IXIA CLOUDLENS WITH RIVERBED APPRESPONSE DEPLOYMENT GUIDE IN AZURE

DEPLOYMENT GUIDE

PROBLEM:

Organizations, even those not typically associated with technology, are migrating to the cloud. This trend is growing because the cloud offers increased flexibility and agility. With this mass migration, organizations have more segments to manage and more potential blind spots in their networks. Regardless of where infrastructure and applications reside, security and compliance needs remain the same. Organizations are finding that their traditional network visibility solutions are unable to meet their needs for visibility of cloud-based data.

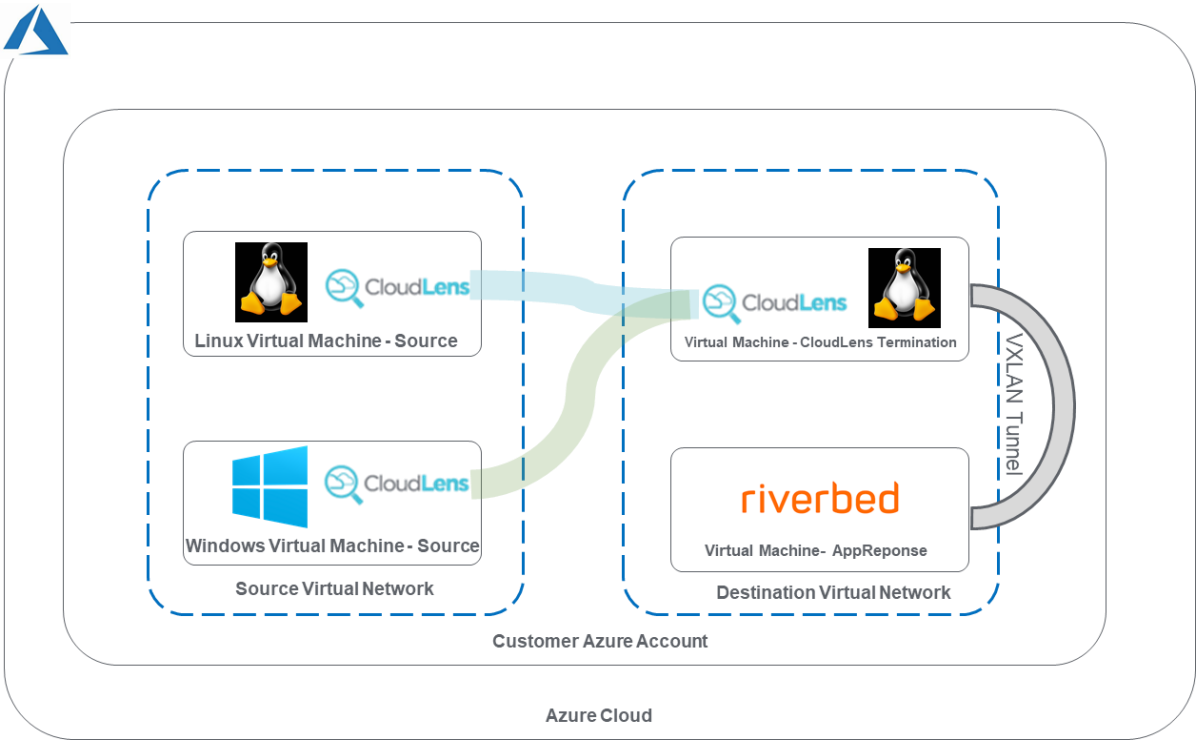
SOLUTION:

CloudLens™, Ixia's platform for public, private and hybrid cloud visibility addresses the challenges of granular data access in the cloud. CloudLens is the first network-level solution that provides Visibility-as-a-Service (VaaS) through a Software-as-a-Service (SaaS). It is also the industry's first cloud service-provider agnostic visibility platform. This guide describes how to deploy Riverbed AppResponse together with CloudLens visibility in Azure.

