ROUTER COMMANDS

Lu Hue Thu - ritalu2610@yahoo.com
Reference : Commands Guide of Cisco
****

Router > enable
Router # configure terminal
Router(config) #

Router # show running-config
Router # show startup-config
Router # write

1. **Hostname**
   Router(config) # **hostname** *{name}*

2. **Set Password** (to access Privileged Mode)
   Router(config) # **enable password** *{password}*
   Router(config) # **enable secret** *{password}* → Password is encrypted

3. **IP Address**
   Router(config) # **interface** {*interface*} {*interface_number}*
   Router(config-if) # **ip address** *{ip-address} { mask}*
   Router(config-if) # **no shutdown** → Turn on the interface

4. **Telnet**
   Router(config)# line vty 0 4
   Router(config-line)# login
   Router(config-line)# password *{password}*

5. **Enable Router to act as an HTTP server**
   Router (config)# ip http server

6. **Set clock rate on DCE Serial interface**
   Router# show controller serial {*interface_number}* → check to find out whether the serial interface is DCE or DTE.
   Router(config) # **interface** {*interface*} {*interface_number}*
   Router(config-if) #**clock rate** *{clock}*

7. **Routing Protocols**
   * Static
   Router(config)# ip route {*network} [mask] {address|interface}* [distance][permanent]
   $\downarrow$ distance metric for this route
   * RIP – Routing Information Protocol
   Router(config) # **router rip** → Defines IP routing protocol.

Router(config-router) # **network** *{network-number}* → specifies a directly connected network.

* <u>IGRP – Interior Gateway Routing Protocol</u>

Router(config)# router igrp *autonomous-system*
Router(config-router) # **network** *{network-number}*

* <u>EIGRP – Enhanced Interior Gateway Routing Protocol</u>

Router(config)# router eigrp *autonomous-system*
Router(config-router) # **network** *{network-number}*

* <u>OSPF – Open Shortest Path First Protocol</u>

Router(config)# router ospf *process-id*
Router(config-router)# network {*address}* *{wildcard-mask}* **area** *{area-id}*

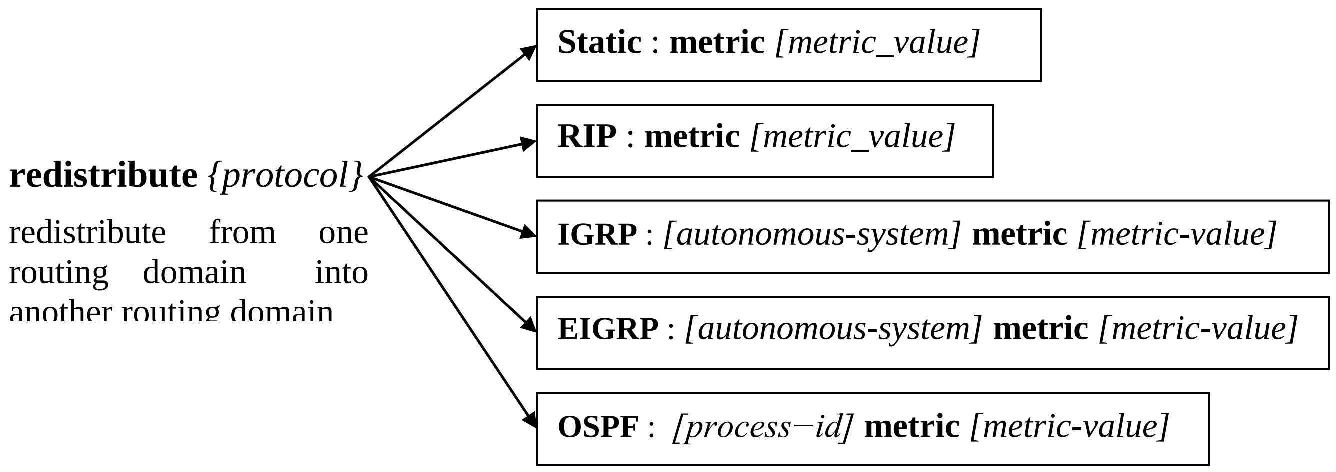
* <u>Show command</u>

Router# show ip route
Router# show ip route w.x.y.z
Router# show ip protocols
Router# show ip *{protocol}* **[interface | database | neighbors | topology]**

Route Source	**Default Distance**
Connected interface	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
EIGRP external route	170
Internal BGP	200
Unknown	255

8. <u>Route Redistribute</u>

default−metric *{bandwidth delay reliability loading mtu}* : set metric value for all redistributed routes (IGRP, EIGRP, OSPF, BGP, EGP).



redistribute *{protocol}*

redistribute from one routing domain into another routing domain

Static : **metric** *[metric_value]*

RIP : **metric** *[metric_value]*

IGRP : *[autonomous-system]* **metric** *[metric-value]*

EIGRP : *[autonomous-system]* **metric** *[metric-value]*

OSPF : *[process−id]* **metric** *[metric-value]*

9. <u>Access Control Lists</u>

<u>Step 1</u> : Choose type of ACL (Standard or Extended) and set parameters for the ACL test statements

- <u>Standard IP ACL</u>
  Router(config) # **access-list** *access-list-number* **{ permit | deny }** *source [mask]*

Parameters	**Description**
access-list-number	Identifies the list that the entry belongs to; a number from 1 to 99
permit | deny	Indicates whether this entry allows or blocks traffic from the specified address
source	Identifies the source IP address
source [mask]	Identifies which bits in the address field are matched; default wildcard mask is 0.0.0.0 <host>

