**Network Automation & Programmability Using Python 3**

**(CCNA 200-301)**

**Lab Setup**

Open Command Prompt as Administrator

Change Directory to GNS3 Directory (Where you have installed GNS3)

* cd C:\Program Files\GNS3

Run following command to change the service config

* sc config npf start= auto

Now Run the Loopback Service Manager

* Loopback-manager.cmd

Choose option 4 for remove all loopback interfaces if installed before.

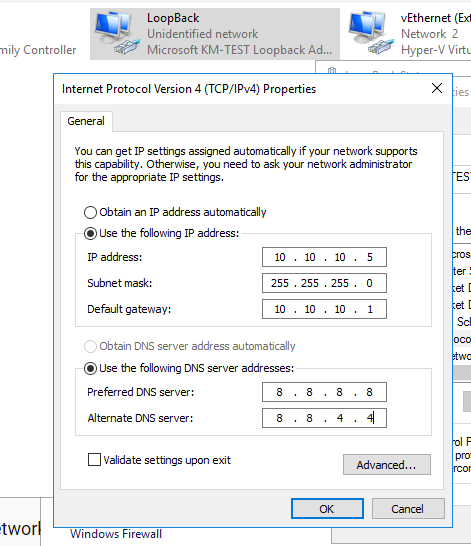
Choose option 2 for install a new loopback interface

Choose option 6 for quit

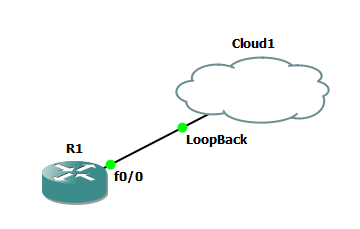
Now Reboot your PC.

Rename the newly installed loopback interface (You can choose any name)

Configure IP Address in Loopback Interface

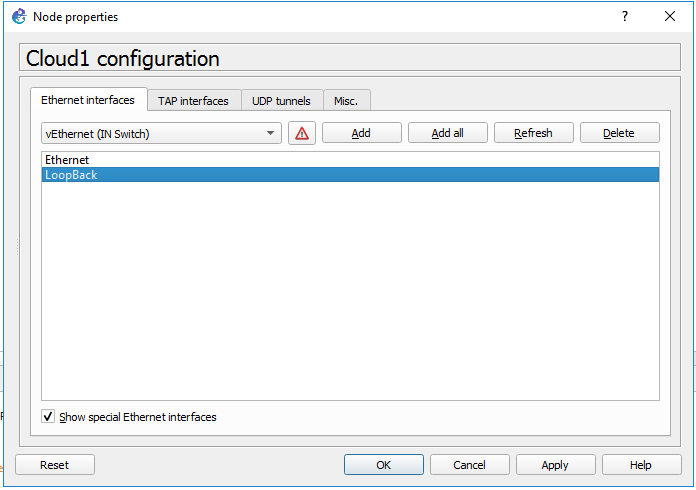


Now go to GNS3 and draw a simple topology like bellow



Open Cloud1 Configuration

Check on Show Special Ethernet Interfaces and Select LoopBack Interface then Add



Now Open Router Console:

R1(config)#interface fastEthernet 0/0

R1(config-if)#ip address 10.10.10.6 255.255.255.0

R1(config-if)#no shutdown

R1#ping 10.10.10.5

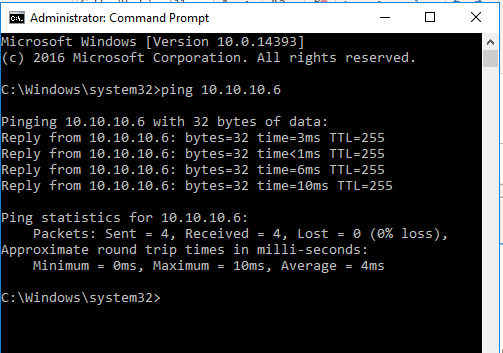
R1(config)#enable password 1234

R1(config)#username admin password 1234

R1(config)#line vty 0 4

R1(config-line)#login local

Now Ping from your Host PC to GNS3 Router



Download and Install Python

Open Py Run Command

**Learn Variables:**

Python is completely object oriented, and not "statically typed". You do not need to declare variables before using them, or declare their type. Every variable in Python is an object. Variables are just names which assign a value.

