

# PUFF.KFT

TRIPLE A TEAM

# A bemutató tartalma:

- I. Történet
- II. Feladatfelosztás egymás közt
- III. Saját részem a projektben
- IV. Szerverek működése
- V. Csapattársaim munkája a projektben
- VI. Tesztelések
- VII. Videó

# *I. Történet*

# A Puff.kft csomag szállító cég három telephelyből áll.

- **Irodaépület:**

A cég központi irodája egy önálló épületben található, ahol 20 munkavállaló dolgozik. A dolgozók között van egy vezető, egy HR-munkatárs, egy reklámmenedzser, egy pénzügyi szakember, egy könyvelő, egy jogász, valamint egy logisztikai mérnök négy asszisztenssel. Az ügyfélszolgálati részlegen jelenleg 6 fő lát el feladatokat, továbbá 3 informatikus biztosítja a technikai háttérrel. A hálózat kialakítása legalább 20 vagy több fő kiszolgálására is képes.

- **Raktár:**

A raktár szintén külön épület, ahol több raktáros dolgozik. Egy központi számítógép is elérhető, amely lehetővé teszi, hogy a dolgozók rendeléseket, üzeneteket és egyéb adatokat tekintsenek meg.

- **Garázs:**

A garázs egy másik önálló épület, ahol két dolgozó felelős a járművek tervezett csomagelosztásáért, azaz nyomon követik, hogy milyen csomagok kerülnek be vagy ki.

## *II. Feladatfelosztás egymás közt*

# Ki - mit csinált?

- **Puhl Alexandra(Én):** Én végeztem el a szerverek telepítését és konfigurációját, mert nekem bírta el a leginkább a számítógémem a sok virtuális gép futtatását egyszerre, illetve én ehhez értem a leginkább, de Attila is hozzájárult, amikor hibába ütköztem. Egy Linux és három Windows szervert hoztam létre, amelyből az egyik egy Kliens gép plusz egy virtuális routert(Mikrotik). A dokumentáció nagy részét én készítettem el, a többiek munkájának a felhasználásával, de Alessia is aktívan részt vett a szerkesztésben. Videóhoz az anyagot én készítettem el, de Alessia vágta össze és szerkesztette meg teljességében.
- **Zamarco Alessia:** Alessia csinálta a videót és a hálózatba is besegített Attilának, illetve a weblapot és a hálózat programozást is ő oldotta meg. A hálózatban az ő felelőssége volt a Garázs és a Raktár. Dokumentáció szerkesztése.
- **Szigunov Attila:** Attila érdeme nagyrészt a hálózat hibátlan működése, mert ő a legjobb a hálózatban. A topológiát közösen tervezte meg Alessiával. Két részre osztották a hálózatot, ő csinálta az Iroda konfigurációját és az Internetét. A topológia később lesz látható a bemutató során.

# Feladatok felosztva (röviden):

## ALESSIA

- Topológia
- Videó
- Weblap
- Hálózat(Raktár, Garázs)
- Hálózat programozás

## ATTILA

- Topológia
- Hálózat(Iroda, Internet)
- Szerverek(Besegített ahol épp kellett)

## Hogy ment a közös munka?

- Trelloba a kezdetek óta vezettük a projekt haladását, mindenki aktívan töltögette fel az anyagokat, fájlokat. Sokszor megbeszéltünk egy napot és össze tudtunk ülni több órákra nálam, hogy dolgozhassunk a vizsgafeladatokon és segítsünk egymásnak. Általában megtudtunk mindent beszélni, ha probléma volt segítettünk egymásnak kivétel nélkül. Mindenkinek fontos volt a véleménye, így bárkinek volt észrevétele egy adott dologgal azt meghallgattuk és figyelembe vettük még akkor is, ha nem feltétlen a saját feladatunk volt.



### *III. Feladatom a projektben*

## Feladatom a projektben:

- Active Directory (címtár)
- DHCP
- DNS
- HTTP/HTTPS
- Automatizált mentés
- Fájl –és nyomtató megosztás
- Automatizált szoftvertelepítés a Klingsre
- Dokumentáció elkészítése

## *IV. Szerverek*

## Szerverek felépítése – VirtualBox szoftverben:

- Windows Server PDC (Iroda)
- Windows Server BDC (Raktár)
- Windows Kliens (Tesztelés)
- Linux Debian 12 (Garázs)
- (Mikrotik Router)

## Windows Server PDC működése:

- Az irodai szerver felelős az IPv4 és az IPv6 címek DHCP kiosztásáért. Az AD (Active Directory) segítségével tárolja és kezeli az iroda hálózatán belüli összes felhasználót (23db). Emellett ez a szerver végzi az automatikus adatmentéseket is, amit a BDC szerverre küld át.

## Windows Server BDC működése:

- A raktárban található szerver egyetlen célja az, hogy az irodai (PDC) szerver meghibásodása esetén átvegye annak feladatait, így biztosítva a rendszer zavartalan működését.

## Windows 10 Kliens működése:

- Ezen a virtuális gépen teszteltem le minden szükséges beállítást. Ilyen például az, hogy a fő szerver leállása esetén is zavartalanul működik tovább minden, azaz, hogy megkapja-e a megfelelő DNS címet a kliens és eléri-e az internetet. Saját meghajtó felcsatolása egy az AD-ban létrehozott felhasználónak. Tanúsítvánnyal rendelkező weboldal elérése az interneten.

## Linux Debian 12 Bookworm működése:

- A garázsban elhelyezett szerver biztosítja a vállalat weboldalának működését, valamint a fájl -és nyomtató megosztást az iroda és a garázs hálózatában. Mindemellett FTP szolgáltatást is nyújt.



