Router>	User mode	
Router#	Privileged executive mode	**enable** to move to privileged mode
Router(config)#	Global configuration mode	**config t** to move to this mode
Router(config-if)#	Interface configuration mode	**int** to move to this mode
Router(config-line)#	Console or vty line mode	**line** to move to this mode

Hostname: R1(config)#**hostname**   Hostname you select (R1 for example)

Passwords: R1(config)#**enable password** *psswd*   Sets enable password
R1(config)#**enable secret** *psswd*   Sets encrypted enable password
R1(config-line)#**password** psswd   Sets console or telnet passwords

R1(config)#**line console 0**   Enters console line mode
R1(config-line)#**password** *psswd*   Sets console line password
R1(config-line)#**login**   Enables password checking at login

R1(config-line)#**line vty 0 4**   Enters vty line mode for all 5 vty lines
R1(config-line)#**password** *psswd*   Sets vty line password
R1(config-line)#**login**   Enables password checking at login

R1(config)#**service password-encryption**   Applies weak encryption to passwords
R1(config)#**no service password-encryption** Turns off password encryption

Interfaces: R1(config)# **int se 0/0/0**   Interface configuration mode for se0/0/0
R1(config)#**fa0/1**   Interface configuration mode for fa0/1
R1(config-if)#**description** *desc*   Optional interface description
R1(config-if)#**ip address** *I.I.I.I S.S.S.S*   Assigns IP address and SNM to the interface
R1(config-if)#**no shutdown**   Turns the interface on
R1(config-if)#**clock rate** *64000*   Assigns clock rate for the interface (Serial DCE int only)

R1(config)#**loopback0**   A loopback interface
R1(config-if)#**ip address** *I.I.I.I S.S.S.S*   Sets the IP and SNM for the loopback interface

Banners: R1(config)#**banner motd** *#banner message#*   Message of the day banner
R1(config)#**banner login** *#login message*#   Login banner

Lookup: R1(config)#**no ip domain-lookup**   Eliminates DNS search for unrecognized commands

Logging: R1(config-line)#**logging synchronous**   Keeps information lines from interrupting commands

Config Files: Router#**copy run start**   Saves running configuration to startup configuration on NVRAM
Router#**copy start run**   Moves startup configuration into running configuration
Router#**erase start**   Deletes the startup configuration from NVRAM

Reboot: Router#**reload**   reload, reload in XX mins, reload at 00:00 (time), reload cancel

Static: R1(config)#**ip route 172.16.20.0 255.255.255.0 172.16.10.2**   Send to this Next Hop Address
R1(config)#**ip route 172.16.20.0 255.255.255.0 se 0/0/0**   Send out this Exit Interface

Default: R1(config)#**ip route 0.0.0.0 0.0.0.0 172.16.10.2**   Send all packets for networks not in my RT to 172.16.10.2
R1(config)#**ip route 0.0.0.0 0.0.0.0 se 0/0/0**   Send all packets for networks not in my RT out interface se0/0/0

Show: Router#**show ?**   Lists all show commands available

Router#**show interfaces**   Statistics for ALL interfaces

Router#**show int se 0/0/0**   Statistics for a specific interface

Router#**show ip interface brief**   Summary of all interfaces including status and IP address assigned

Router#**show arp**   Displays the ARP table

Router#**show run**   Displays the running configuration in RAM

Router#**show start**   Displays the startup configuration stored in NVRAM

Router#**show ip route**   Displays the routing table

Router#**show controllers**   Will show if an interface is DCE or DTE

Map IP to Name: R1(config)#**ip host NAME w.x.y.z**   Maps a name to an IP Address w.x.y.z (Ex: Ping *NAME*)

Routing: R1(config)#**router RIP**   Enables RIP as a routing protocol

R1(config-router)#**version 2**   Optional, enables RIP Version 2 (needed for subnets)

R1(config-router)#**network w.x.y.z**   The network # of the <u>directly connected</u> network to advertise

R1(config-router)#**no network w.x.y.z**   Removes the network from the RIP process

R1(config)#**no router RIP**   Turns off RIP routing

Router#**debug ip rip**   Displays RIP activity in real time

R1(config-router)#**default-information originate**   Advertises a STATIC DEFAULT ROUTE to other routers

R1(config)#**router eigrp *###***   Turns on EIGRP. ### is the mandatory autonomous system number

R1(config-router)#**network w.x.y.z a.b.c.d.** Network to advertise with wildcard (a.b.c.d) mask (not always used)

