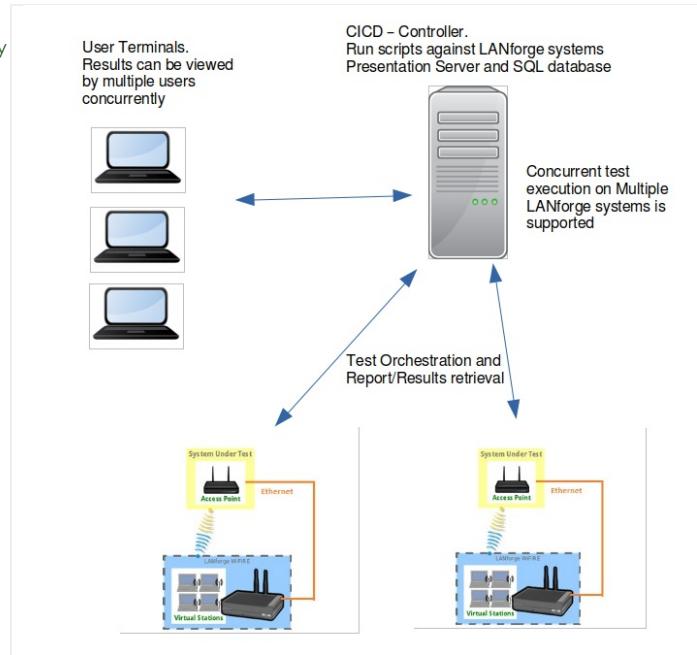


## Basic CICD AP Testing with LANforge

**Goal:** Set up Basic CICD a LANforge system, Regression Automation and Reporting with data from previous runs.

The LANforge CICD framework provides an ability to execute a suite of tests and report results.



### 1. The following steps are discussed

- A. Set Up CICD Controller and Environment
- B. Set Up The JSON Configuration Files
- C. Test Execution
- D. Test Results

### 2. Set Up CICD Controller and Environment

A. clone lanforge-scripts from <https://github.com/greearb/lanforge-scripts>

B. run /lanforge-scripts/py-scripts/update\_dependencies.py to install python packages for generating output

C. Install web server:

The web server is to allow for viewing of results from User Terminals

The CICD - Controller is not dependent on a web server, results may be viewed locally on CICD - Controller

A. **LANforge** LANforge installation using kinstall.pl installs a web server on LANforge

LANforge installation installs an httpd server, LANforge may be used for storing and displaying results.

For the following example a separate LANforge system (Fedora) was used as the CICD - Controller and httpd web server.

install httpd and configure server \$ sudo dnf install httpd

B. **Fedora**

install apache2 and configure server \$ sudo apt install apache2

C. **Ubuntu**

D. Install mail service for email of links to results

For the example below Linux mailx program was used

Installation of mail services is dependent on the environment in which the CICD - Controller is installed.

The CICD - Controller is not dependent on email services

E. Install database sqlite3

A. **Fedora** \$ sudo dnf install sqlite3

B. **Ubuntu** \$ sudo apt-get update \$ sudo apt-get install sqlite3

F. Create a html-reports directory. On lanforge /home/lanforge/html-reports

G. Determine sqlite3 database name and location, sqlite3 db will be created. ./tools/qa\_sqlite3.db

### 3. Set Up The JSON Configuration Files

A. There are three JSON configuration input files described below. For all the JSON configuration files the CAPITALIZED parameters allow for a value to be entered into one location and used in multiple areas of the CICD framework. For example in ssid\_idx=1 the SSID\_USED is set to asus11ax-5. For the test suite below the SSID\_USED may be entered instead of asus11ax-5, thus if the SSID changes, the SSID will need to be modified in ct\_AX88U\_dut, the ct\_tests.json will remain untouched. This reduces the need to modify the ct\_test.json for SSID changes that would affect multiple tests

A. **--json\_rig test\_rig.json** this JSON file describes LANforge test rig, [Example ct\\_test\\_rig.json](#)  
The test\_rig.json describes the LANforge system and test parameters for the CICD - Controller

B. **--json\_dut ct\_AX88U\_dut.json** this JSON file describes the AP, [Example ct\\_AX88U\\_dut.json](#)  
the ct\_AX88U\_dut.json describes the device under test parameters, DUT\_SET\_NAME: DUT\_NAME ASUSSRT-AX88U for example is used by Chamberview Tests

C. **--json\_test ct\_tests.json** this JSON file describes the tests, [Example ct\\_tests.json](#)

The tests may use the CAPITALIZED variables or may be entered with the command line arguments as they would be entered on the command line.

