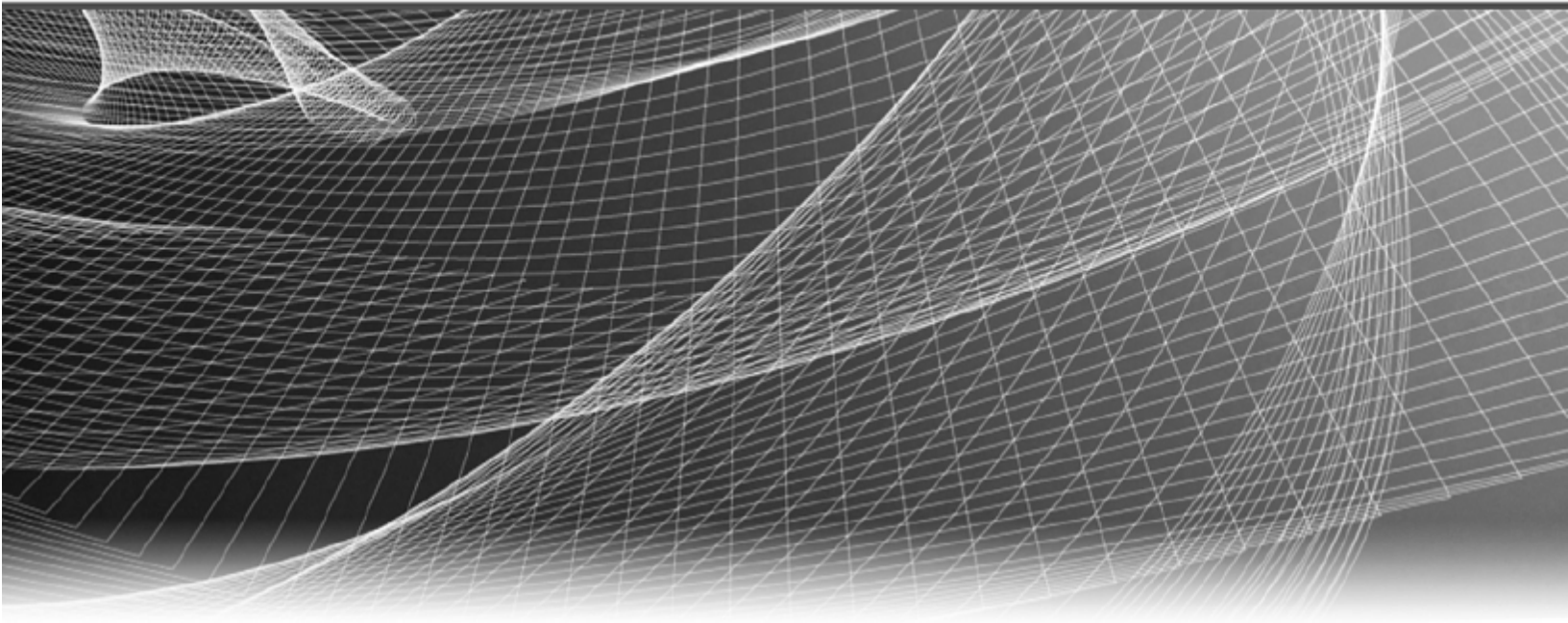


RSA | NetWitness Suite

Ixia CloudLens Integration

RSA

EMC²

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

Contents

RSA NetWitness Suite Integration with Ixia CloudLens **4**

Prerequisites 4

Integration Steps 5

Prepare AWS Environment **6**

Create CloudLens Project **7**

Install Docker Container on Decoder **9**

Install Docker Container on Clients **10**

Create Mapping between NetWitness Decoder and Ixia Clients **11**

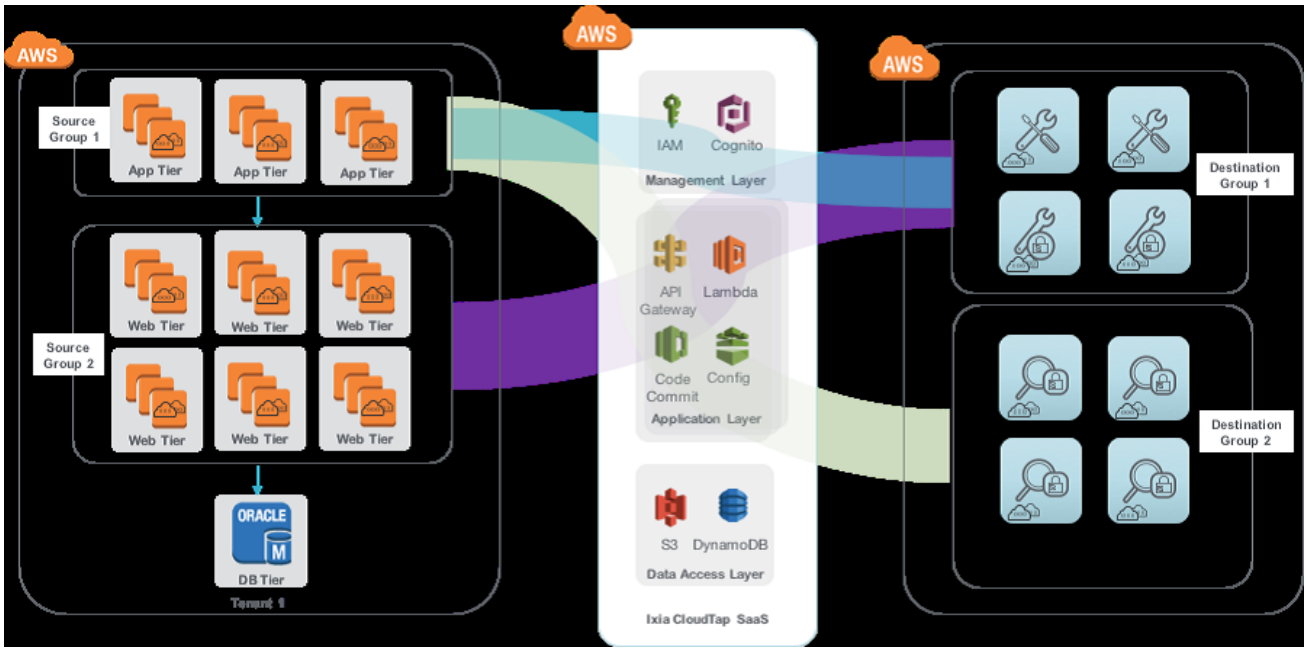
Validate CloudLens Packets Arriving at Decoder **13**

Set the Interface in the Packet Decoder **14**

RSA NetWitness Suite Integration with Ixia CloudLens

Ixia CloudLens™ is the industry's pioneer Software as a Service solution that moves traditional visibility controls down to each workload–enabling each workload to have its own microcosm packet brokering. Because of its Software as a Service architecture, Ixia's solution can span across multiple cloud instantiations and has the capability to provide visibility across private, public and hybrid cloud.

CloudLens Public container is a wire data collection system designed specifically to function seamlessly in public cloud deployments. To do this, it must cope effortlessly with rapid elastic scale, it must be a self-serve install for tenants in the cloud with no help required from the provider, it must operate without network constraints on the customer's network, and it must be out of band, not inline. And CloudLens does this without relying on hypervisor- or provider- specific features, in public or private cloud, for both virtualized network interfaces. CloudLens even works in on-premises bare metal.



Prerequisites

You need the following items before you begin the integration process:

- Ixia account (https://login.ixiacom.com/)
- Access to AWS console
- Network rout-able (and proper AWS Security Groups) for the containers to transfer data to the RSA NetWitness Suite Decoder.

Integration Steps

The remainder of this document discusses the steps for integrating the RSA NetWitness Suite Decoder with Ixia CloudLens.

1. Prepare AWS Environment
2. Create CloudLens Project
3. Install Docker Container on Decoder
4. Install Docker Container on Clients
5. Create Mapping between NetWitness Decoder and Ixia Clients
6. Validate CloudLens Packets Arriving at Decoder
7. Set the Interface in the Packet Decoder

Prepare AWS Environment

Prepare the AWS environment by performing the following procedures:

1. Deploy a NetWitness Decoder instance in AWS. For details, see AWS Deployment Guide.
2. Deploy client machines onto which you want to route the traffic to NetWitness Decoder.
   - See the Ixia CloudLens documentation for specifications needed for supported client machines. Ixia public documentation is available at https://store.ixiacom.com/product/cloudlens-public.
   - See the AWS documentation to deploy AWS on the supported client machines. The AWS Deployment Guide for 10.6.3 is available here: https://community.rsa.com/docs/DOC-75925.

