Private Registry setup for IBM Resilient App Host

Table of Contents

- About this Documentation
- Private Registries
- Mirroring IBM Resilient Containers to Private Registries
- Configuring App Host for Private Registries
- Building Containers for Private Registries

History

7/20 - Initial Release

About this Documentation:

Many customers will require the use of their own private container repository for App Host. One key use includes the building and referencing to their own integrations. Another is to maintain an air-gapped environment with IBM Resilient. This document provides guidance on how to setup a private respository. This documentation covers how to setup your private registry for use with App Host.

Private Registries

Many different solutions are available for both on-premise and cloud-based container repositories. Below is a short list of some of these registries:

Cloud-Based Solutions

- IBM Cloud
- Quay.io
- Github
- Azure
- Google
- JFrog

Amazon AWS ECR doesn't appear to work App Host at this time.

On-premise Solutions

- Quay.io
- Docker
- Github
- JFrog
- Harbor

Mirroring IBM Resilient Containers to Private Registries

Unfortunately, only one registry can be used at a time with App Host. In order to use both the IBM AppExchange published Apps and your own custom integrations, it will be necessary to copy down the published containers from quay.io to your registry. Refer to README file in this directory for information on how to mirror IBM Resilient containers to a private container registry. Unfortunately, only one registry can be used with APP Host at a time, requiring this mirroring process.

Configuring App Host for Private Registries

Once your containers are available in your private container registry, follow these steps to reconfigure App Host to use this registry:

- Login to your App Host
- Run the configureAppHostRegistry tool. You may need to run the command as root.
- Follow the prompts. The registry URL needs to be start with either http:// or https://.
- If your registry is public, no additional authentication is required. Private registries require authentication user and password credentials. Some registries use API tokens and access policies for registry access. Enable read permissions for registry access.

IBM Cloud Container Registry (CR) Setup

To illustrate the setup of a private registry, the following steps were used to configure IBM Cloud Registry for App Host. Each step could be performed using either the ibmcloud cr CLI tool or the web-based console. Documentation on all these steps is found here.

- Configure a namespace for your container images. App Host requires ibmresilient.
- Create a service-id associated with your registry
- Add policies to your service-id for Reader permissions
- Create an API key. The API key will always be named 'iamapikey' but will use the API key secret as its password.

The following example shows the App Host configuration steps when using IBM Cloud Container Registry.

```
$ sudo configureAppHostRegistry
[sudo] password for appadmin:
IBM Resilient: Configuration setup for new AppHost registry
13:13:36.128 [main] DEBUG io.fabric8.kubernetes.client.Config - Trying to configure client from
Kubernetes config...
13:13:36.132 [main] DEBUG io.fabric8.kubernetes.client.Config - Found for Kubernetes config
at: [/root/.kube/config].
Using controllerId: 78ce8a6b-67ff-44b9-b0a7-9200e9d7b2f3 (Ubuntu)
Enter registry URL: https://us.icr.io
Is the registry public? (y/n) n
Enter registry username: iamapikey
Enter registry password:
Reconfiguring AppHost registry...
13:15:44.586 [main] INFO
com.ibm.security.apps.controller.configuration.steps.impl.CreateRegistrySecretStep -
Replacing secret controller-registry-secret in namespace 78ce8a6b-67ff-44b9-b0a7-
9200e9d7b2f3
13:15:44.636 [main] INFO
com.ibm.security.apps.controller.configuration.steps.impl.CreateRegistryCredentialsStep -
```

```
Creating registry credentials
13:15:44.678 [main] INFO
com.ibm.security.apps.controller.configuration.steps.impl.RestartSynchronizerStep - Restarting
deployment 'deployment-synchronizer' in namespace 78ce8a6b-67ff-44b9-b0a7-
9200e9d7b2f3
Successfully configured
```

Building Apps for Private Registries

In order to build apps for a private registry, your development environment will need the following tools:

- IBM Resilient resilient-sdk for App development.
- Docker or Podman for container creation
- Optionally the CLI tool provided by your registry provider (ex. ibmcloud for IBM Cloud Container Registry).

App Development

Refer to the App development guides provided here. There are two steps needed once your App is complete:

- Running resilient-sdk package to build the .zip file used to import your App into Resilient. Information on these procedures are found in the Resilient App Host development guides.

  Ex. resilient-sdk package -p .

- Running docker build or podman build to tag and push your App container to your private registry.

Private Registry Access

When using the CLI tool provided by your registry provider, first sign in to the registry. Refer to your registry provider's documentation on how to install the CLI tools.

IBM Cloud Container Registry has the following syntax:

```
ibmcloud login -a https://cloud.ibm.com -u passcode -p xxxx
ibmcloud cr login
```

Other registries require the developer to edit and use the docker or podman tool for authentication such as:

```
docker login -u iamapikey -p <apikey> us.icr.io
```

Other login mechanism's requires one to edit the tool's configuration file located on the developer's file system such as: ~/.docker/config.json.

Once authenticated, container's can built and pushed to the private registry as following (substitute podman for docker as appropriate):

```
docker build . -t us.icr.io/ibmresilient/fn_integration:1.0.0
docker push us.icr.io/ibmresilient/fn_integration:1.0.0
```

Existing local containers can also be retagged and pushed to your private registry:

```
$ docker images
REPOSITORY                  TAG       IMAGE ID          CREATED           SIZE
ibmresilient/fn_integration    1.0.0      59ccf4795c0b        3 weeks ago        818MB

$ docker tag 59ccf4795c0b us.icr.io/ibmresilient/fn_integration:1.0.0
$ docker push us.icr.io/ibmresilient/fn_integration:1.0.0
```