Keysight Open RAN Simulators, Cloud Edition 5.2

Deployment Guide

KEYSIGHT

202511271155+02:00

Notices

Copyright Notice

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 these 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.

Contact us

Keysight headquarters

1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738
Email address: support@keysight.com
Website: https://support.keysight.com/

Support

Location	Phone number	Local time
Americas		
US, Canada	1-888-829-5558	8h00 – 17h00
Brazil	0800-892-0522	8h00 – 17h00
Mexico	001-888-829-5558	8h00 – 17h00
Other	+1-719-273-6516	8h00 – 17h00
EMEAI		
Belgium	0800-18686	8h30 – 17h30
Finland	0800-913-352	8h30 – 17h30
France	0800-917228	8h30 – 17h30
Germany	0800-0824099	8h30 – 17h30
India	1800-18-02552	8h30 – 17h30
Ireland	1800-949245	8h30 – 17h30
Israel	1-809-454975	8h30 – 17h30
Italy	0800-790571	8h30 – 17h30
Luxembourg	0800-25112	8h30 – 17h30
Netherlands	0800-022-9086	8h30 – 17h30

Romania	0213 015 699	8h30 – 17h30
Spain	800-654386	8h30 – 17h30
Sweden	0201-202266	8h30 – 17h30
United Kingdom	0800-0293882	8h30 – 17h30
Asia and Australia		
Australia	1-800-370-558	8h30 – 17h00
China Mainland	800-810-0005	8h30 – 17h30
	400-810-0005	8h30 – 17h30
Hong Kong	800-931-613	9h00 – 18h00
Japan	0120-421-621	9h00 – 17h30
Malaysia	1800-819 092	8h30 – 17h30
South Korea	080-770-0800	8h30 – 17h30
Singapore	800-101-3797	8h30 – 17h30
Taiwan	0800-699-880	9h00 – 18h00
Other	+65 6215 7600	8h30 – 17h30 (Singapore)

Last updated: 11 September 2025

Table of Contents

Contact us **3**

Chapter 1 Test Methodologies for 5GC **9**

Chapter 2 ORAN-SIM CE Deployment on VMs **10**

VMware ESXi Deployment 10

Prerequisites 10

Hardware Requirements 11

ORAN-SIM CE Installation Procedure 12

Middleware Installation 12

Register a new user 16

Logging in with administrative rights 17

Agent(s) Installation 19

License Manager Installation 25

Amazon AWS Deployment 33

Prerequisites 33

Resource requirements in AWS 34

AWS services and components 34

Create a Virtual Private Cloud 35

Create the management and test subnets 36

Create the Internet Gateway 38

Assign traffic routes 39

Configure Security Groups 40

Create Key Pairs 42

LoadCore Components Installation 42

Middleware Installation 44

Agent(s) Installation 49

LoadCore Agent configuration for Application Traffic 56

License Server Installation 62
Software Upgrades 67
Troubleshooting 68
Monitor the health of the application 69
Backup and recovery 69
AWS Security Best Practices 70
KVM Deployment 71
Prerequisites 71
Hardware Requirements 71
Enable Virtualization 73
Install Linux Distribution 75
Install and prepare KVM 77
Install Gnome and XRDP 80
Install LoadCore VMs 81
Add virtual bridge 85
Azure Cloud Deployment 87
OpenStack Deployment 89
Prerequisites 89
Hardware Requirements 89
Middleware installation 91
Agent installation and configuration 95
License Server installation 99
PCI Passthrough and CPU Pinning 102
HEAT Templates 105
Appendix A Enable PCI Passthrough for KVM deployments 107
PASSTHROUGH MODE ON HOST LINUX 107
KVM VM PASSTHROUGH MODE 108
Appendix B Deploying Open RAN SIM CE with static IP Addresses 112
Middleware 112
Agents 112
Appendix C Ports used in ORAN-SIM CE communication 115

Appendix D CPU Pinning 116
Appendix E Configure SRI-OV with Network Manager 118
Chapter 3 ORAN-SIM CE Deployment on Containers 120
Amazon AWS EKS Deployment 120
How to get an authentication token to be able to push the images into ECR 120
How to push ORAN-SIM CE container images into AWS EKS 121
How to install ORAN-SIM CE Middleware in EKS 121
How to install ORAN-SIM CE Agents on EKS 122
How to uninstall ORAN-SIM CE components 123
OpenShift Deployment 123
Supported NICs in tests with IxStack over DPDK/Raw(TCP based traffic): 125
ORAN-SIM CE Deployment on k8s 125
ORAN-SIM CE Deployment on GKE Anthos 128
Chapter 4 Test Case #1: gNB and 5G Core simulation in B2B scenario 132
Test Configuration 132
License server configuration 132
Configure the test from scratch 133
Import a test configuration 150
Run a test 152
Results Analysis 154
Downloading results, logs and captures 156
Generate and retrieve csv statistics using Rest API 157
Generate and retrieve packet captures 160
Running Application traffic (HTTP) 164
Installing additional interfaces on the test agents 165
Enabling IxStack over DPDK/Raw 166
Configure HugePages on both Agents 167
Application traffic configuration 168
Create Custom Dashboard 170
Log Collection 173
Debug commands 174

Index **175**

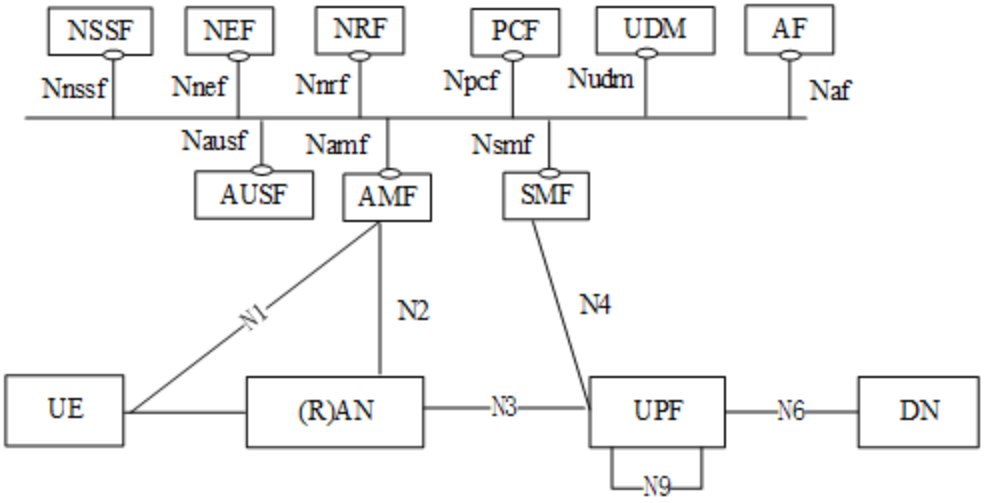
CHAPTER 1

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

ORAN-SIM CE Deployment on VMs

This section contains the following topics:

VMware ESXi Deployment **10**

Amazon AWS Deployment **33**

KVM Deployment **71**

Azure Cloud Deployment **87**

OpenStack Deployment **89**

Appendix A Enable PCI Passthrough for KVM deployments **107**

Appendix B Deploying Open RAN SIM CE with static IP Addresses **112**

Appendix C Ports used in ORAN-SIM CE communication **115**

Appendix D CPU Pinning **116**

Appendix E Configure SRI-OV with Network Manager **118**

VMware ESXi Deployment

This section describes the steps needed to deploy ORAN-SIM CE 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.
- **ORAN-SIM CE images** (three images: Middleware, Agent and License server). License Server image is optional, since it can be collocated with MW, but it is recommended to be installed on a separate VM. The images can be downloaded from KSM: https://ksm.software.keysight.com

  IMPORTANT IP connectivity between VMs must to be ensured.
- You will need to use this machine to run ORAN-SIM CE tests. ***ILU (Windows or Linux) cannot be used to run ORAN-SIM CE tests.***
- **Valid licenses** for License server. For deploying full 5G core and run Control plane and User plane traffic, the following licenses will be required:
  - Control plane licenses:
    - ORAN-SIM CE 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 ORAN-SIM CE 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
- 16 GB RAM
- 256 GB SSD

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.
- 32 GB SSD

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

- 8 x vCPUs
- 32 GB RAM
- 32 GB SSD

Licensing server default resources

- 4 x vCPUs
- 8 GB RAM
- 150 GB SSD

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 X710 10G
- Intel XXV710 25G
- Intel XL710 40G
- Intel E810 25G
- Intel E810 100G
- Mellanox ConnectX-4/5 25G
- Mellanox ConnectX-4/5 100G
- Mellanox ConnectX-6 Dx 100G
- VMWare ESXi Virtual NIC (vmxnet3)

Supported drivers list:

- vmxnet3
- mlx4_core
- mlx5_core
- ixgbe
- ixgbevf
- i40e
- i40evf
- iavf
- ice

ORAN-SIM CE Installation Procedure

ORAN-SIM CE

The ORAN-SIM CE installation procedure requires the following steps:

1. ORAN-SIM CE Middleware installation and configuration.
2. Agent(s) installation and configuration.
3. License Server installation and configuration.

IMPORTANT

The recommended order for booting up the VMs is the following:

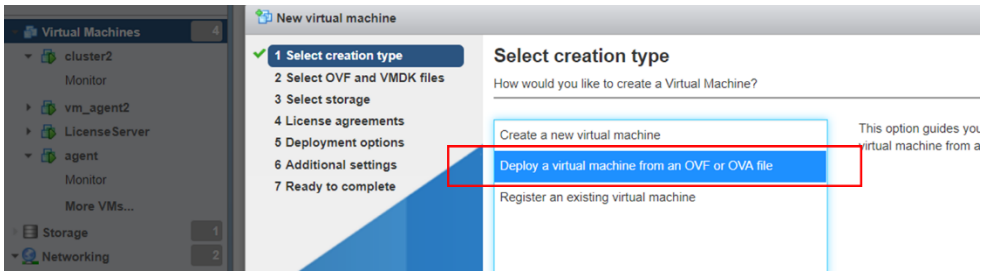
- first, the License Server, if a dedicated one has been deployed (this is the VM that uses the least amount of resources)
- next, the MW
- last, after a few minutes, start the agents one by one.

This order is also according to dependencies/communication between the VMs.

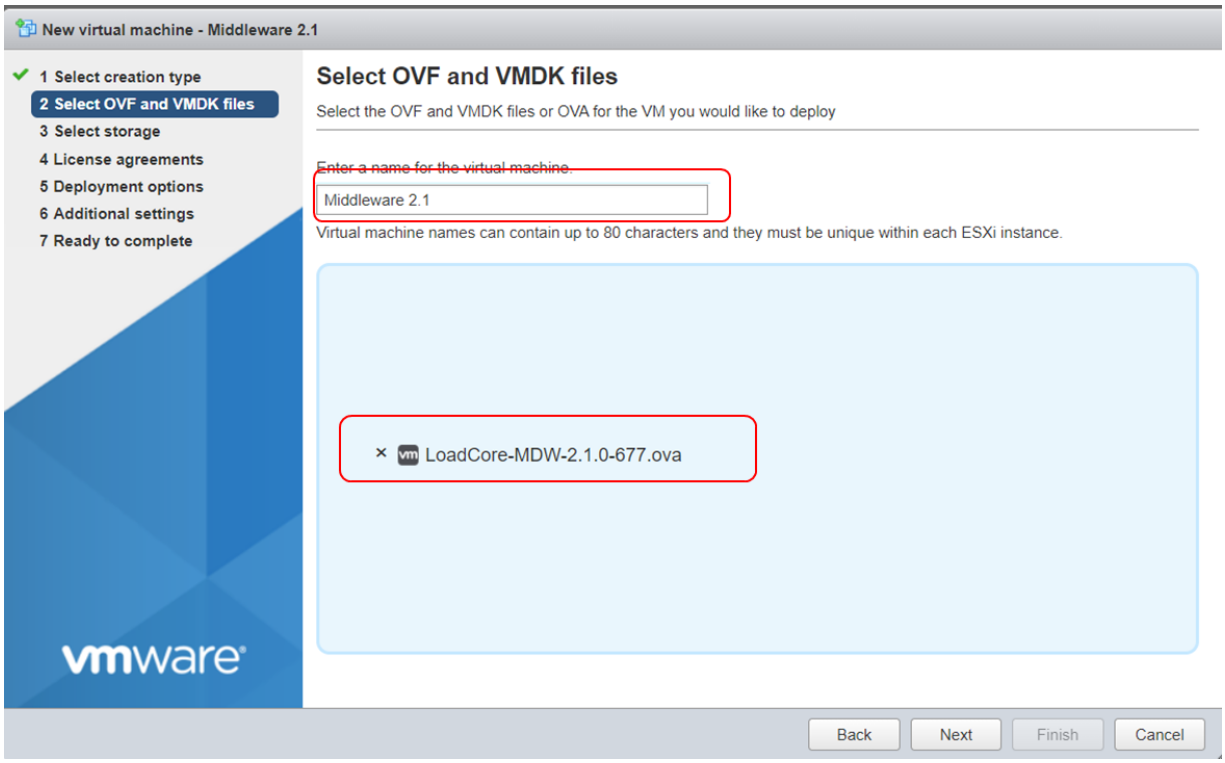
Middleware Installation

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.



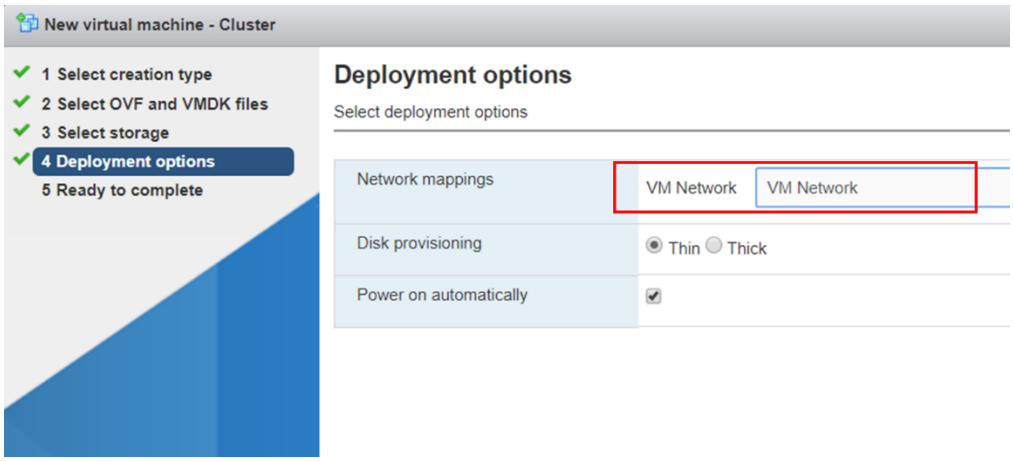
2. Define a name for the VM machine. In this example, the name is **Middleware 2.1**.
3. Select the OVA image and select **Next** > **Next**.



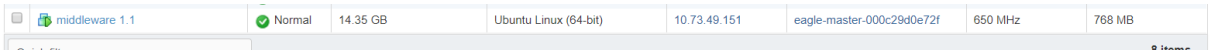
4. Select the network (vSwitch) where the Middleware should connect. By default, this is the **VM Network**.

   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 represents the Agents.

   NOTE Management interfaces (connecting MW and agents) support IPv6 DHCP, but not SLAAC.



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 ORAN-SIM CE 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:

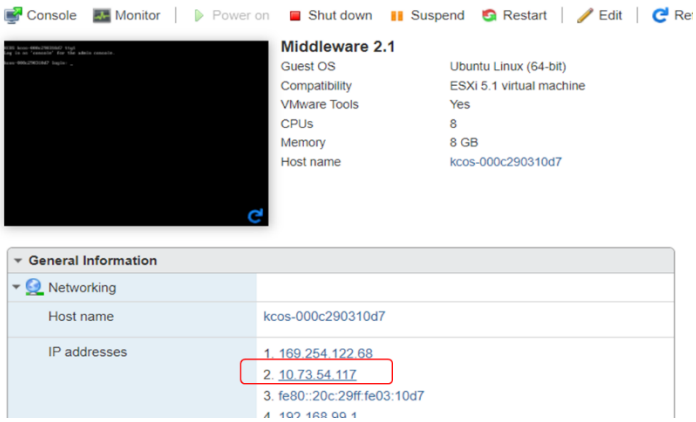
```
-rw-rw-r-- 1 root root 17G Sep 24 12:37 LoadCore-MW-w1.3.0-981.qcow2
```

Inside ORAN-SIM CE 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
```

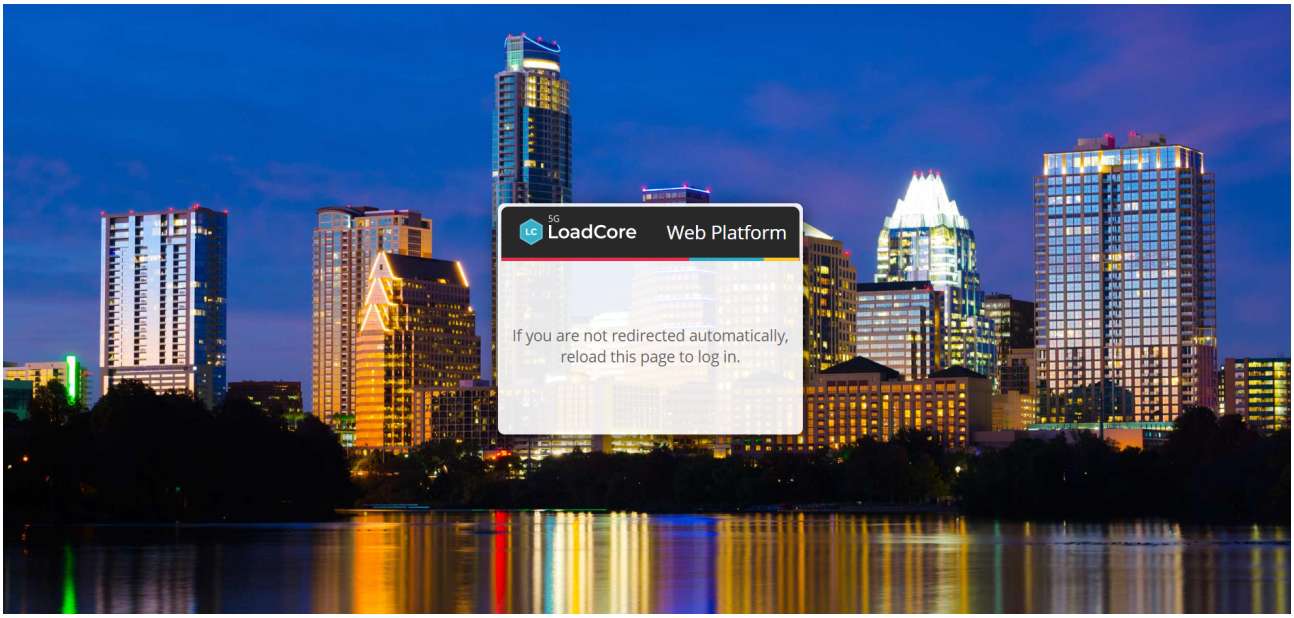
After the Middleware image is deployed, the next step is to access the ORAN-SIM CE UI.

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



Connect to the web 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:

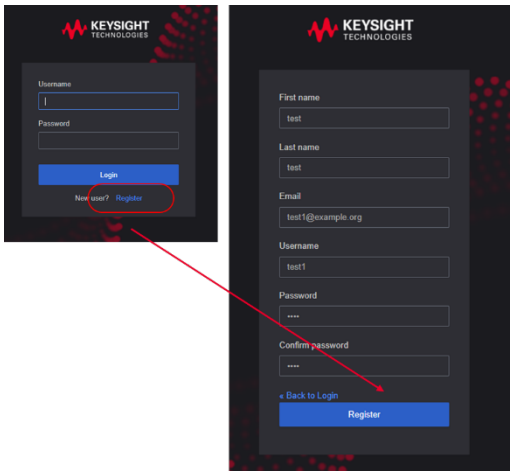
1. Create a new user using the **Register** link.
2. Use the default administrator account : **admin**/**admin**.

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.



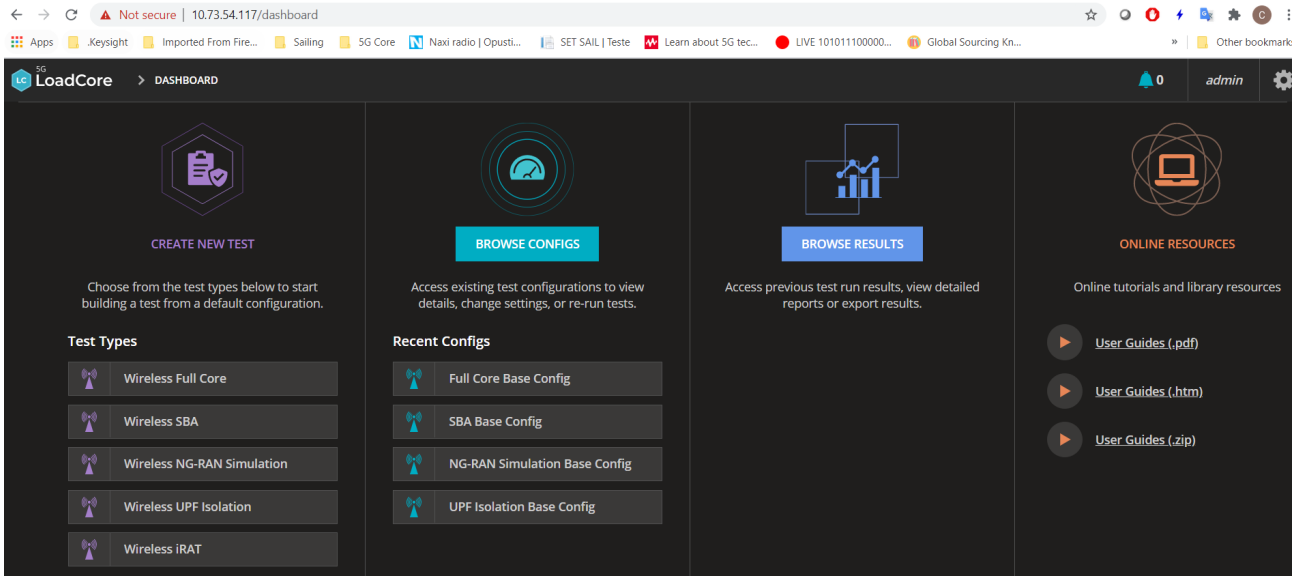
2. To create the user, select **Register**. You will be logged and redirected to the main LoadCore screen.

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.

Logging in with administrative rights

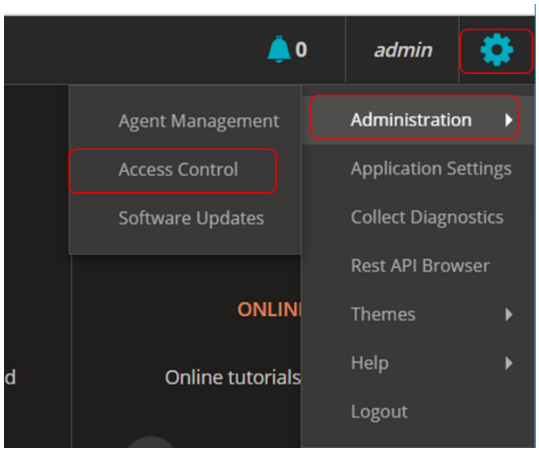
To log in with administrative rights use the following credentials: **admin**/**admin**.

This will log you to the ORAN-SIM CEsystem.

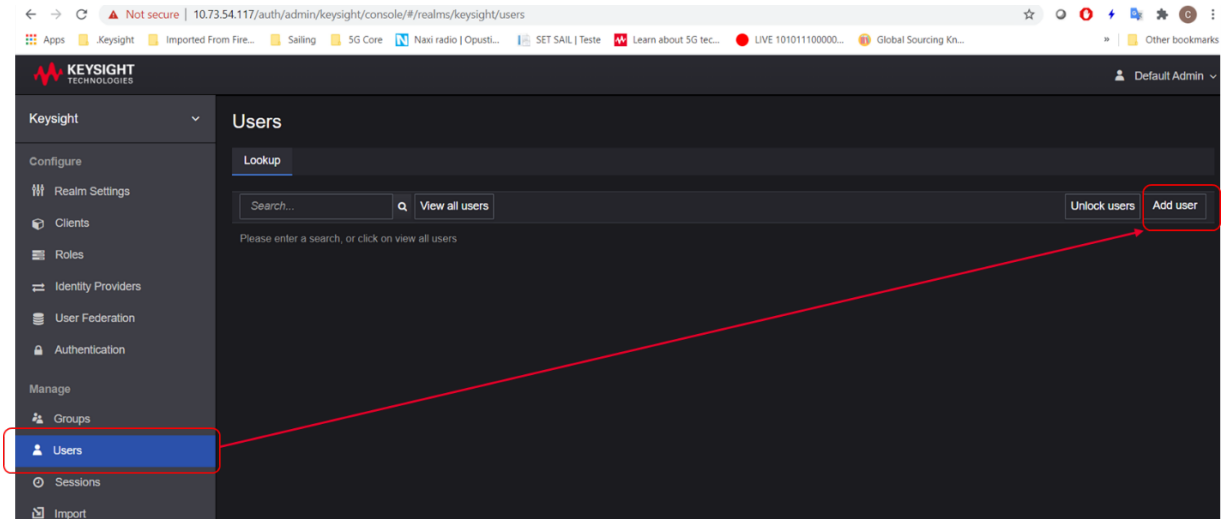


Using this admin rights, you can create the users to access the system, as follows:

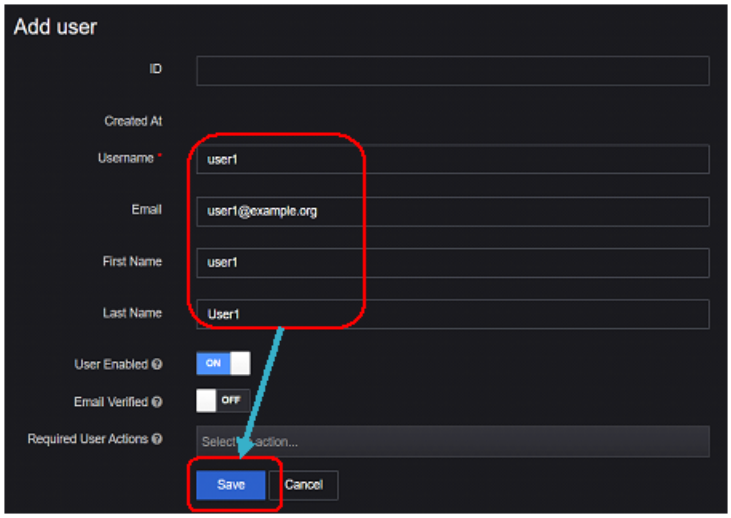
1. Select the **Settings** icon > **Administration** > **Access Control**.



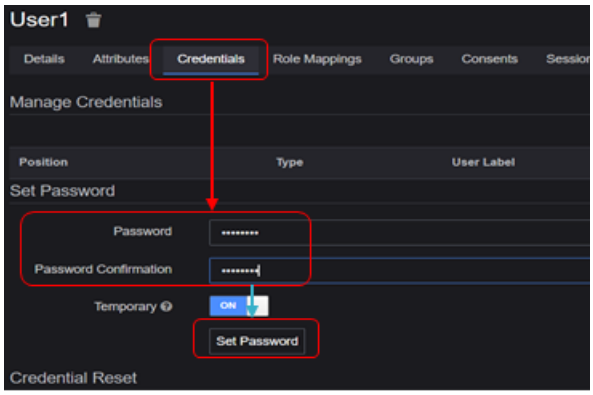
2. The Access Control window opens. Select the **Users** tab and, then, **Add User**.



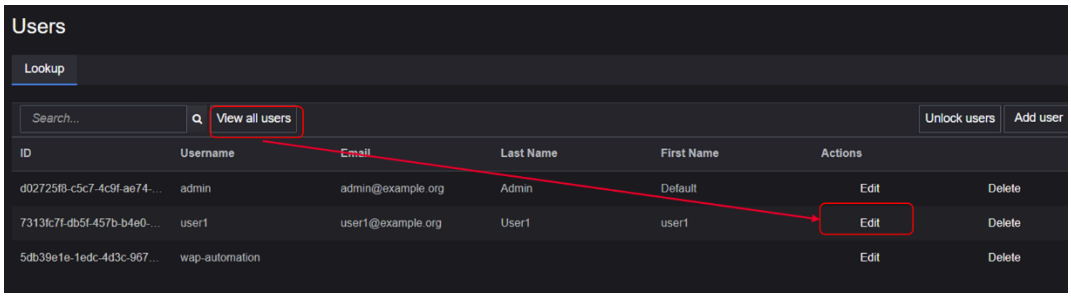
3. Define a user with its corresponding details (in this example, the new user is **user1**).



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



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



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
```

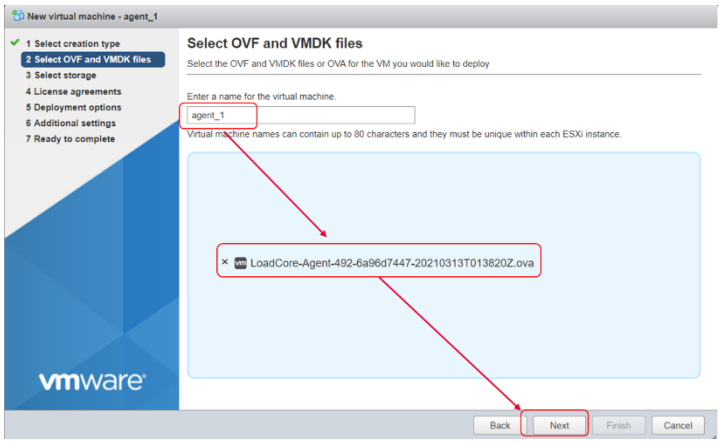
Agent(s) Installation

ORAN-SIM CE 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.

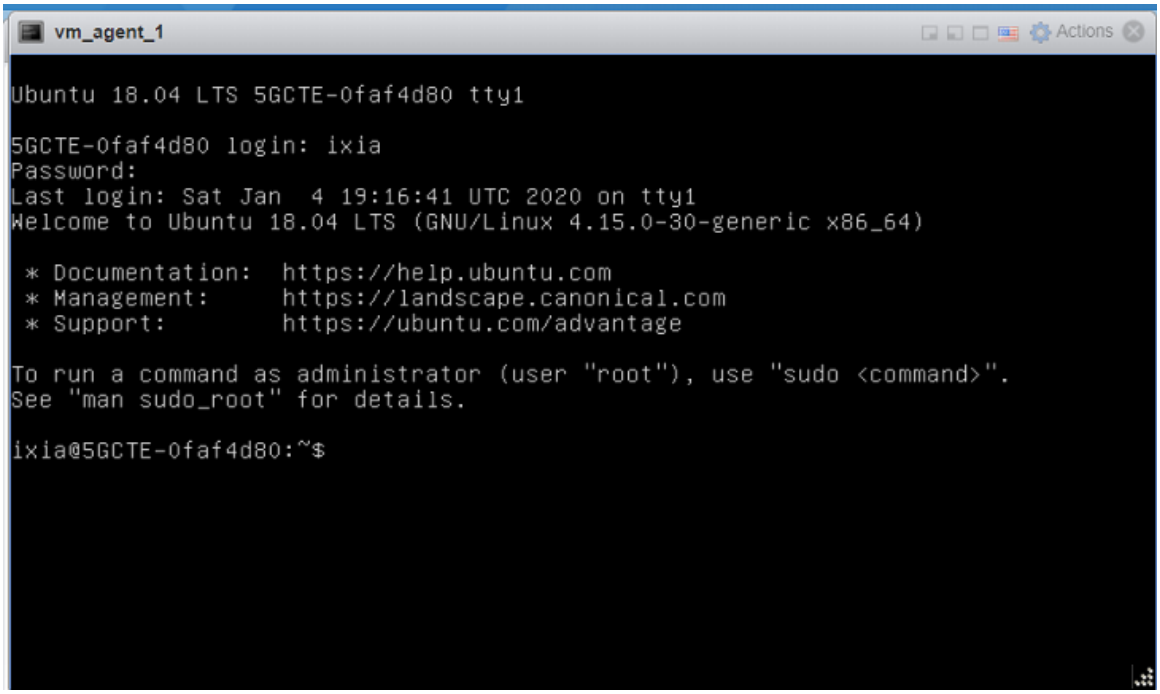
For this deployment, two agents will be installed.

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

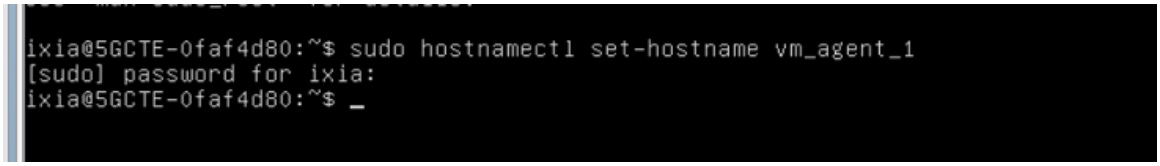
1. Select the agent image when the VM is created.
2. Set a different name for each VM machine (in this deployment **agent_1** is used for the first agent and **agent_2** for the second agent).



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**



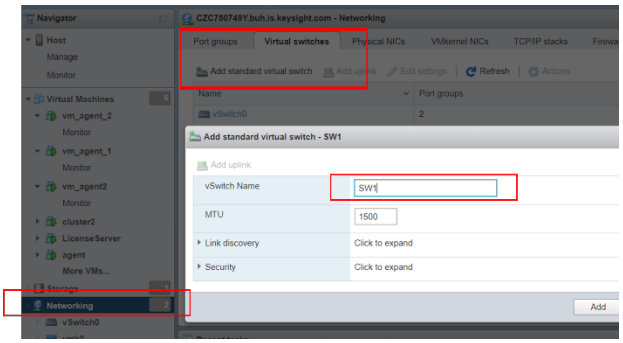
   b. Set the hostname to **agent_1** (use **agent_2** as the name for the second agent).



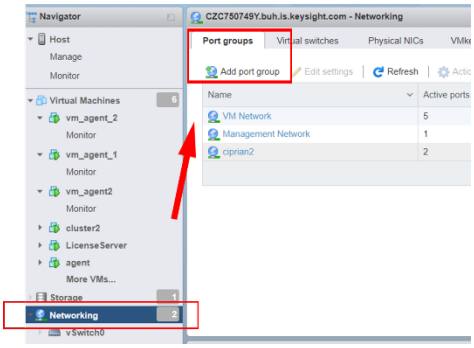
The agents should connect with the Middleware VM on one side and between them on the other side. The second connection represents the **test network**.

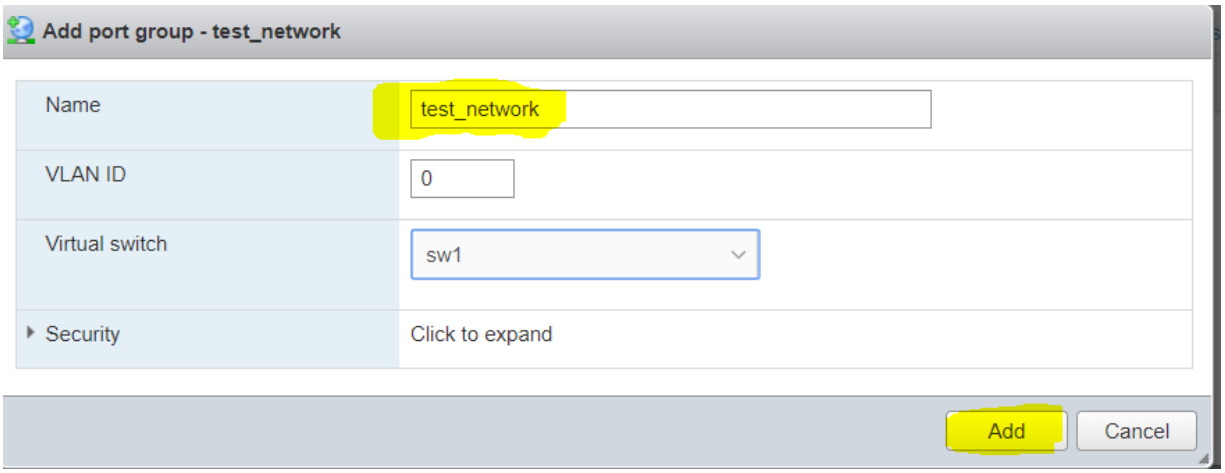
In order to do this, another NIC must be added to each VM agent, and connect it to a new vSwitch and PortGroup (**sw1 / test_network**), as follows:

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



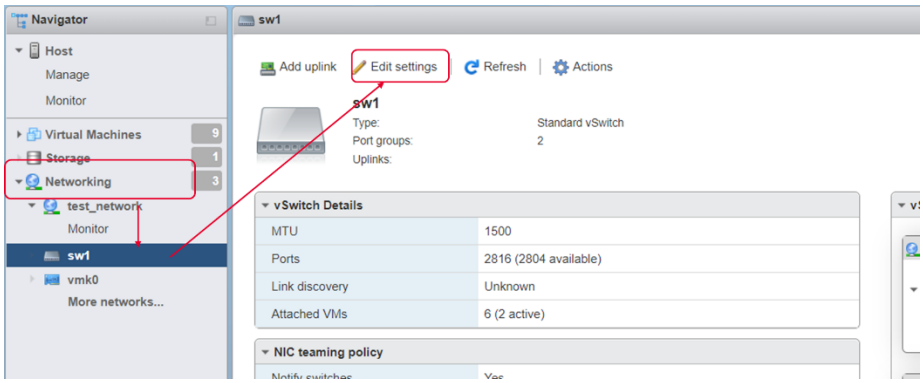
2. Select **Networking** > **Port Groups** > **Add port group** and set the port group name to **test_network**.





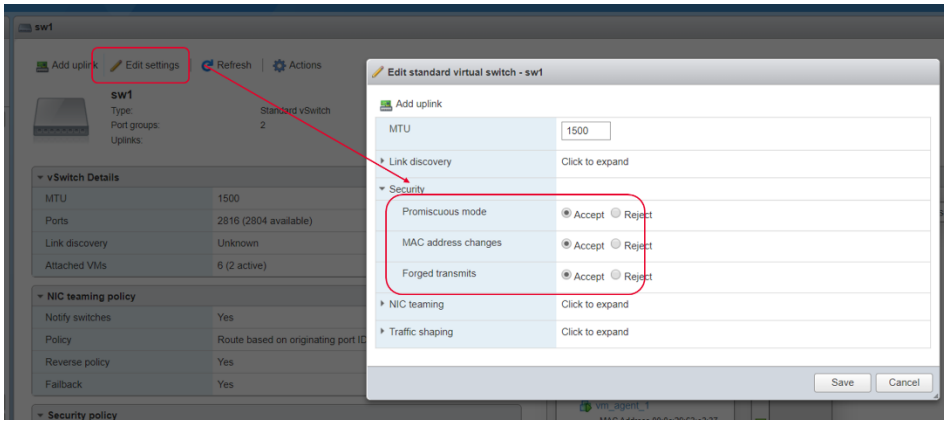
Since the are multiple IP addresses on the same interface, the **Promiscuous mode** on the vSwich that is hosting the *test_network group* must be enabled (in the current setup, *sw1* is the vSwitch that has *test_network group*). This can be done as follows:

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

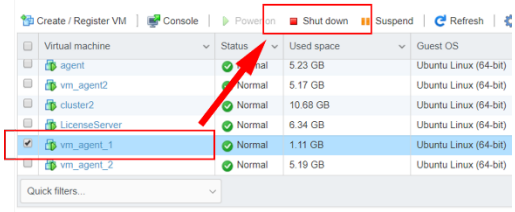


IMPORTANT In case you running Openstack on vmWare with VLANs enabled, you need to enable Promiscuous mode on the vswitches (from Reject to Accept) and set VLAN ID 4095 (all vlans).

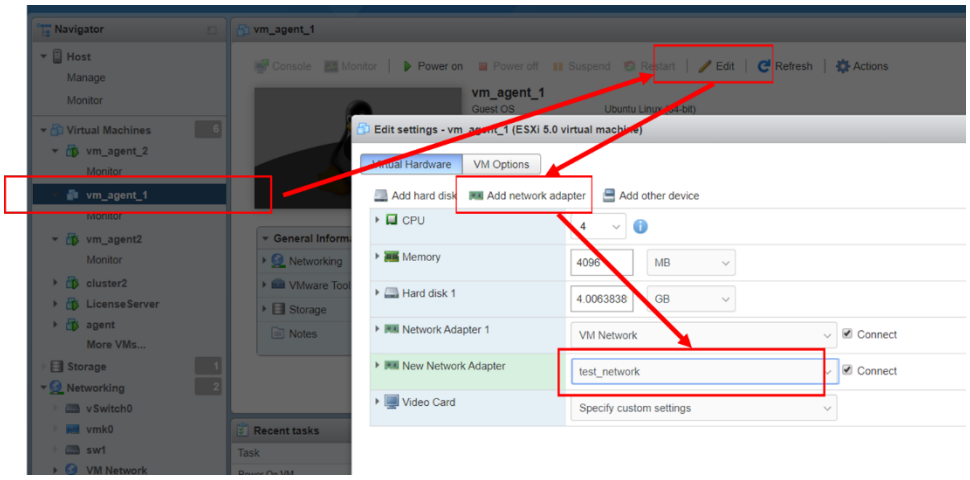
2. From the Security section, enable Promiscuous mode and MAC address change.



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



5. 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**)

NOTE Make sure to change the value from default driver to *vmxnet*.

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:fef3: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:fef3: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 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 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

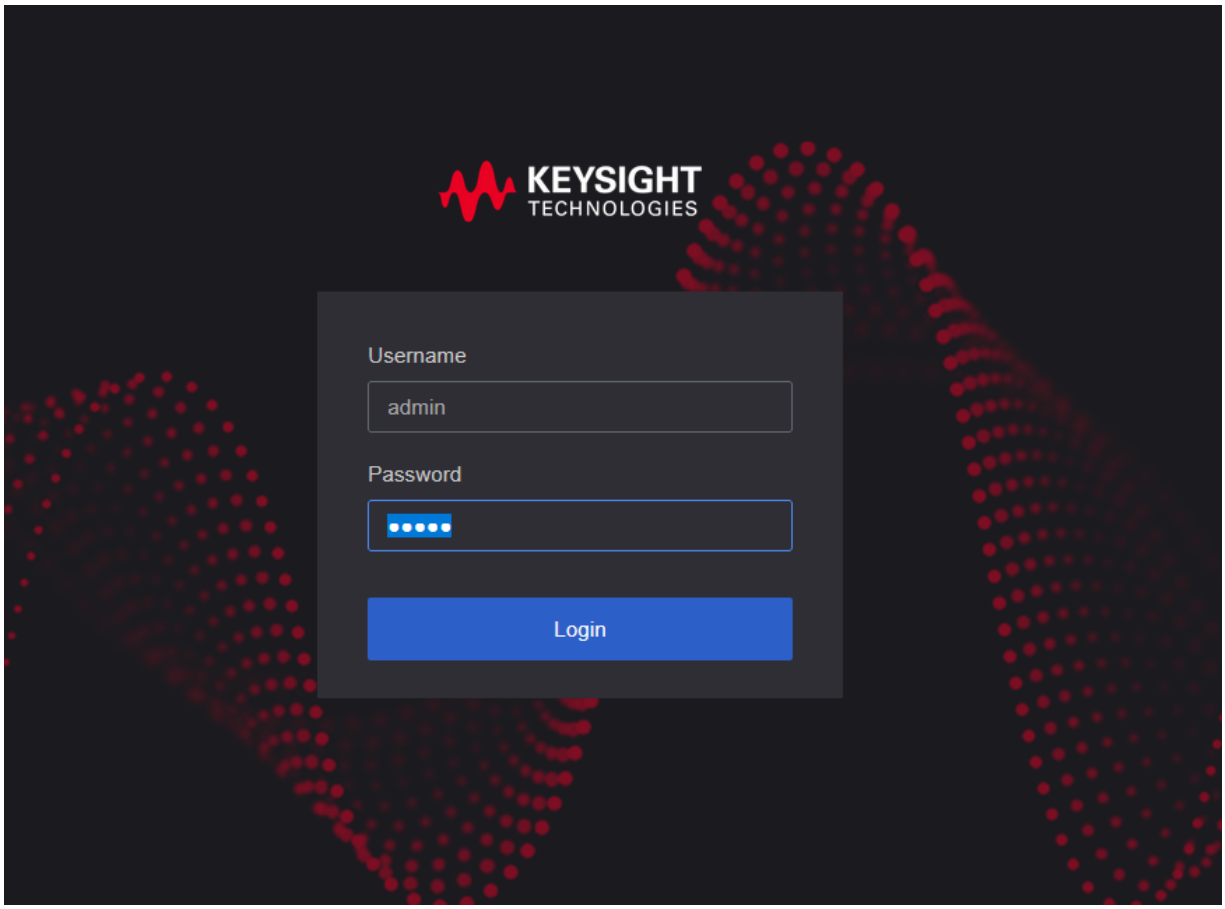
To install the License server on a separate VM machine, do the following:

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 it's 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

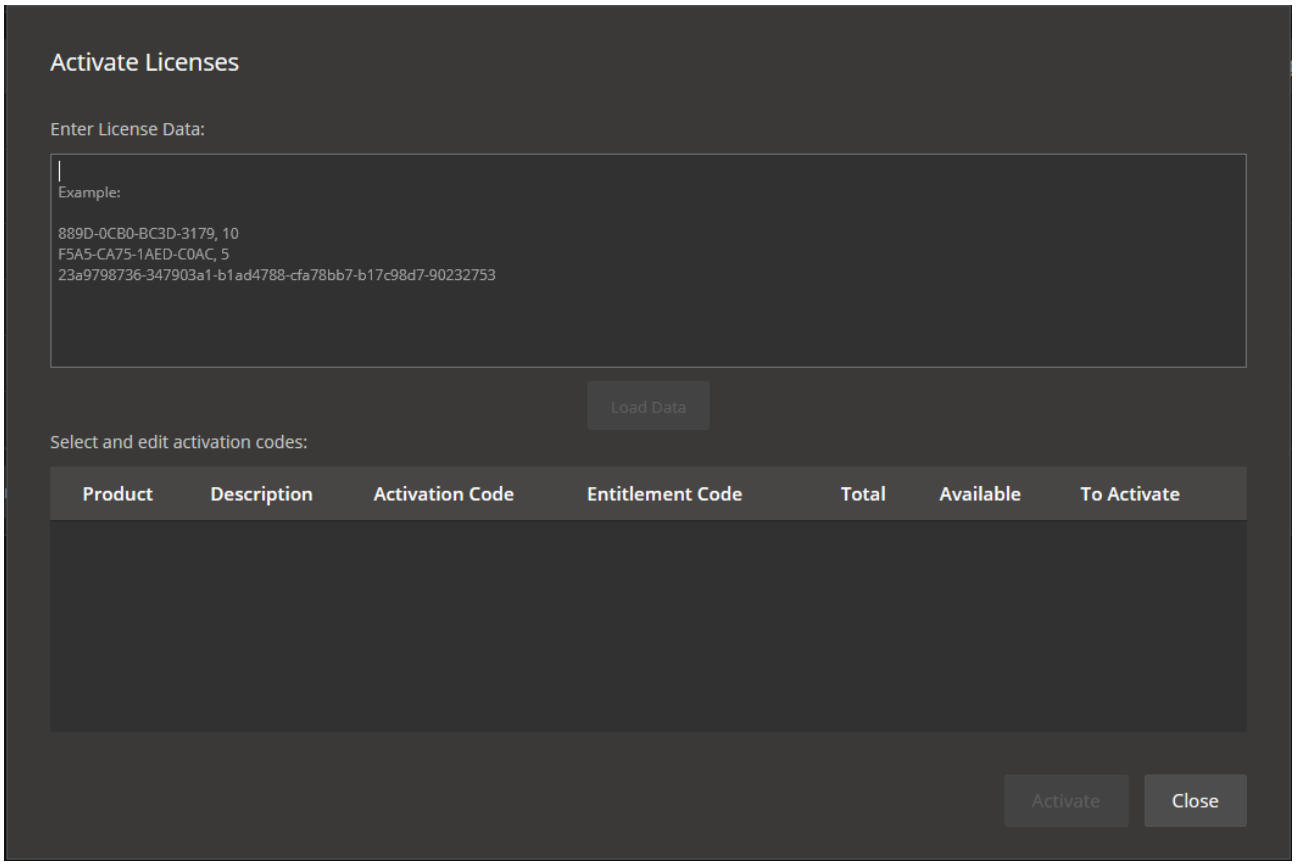
Now you can access the License Server UI and install the licenses:

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**.

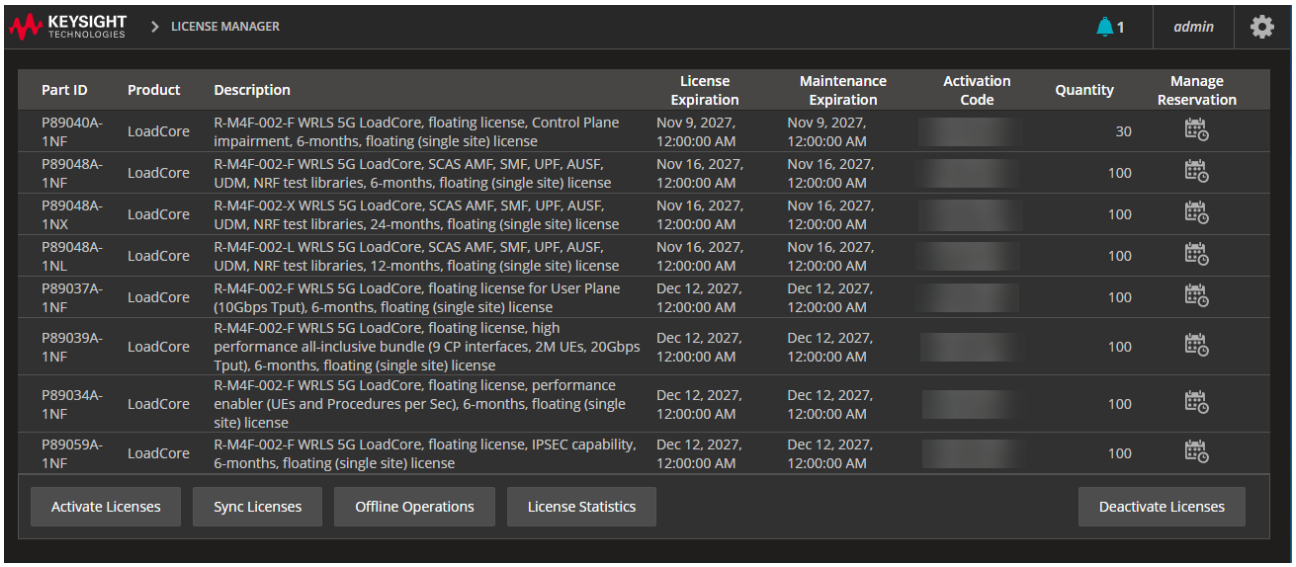


At this point, there are no licenses available on the newly installed server, therefore you must add the license codes you received (evaluation licenses can be obtained by sending an email to: Software-Eval-Request IX <software-eval-request.ix@keysight.com>.

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

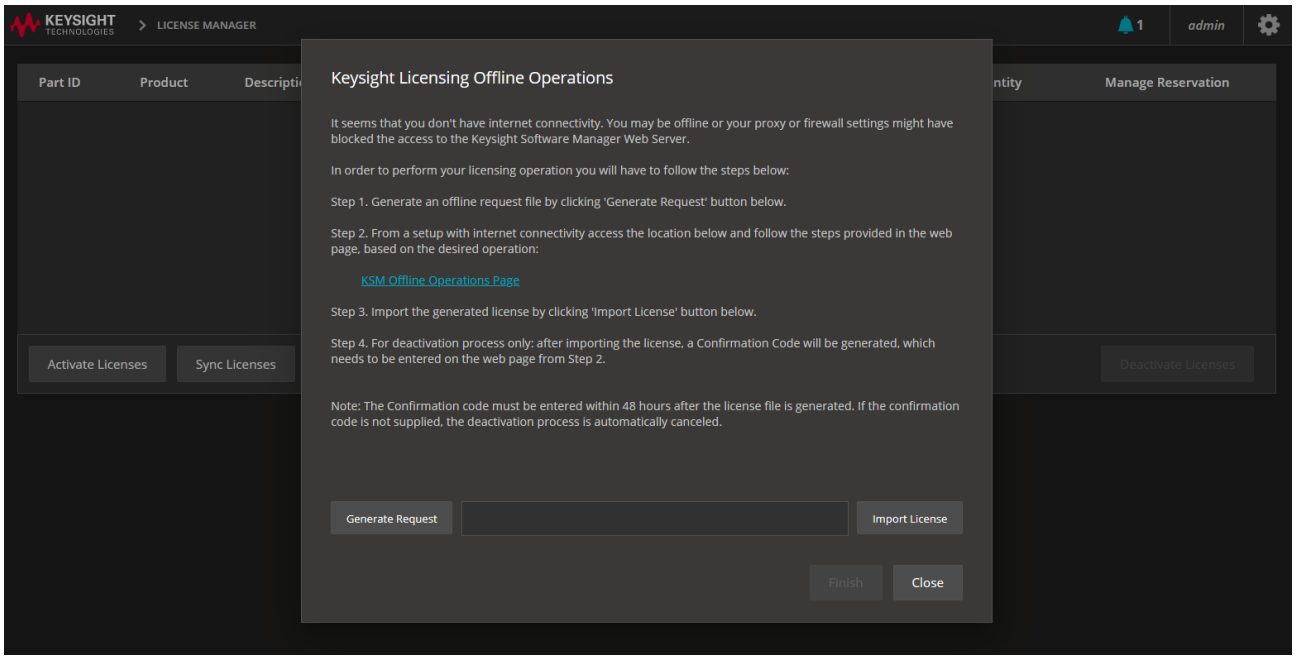


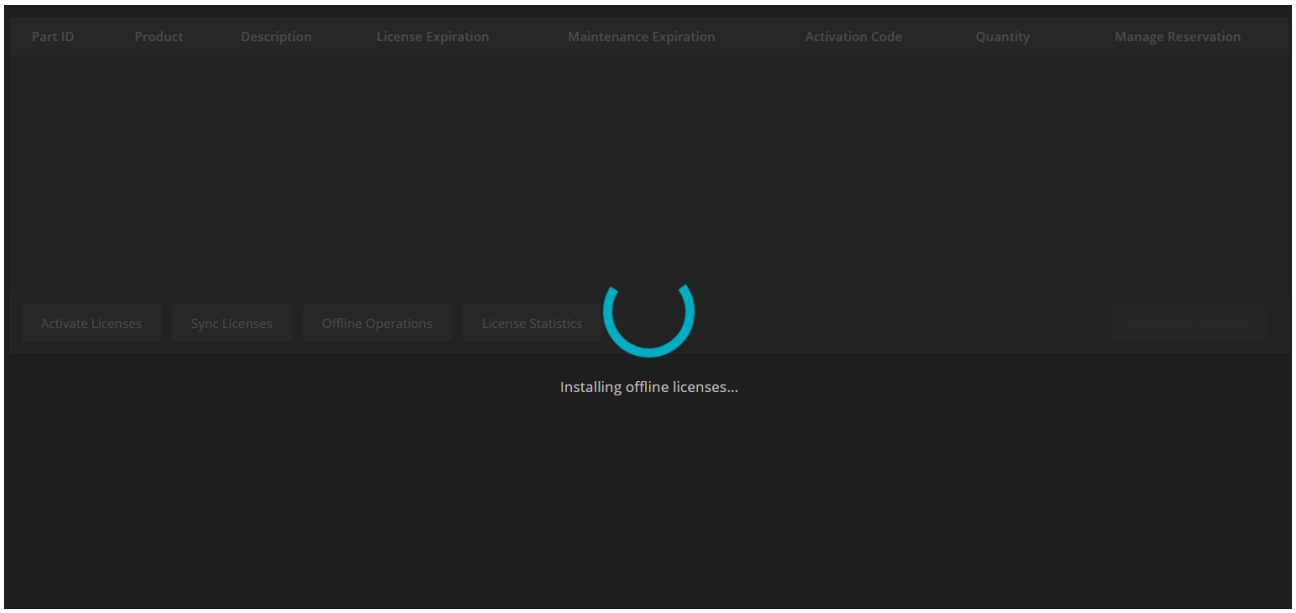
After activation, for each license, details are displayed, as shown below:



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.