Create CloudLens Project

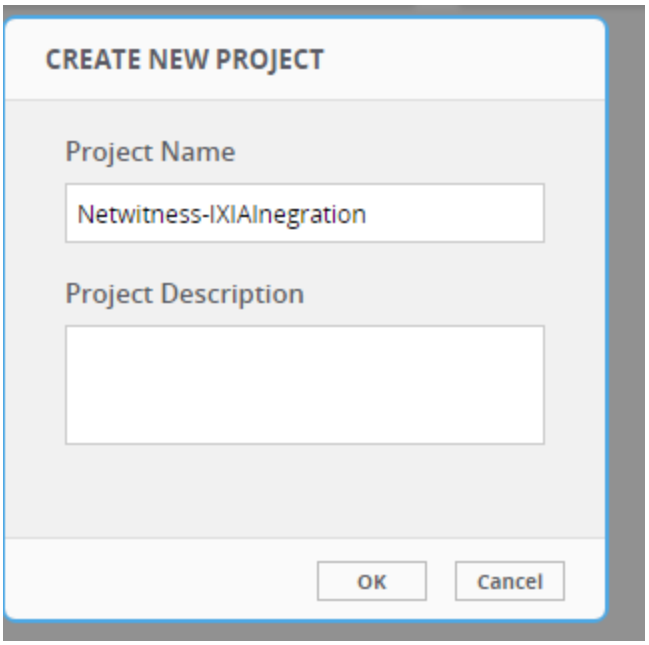
Before you access the CloudLens site for the first time, create an Ixia login account (if you don't already have one) at https://login.ixiacom.com/. Send your Ixia login account email to cloudlens@ixiacom.com so they can provide you with access to the Sandbox.

Access the CloudLens public site at https://www.ixia-sandbox.cloud/ and login with your Ixia account credentials.

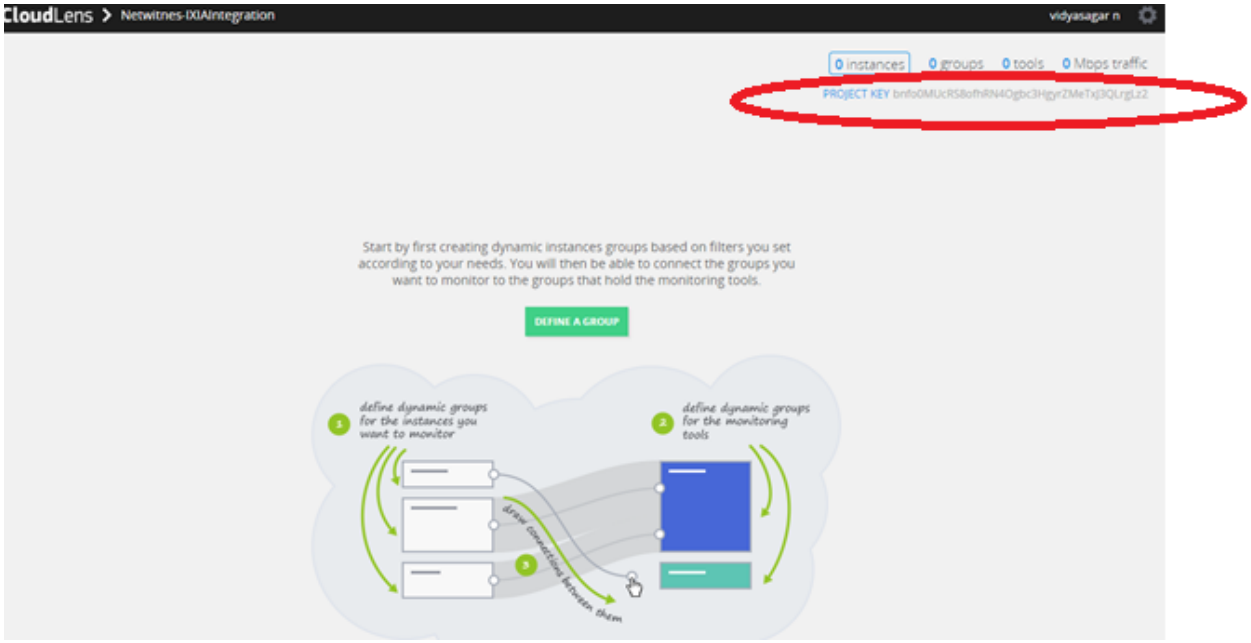
To create a new project and get your project key:

1. Click the + Icon to create a new project.

   In the following image, the project name is **Netwitness-IxiaIntegration**, but you can enter any name you wish.



2. Click on your newly created project and make note of your Project Key. You will need this later for the API key configured on the **Host & Tool agents**.



Install Docker Container on Decoder

Next, you need to install the Docker container onto the NetWitness Decoder.

1. Open an SSH connection the public IP address of the RSA NetWitness Suite Decoder.
2. Install the Docker RPM onto the Decoder:

   ```
   #rpm -iUvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
   # yum -y install docker-io
   ```

   Note: Make sure to enable the required repository, so that you can install it.

3. Start the Docker service:

   ```
   # service docker start
   ```

4. Use the `docker run` command to access the Ixia repository and obtain the cloudlens-sandbox-agent container. Replace the variable **ProjectKeyFromIxiaProjectPortal**, which identifies your project key in Ixia portal, with the Project Key you created in the previous section.

   ```
   docker run --name ca \
   -v /:/host \
   -d --restart=always \
   --net=host \
   --privileged \
   ixiacom/cloudlens-sandbox-agent \
   --server agent.ixia-sandbox.cloud \
   --accept_eula y \
   --apikey ProjectKeyFromIxiaProjectPortal
   ```

Install Docker Container on Clients

You need to install the Docker Container onto the client machines for which you want to route the traffic to the NetWitness Decoder.

1. Open an SSH connection the Public IP address of the AWS Client instance.
2. You must enable root access to the OS CLI (for example, `sudo su -`).
3. Install Docker by running the following command:

   ```
   # yum -y install docker
   ```

   Note: Make sure to enable the required repository, so that you can install it.

4. Start the Docker service:

   ```
   # service docker start
   ```

5. Use the `docker run` command to access the Ixia repository and obtain the **cloudlens-sandbox-agent** container. Replace the variable **ProjectKeyFromIxiaProjectPortal**, which identifies your project key in Ixia portal, with the Project Key you created in the previous section.

   ```
   docker run --name ca \
   -v /:/host \
   -d --restart=always \
   --net=host \
   --privileged \
   ixiacom/cloudlens-sandbox-agent \
   --server agent.ixia-sandbox.cloud \
   --accept_eula y \
   --apikey ProjectKeyFromIxiaProjectPortal
   ```

Warning: If you cut and paste commands from a PDF, it is a good idea to first paste into a test editor such as Notepad, and then confirm the syntax before pasting into the OS CLI. We have seen issues where direct cut and paste between PDF and CLI can contain dashes or other special characters that should not be part of the commands.

Create Mapping between NetWitness Decoder and Ixia Clients

Next, you need to create a mapping between the NetWitness Decoder and the client machines for which you want to route the traffic to the Decoder.

1. From the CloudLens site, open your project by double-clicking it.
   a. Click the **Define Group** button or the Instances count.

