VLAN

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Create a VLAN per sub-interface.

```
interface <technology> <interface #>.<sub-interface #>
```

4). Configure encapsulation protocol (VLAN | IEEE 802.1Q), and VLAN ID.

```
encapsulation dot1Q <VLAN ID [1;4094]>
```

5). Assign a VLAN default gateway (.100 in IP's last octet).

```
ip address <IP address> <IP address' net-mask>
```

6). Go to main interface configuration.

```
interface <technology> <interface #>
```

7). Prevent the interface from going out.

```
no shutdown
```

Switch

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Create VLANs.

```
vlan <VLAN ID [1;4094]>
```

4). Give a VLAN a name.

```
name <name>
```

5). Identify trunk ports.

5.1). Get in identified trunk ports.

```
interface <technology> <interface #>
```

5.2). Set trunk ports.

```
switchport mode trunk
```

5.3). Prevent the interface from going out.

```
no shutdown
```

6). Identify access ports.

6.1). Get in identified access ports.

```
interface <technology> <interface #>
```

6.2). Set trunk ports.

```
switchport mode access
```

6.3). Prevent the interface from going out.

```
switchport access vlan <VLAN # created>
```

6.4). Prevent the interface from going out.

```
no shutdown
```

PC

Configure IP address options and fields.

Static routing

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Set networks for routing.

```
ip route <destination network IP> <network IP's net-mask> <next hop's IP address | inside
router's interface>
```

RIP

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Get in RIP configuration.

```
router rip
```

4). Set RIP version

```
version <1 | 2>
```

5). Specify that giving networks will not be auto-summarized.

```
no auto-summary
```

6). Set networks for routing.

```
network <network IP>
```

OSPF

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Get in OSPF configuration, by assigning a process.

```
router ospf <Process ID [1;65536]>
```

4). Set networks for routing.

```
network <network IP> <network IP's wildcard> area <#>
```

Optional

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Get in an interface which has configured OSPF.

```
interface <technology> <interface #>.<sub-interface #>
```

4). Configure OSPF cost per interface.

```
ip ospf cost <# of cost>
```

Routing redistribution

Router

Static & "another routing protocol"

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Set redistribution in order to reach convergence between both routing protocols.

3.1). In case static routing wants to know "another routing protocol" learned routes: *"The other" goes on duty ...*

3.1.1). Get in routing protocol configuration.

```
router <routing protocol>
```

3.1.2). Establish the redistribution.

```
redistribution static metric <# of routing protocol's metric>
```

3.2). In case "another routing protocol" wants to know static routing learned routes: *"The other" approaches to static routing ...*

3.2.1). Get in routing protocol configuration.

```
router <routing protocol>
```

3.2.2). Establish the redistribution.

```
redistribution connected metric <# of routing protocol's metric>
```

RIP & OSPF

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Set redistribution in order to reach convergence between both routing protocols.

3.1). In case RIP wants to know OSPF learned routes: *OSPF does the job ...*

3.1.1). Get in OSPF configuration.

```
router ospf <Process ID [1;65536]>
```

3.1.2). Establish RIP redistribution.

```
redistribution rip metric <# of metric or cost> subnets
```

3.2). In case OSPF wants to know RIP learned routes: *RIP does the job ...*

3.2.1). Get in RIP configuration.

```
router rip
```

3.2.2). Establish OSPF redistribution.

```
redistribution ospf <Process ID [1;65536]> metric <hop-by-hop #>
```

DHCP

Service device

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Establish DHCP.

3.1). Get in protocol's configuration.

```
ip dhcp pool <pool name string line>
```

3.2). Set a domain name for the server device.

This configuration is up to DNS resolver configuration. E.g. "cisco.com".

```
domain-name <DHCP server name string line>
```

3.3). Set a network to assign dynamically IP addresses.

```
network <network IP> <network IP's net-mask>
```

3.4). Set a default gateway for that previous network.

```
default-router <previous network's default gateway>
```

3.5). Set a DNS server IP address.

```
dns-server <DNS server's IP address>
```

Router

1). Establish DHCP service's IP helper address.

2). Get in enable configuration.

```
enable
```

3). Get in configuration terminal.

```
configure terminal
```

4). Get in identified interface.

The right interface (to establish an IP helper address) is the one that helps the network to find that DHCP service; its assigned IP is part of the network that does not know where is that service device.

```
interface <technology> <interface #>(.<sub-interface #>)
```

5). Set IP helper address.

```
ip helper-address <DHCP service device's IP address>
```

6). Prevent the interface from going out.

```
no shutdown
```

PC

Configure IP address options and fields.

DNS

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Enable DNS server configuration.

```
ip dns server
```

4). Enable domain lookup.

```
ip domain-lookup
```

5). Set a public IP address for the service.

```
ip name-server <DNS' IP address>
```

6). Configure host-names.

```
ip host <host name string line> <host's IP address>
```

PC

Configure DNS' IP address.

NAT

Inside Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Create a pool of public IP addresses.

```
ip nat pool <pool name> <first IP> <last IP> netmask <net-mask>
```

4). Assign a pool to an access-list.

```
ip nat inside source list <[1;99] for basic lists | [100;199] for extended lists> pool <pool
name> overload
```

5). Check if it is necessary to use static NAT configuration.

