



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/>



sepago®



What is Azure Monitor

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar with the placeholder text "Search resources, services, and docs (G+)", and a user profile for Marcel Meurer. The main content area is divided into several sections:

- Azure services:** A row of service tiles including Virtual machines, App Services, Storage accounts, SQL databases, Azure Database for PostgreSQL, Azure Cosmos DB, Kubernetes services, and Function App.
- Recommended services:** Four tiles for Microsoft Learn (Learn Azure with free online training from Microsoft), Azure Monitor (Monitor your apps and infrastructure), Security Center (Secure your apps and infrastructure), and Cost Management (Analyze and optimize your cloud spend for free).
- Recent resources:** A table with columns for NAME, TYPE, and LAST VIEWED, displaying a list of recent resource entries.
- Useful links:** A list of links including Technical Documentation, Azure Services, Recent Azure Updates, Azure Migration Tools, and Find an Azure expert.
- Azure mobile app:** Badges for downloading the app from the App Store and Google Play.

In the left-hand navigation menu, the "Monitor" option is highlighted with a green box.



What is Azure Monitor

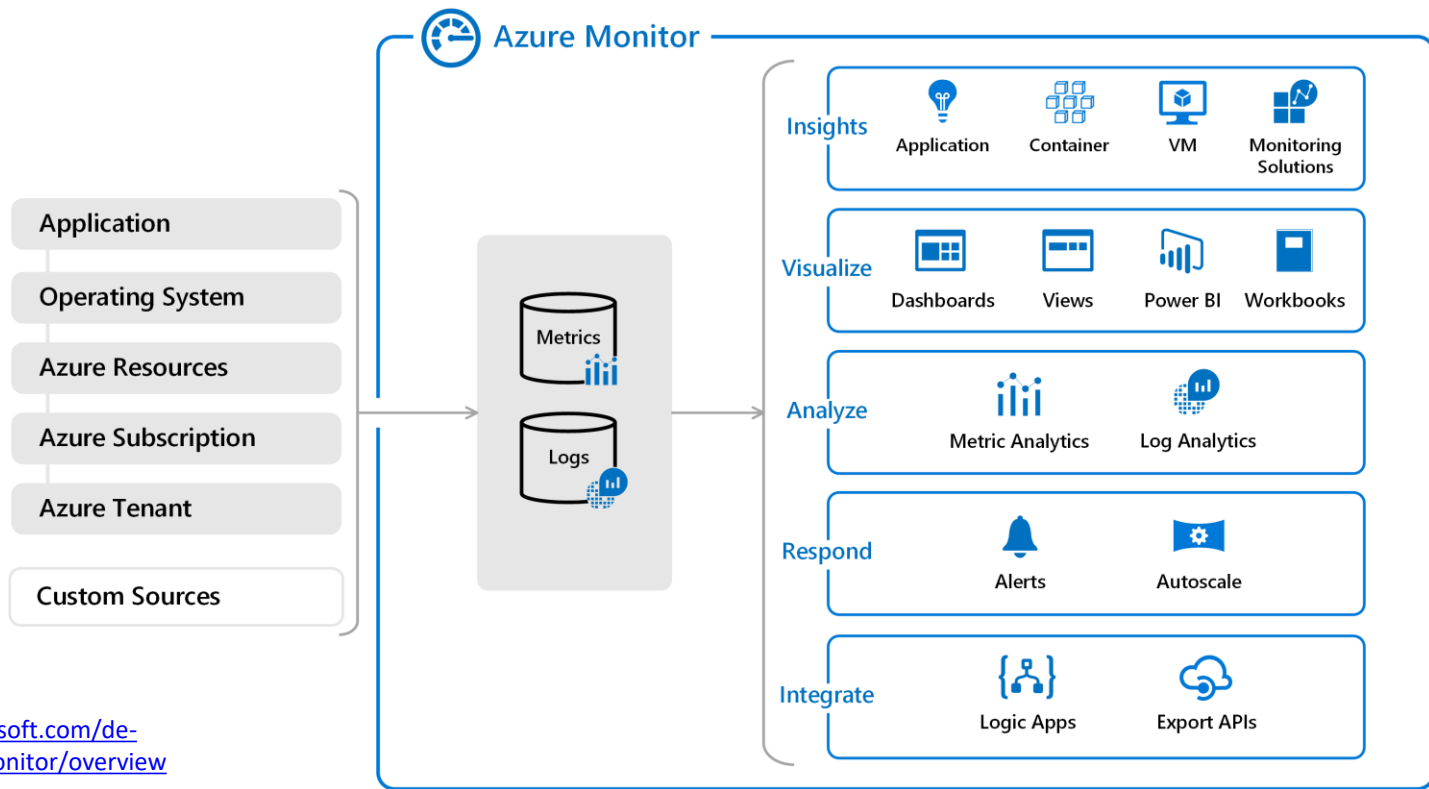
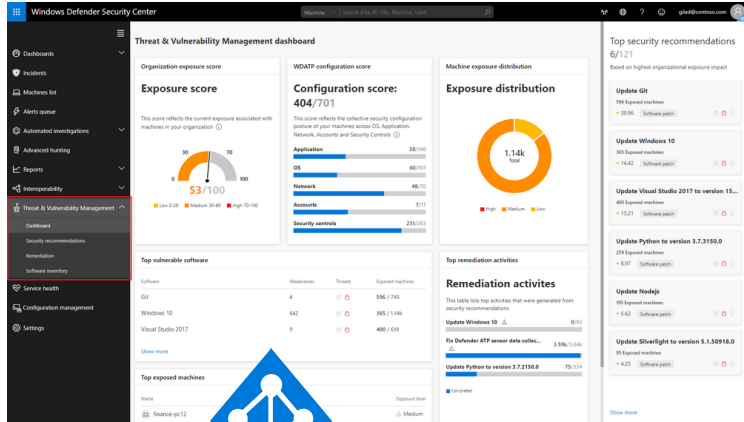


