

Network Bandwidth Measurement & Monitoring Solution using Azure Monitoring (Service Map)

Intended for planning or monitoring SaaS/PaaS network connectivity.

Published: January 2019 Version: 3.0

> This document provides the steps to implement a solution which allows customers to remotely capture information from network connections so as to provide insights into connectivity and also monitor bandwidth utilization of machines or individual processes/services

© 2018 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. This

document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Table of Contents

1	In	troduction	2
2	Pr	rerequisites	3
3	Ar	nticipated Usage Scenarios	4
4	St	tep by Step Guide	6
	4.1	Login to Azure portal	6
	4.2	Creating the Log Analytics workspace	6
	4.3	Configure Log Analytics workspace for Service Map	7
	4.4	Client-Side Configuration	11
	4.5	Analyzing Data in Log Analytics	16
	4.6	Example Queries	22
	4.7	Adding graphs to an Azure Dashboard	25
	4.8	Building a Dashboard in Power BI	26
	4.9	Publish a Power BI Dashboard	29
	4.10	AutoRefresh Power BI Dashboard	30
5	Su	ummary	31

1 Introduction

As the world moves to the cloud, a critical piece of this transformation is around the network which connects users and businesses to their data and services. The path changes from a relatively simple line between the location of the users and the data/services, to a distributed model where data is in multiple places, and users are accessing it from a variety of locations and devices.

Planning physical connectivity for Microsoft cloud users should revolve around local egress as close to the user as possible so that the traffic can reach Microsoft's global network as quickly as possible. More detail on this specific to Office 365 can be found <u>here</u>.

Another major part of this transformation is planning how much bandwidth is required on the circuits which connect out of the enterprise. As user's data and services move from being hosted on the internal network, to being hosted in Microsoft datacenters, the path this data takes changes. in addition, the amount of data users send and receive changes as their available services increases or changes.

However, planning the amount of bandwidth can be a major challenge as every user and enterprise is different so it's impossible to give an average figure, what will be correct for one enterprise will be insufficient for another customer and far too much for another.

The traditional approach to solving this problem is a combination of monitoring pilot users and using process specific calculators to give an estimate based on assumed information or that obtained from pilot users. There are however multiple challenges with this model

- a. It's not always easy to obtain accurate data from pilot users
- b. Calculators provide an estimated answer, if incorrect information is entered, the result is therefore wrong. As the investment in bandwidth is normally a lengthy process which has an associated cost, it is imperative this information is correct so that the right long-term planning decisions can be taken.
- c. Calculators focus on a single service where the planning needs to be done at a higher level covering all services.

With these challenges in mind we have created a more scalable, simpler solution to allow the accurate collection and display of the data required to make informed decisions in this space, and to enable easier monitoring of connectivity of our clients as we move to the cloud.

This solution will allow you to monitor and analyse the following example scenarios:

- Bandwidth used for a particular process or set of processes over a set period of time
- Bandwidth used by the machine over a set period of time
- Bandwidth used in connections to a specific port
- Bandwidth used to a specific IP address or range of addresses
- IP geolocation of the endpoints connected to

The approach in this document can be used for:

- 1. Measuring network bandwidth usage for pilot users on-boarded to Office 365 or network bandwidth usage of on-premises users.
- 2. Endpoint monitoring dashboard post on-boarding users to Office 365

You can apply this concept for measuring any SaaS/PaaS traffic, not just Office 365.

2 Prerequisites

The solution in this document requires a number of components bought together to provide the desired outcome, the good news is they are all easy to implement and should take 30-60 minutes in total to set up. For reference, the requirements are:

1. An Azure Subscription

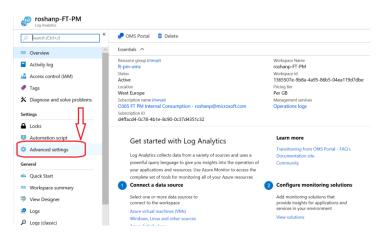
As the solution has Azure at its core, a valid Azure subscription with permissions to add components to that subscription (or the ability to ask an administrator to add them) is required. You can use your work or personal account to login to Azure, if this is for your organization then its recommended you use your work account and a subscription associate with your work account.

2. Azure Log analytics workspace

You could opt to use Log Analytics as an individual service and in this case, billing is based on the data ingested into the workspace (Per GB pricing). This features cloud-friendly, consumption-based pricing that includes a Free tier for testing. You only pay for what you use, here is the pricing.

3. Agent installs

a. Microsoft monitoring agent can be installed from the Azure portal itself by going to Log Analytics workspace and then the Advanced settings blade.



b. You will also need <u>Dependency agent</u> for Windows or Linux

4. Scaling this solution & connecting via OMS Gateway

Enterprise customers have scaled the agent deployment to thousands of workstations/servers by using SCCM. Enterprise customers also tend to use OMS Gateway as a gateway to ingest data to log analytics workspace if direct connection to the workspace from workstations/server is not an option.

5. <u>Resources at Github</u>

All resources referred in this document like scripts, templates are available at <u>Github</u>, there is also an INTRO pdf document for a high-level summary of this solution with visuals.

3 Anticipated Usage Scenarios

The usage scenarios for this solution depends on the network setup in place, or that which is possible for use.

Below are some anticipated scenarios:

A. Client has direct network connectivity (i.e. no proxy)

This is the simplest of scenarios, where the client/s being monitored connects directly connect to internet resources, i.e. they do a DNS lookup and directly connect to the public IP address in question. in this case we simply need to install the agents on the client and point them at our log analytics workspace. This method also works the same way when a transparent proxy is in place.

The Microsoft monitoring agent and Dependency agent is installed on the clients directly.

B. Client uses a proxy where the administrator can install the Agents on the proxy OS

If the clients to be monitored connect to the internet via a managed Linux proxy where the agents can be installed, for example a Squid proxy running on CentOS then we can simply monitor the clients by analysing the traffic leaving the proxy. Supported Linux versions can be found <u>here</u>.

It may be feasible for an organization to set up a temporary proxy of this type in Azure or on Premises simply for the monitoring of Bandwidth to specific endpoints, leaving standard browsing to traverse the standard web proxy.

C. Client uses a proxy where the administrator cannot install the agents.

In this scenario, where the user's only connection method to the internet is through a proxy solution where the agents cannot be installed to the host OS, then we need to get a little creative. One method is to open up secondary/tertiary ports on the existing proxy and edit the PAC file to send specific traffic we wish to monitor to that port.

For example:

- Port 8080 Default Web traffic port
- Port 8081 Exchange Online URLs
- Port 8082 SharePoint/OneDrive URLs

We can then monitor the bandwidth of traffic sent to the specific ports in Log analytics to give us an accurate view of traffic to the particular service. This usage scenario requires the agents to be installed on the clients we wish to monitor, as per scenario A.

D. Monitoring connectivity based on process vs destination IP's

One option which may be used is to monitor connectivity which originates from a single process. The above scenarios are necessary as some cloud services will have data accessed from a multitude of processes, it is therefore necessary to look at the destination address to capture the information for the service in question, regardless of the source process.

When an explicit proxy is used, the destination IP will always be the proxy IP for web bound traffic, thus we need to either capture on the proxy itself where we can see the destination IP to differentiate traffic, or we need to send known endpoints to a unique destination, such as a specific proxy port.

