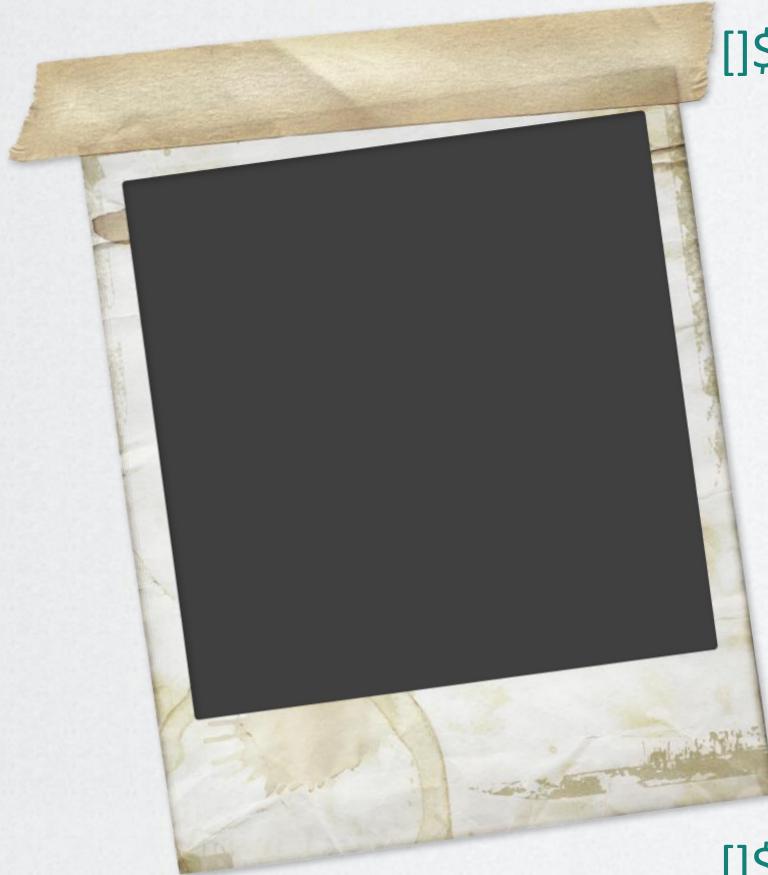


Secure Logstash connections to Microsoft Sentinel with R0t8r



Invoke-AzSpeaker | Format-List





```
[$] Invoke-AzSpeaker | Format-List
```

Koos Goossens

Microsoft Cloud &
Security Consultant



koos.goossens@wortell.nl



@koosgoossens

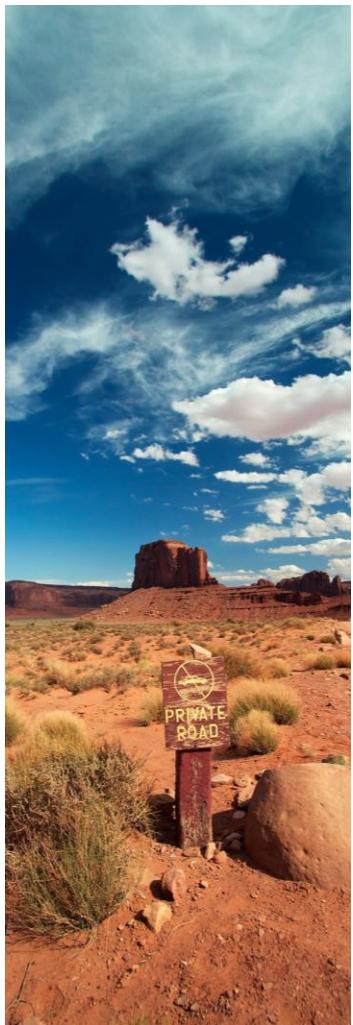


aka.ms/koos

```
[$]
```

_





|



Microsoft Sentinel



Agenda

- Log ingestion challenges
- Logstash?
- Managing keys
- R0t8r to the rescue?



Agenda

-
-
-
-



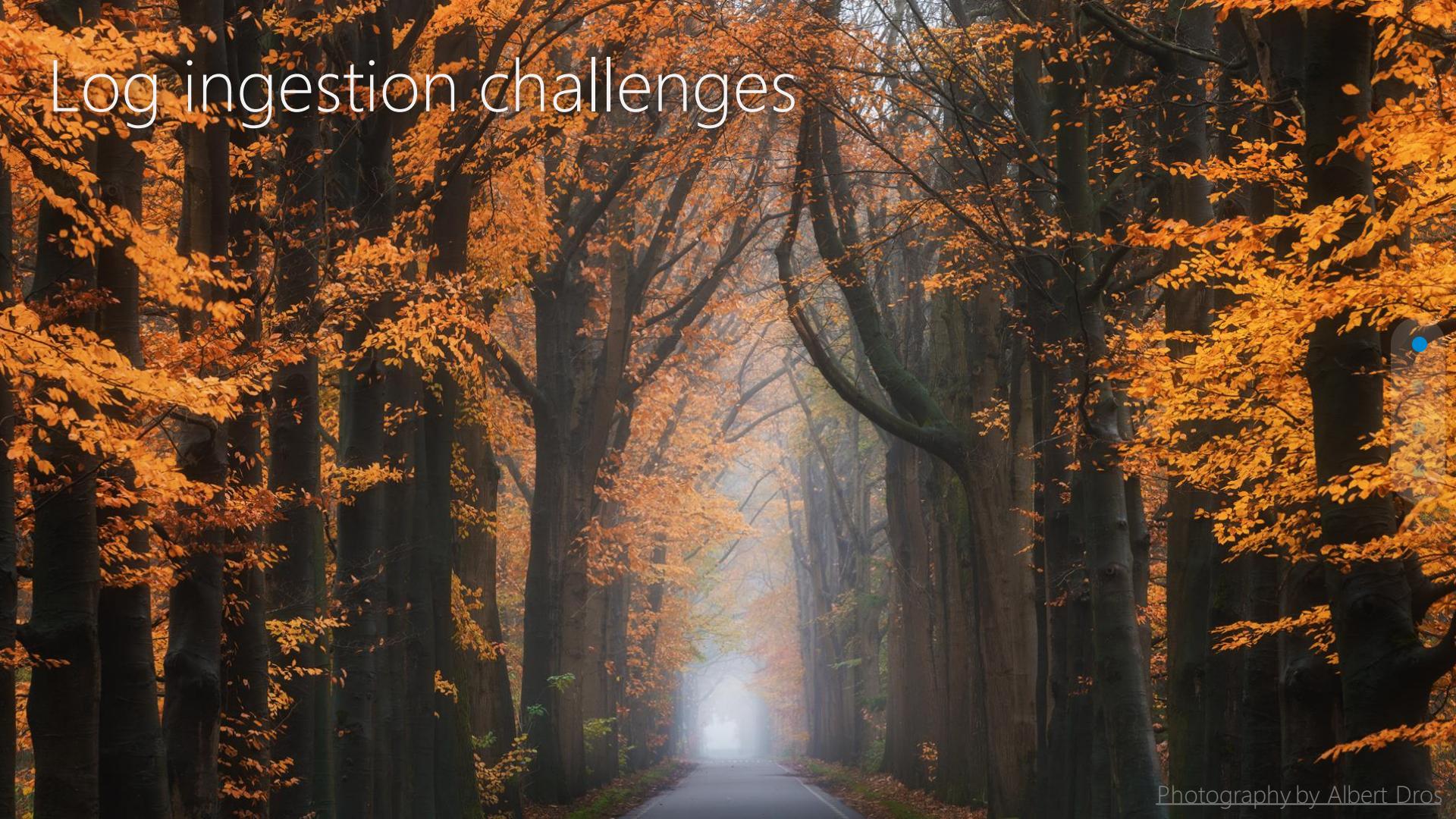


Photography by Albert Dros

Log ingestion challenges

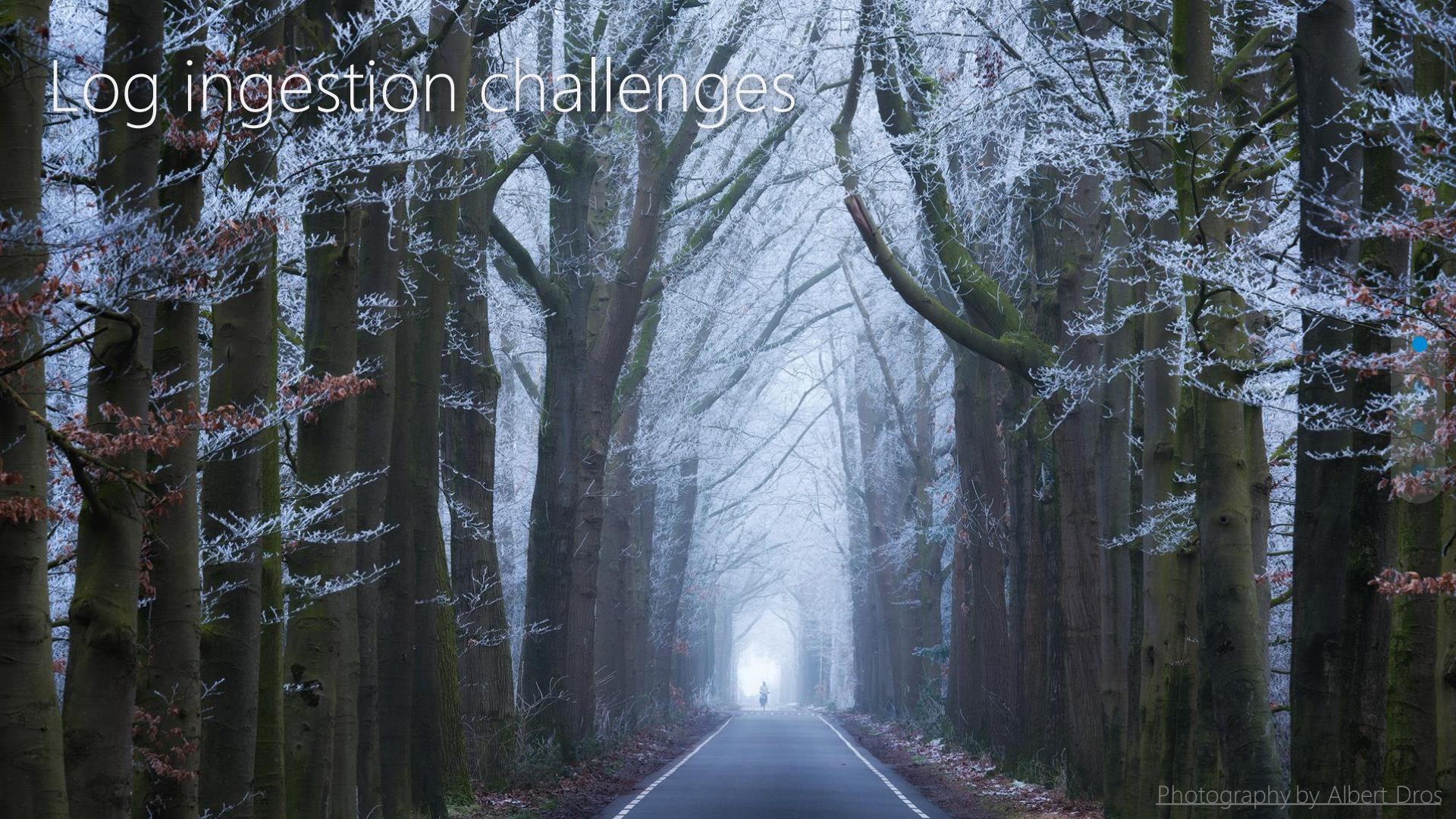
Photography by Albert Dros

Log ingestion challenges



Photography by Albert Dros

Log ingestion challenges

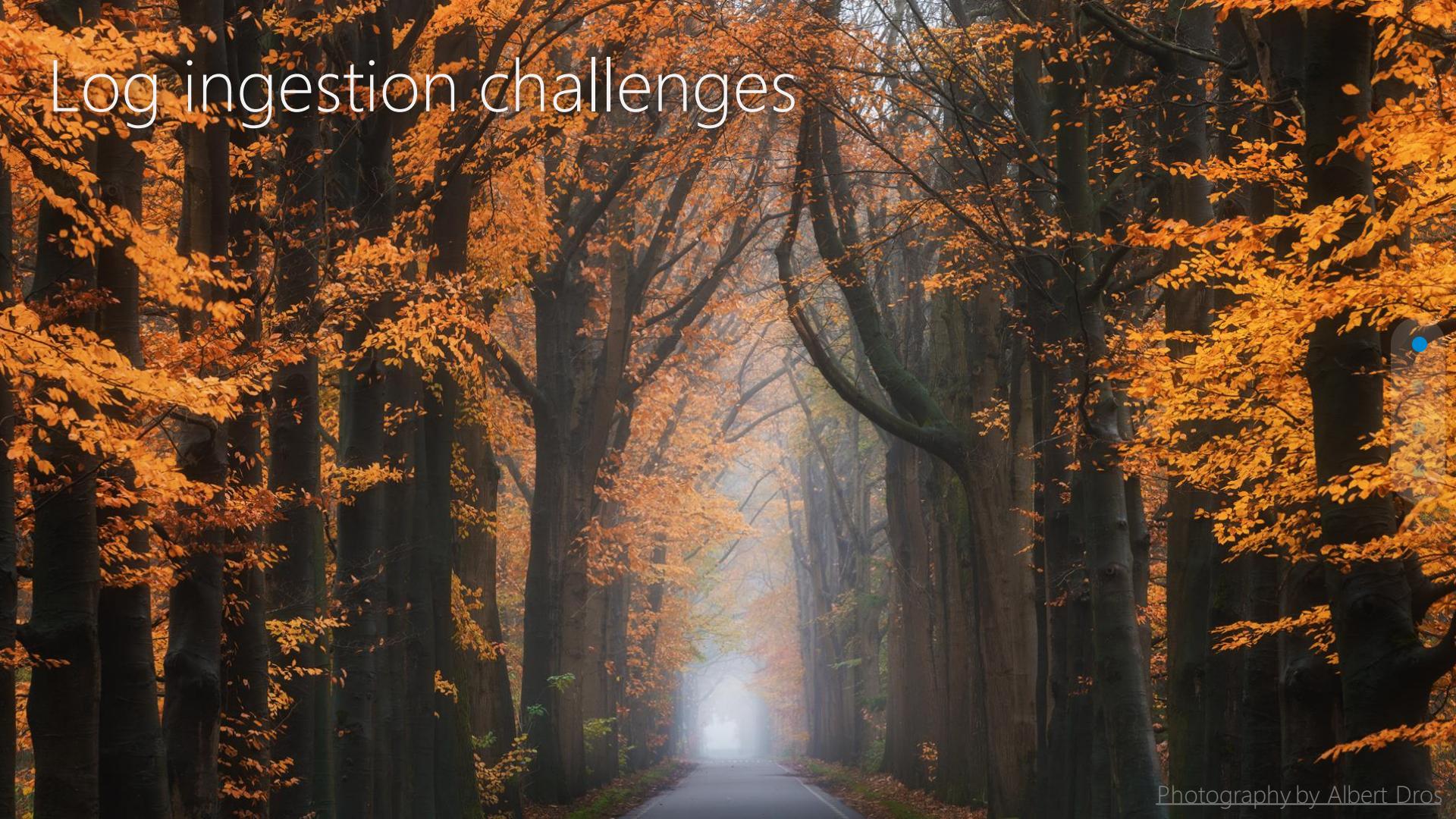
A photograph of a winter forest path. The scene is framed by tall, dark tree trunks on both sides, their branches heavily laden with white frost. A paved path leads into the distance, where a small figure of a person is visible walking away from the viewer. The ground is covered with fallen leaves and patches of snow. The overall atmosphere is serene and cold.