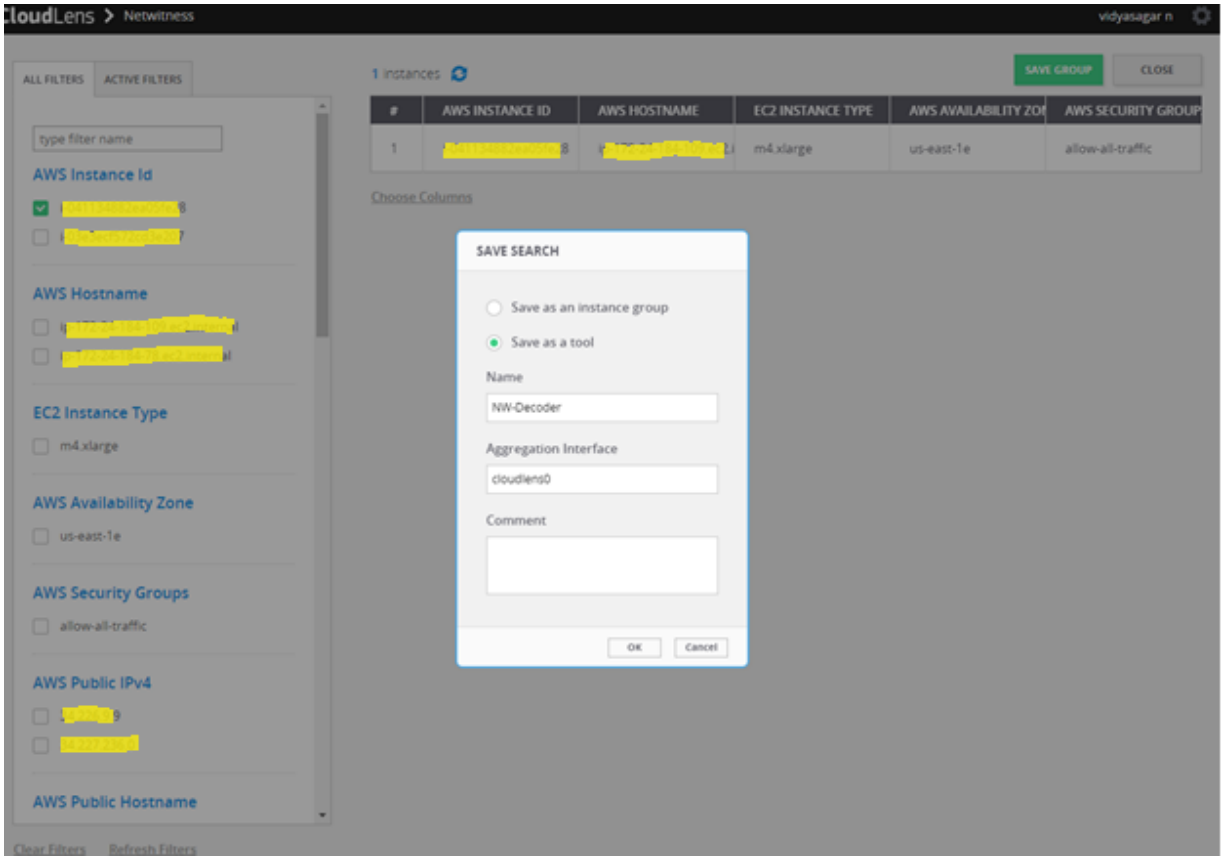
      You should see two instances listed, one for your decoder and the other for the client machines.

   b. Filter for the decoder instance and click **Save Search**.

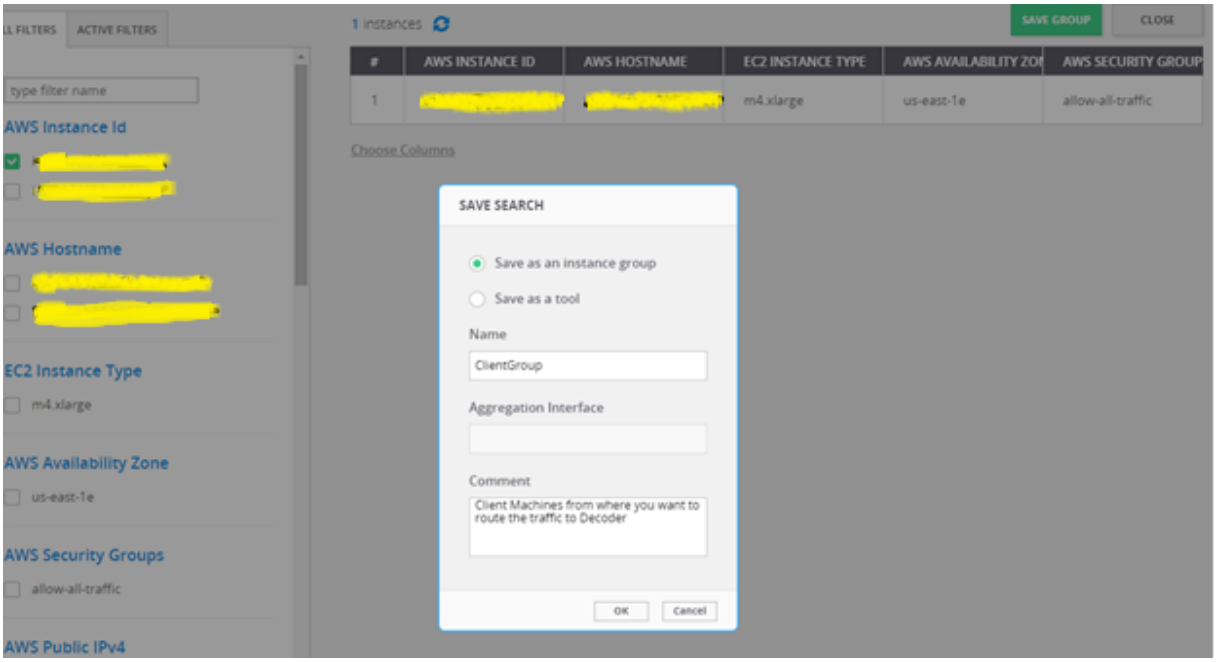
   c. Choose **Save as a tool**.

