Red Hat Integration 2023.q1

Migrating Service Registry deployments

Migrating to Service Registry 2.3

Last Updated: 2023-05-16

Red Hat Integration 2023.q1 Migrating Service Registry deployments

Migrating to Service Registry 2.3

Legal Notice

Abstract

This guide describes the major changes in Service Registry version 2.x, and explains how to migrate an existing Service Registry version 1.1 deployment to version 2.x.

Table of Contents

PREFACE

MAKING OPEN SOURCE MORE INCLUSIVE

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 .

CHAPTER 1. MIGRATION FROM SERVICE REGISTRY 1.1 TO 2.X

Service Registry 2.x includes new features with breaking changes from the previous Service Registry 1.1 release. This section describes the major changes between Service Registry 1.1 and version 2.x.

Because of the breaking changes in 2.x, there is no automatic upgrade and a migration process is required. This involves moving the data in your existing registry to a new registry. You must also review your existing registry client applications and update their configuration to meet the new requirements.

When migrating to version 2.x, you must take the following major changes into account:

1.1. NEW DATA STORAGE OPTIONS

The existing registry data storage options in Service Registry 1.1 (**streams**,**jpa**, and **infinispan**) have been replaced with new storage options in version 2.x (**sql** and **kafkasql**). These new storage options provide more robust, performant, and maintainable Service Registry deployments.

For details on how to deploy Service Registry 2.x with your chosen storage option, see Installing and deploying Service Registry on OpenShift.

1.2. NEW V2 REST API

Service Registry 2.x includes a new REST API with support for artifact groups and improved long term maintainability. Service Registry still supports the original registry v1 REST API and compatibility APIs, for example, Confluent and IBM schema registry APIs. Service Registry now also implements the Schema Registry specification provided in the CNCF Cloud Events specification.

1.3. REFACTORED JAVA CLIENT LIBRARIES

- The Service Registry Java client classes are available in version 2.x in a different Maven module named **apicurio-registry-client**.
- The Kafka client serializer and deserializer (SerDes) classes are available in version 2.x in three different Maven modules, one for each supported data format: Apache Avro, Protobuf, and JSON Schema. You can now use only the module you want without pulling in transitive dependencies that you are not concerned with.

Additional resources

- For more details on the v2 REST API, see the Registry REST API documentation .

CHAPTER 2. MIGRATING SERVICE REGISTRY DATA

Migrating data to Service Registry 2.x requires exporting all data from your existing 1.1 deployment and importing it into the new 2.x deployment. If you are using Service Registry as a schema registry for Kafka applications, data migration is critical because each Kafka message carries the global identifier for the schema stored in Service Registry. This identifier must be preserved during registry data migration.

Service Registry 2.x provides an API to bulk import/export all data from your registry deployment, which guarantees that all identifiers are kept when importing data from your existing registry. The export API downloads a custom **.zip** file containing all the information for your artifacts. The import API accepts this **.zip** and loads all artifacts into the registry in a single batch.

Service Registry 1.1 does not provide an import/export API. However, version 2.x provides an export tool compatible with Service Registry 1.1 to export a **.zip**, which you can import into your 2.x registry. This tool uses common existing APIs to export all content in the registry. However, it is less performant than the 2.x export API, and should only be used when exporting from a 1.1 registry.

Prerequisites

- Running Service Registry instances of the 1.1 server you are exporting from and the 2.x server you are importing into.
- Download the Service Registry exportV1 tool from the Red Hat Customer Portal. This is a Java application that you can run on the command line.

Procedure

1. Export all the data from Service Registry 1.1 using the **exportV1** tool. This generates a **registry-export.zip** file in your current directory:

```
java -jar apicurio-registry-utils-exportV1-2.3.3.Final.jar http://old-registry.my-company.com/api
```

2. Import the **.zip** file into Service Registry 2.x using the import API:

```
curl -X POST "http://new-registry.my-company.com/apis/registry/v2/admin/import" \
  -H "Accept: application/json" -H "Content-Type: application/zip" \
  --data-binary @registry-export.zip
```

3. Check that all the artifacts have been imported into the new 2.x registry by running these commands and comparing the count field:

```
curl "http://old-registry.my-company.com/api/search/artifacts"
```

```
curl "http://new-registry.my-company.com/apis/registry/v2/search/artifacts"
```

Additional resources

- For more details on the import/export REST API, see the Service Registry User Guide .
- For more details on the export tool for migrating from version 1.x to 2.x, see the Apicurio Registry export utility for 1.x versions.

