

Introduction to the Cisco 5921 Embedded Services Router Demonstration and Self-Paced Training v1

Last Updated: 07-OCT-2015

About This Cisco Solution

In this demonstration, we are going to feature the Cisco 5921 Embedded Services Router (ESR) installed on an Ubuntu core Linux system in a command line environment. For most embedded applications, their operating environment is usually that of a “black box” with no displays at all. The demonstration section walks the user through the highlights and features of this solution, to improve the end user’s familiarity with the Cisco 5921 ESR. No previous Linux experience is required, since Router 1 and Router 2 have operational preinstalled Cisco 5921 ESR’s. If you have learned the very basics of the IOS command line for the Integrated Services Routers or Catalyst Switches, you will be quite comfortable with this demonstration.

The purpose of the lab sections of this document is to enhance your skills as an integrator by giving a concrete example of how to qualify a Linux embedded core build for use with the Cisco 5921 Embedded Services Router (ESR), as well as, taking you through the install and troubleshooting of the Cisco 5921 ESR installation on that system. Finally, we have included reference materials to provide additional information about this product.

We use the Cisco 5921 Embedded Services Router Integration Guide as a starting point for this document. For the most part, we will not repeat content covered in the Integration Guide here. However, we will try to give more detail using actual examples in this document. We will also point out where you may find various resources on the Internet. This is by no means a Cisco endorsement of those sites; it is merely intended to save you the time of searching for third party resources and answers to common questions.

Since the finished product may take many forms, you may like to see how integration partners are using this solution. For more information, please visit our virtual partner community on Cisco DevNet: [Finished Products Containing the Cisco 5921 Embedded Services Router](#).

About This Demonstration

Cisco 5921 Embedded Services Router Install v1 includes the following exercises:

Demonstration Scenario:

- Demonstrate and configure the Cisco 5921 Embedded Services Router on a new virtual machine (VM) (Scenario 1)

Lab Exercises:

- Install and configure an Ubuntu server VM for hosting the Cisco c5921 Embedded Services Router (Scenario 2)
- Troubleshoot the installation of the Cisco 5921 ESR on the Linux host (Scenario 3)

Reference Sections:

- Instructions on how to prequalify that the basic Linux dependencies are met (Scenario 4)
- Examples of how the configuration files should look (Appendices A – D)

Demonstration Requirements



The table below outlines the requirements for this preconfigured demonstration.

Table 1. Demonstration Requirements

Required	Optional
<ul style="list-style-type: none"> Laptop, Desktop PC or Tablet 	<ul style="list-style-type: none"> Cisco AnyConnect client

Demonstration Configuration

This demonstration contains preconfigured users and components to illustrate the scripted scenarios and features of **Cisco 5921 Embedded Services Router Install v1**. All information needed to complete the demonstration scenario, is located in the **Topology** and **Servers** menus of your active demonstration.

- **Topology Menu.** Click on any server in the topology and a popup window will appear with available server options.
- **Servers Menu.** Click on  or  next to any server name to display the available server options and credentials.

Demonstration Topology

This demonstration includes several enterprise server VMs. Most of the servers are fully configurable using the administrative level account. Administrative account details are included in the script steps where relevant and in the server details table.

Figure 1. Demonstration Topology Overview

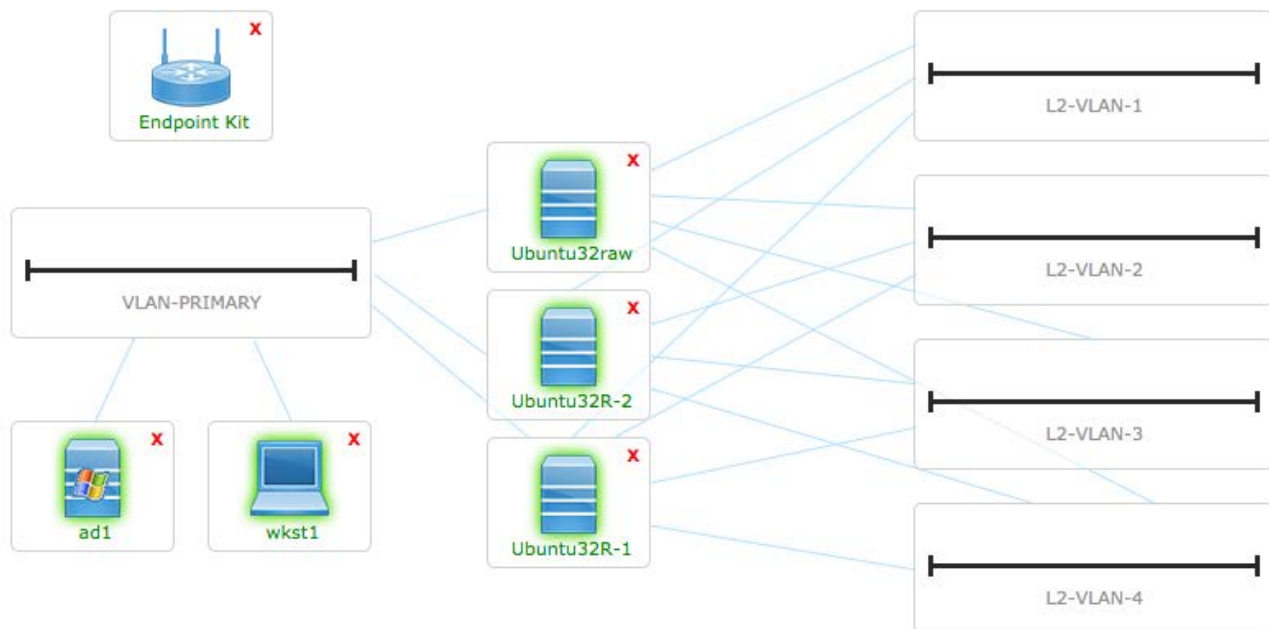


Table 2. Demonstration Server Information

Application	Version	URL	Username	Password
Ubuntu-32bit without ESR pre-installed (Ubuntu32raw)	c5921i86-universalk9-tar.SPA.155-2.T	198.18.134.122	administrator	C1sco12345
Ubuntu-32bit with ESR pre-installed (Ubuntu32R-2)	c5921i86-universalk9-tar.SPA.155-2.T	198.18.134.121	administrator (Ubuntu) root (ESR)	C1sco12345 cisco
Ubuntu-32bit with ESR pre-installed (Ubuntu32R-1)	c5921i86-universalk9-tar.SPA.155-2.T	198.18.134.120	administrator (Ubuntu) root (ESR)	C1sco12345 cisco
Active Directory (ad1)	Windows Server 2008 R2	198.18.133.1	administrator	C1sco12345
Workstation 1 (wkst1)	Windows 7 Pro	198.18.133.36	DCLOUD\administrator	C1sco12345

Demonstration Preparation

BEFORE DEMONSTRATING

We strongly recommend that you go through this process at least once, before presenting in front of a live audience. This will allow you to become familiar with the structure of the document and the demonstration.

PREPARATION IS KEY TO A SUCCESSFUL CUSTOMER PRESENTATION.

Follow the steps below to schedule your demonstration and configure your demonstration environment.

1. Browse to dcloud.cisco.com, choose the location closest to you, and then login with your **Cisco.com credentials**.
2. Schedule a session [[Show Me How](#)].
3. Test your bandwidth from the session location before performing any scenario. [[Show Me How](#)]
4. Verify your session has a status of **Active** under **My Demonstrations** on the **My Dashboard** page in the dCloud UI.
 - It may take up to **25 minutes for your demonstration to become active**.
5. Connect your laptop. [[Show Me How](#)]
6. You can connect your laptop directly to the demonstration. Install and access **Cisco AnyConnect** on your laptop, using the **Cisco AnyConnect** credentials in the Cisco dCloud UI. [[Show Me How](#)]
7. Access the lab **Workstation** using your local **RDP program** (if you are connected to the demo via AnyConnect or behind a dCloud router kit) or **Cisco dCloud Remote Desktop Client** (if you are not connected to the demo via AnyConnect or a dCloud router). [[Show Me How](#)]
 - **Workstation 1** (198.18.133.36), User: **dcloud\administrator**, Password: **C1sco12345**

If you are using the Installation and Troubleshooting sections where you intend to practice and develop skills as a systems integrator, we make the following suggestions:

1. Save all router configurations.
2. On all VMs save:
 - /opt/cisco/c5921/SWROPTIONS
 - /etc/network/interfaces
 - Any other Linux scripts that you created or customized.

NOTE: The most expedient way to accomplish this will be via AnyConnect or behind a dCloud router kit. If you need a detailed guide regarding use of AnyConnect VPN access to dCloud, please see the following: https://dcloud-cms.cisco.com/help/install_anyconnect_pc_mac.

Introduction

Since much of our documentation uses Ubuntu as the reference system, this document will go into a bit more detail regarding how to qualify that the system dependencies are met in another Linux distribution and in the most challenging environment, the embedded system's command line interface.

There are no Linux restrictions as the integrator is responsible for building the system and verifying its functionality. Cisco has tested on Ubuntu, Fedora, CentOS, and OpenWRT. Cisco has also tested in desktop configurations and embedded configurations. If you need additional information, consult the Frequently Asked Questions section, Chapter 5 of the [Cisco 5921 Embedded Services Router Integration Guide](#).

For an Ubuntu "minimal system" there are three main variants:

1. **Default server** = Server kernel + "Basic Ubuntu server" task packages
2. **Minimal system** = Server kernel + no additional packages
3. **Virtual machine** = Virtual kernel + no additional packages

A virtual kernel is a lighter weight version of the kernel with several modules like audio and SATA left out.

The Ubuntu system requirements for this type of installation without a graphical desktop are relatively light. For more information, see <https://help.ubuntu.com/community/Installation/SystemRequirements>.

Ubuntu Server (CLI) Installation

1. 300 MHz x86 processor
2. 192 MB of system memory (RAM)
3. 1 GB of disk space
4. Graphics card and monitor capable of 640x480
5. [CD drive](#) ← Installation without a CD is also possible, for example it could be done from a USB Flash memory drive

We have downloaded **mini.iso** for Ubuntu 14.04 LTS 32-bit and have constructed the dCloud VMs used in this demo as follows:

- 10 GB disk, thin provisioned.
- Five NIC's. One port will serve as the Linux management interface. **This port is not shared with the router.** The other four ports will be one virtual 4-port card for our Cisco 5921 ESR.
- 4 GB of RAM

The download location and Ubuntu documentation is found at <https://help.ubuntu.com/community/Installation/MinimalCD>. If you want to download it yourself, please look for "32-bit PC (i386, x86)" and choose option 3, [Ubuntu 14.04 LTS "Trusty Tahr"](#) 31MB. In a future guide, we might also explore the 64-bit build and cover how to import the needed 32-bit libraries.

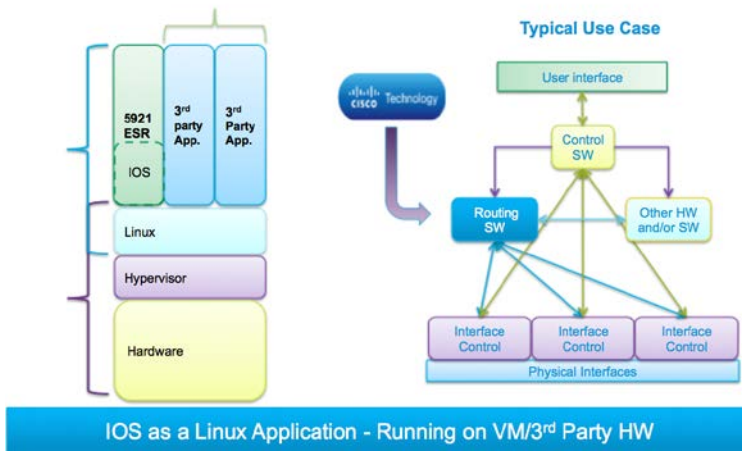
Conventions used in this guide:

- The narrative is in Arial font and computer interaction is shown on Courier. We bold face **user inputs** and **commands**.
- If we have computer interaction from a different system, a different font color will be used.
- When something is highlighted in a NOTE box it is either very important or it is something that we are looking for.

The following graphic depicts the virtual machine environment that we will use for this project:

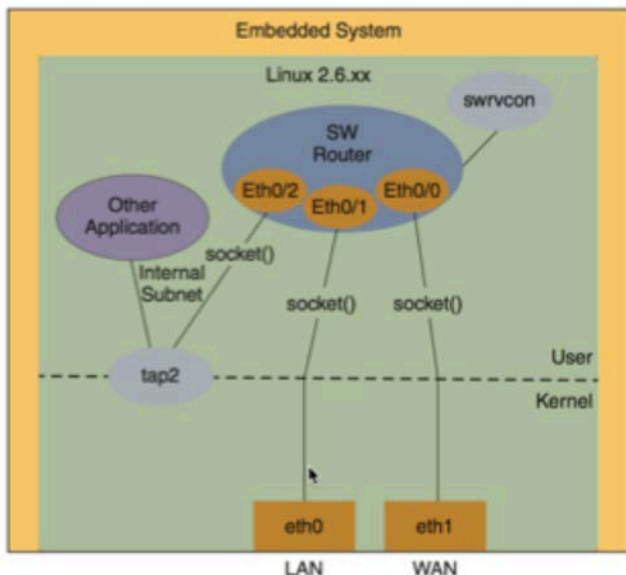
Figure 2. IOS Linux/VM Architecture

5921 IOS Linux/VM Architecture



Note that while it is possible to include third party applications, the 5921 ESR is the focus of this demonstration. This demonstration will not use the “tap” interface in a programmatic manner where it can be used to allow third party applications to interact with the 5921 ESR as if there was a dedicated virtual Ethernet port used for that application as depicted below. Instead, on one of the routers, we will use it as a virtual network to relay ntp between its Linux host and the other routers.

Figure 3. Tap Interface for c5921 ESR

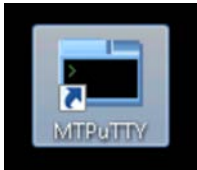


Scenario 1: Solution Demonstration

This demonstration will begin with two running copies on Ubuntu32w-1 and Ubuntu32w-2. You may simply log into those two routers to try out features and test router configurations.

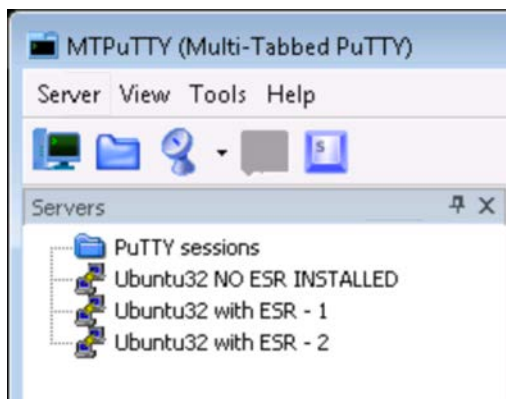
1. Connect to Workstation 1 and once logged into the desktop, double click to launch Multi-Tab PuTTY:

Figure 4. Multi-Tab PuTTY Icon



2. Here you will find preconfigured links to launch ssh sessions to the 3 linux VM's. Double-click on these links under **Servers** to connect to the corresponding VM. Please give the VM a minute to respond after launching a new session.

