16TH & 17TH OCTOBER 2019

PolarConf 2019

The Most Northern Azure Conference For IT Professionals. Brought To You By Finland Azure User Group.

BUILD YOUR OWN AZURE MONITOR SOLUTION

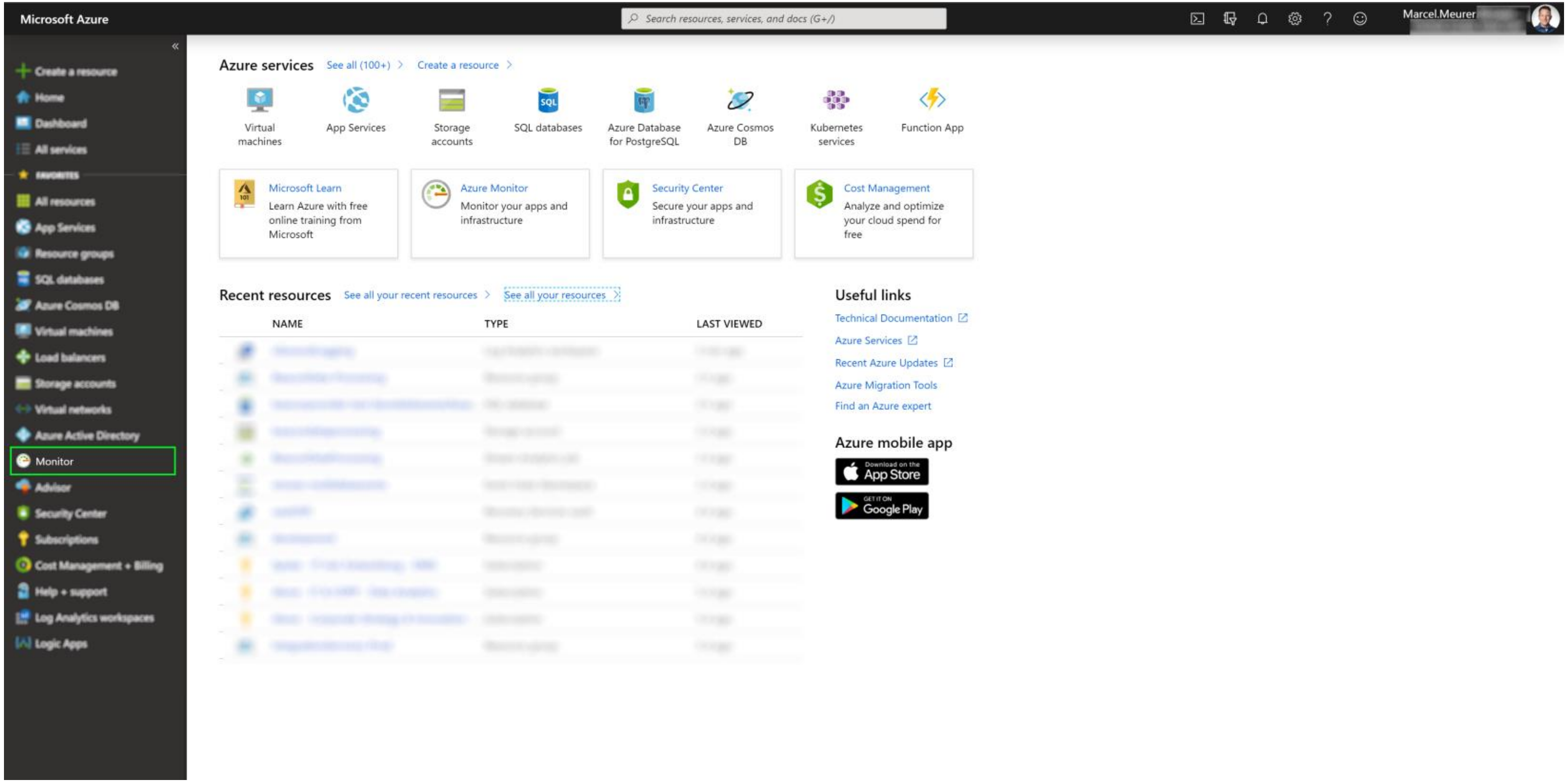
About me

Responsible for Consulting at sepago GmbH

Microsoft Azure
Machine Learning
Azure Monitor / Log Analytics

Mail:	marcel.meurer@sepago.de
Twitter:	https://twitter.com/MarcelMeurer
GitHub:	https://github.com/MarcelMeurer
Blog:	https://blog.itprocloud.de/