## MIKROTIK ROUTER:

A virtuális router tette lehetővé, hogy internetet kapjon minden szervergép. Fix ethernet3 cím lett beállítva, hogy gond nélkül működjön. A tűzfal beállítások a következőképp lettek megadva:

```
mikrotik_router_puff [Running] - Oracle VirtualBox
File Machine View Input Devices Help

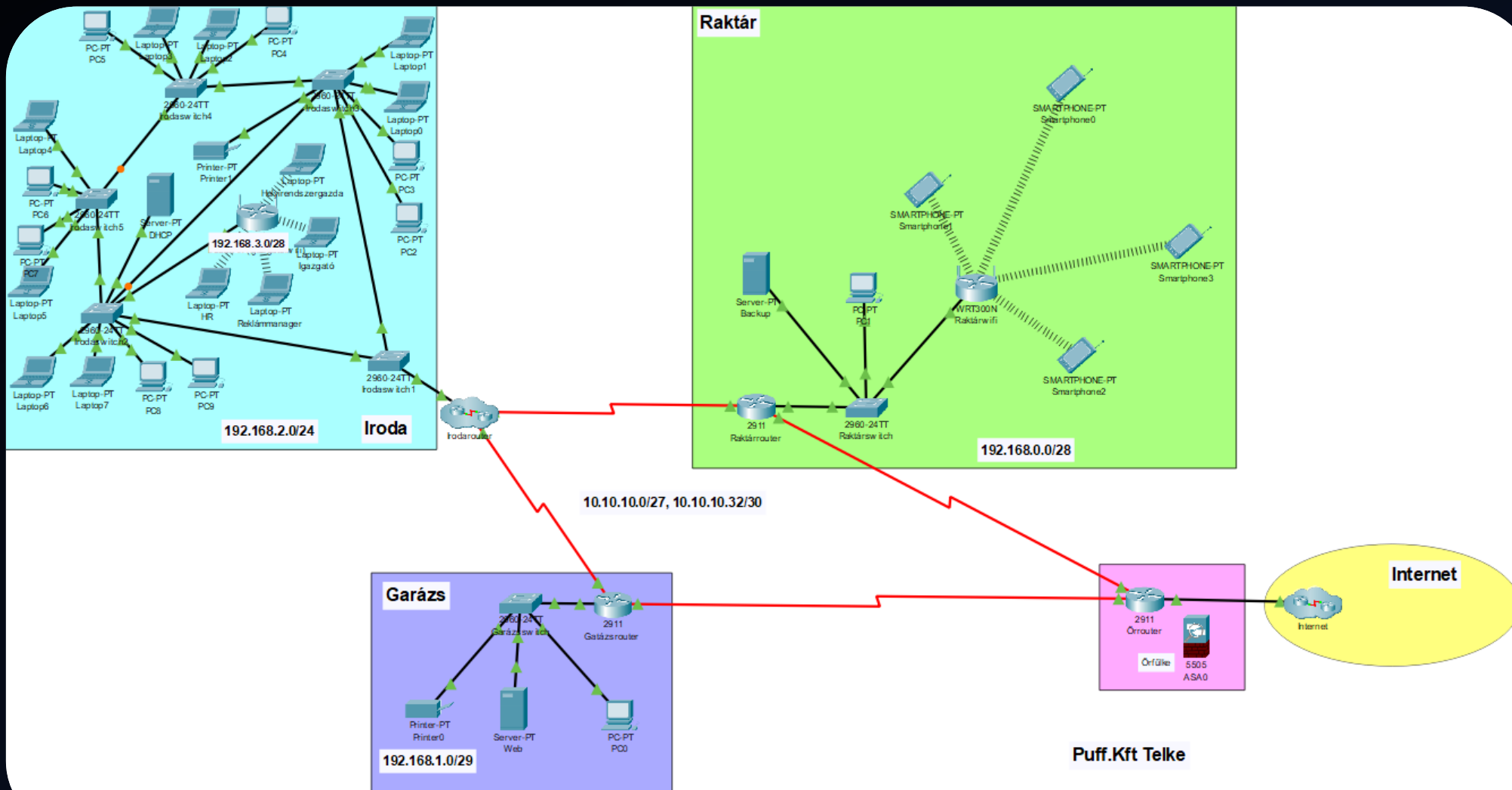
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > ip/address/p
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS NETWORK INTERFACE
0 D 10.0.2.15/24 10.0.2.0 ether1
1 192.168.0.0/22 192.168.0.0 ether2
2 192.168.56.99/24 192.168.56.0 ether3
[admin@MikroTik] > ip/firewall/nat/p
Flags: X - disabled, I - invalid; D - dynamic
0 chain=srcnat action=masquerade out-interface=ether1
1 chain=dstnat action=dst-nat to-addresses=192.168.1.2 to-ports=22
  protocol=tcp in-interface=ether3 dst-port=2222 log=no log-prefix=""
2 chain=dstnat action=dst-nat to-addresses=192.168.0.2 to-ports=3389
  protocol=tcp in-interface=ether3 dst-port=55000 log=no log-prefix=""

  chain=dstnat action=dst-nat to-addresses=192.168.2.130 to-ports=3389
  protocol=tcp in-interface=ether3 dst-port=50000
[admin@MikroTik] >
```

# *V. Csapattársaim munkája a projektben*

# A PUFF.KFT hálózati topológiája:

A biztonság érdekében már a belső hálózaton is NAT-olást alkalmazunk. A telephelyek közötti kommunikáció VPN-en keresztül történik, míg a forgalomirányításhoz OSPF protokollt használunk.



