**ONOS 1.12 Installation Guide**

Stephen Blystone ([smb032100@utdallas.edu](mailto:smb032100@utdallas.edu))

Document Version 1.4

Last Updated: Feb 13, 2018

Table of Contents

[**Revision History** 3](#_Toc506329961)

[**(OPTION 1) VirtualBox OVA File & Distributed ONOS Tutorial** 3](#_Toc506329962)

[(OPTION 1) Download OVA file with ONOS 1.12 and Mininet 3](#_Toc506329963)

[**(REQUIRED STEPS FOR OPTIONS 2 & 3) Setting Up VM for ONOS** 3](#_Toc506329964)

[**OPTION 2 (Get an ONOS Controller up and running):** 5](#_Toc506329965)

[(OPTION 2) Launch ONOS as a Controller 5](#_Toc506329966)

[**OPTION 3 (If you want to dig into the code and see how ONOS works):** 6](#_Toc506329967)

[(OPTION 3) Build ONOS For Development in IntelliJ 6](#_Toc506329968)

[(OPTION 3 Continued) LAUNCHING ONOS: 6](#_Toc506329969)

[(OPTION 3 Continued) ONOS DEVELOPMENT IN INTELLIJ: 7](#_Toc506329970)

[**Using ONOS as Controller with Mininet** 8](#_Toc506329971)

[**APPENDIX** 9](#_Toc506329972)

[Installing Mininet into same VM as ONOS 9](#_Toc506329973)

[Further Resources 9](#_Toc506329974)

[References 9](#_Toc506329975)

[Troubleshooting 10](#_Toc506329976)

# **Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 1.0 | 2/9/2018 | Initial Version |
| 1.1 | 2/9/2018 | Add section for using ONOS as controller in mininet |
| 1.2 | 2/11/2018 | Add OVA file option, and move Creating and Deploying ONOS Applications section to a separate document. |
| 1.3 | 2/11/2018 | Corrected git clone command to ensure we get version 1.12 (HEAD currently points to 1.13-SNAPSHOT). |
| 1.4 | 2/13/2018 | Separate section for Launching ONOS to make it easier to find quickly. |
|  |  |  |

# **(OPTION 1) VirtualBox OVA File & Distributed ONOS Tutorial**

## (OPTION 1) Download OVA file with ONOS 1.12 and Mininet

1. Download OVA file from the following link and Import into Virtualbox

<http://onlab.vicci.org/onos/vm/onos-tutorial-1.12.0.ova>

1. ONOS Tutorial found on: <https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial>
2. Stop here for OPTION 1.

# **(REQUIRED STEPS FOR OPTIONS 2 & 3) Setting Up VM for ONOS**

1. Download VirtualBox-5.2.6
2. Download Ubuntu 16.04 Desktop (64-bit) ISO
   1. NOTE: You MUST use 64-bit for ONOS to work.
3. Set up VM
   1. Click New
   2. Name: “ONOS 1.12”
   3. Type: Linux
   4. Version: Ubuntu (64-bit)
      1. NOTE: You MUST use 64-bit for ONOS to work.
   5. Memory Size: 2048 MB
   6. Hard Disk File Type: VDI, Dynamically allocated, 10GB or 20GB (See note below)
      1. Can probably get by with 10GB if just launching controller (Option 2)
      2. 20GB needed for downloading the source code and building yourself (Option 3)
   7. Create VM
4. Modify VM Settings
   1. Click on VM and click Settings
   2. System Menu
      1. Motherboard Tab:
         1. Base Memory: 2048 MB
         2. Check Enable I/O APIC
      2. Processor Tab:
         1. Processors: 2
   3. Storage Menu
      1. Add Optical Drive – select Ubuntu
   4. Network Menu
      1. Bridged Adapter
5. Launch VM and follow Ubuntu set-up
   1. Hostname: ONOS1-12
   2. Username: onos
   3. Password: rocks
6. System Settings to Change after Ubuntu installed and you are logged in.
   1. Brightness & Lock
      1. Turn screen off when inactive for: Never
      2. Lock: Off
   2. Security & Privacy
      1. Diagnostics
         1. Uncheck:
            1. Send error reports to Canonical
            2. Send occasional system information to Canonical
   3. Details
      1. Install Updates (on Overview tab)
         1. A lot of the updates are needed. Some are not. Can either install all of them or manually select the important ones.
7. Open Terminal
   1. Add sdn user

