LoadCore AWS Deployment

Deployment Guide

KEYSIGHT

202212211206+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.

Contacting Us

Keysight headquarters

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

Support

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

Documentation conventions

The following documentation conventions are used in this guide:

Describing interactions with the UI

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

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

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

Deprecated words

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

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

Table of Contents

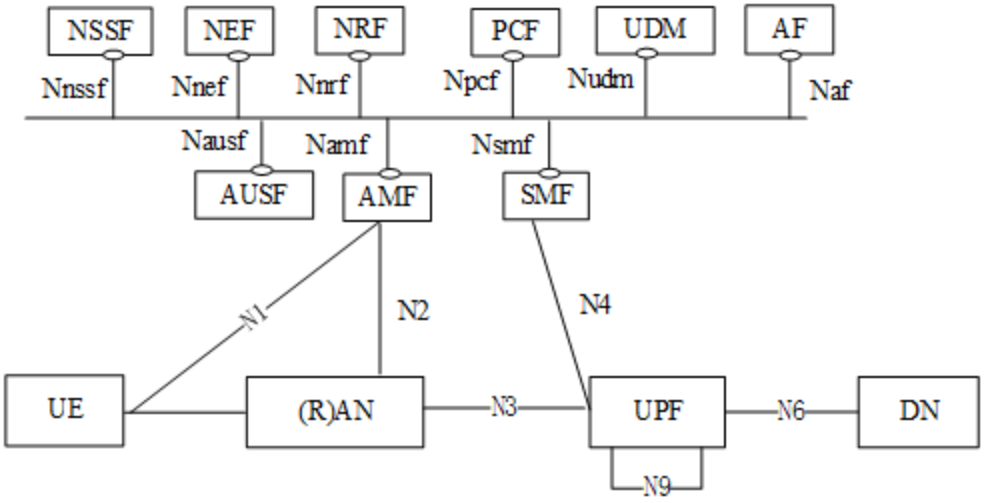
CHAPTER 1

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

Prerequisites

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

- **Wireshark** capable to decode PFCP messages and IEs (at least 2.5.2).
  This will be used to analyze the traffic captures.
- **AWS account**
  The examples presented in this guide were done using a Full Core topology deployed in B2B scenario (no real/external DUT).
- **LoadCore images** (three images: one for Middleware, for Test Agents and one for License Server).
  There are few options to get the images:
  - Get AMI images from AWS Marketplace.
    https://aws.amazon.com/marketplace/search/results?searchTerms=keysight
    Regions supported: eu-north-1, ap-south-1, eu-west-3, eu-west-2, eu-west-1, ap-northeast-2, ap-northeast-1, sa-east-1, ca-central-1, ap-southeast-1, ap-southeast-2, eu-central-1, us-east-1, us-east-2, us-west-1, us-west-2
  - Get the shared images from our AWS account. For this you need to contact the Support Eng. assigned to help with this activity.
- **AWS Cloudformation templates**
  The LoadCore setup can be also deployed using Cloudformation templates. The templates can be found at the following location:
  https://github.com/Keysight/loadcore/tree/main/AWS
  How to use Cloudformation templates:
  https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/GettingStarted.Walkthrough.html
- **Valid licenses** for Ixia License server. For deploying full 5G core and run Control Plane and User Plane traffic, the following licenses are required:
  - Control plane licenses:
    - LoadCore interface license simulation (**P89033A** x 11 pcs).

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

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

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

Resource requirements in AWS

LoadCore Middleware instance resources

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

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

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

LoadCore Test Agent instance resources

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

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

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

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

Licensing server default resources

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

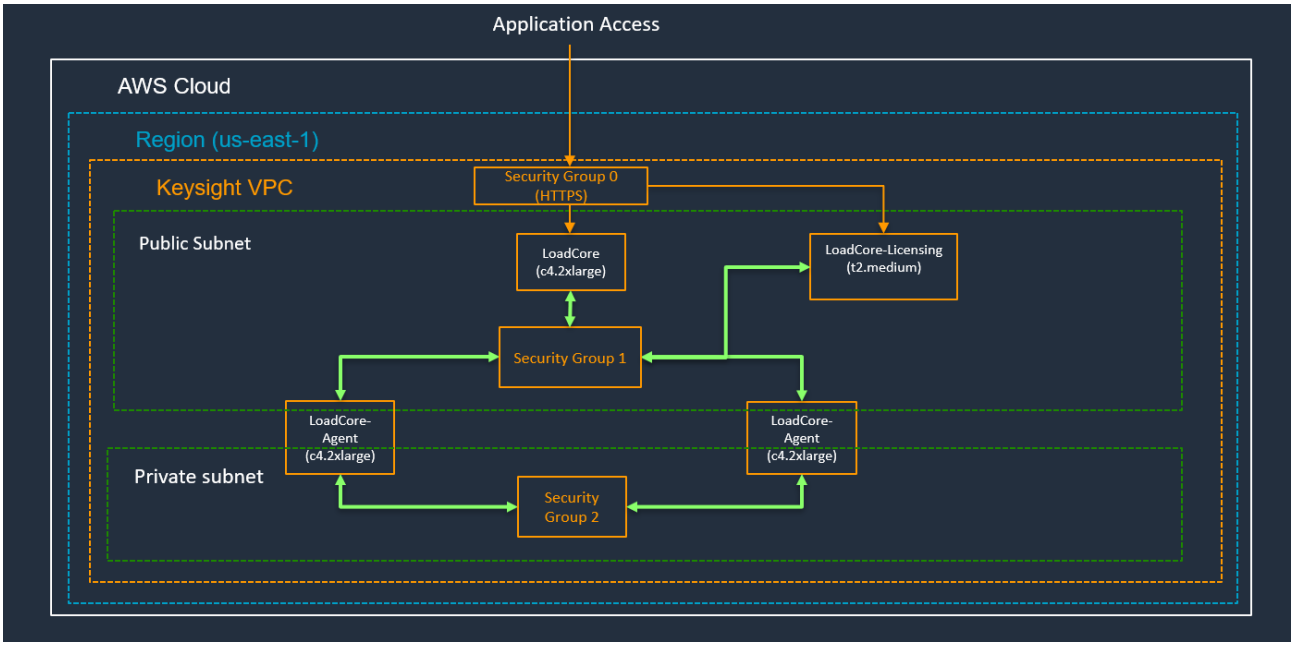
Recommended instance types: **t2.medium**, **t2.large**, **t2.xlarge**, **c4.large**, **c4.xlarge**.

CHAPTER 3

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:



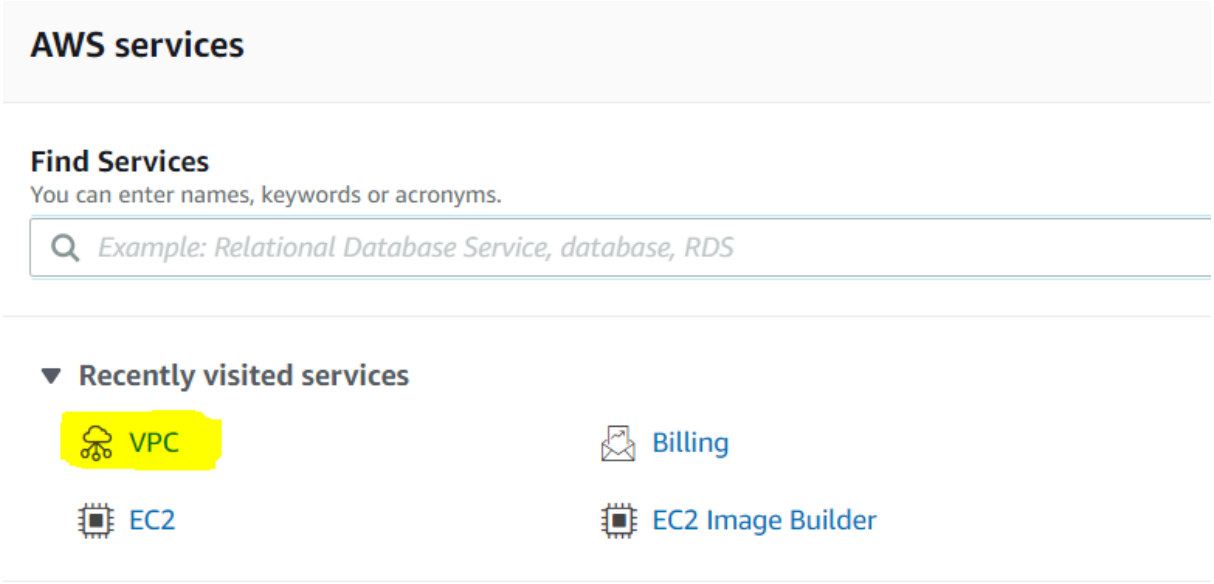
For LoadCore deployment, you can reuse an already existing VPC or create a new one.

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

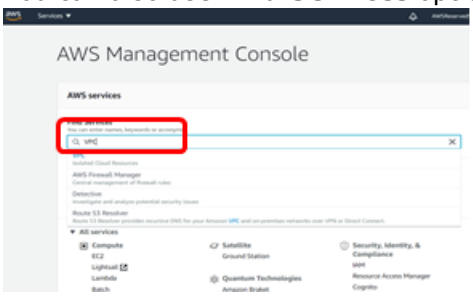
Create a Virtual Private Cloud

You can create an Amazon Virtual Private Cloud as follows:

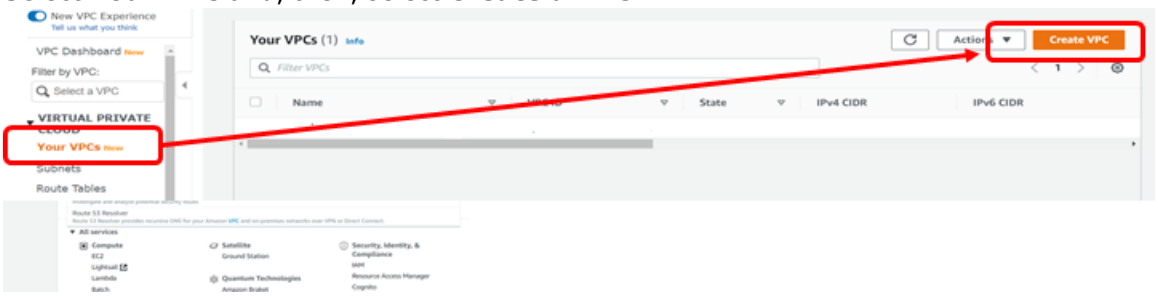
1. Select **AWS Management Console** > **Services** > **VPC**.



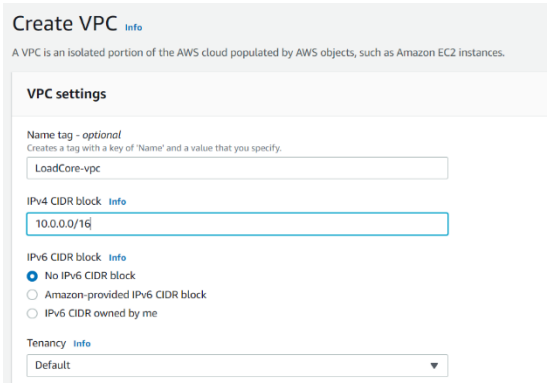
You can also use **Find Services** option and search for VPC:



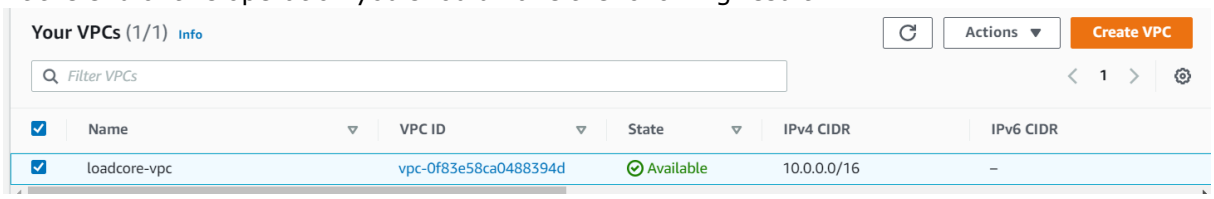
2. Select **Your VPC** and, then, select **Create a VPC**.



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:



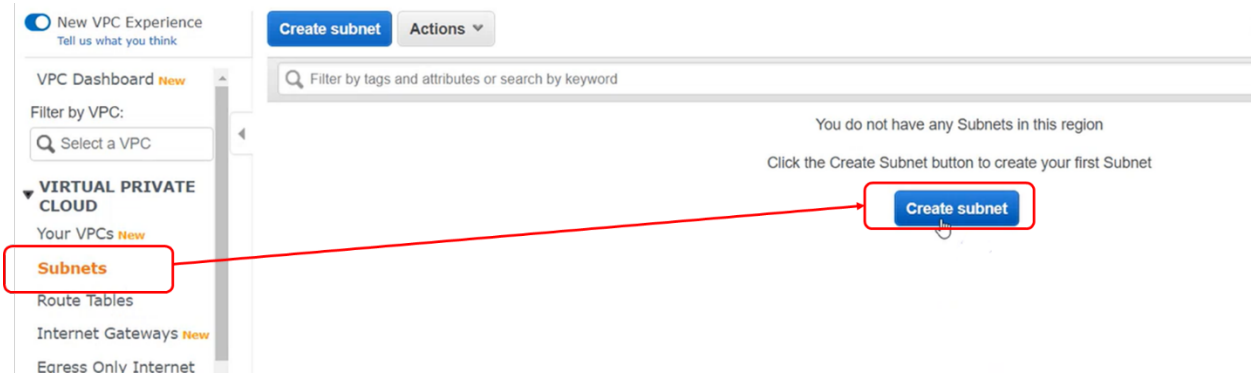
4. At the end of this operation you should have the following result:



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.

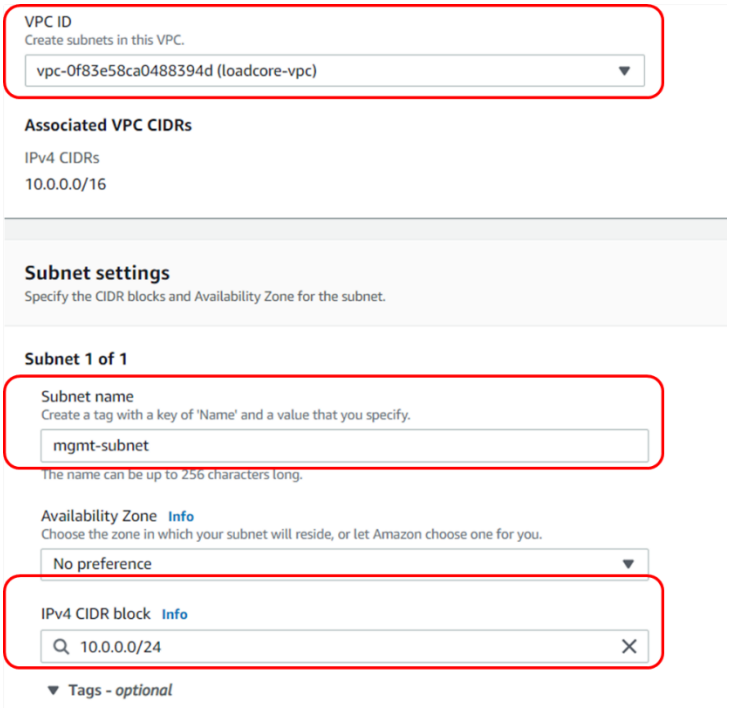


We will use **10.0.0.0/24** for management (with the name **mgmt.-subnet**) and **10.0.10.0/24** for test (with the name **test.-subnet**).

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).

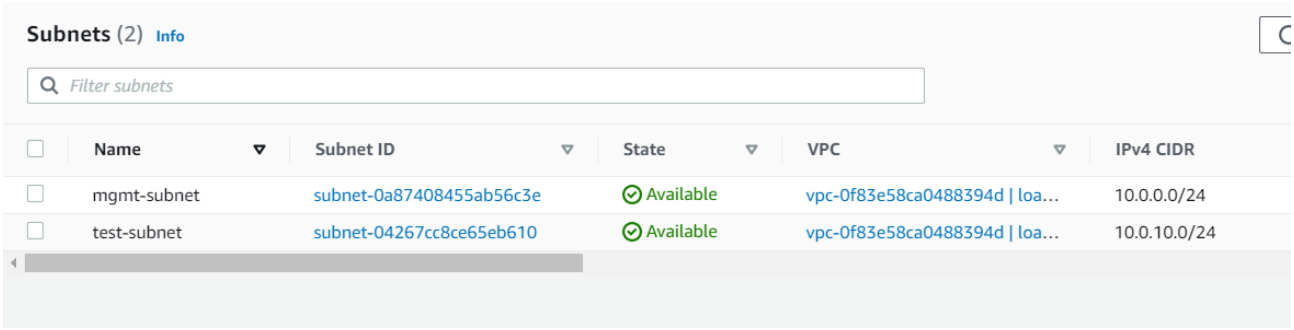
3. Define the subnet IP addresses (in this case **10.0.0.0/24**).
4. Select **Create Subnet**.