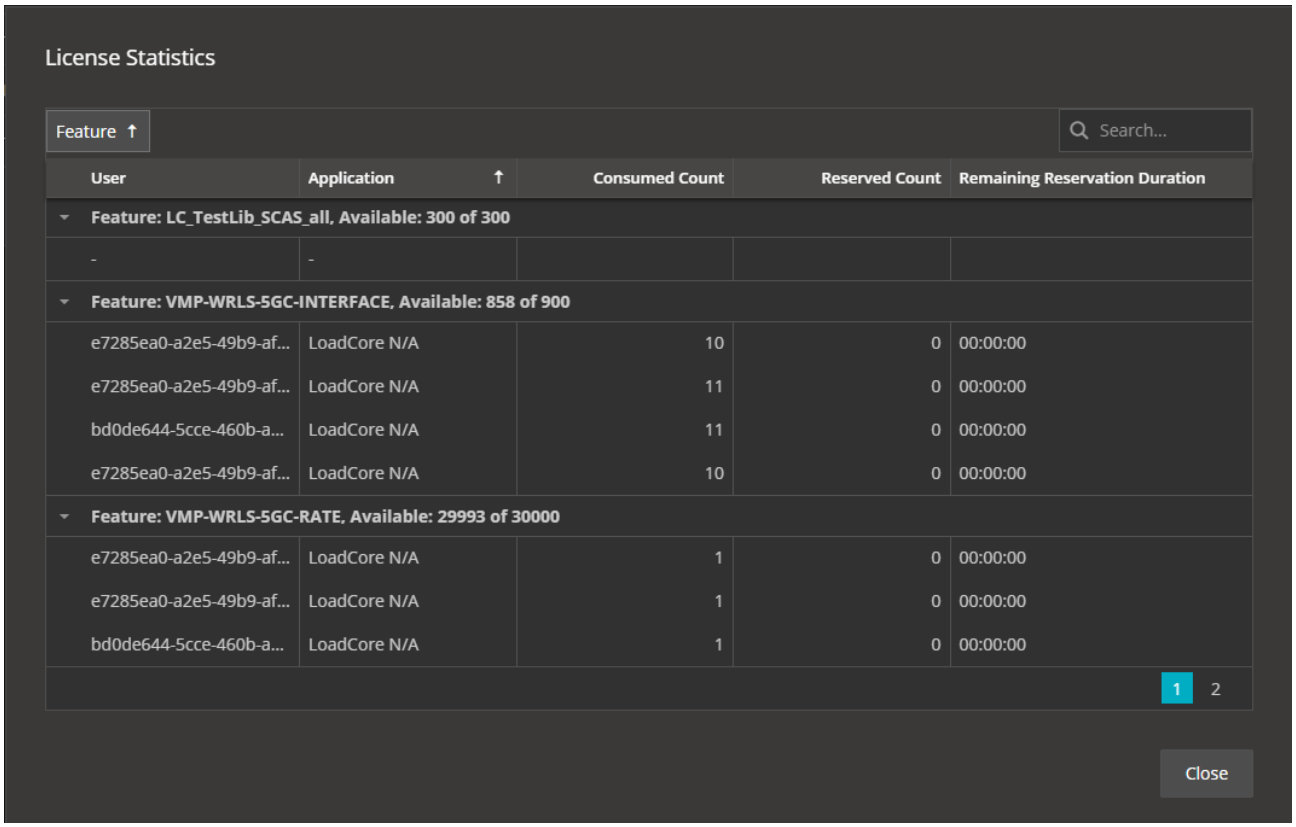
To import licenses, go to **Offline Operations**, select **Import License** and select the file that contains the licenses.





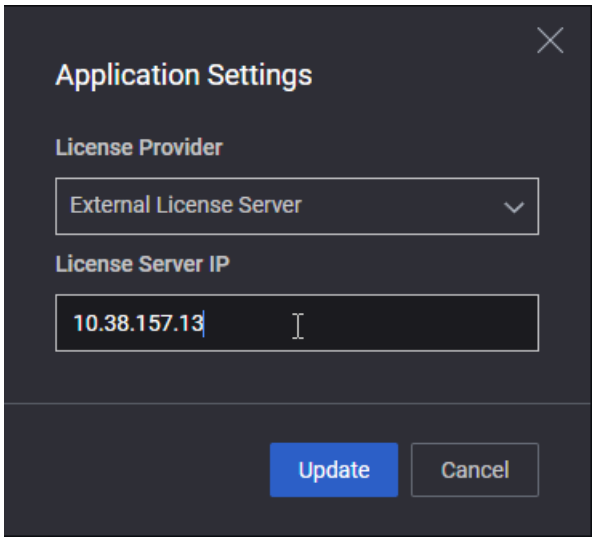
To deactivate licenses, use the **Deactivate Licenses** button. Set the **New Quantity** value for each license and, then, select **Perform Deactivation**.

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.

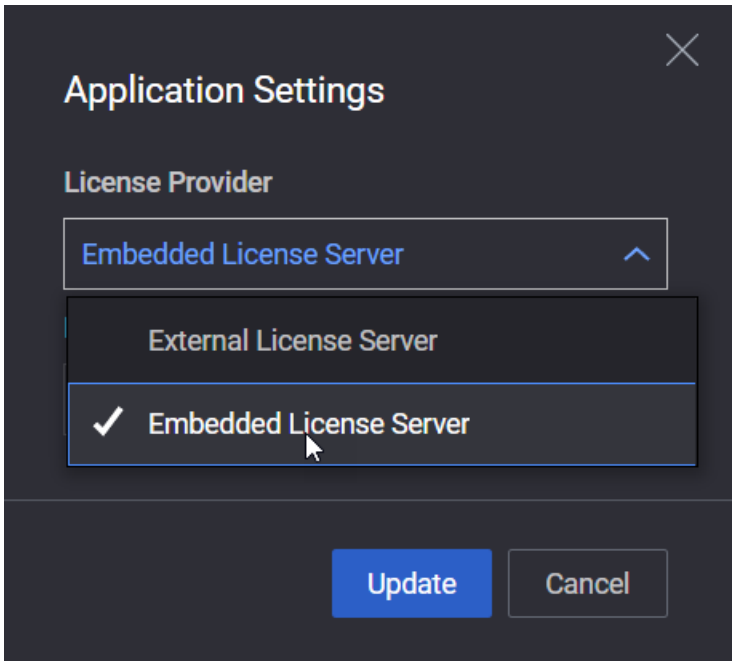


ORAN-SIM CE allows you to use two types of license servers in order to manage licenses. From the ORAN-SIM CE Web UI select gear menu > **Application Settings**:

- **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 ORAN-SIM CEMW.



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 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 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 ORAN-SIM CE 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).
- **ORAN-SIM CE 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 ORAN-SIM CE 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 License server. For deploying full 5G core and run Control Plane and User Plane traffic, the following licenses are required:
  - Control plane licenses:
    - ORAN-SIM CE 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

ORAN-SIM CE 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 SSD

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

ORAN-SIM CE 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.

- 32 GB SSD

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

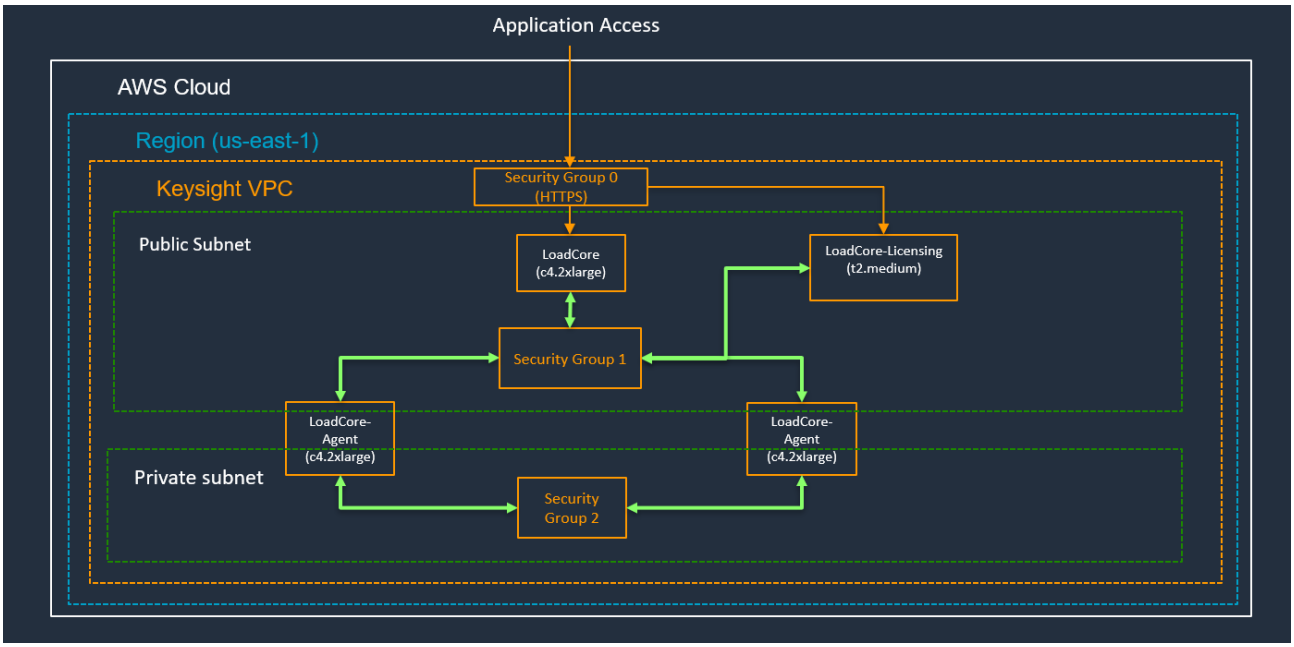
- 2 x vCPUs
- 4 GB RAM
- 11 GB SSD

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:



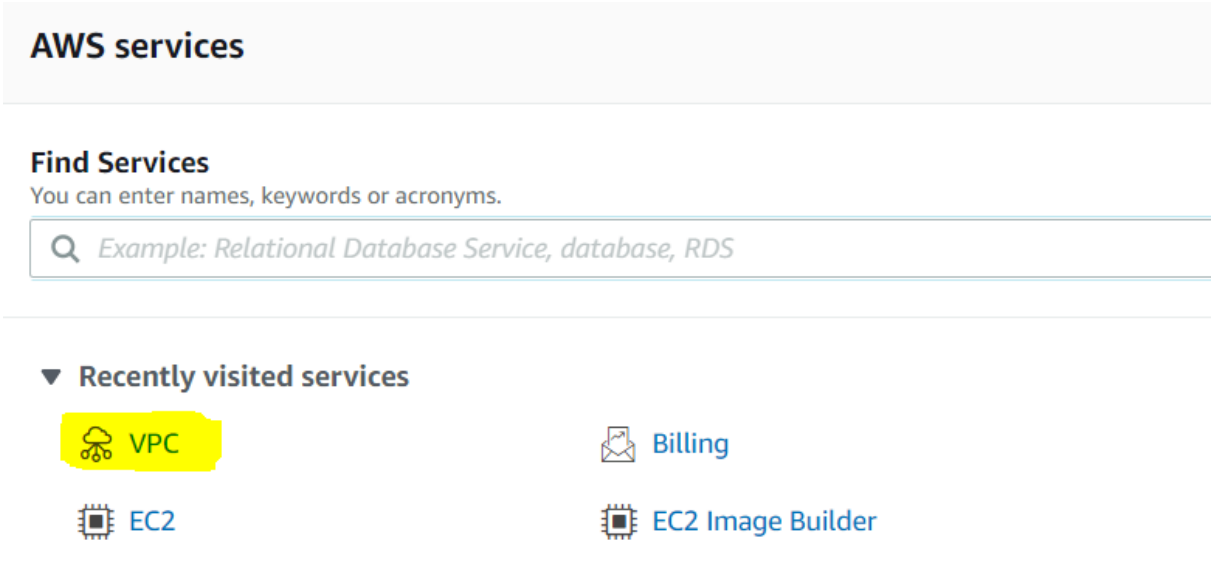
For LoadCore deployment, you can reuse an already existing VPC or create a new one.

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

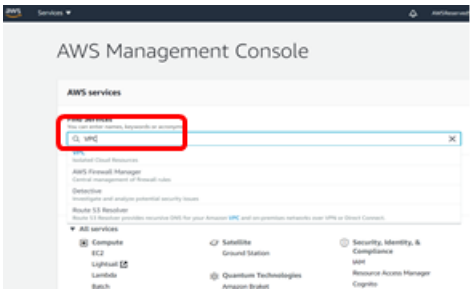
Create a Virtual Private Cloud

You can create an Amazon Virtual Private Cloud as follows:

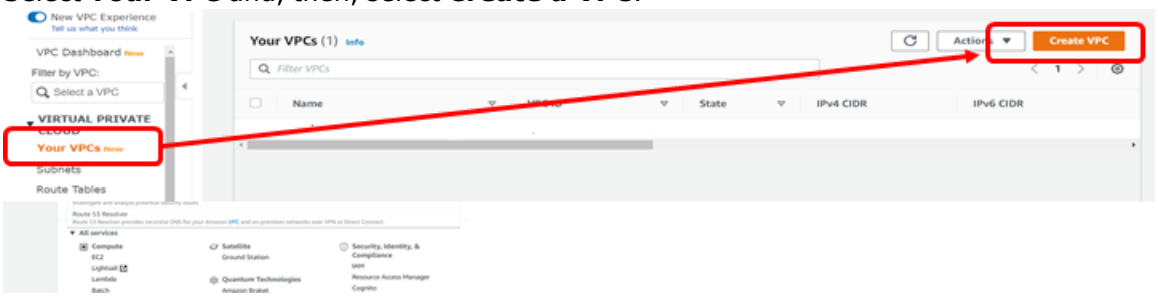
1. Select **AWS Management Console** > **Services** > **VPC**.



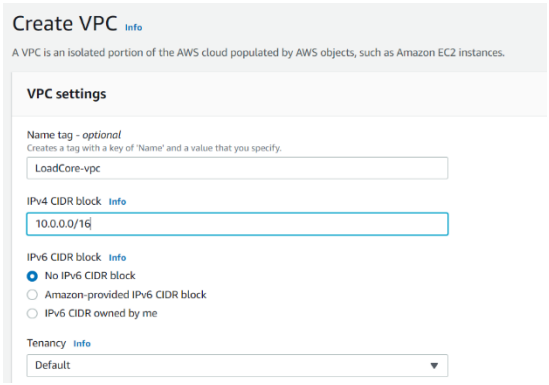
You can also use **Find Services** option and search for VPC:



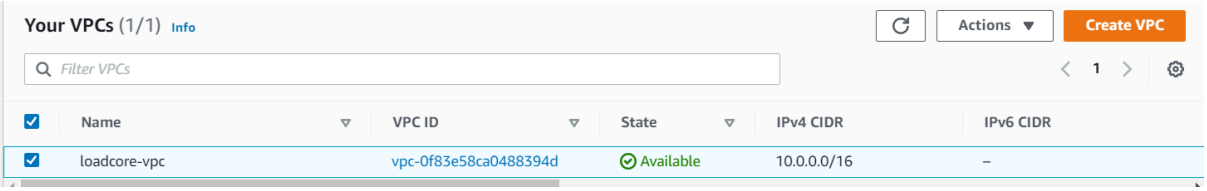
2. Select **Your VPC** and, then, select **Create a VPC**.



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:



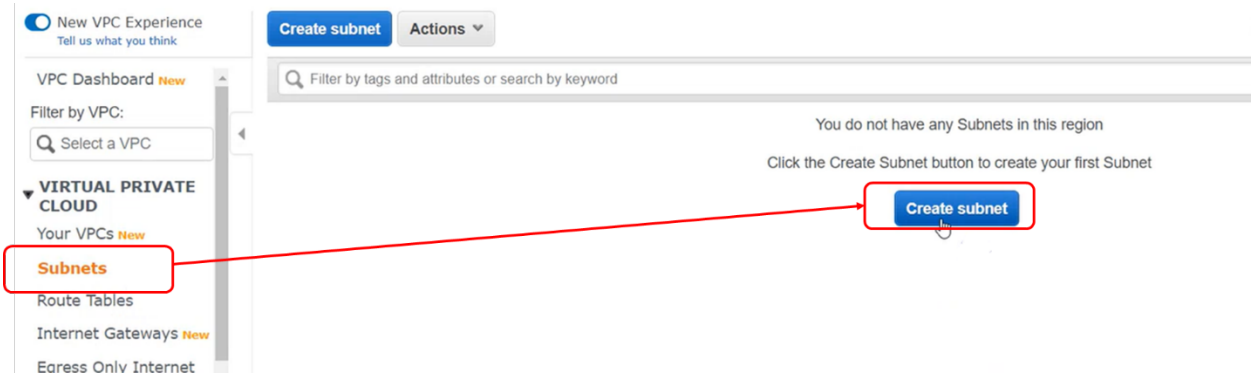
4. At the end of this operation you should have the following result:



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.

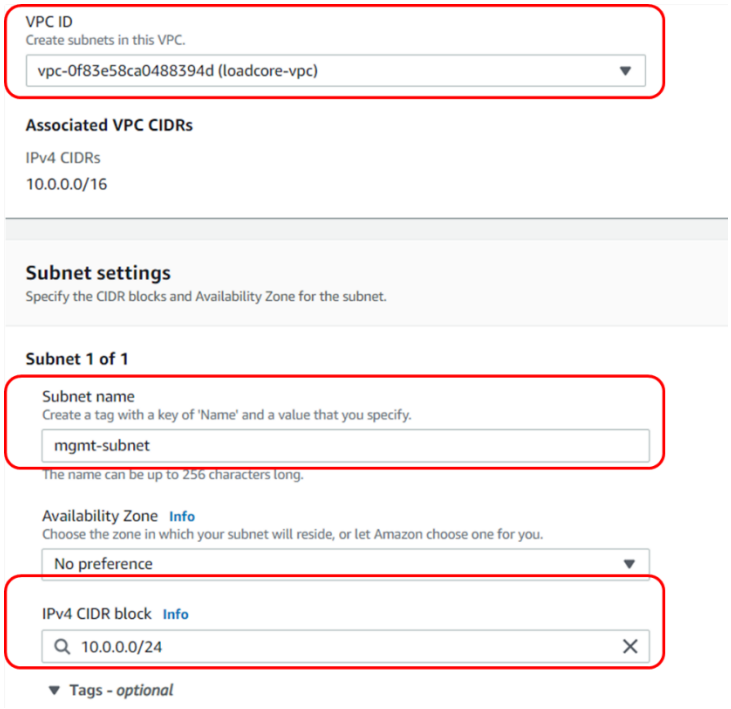
A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that will not be connected to the internet.



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):

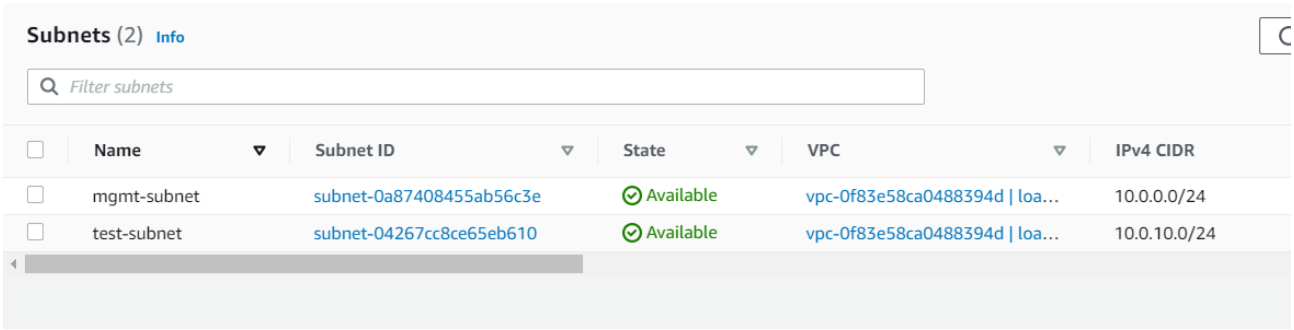
1. Select the **VPC ID** (there will be only one VPC – the one created at the previous step). If you are reusing an existing VPC, make sure to select the correct VPC ID.
2. Define the subnet TAG (his is the name assigned to the subnet).
3. Define the subnet IP addresses (in this case **10.0.0.0/24**).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



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.

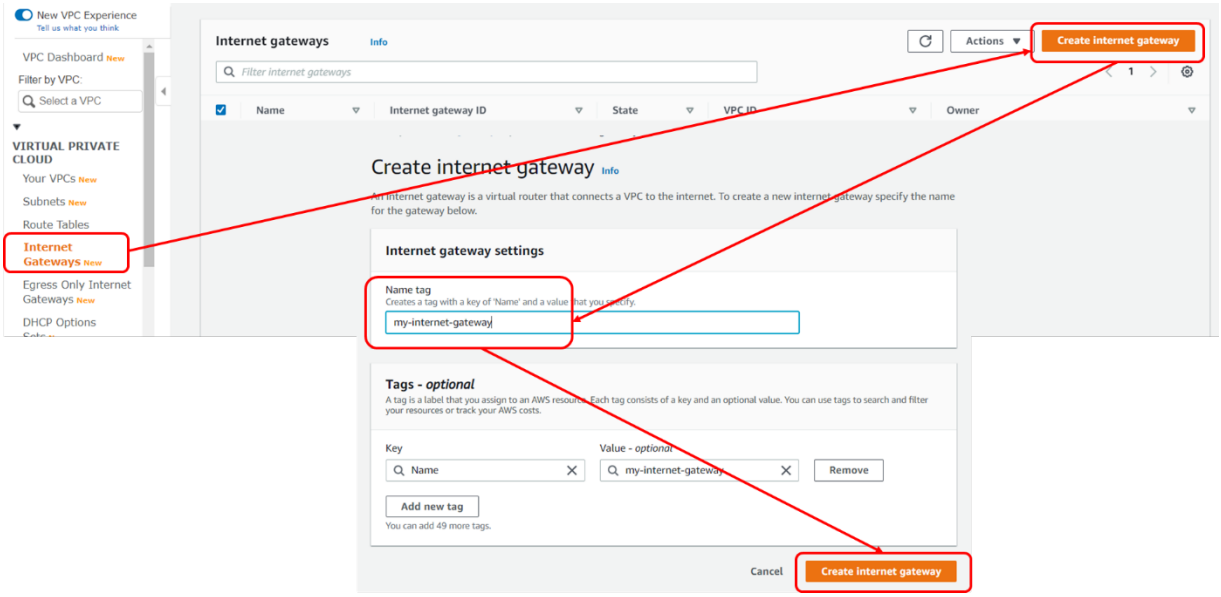


Create the Internet Gateway

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.

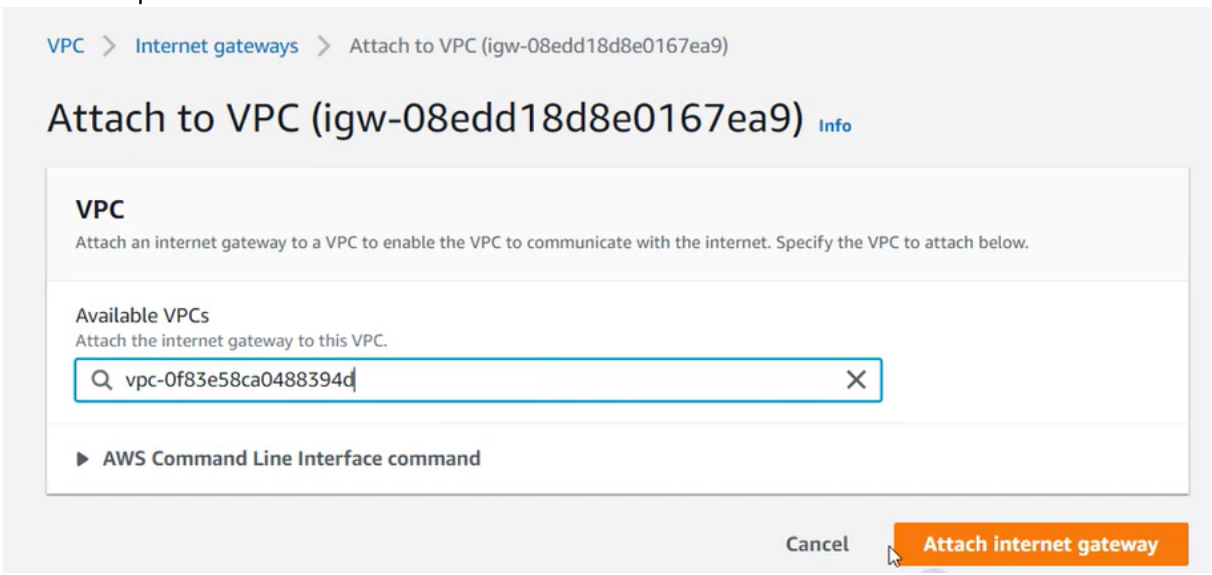
We will name it **my-internet-gateway**, after that, we need to attach it to our VPC.



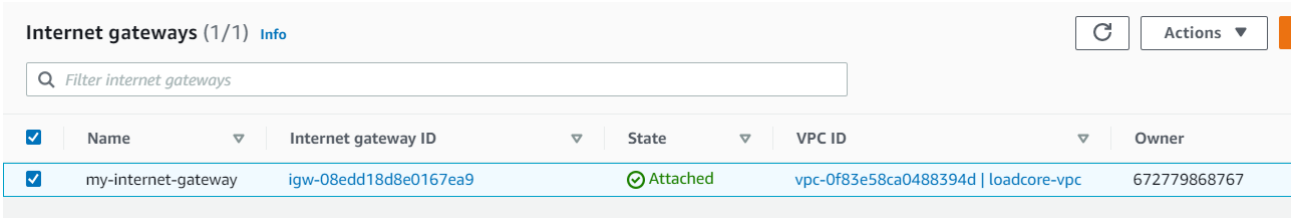
The newly created gateway is in the state **Detached**, with no VPC ID associated. Therefore, we need to attach it to our VPC:

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

3. Select the previous defined VPC.



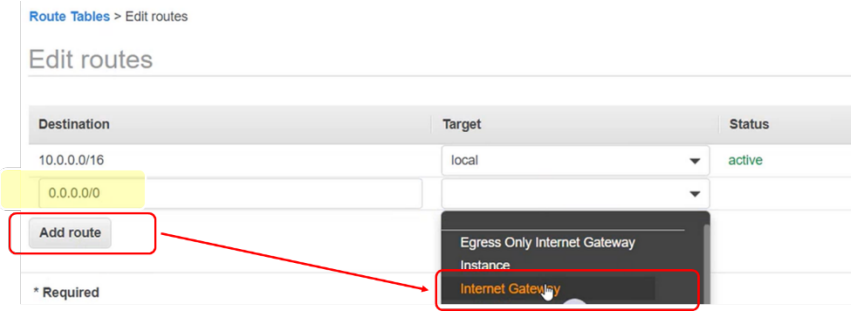
At the end of this step, the following image should be displayed:



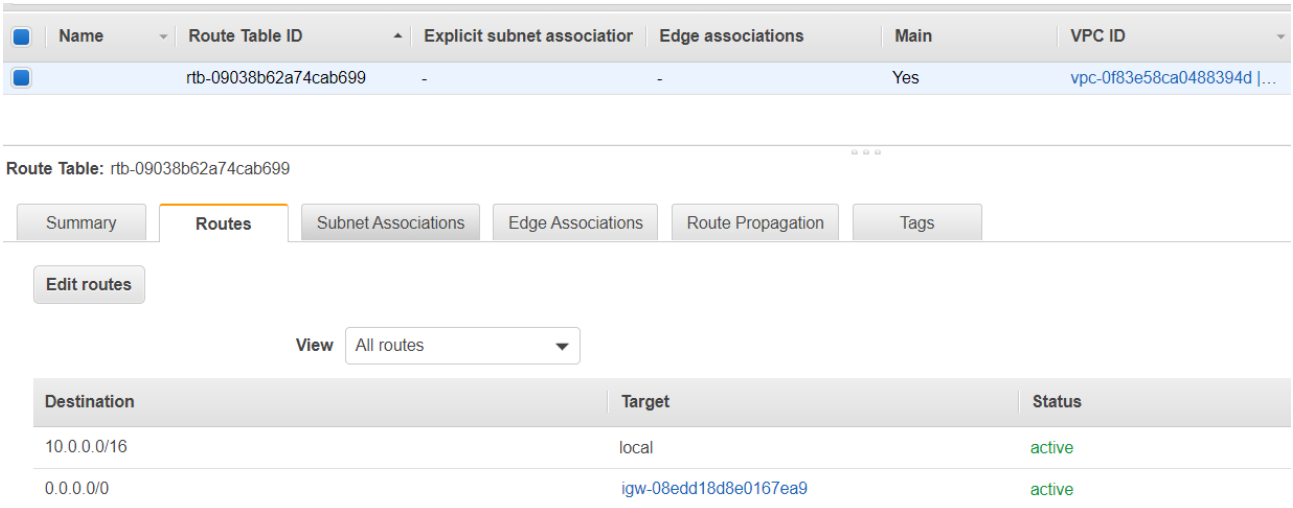
Assign traffic routes

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:

1. 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).
2. Select **Add Route** and define the default rate: **0.0.0.0/0**.
3. For **Target**, select **Internet Gateway** and select the gateway previously defined.
4. Select **Save Routes**.



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



Configure Security Groups

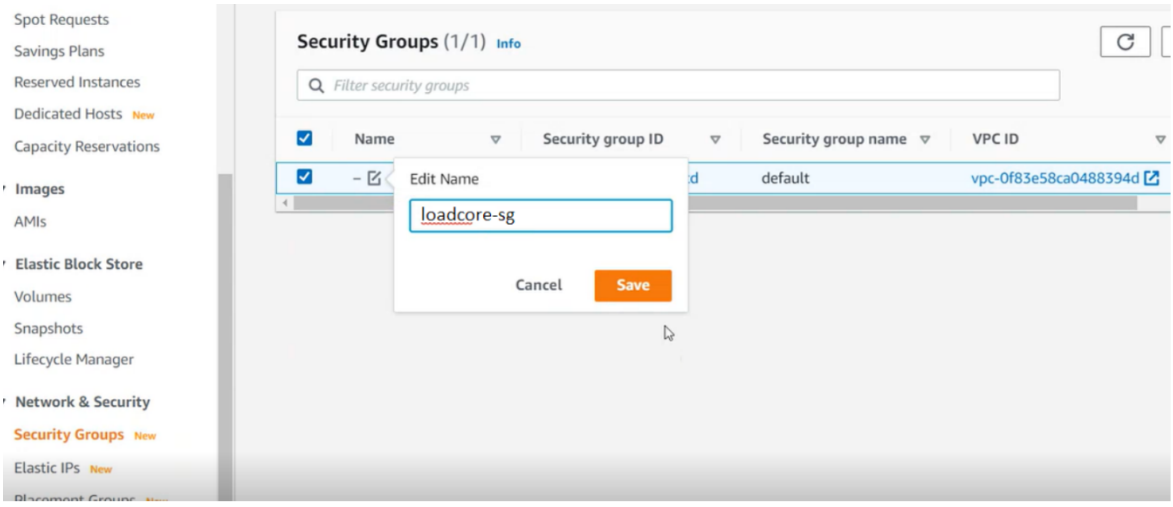
To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

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.

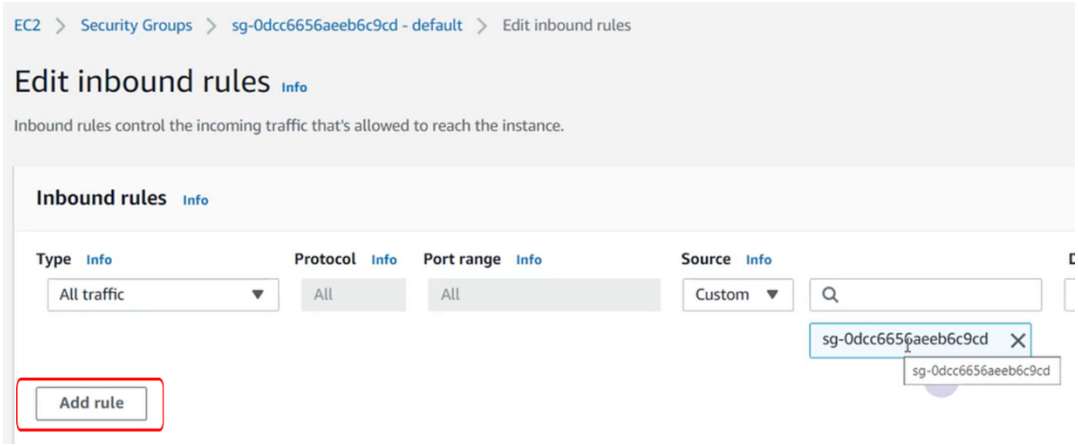
We will define a name for it (**loadcore-sg**) and configure the inbound rules:



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

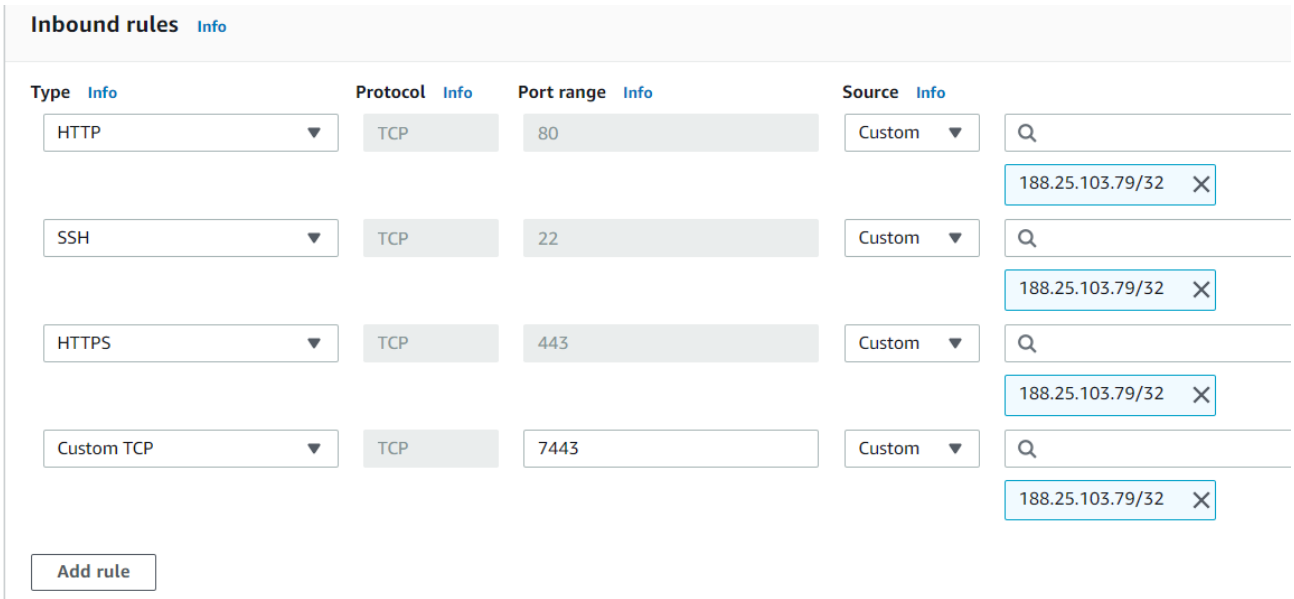
1. Select the security group ID and go on Inbound rules.
2. Select **Add rule**.

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



- We will permit the following:
  - SSH, to access the ORAN-SIM CE VMs,
  - HTTP, HTTPS, used by ORAN-SIM CE for configuration, interaction between nodes,
  - Custom: 7443 port, that is used by the 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**).

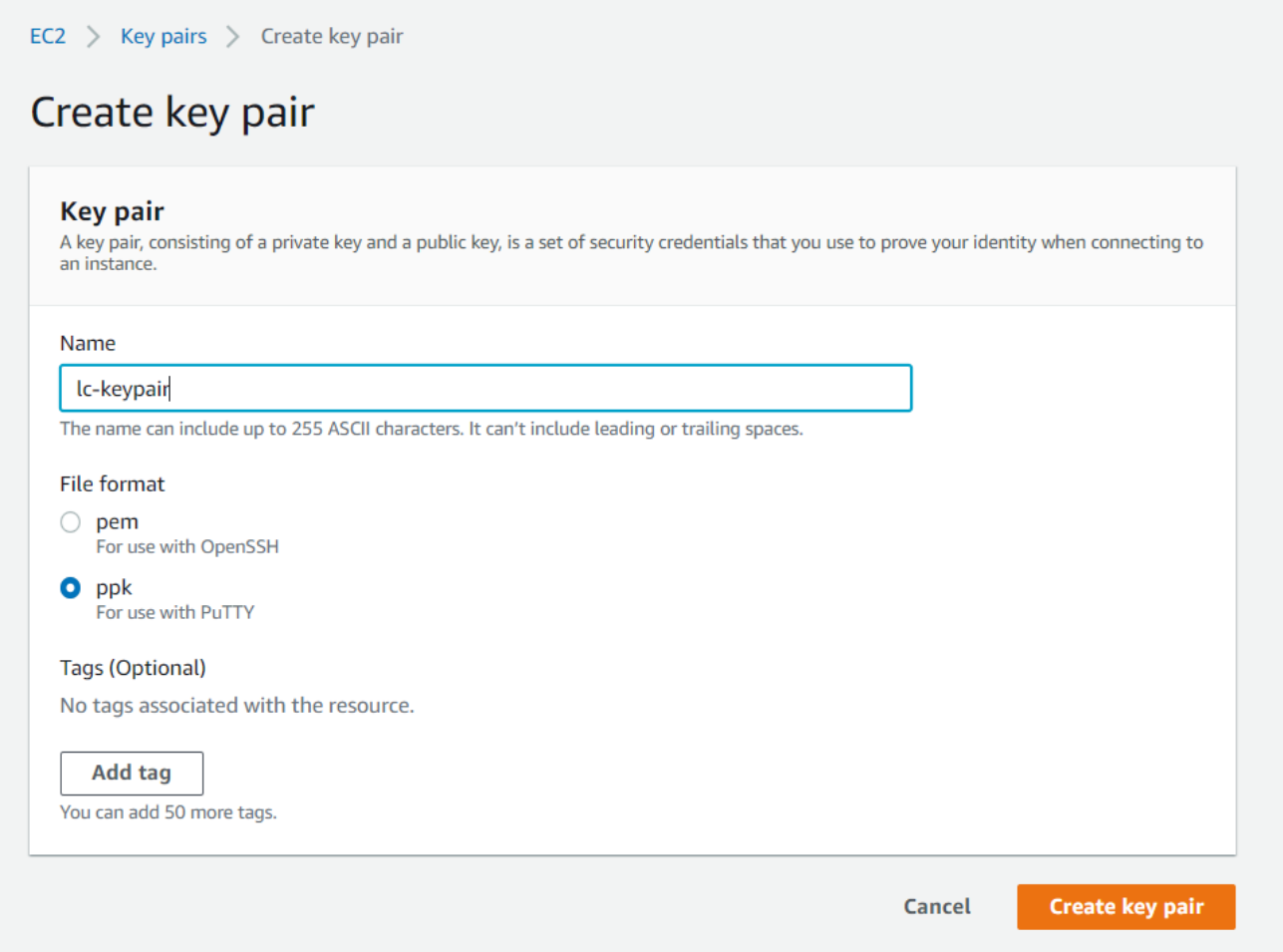
At the end of this step, you should have the following rules in your Security Group:



NOTE If deploying instances in different VPCs, you will have to open other ports in the security groups as well; for example, 4222 for communication between MW and agents, and 7443 for communication between MW and License Server.

Create Key Pairs

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



LoadCore Components Installation

Until now, we have created the minimum infrastructure required to deploy the ORAN-SIM CE components.

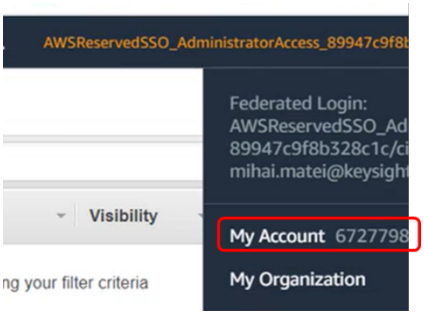
In AWS, the images that we are using are in AMI format.

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.

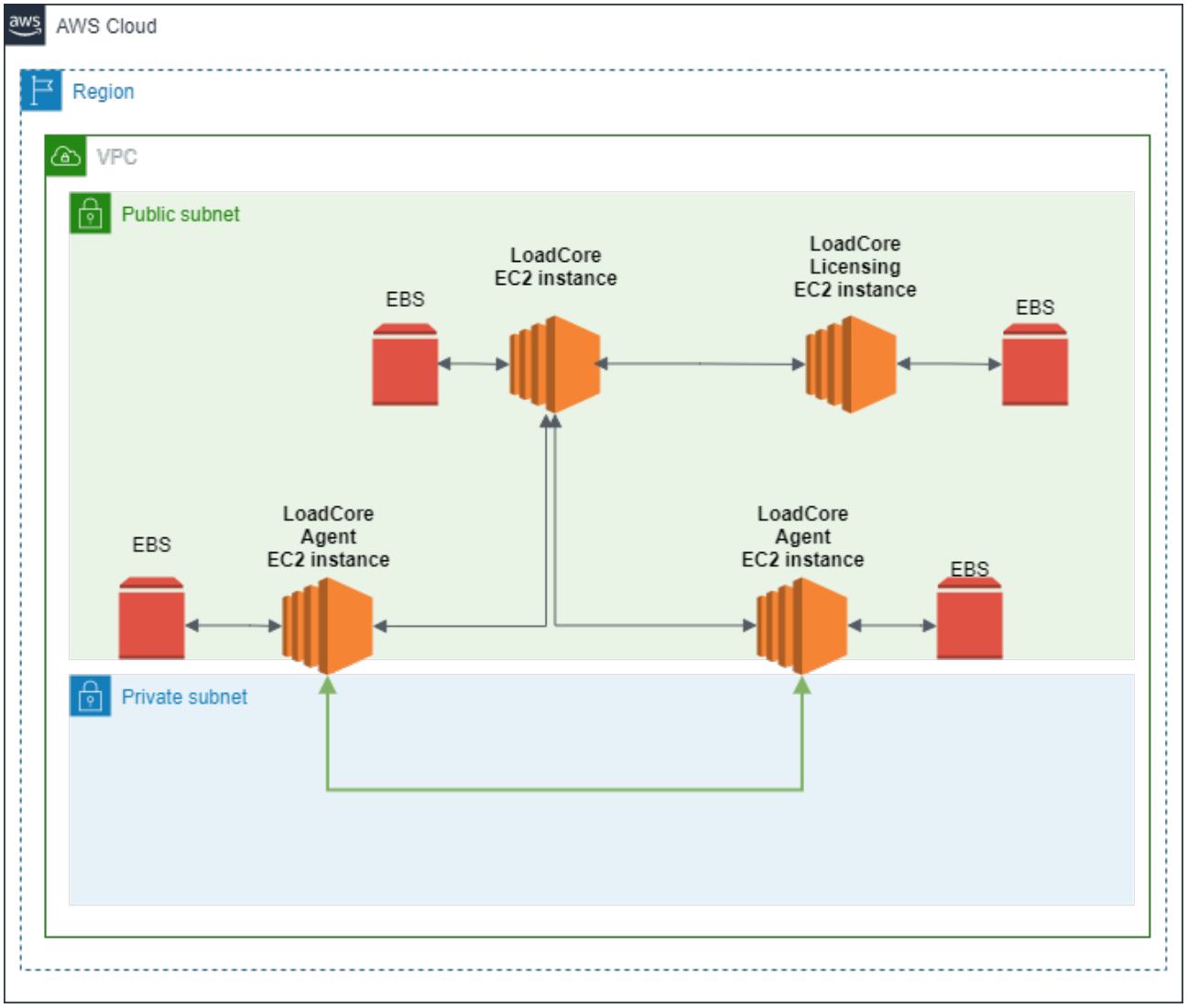
To get access to them, please contact our Support or PM team. You need to provide your AWS account ID to get access to the shared images.



In order to install the ORAN-SIM CE 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 ORAN-SIM CE deployment:



Middleware Installation

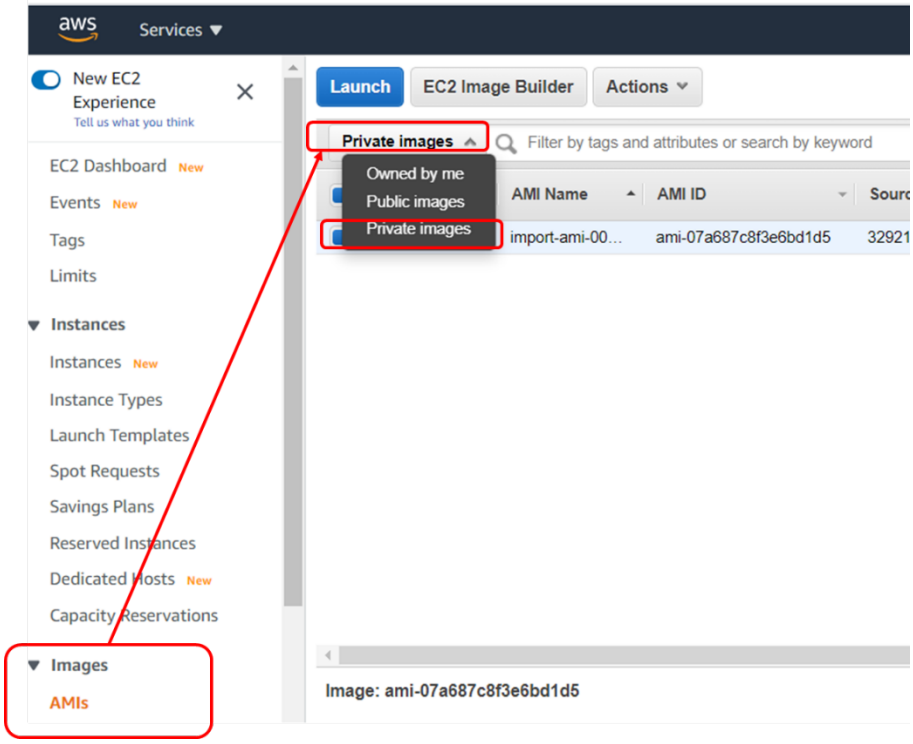
The installation procedure of the Middleware requires the following steps:

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

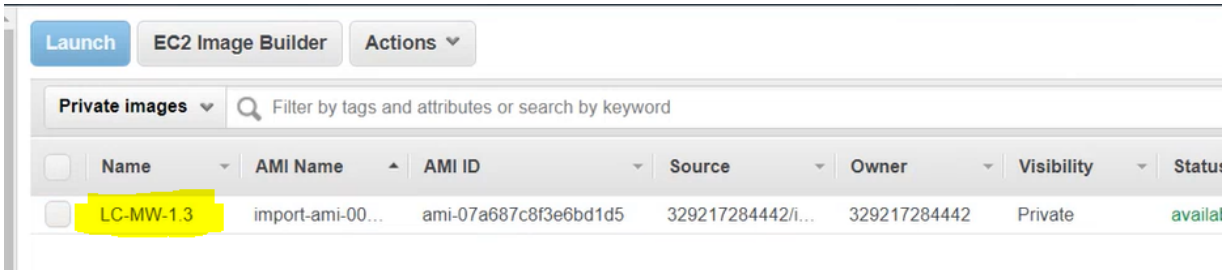
   The browser will automatically display the **images that you own**, therefore, if you do not have other images you created, the list will be empty.

   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



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

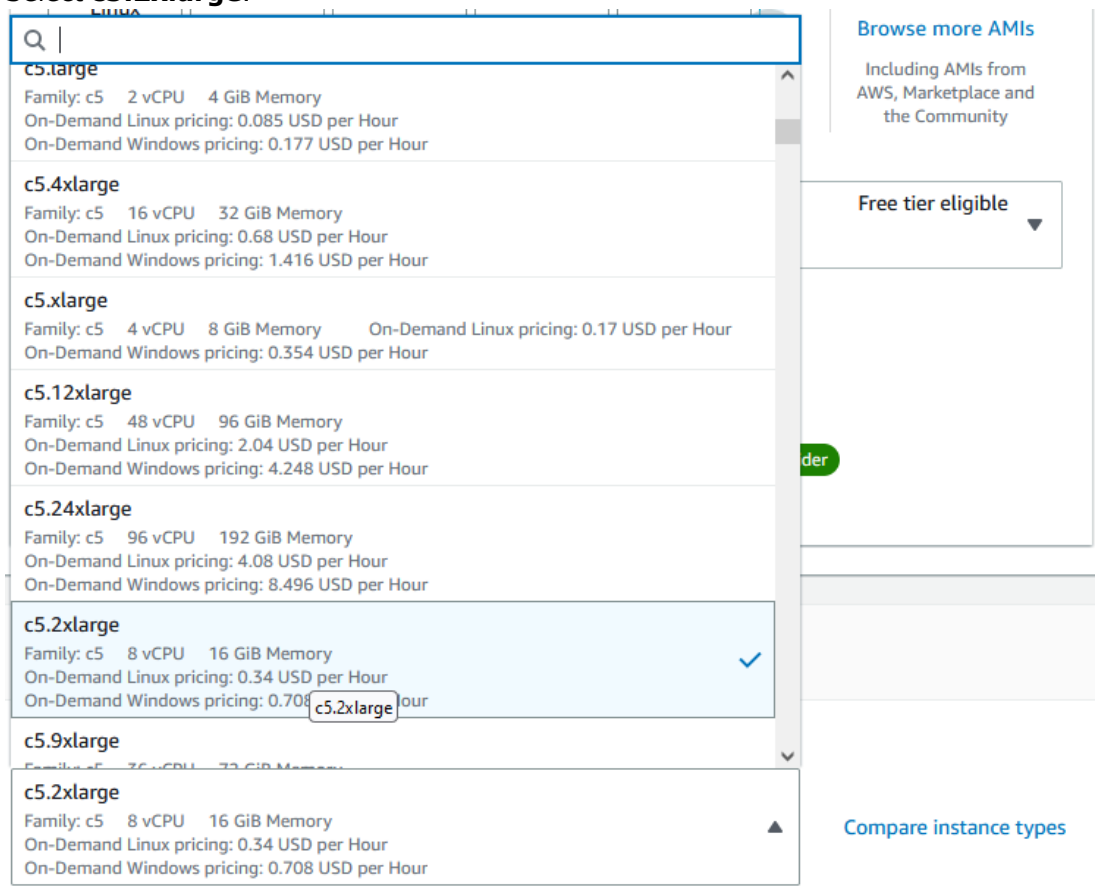


   Once you have the image available, select it and select **Launch**.

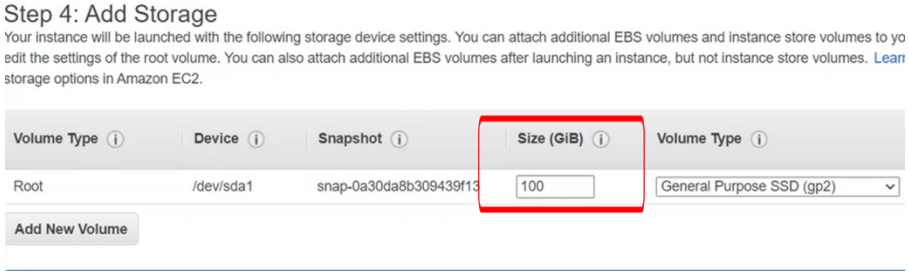
   The launch process will request additional info, which is meant to adjust the resources in AWS.

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

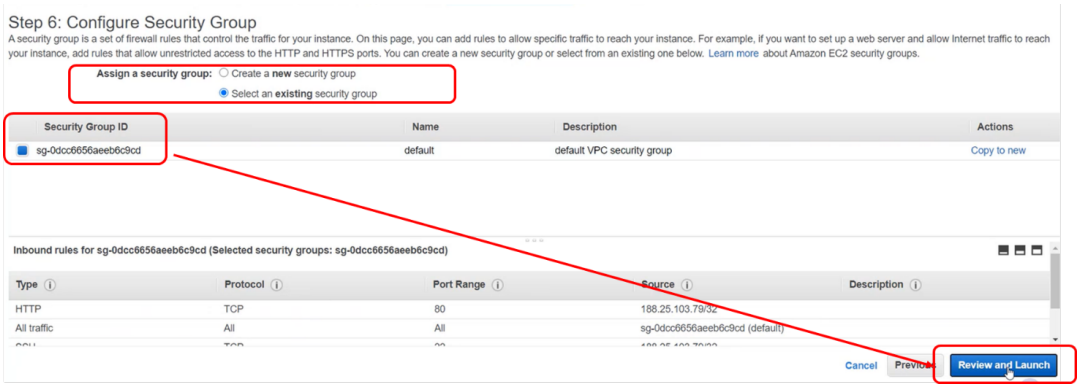
- Select the Instance type:
  - Select **c5.2xlarge**.



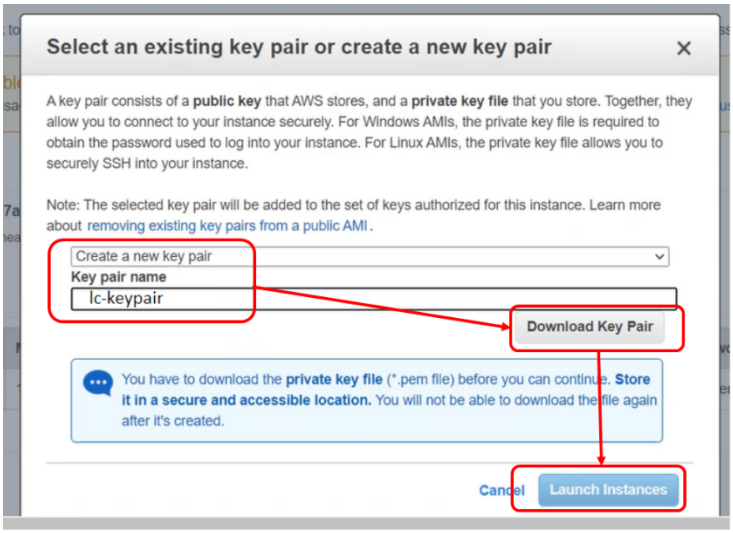
- Configure the Instance Details:
  - VPC – select the VPC we defined. In our case there is only one, but if you have multiple VPCs, you need to select the one you want to use to deploy ORAN-SIM CE.
  - Select the management subnet.
- Select **Add Storage**.
  By default, the **Size** will come with 100 GB.
  For long duration tests, you need to increase the size to 256 GB.