The tests are not limited to only python tests

B. **test\_rig.json**

```
{  
    "test_rig": {  
        "Notes": [  
            "This JSON file describes LANforge system and test run configuration"  
        ]  
    },  
    "test_rig_parameters": {  
        "TEST_BED": "CT-TEST-001",  
        "TEST_RIG": "CT-TEST-001",  
        "DATABASE_SQLITE": "./tools/qa_sqlite3.db",  
        "LF_MGR_IP": "192.168.100.116",  
        "LF_MGR_PORT": "8080",  
        "LF_MGR_USER": "lanforge",  
        "LF_MGR_PASS": "lanforge",  
        "UPSTREAM_PORT": "1.1.eth2",  
        "TEST_TIMEOUT": 600,  
        "EMAIL_LIST_PRODUCTION": "support@candelatech.com",  
        "EMAIL_LIST_TEST": "support@candelatech.com",  
        "EMAIL_TITLE_TXT": "Lanforge QA Testing",  
        "EMAIL_TXT": "Lanforge QA Testing"  
    }  
}
```

C. **ct\_AX88U\_dut.json**

```
{  
    "ct_AX88U_dut": {  
        "Notes": [  
            "The device undertest configuration is contained in this file"  
        ]  
    },  
    "test_dut": {  
        "DUT_SET_NAME": "DUT_NAME ASUSSRT-AX88U",  
        "USE_DUT_NAME": "ASUSSRT-AX88U",  
        "wireless_network_dict": {  
            "ssid_idx=0": {"ssid_idx": "0", "SSID_USED": "asus11ax-2", "SSID_PW_USED": "hello123", "BSSID": "3c:7c:3f:55:4d:60", "S  
            "ssid_idx=1": {"ssid_idx": "1", "SSID_USED": "asus11ax-5", "SSID_PW_USED": "hello123", "BSSID": "3c:7c:3f:55:4d:64", "S  
        }  
    }  
}
```

```

D. ct_tests.json
{
    "ct_tests_001": {
        "Notes": [
            "This JSON file describes tests to be run by LANforge system"
        ],
        "test_suites": {
            "suite_wc": {
                "create_chambeview_dut_wc": {
                    "enabled": "TRUE",
                    "load_db": "skip",
                    "command": "create_chambeview_dut.py",
                    "args": "",
                    "args_list": [
                        " --lfmgr LF_MGR_IP --port LF_MGR_PORT --dut_name DUT_NAME",
                        " --ssid 'ssid_idx=0 ssid=SSID_USED security=SECURITY_USED password=SSID_PW_USED bssid=BSSID'",
                        " --ssid 'ssid_idx=1 ssid=SSID_USED security=SECURITY_USED password=SSID_PW_USED bssid=BSSID'",
                        " --sw_version DUT_SW --hw_version DUT_HW --serial_num DUT_SERIAL --model_num DUT_NAME"
                    ]
                },
                "create_chambeview_wc": {
                    "enabled": "TRUE",
                    "load_db": "skip",
                    "command": "create_chambeview.py",
                    "args": "",
                    "args_list": [
                        " --lfmgr LF_MGR_IP --port LF_MGR_PORT --delete_scenario",
                        " --create_scenario scenario_wpa2_wc",
                        " --raw_line \"profile_link 1.1 STA-AC 19 'DUT: DUT_NAME Radio-1' NA wiphy7,AUTO -1 NA\"",
                        " --raw_line \"profile_link 1.1 upstream-dhcp 1 NA NA UPSTREAM_PORT,AUTO -1 NA\""
                    ]
                },
                "wifi_capacity": {
                    "enabled": "TRUE",
                    "timeout": "600",
                    "iterations": "1",
                    "load_db": "skip",
                    "command": "lf_wifi_capacity_test.py",
                    "args": "",
                    "args_list": [
                        " --mgr LF_MGR_IP --port LF_MGR_PORT --lf_user LF_MGR_USER --lf_password LF_MGR_PASS --instance_name s
                        " --upstream UPSTREAM_PORT --batch_size 1,10,19 --loop_iter 1 --protocol UDP-IPv4 --duration 6000",
                        " --pull_report --local_lf_report_dir REPORT_PATH --test_tag 'wpa2_wc'",
                        " --test_rig TEST_RIG",
                        " --set DUT_SET_NAME"
                    ]
                },
                "lf_qa": {
                    "enabled": "TRUE",
                    "timeout": "600",
                    "load_db": "skip",
                    "command": "./tools/lf_qa.py",
                    "args": "",
                    "args_list": [
                        " --path REPORT_PATH --store --png --database DATABASE_SQLITE"
                    ]
                }
            }
        }
    }
}

