



**5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION,  
SYSTEM INTEGRATION, AND SHOWCASING**

[H2020 - Grant Agreement No. 815178]

Deliverable D5.2

# System-Level Tests and Verification (Release B)

**Editor** T. Sarlas, G. Xilouris (NCSR D)

**Contributors** NCSR D, UMA, UNIS, SHC, FhG, ATOS, ATH, TID, COS, FON, INF, NEM, FOG, REL, IHP, UPV, INT, OA

**Version** 1.0

**Date** 21/09/2021

**Distribution** PUBLIC (PU)



## List of Authors

---

Listed in previous page	All partners involved in T5.1
-------------------------	-------------------------------

## Disclaimer

---

The information, documentation and figures available in this deliverable are written by the 5GENESIS Consortium partners under EC co-financing (project H2020-ICT-815178) and do not necessarily reflect the view of the European Commission.

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The reader uses the information at his/her sole risk and liability.

## Copyright

---

Copyright © 2021 the 5GENESIS Consortium. All rights reserved.

The 5GENESIS Consortium consists of:

NATIONAL CENTER FOR SCIENTIFIC RESEARCH “DEMOKRITOS”	Greece
AIRBUS DS SLC	France
ATHONET SRL	Italy
ATOS SPAIN SA	Spain
AVANTI HYLAS 2 CYPRUS LIMITED	Cyprus
AYUNTAMIENTO DE MALAGA	Spain
COSMOTE KINITES TILEPIKOINONIES AE	Greece
EURECOM	France
FOGUS INNOVATIONS & SERVICES P.C.	Greece
FON TECHNOLOGY SL	Spain
FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Germany
IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK	Germany
INFOLYSIS P.C.	Greece
INSTITUTO DE TELECOMUNICACOES	Portugal
INTEL DEUTSCHLAND GMBH	Germany
KARLSTADS UNIVERSITET	Sweden
L.M. ERICSSON LIMITED	Ireland
MARAN (UK) LIMITED	UK
MUNICIPALITY OF EGALEO	Greece
NEMERGENT SOLUTIONS S.L.	Spain
ONEACCESS	France
PRIMETEL PLC	Cyprus
RUNEL NGMT LTD	Israel
SIMULA RESEARCH LABORATORY AS	Norway
SPACE HELLAS (CYPRUS) LTD	Cyprus
TELEFONICA INVESTIGACION Y DESARROLLO SA	Spain
UNIVERSIDAD DE MALAGA	Spain
UNIVERSITAT POLITECNICA DE VALENCIA	Spain
UNIVERSITY OF SURREY	UK

This document may not be copied, reproduced or modified in whole or in part for any purpose without written permission from the 5GENESIS Consortium. In addition to such written permission to copy, reproduce or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

## Version History

---

Rev. N	Description	Author	Date
1.0	Release of D5.2	T. Sarlas (NCSRD)	21/09/2021

## LIST OF ACRONYMS

---

Acronym	Meaning
ADB	Android Debug Bridge
API	Application Programming Interface
ATDD	Acceptance Test-Driven Development
CRUD	create, read, update and delete
E2E	End To End
ELCM	Experiment Lifecycle Manager
EMS	Element Management System
ESXI	Elastic Sky X Integrated
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
ICT	Information & Communications Technologies
KPI	Key Performance Indicator
MANO	Management and Orchestration
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NSD	Network Service Descriptor
NSI	Network Slice Instance
NSR	NS Record
OS	Operating System
RAN	Radio Access Network
RAT	Radio Access Technology
RC	Release Candidate
REST	Representational State Transfer
SCP	Secure Copy Protocol
SSH	Secure Shell
SW	Software
TAP	Test Automation Platform
UE	User Equipment
VIM	Virtual Infrastructure Manager
VNFD	Virtual Functions Descriptor
VNFR	VNF Record
VPN	Virtual Private Network
WIM	WAN Infrastructure Manager
WP	Work Package
Git	Global Information Tracker

## Executive Summary

---

This deliverable presents the WP5 activities on the integration and testing of the Coordination Layer and the Slice Manager components of the 5GENESIS Facility, and the respective testing towards the validation of the 'Release B' of WP3 components implementations for the Open5GENESIS Suite. The document serves as an update of the previous deliverable that provided the integration and validation for 'Release A' software components of the Open5GENESIS Suite.

To this end, for this document to be self-contained, a summary on the 5GENESIS integration workflow is presented focusing on the three environments used for the development and integration activities. The three environments are the development environment, the integration environment hosted by Athens Platform and finally the staging environment realised in each platform (i.e Athens, Malaga, Berlin, Limassol and Surrey). The integration of the individual components follows a Git based methodology that is used to determine the component versions to be integrated, the verified releases for Platforms' integration, as well as, to offer a systematic channel to provide feedback on the development process. In summary, during the final phase of this work package the implemented components (by WP3) were tagged as final releases and integration tests were conducted. Whenever a bug or a missing functionality was discovered, additional bug fixes were committed. During this phase the integration environment was used. When all components were successful through the integration phase the code was packaged and released for installation at each staging environment.

This deliverable also includes a brief discussion of the new features implemented by the individual components which are called *Open5GENESIS Suite*. The suite comprises of: i) the Experiment Lifecycle Manager, for the overseeing of the experiment, ii) the Monitoring and Analytics Framework for the analysis of the raw data collected during an experiment, iii) the Portal, which provides the main interface to the experimenters, iv) the Slice Manager and v) the Dispatcher an abstraction layer that exposes the OpenAPI and forwards incoming requests to the required service.

This deliverable updates the pool of integration tests that are used for the integration validation of the Open5GENESIS Suite components. The integration tests used in the previous release (Rel. 1) were updated to cover enhancements of the under-validation components. In addition, new tests were defined to cover new implemented functionalities.

The deliverable provides also a summary of the integration test that were run in each staging environment (i.e. 5GENESIS platforms). Each platform re-run the integration tests in its own premises to validate proper operation and integration with specific infrastructure components. Depending on the infrastructure readiness or additional required components configuration all the platforms successfully executed most of the specified tests. In the Annex that accompanies the current document, the reader may find additional execution information and proofs of operation.

Concluding, the result of this deliverable demonstrates the readiness of all 5GENESIS platforms to continue with the execution of KPI validation experiments and the showcasing of the defined Use Cases. The integrations tests employed for the validation of the Open5GENESIS suite are

available in a public repository in order to allow third parties to install, validate and use the suite.

# Table of Contents

---

<b>LIST OF ACRONYMS .....</b>	<b>6</b>
<b>LIST OF FIGURES .....</b>	<b>11</b>
<b>LIST OF TABLES.....</b>	<b>14</b>
<b>1. INTRODUCTION.....</b>	<b>15</b>
1.1. Purpose of the document.....	15
1.2. Structure of the document.....	15
1.3. Target audience.....	16
<b>2. SUMMARY OF VERIFICATION METHODOLOGY .....</b>	<b>17</b>
2.1. Integration and Validation.....	17
2.2. Git-based Approach for Component Integration .....	18
2.2.2. Component Releases .....	18
2.2.3. Semantic Versioning .....	19
2.2.4. Delivery and Deployment of Releases .....	19
2.2.5. GitHub Repository.....	19
<b>3. OPEN5GENESIS SUITE RELEASE B .....</b>	<b>22</b>
3.1. 5GENESIS FACILITY Release B Features.....	22
<b>4. INTEGRATION ENVIRONMENT.....</b>	<b>25</b>
<b>5. TESTING AND VALIDATION PROCESS .....</b>	<b>26</b>
5.1. Test 02-01: ELCM Dashboard .....	28
5.2. Test-02-02: Basic facility configuration .....	28
5.3. Test-02-03: Experiment execution .....	29
5.4. Test-03-01: Dispatcher installation .....	30
5.5. Test-04-01: Portal connectivity with other components.....	31
5.6. Test-04-02: Experiment execution through the Portal .....	32
5.7. Test-04-03: Network service onboarding .....	33
5.8. Test-04-04: Network service deployment .....	35
5.9. Test-05-01: ELCM-InfluxDB integration .....	36
5.10. Test-05-02: Distributed experiment .....	37
5.11. Test-06-01: TAP-MONROE configuration .....	38
5.12. Test-06-02: TAP-Remote Ping agent.....	39
5.13. Test-06-03: TAP-Remote iPerf agent .....	40
5.14. Test-06-04: TAP-ADB Ping agent.....	41

5.15. Test-06-05: TAP-ADB iPerf agent .....	42
5.16. Test-06-06: TAP-ADB Resource agent.....	43
<b>6. TESTING AND VALIDATION RESULTS.....</b>	<b>44</b>
6.1. Athens Platform .....	44
6.2. Berlin Platform .....	45
6.3. Limassol Platform .....	46
6.4. Malaga Platform.....	46
6.5. Surrey Platform .....	47
<b>7. CONCLUSIONS .....</b>	<b>49</b>
<b>8. REFERENCES .....</b>	<b>50</b>
<b>ANNEX 1: ATHENS PLATFORM INTEGRATION ENVIRONMENT .....</b>	<b>51</b>
<b>ANNEX 2: ATHENS PLATFORM INTEGRATION ACTIVITIES .....</b>	<b>53</b>
<b>ANNEX 3: BERLIN PLATFORM INTEGRATION ACTIVITIES .....</b>	<b>71</b>
<b>ANNEX 4: LIMASSOL PLATFORM INTEGRATION ACTIVITIES.....</b>	<b>82</b>
<b>ANNEX 5: MALAGA PLATFORM INTEGRATION ACTIVITIES.....</b>	<b>88</b>
<b>ANNEX 6: SURREY PLATFORM INTEGRATION ACTIVITIES.....</b>	<b>97</b>

## LIST OF FIGURES

---

Figure 1. 5GENESIS development and integration workflow .....	17
Figure 2 Integration GitHub Repository .....	20
Figure 3 Release B Integration Project Board .....	20
Figure 4 Component Example Repository .....	21
Figure 5 OpenStack Integration Environment .....	25
Figure 6 ESXi Integration Environment.....	25
Figure 7 OpenStack Network .....	51
Figure 8 ELCM Dashboard (Athens).....	53
Figure 9 ELCM Basic Facility Config (Athens) .....	53
Figure 10 ELCM Experiment Execution (Athens) .....	53
Figure 11 Dispatcher Unit Test Results (Athens) .....	54
Figure 12 Portal User Registration Form (Athens).....	54
Figure 13 User Activation Email (Athens).....	55
Figure 14 Simple Experiment Creation (Athens).....	55
Figure 15 Simple Experiment Execution Logs (Athens) .....	56
Figure 16 NS Image Onboarding (Athens).....	56
Figure 17 NS and VNF Onboarding (Athens).....	57
Figure 18 Experiment Creation with Slice (Athens) .....	57
Figure 19 Slice Manager CLI at Instantiation (Athens) .....	58
Figure 20 Slice Instances OpenStack (Athens) .....	58
Figure 21 Portal Execution Log for Slice Instantiation (Athens).....	58
Figure 22 InfluxDB Recorded Results (Athens) .....	59
Figure 23 Monroe Experiment Results in InfluxDB (Athens).....	59
Figure 24 Linux Ping Agent Console (Athens) .....	60
Figure 25 Linux Ping Agent Results (Athens).....	62
Figure 26 Linux iPerf Agent Console (Athens).....	62
Figure 27 Linux iPerf Agent Results (Athens) .....	63
Figure 28 Adb Ping Agent Results (Athens).....	65
Figure 29 Adb iPerf Agent TapPlan (Athens).....	65
Figure 30 Adb iPerf Agent Results (Athens) .....	67
Figure 31 Resource Agent Results (Athens) .....	70

Figure 32 ELCM Dashboard (Test-02-01) .....	71
Figure 33 Basic Facility Configuration: Scenario (JSON, Test-02-02) .....	72
Figure 34 Basic Facility Configuration: Test Case (JSON, Test-02-02).....	72
Figure 35 Basic Facility Configuration: UE (JSON, Test-02-02).....	73
Figure 36 Returned Experiment result, as JSON (Test 02-03).....	73
Figure 37 Robot Framework Test Result Report (Test-03-01) .....	74
Figure 38 Portal GUI with successfully registered user (Test-04-01).....	75
Figure 39 Portal shows executed results of Experiment 0 (Test-04-02).....	76
Figure 40 Network Service Onboarding via Portal (Test-04-03) .....	77
Figure 41 Berlin Platform Portal: Creation of a Network Service for Test-04-04.....	78
Figure 42 Berlin Platform Portal: Execution of a sliced experiment (Test-04-04).....	79
Figure 43 Berlin Platform Portal: Execution result for ELCM-Influx test (Test-05-01).....	80
Figure 44 - ELCM Dashboard.....	82
Figure 45 - ELCM logs after adding Testcase, UE, Scenario to ELCM facility.....	82
Figure 46 - ELCM logs during experiment execution.....	83
Figure 47 - InfluxDB entries sent from ELCM .....	83
Figure 48 - Portal empty dashboard after user login.....	83
Figure 49 - Simple experiment creation .....	84
Figure 50 - Simple experiment run logs .....	84
Figure 51 - Network service onboarding dashboard .....	85
Figure 52 - MONROE TAP test plan .....	86
Figure 53 - MONROE test plan entries in Influx DB .....	86
Figure 54 - Ping TAP test plan .....	87
Figure 55 - iPerf TAP test plan .....	87
Figure 56. Initial diagnostics on first run .....	88
Figure 57. Facility logs and endpoint responses after the test .....	89
Figure 58. Logs of the test execution (Run stage only) .....	89
Figure 59. Available entities in the Portal and initial user dashboard .....	89
Figure 60. Experiment execution results.....	90
Figure 61. MONROE agent result generation messages on TAP Log.....	91
Figure 62. Ping agent result generation messages on TAP Log. ....	92
Figure 63. iPerf agent result generation messages on TAP Log.....	93
Figure 64. ADB Ping result generation messages on TAP Log.....	93

Figure 65. ADB iPerf result generation messages on TAP Log. ....	94
Figure 66. Resource agent result generation messages on TAP Log. ....	95
Figure 67 Surrey Platform ELCM Dashboard.....	97
Figure 68 Facility log.....	98
Figure 69 ELCM Dashboard during experiment execution .....	98
Figure 70 Experiment Execution.....	99
Figure 71 Dispatcher Swagger API following the addition of a new user.....	100
Figure 72 Dispatcher log .....	100
Figure 73 Addition of new user.....	101
Figure 74 Portal config.yml file .....	101
Figure 75 Portal initial user dashboard .....	101
Figure 76 Experiment execution through the portal.....	102
Figure 77 InfluxDB showing test results .....	103
Figure 78 Remote Ping agent execution .....	104
Figure 79 Linux iPerf Agent Console.....	104
Figure 80 OpenTAP iperf test plan execution .....	105
Figure 81 Remote iPerf agent execution.....	105

## LIST OF TABLES

---

Table 1. Test Case Template .....	26
Table 2. 5GENESIS Release B integration tests .....	26
Table 3. 5GENESIS Release B Inter-component tests .....	27
Table 4. 5GENESIS Release B Probes tests .....	27
Table 5 Test 02-01, ELCM Dashboard.....	28
Table 6 Test 02-02, Basic Facility Config .....	28
Table 7 Test 02-03, Experiment Execution.....	29
Table 8 Test 03-01, Dispatcher Installation.....	30
Table 9 Test 04-01, Portal Connectivity with Other Components .....	31
Table 10 Test 04-02, Experiment Execution through the Portal.....	32
Table 11 Test 04-03, Network Service Onboarding .....	33
Table 12 Test 04-04, Network Service Deployment .....	35
Table 13 Test 05-01, ELCM-InfluxdB Integration .....	36
Table 14 Test 05-02, Distributed Experiment .....	37
Table 15 Test 06-01, TAP-Monroe Configuration .....	38
Table 16 Test 06-02, TAP-Remote Ping Agent .....	39
Table 17 Test 06-03, TAP-Remote iPerf Agent.....	40
Table 18 Test 06-04, TAP-ADB Ping Agent .....	41
Table 19 Test 06-05, TAP-ADB iPerf Agent.....	42
Table 20 Test 06-06, TAP-ADB Resource Agent .....	43
Table 21 Athens Platform Integration Results .....	44
Table 22 Berlin Platform Integration Results .....	45
Table 23 Limassol Platform Integration Results.....	46
Table 24 Malaga Platform Integration Results.....	46
Table 25 Surrey Platform Integration Results .....	47
Table 26 Integration Components.....	52

# 1. INTRODUCTION

---

## 1.1. Purpose of the document

The 5GENESIS main objective is to provide to 5G experimenters and verticals' developers and testers access to 5G experimentation facilities capable of conducting automated testing and analysis on top of state-of-the-art 5G infrastructure supporting various configurations and deployment topologies. In this frame 5GENESIS project designed and implemented the Open5GENESIS Suite, an infrastructure agnostic software suite that is layered on top of the infrastructure and provides all the means and tools to achieve test design for selection of KPIs to be validated and testbed setup (i.e., radio configuration, network slice setup etc), automated test execution and analytics retrieval and finally raw measurement acquirement.

This Work Package (WP5) and specifically Task 5.1 and Task 5.2, is responsible for overseeing and implementing the integration validation process, providing development, integration and staging environments, respective workflows as well as specifically targeted integration tests. This deliverable is providing a delta from the previous deliverable and discusses the evolution of the Open5GENESIS Suite integration using the latest (Rel. B) components. The individual components, developed in the context of WP3, are collected from all public GitHub repositories and are installed, tested, and integrated in a controlled integration and testing environment in provided by Athens Platform.

As a result of the integration testing over the integration environment, several bugs and misbehaviours were identified and fixed, resulting in a bug free software suite release. As a next step the packaged software release was installed in each platform considered as a staging environment. For each platform this document presents the results of the integration tests and provides integration details specifically for each platform. The more detailed parts are available in the Annex section of this deliverable.

Part of the effort for this deliverable was the definition of integration tests. The software as well as the test and the documentation that are used to validate the integration are hosted in a GitHub repository and can be used by anyone who would be interested deploying Open5GENESIS Suite over his 5G infrastructure. A separate User Guide document is released [1] with the aim to allow externals to the project to deploy and used the Open5GENESIS Suite.

## 1.2. Structure of the document

This deliverable is structured as follows:

- Section 2 describes the 3-phased verification methodology adopted for the final successful integration of the Coordination Layer and Slice Manager components in each 5GENESIS Platform. Specifically, the process workflows have been established and best practice guides are outlined.
- Section 3 provides a description of the 5GENESIS Coordination Layer and Slice Manager, by introducing its main features and components.
- Section 4 describes the dedicated integration and testing environment that was created on the Athens Platform, in order to install, integrate and test the Release B components.

- Section 5 defines the tests cases that were executed in the context of integration and validation activities.
- Section 6 provides a short summary of test case execution results per platform.
- Section 7 provides concluding remarks
- Annex 1 contains additional information about the integration environment that was created in Athens Platform.
- Finally, Annexes 2-6 provide detailed information about the integration and verification activities and test cases executions that took place in each 5Genesis Platform.

### 1.3. Target audience

The target audience of this deliverable includes the ICT professionals or research projects who are interested in performing experimentations, the European Commission, who can use this document as a means for the evaluation of the activities of the Platform with regards to the project objectives, as well as the 5GENESIS consortium, who can use it as a guide and reference regarding future activities.

## 2. SUMMARY OF VERIFICATION METHODOLOGY

This chapter presents the WP5 approach on the integration activities that result in a homogeneous, interoperable software framework (Coordination Layer plus Slice Manager) that is being deployed in each 5GENESIS Platform. The objective of this chapter is to present the basic operations and workflows that need to be realized in order to deliver the integrated 5GENESIS Coordination Layer as soon as each development phase concludes. In this context, WP5 defines the software development workflows, the semantics for designating each component's source code status and the coordination between the different developers in order to deliver the pre-integration source code. Moreover, WP5 is responsible to collect the components from all repositories and provide a full and finite 5GENESIS Release, ready to be on boarded per Platform.

### 2.1. Integration and Validation

This paragraph presents the workflow adopted by WP5 in order to support the component integration activities, validate the integration and provide system level testing. The workflow is presented in Figure 1.

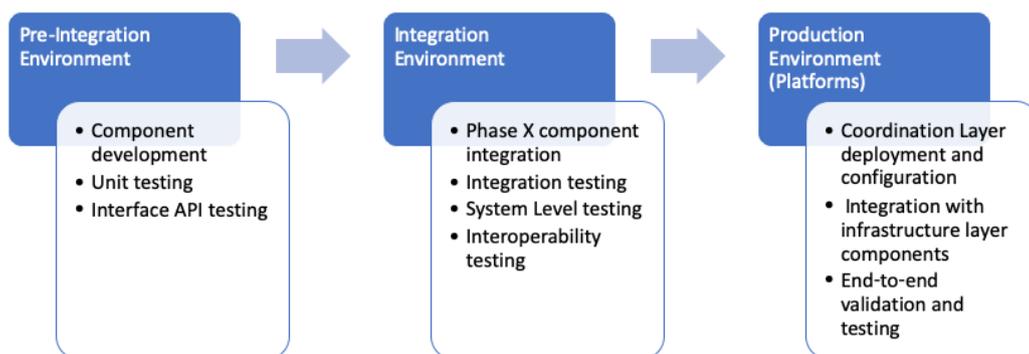


Figure 1. 5GENESIS development and integration workflow

Three phases are considered in 5GENESIS, starting from the development of the individual components, towards their deployment in the respective 5GENESIS Platforms in order to create the 5GENESIS Facility, namely (i) the development phase; (ii) integration phase and (iii) the final deployment phase. Each of these phases is supported and executed in its respective environment. Initially the developers use their own *development environment* (i.e., Pre-integration environment) to develop the components. In this environment, Infrastructure (sandbox environments available at 5GENESIS Platforms) and software tools (e.g., GitHub) are exploited for development and manual functional tests. It is expected that unit tests are executed in this environment. According to the project workplan, each component that is being developed in each separate repository is designated as candidate for release. It is important to note that the project specifies 3 phases that correspond to the deployment of Coordination Layer and Slice Manager releases as well as integration with infrastructure elements. The integration phase starts when the software components are tagged and made available. This

phase is supported by the *Integration Environment*. This environment is created in one of the Platforms and supports computing and network resources exploiting virtualization capabilities available at the Platforms. During this phase, for each component as well as for the whole Coordination Layer, the following actions are performed:

- Deployment and configuration is done according to the documentation/deployment scripts that are available by the developers
- Interoperability tests between peering components are executed
- Integration validation according to well-defined integration tests is executed
- System level tests are executed.
- Documentation and configuration are updated according to the integration findings, fixing omissions and pre-requisites.

When the component(s) integration phase ends successfully, the integrated code versions are tagged as main release(s) and the software is ready to be deployed at their final destination (i.e., the 5GENESIS Platforms). The environment that supports this activity is specific to each Platform, as different infrastructure elements or virtualization technologies may be utilized in each Platform. Each component is deployed using the updated documentation and configuration guidelines provided by WP5. This approach is validated using the integration tests that have been defined by WP5 during the integration phase. Test cases are defined in Section 5.

## 2.2. Git-based Approach for Component Integration

A Git-based approach was introduced to address the integration of the WP3 components i.e., releases, hotfixes and feature enhancements.

The proposed methodology uses the best practices currently employed in software development. The three-pronged approach involves:

1. **Release** - Provides a consistent and well-defined approach that adopts the Git's master/develop/release workflow,
2. **Version** - a common agreed upon semantic versioning scheme,
3. **Deploy** - Provides an installation script that installs in a single step the component on top of a plain OS.

This subsection presents a brief description of this approach, followed by a presentation of the way it was adopted during Rel.B integration activities. A more detailed description of the methodology can be found in Section 2 of deliverable D5.1 [2].

### 2.2.2. Component Releases

Software development is a continuous process and even after a component/software module is released for integration or production, the component is not in its final state in terms of feature development. When a component is said to be released, it only implies that a certain subset of features / requirements that been agreed during the start of the release cycle have been implemented and fulfilled.

New development activities for the component commence at the start of a new release cycle. However, while the new release cycle is ongoing, bugs are invariably discovered on the (previous) released version and fixes for the same must be provided to improve the stability of the release. Git branches provide a clean solution to separate development efforts from bug fixes.

### 2.2.3. Semantic Versioning

Software exists in different versions and developers use versioning to communicate information about their software. Information conveyed during versioning may involve one or more of the following:

1. Time of creation
2. Features
3. Compatibility
4. Target Architecture

### 2.2.4. Delivery and Deployment of Releases

In the context of 5GENESIS, the delivery of every Release and Release Candidate includes an installation script that installs in a single step the delivered component on top of a plain Operating System (OS) (e.g., Ubuntu 18.04 LTS). The installation script can be provided either as an:

1. Shell script, or
2. Ansible<sup>1</sup>

The installation script would be responsible for the deployment and the configuration of the individual components. The integrators (WP5) would then work on bringing the various components together.

### 2.2.5. GitHub Repository

To realise the testing and integration methodology described in this section, WP5 activities were coordinated using a GitHub repository. The repository was created inside the project's GitHub organisation with visibility set to private to avoid exposing sensitive project information. Inside are stored the descriptions of integration tests with all required additional files i.e. configuration files, descriptors, result logs that are needed for each test. Documentation and installation guidelines of SW components are collected and summarised inside a single documentation file for the integrators to use.

The selection of GitHub repository as the tool to manage the integration process, was decided for the following reasons:

- WP3 software components development, documentation and releases were already stored in repositories inside 5Genesis organisation which allowed for the creation of a complete and organised integration environment.

---

<sup>1</sup> Ansible - [https://docs.ansible.com/ansible/latest/user\\_guide/intro\\_getting\\_started.html](https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html)

- The project management features offered by the platform were utilized to coordinate the integration tasks as they described inside the WP5 methodology. Figure 3 shows the Kanban project board that was used for overseeing integration activities. Each test case comes with each own deployment task in the project management table and relates to an issue ticket. In addition, tabs that summarize the integration progress for each platform are created.

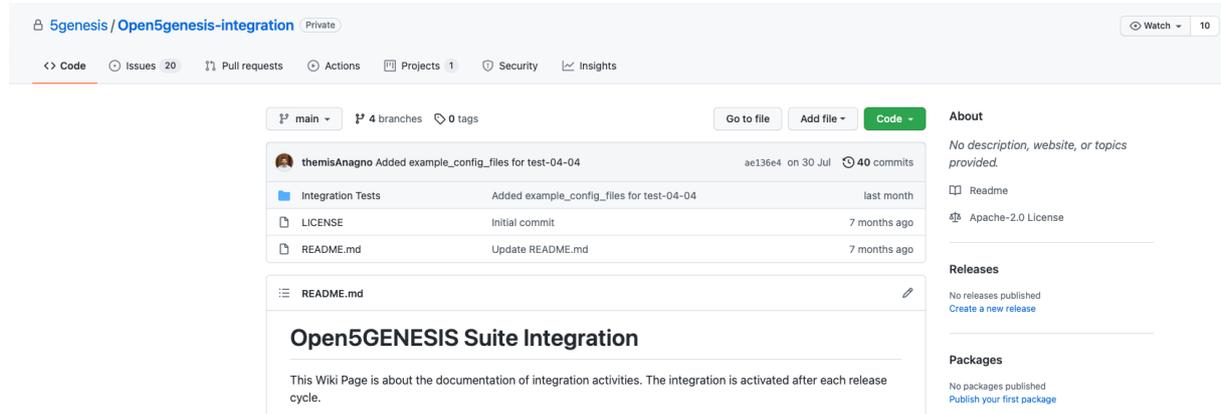


Figure 2 Integration GitHub Repository

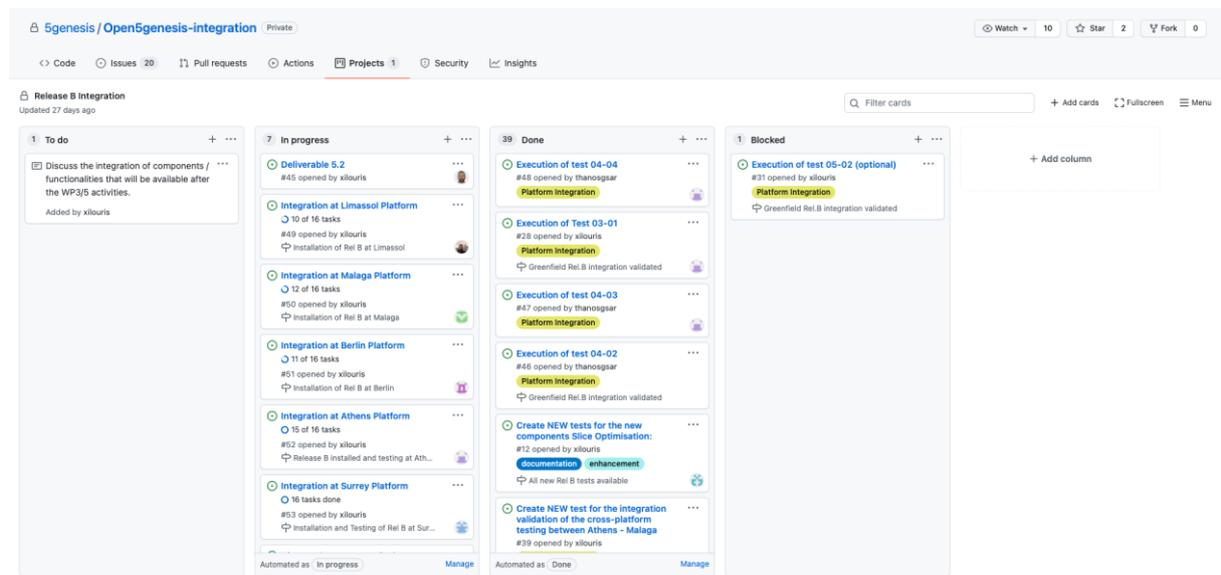
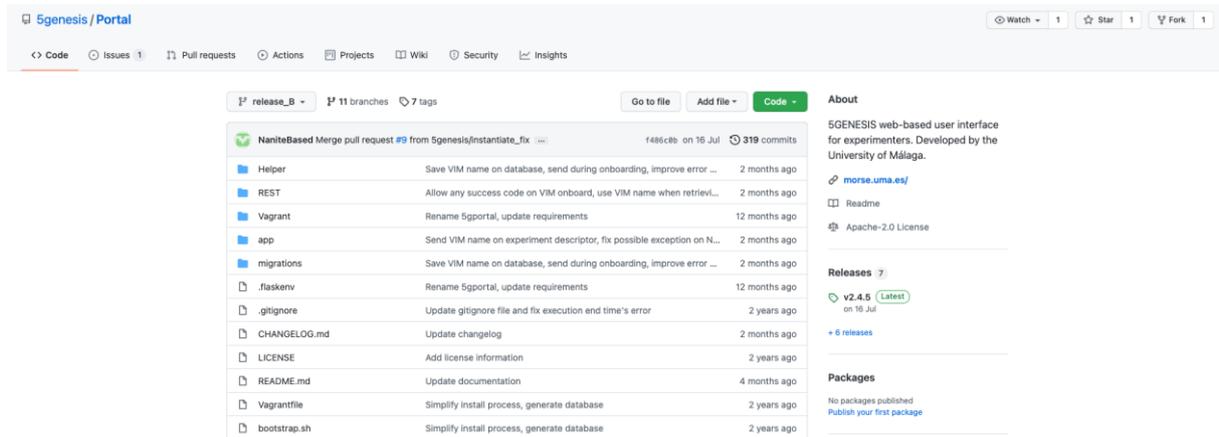


Figure 3 Release B Integration Project Board

In case of a bug, feedback can be provided back to the developers in the form of issue tickets. Each ticket can also facilitate a discussion between involved integrators and developers to enhance collaboration, and is also visible to every organization member for future use.

Software release delivery and versioning is also addressed by using the releases feature. The Open5Genesis platform portal development repository is presented as an example in Figure 4. For each software component, after changes have been applied the developers create a new release package based on the semantic versioning methodology described in this section.

Integrators and anyone planning to install the Open5Genesis Rel. B components is able to, by using the release feature and choosing the latest one.



The screenshot displays the GitHub interface for the repository `5genesis/Portal`. The repository is in the `release_B` branch, which has 11 branches and 7 tags. The repository statistics show 1486 commits, last updated on 16 Jul, with 319 total commits. The file structure is as follows:

File	Description	Last Commit
Helper	Save VIM name on database, send during onboarding, improve error ...	2 months ago
REST	Allow any success code on VIM onboard, use VIM name when retriev...	2 months ago
Vagrant	Rename 5gportal, update requirements	12 months ago
app	Send VIM name on experiment descriptor, fix possible exception on N...	2 months ago
migrations	Save VIM name on database, send during onboarding, improve error ...	2 months ago
.flaskenv	Rename 5gportal, update requirements	12 months ago
.gitignore	Update gitignore file and fix execution end time's error	2 years ago
CHANGELOG.md	Update changelog	2 months ago
LICENSE	Add license information	2 years ago
README.md	Update documentation	4 months ago
Vagrantfile	Simplify install process, generate database	2 years ago
bootstrap.sh	Simplify install process, generate database	2 years ago

The right-hand side of the interface provides an **About** section, stating that 5GENESIS is a web-based user interface for experimenters, developed by the University of Málaga, with a link to `morse.uma.es`. It also lists the license as Apache-2.0. The **Releases** section shows the current version as `v2.4.5`, marked as the latest release, with 6 total releases. The **Packages** section indicates that no packages have been published yet.