Photography by Albert Dros

Log ingestion challenges

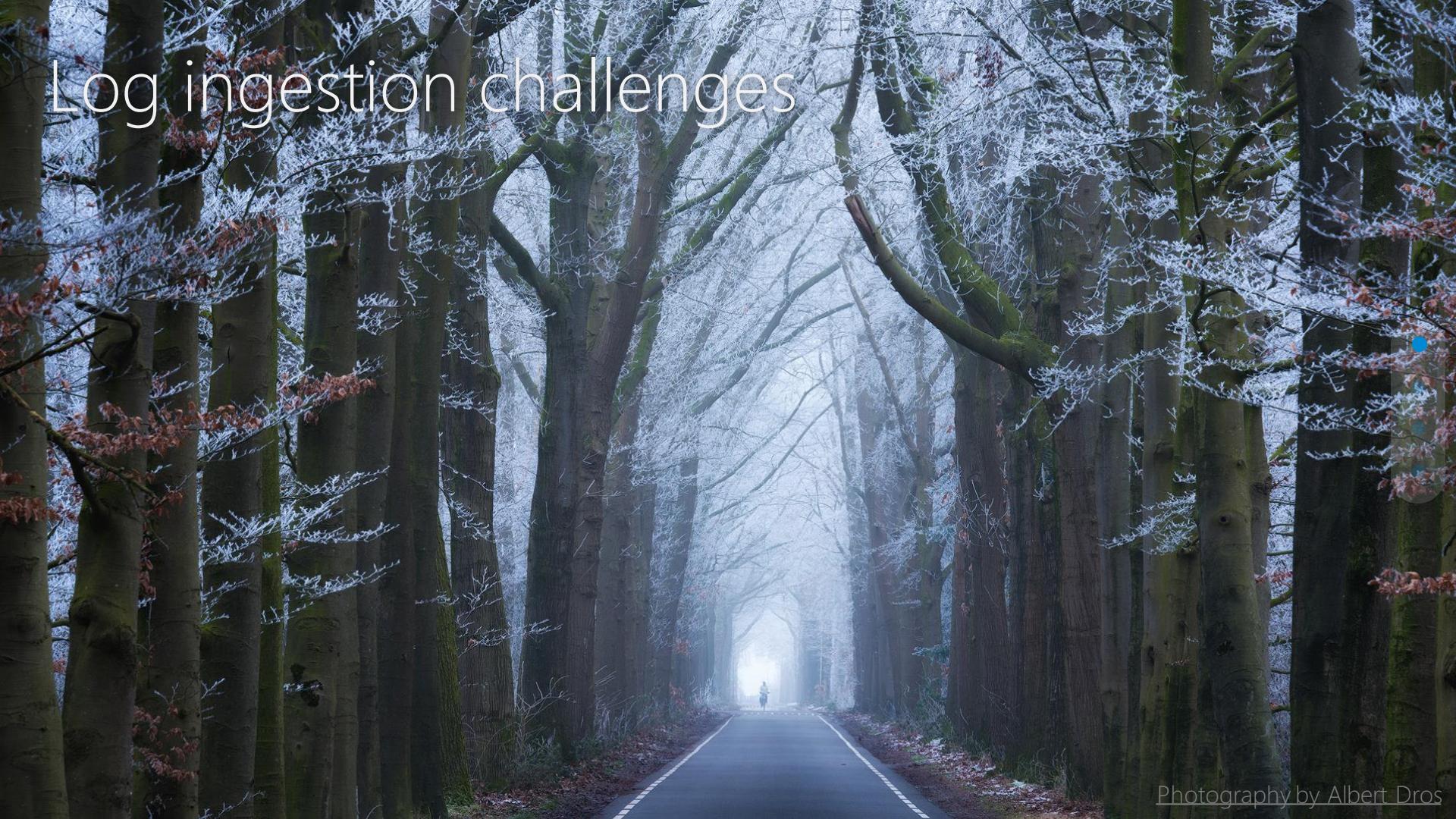
Photography by Albert Dros

Log ingestion challenges



Photography by Albert Dros

Log ingestion challenges

A photograph of a winter forest path. The scene is framed by tall, dark tree trunks on both sides, their branches heavily laden with white frost. A paved path leads into the distance, where a small figure of a person is visible walking away from the viewer. The ground is covered with fallen leaves and patches of snow. The overall atmosphere is serene and cold.

