

YANG PUBSUB: Boron Build and Installation Guide

Contents

- [Introduction](#)
- [Description](#)
- [Software Component Overview](#)
- [Set up Basic Environment \(i.e., Virtual box, Linux, Maven\)](#)
- [Install OpenDaylight Boron](#)
- [Install OpenDaylight Netconf server](#)
- [Run OpenDaylight and Install features for yang-pubsub](#)
- [Reference links](#)

Introduction

This section describes how to use the YANG-PUBSUB feature in OpenDaylight and contains configuration, administration, and management sections for the feature.

Description

The YANG-PUBSUB project allows subscriptions to be placed on targeted subtrees of YANG data stores residing on remote devices. Changes in YANG objects within the remote subtree can be pushed to an OpenDaylight controller as specified without a requiring the controller to make a continuous set of fetch requests.

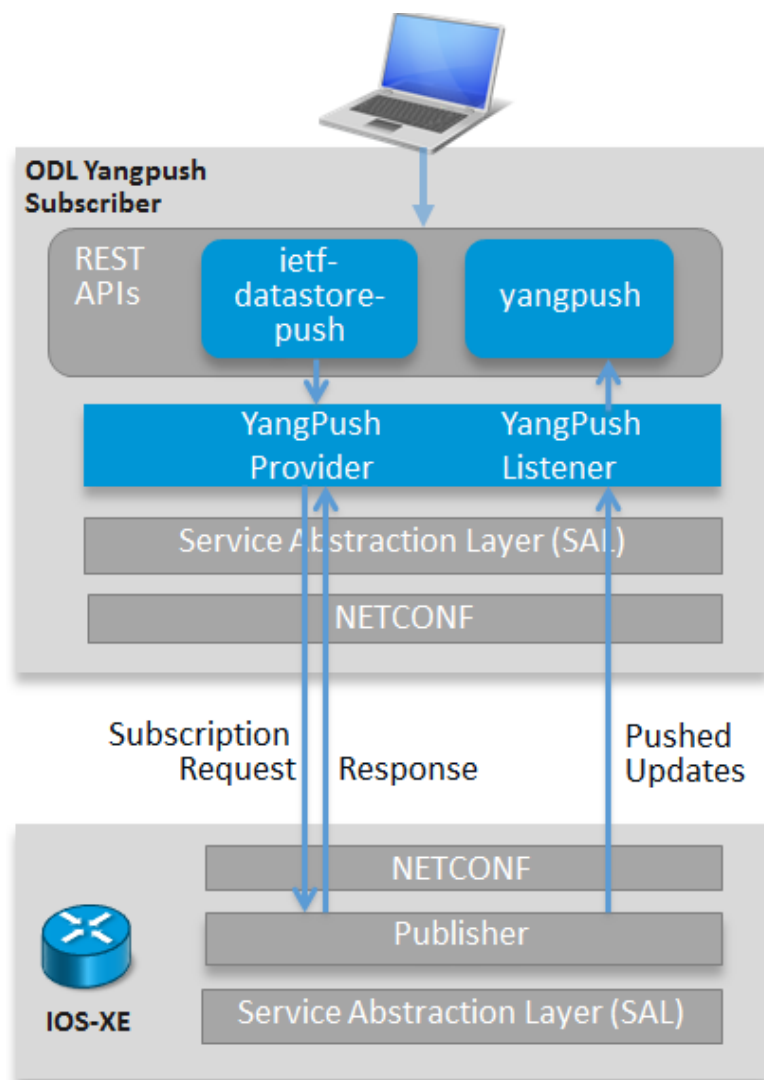
Software Component Overview

Yang Push consists of two components YANGPUSH Provider and YANGPUSH Listener

YANGPUSH Provider receives create-subscription requests from applications and then establishes/registers the corresponding listener, which will receive information pushed by a publisher. In addition, YANGPUSH Provider also invokes an augmented OpenDaylight create-subscription RPC which enables applications to register for notification as per rfc5277.

The YANGPUSH Listener accepts update notifications from a device after they have been de-encapsulated from the NETCONF transport. The YANGPUSH Listener then passes these updates to MD-SAL. Applications should monitor MD-SAL for the availability of newly pushed subscription updates.