KEY FEATURES:

- Elastically scales on-demand – so visibility auto-scales horizontally along with the Virtual Machines monitored and the Virtual Machines that are needed to do the monitoring
- Automates cloud visibility management by providing it as a service – so no architectural changes required
- Reduces errors by eliminating manual configuration
- Easy to use and setup with a drag and drop interface
- Reduces bandwidth to tools by filtering packets at the source Virtual Machines, eliminating unwanted traffic so tools operate optimally

SAMPLE DEPLOYMENT ARCHITECTURE



* Shown above is a sample deployment, while destination components need to in the same subnet, monitored sources instances can be located in any subnet. CloudLens Sensors run on customer Azure Virtual Mach ines, register up to the CloudLens SaaS which manages them and forwards desired traffic to the destination. Sources may be Linux and / or Windows Azure Virtual Machines. On the Destination subnet, a Linux Virtual Machine is required to terminate the CloudLens Tunnel. Another Virtual Machine runs Riverbed AppResponse.

PREPARE AZURE ENVIRONMENT

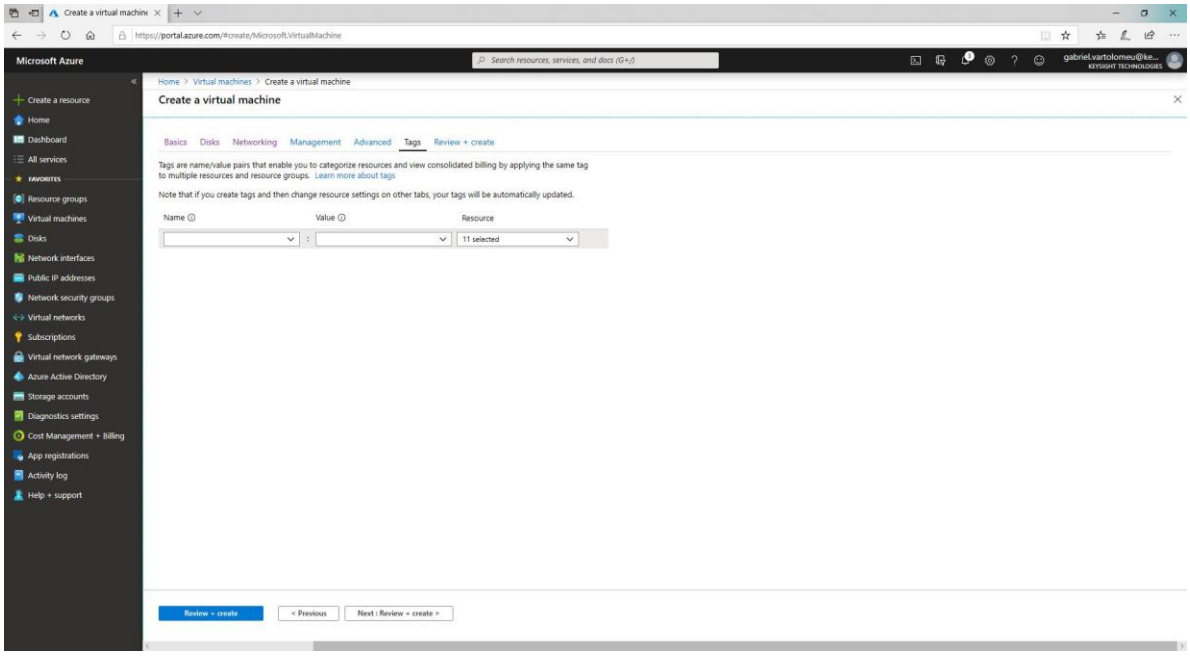
NOTE: IN THIS EXAMPLE WE ARE ASSUMING THE SOURCE VIRTUAL MACHINES ALREADY EXIST, YOU CAN THEN LOAD CLOUDLENS SENSORS ONTO THOSE VIRTUAL MACHIENS AS DESCRIBED ON P 5-6 OF THIS DOCUMENT. IN AZURE, EACH VIRTUAL MACHINES INBOUND PORT RULES MUST BE CONFIGURED TO ALLOWS PROPER FUNCTIONING OF CLOUDLENS AND APPRESPONSE, PLEASE SEE P 3-4 FOR DETAILS

