5Genesis

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

[H2020 - Grant Agreement No. 815178]

Deliverable D5.1

System-Level Tests and Verification

Editor D. Triantafyllopoulou (UNIS)

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

Version 1.0

Date March 3rd, 2020

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

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

Version History

Rev. N	Description	Author	Date
1.0	Release of D5.1	D. Triantafyllopoulou (UNIS)	03/03/2020

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
TAP	Test Automation Platform
UE	User Equipment
VIM	Virtual Infrastructure Manager
VNFD	Virtual Functions Descriptor
VNFR	VNF Record
VPN	Virtual Private Network
Watir	Web Application Testing in Ruby
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 A' WP3 implementations.

To this end, the integration workflow, which consists of three phases, is introduced. The first phase is carried out in the development environment, to delivers the components to be integrated in each release cycle. Then, the second integration phase, executed in a dedicated integration environment in the Athens Platform, performs the deployment, validation and integration of the developed components. Lastly, in the final deployment phase, the validated software release is deployed in each of the 5GENESIS Platforms. For future releases of the software components, appropriate test automation tools are also considered. The integration of the individual components follows a Git based methodology, 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.

The 5GENESIS Coordination Layer provides the experimenters with the necessary tools in order to use the Platforms for executing their experiments. These include the means for the definition and automatic control of the life cycle of an experiment, the storage of the respective experimentation results, and the automated communication with the lower layers of the 5GENESIS architecture for the execution of the experiments. This deliverable also includes a brief overview of the individual components of the 5GENESIS Coordination Layer. These include: i) the Experiment Lifecycle Manager, for the overseeing of the experiment, ii) the Monitoring and Analytics module for the analysis of the raw data collected during an experiment, iii) the Portal, which provides the main interface to the experimenters, and iv) the Slice Manager. The Coordination Layer has three south-bound interfaces that are used for its interconnection with the lower layers of the 5GENESIS architecture.

This deliverable covers the integration of coordination layer components and the "slice-manager" i.e. the south-bound interfaces (but excludes integration with MANO & infrastructure, which are reported in WP3 deliverables). The Slice Manager, although not part of the Coordination Layer, is vital for the abstraction of the underlying infrastructure and as such the deployment and integration is also part of this deliverable.

The integration between platforms via each-west interface, based on extension of OpenAPI, is currently in progress, as part of Phase 3 activities.

The installation, integration and testing of the Release A of the individual components has taken place in a dedicated integration and testing environment, which was created in the Athens Platform. All partners involved in the integration activities have access to this environment via a Virtual Private Network (VPN) connection.

The validation of the components' integration was performed via well-defined integration tests that were used for testing the proper operation of the installed components, as well as their communication. An end-to-end experiment lifecycle test was also created, in order to perform end-to-end testing of the full experimentation cycle. The results of the integration testing per Platform at the time of the deliverable submission are also reported.

Table of Contents

LIST OF FIGURES

List of Tables

1. Introduction

1.1. Purpose of the document

This deliverable reports on the activities of the first two tasks of 5GENESIS WP5, "System-level Verifications and Documentation" i.e. "End-to-End facility integration" (Task 5.1) and "System-level tests and verification" (Task 5.2). The objective of this work is to carry out the integration of the individual components that constitute the 5GENESIS coordination layer plus slice manager components, as well as define and conduct the respective integration testing, resulting in the deployment of Release A components in all 5GENESIS platforms.

Firstly, the necessary development and integration workflow for the delivery of the integrated 5GENESIS Facility is described. To this end, 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 are provided. Automation tools for the extension and automation of the integration testing of future releases of the Facility are also evaluated. The individual components, developed in the context of WP3, are collected from all repositories and are installed, tested and integrated in a controlled integration and testing environment in the Athens Platform.

A high-level overview of Release A of the 5GENESIS Coordination Layer and Slice Manager are also provided. A brief discussion on its features is made, while its main functional components are introduced. Emphasis is given on its south-bound interfaces that are necessary for its interconnection with the underlying components of the 5GENESIS architecture.

Moreover, this deliverable also reports on the WP5 activities regarding the testing and verification of the overall 5GENESIS Facility. More specifically, a set of tests was defined with the aim to validate the component integration of the 5GENESIS Facility Release A. The tests were carried out against concretely defined test cases, following the template of ETSI NFV, with specific pre-defined sequences and success criteria, ensuring that the requirements set out in WP2 were properly met. Finally, a report on the progress of the integration activities in each Platform at the time of the deliverable submission is also provided.

1.2. Structure of the document

This deliverable is structured as follows:

- Section 2 describes the overarching 3-phased 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 practise guides are outlined.
- Section 3 provides a description of the 5GENESIS Coordination Layer and Slice manager, firstly by introducing its main features and components, and then by defining its south-bound interfaces that are used for its interconnection with the lower layers of the 5GENESIS architecture.
- 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 A components.

- Section 5 defines the tests used to validate the integration of the Release A of the Coordination Layer components.
- Section 6 provides the results of the testing and validation activities.
- Section 7 provides concluding remarks.
- Finally, Annex 1 reports on the results of the integration tests for the Platforms that already have proceeded with the integration of the different components of the 5GENESIS Coordination Layer.

1.3. Target audience

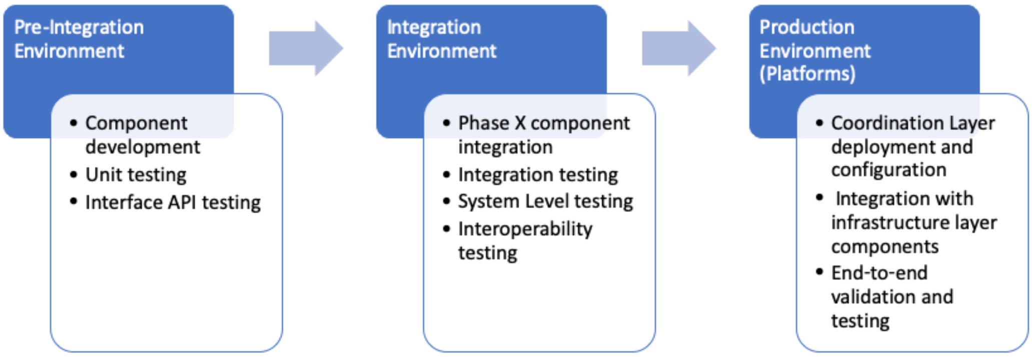
The target audience of this deliverable includes the ICT professionals or research projects who are interested in performing experimentations, the European Commission, who can use this document as a means for the evaluation of the activities of the Platform with regards to the project objectives, as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

2. Overarching 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.


Pre-Integration Environment

- Component development
- Unit testing
- Interface API testing

Integration Environment

- Phase X component integration
- Integration testing
- System Level testing
- Interoperability testing

Production Environment (Platforms)

- Coordination Layer deployment and configuration
- Integration with infrastructure layer components
- End-to-end validation and testing

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., Gitlab) 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 omits 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 maybe utilized in each Platform. There are two possibilities for this final step:

(i) Deploy the resulted integrated components which are provided as pre-packaged virtual machines directly in the compatible virtualization environment of the Platform.

(ii) Deploy each component using the updated documentation and configuration guidelines that are provided by WP5.

Both approaches are validated using the integration tests that have been defined by WP5 during the integration phase. These tests are defined in Section 5.

2.2. Extending and automating integration testing

For future releases (beyond Rel.A) automation of the test process is considered. In order to achieve the automation of the integration activities, the following test automation tools have been evaluated:

- Watir [1] - stands for “Web Application Testing in Ruby” and it is an open source Ruby library for automating tests. Watir interacts with a browser the same way people do: clicking links, filling out forms and validating text.
- Robot [2]- is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data syntax and it utilizes the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either with Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.
- pyTest [3]- is a python-based test framework for testing applications and python libraries. It is used from command line and requires tests to be formatted in a specific way so the framework can identify and execute them.
- Shell - UNIX shell scripting may be used to create testing scripts that use the available Application Programming Interfaces (APIs) to make integration and validation tests.

- jmeter [4]- is a 100% pure Java tool and has an Ubuntu installer in order to be used by command line to perform the tests or via Graphical User Interface. It may be used to test performance both on static and dynamic resources. It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyse overall performance under different load types.
- OpenTAP is a framework for automation that has been used in the automation of the execution of the experiments. This tool can be also used to implement the test cases defined to check the integration of the coordination layer and the slice manager.

Based on evaluations, partner's expertise, python support and reporting features and 5GENESIS requirements, the most appropriate tools for the objectives of 5GENESIS are determined to be Robot and pyTest. These two tools are considered to be the most suitable candidates for Rel B onwards.

2.3. Git-based Approach for Component Integration

In this section we propose a Git based methodology to address the integration of the WP3 components i.e., releases, hotfixes and feature enhancements. The proposed approach described here addresses three fundamental questions:

1. Which version (commit) of *Component X* developed by WP3 should be integrated and deployed by WP5 in the various 5GENESIS Platforms?
2. How WP3 developers can provide the Release Candidates (RC) of their individual components for Platform integration? and,
3. How WP5 integrators can provide feedback to WP3 developers to develop hotfixes and provide 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.

2.3.1. 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.

In the context of 5GENESIS, we propose to use Git based master/develop/release work flow, as illustrated in Figure 2 and Figure 3. The *master* branch is a protected branch that is production ready, while the *develop* branch is where the actual component development and commits happens. Thus, as illustrated in the figures, the *develop* branch initially branches out of the *master* branch, and when ready for release, is merged back into the *master* branch.

Once a set of features / requirements agreed upon at the start of the release cycle are realized, a release candidate (RC) is forked from the *develop* branch. All bug fixes discovered henceforth are committed back on the forked *RC* branch. When the RC is stable for release, the *RC* branch is merged back to the *develop* and *master* branches. Furthermore, a protected and read-only tag of the *master* branch with the correct release version is created.

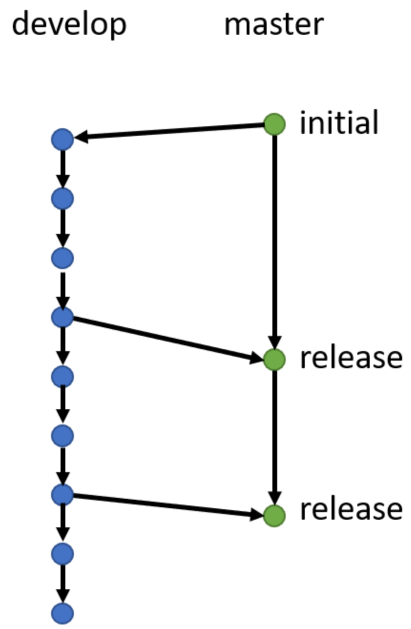


Figure 2. Git based master/develop/release work flow

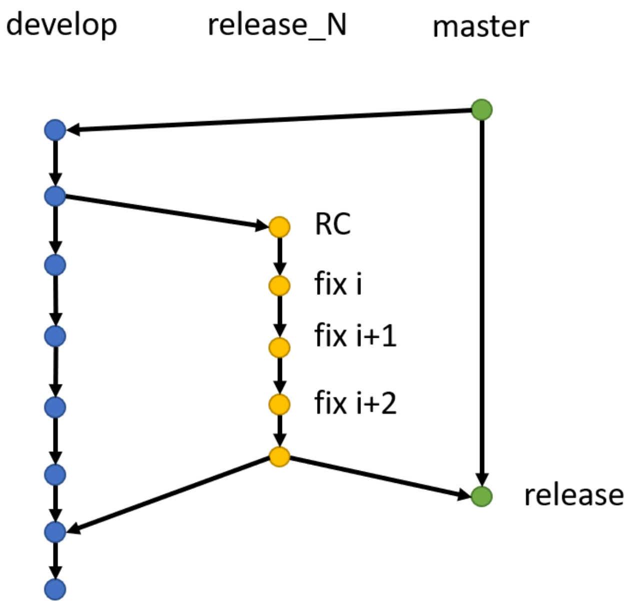


Figure 3. Git based master/develop/release workflow showcasing bug fixes

In the context of 5GENESIS, the git-based master/branch/release workflow is mapped to the work package activities as depicted in Figure 4.

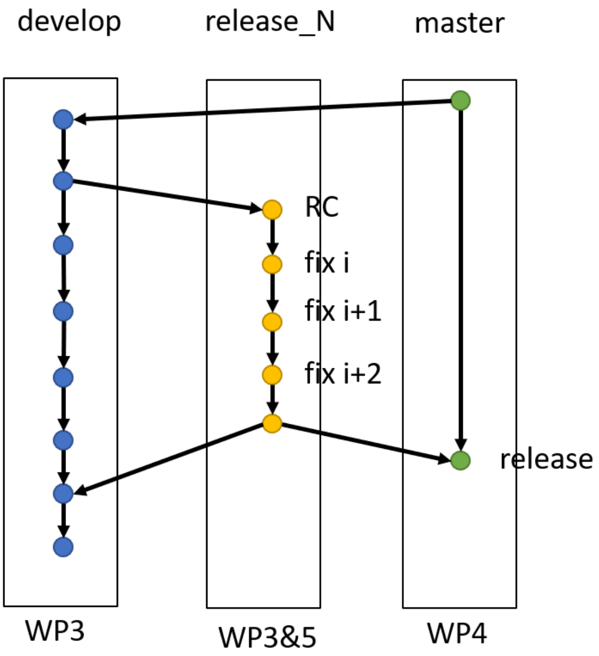


Figure 4. Git based master/develop/release workflow mapped to WPs.

WP3 works on the development of the individual components and produces a RC for WP5. WP5 tests the RC and provides feedback to WP3 to provide bug fixes. Once no more bugs are discovered on the RC, WP3/5 merges the RC with bug fixes back to develop and master branches. WP5 produces a tag from the master branch with correct release name (Release_*N*). WP4 deploys the tested tag Release_*N* in their Platform.

2.3.2. 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

In the context of 5GENESIS, semantic versioning of the components is proposed. The approach consists of three numbers separated by dots in the format:

MAJOR.MINOR.PATCH

The versioning is largely intended for the (dependency) management of the component APIs. Thus, for instance, PATCH part of the version would be incremented when bug fixes with no implications on the APIs offered by the component are made. The MINOR part of the version number is incremented when API additions and changes are made with backward compatibility. When drastic API changes are made with no backwards compatibility, the MAJOR part of the version number is incremented.

Thus, at the start, 0.1.0 is assigned to the initial development release of a component and the minor version incremented for each subsequent release. When the component is ready to be deployed in the production environment (individual Platforms) during the first release cycle, the version number is incremented to 1.0.0. During the next development release, the minor version is incremented to 1.1.0. Bug fixes on this release would increment the PATCH, i.e., 1.1.1 to 1.1.n

2.3.3. 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[1]

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, e.g., by orchestrating the components and services via Kubernetes. WP4, responsible for the appropriate instantiation of validated 5GENESIS releases, then receives a layer (e.g., 5GENESIS Coordination Layer) as a Service, i.e., ready to use k8s deployment.

[1] Ansible - https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html

3. 5GENESIS FACILITY RELEASE A

In this section, a brief overview of the 5GENESIS Facility Layer Release A is provided. Firstly, the features of the Coordination layer and Slice Manager are briefly described, followed by an overview of its different functional components. Finally, the south-bound interfaces for the interconnection of the Coordination Layer and the Slice Manager with the underlying 5GENESIS architecture are defined.

3.1. 5GENESIS FACILITY Release A 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 A 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.
- Communication between the Portal and the Experiment Lifecycle Manager (ELCM) via the initial version of the OPEN API described in D3.7 [5].
- The long-term storage of the results generated by the experiments.
- Automated communication with the Slice Manager and the lower layers for the configuration of probes and instruments required for the execution of experiments.

Based on the 5GENESIS architecture, the experimenter/vertical has two options for performing an experiment:

- Through the 5GENESIS GUI/Portal (Available demo at 5GENESIS booth), where the experiment descriptor is automatically generated and sent to the dispatcher (IDEAL FOR E2E KPI ASSESSMENT)
- Directly via the 5GENESIS open API, allowing the experimenter to use the facility with its own scripts (IDEAL FOR VALIDATION OF A NEW COMPONENT OR SERVICE).

The Dispatcher obtains the experiment descriptor from the Portal, initiates the validation of the descriptor and sends the experimentation plan to the scheduler that 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 The Dispatcher is also able to send part of an experiment descriptor to a Dispatcher on another 5GENESIS Platform for distributed execution of experiments.

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, Slice Manger and analytics.

The Coordination layer is defined in more detail on Section 3.2 of Deliverable D2.2 [6].The Slice manager is detailed in Deliverable 3.3 [9]. Note that the aforementioned SW components have been provided by WP3 deliverables.

- 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 generated by the 5GENESIS Portal in the form of the experiment descriptor and is able to perform the execution of multiple experiments in parallel. By sending requests to 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 Deliverable D3.15 [7].
- Monitoring and Analytics
  The analytics module performs the analysis of the raw data generated during an experiment execution, performing the calculation of the key performance indicators of the experiment. During Release A, several probes have been developed and integrated, as well as scripts for processing the results provided by these probes. More information about these probes is available in Deliverable D3.5 [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. During Release A, the Open API is embedded as part of the Portal and the ELCM, which makes the communication between these two components direct. More information about the Portal can be seen in Section 4 of Deliverable D3.7 [5].
- Slice Manager
  In 5GENESIS the slice view will be provided and controlled from a central software component, i.e., the Slice Manager, a standalone component that 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 A is described in D3.3 [9]. Following the "Study on management and orchestration of network slicing for next generation network"[10], 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 5GENESIS Slice Manager is responsible for the communication with the underlying components of each Platform, as depicted in Figure 5, in order to provide the required resources across the different domains of the testbed and instantiate the network services that constitute an end-to-end communication service.

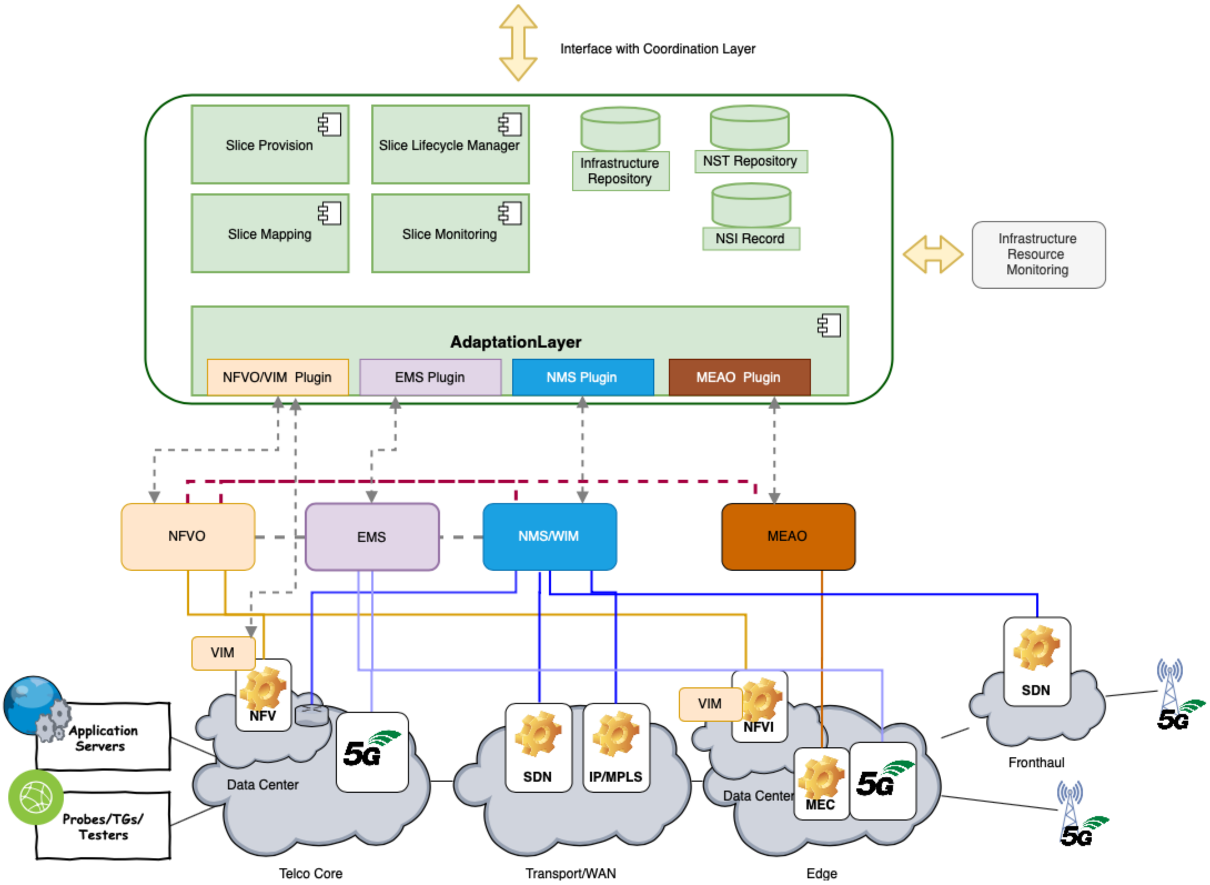


Figure 5. Slice Manager Architecture

The 5GENESIS Slice Manager is based on a highly modular architecture, built as a collection of microservices, each of which is running on a docker container. The key advantages of this architectural approach are that it offers simplicity in building and maintaining applications, flexibility and scalability, while the containerized approach makes the applications independent of the underlying system.

The 5GENESIS Slice Manager provides a set of North-Bound REST APIs, that follow the Open APIs 3 specification [11], together with a built-in Swagger-UI tool, which is used for documenting, testing and consuming the API endpoints. These APIs can be consumed by the 5GENESIS Experiment Life Cycle Manager (ELCM) or by the Slice Manager Administrator in order to trigger some of the Slice Manager functionalities, such as performing create, read, update and delete (CRUD) operations on NSIs, adding South Bound components of the underlying Platform or retrieving information about an instantiated 5G Network Slice.

3.2. Interfaces

This sub-section describes the south-bound interfaces for the interconnection of the Coordination Layer and Slice Manager with the underlying architecture. More specifically, these are the following:

- SM-NMS/NFVO:
  The Slice Manager communicates with the underlying MANO components in order to perform CRUD operations on sub-slice instances or retrieve information about the underlying Platform infrastructure. More specifically, the components with which Slice Manager interacts are the Virtual Infrastructure Managers (VIMs), Network Function Virtualization Orchestrators (NFVOs), WAN Infrastructure Manager (WIM), Element Management System (EMS) and Monitoring Framework. An Adaptation Layer is introduced as part of the Slice Manager architecture, as depicted in Figure 5, in order to provide a level of abstraction regarding the underlaying layer technology, making it feasible for the Slice Manager to operate over any type of the aforementioned Management and Orchestration (MANO) layer components, without any modifications to its core functionality, as long as the proper plugin modules have been loaded. The plugins translate the Slice Manager messages to type-specific messages for the South Bound components.
  Table 1 presents the operations between the Slice Manager and the MANO layer components of the 5GENESIS facilities.

Table 1. Operations between SM and MANO components

Component	Operation	Phase
VIM	Create a new Tenant	Slice Creation – Resource Provisioning
	Delete a Tenant	Slice Termination
NFVO	Read Network Service Descriptors (NSDs) and Virtual Functions Descriptors (VNFDs)	Slice Creation – Placement
	Add a new VIM account (VIM Tenant)	Slice Creation – Resource Provisioning
	Instantiate a new NS	Slice Creation – Activation
	Read NS Records (NSRs) and VNF Records (VNFRs)	Slice Creation – Activation
	Delete an instantiated NS	Slice Termination
	Delete a VIM account (VIM Tenant)	Slice Termination
WIM	Create the transport network graph	Slice Creation – Resource Provisioning
	Activate the network traffic steering for a network slice	Slice Creation – Activation
	Delete the transport network graph	Slice Termination
EMS	Reserve RAN components	Slice Creation – Resource Provisioning
	Configure and start RAN services	Slice Creation – Activation
	Terminate RAN services	Slice Termination
	Release RAN components	Slice Termination
MON	Get information about Platform available resources	Slice Creation – Placement

- Validator to MANO:

  As part of the Open APIs' features, 5GENESIS offers a validator interface to validate VNFDs and NSDs prior to the onboarding in the Platform. This module, located inside the Dispatcher, performs an enhanced syntax validation over the packages, more thorough than the one provided by the NFVO itself, which is too relaxed for the Project needs, allowing descriptors that do not match the accepted information model, hindering the parsing of the descriptors in later phases of their lifecycle. The Validator exposes several functionalities over this interface:
  - VNFD validation
  - VNFD validation + onboarding in the NFVO
  - NSD validation
  - NSD validation + onboarding in the NFVO

  It is possible to validate a descriptor without actually having to onboard it. This is a particularly useful feature for an external user, who can test the validity of the descriptor during the creation process without affecting the rest of the system.