What is Azure Monitor



Microsoft Learn
Learn Azure with free online training from Microsoft

Azure Monitor
Monitor your apps and infrastructure

Security Center
Secure your apps and infrastructure

Cost Management
Analyze and optimize your cloud spend for free

What is Azure Monitor

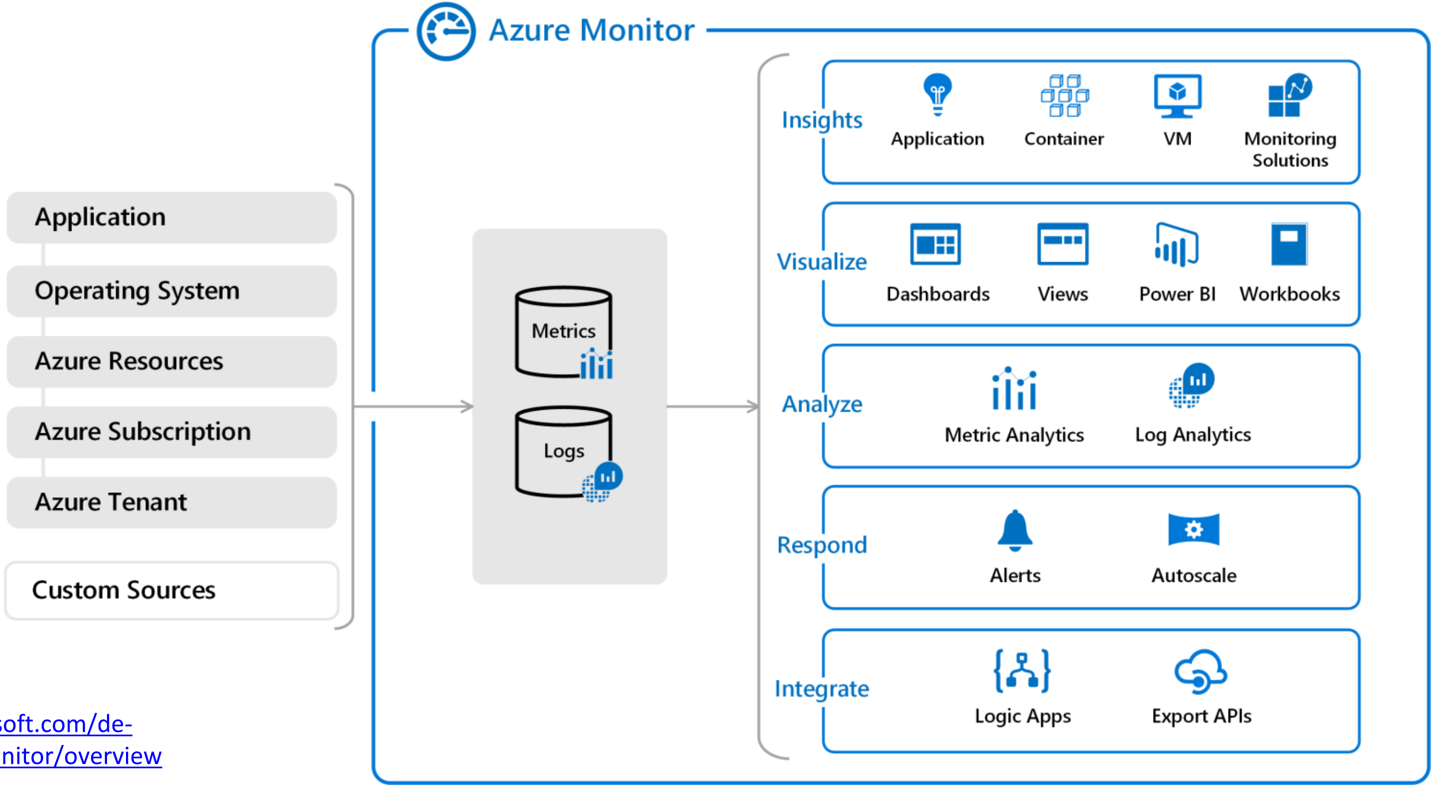
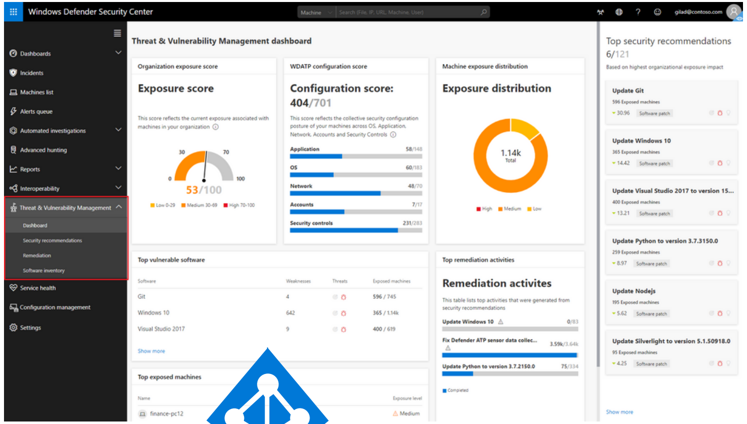


Image source:
https://docs.microsoft.com/de-de/azure/azure-monitor/overview

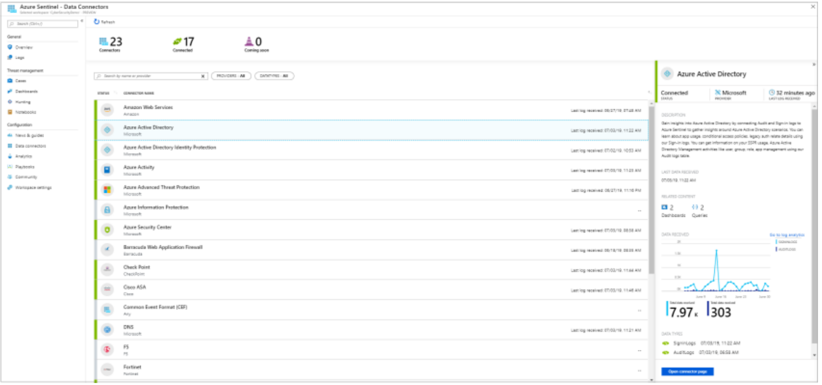
Where is it used in Azure



Azure
Active Directory

Application
Insights

Microsoft
Azure



Logic App

Image source:
https://docs.microsoft.com/en-us/azure/sentinel/overview, https://techcommunity.microsoft.com/t5/Microsoft-Defender-ATP/MDATP-Threat-amp-Vulnerability-Management-now-publicly-available/ba-p/460977, https://docs.microsoft.com/de-de/azure/security-center/