YOU DO ALWAYS NEED TO CREATE ONE LINUX INSTANCE IN AZURE, THIS ONE WILL BE USED FOR TERMINATING THE CLOUDLENS TUNNELS, AND FORWARDING TRAFIC FROM CLOUDLENS TO RIVERBED VIA VXLAN AS DESCRIBED ON P 7. PLEASE CONSULT AZURE DOCUMENTATION FOR FULL DETAILS OF HOW TO CREATE A VM.

ALSO, WE ARE ASSUMING IN THIS EXAMPLE THAT AN INSTANCE OF RIVERBED APPRESPONSE IS ALREADY INSTALLED IN AZURE, PLEASE CONTACT RIVERBED FOR ASSISTANCE IF NEEDED.

LOAD CLOUDLENS SENSORS AND CONFIGURE VMS

Step 1 – not required but recommended, configure Tags for easier identification and grouping of Virtual Machines in CloudLens



Step 2 – Configure Inbound Port Rules in Network Security Groups, and apply to Virtual Machines

Note: *Azure default for Outbound is open for All Traffic. But for Azure Virtual Machine Inbound Port Rules, a few ports numbers need to be explicitly opened:*

Source Virtual Machines:

- UDP 19993 (CloudLens Tunnel) *
- TCP 22 (if Linux) **
- TCP 3389 (if Windows) **

CloudLens Termination Virtual Machine:

- UDP 19993 (CloudLens Tunnel) *
- UDP 4789 (VxLAN Tunnel) ***
- TCP 22 **

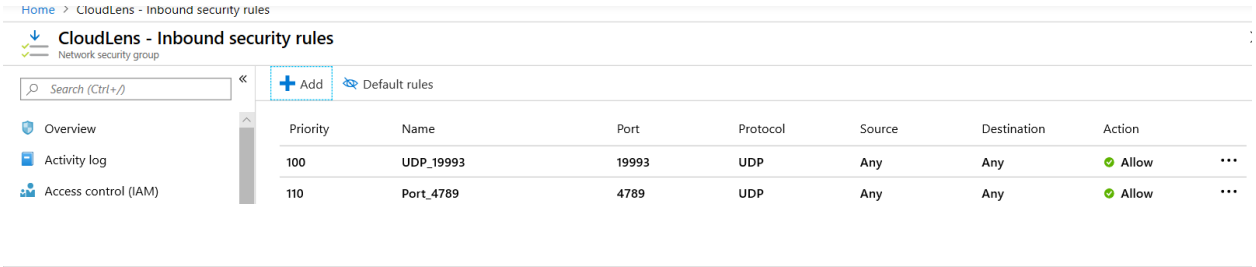
Riverbed AppResponse Virtual Machine:

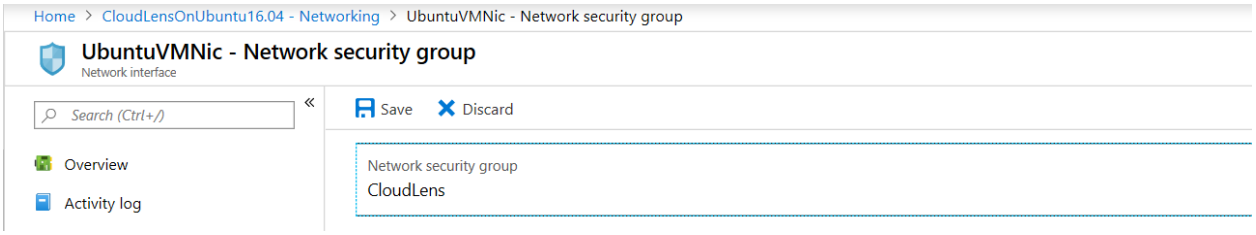
- UDP 4789 (VxLAN Tunnel) ***
- TCP 22 **
- TCP 443 **