- Validator to ELCM:

  Another feature of the Validator is to validate not only NFV descriptors but also 5GENESIS Experiment descriptors. This interface offers also similar functionalities:
  - Experiment descriptor validation
  - Experiment descriptor validation + onboarding in the ELCM

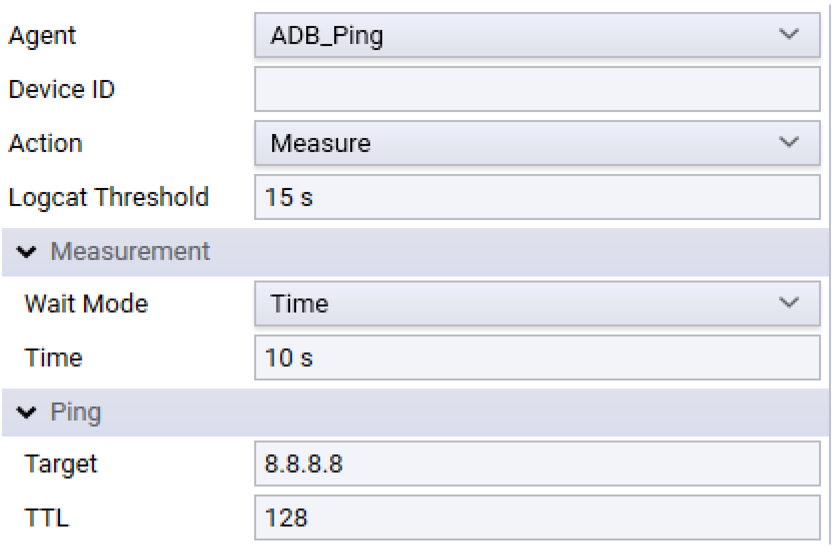
3.2.1. Instrumentation

- UE-side Configuration:

  Several options are available for the management and configuration of UEs and the different instruments available in the Platform: The ELCM includes functionality for running tasks through the command line, which give Platform administrators the possibility of running any application or script required for the configuration of a device as part of an experiment execution.

  Additionally, two generic TAP Plugins have been developed: The SSH and ADB plugins. The SSH plugin can be used for controlling any remote machine through this protocol and is also able to send and retrieve files by using SCP. The ADB (Android Debug Bridge) plugin includes functionality for transferring files to and from an Android device, managing Logcat (the Android logging tool) and execute commands in a generic way. The ADB Plugin provides the basic functionality used by a second plugin (ADB Agents) that is able to control several Android probes, such as the resource monitoring agent and the Ping and iPerf probes.

  Android probes, such as the resource monitoring agent and the Ping and iPerf probes. The sequence of commands sent to the device is similar in all the available Agents, with changes in the intent's name and additional parameters. Below is a detailed description of the commands used while controlling the Ping agent, all the actions are performed automatically by the TAP Plugin using the settings specified by the user. Figure 6 shows the available ping settings.



Agent	ADB_Ping
Device ID	
Action	Measure
Logcat Threshold	15 s
Measurement	
Wait Mode	Time
Time	10 s
Ping	
Target	8.8.8.8
TTL	128

Figure 6: ADB Agent settings

1- First, the plugin sends instructions to the device so that all messages generated by the Ping agent are sent to a file in the device. This file will later be retrieved in order to obtain all the measurements generated:

```
adb.exe logcat -b main -f sdcard/adb_ping_agent.log -v threadtime -r 16384 -n
8 ping.Report:I *:S
```

2- The plugin makes use of the intents exposed by the agent in order to initiate the measurement. At this point also the configuration parameters set on the step are sent to the agent:

```
adb.exe shell am startservice -n com.uma.ping/.PingService -a
com.uma.ping.START -e com.uma.ping.PARAMETERS
"target=8.8.8.8,ttl=128"
```

3- After waiting for a specific amount of time (10 seconds in this case), the plugins send the order to stop the measurement:

```
adb.exe shell am startservice -n com.uma.ping/.PingService -a
com.uma.ping.STOP
```

4- The plugin retrieves the log file created in step 1 and process the contents internally in order to retrieve the generated results. Figure 7 shows an example of the raw results generated by the agent.

```
adb.exe pull "sdcard/adb_ping_agent.log" "[..]\adb_ping_agent.log"
```

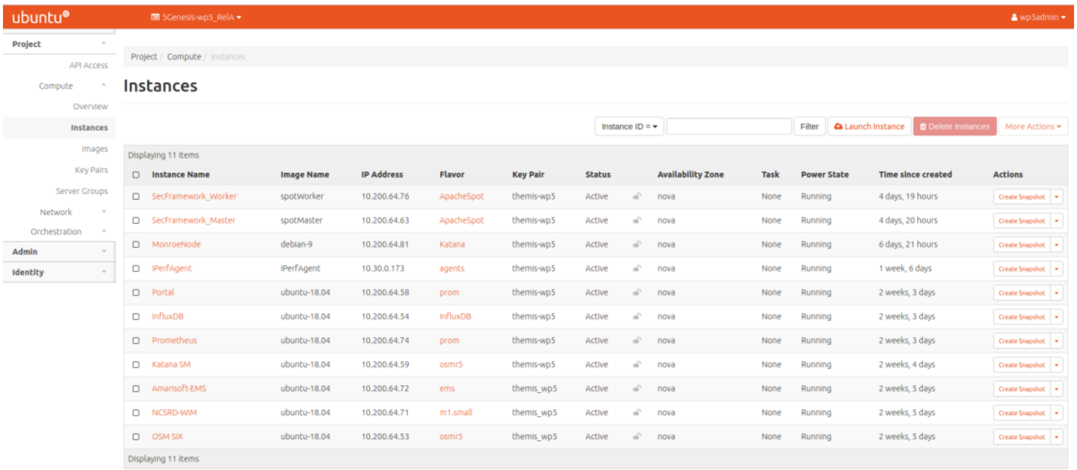
```
 12-12 10:30:24.141 10317 10364 I ping.Report:
<<<Timestamp:1576143024152;Time:0;Delay:32.7>>>
 12-12 10:30:25.161 10317 10364 I ping.Report:
<<<Timestamp:1576143025164;Time:1;Delay:29.8>>>
 12-12 10:30:26.101 10317 10364 I ping.Report:
<<<Timestamp:1576143026108;Time:2;Delay:26.1>>>
 12-12 10:30:27.141 10317 10364 I ping.Report:
<<<Timestamp:1576143027146;Time:3;Delay:28.3>>>
 12-12 10:30:28.171 10317 10364 I ping.Report:
<<<Timestamp:1576143028178;Time:4;Delay:28.2>>>
 12-12 10:30:29.191 10317 10364 I ping.Report:
<<<Timestamp:1576143029195;Time:5;Delay:25.3>>>
 12-12 10:30:30.141 10317 10364 I ping.Report:
<<<Timestamp:1576143030142;Time:6;Delay:25.5>>>
```

```
 12-12 10:30:31.161 10317 10364 I ping.Report:
<<<Timestamp:1576143031166;Time:7;Delay:25.3>>>
 12-12 10:30:32.191 10317 10364 I ping.Report:
<<<Timestamp:1576143032195;Time:8;Delay:27.7>>>
 12-12 10:30:33.141 10317 10364 I ping.Report:
<<<Timestamp:1576143033142;Time:9;Delay:25.6>>>
```

Figure 7: ADB Ping agent logcat output

4. Dedicated Integration Environment

A dedicated integration and testing environment is created on the Athens Platform, used for installation, testing and integrating the 'Release A' of the WP3 components, which will be part of the 5GENESIS 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. Malaga platform for example, has also created such testing environment. 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[2], where all the windows-based components are hosted, as depicted in figures below. **Further details can be found in appendix 1.**



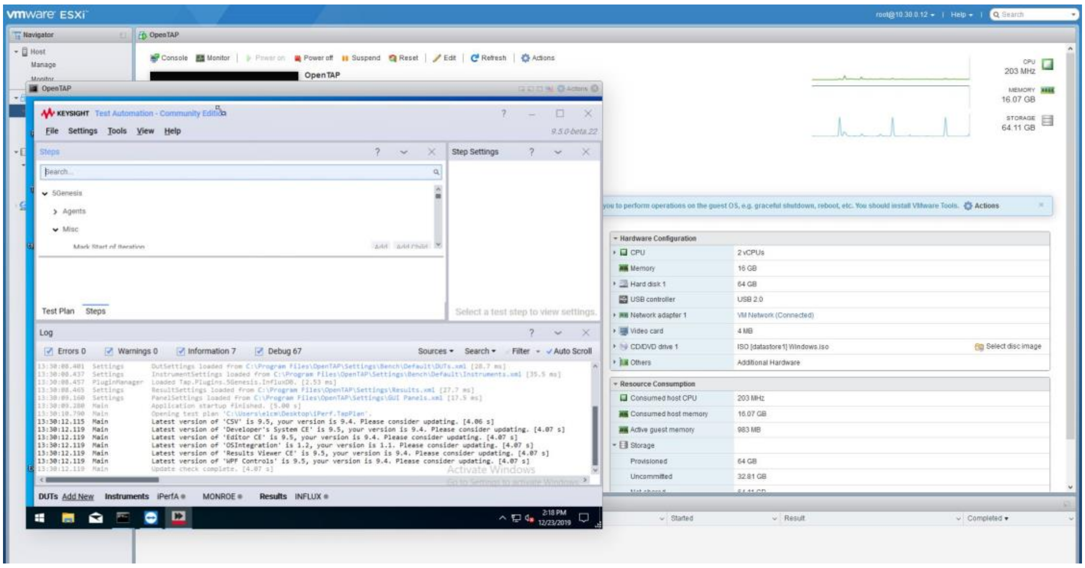
Instances

Displaying 11 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	SecFramework_Worker	spotWorker	10.200.64.76	ApacheSpot	themis-wp5	Active	nova	None	Running	4 days, 19 hours	Create Snapshot
□	SecFramework_Master	spotMaster	10.200.64.63	ApacheSpot	themis-wp5	Active	nova	None	Running	4 days, 20 hours	Create Snapshot
□	MonroeNode	debian-9	10.200.64.81	Katana	themis-wp5	Active	nova	None	Running	6 days, 21 hours	Create Snapshot
□	iPerfAgent	iPerfAgent	10.30.0.173	agents	themis-wp5	Active	nova	None	Running	1 week, 6 days	Create Snapshot
□	Portal	ubuntu-18.04	10.200.64.58	prom	themis-wp5	Active	nova	None	Running	2 weeks, 3 days	Create Snapshot
□	InfluxDB	ubuntu-18.04	10.200.64.54	InfluxDB	themis-wp5	Active	nova	None	Running	2 weeks, 3 days	Create Snapshot
□	Prometheus	ubuntu-18.04	10.200.64.74	prom	themis-wp5	Active	nova	None	Running	2 weeks, 3 days	Create Snapshot
□	Katana SM	ubuntu-18.04	10.200.64.59	osmr5	themis-wp5	Active	nova	None	Running	2 weeks, 4 days	Create Snapshot
□	Amarisoft-EMS	ubuntu-18.04	10.200.64.72	ems	themis_wp5	Active	nova	None	Running	2 weeks, 5 days	Create Snapshot
□	NCSRD-WIM	ubuntu-18.04	10.200.64.71	m1.small	themis_wp5	Active	nova	None	Running	2 weeks, 5 days	Create Snapshot
□	OSM SIX	ubuntu-18.04	10.200.64.53	osmr5	themis_wp5	Active	nova	None	Running	2 weeks, 5 days	Create Snapshot

Displaying 11 items

Figure 8. Openstack Integration Environment



☑ Errors 0 ☑ Warnings 0 ☑ Information 7 ☑ Debug 67

```
13:30:08.401  Settings       DutSettings loaded from C:\Program Files\OpenTAP\Settings\Bench\Default\DUTs.xml [20.7 ms]
13:30:08.437  Settings       InstrumentSettings loaded from C:\Program Files\OpenTAP\Settings\Bench\Default\Instruments.xml [35.5 ms]
13:30:08.457  PluginManager  Loaded Tap.Plugins.5Genesis.InfluxDB. [2.53 ms]
13:30:08.465  Settings       ResultSettings loaded from C:\Program Files\OpenTAP\Settings\Results.xml [27.7 ms]
13:30:09.160  Settings       PanelSettings loaded from C:\Program Files\OpenTAP\Settings\GUI Panels.xml [17.5 ms]
13:30:09.280  Main           Application startup finished. [5.00 s]
13:30:10.790  Main           Opening test plan 'C:\Users\elcm\Desktop\iPerf.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 'WPF 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]
```

you to perform operations on the guest OS, e.g. graceful shutdown, reboot, etc. You should install VMware Tools. Actions

Hardware Configuration	
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 Select disc image
Others	Additional Hardware

Resource Consumption	
Consumed host CPU	203 MHz
Consumed host memory	16.07 GB
Active guest memory	983 MB
Storage	
Provisioned	64 GB
Uncommitted	32.81 GB
Not shared	64.11 GB

Figure 9. VMWare ESXI Integration Environment

[2] 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 [12], SONATA-NFV [13], etc.) and also from the work of ETSI NFV [14], 5GENESIS defines a template for the definition of the integration tests that need to be executed in order to validate component integration. Table 2 depicts the template used for the definition of integration tests.

Table 2. Test Case Template

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

The integration tests that are developed for Release A are summarized in Table 3 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: int-test-*xx*-*yy* (from Integration Test), 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, int-test-02-01 specifies the minimal functionality test that affects the ELCM, while in the future we may specify a new int-test-02-02 that covers some extra functionality added in the next phases of the development.

Table 3. 5GENESIS Release A integration tests

Test case id	Test case name	Test case description	Involved components
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	• Coordinator • Portal
int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	• Coordinator • Portal

int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	• Coordinator • Portal • ELCM
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	• Coordinator • Portal • ELCM • OpenTAP
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	• Slice Mngr • NFVO • VIM
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	• Coordination layer • Slice Mngr • NFVO, VIM, WIM

More specifically, the defined test cases are the following:

Table 4. int-test-01-01: Portal Login

Test Case Name	Portal Login	Test Case id	int-test-01-01
Test Purpose	Verify that the Portal is running, and the internal database is correctly configured		
Configuration	Portal hosting server assigned IP and accessible from the external networks		
Test Tool	Web browser		
Metric	n/a		
References	n/a		
Applicability	Portal		
Pre-test conditions		Portal has been deployed and is listening for connections on a known address. No users have connected to the Portal before.	
Test sequence	Step	Connect to the Portal address with a web browser	The Portal's Login page should be visible
	Step	Click the "Register" button on the top of the page	The Portal's Register page should be visible
	Step	Fill the required information (note the username and password used). Click the "Register" button at the bottom of the page.	The Portal's Login page should be visible, but this time a "You have been registered" message

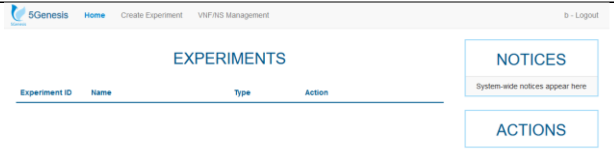
			appears near the top of the page
	Step	Fill the username and password fields with the values used in the previous step. Click the "Sign In" button.	The user's dashboard is visible.
Test Verdict	For newly created users the dashboard should contain an empty table of experiments, three tabs on the top and a logout button. The ACTIONS list should be empty. The NOTICES box may or may not appear.		PASS
Additional Resources	 5Genesis Home Create Experiment VNF/NS Management b - Logout EXPERIMENTS Experiment ID Name Type Action NOTICES System-wide notices appear here ACTIONS		

Table 5. int-test-02-01: ELCM

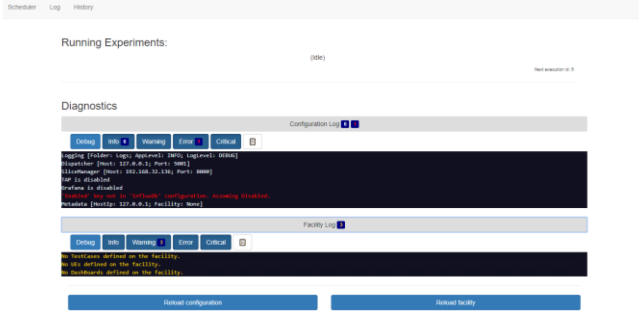
Test Case Name	ELCM	Test Case id	int-test-02-01
Test Purpose	Verify that the ELCM is running		
Configuration	ELCM hosting server assigned with IP and OpenTAP access configured		
Test Tool	Web browser		
Metric	n/a		
References	n/a		
Applicability	ELCM		
Pre-test conditions	The ELCM has been deployed and is listening at a known address. No additional configuration has been performed		
Test sequence	Step	Connect to the ELCM address with a web browser	The ELCM dashboard page should be visible
	Step	Click on the "Configuration Log" and "Facility Log"	The logs should be visible
Test Verdict	Compare the contents of the logs with the reference image (in Additional Resources. The error in the reference image is normal and easily solved). If no additional errors appear the ELCM has been correctly deployed		PASS
Additional Resources	 Scheduler Log History Running Experiments: (Idle) Next execution id: 5 Diagnostics Configuration Log 6 1 Debug Info 6 Warning Error 1 Critical Facility Log 3 Debug Info Warning 3 Error Critical Reload configuration Reload facility Logging [Folder: Logs; AppLevel: INFO; LogLevel: DEBUG] Dispatcher [Host: 127.0.0.1; Port: 5001] SliceManager [Host: 192.168.32.136; Port: 8000] TAP is disabled Grafana is disabled 'Enabled' key not in 'InfluxDb' configuration. Assuming Disabled. Metadata [HostIp: 127.0.0.1; Facility: None] No TestCases defined on the facility. No UEs defined on the facility. No DashBoards defined on the facility.		

Table 6. int-test-03-01: Portal-ELCM Communication

Test Case Name	Portal-ELCM Communication	Test Case id	int-test-03-01
Test Purpose	Verify that the Portal and the ELCM can communicate properly		
Configuration	n/a		
Test Tool	Web browser, Text editor		
Metric	n/a		
References	n/a		
Applicability	Portal, ELCM		
Pre-test conditions		Test-01-01 and Test-02-01 completed successfully, no additional configuration changes to the Portal or ELCM	
Test sequence	Step	Using a text editor, edit the config.yml file in the Portal deployment folder	
	Step	Modify the Host and Port values in the Dispatcher section so that they refer to the IP and port where the ELCM is listening	
	Step	Include a new "TEST" value in the Test Cases list (in config.yml). Save the file.	
	Step	Restart the Portal	
	Step	Connect to the Portal address with a web browser. If necessary, log in.	The user's dashboard should be visible
	Step	Click on the "Create Experiment" button	The "CREATE EXPERIMENT" page should appear. "TEST" can be selected under the "Test Cases" section
	Step	Give a name to the experiment and select TEST in the Test Cases list. Leave all other values as default. Press Add Experiment	The user's dashboard is visible, but an entry for the newly created experiment is on the table
	Step	Using a text editor, edit the config.yml file in the ELCM deployment folder	
	Step	Modify the Host and Port values in the Dispatcher section so that they refer to the IP and port where the Portal is listening. Save the file.	
	Step	Copy the 'test.yml' (in Additional Resources) file to the Test Cases subfolder of the ELCM deployment folder.	

	Step	Connect to the ELCM address with a web browser. Click the "Reload Config" and "Reload Facility" buttons	
	Step	Expand the "Configuration Log" and "Facility Log". Ensure that no unexpected errors appear.	The "1 Test Cases defined on the Facility: TEST." message appears on the Facility Log
	Step	Connect to the Portal address with a web browser. If necessary, log in.	The user's dashboard should be visible
	Step	Click on the "Run Experiment" button	A message similar to "Success: True - Execution Id: > - Message: Created execution > for experiment > (Id:>, User:<?>)" is visible near the top of the page
	Step	Click on the "Executions" button	A list of the experiment executions appear.
	Step	Wait until the experiment execution finishes, then click the "Execution Logs" button	The logs generated during the experiment execution are visible
Test Verdict	Check the contents of the Run Log for a message with the following content: "This is a TEST message". If it's present, the Portal and ELCM communicate and can run experiments correctly.		PASS
Additional Resources	'test.yml' (found in the project gitlab)		

Table 7. int-test-04-01: ELCM-OpenTAP- integration

Test Case Name	ELCM-OpenTAP integration	Test Case id	int-test-04-01
Test Purpose	Verify that OpenTAP is correctly configured and can be used by the ELCM		
Configuration	n/a		
Test Tool	Web browser, Text editor		
Metric	n/a		
References	n/a		
Applicability	ELCM, Portal, OpenTAP		
Pre-test conditions	Test-01-01 to Test-03-01 completed successfully, no additional modifications have been performed. OpenTAP installed in the same machine as the ELCM		

Test sequence	Step	Using a text editor, edit the config.yml file in the ELCM directory.	
	Step	Modify the "Tap" section. Set "Enabled", "OpenTap" and "EnsureClosed" to True. Modify the "Exe", "Folder" and "Results" values if necessary. Save the file.	
	Step	Save the "test.TapPlan" file (Additional Resources) to a known folder in the ELCM/OpenTAP machine	
	Step	Overwrite the contents of "test.yml" (the file saved during Test-03-01) with the new version in Additional Resources	
	Step	Using a text editor, edit "test.yml". Modify the placeholder with the full path where the "test.TapPlan" file has been saved. Save the file.	
	Step	Connect to the ELCM address with a web browser. Click the "Reload Config" and "Reload Facility" buttons	
	Step	Expand the "Configuration Log" and "Facility Log". Ensure that no unexpected errors appear.	The "1 Test Cases defined on the Facility: TEST." message appears on the Facility Log
	Step	Connect to the Portal address with a web browser. If necessary, log in or click on the Home button at the top of the page.	The user's dashboard should be visible
	Step	Click on the "Run Experiment" button	A message similar to "Success: True - Execution Id: > - Message: Created execution > for experiment > (Id:>, User:<?>)" is visible near the top of the page
	Step	Click on the "Executions" button	A list of the experiment executions appear.
	Step	Wait until the execution on top of the list finishes, then click the "Execution Logs" button	The logs generated

			during the experiment execution are visible
Test Verdict	Check the contents of the Run Log for a message with the following content: "This is a message -> *n*", where *n* can be any number. If it's present, the ELCM can make use of OpenTAP, sending the required external parameters.		PASS
Additional Resources	'test.TapPlan' 'test.yml' (found in the project gitlab)		

Table 8. int-test-05-1: Slice Creation

Test Case Name	Slice Creation	Test Case id	int-test-05-01
Test Purpose	Verify that Slice Manager is correctly installed and configured with the South Bound Components of the Platform		
Configuration	Server assigned IP and accessible by the other Northbound components		
Test Tool	Text Editor, curl / Swagger-UI / SM CLI tool		
Metric	Functional test		
References	n/a		
Applicability	Slice Manager, NFVO, VIM		
Pre-test conditions	Slice Manager is correctly installed following the instructions found in the project gitlab- NFVO and VIM are installed with known URLs and credentials - NSDs to be used are onboarded to the NFVO - VM images to be used are onboarded to the VIM		
Test sequence	Step	Using a text editor, create the configuration files (in JSON or YAML format) for the VIMs and NFVO to be added in the Slice Manager, based on the example files found in the project gitlab	
	Step	Add the configuration files to the Slice Manager using (i) the SM CLI tool `katana vim add -f vim_conf.json` and `katana nfvo add -f nfvo_conf.json`, (ii) the REST APIs `curl -X POST -d @vim_conf.json http://katanaSM:800/api/vim` and `curl -X POST -d @nfvo_conf.json http://katanaSM:800/api/nfvo` or (iii) the Swagger-UI tool	Slice Manager should return the UUID of each component
	Step	Using a text editor, create the Network Slice Template (in JSON or YAML format) that describes the slice to be added, based on the example files (found in the project gitlab)	
	Step	Add the NST to the Slice Manager using (i) the SM CLI tool `katana slice`	This step will trigger the slice creation. Slice Manager

		`add -f nst.json`, (ii) the REST APIs `curl -X POST -d @nst.json http://katanaSM:800/api/slice` or (iii) the Swagger-UI tool	should return the UUID of the new slice.
	Step	Check the status of the slice using (i) the SM CLI tool `katana slice inspect <slice_uuid>`, (ii) the REST APIs `curl -X GET http://katanaSM:800/api/slice/<slice_uuid>` or (iii) the Swagger-UI tool	Slice Manager will return information regarding the new slice, including the status (Init/Placement/Provisioning/ Activation/Running)
	Step	Check the NFVO and VIMs involved in the slice for the new Tenant created for the slice and for the instantiated NSs and VNFs in that Tenant space	
	Step	Check the slice deployment time using (i) the SM CLI tool `katana slice deployment_time <slice_uuid>`, (ii) the REST APIs `curl -X GET http://katanaSM:800/api/slice/<slice_uuid>/time` or (iii) the Swagger-UI tool	Slice Manager will return a json file with information about the deployment time of the slice
	Step	Terminate the created slice using (i) the SM CLI tool `katana slice rm <slice_uuid>`, (ii) the REST APIs `curl -X DELETE http://katanaSM:800/api/slice/<slice_uuid>` or (iii) the Swagger-UI tool	Slice Manager will return the status "Terminating"
	Step	Check the NFVO and VIMs involved in the slice for the successful termination of NSs and VNFs and the deletion of the new Tenant	
Test Verdict	If the Slice Manager returned the expected results with no error or warning messages, then a new slice was created and then terminated over the configured VIMs		
Additional Resources	'vim.json', 'nfvo.json', 'nst.json' (found in the project gitlab)		

Table 9. int-test-06-01: End-to-end experiment lifecycle test

Test Case Name	ELCM-Slice Manager Communication	Test Case id	int-test-06-01
Test Purpose	Verify that the ELCM can request the creation and decommission of slices and retrieve results from the Slice Manager. Verify that InfluxDB is correctly configured		
Configuration	n/a		
Test Tool	Web browser, Text editor, SSH Client		

Metric	n/a		
References	n/a		
Applicabilit y	ELCM, Portal, Slice Manager, InfluxDB		
Pre-test conditions	int-test-01-01 to int-test-05-01 completed successfully, no additional modifications have been performed. The Slice Manager is deployed, configured and able to deploy the test NS, a NSD file for this test NS is available. InfluxDB is deployed, configured, and the machine can be reached through SSH.		
Test sequence	Step	Using a text editor, edit the config.yml file in the ELCM directory.	
	Step	Modify the "SliceManager" section. Set the Host and Port values to the ones where the Slice Manager API is listening for connections.	
	Step	Modify the "InfluxDB" section. Set the values required for connecting to the database where the results will be stored. Save the file.	
	Step	Copy the "nsDeployment.yml" file (Additional Resources) to the "Test Cases" subfolder of the ELCM deployment folder	
	Step	Connect to the ELCM address with a web browser. Click the "Reload Config" and "Reload Facility" buttons	
	Step	Expand the "Configuration Log" and "Facility Log". Ensure that no unexpected errors appear.	
	Step	Using a text editor, edit the config.yml file in the Portal directory.	
	Step	Include a new "NsDeployment" value in the Test Cases list (in config.yml). Save the file.	
	Step	Restart the Portal	
	Step	Connect to the Portal address with a web browser. If necessary, log in.	The user's dashboard should be visible
	Step	Click on the "VNF/NS Management" button on the top of the page	The (empty) VNF and NSD repositories are visible
	Step	Click the "Upload NS" button	The "UPLOAD NS" form should be visible
	Step	Fill the "Name" field with "TestNS", and "Description" with "Test NS". Click the "Browse" button and select the NSD file of the test NS. Click "Upload NS"	The VNF and NSD repositories are visible, but now "TestNS" appears

	Step	Click on the "Create Experiment" button	The "CREATE EXPERIMENT" page should appear. "NsDeployment" can be selected under the "Test Cases" section
	Step	Give a name to the experiment and select "NsDeployment" in the Test Cases list. Leave all other values as default. Press Add Experiment	The user's dashboard is visible, but an entry for the newly created experiment is on the table
	Step	Click on the "Run Experiment" button of the NsDeployment experiment	
	Step	Click on the "Executions" button of the NsDeployment experiment	A list of the experiment executions appear.
	Step	Wait until the execution finishes, then click the "Execution Logs" button	The logs generated during the experiment execution are visible
	Step	Look for unexpected error messages. If none appears, click on any "Debug" button on the logs	The DEBUG messages are now visible
	Step	On the Run Log, look for a message similar to `Payload: InfluxPayload['Slice_Creation_Time' - Tags: {'ExperimentId': '293'} - Points: [<2019-09-18 07:27:13.491412 {'Slice_Deployment_Time': 13.8065, 'Placement_Time': 0.0032, 'Provisioning_Time': 3.6709}>]]`. Numeric values will be different.	
	Step	Using an SSH client, connect to the machine hosting the InfluxDB instance.	
	Step	On the command prompt, run "influx"	Some InfluxDB messages appear, ending with "Enter an InfluxQL query"
	Step	Run "use *db*", where *db* is the name of the database that contains the ELCM results	"Using database *db*" appears
	Step	Run "show measurements"	"Slice_Creation_Time" appears in one of the returned lines
	Step	Run "select * from Slice_Creation_Time"	Some results appear

Test Verdict	If some results are visible in the last step, then the ELCM was able to request the creation and decommissioning of a slice and request the deployment times to the Slice Manager. Then, a payload with these results have been successfully generated and received by the InfluxDb instance.	PASS
Additional Resources	'nsDeployment.yml' (found in the project gitlab)	

6. Testing and Validation Results

Based on test cases defined in previous section, validation activity has been conducted by all platforms. The results summary are depicted in the following tables.