Image source:

<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

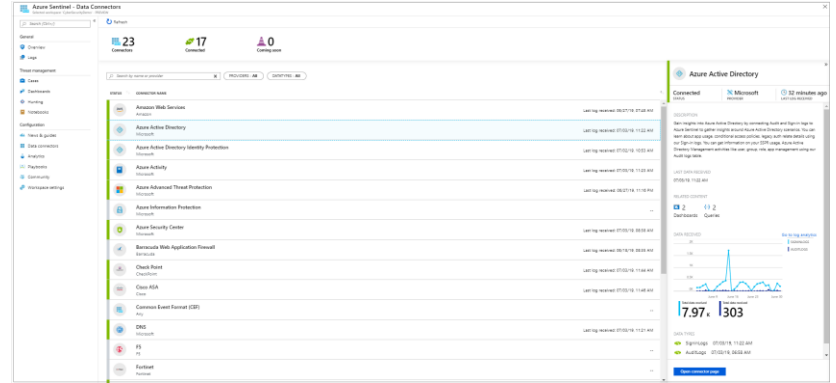


Image source:

<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/>



PolarConf
2019

Together, communities achieve more.

12 October 2019 | 5

The secret of Azure Monitor

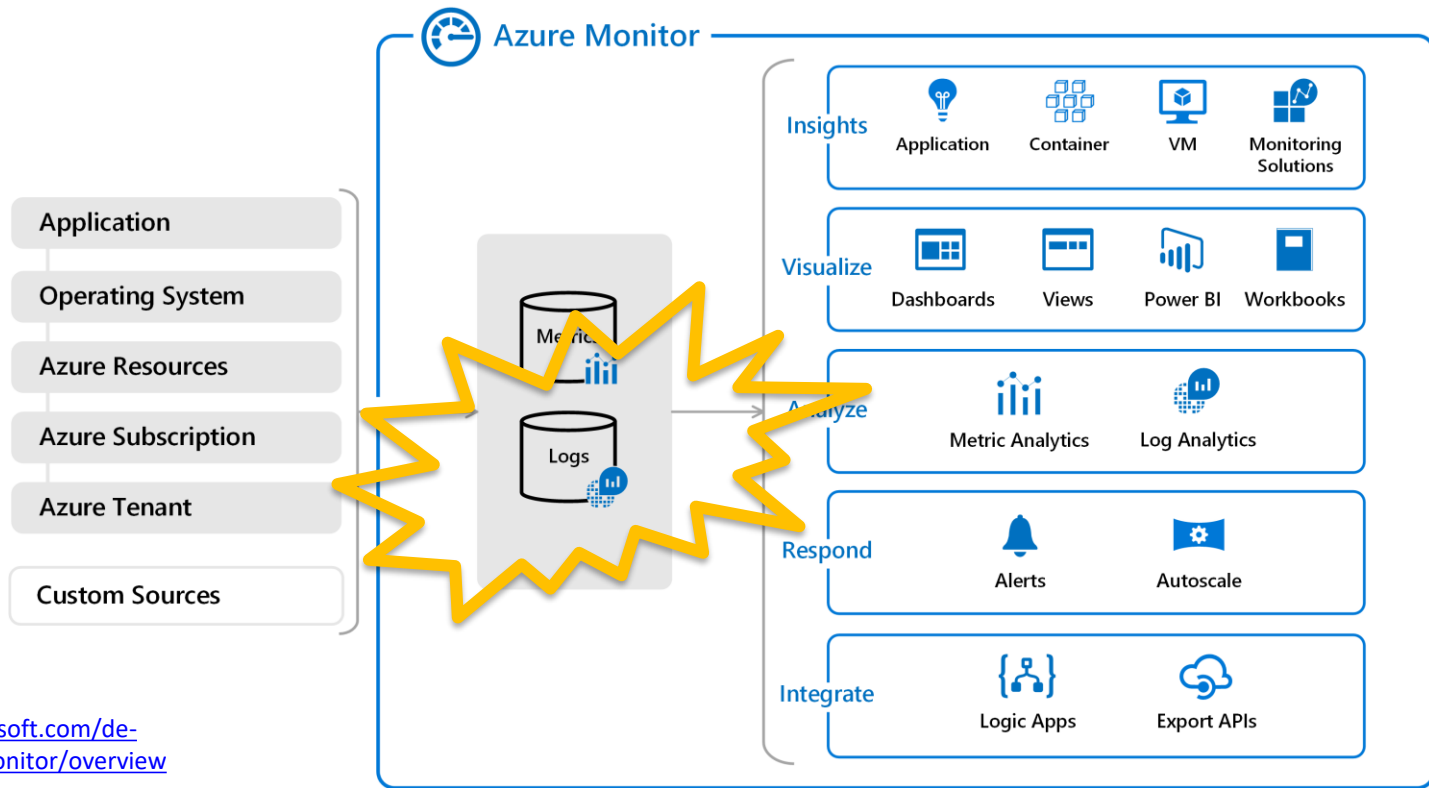


Image source:
<https://docs.microsoft.com/de-de/azure/azure-monitor/overview>



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 ...`
- CaSenSiVe
- Command groups:
 - filters
 - queries
 - selectors
 - logical operations
 - Sorting
 - measurements and aggregate functions

<https://docs.loganalytics.io/index>



Data and Queries

The screenshot displays the Azure Log Analytics interface. The top navigation bar includes the 'Azure Log Analytics' logo, the workspace name 'azureglobal', and the 'Analytics' section. The main area is divided into a 'SCHEMA' pane on the left, a central query editor, and a 'QUERY EXPLORER' pane on the right.

SCHEMA Pane: Lists various data sources and fields. Under 'Twitter_Data_CL', the following fields are visible: Computer, CoordLat_d, CoordLon_d, HasGeo_b, Id_d, LocalTime_t, ManagementGroupName, Place_g, Place_s, and SourceSystem.

Query Editor: Contains the following query: `//Tweets / h Twitter_Data_CL | summarize count() by bin(TimeGenerated, 1h) | render timechart`. The 'RUN' button is highlighted.

Query Explorer: Includes a search bar and a list of 'Solution Queries'.