Photography by Albert Dros

Log ingestion challenges

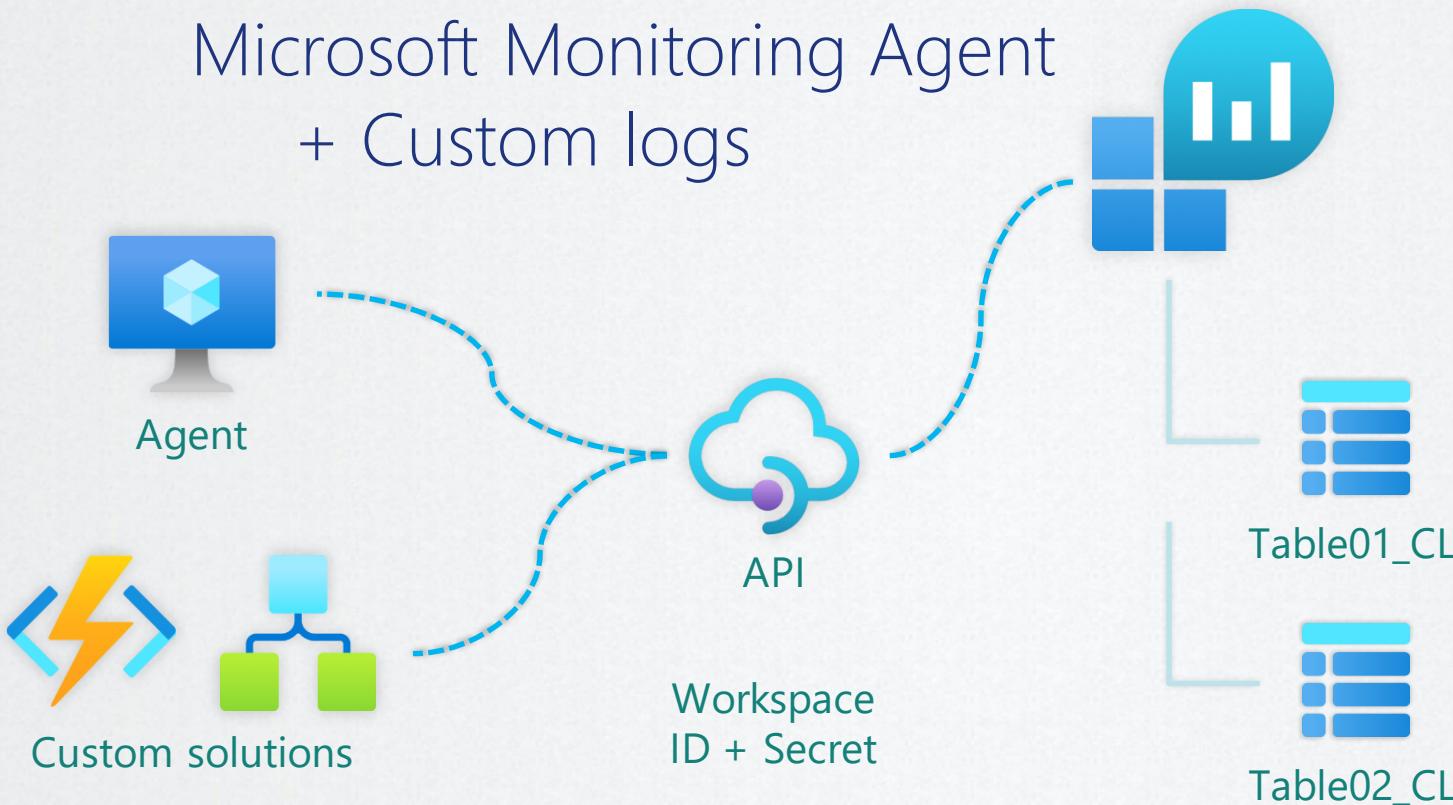


Log challenges

- Cloud is always changing
 - MMA → AMA

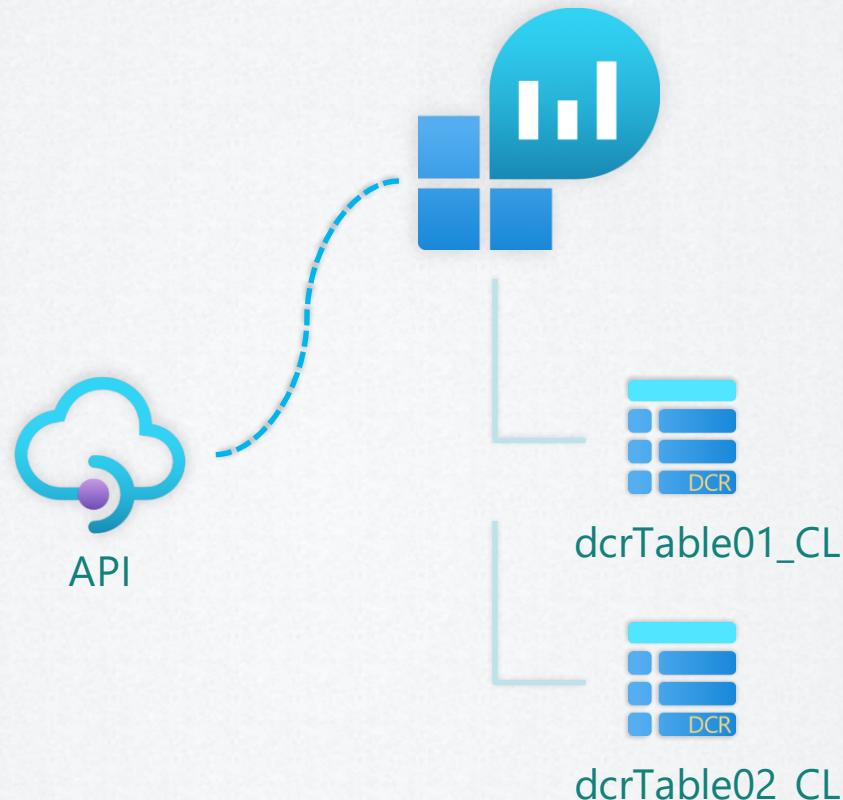


Microsoft Monitoring Agent + Custom logs



Microsoft Monitoring Agent + Custom logs





DCR-based custom logs

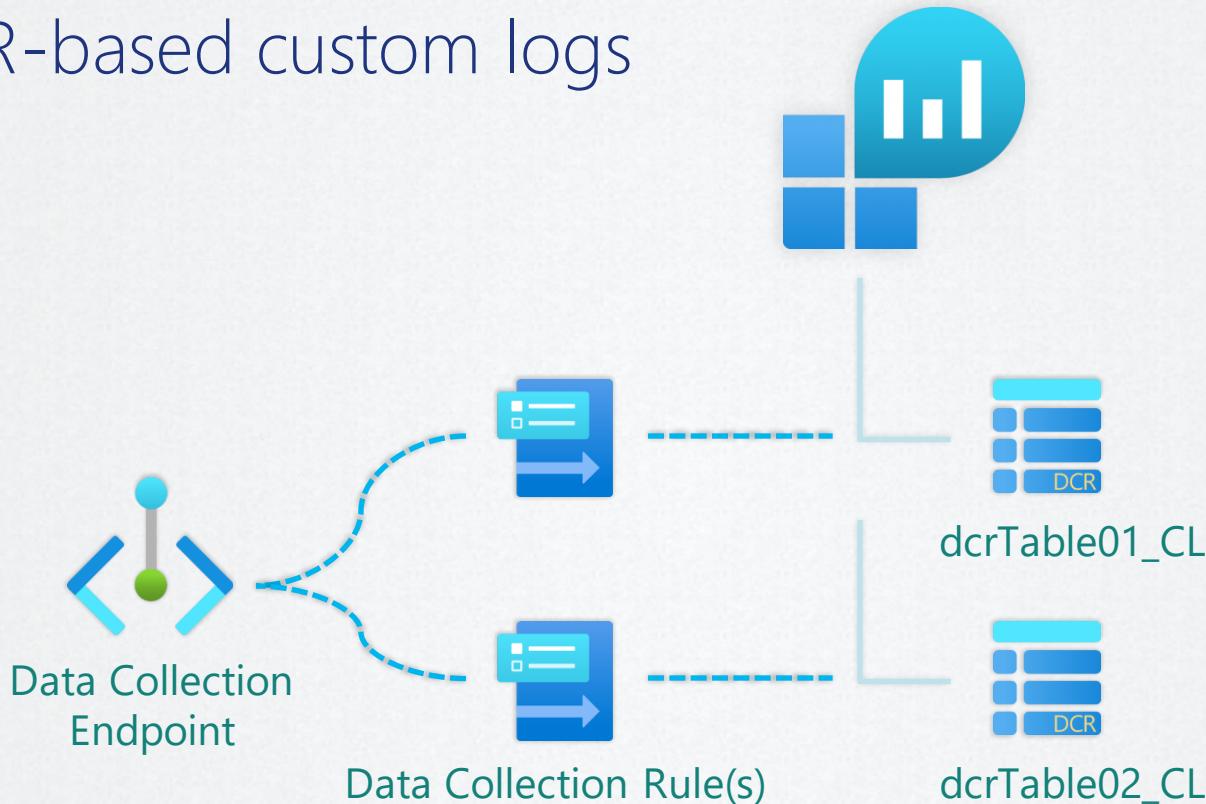


dcrTable01_CL

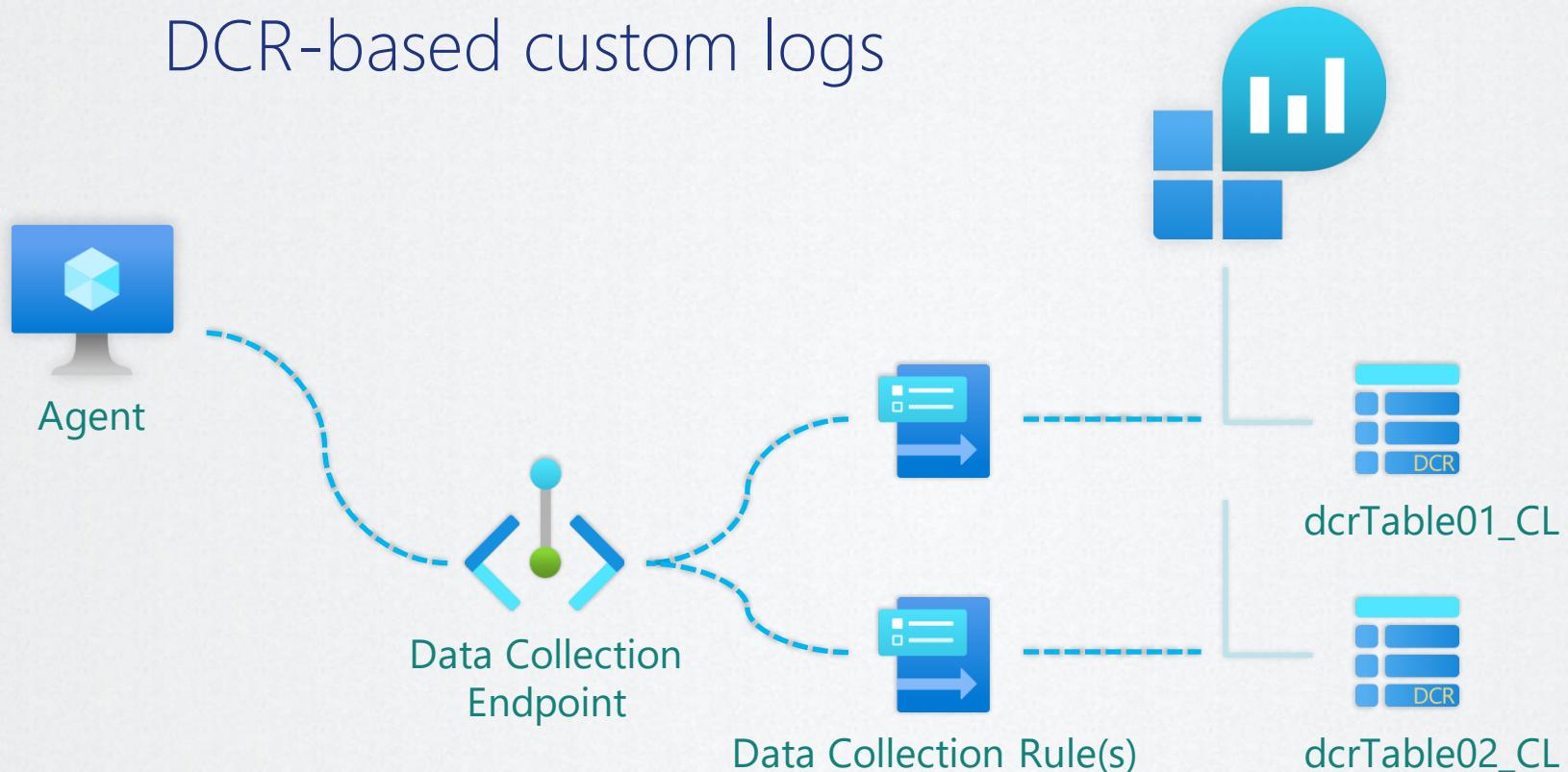


dcrTable02_CL

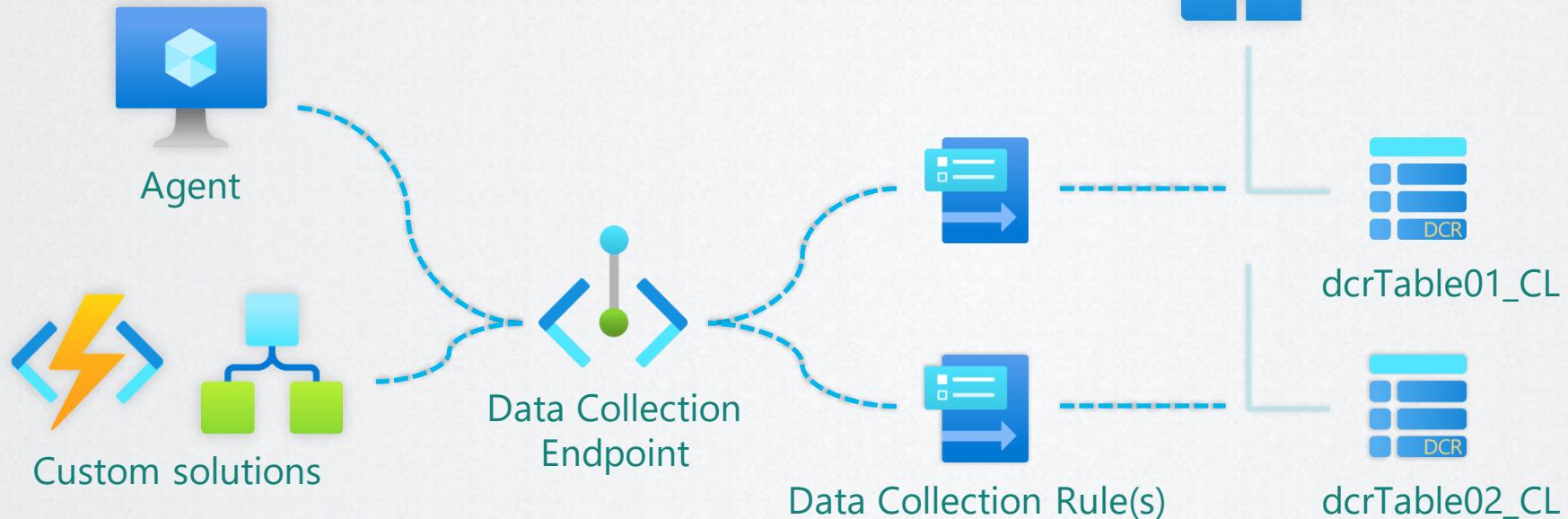
DCR-based custom logs



DCR-based custom logs

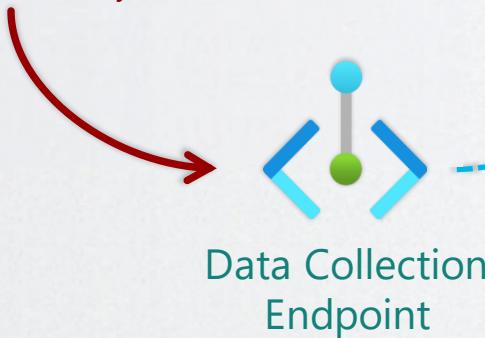


DCR-based custom logs

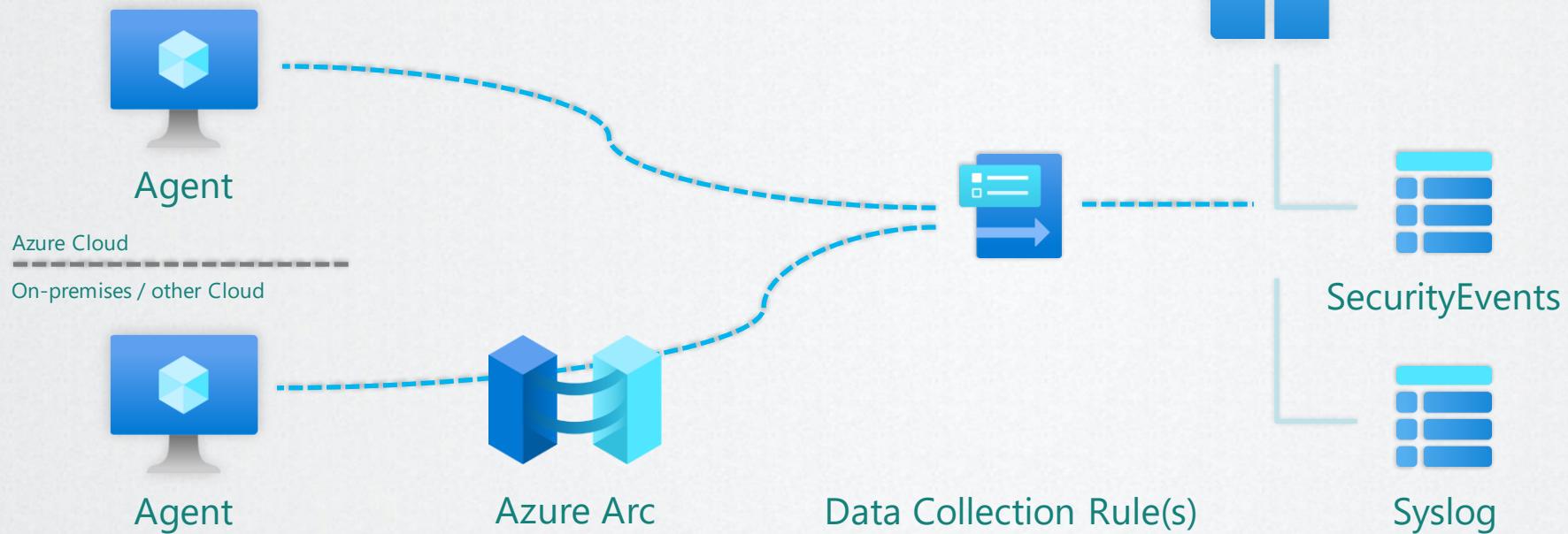


DCR-based custom logs

Authenticate with
application /
managed identity



Azure Monitoring Agent

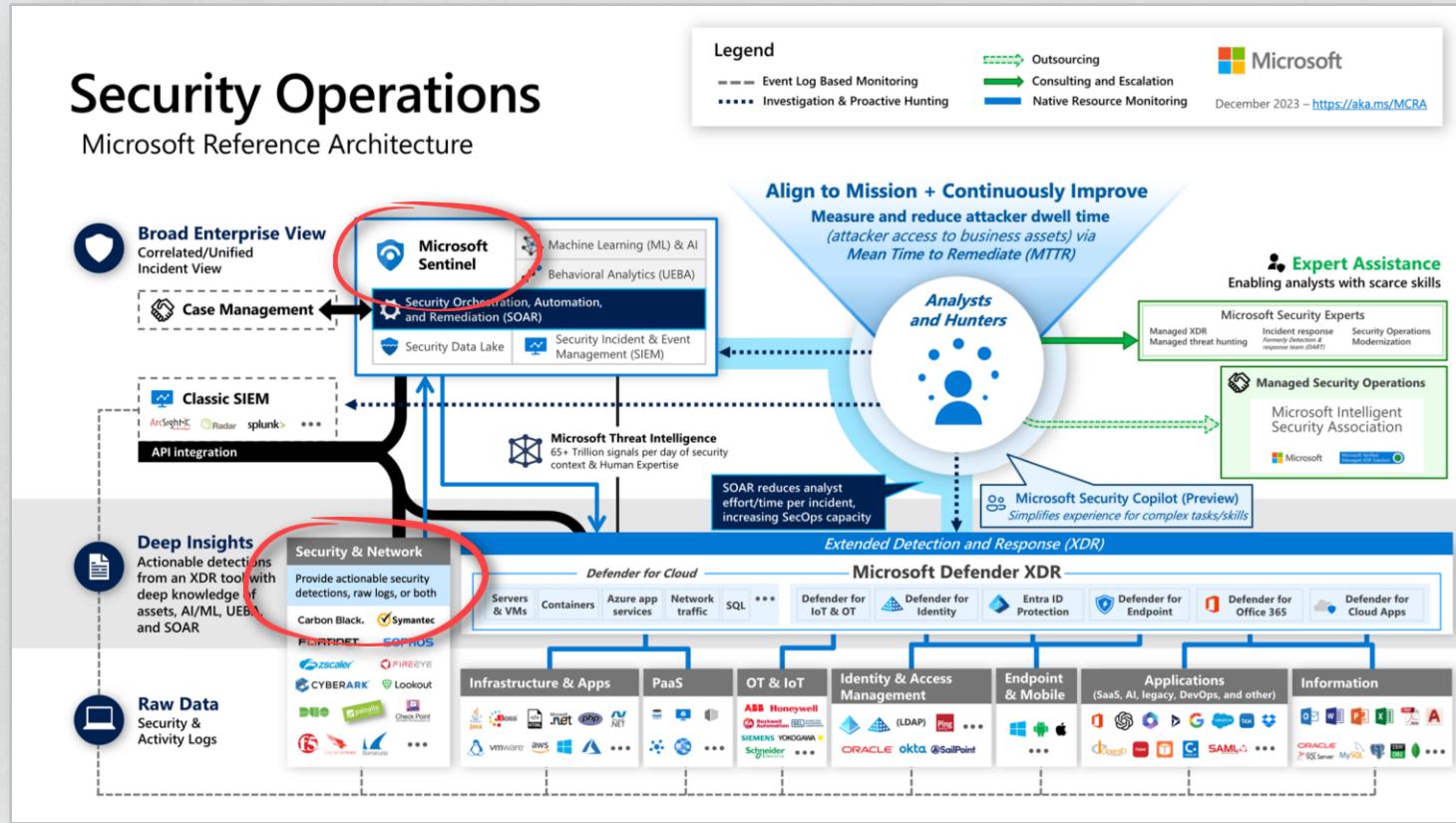


Log ingestion challenges

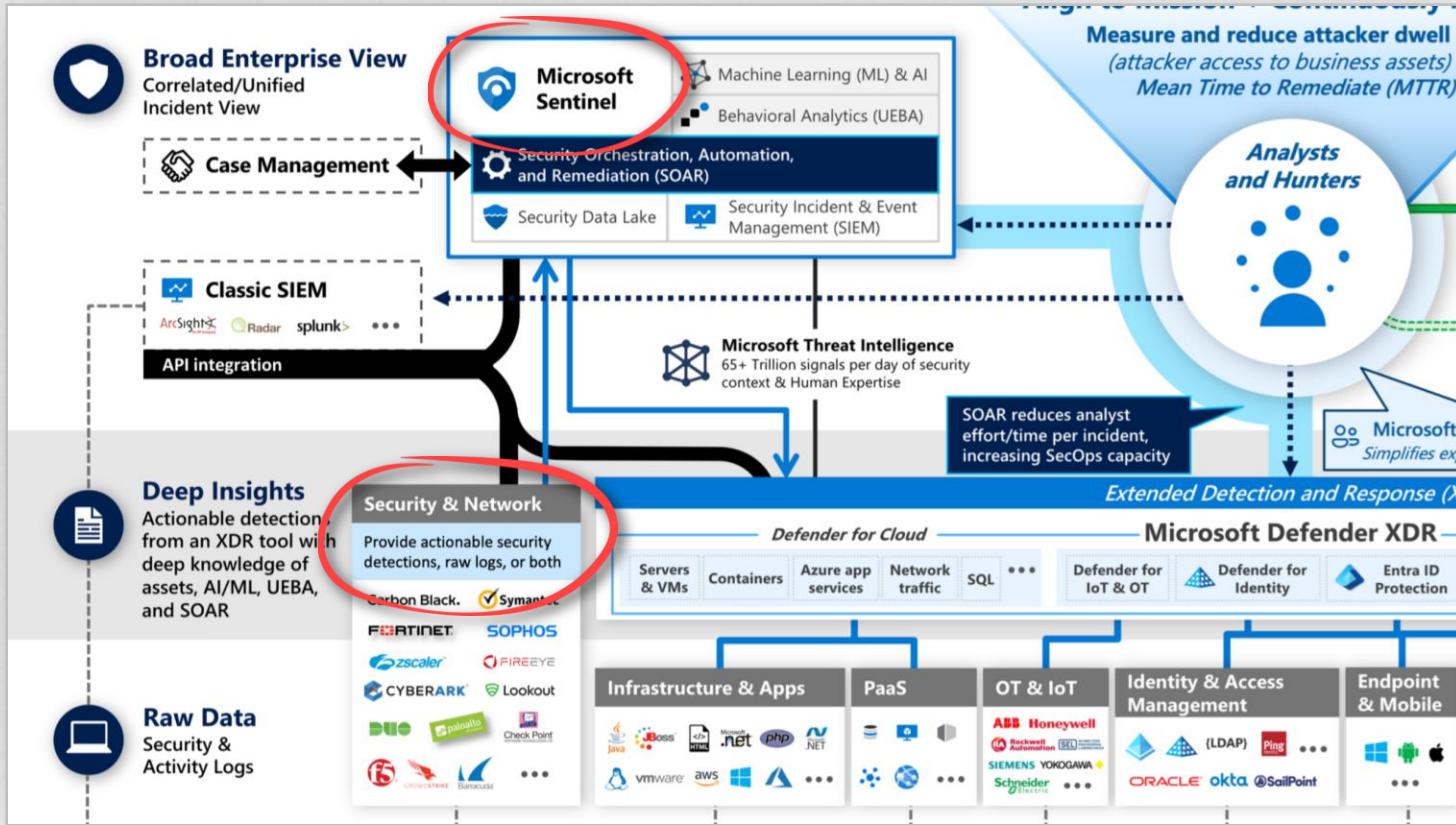
- Cloud is always changing
 - MMA → AMA
 - Lots of specific features (still) in preview
- Azure Arc is not to be underestimated
- Co-operate with other ops teams and vendors
 - Managing DCRs
- Sentinel as a “security data lake”?