An example would be OneDrive for Business where access to data could come from the OneDrive Sync client, or any Office application, or the browser. We therefore need to filter based on the published SharePoint/OneDrive IP addresses to capture access from all sources.

However, some services will only be used by a single process in most cases, such as Outlook client being used for accessing Exchange. Technically OWA could be used in the browser but it would be known in advance if this is likely.

Therefore, for scenarios like this, the egress model does not matter as filtering can be performed against the Outlook.exe process and not the destination address which will differ based on the network egress model. This can also be a useful method for examples such as monitoring bandwidth usage to Exchange when on premises before any cloud deployment to give a view of current usage by the process, which also may aid in planning.

4 Step by Step Guide

4.1 Login to Azure portal

The first step is to login to the Azure portal for the account where you wish to host this solution. If you already have a tenant, simply login at https://portal.azure.com

If you don't already have one, or would like to test this solution in a new Azure instance, you can set up a free trial <u>here</u>.

4.2 Creating the Log Analytics workspace

- a. Once logged into the Azure Portal https://portal.azure.com/
- b. Start typing 'Log Analytics' in the search bar at the top, select 'Log Analytics' under Services
- c. Add a workspace if you don't have one already created
 - OMS Workspace is the name you want to call this instance/workspace
 - Subscription is the one you wish to use for the workspace (as you may have more than one available)
 - Resource group is simply a collection of resources, you can either use an existing one, or create a new one.
 - Location is which Azure region you host the instance in. It's important to select the right location. See the important note below.

 \leftarrow \rightarrow \circlearrowright a https://ms.portal.azure.com/#

Preview	Microsoft Azure	2		Report a bug	$\mathcal P$ log ana	×	Q	>_ ई
«		Dachhaard X - New dathbaard - Ilalaad - H Dawalaad - & Edit		ad 2 Edit A Shara	RESOURCES	0 results		
+ Create	a resource	$Dashboard \checkmark + New dashboard \overline{\star} Upload $		RESOURCE GROUPS	0 results			
E All services					SERVICES			
	Trees .	All resources ALL SUBSCRIPTIONS	Quick	starts + tutorials	💭 Log Analytics			

Home > Log Analytics > Log analytics workspace	
Log Analytics « * ×	Log analytics workspace Create new or link existing one created in OMS
Add Edit columns Filter by name NAME Image: Column to the second	 Create New C Link Existing OMS Workspace C enter workspace name Subscription CI Prod Official Resource group C Create new U Use existing Location West Europe Pricing tier Free

Notice the options for pricing tier, you can select from Free, Per Node or Per GB as explained previously.

Log analytics workspace X Create new or link existing one created in OMS Portal	Pricing Tier 🗖 🗙
 Create New Link Existing MS Workspace enter workspace name Subscription CI Prod Official Resource group Create new Use existing 	The cost of your workspace depends on the pricing tier and what solutions you use. Learn more about Log Analytics pricing. Pricing Tier Free Per Node (OMS) Per GB (Standalone)
 ★ Location West Europe ★ Pricing tier Free 	

IMPORTANT:

As per <u>this</u> article certain elements are only available if the OMS instance is hosted in certain regions. If you set the Location out outside these regions, you'll find your service fails to collect any data.

At the time of writing, the supported regions are:

- East US
- West Europe
- West Central US
- Southeast Asia

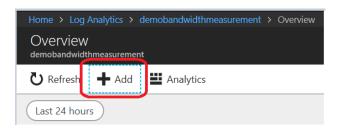
4.3 Configure Log Analytics workspace for Service Map

We will now configure Service Map in the workspace.

a. Go to the workspace you created in Log Analytics and click on 'workspace summary' on the left blade.

٩	demobandwidthmeasurement
	Overview
	Activity log
	Access control (IAM)
	P Tags
Home > Log Analytics > demobandwidthmeasurement	X Diagnose and solve problems
Log Analytics « 🖈 🗙 Microsoft	SETTINGS
Add 📑 Edit columns •••• More	Locks
	Automation script
Filter by name	Advanced settings
NAME TU	GENERAL
	🗳 Quick Start
demobandwidthmeasurement	Workspace summary

b. The workspace will be blank, click on 'Add' and select Service Map from the 'Management Solutions'.



Manag	gement Solutions 🛛 🖈 🗖 🗙
3	System Center Operations Manager Health Check (Preview) Microsoft
.	SQL Health Check Microsoft
0	Security and Audit Microsoft
Ŷ	Service Fabric Analytics Microsoft
10	Service Map Microsoft
•	Start/Stop VMs during off-hours Microsoft
E	Update Compliance Microsoft

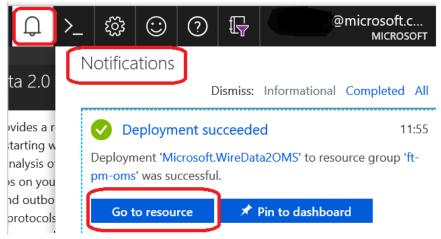
c. Select Service Map and click 'Create', ensure you have the right workspace selected

Service Map			Ŕ	
deliver services and re process dependencies components, service d eliminate the guesswo your environment, and	a view of your servers as y ly on other technologies. in real-time, without any lependencies, and support ork of problem isolation, id J perform Azure migration fervice Map supports Wind	Service Map discovers ar predefinition, and visuali ing infrastructure config entify surprise connectic s knowing that critical sy	nd maps server and zes application uration. This helps you ons and broken links in <i>r</i> stems and endpoints	
Diagnostic and Usage	Data			
Map service. Microsof the Service Map servic operating system and	ly collects usage and perfo t uses this Data to provide ce. Data includes informati version and also includes rate and efficient troubles!	and improve the quality on about the configurati IP address, DNS name, a	r, security and integrity on of your software lik nd Workstation name	e
addresses or other cor	ntact information. on data collection and usa	ige, please see the <mark>Micro</mark>	osoft Online Services	
addresses or other cor For more information Privacy Statement	on data collection and usa	ige, please see the Micro		
addresses or other con For more information Privacy Statement Save for later	on data collection and usa		admissione appoint statione internance	>
addresses or other cor For more information Privacy Statement.	on data collection and usa		Admidiance approv	> =
addresses or other con For more information Privacy Statement. Save for later Thereasters The	on data collection and usa		Admitterio-apport Mainter Inferioration CIVUUtorion	-
addresses or other con For more information Privacy Statement. Save for later Thereasters The	on data collection and usa	∆ingend E @, @,	Admitterio-apport Mainter Inferioration CIVUUtorion	=
addresses or other con For more information Privacy Statement. Save for later filter exclusion informations. Save for later informations. Save sources informations. Save sources inf	on data collection and usa	∆ingend E @, @,	Crutification Crutif	Enternany HEO
addresses or other con For more information Privacy Statement.	on data collection and usa	Lagent E Q Q	Americano-apport Malata Información CIU CIU CIU CIU CIU CIU CIU CIU	
addresses or other con For more information Privacy Statement Save for later	on data collection and usa	tine or the second terms of te	Americano-apport Malata Información CIU CIU CIU CIU CIU CIU CIU CIU	
addresses or other con For more information Privacy Statement. Save for later 10%2954 400 MM - 420 AM (0 MM Filter reactions) 10%2954 400 AM - 420 AM (0 MM Filter reactions) 10%2054 400 AM (0 MM Filter reactions)	on data collection and usa	toriose to	And Andream - Apport Mathematications Characteristic and a second seco	
addresses or other con For more information Privacy Statement.	on data collection and usa	tine or the second terms of te	Construction of the second secon	Here and the second sec

