**BITÁCORA PARA LA CONFIGURACIÓN DE LA RED DE INFRAESTRUCTURA (EIGRP)**

**DATOS DEL ESTUDIANTE:**

MIC-Complexivo 202550 - EIGRP

Denep Maliza

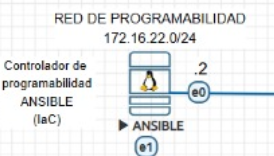
CI: 172961573-0

ID: L00409416

Correo: [demaliza@espe.edu.ec](mailto:demaliza@espe.edu.ec)

**-------- Instalación Ansible -------------**

En nuestra topología, accedemos a la máquina Ubuntu o Controlador de Programabilidad ANSIBLE. Las credenciales serán las siguientes.



Login as: root

Password: Test123

1. **Verificar IP:**

**Ingresar el comando** ip add **(verificamos que no tengamos ninguna ip en nuestra maquina)**

vi /etc/netplan/00-installer-config.yaml (Accedemos al archivo .yaml para agregar las ip)

Copiamos y pegamos todo este texto:

network:

ethernets:

ens3:

dhcp4: no

addresses: [172.16.22.2/24]

gateway4: 172.16.22.1

nameservers:

addresses: [8.8.8.8]

ens4:

dhcp4: no

addresses: [10.10.10.11/24]

gateway4: 10.10.10.1

nameservers:

addresses: [8.8.8.8]

version: 2

netplan apply

*ens3 agregamos la ip de nuestra red de programabilidad y con ens4 tenemos acceso a la red de internet. Para editar el archivo, se pulsa I para insertar y para guardar los cambios ESC y :wq!. Aplicamos los cambios con netplan apply.*

*Aplicado los cambios, se hace nuevamente* ip add para verificar las direcciones ip.

1. **Verificar salida a Internet**

gcloud (EVE-ng - entrar por SSH)

Abrimos ssh en el navegador, entramos como su –

Contraseña: eve

iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o pnet0 -j MASQUERADE

Esta regla permite que toda la red 10.10.10.0/24 salga a Internet usando la IP de la interfaz pnet0, ocultando las IPs internas. Es decir, es NAT de salida para compartir la conexión.

Dato adicional, es recomendable hacer un ping 8.8.8.8 para corroborar nuestra conexión y configuración del. yaml

1. **Update y Upgrade a Ubuntu**

sudo apt update

Para actualizar todo el sistema.

sudo apt upgrade -y

Actualizar todos los paquetes instalados.

Si sale un aviso, simplemente dar enter.

1. **Instalar Ansible**

sudo apt install software-properties-common

Instala el paquete software-properties-common, que incluye herramientas para añadir y gestionar repositorios.

sudo add-apt-repository --yes --update ppa:ansible/ansible

Agrega el repositorio PPA oficial de Ansible a tu sistema y actualiza la lista de paquetes automáticamente.

sudo apt install ansible

Procede a instalar ansible.

ansible – version

Verificamos la versión instalada.

sudo apt install python3-pip

Instalar y gestionar librerías de Python 3 desde la terminal.

pip install --user ansible-pylibssh

Añade soporte **libssh** a Ansible, y lo hace solo para tu usuario, no para todo el sistema.