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.

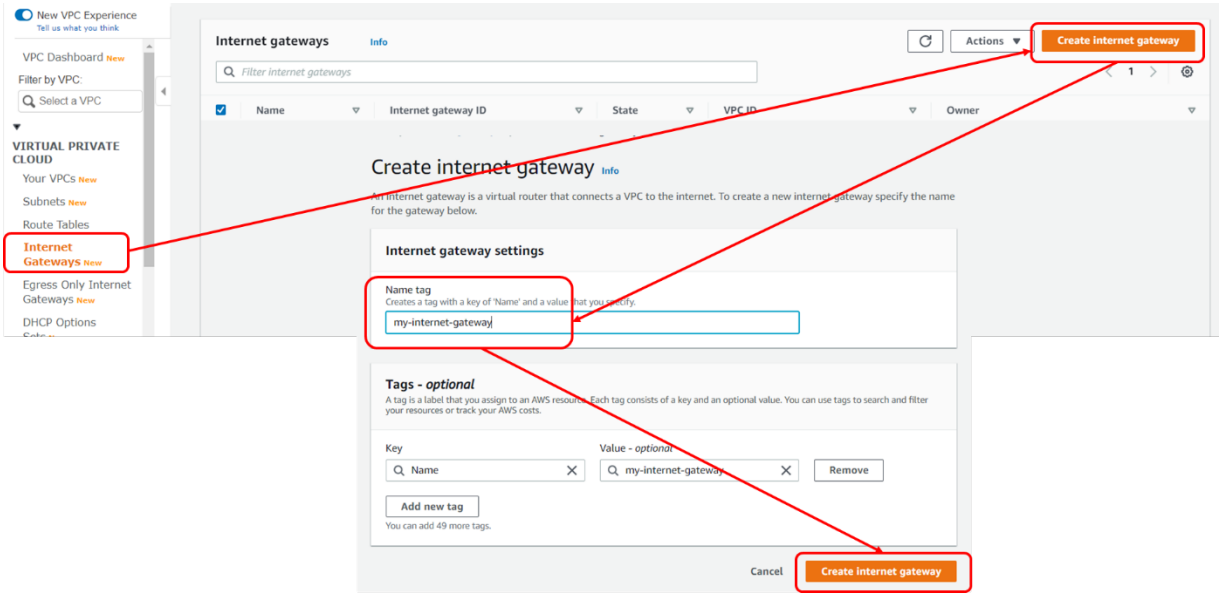


Create the Internet Gateway

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

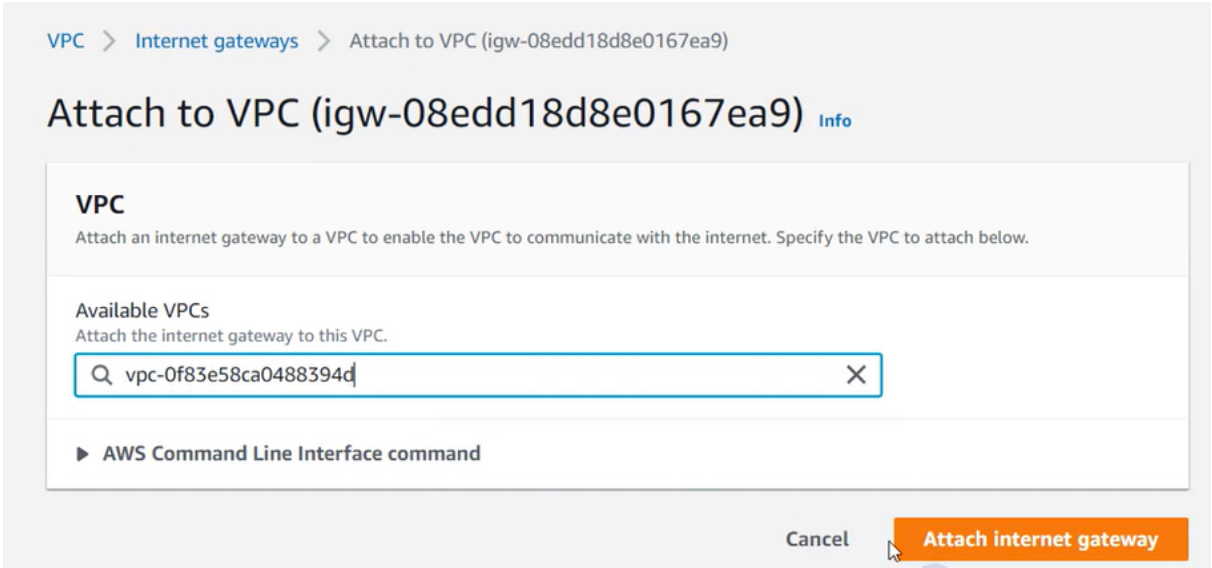
To use an internet gateway, attach it to your VPC and specify it as a target in your subnet route table for internet-routable IPv4 or IPv6 traffic. An internet gateway performs network address translation (NAT) for instances that have been assigned public IPv4 addresses.

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

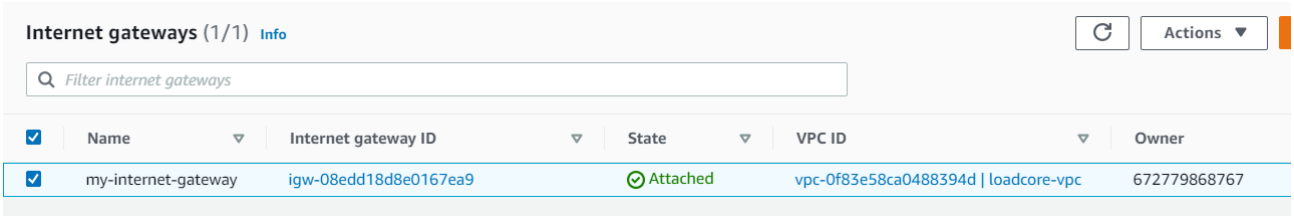


The newly created gateway is in the state **Detached**, with no VPC ID associated. Therefore, we need to attach it to our VPC:

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



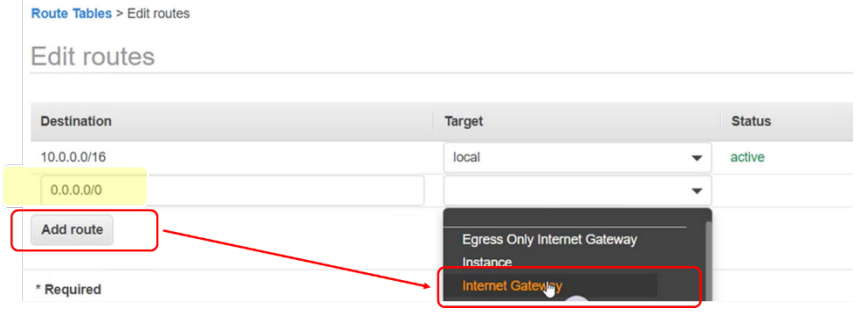
At the end of this step, the following image should be displayed:



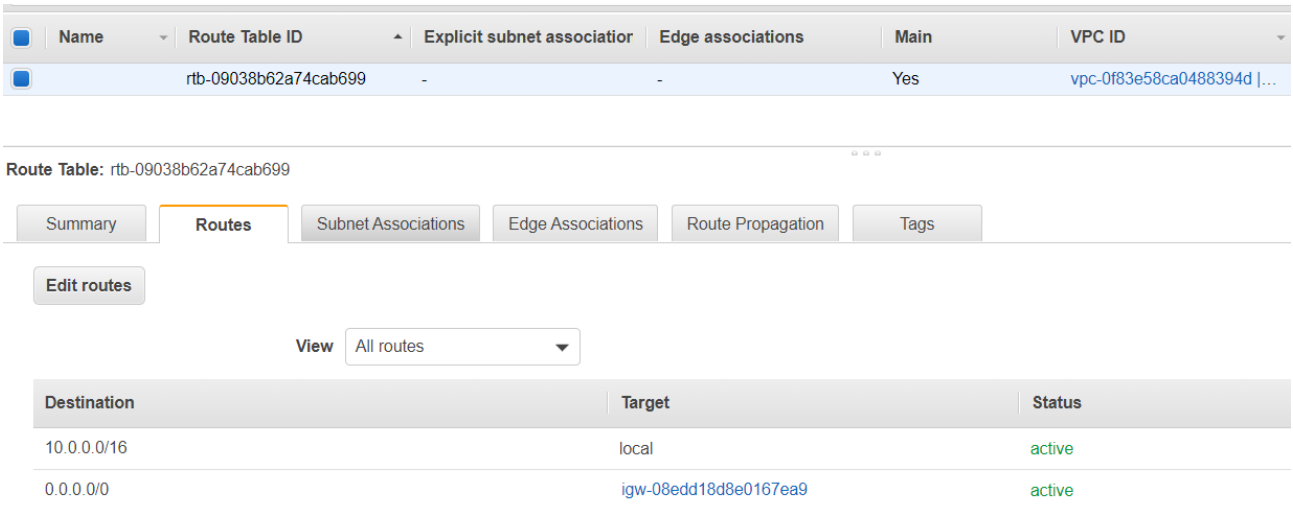
Assign traffic routes

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

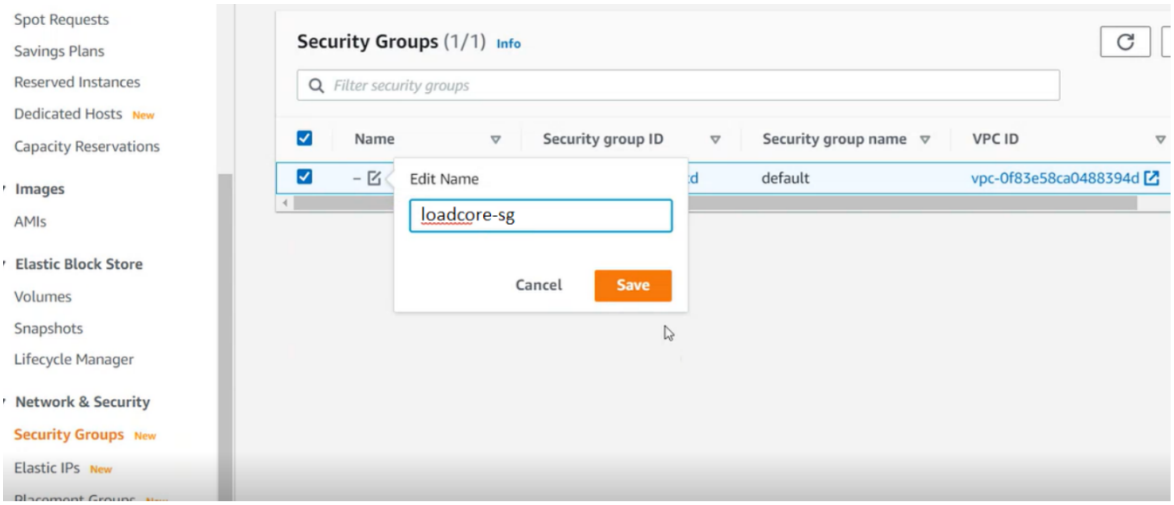
To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

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.

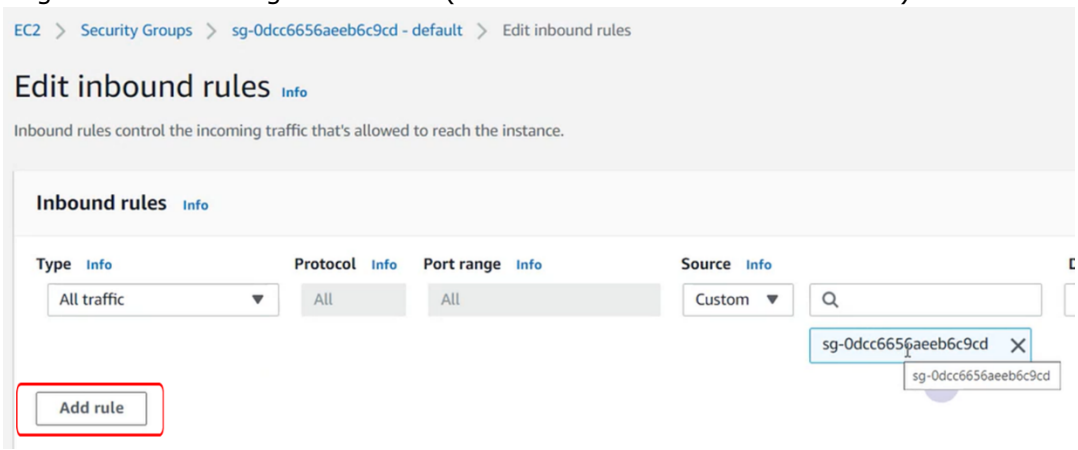
We will define a name for it (**loadcore-sg**) and configure the inbound rules:



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

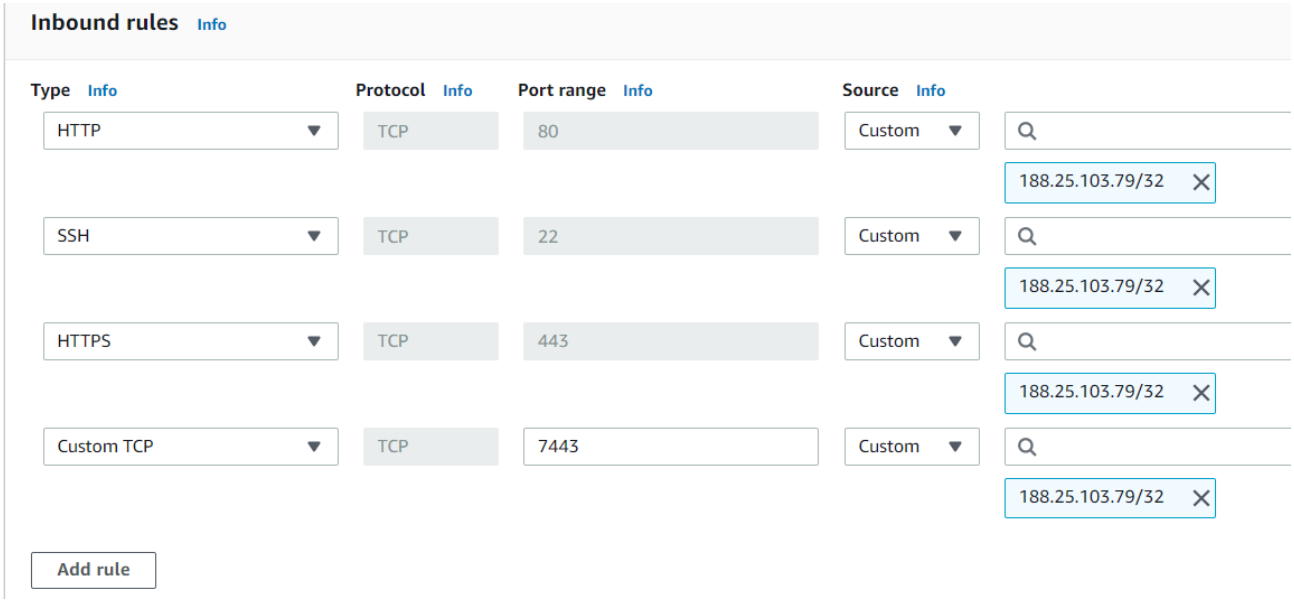
1. Select the security group ID and go on Inbound rules.
2. Select **Add rule**.