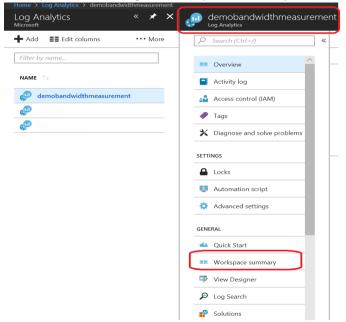
(ensure the the right workspace is selected)



Once the deployment succeeds, go to 'Service Map' by clicking on 'Go to resource' from the Notifications pane



- Find and add 'Network Performance Monitor'
- Find and add 'Service Map'
- d. You can also navigate to the resource (Service Map) by going to Log Analytics, selecting the workspace and clicking on 'workspace summary'



e. Your Solutions will then indicate they require further config, this is mainly the need for client setup and will change once it detects a client connected.

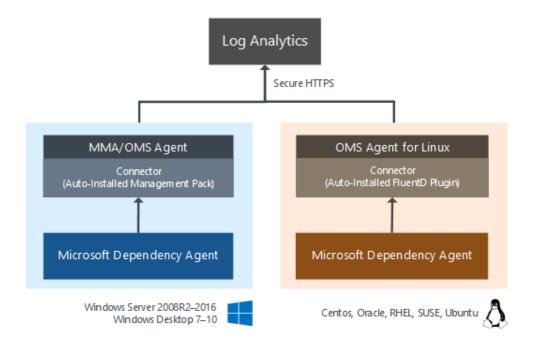
Home > ServiceMap(demobandwidthm	easurement)			
ServiceMap(demobandy	vidthmeasurement)			
✓ Search (Ctrl+/)	🗓 Delete			
I Overview	Essentials 🔨			
Activity log	Resource group	Solution ServiceMap(demobandwidthmeasuremen		
Access control (IAM)	Status Active	Type Microsoft.OperationsManagement/solutio		
✗ Diagnose and solve problems	Location West Europe	Workspace Name demobandwidthmeasurement		
Settings	Subscription name (change)	Management services Operations logs		
Locks	Subscription ID			
😫 Automation script				
General	Summary			
OMS Workspace	Service Map	Solution Resources		
Properties		1 🐢		
★ Saved searches	- 75	ServiceMap(demobandwidthmeasurement)		
Workspace Data Sources				
Virtual machines	Solution requires additional configuration			
Storage accounts logs				
Azure Activity log				
📴 Solution Targeting (Preview)				

And that's it for the server-side setup. Now we need to configure some clients to provide data to the solution.

4.4 Client-Side Configuration

To send the data into our new solution, we need to install two client-side applications 'Microsoft Dependency Agent' and 'Microsoft Monitoring Agent'

The dependency agent collects the data we need and sends it to the Monitoring agent for transmission to Log analytics workspace. You have access to the ingested data from your Log analytics workspace in Azure.



1. Install the <u>Dependency Agent</u> on the machines which you wish to monitor connectivity on. These can be installed on 64-bit modern windows versions, specifically

Windows Server

- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 SP1

Windows Desktop

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7

Install Link for Windows Machines.

The agent can also be installed on 64bit Linux variants, for example when we monitor at the proxy level.

More information on scripting this install and the supported Linux variants can be found here.

- 2. To obtain the correct Microsoft Monitoring Agent you'll need to go through the Azure Portal.
- a. In the Azure Portal click on "All Services" then search for 'Log Analytics' and then click on it



b. Select your workspace, click on "Advanced Settings" from the workspace blade

Microsoft Azı	ıre	Report a bug
*	Home > Log Analytics > demobandwidthmeasurement	
resource	Log Analytics « 🖈 🗙	demobandwidthmeasurement
es	➡ Add 📑 Edit columns •••• More	
	Filter by name	Overview
warehouses	NAME TU	Activity log
smos DB	demobandwidthmeasurement	Access control (IAM)
achines		🖉 Tags
ancers		X Diagnose and solve problems
ccounts		SETTINGS
tworks		Locks
tive Directory		Automation script
		Advanced settings
		GENERAL
Ienter		📣 Quick Start
agement + B		Workspace summary

c. Here we have both the install link for the Monitoring Agent we need, and also some key information to connect it to our Log analytics worksapce.

Microsoft Azure				
«				
+ Create a resource	Advanced settings demobandwidthmeasurement			
i ≡ All services	🐌 Refresh 🗮 Analytics			
★ FAVORITES	A			Windows Servers
A PAVORIES	Connected Sources	Windows Servers	>	Attach any Windows server or client.
🛄 Dashboard				0 WINDOWS COMPUTERS CONNECTED
All resources	🖉 Data 💙	Linux Servers	>	Download Windows Agent (64 bit) Download Windows Agent (32 bit)
Resource groups	Computer Groups	Azure Storage	>	You'll need the Workspace ID and Key to install the agent.
🔇 App Services				WORKSPACE ID
Function Apps		System Center	>	22/280/
🥫 SQL databases				PRIMARY KEY
🥭 Azure Cosmos DB				14Ctem7 Regenerate

- d. Download and install the appropriate Windows Agent (64 bit) and install it on the client/s you wish to monitor
- e. Also copy the Workspace ID and Primary Key.
- f. Once the monitoring agent is installed, a window will pop up. We require the middle option to connect to our OMS instance. Select and click next.

🛃 Microsoft Monitoring Agent Setup	\times
Agent Setup Options	
Specify setup options for this installation of Microsoft Monitoring Agent.	
Enable local collection of IntelliTrace logs (requires .NET Framework 3.5 or higher) This installs a PowerShell interface for gathering advanced application diagnostics data in local iTrace files.	
Connect the agent to Azure Log Analytics (OMS)	
Connects the agent to the Microsoft Azure Log Analytics (OMS) service and lets you to choose the workspace that the agent uses to register with. For more information, see https://www.microsoft.com/oms.	
Connect the agent to System Center Operations Manager	
This connects the agent to System Center Operations Manager and lets you specify the management group for which this agent will participate in monitoring.	
< Back Next > Cancel	

- g. Next, we'll be asked for our Workspace ID and Primary Key we copied from the Azure Portal above. Most customers will require the Azure Commercial instance. If the client uses a proxy to connect to the internet, then this need configuring by clicking the advanced button.
- h. Click next and select whether to use Windows Update for the agent (recommended) then next again.
- To check installation succeeded, open the control panel and select Microsoft Monitoring Agent. If successfully connected you should see a green tick in the status bar on the 'Azure Log Analytics' Tab.

Microsoft	Monitor	ring Agent Properties				
perations N	Manager	Azure Log Analytics (OMS)	Proxy Settings	Properties		
can add, e	dit, check	pring Agent can report to mul the status of, or remove wo rs directly to Log Analytics.				
Workspace	s: Status			Agent Id		
22f2		Microsoft Monitoring Agent ha	C I	-		
			Add	. E	dit	Remove
			(Ж	Cancel	Apply

j. And that's it! Sit back and wait for your client to talk to Log analytics workspace and ingest data which should normally be within 5-10 minutes.