6.1. Athens Platform

Table 10. The Athens Platform verification results

Test case id	Test case name	Test case description	Result
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	Pass
int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	Pass
int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	Pass
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	Pass
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	Pass (MANO Layer Components: - OpenStack Rocky - OSM 5 & 6 - ODL WIM Amarisoft EMS)
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	Pass

The Release A of the components that are comprising the 5GENESIS Coordination Layer, i.e. Portal, ELCM, OpenTAP and Slice Manager, have been integrated in Athens platform without any issues, while the Testing and Validation process described in section 5 has been successfully completed. The results of these integration tests are presented in Table 10. Further details regarding the Coordination Layer components in Athens Platform can be found in D4.2 [15]. The 5GENESIS Coordination Layer enables the automated execution of end-to-end trials and experimentation in the Athens Platform during the Phase 2.

6.2. Berlin Platform

Table 11. The Berlin Platform verification results

Test case id	Test case name	Test case description	Result
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	Pass
int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	Pass
int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	Pass
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	Pass
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	Fail - Initial Slice Manager Connection to Openstack (ver.: Stein) was not successful, resulting in timeout error.)
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	Not Tested - Because the slice creation test was not passed.

The 5GENESIS Portal, ELCM and OpenTAP (Release A) have been successfully deployed and integrated in the Berlin platform. The experiments are working as expected and the experiment results are saved in the influx DB. Additionally, Slice Manager is installed successfully. However, the integration of slice manager with VIM and NFVO is not successful. It happens to be version problem. Since, Berlin platform uses Openstack Stein version. Slice manager does not support the stein release yet. Hence, the Test [int-test-05-01] is not passed. This leads to the blocking of end-to-end test [int-test-06-01]. Currently, the main challenge to be addressed during Phase 3 is to decide the versions of Openstack and OSM to support by the Slice Manager.

6.3. Limassol Platform

Table 12. The Limassol Platform verification results

Test case id	Test case name	Test case description	Result
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	Pass

int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	Pass
int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	Pass
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	Pass
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	Pass (slice containing only VNFs)
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	In-progress

The 5GENESIS Portal, ELCM and OpenTAP (Release A) have been successfully integrated in the Limassol platform and are operating as planned. No major issues have been identified. End-to-end testing will take place over the next couple of months (January – March 2020). One of the main challenges to be tackled during Phase 3 is the integration of the Slice Manager with the underlying management components in order to be able to orchestrate an end-to-end slice across the satellite and terrestrial segments.

6.4. Malaga Platform

Table 13. The Malaga Platform verification results

Test case id	Test case name	Test case description	Result
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	Pass
int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	Pass
int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	Pass
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	Pass
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	Pass
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	Pass

The 5GENESIS Portal, ELCM and OpenTAP (Release A) was integrated initially in the Malaga platforms and the lessons learned were used to guide the rest of the consortium through their deployment in the rest of the platforms. The main pending action for the next Release is the support for control and configuration of gNBs and core network for supporting the automation of the deployment of end to end slices.

6.5. Surrey Platform

Table 14. The Surrey Platform verification results

Test case id	Test case name	Test case description	Result
int-test-01-01 [Table 4]	Portal access and login	Tests access and authentication for experimenters	Pass
int-test-02-01 [Table 5]	ELCM	Tests the operational status of ELCM	Pass
int-test-03-01 [Table 6]	Portal-ELCM	Tests the operation of Portal and ELCM communication	Pass
int-test-04-01 [Table 7]	ELCM-OpenTAP integration	Tests the proper configuration of OpenTAP and its availability on the ELCM	Pass
int-test-05-01 [Table 8]	Slice Creation	Tests the creation of a slice	In-progress
int-test-06-01 [Table 9]	End-to-end experiment lifecycle test	End to end test of the full experimentation cycle	In-progress

The 5GENESIS Portal, ELCM, TAP (Release A of 5GENESIS facility components) have been successfully deployed and integrated in the Surrey platform and are operating as expected. No major issues have been identified, however, slice-creation testing and subsequent EtE full experimentation lifecycle testing, is still in progress, End-to-end testing will take place over the next couple of months (January – March 2020). The results of the integration tests are presented in Table 14.

7. CONCLUSIONS

This document is the first deliverable of WP5 and is used for the reporting of the integration activities performed within the context of the work package.

The integration activities based on Release A of the 5GENESIS Facility components 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. This methodology will be used for all future releases of 5GENESIS. The integration of Release A was performed in a dedicated environment in the Athens Platform.

The 5GENESIS Coordination Layer and Slice Manger were also briefly described, focusing on its features, main components, and its communication with the lower layers of the architecture, in terms of its south-bound interfaces. 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 (selected per-platform screenshots in appendix 2), 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. The results of the integration activities per Platform at the time of the deliverable submission are also reported.

Platforms need to progress on the integration of infrastructure components such as gNBs and 5G core network. The integration of these components will be reported in D5.2, as well as the new experimentation features offered by the Coordination platform and the slice manager. In D5.3 we will report the user-manuals for developing the required plugins to integrate new infrastructure components and the manuals for verticals for executing the experiments.

REFERENCES

[1] Watir – Online: http://watir.com, visited: 23.12.2019

[2] Robot – Online: http://robotframework.org , visited: 23.12.2019

[3] pyTest – Online: https://docs.pytest.org/en/latest/index.html, visited: 23.12.2019

[4] jMeter – Online: https://jmeter.apache.org, visited: 23.12.2019

[5] 5GENESIS Deliverable D3.7, "Open API, service-level functions and interfaces for verticals", 2019, https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.7_v1.0.pdf

[6] 5GENESIS Deliverable D2.2, "Initial overall Facility design and specifications", 2018, https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D2.2_v1.0.pdf

[7] 5GENESIS Deliverable D3.15, "Experiment and Lifecycle Manager", 2019, http://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.15_v1.0.pdf

[8] 5GENESIS Deliverable D3.5, "Monitoring and analytics", 2019, https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.5_v1.0.pdf

[9] 5GENESIS Deliverable D3.3, "Slice management", 2019, https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.3_v1.0.pdf

[10] 3GPP, TR 28.801, "Study on management and orchestration of network slicing for next generation network", 2018

[11] OpenAPI Specification, https://swagger.io/specification/, visited 8.1.2020

[12] 5GTANGO Project, https://www.5gtango.eu/, visited 8.1.2020

[13] SONATA-NFV Platform, https://www.sonata-nfv.eu/, visited 8.1.2020

[14] ETSI GS NFV-TST 001 Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services

[15] 5GENESIS Deliverable D4.2, "The Athens Platform (Release B)", 2020, https://bscw.fokus.fraunhofer.de/bscw/bscw.cgi/d3392161/5GENESIS_D4.2_v1.0.pdf

ANNEX 1: ATHENS PLATFORM INTEGRATION ENVIRONMENT

Two network subnets, namely 10.200.64.0/24 and 10.30.0.0/16, are used for the interconnection of the components. Figure 10 depicts the network topology and the connected instances as shown in the OpenStack dashboard. All partners involved in the integration activities have access to the integration environment using a Virtual Private Network (VPN) connection, while the access to each instance is achieved with the use of shared ssh keys.

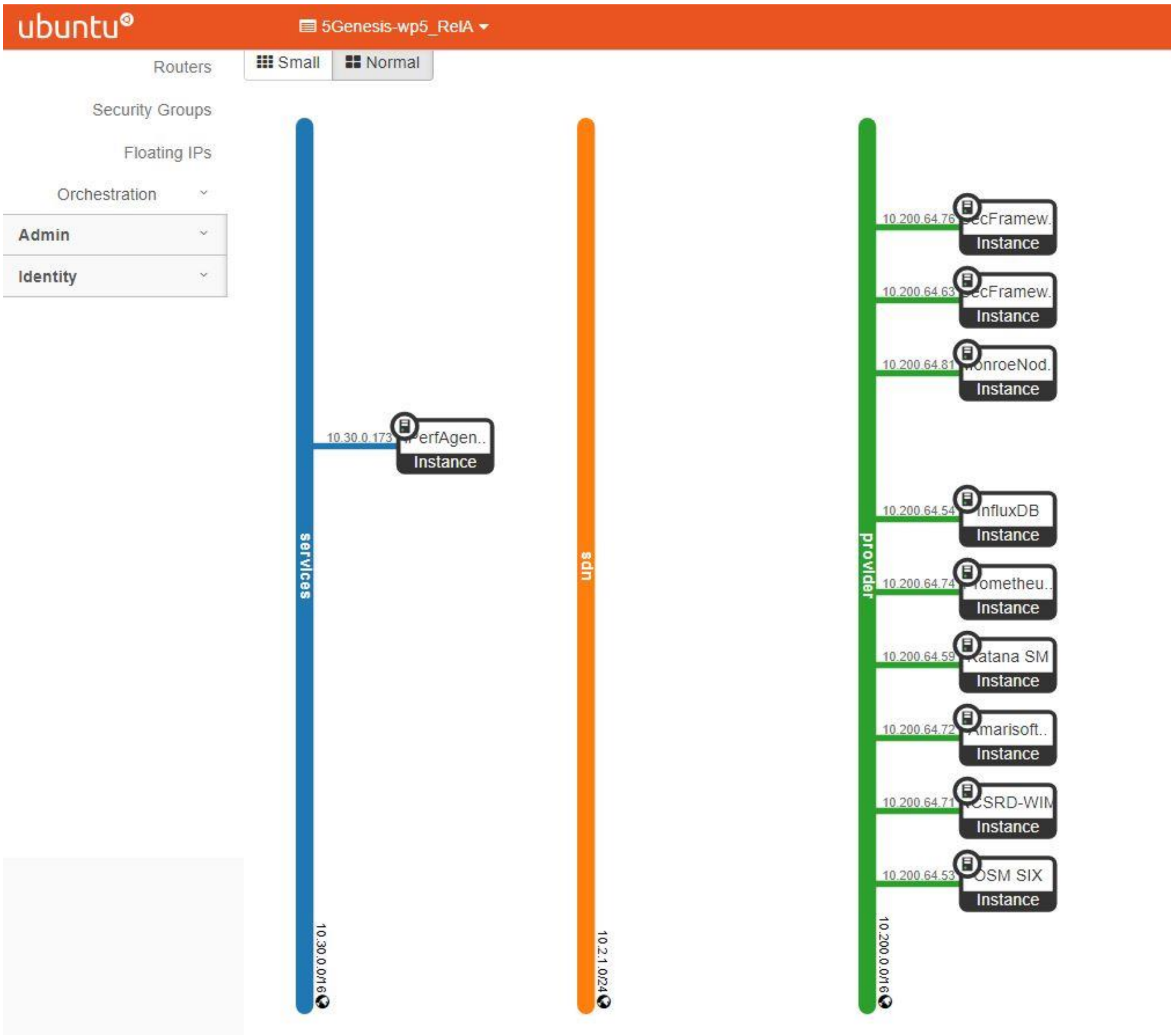


Figure 10. Openstack Networks

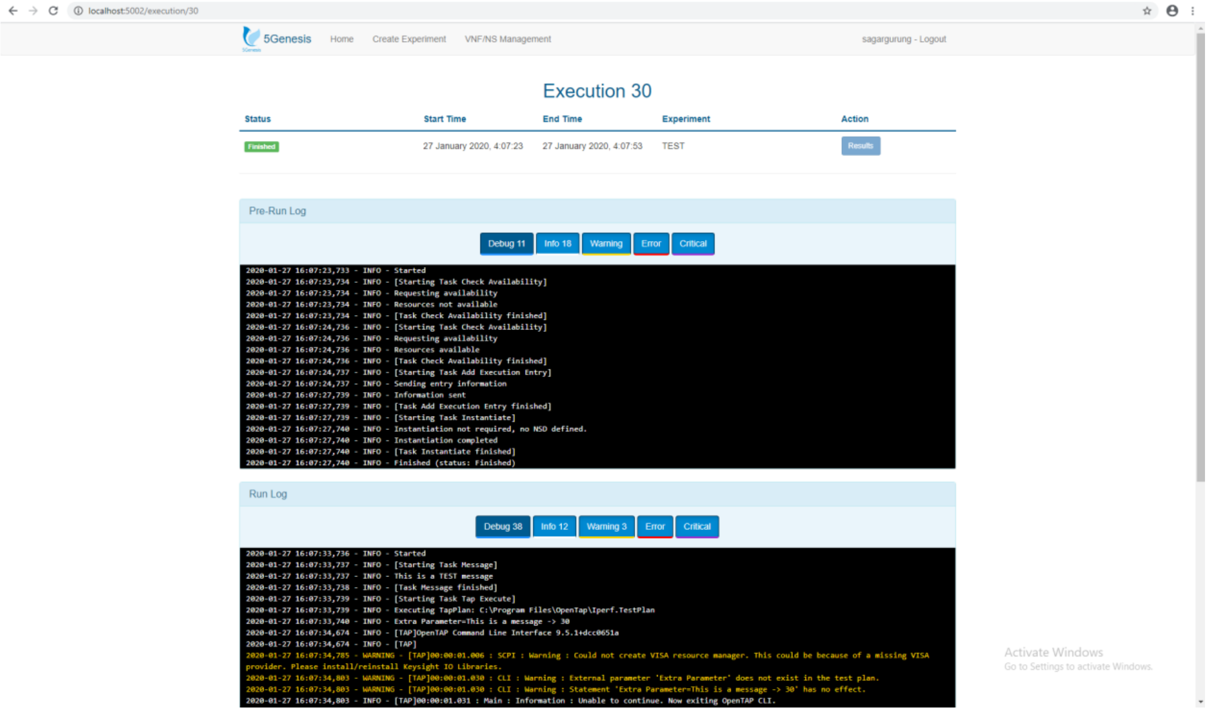
The list of resources used in the Athens Platform are listed below, while Table 15 presents in further detail information for each component:

- OpenStack: 12 instances, 28 VCPUs, 80GB RAM, 560GB Disk
- ESXI: 1 instance, 2 VCPUs, 16GB RAM, 64GB Disk

Table 15. Integration Components

Component	Host	IP Address	Resources
OSM Rel 5	Openstack	10.200.64.55	VCPUs: 2 RAM: 8GB Disk: 40GB
OSM Rel 6	Openstack	10.200.64.53	VCPUs: 2 RAM: 8GB Disk: 40GB
WIM	Openstack	10.200.64.71	VCPUs: 1 RAM: 2GB Disk: 20GB
Amarisoft-EMS	Openstack	10.200.64.72	VCPUs: 2 RAM: 2GB Disk: 80GB
Slice Manager	Openstack	10.200.64.59	VCPUs: 2 RAM: 8GB Disk: 40GB
Prometheus	Openstack	10.200.64.74	VCPUs: 2 RAM: 4GB Disk: 30GB
InfluxDB	Openstack	10.200.64.54	VCPUs: 2 RAM: 4GB Disk: 120GB
Portal – ELCM – OpenTAP	ESXI	10.30.0.250	VCPUs: 2 RAM: 16GB Disk: 64GB
Iperf Agent	Openstack	10.30.0.173	VCPUs: 1 RAM: 4GB Disk: 20GB
Monroe Probe	Openstack	10.200.64.81	VCPUs: 4 RAM: 4GB Disk: 40GB
Security Framework Master Node	Openstack	10.200.64.63	VCPUs: 4 RAM: 16GB Disk: 50GB
Security Framework Worker Node	Openstack	10.200.64.76	VCPUs: 4 RAM: 16GB Disk: 50GB

ANNEX 2: SURREY PLATFORM INTEGRATION ACTIVITIES (SCREENSHOTS)



Execution 30

Status	Start Time	End Time	Experiment	Action
Finished	27 January 2020, 4:07:23	27 January 2020, 4:07:53	TEST	Results

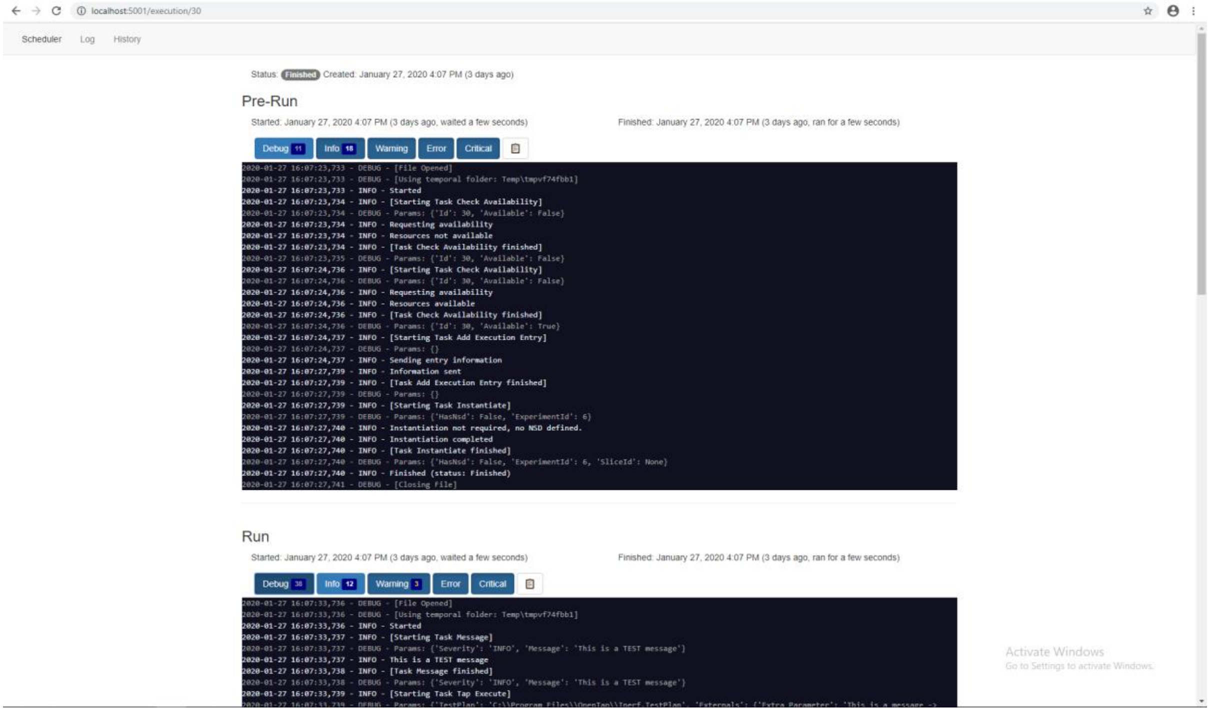
Pre-Run Log

```
2020-01-27 16:07:23,733 - INFO - Started
2020-01-27 16:07:23,734 - INFO - [Starting Task Check Availability]
2020-01-27 16:07:23,734 - INFO - Requesting availability
2020-01-27 16:07:23,734 - INFO - Resources not available
2020-01-27 16:07:23,734 - INFO - [Task Check Availability finished]
2020-01-27 16:07:24,736 - INFO - [Starting Task Check Availability]
2020-01-27 16:07:24,736 - INFO - Requesting availability
2020-01-27 16:07:24,736 - INFO - Resources available
2020-01-27 16:07:24,736 - INFO - [Task Check Availability finished]
2020-01-27 16:07:24,737 - INFO - [Starting Task Add Execution Entry]
2020-01-27 16:07:24,737 - INFO - Sending entry information
2020-01-27 16:07:27,739 - INFO - Information sent
2020-01-27 16:07:27,739 - INFO - [Task Add Execution Entry finished]
2020-01-27 16:07:27,739 - INFO - [Starting Task Instantiate]
2020-01-27 16:07:27,740 - INFO - Instantiation not required, no NSD defined.
2020-01-27 16:07:27,740 - INFO - Instantiation completed
2020-01-27 16:07:27,740 - INFO - [Task Instantiate finished]
2020-01-27 16:07:27,740 - INFO - Finished (status: Finished)
```

Run Log

```
2020-01-27 16:07:33,736 - INFO - Started
2020-01-27 16:07:33,737 - INFO - [Starting Task Message]
2020-01-27 16:07:33,737 - INFO - This is a TEST message
2020-01-27 16:07:33,738 - INFO - [Task Message finished]
2020-01-27 16:07:33,739 - INFO - [Starting Task Tap Execute]
2020-01-27 16:07:33,739 - INFO - Executing TapPlan: C:\Program Files\OpenTap\Iperf.TestPlan
2020-01-27 16:07:33,740 - INFO - Extra Parameter=This is a message -> 30
2020-01-27 16:07:34,674 - INFO - [TAP]OpenTAP Command Line Interface 9.5.1+dcc0651a
2020-01-27 16:07:34,674 - INFO - [TAP]
2020-01-27 16:07:34,785 - WARNING - [TAP]00:00:01.006 : SCPI : Warning : Could not create VISA resource manager. This could be because of a missing VISA
provider. Please install/reinstall Keysight IO Libraries.
2020-01-27 16:07:34,803 - WARNING - [TAP]00:00:01.030 : CLI : Warning : External parameter 'Extra Parameter' does not exist in the test plan.
2020-01-27 16:07:34,803 - WARNING - [TAP]00:00:01.030 : CLI : Warning : Statement 'Extra Parameter=This is a message -> 30' has no effect.
2020-01-27 16:07:34,803 - INFO - [TAP]00:00:01.031 : Main : Information : Unable to continue. Now exiting OpenTAP CLI.
```

Portal



Status: Finished Created: January 27, 2020 4:07 PM (3 days ago)

Pre-Run

Started: January 27, 2020 4:07 PM (3 days ago, waited a few seconds)

