Basic Device Configuration

```
hostname <name>
no ip domain-lookup
line console 0
no exec-timeout
logg sync
copy run start
clock rate <value>
ip address x.x.x.x x.x.x.x
ip default-gateway x.x.x.x
```

Static routing

Static route command

```
R1(config)# ip route <network-address> <subnet-mask> { next-hop ip address |
exit-intf [ip-address]} [distance]

R1(config)# ipv6 unicast-routing
R1(config)# ipv6 route <ipv6-prefix/prefix-length> {next hop ipv6-address |
exit-intf [next hop ipv6-address]} [distance]
```

Default static route command

```
R1(config)# ip route <quad-zero-network-address> <quad-zero-wildcard-address>
{next-hop ip-address | exit-intf}
```

Static route verification

```
show ip route static
```

Single-Area OSPFv2 Configuration

Configure OSPF using network command and wildcard mask syntax

```
R1(config)# router ospf process-id
R1(config-router)# router-id x.x.x.x
R1(config-router)# network <network-address> <wildcard-mask> area <area-id>
```

Configure OSPF routing using interface IP addresses and quad-zero masks

```
R1(config)# router ospf process-id
R1(config-router)# network <interface IPv4 address> <quad-zero wildcard mask>
area <area-id>
```

Configure OSPF on router interfaces using the ip ospf Command

```
R1(config)# router ospf process-id
R1(config-router)# int <router interface>
R1(config-if)# ip ospf <process-id> area <area-id>
```

Passive Interface

```
R1(config-router)# passive-interface <interface name>
```

Modify Router ID

```
R1(config-router)# router-id x.x.x.x
R1 # clear ip ospf process
```

Configure OSPF priority

```
R1(config)# interface <router interface>
R1(config-if)# ip ospf priority <priority-value>
R1 # clear ip ospf process
```

Adjust the Reference Bandwidth

```
R1(config-router)# auto-cost reference-bandwidth <value>
```

Manually Set OSPF Cost Value

```
R1(config)# interface <router interface>
R1(config-if)# ip ospf cost <cost-value>
```

Modify OSPF Intervals

```
R1(config-if)# ip ospf hello-interval [seconds]
R1(config-if)# ip ospf dead-interval [seconds]
```

Default route propagation

```
R1(config)# ip route <quad-zero-network-address>
R1(config)# router ospf <process-id>
R1(config-router)# default-information originate
```

Verify OSPF Neighbors

```
R1 # show ip interface brief
R1 # show ip route
R1 # show ip ospf neighbor
R1 # show ip protocols
R1 # show ip ospf
R1 # show ip ospf interface
```

ACL

Numbered standard IPv4 ACL Syntax

```
R1(config)# access-list <access-list number> {deny | permit | remark text}
<source> [source-wildcard] [log]

R1(config)# int <router-interface>
R1(config)# ip access-group <access-list number> {in | out}
```

Named standard IPv4 ACL Syntax

```
R1(config)# ip access-list standard <access-list name>
R1(config-std-nacl)# {deny | permit | remark text} <source>
[source-wildcard]

R1(config)# int <router-interface>
R1(config-if)# ip access-group <access-list name> {in | out}
```

Modify Standard ACL

```
R1 # sh access-lists
R1(config)# ip access-list standard {access-list number | access-list name}
R1(config-std-nacl)# no <sequence number>
R1(config-std-nacl)# <sequence number> {deny | permit | remark} <source>
[source-wildcard]
```

Numbered extended IPv4 ACL Syntax

```
R1(config)# access-list access-list-number {deny | permit | remark text} protocol source source-wildcard [operator {port}] destination destination-wildcard [operator {port}] [established] [log]
R1(config)# int <router-interface>
R1(config-if)# ip access-group {access-list-number} {in | out}
```

Named extended IPv4 ACL Syntax

```
R1(config)# ip access-list extended <access-list name>
R1(config-ext-nacl)# {deny | permit | remark text} protocol source source-wildcard [operator {port}] destination destination-wildcard
R1(config)# int <router-interface>
R1(config-if)# ip access-group {access-list-name} {in | out}
```

Modify Extended ACL

```
R1 # sh access-lists
R1(config)# ip access-list extended {access-list number | access-list name}
R1(config-ext-nacl)# no <sequence number>
R1(config-ext-nacl)# {deny | permit | remark text} protocol source source-wildcard [operator {port}] destination destination-wildcard
```

Writing Access Lists to Restrict Telnet Access

```
R1(config)# access-list <access-list number> {deny | permit} <source> [source-wildcard]
R1(config)# line vty 0 4
R1(config-line)# access-class {access-list-number | access-list-name} { in | out }


R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class {access-list-number | access-list-name} in
```

Verify ACL

```
R1# show run
R1# show access-lists
```

NAT for IPv4

Configure static NAT

```
R1(config)# ip nat inside source static <inside local address> <inside global address>
R1(config)# interface <router-interface>
R1(config-if)# ip nat inside
R1(config-if)# interface <router-interface>
R1(config-if)# ip nat outside
```

Configure dynamic NAT

```
R2(config)#ip nat pool <pool_name> <starting public address> <ending public
address> netmask <subnet-mask>
R2(config)# access-list <access-list number> {deny | permit} <source>
[source-wildcard]
R2(config)#ip nat inside source list <access-list number> pool <pool_name>
R2(config)# int <router-interface>
R2(config-if)# ip nat outside
R2(config-if)# int <router-interface>
R2(config-if)# ip nat inside
```

Configure PAT to use an address pool

```
R2(config)#ip nat pool <pool_name> <starting public address> <ending public
address> netmask <subnet-mask>
R2(config)# access-list <access-list number> {deny | permit} <source>
[source-wildcard]
R2(config)#ip nat inside source list <access-list number> pool <pool_name>
overload
R2(config)# <router-interface>
R2(config-if)# ip nat outside
R2(config-if)# <router-interface>
R2(config-if)# ip nat inside
```

Configure PAT using an Interface / Configure PAT to use a single IPv4 address

```
R2(config)# access-list <access-list number> {deny | permit} <source>
[source-wildcard]
R2(config)# ip nat inside source list <access-list number> interface
<router-interface> overload
R2(config)# int <router-interface>
R2(config-if)#ip nat inside
R2(config-if)# int <router-interface>
R2(config-if)#ip nat outside
```

NAT Verification

```
show ip nat translations
show ip nat statistics
clear ip nat translation *
```

Implement DHCPv4

```
R1(config)#ip dhcp excluded-address <starting address> <ending address>
R1(config)#ip dhcp pool <pool_name>
R1(dhcp-config)#network <network-address> <subnet-mask>
R1(dhcp-config)#domain-name <value>
R1(dhcp-config)#dns-server x.x.x.x
R1(dhcp-config)#default-router x.x.x.x

R2(config)#int <router-interface>
R2(config-if)#ip helper-address x.x.x.x - IPv4 address for DHCP server
```

ipconfig /renew on PC to acquire an IP address

VPN

Configuring point-to-point GRE VPN Tunnel

```
R1(config)# interface tunnel 0
R1(config-if)# ip address <tunnel ip-address> <subnet-mask>
R1(config-if)# tunnel source {ip-address | interface-type}
R1(config-if)# tunnel destination ip-address
```

Verification

```
show ip int brief
```

Enabling routing over the GRE Tunnel

```
R1(config)# router ospf 1
R1(config-router)# network <tunnel network-address> <wildcard-mask> area
<area-id>
```

Slash	Subnet mask	Wildcard mask
/25	128	
/26	192	
/27	224	
/28	240	
/29	248	
/30	252	
/31	254	
/32	255	