SDN Research Team

ONOS application installation guide

HTTP attack detector & mitigator

Index

Introduction

This document will explain the downloading, compiling and deploying process of an ONOS application, to later on use it to detect & mitigate several attacks running in a simulated topology. It is important to say that a mininet virtual machine with ONOS running through docker will be the starting point of this guide.

Prerequisites

The following list is a bunch of Software the host machine will require in order to ease the process; all of the downloaded executables are installers, and it is recommended to follow a default installation (next->next->...->install) :

- An SSH client (putty recommended): https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- SCP application (WinSCP recommended): https://winscp.net/eng/download.php

Required Software installation

Through this section the will guide show an easy way to install the essential Software the virtual machine requires to run the ONOS application.

Curl

The Curl (Command-line tool for transferring data with URLs) can be easily installed by typing the command:

```
sudo apt-get install curl
```

Maven

1. First, the source files need to be downloaded from the Virtual machine console:

```
wget
http://apache.mirrors.lucidnetworks.net/maven/maven-3/3.3.9/binaries/
apache-maven-3.3.9-bin.tar.gz
```

2. Create the folder where the maven files will be uncompressed:

```
 sudo mkdir -p /usr/local/apache-maven
```

3. Move the downloaded .tar to the created directory:

```
sudo mv apache-maven-3.3.9-bin.tar.gz /usr/local/apache-maven
```

4. Navigate to the created directory and uncompress the downloaded .tar file:

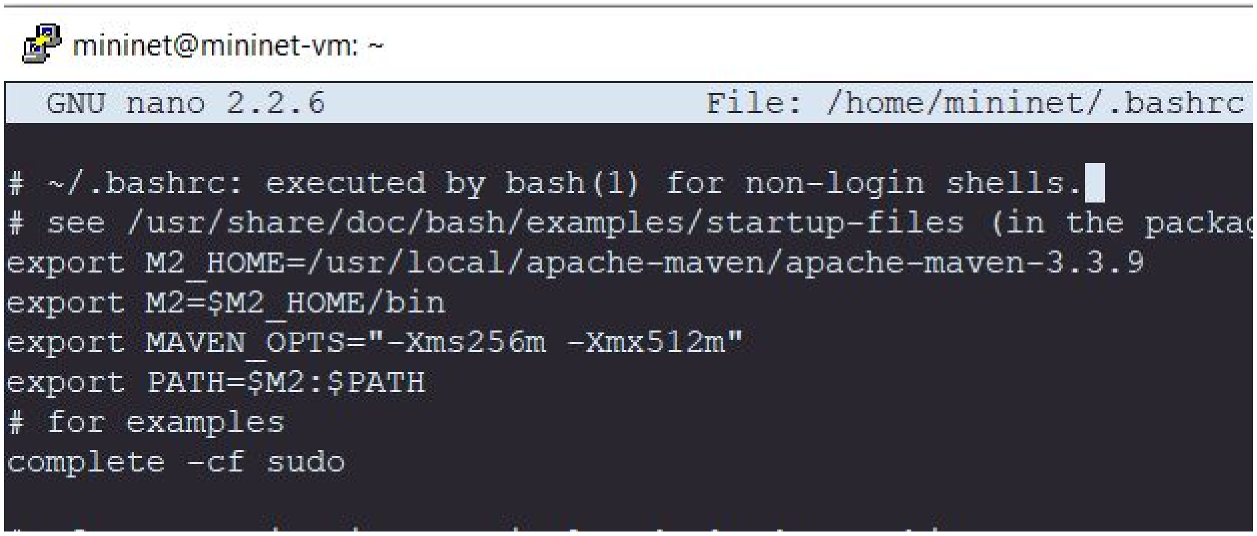
```
cd /usr/local/apache-maven

sudo tar -xzvf apache-maven-3.3.9-bin.tar.gz
```

5. Add the environment variables to the system by accessing .bashrc through nano editor:

```
nano ~/.bashrc
```

6. And copy-Paste the following lines into the first lines of the file so that results similar to the figure 1:

```
export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.9

export M2=$M2_HOME/bin

export MAVEN_OPTS="-Xms256m -Xmx512m"

export PATH=$M2:$PATH
```



mininet@mininet-vm: ~

```
  GNU nano 2.2.6                    File: /home/mininet/.bashrc

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the packag
export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.9
export M2=$M2_HOME/bin
export MAVEN_OPTS="-Xms256m -Xmx512m"
export PATH=$M2:$PATH
# for examples
complete -cf sudo
```

7. After quitting & saving the file pulsating ctrl+x and agreeing with the message, the user can apply the changes by running this command, thus concluding the maven installation:

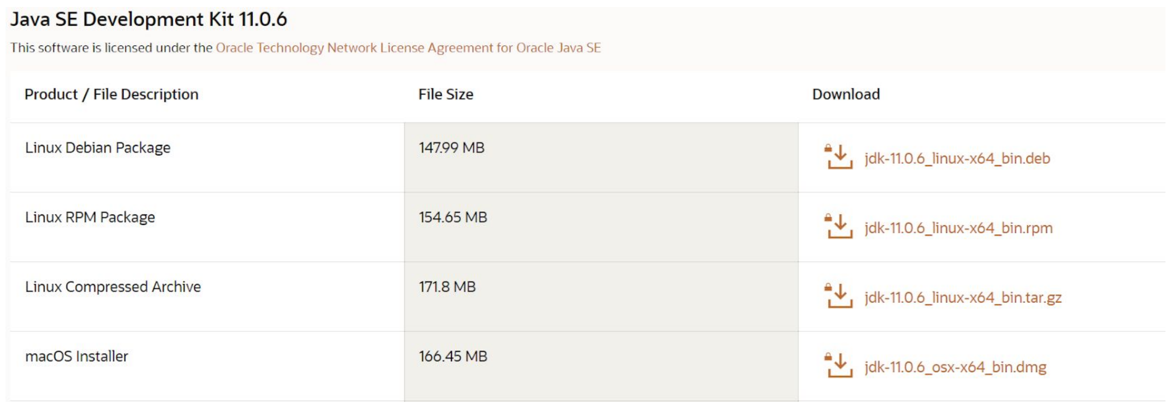
```
. ~/.bashrc
```

Java Development Kit

The ONOS app will be compiled using the JDK 11, unfortunately, at the time of writing this version cannot be found in the apt repositories, so the compressed files need to be downloaded through the official oracle website using a web browser with a graphic user interface.