      You will be asked to specify a name for the tool, and also the **Aggregation Interface**.

   d. Give the Aggregation Interface a meaningful name, for example **CloudTAP**. This is a virtual interface that will appear in the OS where your Tool is installed. You will later need to instruct your tool to 'listen' to that interface.



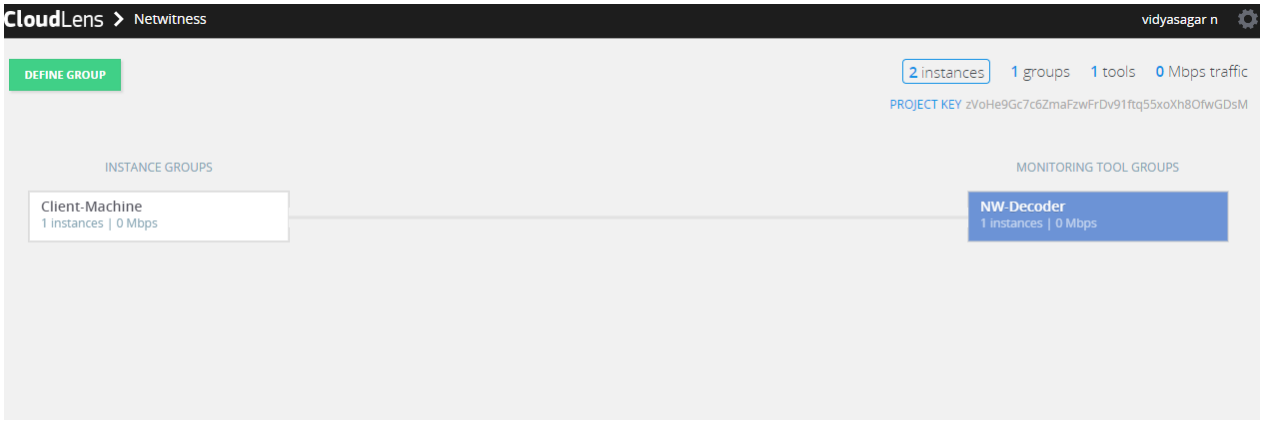
e. Filter the client host instance from the list, and click **Save Search**.



2. Navigate back to the top-level view of the project.

   You will see your client machine and Decoder instances.