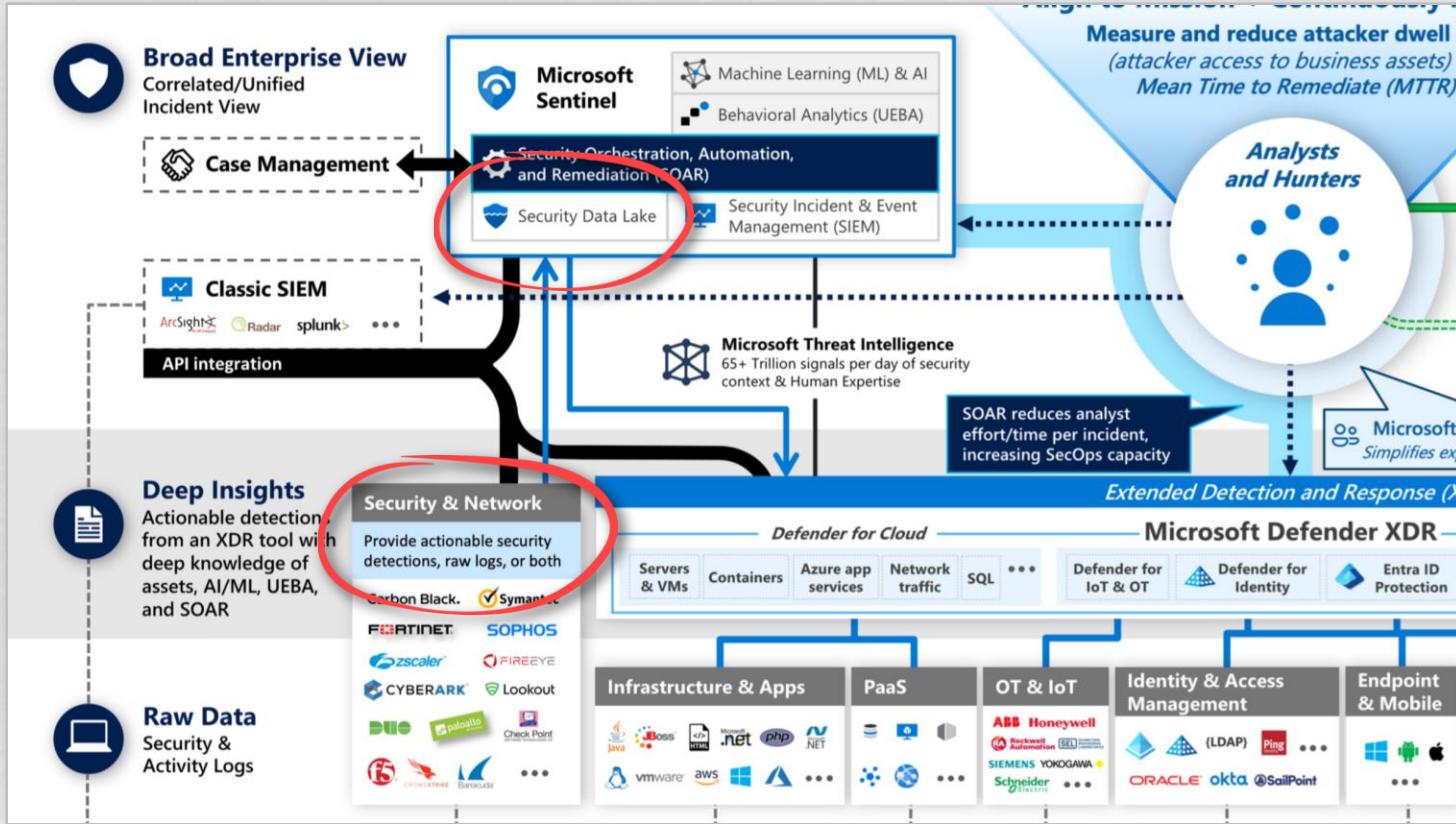
Microsoft Cybersecurity Reference Architectures (MCRA)



Microsoft Cybersecurity Reference Architectures (MCRA)



Microsoft Cybersecurity Reference Architectures (MCRA)



Microsoft Cybersecurity Reference Architectures (MCRA)

Multi-Cloud and Cross-Platform Technology

Secure the enterprise you have

Microsoft Purview

Discovery, Classify, Protect, and Monitor across unstructured data (documents, spreadsheets, files, etc.) and structured data (SQL, Databases, etc.) to identify and mitigate critical risks

Identity & Access

Identity Enablement

Access cloud and legacy applications for Enterprise users and External Identities like Partners (B2B) and Customers/Citizens (B2C)

Microsoft Entra (formerly Azure AD)

Information Protection

Identity Security

Zero Trust Access Control using Behavioral Analytics, Threat Intelligence, and integration of device and app trust signals

GitHub Advanced Security – Secure development capabilities

Securing components common most enterprise software supply chains

Endpoints & Devices

Microsoft Intune

Unified Endpoint Management (UEM)

Software as a Service (SaaS)

Hybrid Infrastructure – IaaS, PaaS, On-Premises

On-Premises

IaaS

AWS

PaaS

Cloud-native application protection platform (CNAPP)

Microsoft Defender (CSPM+CWPP), Entra Permissions Management (CIEM), Azure Security (CSNS), DevSecOps

IoT Devices

ABB YOKOGAWA Schneider Electric MITSUBISHI EATON

Honeywell Bristol-Myers BECKHOFF

SIEMENS MOTOROLA SEL TOSHIBA

Operational Technology (OT)

Security Operations [Center] (SOC)

Microsoft Sentinel

Cloud Native SIEM, SOAR, and UEBA for IT, OT, and IoT

Microsoft Defender XDR – Extended Detection and Response

Threat visibility and capabilities tailored to resources

Microsoft Defender for Endpoint

Unified Endpoint Security

- Endpoint Detection & Response (EDR)
- Data Loss Protection (DLP)
- Web Content Filtering
- Threat & Vuln Management

Microsoft Defender for Cloud Apps

- App Discovery & Risk Scoring (Shadow IT)
- Threat Detection & Response
- Policy Audit & Enforcement
- Session monitoring & control
- Info Protection & Data Loss Prevention (DLP)

Microsoft Azure Arc

Microsoft Defender for Cloud

IaaS, PaaS, and On-Premises

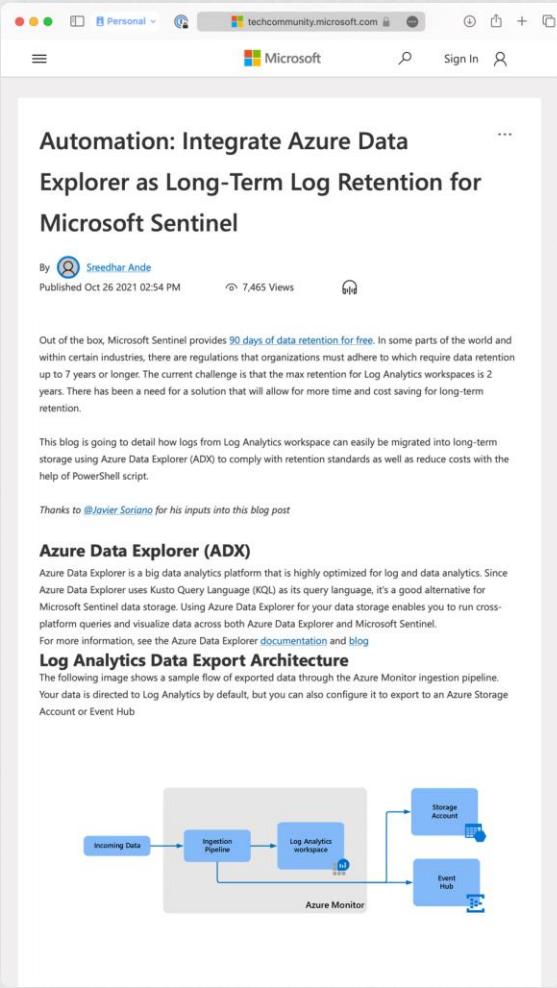
- VMs, Servers, App Environments
- Storage and Databases
- Containers and Orchestration
- DevOps, APIs, CI/CD, and more

Microsoft Defender for IoT

- ICS, SCADA, OT
- Internet of Things (IoT)
- Asset & Vulnerability management
- Industrial IoT (IIoT)
- Threat Detection & Response

Threat Intelligence – 65+ Trillion signals per day of security context



A screenshot of a Microsoft Edge browser window displaying a blog post from the Microsoft Tech Community. The title of the post is "Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel". It was written by Sreedhar Ande and published on October 26, 2021, at 02:54 PM. The post has 7,465 views. The content discusses the challenges of long-term log retention in Microsoft Sentinel and how Azure Data Explorer (ADX) can be used as an alternative. It includes a note of thanks to @Javier_Soriano for inputs. Below the main content, there's a section titled "Azure Data Explorer (ADX)" which provides an overview of the platform and its integration with Microsoft Sentinel. There's also a "Log Analytics Data Export Architecture" diagram.

Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel

By  Sreedhar Ande
Published Oct 26 2021 02:54 PM · 7,465 Views

Out of the box, Microsoft Sentinel provides 90 days of data retention for free. In some parts of the world and within certain industries, there are regulations that organizations must adhere to which require data retention up to 7 years or longer. The current challenge is that the max retention for Log Analytics workspaces is 2 years. There has been a need for a solution that will allow for more time and cost saving for long-term retention.

This blog is going to detail how logs from Log Analytics workspace can easily be migrated into long-term storage using Azure Data Explorer (ADX) to comply with retention standards as well as reduce costs with the help of PowerShell script.

Thanks to [@Javier_Soriano](#) for his inputs into this blog post

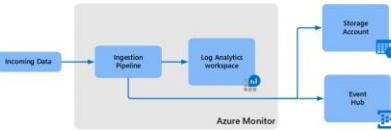
Azure Data Explorer (ADX)

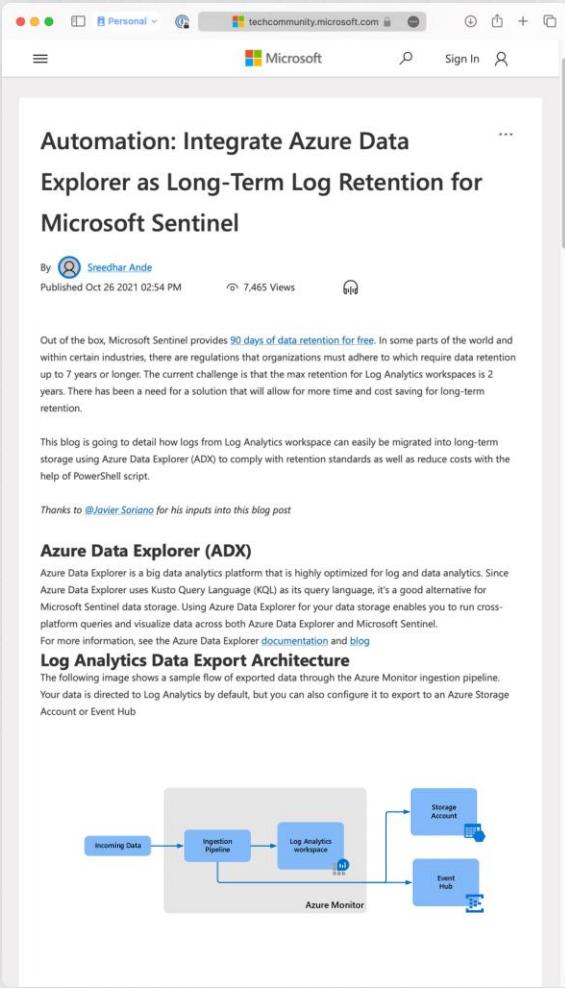
Azure Data Explorer is a big data analytics platform that is highly optimized for log and data analytics. Since Azure Data Explorer uses Kusto Query Language (KQL) as its query language, it's a good alternative for Microsoft Sentinel data storage. Using Azure Data Explorer for your data storage enables you to run cross-platform queries and visualize data across both Azure Data Explorer and Microsoft Sentinel.

For more information, see the [Azure Data Explorer documentation](#) and [blog](#)

Log Analytics Data Export Architecture

The following image shows a sample flow of exported data through the Azure Monitor ingestion pipeline. Your data is directed to Log Analytics by default, but you can also configure it to export to an Azure Storage Account or Event Hub



A screenshot of a Microsoft Edge browser window. The address bar shows 'techcommunity.microsoft.com'. The page title is 'Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel'. Below the title, it says 'By Sreedhar Ande' and 'Published Oct 26 2021 02:54 PM'. It has 7,465 views. The main content discusses integrating Azure Data Explorer for long-term log retention in Microsoft Sentinel, mentioning 90 days of free retention and the need for PowerShell scripts. It also thanks @Javier Soriano for inputs.