Figure 5. Pre-configured PuTTY Sessions



3. If you are connecting via AnyConnect, the IP addresses for the management ports of the VMs are as follows:

- Ubuntu32 NO ESR installed -198.18.134.122
- Ubuntu32 with ESR - 1 -198.18.134.120
- Ubuntu32 with ESR - 2 -198.18.134.121

If you connect via AnyConnect, you may launch ssh sessions via a ssh client of your choice.

NOTE: If you wish to easily capture the configuration files for future reference, AnyConnect would be the preferred method of access.

4. Connect to Ubuntu32 with ESR - 1. Double-click on the link to **ESR – 1** in PuTTY and it will open up a new tab to the right with a CLI connection to this VM. Use this method to connect to any of the VMs in this demonstration.

Please note that CDP does not operate through the virtual switches that connect the routers in the dCloud environment.

You may begin by observing how the IOS file structure is contained within the Linux file structure from an operational perspective.

```
administrator@ubuntu32:~$ sudo su
[sudo] password for administrator:
```

Type password **Cisco12345**.

```
root@ubuntu32:/home/administrator# cd /opt/cisco/c5921 <- This is the Linux directory where installed
root@ubuntu32:/opt/cisco/c5921# ls -al <- the 5921. Let us have a look.
```

```
total 329116
drwxr-xr-x 5 root root      4096 Sep 16 16:32 .
drwxr-xr-x 3 root root      4096 Jul 21 23:22 ..
drwxr-xr-x 2 root root      4096 Jul 21 23:28 c5921i86-universalk9-ms.155-2.T
-r-xr-xr-x 1 root root 150158796 Jul 21 23:29 c5921i86-universalk9-ms.SPA
-rw-r--r-- 1 root root 168263680 Jul 21 23:27 c5921i86-universalk9-tar.SPA.155-2.T
-rwxr-xr-x 1 root root      3266 Sep  3 21:44 c5921-swr-init.sh
drwxr-xr-x 2 root root      4096 Jul 21 23:33 .cs10
drwxr-xr-x 2 root root      4096 Jul 21 23:33 .cs11
-rw-r--r-- 1 root root       132 Sep 16 16:32 .hs_a.pri
-rw-r--r-- 1 root root       132 Sep 16 16:32 .hs_b.pri
-r-xr-xr-x 1 root root 17954572 Jul 21 23:29 libdyncs.so
-rw-r----- 1 root root  524288 Sep 16 09:57 nvram_00100
-rwxr-xr-x 1 root root       489 Sep  3 23:46 staticroutes.sh
-rwxr--r-- 1 root root      1616 Sep  3 09:44 SWROPTIONS
-rwxr--r-- 1 root root      1696 Aug 31 13:04 SWROPTIONS.save
-r-xr-xr-x 1 root root     24752 Jul 21 23:29 swr_reload
-r-xr-xr-x 1 root root     31076 Jul 21 23:29 swrvcon
```

```
root@ubuntu32:/opt/cisco/c5921# cd .cs10 <- The first thing we notice is that there are hidden
root@ubuntu32:/opt/cisco/c5921/.cs10# ls -al <- directories that were created by the 5921 when it
total 40 <- first ran. This is the certificate store.
```

```
drwxr-xr-x 2 root root 4096 Jul 21 23:33 .
drwxr-xr-x 5 root root 4096 Sep 16 16:32 ..
-rw-r--r-- 1 root root 123 Aug 28 14:22 c5921.prs
-rw-r--r-- 1 root root 1052 Jul 21 23:33 lservrc.pri
-rw-r--r-- 1 root root 32 Jul 21 23:33 revocationchk.prs
-rw-r--r-- 1 root root 640 Jul 21 23:33 revocationprs.prs
-rw-r--r-- 1 root root 269 Sep 16 09:57 sa.info
-rw-r--r-- 1 root root 32 Jul 21 23:33 trailprschk.prs
-rw-r--r-- 1 root root 760 Jul 21 23:33 trailprs.prs
-rw-r--r-- 1 root root 193 Aug 28 14:22 udi_hist.pri
```

```
root@ubuntu32:/opt/cisco/c5921/.cs10# cd ..
root@ubuntu32:/opt/cisco/c5921# ./swrvcon 100 <- "swrvcon" is software router virtual console
```

```
c5921_ubuntu32_1>en
Password:
```

Password is **cisco**.

```
c5921_ubuntu32_1#dir <- let us see the file structure from IOS.
```

Directory of unix:/

```
18766 -r-x 150158796 Jul 21 2015 22:29:06 -08:00 c5921i86-universalk9-ms.SPA
18774 -rw- 524288 Sep 16 2015 08:57:37 -08:00 nvram_00100
18770 -rwx 3266 Sep 3 2015 20:44:36 -08:00 c5921-swr-init.sh
265620 -rwx 489 Sep 3 2015 22:46:33 -08:00 staticroutes.sh
18765 -rw- 168263680 Jul 21 2015 22:27:27 -08:00 c5921i86-universalk9-tar.SPA.155-2.T
141090 drwx 4096 Jul 21 2015 22:28:01 -08:00 c5921i86-universalk9-ms.155-2.T
141110 drwx 4096 Jul 21 2015 22:33:41 -08:00 .cs11
18768 -r-x 31076 Jul 21 2015 22:29:41 -08:00 swrvcon
27 -rw- 132 Sep 16 2015 15:32:28 -08:00 .hs_a.pri
17425 -rwx 1696 Aug 31 2015 12:04:26 -08:00 SWROPTIONS.save
18769 -r-x 24752 Jul 21 2015 22:29:56 -08:00 swr_reload
141109 drwx 4096 Jul 21 2015 22:33:41 -08:00 .cs10
18767 -r-x 17954572 Jul 21 2015 22:29:18 -08:00 libdyncs.so
18771 -rwx 1616 Sep 3 2015 08:44:57 -08:00 SWROPTIONS
25 -rw- 132 Sep 16 2015 15:32:28 -08:00 .hs_b.pri
```

6207111168 bytes total (4141420544 bytes free)


```

c5921_ubuntu32_1#cd .cs10
c5921_ubuntu32_1#dir
Directory of unix:/.cs10/

141117  -rw-          32  Jul 21 2015 22:33:41 -08:00  trailprschk.prs
141124  -rw-         269  Sep 16 2015 08:57:30 -08:00  sa.info
141111  -rw-         123  Aug 28 2015 13:22:10 -08:00  c5921.prs
141123  -rw-         193  Aug 28 2015 13:22:10 -08:00  udi_hist.pri
141121  -rw-          32  Jul 21 2015 22:33:41 -08:00  revocationchk.prs
141113  -rw-        1052  Jul 21 2015 22:33:41 -08:00  lservrc.pri
141119  -rw-         640  Jul 21 2015 22:33:41 -08:00  revocationprs.prs
141115  -rw-         760  Jul 21 2015 22:33:41 -08:00  trailprs.prs

6207111168 bytes total (4141420544 bytes free)
c5921_ubuntu32_1#

```

As you can see, the home directory of the 5921, /opt/cisco/c5921, is the device **unix**: which would equate to **flash**: on a typical ISR.

The file nvram_00100 equates to the nvram device in an ISR.

5. Next, we will review some typical IOS commands.

```

c5921_ubuntu32_1#sh ver
Cisco IOS Software, C5921 Software (C5921_I86-UNIVERSALK9-M), Version 15.5(2)T, RELEASE SOFTWARE
(fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 25-Mar-15 15:49 by prod_rel_team

ROM: Bootstrap program is Linux

c5921_ubuntu32_1 uptime is 6 hours, 59 minutes
System returned to ROM by reload at 0
System restarted at 08:56:53 PST Wed Sep 16 2015
System image file is "unix:./c5921i86-universalk9-ms.SPA"
Last reload reason: Unknown reason

```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

Cisco C5921 (Intel-x86) processor with 381079K bytes of memory.
 Processor board ID 100
 8 Ethernet interfaces
 512K bytes of NVRAM.

License Info:

License UDI:

```
-----
Device#      PID          SN
-----
*0          CISCO5921-K9      91J81443B9I
```

Packet forwarding: Enabled

Enforced traffic rate: 50 Mbps ← Take note of this value, we will be using it later.

Configuration register is 0x0

c5921_ubuntu32_1#sh ip rou ← Notice that the IOS command parser behaves exactly as expected.

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override
```

Gateway of last resort is 198.18.134.120 to network 0.0.0.0

```
S*    0.0.0.0/0 [1/0] via 198.18.134.120
      10.0.0.0/32 is subnetted, 2 subnets
C      10.0.0.120 is directly connected, Loopback0
O      10.0.0.121 [110/2] via 172.23.4.2, 07:01:51, Ethernet0/3
          [110/2] via 172.23.3.2, 07:01:51, Ethernet0/2
          [110/2] via 172.23.2.2, 07:01:51, Ethernet0/1
          [110/2] via 172.23.1.121, 07:01:51, Ethernet0/0
      172.23.0.0/16 is variably subnetted, 8 subnets, 2 masks
C      172.23.1.0/24 is directly connected, Ethernet0/0
L      172.23.1.120/32 is directly connected, Ethernet0/0
C      172.23.2.0/24 is directly connected, Ethernet0/1
L      172.23.2.1/32 is directly connected, Ethernet0/1
C      172.23.3.0/24 is directly connected, Ethernet0/2
L      172.23.3.1/32 is directly connected, Ethernet0/2
C      172.23.4.0/24 is directly connected, Ethernet0/3
L      172.23.4.1/32 is directly connected, Ethernet0/3
C      198.18.128.0/18 is directly connected, Ethernet1/0
      198.18.134.0/32 is subnetted, 1 subnets
L      198.18.134.131 is directly connected, Ethernet1/0
```

c5921_ubuntu32_1#sh ip rou ?

← including the “?”

```
Hostname or A.B.C.D Network to display information about or hostname
application         Application routes
bgp                 Border Gateway Protocol (BGP)
connected          Connected
dhcp               Show routes added by DHCP Server or Relay
eigrp              Enhanced Interior Gateway Routing Protocol (EIGRP)
isis              ISO IS-IS
lisp               Locator ID Separation Protocol (LISP)
list               IP Access list
loops              RIB routes forming loops
mobile             Mobile routes
multicast          Multicast global information
next-hop-override Show next-hop-overrides too
nhrrp             Next Hop Resolution Protocol (NHRP)
odr               On Demand stub Routes
ospf              Open Shortest Path First (OSPF)
ospfv3            OSPFv3
```

```

profile                IP routing table profile
repair-paths          Show repair paths too
rip                   Routing Information Protocol (RIP)
static                Static routes
summary              Summary of all routes
supernets-only        Show supernet entries only
tag                   Route Tag
track-table           Tracked static table
vrf                   Display routes from a VPN Routing/Forwarding instance
|                     Output modifiers
<cr>

```

c5921_ubuntu32_1#sh ip rou os ← type the <tab> key to auto fill this command.

c5921_ubuntu32_1#sh ip rou ospf

```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

```

Gateway of last resort is 198.18.134.120 to network 0.0.0.0

```

10.0.0.0/32 is subnetted, 2 subnets
O      10.0.0.121 [110/2] via 172.23.4.2, 07:04:30, Ethernet0/3
        [110/2] via 172.23.3.2, 07:04:30, Ethernet0/2
        [110/2] via 172.23.2.2, 07:04:30, Ethernet0/1
        [110/2] via 172.23.1.121, 07:04:30, Ethernet0/0
c5921_ubuntu32_1#

```

6. Next, compare the **show running-config** command with **show running-config brief**.

Notice how this:

```

crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B

```

```

5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
```

is reduced to this:

```

!
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
```

You can also do the opposite, type **show running-config all** and it will display all the default settings. We will not show that output here. It is very long but give it a try.

7. Now we will telnet over to the other router.

```

c5921_ubuntu32_1#telnet 172.23.1.121
Trying 172.23.1.121 ... Open
```

User Access Verification

```

Password:
c5921_ubuntu32_2>en
Password:                <- This password is cisco
c5921_ubuntu32_2#sh ver
Cisco IOS Software, C5921 Software (C5921_I86-UNIVERSALK9-M), Version 15.5(2)T, RELEASE SOFTWARE
(fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 25-Mar-15 15:49 by prod_rel_team
```

ROM: Bootstrap program is Linux

```

c5921_ubuntu32_2 uptime is 7 hours, 24 minutes
System returned to ROM by reload at 0
System restarted at 08:54:50 PST Wed Sep 16 2015
System image file is "unix:./c5921i86-universalk9-ms.SPA"
Last reload reason: Unknown reason
```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
<http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to

export@cisco.com.

Cisco C5921 (Intel-x86) processor with 381075K bytes of memory.
 Processor board ID 100
 4 Ethernet interfaces
 512K bytes of NVRAM.

License Info:

License UDI:

```
-----
Device#      PID          SN
-----
*0          CISCO5921-K9  9509000IL4J
```

Packet forwarding: Enabled
 Enforced traffic rate: 8 Kbps

← This is the permanent evaluation mode.

Configuration register is 0x0

c5921_ubuntu32_2#

8. We will try a few commands that show the licensing options and then we will enable the 90-day evaluation for the 50 Mbps performance level.

c5921_ubuntu32_2#show platform software license

Packet forwarding: Enabled
 Current enforcement forwarding rate: 8 Kbps
 Unique Device Identifier: CISCO5921-K9:9509000IL4J

License features supported:

Feature	Rate	Status
c5921-x86-default	8 Kbps	In Use
c5921-x86-evaluation	50 Mbps	-
c5921-x86-level1	10 Mbps	-
c5921-x86-level2	25 Mbps	-
c5921-x86-level3	50 Mbps	-

← This is only used for the classic license model.

c5921_ubuntu32_2#

9. The Classic License model is file based where the user installs a license key file. When a virtual machine installation is the operating environment, only cloud based Smart Call Home Client licensing architecture is available. We will turn on level3 but let's see all the details first.

c5921_ubuntu32_2#show license tech support
 Cisco Smart Licensing Agent, Version 1.0.0_development

Smart Licensing Enabled: Yes

UDI:
 PID:CISCO5921-K9,SN:9509000IL4J

Compliance Status: Evaluation

Entitlement Handle: no entry found

Smart Licensing State: unidentified (2)

Licensing Certificates:

ID Cert Info:

Start Date: Not available. Expiry Date: Not available

Signing Cert Info:

Start Date: Not available. Expiry Date: Not available

Upcoming Scheduled Jobs:

Certificate Renewal: Not Available

Certificate Expiration: Not Available

Authorization Renewal: Not Available

Authorization Expiration: Not Available

Daily Job: Sep 17 16:54:59 2015 UTC (16 hours, 18 minutes, 27 seconds remaining)

Component Versions: SA:(1_0_4_throttle)1.0.15, SI:(rel20)1.0.0, CH:(rel4)1.0.13,
PK:(rell6_twig)1.0.1

Other Tech Support Dump:

Grace period time remaining: Not in use

Eval period time remaining: 90 days, 0 hours, 0 minutes, 0 seconds (Not in use)

Stored State Machine State: 2

Transport Mode: Callhome

c5921_ubuntu32_2#

10. Notice that the "clock" is not running yet. We will start this now.

c5921_ubuntu32_2#**configure terminal**

5921_ubuntu32_2(config)#**license platform throughput level c5921-x86-level3**

c5921_ubuntu32_2(config)#**exit**

c5921_ubuntu32_2#**show platform software license**

Packet forwarding: Enabled

Current enforcement forwarding rate: 50 Mbps

Unique Device Identifier: CISCO5921-K9:9509000IL4J

License features supported:

Feature	Rate	Status
c5921-x86-default	8 Kbps	-
c5921-x86-evaluation	50 Mbps	-
c5921-x86-level1	10 Mbps	-
c5921-x86-level2	25 Mbps	-
c5921-x86-level3	50 Mbps	In Use

c5921_ubuntu32_2#**sh lic tech sup**

Cisco Smart Licensing Agent, Version 1.0.0_development

Smart Licensing Enabled: Yes

UDI:

PID:CISCO5921-K9,SN:9509000IL4J

Compliance Status: **Evaluation**