The secret of Azure Monitor

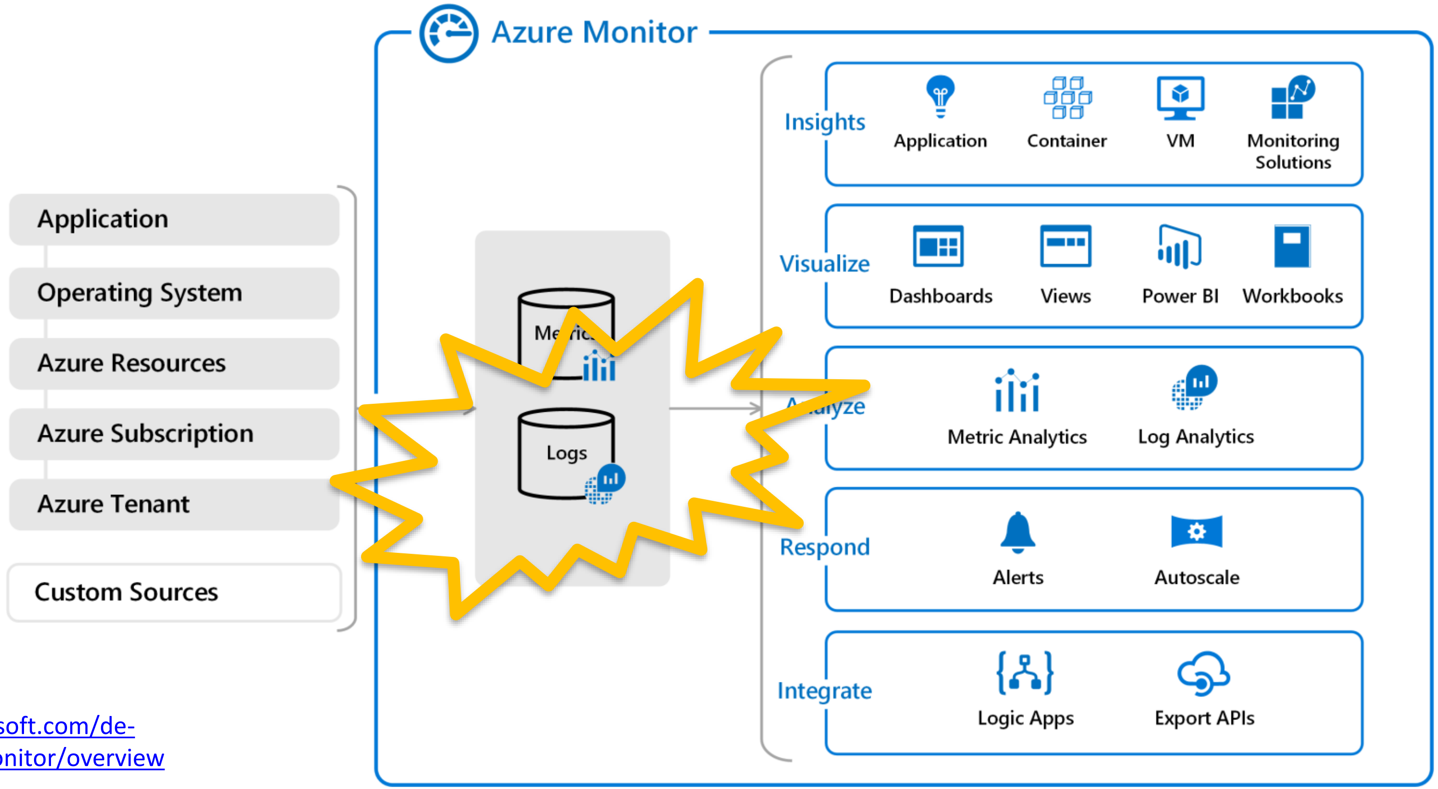


Image source:
https://docs.microsoft.com/de-de/azure/azure-monitor/overview

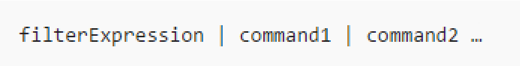
Azure Monitor=Log Analytics

- THE Big Data Container

- No-SQL data storage
- Automatic indexing
- High performance – auto-scale
- Solutions for visualization available
- Build own dashboards, tiles and charts
- Expandable through own agents*

The query language: KUSTO

- No SQL -> focus on small command set and performance
- Pipelined:

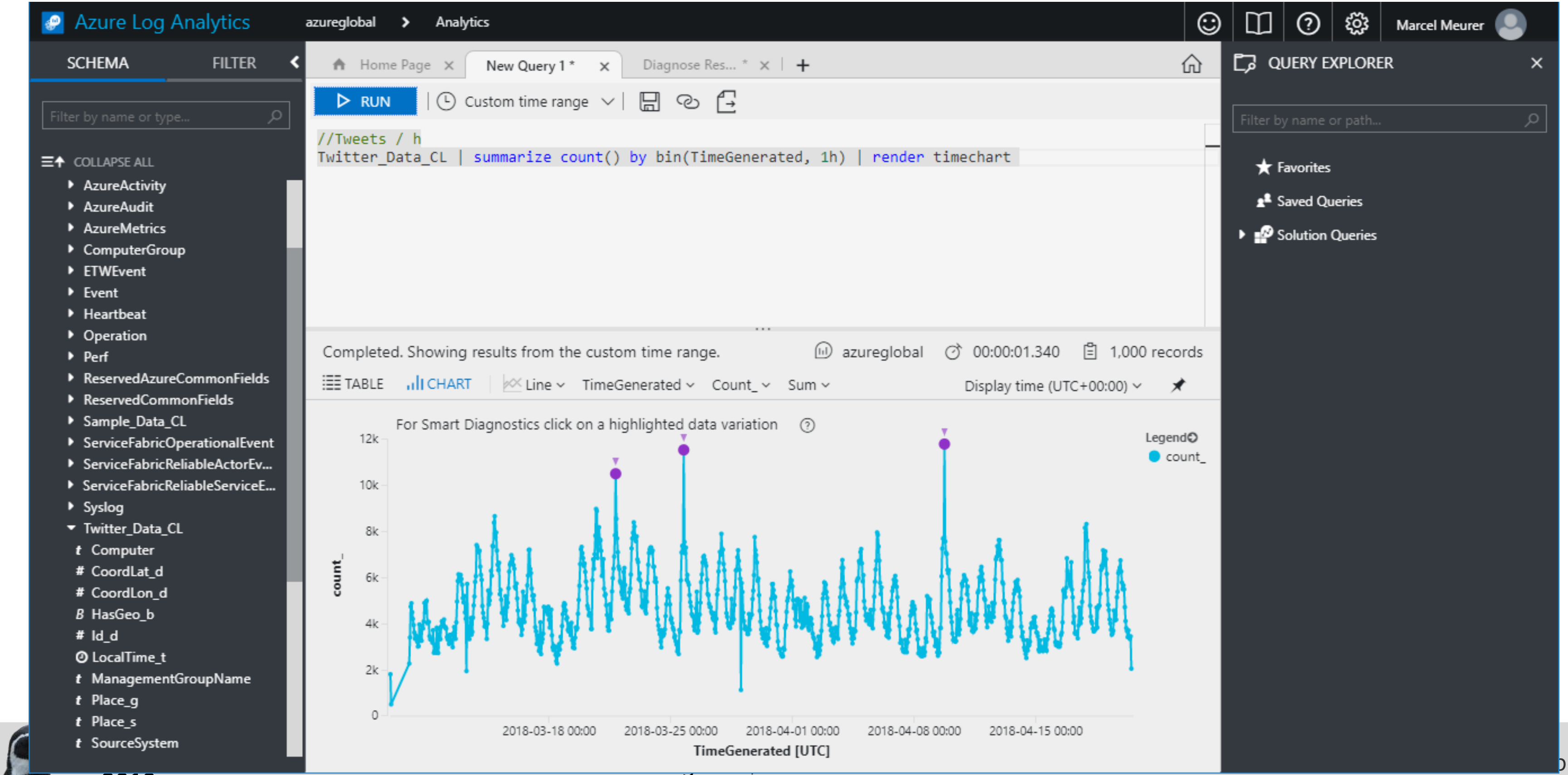


```
filterExpression | command1 | command2 …
```

- CaSenSitiVe
- Command groups:
  - filters
  - queries
  - selectors
  - logical operations
  - Sorting
  - measurements and aggregate functions

https://docs.loganalytics.io/index

Data and Queries

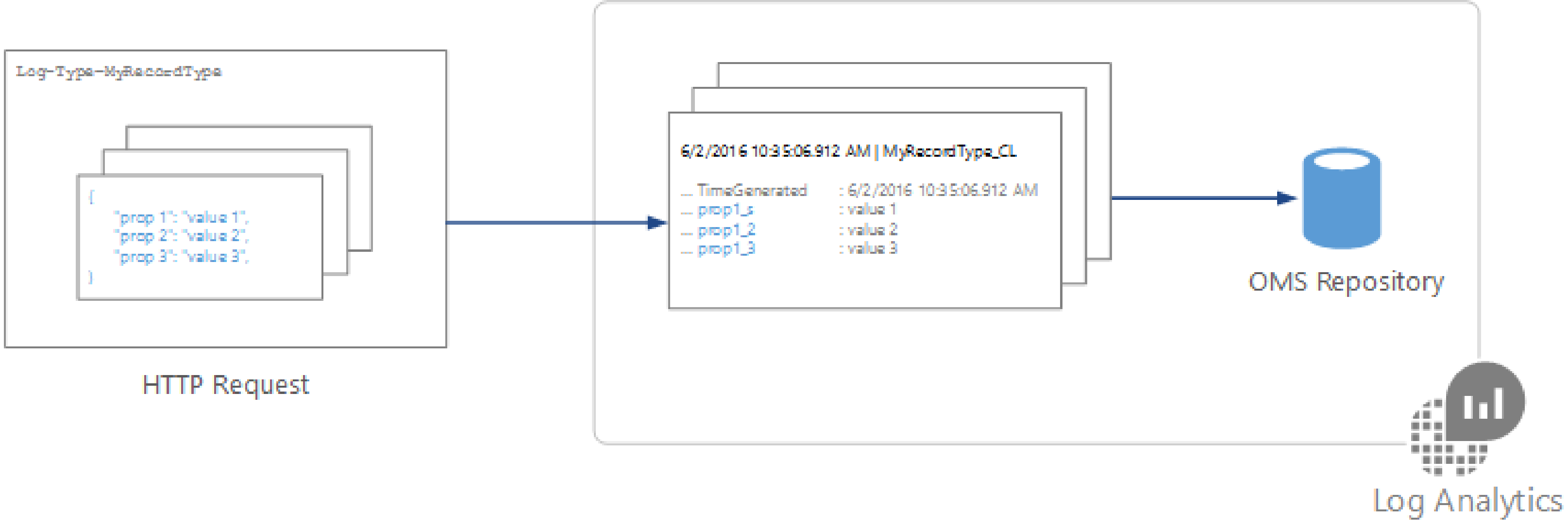


```
//Tweets / h
Twitter_Data_CL | summarize count() by bin(TimeGenerated, 1h) | render timechart
```

Completed. Showing results from the custom time range. azureglobal 00:00:01.340 1,000 records

