KEYSIGHT CLOUDLENS WITH RIVERBED APPRESPONSE QUICKSTART GUIDE IN AZURE



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[™], Keysight's platform for public, private and hybrid cloud visibility addresses the challenges of granular data access in the cloud. CloudLens is a solution that provides network tap and packet brokering services in the cloud. 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 (but CloudLens is also avaibale in AWS, GCP or other clouds).

KEY CLOUDLENS FEATURES:

- Cloud visibility management is controlled by the cloud customer, not reliant on the cloud provider
- 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
- Reduces errors occurring due to complex and manual cloud 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
- Supports monitoring of Linux, Windows, and Containers
- Allows sharing of monitor traffic to multiple destinations.
- Supports monitoring of multi-cloud environments

ABOUT THIS GUIDE;

This guide is meant to summarize steps required for interoperability of Keysight CloudLens and Riverbed AppResponse Cloud. Not all details of every configuration step of each product is detailed here. Full product installation and user guides are available from <u>cloudlens.support@keysight.com</u> and <u>support@riverbed.com</u> respectively. This guide also assumes working familiarity with configuration of Azure. Examples shown in this guide were tested with Keysight CloudLens v6.0.2, and AppResponse Cloud v 11.11.5



SAMPLE DEPLOYMENT ARCHITECTURE

Shown above is a sample deployment, CloudLens Sensors run on customer Azure instances, register up to the CloudLens Manager (running in Customer's Azure Account) which manages the CloudLens Sensors. The CloudLens Sensors forward desired traffic to Riverbed AppResponse (also running in customer's Azure Account) via VxLAN.

Only two source instances are shown in this diagram, however many source instances are permitted (your CloudLens license determines now many CloudLens Sensors which the CloudLens manager is allowed to control. (see CloudLens documentation for instructions on Licensing)

MISCILLANEOUS REQUIREMENTS

In this example we are assuming the Source Virtual Machines already exist in customer's Azure account, you will load cloudlens sensors onto those virtual machiens as described on pages 3-5 of this document.

We are assuming Riverbed Appresponse Cloud has already been deployed into the customer's Azure account, plesae contact riverbed support for assistance if needed.

Specific Port rules must be configured on Virtual Machines to allow functioning of cloudlens, as well as flow of traffic between CloudLens sensors and Riverbed AppResponse Cloud. Please see Appendix on page 10 of this guide for further details.

INSTALL CLOUDLENS MANAGER

Prerequisites

- Obtain the CloudLens-Installer script referenced below from the Keysight Support site
- In your Azure account, deploy a Linux instance (e.g. Ubuntu) with 4 vCPUs, 16GB RAM, and at least 100GB storage is required.
- Copy the installer script onto the Linux Instance, the run the command as shown below (Note: If you try to install on an instance or VM that does not meet these requirements, the installer will prompt you to confirm installing on an under-spec instance, issues may occur)

Note: IF using CentOS 7 or RedHat 7, you must install SNAP before you install CloudLens Manager. For the procedure to install SNAP, see: <u>https://snapcraft.io/docs/installing-snap-on-centos</u>. After installing SNAP, you must reboot the host.

Installation

Executed steps on the instance or virtual machine # Upload the Cloudlens-Installer-<version>.sh from the release download page to the instance or virtual machine # Run the installer: chmod +x CloudLens-Installer-<version>.sh ./CloudLens-Installer-<version>.sh # Depending on the presence of the user performing the operation in the sudoers list, a prompt will appear to ask for the user's password. # After the installation finishes, wait 10-30 minutes for CloudLens Manager to become available, then use a browser to connect to CloudLens Manager at: https://<cl_ manager_vm_ip>. # Ensure that HTTPS (TCP port 443) is allowed between the host you are connecting from and the CloudLens Manager instance.

LOAD CLOUDLENS SENSORS ONTO SOURCE INSTANCES

Step 1 - Log into https://<ipaddress-cloudlens-manager/startup>

Note: default credentials are admin / Cl0udLens@dm!n

Step 2 - Create a new Project, give it a Name and click OK



Click 'Show Project Key' and keep a record of the value, you may need it later

CloudLens > GREG-AZURE		Account: Greg Copelan	DASHBOARD	CONFIGURE	ŵ	\$
DEFINE GROUP DESTINATIONS	LAUNCH AGENT	5 instances 2	groups <mark>3</mark> too	s 0.02 Mb		

Step 3 – click 'Launch Agent'

In this example we see the docker run command required for Linux, it is pre-populated with correct Project Key, and the IP Address of the CloudLens Manager – click the Icon to the right and copy the command to Notepad or similar.