```
Entitlement Handle: 1
  Entitlement Status: In Use
  Tag: regid.2014-08.com.cisco.c5921-x86-level3,1.0_785fc8c0-0f25-4ad5-9397-770c97358bf5
  Version: 1.0, Enforce Mode: Eval period
  Requested Time: Wed Sep 16 16:40:10.660, Requested Count: 1
  Vendor String:
```

```
Smart Licensing State: unidentified (2)
```

```
Licensing Certificates:
  ID Cert Info:
    Start Date: Not available. Expiry Date: Not available
  Signing Cert Info:
    Start Date: Not available. Expiry Date: Not available
```

```
Upcoming Scheduled Jobs:
  Certificate Renewal: Not Available
  Certificate Expiration: Not Available
  Authorization Renewal: Not Available
  Authorization Expiration: Not Available
  Daily Job: Sep 17 16:54:59 2015 UTC (16 hours, 12 minutes, 2 seconds remaining)
```

```
Component Versions: SA:(1_0_4_throttle)1.0.15, SI:(rel20)1.0.0, CH:(rel4)1.0.13,
PK:(rel16_twig)1.0.1
```

```
Other Tech Support Dump:
  Grace period time remaining: Not in use
  Eval period time remaining: 89 days, 23 hours, 57 minutes, 12 seconds (In use)
  Stored State Machine State: 2
```

```
Transport Mode: Callhome
```

```
c5921_ubuntu32_2#
```

The 90 day evaluation clock is now counting down.

Full entitlement has steps that will tie the license to the organization purchasing the license and the CCO ID of the authorized administrator as well as the UDI of the product. It is beyond the scope of this demonstration. However, we have introduced some key concepts and commands pertinent to Smart Licensing.

11. Debug commands are also complete, as one would expect in any other IOS device. Here are a couple of examples:

```
c5921_ubuntu32_2#term mon
c5921_ubuntu32_2#debug ntp all
NTP events debugging is on
NTP core messages debugging is on
NTP clock adjustments debugging is on
NTP reference clocks debugging is on
NTP packets debugging is on

Sep 17 00:57:52.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:52.036: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:52.036: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:57:52.036: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:57:54.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:54.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
```

```

Sep 17 00:57:54.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:57:54.037: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:57:56.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:56.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:56.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:57:56.037: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:57:58.037: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:58.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:57:58.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:57:58.038: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:58:00.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:00.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:00.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:58:00.037: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:58:02.037: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:02.038: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:02.038: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:58:02.038: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:58:04.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:04.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:04.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:58:04.037: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.
Sep 17 00:58:06.036: NTP message sent to 198.18.134.132, from interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:06.037: NTP message received from 198.18.134.132 on interface 'Ethernet0/1'
(172.23.2.2).
Sep 17 00:58:06.037: NTP Core(DEBUG): ntp_receive: message received
Sep 17 00:58:06.037: NTP Core(DEBUG): ntp_receive: peer is 0xA30F2218, next action is 1.

```

```
c5921_ubuntu32_2#undebug all
```

```
All possible debugging has been turned off
```

```
c5921_ubuntu32_2# c5921_ubuntu32_2#debug ip ospf hello
```

```
OSPF hello debugging is on
```

```
c5921_ubuntu32_2#
```

```

Sep 17 01:02:07.501: OSPF-100 HELLO Et0/3: Rcv hello from 10.0.0.120 area 0 172.23.4.1
Sep 17 01:02:07.807: OSPF-100 HELLO Et0/0: Rcv hello from 10.0.0.120 area 0 172.23.1.120
Sep 17 01:02:08.453: OSPF-100 HELLO Et0/0: Send hello to 224.0.0.5 area 0 from 172.23.1.121
Sep 17 01:02:09.439: OSPF-100 HELLO Et0/2: Rcv hello from 10.0.0.120 area 0 172.23.3.1
Sep 17 01:02:10.040: OSPF-100 HELLO Et0/1: Rcv hello from 10.0.0.120 area 0 172.23.2.1
Sep 17 01:02:10.460: OSPF-100 HELLO Et0/1: Send hello to 224.0.0.5 area 0 from 172.23.2.2
Sep 17 01:02:13.130: OSPF-100 HELLO Et0/3: Send hello to 224.0.0.5 area 0 from 172.23.4.2
Sep 17 01:02:13.968: OSPF-100 HELLO Et0/2: Send hello to 224.0.0.5 area 0 from 172.23.3.2
Sep 17 01:02:16.851: OSPF-100 HELLO Et0/3: Rcv hello from 10.0.0.120 area 0 172.23.4.1
Sep 17 01:02:17.302: OSPF-100 HELLO Et0/0: Rcv hello from 10.0.0.120 area 0 172.23.1.120
Sep 17 01:02:18.171: OSPF-100 HELLO Et0/0: Send hello to 224.0.0.5 area 0 from 172.23.1.121
Sep 17 01:02:18.560: OSPF-100 HELLO Et0/2: Rcv hello from 10.0.0.120 area 0 172.23.3.1
Sep 17 01:02:19.061: OSPF-100 HELLO Et0/1: Rcv hello from 10.0.0.120 area 0 172.23.2.1
Sep 17 01:02:19.539: OSPF-100 HELLO Et0/1: Send hello to 224.0.0.5 area 0 from 172.23.2.2

```

```
c5921_ubuntu32_2#
```

```
Sep 17 01:02:22.437: OSPF-100 HELLO Et0/3: Send hello to 224.0.0.5 area 0 from 172.23.4.2
```

```
Sep 17 01:02:23.652: OSPF-100 HELLO Et0/2: Send hello to 224.0.0.5 arundebug all
```

```
All possible debugging has been turned off
```

```
c5921_ubuntu32_2#
```

```
Sep 17 01:02:26.324: OSPF-100 HELLO Et0/0: Rcv hello from 10.0.0.120 area 0 172.23.1.120
```

```
Sep 17 01:02:26.699: OSPF-100 HELLO Et0/3: Rcv hello from 10.0.0.120 area 0 172.23.4.1
```



```
c5921_ubuntu32_2#
```

```
Type exit in order to exit from the telnet session:
```

```
c5921_ubuntu32_2#exit
```

Feel free to explore further on your own.

For additional ideas or to contribute some of your own, please visit our DevNet Community called 5921 ESR Development: <https://communities.cisco.com/community/developer/networking/internet-of-things/embedded-networks/5921-esr-development>

Based on each topic, look for the following specific subspaces:

- For other router configurations to try or if you have created something interesting that you may want to share, please visit:

[Contributed Sample Configurations for the 5921](#)

- If you would like to share an interesting Python application that works with the 5921, please visit:

[Code Drop - 5921 Complementary Applications](#)

This concludes this Demonstration Lab activity.

Scenario 2: Installation Lab Exercise

Ensure you are still connected to Workstation 1 from the Scenario 1 exercise or that you follow the instructions to connect to the dCloud environment via AnyConnect.

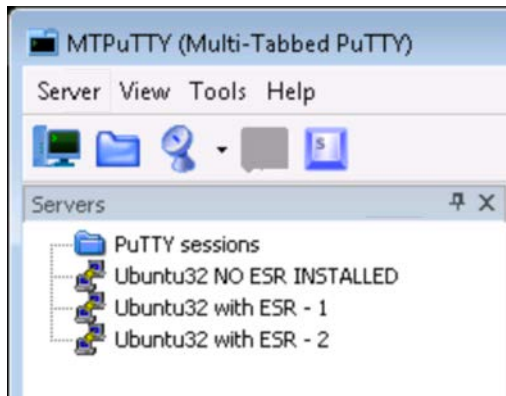
- Once logged into the desktop, double click to launch Multi-Tab PuTTY:

Figure 6. Multi-Tab PuTTY Icon



- Here you will find preconfigured links to launch ssh sessions to the 3 linux VM's. Double-click on these links under **Servers** to connect to the corresponding VM. Please give the VM a minute to respond after launching a new session.

Figure 7. Pre-configured PuTTY Sessions



- If you are connecting via AnyConnect, the IP addresses for the management ports of the VMs are as follows:

- Ubuntu32 NO ESR installed -198.18.134.122
- Ubuntu32 with ESR - 1 -198.18.134.120
- Ubuntu32 with ESR - 2 -198.18.134.121

If you connect via AnyConnect, you may launch ssh sessions via a ssh client of your choice.

NOTE: If you wish to easily capture the configuration files for future reference, AnyConnect would be the preferred method of access.

- Connect via ssh to Ubuntu32 NO ESR installed – 198.18.134.122 to proceed with the c5921 installation.

```
administrator@ubuntu32:~$ sudo su
[sudo] password for administrator:
```

(Enter password **C1sco12345**)

```

root@ubuntu32:/home/administrator# ls -l
root@ubuntu32:/home/administrator# ls -l
total 164324
-rw-r--r-- 1 root root 168263680 Jul 21 22:21 c5921i86-universalk9-tar.SPA.155-2.T
root@ubuntu32:/home/administrator# cd /opt
root@ubuntu32:/opt# ls
root@ubuntu32:/opt# mkdir cisco
root@ubuntu32:/opt# cd cisco
root@ubuntu32:/opt/cisco# mkdir c5921
root@ubuntu32:/opt/cisco# ls
c5921
root@ubuntu32:/opt/cisco# cd /home/administrator
root@ubuntu32:/home/administrator# cp c5921i86-universalk9-tar.SPA.155-2.T /opt/cisco/c5921
root@ubuntu32:/home/administrator# cd /opt/cisco/c5921
root@ubuntu32:/opt/cisco/c5921# ls
c5921i86-universalk9-tar.SPA.155-2.T
root@ubuntu32:/opt/cisco/c5921# tar -xf c5921i86-universalk9-tar.SPA.155-2.T
root@ubuntu32:/opt/cisco/c5921# ls
c5921i86-universalk9-ms.155-2.T c5921i86-universalk9-tar.SPA.155-2.T
root@ubuntu32:/opt/cisco/c5921# cd c5921i86-universalk9-ms.155-2.T
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp c5921i86-universalk9-ms.S*
../
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp libdyncs.so ../
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# ls
c5921i86-universalk9-ms.md5 c5921-swr-init.sh libdyncs.so RELEASE_NOTES_C5921.txt
swroptions.1 swr_reload swrvcon
c5921i86-universalk9-ms.SPA FAQ_C5921.txt README_C5921.txt swr-application.1
SWROPTIONS.example.txt swr_reload.1 swrvcon.1
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp swrvcon ../
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp swr_reload ../
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp c5921-swr-init.sh ../
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cp SWROPTIONS.example.txt
../SWROPTIONS
root@ubuntu32:/opt/cisco/c5921/c5921i86-universalk9-ms.155-2.T# cd ..
root@ubuntu32:/opt/cisco/c5921# ls
c5921i86-universalk9-ms.155-2.T c5921i86-universalk9-ms.SPA c5921i86-universalk9-tar.SPA.155-
2.T c5921-swr-init.sh libdyncs.so SWROPTIONS swr_reload swrvcon
root@ubuntu32:/opt/cisco/c5921# chmod 744 SWROPTIONS
root@ubuntu32:/opt/cisco/c5921# nano SWROPTIONS

```

← We are careful to check each step of the path in the example.

The options set by SWROPTIONS are explained in detail in Chapter 3 of the *Cisco 5921 Embedded Services Router Integration Guide*.

Set the number of Ethernet slots to 1 and model all of the Ethernet ports as shown for e0/0. Finished SWROPTIONS files are shown in the appendices of this guide and are referenced for the rest of this scenario.

```
##### EXAMPLE SWROPTIONS FILE #####
```

```

soft-rommon= ./
launchapp=c5921i86-universalk9-ms.SPA
ram=512
ethernet-slots=1    ← Change 2 to 1. In Appendix B, this will remain 2 when we show the use of the tap0 interface.

node-lock-type=machine

### FILE MAPPING SECTION ###

[filemap]
ios=flash0
linux=/opt/cisco/c5921/

```

```
access=rw
```

```
### INTERFACE MAPPING SECTION ###
```

```
# Map Linux eth0 to IOS e0/0, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/0 in IOS.
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth0 to IOS e0/0.
# Push changes like speed/duplex/MAC of IOS e0/0 to Linux eth0.
```

```
[interface]
linux=eth0
ios=e0/0
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs are available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence could
# pushed to Linux eth1.
```

```
[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true ← Change false to true.
```

```
# Map Linux eth2 to IOS e0/2, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/2 in IOS
# DO NOT Monitor/Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth2 to IOS e0/2.
# DO NOT Push IOS e0/2 changes to Linux eth2.
```

```
[interface]
linux=eth2
ios=e0/2
type=raw
promiscuous=true
monitor-state=true ← Change false to true.
push-mon-int=true
```

```
[interface] ← Add this interface section.
linux=eth3
ios=e0/3
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# NOTE:
#
# The tap interface will not be used for the implementation lab part of demo on the
```

```
# third router where the user performs the installation, so we have commented
# out its config lines. For router ESR 1 where we have the Linux host act as an NTP
# proxy, we show a usage of tap0.
#
# Please compare Appendix A and Appendix B to see the minor changes required to
# implement this useful feature.
```

```
# [interface]
# linux=tap0
# ios=e1/0
# type=tap
# monitor-state=true
```

Type CTRL+O to save your changes and CTRL+X to exit the **nano** editor. Check that your SWROPTIONS file has been properly saved with all the changes by using the **cat** command:

```
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS
```

```
root@ubuntu32:/opt/cisco/c5921# ln -s /opt/cisco/c5921/c5921-swr-init.sh /etc/init.d/
root@ubuntu32:/opt/cisco/c5921# chmod 744 /etc/init.d/c5921-swr-init.sh
```

5. Start the Cisco 5921 ESR with the following command:

```
root@ubuntu32:/opt/cisco/c5921# /etc/init.d/c5921-swr-init.sh start
```

```
Loading Image:./c5921i86-universalk9-ms.SPA
./c5921i86-universalk9-ms.SPA running SWR the background, pid=1183, SWR=1186
Child process will exec swr image now....
* starting ./swr_reload...
```

6. Access the virtual console with the following command. Note that you must issue the command as root:

```
root@ubuntu32:/opt/cisco/c5921# ./swrvcon 100
```

NOTE: As with physical Cisco routers, there are pauses in the router's console boot messages while IOS boots. There is no output during these pauses on the console. This is normal, please be patient while IOS boots up after you access the virtual console.