# IP címek:

## Az iroda hálózatában az alábbi IP-címek kerülnek kiosztásra:

- ❑ 192.168.2.0/24 az általános irodai hálózat számára
- ❑ 192.168.3.0/29 a Wi-Fi router számára

## A hálózat VLAN szegmensei az alábbiak:

- ❑ V10: 192.168.2.0/27 / IPv6: FC10::0/64
- ❑ V20: 192.168.2.32/27 / IPv6: FC20::0/64
- ❑ V30: 192.168.2.64/27 / IPv6: FC30::0/64
- ❑ V40: 192.168.2.96/27 / IPv6: FC40::0/64
- ❑ V99: 192.168.2.128/29 / IPv6: FC99::0/64

## A routerek közötti címek:

- ❑ R4,1 - R4,2: 192.168.2.136/30
- ❑ R4,1 - R4,3: 192.168.2.140/30
- ❑ R4,2 - R4,4: 192.168.2.144/30
- ❑ R4,3 - R4,4: 192.168.2.148/30

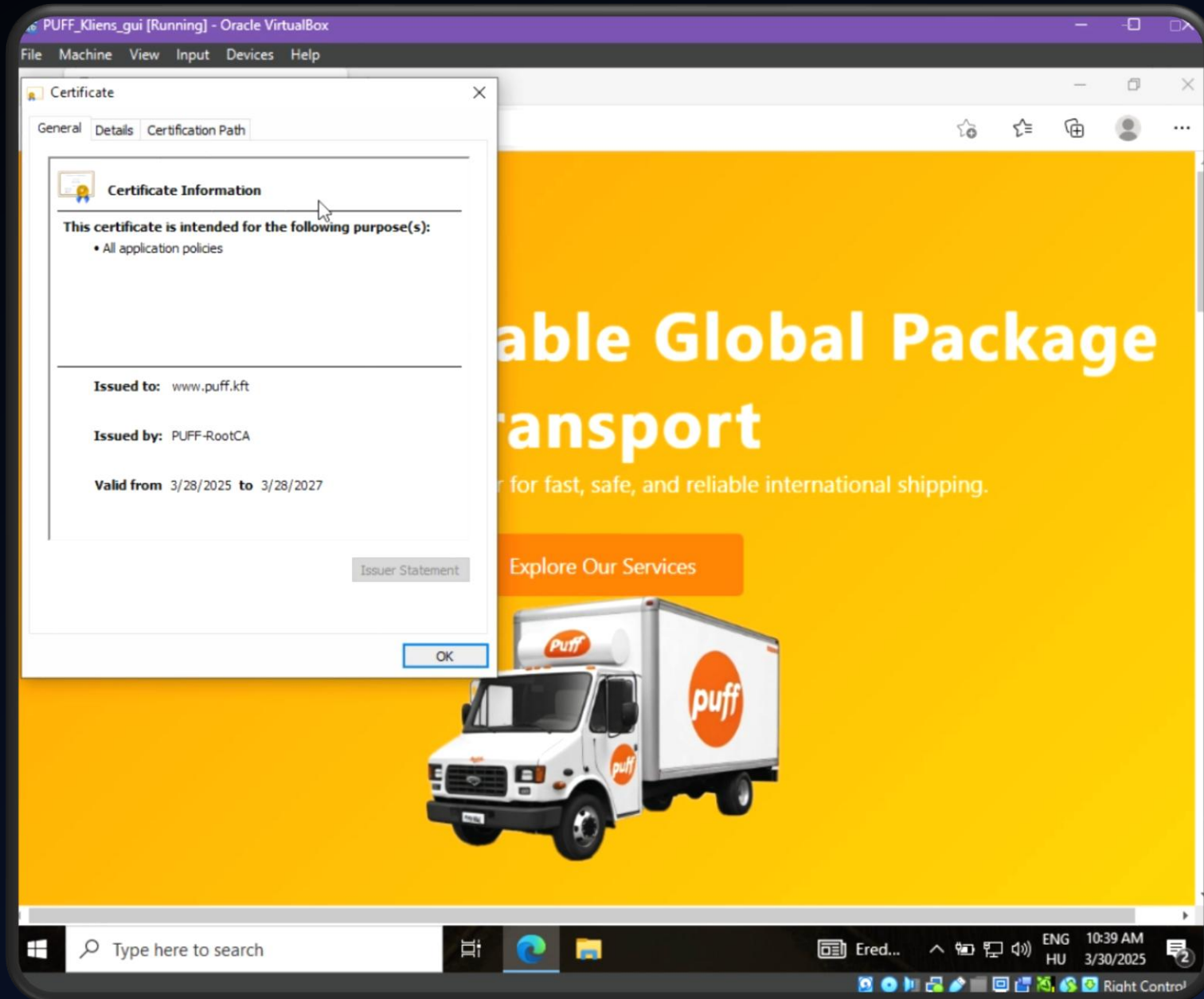
## A külső hálózati címkiosztás:

- ❑ Internet - R1: 10.10.10.0/30
- ❑ R1 - R2: 10.10.10.4/30
- ❑ R1 - R3: 10.10.10.8/30
- ❑ R2 - R4,1: 10.10.10.16/30
- ❑ R3 - R4,1: 10.10.10.12/30