Note: download link is also available for the Windows sensor installer (instructions not shown here, please consult CloudLens documentation for details)

LAUNCH AGEN		5 inst
ſ	START NEW AGENTS	
SROUPS	SSL Verify Enabled (requires a TLS certificate to be uploaded into CloudLens)	
	Linux agents	
5	Run this command:	
	\$ sudo docker run -v /lib/modules:/lib/modules -v /var/log:/var/log/cloudlens -v /:/host -v /var/run/docker.sock:/var/run/docker.sockprivilegedname cloudlens-agent -drestart=on-failurenet=hostlog-opt max-size=50mlog-opt max-file=3 10.4.0.4/sensoraccept_eula yesproject_key 4613fb7fad134cb1b747346d91cef0dfserver 10.4.0.4ssl_verify no	
	If you are deploying agents into Google Cloud or Amazon Cloud please also check: Collector Deploy Guide	
	Windows agents	
	Download and run any of the following executable files:	
	<u>cloudlens-win-agent.exe</u>	
		CLOSE

Step 4 – 'Connect' via SSH of Source Virtual Machine from Azure Console (or use RDP for Window), or using your favorite SSH client such as Putty



Then on the Linux Source host(s) and install Docker engine (if not already present)

e.g. in case of Ubuntu - commands may differ depending on Linux version

sudo apt update

sudo apt-get install -y docker.io

Step 5 - From the Linux Source host(s), run the docker commend which you saved in Step 3

Note: In a few minutes you will see the Instance counter in your Project increment, indicating that the Sensor(s) has successfully registered to the CloudLens Manager (in the example shown below 5 Instances were registered, numbers will differ depending on how many hosts you issue the docker run command)

CloudLens >	GREG-AZURE		Account	t: Greg Copela
DEFINE GROUP	DESTINATIONS	LAUNCH AGENT	5	instances

Once your CloudLens Sensors are successfully registered, you can proceed to the next section. If you have trouble registering your sensors, contact <u>cloudlens.support@keysight.com</u> before continuing

CONFIGURING CLOUDLENS TO SEND TRAFFIC TO APPRESPONSE

Step 1 - Log into https://<ipaddress-cloudlens-manager/startup>

Note: default credentials are admin / Cl0udLens@dm!n

- Then open your previously created Project
- Step 2 Define a Group(s) for your Source Instances
 - Click 'Define Group' from the Project Screen
 - Select the Filter criteria that best identifies the Source Instance(s) that you want 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)
 - Click 'Save Group'
 - Then choose 'Save as an Instance Group' give it a 'Name', then click OK

CloudLens > GREG-AZURE						CONFIGURE
ALL FILTERS ACTIVE FILTERS	1 instance	s Ø			SAVE G	ROUP
Ubuntu-CloudLens-Src	#	TAG: NAME n/a	TAG: TOOL	TAP ID 3e74d541ea	SYSTEM HOSTNAME	OPERATING SY ubuntu 18.04 t
Operating System ubuntu 18.04 bionic Memory (MB) min CPU Model	Choose Co	SAVE SEARCH Save as an in Save as a too Name Ubuntu-Srd Aggregation Inter Comment	л — — — — — — — — — — — — — — — — — — —			
FQDN ubuntu-cloudiens-src.internal.cloudapp.net ubuntu-cl-tool.internal.cloudapp.net IPv4			OK Cancel			

Step 3 - Configure a Destination to the IP address of AppResponse Cloud

Note: this may be the Private or Public IP address AppResponse Cloud, however is it assumed the Source instances have valid Azure route to reach the chosen IP address.

- Click 'Destinations' from the CloudLens Project screen
- Then Click 'New Static Destination'
 - o Give is a Name
 - o Specify the IP Address of AppResponse Cloud
 - Click OK

			Account:	Greg Copeland	DASHBOARD	CONFIGURE	Å 0
				NEW		ION CL	.OSE
RES	VLAN ID	TAGS					
125	ADD DESTIN	ATION					
103	Destinat Name	ion Enabled		ol: generic			
233	RiverbedApp	Response					
	IP Address		•				
	Tags:						
	Key type key	Value type value	+				
		ОК	Cancel				

- **Step 4** Configure your CloudLens Tool Group.
 - First Select the CloudLens Destination which you just configured in the last step

L FILTERS ACTIVE FILTERS		1 instand	ces 🛿				SAVE GROUP
 Riverbed-AppResponse 	•	#	TAG: NAME	TAG: TOOL	TAP ID	SYSTEM HOSTNAME	OPERATING SYSTEM
		1	Riverbed-AppResponse	n/a	a9af1e17a7	n/a	n/a
Tag: tool generic	- 1	Choose (Columns				

- Click 'Save Group'
 - Choose 'Save as a Tool'
 - o Give is a 'Name'
 - Specify 'Aggregation Interface' (by default this will be 'primary-vxlan')