Data upload

- Data Collector API
  https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-collector-api

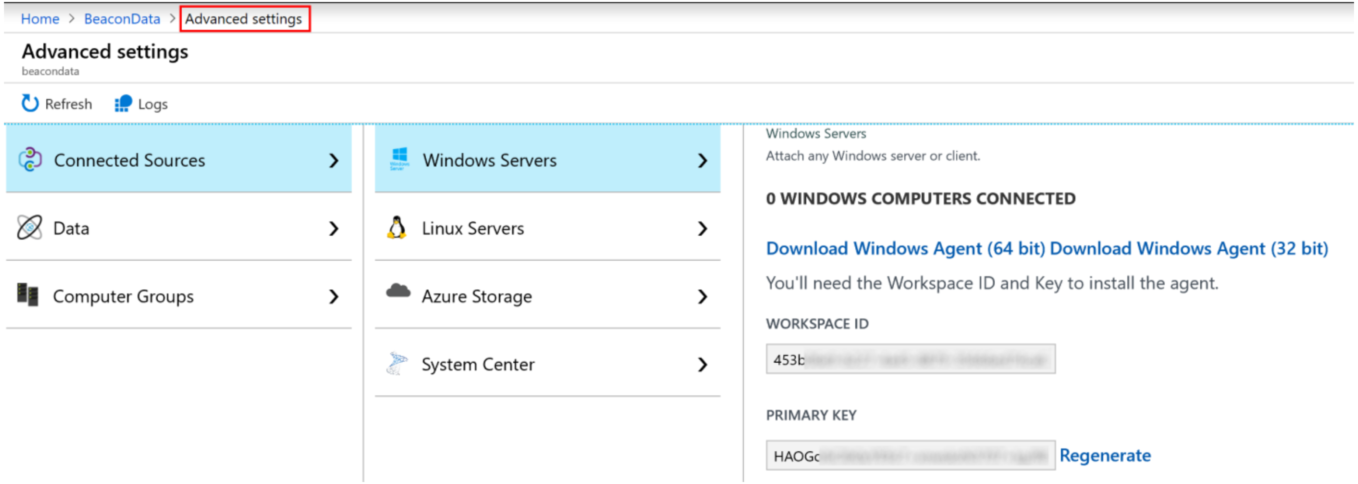


```
{
    "prop 1": "value 1",
    "prop 2": "value 2",
    "prop 3": "value 3",
}
```

```
6/2/2016 10:35:06.912 AM | MyRecordType_CL

_ TimeGenerated      : 6/2/2016 10:35:06.912 AM
_ prop1_s            : value 1
_ prop1_2            : value 2
_ prop1_3            : value 3
```

Data upload

- Data Collector API
  https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-collector-api
-



Advanced settings
beacondata

Windows Servers
Attach any Windows server or client.

0 WINDOWS COMPUTERS CONNECTED

You'll need the Workspace ID and Key to install the agent.

WORKSPACE ID
453b[redacted]

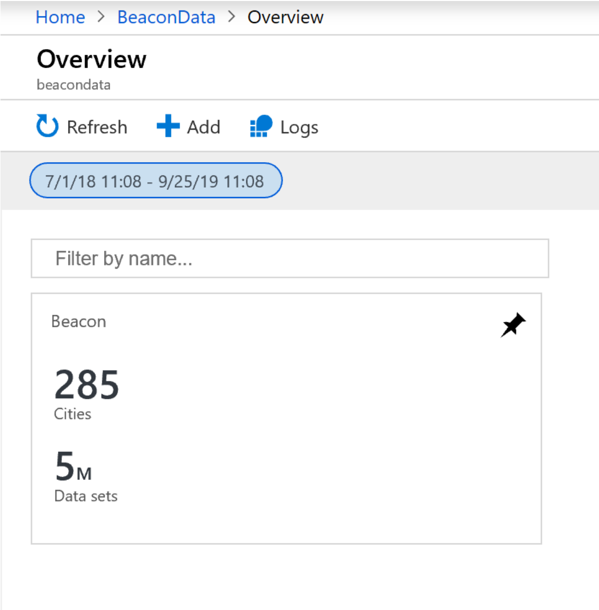
PRIMARY KEY
HAOGc[redacted]

- What do you need?
  - Workspace Id
  - Key

Creating views

n vi Monitor /

e P s



Overview

beacondata

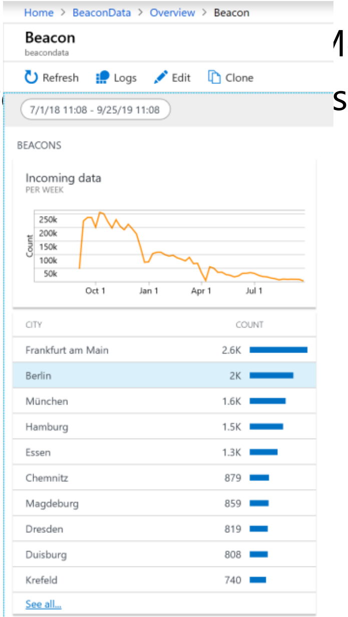
7/1/18 11:08 - 9/25/19 11:08

Filter by name...