This tutorial will go over a few basic types of variables.

For example: you are working on a company. There are so many employee in the company.

Emp = Employee

In Py Shell Type:

>>> Emp1 = "John"

Now if you type Emp1, it will answer ‘John’

>>> Emp1

'John'

Here Emp1 is variable and John is value

Now input some other employee

>>> Emp2 = "Paul"

>>> Emp3 = "July"

Now See all output:

>>> Emp1

'John'

>>> Emp2

'Paul'

>>> Emp3

'July'

Now we can assign values of multiple variables in a single command:

>>> Emp4, Emp5, Emp6 = "Ruby", "Sharkar", "Shoumy"

>>> Emp1

'John'

>>> Emp2

'Paul'

>>> Emp3

'July'

>>> Emp4

'Ruby'

>>> Emp5

'Sharkar'

>>> Emp6

'Shoumy'

Or you can see output of values of multiple variables in a single command:

>>> Emp1, Emp2, Emp3, Emp4, Emp5, Emp6

('John', 'Paul', 'July', 'Ruby', 'Sharkar', 'Shoumy')

Value of variable is not permanent, You can change it or you can assign same value for multiple Variables.

>>> Emp5=Emp6=Emp7= "Mahmud"

>>> Emp5

'Mahmud'

>>> Emp6

'Mahmud'

>>> Emp7

'Mahmud'

**Learn Print:**

The print function in Python is a function that outputs to your console window whatever you say you want to print out.

At first blush, it might appear that the print function is rather useless for programming, but it is actually one of the most widely used functions in all of python. The reason for this is that it makes for a great debugging tool.

"Debugging" is the term given to the act of finding, removing, and fixing errors and mistakes within code. If something isn't acting right, you can use the print function to print out what is happening in the program. Print is a built-in function in Python.

>>> print ('Hello Friends')

Hello Friends

You can use print function with variables.

>>> print(Emp3)

July

>>> print(Emp3,Emp5,Emp1)

July Mahmud John

**Operators: Integer, String and Slicing**

**Integer:**

>>> age = 50

>>> type(age)

<class 'int'>

>>> age = 50

>>> type(age)

<class 'int'>

>>> age1 = 60

>>> age2 = 90

>>> age1-age2

-30

>>> age2-age1

30

>>> age1+age2

150

>>> age1\*age2

5400

>>> age1/age2

0.6666666666666666

>>> age1%age2

60

**String:**

String will always be in Quotation Mark. That can be single or double quotation.

>>> Name="Kakoli"

>>> type(Name)

<class 'str'>

>>>

>>> FirstName= "Quazi"

>>> MiddleName= "Mahmudul"

>>> LastName= "Huq"

>>> FullName= FirstName +MiddleName +LastName

>>> FullName

'QuaziMahmudulHuq'

>>> FullName= FirstName+" "+MiddleName+" "+LastName

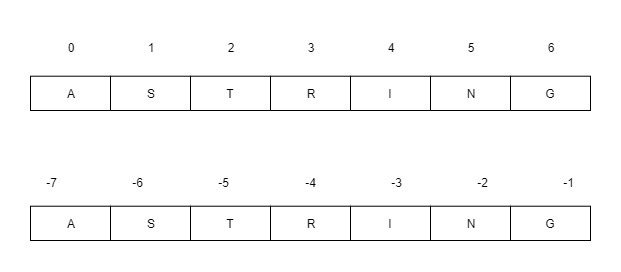
>>> FullName

'Quazi Mahmudul Huq'

**Slicing the String:**

**Python slicing** is about obtaining a sub-string from the given [string](https://www.geeksforgeeks.org/python-strings/) by slicing it respectively from start to end.

**Index tracker for positive and negative index:**  
Python Start index counting from 0. Negative comes into considers when tracking the string in reverse.



>>> FullName

'Quazi Mahmudul Huq'

We want to find 6th Character of given string of FullName (Note: Space is also a value here)

>>> FullName[6]

'M'

We want to find 6th Character of given string of FullName

>>> FullName[0]

'Q'

Here want to find 1st Name:

>>> FullName[0:5]

'Quazi'

Here [0:5] means that from 0 to before 5. Not from 0 to 5.

