

LoadCore Deployment

Deployment Guide

Notices

Copyright Notice

© Keysight Technologies 2020–2023

No part of this document may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly,

Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of those rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contacting Us

Keysight headquarters

1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738
www.ixiacom.com/contact/info

Support

Global Support	+1 818 595 2599	support@ixiacom.com
<i>Regional and local support contacts:</i>		
APAC Support	+91 80 4939 6410	support@ixiacom.com
Australia	+61-742434942	support@ixiacom.com
EMEA Support	+40 21 301 5699	support-emea@ixiacom.com
Greater China Region	+400 898 0598	support-china@ixiacom.com
Hong Kong	+852-30084465	support@ixiacom.com
India Office	+91 80 4939 6410	support-india@ixiacom.com
Japan Head Office	+81 3 5326 1980	support-japan@ixiacom.com
Korea Office	+82 2 3461 0095	support-korea@ixiacom.com
Singapore Office	+65-6215-7700	support@ixiacom.com
Taiwan (local toll-free number)	00801856991	support@ixiacom.com

Documentation conventions

The following documentation conventions are used in this guide:

Describing interactions with the UI

You can interact with products by using different input methods: keyboard, mouse, touch, and more. So in most parts of the user documentation, generic verbs have been used that work with any input method. In cases where input-neutral verbs do not work, mouse-specific verbs are used as the first choice, followed by touch-specific verbs as the second choice.

See the following table for examples on how you can interpret the different input methods.

Input-neutral	Mouse	Touch
Select Modify .	Click Modify .	Tap Modify .
Select Accounts > Other accounts > Add an account .	Click Accounts > Other accounts > Add an account .	Tap Accounts > Other accounts > Add an account .
To open the document in Outline view, select View > Outline .	To open the document in Outline view, click View > Outline .	To open the document in Outline view, tap View > Outline .
Select Protocols .	Click the Protocols tab.	Tap Protocols .
-NA-	Double-click the Client wizard.	Double-tap the Client wizard.
Open the Packages context menu.	Right-click Packages to open the shortcut menu.	Long tap Packages to open the shortcut menu.

Deprecated words

The following words have been replaced with new words, considering the audience profile, our modern approach to voice and style, and our emphasis to use input-neutral terms that support all input methods.

Old usage...	New usage...
shortcut menu, right-click menu	context menu
click, right-click	select
drag and drop	drag

Table of Contents

Contacting Us	3
Documentation conventions	4
Chapter 1 Test Methodologies for 5GC	1
Chapter 2 LoadCore Deployment on VMs	2
VMware ESXi Deployment	2
Prerequisites	2
LoadCore Installation Procedure	4
Amazon AWS Deployment	28
Prerequisites	28
AWS services and components	29
LoadCore Components Installation	37
AWS Security Best Practices	64
Azure Cloud Deployment	65
Appendix A Enable PCI Passthrough for KVM deployments	66
Appendix B Deploying Loadcore with static IP Addresses	69
Appendix C Ports used in LoadCore communication	73
Chapter 3 LoadCore Deployment on Containers	74
Amazon AWS EKS Deployment	74
OpenShift Deployment	77
LoadCore Deployment on k8s	79
LoadCore Deployment on GKE Anthos	80
Chapter 4 Test Case #1: gNB and 5G Core simulation in B2B scenario	82
Test Configuration	82
Import a test configuration	100
Run a test	102
Results Analysis	104

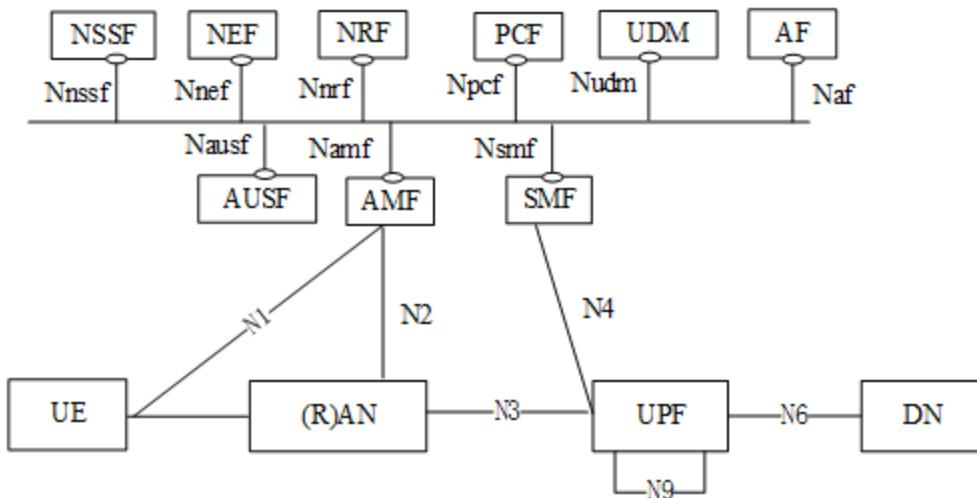
Downloading results, logs and captures	106
Generate and retrieve csv statistics using Rest API	107
Generate and retrieve packet captures	110
Running Application traffic (HTTP)	114
Installing additional interfaces on the test agents	115
Enabling IxStack over DPDK/Raw	116
Configure HugePages on both Agents	117
Application traffic configuration	118
Create Custom Dashboard	120
Log Collection	123
Debug commands	124
Index	125

CHAPTER 1**Test Methodologies for 5GC**

The 5G will lead in a large number of new applications – for use cases that are not even close to being possible in the 3G/4G regime. According to reasonable estimates, the world will have more than 1.1 billion 5G connections by 2025 (accounting for ~15% of the total connections). Along with artificial intelligence (AI) and edge computing, 5G wireless technology will be right at the heart of the burgeoning IoT revolution over the next half a decade or so, expected to play a major role in the development of Industry 4.0 in general, smart city applications, smart industrial software, powering connected cars, and smart homes & buildings. Seamless mobility, negligible latency, full scalability, and reliability will help 5G in making many high-end, mission-critical IoT projects implementable with ease.

Unlike 4G, which is determined by modulation and frequency (i.e., interface-defined), 5G will be the first-ever software-defined wireless standard. And this is a key aspect of the Control User Plane separation of SMF/UPF where LTE CUPS is being reused.

The 5G Core is represented by following network topology:



CHAPTER 2**LoadCore Deployment on VMs**

This section contains the following topics:

VMware ESXi Deployment	2
Amazon AWS Deployment	28
Azure Cloud Deployment	65
Appendix A Enable PCI Passthrough for KVM deployments	66
Appendix B Deploying Loadcore with static IP Addresses	69
Appendix C Ports used in LoadCore communication	73

VMware ESXi Deployment

This section describes the steps needed to deploy LoadCore on VMware ESXi.

Prerequisites

For a complete and correct functioning setup, make sure you have downloaded and installed the packages on your test environment:

- **Wireshark** capable to decode PFCP messages and IEs (at least 2.5.2).
This will be used to analyze the traffic captures.
- **Hypervisor** with available resources to deploy the virtual machine(s) hosting the Middleware, the Test Agents and License Server.
The examples below were done using VMware ESXi.
- **LoadCore images** (two images: one for Middleware and one for Agents).
The images can be downloaded from Ixia support website
<https://support.ixiacom.com/software-downloads/236052>.
- **Ixia License server**. This is the Linux based License Server that can be installed either on the same machine with the Hypervisor or intranet.
The license server installed should be downloaded from Ixia support website
<https://support.ixiacom.com/software-downloads/236052>.

IMPORTANT IP connectivity between VMs and License server needs to be ensured.

- You will need to use this machine to run LoadCore tests. **ILU (Windows or Linux) cannot be used to run LoadCore tests.**
 - **Valid licenses** for Ixia License server. For deploying full 5G core and run Control plane and User plane traffic, the following licenses will be required:
 - Control plane licenses:
 - LoadCore interface license simulation (**P89033A** x 11 pcs).
- This license will enable Control plane testing. One license is needed for each simulated interface.

- 5GCore Performance enabler on VM: 1M UEs and 10k TPS for VM (**P89034A** x 1 pcs).
- User plane licenses:
 - User Plane Flow-based license (N3 and N6). Multiple QTY are needed if multiple flows are active simultaneously. Includes three Application Traffic Flows (TCP/UDP) and 10Gbps throughput capacity (**P89037A** x 2 pcs). The recommendation is to use this type of license as this will be suggested for all new customers.
 - As an alternative for the above license you can also use Tier-4 license (**P89030A** or **P89031A** x 2 pcs).

IMPORTANT These license types will enable User plane traffic and are not needed if ONLY control plane traffic is needed for future tests.

- Minimum supported resolution for LoadCore UI is 1920x1080 (Full HD).

Hardware Requirements

By default, the VM for Middleware will reserve the following compute resources when installing the OVA:

- 8 x vCPUs
- 20 GB RAM
- 256 GB HDD (in LoadCore 1.3 the HDD space was increased to 256 GB since the test results are now stored on Middleware and additional space was needed for this)

The test agent will reserve the following compute resources when installing the OVA:

- 4 x vCPUs
- 4 GB RAM **out of which 1GB is reserved for HUGE MEM(IxStack over DPDK/Raw)**

IMPORTANT This value is for control plane only. If you are running app traffic the recommendation is to allocate minimum 16 GB RAM.
- 20 GB HDD

Running application traffic will require increasing agent resources. The recommendation for each agent will be in this case:

- 8 x vCPUs
- 32 GB RAM
- 20 GB HDD

Licensing server default resources

- 2 x vCPUs
- 4 GB RAM
- 256 GB HDD

Supported NICs in tests with IxStack over DPDK/Raw(TCP based traffic)

The list of supported NICs in tests with IxStack over DPDK/Raw(TCP based traffic):

- Intel 10G, 25G, 40G - **ixgbe driver**
- Mellanox ConnectX 4/5 25G, 100G - **mlx4_core/mlx5_core driver**

- vmxnet3 – **vmxnet3 driver** (built-in ESXi hypervisor)

Supported drivers list:

- vmxnet3
- mlx4_core
- mlx5_core
- ixgbe
- ixgbefv
- i40e
- i40evf
- ena
- Virtio

Supported list of hypervisors

The list of supported hypervisors:

- KVM/Openstack – **IxStack over DPDK/Raw is supported with PCI NICs(PCI passthrough)**
 - Intel 10G & 40G – **ixgbe driver**
 - Mellanox ConnectX 4/5 100G - **mlx4_core/mlx5_core driver**
- IxStack over DPDK/Raw is also supported with SR-IOV passthrough, in which case **ixgbefv** and **i40evf** drivers are used.
- ESXi
 - Intel 10G & 40G – **ixgbe driver**
 - Mellanox ConnectX 4/5 100G - **mlx4_core/mlx5_core driver**
 - vmxnet3 – **vmxnet3 driver** (built-in ESXi hypervisor)
- OpenShift 4.5.6
 - a. Intel 10G & 40G – ixgbe driver
 - b. Mellanox ConnectX 4/5 100G - mlx4_core/ mlx5_core driver

LoadCore Installation Procedure

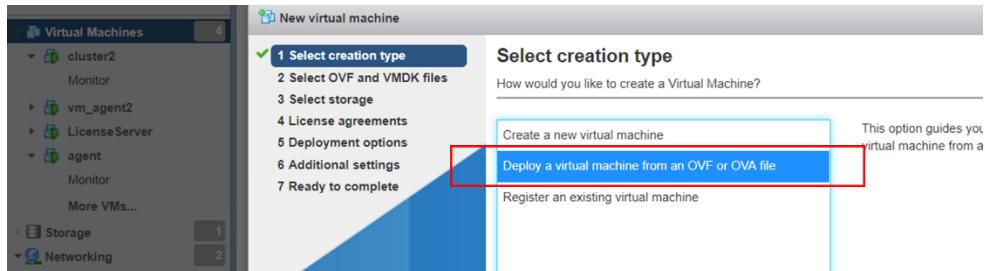
The LoadCore installation procedure requires the following steps:

1. LoadCore Middleware installation and configuration.
2. Migrating users and test configs from LoadCore 2.0 to LoadCore 2.1.
3. Agent(s) installation and configuration.
4. License Server installation and configuration.

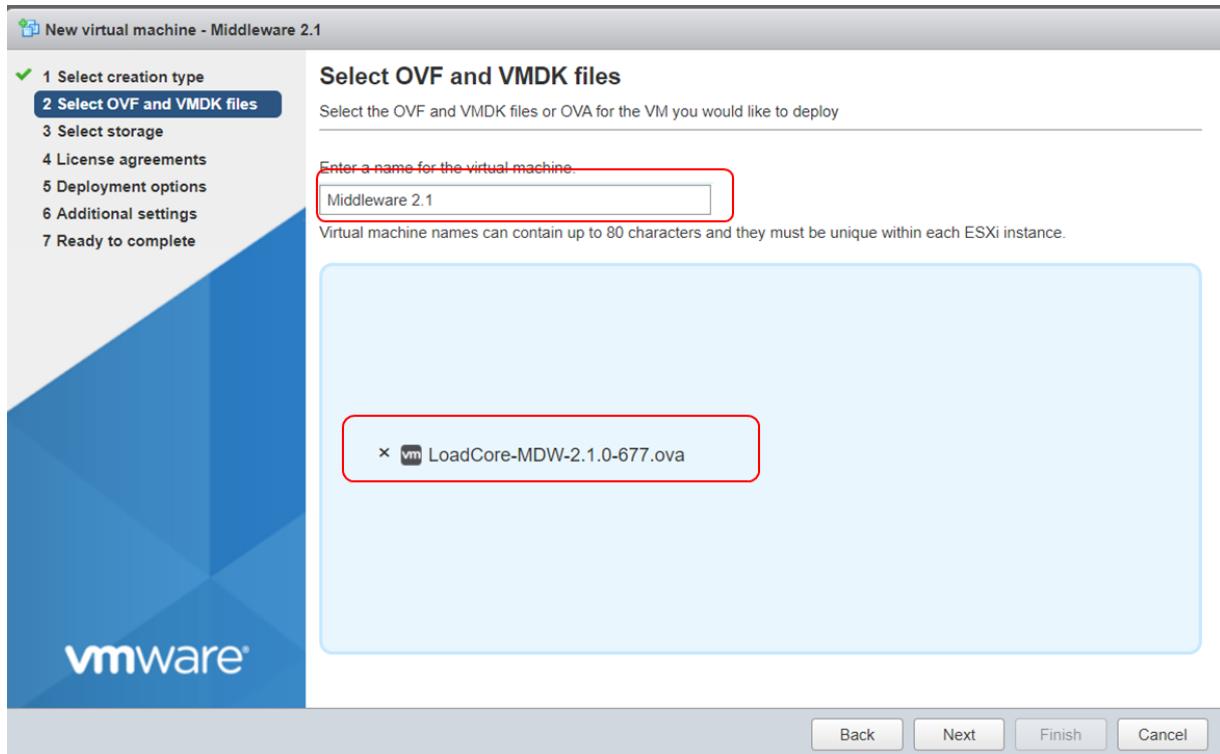
Middleware Installation

The installation procedure of the Middleware requires the following steps:

1. Log into your ESXi server and create a new VM machine using the Middleware OVA image.

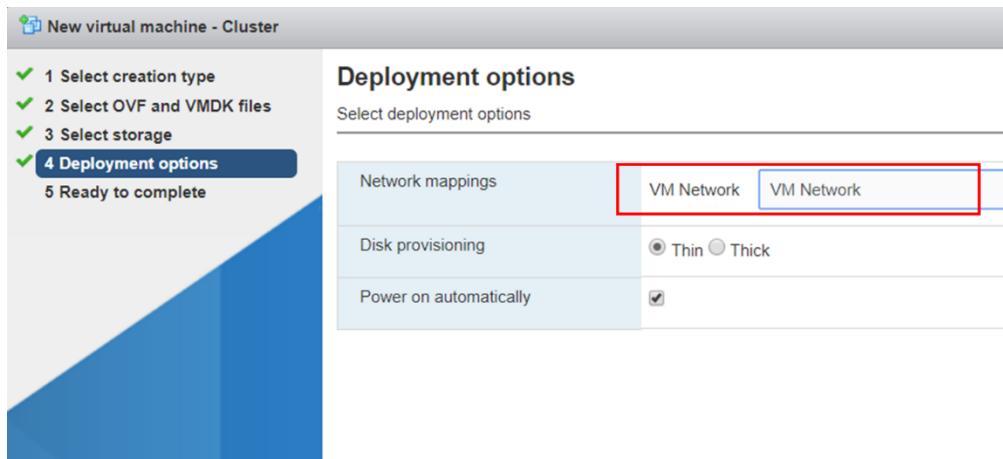


2. Define a name for the VM machine. In this example, the name is **Middleware 2.1**.
3. Select the OVA image and select **Next > Next**.

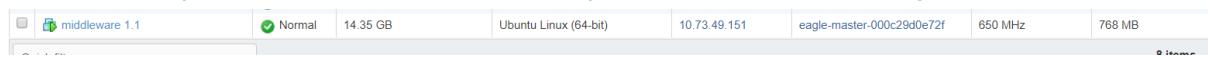


4. Select the network (vSwitch) where the Middleware should connect. By default, this is the **VM Network**.

You can select a different vSwitch/Network that you already created, just make sure that later on, there will be connectivity between this VM and the ones that will represent the Agents.



5. Select **Next** and, then, **Finish**.
6. Wait for complete OVA deployment. When this is completed, you will see in the VM list on the ESXI console, your cluster VM, booted normally and with IP address assigned.



Thin provisioning method is also used in KVM setups. The disk size inside the Loadcore Cluster VM is 250GB, but the `qcow2` image can be much smaller. The `qcow2` image is getting bigger with time according to demand.

For example ...

Linux host machine:

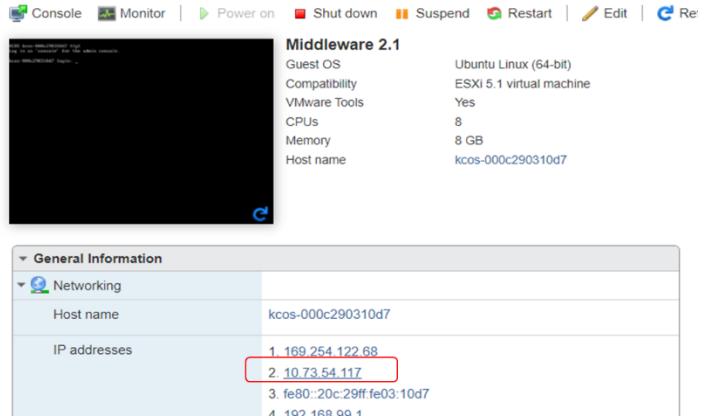
```
-rw-rw-r-- 1 root root 17G Sep 24 12:37 LoadCore-MW-w1.3.0-981.qcow2
```

Inside Loadcore Cluster VM:

```
appsec@eagle-master-525400e6e71e:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G  0% /dev
tmpfs           798M  6.0M  792M  1% /run
/dev/mapper/main-root  251G  14G  224G  6% /
tmpfs           3.9G   0    3.9G  0% /dev/shm
tmpfs           5.0M   0    5.0M  0% /run/lock
tmpfs           3.9G   0    3.9G  0% /sys/fs/cgroup
/dev/vda3        465M  61M  377M  14% /boot
/dev/vda2        241M  512  241M  1% /boot/efi
tmpfs           798M   0    798M  0% /run/user/1000
```

After the Middleware image is deployed, the next step is to access the LoadCore UI.

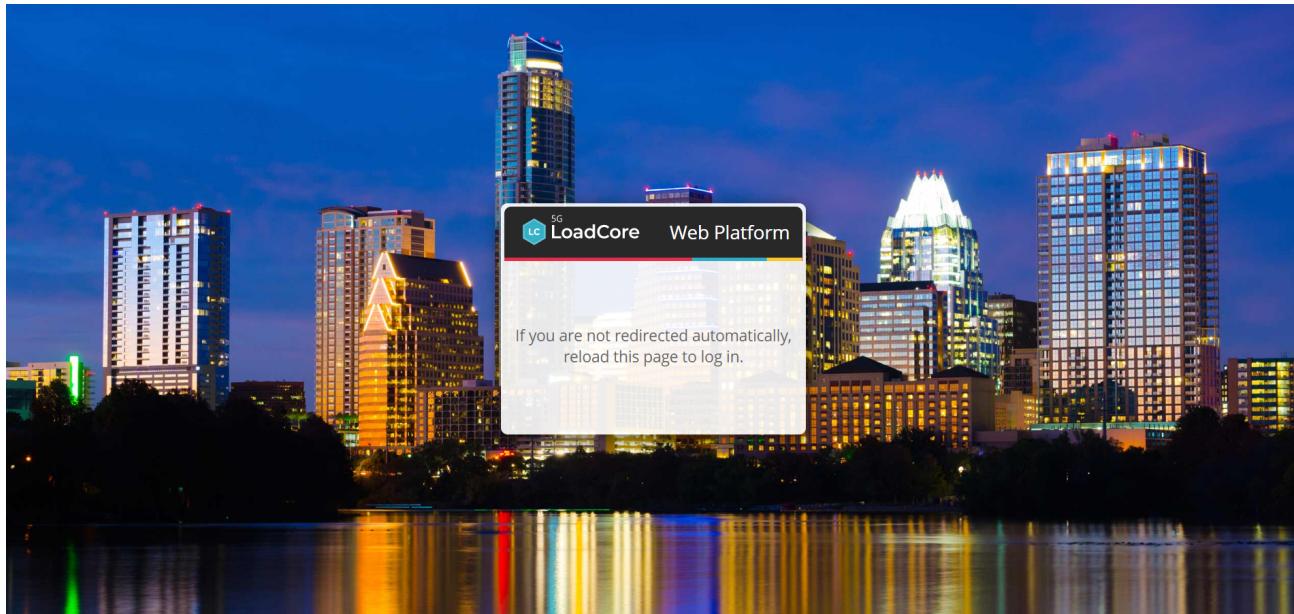
To do this, use the IP address that was assigned to the Middleware VM machine.



IMPORTANT Starting with LoadCore 2.1 there is no need to run any additional installer, everything is embedded in the OVA image.

Connect to the LoadCore UI to make sure that everything is working fine.

On your browser, type the IP address that was assigned for the Cluster - in this case <https://10.73.53.201>.



After accepting the EULA, you will be automatically routed to the authentication window.

There are two options:

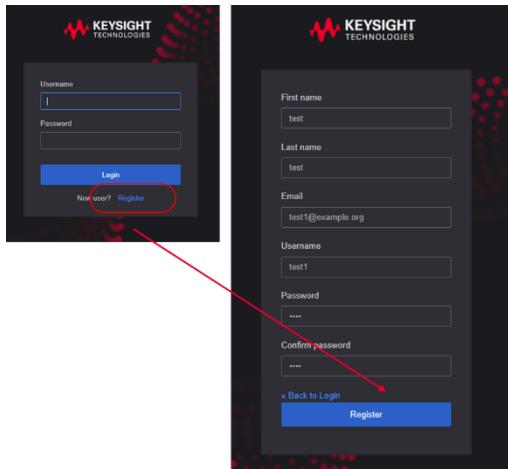
1. Create a [new user](#) using the **Register** link.
2. Use the default [administrator](#) account : **admin/admin**.

The administrator account will have access to the Access Control menu which includes additional configuration options, including user editing (for example, changing passwords).

Register a new user

To register a new user, do the following:

1. Select the **Register** link from the authentication window and fill in all the necessary info.



2. To create the user, select **Register**. You will be logged and redirected to the main LoadCore screen.

If needed, you can, you can click on the username(in this case **user1**) and the user properties window will be open, allowing you to change the your settings.

Edit Account	
Username	test1
Email *	test1@example.org
First name *	test
Last name *	test

Logging in with administrative rights

To log in with administrative rights use the following credentials: **admin/admin**.

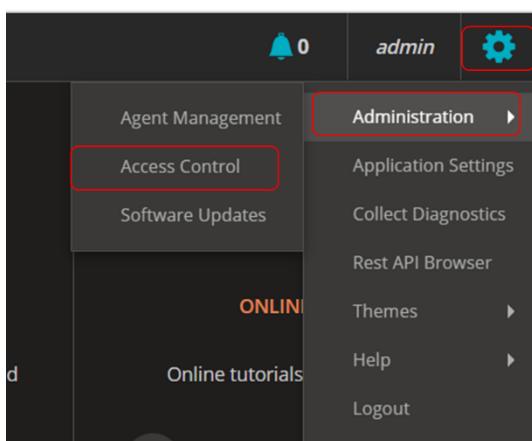
This will log you to the LoadCore system.

The screenshot shows the LoadCore dashboard interface. It features four main cards:

- CREATE NEW TEST**: A card with a hexagonal icon containing a checklist. Below it says "Choose from the test types below to start building a test from a default configuration." It lists five test types:
 - Wireless Full Core
 - Wireless SBA
 - Wireless NG-RAN Simulation
 - Wireless UPF Isolation
 - Wireless iRAT
- BROWSE CONFIGS**: A card with a circular icon containing a gear. Below it says "Access existing test configurations to view details, change settings, or re-run tests." It lists four recent configurations:
 - Full Core Base Config
 - SBA Base Config
 - NG-RAN Simulation Base Config
 - UPF Isolation Base Config
- BROWSE RESULTS**: A card with a bar chart icon. Below it says "Access previous test run results, view detailed reports or export results."
- ONLINE RESOURCES**: A card with a computer monitor icon. Below it says "Online tutorials and library resources" and lists three links:
 - User Guides (.pdf)
 - User Guides (.htm)
 - User Guides (.zip)

Using this admin rights, you can create the users to access the system, as follows:

1. Select the **Settings** icon > **Administration** > **Access Control**.



2. The Access Control window opens. Select the **Users** tab and, then, **Add User**.

The screenshot shows the 'Users' page in the Keysight LoadCore Admin Console. The left sidebar has a 'Users' option highlighted with a red box. The top right corner has an 'Add user' button highlighted with a red box.

3. Define a user with its corresponding details (in this example, the new user is **user1**).

The 'Add user' dialog box shows the following fields filled with 'user1': Username, Email, First Name, and Last Name. The 'User Enabled' switch is set to 'ON'. The 'Save' button at the bottom is highlighted with a red box.

4. To configure a password for the user, select **Credentials** and define the password.

The 'User1' profile page shows the 'Credentials' tab selected. In the 'Set Password' section, the 'Password' field contains '*****' and the 'Temporary' switch is set to 'ON'. The 'Set Password' button is highlighted with a red box.

In case you added multiple users, select **View all users** and select the **Edit** option for the user (in this example, **user1**).

ID	Username	Email	Last Name	First Name	Actions
d02725f8-c5c7-4c9f-ae74...	admin	admin@example.org	Admin	Default	Edit Delete
7313fc7f-db5f-457b-b4e0...	user1	user1@example.org	User1	user1	Edit Delete
5db39e1e-1edc-4d3c-967...	wap-automation				Edit Delete

If no DHCP server is available or you want to manually assign the IP address to be used by Middleware, you can log on the VM console for the deployed machine (for example, the console from ESXi hypervisor).

Log in with *console* credentials.

This will give you access to a predefined menu from where you can configure the networking settings.

Logging in with *console* means log in as console first, no password required and then log in as **admin** with password **admin** (all in small letters).

The IP address can be changed with command:

```
kcos networking ip set
```

Migrating users and test configs from LoadCore 2.0 to LoadCore 2.1

In order to migrate the tests configurations and the users from LoadCore 2.0 to LoadCore 2.1, you need to connect using ssh on LoadCore 2.0 Middleware and run some scripts.

The scripts can be downloaded from the support page:

- <https://support.ixiacom.com/public/software-downloads/236052>

NOTE You have to run this script from a machine where you have *sshpass* and *jq* installed.

You will be provided with the following folders that include your scripts:

- migrate-keycloak-wap
- migrate-wdm-configs
- wireless-data-model

The users in LoadCore 2.1 before migrating will be as below:

ID	Username	Email	Last Name	First Name	Actions
49c82f7a-de32-4455-8ba0-278bd4...	admin	admin@example.org	Admin	Default	Edit Delete
207d08e6-f54d-439d-a3be-5fde0a...	wap-automation				Edit Delete

Migrating procedure:

- Save your source Middleware IP address (from the older version of LoadCore) and your Middleware IP address (in this case the 2.1 LoadCore IP).

Your older version of LoadCore will be the source IP address, while the new version of LoadCore will be the destination IP address.(the source IP address and destination IP address will be used in the next steps).

In this example, the source IP address is 10.38.156.160 and destination IP address is 10.38.154.203.

NOTE

If you want to move all your accounts from 2.0 to 2.1, you can proceed with next step. If you want to move only the tests configs (and define new users), you can skip this step.

- In the `migrate-keycloak-wap` folder, you will find the following scripts:

- `copy-keycloak-wap.sh`
- `export-keycloak-wap-realm.sh`

Run the script `copy-keycloak-wap`. In this case it is:

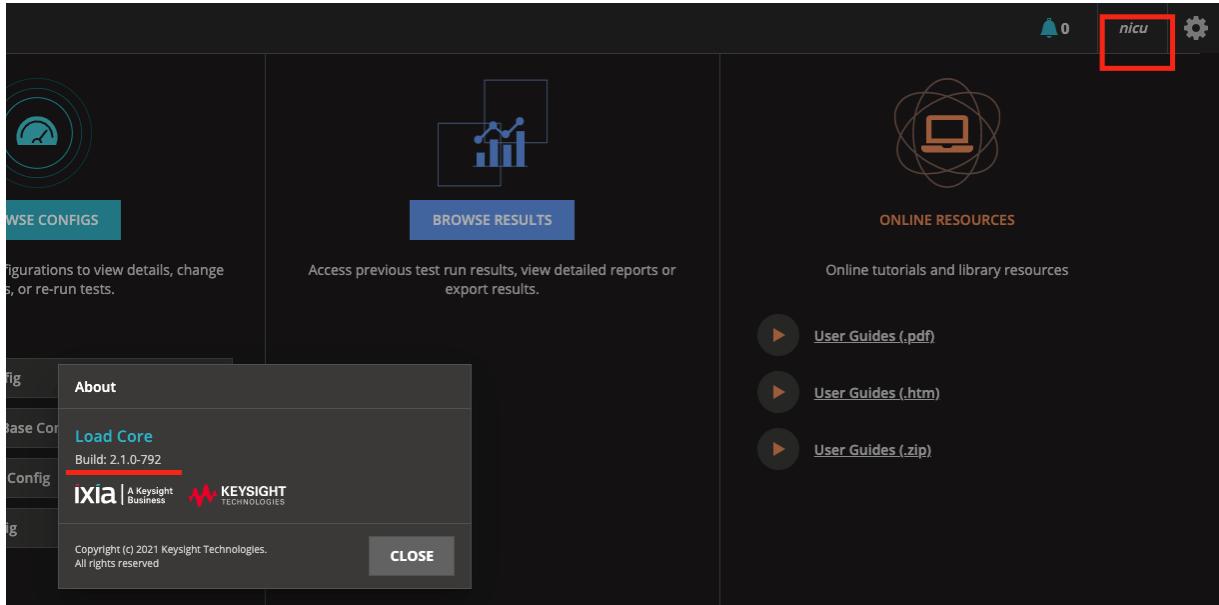
```
./copy-keycloak-wap.sh -s 10.38.156.160 -d 10.38.154.203
```

The result is:

```
nicu@nicu-VirtualBox:/lizard-me-tests$ ./copy-keycloak-wap.sh -s 10.38.156.160 -d 10.38.154.203
Exporting ...
10.38.154.203 INFO [org.keycloak.services] (ServerService Thread Pool -- 62) KC-SERVICES0#035: Export finished successfully
killing export processes
/tmp/export-keycloak-wap-realm.sh: line 23: 1398 Killed
/tmp/export-keycloak-wap-realm.sh: line 32: 1387 Killed
/tmp/export-keycloak-wap-realm.sh: line 32: 1387 Killed
/opt/jboss/keycloak/bin/standalone.sh -Okeycloak.migration.action=export -Dkeycloak.migration.provider=dir -Dkeycloak.migration.usersExportStrategy=SAME_FILE -Dkeycloak.migration.realmMappings=[{"realm": "testrealm", "migration.dir": "JSON_EXPORT_DIR"}] -Djboss.socket.binding.port-offset=99 > ${LOGFILE}
realmName=testrealm
realmId=1
realmMappings=[{"realm": "testrealm", "migration.dir": "JSON_EXPORT_DIR"}]
usersFile=users-0.json
Total  N Received N Xferd  Average Speed   Time Current
Dataload Total  Syncnt  Up  Down
100 1929 100 1868 100 69 12156 466 --:--:--:--:--:-- 12087
(*overwritten*:0,"added":0,"skipped":2,"results":[{"action":"SKIPPED","resourceType":"USER","resourceName":"wap-automation","id":"287d09e6-f54d-439d-a3ba-5fde0a9b8cb7"}, {"action":"SKIPPED","resourceType":"USER","resourceName":"nicu","id":"2bb93f76-1ae4-4ba3-b231-9df3ed08af5a"}])
nicu@nicu-VirtualBox:/lizard-me-tests$
```

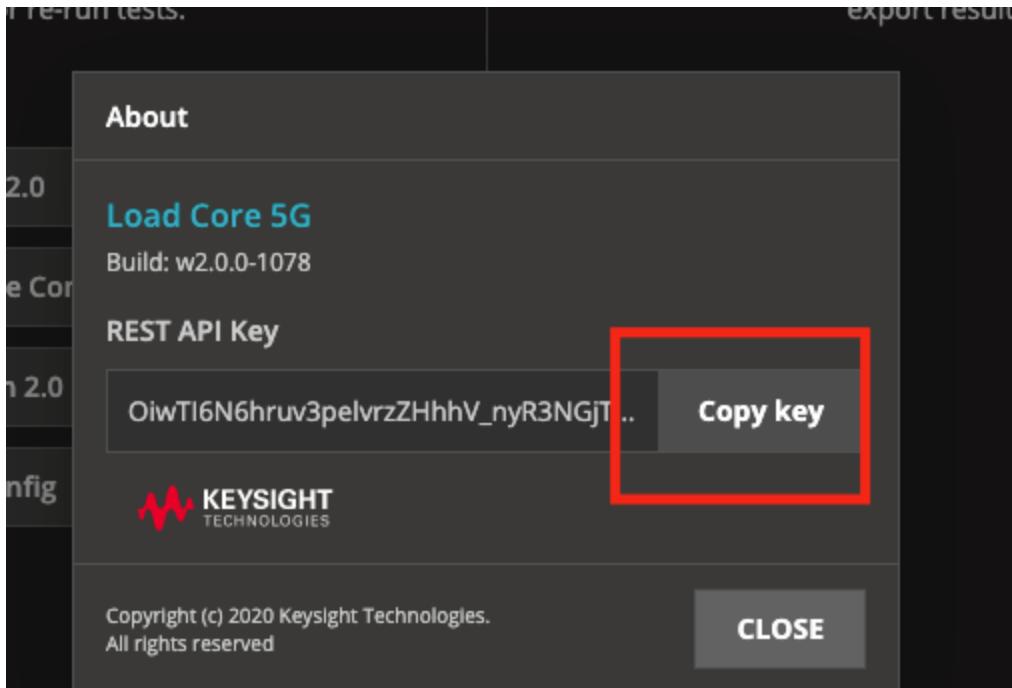
Also, you can see that on Middleware LoadCore 2.1, this user appears here:

ID	Username	Email	Last Name	First Name	Actions
49d827a-de32-4455-8ba0-278bd4...	admin	admin@example.org	Admin	Default	Edit Delete
2bb93f76-1ae4-4ba3-b231-9df3ed08af5a	nicu	nicu@keysight.com	nicu	nicu	Edit Delete
207d09e6-f54d-439d-a3ba-5fde0a9b8cb7	wap-automation				Edit Delete



- If you are moving just the test configurations from LoadCore 2.0 to Loadcore 2.1, open the `migrate-wdm-configs` folder. The following script can be found here:
 - `copy-configs.sh`

Open this script and override the value of `SOURCE_TOKEN` on line 27. The new value of `SOURCE_TOKEN` will be the token present on Loadcore 2.0 at **About**.



NOTE

Keep in mind that the source token is refreshed every time you logout from LoadCore. Also, each user has a different source token.

You can copy all the configs from LoadCore 2.0 to a specific user in LoadCore 2.1 using `-u` and `-p` as extra parameters.

NOTE

You want to copy the configs from LoadCore 2.0 to LoadCore 2.1 but to a specific user. By default, if no extra parameters are set, all the configs will go to the admin user.

For example ...

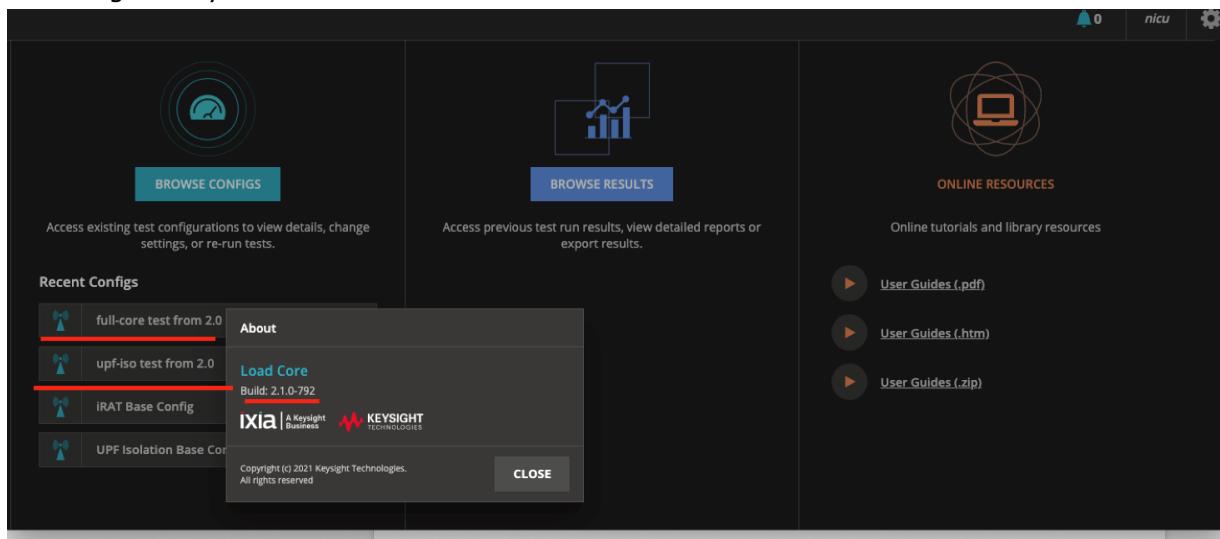
In order to send the configs from `SOURCE_TOKEN` of LoadCore 2.0 to LoadCore 2.1 to user **nicu**:

```
./copy-configs.sh -s 10.38.156.160 -d 10.38.154.203 -u nicu -p nicu
```

where:

- `-s` = source
- `-d` = destination
- `-u` = user_destination
- `-p` = password_user_destination

After migration you should see:



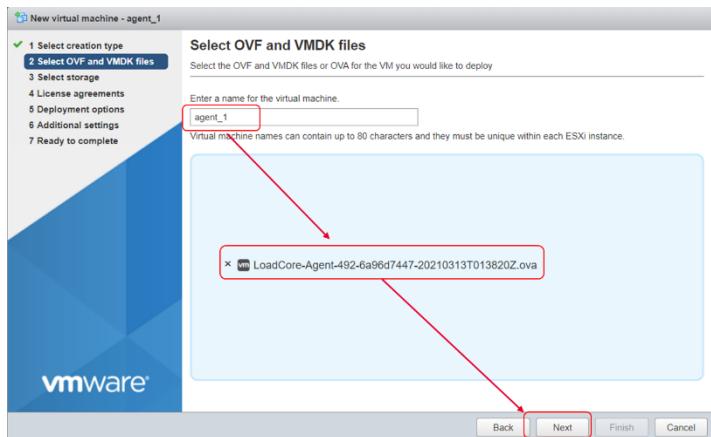
Agent(s) Installation

LoadCore requires at least one Agent to emulate the 5G core nodes. The full 5G core topology can be run within a single agent if stateless UDP traffic will be required. For TCP based traffic you will need to install/use two agents.

For this deployment, two agents will be installed.