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



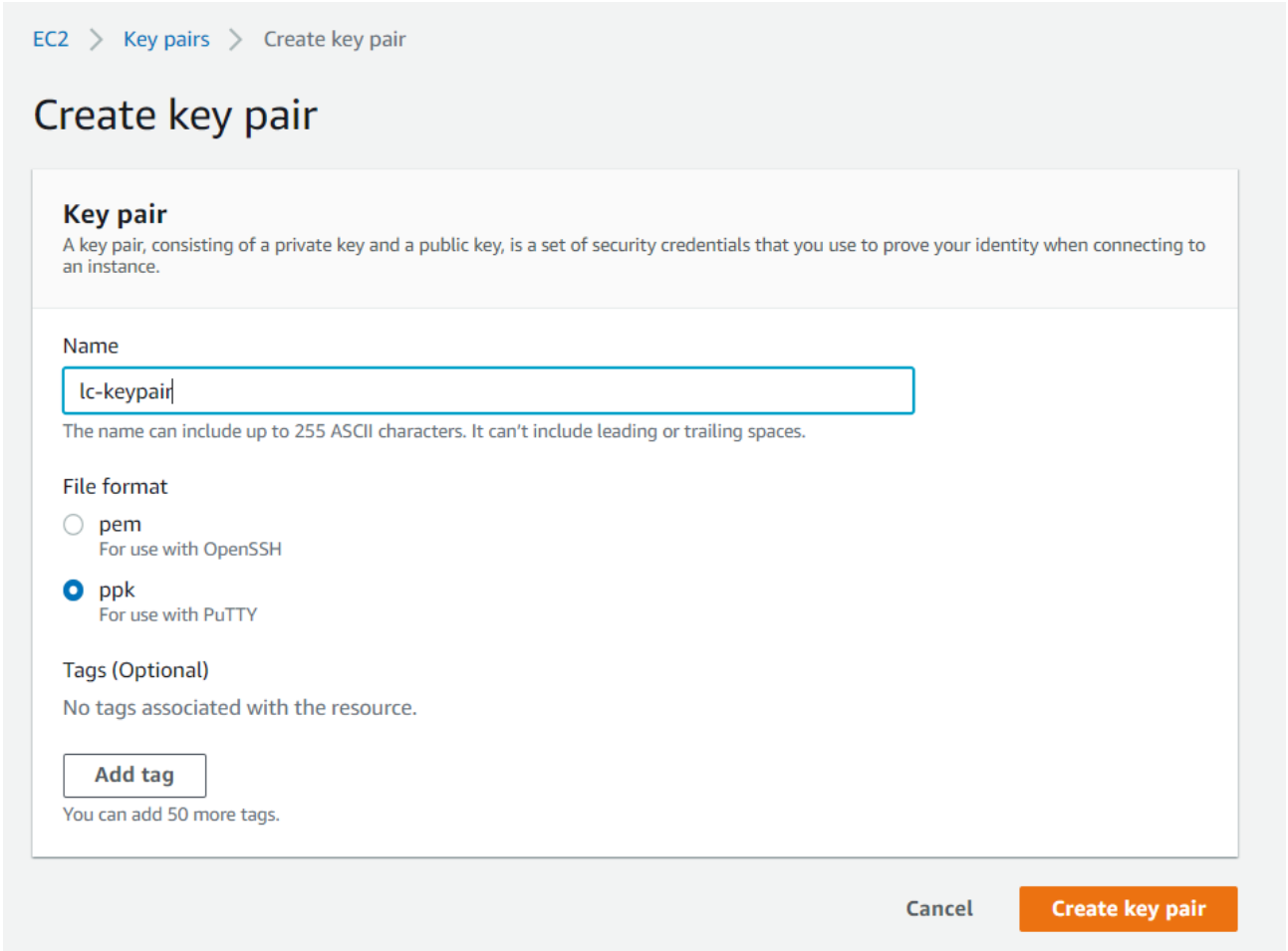
- We will permit the following:
  - SSH, to access the LoadCore VMs,
  - HTTP, HTTPS, used by LoadCore for configuration, interaction between nodes,
  - Custom: 7443 port, that is used by the LoadCore License Server,
  - For all traffic we will select *My IP*, and the browser will automatically fill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

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



Create Key Pairs

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



CHAPTER 4

LoadCore Components Installation

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

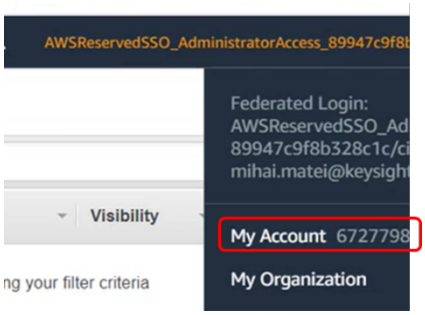
In AWS, the images that we are using are in AMI format.

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

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

In this deployment, we will use shared images.

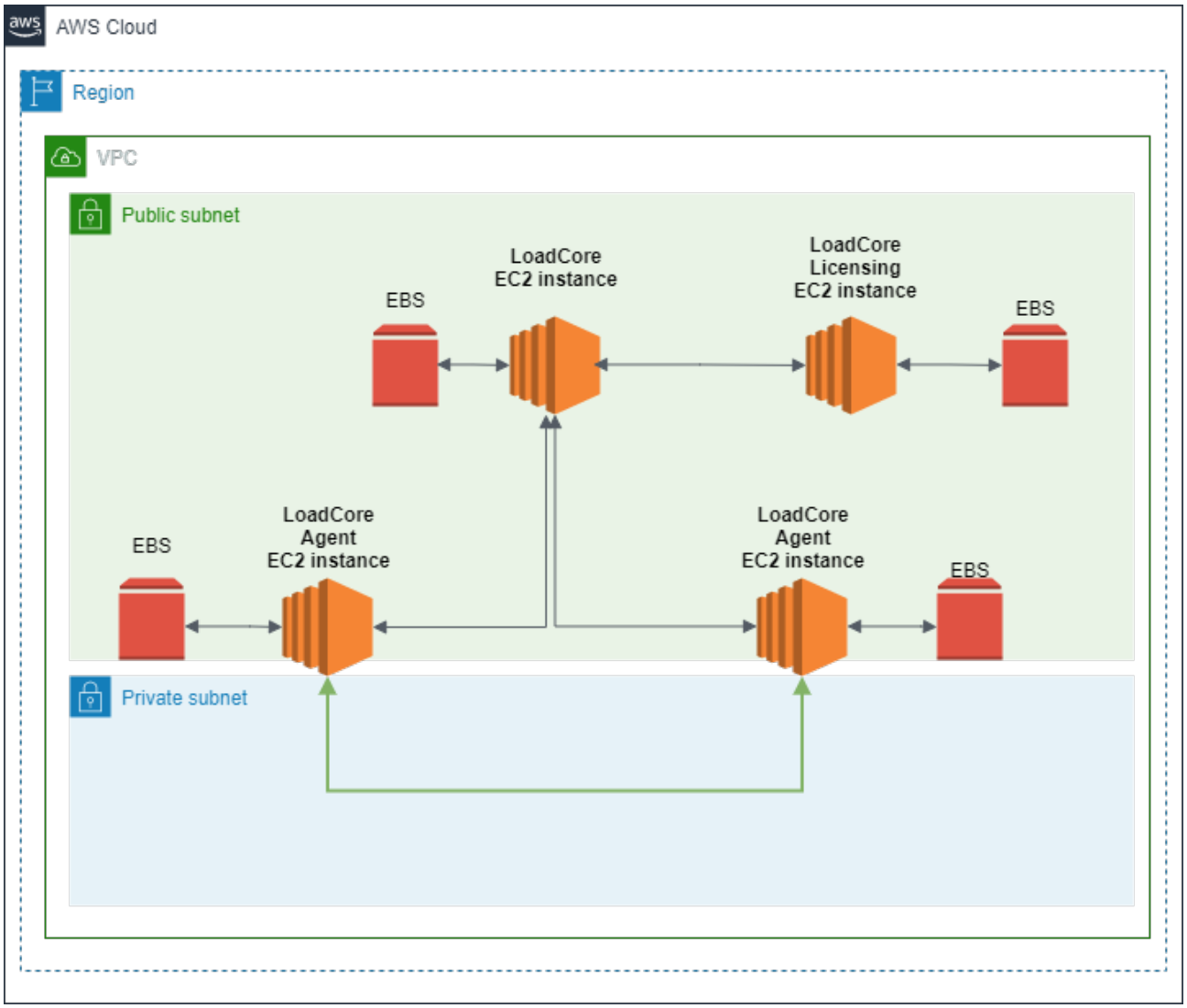
To get access to them, please contact our Support or PM team. You need to provide your AWS account ID to get access to the shared images.



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

The installation process basically consists of launching the AMI images.

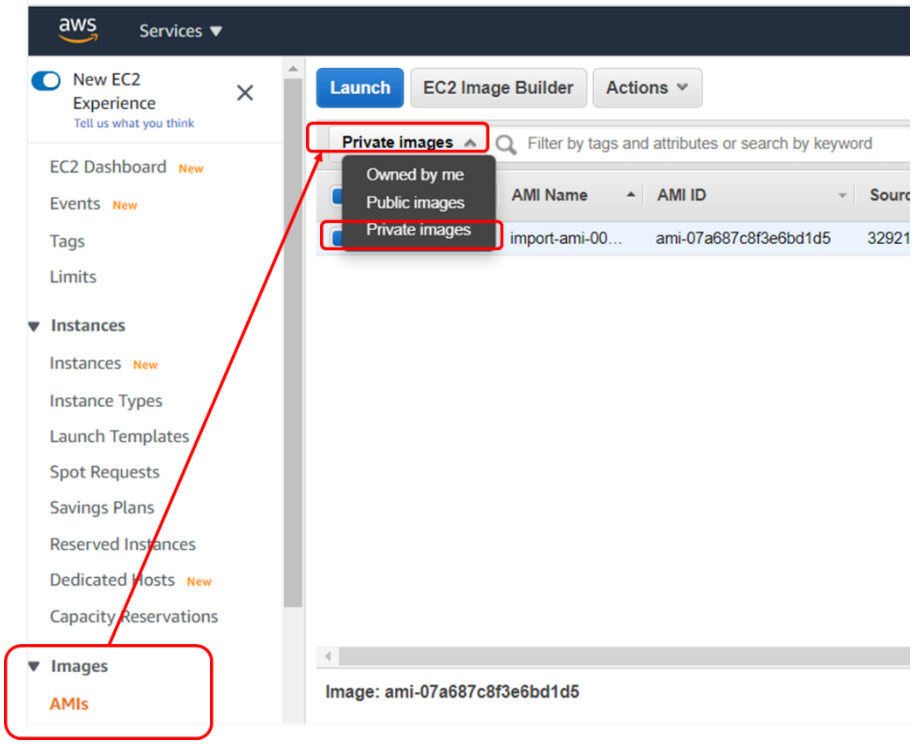
The following diagram illustrates some of the AWS services used by LoadCore deployment:



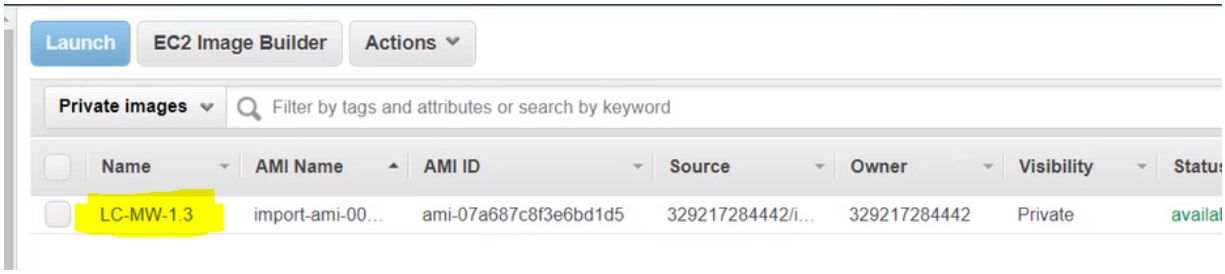
Middleware Installation

The installation procedure of the Middleware requires the following steps:

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**).

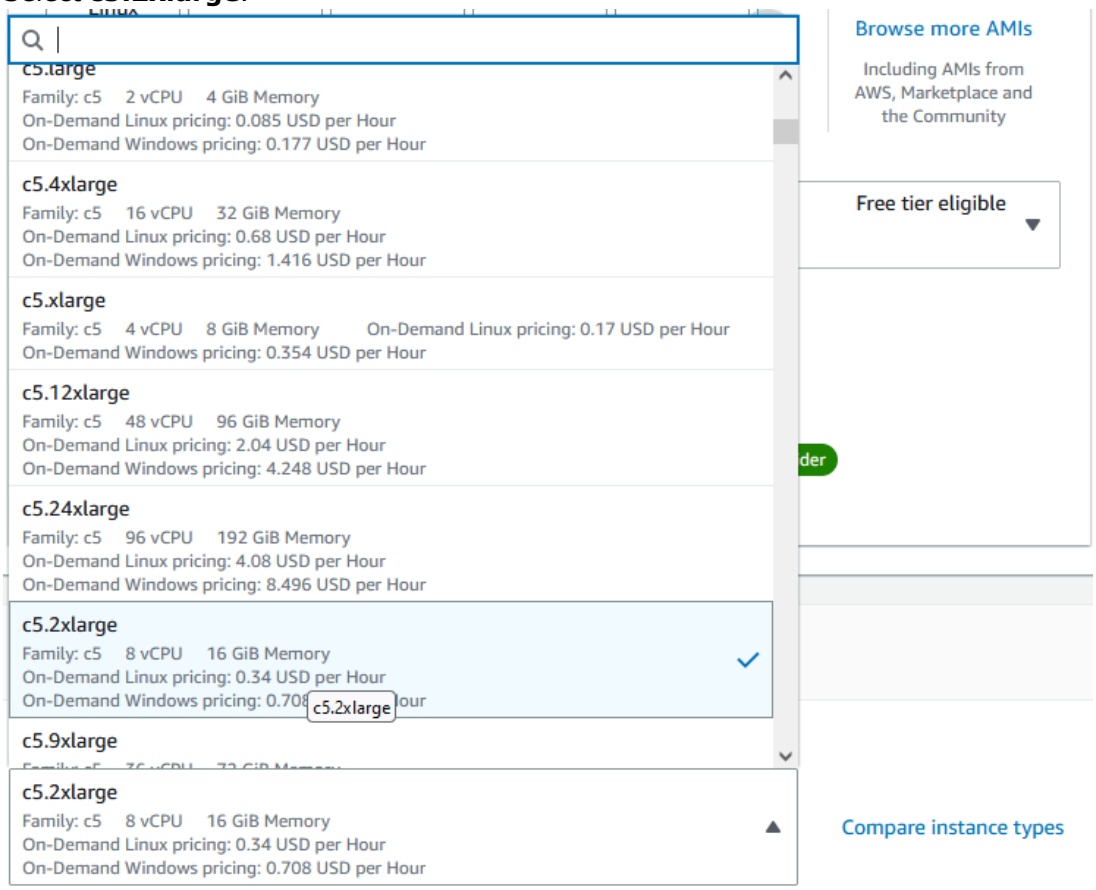


Once you have the image available, select it and select **Launch**.

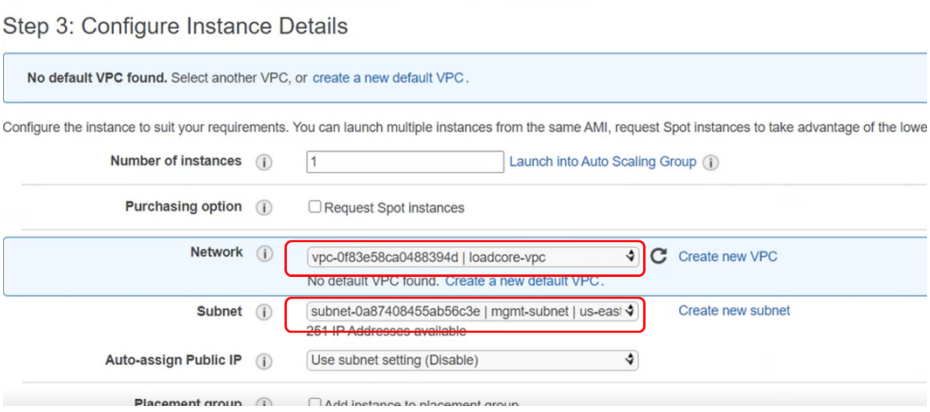
The launch process will request additional info, which is meant to adjust the resources in AWS.

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**.