The approach this guide will take is downloading the files in the host machine, to later on transferring them into the virtual machine employing WinSCP, a graphic FTP client which uses SSH connections.

1. In the host PC, navigate to the oracle download page and localize the Java SE Development Kit 11.0.6 and click the *Linux Compressed Archive* download, it will have a .tar.gz termination:

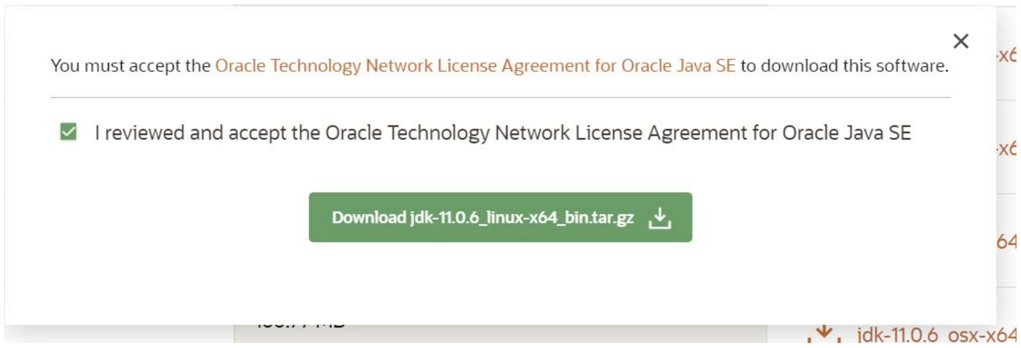


Java SE Development Kit 11.0.6

This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE

Product / File Description	File Size	Download
Linux Debian Package	147.99 MB	jdk-11.0.6_linux-x64_bin.deb
Linux RPM Package	154.65 MB	jdk-11.0.6_linux-x64_bin.rpm
Linux Compressed Archive	171.8 MB	jdk-11.0.6_linux-x64_bin.tar.gz
macOS Installer	166.45 MB	jdk-11.0.6_osx-x64_bin.dmg

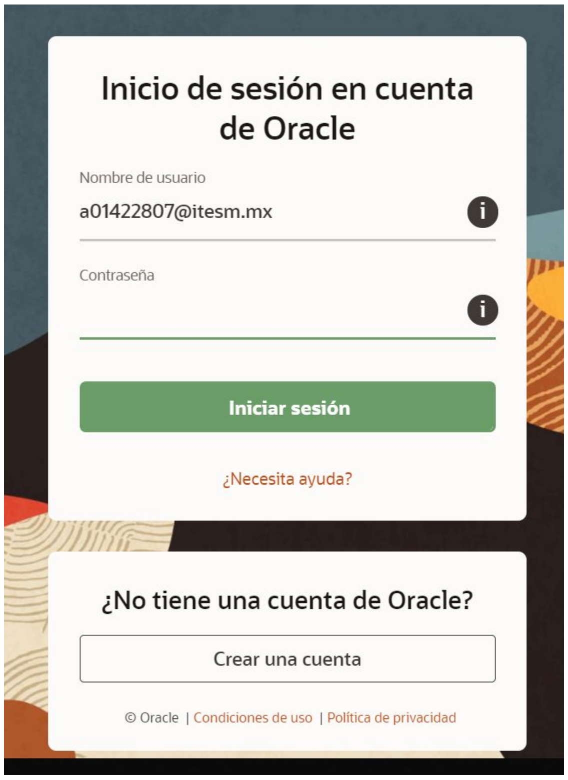
2. A prompt will be displayed to accept the license agreement; accept it and click the download button:



You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

✓ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

3. The user will need to login with an Oracle account



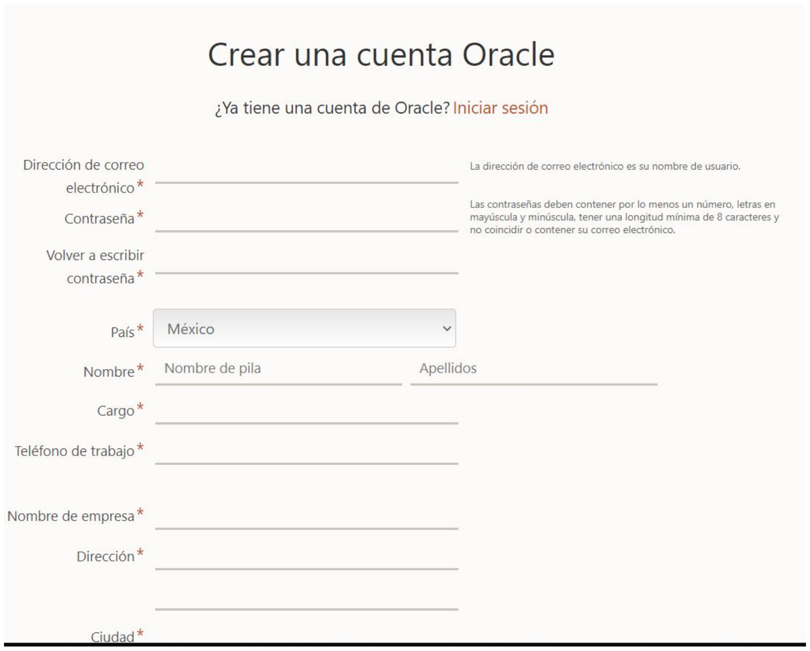
Inicio de sesión en cuenta de Oracle

Nombre de usuario: a01422807@itesm.mx

Contraseña:

Iniciar sesión
¿Necesita ayuda?
¿No tiene una cuenta de Oracle?
Crear una cuenta
© Oracle | Condiciones de uso | Política de privacidad

4. If the user doesn't have an account, simply click the Create Account Button and fill the blanks (it may be necessary to confirm the introduced email afterwards)