Beacon

285
Cities

5M
Data sets



Beacon

beacondata

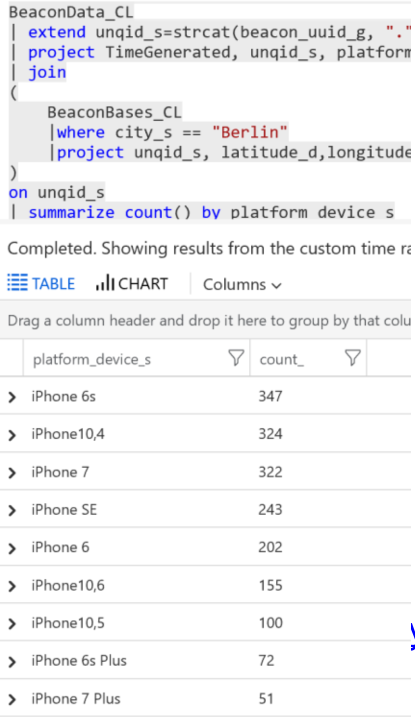
7/1/18 11:08 - 9/25/19 11:08

BEACONS

Incoming data
PER WEEK

CITY	COUNT
Frankfurt am Main	2.6K
Berlin	2K
München	1.6K
Hamburg	1.5K
Essen	1.3K
Chemnitz	879
Magdeburg	859
Dresden	819
Duisburg	808
Krefeld	740

See all...



```
BeaconData_CL
| extend unqid_s=strcat(beacon_uuid_g, "."
| project TimeGenerated, unqid_s, platform
| join
(
    BeaconBases_CL
    |where city_s == "Berlin"
    |project unqid_s, latitude_d,longitude
)
on unqid_s
| summarize count() by platform device s
```

Completed. Showing results from the custom time ra

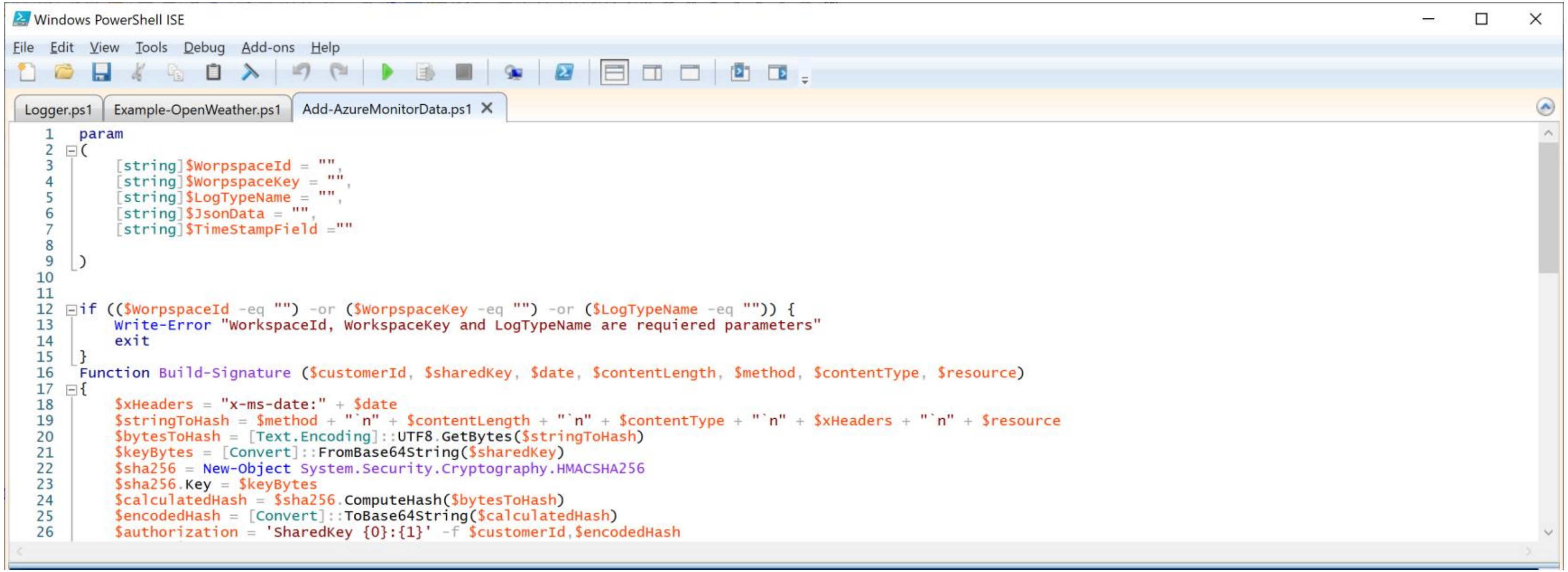
Drag a column header and drop it here to group by that colu

platform_device_s	count_
iPhone 6s	347
iPhone10,4	324
iPhone 7	322
iPhone SE	243
iPhone 6	202
iPhone10,6	155
iPhone10,5	100
iPhone 6s Plus	72
iPhone 7 Plus	51