Finished: January 27, 2020 4:07 PM (3 days ago, ran for a few seconds)

```
2020-01-27 16:07:23,733 - DEBUG - [File Opened]
2020-01-27 16:07:23,733 - DEBUG - [Using temporal folder: Temp\tmpvf74fbb1]
2020-01-27 16:07:23,733 - INFO - Started
2020-01-27 16:07:23,734 - INFO - [Starting Task Check Availability]
2020-01-27 16:07:23,734 - DEBUG - Params: {'Id': 30, 'Available': False}
2020-01-27 16:07:23,734 - INFO - Requesting availability
2020-01-27 16:07:23,734 - INFO - Resources not available
2020-01-27 16:07:23,734 - INFO - [Task Check Availability finished]
2020-01-27 16:07:23,735 - DEBUG - Params: {'Id': 30, 'Available': False}
2020-01-27 16:07:24,736 - INFO - [Starting Task Check Availability]
2020-01-27 16:07:24,736 - DEBUG - Params: {'Id': 30, 'Available': False}
2020-01-27 16:07:24,736 - INFO - Requesting availability
2020-01-27 16:07:24,736 - INFO - Resources available
2020-01-27 16:07:24,736 - INFO - [Task Check Availability finished]
2020-01-27 16:07:24,736 - DEBUG - Params: {'Id': 30, 'Available': True}
2020-01-27 16:07:24,737 - INFO - [Starting Task Add Execution Entry]
2020-01-27 16:07:24,737 - DEBUG - Params: {}
2020-01-27 16:07:24,737 - INFO - Sending entry information
2020-01-27 16:07:27,739 - INFO - Information sent
2020-01-27 16:07:27,739 - INFO - [Task Add Execution Entry finished]
2020-01-27 16:07:27,739 - DEBUG - Params: {}
2020-01-27 16:07:27,739 - INFO - [Starting Task Instantiate]
2020-01-27 16:07:27,739 - DEBUG - Params: {'HasNsd': False, 'ExperimentId': 6}
2020-01-27 16:07:27,740 - INFO - Instantiation not required, no NSD defined.
2020-01-27 16:07:27,740 - INFO - Instantiation completed
2020-01-27 16:07:27,740 - INFO - [Task Instantiate finished]
2020-01-27 16:07:27,740 - DEBUG - Params: {'HasNsd': False, 'ExperimentId': 6, 'SliceId': None}
2020-01-27 16:07:27,740 - INFO - Finished (status: Finished)
2020-01-27 16:07:27,741 - DEBUG - [Closing File]
```

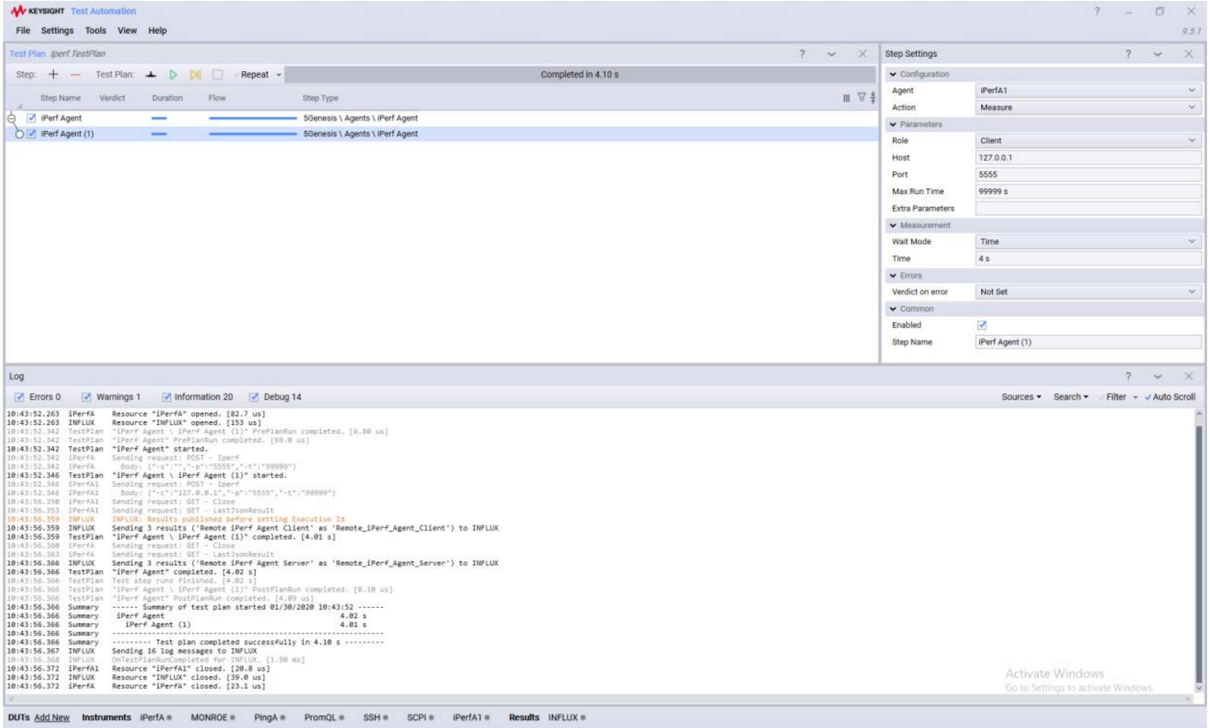
Run

Started: January 27, 2020 4:07 PM (3 days ago, waited a few seconds)

Finished: January 27, 2020 4:07 PM (3 days ago, ran for a few seconds)

```
2020-01-27 16:07:33,736 - DEBUG - [File Opened]
2020-01-27 16:07:33,736 - DEBUG - [Using temporal folder: Temp\tmpvf74fbb1]
2020-01-27 16:07:33,736 - INFO - Started
2020-01-27 16:07:33,737 - INFO - [Starting Task Message]
2020-01-27 16:07:33,737 - DEBUG - Params: {'Severity': 'INFO', 'Message': 'This is a TEST message'}
2020-01-27 16:07:33,737 - INFO - This is a TEST message
2020-01-27 16:07:33,738 - INFO - [Task Message finished]
2020-01-27 16:07:33,738 - DEBUG - Params: {'Severity': 'INFO', 'Message': 'This is a TEST message'}
2020-01-27 16:07:33,739 - INFO - [Starting Task Tap Execute]
2020-01-27 16:07:33,739 - DEBUG - Params: {'TestPlan': 'C:\\Program Files\\OpenTap\\Iperf.TestPlan', 'Externals': {'Extra Parameter': 'This is a message ->
```

ELCM


KEYSIGHT Test Automation
File Settings Tools View Help
9.5.1
Test Plan Iperf.TestPlan
Step: + — Test Plan: Repeat
Completed in 4.10 s
Step Settings
Log
Sources Search Filter ✓ Auto Scroll

	Step Name	Verdict	Duration	Flow	Step Type
☑	iPerf Agent		—		5Genesis \ Agents \ iPerf Agent
☑	iPerf Agent (1)		—		5Genesis \ Agents \ iPerf Agent

Configuration

Agent	iPerfA1
Action	Measure

Parameters

Role	Client
Host	127.0.0.1
Port	5555
Max Run Time	99999 s
Extra Parameters	

Measurement

Wait Mode	Time
Time	4 s

Errors

Verdict on error	Not Set

Common

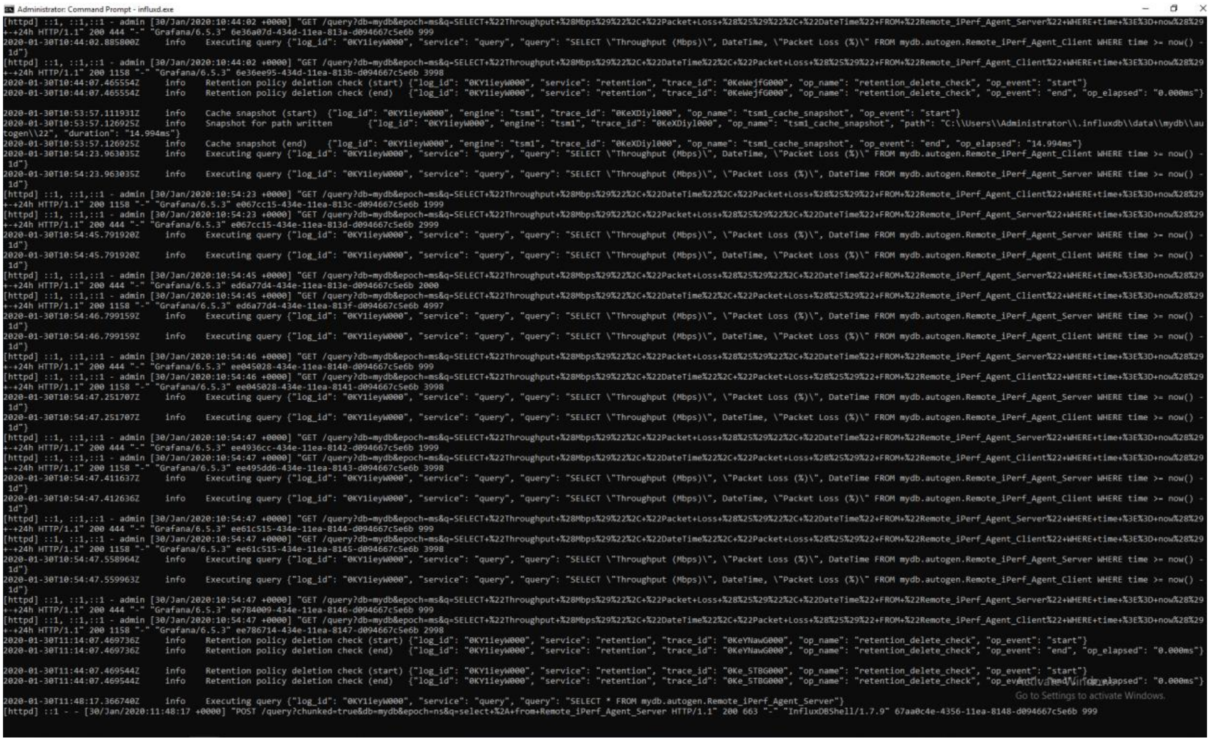
Enabled	☑
Step Name	iPerf Agent (1)

☑ Errors 0 ☑ Warnings 1 ☑ Information 20 ☑ Debug 14

```
10:43:52.263  iPerfA    Resource "iPerfA" opened. [82.7 us]
10:43:52.263  INFLUX    Resource "INFLUX" opened. [153 us]
10:43:52.342  TestPlan  "iPerf Agent \ iPerf Agent (1)" PrePlanRun completed. [6.80 us]
10:43:52.342  TestPlan  "iPerf Agent" PrePlanRun completed. [69.0 us]
10:43:52.342  TestPlan  "iPerf Agent" started.
10:43:52.342  iPerfA    Sending request: POST - Iperf
10:43:52.342  iPerfA      Body: {"-s":"","-p":"5555","-t":"99999"}
10:43:52.346  TestPlan  "iPerf Agent \ iPerf Agent (1)" started.
10:43:52.346  iPerfA1   Sending request: POST - Iperf
10:43:52.346  iPerfA1     Body: {"-c":"127.0.0.1","-p":"5555","-t":"99999"}
10:43:56.350  iPerfA1   Sending request: GET - Close
10:43:56.353  iPerfA1   Sending request: GET - LastJsonResult
10:43:56.359  INFLUX    INFLUX: Results published before setting Execution Id
10:43:56.359  INFLUX    Sending 3 results ('Remote iPerf Agent Client' as 'Remote_iPerf_Agent_Client') to INFLUX
10:43:56.359  TestPlan  "iPerf Agent \ iPerf Agent (1)" completed. [4.01 s]
10:43:56.360  iPerfA    Sending request: GET - Close
10:43:56.363  iPerfA    Sending request: GET - LastJsonResult
10:43:56.366  INFLUX    Sending 3 results ('Remote iPerf Agent Server' as 'Remote_iPerf_Agent_Server') to INFLUX
10:43:56.366  TestPlan  "iPerf Agent" completed. [4.02 s]
10:43:56.366  TestPlan  Test step runs finished. [4.02 s]
10:43:56.366  TestPlan  "iPerf Agent \ iPerf Agent (1)" PostPlanRun completed. [0.10 us]
10:43:56.366  TestPlan  "iPerf Agent" PostPlanRun completed. [4.09 us]
10:43:56.366  Summary   ------ Summary of test plan started 01/30/2020 10:43:52 ------
10:43:56.366  Summary    iPerf Agent                                           4.02 s
10:43:56.366  Summary      iPerf Agent (1)                                     4.01 s
10:43:56.366  Summary   -------------------------------------------------------------
10:43:56.366  Summary   --------- Test plan completed successfully in 4.10 s ---------
10:43:56.367  INFLUX    Sending 16 log messages to INFLUX
10:43:56.368  INFLUX    OnTestPlanRunCompleted for INFLUX. [1.50 ms]
10:43:56.372  iPerfA1   Resource "iPerfA1" closed. [20.8 us]
10:43:56.372  INFLUX    Resource "INFLUX" closed. [39.0 us]
10:43:56.372  iPerfA    Resource "iPerfA" closed. [23.1 us]
```

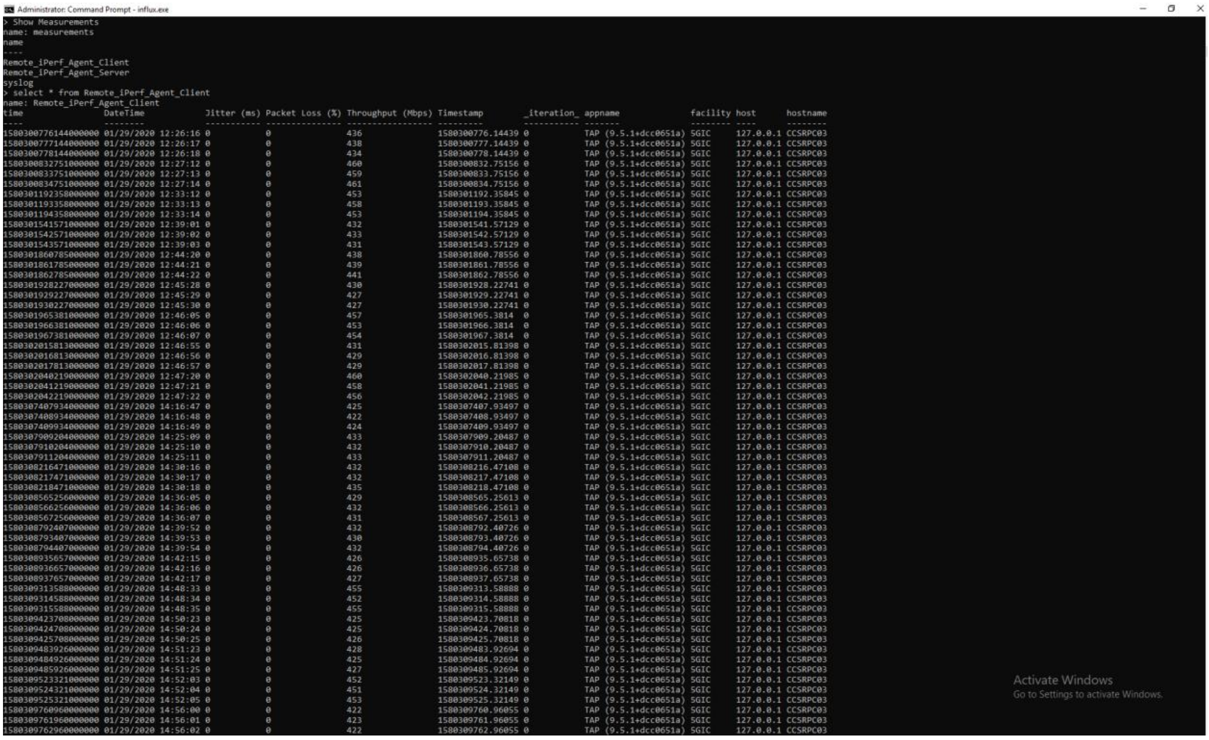
DUTs Add New Instruments iPerfA MONROE PingA PromQL SSH SCPI iPerfA1 Results INFLUX

TAP



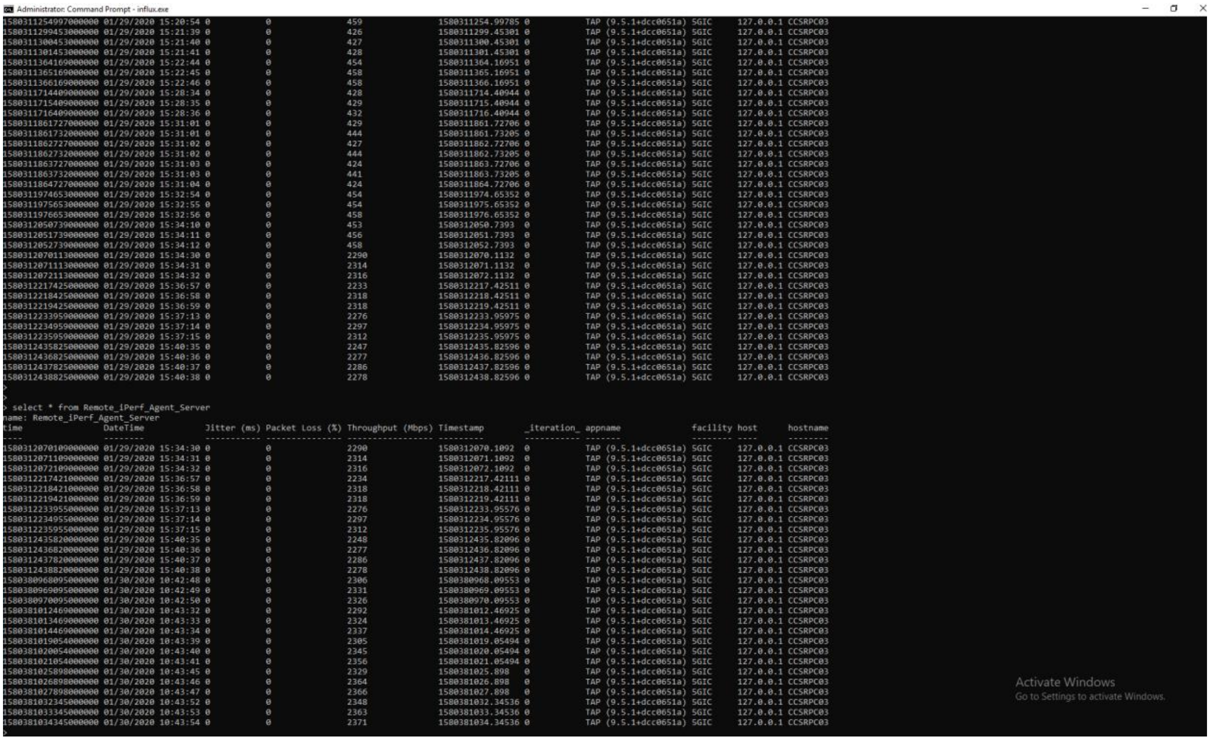
```
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:44:02 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" 6e36a07d-434d-11ea-813a-d094667c5e6b 999
2020-01-30T10:44:02.885800Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:44:02 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" 6e36ee95-434d-11ea-813b-d094667c5e6b 3998
2020-01-30T10:44:07.465554Z     info    Retention policy deletion check (start) {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0KeWejfG000", "op_name": "retention_delete_check", "op_event": "start"}
2020-01-30T10:44:07.465554Z     info    Retention policy deletion check (end)   {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0KeWejfG000", "op_name": "retention_delete_check", "op_event": "end", "op_elapsed": "0.000ms"}

2020-01-30T10:53:57.111931Z     info    Cache snapshot (start)  {"log_id": "0KY1ieyW000", "engine": "tsm1", "trace_id": "0KeXDiyl000", "op_name": "tsm1_cache_snapshot", "op_event": "start"}
2020-01-30T10:53:57.126925Z     info    Snapshot for path written        {"log_id": "0KY1ieyW000", "engine": "tsm1", "trace_id": "0KeXDiyl000", "op_name": "tsm1_cache_snapshot", "path": "C:\\Users\\Administrator\\.influxdb\\data\\mydb\\au
togen\\22", "duration": "14.994ms"}
2020-01-30T10:53:57.126925Z     info    Cache snapshot (end)    {"log_id": "0KY1ieyW000", "engine": "tsm1", "trace_id": "0KeXDiyl000", "op_name": "tsm1_cache_snapshot", "op_event": "end", "op_elapsed": "14.994ms"}
2020-01-30T10:54:23.963035Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
2020-01-30T10:54:23.963035Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:23 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" e067cc15-434e-11ea-813c-d094667c5e6b 1999
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:23 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" e067cc15-434e-11ea-813d-d094667c5e6b 2999
2020-01-30T10:54:45.791920Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
2020-01-30T10:54:45.791920Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:45 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" ed6a77d4-434e-11ea-813e-d094667c5e6b 2000
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:45 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" ed6a77d4-434e-11ea-813f-d094667c5e6b 4997
2020-01-30T10:54:46.799159Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
2020-01-30T10:54:46.799159Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:46 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" ee045028-434e-11ea-8140-d094667c5e6b 999
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:46 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" ee045028-434e-11ea-8141-d094667c5e6b 3998
2020-01-30T10:54:47.251707Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
2020-01-30T10:54:47.251707Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" ee4936cc-434e-11ea-8142-d094667c5e6b 1999
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" ee495dd6-434e-11ea-8143-d094667c5e6b 3998
2020-01-30T10:54:47.411637Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
2020-01-30T10:54:47.412636Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" ee61c515-434e-11ea-8144-d094667c5e6b 999
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" ee61c515-434e-11ea-8145-d094667c5e6b 3998
2020-01-30T10:54:47.558964Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", \"Packet Loss (%)\", DateTime FROM mydb.autogen.Remote_iPerf_Agent_Server WHERE time >= now() -
 1d"}
2020-01-30T10:54:47.559963Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT \"Throughput (Mbps)\", DateTime, \"Packet Loss (%)\" FROM mydb.autogen.Remote_iPerf_Agent_Client WHERE time >= now() -
 1d"}
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22Packet+Loss+%28%25%29%22%2C+%22DateTime%22+FROM+%22Remote_iPerf_Agent_Server%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 444 "-" "Grafana/6.5.3" ee784009-434e-11ea-8146-d094667c5e6b 999
[httpd] ::1, ::1,::1 - admin [30/Jan/2020:10:54:47 +0000] "GET /query?db=mydb&epoch=ms&q=SELECT+%22Throughput+%28Mbps%29%22%2C+%22DateTime%22%2C+%22Packet+Loss+%28%25%29%22+FROM+%22Remote_iPerf_Agent_Client%22+WHERE+time+%3E%3D+now%28%29
+-+24h HTTP/1.1" 200 1158 "-" "Grafana/6.5.3" ee786714-434e-11ea-8147-d094667c5e6b 2998
2020-01-30T11:14:07.469736Z     info    Retention policy deletion check (start) {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0KeYNawG000", "op_name": "retention_delete_check", "op_event": "start"}
2020-01-30T11:14:07.469736Z     info    Retention policy deletion check (end)   {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0KeYNawG000", "op_name": "retention_delete_check", "op_event": "end", "op_elapsed": "0.000ms"}

2020-01-30T11:44:07.469544Z     info    Retention policy deletion check (start) {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0Ke_5TBG000", "op_name": "retention_delete_check", "op_event": "start"}
2020-01-30T11:44:07.469544Z     info    Retention policy deletion check (end)   {"log_id": "0KY1ieyW000", "service": "retention", "trace_id": "0Ke_5TBG000", "op_name": "retention_delete_check", "op_event": "end", "op_elapsed": "0.000ms"}

2020-01-30T11:48:17.366740Z     info    Executing query {"log_id": "0KY1ieyW000", "service": "query", "query": "SELECT * FROM mydb.autogen.Remote_iPerf_Agent_Server"}
[httpd] ::1 - - [30/Jan/2020:11:48:17 +0000] "POST /query?chunked=true&db=mydb&epoch=ns&q=select+%2A+from+Remote_iPerf_Agent_Server HTTP/1.1" 200 663 "-" "InfluxDBShell/1.7.9" 67aa0c4e-4356-11ea-8148-d094667c5e6b 999
```

InfluxDB Running in Background


Administrator: Command Prompt - influx.exe
Activate Windows
Go to Settings to activate Windows.