Figure 4 Component Example Repository

## 3. OPEN5GENESIS SUITE RELEASE B

---

### 3.1. 5GENESIS FACILITY Release B Features

The 5GENESIS Facility is the entry point for experimenters who wish to make use of the Platforms for the execution of their experiments. The Release B of the Coordination Layer provides:

- A web Portal that allows the definition of experiments that can be executed in the Platform, and the visualization of the most important results of an execution.
- The automatic control of the life cycle of such experiments by the Experiment Lifecycle Manager (ELCM), described in D3.16 [3].
- Communication between the Portal and the Experiment Lifecycle Manager (ELCM) via the OpenAPI implemented by the Dispatcher component described in D3.8 [4].
- The long-term storage of the results generated by the experiments.
- Automated communication with the Slice Manager (described in D3.4 [5]) and the lower layers for the configuration of probes and instruments required for the execution of experiments.
- An Analytics framework capable of providing data-driven insights that is described in D3.6 [6].
- The possibility of executing distributed experiments that are coordinated between two different 5Genesis platforms.

Based on the Open5GENESIS architecture, the experimenter/vertical has two options for performing an experiment:

- Through the 5GENESIS Portal which provides a graphical user interface for the creation of experiment descriptors and access to graphical representations of the generated results, making it ideal for E2E KPI assessment.
- Directly via the 5GENESIS OpenAPI, allowing the experimenter to use the facility with its own automation scripts, making it ideal for the fast validation of a new component or service.

The Dispatcher receives the experiment descriptor from the Portal or through the Open API, initiates the validation of the descriptor and sends the execution request to the ELCM's scheduler, which enqueues the execution until all necessary resources are available. Once the Management and Orchestration Layer confirms that the required resources are available then the execution of the experiment starts.

Upon availability of the resources the Slice Manager creates the requested E2E network slice instance allowing the multi-tenant use of the facility by different experimenters. The created network slice instance crosses all the components of infrastructure, starting from the Core NFVI, the transport network, the Edge, the RAT and finally the UEs.

The scope of interfaces and components covered in this report are the Portal, ELCM, Dispatcher, Slice Manger and Analytics.

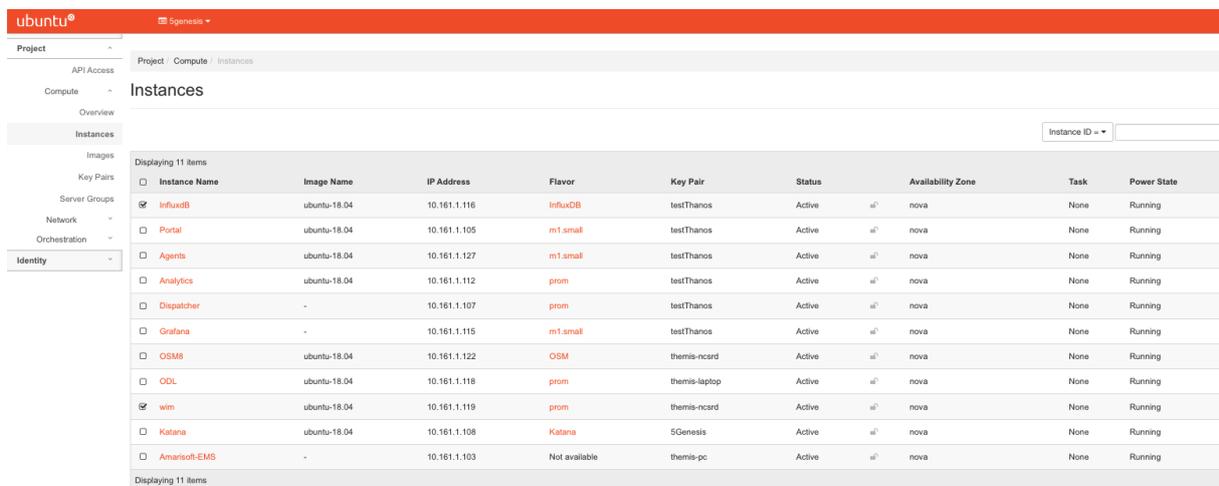
The Coordination layer is defined in more detail on Section 3 of D2.4 [7]. The following SW components have been developed as part of WP3:

- **ELCM:** The Experiment Lifecycle Manager is the entity that oversees the execution of an experiment from the start until the end of the experiment. The ELCM is able to receive execution requests validated by the Dispatcher in the form of the experiment descriptor and is able to perform the execution of multiple experiments in parallel. By interacting with the Slice Manager's REST API, the ELCM is able to instantiate the network services required by the experiment, and decommission them once the execution finishes, freeing the resources for other experiments. More information about the development and functionality of this component can be seen in D3.16 [3].
- **Monitoring and Analytics:** The monitoring and analytics framework perform the collection and analysis of the raw experimental data generated during an experiment execution and collected by dedicated monitoring probes. During Release B, a new Analytics dashboard has been developed and integrated. Features include 5G KPI validation and prediction, statistical analysis and time series management. More information about the framework is available in D3.6 [8].
- **Portal:** The Portal provides a Web-based user interface that experimenters interact with in order to define and execute experiments in the Platforms. The Portal also allows experimenters to view a selection of the most relevant results generated by their experiments in the form of custom Grafana dashboards and gives access to the Analytics Dashboard for an in-depth analysis of the results. The Portal interacts with ELCM via the Open API provided by the Dispatcher. More information about the Portal can be seen in Section 7 of D3.8 [4].
- **Dispatcher:** The Dispatcher is the component engine that exposes the 5Genesis platform Open API and redirects incoming requests to the required service in the infrastructure. Dispatcher's REST API modules include the Authenticator that provides users secure access to platform and handles all registration procedures, the MANO wrapper that manages and indexes VIM and NFVO related resources and ELCM interfaces to oversee and initiate experiment execution. More information about the Dispatcher module can be found in D3.8 [4].
- **Slice Manager:** In 5GENESIS, the Slice Manager is a standalone component responsible for performing life cycle management of the End-to-End network slices and monitoring the network slices utilization. It is implemented as part of the 5GENESIS Coordination Layer and is deployed in all 5GENESIS Platforms. The Slice Manager is developed in the scope of the WP3 activities, it is an open-source project under the Apache 2 license, and Release B is described in D3.4 [5]. Following the "Study on management and orchestration of network slicing for next-generation network" [9], a Network Slice Instance (NSI) is a managed entity which can be described as the sum of various sub-slices of different network domains, such as the Radio Access Network (RAN), the transport network, the Core Cloud and the Edge Cloud. The NSI contains the network functions that offer all the functionalities and resources necessary to support a particular set of communication services.
- The 5GENESIS Slice Manager is based on a highly modular architecture, built as a mesh of microservices integrated and working collectively to offer slice management services. Each microservice is a module running as a Docker container, forming the Slice Manager's software stack. The key advantages of this architectural approach are that it offers simplicity in building and maintaining applications, flexibility, and scalability. In addition, the containerized approach makes the applications independent of the

underlying system. Slice Manager provides the necessary set of interfaces and communication mechanisms that allow the integration and interconnection with the other components of the 5GENESIS architecture.

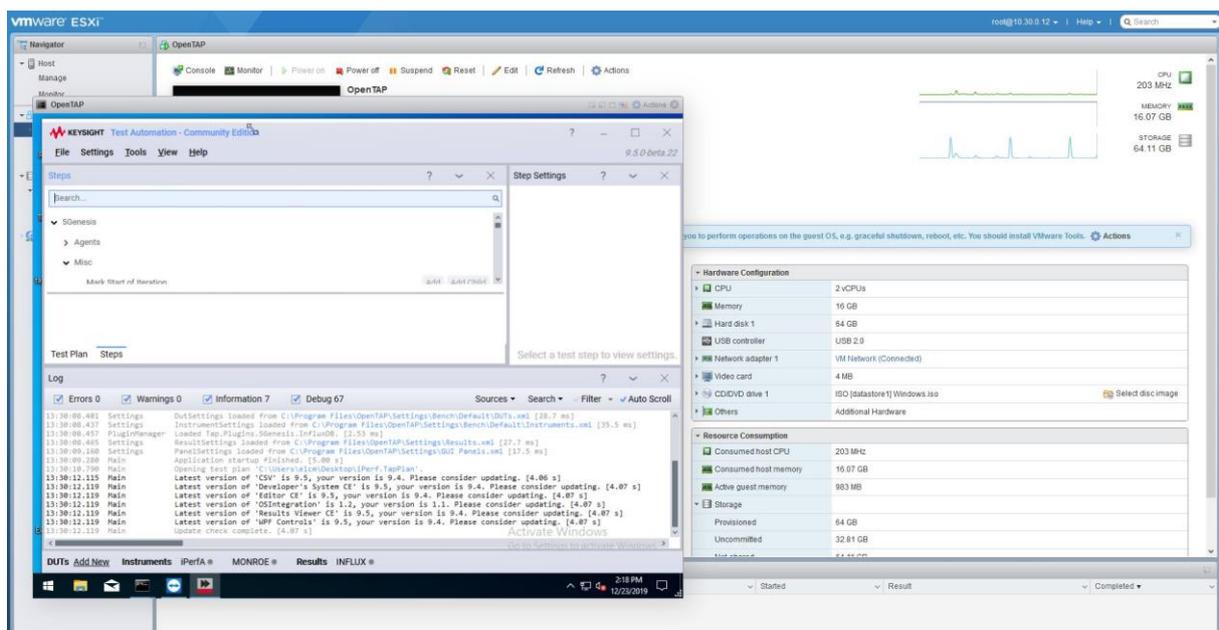
## 4. INTEGRATION ENVIRONMENT

A dedicated integration and testing environment created on the Athens Platform is used for installation, testing and integrating the ‘Release B’ of the WP3 components, which will be part of the Open5GENESIS Facility. It is recommended that a dedicated testing environment is created by all the platforms to facilitate reproducibility of the integration before deployment in the production platforms. The testing environment in the Athens Platform is comprised of an OpenStack cloud, where all the Linux-based components are hosted, and a VMWare ESXI<sup>2</sup>, where all the windows-based components are hosted, as depicted in Figure 5 and Figure 6 respectively. Further details can be found in Annex 1: Athens Platform Integration Environment.



Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State
InfluxDB	ubuntu-18.04	10.161.1.116	InfluxDB	testThanos	Active	nova	None	Running
Portal	ubuntu-18.04	10.161.1.105	m1.small	testThanos	Active	nova	None	Running
Agents	ubuntu-18.04	10.161.1.127	m1.small	testThanos	Active	nova	None	Running
Analytics	ubuntu-18.04	10.161.1.112	prom	testThanos	Active	nova	None	Running
Dispatcher	-	10.161.1.107	prom	testThanos	Active	nova	None	Running
Grafana	-	10.161.1.115	m1.small	testThanos	Active	nova	None	Running
OSM	ubuntu-18.04	10.161.1.122	OSM	themis-ncard	Active	nova	None	Running
ODL	ubuntu-18.04	10.161.1.118	prom	themis-laptop	Active	nova	None	Running
wim	ubuntu-18.04	10.161.1.119	prom	themis-ncard	Active	nova	None	Running
Katana	ubuntu-18.04	10.161.1.108	Katana	5Genesis	Active	nova	None	Running
Amansoft-EMS	-	10.161.1.103	Not available	themis-pc	Active	nova	None	Running

Figure 5 OpenStack Integration Environment



The screenshot displays the VMware ESXi management console. The main window shows the 'OpenTAP' test automation tool interface, which includes a 'Steps' pane on the left, a 'Log' pane at the bottom, and a 'Hardware Configuration' pane on the right. The log shows the following output:

```

13:30:08.481 Settings OutSettings loaded from C:\Program Files\OpenTAP\Settings\Bench\Default\OUTs.xml [28.7 ms]
13:30:08.487 Settings InstrumentSettings loaded from C:\Program Files\OpenTAP\Settings\Bench\Default\Instruments.xml [35.5 ms]
13:30:08.497 PluginManager Loaded Tap.Plugins.Stonix.InfluxDB [12.53 ms]
13:30:08.465 Settings ResultSettings loaded from C:\Program Files\OpenTAP\Settings\Results.xml [27.7 ms]
13:30:09.200 Settings PanelSettings loaded from C:\Program Files\OpenTAP\Settings\OUT\Panel.s.xml [17.5 ms]
13:30:09.208 Settings Application startup finished. [5.88 s]
13:30:10.096 Main Opening test plan 'C:\Users\test\Documents\OpenTAP\TapPlan'
13:30:12.115 Main Latest version of 'CSV' is 9.5, your version is 9.4. Please consider updating. [4.06 s]
13:30:12.119 Main Latest version of 'Developer's System CE' is 9.5, your version is 9.4. Please consider updating. [4.07 s]
13:30:12.119 Main Latest version of 'Editor CE' is 9.5, your version is 9.4. Please consider updating. [4.07 s]
13:30:12.119 Main Latest version of 'OSIntegration' is 1.2, your version is 1.1. Please consider updating. [4.07 s]
13:30:12.119 Main Latest version of 'Results Viewer CE' is 9.5, your version is 9.4. Please consider updating. [4.07 s]
13:30:12.119 Main Latest version of 'WFF Controls' is 9.5, your version is 9.4. Please consider updating. [4.07 s]
13:30:12.119 Main Update check complete. [4.07 s]
  
```

The hardware configuration pane shows the following details:

- CPU: 2 vCPUs
- Memory: 16 GB
- Hard-disk 1: 64 GB
- USB controller: USB 2.0
- Network adapter 1: VM Network (Connected)
- Video card: 4 MB
- CD/DVD drive 1: ISO (datastore1) Windows.iso

The resource consumption pane shows:

- Consumed host CPU: 203 MHz
- Consumed host memory: 16.07 GB
- Active guest memory: 983 MB
- Storage: Provisioned 64 GB, Uncommitted 32.81 GB

Figure 6 ESXi Integration Environment

<sup>2</sup> ESXi stands for Elastic Sky X Integrated is an enterprise server virtualization platform by VMware.

## 5. TESTING AND VALIDATION PROCESS

Based on previous experience from other projects that worked with virtualized integration environments for 5G and NFV (i.e., 5GTANGO [10], SONATA-NFV [11]etc.) and also from the work of ETSI NFV [12] ,5GENESIS defines a template for the definition of the integration tests that need to be executed in order to validate component integration. Table 1 depicts the template used for the definition of integration tests.

**Table 1. Test Case Template**

Test Case Name			Test Case id	
Test Purpose	<i>Interfaces to be tested</i>			
Configuration	<i>NS to be used, configuration of Infrastructure etc</i>			
Test Tool	<i>Test tools used</i>			
Metric	<i>Measured metrics</i>			
References	<i>e.g., RFC XXX</i>			
Applicability	<i>Components that are applicable for this test</i>			
Pre-test conditions	<i>Monitoring configuration, additional metrics etc</i>			
Test sequence	Step	Description	Result	
	Step	Description	Result	
Test Verdict	<i>Descriptive text here</i>			
Additional Resources	<i>Graphs, etc.</i>			

The integration tests that are developed for Release B are summarized in Table 2 and presented below. The executed tests and their results, following the template above are linked next to each test case. In order to protect information that is confidential to the project consortium, links to private project repositories are removed.

Test case IDs are assigned using the following format: test-xx-yy, where xx is an integer value that is assigned to the general functionality that the test covers, and yy is an integer assigned in order to differentiate test cases that target the same component, but a different (or greater) sub-set of the functionality. For example, test-02-01 specifies the minimal functionality test that affects the ELCM, while in the future we may specify a new test-02-02 that covers some extra functionality added in the next phases of the development.

**Table 2. 5GENESIS Release B integration tests**

Test case id	Test case name	Test case description	Involved components
Test-02-01	ELCM Dashboard	Verify that the ELCM is running and network reachable	<ul style="list-style-type: none"> <li>ELCM</li> </ul>
Test-02-02	Basic facility configuration	Verify that the ELCM correctly reads the Facility Registry	<ul style="list-style-type: none"> <li>ELCM</li> </ul>
Test-02-03	Experiment execution	Verify that the ELCM is able to run experiments	<ul style="list-style-type: none"> <li>ELCM</li> </ul>
Test-03-01	Dispatcher installation	Verify the Dispatcher is correctly installed and configured with the MANO Components of the	<ul style="list-style-type: none"> <li>Dispatcher</li> <li>NFVO</li> <li>VIM</li> </ul>

		platform - Validate all Dispatcher features	
Test-04-01	Portal connectivity with other components	Verify that the Portal is reachable and has connectivity with the Dispatcher and ELCM	<ul style="list-style-type: none"> <li>• Portal</li> <li>• Dispatcher</li> <li>• ELCM</li> </ul>
Test-04-02	Experiment execution through the Portal	Verify that the execution of experiments works correctly through the Portal	<ul style="list-style-type: none"> <li>• Portal</li> <li>• Dispatcher</li> <li>• ELCM</li> </ul>
Test-04-03	Network service onboarding	Verify that the onboarding of network service's artifacts is correctly configured	<ul style="list-style-type: none"> <li>• Portal</li> <li>• Dispatcher</li> <li>• NFVO</li> <li>• VIM</li> </ul>
Test-04-04	Network service deployment	Verify that network services are correctly deployed during an experiment execution	<ul style="list-style-type: none"> <li>• Portal</li> <li>• Dispatcher</li> <li>• NFVO</li> <li>• VIM</li> <li>• ELCM</li> <li>• Slice Manager</li> </ul>

Table 3. 5GENESIS Release B Inter-component tests

Test case id	Test case name	Test case description	Involved components
Test-05-01	ELCM-InfluxDB integration	Verify that the connection between the ELCM and InfluxDB is correctly configured	<ul style="list-style-type: none"> <li>• ELCM</li> <li>• InfluxDB</li> </ul>
Test-05-02	Distributed experiment	Verify that two platforms can execute distributed experiments	On two platforms: <ul style="list-style-type: none"> <li>• Portal</li> <li>• Dispatcher</li> <li>• ELCM</li> <li>• InfluxDB</li> </ul>

Table 4. 5GENESIS Release B Probes tests

Test case id	Test case name	Test case description	Involved components
Test-06-01	TAP-MONROE configuration	This test tests the proper deployment and communication of TAP and a MONROE node	<ul style="list-style-type: none"> <li>• TAP</li> <li>• MONROE</li> </ul>
Test-06-02	TAP-Remote Ping agent	This test tests the proper deployment and communication of TAP and a Remote ping agent	<ul style="list-style-type: none"> <li>• TAP</li> <li>• Remote Ping agent</li> </ul>
Test-06-03	TAP-Remote iPerf agent	This test tests the proper deployment and communication of TAP and a Remote iPerf agent	<ul style="list-style-type: none"> <li>• TAP</li> <li>• Remote iPerf agent</li> </ul>

Test-06-04	TAP-ADB Ping agent	This test tests the proper deployment and communication of TAP and an ADB Ping agent	<ul style="list-style-type: none"> <li>• TAP</li> <li>• ADB ping agent</li> </ul>
Test-06-05	TAP-ADB iPerf agent	This test tests the proper deployment and communication of TAP and an ADB iPerf agent	<ul style="list-style-type: none"> <li>• TAP</li> <li>• ADB iPerf agent</li> </ul>
Test-06-06	TAP-ADB Resource agent	This test tests the proper deployment and communication of TAP and an ADB Resource agent	<ul style="list-style-type: none"> <li>• TAP</li> <li>• ADB Resource agent</li> </ul>

## 5.1. Test 02-01: ELCM Dashboard

Table 5 Test 02-01, ELCM Dashboard

Test Case Name	ELCM dashboard		Test Case id	Test-02-01
Test Purpose	Verify that the ELCM is running and network reachable			
Configuration				
Test Tool	Web browser			
Metric				
References				
Applicability	ELCM			
Pre-test conditions	The ELCM has been deployed and is listening for connections on a known address.			
Test sequence	Step	Connect to the ELCM address with a web browser	The ELCM Dashboard should be visible	
	Step	On the <i>Diagnostics</i> section, expand the Configuration and Facility logs	Several validation messages should be visible	
	Step	Review the contents of the logs, ensuring that no unexpected messages appear. If more information is required, remember that the <i>Debug</i> messages are hidden by default.		
Test Verdict	If no unexpected issues arise, then the ELCM is active and listening for connections			
Additional Resources				

## 5.2. Test-02-02: Basic facility configuration

Table 6 Test 02-02, Basic Facility Config

Test Case Name	Basic facility configuration		Test Case id	Test-02-02
Test Purpose	Verify that the ELCM correctly reads the Facility Registry			
Configuration				
Test Tool	File system, Web browser			
Metric				

References			
Applicability	ELCM		
Pre-test conditions	Test-02-01 has been completed successfully.		
Test sequence	Step	From the <i>Additional Resources</i> section below, download the provided files	
	Step	Save the <i>simpleResource.yml</i> file on the <i>Resources</i> subfolder of the ELCM	
	Step	Save the <i>simpleScenario.yml</i> file on the <i>Scenarios</i> subfolder of the ELCM	
	Step	Save the <i>simpleTestCase.yml</i> file on the <i>TestCases</i> subfolder of the ELCM	
	Step	Save the <i>simpleUE.yml</i> file on the <i>UEs</i> subfolder of the ELCM	
	Step	Either restart the ELCM or click on the <i>Reload Facility</i> button of the Dashboard	
	Step	Review the contents of the Facility Log, ensure that no unexpected messages appear	
	Step	Using a web browser or a tool similar to <i>curl</i> , access to the <i>facility/scenarios</i> endpoint	The (raw) response should be similar to <code>{"Scenarios": [{"SimpleScenario"}]}</code>
	Step	Access to the <i>facility/testcases</i> endpoint	The (raw) response should be similar to <code>{"TestCases": [{"Distributed": false, "Name": "Simple Test Case", "Parameters": [], "PrivateCustom": [], "PublicCustom": false, "Standard": true}]}</code>
Step	Access to the <i>facility/ues</i> endpoint	The (raw) response should be similar to <code>{"UEs": [{"SimpleUE"}]}</code>	
Test Verdict	If no unexpected issues arise, then the ELCM loaded the facility information correctly and the facility endpoints are available		
Additional Resources	<a href="#">simpleResource.yml</a> - <a href="#">simpleScenario.yml</a> - <a href="#">simpleTestCase.yml</a> - <a href="#">simpleUE.yml</a>		

### 5.3. Test-02-03: Experiment execution

Table 7 Test 02-03, Experiment Execution

Test Case Name	Experiment execution	Test Case id	Test-02-03
Test Purpose	Verify that the ELCM is able to run experiments		
Configuration			
Test Tool	<i>curl</i> -like tool, Web browser		
Metric			
References			
Applicability	ELCM		
Pre-test conditions	Test-02-03 has been completed successfully		

Test sequence	Step	From the <i>Additional Resources</i> section below, download <i>descriptor.json</i>	
	Step	Using <i>curl</i> , or a similar tool, send the contents of the file as a json payload to the <i>api/v0/run</i> endpoint of the ELCM. For <i>curl</i> the command is: <i>curl -X POST -d @descriptor.json -H "Content-Type: application/json" &lt;ELCM_Address&gt;/api/v0/run</i>	The reply is: {"ExecutionId": <Execution_Number>}
	Step	In the ELCM dashboard, check if the experiment is running. If it has already finished, check the <i>History</i> tab and look for the <Execution_Number> entry.	
	Step	Review the experiment's execution logs to ensure that no unexpected errors appear.	
	Step	Using <i>curl</i> or the web browser, access to the <i>execution/&lt;Execution_Number&gt;/logs</i> endpoint	The same log contents, but in JSON format should appear.
Test Verdict	If no unexpected issues arise, then the ELCM is able to run experiments		
Additional Resources	<a href="#">descriptor.json</a>		

## 5.4. Test-03-01: Dispatcher installation

Table 8 Test 03-01, Dispatcher Installation

Test Case Name	Dispatcher installation		Test Case id	Test-03-01
Test Purpose	Verify the Dispatcher is correctly installed and configured with the MANO Components of the platform - Validate all Dispatcher features			
Configuration				
Test Tool	Robot Framework			
Metric				
References				
Applicability	Dispatcher, NFVO, VIM			
Pre-test conditions	Dispatcher is correctly installed following the <a href="#">instructions</a> - NFVO and VIM are installed with known URLs and credentials			
Test sequence	Step	Register new user	Passed	
	Step	Validate user	Passed	
	Step	Show registered users	Passed	
	Step	Get user token	Passed	
	Step	List platform VIMs	Passed	
	Step	Upload dummy image to VIM	Passed	
	Step	Register VIM Image (Basic Auth)	Passed	
	Step	Get image list from the catalogue	Passed	
	Step	Index Faulty VNFD (Token Auth)	Passed	
	Step	Index VNFD (Token Auth)	Passed	
	Step	Get VNFD list (Token Auth)	Passed	
	Step	Get VNFD list (Basic Auth)	Passed	
	Step	Index Faulty NSD (Token Auth)	Passed	

	Step	Index NSD (Token Auth)	Passed
	Step	Get NSD list (Token Auth)	Passed
	Step	Validate Bad Experiment Descriptor	Passed
	Step	Validate Experiment Descriptor	Passed
	Step	Onboard NSD (Token Auth)	Passed
	Step	Delete NSD	Passed
	Step	Delete user	Passed
	Step	Drop user database	Passed
Test Verdict	This script starts a web server to show the reports after the execution of all the tests. The report is accessible via web browser on port 8200.		
Additional Resources			

## 5.5. Test-04-01: Portal connectivity with other components

Table 9 Test 04-01, Portal Connectivity with Other Components

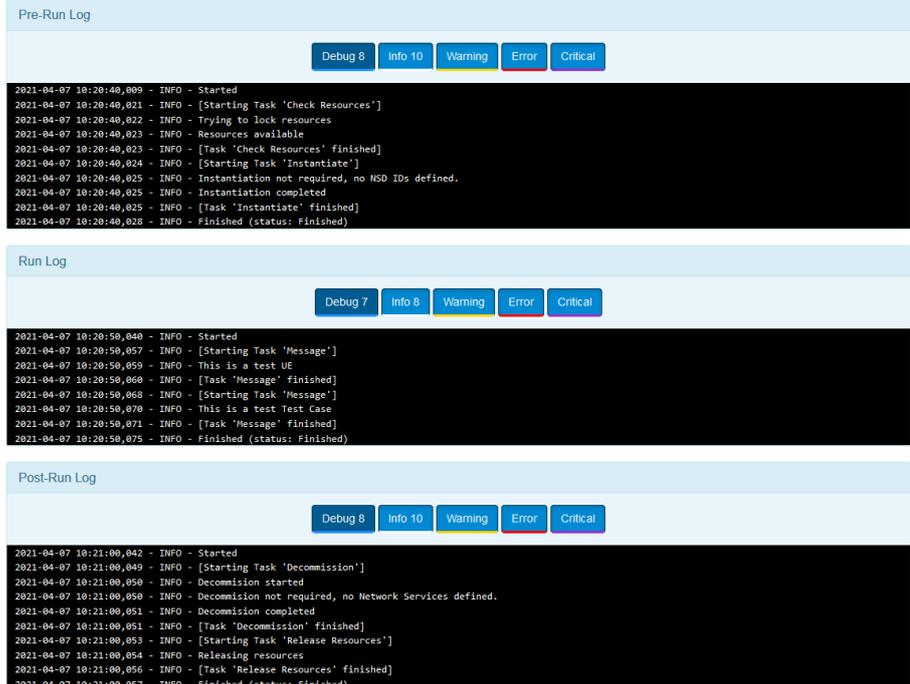
Test Case Name	Portal connectivity with other components		Test Case id	Test-04-01
Test Purpose	Verify that the Portal is reachable and has connectivity with the Dispatcher and ELCM			
Configuration				
Test Tool	Web browser			
Metric				
References				
Applicability	Portal, Dispatcher, ELCM			
Pre-test conditions	ELCM and Dispatcher have been deployed, configured and tested			
Test sequence	Step	Start the Portal and review the log messages, ensure that no unexpected errors appear.		
	Step	Near the start of the log, look for two lines that describe the number of available entities.	These lines should state (at least) <i>1 UEs, 1 Scenarios, 0 Slice Descriptors</i> and <i>TestCases: 1 standard, 0 public custom, 0 distributed</i>	
	Step	Open the Portal using a Web browser and click on the <i>Register</i> tab.	The registration form should be visible	
	Step	Fill in the form and click the <i>Register</i> button.	You should be redirected to the <i>Login</i> tab. A blue notice should be visible near the top, stating <i>User registered. Keep an eye with your email for knowing when your account is activated</i>	
	Step	Complete the user activation process on the Dispatcher by following the Dispatcher's documentation.		

	Step	Use your access credentials on the <i>Login</i> tab.	Your (empty) experimenter dashboard should be visible.
Test Verdict	If no unexpected issues arise, then the Portal is available and has connectivity with the required components		
Additional Resources			

## 5.6. Test-04-02: Experiment execution through the Portal

Table 10 Test 04-02, Experiment Execution through the Portal

Test Case Name	Portal experiment execution		Test Case id	Test-04-02
Test Purpose	Verify that the execution of experiments works correctly through the Portal			
Configuration				
Test Tool	Web browser			
Metric				
References				
Applicability	Portal, Dispatcher, ELCM			
Pre-test conditions	Test-04-01 has been completed successfully			
Test sequence	Step	If not already done, login to the Portal with the user created in Test-04-01		
	Step	Click on the <i>Create Experiment</i> tab on the top.	The experiment creation screen should appear	
	Step	Set a name (any name) for the experiment, and ensure that the experiment Type is set to <i>Standard</i>	<i>Simple Test Case</i> and <i>SimpleUE</i> should be selectable in the <i>Test Cases</i> and <i>UEs</i> fields	
	Step	Select both <i>Simple Test Case</i> and <i>SimpleUE</i> , do not enable the <i>Network slicing</i> section. Click on <i>Add Experiment</i>	You will be returned to the general dashboard, the new experiment should be visible	
	Step	Click the <i>Run</i> button on the row of the newly created experiment, wait for a few seconds	The screen will automatically reload	
	Step	Click the <i>History</i> button on the row of the newly created experiment	You will be taken to the execution history dashboard of the experiment, at least one Execution should appear	
	Step	Wait until the execution status changes to Finished (in case this does not happen after a few seconds, reload the page). Click on the  button (the leftmost one) on the row.	The execution logs should appear.	
	Step	Compare the execution logs with the snapshot available in <i>Additional Resources</i> below.		

Test Verdict	If the messages in the execution logs are similar, and no unexpected errors appear, then the execution of experiments through the Portal is working correctly
Additional Resources	 <p>The screenshot displays three log sections: Pre-Run Log, Run Log, and Post-Run Log. Each section has filter buttons for Debug, Info, Warning, Error, and Critical. The logs contain timestamps and status messages such as 'Started', '[Starting Task ...]', 'Finished (status: Finished)', and 'Decommission completed'.</p>

## 5.7. Test-04-03: Network service onboarding

Table 11 Test 04-03, Network Service Onboarding

Test Case Name	Test-04-03: Network service onboarding		Test Case id	Test-04-03
Test Purpose	Verify that the onboarding of network service's artifacts if correctly configured			
Configuration				
Test Tool	Web browser			
Metric				
References				
Applicability	Portal, Dispatcher			
Pre-test conditions	Test-04-01 has been completed successfully, the MANO configuration on the Dispatcher has been completed with at least one defined VIM location			
Test sequence	Step	Download the files in <i>Additional Resources</i> (below) to a known location		
	Step	If not already done, login to the Portal with the user created in Test-04-01		
	Step	Click on the <i>Network Services</i> tab on the top. On the network services dashboard click on the <i>New Network Service</i> button	The first step of the network service creation should appear	
	Step	Set a name (any name) for the network service and a location where the network service will be instantiated (Visibility and Description are not important for this test). Click on the <i>Create</i> button.	The network service configuration screen should appear.	