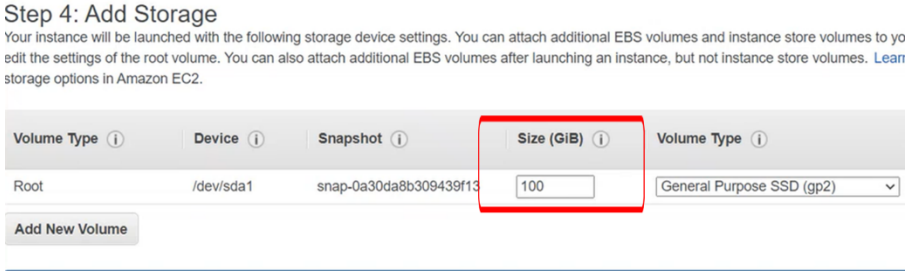


- Configure the Instance Details:
  - VPC – select the VPC we defined. In our case there is only one, but if you have multiple VPCs, you need to select the one you want to use to deploy LoadCore.
  - Select the management subnet.

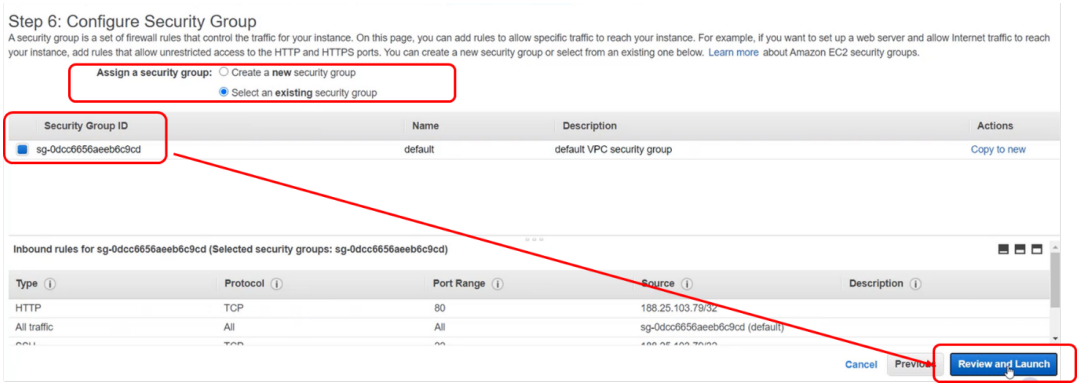


- Select **Add Storage**.
  By default, the **Size** will come with 100GB.

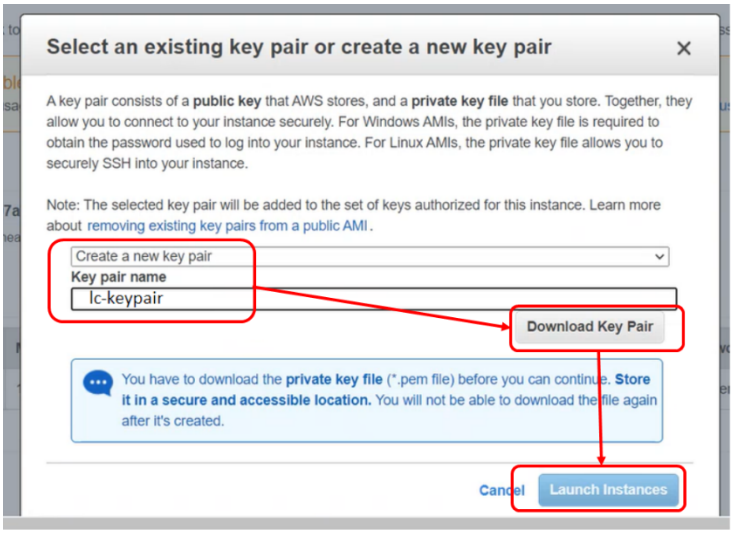
For long duration tests, you need to increase the size to 256GB.



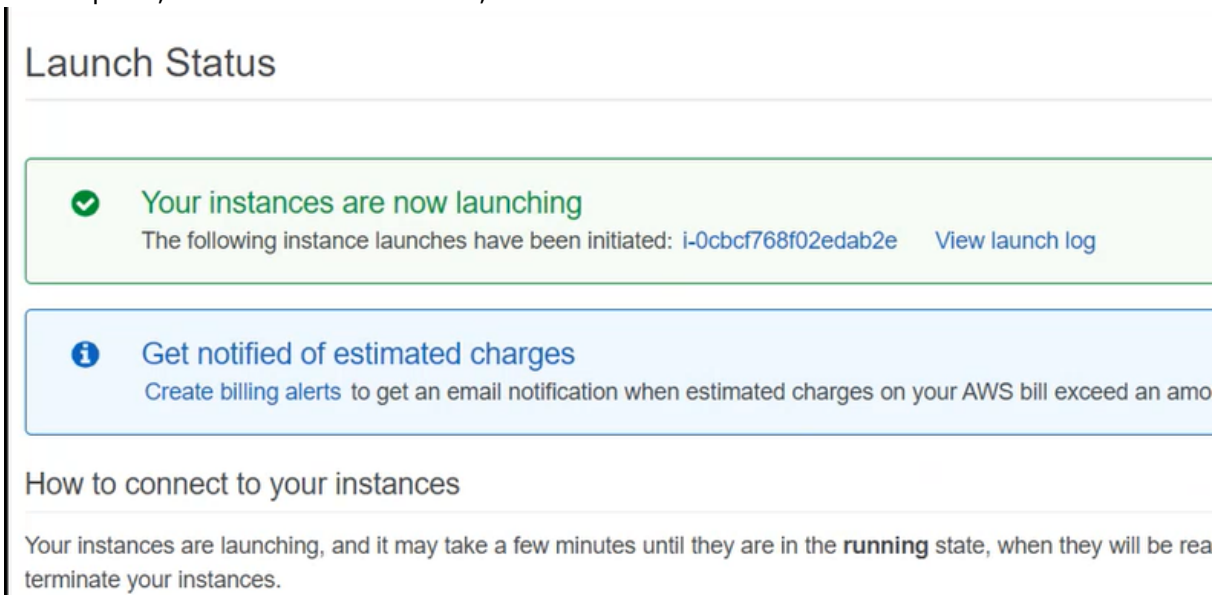
- There is no need to define Tags, select **Configure Security Group**.
  - Select the security group we previously defined.
  - Select **Review and Launch**.



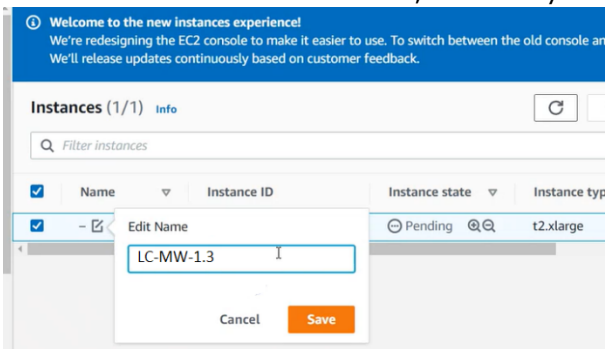
2. Select **Launch**.
   This will display a pop-up menu from where you can select/create a key pair. If you already have a key pair (and you downloaded it), you can select that one, otherwise you can create a new one, and download it from here, as shown below:
   - Select **Create a new key pair**.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



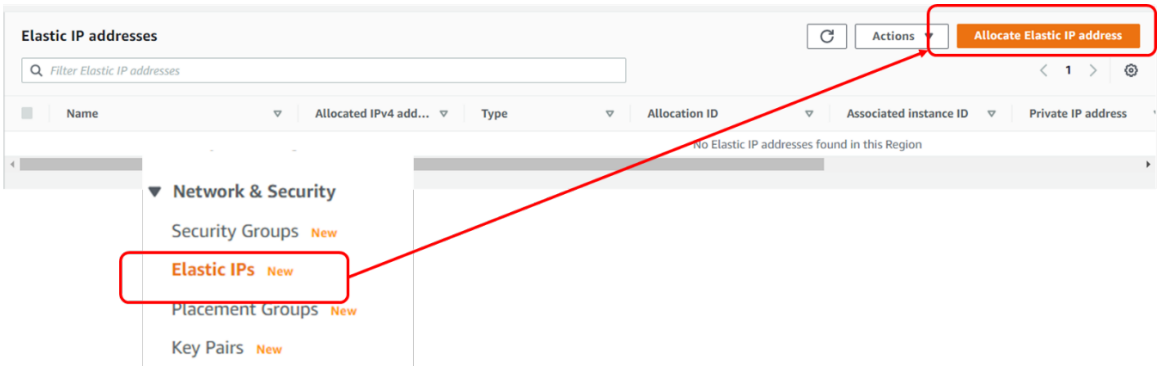
3. At this point, the instance is launched, as shown below.

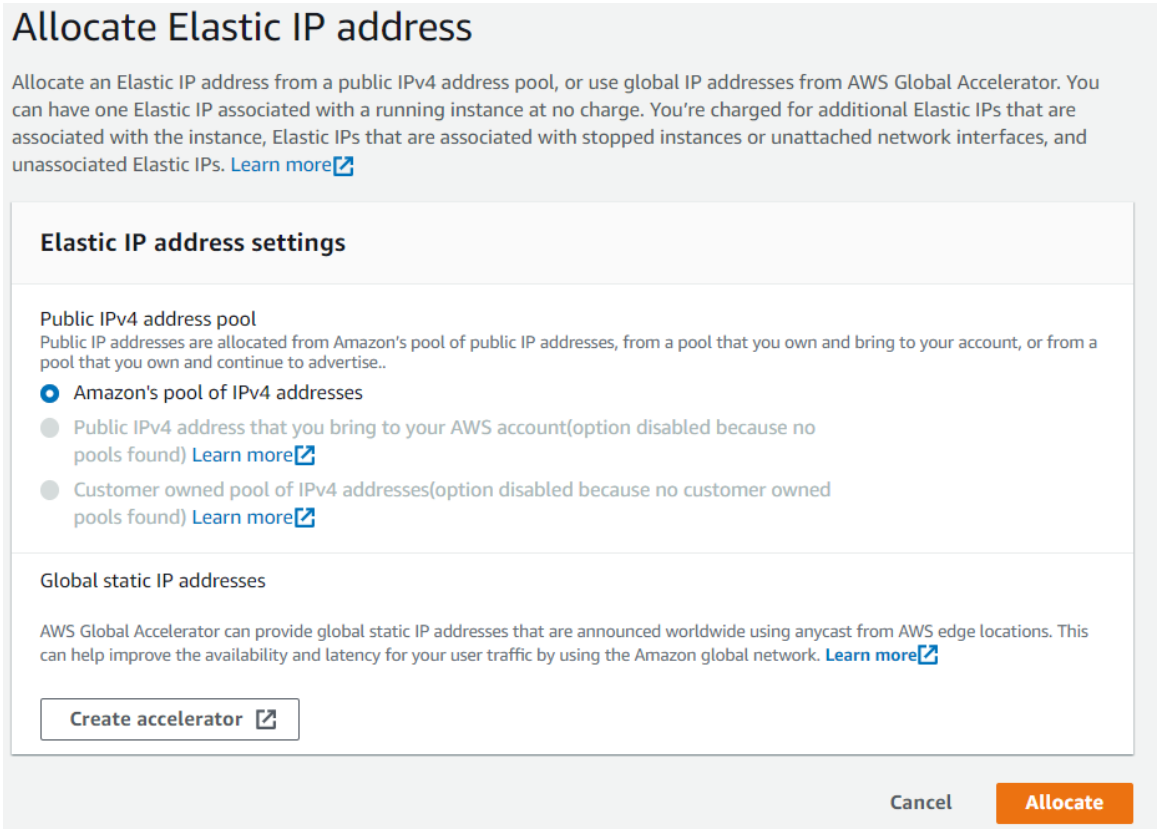


You can also define a name for it, to be easy to identify it (in this case **LC-MW-1.3**).

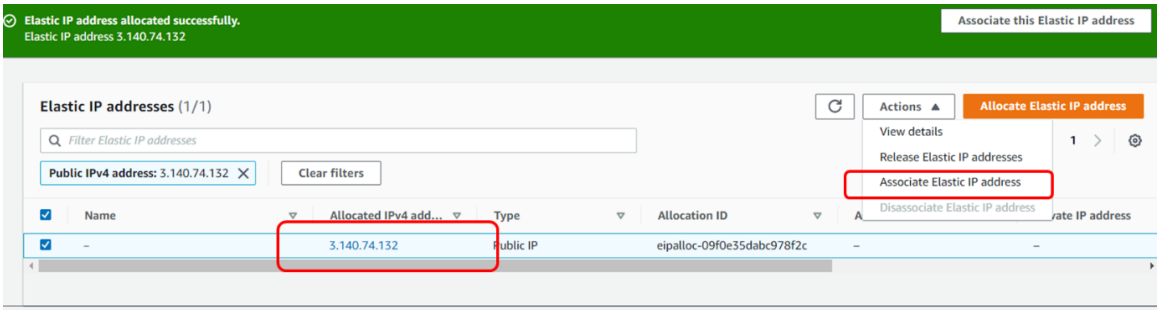


4. To access the LoadCore Middleware UI, we will need to assign an Elastic IP.
   To do this, select the Elastic IP menu:
   - Select **Allocate Elastic IP Address**.

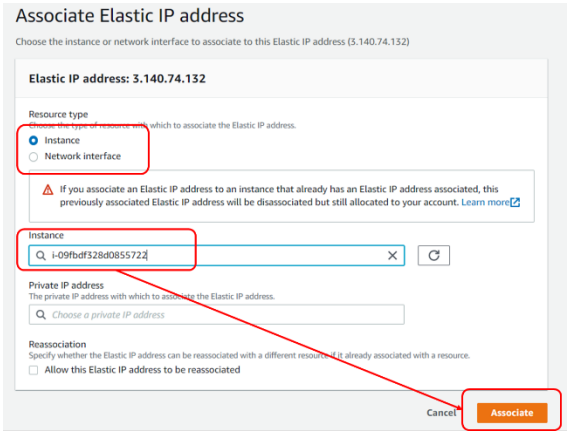




- The new IPv4 Public IP address will be allocated from AWS pool.
- Associate this IP address with the Middleware instance.



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



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

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

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

Agent(s) Installation

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

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

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

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

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

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

If Application traffic is needed, we 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

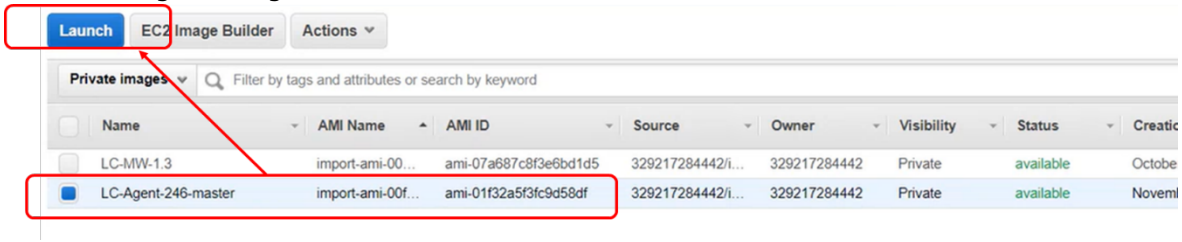
The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

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**.

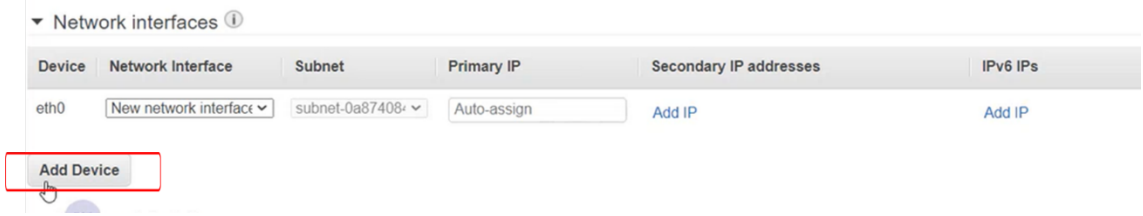


   Select **t2.xlarge** for instance type:

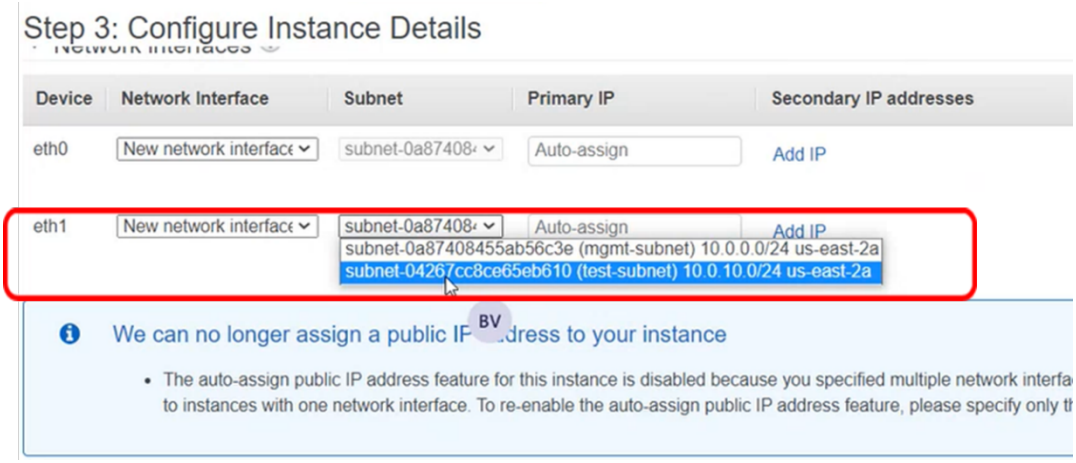
   Select **Next**.

