

# **CCNA**

## **ICND-1 Command Reference**

**Somenath Sinha**

Nov 2018

# Contents

<b>I</b>	<b>ICND1 - Switches</b>	<b>2</b>
<b>1</b>	<b>Basic Cisco Catalyst Switch Configuration</b>	<b>3</b>
1.1	Port Addressing . . . . .	3
1.2	Configuring a Management IP Address . . . . .	3
1.2.1	View how we're connected to a Switch . . . . .	3
1.2.2	Privileged mode, auto-completion and showing possible commands .	4
1.2.3	Configuring the Terminal . . . . .	4
1.2.4	Interface Configuration Mode . . . . .	5
1.2.5	Adding a Management IP . . . . .	5
1.3	Configuring a Default Gateway . . . . .	5
1.4	Setting Console and VTY Passwords . . . . .	5
1.4.1	Configuring a password for the VTY lines . . . . .	6
1.5	Show Running Configuration . . . . .	6
1.6	Checking for connectivity using Ping . . . . .	7

## **Part I**

# **ICND1 - Switches**

# Chapter 1

## Basic Cisco Catalyst Switch Configuration

### 1.1 Port Addressing

The addressing of ports on a Cisco Catalyst Switch follows the format:

```
interface-type Stack #/Card #/Port #
```

Let us consider the **Catalyst 3750** switch. This switch isn't modular and thus can't have multiple cards, but it supports Stack-wise, which is a Cisco technology to combine multiple physical switches in a stack into a single logical switch. Thus, the middle number when referring to a particular interface (i.e., port on a physical switch) will **always be 0**. Thus, the Giga-bit port that's 16<sup>th</sup> port on the 3<sup>rd</sup> stack member of switches will be written as: GigabitEthernet 3/0/16, or **g3/0/16** for short.

For a **Cisco 2960** switch, there's no option for Stack-wise, and hence we don't need the first number. Then, the name for the interface becomes GigabitEthernet 0/16 or **g0/16**.

### 1.2 Configuring a Management IP Address

We can connect to the switch physically via the console port, or through one of the interfaces using Ethernet. In case of the later, we have to set up a management IP address through which we can use telnet or SSH to control and configure the switch.

#### 1.2.1 View how we're connected to a Switch

To view how we're presently connected to a switch, we can use the `show line` command:

```
1  sw1>show line
2      Tty Typ      Tx/Rx      A Modem  Roty Acc0 AccI   Uses   Noise  Overruns  Int
3  *    0 CTY          - -          - -      - -    0      0      0/0      -
4      1 AUX    9600/9600    - -          - -      - -    0      0      0/0      -
5      2 VTY          - -          - -      - -    1      0      0/0      -
6      3 VTY          - -          - -      - -    0      0      0/0      -
7      4 VTY          - -          - -      - -    0      0      0/0      -
```

8	5 VTY	-	-	-	-	-	0	0	0/0	-
9	6 VTY	-	-	-	-	-	0	0	0/0	-

The **asterisk (\*)** at the beginning of the line shows that it's the port currently being used. The **CTY** refers to a console port, which is the port we're using. If we were using telnet or SSH, i.e., log in remotely to the switch/router, then one of the **VTY** (*Virtual Terminal*) would've been used.

## 1.2.2 Privileged mode, auto-completion and showing possible commands

The > sign at the end of the prompt signals that we're currently in *user mode*. To change the configuration on the equipment we need to be in *Privileged mode*, which is indicated by a # prompt. To enter the privileged mode, we use the command `enable`. To leave the privileged mode, we use `disable`:

```
1 sw1>enable
2 sw1#disable
3 sw1>
```

To go into privileged mode, we need not type the entire command. Cisco **IOS** (*Internetwork Operating System*) can show all possible commands that can follow the current string on the present line on pressing the ? key. For example:

```
1 sw1>e?
2 enable ethernet exit
3
4 sw1>en?
5 enable
```

This also means that we don't need to type out the entire command. We can simply use short-cuts for the commands, which are the minimum number of letters in the word of a command that is unique. Thus, in the case of the `enable`, the minimum characters required are 2 since there are 3 commands starting with 'e' but only 1 starting with 'en', i.e., `enable`.

```
1 sw1>en
2 sw1#
```

## 1.2.3 Configuring the Terminal

While we may be in privileged mode, we still need to tell the switch that we want to configure from the terminal. For this, we use the command `configure terminal`, or **conf t** for short. This would bring us into the **global config** mode:

```
1 sw1#conf t
2 Enter configuration commands, one per line. End with CNTL/Z.
3 sw1(config)#
```

To exit the global configuration mode, we use the `exit` command:

```
1 sw1(config)#exit
2 sw1#
3 *Nov 11 21:36:07.353: %SYS-5-CONFIG_I: Configured from console by console
4 sw1#
```

## 1.2.4 Interface Configuration Mode

To set the management IP address we need to go to the interface configuration port. On a Cisco Switch, unlike on a PC/Laptop, every interface gets associated to the IP address which is of the network it's connected to. Layer-2 Switches are generally blind to the IP addresses since they don't make forwarding decisions on the basis of IP addresses, but this IP Address is required to be associated with the switch to uniquely identify the switch on the network for remote access through telnet/SSH.

By default, on the switch, all the ports belong to a specific **VLAN(Virtual LAN)**. Thus, we can set the IP address of the *default VLAN*, **vlan1**, to set up the management IP address of the switch. To enter the interface configuration for vlan1, we use: `interface vlan1`, or simply **int vlan1** :

---

```
1 sw1(config)#int vlan 1
2 sw1(config-if)#
```

---

The prompt changed to `(config-if)#` to show that we've now entered the *Interface config mode*.

## 1.2.5 Adding a Management IP

Just like with a PC, we need to set up both the IP address as well as the Subnet Mask. Let us consider the switch needs to have the IP **192.168.1.11/24**. We can't enter the value in suffix notation. The equivalent subnet mask is `255.255.255.0`. Thus, the command becomes:

---

```
1 sw1(config-if)#ip add 192.168.1.11 255.255.255.0
2 sw1(config-if)#no shutdown
3 sw1(config-if)#
4 *Nov 11 21:51:57.295: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
5 *Nov 11 21:51:58.298: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
↔ state to up
6 sw1(config-if)#
```

---

Typically the interface `vlan1` (i.e, the 1st virtual LAN) is turned on by default, but sometimes it may have been administratively shut-down, which we can fix using `no shutdown` command, or for short **no shut**.

## 1.3 Configuring a Default Gateway

A default gateway is required for the switch to determine which path to take to exit it's own subnet and reach another network. While the IP address has to be set for an interface, the default gateway has to be set in global config mode:

---

```
1 sw1(config)#ip default-gateway 192.168.1.1
```

---

## 1.4 Setting Console and VTY Passwords

To set the password for the console port login, we need to enter its configuration, i.e., console line configuration, and then set the password. Let's say we want to set it to *cisco*. We can do this by:

---

```
1 sw1(config-line)#password cisco
2 sw1(config-line)#login
```

---

The `login` command enables the password check at login. Unless it's set, even if there's a valid password, the system won't ask for credentials while logging in. Further, to disable password check, we don't have to change/remove the password. we simply need to do `no login`. The login command doesn't work unless there's a password set:

---

```
1 sw1(config-line)#login
2 % Login disabled on line 0, until 'password' is set
```

---

### 1.4.1 Configuring a password for the VTY lines

We could set the password for each VTY line individually, but if we're setting them (or a subset of them) all to an identical config, we can use:

---

```
1 sw1(config)#line vty 0 5
2 sw1(config-line)#password cisco
3 sw1(config-line)#login
4 sw1(config-line)#end
5 sw1#
```

---

This is assuming we want to set the password for lines 0-5 and not 6 of the available VTY lines. Also notice that we jump levels till we end the configuration of the switch by using `end` command. Unlike the `exit` command, the `end` command doesn't just jump up to the parent config level, but ends the configuration all together and takes us out of the configuration mode all together.

## 1.5 Show Running Configuration

To see the current configuration we created (but not yet saved), i.e., the configuration that the device is currently using, we use the command:

---

```
1 sw1#sh run
2 Building configuration...
3
4 Current configuration : 2911 bytes
5 !
6 ! Last configuration change at 22:48:16 UTC Sun Nov 11 2018
7 !
8 version 15.2
9 service timestamps debug datetime msec
10 service timestamps log datetime msec
11 no service password-encryption
12 service compress-config
13 !
14 hostname sw1
15 ...
16 !
17 interface Vlan1
18 ip address 192.168.1.11 255.255.255.0
```

```
19  !
20  ip default-gateway 192.168.1.1
21  ip forward-protocol nd
22  !
23  ...
24  !
25  line con 0
26    exec-timeout 0 0
27    password cisco
28    login
29  line aux 0
30  line vty 0 4
31    password cisco
32    login
33  line vty 5
34    password cisco
35    login
36  line vty 6
37    login
38  !
39  !
40  end
41
42  sw1#
```

---

At the very bottom we have the details for the console and the VTY lines. It's completely normal for IOS to break down the configuration for 0-6 into ranges (0-4,5,6), even though we configured them together at the same time.

## 1.6 Checking for connectivity using Ping

The ping command sends **ICMP (*Internet Control Message Protocol*)** packets to the destination device and waits for *ICMP replies* from it to determine whether the host is up. It can help detect delays and packet drops, and whether the host is even connected i.e., reachable. We can use it by:

---

```
1  PC-1> ping 192.168.1.11
2  84 bytes from 192.168.1.11 icmp_seq=1 ttl=255 time=3.973 ms
3  84 bytes from 192.168.1.11 icmp_seq=2 ttl=255 time=3.969 ms
4  84 bytes from 192.168.1.11 icmp_seq=3 ttl=255 time=5.951 ms
5  84 bytes from 192.168.1.11 icmp_seq=4 ttl=255 time=3.472 ms
6  84 bytes from 192.168.1.11 icmp_seq=5 ttl=255 time=10.913 ms
```

---