Results: A timechart is displayed, showing the 'count_' metric over time. The x-axis is labeled 'TimeGenerated [UTC]' and ranges from 2018-03-16 00:00 to 2018-04-15 00:00. The y-axis is labeled 'count_' and ranges from 0 to 12k. The chart shows a fluctuating line with several peaks, the highest reaching approximately 11k. A legend indicates the data series is 'count_'.

Below the chart, the status bar indicates: 'Completed. Showing results from the custom time range. azureglobal 00:00:01.340 1,000 records'. The chart controls show 'TABLE' and 'CHART' options, with 'Line' selected for the chart type. The y-axis is set to 'Count' and the x-axis to 'TimeGenerated'.

Data upload

- Data Collector API

<https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-collector-api>



Data upload

- Data Collector API

<https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-collector-api>

Home > BeaconData > **Advanced settings**

Advanced settings
beacondata

Refresh Logs

- Connected Sources >
 - Windows Servers** >
 - Windows Servers
Attach any Windows server or client.
 - 0 WINDOWS COMPUTERS CONNECTED**
 - [Download Windows Agent \(64 bit\)](#) [Download Windows Agent \(32 bit\)](#)
 - You'll need the Workspace ID and Key to install the agent.
 - WORKSPACE ID
453e: [redacted]
 - PRIMARY KEY
HAOGc: [redacted] [Regenerate](#)
 - Data >
 - Computer Groups >
- Linux Servers >
- Azure Storage >
- System Center >

- What do you need?

- Workspace Id
- Key



Creating views

Home > BeaconData > Overview

Overview

beacondata

Refresh Add Logs

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

Filter by name...