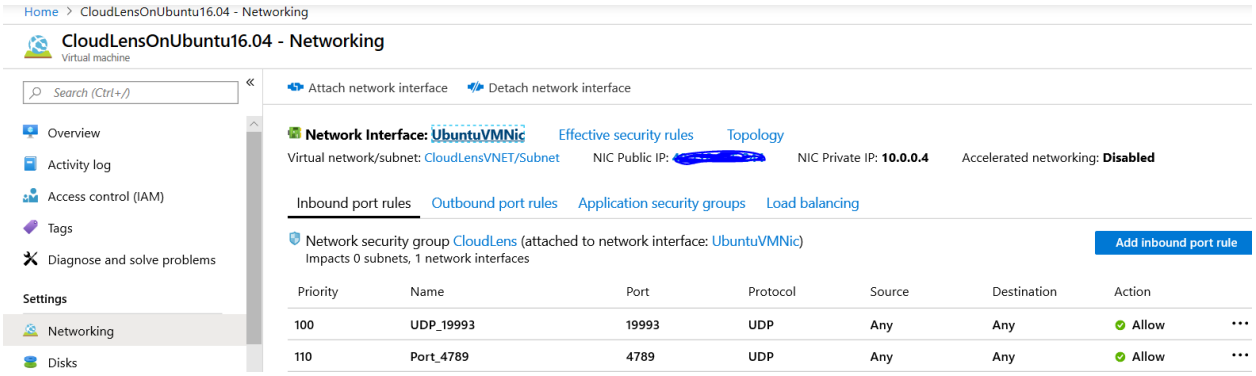
* Leave open all IP addresses, however if stricter controls are required contact Ixia support

** Specify IP addresses of customer administrators

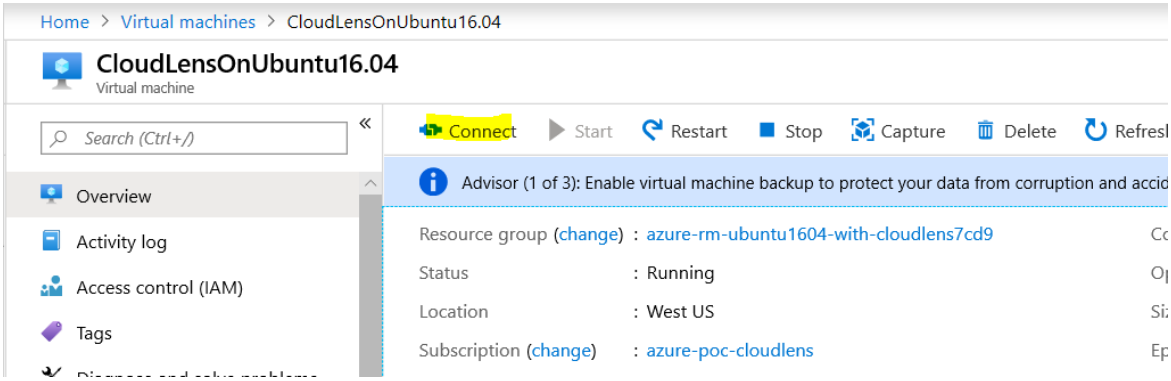
*** Leave open all IP addresses, however if stricter controls are required contact Ixia support







Step 3 – 'Connect' to SSH of Source Virtual Machines from Azure Console (or use RDP for Windows)



Step 4 – Install CloudLens Container on Linux Source Virtual Machines (if Windows, follow Step 5)

AS WELL AS installing CloudLens container on Linux Virtual Machine used for CloudLens Termination

Note: Customers are assumed to have already created an account at http://ixia.cloud before completing the next step. If you don't have an account, you can sign up for a 45-day free trial. After login please create or view your CloudLens Project, and make note of the Project Key (aka API key) which you will need in Step B below.