	Step	On the <i>Virtualized Infrastructure Manager</i> section, click on the <i>Browse</i> button, select the <i>VIM_image</i> file previously downloaded. Click on the <i>Pre-Load</i> button.	The file will be uploaded to an intermediate repository (the duration of this process depends on the size of the image and performance of the network connection, do not close the browser tab until finished). Once finished, the screen will refresh and the <i>Virtualized Infrastructure Manager</i> section contents will change.
	Step	On the <i>Virtualized Infrastructure Manager</i> section, click on the <i>Onboard</i> button.	The VIM image onboarding will start in the background and a notice will appear on the <i>Virtualized Infrastructure Manager</i> section.  <p>During this process the browser tab can be closed and the test resumed by clicking on the corresponding <i>Edit</i> button in the network services dashboard.</p>
	Step	Wait until the onboarding has finished, reloading the tab in order to update the status. Once finished, click on the <i>Commit</i> button	The VIM image should be visible in the <i>Available images</i> dropdown.
	Step	Select the newly uploaded <i>VIM image</i> on the dropdown and click on the <i>Select</i> button.	The screen will reload and the <i>VNFD Packages</i> and <i>Network Service Descriptor</i> sections will be enabled.
	Step	On the <i>VNFD Packages</i> section repeat the process of Pre-Loading and Onboarding, using the <i>VNFD_package</i> file from <i>Additional Resources</i> .	The newly onboarded <i>VNFD_package</i> should be visible in the <i>Available VNFDs</i> dropdown.
	Step	Select the previously uploaded file in the <i>Available VNFDs</i> dropdown and click on the <i>Add</i> button	The screen will reload and a new row corresponding to the VNFD file will be visible above the <i>Available VNFDs</i> dropdown.
	Step	On the <i>Network Service Descriptor</i> section, repeat the Pre-load and Onboard process, using the <i>NSD_file</i> from <i>Additional Resources</i>	

Test Verdict	If the <i>Basic Information</i> section displays a notice like the one below, then the onboarding of network services is working correctly.  <b>Network service ready</b>
Additional Resources	<a href="#">VIM image</a> - <a href="#">VNFD package</a> - <a href="#">NSD file</a>

## 5.8. Test-04-04: Network service deployment

Table 12 Test 04-04, Network Service Deployment

Test Case Name	Network service deployment		Test Case id	Test-04-04
Test Purpose	Verify that network services are correctly deployed during an experiment execution			
Configuration				
Test Tool	Web browser			
Metric				
References				
Applicability	Portal, Dispatcher, ELCM, Slice Manager			
Pre-test conditions	Test-04-03 has been completed successfully, the Slice Manager is configured on the ELCM. At least one Base Slice Descriptor is configured in the Slice Manager.			
Test sequence	Step	Repeat steps 1 to 4 of Test-04-02, but this time enable the <i>Network Slicing</i> section.		
	Step	On the <i>Network Slice</i> dropdown, select one of the available descriptors, on the <i>Scenario</i> dropdown, select <i>SimpleScenario</i> . Ensure that the selected Descriptor is compatible with the configuration on <i>SimpleScenario</i> ( <i>ue_DL_throughput: guaranteed: 1500000</i> )		
	Step	On the <i>Network Services</i> count, select <i>1</i>	A new table with a single row should appear.	
	Step	On the dropdown in the table, select the network service created in Test-04-03. Click on the <i>Add Experiment</i> button		
	Step	Initiate the experiment execution as explained in Test-04-02, review the execution logs.		
Test Verdict	If no unexpected issues arise and no errors appear in the execution log, then the instantiation of network services during an experiment execution is working correctly.			
Additional Resources				

## 5.9. Test-05-01: ELCM-InfluxDB integration

Table 13 Test 05-01, ELCM-InfluxDB Integration

Test Case Name	ELCM-InfluxDB integration		Test Case id	Test-05-01
Test Purpose	Verify that the connection between the ELCM and InfluxDB is correctly configured			
Configuration				
Test Tool	Portal			
Metric				
References				
Applicability	ELCM, InfluxDB			
Pre-test conditions	The Portal-Dispatcher-ELCM integration is working correctly. The connection with InfluxDB has been added to the ELCM configuration.			
Test sequence	Step	Copy the <i>InfluxDBTest.yml</i> file (in <i>Additional Resources</i> ) to the <i>TestCases</i> folder of the ELCM.		
	Step	Copy the <i>Results.csv</i> file (in <i>Additional Resources</i> ) to a known folder. Edit the <i>InfluxDBTest.yml</i> file so that the CSV field points to this file.		
	Step	Restart the ELCM or reload the configuration/facility so that the InfluxDB settings and new test case are included. Review the logs.	No unexpected errors should appear	
	Step	Restart the Portal so that the new testcase appears on the testcases list.		
	Step	In the Portal, create a new standard experiment. Select only the <i>InfluxDBTest</i> testcase.		
	Step	Run the experiment. Review the logs.		
	Step	Connect to the machine hosting the InfluxDB instance using SSH or similar.		
	Step	On the command prompt, run ' <i>influx</i> '	Some InfluxDB messages appear, ending with " <i>Enter an InfluxQL query</i> "	
	Step	Run " <i>use &lt;db&gt;</i> ", where <i>&lt;db&gt;</i> is the name of the database that contains the ELCM results	<i>"Using database &lt;db&gt;"</i> appears	
	Step	Run " <i>show measurements</i> "	<i>"Test0501Results"</i> appears in one of the returned lines	
	Step	Run " <i>select * from Test0501Results</i> "	Some results appear	
Test Verdict	If the table has been created and contains some results, then the connection of the ELCM with InfluxDB is working correctly			
Additional Resources	<a href="#">InfluxDBTest.yml</a> – <a href="#">Results.csv</a>			

## 5.10. Test-05-02: Distributed experiment

Table 14 Test 05-02, Distributed Experiment

Test Case Name	Distributed experiment		Test Case id	Test-05-02
Test Purpose	Verify that two platforms can execute distributed experiments			
Configuration				
Test Tool	Portal			
Metric				
References				
Applicability	Portal, Dispatcher, ELCM, InfluxDB			
Pre-test conditions	The Portal-Dispatcher-ELCM integration and Test-05-01 have been completed successfully in both platforms. Platform registration has been completed between the two Dispatchers, connection details are configured in the Portal and ELCM of both platforms and the East/West interface of all components is enabled.			
Test sequence	Step	If not already done, select which platform will act as <i>Main</i> , and which one will be <i>Remote</i> from the point of view of the user performing the experiment.		
	Step	Copy the <i>MainSide.yml</i> (in <i>Additional Resources</i> ) to the <i>TestCases</i> folder of the <i>Main</i> ELCM.		
	Step	Copy the <i>RemoteSide.yml</i> (in <i>Additional Resources</i> ) to the <i>TestCases</i> folder of the <i>Remote</i> ELCM.		
	Step	Copy the <i>Results.csv</i> (in <i>Additional Resources</i> ) to a known folder in both platforms (you may reuse this file from Test-05-01).		
	Step	Edit the <i>MainSide.yml</i> and <i>RemoteSide.yml</i> so that the <i>CSV</i> field points to the <i>Results.csv</i> file in the corresponding platform.		
	Step	Restart or reload the facility/configuration in both ELCM instances so that the changes are applied. Review the logs.		
	Step	Restart the Portal in both platforms. Review the logs.		
	Step	In the Portal of the <i>Main</i> platform, create a new distributed experiment. Select only the <i>MainSide</i> test case.		
	Step	Ensure that the <i>Remote</i> platform is selected in the <i>Remote Platform</i> dropdown and click <i>Continue</i>	The Portal of the <i>Main</i> side will connect with the <i>Remote</i> Portal and a new selection of testcases should appear	
	Step	In the second screen, select only the <i>RemoteSide</i> test case and click <i>Add Experiment</i>		

	Step	Execute the experiment. The execution will take a couple of minutes to complete.	
	Step	Review the execution logs on both sides (from the <i>Main</i> Portal). No unexpected errors should appear.	
	Step	Follow the instructions for viewing InfluxDB results described in Test-05-01 on the <i>Main</i> platform. Look for two tables called <i>TestResultsMain</i> and <i>Remote_TestResultsRemote</i> . Check if they contain results.	Both tables should contain some results
Test Verdict	If the execution of the test was successful and results appear in both tables, then the integration between the two platforms works correctly.		
Additional Resources	<a href="#">MainSide.yml</a> - <a href="#">RemoteSide.yml</a> - <a href="#">Results.csv</a>		

## 5.11. Test-06-01: TAP-MONROE configuration

Table 15 Test 06-01, TAP-Monroe Configuration

Test Case Name	TAP-MONROE configuration		Test Case id	Test-06-01
Test Purpose	Verify that the MONROE node is correctly configured and can be automated via TAP			
Configuration				
Test Tool	TAP GUI			
Metric				
References				
Applicability	TAP, MONROE			
Pre-test conditions	The MONROE node is configured, has Internet access and is reachable from the TAP machine, the MONROE TAP Agent is listening for connections.			
Test sequence	Step	If not already done, open the TAP GUI and configure a new MONROE instrument using the connection values that correspond to your MONROE node.		
	Step	Copy the <i>MONROE_Test.TapPlan</i> (in Additional Resources) file to a known folder in the TAP machine. Open the file (either by double-click or manually through the TAP GUI)	A test plan with three steps (Start Experiment, Delay and Stop Experiment) appears.	
	Step	Click on Run test plan (or press F5)	The test plan should start running. A Ping experiment will be started on the MONROE node, and will run for approximately 10 seconds.	
	Step	Review the log messages, there should be no Error or Warning messages. Look for an Info message similar to "Published 10 results of type MONROE.EXP.PING" ( $\pm 1$ result)		

Test Verdict	If around 10 Ping results were published, the MONROE instance is correctly configured and accessible through TAP
Additional Resources	<a href="#">MONROE_Test.TapPlan</a>

## 5.12. Test-06-02: TAP-Remote Ping agent

Table 16 Test 06-02, TAP-Remote Ping Agent

Test Case Name	TAP-Remote Ping agent		Test Case id	Test-06-02
Test Purpose	Verify that the Remote ping agent is correctly configured and can be automated via TAP			
Configuration				
Test Tool	TAP GUI, SSH			
Metric				
References				
Applicability	TAP, Remote ping agent			
Pre-test conditions	The remote Ping agent is deployed and listening at an address reachable from the TAP PC. The agent has Internet access (and Google's DNS servers are working)			
Test sequence	Step	If not already done, open the TAP GUI and configure a new Ping Agent instrument with the correct connection values.		
	Step	On a new test plan, add a Ping Agent step (5Genesis > Agents > Ping Agent). The default parameters are fine for this test.		
	Step	Click on Run test plan (or press F5)	The test plan should start running. A Ping experiment will be started on the remote machine, and will last for approximately 4 seconds.	
	Step	Review the log messages, there should be no Error or Warning messages.		
	Step	Follow the instructions in Test-05-01 for viewing the results contained in InfluxDB, but look for a table called "Remote_Ping_Agent".		
	Step	Run " <i>select * from Remote_Ping_Agent</i> "	Some results appear	
Test Verdict	Test-05-01 completed successfully. If some results appear then the agent successfully performed the ping test and the results were correctly retrieved by TAP.			
Additional Resources				

## 5.13. Test-06-03: TAP-Remote iPerf agent

Table 17 Test 06-03, TAP-Remote iPerf Agent

Test Case Name	TAP-Remote iPerf agent		Test Case id	Test-06-03
Test Purpose	Verify that the remote iPerf agent is correctly configured and can be automated via TAP			
Configuration				
Test Tool	TAP GUI, command line, SSH			
Metric				
References				
Applicability	TAP, Remote iPerf agent			
Pre-test conditions	Test-05-01 completed successfully. The remote iPerf agent is deployed and listening at an address reachable from the TAP PC.			
Test sequence	Step	On a machine reachable from the agent's PC (this may be the machine where TAP is running, the same machine where the agent is listening, or other), download and manually start an iPerf server instance (iPerf[.exe] -s ). Note the port number where the server is listening.		
	Step	If not already done, open the TAP GUI and configure a new iPerf Agent instrument with the correct connection values.		
	Step	On a new test plan, add an iPerf Agent step (5Genesis > Agents > iPerf Agent). Modify the step parameters so that the connection settings correspond to those of the iPerf server manually created. The rest of the parameters are valid for this test.		
	Step	Click on Run test plan (or press F5)	The test plan should start running. An iPerf experiment will be started on the remote machine, connecting to the server manually created.	
	Step	Review the log messages, there should be no Error or Warning messages.		
	Step	Follow the instructions in Test-05-01 for viewing the results contained in InfluxDB, but look for a table called "Remote_iPerf_Agent_Client".		
	Step	Run <code>"select * from Remote_iPerf_Agent_Client"</code>	Some results appear	
Test Verdict	If some results appear then the agent successfully performed the iPerf test and the results were correctly retrieved by TAP.			

Additional Resources	
----------------------	--

## 5.14. Test-06-04: TAP-ADB Ping agent

Table 18 Test 06-04, TAP-ADB Ping Agent

Test Case Name	TAP-ADB Ping agent		Test Case id	Test-06-04
Test Purpose	Verify that the Android Ping agent is correctly configured and can be automated via TAP			
Configuration				
Test Tool	TAP GUI, SSH			
Metric				
References				
Applicability	TAP, ADB ping agent			
Pre-test conditions	Test-05-01 completed successfully. Adb is installed on the TAP PC, the phone (and only one phone) is connected through USB, the ping Agent has been installed on the phone. The phone has access to the Internet (and Google servers are working)			
Test sequence	Step	If not already done, open the TAP GUI and configure new ADB and ADB Ping Agent instruments.		
	Step	On a new test plan, add an Adb Ping Agent step (UMA > Agents > Adb Ping Agent). The default parameters are fine for this test.		
	Step	Click on Run test plan (or press F5)	The test plan should start running. The agent will send ping requests to <a href="http://www.google.com">www.google.com</a> .	
	Step	Review the log messages, there should be no Error or Warning messages.		
	Step	Follow the instructions in Test-05-01 for viewing the results contained in InfluxDB,		

		but look for a table called "ADB_Ping_Agent".	
	Step	Run "select * from ADB_Ping_Agent"	Some results appear
Test Verdict	If some results appear then the agent successfully performed the ping test and the results were correctly retrieved by TAP.		
Additional Resources			

## 5.15. Test-06-05: TAP-ADB iPerf agent

Table 19 Test 06-05, TAP-ADB iPerf Agent

Test Case Name	TAP-ADB iPerf agent		Test Case id	Test-06-05
Test Purpose	Verify that the Android iPerf agent and device are correctly configured and accessible through TAP			
Configuration				
Test Tool	TAP GUI, SSH			
Metric				
References				
Applicability	TAP, ADB iPerf agent			
Pre-test conditions	Test-05-01 completed successfully. Adb is installed on the TAP PC, the phone (and only one phone) is connected through USB, the iPerf Agent has been installed on the phone.			
Test sequence	Step	If not already done, open the TAP GUI and configure new ADB and ADB iPerf Agent instruments.		
	Step	Copy the <i>ADB_iPerf_Test.TapPlan</i> (in <i>Additional Resources</i> ) file to a known folder in the TAP machine. Open the file (either by double-click or manually through the TAP GUI)	A test plan with two steps (as parent-child) appears.	
	Step	Click on Run test plan (or press F5)	The test plan should start running. The two available iPerf instances on the agent will connect to each other. The test will last for around 10 seconds.	
	Step	Review the log messages, in this case there will be Warning messages related to the parsing of the iPerf output, but no Errors.		
	Step	Follow the instructions in Test-05-01 for viewing the results contained in InfluxDB, but look for two tables called "ADB_iPerf_Agent_Client" and "ADB_iPerf_Agent_Server".		
	Step	Run "select * from ADB_iPerf_Agent_Client"	Some results appear	

	Step	Run <code>"select * from ADB_iPerf_Agent_Server"</code>	Some results appear
Test Verdict	If results appear on both tables, then the agent successfully performed the iPerf test and the results were correctly retrieved by TAP.		
Additional Resources	<a href="#">ADB_iPerf_Test.TapPlan</a>		

## 5.16. Test-06-06: TAP-ADB Resource agent

Table 20 Test 06-06, TAP-ADB Resource Agent

Test Case Name	TAP-ADB Resource agent		Test Case id	Test-06-06
Test Purpose	Verify that the Android Resource agent is correctly configured and can be automated via TAP			
Configuration				
Test Tool	TAP GUI, SSH			
Metric				
References				
Applicability	TAP, ADB Resource agent			
Pre-test conditions	Test-05-01 completed successfully. Adb is installed on the TAP PC, the phone (and only one phone) is connected through USB, the Resource Agent has been installed on the phone.			
Test sequence	Step	If not already done, open the TAP GUI and configure new ADB and ADB Resource Agent instruments.		
	Step	On a new test plan, add an Adb Resource Agent step (UMA > Agents > Adb Resource Agent). The default parameters are fine for this test.		
	Step	Click on Run test plan (or press F5)	The test plan should start running. The agent will take measurements on the phone for around 10 seconds.	
	Step	Review the log messages, there should be no Error or Warning messages.		
	Step	Follow the instructions in Test-05-01 for viewing the results contained in InfluxDB, but look for a table called "ADB_Resource_Agent".		
	Step	Run <code>"select * from ADB_Resource_Agent"</code>	Some results appear	
Test Verdict	If some results appear then the agent successfully collected performance measurements from the phone and these were correctly retrieved by TAP.			
Additional Resources				

## 6. TESTING AND VALIDATION RESULTS

Based on test cases defined in previous section, validation activity has been conducted by all platforms. The results summary is depicted in the following tables. Further details and screenshots about the integration process and can be found in Annex.

### 6.1. Athens Platform

Integration activities were concluded successfully in Athens Platform with the execution of all required tests. Results are summarised in Table 21.

**Table 21 Athens Platform Integration Results**

Test case id	Test case name	Result
<b>(Component) Integration Tests</b>		
Test-02-01	ELCM Dashboard	PASS
Test-02-02	Basic facility configuration	PASS
Test-02-03	Experiment execution	PASS
Test-03-01	Dispatcher installation	PASS
Test-04-01	Portal connectivity with other components	PASS
Test-04-02	Experiment execution through the Portal	PASS
Test-04-03	Network service onboarding	PASS
Test-04-04	Network service deployment	PASS
<b>Inter-Component Tests</b>		
Test-05-01	ELCM-InfluxDB integration	PASS
Test-05-02	Distributed experiment	POSTPONED
<b>Probe Tests</b>		
Test-06-01	TAP-MONROE configuration	PASS
Test-06-02	TAP-Remote Ping agent	PASS
Test-06-03	TAP-Remote iPerf agent	PASS
Test-06-04	TAP-ADB Ping agent	PASS
Test-06-05	TAP-ADB iPerf agent	PASS
Test-06-06	TAP-ADB Resource agent	PASS

However, test case 05-02 which refers to execution of distributed experiments is declared optional for all platforms and is postponed. Execution will involve coordination between Athens and Malaga platforms. Detailed information about the integration process that took place in Athens, can be found in Annex 2: Athens Platform Integration Activities

## 6.2. Berlin Platform

The results of the integration tests in Berlin Platform are summarized in Table 22. Details for the executed tests are provided along with relevant snapshots of GUIs and result listings in a dedicated section in Annex 3: Berlin Platform Integration Activities

**Table 22 Berlin Platform Integration Results**

Test case id	Test case name	Result
<b>(Component) Integration Tests</b>		
Test-02-01	ELCM Dashboard	PASS
Test-02-02	Basic facility configuration	PASS
Test-02-03	Experiment execution	PASS
Test-03-01	Dispatcher installation	PASS
Test-04-01	Portal connectivity with other components	PASS
Test-04-02	Experiment execution through the Portal	PASS
Test-04-03	Network service onboarding	PASS
Test-04-04	Network service deployment	PASS
<b>Inter-Component Tests</b>		
Test-05-01	ELCM-InfluxDB integration	PASS
Test-05-02	Distributed experiment	IGNORED
<b>Probe Tests</b>		
Test-06-01	TAP-MONROE configuration	PASS (with slight adjustments, see additional notes)
Test-06-02	TAP-Remote Ping agent	PASS (with slight adjustments, see additional notes)
Test-06-03	TAP-Remote iPerf agent	PASS (with slight adjustments, see additional notes)
Test-06-04	TAP-ADB Ping agent	IGNORED
Test-06-05	TAP-ADB iPerf agent	IGNORED
Test-06-06	TAP-ADB Resource agent	IGNORED

All 5Genesis components that are relevant in the integration tests were deployed into a virtual machine-based infrastructure at Berlin. With only few exceptions, 5Genesis components have been deployed using Ansible (a configuration management solution). To be able to do so,

installation scripts of nearly all 5Genesis components, as also dependent components (like OpenTAP), have been integrated or transferred into Ansible scripts. Exceptions are: Dispatcher and Katana Slice Manager.

### 6.3. Limassol Platform

The results of integration process in Limassol Platform are summarised in Table 23.

Table 23 Limassol Platform Integration Results

Test case id	Test case name	Result
<b>(Component) Integration Tests</b>		
Test-02-01	ELCM Dashboard	PASS
Test-02-02	Basic facility configuration	PASS
Test-02-03	Experiment execution	PASS
Test-03-01	Dispatcher installation	PARTIAL PASS
Test-04-01	Portal connectivity with other components	PASS
Test-04-02	Experiment execution through the Portal	PASS
Test-04-03	Network service onboarding	PASS
Test-04-04	Network service deployment	PENDING
<b>Inter-Component Tests</b>		
Test-05-01	ELCM-InfluxDB integration	PASS
Test-05-02	Distributed experiment	IGNORED
<b>Probe Tests</b>		
Test-06-01	TAP-MONROE configuration	PASS
Test-06-02	TAP-Remote Ping agent	PASS
Test-06-03	TAP-Remote iPerf agent	PASS
Test-06-04	TAP-ADB Ping agent	IGNORED
Test-06-05	TAP-ADB iPerf agent	IGNORED
Test-06-06	TAP-ADB Resource agent	IGNORED

Functionality and communication between components on most cases is successfully tested, more troubleshooting needs to be performed on image upload step on the VIM. This step is affecting test-03-01 which is set to PARTIAL PASS and test-04-03 which is completed by pre-uploading the image before test execution. Tests 06-04, 06-05, 06-06 are ignored, since the main UE is Waveshare 5G Hat for Raspberry Pi and Linux agent is used.

Detailed information about test executions can be found in Annex 4: Limassol Platform Integration Activities

### 6.4. Malaga Platform

Table 24 Malaga Platform Integration Results

Test case id	Test case name	Result
<b>(Component) Integration Tests</b>		
Test-02-01	ELCM Dashboard	PASS

Test-02-02	Basic facility configuration	PASS
Test-02-03	Experiment execution	PASS
Test-03-01	Dispatcher installation	PARTIAL PASS
Test-04-01	Portal connectivity with other components	PASS
Test-04-02	Experiment execution through the Portal	PASS
Test-04-03	Network service onboarding	POSTPONED
Test-04-04	Network service deployment	POSTPONED
<b>Inter-Component Tests</b>		
Test-05-01	ELCM-InfluxDB integration	PASS
Test-05-02	Distributed experiment	POSTPONED
<b>Probe Tests</b>		
Test-06-01	TAP-MONROE configuration	PASS
Test-06-02	TAP-Remote Ping agent	PASS
Test-06-03	TAP-Remote iPerf agent	PASS
Test-06-04	TAP-ADB Ping agent	PASS
Test-06-05	TAP-ADB iPerf agent	PASS
Test-06-06	TAP-ADB Resource agent	PASS

Most tests were completed successfully with no issues to report; however, 4 tests remain pending at the time of writing:

- Test-03-01 was partially completed and was successful on all sub-tests that do not make use of the MANO components. Similarly, the execution of test-04-03 and Test-04-04 has been delayed and is pending to the final integration between the Coordination layer and the MANO infrastructure, which is scheduled to be performed during September 2021.
- Test-05-02 is expected to be performed in collaboration with the Athens platform, with work starting on the month of September.

Detailed information about test executions can be found in Annex 5: Malaga Platform Integration Activities

## 6.5. Surrey Platform

Table 25 Surrey Platform Integration Results

Test case id	Test case name	Result
<b>(Component) Integration Tests</b>		
Test-02-01	ELCM Dashboard	PASS
Test-02-02	Basic facility configuration	PASS
Test-02-03	Experiment execution	PASS
Test-03-01	Dispatcher installation	PASS
Test-04-01	Portal connectivity with other components	PASS
Test-04-02	Experiment execution through the Portal	PASS
Test-04-03	Network service onboarding	PASS
Test-04-04	Network service deployment	PASS

Inter-Component Tests		
Test-05-01	ELCM-InfluxDB integration	PASS
Test-05-02	Distributed experiment	IGNORED
Probe Tests		
Test-06-01	TAP-MONROE configuration	IGNORED
Test-06-02	TAP-Remote Ping agent	PASS
Test-06-03	TAP-Remote iPerf agent	PASS
Test-06-04	TAP-ADB Ping agent	IGNORED
Test-06-05	TAP-ADB iPerf agent	IGNORED
Test-06-06	TAP-ADB Resource agent	IGNORED

Detailed information about test executions can be found in Annex 6: Surrey Platform Integration Activities.

## 7. CONCLUSIONS

---

This document presented the integration process of the Release B Open5Genesis components, performed within the context of the Work Package 5.

The integration activities had followed a well-defined methodology, which determines the basic operations from the stage of component development until the integration of the Coordination Layer and Slice Manager in each Platform, the guidelines for the respective tests that are used for the validation of each step of the process, and the conventions for software versioning, as well as the production of the respective documentation. The integration of Release B was performed in a dedicated environment in the Athens Platform. Documentation and coordination of integration activities was done using a GitHub repository.

The 5GENESIS Coordination Layer and Slice Manger were also briefly described, focusing on its features, main components, and functionality. Its main purpose is to allow experimenters to successfully perform a variety of experiments in the 5GENESIS Platforms.

The validation of the integration activities was performed with the use of a set of integration tests, following the ETSI NFV paradigm, which allow for the validation of the operation of the individual components, their proper communication, as well as the whole experimentation lifecycle. Finally, the results of the integration activities per Platform at the time of the deliverable submission are also reported.

## 8. REFERENCES

---

- [1] 5Genesis Consortium, "D5.4 Documentation and supporting material for 5G stakeholders (Release B)," 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/08/5GENESIS-D5.4\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2021/08/5GENESIS-D5.4_v1.0.pdf). [Accessed 19 July 2021].
- [2] 5GENESIS Consortium, "D5.1 System-Level Tests and Verification," March 2020. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2020/03/5GENESIS\\_D5.1\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2020/03/5GENESIS_D5.1_v1.0.pdf). [Accessed 19 July 2021].
- [3] 5Genesis Consortium, "D3.16 Experiment Lifecycle Manager (Release B)," 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/04/5GENESIS\\_D3.16\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2021/04/5GENESIS_D3.16_v1.0.pdf). [Accessed 19 July 2021].
- [4] 5Genesis Consortium, "D3.8 Open APIs, service level functions and interfaces for verticals (Release B)," 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/04/5GENESIS\\_D3.8\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2021/04/5GENESIS_D3.8_v1.0.pdf). [Accessed 19 July 2021].
- [5] 5Genesis Consortium, "D3.4 Slice Management (Release B)," 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS\\_D3.4\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS_D3.4_v1.0.pdf). [Accessed 19 July 2021].
- [6] "D3.6 Monitoring and Analytics (Release B)," 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS\\_D3.6\\_v1.0\\_FINAL.pdf](https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS_D3.6_v1.0_FINAL.pdf). [Accessed 19 July 2021].
- [7] 5Genesis Consortium, "D2.4 Final report on facility design and experimentation planning," 2020. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2020/07/5GENESIS\\_D2.4\\_v1.0.pdf](https://5genesis.eu/wp-content/uploads/2020/07/5GENESIS_D2.4_v1.0.pdf). [Accessed 19 July 2021].
- [8] 5GENESIS Consortium, "D3.6 Monitoring and Analytics (Release B)," April 2021. [Online]. Available: [https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS\\_D3.6\\_v1.0\\_FINAL.pdf](https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS_D3.6_v1.0_FINAL.pdf). [Accessed 19 July 2021].
- [9] 3GPP, "Study on management and orchestration of network slicing for next generation network," 2018. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3091>.
- [10] "5GTANGO Project," [Online]. Available: <https://www.5gtango.eu/>. [Accessed 23 August 2021].
- [11] "SONATA-NFV Platform," [Online]. Available: <https://www.sonata-nfv.eu/>. [Accessed 23 August 2021].
- [12] "ETSI GS NFV-TST 001 Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services".

# ANNEX 1: ATHENS PLATFORM INTEGRATION ENVIRONMENT

Two network subnets, namely 10.161.1.0/24 and 10.30.0.0/24 were used to perform the integration and testing activities of the Open5Genesis Rel. B software. Linux-based instances that were deployed on the OpenStack environment described in Section 4, have interfaces connected to services network 10.161.1.0/24 as depicted in Figure 7. Windows VMs deployed using ESXi are connected to subnet 10.30.0.0/24. Network traffic between the subnets is done by an external router.

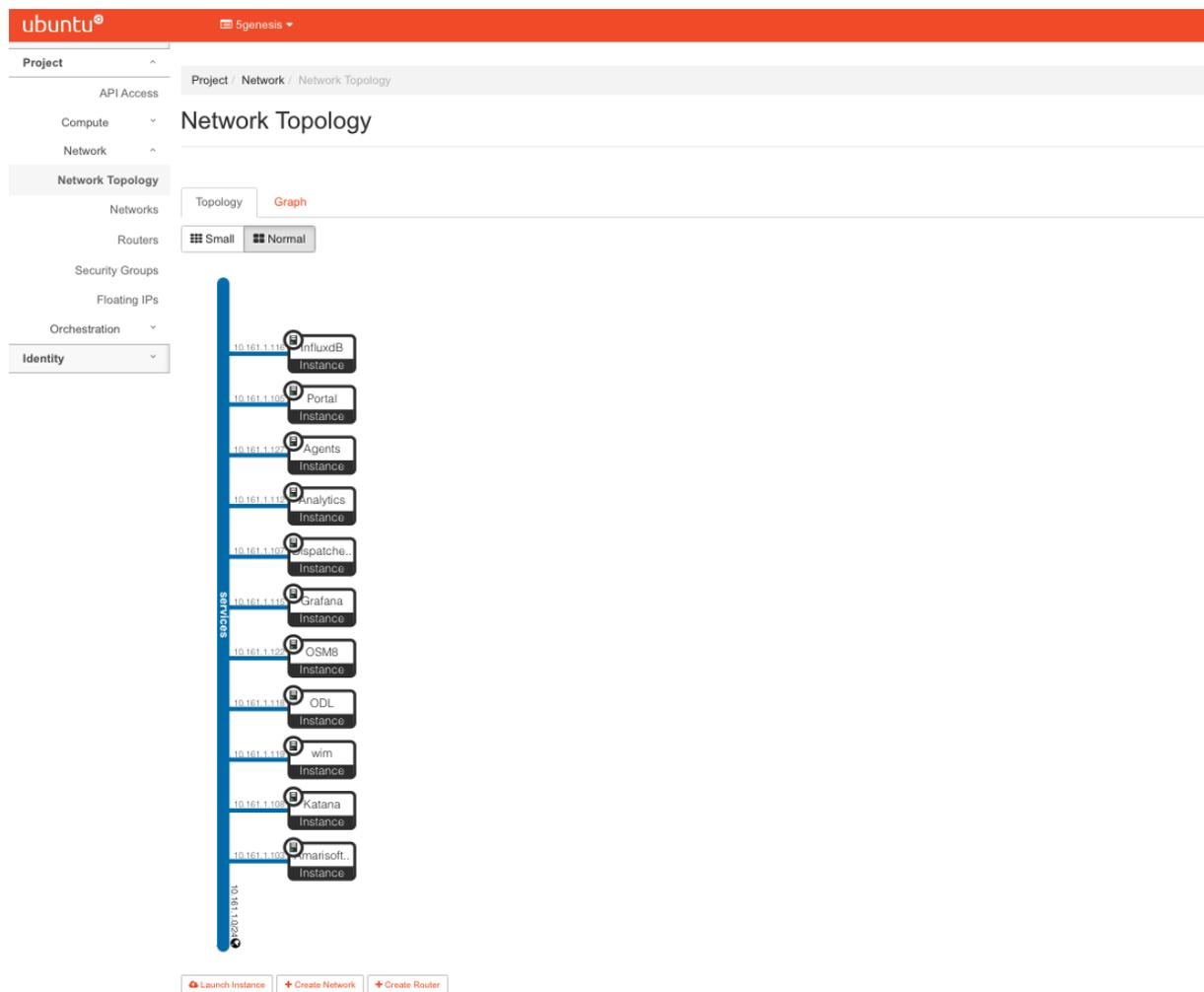


Figure 7 OpenStack Network

The list of resources used in the Athens Platform are listed below, while

Table 26 presents detailed information per component.

