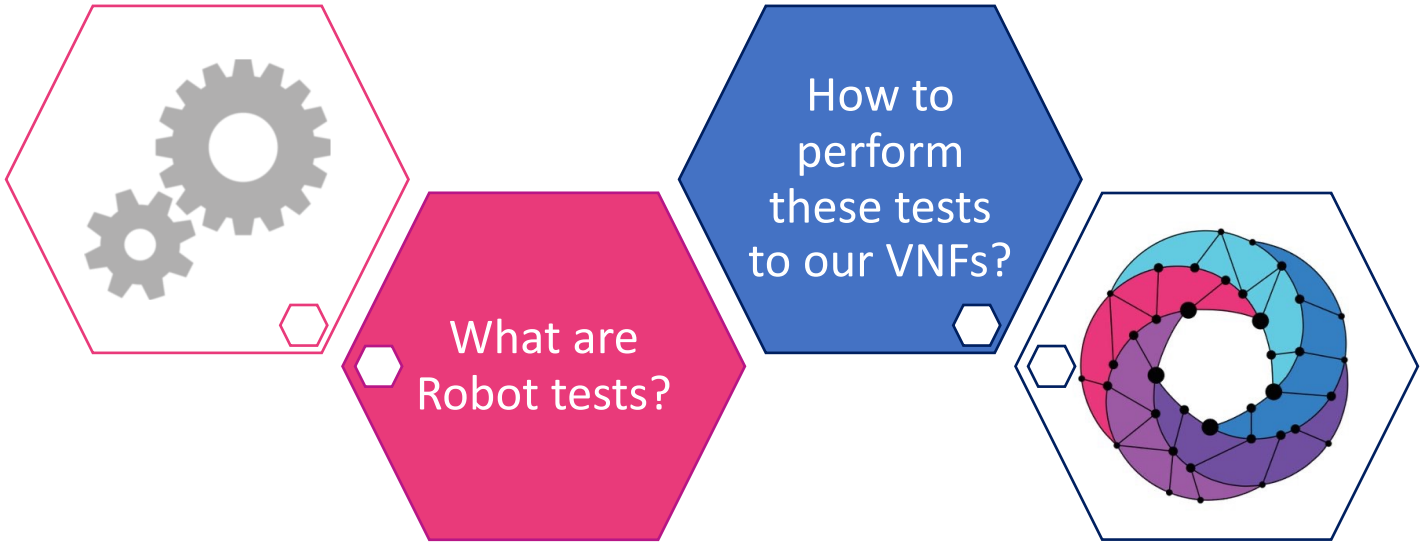
# How to test your VNFs with the Robot Framework



# How will we perform the test?

Robot Framework

ROBOT
FRAME
WORK/

# What do Robot tests perform?

Perform network tests between 2 VNFs:

- Bandwidth
- Open Ports
- Packet Loss
- Transmission Speed
- Jitter
- etc

# First of all, start by installing the needed framework

```
# install the robot framework
$ pip install robotframework

# check the installation
$ robot --version
```

# Create the needed files to perform the test

We will be making a Jitter test

```
# create the folder
$ mkdir jitter
$ cd jitter/

# create the files
$ touch jitter.py
$ touch testJitter.robot
```

# After running these commands, you should have the following file structure

```
.
└── robot_test_tutorial
    └── jitter
        ├── jitter.py
        └── testJitter.robot
```

# How to use Robot framework to perform network tests between VNFs?

# We will use environment variables to define the 2 VNFs and the Jitter values

Start by exporting these variables.

```
# VNF 1
$ export jitter_host1_ip=10.0.12.223 # change to your VNF 1 IP
# change to your credentials
$ export jitter_host1_username=ubuntu
$ export jitter_host1_password=password

# VNF 2
$ export jitter_host1_ip=10.0.12.212 # change to your VNF 2 IP
# change to your credentials
$ export jitter_host1_username=ubuntu
$ export jitter_host1_password=password

# Jitter
$ export jitter_comparator=less_or_equal_than
$ export jitter_threshold=0.03
```

Edit the *jitter/jitter.py* file.
Add the following content:

```python
import json
import os
import pip
import importlib.util

host1 = os.getenv('jitter_host1_ip')
username1 = os.getenv('jitter_host1_username')
password1 = os.getenv('jitter_host1_password')

host2 = os.getenv('jitter_host2_ip')
username2 = os.getenv('jitter_host2_username')
password2 = os.getenv('jitter_host2_password')



packages = ['paramiko','robotframework']
for package in packages:
    if (spec := importlib.util.find_spec(package)) is  None:
        pip.main(['install', package])

import paramiko
```

Edit the *jitter/jitter.py* file.
Add the test function:

```python
#test
def jitter():
    machine1 = paramiko.SSHClient()
    machine1.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    machine2 = paramiko.SSHClient()
    machine2.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    try:
        machine1.connect(hostname=host1, username=username1, password=password1)
        machine2.connect(hostname=host2, username=username2, password=password2)
    except:
        print("[!] Cannot connect to the SSH Server")
        exit()

    # Executing iPerf commands
    machine1.exec_command("iperf3 -s -1")
    stdin, stdout, stderr = machine2.exec_command(f"iperf3 -c {host1} -u --json -t 5")
    iperfResult = stdout.read().decode()
    obj = json.loads(iperfResult)
    try:
        jitter_ms = float(obj['end']['sum']['jitter_ms'])
        print(f"Jitter: {jitter_ms}")
    except:
        return "Not found"
    return jitter_ms


if __name__ == '__main__':
    jitter()
```

