

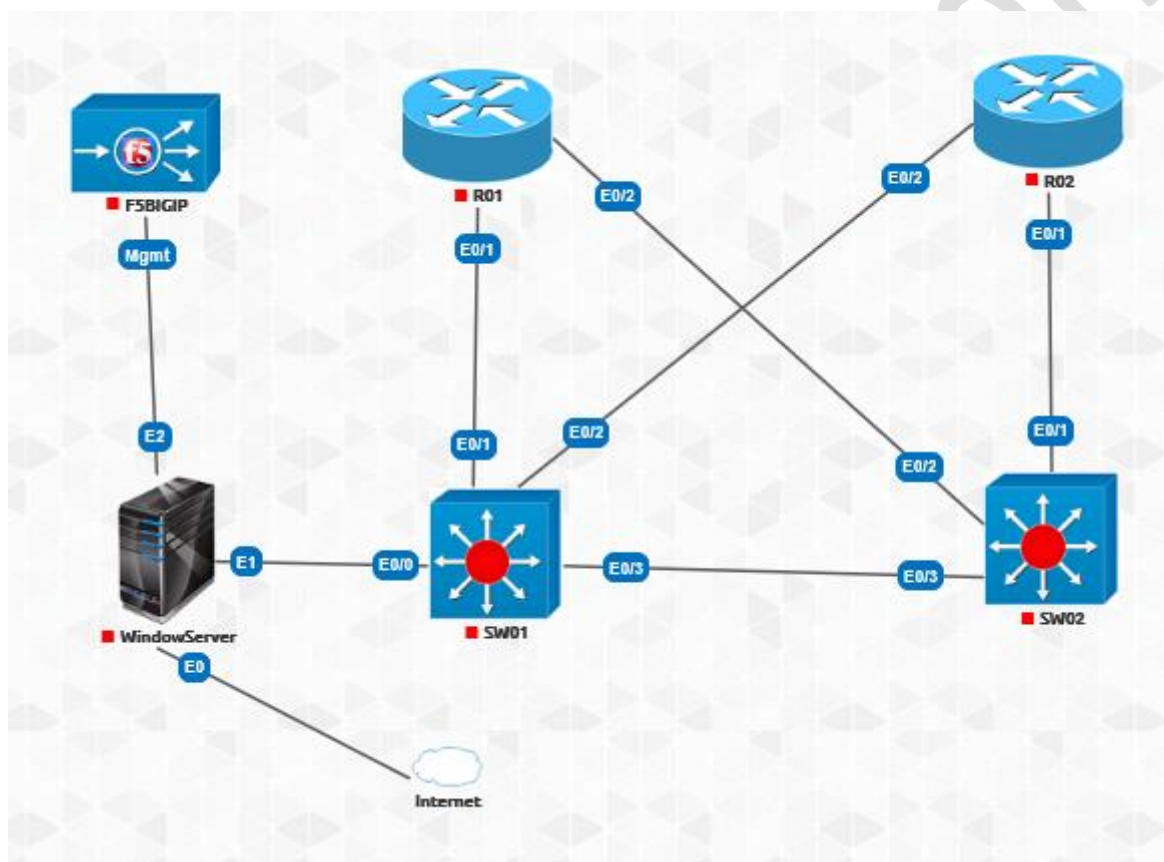
## Telnetting to device for fetching information

**Platform:** <https://racks.uninets.com>

**Lab Name:** Python Networking

### Topology

Telnetting to device for fetching information



### Task

- Make a script to enter in the device through telnet and check “sh int status” output.
- Configure a new loopback 100 and give its description as Using-Python-Automation.
- Configure a new layer2 vlan, id 2 and give its description as Python-vlan-2.
- Configure some new layer2 vlans, id from 2 to 50 and give its description as Python-vlan-x using loops.

### Explanation

We can use web python documentations for reference of basic telnet script, after that with little emendation can achieve all of the above mentioned tasks. Here we are leveraging telnet library and get pass for telnet ting to device and receiving password in secure way.