- OpenStack: 11 Instances, 22 vCPUs, 42 GB RAM, 470 GB Disk
- VMware ESXi: 1 VM, 2 vCPUs, 16 GB RAM, 64 GB Disk

Table 26 Integration Components

Component	Host	IP Address	Resources
OSM Rel 8	OpenStack	10.161.1.122	VCPUs: 2 RAM: 8GB Disk: 40GB
Portal	OpenStack	10.161.1.105	VCPUs: 2 RAM: 4GB Disk: 30GB
WIM	OpenStack	10.161.1.119	VCPUs: 2 RAM: 4GB Disk: 30GB
Amarisoft-EMS	OpenStack	10.161.1.103	VCPUs: 2 RAM: 2GB Disk: 40GB
Slice Manager	OpenStack	10.161.1.108	VCPUs: 4 RAM: 4GB Disk: 40GB
ODL	OpenStack	10.161.1.118	VCPUs: 2 RAM: 4GB Disk: 30GB
InfluxDB	OpenStack	10.161.1.116	VCPUs: 2 RAM: 4GB Disk: 120GB
ELCM & OpenTAP	ESXI	10.30.0.206	VCPUs: 2 RAM: 16GB Disk: 64GB
Agents	OpenStack	10.161.1.127	VCPUs: 1 RAM: 2GB Disk: 20GB
Analytics Framework	OpenStack	10.161.1.112	VCPUs: 2 RAM: 4GB Disk: 30GB
Dispatcher	OpenStack	10.161.1.107	VCPUs: 2 RAM: 4GB Disk: 30GB
Grafana	OpenStack	10.161.1.115	VCPUs: 1 RAM: 2GB Disk: 20GB

# ANNEX 2: ATHENS PLATFORM INTEGRATION ACTIVITIES

## Details for Test-02-01 to Test-02-03 (ELCM)

The screenshot shows the ELCM Dashboard interface. At the top, there's a browser address bar with '10.30.0.206:5001'. Below it, the 'Running Experiments' section shows a single experiment in a '(idle)' state. The 'Resources' section displays a green bar representing resource usage. The 'Diagnostics' section is expanded to show the 'Configuration Log' and 'Facility Log'. The Configuration Log shows details for logging, portal, slice maps, tap, and metadata. The Facility Log shows the loading of resources, test cases, and scenarios, along with dashboard definitions.

Figure 8 ELCM Dashboard (Athens)

```

thanosar@thanos-macbook ~ % curl http://10.30.0.206:5001/facility/testcases
{"TestCases":[{"Distributed":false,"Name":"InfluxDbTest","Parameters":{"PrivateCustom":{},"PublicCustom":{},"Standard":true},"Distributed":false,"Name":"Simple Test Case","Parameters":{"PrivateCustom":{},"PublicCustom":{},"Standard":true}}]}
thanosar@thanos-macbook ~ % curl http://10.30.0.206:5001/facility/ues
{"UES":["SimpleUE"]}
thanosar@thanos-macbook ~ % curl http://10.30.0.206:5001/facility/scenarios
{"Scenarios":["SimpleScenario"]}

```

Figure 9 ELCM Basic Facility Config (Athens)

```

thanosar@thanos-macbook Desktop % curl -X POST -d @descriptor.json -H "Content-Type:application/json" http://10.30.0.206:5001/api/v0/run
{"ExecutionId":9}
thanosar@thanos-macbook Desktop % curl http://10.30.0.206:5001/execution/9/logs
{"Executor":{"Count":{"Critical":0,"Debug":7,"Error":8,"Info":8,"Warning":0},"Log":{"Debug":["2021-08-29 17:15:53,857 - DEBUG - [File Opened]\n"],["Debug":["2021-08-29 17:15:53,858 - DEBUG - [Using temporal folder: Temp\tempvpx_vtqg]\n"],["Info":["2021-08-29 17:15:53,858 - INFO - Started\n"],["Info":["2021-08-29 17:15:53,861 - INFO - [Starting Task 'Message '\n"],["Debug":["2021-08-29 17:15:53,862 - DEBUG - Params: {'Message': 'This is a test UE', 'Severity': 'INFO'}\n"],["Info":["2021-08-29 17:15:53,862 - INFO - This is a test UE\n"],["Info":["2021-08-29 17:15:53,863 - INFO - [Task 'Message' finished]\n"],["Debug":["2021-08-29 17:15:53,863 - DEBUG - Params: {'Message': 'This is a test UE', 'Severity': 'INFO'}\n"],["Info":["2021-08-29 17:15:53,869 - INFO - [Starting Task 'Message'\n"],["Debug":["2021-08-29 17:15:53,881 - DEBUG - Params: {'Message': 'This is a test Test Case', 'Severity': 'INFO'}\n"],["Info":["2021-08-29 17:15:53,882 - INFO - This is a test Test Case\n"],["Info":["2021-08-29 17:15:53,883 - INFO - [Task 'Message' finished]\n"],["Debug":["2021-08-29 17:15:53,883 - DEBUG - Params: {'Message': 'This is a test Test Case', 'Severity': 'INFO'}\n"],["Info":["2021-08-29 17:15:53,886 - INFO - Finished (status: Finished)\n"],["Debug":["2021-08-29 17:16:03,889 - DEBUG - [Closing File]\n"]],"PostRun":{"Count":{"Critical":0,"Debug":8,"Error":10,"Warning":0},"Info":10,"Warning":0},"Log":{"Debug":["2021-08-29 17:16:03,878 - DEB\n - [File Opened]\n"],["Debug":["2021-08-29 17:16:03,878 - DEBUG - [Using temporal folder: Temp\tempvpx_vtqg]\n"],["Info":["2021-08-29 17:16:03,879 - INFO - Started\n"],["Info":["2021-08-29 17:16:03,895 - INFO - [Starting Task 'Decommission'\n"],["Debug":["2021-08-29 17:16:03,895 - INFO - Decommission not required, no Slice deployed.\n"],["Info":["2021-08-29 17:16:03,896 - INFO - Decommission completed\n"],["Info":["2021-08-29 17:16:03,897 - INFO - [Task 'Decommission' finished]\n"],["Debug":["2021-08-29 17:16:03,909 - DEBUG - Params: {'DeployedSliceId': None, 'NetworkServices': []}\n"],["Info":["2021-08-29 17:16:03,910 - INFO - [Starting Task 'Release Resources'\n"],["Debug":["2021-08-29 17:16:03,910 - DEBUG - Params: {'Id': 9, 'Available': False, 'Requirements': []}\n"],["Info":["2021-08-29 17:16:03,910 - INFO - Release resources\n"],["Debug":["2021-08-29 17:16:03,911 - DEBUG - Local Requirements: {'Id': 9, 'Available': False, 'Requirements': []}\n"],["Info":["2021-08-29 17:16:03,911 - INFO - [Task 'Release Resources' finished]\n"],["Debug":["2021-08-29 17:16:03,912 - DEBUG - Params: {'Id': 9, 'Available': False, 'Requirements': []}\n"],["Info":["2021-08-29 17:16:03,915 - INFO - Finished (status: Finished)\n"],["Debug":["2021-08-29 17:16:03,915 - DEBUG - [Closing File]\n"]],"PreRun":{"Count":{"Critical":0,"Debug":8,"Error":0,"Info":10,"Warning":0},"Log":{"Debug":["2021-08-29 17:15:43,847 - DEBUG - [File Opened]\n"],["Debug":["2021-08-29 17:15:43,848 - DEBUG - [Using temporal folder: Temp\tempvpx_vtqg]\n"],["Info":["2021-08-29 17:15:43,848 - INFO - Started\n"],["Info":["2021-08-29 17:15:43,865 - INFO - [Starting Task 'Check Resources'\n"],["Debug":["2021-08-29 17:15:43,865 - DEBUG - Params: {'Id': 9, 'Available': False, 'Requirements': [], 'NetworkServices': []}\n"],["Info":["2021-08-29 17:15:43,866 - INFO - Trying to lock resources\n"],["Debug":["2021-08-29 17:15:43,866 - DEBUG - Local Requirements: []}\n"],["Info":["2021-08-29 17:15:43,866 - INFO - Resources available\n"],["Info":["2021-08-29 17:15:43,867 - INFO - [Task 'Check Resources' finished]\n"],["Debug":["2021-08-29 17:15:43,880 - DEBUG - Params: {'Id': 9, 'Available': True, 'Requirements': True, 'NetworkServices': [], 'Feasible': True}\n"],["Info":["2021-08-29 17:15:43,880 - INFO - [Starting Task 'Instantiate'\n"],["Debug":["2021-08-29 17:15:43,880 - DEBUG - Params: {'NetworkServices': [], 'NEST': None, 'Slice': None}\n"],["Info":["2021-08-29 17:15:43,880 - INFO - Instantiation not required, base slice not defined.\n"],["Info":["2021-08-29 17:15:43,880 - INFO - Instantiation completed.\n"],["Info":["2021-08-29 17:15:43,881 - INFO - [Task 'Instantiate' finished]\n"],["Debug":["2021-08-29 17:15:43,881 - DEBUG - Params: {'NetworkServices': [], 'NEST': None, 'Slice': None}\n"],["Info":["2021-08-29 17:15:43,882 - INFO - Finished (status: Finished)\n"],["Debug":["2021-08-29 17:15:43,884 - DEBUG - [Closing File]\n"]],"Status":"Success"}

```

Figure 10 ELCM Experiment Execution (Athens)

## Details for Test-03-01 (Dispatcher installation)

Validation of Dispatcher installation is done by executing the automated unit tests that were created using Robot Framework. A detailed report is shown in Figure 11.

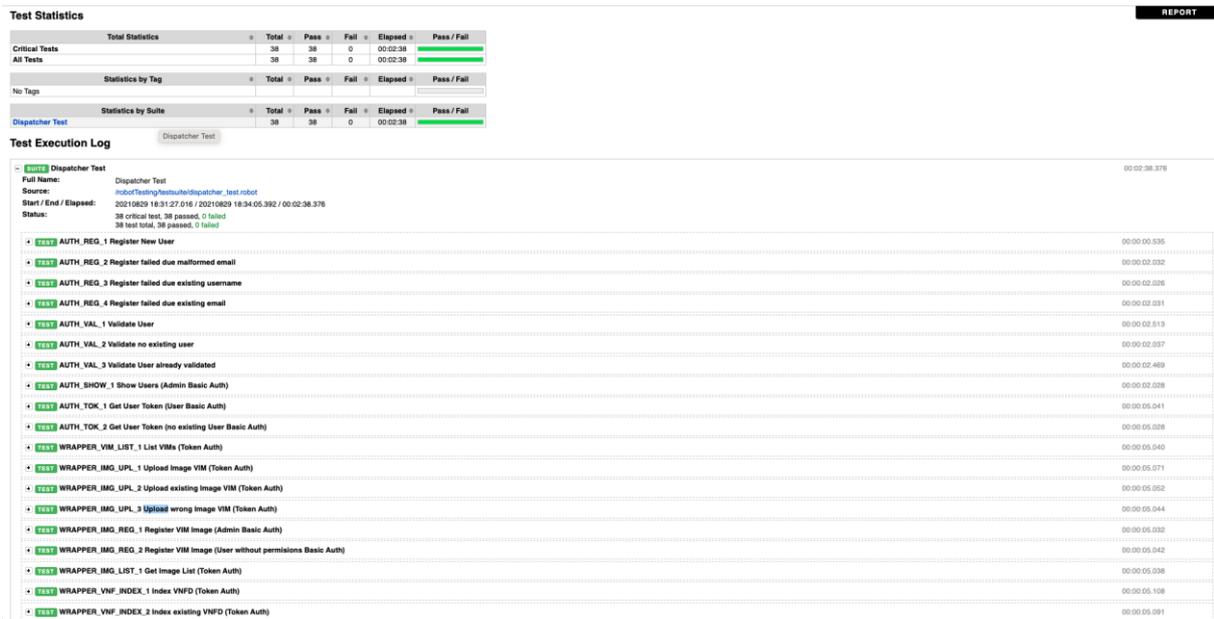


Figure 11 Dispatcher Unit Test Results (Athens)

### Details for Test 04-01 to Test 04-04

Portal is installed, running on address 10.161.1.105 and the user registration form is shown on Figure 12.

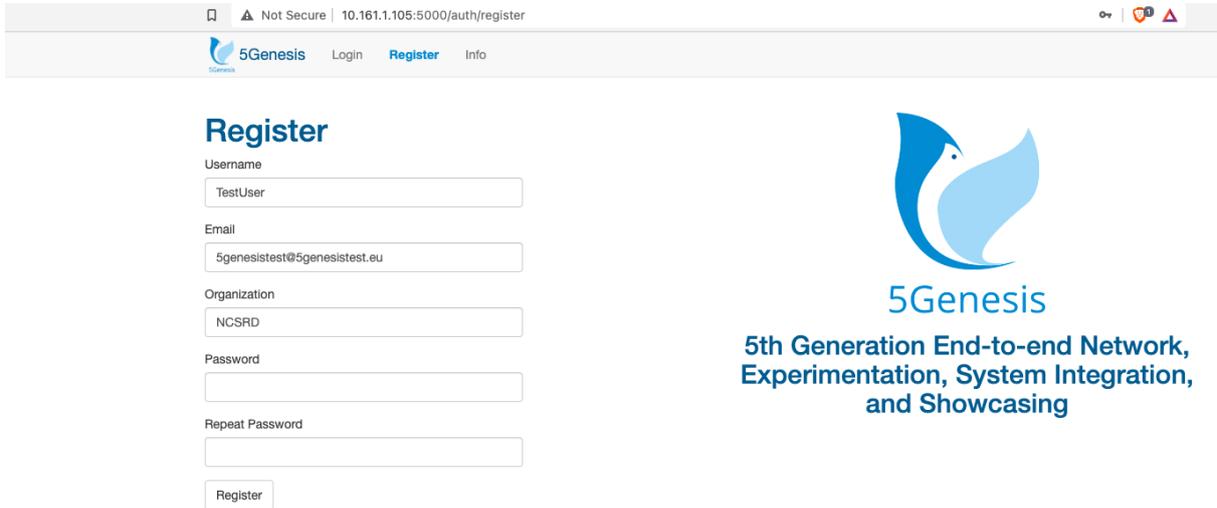


Figure 12 Portal User Registration Form (Athens)

Registration is concluded with the user activation step, performed by each Platform Administrator via email, as shown in Figure 13.

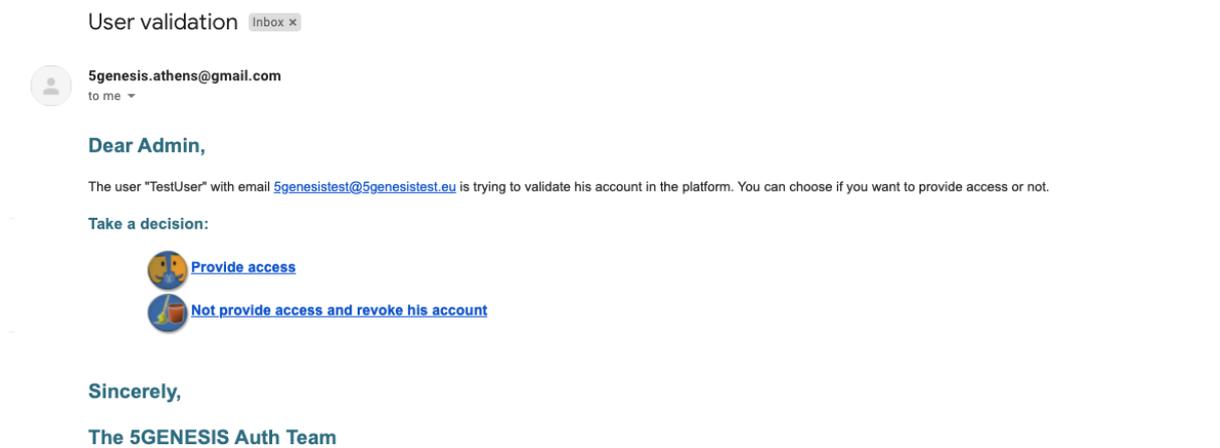


Figure 13 User Activation Email (Athens)

The simple experiment creation process described on Test 04-02 is presented in Figure 14. After the experiment execution the execution logs retrieved from ELCM can be found in Figure 15.

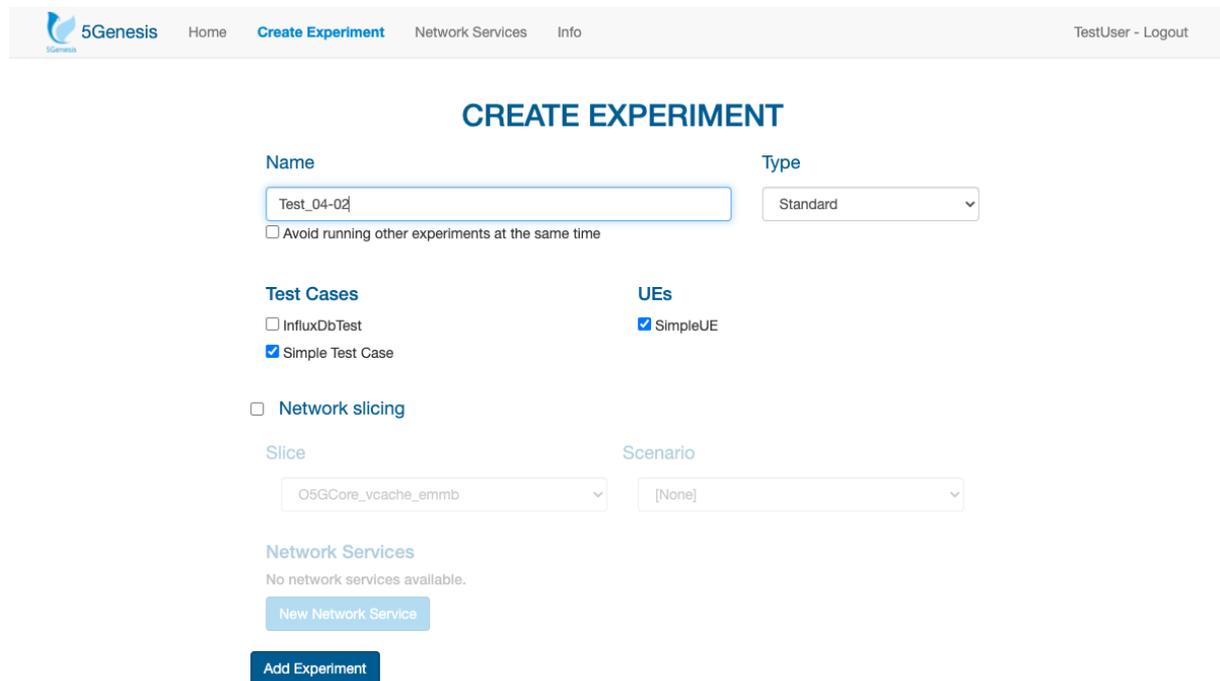


Figure 14 Simple Experiment Creation (Athens)

### Execution 10

Status	Start Time	End Time	Experiment	Action
Finished	30 August 2021, 1:45:46	30 August 2021, 1:46:16	Test_04-02	

#### Pre-Run Log

Debug 8 Info 10 Warning Error Critical

```

2021-08-30 10:45:45.912 - INFO - Started
2021-08-30 10:45:45.923 - INFO - [Starting Task 'Check Resources']
2021-08-30 10:45:45.924 - INFO - Trying to lock resources
2021-08-30 10:45:45.925 - INFO - Resources available
2021-08-30 10:45:45.926 - INFO - [Task 'Check Resources' finished]
2021-08-30 10:45:45.927 - INFO - [Starting Task 'Instantiate']
2021-08-30 10:45:45.927 - INFO - Instantiation not required, base slice not defined.
2021-08-30 10:45:45.927 - INFO - Instantiation completed
2021-08-30 10:45:45.928 - INFO - [Task 'Instantiate' finished]
2021-08-30 10:45:45.929 - INFO - Finished (status: Finished)
                
```

#### Run Log

Debug 7 Info 8 Warning Error Critical

```

2021-08-30 10:45:55.917 - INFO - Started
2021-08-30 10:45:55.919 - INFO - [Starting Task 'Message']
2021-08-30 10:45:55.921 - INFO - This is a test UE
2021-08-30 10:45:55.921 - INFO - [Task 'Message' finished]
2021-08-30 10:45:55.926 - INFO - [Starting Task 'Message']
2021-08-30 10:45:55.938 - INFO - This is a test Test Case
2021-08-30 10:45:55.938 - INFO - [Task 'Message' finished]
2021-08-30 10:45:55.939 - INFO - Finished (status: Finished)
                
```

#### Post-Run Log

Debug 8 Info 10 Warning Error Critical

```

2021-08-30 10:46:05.930 - INFO - Started
2021-08-30 10:46:05.937 - INFO - [Starting Task 'Decommission']
2021-08-30 10:46:05.938 - INFO - Decommission started
2021-08-30 10:46:05.938 - INFO - Decommission not required, no Slice deployed.
2021-08-30 10:46:05.939 - INFO - Decommission completed
2021-08-30 10:46:05.939 - INFO - [Task 'Decommission' finished]
2021-08-30 10:46:05.953 - INFO - [Starting Task 'Release Resources']
2021-08-30 10:46:05.953 - INFO - Releasing resources
2021-08-30 10:46:05.953 - INFO - [Task 'Release Resources' finished]
2021-08-30 10:46:05.955 - INFO - Finished (status: Finished)
                
```

Figure 15 Simple Experiment Execution Logs (Athens)

For test 04-03 a basic network service was created using the portal ‘Network Services’ tab. At first, a VIM was chosen from the VIM repository kept inside the Dispatcher mano module. Then, the network service image *test\_image2.qcow* was uploaded and onboarded as shown in Figure 16. To conclude the network service creation, a VNF descriptor, responsible for the deployment of the image and a NS descriptor were onboarded. (Figure 17)

### Virtualized Infrastructure Manager

Vim Image: test\_image2.qcow2

**Onboarding VIM image**

VIM Image successfully onboarded

Figure 16 NS Image Onboarding (Athens)

### VNFD Packages

hackfest1-vnf      ID: hackfest1-vnf

Available VNFDs:

Add VNFD package

---

### Network Service Descriptor

hackfest\_1\_nsd\_fixed.targzID: None

Onboarding NSD package
NSD file successfully onboarded

Figure 17 NS and VNF Onboarding (Athens)

This test group is completed with Test 04-04 which summarizes all previous test results to be used for an experiment execution with slicing features in order to validate interoperability with the Slice Manager. For this test, an additional Test Case was defined that executes a TapPlan with a simple Delay step during the “Run” phase, to simulate experiment execution and provide time for Slice Manager to instantiate the slice. A base slice descriptor was chosen comprised by Sample 5GCore VNF. Additionally, the network service defined on test 04-03 was included to the experiment descriptor. Experiment creation is presented in Figure 18. After experiment execution the activated slice with its unique ID is shown on Slice Manager cli (Figure 19). Deployed instances on the newly created OpenStack project are shown in Figure 20 and portal execution logs in Figure 21.

Home   [Create Experiment](#)   Network Services   Info
TestUser - Logout

## CREATE EXPERIMENT

**Name**       **Type**

Avoid running other experiments at the same time

**Test Cases**

Delay\_TapPlan

Drone\_Variant\_2

InfluxDbTest

Simple Test Case

System\_Level\_Test

**UEs**

SimpleUE

Network slicing

**Slice**       **Scenario**

**Network Services**

1	1	<input type="text" value="test_04-03"/>
---	---	---

Figure 18 Experiment Creation with Slice (Athens)

```
ubuntu@katana:~$ katana slice ls
SLICE_ID                                CREATED AT                                STATUS
73b20e09-14f9-4eaa-a895-a8c84bbb8e88    2021-08-30 15:40:51                      Activation
ubuntu@katana:~$ katana slice ls
SLICE_ID                                CREATED AT                                STATUS
73b20e09-14f9-4eaa-a895-a8c84bbb8e88    2021-08-30 15:40:51                      Running
```

Figure 19 Slice Manager CLI at Instantiation (Athens)

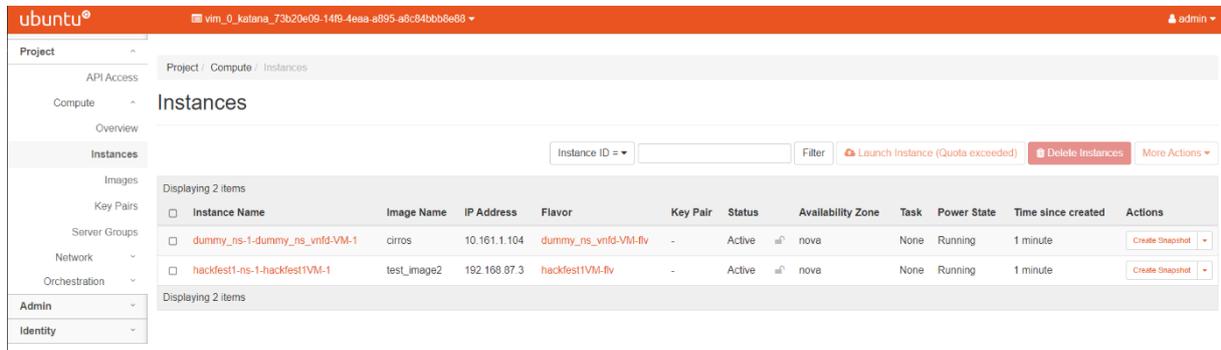


Figure 20 Slice Instances OpenStack (Athens)

5Genesis Home Create Experiment Network Services Info TestUser - Logout

### Execution 12

Status	Start Time	End Time	Experiment	Action
Finished	30 August 2021, 6:40:51	30 August 2021, 6:44:21	Test_04-04	[Refresh] [Download] [Share]

**Pre-Run Log**

Debug 10 Info 10 Warning Error Critical

```
2021-08-30 15:40:50,092 - INFO - Started
2021-08-30 15:40:50,103 - INFO - [Starting Task Check Resources]
2021-08-30 15:40:50,104 - INFO - Trying to lock resources
2021-08-30 15:40:50,142 - INFO - Resources available
2021-08-30 15:40:50,142 - INFO - [Task Check Resources' finished]
2021-08-30 15:40:50,143 - INFO - [Starting Task Instantiate]
2021-08-30 15:40:50,143 - INFO - Experiment contains 1 NSD IDs over Base Slice 'Scratch_OS0'. Requesting instantiation.
2021-08-30 15:40:50,300 - INFO - Instantiation completed
2021-08-30 15:40:50,300 - INFO - [Task Instantiate' finished]
2021-08-30 15:40:50,302 - INFO - Finished (status: Finished)
```

**Run Log**

Debug 44 Info 33 Warning Error Critical

```
2021-08-30 15:41:00,102 - INFO - Started
2021-08-30 15:41:00,103 - INFO - [Starting Task Message]
2021-08-30 15:41:00,104 - INFO - This is a test UI
2021-08-30 15:41:00,104 - INFO - [Task Message' finished]
2021-08-30 15:41:00,106 - INFO - [Starting Task Tap Execute]
2021-08-30 15:41:00,109 - INFO - Executing TestPlan C:\Users\test\Desktop\Main\TapPlan\Sein_dslpy_TapPlan
2021-08-30 15:41:00,934 - INFO - [TAP]OpenTAP Command Line Interface 9.14.0a51c7081e
2021-08-30 15:41:00,934 - INFO - [TAP]
2021-08-30 15:41:01,008 - INFO - [TAP]01:00:00:747: CLI - Information: Loaded test plan from C:\Users\test\Desktop\Main\TapPlan\Sein_dslpy_TapPlan [37.2 ms]
2021-08-30 15:41:01,028 - INFO - [TAP]01:00:00:747: Meta - Information: Test Plan 'Sein_dslpy
2021-08-30 15:41:01,028 - INFO - [TAP]01:00:00:832: TestPlan - Information: -----
2021-08-30 15:41:01,151 - INFO - [TAP]01:00:00:833: TestPlan - Information: Starting TestPlan 'Sein_dslpy' on 08/30/2021 15:41:01, 2 of 2 TestSteps enabled.
2021-08-30 15:41:01,261 - INFO - [TAP]01:00:00:996: integration - Information: Resource 'integrator' opened. [15.0 ms]
2021-08-30 15:41:01,261 - INFO - [TAP]01:00:00:996: TestPlan - Information: Delay 'started'.
2021-08-30 15:41:02,597 - INFO - [TAP]01:00:02:294: UpdateCheck - Information: Updates available for:
2021-08-30 15:41:02,597 - INFO - [TAP]01:00:02:295: UpdateCheck - Information: - Developer's System CE: 9.14.1a01996a6 -> 9.15.2a238682f7
2021-08-30 15:41:02,597 - INFO - [TAP]01:00:02:295: UpdateCheck - Information: - OpenTAP: 9.14.0a51c7081e -> 9.15.2a3962ca2
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:013: TestPlan - Information: Delay 'completed. [189 s]
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:012: TestPlan - Information: Log Output' started.
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:013: TestStep - Information: Slice instantiated
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:013: TestPlan - Information: Log Output' completed. [674 ms]
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:020: Summary - Information: ----- Summary of test plan started 08/30/2021 15:41:01 -----
2021-08-30 15:44:01,228 - INFO - [TAP]01:03:01:032: Summary - Information: Delay 189 s
2021-08-30 15:44:01,229 - INFO - [TAP]01:03:01:032: Summary - Information: Log Output 674 ms
2021-08-30 15:44:01,229 - INFO - [TAP]01:03:01:032: Summary - Information: -----
2021-08-30 15:44:01,229 - INFO - [TAP]01:03:01:033: Summary - Information: ----- Test plan completed successfully in 180 s -----
2021-08-30 15:44:01,539 - INFO - [TAP]01:03:01:043: integration - Information: Sending 15 log messages to integration
2021-08-30 15:44:03,347 - INFO - [TAP]01:03:01:043: TestPlan - Information: Waiting for OnTestPlanCompleted for integration.
2021-08-30 15:44:03,640 - INFO - [TAP]01:03:01:251: integration - Information: Waiting for result propagation for integration
2021-08-30 15:44:03,673 - INFO - [TAP]01:03:01:251: integration - Information: Resource 'integrator' closed. [259 us]
2021-08-30 15:44:03,673 - INFO - [TAP]01:03:01:251: integration - Information: Results collection finished, skipping.
2021-08-30 15:44:03,673 - INFO - [TAP]01:03:01:251: integration - Information: [Task Tap Execute' finished]
2021-08-30 15:44:03,674 - INFO - Finished (status: Finished)
```

**Post-Run Log**

Debug 8 Info 11 Warning Error Critical

```
2021-08-30 15:44:10,286 - INFO - Started
2021-08-30 15:44:10,295 - INFO - [Starting Task Decommission]
2021-08-30 15:44:10,296 - INFO - Decommission started
2021-08-30 15:44:10,296 - INFO - Experiment has 1 network services with slice ID: 73b20e09-14f9-4eaa-a895-a8c84bbb8e88. Requesting decommission
2021-08-30 15:44:10,459 - INFO - Slice decommissioned
2021-08-30 15:44:10,459 - INFO - Decommission completed
2021-08-30 15:44:10,459 - INFO - [Task Decommission' finished]
2021-08-30 15:44:10,461 - INFO - [Starting Task Release Resources]
2021-08-30 15:44:10,461 - INFO - Releasing resources
2021-08-30 15:44:10,461 - INFO - [Task Release Resources' finished]
2021-08-30 15:44:10,603 - INFO - Finished (status: Finished)
```

Figure 21 Portal Execution Log for Slice Instantiation (Athens)

## Details for Test 05-01

The results recorded to influxDB during this test are shown using Chronograf Dashboard in Figure 22.

time	InfluxDbTestResults.ExecutionId	InfluxDbTestResults.Jitter (ms)	InfluxDbTestResults.Name	InfluxDbTestResults.Packet Loss (%)	InfluxDbTestResults.PlanName	InfluxDbTestResults.ResultType	InfluxDbTestResults.StepDuration
11/26/2020 18:35:32	14.00	0.56	Adb iPerf Agent	94.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:33	14.00	0.12	Adb iPerf Agent	61.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:34	14.00	0.12	Adb iPerf Agent	38.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:35	14.00	0.09	Adb iPerf Agent	59.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:36	14.00	0.23	Adb iPerf Agent	41.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:37	14.00	0.18	Adb iPerf Agent	37.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:38	14.00	0.18	Adb iPerf Agent	25.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:39	14.00	0.05	Adb iPerf Agent	28.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:40	14.00	0.14	Adb iPerf Agent	36.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:41	14.00	0.06	Adb iPerf Agent	14.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:42	14.00	0.07	Adb iPerf Agent	19.00	Untitled	Adb iPerf Agent Server	
11/26/2020 18:35:43	14.00	0.07	Adb iPerf Agent	73.00	Untitled	Adb iPerf Agent Server	

Figure 22 InfluxDB Recorded Results (Athens)

## Details for Test 06-01 to Test 06-06

This test group is responsible for validating the installation and interoperability of OpenTAP engine with the monitoring probes via the TAP plugins. Results are presented in the following order:

- Monroe Probe, ping rtt test
- Linux Ping Agent, ping rtt test
- Linux iPerf Agent, iPerf throughput test
- Adb Ping Agent, ping rtt test (Mobile Phone)
- Adb iPerf Agent, iPerf throughput test (Mobile Phone)
- Adb Resource Agent, additional performance monitoring (Mobile Phone)

time	MONROE_EXP_PING.Bytes	MONROE_EXP_PING.DataId	MONROE_EXP_PING.DataVersion	MONROE_EXP_PING.Guid
03/13/2020 20:17:38	84.00	MONROE_EXP_PING	2.00	sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experimen
03/13/2020 20:17:39	84.00	MONROE_EXP_PING	2.00	sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experimen
03/13/2020 20:17:40	84.00	MONROE_EXP_PING	2.00	sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experimen
03/13/2020 20:17:41	84.00	MONROE_EXP_PING	2.00	sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experimen
03/13/2020 20:17:42	84.00	MONROE_EXP_PING	2.00	sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experimen