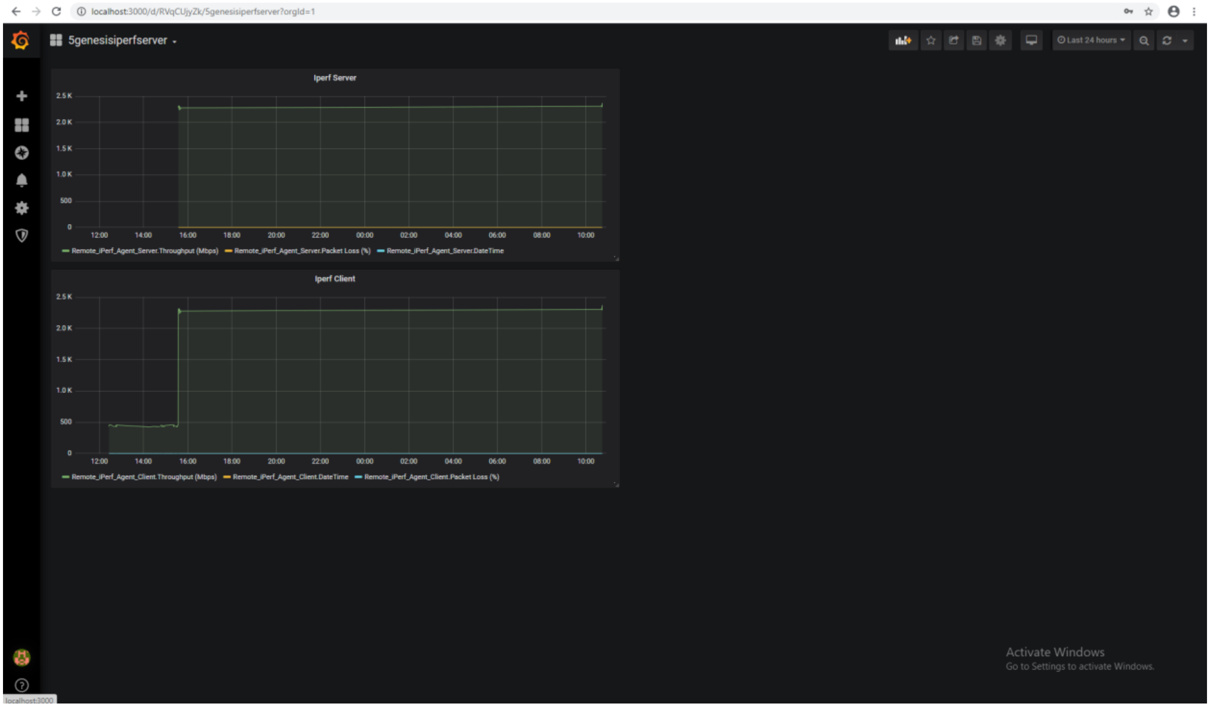
```
> Show Measurements
name: measurements
name
----
Remote_iPerf_Agent_Client
Remote_iPerf_Agent_Server
syslog
> select * from Remote_iPerf_Agent_Client
name: Remote_iPerf_Agent_Client
time                DateTime            Jitter (ms) Packet Loss (%) Throughput (Mbps) Timestamp        _iteration_ appname              facility host      hostname
----                --------            ----------- --------------- ----------------- ---------        ----------- -------              -------- ----      --------
1580300776144000000 01/29/2020 12:26:16 0           0               436               1580300776.14439 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580300777144000000 01/29/2020 12:26:17 0           0               438               1580300777.14439 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580300778144000000 01/29/2020 12:26:18 0           0               434               1580300778.14439 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580300832751000000 01/29/2020 12:27:12 0           0               460               1580300832.75156 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580300833751000000 01/29/2020 12:27:13 0           0               459               1580300833.75156 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580300834751000000 01/29/2020 12:27:14 0           0               461               1580300834.75156 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301192358000000 01/29/2020 12:33:12 0           0               453               1580301192.35845 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301193358000000 01/29/2020 12:33:13 0           0               458               1580301193.35845 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301194358000000 01/29/2020 12:33:14 0           0               453               1580301194.35845 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301541571000000 01/29/2020 12:39:01 0           0               432               1580301541.57129 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301542571000000 01/29/2020 12:39:02 0           0               433               1580301542.57129 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301543571000000 01/29/2020 12:39:03 0           0               431               1580301543.57129 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301860785000000 01/29/2020 12:44:20 0           0               438               1580301860.78556 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301861785000000 01/29/2020 12:44:21 0           0               439               1580301861.78556 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301862785000000 01/29/2020 12:44:22 0           0               441               1580301862.78556 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301928227000000 01/29/2020 12:45:28 0           0               430               1580301928.22741 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301929227000000 01/29/2020 12:45:29 0           0               427               1580301929.22741 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301930227000000 01/29/2020 12:45:30 0           0               427               1580301930.22741 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301965381000000 01/29/2020 12:46:05 0           0               457               1580301965.3814  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301966381000000 01/29/2020 12:46:06 0           0               453               1580301966.3814  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580301967381000000 01/29/2020 12:46:07 0           0               454               1580301967.3814  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302015813000000 01/29/2020 12:46:55 0           0               431               1580302015.81398 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302016813000000 01/29/2020 12:46:56 0           0               429               1580302016.81398 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302017813000000 01/29/2020 12:46:57 0           0               429               1580302017.81398 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302040219000000 01/29/2020 12:47:20 0           0               460               1580302040.21985 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302041219000000 01/29/2020 12:47:21 0           0               458               1580302041.21985 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580302042219000000 01/29/2020 12:47:22 0           0               456               1580302042.21985 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307407934000000 01/29/2020 14:16:47 0           0               425               1580307407.93497 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307408934000000 01/29/2020 14:16:48 0           0               422               1580307408.93497 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307409934000000 01/29/2020 14:16:49 0           0               424               1580307409.93497 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307909204000000 01/29/2020 14:25:09 0           0               433               1580307909.20487 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307910204000000 01/29/2020 14:25:10 0           0               432               1580307910.20487 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580307911204000000 01/29/2020 14:25:11 0           0               433               1580307911.20487 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308216471000000 01/29/2020 14:30:16 0           0               432               1580308216.47108 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308217471000000 01/29/2020 14:30:17 0           0               432               1580308217.47108 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308218471000000 01/29/2020 14:30:18 0           0               435               1580308218.47108 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308565256000000 01/29/2020 14:36:05 0           0               429               1580308565.25613 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308566256000000 01/29/2020 14:36:06 0           0               432               1580308566.25613 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308567256000000 01/29/2020 14:36:07 0           0               431               1580308567.25613 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308792407000000 01/29/2020 14:39:52 0           0               432               1580308792.40726 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308793407000000 01/29/2020 14:39:53 0           0               430               1580308793.40726 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308794407000000 01/29/2020 14:39:54 0           0               432               1580308794.40726 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308935657000000 01/29/2020 14:42:15 0           0               426               1580308935.65738 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308936657000000 01/29/2020 14:42:16 0           0               426               1580308936.65738 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580308937657000000 01/29/2020 14:42:17 0           0               427               1580308937.65738 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309313588000000 01/29/2020 14:48:33 0           0               455               1580309313.58888 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309314588000000 01/29/2020 14:48:34 0           0               452               1580309314.58888 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309315588000000 01/29/2020 14:48:35 0           0               455               1580309315.58888 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309423708000000 01/29/2020 14:50:23 0           0               425               1580309423.70818 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309424708000000 01/29/2020 14:50:24 0           0               425               1580309424.70818 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309425708000000 01/29/2020 14:50:25 0           0               426               1580309425.70818 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309483926000000 01/29/2020 14:51:23 0           0               428               1580309483.92694 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309484926000000 01/29/2020 14:51:24 0           0               425               1580309484.92694 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309485926000000 01/29/2020 14:51:25 0           0               427               1580309485.92694 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309523321000000 01/29/2020 14:52:03 0           0               452               1580309523.32149 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309524321000000 01/29/2020 14:52:04 0           0               451               1580309524.32149 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309525321000000 01/29/2020 14:52:05 0           0               453               1580309525.32149 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309760960000000 01/29/2020 14:56:00 0           0               422               1580309760.96055 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309761960000000 01/29/2020 14:56:01 0           0               423               1580309761.96055 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580309762960000000 01/29/2020 14:56:02 0           0               422               1580309762.96055 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
```

InfluxDB-iPerf Client Data


Administrator: Command Prompt - influx.exe
Activate Windows
Go to Settings to activate Windows.

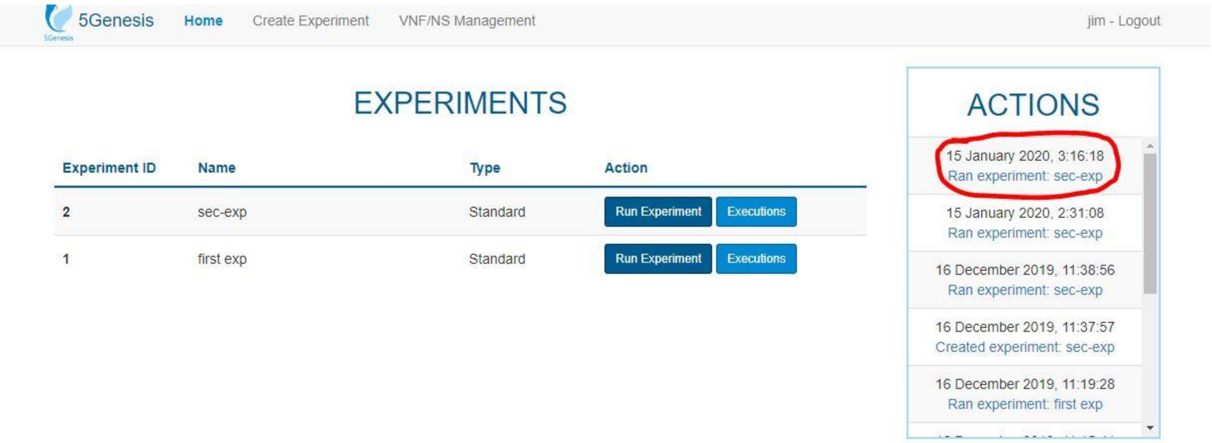
```
1580311254997000000 01/29/2020 15:20:54 0           0               459               1580311254.99785 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311299453000000 01/29/2020 15:21:39 0           0               426               1580311299.45301 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311300453000000 01/29/2020 15:21:40 0           0               427               1580311300.45301 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311301453000000 01/29/2020 15:21:41 0           0               428               1580311301.45301 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311364169000000 01/29/2020 15:22:44 0           0               454               1580311364.16951 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311365169000000 01/29/2020 15:22:45 0           0               458               1580311365.16951 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311366169000000 01/29/2020 15:22:46 0           0               458               1580311366.16951 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311714409000000 01/29/2020 15:28:34 0           0               428               1580311714.40944 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311715409000000 01/29/2020 15:28:35 0           0               429               1580311715.40944 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311716409000000 01/29/2020 15:28:36 0           0               432               1580311716.40944 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311861727000000 01/29/2020 15:31:01 0           0               429               1580311861.72706 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311861732000000 01/29/2020 15:31:01 0           0               444               1580311861.73205 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311862727000000 01/29/2020 15:31:02 0           0               427               1580311862.72706 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311862732000000 01/29/2020 15:31:02 0           0               444               1580311862.73205 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311863727000000 01/29/2020 15:31:03 0           0               424               1580311863.72706 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311863732000000 01/29/2020 15:31:03 0           0               441               1580311863.73205 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311864727000000 01/29/2020 15:31:04 0           0               424               1580311864.72706 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311974653000000 01/29/2020 15:32:54 0           0               454               1580311974.65352 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311975653000000 01/29/2020 15:32:55 0           0               454               1580311975.65352 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580311976653000000 01/29/2020 15:32:56 0           0               458               1580311976.65352 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312050739000000 01/29/2020 15:34:10 0           0               453               1580312050.7393  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312051739000000 01/29/2020 15:34:11 0           0               456               1580312051.7393  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312052739000000 01/29/2020 15:34:12 0           0               458               1580312052.7393  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312070113000000 01/29/2020 15:34:30 0           0               2290              1580312070.1132  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312071113000000 01/29/2020 15:34:31 0           0               2314              1580312071.1132  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312072113000000 01/29/2020 15:34:32 0           0               2316              1580312072.1132  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312217425000000 01/29/2020 15:36:57 0           0               2233              1580312217.42511 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312218425000000 01/29/2020 15:36:58 0           0               2318              1580312218.42511 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312219425000000 01/29/2020 15:36:59 0           0               2318              1580312219.42511 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312233959000000 01/29/2020 15:37:13 0           0               2276              1580312233.95975 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312234959000000 01/29/2020 15:37:14 0           0               2297              1580312234.95975 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312235959000000 01/29/2020 15:37:15 0           0               2312              1580312235.95975 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312435825000000 01/29/2020 15:40:35 0           0               2247              1580312435.82596 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312436825000000 01/29/2020 15:40:36 0           0               2277              1580312436.82596 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312437825000000 01/29/2020 15:40:37 0           0               2286              1580312437.82596 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312438825000000 01/29/2020 15:40:38 0           0               2278              1580312438.82596 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
>
>
> select * from Remote_iPerf_Agent_Server
name: Remote_iPerf_Agent_Server
time                DateTime            Jitter (ms) Packet Loss (%) Throughput (Mbps) Timestamp        _iteration_ appname              facility host      hostname
----                --------            ----------- --------------- ----------------- ---------        ----------- -------              -------- ----      --------
1580312070109000000 01/29/2020 15:34:30 0           0               2290              1580312070.1092  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312071109000000 01/29/2020 15:34:31 0           0               2314              1580312071.1092  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312072109000000 01/29/2020 15:34:32 0           0               2316              1580312072.1092  0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312217421000000 01/29/2020 15:36:57 0           0               2234              1580312217.42111 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312218421000000 01/29/2020 15:36:58 0           0               2318              1580312218.42111 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312219421000000 01/29/2020 15:36:59 0           0               2318              1580312219.42111 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312233955000000 01/29/2020 15:37:13 0           0               2276              1580312233.95576 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312234955000000 01/29/2020 15:37:14 0           0               2297              1580312234.95576 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312235955000000 01/29/2020 15:37:15 0           0               2312              1580312235.95576 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312435820000000 01/29/2020 15:40:35 0           0               2248              1580312435.82096 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312436820000000 01/29/2020 15:40:36 0           0               2277              1580312436.82096 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312437820000000 01/29/2020 15:40:37 0           0               2286              1580312437.82096 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580312438820000000 01/29/2020 15:40:38 0           0               2278              1580312438.82096 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580380968095000000 01/30/2020 10:42:48 0           0               2306              1580380968.09553 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580380969095000000 01/30/2020 10:42:49 0           0               2331              1580380969.09553 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580380970095000000 01/30/2020 10:42:50 0           0               2326              1580380970.09553 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381012469000000 01/30/2020 10:43:32 0           0               2292              1580381012.46925 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381013469000000 01/30/2020 10:43:33 0           0               2324              1580381013.46925 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381014469000000 01/30/2020 10:43:34 0           0               2337              1580381014.46925 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381019054000000 01/30/2020 10:43:39 0           0               2305              1580381019.05494 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381020054000000 01/30/2020 10:43:40 0           0               2345              1580381020.05494 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381021054000000 01/30/2020 10:43:41 0           0               2356              1580381021.05494 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381025898000000 01/30/2020 10:43:45 0           0               2329              1580381025.898   0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381026898000000 01/30/2020 10:43:46 0           0               2364              1580381026.898   0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381027898000000 01/30/2020 10:43:47 0           0               2366              1580381027.898   0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381032345000000 01/30/2020 10:43:52 0           0               2348              1580381032.34536 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381033345000000 01/30/2020 10:43:53 0           0               2363              1580381033.34536 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
1580381034345000000 01/30/2020 10:43:54 0           0               2371              1580381034.34536 0           TAP (9.5.1+dcc0651a) 5GIC     127.0.0.1 CCSRPC03
>
```

InfluxDB-iPerf Server Data


localhost:3000/d/RVqCUjyZk/5genesisiperfserver?orgId=1
5genesisiperfserver
Last 24 hours
Iperf Server
2.5 K
2.0 K
1.5 K
1.0 K
500
0
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
Remote_iPerf_Agent_Server.Throughput (Mbps)
Remote_iPerf_Agent_Server.Packet Loss (%)
Remote_iPerf_Agent_Server.DateTime
Iperf Client
2.5 K
2.0 K
1.5 K
1.0 K
500
0
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
Remote_iPerf_Agent_Client.Throughput (Mbps)
Remote_iPerf_Agent_Client.DateTime
Remote_iPerf_Agent_Client.Packet Loss (%)
Activate Windows
Go to Settings to activate Windows.

Grafana visualizing data from InfluxDB (iPerf Client & iPerf Server)

Limassol Platform Integration activities (screenshots):



EXPERIMENTS

Experiment ID	Name	Type	Action
2	sec-exp	Standard	Run Experiment Executions
1	first exp	Standard	Run Experiment Executions

ACTIONS

15 January 2020, 3:16:18
Ran experiment: sec-exp

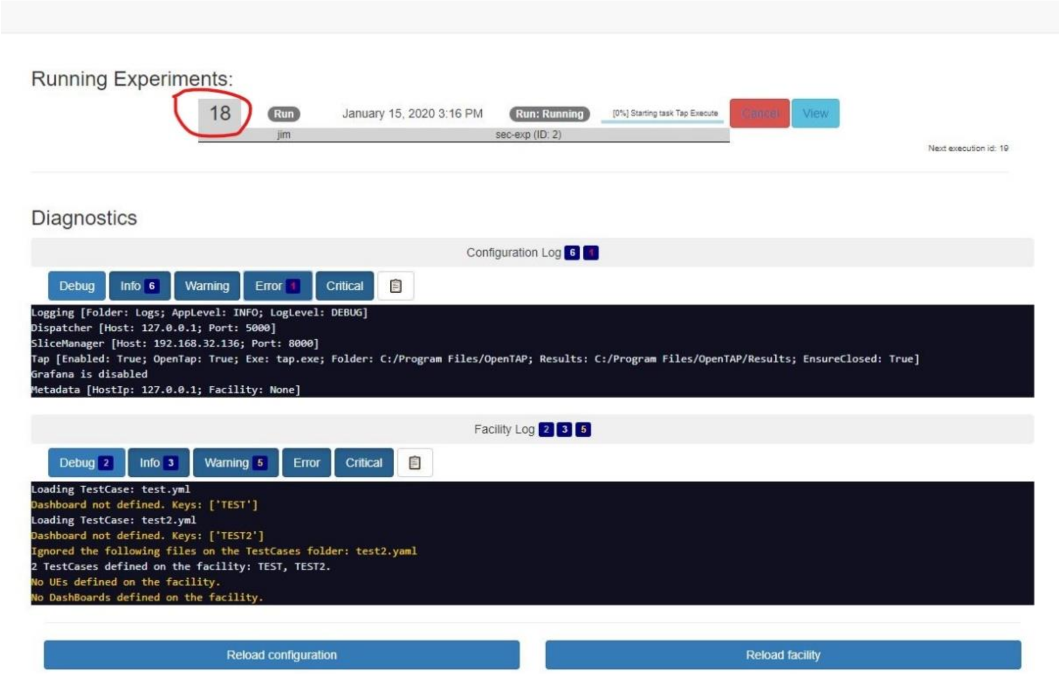
15 January 2020, 2:31:08
Ran experiment: sec-exp

16 December 2019, 11:38:56
Ran experiment: sec-exp

16 December 2019, 11:37:57
Created experiment: sec-exp

16 December 2019, 11:19:28
Ran experiment: first exp

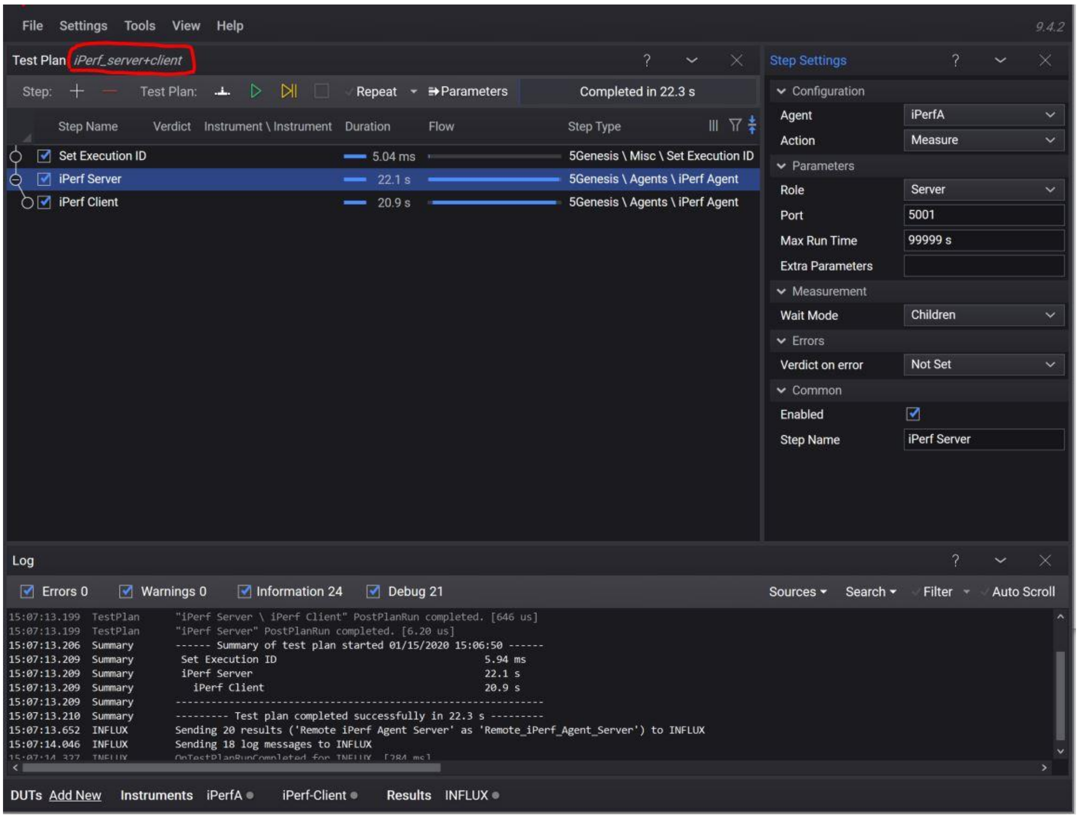
Experiment configured via the Portal



```
Logging [Folder: Logs; AppLevel: INFO; LogLevel: DEBUG]
Dispatcher [Host: 127.0.0.1; Port: 5000]
SliceManager [Host: 192.168.32.136; Port: 8000]
Tap [Enabled: True; OpenTap: True; Exe: tap.exe; Folder: C:/Program Files/OpenTAP; Results: C:/Program Files/OpenTAP/Results; EnsureClosed: True]
Grafana is disabled
Metadata [HostIp: 127.0.0.1; Facility: None]
```

```
Loading TestCase: test.yml
Dashboard not defined. Keys: ['TEST']
Loading TestCase: test2.yml
Dashboard not defined. Keys: ['TEST2']
Ignored the following files on the TestCases folder: test2.yaml
2 TestCases defined on the facility: TEST, TEST2.
No UEs defined on the facility.
No DashBoards defined on the facility.
```

Experiment monitored via the Portal



Step Name	Verdict	Instrument \ Instrument	Duration	Flow	Step Type
☑ Set Execution ID			5.04 ms		5Genesis \ Misc \ Set Execution ID
☑ iPerf Server			22.1 s		5Genesis \ Agents \ iPerf Agent
☑ iPerf Client			20.9 s		5Genesis \ Agents \ iPerf Agent

Configuration
Agent: iPerfA
Action: Measure

Parameters
Role: Server
Port: 5001
Max Run Time: 99999 s
Extra Parameters:

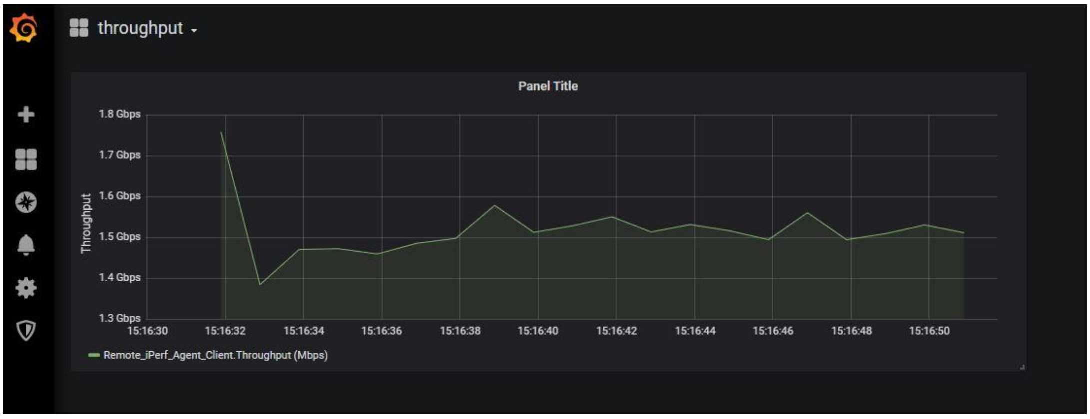
Measurement
Wait Mode: Children

Errors
Verdict on error: Not Set

Common
Enabled: ☑
Step Name: iPerf Server

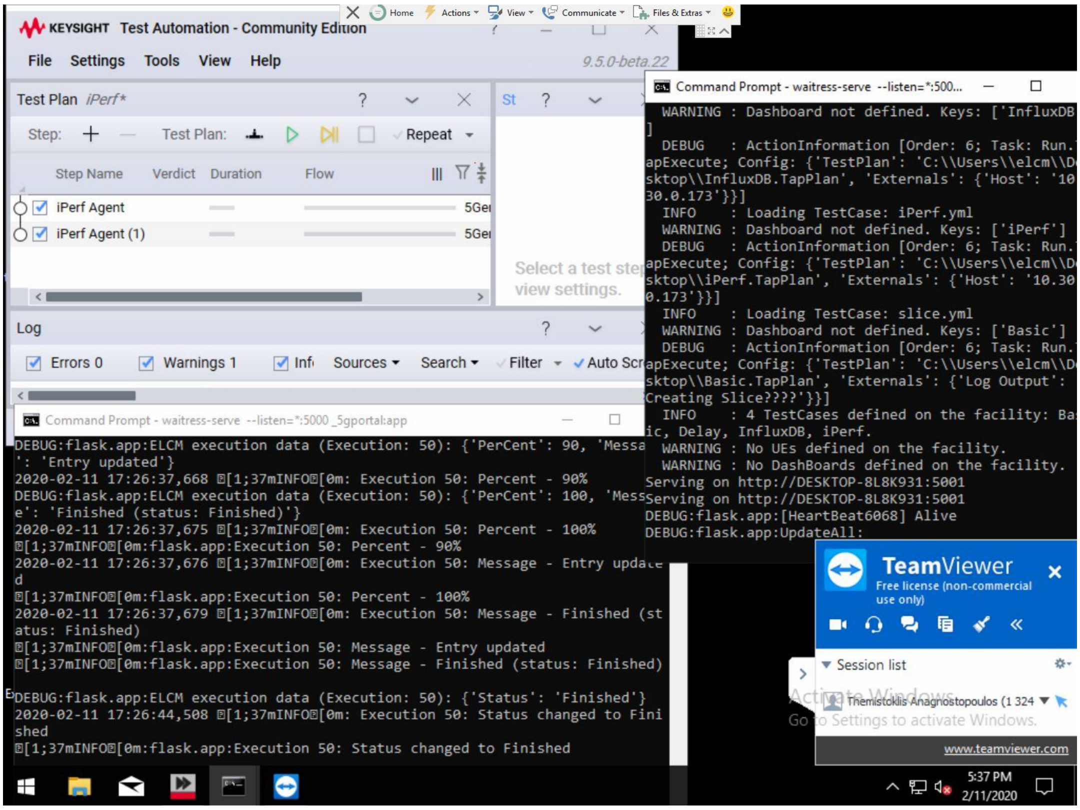
```
15:07:13.199  TestPlan     "iPerf Server \ iPerf Client" PostPlanRun completed. [646 us]
15:07:13.199  TestPlan     "iPerf Server" PostPlanRun completed. [6.20 us]
15:07:13.206  Summary      ------ Summary of test plan started 01/15/2020 15:06:50 ------
15:07:13.209  Summary       Set Execution ID                                   5.94 ms
15:07:13.209  Summary       iPerf Server                                       22.1 s
15:07:13.209  Summary         iPerf Client                                     20.9 s
15:07:13.209  Summary      ---------------------------------------------------------------
15:07:13.210  Summary      --------- Test plan completed successfully in 22.3 s ---------
15:07:13.652  INFLUX       Sending 20 results ('Remote iPerf Agent Server' as 'Remote_iPerf_Agent_Server') to INFLUX
15:07:14.046  INFLUX       Sending 18 log messages to INFLUX
15:07:14.327  INFLUX       OnTestPlanRunCompleted for INFLUX  [284 ms]
```