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**.

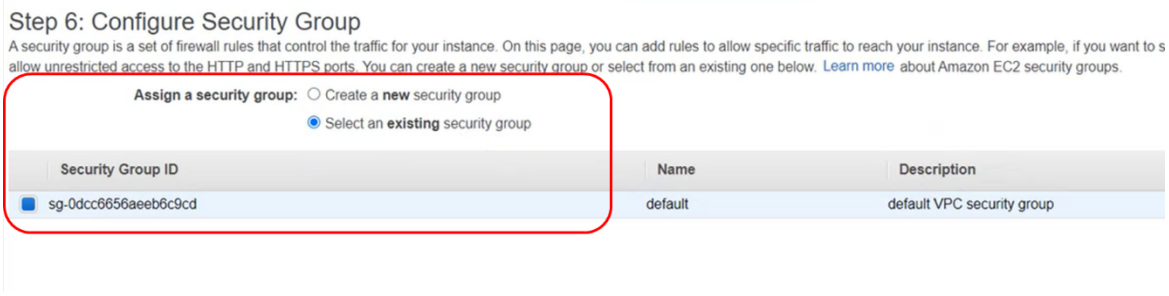


- Assign it to the test subnet (previously defined).



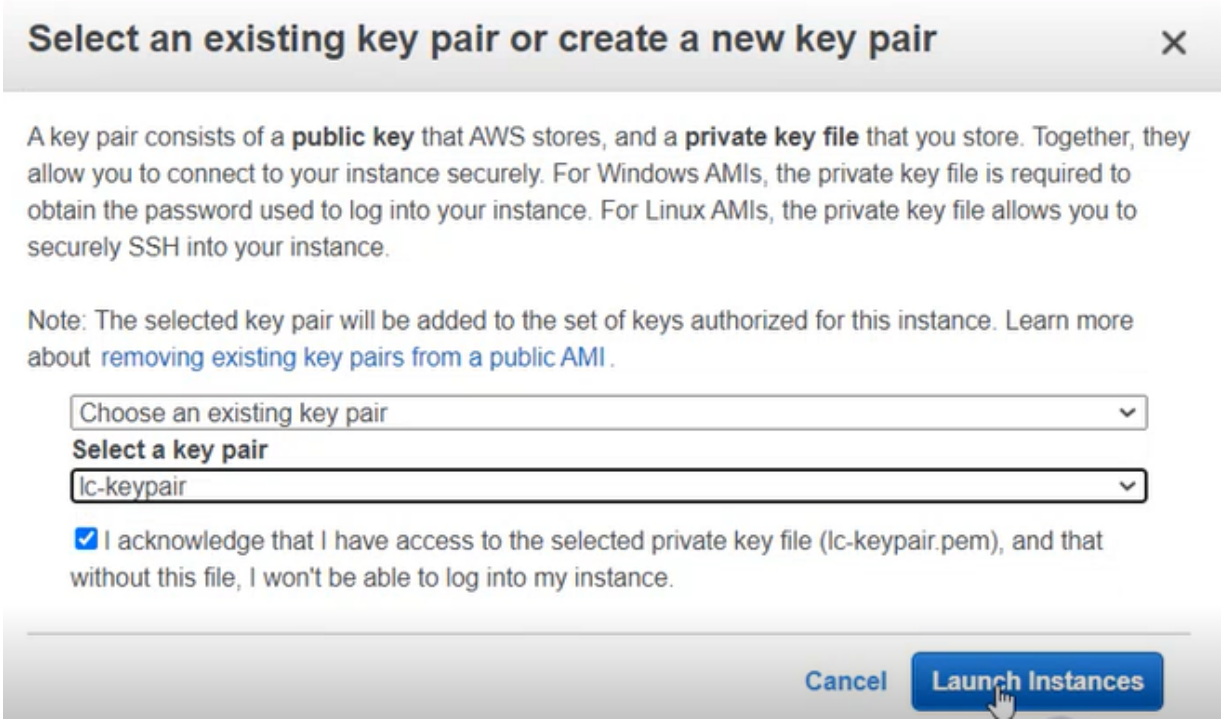
Select **Next**.

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.

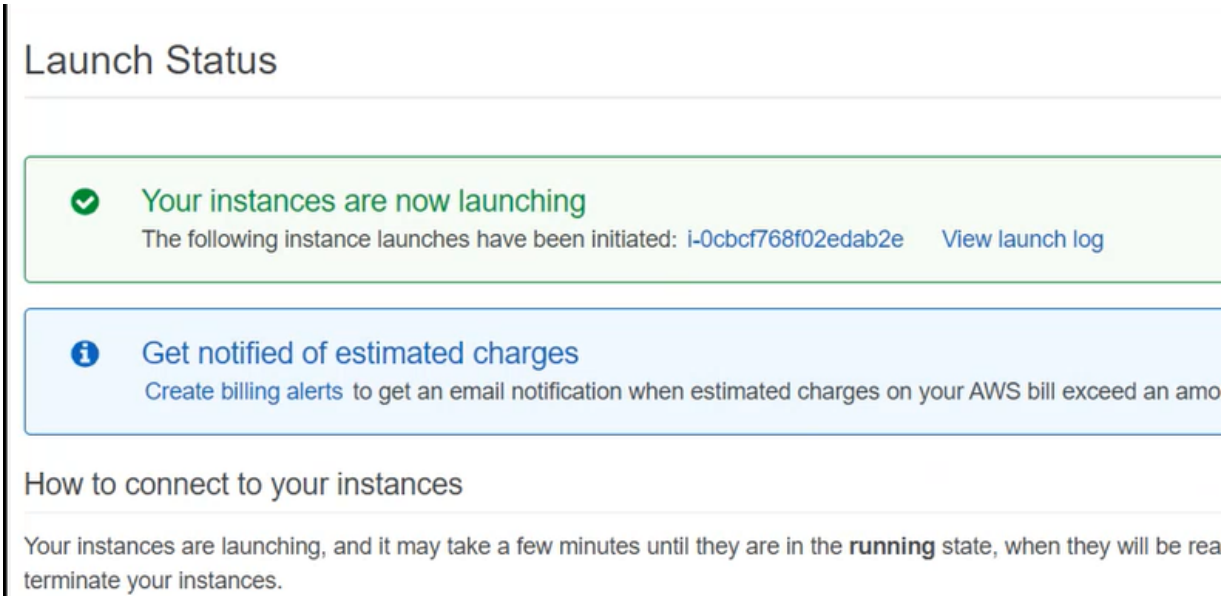


6. Select **Review** and **Launch** > **Launch**.

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

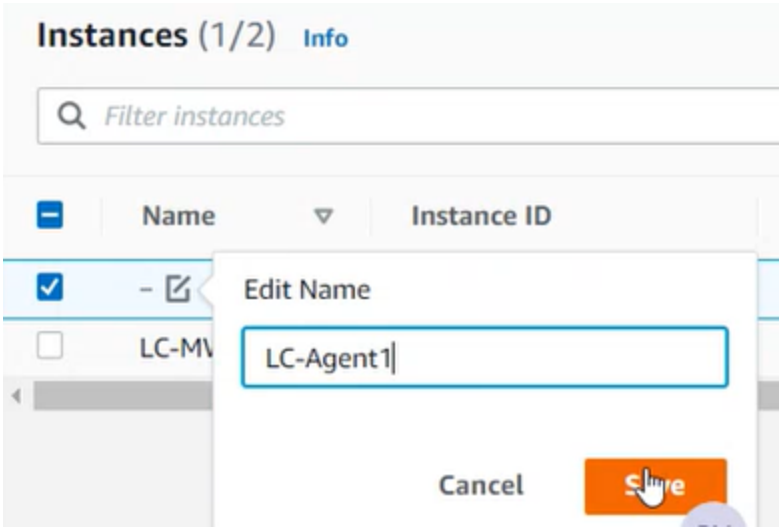


8. Select **Launch Instances**.



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

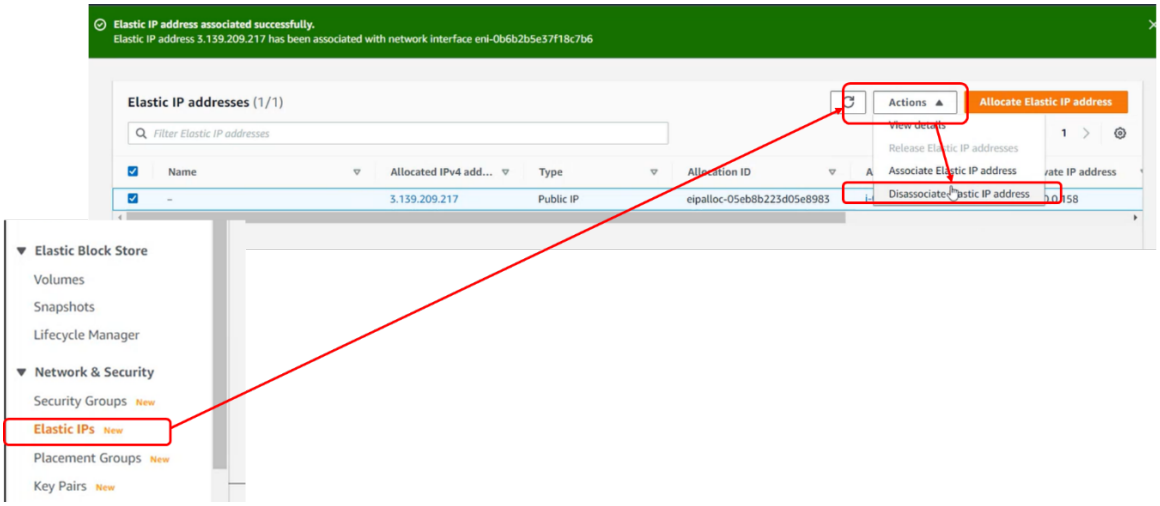
   Define a name for the test agent instance (by default, there is no name).
   In this example we will name it **LC-Agent1**.

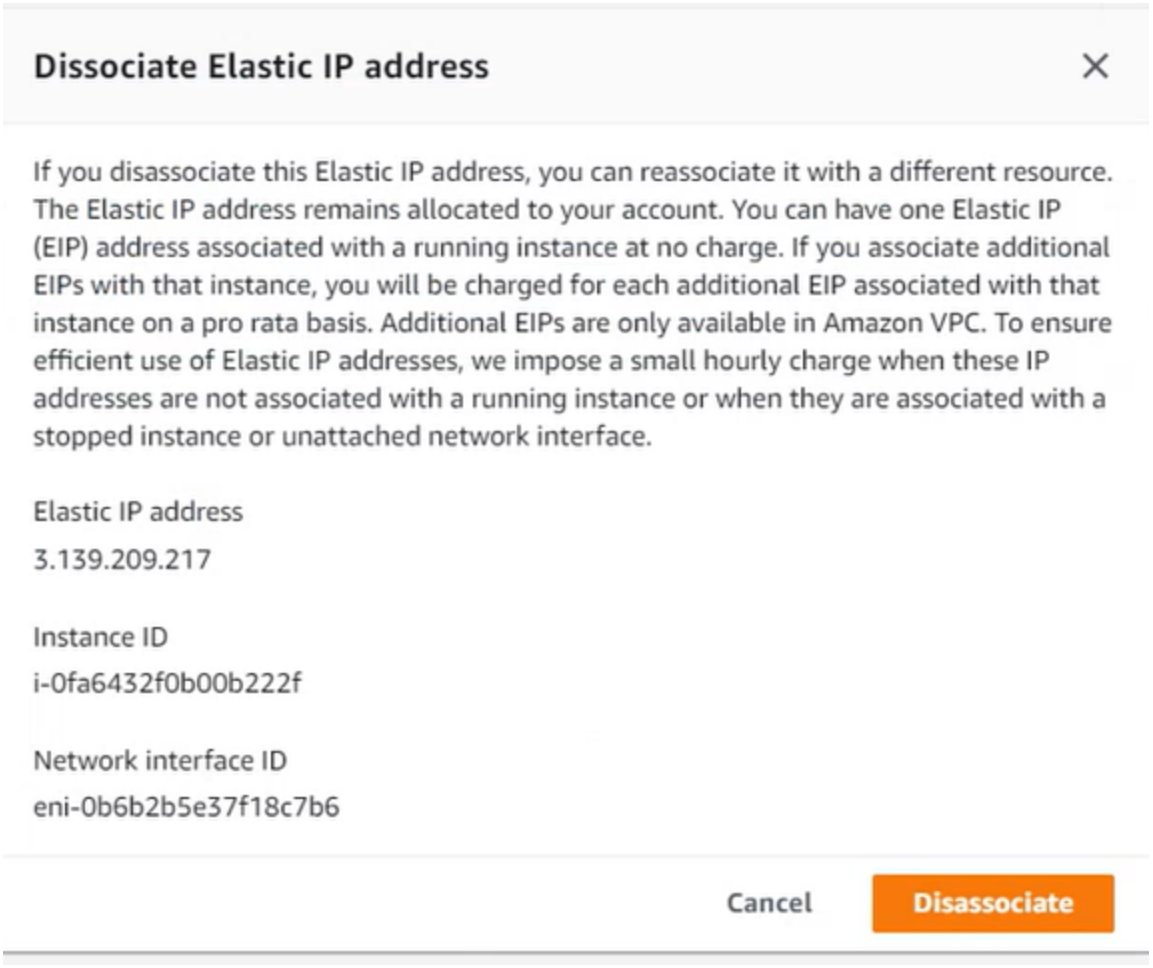


10. In order to access the agent, we need to assign an elastic IP.

    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**.





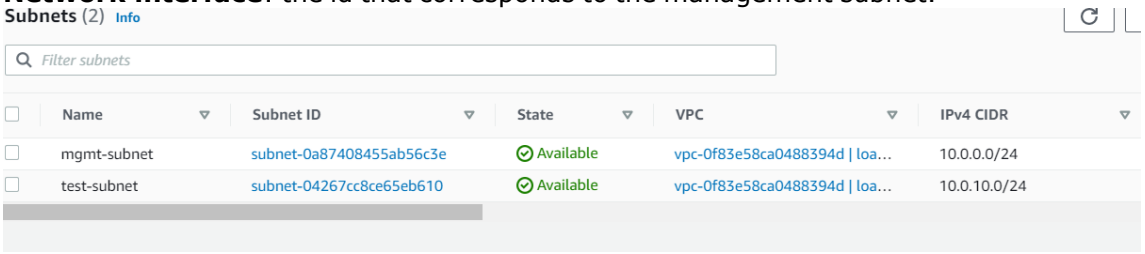
- The Elastic IP was disassociated and now should be reassociation to the Management interface of the test agent.