3. Drag a connection between them to allow the flow of packets.

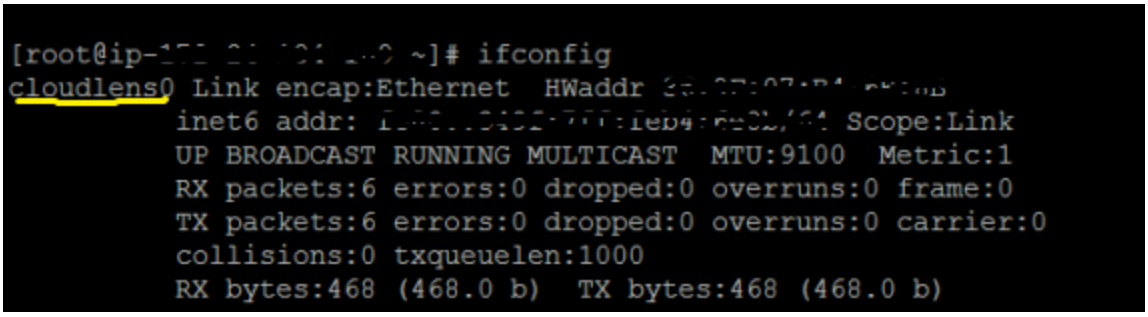


Validate CloudLens Packets Arriving at Decoder

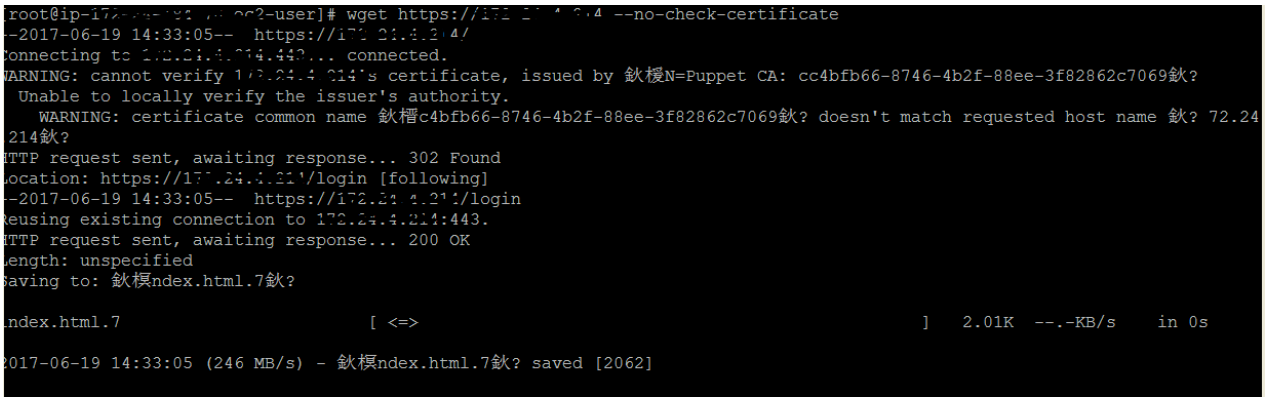
Once you have the mapping between your clients and the NetWitness Decoder, the next step is to validate that packets are actually arriving at the Decoder.

1. Open an SSH connection the public IP address of the Decoder.
2. Run the `ifconfig` command.

   You will see the new aggregation interface which you created.

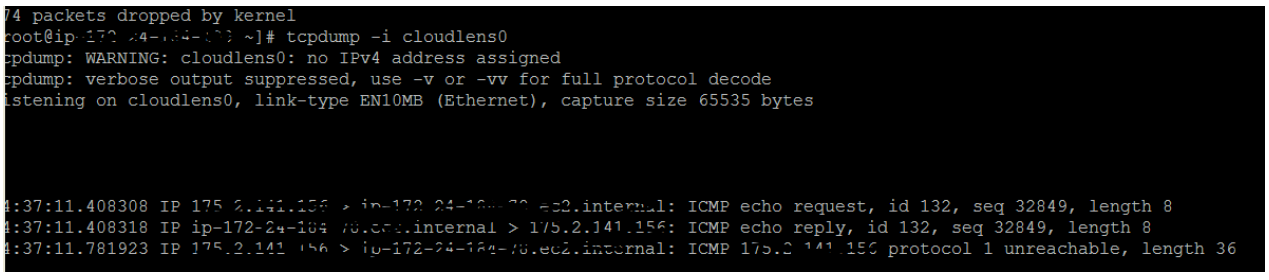


3. Generate traffic from the client OS instance CLI. For example, run `wget http://www.google.com/`.



4. Open an SSH connection the public IP address of the Decoder to go to your Decoder instance CLI, and look for suitable results in the tcpdump by running the following command:

   `tcpdump -I Cloudlens0`



Set the Interface in the Packet Decoder

Finally, in your RSA NetWitness Suite Packet Decoder, set the interface to use for the Ixia integration.