ALESSIA

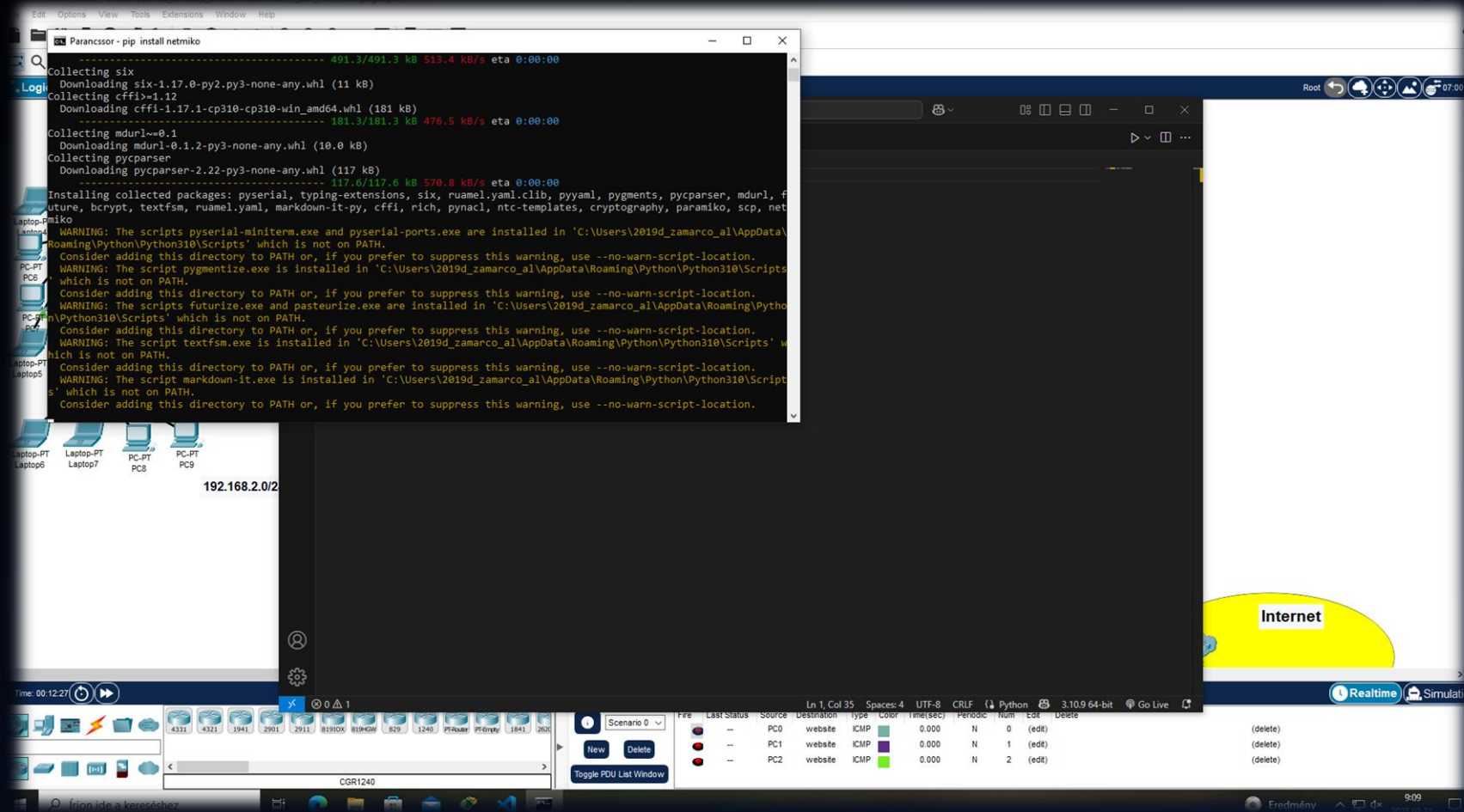
### Weboldal:

Reszponzív a weblap, tehát ahogy változik az oldal mérete úgy alkalmazkodik hozzá a weblap is. Vannak működő gyorsgombok (pl. „Explore Our Services”), aminek segítségével odaugrik az oldal az adott tartalomhoz. E-mail küldési lehetőség is van a cég számára.





# Netmiko telepítése a Python-hoz:



## Mire szolgál a Python kód?

- Sorban kapcsolódik több switch-hez
- Lefuttat 3 parancsot (sh running-config | i hostname, sh ver, sh vlan brief)
- Ezen parancsok kimenetét kiírja egy fájlba
- Fájlokat a z:\vizsgaremek\{helyi\_felhasznalo}\Python\ mappába menti
- Minden switch-hez külön fájl jön létre Config\_{switch} néven
- A kód a Netmiko könyvtárat használja SSH kapcsolat létrehozására.

```
File Edit Selection View Go Run Terminal Help
vian.py 5
Z: > vizsgaremek > vian.py > ...
1 from netmiko import ConnectHandler
2 import subprocess
3
4 List_of_Switches= []
5 Number_of_Switches= int(input("Mennyi switch?: "))
6
7 for switch in range (1,Number_of_Switches +1):
8     Ask= input("Add meg a switchnek a számát" + str(switch) + ":")
9     List_of_Switches.append(Ask)
10
11 User = input("Mi a felhasználónév?: ")
12 Pass = input("Mi a jelszó?: ")
13 Enable_Pass = input("Mi a titkosított jelszó?: ")
14
15
16
17
18
19 def Username ():
20     Position_Counter = 0
21     Result = subprocess.getoutput ("kivagyokén")
22     for words in Result:
23         if words=="\\":
24             Position = Position_Counter
25             break
26     Position_Counter +=1
27     return Result[Position +1: ]
28
29 Local_Computer_Username = Username ()
30
31 for switches in List_of_Switches:
32
33     Network_Device= {"host": switches,
34                     "username":User,
35                     "password":Pass,
36                     "secret":Enable_Pass,
37
38
39
40
41 Connect_To_Device=ConnectHandler(**Network_Device)
42 Connect_To_Device.enable()
43
44 List_Of_Commands=["sh running-config | i hostname","sh ver",
45                  "sh vlan brief"]
46
47 for command in List_Of_Commands:
48     output =Connect_To_Device.send_command()
```

