --values PATH_TO_VALUES_FILE --set-string appDomain=$APP_DOMAIN_URL
    ```

    Example

    ```
    $ helm install redhat-trusted-profile-analyzer openshift-helm-charts/redhat-trusted-profile-analyzer -n $NAMESPACE --values values-rhtpa-aws.yaml --set-string appDomain=$APP_DOMAIN_URL
    ```

    NOTE

    You can run this Helm chart many times to apply the currently configured state from the values file.

13. Once the installation finishes, you can log in to the RHTPA console by using a user's credentials from the Cognito user pool. You can find the RHTPA console URL by running the following command:

    Example

    ```
    $ oc -n $NAMESPACE get route --selector app.kubernetes.io/name=spog-ui -o jsonpath='https://{.items[0].status.ingress[0].host}{"\n"}'
    ```

14. A scheduled Cron job runs each day to gather the latest Common Vulnerabilities and Exposures (CVE) data for RHTPA. Instead of waiting, you can manually start this Cron job by running the following command:

    Example

    ```
    $ oc -n $NAMESPACE create job --from=cronjob/v11y-walker v11y-walker-now
    ```

    Once the Cron job finishes, delete this Cron job:

    Example

    ```
    $ oc -n $NAMESPACE delete job v11y-walker-now
    ```

Additional resources

- Amazon Simple Storage Service (S3) endpoints and quota documentation.
- Amazon Simple Queue Service (SQS) documentation.
- Amazon Cognito documentation.
- Amazon Relational Database Service (RDS) documentation.
- Creating an Amazon S3 bucket .
- Creating a standard Amazon SQS queue .

1.3. INSTALLING TRUSTED PROFILE ANALYZER BY USING HELM WITH OTHER SERVICES

You can install Red Hat's Trusted Profile Analyzer (RHTPA) service on OpenShift by using a Helm chart from Red Hat. You need to have a Simple Storage Service (S3) compatible storage infrastructure, an OpenID Connect (OIDC) provider, a PostgreSQL database, and use Red Hat AMQ Streams for OpenShift. This procedure guides you on integrating these various services with RHTPA by using a customized values file for Helm.

IMPORTANT

If the secret values change after the installation, OpenShift redeploys RHTPA.

Prerequisites

- A Red Hat OpenShift Container Platform cluster running version 4.14 or later.
  - Support for the Ingress resource to serve publicly trusted certificates that use HTTPS.
- Have the following S3 bucket names created:
  - **bombastic-default**
  - **vexination-default**
  - **v11y-default**
- The AMQ Streams on OpenShift service with the following topic names created:
  - **bombastic-failed-default**
  - **bombastic-indexed-default**
  - **bombastic-stored-default**
  - **vexination-failed-default**
  - **vexination-indexed-default**
  - **vexination-stored-default**
  - **v11y-failed-default**
  - **v11y-indexed-default**

- **v11y-stored-default**

- An OIDC provider for authentication.
- A new PostgreSQL database.
- Access to the OpenShift web console with the **cluster-admin** role.
- A workstation with the **oc**, and the **helm** binaries installed.

Procedure

1. On your workstation, open a terminal, and log in to OpenShift by using the command-line interface:

   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 login token and URL from the OpenShift web console to use on the command line. Log in to the OpenShift web console. Click your user name, and click Copy login command. Offer your user name and password again, and click Display Token to view the command.

2. Create a new project for the RHTPA deployment:

   Syntax

   ```
   oc new-project PROJECT_NAME
   ```

   Example

   ```
   $ oc new-project trusted-profile-analyzer
   ```

3. Open a new file for editing:

   Example

   ```
   $ vi values-rhtpa.yaml
   ```

4. Copy and paste the RHTPA values file template into the new **values-rhtpa.yaml** file.

5. Update the **values-rhtpa.yaml** file with your information.

   a. Replace *S3_ENDPOINT_URL* with your relevant S3 storage information.

b. Replace *AMQ_ENDPOINT_URL*, and *USER_NAME* with your relevant AMQ Streams information.

c. Replace *OIDC_ISSUER_URL*, *FRONTEND_CLIENT_ID* and *WALKER_CLIENT_ID* with your relevant OIDC information.

d. Save the file, and quit the editor.