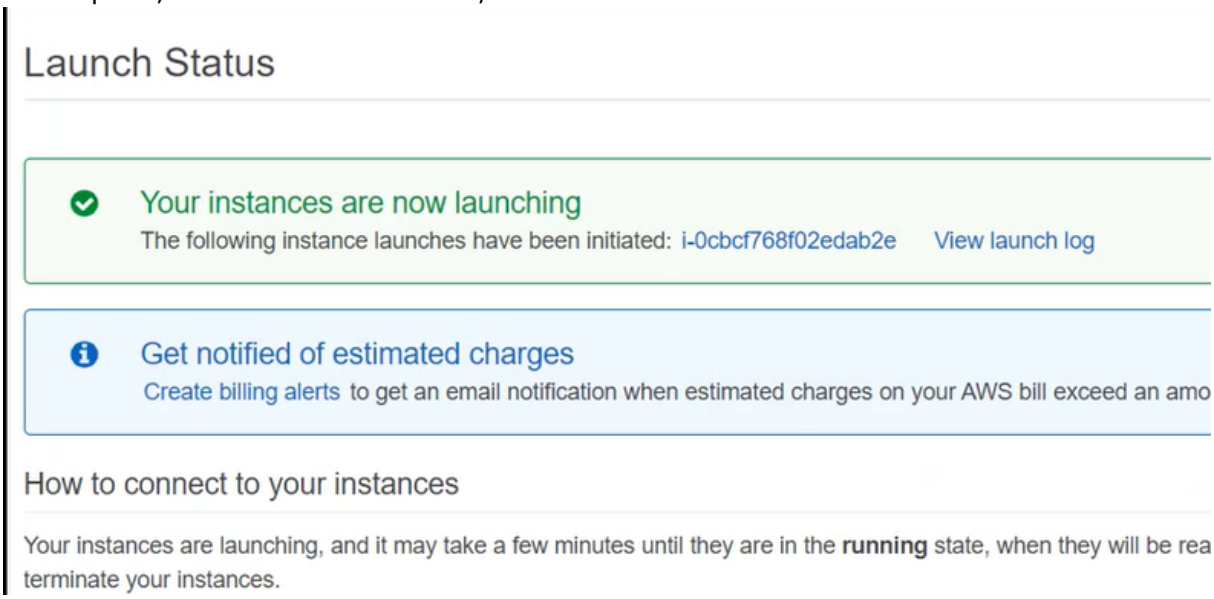
- There is no need to define Tags, select **Configure Security Group**.
  - Select the security group we previously defined.
  - Select **Review and Launch**.



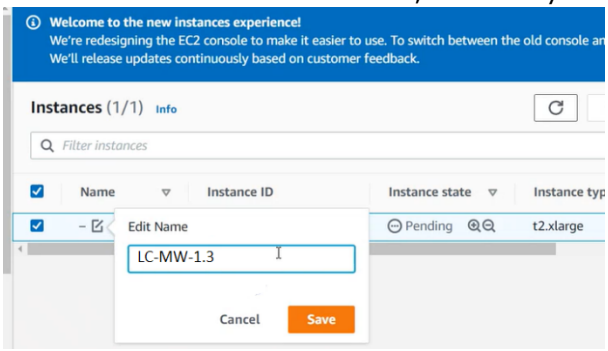
2. Select **Launch**.
   This will display a pop-up menu from where you can select/create a key pair. If you already have a key pair (and you downloaded it), you can select that one, otherwise you can create a new one, and download it from here, as shown below:
   - Select **Create a new key pair**.
   - Define a name for it: **lc-keypair**.
   - Select **Download Key Pair**.
   - Select **Launch Instances**.



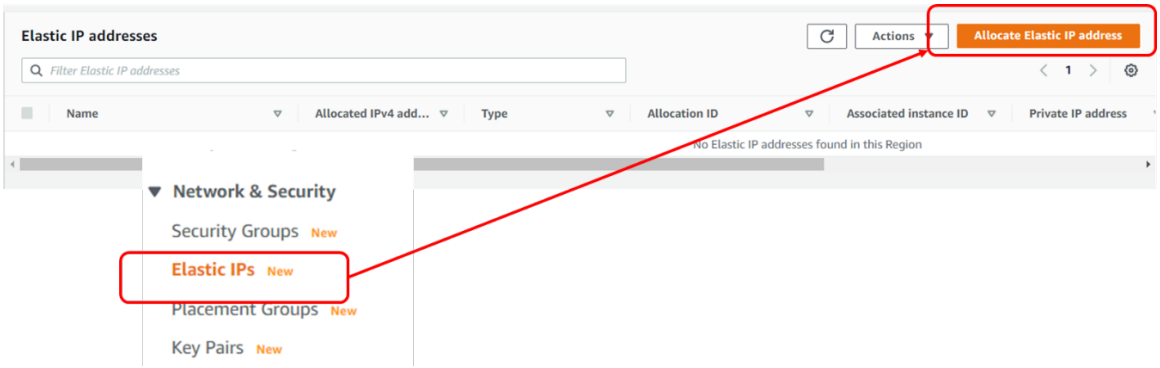
3. At this point, the instance is launched, as shown below.

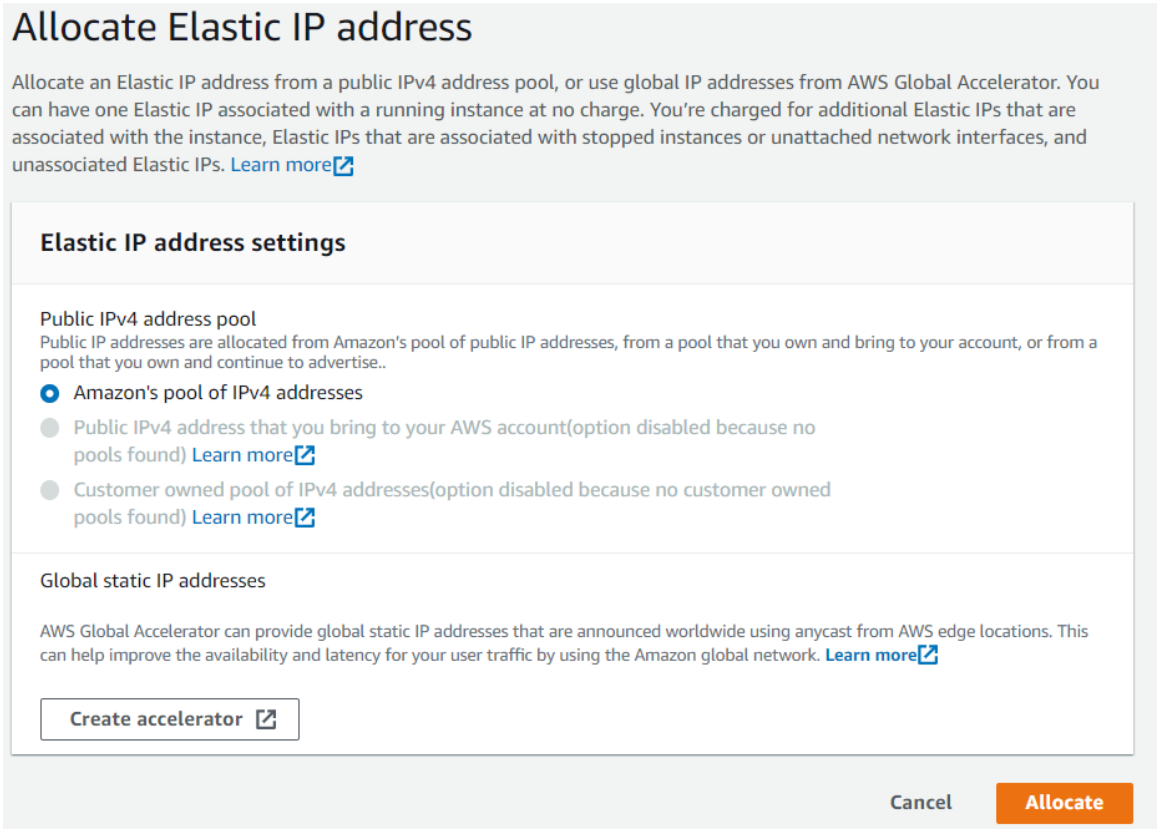


You can also define a name for it, to be easy to identify it (in this case **LC-MW-1.3**).

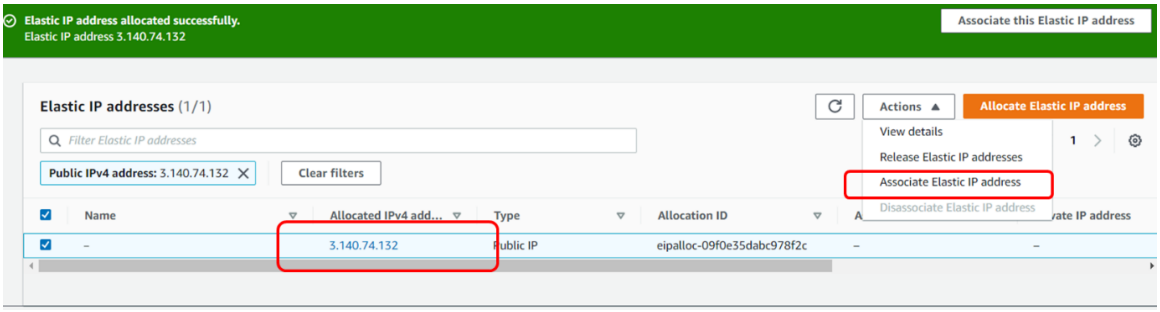


4. To access the Middleware UI, we will need to assign an Elastic IP.
   To do this, select the Elastic IP menu:
   - Select **Allocate Elastic IP Address**.

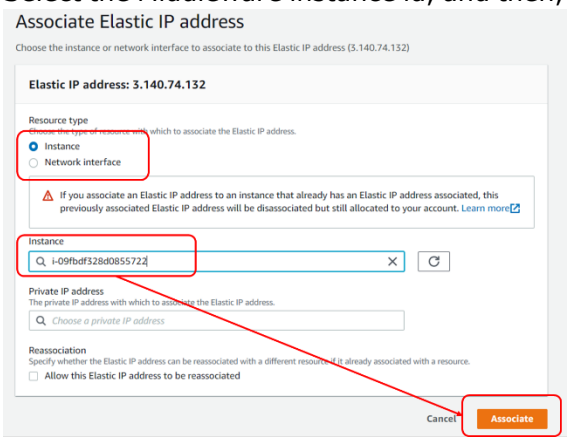




- The new IPv4 Public IP address will be allocated from AWS pool.
- Associate this IP address with the Middleware instance.



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



IMPORTANT This is the IP address of the web 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 willuse 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

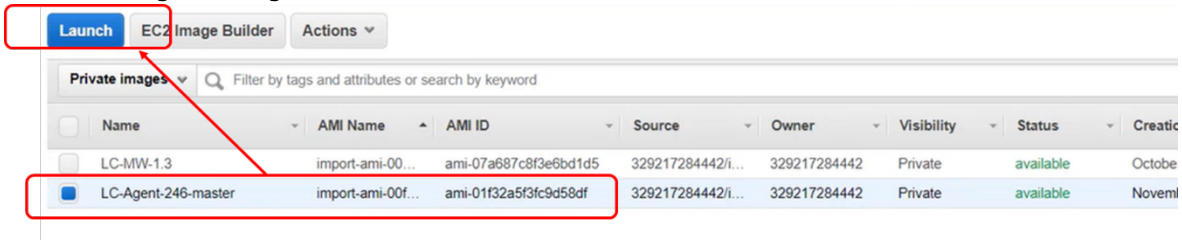
The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

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:

1. 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**.

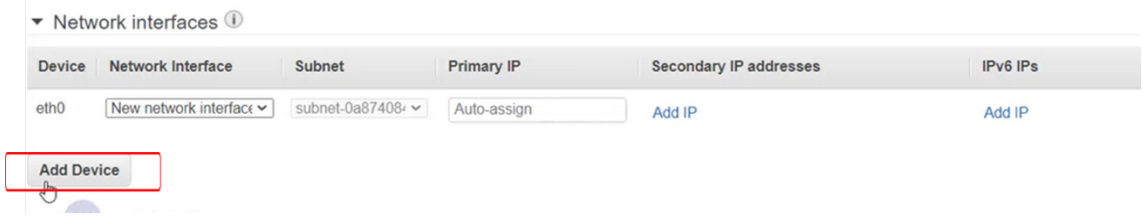


   Select **t2.xlarge** for instance type:

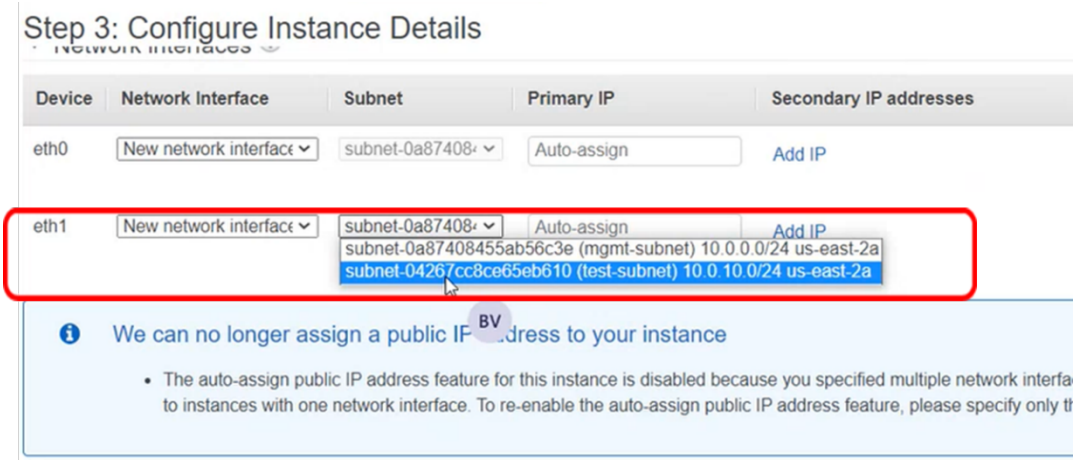
   Select **Next**.

2. 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**.

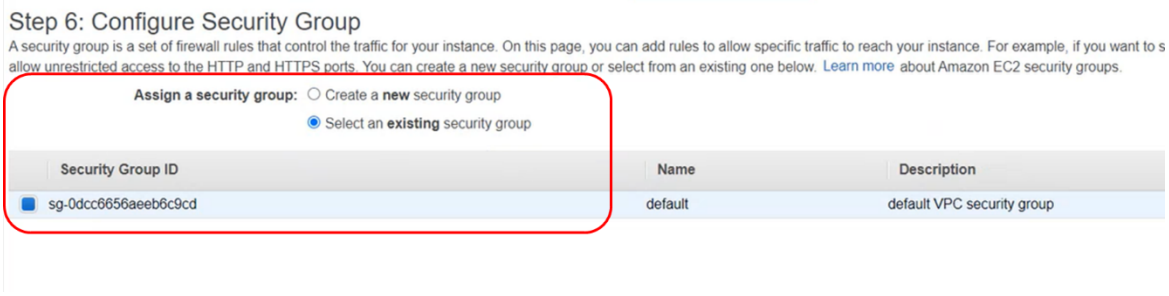


- Assign it to the test subnet (previously defined).



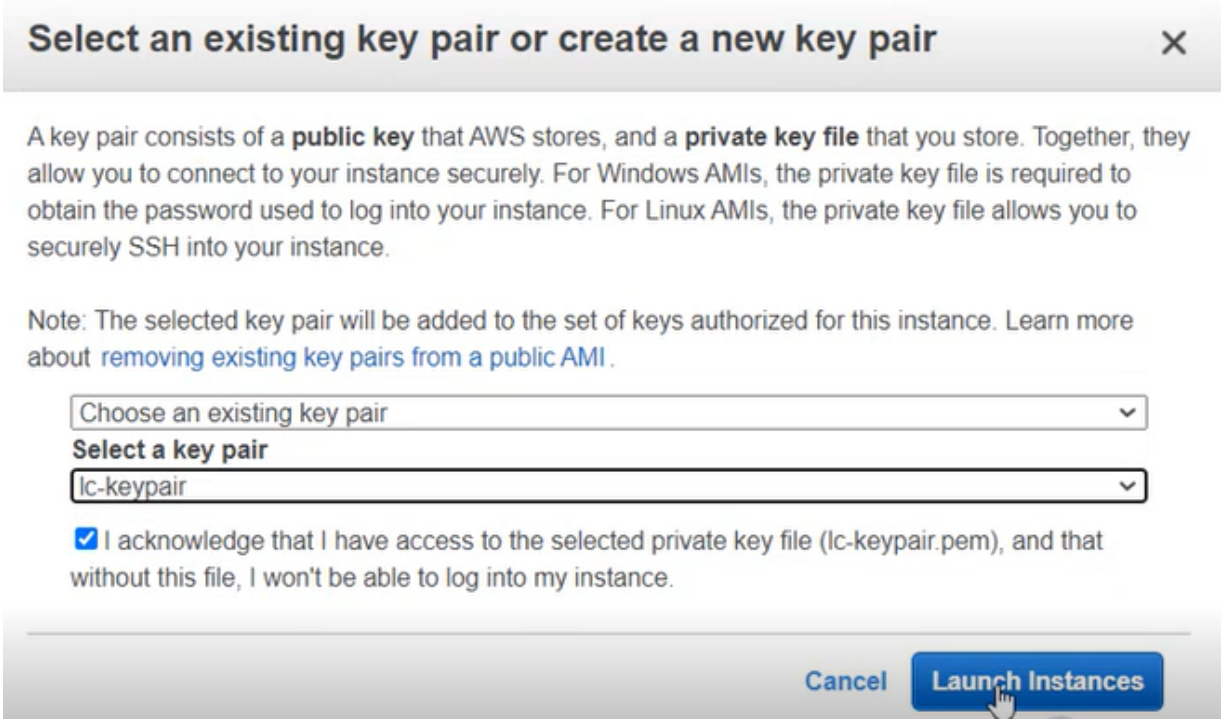
Select **Next**.

3. Add Storage.
   By default, the image comes with 17GB assigned. This can be increased is 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.

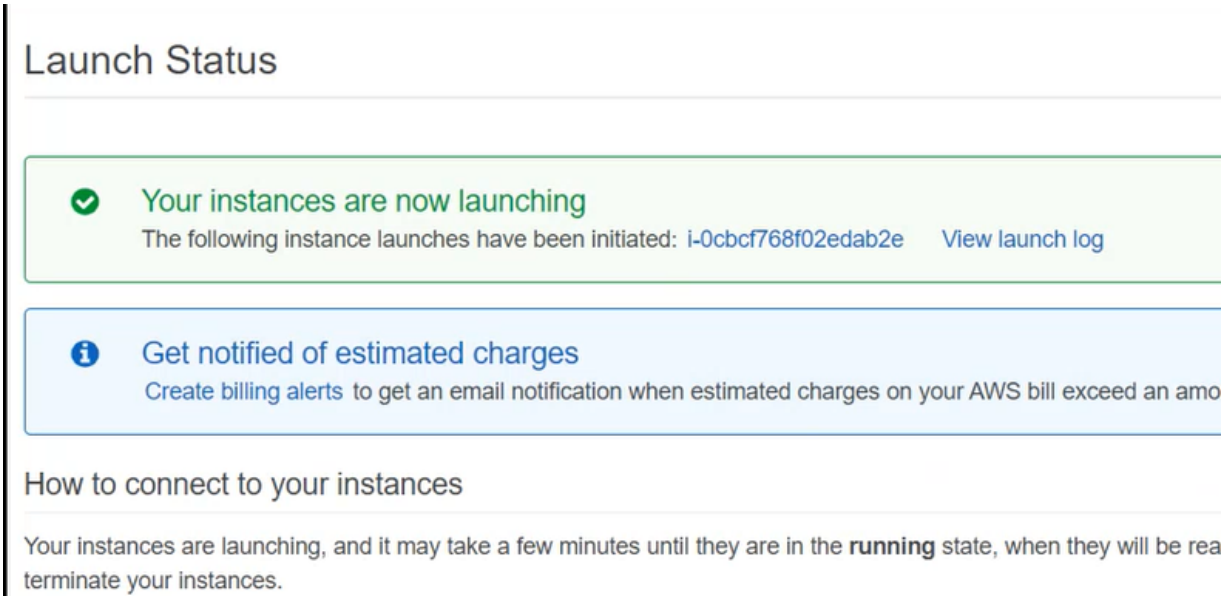


6. Select **Review** and **Launch** > **Launch**.

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

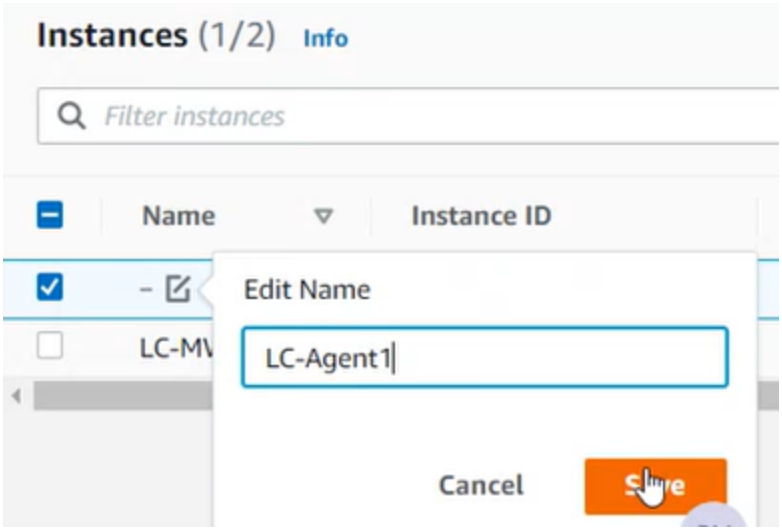


8. Select **Launch Instances**.



9. On the available instances, you should see two running instances:
   - The one containing the Middleware.
   - The new one, containing the Agent.

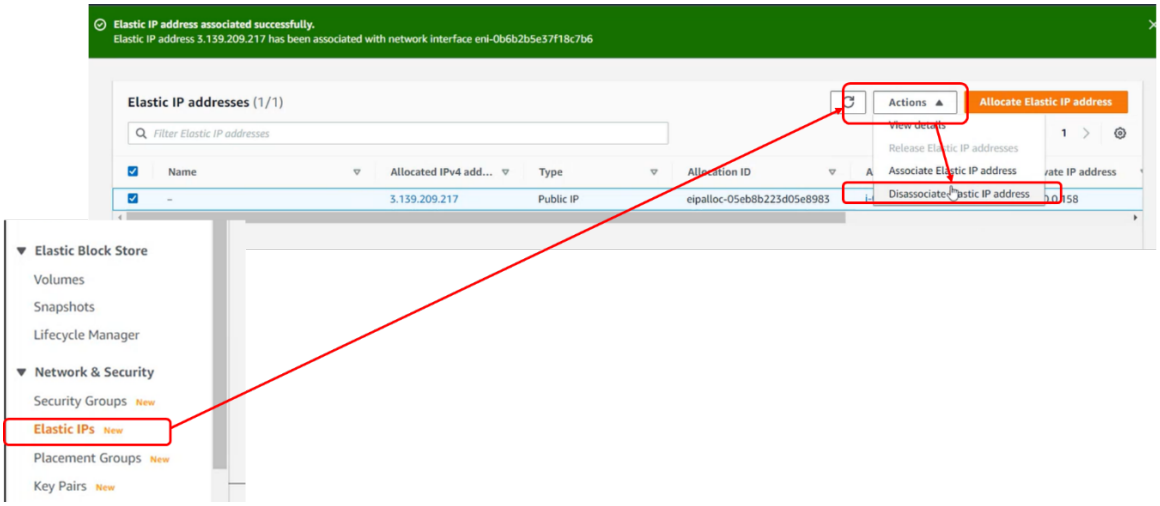
   Define a name for the test agent instance (by default, there is no name).
   In this example we will name it **LC-Agent1**.

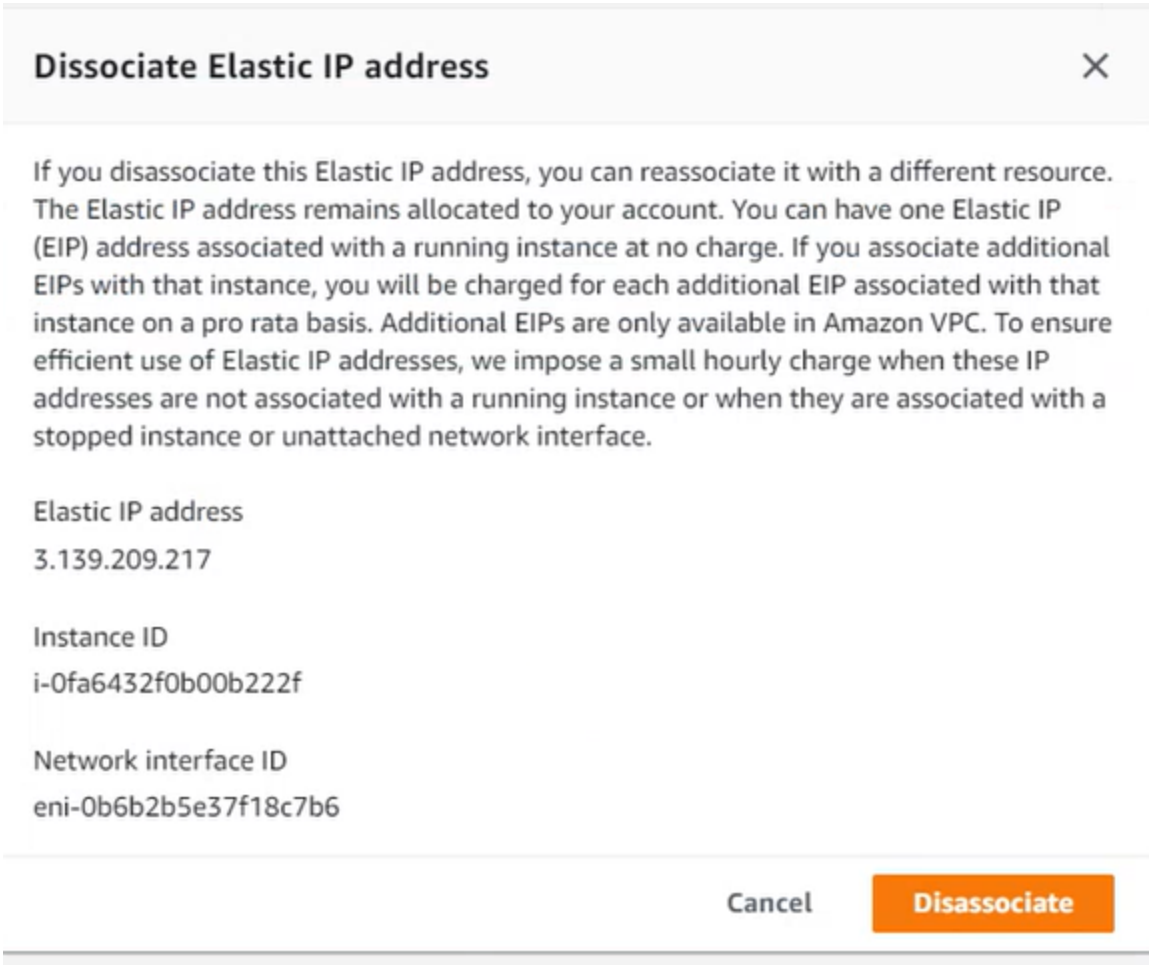


10. In order to access the agent, we need to assign an elastic IP.

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

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





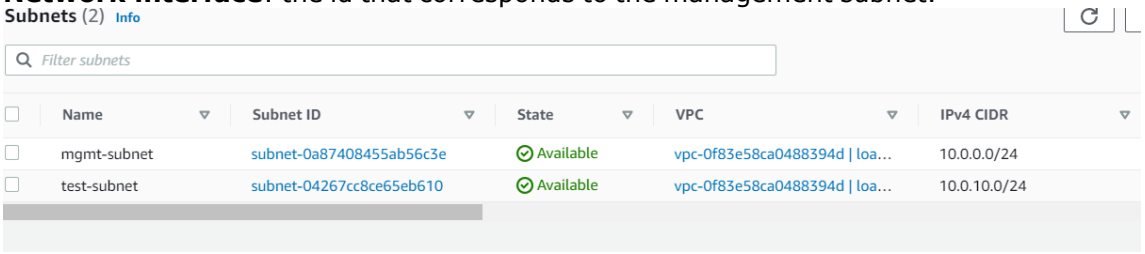
- The Elastic IP was disassociated and now should be reassociation to the Management interface of the test agent.

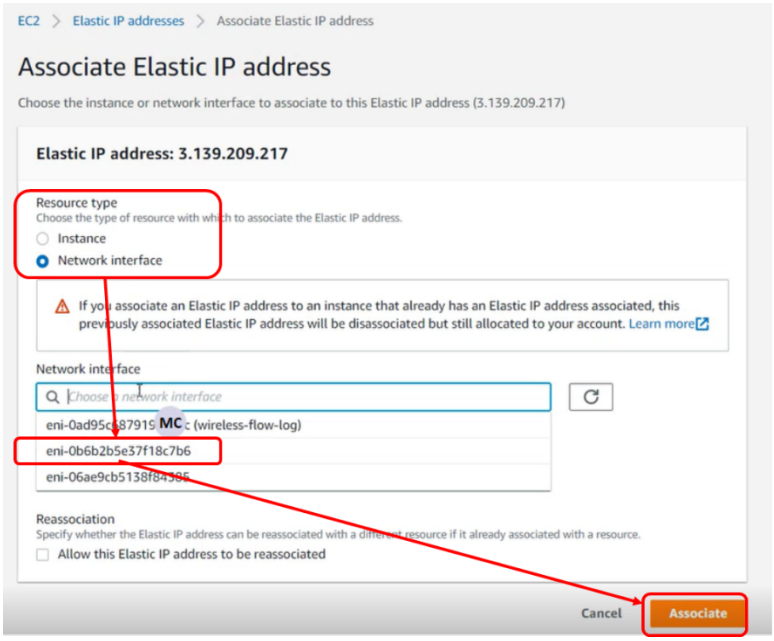
In the previous image deployment we associated the IP address to the Middleware instance, but here, since we have two network interfaces, we need to associate the Elastic IP address to the management network interface.

Select:

- **Resource type**: **network interface**.

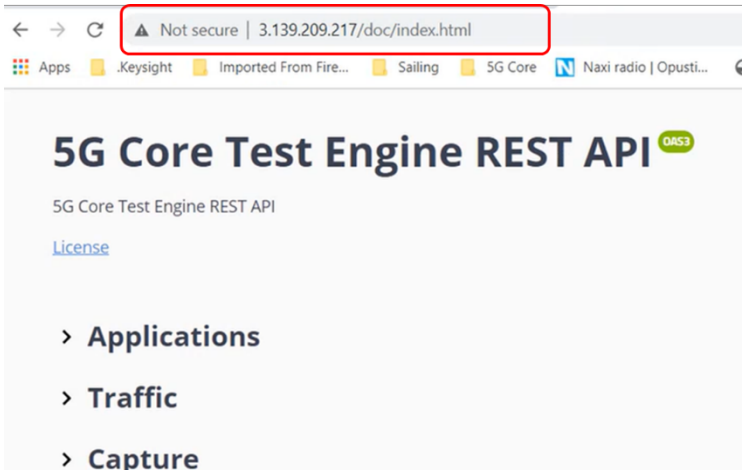
  Network interface: the id that corresponds to the management subnet.

- 



- **Associate**.

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



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
      ```

- 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, ORAN-SIM CE 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 Agent instances.

IMPORTANT Please make sure you enable the SRIoV option on each simulated 5G node in ORAN-SIM CE UI **Agent Assignment** > **Network Manangement** window.

C4 family will require a special parameter to be configured on 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

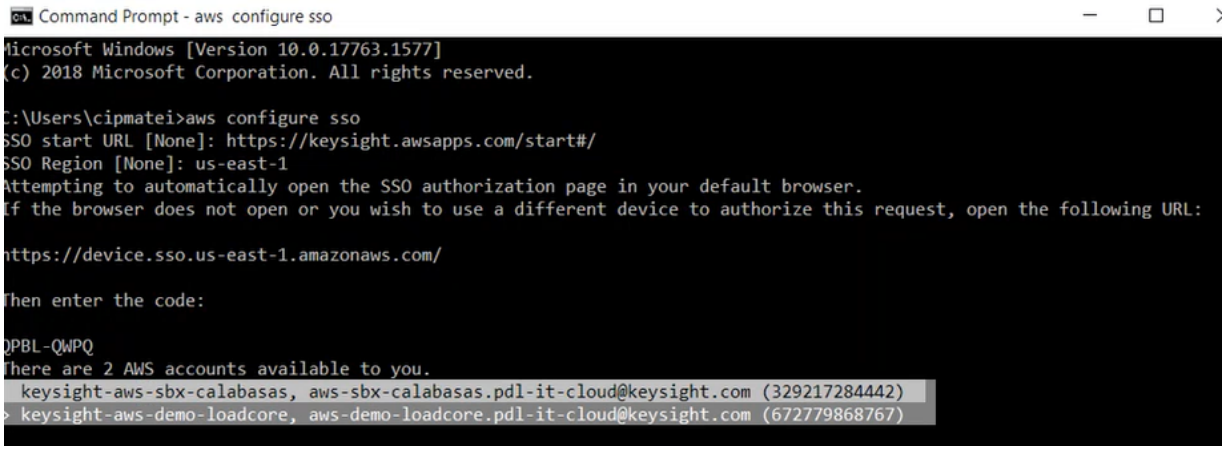
In order to use SSO to connect AWS CLI to your account, AWS SSO needs to be enabled on your account. For more details, refer to:
https://docs.aws.amazon.com/singlesignon/latest/userguide/step1.html

Open a CLI terminal and configure SSO, as follows:

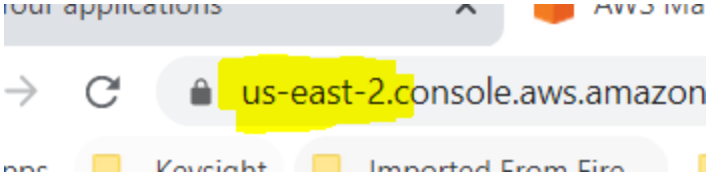
1. **AWS configure SSO**
2. SSO start URL: <**your SSO link**> (e.g. **https://keysight.awsapps.com/start**)
   You need to replace <**your SSO link**> with the SSO link used by your organization to connect to AWS.
3. SSO Region: <**SSO instance region**> (e.g. **us-east-1**)
   You need to check the 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:

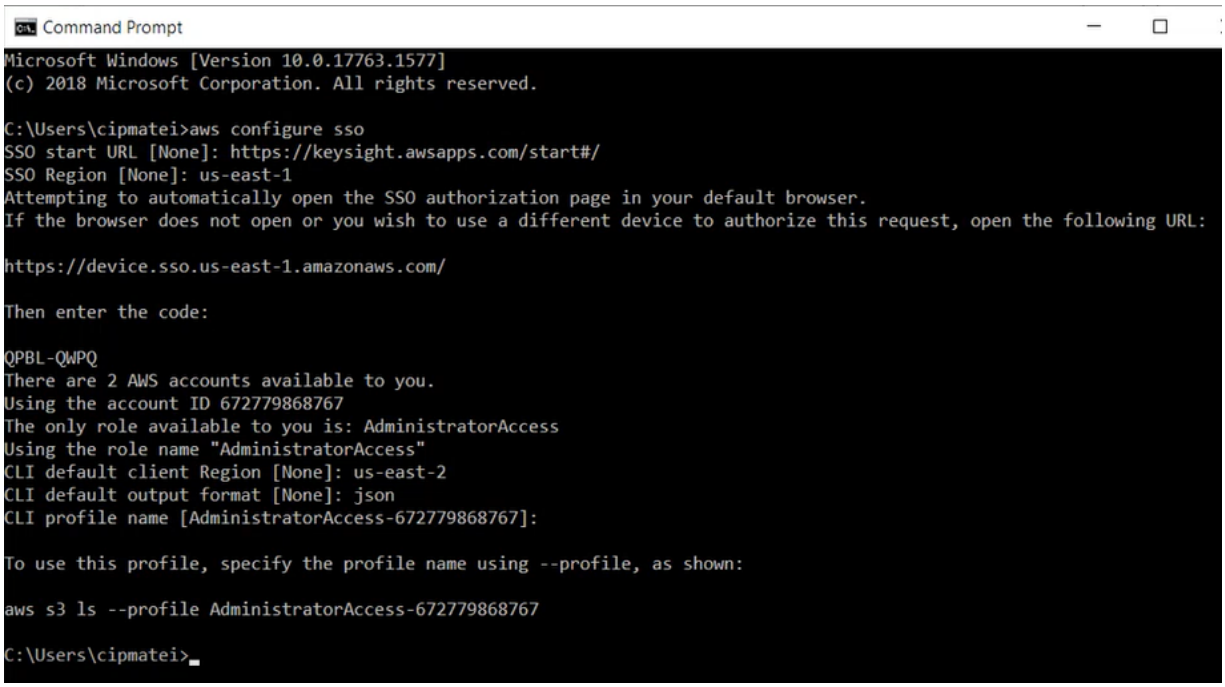
5. Select **Sign to AWS CLI**.
6. On the terminal CLI you will see the available accounts. Select the one you are using:



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**.



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

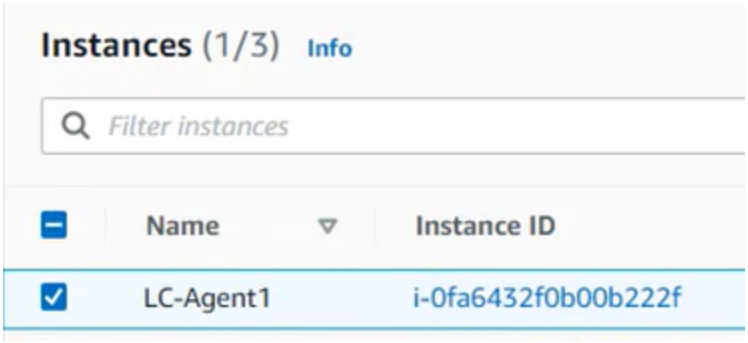
Configure SRIOV support

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

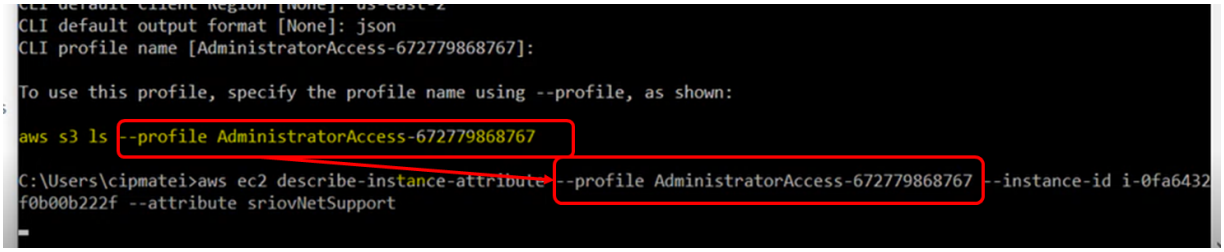
- Check if our test client instance has SRIOV support:
  On CLI terminal type:

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

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



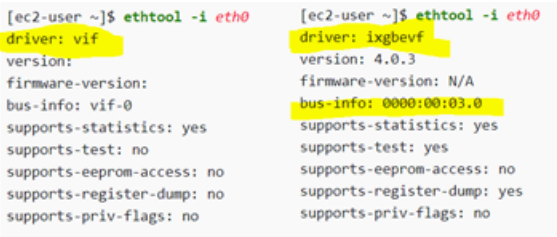
  If you have multiple accounts, you'll need to specify the profile id in the above command. If you have only one account this can be skipped:



  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).

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).



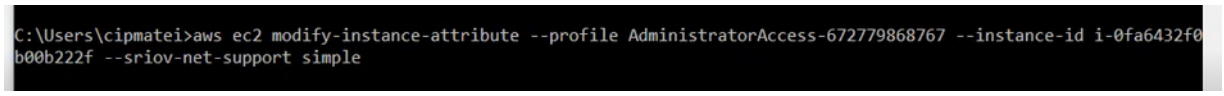
- We will now modify the instance attribute to support SRIOV.

  Stop the test agent instance.

  In CLI type the below command:

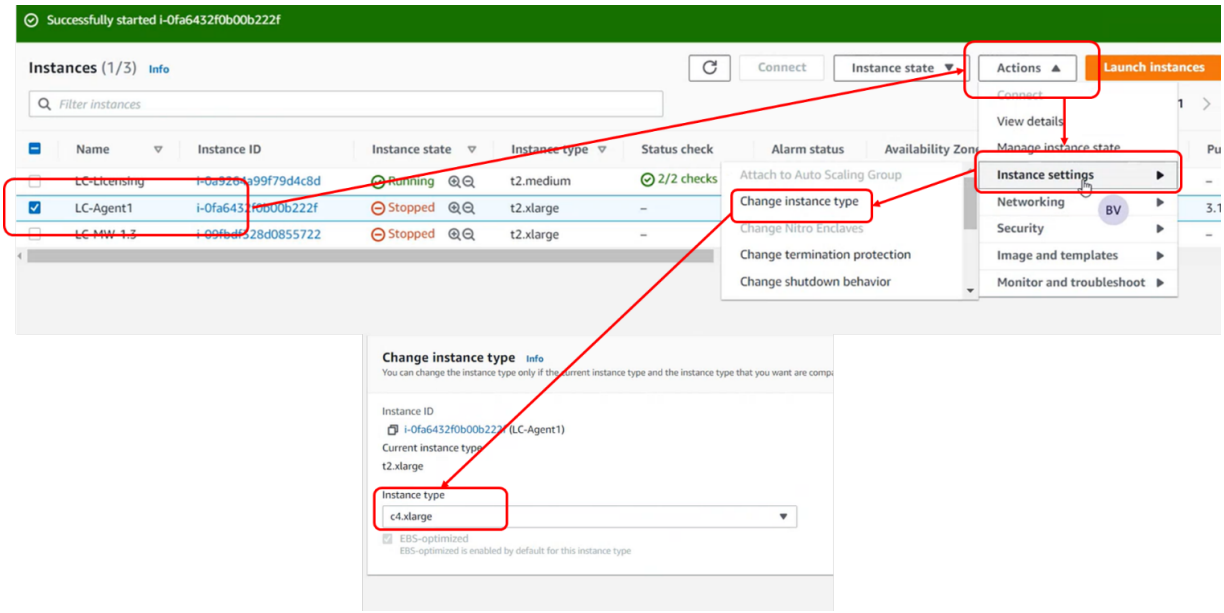
  ```
  aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support
  simple
  ```

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



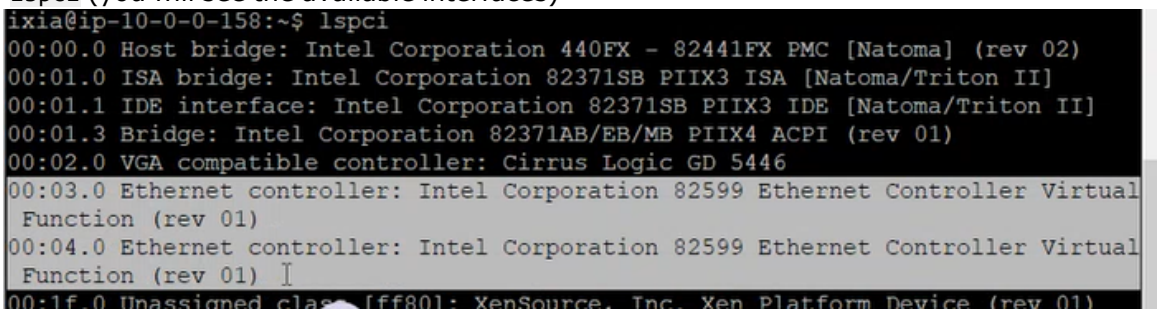
- Make sure you have an Elastic IP associated with the test agent.
- Stop the test agent instance and change the instance type.

- Select **c4.xlarge** type.

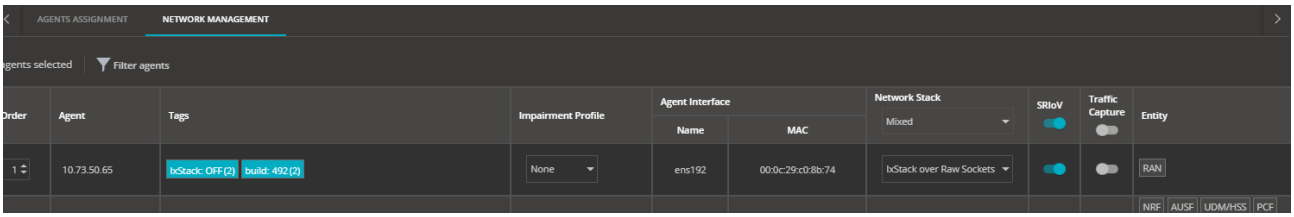


- Start the instance.
  Now the test agent instance is properly configured, has SRIOV support.
  You can check this using SSH on test agent and type:
  - `ethtool -I eth0` (you should see the `ixgbevf driver`)

  - `lspci` (you will see the available interfaces)



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.



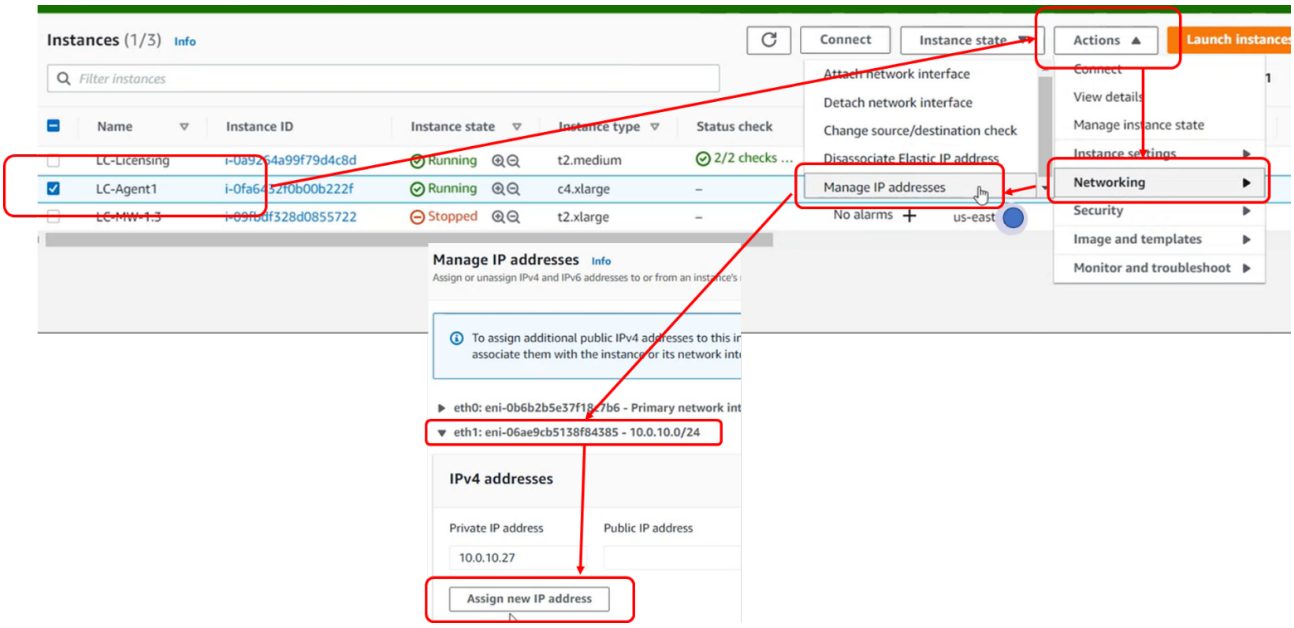
Now the agent is able to run application traffic on `eth1`.

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):

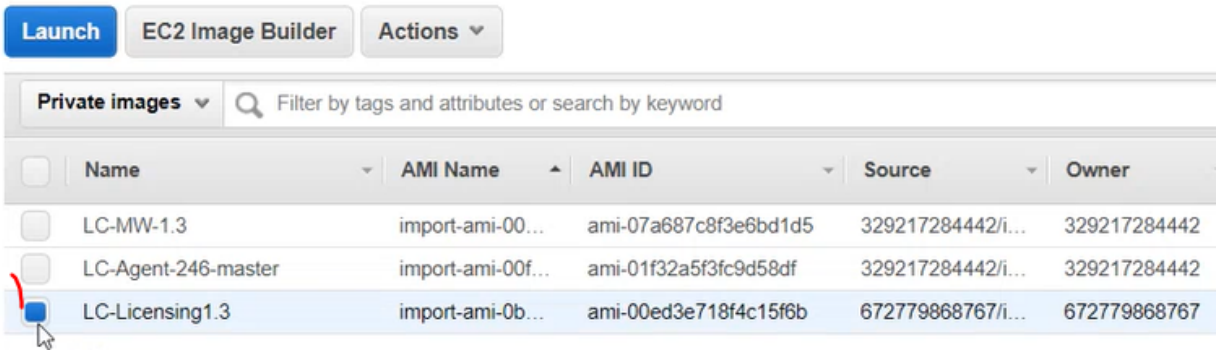
1. Go to **Instance** > **Networking** > **Manage IP addresses** > select **eth1** > **Assign new IP addresses**.
2. Add all the IP addresses that are used in the test.



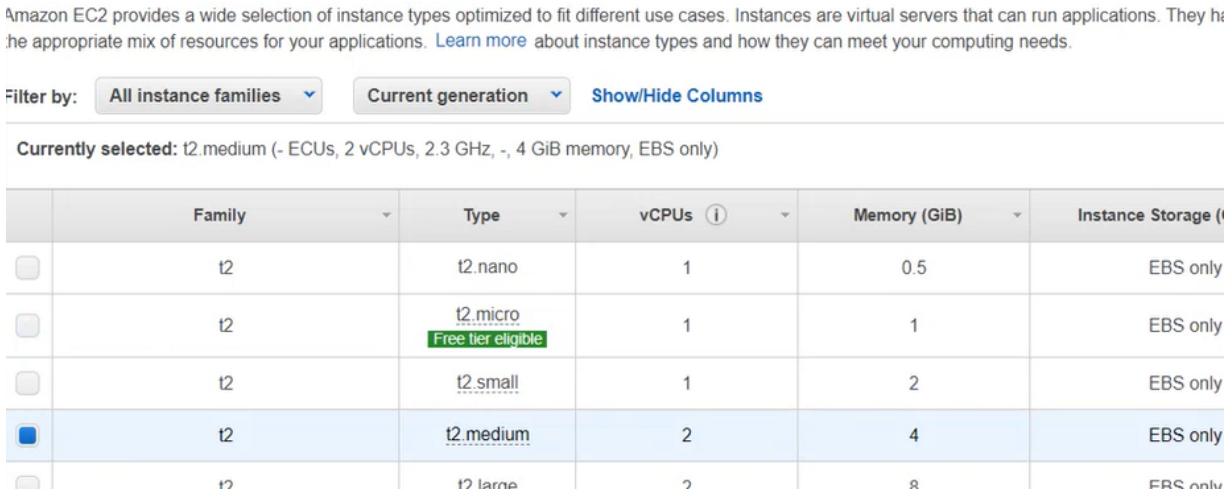
License Server Installation

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

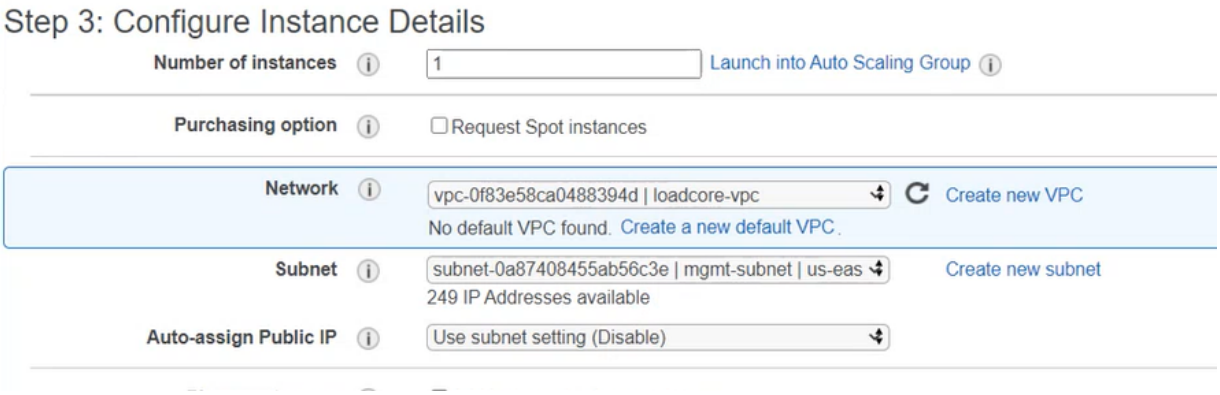
1. From the main console, select **Services** > **EC2** > **Images** > **AMI**.
   The browser will display the AMI images shared for your account.
   Select the License Server image and select **Launch**.



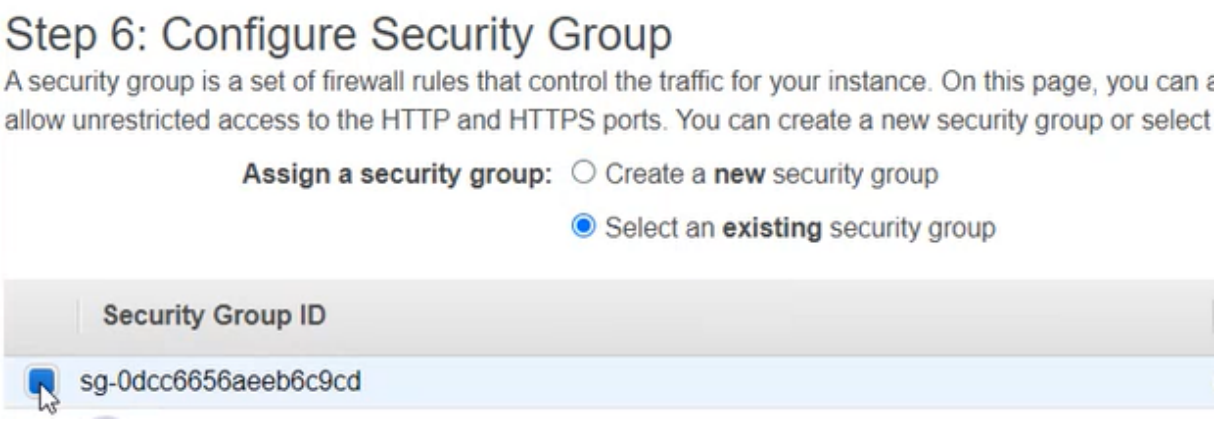
2. Select **t2.medium** for instance type:



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



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

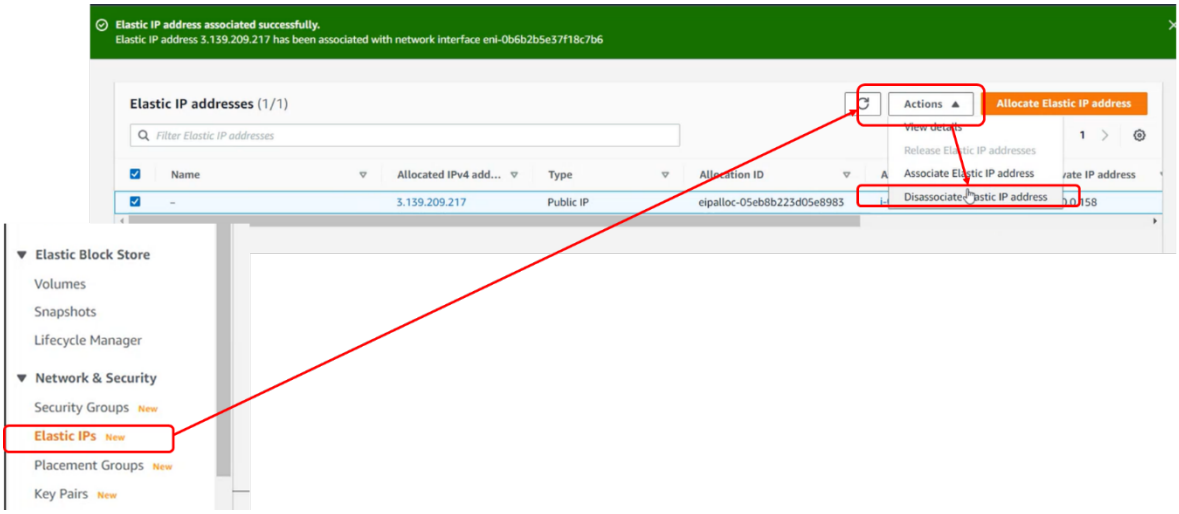


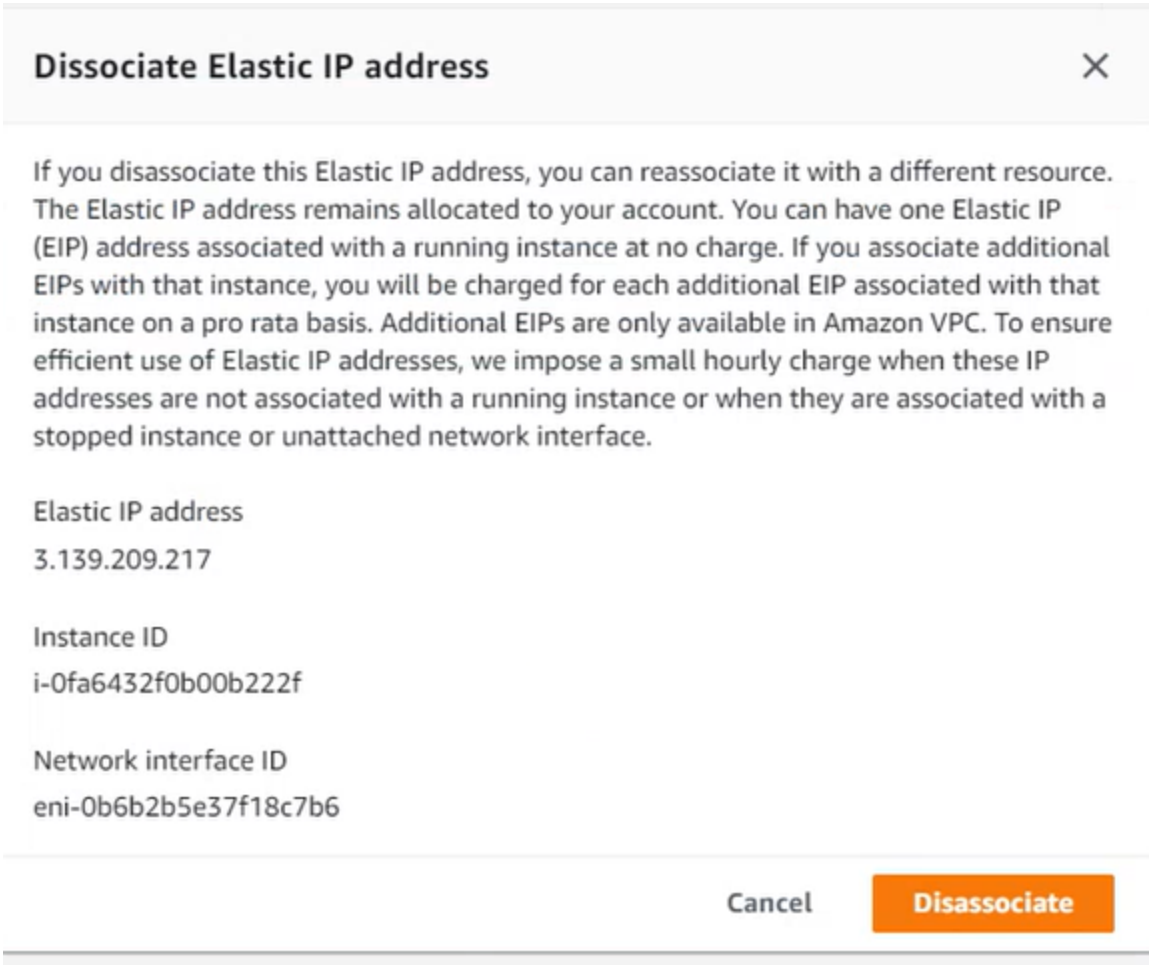
5. Leave the other parameters unchanged and launch the instance.