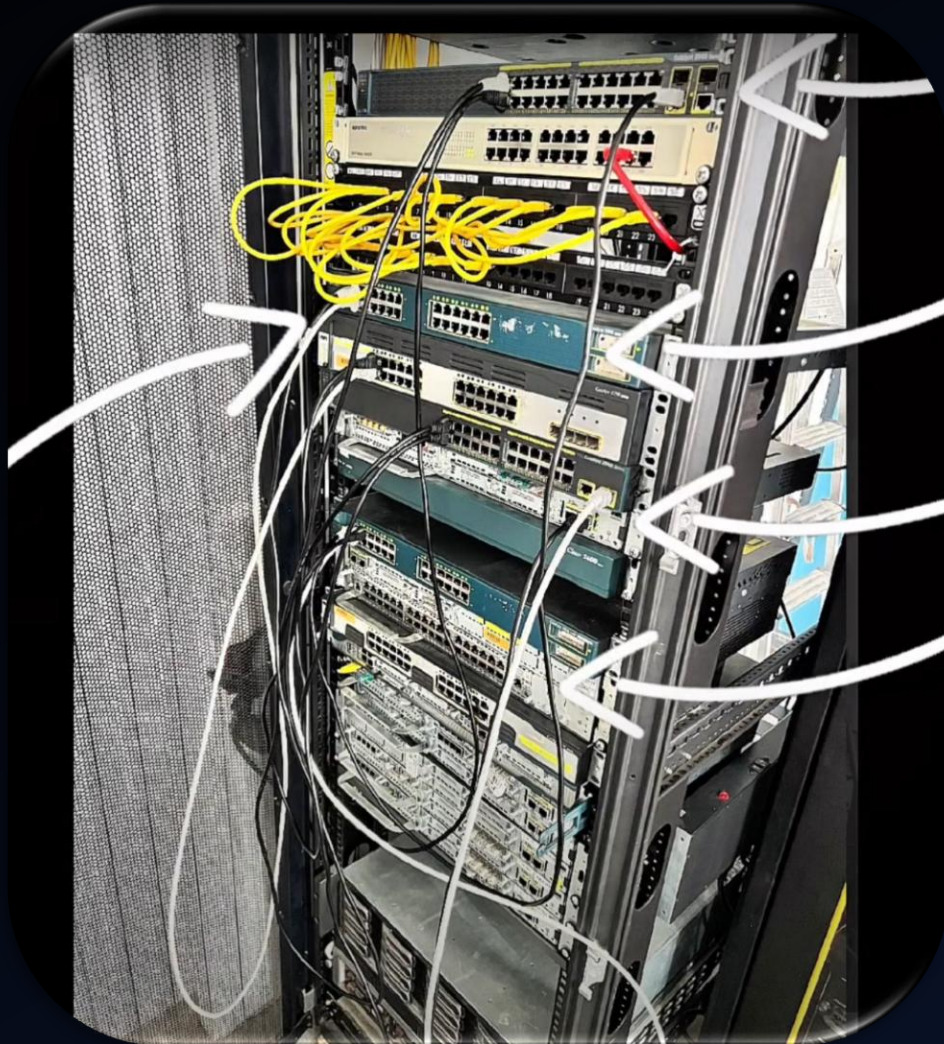
```
File Edit Selection View Go Run Terminal Help
vian.py 5
Z: > vizsgaremek > vian.py > ...
19 def Username ():
20     if words=="\\":
21         Position = Position_Counter
22         break
23     Position_Counter +=1
24     return Result[Position +1: ]
25
26 Local_Computer_Username = Username ()
27
28 for switches in List_of_Switches:
29
30     Network_Device= {"host": switches,
31                     "username":User,
32                     "password":Pass,
33                     "secret":Enable_Pass,
34
35
36
37
38
39
40
41 Connect_To_Device=ConnectHandler(**Network_Device)
42 Connect_To_Device.enable()
43
44 List_Of_Commands=["sh running-config | i hostname","sh ver",
45                  "sh vlan brief"]
46
47 for command in List_Of_Commands:
48     output =Connect_To_Device.send_command()
49
50     with open("z:\\vizsgaremek\\" +Local_Computer_Username+ "\\Python\\Config_"+switches+""):
51         f.write("\n")
52         f.write(switches + "#" + command)
53         f.write("\n")
54         f.write(Connect_To_Device.send_command(command))
55         f.write("\n")
56     with open("z:\\vizsgaremek\\" +Local_Computer_Username+ "\\Python\\Config_"+switches+"", "a") as f:
57         f.write(switches + "#" )
58         f.write("\n"*3)
59         f.write("\nEnd of this device/End of this device/End of this device/End of this device"*4)
60         f.write("\n"*3)
61
62 print("\nCompleted")
63
64
65
66
```

Programozott  
hálózatkonfiguráció:

# *VI. Tesztelés*



SSH eléréssel elvégeztük a beállításokat az 5 Switchen és 1 Routeren, mivel fizikálisan nem volt szerver és a videóban a router vette fel a dhcp kiosztó feladatot:



```
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 255.255.240.0

Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix . . : 
Link-local IPv6 Address . . . . . : fe80::d20a:6eb1:6e4:a731%17
IPv4 Address. . . . . : 192.168.2.97
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.2.97

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . . : 
Link-local IPv6 Address . . . . . : fe80::daa:2e52:fe6b:79cc%2
IPv4 Address. . . . . : 192.168.2.98
Subnet Mask . . . . . : 255.255.255.224
Default Gateway . . . . . : 192.168.2.97

Ethernet adapter VMware Network Adapter VMnet1:

Connection-specific DNS Suffix . . : 
Link-local IPv6 Address . . . . . : fe80::b07:6fbc:9efd:964f%16
IPv4 Address. . . . . : 192.168.119.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 

Ethernet adapter VMware Network Adapter VMnet8:

Connection-specific DNS Suffix . . :
```