```
--- System Configuration Dialog ---
```

```
Would you like to enter the initial configuration dialog? [yes/no]:
% Please answer 'yes' or 'no'.
Would you like to enter the initial configuration dialog? [yes/no]: yes
```

```
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
```

```
Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system
```

```
Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:
```

```
Enter host name [Router]: c5921_ubuntu32
```

The enable secret is a password used to protect access to privileged EXEC and configuration modes. This password, after entered, becomes encrypted in the configuration.

Enter enable secret: **cisco**

The enable password is used when you do not specify an enable secret password, with some older software versions, and some boot images.

Enter enable password: **cisco**

% Please choose a password that is different from the enable secret

Enter enable password: **cisco**

The virtual terminal password is used to protect access to the router over a network interface.

Enter virtual terminal password: **cisco**

Configure SNMP Network Management? [yes]:

Community string [public]:

Current interface summary

Any interface listed with OK? value "NO" does not have a valid configuration

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	NO	unset	up	up
Ethernet0/1	unassigned	NO	unset	up	up
Ethernet0/2	unassigned	NO	unset	up	up
Ethernet0/3	unassigned	NO	unset	up	up

Enter interface name used to connect to the management network from the above interface summary: **Ethernet0/0**

Configuring interface Ethernet0/0:

Configure IP on this interface? [yes]:

IP address for this interface: **172.23.1.165**

Subnet mask for this interface [255.255.0.0] : **255.255.255.0**

Class B network is 172.23.0.0, 24 subnet bits; mask is /24

The following configuration command script was created:

```
hostname c5921_ubuntu32
enable secret 5 $1$x.60$4UZX1B51YLF1DeOul7GiV1
enable password cisco
line vty 0 4
password cisco
snmp-server community public
!
!
interface Ethernet0/0
no shutdown
ip address 172.23.1.165 255.255.255.0
no mop enabled
!
interface Ethernet0/1
shutdown
no ip address
!
interface Ethernet0/2
shutdown
no ip address
!
interface Ethernet0/3
shutdown
no ip address
!
```

```
end
```

```
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.
```

```
Enter your selection [2]: 2
The enable password you have chosen is the same as your enable secret.
This is not recommended. Re-enter the enable password.
```

```
Building configuration...
```

From here onward, we assume that you are at the familiar IOS command line interface (CLI). Please refer to the appendices for complete router configurations. Here are a few minor differences and helpful tips:

- On a hardware router, you may use Control-C to end an edit session. If you type a Control-C in the CLI of the 5921, you will end your swrvcon session and be returned to the Linux command prompt. Instead, use Control-Z or the “exit” command. If you forget, there is an easy recovery. Simply arrow up to your last Linux command “./swrvcon 100”. You will be right back to where you left off.
- If you execute a “dir” command, you will notice the device is not “flash:”, instead the output says, “Directory of unix:” and this equates to /opt/cisco/c5921. The router configuration is stored in nvram_00100, effectively this is a virtual nvram device.
- If your running configuration contains many digital certificates, try “sh running-config brief” instead of “sh run” or “wr t”. Certificates will be displayed in an abbreviated form instead of large blocks of hex digits.
- The Cisco 5921 ESR does not support physical serial interfaces. However, access via the vty’s using telnet and ssh works the same as a hardware router.

Setting the Cisco 5921 to launch at Linux boot up:

You should see this output as the router is booting.

```
root@ubuntu32:/opt/cisco/c5921# update-rc.d c5921-swr-init.sh start 30 3 5 . stop 80 0 1 2 6 .
Adding system startup for /etc/init.d/c5921-swr-init.sh ...
/etc/rc0.d/K80c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
/etc/rc1.d/K80c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
/etc/rc2.d/K80c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
/etc/rc6.d/K80c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
/etc/rc3.d/S30c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
/etc/rc5.d/S30c5921-swr-init.sh -> ../init.d/c5921-swr-init.sh
root@ubuntu32:/opt/cisco/c5921#
```

1. To confirm that the Cisco 5921 ESR is running you may use the following command at the Linux prompt.

If you see this output, the router is running:

```
c5921-Ubuntu#root@ubuntu:/opt/cisco/c5921# ps -ef | grep c5921
root      2836  1902  2 18:21 pts/0      00:06:12 ./c5921i86-universalk9-ms.SPA -L 100 -o
/opt/cisco/c5921/SWROPTIONS
root      3857  2695  0 22:34 pts/0      00:00:00 grep --color=auto c5921
root@ubuntu:/opt/cisco/c5921#
```

If you see this output, the router is NOT running:

```
root@ubuntu32:/etc/network# ps -ef | grep c5921
root      1258  1242  0 22:38 pts/0    00:00:00 grep --color=auto c5921
root@ubuntu32:/etc/network#
```

2. Issue a reboot command in order to sync the c5921 and Linux VM. You will be disconnected from the VM. Please allow about 2 - 3 minutes for the reboot and perform the commands described in step 1 above to confirm that the 5921 is launching automatically.

```
root@ubuntu32:/etc/network# reboot
```

Wait 2 - 3 minutes

```
root@ubuntu32:/etc/network# ps -ef | grep c5921
```

This concludes this Installation Lab activity.

Scenario 3: Troubleshooting Lab Exercise

In order to provide some troubleshooting experience, we will now break the network, show why it broke, and finally explain what we need to do to fix it.

We will assume that you are familiar with network troubleshooting using show and debug commands. In fact, in the demonstration section of this lab, we saw that the Cisco 5921 ESR behaves very similar to what one would expect on a typical Cisco Integrated Services Router.

We will therefore focus on one of the unique aspects of this product, one of the critical configuration files added to Linux that controls port mappings between Linux and the 5921. This file is "SWROPTIONS" – [software router options](#).

1. First, we connect to ESR1:

```
c5921_ubuntu32_1#sh run brief
Building configuration...

Current configuration : 2560 bytes
!
! No configuration change since last restart
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service call-home
!
hostname c5921_ubuntu32_1
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$x.6O$4UZX1B51YLF1DeOul7GiV1
enable password cisco
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
call-home
  ! if contact email address in call-home is configured as sch-smart-licensing@cisco.com
  ! The email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications
  contact-email-addr sch-smart-licensing@cisco.com
  profile "CiscoTAC-1"
    active
    destination transport-method http
    no destination transport-method email
!
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
```

```

crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
  certificate ca 01
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
redundancy
!
!
interface Loopback0
  ip address 10.0.0.120 255.255.255.255
!
interface Ethernet0/0
  ip address 172.23.1.120 255.255.255.0
  duplex full
  speed 1000
  no mop enabled
!
interface Ethernet0/1
  ip address 172.23.2.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet0/2
  ip address 172.23.3.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet0/3
  ip address 172.23.4.1 255.255.255.0
  duplex full
  speed 1000
!

c5921_ubuntu32_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
c5921_ubuntu32_1(config)#int e 0/1
c5921_ubuntu32_1(config-if)#speed 100
c5921_ubuntu32_1(config-if)#do sh run int e 0/1
Building configuration...

Current configuration : 90 bytes
!
interface Ethernet0/1
  ip address 172.23.2.1 255.255.255.0
  duplex full
  speed 100
end

c5921_ubuntu32_1(config-if)#exit
c5921_ubuntu32_1(config)#

```

← Notice that speed and duplex are showing as configurable options.

← Notice that speed and duplex are showing as configurable options.

← Notice that speed and duplex are showing as configurable options.

← Notice that speed and duplex are showing as configurable options.

← Yes, we can make changes.

2. Type Control C to exit to a Linux prompt.

Here we will make the assumption that the person who initially installed the 5921 simply copied the sample SWROPTIONS file without considering that some settings may not be appropriate for their environment. Our goal is to have speed and duplex

configurable on all Ethernet ports but this would not be the case if we simply copy SWROPTIONS.example.txt to the active SWROPTIONS file.

```
root@ubuntu32:/opt/cisco/c5921# cp SWROPTIONS SWROPTIONS.save ← Save the working copy first.
ubuntu32:/opt/cisco/c5921#
root@ubuntu32:/opt/cisco/c5921# ls
c5921i86-universalk9-ms.155-2.T c5921i86-universalk9-tar.SPA.155-2.T libdyncs.so
staticroutes.sh SWROPTIONS.save swrvcon
c5921i86-universalk9-ms.SPA c5921-swr-init.sh nvram_00100 SWROPTIONS
swr_reload
```

3. Now we will break it! We will copy the default example to the active SWROPTIONS file with no adjustments for our environment.

```
root@ubuntu32:/opt/cisco/c5921# cp c5921i86-universalk9-ms.155-2.T/SWROPTIONS.example.txt
SWROPTIONS
```

4. Next, we will do a fully cold restart of the 5921.

```
root@ubuntu32:/opt/cisco/c5921# ps -ef | grep 5921
root      953      1  1 Sep16 ?          00:28:00 ./c5921i86-universalk9-ms.SPA -L 100 -o
/opt/cisco/c5921/SWROPTIONS
root      2005    1909  0 09:20 pts/1      00:00:00 grep --color=auto 5921
root@ubuntu32:/opt/cisco/c5921# kill 953
root@ubuntu32:/opt/cisco/c5921# ps -ef | grep 5921
root      2007    1909  0 09:21 pts/1      00:00:00 grep --color=auto 5921
root@ubuntu32:/opt/cisco/c5921# /etc/init.d/c5921-swr-init.sh start
```

```
Loading Image:./c5921i86-universalk9-ms.SPA
./c5921i86-universalk9-ms.SPA running SWR the background, pid=2012, SWR=2015
Child process will exec swr image now....
* starting ./swr_reload...
```

```
Adding Static Routes          ← These are static routes that we created in another script for the tap interface.
SIOCADDRT: File exists       ← The system is telling us the routes already exist and we are trying to add them again.
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
```

5. Now we will reconnect to the console and observe what happened. The first thing we see is that during the boot messages the router is having issues with the duplex and speed settings that we configured manually.

```
root@ubuntu32:/opt/cisco/c5921# ./swrvcon 100
Cisco C5921 (Intel-x86) processor with 381071K bytes of memory.
Processor board ID 100
8 Ethernet interfaces
512K bytes of NVRAM.
Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

duplex full
^
% Invalid input detected at '^' marker.

speed 1000
```

```

^
% Invalid input detected at '^' marker.

duplex full
^
% Invalid input detected at '^' marker.

speed 1000
^
% Invalid input detected at '^' marker.
Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

Press RETURN to get started!

*Sep 17 16:21:55.271: %LICENSE_C5920-6-UDI_RETRIEVED: Existing UDI retrieved. UDI: CISCO5921-
K9:91J81443B9I
*Sep 17 16:21:55.441: %LICENSE_C5920-6-DEFAULT_BANDWIDTH_ACTIVATED: Forwarding bandwidth limited
to 8 Kbps
*Sep 17 16:21:55.442: %SMART_LIC-6-AGENT_READY: Smart Agent for Licensing is initialized
*Sep 17 16:21:55.442: %SMART_LIC-6-AGENT_ENABLED: Smart Agent for Licensing is enabled
*Sep 17 16:21:55.442: %LICENSE_C5920-6-DEFAULT_BANDWIDTH_ACTIVATED: Forwarding bandwidth limited
to 8 Kbps
*Sep 17 16:22:02.304: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0, changed state
to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet0/2, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet0/3, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet1/0, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet1/1, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet1/2, changed state to up
*Sep 17 16:22:02.309: %LINK-3-UPDOWN: Interface Ethernet1/3, changed state to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/1, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/2, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/3, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/1, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/2, changed state
to up
*Sep 17 16:22:04.362: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/3, changed state
to up
*Sep 17 16:22:06.385: %LICENSE_C5920-6-LICENSE_ACTIVATED: Installed license for feature c5921-
x86-level3 now in use. Forwarding bandwidth limited to 50 Mbps
*Sep 17 16:22:07.474: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state
to up
*Sep 17 16:22:07.503: %SYS-5-CONFIG_I: Configured from memory by console
*Sep 17 16:22:08.292: %SYS-5-RESTART: System restarted --
Cisco IOS Software, C5921 Software (C5921_I86-UNIVERSALK9-M), Version 15.5(2)T, RELEASE SOFTWARE
(fcl)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 25-Mar-15 15:49 by prod_rel_team

```

```

*Sep 17 16:22:08.298: %SNMP-5-COLDSTART: SNMP agent on host c5921_ubuntu32_1 is undergoing a
cold start
*Sep 17 16:22:08.536: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Sep 17 16:22:08.536: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Sep 17 16:22:08.955: %LINK-5-CHANGED: Interface Ethernet1/1, changed state to administratively
down
*Sep 17 16:22:08.955: %LINK-5-CHANGED: Interface Ethernet1/2, changed state to administratively
down
*Sep 17 16:22:09.160: %LINK-5-CHANGED: Interface Ethernet1/3, changed state to administratively
down
*Sep 17 16:22:09.993: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/1, changed state
to down
*Sep 17 16:22:09.993: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/2, changed state
to down
*Sep 17 16:22:10.198: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/3, changed state
to down
*Sep 17 16:22:14.243: %OSPF-5-ADJCHG: Process 100, Nbr 10.0.0.121 on Ethernet0/2 from LOADING to
FULL, Loading Done
*Sep 17 16:22:14.243: %OSPF-5-ADJCHG: Process 100, Nbr 10.0.0.121 on Ethernet0/1 from LOADING to
FULL, Loading Done
*Sep 17 16:22:14.243: %OSPF-5-ADJCHG: Process 100, Nbr 10.0.0.121 on Ethernet0/0 from LOADING to
FULL, Loading Done
*Sep 17 16:22:20.322: %LINK-3-UPDOWN: Interface Ethernet0/3, changed state to down
*Sep 17 16:22:21.322: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/3, changed state
to downUnable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

c5921_ubuntu32_1>Unable to get settings via ethtool. Not setting: Operation not supported
[OK]
*Sep 17 16:22:25.930: %CALL_HOME-6-CALL_HOME_ENABLED: Call-home is enabled by Smart Agent for
Licensing.
*Sep 17 16:22:25.932: %PKI-6-CONFIGAUTOSAVE: Running configuration saved to NVRAM
c5921_ubuntu32_1>en
Password:
c5921_ubuntu32_1#sh run brief
Building configuration...
Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

Unable to get settings via ethtool. Not setting: Operation not supported
Unable to get settings via ethtool. Not setting: Operation not supported

Current configuration : 2463 bytes
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service call-home
!
hostname c5921_ubuntu32_1
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$x.6O$4UZX1B51YLF1DeOul7GiV1
enable password cisco
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure

```

```

no mmi pvc
mmi snmp-timeout 180
call-home
  ! if contact email address in call-home is configured as sch-smart-licensing@cisco.com
  ! The email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications
  contact-email-addr sch-smart-licensing@cisco.com
  profile "CiscoTAC-1"
    active
    destination transport-method http
    no destination transport-method email
  !
  !
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
  certificate ca 01
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
redundancy
!
!
interface Loopback0
  ip address 10.0.0.120 255.255.255.255
!
interface Ethernet0/0
  ip address 172.23.1.120 255.255.255.0
  duplex full
  speed 1000
  no mop enabled
!
interface Ethernet0/1
  ip address 172.23.2.1 255.255.255.0 ← We see speed and duplex settings are missing from some ports.
!
interface Ethernet0/2
  ip address 172.23.3.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet0/3
  ip address 172.23.4.1 255.255.255.0 ← We see speed and duplex settings are missing from some ports.
!
interface Ethernet1/0
  description TAP interface
  ip address 198.18.134.131 255.255.192.0
!
interface Ethernet1/1
  no ip address
  shutdown
!
--More--

```

6. Type "q" to stop the output.