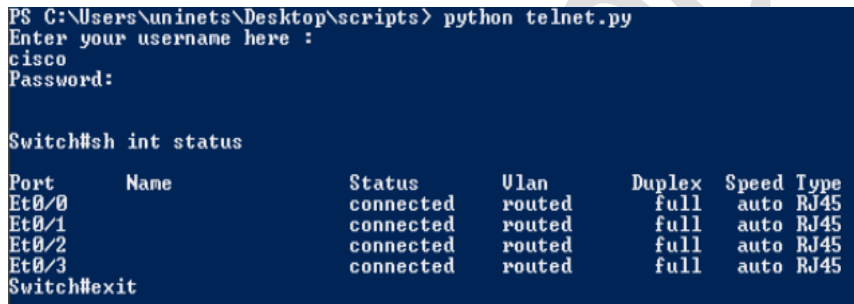
## Configuration and Verification

Below is the script to run code for telnetting to device and interface status output, result is presented in blue screen from real devices, '1.1.1.1' is the loopback IP of Switch01.

```
import getpass
import telnetlib

V_router = '1.1.1.1'
V_username = input ('Enter your username here : \n')
V_password = getpass.getpass()
V_script_start = telnetlib.Telnet(V_router)
V_script_start.read_until(b"Username: ")
V_script_start.write(V_username.encode('ascii') + b'\n')
if V_password:
    V_script_start.read_until(b'Password:')
    V_script_start.write(V_password.encode('ascii') + b'\n')

V_script_start.write (b'ter len 0 \n')
V_script_start.write (b'sh int status \n')
V_script_start.write (b'exit \n')
a = V_script_start.read_all()
print(str(a,encoding= 'utf-8')) or print(a.decode('ascii'))
```



```
PS C:\Users\uninets\Desktop\scripts> python telnet.py
Enter your username here :
cisco
Password:

Switch#sh int status

Port      Name      Status      Ulan      Duplex  Speed  Type
Et0/0     connected routed      full     auto   RJ45
Et0/1     connected routed      full     auto   RJ45
Et0/2     connected routed      full     auto   RJ45
Et0/3     connected routed      full     auto   RJ45
Switch#exit
```

Below is the code for configuring a loopback along with a description, output presented as the blue screen where in previous code making a little change and achieving different output, all changes are highlighted in red colour.

```
import getpass
import telnetlib

V_router = '1.1.1.1'
V_username = input ('Enter your username here : \n')
V_password = getpass.getpass()
V_script_start = telnetlib.Telnet(V_router)
V_script_start.read_until(b"Username: ")
V_script_start.write(V_username.encode('ascii') + b'\n')
if V_password:
    V_script_start.read_until(b'Password:')
    V_script_start.write(V_password.encode('ascii') + b'\n')

V_script_start.write (b'conf ter \n')
V_script_start.write (b'int loop 100 \n')
V_script_start.write (b'description Using-Python-Automation \n')
V_script_start.write (b'end \n')
```

```
V_script_start.write(b'ter len 0 \n')
V_script_start.write(b'sh int description \n')
V_script_start.write(b'exit \n')
```

```
a = V_script_start.read_all()
print(str(a,encoding= 'utf-8')) or print(a.decode('ascii'))
```

```
PS C:\Users\uninets\Desktop\scripts> ls

Directory: C:\Users\uninets\Desktop\scripts

Mode                LastWriteTime         Length Name
----                -
-a---             12/10/2019   4:31 PM           738 Device_uptime.py
-a---             12/8/2019    5:16 AM           224 print.py
-a---             12/10/2019   4:31 PM           884 Telnet_2_router.py
-a---             12/21/2019   2:47 AM           573 Workbook_1_1_First-script-telnet.py
-a---             12/21/2019   3:34 AM           831 Workbook_1_2_First-script-telnet.py

PS C:\Users\uninets\Desktop\scripts> python Workbook_1_2_First-script-telnet.py
Enter your username here :
cisco
Password:

Switch#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int loop 100
Switch(config-if)#description Using-Python-Automation
Switch(config-if)#exit
Switch(config)#exit
Switch#ter len 0
Switch#show int description
Interface                Status      Protocol Description
-----                -
Et0/0                    up          up
Et0/1                    up          up
Et0/2                    up          up
Et0/3                    up          up
Lo0                      up          up
Lo100                   up          up      Using-Python-Automation
V11                      admin down down
Switch#exit

PS C:\Users\uninets\Desktop\scripts>
```