R1(config-router)#**no network w.x.y.z a.b.c.d.** Removes the network from the EIGRP process

R1(config)# **ip route 0.0.0.0 0.0.0.0 *IP or exit int***   Creates a static DEFAULT ROUTE using next hop IP or exit interface

R1(config-router)#**redistribute static**   Advertises a STATIC DEFAULT ROUTE to other routers

R1(config)#**no router eigrp *###***   Turns off EIGRP process (AS) ###

R1(config-router)#**no auto-summary**   Turns off network auto-summarization

Router#**show ip eigrp neighbors**   Displays the EIGRP neighbor table

Router#**show ip eigrp neighbors detail**   Displays a detailed EIGRP neighbor table

Router#**show ip eigrp topology**   Displays the topology table with feasible successors

R1(config)#**router ospf ###**   Turns on OSPF. ### is a mandatory process ID #

R1(config-router)#**network w.x.y.z a.b.c.d. area ##** Network to advertise with wildcard (a.b.c.d) mask and mandatory *area-id*

Router#**show ip protocols**   Shows router ID and other information

R1(config)#**router-id w.x.y.z**   Sets the ROUTER'S ID # where w.x.y.z is an IP address

R1#**show ip ospf neighbor**   Shows neighbor relationships

R1#**show ip ospf**   Shows process ID, router ID and other information

R1#**show ip ospf interface *interface ID*** Shows interface information including Hello and Dead intervals

R1#**show interface *interface ID***   Shows interface information including bandwidth

R1(config-if)#**bandwidth *bandwidth-kbps*** Sets the BANDWIDTH for a given interface

R1(config-if)#**ip ospf cost ###**   Sets the COST for a given interface

R1(config-if)# **ip ospf priority *(0-255)*** Sets the PRIORITY for a given interface

R1(config)# **ip route 0.0.0.0 0.0.0.0 *IP or exit int***   Creates a static DEFAULT ROUTE using next hop IP or exit interface

R1(config-router)#**default-information originate**   Advertises a STATIC DEFAULT ROUTE to other routers

R1#**clear ip ospf process**   RESETS <u>*entire*</u> OSPF PROCESS, forcing ospf to recreate neighbors, database & routing tab

R1(config)#**interface loopback #**   Configures a loopback interface

R1(config-if)#**ip helper-address *Ip addr*** Configured on an interface, forwards DHCP broadcasts to DHCP server

Router on a Stick

R1(config)#int fa0/1.10	Creates router sub interface fa0/1.10
R1(config-subif)#encapsulation dot1q 10	Assigns VLAN 10 to this router sub-interface
R1(config-subif)#ip address A.B.C.D W.X.Y.Z	Assigns router interface IP address and SN mask
R1(config-subif)#interface fa0/1	Returns to the main router interface
R1(config-if)#no shut	Enables (Brings it up) the router interface

Access Control Lists   Three Ps: Per Protocol, Per Interface, Per Direction
Placement: Standard – closest to destination network; Extended – closest to source network

Standard: Access-list *[1-99] [1300-1999] [permit/deny] [source address] [wildcard mask] [log]*

R1(config)# access-list 10 permit I.I.I.I W.W.W.W where I = IP & W = Wildcard Mask	
R1(config)# access-list 10 deny I.I.I.I W.W.W.W where I = IP & W = Wildcard Mask	
R1(config)# access-list 10 permit any	**Permit all packets with any source IP address** (*** removes implicit deny ***)
R1(config-if) ip access-group 10 in/out	Access list lines in group 10 applied to interface in inbound/outbound manner
R1# show ip interface fa0/1	Displays any ACLs applied to that interface
R1# show access-lists	Displays contents of all ACLs on router
R1# show access-list *ACL #*	Displays the contents of the ACL by # specified
R1(config)# no access-list 10	Removes ACL # 10

Extended: access-list *[100-199] [2000-2699] [permit/deny] [protocol] [source address] [wildcard mask] [destination address]*
                                    [wildcard mask] [operator [port]] [log]   Operators: eq, neq, gt, lt, range 80 88

R1(config)# access-list 100 permit ip any any (*** removes implicit deny ***)	
R1(config-if) ip access-group 10 in/out	Access list lines in group 10 applied to interface in inbound/outbound manner

Named Access Control Lists

R1(config)# ip access-list [standard/extended]Accounting	Creates a named standard/extended ACL named Accounting
R1(config-ext-nacl)# *permit tcp any host 131.108.101.99 eq smtp*	Permits mail packets from any source to reach 131.108.101.99
R1(config-ext-nacl)# **10** permit tcp any host 131.108.101.99 eq	Use sequence # 10 for this line
R1(config-ext-nacl)# **20** deny ip any any log	Use sequence # 20 for this line
R1(config-ext-nacl)# no 20	Removes line 20 from the list