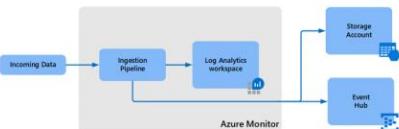
Azure Data Explorer (ADX)

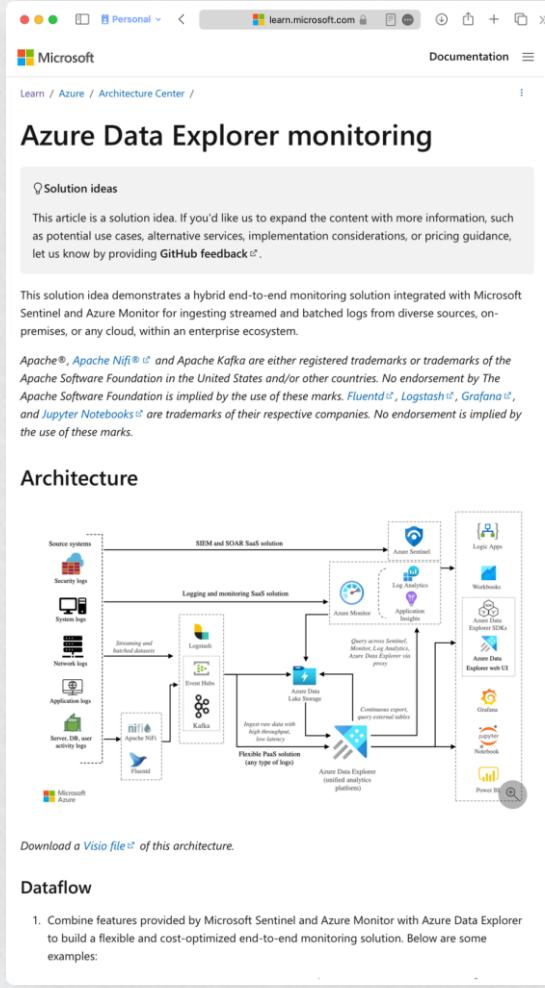
Azure Data Explorer is a big data analytics platform that is highly optimized for log and data analytics. Since Azure Data Explorer uses Kusto Query Language (KQL) as its query language, it's a good alternative for Microsoft Sentinel data storage. Using Azure Data Explorer for your data storage enables you to run cross-platform queries and visualize data across both Azure Data Explorer and Microsoft Sentinel.

For more information, see the Azure Data Explorer [documentation](#) and [blog](#).

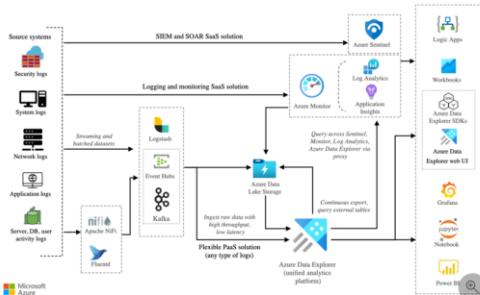
Log Analytics Data Export Architecture

The following image shows a sample flow of exported data through the Azure Monitor ingestion pipeline. Your data is directed to Log Analytics by default, but you can also configure it to export to an Azure Storage Account or Event Hub.



A screenshot of a Microsoft Edge browser window. The address bar shows 'learn.microsoft.com'. The page title is 'Azure Data Explorer monitoring'. It features a 'Solution ideas' section with a note about potential use cases and GitHub feedback. The main content describes a hybrid end-to-end monitoring solution integrated with Microsoft Sentinel and Azure Monitor for ingesting streamed and batched logs from diverse sources, on-premises, or any cloud, within an enterprise ecosystem. It includes a detailed architecture diagram and a 'Dataflow' section with numbered steps.

Architecture

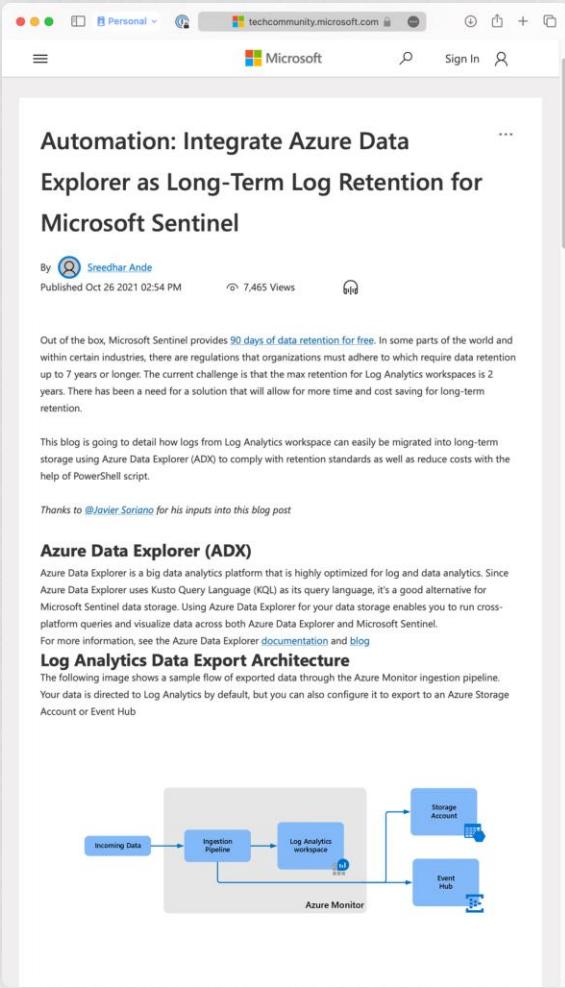


[Download a Visio file](#) of this architecture.

Dataflow

1. Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:



A screenshot of a Microsoft Edge browser window. The address bar shows 'techcommunity.microsoft.com'. The page title is 'Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel'. Below the title, it says 'By Sreedhar Ande' and 'Published Oct 26 2021 02:54 PM'. There are 7,465 views. The main content discusses the integration of Azure Data Explorer for long-term log retention in Microsoft Sentinel, mentioning 90 days of data retention for free and regulations requiring up to 7 years. It also notes the current challenge of max retention for Log Analytics workspaces being 2 years. The author thanks @Javier Soriano for inputs.

Out of the box, Microsoft Sentinel provides 90 days of data retention for free. In some parts of the world and within certain industries, there are regulations that organizations must adhere to which require data retention up to 7 years or longer. The current challenge is that the max retention for Log Analytics workspaces is 2 years. There has been a need for a solution that will allow for more time and cost saving for long-term retention.

This blog is going to detail how logs from Log Analytics workspace can easily be migrated into long-term storage using Azure Data Explorer (ADX) to comply with retention standards as well as reduce costs with the help of PowerShell script.

Thanks to [@Javier Soriano](#) for his inputs into this blog post

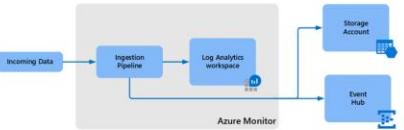
Azure Data Explorer (ADX)

Azure Data Explorer is a big data analytics platform that is highly optimized for log and data analytics. Since Azure Data Explorer uses Kusto Query Language (KQL) as its query language, it's a good alternative for Microsoft Sentinel data storage. Using Azure Data Explorer for your data storage enables you to run cross-platform queries and visualize data across both Azure Data Explorer and Microsoft Sentinel.

For more information, see the Azure Data Explorer [documentation](#) and [blog](#).

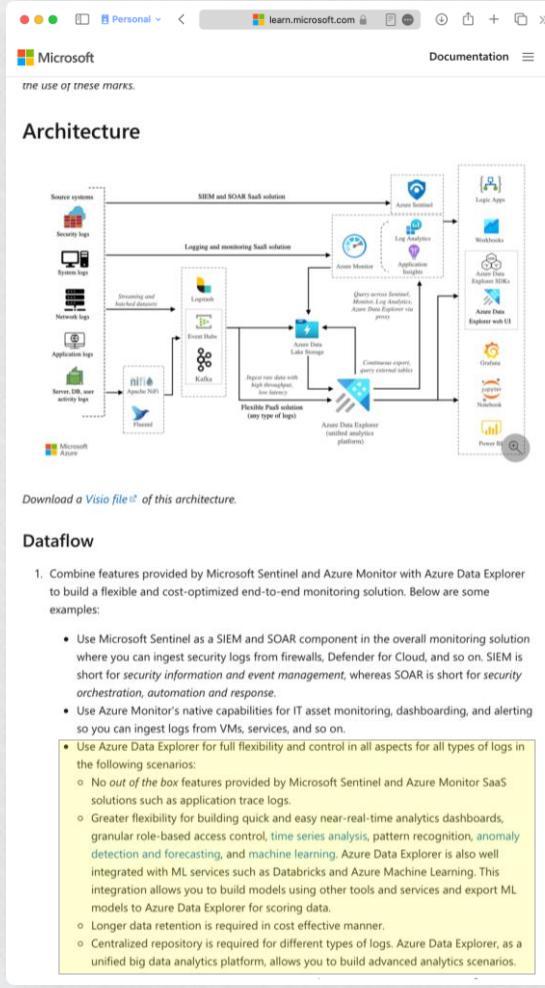
Log Analytics Data Export Architecture

The following image shows a sample flow of exported data through the Azure Monitor ingestion pipeline. Your data is directed to Log Analytics by default, but you can also configure it to export to an Azure Storage Account or Event Hub.



```

graph LR
    ID[Incoming Data] --> IP[Ingestion Pipeline]
    IP --> LA[Log Analytics workspace]
    IP --> AM[Event Hub]
    LA --> SA[Storage Account]
    LA --> EH[Event Hub]
    style LA fill:#0072bc,color:#fff
    style AM fill:#0072bc,color:#fff
    style SA fill:#0072bc,color:#fff
    style EH fill:#0072bc,color:#fff
  
```

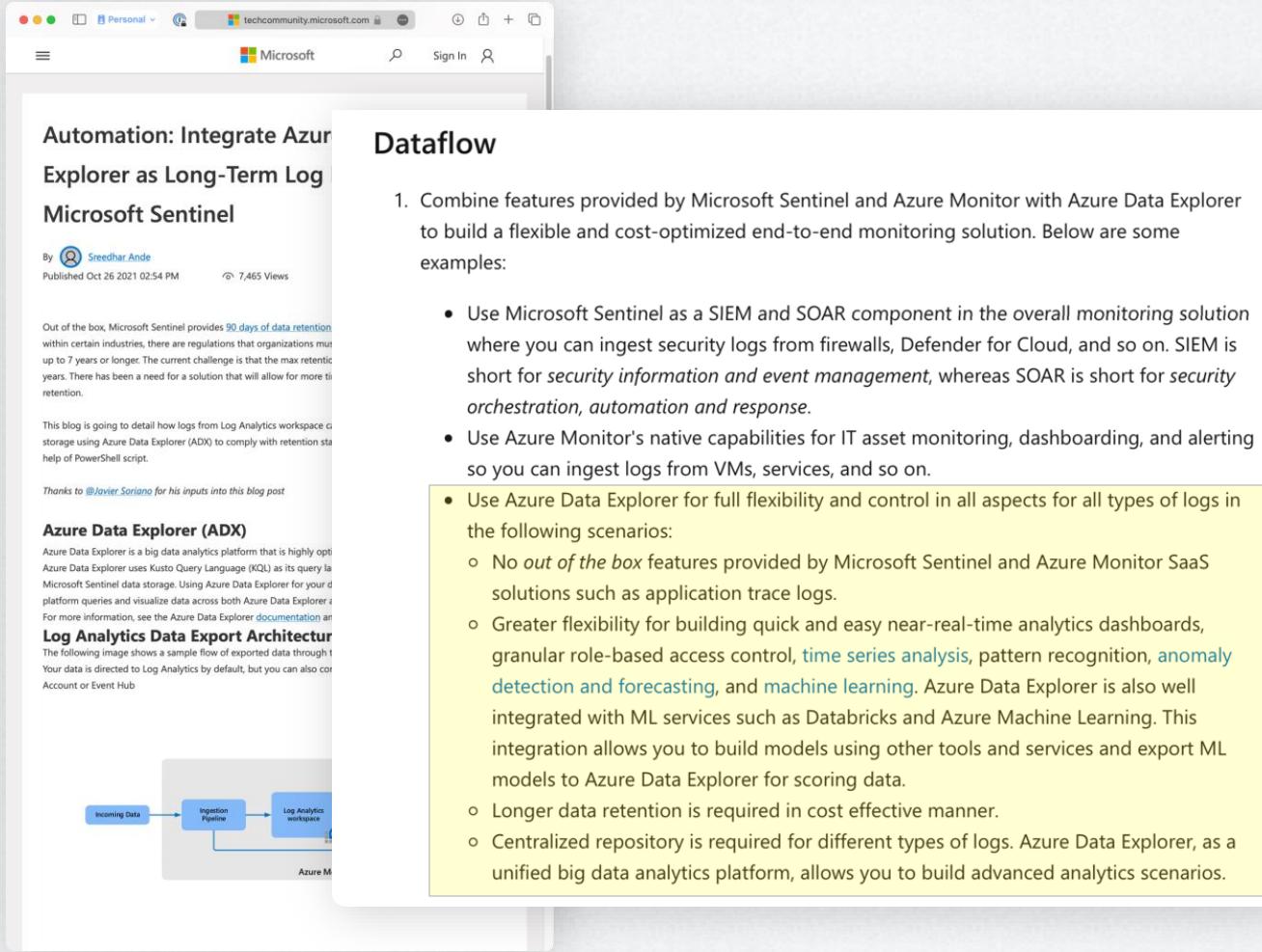
A screenshot of a Microsoft Edge browser window showing a detailed architecture diagram for Log Analytics Data Export. The diagram illustrates the flow of data from various source systems (Security Log, System Log, Network Log, Application Log, Server DB, user activity log) through the SIEM and SOAR SaaS solution (Azure Monitor) and the Logging and Monitoring SaaS solution (Azure Monitor). The data then flows into the Azure Data Lake Storage, which is connected to Azure Data Explorer via a Flexible Path solution. Azure Data Explorer is shown as a central hub, interfacing with various analytical platforms like Log Analytics, Application Insights, and Power BI. It also connects to Log Apps, Workbooks, and the Azure Data Explorer UI. The diagram includes a note about querying across multiple data sources using the Azure Data Explorer UI. A link to download a Visio file is provided.

Download a [Visio file](#) of this architecture.

Dataflow

- Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:
 - Use Microsoft Sentinel as a SIEM and SOAR component in the overall monitoring solution where you can ingest security logs from firewalls, Defender for Cloud, and so on. SIEM is short for *security information and event management*, whereas SOAR is short for *security orchestration, automation and response*.
 - Use Azure Monitor's native capabilities for IT asset monitoring, dashboarding, and alerting so you can ingest logs from VMs, services, and so on.
 - Use Azure Data Explorer for full flexibility and control in all aspects for all types of logs in the following scenarios:
 - No *out of the box* features provided by Microsoft Sentinel and Azure Monitor SaaS solutions such as application trace logs.
 - Greater flexibility for building quick and easy near-real-time analytics dashboards, granular role-based access control, time series analysis, pattern recognition, anomaly detection and forecasting, and machine learning. Azure Data Explorer is also well integrated with ML services such as Databricks and Azure Machine Learning. This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.
 - Longer data retention is required in a cost effective manner.
 - Centralized repository is required for different types of logs. Azure Data Explorer, as a unified big data analytics platform, allows you to build advanced analytics scenarios.