sudo adduser sdn -–system -–group

* 1. Run Java8 commands and curl install commands found on <https://wiki.onosproject.org/display/ONOS/Requirements>
     1. Java8 commands:

sudo apt-get install software-properties-common -y && \

sudo add-apt-repository ppa:webupd8team/java -y && \

sudo apt-get update && \

echo "oracle-java8-installer shared/accepted-oracle-license-v1-1 select true" | sudo debconf-set-selections && \

sudo apt-get install oracle-java8-installer oracle-java8-set-default -y

* + 1. Curl commands:

sudo apt-get install curl

1. Add JAVA\_HOME to your .profile file and to PATH

vi ~/.profile

JAVA\_HOME=/usr/lib/jvm/java-8-oracle

PATH=$PATH:$JAVA\_HOME

1. Log out and back in to enable the changes.

# **OPTION 2 (Get an ONOS Controller up and running):**

## (OPTION 2) Launch ONOS as a Controller

1. Follow instructions on <https://wiki.onosproject.org/display/ONOS/Installing+on+a+single+machine>

cd /opt

sudo wget -c <http://repo1.maven.org/maven2/org/onosproject/onos-releases/onos-1.12.0/onos-1.12.0.tar.gz>

sudo tar xzf onos-1.12.0.tar.gz

sudo mv onos-1.12.0 onos

/opt/onos/bin/onos-service start

1. Run commands to run ONOS as a service: <https://wiki.onosproject.org/display/ONOS/Running+ONOS+as+a+service>

sudo cp /opt/onos/init/onos.initd /etc/init.d/onos

sudo cp /opt/onos/init/onos.service /etc/systemd/system/

sudo systemctl daemon-reload

sudo systemctl enable onos

1. Edit /opt/onos/options file per instructions: <https://wiki.onosproject.org/display/ONOS/Running+ONOS+as+a+service>
   1. ONOS\_USER=sdn
      1. This is the user we created earlier.
2. Start the ONOS Service

sudo systemctl {start|stop|status|restart} onos.service

1. Stop here for OPTION 2.

# **OPTION 3 (If you want to dig into the code and see how ONOS works):**

## (OPTION 3) Build ONOS For Development in IntelliJ

1. Install git and Maven

sudo apt-get install git maven

1. Clone the ONOS Project (version 1.12) to your directory
   1. Run command, specifying you want version 1.12:

git clone <http://gerrit.onosproject.org/onos> -b onos-1.12

1. Change directory to the onos directory

cd ~/onos

1. Source the Bash Profile to augment the environment with path configuration, and shell aliases and shortcuts.

source tools/dev/bash\_profile

1. Build ONOS using onos-buck

onos-buck build onos

1. Edit your .profile file

vi ~/.profile

export ONOS\_ROOT=~/onos

source $ONOS\_ROOT/tools/dev/bash\_profile

* 1. Log out and back in to enable the changes.

1. Run Unit tests/Integration tests/etc.

cd ~/onos

onos-buck test

## (OPTION 3 Continued) LAUNCHING ONOS:

1. Run ONOS locally. This creates a local ONOS installation in the User’s tmp directory and start the ONOS server in the background while displaying the ONOS log in the foreground.

cd ~/onos

onos-buck run onos-local

1. To connect to ONOS command-line console, open a new terminal window:

cd ~/onos

onos localhost

1. To connect to ONOS GUI:
   1. From terminal (ONLY WORKS ON MAC, see note below):

cd ~/onos

onos-gui localhost

NOTE: The onos-gui shell script calls the “open” command which only works on Mac, and not on Ubuntu. I had to use the browser window.

