

KQL Café | November 2024

# Your | hosts

## Alex Verboon



## Gianni Castaldi



<https://twitter.com/alexverboon>

<https://www.linkedin.com/in/verboonalex/>

<https://github.com/alexverboon>

<https://www.verboon.info/>

[https://twitter.com/castello\\_johnny](https://twitter.com/castello_johnny)

<https://www.linkedin.com/in/giannicastaldi/>

<https://github.com/KustoKing>

<https://www.kustoking.com/>



**KQL** | Cafe

Show | Agenda

Welcome

KustoCon Recap

What is new/updates for KQL

Our guest: Mehmet Ergene

What did you do with KQL this month?

# KustoCon

Learn | Share | Practice

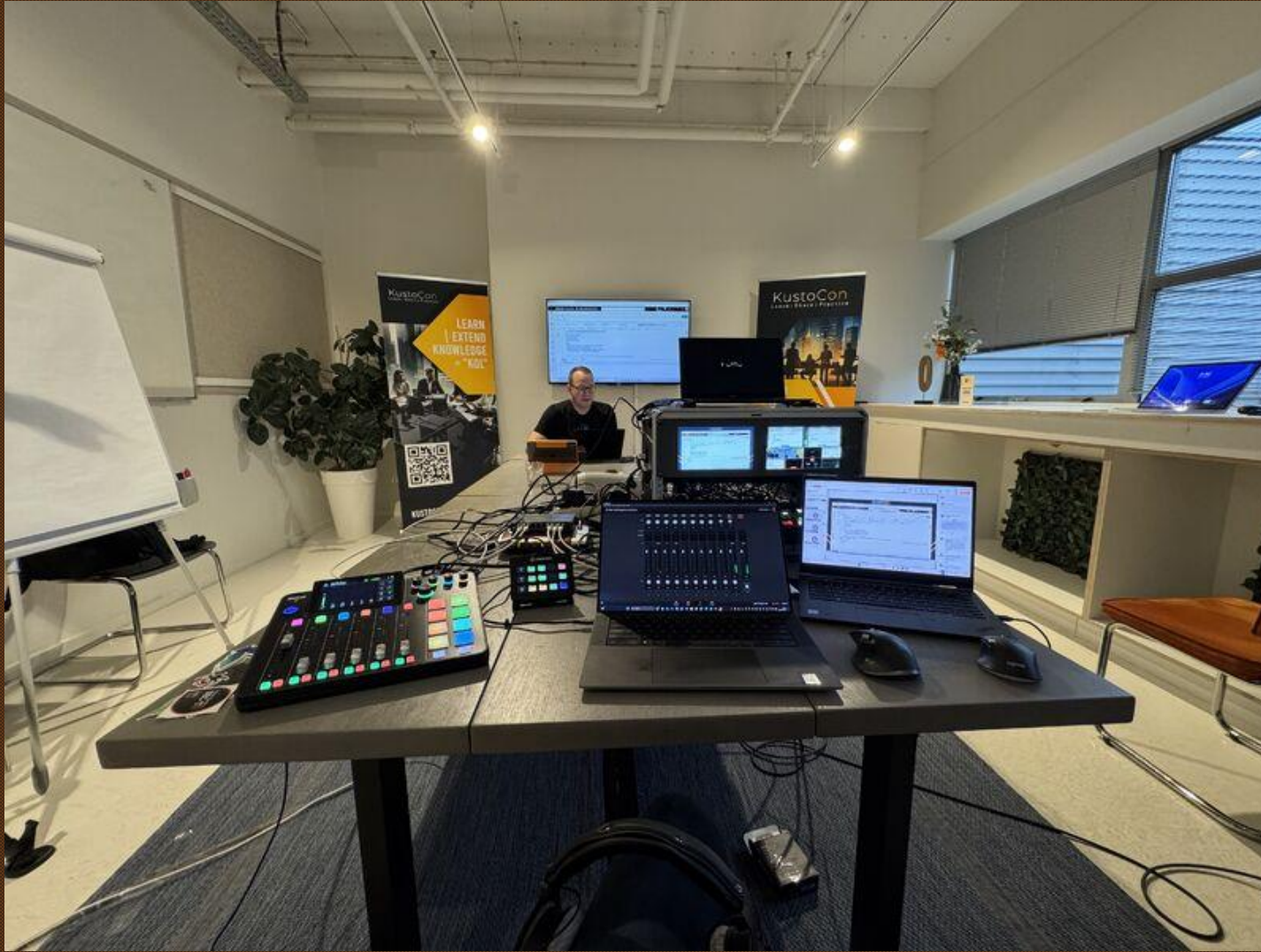


Throughout the Day we had between  
**350** and **290** online attendees





## News - KustoCon Recap – Behind the scenes



A BIG Thank you to Frans Oudendorp who managed the live stream



## News - KustoCon Recap - What Attendees Loved

### Practical Knowledge:

- Building KQL queries with examples of functions and their use cases.
- Real-world applications of KQL in threat hunting and security detection.



## News - KustoCon Recap - What Attendees Loved

### Exceptional Speakers:

- Olaf, Bert-Jan, Henning, and others received high praise for their expertise and presentation skills.
- Community-driven discussions with depth and technical rigor.

## News - KustoCon Recap - What Attendees Loved

### Valuable Insights:

- Sessions on Graph and Scan operators, Regex usage, and detection engineering.
- Learning about advanced topics like lateral movement, graph queries, and historical use of KQL.

## News - KustoCon Recap – Our Sponsors

baseVISION

glueckkanja

water

IT Security & Defense

prospex

KustoWorks

# News - Add icons to our KQL query results!

Sergio Albea

DeviceEvents

```
| where ActionType == "TamperingAttempt"  
| extend OriginalRegistryValue = tostring(parse_json(AdditionalFields).OriginalValue)  
| extend Status = tostring(parse_json(AdditionalFields).Status)  