** Don't forget "permit any" statements to eliminate implicit deny **

R1(config-if) ip access-group Accounting in/out   Access list lines in "Accounting" applied to interface in in/out manner
Sequence numbers start at 10 and increment by 10. If you forget to add a sequence number, the line is added to the end of the list and assigned a # that is 10 > the last #. You can add seq #s i.e. 32 to place it in the correct position. On router reload, seq #s are changed to a 10 increment. 10, 20, 30, 32, 40 becomes 10, 20, 30, 40, 50.

By Port Number				By Process Name			Standard ACL filters only on source IP			
Port	Process	Protocol		Port	Process	Protocol	Extended ACLs filter on source IP, dest IP, protocol #, port #			
20	FTP Data Port	TCP		68	DHCPv4 to client	UDP	Only one ACL per interface, per protocol, per direction			
21	FTP Control Port	TCP		67	DHCPv4 to server	UDP	Place most specific stmts at top of ACL, most general at bottom			
22	SFTP (using SSH)	TCP		53	DNS	TCP & UDP	The last test in any ACL is an **implicit deny**			
22	SSH, SCP	TCP		21	FTP Control Port	TCP	Every ACL must have at least one permit stmt or everything will be denied			
23	Telnet	TCP		20	FTP Data Port	TCP	Place extended ACLs close to the source network or device			
25	SMTP	TCP		80	HTTP	TCP & UDP	Place std ACLs close to the destination network or device			
49	TACACS/TACACS+	TCP		443	HTTPS	TCP	You can use numbers when creating a **named** ACL			
53	DNS	TCP & UDP		143	IMAP4	TCP	Ex: ip access-list extended 150			
67	DHCPv4 to server	UDP		500	IPsec (for VPN with IKE)	UDP	When restricting access through Telnet use access-**class**			
68	DHCPv4 to client	UDP		88	Kerberos	UDP				
69	TFTP	UDP		1701	L2TP	UDP	access-list 100 deny icmp any any echo (denies pings)			
80	HTTP	TCP & UDP		389	LDAP	UDP	access-list 100 deny icmp any any echo-reply (denies ping replies)			
88	Kerberos	UDP		636	LDAP/SSL	UDP	access-list 102 permit icmp any any echo-reply (permits ping replies)			
110	POP3	TCP		636	LDAP/TLS	UDP				
123	NTP	TCP		123	NTP	TCP				
143	IMAP4	TCP		110	POP3	TCP				
161	SNMP	UDP		1723	PPTP	UDP				
443	HTTPS	TCP		3389	Remote Desktop Services	TCP				
389	LDAP	UDP		22	SFTP (using SSH)	TCP				
500	IPsec (for VPN with IKE)	UDP		25	SMTP	TCP				
636	LDAP/TLS	UDP		161	SNMP	UDP				
636	LDAP/SSL	UDP		22	SSH, SCP	TCP				
1701	L2TP	UDP		49	TACACS/TACACS+	TCP				
1723	PPTP	UDP		23	Telnet	TCP				
3389	Terminal Services	TCP		3389	Terminal Services	TCP				
3389	Remote Desktop Services	TCP		69	TFTP	UDP				

NAT/PAT – All Versions

If necessary, enable a routing protocol (RIP, EIGRP). If necessary, set a default network so routers know how to get to the internet.

If necessary, define a static route on the remote (ISP) router stating where public addresses should be routed.

Dynamic Nat – One Private to One Public Address Translation   Dynamic Nat

1. Define a pool of useable <u>public</u> IP addresses on router performing NAT.
       R1(config)# ip nat pool Tom 64.64.64.70 64.64.64.126 netmask 255.255.255.128

2. Create an ACL that identifies which <u>private</u> IP addresses will be translated: **R1(config)# access-list 1 permit 172.16.10.0 0.0.0.255 any**
       Or   R1(config)# access-list 110 permit ip 172.16.10.0 0.0.0.255 any

3. Link the ACL to the pool of addresses (create the translation): **R1(config)# ip nat inside source list 1 (or 110) pool Tom**