Figure 23 Monroe Experiment Results in InfluxDB (Athens)

During test-06-02 execution, ICMP requests were sent from ping agent installation to address 8.8.8.8 as shown in Figure 24. Results were recorded in influxDB under ExecutionId, 'test-06-02' (Figure 25).

```

ubuntu@agents: ~/Remote_Ping_Agent
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://10.161.1.127:5001/ (Press CTRL+C to quit)

Final CLI parameters: ['ping', '-i', '1.0', '-0', '8.8.8.8']
10.30.0.206 - - [03/Sep/2021 17:06:55] "GET /Ping/8.8.8.8?size=0&ttl=0&interval=1 HTTP/1.1" 200 -
ping running
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=23.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=22.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=22.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=22.9 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=22.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=22.9 ms
10.30.0.206 - - [03/Sep/2021 17:07:01] "GET /Close HTTP/1.1" 200 -
Final JSON results
{'icmp_replies': [{'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 1,
                  'time': 23.0,
                  'timestamp': 1630688816.797721,
                  'ttl': 116},
                 {'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 2,
                  'time': 22.9,
                  'timestamp': 1630688817.797721,
                  'ttl': 116},
                 {'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 3,
                  'time': 22.9,
                  'timestamp': 1630688818.797721,
                  'ttl': 116},
                 {'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 4,
                  'time': 22.9,
                  'timestamp': 1630688819.797721,
                  'ttl': 116},
                 {'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 5,
                  'time': 22.8,
                  'timestamp': 1630688820.797721,
                  'ttl': 116},
                 {'bytes': 64,
                  'destination': '8.8.8.8',
                  'duplicate': False,
                  'icmp_seq': 6,
                  'time': 22.9,
                  'timestamp': 1630688821.797721,
                  'ttl': 116}]}

```

Figure 24 Linux Ping Agent Console (Athens)

OpenTAP logs for this execution are shown below:

```

17:06:55.281 TestPlan -----
-----
17:06:55.313 TestPlan Starting TestPlan 'Remote Ping' on 09/03/2021
17:06:55, 2 of 2 TestSteps enabled.
17:06:55.347 TestPlan Saved Test Plan XML [2.98 ms]
17:06:55.439 Resolver Found 32/94 assembly files. [18.5 ms]
17:06:55.440 Resolver Found match for InfluxDB.LineProtocol,
Version=1.1.0.0, Culture=neutral, PublicKeyToken=null in C:\Program
Files\OpenTAP\Dependencies\InfluxDB.LineProtocol.1.1.0.0\InfluxDB.LineProto
col.dll
17:06:55.443 integration Resource "integration" opened. [24.3 ms]
17:06:55.532 Resolver Found match for RestSharp, Version=106.6.9.0,
Culture=neutral, PublicKeyToken=598062e77f915f75 in C:\Program
Files\OpenTAP\Dependencies\RestSharp.106.6.9.0\RestSharp.dll
17:06:55.548 PingA NCSR Cloud Resource "PingA NCSR Cloud" opened. [127
ms]
17:06:55.564 TestPlan "Ping Agent" PrePlanRun completed. [2.02 ms]
17:06:55.564 TestPlan PrePlanRun Methods completed [4.40 ms]
17:06:55.577 TestPlan "Set Execution ID" started.
17:06:55.580 TestStep Setting ExecutionId to test-06-02 (integration)
17:06:55.587 TestPlan "Set Execution ID" completed. [7.90 ms]
17:06:55.588 TestPlan "Ping Agent" started.
17:06:55.592 PingA NCSR Cloud Sending request: GET - Ping/8.8.8.8
17:06:55.594 PingA NCSR Cloud Parameters: size:0; ttl:0; interval:1;
17:06:55.773 PingA NCSR Cloud {"Message":"Successfully executed
ping","Status":"Success"}

17:07:01.274 PingA NCSR Cloud Sending request: GET - Close
17:07:01.284 PingA NCSR Cloud {"Message":"Successfully closed
ping","Status":"Success"}

17:07:02.286 PingA NCSR Cloud Sending request: GET - LastJsonResult

```

```
17:07:02.298 PingA NCSR Cloud {"Message":"Successfully retrieved last
json
result","Result":{"icmp_replies":[{"bytes":64,"destination":"8.8.8.8","dupl
icate":false,"icmp_seq":1,"time":23.0,"timestamp":1630688816.797721,"ttl":1
16},{"bytes":64,"destination":"8.8.8.8","duplicate":false,"icmp_seq":2,"tim
e":22.9,"timestamp":1630688817.797721,"ttl":116},{"bytes":64,"destination":
"8.8.8.8","duplicate":false,"icmp_seq":3,"time":22.9,"timestamp":1630688818
.797721,"ttl":116},{"bytes":64,"destination":"8.8.8.8","duplicate":false,"i
cmp_seq":4,"time":22.9,"timestamp":1630688819.797721,"ttl":116},{"bytes":64
,"destination":"8.8.8.8","duplicate":false,"icmp_seq":5,"time":22.8,"timest
amp":1630688820.797721,"ttl":116},{"bytes":64,"destination":"8.8.8.8","dupl
icate":false,"icmp_seq":6,"time":22.9,"timestamp":1630688821.797721,"ttl":1
16}], "success":6, "total":6}, "Status":"Success"}

17:07:02.311 TestPlan "Ping Agent" completed. [6.72 s]
17:07:02.313 TestPlan Test step runs finished. [6.74 s]
17:07:02.318 TestPlan "Ping Agent" PostPlanRun completed. [151 us]
17:07:02.328 integration Sending 6 results ('Remote Ping Agent' as
'Remote_Ping_Agent') to integration
17:07:02.332 Summary ----- Summary of test plan started 09/03/2021
17:06:55 -----
17:07:02.335 Summary Set Execution ID
7.90 ms
17:07:02.335 Summary Ping Agent
6.72 s
17:07:02.335 Summary -----
-----
17:07:02.336 Summary ----- Test plan completed successfully in
6.96 s -----
17:07:02.356 integration Sending 1 results ('Remote Ping Agent
Aggregated' as 'Remote_Ping_Agent_Aggregated') to integration
17:07:02.364 integration Sending 14 log messages to integration
17:07:02.368 integration OnTestPlanRunCompleted for integration. [6.36
ms]
17:07:02.377 integration Resource "integration" closed. [264 us]
17:07:02.377 PingA NCSR Cloud Resource "PingA NCSR Cloud" closed. [221
us]
```

time	Remote_Ping_Agent.DateTime	Remote_Ping_Agent.Delay (ms)	Remote_Ping_Agent.Duplicated	Remote_Ping_Agent.ExecutionId	Remote_Ping_Agent.ICMP Seq	Ret
09/03/2021 17:06:56	09/03/2021 17:06:56	23.80	false	test-06-02		1.00
09/03/2021 17:06:57	09/03/2021 17:06:57	22.90	false	test-06-02		2.00
09/03/2021 17:06:58	09/03/2021 17:06:58	22.90	false	test-06-02		3.00
09/03/2021 17:06:59	09/03/2021 17:06:59	22.90	false	test-06-02		4.00
09/03/2021 17:07:00	09/03/2021 17:07:00	22.80	false	test-06-02		5.00
09/03/2021 17:07:01	09/03/2021 17:07:01	22.90	false	test-06-02		6.00

SELECT \* FROM "integration"."autogen"."Remote\_Ping\_Agent" Where "ExecutionId"='test-06-02'

Figure 25 Linux Ping Agent Results (Athens)

For test-06-03, TCP traffic was sent from Linux Agent in address 10.161.1.127 to a server running at 10.161.1.107 for a duration of 10 seconds. Request from OpenTAP to Agent console is shown in Figure 26, followed by OpenTAP execution logs and results recorded in influxDB under ExecutionId 'test-06-03' in Figure 27.

```

ubuntu@agents:~/Remote_iPerf_agents$ ./start.sh
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production NSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://10.161.1.127:5000/ (Press CTRL+C to quit)
Received parameters: {'-c': '10.161.1.107', '-p': '5001', '-i': '1', '-t': '99999'}
Shorted parameters: {'-c': '10.161.1.107', '-p': '5001', '-i': '1', '-t': '99999'}
Final CLI parameters: ['/usr/bin/iperf', '-c', '10.161.1.107', '-p', '5001', '-i', '1', '-t', '99999', '-f', 'm']
Protocol: TCP; Parallel: False (Count: 1); Interval: 1
10.30.0.206 -- [03/Sep/2021 17:25:46] "POST /iperf HTTP/1.1" 200 -
Client running

-----
Client connecting to 10.161.1.107, TCP port 5001
TCP window size: 0.08 MByte (default)

-----
[ 3] local 10.161.1.127 port 34454 connected with 10.161.1.107 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  2118 MBytes 17766 Mbits/sec
[ 3] 1.0- 2.0 sec  2126 MBytes 17838 Mbits/sec
[ 3] 2.0- 3.0 sec  2157 MBytes 18093 Mbits/sec
[ 3] 3.0- 4.0 sec  2214 MBytes 18571 Mbits/sec
[ 3] 4.0- 5.0 sec  2173 MBytes 18232 Mbits/sec
[ 3] 5.0- 6.0 sec  2162 MBytes 18136 Mbits/sec
[ 3] 6.0- 7.0 sec  2083 MBytes 17473 Mbits/sec
[ 3] 7.0- 8.0 sec  2201 MBytes 18465 Mbits/sec
[ 3] 8.0- 9.0 sec  2232 MBytes 18720 Mbits/sec
[ 3] 9.0-10.0 sec  2241 MBytes 18799 Mbits/sec
10.30.0.206 -- [03/Sep/2021 17:25:56] "GET /Close HTTP/1.1" 200 -
[ 3] 0.0-10.5 sec  22813 MBytes 18214 Mbits/sec
Client finished.
Last Json Result: [{"timestamp": 1630689946.23116, "throughput": 17766.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689947.23116, "throughput": 17838.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689948.23116, "throughput": 18093.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689949.23116, "throughput": 18571.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689950.23116, "throughput": 18232.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689951.23116, "throughput": 18136.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689952.23116, "throughput": 17473.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689953.23116, "throughput": 18465.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689954.23116, "throughput": 18720.0, "jitter": 0, "packetLoss": 0}, {"timestamp": 1630689955.23116, "throughput": 18799.0, "jitter": 0, "packetLoss": 0}]
10.30.0.206 -- [03/Sep/2021 17:25:57] "GET /LastJsonResult HTTP/1.1" 200 -

```

Figure 26 Linux iPerf Agent Console (Athens)

OpenTAP logs for this execution are shown below:

```

17:25:46.085 TestPlan -----
-----
17:25:46.085 TestPlan Starting TestPlan 'Remote iPerf' on 09/03/2021
17:25:46, 2 of 2 TestSteps enabled.
17:25:46.086 integration Resource "integration" opened. [79.3 us]

```

```

17:25:46.086 iPerfA NCSR Cloud Resource "iPerfA NCSR Cloud" opened. [196
us]
17:25:46.088 TestPlan Saved Test Plan XML [2.91 ms]
17:25:46.143 TestPlan "iPerf Agent" PrePlanRun completed. [50.3 us]
17:25:46.143 TestPlan PrePlanRun Methods completed [70.1 us]
17:25:46.144 TestPlan "Set Execution ID" started.
17:25:46.144 TestStep Setting ExecutionId to test-06-03 (integration)
17:25:46.144 TestPlan "Set Execution ID" completed. [85.8 us]
17:25:46.144 TestPlan "iPerf Agent" started.
17:25:46.144 iPerfA NCSR Cloud Sending request: POST - Iperf
17:25:46.144 iPerfA NCSR Cloud Body: {"-c":"10.161.1.107","-
p":"5001","-i":"1","-t":"99999"}
17:25:56.661 iPerfA NCSR Cloud Sending request: GET - Close
17:25:57.682 iPerfA NCSR Cloud Sending request: GET - LastJsonResult
17:25:57.695 TestPlan "iPerf Agent" completed. [11.5 s]
17:25:57.695 TestPlan Test step runs finished. [11.5 s]
17:25:57.695 TestPlan "iPerf Agent" PostPlanRun completed. [24.4 us]
17:25:57.701 Summary ----- Summary of test plan started 09/03/2021
17:25:46 -----
17:25:57.701 Summary Set Execution ID
85.8 us
17:25:57.701 Summary iPerf Agent
11.5 s
17:25:57.701 Summary -----
-----
17:25:57.701 Summary ----- Test plan completed successfully in
11.6 s -----
17:25:57.707 integration Sending 10 results ('Remote iPerf Agent Client'
as 'Remote_iPerf_Agent_Client') to integration
17:25:57.721 integration Sending 13 log messages to integration
17:25:57.724 integration OnTestPlanRunCompleted for integration. [3.29
ms]
17:25:57.725 integration Resource "integration" closed. [45.1 us]
17:25:57.725 iPerfA NCSR Cloud Resource "iPerfA NCSR Cloud" closed.
[45.2 us]

```

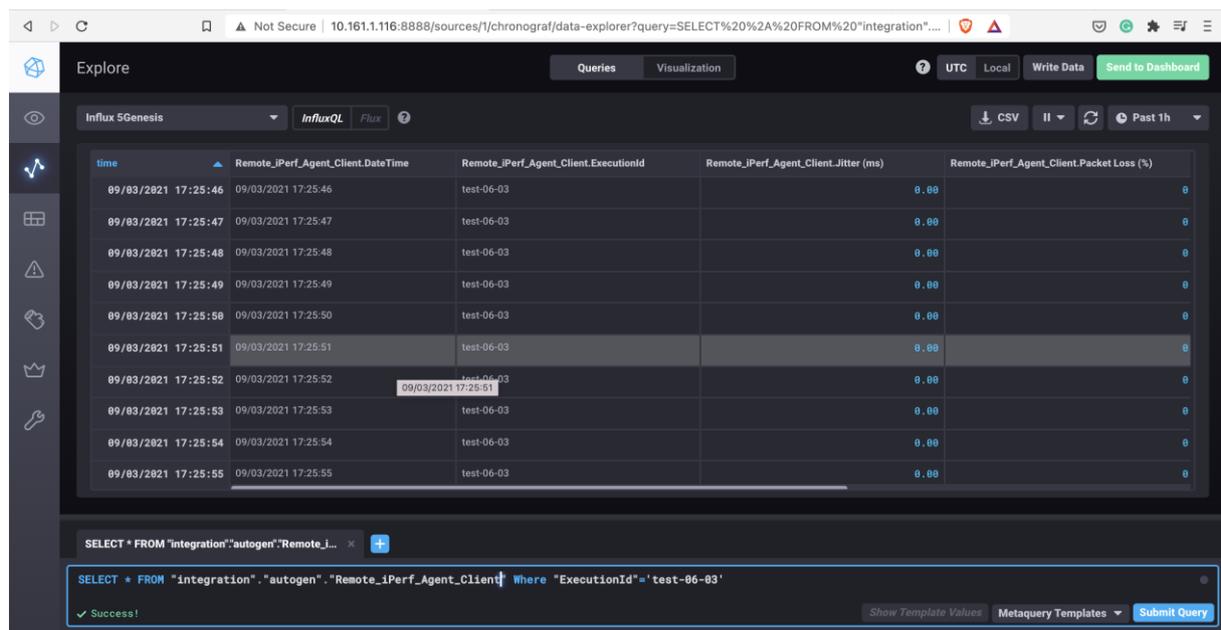


Figure 27 Linux iPerf Agent Results (Athens)

For test-06-04, ICMP requests were sent using Huawei P40 5G UE to address 1.1.1.1. The TapPlan execution logs and results are shown below.

```

16:31:56.235 TestPlan -----
-----
16:31:56.236 TestPlan Starting TestPlan 'ADB Ping' on 09/03/2021
16:31:56, 2 of 2 TestSteps enabled.
16:31:56.236 ADB_LeftMac Resource "ADB_LeftMac" opened. [17.0 us]
16:31:56.236 integration Resource "integration" opened. [70.4 us]
16:31:56.236 ADB_Ping Resource "ADB_Ping" opened. [8.80 us]
16:31:56.238 TestPlan Saved Test Plan XML [2.60 ms]
16:31:56.276 TestPlan PrePlanRun Methods completed [7.00 us]
16:31:56.276 TestPlan "Set Execution ID" started.
16:31:56.276 TestStep Setting ExecutionId to test-06-04 (integration)
16:31:56.276 TestPlan "Set Execution ID" completed. [74.0 us]
16:31:56.276 TestPlan "Adb Ping Agent" started.
16:31:56.276 ADB_LeftMac Executing in background:
..\..\Users\media\Desktop\platform-tools\adb.exe -H 10.30.0.18 -P 5037 -s
K5J0220B24001650 logcat -b main -f sdcard/adb_ping_agent_210903_163156.log
-v threadtime -r 16384 -n 8 ping.Report:I *:S
16:31:56.276 ADB_LeftMac Added new background command; 1 background
command(s)
16:31:56.279 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.ping/com.uma.ping.PingActivity -f 0x20000000
16:31:57.027 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.ping/.PingService -a com.uma.ping.START -e
com.uma.ping.PARAMETERS "target=1.1.1.1,ttl=128,size=56,interval=1" --user
0
16:32:07.252 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.ping/com.uma.ping.PingActivity -f 0x20000000
16:32:07.990 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.ping/.PingService -a com.uma.ping.STOP --user 0
16:32:08.720 ADB_LeftMac Removed background command; 0 background
command(s)
16:32:09.221 TestStep Success -> True, Length: 0
16:32:09.720 TestPlan "Adb Ping Agent" completed. [13.4 s]
16:32:09.720 TestPlan Test step runs finished. [13.4 s]
16:32:09.726 Summary ----- Summary of test plan started 09/03/2021
16:31:56 -----
16:32:09.726 Summary Set Execution ID
74.0 us
16:32:09.726 Summary Adb Ping Agent
13.4 s
16:32:09.726 Summary -----
-----
16:32:09.726 Summary ----- Test plan completed successfully in
13.5 s -----
16:32:09.726 integration Sending 14 log messages to integration
16:32:09.746 integration OnTestPlanRunCompleted for integration. [19.6
ms]
16:32:09.751 integration Resource "integration" closed. [9.80 us]
16:32:09.751 ADB_Ping Resource "ADB_Ping" closed. [23.7 us]
16:32:09.751 ADB_LeftMac Resource "ADB_LeftMac" closed. [10.8 us]

```

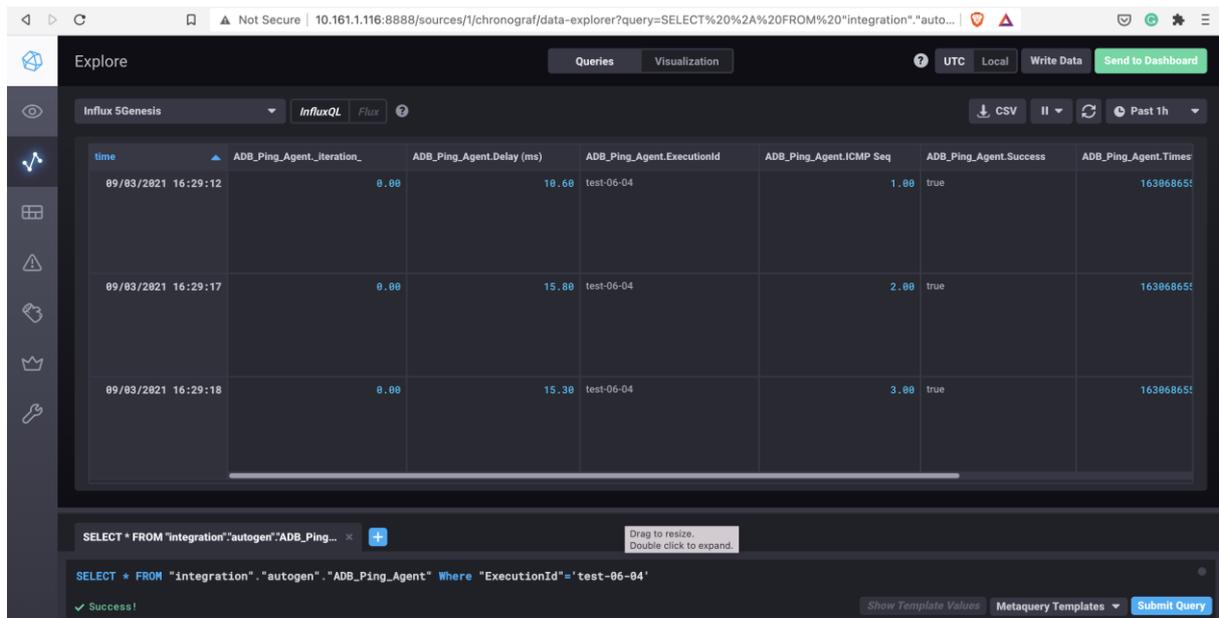


Figure 28 Adb Ping Agent Results (Athens)

For test-06-05, TCP traffic was sent using Huawei P40 5G UE to iPerf server running at 10.30.0.129. The TapPlan execution logs and results are shown below.

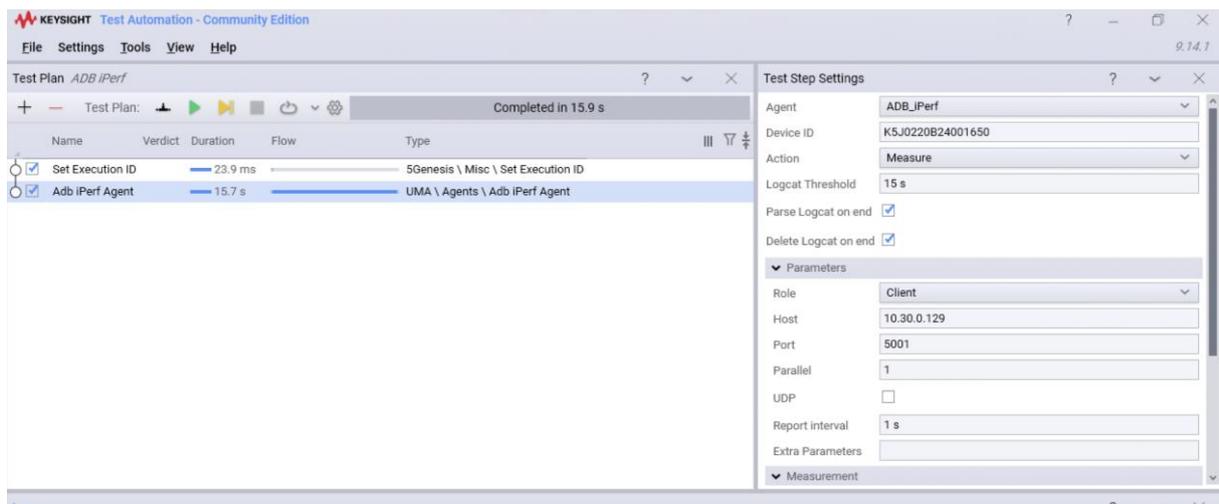


Figure 29 Adb iPerf Agent TapPlan (Athens)

ADB iPerf Agent TapPlan execution logs are show below:

```

16:54:21.367 TestPlan -----
-----
16:54:21.393 TestPlan Starting TestPlan 'ADB iPerf' on 09/03/2021
16:54:21, 2 of 2 TestSteps enabled.
16:54:21.444 TestPlan Saved Test Plan XML [3.15 ms]
16:54:21.511 ADB_LeftMac Resource "ADB_LeftMac" opened. [370 us]
16:54:21.513 ADB_iPerf Resource "ADB_iPerf" opened. [31.3 us]
16:54:21.528 Resolver Found 32/94 assembly files. [16.2 ms]
16:54:21.529 Resolver Found match for InfluxDB.LineProtocol,
Version=1.1.0.0, Culture=neutral, PublicKeyToken=null in C:\Program
Files\OpenTAP\Dependencies\InfluxDB.LineProtocol.1.1.0.0\InfluxDB.LineProto
col.dll
16:54:21.532 integration Resource "integration" opened. [21.6 ms]

```

```

16:54:21.549 TestPlan PrePlanRun Methods completed [3.35 ms]
16:54:21.564 TestPlan "Set Execution ID" started.
16:54:21.582 TestStep Setting ExecutionId to test-06-05 (integration)
16:54:21.590 TestPlan "Set Execution ID" completed. [24.1 ms]
16:54:21.591 TestPlan "Adb iPerf Agent" started.
16:54:21.633 ADB_LeftMac Executing in background:
..\..\Users\media\Desktop\platform-tools\adb.exe -H 10.30.0.18 -P 5037 -s
K5J0220B24001650 logcat -b main -f
sdcard/adb_iperf_agent_client_210903_165421.log -v threadtime -r 16384 -n 8
iperf.Client:I *:S
16:54:21.633 ADB_LeftMac Added new background command; 1 background
command(s)
16:54:21.643 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.iperf/com.uma.iperf.iPerfActivity -f 0x20000000
16:54:22.385 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.iperf/.iPerfService -a com.uma.iperf.CLIENTSTART -e
com.uma.iperf.PARAMETERS "-c,10.30.0.129,-p,5001,-t,999999,-i,1,-f,m" --
user 0
16:54:32.603 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.iperf/com.uma.iperf.iPerfActivity -f 0x20000000
16:54:33.343 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.iperf/.iPerfService -a com.uma.iperf.CLIENTSTOP --
user 0
16:54:34.066 ADB_LeftMac Removed background command; 0 background
command(s)
16:54:34.566 TestStep Success -> True, Length: 0
16:54:34.568 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell "ls
sdcard/adb_iperf_agent_client_210903_165421.log*"
16:54:34.781 ADB_LeftMac Pulling log files:
sdcard/adb_iperf_agent_client_210903_165421.log
16:54:34.781 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 pull
"sdcard/adb_iperf_agent_client_210903_165421.log"
"C:\Users\media\AppData\Local\Temp\g4wjxjos.sns\adb_iperf_agent_client_2109
03_165421.log"
16:54:35.514 TestStep Parsing ADB iPerf Agent Client results from
logcat (starting at 16:54:06). Logcat length: 10
16:54:35.514 TestStep 09-03 16:54:23.660 20142 28364 I iperf.Client:
<<< Timestamp: 1630688063660 ; Output: [ 3] 0.0- 1.0 sec 7.88 MBytes
66.1 Mbits/sec >>>
16:54:35.529 TestStep 09-03 16:54:25.019 20142 28364 I iperf.Client:
<<< Timestamp: 1630688065019 ; Output: [ 3] 1.0- 2.0 sec 2.62 MBytes
22.0 Mbits/sec >>>
16:54:35.529 TestStep 09-03 16:54:25.019 20142 28364 I iperf.Client:
<<< Timestamp: 1630688065019 ; Output: [ 3] 2.0- 3.0 sec 0.00 MBytes
0.00 Mbits/sec >>>
16:54:35.529 TestStep 09-03 16:54:26.279 20142 28364 I iperf.Client:
<<< Timestamp: 1630688066279 ; Output: [ 3] 3.0- 4.0 sec 2.62 MBytes
22.0 Mbits/sec >>>
16:54:35.529 TestStep 09-03 16:54:27.539 20142 28364 I iperf.Client:
<<< Timestamp: 1630688067539 ; Output: [ 3] 4.0- 5.0 sec 2.75 MBytes
23.1 Mbits/sec >>>
16:54:35.529 TestStep 09-03 16:54:28.783 20142 28364 I iperf.Client:
<<< Timestamp: 1630688068783 ; Output: [ 3] 5.0- 6.0 sec 2.62 MBytes
22.0 Mbits/sec >>>

```

```

16:54:35.529 TestStep      09-03 16:54:29.999 20142 28364 I iperf.Client:
<<< Timestamp: 1630688069999 ; Output: [ 3] 6.0- 7.0 sec 2.62 MBytes
22.0 Mbits/sec >>>
16:54:35.529 TestStep      09-03 16:54:30.000 20142 28364 I iperf.Client:
<<< Timestamp: 1630688069999 ; Output: [ 3] 7.0- 8.0 sec 0.00 MBytes
0.00 Mbits/sec >>>
16:54:35.529 TestStep      09-03 16:54:31.191 20142 28364 I iperf.Client:
<<< Timestamp: 1630688071191 ; Output: [ 3] 8.0- 9.0 sec 2.62 MBytes
22.0 Mbits/sec >>>
16:54:35.529 TestStep      09-03 16:54:32.405 20142 28364 I iperf.Client:
<<< Timestamp: 1630688072405 ; Output: [ 3] 9.0-10.0 sec 2.75 MBytes
23.1 Mbits/sec >>>
16:54:35.534 TestStep      Published 10 results, 0 logcat lines ignored
(previous to 16:54:06)
16:54:35.548 integration Sending 10 results ('ADB iPerf Agent Client' as
'ADB_iPerf_Agent_Client') to integration
16:54:36.534 ADB_LeftMac Deleting existing log files:
sdcard/adb_iperf_agent_client_210903_165421.log*
16:54:36.534 ADB_LeftMac Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell "rm -f
sdcard/adb_iperf_agent_client_210903_165421.log*"
16:54:37.254 TestPlan      "Adb iPerf Agent" completed. [15.6 s]

```

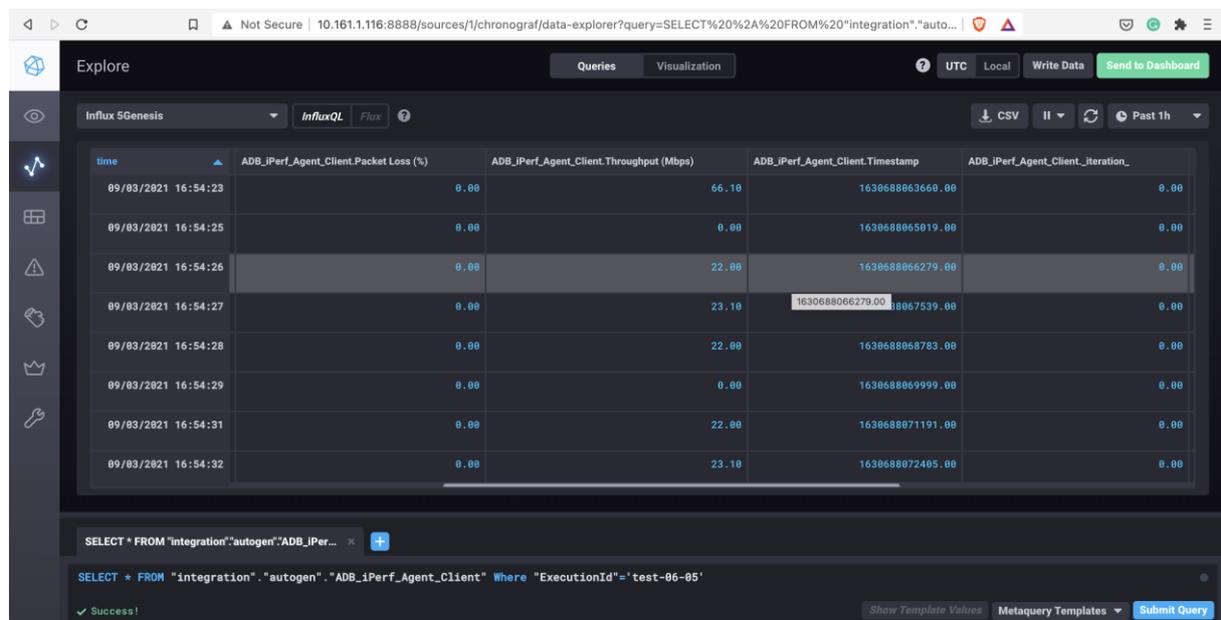


Figure 30 Adb iPerf Agent Results (Athens)

During Test 06-06 ADB Resource Agent TapPlan was executed. Execution logs and test results recorded in influxDB are shown below:

```

12:15:56.191 TestPlan      -----
-----
12:15:56.191 TestPlan      Starting TestPlan 'Resource Agent' on 09/15/2021
12:15:56, 2 of 2 TestSteps enabled.
12:15:56.191 integration Resource "integration" opened. [92.4 us]
12:15:56.191 ADB_LeftMac Resource "ADB_LeftMac" opened. [172 us]
12:15:56.192 ADB_Resource Resource "ADB_Resource" opened. [24.0 us]
12:15:56.195 TestPlan      Saved Test Plan XML [3.52 ms]
12:15:56.229 TestPlan      PrePlanRun Methods completed [5.30 us]
12:15:56.229 TestPlan      "Set Execution ID" started.

