

OVSDB:Installation

- [OVSDB:Installation Guide](#)
- [Introduction](#)
- [Installation](#)
 - [Prerequisites](#)
 - [Downloading the Controller Distribution](#)
- [Starting the Controller](#)
- [OVSDB Developer Installation for Hacking and Development](#)
 - [Pre-requisites for Fedora, CentOS, and RHEL](#)
 - [Pre-requisites for Mac OS X](#)
 - [Downloading and Building OVSDB Source Code](#)
 - [Building and Running Edited OVSDB Project Code](#)
 - [Generate SSH Keys for Git](#)
 - [Installing Open vSwitch](#)
 - [Additional Tools, Postman, Mininet and Wireshark](#)

OVSDB:Installation Guide

Information about Installing OVSDB for the Hydrogen release is available at: [here](#).

OVSDB has a wide range of use cases in which it can be deployed or developed. The use cases have different installations and environments. We will briefly cover basic installations, and place more specific implementation details in the tutorials section.

- Also, visit the OVSDB YouTube channel for tutorials, and the weekly Hangouts which are open to anyone:

[OpenDaylight OVSDB Youtube Channel](#)

Introduction

To set up the latest OVSDB for a test/dev environment is as simple as downloading the OpenDaylight Karaf distribution from the following location:

<http://nexus.opendaylight.org/content/repositories/opendaylight.snapshot/org/opendaylight/integration/distribution-karaf/0.2.0-SNAPSHOT/>

Installation

Prerequisites

The OpenDaylight Controller requires Java 7. First, verify that you have the necessary Java libraries to run the JVM.

For example, on a Fedora, RHEL, or CentOS host, you can run the following to satisfy any Java dependencies:

```
$ sudo yum install java-1.7.0-openjdk
```

- If you are on a Mac, download and install the Java JDK available at: [Java SE Downloads](#).

Downloading the Controller Distribution

```
$ wget http://nexus.opendaylight.org/content/repositories/automatedweeklyreleases-1024/org/opendaylight/integration/distribution-karaf/0.2.0-Helium/distribution-karaf-0.2.0-Helium.zip
$ unzip distribution-karaf-*.zip
```

Starting the Controller

To load the controller with the OpenFlow v1.3 plugin, use the following commands:

```
$ cd distribution-karaf-0.2.0-SNAPSHOT/
$ bin/karaf
$ feature:install odl-base-jackson odl-adsal-compatibility odl-ovsdb-openstack odl-ovsdb-northbound
```

OVSDB Developer Installation for Hacking and Development

For developing in OpenDaylight you will also need Apache Maven, a software project management and comprehension tool. You also require the Java Framework. Using Git, you can clone the OVSDB repository, hack code, build the jars, and copy them into the distribution to execute the edited code.

Pre-requisites for Fedora, CentOS, and RHEL

1. Install both Maven3 and the Java JDK:

```
$ sudo yum install maven3 java-1.7.0-openjdk
```

2. Next, export the following ENV variables for Java and Maven. Add this to your `~/.bashrc` directory:

```
$ vi ~/.bashrc
$ export M2_REPO=/Users/<username>/.m2/repository
$ export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/
#If plenty of memory is available on the target system, you can go as high as:
$ export MAVEN_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=2048m"
# Or as low as
$ export MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=512m"
```

3. Verify the ENV with the following:

```
$ export
declare -x MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=512m"
declare -x JAVA_HOME="/Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/"
declare -x M2_REPO="/Users/username/.m2/repository"
declare -x MAVEN_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=2048m"
```

Pre-requisites for Mac OS X

1. Download and install the Java JDK available at: [Java SE Downloads](#).
2. There are various ways to install Maven on a Mac. Two easy options are to install via [Mac Ports](#) or [Homebrew](#).
3. Once installed, run the following from the terminal:

```
$ sudo port install maven3
$ mvn --version
Apache Maven 3.0.5 (r01del4724cdef164cd33c7c8c2fel55faf9602da
Maven home: /opt/local/share/java/maven3
Java version: 1.7.0_51, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.9.1", arch: "x86_64", family: "mac"
```

4. Next, setup your ENV variables:

```
vi ~/.bash_profile
```

5. Add the following:

```
export M2_REPO=/Users/brent/.m2/repository
export MAVEN_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=2048m"
```

Downloading and Building OVSDB Source Code

1. There are various ways to pull the controller code and artifacts using Maven. The easiest way to avoid any float between the upstream Nexus and your local `.m2` cache is to download the controller zip. The zipped distribution is updated every 30 minutes by an upstream job. So, periodically download a fresh copy as code is committed upstream in order to stay consistent.

For information about using Maven for dependency management, see: [link](#).

```
$ wget http://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/distributions-virtualization/0.1.0/distributions-virtualization-0.1.0-osgipackage.zip
```

```
$ unzip distributions-virtualization-0.1.0-osgipackage.zip
$ cd opendaylight/
```

2. Next clone and build the OVSDB project from Gerrit:

```
$ sudo yum install git
$ git clone https://git.opendaylight.org/gerrit/ovsdb
$ cd ovsdb
$ mvn clean install -DskipTests
```

You can then import the OVSDB project into your IDE. See the following tutorials for IDE specific imports: [Importing and Developing OpenDaylight Using IntelliJ](#)

[Importing Project Into Eclipse](#)

[OpenDaylight OVSDB Developer Getting Started Youtube Video](#)

Building and Running Edited OVSDb Project Code

After you change a piece of code, for example, in the ovsdb/neutron bundle, you must build and copy that jar to the controller distribution directory where you run the controller.

```
$ cd neutron
$ mvn clean install -DskipTests
# copy the newly built jar to the controller to load
$ cp target/ovsdb.neutron*.jar /Users/roomba/opendaylight/plugins/
```

- Note: It is not necessary to stop a running controller to refresh a bundle. OSGI will automatically reload the jar/bundle for you.

For committing code back upstream, you will need to set up your ssh keys, and register them with Gerrit. See the following tutorial: [OpenDaylight Getting Started with Gerrit](#)

Generate SSH Keys for Git

A brief example of generating the SSH key is the following:

```
roomba$ cd ~/.ssh
roomba$ ls
known_hosts
$ ssh-keygen -t rsa -C "roomba@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/roomba/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/roomba/.ssh/id_rsa.
Your public key has been saved in /Users/roomba/.ssh/id_rsa.pub.
```

The key fingerprint is:

<Super Top Secret Roomba's eyes only> roomba@gmail.com

The key's randomart image is:

```
+--[ RSA 2048 ]-----+
|  E+=*+..  .  |
|    ++oo.  .  |
|   ..+. o   .  |
|  . + .o.  .  |
| + o S.    .  |
|  o ..       |
|              |
+-----+
$ ssh-add id_rsa
Identity added: id_rsa (id_rsa)
```

```
$ pbcopy < ~/.ssh/id_rsa.pub
# !! Copy this key to your account on Gerrit
$ ssh -p 29418 roomba@git.opendaylight.org
# Clone the OVSDb repo via ssh now
git clone ssh://roomba@git.opendaylight.org:29418/ovsdb.git
```

Installing Open vSwitch

The following will install Open vSwitch v2.x+ and the associated kernel modules.

```
$ sudo yum install openvswitch
$ sudo /sbin/service openvswitch start
```

Verify the kernel modules that were loaded with the following:

```
$ lsmod | grep openv
openvswitch      66772  0
vxlan            37238  1 openvswitch
gre              13888  1 openvswitch
libcrc32c        12603  1 openvswitch
```

[More Tutorials for Installing Open vSwitch](#)

Additional Tools, Postman, Mininet and Wireshark

The Postman, Mininet and Wireshark are some handy tools that are used in various tutorials and Dev environments:

Install Postman REST Tool

Postman is a Chrome App used for performing and sharing nerdy REST API calls with your friends!

1. [Download Postman](#)
2. After installing Postman, you can import the shared collection [found here](#)

Install Wireshark

Install Wireshark for a Fedora host not running a desktop manager and using X over SSH

1. Install RPMs

```
$ sudo yum install wireshark xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils wireshark-gnome wireshark
```

2. Disable selinux (for Dev only).

```
$ sudo sed -i 's/#X11Forwarding\ no/X11Forwarding\ yes/' /etc/ssh/sshd_config  
$ sudo systemctl restart sshd.service
```

3. You also may need to disable selinux. Edit /etc/selinux/config for a persistent disable across reboots.

```
$ sudo sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

4. Disable with setenforce for a temporary disable until the next reboot.

```
$ getenforce  
Enforcing  
$ sudo setenforce 0  
$ getenforce  
Permissive
```

5. Run Wireshark over X

```
$ ssh -X roomba@x.x.x.x
```

For example:

```
$ ssh -X roomba@192.168.1.10  
$ sudo wireshark
```

6. Start the packet capture. 7. Add a filter of tcp.port == 6653 && tcp.port == 6633 8. Right-click on a packet and "decode as" Open Flow.

Installing Mininet

1. As root, execute the following operations:

- a. Install git

```
$ sudo yum install git
```

- b. Create a user account (for example, mininet) and add it to the wheel group:

```
$ sudo useradd -d mininet  
$ sudo usermod -a -G wheel mininet
```

- c. Change the SELinux setting to permissive. It can be done temporarily with:

```
$ sudo setenforce 0
```

2. Next, login with the new account (e.g. mininet) and do the following: a. Clone the Mininet repository. git clone [git://github.com/mininet/mininet.git](https://github.com/mininet/mininet.git)

3. Install Mininet, the OpenFlow reference implementation, and Open vSwitch:

```
$ sudo mininet/util/install.sh -f
```

4. Enable and start openvswitch:

```
$ sudo systemctl enable openvswitch  
$ sudo systemctl start openvswitch
```

5. Test the mininet installation. sudo mn --test pingall

More information on Mininet is available at: mininet.org