4. Define inside interface (contains the private addresses): **R1(config)# int fa0/1; R1(config-if)# ip nat inside**

5. Define outside interface (leading to the public network): **R1(config)# int s0/0/0; R1(config-if)# ip nat outside**

PAT – Many Private to One Public Address Translation   PAT

1. Define a pool of useable <u>public</u> IP addresses on router performing PAT. Start & end of the pool is the same.
       R1(config)# Ip nat pool Tom 64.64.64.70 64.64.64.70 netmask 255.255.255.128 [Note same IP address]
              *** Not needed if you just use the router interface address as in option 3B below ***

2. Create an ACL that identifies which <u>private</u> IP addresses will be translated: **R1(config)# access-list 1 permit 172.16.10.0 0.0.0.255 any**
              Or R1(config)# access-list 110 permit ip 172.16.10.0 0.0.0.255 any

3A. Link the ACL to the pool of addresses (create the translation): **R1(config)# ip nat inside source list 1 (or 110) pool Tom overload**
Or,
3B. Just use the router interface for the translation address: **R1(config)# ip nat inside source list 1 (or 110) int s0/0/0 overload**

4. Define inside interface (contains the private addresses): **R1(config)# int fa0/1; R1(config-if)# ip nat inside**

5. Define outside interface (leading to the public network): **R1(config)# int s0/0/0; R1(config-if)# ip nat outside**

Static Nat – One Private to one Permanent Public Address Translation   Static Nat

1. Create STATIC mapping on router that will perform NAT: **R1(config)# ip nat inside source static 172.16.10.5 64.64.64.65**
       Permanently translates the inside address of 172.16.10.5 to 64.64.64.65.

2. Define inside interface (contains the private addresses): **R1(config)# int fa0/1; R1(config-if)# ip nat inside**

3. Define outside interface (leading to the public network): **R1(config)# int s0/0/0; R1(config-if)# ip nat outside**

Verifying NAT and PAT

R1# show access-list	Displays the access lists
R1# show ip nat translations	Displays the translation table
R1# clear ip nat translations *	Clears the entire translation table before the entries time out.
R1# debug ip nat	Shows information about every packet that is translated.

DHCP Server

1. Exclude statically assigned or other addresses.   **R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.10** (a range here)
2. Create the DHCP pool named <u>Tom</u>.   **R1(config)# ip dhcp pool** *Tom* (Note: each network needs its own pool)
3. Specify the subnet used to assign the addresses.   **R1(dhcp-config)# network 192.168.10.0 255.255.255.0**
4. Identify the DNS server for the network.   **R1(dhcp-config)# dns-server 192.168.11.5**
5. Set the default router (gateway) for the network.   **R1(dhcp-config)# default-router 192.168.10.1**
6. If needed, configure a DHCP helper address.   **R1(config)# int fa0/0** (moves to interface configuration mode)
    R1(config-if)# ip helper-address 192.168.11.5 (forwards DHCP broadcasts as unicasts)

    Placed on the router interface where the DHCP broadcast requests from clients arrive as routers normally drop broadcasts.

7. Optional: a router interface can acquire an ip address via DHCP.   **R1(config)# int fa0/0** (moves to interface configuration mode)
    R1(config-if)# ip address dhcp

Port Forwarding   Similar to Static NAT

R1(config)# **ip nat inside source static <protocol> <inside IP address> <inside port #> <Inside Global IP or Interface> <External Port>**
    Example from below: tcp   192.168.1.50   80   209.165.200.50   80
R1(config)# **int fa0/1;** R1(config-if**)# ip nat inside**
R1(config)# **int fa0/0;** R1(config-if)# **ip nat outside**
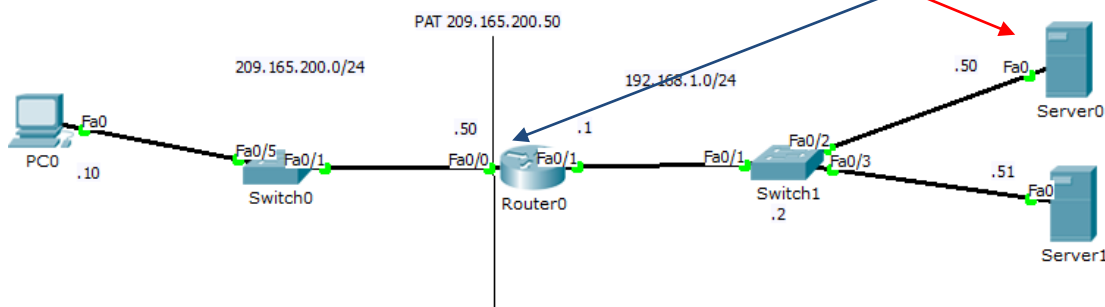
ip nat inside source list <u>1</u> interface FastEthernet0/0 overload   (Sets up PAT)
ip nat inside source static tcp 192.168.1.50 80 209.165.200.50 80
ip nat inside source static tcp 192.168.1.51 80 209.165.200.50 2000
ip nat inside source static tcp 192.168.1.2 23 209.165.200.50 23
!
access-list <u>1</u> permit 192.168.1.0 0.0.0.255