The installation procedure is the same as for the Middleware (for more details, refer to [Middleware Installation](#)), with the following differences:

1. Select the agent image when the VM is created.
2. Set a different name for each VM machine (in this deployment **agent_1** is used for the first agent and **agent_2** for the second agent).



3. Accept the End User License Agreement.
4. Connect the agent to the same vSwitch represented by the **VM Network**.
5. Optionally, after the complete VM installation, log on to each agent using the console from ESXi or ssh, and set the agent name:
 - a. Default username/password : **ixia /ixia**

```

Ubuntu 18.04 LTS 5GCTE-0faf4d80 tty1

5GCTE-0faf4d80 login: ixia
Password:
Last login: Sat Jan  4 19:16:41 UTC 2020 on tty1
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ixia@5GCTE-0faf4d80:~$
```

- b. Set the hostname to **agent_1** (use **agent_2** as the name for the second agent).

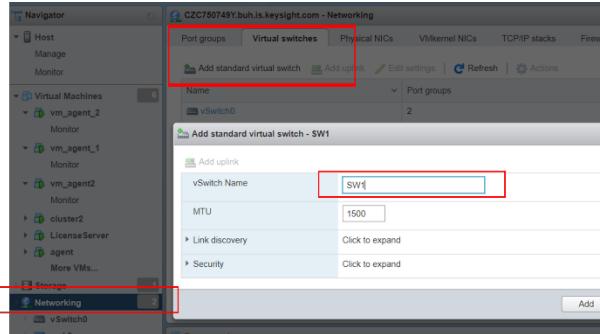
```

ixia@5GCTE-0faf4d80:~$ sudo hostnamectl set-hostname vm_agent_1
[sudo] password for ixia:
ixia@5GCTE-0faf4d80:~$ _
```

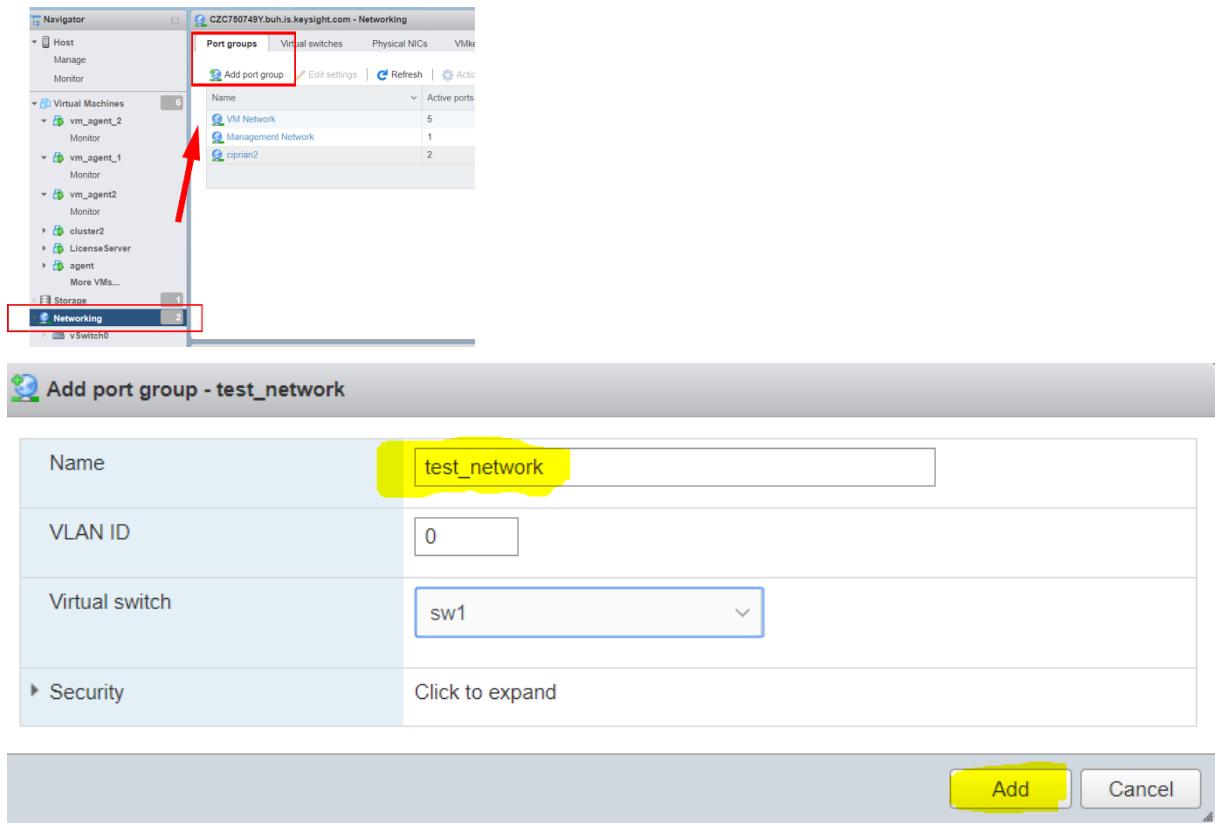
The agents should connect with the Middleware VM on one side and between them on the other side. The second connection represents the **test network**.

In order to do this, another NIC must be added to each VM agent, and connect it to a new vSwitch and PortGroup (**sw1 / test_network**), as follows:

1. Select **Networking > Virtual Switches > Add standard virtual switches** and set the **VSwitch Name** to **sw1**.

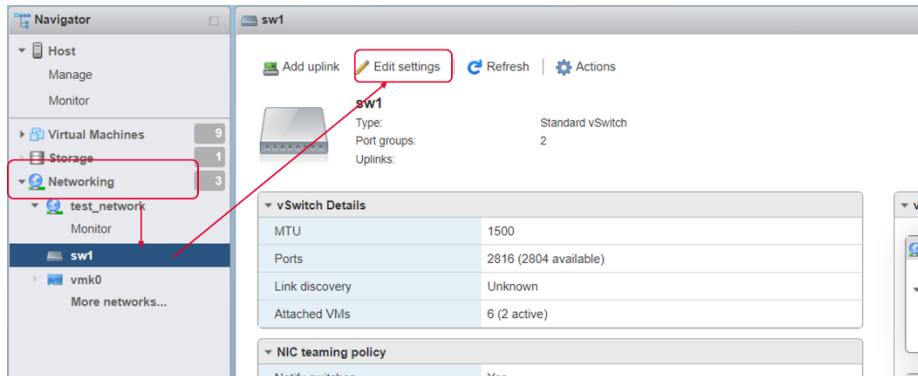


2. Select **Networking > Port Groups > Add port group** and set the port group name to **test_network**.



Since there are multiple IP addresses on the same interface, the **Promiscuous mode** on the vSwitch that is hosting the *test_network group* must be enabled (in the current setup, *sw1* is the vSwitch that has *test_network group*). This can be done as follows:

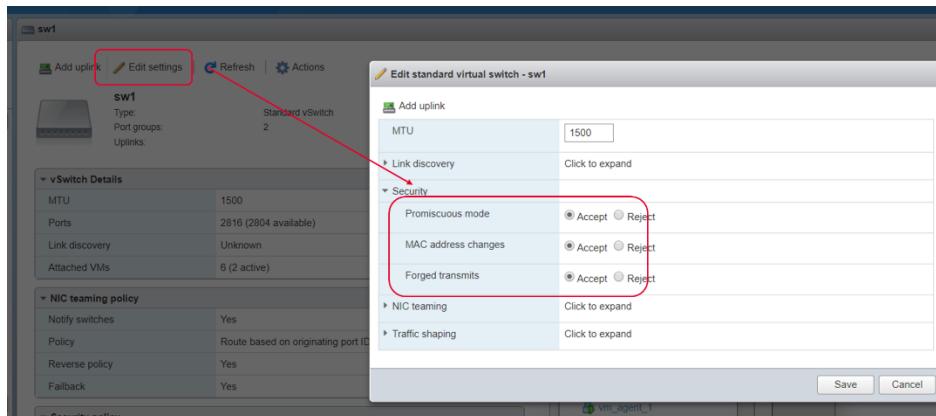
- From the ESXi console, select **Networking > sw1 > Edit settings**.



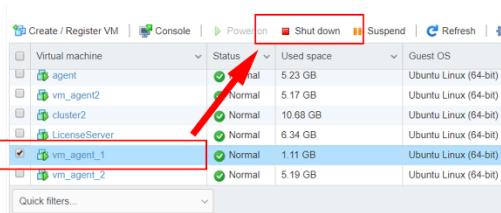
IMPORTANT

In case you are running Openstack on vmWare with VLANs enabled, you need to enable Promiscuous mode on the vswitches (from Reject to Accept) and set VLAN ID 4095 (all vlans).

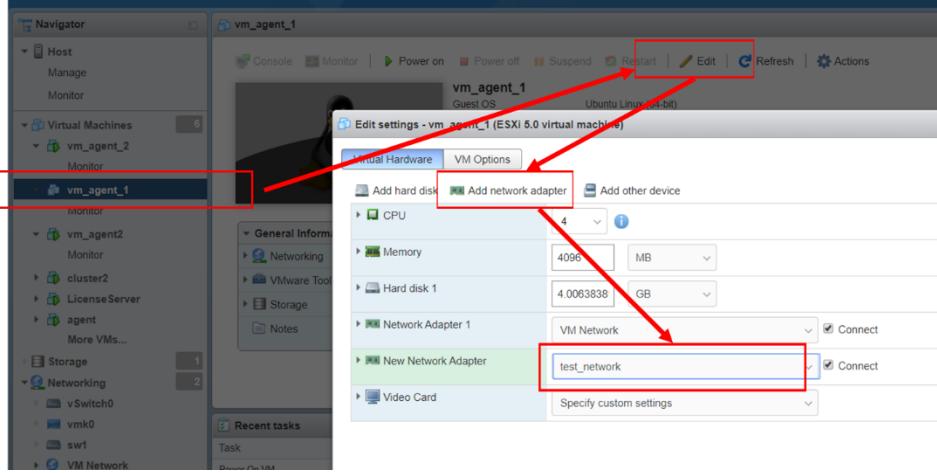
- From the Security section, enable Promiscuous mode and MAC address change.



- Increase the HDD size to 20GB.
- Power off the VM machines (**agent_1** and **agent_2**).



- Add additional NIC to each machine and connect it to test_network port group.



Each agent has two interfaces:

- one for management (**ens32**)
- one for test (**ens160**)

These interfaces can be displayed by running the `if config` command in the agent ssh console:

```
ixia@agent_1:~$ ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.73.51.91 netmask 255.255.248.0 broadcast 10.73.55.255
      inet6 fe80::20c:29ff:fe3:ca9a prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:f3:ca:9a txqueuelen 1000 (Ethernet)
        RX packets 126020 bytes 18866111 (18.8 MB)
        RX errors 0 dropped 5716 overruns 0 frame 0
        TX packets 676 bytes 78634 (78.6 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::20c:29ff:fe3:caa4 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:f3:ca:a4 txqueuelen 1000 (Ethernet)
        RX packets 13 bytes 780 (780.0 B)
        RX errors 0 dropped 11 overruns 0 frame 0
        TX packets 12 bytes 920 (920.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

The next step is to connect the agents VMs with the Middleware VM:

1. Connect to `agent_1` using ssh.
2. Run the `agent-setup.sh` script: `sudo ./agent_setup.sh`.
3. Provide the required information regarding the interfaces used in the test bed:
 - a. Enter the IP address or hostname of the middleware - in this setup **10.73.54.117**.
 - b. Would you like to change the management interface?: **n**. In this case, the **ens32** interface will be used as management interface (if you need to use another interfaces, type **y** and select it from the list).
 - c. Allow the agent to be rebooted from LoadCore UI : **[y]**.

```
ixia@5GCTE-3fd2390080:~$ sudo ./agent-setup.sh
[sudo] password for ixia:

Enter the IP address or hostname of the middleware: 1.1.1.1

Selected interface ens32 with address 10.38.159.102 as management interface
Would you like to change the management interface? [y/n]: n

Do you want to allow this agent to be rebooted from the UI? [y/n]: y

Would you like to change the hostname to 5GCTE-00-0c-29-b0-d6-e0? [y/n]: n

Configuring NTP...
Restarting services...

Agent configuration complete
ixia@5GCTE-3fd2390080:~$
```

- d. Change the hostname: **n**.

Repeat the above steps for second agent (agent_2):

1. Connect to vm_agent_2 using ssh.
2. Run the agent-setup.sh script: `sudo ./agent_setup.sh`.
3. Provide the required information regarding the interfaces used in the test bed:
 - a. The middleware IP address is **10.73.54.117** (the second agent is connecting to the same Cluster).
 - b. The management interface **ens32**.
 - c. Allow the agent to be rebooted from LoadCore UI : **[y]**.
 - d. Change the hostname: **n**.

The agent-setup.sh script can also be run in a non-interactive way using the command: `sudo ./agent-setup.sh <mdw-ip> auto`.

This command will use as management interface the one which has an IP address assigned, the other interfaces found on the agent will be used for test.

If there is no DHCP server in the network, a static IP address needs to assigned to the agent.

```

Enter the IP address or hostname of the middleware: 1.1.1.1

Available network interfaces:
- ens32 - 00:0c:29:b0:d6:e0
- ens160 - 00:0c:29:b0:d6:ea
- ens192 - 00:0c:29:b0:d6:f4

Enter the name of the management interface: 32
Please enter a valid network interface name.
Enter the name of the management interface: ens32
Would you like to configure the management interface? [y/n]: y
  IPv4 address: 1.1.1.2
  Prefix length: 24
  Gateway (enter for none): 1.1.1.1
  DNS (enter for none):

Do you want to allow this agent to be rebooted from the UI? [y/n]: y

Would you like to change the hostname to 5GCTE-00-0c-29-b0-d6-e0? [y/n]: n

Configuring NTP...
Restarting services...

Agent configuration complete

```

If you need to change the automatically assigned IP address of the agent, refer to the instructions presented in [Appendix C](#).

License Manager Installation

To install the License server on a separate VM machine, do the following:

1. Create a new VM machine on the ESXi server and select the License server OVA image.
2. Set a name for the VM (in this deployment the used name is **LicenseServer**).
3. Connect it to the same vSwitch with the Middleware and Agents (VM Network).
4. When the installation is completed, the license server boots normally in the ESXi console, with its IP address assigned:
 - In this deployment, the IP address is **10.73.53.17**.

Virtual machine	Status	Used space	Guest OS	IP address	Host name	Host CPU	Host memory
LicenseServer	Normal	6.36 GB	Ubuntu Linux (64-bit)	10.73.53.17	licensing-vm	27 MHz	908 MB

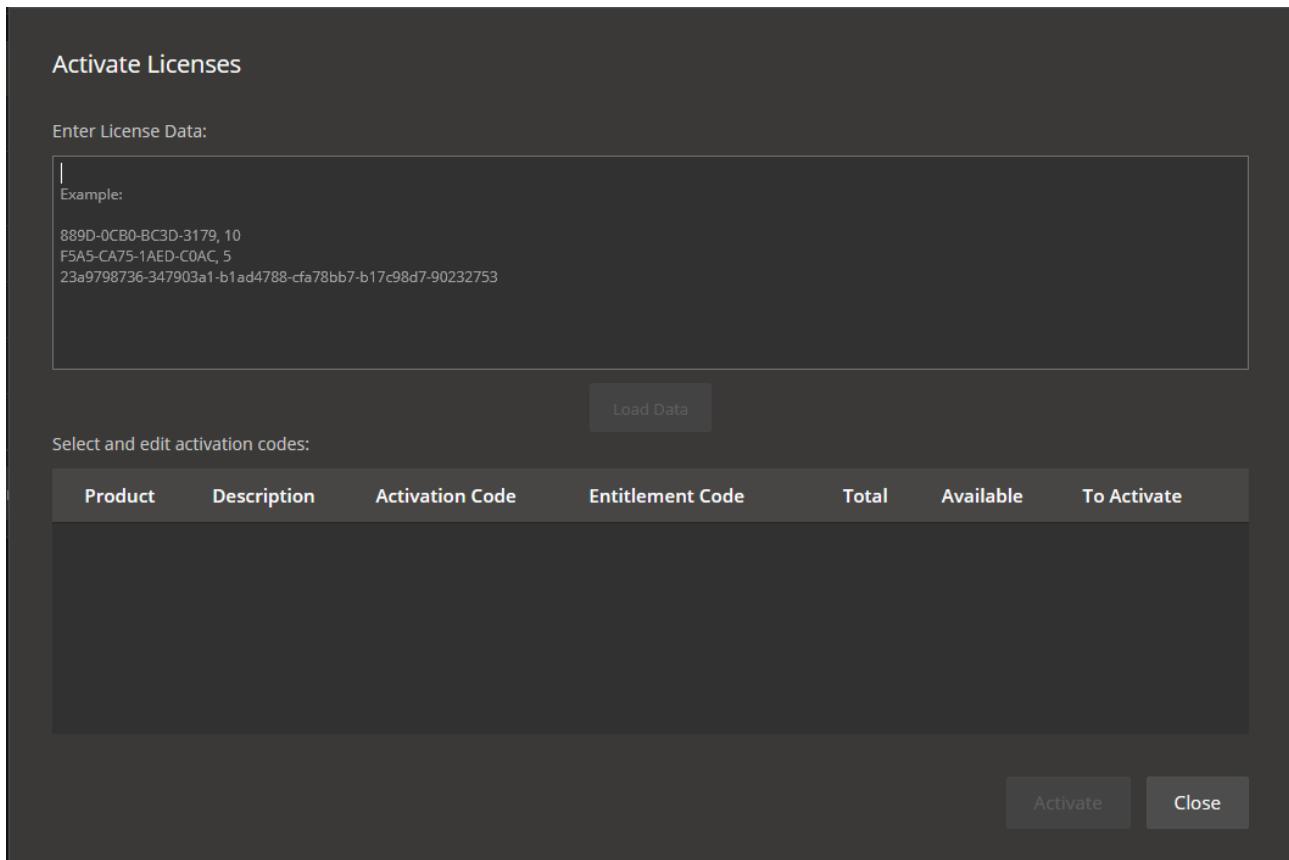
Now you can access the License Server UI and install the licenses:

1. On your browser, type the IP address that was assigned for the License Server.
2. Log in using the default username/password: **admin/admin**.



At this point, there are no licenses available on the newly installed server, therefore you must add the license codes you received (evaluation licenses can be obtained by sending an email to: Software-Eval-Request IX <software-eval-request.ix@keysight.com>.

To activate licenses, select the **Activate Licenses** button and add the license codes, select **Load Data** and then **Activate**.



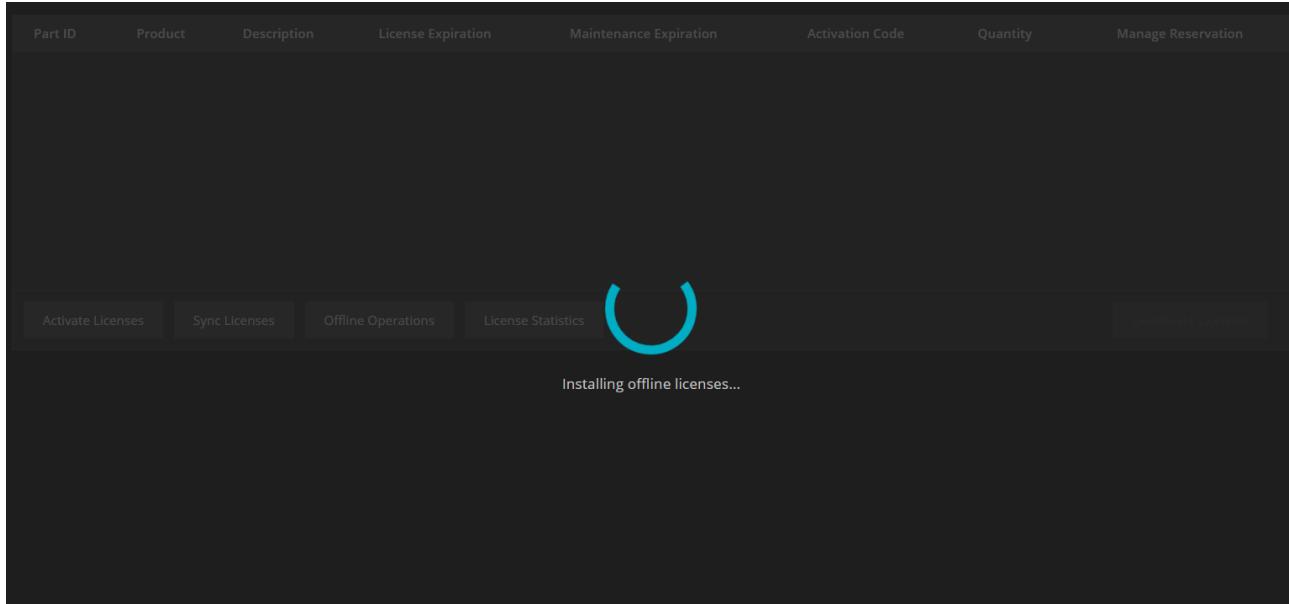
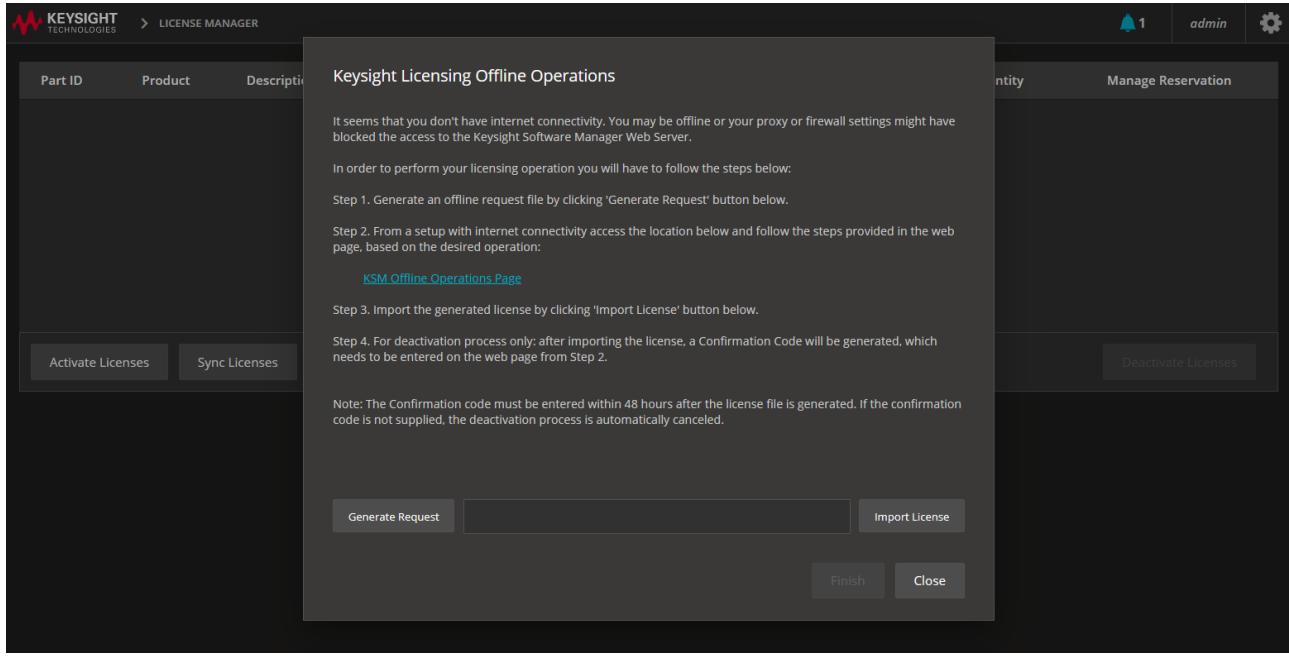
After activation, for each license, details are displayed, as shown below:

LICENSE MANAGER								1	admin	⚙️
Part ID	Product	Description	License Expiration	Maintenance Expiration	Activation Code	Quantity	Manage Reservation			
P89040A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, floating license, Control Plane impairment, 6-months, floating (single site) license	Nov 9, 2027, 12:00:00 AM	Nov 9, 2027, 12:00:00 AM	[REDACTED]	30	[REDACTED]			
P89048A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, SCAS AMF, SMF, UPF, AUSF, UDM, NRF test libraries, 6-months, floating (single site) license	Nov 16, 2027, 12:00:00 AM	Nov 16, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89048A-1NX	LoadCore	R-M4F-002-X WRLS 5G LoadCore, SCAS AMF, SMF, UPF, AUSF, UDM, NRF test libraries, 24-months, floating (single site) license	Nov 16, 2027, 12:00:00 AM	Nov 16, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89048A-1NL	LoadCore	R-M4F-002-L WRLS 5G LoadCore, SCAS AMF, SMF, UPF, AUSF, UDM, NRF test libraries, 12-months, floating (single site) license	Nov 16, 2027, 12:00:00 AM	Nov 16, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89037A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, floating license for User Plane (10Gbps Tput), 6-months, floating (single site) license	Dec 12, 2027, 12:00:00 AM	Dec 12, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89039A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, floating license, high performance all-inclusive bundle (9 CP interfaces, 2M UEs, 20Gbps Tput), 6-months, floating (single site) license	Dec 12, 2027, 12:00:00 AM	Dec 12, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89034A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, floating license, performance enabler (UEs and Procedures per Sec), 6-months, floating (single site) license	Dec 12, 2027, 12:00:00 AM	Dec 12, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			
P89059A-1NF	LoadCore	R-M4F-002-F WRLS 5G LoadCore, floating license, IPSEC capability, 6-months, floating (single site) license	Dec 12, 2027, 12:00:00 AM	Dec 12, 2027, 12:00:00 AM	[REDACTED]	100	[REDACTED]			

At the bottom are buttons for "Activate Licenses", "Sync Licenses", "Offline Operations", "License Statistics", and "Deactivate Licenses".

To synchronize licenses, select the **Sync Licenses** button. This way, if licenses are activated, deactivated or renewed using the offline feature, you can sync all the changes.

To import licenses, go to **Offline Operations**, select **Import License** and select the file that contains the licenses.



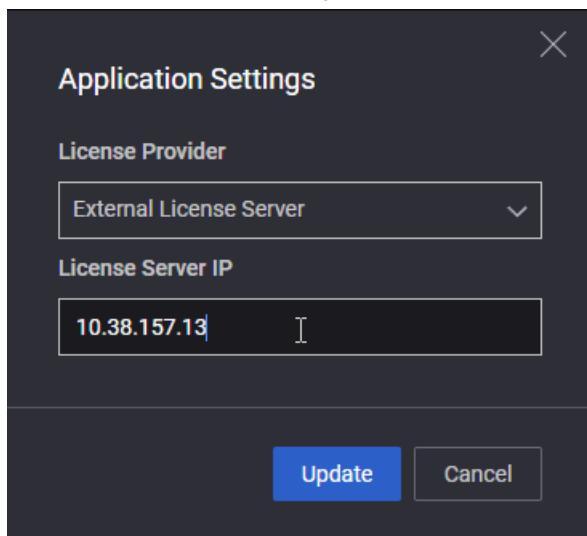
To deactivate licenses, use the **Deactivate Licenses** button. Set the **New Quantity** value for each license and, then, select **Perform Deactivation**.

By selecting the **License Statistics** button, you can view the details of the license features that are associated to the license that you have activated.

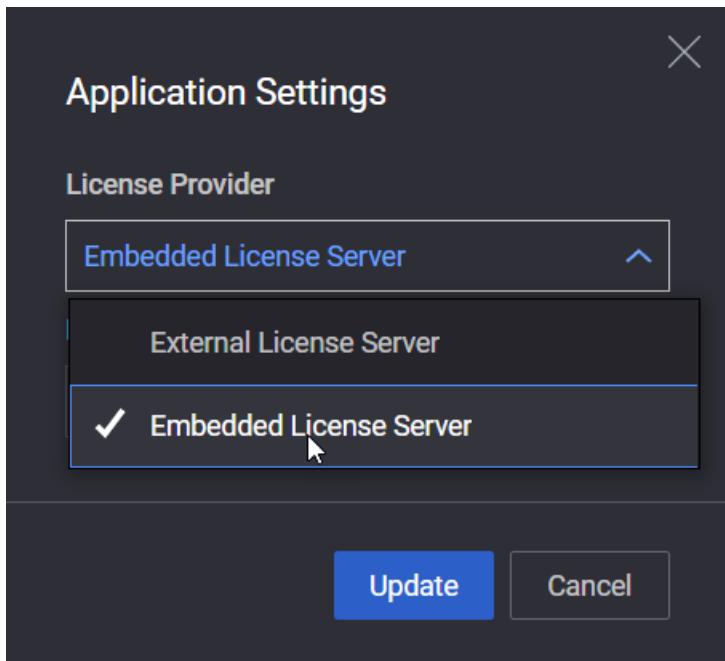
License Statistics							
Feature ↑		Search...					
User	Application	Consumed Count	Reserved Count	Remaining Reservation Duration			
Feature: LC_TestLib_SCAS_all, Available: 300 of 300							
-	-						
Feature: VMP-WRLS-5GC-INTERFACE, Available: 858 of 900							
e7285ea0-a2e5-49b9-af...	LoadCore N/A	10	0	00:00:00			
e7285ea0-a2e5-49b9-af...	LoadCore N/A	11	0	00:00:00			
bd0de644-5cce-460b-a...	LoadCore N/A	11	0	00:00:00			
e7285ea0-a2e5-49b9-af...	LoadCore N/A	10	0	00:00:00			
Feature: VMP-WRLS-5GC-RATE, Available: 29993 of 30000							
e7285ea0-a2e5-49b9-af...	LoadCore N/A	1	0	00:00:00			
e7285ea0-a2e5-49b9-af...	LoadCore N/A	1	0	00:00:00			
bd0de644-5cce-460b-a...	LoadCore N/A	1	0	00:00:00			
1 2							
Close							

LoadCore allows you to use two types of license servers in order to manage licenses. From the LoaCore Web UI select gear menu > **Application Settings**:

- **External License Server** - select this option to set an external license server (provide the license server IP address).



- **Embedded License Server** - the license server that is included in LoadCore MW.



Steps to Upgrade your License Server

NOTE

The following procedures applies to LoadCore version 3.4 or earlier.

If you are using an embedded license server in an ESX/KVM/AWS/Azure virtual machine environment:

1. Download the licensing upgrade .tar file [here](#).
2. Sign in to the LoadCore UI by using the **admin** user credentials.
3. On the top right side, select **Administration > Software Updates**.
4. Choose **Select Packages for Upload**.
5. Select the downloaded .tar file.
6. Select **Start Update**. After the process starts, a progress indicator appears. When the patch installation is complete, you will receive a notification.

After this stage, the license server that is embedded inside the Virtual Machine is updated and ready to use.

If you are using an embedded license server in a Kubernetes (k8s/Anthos/OpenShift/EKS) environment:

1. Download the container licensing patch .zip file [here](#). In the zip archive, you will find the docker image for the new Embedded License Server, and also the folder with the charts modified.
2. Unzip the archive. After unzipping this archive, you will find the docker image and a folder.

3. Where you extracted `LoadCore-MDW-3.4.0-3490-268-AGENT-3.4.0-9` archive, find a charts directory. Delete from that directory the `kcos-licensing` as follows:

- `rm -rf charts/kcos-licensing`
- Move the `kcos-licensing` folder (which is compressed in the zip archived) to the charts directory.

4. Load the docker image by using:

```
docker load -i <docker image name>
```

For example: `docker load -i kcos-licensing-0.3.5_20220727.134517.db24dc3f.tar.gz`

5. Tag the docker image as follows:

```
docker tag <IMAGE ID> <registry_name>/<project_name>/kcos-licensing:0.3.5_
20220727.134517.db24dc3f
```

For example: `docker tag 014439e08129 default-route-openshift-image-
registry.apps.cluster.keysight.lab/loadcore-root/kcos-licensing:0.3.5_
20220727.134517.db24dc3f`

6. Push the image inside our repository.

```
docker push <registry_name>/<project_name>/kcos-licensing:0.3.5_
20220727.134517.db24dc3f
```

For example: `docker push default-route-openshift-image-
registry.apps.cluster.keysight.lab/loadcore-root/kcos-licensing:0.3.5_
20220727.134517.db24dc3f`

7. Install LoadCore with the new Embedded License Server:

```
./loadcore-kube-setup install-mw
```

After this stage, the license server that is embedded inside the Kubernetes is updated and ready to use.

If you are using LoadCore-Licensing-1.7.0-4 and want to upgrade the image in-place:

1. Download the licensing patch .tar file [here](#).
2. Sign in to the Licensing UI by using the **admin** user credentials.
3. In the Licensing UI, select the **Gear** menu > **Administration** > **Software Updates**.
4. Choose **Select Packages for Upload** and select the downloaded .tar file.
5. Select **Start Update**. After the process starts, a progress indicator appears. When the patch installation is complete, you will receive a notification in the bottom right corner.

After this stage, the license server is updated and ready to use.

If you are using legacy licensing or LoadCore-Licensing-1.7.0-4 and want to deploy a new image:

1. Download the licensing image:
 - ESX image [here](#)
 - KVM image [here](#)
2. De-activate your licenses from the old licensing deployment.

If you are using offline licenses, please contact Support for deactivation / reactivation.

3. Deploy the downloaded images.
4. Activate your licenses on the new deployment.
5. Point your middleware to the new licensing deployment.

IMPORTANT

If you are upgrading from License Server release earlier than 1.7.04, you must be aware that the CLI has changed.

To log in, type `console`, and when prompted, use the default username and password:
admin/admin.

If necessary, please refer to the [*KCOS CLI Reference Guide*](#) in order to obtain the DHCP IP address of the License Server.

To interact with the License Server, refer to the [License Manager Installation](#) section presented above.

Amazon AWS Deployment

This section describes the steps needed to deploy LoadCore in Amazon AWS.

Prerequisites

For a complete and correct functioning setup, make sure you have downloaded and installed the packages on your test environment:

- **Wireshark** capable to decode PFCP messages and IEs (at least 2.5.2).
This will be used to analyze the traffic captures.
- **AWS account**
The examples presented in this guide were done using a Full Core topology deployed in B2B scenario (no real/external DUT).
- **LoadCore images** (three images: one for Middleware, for Test Agents and one for License Server).

There are few options to get the images:

- Get AMI images from AWS Marketplace.
<https://aws.amazon.com/marketplace/search/results?searchTerms=keysight>
Regions supported: eu-north-1, ap-south-1, eu-west-3, eu-west-2, eu-west-1, ap-northeast-2, ap-northeast-1, sa-east-1, ca-central-1, ap-southeast-1, ap-southeast-2, eu-central-1, us-east-1, us-east-2, us-west-1, us-west-2
- Get the shared images from our AWS account. For this you need to contact the Support Eng. assigned to help with this activity.

- **AWS Cloudformation templates**

The LoadCore setup can be also deployed using Cloudformation templates. The templates can be found at the following location:

<https://github.com/Keysight/loadcore/tree/main/AWS>

How to use Cloudformation templates:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/GettingStarted.Walkthrough.html>

- **Valid licenses** for Ixia License server. For deploying full 5G core and run Control Plane and User Plane traffic, the following licenses are required:

- Control plane licenses:
 - LoadCore interface license simulation (**P89033A** x 11 pcs).
This license will enable Control Plane testing. One license is needed for each simulated interface.
 - 5G Core Performance enabler on VM: 1M UEs and 10k TPS for VM (**P89034A** x 1 pcs).
- User Plane licenses:
 - User Plane Flow-based license (N3 and N6). Multiple QTY are needed if multiple flows are active simultaneously. It includes three Application Traffic Flows (TCP/UDP) and 10Gbps throughput capacity (**P89037A** x 2 pcs). The recommendation is to use this type of license as this will be suggested for all new customers.

- As an alternative for the above license you can also use Tier-4 license (**P89030A** or **P89031A** x 2 pcs).

IMPORTANT These license types will enable User Plane traffic and are not needed if only Control Plane traffic is needed for future tests.

- **Skills needed by a user:**

- 5G core network knowledge
- familiarity with AWS EC2, EBS and VPC services
- in case the user wants to run automated tests, REST API knowledge is required in their choice programming language

Resource requirements in AWS

LoadCore Middleware instance resources

By default, the VM for Middleware will reserve the following compute resources:

- 8 x vCPUs
- At least 16 GB RAM
- 256 GB HDD (in LoadCore 1.3 the HDD space was increased to 256G since the test results are now stored on Middleware and additional space was needed for this)

Recommended instance types: **m4.2xlarge, t2.2xlarge, c4.4xlarge, c5.2xlarge, c5.4xlarge**.

LoadCore Test Agent instance resources

The **Test Agent** will reserve the following compute resources:

- 4 x vCPUs
- 4 GB RAM **out of which 1GB is reserved for HUGE MEM(DPDK)**

IMPORTANT This value is for Control Plane only. If you are running app traffic the recommendation is to allocate minimum 16 GB RAM.

- 20 GB HDD

Recommended instance types: **c4.2xlarge, c5.2xlarge, t2.xlarge, t2.2xlarge, c4.4xlarge, c4.8xlarge, c5.xlarge, c5.4xlarge, c5.9xlarge**.

Licensing server default resources

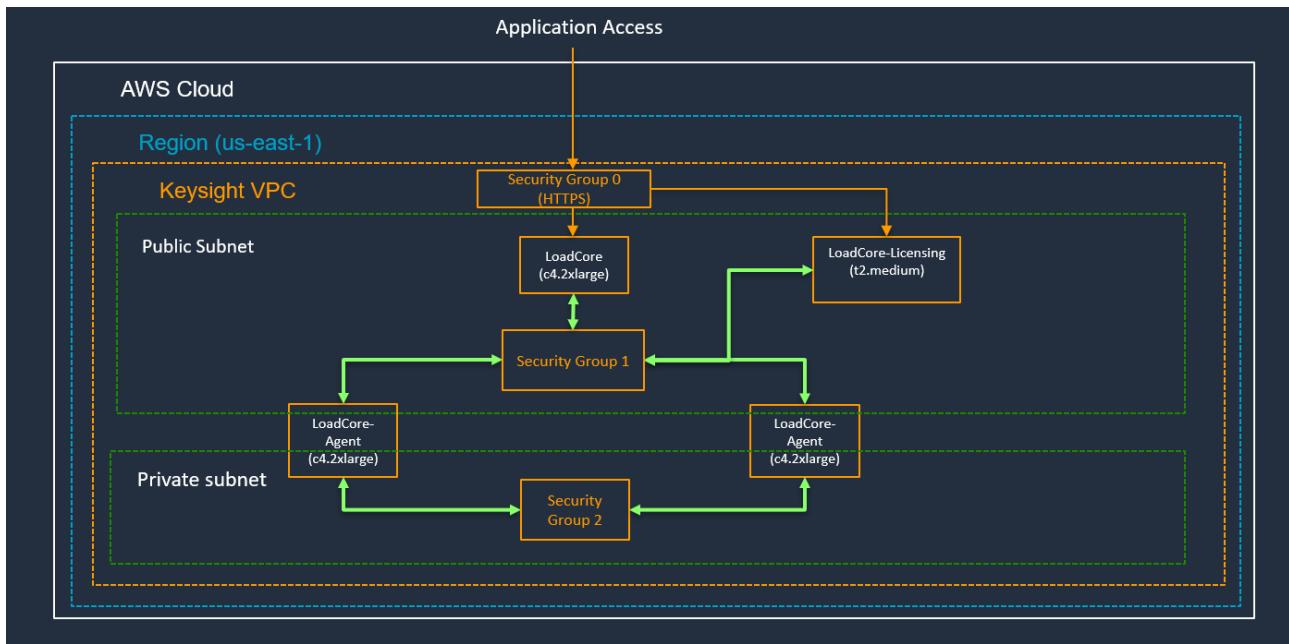
- 2 x vCPUs
- 4 GB RAM
- 11 GB HDD

Recommended instance types: **t2.medium, t2.large, t2.xlarge, c4.large, c4.xlarge**.

AWS services and components

Amazon **Virtual Private Cloud** (Amazon VPC) enables you to launch AWS resources into a virtual network that you have defined. Amazon VPC is the networking layer for Amazon EC2 and it is logically isolated from other virtual networks in the AWS Cloud.

The following diagram explains how LoadCore can be deployed in AWS:



For LoadCore deployment, you can reuse an already existing VPC or create a new one.

In this guide, a new VPC is created, as presented in the following topics.

Create a Virtual Private Cloud

You can create an Amazon Virtual Private Cloud as follows:

1. Select **AWS Management Console > Services > VPC**.

AWS services

Find Services

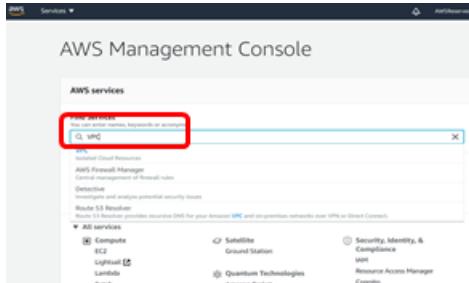
You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

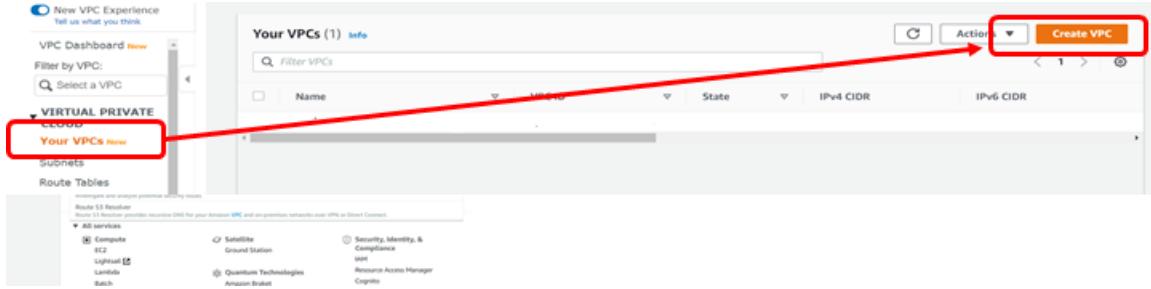
▼ Recently visited services



You can also use **Find Services** option and search for VPC:



2. Select **Your VPC** and, then, select **Create a VPC**.



3. We will define **loadcore-vpc** with a **10.0.0.0/16** mask network (IPv4 CIDR). Later on, we will define two subnets in this CIDR block:

Create VPC [Info](#)

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

IPv4 CIDR block [Info](#)

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)

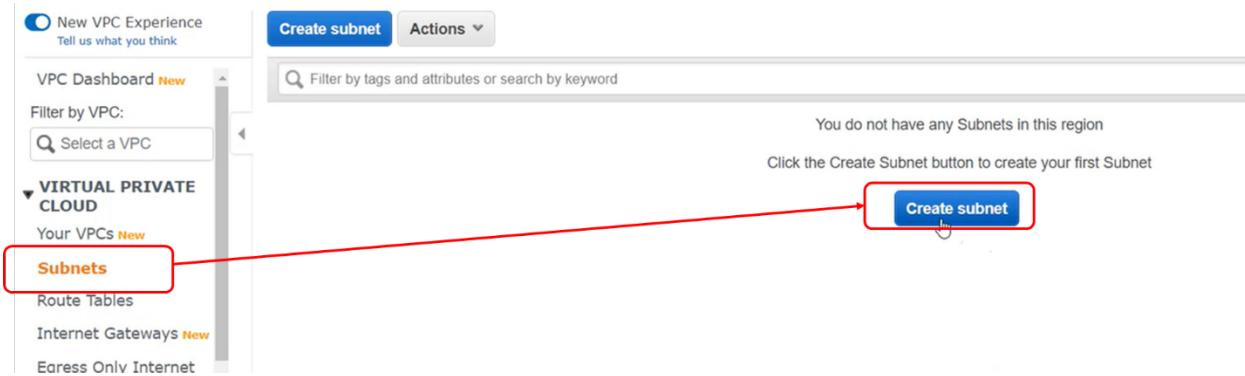
4. At the end of this operation you should have the following result:

Your VPCs (1/1)						Actions	Create VPC		
<input type="text" value="Filter VPCs"/>						<	1	>	⚙️
<input checked="" type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR				
<input checked="" type="checkbox"/>	loadcore-vpc	vpc-0f83e58ca0488394d	Available	10.0.0.0/16	-				

Create the management and test subnets

You need to create two subnets inside the VPC defined at the previous step, one subnet to be used for management and the other one to be used for test.

A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that will not be connected to the internet.



We will use **10.0.0.0/24** for management (with the name **mgmt.-subnet**) and **10.0.10.0/24** for test (with the name **test.-subnet**).

The subnet can be created as follows (in this example, the management subnet):

1. Select the **VPC ID** (there will be only one VPC – the one created at the previous step). If you are reusing an existing VPC, make sure to select the correct VPC ID.
2. Define the subnet TAG (this is the name assigned to the subnet).
3. Define the subnet IP addresses (in this case **10.0.0.0/24**).
4. Select **Create Subnet**.

VPC ID
Create subnets in this VPC.
vpc-0f83e58ca0488394d (loadcore-vpc)

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference

IPv4 CIDR block Info
10.0.0.0/24

▼ Tags - optional

Repeat the steps presented above for test subnet also (using **10.0.10.0/24** IP definition).

After creating the two subnets, your console should display the following image:

- The **Subnet ID** is automatically assigned.
- Both subnets should belong to the same VPC.

Subnets (2) Info						
	Name	Subnet ID	State	VPC	IPv4 CIDR	
<input type="checkbox"/>	mgmt-subnet	subnet-0a87408455ab56c3e	Available	vpc-0f83e58ca0488394d lo...	10.0.0.0/24	Actions
<input type="checkbox"/>	test-subnet	subnet-04267cc8ce65eb610	Available	vpc-0f83e58ca0488394d lo...	10.0.10.0/24	Actions

Create the Internet Gateway

An internet gateway is a VPC component that enables communication between your VPC and the Internet.

To use an internet gateway, attach it to your VPC and specify it as a target in your subnet route table for internet-routable IPv4 or IPv6 traffic. An internet gateway performs network address translation (NAT) for instances that have been assigned public IPv4 addresses.

We will name it **my-internet-gateway**, after that, we need to attach it to our VPC.

The screenshot shows the AWS VPC Internet Gateways creation interface. On the left, a sidebar lists 'Internet Gateways' under 'VIRTUAL PRIVATE CLOUD'. The main panel shows a list of existing Internet Gateways and a 'Create internet gateway' form. The 'Create internet gateway' form includes fields for 'Name tag' (containing 'my-internet-gateway') and 'Tags - optional' (with a single tag 'Name: my-Internet-gateway'). The 'Create internet gateway' button is highlighted with a red box in three different locations: at the top right of the main panel, at the bottom right of the settings panel, and again at the bottom right of the main panel.

The newly created gateway is in the state **Detached**, with no VPC ID associated. Therefore, we need to attach it to our VPC:

1. Select the gateway.
2. Select **Actions** and, then, select **Attach to VPC**.

- Select the previous defined VPC.

VPC > Internet gateways > Attach to VPC (igw-08edd18d8e0167ea9)

Attach to VPC (igw-08edd18d8e0167ea9) [Info](#)

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

X

▶ AWS Command Line Interface command

[Cancel](#) [Attach internet gateway](#)

At the end of this step, the following image should be displayed:

Internet gateways (1/1) Info						Actions
<input checked="" type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner	
<input checked="" type="checkbox"/>	my-internet-gateway	igw-08edd18d8e0167ea9	Attached	vpc-0f83e58ca0488394d loadcore-vpc	672779868767	

Assign traffic routes

The next step is to assign traffic routes, in order to pass the external traffic through the gateway that we just defined. This can be done as follows:

- Select the **Route table** menu (a *route table* contains a set of rules, called *routes*, that are used to determine where network traffic from your subnet or gateway is directed).
- Select **Add Route** and define the default rate: **0.0.0.0/0**.
- For **Target**, select **Internet Gateway** and select the gateway previously defined.
- Select **Save Routes**.

Route Tables > Edit routes

Edit routes

Destination	Target	Status
10.0.0.0/16	local	active
0.0.0.0/0	local	active

* Required

Add route

Egress Only Internet Gateway Instance

Internet Gateway

If you select **Routes**, the following configuration should be displayed:

Name	Route Table ID	Explicit subnet associations	Edge associations	Main	VPC ID
	rtb-09038b62a74cab699	-	-	Yes	vpc-0f83e58ca0488394d ...

Route Table: rtb-09038b62a74cab699										
Summary	Routes									
Edit routes	View All routes									
	<table border="1"> <thead> <tr> <th>Destination</th> <th>Target</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>10.0.0.0/16</td> <td>local</td> <td>active</td> </tr> <tr> <td>0.0.0.0/0</td> <td>igw-08edd18d8e0167ea9</td> <td>active</td> </tr> </tbody> </table>	Destination	Target	Status	10.0.0.0/16	local	active	0.0.0.0/0	igw-08edd18d8e0167ea9	active
Destination	Target	Status								
10.0.0.0/16	local	active								
0.0.0.0/0	igw-08edd18d8e0167ea9	active								

Configure Security Groups

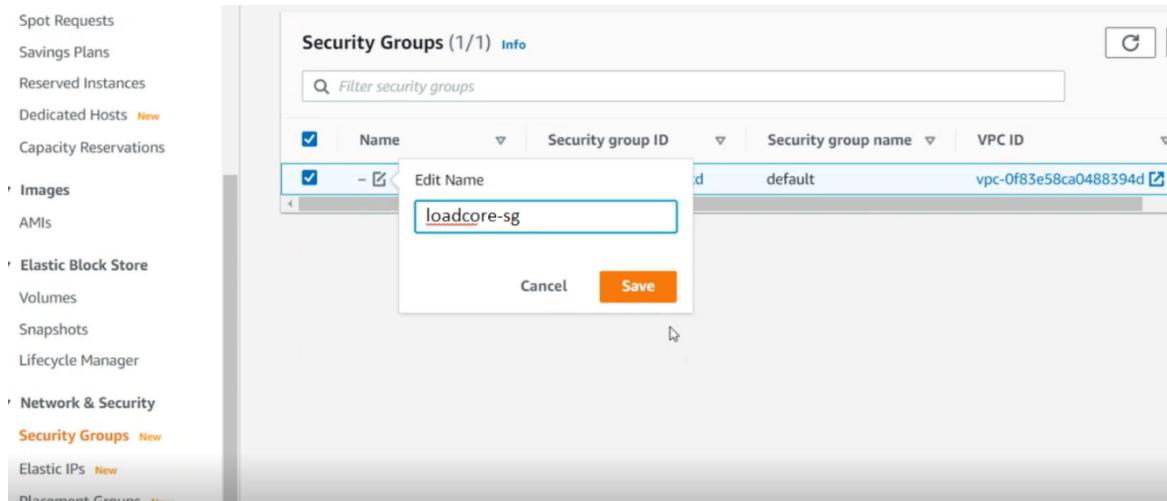
To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.

You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

By default, when the VPC was created, also a Security Group was automatically defined.

We will define a name for it (**loadcore-sg**) and configure the inbound rules:



Now we need to define Inbound security rules, as follows:

1. Select the security group ID and go on Inbound rules.
2. Select **Add rule**.

- For security reasons we will allow the traffic only for specific ports used by LoadCore and originated from a single IP address(the one used to access the internet).

Inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Type Info	Protocol Info	Port range Info	Source Info
All traffic	All	All	Custom sg-0dcc6656aeeb6c9cd

[Add rule](#)

- We will permit the following:
 - SSH, to access the LoadCore VMs,
 - HTTP, HTTPS, used by LoadCore for configuration, interaction between nodes,
 - Custom: 7443 port, that is used by the LoadCore License Server,
 - For all traffic we will select *My IP*, and the browser will automatically fill it with the outgoing IP address that you have (in this case **188.25.103.79**).

At the end of this step, you should have the following rules in your Security Group:

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info
HTTP	TCP	80	Custom 188.25.103.79/32
SSH	TCP	22	Custom 188.25.103.79/32
HTTPS	TCP	443	Custom 188.25.103.79/32
Custom TCP	TCP	7443	Custom 188.25.103.79/32

[Add rule](#)

Create Key Pairs

The final step is to create the key pair, as presented below.

The screenshot shows the 'Create key pair' wizard in the AWS Management Console. The top navigation bar includes 'EC2 > Key pairs > Create key pair'. The main title is 'Create key pair'. A 'Key pair' section defines it as a set of security credentials. The 'Name' field contains 'lc-keypair'. Below it, a note says 'The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.' The 'File format' section shows 'pem' (selected) and 'ppk' (selected). The 'Tags (Optional)' section indicates 'No tags associated with the resource.' and has an 'Add tag' button. At the bottom right are 'Cancel' and 'Create key pair' buttons.

EC2 > Key pairs > Create key pair

Create key pair

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
lc-keypair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

File format

pem
For use with OpenSSH

ppk
For use with PuTTY

Tags (Optional)
No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel **Create key pair**

LoadCore Components Installation

Until now, we have created the minimum infrastructure required to deploy the LoadCore components.

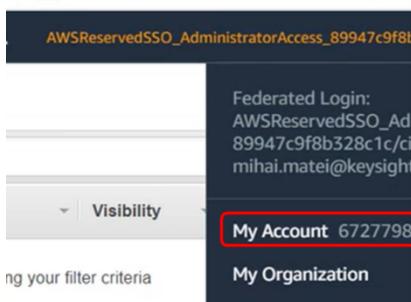
In AWS, the images that we are using are in AMI format.

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

The AMI images can be found on AWS marketplace or can be shared from our AWS account.

In this deployment, we will use shared images.

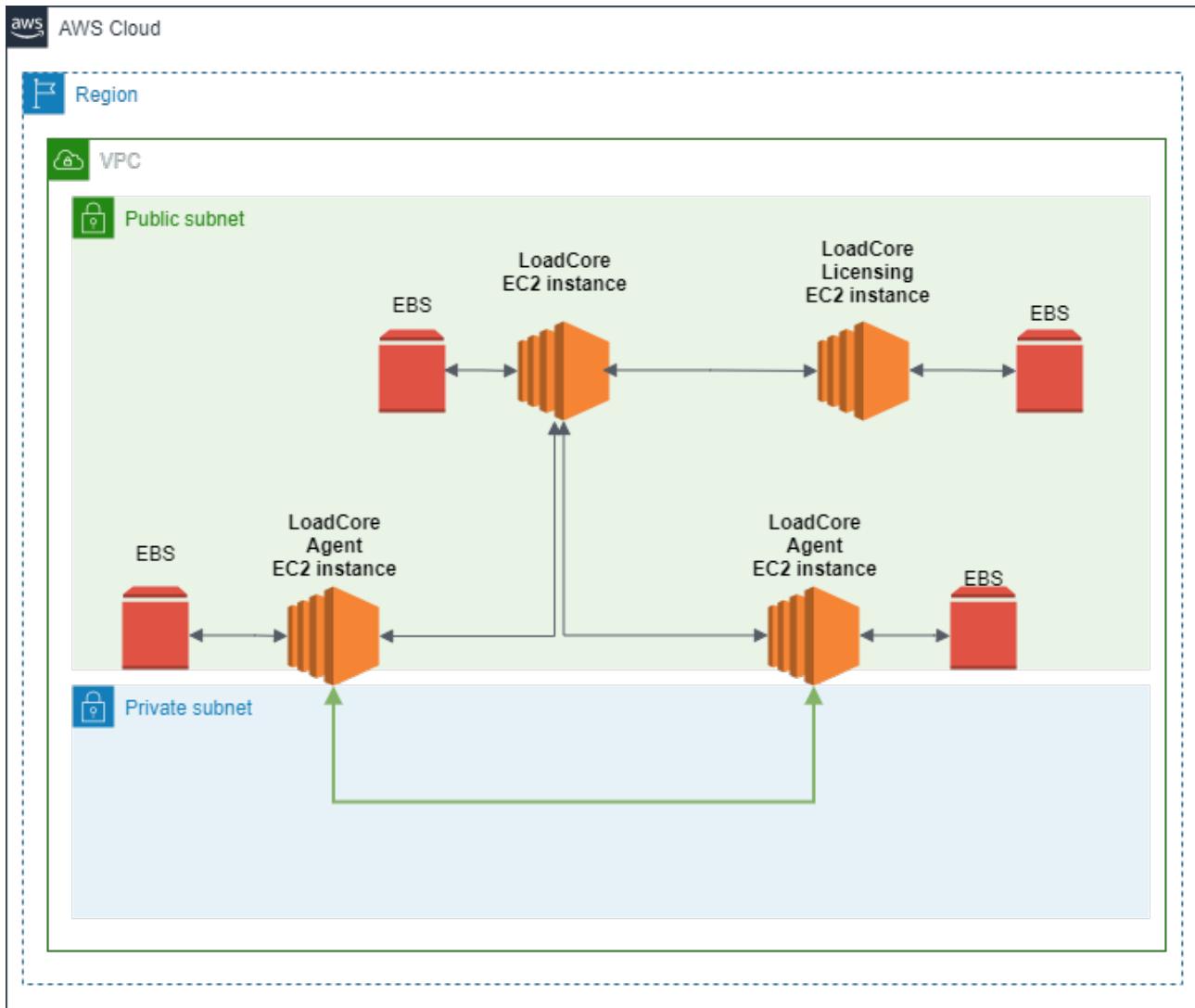
To get access to them, please contact our Support or PM team. You need to provide your AWS account ID to get access to the shared images.



In order to install the LoadCore Middleware, Test Agents and License server, the AMI images should be available for you, in your AWS account.

The installation process basically consists of launching the AMI images.

The following diagram illustrates some of the AWS services used by LoadCore deployment:



Middleware Installation

The installation procedure of the Middleware requires the following steps:

- From the main console, select **Services > EC2 > Images > AMI**.

The browser will automatically display the **images that you own**, therefore, if you do not have other images you created, the list will be empty.

Select **Private Image** and you will find the installation images that were shared with you.

The AMIs can be also found on AWS Marketplace at the following location:

<https://aws.amazon.com/marketplace/search/results?searchTerms=keysight>

The screenshot shows the AWS EC2 Images interface. On the left, there's a sidebar with 'New EC2 Experience' and sections for EC2 Dashboard, Events, Tags, Limits, Instances, and Images/AMIs. The 'Images' section is highlighted with a red box. On the right, there's a search bar and a table of AMIs. A dropdown menu above the table shows filter options: 'Owned by me', 'Public images', and 'Private images', with 'Private images' selected and highlighted with a red box. A red arrow points from the 'Images' box in the sidebar to the 'Private images' option in the dropdown. Below the table, it says 'Image: ami-07a687c8f3e6bd1d5'.

You can also give a name to the AMI image, to be easily identified (in this case **LC-MW-1.3**).

The screenshot shows the same AWS EC2 Images interface as before, but now the table includes a column for 'Name'. One row in the table has the name 'LC-MW-1.3' highlighted with a yellow box. The rest of the table columns include 'AMI Name', 'AMI ID', 'Source', 'Owner', 'Visibility', and 'Status'.

Once you have the image available, select it and select **Launch**.

The launch process will request additional info, which is meant to adjust the resources in AWS.

The general advice here is to select the same amount of resources as in the other installation (for example, ESXi):

- Select the Instance type:

- Select **c5.2xlarge**.

The screenshot shows a dropdown menu for selecting an instance type. The 'c5.2xlarge' option is highlighted with a green checkmark. Other options listed include c5.large, c5.4xlarge, c5.xlarge, c5.12xlarge, c5.24xlarge, c5.9xlarge, and another c5.2xlarge entry. A sidebar on the right provides filtering options like 'Free tier eligible' and 'Compare instance types'.

Instance Type	Family	vCPU	Memory	On-Demand Linux pricing	On-Demand Windows pricing
c5.large	c5	2 vCPU	4 GiB Memory	0.085 USD per Hour	0.177 USD per Hour
c5.4xlarge	c5	16 vCPU	32 GiB Memory	0.68 USD per Hour	1.416 USD per Hour
c5.xlarge	c5	4 vCPU	8 GiB Memory	0.17 USD per Hour	0.354 USD per Hour
c5.12xlarge	c5	48 vCPU	96 GiB Memory	2.04 USD per Hour	4.248 USD per Hour
c5.24xlarge	c5	96 vCPU	192 GiB Memory	4.08 USD per Hour	8.496 USD per Hour
c5.2xlarge	c5	8 vCPU	16 GiB Memory	0.34 USD per Hour	0.708 USD per Hour
c5.9xlarge	c5	72 vCPU	172 GiB Memory	10.04 USD per Hour	20.96 USD per Hour
c5.2xlarge	c5	8 vCPU	16 GiB Memory	0.34 USD per Hour	0.708 USD per Hour

- Configure the Instance Details:

- VPC – select the VPC we defined. In our case there is only one, but if you have multiple VPCs, you need to select the one you want to use to deploy LoadCore.
- Select the management subnet.

- Select **Add Storage**.

By default, the **Size** will come with 100GB.

For long duration tests, you need to increase the size to 256GB.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance. You can edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2](#).

The screenshot shows the storage configuration for a Lambda function. It includes fields for Volume Type, Device, Snapshot, Size (GiB), and Volume Type. The 'Size (GiB)' field for the root volume is highlighted with a red box and contains the value '100'. There is also an 'Add New Volume' button.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type
Root	/dev/sda1	snap-0a30da8b309439f13	100	General Purpose SSD (gp2)

- There is no need to define Tags, select **Configure Security Group**.
 - Select the security group we previously defined.
 - Select **Review and Launch**.

Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:	<input type="radio"/> Create a new security group	<input checked="" type="radio"/> Select an existing security group					
Security Group ID	sg-0dcc6656aeeb6c9cd	Name	default	Description	default VPC security group	Actions	Copy to new
Inbound rules for sg-0dcc6656aeeb6c9cd (Selected security groups: sg-0dcc6656aeeb6c9cd)							
Type	Protocol	Port Range	Source	Description			
HTTP	TCP	80	188.25.103.79/32				
All traffic	All	All	sg-0dcc6656aeeb6c9cd (default)				
SSH	TCP	22	188.25.103.79/32				

Review and Launch

2. Select **Launch**.

This will display a pop-up menu from where you can select/create a key pair. If you already have a key pair (and you downloaded it), you can select that one, otherwise you can create a new one, and download it from here, as shown below:

- Select **Create a new key pair**.
- Define a name for it: **lc-keypair**.
- Select **Download Key Pair**.
- Select **Launch Instances**.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

<input type="radio"/> Create a new key pair
Key pair name
lc-keypair

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel **Launch Instances**

3. At this point, the instance is launched, as shown below.

Launch Status

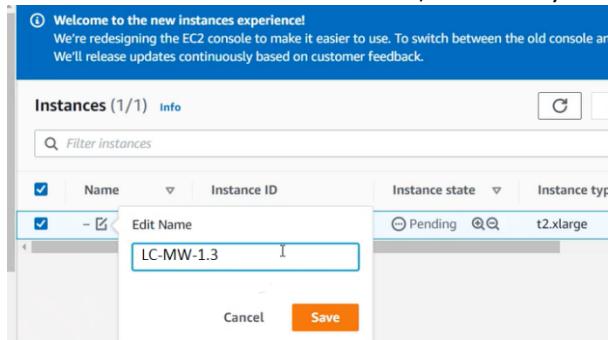
Your instances are now launching
The following instance launches have been initiated: i-0cbcf768f02edab2e [View launch log](#)

i Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amo

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready to terminate your instances.

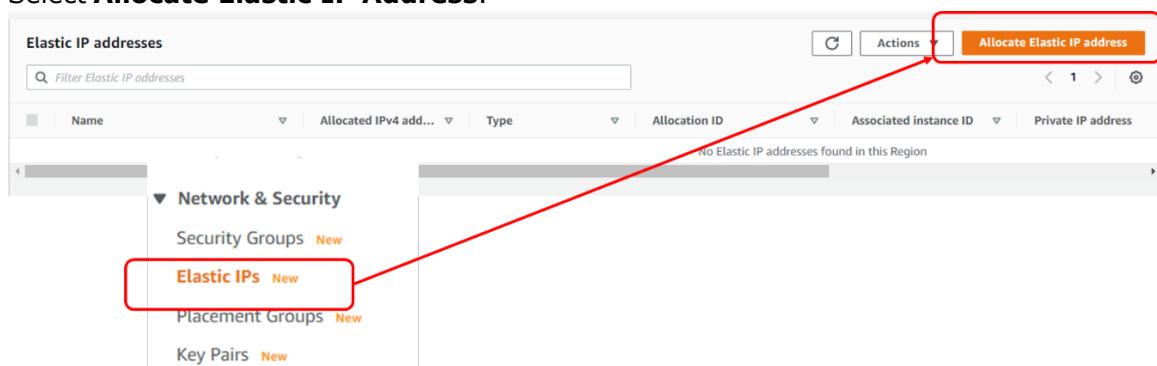
You can also define a name for it, to be easy to identify it (in this case **LC-MW-1.3**).



4. To access the LoadCore Middleware UI, we will need to assign an Elastic IP.

To do this, select the Elastic IP menu:

- Select **Allocate Elastic IP Address**.



Allocate Elastic IP address

Allocate an Elastic IP address from a public IPv4 address pool, or use global IP addresses from AWS Global Accelerator. You can have one Elastic IP associated with a running instance at no charge. You're charged for additional Elastic IPs that are associated with the instance, Elastic IPs that are associated with stopped instances or unattached network interfaces, and unassociated Elastic IPs. [Learn more](#)

Elastic IP address settings

Public IPv4 address pool
Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise..

- Amazon's pool of IPv4 addresses
- Public IPv4 address that you bring to your AWS account(option disabled because no pools found) [Learn more](#)
- Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) [Learn more](#)

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

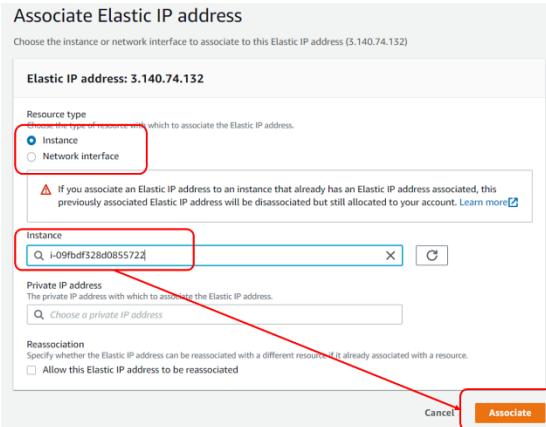
[Cancel](#)
Allocate

- The new IPv4 Public IP address will be allocated from AWS pool.
- Associate this IP address with the Middleware instance.

⌚ Elastic IP address allocated successfully.
Elastic IP address 3.140.74.132
Associate this Elastic IP address

Elastic IP addresses (1/1)					
				Actions	
				View details Release Elastic IP addresses Associate Elastic IP address Disassociate Elastic IP address Create IP address	
				1 > @	
Name	Allocated IPv4 add...	Type	Allocation ID	A	Actions
-	3.140.74.132	Public IP	eipalloc-09f0e35dabc978f2c	-	Disassociate Elastic IP address Create IP address

- Select the Middleware instance id, and then, select **Associate**:



IMPORTANT This is the IP address of the LoadCore UI and you will also use it to install the wireless components.

Using an SSH client we can connect to the middleware using public key authentication:

```
ssh -i <private-key> loadcore@<Elastic_IP_address>
```

Agent(s) Installation

In LoadCore, you can use two different stacks, Linux Stack or IxStack:

- Linux Stack is used for Stateless UDP traffic.
- IxStack is used for Application traffic. This can work with Raw Sockets or over DPDK/Raw.

There are environments where you cannot use IxStack over DPDK/Raw or is not available.

In AWS is not possible to bind the interface to IxStack over DPDK/Raw, therefore we will use raw sockets for Application traffic.

In order to use raw sockets, you need to make some changes.

By default, when you create an instance, you assign from AWS some default interfaces, called VIF, but these interfaces cannot be bind, as they do not have a PCI address. With this default interface you can run only Stateless UDP tests (with Linux Stack).

If Application traffic is needed, we will use a different instance that will give you access to two different interfaces/drivers:

- intel sriov, driver ixgbe vf
- ena driver

Enhanced networking support

All [current generation](#) instance types support enhanced networking, except for T2 instances.

You can enable enhanced networking using one of the following mechanisms:

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 100 Gbps for supported instance types.

The current generation instances use ENA for enhanced networking, except for C4, D2, and M4 instances smaller than m4.16xlarge.

Intel 82599 Virtual Function (VF) interface

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

The following instance types use the Intel 82599 VF interface for enhanced networking: C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3.

SRIOV support should be enabled from AWS CLI.

The installation procedure of the Agent(s) requires the following steps:

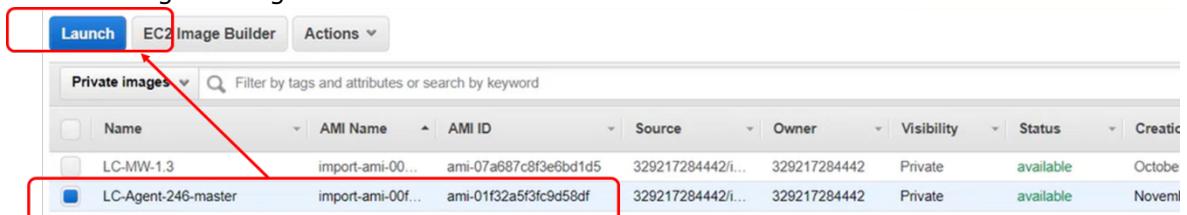
- From the main console, select **Services > EC2 > Images > AMI**.

The browser will display the AMI images shared for your account.

The AMIs can be also found on AWS Marketplace at the following location:

<https://aws.amazon.com/marketplace/search/results?searchTerms=keysight>

Select the Agent image and select **Launch**.



Select **t2.xlarge** for instance type:

<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate

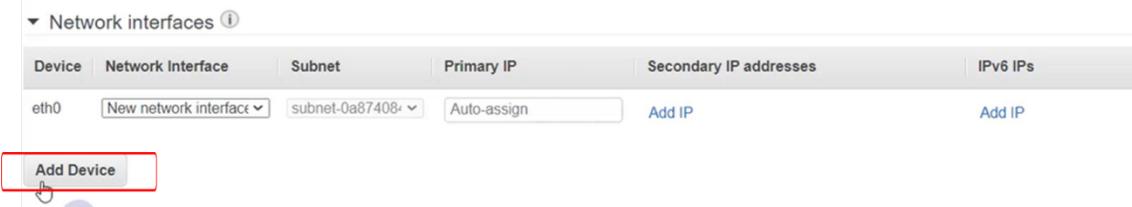
Select **Next**.

- Configure the instance details.

Select the VPC and assign the network interface in the management subnet.

By default, the instance comes with only one interface. For our agent installation we need to add the second interface to be used for test. To do that:

- Scroll down to the network interfaces and select **Add Device**.



- Assign it to the test subnet (previously defined).

Step 3: Configure Instance Details

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-0a87408e	Auto-assign	Add IP	
eth1	New network interface	subnet-0a87408e	Auto-assign	Add IP	

A red box highlights the 'Add IP' button for 'eth1' and the dropdown menu which lists 'subnet-0a87408e (mgmt-subnet) 10.0.0.0/24 us-east-2a' and 'subnet-04267cc8ce65eb610 (test-subnet) 10.0.10.0/24 us-east-2a'.

Info: We can no longer assign a public IP address to your instance

- The auto-assign public IP address feature for this instance is disabled because you specified multiple network interfaces for instances with one network interface. To re-enable the auto-assign public IP address feature, please specify only one network interface.

Select **Next**.

3. Add Storage.

By default, the image comes with 17GB assigned. This can be increased if more space is necessary. Simple tests can run with this value, otherwise define 20GB.

Select **Next**.

4. There is no need to add tags. Select **Next**.

5. Define the Security Group by selecting the previously defined security group.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description
sg-0dcc6656aeeb6c9cd	default	default VPC security group

6. Select **Review and Launch > Launch**.

7. For the key pair, you can reuse the previously generated key pair.

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

Ic-keypair

I acknowledge that I have access to the selected private key file (Ic-keypair.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

8. Select **Launch Instances**.

Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: i-0cbcf768f02edab2e [View launch log](#)

i Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amo

How to connect to your instances

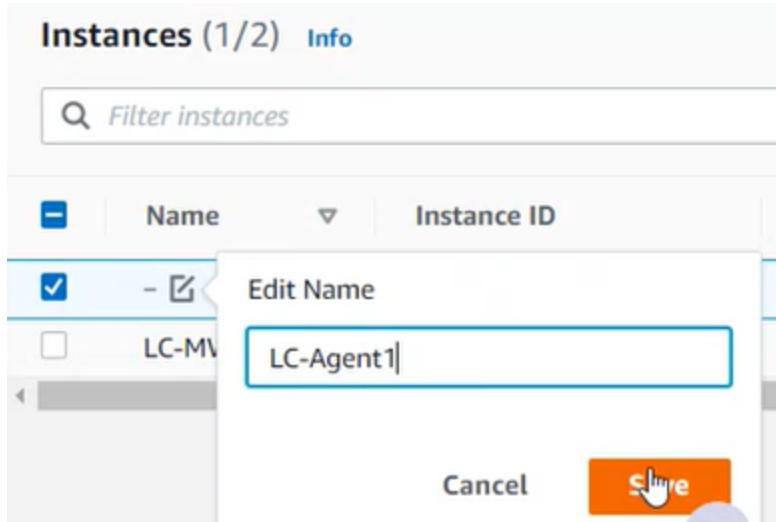
Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready to terminate your instances.

9. On the available instances, you should see two running instances:

- The one containing the Middleware.
- The new one, containing the Agent.

Define a name for the test agent instance (by default, there is no name).

In this example we will name it **LC-Agent1**.

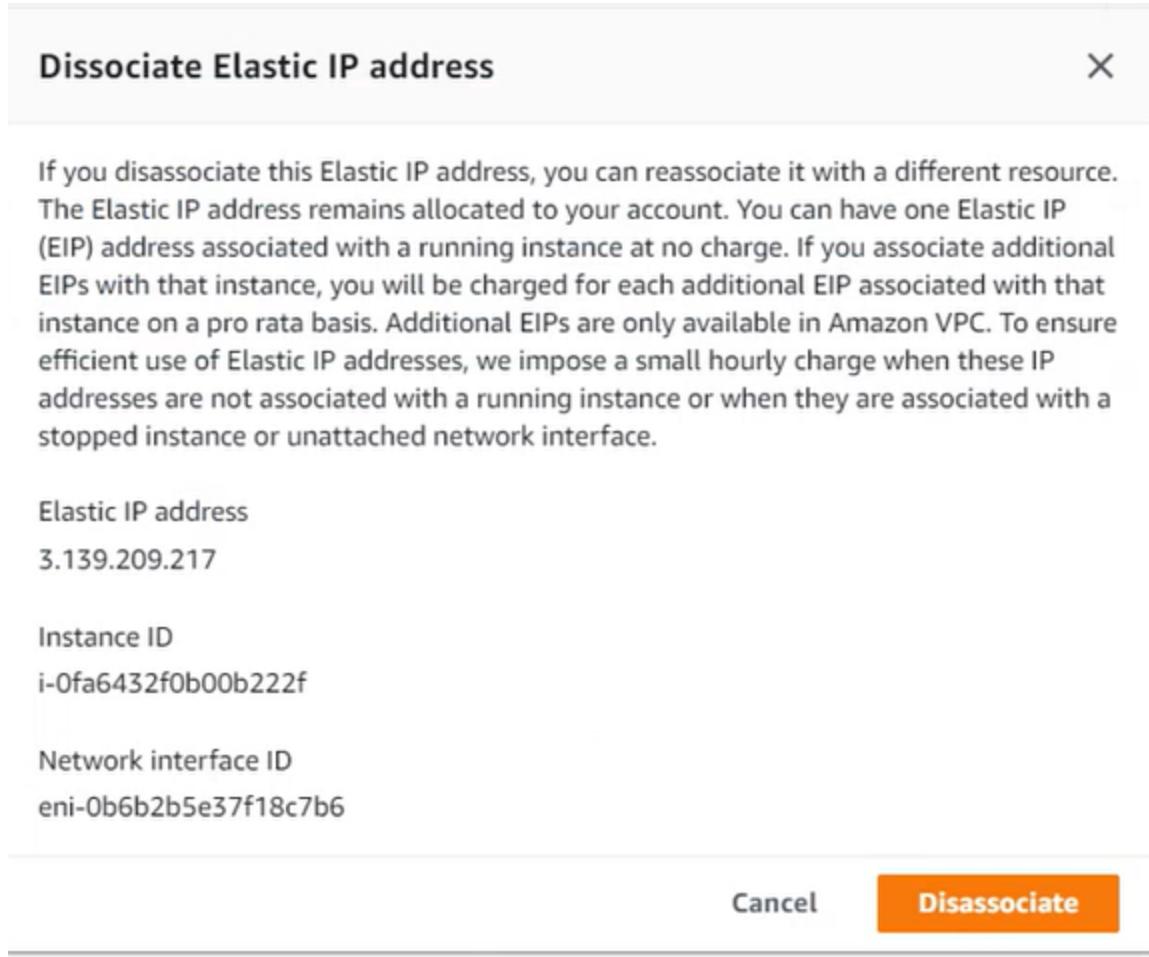


10. In order to access the agent, we need to assign an elastic IP.

There are two options here: assign a new one or reuse the previous one from Middleware. In this example we will reuse the previous one:

- Select **Elastic IP > Action > Disassociate.**

Name	Allocated IPv4 address	Type	Allocation ID	A	Action
3.139.209.217	Public IP	eipalloc-05eb8b223d05e8983			Associate Elastic IP address Disassociate Elastic IP address



- The Elastic IP was disassociated and now should be reassociation to the Management interface of the test agent.

In the previous image deployment we associated the IP address to the Middleware instance, but here, since we have two network interfaces, we need to associate the Elastic IP address to the management network interface.

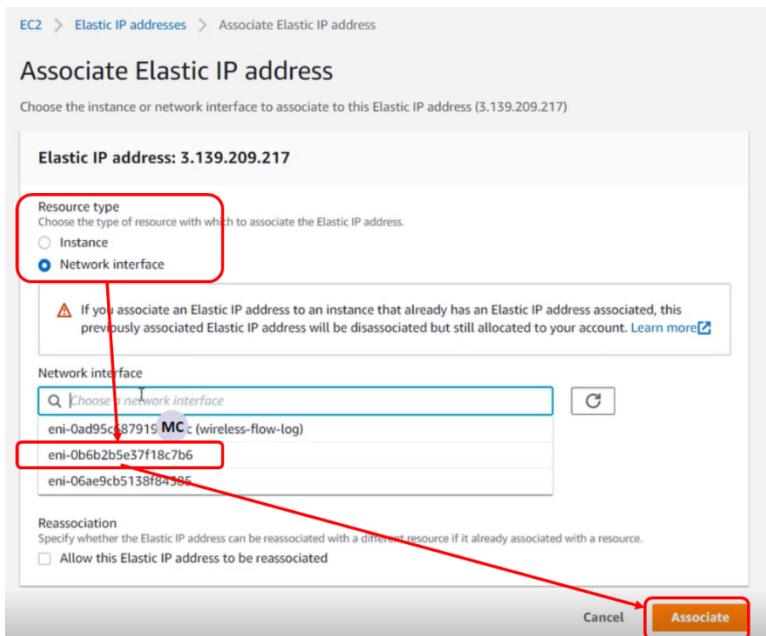
Select:

- Resource type: network interface.**

Network interface: the id that corresponds to the management subnet.

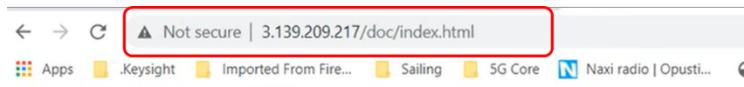
Subnets (2) [Info](#)

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	mgmt-subnet	subnet-0a87408455ab56c3e	<input checked="" type="radio"/> Available	vpc-0f83e58ca0488394d lo...	10.0.0.0/24
<input type="checkbox"/>	test-subnet	subnet-04267cc8ce65eb610	<input checked="" type="radio"/> Available	vpc-0f83e58ca0488394d lo...	10.0.10.0/24



- **Associate.**

Now you can open a web browser and check if the agent Web UI interface is accessible on the elastic IP address:



5G Core Test Engine REST API DAS3

5G Core Test Engine REST API

[License](#)

› **Applications**

› **Traffic**

› **Capture**

11. Connect the test Agent to the Middleware.

This process is the standard one, as in standalone deployments:

- Using an SSH client we can connect to the agent using public key authentication:
`ssh -i <private-key> loadcore@<Elastic_IP_address>`
- Run `agent-setup.sh` script:
`sudo ./agent-setup.sh`
 OR
`sudo /home/ixia/agent-setup.sh` – this is used for LoadCore 2.1
- For Middleware IP address, you will specify the private IP address that is assigned for the Middleware instance, from management subnet.

- For management interface you will select the agent management interface (for example, `eth0`).
- Do you want to allow this agent to be rebooted from the UI? [y/n]: **y**.

12. This completes test Agent deployment procedure.

You can repeat the above steps if multiple Agents should be deployed.

In this moment, the test setup can perform stateless UDP testing.

For application traffic, the agent instance type should be changed, and raw sockets should be configured (this is covered in the next chapter).

LoadCore Agent configuration for Application Traffic

Application traffic in AWS is done using ixStack over Raw Sockets.

In order to run Application traffic with LoadCore, AWS EC2 provides enhanced networking capabilities through two types of mechanisms:

1. Elastic Network Adapter (ENA) which uses `ena` driver
2. Intel 82599 VF interface which uses the Intel `ixgbevf` driver

For running Application traffic, LoadCore supports the following instances types:

- c5 family which enables access to ENA
- m5 family which enables access to ENA
- c4 family which enables access to Intel 82599 VF interface (`ixgbevf` driver)

Both c5 and m5 can be used without any other changes on the LoadCore Agent instances.

IMPORTANT Please make sure you enable the SRIOV option on each simulated 5G node in **LoadCore UI Agent Assignment > Network Management** window.

C4 family will require a special parameter to be configured on LoadCore Agent instances in order to take advantage of Intel 82599 VF interface. This parameter is only available using AWS CLI. Therefore, you will need to install AWS CLI on your computer.

For other information regarding the Intel 82599 VF interface, refer to:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html>

AWS CLI Installation guidelines

The full details can be found on AWS support page:

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>

Download and run the MSI installer:

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>

You will need admin/elevated permission to install it. This is a straight forward process (**Next > Next > ..Finish**).

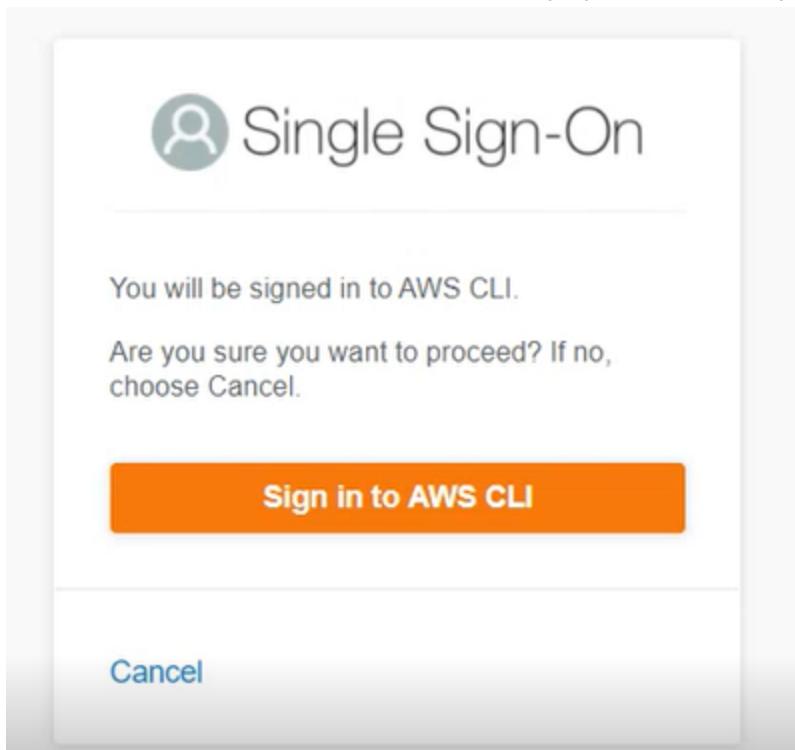
Use SSO to connect AWS CLI to your account

In order to use SSO to connect AWS CLI to your account, AWS SSO needs to be enabled on your account. For more details, refer to:

<https://docs.aws.amazon.com/singlesignon/latest/userguide/step1.html>

Open a CLI terminal and configure SSO, as follows:

1. **AWS configure SSO**
2. SSO start URL: <your SSO link> (e.g. <https://keysight.awsapps.com/start>)
You need to replace <your SSO link> with the SSO link used by your organization to connect to AWS.
3. SSO Region: <SSO instance region> (e.g. **us-east-1**)
You need to check the region where your AWS accounts belong to.
NOTE If an *Invalid grant provided* error is displayed, it usually means that you are using an incorrect SSO Region. For more details, refer to:
<https://github.com/aws/aws-cli/issues/5058>.
4. Press **Enter**. The browser will automatically open and drives you to AWS SSO:



5. Select **Sign to AWS CLI**.
6. On the terminal CLI you will see the available accounts. Select the one you are using:

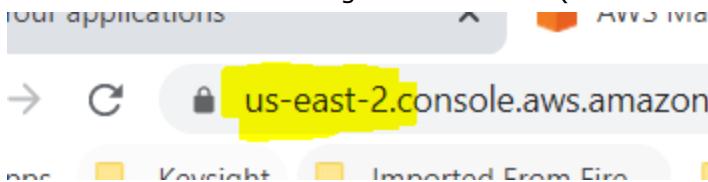
```
C:\ Command Prompt - aws configure sso
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\cipmatei>aws configure sso
SSO start URL [None]: https://keysight.awsapps.com/start#
SSO Region [None]: us-east-1
Attempting to automatically open the SSO authorization page in your default browser.
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:
https://device.sso.us-east-1.amazonaws.com/