Step A – If Docker Engine is not already present, install Docker from https://docs.docker.com/install/

e.g. in the case of Ubuntu, docker can be installed as follows;

```
sudo apt update
sudo apt-get install -y docker.io
```

Step B – Run the CloudLens container

Note: You will need to substitute your CloudLens Project Key (aka API key) here

```
sudo docker run \
--name cloudlens \
-v /:/host \
-v /var/run/docker.sock:/var/run/docker.sock \
-d --restart=always \
--net=host \
--privileged \
ixiacom/cloudlens-agent:latest \
--server agent.ixia.cloud \
--accept_eula y \
--apikey <substitute your CloudLens Project Key here> \
```

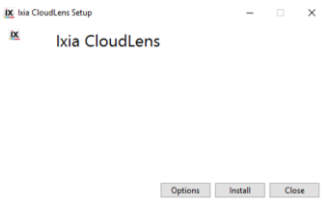
Step 5 – where applicable Install CloudLens Sensor onto Windows Virtual Machines

Note: Customers are assumed to have already created an account at http://ixia.cloud before completing the next step. If you don't have an account, you can sign up for a 45-day free trial. After login please create or view your CloudLens Project, and make note of the Project Key (aka API key) which you will need in Step C below

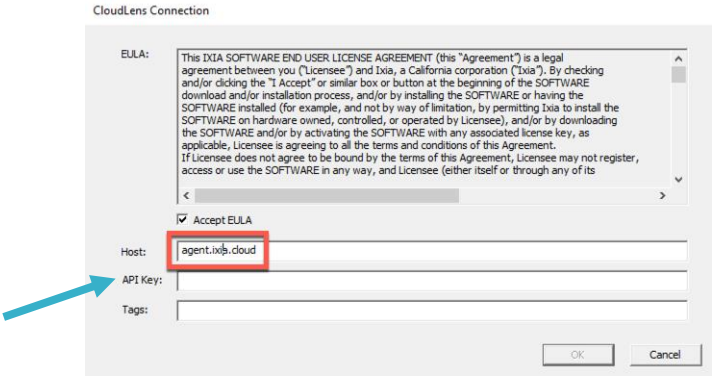
Step A – Download Ixia's CloudLens agent from the link provided on Window Server

https://agent.ixia.cloud/updates/windows/latest

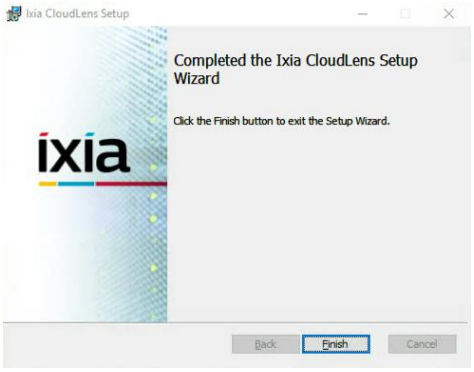
Step B – Install CloudLens agent (click through defaults – until Step C shown next page)



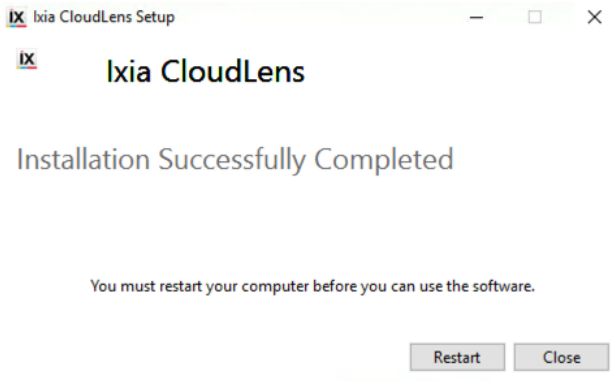
Step C – The Windows CloudLens Sensor should be associated with a Project previously created in https://ixia.cloud. The value of Host: agent.ixia.cloud . You must also specify your own Project Key (aka API key)



Step D – Finish CloudLens sensor installation



Step E – Restart the instance and verify the instance is associated with the CloudLens project created.