| extend TamperingAction = tostring(parse_json(AdditionalFields).TamperingAction)  
| extend AttemptedRegistryValue = tostring(parse_json(AdditionalFields).TamperingAttemptedValue)  
| where TamperingAction == "RegistryModification"  
| extend TamperingAttemptStatus = case(  
    Status contains "Blocked", 0,  
    Status contains "Ignored", 1,  
    -1 )// Default value if neither "Blocked" nor "Ignored" is found)  
| extend Status_Result = iif(TamperingAttemptStatus == 0, '🟢💡', '🔴🚨')  
| distinct DeviceName, TamperingAction, Status_Result, Status, OriginalRegistryValue, AttemptedRegistryValue
```

TamperingAction	Status_Result	Status ↓
RegistryModification	🔴🚨	Ignored
RegistryModification	🔴🚨	Ignored
RegistryModification	🟢💡	Blocked
RegistryModification	🟢💡	Blocked
RegistryModification	🟢💡	Blocked
RegistryModification	🟢💡	Blocked

[https://www.linkedin.com/posts/sergioalbea\\_kqlquery-defenderxdr-cybersecurity-activity-7257747590866202626-Agl7?utm\\_source=share&utm\\_medium=member\\_desktop](https://www.linkedin.com/posts/sergioalbea_kqlquery-defenderxdr-cybersecurity-activity-7257747590866202626-Agl7?utm_source=share&utm_medium=member_desktop)

# News - Save money on your Sentinel ingestion costs with Data Collection Rules

*Brian Delaney, Andrea Fisher, and Jon Shectman*

This blog post explores an effective strategy to optimize data management by reducing log data volume while retaining critical information. It explains how to use Data Collection Rules (DCRs) to filter out less valuable log information, saving costs on data ingress and long-term storage while minimizing analyst fatigue. The post covers the decision-making process for identifying essential log data and provides step-by-step examples of applying DCRs to streamline log collection efficiently.

```
1 Usage
2 | where TimeGenerated > ago(30d)
3 | where IsBillable
4 | summarize SizeInGB=sum(Quantity) / 1000 by DataType
5 | sort by SizeInGB desc
```