```

c5921_ubuntu32_1#
c5921_ubuntu32_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
c5921_ubuntu32_1(config)#int e 0/1
c5921_ubuntu32_1(config-if)#speed 100 ← We see the router will not allow changes for speed or duplex!
      ^
% Invalid input detected at '^' marker.

c5921_ubuntu32_1(config-if)#exit
c5921_ubuntu32_1(config)#exit
c5921_ubuntu32_1#
Sep 17 16:28:03.013: %SYS-5-CONFIG_I: Configured from console by console
c5921_ubuntu32_1#

```

NOTE: SWROPTIONS is a very important file and has a profound effect on how the Cisco 5921 ESR will behave in the Linux environment where it has been installed.

Now we will compare key differences between the working SWROPTIONS (now contained in “SWROPTIONS.save”) and the default example that we copied earlier.

7. Type Control C to exit to a Linux prompt.

Here is our working file:

```
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS.save
```

(Output abbreviated – key section shown.)

```

# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.

[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

```

8. We can see that configuring an interface in the manner shown above allows the interface to be configured from both Linux and IOS. So what did we change?

```
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS
```

(Output abbreviated – key section shown.)

```

# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs NOT available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could not be changed by IOS and hence not

```

```
# pushed to Linux eth1.
```

```
[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=false
```

9. Reading the comments, we see that changing the **push-mon-int** parameter from true to **false** affected our ability to change speed and duplex on the Ethernet port from IOS. While there may be cases where such a restriction may be desirable, that was not our design goal. Let us fix it.

10. Since we had a working file, we can simply copy the saved back to the active.

```
root@ubuntu32:/opt/cisco/c5921# cp SWROPTIONS.save SWROPTIONS
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS
```

(Output abbreviated – key section shown.)

```
# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.
```

```
[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

11. Looks correct. Cold boot the router and retest.

```
root@ubuntu32:/opt/cisco/c5921# ps -ef | grep 5921
root      2015      1  2 09:21 pts/1    00:03:26 ./c5921i86-universalk9-ms.SPA -L 100 -o
/opt/cisco/c5921/SWROPTIONS
root      2063    1909  0 11:33 pts/1      00:00:00 grep --color=auto 5921
root@ubuntu32:/opt/cisco/c5921# kill 2015
root@ubuntu32:/opt/cisco/c5921# ps -ef | grep 5921
root      2065    1909  0 11:34 pts/1      00:00:00 grep --color=auto 5921
root@ubuntu32:/opt/cisco/c5921# /etc/init.d/c5921-swr-init.sh start
```

```
Loading Image:./c5921i86-universalk9-ms.SPA
./c5921i86-universalk9-ms.SPA running SWR the background, pid=2069, SWR=2072
Child process will exec swr image now....
* starting ./swr_reload...
```

```
Adding Static Routes
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
SIOCADDRT: File exists
```



```

SIOCADDRT: File exists
SIOCADDRT: File exists
root@ubuntu32:/opt/cisco/c5921# ./swrvcon 100
Cisco C5921 (Intel-x86) processor with 381071K bytes of memory.
Processor board ID 100
8 Ethernet interfaces
512K bytes of NVRAM.

```

--- Router rebooting - Output abbreviated ---

```

!
interface Ethernet0/0
 ip address 172.23.1.120 255.255.255.0
 duplex full
 speed 1000
 no mop enabled
!
interface Ethernet0/1
 ip address 172.23.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface Ethernet0/2
 ip address 172.23.3.1 255.255.255.0
 duplex full
 speed 1000
!
interface Ethernet0/3
 ip address 172.23.4.1 255.255.255.0
 duplex auto
 speed auto
!
interface Ethernet1/0
 description TAP interface
 ip address 198.18.134.131 255.255.192.0
!

```

12. Let us try to alter the speed on Ethernet0/1.

```

c5921_ubuntu32_1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
c5921_ubuntu32_1(config)#int e 0/1
c5921_ubuntu32_1(config-if)#speed 1000
c5921_ubuntu32_1(config-if)#do sh run int e 0/1
Building configuration...

Current configuration : 91 bytes
!
interface Ethernet0/1
 ip address 172.23.2.1 255.255.255.0
 duplex auto
 speed 1000
end

c5921_ubuntu32_1(config-if)#exit
c5921_ubuntu32_1(config)#exit
c5921_ubuntu32_1#
Sep 17 18:48:35.071: %SYS-5-CONFIG_I: Configured from console by console
c5921_ubuntu32_1#

```

You may have encountered the following message during this exercise:

```
"Unable to get settings via ethtool. Not setting: Operation not supported"
```

This message is telling you that the interface configuration for Linux and the 5921 are out of sync with each other. This can be caused by both the speed and duplex settings. Consider the following and ask yourself if they are EXACTLY the same.

Here are the settings in IOS:

```
interface Ethernet0/0
 ip address 172.23.1.120 255.255.255.0
 duplex full
 speed 1000
 no mop enabled
!
interface Ethernet0/1
 ip address 172.23.2.1 255.255.255.0
 duplex auto
 speed 1000
!
interface Ethernet0/2
 ip address 172.23.3.1 255.255.255.0
 duplex full
 speed 1000
!
interface Ethernet0/3
 ip address 172.23.4.1 255.255.255.0
 duplex auto
 speed 1000
```

Here is what we set in the Linux configuration file `/etc/network/interfaces`:

```
# These interfaces will be used by the Cisco 5921 ESR.

auto eth0
iface eth0 inet static
pre-up /sbin/ethtool -s eth0 speed 1000 duplex full

auto eth1
iface eth1 inet static
pre-up /sbin/ethtool -s eth1 speed 1000 duplex full

auto eth2
iface eth2 inet static
pre-up /sbin/ethtool -s eth2 speed 1000 duplex full

auto eth3
iface eth3 inet static
pre-up /sbin/ethtool -s eth3 speed 1000 duplex full
```

13. Look at the duplex settings for interface Ethernet0/1 and interface Ethernet0/3. The settings “duplex auto” and “duplex full” are NOT the same! We can resolve this by making the two the same and rebooting Linux. So lets go back to our original configuration in IOS:

```
interface Ethernet0/0
 ip address 172.23.1.120 255.255.255.0
 duplex full
 speed 1000
 no mop enabled
!
interface Ethernet0/1
 ip address 172.23.2.1 255.255.255.0
 duplex full
 speed 1000
!
interface Ethernet0/2
 ip address 172.23.3.1 255.255.255.0
 duplex full
 speed 1000
!
interface Ethernet0/3
 ip address 172.23.4.1 255.255.255.0
 duplex full
```

← Duplex set to “full” not “auto”

← Duplex set to “full” not “auto”

```
speed 1000
!
```

Save the router changes.

From Linux

```
root@ubuntu32:/opt/cisco/c5921# reboot
root@ubuntu32:/opt/cisco/c5921#
Broadcast message from administrator@ubuntu32

    (/dev/pts/0) at 13:47 ...
```

From what we see above, we are now back to the desired state where speed and duplex of the Ethernet interfaces can be changed by IOS. Remember to make the corresponding changes in Linux afterwards. Such that Linux and IOS are in sync when you reboot.

We have learned that a key difference between working with one of the Cisco Integrated Services Routers and the Cisco 5921 Embedded Services Router is that we must correctly configure the files that integrate Cisco IOS into the Linux host. What we have gained for our extra effort is the ability to bring Cisco Technology into any form factor where our integration partners create hardware platforms that Cisco does not manufacture whether it is a device that goes into deep space or the bottom of the ocean or a wearable device.

Feel free to experiment with your own variations. Perhaps you could switch the port mappings in SWROPTIONS and then observe the effects in IOS. Be sure that you have saved your working copy.

While the next section, Scenario 4, Install and Configure Ubuntu Server Virtual Machine, is mainly reference in that all of the steps have been done for you, you can still practice checking dependencies and looking at the low level NIC configuration. This is another area where mistakes can happen. It would be wise to look but not make changes as you could create a stranded VM that becomes unreachable. At that point, your only option will be to end and restart the dCloud lab session.

Frequently Asked Questions:

Q: What do I do if I need help or suspect something is wrong with my Linux copy?

A: When you downloaded the installation files, did you check for download errors using the published md5 or sha?

This is one disadvantage of using a “net install”. With a net install the advantage is that you can create an installation disk on a CD instead of a DVD. When the installer actually runs, your computer must be connected to the Internet as it will download the necessary packages after you specify the build model. (minimal, server, minimal desktop, developer, etc.) However, you never get the opportunity to check the entire image holistically.

Most of the major Linux distributions have an official documentation site. Here is the link for Ubuntu:

Official Ubuntu Documentation

<https://help.ubuntu.com/>

Most of the major Linux distributions have a site for reporting and searching bugs. Here are links for Ubuntu:

ReportingBugs

<https://help.ubuntu.com/community/ReportingBugs>

<https://launchpad.net/ubuntu/+bugs>

Also check their Wiki: “This user-created and maintained wiki is a reference for various Ubuntu-related Howto's, tips, tricks and hacks. “

CommunityHelpWiki

<https://help.ubuntu.com/community/CommunityHelpWiki>

This concludes the Troubleshooting Lab Section.

Scenario 4: Reference - Install and Configure Ubuntu Server Virtual Machine

NOTE: The installation of the Ubuntu server has already been done for you in Cisco dCloud, but is shown here mainly as a reference, you can still practice checking dependencies and looking at the low-level NIC configuration. Please do NOT perform the steps in this chapter or the VMs may be corrupted.

Linux Prequalification:

You begin by going to the administrator home dir and find that the tftp modules are not installed. This is a very minimal installation to start with.

```
root@ubuntu32:~# cd /home/administrator
root@ubuntu32:/home/administrator# tftp 172.23.1.46
The program 'tftp' can be found in the following packages:
 * tftp-hpa
 * tftp
Try: apt-get install <selected package>
```

1. Download and install the tftp package.

```
root@ubuntu32:/home/administrator# apt-get install tftp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 tftp
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.6 kB of archives.
After this operation, 81.9 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/trusty/universe tftp i386 0.17-18ubuntu2 [15.6 kB]
Fetched 15.6 kB in 0s (47.0 kB/s)
Selecting previously unselected package tftp.
(Reading database ... 58518 files and directories currently installed.)
Preparing to unpack ../tftp_0.17-18ubuntu2_i386.deb ...
Unpacking tftp (0.17-18ubuntu2) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up tftp (0.17-18ubuntu2) ...
```

2. Now that we have tftp installed, we will use it to load the c5921 tar file.

```
root@ubuntu32:/home/administrator# tftp 172.23.1.46 ← A server that has the c5921 tar file
tftp> ? (Address may be different in your environment.)
```

Commands may be abbreviated. Commands are:

```
connect    connect to remote tftp
mode       set file transfer mode
put        send file
get        receive file
quit       exit tftp
verbose    toggle verbose mode
trace      toggle packet tracing
```

```

status      show current status
binary      set mode to octet
ascii       set mode to netascii
rexmt       set per-packet retransmission timeout
timeout     set total retransmission timeout
?           print help information

```

```

tftp> binary
tftp> get c5921i86-universalk9-tar.SPA.155-2.T

```

```

Received 168263680 bytes in 120.5 seconds
tftp> quit
root@ubuntu32:/home/administrator# ls -al

```

- The `ls -al` command is issued to check that the package was downloaded properly. Be sure to check the package name, location, and size as shown below.

```

total 164348
drwxr-xr-x 3 administrator administrator 4096 Jul 21 22:19 .
drwxr-xr-x 3 root root 4096 Jul 21 22:06 ..
-rw-r--r-- 1 administrator administrator 220 Jul 21 22:06 .bash_logout
-rw-r--r-- 1 administrator administrator 3637 Jul 21 22:06 .bashrc
-rw-r--r-- 1 root root 168263680 Jul 21 22:21 c5921i86-universalk9-
tar.SPA.155-2.T
drwx----- 2 administrator administrator 4096 Jul 21 22:07 .cache
-rw-r--r-- 1 administrator administrator 675 Jul 21 22:06 .profile
root@ubuntu32:/home/administrator#

```

- We have the Cisco 5921 ESR tar file loaded, but may not have all of the necessary dependencies installed. The following output is from a working system, we will use this to compare and see if all of the dependencies are installed our server.

Finding the libraries that we need:

```

root@ubuntu:/opt/cisco/c5921# ldd c5921i86-universalk9-ms.SPA
linux-gate.so.1 => (0xb7798000)
libnsl.so.1 => /lib/i386-linux-gnu/libnsl.so.1 (0xb7768000)
libdl.so.2 => /lib/i386-linux-gnu/libdl.so.2 (0xb7763000)
libpthread.so.0 => /lib/i386-linux-gnu/libpthread.so.0 (0xb7746000)
librt.so.1 => /lib/i386-linux-gnu/librt.so.1 (0xb773d000)
libdyncs.so => not found <- This library comes with the c5921 ESR tar file.
libm.so.6 => /lib/i386-linux-gnu/libm.so.6 (0xb76f7000)
libgcc_s.so.1 => /lib/i386-linux-gnu/libgcc_s.so.1 (0xb76da000)
libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb752b000)
/lib/ld-linux.so.2 (0xb779b000)
root@ubuntu:/opt/cisco/c5921#

root@ubuntu:/opt/cisco/c5921# ldd swrvcon
linux-gate.so.1 => (0xb77be000)
libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75f9000)
/lib/ld-linux.so.2 (0xb77c1000)
root@ubuntu:/opt/cisco/c5921#

root@ubuntu:/opt/cisco/c5921# ldd swr_reload
linux-gate.so.1 => (0xb76ff000)
libpthread.so.0 => /lib/i386-linux-gnu/libpthread.so.0 (0xb76cc000)
libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb751e000)
/lib/ld-linux.so.2 (0xb7702000)
root@ubuntu:/opt/cisco/c5921#

```

5. Coming back to our installation, we can do the following to show the installed libraries. We will capture the output, and we will highlight what we find in the list:

Understanding what is installed:

```

root@ubuntu32:~# /sbin/ldconfig -p
204 libs found in cache `/etc/ld.so.cache'
  libz.so.1 (libc6) => /lib/i386-linux-gnu/libz.so.1
  libxtables.so.10 (libc6) => /lib/libxtables.so.10
  libxml2.so.2 (libc6) => /usr/lib/i386-linux-gnu/libxml2.so.2
  libxcb.so.1 (libc6) => /usr/lib/i386-linux-gnu/libxcb.so.1
  libxapian.so.22 (libc6, hwcap: 0x000000004000000) => /usr/lib/sse2/libxapian.so.22
  libxapian.so.22 (libc6) => /usr/lib/libxapian.so.22
  libwrap.so.0 (libc6) => /lib/i386-linux-gnu/libwrap.so.0
  libwind.so.0 (libc6) => /usr/lib/i386-linux-gnu/libwind.so.0
  libvmGuestLibJava.so (libc6,x86-64) => /usr/lib/vmware-
tools/lib64/libvmGuestLibJava.so/libvmGuestLibJava.so
  libvmGuestLibJava.so (libc6) => /usr/lib/vmware-
tools/lib32/libvmGuestLibJava.so/libvmGuestLibJava.so
  libvmGuestLib.so (libc6,x86-64) => /usr/lib/vmware-
tools/lib64/libvmGuestLib.so/libvmGuestLib.so
  libvmGuestLib.so (libc6) => /usr/lib/vmware-tools/lib32/libvmGuestLib.so/libvmGuestLib.so
  libuuid.so.1 (libc6) => /lib/i386-linux-gnu/libuuid.so.1
  libutil.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libutil.so.1
  libustr-1.0.so.1 (libc6) => /usr/lib/i386-linux-gnu/libustr-1.0.so.1
  libusb-1.0.so.0 (libc6) => /lib/i386-linux-gnu/libusb-1.0.so.0
  libusb-0.1.so.4 (libc6) => /lib/i386-linux-gnu/libusb-0.1.so.4
  libulockmgr.so.1 (libc6) => /lib/i386-linux-gnu/libulockmgr.so.1
  libudev.so.1 (libc6) => /lib/i386-linux-gnu/libudev.so.1
  libtinfo.so.5 (libc6) => /lib/i386-linux-gnu/libtinfo.so.5
  libtic.so.5 (libc6) => /usr/lib/i386-linux-gnu/libtic.so.5
  libthread_db.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libthread_db.so.1
  libtasn1.so.6 (libc6) => /usr/lib/i386-linux-gnu/libtasn1.so.6
  libsystemd-login.so.0 (libc6) => /lib/i386-linux-gnu/libsystemd-login.so.0
  libsystemd-daemon.so.0 (libc6) => /lib/i386-linux-gnu/libsystemd-daemon.so.0
  libstdc++.so.6 (libc6) => /usr/lib/i386-linux-gnu/libstdc++.so.6
  libssl.so.1.0.0 (libc6) => /lib/i386-linux-gnu/libssl.so.1.0.0
  libss.so.2 (libc6) => /lib/i386-linux-gnu/libss.so.2
  libsqlite3.so.0 (libc6) => /usr/lib/i386-linux-gnu/libsqlite3.so.0
  libslang.so.2 (libc6) => /lib/i386-linux-gnu/libslang.so.2
  libsigsegv.so.2 (libc6) => /usr/lib/i386-linux-gnu/libsigsegv.so.2
  libsigc-2.0.so.0 (libc6) => /usr/lib/i386-linux-gnu/libsigc-2.0.so.0
  libsepol.so.1 (libc6) => /lib/i386-linux-gnu/libsepol.so.1
  libsemanage.so.1 (libc6) => /usr/lib/i386-linux-gnu/libsemanage.so.1
  libselinux.so.1 (libc6) => /lib/i386-linux-gnu/libselinux.so.1
  libsas12.so.2 (libc6) => /usr/lib/i386-linux-gnu/libsas12.so.2
  librtmp.so.0 (libc6) => /usr/lib/i386-linux-gnu/librtmp.so.0
  librt.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/librt.so.1
  libroken.so.18 (libc6) => /usr/lib/i386-linux-gnu/libroken.so.18
  libresolv.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libresolv.so.2
  libreadline.so.6 (libc6) => /lib/i386-linux-gnu/libreadline.so.6
  libreadline.so.5 (libc6) => /lib/i386-linux-gnu/libreadline.so.5
  libp11-kit.so.0 (libc6) => /usr/lib/i386-linux-gnu/libp11-kit.so.0
  libpython2.7.so.1.0 (libc6) => /usr/lib/i386-linux-gnu/libpython2.7.so.1.0
  libpthread.so.0 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libpthread.so.0
  libprocps.so.3 (libc6) => /lib/i386-linux-gnu/libprocps.so.3
  libpopt.so.0 (libc6) => /lib/i386-linux-gnu/libpopt.so.0
  libpolkit-gobject-1.so.0 (libc6) => /usr/lib/i386-linux-gnu/libpolkit-gobject-1.so.0
  libpolkit-backend-1.so.0 (libc6) => /usr/lib/i386-linux-gnu/libpolkit-backend-1.so.0
  libpolkit-agent-1.so.0 (libc6) => /usr/lib/i386-linux-gnu/libpolkit-agent-1.so.0
  libpng12.so.0 (libc6) => /lib/i386-linux-gnu/libpng12.so.0
  libpng12.so.0 (libc6) => /usr/lib/i386-linux-gnu/libpng12.so.0

```

```

libply.so.2 (libc6) => /lib/i386-linux-gnu/libply.so.2
libply-splash-graphics.so.2 (libc6) => /lib/i386-linux-gnu/libply-splash-graphics.so.2
libply-splash-core.so.2 (libc6) => /lib/i386-linux-gnu/libply-splash-core.so.2
libply-boot-client.so.2 (libc6) => /lib/i386-linux-gnu/libply-boot-client.so.2
libpipeline.so.1 (libc6) => /usr/lib/i386-linux-gnu/libpipeline.so.1
libpcsclite.so.1 (libc6) => /lib/i386-linux-gnu/libpcsclite.so.1
libpcreposix.so.3 (libc6) => /usr/lib/i386-linux-gnu/libpcreposix.so.3
libpcre.so.3 (libc6) => /lib/i386-linux-gnu/libpcre.so.3
libpccprofile.so (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libpccprofile.so
libpci.so.3 (libc6) => /lib/i386-linux-gnu/libpci.so.3
libpcap.so.0.8 (libc6) => /usr/lib/i386-linux-gnu/libpcap.so.0.8
libparted.so.0 (libc6) => /lib/i386-linux-gnu/libparted.so.0
libpanelw.so.5 (libc6) => /usr/lib/i386-linux-gnu/libpanelw.so.5
libpanel.so.5 (libc6) => /usr/lib/i386-linux-gnu/libpanel.so.5
libpamc.so.0 (libc6) => /lib/i386-linux-gnu/libpamc.so.0
libpam_misc.so.0 (libc6) => /lib/i386-linux-gnu/libpam_misc.so.0
libpam.so.0 (libc6) => /lib/i386-linux-gnu/libpam.so.0
libnuma.so.1 (libc6) => /usr/lib/i386-linux-gnu/libnuma.so.1
libntfs-3g.so.841 (libc6) => /lib/i386-linux-gnu/libntfs-3g.so.841
libnss_nisplus.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-
gnu/libnss_nisplus.so.2
libnss_nis.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnss_nis.so.2
libnss_hesiod.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnss_hesiod.so.2
libnss_files.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnss_files.so.2
libnss_dns.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnss_dns.so.2
libnss_compat.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnss_compat.so.2
libnsl.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libnsl.so.1
libnl-3.so.200 (libc6) => /lib/i386-linux-gnu/libnl-3.so.200
libnl-genl-3.so.200 (libc6) => /lib/i386-linux-gnu/libnl-genl-3.so.200
libnloh.so.1 (libc6) => /lib/i386-linux-gnu/libnloh.so.1
libnloh-dbus.so.1 (libc6) => /lib/i386-linux-gnu/libnloh-dbus.so.1
libnfnetworklink.so.0 (libc6) => /usr/lib/i386-linux-gnu/libnfnetworklink.so.0
libnewt.so.0.52 (libc6) => /lib/i386-linux-gnu/libnewt.so.0.52
libncursesw.so.5 (libc6) => /lib/i386-linux-gnu/libncursesw.so.5
libncurses.so.5 (libc6) => /lib/i386-linux-gnu/libncurses.so.5
libnmpdec.so.2 (libc6) => /usr/lib/i386-linux-gnu/libnmpdec.so.2
libmount.so.1 (libc6) => /lib/i386-linux-gnu/libmount.so.1
libmenuw.so.5 (libc6) => /usr/lib/i386-linux-gnu/libmenuw.so.5
libmenu.so.5 (libc6) => /usr/lib/i386-linux-gnu/libmenu.so.5
libmemusage.so (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libmemusage.so
libmagic.so.1 (libc6) => /usr/lib/i386-linux-gnu/libmagic.so.1
libm.so.6 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libm.so.6
liblzma.so.5 (libc6) => /lib/i386-linux-gnu/liblzma.so.5
liblwres.so.90 (libc6) => /usr/lib/liblwres.so.90
liblockfile.so.1 (libc6) => /usr/lib/i386-linux-gnu/liblockfile.so.1
libldap_r-2.4.so.2 (libc6) => /usr/lib/i386-linux-gnu/libldap_r-2.4.so.2
liblber-2.4.so.2 (libc6) => /usr/lib/i386-linux-gnu/liblber-2.4.so.2
libk5crypto.so.3 (libc6) => /usr/lib/i386-linux-gnu/libk5crypto.so.3
libkrb5support.so.0 (libc6) => /usr/lib/i386-linux-gnu/libkrb5support.so.0
libkrb5.so.26 (libc6) => /usr/lib/i386-linux-gnu/libkrb5.so.26
libkrb5.so.3 (libc6) => /usr/lib/i386-linux-gnu/libkrb5.so.3
libkmod.so.2 (libc6) => /lib/i386-linux-gnu/libkmod.so.2
libkeyutils.so.1 (libc6) => /lib/i386-linux-gnu/libkeyutils.so.1
libjson-c.so.2 (libc6) => /lib/i386-linux-gnu/libjson-c.so.2
libiw.so.30 (libc6) => /lib/i386-linux-gnu/libiw.so.30
libiscfg.so.90 (libc6) => /usr/lib/libiscfg.so.90
libisccc.so.90 (libc6) => /usr/lib/libisccc.so.90
libisc.so.95 (libc6) => /usr/lib/libisc.so.95
libip6tc.so.0 (libc6) => /lib/libip6tc.so.0
libip4tc.so.0 (libc6) => /lib/libip4tc.so.0
libiptc.so.0 (libc6) => /lib/libiptc.so.0
libidn.so.11 (libc6) => /usr/lib/i386-linux-gnu/libidn.so.11
libhx509.so.5 (libc6) => /usr/lib/i386-linux-gnu/libhx509.so.5
libhistory.so.6 (libc6) => /lib/i386-linux-gnu/libhistory.so.6
libhistory.so.5 (libc6) => /lib/i386-linux-gnu/libhistory.so.5

```



```

libheimntlm.so.0 (libc6) => /usr/lib/i386-linux-gnu/libheimntlm.so.0
libheimbase.so.1 (libc6) => /usr/lib/i386-linux-gnu/libheimbase.so.1
libhcrypto.so.4 (libc6) => /usr/lib/i386-linux-gnu/libhcrypto.so.4
libgthread-2.0.so.0 (libc6) => /usr/lib/i386-linux-gnu/libgthread-2.0.so.0
libgssapi_krb5.so.2 (libc6) => /usr/lib/i386-linux-gnu/libgssapi_krb5.so.2
libgssapi.so.3 (libc6) => /usr/lib/i386-linux-gnu/libgssapi.so.3
libgpm.so.2 (libc6) => /usr/lib/i386-linux-gnu/libgpm.so.2
libgpg-error.so.0 (libc6) => /lib/i386-linux-gnu/libgpg-error.so.0
libgobject-2.0.so.0 (libc6) => /usr/lib/i386-linux-gnu/libgobject-2.0.so.0
libgnutls.so.26 (libc6) => /usr/lib/i386-linux-gnu/libgnutls.so.26
libgnutls-openssl.so.27 (libc6) => /usr/lib/i386-linux-gnu/libgnutls-openssl.so.27
libgnutls-extra.so.26 (libc6) => /usr/lib/i386-linux-gnu/libgnutls-extra.so.26
libgmodule-2.0.so.0 (libc6) => /usr/lib/i386-linux-gnu/libgmodule-2.0.so.0
libglib-2.0.so.0 (libc6) => /lib/i386-linux-gnu/libglib-2.0.so.0
libgirepository-1.0.so.1 (libc6) => /usr/lib/libgirepository-1.0.so.1
libgio-2.0.so.0 (libc6) => /usr/lib/i386-linux-gnu/libgio-2.0.so.0
libgdbm_compat.so.3 (libc6) => /usr/lib/i386-linux-gnu/libgdbm_compat.so.3
libgdbm.so.3 (libc6) => /usr/lib/i386-linux-gnu/libgdbm.so.3
libgcrypt.so.11 (libc6) => /lib/i386-linux-gnu/libgcrypt.so.11
libgcr-base-3.so.1 (libc6) => /usr/lib/i386-linux-gnu/libgcr-base-3.so.1
libgck-1.so.0 (libc6) => /usr/lib/i386-linux-gnu/libgck-1.so.0
libgccpp.so.1 (ELF) => /usr/lib/i386-linux-gnu/libgccpp.so.1
libgcc_s.so.1 (libc6) => /lib/i386-linux-gnu/libgcc_s.so.1
libgc.so.1 (libc6) => /usr/lib/i386-linux-gnu/libgc.so.1
libfuse.so.2 (libc6) => /lib/i386-linux-gnu/libfuse.so.2
libfribidi.so.0 (libc6) => /usr/lib/i386-linux-gnu/libfribidi.so.0
libfreetype.so.6 (libc6) => /usr/lib/i386-linux-gnu/libfreetype.so.6
libformw.so.5 (libc6) => /usr/lib/i386-linux-gnu/libformw.so.5
libform.so.5 (libc6) => /usr/lib/i386-linux-gnu/libform.so.5
libffi.so.6 (libc6) => /usr/lib/i386-linux-gnu/libffi.so.6
libe2p.so.2 (libc6) => /lib/i386-linux-gnu/libe2p.so.2
libext2fs.so.2 (libc6) => /lib/i386-linux-gnu/libext2fs.so.2
libexpatw.so.1 (libc6) => /usr/lib/i386-linux-gnu/libexpatw.so.1
libexpat.so.1 (libc6) => /lib/i386-linux-gnu/libexpat.so.1
libevent-2.0.so.5 (libc6) => /usr/lib/i386-linux-gnu/libevent-2.0.so.5
libestr.so.0 (libc6) => /usr/lib/i386-linux-gnu/libestr.so.0
libept.so.1.aptpkg4.12 (libc6) => /usr/lib/i386-linux-gnu/libept.so.1.aptpkg4.12
libelf.so.1 (libc6) => /usr/lib/i386-linux-gnu/libelf.so.1
libedit.so.2 (libc6) => /usr/lib/i386-linux-gnu/libedit.so.2
libdrm.so.2 (libc6) => /usr/lib/i386-linux-gnu/libdrm.so.2
libdns.so.100 (libc6) => /usr/lib/libdns.so.100
libdl.so.2 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libdl.so.2
libdiscover.so.2 (libc6) => /usr/lib/libdiscover.so.2
libdevmapper.so.1.02.1 (libc6) => /lib/i386-linux-gnu/libdevmapper.so.1.02.1
libdebconfclient.so.0 (libc6) => /usr/lib/i386-linux-gnu/libdebconfclient.so.0
libdbus-1.so.3 (libc6) => /lib/i386-linux-gnu/libdbus-1.so.3
libdbus-glib-1.so.2 (libc6) => /usr/lib/i386-linux-gnu/libdbus-glib-1.so.2
libdb-5.3.so (libc6) => /usr/lib/i386-linux-gnu/libdb-5.3.so
libcwidget.so.3 (libc6) => /usr/lib/libcwidget.so.3
libcurl.so.4 (libc6) => /usr/lib/i386-linux-gnu/libcurl.so.4
libcurl-gnutls.so.4 (libc6) => /usr/lib/i386-linux-gnu/libcurl-gnutls.so.4
libcrypto.so.1.0.0 (libc6) => /lib/i386-linux-gnu/libcrypto.so.1.0.0
libcrypt.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libcrypt.so.1
libcom_err.so.2 (libc6) => /lib/i386-linux-gnu/libcom_err.so.2
libck-connector.so.0 (libc6) => /usr/lib/i386-linux-gnu/libck-connector.so.0
libcidn.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libcidn.so.1
libcgmanager.so.0 (libc6) => /lib/i386-linux-gnu/libcgmanager.so.0
libcap.so.2 (libc6) => /lib/i386-linux-gnu/libcap.so.2
libcap-ng.so.0 (libc6) => /usr/lib/i386-linux-gnu/libcap-ng.so.0
libc.so.6 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libc.so.6
libbz2.so.1.0 (libc6) => /lib/i386-linux-gnu/libbz2.so.1.0
libbsd.so.0 (libc6) => /lib/i386-linux-gnu/libbsd.so.0
libboost_iostreams.so.1.54.0 (libc6) => /usr/lib/i386-linux-
gnu/libboost_iostreams.so.1.54.0
libblkid.so.1 (libc6) => /lib/i386-linux-gnu/libblkid.so.1

