

Workshop Lab Setup

Overview

This lab guide will primarily be used for New Trainers, Community Trainers induction/onboarding as well as accessible to the Internet Community. This guide enables how to setup different APNIC workshop lab environment, that includes:

- Advance Routing
- IPv6 Deployment
- Secure Internet Routing – RPKI
- IPSEC
- DNS
- Segment Routing

This script is designed to work on Ubuntu 18.04 LTS. It should be run under root. Please install Ubuntu 18.04 LTS on Bare-metal hardware or you can deploy it on any Virtualization Software i.e. VMware etc.

All the workshop essentials can be clone from the APNIC GitHub profile:

```
https://github.com/githubapnic
```

Please note that APNIC GitHub profile is publicly accessible.

For the installation/cloning on your own Hardware/VM you can use your own `Username` and `Password` .

Under the APNIC GitHub profile you will find the `Ubuntu` repository. Click to access the repository.

 githubapnic committed <code>ccead42</code> 15 days ago	 186 commits	 1 branch	 0 tags
 UnattendInstall	Update README.md		4 months ago
 installpackages	Update readme.md		4 months ago
 removepackages	Update readme.md		5 months ago
 staticIP	Update README.md		5 months ago
 workshops	Add files via upload		15 days ago
 LICENSE	Initial commit		5 months ago
 README.md	Update README.md		4 months ago

All the Labs are located under the workshop folder.

Please read the README.md under the workshop before moving ahead. All the important and troubleshooting steps are listed there.

Ubuntu 18.04 Installation

From the Ubuntu website you can download the Ubuntu 18.04 OS.

```
https://releases.ubuntu.com/18.04.5/
```

Based on your Operating System, Hardware and VM you can install the Ubuntu 18.04 from the following:

```
https://howtoubuntu.org/how-to-install-ubuntu-18-04-bionic-beaver
```

Cloning GitHub Repository

After completing the installation, clone the GitHub repo in Ubuntu 18.04 by executing the following:

```
cd ~
git clone https://github.com/githubapnic/Ubuntu.git
```

```
apnic@apnic:~$ git clone https://github.com/githubapnic/Ubuntu.git
Cloning into 'Ubuntu'...
remote: Enumerating objects: 974, done.
remote: Total 974 (delta 0), reused 0 (delta 0), pack-reused 974
Receiving objects: 100% (974/974), 522.85 KiB | 663.00 KiB/s, done.
Resolving deltas: 100% (545/545), done.
apnic@apnic:~$
```

Go back to the home '~' directory again to change the permissions on all the shell scripts to allow it to be run.

```
cd ~
find Ubuntu/ -type f -iname "*.sh" -exec chmod u+x {} \;
find Ubuntu/ -type f -iname "run-dynamips" -exec chmod u+x {} \;
```

```
apnic@apnic:~$ find Ubuntu/ -type f -iname "*.sh" -exec chmod u+x {} \;
apnic@apnic:~$ find Ubuntu/ -type f -iname "run-dynamips" -exec chmod u+x {} \;
```

Update the libraries and dependencies.

```
sudo apt-get update
sudo apt-get upgrade
```

Lab_RPKI Installation

All the workshops are located under `cd /Ubuntu/workshops` directory.

```
apnic@apnic:~/Ubuntu/workshops$ ll
total 32
drwxrwxr-x 7 apnic apnic 4096 Jul 21 08:19 ./
drwxrwxr-x 8 apnic apnic 4096 Jul 21 08:19 ../
drwxrwxr-x 2 apnic apnic 4096 Jul 21 08:19 aptcache/
drwxrwxr-x 3 apnic apnic 4096 Jul 21 08:19 ipsec/
drwxrwxr-x 4 apnic apnic 4096 Jul 21 08:19 ipv6/
-rw-rw-r-- 1 apnic apnic 2107 Jul 21 08:19 README.md
drwxrwxr-x 4 apnic apnic 4096 Jul 21 08:19 routing/
drwxrwxr-x 4 apnic apnic 4096 Jul 21 09:30 rpki/
apnic@apnic:~/Ubuntu/workshops$
```