4.5 Analyzing Data in Log Analytics

After about 5 minutes you should be able to see your machine is reporting into Log analytics by logging into your portal. You can do this in two ways.

a. Log into your Azure Portal at https://portal.azure.com click on 'All Resources' and click on your Log Analytics workspace, if you go to the 'Workspace Summary' you will notice that Service Map has been configured and collecting data

Microsoft Azur	e		Report a
« resource	Home > Log Analytics > demoban Log Analytics Microsoft	dwidthmeasurement « 🖈 🗙	demobandwidthmeasuren
es	Add ■■ Edit columns Filter by name	•••• More	Search (Ctrl+/) Overview
warehouses	NAME TU		Activity log
smos DB	demobandwidthmeasurer	ment	Access control (IAM)
achines			 Tags Diagnose and solve problems
accounts			SETTINGS
etworks			Locks
tive Directory			 Automation script Advanced settings
Center			GENERAL Quick Start
nagement + B			E Workspace summary
Microsoft Azure			eport a bug
« esource	Dashboard > Log Analytics Overview roshanp-ft-pm	> roshanp-FT-PM	> Overview
t s	🕐 Refresh 🕇 Add 🗮	Analytics	
	Last 24 hours		
ces	ser		×
groups ces	Service Map		*
ases	1	3	

All-time machines reporting

3 🚺 0 🔬

Machines reporting

(Last 30 min)

warehouses

mos DB chines

- b. Then click on 'Analytics' at the top of the page which opens up in a new tab. If you click on Service Map tab it will show you a summary of network traffic information collected. We will use Analytics so you can run queries against the workspace to extract specific information we need for Bandwidth.
- c. There are a couple of things to take note there if you are using Log analytics for the first time.

Notice you are now in Log analytics playground, you can use this console to run log analytics queries and review data that is stored in the workspace. You also have the ability to specify time range or save queries for frequent use in 'Query Explorer'

Dashboard > Log Analytics > roshanp-	T-PM > Overview > Logs	
Logs roshanp-ft-pm		\$ ×
New Query 1 +		Help 🖏 Settings 🏳 Query explorer
roshanp-ft-pm	► Run Time range: Last 24 hours	🔚 Save 👁 Copy link 🕒 Export 🕂 New alert rule 🖈 Pin
Schema Filter (preview) ≪ Filter by name or type ∅ JF Collapse all Active • Ø roshanp.ft.pm • LopManagement ★	Type your query here	
 LogManagement NetworkMonitoring Office185 ServiceMap WireData2 Custom Logs <i>f</i> Functions Favorite workspaces <i>i</i> poshanp-ft-pm 	Select queries Heartbeat Performance Usage Chart the number of reporting computers each hour Heartbeat summarize dcount(ComputerIP) by bin(TimeGenerated, 1h) render timechart List all computer heartbeats from the last hour Heartbeat where TimeGenerated > ago(1h)	Learn more ☐ Documentation ➢ Online course Query Language & [®] Community ■ What's new

d. Run a simple query like the following, replace 'DESKTOP' with your computer name. Notice the time range by default is set to Last 24 hours.

VMConnection | where Computer contains "DESKTOP"

| where ProcessName contains "outlook"

Run Time ra	nge: Last hour	<				🔚 Save 👁 Cop	y link 🔁 Expo	ort 🕂 New alert r
MConnection where Computer conta where ProcessName co								
							-	
	olumns ~						Ø	
Completed. Showing result	olumns ✓				7			Display time (U
TABLE , I CHART C	olumns ✓		✓ ProcessName	V Sourcelp 7	7 DestinationIp	√ DestinationPort	✓ Protocol	
TABLE ,ICHART C	p it here to group by that co		ProcessName OUTLOOK	∑ Sourcelp ∑ 192.168.1.14	 DestinationIp 52.97.134.6 	DestinationPort 443		Display time (U
TABLE , I CHART C Drag a column header and drop TimeGenerated [UTC]	columns ~ p it here to group by that co V Computer V ROSHWORK-DXB	lirection	p Trocessivanie	,	bostinationip	,	7 Protocol	Display time (U
TABLE II CHART C orag a column header and drop TimeGenerated [UTC] Yours of the column	iolumns ∨ p it here to group by that cc ∇ Computer ∇ D ROSHWORK-DXB or ROSHWORK-DXB or	lirection	OUTLOOK	192.168.1.14	52.97.134.6	443	Protocol tcp	Display time (U Remotelp 52.97.134.6

If the query runs successfully you should see the results showing the data table and records. In the example above we are also setting the time range for the last one hour.

e. Service Map has the following records we can search from, you can see this in the Schema on the left hand side, expand the Service Map, VMConnection table

Logs roshanp-ft-pm		
New Query 1* +		
roshanp-ft-pm	- <mark>-0</mark>	Run (Time range: Last hour)
Schema Filter (preview)	«	VMConnection where Computer contains "roshwork" where ProcessName contains "outlook"
Filter by name or type	Q	where recessivalite contrains outlook
ļ≓ Colla	anse all	
Active	^	
🔻 💷 roshanp-ft-pm	*	
 LogManagement 		Consultated Chausing group to form the last hours
 NetworkMonitoring 		Completed. Showing results from the last hour.
 Office365 		TABLE ,II CHART Columns ~
 ServiceMap 		Drag a column header and drop it here to group by tha
VMBoundPort		TimeGenerated [UTC] ∇ Computer ∇
 VMConnection 		
t AgentId		> 2018-11-14T08:07:12.934 ROSHWORK-DXB
<pre># BytesReceived</pre>		> 2018-11-14T08:07:12.934 ROSHWORK-DXB
# BytesSent		
 t Computer t Confidence 		> 2018-11-14T08:07:12.934 ROSHWORK-DXB
t ConnectionId		> 2018-11-14T08:07:12.934 ROSHWORK-DXB
t Description		> 2018-11-14T08:07:12.934 ROSHWORK-DXB
t DestinationIp		
# DestinationPort		> 2018-11-14T08:08:12.934 ROSHWORK-DXB
t Directiont FirstReportedDateTime		> 2018-11-14T08:08:12.934 ROSHWORK-DXB
<i>t</i> IndicatorThreatType		> 2018-11-14T08:08:12.934 ROSHWORK-DXB
t IsActive		
t LastReportedDateTime		> 2018-11-14T08:08:12.934 ROSHWORK-DXB

Service Map has this information which is aggregated in 1 minute intervals. For example, a single TCP session will have the bytes sent/received/total for the previous 1 minute logged as an entry. This enables us to query the data for information such as the following examples:

- Bandwidth used for a particular process or set of processes over a set period of time
- Bandwidth used by the machine over a set period of time
- Bandwidth used in connections to a specific port
- Bandwidth used to a specific IP address or range of addresses
- Number of TCP connections a process has open at any one time
- IP geolocation of the endpoints connected to

Essentially, if we know the process name or the destination IP address we can easily monitor the endpoints used for any of the fields above.

From an Office 365 perspective this allows us to collect very accurate data for bandwidth usage for pilot users to gain an accurate figure for planning for a wider rollout.

For example, you may wish to see the bandwidth usage of the Outlook client over the course of 24 hours to see how much bandwidth your Exchange connectivity may require.