- <u>Extended IP ACL</u>
  Router(config) # **access-list** *access-list-number* **{ permit | deny }** *protocol source source-wildcard [operator port] destination destination-wildcard [operator port] [established] [log]*

Parameters	**Description**
access-list-number	Identifies the list using a number in the ranges of 100 to 199 or 2000 to 2699.
permit | deny	Indicates whether this entry allows or blocks traffic from the specified address
protocol	IP, TCP, UDP, ICMP, IGMP,..
source / destination	Identifies the source / destination IP addresses
source-wildcard destination-wildcard	Wildcard mask (0s : match, 1s : indicate “don’t care” positions)
operator port	It(less than), gt (greater than), eq (equal), neq (not equal) and a port number.
established	For inbound TCP only, allows TCP traffic to pass if the packet uses an established connection. (Ex : it has ACK bits set)

log	Sends a logging message to the console

Step 2 : Enable an interface to use the specified ACL

Router(config) # **ip access-group** *access-list-number* **{ in | out }**

Parameters	**Description**
access-list-number	Indicates number of ACL to be linked to this interface
in | *out*	Selects whether the ACL is applied as an incoming or outgoing filter; out is default.

Note : WILDCARD MASK

Address filtering occurs when you use ACL address wildcard masking to identify how to check or ignore corresponding IP address bits. Wildcard mask for IP address bits uses the number 1 and 0 to identify how to treat the corresponding IP address bits.

- **Wildcard mask bit 0** : **Check** the corresponding bit value in the address.
- **Wildcard mask bit 1** : **Ignore** (do not check) the corresponding bit value in the address.

Ex : Host : 172.30.16.29 → Wildcard Mask : 0.0.0.0 (**host**)
Subnet **:** 172.16.1.0 → Wildcard Mask : 0.255.255.255.
All traffic : 0.0.0.0 → Wildcard Mark : 255.255.255.255 (**any**)

10. Network Address Translation

Private Addresses

Class	**RFC 1918 Internal Address Range**	**CIDR Prefix**
A	10.0.0.0 → 10.255.255.255	10.0.0.0/8
B	172.16.0.0 → 172.31.255.255	172.16.0.0/12
C	192.168.0.0 → 192.168.255.255	192.168.0.0/16

* Step 1 : Configuring NAT

- Static NAT - One Private to One Permanent Public Address Translation

  Router(config)#ip nat inside source static *{local-ip} {global-ip}*

- Dynamic NAT - One Private to One Public Address Translation

  Create an ACL that will identify which private IP addresses

  Router(config) # **access-list** *access-list-number* **permit** *source {mask}*

  Define a pool of usable public IP addresses

  Router(config)#ip nat pool name *{start-ip} {end-ip}* **netmask** *{ prefix-length}*

  Link ACL to the pool of addresses.

  Router(config)#ip nat inside source list *{access-list-number* | *name}* **pool** *{name}*

- PAT - Many Private to One Public Address Translation

  Create an ACL that will identify which private IP addresses

  Router(config) # **access-list** *access-list-number* **{ permit | deny }** *source [mask]*

  Define a pool of usable public IP addresses

Router(config)# ip nat pool *{name} {start-ip end-ip}* **netmask** *{ prefix-length}*
Router(config)# ip nat pool *{name}* **interface** *{interface-number}* **netmask** *{ prefix-length}*

Link ACL to the pool of addresses
Router(config)#ip nat inside source list *{access-list-number | name}* **pool** *{name}* **overload**

* <u>Step 2</u> : Apply NAT on interface

- Define the inside interface
  Router(config)# interface *{interface-number}*
  Router(config-if)# ip nat inside
- Define the outside interface
  Router(config)# int *{interface-number}*
  Router(config-if)# ip nat outside

* <u>Verifying - Troubleshooting NAT and PAT Configuration</u>

Router#show ip nat translations → Displays translation table
Router#show ip nat statistics → Displays NAT statistics
Router#clear ip nat translations inside a.b.c.d **outside** e.f.g.h → Clears a specific translation from the table before it times out
Router#clear ip nat translations *
Router#debug ip nat → Displays information about every packet that is translated
Router#debug ip nat detailed

11. <u>Enable SNMP (Simple Network Management Protocol)</u>

Router(config) # snmp-server community *snmp-community-string* [**acl | acl_name | ipv6 | ro | rw | view**]

<1-99>	Standard IP access list allowing access with this community string
<1300-1999>	Expanded IP access list allowing access with this community string
WORD	Access-list name
ipv6	Specify IPv6 Named Access-List
ro	Read-only access with this community string
rw	Read-write access with this community string
view	Restrict this community to a named MIB view

12. <u>Netflow</u>

- <u>Step 1</u> : Enabling NetFlow Export on interface
  Router(config) # interface *{interface} {interface_number}*
  Router(config-if) # ip route-cache flow
  Router(config-if) # bandwidth *<kbps>*
- <u>Step 2</u> : Exporting NetFlow Data
  Router(config) # ip flow-export source *{interface} {interface_number}*
  Router(config) # ip flow-export destination *{hostname| ip_address} {port}*
  Router(config) # ip flow-export version *{netflow-version}*

- Step 3 : Verifying Device Configuration

  Router # **show ip flow export**

  Router # **show ip cache flow**

  Router # **show ip cache verbose flow**