Described followscode is for configuring vlan with a name and by adding someadditional commands, more than single vlan can be added. Output presented as the blue screen again and all changes are highlighted in red colour wrt baseline script.

```
import getpass
import telnetlib

V_router = '1.1.1.1'
V_username = input("Enter your username here : \n")
V_password = getpass.getpass()
V_script_start = telnetlib.Telnet(V_router)
V_script_start.read_until(b"Username: ")
V_script_start.write(V_username.encode('ascii') + b'\n')
if V_password:
    V_script_start.read_until(b'Password:')
    V_script_start.write(V_password.encode('ascii') + b'\n')

V_script_start.write(b'conf ter \n')
V_script_start.write(b'int vlan2 \n')
V_script_start.write(b'namePython-vlan-2 \n')
V_script_start.write(b'end \n')
V_script_start.write(b'ter len 0 \n')
V_script_start.write(b'sh int description \n')
V_script_start.write(b'exit \n')

a = V_script_start.read_all()
print(str(a,encoding= 'utf-8')) or print(a.decode('ascii'))
```

```

Enter your username here :
cisco
Password:

Switch#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name Python-vlan-2
Switch(config-vlan)#exit
Switch(config)#exit
Switch#ter len 0
Switch#show vlan

VLAN Name                Status    Ports
-----
1    default                active
2    Python-vlan-2         active
1002 fddi-default          act/unsup
1003 token-ring-default    act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default       act/unsup

VLAN Type  SAID      MTU   Parent  RingNo BridgeNo  Stp    BrdgMode Trans1 Trans2
-----
1    enet  100001   1500  -       -       -       -       -       0      0
2    enet  100002   1500  -       -       -       -       -       0      0
1002 fddi  101002   1500  -       -       -       -       -       0      0
1003 tr    101003   1500  -       -       -       -       -       0      0
1004 fdnet 101004   1500  -       -       -       ieee  -       0      0
1005 trnet 101005   1500  -       -       -       ibm   -       0      0

Remote SPAN VLANs
-----

Primary Secondary Type          Ports
-----

Switch#exit
PS C:\Users\uninets\Desktop\scripts>

```

For creating multiple vlans, not adding manual commands but leveraging the for loop functionality (changes are shown in red color). Results are not mentioned as it is huge output and give it a try by yourself.

```

import getpass
import telnetlib

V_router = '1.1.1.1'
V_username = input('Enter your username here : \n')
V_password = getpass.getpass()
V_script_start = telnetlib.Telnet(V_router)
V_script_start.read_until(b'Username: ')
V_script_start.write(V_username.encode('ascii') + b'\n')
if V_password:
    V_script_start.read_until(b'Password:')
    V_script_start.write(V_password.encode('ascii') + b'\n')

V_script_start.write(b'conf ter \n')
for Var_iterate in range(2,51):
    V_script_start.write(b'Vlan ' + str(Var_iterate).encode('ascii') + b'\n')
    V_script_start.write(b'name Python-vlan-' + str(Var_iterate).encode('ascii') + b'\n')
V_script_start.write(b'end \n')
V_script_start.write(b'ter len 0 \n')
V_script_start.write(b'vlan \n')
V_script_start.write(b'exit \n')

a = V_script_start.read_all()
print(str(a,encoding= 'utf-8')) or print(a.decode('ascii'))

```