**Configuración Manual**

**\*\*\*\*MATRIZ\*\*\*\***

**--S8--**

enable

config t

hostname S8

vlan 220

name PRESIDENCIA

vlan 222

name DC

exit

int e0/1

switchport mode access

switchport access vlan 220

int e0/2

switchport mode Access

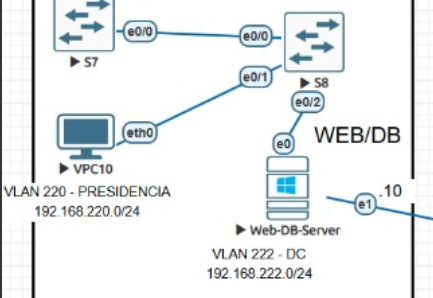
switchport access vlan 222

int e0/0

switchport trunk encap dot1q

switchport mode trunk

do wr



**--S7--**

enable

config t

hostname S7

vlan 220

name PRESIDENCIA

vlan 222

name DC

exit

int e0/0

switchport trunk encap dot1q

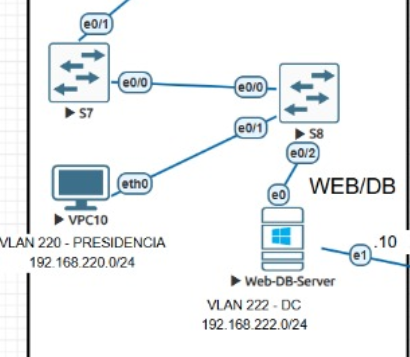
switchport mode trunk

int e0/1

switchport trunk encap dot1q

switchport mode trunk

do wr



**--R1--**

enable

config t

hostname R1

int e0/1.220

encap dot1q 220

ip add 192.168.220.1 255.255.255.0

int e0/1.222

encap dot1q 222

ip add 192.168.222.1 255.255.255.0

int e0/1

no shut

int s1/0

ip add 5.5.22.1 255.255.255.252

no shut

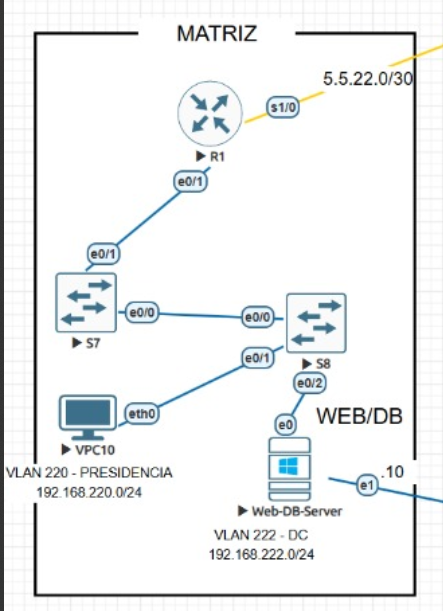
router eigrp 22

network 192.168.220.0 0.0.0.255

network 192.168.222.0 0.0.0.255

network 5.5.22.0 0.0.0.3

do wr



**\*\*\*\*SUCURSAL\*\*\*\***

**---S9---**

enable

config t

hostname S9

vlan 22

name MIN1

vlan 25

name SEGURIDAD

exit

int e0/1

switchport mode access

switchport access vlan 22

int e0/3

switchport mode access

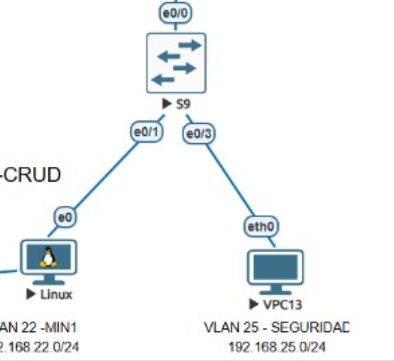
switchport access vlan 25

int e0/0

switchport trunk encap dot1q

switchport mode trunk

do wr



**--R6--**

enable

config t

hostname R6

int e0/0.22

encap dot1q 22

ip add 192.168.22.1 255.255.255.0

int e0/0.25

encap dot1q 25

ip add 192.168.25.1 255.255.255.0

int e0/0

no shut

int s1/1

ip add 5.5.22.22 255.255.255.252

no shut

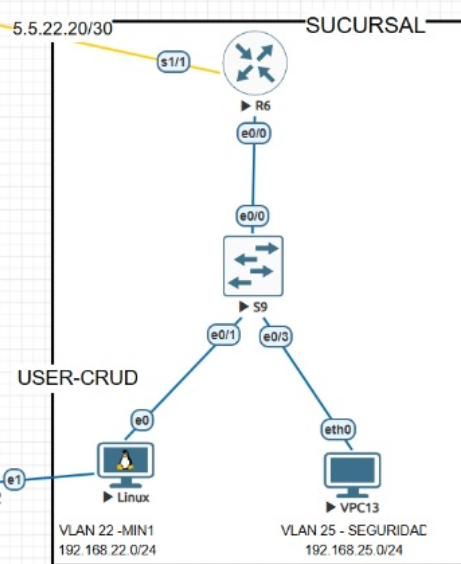
router eigrp 22

network 192.168.22.0 0.0.0.255

network 192.168.25.0 0.0.0.255