```

#### E. sample command with above data:

```

./lf_check.py --json_rig ct_test_rig.json \
--json_dut ct_AX88U_dut.json \
--json_test ct_tests.json \
--suite "suite_wc" \
--path '/home/lanforge/html-reports/ct_results_directory'

```

#### 4. Set Up The JSON Configuration Files

A. The lf\_check.py is run form the lanforge-scripts/py-scripts/tools directory

B. lf\_check.py uses three JSON files as input:

For Example:

**ct\_test\_rig.json** - describes the LANforge test rig configuration  
**ct\_AX88U\_dut.json** - describes the device under test  
**ct\_tests.json** - describe the tests to be run.

#### 5. lf\_check.py execution, simple command example

A. ./lf\_check.py --json\_rig ct\_test\_rig.json \
--json\_dut ct\_AX88U\_dut.json \
--json\_test ct\_tests.json \
--suite "suite\_wc" \
--path '/home/lanforge/html-reports/ct\_results\_directory'

#### 6. Sample email sent on run

Get Messages Write chat Address Book Tag Quick Filter Search <Ctrl+K>

From LANforge <lanforge@codelatech.com> Reply Reply All Forward Archive Junk Delete More

Subject Lanforge QA Testing [192.168.95.6] 2021-10-13 06:21:14.071302 7:21 AM

To support@codelatech.com

Lanforge QA Testing lanforge target 192.168.100.116  
Results from 192.168.100.116:  
[http://192.168.100.116/html-reports/ct\\_results\\_directory/2021-10-13-06-18-12\\_lf\\_check/2021-10-13-06-18-12-lf\\_check.html](http://192.168.100.116/html-reports/ct_results_directory/2021-10-13-06-18-12_lf_check/2021-10-13-06-18-12-lf_check.html)

QA Report Dashboard:  
[http://192.168.100.116/html-reports/ct\\_results\\_directory/2021-10-13-06-18-12\\_lf\\_check/2021-10-13-06-21-11\\_lf\\_qa/2021-10-13-06-21-11-lf\\_qa.html](http://192.168.100.116/html-reports/ct_results_directory/2021-10-13-06-18-12_lf_check/2021-10-13-06-21-11_lf_qa/2021-10-13-06-21-11-lf_qa.html)

NOTE: Diagrams are links in dashboard

## 7. If\_check.py: sample If\_check.py Report



### Objective

Run QA Tests

LANForge

LANforge	kernel version	server version	gitl version	gitl build date	gitl git sha	scripts gitl sha
c1523c-3b7b	Version: 3.4.4 Compiled on: Mon Oct 11 11:30:20 2021 05:51:28 PM PDT	5.15.0-rc5+	5.4.4	Mon 11 Oct 2021 06:59:32 PM PDT	e7462c3f07bd9fb139b240edc0cb7437871bcb319	e9888fc23d5cc0e429de7b877b8661e408ec7560\n