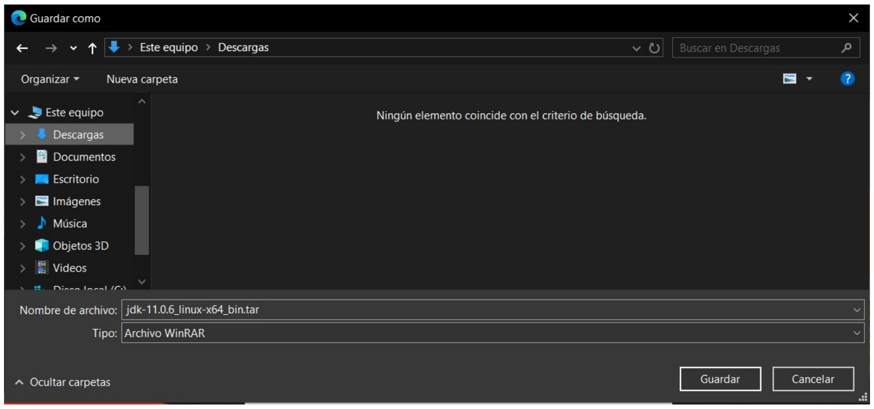


Crear una cuenta Oracle

¿Ya tiene una cuenta de Oracle? Iniciar sesión

Dirección de correo electrónico*		La dirección de correo electrónico es su nombre de usuario.
Contraseña*		Las contraseñas deben contener por lo menos un número, letras en mayúscula y minúscula, tener una longitud mínima de 8 caracteres y no coincidir o contener su correo electrónico.
Volver a escribir contraseña*		
País*	México	
Nombre*	Nombre de pila	Apellidos
Cargo*		
Teléfono de trabajo*		
Nombre de empresa*		
Dirección*		
Ciudad*		

5. When the login is successful, a new window will appear to download the compressed archives. It is recommended to save the file into an accessible folder.



Guardar como

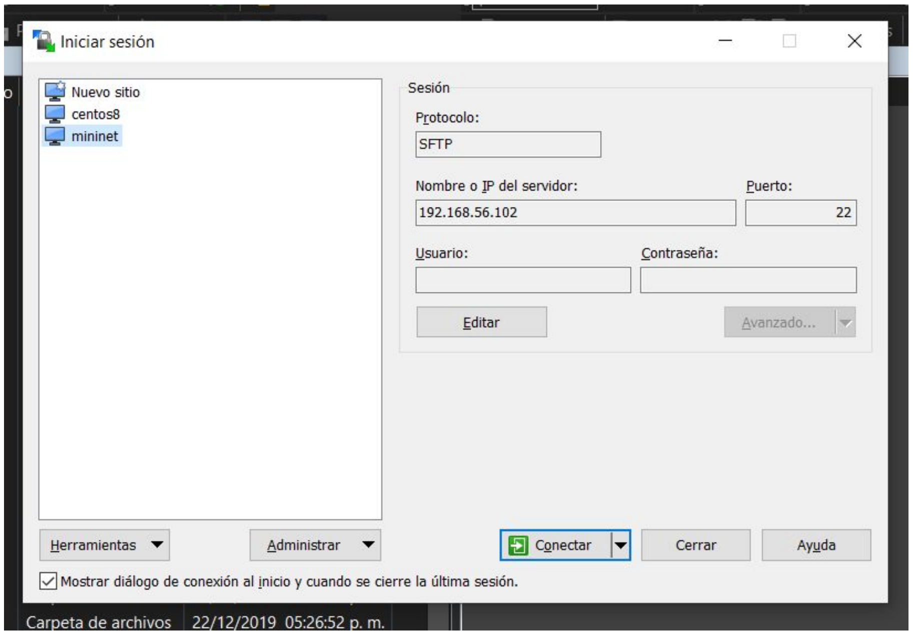
Este equipo > Descargas

Ningún elemento coincide con el criterio de búsqueda.

Nombre de archivo: jdk-11.0.6_linux-x64_bin.tar

Tipo: Archivo WinRAR

6. Open WinSCP, the application will be launched with a login prompt containing all saved SSH connections. Select the mininet option and click the Connect button.


Iniciar sesión

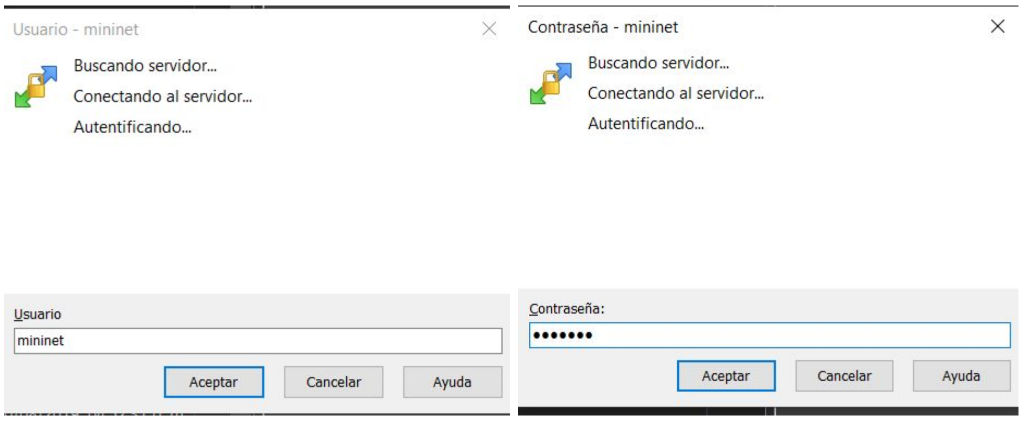
Nuevo sitio
centos8
mininet

Sesión
Protocolo: SFTP
Nombre o IP del servidor: 192.168.56.102
Puerto: 22
Usuario:
Contraseña:

☑ Mostrar diálogo de conexión al inicio y cuando se cierre la última sesión.

Carpeta de archivos 22/12/2019 05:26:52 p. m.

7. Login as you would do it normally in the Virtual Machine, introducing the user and password.


Usuario - mininet

Buscando servidor...
Conectando al servidor...
Autentificando...

Usuario: mininet

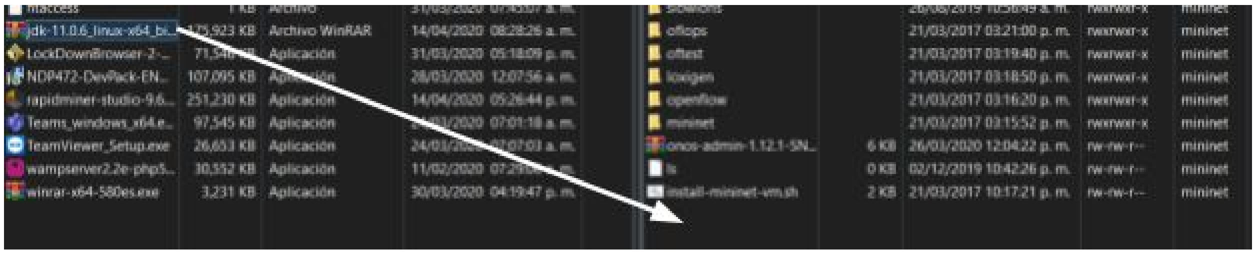
Contraseña - mininet

Buscando servidor...
Conectando al servidor...
Autentificando...

Contraseña: •••••••

8. A splitted window will appear. On the left side is the host machine directories, while in the right side of the window will be the Linux home directory. Localize the downloaded JDK compressed file in the host machine side, and drag & drop it into

the Linux user’s home directory:



.htaccess	1 KB	Archivo	31/03/2020 07:43:07 a. m.
jdk-11.0.6_linux-x64_bi...	173,923 KB	Archivo WinRAR	14/04/2020 08:28:26 a. m.
LockDownBrowser-2-...	71,54[illegible]	Aplicación	31/03/2020 05:18:09 p. m.
NDP472-DevPack-EN...	107,095 KB	[illegible]	28/03/2020 12:07:56 a. m.
rapidminer-studio-9.6...	251,230 KB	Aplicación	14/04/2020 05:26:44 p. m.
Teams_windows_x64.e...	97,545 KB	Aplicación	[illegible]/03/2020 07:01:18 a. m.
TeamViewer_Setup.exe	26,653 KB	Aplicación	24/03/20[illegible] 07:07:03 a. m.
wampserver2.2e-php5...	30,552 KB	Aplicación	11/02/2020 07:2[illegible] m.
winrar-x64-580es.exe	3,231 KB	Aplicación	30/03/2020 04:19:47 p. m.

[illegible]		26/08/2019 10:56:49 a. m.	rwxrwxr-x	mininet
oflops		21/03/2017 03:21:00 p. m.	rwxrwxr-x	mininet
oftest		21/03/2017 03:19:40 p. m.	rwxrwxr-x	mininet
loxigen		21/03/2017 03:18:50 p. m.	rwxrwxr-x	mininet
openflow		21/03/2017 03:16:20 p. m.	rwxrwxr-x	mininet
mininet		21/03/2017 03:15:52 p. m.	rwxrwxr-x	mininet
onos-admin-1.12.1-SN...	6 KB	26/03/2020 12:04:22 p. m.	rw-rw-r--	mininet
ls	0 KB	02/12/2019 10:42:26 p. m.	rw-rw-r--	mininet
install-mininet-vm.sh	2 KB	21/03/2017 10:17:21 p. m.	rw-rw-r--	mininet

Although the file will be moving to an external directory, the reason to drop the compressed file in the home directory is the need of permissions to access those external directories from WinSCP. When the file transfer is finished, WinSCP can be closed.

9. Open a normal SSH connection with the client of preference, when logged in, create the folder where the installation will take place with the following command:

```
sudo mkdir /var/cache/oracle-jdk11-installer-local/
```

10. Move the compressed JDK files to the recently created folder, navigate to it and uncompress them:

sudo mv jdk-11.0.6_linux-x64_bin.tar.gz /var/cache/oracle-jdk11-installer-local/

```
cd /var/cache/oracle-jdk11-installer-local/
```

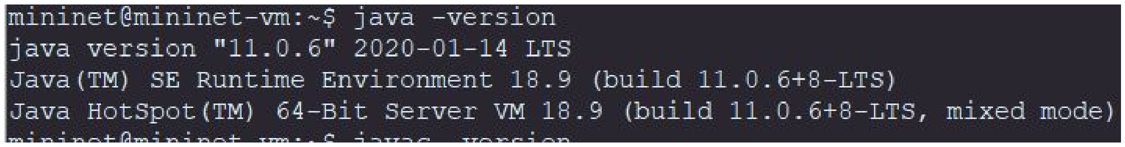
```
sudo tar xvzf jdk-11.0.6_linux-x64_bin.tar.gz
```

11. Finally, perform a local install typing the next command:

```
sudo apt-get install oracle-java11-installer-local
```

12. The java version can be seen with:

```
java -version
```



```
mininet@mininet-vm:~$ java -version
java version "11.0.6" 2020-01-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.6+8-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.6+8-LTS, mixed mode)
```

ONOS Admin Tools

1. The onos admin tools can be downloaded through the oficial google drive link doing typing a curl command in the user's home directory:

```
curl
https://drive.google.com/file/d/1E-OqKNFuzSa_iydN04tNKb7kz9ozAb7z/vie
w
```

2. It can be decompressed safely using the tar command, to later on move the new generated folder to the root directory :

```
tar xvzf onos-admin-1.12.1-SNAPSHOT.tar.gz

sudo mv onos-admin-1.12.1-SNAPSHOT /
```

3. Adding the path isn't necessary, but it will ease the app compiling & installing process:

```
PATH=$PATH:/onos-admin-1.12.1-SNAPSHOT
```

Note: The previous command only saves the PATH variable temporarily and will be discarded once the VM had been restarted. Nevertheless, it is considered a great way to make the installation process more legible & easy to replicate.

Download & install HTTPDetector

This application aims to receive flows from the switches and call the jar file located in the root folder of the container, to later obtain a response regarding the type of flow and take appropriate measures if it is a malicious flow. On the other hand, the process-able.jar loads the trained models to classify the given flow.

In this section the http detector will be downloaded from the git repository, so it is not necessary to create an entire ONOS Project from scratch.