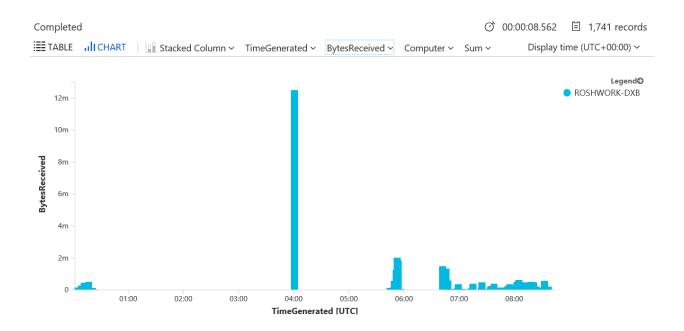
To do this we query Service Map for a specific PC (or multiple if required) in this case a machine called 'DESKTOP' where the process name contains 'Outlook', where the TCP session start time was in the last day and output the total bytes used in each sample of 1 minute (TimeGenerated).

Add the above to the query then hit 'Run'. You need to ensure the timeframe for the query is set correctly at the top, in this case as we're specifying one day, it needs to be at least 'Last 24 Hours' which allows us to search through the data for the last 24 hours or you can specify the time range in the query itself as shown below

	let startTime=datetime(2018-11-14T00:01:00);								
	let endTime=datetime(2018-11-14T11:59:59);								
	VMConnection								
	where Computer contains "roshwork"								
	where Proces	sName contai	ins "outlook'	ı					
	where TimeG	enerated > sta	artTime and	TimeGenerated	l < endTime				
	Run Time range	e: Set in query		📙 Save 🤏	o Copy link 🗧	Export 🕂 New al	ert rule 🔗 Pin		
<pre>let startTime=datetime(2018-11-14T00:01:00); let endTime=datetime(2018-11-14T11:59:59); VMConnection where Computer contains "roshwork" where ProcessName contains "outlook" where TimeGenerated > startTime and TimeGenerated < endTime</pre>									
			imeGenerated <	: endTime					
ļν			imeGenerated <	c endTime		⑦ 00:00:01.399 E	리 1,699 records		
ί v Cc	where TimeGenerated > :		imeGenerated <	c endTime		•	린 1,699 records e (UTC+00:00) ~		
Cc	where TimeGenerated > : ompleted	startTime and T umns ~		c endTime		•			
Cc	where TimeGenerated > : ompleted TABLE , CHART Colu	startTime and T umns ~		c endTime ProcessName ∽	Sourcelp 🗸	•			
Ccc Dra	where TimeGenerated > s ompleted TABLE ,IICHART Colu ag a column header and drop it	startTime and T umns ~ here to group by that	t column			Display time	e (UTC+00:00) ∨		
Ccc Dra >	where TimeGenerated > sompleted TABLE ,, CHART Colu ag a column header and drop it TimeGenerated [UTC] 7	startTime and T umns ~ here to group by that Computer ∇	t column Direction 🛛 🏹	 ProcessName ∽	Sourcelp 🖓	Display time	DestinationPort		
Ccc Dra >	where TimeGenerated > sompleted TABLE ILICHART Colu ag a column header and drop it TimeGenerated [UTC] 2018-11-14T08:22:12.936	Imms ~ here to group by that Computer ∇ ROSHWORK-DXB	t column Direction \bigtriangledown outbound	 ProcessName OUTLOOK	Sourcelp 7 192.168.1.14	Display time DestinationIp 52.97.131.130	e (UTC+00:00) ~ DestinationPort 443		

If the query worked, you'll see a table of data output. We can then turn this into a graph within Analytics by clicking 'Chart' then when rendered, click 'Stacked Column' and change to Line or any other graph you prefer.

In the graph produced, we can clearly see the bytes used by the Outlook process for the time range specified on this machine. This allows us to see the peaks and normal usage and thus allow us to plan for bandwidth across all our clients.



If we just want to find out the key points, i.e. 95th percentile of sent/received Kbps and Total sent/received MB over a time period, we can run the following query:

```
let startTime=datetime(2018-11-14T00:01:00);
let endTime=datetime(2018-11-14T11:59:59);
VMConnection
| where Computer contains "desktop"
| where ProcessName contains "outlook"
| where TimeGenerated > startTime and TimeGenerated < endTime
                                         (sum(BytesSent)/1024*8/60),
     summarize
                    SentKbps
                                                                        ReceivedKbps
                                  =
                                                                                           =
(sum(BytesReceived)/1024*8/60) by bin(TimeGenerated, 1m), Computer
summarize Percentile95SentKbps = round(percentile(SentKbps, 95),2), Percentile95ReceivedKbps
= round(percentile(ReceivedKbps, 95),2) by Computer
| join (
  VMConnection
  | where Computer contains "desktop"
  | where ProcessName contains "outlook"
  | where TimeGenerated > startTime and TimeGenerated < endTime
      summarize
                    TotalSentMB
                                         sum(BytesSent)/1024/1024,
                                                                      TotalReceivedMB
                                    =
                                                                                           =
sum(BytesReceived)/1024/1024 by Computer
) on Computer
    project
              Computer,
                           Percentile95SentKbps,
                                                   Percentile95ReceivedKbps,
                                                                               TotalSentMB,
TotalReceivedMB
```

	ompleted ≣TABLE , , CHART │ Columns ~					O:00:01.974	曽 1 records
D	rag a column header and drop it here to g	roup by tha	t column				
	Computer 🛛 🖓 Percentile95SentKb	ps 🖓	Percentile95ReceivedKbps	\bigtriangledown	TotalSentMB 🛛 🖓	TotalReceivedMB 🛛 🖓	
~	ROSHWORK-DXB 78		67		21	34	
	Computer	ROSHWC	ORK-DXB				
	Percentile95SentKbps	78					
	Percentile95ReceivedKbps	67					
	TotalSentMB	21					
	TotalReceivedMB	34					

This method can be used for any process running on the machine and will work regardless of whether a proxy is used or not because we're monitoring the process, not the IP address of the endpoint (which will always be the proxy IP if one is used for all non-internal traffic).

4.6 Example Queries

The data can be analyzed to a limited extent within the Analytics portal itself. For more comprehensive analysis we can use Power BI which is covered in the next session. Regardless, the Power BI query can be obtained within Azure Analytics via the example queries below.