Then enter the code:

QPBL-QWPQ
There are 2 AWS accounts available to you.
  keysight-aws-sbx-calabasas, aws-sbx-calabasas.pdl-it-cloud@keysight.com (329217284442)
> keysight-aws-demo-loadcore, aws-demo-loadcore.pdl-it-cloud@keysight.com (672779868767)
```

7. Select CLI default client region: **us-east-2** (this can be seen in the browser link):



8. Set the output format to **.json**.
9. CLI profile name: press **Enter**.

```
C:\ Command Prompt
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\cipmatei>aws configure sso
SSO start URL [None]: https://keysight.awsapps.com/start#
SSO Region [None]: us-east-1
Attempting to automatically open the SSO authorization page in your default browser.
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:
https://device.sso.us-east-1.amazonaws.com/

Then enter the code:

QPBL-QWPQ
There are 2 AWS accounts available to you.
Using the account ID 672779868767
The only role available to you is: AdministratorAccess
Using the role name "AdministratorAccess"
CLI default client Region [None]: us-east-2
CLI default output format [None]: json
CLI profile name [AdministratorAccess-672779868767]:
```

To use this profile, specify the profile name using --profile, as shown:

```
aws s3 ls --profile AdministratorAccess-672779868767
```

C:\Users\cipmatei>

Now you are logged on AWS using your CLI terminal.

Use Access key ID and secret access key to connect AWS CLI to your account

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE <- your AWS Access Key ID
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY <- your AWS
```

```
Secret Access Key
Default region name [None]: us-west-2
Default output format [None]: json
```

For details on how to get your **AWS Access Key ID** and **AWS Secret Access Key**, refer to the AWS documentation:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

For more information on how AWS credentials work, refer to:

<https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html>

To see how to create a new secret access key and for more details, refer to the AWS CLI documentation:

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>

Configure SRIOV support

Now we will configure the SRIOV support. Multiple details can be found on AWS documentation:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html>

- Check if our test client instance has SRIOV support:

On CLI terminal type:

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

Replace the **instance_id** parameter with the instance id of the test agent.

Instances (1/3) Info		
<input type="button" value="Filter instances"/>		
	Name	Instance ID
<input checked="" type="checkbox"/>	LC-Agent1	i-0fa6432f0b00b222f

If you have multiple accounts, you'll need to specify the profile id in the above command. If you have only one account this can be skipped:

```
CLI default client region [None]: us-east-2
CLI default output format [None]: json
CLI profile name [AdministratorAccess-672779868767]:
;
To use this profile, specify the profile name using --profile, as shown:
;
aws s3 ls --profile AdministratorAccess-672779868767
C:\Users\cipmatei>aws ec2 describe-instance-attribute --profile AdministratorAccess-672779868767 --instance-id i-0fa6432f0b00b222f --attribute sriovNetSupport
```

The output of the above command will display the support for SRIOV. If it is empty (like in the picture below), it means the instance type used for the test agent does not have SRIOV and, in this case, we need to modify the instance attribute or the agent should be redeployed selecting a C4 instance type (or another one that has ENA support).

```
aws s3 ls --profile AdministratorAccess-672779868767
C:\Users\cipmatei>aws ec2 describe-instance-attribute --profile AdministratorAccess-672779868767 --instance-id i-0fa6432f0b00b222f --attribute sriovNetSupport
{
    "InstanceId": "i-0fa6432f0b00b222f",
    "SriovNetSupport": {}
}
```

Currently, the driver is **vif** and we need to do some changes and see **ixgbevf** and a **PCI address** (to be used to bind this to raw sockets).

<pre>[ec2-user ~]\$ ethtool -i eth0 driver: vif version: firmware-version: bus-info: vif-0 supports-statistics: yes supports-test: no supports-eeprom-access: no supports-register-dump: no supports-priv-flags: no</pre>	<pre>[ec2-user ~]\$ ethtool -i eth0 driver: ixgbevf version: 4.0.3 firmware-version: N/A bus-info: 0000:00:03.0 supports-statistics: yes supports-test: yes supports-eeprom-access: no supports-register-dump: yes supports-priv-flags: no</pre>
---	--

- We will now modify the instance attribute to support SRIOV.

Stop the test agent instance.

In CLI type the below command:

```
aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

Here you will replace **instance_id** with the test agent instance id (see above). Also, add the profile id in the command (as above).

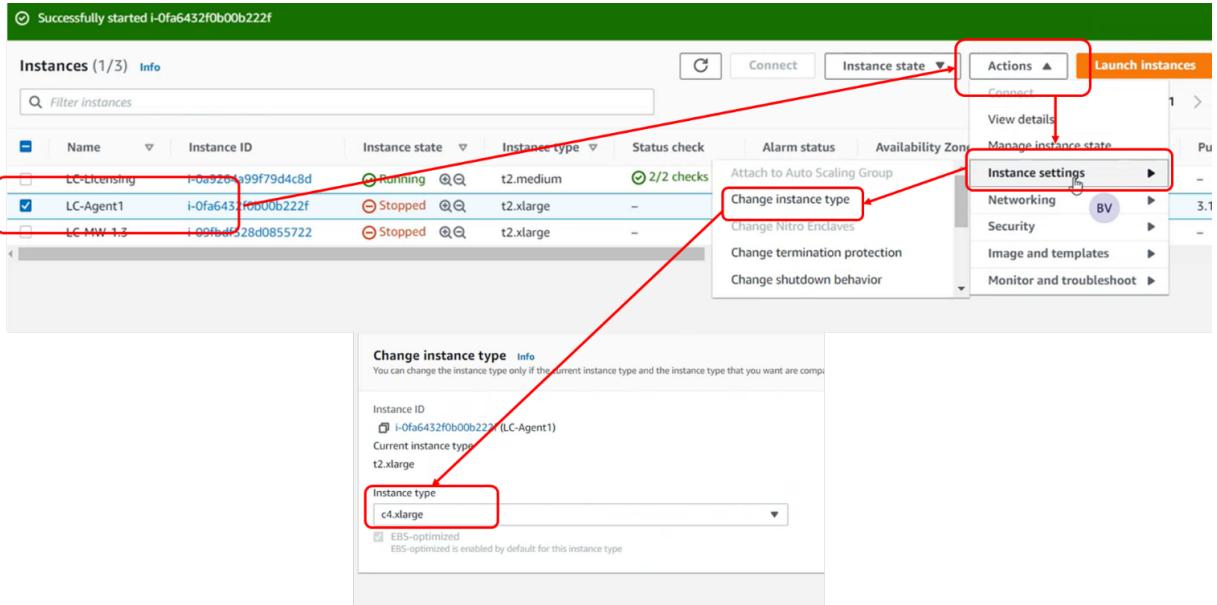
```
C:\Users\cipmatei>aws ec2 modify-instance-attribute --profile AdministratorAccess-672779868767 --instance-id i-0fa6432f0b00b222f --sriov-net-support simple
```

- Make sure you have an Elastic IP associated with the test agent.

- Stop the test agent instance and change the instance type.



- Select **c4.xlarge** type.



- Start the instance.

Now the test agent instance is properly configured, has SRIOV support.

You can check this using SSH on test agent and type:

- ethtool -I eth0 (you should see the ixgbevf driver)

```
ixia@ip-10-0-0-158:~$ ethtool -I eth0
driver: ixgbevf
version: 4.1.0-k
firmware-version:
expansion-rom-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
ixia@ip-10-0-0-158:~$
```

- lspci (you will see the available interfaces)

```
ixia@ip-10-0-0-158:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
00:04.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
00:1f.0 Unassigned class [ff801]: XenSource, Inc. Xen Platform Device (rev 01)
```

Starting with LoadCore 2.1 release, the SRIOV support is enabled per test interface. This can be found in **Agent Assignment > Network Management**. This will activate the usage of the vNIC MAC address for all protocols on that interface.

AGENTS ASSIGNMENT		NETWORK MANAGEMENT						
Order	Agent	Tags	Impairment Profile	Agent Interface	Network Stack	SRIoV	Traffic Capture	Entity
				Name	MAC			
1	10.73.50.65	IxStack: OFF (2) build: 492 (2)	None	ens192	00:0c:29:c0:8b:74	IxStack over Raw Sockets	ON	RAN
								NRF AUSF UDM/HSS PCF

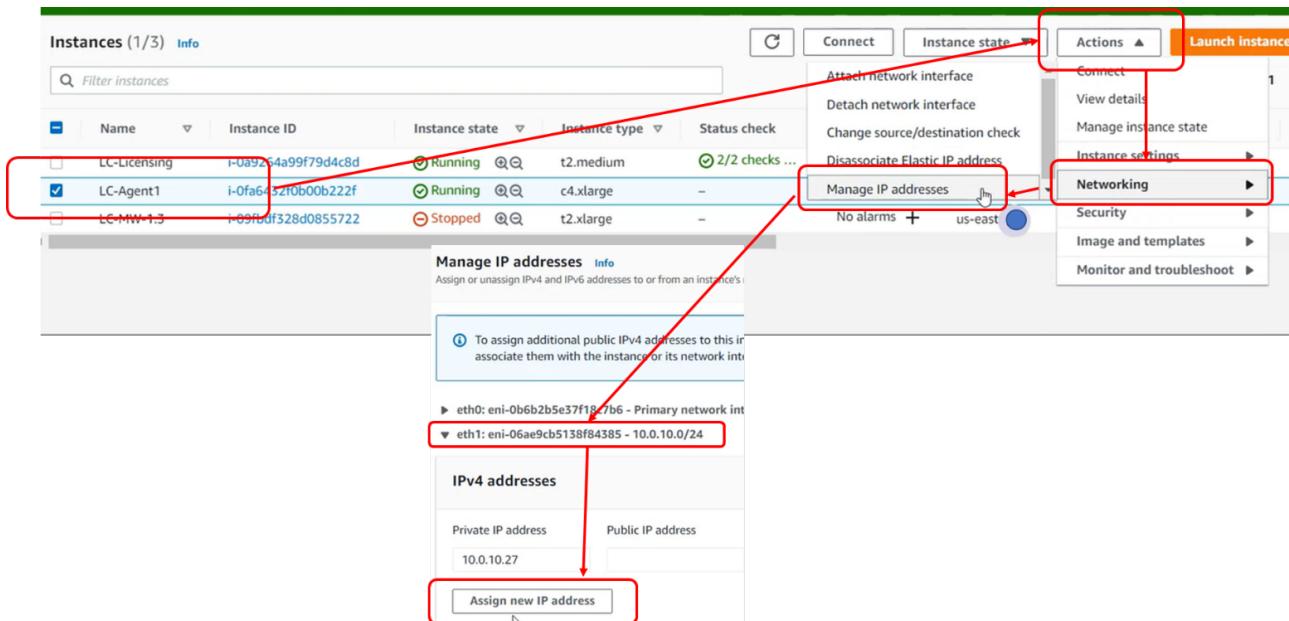
Now the agent is able to run application traffic on eth1.

You can repeat this process for the second agent also you can add multiple interfaces to the test agent.

Add the test IP addresses to the AWS infrastructure

In AWS, in order to allow traffic between test IP addresses, these should be added in AWS infrastructure, on the test agent instance on test interface(s):

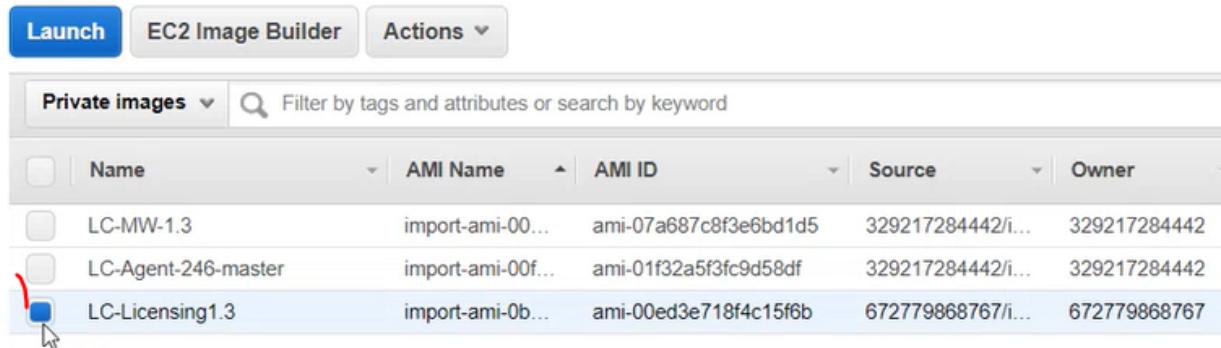
1. Go to **Instance > Networking > Manage IP addresses** > select **eth1** > **Assign new IP addresses**.
2. Add all the IP addresses that are used in the test.



License Server Installation

The installation procedure of the License Server requires the following steps:

1. From the main console, select **Services > EC2 > Images > AMI**.
The browser will display the AMI images shared for your account.
Select the License Server image and select **Launch**.



	Name	AMI Name	AMI ID	Source	Owner
<input type="checkbox"/>	LC-MW-1.3	import-ami-00...	ami-07a687c8f3e6bd1d5	329217284442/i...	329217284442
<input type="checkbox"/>	LC-Agent-246-master	import-ami-00f...	ami-01f32a5f3fc9d58df	329217284442/i...	329217284442
<input checked="" type="checkbox"/>	LC-Licensing1.3	import-ami-0b...	ami-00ed3e718f4c15f6b	672779868767/i...	672779868767

2. Select **t2.medium** for instance type:

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.medium (- ECUs, 2 vCPUs, 2.3 GHz, -, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (i)
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only
<input type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only
<input type="checkbox"/>	t2	t2.small	1	2	EBS only
<input checked="" type="checkbox"/>	t2	t2.medium	2	4	EBS only
<input type="checkbox"/>	t2	t2.large	2	8	EBS only

3. In **Instance details**, the interface should belong to the management network.

Step 3: Configure Instance Details

Number of instances Launch into Auto Scaling Group [\(i\)](#)

Purchasing option [\(i\)](#) Request Spot instances

Network [\(i\)](#) [Create new VPC](#)

No default VPC found. [Create a new default VPC](#).

Subnet [\(i\)](#) [Create new subnet](#)

249 IP Addresses available

Auto-assign Public IP [\(i\)](#)

4. For Security Group, select the one we previously defined:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select

- Assign a security group:
- Create a **new** security group
 - Select an **existing** security group

Security Group ID

sg-0dcc6656aeeb6c9cd

- Leave the other parameters unchanged and launch the instance.

To install the licenses on the License Server, you need to access it using the Web UI.

To do this, you need to reassign the Elastic IP address or define a new Elastic IP and assign it to this server:

- Select **Elastic IP > Action > Disassociate**.

Elastic IP address associated successfully.
Elastic IP address 3.139.209.217 has been associated with network interface eni-0b6b2b5e37f18c7b6

Elastic IP addresses (1/1)					
<input type="button" value="Actions ▲"/> <input type="button" value="Allocate Elastic IP address"/> <small>View details</small> <input type="button" value="Release Elastic IP addresses"/> <input type="button" value="1 > @"/>					
Name	Allocated IPv4 add...	Type	Allocation ID	Action	
3.139.209.217	Public IP	eipalloc-05eb8b223d05e8983	<input type="button" value="Associate Elastic IP address"/>	<input type="button" value="Disassociate Elastic IP address"/>	<input type="button" value="Delete"/>

Elastic Block Store

- Volumes
- Snapshots
- Lifecycle Manager

Network & Security

- Security Groups New
- Elastic IPs New**
- Placement Groups New
- Key Pairs New

Dissociate Elastic IP address

If you disassociate this Elastic IP address, you can reassociate it with a different resource. The Elastic IP address remains allocated to your account. You can have one Elastic IP (EIP) address associated with a running instance at no charge. If you associate additional EIPs with that instance, you will be charged for each additional EIP associated with that instance on a pro rata basis. Additional EIPs are only available in Amazon VPC. To ensure efficient use of Elastic IP addresses, we impose a small hourly charge when these IP addresses are not associated with a running instance or when they are associated with a stopped instance or unattached network interface.

Elastic IP address

3.139.209.217

Instance ID

i-0fa6432f0b00b222f

Network interface ID

eni-0b6b2b5e37f18c7b6

Cancel

Disassociate

- The Elastic IP was disassociated and now should be reassociation to the Instance of the License Server.

Select the following:

- **Resource type: Instance.**
- **Instance id:** select the id of the License server instance.

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (3.139.209.217)

Elastic IP address: 3.139.209.217

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance
 Network interface

⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

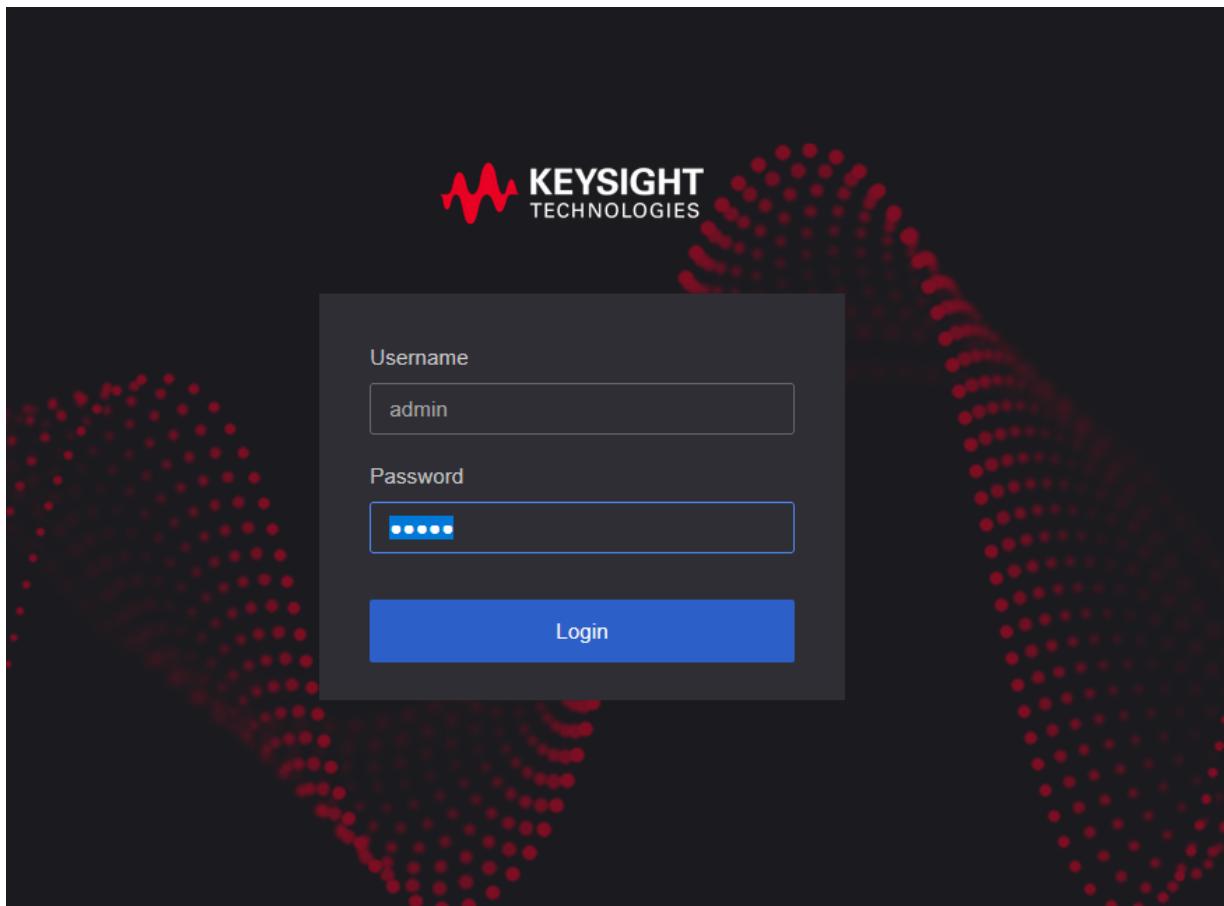
Instance
 X C

Private IP address
The private IP address with which to associate the Elastic IP address.

- Select **Associate**.

Now you can access the License Server UI and install the licenses:

1. On your browser, type the IP address that was assigned for the License Server.
2. Log in using the default username/password: **admin/admin**.



At this point, there are no licenses available on the newly installed server, therefore you must add the license codes you received (evaluation licenses can be obtained by sending an email to: Software-Eval-Request IX <software-eval-request.ix@keysight.com>.

For the licenses activation procedure and related actions, refer to [License Manager](#) section.

Software Upgrades

Middleware Upgrade

You can upgrade the Middleware software as follows:

1. After connecting to the LoadCore setup, open the gear icon in the upper right to display the gear menu.
2. Select **Administration > Software Updates**.
3. Select **Browse** and use the opened window to find the LoadCore upgrade .tar file on the disk.
4. Select **Start Update** to apply the patch and wait for the procedure to end.
5. Access the same menu and check the version of the setup, to make sure that the patch has been applied successfully.

Agent(s) Upgrade

You can upgrade the Agent(s) software as follows:

1. Connect to one of the LoadCore agents.
2. Copy LoadCore-Agent-Update- < version >.tgz to /home/ixia.
3. Run the command:

```
curl -kX PUT https://< agent IP address > /api/v1/update --upload-file /home/ixia/LoadCore-Agent-Update- < version >.tgz
```
4. Using this agent, you will be able to upgrade all agents including the agent that you are connected from.
5. Make sure that the upgrade was successful:

```
curl -kX GET http://< agent IP address > /api/v1/version
```

Expected response:

```
{"revision": "b02c33cb", "tag": "jenkins-lizard-dist-249", "timestamp": "2020-03-20T21:02:13Z"}
```

Optionally, the agents can also be upgraded from Insomnia or any REST agent.

Repeat the steps presented above to deploy as many agents as required.

Troubleshooting

How to recover configuration files when LoadCore UI becomes unresponsive

1. Open a SSH connection to LoadCore Middleware instance using public key authentication.
2. `kubectl exec -it -n keysight-wap wap-db[TAB] /bin/bash [ENTER]`
3. `psql wapdb -U user`
4. `\dt+`
5. `\c wapdb`
6. `SELECT * FROM wireless_sessionconfigs;`
7. `SELECT session_id FROM wireless_sessionconfigs;`
8. `SELECT raw_config FROM wireless_sessionconfigs WHERE session_id='<session-id>';`

How to recover a session that got stuck in starting/stopping/running state

1. Open the REST API Browser from LoadCore UI gear wheel menu (the upper right corner of the LoadCore UI).
2. Go to **Sessions**.
3. Identify your session and click on **Sessions** button (Instance column).
4. Select **test** under selected session on the left side of the view.
5. Select **Edit** and go to status attribute. Change the status to **STOPPED** and select the commit button.
6. Return to LoadCore UI and check the session status.

How to recover a LoadCore Agent that does not respond when trying to run a test from LoadCore UI

1. Reboot the LoadCore Agent.
2. Connect to the LoadCore Agent using SSH.
3. Run the `agent-setup.sh` script: `sudo ./agent_setup.sh`.

4. Provide the required information regarding the interfaces used in the testbed:
 - a. The IP address of the LoadCore Middleware.
 - b. The management interface.
5. Allow the agent to be rebooted from the LoadCore UI : [y].

Monitor the health of the application

Check that all pods are up and running

1. SSH to cluster console.
2. Run `kubectl get pods -A` command.
3. All pods should have running or completed status.

Check the available free space - the free % should be > 15%

1. SSH to cluster console.
2. Run `df -h /` or check events page in WebUI for errors/warnings regarding disk space.
3. The available free disk space % should be > 15%.

Date should be the same on MW and agents

1. Run `date` command on MW and agents. The MW and agents should have the same date/time.
2. If not:
 - check NTP status on MW:
 - SSH to cluster console
 - run `kubectl exec -it -n keysight-wap wap-ntp-server-<id> -- /bin/bash -c "/usr/bin/ntpq -pn"`
 - check NTP status on agents: `ntpq -pn` (ntp server should have MW IP)

Check agents status

1. Check agents status in gear menu > **Administration > Agent Management**.
2. All agents used in test should be online.

Backup and recovery

AWS provides tools that offers the possibility to backup the EBS volumes.

For more details, please refer to:

<https://aws.amazon.com/getting-started/hands-on/amazon-ebs-backup-and-restore-using-aws-backup/>

AWS Security Best Practices

IAM Service

AWS provides a set of security guidelines to help secure your resources. For more details, please refer to:

- <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

Azure Cloud Deployment

The requirements for deploying 5G LoadCore appliances in Azure cloud are:

1. The existence of a resource group.
 - a. To create a resource group, select **Resource Groups** from the Azure portal.
 - b. Select **Add** and enter the property values.
 - c. Type a name and select a location.
 - d. Select **Review + create** to review the values and then select **Create**.
2. The existence of a storage account named *keysight*.
 - a. On the Azure portal, select **Resource Groups**.
 - b. Select the created resource group.
 - c. On the resource group window, select **+ Add**.
 - d. In the list of resources, type **Storage account**. On the storage account window, select **Create**.
 - e. Enter the *keysight* name for your storage account and select the location that should be the same as the resource group location.
 - f. Select **Review + create** to review the storage account settings and then select **Create**.
3. The existence of a container named *keysight* in the *keysight* storage account
 - a. On the Azure portal, select **Resource Groups**.
 - b. Select the created resource group.
 - c. On the resource group window, select the *keysight* storage account.
 - d. In the left menu for storage account, scroll to the Blob service section and select **Containers**.
 - e. Select the **+ Container** button.
 - f. Type the *keysight* name for the new container.
 - g. Select **Create**.
4. The existence of vhds in the *keysight* container.
 - a. In the Azure portal, navigate to the container you created and select the *keysight* container.
 - b. Select the **Upload** button to open the upload blade and browse your local file system to find the mw vhd with the prefix *LoadCore-MW*. Select the **Upload** button to upload the blob.
 - c. The previous point apply for the agent vhd with the prefix *LoadCore-Agent* and the licensing vhd with the prefix *licensing*.
5. The existence of templates in Azure portal.
 - a. To add templates, select *Templates* from the Azure portal.
 - b. On the Templates window, select **+ Add**.
 - c. Type a name for the first template, add a description and then select **OK**.
 - d. Add the template content and then select **OK**.

- e. Select **Add**.
- f. The previous steps apply also for the rest of templates.

Appendix A Enable PCI Passthrough for KVM deployments

PASSTHROUGH MODE ON HOST LINUX

Add the host NIC directly (passthrough) to the VM machine using the following procedure:

- **On HOST machine:**

1. Enable passthrough for PCI devices on the **host** machine by adding `iommu=1` to the `/etc/default/grub` file on the `GRUB_CMDLINE_LINUX_DEFAULT` line:
`GRUB_CMDLINE_LINUX_DEFAULT="quiet splash iommu=1 intel_iommu=on"`
2. Update grub:
`sudo update-grub`
3. Reboot the host machine.
4. Map the host's CPU cores to the VM machines to match NUMA node0 CPU(s) assigned to the Network interface:
 - `lscpu` to get CPUs configuration (for example, NUMA node0 CPU(s): 0-17,36-53)
 - Map CPUs using the command:
`virsh: for i in {0..15}; do sudo virsh vcpupin --config TL150 $i $i; done`
 In the example above `TL150` is the VM name obtained with `virsh list -all`. The **VM Id** from the same list could be used instead of name.
 Refer also to `virsh vcpupin --help`.

5. Ensure performance mode for each CPU (default could be **powersave** mode):

```
sudo echo performance >/sys/devices/system/cpu/cpu$i/cpufreq/scaling_governor
```

Or script for all targeted CPUs:

```
for i in {0..15}; sudo echo performance>/sys/devices/system/cpu/cpu$i/cpufreq/scaling_governor ; done
```

6. Verify and modify the CPU speed:

```
/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq  
/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
```

NOTE

For **virsh** commands refer to
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/virtualization/ch33s08.

- **On VM:**

1. On the VM add a new hardware PCI selecting the PCI address of targeted host network interface (find it using `ethtool -i interface_name` on host machine). The VM will see directly that Interface NIC (see the same MAC on VM). In the same time the host machine will lose that network interface (not seen any more on `ip -a addr`).

2. Depending on the Network Interface Type (Intel, Mellanox, VMWare -vmxnet3 driver) and the number of cores configured on the agent you will need to configure the values for RAM and the huge pages number. The **minimum** requirements for these values can be calculated using this online calculator: <https://loadcore.htmlsave.net/>. You can add an extra 500MB-1GB to the minimum RAM value.

To configure huge pages number on the agent use the following steps:

- a. `sudo nano /boot/grub/grub.cfg`
- b. `hugepagesz=2M hugepages=xyz ro quiet` - where the huge pages value is the one computed by the above calculator
- c. reboot the agent to enforce the changes

After following the above steps, the agent can now be used in Application Data scenarios. You will need a minimum of two agents to run Application Data scenarios.

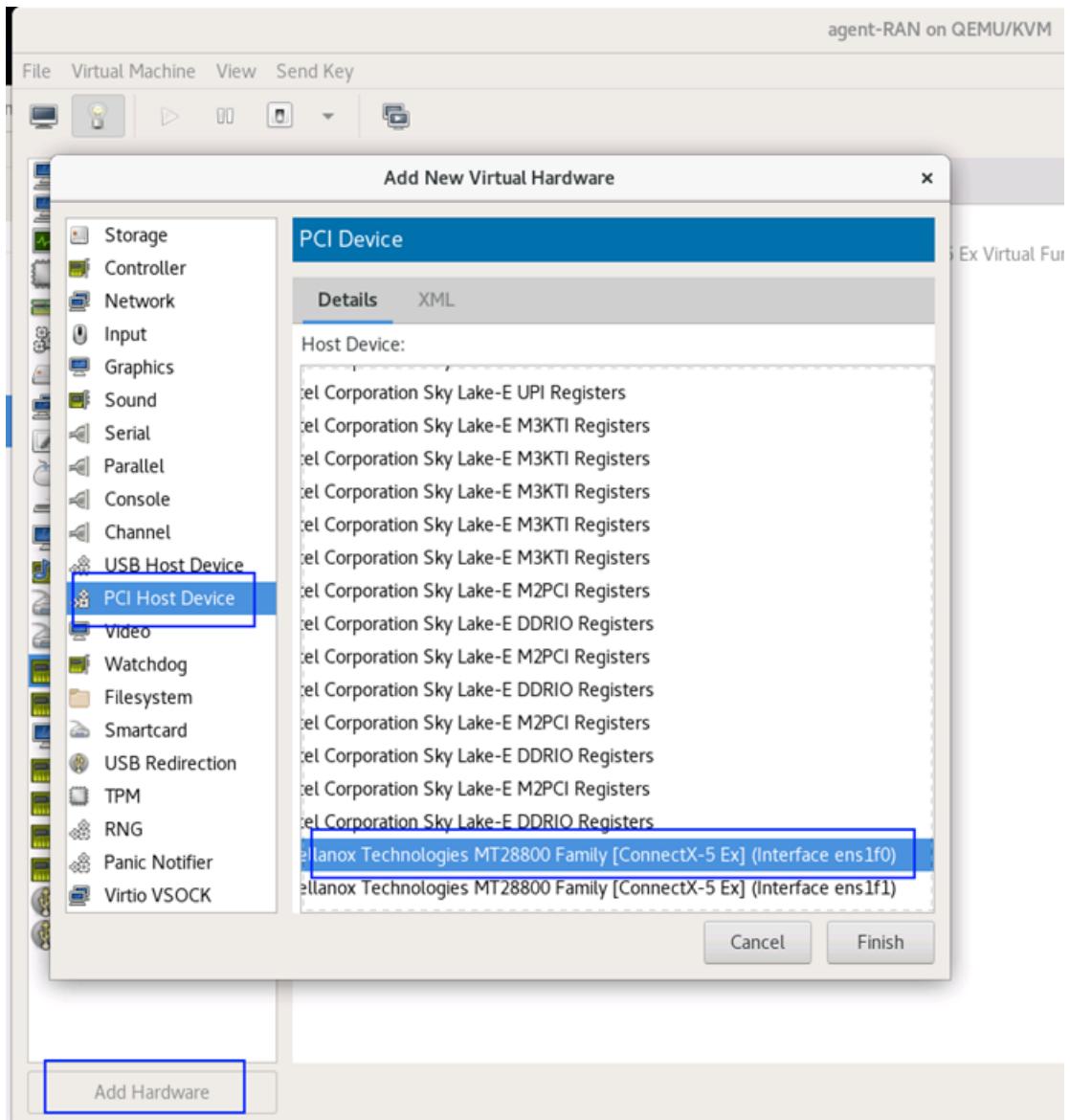
NOTE

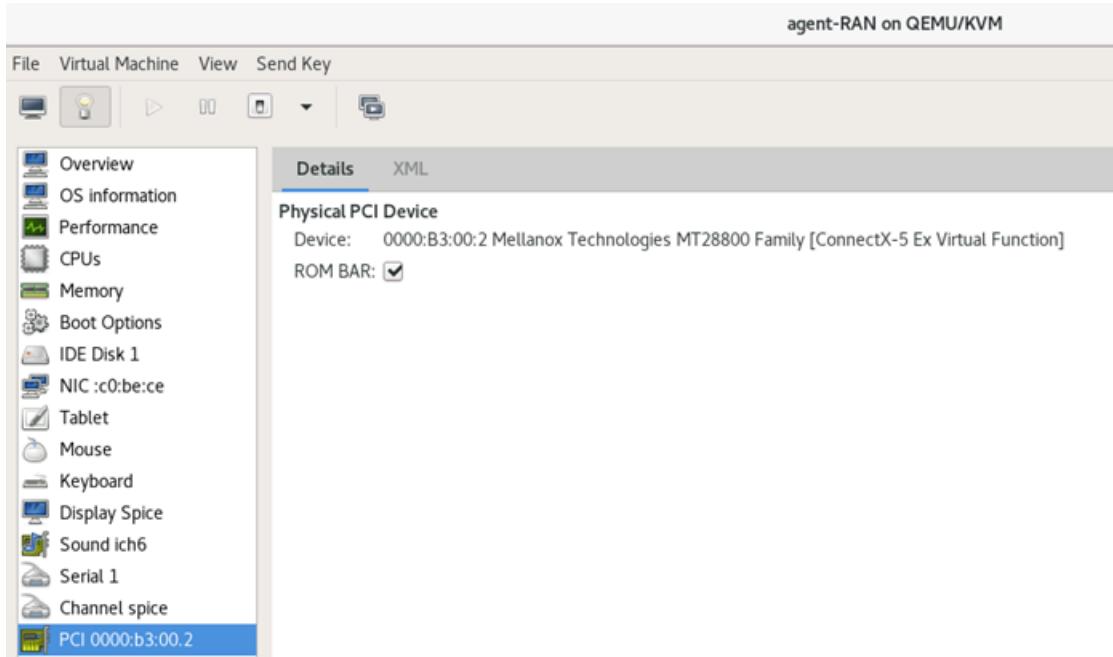
The maximum value for the huge pages number can be 8192. If the requirements surpasses this value, we recommend increasing the number of agents.

KVM VM PASSTHROUGH MODE

Dedicated physical Network Interface using Passthrough Source mode for selected physical interface.

First a **PCI Host Device** must be added and after that the corresponding **Device** (corresponding to that NIC). In this way the interface will have the NIC's driver.





Appendix B Deploying Loadcore with static IP Addresses

Middleware

After the Middleware VM has been installed (the normal procedure installation for ESXi or KVM) you can log on the VM console for the deployed machine (for example, the console from ESXi hypervisor).

You need to log in with console credentials. Logging in with console means log in as **console** first (no password required) and then log in as **admin** with password **admin** (all lowercase letters). This will give you access to a predefined menu from where you can configure the networking settings.

The IP address can be set with the command `kcos networking ip set` like in the following example:

```
kcos networking ip set ens32 10.38.50.100/24 10.38.50.1
```

`ens32` can be replaced with the interface you want to use for the management, `10.38.50.100/24` is the IP and subnet of the Middleware and `10.38.50.1` is the gateway's IP.

NOTE The gateway is mandatory and IPv6 can also be used.

NTP is highly recommended to be configured in static IP environments. The default NTP for LoadCore Middleware is `ntp.ubuntu.com`. If you are using a local or another NTP server it is best to change it with `kcos date-time ntp-servers set` (it should also be the same as the one set in ESX), for example:

```
kcos networking date-time ntp-servers set 10.38.50.50
```

The `kcos` commands also allow the configuration of other settings, including DNS, and has good help menu, with examples, that can be found with `-h` parameter (at the end of the command).

IMPORTANT

Do not assign IP addresses from 10.32.0.xxx range as it is used by the internal cluster and will create conflicts.

Agents

After the agent VM has been installed (the normal procedure for ESXi or KVM) you can log on the VM console for the deployed machine (for example, the console from ESXi hypervisor) with username **ixia** and password **ixia** (all lowercase letters).

The IP can be changed by modifying the netplan:

```
sudo nano /etc/netplan/network.yaml
```

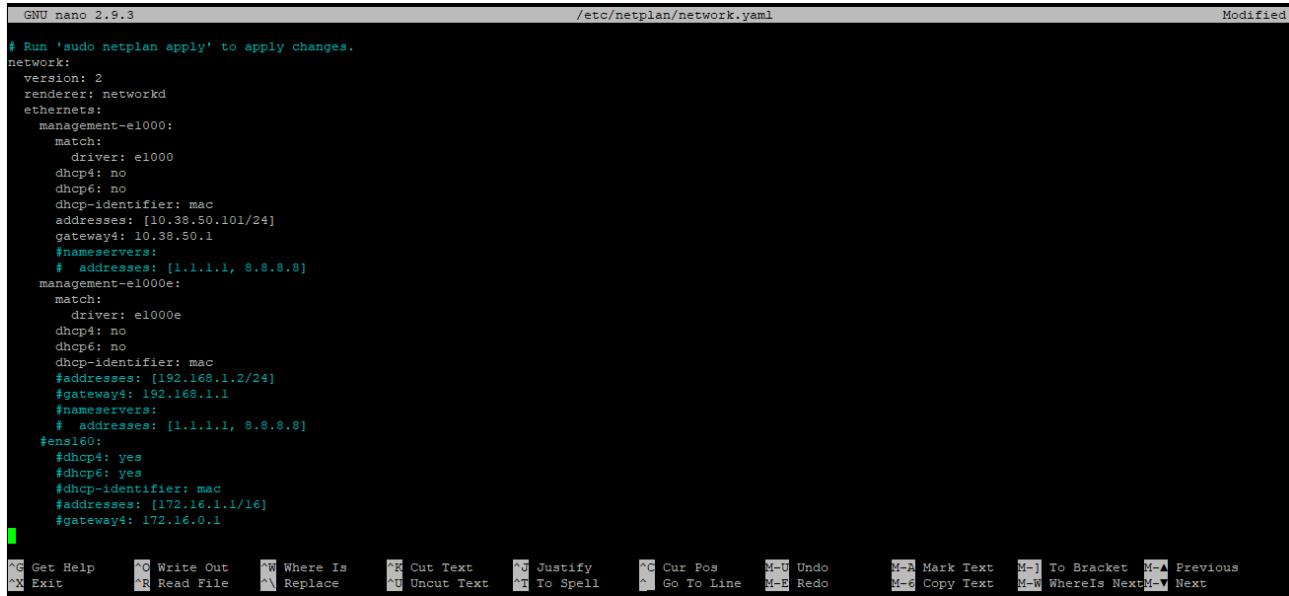
By default, it will display the following:

```
GNU nano 2.9.3                               /etc/netplan/network.yaml

Run 'sudo netplan apply' to apply changes.
network:
  version: 2
  renderer: networkd
  ethernets:
    management-e1000:
      match:
        driver: e1000
      dhcp4: yes
      dhcp6: yes
      dhcp-identifier: mac
      #addresses: [192.168.1.2/24]
      #gateway4: 192.168.1.1
      #nameservers:
      #  addresses: [1.1.1.1, 8.8.8.8]
    management-e1000e:
      match:
        driver: e1000e
      dhcp4: yes
      dhcp6: yes
      dhcp-identifier: mac
      #addresses: [192.168.1.2/24]
      #gateway4: 192.168.1.1
      #nameservers:
      #  addresses: [1.1.1.1, 8.8.8.8]
    #ens160:
    #  dhcp4: yes
    #  dhcp6: yes
    #  dhcp-identifier: mac
    #  addresses: [172.16.1.1/16]
    #  gateway4: 172.16.0.1

^G Get Help   ^Q Write Out   ^W Where Is   ^X Cut Text   ^J Justify   ^C Cur Pos   M-U Undo   M-A Mark Text   M-] To Bracket   ^B Back
^X Exit      ^R Read File   ^\ Replace    ^U Uncut Text  ^T To Spell   ^ Go To Line  M-F Redo   M-6 Copy Text   M-W WhereIs Next  F Forward
```

DHCP must be disabled and an IP address and gateway must be assigned to the interface:



```

GNU nano 2.9.3                               /etc/netplan/network.yaml                         Modified

# Run 'sudo netplan apply' to apply changes.

network:
  version: 2
  renderer: networkd
  ethernets:
    management-e1000:
      match:
        driver: e1000
      dhcp4: no
      dhcp6: no
      dhcp-identifier: mac
      addresses: [10.38.50.101/24]
      gateway4: 10.38.50.1
      #nameservers:
      #  addresses: [1.1.1.1, 8.8.8.8]
    management-e1000e:
      match:
        driver: e1000e
      dhcp4: no
      dhcp6: no
      dhcp-identifier: mac
      #addresses: [192.168.1.2/24]
      #gateway4: 192.168.1.1
      #nameservers:
      #  addresses: [1.1.1.1, 8.8.8.8]
    ens160:
      #dhcp4: yes
      #dhcp6: yes
      #dhcp-identifier: mac
      #addresses: [172.16.1.1/16]
      #gateway4: 172.16.0.1

```

The terminal shows the bottom of the nano editor with various keyboard shortcuts for navigating and editing the file.

After this, save the changes to the `.yaml` file and do:

```
sudo netplan apply
```

If you are unsure of the changes you can also do `sudo netplan try` (before `apply`) so it can be checked that the changes are valid before applying.

Now, you can do the script to bind the agent to the Middleware, same as in the normal deploy, with:

```
sudo ./agent-setup.sh
```

NOTE

Start the NTP service on the agents (usually done when `agent-setup.sh` is run) only after setting the clock/NTP server on the Middleware. Setting the clock on the Middleware after the ntp service started on the agents can lead to it panicking (agent side) on big adjustments on sync. Restarting the ntp agent side (`sudo systemctl restart ntp`) should fix this.

License server

After the License Manager has been deployed (the normal procedure installation for ESXi or KVM) you can log on the VM console for the machine (for example, the console from ESXi hypervisor) with username **netadmin** and password **netadmin** (all lowercase letters). A menu will appear, as follows:

```
*****
*      System Configuration      *
*****  

1. Show Current Configuration.  

2. Show DHCP status.  

3. Set Hostname.  

4. Set IP, Network Mask and default Gateway.  

5. Set DNS.  

6. Set Search Domain.  

7. Enable / Disable DHCP.  

8. Send ping.  

9. Send traceroute.  

10. Exit.
```

Enter your choice (number) :

First, you have to disable DHCP:

```
*****
*      System Configuration      *
*****  

1. Show Current Configuration.  

2. Show DHCP status.  

3. Set Hostname.  

4. Set IP, Network Mask and default Gateway.  

5. Set DNS.  

6. Set Search Domain.  

7. Enable / Disable DHCP.  

8. Send ping.  

9. Send traceroute.  

10. Exit.
```

Enter your choice (number) :
7

DHCP is currently enabled please type:
--> 'yes' to disable it;
--> 'no' and return to the previous menu.
yes_

And then you will have to input the IP of the License Server, the IP of the gateway and the subnet mask:

```

1. Show Current Configuration.
2. Show DHCP status.
3. Set Hostname.
4. Set IP, Network Mask and default Gateway.
5. Set DNS.
6. Set Search Domain.
7. Enable / Disable DHCP.
8. Send ping.
9. Send traceroute.
10. Exit.

Enter your choice (number) :
4

Current IP      : 10.38.153.163
Current Gateway : 10.38.152.1
Current Mask    : 255.255.252.0
Please enter the new IP address or type 'quit' and return to the previous menu.
10.38.50.110
Please enter the new gateway or type 'quit' and return to the previous menu.
10.38.50.1
Please enter the new network mask or type 'quit' and return to the previous menu
.
255.255.255.0

```

Appendix C Ports used in LoadCore communication

The following are ports used by LoadCore that need to be kept open in the network:

Port Number	Used for...
22	SSH service
80	connecting to the browser UI
123	NTP communication between MW and agents
443	ingress, for rest api commands/to connect to webserver
7443	communication between MW and license server
9022	SSH to connect as root
30422	NATS communication between MW and agents
31123	per-UE stats communication between MW and agents
32100	EULA (EULA can also be accepted through console)

CHAPTER 3**LoadCore Deployment on Containers**

This section contains the following topics:

Amazon AWS EKS Deployment	74
OpenShift Deployment	77
LoadCore Deployment on k8s	79
LoadCore Deployment on GKE Anthos	80

Amazon AWS EKS Deployment

This section describes the steps need for LoadCore deployment in AWS EKS.

The deployment procedure requires a LoadCore tool named `loadcore-kube-setup` to upload the images and install the product.

How to get an authentication token to be able to push the images into ECR

In order to push the images into the Registry, you will need a token. This procedure uses the AWS account provided by Keysight to authenticate on AWS by using the `aws cli` tool.

To install the `aws cli` tool, please refer to

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>.

To log into AWS using `aws cli`, please refer to the [Use SSO to connect AWS CLI to your account](#) section.

After the authentication is successful, you can use the following command to get the authentication token:

```
aws ecr get-login-password --region us-east-1
```

AWS Docs:

- <https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html>
- <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-sso.html>
- <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>

How to push LoadCore container images into AWS EKS

This step can be done from a Linux machine.

Before pushing the images, you need to create a repository for each Loadcore image. There is no support for this in `loadcore-kube-setup` tool until now, so it needs to be done manually by using a bash command.

https://docs.amazonaws.cn/en_us/AmazonECR/latest/userguide/repository-create.html

<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecr/create-repository.html>

After creating all the repositories, we can use the `loadcore-kube-setup` tool to upload the images.

Download the LoadCore k8s build and extract it.

Modify the `load-core.yaml` file as follows:

```
privateRegistry:  
  name: <the-name-of-ECR-registry>  
  storePrivateImages: true  
  auth:  
    user: AWS  
    password: <the-token-from-step-1>
```

After modifying the `load-core.yaml` file, you can upload the images by using the following command:

```
./loadcore-kube-setup upload-images
```

How to install LoadCore Middleware in EKS

In case the EKS cluster exists, you will need to have the `kubeconfig` file on the same machine where you are running `loadcore-kube-setup` tool because it will use it to create all the resources needed by LoadCore.

In case you do not have any cluster created, you can use the following links:

- <https://docs.aws.amazon.com/eks/latest/userguide/create-cluster.html>
- <https://docs.aws.amazon.com/eks/latest/userguide/create-managed-node-group.html>

The requirements for the EKS cluster workers are as follows:

- they need use Ubuntu as operating system because the SCTP module is not available on Amazon images.
- the instance type should have at least 8 vCPUs and 32 GB RAM.

To create the `kubeconfig` file, please refer to

<https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html>.

After generating the `kubeconfig` file, make sure to add the path to it into `load-core.yaml` file on `kubeConfigPath` attribute. The default path is `~/.kube/config`.

Before installing the LoadCore Middleware, make sure that you have access to the EKS cluster by using any `kubectl` command: `kubectl get nodes` or `kubectl get pods -A`.

You will need to modify the storage class inside the `load-core.yaml` file (in this case `gp2`).

```
storage:  
  class: gp2
```

The environment attribute inside `load-core.yaml` file should be set to: `environment: onPremise`.

After verifying the access to the EKS cluster, use the following command to install the LoadCore Middleware:

```
./loadcore-kube-setup install-mw
```

NOTE The installation will take at least 10 minutes depending on workers instance type.

Use the following command to get the public IP address of the LoadCore UI:

```
kubectl get svc -n kcos-framework kcos-ingress-v1-ingress-nginx-controller -o json
| jq '.status.loadBalancer.ingress[0].ip'
```

or

kubectl get svc -A and search on External IP column

How to install LoadCore Agents on EKS

To run tests in Loadcore, the Agents use `multus CNI` for separating the management traffic from test traffic.

The `multus CNI` needs to be installed on the EKS cluster before deploying the LoadCore Agents:

- Install `multus CNI` using the following command:

```
kubectl apply -f https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/master/config/multus/v3.7.2-eksbuild.1/aws-k8s-multus.yaml
```

- Check the `multus` installation by using the command:

```
kubectl get pods -n kube-system
```

Each node should have one pod called `kube-multus-ds`.

Some changes need to be done in the `load-core.yaml` file regarding the additional testing interface for the agents. You can use the following `config` example. The `master` attribute from below can vary on each environment. This is the name of the worker interface. The `type` attribute can be `ipvlan` or `macvlan`.

```
config: |
  {
    "cniVersion": "0.3.1",
    "type": "ipvlan",
    "master": "ens7",
    "mode": "12",
    "ipam": {
      "type": "host-local",
      "subnet": "10.0.100.0/24",
      "rangeStart": "10.0.100.205",
      "rangeEnd": "10.0.100.215"
    }
  }
```

Multus CNI examples:

- <https://github.com/k8snetworkplumbingwg/multus-cni/tree/master/examples>.

In case you need multiple agents, you can play with the `replicas` attribute from the file (`replicas: 1`).

After changing the configuration, you can use the following command to install the agents:

```
./loadcore-kube-setup install-agent
```

Using the default configuration, the tests will work only if the agents are deployed on the same worker.

In case you need a more complex scenario, AWS released a guide on how to use `multus` in EKS:

- <https://aws.amazon.com/blogs/containers/amazon-eks-now-supports-multus-cni/>

How to uninstall LoadCore components

To uninstall agents:

```
./loadcore-kube-setup uninstall-agent
```

To uninstall Middleware:

```
./loadcore-kube-setup uninstall-mw
```

OpenShift Deployment

The configuration required to set up a LoadCore Agent in OpenShift:

1. After connecting to the proxy, export the `KUBECONFIG` env var:

```
%> export KUBECONFIG=~/ocp-install/auth/kubeconfig
```

For this environment, it was added to the `.bashrc` file.

2. Login user the `kubeadmin` user:

```
%> oc login -u kubeadmin -p BGwxz-k9GNo-hYoED-Tqp6z  
https://api.lab.ocp.lan:6443
```

3. Make sure that SCTP is enabled on the workers. For this, execute:

```
%> oc create -f load-sctp-module.yaml
```

More details about this topic here:

<https://docs.openshift.com/containerplatform/4.5/networking/using-sctp.html>

4. Create an OpenShift project named `loadcore-root`. The command used is:

```
%> oc new-project loadcore-root --description="LoadCore project" --display-name="loadcore-root"
```

5. Configure OpenShift to allow running containers under root and gave full privileges to the containers:

```
%> oc adm policy add-scc-to-user anyuid -z default -n `oc project -q` --as=system:admin  
%> oc adm policy add-scc-to-user privileged -z default -n `oc project -q` --as=system:admin
```

6. Next, add a default route to the internal image registry and get the registry path into the `REGISTRY` variable:

```
%> oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec":{"defaultRoute":true}}' --type=merge  
%> oc adm policy add-scc-to-user privileged -z default -n `oc project -q` --as=system:admin  
%> echo $REGISTRY  
default-route-openshift-image-registry.apps.lab.ocp.lan
```

7. Log in to the internal image registry using podman:

```
%> podman login --tls-verify=false -u unused -p $(oc whoami -t) ${REGISTRY}
```

8. Load the LoadCore agent image using podman:

```
%> podman load -i LoadCore-Agent-Docker-0.1-15f7cb3ba-20201202T161108Z.tar.gz
```

9. See the loaded image:

```
%> podman images
...
localhost/loadcore-agent 0.1-15f7cb3ba ca0aa01b2c11 6 days ago 608 MB
```

10. Tag the loadcore image using the same image ID returned by podman images:

```
%> podman tag ca0aa01b2c11 $REGISTRY/loadcore-root/loadcore-agent:0.1-15f7cb3ba
```

11. Push the loadcore image to the internal image registry:

```
%> podman push --tls-verify=false $REGISTRY/loadcore-root/loadcore-agent:0.1-15f7cb3ba
```

12. Now, configure an extra interface to be used as test interface. Multus is already configured on OpenShift.

Here, a multus network named `loadcore-root-network` is added.

```
%> oc edit networks.operator.openshift.io cluster
```

Then, add the following:

```
spec:
```

```
  additionalNetworks:
    - name: loadcore-root-network
      namespace: loadcore-root
      type: Raw
      rawCNICConfig: '{"cniVersion": "0.3.1", "name": "ixia-network", "type": "bridge", "master": "eth0", "ipam": { "type": "whereabouts", "range": "192.168.1.0/24", "exclude": ["192.168.1.0/32", "192.168.1.1/32", "192.168.1.254/32"] }}'
```

13. Get the loadcore image tag inside the registry:

```
%> oc describe is loadcore-agent
```

Observe the image registry tag, it is something like this:

```
image-registry.openshift-image-registry.svc:5000/loadcore-root/loadcore-agent@sha256:718cb38c0c4dd9d71e09a5b05d9edfe462f2ca3c5fe9b0c30e5a1e6f6d83d3fa
```

14. Edit the `loadcore-agent-deployment-loadcore-root.yaml` and specify the above image registry tag:

```
...
```

```
spec:
```

```
  containers:
    - name: loadcore-agent
      loadimage: YOUR-IMAGE-REGISTRY-TAG
```

```
...
```

Also, specify the IP address of your middleware in the container `args` section.

15. Apply the LoadCore agent deployment:

```
%> oc apply -f loadcore-agent-deployment-loadcore-root.yaml
```

16. Take a look at the started LoadCore pods:

```
%> oc get pods
```

17. Get a shell inside a LoadCore agent pod:

```
%> oc exec -it loadcore-agent-65bdc89cf8-58sxt -- bash
```

18. Verify the IP interfaces:

```
%> ip a
```

You should see a `net1` interface, added by multus.

19. Get the list of running processes:

```
%> ps aux
```

Among other processes, you should see the `5GTestEngineService` (lizard) and the portmanager running.

LoadCore Deployment on k8s

The following procedure describes the steps needed in order to deploy LoadCore on k8s:

1. Install `helm` and `helmfile` in ~ directory and make sure they have 755 permissions.

2. Make sure you have `multus` deployed.

In case you do not have it installed, use:

```
kubectl apply -f https://raw.githubusercontent.com/intel/multus-cni/master/deployments/multus-daemonset.yaml
```

3. Create a folder for MW and copy the `.tar` file in that folder. (`.tar` file name is e.g: `LoadCore-MDW-3.0.0-3304-184-AGENT-3.0.0-640`).

- a. untar the `.tar` file:

```
tar -xvf LoadCore-MDW-3.0.0-3304-184-AGENT-3.0.0-640
```

- b. In that directory you will find a file named `load-core.yaml`. In that file you will find all the configurations you will need to add/modify in order to deploy LoadCore.

NOTE

By default, `load-core.yaml` comes as a *NodePort*. This means that your LoadCore application will be reachable at port **30443**. In case you need to connect to port **443**, you have to change the type to **LoadBalancer** and also deploy a load balancer in your cluster.

4. In case you have a private registry, after you added everything in `load-core.yaml`, you will need to push the images into the registry. If you do not have a private registry, you can skip this step.

```
./loadcore-kube-setup upload-images
```

5. In order to install the MW, use the following command:

```
./loadcore-kube-setup install-mw
```

6. In order to install the agent, use the following command:

```
./loadcore-kube-setup install-agent
```

NOTE

In `load-core.yaml` you will find a `replicas` parameter. By default is set to 2. This means 2 agents will be deployed. If you need more agents you will need to change this value.

7. In order to uninstall the agent, use the following command:

```
./loadcore-kube-setup uninstall-agent
```

8. In order to uninstall LoadCore, use the following command:

```
./loadcore-kube-setup uninstall-mw
```

IMPORTANT

T You need to first deploy the MW and after to deploy the agents. When uninstalling, first you need to uninstall the agent and after that uninstall the mw.

NOTE

If you need to deploy LoadCore on Google Anthos, change the environment value from `load-core.yaml` into **gke** (you do not need to use commas).

LoadCore Deployment on GKE Anthos

The following procedure describes the steps needed in order to deploy LoadCore on GKE Anthos:

1. Create an Anthos Sample Deployment cluster.
2. Go to **Kubernetes Engine** and select the Anthos cluster previously created. Go to **Networking** and enable **Network Policy**.
3. Connect to Cloud Shell and run:

```
gcloud services enable containerregistry.googleapis.com (this will enable your container registry)
```

In this case, `eu.gcr.io` container registry will be created. For more details, refer to:
<https://cloud.google.com/container-registry/docs/overview>

In `load-core.yaml`, which will appear after you do step 5, you have to type at `privateRegistry/name: {hostname}/{project-id}` as per <https://cloud.google.com/container-registry/docs/overview>.

After you push your first image to your `privateRegistry`, in Cloud Storage you will find your artifactory created where all your images will be held.

Steps to push an image:

- docker pull hello-world
- docker tag hello-world:latest eu.gcr.io/{project-id}/hello-world:ABC
- docker push eu.gcr.io/{project-id}/hello-world:ABC

4. Go to **Cloud Storage** and create a bucket where you will upload your LoadCore `tar.gz` image.
5. Connect to Cloud Shell and download the LoadCore `tar.gz` images as follows:

a. `cd /root` -> we are going to root directory as this is the only one who has enough space. Be careful that after some time this directory is refreshed automatically.

b. Download the LoadCore image to `/root` directory using:

```
gsutil cp gs://{{bucket name}}/{{image name}}
```

For example:

```
gsutil cp gs://anthos-sample-bucket/LoadCore-MDW-3.0.0-3304-184-AGENT-3.0.0-640
```

c. Untar LoadCore images using:

```
tar -xvf LoadCore-xxx
```

d. In the `~` directory, create a dir for your LoadCore. In this case, a `mw3.0` folder was created. Move all the files excluding `images.tar` to your created folder.

e. Generate a new key for your private registry with the following command:

```
gcloud iam service-accounts keys create our_key --iam-account="your iam-account" -> as per https://cloud.google.com/iam/docs/creating-managing-service-account-keys#iam-service-account-keys-create-console
```

The information from the file `our_key` has to be copied into `load-core.yaml` at `privateRegistry/auth/password`.

In `load-core.yaml` you also have the `kubeConfigPath`. Change this path to your current `.kube` path.

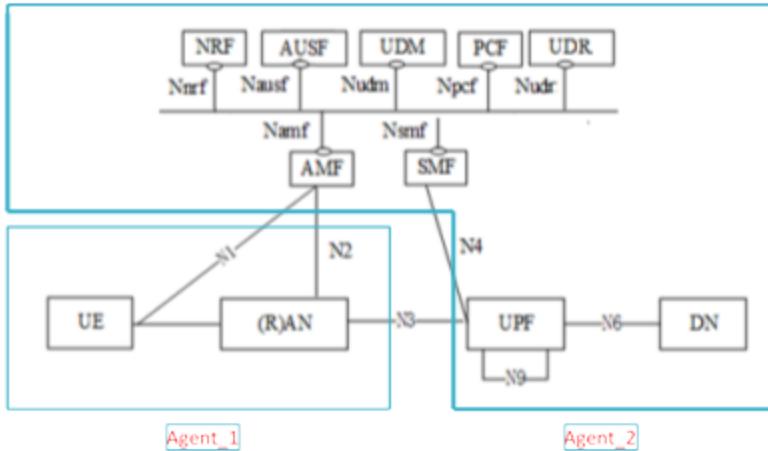
- f. Upload the containers to your private registry:
 - cd /root
 - sudo ~/mw3.0/loadcore-kube-setup upload-images -c ~/mw3.0/load-core.yaml
 6. Install `helm` and `helmfile` in ~ directory and make sure they have 755 permissions,
 7. Add your path to `bashrc`.
For example:

```
export PATH=/home/{}/mw3.0:$PATH
```
 8. Go to **Kubernetes Engine** and select your cluster. Go to **Nodes** and create a new pool of nodes.
In this case, there were created 2 nodes, with Image Type = Ubuntu with Docker, series E2 and machine type = e2-standard-8.
 9. Install the MW:
`./loadcore-kube-setup install-mw`
After the installation is completed, in order to find the LoadCore IP (which will be the ingress ip) you have to:
`kubectl get svc -A -o wide | grep kcos-framework-v1-ingress ->` and take the 2nd IP which is the External IP. This IP will be put in `load-core.yaml` at the management IP to which the agent will connect.
 10. In order to find out the management interface and add it in the `load-core.yaml`, you have to go to **Compute Engine > VM Instances** and connect via SSH (there is a ssh button in the right of each node). If you take the output of `ip a` you will see that the management interface of each node is `ens4`. This means that in `load-core.yaml`, you will have to specify `ens4`.
 11. Install the agent:
 - a. Install multus -> `kubectl apply -f "multus-for-gke.yaml"`
 - b. `./loadcore-kube-setup install-agent`

NOTE In `load-core.yaml` you will find a `replicas` parameter. By default is set to 2. This means 2 agents will be deployed. If you need more agents you will need to change this value.
 12. Uninstall the agent:
`./loadcore-kube-setup uninstall-agent`
 13. Uninstall the MW:
`./loadcore-kube-setup uninstall-mw`
- IMPORTANT** T You need to first deploy the MW and after to deploy the agents. When uninstalling, first you need to uninstall the agent and after that uninstall the mw.
- NOTE** For Anthos, the environment value from `load-core.yaml` has to be **gke** (without commas).
- NOTE** To disable the container registry:
`gcloud services disable containerregistry.googleapis.com --force`
- NOTE** Anthos cluster already has LoadBalancer installed. This means that you will need to set in `load-core.yaml` at type of the cluster **LoadBalancer**.

CHAPTER 4**Test Case #1: gNB and 5G Core simulation in B2B scenario**

The purpose of this test is to emulate the entire 5G network, in a back-to-back topology. LoadCore will be used to simulate UEs and gNBs on one agent and the rest of 5G core elements will be simulated by the second agent. In a real environment, any of these agents can be replaced with a real DUT.



The topology involves two entities:

- One agent (located on VM machine agent_1), simulating the UEs and gNB
- One agent (located on VM machine agent_2), simulating the rest of 5G core elements (NRF, AUSF, UDM, PCF, UDR, AMF, SMF and DN)

A typical test will validate the following aspects on the network:

- Successful UE attach, create security context and PDN session establishment.
- Validate the core network is able to support huge number of UEs.
- Various types and volumes of traffic passing from UEs to DN.
- Extended QoS is assigned per UE, per flow, and per Uplink and Downlink directions.

Test Configuration

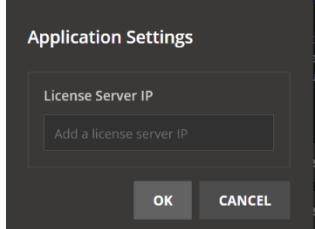
Two options are available for configuring the first LoadCore test:

- Load the Full Core topology from Repository and configure each network element with proper IP addresses, plus traffic flows. This is the option will be described below.
- Load a predefined test configuration that matches the current deployment.

License server configuration

To configure the license server:

1. Select the wheel icon on the top right corner of the Dashboard page. The general settings menu appears.
2. Select **Application settings**. The Application Settings window appears.
3. Add the IP address in the License Server IP field.

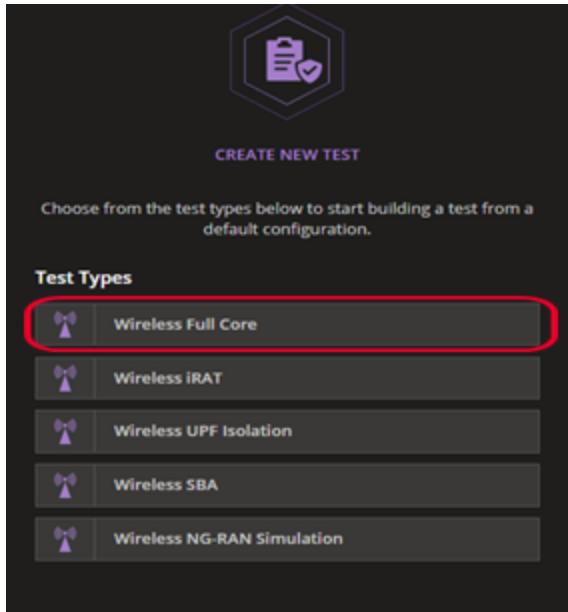


Is this setup the license server IP is: 10.73.53.17 (the IP address the license server got during the VM deployment).

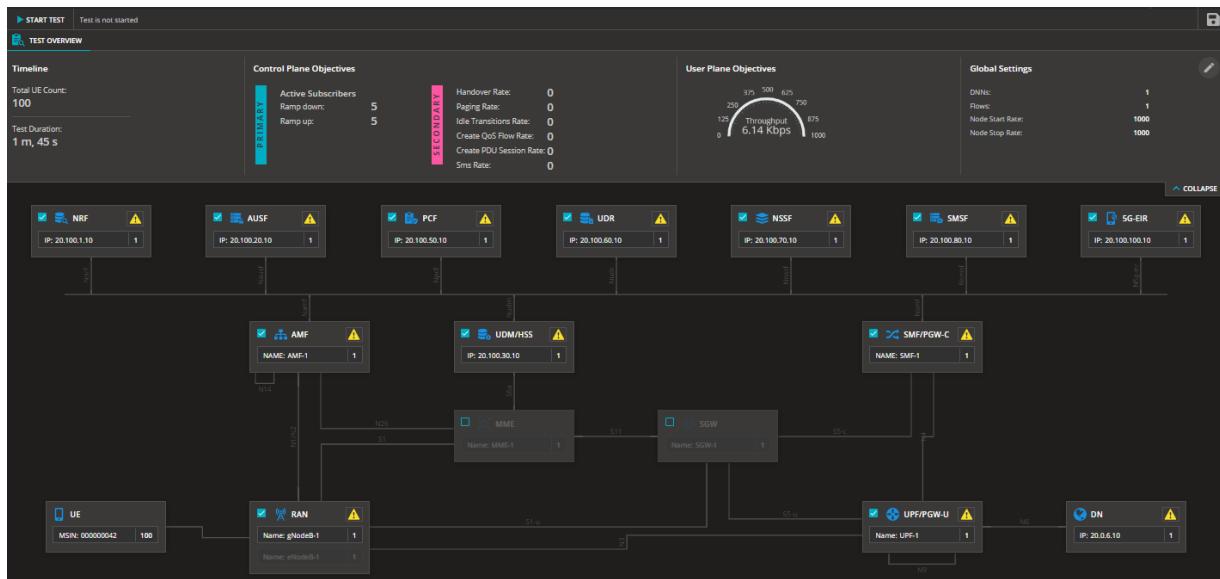
Configure the test from scratch

To configure the test using this method, do the following:

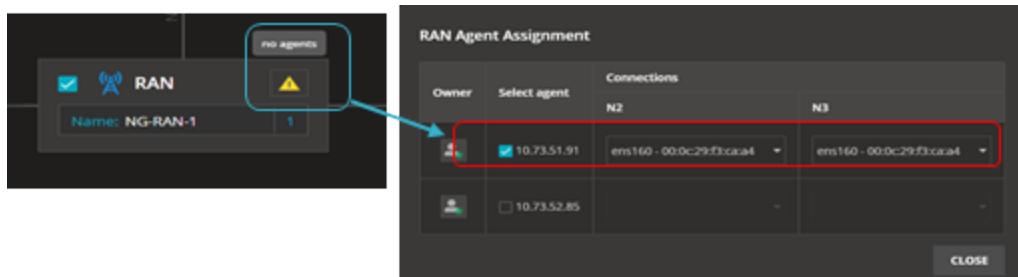
1. Connect to LoadCore UI using the IP that was assigned to Middleware (in this setup is 10.73.53.201)
2. On the Dashboard page, under the Create New Test section, select **Wireless Full Core**.



This creates a new session with Full Core topology, that is already configured with some parameters. The IP addresses assigned to the nodes must be changed in order to match the VM deployment.



- Select the yellow warning sign on the nodes and assign an agent to simulate it. In the drop-down list, there are the IP address of the agents that were previously defined:



- Agent 1** has: 10.73.50.65 and will be assigned for emulating the gNB
- Agent 2** has: 10.73.48.241 and will be assigned to the rest of the nodes

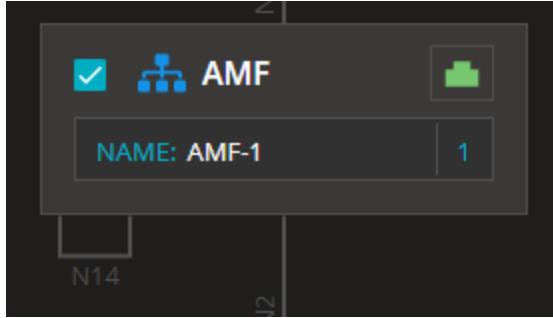
This screenshot shows the 'AGENTS ASSIGNMENT' interface. On the left, a sidebar lists various network nodes. The 'RAN' node is selected and highlighted with a red box. In the main area, the 'RAN Agent Assignment' table shows one agent assigned to the RAN node. The table has columns for 'Owner', 'Select Agent', 'Tags', and 'Connections' (N2, N3, Passthrough Device, S1). The 'Select Agent' column shows '10.73.50.65' with a checked checkbox. The 'Connections' columns show specific interface assignments for each connection type.

IMPORTANT

There is no **Back** button implemented in this release. In order to go back to previous screen, you need to click on session name, as shown below.



At the end of this operation all nodes should have a green icon for the agents assignment.



You can see the complete agent assignment by using Network Management tab. This can be opened by selecting any green icon and, then, select the **Network Management** tab.

AGENTS ASSIGNMENT			NETWORK MANAGEMENT									
			Filter agents									
Order	Agent	Tags	Impairment Profile	Agent Interface		Network Stack		SRIOV	Traffic Capture	Entity		
				Name	MAC	Mixed	Linux Stack				NRF	AUSF
1	10.73.50.65	IxStack ON (2) DPDK (2) build: 492 (2)	None	ens192	00:0c:29:c0:8b:74	Linux Stack	Linux Stack	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RAN	PCF	UDR
				ens160	00:0c:29:c0:8b:6a	IxStack over DPDK	IxStack over DPDK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RAN	NSSF	SMSF
2	10.73.48.241	IxStack ON (2) DPDK (2) build: 492 (2)	None	ens192	00:0c:29:07:57:a7	Linux Stack	Linux Stack	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NRF	AUSF	UDM/HSS
				ens160	00:0c:29:07:57:9d	IxStack over DPDK	IxStack over DPDK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5G-EIR	AMF	SMF/PGW-C
											UPF/PGW-U	

From here you can also enable traffic capture or configure different stacks to be used on a specific test interface (for example, application traffic can run only over IxStack Raw Sockets or IxStack over DPDK/Raw).

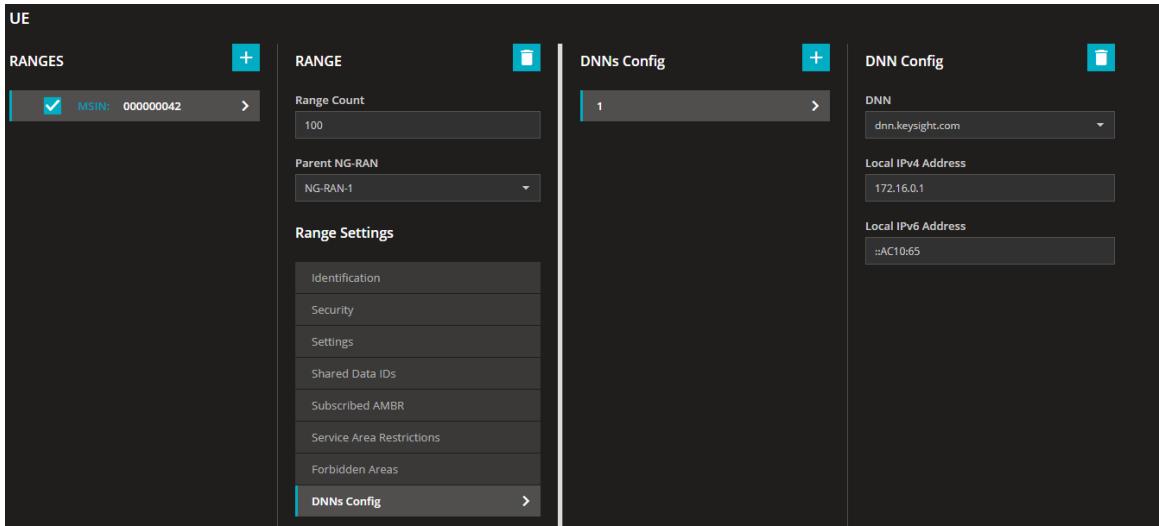
The default **Wireless Full Core** topology comes with predefined IP configuration. These parameters are listed below. You can run a basic test with this IP assignment. To do this, after assigning the agents, start the test.

When the test starts, the LoadCore application will configure these IP addresses on the deployed nodes (no need for manual configuration with `yaml` file, like in the previous LoadCore release).

The IP allocation is done as follows:

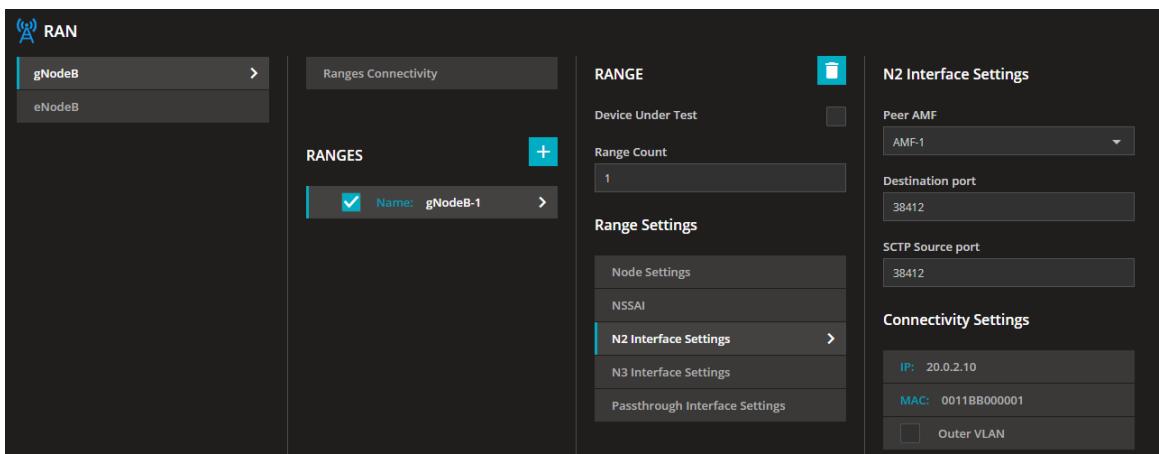
- **UE configuration**

- DNNs config: UE IP Address: **172.16.0.1**. The IP address is per DNN.



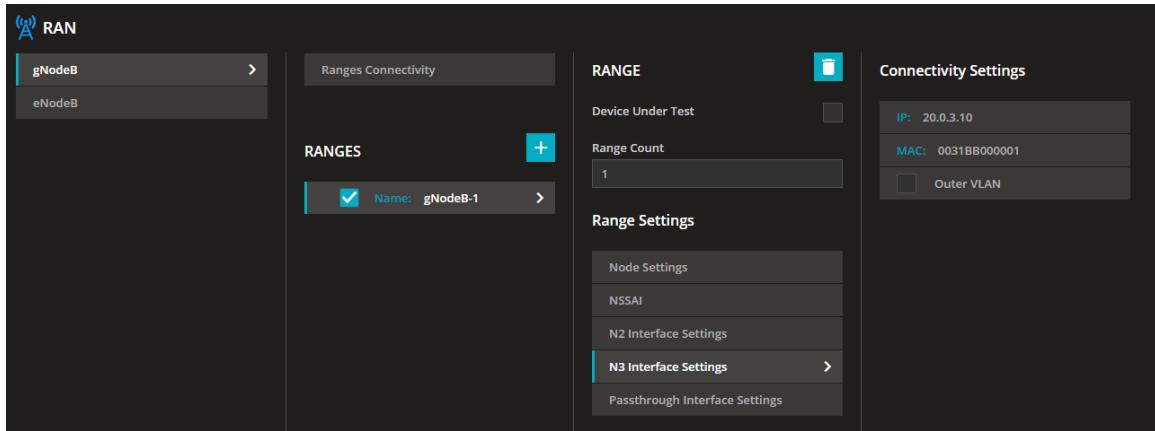
- **RAN configuration:**

- N2 Interface settings
 - IP address: **20.0.2.10**
 - Peer AMF: **AMF-1**



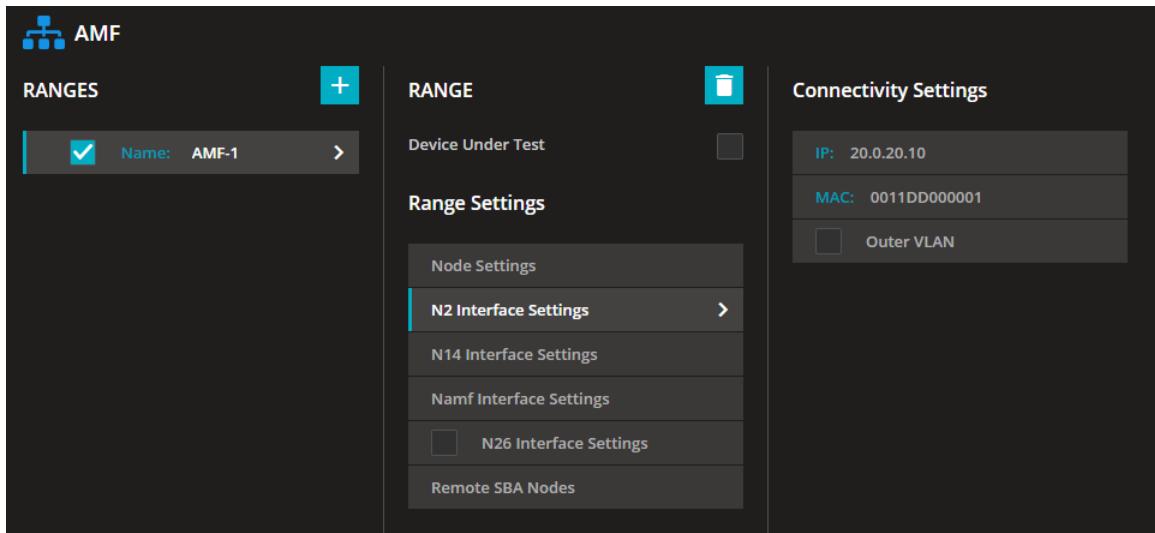
- N3 Interface settings:

- IP address: **20.0.3.10**

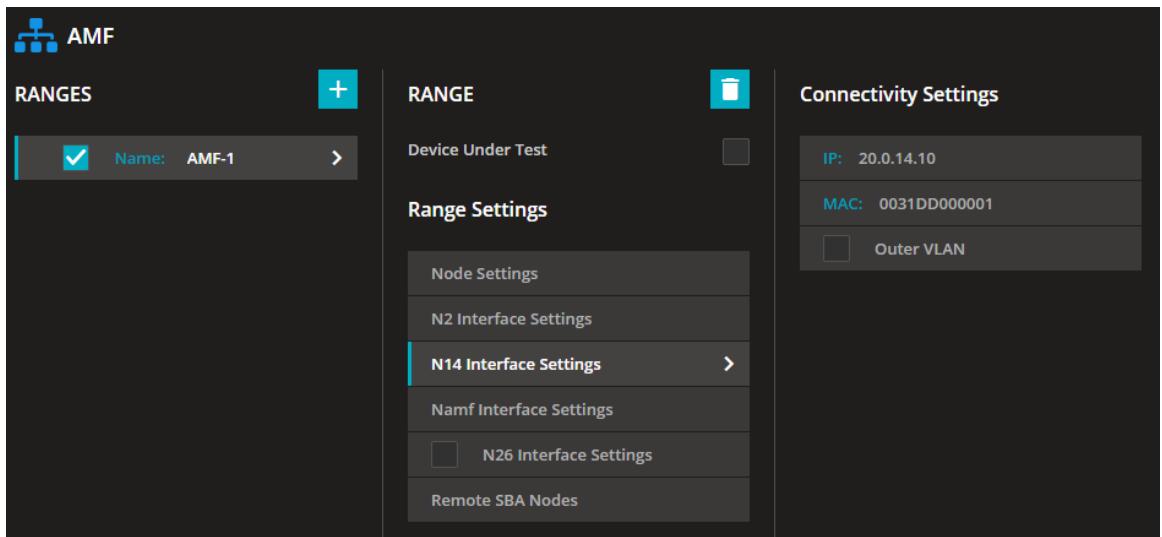


- **AMF configuration:**

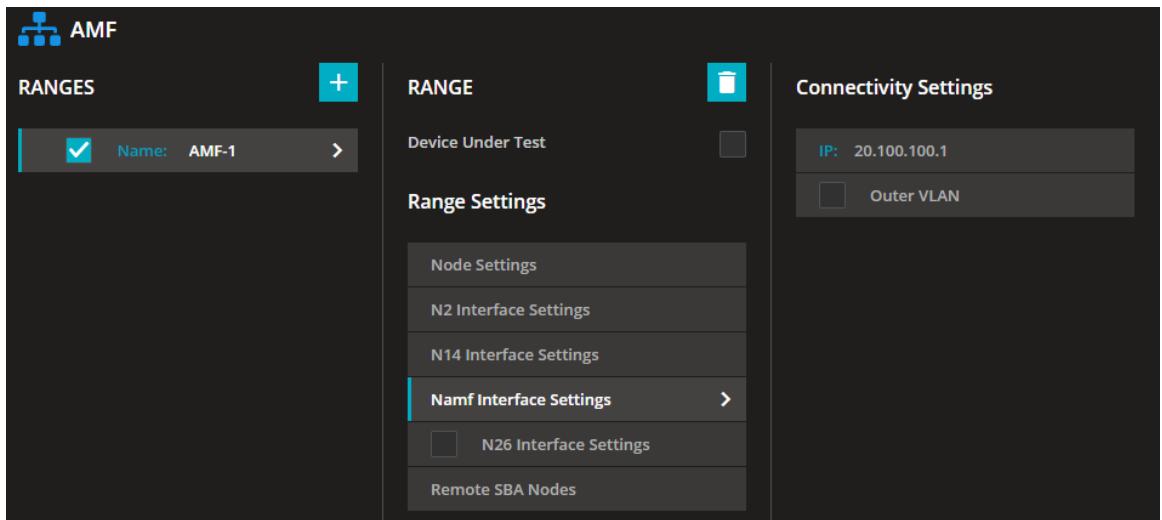
- N2 Interface settings
 - IP address: **20.0.20.10**



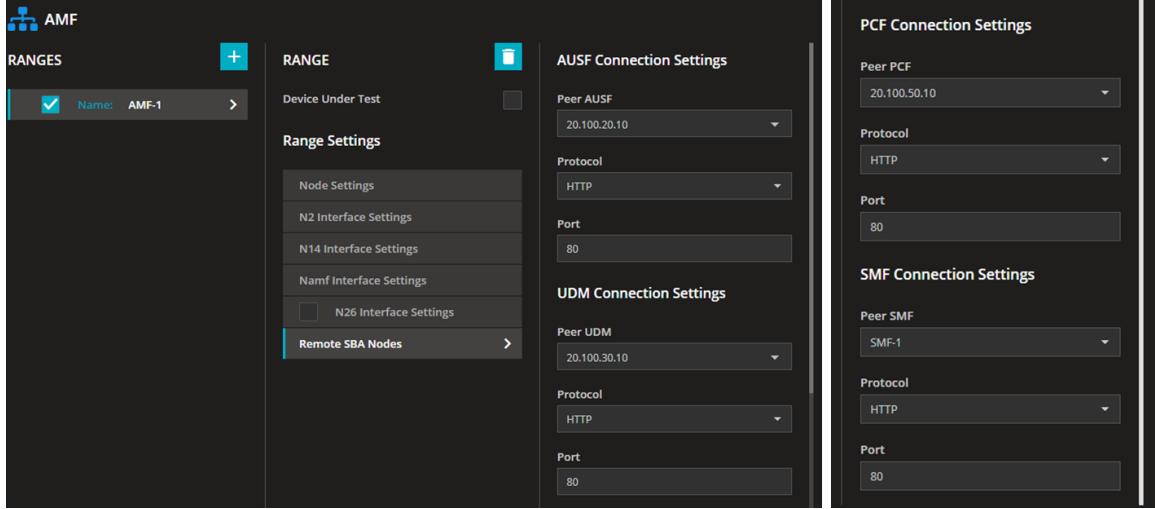
- N14 Interface settings
 - Local IP address: **20.0.14.10**



- Namf Interface settings
 - IP address: **20.100.100.1**

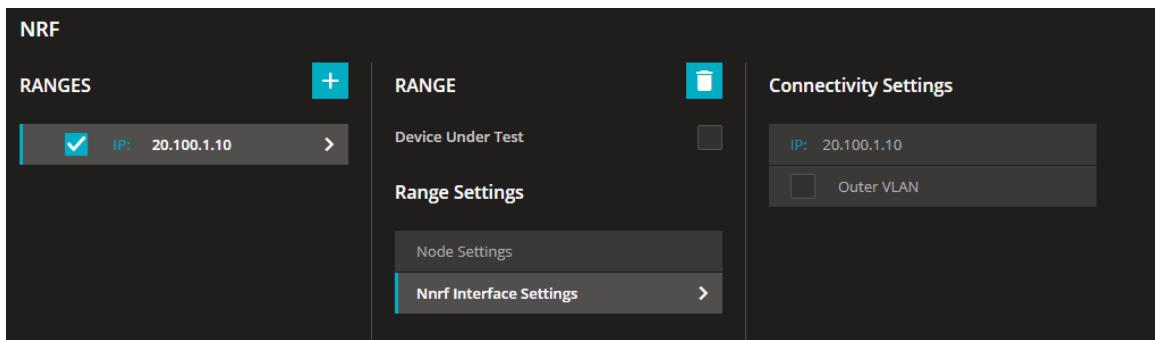


- Remote SBA nodes:
 - AUSF Connection Settings
 - IP address: **20.100.20.10**
 - UDM Connection Settings
 - IP address: **20.100.10.10**
 - PCF Connection Settings
 - IP address: **20.100.30.10**
 - SMF Connection Settings:
 - IP address: **SMF-1** (the IP address is automatically retrieved from the SMF configuration)



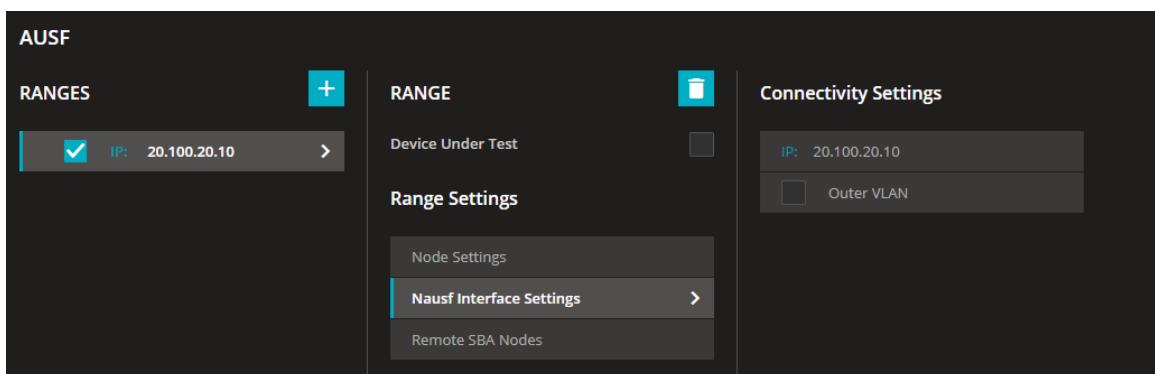
• NRF Configuration

- Nnrf Interface Settings
 - IP address: **20.100.1.10**



• AUSF Configuration

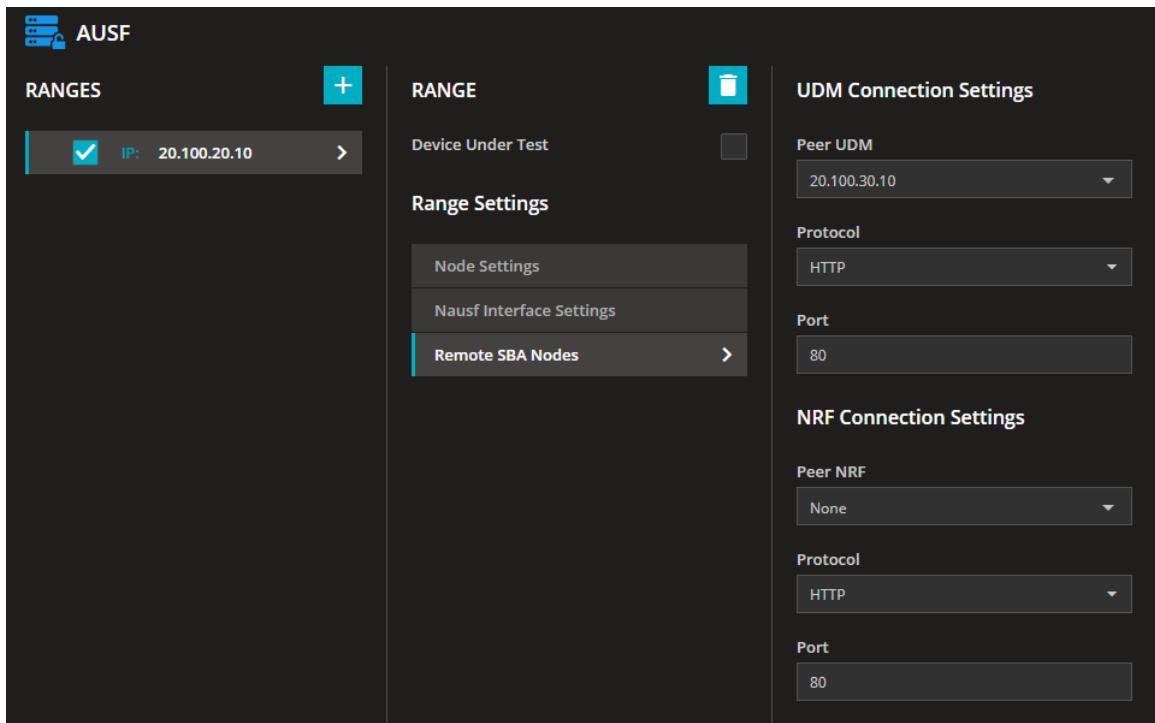
- Nausf Interface Settings
 - IP address: **20.100.20.10**



- Remote SBA Nodes:
 - UDM Connection Settings
 - IP address: **20.100.30.10**
 - Peer NRF: **None**

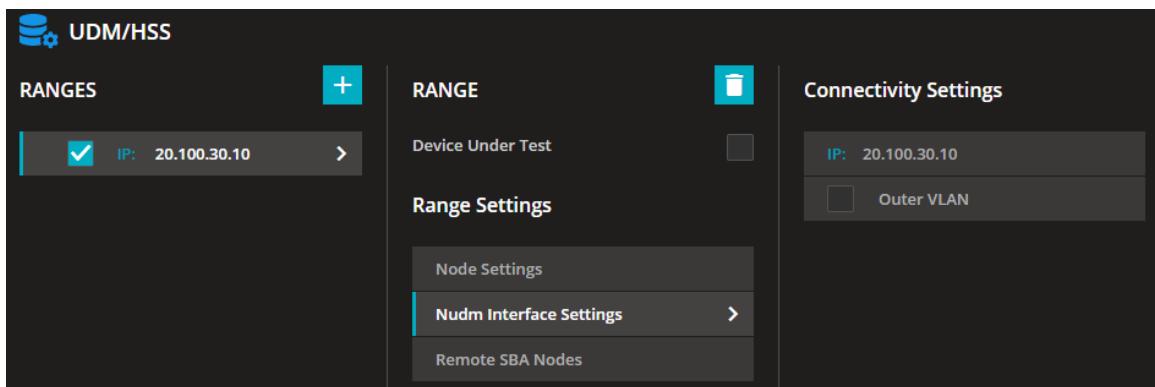
IMPORTANT

In all SBA nodes configuration, a Peer NRF set on **None** means the SBA nodes will not perform the registration process towards NRF. The tests will work because all nodes will know about each other from the IP addresses configured in LoadCore UI/



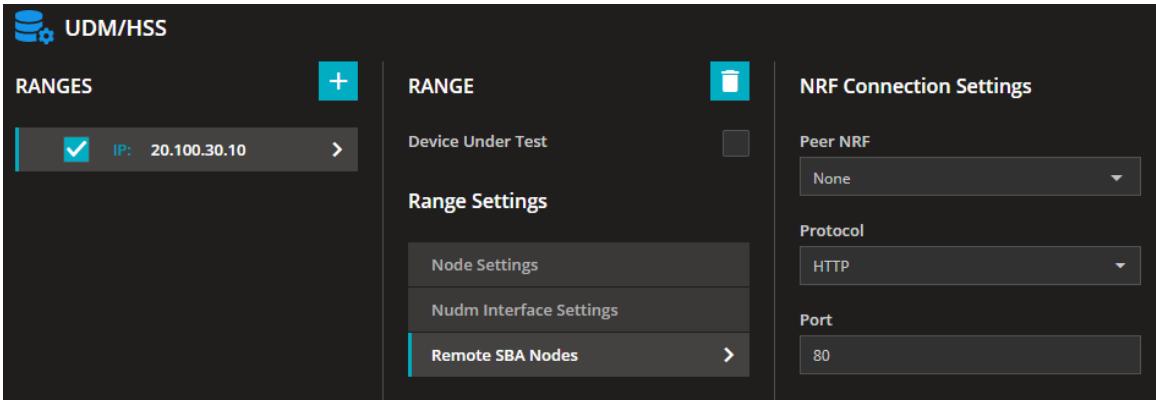
- **UDM/HSS Configuration**

- Nudm Interface Settings
 - IP address: **20.100.30.10**



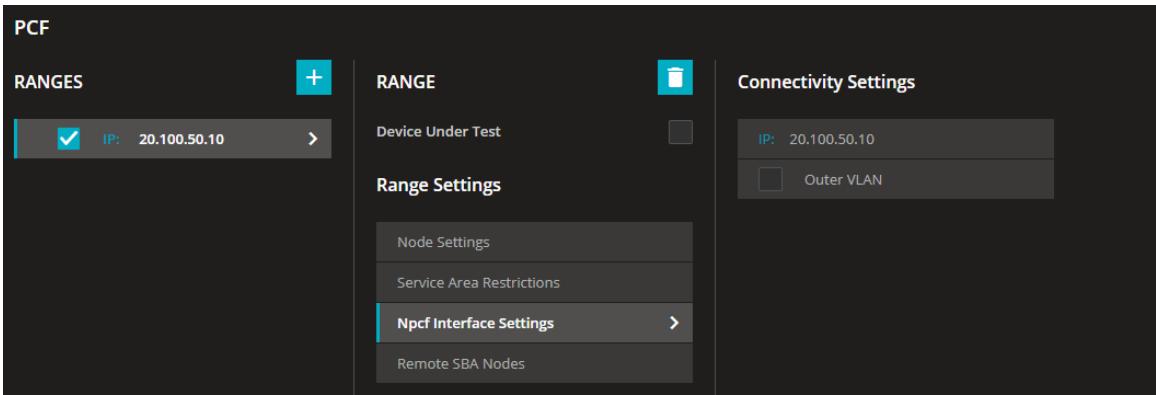
- Remote SBA Nodes:

- Peer NRF: **None**



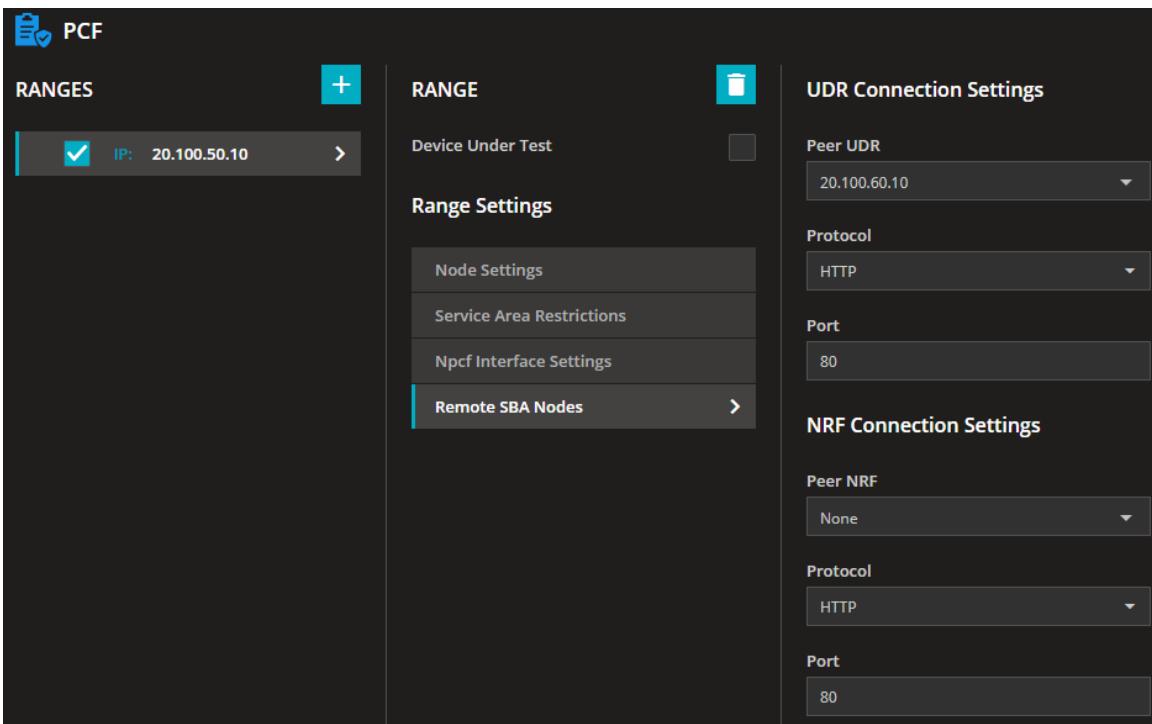
- **PCF configuration**

- Npcf Interface Settings
 - IP address: **20.100.50.10**



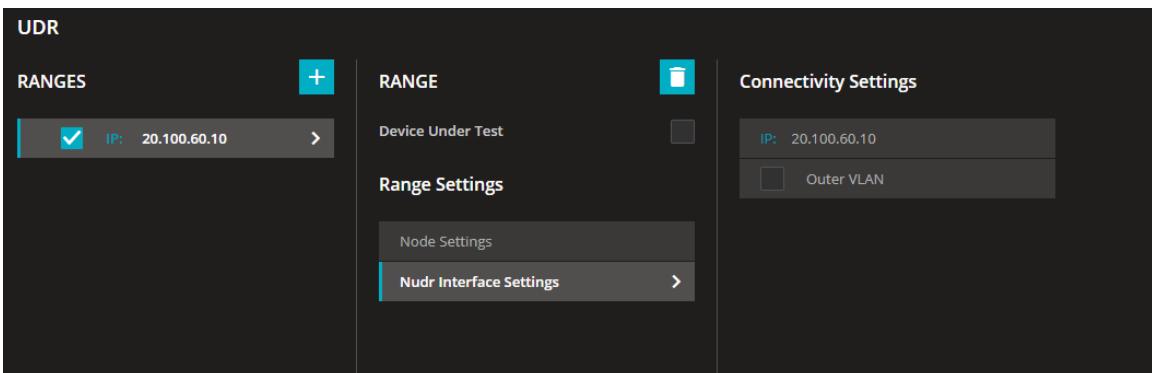
- Remote SBA Nodes
 - UDR Connectivity Settings
 - IP address: **20.100.60.10**

- Peer NRF: **None**



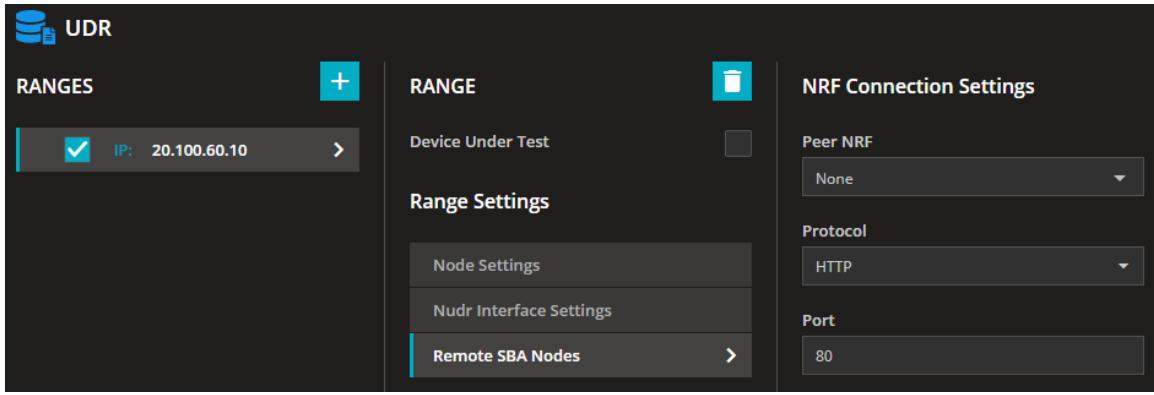
• UDR Configuration

- Nudr Interface Settings
 - IP address: **20.100.60.10**



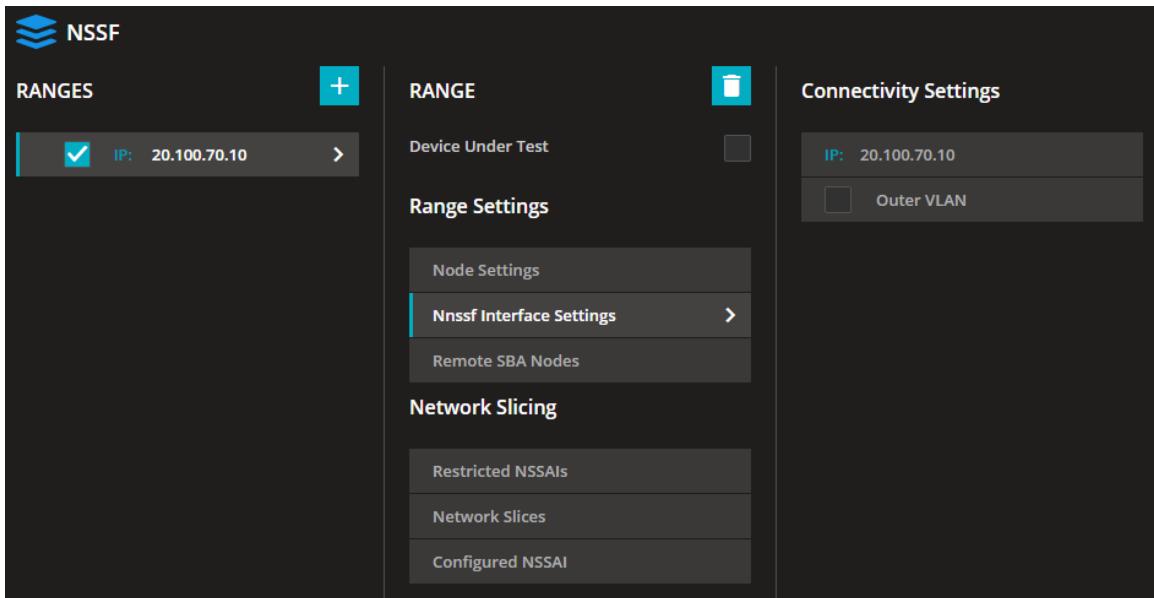
- Remote SBA Nodes

- Peer NRF: **None**



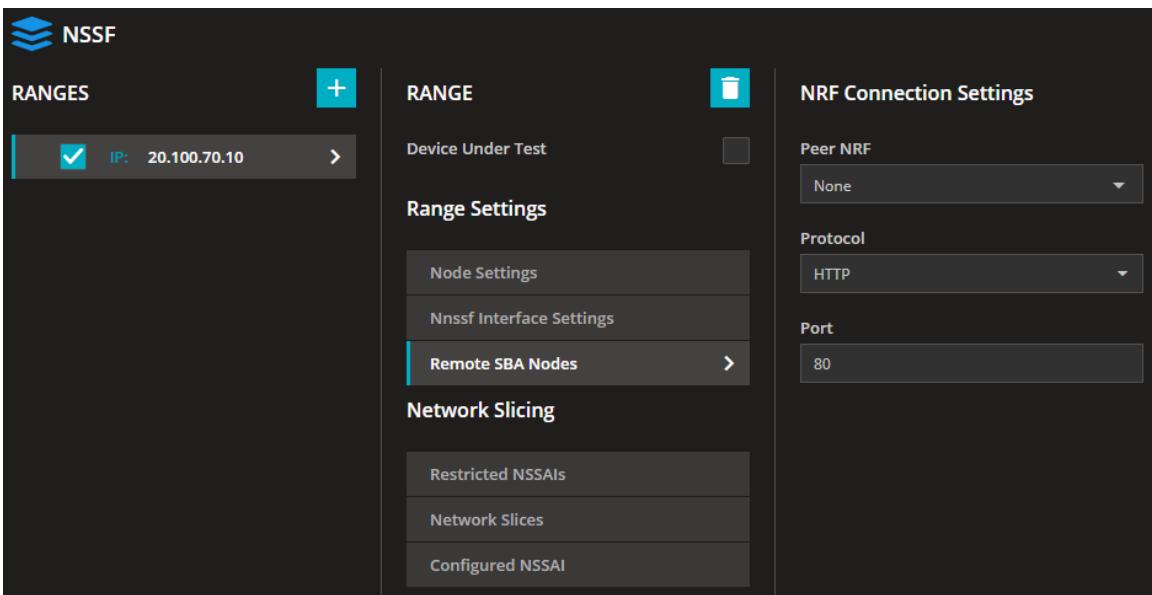
- **NSSF Configuration**

- Nnssf Interface Settings
 - IP address: **20.100.70.10**



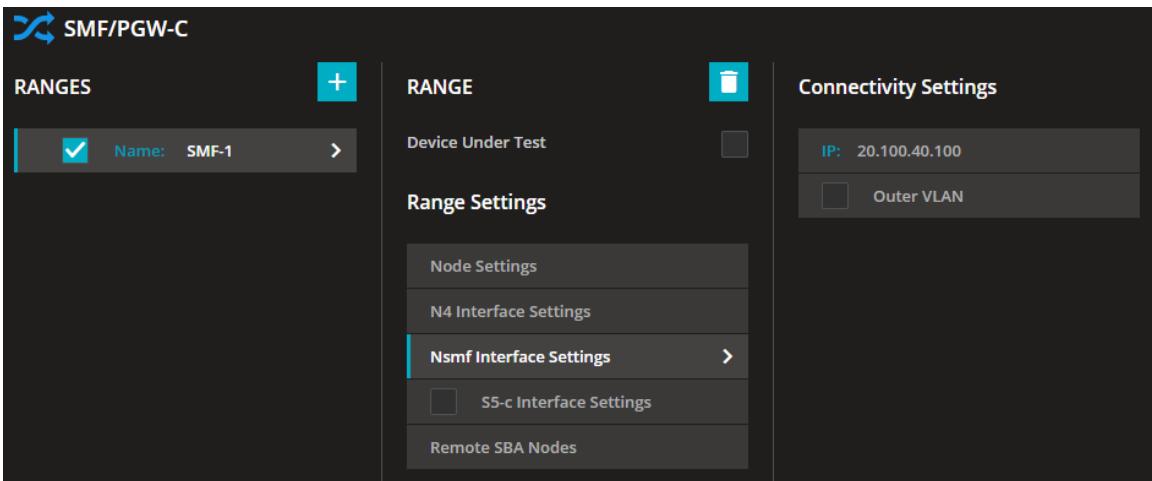
- Remote SBA Nodes

- Peer NRF: **None**



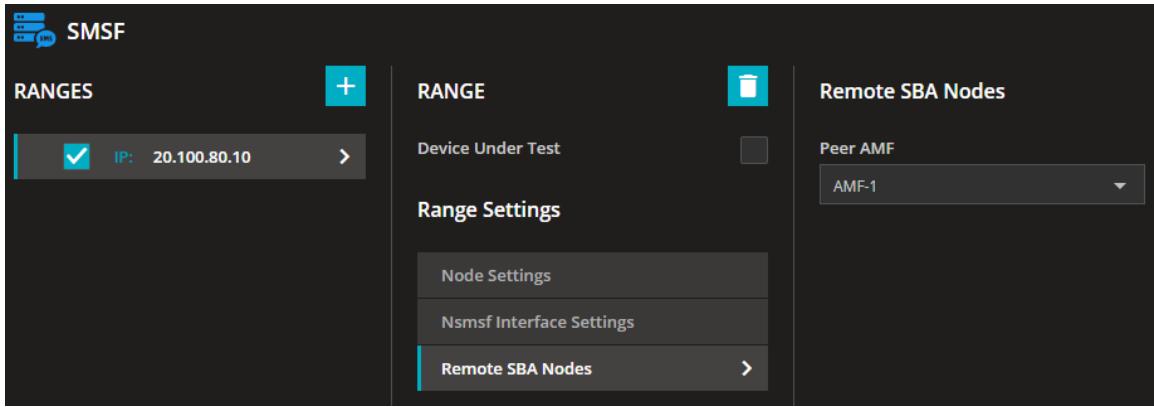
• SMSF Configuration

- Nsmsf Interface Settings
 - IP address: **20.100.80.10**



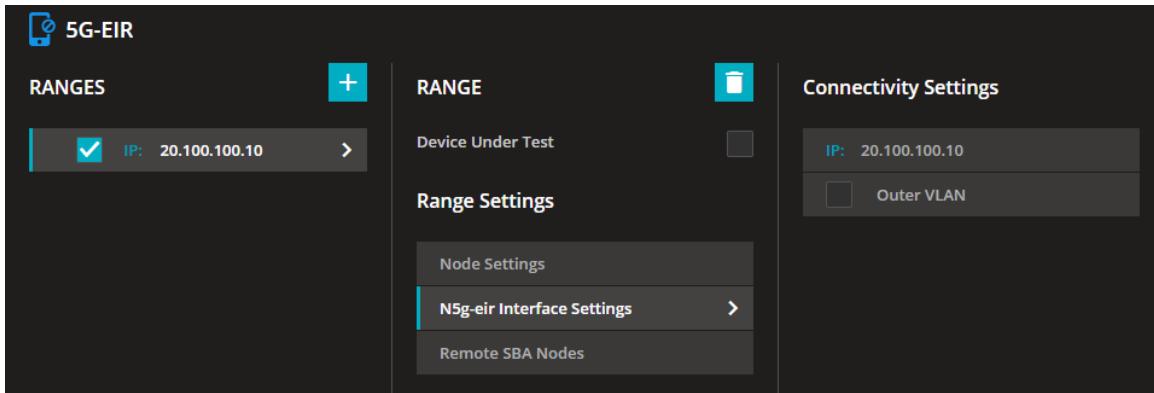
- Remote SBA Nodes

- Peer AMF: **AMF-1**