network 5.5.22.20 0.0.0.3

do wr



**\*\*\*\*ISP\*\*\*\***

**---RED DE PROGRAMABILIDAD---**

**--R3--**

enable

config t

hostname R3

int e0/3

ip add 172.16.22.33 255.255.255.0

no shut

exit

ip domain-name dcco.espe.ec

username ESPE22 privilege 15 secret ITIV22

enable secret ESPE22

crypto key generate rsa

1024

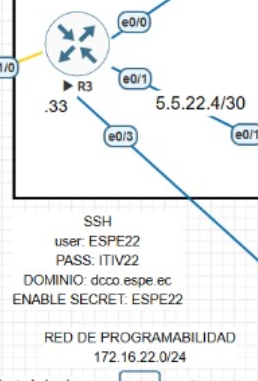
ip ssh version 2

line vty 0 4

transport input ssh

login local

do wr



--R4--

enable

config t

hostname R4

int e0/3

ip add 172.16.22.44 255.255.255.0

no shut

exit

ip domain-name dcco.espe.ec

username ESPE22 privilege 15 secret ITIV22

enable secret ESPE22

crypto key generate rsa

1024

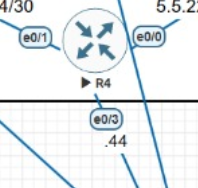
ip ssh version 2

line vty 0 4

transport input ssh

login local

do wr



**--R5--**

enable

config t

hostname R5

int e0/3

ip add 172.16.22.55 255.255.255.0

no shut

exit

ip domain-name dcco.espe.ec

username ESPE22 privilege 15 secret ITIV22

enable secret ESPE22

crypto key generate rsa

1024

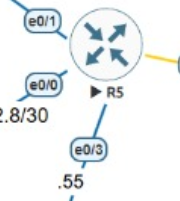
ip ssh version 2

line vty 0 4

transport input ssh

login local

do wr



**--R2--**

enable

config t

hostname R2

int e0/3

ip add 172.16.22.22 255.255.255.0

no shut

exit

ip domain-name dcco.espe.ec

username ESPE22 privilege 15 secret ITIV22

enable secret ESPE22

crypto key generate rsa

1024

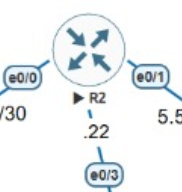
ip ssh version 2

line vty 0 4

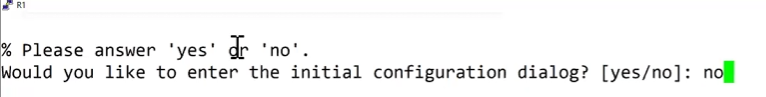
transport input ssh

login local

do wr



Una vez terminada de ingresar las configuraciones de la red de programabilidad, se procede a copiar y pegarlo en cada uno de los equipos.

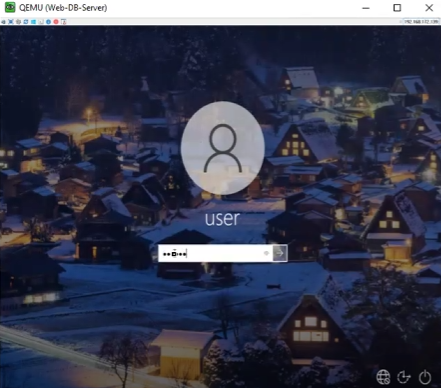


En el caso de los routers, suele salir esa pregunta, debemos poner “no”

Con los switches no hay problema.