*Time is defined in UTC by default, you can change this by going to the 'Settings' tab

Help 🐯 Settings	Query explorer
🔚 Save 👁 Copy link 📑 Export 🕂 New alert rule 🖈 Pin	Filter by name or path

*Full Query = Give me all the data and I will perform my own query to filter it

*Key Points Query = Filter the data and display only the key points

EXO Full Query (Exchange Online)

let startTime=datetime(2018-09-03T00:01:00); let endTime=datetime(2018-09-07T23:59:59); VMConnection | where Computer contains "XXXX" | where ProcessName contains "outlook" | where TimeGenerated > startTime and TimeGenerated < endTime</pre>

EXO Key Points Query

let startTime=datetime(2018-09-03T00:01:00); let endTime=datetime(2018-09-07T23:59:59); VMConnection | where Computer contains "XXXX" | where ProcessName contains "outlook" | where TimeGenerated > startTime and TimeGenerated < endTime</pre>

| summarize SentKbps = (sum(BytesSent)/1024*8/60), ReceivedKbps = (sum(BytesReceived)/1024*8/60) by bin(TimeGenerated, 1m), Computer | summarize Percentile95SentKbps = round(percentile(SentKbps, 95),2), Percentile95ReceivedKbps = round(percentile(ReceivedKbps, 95),2) by Computer | join (VMConnection | where Computer contains "robin" | where ProcessName contains "outlook" | where TimeGenerated > startTime and TimeGenerated < endTime summarize TotalSentMB sum(BytesSent)/1024/1024, TotalReceivedMB = = sum(BytesReceived)/1024/1024 by Computer) on Computer | project Computer, Percentile95SentKbps, Percentile95ReceivedKbps, TotalSentMB, TotalReceivedMB

The query logic is the same for Skype for Business and Teams too but just replaced 'outlook' with 'lync' or 'teams'

The query logic for **SPO (Sharepoint Online)** and **OD4B (OneDrive for Business)** is as follows since it is based on destination IP ranges:

SPO/OD4B Full Query

let startTime=datetime(2018-09-02T01:00:00);						
let endTime=datetime(2018-09-07T23:30:00);						
VMConnection						
where Computer contains "XXXX"						
where TimeGenerated > startTime and TimeGenerated < endTime						
where (parse_ipv4(DestinationIp) >= parse_ipv4('13.107.6.150') and parse_ipv4(Dest	tinationlp) <=					
parse_ipv4('13.107.6.151'))						
or (parse_ipv4(DestinationIp) == parse_ipv4('13.107.6.168'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('13.107.9.150') and parse_ipv4(Destin	nationlp) <=					
parse_ipv4('13.107.9.151'))						
or (parse_ipv4(DestinationIp) == parse_ipv4('13.107.9.168'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('13.107.136.0') and parse_ipv4(Destin	nationlp) <=					
parse_ipv4('13.107.139.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('40.108.0.0') and parse_ipv4(Destin	nationIp) <=					
parse_ipv4('40.108.31.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destin	nationlp) <=					
parse_ipv4('40.108.255.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destin	nationIp) <=					
parse_ipv4('52.107.255.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destin	nationIp) <=					
parse_ipv4('104.146.31.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Desti	nationIp) <=					
parse_ipv4('104.146.255.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp)	nationIp) <=					
parse_ipv4('134.170.207.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Desti	nationIp) <=					
parse_ipv4('134.170.215.255'))						

or	(parse_ipv4(Destinationlp)	>=	parse_ipv4('150.171.40.0')	and	parse_ipv4(DestinationIp)	<=
ра	rse_ipv4('150.171.43.255'))					
or	(parse_ipv4(DestinationIp)	>=	parse_ipv4('191.232.0.0')	and	parse_ipv4(Destinationlp)	< =
pai	rse_ipv4('191.232.1.255'))					
or	(parse_ipv4(DestinationIp)	>=	parse_ipv4('191.235.0.0')	and	parse_ipv4(DestinationIp)	<=
ра	rse_ipv4('191.235.15.255'))					

SPO/OD4B Key Points Query

let startTime=datetime(2018-09-02T01:00:00);	
let endTime=datetime(2018-09-07T23:30:00);	
VMConnection	
where Computer contains "XXXX"	
where TimeGenerated > startTime and TimeGenerated < en	dTime
where (parse_ipv4(DestinationIp) >= parse_ipv4('13.10	
	07.6.150') and parse_ipv4(DestinationIp) <=
parse_ipv4('13.107.6.151')) or (parse_ipv4(DestinationIp) == parse_ipv4('13.107.6.168'))	
	1501) and many in ((Destination la)
or (parse_ipv4(DestinationIp) >= parse_ipv4('13.107.9	0.150') and parse_ipv4(DestinationIp) <=
parse_ipv4('13.107.9.151'))	
or (parse_ipv4(DestinationIp) == parse_ipv4('13.107.9.168'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('13.107.1	136.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('13.107.139.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('40.108	.0.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('40.108.31.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('40.108.1	128.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('40.108.255.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('52.104	.0.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('52.107.255.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146	5.0.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('104.146.31.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146."	128.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('104.146.255.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.2	200.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('134.170.207.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.2	208.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('134.170.215.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.	.40.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('150.171.43.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232	2.0.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('191.232.1.255'))	
or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235	5.0.0') and parse_ipv4(DestinationIp) <=
parse_ipv4('191.235.15.255'))	
summarize SentKbps = (sum(BytesSent)/1024*8/60), Recei	ivedKbps = (sum(BytesReceived)/1024*8/60) by
bin(TimeGenerated, 1m), Computer	
summarize Percentile95SentKbps = round(percentile(Set	entKbps, 95),2), Percentile95ReceivedKbps =
round(percentile(ReceivedKbps, 95),2) by Computer	
join (
VMConnection	
where Computer contains "XXXX"	
where TimeGenerated > startTime and TimeGenerated <	endTime
where (parse_ipv4(DestinationIp) >= parse_ipv4('13.	
parse_ipv4('13.107.6.151'))	