1. Clone the repository in a convenient directory:

```
git clone https://github.com/jatj/httpDetector.git
```

Note: If desired, the updated and current project can be downloaded by changing the git url and branch, but the testing step of this section (step 5) will be postponed because auxiliar files are required in order to ensure functionality:

- Jar file classifier:

```
git clone -b process-able https://github.com/A01422807/httpDetector.git
```

- Web service classifier:

```
git clone -b http https://github.com/A01422807/httpDetector.git
```

2. The process will create the httpDetector maven project, enter this folder and compile the project with these commands:

```
cd httpDetector/

mvn clean install
```

3. Now the project is compiled, start the docker container with this command:

```
sudo docker start onos
```

4. Assuming the ONOS ip address is 172.17.0.1, the app can be easily installed by accessing the generated .oar file, normally found in the target folder:

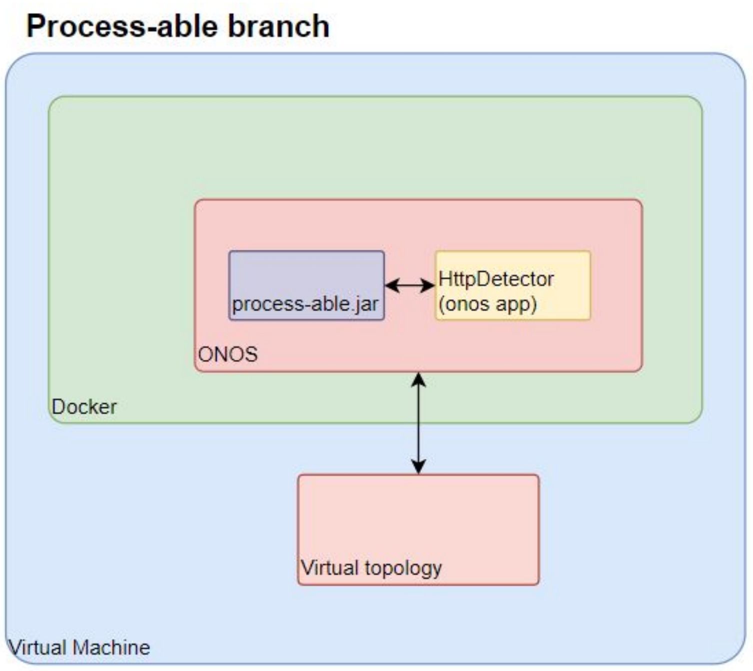
```
cd httpDetector/

onos-app 172.17.0.1 install! target/httpddosdetector-1.0-SNAPSHOT.oar
```

5. Finally, SSH to ONOS as usual and activate the app when logged-in:

```
ssh -p 8101 karaf@172.17.0.1

app activate mx.itesm.httpddosdetector

log:tail HttpDdosDetector
```

At this point, the original ONOS app tutorial has been concluded and a mininet topology can be deployed to test the application, so the guide will focus in the two current implementations from now on and assumes you've downloaded process-able branch or the http branch.

Process-Able Branch: Exporting .jar file and it's models



This section will cover the process-able branch installation. First, the user will need to download the .jar and trained models from the onos auxiliar files, and insert them in the virtual machine directories using the previous, WinSCP method.

Developer's note: For convenience purposes, the .jar file was coded & compiled using maven & JDK version 1.8, you can also do the same thing cloning the process-able git repository in the machine, and compile it using the same command seen in previous steps (it has to be introduced while in the project's directory):

```
mvn clean install
```

Due to the installed Docker version (gotten from the official apt repositories) is outdated, the command that lets the user share files between the machine and the containers tends to malfunction. Consequently, copying files directly to the container source folders was the way to go.

1. Find the ONOS container id:

```
docker inspect -f   '{{.Id}}'  onos
```

2. Change to root user mode:

```
sudo su -
```

3. We will be using the next command syntax to copy the models and jar file :

cp file-to-copy /var/lib/docker/aufs/mnt/FULL_CONTAINER_ID/PATH-NEW-FILE

Note: The blue parts are the things to change.

4. ONOS runs in an apache linux container, which runs in an Docker container simultaneously; a convenient directory was found to place the downloaded files inside ONOS: root/onos/apache-karaf-4.2.6, therefore, the final command will look similar to the following:

```
sudo cp process-able.jar
/var/lib/docker/aufs/mnt/918f25a03d3ce7f1c0c1ad9b6156848cde455fdffe35
9ba57d5b6863ed5a947c/root/onos/apache-karaf-4.2.6/
```

Note: The previous command should be taken as an example rather than a copy-paste command, that's because both the container's id and files/directories may be different. Please input this command for all the downloaded models and the jar file.

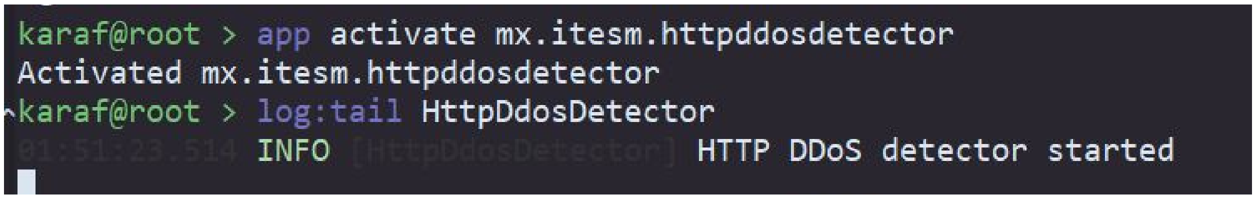
Now that everything has been configured, it is time to test the deployed application:

```
ssh -p 8101 karaf@172.17.0.1
```

```
app activate mx.itesm.httpddosdetector
```

```
log:tail HttpDdosDetector
```

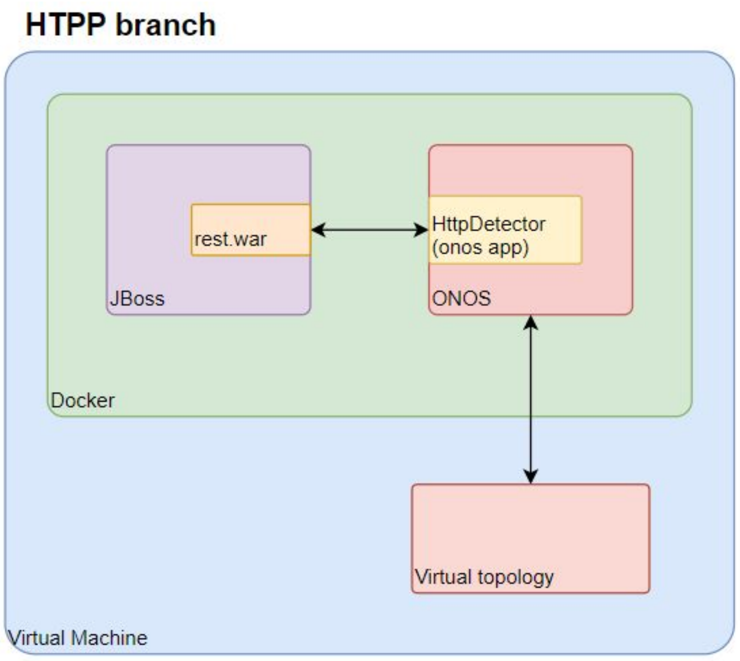
Upon the log activation, there should be a message saying the component has started correctly:


```
karaf@root > app activate mx.itesm.httpddosdetector
Activated mx.itesm.httpddosdetector
karaf@root > log:tail HttpDdosDetector
01:51:23.514 INFO [HttpDdosDetector] HTTP DDoS detector started
```

Finally, a topology deployment can be achieved by typing the following command in another SSH session:

```
sudo mn --controller remote,ip=172.17.0.1
```

Http branch: Deploying web service

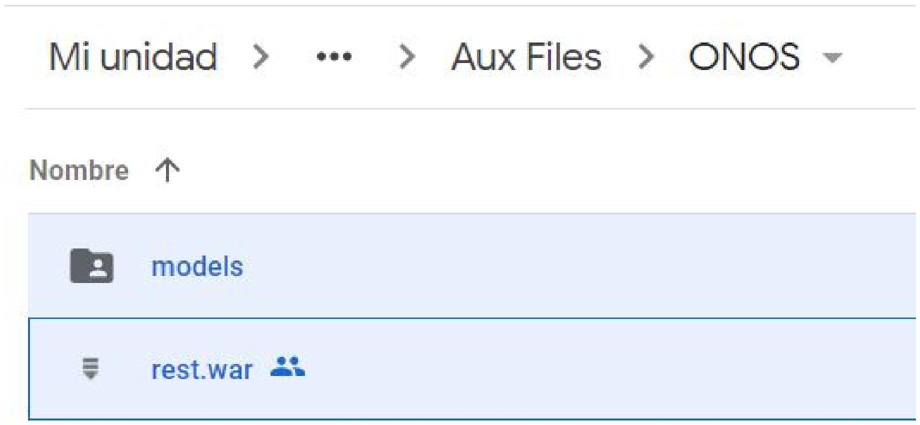


This section will explain the http branch: An onos detection & mitigation system which communicates with a rest service, deployed locally in a specialized JBoss container using docker. Consequently, this section will run under the premise that the user downloaded and installed the http branch from the github repository.

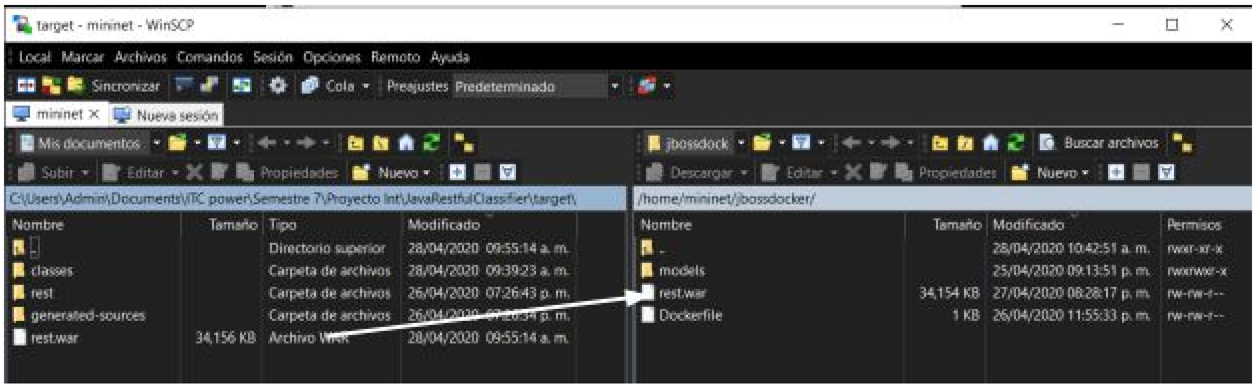
It is important to mention the ONOS container has to be deployed first, because the application assumes the web server will have the 172.17.0.2 ip address from the docker DHCP service, that is taken by the second container executed.

Developer's note: For convenience purposes, the http project was implemented using a maven web application archetype, and was tested first in the host machine in order to ease the development workflow. Later, the generated .war file was copied to the virtual machine & deployed using the steps of this section.

1. First, download the rest.war from the auxiliar project folder, (the models are already exported to the .war file):



2. Move the file to the virtual machine using the WinSCP method:


target - mininet - WinSCP
Local Marcar Archivos Comandos Sesión Opciones Remoto Ayuda
Sincronizar
Cola
Preajustes Predeterminado
mininet
Nueva sesión
Mis documentos
jbossdock
Buscar archivos
Subir
Editar
Propiedades
Nuevo
Descargar

```
C:\Users\Admin\Documents\ITC power\Semestre 7\Proyecto Int\JavaRestfulClassifier\target\
Nombre              Tamaño    Tipo                  Modificado
..                            Directorio superior   28/04/2020 09:55:14 a. m.
classes                       Carpeta de archivos   28/04/2020 09:39:23 a. m.
rest                          Carpeta de archivos   26/04/2020 07:26:43 p. m.
generated-sources             Carpeta de archivos   26/04/2020 07:26:34 p. m.
rest.war            34,156 KB Archivo WAR           28/04/2020 09:55:14 a. m.
```