```

```
12:15:56.229 TestStep      Setting ExecutionId to test-06-06_
(integration)
12:15:56.229 TestPlan      "Set Execution ID" completed. [69.4 us]
12:15:56.229 TestPlan      "Adb Resource Agent" started.
12:15:56.229 ADB_LeftMac   Executing in background:
..\..\Users\media\Desktop\platform-tools\adb.exe -H 10.30.0.18 -P 5037 -s
K5J0220B24001650 logcat -b main -f
sdcard/adb_resource_agent_210915_121556.log -v threadtime -r 16384 -n 8
resourceAgent.ResourceAgentTask:I *:S
12:15:56.229 ADB_LeftMac   Added new background command; 1 background
command(s)
12:15:56.232 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.resourceAgent/com.uma.resourceAgent.ResourceAgentActivity -f
0x20000000
12:15:56.966 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.resourceAgent/.ResourceAgentService -a
com.uma.resourceAgent.START --user 0
12:16:02.195 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am start -n
com.uma.resourceAgent/com.uma.resourceAgent.ResourceAgentActivity -f
0x20000000
12:16:02.932 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell am
startservice -n com.uma.resourceAgent/.ResourceAgentService -a
com.uma.resourceAgent.STOP --user 0
12:16:03.657 ADB_LeftMac   Removed background command; 0 background
command(s)
12:16:04.157 TestStep      Success -> True, Length: 0
12:16:04.157 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell "ls
sdcard/adb_resource_agent_210915_121556.log*"
12:16:04.372 ADB_LeftMac   Pulling log files:
sdcard/adb_resource_agent_210915_121556.log
12:16:04.372 ADB_LeftMac   Executing: ....\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 pull
"sdcard/adb_resource_agent_210915_121556.log"
"C:\Users\media\AppData\Local\Temp\11151mk5.xhk\adb_resource_agent_210915_1
21556.log"
12:16:05.062 TestStep      Parsing ADB Resource Agent results from logcat
(starting at 12:15:46). Logcat length: 7
12:16:05.062 TestStep      09-15 12:07:51.635 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.284 sec ; Timestamp
1631707670843 ; CPU usage 0.0% ; Ram used 3151MBs ; Available Ram 4260MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.3 ; CQI 2147483647 ; RSRQ -3 >>>
12:16:05.062 TestStep      09-15 12:15:57.482 5288 6089 I
resourceAgent.ResourceAgentTask: Starting Resource Agent task:
resourceAgent<7>
12:16:05.062 TestStep      Could not parse logcat line '09-15 12:15:57.482
5288 6089 I resourceAgent.ResourceAgentTask: Starting Resource Agent task:
resourceAgent<7>'
12:16:05.062 TestStep      09-15 12:15:59.566 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.323 sec ; Timestamp
1631708158523 ; CPU usage 0.0% ; Ram used 3157MBs ; Available Ram 4254MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.4 ; CQI 2147483647 ; RSRQ -3 >>>
```

```

12:16:05.063 TestStep      09-15 12:16:00.609 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.292 sec ; Timestamp
1631708159566 ; CPU usage 0.0% ; Ram used 3157MBs ; Available Ram 4254MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.6 ; CQI 2147483647 ; RSRQ -3 >>>
12:16:05.063 TestStep      09-15 12:16:01.651 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.323 sec ; Timestamp
1631708160609 ; CPU usage 0.0% ; Ram used 3159MBs ; Available Ram 4252MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.6 ; CQI 2147483647 ; RSRQ -3 >>>
12:16:05.063 TestStep      09-15 12:16:02.660 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.321 sec ; Timestamp
1631708161651 ; CPU usage 0.0% ; Ram used 3159MBs ; Available Ram 4252MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.6 ; CQI 2147483647 ; RSRQ -3 >>>
12:16:05.063 TestStep      09-15 12:16:03.437 5288 6089 I
resourceAgent.ResourceAgentTask: <<< Elapsed time 0.284 sec ; Timestamp
1631708162660 ; CPU usage 0.0% ; Ram used 3159MBs ; Available Ram 4252MBs ;
Packets Received 0 ; Packets Transmitted 0 ; Bytes Received 0 ; Bytes
Transmitted 0 ; Operator USIM ; Network LTE ; Cell ID 27447297 ; LAC 1 ;
RSSI 2147483647 ; PSC 0 ; RSRP -77 ; SNR 3.6 ; CQI 2147483647 ; RSRQ -3 >>>
12:16:05.063 TestStep      Published 5 results, 1 logcat lines ignored
(previous to 12:15:46)
12:16:05.063 integration  Sending 5 results ('ADB Resource Agent' as
'ADB_Resource_Agent') to integration
12:16:06.063 ADB_LeftMac  Deleting existing log files:
sdcard/adb_resource_agent_210915_121556.log*
12:16:06.063 ADB_LeftMac  Executing: ..\..\Users\media\Desktop\platform-
tools\adb.exe -H 10.30.0.18 -P 5037 -s K5J0220B24001650 shell "rm -f
sdcard/adb_resource_agent_210915_121556.log*"
12:16:06.780 TestPlan      "Adb Resource Agent" completed. [10.5 s]
12:16:06.780 TestPlan      Test step runs finished. [10.5 s]
12:16:06.786 Summary      ----- Summary of test plan started 09/15/2021
12:15:56 -----
12:16:06.786 Summary      Set Execution ID
69.3 us
12:16:06.786 Summary      Adb Resource Agent
10.5 s
12:16:06.786 Summary      -----
-----
12:16:06.786 Summary      ----- Test plan completed successfully in
10.6 s -----
12:16:06.786 integration  Sending 17 log messages to integration
12:16:06.843 integration  OnTestPlanRunCompleted for integration. [56.5
ms]
12:16:06.843 ADB_Resource  Resource "ADB_Resource" closed. [38.2 us]
12:16:06.843 integration  Resource "integration" closed. [48.9 us]
12:16:06.843 ADB_LeftMac  Resource "ADB_LeftMac" closed. [17.9 us]

```

Time	ADB_Resource_Agent.appname	ADB_Resource_Agent.facility	ADB_Resource_Agent.hostname	ADB_Resource_Agent.CQI	ADB_Resource_Agent.RSRP	ADB_Resource_Agent.RSRQ	ADB_Resource_Agent.RSSI	ADB_Resource_Agent.SNR
09/15/2021 15:07:46 0.00	TAP (0.14.0451e7081e)	ncard	DESKTOP-STK9403	2147483647.00	-77.00	-3.00	2147483647.00	3.38
09/15/2021 15:07:47 0.00	TAP (0.14.0451e7081e)	ncard	DESKTOP-STK9403	2147483647.00	-77.00	-3.00	2147483647.00	3.38
09/15/2021 15:07:48 0.00	TAP (0.14.0451e7081e)	ncard	DESKTOP-STK9403	2147483647.00	-77.00	-3.00	2147483647.00	3.38
09/15/2021 15:07:49 0.00	TAP (0.14.0451e7081e)	ncard	DESKTOP-STK9403	2147483647.00 2147483647.00	-77.00	-3.00	2147483647.00	3.38
09/15/2021 15:07:50 0.00	TAP (0.14.0451e7081e)	ncard	DESKTOP-STK9403	2147483647.00	-77.00	-3.00	2147483647.00	3.38

SELECT \* FROM "integration"."autogen"."ADB\_Res..."

SELECT \* FROM "integration"."autogen"."ADB\_Resource\_Agent" WHERE "ExecutionId"="test-06-06..."

Success!

Figure 31 Resource Agent Results (Athens)

# ANNEX 3: BERLIN PLATFORM INTEGRATION ACTIVITIES

## Details for Test-02-01 to Test-02-03

All three tests run without any surprises. A couple of GUI snapshots and other figures have been provided below. Mainly these are:

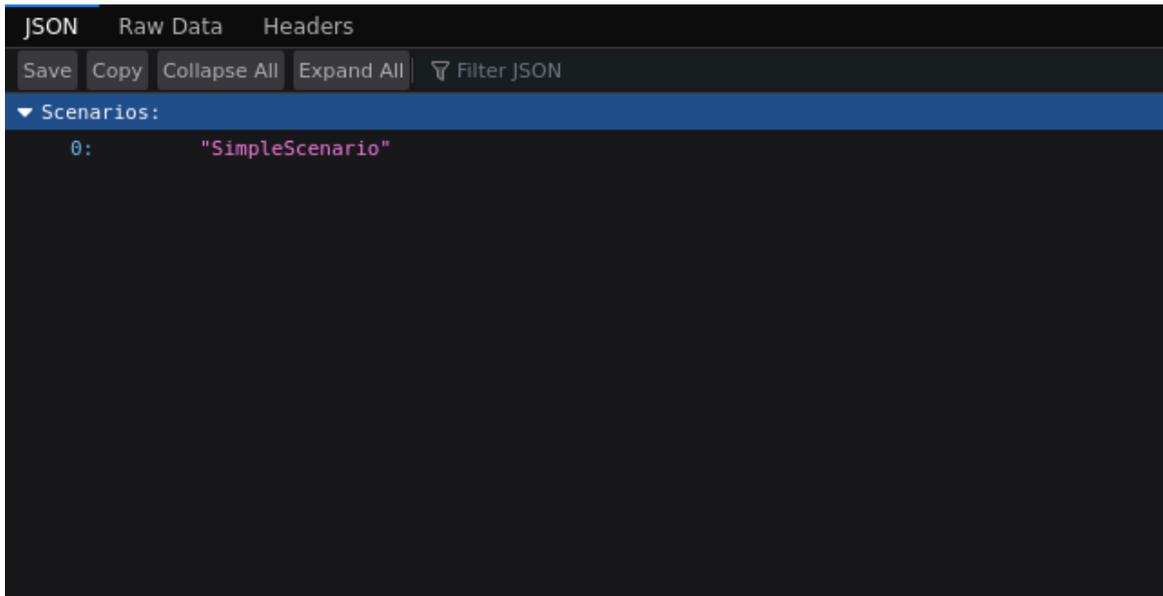
- Figure 32 ELCM Dashboard (Test-02-01)
- Figure 32 ELCM Dashboard (Test-02-01)Figure 36 Returned Experiment result, as JSON (Test 02-03)
- Figure 34 Basic Facility Configuration: Test Case (JSON, Test-02-02)
- Figure 35 Basic Facility Configuration: UE (JSON, Test-02-02)
- Figure 36 Returned Experiment result, as JSON (Test 02-03)

The screenshot displays the ELCM Dashboard interface. At the top, there are navigation tabs for 'Scheduler', 'Log', and 'History'. The main content area is divided into several sections:

- Running Experiments:** Shows a status of '(idle)' and 'Next execution at: 3'.
- Resources:** A horizontal bar with a red segment on the left and a green segment on the right, with a small green square icon in the green segment.
- Diagnostics:** Contains two log panels.
  - Configuration Log:** Shows system logs including 'Logging', 'Portal', 'SliceManager', 'Tap', 'Grafana is disabled', 'InfluxDb is disabled', 'Metadata', and 'Elasticsearch'.
  - Facility Log:** Shows detailed logs for loading resources, test cases, scenarios, and UEs, including configuration details and defined components.

At the bottom of the dashboard, there are two buttons: 'Rebuild configuration' and 'Rebuild facility'.

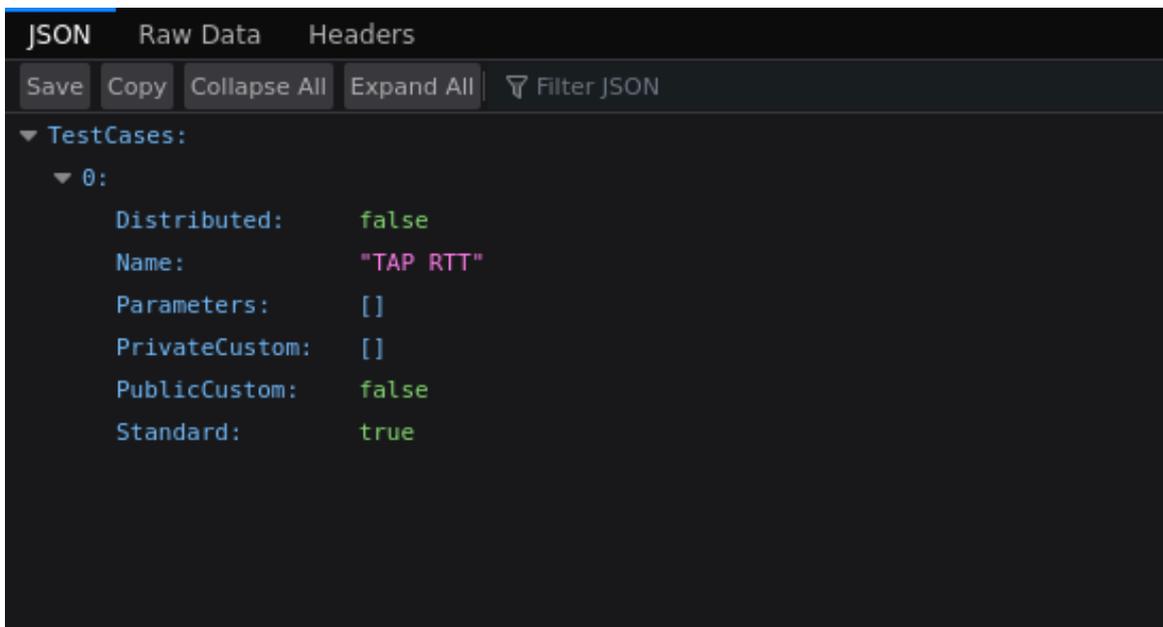
Figure 32 ELCM Dashboard (Test-02-01)



The screenshot shows a JSON viewer interface with tabs for 'JSON', 'Raw Data', and 'Headers'. Below the tabs are buttons for 'Save', 'Copy', 'Collapse All', 'Expand All', and a 'Filter JSON' search box. The main content area displays a JSON object under the heading 'Scenarios:'. The object has a single key '0' with the value 'SimpleScenario'.

```
Scenarios:
  0: "SimpleScenario"
```

Figure 33 Basic Facility Configuration: Scenario (JSON, Test-02-02)



The screenshot shows a JSON viewer interface with tabs for 'JSON', 'Raw Data', and 'Headers'. Below the tabs are buttons for 'Save', 'Copy', 'Collapse All', 'Expand All', and a 'Filter JSON' search box. The main content area displays a JSON object under the heading 'TestCases:'. The object has a key '0' which is expanded to show several properties: 'Distributed' (false), 'Name' ('TAP RTT'), 'Parameters' ([]), 'PrivateCustom' ([]), 'PublicCustom' (false), and 'Standard' (true).

```
TestCases:
  0:
    Distributed: false
    Name: "TAP RTT"
    Parameters: []
    PrivateCustom: []
    PublicCustom: false
    Standard: true
```

Figure 34 Basic Facility Configuration: Test Case (JSON, Test-02-02)

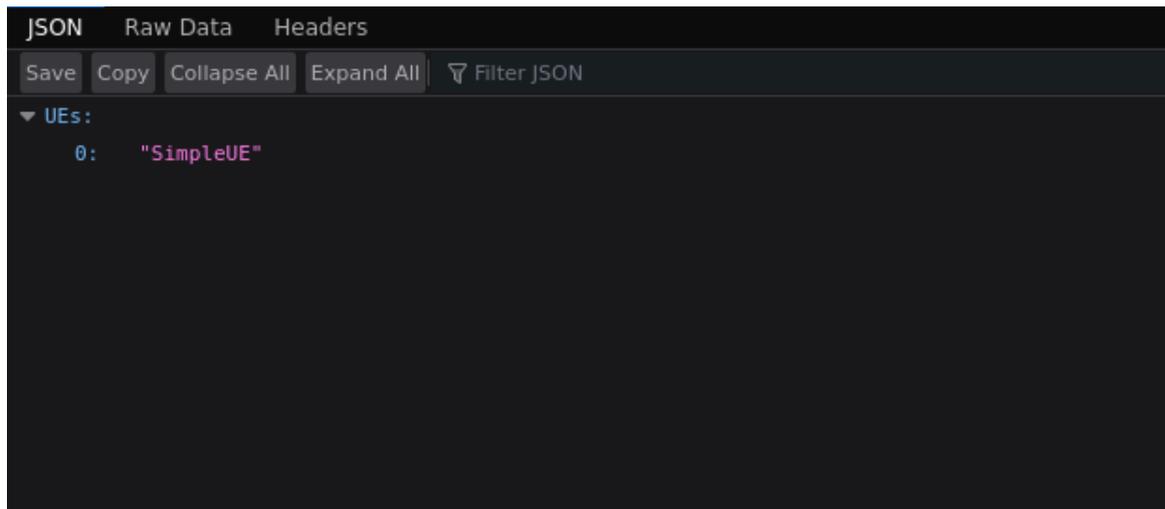


Figure 35 Basic Facility Configuration: UE (JSON, Test-02-02)

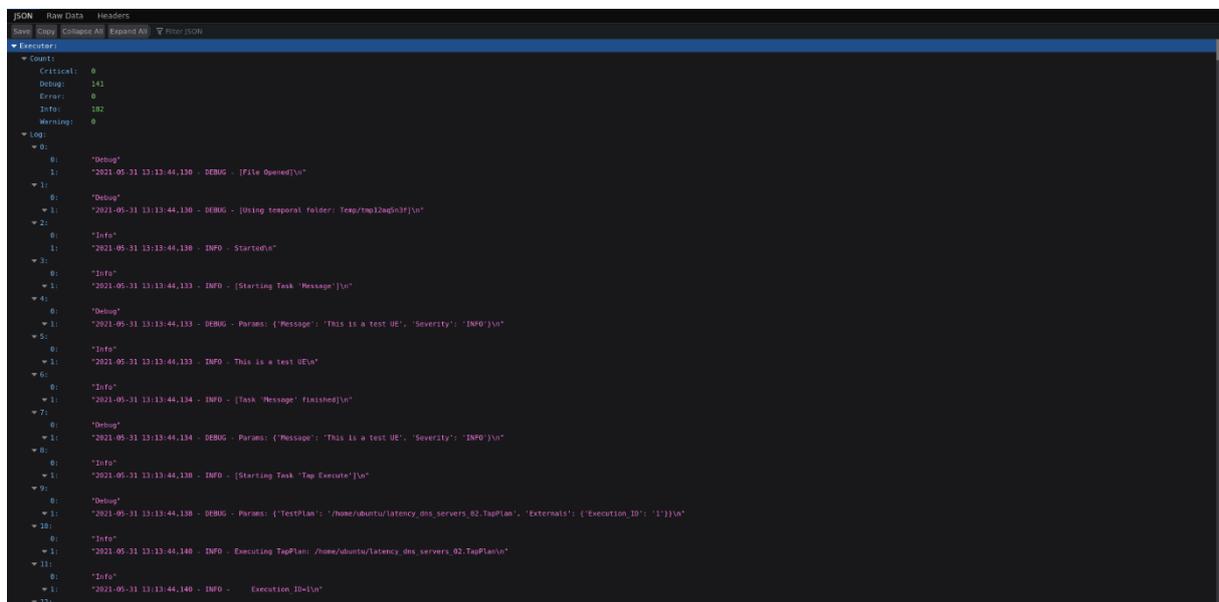


Figure 36 Returned Experiment result, as JSON (Test 02-03)

## Details for Test-03-01

Note: Originally, due to infrastructure upgrades in the Berlin platform, the plan for Berlin was to upgrade – for Release B – from Open Source MANO (OSM) release 8 to release 9 (or newer) and accordingly also for OpenStack. Since 5Genesis-provided integration test script – for Test-03-01 – only support the prior-to-OSM-release-9 NFV descriptor model, these plans need to be dropped and OSM release 8 was used instead.

The tests for 03-01 involve running a robot test script<sup>3</sup>, to assess the functionality of the Dispatcher. The robot test script could be successfully completed using the following component versions:

- OSM release 8
- OpenStack release “Ussuri”

<sup>3</sup> <https://github.com/5genesis/Dispatcher/blob/master/robottest/README.md#execution>

- ELCM version 2.4.3
- Dispatcher Release B (commit 8e667912c8934313369cbfd29e62e5804d008992)
- Katana version 2.3.0 (commit 319dac9ab850ce38d0a165f2f1a84bc1ca2df233)

Figure 37 provides a screenshot of the results of the robot test scripts.

For the Dispatcher Test DISTR\_ED\_VALIDATION (Validate Experiment Descriptor) slight adjustments had to be taken into consideration. To allow a successful pass the presence of certain resources is assumed. Note: the required resources are referenced in the experiment descriptor file: `exp_fixed.json`. Especially, following resources had to be considered:

- A slice with the ID 'sliceid' @ Slice Manager
- two UEs: UE1 (see Listing 2) and UE2 (see Listing 3)
- two TestCases: TC1 (see Listing 4) and TC2 (see Listing 5)
- the Scenario 'scenario1' shown in Listing 1

The screenshot displays a 'Dispatcher Test Report' with the following sections:

- Summary Information:**
  - Status: All tests passed
  - Start Time: 20210723 18:00:14.766
  - End Time: 20210723 18:02:58.095
  - Elapsed Time: 00:02:43.329
  - Log File: [log.html](#)
- Test Statistics:**

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	38	38	0	00:02:43	<div style="width: 100%; height: 10px; background-color: green;"></div>
All Tests	38	38	0	00:02:43	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Dispatcher Test	38	38	0	00:02:43	<div style="width: 100%; height: 10px; background-color: green;"></div>
- Test Details:**
  - Totals | Tags | Suites | Search
  - Type:
    - Critical Tests
    - All Tests

Figure 37 Robot Framework Test Result Report (Test-03-01)

```
scenario1:
  ue_DL_throughput:
    guaranteed: 15000
```

Listing 1 Slice Manager Scenario 1 definition (YAML)

```
UE1:
  - Order: 0
    Task: Run.Message
    Config:
      Message: "This is a test UE: ONE!"
      Severity: INFO
```

Listing 2 Slice Manager User Equipment 1 definition (YAML)

```
UE2:
  - Order: 0
    Task: Run.Message
    Config:
```

```
Message: "This is a test UE: TWO!!"
Severity: INFO
```

Listing 3 Slice Manager User Equipment 2 definition (YAML)

```
TC1:
  - Order: 1
    Task: Run.dummy
    Config:
      ExperimentId: "@{ExperimentId}"
      WaitForRunning: True
      Timeout: 60
      SliceId: "@{SliceId}"
Standard: True
Distributed: False
Dashboard: {}
```

Listing 4 Slice Manager Test Case 1 definition (YAML)

```
TC2:
  - Order: 1
    Task: Run.dummy
    Config:
      ExperimentId: "@{ExperimentId}"
      WaitForRunning: True
      Timeout: 60
      SliceId: "@{SliceId}"
  - Order: 2
    Task: Run.delay
    Config:
      Time: 2
  - Order: 3
    Task: Run.message
    Config:
      Severity: INFO
      Message: Hello 5GENESIS Berlin!
Standard: True
Distributed: False
Dashboard: {}
```

Listing 5 Slice Manager Test case two definition (YAML)

## Details for Test-04-01 to Test-04-04

For Test-04-01, a new Portal user was created during the test that will be allowed to execute experiments. The successful registration of the user is shown in Figure 38. Here, the user is successfully logged-in to the Portal.

The screenshot shows the 5Genesis Portal interface. At the top, there is a navigation bar with the 5Genesis logo, 'Home', 'Create Experiment', 'Network Services', and 'Info' links, and a user profile 'berlino - Logout'. The main content area is titled 'EXPERIMENTS' and contains a table with the following data:

ID	Name	Type	Actions
1	5genesis-local-01	Standard	TestCases: TAP RTT UEs: SimpleUE Run History Descriptor

On the right side, there is a sidebar titled 'ACTIONS' with a list of recent activities:

- 31 May 2021, 3:23:17  
Ran experiment: 5genesis-local-01
- 31 May 2021, 3:13:29  
Ran experiment: 5genesis-local-01
- 31 May 2021, 3:09:50  
Ran experiment: 5genesis-local-01
- 31 May 2021, 3:09:38  
Created experiment: 5genesis-local-01

Figure 38 Portal GUI with successfully registered user (Test-04-01)

For Test-04-02, the newly registered user is capable of executing an experiment. The results of the successful triggering of an experiment is observable in the ELCM component, which is shown in Figure 39.

The screenshot displays the 5Genesis Portal interface. At the top, there is a navigation bar with the 5Genesis logo and links for Home, Create Experiment, Network Services, and Info. The user 'berlino' is logged out. The main heading is 'Execution 1'. Below this is a table with the following data:

Status	Start Time	End Time	Experiment	Action
Finished	31 May 2021, 3:13:34	31 May 2021, 3:18:24	5genesis-local-01	[Icons]

Below the table are two log sections:

**Pre-Run Log**

Buttons: Debug 8, Info 10, Warning, Error, Critical

```

2021-05-31 13:13:34,123 - INFO - Started
2021-05-31 13:13:34,129 - INFO - [Starting Task 'Check Resources']
2021-05-31 13:13:34,129 - INFO - Trying to lock resources
2021-05-31 13:13:34,129 - INFO - Resources available
2021-05-31 13:13:34,129 - INFO - [Task 'Check Resources' finished]
2021-05-31 13:13:34,130 - INFO - [Starting Task 'Instantiate']
2021-05-31 13:13:34,130 - INFO - Instantiation not required, no NSD IDs defined.
2021-05-31 13:13:34,130 - INFO - Instantiation completed
2021-05-31 13:13:34,130 - INFO - [Task 'Instantiate' finished]
2021-05-31 13:13:34,131 - INFO - Finished (status: Finished)

```

**Run Log**

Buttons: Debug 141, Info 182, Warning, Error, Critical

```

2021-05-31 13:13:44,130 - INFO - Started
2021-05-31 13:13:44,133 - INFO - [Starting Task 'Message']
2021-05-31 13:13:44,133 - INFO - This is a test UE
2021-05-31 13:13:44,134 - INFO - [Task 'Message' finished]
2021-05-31 13:13:44,138 - INFO - [Starting Task 'Tap Execute']
2021-05-31 13:13:44,140 - INFO - Executing TapPlan: /home/ubuntu/latency_dns_servers_02.TapPlan
2021-05-31 13:13:44,140 - INFO - Execution_ID=1
2021-05-31 13:13:44,868 - INFO - [TAP]OpenTAP Command Line Interface 9.12.0+78ddca2e
2021-05-31 13:13:44,868 - INFO - [TAP]
2021-05-31 13:13:45,033 - INFO - [TAP]00:00:00.717 : CLI : Information : Loaded test plan from /home/ubuntu/latency_dns_servers_02.TapPlan [142 ms]
2021-05-31 13:13:45,034 - INFO - [TAP]00:00:00.722 : Main : Information : Test Plan: latency_dns_servers_02
2021-05-31 13:13:45,043 - INFO - [TAP]00:00:00.740 : TestPlan : Information : .....
2021-05-31 13:13:45,097 - INFO - [TAP]00:00:00.760 : TestPlan : Information : Starting TestPlan 'latency_dns_servers_02' on 05/31/2021 13:13:45, 6 of 6 TestSteps

```

Figure 39 Portal shows executed results of Experiment 0 (Test-04-02)

For Test-04-03, the registered user is capable to onboard a new Network Service via the Portal. This shows that the Portal-to-Dispatcher integration works properly. Figure 40 shows the Portal GUI during the test.

The screenshot displays the 5Genesis Network Services interface. At the top, there is a navigation bar with the 5Genesis logo and links for Home, Create Experiment, Network Services, and Info. The main content area is titled "Basic Information" and contains a form for configuring a network service. The form includes fields for Name (set to "anyname"), Location (set to "5genesis-openstack"), and Visibility (set to "Private"). There is also a Description field and an "Update" button. A status indicator shows "✓ Network service ready". Below this, the "Virtualized Infrastructure Manager" section shows the "Vim Image" set to "test\_image". The "VNFD Packages" section lists two packages: "hackfest\_1\_vnfd\_fixed.tar.gz" with ID "hackfest1-vnf". An "Available VNFDs" dropdown menu is set to "hackfest1-vnf", with "Add" and "Pre-load" buttons. An "Add VNFD package" button is also present. The "Network Service Descriptor" section shows "hackfest\_1\_nsd\_fixed.tar.gz" with ID "hackfest1-ns".

Figure 40 Network Service Onboarding via Portal (Test-04-03)

For Test-04-04, several resources had to be provisioned in advance since the test itself requires the ELCM to successfully trigger the creation of a slice using the Slice Manager.

The example configurations, as provided with the test case, were used as a basis to provision these resources. A VNFD and NSD were created on the underlying OSM instance. In the Slice Manager, core and radio network functions for the slice and a base slice description, referencing the radio VNF's location, were created. The test was completed, using the same component versions (OSM, OpenStack, ...) as listed in Details for Test-03-01.

Figure 41 shows the creation of the network service provided with the test case on the Portal. The execution view of the Portal – for a completed sliced experiment run – can be seen in Figure 42.

5Genesis Home Create Experiment **Network Services** Info fabian - Logout

### Basic Information

Name	Location	Visibility
<input type="text" value="dummysns"/>	edge	Public

Description

✓ Network service ready

---

### Virtualized Infrastructure Manager

Vim Image: cirros-0.5.2-x86\_64-disk

---

### VNFD Packages

hackfest1-vnf	ID: hackfest1-vnf		
Available VNFDs:	<input type="text" value="hackfest1-vnf"/>	<input type="button" value="Add"/>	
Add VNFD package	<input type="button" value="Browse"/>	<input type="button" value="Pre-load"/>	

---

### Network Service Descriptor

hackfest1-ns	ID: hackfest1-ns	
--------------	------------------	--

Figure 41 Berlin Platform Portal: Creation of a Network Service for Test-04-04

5Genesis

[Home](#)
[Create Experiment](#)
[Network Services](#)
[Info](#)

fabian - Logout

## Execution 43

Status	Start Time	End Time	Experiment	Action
Finished	06 August 2021, 3:42:47	-	todayisagooddayithinkforhorsesontherooftopdancingtogetherwith	

Pre-Run Log

Debug 10
Info 10
Warning
Error
Critical

```

2021-08-06 15:42:47,032 - INFO - Started
2021-08-06 15:42:47,047 - INFO - [Starting Task 'Check Resources']
2021-08-06 15:42:47,047 - INFO - Trying to lock resources
2021-08-06 15:42:47,058 - INFO - Resources available
2021-08-06 15:42:47,058 - INFO - [Task 'Check Resources' finished]
2021-08-06 15:42:47,058 - INFO - [Starting Task 'Instantiate']
2021-08-06 15:42:47,058 - INFO - Experiment contains 1 NSD IDs over Base Slice 'Sample_05G'. Requesting instantiation.
2021-08-06 15:42:47,217 - INFO - Instantiation completed
2021-08-06 15:42:47,218 - INFO - [Task 'Instantiate' finished]
2021-08-06 15:42:47,219 - INFO - Finished (status: Finished)

```

Run Log

Debug 7
Info 8
Warning
Error
Critical

```

2021-08-06 15:42:57,045 - INFO - Started
2021-08-06 15:42:57,054 - INFO - [Starting Task 'Message']
2021-08-06 15:42:57,054 - INFO - This is a test UE
2021-08-06 15:42:57,054 - INFO - [Task 'Message' finished]
2021-08-06 15:42:57,057 - INFO - [Starting Task 'Message']
2021-08-06 15:42:57,057 - INFO - This is a test Test Case
2021-08-06 15:42:57,057 - INFO - [Task 'Message' finished]
2021-08-06 15:42:57,062 - INFO - Finished (status: Finished)

```

Post-Run Log

Debug 8
Info 11
Warning
Error
Critical

```

2021-08-06 15:43:07,062 - INFO - Started
2021-08-06 15:43:07,080 - INFO - [Starting Task 'Decommission']
2021-08-06 15:43:07,080 - INFO - Decommission started
2021-08-06 15:43:07,081 - INFO - Experiment has 1 network services with slice ID: 3b1ddd26-ee4b-49a4-b2f4-26a5f765d690. Requesting decommission
2021-08-06 15:43:07,215 - INFO - Slice decommissioned
2021-08-06 15:43:07,215 - INFO - Decommission completed
2021-08-06 15:43:07,215 - INFO - [Task 'Decommission' finished]
2021-08-06 15:43:07,216 - INFO - [Starting Task 'Release Resources']
2021-08-06 15:43:07,216 - INFO - Releasing resources
2021-08-06 15:43:07,217 - INFO - [Task 'Release Resources' finished]
2021-08-06 15:43:07,218 - INFO - Finished (status: Finished)

```

Figure 42 Berlin Platform Portal: Execution of a sliced experiment (Test-04-04)

## Details for Test-05-01

The results of the execution of this test are reflected through a screenshot of the Portal, see Figure 43.

© 5GENESIS Consortium

Page 79 of 105

The screenshot shows the Berlin Platform Portal interface for 'Execution 1'. At the top, there is a navigation bar with '5Genesis' logo, 'Home', 'Create Experiment', 'Network Services', 'Info', and 'berlino2 - Logout'. Below this is a table with the following columns: Status, Start Time, End Time, Experiment, and Action. The table contains one row with the status 'Init', start time '07 July 2021, 11:29:27', end time '-', and experiment name 'influxtest-01'. Below the table are three log sections: 'Pre-Run Log', 'Run Log', and 'Post-Run Log'. Each log section has a filter bar with buttons for 'Debug', 'Info', 'Warning', 'Error', and 'Critical'. The 'Pre-Run Log' shows a series of INFO messages from 2021-07-07 09:29:14,066 to 09:29:14,120, indicating the start of the process, resource checks, and completion. The 'Run Log' shows INFO messages from 2021-07-07 09:29:24,077 to 09:29:24,482, detailing the start of the 'Csv To Influx' task, conversion of a CSV file to a payload, and its successful execution. The 'Post-Run Log' section is currently empty.