<https://techcommunity.microsoft.com/blog/microsoftsentinelblog/save-money-on-your-sentinel-ingestion-costs-with-data-collection-rules/4270256>

# News - Country and Region Information in current\_principal\_details

The screenshot shows the Azure Data Explorer interface. At the top, there's a toolbar with a 'Run' button, a 'Recall' button, and a 'KQL tools' dropdown. The current database is 'MyFreeCluster/MyDatabase'. Below the toolbar, a KQL query is entered: `print details=current_principal_details()`. The results are displayed in a table view labeled 'Table 1'. The table has one column named 'details'. The data is a JSON object representing the current principal's details. The JPath is set to '/details', and the view is set to 'Full'.

```
1 print details=current_principal_details()
2
```

Table 1

details

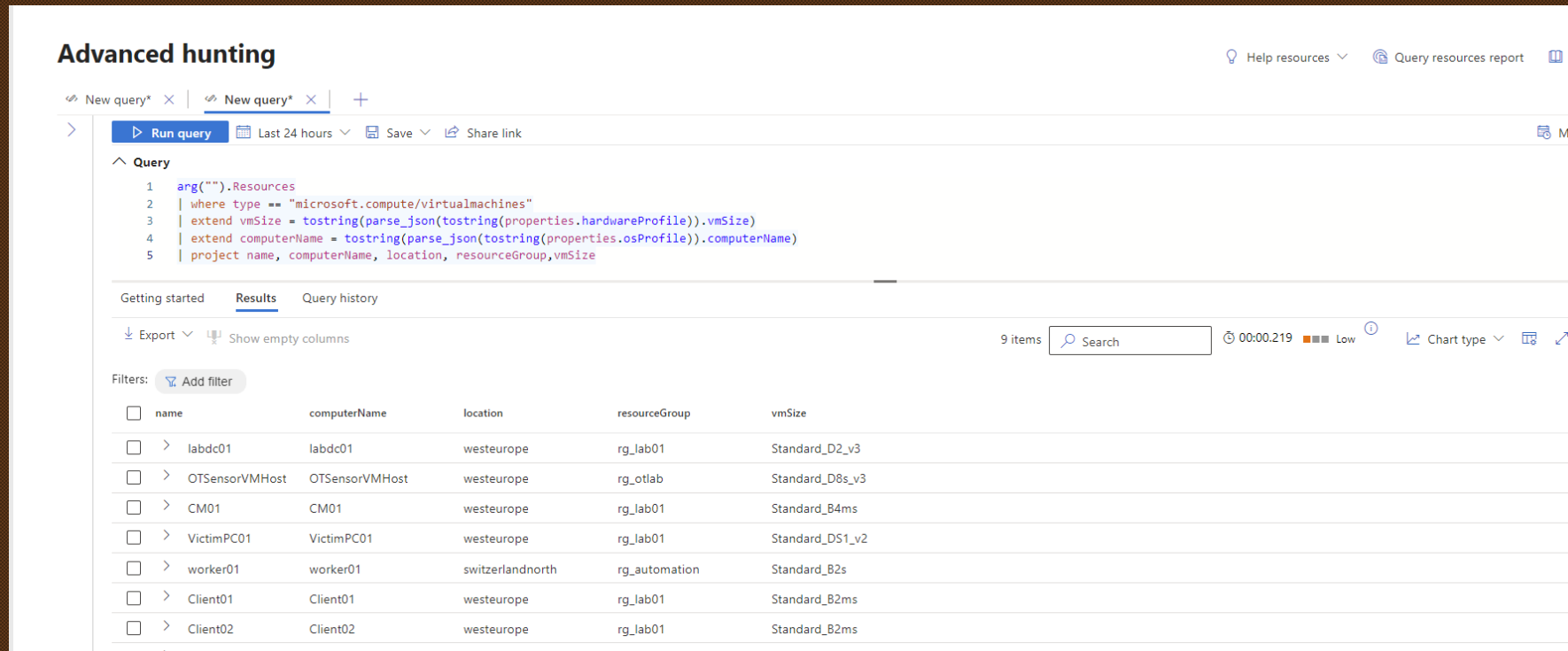
{ "TenantCountry": "CH", "TenantRegion": "EU", "UserPrincipalName": "oa@verboon.online", "Type": "aaduser", "IdentityProvider": "https://sts.windows.net", "DisplayName": "Alex Verboon - MTPLab (upn: [redacted])", "Authority": "[redacted]", "ObjectId": "[redacted]", "Mfa": "True", "FQN": "aaduser=[redacted]" }