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:

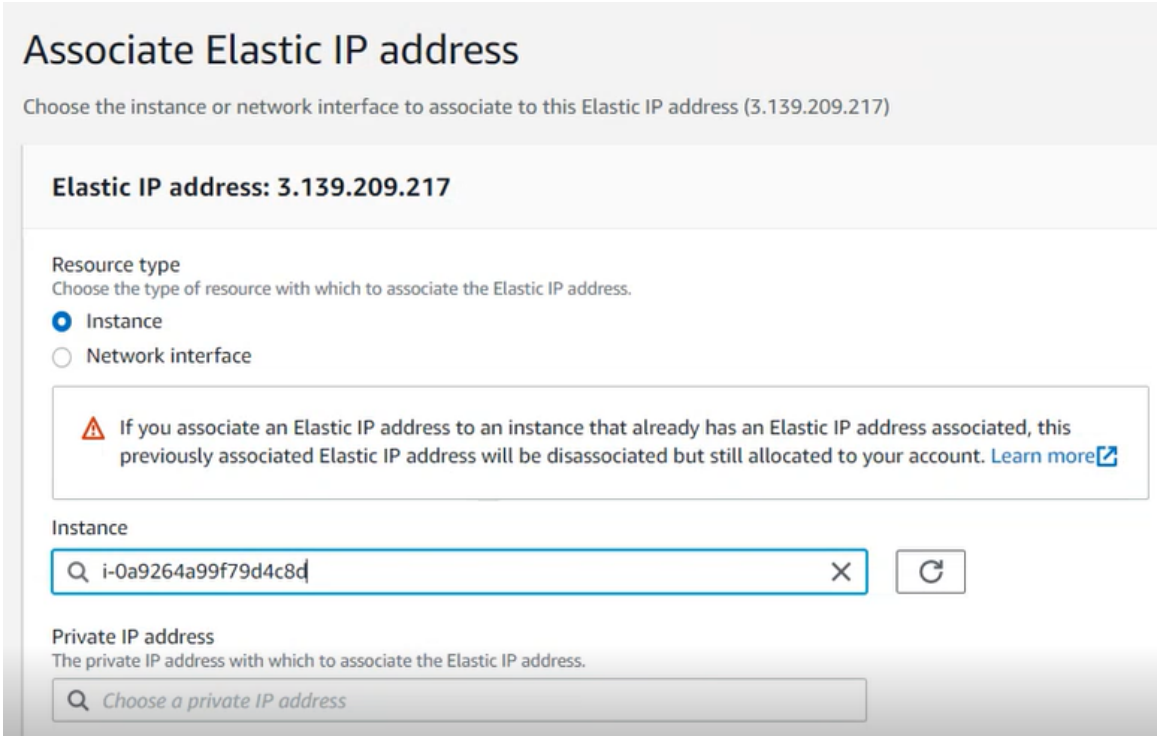
- Select **Elastic IP** > **Action** > **Disassociate**.





- The Elastic IP was disassociated and now should be re-association to the Instance of the License Server.
  Select the following:

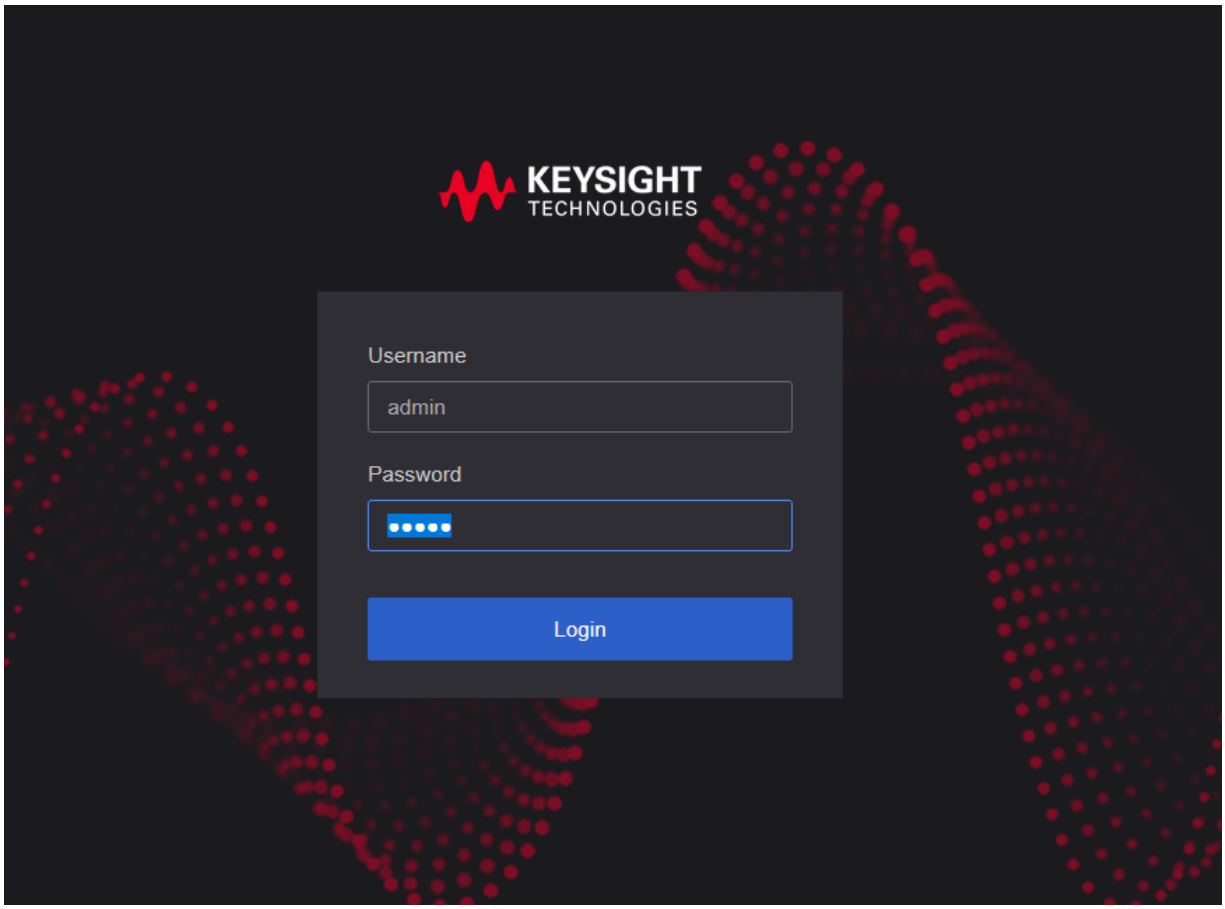
- **Resource type**: **Instance**.
- **Instance id**: select the if of the License server instance.



- 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 ORAN-SIM CE 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 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 ORAN-SIM CE 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 ORAN-SIM CEUI becomes unresponsive

1. Open a SSH connection to 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 ORAN-SIM CE UI gear wheel menu (the upper right corner of the ORAN-SIM CE 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 ORAN-SIM CEUI and check the session status.
   ```

How to recover a ORAN-SIM CE Agent that does not respond when trying to run a test from ORAN-SIM CE UI

1. Reboot the ORAN-SIM CE Agent.
2. Connect to the ORAN-SIM CE 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 ORAN-SIM CE Middleware.
   b. The management interface.
5. Allow the agent to be rebooted from the ORAN-SIM CE 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

KVM Deployment

This section describes the steps needed to deploy ORAN-SIM CE on KVM.

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. kvm64 CPUs are outdated and not supported by QEMU.
  The examples below were done using VMware ESXi.
- **ORAN-SIM CE images** (three images: Middleware, Agent and License server). License Server image is optional, since it can be collocated with MW, but it is recommended to be installed on a separate VM. The images can be downloaded from KSM: https://ksm.software.keysight.com

  IMPORTANT IP connectivity between VMs must to be ensured.
- You will need to use this machine to run ORAN-SIM CE tests. ***ILU (Windows or Linux) cannot be used to run ORAN-SIM CE tests.***
- **Valid licenses** for License server. For deploying full 5G core and run Control plane and User plane traffic, the following licenses will be required:
  - Control plane licenses:
    - ORAN-SIM CE 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 ORAN-SIM CE 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
- 16 GB RAM

- 256 GB SSD

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.
- 32 GB SSD

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

- 8 x vCPUs
- 32 GB RAM
- 32 GB SSD

Licensing server default resources

- 4 x vCPUs
- 8 GB RAM
- 150 GB SSD

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 X710 10G
- Intel XXV710 25G
- Intel XL710 40G
- Intel E810 25G
- Intel E810 100G
- Mellanox ConnectX-4/5 25G
- Mellanox ConnectX-4/5 100G
- Mellanox ConnectX-6 Dx 100G
- KVM Virtual NIC (virtio)

Supported drivers list:

- mlx4_core
- mlx5_core
- ixgbe
- ixgbevf
- i40e
- i40evf
- ice

- iavf
- Virtio

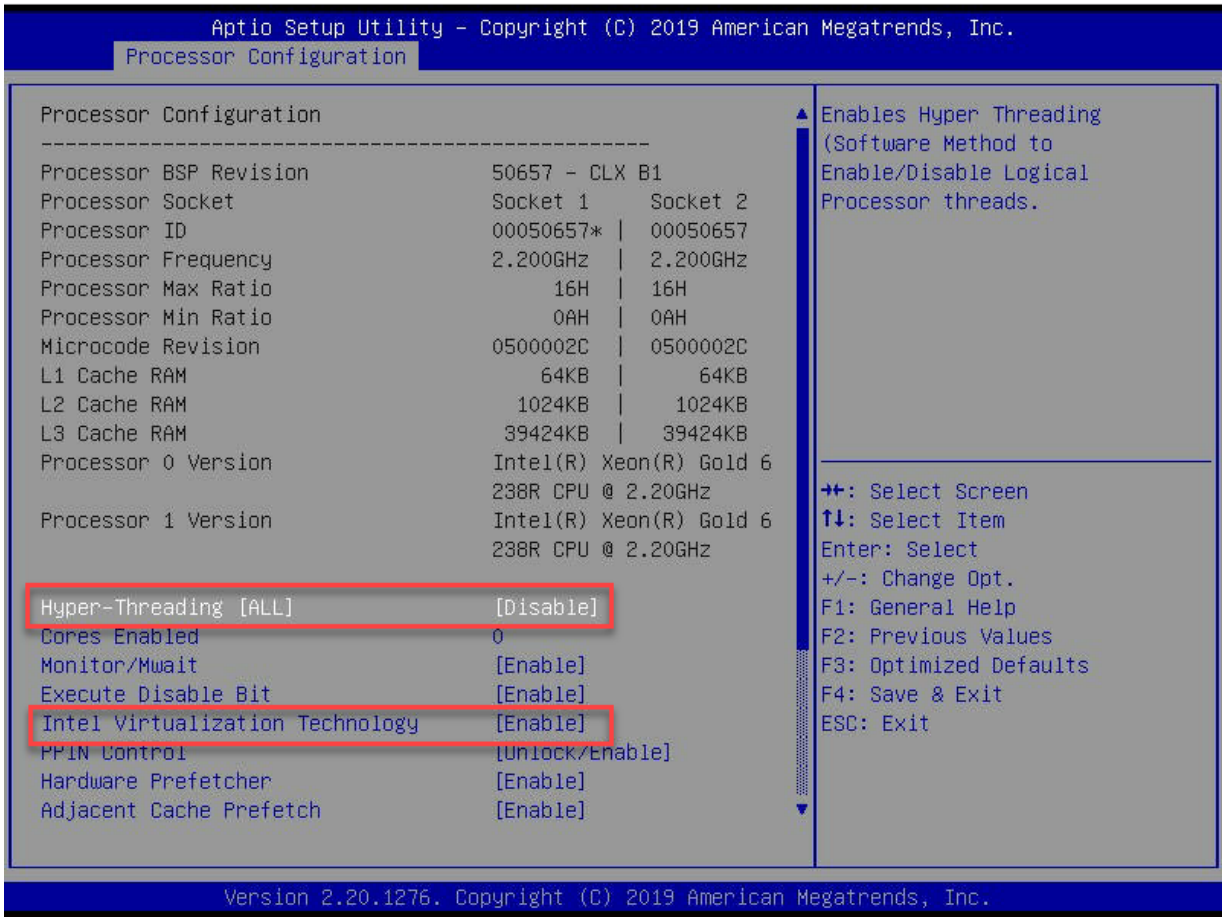
Enable Virtualization

Virtualization needs to be enabled from the hypervisor BIOS settings:

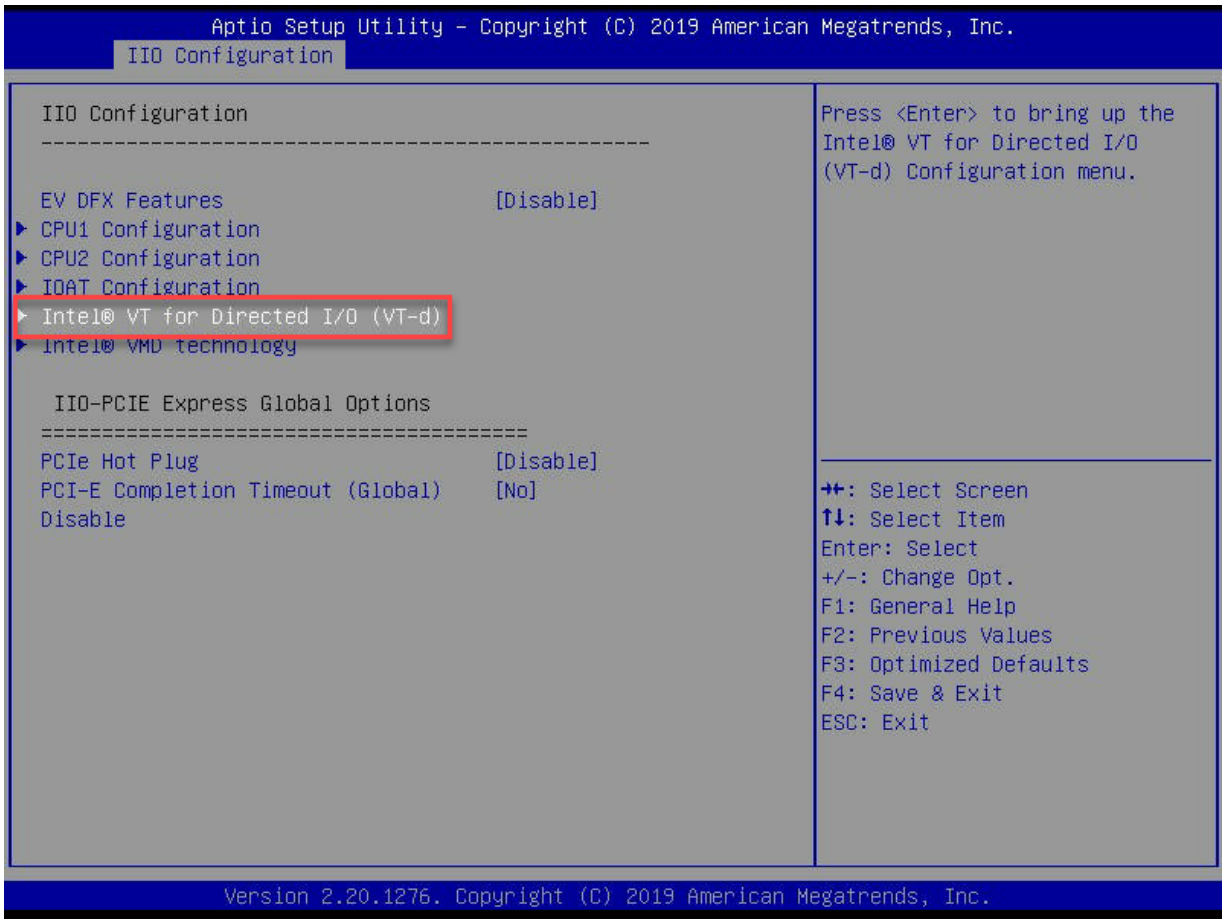
1. Go to **Processor Configuration** in order to enable Intel Virtualization technology.

   NOTE These settings will appear different in BIOS if the hypervisor has an AMD CPU.

   NOTE It is recommended to disable Hyper-threading as it can improve the VM performance.

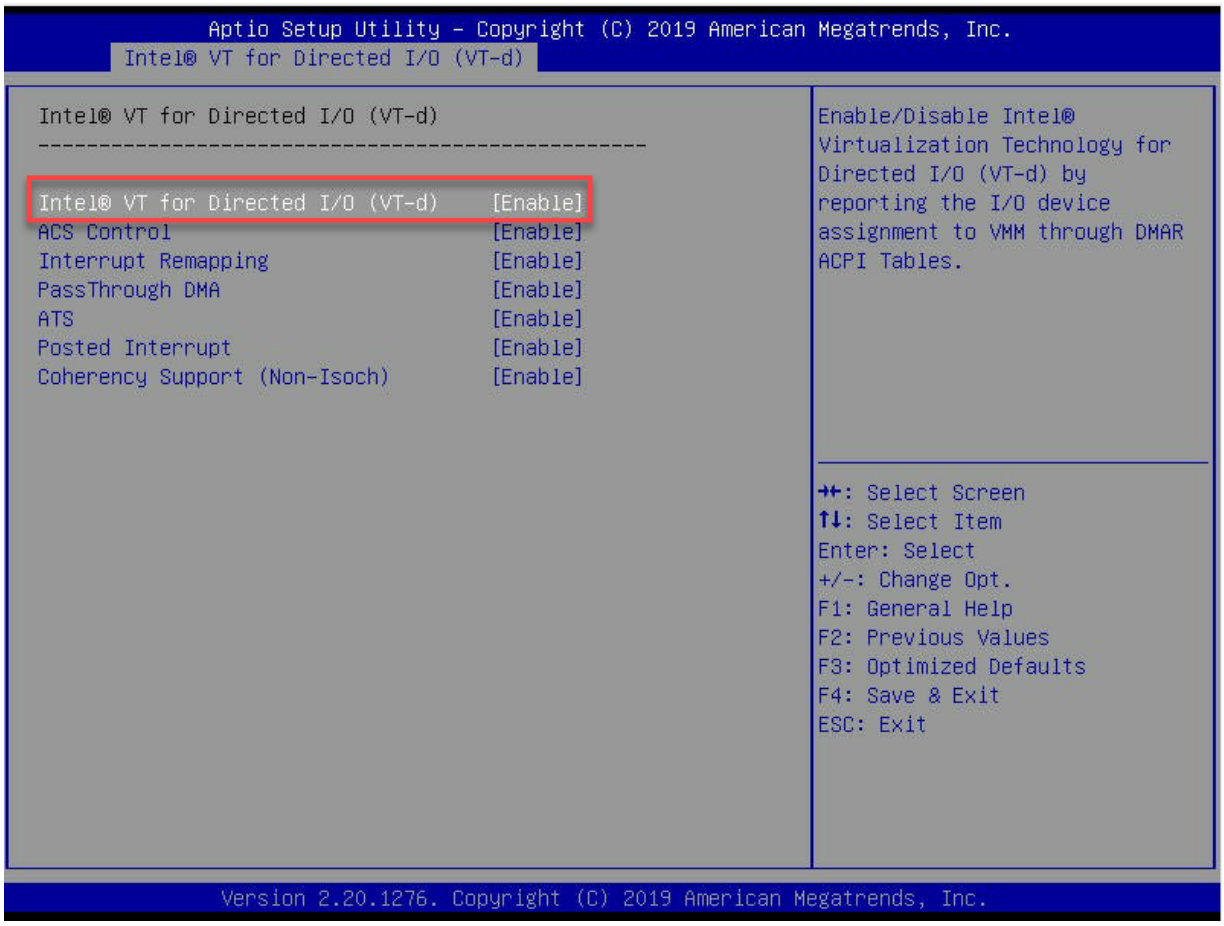


2. From Chipset Configuration, select **North Bridge** > **IIO Configuration**, select **Intel VT for Directed I/O (VT-d)** and press **Enter**.



3. Enable **Intel VT for Directed I/O (VT-d)**. This should automatically enable all the other settings as well.

   NOTE In some configurations, it might display **Intel VT-x**. Also, these settings will be different in case the hypervisor has an AMD CPU.

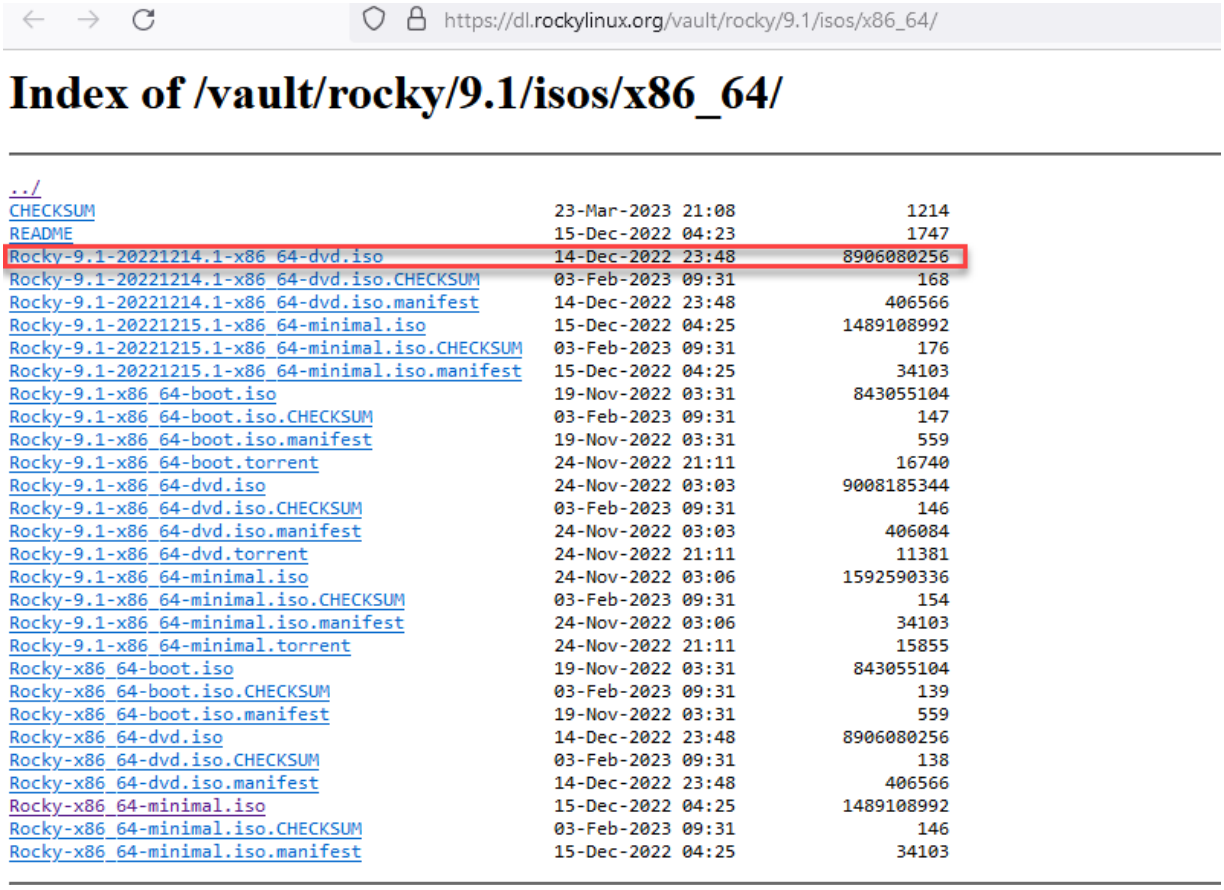


4. After the BIOS settings have been saved, the hypervisor needs to be powered off and on (rebooting is not sufficient to apply the changes).

Install Linux Distribution

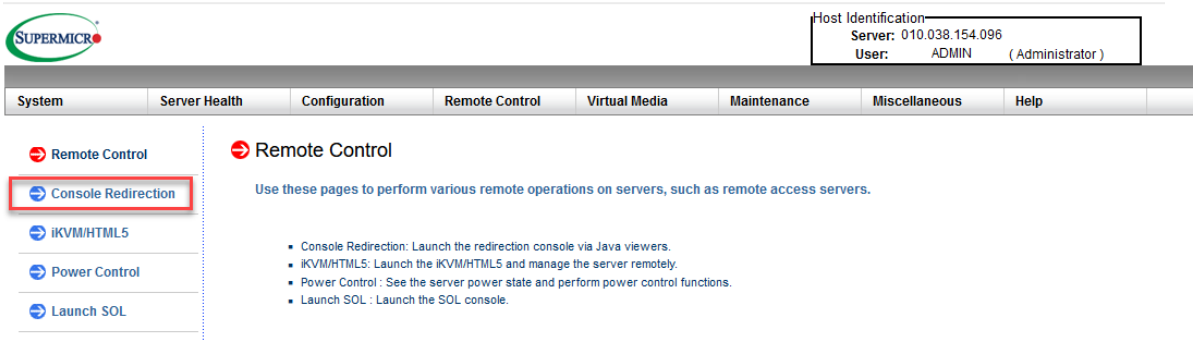
This section provides the steps to install RockyLinux (recommended version 9.1). You can use it to install other Linux distributions, for example CentOS or Ubuntu (recommended version 20.04).

1. Select a `.iso` file as shown below, or use the custom Rocky Linux image provided with the eLSU. For example:

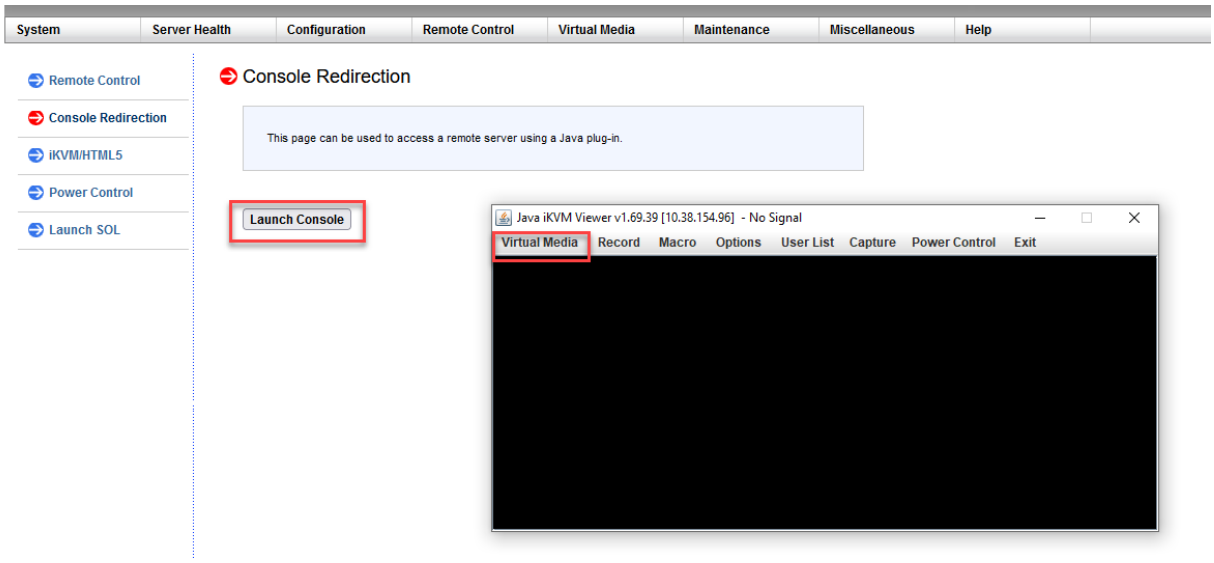


2. After download, go to the IPMI browser page, to Remote Control, and select **Console Redirection:**

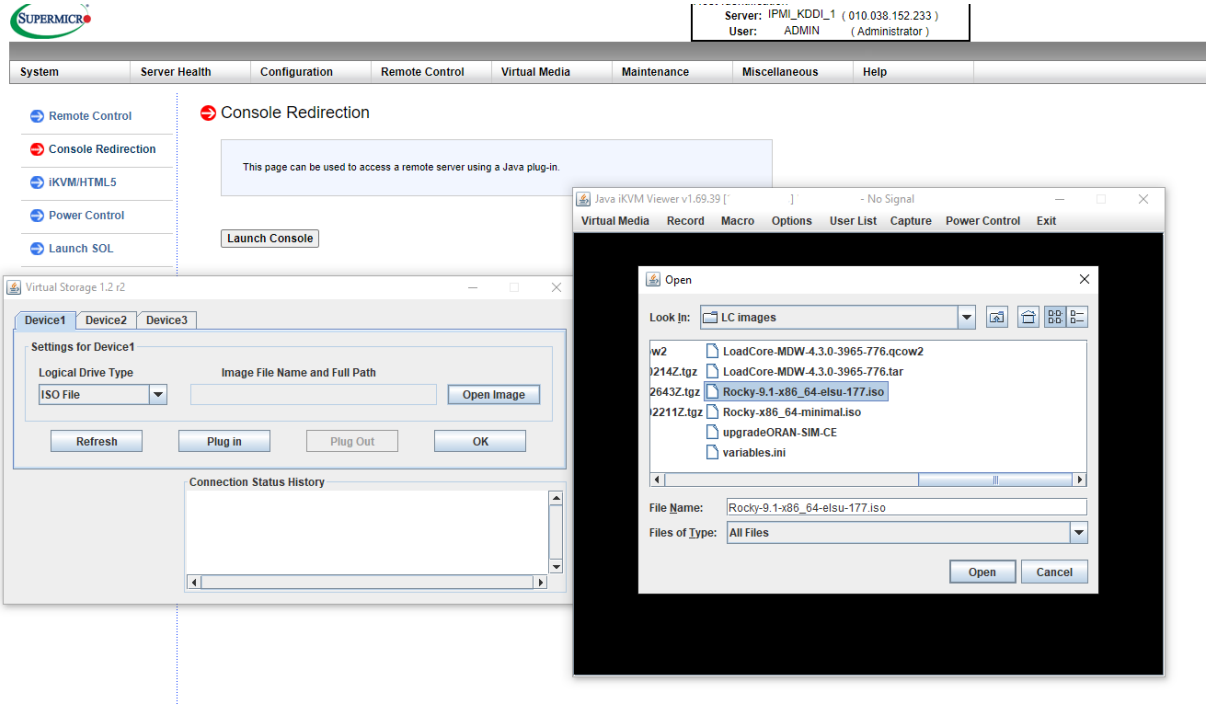
NOTE This step may differ depending on the server manufacturer.



3. Select **Launch Console**, continue through the Java menus until the new window is displayed, and then select **Virtual Media**.



4. Select **Virtual Storage**, set **Device type** as **ISO File** and select the `.iso` file downloaded earlier.

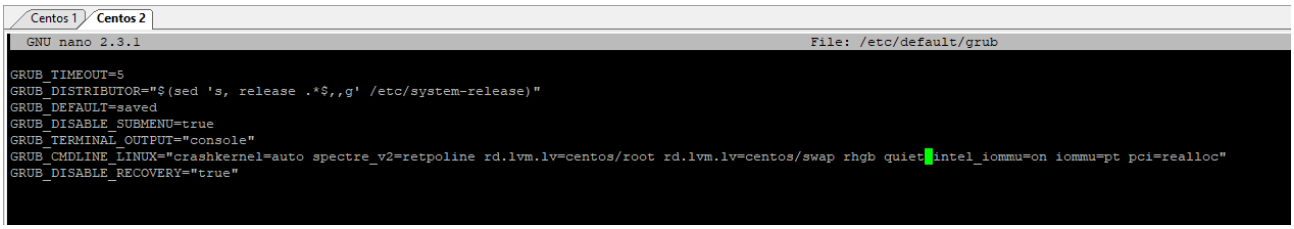


After these steps, it will be possible to install and boot from the `.iso` image.

Install and prepare KVM

Rocky Linux

In `/etc/default/grub` add `intel_iommu=on iommu=pt pci=realloc` at the line mentioned below. For AMD CPUs, it will be `amd_iommu=on`. These settings are needed for KVM device passthrough.



Then do the following command to change the grub file. A reboot will be needed.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

NOTE

You can check if the grub settings have taken effect using the `cat /proc/cmdline` command. If you do not see the new parameters here, you can add them with the following command (reboot after):
`sudo grubby --update-kernel=ALL --args="intel_iommu=on iommu=pt pci=realloc"`

Make sure that the CPU has the virtualization parameter enabled (it should return *VT-d* or *VT-x*):

```
lscpu | grep Virt
```

Or, the following should return *vmx*:

```
cat /proc/cpuinfo | egrep -i "vmx|svm"
```

Use the following commands to install and check KVM:

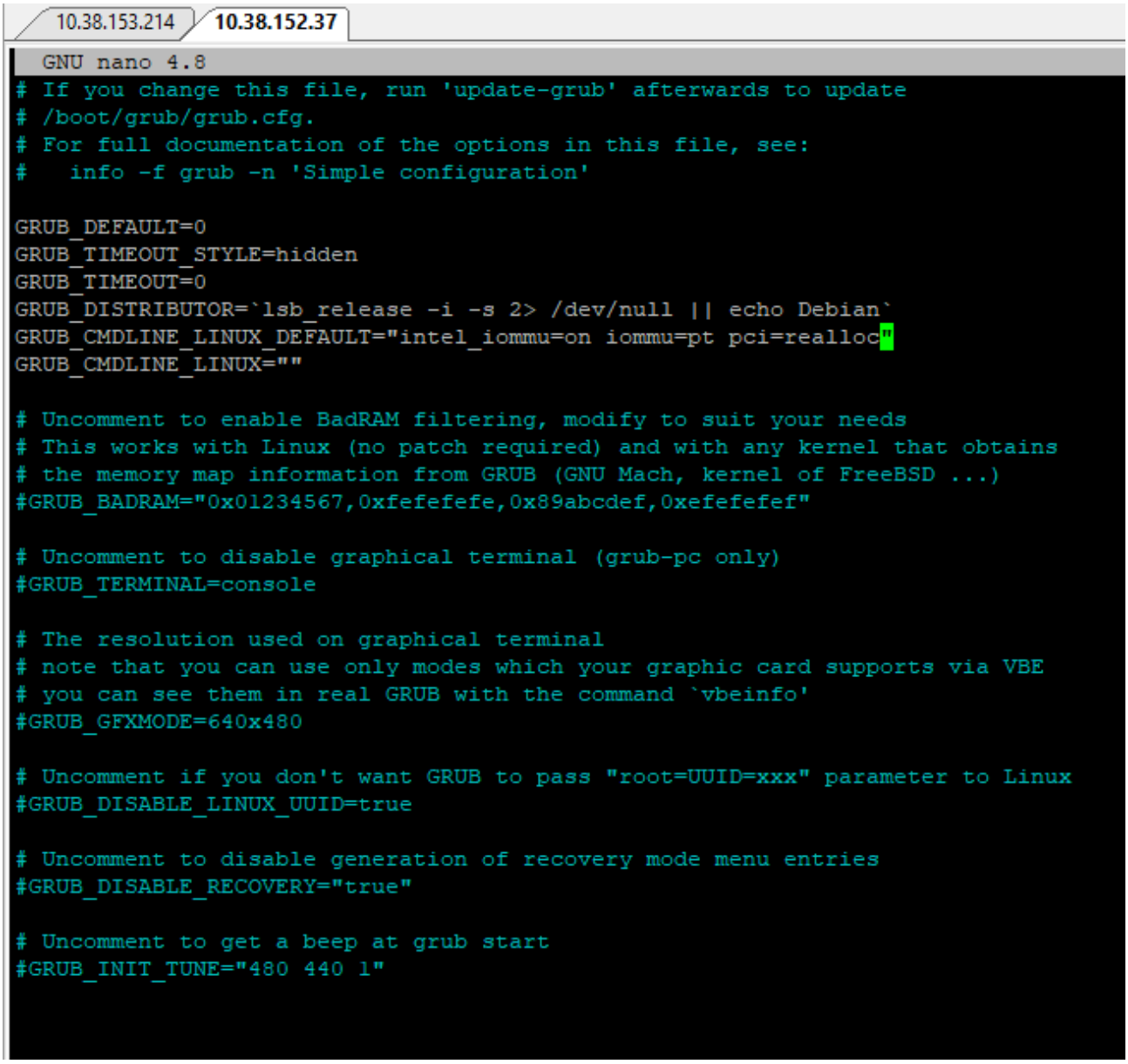
```
dnf update
dnf install qemu-kvm libvirt virt-manager virt-install
systemctl enable --now libvirtd
systemctl status libvirtd
lsmod | grep -i kvm
```

NOTE

Sometimes it is needed to add the (`root`) user to the `kvm`, `qemu` and/or `libvirt` groups. To do this run the following commands (replace `root` with other needed user):
`usermod -a -G libvirt root`
`usermod -a -G kvm root`
`usermod -a -G qemu root`
`groups root`
Sometimes adding the user and the group accordingly in `/etc/libvirt/qemu.conf` is also required.

Ubuntu

In `/etc/default/grub` add `intel_iommu=on iommu=pt pci=realloc` at the line mentioned below. For AMD CPUs, it will be `amd_iommu=on`. These settings are required for KVM device passthrough.



Then do the following command to change the grub file. A reboot will be required.

```
grub-mkconfig -o /boot/grub/grub.cfg
```

Make sure that the CPU has the virtualization parameter enabled (it should return *VT-d* or *VT-x*):

```
lscpu | grep Virt
```

Or, the following should return *vmx*:

```
cat /proc/cpuinfo | egrep -i "vmx|svm"
```

Use the following commands to install and check KVM:

```
apt update
apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager
systemctl enable --now libvirtd
```

```
systemctl status libvirtd
lsmod | grep -I kvm
```

NOTE

Sometimes it is needed to add the (`root`) user to the `kvm`, `qemu` and/or `libvirt` groups. To do this run the following commands (replace `root` with other needed user):

```
usermod -a -G libvirt root
usermod -a -G kvm root
usermod -a -G qemu root
groups root
```

Install Gnome and XRDP

Rocky Linux

Use the following commands to verify if it is set to `graphical.target`, and if not, set it as default:

```
systemctl get-default
systemctl isolate graphical.target
systemctl set-default graphical.target
```

Use the following commands to install and check Gnome:

```
dnf group install "Server with GUI"
gnome-shell --version
```

Use the following commands to install and check XRDP:

```
dnf install epel-release
dnf install xrdp
systemctl enable xrdp.service
systemctl start xrdp.service
systemctl status xrdp.service
```

Make sure port 3389 is added in the firewall:

```
firewall-cmd --add-port=3389/tcp –permanent
firewall-cmd –-reload
```

Ubuntu

Use the following commands to check if it is set to `graphical.target`, and if not, set it as default:

```
systemctl get-default
systemctl isolate graphical.target
systemctl set-default graphical.target
```

Use the following commands to install and check Gnome:

```
apt install ubuntu-gnome-desktop
gnome-shell --version
```

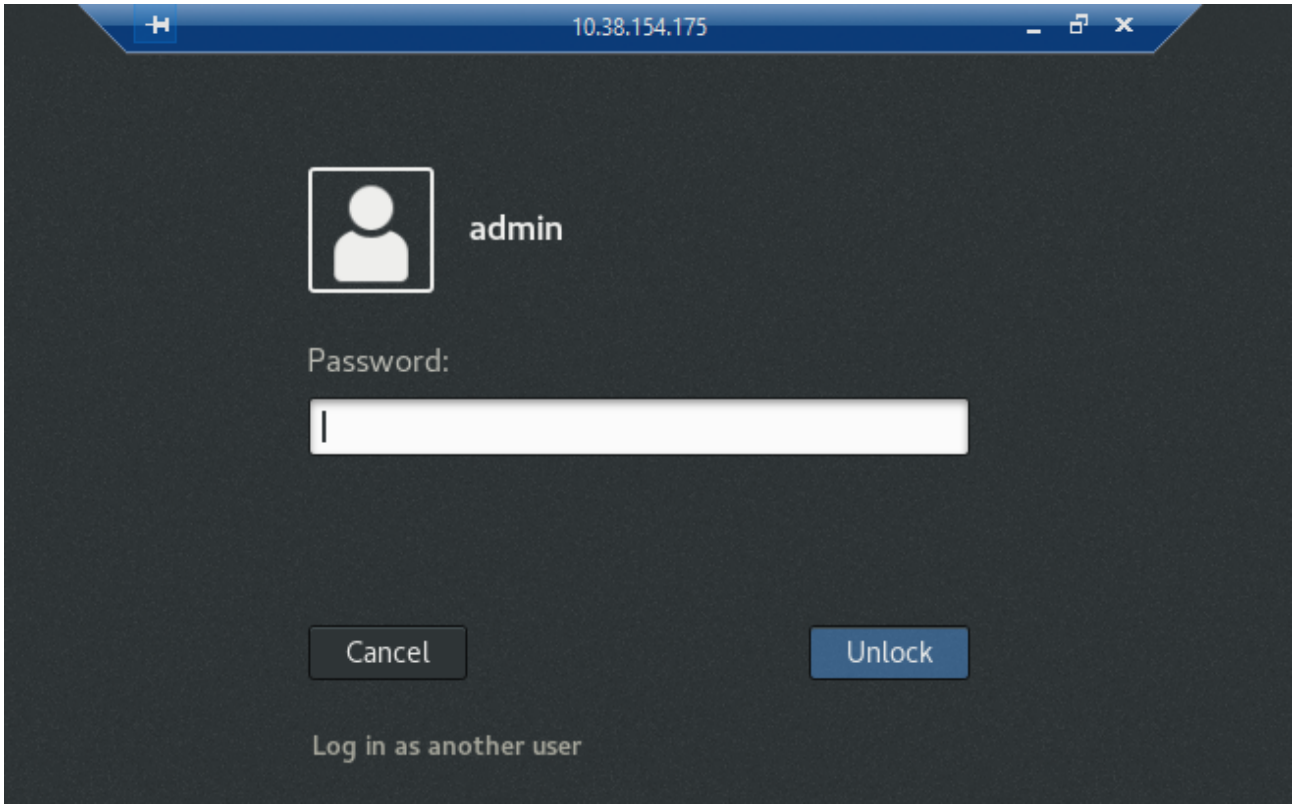
Use the following commands to install and check XRDP:

```
apt install xrdp
systemctl enable xrdp.service
systemctl start xrdp.service
systemctl status xrdp.service
```

Make sure port 3389 is added in the firewall:

```
ufw allow 3389/tcp
```

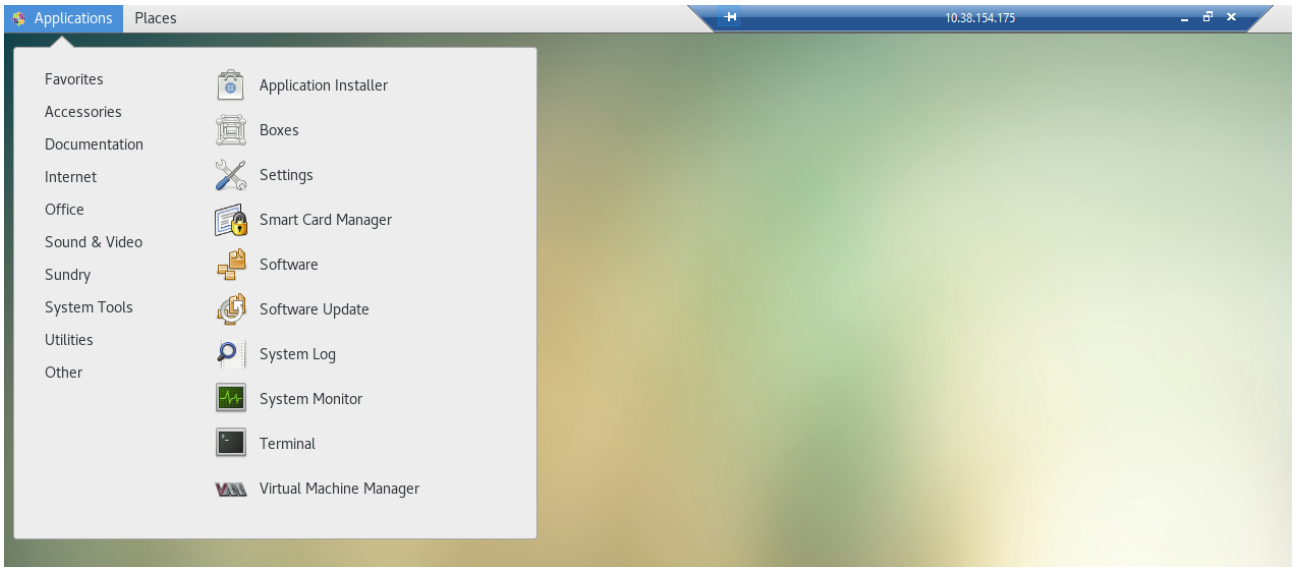
Now it should be possible to connect to the hypervisor with a Remote Desktop application:



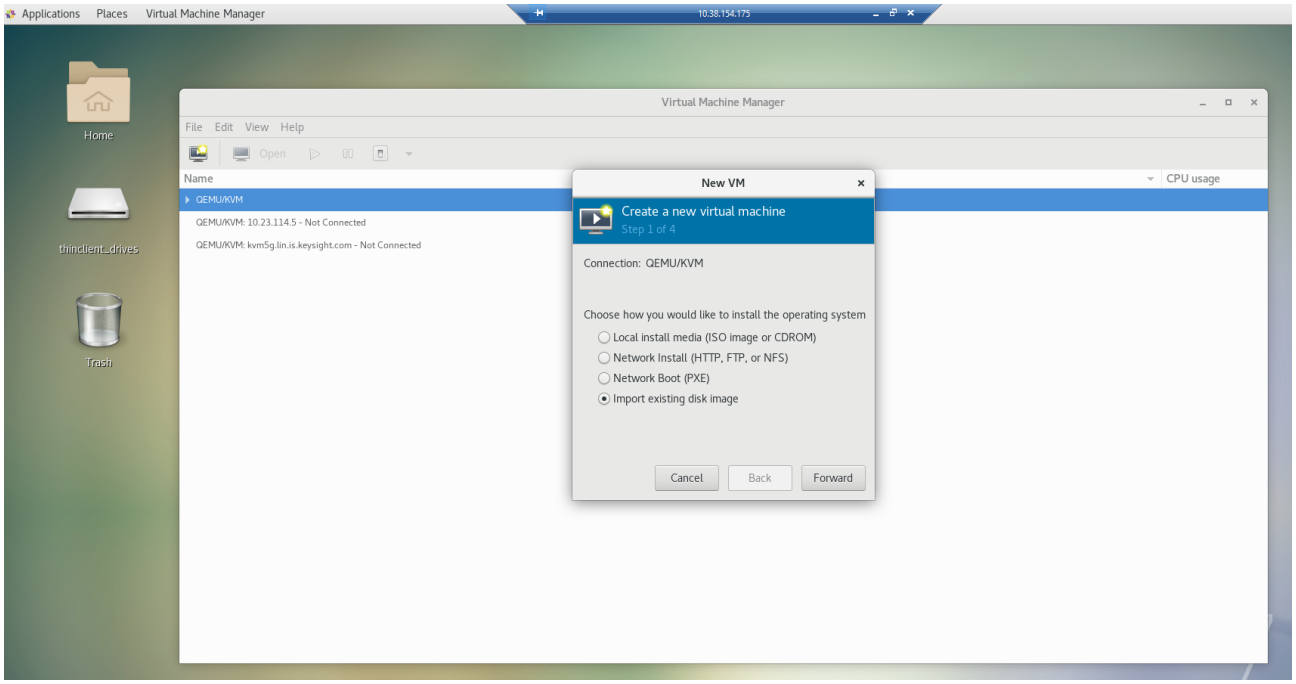
Install LoadCore VMs

Upload the needed QCOW images to the hypervisor through WinSCP, Moba, VNC or any other application that can use FTP.

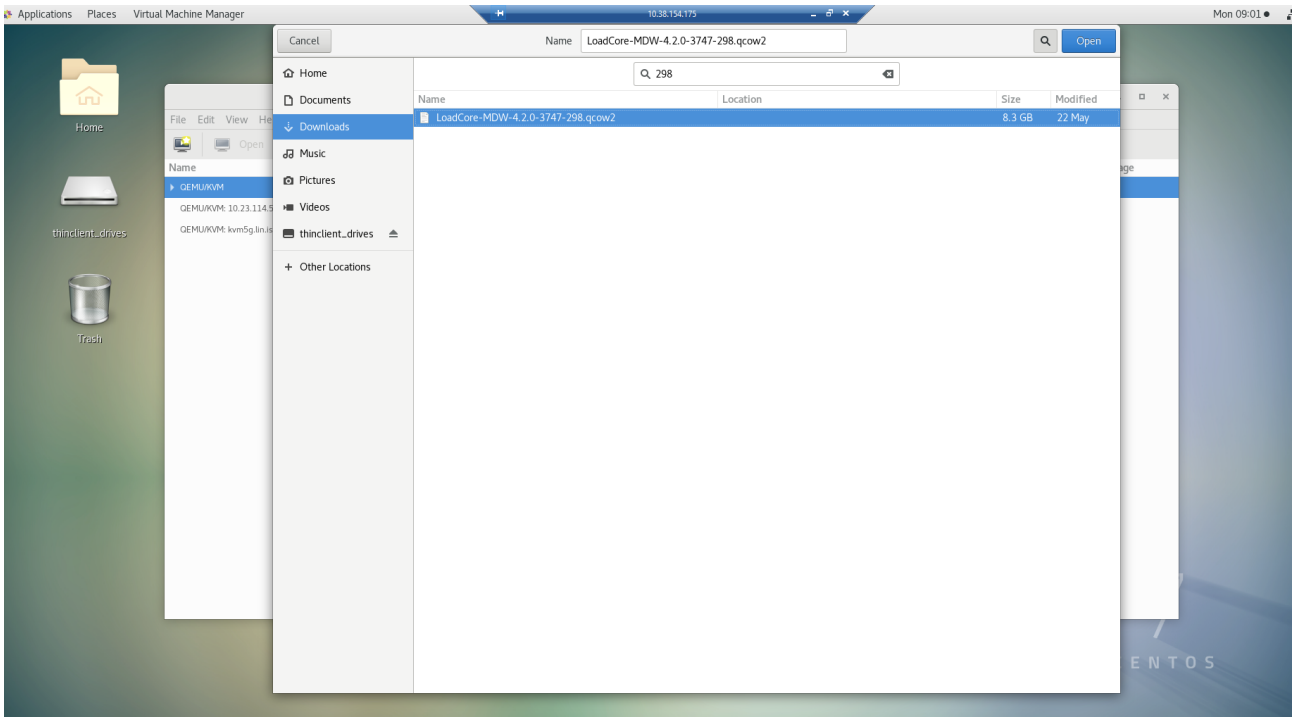
Go to the **Applications** button> **System tools** and then open **Virtual Machine Manager**:



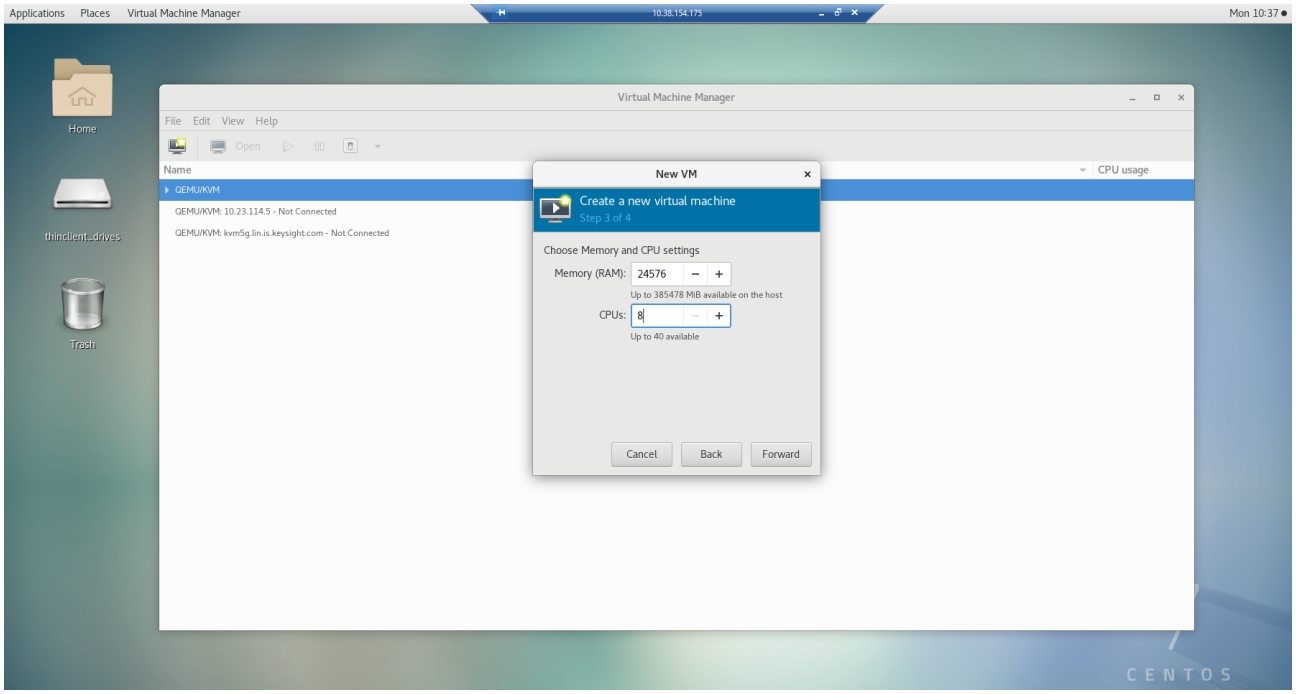
Select the new VM button (upper left) and select the **Import existing disk image** option:



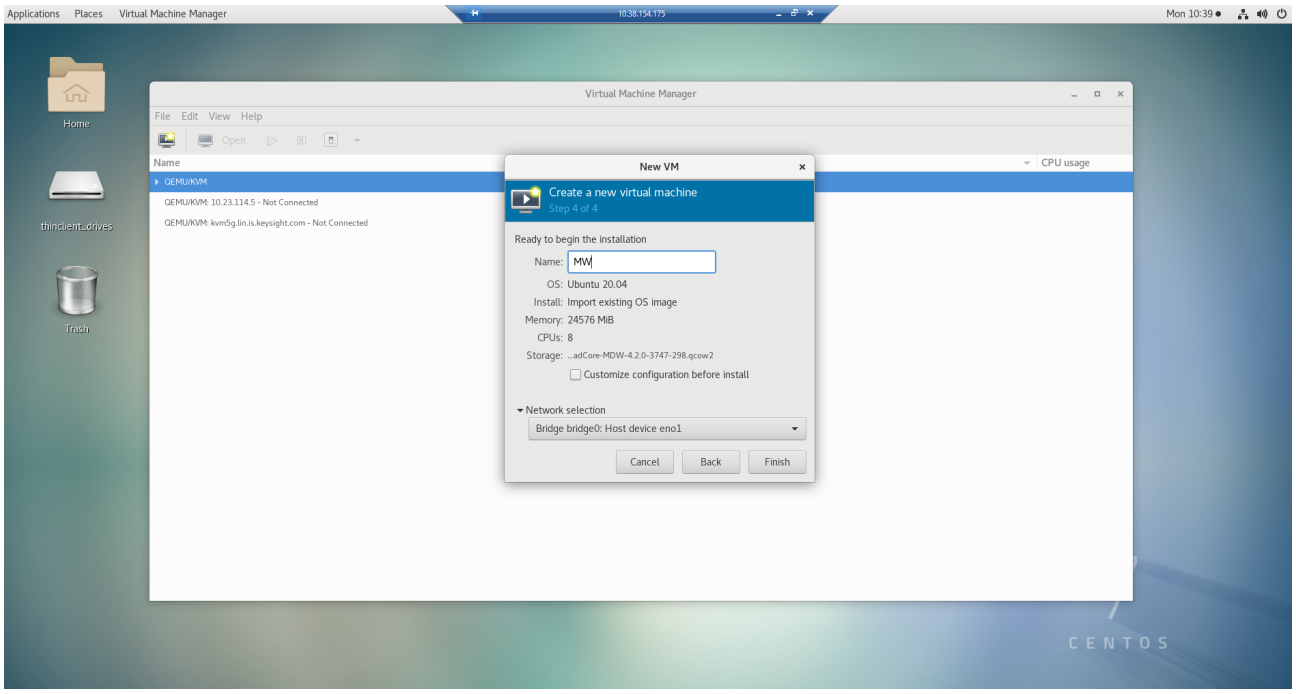
Select the image to be loaded:



Select the appropriate resources:



Provide the VM name and select a management interface:



Alternatively the VMs can also be deployed from the hypervisor command line, with commands like the following:

```
virt-install --name LoadCore-MW --os-type=Linux --os-variant=ubuntu22.04 --memory=24576
--vcpus=8 --disk=/home/keysight/Desktop/qcow2_files/ LoadCore-MDW-4.2.0-3747-298.qcow2 -
-network bridge=bridge0,model=e1000 --import –noautoconsole
```

or

```
sudo virt-install --import --name loadcore-agent-1 \
      --memory 32768 --vcpus 8 --cpu host-passthrough \ <- 32GB ram and 8 CPUs
recomanded for agent VMs
      --disk loadcore-agent-2.qcow2,format=qcow2,bus=virtio \ <- provide disk image
      --network type=direct,source=eno1,source_mode=bridge,model=e1000e \ <- configures
management network as dirrectly connected to host's network
      --os-variant=ubuntu22.04 \
      --host-device=pci_0000_d8_00_1 \ <- enable pci passthrough for test NIC
      --noautoconsole
```

To connect from the hypervisor to the VM:

```
virsh console LoadCore-MW
```

IMPORTANT

The recommended order for booting up the VMs is the following:

- first, the License Server, if a dedicated one has been deployed (this is the VM that uses the least amount of resources)
- next, the MW
- last, after a few minutes, start the agents one by one.

This order is also according to dependencies/communication between the VMs.

Add virtual bridge

This step is optional, but recommended to use if there is only one interface for management assigned on the hypervisor.

Rocky Linux

First, create the bridge with the following command (replace `br0` with the name of the bridge you want to give):

```
nmcli con add type bridge ifname br0
```

Assign a slave interface to the bridge (replace `enp6s0` with the desired interface):

```
nmcli con add type bridge-slave ifname enp6s0 master br0
```

Assign a static IP address and a gateway to the bridge (DHCP can also be used):

```
nmcli con mod bridge-br0 ipv4.addresses 192.168.1.100/24 ipv4.gateway 192.168.1.1
connection.autoconnect true
```

Bring down and delete the old connection on enp6s0:

```
nmcli con down enp6s0
nmcli con delete enp6s0
```

Bring the bridge up:

```
nmcli con up br0
```

You can also check the device and connection status with:

```
nmcli dev status
nmcli con show
```

NOTE

In some setups you may need to disable spanning tree, by executing the following command:

`nmcli connection modify bridge-br0 bridge.stp no`

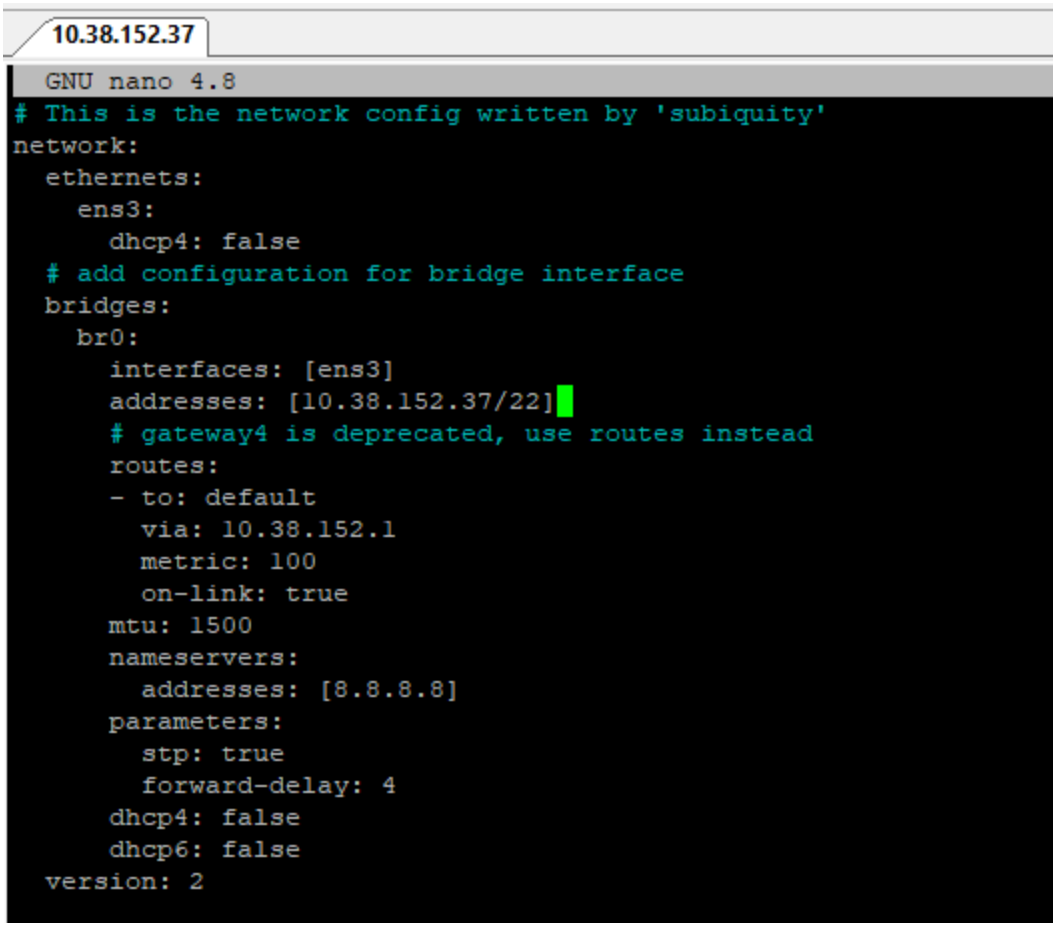
Ubuntu

First install `bridge-utils`:

```
apt install bridge-utils -y
```

Open the *yaml* file found in `/etc/netplan` (it can be `00-installer-config.yaml`, `network.yaml`, `01-network-manager-all.yaml`). Add the bridge lines as in the example below for static IP:

```
bridges:
    br0:
      interfaces: [ ens3 ]
      dhcp4: false
      addresses: [10.38.152.37/22]
      routes:
      - to: default
        via: 10.38.152.1
```



For DHCP you can just set the `dhcp4` parameter to true and comment or remove the static ip address:

```
bridges:
    br0:
      interfaces: [ ens3 ]
      dhcp4: true
```

Then `try` and `apply` the `netplan`:

```
netplan try
netplan apply
```

Azure Cloud Deployment

The requirements for deploying ORAN-SIM CE 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.

OpenStack Deployment

This section describes the steps needed to deploy ORAN-SIM CE on OpenStack.

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.
- **ORAN-SIM CE images** (three images: Middleware, Agent and License server). License Server image is optional, since it can be collocated with MW, but it is recommended to be installed on a separate VM. The images can be downloaded from KSM: https://ksm.software.keysight.com

  IMPORTANT IP connectivity between VMs must to be ensured.
- You will need to use this machine to run ORAN-SIM CE tests. ***ILU (Windows or Linux) cannot be used to run ORAN-SIM CE tests.***
- **Valid licenses** for License server. For deploying full 5G core and run Control plane and User plane traffic, the following licenses will be required:
  - Control plane licenses:
    - ORAN-SIM CE 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 ORAN-SIM CE 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
- 16 GB RAM

- 256 GB SSD

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.
- 32 GB SSD

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

- 8 x vCPUs
- 32 GB RAM
- 32 GB SSD

Licensing server default resources

- 2 x vCPUs
- 8 GB RAM
- 100 GB SSD

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 X710 10G
- Intel XXV710 25G
- Intel XL710 40G
- Intel E810 25G
- Intel E810 100G
- Mellanox ConnectX-4/5 25G
- Mellanox ConnectX-4/5 100G
- Mellanox ConnectX-6 Dx 100G
- KVM Virtual NIC (virtio)

Supported drivers list:

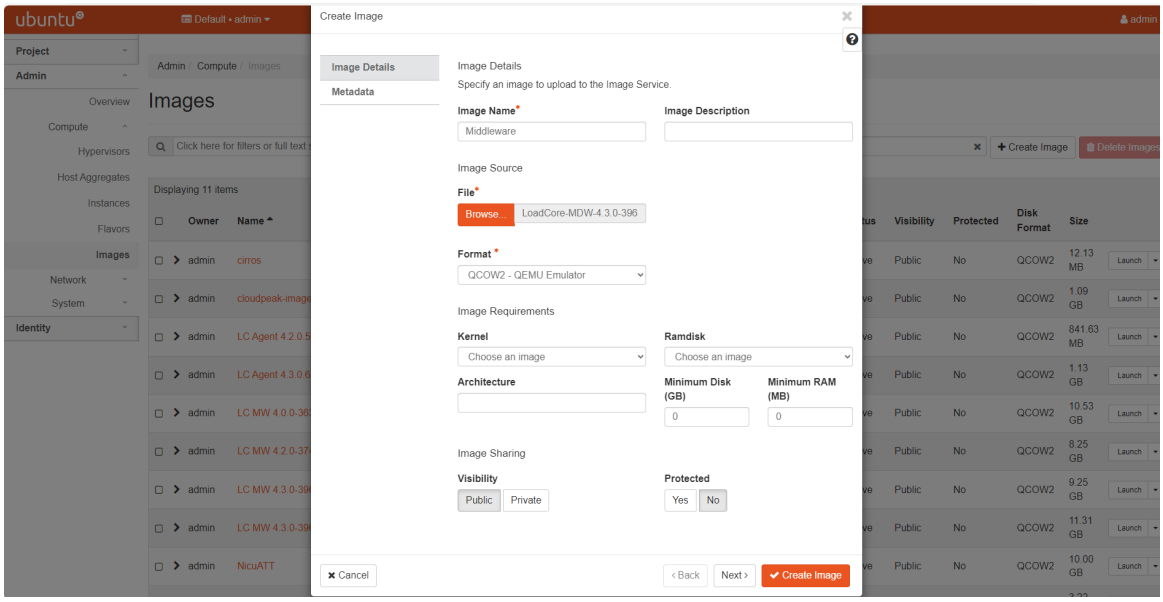
- mlx4_core
- mlx5_core
- ixgbe
- ixgbevf
- i40e
- i40evf
- ice

- iavf
- Virtio

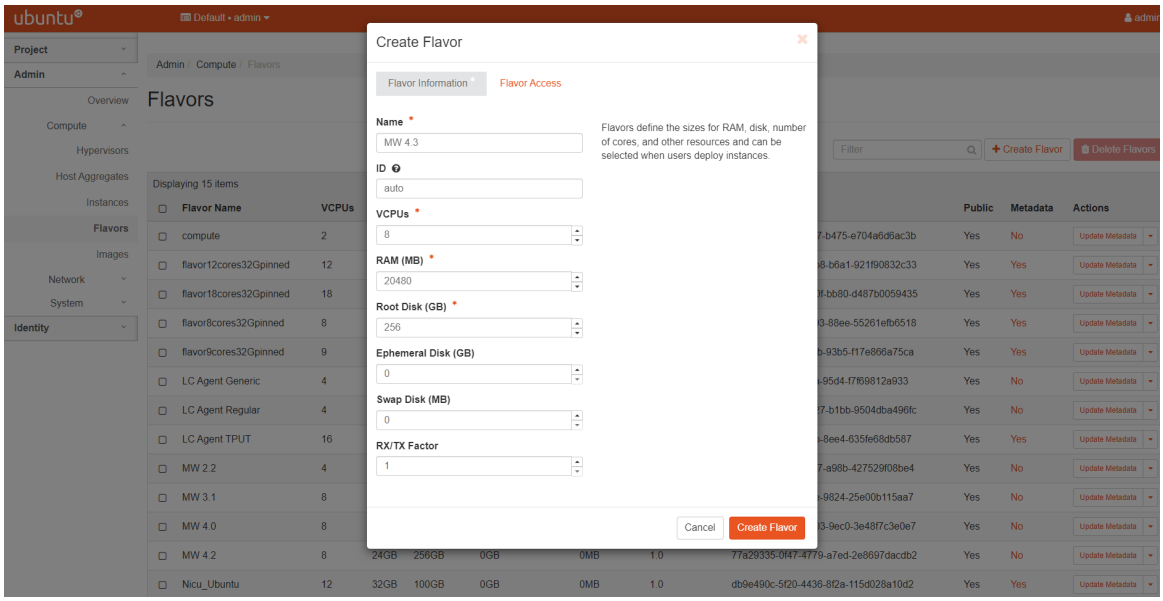
Middleware installation

The installation procedure of the Middleware requires the following steps:

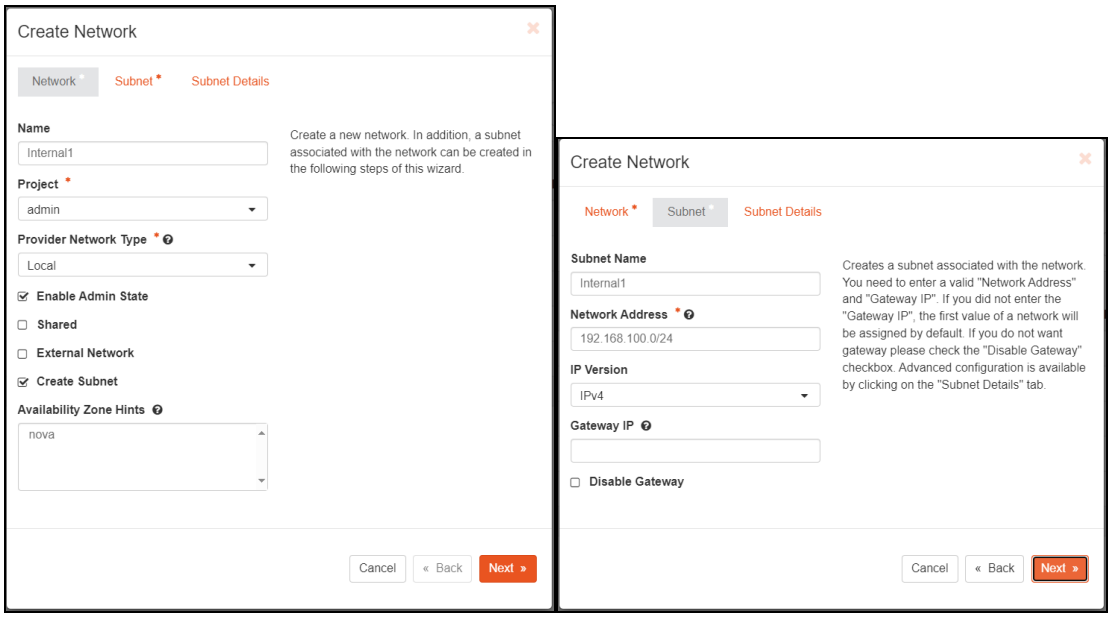
1. In OpenStack, navigate the left menu to **Admin > Compute > Images** section and select **Create Image**. Input an appropriate image name, browse and select the *MW qcow2* file, select the QCOW2 format and finalize by clicking on **Create Image** button.



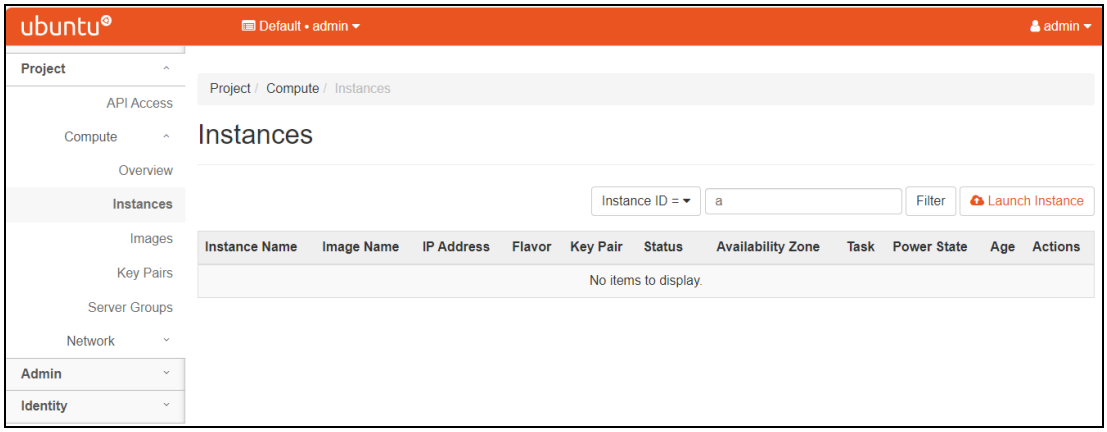
2. Navigate to **Admin > Compute > Flavors** section and select **Create Flavor**. Input an appropriate flavor name and choose the required resources to be assigned (in this case 8 vCPUs, 32 GB RAM, and 256 GB storage), and press **Create Flavor** to confirm.



3. If not already created, it is recommended to create a management network to be used for communication between middleware, agents and license server. To do this, navigate to **Admin > Networks** section and select **Create Network**. Input an appropriate name, select the project for which the network will be created, and choose a network address. Click **Next** to confirm.

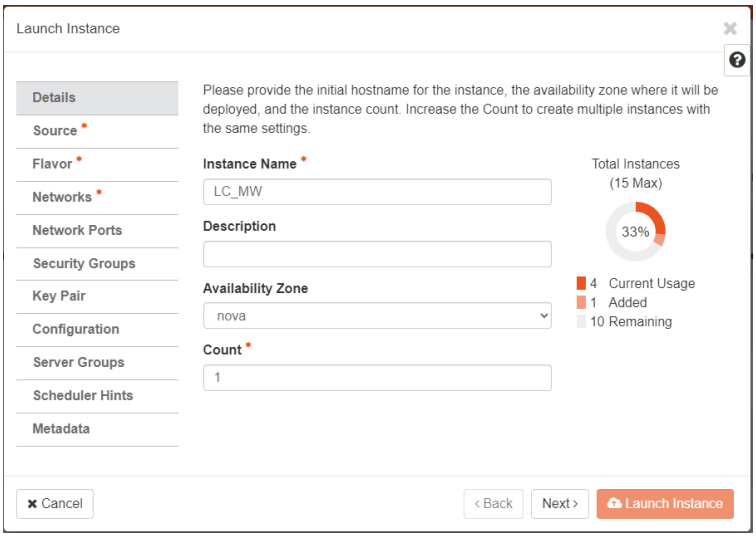


4. Navigate to **Project > Compute > Instances** section and click **Launch Instance**.

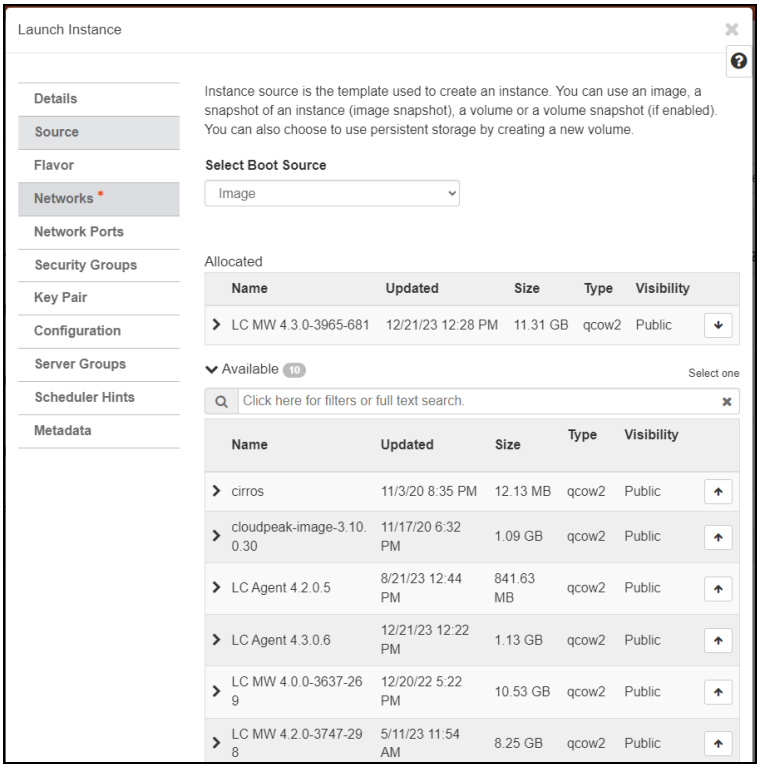


5. In the **Launch Instance** dialog (these are mandatory configurations marked by asterisk):

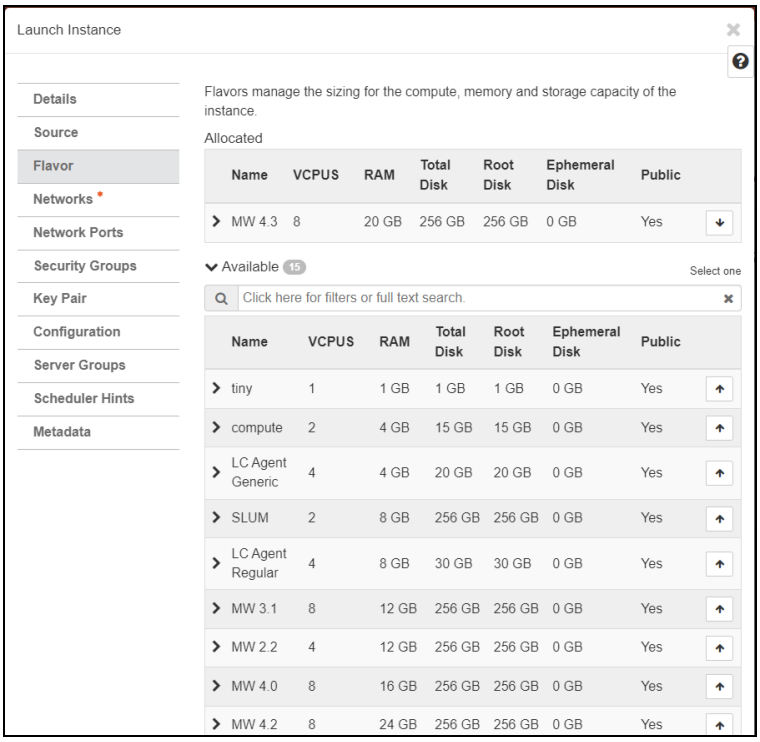
- select **Details** section to input an appropriate *name* to this instance



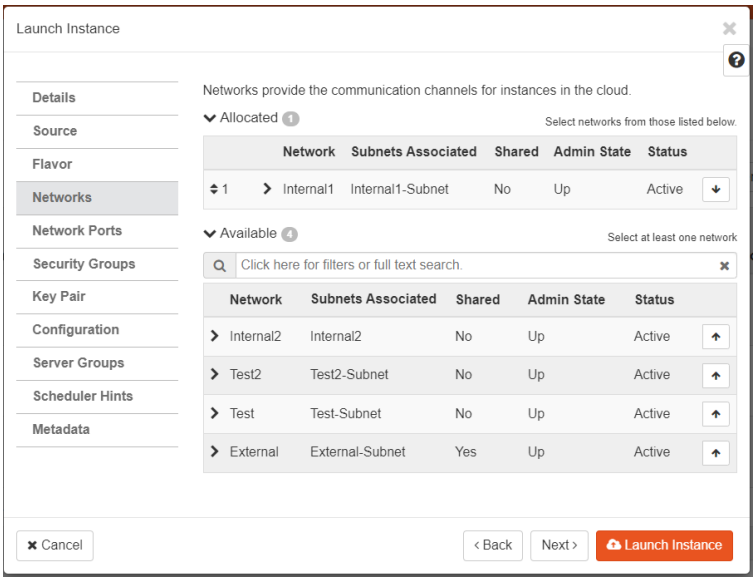
- select **Source** section and choose the newly created *image* as boot source



- select the **Flavor** section to add the newly created *flavor* that allocates the resources

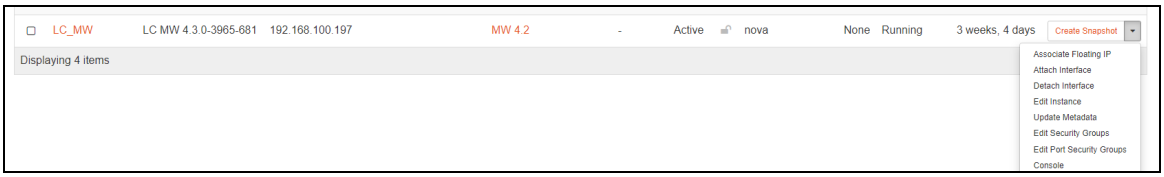


- select the **Network** section and add the new *network* as communication channel.



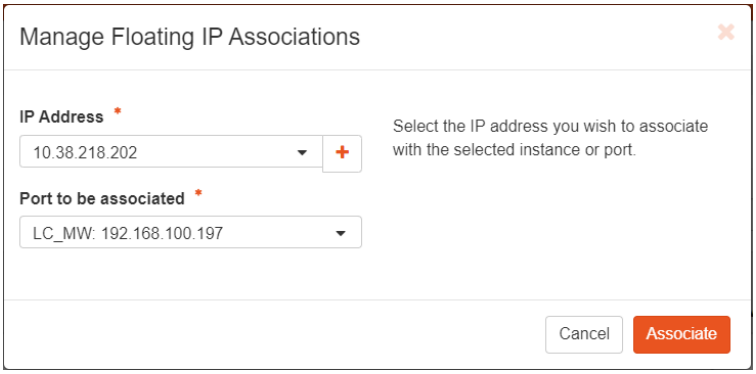
Confirm the configuration by pressing the **Launch Instance** button.

6. **Optional Step:**After the Middleware instance has been deployed, assign a floating IP so you can access it from an external network:
   a. from the **Project > Compute > Instances** section, go to the allocated image and, from the drop-down menu at the end of the row select **Associate Floating IP**.



b. in the **Manage Floating IP Association** dialog, select an *IP address* from the drop-down list, or create one by clicking the **+** button.

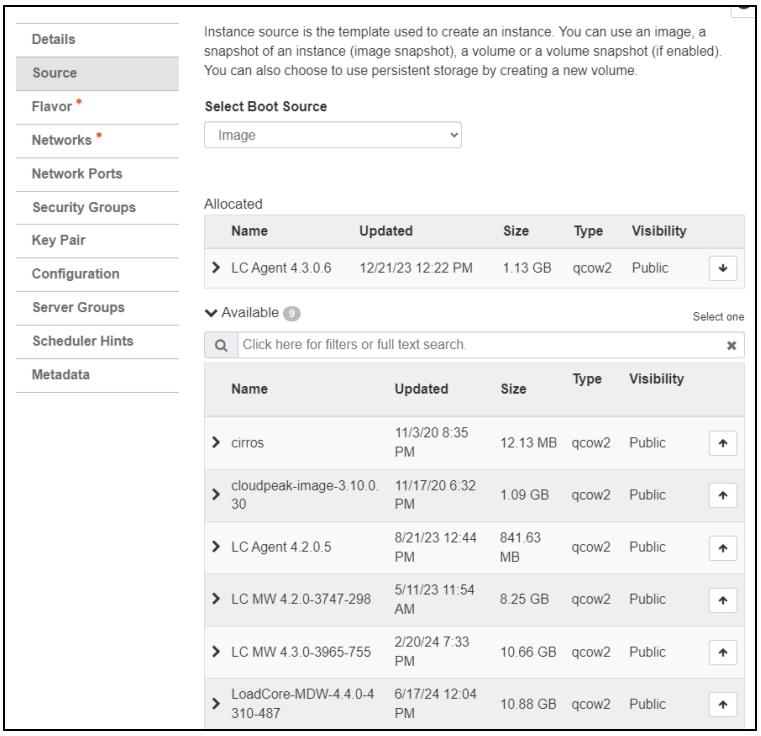
c. click **Associate** to conclude this step.



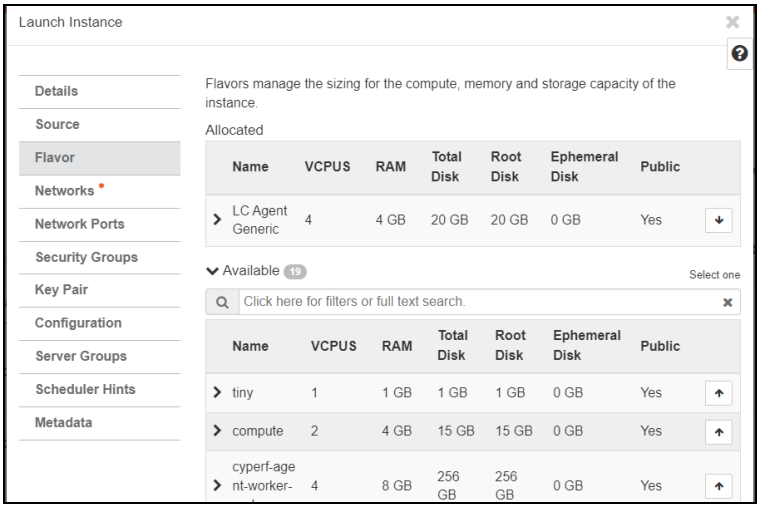
Agent installation and configuration

Agent installation follows similar steps to Middleware.

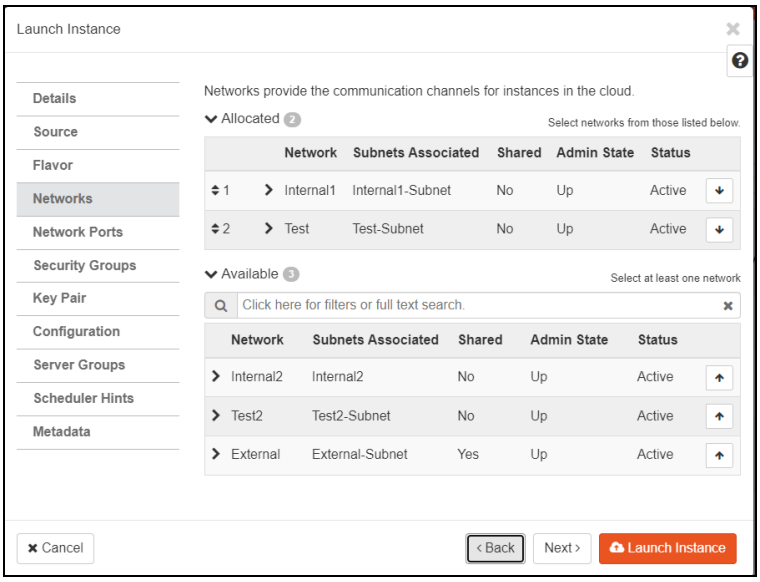
1. In OpenStack, navigate the left menu to **Admin > Compute > Images** section and select **Create Image**. Input an appropriate agent name, such as *LC Agent 4.3.0.6*, then browse and select the LC agent image file, select the QCOW2 format and finalize by clicking on **Create Image** button.
2. Navigate to **Admin > Compute > Flavors** section and select **Create Flavor**. Input an appropriate flavor name, such as *LC_Agent Generic*, and choose the required resources to be assigned, and press **Create Flavor** to confirm.
3. Agents require at least one management interface for communication with middleware, and at least a test interface, that will be used to communicate to either another LC agent or a DUT. Navigate to **Admin > Networks** section and select **Create Network**, then name it as *Test*; select the project for which the network will be created, and choose a network address. Click **Next** to confirm.
4. Navigate to **Project > Compute > Instances** section and click **Launch Instance**.
   - select **Details** section to input an appropriate *name* to this agent instance, such as *LC_ Agent_1*.
   - select **Source** section and choose the created agent image



- select the **Flavor** section to add the agent's resource. Note that the agent flavor can be different according to the agent purpose. For example, if it is used for Application Traffic, then pinned CPUs and PCI passthrough may be required, or if used only for simulating Control Plane procedures, it can have lower resources.



- select the **Network** section and add the two network types.



5. Confirm the configuration by pressing the **Launch Instance** button.
6. To register the agent to the Middleware, connect via SSH (user/password are *ixia*/*ixia*), and run `sudo /home/ixia/agent-setup.sh` command. Input the MW IP (either the internal IP or the floating IP), and answer with *y*/*n* the questions that follow.

```
18/06/2024 11:05.36 /home/mobaxterm ssh ixia@10.38.2.82
Ubuntu 22.04 LTS
X11 forwarding request failed on channel 0
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Mon Jun 17 11:18:43 2024 from 10.20.132.215
ixia@lc-agent-1:~$ sudo ./agent-setup.sh
[sudo] password for ixia:

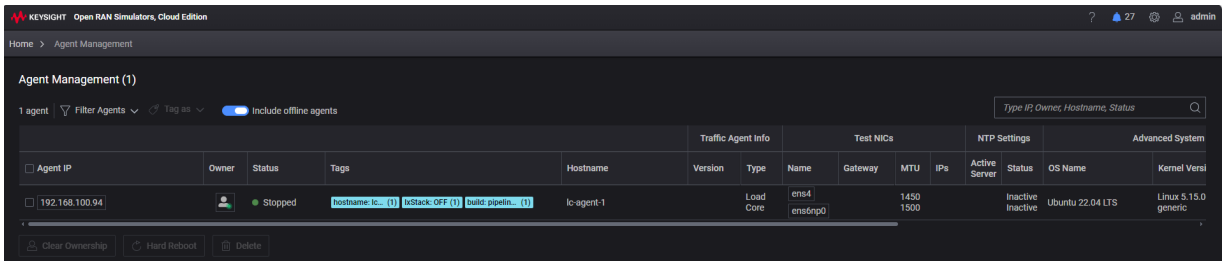
Enter the IP address or hostname of the middleware: 10.38.2.86

Selected interface ens3 with address 192.168.100.94 as management interface
Would you like to change the management interface? [y/n]: n

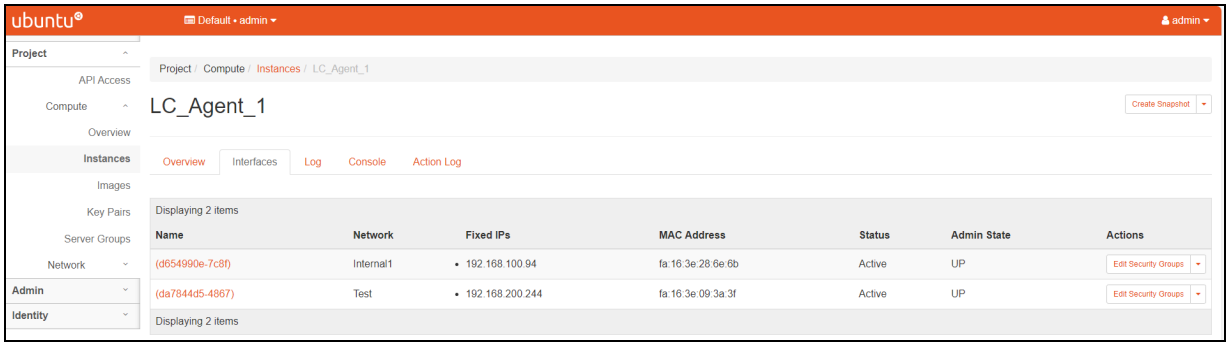
Do you want to allow this agent to be rebooted from the UI? [y/n]: y

Would you like to change the hostname to 5GCTE-fa-16-3e-28-6e-6b? [y/n]: n
```

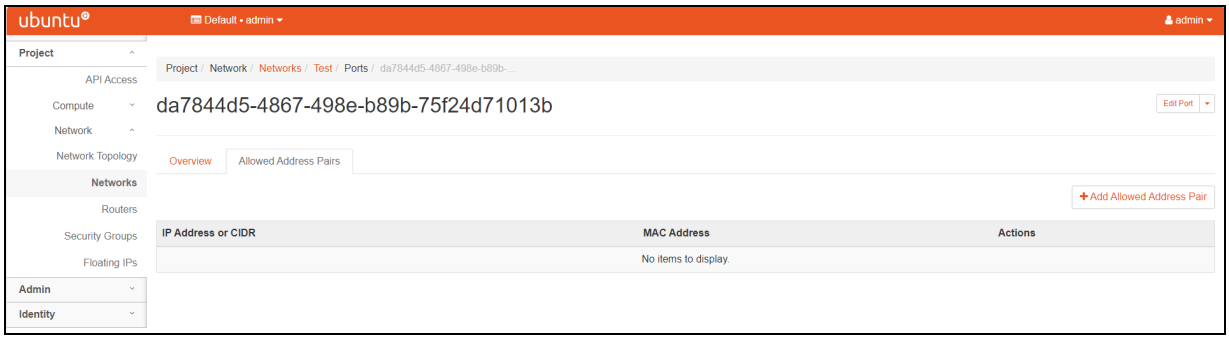
7. To make sure that this step was successful, check on the Middleware user interface: click on the **Settings** (⚙) button on the upper right corner, and select **Agent Management**, where the agent you just registered should be displayed.



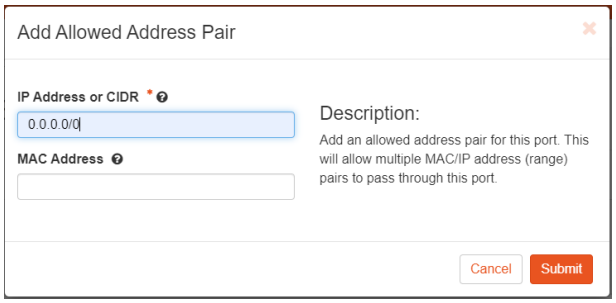
8. Once the Agent has been deployed, if a test subnet has been used, it is also required to allow traffic to flow through it. Back in OpenStack, go to **Project > Compute > Instances** and select the Agent instance, then go to the **Interfaces** tab.



9. Under the **Name** column, click the *Test* interface to open it for editing. You will be redirected to **Project > Network > Networks > *Test***.

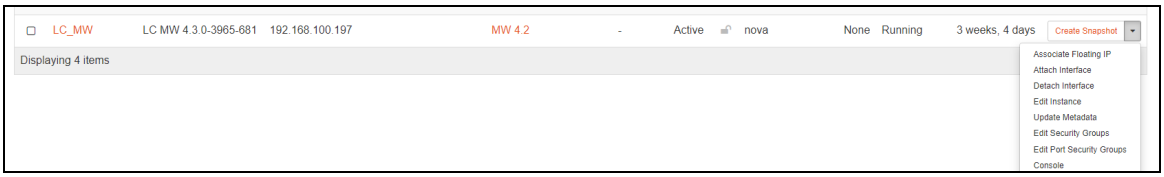


10. Select the **Allowed Address Pairs** tab, and press the **+Add Allowed Address Pair** button. In the dialog, set the IP address range on which traffic is allowed through the interface (20.0.0.0/8 is the default IP range used in LC tests, 0.0.0.0/0 will allow all traffic, or you can use another range), and click **Submit**.



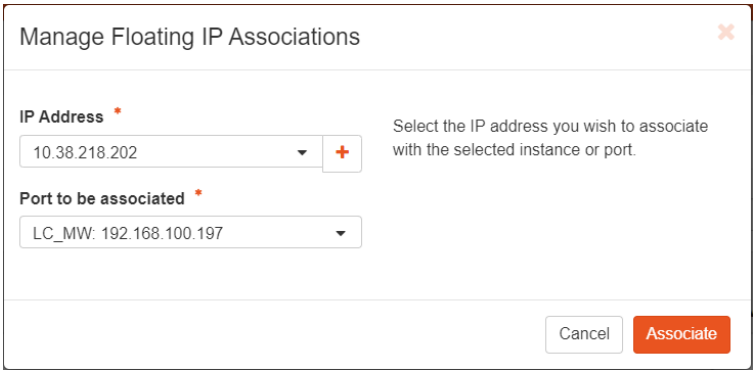
11. **Optional Step:** After the Agent instance has been deployed, assign a floating IP so you can access it from an external network:

a. from the **Project > Compute > Instances** section, go to the allocated image and, from the drop-down menu at the end of the row select **Associate Floating IP**.



b. in the **Manage Floating IP Association** dialog, select an *IP address* from the drop-down list, or create one by clicking the **+** button.

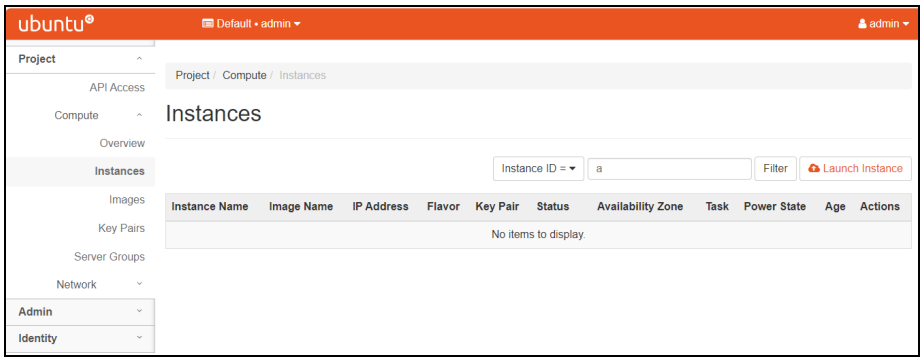
c. click **Associate** to conclude this step.



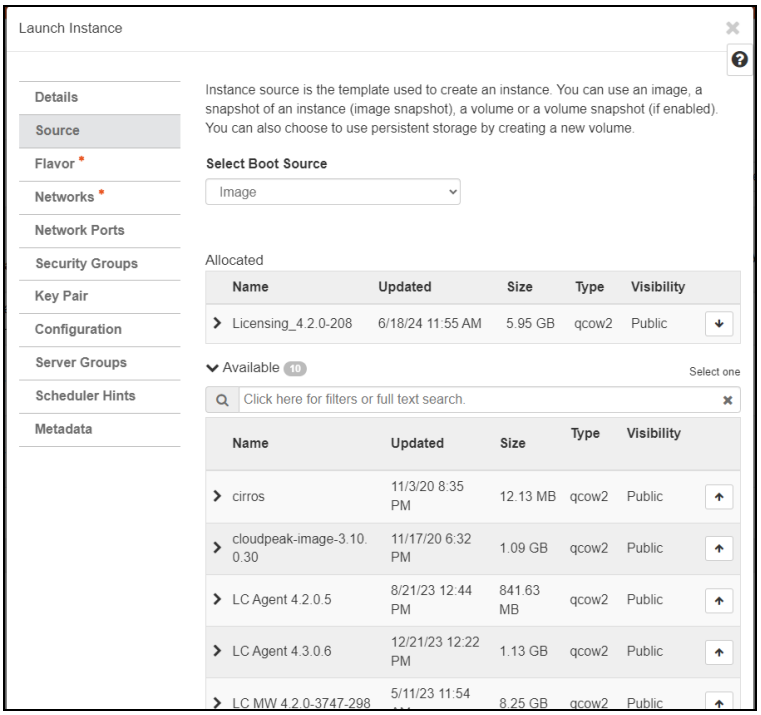
License Server installation

The installation procedure of the license server requires the following steps:

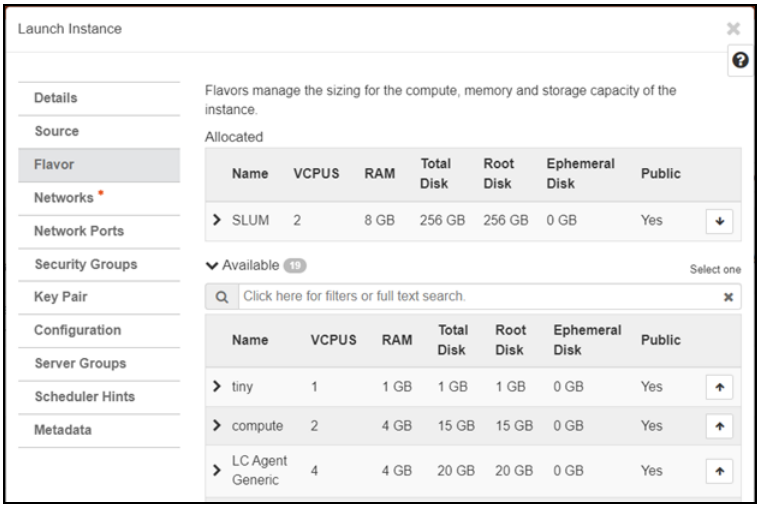
1. In OpenStack, navigate the left menu to **Admin > Compute > Images** section and select **Create Image**. Input an appropriate license server name, then browse and select the server image file, select the QCOW2 format and finalize by clicking on **Create Image** button.
2. Navigate to **Admin > Compute > Flavors** section and select **Create Flavor**. Input an appropriate flavor name, such as *SLUM*, and choose the required resources to be assigned, and press **Create Flavor** to confirm.
3. Navigate to **Project > Compute > Instances** section and click **Launch Instance**.



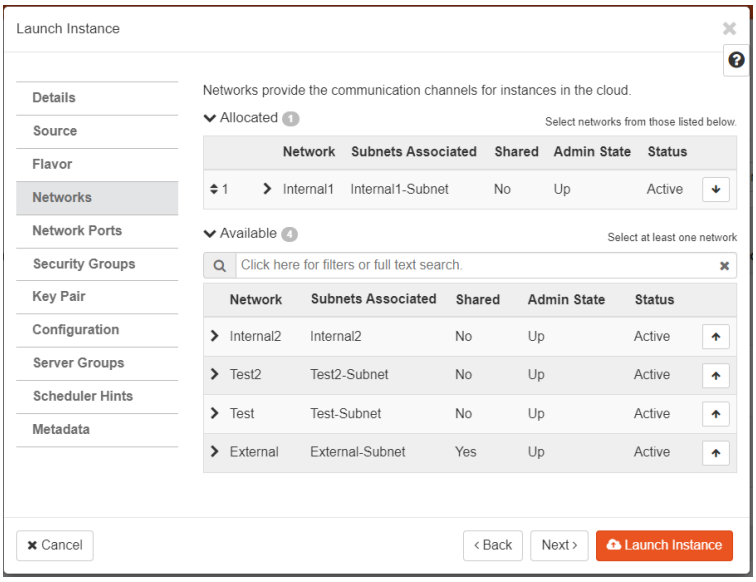
4. In the **Launch Instance** dialog (these are mandatory configurations marked by asterisk):
   - select **Details** section to input an appropriate *name* to this instance
   - select **Source** section and choose the newly created license server *image* as boot source



- select the **Flavor** section to add the newly created *flavor* that allocates the resources

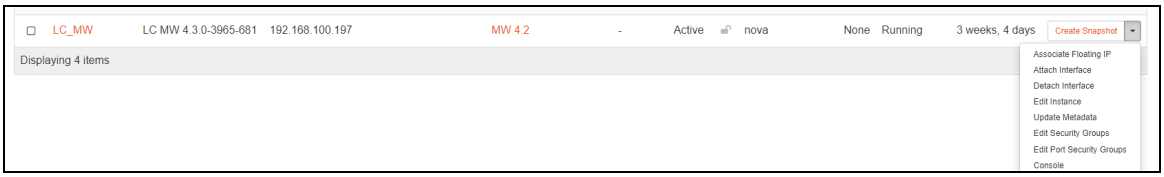


- select the **Network** section and choose the same management subnet that was previously added for MW and agent. This will be used for communication between License Server and MW

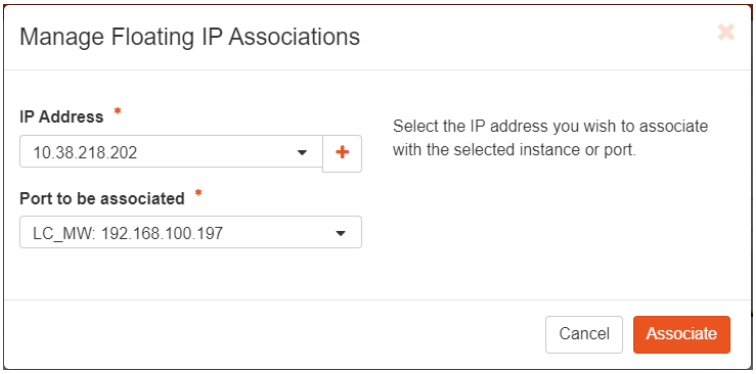


Confirm the configuration by pressing the **Launch Instance** button.

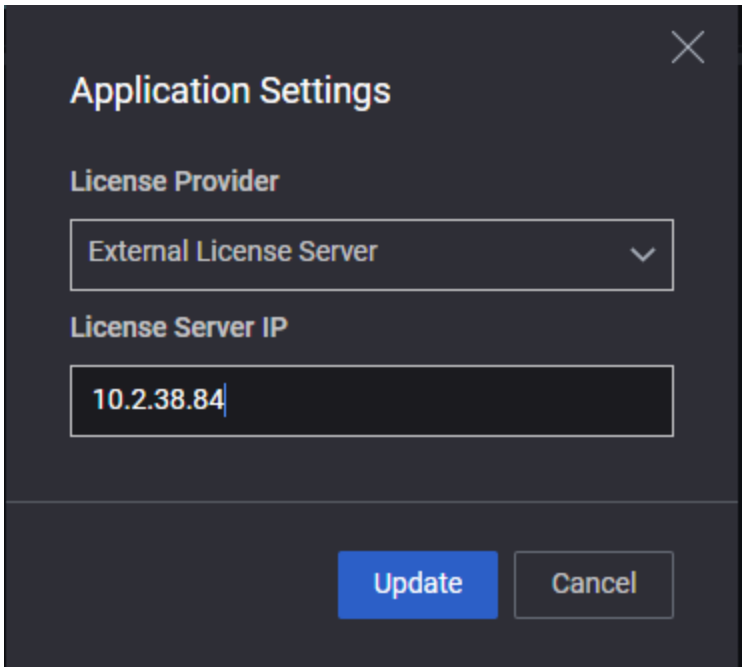
5. **Optional Step:**After the license server instance has been deployed, assign a floating IP so you can access it from an external network:
   a. from the **Project > Compute > Instances** section, go to the allocated image and, from the drop-down menu at the end of the row select **Associate Floating IP**.



   b. in the **Manage Floating IP Association** dialog, select an *IP address* from the drop-down list, or create one by clicking the **+** button.
   c. click **Associate** to conclude this step.



6. To finalize, go to the Middleware user interface, click on the **Settings** (⚙) button on the upper right corner, and select **Application Settings**. In the dialog, select **External License Server** and input the management IP (either internal or floating).



PCI Passthrough and CPU Pinning

PCI Passthrough and CPU pinning are used to obtain maximum performance out of the compute nodes' resources. These settings are recommended when simulating Application Traffic.

PCI Passthrough

1. Find the PCI of the interface that will be used for passthrough, vendorID and product ID. To do so, connect via SSH to the compute node, and issue the following command:

   ```
   lspci -nnn
   ```

   ```
   cristi@sa-os-compute1:~$ lspci -nnn | grep Mellanox
   d8:00.0 Ethernet controller [0200]: Mellanox Technologies MT27800 Family [ConnectX-5] [15b3:1017]
   d8:00.1 Ethernet controller [0200]: Mellanox Technologies MT27800 Family [ConnectX-5] [15b3:1017]
   cristi@sa-os-compute1:~$
   ```

2. With the information obtained, go to `/etc/nova/nova.conf` file and in the `[pci]` section add the device address to the `passthrough_whitelist`. Alternatively:
   - specify multiple PCI devices using `vendor_id` and `product_id`
   - add a PCI alias.

   For example:

   ```
   vi /etc/nova/nova.conf
   [pci]
   alias = { "vendor_id":"15b3", "product_id":"1017", "device_type":"type-PF",
   "name":"mlxcx5" passthrough_whitelist = [..,{"vendor_id":"15b3", "product_
   id":"1017"}] #OR passthrough_whitelist = {"address": "0000:d8:00.0"}
   ```

3. Restart the compute service with the following command:

   ```
   systemctl restart nova-compute.service
   ```

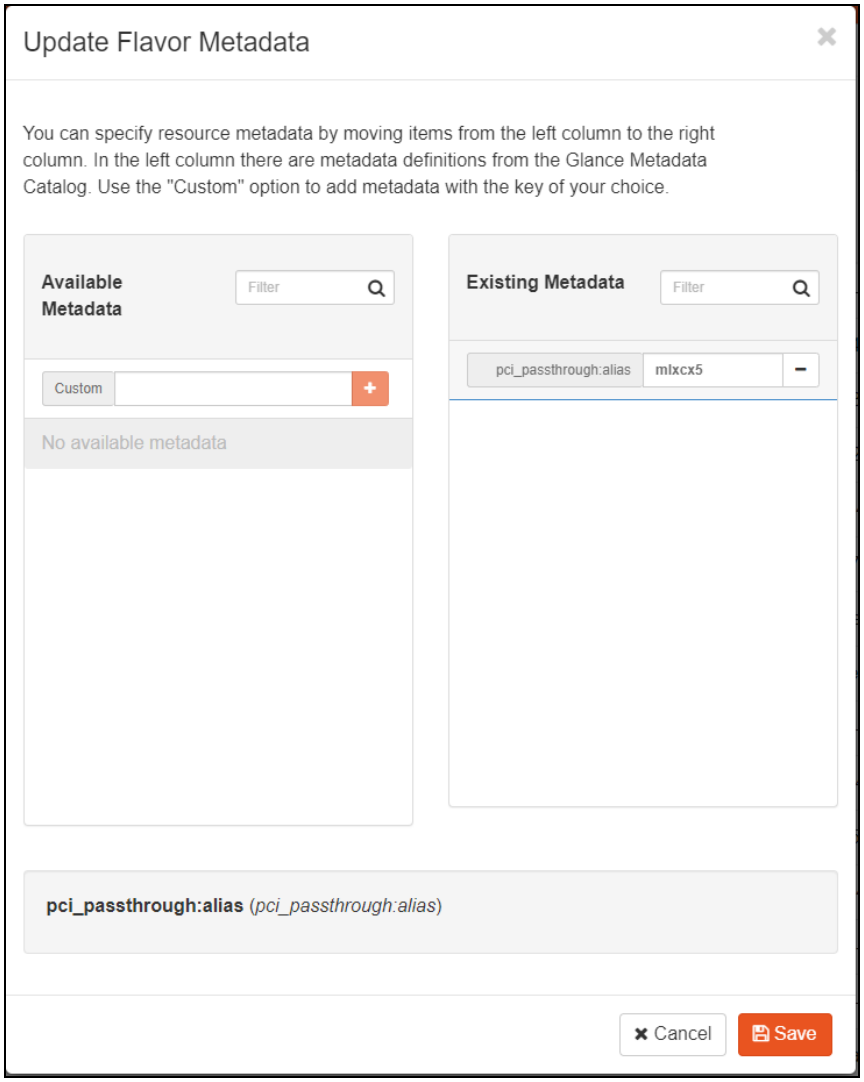
4. Connect via SSH to the controller node, and configure the same alias in the same file, under `[pci]`:

```
vi /etc/nova/nova.conf
[pci]
alias = { "vendor_id":"15b3", "product_id":"1017", "device_type":"type-PF",
"name":"mlxcx5" }
```

5. Restart the `nova-api.service` on the controller:

```
systemctl restart nova-api.service
```

6. In OpenStack, add the following metadata (property) to an existing flavor: go to the **Flavors** screen, click **Update Metadata**, and use `pci_passthrough:alias` and the alias from the earlier step:

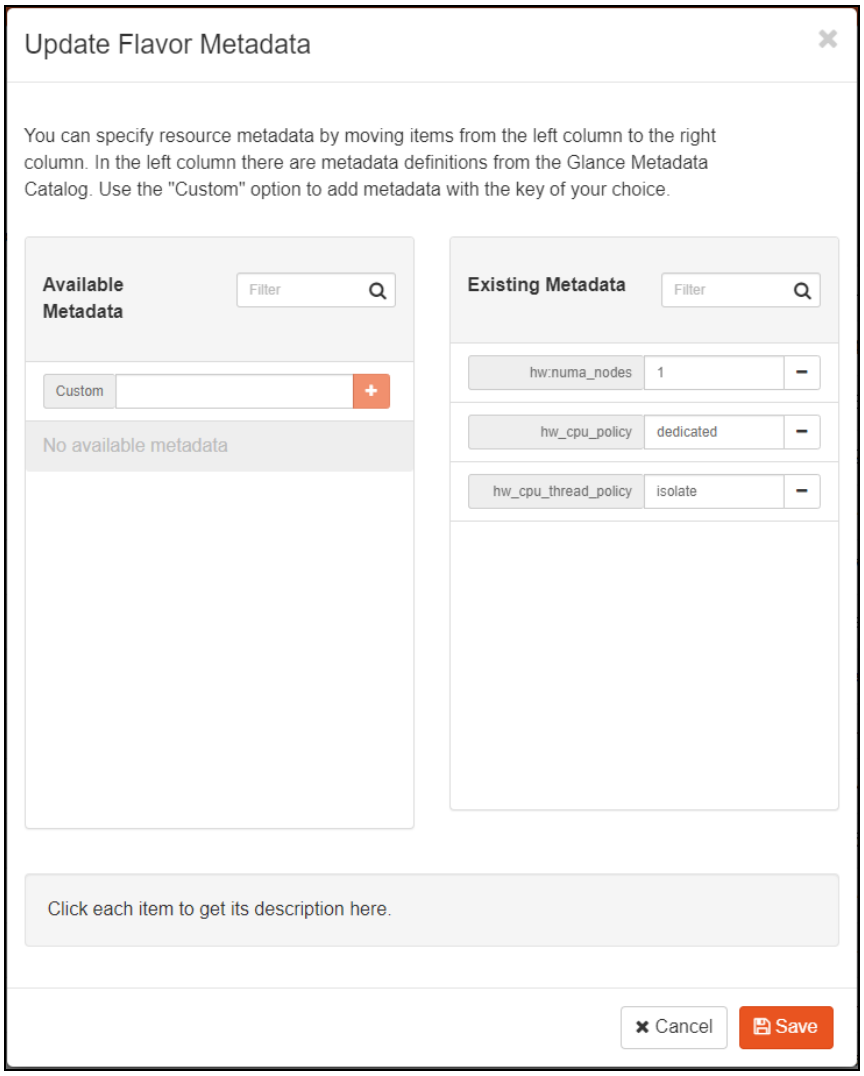


By using a flavor that is configured with direct PCI passthrough, the deployed instance will automatically have a test interface with PCI-PT.

CPU Pinning

In case the application traffic will be simulated, it is recommended to disable hyperthreading and enable CPU pinning on the agents:

1. Add the following metadata to the flavor, which means that the VM will have the hyperthreading disabled (thread policy), and that it will take the specified number of CPUs from one NUMA node.



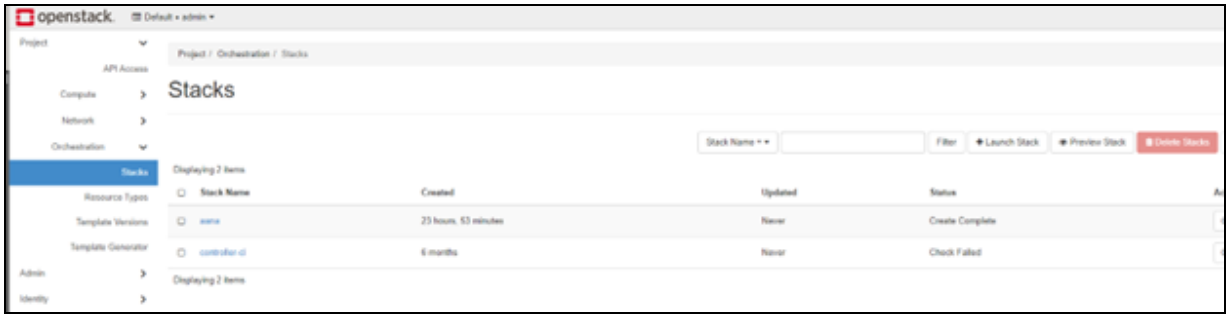
IMPORTANT Make sure that the number of CPUs set on the flavor is smaller than the number of CPUs in a NUMA node. This information can be verified via SSH connect to the compute node, issuing the `lscpu` command.

Note that, for optimal performance when simulating application traffic, the above metadata can also be combined with the PCI Passthrough.

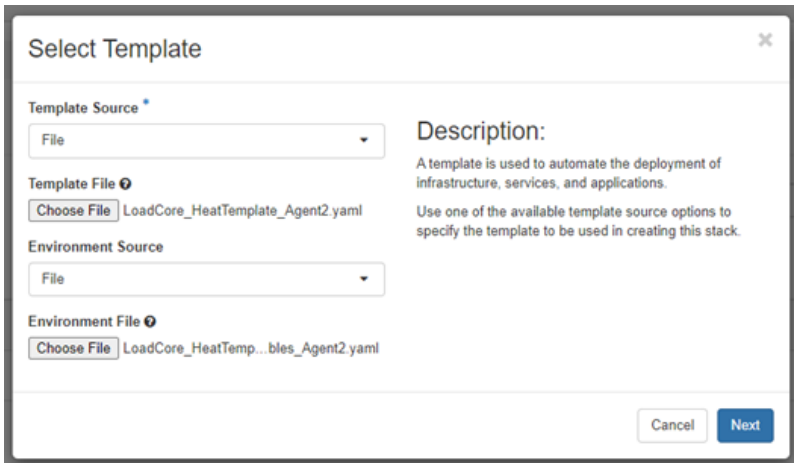
HEAT Templates

An alternative method to deploy a VM is by using a Heat template. This requires installation of OpenStack Orchestration service, and download the LC templates, found at the following link: [oran-sim-ce/Openstack/Open RAN SIM CE 2.0/Heat Templates at main · Keysight/oran-sim-ce · GitHub]

1. In OpenStack, go to **Project > Orchestration > Stacks**and select **+Launch Stacks**:

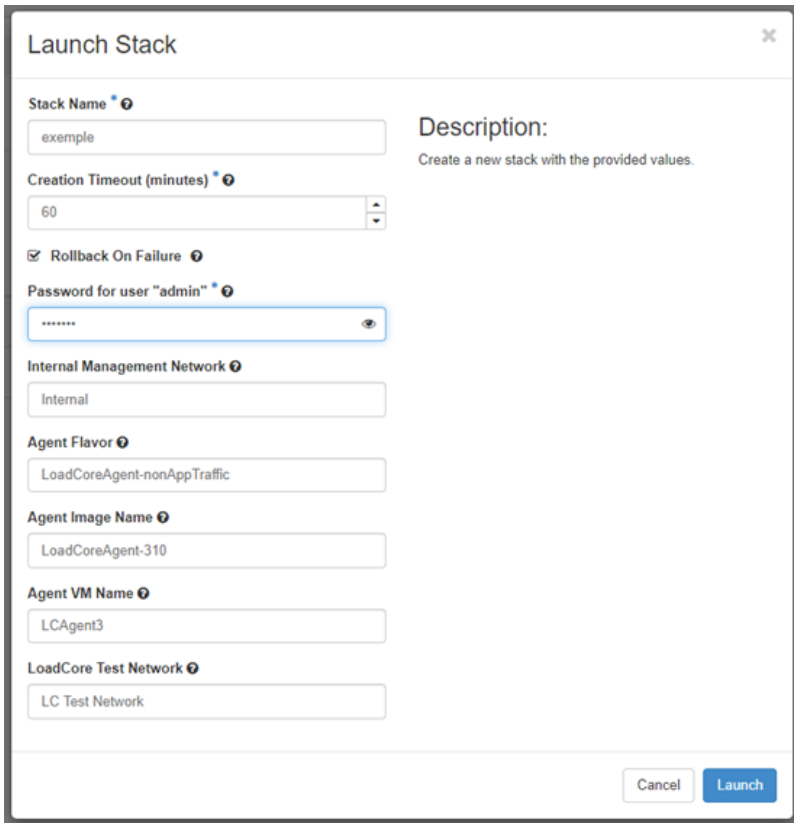


2. In the **Select Template** dialog, make sure to select the appropriate Heat template for the instance you want to deploy. Click **Next** to continue.



3. In **Launch Stack** dialog, select the parameters that will be used for the stack and instance

creation, such as network and VM name.



If the Orchestration menu is not present in the OpenStack UI, you can deploy with Heat templates from command line:

1. Upload the YAML files to the controller and then do a SSH connection.
2. Deploy the MW with a command such as:

```
openstack stack create -t LoadCore_HeatTemplate_MW1.yaml --parameter LoadCore_MW_
Name=LC_MW --parameter "LoadCore_MW_Image_Name=LC MW 4.3.0-3965-681" --parameter
"LoadCore_MW_Flavor=MW 4.2" --parameter Internal_Management_Network=Internal1
MWStack
```

3. Deploy the agent with a command such as:

```
openstack stack create -t LoadCore_HeatTemplate_Agent2.yaml --parameter LoadCore_
Agent_Name=LC_Agent_1 --parameter "LoadCore_Agent_Image_Name=LC Agent 4.3.0.6" --
parameter LoadCore_Agent_Flavor=flavor8cores32Gpinned --parameter Internal_
Management_Network=Internal1 --parameter Test_Network=Test AgentStack1
```

4. Check the existing stacks with:

```
openstack stack list
```

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 **powersafe** 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

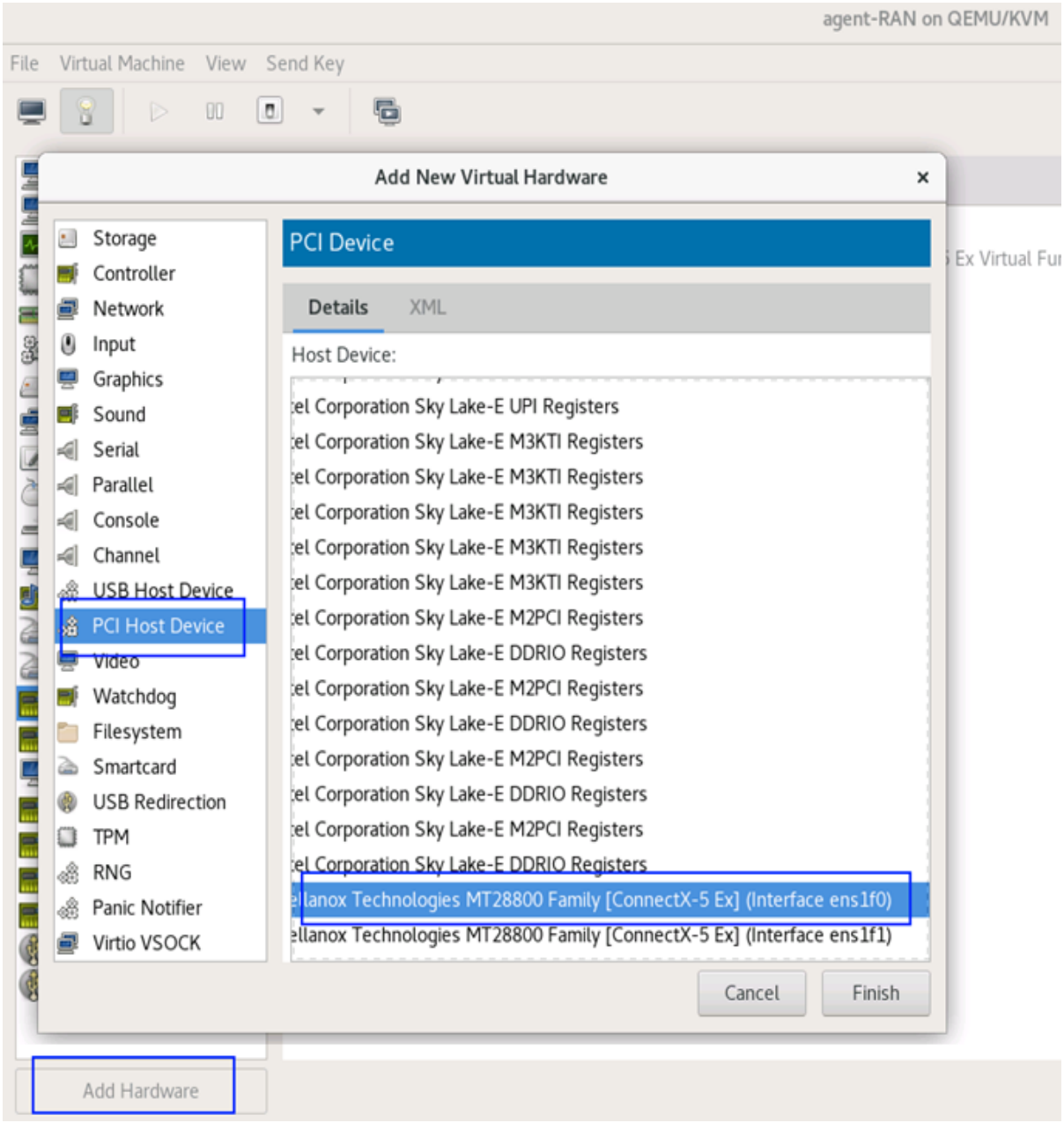
After following the above steps, the agent can now be used in Application Data scenarios..

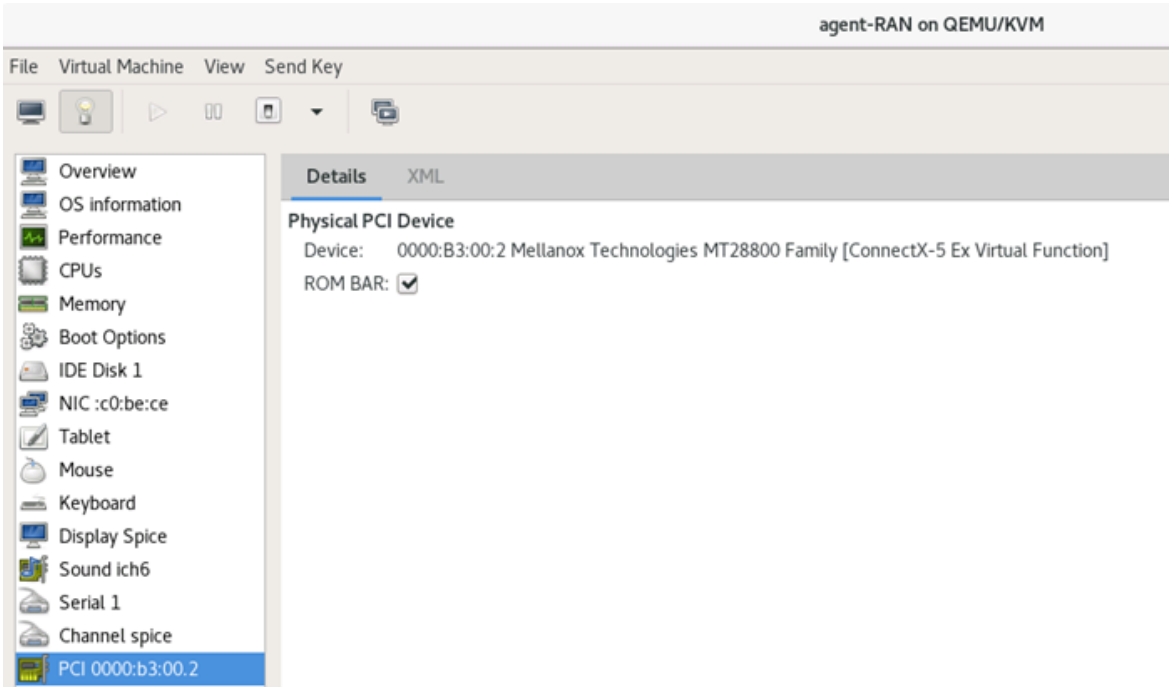
NOTE The maximum value for the huge pages number can be 8192. If the requirements surpasses this value, we recommend increasing the number of agents.

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.


agent-RAN on QEMU/KVM
File
Virtual Machine
View
Send Key
Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
Virtio VSOCK
PCI Device
Details
XML
Host Device:
el Corporation Sky Lake-E UPI Registers
el Corporation Sky Lake-E M3KTI Registers
el Corporation Sky Lake-E M3KTI Registers
el Corporation Sky Lake-E M3KTI Registers
el Corporation Sky Lake-E M3KTI Registers
el Corporation Sky Lake-E M3KTI Registers
el Corporation Sky Lake-E M2PCI Registers
el Corporation Sky Lake-E DDRIO Registers
el Corporation Sky Lake-E M2PCI Registers
el Corporation Sky Lake-E DDRIO Registers
el Corporation Sky Lake-E M2PCI Registers
el Corporation Sky Lake-E DDRIO Registers
el Corporation Sky Lake-E M2PCI Registers
el Corporation Sky Lake-E DDRIO Registers
lanox Technologies MT28800 Family [ConnectX-5 Ex] (Interface ens1f0)
llanox Technologies MT28800 Family [ConnectX-5 Ex] (Interface ens1f1)
Cancel
Finish
Add Hardware



To passthrough a PCI network card from the hypervisor CLI to the agent VM, first do the next command, which will show all the network cards, and their PCI information:

```
lshw -c network -businfo
```

Centos 2

```
[root@dhcp-10-38-154-175 ~]# lshw -c network -businfo
Bus info          Device      Class          Description
========================================================
pci@0000:18:00.0  eno1        network        I350 Gigabit Network Connection
pci@0000:18:00.1  eno2        network        I350 Gigabit Network Connection
pci@0000:18:00.2  eno3        network        I350 Gigabit Network Connection
pci@0000:18:00.3  eno4        network        I350 Gigabit Network Connection
pci@0000:3c:00.0  enp60s0f0   network        Ethernet Controller XL710 for 40GbE QSFP+
pci@0000:3c:00.1  enp60s0f1   network        Ethernet Controller XL710 for 40GbE QSFP+
pci@0000:5e:00.0  enp94s0f0   network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:5e:00.1  enp94s0f1   network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:af:00.0  enp175s0f0  network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:af:00.1  enp175s0f1  network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:d8:00.0  enp216s0f0  network        Ethernet Controller XL710 for 40GbE QSFP+
pci@0000:d8:00.1  enp216s0f1  network        Ethernet Controller XL710 for 40GbE QSFP+
                  virbr1-nic  network        Ethernet interface
                  virbr0-nic  network        Ethernet interface
                  virbr0      network        Ethernet interface
                  bridge1     network        Ethernet interface
                  bridge2     network        Ethernet interface
                  bridge0     network        Ethernet interface
                  virbr1      network        Ethernet interface
```

From this output choose the card that you want, for example device *enp60s0f1*, which is *Intel XL710* port and add it to the agent:

```
virsh attach-device LC_Agent pci_0000_3c_00_1.xml
```

The *.xml* of the PCI can also be checked with :

```
virsh nodedev-dumpxml pci_0000_3c_00_1
```

Alternatively, you can also edit the *.xml* file of the VM:

```
virsh edit LC_Agent
```

Add the lines in the `<devices>` section:

```
<hostdev mode='subsystem' type='pci' managed='yes'>
 <source>
  <address domain='0x000' bus='0x3c' slot='0x00' function='0x1'/>
 </source>
</hostdev>
```

Appendix B Deploying Open RAN SIM CE 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 mgmt0 10.38.50.100/24 10.38.50.1
```

`mgmt0` 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 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 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.

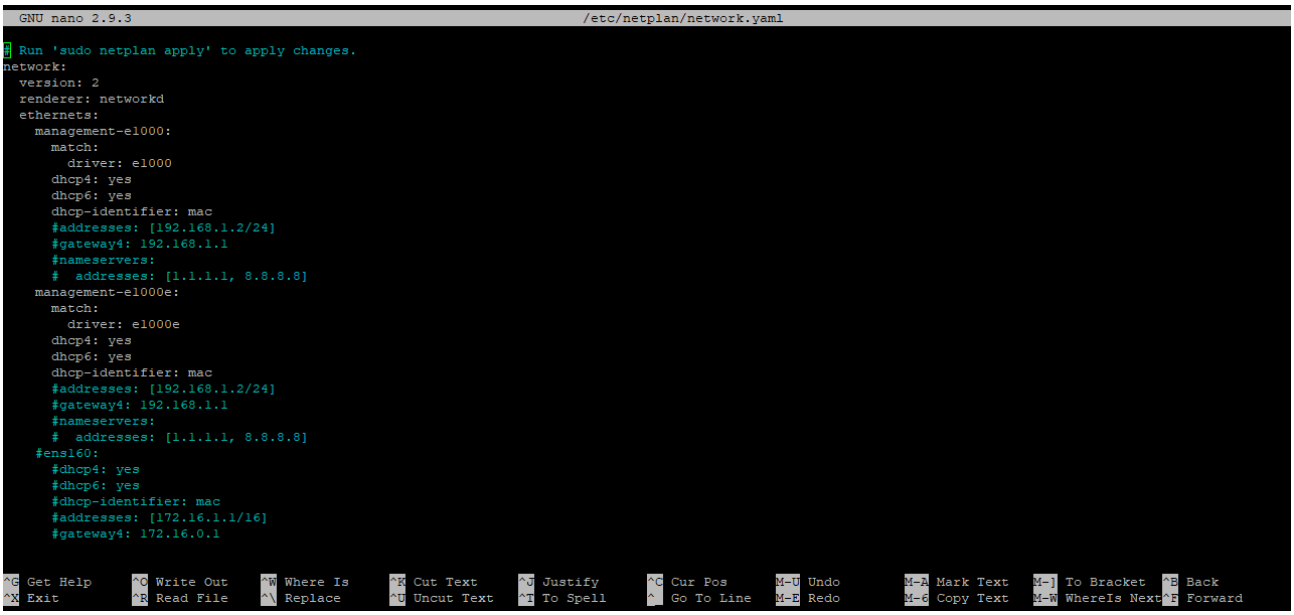
Agents

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).

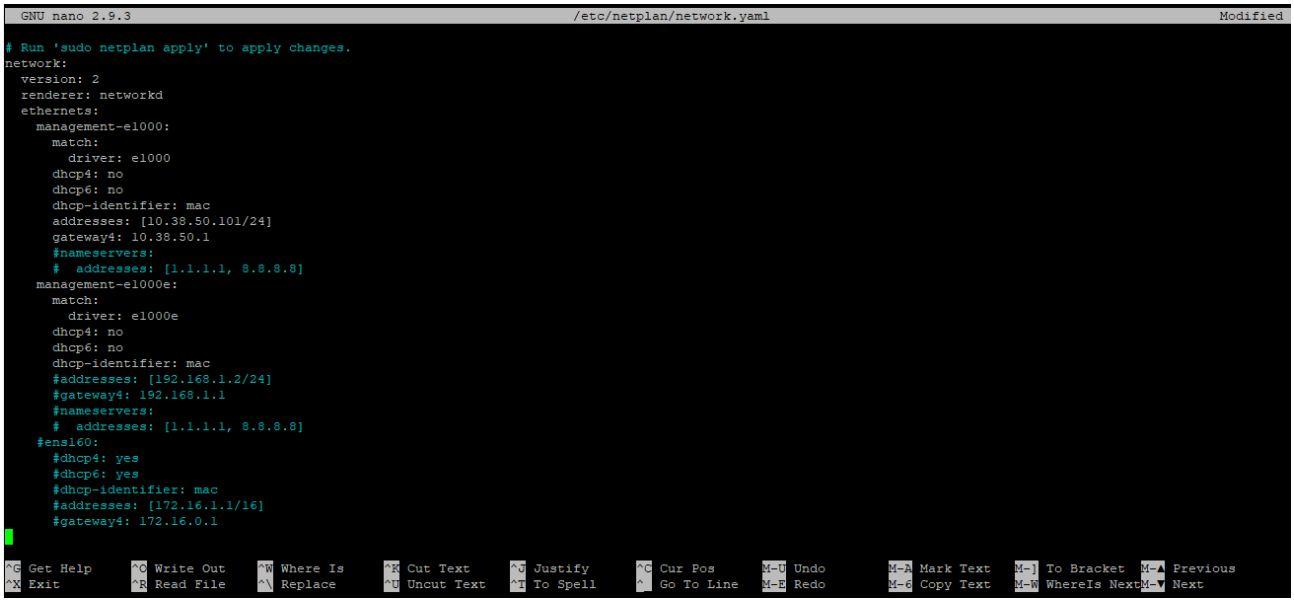
The IP can be changed by modifying the `netplan`:

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

By default, it will display the following:



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



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.

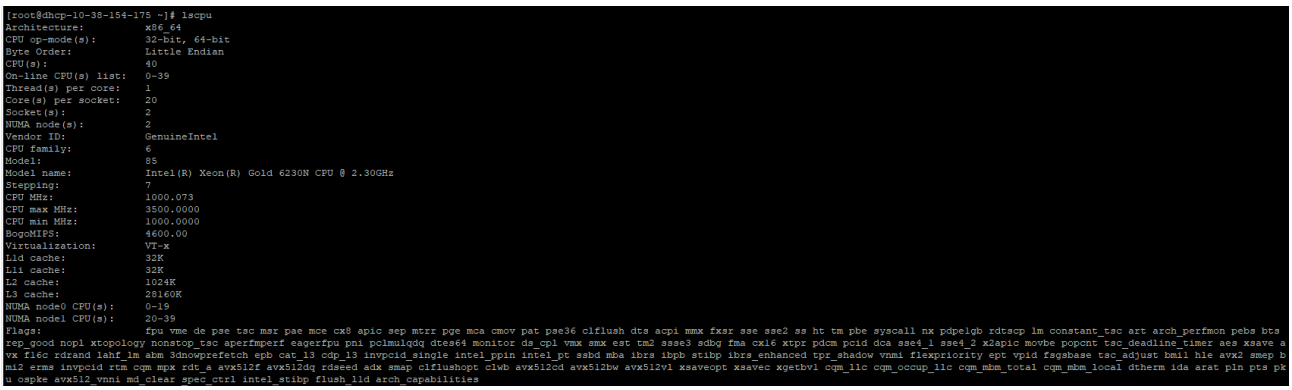
Appendix C Ports used in ORAN-SIM CE communication

The following are ports used by Open RAN SIM CE 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 MWand agents
443	ingress, for rest api commands/to connect to webserver
4222	NATS communication between MW and agents
7443	communication between MW and license server
9022	SSH to connect as root
31123	per-UE stats communication between MW and agents
32100	EULA (EULA can also be accepted through console)

Appendix D CPU Pinning

To make sure that CPU pinning is done correctly, first the number of NUMA nodes has to be checked. Also it is recommended to have hyperthreading disabled. This can be done with `lscpu`, and in the example below there are 2 NUMA nodes and 1 Thread per core (hyperthreading disabled).



It is best that all the CPUs that will be pinned on one VM belong to the same NUMA node. In the case above the 1st NUMA node has CPUs 0-19 and the 2nd NUMA node has CPUs 20-39.

IMPORTANT Host aggregates should be used to separate pinned instances from unpinned instances as the latter will not respect the resourcing requirements of the former. This means that, for maximum performance and vCPU separation, all VMs will have CPU pinning, including MW, license server, etc.

Once it has been established which CPUs will be assigned to the VM, it can be done with the `virsh vcpupin` command. In the example below a `for` loop has been used to assign the first 8 CPUs:

```
for i in {0..7}; do virsh vcpupin --config LC_Agent $i $i; done
```

After this has been done it can be checked from the XML of the VM, and it will look like:

```
<vcpu placement='static'>8</vcpu>
 <cputune>
  <vcpupin vcpu='0' cpuset='0'/>
  <vcpupin vcpu='1' cpuset='1'/>
  <vcpupin vcpu='2' cpuset='2'/>
  <vcpupin vcpu='3' cpuset='3'/>
  <vcpupin vcpu='4' cpuset='4'/>
  <vcpupin vcpu='5' cpuset='5'/>
  <vcpupin vcpu='6' cpuset='6'/>
  <vcpupin vcpu='7' cpuset='7'/>
 </cputune>
```

On the 2nd agent the next 8 CPUs can be assigned, as follows:

```
for i in {8..15}; do virsh vcpupin --config LC_Agent_2 $(($i - 8)) $i; done
```

And it will look like:

```
<vcpu placement='static'>8</vcpu>
 <cputune>
  <vcpupin vcpu='0' cpuset='8'/>
  <vcpupin vcpu='1' cpuset='9'/>
  <vcpupin vcpu='2' cpuset='10'/>
  <vcpupin vcpu='3' cpuset='11'/>
  <vcpupin vcpu='4' cpuset='12'/>
  <vcpupin vcpu='5' cpuset='13'/>
  <vcpupin vcpu='6' cpuset='14'/>
  <vcpupin vcpu='7' cpuset='15'/>
 </cputune>
```

The XML can also be changed directly.

To revert the CPUs that were pinned to a certain VM, either delete the lines from the XML file, or simply assign access to all the CPUs, in this case 0-39:

```
for i in {0..7}; do virsh vcpupin --config LC_Agent $i 0-39; done
```

Appendix E Configure SRI-OV with Network Manager

1. After you have chosen an appropriate interface to use for single-root input/output virtualization (SRI-OV), you can configure the number of VFs that will be created from it (in this case four of them):

   ```
   sudo nmcli con modify ens1f0np0 sriov.total-vfs 4 sriov.autoprobe-drivers true
   ```

2. Modify the the VFs and add a different MAC address to each:

   ```
   sudo nmcli con modify ens1f1np1 sriov.vfs '0 mac=01:64:02:00:05:01 trust=true, 1
   mac=01:64:02:01:06:02 trust=true, 2 mac=01:64:02:02:07:03 trust=true, 3
   mac=01:64:02:03:08:04 trust=true'
   ```

3. Do a rebbot.

4. To check that the setting has taken effect, use the following command:

   ```
   sudo nmcli con show ens1f0np0 | grep -i sriov
   ```

5. The output should look as follows:

   ```
   sriov.total-vfs: 4
   sriov.vfs: 0 mac=01:64:02:00:05:01 trust=true, 1 mac=01:64:02:01:06:02 trust=true, 2
   mac=01:64:02:02:07:03 trust=true, 3 mac=01:64:02:03:08:04 trust=true
   sriov.autoprobe-drivers: 1 (true)
   ```

6. Check if the VFs appear and their PCI number using the following command:

   ```
   sudo lshw -c network -businfo
   ```

   You should now be able to use passthrough and assign the VFs to the agent VMs.

NOTE

It is possible to experience the VFs to appear fluctuating in the output of the nmcli connection. This occurs because the VFs do not have an IP connection (DHCP is enabled by default) or because the port on which they are created does not have a physical connection yet. This situation can be ignored since it is only important to use the VFs with PCI passthrough on the VMs, and assign the IPs during a test.

This page intentionally left blank.

CHAPTER 3

ORAN-SIM CE Deployment on Containers

This section contains the following topics:

Amazon AWS EKS Deployment **120**
How to get an authentication token to be able to push the images into ECR 120
How to push ORAN-SIM CE container images into AWS EKS 121
How to install ORAN-SIM CE Middleware in EKS 121
How to install ORAN-SIM CE Agents on EKS 122
How to uninstall ORAN-SIM CE components 123
OpenShift Deployment **123**
Supported NICs in tests with IxStack over DPDK/Raw(TCP based traffic): 125
ORAN-SIM CE Deployment on k8s **125**
ORAN-SIM CE Deployment on GKE Anthos **128**

Amazon AWS EKS Deployment

This section describes the steps need for ORAN-SIM CE deployment in AWS EKS.

The deployment procedure requires a ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE.

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 ORAN-SIM CE 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 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 ORAN-SIM CE 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 ORAN-SIM CE Agents on EKS

To run tests in ORAN-SIM CE , 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 ORAN-SIM CE Agents:

- Install `multus CNI` using the following command:
  ```
  kubectl apply -f 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": "l2",
        "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 replica 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 ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE 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 ORAN-SIM CE image using the same image ID returned by `podman images`:
```
%> podman tag ca0aa01b2c11 $REGISTRY/loadcore-root/loadcore-agent:0.1-15f7cb3ba
```
11. Push the ORAN-SIM CE 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
    rawCNIConfig: '{"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 ORAN-SIM CE 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 ORAN-SIM CE agent deployment:

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

16. Take a look at the started ORAN-SIM CE pods:

```
%> oc get pods
```

17. Get a shell inside a ORAN-SIM CEagent pod:

```
%> oc exec -it loadcore-agent-65bdc89cfc-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.

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

- Intel X710 10G
- Intel XXV710 25G
- Intel XL710 40G
- Intel E810 25G
- Intel E810 100G
- Mellanox ConnectX-4/5 25G
- Mellanox ConnectX-4/5 100G
- Mellanox ConnectX-6 Dx 100G

ORAN-SIM CE Deployment on k8s

The following procedure describes the steps needed in order to deploy ORAN-SIM CE 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.yml
   ```

   NOTE

   Alternatively, you can upload and install both the kubernetes cluster along with `multus` with the `kube-install.py` script; replace the user and the interface with the corresponding ones from the setup:

   ```
   sudo python3 kubeinstall.py --setup-system --user=ixia --net-interface=ens3 --config=config.json
   ```

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 ORAN-SIM CE.

      NOTE

      By default, `load-core.yaml` comes as a *LoadBalancer*. This means that your ORAN-SIM CE application will be reachable at port **443**. However, it can be easier to configure it as *NodePort*, and the application will be later accesible on port **30443**.

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.

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

   If you do not have a private registry, you will have to change the status (*true* to *false*) in `load-core.yaml`, `storePrivateImages` and `storePublicImages` as in the following example:

   ```
   privateRegistry:
     name: <REGISTRY-URL>
     storePrivateImages: false
     storePublicImages: false
     auth:
       user: <REGISTRY-USER>
       password: <REGISTRY-PASSWORD>
     pullSecret:
       name: load-core-secret
       create: true
     pushTag: ""
   ```

5. In `load-core.yaml` you need to specify the storage class. If `kubeinstall.py` script was used, by default will be `local-path`:

   ```
   persistence:
       storage:
         # It is mandatory to specify a storage class to be used by the volumes
   provisioner
   ```

```
      # (for storing data in the cluster).
      # Also, the size of the provisioned volumes can be tweaked.
      class: local-path
      volumesCapacity:
        testResults: 64Gi
        elasticsearch: 16Gi
        postgresql: 8Gi
```

You can also query it with the following command:

```
kubectl get storageclass -A
```

6. In case there is no specific worker node to be used for licensing, the appropriate line can be uncommented in `load-core.yaml`:

```
# worker: <WORKER-NAME>
    # Alternatively, the worker can be omitted and a random worker may be used.
     useRandomWorker: true
```

7. If you want to change the admin password on the initial log-in, uncomment and set the following variable to in `load-core.yaml` if you want to change the admin password on initial login:

```
updateAdminPassword: true
```

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

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

9. Before deploying the agent, you need to use an appropriate interface. It is recommended to use one that is separate from management. Default is set to be the "*whereabouts*" type, but it can be changed to `ipvlan` as in the following example from `load-core.yaml` (replace interface and IPs as appropriate):

```
config: |
        {
         "cniVersion": "0.3.1",
         "type": "ipvlan",
         "master": "ens8",
         "mode": "l2",
         "ipam":
               {
               "type": "host-local",
               "subnet": "20.0.0.0/8",
               "rangeStart": "20.0.0.2",
               "rangeEnd": "20.255.255.254"
               }
         }
```

10. 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 two agents will be deployed. If you need more agents, you will need to change this value.

11. In order to uninstall the agent, use the following command:
```
./loadcore-kube-setup uninstall-agent
```
12. In order to uninstall ORAN-SIM CE , use the following command:
```
./loadcore-kube-setup uninstall-mw
```
IMPORTANT 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 middleware.

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

ORAN-SIM CE Deployment on GKE Anthos

The following procedure describes the steps needed in order to deploy ORAN-SIM CE 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
```
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 ORAN-SIM CE `tar.gz` image.
5. Connect to Cloud Shell and download the ORAN-SIM CE `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 ORAN-SIM CE 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 ORAN-SIM CE images using:

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

d. In the ~ directory, create a dir for your ORAN-SIM CE . 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 (as per https://cloud.google.com/iam/docs/creating-managing-service-account-keys#iam-service-account-keys-create-console):

```
gcloud iam service-accounts keys create our_key --iam-account="your iam-account"
```

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 ORAN-SIM CE IP (which will be the ingress ip) you have to:

```
kubectl get svc -A -o wide | grep kcos-framework-v1-ingress ->
```

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 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 middleware.

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`

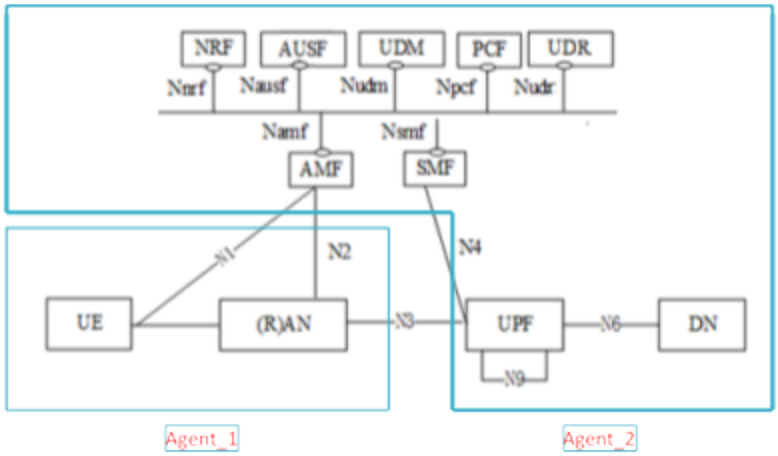
NOTE Anthos cluster already has LoadBalancer installed. This means that you will need to set in `load-core.yaml` at type of the cluster **LoadBalancer**.

This page intentionally left blank.

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

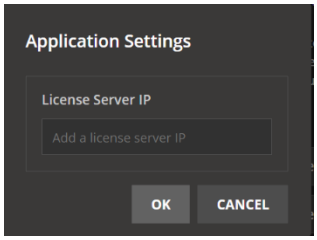
Two options are available for configuring the first LoadCore test:

- 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.

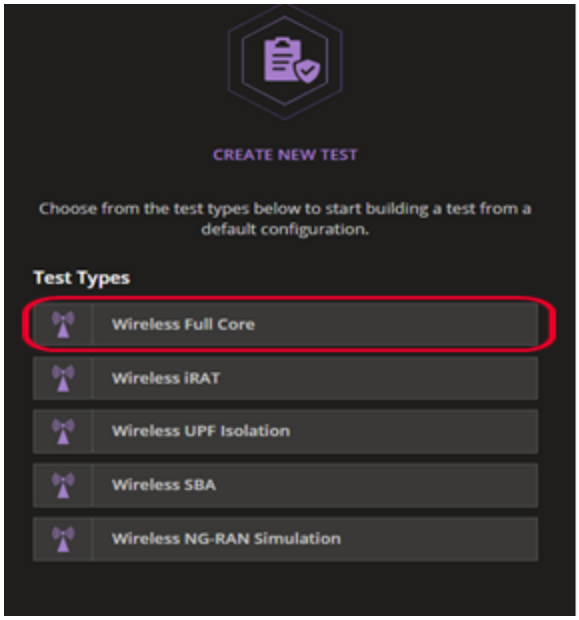


Is this setup the license server IP is: 10.73.53.17 (the IP address the license server got during the VM deployment).

Configure the test from scratch

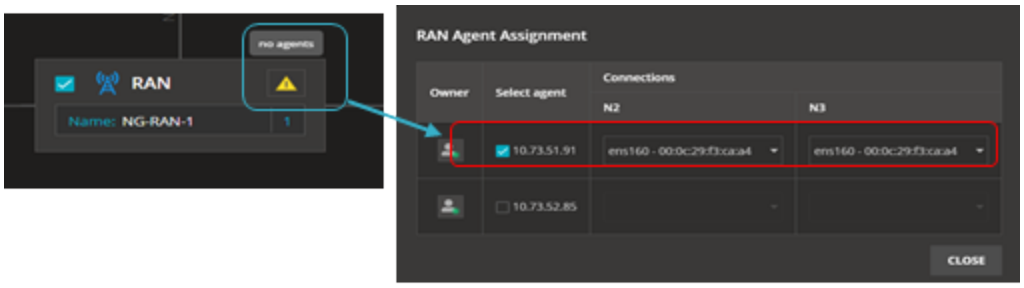
To configure the test using this method, do the following:

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**.

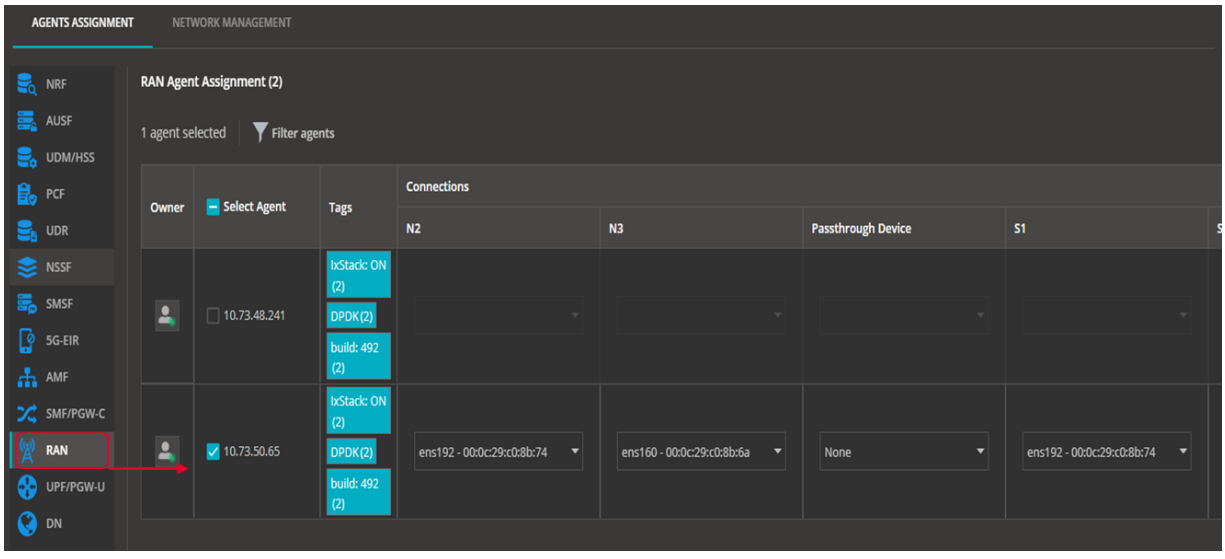


This creates a new session with Full Core topology, that is already configured with some parameters. The IP addresses assigned to the nodes must be changed in order to match the VM deployment.

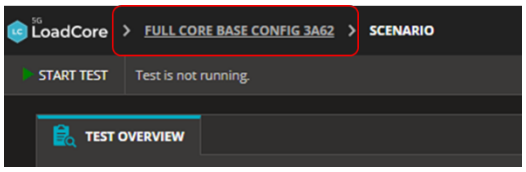
3. Select the yellow warning sign on the nodes and assign an agent to simulate it. In the drop-down list, there are the IP address of the agents that were previously defined:



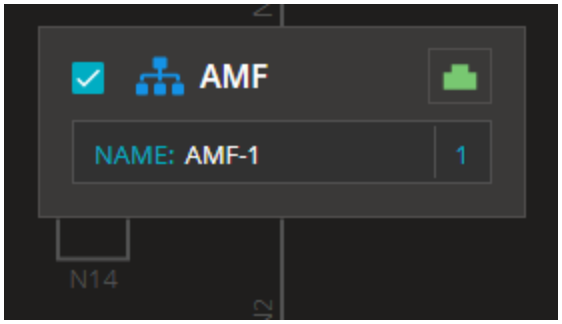
- **Agent 1** has: 10.73.50.65 and will be assigned for emulating the gNB
- **Agent 2** has: : 10.73.48.241 and will be assigned to the rest of the nodes



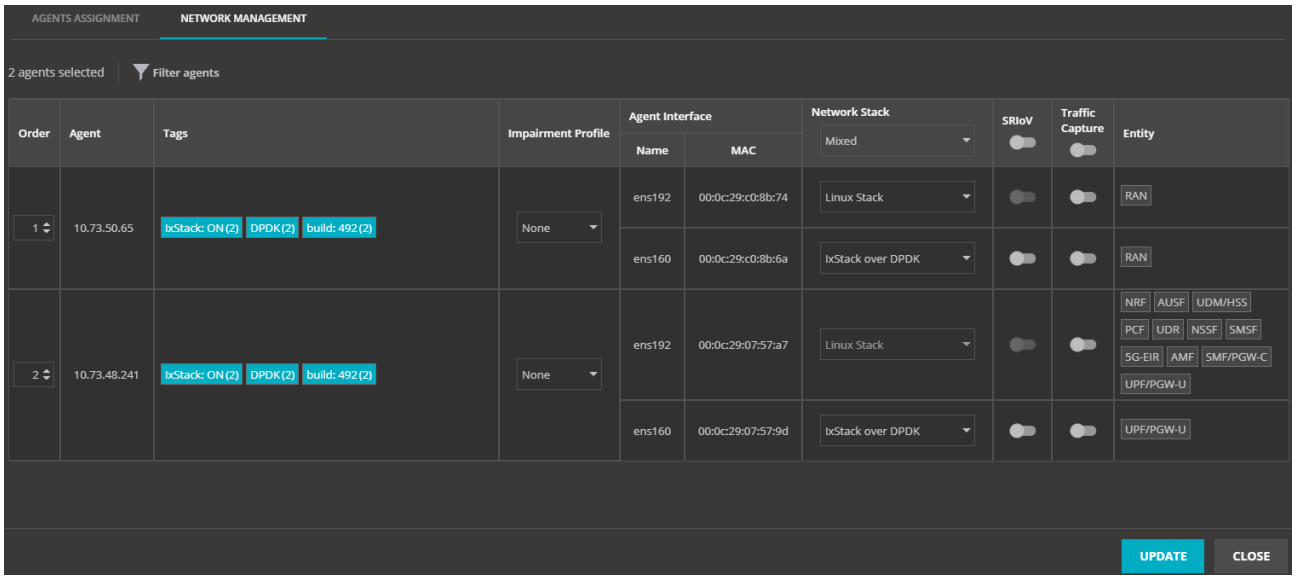
IMPORTANT It is recommended to separate CP and UP interfaces between the agent interfaces. For example, N2 or S1-MME on ens160, and N3 or S1-u on ens192.



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.



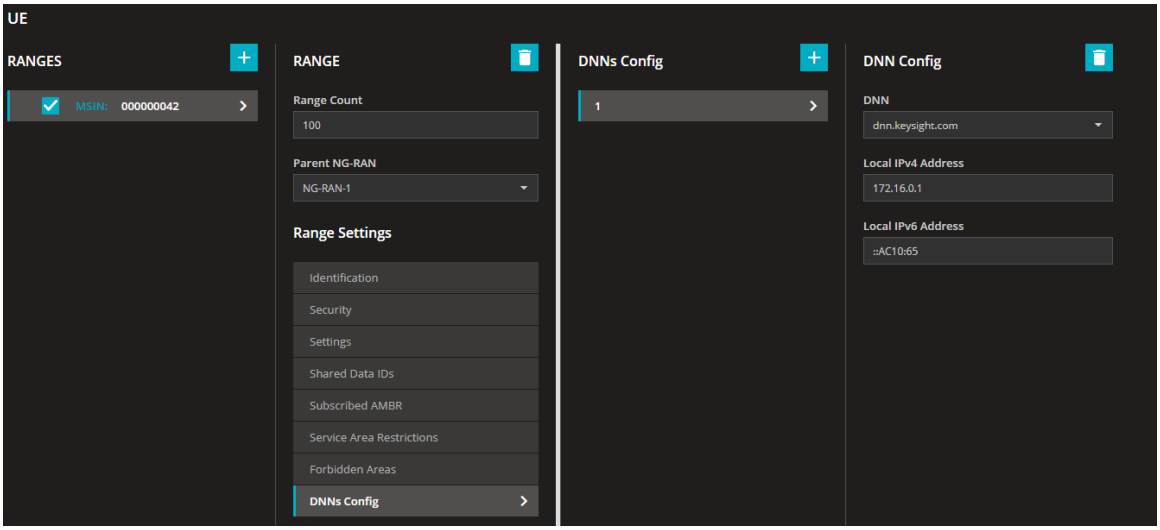
From here you can also enable traffic capture or configure different stacks to be used on a specific test interface (for example, application traffic can run only over IxStack Raw Sockets or IxStack over DPDK/Raw).

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.

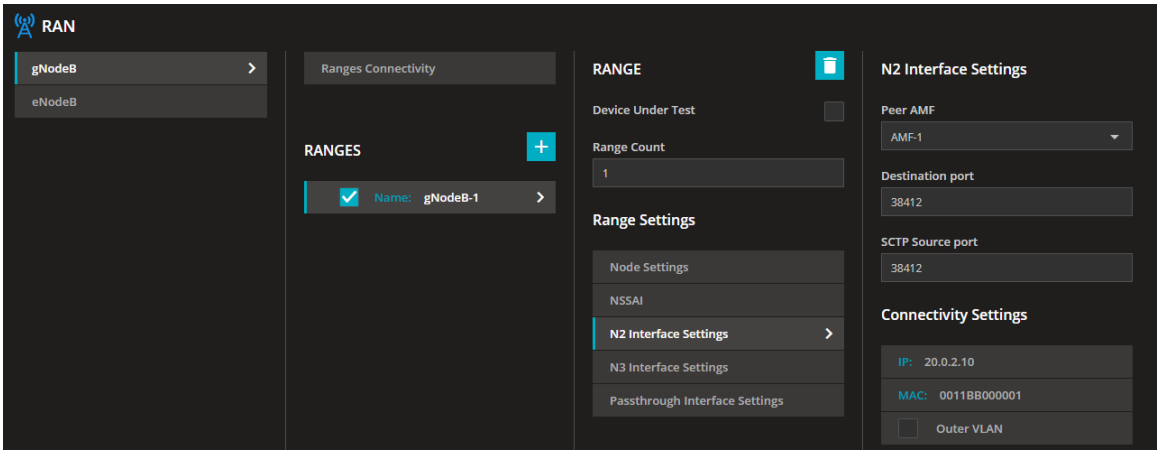
When the test starts, the LoadCore application will configure these IP addresses on the deployed nodes (no need for manual configuration with `yaml` file, like in the previous LoadCore release).