>>> FullName[0:18]

'Quazi Mahmudul Huq'

>>>

## **List**

A list is a collection which is ordered and changeable. Allows duplicate members. In Python lists are written with square brackets.

>>> list=["Cat","Dog","Cow","Goat"]

>>> type(list)

0: <class 'list'>

List Serial No. start with 0

>>> list[0]

1: 'Cat'

>>> list[1]

2: 'Dog'

>>> list[2]

3: 'Cow'

>>> list[3]

4: 'Goat'

>>> list

5: ['Cat', 'Dog', 'Cow', 'Goat']

>>> num1=[10, 20, 30]

>>> num2=[40, 50]

>>> num3=num1+num2

>>> num3

6: [10, 20, 30, 40, 50]

>>> type(num3)

7: <class 'list'>

Delete from list:

>>> del list[1]

>>> list

8: ['Cat', 'Cow', 'Goat']

Add to list:

>>> list.append("Dog")

>>> list

9: ['Cat', 'Cow', 'Goat', 'Dog']

Check Length of List:

>>> len(list)

10: 4

You can Add Same count Multiple Times:

>>> list.append("Cat")

>>> list

11: ['Cat', 'Cow', 'Goat', 'Dog', 'Cat']

Check How many times “Cat” is in the list.

>>> list.count("Cat")

12: 2

Add character in num3:

>>> num3.append(-20)

>>> num3

13: [10, 20, 30, 40, 50, -20]

Check the maximum number in num3 list:

>>> max(num3)

14: 50

Check the minimum number in num3 list:

>>> min(num3)

15: -20

Remove a number from num3 list:

>>> num3.remove(40)

>>> num3

16: [10, 20, 30, 50, -20]

Reverse order of list num3:

>>> num3.reverse()

>>> num3

18: [-20, 50, 30, 20, 10]

Sort list in ascending order:

>>> num3.sort()

>>> num3

19: [-20, 10, 20, 30, 50]

# Dictionary

Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair.

"john":10 = Here john is key and 10 is value

Dictionaries are optimized to retrieve values when the key is known.

Example: 3 students of class 1 and their Roll Number:

>>> class1={"john":10,"Paul":20,"Natasha":30}

>>> type(class1)

20: <class 'dict'>

Delete a key from Dictionary class1:

>>> del class1["john"]

>>> class1

21: {'Natasha': 30, 'Paul': 20}

Add a key to Dictionary class1:

>>> class1["John"]=10

>>> class1

22: {'John': 10, 'Natasha': 30, 'Paul': 20}

Find the value of Key Natasha:

>>> class1.get("Natasha")

23: 30

How many key have in class1:

>>> class1.keys()

24: dict\_keys(['Paul', 'Natasha', 'John'])

Check the values in class1:

>>> class1.values()

25: dict\_values([20, 30, 10])

How many items you have in Dictionary:

>>> class1.items()

26: dict\_items([('Paul', 20), ('Natasha', 30), ('John', 10)])

Check Items Length:

>>> len(class1)

27: 3

Is Paul a student of class1:

>>> "Paul" in class1

30: True

>>> "Popy" in class1

31: False

**Tuple**

A **tuple** is a collection of objects which ordered and immutable. **Tuples** are sequences, just like lists. The differences between **tuples** and lists are, the **tuples** cannot be changed unlike lists and **tuples** use parentheses, whereas lists use square brackets.

>>> tup1=("Jannat",10,20,30,"Rakhi")

>>> tup1

0: ('Jannat', 10, 20, 30, 'Rakhi')

>>> type(tup1)

1: <class 'tuple'>

>>> tup2=(10,20,30,40,50)

>>> tup2

2: (10, 20, 30, 40, 50)

Count, how many times 20 have in tup2:

>>> tup2.count(20)

3: 1

Slicing the tuple:

>>> tup2[0:3]

4: (10, 20, 30)

Check Serial Number (index) of a value of tup2:

>>> tup2.index(40)

5: 3

We cannot update a tuple, we can delete a tuple:

>>> del tup1

>>> tup1

Traceback (most recent call last):

File "<pyshell#9>", line 1, in <module>

tup1

NameError: name 'tup1' is not defined

>>> tup2

6: (10, 20, 30, 40, 50)

**Python Scripting**

**Modules in Python** are simply **Python** files with a .py extension. The name of the **module** will be the name of the file. A **Python module** can have a set of functions, classes or variables defined and implemented. We have many built-in modules in Python.