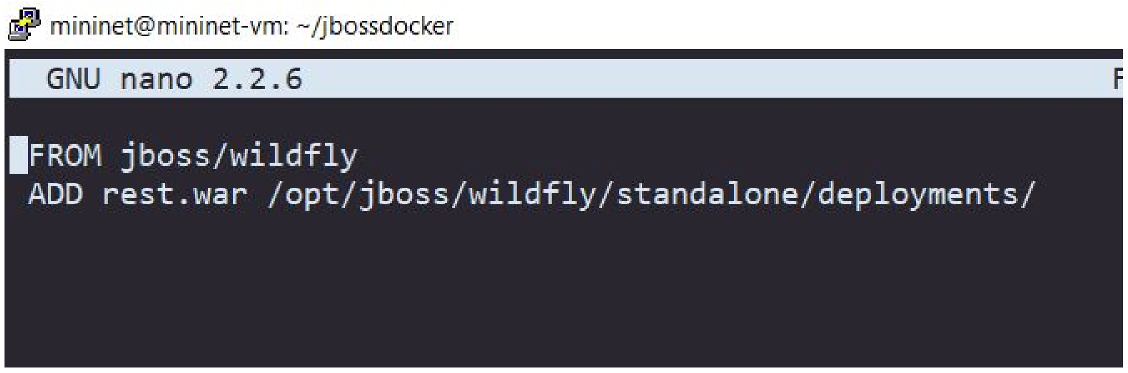
```
/home/mininet/jbossdocker/
Nombre        Tamaño    Modificado                  Permisos
..                      28/04/2020 10:42:51 a. m.   rwxr-xr-x
models                  25/04/2020 09:13:51 p. m.   rwxrwxr-x
rest.war      34,154 KB 27/04/2020 08:28:17 p. m.   rw-rw-r--
Dockerfile    1 KB      26/04/2020 11:55:33 p. m.   rw-rw-r--
```

3. Now establish an SSH connection with the virtual machine, locate the directory where the rest.war file is, and create a file named Dockerfile (in this case, the rest.war is stored in a recently created folder named jbossdocker):

```
cd jbossdocker
```

```
nano Dockerfile
```

4. The file will contain the Docker instructions to build a custom Jboss container by simply copying the rest.war file to the specified route:



```
mininet@mininet-vm: ~/jbossdocker
  GNU nano 2.2.6                                                          F

FROM jboss/wildfly
ADD rest.war /opt/jboss/wildfly/standalone/deployments/
```

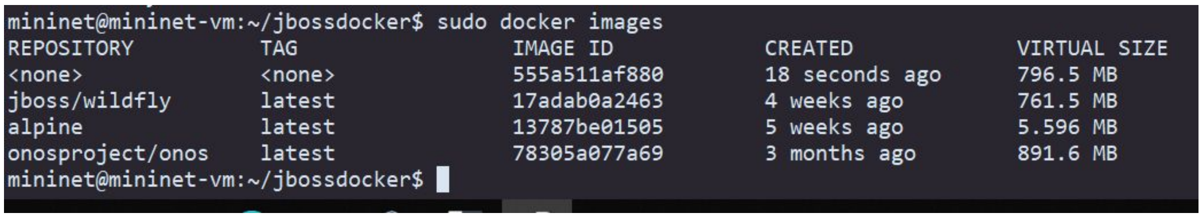
5. The specified route doesn't exist yet, so the following step consists in saving the Dockerfile and building an image with it:

```
sudo docker build -f Dockerfile ./
```

6. Next, we will start a container using the built image ID, by typing:

```
sudo docker images
```

The result will be similar to this illustration:



```
mininet@mininet-vm:~/jbossdocker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
<none>              <none>              555a511af880        18 seconds ago      796.5 MB
jboss/wildfly       latest              17adab0a2463        4 weeks ago         761.5 MB
alpine              latest              13787be01505        5 weeks ago         5.596 MB
onosproject/onos    latest              78305a077a69        3 months ago        891.6 MB
mininet@mininet-vm:~/jbossdocker$
```

The point of interest resides in the IMAGE ID column, please refer to the most recent built image (in this case being the one with the following id: 555a511af880)

7. Now, the time to deploy the web application has arrived and it can be achieved by following this type of command structure, the parts to be modified are marked in blue:

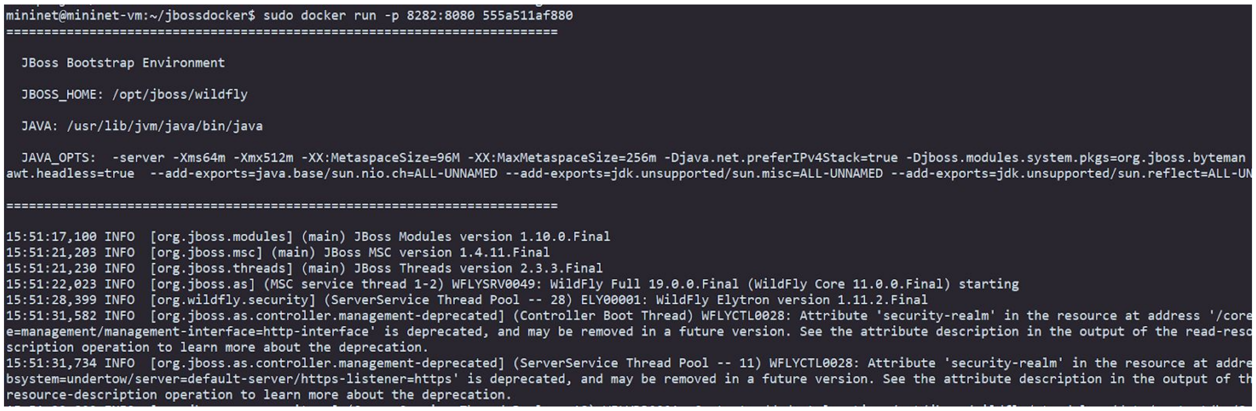
```
sudo docker run -p externalPort:internalPort imageId
```

The -p flag establishes an external port that can be accessed outside the virtual machine, mapping the incoming packets to the internal port. That means the deployed web container can be tested by using the host machine, easing the debugging process.

Now, following the previous command structure, the current example will lead to this:

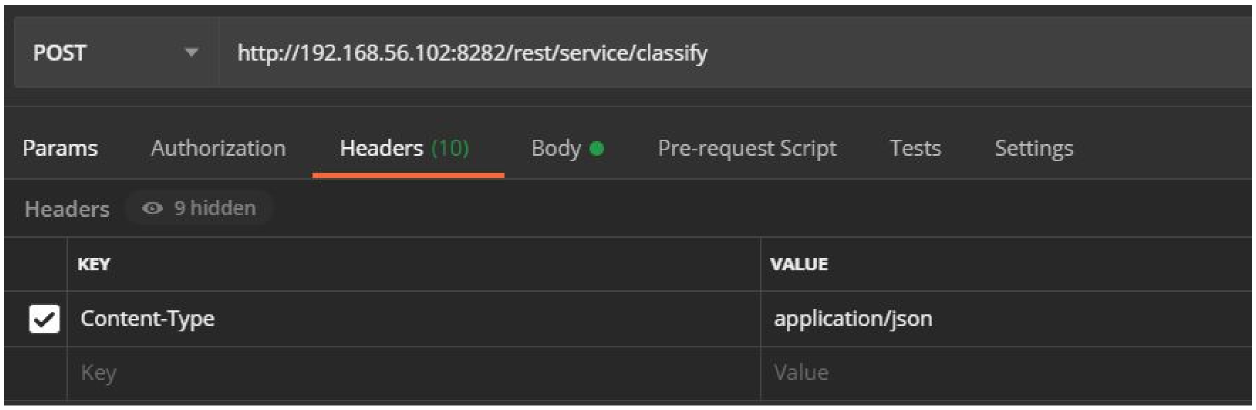
```
sudo docker run -p 8282:8080 555a511af880
```