or (parse_ipv4(Destinationlp) == parse_ipv4('13.107.6.168')) or (parse_ipv4(Destinationlp) >= parse_ipv4('13.107.9.150') and parse_ipv4(Destinationlp) <= parse_ipv4(Destinationlp) >= parse_ipv4('13.107.9.168')) or (parse_ipv4(Destinationlp) >= parse_ipv4('13.107.136.0') and parse_ipv4(Destinationlp) <= parse_ipv4('13.107.139.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('10.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.25.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) J summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer								
<pre>parse_ipv4('13.107.9.151')) or (parse_ipv4(Destinationlp) == parse_ipv4('13.107.9.168')) or (parse_ipv4(Destinationlp) >= parse_ipv4('13.107.136.0') and parse_ipv4(Destinationlp) <= parse_ipv4('13.107.139.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.25.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('52.107.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.</pre>	or (parse_ipv4(Destinationlp) ==	parse_	ipv4('13.107.6.168'))				
or (parse_ipv4(Destination p) == parse_ipv4('13.107.9.168')) or (parse_ipv4(Destination p) >= parse_ipv4('13.107.136.0') and parse_ipv4(Destination p) <= parse_ipv4(13.107.139.255')) or (parse_ipv4(Destination p) >= parse_ipv4('40.108.0.0') and parse_ipv4(Destination p) <= parse_ipv4('0.108.31.255')) or (parse_ipv4(Destination p) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destination p) <= parse_ipv4('0.108.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destination p) <= parse_ipv4('52.107.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destination p) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destination p) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destination p) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destination p) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destination p) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.232.0') and parse_ipv4(Destination p) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.235.0') and parse_ipv4(Destination p) <= parse_ipv4('191.235.15.255')] summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer	or	(parse_ipv4(Destinationlp)	>=	parse_ipv4('13.107.9.150')	and	parse_ipv4(Destinationlp)	<=	
or(parse_ipv4(Destinationlp)>=parse_ipv4('13.107.136.0')andparse_ipv4(Destinationlp)<=	pars	se_ipv4('13.107.9.151'))						
parse_ipv4('13.107.139.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.00') and parse_ipv4(Destinationlp) <= parse_ipv4(Uotol8.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4(Destinationlp) >= parse_ipv4('52.104.00') and parse_ipv4(Destinationlp) <= parse_ipv4(Destinationlp) >= parse_ipv4('104.146.00') and parse_ipv4(Destinationlp) <= parse_ipv4(Destinationlp) >= parse_ipv4('104.146.00') and parse_ipv4(Destinationlp) <= parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4(104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4(134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) [summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer	or (or (parse_ipv4(DestinationIp) == parse_ipv4('13.107.9.168'))						
or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024/1024/1024/1024/1024/1024	or	(parse_ipv4(DestinationIp)	>=	parse_ipv4('13.107.136.0')	and	parse_ipv4(Destinationlp)	<=	
parse_ipv4('40.108.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('40.108.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('40.108.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('14.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('161.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')] or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')] summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024/1024/1024/1024/1024/1024	par	se_ipv4('13.107.139.255'))						
or (parse_ipv4(DestinationIp) >= parse_ipv4('40.108.128.0') and parse_ipv4(DestinationIp) <= parse_ipv4('40.108.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('52.104.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('52.107.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.128.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.0.1') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.0.1') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.0.1') and par	or	(parse_ipv4(DestinationIp)	>=	parse_ipv4('40.108.0.0')	and	parse_ipv4(DestinationIp)	<=	
parse_ipv4('40.108.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destination p) <= parse_ipv4('52.107.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destination p) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destination p) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destination p) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destination p) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destination p) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destination p) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destination p) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destination p) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destination p) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destination p) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destination p) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destination p) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destination p) >= parse_ipv4('104.1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer	par	se_ipv4('40.108.31.255'))						
or (parse_ipv4(Destinationlp) >= parse_ipv4('52.104.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('52.107.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) j summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024 by Computer) on Computer	or	(parse_ipv4(DestinationIp)	>=	parse_ipv4('40.108.128.0')	and	parse_ipv4(Destinationlp)	<=	
parse_ipv4('52.107.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.128.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) j summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024, by Computer) on Computer	pars							
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.31.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.128.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024/1024, by Computer) on Computer			>=	parse_ipv4('52.104.0.0')	and	parse_ipv4(DestinationIp)	<=	
<pre>parse_ipv4('104.146.31.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('104.146.128.0') and parse_ipv4(Destinationlp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.200.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) </pre>	pars							
or (parse_ipv4(DestinationIp) >= parse_ipv4('104.146.128.0') and parse_ipv4(DestinationIp) <= parse_ipv4('104.146.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('104.146.0.0')	and	parse_ipv4(DestinationIp)	<=	
<pre>parse_ipv4('104.146.255.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (DestinationIP) == parse_ipv4('1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer</pre>	pars							
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.200.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.207.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('104.146.128.0')	and	parse_ipv4(Destinationlp)	<=	
<pre>parse_ipv4('134.170.207.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('134.170.208.0') and parse_ipv4(Destinationlp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) </pre>	par							
or (parse_ipv4(DestinationIp) >= parse_ipv4('134.170.208.0') and parse_ipv4(DestinationIp) <= parse_ipv4('134.170.215.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('134.170.200.0')	and	parse_ipv4(Destinationlp)	<=	
<pre>parse_ipv4('134.170.215.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('150.171.40.0') and parse_ipv4(Destinationlp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.232.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (parse_ipv4(Destinationlp) >= parse_ipv4('191.235.0.0') and parse_ipv4(Destinationlp) <= parse_ipv4('191.235.15.255')) or (Destinationlp) = parse_ipv4('1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer</pre>	par							
or (parse_ipv4(DestinationIp) >= parse_ipv4('150.171.40.0') and parse_ipv4(DestinationIp) <= parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('134.170.208.0')	and	parse_ipv4(Destinationlp)	<=	
<pre>parse_ipv4('150.171.43.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer</pre>	par	•						
or (parse_ipv4(DestinationIp) >= parse_ipv4('191.232.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('150.171.40.0')	and	parse_ipv4(Destinationlp)	<=	
<pre>parse_ipv4('191.232.1.255')) or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer</pre>	•							
or (parse_ipv4(DestinationIp) >= parse_ipv4('191.235.0.0') and parse_ipv4(DestinationIp) <= parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('191.232.0.0')	and	parse_ipv4(Destinationlp)	<=	
parse_ipv4('191.235.15.255')) summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer	par							
summarize TotalSentMB = sum(BytesSent)/1024/1024, TotalReceivedMB = sum(BytesReceived)/1024/1024 by Computer) on Computer			>=	parse_ipv4('191.235.0.0')	and	parse_ipv4(Destinationlp)	<=	
by Computer) on Computer		•						
) on Computer			(Bytes	Sent)/1024/1024, TotalReceiv	/edMB	= sum(BytesReceived)/1024/	1024	
	-	•						
project Computer, Percentile95SentKbps, Percentile95ReceivedKbps, TotalSentMB, TotalReceivedMB		•						
	pro	oject Computer, Percentile95S	SentKb	ps, Percentile95ReceivedKbp	s, Total	SentMB, TotalReceivedMB		

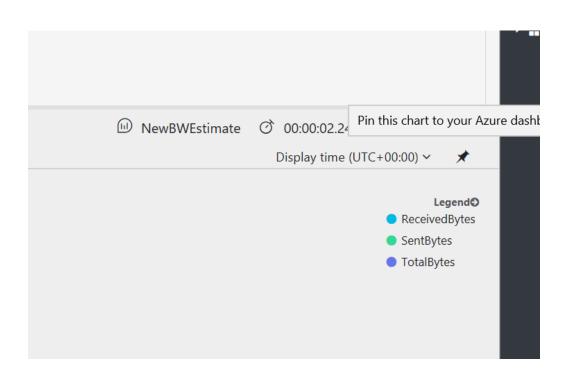
Only for SPO/OD4B the highlighted section in orange may need to be replaced if the IP ranges used for SPO/OD4B changes, the ranges are published in https://aka.ms/o365ip

You can use the Powershell script Generate-LAQuery-v3.ps1 in <u>Github</u> to generate the query for the highlighted section only. Syntax examples are available in the comments section of the file.

4.7 Adding graphs to an Azure Dashboard

With each of the above queries, when you create a graph to show the information, you have the option to pin the graph to an Azure dashboard

Once the table is displayed, click 'Chart' then select the view you wish. Once the graph is displayed, simply click on the pin icon in the top right-hand corner of the graph



You can add multiple graphs to create a dashboard to show the data you need to view as soon as you log into Azure. These graphs can easily be refreshed to keep them current without having to run the query in Azure Analytics again.

Bandwidth Measurement $\checkmark \ + \ {\tt New \ dashboard} \ \ {\color{red}{\bf \pi}} \ {\tt Upload} \ \ {\color{red}{\bf \pm}}$	Download 🥒 Edit. Q Unshare 🖉 Full screen 🧬 Clone 🛞 Defete		
And Andrew Andre		Besite Spin - Outlook Announ mit Monotoni Log mit 3 SECOM Control Contro	Codok Process Terra ante caso over para locales Serve Terra ante caso over para locales esse to esse to
Outlook proces the construction const avecus	Machine Bandwidth Usage Useron Description		

4.8 Building a Dashboard in Power BI

Whilst Azure Analytics provides some great features to quickly analyse the data, it does have some limitations as it is not intended for extensive analysis. Firstly the number of rows returned in a query is limited to 10000 which may cause issues with larger sample sets.