CONFIGURE CLOUDLENS TERMINATION VIRTUAL MACHINE

Step 1 – Connect to SSH of Linux Virtual Machine

Step 2 – Configure VxLAN tunnel between CloudLens and Riverbed AppResponse (e.g. below)

- Create a vxlan interface on the VM

  sudo ip link add vxlan0 type vxlan id 42 group 239.1.1.1 dev eth0 dstport 4789

- Create a bridge between the vxlan interface and the destination IP of the AR11

  sudo bridge fdb append to 00:00:00:00:00:00 dst Riverbed_AR_destination_IP dev vxlan0

- Add an address on the vxlan0 interface

  sudo ip addr add 192.168.200.1/24 dev vxlan0

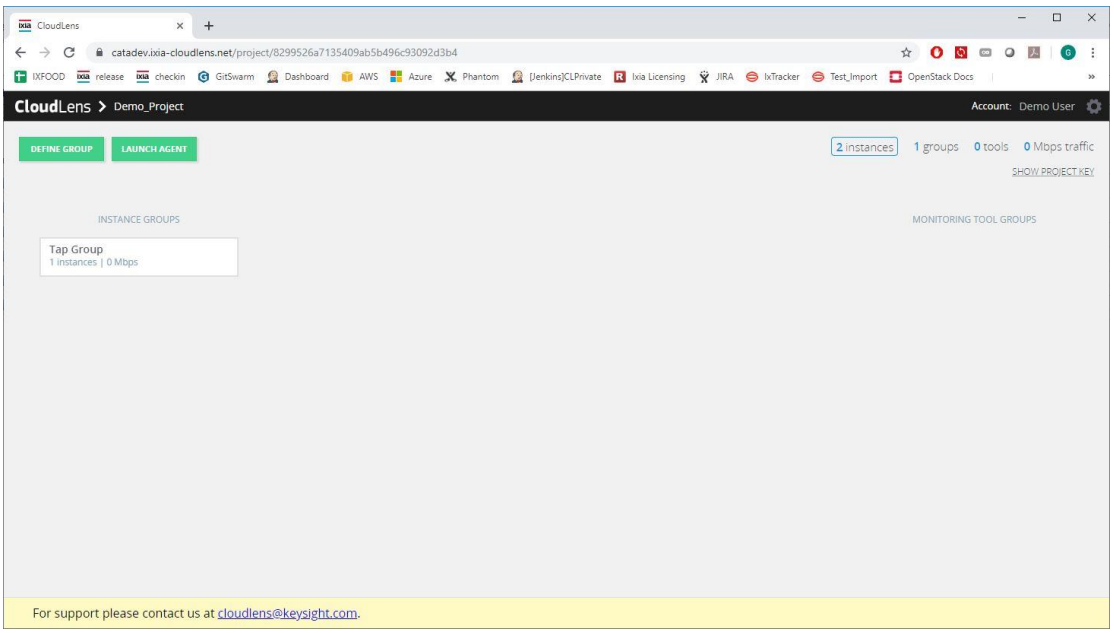
- Enable the interface

  sudo ip link set up dev vxlan0

USING CLOUDLENS SAAS PORTAL

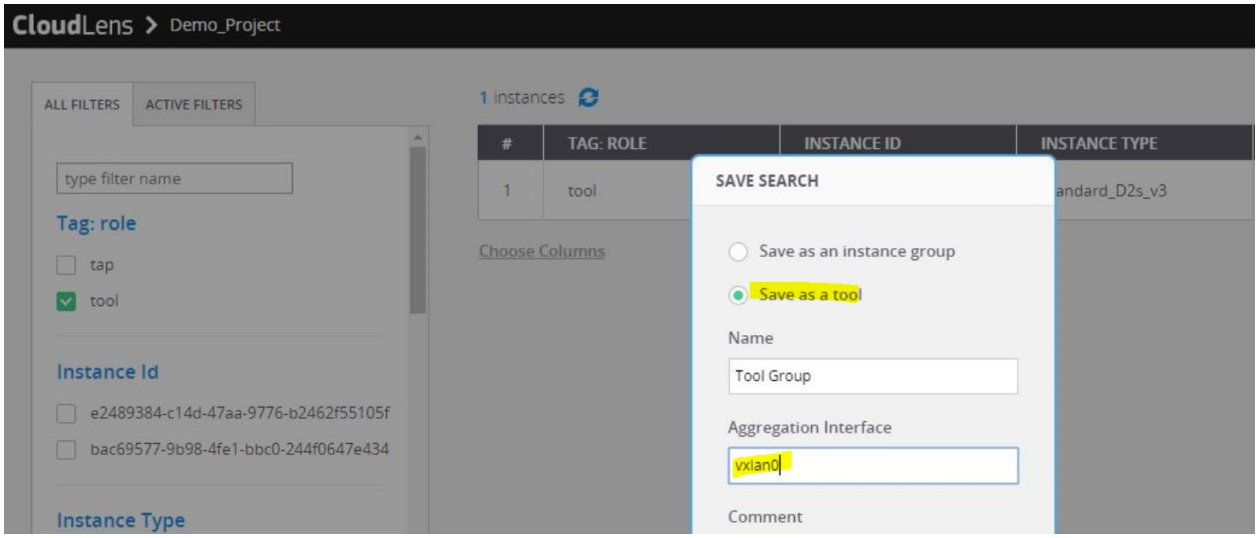
Step 1 – Log into http://ixia.cloud and open your previously created CloudLens Project

Step 2 – Click on Instances counter, find your Source Virtual Machines(s), then create an 'Instance Group' for the instances you wish to monitor (optionally you may create multiple Tap Groups for different types of Source VMs – if you previously added Tags to your VMs this can help with grouping)