The IP allocation is done as follows:

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

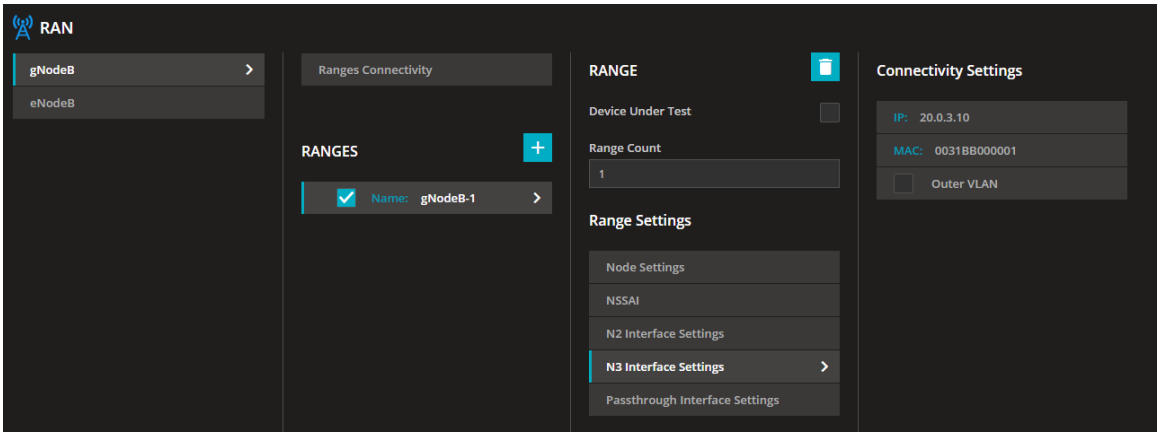


- **RAN configuration**:
  - N2 Interface settings
    - IP address: **20.0.2.10**
    - Peer AMF: **AMF-1**

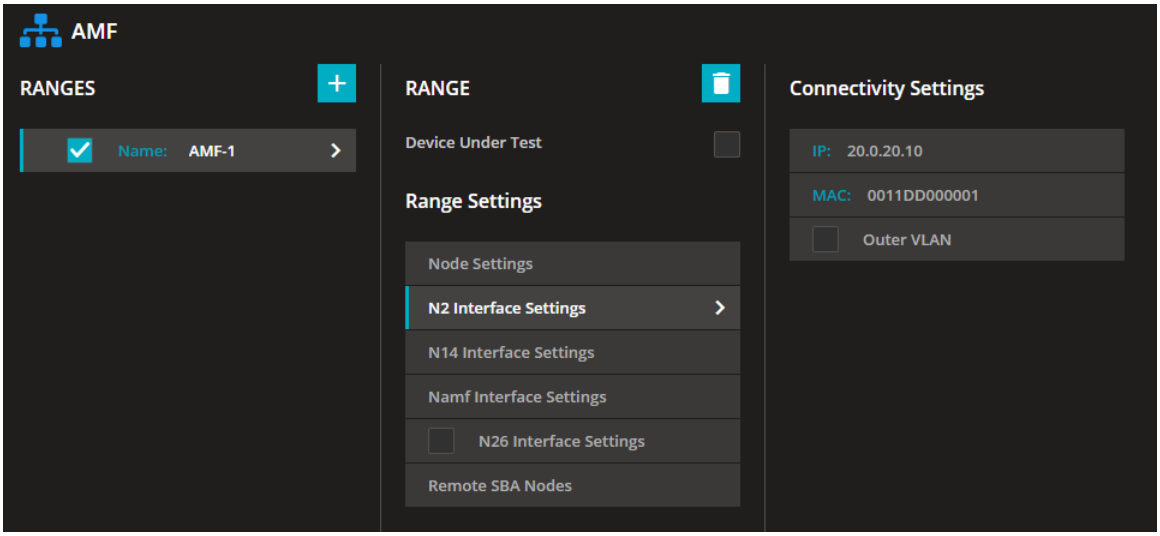


  - N3 Interface settings:

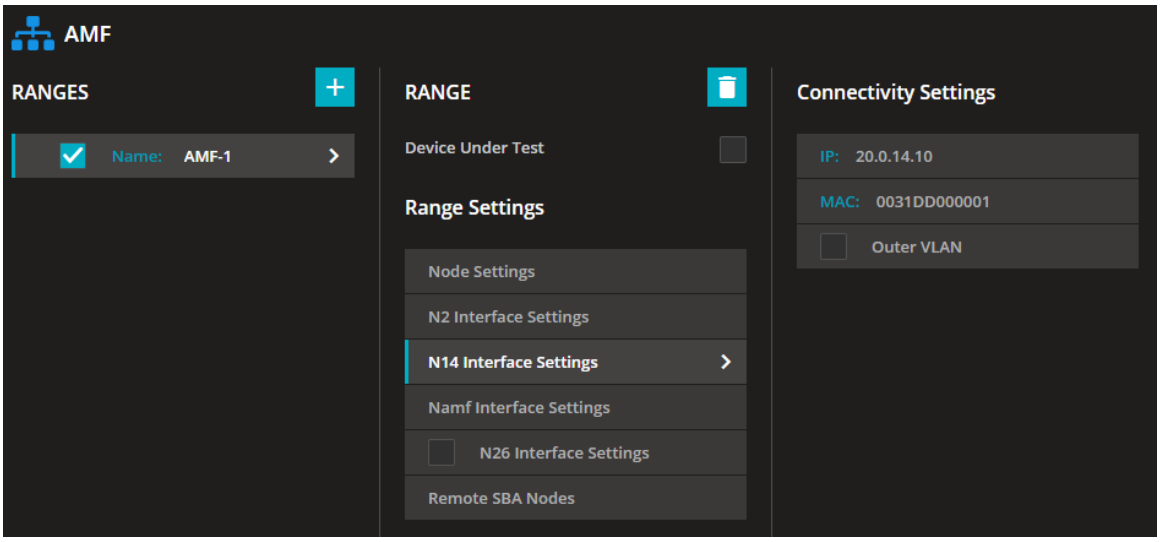
- IP address: **20.0.3.10**



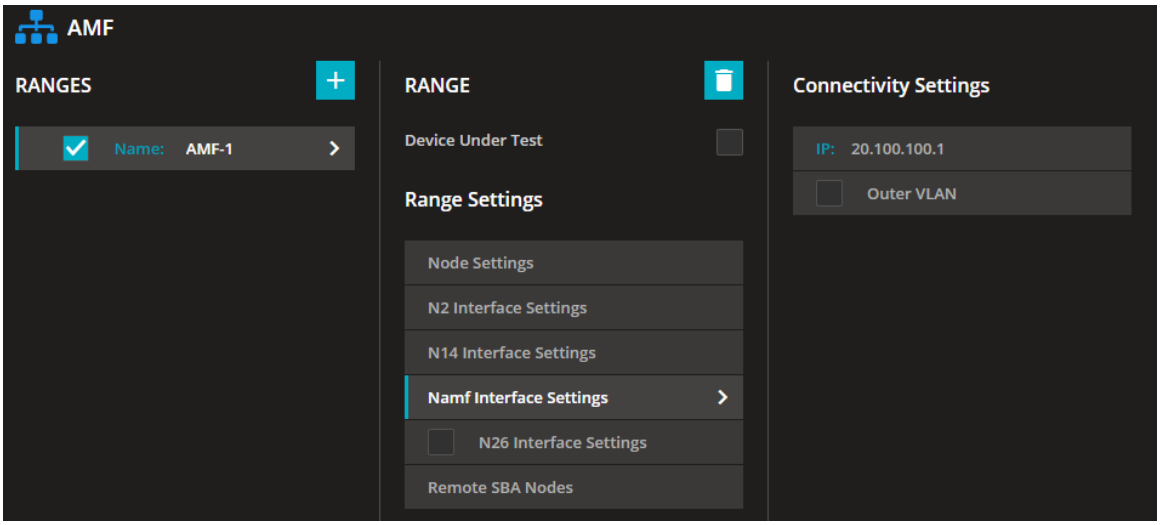
- **AMF configuration**:
  - N2 Interface settings
    - IP address: **20.0.20.10**



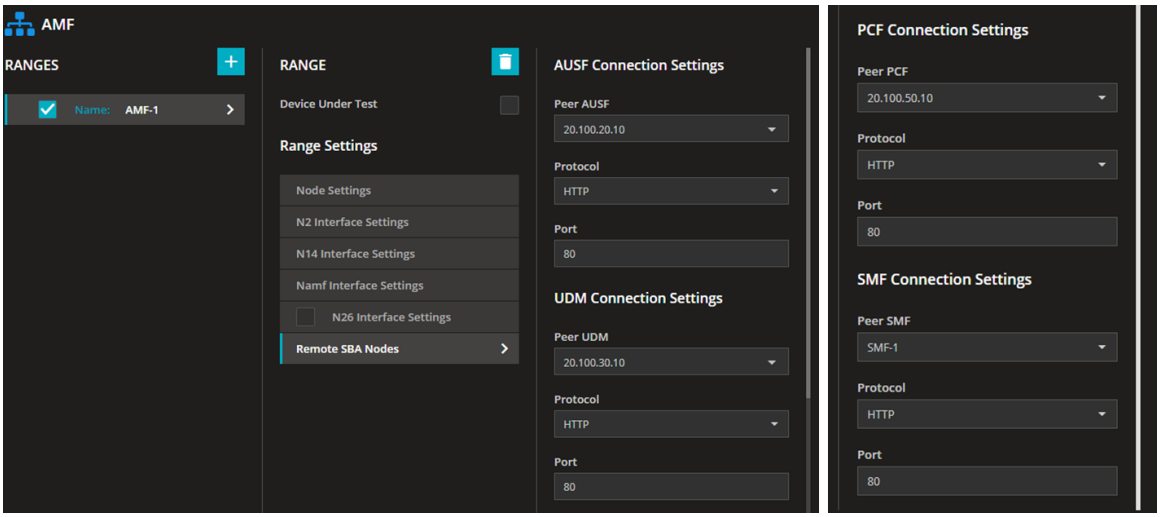
  - N14 Interface settings
    - Local IP address: **20.0.14.10**



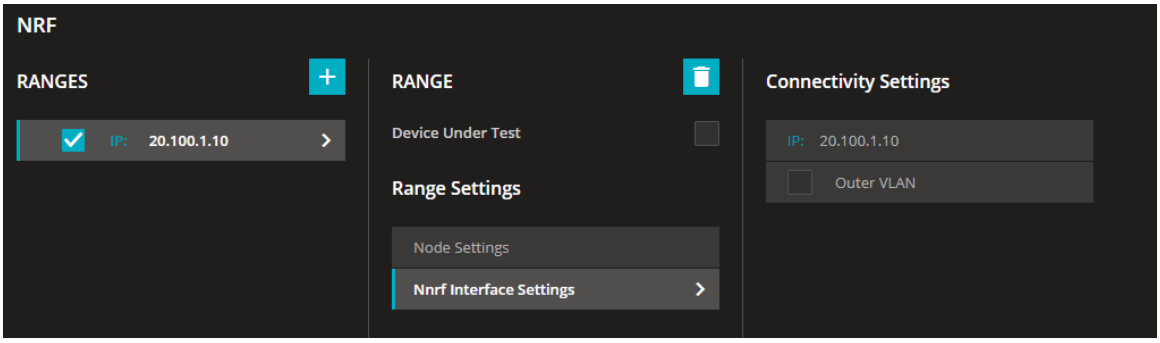
- Namf Interface settings
  - IP address: **20.100.100.1**



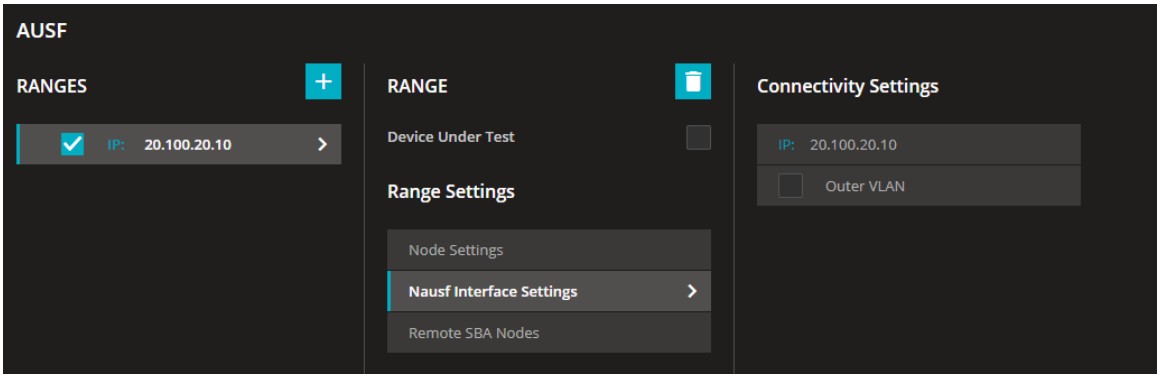
- Remote SBA nodes:
  - AUSF Connection Settings
    - IP address: **20.100.20.10**
  - UDM Connection Settings
    - IP address: **20.100.10.10**
  - PCF Connection Settings
    - IP address: **20.100.30.10**
  - SMF Connection Settings:
    - IP address: **SMF-1** (the IP address is automatically retrieved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



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



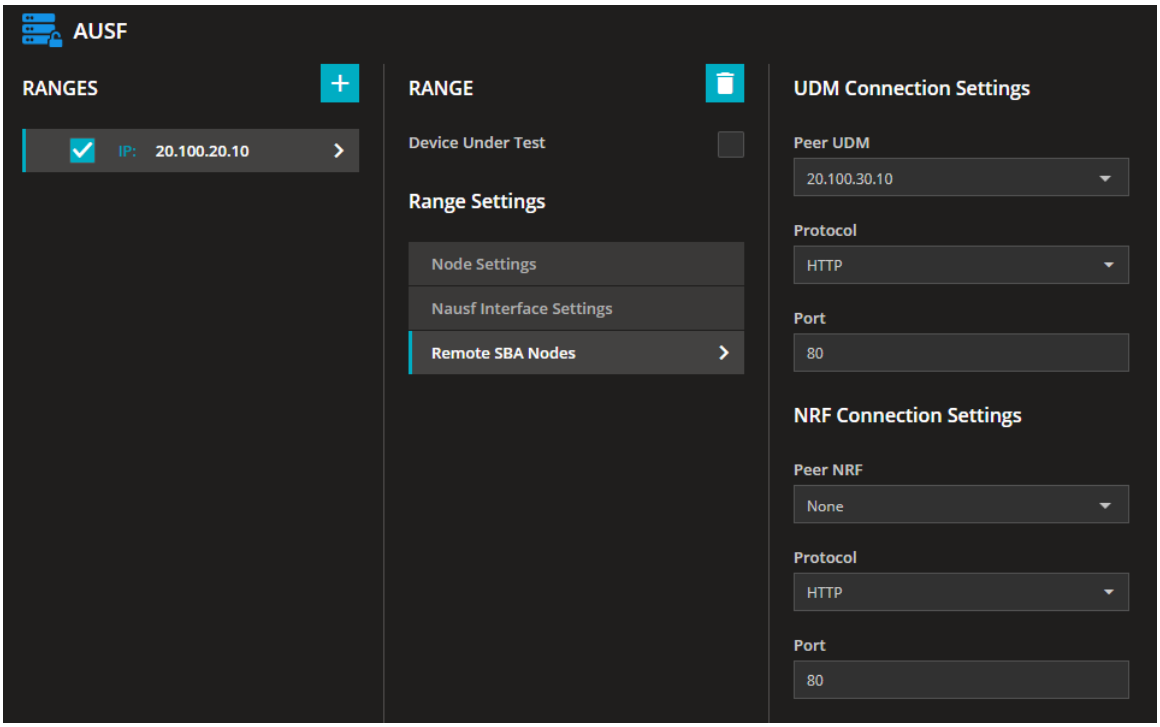
- **AUSF Configuration**
  - Nausf Interface Settings
    - IP address: **20.100.20.10**



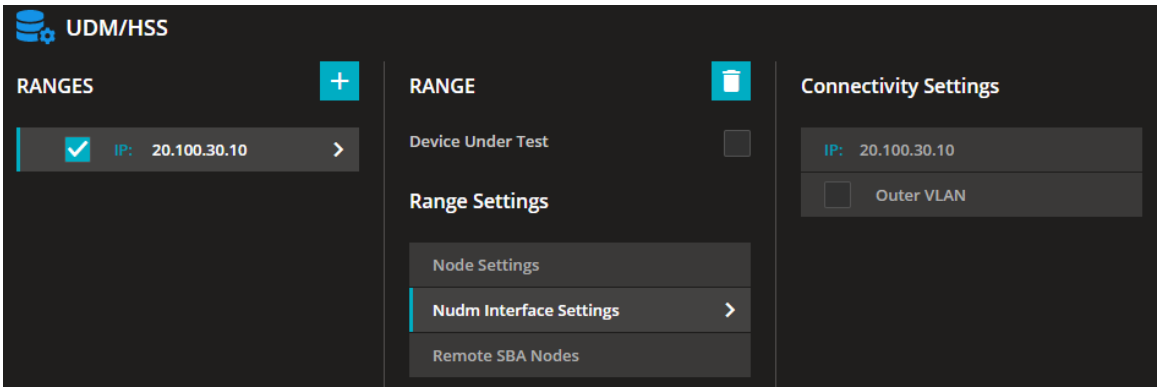
  - Remote SBA Nodes:
    - UDM Connection Settings
      - IP address: **20.100.30.10**
    - Peer NRF: **None**

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/

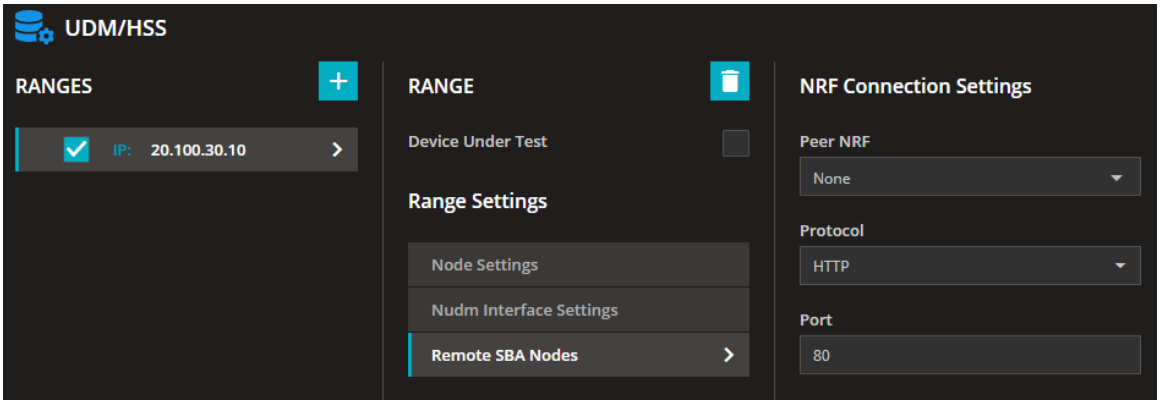


- **UDM/HSS Configuration**
  - Nudm Interface Settings
    - IP address: **20.100.30.10**

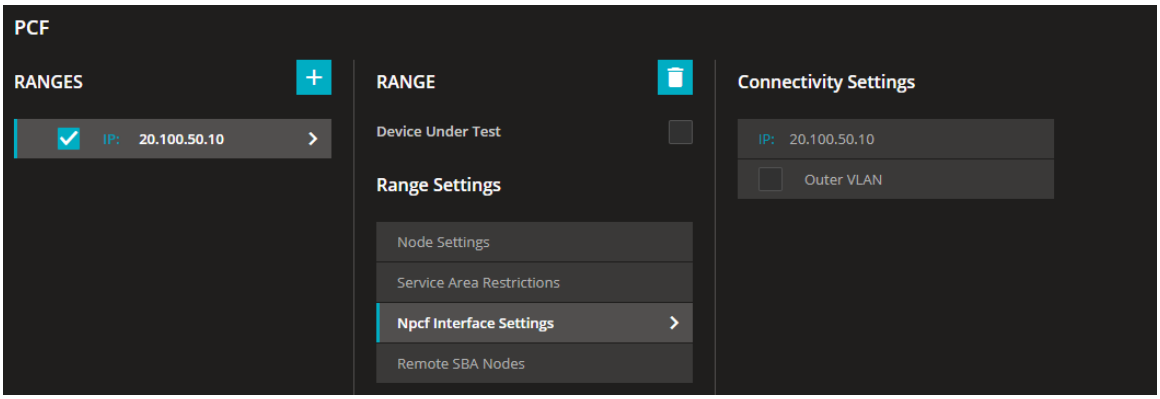


  - Remote SBA Nodes:

- Peer NRF: **None**

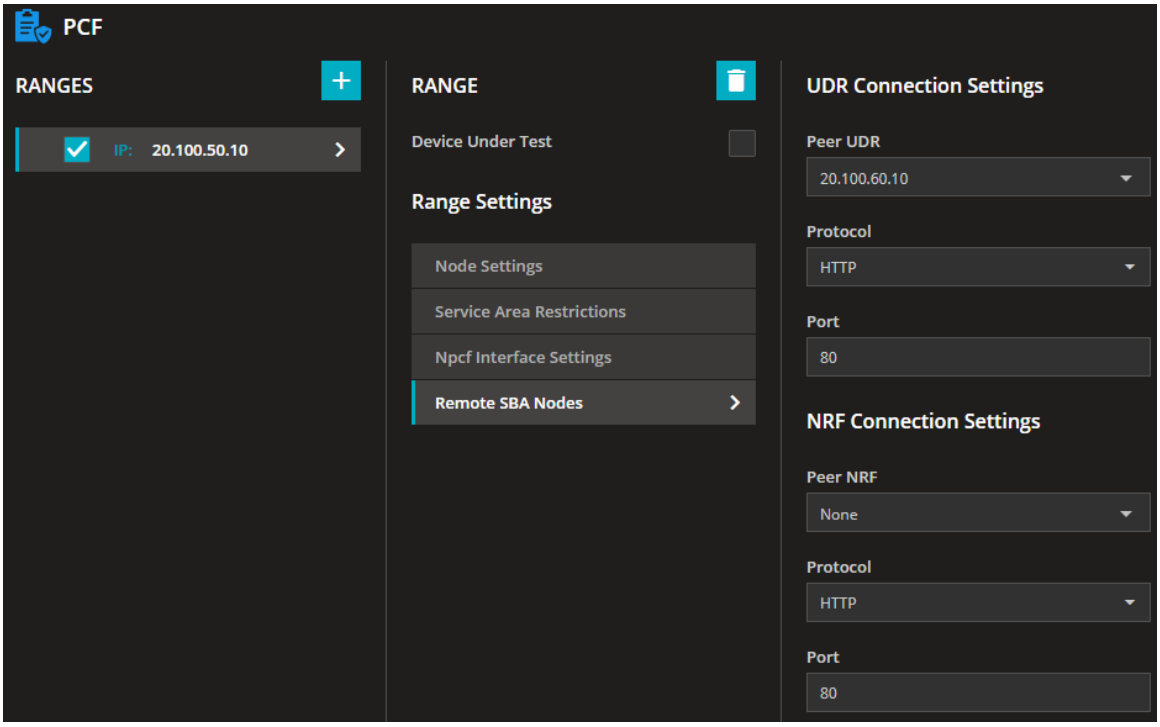


- **PCF configuration**
  - Npcf Interface Settings
    - IP address: **20.100.50.10**

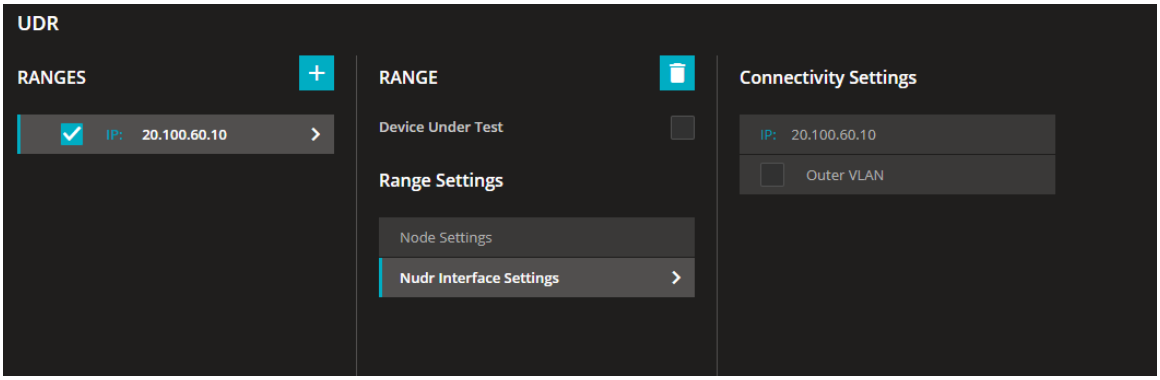


  - Remote SBA Nodes
    - UDR Connectivity Settings
      - IP address: **20.100.60.10**

- Peer NRF: **None**

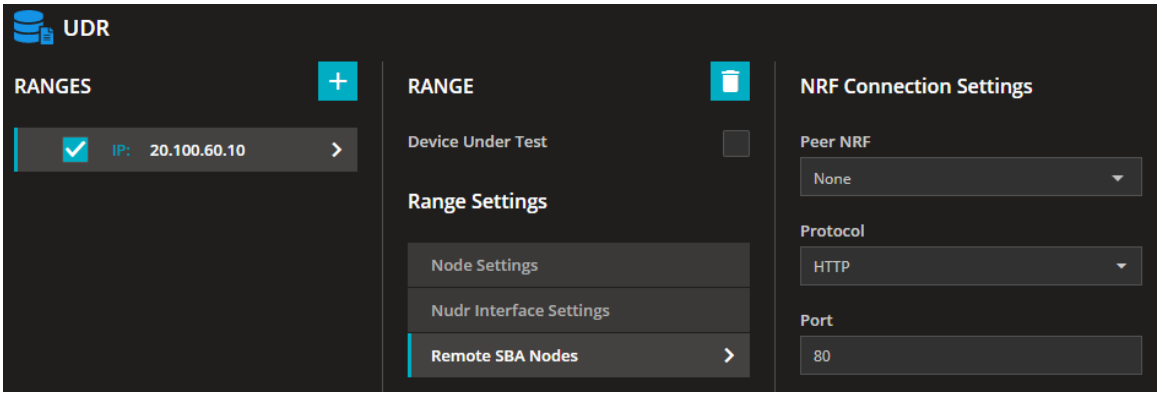


- **UDR Configuration**
  - Nudr Interface Settings
    - IP address: **20.100.60.10**

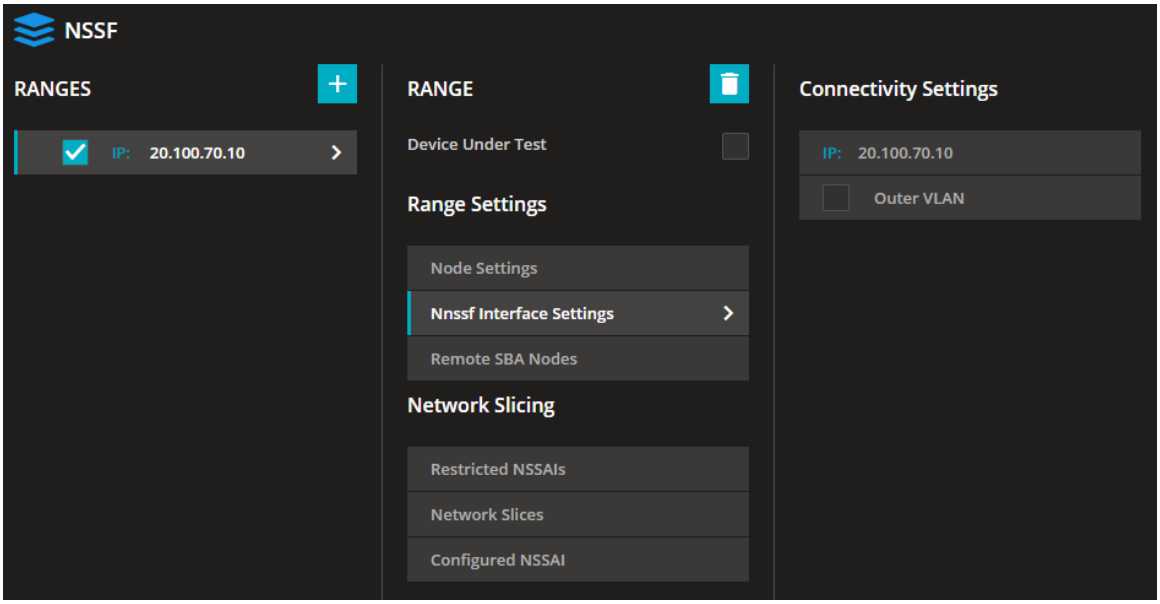


  - Remote SBA Nodes

- Peer NRF: **None**

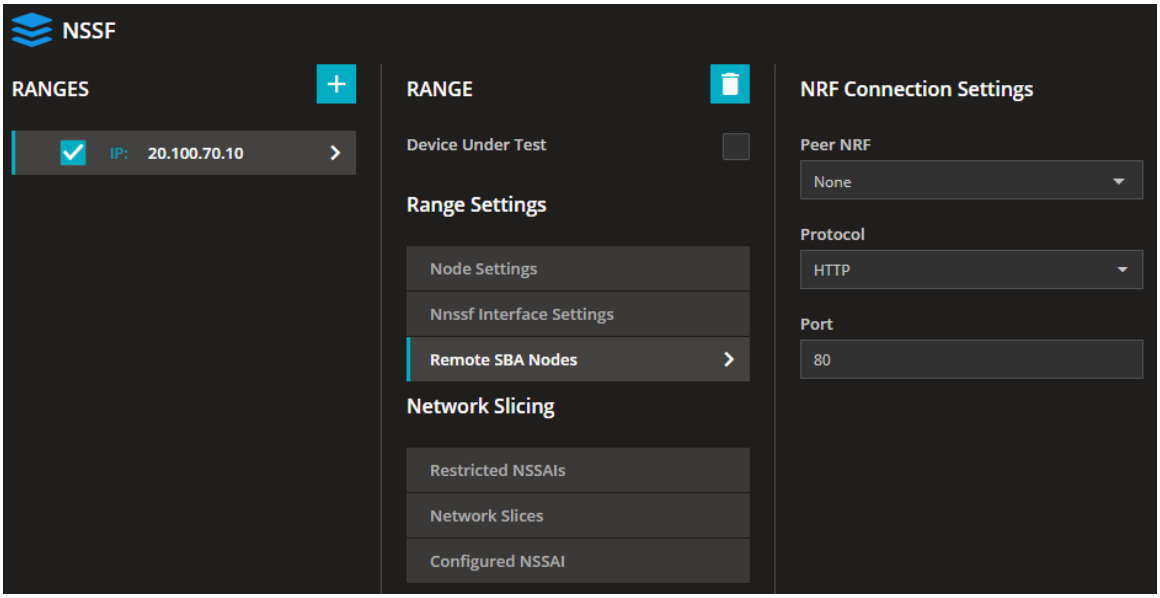


- **NSSF Configuration**
  - Nnssf Interface Settings
    - IP address: **20.100.70.10**

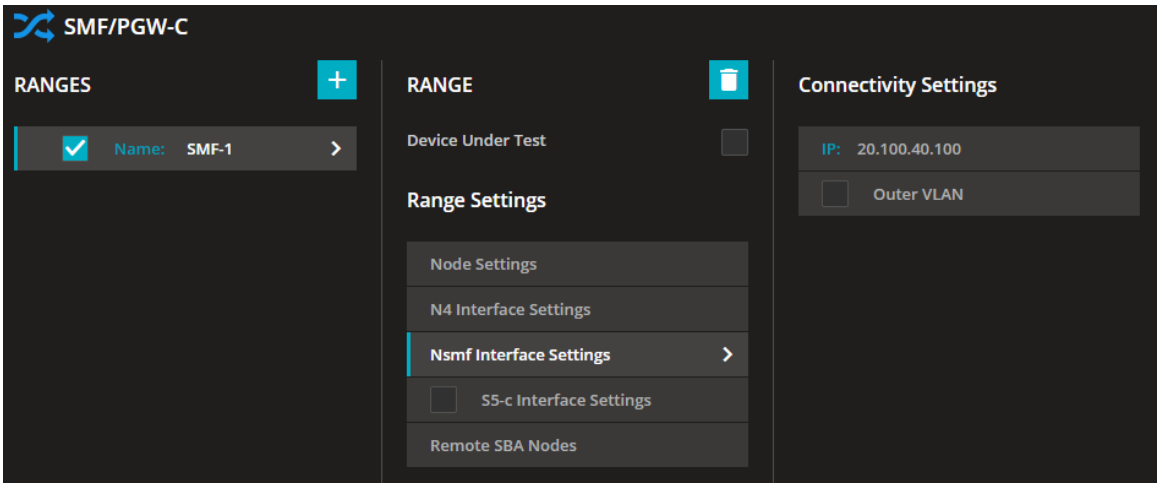


  - Remote SBA Nodes

- Peer NRF: **None**

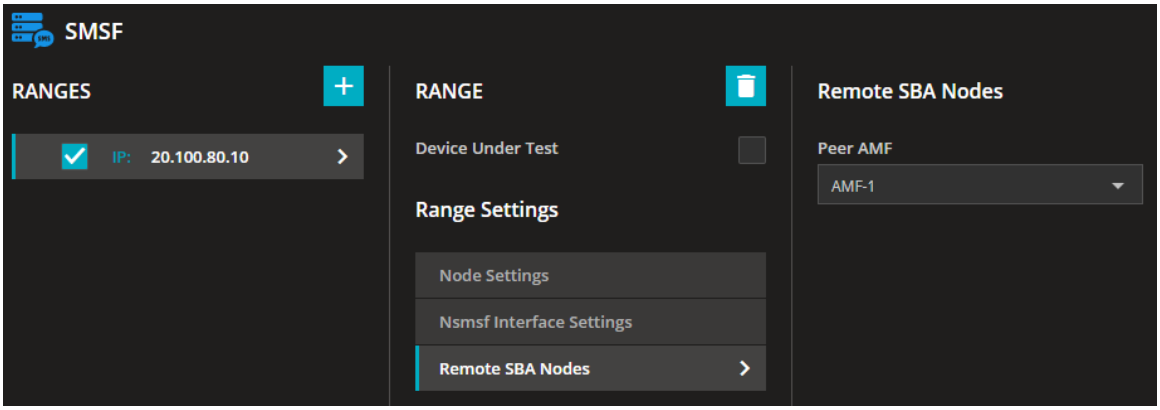


- **SMSF Configuration**
  - Nsmsf Interface Settings
    - IP address: **20.100.80.10**

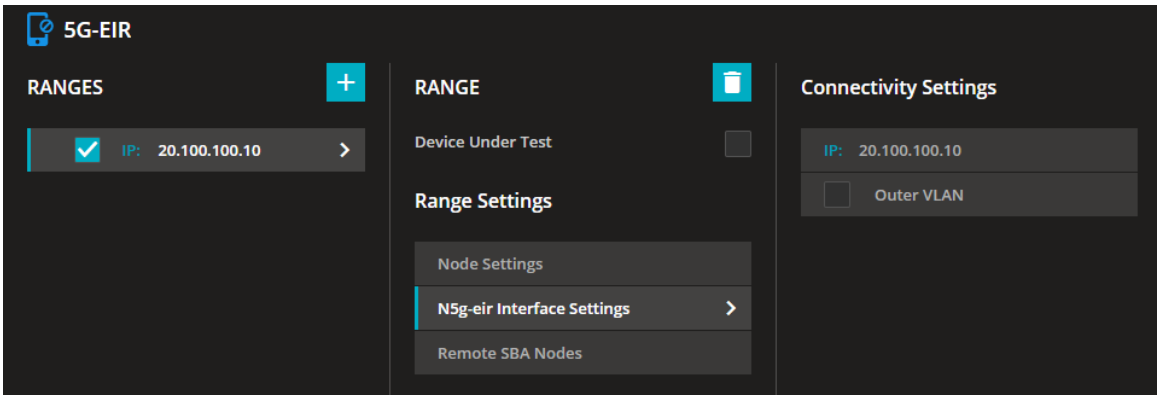


  - Remote SBA Nodes

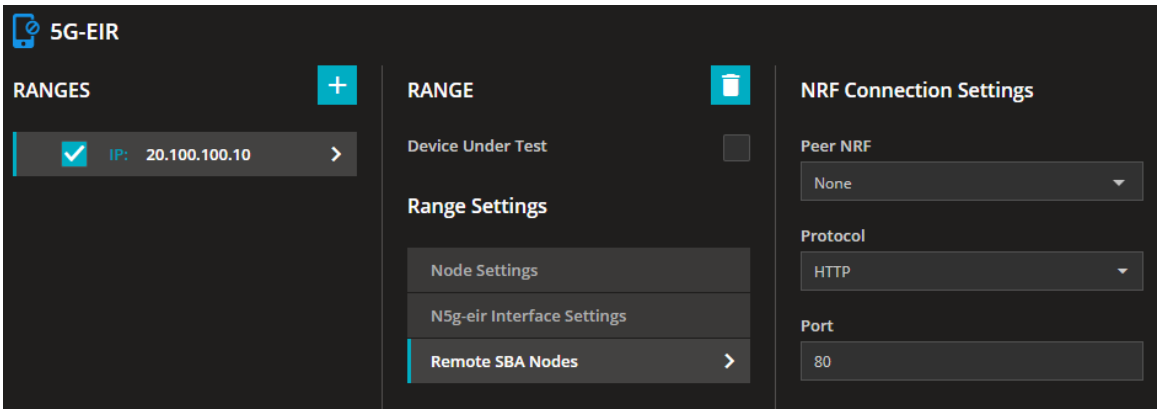
- Peer AMF: **AMF-1**



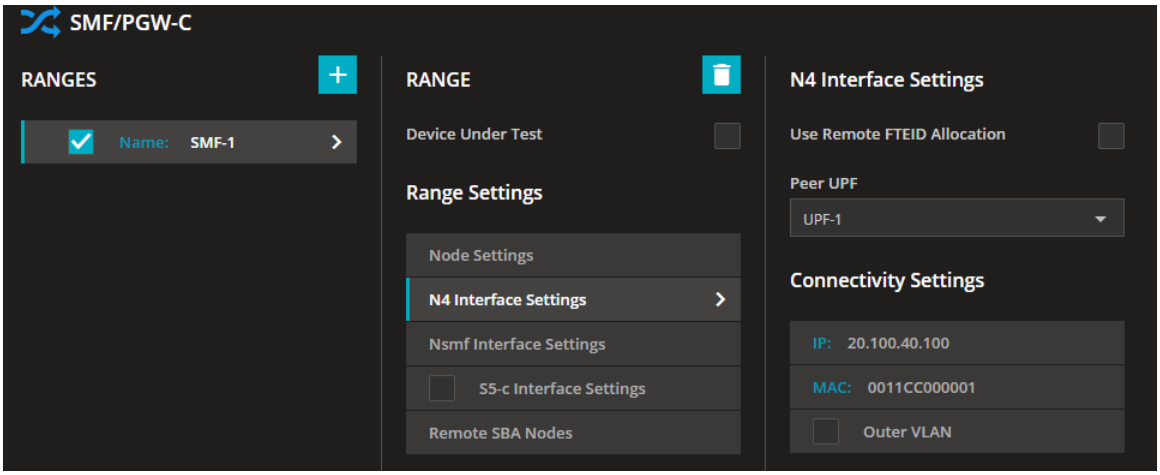
- **5G-EIR Configuration**
  - N5g-eir Interface Settings
    - IP address: **20.100.100.10**



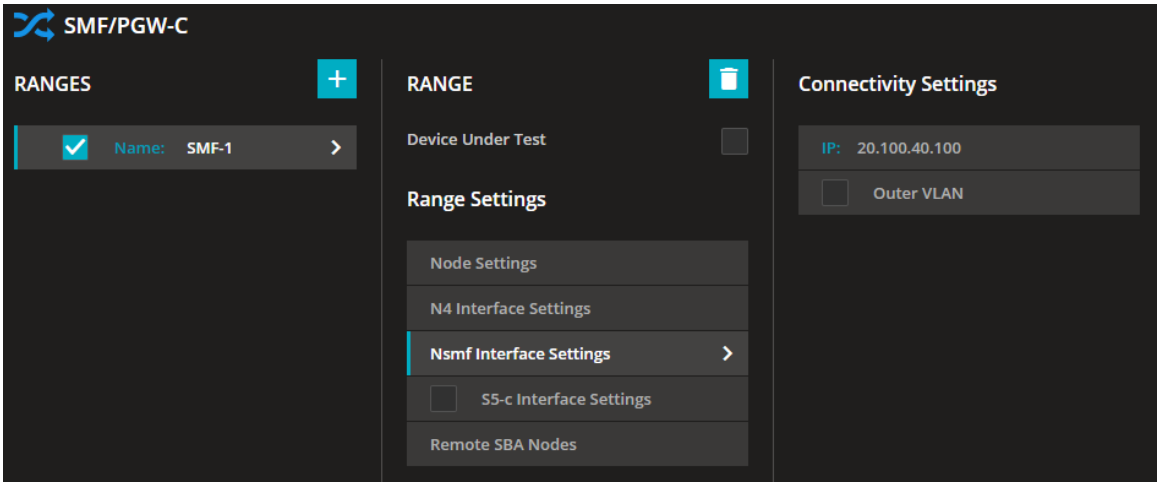
  - Remote SBA Nodes
    - Peer NRF: **None**



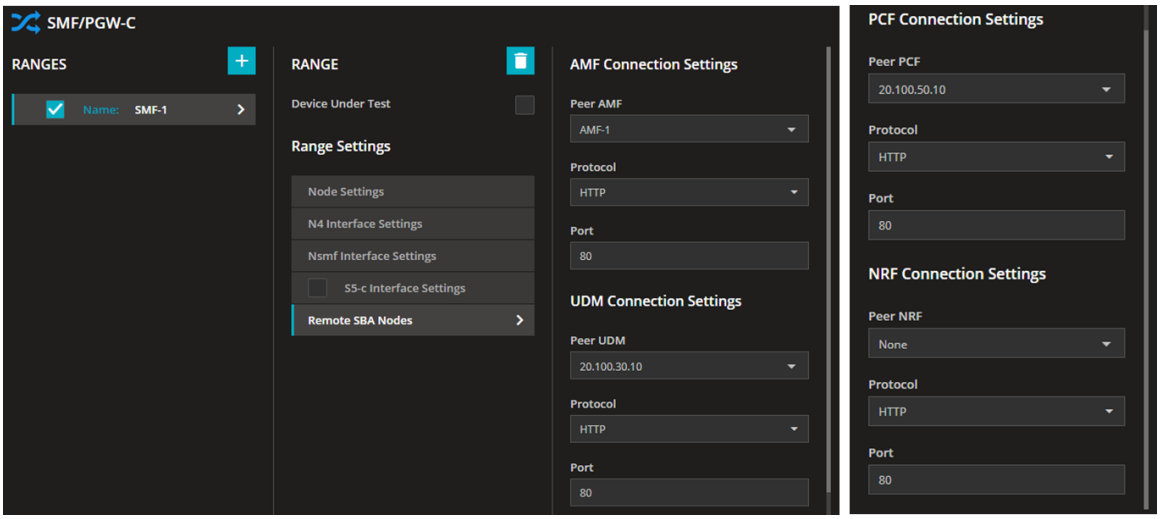
- **SMF/PGW-C configuration**
  - N4 Interface Settings
    - IP address: **20.100.40.100**
    - Peer-UPF : **UPF-1**



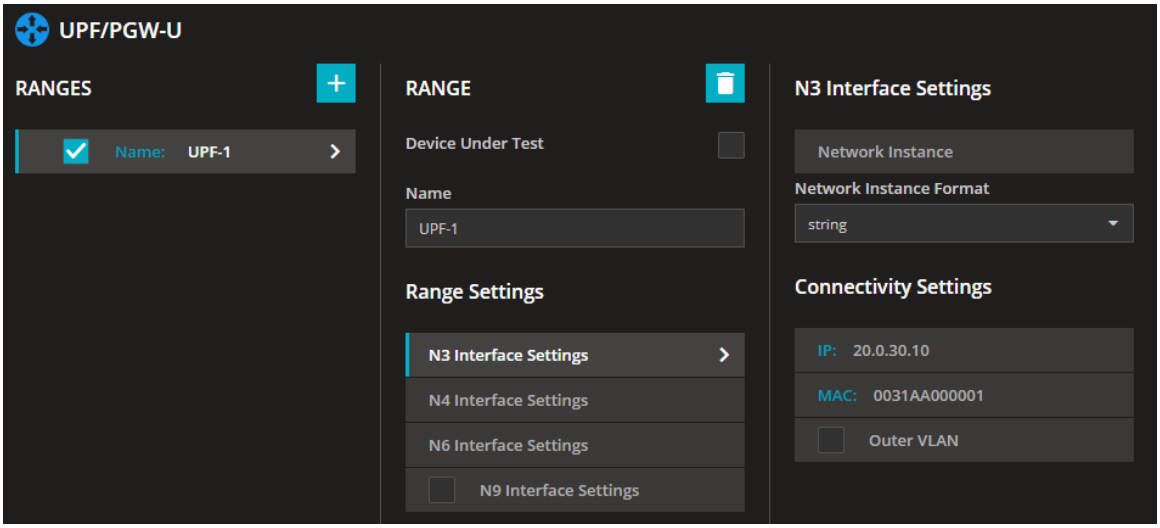
- Nsmf Interface Settings
  - IP Address: **20.100.40.10** (same IP address as N4)



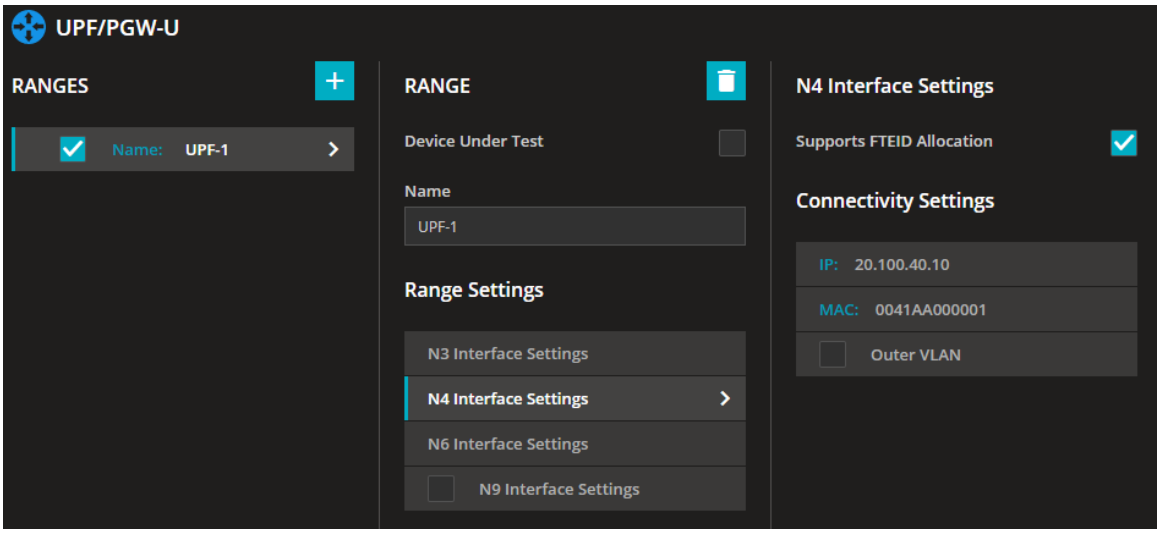
- Remote SBA Nodes
  - AMF Connection Settings
    - IP Address: **AMF-1**
  - UDM Connection Settings:
    - IP Address: **20.100.30.10**
  - PCF Connection Settings
    - IP Address: **20.100.50.10**
  - Peer NRF: **None**



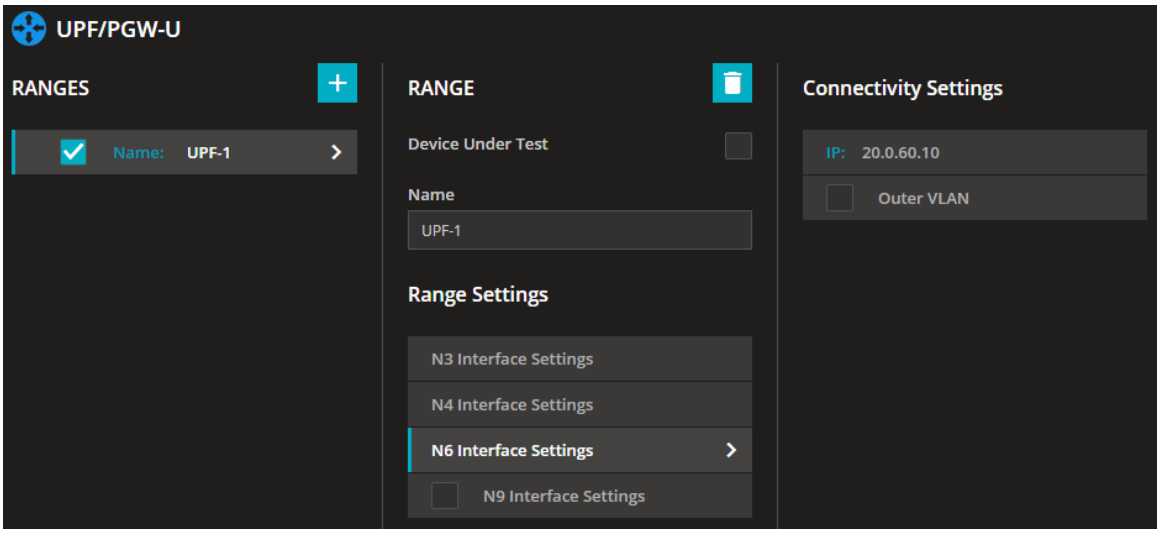
- **UPF/PGW-U Configuration**
  - N3 Interface Settings
    - IP Address: **20.0.30.10**



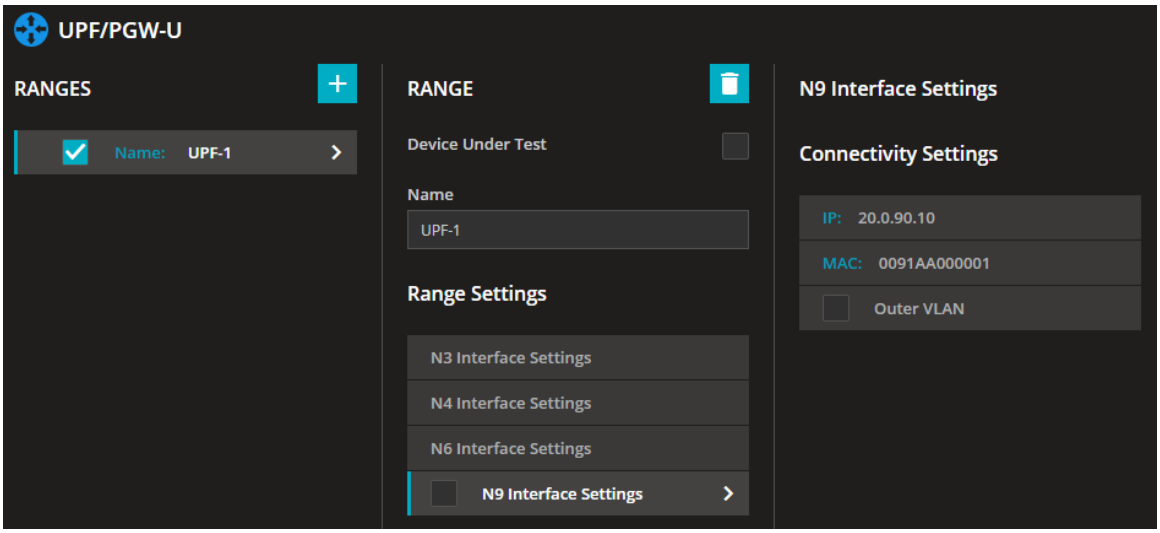
  - N4 Interface Settings
    - IP Address: **20.100.40.10**



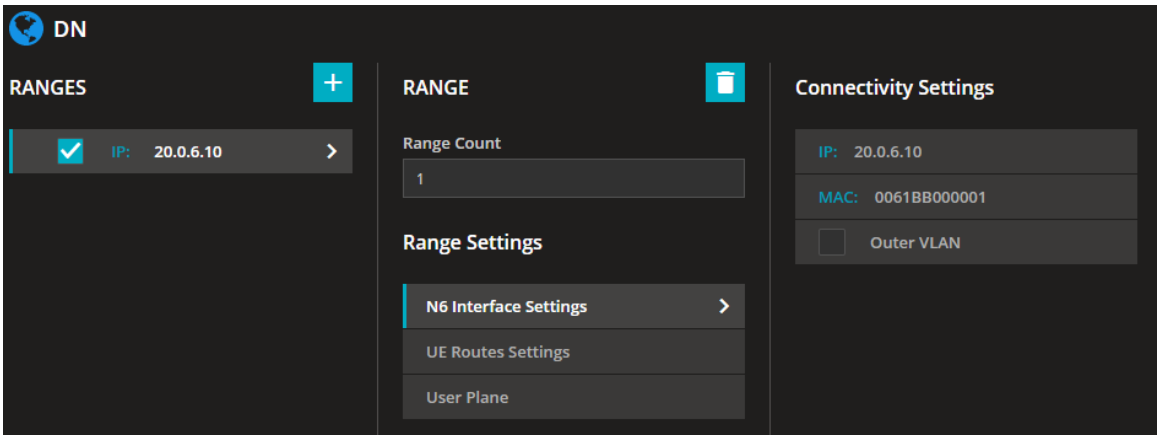
- N6 Interface Settings
  - IP Address: **20.0.60.10**



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



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

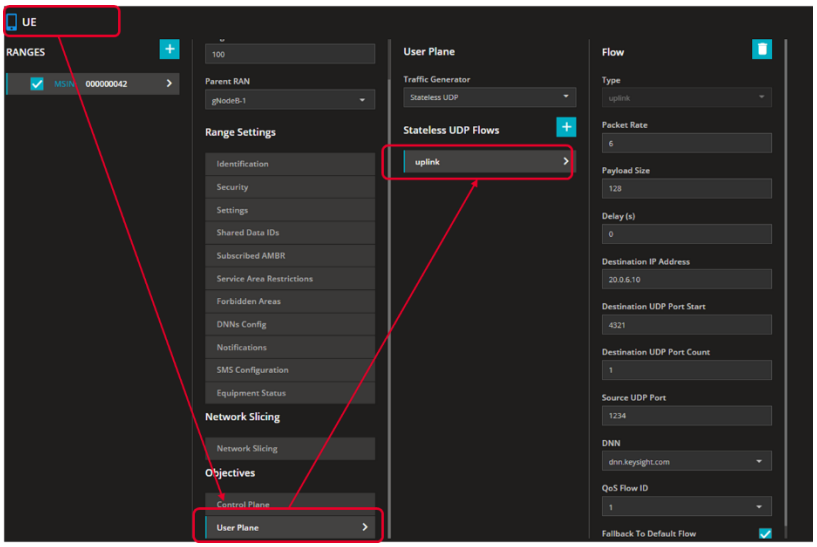


After completing all these steps, you should have the UE range configured and connectivity between 5G Core elements. The next step is to configure traffic flows.

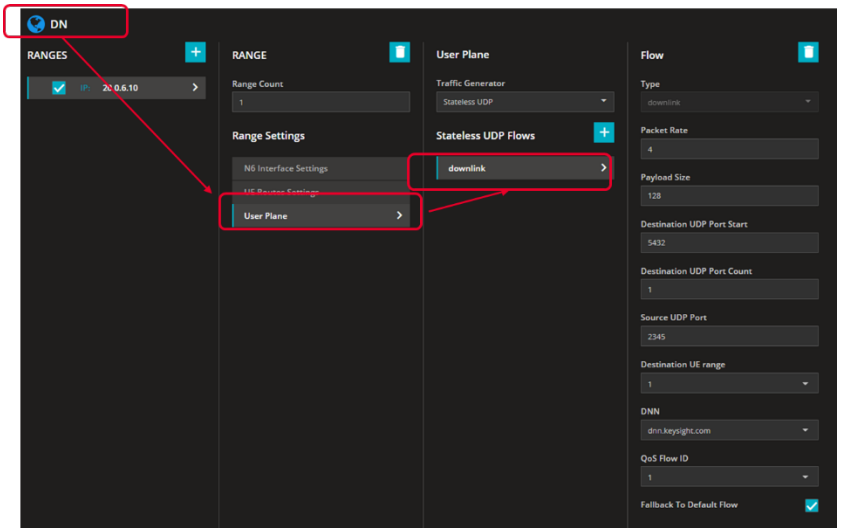
In a wireless network the traffic is generated or received by UE, besides of other activities UE is doing (idle, handover,..). In LoadCore, traffic definition is implemented as UE Objective, therefore the configuration is done by accessing the UE menu:

- 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.

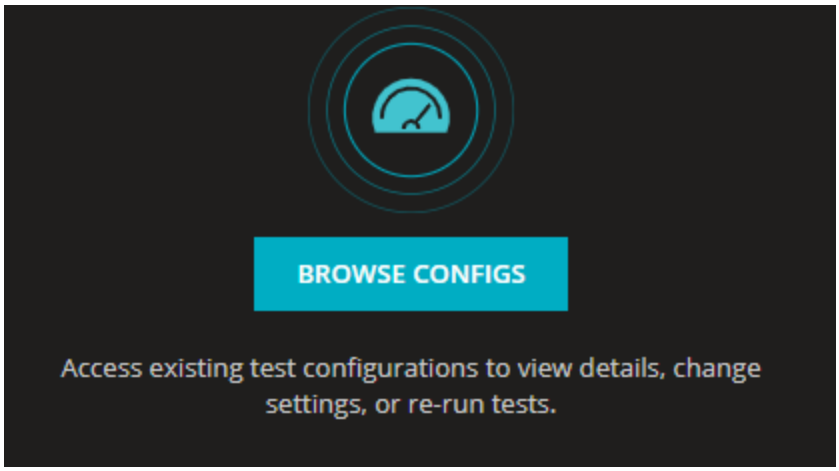
Import a test configuration

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).

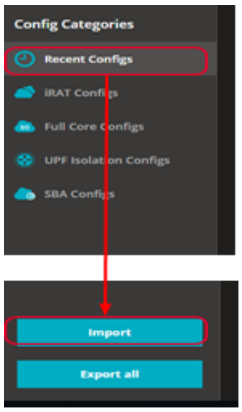
Sample tests can be found on Ixia's download page. You can save the archive and select the tests that best matches your needs.