OS (Operating System) is a built-in module

>>> import os

To see the Current Working Directory of OS:

>>> os.getcwd()

7: 'C:\\Users\\Administrator\\AppData\\Local\\Programs\\Python\\Python38-32'

 The random module gives access to various useful functions and one of them being able to generate random numbers, which is randint(). **randint()** is an inbuilt function of the random module in Python3.

>>> import random

>>> r1 = random.randint(0,5)

>>> print(r1)

3

The **input()** function get the input from the user and returns a string by stripping a trailing newline. We need input() function in programing necessarily because whenever we need update anything on the device so we need to put the IP Address and our credential (username & password) so that we will use the **input()** function at that time

>>> input("Enter your name: ")

Enter your name: Mahmud

8: 'Mahmud'

You should download and install notepad++ to write script.

Our Mission is to create a script program and a document file. Our target is to read the document file in PowerShell using the Script Program.

Write a script in notepad++ as bellow:

import getpass

import sys

import telnetlib

HOST = "10.10.10.6"

user = input("Enter your telnet username: ")

password = getpass.getpass()

tn = telnetlib.Telnet(HOST)

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

tn.write(b"enable\n")

tn.write(b"1234\n")

tn.write(b"conf t\n")

tn.write(b"hostname CCNA\n")

tn.write(b"int loop 0\n")

tn.write(b"ip address 1.1.1.1 255.255.255.255\n")

tn.write(b"int loop 1\n")

tn.write(b"ip address 2.2.2.2 255.255.255.255\n")

tn.write(b"router ospf 10\n")

tn.write(b"network 0.0.0.0 255.255.255.255 area 0\n")

tn.write(b"end\n")

tn.write(b"exit\n")

print(tn.read\_all().decode('ascii'))

Now save as the file named anyname.py

Save the document file.

Keep the document file and script in same location.

Now go to PowerShell and change location to that location where you have saved your script:

PS C:\Users\Administrator> d:

PS D:\> cd '.\Network Automation\'

PS D:\Network Automation> python net.py

Enter your telnet username: admin

Password:

R1>enable

Password:

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#hostname CCNA

CCNA(config)#int loop 0

CCNA(config-if)#ip address 1.1.1.1 255.255.255.255

CCNA(config-if)#int loop 1

CCNA(config-if)#ip address 2.2.2.2 255.255.255.255

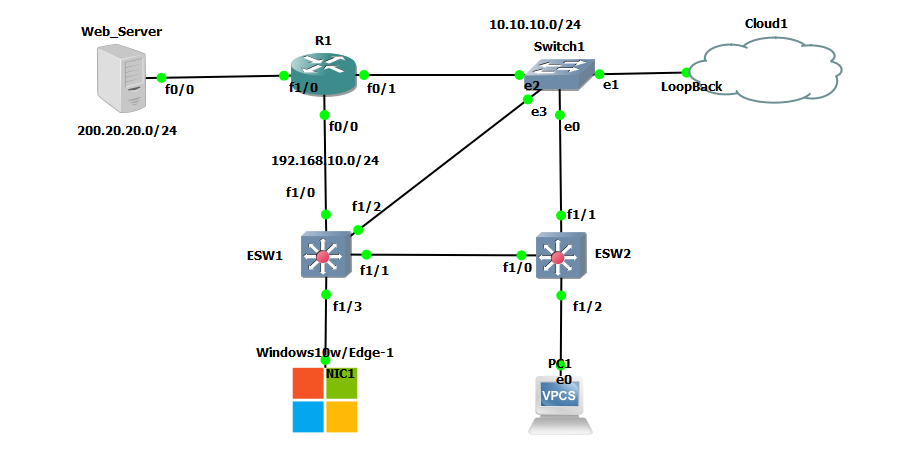
CCNA(config-if)#router ospf 10

CCNA(config-router)#network 0.0.0.0 255.255.255.255 area 0

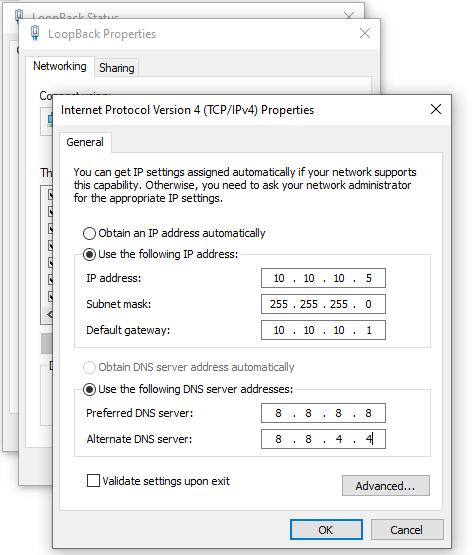
CCNA(config-router)#end

CCNA#exit

Now Check your Router Configuration !!!!



Loopback Interface configuration:



Web Server Configuration:

Web\_Server#configure terminal

Web\_Server(config)#ip http server

Web\_Server(config)#interface fastEthernet 0/0

Web\_Server(config-if)#ip address 200.20.20.2 255.255.255.0

Web\_Server(config-if)#no shutdown

Web\_Server(config)#router ospf 10

Web\_Server(config-router)#network 200.20.20.0 0.0.0.255 area 0

Web\_Server#wr

Router Configuration:

R1#configure terminal

R1(config)#enable password 1234

R1(config)#username admin password 1234

R1(config)#line vty 0 4

R1(config-line)#login local

R1(config-line)#transport input telnet

R1(config-line)#exit

R1(config)#interface fastEthernet 0/1

R1(config-if)#ip address 10.10.10.6 255.255.255.0

R1(config-if)#no shutdown

R1#ping 10.10.10.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.5, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 40/46/52 ms

Configuration ESW1:

ESW1#configure terminal

ESW1(config)#enable password 1234

ESW1(config)#username admin password 1234

ESW1(config)#line vty 0 4

ESW1(config-line)#login local

ESW1(config-line)#transport input telnet

ESW1(config-line)#exit

ESW1(config)#interface fastEthernet 1/2

ESW1(config-if)#ip address 10.10.10.7 255.255.255.0

ESW1(config-if)#no shutdown

ESW1#ping 10.10.10.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/270/1052 ms

Configuration ESW2:

ESW2#configure terminal

ESW2(config)#enable password 1234

ESW2(config)#username admin password 1234

ESW2(config)#line vty 0 4

ESW2(config-line)#login local

ESW2(config-line)#transport input telnet

ESW2(config-line)#exit

ESW2(config)#interface fastEthernet 1/1

ESW2(config-if)#ip address 10.10.10.8 255.255.255.0

ESW2(config-if)#no shutdown

ESW2#ping 10.10.10.5

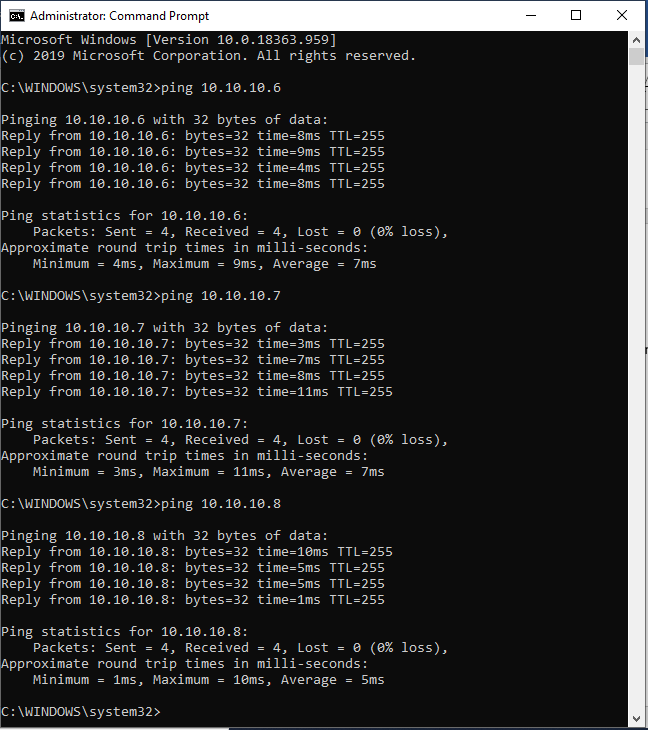
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 68/270/1060 ms

Now Ping from Host PC to GNS3 Router and Ethernet Switches:



PS D:\Network Automation\Test> Python Router.py

Enter your telnet username: admin

Password:

CCNA>enable

Password:

CCNA#conf t

Enter configuration commands, one per line. End with CNTL/Z.

