ODL Micro Project

Name

ODL Micro Project

Repo Name

odlmicro

Description

Run a typical OpenDaylight SDN component such as NETCONF or OpenFlowPlugin without requiring the Apache Karaf OSGi runtime container.

What is the main goal?

- Simplifying the ODL deployment
- Reducing the time it takes to develop and debug ODL features
- Reducing the runtime memory footprint
- Reduce the start up time

The project takes forward work done previously under the "Simplifying ODL" led by Unknown User (vorburger) .

For more information see these presentations:

- 2018.09 at the ODL DDF during ONS Europe in Amsterdam presented by Michael Voburger.
- Micro-distributions based on ODL-Simple presented by Tejas Nevrekar
- Past Meetings in ODL led by MIchael Voburger

This is required to be a **self-managed project**.

Scope

The project currently uses Guice as dependency injection (DI) framework, instead the OSGi Blueprint (BP) implementation Apache Aries which is used in Karaf. This choice was made just because Guice was already used for "Component Tests" in ODL, and infrautils has some useful related helpers.

- ODL projects' code is not made to depend on Guice. We simply re-use declarative DI by javax.inject standard @Singleton & @Inject annotations, which work for both Blueprint (with the blueprint-maven-plugin) as well as Guice.
- ODL projects are encouraged to migrate their use of <bean>, <reference> and <service> elements in BP XML to annotations to ease this integration. When we exceptionally need to "stitch together" objects in a non-trivial fashion (e.g. custom object factories & external configuration), we migrate BP XML logic into simple *Wiring classes. Wiring classes are not Guice specific, but pure plain old simple Java. We then write small *Module classes, which are Guice specific, to tie everything together.
- This includes support for custom Karaf CLI Commands, based on the ch.vorburger.karaf.simple POC.
- We will still support the existing karaf based ODL deployment. As a part of the self-managed project the additional simple-artifacts will be made in an alternate distribution.
- The system integration tests can be reused with a different installation procedure to validate these respective micro-distributions.
- The mycilla dependencies will be part of this project.

Open questions

- During the development cycle how karaf and odl-simple sit side by side and work? Or we have to first make all changes from BP to POJOs with annotation and test all together later when everything is done. Any specific input on setting up the development environment ?
- Do we create a separate project or perform it as a part of *integration*?
  1. Start with a separate repository & a self-managed project for that. The project only builds its own code. This is our preferred option.
  2. An ideal solution would have been patching all upstream projects and move all this code to the respective ones like netconf, openflowplugin etc. This could slow down the prototyping necessary to make this work.
  3. Re-use auto-release by picking code from the upstream and adding our own code there. This also needs its own repository.
- How do we handle netvirt/genius like projects which have multiple dependencies?
  1. The primary motive is to move the plugin projects into its own separate containers for scale. However, we need to solve the problem of communications between these components and other applications like netvirt, genius, etc. This needs a collaborative effort to separate the application logic from the southbound plugins and develop a communication mechanism between the two either via gRPC or message bus.
  2. On discussing further, the main intention here is to generate a micro-distribution for netvirt that includes all its dependencies at compile time. This can be accomplished readily by the current scope of micro-odl as outlined above.
- Need to confirm whether there are licensing concerns that we need to consider before bringing this code form github. It is already EPLv1.0.

Future Work

- Define a solution and implemenation plan for scaling ODL including the solution for communication between containers

Resources Committed (developers committed to working)

- Tejas Nevrekar - tejas.nevrekar@gmail.com
- Balaji Varaaraju "bvaadar@luminanetworks.com"
- Jamo Luhrsen "jluhrsen@luminanetworks.com"
- Dhiman Ghosh "dhiman.ghosh@ericsson.com"

Initial Committers

- Tejas Nevrekar - tejas.nevrekar@gmail.com
- Balaji Varaaraju "bvaadar@luminanetworks.com"
- Jamo Luhrsen "jluhrsen@luminanetworks.com"

Vendor Neutral

Current codebase - https://github.com/vorburger/opendaylight-simple

License is compatible (Eclipse Public License - v 1.0)

Past commits to ODL - https://git.opendaylight.org/gerrit/#/q/topic:simple-dist

Meets Board Policy (including IPR)