<https://techcommunity.microsoft.com/blog/azuredataexplorer/country-and-region-information-in-current-principal-details/4275454>



# News – Defender XDR – arg() Operator

(GA) The arg() operator in advanced hunting in Microsoft Defender portal is now generally available. Users can now use the arg() operator for Azure Resource Graph queries to search over Azure resources, and no longer need to go to Log Analytics in Microsoft Sentinel to use this operator if already in Microsoft Defender.



The screenshot displays the 'Advanced hunting' interface in the Microsoft Defender portal. At the top, there's a header with 'Advanced hunting' and navigation links like 'Help resources', 'Query resources report', and 'Share link'. Below the header, a tab bar shows 'New query\*' and 'Run query'. The 'Run query' tab is active, displaying a Kusto query:

```
1 arg("").Resources
2 | where type == "microsoft.compute/virtualmachines"
3 | extend vmSize = tostring(parse_json(tostring(properties.hardwareProfile)).vmSize)
4 | extend computerName = tostring(parse_json(tostring(properties.osProfile)).computerName)
5 | project name, computerName, location, resourceGroup, vmSize
```

Below the query, the 'Results' tab is selected, showing a table with 9 items. The table has columns: name, computerName, location, resourceGroup, and vmSize. The results are as follows:

name	computerName	location	resourceGroup	vmSize
labdc01	labdc01	westeurope	rg_lab01	Standard_D2_v3
OTSensorVMHost	OTSensorVMHost	westeurope	rg_otlab	Standard_D8s_v3
CM01	CM01	westeurope	rg_lab01	Standard_B4ms
VictimPC01	VictimPC01	westeurope	rg_lab01	Standard_DS1_v2
worker01	worker01	switzerlandnorth	rg_automation	Standard_B2s
Client01	Client01	westeurope	rg_lab01	Standard_B2ms
Client02	Client02	westeurope	rg_lab01	Standard_B2ms
Client03	Client03	westeurope	rg_lab01	Standard_DS1_v2

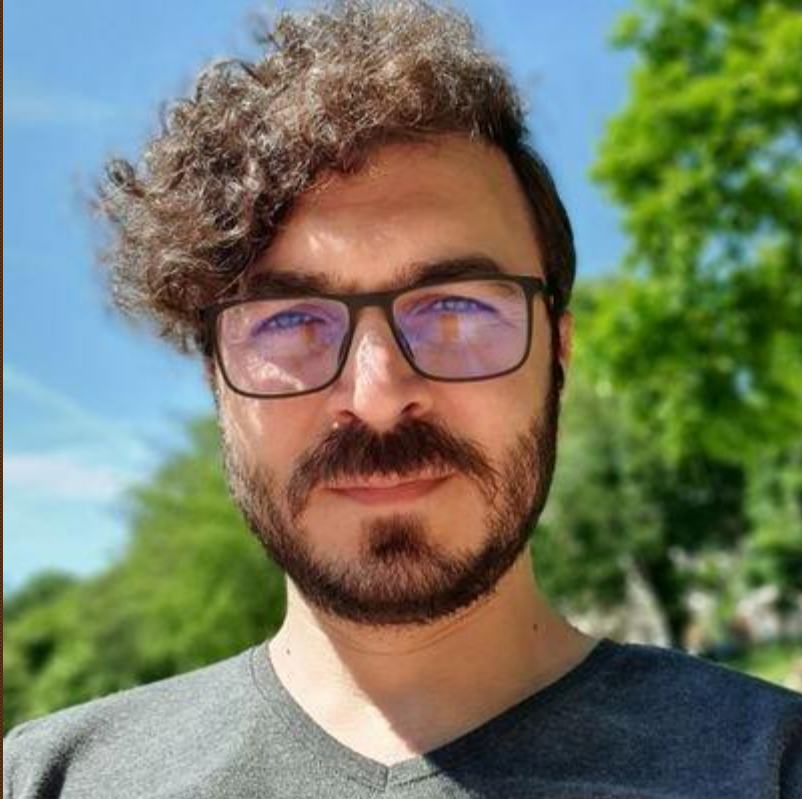
## News - Defender XDR - CloudProcessEvents