Experiment execution automated in OpenTAP



Experiment results visualized in Grafana front-end

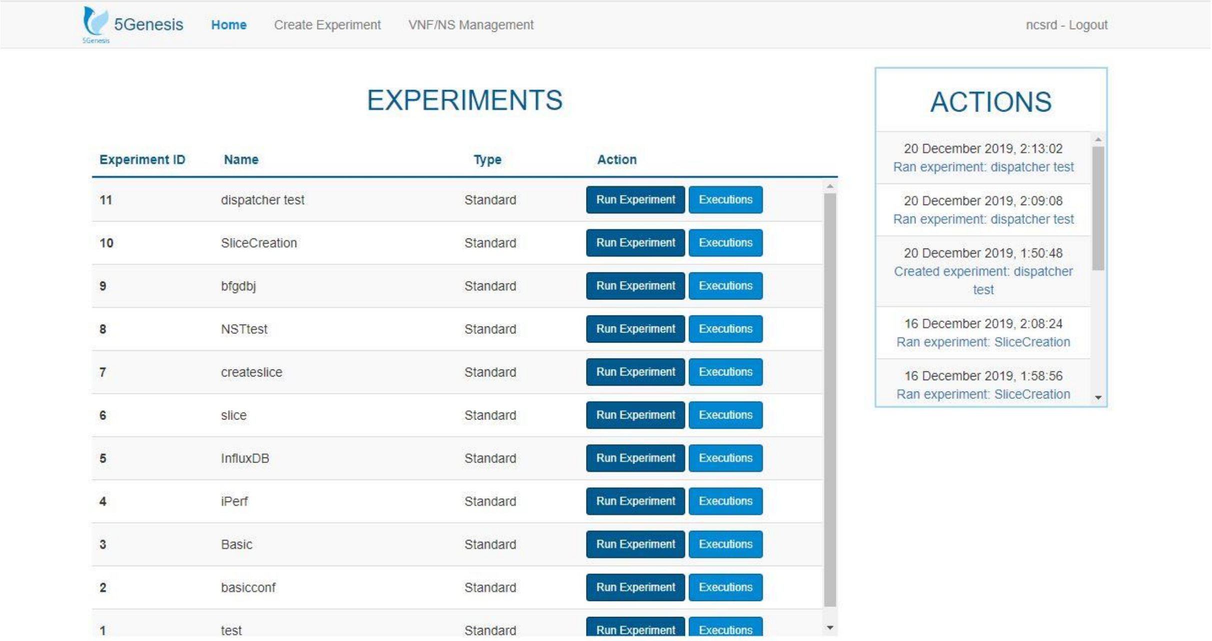
Athens Platform Integration activities (screenshots):



```
 WARNING : Dashboard not defined. Keys: ['InfluxDB
]
 DEBUG   : ActionInformation [Order: 6; Task: Run.
apExecute; Config: {'TestPlan': 'C:\\Users\\elcm\\De
sktop\\InfluxDB.TapPlan', 'Externals': {'Host': '10
30.0.173'}}]
 INFO    : Loading TestCase: iPerf.yml
 WARNING : Dashboard not defined. Keys: ['iPerf']
 DEBUG   : ActionInformation [Order: 6; Task: Run.
apExecute; Config: {'TestPlan': 'C:\\Users\\elcm\\De
sktop\\iPerf.TapPlan', 'Externals': {'Host': '10.30
0.173'}}]
 INFO    : Loading TestCase: slice.yml
 WARNING : Dashboard not defined. Keys: ['Basic']
 DEBUG   : ActionInformation [Order: 6; Task: Run.
apExecute; Config: {'TestPlan': 'C:\\Users\\elcm\\De
sktop\\Basic.TapPlan', 'Externals': {'Log Output':
Creating Slice????'}}]
 INFO    : 4 TestCases defined on the facility: Bas
ic, Delay, InfluxDB, iPerf.
 WARNING : No UEs defined on the facility.
 WARNING : No DashBoards defined on the facility.
Serving on http://DESKTOP-8L8K931:5001
Serving on http://DESKTOP-8L8K931:5001
DEBUG:flask.app:[HeartBeat6068] Alive
DEBUG:flask.app:UpdateAll:
```

```
DEBUG:flask.app:ELCM execution data (Execution: 50): {'PerCent': 90, 'Messa
': 'Entry updated'}
2020-02-11 17:26:37,668 ⊡[1;37mINFO⊡[0m: Execution 50: Percent - 90%
DEBUG:flask.app:ELCM execution data (Execution: 50): {'PerCent': 100, 'Mess
e': 'Finished (status: Finished)'}
2020-02-11 17:26:37,675 ⊡[1;37mINFO⊡[0m: Execution 50: Percent - 100%
⊡[1;37mINFO⊡[0m:flask.app:Execution 50: Percent - 90%
2020-02-11 17:26:37,676 ⊡[1;37mINFO⊡[0m: Execution 50: Message - Entry update
d
⊡[1;37mINFO⊡[0m:flask.app:Execution 50: Percent - 100%
2020-02-11 17:26:37,679 ⊡[1;37mINFO⊡[0m: Execution 50: Message - Finished (st
atus: Finished)
⊡[1;37mINFO⊡[0m:flask.app:Execution 50: Message - Entry updated
⊡[1;37mINFO⊡[0m:flask.app:Execution 50: Message - Finished (status: Finished)

DEBUG:flask.app:ELCM execution data (Execution: 50): {'Status': 'Finished'}
2020-02-11 17:26:44,508 ⊡[1;37mINFO⊡[0m: Execution 50: Status changed to Fini
shed
⊡[1;37mINFO⊡[0m:flask.app:Execution 50: Status changed to Finished
```

Portal – ELCM – OpenTAP VM



EXPERIMENTS

Experiment ID	Name	Type	Action
11	dispatcher test	Standard	Run Experiment Executions
10	SliceCreation	Standard	Run Experiment Executions
9	bfgdbj	Standard	Run Experiment Executions
8	NSTtest	Standard	Run Experiment Executions
7	createslice	Standard	Run Experiment Executions
6	slice	Standard	Run Experiment Executions
5	InfluxDB	Standard	Run Experiment Executions
4	iPerf	Standard	Run Experiment Executions
3	Basic	Standard	Run Experiment Executions
2	basicconf	Standard	Run Experiment Executions
1	test	Standard	Run Experiment Executions

ACTIONS

20 December 2019, 2:13:02
Ran experiment: dispatcher test

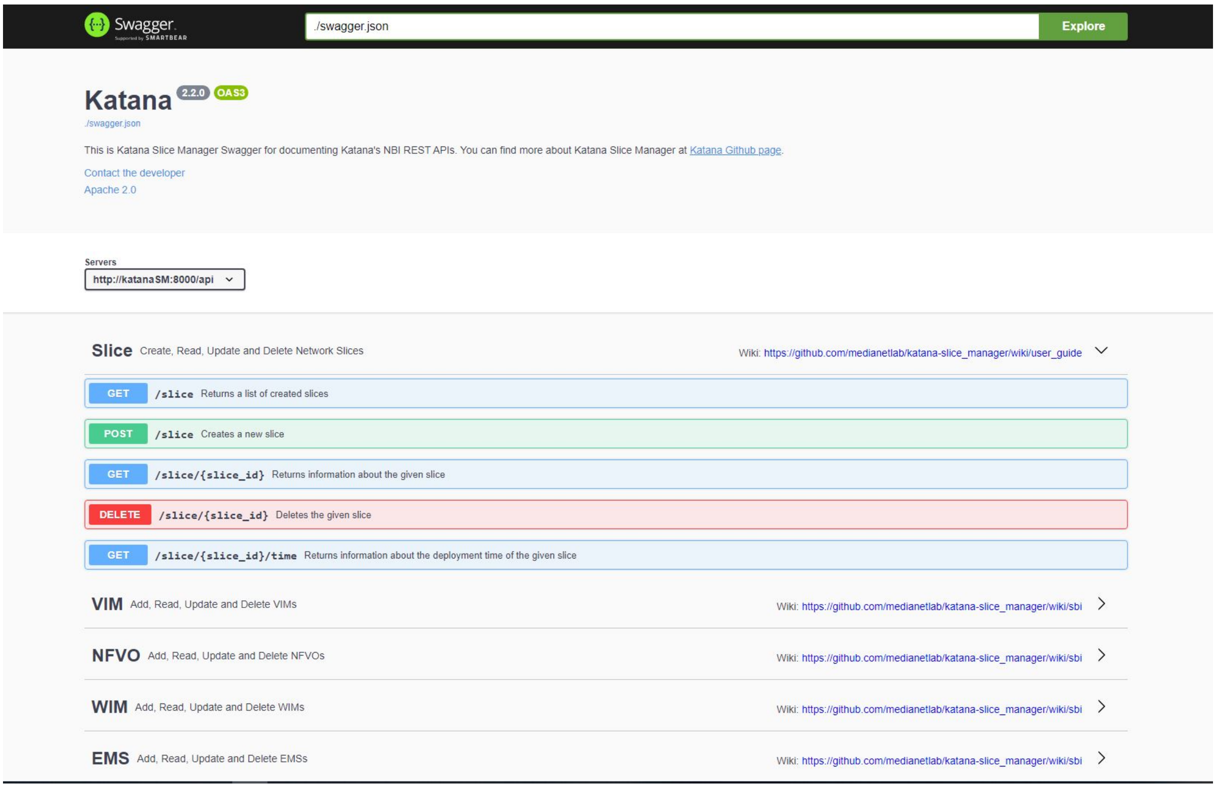
20 December 2019, 2:09:08
Ran experiment: dispatcher test

20 December 2019, 1:50:48
Created experiment: dispatcher test

16 December 2019, 2:08:24
Ran experiment: SliceCreation

16 December 2019, 1:58:56
Ran experiment: SliceCreation

Portal WEB UI



Katana

./swagger.json

This is Katana Slice Manager Swagger for documenting Katana's NBI REST APIs. You can find more about Katana Slice Manager at Katana Github page.

Contact the developer

Apache 2.0

Servers
http://katanaSM:8000/api

Slice Create, Read, Update and Delete Network Slices — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/user_guide

- GET `/slice` Returns a list of created slices
- POST `/slice` Creates a new slice
- GET `/slice/{slice_id}` Returns information about the given slice
- DELETE `/slice/{slice_id}` Deletes the given slice
- GET `/slice/{slice_id}/time` Returns information about the deployment time of the given slice

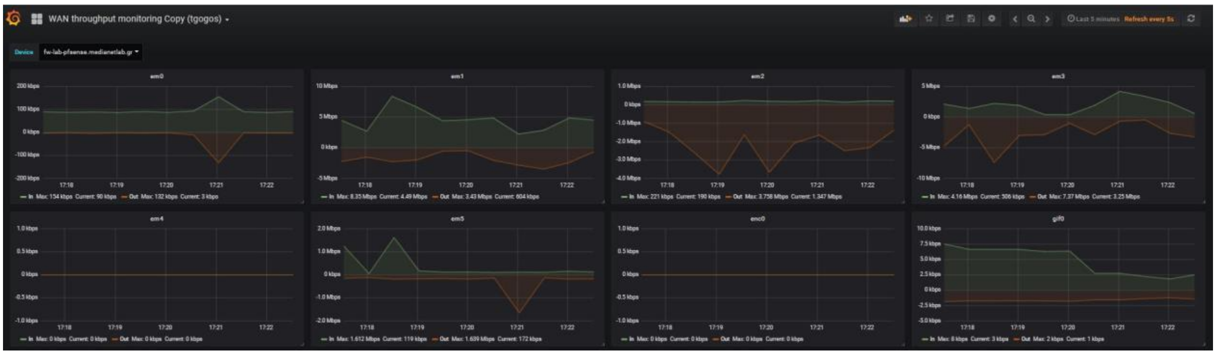
VIM Add, Read, Update and Delete VIMs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi

NFVO Add, Read, Update and Delete NFVOs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi

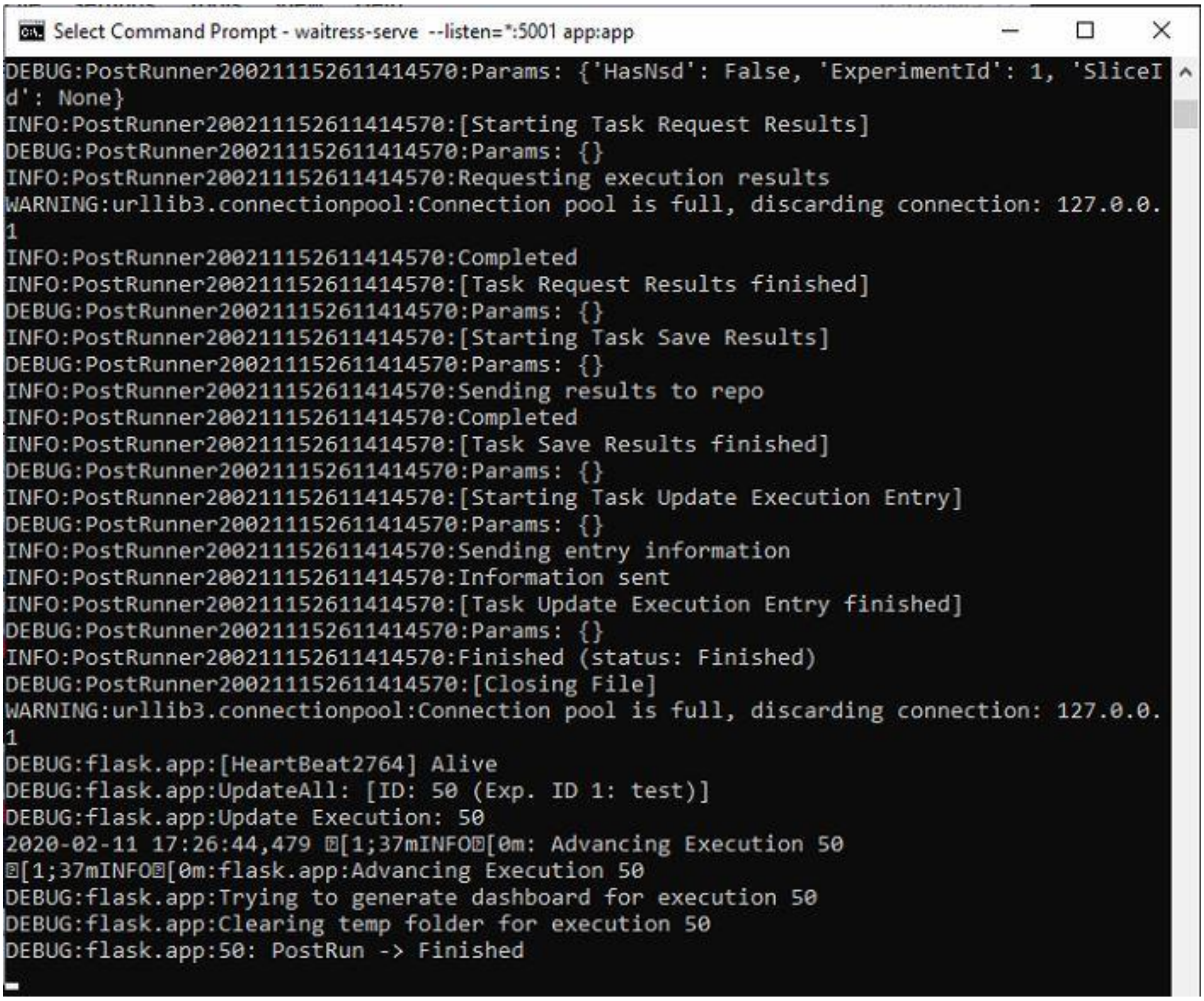
WIM Add, Read, Update and Delete WIMs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi

EMS Add, Read, Update and Delete EMSs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi

Slice Manager Swagger


WAN throughput monitoring Copy (tgogos)
Device fw-lab-pfsense.medianetlab.gr
Last 5 minutes Refresh every 5s
em0
em1
em2
em3
em4
em5
enc0
gif0
17:18
17:19
17:20
17:21
17:22
In Max: 134 kbps Current: 90 kbps Out Max: 132 kbps Current: 3 kbps
In Max: 8.35 Mbps Current: 4.49 Mbps Out Max: 3.43 Mbps Current: 604 kbps
In Max: 221 kbps Current: 190 kbps Out Max: 3.758 Mbps Current: 1.347 Mbps
In Max: 4.16 Mbps Current: 506 kbps Out Max: 7.37 Mbps Current: 3.25 Mbps
In Max: 0 kbps Current: 0 kbps Out Max: 0 kbps Current: 0 kbps
In Max: 1.612 Mbps Current: 119 kbps Out Max: 1.639 Mbps Current: 172 kbps
In Max: 0 kbps Current: 0 kbps Out Max: 0 kbps Current: 0 kbps
In Max: 8 kbps Current: 3 kbps Out Max: 2 kbps Current: 1 kbps

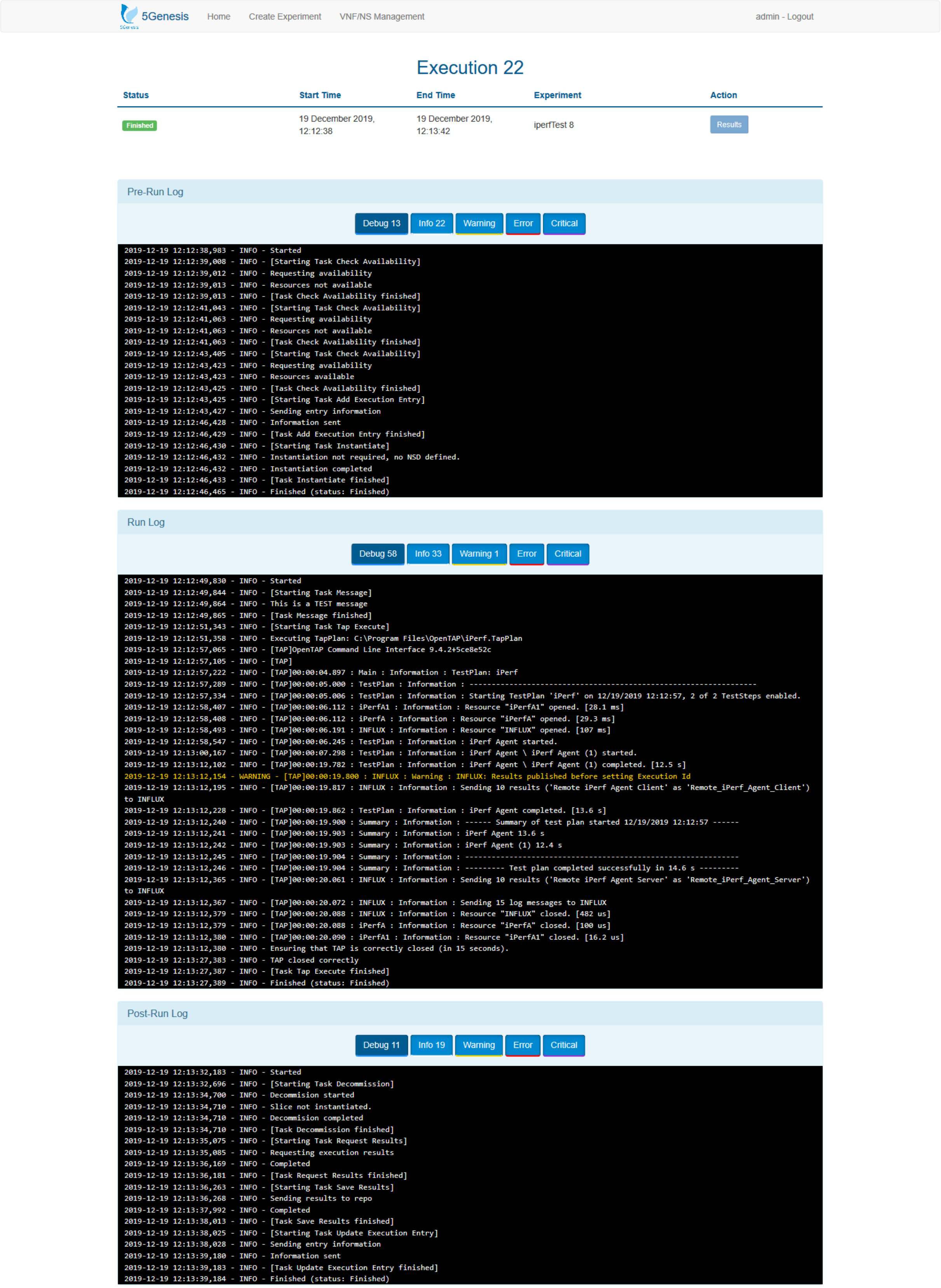
Throughput Experiment Results in Grafana


Select Command Prompt - waitress-serve --listen=*:5001 app:app

```
DEBUG:PostRunner20021115261141457O:Params: {'HasNsd': False, 'ExperimentId': 1, 'SliceI
d': None}
INFO:PostRunner200211152611414570:[Starting Task Request Results]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:Requesting execution results
WARNING:urllib3.connectionpool:Connection pool is full, discarding connection: 127.0.0.
1
INFO:PostRunner200211152611414570:Completed
INFO:PostRunner200211152611414570:[Task Request Results finished]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:[Starting Task Save Results]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:Sending results to repo
INFO:PostRunner200211152611414570:Completed
INFO:PostRunner200211152611414570:[Task Save Results finished]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:[Starting Task Update Execution Entry]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:Sending entry information
INFO:PostRunner200211152611414570:Information sent
INFO:PostRunner200211152611414570:[Task Update Execution Entry finished]
DEBUG:PostRunner200211152611414570:Params: {}
INFO:PostRunner200211152611414570:Finished (status: Finished)
DEBUG:PostRunner200211152611414570:[Closing File]
WARNING:urllib3.connectionpool:Connection pool is full, discarding connection: 127.0.0.
1
DEBUG:flask.app:[HeartBeat2764] Alive
DEBUG:flask.app:UpdateAll: [ID: 50 (Exp. ID 1: test)]
DEBUG:flask.app:Update Execution: 50
2020-02-11 17:26:44,479 □[1;37mINFO□[0m: Advancing Execution 50
□[1;37mINFO□[0m:flask.app:Advancing Execution 50
DEBUG:flask.app:Trying to generate dashboard for execution 50
DEBUG:flask.app:Clearing temp folder for execution 50
DEBUG:flask.app:50: PostRun -> Finished
```

Experiment Execution in ELCM

Berlin Platform Integration activities (screenshots):



Execution 22

Status	Start Time	End Time	Experiment	Action
Finished	19 December 2019, 12:12:38	19 December 2019, 12:13:42	iperfTest 8	Results

Pre-Run Log

```
2019-12-19 12:12:38,983 - INFO - Started
2019-12-19 12:12:39,008 - INFO - [Starting Task Check Availability]
2019-12-19 12:12:39,012 - INFO - Requesting availability
2019-12-19 12:12:39,013 - INFO - Resources not available
2019-12-19 12:12:39,013 - INFO - [Task Check Availability finished]
2019-12-19 12:12:41,043 - INFO - [Starting Task Check Availability]
2019-12-19 12:12:41,063 - INFO - Requesting availability
2019-12-19 12:12:41,063 - INFO - Resources not available
2019-12-19 12:12:41,063 - INFO - [Task Check Availability finished]
2019-12-19 12:12:43,405 - INFO - [Starting Task Check Availability]
2019-12-19 12:12:43,423 - INFO - Requesting availability
2019-12-19 12:12:43,423 - INFO - Resources available
2019-12-19 12:12:43,425 - INFO - [Task Check Availability finished]
2019-12-19 12:12:43,425 - INFO - [Starting Task Add Execution Entry]
2019-12-19 12:12:43,427 - INFO - Sending entry information
2019-12-19 12:12:46,428 - INFO - Information sent
2019-12-19 12:12:46,429 - INFO - [Task Add Execution Entry finished]
2019-12-19 12:12:46,430 - INFO - [Starting Task Instantiate]
2019-12-19 12:12:46,432 - INFO - Instantiation not required, no NSD defined.
2019-12-19 12:12:46,432 - INFO - Instantiation completed
2019-12-19 12:12:46,433 - INFO - [Task Instantiate finished]
2019-12-19 12:12:46,465 - INFO - Finished (status: Finished)
```