**------------SSH EN ANSIBLE---------**

**Pare este paso, entramos a la maquina Windows server**

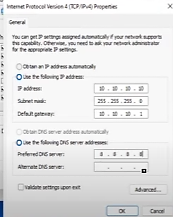
****

**Contraseña:** Test123

Una vez dentro, vamos a la tarjeta de red a ingresar las direcciones ip.

****

**La red es va la dirección ip de la web/db y la red 2 es para el internet. En este paso solo necesitamos la segunda red para internet.**

****

Para realizar el paso de PUTTY, es recomendable PRIMERO cambiar todo esto de ANSIBLE como tenemos en nuestra topología, es decir la direcciones ip. AUN NO PEGAMOS ESTO EN NINGUN ARCHIVO.

**\*\*\*ANSIBLE\*\*\***

cd /etc/ansible

vi hosts

[R3]

172.16.22.33

[R4]

172.16.22.44

[R5]

172.16.22.55

[R2]

172.16.22.22

[ISP:children]

R3

R4

R5

R2

[ISP:vars]

ansible\_network\_os=cisco.ios.ios

ansible\_connection=ansible.netcommon.network\_cli

ansible\_user=ESPE22

ansible\_password=ITIV22

-----

ansible-inventory --list -y

Con este revisamos que esta bien

-----

vi MIC-2.yaml

---

- hosts: R3

tasks:

- name: Configuracion de interface R3-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.5 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R3-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.13 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R3-s1/0

ios\_config:

parents: "interface s1/0"

lines:

- ip add 5.5.22.2 255.255.255.252

- no shut

- exit

- hosts: R4

tasks:

- name: Configuracion de interface R4-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.6 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R4-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.9 255.255.255.252

- no shut

- exit

- hosts: R5

tasks:

- name: Configuracion de interface R5-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.18 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R5-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.10 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R5-s1/1

ios\_config:

parents: "interface s1/1"

lines:

- ip add 5.5.22.21 255.255.255.252

- no shut

- exit

- hosts: R2

tasks:

- name: Configuracion de interface R2-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.17 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R2-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.14 255.255.255.252

- no shut

- exit

- hosts: ISP

tasks:

- name: Configuración de EIGRP-22

ios\_config:

parents: "router eigrp 22"

lines:

- network 0.0.0.0 0.0.0.0

- do wr

- hosts: ISP

tasks:

- name: Grabar configuraciones

ios\_config:

lines:

- do wr

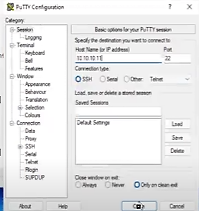
------

ansible-playbook -i hosts MIC-2.yaml

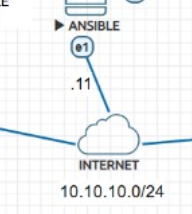
Con este comando, probamos finalmente si todo ha salido bien

Una vez cambiado las direcciones como indica nuestra topología, lo mejor será mandar este archivo a nuestra maquina Windows para copiar y pegar. (Aquí nota, lo mejor es mandar este archivo o txt antes del examen para no tener problemas, y simplemente ir cambiando lo que es este paso en la maquina Windows server).

Ya con el archivo en nuestra maquina Windows server, abrimos putty.



Ingresamos la dirección de internet de nuestra topología:



Si al aceptar aparece un mensaje, simplemente le damos en aceptar.

Login: root

Password: Test123

Si sale todo bien, habremos entrado conexion a Ansible.

vi ~/.ssh/config

En la terminal ingresamos ese comando y se abre este archivo config

KexAlgorithms +diffie-hellman-group14-sha1

Ciphers +aes128-cbc

PubkeyAcceptedAlgorithms +ssh-rsa

HostkeyAlgorithms +ssh-rsa

Se ingresan estas líneas para forzar o habilitar algoritmos antiguos que en versiones recientes de OpenSSH vienen deshabilitados por seguridad. Recordar aquí, pulsar la tecla I para insertar y ESC :wq! para guardar los cambios.