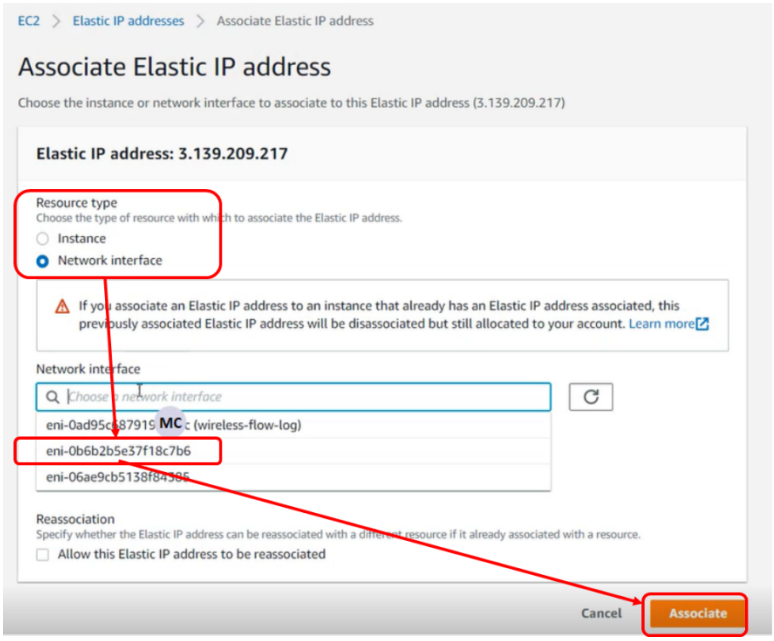
In the previous image deployment we associated the IP address to the Middleware instance, but here, since we have two network interfaces, we need to associate the Elastic IP address to the management network interface.

Select:

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

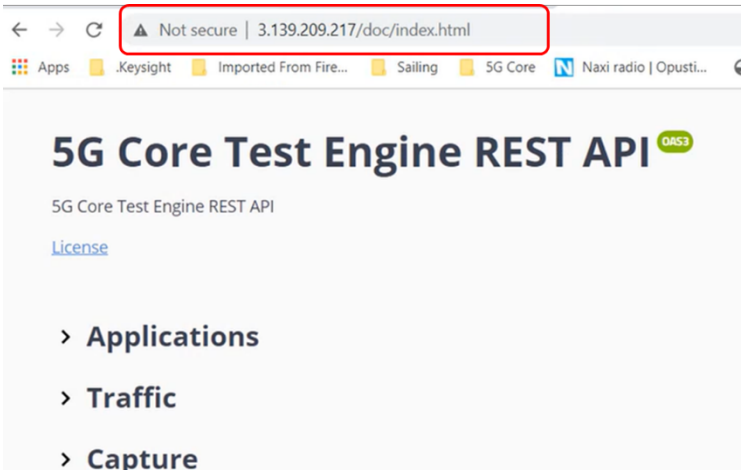
  Network interface: the id that corresponds to the management subnet.

- 



- **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` – this is used for LoadCore 2.1
    - For Middleware IP address, you will specify the private IP address that is assigned for the Middleware instance, from management subnet.

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

12. This completes test Agent deployment procedure.
    You can repeat the above steps if multiple Agents should be deployed.
    In this moment, the test setup can perform stateless UDP testing.
    For application traffic, the agent instance type should be changed, and raw sockets should be configured (this is covered in the next chapter).

LoadCore Agent configuration for Application Traffic

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

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

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

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

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

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

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

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

For other information regarding the Intel 82599 VF interface, refer to:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html

AWS CLI Installation guidelines

The full details can be found on AWS support page:
https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html

Download and run the MSI installer:
https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html

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

Use SSO to connect AWS CLI to your account

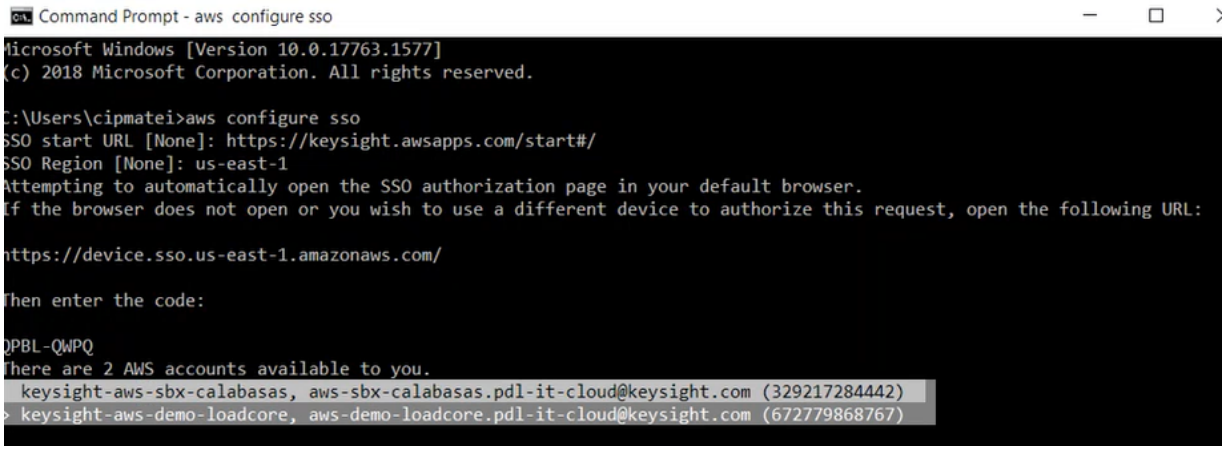
In order to use SSO to connect AWS CLI to your account, AWS SSO needs to be enabled on your account. For more details, refer to:
https://docs.aws.amazon.com/singlesignon/latest/userguide/step1.html

Open a CLI terminal and configure SSO, as follows:

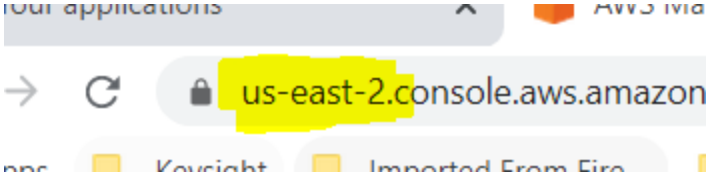
1. **AWS configure SSO**
2. SSO start URL: <**your SSO link**> (e.g. **https://keysight.awsapps.com/start**)
   You need to replace <**your SSO link**> with the SSO link used by your organization to connect to AWS.
3. SSO Region: <**SSO instance region**> (e.g. **us-east-1**)
   You need to check the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

   NOTE If an *Invalid grant provided* error is displayed, it usually means that you are using an incorrect SSO Region. For more details, refer to: https://github.com/aws/aws-cli/issues/5058.
4. Press **Enter**. The browser will automatically open and drives you to AWS SSO:

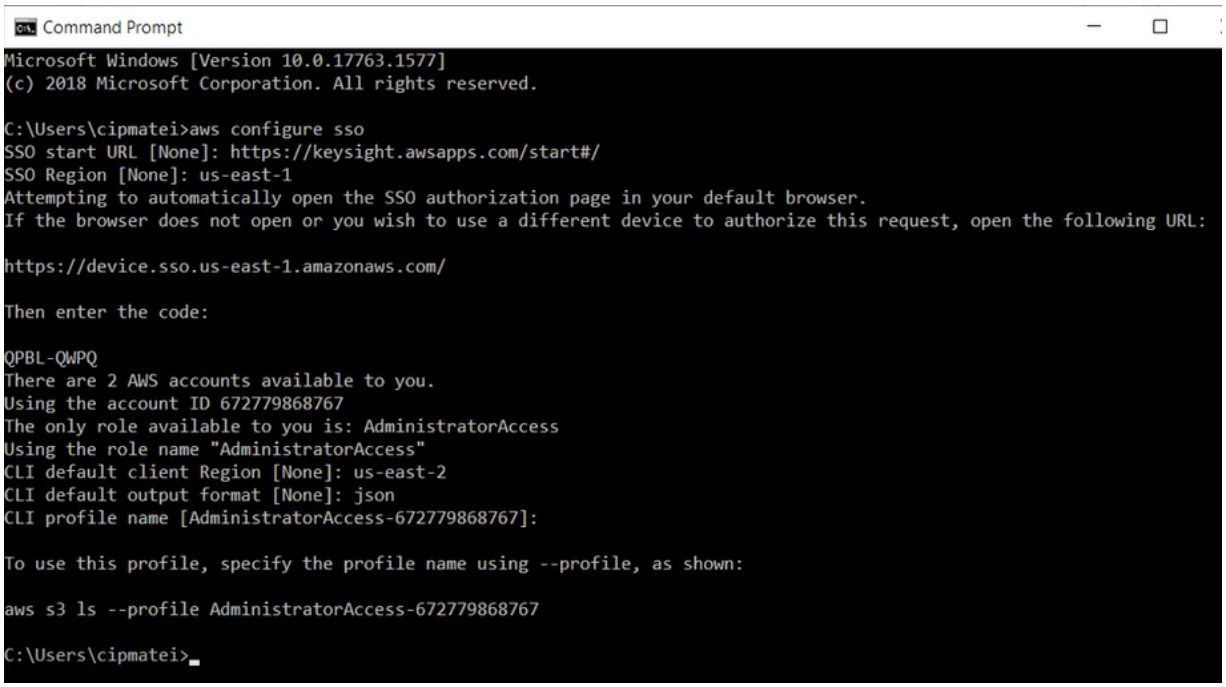
5. Select **Sign to AWS CLI**.
6. On the terminal CLI you will see the available accounts. Select the one you are using:



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

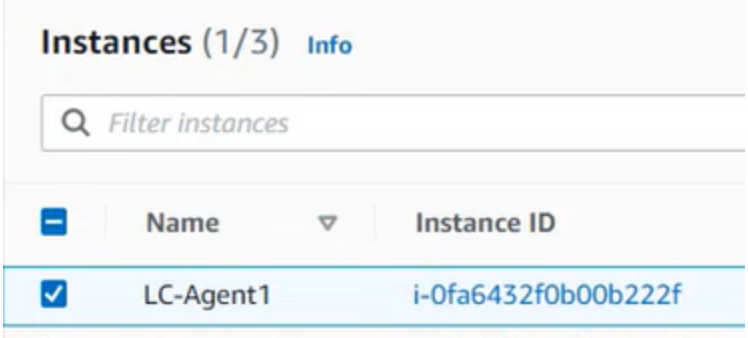
For more information on how AWS credentials work, refer to:
https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html

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

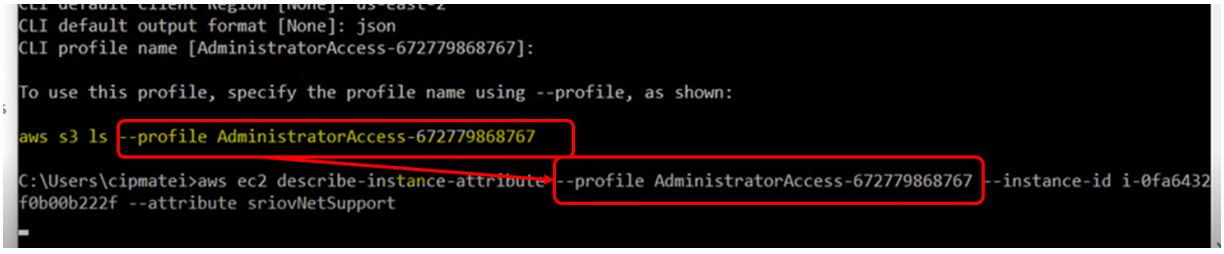
Configure SRIOV support

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).

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

C:\Users\cipmatei>aws ec2 describe-instance-attribute --profile Adminis
f0b00b222f --attribute sriovNetSupport
{
    "InstanceId": "i-0fa6432f0b00b222f",
    "SriovNetSupport": {}
}
```

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

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

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

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

  Stop the test agent instance.

  In CLI type the below command:

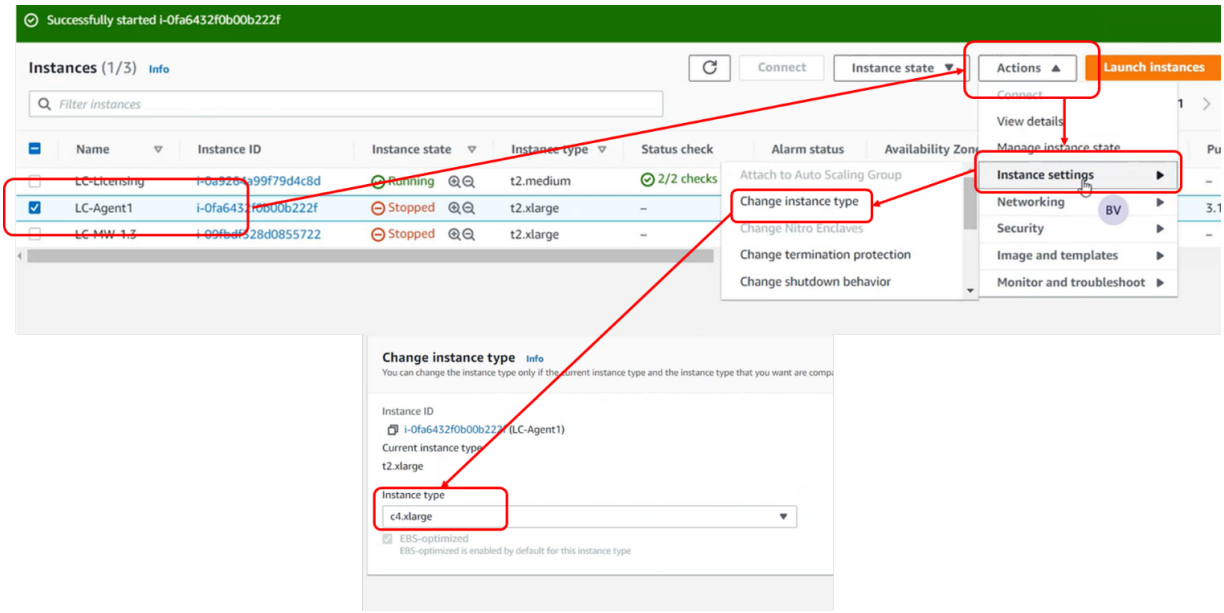
  `aws ec2 modify-instance-attribute --instance-id` **`instance_id`** `--sriov-net-support simple`

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