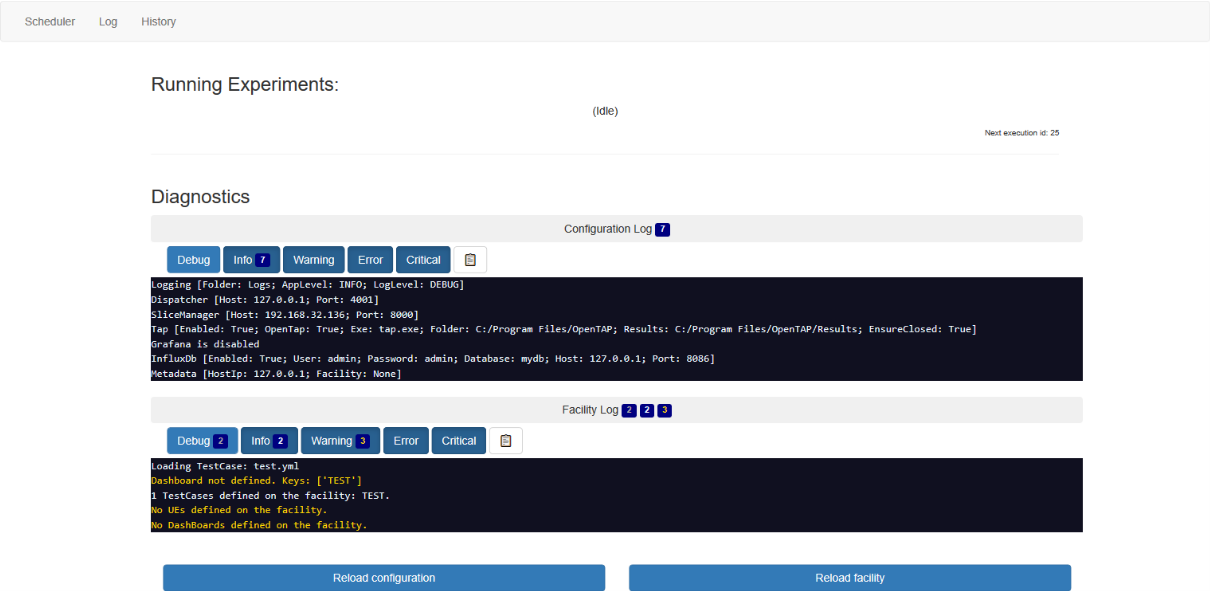
Run Log

```
2019-12-19 12:12:49,830 - INFO - Started
2019-12-19 12:12:49,844 - INFO - [Starting Task Message]
2019-12-19 12:12:49,864 - INFO - This is a TEST message
2019-12-19 12:12:49,865 - INFO - [Task Message finished]
2019-12-19 12:12:51,343 - INFO - [Starting Task Tap Execute]
2019-12-19 12:12:51,358 - INFO - Executing TapPlan: C:\Program Files\OpenTAP\iPerf.TapPlan
2019-12-19 12:12:57,065 - INFO - [TAP]OpenTAP Command Line Interface 9.4.2+5ce8e52c
2019-12-19 12:12:57,105 - INFO - [TAP]
2019-12-19 12:12:57,222 - INFO - [TAP]00:00:04.897 : Main : Information : TestPlan: iPerf
2019-12-19 12:12:57,289 - INFO - [TAP]00:00:05.000 : TestPlan : Information : -----------------------------------------------------------------
2019-12-19 12:12:57,334 - INFO - [TAP]00:00:05.006 : TestPlan : Information : Starting TestPlan 'iPerf' on 12/19/2019 12:12:57, 2 of 2 TestSteps enabled.
2019-12-19 12:12:58,407 - INFO - [TAP]00:00:06.112 : iPerfA1 : Information : Resource "iPerfA1" opened. [28.1 ms]
2019-12-19 12:12:58,408 - INFO - [TAP]00:00:06.112 : iPerfA : Information : Resource "iPerfA" opened. [29.3 ms]
2019-12-19 12:12:58,493 - INFO - [TAP]00:00:06.191 : INFLUX : Information : Resource "INFLUX" opened. [107 ms]
2019-12-19 12:12:58,547 - INFO - [TAP]00:00:06.245 : TestPlan : Information : iPerf Agent started.
2019-12-19 12:13:00,167 - INFO - [TAP]00:00:07.298 : TestPlan : Information : iPerf Agent \ iPerf Agent (1) started.
2019-12-19 12:13:12,102 - INFO - [TAP]00:00:19.782 : TestPlan : Information : iPerf Agent \ iPerf Agent (1) completed. [12.5 s]
2019-12-19 12:13:12,154 - WARNING - [TAP]00:00:19.800 : INFLUX : Warning : INFLUX: Results published before setting Execution Id
2019-12-19 12:13:12,195 - INFO - [TAP]00:00:19.817 : INFLUX : Information : Sending 10 results ('Remote iPerf Agent Client' as 'Remote_iPerf_Agent_Client')
to INFLUX
2019-12-19 12:13:12,228 - INFO - [TAP]00:00:19.862 : TestPlan : Information : iPerf Agent completed. [13.6 s]
2019-12-19 12:13:12,240 - INFO - [TAP]00:00:19.900 : Summary : Information : ------ Summary of test plan started 12/19/2019 12:12:57 ------
2019-12-19 12:13:12,241 - INFO - [TAP]00:00:19.903 : Summary : Information : iPerf Agent 13.6 s
2019-12-19 12:13:12,242 - INFO - [TAP]00:00:19.903 : Summary : Information : iPerf Agent (1) 12.4 s
2019-12-19 12:13:12,245 - INFO - [TAP]00:00:19.904 : Summary : Information : --------------------------------------------------------------
2019-12-19 12:13:12,246 - INFO - [TAP]00:00:19.904 : Summary : Information : --------- Test plan completed successfully in 14.6 s ---------
2019-12-19 12:13:12,365 - INFO - [TAP]00:00:20.061 : INFLUX : Information : Sending 10 results ('Remote iPerf Agent Server' as 'Remote_iPerf_Agent_Server')
to INFLUX
2019-12-19 12:13:12,367 - INFO - [TAP]00:00:20.072 : INFLUX : Information : Sending 15 log messages to INFLUX
2019-12-19 12:13:12,379 - INFO - [TAP]00:00:20.088 : INFLUX : Information : Resource "INFLUX" closed. [482 us]
2019-12-19 12:13:12,379 - INFO - [TAP]00:00:20.088 : iPerfA : Information : Resource "iPerfA" closed. [100 us]
2019-12-19 12:13:12,380 - INFO - [TAP]00:00:20.090 : iPerfA1 : Information : Resource "iPerfA1" closed. [16.2 us]
2019-12-19 12:13:12,380 - INFO - Ensuring that TAP is correctly closed (in 15 seconds).
2019-12-19 12:13:27,383 - INFO - TAP closed correctly
2019-12-19 12:13:27,387 - INFO - [Task Tap Execute finished]
2019-12-19 12:13:27,389 - INFO - Finished (status: Finished)
```

Post-Run Log

```
2019-12-19 12:13:32,183 - INFO - Started
2019-12-19 12:13:32,696 - INFO - [Starting Task Decommission]
2019-12-19 12:13:34,700 - INFO - Decommision started
2019-12-19 12:13:34,710 - INFO - Slice not instantiated.
2019-12-19 12:13:34,710 - INFO - Decommision completed
2019-12-19 12:13:34,710 - INFO - [Task Decommission finished]
2019-12-19 12:13:35,075 - INFO - [Starting Task Request Results]
2019-12-19 12:13:35,085 - INFO - Requesting execution results
2019-12-19 12:13:36,169 - INFO - Completed
2019-12-19 12:13:36,181 - INFO - [Task Request Results finished]
2019-12-19 12:13:36,263 - INFO - [Starting Task Save Results]
2019-12-19 12:13:36,268 - INFO - Sending results to repo
2019-12-19 12:13:37,992 - INFO - Completed
2019-12-19 12:13:38,013 - INFO - [Task Save Results finished]
2019-12-19 12:13:38,025 - INFO - [Starting Task Update Execution Entry]
2019-12-19 12:13:38,028 - INFO - Sending entry information
2019-12-19 12:13:39,180 - INFO - Information sent
2019-12-19 12:13:39,183 - INFO - [Task Update Execution Entry finished]
2019-12-19 12:13:39,184 - INFO - Finished (status: Finished)
```

Portal



Running Experiments:

(Idle)

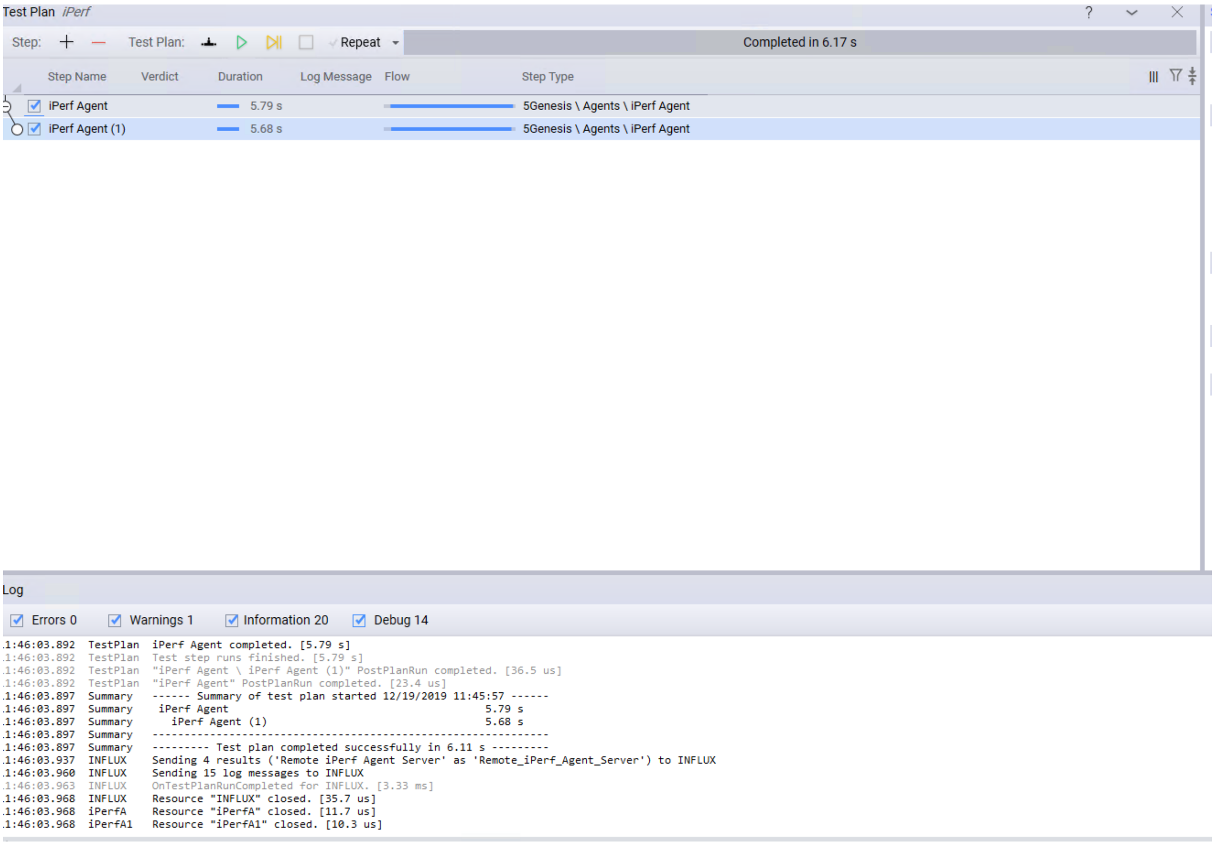
Next execution id: 25

Diagnostics

```
Logging [Folder: Logs; AppLevel: INFO; LogLevel: DEBUG]
Dispatcher [Host: 127.0.0.1; Port: 4001]
SliceManager [Host: 192.168.32.136; Port: 8000]
Tap [Enabled: True; OpenTap: True; Exe: tap.exe; Folder: C:/Program Files/OpenTAP; Results: C:/Program Files/OpenTAP/Results; EnsureClosed: True]
Grafana is disabled
InfluxDb [Enabled: True; User: admin; Password: admin; Database: mydb; Host: 127.0.0.1; Port: 8086]
Metadata [HostIp: 127.0.0.1; Facility: None]
```

```
Loading TestCase: test.yml
Dashboard not defined. Keys: ['TEST']
1 TestCases defined on the facility: TEST.
No UEs defined on the facility.
No DashBoards defined on the facility.
```

ELCM



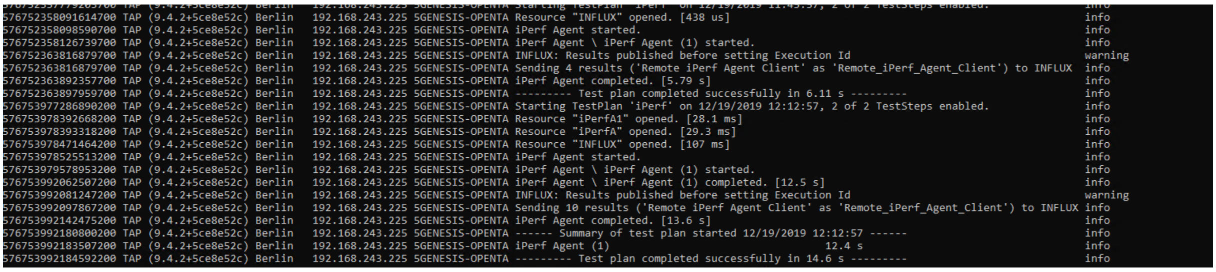
Step Name	Verdict	Duration	Log Message	Flow	Step Type
☑ iPerf Agent		5.79 s			5Genesis \ Agents \ iPerf Agent
☑ iPerf Agent (1)		5.68 s			5Genesis \ Agents \ iPerf Agent

Log

☑ Errors 0 ☑ Warnings 1 ☑ Information 20 ☑ Debug 14

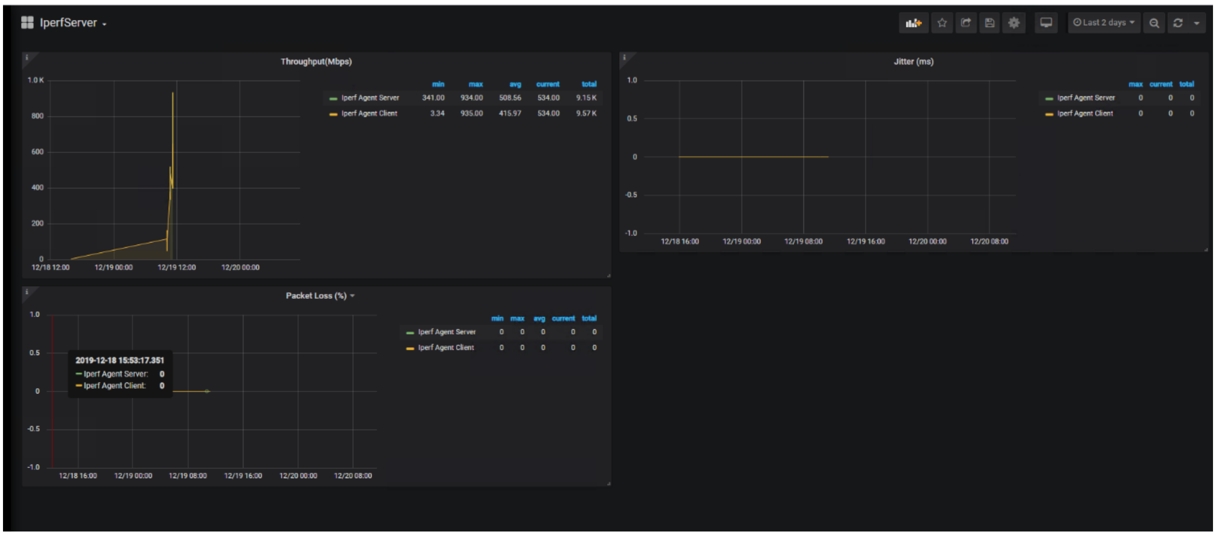
```
.1:46:03.892  TestPlan  iPerf Agent completed. [5.79 s]
.1:46:03.892  TestPlan  Test step runs finished. [5.79 s]
.1:46:03.892  TestPlan  "iPerf Agent \ iPerf Agent (1)" PostPlanRun completed. [36.5 us]
.1:46:03.892  TestPlan  "iPerf Agent" PostPlanRun completed. [23.4 us]
.1:46:03.897  Summary   ------ Summary of test plan started 12/19/2019 11:45:57 ------
.1:46:03.897  Summary    iPerf Agent                                           5.79 s
.1:46:03.897  Summary      iPerf Agent (1)                                     5.68 s
.1:46:03.897  Summary   ------------------------------------------------------------
.1:46:03.897  Summary   --------- Test plan completed successfully in 6.11 s ---------
.1:46:03.937  INFLUX    Sending 4 results ('Remote iPerf Agent Server' as 'Remote_iPerf_Agent_Server') to INFLUX
.1:46:03.960  INFLUX    Sending 15 log messages to INFLUX
.1:46:03.963  INFLUX    OnTestPlanRunCompleted for INFLUX. [3.33 ms]
.1:46:03.968  INFLUX    Resource "INFLUX" closed. [35.7 us]
.1:46:03.968  iPerfA    Resource "iPerfA" closed. [11.7 us]
.1:46:03.968  iPerfA1   Resource "iPerfA1" closed. [10.3 us]
```

OpenTAP



```
576752357779203700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Starting TestPlan 'iPerf' on 12/19/2019 11:43:57, 2 of 2 TestSteps enabled.                info
576752358091614700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Resource "INFLUX" opened. [438 us]                                                     info
576752358098590700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent started.                                                                    info
576752358126739700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent \ iPerf Agent (1) started.                                                  info
576752363816879700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA INFLUX: Results published before setting Execution Id                                  warning
576752363816879700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Sending 4 results ('Remote iPerf Agent Client' as 'Remote_iPerf_Agent_Client') to INFLUX  info
576752363892357700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent completed. [5.79 s]                                                         info
576752363897959700 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA --------- Test plan completed successfully in 6.11 s ---------                          info
576753977286890200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Starting TestPlan 'iPerf' on 12/19/2019 12:12:57, 2 of 2 TestSteps enabled.                info
576753978392668200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Resource "iPerfA1" opened. [28.1 ms]                                                    info
576753978393318200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Resource "iPerfA" opened. [29.3 ms]                                                     info
576753978471464200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Resource "INFLUX" opened. [107 ms]                                                      info
576753978525513200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent started.                                                                    info
576753979578953200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent \ iPerf Agent (1) started.                                                  info
576753992062507200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent \ iPerf Agent (1) completed. [12.5 s]                                       info
576753992081247200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA INFLUX: Results published before setting Execution Id                                  warning
576753992097867200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA Sending 10 results ('Remote iPerf Agent Client' as 'Remote_iPerf_Agent_Client') to INFLUX info
576753992142475200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent completed. [13.6 s]                                                         info
576753992180800200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA ------ Summary of test plan started 12/19/2019 12:12:57 ------                          info
576753992183507200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA iPerf Agent (1)                                      12.4 s                             info
576753992184592200 TAP (9.4.2+5ce8e52c) Berlin   192.168.243.225 5GENESIS-OPENTA --------- Test plan completed successfully in 14.6 s ---------                          info
```

InfluxDB



Throughput(Mbps)

	min	max	avg	current	total
Iperf Agent Server	341.00	934.00	508.56	534.00	9.15 K
Iperf Agent Client	3.34	935.00	415.97	534.00	9.57 K

Jitter (ms)

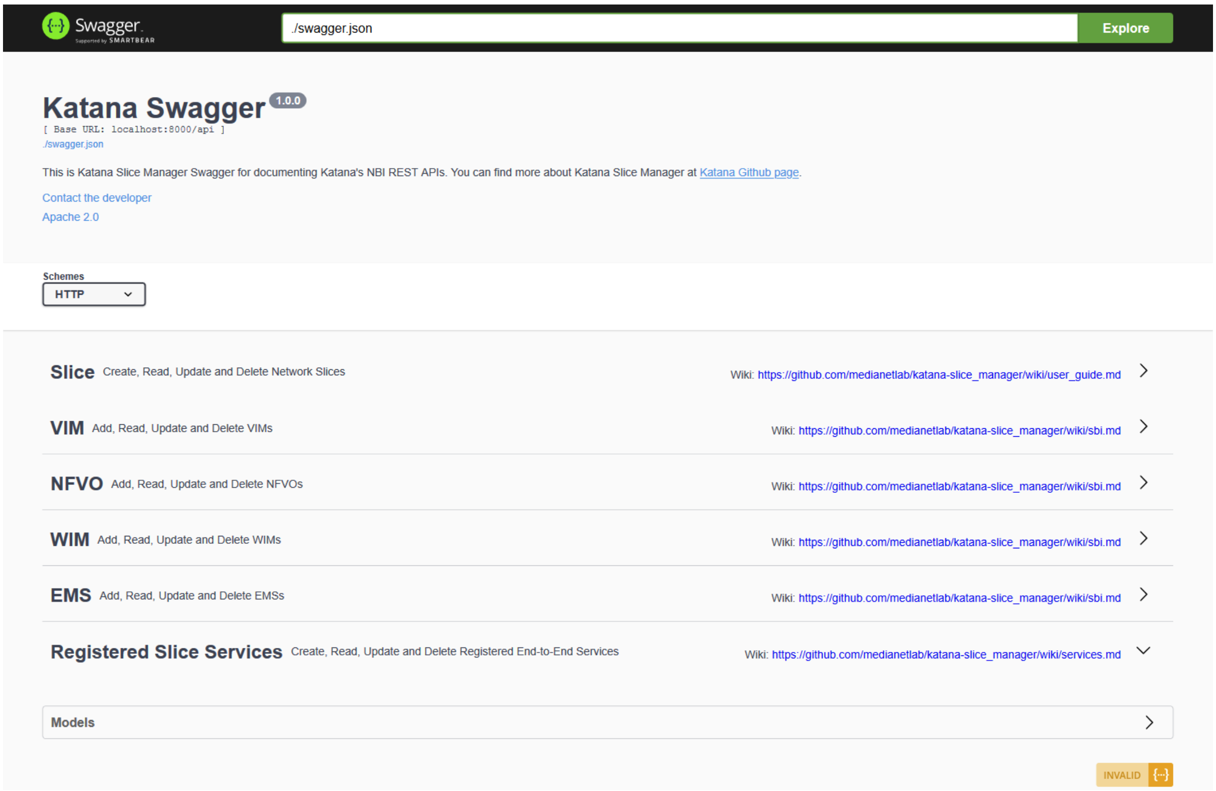
	max	current	total
Iperf Agent Server	0	0	0
Iperf Agent Client	0	0	0

Packet Loss (%)

	min	max	avg	current	total
Iperf Agent Server	0	0	0	0	0
Iperf Agent Client	0	0	0	0	0

2019-12-18 15:53:17.351
Iperf Agent Server: 0
Iperf Agent Client: 0

Throughput visualization in Grafana



Katana Swagger 1.0.0

[Base URL: localhost:8000/api]

./swagger.json

This is Katana Slice Manager Swagger for documenting Katana's NBI REST APIs. You can find more about Katana Slice Manager at Katana Github page.