Also, the graph options are fairly limited so may not contain the type you require.

Thankfully the Analytics portal makes it easy to export the data for analysis elsewhere. Once you enter a query there is an option to export the data to CSV for loading into Microsoft Excel, or it will re-write the query into a Power BI ready version.

Here we'll walk through the steps to create a Power BI dashboard and use the Outlook process as an example.

1. Install Power BI Desktop

Firstly, you'll need to install Power BI desktop from <u>here</u>. Power BI desktop is the place where we'll create our dashboard which can then be published for viewing via a browser.

2. Obtain the Power BI query from Azure Log Analytics

Once Power BI Desktop is installed we'll first have to obtain a query to use from Azure Log Analytics. To export a query, we simply enter the query as normal into Azure Analytics, and click the page icon with an arrow pointing to the right, then click on 'Export to Power BI (M Query)

Run Time range: Set in query	🚽 Save 🐵 Copy link 🔁 Export 🕂 New alert rule 🖍 Pin
<pre>let startTime=datetime(2018-11-14T00:01:00); let endTime=datetime(2018-11-14T11:59:59);</pre>	Export to CSV - All Columns
VMConnection where Computer contains "roshwork"	Export to CSV - Displayed Columns
<pre>where ProcessName contains "outlook" where TimeGenerated > startTime and TimeGenerated < endTime</pre>	Export to Power BI (M Query)
<pre>summarize <u>SentKbps</u> = (sum(<u>BytesSent</u>)/1024*8/60), <u>ReceivedK</u> (<u>TimeGenerated</u>, 1m), <u>Computer</u> <u>summarize <u>Percentile95SentKbps</u> = round(percentile(<u>SentKbps</u> (percentile(ReceivedKbps, 95),2) by Computer</u></pre>	

In this instance, to build an Outlook dashboard we don't have to be very specific, the following query will export <u>all</u> data related to Outlook for all users. For a specific machine uncomment the bottom line and add the machine name.

let startTime=datetime(2018-11-14T00:01:00); let endTime=datetime(2018-11-14T11:59:59); VMConnection | where ProcessName contains "outlook" | where TimeGenerated > startTime and TimeGenerated < endTime</pre>

By exporting all the data, we can choose what to display from within Power BI in different graphs in the same dashboard. Ensure the timeframe set at the top is what you wish to view in Power BI e.g. 24 Hours.

- 3. Create a new Dashboard in Power BI Desktop
- a. Open Power BI Desktop and you should see a start-up screen displayed. Select 'Get Data' from the options on the left
- b. On the next window which opens, scroll to the bottom of the options and select 'Blank Query' then click 'Connect'

Get Data		×
Search	All	
All	💿 Planview Enterprise One - CTM (Beta)	^
File	(ii) Twilio (Beta)	
Database	G tyGraph (Beta)	
Power BI	Webtrends (Beta)	
Azure	🛟 Zendesk (Beta)	
Online Services	Web	
Other	S SharePoint list	
	OData Feed	
	as Active Directory	
	Microsoft Exchange	
	Hadoop File (HDFS)	
	☆ Spark (Beta)	
	R script	
	ODBC	
	OLE DB	
	E Blank Query	\sim
		Write a que
Certified Connectors	Connect Canc	ei

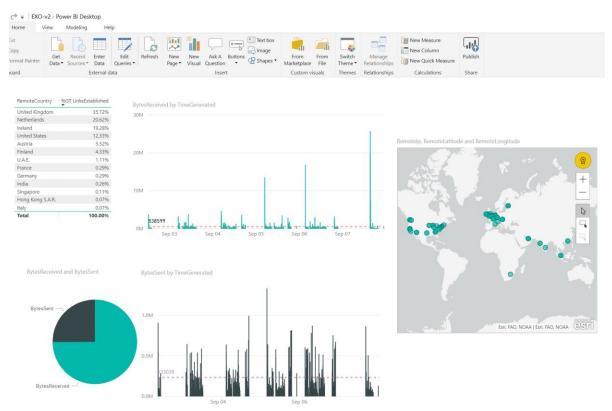
- c. In the window which opens we'll import the Power BI query we exported from Azure Log Analytics. Click 'Get Data' from the menu at the top, then 'Blank Query'.
- d. In the new window which opens, click 'Advanced Editor' delete what is in the window and paste your query from Log Analytics. If all is well, a green tick should appear showing no syntax errors. If this does not occur, it's likely the copy/paste has copied some of the instructions at the top of the export.
- e. This will load the table of data into the window and when complete, click "Close and Apply" at the top to import into our blank dashboard.

Tip: Ensure you are logged in to PowerBI with the same Organizational account that you used to login to Azure and Log Analytics. Otherwise PowerBI will throw an Authentication warning since you won't have permissions to access the workspace.

f. Now we have the data loaded into the dashboard we can begin to create graphs showing the data we wish to see.

4. Create graphs in the Power BI Dashboard

You can create your own graphs/visuals in Power BI if you are familiar within otherwise, we have provided example templates for EXO, SPO, SFBO and Teams to help you with getting started. Dashboard templates (Dashboards-v2.zip) is available for download at <u>Github</u>



4.9 Publish a Power BI Dashboard

Once you're happy with your dashboard it's time to publish it. Simply click the Publish button on the top taskbar.

```
OutlookDashboard - Power BI Desktop
```



Once saved, you'll be presented a list of possible places to publish the dashboard. Select wherever you wish, and you should receive confirmation it's been published.

Once complete, login to <u>https://powerbi.com</u> via a browser and navigate to where you published the dashboard and you should see what you created in Power BI Desktop.

When logged in, to share the dashboard, simply click share in the top right-hand corner and enter the users or groups you wish to share it with.

4.10 AutoRefresh Power BI Dashboard

To refresh the dashboard with new data every day, we need to follow these steps. In the browser, when in your Power BI dashboard:

- Click on 'My Workspace' if this is where you published it.
- Next select the workspace under 'DATASETS'.
- Click the three dots to launch the menu then select "schedule refresh' from the list.

REPORTS MachineWorkspace	Outlook Dashboard	\times
NWA_Results_Analyzer with th	LAST REFRESH FAILED:	
OutlookDashboard ····	⊗ <u>Wed May 16 2018 00:18:24 GMT+0100 (GMT</u>	
WORKBOOKS	<u>Summer Time)</u>	
You have no workbooks	Next refresh : Thu May 17 2018 00:00:00 GMT+0100 (GMT Summer Time)	
DATASETS		
MachineWorkspace		-
NWA_Results_Analyzer with th	RENAME	
NWA_Results_Analyzer with th	REM	
OutlookDashboard ···	SCHEDULE REFRESH	
	SCHEDULE REFRESH	
	REFRESH NOW	
	ANALYZE IN EXCEL	
	QUICK INSIGHTS	
	DOWNLOAD .PBIX	

• Select 'Schedule Refresh' (again) from the next menu and enable and configure the feature to your requirements.

5 Summary

The approach in this document can be used for:

- 3. Measuring network bandwidth usage for pilot users on-boarded to Office 365 or network bandwidth usage of on-premises users.
- 4. Endpoint monitoring dashboards post on-boarding users to Office 365

You can apply this concept for measuring any SaaS/PaaS traffic, not just Office 365. Future improvements to this project can be tracked at <u>Github</u>