**(Preview)** The CloudProcessEvents table is now available for preview in advanced hunting. It contains information about process events in multicloud hosted environments such as Azure Kubernetes Service, Amazon Elastic Kubernetes Service, and Google Kubernetes Engine

You can use it to discover threats that can be observed through process details, like malicious processes or command-line signatures.

[CloudProcessEvents table in the advanced hunting schema - Microsoft Defender XDR | Microsoft Learn](#)

Our Guest:  
**Mehmet Ergene**



## Time Travelling in Logs



# What did you do with KQL this month?

## Azure DevOps – Organization Policy Change Monitoring

AzureDevOps - Additional Protection when using public package registries

AzureDevOps - Allow Public Projects

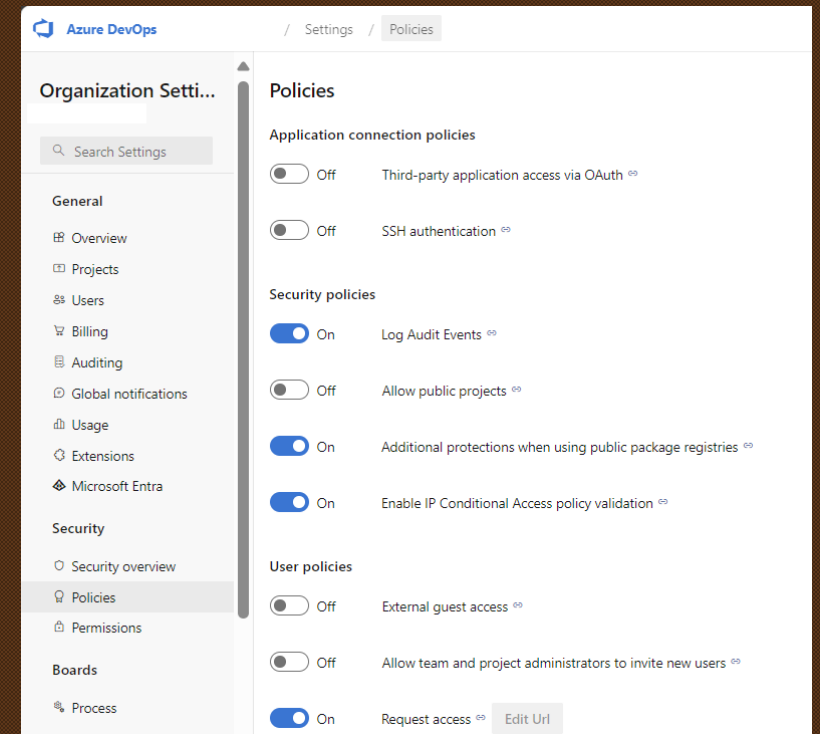
AzureDevOps - Enable IP Conditional Access policy validation

AzureDevOps - External Guest Access

AzureDevOps - Log Audit Events

AzureDevOps - SSH Authentication

AzureDevOps - Third-Party application Access via OAuth



<https://github.com/alexverboon/Hunting-Queries-Detection-Rules/tree/main/AzureDevOps>