6. Create the S3 storage secret object with your credentials:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: s3-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  user: USER_NAME
  password: PASSWORD
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: s3-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  user: root
  password: example123
```

7. Create the AMQ Streams secret object with your credentials:

Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: kafka-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  client_password: PASSWORD
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
```

```
  name: kafka-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  client_password: example123
```

8. Create the two PostgreSQL database secret objects with your database credentials.

   a. A PostgreSQL standard user secret object:

   Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  db.host: DB_HOST
  db.name: DB_NAME
  db.user: USERNAME
  db.password: PASSWORD
  db.port: PORT
```

   Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  data:
  db.host: rds.us-east-1.amazonaws.com
  db.name: rhtpadb
  db.user: jdoe
  db.password: example1234
  db.port: 5432
```

   b. A PostgreSQL administrator secret object:

   Syntax

```
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-admin-credentials
  namespace: PROJECT_NAME
type: Opaque
data:
  db.host: DB_HOST
  db.name: DB_NAME
```

```
  db.user: USERNAME
  db.password: PASSWORD
  db.port: PORT
```

Example

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-admin-credentials
  namespace: trusted-profile-analyzer
type: Opaque
data:
  data:
  db.host: rds.us-east-1.amazonaws.com
  db.name: rhtpadb
  db.user: admin
  db.password: example1234
  db.port: 5432
```

9. Set up your shell environment:

Syntax

```
export NAMESPACE=PROJECT_NAME
export APP_DOMAIN_URL=-$NAMESPACE.$(oc -n openshift-ingress-operator get
ingresscontrollers.operator.openshift.io default -o jsonpath='{.status.domain}')
```

Example

```
$ export NAMESPACE=trusted-profile-analyzer
$ export APP_DOMAIN_URL=-$NAMESPACE.$(oc -n openshift-ingress-operator get
ingresscontrollers.operator.openshift.io default -o jsonpath='{.status.domain}')
```

10. Add the OpenShift Helm chart repository:

Example

```
$ helm repo add openshift-helm-charts https://charts.openshift.io/
```

11. Get the latest chart information from the Helm chart repositories:

Example

```
$ helm repo update
```

12. Run the Helm chart:

Syntax

```
helm install redhat-trusted-profile-analyzer openshift-helm-charts/redhat-trusted-profile-analyzer -n $NAMESPACE --values PATH_TO_VALUES_FILE --set-string appDomain=$APP_DOMAIN_URL
```

Example

```
$ helm install redhat-trusted-profile-analyzer openshift-helm-charts/redhat-trusted-profile-analyzer -n $NAMESPACE --values values-rhtpa.yaml --set-string appDomain=$APP_DOMAIN_URL
```

NOTE

You can run this Helm chart many times to apply the currently configured state from the values file.

13. Once the installation finishes, you can log in to the RHTPA console by using a user's credentials from your OIDC provider. You can find the RHTPA console URL by running the following command:

Example

```
$ oc -n $NAMESPACE get route --selector app.kubernetes.io/name=spog-ui -o jsonpath='https://{.items[0].status.ingress[0].host}{"\n"}'
```

14. A scheduled Cron job runs each day to gather the latest Common Vulnerabilities and Exposures (CVE) data for RHTPA. Instead of waiting, you can manually start this Cron job by running the following command:

Example

```
$ oc -n $NAMESPACE create job --from=cronjob/v11y-walker v11y-walker-now
```

Once the Cron job finishes, delete this Cron job:

Example

```
$ oc -n $NAMESPACE delete job v11y-walker-now
```

APPENDIX A. RED HAT TRUSTED PROFILE ANALYZER WITH AWS VALUES FILE TEMPLATE

Red Hat's Trusted Profile Analyzer (RHTPA) with Amazon Web Services (AWS) values file template for use by the RHTPA Helm chart.

Template