```
C:\Users\cipmatei>aws ec2 modify-instance-attribute --profile AdministratorAccess-672779868767 --instance-id i-0fa6432f0
b00b222f --sriov-net-support simple
```

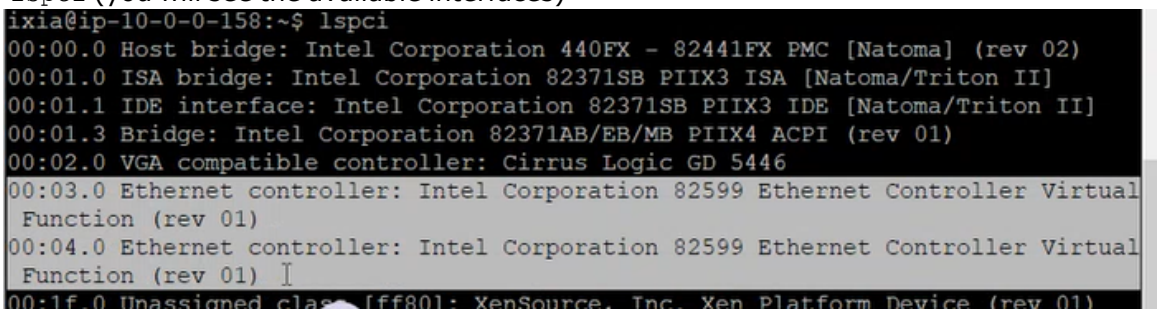
- Make sure you have an Elastic IP associated with the test agent.
- Stop the test agent instance and change the instance type.

- Select **c4.xlarge** type.

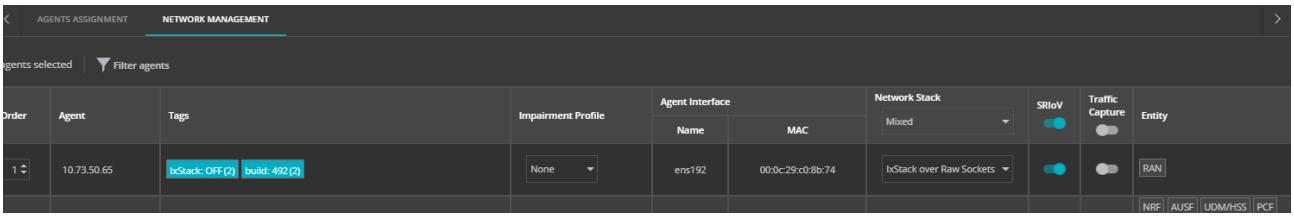


- Start the instance.
  Now the test agent instance is properly configured, has SRIOV support.
  You can check this using SSH on test agent and type:
  - `ethtool -I eth0` (you should see the `ixgbevf driver`)

  - `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.



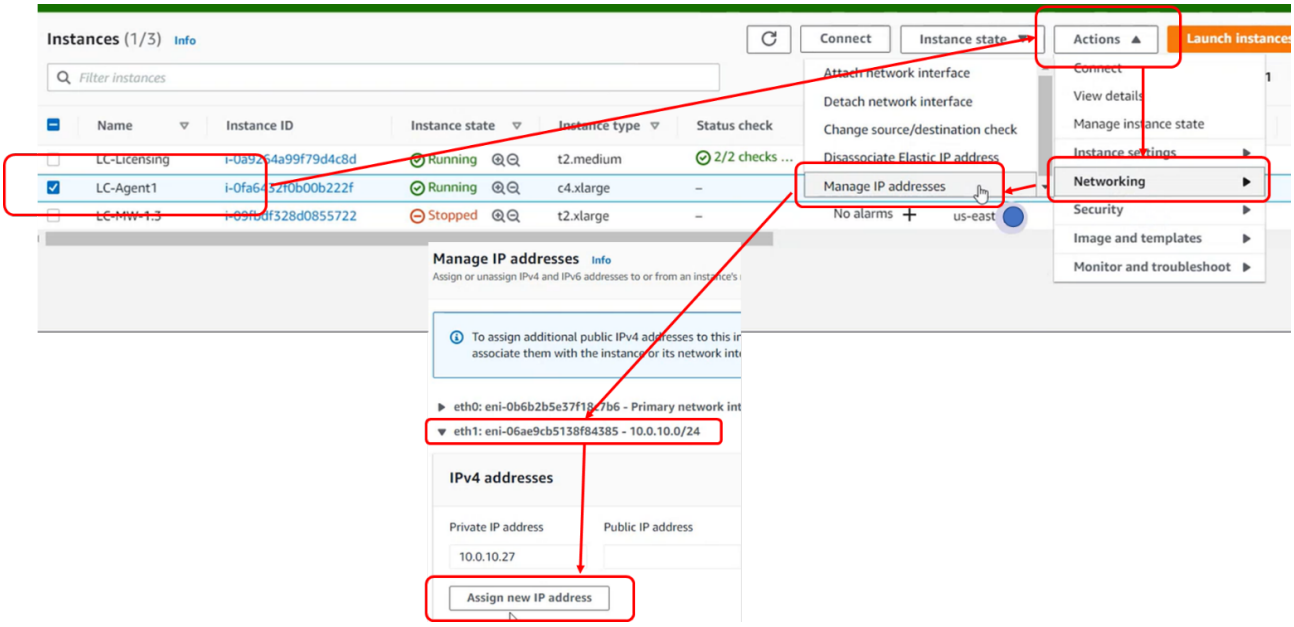
Now the agent is able to run application traffic on `eth1`.

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):

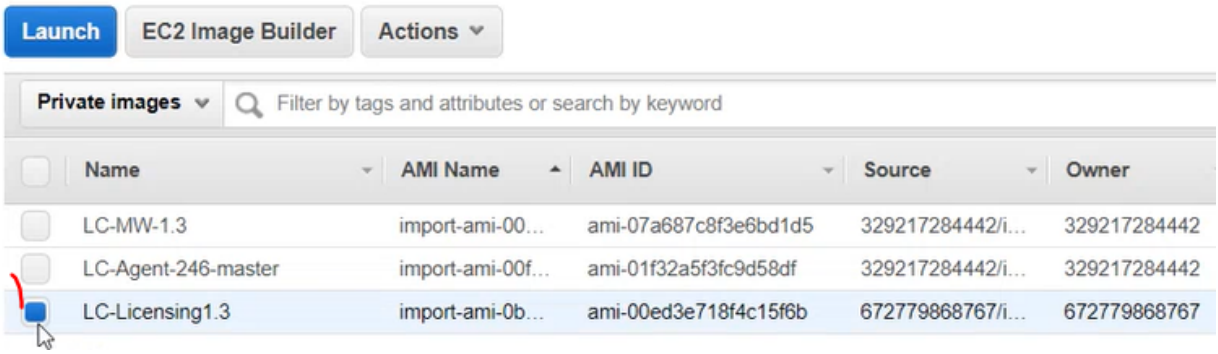
1. Go to **Instance** > **Networking** > **Manage IP addresses** > select **eth1** > **Assign new IP addresses**.
2. Add all the IP addresses that are used in the test.



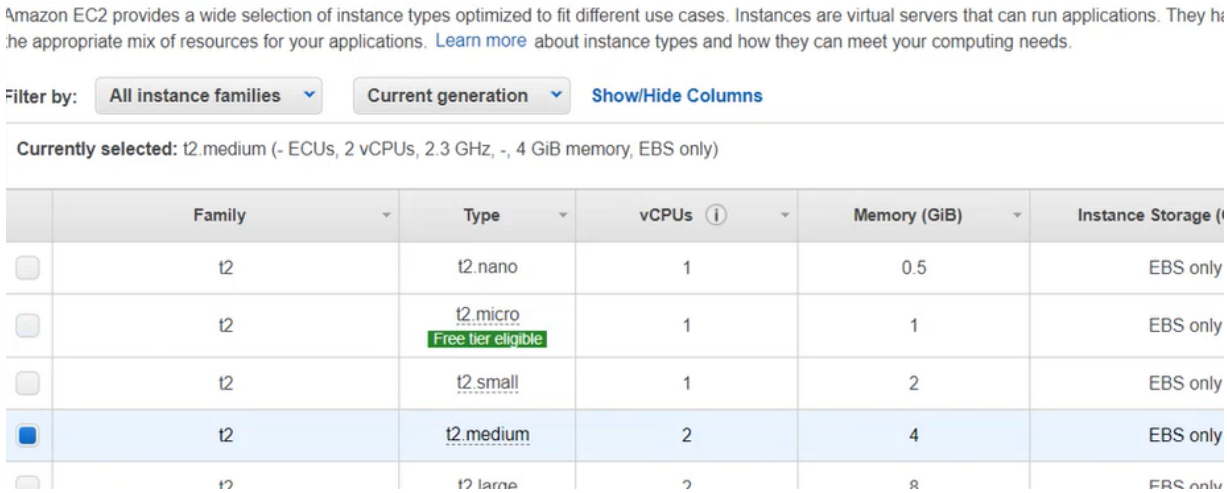
License Server Installation

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

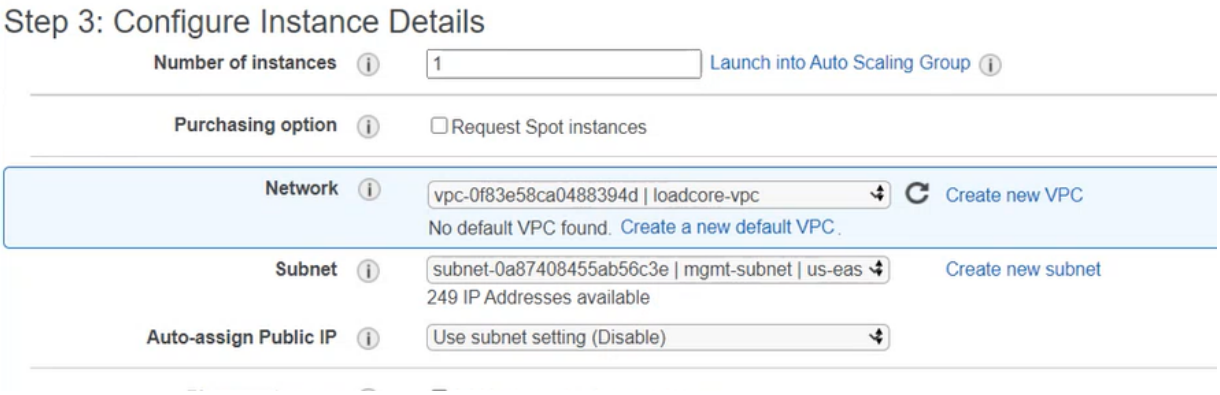
1. From the main console, select **Services** > **EC2** > **Images** > **AMI**.
   The browser will display the AMI images shared for your account.
   Select the License Server image and select **Launch**.



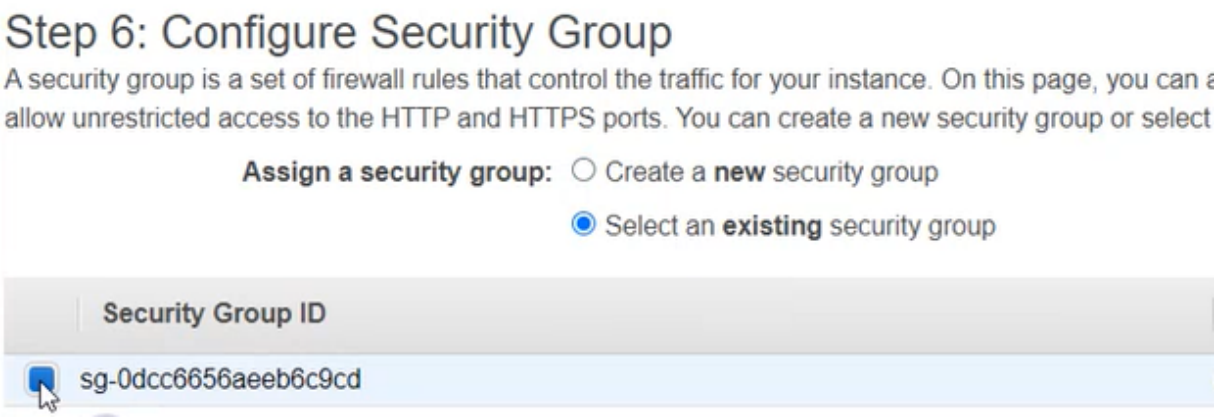
2. Select **t2.medium** for instance type:



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



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

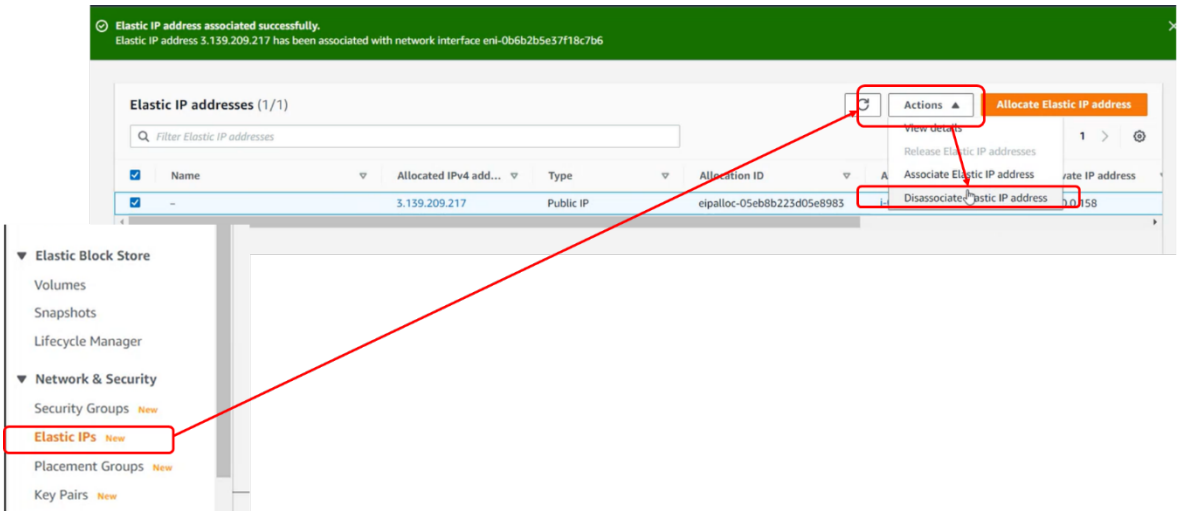


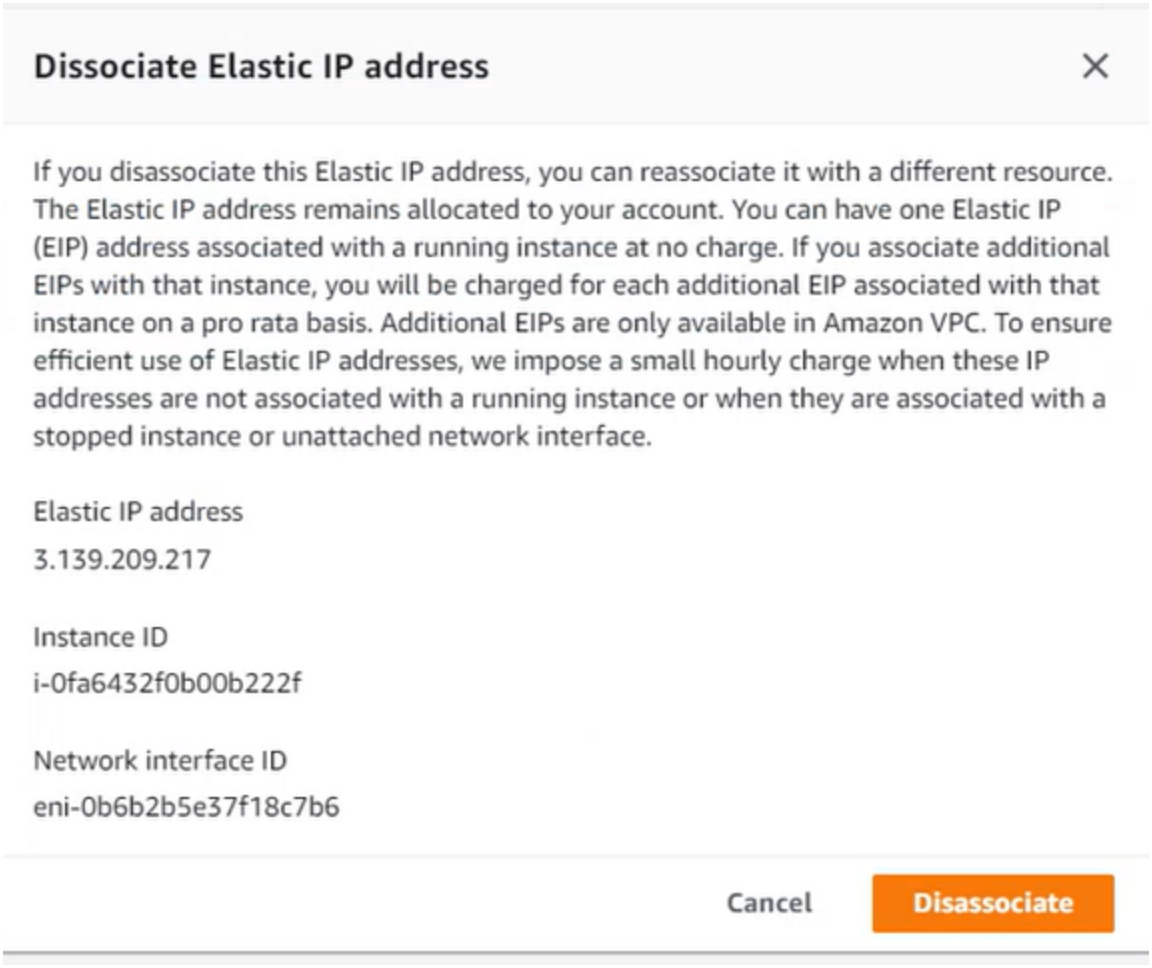
5. Leave the other parameters unchanged and launch the instance.

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:

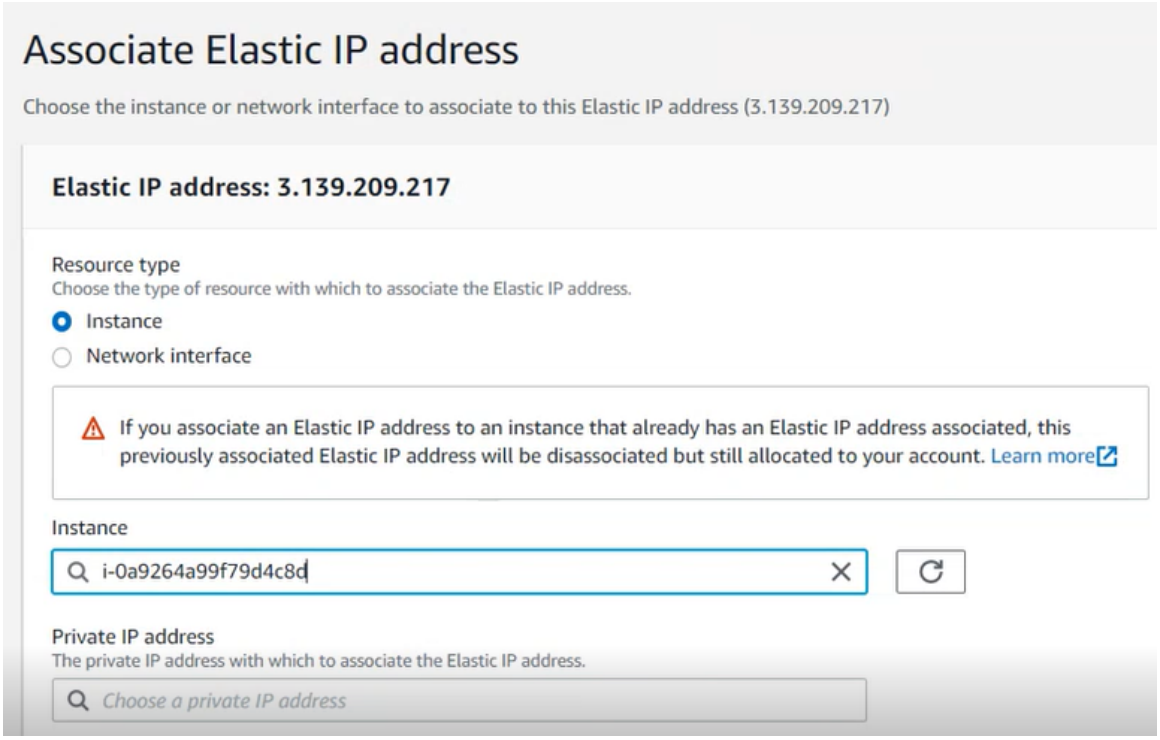
- Select **Elastic IP** > **Action** > **Disassociate**.





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

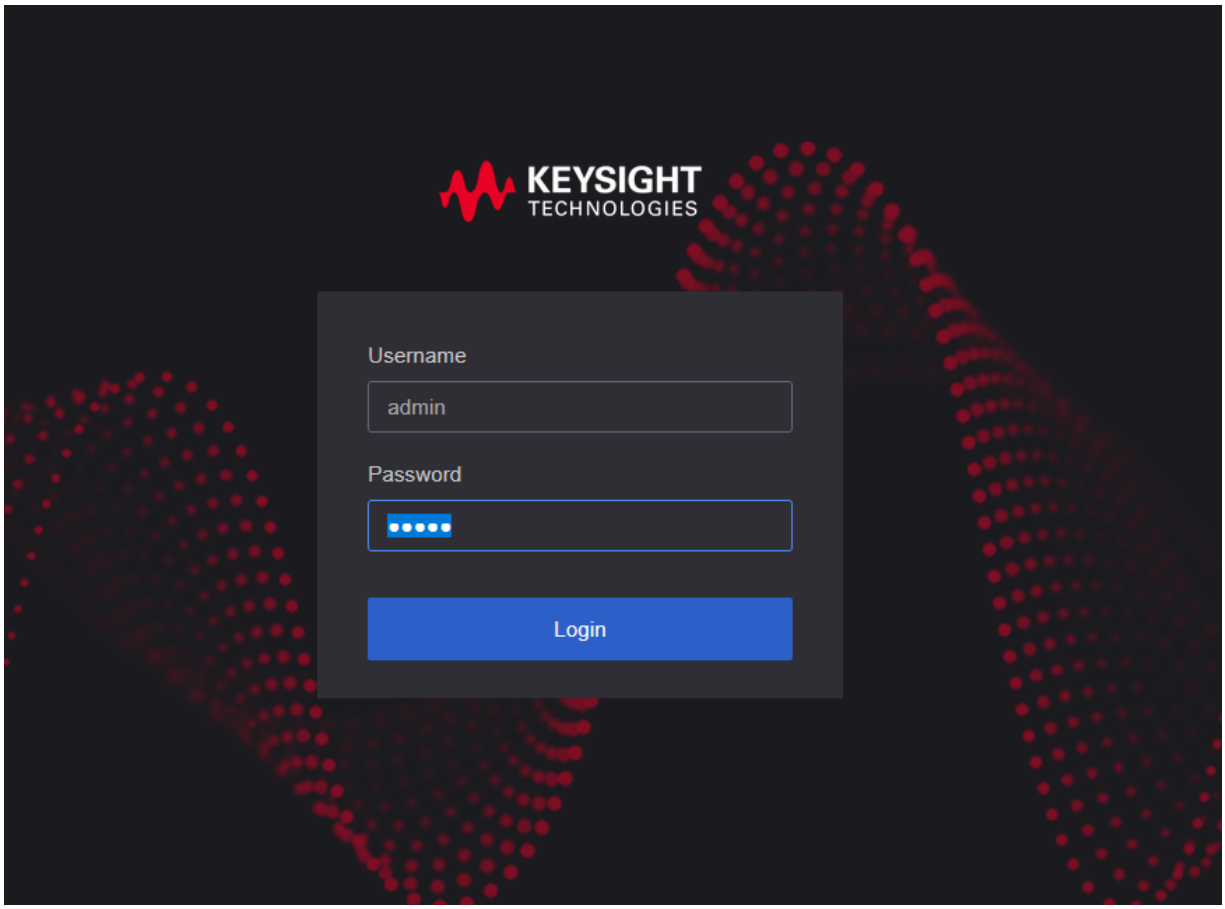
- **Resource type**: **Instance**.
- **Instance id**: select the if of the License server instance.



- Select **Associate**.

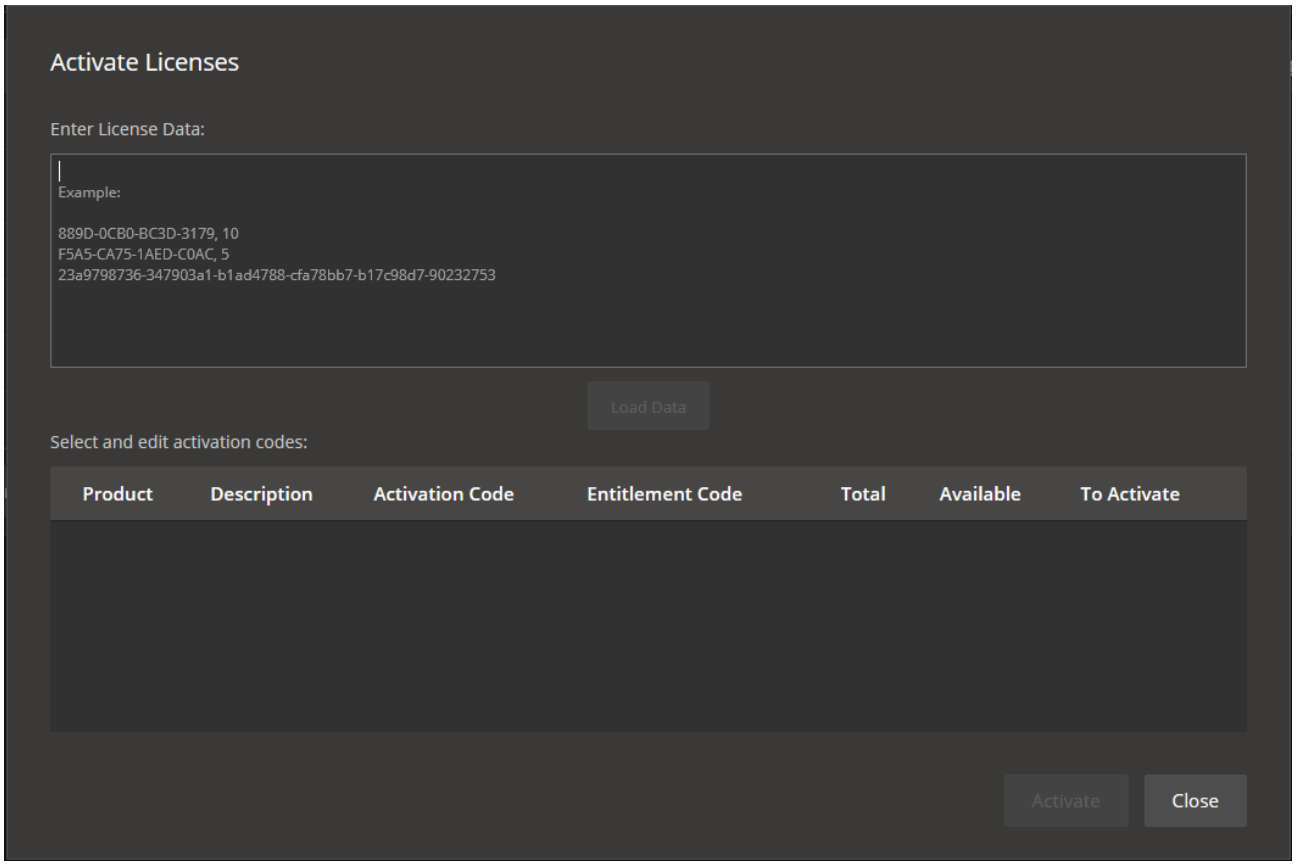
Now you can access the License Server UI and install the licenses:

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**.

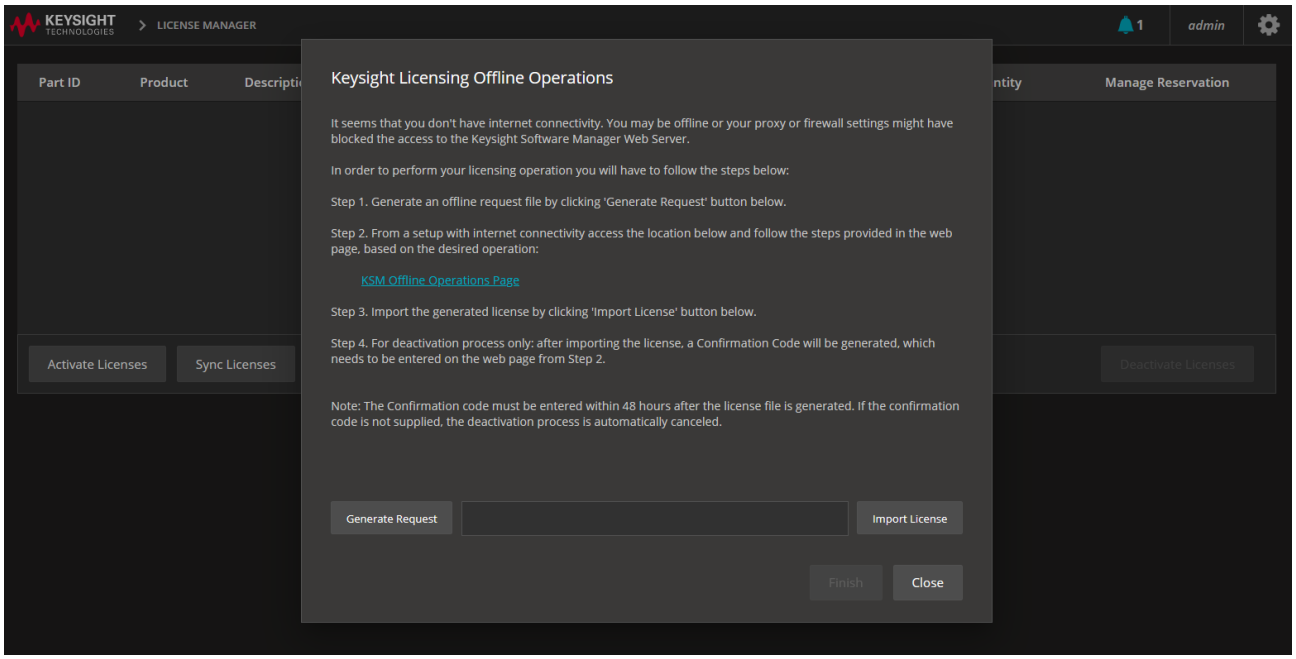


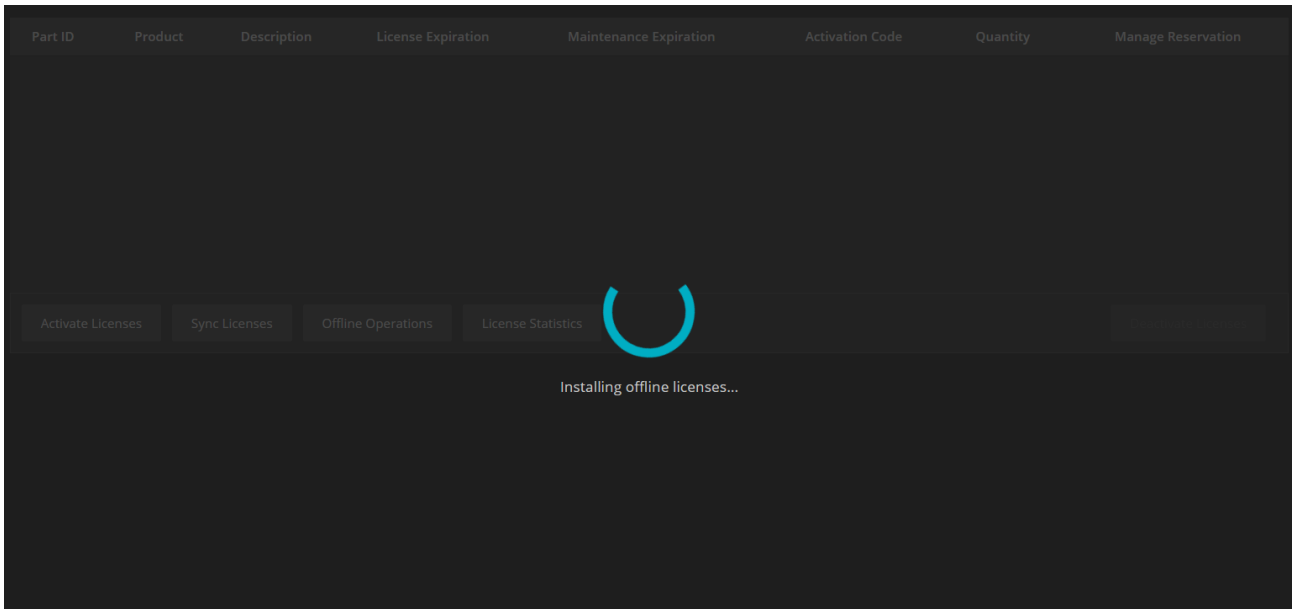
At this point, there are no licenses available on the newly installed server, therefore you must add the license codes you received (evaluation licenses can be obtained by sending an email to: Software-Eval-Request IX <software-eval-request.ix@keysight.com>.

Select the **Activate Licenses** button and add the license codes, select **Load Data** and then **Activate**.



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





Software Upgrades

Middleware Upgrade

You can upgrade the Middleware software as follows:

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

Agent(s) Upgrade

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

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

Expected response:

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

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

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

Troubleshooting

How to recover configuration files when LoadCore UI becomes unresponsive

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

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

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

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

1. Reboot the LoadCore Agent.
2. Connect to the LoadCore Agent using SSH.
3. Run the `agent-setup.sh` script: `sudo ./agent_setup.sh`.
4. Provide the required information regarding the interfaces used in the testbed:
   a. The IP address of the LoadCore Middleware.