5.1). Directly associate a private IP with a public IP (registered in NAT's pool).

```
ip nat inside source static <private IP> <public IP>
```

6). Configure an access-list for private IP addresses.

```
access-list <list #> permit <network IP> <network IP's wildcard>
```

7). Identify inside or outside interfaces.

7.1). Get in identified interface.

```
interface <technology> <interface #>(.<sub-interface #>)
```

7.2). Define the role of the port in the NAT configuration.

```
ip nat <inside | outside>
```

7.3). Prevent the interface from going out.

```
no shutdown
```

8). Define a by default static route.

```
ip route 0.0.0.0 0.0.0.0 <next hop's IP address | inside router's interface>
```

Outside Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Configure static routing for public IP addresses pool.

```
ip route <destination network IP> <network IP's net-mask> <next hop's IP address | inside
router's interface>
```

4). Redistribute learned addresses (from static to another routing protocol).

4.1). Get in routing protocol configuration.

```
router <routing protocol>
```

4.2). Establish the redistribution.

```
redistribution static metric <# of routing protocol's metric>
```

ACL

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Create an access list.

3.1). Create a basic access list.

```
access-list <ID # [1;99]> <options> <Source's IP address> <Source's wildcard of IP address>
```

3.2). Create a extended access list.

```
access-list <ID # [100;199]> <options> <protocol> <Source's IP address> <Source's wildcard
of IP address> (eq <port>) <Destination's IP address> <Destination's wildcard of IP address>
(eq <port>)
```

4). Assign an access list to a port.

4.1). Get in interface configuration.

```
interface <technology> <interface #>(.<sub-interface #>)
```

4.1.1). Apply access list from inside network to outside network traffic.

```
ip access-group <ID #> in
```

4.1.2). Apply access list from outside network to inside network traffic.

```
ip access-group <ID #> out
```

4.2). Prevent the interface from going out.

```
no shutdown
```

FTP client

PC

1). Establish FTP connection.

1.1). Get in server access.

```
ftp <FTP server's IP address>
```

1.2). Validate secure device features through FTP credentials.

Username:

```
<FTP username set on server>
```

Password:

```
<FTP password set on server>
```

VTY

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

3). Establish VTY.

3.1). Get in VTY configuration.

```
line vty <minimum # of lines [0;15]> <maximum # of lines [0;15]>
```

3.2). Set password settings.

3.2.1). Set no password.

```
no password
```

3.2.2). Set password.

```
password <password string line>
```

3.3). Set login settings with VTY credentials, with no extra options.

3.3.1). Set no login.

```
no login
```

3.3.2). Set login.

```
login
```

4). Get in interface configuration.

```
interface loopback <interface #>
```

It is possible to choose another technology. However, since VTY is a virtual terminal protocol, then it is profitable to use a virtual interface also; in order to leverage other routes which support the connection.

4.1). Add an IP address to a loopback interface.

```
ip address <IP address> <IP address' net-mask>
```

*). It is not necessary to prevent the interface from going out, because it is a virtual interface. If the device is on, its virtual interfaces too.

5). Define password configuration for the device.

```
enable secret (<options>) <password string line>
```

PC

1). Establish VTY connection.

1.1). Get in router access.

```
telnet <Loop-back IP address>
```

1.2). Validate secure protocol communication.

Password:

```
<VTY password set on router>
```

1.3). Get in enable configuration.

```
enable
```

1.4). Validate secure device features.

Password:

```
<device secret password set on router>
```

SSH

Router

1). Get in enable configuration.

```
enable
```

2). Get in configuration terminal.

```
configure terminal
```

4). Set a different host name. -Mandatory-

```
hostname <host name string line>
```

5). Set a domain name for the SSH server.

This configuration is up to DNS resolver configuration. E.g. "cisco.com".

```
ip domain-name <SSH server name string line>
```

6). Define username and password configuration for the device.

```
username <username string line> privilege <user privilege level #> secret (<options>)
<password string line>
```

7). Define security keys.

```
crypto key generate rsa
```

Do you really want to replace them?:

```
yes
```

How many bits in the modulus:

```
<[360;2048]>
```

8). Set SSH version.

```
ip ssh version 2
```

9). Define SSH through VTY.

9.1). Get in VTY configuration.

```
line vty <minimum # of lines [0;15]> <maximum # of lines [0;15]>
```

9.1.1). Define what protocol is used through VTY.

```
transport input ssh
```

9.1.2). Set login settings for local authentication (with router's credentials).

```
login local
```

10). Get in interface configuration.

```
interface loopback <interface #>
```

It is possible to choose another technology. However, since VTY is a virtual terminal protocol, then it is profitable to use a virtual interface also; in order to leverage other routes which support the connection.

10.1). Add an IP address to a loopback interface.

```
ip address <IP address> <IP address' net-mask>
```

*). It is not necessary to prevent the interface from going out, because it is a virtual interface. If the device is on, its virtual interfaces too.

PC

1). Establish SSH connection.

1.1). Get in router access.

```
ssh -l <Router's host-name> <Loop-back IP address>
```

1.2). Validate secure device features.

Password:

```
<device secret password set on router>
```