```

```

libbind9.so.90 (libc6) => /usr/lib/libbind9.so.90
libaudit.so.1 (libc6) => /lib/i386-linux-gnu/libaudit.so.1
libattr.so.1 (libc6) => /lib/i386-linux-gnu/libattr.so.1
libasprintf.so.0 (libc6) => /usr/lib/i386-linux-gnu/libasprintf.so.0
libasnl.so.8 (libc6) => /usr/lib/i386-linux-gnu/libasnl.so.8
libapt-private.so.0.0 (libc6) => /usr/lib/i386-linux-gnu/libapt-private.so.0.0
libapt-pkg.so.4.12 (libc6) => /usr/lib/i386-linux-gnu/libapt-pkg.so.4.12
libapt-inst.so.1.5 (libc6) => /usr/lib/i386-linux-gnu/libapt-inst.so.1.5
libapparmor.so.1 (libc6) => /usr/lib/i386-linux-gnu/libapparmor.so.1
libanl.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libanl.so.1
libacl.so.1 (libc6) => /lib/i386-linux-gnu/libacl.so.1
libaccountsservice.so.0 (libc6) => /usr/lib/i386-linux-gnu/libaccountsservice.so.0
libX11.so.6 (libc6) => /usr/lib/i386-linux-gnu/libX11.so.6
libXmu.so.1 (libc6) => /usr/lib/i386-linux-gnu/libXmu.so.1
libXext.so.6 (libc6) => /usr/lib/i386-linux-gnu/libXext.so.6
libXdmcp.so.6 (libc6) => /usr/lib/i386-linux-gnu/libXdmcp.so.6
libXau.so.6 (libc6) => /usr/lib/i386-linux-gnu/libXau.so.6
libSegFault.so (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-gnu/libSegFault.so
libGeoIP.so.1 (libc6) => /usr/lib/i386-linux-gnu/libGeoIP.so.1
libDeployPkg.so (libc6,x86-64) => /usr/lib/vmware-
tools/lib64/libDeployPkg.so/libDeployPkg.so
libDeployPkg.so (libc6) => /usr/lib/vmware-tools/lib32/libDeployPkg.so/libDeployPkg.so
libBrokenLocale.so.1 (libc6, OS ABI: Linux 2.6.24) => /lib/i386-linux-
gnu/libBrokenLocale.so.1
ld-linux.so.2 (ELF) => /lib/i386-linux-gnu/ld-linux.so.2
ld-linux.so.2 (ELF) => /lib/ld-linux.so.2
root@ubuntu32:~#

```

6. There were two dependencies that we could not find using “`/sbin/ldconfig -p`”. We need to determine why this occurred. We issue the command below to determine to where the file links.

```

administrator@ubuntu32:~$ ls -l /lib | grep ld-linux.so.2
lrwxrwxrwx 1 root root 25 Feb 25 08:58 ld-linux.so.2 -> i386-linux-gnu/ld-2.19.so
administrator@ubuntu32:~$

```

7. This was difficult to determine, but the new file links to a file with a different name. Below is the explanation for this, along with the links for two different Internet articles for more detailed information.

Why are they displayed differently? For `linux-gate.so.1` it is because it is not actually a file on-disk. It is exposed by the kernel as the mechanism for making system calls. For `/lib/ld-linux.so.2` it is because this is the program interpreter that is used for actually running the application.

<http://stackoverflow.com/questions/19981862/what-are-ld-linux-so-2-and-linux-gate-so-1>

<http://www.trilithium.com/johan/2005/08/linux-gate/>

Now we will work on the Ethernet NICs. When we created this virtual machine, we created five NIC's. One port will serve as the Linux management interface. This is not shared with the router. The other four ports will be one virtual 4-port card for our Cisco 5921 ESR. Linux will not manage these ports.

8. Issue the following commands to setup the network interfaces.

```

root@ubuntu32:/home/administrator# cd /etc/network

root@ubuntu32:/etc/network# ls
if-down.d if-post-down.d if-pre-up.d if-up.d interfaces interfaces.d run
root@ubuntu32:/etc/network# more interfaces

```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth4
iface eth4 inet dhcp
```

9. Next, we will make some adjustments as the current configuration uses dhcp for the Linux management port and we want to make the IP address static for better predictability. The “nano” interface is a newer and easier to use editor than “vi”, especially if you spend most of your time in the graphical world.

```
root@ubuntu32:/etc/network# nano interfaces
```

10. Below are the edits for our interface configuration file located at /etc/network/. The following would be typical in the Cisco dCloud environment or an actual production environment:

```
root@ubuntu32:/etc/network# more interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth4
iface eth4 inet static
    address 198.18.134.122
    gateway 198.18.128.1
    netmask 255.255.192.0
    network 198.18.128.0
    broadcast 198.18.191.255
    dns-nameservers 8.8.8.8 8.8.4.4
```

```
# These interfaces will be used by the Cisco 5921 ESR.
```

```
auto eth0
iface eth0 inet static
pre-up /sbin/ethtool -s eth0 speed 1000 duplex full
```

```
auto eth1
iface eth1 inet static
pre-up /sbin/ethtool -s eth1 speed 1000 duplex full
```

```
auto eth2
iface eth2 inet static
pre-up /sbin/ethtool -s eth2 speed 1000 duplex full
```

```
auto eth3
iface eth3 inet static
pre-up /sbin/ethtool -s eth3 speed 1000 duplex full
```

11. Next, we will configure the fully qualified domain name in the hosts file.

```
root@ubuntu32:/etc/network# cat /etc/hosts
```

```
127.0.0.1 localhost
127.0.1.1 ubuntu32.dcloud.cisco.com          ubuntu32
198.18.134.122 ubuntu32.dcloud.cisco.com     ubuntu32
```

```
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@ubuntu32:/etc/network#
```

```
root@ubuntu32:/etc/network# reboot
```

```
Broadcast message from administrator@ubuntu32
(/dev/pts/0) at 22:34 ...
```

```
The system is going down for reboot NOW!
Connection to 198.18.134.120 closed by remote host.
Connection to 198.18.134.120 closed.
```

More information can be found at the Configure Network Interface Using Command-Line help page:

<https://help.ubuntu.com/community/NetworkConfigurationCommandLine>

- If you are building a virtual machine for a VMware environment, you will want to install VMware tools for the additional manageability that it provides. From within VMware, you must invoke the procedure to install VMware tools, which will mount a virtual CD. This step is already done for you in Cisco dCloud and shown below for reference:

```
root@ubuntu32:/home/administrator# mkdir /mnt/cdrom
root@ubuntu32:/home/administrator# mount /dev/cdrom /mnt/cdrom
mount: block device /dev/sr0 is write-protected, mounting read-only
root@ubuntu32:/home/administrator# ls /mnt/cdrom
manifest.txt  VMwareTools-9.0.0-782409.tar.gz
root@ubuntu32:/home/administrator# tar xzvf /mnt/cdrom/VMwareTools-9.0.0-782409.tar.gz -C /tmp/
vmware-tools-distrib/
vmware-tools-distrib/bin/
vmware-tools-distrib/bin/vm-support
vmware-tools-distrib/bin/vmware-config-tools.pl
vmware-tools-distrib/bin/vmware-uninstall-tools.pl
vmware-tools-distrib/installer/
```

----- Output abridged. -----

```
vmware-tools-distrib/etc/tpvmlp.conf
vmware-tools-distrib/etc/xsession-xdm.pl
vmware-tools-distrib/etc/resume-vm-default
vmware-tools-distrib/etc/vmware-user.desktop
vmware-tools-distrib/etc/suspend-vm-default
vmware-tools-distrib/etc/poweron-vm-default
vmware-tools-distrib/etc/scripts/
vmware-tools-distrib/etc/scripts/vmware/
vmware-tools-distrib/etc/scripts/vmware/network
vmware-tools-distrib/etc/vmware-tools/
vmware-tools-distrib/etc/xsession-gdm.sh
vmware-tools-distrib/etc/not_configured
vmware-tools-distrib/etc/poweroff-vm-default
root@ubuntu32:/home/administrator# cd /tmp/vmware-tools-distrib
root@ubuntu32:/tmp/vmware-tools-distrib# ./vmware-install.pl -d
Creating a new VMware Tools installer database using the tar4 format.
```

Installing VMware Tools.

In which directory do you want to install the binary files?
[/usr/bin]

What is the directory that contains the init directories (rc0.d/ to rc6.d/)?
[/etc]

What is the directory that contains the init scripts?
[/etc/init.d]

In which directory do you want to install the daemon files?
[/usr/sbin]

In which directory do you want to install the library files?
[/usr/lib/vmware-tools]

The path "/usr/lib/vmware-tools" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?
[yes]

In which directory do you want to install the documentation files?
[/usr/share/doc/vmware-tools]

The path "/usr/share/doc/vmware-tools" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want? [yes]

The installation of VMware Tools 9.0.0 build-782409 for Linux completed successfully. You can decide to remove this software from your system at any time by invoking the following command: "/usr/bin/vmware-uninstall-tools.pl".

Before running VMware Tools for the first time, you need to configure it by invoking the following command: "/usr/bin/vmware-config-tools.pl". Do you want this program to invoke the command for you now? [yes]

Initializing...

Making sure services for VMware Tools are stopped.

The VMware FileSystem Sync Driver (vmsync) allows external third-party backup software that is integrated with vSphere to create backups of the virtual machine. Do you wish to enable this feature? [no]

The module vmci has already been installed on this system by another installer or package and will not be modified by this installer. Use the flag --clobber-kernel-modules=vmci to override.

The module vsock has already been installed on this system by another installer or package and will not be modified by this installer. Use the flag --clobber-kernel-modules=vsock to override.

The module vmxnet3 has already been installed on this system by another installer or package and will not be modified by this installer. Use the flag --clobber-kernel-modules=vmxnet3 to override.

The module pvscsi has already been installed on this system by another installer or package and will not be modified by this installer. Use the flag --clobber-kernel-modules=pvscsi to override.

The module vmmemctl has already been installed on this system by another installer or package and will not be modified by this installer. Use the flag --clobber-kernel-modules=vmmemctl to override.

The VMware Host-Guest Filesystem allows for shared folders between the host OS and the guest OS in a Fusion or Workstation virtual environment. Do you wish to enable this feature? [no]

The vmxnet driver is no longer supported on kernels 3.3 and greater. Please upgrade to a newer virtual NIC. (e.g., vmxnet3 or e1000e)

The vmblock enables dragging or copying files between host and guest in a Fusion or Workstation virtual environment. Do you wish to enable this feature?
[no]

!!! [EXPERIMENTAL] !!!

VMware automatic kernel modules enables automatic building and installation of VMware kernel modules at boot that are not already present. By selecting yes, you will be enabling this experimental feature. You can always disable this feature by re-running vmware-config-tools.pl.

Would you like to enable VMware automatic kernel modules?

[no]

No X install found.

Creating a new initrd boot image for the kernel.

update-initramfs: Generating /boot/initrd.img-3.13.0-57-generic

vmware-tools start/running

The configuration of VMware Tools 9.0.0 build-782409 for Linux for this running kernel completed successfully.

You must restart your X session before any mouse or graphics changes take effect.

You can now run VMware Tools by invoking "/usr/bin/vmware-toolbox-cmd" from the command line.

To enable advanced X features (e.g., guest resolution fit, drag and drop, and file and text copy/paste), you will need to do one (or more) of the following:

1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session; and,
3. Restart your X session.

Enjoy,

--the VMware team

Found VMware Tools CDROM mounted at /mnt/cdrom. Ejecting device /dev/sr0 ...

root@ubuntu32:/tmp/vmware-tools-distrib# **reboot**

root@ubuntu32:/tmp/vmware-tools-distrib#

Broadcast message from administrator@ubuntu32

(/dev/pts/0) at 19:44 ...

The system is going down for reboot NOW!

13. Next, we will log back into the system to check the changes.

[gransysc02:~] fcolumbu% **ssh -l administrator 198.18.134.122**

administrator@198.18.134.120's password:

Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-57-generic i686)

* Documentation: <https://help.ubuntu.com/>

System information as of Tue Jul 21 22:34:59 PDT 2015

System load: 0.0	Memory usage: 0%	Processes: 68
Usage of /: 19.5% of 5.78GB	Swap usage: 0%	Users logged in: 0

Graph this data and manage this system at:

<https://landscape.canonical.com/>

```

Last login: Tue Jul 21 22:12:24 2015 from gransysc02.ct.ri.cox.cci
administrator@ubuntu32:~$ sudo su
[sudo] password for administrator:
root@ubuntu32:/home/administrator# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8f:f5:06
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:30 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:19 Base address:0x2000

eth1      Link encap:Ethernet  HWaddr 00:0c:29:8f:f5:10
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:30 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16 Base address:0x2080

eth2      Link encap:Ethernet  HWaddr 00:0c:29:8f:f5:1a
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:30 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:17 Base address:0x2400

eth3      Link encap:Ethernet  HWaddr 00:0c:29:8f:f5:24
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:30 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:18 Base address:0x2480

eth4      Link encap:Ethernet  HWaddr 00:0c:29:8f:f5:2e
          inet addr:198.18.134.122  Bcast:198.18.191.255  Mask:255.255.192.0
          inet6 addr: fe80::20c:29ff:fe8f:f52e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:370 errors:0 dropped:0 overruns:0 frame:0
          TX packets:119 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33901 (33.9 KB)  TX bytes:14731 (14.7 KB)
          Interrupt:19 Base address:0x2800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@ubuntu32:/home/administrator#

root@ubuntu32:/etc/network# uname -a
Linux ubuntu32 3.13.0-57-generic #95-Ubuntu SMP Fri Jun 19 09:27:48 UTC 2015 i686 i686 i686
GNU/Linux
root@ubuntu32:/etc/network# ping www.google.com
PING www.google.com (74.125.224.83) 56(84) bytes of data.
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=1 ttl=56 time=12.6 ms
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=2 ttl=56 time=12.9 ms

```

```
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=3 ttl=56 time=11.7 ms
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=4 ttl=56 time=12.6 ms
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=5 ttl=56 time=11.1 ms
64 bytes from lax17s02-in-f19.1e100.net (74.125.224.83): icmp_seq=6 ttl=56 time=12.4 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 11.159/12.259/12.935/0.606 ms
root@ubuntu32:/etc/network#
```

This concludes this Linux Prequalification activity.

Appendix A: Key configuration files for Ubuntu32w-1 / ESR-1 before configuring the tap0 interface

The examples below show the configuration files for the Ubuntu server before configuring the top0 interface. The SWROPTIONS file is located in /opt/cisco/c5921.

The IP address for the Linux management port eth4 is **198.18.134.120**.

The original sample was edited and used as the basis for what follows:

```
##### EXAMPLE SWROPTIONS FILE #####

soft-rommon=./
launchapp=c5921i86-universalk9-ms.SPA
ram=512
ethernet-slots=1
node-lock-type=machine

### FILE MAPPING SECTION ###

[filemap]
ios=flash0
linux=/opt/cisco/c5921/
access=rw

### INTERFACE MAPPING SECTION ###

# Map Linux eth0 to IOS e0/0, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/0 in IOS.
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth0 to IOS e0/0.
# Push changes like speed/duplex/MAC of IOS e0/0 to Linux eth0.