CHAPTER 3. MIGRATING SERVICE REGISTRY CLIENT APPLICATIONS

You must review your existing Service Registry client applications to ensure that the Maven dependencies and Java client configuration meet the new requirements for version 2.x. For example, this includes new Maven dependencies for the Service Registry Java REST client libraries or Kafka client serializer/deserializer (Serdes) libraries. You must also update your Java application configuration with the new registry v2 API path.

Prerequisites

- Existing Service Registry 1.1 Java client application or Kafka client producer and consumer Java applications with SerDes

Procedure

1. If you are using the Service Registry Java REST client, you must change the Maven dependencies for the Service Registry Java client libraries, which have been repackaged in version 2.x:

```
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-client</artifactId>
    <version>2.3.3.Final</version>
</dependency>
```

2. In your Java client application, you must change your registry URL configuration, from pointing to the existing v1 API path to the new v2 path. For example:

```
public class ClientExample {

    private static final RegistryRestClient client;

     public static void main(String[] args) throws Exception {
        // Create a registry client
        String registryUrl = "https://new-registry.my-company.com/apis/registry/v2";
        RegistryClient client = RegistryClientFactory.create(registryUrl);
    }
}
```

   You can find more details on the Java client in the Service Registry User Guide .

3. If you are using the Service Registry SerDes libraries, you must change the Maven dependencies, which have been repackaged in version 2.x. In Service Registry 1.1, the SerDes libraries were all provided with only one Maven dependency:

```
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-utils-serde</artifactId>
    <version>1.3.2.Final</version>
</dependency>
```

In Service Registry 2.x, the SerDes libraries have been split into three Maven dependencies, one for each supported data format: **avro**, **protobuf**, and **json schema**, depending on your use cases:

```
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-serdes-avro-serde</artifactId>
    <version>2.3.3.Final</version>
</dependency>
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-serdes-protobuf-serde</artifactId>
    <version>2.3.3.Final</version>
</dependency>
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-serdes-jsonschema-serde</artifactId>
    <version>2.3.3.Final</version>
</dependency>
```

4. In your Kafka producer and consumer Java applications, you must change your registry URL configuration from pointing to the existing v1 API path to the new v2 path. For example:
Existing registry v1 API path :

```
props.putIfAbsent(AbstractKafkaSerDe.REGISTRY_URL_CONFIG_PARAM, "http://old-registry.my-company.com/api");
```

New registry v2 API path:

```
props.putIfAbsent(SerdeConfig.REGISTRY_URL, "http://new-registry.my-company.com/apis/registry/v2");
```

The refactored SerDes libraries also include other important changes to configuration properties. For more details on SerDes configuration, see the Service Registry User Guide .

Additional resources

- For detailed configuration examples, see the Apicurio Registry example applications .

APPENDIX A. USING YOUR SUBSCRIPTION

Service Registry is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing your account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a subscription

1. Go to access.redhat.com.
2. Navigate to **My Subscriptions**.
3. Navigate to **Activate a subscription** and enter your 16-digit activation number.

Downloading ZIP and TAR files

To access ZIP or TAR files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal **Product Downloads** page at access.redhat.com/downloads.
2. Locate the **Red Hat Integration** entries in the **Integration and Automation** category.
3. Select the desired Service Registry product. The **Software Downloads** page opens.
4. Click the **Download** link for your component.

Registering your system for packages

To install RPM packages on Red Hat Enterprise Linux, your system must be registered. If you are using ZIP or TAR files, this step is not required.

1. Go to access.redhat.com.
2. Navigate to **Registration Assistant**.
3. Select your OS version and continue to the next page.
4. Use the listed command in your system terminal to complete the registration.

To learn more see How to Register and Subscribe a System to the Red Hat Customer Portal .