1 instand	es 🛿				SAVE GROUP
#	TAG: NAME	TAG: TOOL	TAP ID	SYSTEM HOSTNAME	OPERATING SYSTE
1	Riverbed-AppResponse	n/a	a9af1e17a7	n/a	n/a
Choose (olumns				
		SAVE SEARCH			
		Save as an instance gr	QUD		
		 Save as a tool 			
		Name			
		RVBD-AppResponse			
		Aggregation Interface			
		primary-vxlan	_		
		Comment			
		ок	Cancel		

Note: You must specify the Aggregation Interface to match the name of the Monitor Interface in AppResponse Cloud (by default this name is '**primary-vxlan**' – however it may differ in your environment and the value can be checked in the Administrator settings of AppResponse)

			HOME INSIGHTS	NAVIGATOR TRANSACTIONS	REPORTS DEFINITIONS	ADMINISTRATION HELP
Capture Jobs	s/Interfaces @					
Capture Jobs Monitori	ng Interfaces Virtual Interface Groups					
Monitoring Inte	erfaces					
Name :	Description 0	Type ::	Link Status 🗄	Link Speed ::	Bytes Received :	Packets Received
primary-vxlan	AppResponse Interface (primary-vxlan)	10G Virtual	UP	10 Gbps Full Duplex	40170709	36542
General Configu	uration					
Packet Broker: None	▼					
Enable UDP Deduplie	cation					
Enable TCP Deduplic	tation (when IP ID = 0)					
Apply Re						

Step 5 – Drag a Connection path between source and tool groups

- Change the 'Encapsulation Protocol' from the default to 'VXLAN'
- Specify a VNI (if you create additional Connections later, you will need to specific unique VNIs)

CONNECTION PROP	RTIES		
source Ubuntu-Src	RVBD-		
Capture			
Traffic Filter (BPF s	(ntax)		
ip			
Traffic direction			
вотн	•		
Process			
Packet type			
RAW			
Deliver			
Encapsulation prot	ocol		
VXLAN	•		
VNI:	TOS:		
190	0		
	OK Cancel		
udLens > GREG-AZURE		Account: Greg Copeland DASHBOARD CONFIGUR	E 🐥
FINE GROUP DESTINATIONS	LAUNCH AGENT	5 instances 2 groups 3 tools 0 M	/bps tra
			PROJECT
INSTANCE GROUPS		MONITORING TOOL GROUPS	5
Ubuntu-Src		RVBD-AppResponse	-

Step 6 – login to Riverbed AppResponse hosted in Azure

Verify that network traffic occurring on Source Virtual Machine(s) is visible in Riverbed AppResponse

			HOME INSIGHTS	NAVIGATOR TRANSACTIONS	REPORTS DEFINITIONS ADMINISTRAT	ON HELP Search
raffic 🛛					🔇 Today 11:51 AM - 12:51 PM 👂 15m 🕕 1d	1w 1M 🗆 Auto-Update 🗸 🛛 🗎 🖄 🗎 🥖
User Respor ⊚ 17.8		Round Trip Time	Total Throughput 475.4 kbps	Connection Requests	Connections Failed	% Payload Retrans
3			Total T	hroughput		
2.5						
2						
sd qu 1.5						
1						
0.5						
0 12:48 PM		12:49 PM		12:50 PM		12:51 PM
Applications Server IF	Ps Client IPs IP Conve	rsations				
Application 0	Server Turns 🛊	Response Time Composition [ms] =			User Response Time	
TCP/443 https	26	25.401		40		
HTTP	26	2.728				
CNN	25	15.657		30		
TCP/80 http	14	0.065		Ĕ 20-		
SSL	10	32.999				
TCP/32526 TCP/23456	0	0		10		
SSH SSH	0	63.868		0		and an
3311					PM 12:10 PM 12:15 PM 12:20 PM 12:25 PM 1	2:30 PM 12:35 PM 12:40 PM 12:45 PM
IPv6-ICMP	0	0		 TCP/443 https 		

APPENDIX: CONFIGURE INBOUND PORT RULES IN NETWORKING

<u>Note:</u> Azure default for Outbound is open for All Traffic. But for Azure Virtual Machines Inbound Port Rules, a few ports numbers need to be explicitly opened to allow CloudLens and AppResponse to work together:

Source Virtual Machines:

- TCP 22 (if Linux) **
- TCP 3389 (if Windows) **
- HTTPS 443 open from IP address of CloudLens Manager

CloudLens Manager

- HTTPS 443 **

Riverbed AppResponse Virtual Machine:

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

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

** Specify IP addresses of customer administrators

WHERE TO GET HELP

If you experience technical difficulties, please email <u>cloudlens.support@keysight.com</u> for assistance