# We will now add the needed code to the robot test file.

Add the following to *jitter/testJitter.robot:*

```
*** Settings ***
Library        jitter.py

*** Test Cases ***
Testing if the jitter is %{jitter_comparator} %{jitter_threshold} ms
    ${COMPARATOR}=       Run Keyword If      '%{jitter_comparator}' == 'more_than'        Set Variable    >
    ...    ELSE IF    '%{jitter_comparator}' == 'more_or_equal_than'        Set Variable    >=
    ...    ELSE IF    '%{jitter_comparator}' == 'less_than'        Set Variable    <
    ...    ELSE IF    '%{jitter_comparator}' == 'less_or_equal_than'        Set Variable    <=
    ...    ELSE     Fail  \nComparator has not been defined


    ${jitter_ms}=    jitter
    IF     '${jitter_ms}' != '-1'
    Should Be True    ${jitter_ms} ${COMPARATOR} %{jitter_threshold}
    ELSE
    FAIL    \nImpossible to compute Jitter
    END
```

Before running the tests, you need to have iperf3 installed in both VNFs.

This is the package that will obtain the values so that we can run the tests.

You have 2 options:

- Install through SSH command in each VNF : sudo apt-get install iperf3
- Through the VNF cloud-init files

# Now, let’s run our test:

```
# if you haven’t already exported the environment variables, go back in the tutorial and export them
# In this case, we are checking if the average Jitter between the VNFs is less than 0.03 ms
$ robot jitter # this references the folder
```

# If everything goes accordingly, you should have this output:

```
•100% → robot jitter
==============================================================================
Jitter
==============================================================================
Jitter.testJitter
==============================================================================
Testing if the jitter is less_or_equal_than 0.03 ms                   | PASS |
------------------------------------------------------------------------------
Jitter.testJitter                                                     | PASS |
1 test, 1 passed, 0 failed
==============================================================================
Jitter                                                                | PASS |
1 test, 1 passed, 0 failed
==============================================================================
Output:  /Users/hacker/Bolsa/tutoriais/4 - Robot_tests/output.xml
Log:     /Users/hacker/Bolsa/tutoriais/4 - Robot_tests/log.html
Report:  /Users/hacker/Bolsa/tutoriais/4 - Robot_tests/report.html
```

# Inside log.html you can find pore information on the test:

**Test Execution Log**

SUITE **Jitter** 00:00:08.978

**Full Name:** Jitter
**Source:** /Users/hacker/Bolsa/tutoriais/4 - Robot_tests/jitter
**Start / End / Elapsed:** 20220224 16:46:00.428 / 20220224 16:46:09.406 / 00:00:08.978
**Status:** 1 test total, 1 passed, 0 failed, 0 skipped

SUITE **testJitter** 00:00:08.943

**Full Name:** Jitter.testJitter
**Source:** /Users/hacker/Bolsa/tutoriais/4 - Robot_tests/jitter/testJitter.robot
**Start / End / Elapsed:** 20220224 16:46:00.461 / 20220224 16:46:09.404 / 00:00:08.943
**Status:** 1 test total, 1 passed, 0 failed, 0 skipped

TEST **Testing if the jitter is less_or_equal_than 0.03 ms** 00:00:06.866

**Full Name:** Jitter.testJitter.Testing if the jitter is less_or_equal_than 0.03 ms
**Start / End / Elapsed:** 20220224 16:46:02.537 / 20220224 16:46:09.403 / 00:00:06.866
**Status:** PASS

KEYWORD ${COMPARATOR} = Builtin.**Run Keyword If** '%{jitter_comparator}' == 'more_than', Set Variable, >, ELSE IF, '%{jitter_comparator}' == 'more_or_equal_than', Set Variable, >=, ELSE IF, '%{jitter_comparator}' == 'less_than', Set Variable, <, ELSE IF, '%{jitter_comparator}' == 'less_or_equal_than', Set Variable, <=, ELSE, Fail, \nComparator has not been defined 00:00:00.001

**Documentation:** Runs the given keyword with the given arguments, if `condition` is true.
**Start / End / Elapsed:** 20220224 16:46:02.538 / 20220224 16:46:02.539 / 00:00:00.001

KEYWORD Builtin.**Set Variable** <= 00:00:00.000

```
16:46:02.539  INFO  ${COMPARATOR} = <=
```

KEYWORD ${jitter_ms} = jitter.**Jitter** 00:00:06.860

IF **'${jitter_ms}' != '-1'** 00:00:00.001

ELSE 00:00:00.000

# The code developed during this tutorial is available at

*https://github.com/5gasp/tutorials*

If you have any questions regarding the contents addressed in this tutorial you can send an e-mail to Rafael Direito rdireito@av.it.pt, or contact us via contact@5gasp.eu