The server deployment shall display the logs:



```
mininet@mininet-vm:~/jbossdocker$ sudo docker run -p 8282:8080 555a511af880
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: /opt/jboss/wildfly

  JAVA: /usr/lib/jvm/java/bin/java

  JAVA_OPTS:  -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman
awt.headless=true  --add-exports=java.base/sun.nio.ch=ALL-UNNAMED --add-exports=jdk.unsupported/sun.misc=ALL-UNNAMED --add-exports=jdk.unsupported/sun.reflect=ALL-UN

=========================================================================

15:51:17,100 INFO  [org.jboss.modules] (main) JBoss Modules version 1.10.0.Final
15:51:21,203 INFO  [org.jboss.msc] (main) JBoss MSC version 1.4.11.Final
15:51:21,230 INFO  [org.jboss.threads] (main) JBoss Threads version 2.3.3.Final
15:51:22,023 INFO  [org.jboss.as] (MSC service thread 1-2) WFLYSRV0049: WildFly Full 19.0.0.Final (WildFly Core 11.0.0.Final) starting
15:51:28,399 INFO  [org.wildfly.security] (ServerService Thread Pool -- 28) ELY00001: WildFly Elytron version 1.11.2.Final
15:51:31,582 INFO  [org.jboss.as.controller.management-deprecated] (Controller Boot Thread) WFLYCTL0028: Attribute 'security-realm' in the resource at address '/core
e=management/management-interface=http-interface' is deprecated, and may be removed in a future version. See the attribute description in the output of the read-reso
scription operation to learn more about the deprecation.
15:51:31,734 INFO  [org.jboss.as.controller.management-deprecated] (ServerService Thread Pool -- 11) WFLYCTL0028: Attribute 'security-realm' in the resource at addre
bsystem=undertow/server=default-server/https-listener=https' is deprecated, and may be removed in a future version. See the attribute description in the output of th
resource-description operation to learn more about the deprecation.
```

8. The web service can be optionally tested using Postman from the host machine (please refer to the official repository for the whole request body)):

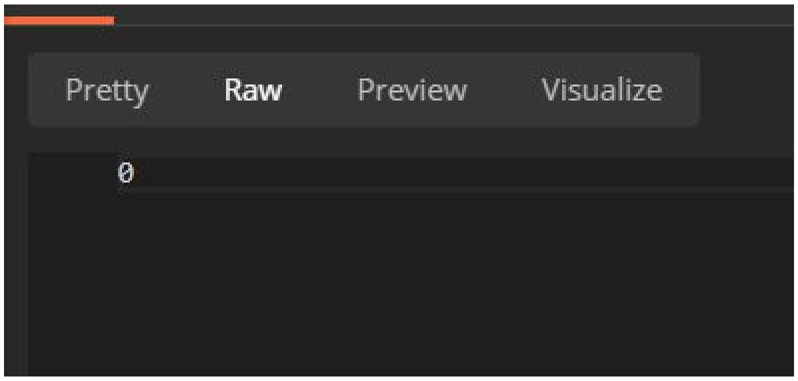


POST http://192.168.56.102:8282/rest/service/classify

Headers (10)

KEY	VALUE
✓ Content-Type	application/json
Key	Value

When the send button is clicked, the response will be an Integer number (The number 0 is for normal flow traffic):



```
0
```

9. Now that everything has been configured, it is time to test the deployed onos application:

```
ssh -p 8101 karaf@172.17.0.1

app activate mx.itesm.httpddosdetector

log:tail HttpDdosDetector
```

10. A topology can be instantiated to test the results:

```
sudo mn --controller remote,ip=172.17.0.1
```

Troubleshooting

Service org.onosproject.security.AuditService not found

The appearance of this message while typing a command on the ONOS application means that one or more essential modules have not been loaded. Try waiting 30-60 seconds more, if the problem persists permanently, it may be a file corruption issue that can be addressed by updating the software & hardware system.

Error 404 during ONOS app installation

If introducing the `onos-app 172.17.0.1 reinstall!`... command results in a 404 prompt, it may indicate to the ONOS container can be inactive or has another IP. Please ensure this container is active by typing the following command:

```
sudo docker start onos
```

Later, ensure the ONOS instance has the first IP assignable address (172.17.0.1):

```
sudo docker inspect onos | grep \"IPAddress\"
```

If this is not the case, the easiest way to reset the docker IP assignation is by restarting its service and then starting the ONOS container again:

```
sudo systemctl restart docker

sudo docker start onos
```

You can always reset the virtual machine, although it requires a lot more time.

Error 409 during ONOS app installation

This error comes out because there's another ONOS application running with the same name (that is to say, a previous version of the app). Please establish a SSH connection to ONOS and deactivate the current version and try again:

```
ssh -p 8101 karaf@172.17.0.1

app deactivate mx.itesm.httpddosdetector

logout

onos-app 172.17.0.1 reinstall!
target/httpddosdetector-1.0-SNAPSHOT.oar
```

References

http://javedmandary.blogspot.com/2016/09/install-maven-339-on-ubuntu.html

https://stackoverflow.com/questions/22907231/copying-files-from-host-to-docker-containe
r

https://groups.google.com/a/onosproject.org/forum/#!topic/onos-dev/WcxHkHK_Wkc

https://github.com/A01422807/httpDetector/tree/master

https://github.com/epicLevi/process-able-classifier

https://github.com/A01422807/httpDetector/tree/process-able

https://github.com/epicLevi/JavaRestfulClassifier

https://github.com/A01422807/httpDetector/tree/http