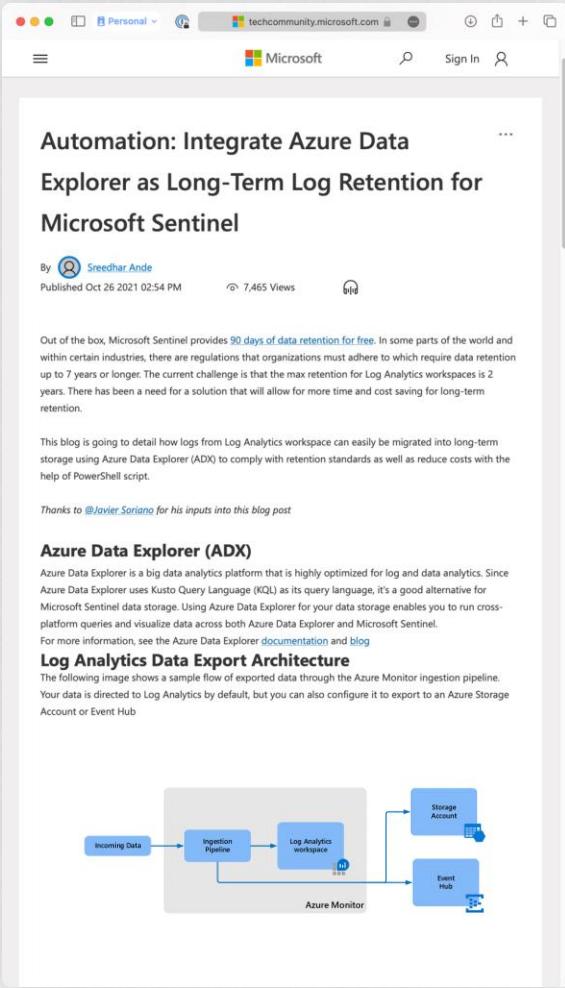




Dataflow

1. Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:
 - Use Microsoft Sentinel as a SIEM and SOAR component in the overall monitoring solution where you can ingest security logs from firewalls, Defender for Cloud, and so on. SIEM is short for *security information and event management*, whereas SOAR is short for *security orchestration, automation and response*.
 - Use Azure Monitor's native capabilities for IT asset monitoring, dashboarding, and alerting so you can ingest logs from VMs, services, and so on.
 - Use Azure Data Explorer for full flexibility and control in all aspects for all types of logs in the following scenarios:
 - No *out of the box* features provided by Microsoft Sentinel and Azure Monitor SaaS solutions such as application trace logs.
 - Greater flexibility for building quick and easy near-real-time analytics dashboards, granular role-based access control, *time series analysis*, pattern recognition, *anomaly detection and forecasting*, and *machine learning*. Azure Data Explorer is also well integrated with ML services such as Databricks and Azure Machine Learning. This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.
 - Longer data retention is required in cost effective manner.
 - Centralized repository is required for different types of logs. Azure Data Explorer, as a unified big data analytics platform, allows you to build advanced analytics scenarios.



A screenshot of a Microsoft Edge browser window displaying a blog post titled "Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel". The post is authored by Sreedhar Ande and published on October 26, 2021, at 02:54 PM, with 7,465 views. The content discusses the challenges of long-term log retention in Microsoft Sentinel and how to migrate logs from Log Analytics to Azure Data Explorer.

Out of the box, Microsoft Sentinel provides 90 days of data retention for free. In some parts of the world and within certain industries, there are regulations that organizations must adhere to which require data retention up to 7 years or longer. The current challenge is that the max retention for Log Analytics workspaces is 2 years. There has been a need for a solution that will allow for more time and cost saving for long-term retention.

This blog is going to detail how logs from Log Analytics workspace can easily be migrated into long-term storage using Azure Data Explorer (ADX) to comply with retention standards as well as reduce costs with the help of PowerShell script.

Thanks to [@Javier Soriano](#) for his inputs into this blog post

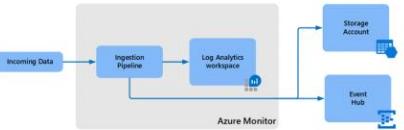
Azure Data Explorer (ADX)

Azure Data Explorer is a big data analytics platform that is highly optimized for log and data analytics. Since Azure Data Explorer uses Kusto Query Language (KQL) as its query language, it's a good alternative for Microsoft Sentinel data storage. Using Azure Data Explorer for your data storage enables you to run cross-platform queries and visualize data across both Azure Data Explorer and Microsoft Sentinel.

For more information, see the [Azure Data Explorer documentation](#) and [blog](#).

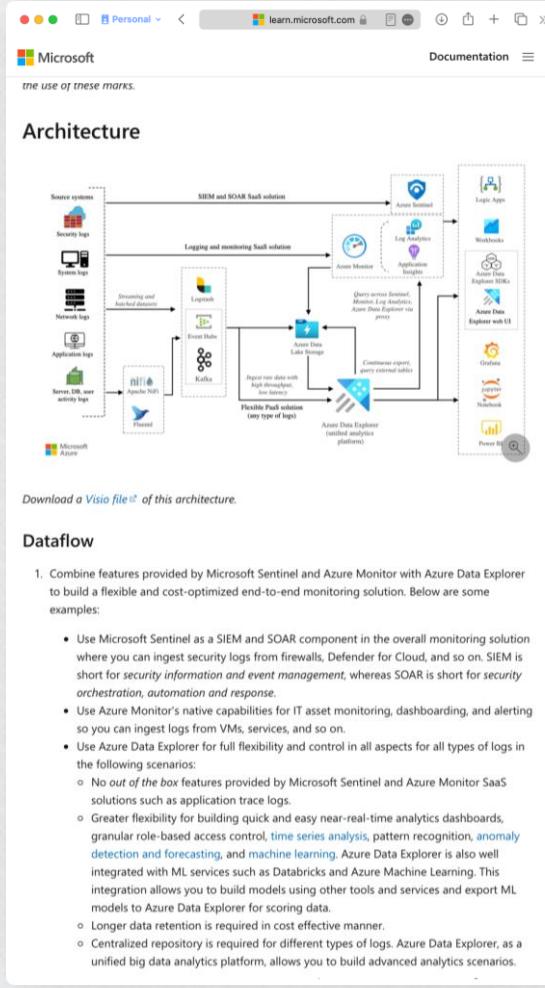
Log Analytics Data Export Architecture

The following image shows a sample flow of exported data through the Azure Monitor ingestion pipeline. Your data is directed to Log Analytics by default, but you can also configure it to export to an Azure Storage Account or Event Hub.



```

graph LR
    ID[Incoming Data] --> IP[Ingestion Pipeline]
    IP --> LA[Log Analytics workspace]
    IP --> SA[Storage Account]
    IP --> EH[Event Hub]
    LA --> SA
    LA --> EH
  
```

A screenshot of a Microsoft Edge browser window showing the architecture of the SIEM and SOAR Stack solution. The diagram illustrates the flow of data from various source systems (Security logs, System logs, Network logs, Application logs, Server DB, user activity logs) through the Logging and Monitoring Stack solution (using Logstash, Event Hubs, Kafka, nifi, Apache NIFI, Fluentd) into the SIEM and SOAR Stack solution (using Azure Monitor). This solution then integrates with Azure Data Explorer via a Log Analytics workspace, which provides features like continuous export, low latency, and flexible push solutions. The architecture also includes integration with Log Analytics, Application Insights, and other Microsoft services like Logic Apps, Workbooks, and Power BI. A note indicates that "Query across Immortal, Managed and Federated Azure Data Explorer via proxy".

Download a [Visio file](#) of this architecture.

Dataflow

- Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:
 - Use Microsoft Sentinel as a SIEM and SOAR component in the overall monitoring solution where you can ingest security logs from firewalls, Defender for Cloud, and so on. SIEM is short for *security information and event management*, whereas SOAR is short for *security orchestration, automation and response*.
 - Use Azure Monitor's native capabilities for IT asset monitoring, dashboarding, and alerting so you can ingest logs from VMs, services, and so on.
 - Use Azure Data Explorer for full flexibility and control in all aspects for all types of logs in the following scenarios:
 - No *out of the box* features provided by Microsoft Sentinel and Azure Monitor SaaS solutions such as application trace logs.
 - Greater flexibility for building quick and easy near-real-time analytics dashboards, granular role-based access control, *time series analysis*, pattern recognition, *anomaly detection and forecasting*, and *machine learning*. Azure Data Explorer is also well integrated with ML services such as Databricks and Azure Machine Learning. This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.
 - Longer data retention is required in a cost effective manner.
 - Centralized repository is required for different types of logs. Azure Data Explorer, as a unified big data analytics platform, allows you to build advanced analytics scenarios.



Automation: Integrate Azure Data Explorer as Long-Term Log Retention for Microsoft Sentinel

By Sreedhar Ande
Published Oct 26 2021 02:54 PM
7,465 Views

Out of the box, Microsoft Sentinel provides 90 days of data retention for free. In some parts of the world and within certain industries, there are regulations that organizations must adhere to which require data retention up to 7 years or longer. The current challenge is that the max retention for Log Analytics workspaces is 2 years. There has been a need for a solution that will allow for more time and cost saving for long-term retention.

This blog is going to detail how logs from Log Analytics workspace can easily be migrated into long-term storage using Azure Data Explorer (ADX) to comply with retention standards as well as reduce costs with the help of PowerShell script.

Thanks to [@Javier Soriano](#) for his inputs into this blog post

Azure Data Explorer (ADX)

Azure Data Explorer is a big data analytics platform that is highly optimized for log and data analytics. Since Azure Data Explorer uses Kusto Query Language (KQL) as its query language, it's a good alternative for Microsoft Sentinel data storage. Using Azure Data Explorer for your data storage enables you to run cross-platform queries and visualize data across both Azure Data Explorer and Microsoft Sentinel.

For more information, see the Azure Data Explorer [documentation](#) and [blog](#).

Log Analytics Data Export Architecture

The following image shows a sample flow of exported data through the Azure Monitor ingestion pipeline. Your data is directed to Log Analytics by default, but you can also configure it to export to an Azure Storage Account or Event Hub.

Architecture

Download a [Visio file](#) of this architecture.

Dataflow

- Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:
 - Use Microsoft Sentinel as a SIEM and SOAR component in the overall monitoring solution where you can ingest security logs from firewalls, Defender for Cloud, and so on. SIEM is short for *security information and event management*, whereas SOAR is short for *security orchestration, automation and response*.
 - Use Azure Monitor's native capabilities for IT asset monitoring, dashboarding, and alerting so you can ingest logs from VMs, services, and so on.
 - Use Azure Data Explorer for full flexibility and control in all aspects for all types of logs in the following scenarios:
 - No out of the box* features provided by Microsoft Sentinel and Azure Monitor SaaS solutions such as application trace logs.
 - Greater flexibility for building quick and easy near-real-time analytics dashboards, granular role-based access control, *time series analysis*, pattern recognition, *anomaly detection and forecasting*, and *machine learning*. Azure Data Explorer is also well integrated with ML services such as Databricks and Azure Machine Learning. This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.
 - Longer data retention is required in cost effective manner.
 - Centralized repository is required for different types of logs. Azure Data Explorer, as a unified big data analytics platform, allows you to build advanced analytics scenarios.

What's New: Azure Sentinel Hunting supports ADX cross-resource queries

By Ben.Nick
Published Jul 14 2021 12:00 PM
7,744 Views

Now in preview, you can use Azure Data Explorer (ADX) [cross-resource queries](#) from within the hunting query page, the live stream page, and the logs (Log Analytics) page. Although Log Analytics remains the primary data storage location for performing analysis with Azure Sentinel, there are cases where ADX is required to store data due to cost, retention periods, or other factors.

You can learn more about sending logs from Azure Sentinel to Azure Data Explorer for long-term retention here: [Integrate Azure Data Explorer for long-term log retention](#)

Creating cross-resource queries

To query data stored in ADX clusters, simply use the `adx()` function to specify the ADX cluster, database name, and desired table. You can then query the output as you would any other table. If you have access to an ADX cluster with active data, it is super easy to try:

```
Here is a brief summary of the adx() function syntax to help get you started:  
adx("<Cluster URI>/<Database Name>.<Table Name>")
```

Here is an example query that accesses public data:
`adx("https://help.kusto.windows.net/Samples").StormEvents | take 5`

You can find the full details here: [Cross-query your Log Analytics or Application Insights resources and Azure Data Explorer](#)

Using cross-resource queries on the hunting queries, live stream, and logs pages

Once you know how to construct cross-reference queries, using them in the hunting experience is easy. Go to the hunting queries page and click "+ New query" to create a new custom query. Add your cross-resource query to the "Custom Query" field as you would for any other hunting query.

Log ingestion challenges

- Cloud is always changing
 - MMA → AMA
 - Lots of specific features (still) in preview
- Azure Arc is not to be underestimated
- Co-operate with other ops teams and vendors
 - Managing DCRs
- Sentinel as a “security data lake”?
 - Splitting and filtering logs into different destinations
- Virtual Machine vs Container
- Microsoft vs third-party option





Logstash



Logstash

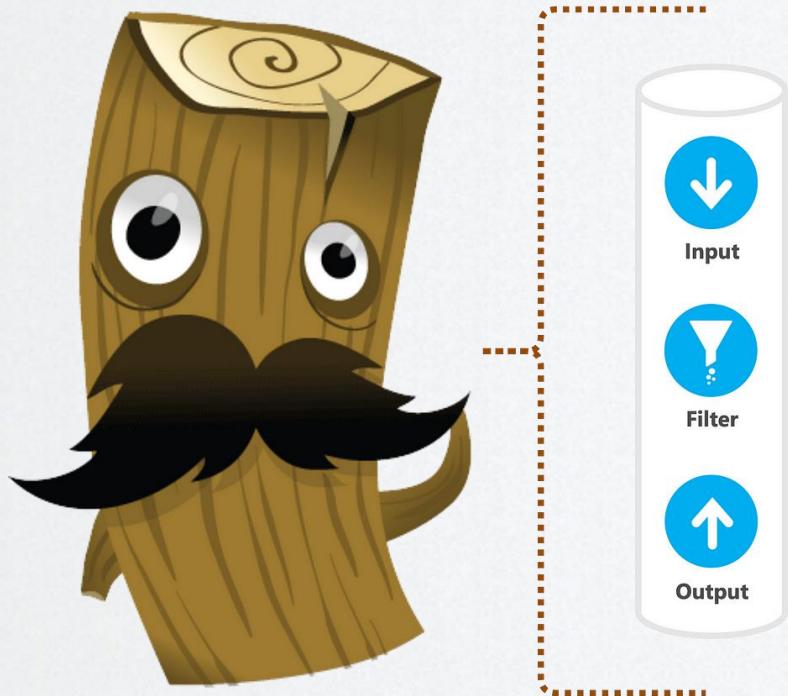
- “Swiss army knife” 
- VERY flexible!
 - Plethora of input and output plug-ins available
 - Syslog, files, shares, databases
 - Log Analytics, Azure Data Explorer, Event Hub, Blob
 - On-the-fly parsing and filtering
 - Run within a container (ACI, AKS, ...)