A tesztelés sikeres  
a kliens megfelelő ip címet kapott: ➡

# Szerver ping tesztelés:

```
Machine View Input Devices Help
Command Prompt
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
approximate round trip times in milli-seconds:
    Minimum = 187ms, Maximum = 265ms, Average = 229ms

Z:\>ping 192.168.2.130

Pinging 192.168.2.130 with 32 bytes of data:
Reply from 192.168.2.130: bytes=32 time=1ms TTL=128
Reply from 192.168.2.130: bytes=32 time<1ms TTL=128
Reply from 192.168.2.130: bytes=32 time=1ms TTL=128
Reply from 192.168.2.130: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.2.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

Z:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64
Reply from 192.168.1.2: bytes=32 time<1ms TTL=64
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64
Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

Z:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=2ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

Z:\>
```

```
Machine View Input Devices Help
Command Prompt
Microsoft Corporation. All rights reserved.

Z:\>ipconfig /all

Windows IP Configuration

Host Name . . . . . : winclient
Primary Dns Suffix . . . . . : puff.kft
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : puff.kft

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . . : 
    Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
    Physical Address. . . . . : 08-00-27-F1-75-C9
    DHCP Enabled. . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::35b1:ead:e145:27a1%4(Preferred)
    IPv4 Address. . . . . : 192.168.2.2(Preferred)
    Subnet Mask . . . . . : 255.255.252.0
    Default Gateway . . . . . : 192.168.0.1
    DHCPv6 IAID . . . . . : 101187623
    DHCPv6 Client DUID. . . . . : 00-01-00-01-2F-78-38-49-08-00-27-F1-75-C9
    DNS Servers . . . . . : 192.168.2.130
                           192.168.0.2
    NetBIOS over Tcpip. . . . . : Enabled

Z:\>ping cisco.com

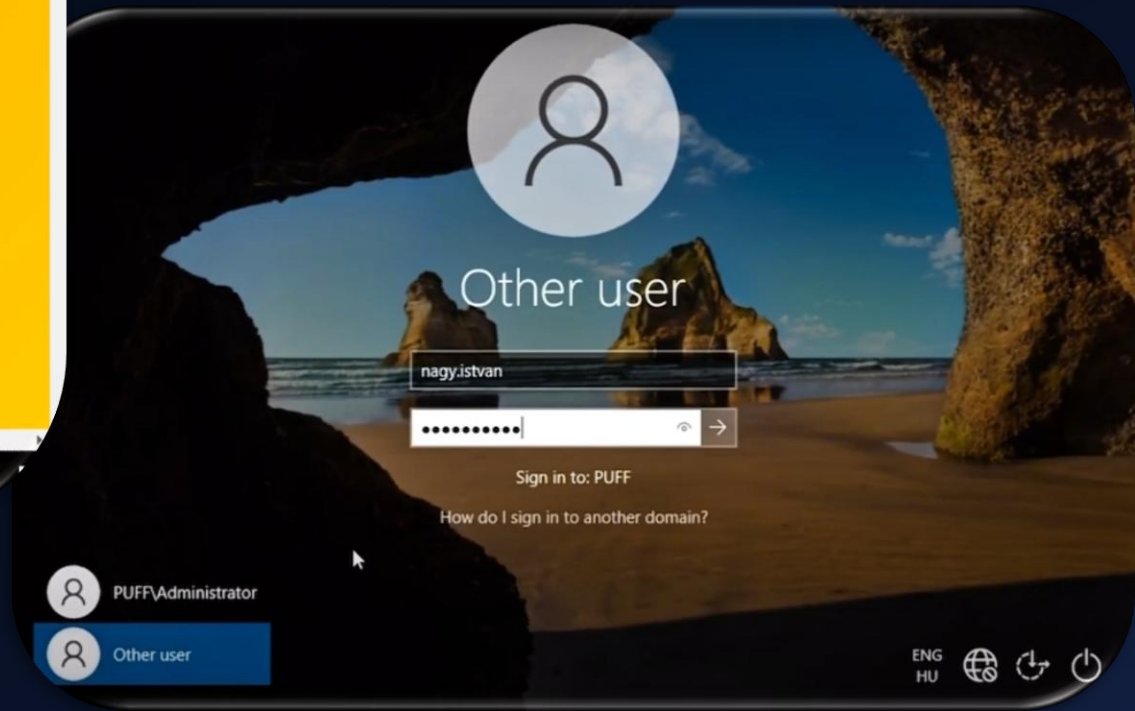
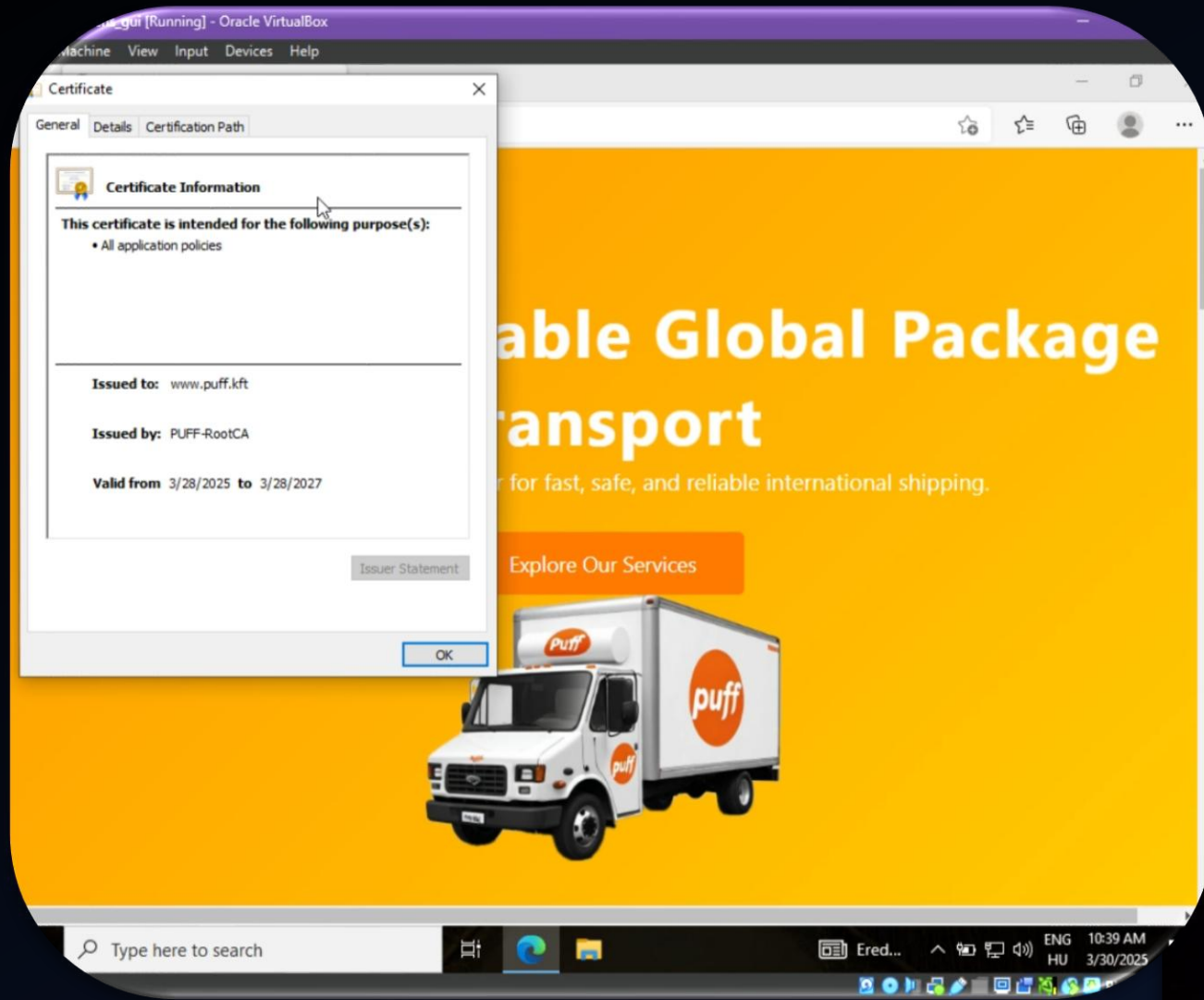
Pinging cisco.com [72.163.4.185] with 32 bytes of data:
Reply from 72.163.4.185: bytes=32 time=236ms TTL=254
Reply from 72.163.4.185: bytes=32 time=187ms TTL=254