CCNA(config)#hostname CCNA

CCNA(config)#int fa 1/0

CCNA(config-if)#ip address 200.20.20.1 255.255.255.0

CCNA(config-if)#no shut

CCNA(config-if)#int fa 0/1

CCNA(config-if)#no shut

CCNA(config-if)#int fa 0/1.10

CCNA(config-subif)#encapsulation dot1Q 10

CCNA(config-subif)#ip address 192.168.10.1 255.255.255.0

CCNA(config-subif)#no shut

CCNA(config-subif)#ip access-group 150 in

CCNA(config-subif)#ip dhcp pool CCNA

CCNA(dhcp-config)#network 192.168.10.0 255.255.255.0

CCNA(dhcp-config)#default-router 192.168.10.1

CCNA(dhcp-config)#dns-server 8.8.8.8

CCNA(dhcp-config)#router ospf 10

CCNA(config-router)#network 200.20.20.0 0.0.0.255 area 0

CCNA(config-router)#network 192.168.10.0 0.0.0.255 area 0

CCNA(config-router)#$ 150 permit tcp host 192.168.10.2 host 200.20.20.2 eq 80

CCNA(config)#access-list 150 deny icmp host 192.168.10.2 host 200.20.20.2

CCNA(config)#access-list 150 permit ip any any

CCNA(config)#end

CCNA#wr

Building configuration...

[OK]

CCNA#exit

PS D:\Network Automation\Test>

PS D:\Network Automation\Test> python vlanMultiple.py

Enter your telnet username: admin

Password:

Telnet to host7

Telnet to host8

ESW2>enable

Password:

ESW2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

ESW2(config)#vlan 8

ESW2(config-vlan)#name Auto\_VLAN\_8

ESW2(config-vlan)#vlan 9

ESW2(config-vlan)#name Auto\_VLAN\_9

ESW2(config-vlan)#vlan 10

ESW2(config-vlan)#name Auto\_VLAN\_10

ESW2(config-vlan)#vlan 11

ESW2(config-vlan)#name Auto\_VLAN\_11

ESW2(config-vlan)#vlan 12

ESW2(config-vlan)#name Auto\_VLAN\_12

ESW2(config-vlan)#vlan 13

ESW2(config-vlan)#name Auto\_VLAN\_13

ESW2(config-vlan)#vlan 14

ESW2(config-vlan)#name Auto\_VLAN\_14

ESW2(config-vlan)#exit

ESW2(config)#exit

ESW2#wr

Building configuration...

[OK]

ESW2#exit

PS D:\Network Automation\Test> python ESW1.py

Enter your telnet username: admin

Password:

ESW1>enable

Password:

ESW1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

ESW1(config)#hostname CCNA

CCNA(config)#int fa 1/0

CCNA(config-if)#switchport mode trunk

CCNA(config-if)#int fa 1/1

CCNA(config-if)#switchport mode trunk

CCNA(config-if)#switchport trunk allowed vlan 10

CCNA(config-if)#int fa 1/2

CCNA(config-if)#switchport mode access

CCNA(config-if)#switchport access vlan 10

CCNA(config-if)#exit

CCNA(config)#exit

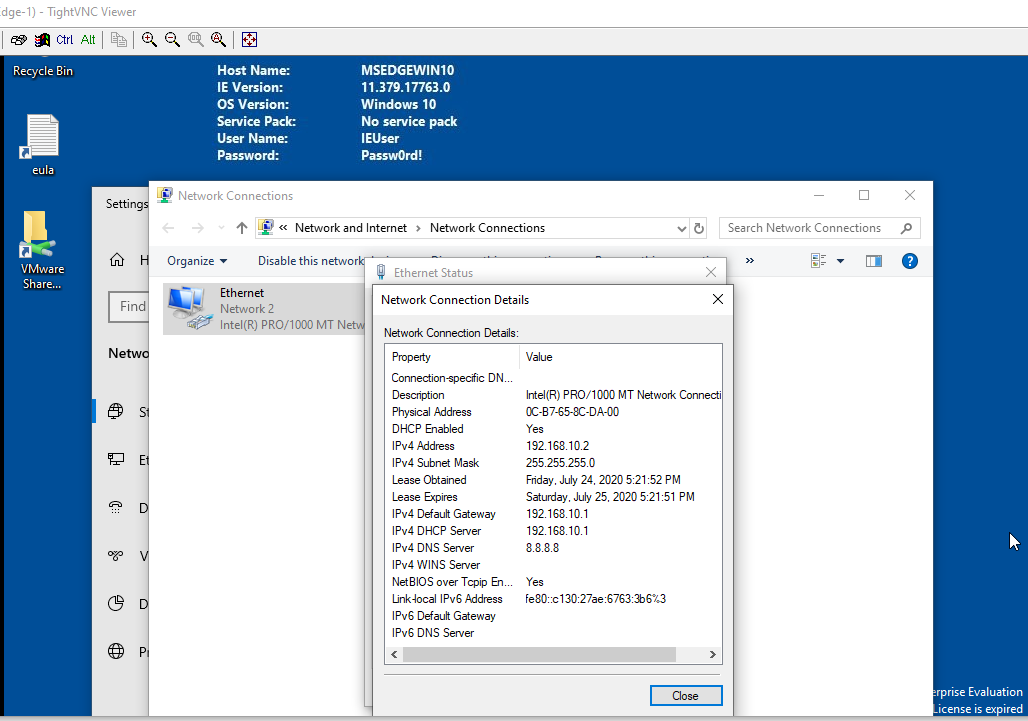
CCNA#wr

Building configuration...

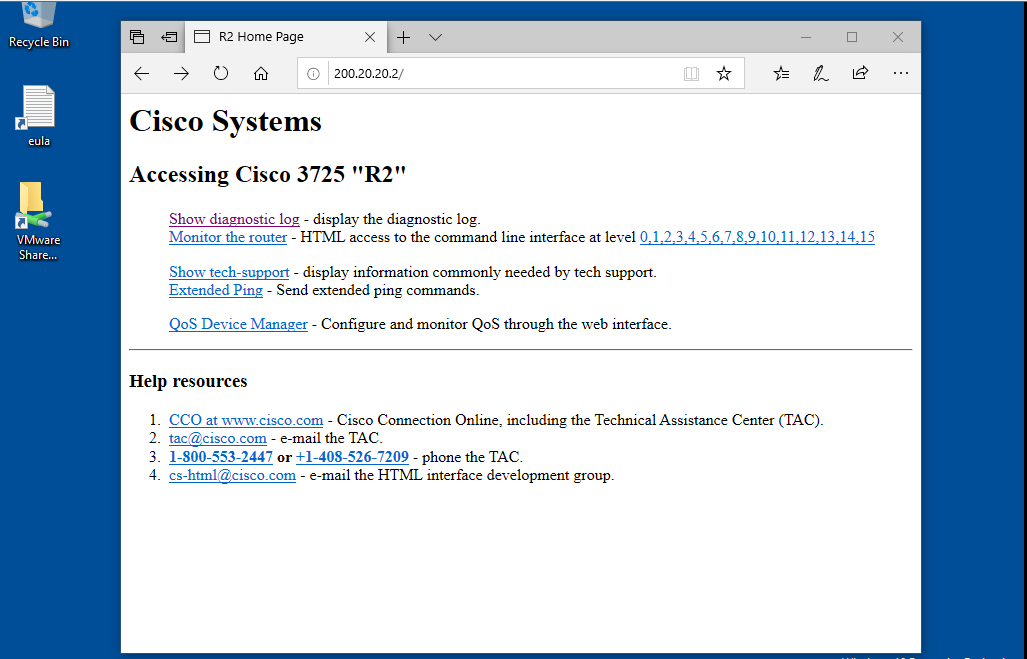
[OK]

CCNA#exit

Now Check your GNS3 Windows 10 Interface that It already configured from DHCP Server



Now Browse the Web Server from GNS3 Windows 10 PC:



Now Ping the GNS3 Web Server:

