ONOS Single Instance Installation

James Won-Ki Hong, Jian Li, Seyeon Jeong

Dept. of Computer Science & Engineering

POSTECH

http://dpnm.postech.ac.kr/~jwkhong

jwkhong@postech.ac.kr

Single ONOS Instance Installation (1/8)

❖ Install Dependencies

- Install git

```
# yum install -y git
# git config --global http.sslverify false
```

- Install Oracle Java 8 JDK
  - Need to uninstall prior version of Java (e.g., Java 6 or Java 7)
  - Download Java 8 JDK binary

```
# cd /usr/local/src
# wget --header "Cookie: oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u131-
b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz
# tar xvfz jdk-8u131-linux-x64.tar.gz -C ./
# mv jdk1.8.0_131 /usr/local/java
```

  - Install Java 8 JDK

```
# alternatives --install /usr/bin/java java /usr/local/java/bin/java 2
# alternatives --config java
# alternatives --install /usr/bin/jar jar /usr/local/java/bin/jar 2
# alternatives --install /usr/bin/javac javac /usr/local/java/bin/javac 2
# alternatives --set jar /usr/local/java/bin/jar
# alternatives --set javac /usr/local/java/bin/javac
```

Single ONOS Instance Installation (2/8)

❖ Install Dependencies

- Download and install maven (optional)

```
# cd /usr/local/src
# wget https://archive.apache.org/dist/maven/maven-
3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz
# tar xvfz apache-maven-3.3.3-bin.tar.gz -C ./
# mv apache-maven-3.3.3 /usr/local/maven
# vi /etc/profile
export MAVEN_HOME=/usr/local/maven
PATH=$PATH:$MAVEN_HOME/bin
# source /etc/profile
```

- Configure environment variables
  - Open /etc/environment, and input following content

```
JAVA_HOME=/usr/local/java
JRE_HOME=/usr/local/java/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

  - Source the profile

```
# source /etc/environment
```

Single ONOS Instance Installation (3/8)

❖ Install Dependencies

- Install karaf 3.x (e.g., 3.0.8) (if you build with BUCK, this is optional)

```
$ mkdir -p /home/sdn/apps && cd /home/sdn/apps
$ wget http://ftp.vim.org/pub/internet/apache/karaf/3.0.8/apache-karaf-3.0.8.tar.gz
$ tar xvfz apache-karaf-3.0.8.tar.gz -C ./
```

- Install other dependencies

```
# yum install -y zip unzip
```

- Add port 8101 and 8181 into white list using firewalld
  - 8101: used by karaf to ssh to remote host
  - 8181: used for providing web service by jetty

```
# firewall-cmd --permanent --zone=public --add-port=8181/tcp
# firewall-cmd --permanent --zone=public --add-port=8101/tcp
# firewall-cmd --reload
```

Single ONOS Instance Installation (4/8)

❖ Install ONOS

- Download ONOS source
  - Use git to clone ONOS repository

```
$ cd ~
$ git clone https://gerrit.onosproject.org/onos
```

- Configure ONOS environment variables
  - Add following content into ~/.bash_profile

```
export ONOS_ROOT=~/onos
export KARAF_ROOT=~/apps/apache-karaf-3.0.8
export M2_REPO=~/.m2/repository
. ~/onos/tools/dev/bash_profile
export KARAF_TAR=~/apps/apache-karaf-3.0.8.tar.gz
```

  - Source the .bash_profile

```
$ source ~/.bash_profile
$ env | grep ONOS
...
ONOS_APPS=drivers,openflow,fwd,proxyarp,mobility
ONOS_CELL=local
...
```

Single ONOS Instance Installation (5/8)

❖ Install ONOS

- Branching
  - Checkout corresponding ONOS branch
  - In this example, we will checkout ONOS v1.9.x

```
$ cd ~/onos
$ git branch
* master
$ git checkout onos-1.9
Branch onos-1.9 set up to track remote branch onos-1.9 from origin.
Switched to a new branch 'onos-1.9'
$ git branch
  master
* onos-1.9
```

- Two ways of source builds
  - BUCK builds
    - New Java build tool, introduced by Facebook
    - Very fast, provide incremental builds
    - Highly recommended to use, default build tool in ONOS
  - Maven builds
    - Legacy Java build tool
    - Very slow, but compatible with most of existing Java dependencies

Single ONOS Instance Installation (6/8)

❖ Install ONOS

- BUCK builds (option #1, highly recommended)
  - This will automatically download BUCK binary, and start to build the sources
  - Only compile all Java sources
  - “onos-buck test” needs to be issued for running unit tests

```
$ onos-buck build onos
Archive:  cache/buck-v2016.12.02.01.zip
  inflating: buck
 extracting: .buck_version
   creating: plugins/
  inflating: plugins/onos.jar
  inflating: plugins/yang.jar
Successfully updated Buck in /home/sdn/onos/bin/buck to buck-v2016.12.02.01.zip

Not using buckd because NO_BUCKD is set.
[-] PROCESSING BUCK FILES...FINISHED 8.7s [100%] 👉  New buck daemon
[+] DOWNLOADING... (0.00 B/S, TOTAL: 0.00 B, 0 Artifacts)
[+] BUILDING...1m25.3s [10%] (3/813 JOBS, 3 UPDATED, 3 [0.4%] CACHE MISS)
```

- Run ONOS using BUCK

```
$ o
$ onos-buck run onos-local -- clean debug
```

Single ONOS Instance Installation (7/8)

❖ Install ONOS

- Setup BUCK path

```
$ mkdir -p /home/sdn/bin && cd /home/sdn/bin
$ ln -s ~/onos/bin/buck buck
```

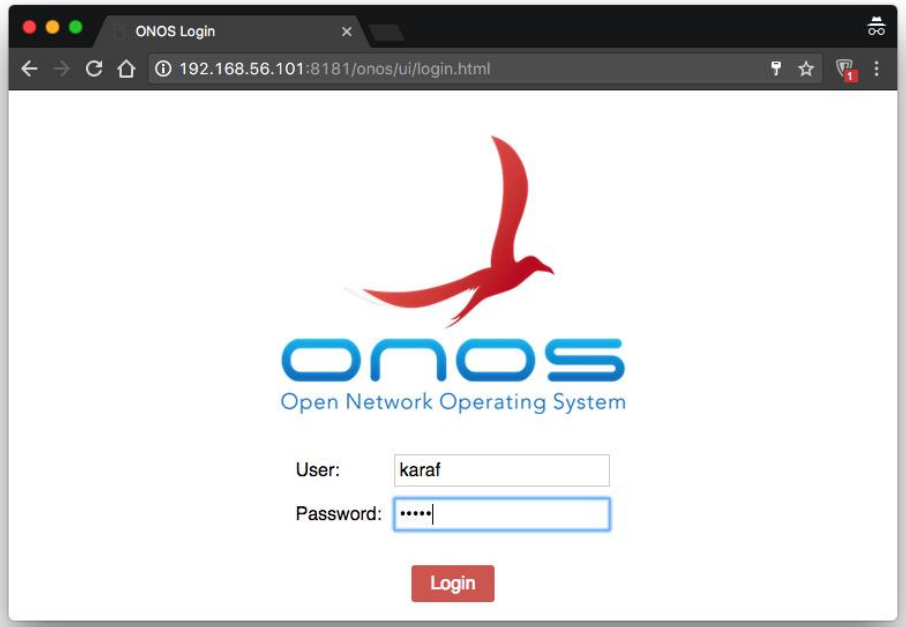
- Maven builds (option #2)
  - Note that maven should be pre-installed
  - Compile → run unit tests

```
$ mvn clean install
Or
$ mci
```

Single ONOS Instance Installation (8/8)

❖ Access ONOS GUI

- http://{ONOS_IP}:8181/onos/ui
  - ID: karaf, PW: karaf



❖ Access ONOS Shell

```
$ onos localhost
Welcome to Open Network Operating System (ONOS)!

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org


Come help out! Find out how at: ...


Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or
'logout' to shutdown ONOS.


onos>
```

Supplementary Tools Installation

❖ Mininet Installation

- Download mininet source from github repository

```
# git clone git://github.com/mininet/mininet.git
```

- Edit installation script which located in the path of mininet/util/install.sh

```
test -e /etc/centos-release && DIST="CentOS"
if [ "$DIST" = "CentOS" ]; then
    install='sudo yum -y install'
    remove='sudo yum -y erase'
    pkginst='sudo rpm -ivh'
    # Prereqs for this script
    if ! which lsb_release &> /dev/null; then
        $install redhat-lsb-core
    fi
fi
```

  - Add CentOS into following line

```
if ! echo $DIST | egrep 'Ubuntu|Debian|Fedora|CentOS|RedHatEnterpriseServer|
SUSE LINUX'; then
```

- Install mininet

```
# mininet/util/install.sh -nf
```

Supplementary Tools Installation

❖ OpenvSwitch Installation

- Get the dependencies needed to build OVS

```
# yum -y install gcc make python-devel openssl-devel kernel-devel
graphviz \ kernel-debug-devel autoconf automake rpm-build redhat-rpm-
config \ libtool
```

- Build RPM binary from OVS source

```
# mkdir -p ~/rpmbuild/SOURCES/
# cd ~/rpmbuild/SOURCES/
# wget http://openvswitch.org/releases/openvswitch-2.5.0.tar.gz
# tar zxvf openvswitch-2.5.0.tar.gz
# cd openvswitch-2.5.0
# rpmbuild -bb --without check rhel/openvswitch.spec
# rpm -ivh --nodeps ~/rpmbuild/RPMS/x86_64/openvswitch*.rpm
```

- Launch OVS and check OVS version

```
# systemctl start openvswitch
# ovs-vsctl show
1d0e71af-d5ff-4c7b-8224-2804849824c3
    ovs_version: "2.5.0"
```

Test Single ONOS Instance

❖ Construct Data Plane

- Generate tree topology with 2 levels and 3 leaves
  - Make sure that ONOS has been launched before starts mininet

```
$ sudo mn --topo tree,2,3 --controller=remote,ip=127.0.0.1,port=6633
```

- Make sure proxyARP and reactive forwarding app are activated

```
onos> app activate org.onosproject.proxyarp
onos> app activate org.onosproject.fwd

onos> apps -s
...
*  19 org.onosproject.fwd              1.9.1    Reactive forwarding ...
*  20 org.onosproject.proxyarp         1.9.1    Proxy ARP/NDP application
...
```

- Test reachability
  - Send ICMP requests to all hosts

```
mininet> pingall
```

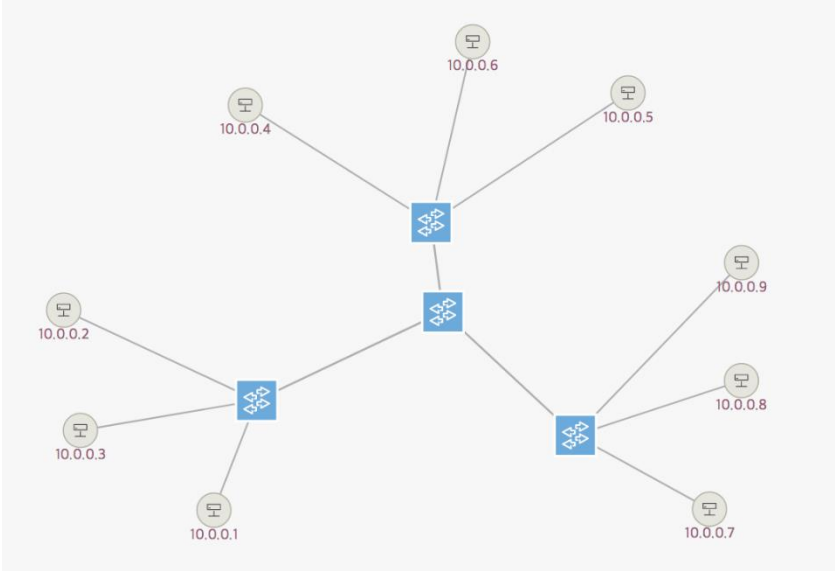
Test Single ONOS Instance

❖ Inter-connect ONOS with Data Plane

- Result of test reachability
  - 8 x 9 = 72 ICMP requests and replies

```
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
```

- Check ONOS dashboard
  - Press “H” key to view all switches with hosts connected



DEMO

```
+-----------------+
[sdn@localhost ~]$ ssh-copy-id localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 2e:e6:3d:20:54:10:5f:04:4d:ea:16:e3:49:87:2a:94.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
 that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
sdn@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

[sdn@localhost ~]$ exit
logout
[root@localhost ~]# vi /etc/sudoers
[root@localhost ~]#
```