* 1. From browser (preferred browser is Chrome, although Firefox also works in my basic tests).

**See Note:** “The GUI has been developed to work on Google Chrome. The GUI has been tested on Safari and Firefox and minor compatibility adjustments have been made; these and other browsers may work, but have not been extensively tested, and are not actively supported, at this time.” (<https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI>)

* 1. Browser URL:

<http://localhost:8181/onos/ui>

* 1. Default Username is: onos
  2. Default Password is: rocks

## (OPTION 3 Continued) ONOS DEVELOPMENT IN INTELLIJ:

1. Install IntelliJ (<https://blog.jetbrains.com/idea/2017/11/install-intellij-idea-with-snaps/>)

sudo snap install intellij-idea-community --classic --edge

1. To import ONOS project into IntelliJ IDE

cd ~/onos

onos-buck project

1. To run IntelliJ IDE from command-line:

snap run intellij-idea-community &

1. Open root pom.xml file in IntelliJ IDE as Project
   1. Click Open
   2. Select file /home/onos/onos/pom.xml
   3. Open as Project
   4. Open Existing Project
2. If a message pops up after a few minutes saying “Non-managed pom.xml file found”:
   1. Add as Maven Project
3. Ensure copyright is on all files
   1. Follow instructions for “Importing the Copyright header” on: <https://wiki.onosproject.org/display/ONOS/Importing+ONOS+projects+into+IntelliJ+IDEA>

# **Using ONOS as Controller with Mininet**

1. Launch mininet VM
2. In ONOS VM, launch a terminal and run “ifconfig” to see IP address
3. In mininet VM, run:

sudo mn --controller=remote,ip=10.28.34.39,port=6633

REPLACE 10.28.34.39 with your IP address for the ONOS controller.

1. In mininet VM, have h1 ping h2

h1 ping h2

1. In ONOS VM, launch web GUI. <http://localhost:8181/onos/ui/>
2. On Menu, select Devices to see Open vSwitch
3. On Menu, select Hosts to see h1 and h2

# **APPENDIX**

## Installing Mininet into same VM as ONOS

1. Install Mininet (‘mn’ and the Python API):

sudo apt-get install mininet

## Further Resources

1. Tutorials

<https://wiki.onosproject.org/display/ONOS/Tutorials>

1. Guides

<https://wiki.onosproject.org/display/ONOS/Guides>

1. JavaDocs

<http://api.onosproject.org/1.12.0/>

1. FAQ

<https://wiki.onosproject.org/display/ONOS/FAQs>

1. ONOS Karaf CLI Command reference

<http://sdnhub.org/tutorials/onos/>

## References

Some of the Links used (in addition to what was mentioned above), **many are outdated** but have some useful information on them:

<https://wiki.onosproject.org/display/ONOS/ONOS+from+Scratch>

<https://wiki.onosproject.org/display/ONOS/Getting+started+developing+with+ONOS>

<https://www.youtube.com/watch?v=wEMVmbsNyS4>

<https://wiki.onosproject.org/display/test/Building+ONOS>

<https://wiki.onosproject.org/display/ONOS/Installing+on+a+single+machine>

<https://wiki.onosproject.org/display/ONOS/Running+ONOS+as+a+service>

<https://wiki.onosproject.org/display/test/Environment+setup+with+Mininet+and+onos.py>

<https://blog.jetbrains.com/idea/2017/11/install-intellij-idea-with-snaps/>

## Troubleshooting

* Windows 10 blue screen’s when launching VM
  + Turn off Hyper-V on Windows
    - Start Menu
    - “Turn Windows Features on or off”
    - Uncheck Hyper-V
    - Restart computer
    - Try again
* Virtualbox error mentioning VT-x
  + Go to BIOS on computer and ensure virtualization features are enabled