**------Prueba SSH (R3)-----**

ssh -l ESPE22 172.16.22.33

Esa letra después del ssh es una ele minúscula. Este comando lo podemos probar en la máquina Ubuntu, ósea ANSIBLE, ya no en putty. Si no se sabe de donde saca esa dirección IP, es simplemente de aquí. De lo que cambiamos antes. Esa dirección es de la red de programabilidad, lo que cambia es el ultimo octeto para su router.

[R3]

172.16.22.33

[R4]

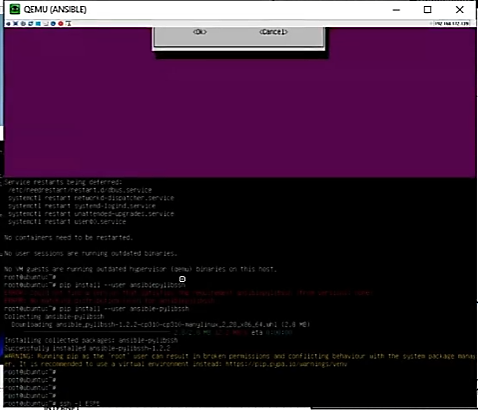
172.16.22.44

[R5]

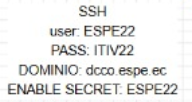
172.16.22.55

[R2]

172.16.22.22



Aquí nos pedirá las credenciales, en este caso USER y PASS.



Si en la terminal nos indica “R3#:” es porque estamos dentro del router 3 y confirmando que si funciona el ssh. Para probar con los demás routers, escribimos “exit” y hacemos el mismo para otra dirección.

Una vez comprobado el ssh de cada router, regresamos a PUTTY e ingresamos el siguiente comando:

cd /etc/ansible

y después de eso, este siguiente comando:

vi hosts

Una vez dentro de ese archivo, va a encontrar muchas líneas, deberá bajar hasta el final y pegar todo esto: (Como se menciona antes, ya debería tener cambiado esto a la topología que le haya tocado para pegar).

[R3]

172.16.22.33

[R4]

172.16.22.44

[R5]

172.16.22.55

[R2]

172.16.22.22

[ISP:children]

R3

R4

R5

R2

[ISP:vars]

ansible\_network\_os=cisco.ios.ios

ansible\_connection=ansible.netcommon.network\_cli

ansible\_user=ESPE22

ansible\_password=ITIV22

Entonces, pulsar I para insertar y :wq! para guardar los cambios. Recordar que para pegar solo le damos click derecho.

Con esto, podemos comprobar si esta todo correcto con este comando el archivo de inventario:

ansible-inventory --list -y



Deberia mostrar los routers ingresados. Si sale todo eso, es que esta bien.

Dentro de esta misma carpeta cd /etc/ansible ingresamos el siguiente comando para crear el archivo yaml.

vi MIC-2.yaml

El nombre de “MIC-2” puede cambiar, ósea puedes poner 3 o 4 si anteriormente ya se creo el archivo.

Entonces dentro del archivo, pegamos lo siguiente:

---

- hosts: R3

tasks:

- name: Configuracion de interface R3-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.5 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R3-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.13 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R3-s1/0

ios\_config:

parents: "interface s1/0"

lines:

- ip add 5.5.22.2 255.255.255.252

- no shut

- exit

- hosts: R4

tasks:

- name: Configuracion de interface R4-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.6 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R4-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.9 255.255.255.252

- no shut

- exit

- hosts: R5

tasks:

- name: Configuracion de interface R5-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.18 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R5-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.10 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R5-s1/1

ios\_config:

parents: "interface s1/1"

lines:

- ip add 5.5.22.21 255.255.255.252

- no shut

- exit

- hosts: R2

tasks:

- name: Configuracion de interface R2-e0/1

ios\_config:

parents: "interface e0/1"

lines:

- ip add 5.5.22.17 255.255.255.252

- no shut

- exit

- name: Configuracion de interface R2-e0/0

ios\_config:

parents: "interface e0/0"

lines:

- ip add 5.5.22.14 255.255.255.252

- no shut

- exit

- hosts: ISP

tasks:

- name: Configuración de EIGRP-22

ios\_config:

parents: "router eigrp 22"

lines:

- network 0.0.0.0 0.0.0.0

- do wr

- hosts: ISP

tasks:

- name: Grabar configuraciones

ios\_config:

lines:

- do wr

Aquí recordar que al inicio se ingresa con todo y líneas ---

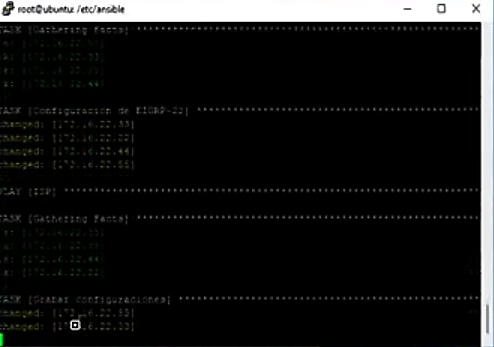
Entonces guardamos los cambios. Podemos revisar todo lo ingresado o el archivo yaml con

cat MIC-2.yaml

Si todo esta en orden, ingresamos el siguiente comando para verificar que todo esta correcto:

ansible-playbook -i hosts MIC-2.yaml

Aquí comenzará una ejecución y debe mostrar esto:

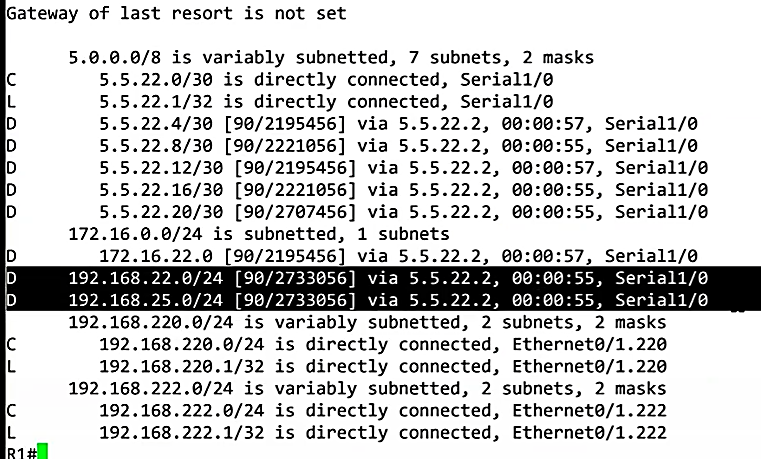


Debe mostrar las configuraciones de los router R3,R4,R2,R5

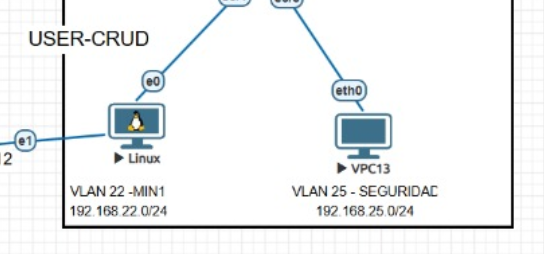
Con esto hecho, abrimos el router1 e ingresamos:

enable

show ip router



Debe mostrar la dirección ip de la sucursal

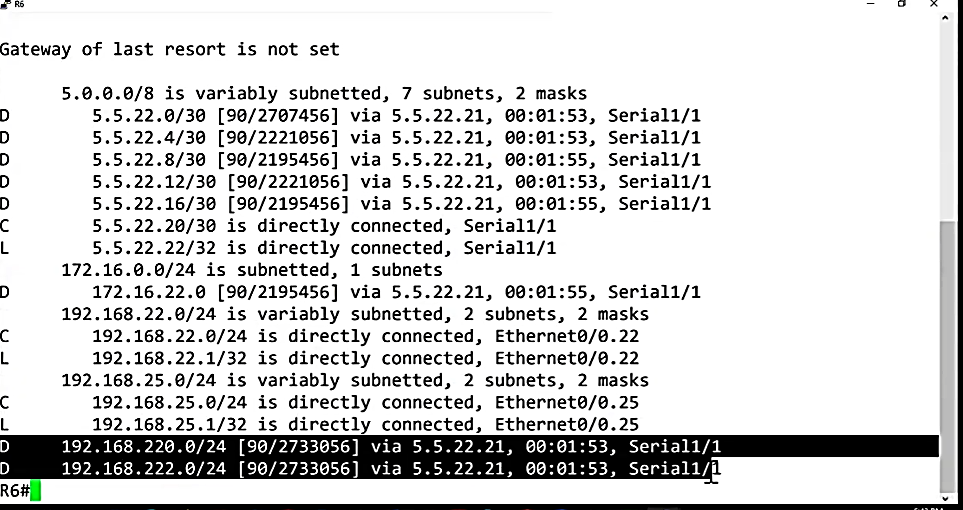


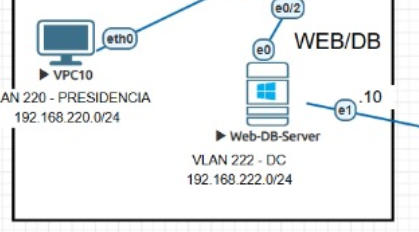
Lo mismo para el router6, entramos e ingresamos:

enable

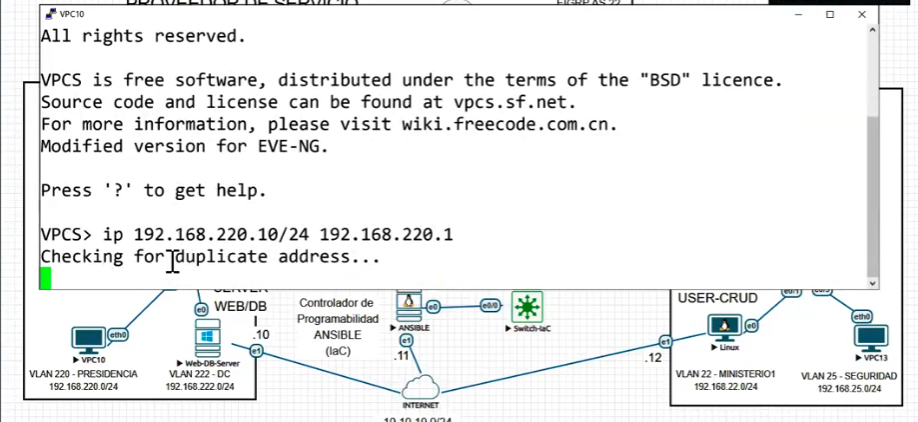
show ip route

Debe mostrar la dirección ip del otro lado



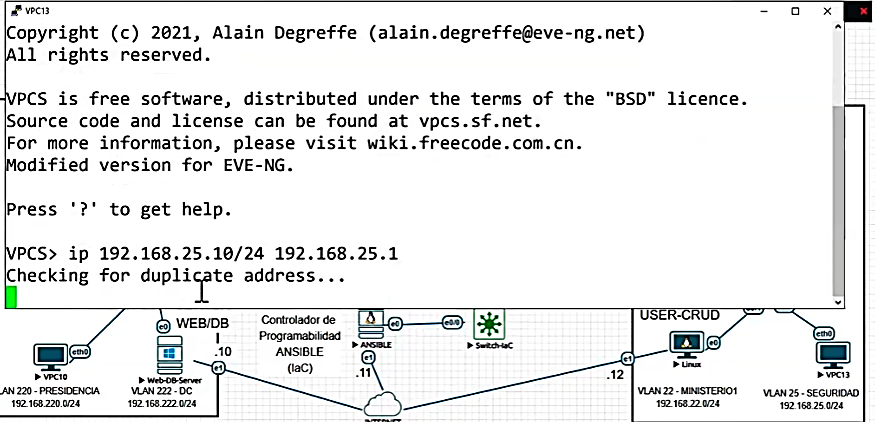


Con esto comprobado, vamos a abrir la VPC10 e ingresamos el siguiente comando:



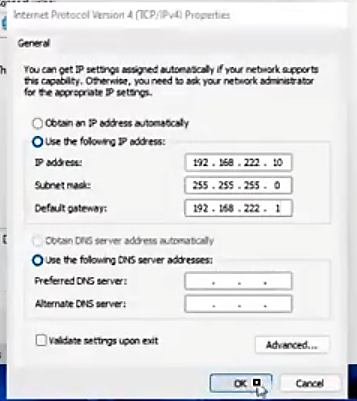
La primera dirección se ingresa de la pc con el .10 de la salida de la web/db. La ultima dirección es el Gateway de la vpc10. Con esto hemos configurado el direccionamiento IP. Finalmente escribimos “sabe” y queda guardado los cambios.

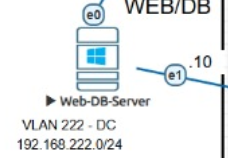
Ahora lo mismo para la VPC13.



La primera dirección se ingresa de la pc con el .10 de la salida de la web/db. La ultima dirección es el Gateway de la vpc13. Con esto hemos configurado el direccionamiento IP. Finalmente escribimos “sabe” y queda guardado los cambios.

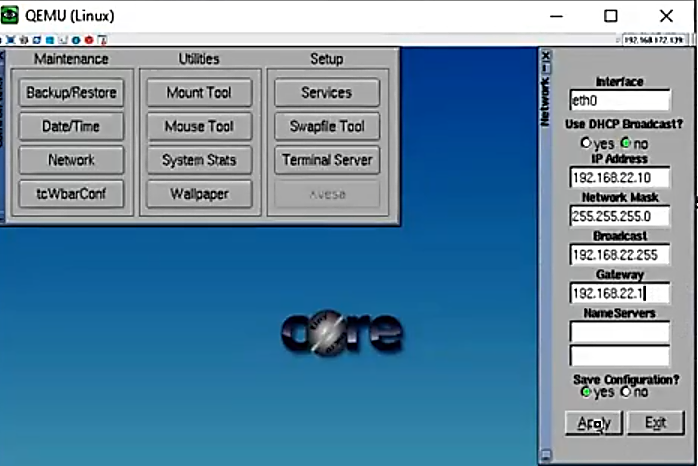
Con esto hecho, nos vamos a la maquina Windows Server y desactivamos la tarjeta de red 2 de internet e ingresamos en la tarjeta 1 de red la dirección ip siguiente:

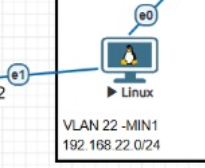




Agregamos la dirección ip de DC.

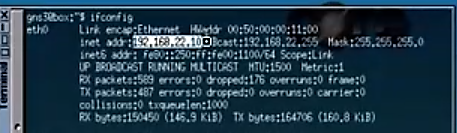
Con eso hecho en Windows Server, pasamos a la maquina de Tinicore Linux.





Ingresamos la direccion ip de MIN1.

Abrimos ahí mismo un terminal:



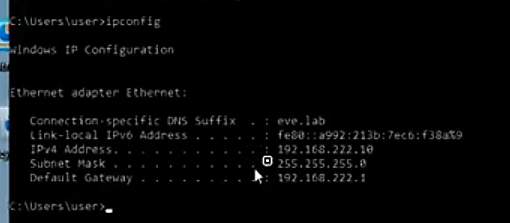
Y revisamos que este bien configurado el eth0.

Probamos a hacer ping a nuestro Gateway. 192.168.22.1

Probamos hacer ping al Gateway del web/bd. 192.168.222.1

Probamos hacer ping al Windows. 192.168.222.10.

Hecho esto, pasamos al Windows Server para probar el ping.



Revisamos nuestra dirección ip ingresado en la tarjeta de red.

Probamos haciendo ping a la 192.168.22.10

En tinicore, Probamos hacer ping al Windows. 192.168.222.10.

Con esto se habrá terminado todo sobre TECNOLOGIAS EMERGENTES.