### LANForge Radios

Radio	WiFi-Radio Driver	Radio Capabilities	Firmware Version	max_sta	max_vap	max_vifs
1.1.wiphy0	ath10k(9984)	802.11bgn-AC	10.4b-c-9984 xH-13-774302ee5	128	24	64
1.1.wiphy1	ath10k(9984)	802.11an-AC	10.4b-c-9984 xH-13-774302ee5	128	24	64
1.1.wiphy2	ath9k()	802.11abgn	<ath9k radios lack firmware>	2048	32	2048
1.1.wiphy3	ath10k(9984)	802.11abgn-AC	10.1-c1-8a..._xH-022... bcdb24ff	127	24	64
1.1.wiphy4	iwifi(AX200)	802.11abgn-AX	release/core@2.3eccbd0da	1	1	1
1.1.wiphy5	iwifi(AX210)	802.11abgn-AX	release/core@2.3eccbd0da	1	1	1
1.1.wiphy6	iwifi(AX210)	802.11abgn-AX	release/core@2.3eccbd0da	1	1	1
1.1.wiphy7	mt7915e()	802.11abgn-AX	<no firmware data>	19	16	19

### LF Check Test Results

Test	Command	Duration	Start	End	Result	STDOUT	STDERR
create_chamberview_dut_wc	/create_chamberview_dut.py	0d 2h 823293 ms	2021-10-13-06-18-14	2021-10-13-06-18-15	Success	STDOUT	
create_chamberview_wc	/create_chamberview.py	0d 5h 987380 ms	2021-10-13-06-18-19	2021-10-13-06-18-20	Success	STDOUT	
wifi_capacity	/rf_wifi_capacity_test.py	0d 1h 5m 770807 ms	2021-10-13-06-19-21	2021-10-13-06-21-11	Success	STDOUT	
If_qa	/tools/lf_qa.py	0d 2h 220803 ms	2021-10-13-06-21-11	2021-10-13-06-21-12	Success	STDOUT	



## 8. If\_qa.py

If\_qa.py: process kpi.csv, produces html/pdf results, produces plotly png and interactive graphs from test run kpi

### sample command:

```
./lf_qa.py --path /home/lanforge/html-reports/ct_results_directory/(results dir of lf_check.py)\n  --store \\n  --png \\n  --database ./tools/qa_aqlite3.db
```

## 9. If\_qa.py: sample If\_qa.py Report



#### Objective

QA Verification

#### Device Under Test

DUT	SW version	HW version	SN
ASUSRT-AX88U	DUT_SW_NA	DUT_HW_NA	NA

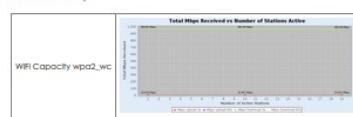
#### Test Rig: CT-TEST-001 Links

[PDF Report](#)  
[Current Test Suite Results Directory](#)  
[All Test Rig Test Suites Results Directory](#)

#### Test Suite

Test	Test_Tag	Links
WiFi Capacity	wpa2_wc	<a href="#">html</a> / <a href="#">pdf</a>

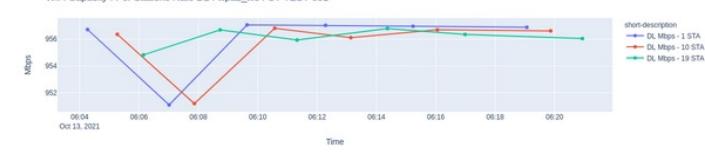
#### Suite Summary



#### QA Test Results



#### WiFi Capacity : Per Stations Rate DL : wpa2\_wc : CT-TEST-001



## 10. Sample If\_heck.py Output example If\_check Report

### 11. Test Control Inputs in Test Suite JSON

- A.   
Allows for individual test enable and disable of the test.
- B.   
Allows for loading a LANforge database prior to the test run.
- C.   
Allows for test to have individual timeout other then default.
- D.   
Allows for test to run multiple iterations.