Beacon

285
Cities

5M
Data sets

Home > BeaconData > Overview > Beacon

Beacon

Refresh Logs Edit Clone

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...](#)

Monitor /

```
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_d
)
on unqid_s
| summarize count() by platform device_s
```

Completed. Showing results from the custom time range

TABLE CHART Columns v

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

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

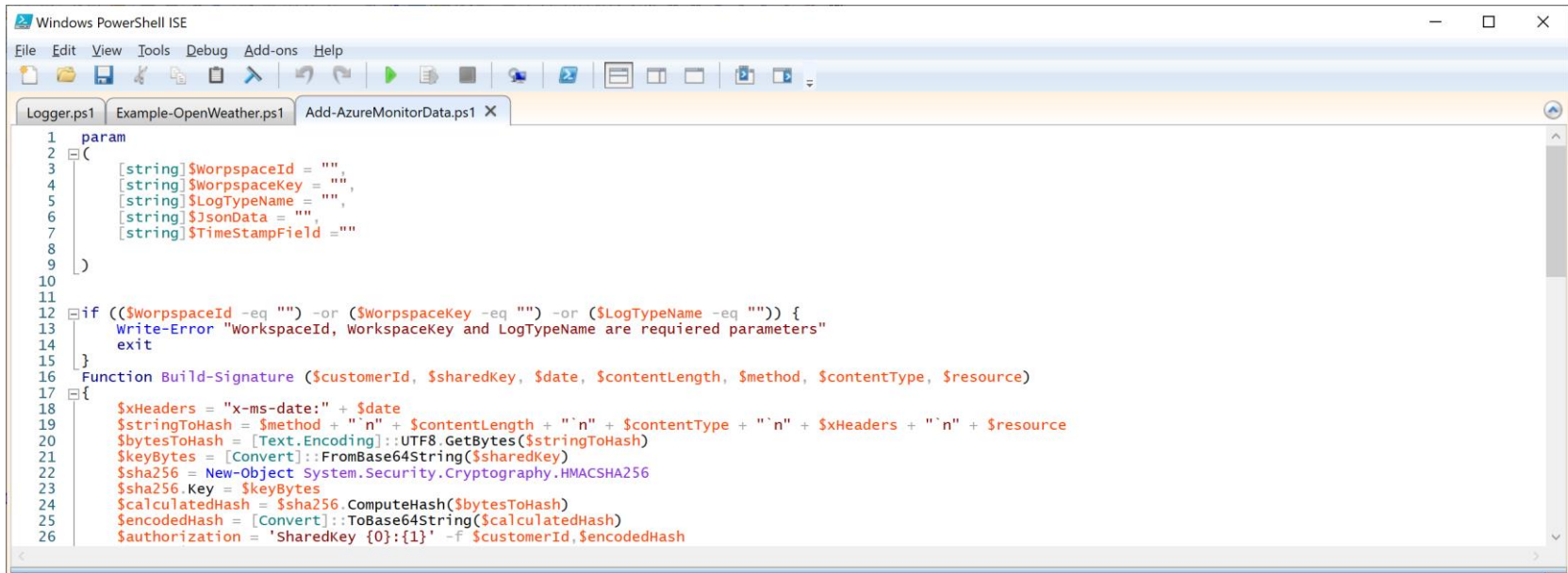
■ <https://blogs.technet.microsoft.com/11/11/2018/11/20/11/30/oms/>

[your-data-your-way/](#)



Demo-Simple PowerShell

- Upload data with PowerShell



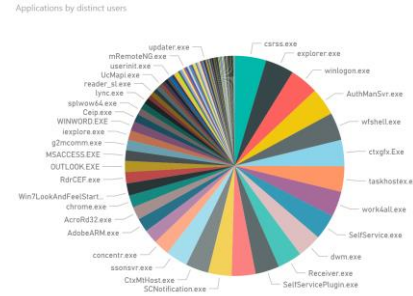
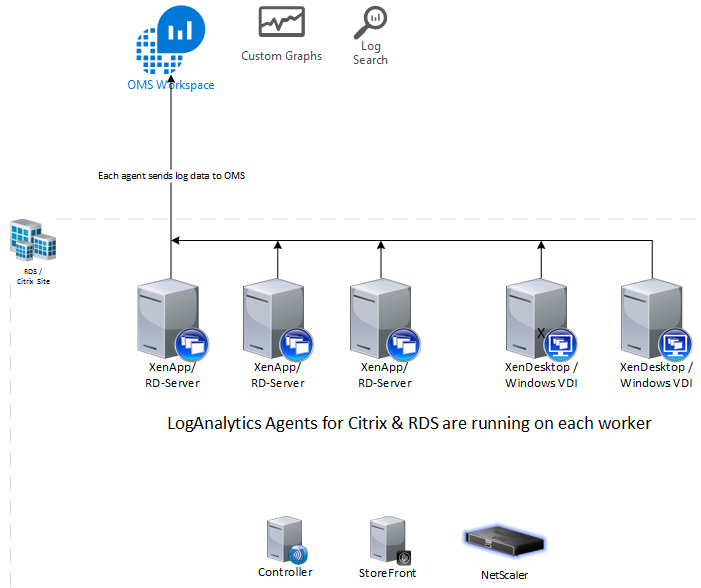
```
1 param
2 (
3     [string]$WorkspaceId = "",
4     [string]$WorkspaceKey = "",
5     [string]$LogTypeName = "",
6     [string]$JsonData = "",
7     [string]$TimeStampField = ""
8 )
9
10
11
12 if (($WorkspaceId -eq "") -or ($WorkspaceKey -eq "") -or ($LogTypeName -eq "")) {
13     Write-Error "WorkspaceId, WorkspaceKey and LogTypeName are required parameters"
14     exit
15 }
16 Function Build-Signature ($customerId, $sharedKey, $date, $contentLength, $method, $contentType, $resource)
17 {
18     $xHeaders = "x-ms-date:" + $date
19     $stringToHash = $method + "`n" + $contentLength + "`n" + $contentType + "`n" + $xHeaders + "`n" + $resource
20     $bytesToHash = [Text.Encoding]::UTF8.GetBytes($stringToHash)
21     $keyBytes = [Convert]::FromBase64String($sharedKey)
22     $sha256 = New-Object System.Security.Cryptography.HMACSHA256
23     $sha256.Key = $keyBytes
24     $calculatedHash = $sha256.ComputeHash($bytesToHash)
25     $encodedHash = [Convert]::ToBase64String($calculatedHash)
26     $authorization = 'SharedKey {0}:{1}' -f $customerId,$encodedHash
```

Script: <https://bit.ly/317SIDj>



Demo – Citrix & WVD

- My monitoring agent for Citrix and Windows Virtual Desktop



Processes, Users and Workers

Name_s	UserAndDomain
Zeterfassung.exe	sepago\at
scopy.exe	sepago\ak
WSHost.exe	sepago\ac
wscrip.exe	sepago\ar
workkall.exe	sepago\as
workkall.Zeterfassung.exe	sepago\at
wmplayer.exe	sepago\bt
WINWORD.EXE	sepago\bc
winwordsetup.exe	sepago\by
WinOS.exe	sepago\be
WinRAR.exe	sepago\bu
winlogon.exe	sepago\ct
Win7LookAndFreeStart	sepago\ct
chrome.exe	sepago\ct
AcroR32.exe	sepago\ck
AdobeARM.exe	sepago\ck
concentr.exe	sepago\cp
issconver.exe	sepago\cs
CusMkInst.exe	sepago\cs
SCHNotification.exe	sepago\cs
vs_installer.exe	sepago\cs
vs_installer.exe	sepago\cs
vs_installer.exe	sepago\cs
VSIO.EXE	sepago\ds
VC_redist.x86.exe	sepago\ds

Azure Marketplace:

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



Demo – Twitter

- Collecting data from Twitter



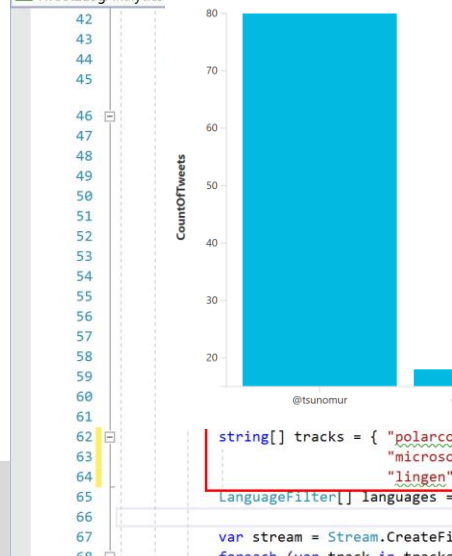
Web App

```
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 i

Error List ... Ap TABLE CHART Stacked Column v

Tweet2LogAnalytics



```
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/317SIDj>



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/317SIDj>



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/317SIDj>



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/317SIDj>



Workshop

Let's go!

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

or

- <https://bit.ly/317SIDj>



Questions