[interface]
linux=eth0
ios=e0/0
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.

[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# Map Linux eth2 to IOS e0/2, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/2 in IOS
# Monitor/Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth2 to IOS e0/2.
# Push IOS e0/2 changes to Linux eth2.
```

```
[interface]
linux=eth2
ios=e0/2
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
[interface]
linux=eth3
ios=e0/3
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# [interface]
# linux=tap0
# ios=e1/0
# type=tap
# monitor-state=true
```

Configuration that would be pre-loaded on the router before implementing the tap0 Linux interface:

```
c5921_ubuntu32_1#sh run
Building configuration...

Current configuration : 4007 bytes
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service call-home
!
hostname c5921_ubuntu32_1
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$x.6O$4UZX1B51YLF1DeOul7GiV1
enable password cisco
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
call-home
! if contact email address in call-home is configured as sch-smart-licensing@cisco.com
! The email address configured in Cisco Smart License Portal will be used as contact email
```

```

address to send SCH notifications
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
!
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
redundancy
!
!
interface Loopback0
  ip address 10.0.0.120 255.255.255.255
!
interface Ethernet0/0
  ip address 172.23.1.120 255.255.255.0
  duplex full
  speed 1000
  no mop enabled
!
interface Ethernet0/1

```

```
ip address 172.23.2.1 255.255.255.0
duplex full
speed 1000
!
interface Ethernet0/2
ip address 172.23.3.1 255.255.255.0
duplex full
speed 1000
!
interface Ethernet0/3
ip address 172.23.4.1 255.255.255.0
duplex full
speed 1000
!
router ospf 100
network 10.0.0.120 0.0.0.0 area 0
network 172.23.1.0 0.0.0.255 area 0
network 172.23.2.0 0.0.0.255 area 0
network 172.23.3.0 0.0.0.255 area 0
network 172.23.4.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
snmp-server community public RO
!
!
control-plane
!
!
line con 0
logging synchronous
line aux 0
line vty 0 4
password cisco
login
transport input all
!
ntp server pool.ntp.org
!
end

c5921_ubuntu32_1#
```

← Note the last interface.

Appendix B: Differences on ESR-1 to implement the tap0 interface such that the Cisco 5921 uses its Linux host as an NTP proxy

This section highlights the changes you would make to implement the tap0 interface in the router configuration file. The differences between the default file and the edited file are highlighted.

```
c5921_ubuntu32_1#sh running-config brief ← The "brief" option reduces installed certificates to a single line.  
Building configuration...
```

```
Current configuration : 2509 bytes  
!  
! No configuration change since last restart  
!  
version 15.5  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
service call-home  
!  
hostname c5921_ubuntu32_1  
!  
boot-start-marker  
boot-end-marker  
!  
!  
enable secret 5 $1$x.6O$4UZx1B5lYLF1DeOul7GiVl  
enable password cisco  
!  
no aaa new-model  
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2  
clock timezone PST -8 0  
mmi polling-interval 60  
no mmi auto-configure  
no mmi pvc  
mmi snmp-timeout 180  
call-home  
! if contact email address in call-home is configured as sch-smart-licensing@cisco.com  
! The email address configured in Cisco Smart License Portal will be used as contact email  
address to send SCH notifications  
contact-email-addr sch-smart-licensing@cisco.com  
profile "CiscoTAC-1"  
active  
destination transport-method http  
no destination transport-method email  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
ip name-server 8.8.8.8  
ip cef  
no ipv6 cef
```

```

!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
  certificate ca 01
license udi pid CISCO5921-K9 sn 91J81443B9I
license platform throughput level c5921-x86-level3
!
redundancy
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback0
  ip address 10.0.0.120 255.255.255.255
!
interface Ethernet0/0
  ip address 172.23.1.120 255.255.255.0
  duplex full
  speed 1000
  no mop enabled
!
interface Ethernet0/1
  ip address 172.23.2.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet0/2
  ip address 172.23.3.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet0/3
  ip address 172.23.4.1 255.255.255.0
  duplex full
  speed 1000
!
interface Ethernet1/0
  description TAP interface      <- To the router, the tap0 interface behaves like a simple Ethernet port that
connects to the Linux host.

```

```

ip address 198.18.134.131 255.255.192.0
!
interface Ethernet1/1
  no ip address
  shutdown      ← Ethernet 1/1 – 1/3 remain shutdown as they map to no Linux ports and are therefore not used.
!
interface Ethernet1/2
  no ip address
  shutdown
!
interface Ethernet1/3
  no ip address
  shutdown
!
router ospf 100
  network 10.0.0.120 0.0.0.0 area 0
  network 172.23.1.0 0.0.0.255 area 0
  network 172.23.2.0 0.0.0.255 area 0
  network 172.23.3.0 0.0.0.255 area 0
  network 172.23.4.0 0.0.0.255 area 0
  network 198.18.128.0 0.0.63.255 area 0
  default-information originate
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 198.18.134.120
!
!
snmp-server community public RO
!
!
control-plane
!
!
line con 0
  logging synchronous
line aux 0
line vty 0 4
  password cisco
  login
  transport input all
!
ntp server 198.18.134.132      ← IP address of “tap0” on the Linux host
!
end

```

```
c5921_ubuntu32_1#
```

```
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS
```

```
##### EXAMPLE SWROPTIONS FILE #####
```

```

soft-rommon=./
launchapp=c5921i86-universalk9-ms.SPA
ram=512
ethernet-slots=2      ← 4 ports per slot – adding the tap0 interface requires an additional slot.
node-lock-type=machine

```

FILE MAPPING SECTION

```
[filemap]
ios=flash0
linux=/opt/cisco/c5921/
access=rw
```

INTERFACE MAPPING SECTION

```
# Map Linux eth0 to IOS e0/0, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/0 in IOS.
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth0 to IOS e0/0.
# Push changes like speed/duplex/MAC of IOS e0/0 to Linux eth0.
```

```
[interface]
linux=eth0
ios=e0/0
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.
```

```
[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# Map Linux eth2 to IOS e0/2, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/2 in IOS
# Monitor/Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth2 to IOS e0/2.
# Push IOS e0/2 changes to Linux eth2.
```

```
[interface]
linux=eth2
ios=e0/2
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
[interface]
linux=eth3
ios=e0/3
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```



```
[interface]
linux=tap0           <- tap0 is now active (not commented out) and mapped to e 1/0 of the router.
ios=e1/0
type=tap
monitor-state=true

root@ubuntu32:/opt/cisco/c5921#
```

Configuration of the Linux IP ports - /etc/network/interfaces

```
root@ubuntu32:/etc/network# cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth4
iface eth4 inet static
    address 198.18.134.120           <- Management IP for the Linux VM
    gateway 198.18.128.1
    netmask 255.255.192.0
    network 198.18.128.0
    broadcast 198.18.191.255
    dns-nameservers 8.8.8.8 8.8.4.4

# These interfaces will be used by the Cisco 5921 ESR.

auto eth0
iface eth0 inet static
pre-up /sbin/ethtool -s eth0 speed 1000 duplex full

auto eth1
iface eth1 inet static
pre-up /sbin/ethtool -s eth1 speed 1000 duplex full

auto eth2
iface eth2 inet static
pre-up /sbin/ethtool -s eth2 speed 1000 duplex full

auto eth3
iface eth3 inet static
pre-up /sbin/ethtool -s eth3 speed 1000 duplex full

auto tap0
iface tap0 inet static
    pre-up tuncctl -u root -t tap0
    address 198.18.134.132           <- tap0 IP – Virtual network to the c5921
    gateway 198.18.128.1
    netmask 255.255.192.0
    network 198.18.128.0
    broadcast 198.18.191.255
    dns-nameservers 8.8.8.8 8.8.4.4

root@ubuntu32:/etc/network#
```

Appendix C: Key configuration files for Ubuntu32w-2 / ESR-2

This section highlights the key configuration files for the Ubuntu server Ubuntu32s-2 / ESR-2.

The IP address for the Linux management port eth4 is 198.18.134.121.

Enter the following commands to see the configuration files.

```
root@ubuntu32:/opt/cisco/c5921# cat SWROPTIONS

##### EXAMPLE SWROPTIONS FILE #####

soft-rommon=./
launchapp=c5921i86-universalk9-ms.SPA
ram=512
ethernet-slots=1
node-lock-type=machine

### FILE MAPPING SECTION ###

[filemap]
ios=flash0
linux=/opt/cisco/c5921/
access=rw

### INTERFACE MAPPING SECTION ###

# Map Linux eth0 to IOS e0/0, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/0 in IOS.
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth0 to IOS e0/0.
# Push changes like speed/duplex/MAC of IOS e0/0 to Linux eth0.

[interface]
linux=eth0
ios=e0/0
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.

[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# Map Linux eth2 to IOS e0/2, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/2 in IOS
# Monitor/Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth2 to IOS e0/2.
# Push IOS e0/2 changes to Linux eth2.
```

```
[interface]
linux=eth2
ios=e0/2
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
[interface]
linux=eth3
ios=e0/3
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```
# [interface]
# linux=tap0
# ios=e1/0
# type=tap
# monitor-state=true
```

```
root@ubuntu32:/opt/cisco/c5921#
```

Configuration pre-loaded on the router:

```
c5921_ubuntu32_2#sh run
Building configuration...
```

```
Current configuration : 4007 bytes
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service call-home
!
hostname c5921_ubuntu32_2
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$x.6O$4UZX1B5lYLF1DeOul7GiV1
enable password cisco
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
call-home
! if contact email address in call-home is configured as sch-smart-licensing@cisco.com
```

```
! The email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications
```

```
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
!
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
  certificate ca 01
    30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
    32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
    6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
    3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
    43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
    526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
    82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
    CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
    1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
    4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
    7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
    68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
    C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
    C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
    DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
    06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
    4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
    03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
    604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
    D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
    467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
    7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
    5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
    80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
    418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
    D697DF7F 28
  quit
license udi pid CISCO5921-K9 sn 9509000IL4J
!
redundancy
!
!
interface Loopback0
  ip address 10.0.0.121 255.255.255.255
!
interface Ethernet0/0
  ip address 172.23.1.121 255.255.255.0
  duplex full
  speed 1000
  no mop enabled
!
interface Ethernet0/1
```

```
ip address 172.23.2.2 255.255.255.0
duplex full
speed 1000
!
interface Ethernet0/2
ip address 172.23.3.2 255.255.255.0
duplex full
speed 1000
!
interface Ethernet0/3
ip address 172.23.4.2 255.255.255.0
duplex full
speed 1000
!
router ospf 100
network 10.0.0.121 0.0.0.0 area 0
network 172.23.1.0 0.0.0.255 area 0
network 172.23.2.0 0.0.0.255 area 0
network 172.23.3.0 0.0.0.255 area 0
network 172.23.4.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
snmp-server community public RO
!
!
control-plane
!
!
line con 0
logging synchronous
line aux 0
line vty 0 4
password cisco
login
transport input all
!
ntp server pool.ntp.org
!
end

c5921_ubuntu32_2#
```

Appendix D: Key configuration files for Ubuntu32

This section highlights the key configuration files for the Ubuntu server Ubuntu32.

The IP address for the Linux management port eth4 is 198.18.134.122.

Enter the following commands to see the configuration files.

```
root@ubuntu32:/opt/cisco/c5921# cat SWOPTIONS

##### EXAMPLE SWOPTIONS FILE #####

soft-rommon=./
launchapp=c5921i86-universalk9-ms.SPA
ram=512
ethernet-slots=1
node-lock-type=machine

### FILE MAPPING SECTION ###

[filemap]
ios=flash0
linux=/opt/cisco/c5921/
access=rw

### INTERFACE MAPPING SECTION ###

# Map Linux eth0 to IOS e0/0, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/0 in IOS.
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth0 to IOS e0/0.
# Push changes like speed/duplex/MAC of IOS e0/0 to Linux eth0.

[interface]
linux=eth0
ios=e0/0
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

# Map Linux eth1 to IOS e0/1, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/1 in IOS
# Monitor and Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth1 to IOS e0/1.
# Push MAC changes of IOS e0/1 to Linux eth1.
# Speed/Duplex of e0/1 could be changed by IOS and hence
# pushed to Linux eth1.

[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

```

# Map Linux eth2 to IOS e0/2, type raw
# Set promiscuous true
# Make speed/duplex interface configs available for e0/2 in IOS
# Monitor/Pull Linux interface changes like speed/duplex/MTU/MAC
# of eth2 to IOS e0/2.
# Push IOS e0/2 changes to Linux eth2.

[interface]
linux=eth2
ios=e0/2
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

[interface]
linux=eth3
ios=e0/3
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true

# [interface]
# linux=tap0
# ios=e1/0
# type=tap
# monitor-state=true

root@ubuntu32:/opt/cisco/c5921#

```

Suggested initial configuration for the router:

```

c5921_ubuntu32_3#sh run
Building configuration...

Current configuration : 4023 bytes
!
! Last configuration change at 16:03:10 PST Mon Aug 31 2015
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service call-home
!
hostname c5921_ubuntu32_3
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$kdqQ$V2L6R4mkyPv9AKcuzqEa..
enable password cisco
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure

```

```

no mmi pvc
mmi snmp-timeout 180
call-home
! if contact email address in call-home is configured as sch-smart-licensing@cisco.com
! The email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
active
destination transport-method http
no destination transport-method email
!
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12
revocation-check crl
!
!
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9B8 E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
license udi pid CISCO5921-K9 sn 9G7598216GZ
!
redundancy
!
!
interface Loopback0
ip address 10.0.0.122 255.255.255.255
!
interface Ethernet0/0
ip address 172.23.1.123 255.255.255.0
duplex full
speed 1000

```



```

no mop enabled
!
interface Ethernet0/1
 ip address 172.23.2.3 255.255.255.0
 duplex full
 speed 1000
!
interface Ethernet0/2
 ip address 172.23.3.3 255.255.255.0
 duplex auto
 speed 1000
!
interface Ethernet0/3
 ip address 172.23.4.3 255.255.255.0
 duplex full
 speed 1000
!
router ospf 100
 network 10.0.0.122 0.0.0.0 area 0
 network 172.23.1.0 0.0.0.255 area 0
 network 172.23.2.0 0.0.0.255 area 0
 network 172.23.3.0 0.0.0.255 area 0
 network 172.23.4.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
snmp-server community public RO
!
!
control-plane
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
 transport input none
!
!
end

c5921_ubuntu32_3#

```

End of Lab Guide



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)