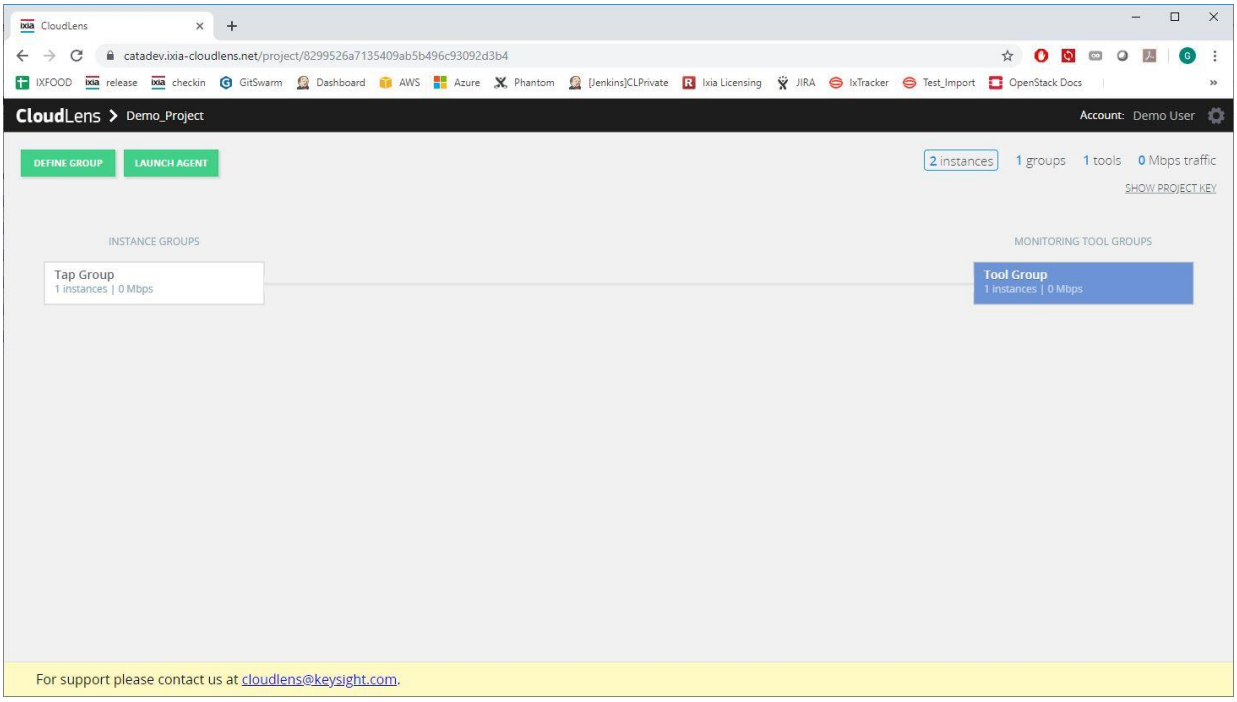


Step 3 - Configure your CloudLens Termination node. Click on the Virtual Machine that is acting as the instance which terminates the CloudLens tunnel, and bridges via VxLAN to the Riverbed AppResponse.

- This time configure the Group as a 'Tool'
- You must specify the Aggregation Interface to match what you setup earlier e.g. vxlan0