Figure 43 Berlin Platform Portal: Execution result for ELCM-Influx test (Test-05-01)

## Details for Test-06-01

The Monroe VN node used in the Berlin platform is based on a virtual machine and not on a dedicated Monroe hardware node. Since the most current version (as of August 2021) of the Monroe VN installations scripts will fail to install a properly running Monroe VN node, for Ubuntu 18.04.x (Focal) as also for Debian 9.x (Stretch), the installation routine of branch “ReleaseA” was used (<https://github.com/MONROE-PROJECT/monroe-experiment-core/tree/ReleaseA>).

Furthermore, the original OpenTAP test plan, as provided by Test-06-01, caused errors with the specific Monroe node in the Berlin testcase, for unknown reasons. After several failed tryouts, it was decided to adjust the provided OpenTAP testplan for Berlin to something simpler – since Test-06-01 is about OpenTAP-to-MonroeVN integration.

The originally provided OpenTAP testplan was adjusted as follows:

- Removing test steps for start and stop of the original test actions (preventing to execute script “monroe/ping”)
- Inserting a test step to list all experiments existing in the Monroe node

```
<TestStep          type="Tap.Plugins._5Genesis.Monroe.Steps.MonroeListStep"
Version="2.0.2" Id="d8d23283-3cf3-44e9-abef-95af338480b4">
  <Instrument
Source="OpenTap.InstrumentSettings">MONROE_LOCAL</Instrument>
  <VerdictOnError>
    <Value>Error</Value>
    <IsEnabled>>false</IsEnabled>
  </VerdictOnError>
  <Enabled>>true</Enabled>
  <Name>List Experiments</Name>
  <ChildTestSteps />
  <BreakConditions>Inherit</BreakConditions>
  <OpenTap.Description />
</TestStep>
```

By this, the adjusted OpenTAP testplan will list all ever executed test scripts in the node, including failed ones.

After the adjustments taken, the OpenTAP testplan runs through without errors and shows that OpenTAP can interact with Monroe nodes properly. Please note: This also shows that the encrypted OpenTAP-to-MonroeVN channel is configured correctly.

Further notes: Be reminded that in the Berlin platform OpenTAP is used based on the Linux platform. This allows to use only the Text-based UI (TUI) for test configuration, instead of the GUI-based approach, as is suggested by the original testplan of Test-06-01.

### Details for Test-06-02 and Test-06-03

Tests were successfully executed using the following two approaches:

- Using direct command line access to OpenTAP binary
- Executing Tests via Portal (=> ELCM accessing OpenTAP binary)

OpenTAP testplans were provided, regarding RTT and Throughput tests:

- `latency_dns_servers.TapPlan` : ICMP-based latency tests for multiple DNS servers on public Internet and inside Berlin Platform
- `throughput_vm1_to_vm2.TapPlan` : Testing throughput between two hosts in the Berlin platform, leveraging a data stream, based on Iperf

Note: All provided testplans need to be directly accessible by the OpenTAP (“tap”) binary. Additionally, TestCase descriptors were provided to the ELCM, which reference the above listed testplans directly via their path location (using the above shown spelling).

Measurement results were visible directly in the provisioned InfluxDB database, as also accessible via the additionally provided 5Genesis Analytics GUI.

### Details for Test-06-04 to Test-06-06

Since the ADB tests do not play any role in the Berlin platform, any ADB-related tests were ignored.

# ANNEX 4: LIMASSOL PLATFORM INTEGRATION ACTIVITIES

## Test-02-01 - ELCM Dashboard

The screenshot shows the ELCM Dashboard interface. At the top, there are tabs for 'Scheduler', 'Log', and 'History'. Below this, the 'Running Experiments' section shows '(Idle)' and 'Next execution id: 13'. The 'Resources' section features a horizontal bar with a red segment on the left and a green segment on the right, with a small green icon in the green segment. The 'Diagnostics' section includes a 'Configuration Log' with 8 entries and a 'Facility Log' with 8, 11, and 1 entries. Below the logs are two buttons: 'Reload configuration' and 'Reload facility'.

Figure 44 - ELCM Dashboard

## Test-02-02 - Basic facility configuration

The screenshot displays the ELCM logs after adding a Testcase, UE, and Scenario to the facility. The logs are organized into sections: 'Configuration Log' (8 entries) and 'Facility Log' (6, 10 entries). The facility log entries include:

- Debug 6
- Info 10
- Warning
- Error
- Critical

```

Loading Resource: C:\Users\dlioprasitis\Documents\ELCM\Resources\simpleResource.yml
Loading TestCase: C:\Users\dlioprasitis\Documents\ELCM\TestCases\MONROE_Base.yml
ActionInformation [Order: 5; Task: Run.TapExecute; Config: {'TestPlan': '<<Replace with the location of your MONROE_Base testplan.>>', 'Externals': {'Execution ID': '@{ExecutionId}', 'Application': '@{Application}', 'Parameters': '@{JSONParameters}', 'WaitTime': '@{ReservationTimeSeconds}'}}]
Defined 0 dashboard panels
Loading TestCase: C:\Users\dlioprasitis\Documents\ELCM\TestCases\simpleTestCase.yml
ActionInformation [Order: 5; Task: Run.Me@sage; Config: {'Message': 'This is a test Test Case', 'Severity': 'INFO'}]
Defined 0 dashboard panels
Loading UE: C:\Users\dlioprasitis\Documents\ELCM\UEs\simpleUE.yml
ActionInformation [Order: 0; Task: Run.Message; Config: {'Message': 'This is a test UE', 'Severity': 'INFO'}]
Loading Scenario: C:\Users\dlioprasitis\Documents\ELCM\Scenarios\simpleScenario.yml
SimpleScenario: {'ue_DL_throughput': {'guaranteed': 1500000}}
2 TestCases defined on the facility: MONROE_Base, Simple Test Case.
1 UEs defined on the facility: SimpleUE.
2 DashBoards defined on the facility: MONROE_Base, Simple Test Case.
1 Resources defined on the facility: simpleResource.
1 Scenarios defined on the facility: SimpleScenario.
  
```

Figure 45 - ELCM logs after adding Testcase, UE, Scenario to ELCM facility

## Test-02-03 – Experiment execution

### Run

Started: June 14, 2021 2:37 AM (4 minutes ago, waited a few seconds)

Finished: June 14, 2021 2:37 AM (4 minutes ago, ran for a few seconds)

Debug 7
Info 8
Warning
Error
Critical

```

2021-06-14 02:37:19,258 - DEBUG - [File Opened]
2021-06-14 02:37:19,258 - DEBUG - [Using temporal folder: Temp\tmpjuhbehf7]
2021-06-14 02:37:19,258 - INFO - Started
2021-06-14 02:37:19,260 - INFO - [Starting Task 'Message']
2021-06-14 02:37:19,261 - DEBUG - Params: {'Message': 'This is a test UE', 'Severity': 'INFO'}
2021-06-14 02:37:19,261 - INFO - This is a test UE
2021-06-14 02:37:19,261 - INFO - [Task 'Message' finished]
2021-06-14 02:37:19,261 - DEBUG - Params: {'Message': 'This is a test UE', 'Severity': 'INFO'}
2021-06-14 02:37:19,263 - INFO - [Starting Task 'Message']
2021-06-14 02:37:19,263 - DEBUG - Params: {'Message': 'This is a test Test Case', 'Severity': 'INFO'}
2021-06-14 02:37:19,263 - INFO - This is a test Test Case
2021-06-14 02:37:19,264 - INFO - [Task 'Message' finished]
2021-06-14 02:37:19,264 - DEBUG - Params: {'Message': 'This is a test Test Case', 'Severity': 'INFO'}
2021-06-14 02:37:19,264 - INFO - Finished (status: Finished)
2021-06-14 02:37:19,265 - DEBUG - [Closing File]
    
```

Figure 46 - ELCM logs during experiment execution

## Test-05-01 – ELCM - Influx integration

```

> use tapdb
Using database tapdb
> select * from InfluxDbTestResults order by desc limit 10
name: InfluxDbTestResults
time
-----
ExecutionId Jitter (ms) Name Packet Loss (%) PlanName ResultType StepDuration Throughput (Mbps) appname host hostname
-----
1606410036129000000 1 1.511 Adb iPerf Agent 55 Untitled ADB iPerf Agent Server 0 89.1 ELCM 127.0.0.1 opentap
1606410034917000000 1 0.047 Adb iPerf Agent 65 Untitled ADB iPerf Agent Server 0 65.4 ELCM 127.0.0.1 opentap
1606410033912000000 1 0.077 Adb iPerf Agent 56 Untitled ADB iPerf Agent Server 0 27.2 ELCM 127.0.0.1 opentap
1606410032909000000 1 0.099 Adb iPerf Agent 99 Untitled ADB iPerf Agent Server 0 16.9 ELCM 127.0.0.1 opentap
1606410032298000000 1 1.893 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 6.14 ELCM 127.0.0.1 opentap
1606410029883000000 1 0.467 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 7.32 ELCM 127.0.0.1 opentap
1606410028880000000 1 0.375 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 7.29 ELCM 127.0.0.1 opentap
1606410027876000000 1 0.456 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 7.39 ELCM 127.0.0.1 opentap
1606410026871000000 1 0.434 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 7.41 ELCM 127.0.0.1 opentap
1606410025864000000 1 0.456 Adb iPerf Agent 0 Untitled ADB iPerf Agent Server 0 7.45 ELCM 127.0.0.1 opentap
    
```

Figure 47 - InfluxDB entries sent from ELCM

## Test-04-01 – Portal connectivity with other components

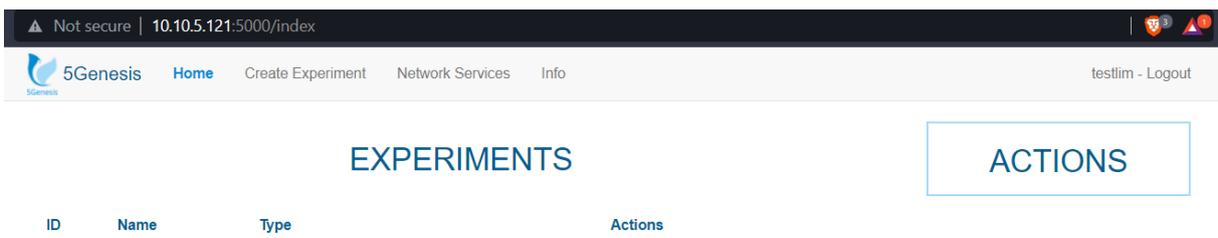


Figure 48 - Portal empty dashboard after user login

## Test-04-02 – Experiment execution through the portal

The screenshot shows the 5Genesis portal interface. At the top, there is a navigation bar with the 5Genesis logo, 'Home', 'Create Experiment', 'Network Services', and 'Info' links. The user 'testlim' is logged out. The main content area is titled 'EXPERIMENTS' and contains a table with one entry:

ID	Name	Type	Actions
11	test 4-2	Standard	TestCases: Simple Test Case UEs: SimpleUE [Run] [History] [Descriptor]

To the right of the experiments table is an 'ACTIONS' panel with two entries:

- 02 July 2021, 5:26:59  
Ran experiment: test 4-2
- 02 July 2021, 5:26:49  
Created experiment: test 4-2

Figure 49 - Simple experiment creation

The screenshot shows the 'Run Log' section of the 5Genesis portal. It displays a table with the following data:

Status	Start Time	End Time	Experiment	Action
Finished	02 July 2021, 5:27:08	02 July 2021, 5:27:38	test 4-2	[View] [Download] [Print]

Below the table are three log sections:

- Pre-Run Log:** Filtered by 'Warning' (8 items). Log entries include:
  - 2021-07-02 14:27:08,881 - INFO - Started
  - 2021-07-02 14:27:08,894 - INFO - [Starting Task 'Check Resources']
  - 2021-07-02 14:27:08,894 - INFO - Trying to lock resources
  - 2021-07-02 14:27:08,894 - INFO - Resources available
  - 2021-07-02 14:27:08,894 - INFO - [Task 'Check Resources' finished]
  - 2021-07-02 14:27:08,894 - INFO - [Starting Task 'Instantiate']
  - 2021-07-02 14:27:08,895 - INFO - Instantiation not required, base slice not defined.
  - 2021-07-02 14:27:08,895 - INFO - Instantiation completed
  - 2021-07-02 14:27:08,895 - INFO - [Task 'Instantiate' finished]
  - 2021-07-02 14:27:08,899 - INFO - Finished (status: Finished)
- Run Log:** Filtered by 'Warning' (7 items). Log entries include:
  - 2021-07-02 14:27:18,896 - INFO - Started
  - 2021-07-02 14:27:18,902 - INFO - [Starting Task 'Message']
  - 2021-07-02 14:27:18,902 - INFO - This is a test UE
  - 2021-07-02 14:27:18,902 - INFO - [Task 'Message' finished]
  - 2021-07-02 14:27:18,909 - INFO - [Starting Task 'Message']
  - 2021-07-02 14:27:18,910 - INFO - This is a test Test Case
  - 2021-07-02 14:27:18,910 - INFO - [Task 'Message' finished]
  - 2021-07-02 14:27:18,912 - INFO - Finished (status: Finished)
- Post-Run Log:** Filtered by 'Warning' (8 items). (No log entries are visible in this section).

Figure 50 - Simple experiment run logs

## Test-04-03 – Network service onboarding

### Basic Information

Name	Location	Visibility
<input type="text" value="test-4-3 ns onboard"/>	 limassol-core	 Public
Description		
<input type="text"/>		
<input type="button" value="Update"/>	 <b>Network service ready</b>	

### Virtualized Infrastructure Manager

Vim Image: test\_image 

### VNFD Packages

hackfest\_1\_vnfd\_fixed.tar.gz ID: hackfest1-vnf

Available VNFDs:

Add VNFD package

### Network Service Descriptor

hackfest\_1\_nsd\_fixed.tar.gz ID: hackfest1-ns

Figure 51 - Network service onboarding dashboard

### Test-06-01 – TAP-MONROE configuration

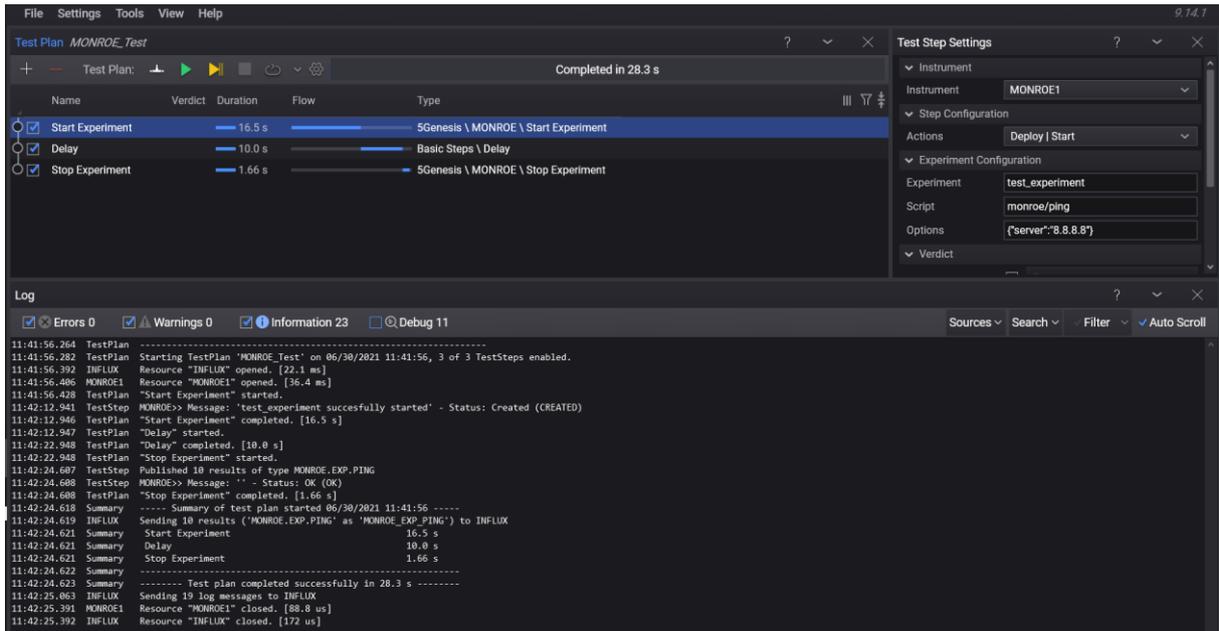


Figure 52 - MONROE TAP test plan

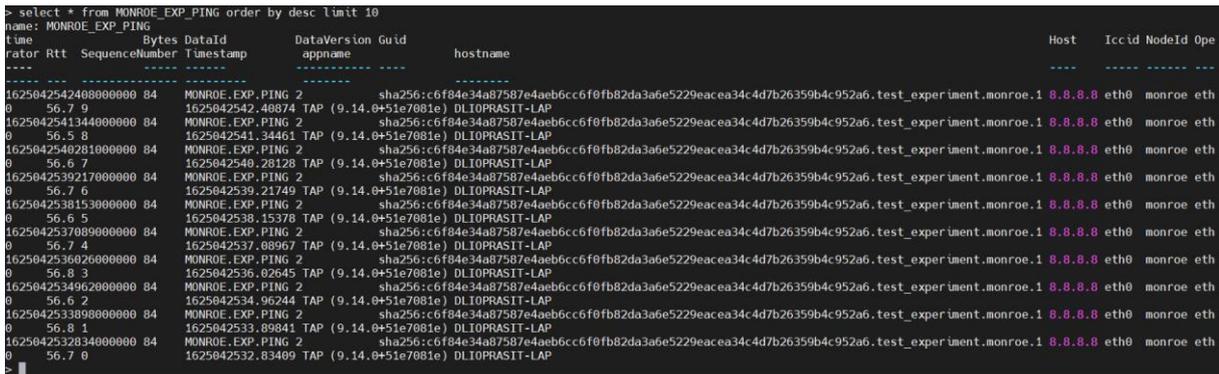


Figure 53 - MONROE test plan entries in Influx DB

## Test-06-02 – TAP-Remote Ping agent

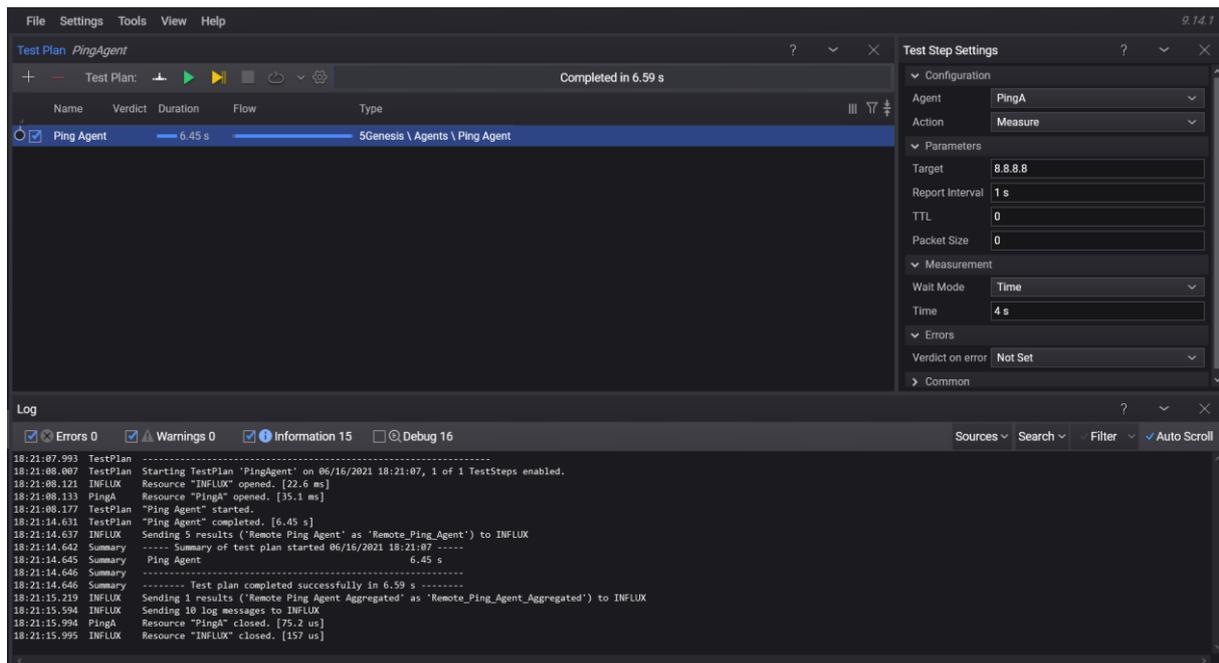


Figure 54 - Ping TAP test plan

## Test-06-03 – TAP-Remote iPerf agent

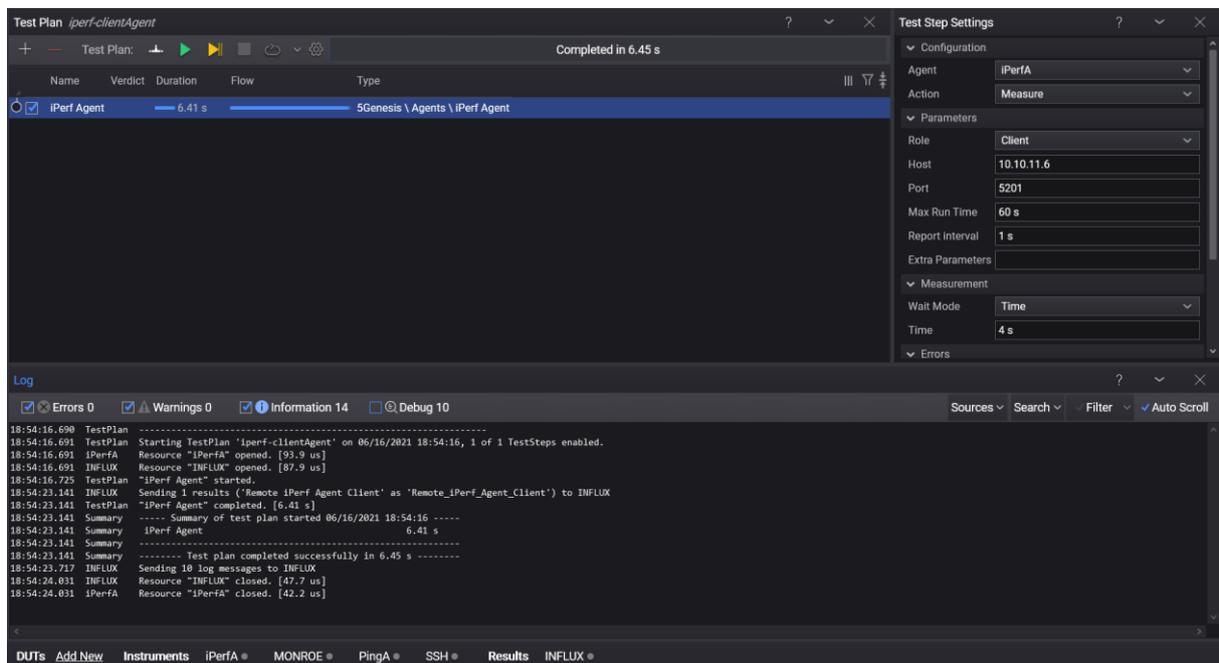


Figure 55 - iPerf TAP test plan

## ANNEX 5: MALAGA PLATFORM INTEGRATION ACTIVITIES

### Test-02-01 - ELCM Dashboard

The warning messages shown on Figure 56 are to be expected due to the lack of facility configuration.

#### Diagnostics

The screenshot displays two log panels. The top panel, titled 'Configuration Log' with a count of 8, shows a filter menu with 'Info' selected. The log content includes: Logging [Folder: Logs; AppLevel: DEBUG; LogLevel: DEBUG], Portal [Host: [REDACTED]; Port: [REDACTED]], SliceManager [Host: [REDACTED]; Port: [REDACTED]], TAP is disabled, Grafana is disabled, InfluxDb [Enabled: True; User: [REDACTED]; Password: [REDACTED]; Database: mydb; Host: [REDACTED]; Port: [REDACTED]], Metadata [HostIp: 127.0.0.1; Facility: None], and EastWest [Enabled: True; Timeout: 120].

The bottom panel, titled 'Facility Log' with counts of 4 and 5, shows a filter menu with 'Warning' selected. The log content lists auto-generated folders: C:\I\E\_Malaga\TestCases, C:\I\E\_Malaga\UEs, C:\I\E\_Malaga\Resources, and C:\I\E\_Malaga\Scenarios. It then reports: No TestCases defined on the facility, No UEs defined on the facility, No DashBoards defined on the facility, No Resources defined on the facility, and No Scenarios defined on the facility.

Figure 56. Initial diagnostics on first run

### Test-02-02 - Basic facility configuration

The screenshot shows the 'Facility Log' panel with counts of 4 and 9. The filter menu has 'Info' selected. The log content shows: Loading Resource: C:\I\E\_Malaga\Resources\simpleResource.yml, Loading TestCase: C:\I\E\_Malaga\TestCases\simpleTestCase.yml, Loading UE: C:\I\E\_Malaga\UEs\simpleUE.yml, Loading Scenario: C:\I\E\_Malaga\Scenarios\simpleScenario.yml, 1 TestCases defined on the facility: Simple Test Case, 1 UEs defined on the facility: SimpleUE, 1 DashBoards defined on the facility: Simple Test Case, 1 Resources defined on the facility: archive, and 1 Scenarios defined on the facility: SimpleScenario.

```

{"Scenarios": [
  "SimpleScenario"
]}

{"TestCases": [
  {

```

```

    "Distributed": false,
    "Name": "Simple Test Case",
    "Parameters": [],
    "PrivateCustom": [],
    "PublicCustom": false,
    "Standard": true
  }
]
}
{"UEs": [
  "SimpleUE"
]}
}

```

Figure 57. Facility logs and endpoint responses after the test

## Test-02-03 - Experiment execution

### Run

Started: April 30, 2021 11:18 AM (a month ago, waited a few seconds)

Finished: April 30, 2021 11:18 AM (a month ago, ran for a few seconds)

[Debug 7](#)
[Info 8](#)
[Warning](#)
[Error](#)
[Critical](#)


```

2021-04-30 11:18:59,604 - INFO - Started
2021-04-30 11:18:59,606 - INFO - [Starting Task 'Message']
2021-04-30 11:18:59,606 - INFO - This is a test UE
2021-04-30 11:18:59,606 - INFO - [Task 'Message' finished]
2021-04-30 11:18:59,608 - INFO - [Starting Task 'Message']
2021-04-30 11:18:59,609 - INFO - This is a test Test Case
2021-04-30 11:18:59,610 - INFO - [Task 'Message' finished]
2021-04-30 11:18:59,610 - INFO - Finished (status: Finished)

```

Figure 58. Logs of the test execution (Run stage only)

## Test-04-01 - Portal connectivity with other components

```

2021-05-03 09:36:39,890 INFO: 1 UEs, 1 Scenarios, 0 Slice Descriptors
2021-05-03 09:36:39,890 INFO: TestCases: 1 standard, 0 public custom, 0 distributed

```



## EXPERIMENTS

ID	Name	Type	Actions
----	------	------	---------

Figure 59. Available entities in the Portal and initial user dashboard

## Test-04-02 - Experiment execution through the Portal

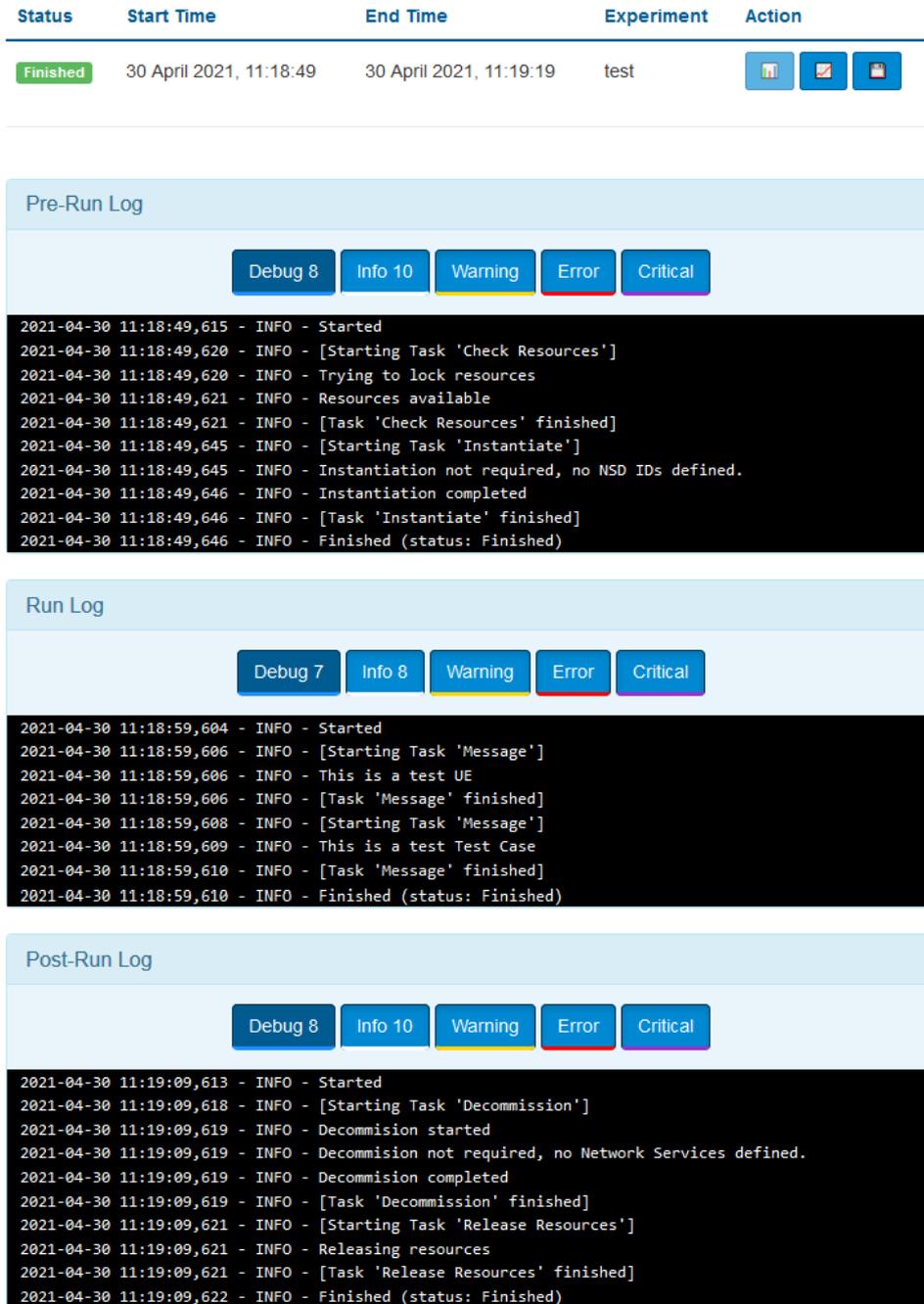


Figure 60. Experiment execution results

### Test-05-01 - ELCM-InfluxDB integration

```

> select * from InfluxDBTestResults limit 5
name: InfluxDBTestResults
time                ExecutionId Jitter (ms) Name                Packet Loss (%)
PlanName ResultType                StepDuration Throughput (Mbps) appname
facility host                hostname
-----
-----
-----
-----
-----
1606408532771000000 20010                0.564                Adb iPerf Agent 94
Untitled ADB iPerf Agent Server 0                17.6                ELCM                UMA
172.23.2.220 DESKTOP-MJQ18JK
                
```

1606408533776000000	20010	0.123		Adb	iPerf	Agent	61
Untitled ADB iPerf Agent Server 0			28.8			ELCM	UMA
172.23.2.220	DESKTOP-MJQ18JK						
1606408534779000000	20010	0.122		Adb	iPerf	Agent	30
Untitled ADB iPerf Agent Server 0			90.1			ELCM	UMA
172.23.2.220	DESKTOP-MJQ18JK						
1606408551751000000	20010	0.274		Adb	iPerf	Agent	38
Untitled ADB iPerf Agent Server 0			36.7			ELCM	UMA
172.23.2.220	DESKTOP-MJQ18JK						
1606408552757000000	20010	1.495		Adb	iPerf	Agent	45
Untitled ADB iPerf Agent Server 0			15.9			ELCM	UMA
172.23.2.220	DESKTOP-MJQ18JK						