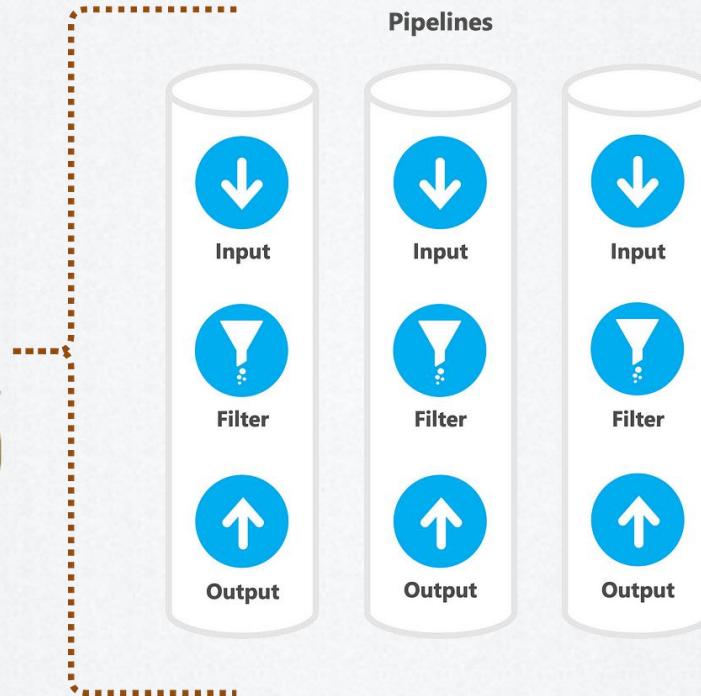
Logstash



Logstash

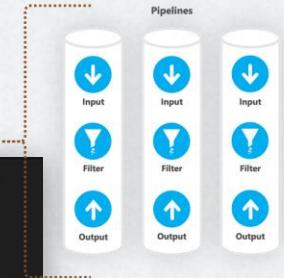


Logstash



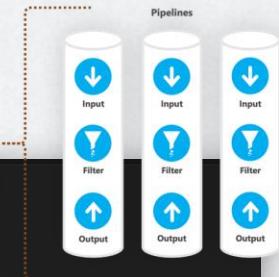
Logstash

```
1 input {
2     jdbc {
3         # Postgres jdbc connection string to our database, mydb
4         jdbc_connection_string => "jdbc:postgresql://localhost:5432/security"
5         # The user we wish to execute our statement as
6         jdbc_user => "dbuser"
7         # The user we wish to execute our statement as
8         jdbc_password => "{$dbpassword}"
9         # The path to our downloaded jdbc driver
10        jdbc_driver_library => "/usr/share/logstash/modules/postgres/postgresql-42.2.12.jar"
11        # The name of the driver class for Postgresql
12        jdbc_driver_class => "org.postgresql.Driver"
13        # postgres query with column tracking to avoid importing duplicates
14        statement => "SELECT uid, timestamp, event, message, username, ipaddress FROM authentications WHERE uid > :sql_last_value"
15        use_column_value => true
16        tracking_column => "uid"
17        # schedule to run every 5 minutes
18        schedule => "*/* * * * *"
19    }
20 }
21
22 output {
23     microsoft-logstash-output-azure-loganalytics {
24         workspace_id => "${SENTINEL_ID}"
25         workspace_key => "${SENTINEL_KEY}"
26         log_type => "postgreSQLauthentications"
27         key_names => ['timestamp','event','message','username','ipaddress']
28         flush_items => 10
29         flush_interval_time => 5
30     }
31     # for debug
32     stdout { codec => rubydebug }
33 }
```



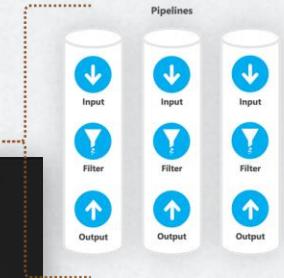
Logstash

```
1 input {  
2     jdbc {  
3         # Postgres jdbc connection string to our database, mydb  
4         jdbc_connection_string => "jdbc:postgresql://localhost:5432/security"  
5         # The user we wish to execute our statement as  
6         jdbc_user => "dbuser"  
7         # The user we wish to execute our statement as  
8         jdbc_password => "{$dbpassword}"  
9         # The path to our downloaded jdbc driver  
10        jdbc_driver_library => "/usr/share/logstash/modules/postgres/postgresql-42.2.12.jar"  
11        # The name of the driver class for Postgresql  
12        jdbc_driver_class => "org.postgresql.Driver"  
13        # postgres query with column tracking to avoid importing duplicates  
14        statement => "SELECT uid, timestamp, event, message, username, ipaddress FROM authentications WHERE uid  
15        use_column_value => true  
16        tracking_column => "uid"  
17        # schedule to run every 5 minutes  
18        schedule => "*/5 * * * *"  
19    }  
20 }
```



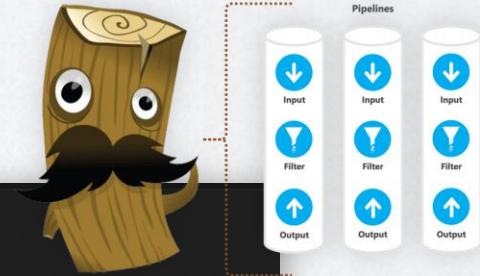
Logstash

```
1 input {
2     jdbc {
3         # Postgres jdbc connection string to our database, mydb
4         jdbc_connection_string => "jdbc:postgresql://localhost:5432/security"
5         # The user we wish to execute our statement as
6         jdbc_user => "dbuser"
7         # The user we wish to execute our statement as
8         jdbc_password => "{$dbpassword}"
9         # The path to our downloaded jdbc driver
10        jdbc_driver_library => "/usr/share/logstash/modules/postgres/postgresql-42.2.12.jar"
11        # The name of the driver class for Postgresql
12        jdbc_driver_class => "org.postgresql.Driver"
13        # postgres query with column tracking to avoid importing duplicates
14        statement => "SELECT uid, timestamp, event, message, username, ipaddress FROM authentications WHERE uid > :sql_last_value"
15        use_column_value => true
16        tracking_column => "uid"
17        # schedule to run every 5 minutes
18        schedule => "*/* * * * *"
19    }
20 }
21
22 output {
23     microsoft-logstash-output-azure-loganalytics {
24         workspace_id => "${SENTINEL_ID}"
25         workspace_key => "${SENTINEL_KEY}"
26         log_type => "postgreSQLauthentications"
27         key_names => ['timestamp','event','message','username','ipaddress']
28         flush_items => 10
29         flush_interval_time => 5
30     }
31     # for debug
32     stdout { codec => rubydebug }
33 }
```



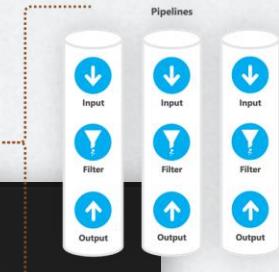
Logstash

```
1 input {  
2     syslog {  
3         port => 5514  
4     }  
5 }  
6  
7 output {  
8     microsoft-logstash-output-azure-loganalytics {  
9         workspace_id => "${SENTINEL_ID}"  
10        workspace_key => "${SENTINEL_KEY}"  
11        proxy => "<http://proxy.domain.com:8080>"  
12        retransmission_time => 2  
13        plugin_flush_interval => 2  
14        custom_log_table_name => "${TABLE_NAME}"  
15    }  
16 }
```

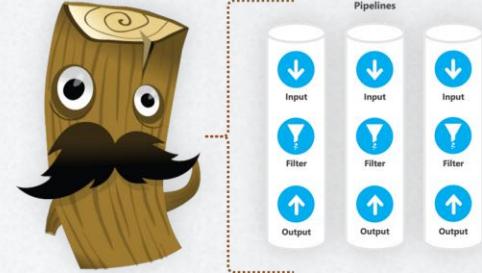


Logstash

```
1 input {  
2     syslog {  
3         port => 5514  
4     }  
5 }  
6  
7 output {  
8     microsoft-sentinel-logstash-output-plugin {  
9         client_app_Id => "${APP_ID}"  
10        client_app_secret => "${APP_SECRET}"  
11        tenant_id => "${TENANT_ID}"  
12        proxy => "http://proxy.domain.com:8080"  
13        data_collection_endpoint => "https://\${DCE\_URI}.ingest.monitor.azure.com"  
14        dcr_immutable_id => "dcr-${IMMUTABLE_ID}"  
15        dcr_stream_name => "Custom-${TABLE_NAME}_CL"  
16    }  
17 }
```

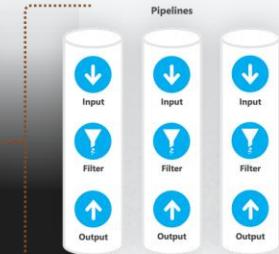


Logstash



```
1  input {
2    tcp {
3      port => 514
4      type => syslog
5      host => "141.93.182.143"
6      tags => ["gso_sentinel"]
7    }
8  }
9  input {
10   udp {
11     port => 514
12     type => syslog
13     host => "141.93.182.143"
14     queue_size => 16384
15     workers => 4
16     receive_buffer_bytes => 24576000
17     codec => plain { charset => "ISO-8859-1" }
18     tags => ["gso_sentinel"]
19   }
20 }
21 input {
22   tcp {
23     port => 7514
24     type => syslog
25     mode => "server"
26     host => "141.93.182.143"
27     ssl_enable => true
28     ssl_cert => "/apps1/log-jars/KPNCert/syslogserver.pem"
```

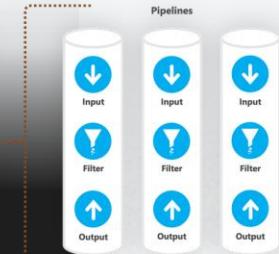
```
quorum_size => 16384
workers => 4
receive_buffer_bytes => 24576000
17   codecs {
18     main { charset => "ISO-8859-1" }
19   }
20 }
21 input {
22   tcp {
23     port => 7514
24     type => syslog
25     mode => "server"
26     host => "141.93.182.143"
27     ssl_enable => true
28     ssl_cert => "/apps1/log-jars/KPNCert/syslogserver.pem"
29     ssl_key => "/apps1/log-jars/KPNCert/syslogserverkey.pem"
30     ssl_certificateAuthorities => [ "/apps1/log-jars/KPNCert/RootCA.pem" ]
31     ssl_verify => true
32     ssl_key_passphrase => "${CERT_PASS}"
33     ssl_supported_protocols => ['TLSv1.1', 'TLSv1.2', 'TLSv1.3']
34     tags => ["gso_sentinel"]
35   }
36 }
37
38 filter {
39   if ("gso_sentinel" in [tags]){
40     mutate {
41       add_field => { "Country" => "NL" }
42       add_field => { "connector_host" => "hostname.domain.com" }
43     }
44     if ([host] = "17.93.180.49") or ([host] = "17.93.126.47") {
45       mutate {
46         add_field => { "sentinelTable" => "UX_rhlinux" }
47       }
48     }
49     else if ([host] = "17.93.177.62") or ([host] = "17.93.178.219") or ([host] = "17.93.178.79") or ([host] = "17.93.126.92")
50     or ([host] = "17.93.125.114") or ([host] = "17.93.125.126") or ([host] = "17.93.126.40") or ([host] = "17.93.178.204") or
51     or ([host] = "17.93.180.43") or ([host] = "17.93.125.149") or ([host] = "17.93.125.124") or ([host] = "17.93.179.215") or
52     or ([host] = "17.93.188.100") or ([host] = "17.93.178.31") {
53       mutate {
54         add_field => { "sentinelTable" => "UX_aix" }
```



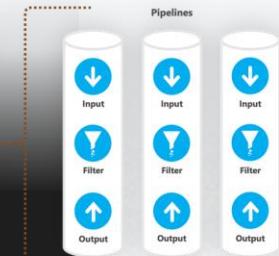
```

67
68     else if ([host] == "10.16.128.30") {
69         mutate {
70             add_field => { "sentinelTable" => "AC_Aruba" }
71         }
72     } else {
73         mutate {
74             add_field => { "sentinelTable" => "New_SourcesAlert" }
75         }
76     }
77     grok {
78         keep_empty_captures => "true"
79         match => { "@timestamp" => "%{YEAR:sys_year}-%{MONTHNUM:sys_month}-%{MONTHDAY:sys_day}%{GREEDYDATA}" }
80     }
81 }
82
83
84 output {
85     if("gso_sentinel" in [tags]){
86         if([sentinelTable] == "UX_rhelinux") {
87             microsoft-logstash-output-azure-loganalytics {
88                 workspace_id => "${SENTINEL_ID}"
89                 workspace_key => "${SENTINEL_KEY}"
90                 proxy => "http://proxy.domain.com:8080"
91                 retransmission_time => 2
92                 plugin_flush_interval => 2
93                 custom_log_table_name => "UX_rhelinux"
94             }
95         }
96     }
97     else if([sentinelTable] == "UX_aix") {
98         microsoft-logstash-output-azure-loganalytics {
99             workspace_id => "${SENTINEL_ID}"
100            workspace_key => "${SENTINEL_KEY}"
101            proxy => "http://proxy.domain.com:8080"
102            retransmission_time => 2
103            plugin_flush_interval => 2
104            custom_log_table_name => "UX_aix"
105        }
106    }

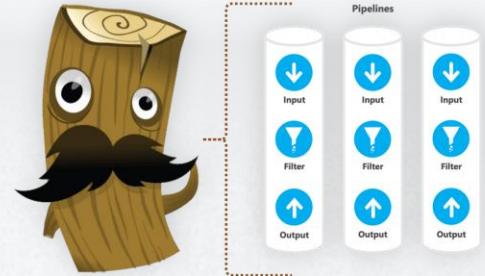
```



```
100     client_app_id => "<client_app_id>"  
101     tenant_id => "<tenant_id>"  
102     data_collection_endpoint => "https://dce-uri.westeurope-1.ingest.monitor.azure.com"  
103     dcr_immutable_id => "dcr-<id>"  
104     dcr_stream_name => "Custom-UX_Hpunix"  
105   }  
106 }  
107 else if([sentinelTable] == "UX_Solaris") {  
108   kusto {  
109     path => "/tmp/kusto/%{+YYYY-MM-dd-HH-mm-ss}.txt"  
110     ingest_url => "https://ingest-clustername.westeurope.kusto.windows.net"  
111     app_id => "<client_app_Id>"  
112     app_key=> "<client_app_secret>"  
113     app_tenant => "<tenant_id>"  
114     database => "logstash-archive"  
115     table => "Cisco_RO_CL"  
116     json_mapping => "UX_Solaris"  
117   }  
118 }  
119 else if([sentinelTable] == "AC_Aruba") {  
120   microsoft-sentinel-logstash-output-plugin {  
121     client_app_Id => "<client_app_Id>"  
122     client_app_secret => "<client_app_secret>"  
123     tenant_id => "<tenant_id>"  
124     data_collection_endpoint => "https://dce-uri.westeurope-1.ingest.monitor.a_r_u_com"  
125     dcr_immutable_id => "dcr-<id>"  
126     dcr_stream_name => "Custom-AC_Aruba"  
127   }  
128 }  
129 else if([sentinelTable] == "New_SourcesAlert") {  
130   microsoft-logstash-output-azure-loganalytics {  
131     workspace_id => "${SENTINEL_ID}"  
132     workspace_key => "${SENTINEL_KEY}"  
133     proxy => "http://proxy.domain.com:8080"  
134     retransmission_time => 2  
135     plugin_flush_interval => 2  
136     custom_log_table_name => "New_SourcesAlert"  
137   }  
138 }  
139 }  
140 }
```