- https://blogs.technet.microsoft.com/msoms/2016/06/30/oms- your-data-your-way/

Demo-Simple PowerShell

- Upload data with PowerShell

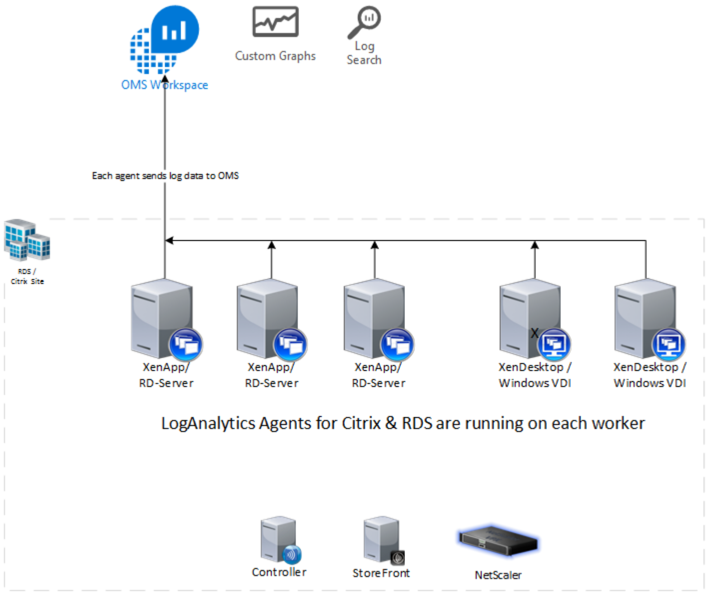


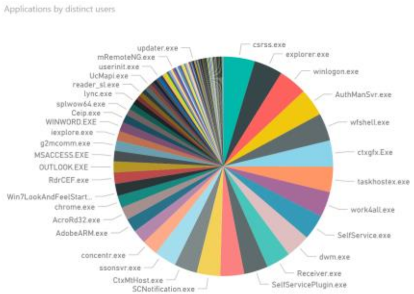
```powershell
param
(
    [string]$WorpspaceId = "",
    [string]$WorpspaceKey = "",
    [string]$LogTypeName = "",
    [string]$JsonData = "",
    [string]$TimeStampField =""

)


if (($WorpspaceId -eq "") -or ($WorpspaceKey -eq "") -or ($LogTypeName -eq "")) {
    Write-Error "WorkspaceId, WorkspaceKey and LogTypeName are requiered parameters"
    exit
}
Function Build-Signature ($customerId, $sharedKey, $date, $contentLength, $method, $contentType, $resource)
{
    $xHeaders = "x-ms-date:" + $date
    $stringToHash = $method + "`n" + $contentLength + "`n" + $contentType + "`n" + $xHeaders + "`n" + $resource
    $bytesToHash = [Text.Encoding]::UTF8.GetBytes($stringToHash)
    $keyBytes = [Convert]::FromBase64String($sharedKey)
    $sha256 = New-Object System.Security.Cryptography.HMACSHA256
    $sha256.Key = $keyBytes
    $calculatedHash = $sha256.ComputeHash($bytesToHash)
    $encodedHash = [Convert]::ToBase64String($calculatedHash)
    $authorization = 'SharedKey {0}:{1}' -f $customerId,$encodedHash
```

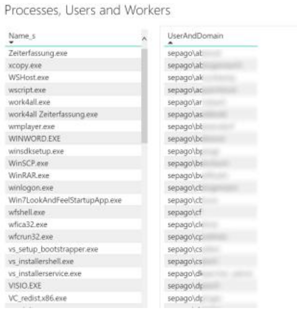
Script: https://bit.ly/317SlDj

Demo – Citrix & WVD

- My monitoring agent for Citrix and Windows Virtual Desktop







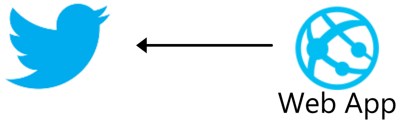
Processes, Users and Workers

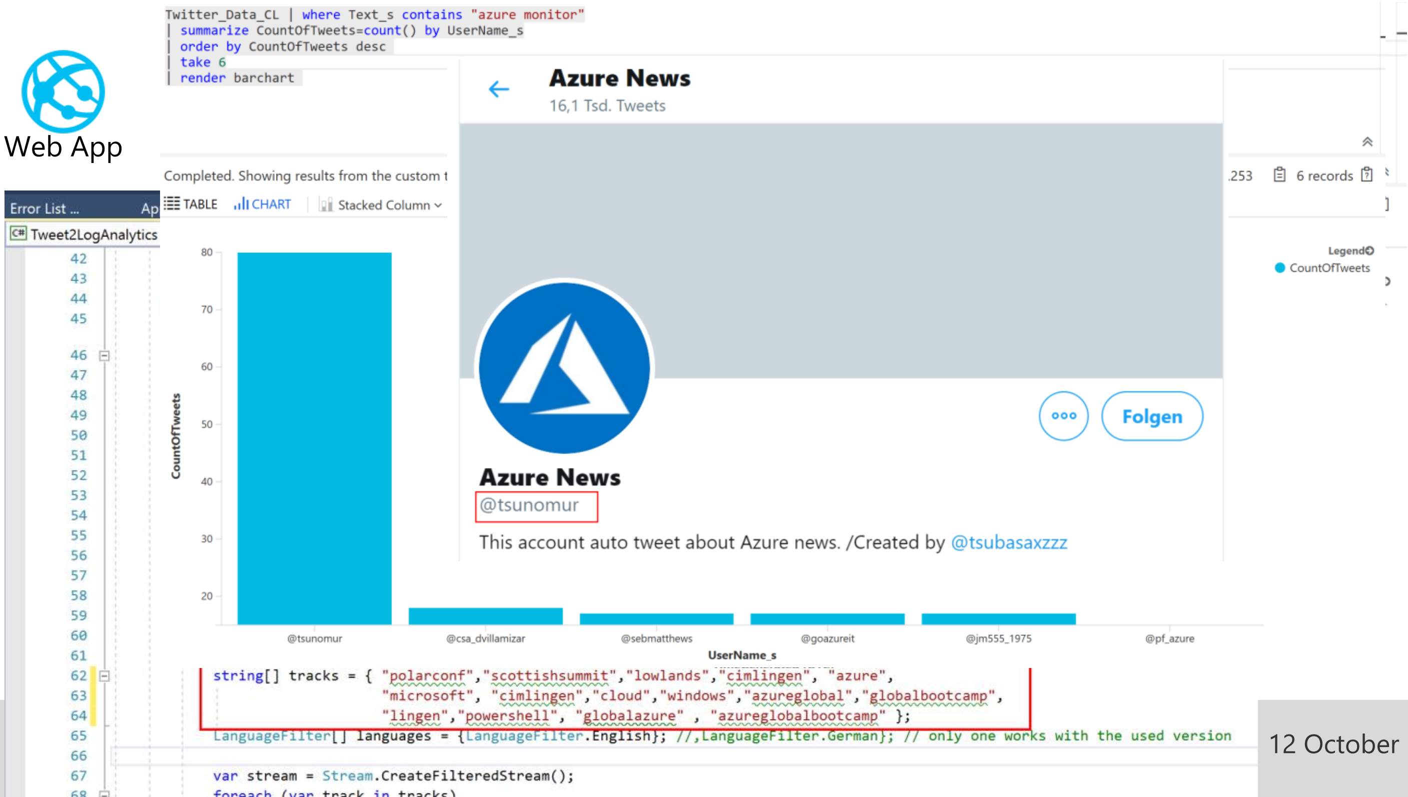
Name_s	UserAndDomain
Zeiterfassung.exe	sepago\ab[redacted]
xcopy.exe	sepago\ab[redacted]
WSHost.exe	sepago\ak[redacted]
wscript.exe	sepago\ac[redacted]
work4all.exe	sepago\ar[redacted]
work4all Zeiterfassung.exe	sepago\as[redacted]
wmplayer.exe	sepago\bb[redacted]
WINWORD.EXE	sepago\bc[redacted]
winsdksetup.exe	sepago\bp[redacted]
WinSCP.exe	sepago\bs[redacted]
WinRAR.exe	sepago\bv[redacted]
winlogon.exe	sepago\cb[redacted]
Win7LookAndFeelStartupApp.exe	sepago\cb[redacted]
wfshell.exe	sepago\cf[redacted]
wfica32.exe	sepago\ck[redacted]
wfcrun32.exe	sepago\cp[redacted]
vs_setup_bootstrapper.exe	sepago\cs[redacted]
vs_installershell.exe	sepago\cs[redacted]
vs_installerservice.exe	sepago\dk[redacted]
VISIO.EXE	sepago\dp[redacted]
VC_redist.x86.exe	sepago\dp[redacted]