```
appDomain: $APP_DOMAIN_URL

tracing: {}

ingress:
  className: openshift-default

storage:
  region: REGIONAL_ENDPOINT
  accessKey:
    valueFrom:
     secretKeyRef:
       name: storage-credentials
       key: aws_access_key_id
  secretKey:
    valueFrom:
      secretKeyRef:
        name: storage-credentials
        key: aws_secret_access_key

eventBus:
  type: sqs
  region: REGIONAL_ENDPOINT
  accessKey:
    valueFrom:
      secretKeyRef:
        name: event-bus-credentials
        key: aws_access_key_id
  secretKey:
    valueFrom:
      secretKeyRef:
        name: event-bus-credentials
        key: aws_secret_access_key

authenticator:
  type: cognito
  cognitoDomainUrl: COGNITO_DOMAIN_URL

oidc:
  issuerUrl: https://cognito-idp.REGION.amazonaws.com/USER_POOL_ID
  clients:
    frontend:
      clientId: FRONTEND_CLIENT_ID
    walker:
      clientId: WALKER_CLIENT_ID
      clientSecret:
        valueFrom:
```

```
        secretKeyRef:
          name: oidc-walker
          key: client-secret

bombastic:
  bucket: bombastic-default
  topics:
    failed: bombastic-failed-default
    indexed: bombastic-indexed-default
    stored: bombastic-stored-default

vexination:
  bucket: vexination-default
  topics:
    failed: vexination-failed-default
    indexed: vexination-indexed-default
    stored: vexination-stored-default

v11y:
  bucket: v11y-default
  topics:
    failed: v11y-failed-default
    indexed: v11y-indexed-default
    stored: v11y-stored-default

guac:
  database:
    name:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.name
    host:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.host
    port:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.port
    username:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.user
    password:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.password

  initDatabase:
    name:
      valueFrom:
```

```
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.name
host:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.host
port:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.port
username:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.user
password:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.password
```

APPENDIX B. RED HAT TRUSTED PROFILE ANALYZER WITH OTHER SERVICES VALUES FILE TEMPLATE

Red Hat's Trusted Profile Analyzer (RHTPA) with other services values file template for use by the RHTPA Helm chart.

Template

```
appDomain: $APP_DOMAIN_URL
tracing: {}
ingress:
  className: openshift-default

storage:
 endpoint: S3_ENDPOINT_URL
 accessKey:
   valueFrom:
     secretKeyRef:
       name: s3-credentials
         key: user
 secretKey:
   valueFrom:
     secretKeyRef:
       name: s3-credentials
       key: password

eventBus:
  type: kafka
  bootstrapServers: AMQ_ENDPOINT_URL:9092
  config:
    securityProtocol: SASL_PLAINTEXT
    username: "USER_NAME"
    password:
      valueFrom:
        secretKeyRef:
          name: kafka-credentials
          key: client_password
    mechanism: SCRAM-SHA-512

oidc:
  issuerUrl: OIDC_ISSUER_URL
  clients:
   frontend:
     clientId: FRONTEND_CLIENT_ID
   walker:
     clientId: WALKER_CLIENT_ID
     clientSecret:
       valueFrom:
         secretKeyRef:
           name: oidc-walker
           key: client-secret

bombastic:
 bucket: bombastic-default
 topics:
```

```
    failed: bombastic-failed-default
    indexed: bombastic-indexed-default
    stored: bombastic-stored-default

  vexination:
   bucket: vexination-default
   topics:
    failed: vexination-failed-default
    indexed: vexination-indexed-default
    stored: vexination-stored-default

  v11y:
   bucket: v11y-default
   topics:
    failed: v11y-failed-default
    indexed: v11y-indexed-default
    stored: v11y-stored-default

  guac:
   database:
    name:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.name
    host:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.host
    port:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.port
    username:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.user
    password:
      valueFrom:
        secretKeyRef:
          name: postgresql-credentials
          key: db.password

   initDatabase:
    name:
      valueFrom:
        secretKeyRef:
          name: postgresql-admin-credentials
          key: db.name
    host:
      valueFrom:
        secretKeyRef:
          name: postgresql-admin-credentials
```

```
      key: db.host
port:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.port
username:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.user
password:
  valueFrom:
    secretKeyRef:
      name: postgresql-admin-credentials
      key: db.password
```