Reply from 72.163.4.185: bytes=32 time=265ms TTL=254
Reply from 72.163.4.185: bytes=32 time=230ms TTL=254

Ping statistics for 72.163.4.185:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 187ms, Maximum = 265ms, Average = 229ms

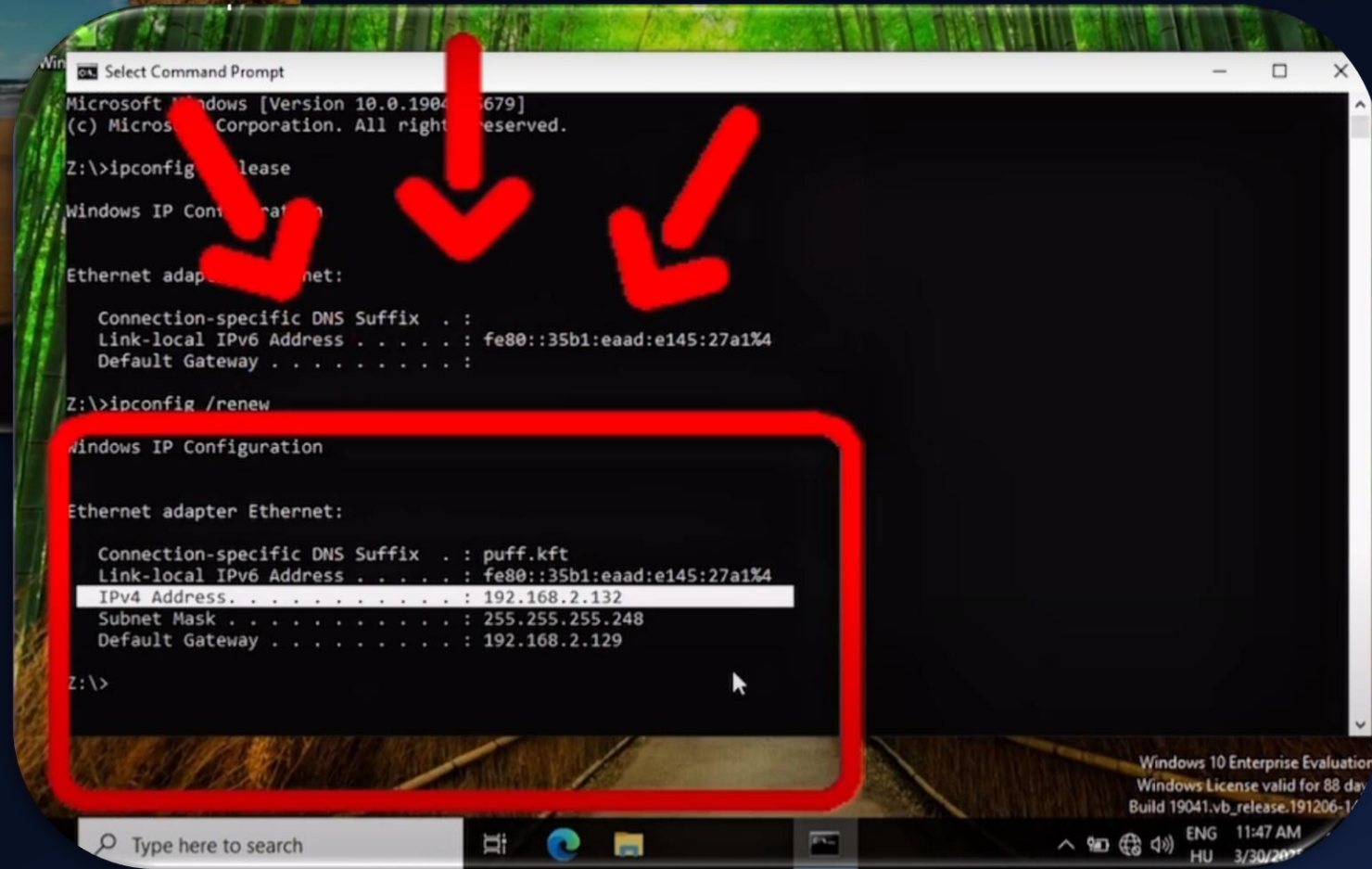
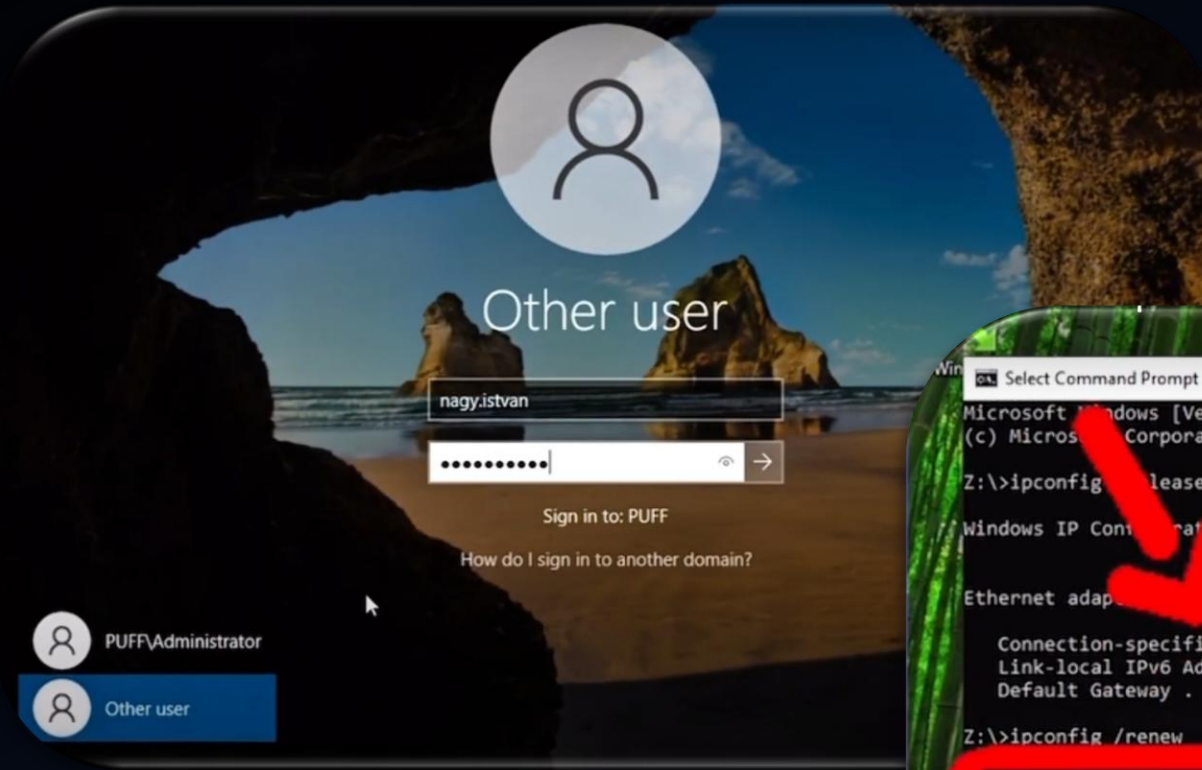
Z:\>ping 192.1
```



# Szerver HTTPS elérés:



# Szerver DHCP elérés:



## *VII. Videó*



The background is a solid dark blue. On the left side, there are several vertical and slightly angled teal lines of varying thicknesses, creating a sense of depth and structure. On the bottom right, there are several diagonal teal lines that extend towards the corner, also in varying thicknesses.

KÖSZÖNÖM MEGTISZTELŐ  
FIGYELMÜKET!

*PUHL ALEXANDRA*

# Témára visszaugrás, ha szükséges:

<p>PUFF.KFT TRIPLE A TEAM</p>	<p>I. Történet</p>	<p>II. Feladatfelosztás egymás közt</p>
<p>III. Feladatom a projektben</p>	<p>IV. Szerverek</p>	<p>V. Csapattársaim munkája a projektben</p>
<p>A PUFF.KFT hálózati topológiája:</p> 	<p>VI. Tesztelés</p>	<p>VII. Videó</p>