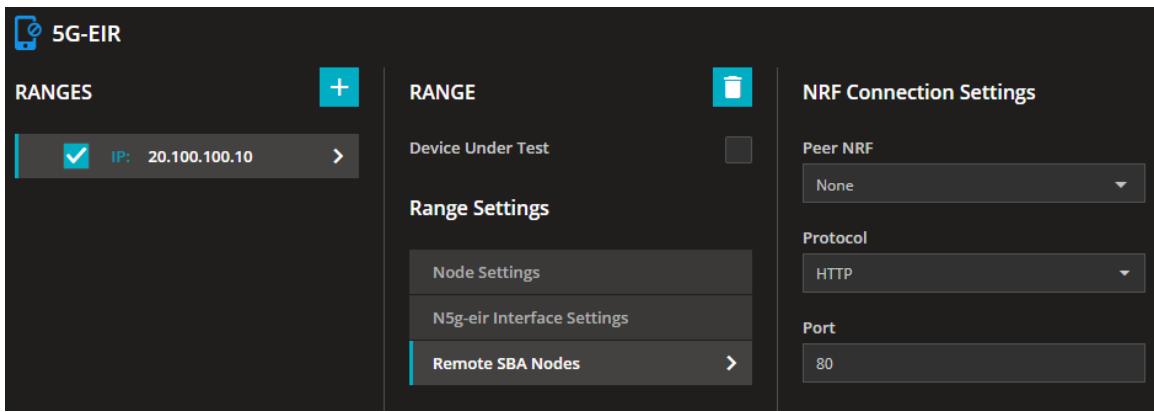
- **5G-EIR Configuration**

- N5g-eir Interface Settings
 - IP address: **20.100.100.10**



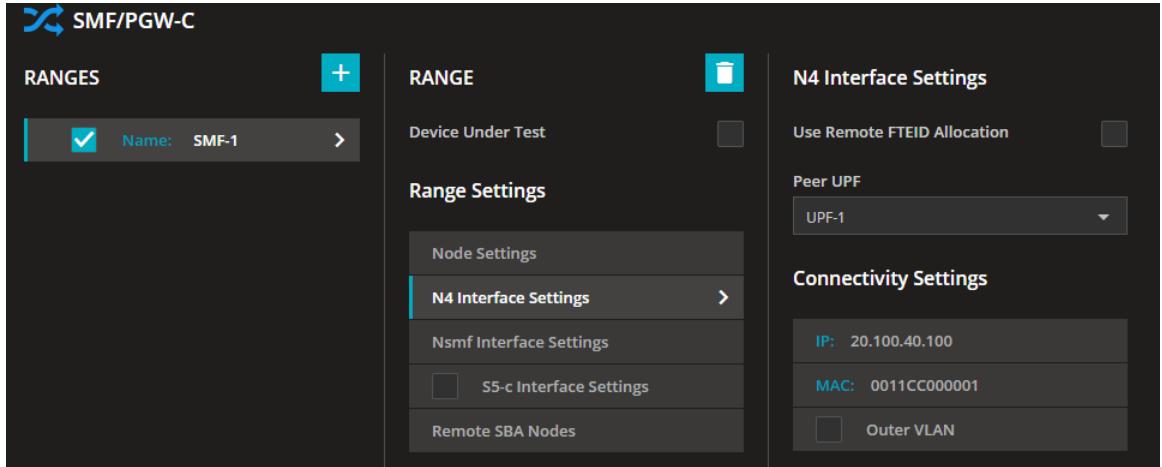
- Remote SBA Nodes

- Peer NRF: **None**

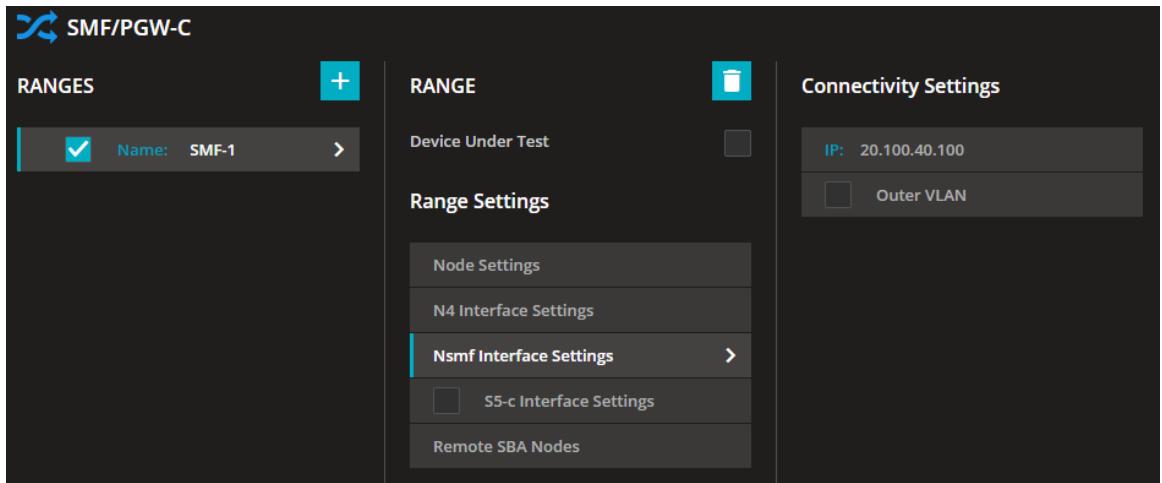


- **SMF/PGW-C configuration**

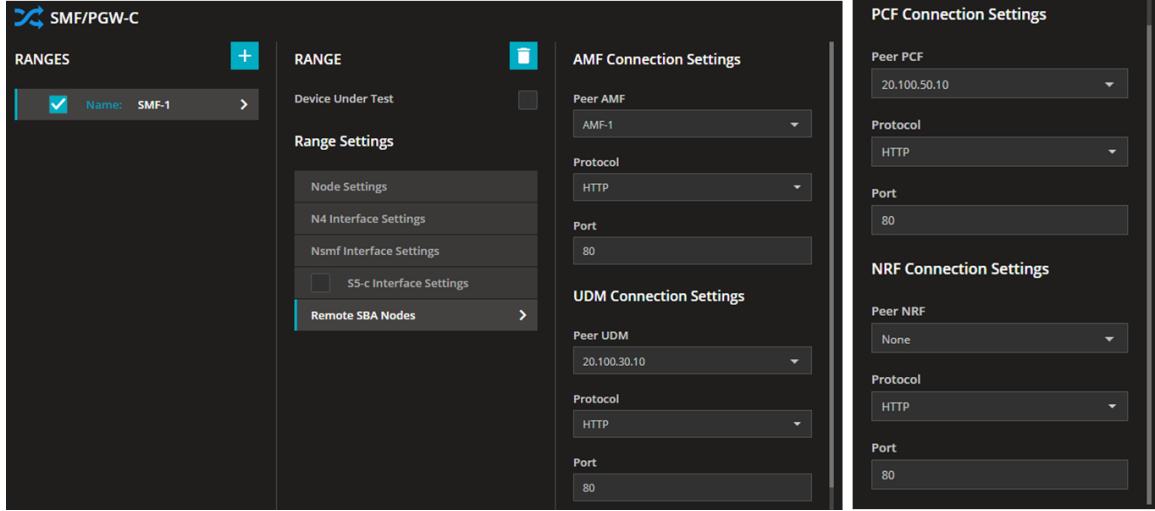
- N4 Interface Settings
 - IP address: **20.100.40.100**
 - Peer-UPF : **UPF-1**



- **Nsmf Interface Settings**
 - IP Address: **20.100.40.10** (same IP address as N4)

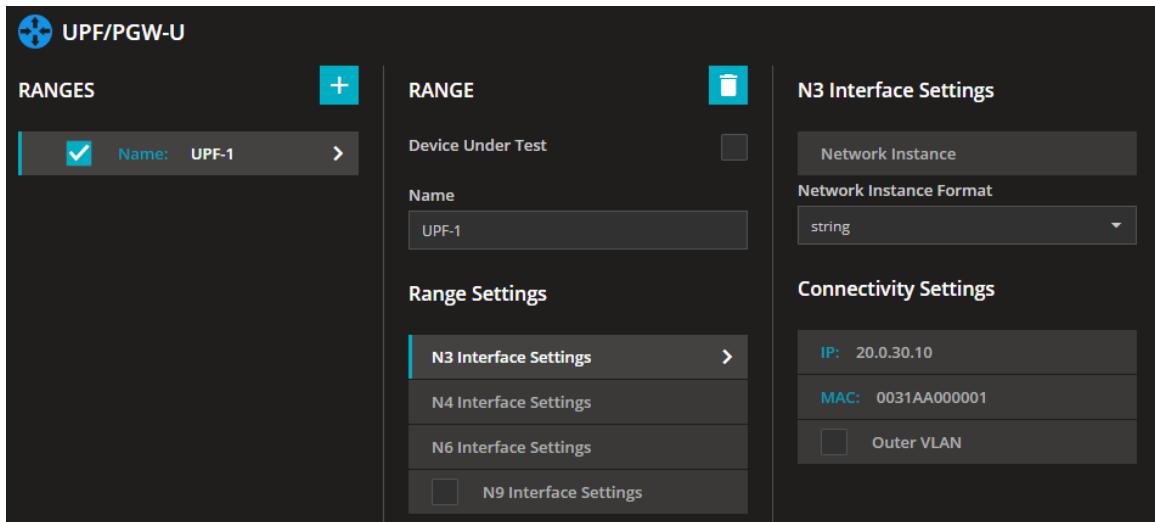


- **Remote SBA Nodes**
 - AMF Connection Settings
 - IP Address: **AMF-1**
 - UDM Connection Settings:
 - IP Address: **20.100.30.10**
 - PCF Connection Settings
 - IP Address: **20.100.50.10**
 - Peer NRF: **None**

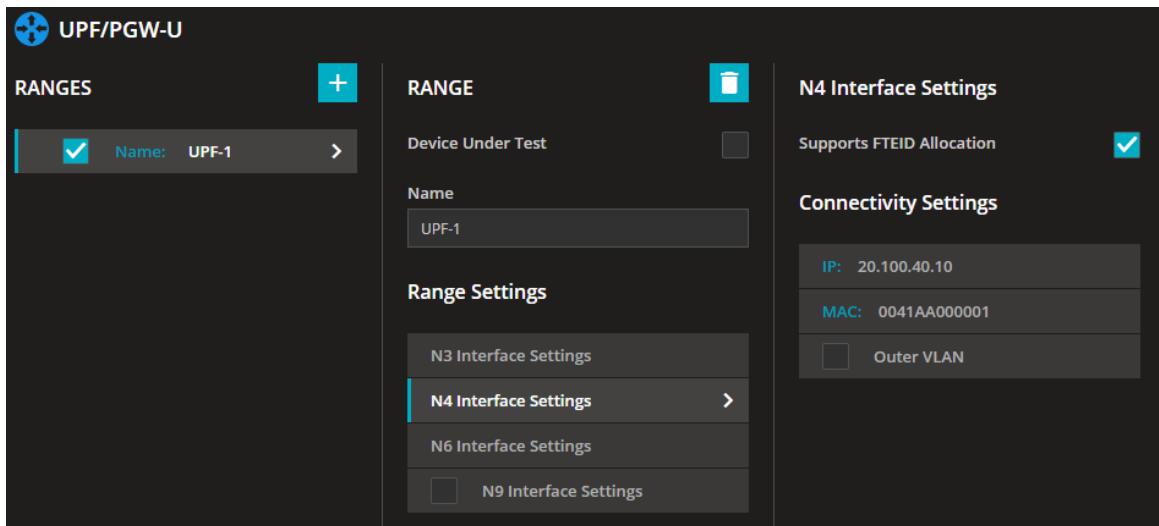


• UPF/PGW-U Configuration

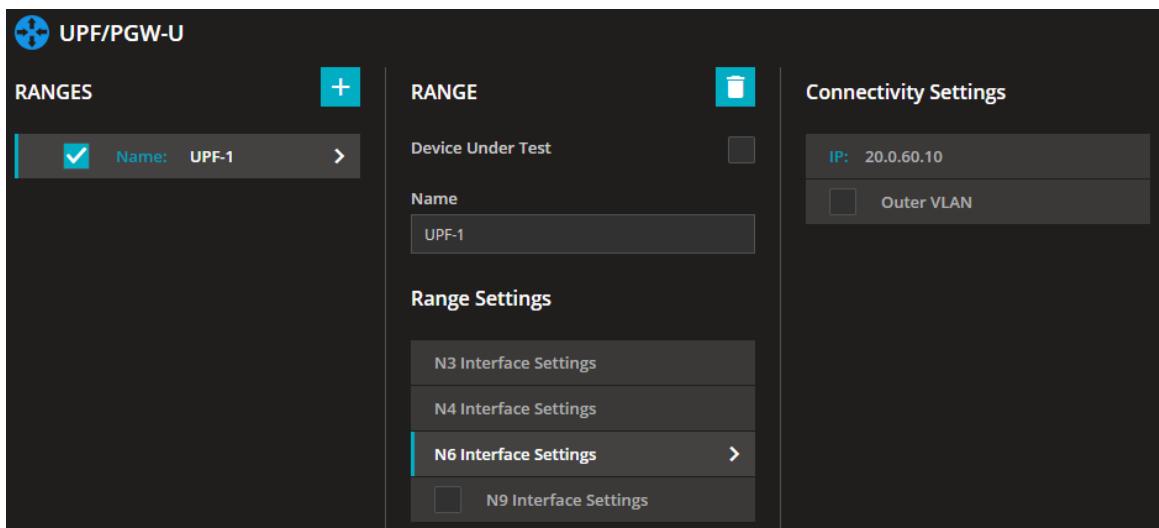
- N3 Interface Settings
 - IP Address: **20.0.30.10**



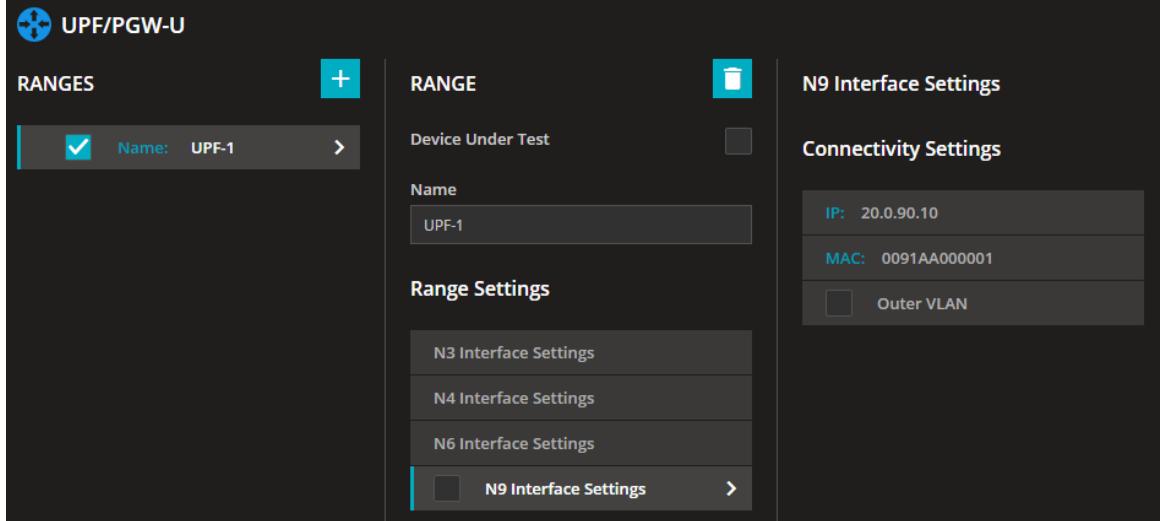
- N4 Interface Settings
 - IP Address: **20.100.40.10**



- N6 Interface Settings
 - IP Address: **20.0.60.10**

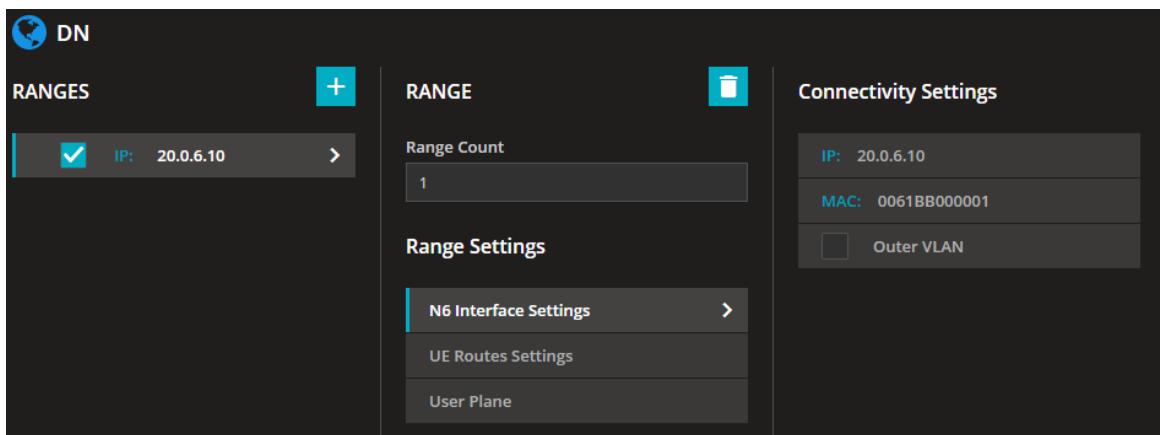


- N9 Interface Settings
 - IP Address: **20.0.90.10**



- **DN Configuration**

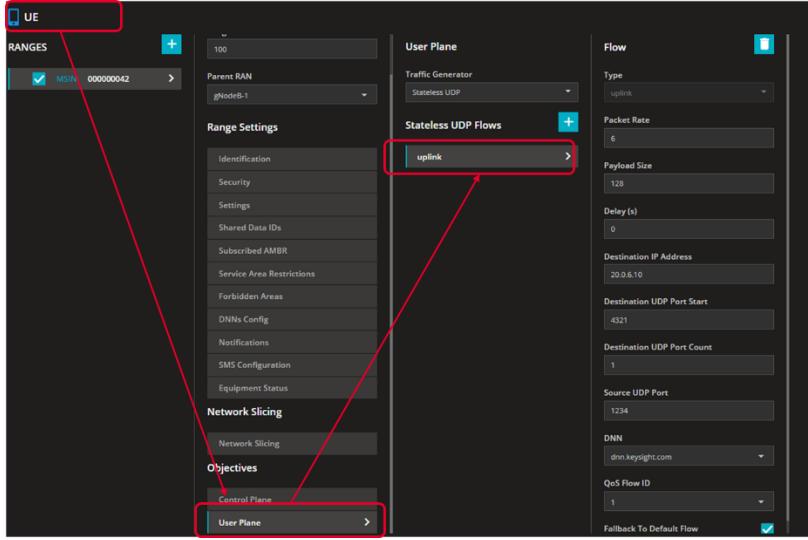
- N6 Interface Settings
 - IP Address: **20.0.6.10**



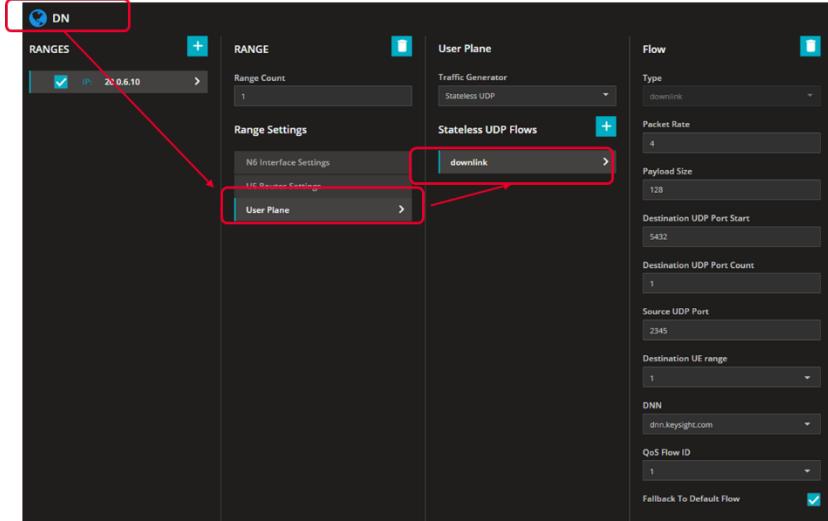
After completing all these steps, you should have the UE range configured and connectivity between 5G Core elements. The next step is to configure traffic flows.

In a wireless network the traffic is generated or received by UE, besides of other activities UE is doing (idle, handover,...). In LoadCore, traffic definition is implemented as UE Objective, therefore the configuration is done by accessing the UE menu:

- Select the **User Plan** objective and set **Stateless UDP** for traffic generator. Other traffic types can be configured too, but for this IxStack should be enabled – over Raw Sockets or over DPDK/Raw.
For this basic test UDP is used.
- There are two places where stateless UDP traffic is configured, according to the traffic direction:
 - For Uplink (UE towards DN traffic), you need to configure it on **UE > User Plane** objectives. The traffic direction cannot be changed, in this menu is only uplink traffic.



- For Downlink (DN towards UE traffic), you need to configure it on the DN node.



For each direction you can play with the packet size, rate, destination IP and port. You can change the values or leave them as they are, the test will run with these defaults.

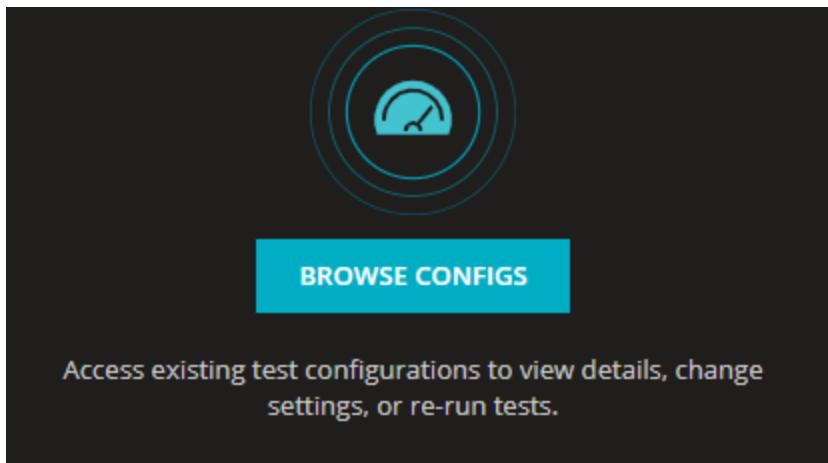
Import a test configuration

If you want to use an already configured test and run it, you can do this by importing the test configuration file (a `.json` file).

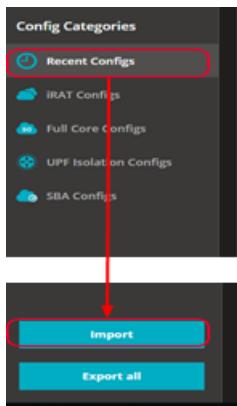
Sample tests can be found on Ixia's download page. You can save the archive and select the tests that best matches your needs.

To import the test configuration file:

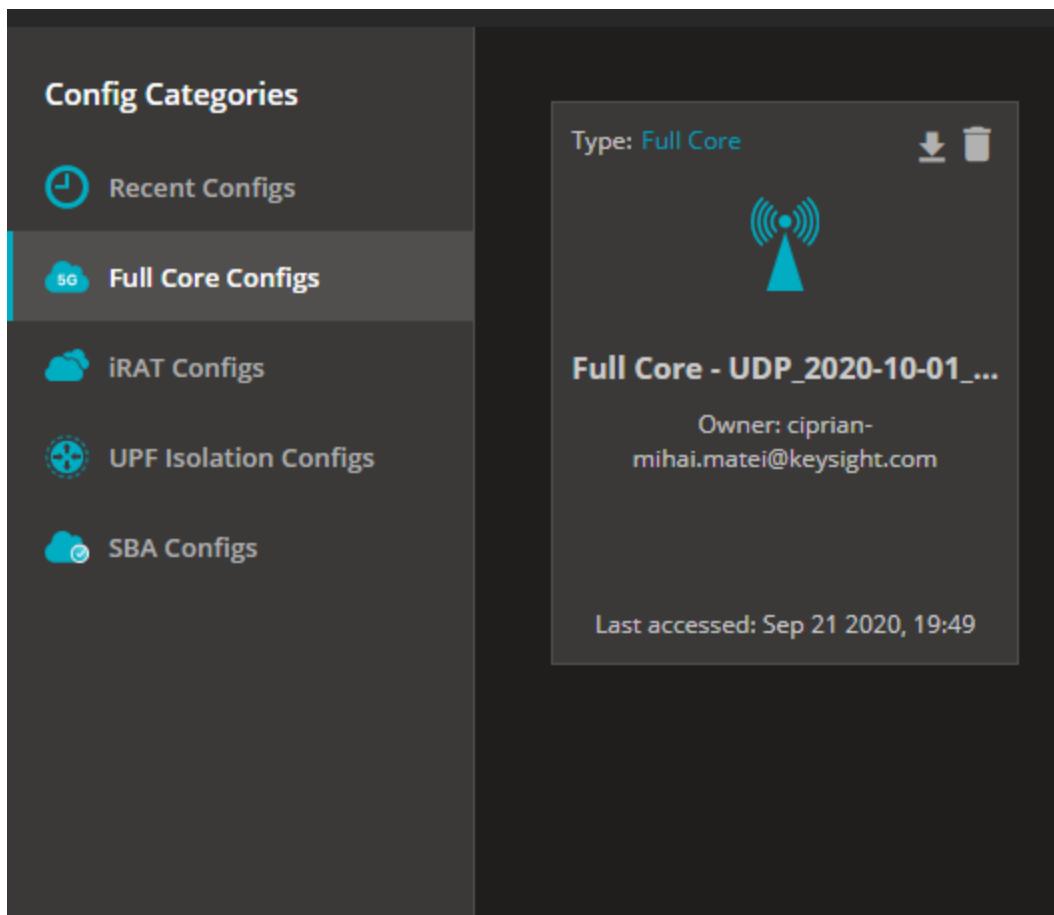
1. Select **LoadCore > Browse Configs**:



2. Select **Recent Configs > Import** and select the test .json file.



3. The test is added in the **Browse Configs > Full Core Configs** section from main LoadCore menu. Select it to create a new session based on this config and run it.



The download page containing the samples can be found at:

- <https://support.ixiacom.com/public/software-downloads/236052>

Run a test

After the complete configuration of the core elements, UEs and test objective definition, to run the test, select **Start Test**. The upper side of the Dashboard will show a summary of the session configuration.

Wait for the complete test initialization and run. If something is not properly configured/working, a message will be added on the Events tab. Select the bell icon and, then, select **Go to Events Page**.

0 OVERVIEW STATISTICS ciprian

Events

All i Info ! Warning ! Error

Date ↑	Type	Message
Mar 28, 2021, 11:40:32 PM	i INFO	Validating test configuration
Mar 28, 2021, 11:40:33 PM	i INFO	Acquire license
Mar 28, 2021, 11:40:35 PM	i INFO	Validate configuration
Mar 28, 2021, 11:40:35 PM	i INFO	Distribute configuration
Mar 28, 2021, 11:40:35 PM	i INFO	Polling Interval is 3
Mar 28, 2021, 11:40:35 PM	i INFO	Registering result
Mar 28, 2021, 11:40:36 PM	i INFO	Reserving 2 test agent(s)
Mar 28, 2021, 11:40:39 PM	i INFO	Initializing 2 newly reserved test agents
Mar 28, 2021, 11:40:39 PM	i INFO	Configuring 2 agent(s)
Mar 28, 2021, 11:40:39 PM	i INFO	Starting 2 agent(s)
Mar 28, 2021, 11:40:51 PM	i INFO	Test running

[GO TO EVENTS PAGE](#) [CLOSE](#)

■ STOP TEST

Filter events by

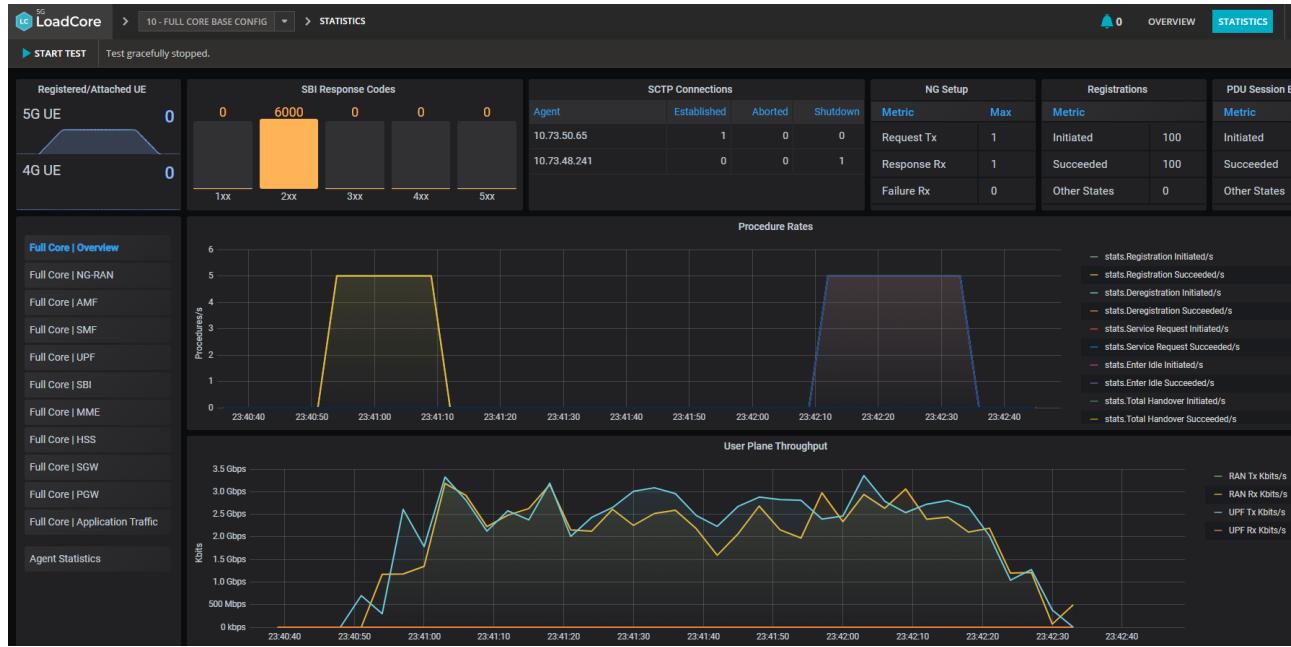
Message	From	To	Notification type
<input type="text"/> Type keywords	Select a date	Select a date	<input checked="" type="checkbox"/> All <input checked="" type="checkbox"/> i Info <input checked="" type="checkbox"/> ! Warning <input checked="" type="checkbox"/> ! Error

Date ↑	Type	Message
Mar 28, 2021, 11:39:43 PM	i INFO	Reserving 2 test agent(s)
Mar 28, 2021, 11:39:46 PM	i INFO	Initializing 2 newly reserved test agents
Mar 28, 2021, 11:39:46 PM	i INFO	Configuring 2 agent(s)
Mar 28, 2021, 11:39:46 PM	i INFO	Starting 2 agent(s)
Mar 28, 2021, 11:39:52 PM	i INFO	Test execution completed at 2021-03-28 20:39:52. Cleaning up and releasing resources!
Mar 28, 2021, 11:39:52 PM	i INFO	Exporting capture files from agents.
Mar 28, 2021, 11:39:53 PM	i INFO	Releasing 2 agents assigned to the test

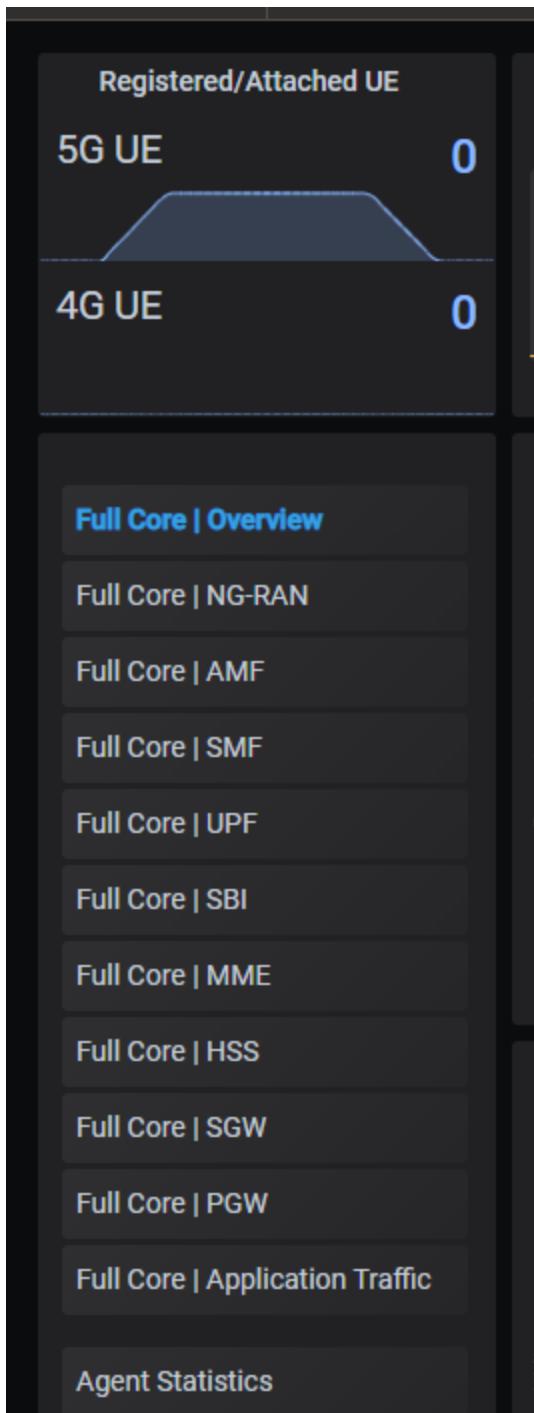
Results Analysis

The following section provides an overview for reading the statistics from LoadCore UI.

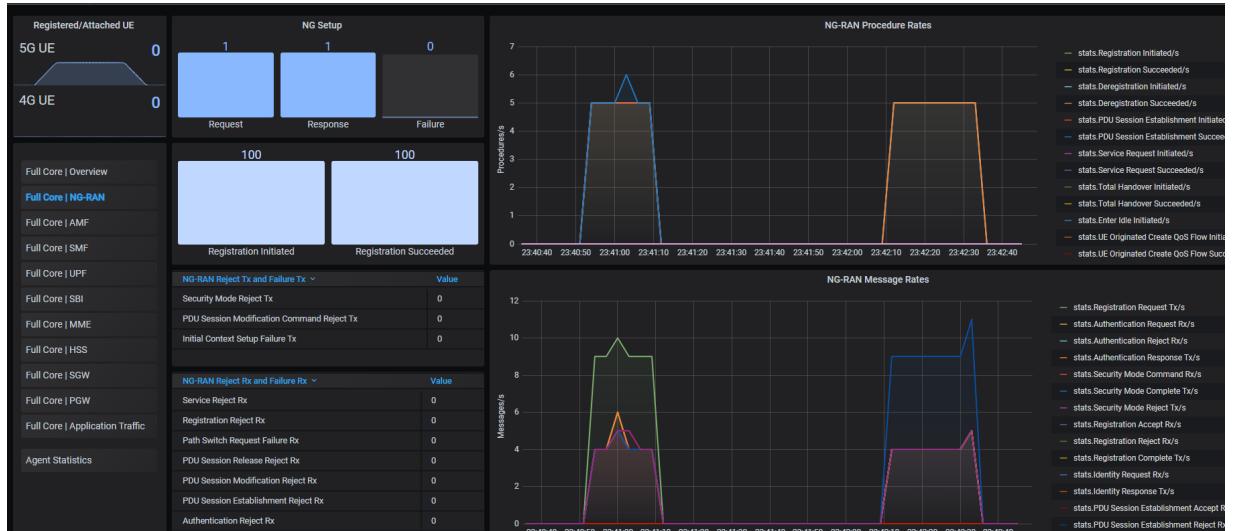
All stats are available under the **Statistic** tab.



The available statistics are displayed here in real-time and grouped under predefined **Dashboards**, per test topology and core node.

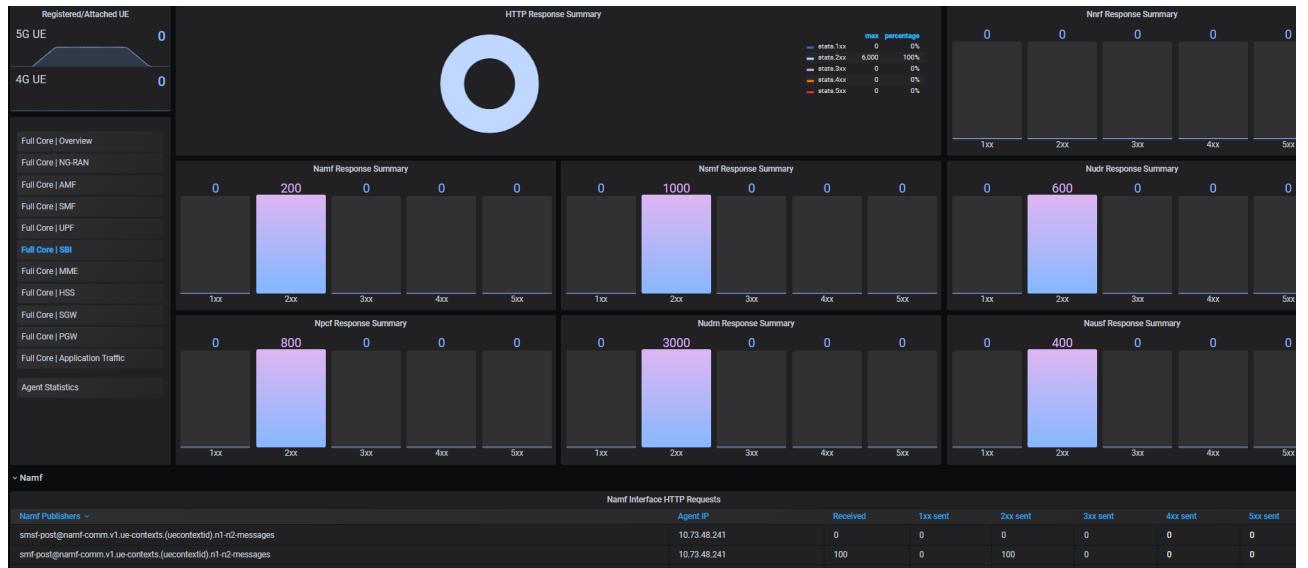


- **Full Core | Overview:** general dashboard displaying a global test overview: how many UEs attached, traffic volume on user plane and control plane.
- **Full Core | Core node:** displays relevant statistics for that node: procedures, rates, errors.



- **Full Core | SBI:** All messages from core SBI region, grouped per message response code.

A successful SBI message has a response code **2xx** (200 or 201) while an error is reported with a different code (for example, starting with **3xx**).



Downloading results, logs and captures

Starting with LoadCore 1.3, the Browse Results section can be accessed in order to retrieve the results, logs and captures of a test that was previously run.

Test Results (Hold down Ctrl, for multiple selection)	
Config Name	Test Name
142 - SBA Base Config	Wireless-5742186d-fb45-4d1b-be3d-0e3e09c59e2
139 - iRAT Base Config	Wireless-0be7f7cf-4cfa-4d90-b537-8ae56d6bd8df
136 - iRAT Base Config	Wireless-e364df0b-513d-4386-b6ff-49aa0ac9d781
135 - iRAT Base Config	Wireless-c3eab0c9-9e74-4616-1b43-4a35e9f94e6
133 - Full Core Base Config	Wireless-563a9c70-2045-4522-bc1c9e8617f1c03
132 - Copy of SBA Base Config (copy from Apr 23 07:31:42)	Wireless-cb59cbb-22e9-4d0d-a70c-30834cd0bb5
131 - Fullcore simulation (2021-03-19 11:11:08) (copy from Apr 23 07:28:53)	Wireless-272a4d9d-7a0d-4d9e-a120-72fb9ee24d3
130 - Copy of SBA Base Config	Wireless-bf24345e-910a-48d9-80c3-b13fa013edc
129 - Fullcore simulation (2021-03-19 11:11:08)	Wireless-51bd3d20-b01b-4a14-98c3-3e64dc1aef0
112 - SBA Base Config	Wireless-23e9fc1c3118-4cc9-bcf5-45483821a20
109 - longer mobility path 2020-08-18_11-16-06 (copy from Apr 22 19:43:20)	Wireless-eb0b58db-556a-41a4-a7b7-7026a96301b
108 - UPF isolation IPv4_PassThru_RoutersInBackhaul (copy from Apr 22 19:45:49)	Wireless-0e348v9-dc2c-4811-8dec-5d9f14e720006
107 - Sanity test_UPF_Isolation (copy from Apr 22 19:45:18)	Wireless-794551fc-f2c2-4b46-9222-6b5f11cc965
109 - UPF Iso - Tiger + Multiple QoS Flows - Multiple UE Ranges (copy from Apr 22 19:44:5..)	Wireless-f8a892b8-9f9c-4686-b339-63a6a5b5a06
105 - ApplicationTrafficXnIO_noTAC_ImgGET (copy from Apr 22 19:44:15) (copy from Ap..)	Wireless-8d4692c3132-447c-82b8-756371be472
104 - ApplicationTrafficXnIO_noTACchange_httpPUT (copy from Apr 22 19:42:03) (copy fr..)	Wireless-b506-996-08a4-40re-a278-0dd5aecd985
103 - UPF Iso - Tiger + Multiple QoS Flows - Multiple UE Ranges (copy from Apr 22 19:39:03)	Wireless-b22ecbc7666-41e5-be94-a798b03956da
102 - UPF Iso - Tiger + Multiple QoS Flows - 2 HTTPs 2 HTTP (copy from Apr 22 19:37:01)	Wireless-7a71:t165-01a2-49b8-8757-4169139e3518

On this section, the following actions are possible:

- Load the test configuration in a new session.
Download the test results.
- Download LoadCore logs and packet capture.

NOTE

To download the captures you need to enable traffic capture on the test agents.

- Delete the test results, by selecting the **Delete** button.

NOTE

At this moment, LoadCore does not have an automatic mechanism to delete old results, therefore this operation must be done manually in order to prevent MW disk to become full (especially running long duration tests).

Generate and retrieve csv statistics using Rest API

The agent's REST API User Guide can be accessed and the commands can be executed from there.

The example below uses Insomnia Rest API client to get the `csv` stats. Same operation can be done also from the agent's REST API User Guide.

NOTE

Replace `{ {demo_url} }` with the IP address of the agent from where you want to download the stats.

In this example, the agents are:

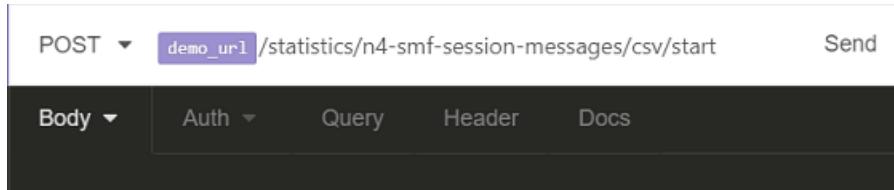
- agent_1**: 10.73.51.91 > `<demo_url>` is <https://10.73.51.91:443/api/v1>
- agent_2**: 10.73.52.85 > `<demo_url>` is <https://10.73.52.85:443/api/v1>

agent_1	Normal	5.33 GB	Ubuntu Linux (64-bit)	10.73.51.91	agent_1	30 MHz	540 MB
agent_2	Normal	5.33 GB	Ubuntu Linux (64-bit)	10.73.52.85	agetrn_2	42 MHz	680 MB
LicenseServer	Normal	6.36 GB	Ubuntu Linux (64-bit)	10.73.53.17	licensing-vm	32 MHz	910 MB
Middleware 1.1	Normal	14.35 GB	Ubuntu Linux (64-bit)	10.73.53.201	middleware11	2.2 GHz	5.66 GB

The command cycle is **Start** > **Stop** > **Get**, as follows:

1. To Start csv logging, issue the following command:

```
POST {{ demo_url }}/statistics/n4-smf-session-messages/csv/start
```



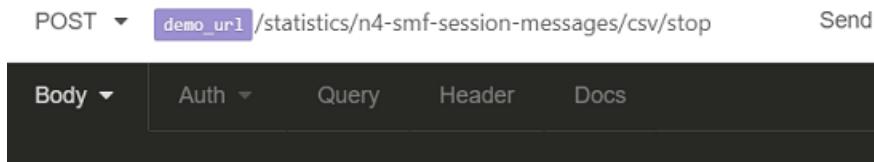
POST ▾ `demo_url` /statistics/n4-smf-session-messages/csv/start Send

Body ▾ Auth ▾ Query Header Docs

To capture all stats, issue the command after applying the configuration (PUT command) and before starting the execution (as detailed in Test execution flow).

2. To Stop csv logging, issue the following command:

```
POST {{ demo_url }}/statistics/n4-smf-session-messages/csv/stop
```

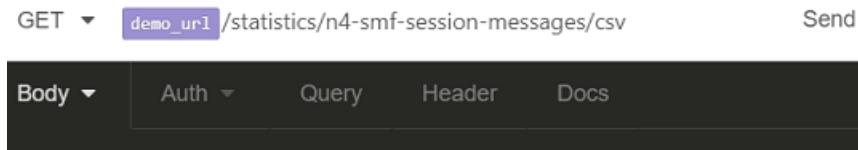


POST ▾ `demo_url` /statistics/n4-smf-session-messages/csv/stop Send

Body ▾ Auth ▾ Query Header Docs

3. Once the test has stopped, you can retrieve entire csv using command:

```
GET {{ demo_url }}/statistics/n4-smf-session-messages/csv
```



GET ▾ `demo_url` /statistics/n4-smf-session-messages/csv Send

Body ▾ Auth ▾ Query Header Docs

4. Issuing the above command for (publisher) n4-smf-session-messages will result in stats being listed within 200OK response window:

200 OK TIME 0 ms SIZE 25 KB

Preview	Header	Cookie	Timeline						
Timestamp	PFCP Session Deletion Request Retries	PFCP Session Deletion Request Tx	PFCP Session Deletion Response Rx	PFCP Session Establishment Request Retries	PFCP Session Establishment Request Tx	PFCP Session Establishment Response Rx	PFCP Session Modification Request Retries	PFCP Session Modification Request Tx	PFCP Session Modification Response Rx
12:23:39:544	0	0	0	0	0	0	0	0	0
12:23:40:560	0	0	0	0	0	0	0	0	0
12:23:41:571	0	0	0	0	0	0	0	0	0
12:23:42:586	0	0	0	0	0	0	0	0	0
12:23:43:771	0	0	0	0	0	0	0	0	0
12:23:44:788	0	0	0	0	0	0	0	0	0
12:23:45:800	0	0	0	0	0	0	0	0	0
12:23:46:803	0	0	0	0	0	0	0	0	0
12:23:47:817	0	0	0	0	0	0	0	0	0
12:23:48:830	0	0	0	0	0	0	0	0	0
12:23:49:842	0	0	0	0	0	0	0	0	0
12:23:50:861	0	0	0	0	0	0	0	0	0
12:23:51:872	0	0	0	0	0	0	0	0	0
12:23:52:880	0	0	0	0	0	0	0	0	0
12:23:53:900	0	0	0	3185	3174	0	0	0	0
12:23:55:942	0	0	0	23675	23674	0	0	0	0
12:23:56:946	0	0	0	33725	33724	0	0	0	0
12:23:57:951	0	0	0	43775	43774	0	0	0	0
12:23:58:951	0	0	0	53775	53774	0	0	0	0
12:23:59:955	0	0	0	63229	63228	0	0	0	0
12:24:00:955	0	0	0	73199	73198	0	0	0	0
12:24:01:959	0	0	0	83199	83199	0	0	0	0
12:24:02:961	0	0	0	93138	93124	0	0	0	0
12:24:03:972	0	0	0	103199	103199	0	0	0	0
12:24:04:972	0	0	0	113199	113197	0	0	0	0

5. You can generate csv stats for any available publisher. For example, to generate csv for SMF session rates:

```
POST {{ demo_url }}/statistics/n4-smf-session-messages-rate/csv/start
POST {{ demo_url }}/statistics/n4-smf-session-messages-rate/csv/stop
GET {{ demo_url }}/statistics/n4-smf-session-messages-rate/csv
```

With the following result:

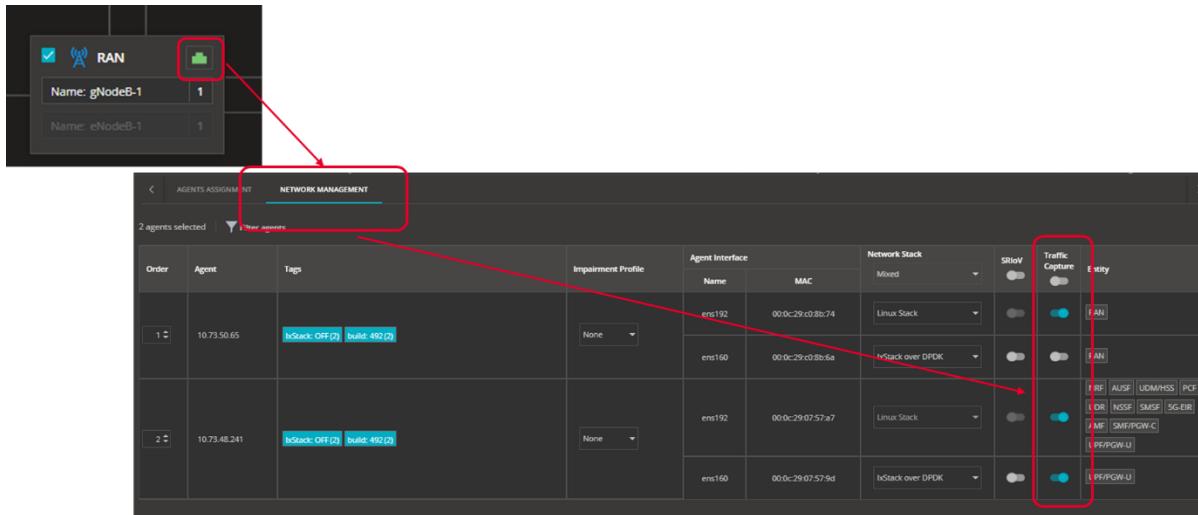
200 OK TIME 234 ms SIZE 1524 B

Preview	Header	Cookie	Timeline						
Timestamp	PFCP Session Deletion Request Retries/s	PFCP Session Deletion Request Tx/s	PFCP Session Deletion Response Rx/s	PFCP Session Establishment Request Retries/s	PFCP Session Establishment Request Tx/s	PFCP Session Establishment Response Rx/s	PFCP Session Modification Request Retries/s	PFCP Session Modification Request Tx/s	PFCP Session Modification Response Rx/s
13:05:56:889	0	0	0	0	0	0	0	0	0
13:05:57:903	0	0	0	0	0	0	0	0	0
13:05:58:918	0	0	0	0	0	0	0	0	0
13:05:59:929	0	0	0	0	0	0	0	0	0
13:06:00:940	0	0	0	0	0	0	0	0	0

Generate and retrieve packet captures

Using LoadCore UI

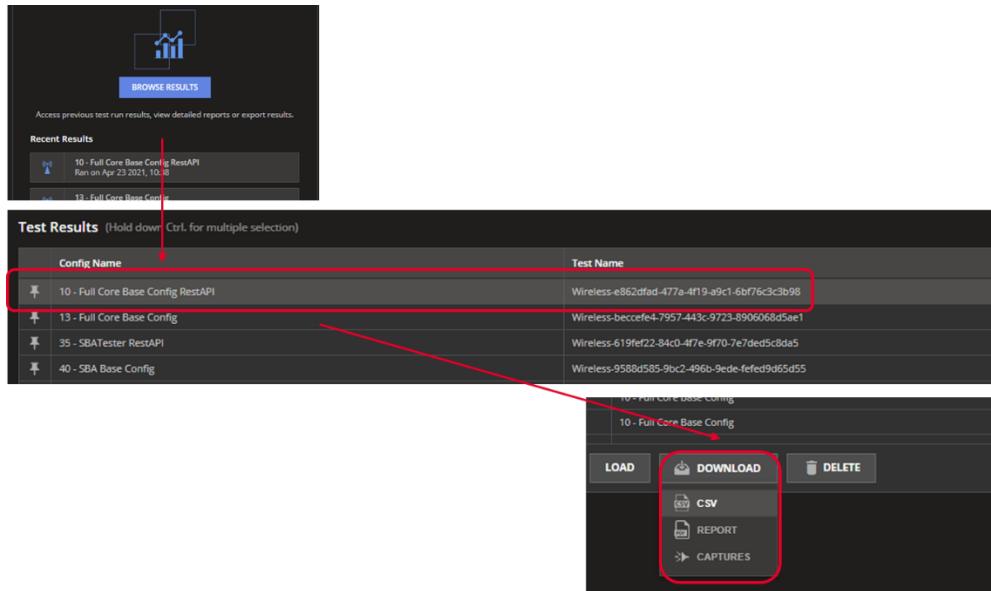
To enable packet capture from the LoadCore UI, select the agent assignment icon on any node, and select the Network Management tab.



IMPORTANT This operation will activate the traffic captures on the selected interfaces and the .pcap files will be automatically saved in the results folder for the test run.

Separate .pcap file will be generated for each test agent.

To download the .pcap files you need to select on Browse Results menu and select the test run you want the captures for, select **Download** and, then, select **Captures**.



If you are using a single agent in the test, then the traffic capture will not contain any packets, since all traffic is performed on loopback interface, not going out. To overcome this, you need to activate/get the capture on that agent using RestAPI calls and apply *any* as interface filter.

Using Rest API commands

The packet capture options are accessible via `/api/v1/capture`. The following operations are currently supported:

- Configuring the packet capture filter (in this case the test interface is **ens160**)
- Starting the packet capture
- Stopping the packet capture
- Downloading the packet capture file

In this example the agents are:

- **agent_1**: 10.73.50.65 > <demo_url> is **<https://10.73.50.65:443/api/v1>**
- **agent_2**: 10.73.48.241 > <demo_url> is **<https://10.73.48.241:443/api/v1>**

	Virtual machine ▾	Status	Used space	IP address	Host name
□	Middleware 2.1	Normal	17.31 GB	169.254.122.68	kcos-000c290310d7
□	LicenseServer	Normal	6.38 GB	10.73.53.17	licensing-vm
□	agent2	Normal	9.8 GB	10.73.50.65	5GCTE-6a96d7447
□	agent1	Normal	9.81 GB	10.73.48.241	5GCTE-6a96d7447

1. Set up the capture filter using the PATCH command:

```
PATCH {{ demo_url }}/capture/filter -i ens160
```

2. The packet capture can be started using:

```
POST {{ demo_url }}/capture/start
```

3. The packet capture can be stopped using:

```
POST {{ demo_url }}/capture/stop
```

4. The latest packet capture file can be accessed using:

```
GET {{ demo_url }}/capture
```

Using commands from Agent REST API User Guide

The agent's REST API User Guide can be accessed and the commands can be executed from there, as follows:

1. Connect to the Agent User Guide (http://<IP_address_of_the_Agent>).

The screenshot shows a web browser window with the following details:

- Address bar: `Not secure | 10.73.53.132/doc/index.html`
- Title bar: **5G Core Test Engine REST API**
- Content area:
 - Link: License
 - Menu items: Applications, Traffic, Capture

- From main menu of the web interface access select **Capture** and first define the interface from which the traffic will be captured

SG Test Engine REST API

Capture

Filter Operations for configuring packet capture options

- GET** /capture/filter Retrieve the filter configured for packet capture
- PATCH** /capture/filter Modify the packet capture filter

Capture Operations for controlling packet capture

- POST** /capture/start Start the packet capture
- POST** /capture/stop Stop the packet capture
- GET** /capture/status Retrieve the packet capture status
- GET** /capture Retrieve the packet capture file, in tcpdump pcap format

In this example the interface name is **ens160** and this value is updated in the message body. This is the test interface used to connect the two agents.

- Select **Filter** and, then, the **PATCH** operation. Update the interface name and select the **Execute** command.

Filter Operations for configuring packet capture options

PATCH /capture/filter Modify the packet capture filter

The filter can contain any command line options available for the tcpdump command.

Parameters

No parameters

Request body required

Example Value Model

```
{
  "values": "-i ens160"
}
```

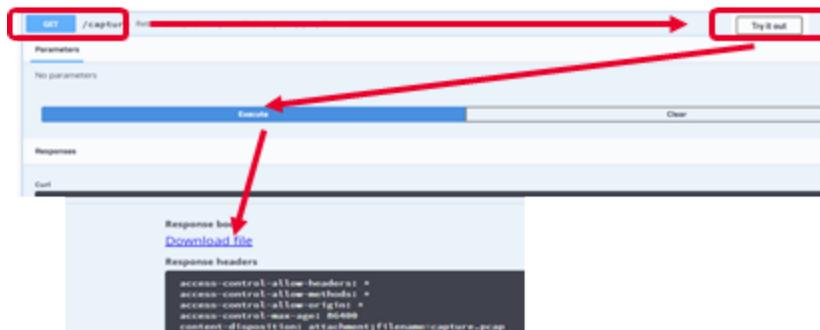
Try it out

- The capture operation can be started and stopped after running the test.

Capture Operations for controlling packet capture

- POST** /capture/start Start the packet capture
- POST** /capture/stop Stop the packet capture
- GET** /capture/status Retrieve the packet capture status
- GET** /capture Retrieve the packet capture file, in tcpdump pcap format

- To access the capture file, select the **GET** operation and after executing it, an updated download link is returned in the message body, from where the .pcap file can be downloaded.



To open the .pcap and analyze the captured traffic, use Wireshark.

Running Application traffic (HTTP)

In order to run Application traffic (other than UDP), you need to enable a different traffic engine. This will enable you to configure HTTP GET/PUT, FTP flows or voice traffic.

This traffic engine was designed for performance; therefore, it uses **IxStack over DPDK/Raw** or **Raw Sockets**.

The IxStack over DPDK/Raw mode required additional resources to be allocated to the traffic agents.

IMPORTANT The recommendation for an agent running IxStack over DPDK/Raw, for small number of UEs/TPS, is to assign 16 GB RAM and 8 CPU vCores.

If you need more performance, you need to increase the hardware resources assigned to the IxStack over DPDK/Raw agents, as specified below.

IMPORTANT The amount of RAM memory depends of the NIC you are using. For details, refer to <https://loadcore.htmlsave.net/>.

1. The minimum hardware requirements for Control Plane Only tests (no Application traffic) - default VM:
 - 4 GB RAM out of which 1GB is reserved for HUGE MEM(IxStack over DPDK/Raw)
 - 4 x vCPUs
 - 20 GB HDD
 - 1.2M UEs on gNB
2. The minimum hardware requirements for functional tests with Application traffic (10K UEs):
 - 8 x vCPUs
 - 1 x 10G NIC OR 1 x 40G NIC

	Mellanox	Intel	VMware
RAM (MB)	9971 MB	7686 MB	8011 MB
HugePages	1583 huge pages	440 huge pages	602 huge pages

3. VM with 10GBE NIC:
 - 8 x vCPUs
 - 1 x 10G NIC OR 1 x 40G NIC

	Mellanox	Intel	VMware
RAM (MB)	25196 MB	22911 MB	23235 MB
HugePages	8555 huge pages	7413 huge pages	7575 huge pages

4. VM with 100G NIC:
 - 18 x vCPUs
 - 80 Gbps TPUT with 1000 UEs

	Mellanox	Intel	VMware
RAM (MB)	22644 MB	17503 MB	18234 MB
HugePages	7416 huge pages	4845 huge pages	5211 huge pages

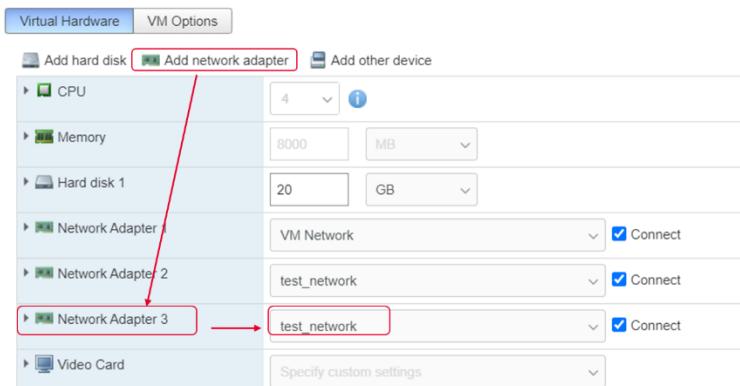
For different scenarios, two additional agents will be deployed having IxStack over DPDK/Raw on. In this way, you can run a simple UDP test for demo purposes with the previously deployed agents (agent_1 and agent_2) and when necessary to show TCP traffic type, you can run the test with the IxStack over DPDK/Raw enabled agents.

Installing additional interfaces on the test agents

In order to have full flexibility we will add an additional test interface to each agent. Also, this extra interface is necessary because the SBI nodes does not have yet support for IxStack over DPDK/Raw.

We will use the existing test interface (**ens160**) and just switch from Linux stack to IxStack over DPDK/Raw, and the new test interface will be used with Linux stack (and assigned for SBI nodes).

To do this, we will log to the ESXi interface, stop the agents and add one more interface (**ens192** in this case), and connect it to the same test network.



For both agents increase the HW resources as displayed below:

Hardware Configuration	
CPU	8 vCPUs
Memory	16 GB
Hard disk 1	20 GB
Network adapter 1	VM Network (Connected)
Network adapter 2	test_network (Connected)

In these tests the following are used:

- Ens160 – for all tests with app traffic (HTTP, HTTPS, FTP, Voice)
- Ens192 – for all tests with Stateless UDP flows and for simulating the SBI nodes
- Agent_1 (three interfaces):

- One for management (**ens32**)
- One for test, running IxStack over DPDK/Raw, for User Plane (**ens160**)
- One for test, for Control Plane, no IxStack over DPDK/Raw (**ens192**)
- Agent_2 (three interfaces)
 - One for management (**ens32**)
 - One for test, running IxStack over DPDK/Raw, for User Plane (**ens160**)
 - One for test, for Control Plane , simulating the rest of the modes (AMF_SMF_SBI part), no IxStack over DPDK/Raw(**ens192**)

Enabling IxStack over DPDK/ Raw

To run application traffic, you need to use IxStack on test interfaces that are assigned for user plane (N3, N6).

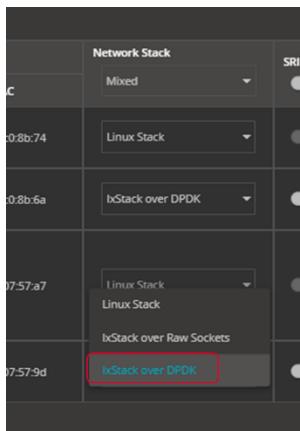
From the main topology, select the green icon (for agent assignment) and for NG-RAN and UPF/PGW-U assign the interface **ens160** for N3 and all the other interfaces on the nodes will remain on **ens192**.

Connections		N2	N3	Passthrough Device	S1	S1-u
Owner	Select Agent					
	10.73.48.241	IxStack: OFF (2) build: 492 (2)				
	10.73.50.65	IxStack: OFF (2) build: 492 (2)	ens192 - 00:0c:29:c0:8b:74	ens160 - 00:0c:29:c0:8b:6a	None	ens192 - 00:0c:29:c0:8b:74

After completing this step, select the **Network Management** tab. Notice that **ens160** is assigned to two nodes, and **ens192** is assigned to all nodes, as follows:

Order	Agent	Tags	Impairment Profile	Agent Interface		Network Stack	SRIOV	Traffic Capture	Entity
				Name	MAC				
1	10.73.50.65	IxStack: OFF (2) build: 492 (2)	None	ens192	00:0c:29:c0:8b:74	Linux Stack	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RAN
				ens160	00:0c:29:c0:8b:6a	Linux Stack	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RAN
2	10.73.48.241	IxStack: OFF (2) build: 492 (2)	None	ens192	00:0c:29:07:57:a7	Linux Stack	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NRF AUSF UDM/HSS PCE UDR NSSF SMSF SG-EIR AMF SMF/PGW-C UPF/PGW-U
				ens160	00:0c:29:07:57:9d	Linux Stack	<input type="checkbox"/>	<input checked="" type="checkbox"/>	UPF/PGW-U

For the rows in the table with **ens160**, set **Network Stack** to **IxStack over DPDK/ Raw** by selecting it from the drop-down list.



AGENTS ASSIGNMENT		NETWORK MANAGEMENT						
2 agents selected		Filter agents						
Order	Agent	Tags	Impairment Profile	Agent Interface	Network Stack	SRIoV	Traffic Capture	Entity
1	10.73.50.65	IxStack: OFF (2) build: 492 (2)	None	ens192	00:0c:29:c0:8b:74	Linux Stack	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				ens160	00:0c:29:c0:8b:6a	IxStack over DPDK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	10.73.48.241	IxStack: OFF (2) build: 492 (2)	None	ens192	00:0c:29:07:57:a7	Linux Stack	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				ens160	00:0c:29:07:57:9d	IxStack over DPDK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Configure HugePages on both Agents

IxStack over DPDK/Raw and high performance requires hugepages support configured on both agents running with IxStack over DPDK/Raw.

IMPORTANT This step is not needed if you are building a testbed for demo purposes, with small number of UEs and less throughput.

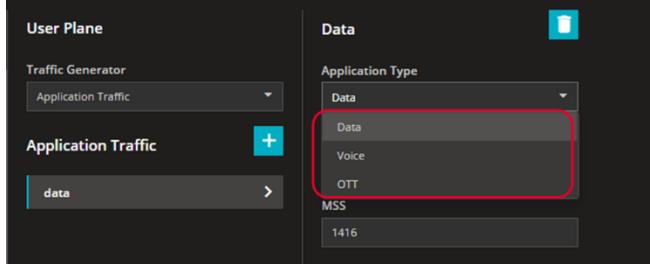
To do this, edit the `/boot/grub/grub.cfg` file and add `hugepages=8192`.

```
ixia@5GCTE-6a96d7447: /boot/grub
set timeout=1
menuentry '5G Core Test Engine - Linux 4.15.0-74-generic' {
    linux /boot/vmlinuz-4.15.0-74-generic root=UUID=b0cc522e-5f39-498b-b81b-7058
42d07744 net.ifnames=1 hugepagesz=2M hugepages=8192 ro quiet
    initrd /boot/initrd.img-4.15.0-74-generic
}
~
```

To complete the process, reboot the agent.

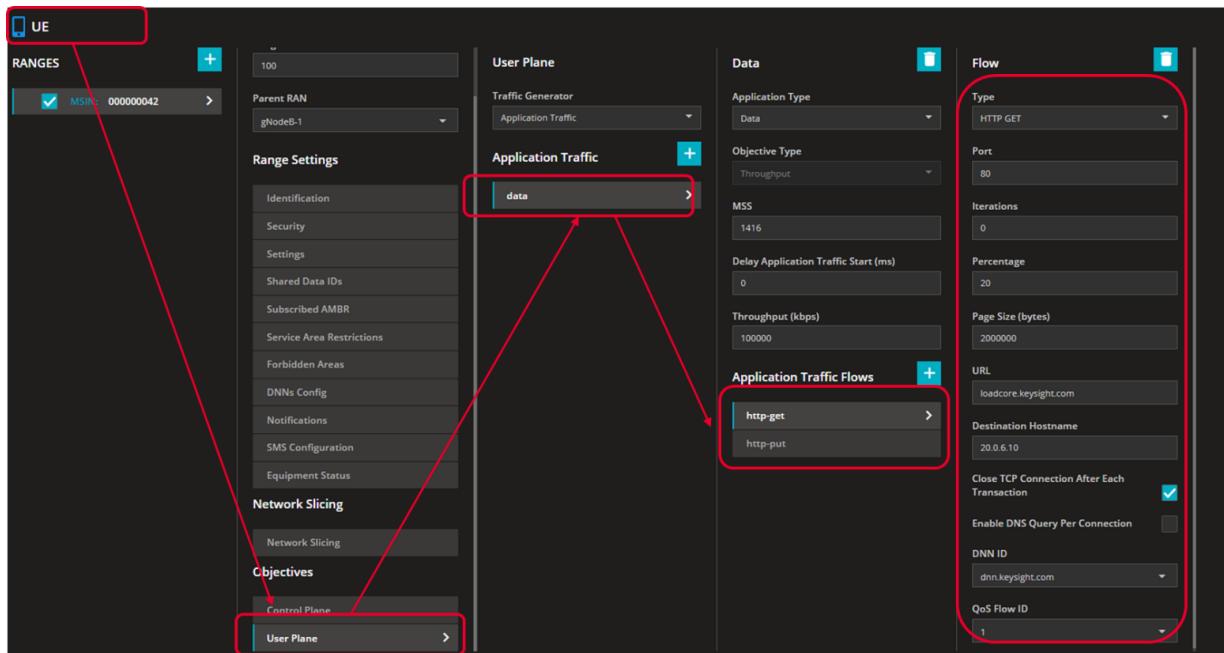
Application traffic configuration

Starting with LoadCore 2.1, there are three types of application traffic that can be used: **data**, **voice** and **video**.



The configuration of this type of traffic be done in two places:

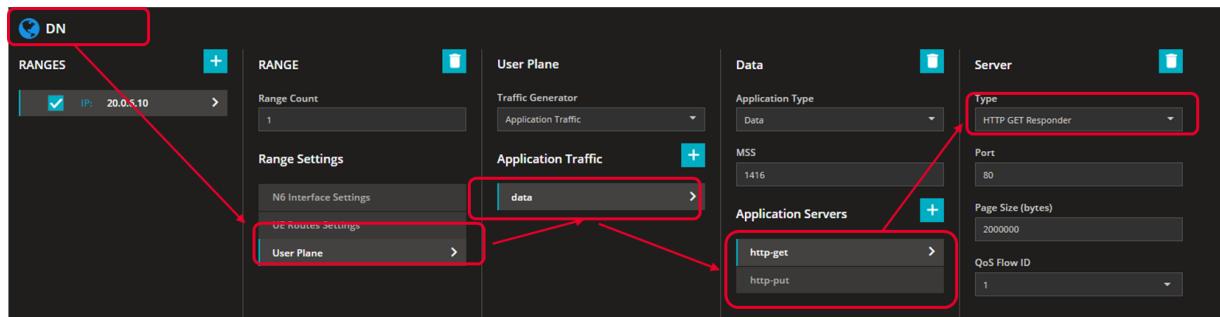
- On **UE > User Plane** objective: this controls the traffic that is sent or received by the UE, for both directions.



- On **DN > User Plane**: here you are configuring the server settings in order to answer the traffic the UE is sending. The configuration should match the settings you defined in the **UE > Control Plane > App traffic** (from the step above).

Here you are configuring the parameters for the endpoint considered *the responder* of the

traffic.



Create Custom Dashboard

There are two way to create a custom dashboard in order to display statistics:

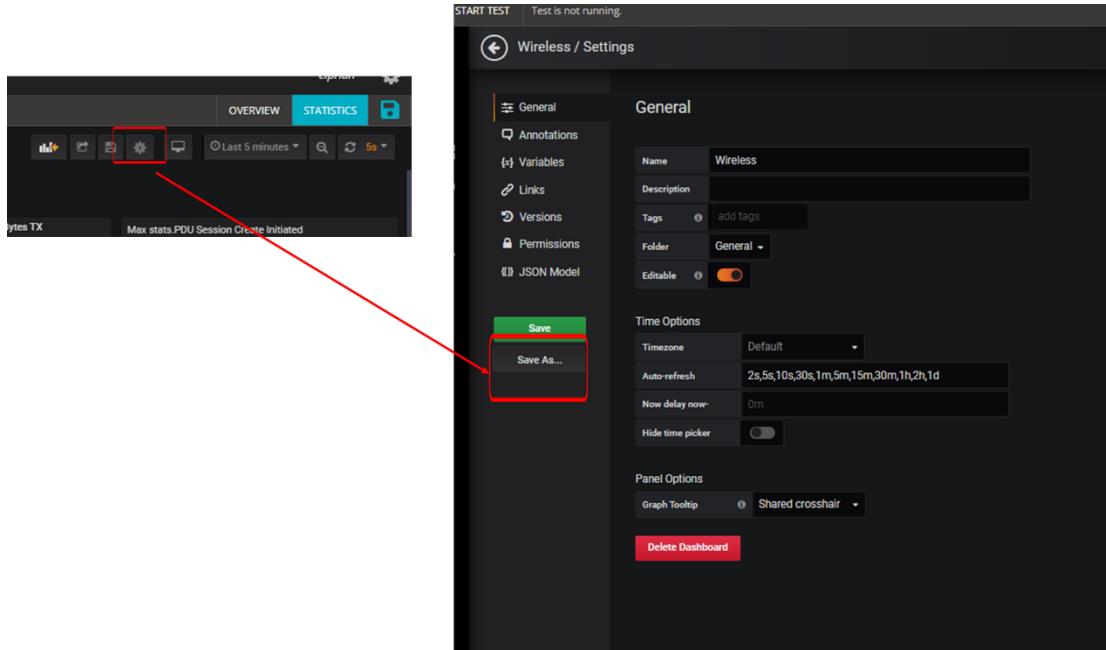
- create a dashboard based on an existing one (a pre-defined dashboard), or
- create a new dashboard and customize it based on your requirements

Create a dashboard based on an existing one

For consistency purposes (in order to keep same look and feel of the upper part of the dashboard), it is recommended to start from an already defined dashboard, save a copy of it and then modify it.

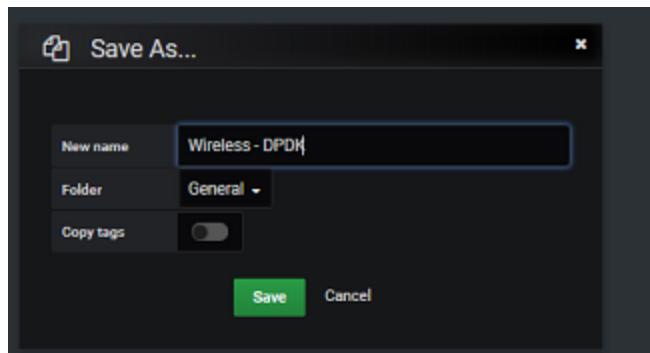
In this example, a new dashboard **Wireless-DPDK**, is created starting from the general one (**Wireless**), as follows:

1. From the Statistics tab, select the Wireless dashboard.
2. Select **View** and press the **ESC** key twice. This will open the top menu and a new side menu to customize dashboards.
3. From the top menu, select the **Settings** icon and, then, select **Save as**.



4. On the Save as window, you can save a copy of the current dashboard (**Wireless**) with a different name, and modify it according to your needs:

- Set the name of the new dashboard to **Wireless-DPDK**.

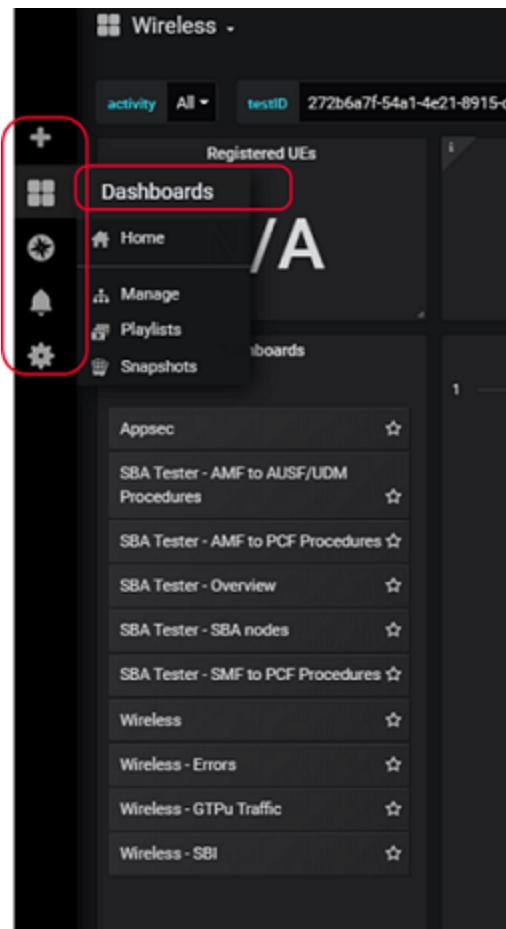


- Select the location of the new dashboard (it is recommended to keep it under the **General** folder, to be on the same list with the default dashboards).
- Select **Save**.

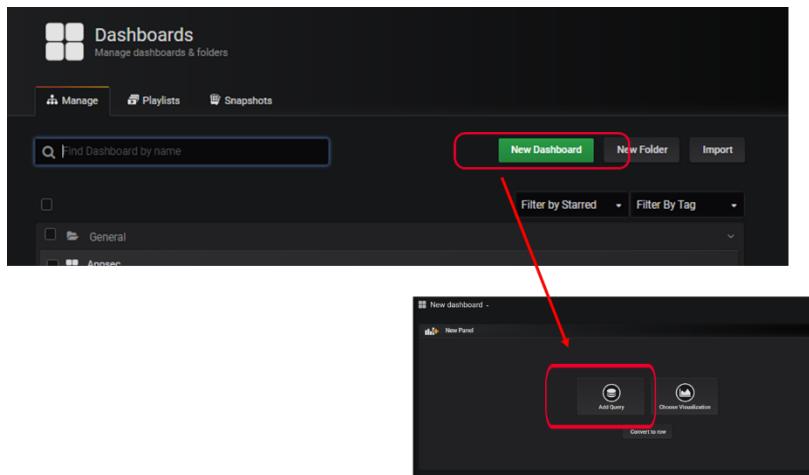
Create a new dashboard

To create a completely new dashboard, do the following:

- From the upper-left menu, select + icon and, then, select **Create a new dashboard**.

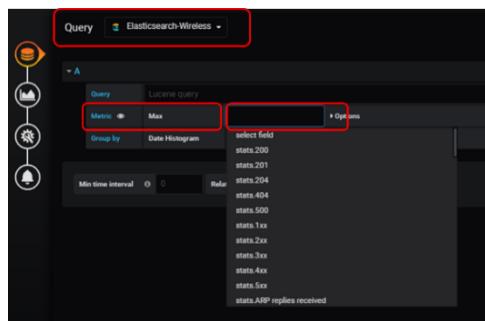


- Select **New Dashboard** and, then, from the New Dashboard window, select **Add Query**.



3. Configure the new dashboard:

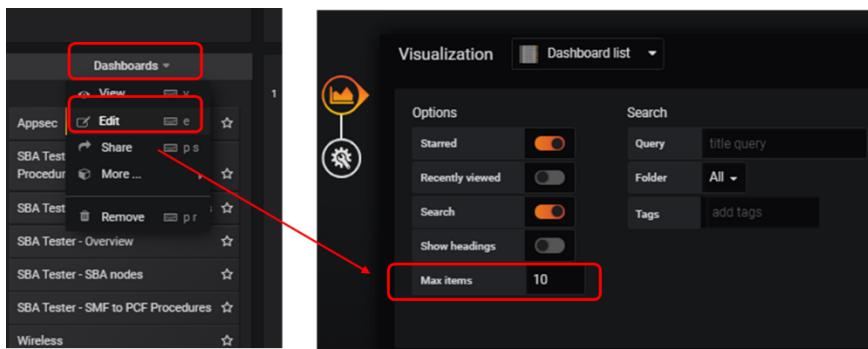
- Make sure you are selecting elastic search for the query type.



- Select **Metric Type : Max**.
- Select the metric you want to display.
- Save the dashboard with a relevant name.

IMPORTANT if the newly defined dashboard is not present in the list of available dashboards, then the number of displayed dashboards must be increased.

To modify the number of displayed dashboards, select **Dashboards > Edit > Visualization**, and change the value of **Max items** to increase/decrease the number of dashboard to be displayed. By default, the **Max items** value is set to 10.



Log Collection

Starting with LoadCore 1.3, logs collections can be easily retrieved from LoadCore UI, as follows:

1. From the LoadCore UI dashboard page, select **Browse Results**.
2. From the Test Results table, select the test for which you want to download the log collection.

The screenshot shows the LoadCore UI interface. At the top, there is a header bar with the title "Test Results" and a note "(Hold down Ctrl. for multiple selection)". Below this is a table with two columns: "Config Name" and "Test Name". The table contains five rows of data. At the bottom of the table is a toolbar with three buttons: "LOAD", "DOWNLOAD", and "DELETE". A red box highlights the "DOWNLOAD" button and its dropdown menu. The dropdown menu contains four items: "CSV", "Report", "Captures & Logs", and "Captures & Logs (Advanced)".

Config Name	Test Name
132 - Copy of SBA Base Config (copy from Apr 23 07:31:42)	Wireless-cb53f2f6-22e9-4d0d-a70c-3083d4cdedb5
131 - fullcore simulation (2021-03-10 11:11:08) (copy from Apr 23 07:28:53)	Wireless-272a4d9d-7a06-4d9e-a12d-a72bf9ae2f43
130 - Copy of SBA Base Config	Wireless-bf24345e-910a-48d9-80c3-b15fac013edc
129 - fullcore simulation (2021-03-10 11:11:08)	Wireless-51b2d520-b01b-4a14-98c3-3ab4d1ca4cb0

3. Select **Download Captures/Logs** and select the download location. This will download an archive containing both MW and Agent logs.

The logs can be also accessed using ssh on MW and Agents (as in the previous versions):

- **Agent logs:**

1. Connect to the agent using ssh (credentials **ixia /ixia**).
2. `sudo tail -f /var/log/syslog`
 - this contains communication between the port - manager on the agent and port manager on the MW machine
3. Check `/opt/5gc-test-engine/lizard-log.txt`.
4. Check `/opt/5gc-test-engine/logs/service.log`.
 - this is the LoadCore log (if you reached this point, this means that the communication between the agents and the MW is already working)

- **Middleware (Cluster) logs:**

Kubectl commands can be done when logged in as root (there is separate procedure to log in as root, not to be confused with console login):

```
kubectl get pods --all-namespaces
```

```
kubectl get pods -n keysight-wap
```

```
kubectl logs podname -n keysight-wap (for example: kubectl logs agent-controller-5566944d54-w8w2k -n keysight-wap)
```

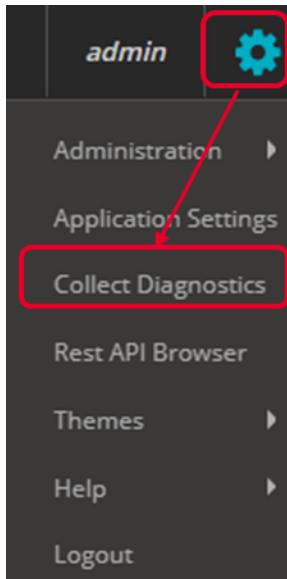
To collect diagnostics run the following command after logging in as console:

```
kcos logs diagnostics collect -a
```

Debug commands

How to collect diagnostics

Starting with LoadCore 2.1, the diagnostics files can be collected automatically from LoadCore UI by selecting the Settings icon and, then, select **Collect Diagnostics**.



An archive will be downloaded will all the diagnostic files from Middleware and agent.

Index

C

customer assistance 3

D

documentation conventions 4

K

keyboard interactions 4

M

mouse interactions 4

P

product support 3

T

technical support 3

touch interactions 4



© Keysight Technologies, 2023

This information is subject to change
without notice.

www.keysight.com