Change the directory to /rpki to proceed with the installation of required scripts and tools.

```
cd ~ /Ubuntu/workshops/rpki
```

```
apnic@apnic:~/Ubuntu/workshops/rpki$ ll
total 172
drwxrwxr-x 4 apnic apnic 4096 Jul 21 09:30 ./
drwxrwxr-x 7 apnic apnic 4096 Jul 21 08:19 ../
-rw-rw-r-- 1 apnic apnic 259 Jul 21 08:19 10-lxc.yaml
-rw-rw-r-- 1 apnic apnic 558 Jul 21 08:19 dnsmasq.conf
drwxrwxr-x 3 apnic apnic 4096 Jul 21 08:19 dynamips/
-rw-r--r-- 1 root root 121497 Jul 21 11:23 install.log
-rwxrw-r-- 1 apnic apnic 465 Jul 21 08:19 installRoutinator.sh*
-rw-rw-r-- 1 apnic apnic 1500 Jul 21 08:19 README.md
drwxrwxr-x 2 apnic apnic 4096 Jul 21 08:19 scripts/
-rwxr--r-- 1 apnic apnic 14602 Jul 21 08:19 setup_rpki_workshop.sh*
apnic@apnic:~/Ubuntu/workshops/rpki$
```

If you get the error from the windows OS or text editors, it will then replace the suspicious character using the following command:

```
/bin/bash^M
sed -i -e 's/\r$//' <filename>
```

Executing Script

Execute the script under the following path:

```
cd~/Ubuntu/workshop/rpki/  
  
sudo ./setup_rpki_workshop.sh
```

```
apnic@apnic:~/Ubuntu/workshops/rpki$ sudo ./setup_rpki_workshop.sh  
[sudo] password for apnic: █
```

Approximately the process will take ~10minutes to install the dependencies.

Open a separate session, change into same directory to show the logs, by typing the following:

```
cd~/Ubuntu/workshops/rpki  
tail -f install.log
```

```
apnic@apnic:~/Ubuntu/workshops/rpki$ tail -f install.log
```

