Apache Knox v1.0

Apache Knox gateway is a specialised reverse proxy gateway for various Hadoop REST APIs. However, the gateway is built entirely upon a fairly generic framework. This framework is used to “plug-in” all of the behaviour that makes it specific to Hadoop in general and any particular Hadoop REST API. It would be equally as possible to create a customised reverse proxy for other non-Hadoop HTTP endpoints. This approach is taken to ensure that the Apache Knox gateway can scale with the rapidly evolving Hadoop ecosystem.

Throughout this guide we will be using a publicly available REST API to demonstrate the development of various extension mechanisms. http://openweathermap.org/

Build prerequisites

- OpenJDK8
- knox
- git
- maven@v3.3.9
- docker
- docker.io

Install OpenJDK

$ sudo apt-get install openjdk-8-jdk

Make sure you have the right OpenJDK version

$ java -version

It should display 1.8.0_111

Set JAVA_HOME

$ export JAVA_HOME=`readlink -f /usr/bin/java | sed "s:jre/bin/java::"`

Building Apache Knox

$ git clone https://github.com/apache/knox.git
$ cd knox
knox$ mvn -DskipTests clean package

Building Docker Image

Apache Knox ships with a docker Maven module that will build a Docker image. To build the Knox Docker image, you must have Docker running on your machine. The following Maven command will build Knox and package it into a Docker image.
This will build 2 Docker images:

- apache/knox:gateway-1.3.0
- apache/knox:ldap-1.3.0

The gateway image will use an entry point to start Knox Gateway. The ldap image will use an entry point to start Knox Demo LDAP.

An example of using the Docker images would be the following:

```bash
$ docker run -d --name knox-ldap -p 33389:33389 apache/knox:ldap-1.3.0
$ docker run -d --name knox-gateway -p 8443:8443 apache/knox:gateway-1.3.0
```

Using docker-compose that would look like this:

```bash
$ docker-compose -f gateway-docker/src/main/resources/docker-compose.yml up
```

The images are designed to be a base that can be built on to add your own providers, descriptors, and topologies as necessary.

References

The reference documentation here