Red Hat OpenShift

Configuring an htpasswd identity provider

Configure the `htpasswd` identity provider to allow users to log in to OpenShift Container Platform with credentials from an htpasswd file.

To define an htpasswd identity provider, perform the following tasks:

- Create an `htpasswd` file to store the user and password information.
- Create a secret to represent the `htpasswd` file.
- Define an htpasswd identity provider resource that references the secret.
- Apply the resource to the default OAuth configuration to add the identity provider.

About identity providers in OpenShift Container Platform

By default, only a `kubeadmin` user exists on your cluster. To specify an identity provider, you must create a custom resource (CR) that describes that identity provider and add it to the cluster.

OpenShift Container Platform user names containing `/`, `:`, and `%` are not supported.

About htpasswd authentication

Using htpasswd authentication in OpenShift Container Platform allows you to identify users based on an htpasswd file. An htpasswd file is a flat file that contains the user name and hashed password for each user. You can use the `htpasswd` utility to create this file.

Do not use htpasswd authentication in OpenShift Container Platform for production environments. Use htpasswd authentication only for development environments.

Creating the htpasswd file

See one of the following sections for instructions about how to create the htpasswd file:

- Creating an htpasswd file using Linux
- Creating an htpasswd file using Windows

Creating an htpasswd file using Linux

To use the htpasswd identity provider, you must generate a flat file that contains the user names and passwords for your cluster by using `htpasswd`.

Prerequisites

- Have access to the `htpasswd` utility. On Red Hat Enterprise Linux this is available by installing the `httpd-tools` package.

Procedure

- Create or update your flat file with a user name and hashed password:

```
$ htpasswd -c -B -b </path/to/users.htpasswd> <username>
<password>
```

The command generates a hashed version of the password.

For example:

```
$ htpasswd -c -B -b users.htpasswd <username> <password>
```

Example output

```
Adding password for user user1
```

- Continue to add or update credentials to the file:

```
$ htpasswd -B -b </path/to/users.htpasswd> <user_name>
<password>
```

Creating an htpasswd file using Windows

To use the htpasswd identity provider, you must generate a flat file that contains the user names and passwords for your cluster by using `htpasswd`.

Prerequisites

- Have access to `htpasswd.exe`. This file is included in the `\bin` directory of many Apache httpd distributions.

Procedure

- Create or update your flat file with a user name and hashed password:

```
> htpasswd.exe -c -B -b <\path\to\users.htpasswd> <username>
<password>
```

The command generates a hashed version of the password.

For example:

```
> htpasswd.exe -c -B -b users.htpasswd <username> <password>
```

Example output

```
Adding password for user user1
```

- Continue to add or update credentials to the file:

```
> htpasswd.exe -b <\path\to\users.htpasswd> <username>
<password>
```

Creating the htpasswd secret

To use the htpasswd identity provider, you must define a secret that contains the htpasswd user file.

Prerequisites

- Create an htpasswd file.

Procedure

- Create a `Secret` object that contains the htpasswd users file:

```
$ oc create secret generic htpass-secret --from-file=htpasswd=
<path_to_users.htpasswd> -n openshift-config (1)
```

1 The secret key containing the users file for the `--from-file` argument must be named `htpasswd`, as shown in the above command.

You can alternatively apply the following YAML to create the secret:

```
apiVersion: v1
kind: Secret
metadata:
  name: htpass-secret
  namespace: openshift-config
type: Opaque
data:
  htpasswd:
<base64_encoded_htpasswd_file_contents>
```

Sample htpasswd CR

The following custom resource (CR) shows the parameters and acceptable values for an htpasswd identity provider.

htpasswd CR

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
  - name: my_htpasswd_provider (1)
    mappingMethod: claim (2)
    type: HTPasswd
    htpasswd:
      fileData:
        name: htpass-secret (3)
```

1 This provider name is prefixed to provider user names to form an identity name.

2 Controls how mappings are established between this provider's identities and `User` objects.

3 An existing secret containing a file generated using `htpasswd`.

Additional resources

- See Identity provider parameters for information on parameters, such as `mappingMethod`, that are common to all identity providers.

Adding an identity provider to your cluster

After you install your cluster, add an identity provider to it so your users can authenticate.

Prerequisites

- Create an OpenShift Container Platform cluster.
- Create the custom resource (CR) for your identity providers.
- You must be logged in as an administrator.

Procedure

- Apply the defined CR:

```
$ oc apply -f </path/to/CR>
```

If a CR does not exist, `oc apply` creates a new CR and might trigger the following warning: `Warning: oc apply should be used on resources created by either oc create --save-config or oc apply` . In this case you can safely ignore this warning.

- Log in to the cluster as a user from your identity provider, entering the password when prompted.

```
$ oc login -u <username>
```

- Confirm that the user logged in successfully, and display the user name.

```
$ oc whoami
```

Updating users for an htpasswd identity provider

You can add or remove users from an existing htpasswd identity provider.

Prerequisites

- You have created a `Secret` object that contains the htpasswd user file. This procedure assumes that it is named `htpass-secret` .
- You have configured an htpasswd identity provider. This procedure assumes that it is named `my_htpasswd_provider` .
- You have access to the `htpasswd` utility. On Red Hat Enterprise Linux this is available by installing the `httpd-tools` package.
- You have cluster administrator privileges.

Procedure

- Retrieve the htpasswd file from the `htpass-secret` `Secret` object and save the file to your file system:

```
$ oc get secret htpass-secret -ojsonpath={.data.htpasswd} -n
openshift-config | base64 --decode > users.htpasswd
```

- Add or remove users from the `users.htpasswd` file.
  - To add a new user:

```
$ htpasswd -bB users.htpasswd <username> <password>
```

Example output

```
Adding password for user <username>
```

- To remove an existing user:

```
$ htpasswd -D users.htpasswd <username>
```

Example output

```
Deleting password for user <username>
```

- Replace the `htpass-secret Secret` object with the updated users in the `users.htpasswd` file:

```
$ oc create secret generic htpass-secret --from-
file=htpasswd=users.htpasswd --dry-run=client -o yaml -n
openshift-config | oc replace -f -
```

You can alternatively apply the following YAML to replace the secret:

```
apiVersion: v1
kind: Secret
metadata:
  name: htpass-secret
  namespace: openshift-config
type: Opaque
data:
  htpasswd:
<base64_encoded_htpasswd_file_contents>
```

- If you removed one or more users, you must additionally remove existing resources for each user.
  - Delete the `User` object:

```
$ oc delete user <username>
```

Example output

```
user.user.openshift.io "<username>" deleted
```

Be sure to remove the user, otherwise the user can continue using their token as long as it has not expired.

- Delete the `Identity` object for the user:

```
$ oc delete identity my_htpasswd_provider:<username>
```

Example output

```
identity.user.openshift.io "my_htpasswd_provider:
<username>" deleted
```

Configuring identity providers using the web console

Configure your identity provider (IDP) through the web console instead of the CLI.

Prerequisites

- You must be logged in to the web console as a cluster administrator.

Procedure

- Navigate to **Administration** → **Cluster Settings**.
- Under the **Configuration** tab, click **OAuth**.
- Under the **Identity Providers** section, select your identity provider from the **Add** drop-down menu.

You can specify multiple IDPs through the web console without overwriting existing IDPs.