IPv6

R1(config)# ipv6 unicast-routing	Enables IPv6 traffic forwarding
R1(config)# int fa0/0	Moves to interface configuration mode
R1(config-if)# ipv6 address 2001:db8:3c4d:1:0260:6dFF.FE73:1987/64	Specifies the entire 128-bit global IPv6 address
R1(config-if)# ipv6 address 2001:db8:3c4d:1::/64 eui-64	Uses the MAC address + padding to complete the address
R1(config-if)# ipv6 enable	Automatically configures a link-local address

RIPng R1(config-if)# ipv6 rip *[process-id name or number]* enable   Enables RIP on the <u>INTERFACE</u>
    R1(config)# ipv6 router rip *[process-id name or number]*   Enters router configuration for redistribution, etc.

EIGRPv6   R1(config)# ipv6 router eigrp *[AS number]*   Enters router configuration mode
    R1(config-rtr)# no shutdown   <u>No shutdown is required</u>
    R1(config-if)# ipv6 eigrp *[AS number]*   Enables IPv6 on the interface

OSPFv3   R1(config)# ipv6 router ospf *[process #]*   Creates the OSPFv3 process
    R1(config-if)# ipv6 ospf *[process #]* area 0   Enables OSPFv3 on the interface into Area 0

Frame Relay (sub interfaces, dlcis & ip addresses for example only)

R1(config)# int s0/0
R1(config-if)# encapsulation frame relay
R1(config-if)# frame-relay lmi-type (ansi | cisco | q933a)   Sets the LMI type
R1(config-if)# ip address 172.16.20.1 255.255.255.0   Sets the ports IP address
R1(config-if)# frame-relay interface-dlci 101   Assigns a DLCI to the interface
Subinterfaces:
R1(config)# int s0/0
R1(config-if)# encapsulation frame-relay
R1(config-if)# interface s0/0.12 point-to-point   Creates sub-interface 12
R1(config-subif)# ip address 192.168.1.1 255.255.255.0   Assigns IP and netmask to sub-interface
R1(config-subif)# frame-relay interface-dlci 12 point-to-point Assigns DLCI number 12

Point-to-Point & HDLC

High-Level Data-Link Control (HDLC) is the default encapsulation used by Cisco routers on serial links. PPP is an open standard and the most used today.
R1(config)# int s0/0
R1(config-if)# encapsulation ppp   (must be enabled on both interfaces)

PPP Authentication: **Note: because PAP does not encrypt passwords, CHAP is recommended**

On Equipment:
R1(config)# username *(router's name i.e. Router1)* password *password* Ex: username Router1 password cisco
R1(config)# int s0/0
R1(config-if)# ppp authentication <u>pap</u>   PAP
R1(config-if)# ppp <u>pap</u> sent-username (name of router connecting to) password *password* **Example follows:**
R1(config-if)# ppp <u>pap</u> sent-username Router1 password cisco   PAP
R1(config-if)# ppp authentication <u>chap</u>   CHAP
R1(config-if)# ppp <u>chap</u> sent-username (name of router connecting to) password password **Example follows:**
R1(config-if)# ppp <u>chap</u> sent-username Router1 password cisco   CHAP

On Packet Tracer:
R1(config)#username **R2** password cisco   **Note: username R2 is the router your are connecting to**
R1(config)#interface s0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
R1(config-if)#

R2(config)#username **R1** password cisco   **Note: username R1 is the router your are connecting to**
R2(config)#interface s0/0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication chap

VoIP – Add a Phone

R1(config)# ephone-dn *5*   (next dial number i.e. 5)
R1(config-ephone-dn)# *number*   (phone number i.e. 5555)
R1(config-ephone-dn)# exit
R1(config)# ephone *5*   (ephone dial number i.e. 5)
R1(config-ephone)# mac-address *the phone's mac address*
R1(config-ephone)# type *7960*   (the phone's model type i.e. 7960)
R1(config-ephone)# button 1:*5*   (ephone dial button i.e. 1:5)
Note: On the switch set the port: switchport voice vlan __