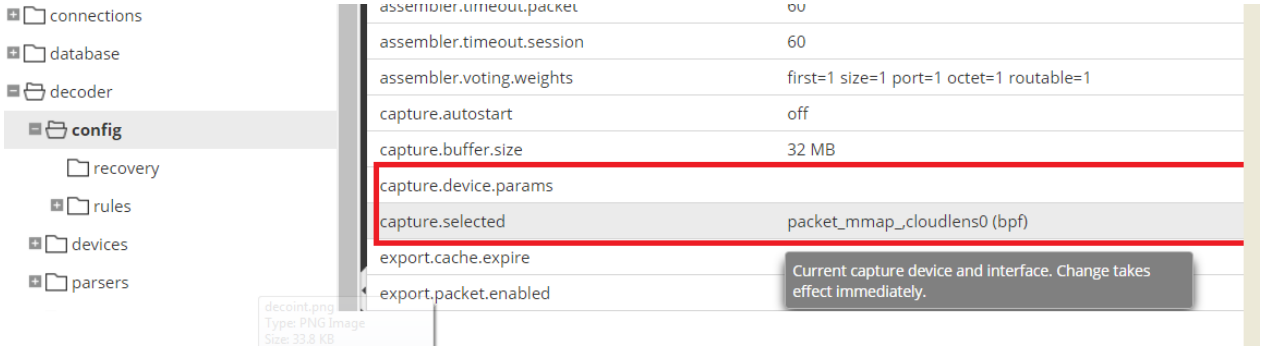
1. SSH to Decoder and restart decoder service.

   ```
   $ sudo restart nwdecoder
   ```

   The user should now be set to capture the network traffic in Decoder.

2. Log onto NetWitness and n the RSA NetWitness menu, select **Administration > Services**.
3. In the Admin Services view, select a Decoder service, and select **View > Explore**.
4. Expand **decoder** and then click **config** to view the configuration settings.
5. For the **capture.selected** parameter, set it to the following:

   ```
   packet_mmap_,cloudlens0(bpf)
   ```

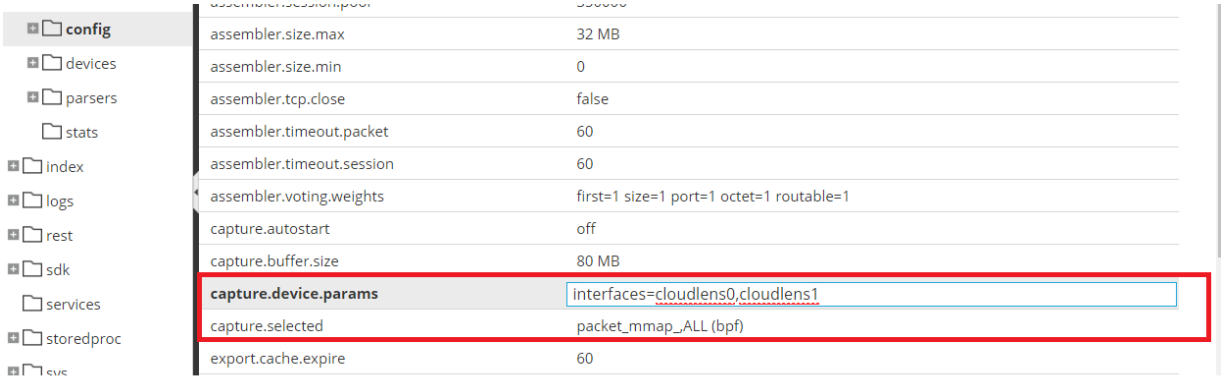


6. (Conditional) - If you have multiple capture interfaces on the Packet Decoder, set your parameters as follows:

   a. Set the **capture.device.params** and the **captured.selected** parameters. For example, if you have two interfaces, you could set the parameters as follows:

   ```
   capture.device.params --> interfaces=cloudlens0,cloudlens1
   ```

   ```
   capture.selected --> packet_mmap_,All
   ```



b. Restart Decoder service after you set the **capture.selected** parameter.

This completes the RSA NetWitness integration with Ixia CloudLens.