# What did you do with KQL this month?

## KustoCon - How did we do

```
1 KustoCon2024
2 | summarize count(),
3   ['How many sessions did you view today_'] =
4     avg(['How many sessions did you view today_']), // 0-6
5   ['How would you rate the overall quality of the sessions_'] =
6     avg(['How would you rate the overall quality of the sessions_']), // 0-5
7   ['How would you rate the relevance of the session topics_'] =
8     avg(['How would you rate the relevance of the session topics_']), // 0-5
9   ['_How satisfied were you with the speakers_ expertise_'] =
10    avg(['_How satisfied were you with the speakers_ expertise_']) // 0-5
```

Table 1

+ Add visual

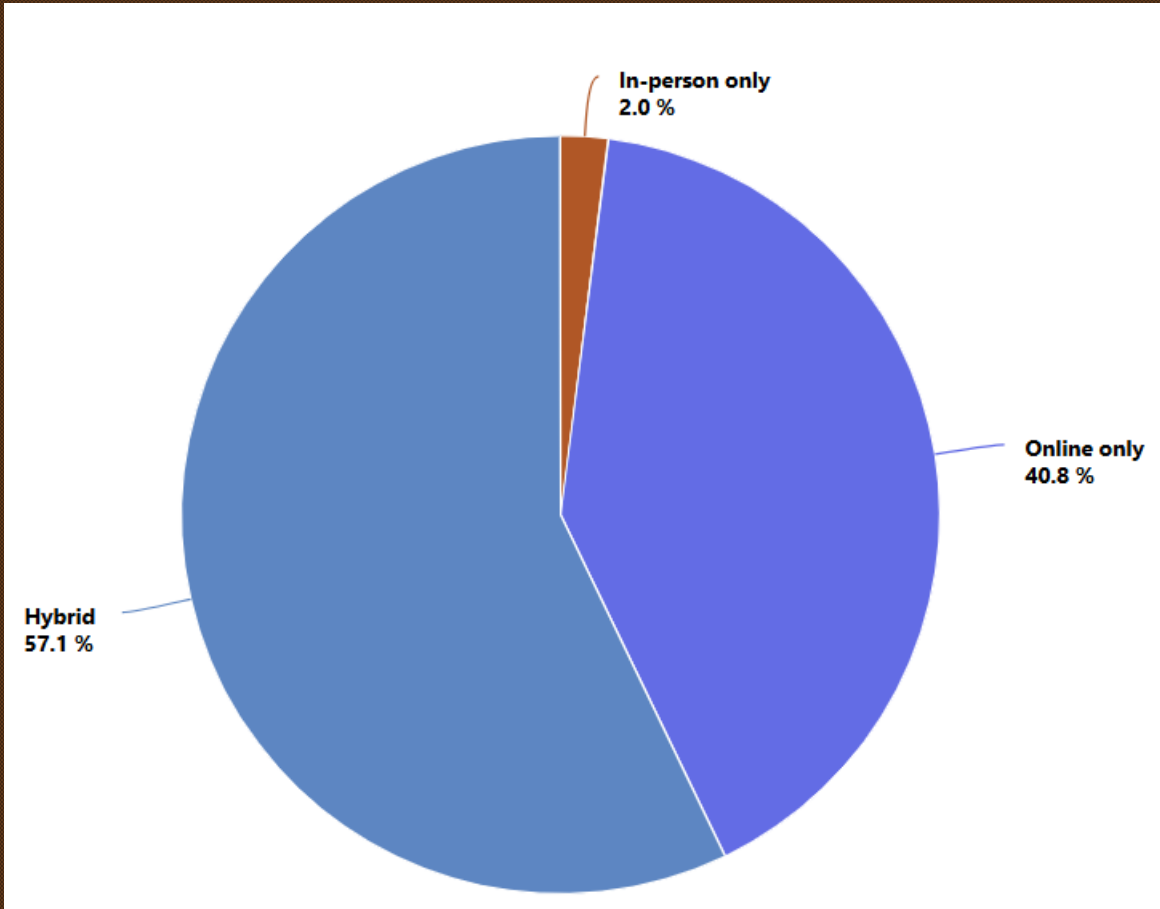
© Stats

Search

	count_	How many sessions did you view today_	How would you rate the overal...	How would you rate the releva...	_How satisfied were you with t...
>	49	5.020408163265306	4.6938775510204085	4.73469387755102	4.959183673469388