   b. The management interface.
5. Allow the agent to be rebooted from the LoadCore UI : [**y**].

Monitor the health of the application

Check that all pods are up and running

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

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

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

Date should be the same on MW and agents

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

Check agents status

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

Backup and recovery

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

For more details, please refer to:

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

CHAPTER 5

Amazon AWS EKS Deployment

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

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

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

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

To install the `aws cli` tool, please refer to https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html.

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

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

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

AWS Docs:

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

How to push LoadCore container images into AWS EKS

This step can be done from a Linux machine.

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

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

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

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

Download the LoadCore k8s build and extract it.

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

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

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

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

How to install LoadCore Middleware in EKS

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

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

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

The requirements for the EKS cluster workers are as follows:

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

To create the `kubeconfig` file, please refer to https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html.

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

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

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

```
storage:
   class: gp2
```

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

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

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

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

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

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

or

`kubectl get svc -A` and search on External IP column

How to install LoadCore Agents on EKS

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

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

- Install `multus CNI` using the following command:
  `kubectl apply -f https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/master/config/multus/v3.7.2-eksbuild.1/aws-k8s-multus.yaml`
- Check the multus installation by using the command:
  `kubectl get pods -n kube-system`
  Each node should have one pod called `kube-multus-ds`.

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

```
config: |
  {
   "cniVersion": "0.3.1",
   "type": "ipvlan",
   "master": "ens7",
   "mode": "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 LoadCore components

To uninstall agents:

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

To uninstall Middleware:

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

CHAPTER 6

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

This page intentionally left blank.

Index

www.keysight.com