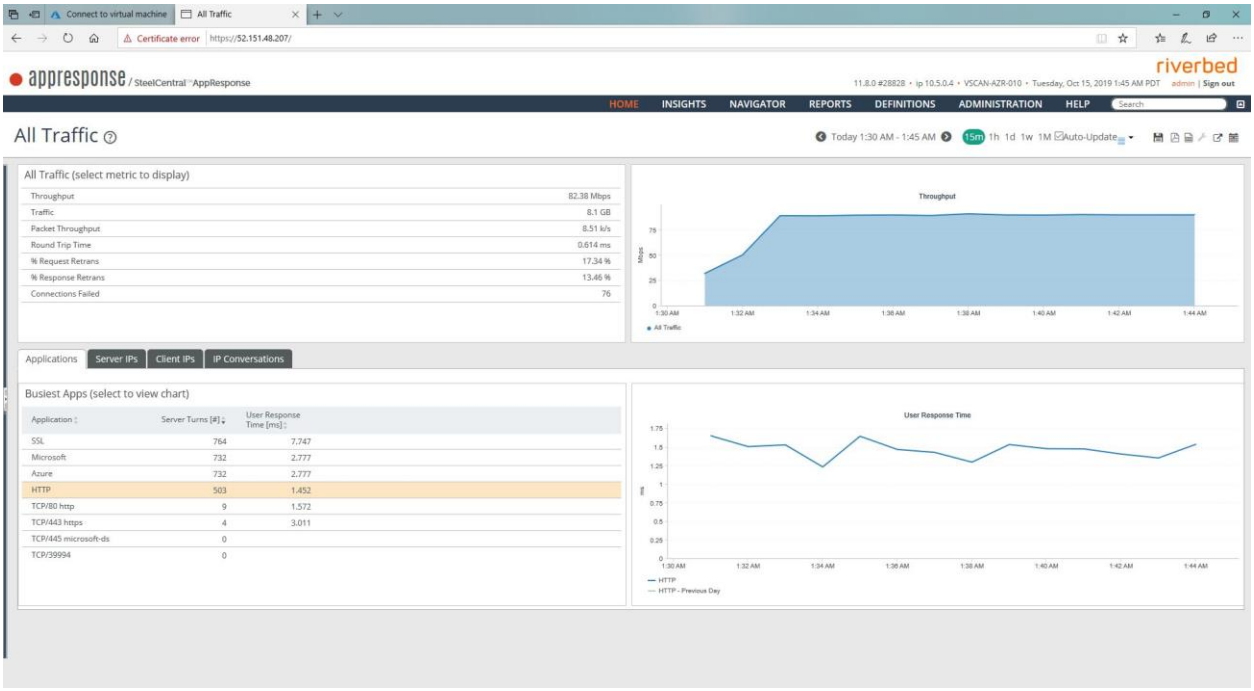


Step 4 – Drag a secure visibility path between source and tool groups



Step 5 – login to Riverbed AppResponse hosted in Azure

Verify traffic from Source Virtual Machines are available in Riverbed AppResponse



WHERE TO GET HELP

If you experience technical difficulties, please email cloudlens@keysight.com for assistance