## Test-06-01 - TAP-MONROE configuration

11:41:29.364 TestStep Published 10 results of type MONROE.EXP.PING

Figure 61. MONROE agent result generation messages on TAP Log.

```

name: MONROE_EXP_PING
time                               Bytes DataId                               DataVersion Guid
Host      Iccid NodeId                Operator Rtt      SequenceNumber Timestamp
_iteration_ appname                facility host      hostname
-----
-----
-----
1625218877005000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 17.3 0
1625218877.00517 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218878037000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 17.6 1
1625218878.03718 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218879066000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 16.5 2
1625218879.06606 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218880087000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 16.9 3
1625218880.08797 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218881116000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 16.5 4
1625218881.11632 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218882147000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 16.8 5
1625218882.14733 0      TAP (9.15.0+3ed01264) UMA      172.23.2.220
DESKTOP-MJQ18JK
1625218883178000000      84                               MONROE.EXP.PING      2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0 16.7 6

```

```

1625218883.17812 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521888420900000          84          MONROE.EXP.PING          2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0          16.7 7
1625218884.20949 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521888523900000          84          MONROE.EXP.PING          2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0          16.6 8
1625218885.23965 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521888626100000          84          MONROE.EXP.PING          2
sha256:c6f84e34a87587e4aeb6cc6f0fb82da3a6e5229eacea34c4d7b26359b4c952a6.test_experiment.contrib-stretch.1 8.8.8.8 eth0 contrib-stretch eth0          16.5 9
1625218886.26132 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK

```

## Test-06-02 - TAP-Remote Ping agent

```

11:05:05.363 INFLUX    Sending 5 results ('Remote Ping Agent' as 'Remote_Ping_Agent') to INFLUX
11:05:05.400 INFLUX    Sending 1 results ('Remote Ping Agent Aggregated' as 'Remote_Ping_Agent_Aggregated') to INFLUX

```

Figure 62. Ping agent result generation messages on TAP Log.

```

name: Remote_Ping_Agent
time          DateTime          Delay (ms) Duplicated ICMP Seq Success
Timestamp    _iteration_ appname          facility host          hostname
-----
-----
162521665410800000 07/02/2021 09:04:14 16.5          false          1          16.5
1625216654.10873 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521665510800000 07/02/2021 09:04:15 16.5          false          2          16.5
1625216655.10873 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521665610800000 07/02/2021 09:04:16 16.7          false          3          16.7
1625216656.10873 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521665710800000 07/02/2021 09:04:17 16.5          false          4          16.5
1625216657.10873 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK
162521665810800000 07/02/2021 09:04:18 16.6          false          5          16.6
1625216658.10873 0          TAP (9.15.0+3ed01264) UMA          172.23.2.220
DESKTOP-MJQ18JK

name: Remote_Ping_Agent_Aggregated
time          Failed Failed Ratio Success Success Ratio Timestamp
Total _iteration_ appname          facility host          hostname
-----
-----
162521665610800000 0          0          5          1          1625216656.10873
5          0          TAP (9.15.0+3ed01264) UMA          172.23.2.220 DESKTOP-MJQ18JK

```

## Test-06-03 - TAP-Remote iPerf agent

```

11:16:57.931 INFLUX    Sending 4 results ('Remote iPerf Agent Client' as 'Remote_iPerf_Agent_Client') to INFLUX

```

Figure 63. iPerf agent result generation messages on TAP Log.

```

name: Remote_iPerf_Agent_Client
time           DateTime           ExecutionId Jitter (ms) Packet Loss
(%) Throughput (Mbps) Timestamp   _iteration_ appname           facility
host          hostname
-----
-----
-----
162521741540600000 07/02/2021 09:16:55           0           0
20710             1625217415.40633 0           TAP (9.15.0+3ed01264) UMA
172.23.2.220 DESKTOP-MJQ18JK
162521741440600000 07/02/2021 09:16:54           0           0
20035             1625217414.40633 0           TAP (9.15.0+3ed01264) UMA
172.23.2.220 DESKTOP-MJQ18JK
162521741340600000 07/02/2021 09:16:53           0           0
16026             1625217413.40633 0           TAP (9.15.0+3ed01264) UMA
172.23.2.220 DESKTOP-MJQ18JK
162521741240600000 07/02/2021 09:16:52           0           0
20966             1625217412.40633 0           TAP (9.15.0+3ed01264) UMA
172.23.2.220 DESKTOP-MJQ18JK

```

## Test-06-04 - TAP-ADB Ping agent

```

11:54:31.416 INFLUX   Sending 13 results ('ADB Ping Agent' as 'ADB_Ping_Agent') to INFLUX
11:54:31.425 INFLUX   Sending 1 results ('ADB Ping Agent Aggregated' as 'ADB_Ping_Agent_Aggregated') to INFLUX

```

Figure 64. ADB Ping result generation messages on TAP Log.

```

name: ADB_Ping_Agent
time           Delay Delay (ms) ExecutionId ICMP Seq ICMP_Seq Success
Timestamp     _iteration_ appname           facility host           hostname
-----
-----
-----
--
1625046872214000000           38.8           11           true
1625046872214 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046871246000000           45.8           10           true
1625046871246 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046870154000000           61.2           9           true
1625046870154 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046869070000000           41.5           8           true
1625046869070 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046868114000000           41.5           7           true
1625046868114 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046867036000000           40.1           6           true
1625046867036 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK
1625046865970000000           35.9           5           true
1625046865970 0           TAP (9.13.0+a5998b9e) UMA           172.23.2.220 DESKTOP-
MJQ18JK

```

```

1625046865014000000      41.3      4      true
1625046865014 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625046864062000000      53.6      3      true
1625046864062 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625046863015000000      59.6      2      true
1625046863015 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625046861958000000      38.4      1      true
1625046861958 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625046853430000000      40      2      true
1625046853430 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625046852427000000      44.4      1      true
1625046852427 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK

name: ADB_Ping_Agent_Aggregated
time      ExecutionId Failed Failed Ratio Success Success Ratio
Timestamp      Total      _iteration_ appname      facility host
hostname
-----
-----
-----
1625046864900000000      0      0      13      1
1625046864900 13      0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220
DESKTOP-MJQ18JK

```

## Test-06-05 - TAP-ADB iPerf agent

```

12:10:34.728 ADB_Vysor Pulling log files: sdcard/adb_iperf_agent_client_210630_101028.log
12:10:35.299 TestStep Parsing ADB iPerf Agent results from logcat (starting at 12:10:13). Logcat length: 4
12:10:35.309 INFLUX Sending 4 results ('ADB iPerf Agent Client' as 'ADB_iPerf_Agent_Client') to INFLUX
12:10:36.938 TestPlan "Adb iPerf Agent \ Adb iPerf Agent" completed. [8.70 s]
12:10:38.739 ADB_Vysor Pulling log files: sdcard/adb_iperf_agent_server_210630_101027.log
12:10:39.286 TestStep Parsing ADB iPerf Agent Server results from logcat (starting at 12:10:12). Logcat length: 5

```

Figure 65. ADB iPerf result generation messages on TAP Log.

```

name: ADB_iPerf_Agent_Client
time      ExecutionId Jitter (ms) Packet Loss (%) Throughput (Mbps)
Timestamp      _iteration_ appname      facility host      hostname
-----
-----
-----
1625047835791000000      0      0      11107
1625047835791 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625047834914000000      0      0      11137
1625047834914 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625047833858000000      0      0      10571
1625047833858 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK
1625047832810000000      0      0      5188
1625047832810 0      TAP (9.13.0+a5998b9e) UMA      172.23.2.220 DESKTOP-
MJQ18JK

name: ADB_iPerf_Agent_Server

```

time Timestamp	ExecutionId _iteration_	Jitter (ms) appname	Packet Loss (%) facility	Throughput (Mbps) host	hostname
1625047836473000000		0	0	9730	
1625047836473 0	TAP	(9.13.0+a5998b9e)	UMA	172.23.2.220	DESKTOP-MJQ18JK
1625047835876000000		0	0	11124	
1625047835876 0	TAP	(9.13.0+a5998b9e)	UMA	172.23.2.220	DESKTOP-MJQ18JK
1625047834859000000		0	0	11192	
1625047834859 0	TAP	(9.13.0+a5998b9e)	UMA	172.23.2.220	DESKTOP-MJQ18JK
1625047833858000000		0	0	10537	
1625047833858 0	TAP	(9.13.0+a5998b9e)	UMA	172.23.2.220	DESKTOP-MJQ18JK
1625047832891000000		0	0	5201	
1625047832891 0	TAP	(9.13.0+a5998b9e)	UMA	172.23.2.220	DESKTOP-MJQ18JK

## Test-06-06 - TAP-ADB Resource agent

12:29:58.405 INFLUX Sending 12 results ('ADB Resource Agent' as 'ADB\_Resource\_Agent') to INFLUX

Figure 66. Resource agent result generation messages on TAP Log.

```
name: ADB_Resource_Agent
time Available RAM Available RAM (MB) Bytes Received Bytes
Sent CQI Cell ID ExecutionId LAC Network Operator PSC Packets Received
Packets Sent PacketsReceived RSRP RSRQ RSSI SNR Timestamp Total RAM Total
RAM (MB) Used CPU (%) Used CPU Per Cent Used RAM Used RAM (%) Used RAM
(MB) Used RAM Per Cent _iteration_ appname facility host
hostname
```

1625048999230000000		3828		11400	11400
null	null	LTE	RedES	null	19
1625048999230		7640		0	0
49.8952879581152	3812			0	TAP
(9.15.0+3ed01264)	UMA	172.23.2.220	DESKTOP-MJQ18JK		
1625048998185000000		3828		14640	14640
null	null	LTE	RedES	null	24
1625048998185		7640		0	0
49.8952879581152	3812			0	TAP
(9.15.0+3ed01264)	UMA	172.23.2.220	DESKTOP-MJQ18JK		
1625048997135000000		3828		13420	13420
null	null	LTE	RedES	null	22
1625048997135		7640		0	0
49.8952879581152	3812			0	TAP
(9.15.0+3ed01264)	UMA	172.23.2.220	DESKTOP-MJQ18JK		
1625048996085000000		3828		13420	13420
null	null	LTE	RedES	null	22
1625048996085		7640		0	0
49.8952879581152	3812			0	TAP
(9.15.0+3ed01264)	UMA	172.23.2.220	DESKTOP-MJQ18JK		

1625048995037000000		3828		12200	12200
null	null LTE	RedES	null 20		20
1625048995037		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048993986000000		3828		13420	13420
null	null LTE	RedES	null 22		22
1625048993986		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048992935000000		3828		13420	13420
null	null LTE	RedES	null 22		22
1625048992935		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048991883000000		3828		13420	13420
null	null LTE	RedES	null 22		22
1625048991883		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048990833000000		3828		10932	10932
null	null LTE	RedES	null 18		18
1625048990833		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048989787000000		3828		2424	2424
null	null LTE	RedES	null 4		4
1625048989787		7640			0
49.8952879581152 3812				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048970612000000		3822		16288	16288
null	null LTE	RedES	null 23		21
1625048970612		7640			0
49.9738219895288 3818				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			
1625048969567000000		3821		17112	17112
null	null LTE	RedES	null 25		25
1625048969567		7640			0
49.9869109947644 3819				0	TAP
(9.15.0+3ed01264) UMA	172.23.2.220	DESKTOP-MJQ18JK			

# ANNEX 6: SURREY PLATFORM INTEGRATION ACTIVITIES

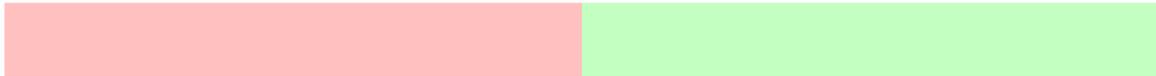
## Test-02-01 - ELCM Dashboard

Running Experiments:

(Idle)

Next execution id: 0

### Resources



### Diagnostics

Configuration Log 8

Debug Info 8 Warning Error Critical

```
Logging [Folder: Logs; AppLevel: INFO; LogLevel: DEBUG]
Portal [Host:      Port:
SliceManager [Host:      Port:
Tap [Enabled: True; OpenTap: True; Exe: tap.exe; Folder: C:/Program Files/OpenTAP; Results: C:/Program Files/OpenTAP/Results; EnsureClosed: True; EnsureAdbClosed:
False]
Grafana [Enabled: True; Bearer: None; ReportGenerator: None; Host:      ; Port:
InfluxDb [Enabled: True; User:      Password:      Database: mydb; Host:      Port:
Metadata [HostIp:      Facility: 6GIC]
EastWest [Enabled: False; Timeout: 120]
```

Facility Log 2 3 3

Debug 2 Info 3 Warning 3 Error Critical

```
Loading TestCase: C:\5genesis\ELCM\TestCases\MONROE_Base.yml
1 TestCases defined on the facility: MONROE_Base.
No UEs defined on the facility.
1 DashBoards defined on the facility: MONROE_Base.
No Resources defined on the facility.
No Scenarios defined on the facility.
```

Reload configuration

Reload facility

Figure 67 Surrey Platform ELCM Dashboard

### Test-02-02 - Basic facility configuration

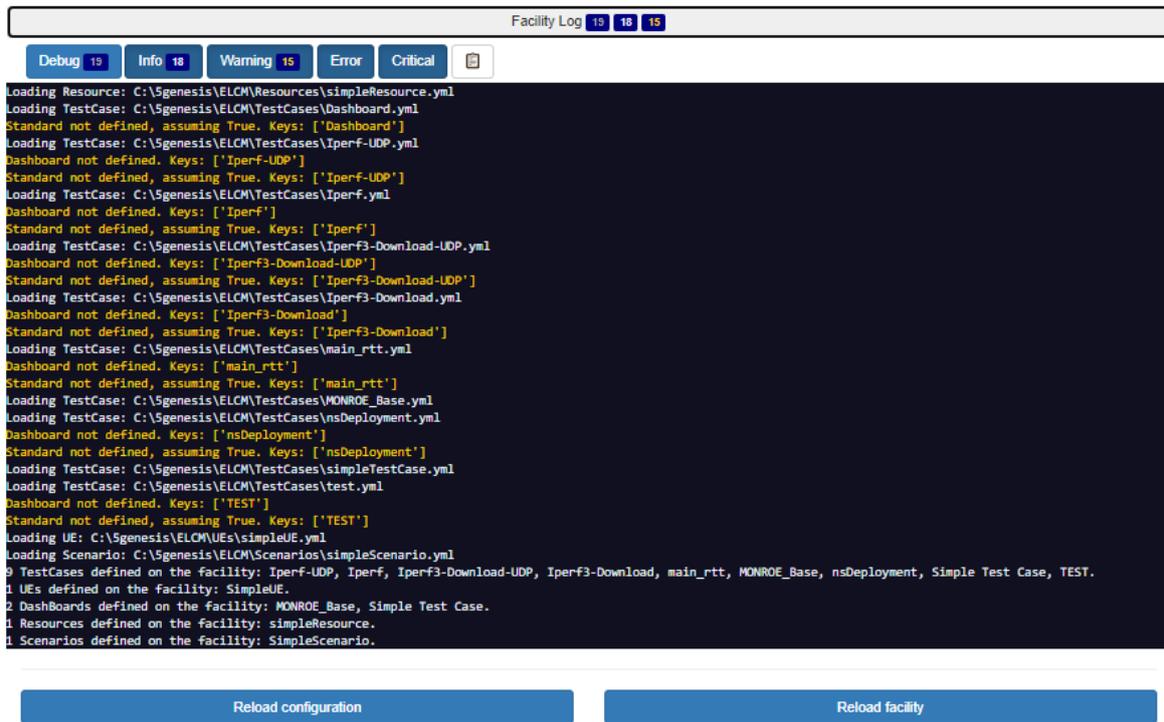


Figure 68 Facility log

### Test-02-03 – Experiment Execution

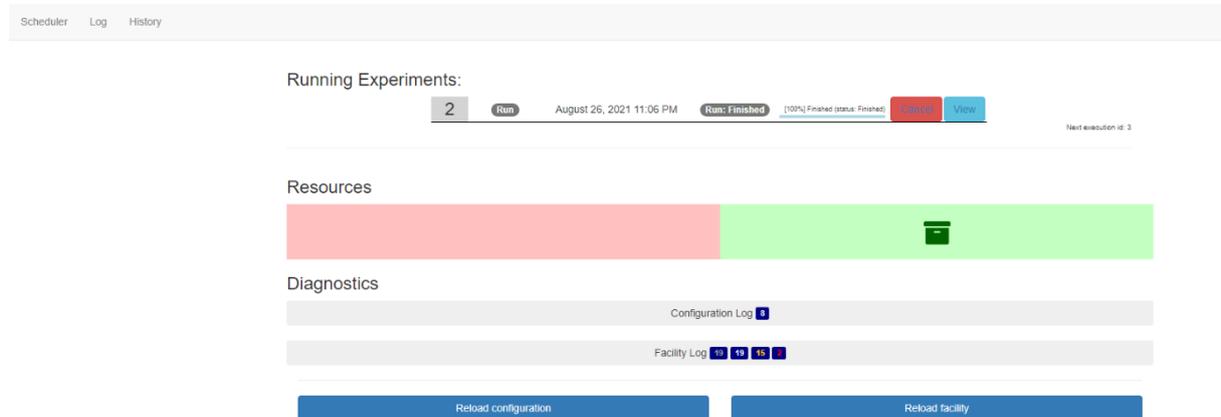


Figure 69 ELCM Dashboard during experiment execution

Scheduler Log History

Status: **Finished** Created: August 26, 2021 11:06 PM (3 minutes ago)

**Pre-Run**  
 Started: August 26, 2021 11:06 PM (3 minutes ago, waited a few seconds) Finished: August 26, 2021 11:06 PM (3 minutes ago, ran for a few seconds)

Debug 5 Info 10 Warning Error Critical

```

2021-08-26 23:06:35,329 - INFO - Started
2021-08-26 23:06:35,332 - INFO - [Starting Task 'Check Resources']
2021-08-26 23:06:35,333 - INFO - Trying to lock Resources
2021-08-26 23:06:35,333 - INFO - Resources available
2021-08-26 23:06:35,333 - INFO - [Task 'Check Resources' finished]
2021-08-26 23:06:35,334 - INFO - [Starting Task 'Instantiate']
2021-08-26 23:06:35,334 - INFO - Instantiation not required, no NSD IDs defined.
2021-08-26 23:06:35,334 - INFO - Instantiation completed
2021-08-26 23:06:35,334 - INFO - [Task 'Instantiate' finished]
2021-08-26 23:06:35,335 - INFO - Finished (status: Finished)
    
```

**Run**  
 Started: August 26, 2021 11:06 PM (3 minutes ago, waited a few seconds) Finished: August 26, 2021 11:06 PM (3 minutes ago, ran for a few seconds)

Debug 7 Info 7 Warning 1 Error Critical

```

2021-08-26 23:06:45,331 - INFO - Started
2021-08-26 23:06:45,332 - INFO - [Starting Task 'Message']
2021-08-26 23:06:45,332 - WARNING - TestCase "Simple Test Case - Surrey 26/08/2021 - Sagar" did not generate any actions
2021-08-26 23:06:45,332 - INFO - [Task 'Message' finished]
2021-08-26 23:06:45,334 - INFO - [Starting Task 'Message']
2021-08-26 23:06:45,335 - INFO - This is a test UE
2021-08-26 23:06:45,335 - INFO - [Task 'Message' finished]
2021-08-26 23:06:45,335 - INFO - Finished (status: Finished)
    
```

Figure 70 Experiment Execution

### Test-03-01 - Dispatcher installation

**Auth** Operations for users in order to access to the different microservices

GET /auth/get\_token Get token by BasicAuth

PUT /auth/change\_password Change Password

POST /auth/register Register User in the platform (But not activated)

Register user in the platform

Parameters Cancel

Name	Description
password * required	string (formData)
username * required	string (formData)
email * required	string (formData)

Execute Clear

Responses Response content type: application/json

```

curl -X POST "https://[redacted]/auth/register" -H "accept: application/json" -H "Content-Type: application/x-www-form-urlencoded" -d "password=testpass&username=testuser&email=sg06340@surrey.ac.uk"
    
```

Responses Response content type **application/json**

Curl

```
curl -X POST "https://[redacted]auth/register" -H "accept: application/json" -H "Content-Type: application/x-www-form-urlencoded" -d [redacted]
```

Request URL

https://[redacted]auth/register

Server response

Code	Details
200	<p>Response body</p> <pre>{   "result": "User registered. Keep an eye with your email for knowing when your account is activated" }</pre> <p>Response headers</p> <pre>content-length: 198 content-type: application/json</pre>

Responses

Code	Description
200	Successful registration
400	Auth error

Figure 71 Dispatcher Swagger API following the addition of a new user

Dispatcher Test Log Generated: 20210617 14:43:53 UTC+01:00  
2 minutes 14 seconds ago REPORT

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	30	34	4	00:02:40	<span style="color: green;">█</span>
All Tests	30	34	4	00:02:40	<span style="color: green;">█</span>

Statistics by Tag

No Tags

Statistics by Suite

Dispatcher Test	Total	Pass	Fail	Elapsed	Pass / Fail
Dispatcher Test	30	34	4	00:02:41	<span style="color: green;">█</span>

Test Execution Log

- 00:02:40:508 - **CRITICAL** Dispatcher Test
- Full Name: Dispatcher Test
- Source: /robot/testing/testsuite/dispatcher\_test.robot
- Start / End / Elapsed: 20210617 14:41:12.522 / 20210617 14:43:53.110 / 00:02:40.588
- Status: 30 critical test, 34 passed, 4 failed, 30 test total, 34 passed, 4 failed
- 00:00:00:591 - **INFO** AUTH\_REG\_1 Register New User
- 00:00:02:915 - **ERROR** AUTH\_REG\_2 Register failed due malformed email
- 00:00:02:813 - **ERROR** AUTH\_REG\_3 Register failed due existing username
- 00:00:02:817 - **ERROR** AUTH\_REG\_4 Register failed due existing email
- 00:00:02:406 - **INFO** AUTH\_VAL\_1 Validate User
- 00:00:02:020 - **INFO** AUTH\_VAL\_2 Validate no existing user
- 00:00:02:398 - **INFO** AUTH\_VAL\_3 Validate User already validated
- 00:00:02:812 - **INFO** AUTH\_SHOW\_1 Show Users (Admin Basic Auth)
- 00:00:05:027 - **INFO** AUTH\_TOK\_1 Get User Token (User Basic Auth)
- 00:00:05:013 - **INFO** AUTH\_TOK\_2 Get User Token (no existing User Basic Auth)
- 00:00:05:027 - **INFO** WRAPPER\_VIM\_LIST\_1 List VIMs (Token Auth)
- 00:00:06:648 - **INFO** WRAPPER\_IMG\_UPL\_1 Upload Image VIM (Token Auth)
- 00:00:05:033 - **INFO** WRAPPER\_IMG\_UPL\_2 Upload existing Image VIM (Token Auth)
- 00:00:06:585 - **INFO** WRAPPER\_IMG\_UPL\_3 Upload wrong Image VIM (Token Auth)
- 00:00:05:816 - **INFO** WRAPPER\_IMG\_DEF\_1 Register VIM image (Admin Basic Auth)

Figure 72 Dispatcher log

## Test-04-01 - Portal connectivity with other components

```
2021-06-24 08:26:09,545 - INFO - 5Genesis startup
[2021-06-24 10:09:10,820 - DEBUG - [50820845] >> [POST] https://[redacted]auth/register
2021-06-24 10:09:10,820 - DEBUG - [50820845] >> Headers: {'Content-Type': 'application/x-www-form-urlencoded'}
2021-06-24 10:09:10,822 - DEBUG - [50820845] >> Body: username=[redacted]sho250:2a0000$Md5FvtSaER8j0n4F$21a0131132becab0850a8413290dcd0bb09545038ee0f9
2021-06-24 10:09:11,329 - DEBUG - [50820845] << [Code 200] {'result': 'User registered. Keep an eye with your email for knowing when your account is activated'}
```

```

admin5g05N-3130~$ sudo docker container ls
[sudo] password for admin5g:
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                                     NAMES
e8f768a04bd9   swaggerapi/swagger-ui:v3.25.0      "sh /usr/share/nginx..." 6 days ago    Up 6 days    80/tcp, 0.0.0.0:5002->8080/tcp, :::5002->8080/tcp           swagger
299607101ca4   nginx:latest                         "/docker-entrypoint..." 6 days ago    Up 6 days    0.0.0.0:443->443/tcp, :::443->443/tcp, 80/tcp, 0.0.0.0:8082->8082/tcp, :::8082->8082/tcp   dispatcher
796d970995f    mano                                 "python mano.py"          6 days ago    Up 6 days    0.0.0.0:5101->5101/tcp, :::5101->5101/tcp                 mano
1a5f1c023af9   auth                                 "python auth.py"          6 days ago    Up 6 days    0.0.0.0:2000->2000/tcp, :::2000->2000/tcp                 auth
d4890ab9b424   distributor                           "python distributor..." 6 days ago    Up 6 days    0.0.0.0:5100->5100/tcp, :::5100->5100/tcp                 distributor
ecf45f70e3f    dispatcher_robottest                 "sh -c 'sleep infini..." 6 days ago    Up 6 days    8011/tcp, 0.0.0.0:8200->80/tcp, :::8200->80/tcp            robottest
39c9cebeb124   mongo:latest                         "docker-entrypoint.s..." 6 days ago    Up 6 days    0.0.0.0:27017-27019->27017-27019/tcp, :::27017-27019/tcp  database
    
```

Figure 73 Addition of new user

```

Portal.log x  config.yml x
1  Logging:
2  Folder: 'Logs'
3  AppLevel: INFO
4  LogLevel: DEBUG
5  Dispatcher:
6  Host: ██████████
7  Port: ████████
8  TokenExpiry: 240
9  ELCM:
10 Host: ██████████
11 Port: ████████
12 Grafana URL: http://localhost:██████
13 Platform: 5GenesisSurreyPlatform
14 PlatformDescriptionPage: platform.html
15 Description: 5th Generation End-to-end Network, Experimentation, System Integration, and Showcasing
16 EastWest:
17 Enabled: False
18 Remotes: {} # One key for each remote Portal, each key containing 'Host' and 'Port' values
19 Analytics:
20 Enabled: False
21 URL: <Internet address>/dash # External URL of the Analytics Dashboard
22 Secret: # Secret key shared with the Analytics Dashboard, used in order to create secure URLs
23
    
```

Figure 74 Portal config.yml file

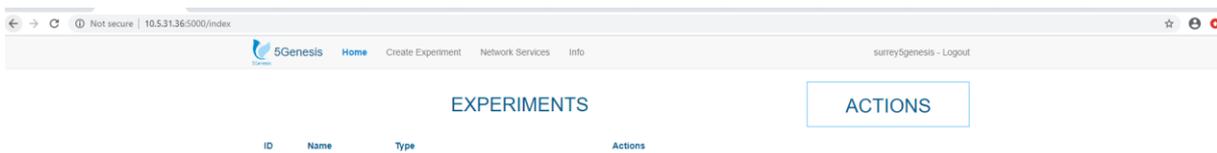


Figure 75 Portal initial user dashboard

## Test-04-02 - Experiment execution through the Portal

5Genesis Home Create Experiment Network Services Info test2308 - Logout

Experiment 2: test2608  
· Type: Standard

Run Experiment View descriptor

### EXECUTIONS

Execution ID	Status	Start Time	End Time	Action
3	Run Task Tap Execute finished	26 August 2021, 11:20:25	-	
1	Finished	26 August 2021, 10:08:15	26 August 2021, 10:08:45	

5Genesis Home Create Experiment Network Services Info test2308 - Logout

5Genesis Home Create Experiment Network Services Info test2308 - Logout

Experiment 2: test2608  
· Type: Standard

Run Experiment View descriptor

### EXECUTIONS

Execution ID	Status	Start Time	End Time	Action
3	Finished	26 August 2021, 11:20:25	26 August 2021, 11:20:55	
1	Finished	26 August 2021, 10:08:15	26 August 2021, 10:08:45	

5Genesis Home Create Experiment Network Services Info test2308 - Logout

Figure 76 Experiment execution through the portal





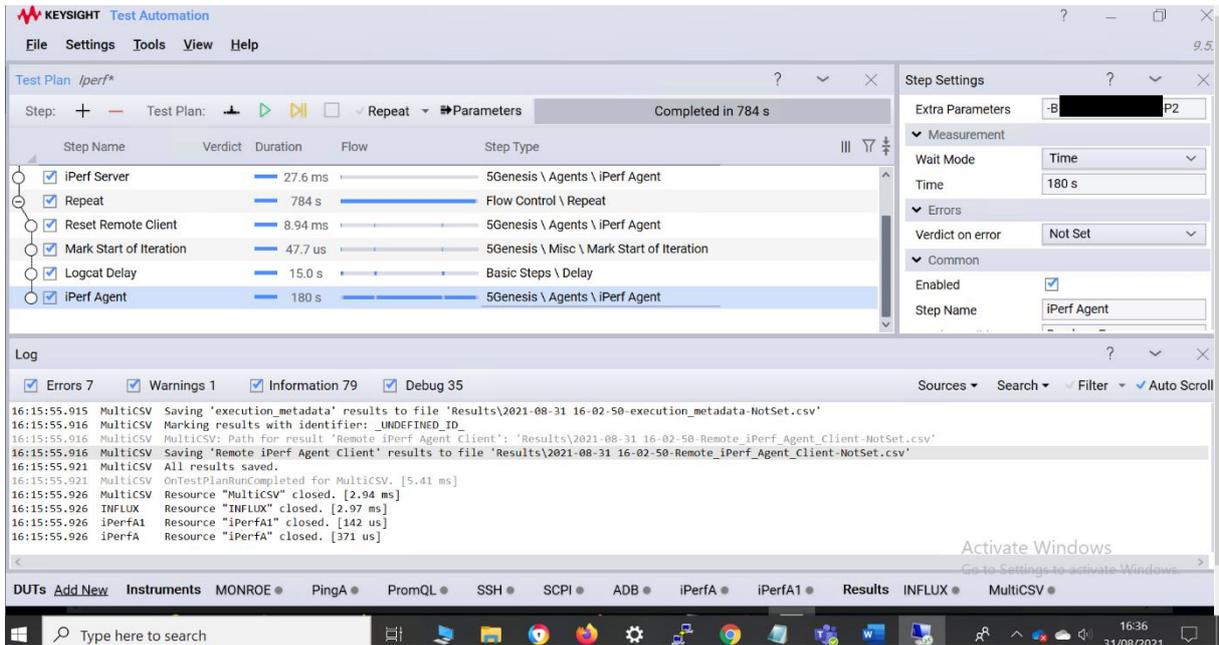


Figure 80 OpenTAP iPerf test plan execution

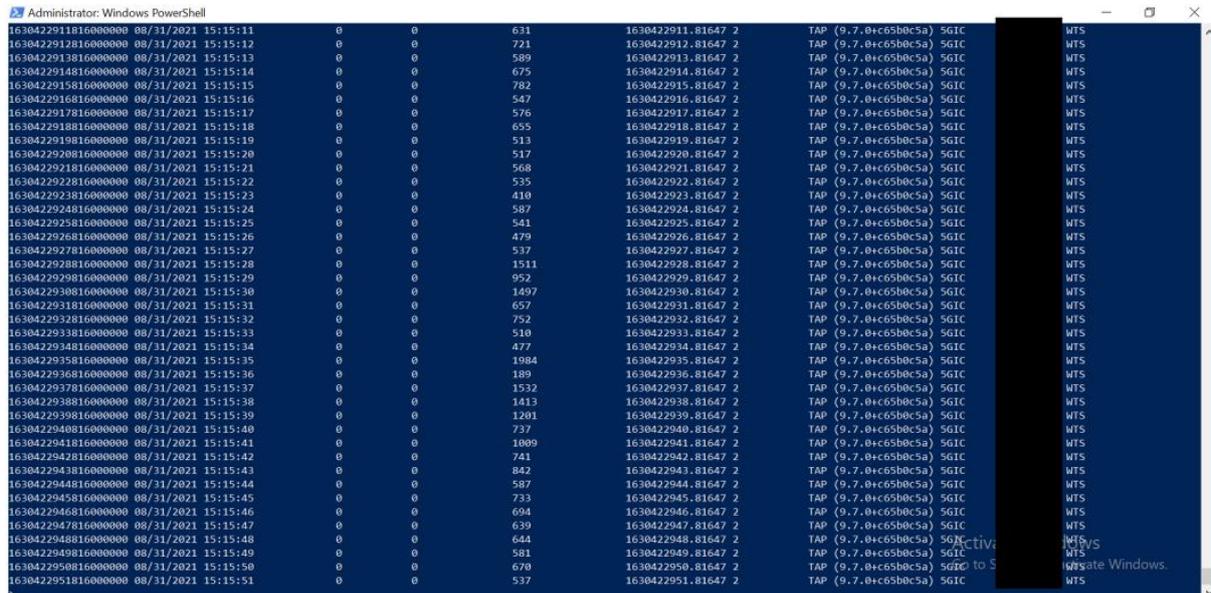


Figure 81 Remote iPerf agent execution