Ref. logs will disapeared on the log window and verify the packages.

```
Setting up librbdl (12.2.13-0ubuntu0.18.04.2) ...  
Setting up qemu-block-extra:amd64 (1:2.11+dfsg-1ubuntu7.28) ...  
Setting up qemu-utils (1:2.11+dfsg-1ubuntu7.28) ...  
Setting up cloud-image-utils (0.30-0ubuntu5) ...  
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
Processing triggers for install-info (6.5.0.dfsg.1-2) ...  
##### Creating template.apnictraining.net
```

```

# The default user is 'apnic' with password 'training'!
# Use the 'sudo' command to run tasks as root in the container.
##

##### Update IP details to 192.168.30.100
addresses: [ 192.168.30.100/24 ]
addresses: [ 8.8.8.8, 8.8.4.4 ]
127.0.1.1 template.apnictraining.net
##### Creating RPKI container
##### Copying template.apnictraining.net to rpki.apnictraining.net & rpki
##### Update IP and Host details to:
addresses: [ 192.168.30.240/24 ]
addresses: [ 8.8.8.8, 8.8.4.4 ]
127.0.1.1 rpki.apnictraining.net
##### Creating apt.apnictraining.net
##### Please wait while apt.apnictraining.net is downloaded and created ...
Checking cache download in /var/cache/lxc/bionic/rootfs-amd64 ...
Copy /var/cache/lxc/bionic/rootfs-amd64 to /var/lib/lxc/apt.apnictraining.net/rootfs ...
Copying rootfs to /var/lib/lxc/apt.apnictraining.net/rootfs ...
rsync: write failed on "/var/lib/lxc/apt.apnictraining.net/rootfs/usr/share/doc/initramfs-tools-core/examples/example_
script": No space left on device (28)
rsync error: error in file IO (code 11) at receiver.c(393) [receiver=3.1.2]
lxc-create: apt.apnictraining.net: lxccontainer.c: create_run_template: 1617 Failed to create container from template
lxc-create: apt.apnictraining.net: tools/lxc_create.c: main: 327 Failed to create container apt.apnictraining.net
##### Update IP details to 192.168.30.248
addresses: [ 192.168.30.248/24 ]
addresses: [ 8.8.8.8, 8.8.4.4 ]
##### Set lxcbr0 DHCP settings to 192.168.30.0 range
LXC_ADDR="192.168.30.254"
LXC_NETWORK="192.168.30.0/24"
LXC_DHCP_RANGE="192.168.30.65,192.168.30.94"
# dnsmasq. For instance, you can use 'dhcp-host=mail1,192.168.30.25400' to have
# container 'mail1' always get ip address 192.168.30.25400.
# domain. You can then add "server=/lxc/192.168.30.254' (or your actual $LXC_ADDR)
##### Update name server details:
##### Copy some useful LXC scripts
##### Copy RPKI topology files
##### Change Timezone to NYC for dynamips to work
##### Installation Finished. Workshop files are located:
/home/apnic/virtual_labs/rpki
/home/apnic/Ubuntu/workshops/rpki
/home/apnic/Documents/scripts/
/var/lib/lxc/rpki.apnictraining.net/
##### Please update /home/apnic/virtual_labs/rpki/topology.net file with the IOS image.
##### Current image name in /home/apnic/virtual_labs/images:

##### Current image name in topology file:
image = ../images/c7200-advipservicesk9-mz.152-4.S3.image
#####

```

Multiple dependencies will be installed during script execution.

```
apnic@apnic:~/Ubuntu/workshops/rpki$ sudo ./setup_rpki_workshop.sh
##### Checking Ubuntu Version
##### Updating Packages
Extracting templates from packages: 100%
##### SSH already installed
##### Screen already installed
##### Installing dynamips
Success!
##### Installing dynagen
Success!
##### Enable IPv4 and IPv6 Forwarding
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
##### Installing LXC LXCTL LXC-Templates
Extracting templates from packages: 100%
Success!
##### Creating template.apnictraining.net
Enter username for LXC template: █
```

For LXC template, it will prompt to enter the and .

Change the directory to `cd~/virtual_labs/images/` and copy the cisco image here:

Note: This part of installation shows you the IMAGE NAME make sure you have the image with that name or if you are using some different image you need to make the changes under topology.net file.

```
cd~/home/apnic/virtual_labs/images
```

```
https://www.cisco.com/c/en/us/support/routers/7200-series-routers/tsd-products-support-series-home.html
```

Start the Lab

Follow the below sequence to start the rpki container, dynamips and tap interfaces.

1. Change the directory;

```
cd ~/Documents/scripts
screen
sudo ./start_rpkiServer_routers.sh
```

2. Detach from the dynagen screen, press `ctrl+a+d`

3. Start the tap interfaces;

```
sudo ./start_rpki_tap.sh
```

4. Verify the containers status;

```
sudo lxc-ls -f | grep run
```

5. Confirm tap interfaces are connected to bridge

```
brctl show
```

Dynamips Help:

1. Type 'help' to view the options under dynamips

show device - to show the device status show run – to show the running configuration

2. Under the dynamips, start and stop all the routers;

```
start/all  
stop/all  
exit
```

Stop the Lab

1. Stop the rpki container

```
sudo lxc-stop -n rpki.apnictraining.net
```

Reference:

Please watch the following installation video:

https://www.youtube.com/watch?v=_qFx2GLURLs&list=PLbCDrr5wzG7FABZNhP2IGI8iRpKqRSdEC&index=5