To import the test configuration file:

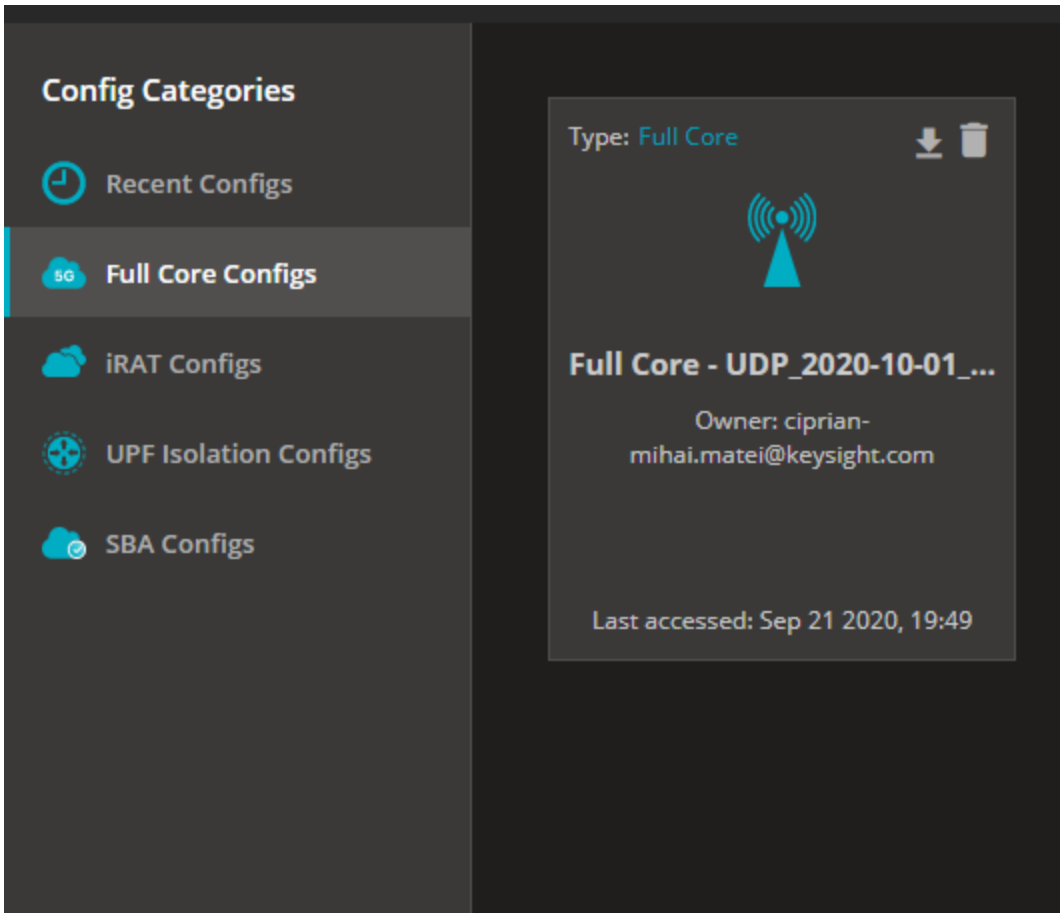
1. Select **LoadCore** > **Browse Configs**:



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.

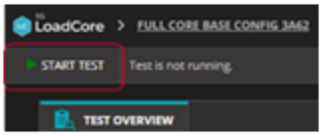


The download page containing the samples can be found at:

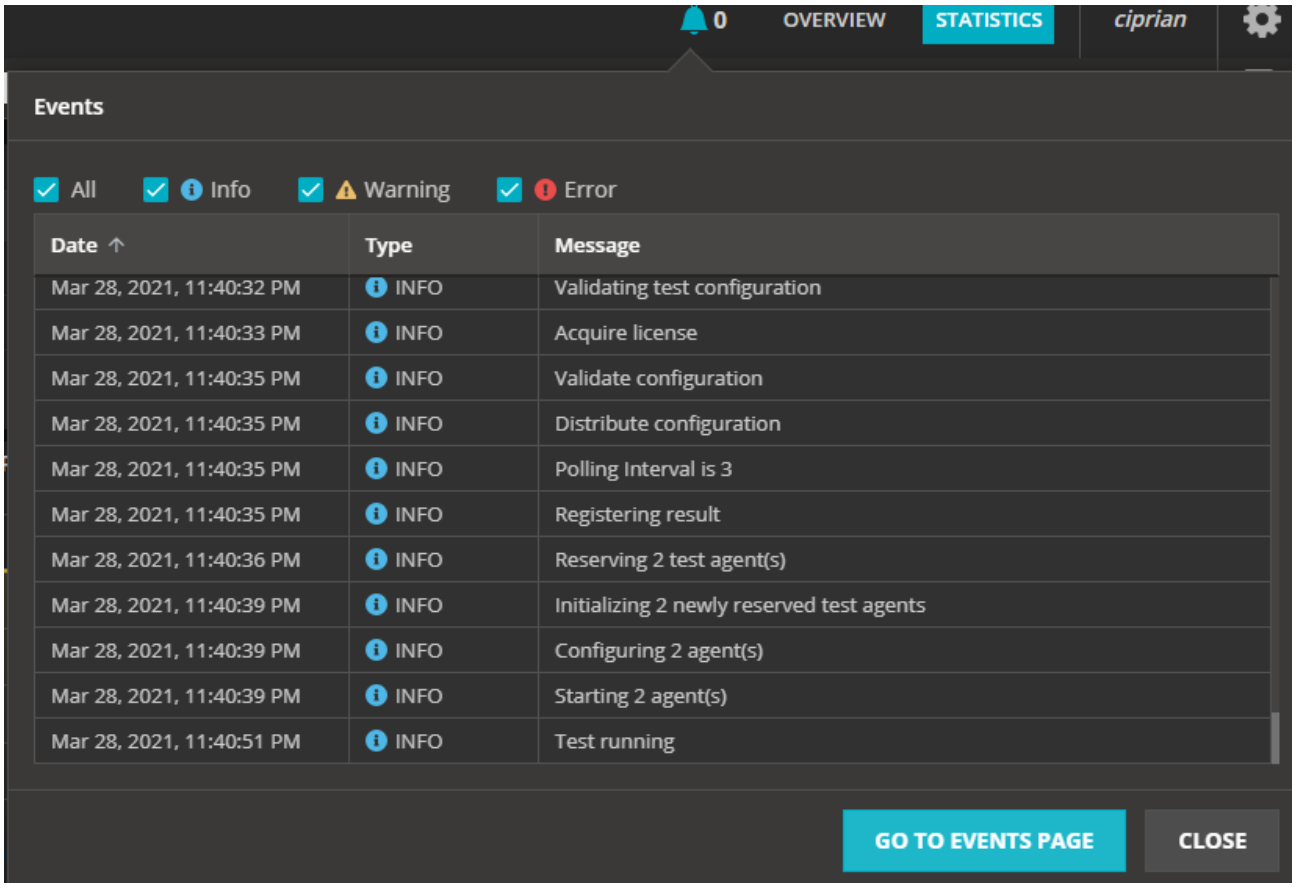
- https://support.ixiacom.com/public/software-downloads/236052

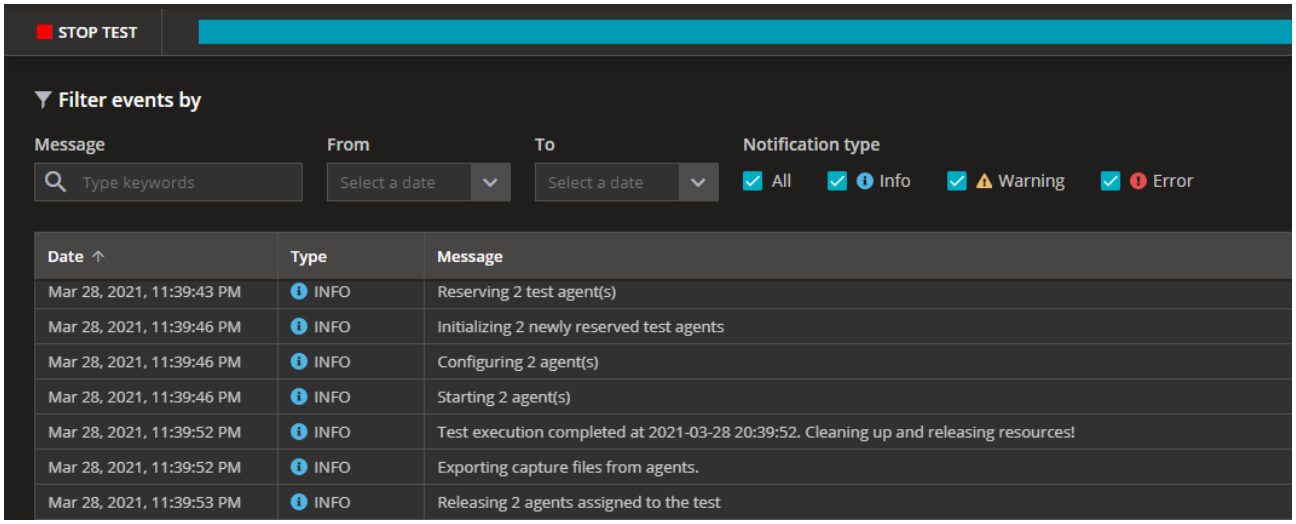
Run a test

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

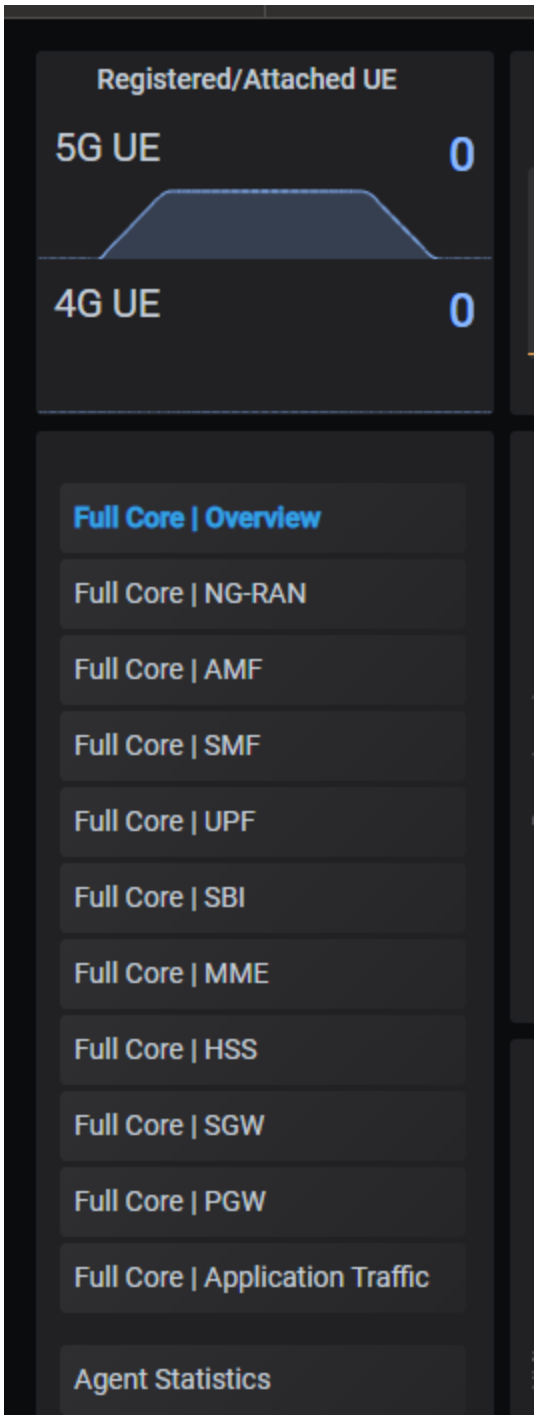

STOP TEST
Filter events by
Message
Type keywords
From
Select a date
To
Select a date
Notification type
All
Info
Warning
Error
Date
Type
Message
Mar 28, 2021, 11:39:43 PM INFO Reserving 2 test agent(s)
Mar 28, 2021, 11:39:46 PM INFO Initializing 2 newly reserved test agents
Mar 28, 2021, 11:39:46 PM INFO Configuring 2 agent(s)
Mar 28, 2021, 11:39:46 PM INFO Starting 2 agent(s)
Mar 28, 2021, 11:39:52 PM INFO Test execution completed at 2021-03-28 20:39:52. Cleaning up and releasing resources!
Mar 28, 2021, 11:39:52 PM INFO Exporting capture files from agents.
Mar 28, 2021, 11:39:53 PM 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.



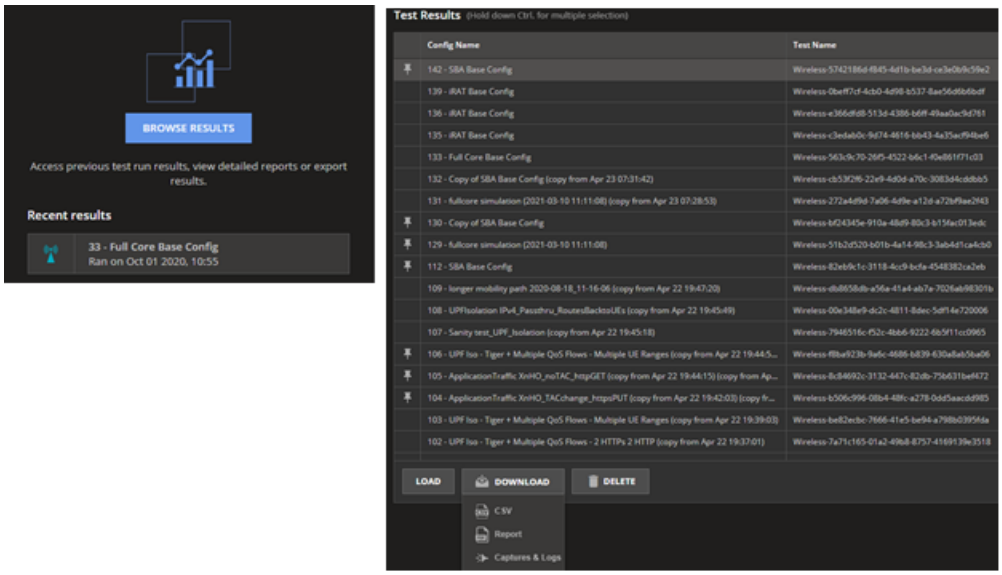
- **Full Core | Overview**: general dashboard displaying a global test overview: how menu UEs attached, traffic volume on user plane and control plane.
- **Full Core | Core node**: displays relevant statistics for that node: procedures, rates, errors.

- **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.



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:

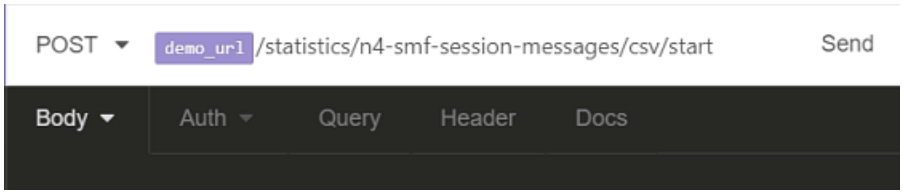
- **agent_1**: 10.73.51.91 > `<demo_url>` is **https://10.73.51.91:443/api/v1**
- **agent_2**: 10.73.52.85 > `<demo_url>` is **https://10.73.52.85:443/api/v1**

	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	agetn_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:

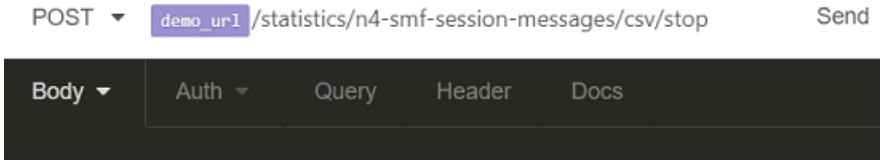
   POST {{ demo_url }}/statistics/n4-smf-session-messages/csv/start



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

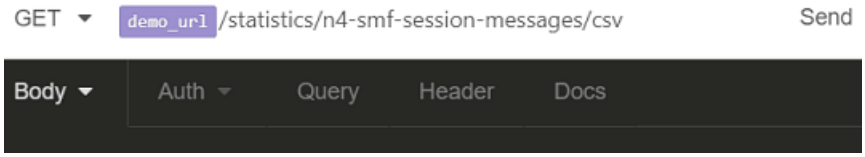
2. To Stop csv logging, issue the following command:

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

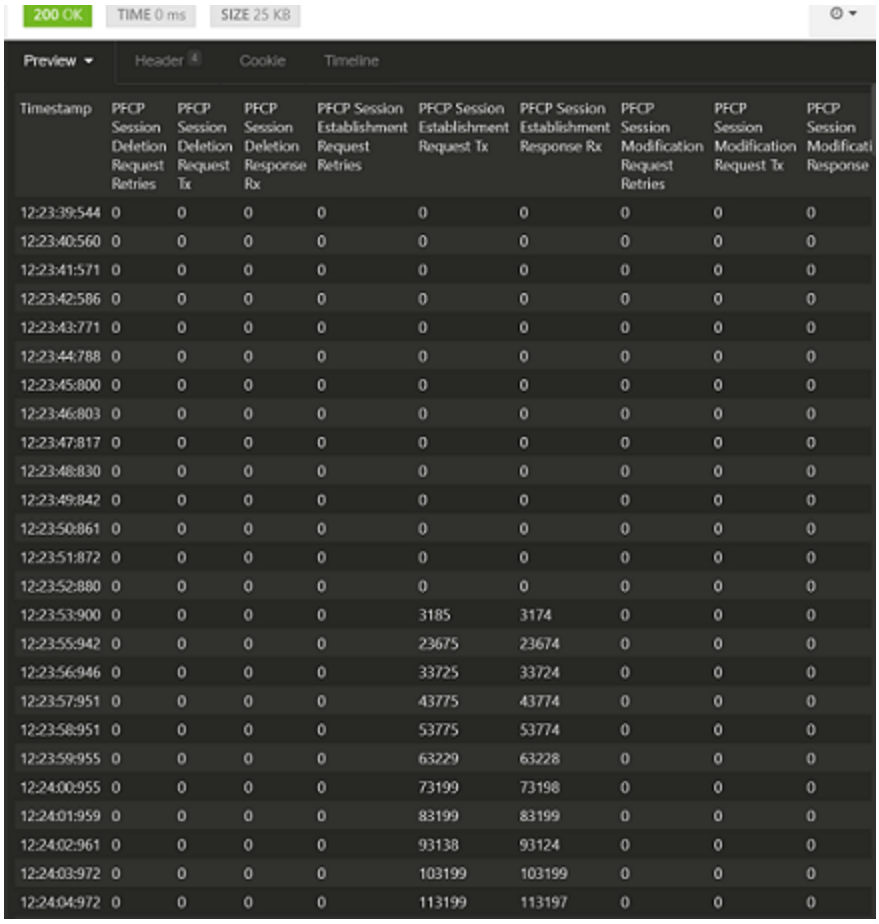


3. Once the test has stopped, you can retrieve entire `csv` using command:

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



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 4 | 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 Modificati Response
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	0	3185	3174	0	0	0
12:23:55:942	0	0	0	0	23675	23674	0	0	0
12:23:56:946	0	0	0	0	33725	33724	0	0	0
12:23:57:951	0	0	0	0	43775	43774	0	0	0
12:23:58:951	0	0	0	0	53775	53774	0	0	0
12:23:59:955	0	0	0	0	63229	63228	0	0	0
12:24:00:955	0	0	0	0	73199	73198	0	0	0
12:24:01:959	0	0	0	0	83199	83199	0	0	0
12:24:02:961	0	0	0	0	93138	93124	0	0	0
12:24:03:972	0	0	0	0	103199	103199	0	0	0
12:24:04:972	0	0	0	0	113199	113197	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

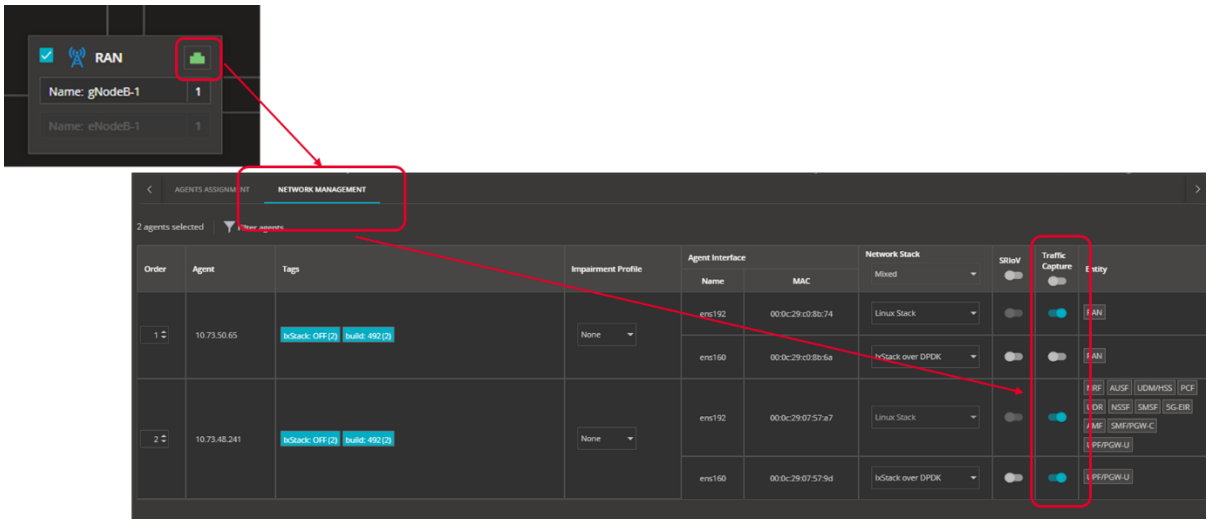
Preview | Header 4 | Cookie | Timeline

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 Modificati 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

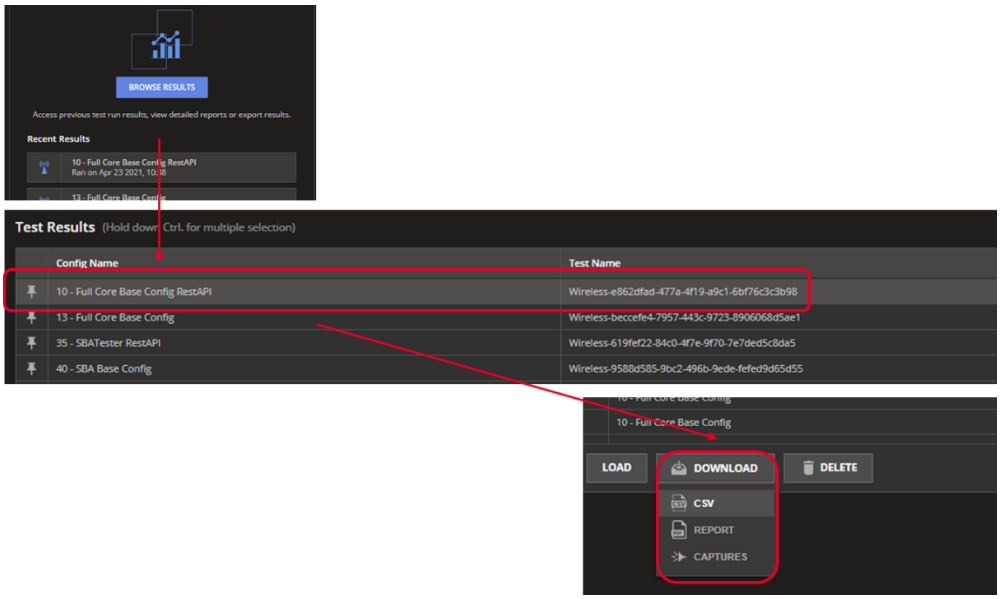
To enable packet capture from the LoadCore UI, select the agent assignment icon on any node, and select the Network Management tab.



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.

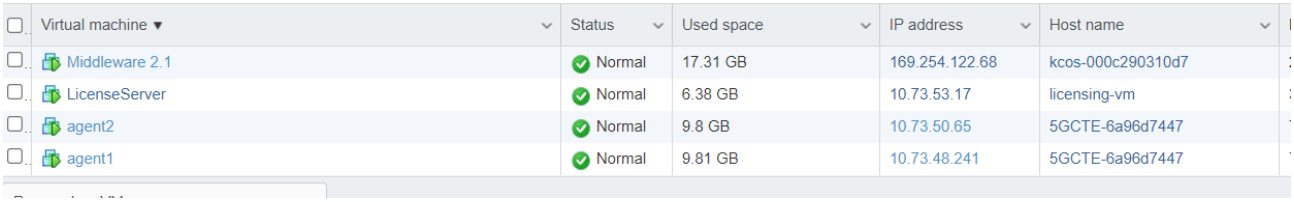
Using Rest API commands

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

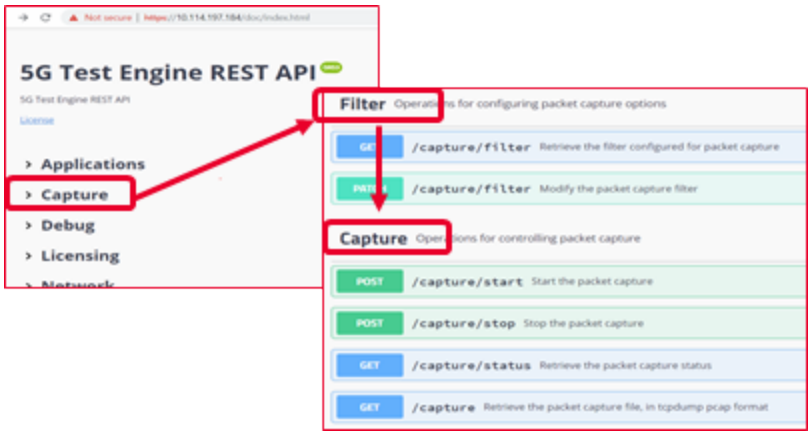
1. Set up the capture filter using the PATCH command:
   ```
   PATCH {{ demo_url }}/capture/filter -i ens160
   ```
2. The packet capture can be started using:
   ```
   POST {{ demo_url }}/capture/start
   ```
3. The packet capture can be stopped using:
   ```
   POST {{ demo_url }}/capture/stop
   ```
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:

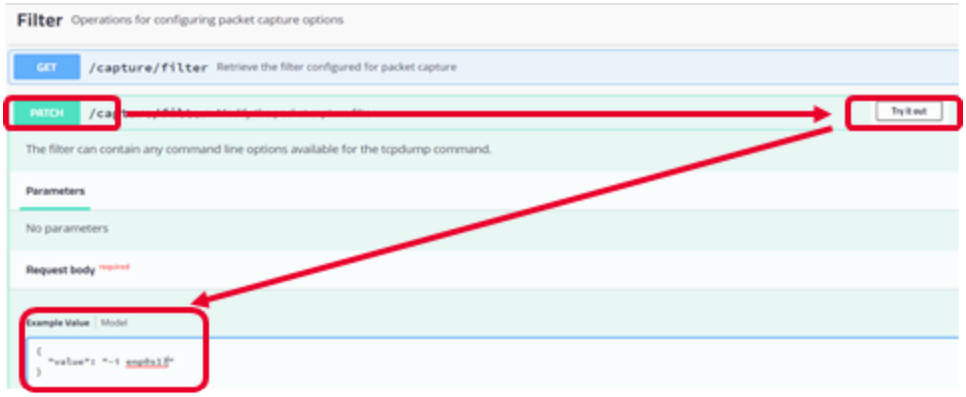
1. Connect to the Agent User Guide (`http://<IP_address_of_the_Agent>`).

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

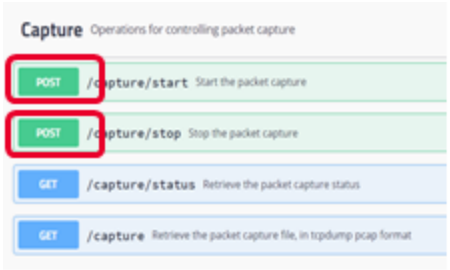


   In this example the interface name is **ens160** and this value is updated in the message body. This is the test interface used to connect the two agents.

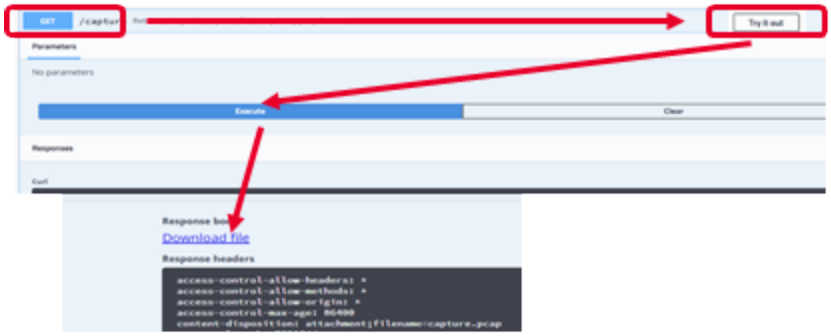
3. Select **Filter** and, then, the **PATCH** operation. Update the interface name and select the **Execute** command.



4. The capture operation can be started and stopped after running the test.



5. 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
   - 32 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

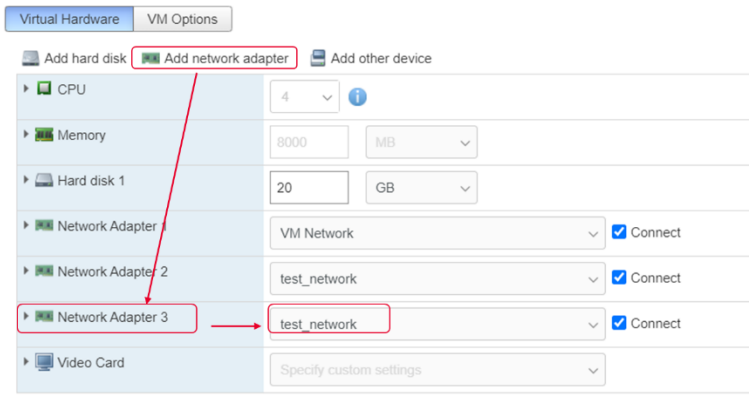
For different scenarios, two additional agents will be deployed having IxStack over DPDK/Raw on. In this way, you can run a simple UDP test for demo purposes with the previously deployed agents (agent_1 and agent_2) and when necessary to show TCP traffic type, you can run the test with the IxStack over DPDK/Raw enabled agents.

Installing additional interfaces on the test agents

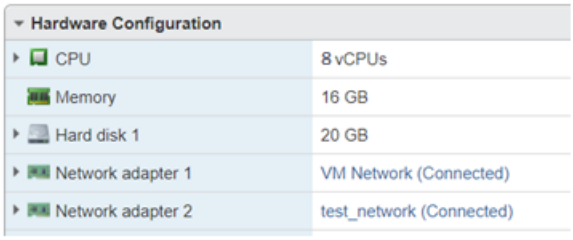
In order to have full flexibility we will add an additional test interface to each agent. Also, this extra interface is necessary because the SBI nodes does not have yet support for IxStack over DPDK/Raw.

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:



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):

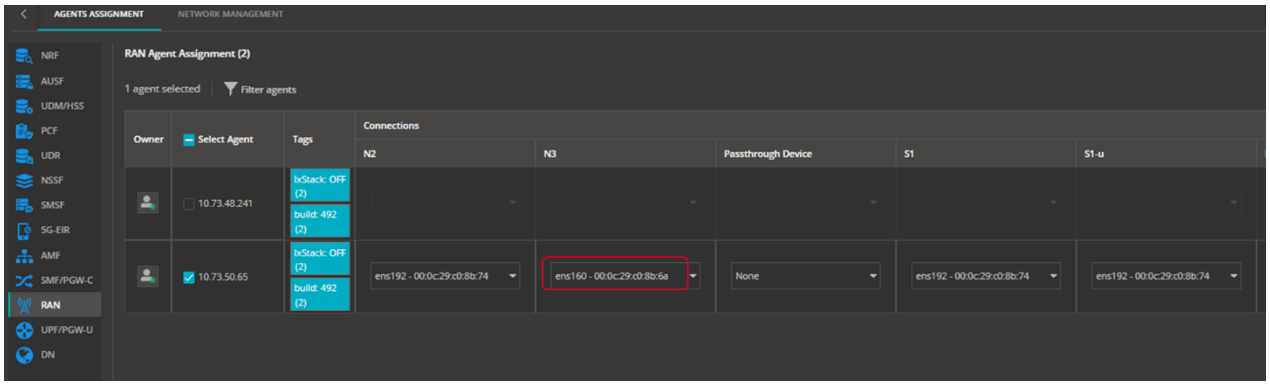
- One for management (**ens32**)
- One for test, running IxStack over DPDK/Raw, for User Plane (**ens160**)
- One for test, for Control Plane, no IxStack over DPDK/Raw (**ens192**)

- 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**)

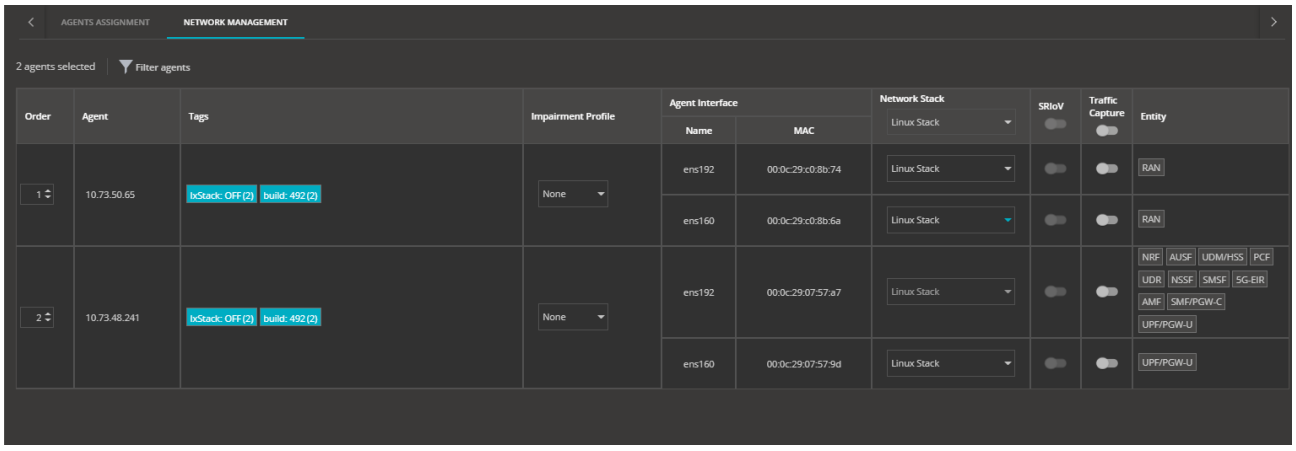
Enabling IxStack over DPDK/Raw

To run application traffic, you need to use IxStack on test interfaces that are assigned for user plane (N3, N6).

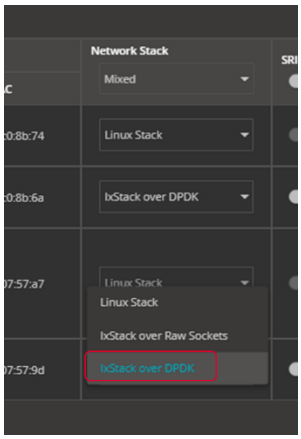
From the main topology, select the green icon (for agent assignment) and for NG-RAN and UPF/PGW-U assign the interface **ens160** for N3 and all the other interfaces on the nodes will remain on **ens192**.

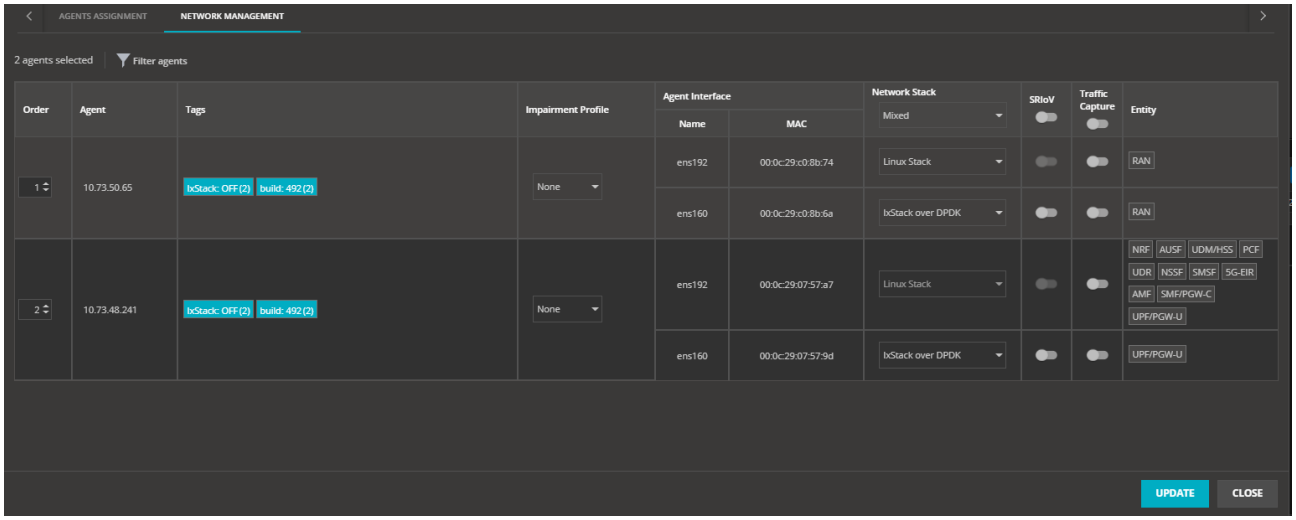


After completing this step, select the **Network Management** tab. Notice that **ens160** is assigned to two nodes, and **ens192** is assigned to all nodes, as follows:



For the rows in the table with **ens160**, set **Network Stack** to **IxStack over DPDK/Raw** by selecting it from the drop-down list.



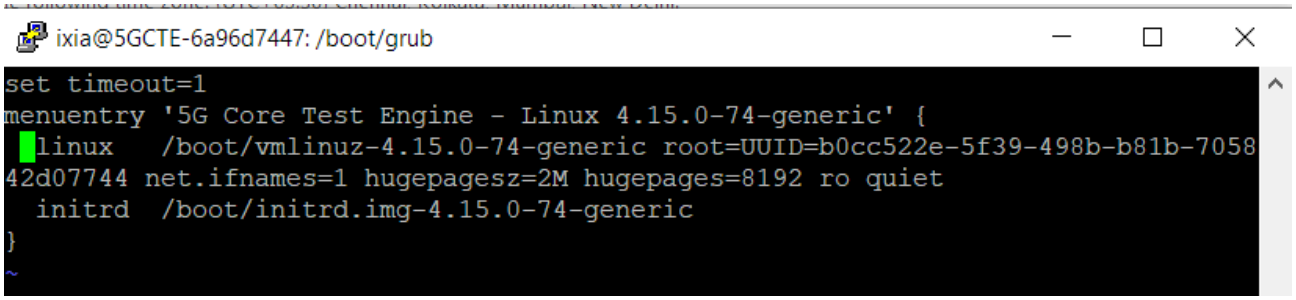


Configure HugePages on both Agents

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.

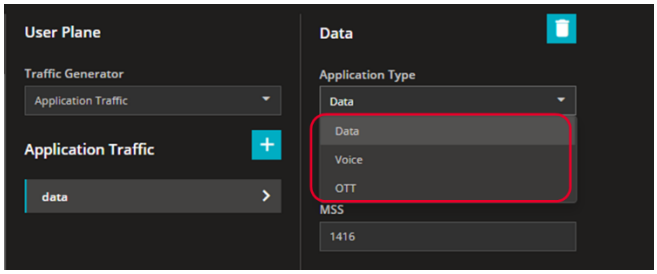
To do this, edit the `/boot/grub/grub.cfg` file and add `hugepages=8192`.



To complete the process, reboot the agent.

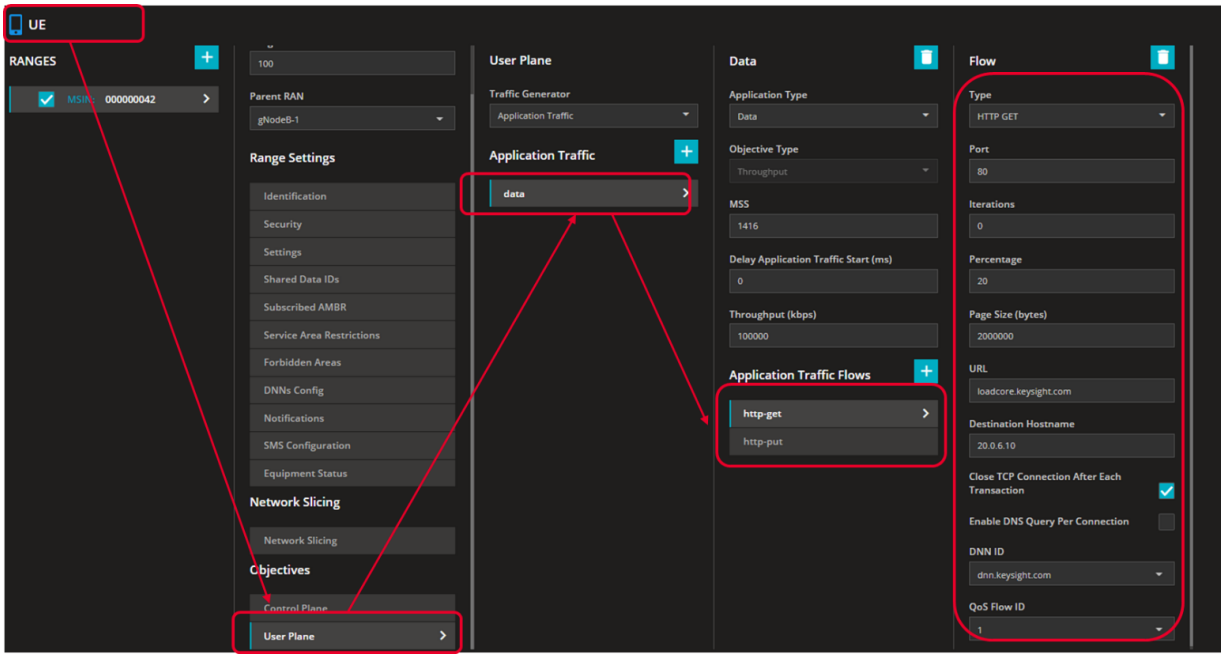
Application traffic configuration

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:

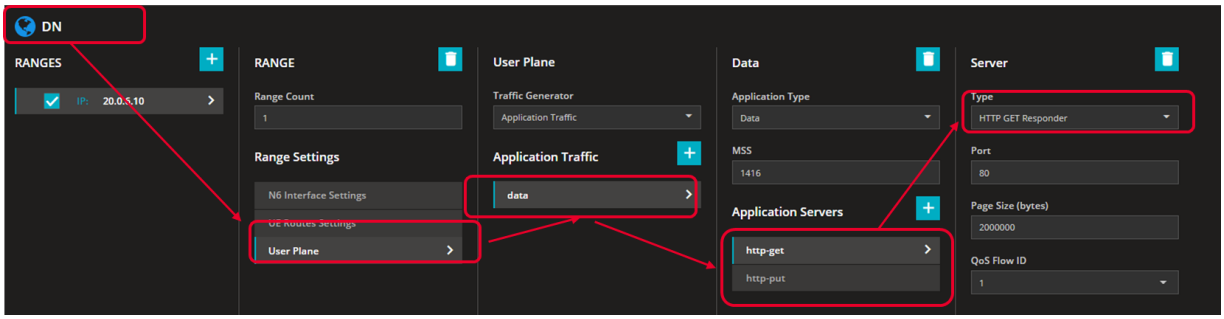
- On **UE** > **User Plane** objective: this controls the traffic that is sent or received by the UE, for both directions.



- 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.


DN
RANGES
IP: 20.0.5.10
RANGE
Range Count
1
Range Settings
N6 Interface Settings
User Plane
User Plane
Traffic Generator
Application Traffic
Application Traffic
data
Data
Application Type
Data
MSS
1416
Application Servers
http-get
http-put
Server
Type
HTTP GET Responder
Port
80
Page Size (bytes)
2000000
QoS Flow ID
1

Create Custom Dashboard

There are two way to create a custom dashboard in order to display statistics:

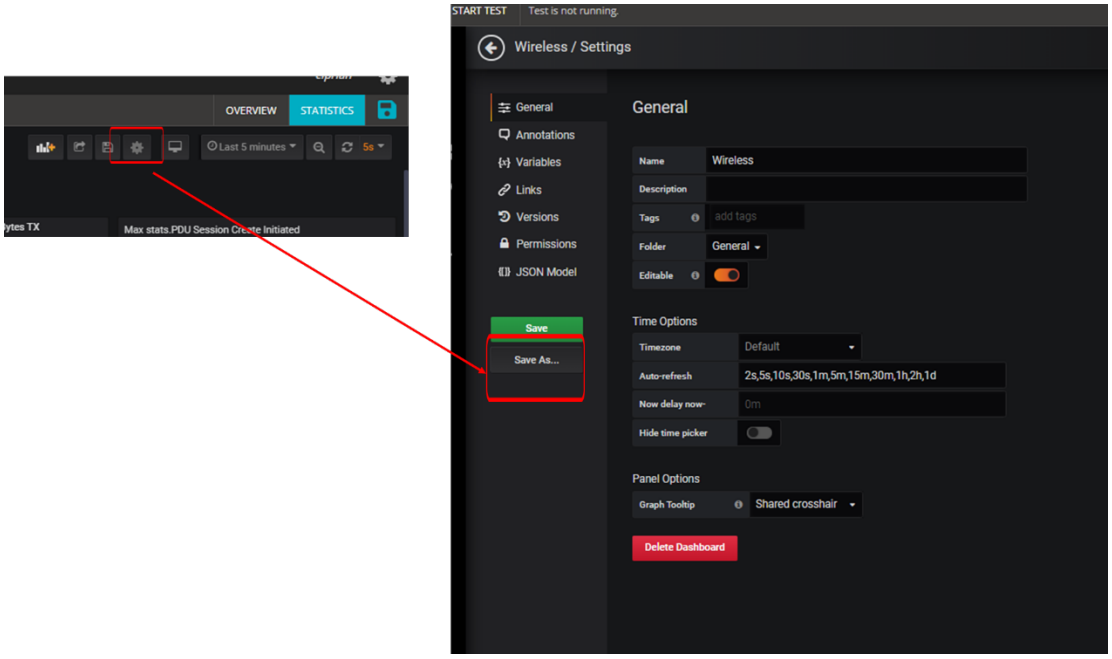
- create a dashboard based on an existing one (a pre-defined dashboard), or
- create a new dashboard and customize it based on your requirements

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.

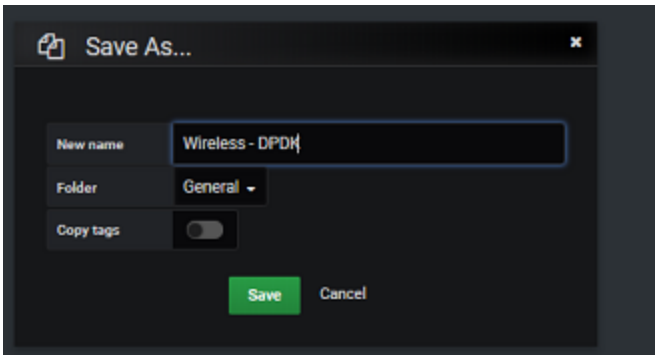
In this example, a new dashboard **Wireless-DPDK**, is created starting from the general one (**Wireless**), as follows:

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:

a. Set the name of the new dashboard to **Wireless-DPDK**.



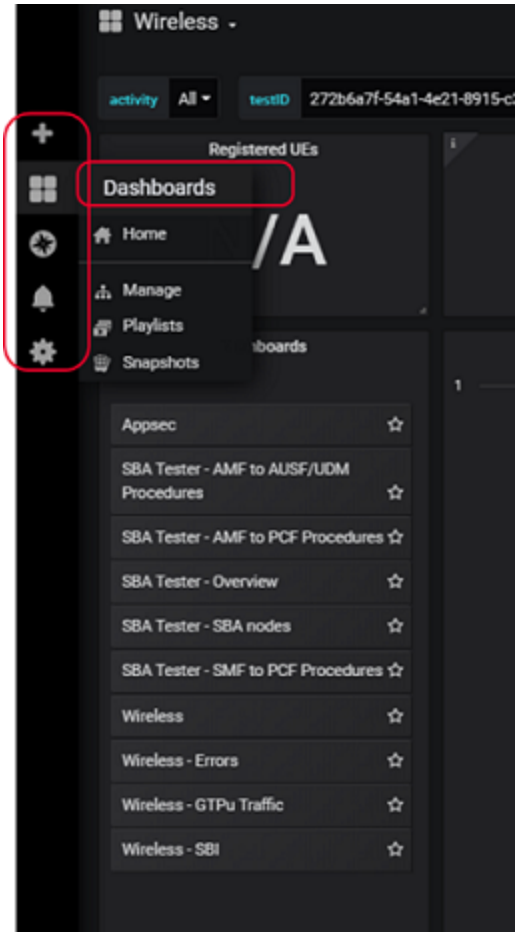
b. Select the location of the new dashboard (it is recommended to keep it under the **General** folder, to be on the same list with the default dashboards).

c. Select **Save**.

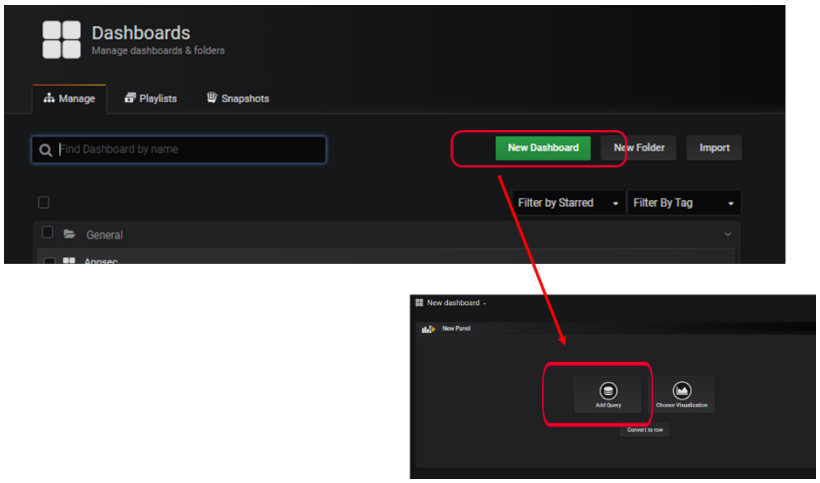
Create a new dashboard

To create a completely new dashboard, do the following:

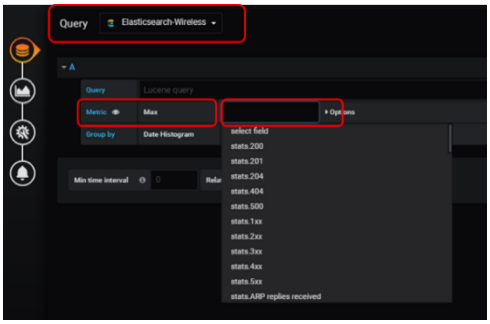
1. From the upper-left menu, select **+** icon and, then, select **Create a new dashboard**.



2. Select **New Dashboard** and, then, from the New Dashboard window, select **Add Query**.



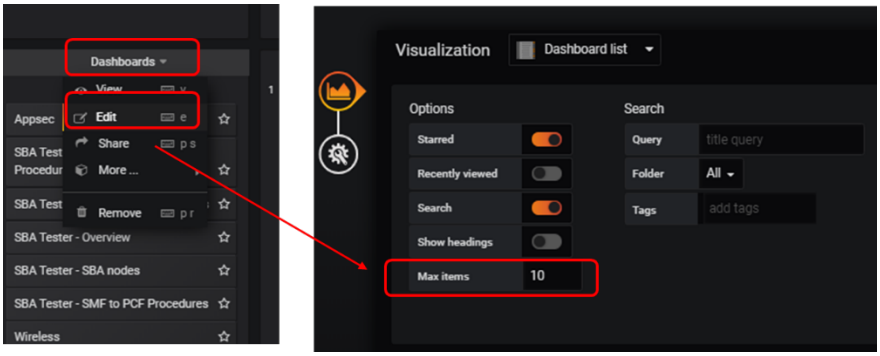
3. Configure the new dashboard:
   a. Make sure you are selecting elastic search for the query type.



   b. Select **Metric Type** : **Max**.
   c. Select the metric you want to display.
   d. 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.

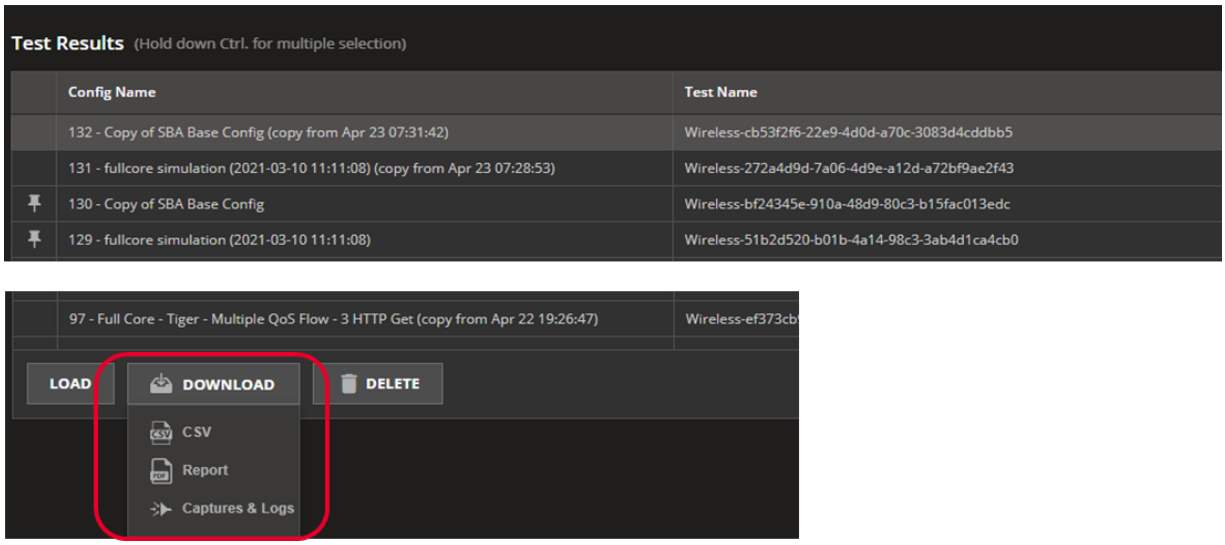
To modify the number of displayed dashboards, select **Dashboards** > **Edit** > **Visualization**, and change the value of **Max items** to increase/decrease the number of dashboard to be displayed. By default, the **Max items** value is set to 10.



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 what to download the log collection.



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
  ubectl get pods -n keysight-wap
  kubectl logs podname -n keysight-wap
  ```

  (for example: `kubectl logs agent-controller-5566944d54-w8w2k -n keysight-wap`)
  ```

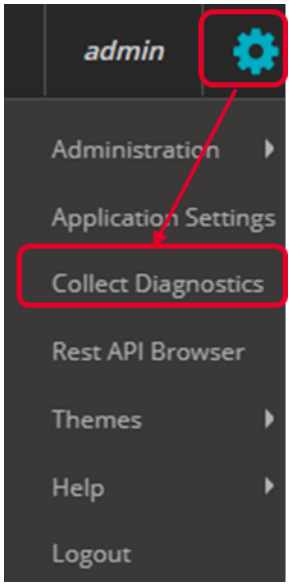
To collect diagnostics run the following command after logging in as console:

```
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

P

product support 3

T

technical support 3

www.keysight.com