# What did you do with KQL this month?

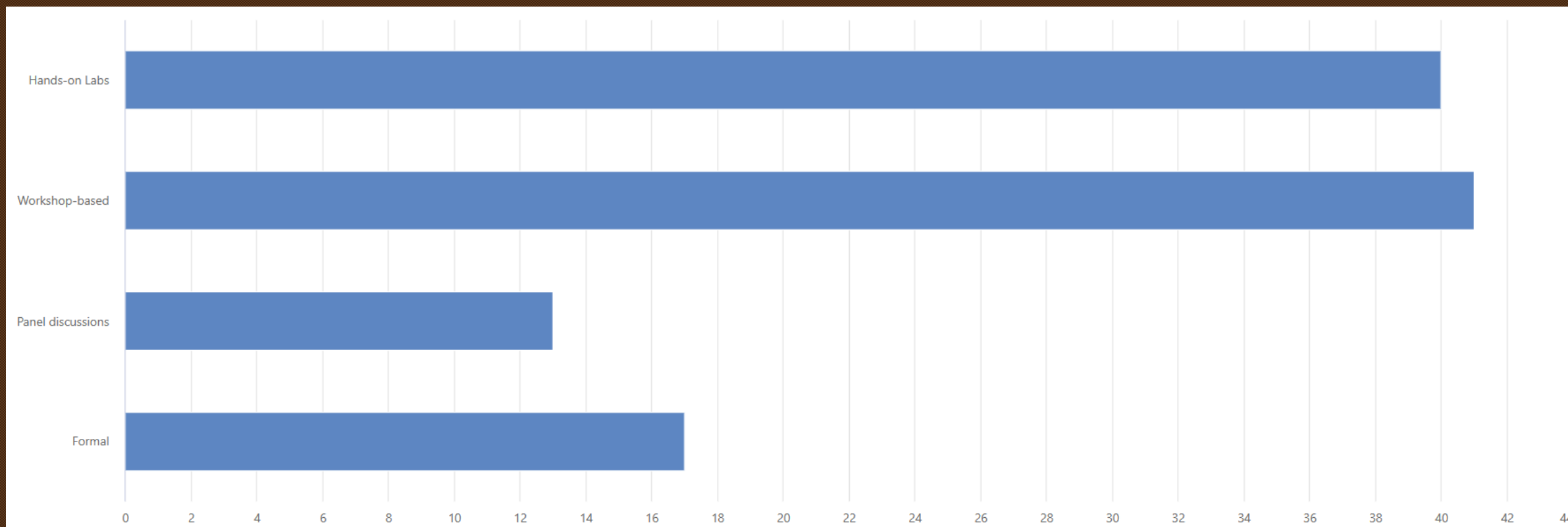
## KustoCon – Next KustoCon preferences





# What did you do with KQL this month?

## KustoCon – More Surveys....



# What did you do with KQL this month?

## KustoCon – Attendees



RAFFLE TIME



# The Definitive Guide to KQL

Using Kusto Query Language  
for operations, defending, and  
threat hunting

*Foreword by*

**Ann Johnson,**

Corporate Vice President, Security, Microsoft

Mark Morowczynski • Rod Trent • Matthew Zorich



Sample files  
on the web

# KustoCon

Learn | Share | Practice



<https://www.youtube.com/@KQLCafe/videos>



**KQL** | Cafe

Thanks  
for  
attendi  
ng