More data can found in [draft-ietf-netconf-yang-push-00](#)



Set up Basic Environment (i.e., Virtual box, Linux, Maven)

STEP 1: Install Ubuntu Linux in Virtual Box on Windows 1

Download VM version 5.1.10 and Ubuntu 64-bit from the links

Virtual box VMM: <https://www.virtualbox.org/>

Ubuntu 64-bit: <http://www.ubuntu.com/download/desktop>

Note: This link provides the completes installation instructions

<https://www.youtube.com/watch?v=ncA85gRAJxk>

STEP 2: Download and install the Java JDK

- `sudo apt-get install openjdk-8-jdk`
- `sudo apt-get install openjdk-8-jre`
- `sudo apt-get update`
- `sudo apt-get install default-jre-headless`

STEP 3: Set JAVA_HOME=/usr/lib/jvm/default-java environment variable

- `vi ~/.bashrc`

`export JAVA_HOME=/usr/lib/jvm/default-java` in bashrc file

Note: Environment variable can placed at anywhere in the bashrc file

STEP 4: Download and install maven

- `sudo apt-get install maven`

STEP 5: Check if maven installed

- `mvn -version`

Note: If maven is installed then check if .m2 is available at \$HOME\.m2 If it is not available, just run mvn once, and the .m2 should be created at \$HOME\.m2

STEP 6: Install Git

- `sudo apt-get install git-core`

Install OpenDaylight Boron

STEP 1: Create directory and pull the distribution code

Ex: if you are at your home dir then `cd /home` and create dir of your convenient name

```
mkdir odl-17
```

```
mkdir odl-17/gerrit
```

```
cd /home/odl-17/gerrit
```

```
git clone https://git.opendaylight.org/gerrit/p/integration/distribution.git
```

STEP 2: Navigate to the directory and import maven repositories and build dependencies

Ex: `cd distribution`

```
$cp -n ~/.m2/settings.xml{,.orig} ; \
```

```
wget -q -O -
```

```
https://raw.githubusercontent.com/opendaylight/odlparent/master/settings.xml > ~/.m2/settings.xml
```

STEP 3: Checkout the latest distribution branch

Ex: `git checkout stable/boron`

you will be prompted with the message "Already on 'stable/boron'"

STEP 4: Check whether the branch is correct

```
$git branch
```

You will be prompted with the below outcome

```
master
```

```
*stable/boron"
```

STEP 5: Start the build, Successful Build will take 40-50 minutes

```
Ex: cd /home/odl-17/gerrit/distribution
```

```
mvn clean install -Pq
```

Important Note: To speedup compilation use `-DskipTests` option, At this point build will may fail for certain reasons, to fix them please re-run the command given in STEP 2.

Make sure `settings.xml` should not be empty. For more information, please see the reference links at the end of document.

Install OpenDaylight Netconf server

STEP 1: Clone the opendaylight netconf server code

```
cd /home/odl-17/gerrit
```

```
mkdir netconf
```

```
git clone https://git.opendaylight.org/gerrit/netconf.git
```

Important Note: The netconf code should cloned at the same level where distribution is cloned, the directory is having 2 sub-trees under it one for distribution and one for netconf.

STEP 2: Navigate to the directory and import maven repositories and build dependencies

Ex: `cd /home/odl-17/gerrit/netconf`

run the below command

`cp -n ~/.m2/settings.xml{,.orig} ; \`

`wget -q -O -`

`https://raw.githubusercontent.com/opendaylight/odlparent/master/settings.xml > ~/.m2/settings.xml`

STEP 3: Checkout the latest distribution branch

Ex: `git checkout stable/boron`

`cd /home/odl-17/gerrit/netconf`

you will be prompted with the message "Already on 'stable/boron'"

STEP 4: Check whether the branch is correct

`git branch`

You will be prompted with the below outcome

master

*stable/boron

STEP 5: Start the build, Successful Build will take 20-30 mins

Ex: `cd /home/odl-17/gerrit/netconf`

`mvn clean install -DskipTests`

STEP 6: Once the build success at netconf directory rerun the

"mvn clean install -DskipTests" at distribution directory build will generated at distribution/distribution-karaf/target/assembly/bin/karaf

Note: The build location would be /home/odl-17/gerrit/distribution/distribution-karaf/target/assembly/bin/karaf

Run OpenDaylight and Install features for yang-pubsub

STEP 1: Run image from distribution directory

```
cd /home/odl-17/gerrit/distribution  
./distribution-karaf/target/assembly/bin/karaf
```

STEP 2: Enable the required features to test

```
feature:install odl-netconf-mdsal  
feature:install odl-netconf-all  
feature:install odl-yangpush
```

STEP 3: This is an additional information, if something went wrong (ex: connection establishment, feature installation) at server side we option to enable the logs and analyze

```
log:tail: To check the latest logs  
log:display: For the whole current logs  
log:set <LEVEL>: For log levels  
logout: To exit from ODL
```

Reference links

1. [GettingStarted:Development Environment Setup](#)
2. [GettingStarted:Checkout Stable branch](#)
3. [opendaylight-setup](#)