Azure Marketplace:
https://azuremarketplace.microsoft.com/en-us/marketplace/apps/sepagogmbh.loganalyticsagent-rds?tab=Overview

Demo – Twitter

- Collecting data from Twitter





```
Twitter_Data_CL | where Text_s contains "azure monitor"
| summarize CountOfTweets=count() by UserName_s
| order by CountOfTweets desc
| take 6
| render barchart
```

Completed. Showing results from the custom t

.253 6 records

Azure News

16,1 Tsd. Tweets

Azure News

@tsunomur

This account auto tweet about Azure news. /Created by @tsubasaxzzz

```
string[] tracks = { "polarconf","scottishsummit","lowlands","cimlingen", "azure",
                    "microsoft", "cimlingen","cloud","windows","azureglobal","globalbootcamp",
                    "lingen","powershell", "globalazure" , "azureglobalbootcamp" };
LanguageFilter[] languages = {LanguageFilter.English}; //,LanguageFilter.German}; // only one works with the used version

var stream = Stream.CreateFilteredStream();
foreach (var track in tracks)
```

Workshop

Fulfill the Prerequisites

- Create your private Azure Monitor Log Analytics Workspace
  - https://portal.azure.com/#create/Microsoft.LogAnalyticsOMS
  - Extend the retention: Usage and estimated costs / Data Retention
  - Grab the workspace id and workspace key: Advanced settings

- Create a local working folder
  - Download PowerShell utilities and sample scripts from https://github.com/MarcelMeurer/Workshop-AzureMonitor
    - Use: git clone https://github.com/MarcelMeurer/Workshop-AzureMonitor.git
    - Or download zip-file
    - Allow PowerShell scripts for today:
      Set-ExecutionPolicy -ExecutionPolicy Unrestricted

- Documentation: https://bit.ly/317SlDj

Workshop

Mission: Store information about the running processes from your computer

- Collect the process information from your computer each 30 seconds and send these data to your Log Analytics workspace. Use PowerShell to automate this mission.
  - Select an app and use this app to “overload” your CPU.
  - If data are visible in Log Analytics, build a custom dashboard by using “Log” to query the data.
  - Find out:
    - Count of distinct processes
    - Average CPU load over time (all processes). Render a time chart
    - Render a time chart for the app you used to overload the CPU

- Documentation: https://bit.ly/317SlDj

Workshop

Mission: Store temperature data for multiple cities

- Collect data from OpenWeatherMap
  - https://openweathermap.org/
  - Create an account and api key
  - Test your key (it can take some minutes): https://api.openweathermap.org/data/2.5/weather?q=Bonn&APIKEY=xxxxxxx

- Build an PowerShell
  - Build a script that retrieves the data regularly (every 30 seconds) for three cities
  - Push the Data to your Log Analytics workspace

- Build a Dashboard showing some data
  - Cities and Temperatur
  - Cities and Humidity

- Documentation: https://bit.ly/317SlDj

Workshop

Mission: Build your own log-writer function

- Build a log-writer function for your own PowerShell scripts using Log Analytics. There are some request to your solution:

- Have the following columns:
  - TimeStamp (as TimeGeneratedField)
  - Serverity (Debug Information Warning Error)
  - Message (Text)
  - ScriptName (Name of the script using your function)

- Documentation: https://bit.ly/317SlDj

Workshop

Let’s go!

- https://blog.itprocloud.de/Workshop-Azure-Monitor-Examples

or

- https://bit.ly/317SIDj

Questions

THANK YOU