Contact the developer

Apache 2.0

Schemes: HTTP

Slice Create, Read, Update and Delete Network Slices — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/user_guide.md

VIM Add, Read, Update and Delete VIMs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi.md

NFVO Add, Read, Update and Delete NFVOs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi.md

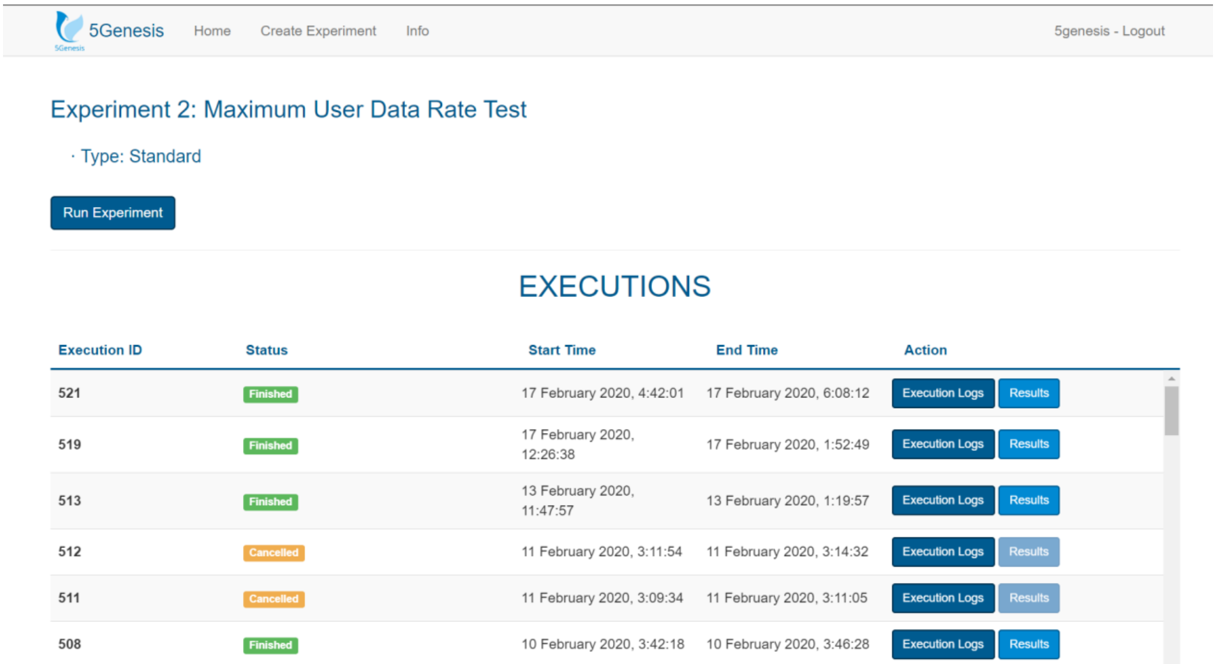
WIM Add, Read, Update and Delete WIMs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi.md

EMS Add, Read, Update and Delete EMSs — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/sbi.md

Registered Slice Services Create, Read, Update and Delete Registered End-to-End Services — Wiki: https://github.com/medianetlab/katana-slice_manager/wiki/services.md

Slice Manager Swagger API

Malaga platform integration activities (screenshots):



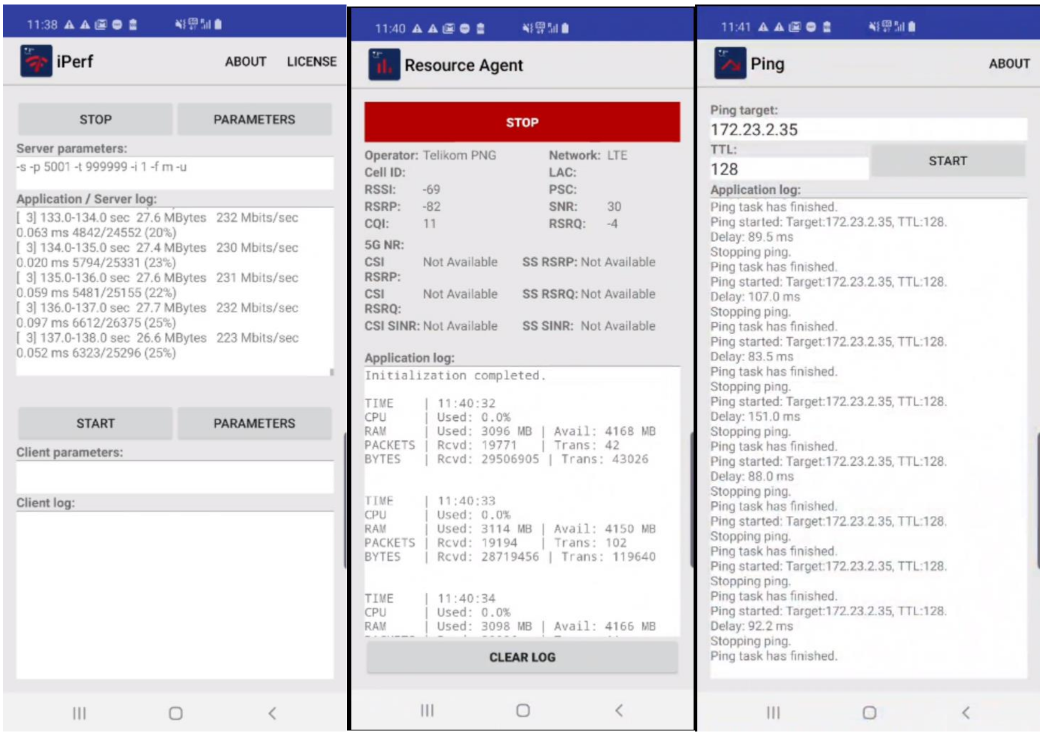
Experiment 2: Maximum User Data Rate Test

- Type: Standard

EXECUTIONS

Execution ID	Status	Start Time	End Time	Action
521	Finished	17 February 2020, 4:42:01	17 February 2020, 6:08:12	Execution Logs Results
519	Finished	17 February 2020, 12:26:38	17 February 2020, 1:52:49	Execution Logs Results
513	Finished	13 February 2020, 11:47:57	13 February 2020, 1:19:57	Execution Logs Results
512	Cancelled	11 February 2020, 3:11:54	11 February 2020, 3:14:32	Execution Logs Results
511	Cancelled	11 February 2020, 3:09:34	11 February 2020, 3:11:05	Execution Logs Results
508	Finished	10 February 2020, 3:42:18	10 February 2020, 3:46:28	Execution Logs Results

Portal integration in Malaga platform



Server parameters:
-s -p 5001 -t 999999 -i 1 -f m -u

Application / Server log:

```
[  3] 133.0-134.0 sec  27.6 MBytes   232 Mbits/sec
0.063 ms 4842/24552 (20%)
[  3] 134.0-135.0 sec  27.4 MBytes   230 Mbits/sec
0.020 ms 5794/25331 (23%)
[  3] 135.0-136.0 sec  27.6 MBytes   231 Mbits/sec
0.059 ms 5481/25155 (22%)
[  3] 136.0-137.0 sec  27.7 MBytes   232 Mbits/sec
0.097 ms 6612/26375 (25%)
[  3] 137.0-138.0 sec  26.6 MBytes   223 Mbits/sec
0.052 ms 6323/25296 (25%)
```

Client parameters:

Client log:

Operator: Telikom PNG   Network: LTE
Cell ID: LAC:
RSSI: -69   PSC:
RSRP: -82   SNR: 30
CQI: 11   RSRQ: -4

5G NR:
CSI RSRP: Not Available   SS RSRP: Not Available
CSI RSRQ: Not Available   SS RSRQ: Not Available
CSI SINR: Not Available   SS SINR: Not Available

Application log:

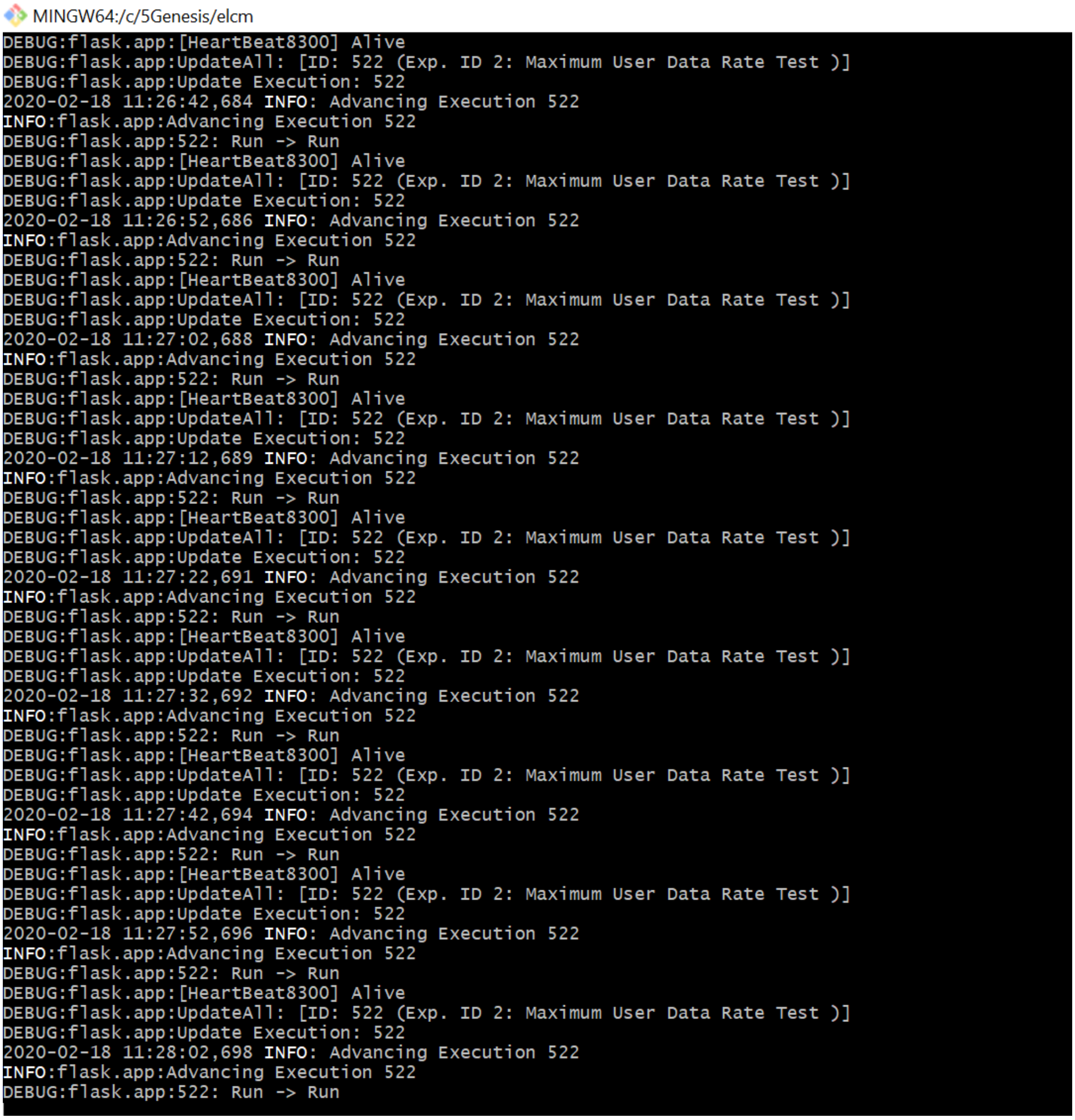
```
Initialization completed.

TIME    | 11:40:32
CPU     | Used: 0.0%
RAM     | Used: 3096 MB | Avail: 4168 MB
PACKETS | Rcvd: 19771    | Trans: 42
BYTES   | Rcvd: 29506905 | Trans: 43026

TIME    | 11:40:33
CPU     | Used: 0.0%
RAM     | Used: 3114 MB | Avail: 4150 MB
PACKETS | Rcvd: 19194    | Trans: 102
BYTES   | Rcvd: 28719456 | Trans: 119640

TIME    | 11:40:34
CPU     | Used: 0.0%
RAM     | Used: 3098 MB | Avail: 4166 MB
```

Ping target:
172.23.2.35
TTL:
128

Application log:

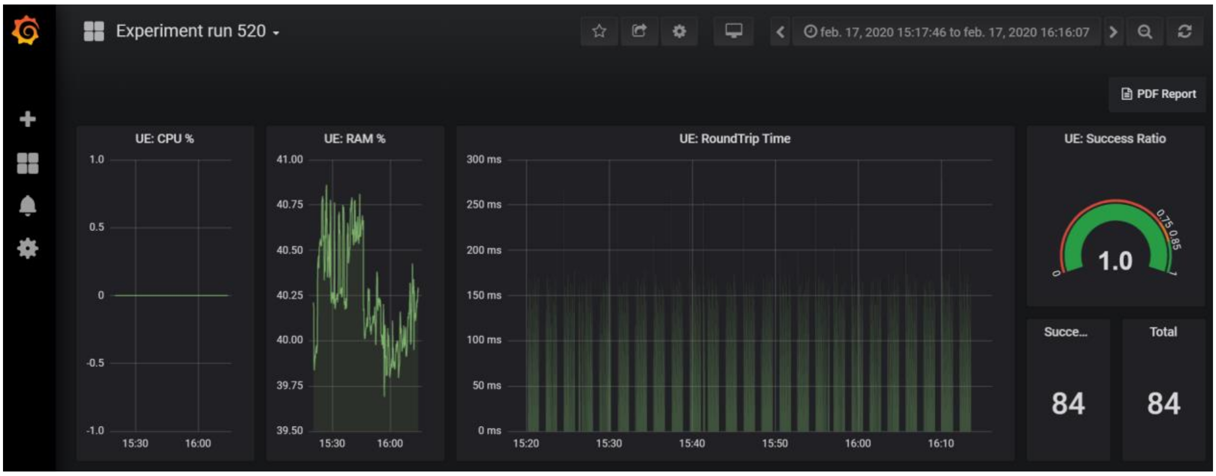
```
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 89.5 ms
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 107.0 ms
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 83.5 ms
Ping task has finished.
Stopping ping.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 151.0 ms
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 88.0 ms
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Stopping ping.
Ping task has finished.
Ping started: Target:172.23.2.35, TTL:128.
Delay: 92.2 ms
Stopping ping.
Ping task has finished.
```

iPerf, Ping and Resources agents integrated into the Malaga platform

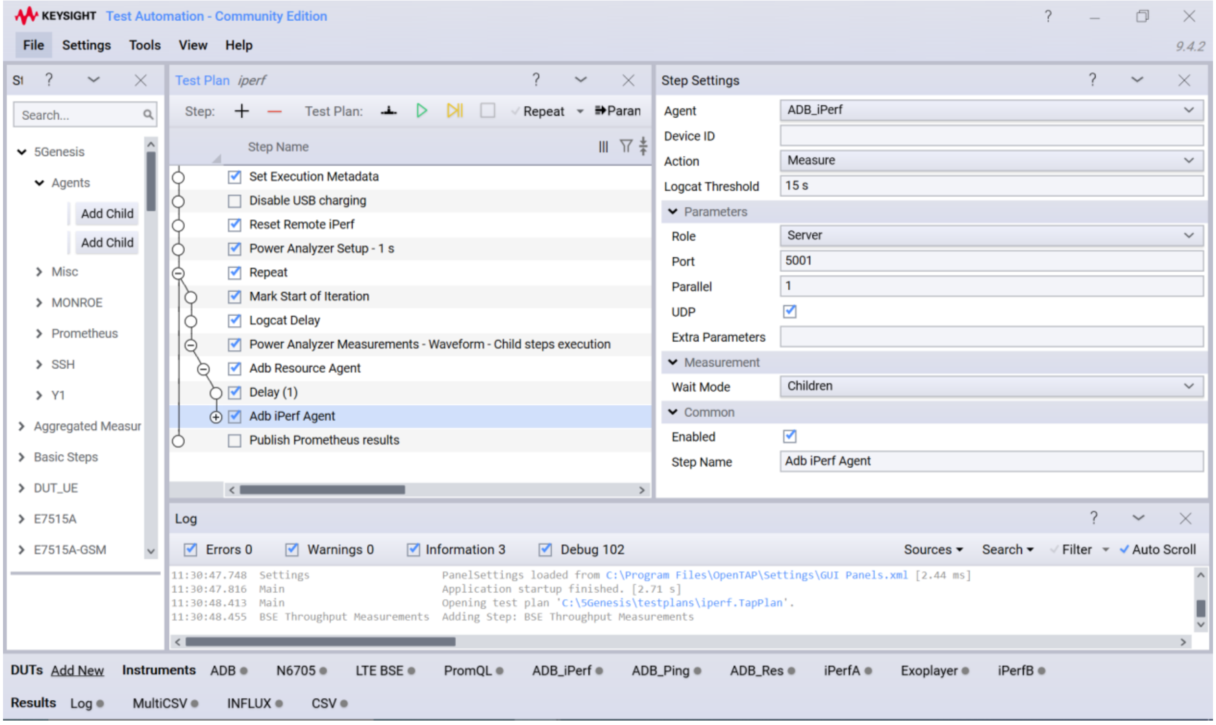


```
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:26:42,684 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:26:52,686 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:02,688 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:12,689 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:22,691 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:32,692 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:42,694 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:27:52,696 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
DEBUG:flask.app:[HeartBeat8300] Alive
DEBUG:flask.app:UpdateAll: [ID: 522 (Exp. ID 2: Maximum User Data Rate Test )]
DEBUG:flask.app:Update Execution: 522
2020-02-18 11:28:02,698 INFO: Advancing Execution 522
INFO:flask.app:Advancing Execution 522
DEBUG:flask.app:522: Run -> Run
```

ECLM integration in Malaga platform



Grafana integration in Malaga platform



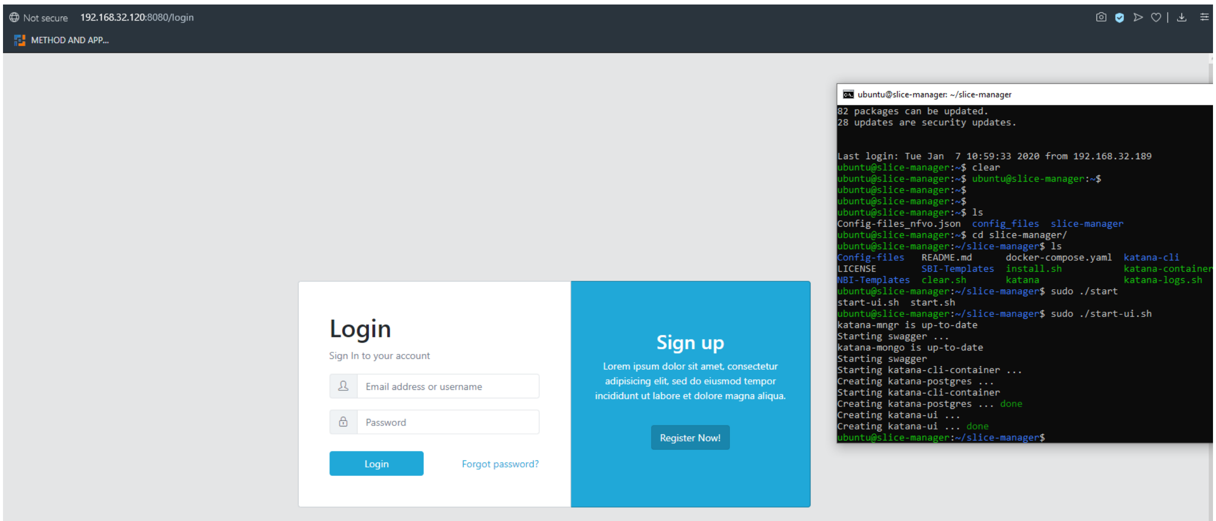
- ☑ Set Execution Metadata
- □ Disable USB charging
- ☑ Reset Remote iPerf
- ☑ Power Analyzer Setup - 1 s
- ☑ Repeat
- ☑ Mark Start of Iteration
- ☑ Logcat Delay
- ☑ Power Analyzer Measurements - Waveform - Child steps execution
- ☑ Adb Resource Agent
- ☑ Delay (1)
- ☑ Adb iPerf Agent
- □ Publish Prometheus results

Agent	ADB_iPerf
Device ID	
Action	Measure
Logcat Threshold	15 s
Parameters	
Role	Server
Port	5001
Parallel	1
UDP	☑
Extra Parameters	
Measurement	
Wait Mode	Children
Common	
Enabled	☑
Step Name	Adb iPerf Agent

☑ Errors 0 ☑ Warnings 0 ☑ Information 3 ☑ Debug 102

```
11:30:47.748  Settings                    PanelSettings loaded from C:\Program Files\OpenTAP\Settings\GUI Panels.xml [2.44 ms]
11:30:47.816  Main                        Application startup finished. [2.71 s]
11:30:48.413  Main                        Opening test plan 'C:\5Genesis\testplans\iperf.TapPlan'.
11:30:48.455  BSE Throughput Measurements Adding Step: BSE Throughput Measurements
```

OpenTAP integration in Malaga platform



Login

Sign In to your account

Email address or username

Password

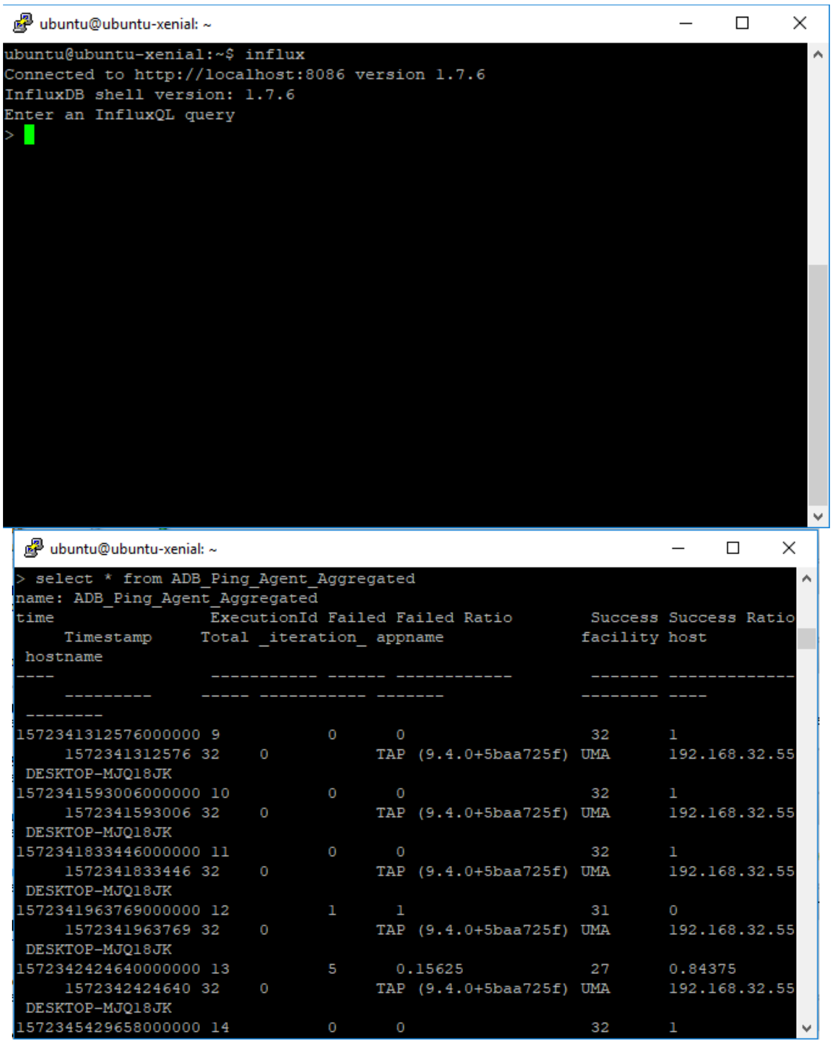
Sign up

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

ubuntu@slice-manager: ~/slice-manager

```
82 packages can be updated.
28 updates are security updates.


Last login: Tue Jan  7 10:59:33 2020 from 192.168.32.189
ubuntu@slice-manager:~$ clear
ubuntu@slice-manager:~$ ubuntu@slice-manager:~$
ubuntu@slice-manager:~$
ubuntu@slice-manager:~$
ubuntu@slice-manager:~$ ls
Config-files_nfvo.json  config_files  slice-manager
ubuntu@slice-manager:~$ cd slice-manager/
ubuntu@slice-manager:~/slice-manager$ ls
Config-files   README.md      docker-compose.yaml  katana-cli
LICENSE        SBI-Templates  install.sh           katana-container
NBI-Templates  clear.sh       katana               katana-logs.sh
ubuntu@slice-manager:~/slice-manager$ sudo ./start
start-ui.sh  start.sh
ubuntu@slice-manager:~/slice-manager$ sudo ./start-ui.sh
katana-mngr is up-to-date
Starting swagger ...
katana-mongo is up-to-date
Starting swagger
Starting katana-cli-container ...
Creating katana-postgres ...
Starting katana-cli-container
Creating katana-postgres ... done
Creating katana-ui ...
Creating katana-ui ... done
ubuntu@slice-manager:~/slice-manager$
```

Slice Manager integration in Malaga platform



```
ubuntu@ubuntu-xenial:~$ influx
Connected to http://localhost:8086 version 1.7.6
InfluxDB shell version: 1.7.6
Enter an InfluxQL query
>
```

```
> select * from ADB_Ping_Agent_Aggregated
name: ADB_Ping_Agent_Aggregated
time                ExecutionId Failed Failed Ratio        Success Success Ratio
     Timestamp     Total _iteration_ appname               facility host
 hostname
----                ----------- ------ ------------        ------- -------------
     ---------     ----- ----------- -------               -------- ----
 --------
1572341312576000000 9           0      0                   32      1
     1572341312576 32    0           TAP (9.4.0+5baa725f)  UMA      192.168.32.55
 DESKTOP-MJQ18JK
1572341593006000000 10          0      0                   32      1
     1572341593006 32    0           TAP (9.4.0+5baa725f)  UMA      192.168.32.55
 DESKTOP-MJQ18JK
1572341833446000000 11          0      0                   32      1
     1572341833446 32    0           TAP (9.4.0+5baa725f)  UMA      192.168.32.55
 DESKTOP-MJQ18JK
1572341963769000000 12          1      1                   31      0
     1572341963769 32    0           TAP (9.4.0+5baa725f)  UMA      192.168.32.55
 DESKTOP-MJQ18JK
1572342424640000000 13          5      0.15625             27      0.84375
     1572342424640 32    0           TAP (9.4.0+5baa725f)  UMA      192.168.32.55
 DESKTOP-MJQ18JK
1572345429658000000 14          0      0                   32      1
```

InfluxDB integration in Malaga platform