Logstash





Managing keys



Managing keys

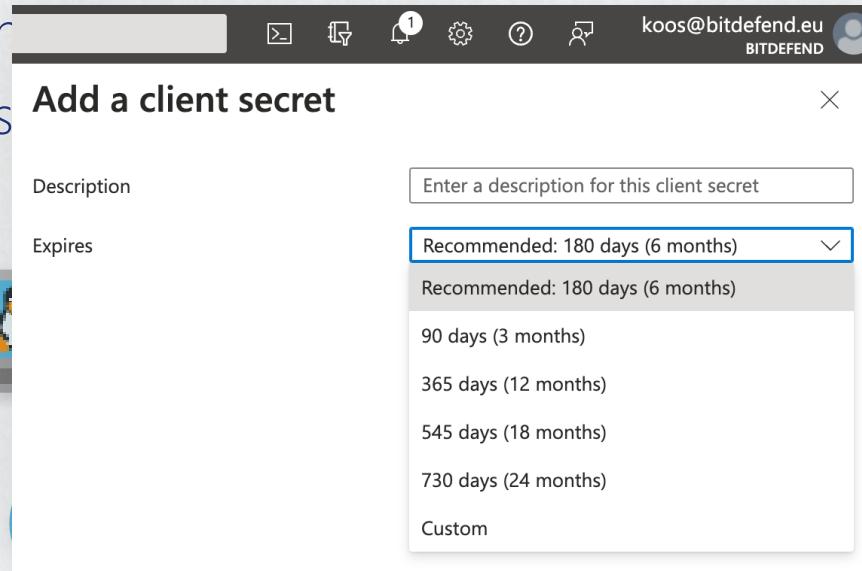
- (Enterprise) Applications secrets

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation bar includes Home, BitDefend | App registrations, vm-logstash-01, Overview, Quickstart, Integration assistant, Manage (selected), Branding & properties, Authentication, Certificates & secrets (selected), Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest, Support + Troubleshooting, Troubleshooting, and New support request. The main content area displays the 'Certificates & secrets' blade for the application 'vm-logstash-01'. It shows a table with one row under 'Client secrets (0)' and a note: 'No client secrets have been created for this application yet.' A modal dialog titled 'Add a client secret' is open, prompting for a 'Description' (with a placeholder 'Enter a description for this client secret') and an 'Expires' date. A dropdown menu lists several options: Recommended: 180 days (6 months) (selected), Recommended: 180 days (6 months), 90 days (3 months), 365 days (12 months), 545 days (18 months), 730 days (24 months), and Custom.



Managing keys

- (Enterprise) Applications secrets
- Change password
- Use client secret



Managing keys

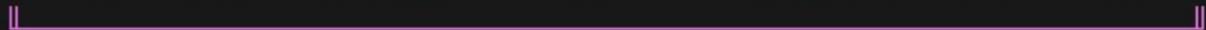
- (Enterprise) Applications secrets
- Check for older secrets which may never expire
- Use Managed-Identities if possible
- Implement key-rotation mechanism





L G S T R A S H

```
::::::: ::::::: ::::::: ::::::: :::::::  
:+: :+: :+: :+: :+: :+: :+: :+:  
:+: +:+ :+:+ :+:+ :+:+ :+:+ :+:+ :+:  
+#++:++#: +#+ :+:+ :#++ :#++:++# :#++:++#:  
+#+ +#+ +#+ +#+ +#+ +#+ +#+ +#+ +#+  
#+# #+# #+# #+# #+# #+# #+# #+#  
### ### ##### ##### ##### ##### #####
```



Logstash r0t8r

- Sequel to Azure DevOps "r0t8r"



Logstash r0t8r

- Sequel to Azure DevOps "r0t8r"

Setting up the pipeline

A copy of all source code listed below can be found on [my Github page here](#).

```
# This pipeline performs key rotation for the Azure AD Application of a specific Service Connection.
# The schedule in this example is set to run every day at 05:00
trigger: none
schedules:
- cron: "0 * * * *"
  displayName: Daily Service Connection key rotation
branches:
- main:
  - master
  - always: true
variables:
  keyrotationscript: "$(System.DefaultWorkingDirectory)/scripts/Set-AzureDevopsKeyRotation.ps1"
  serviceConnectionNameLUTS: "LUTS"
pool:
  - vmImage: 'ubuntu-latest'
stages:
- stage: service_connection_key_rotation
  displayName: Azure DevOps Service Connection key rotation
  jobs:
    - job: LUTS
      displayName: Service Connection DevOps LUTS
      steps:
        - template: templates/template-key-rotation.yml
          parameters:
            serviceConnection_name: ${{ variables.serviceConnectionNameLUTS }}
            keyrotation_script: ${{ variables.keyrotationscript }}
```

Note: the only changes you should make is the serviceConnectionName variable.



Koos Goossens • You

Microsoft Security MVP | Azure | Sentinel | Defender 365 | DevOps
2yr • Edited •

Me and my colleague [Dennis van den Akker](#) wrote a blog about implementing automatic key rotation for your ARM service connections in Azure DevOps.

💡 Stop not letting your app registration secrets expire!

🤖 Automate as much as you can and embrace a DevOps mindset. This leads to a higher security posture and less shadow IT.

<https://lnkd.in/gFQvaet>



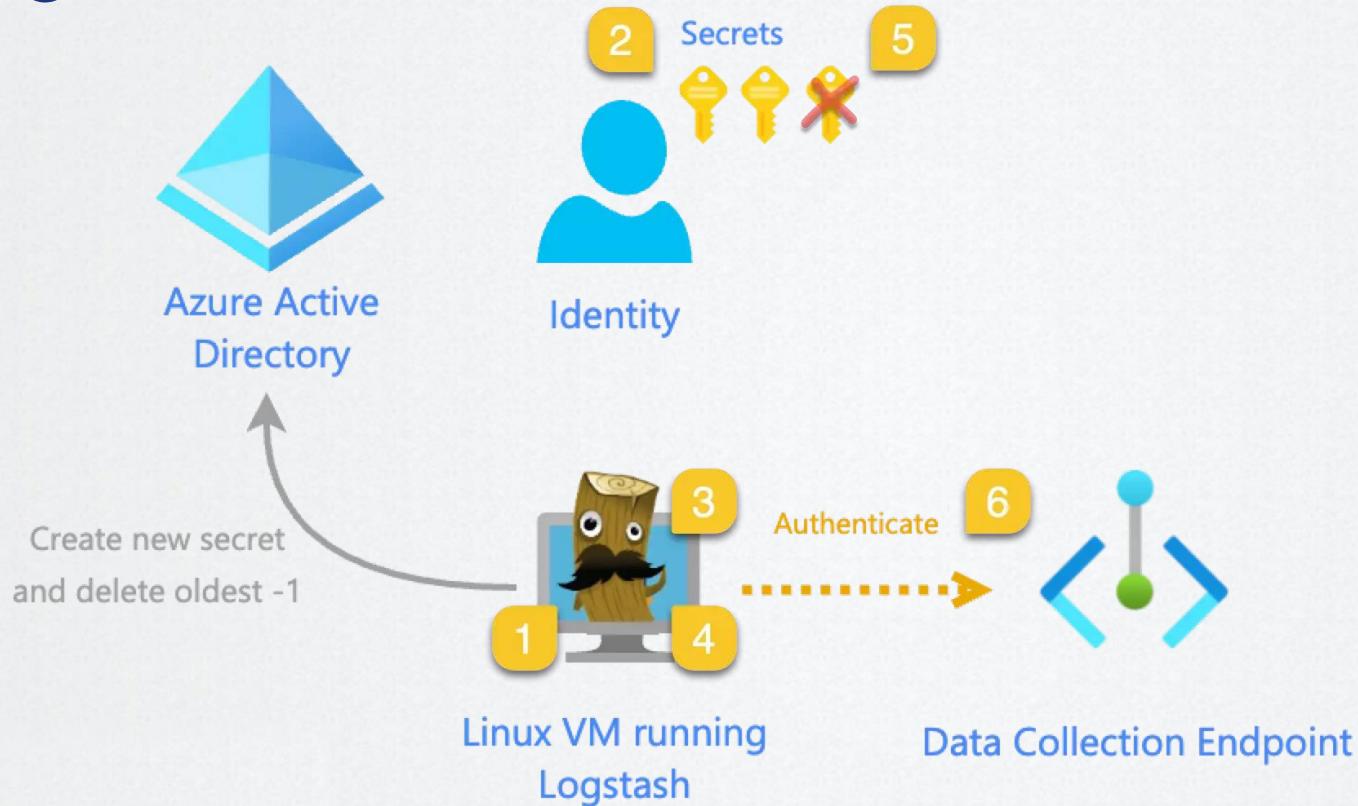
Logstash r0t8r

- Sequel to Azure DevOps “r0t8r”
- We can run PowerShell on Linux too!

Microsoft ❤️ Linux



Logstash r0t8r



Logstash r0t8r

- PowerShell Credentials
 - appId.CRED file containing System.Security.SecureString
 - Avoids storing sensitive strings in memory as plain text
 - The value is automatically protected when the instance is initialized or when the value is modified.



Logstash r0t8r

- One more thing: minor preparations
 - Application needs to be owner of itself
 - Application has Application.ReadWrite.OwnedBy Microsoft.Graph permissions
- Use my script Add-AppOwner.ps1 (*one time only*)



Logstash r0t8r

- One more thing about Logstash preparations



Logstash r0t8r



Logstash r0t8r

- How to run r0t8r?

```
sudo pwsh \  
    -file logstash-rot8r.ps1 \  
    -tenantId <tenantId> \  
    -applicationId <appId> \  
    -logstashConfigLocation /etc/../example.conf
```



Logstash r0t8r

- Schedule with cronjob

```
sudo crontab -e
```

```
# ┌───────── minute [ 0 – 59 ]
# | ┌────── hour [ 0 – 23 ]
# | | ┌─── day of the month [ 1 – 31 ]
# | | | ┌── month [ 1 – 12 ]
# | | | | ┌── day of the week [ 0 – 6 ] (Sunday – Saturday)
# | | | |
# | | |
# | |
# |
# |
# * * * * * command or script
```



Logstash r0t8r

- Schedule with crontab

```
sudo crontab -e
```



Logstash r0t8r

The screenshot shows a macOS desktop environment. On the left, a terminal window is open with the command `sudo` entered, with the password field partially visible. To the right of the terminal is a Safari browser window displaying the Microsoft Azure portal. The URL in the address bar is `portal.azure.com`. The page shown is for a service named `vm-logstash-01`, specifically the `Certificates & secrets` section. The sidebar on the left of the Azure page lists various management options like Overview, Quickstart, Integration assistant, Manage, Branding & properties, Authentication, Certificates & secrets (which is selected), Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, and Manifest. The main content area shows a table for Client secrets. One entry is listed:

Description	Expires	Value	Secret ID
Initial-secret	5/30/2023	x6l&Q==--cxWl2vR5Req4SpGtB19V3OIK... (redacted)	e41229b5-ecb0-4236-b299-d923535c6f8e

Logstash r0t8r

- Using Logstash Keystore 

```
client_app_secret => "pg%lecateMnsr&@p"
```



Logstash r0t8r

- Using Logstash Keystore 

```
client_app_secret => "${app_secret}"
```



Logstash r0t8r

- Using Logstash Keystore 



logstash.keystore

```
client_app_secret => "${app_secret}"
```

Logstash r0t8r

- Using Logstash Keystore 



logstash.keystore



LOGSTASH_KEYSTORE_PASS

```
client_app_secret => "${app_secret}"
```

Logstash r0t8r

- Using Logstash Keystore 



LOGSTASH_KEYSTORE_PASS



logstash.keystore

client_app_secret => "\${app_secret}"



Logstash r0t8r

- Using Logstash Keystore 



LOGSTASH_KEYSTORE_PASS



logstash.keystore



client_app_secret => "pg%lecateMnsr&@p"

Logstash r0t8r

- Using Logstash to...



client_app_secret => "pg%lecateMnsr&@p"

Logstash r0t8r

```
bitdefend@vm-logstash-01: ~ (ssh)
koos > scripts > main > ssh bitdefend@logstash01-bitdefend.westeurope.cloudapp.azure.com
bitdefend@vm-logstash-01: ~ (ssh)
in pwsh at 01:07:19

LOGSTASH

root@vm-logstash-01:
OS: Ubuntu 22.04.2 LTS x86_64
Host: Virtual Machine Hyper-V UEFI Release v4.1
Kernel: 5.15.0-1038-azure
Uptime: 2 hours, 16 mins
Packages: 737 (dpkg), 6 (snap)
Shell: bash 5.1.16
Resolution: 1024x768
Terminal: run-parts
CPU: Intel Xeon Platinum 8272CL (2) @ 2.593GHz
Memory: 1305MiB / 3925MiB

Expanded Security Maintenance for Applications is enabled.

0 updates can be applied immediately.

Last login: Mon May 22 23:07:05 2023 from 85.145.24.60
bitdefend@vm-logstash-01:~$
```

References

The image shows two side-by-side screenshots of Microsoft Learn articles.

Left Article: Install PowerShell on Linux

- Header: Microsoft | Learn Documentation Training Certifications Q&A Code Samples Assessments Shows Events
- Section: Version (PowerShell 7.3)
- Section: How to use this documentation (Overview, Install, Overview, Installing PowerShell on Windows, Overview, Installing on Alpine, Installing on Debian, Installing on Raspberry Pi OS, Installing on RHEL, Installing on Ubuntu, Community support for Linux, Alternate ways to install on Linux, Installing PowerShell on macOS, Installing PowerShell on Arm, Using PowerShell in Docker, Microsoft Update FAQ for PowerShell)
- Section: In this article (Alpine, Debian, Red Hat Enterprise Linux (RHEL), Ubuntu, Raspberry Pi OS)
- Content: PowerShell can be installed on different Linux distributions. Most Linux platforms and distributions include a package manager that makes it easy to install PowerShell. This article provides instructions for installing PowerShell on various Linux distributions.

Right Article: Migrate to Azure Monitor Agent from Log Analytics agent

- Header: Microsoft | Learn Documentation Training Certifications Q&A Code Samples Assessments Shows Events
- Section: Filter by title (Azure Monitor Documentation, Overview, Concepts, Tutorials, Samples, Data sources, Data collection, Azure Monitor Agent, Overview, Install Azure Monitor Agent, Configure data collection, Define network settings, Migrate from Log Analytics Agent, Migration guidance, Migration tools, Migration custom log tables, Performance, FAQ, Legacy agents)
- Section: In this article (Benefits, Migration guidance, Migrate additional services and features, Next steps)
- Content: Azure Monitor Agent (AMA) replaces the Log Analytics agent (also known as MMA and OMS) for Windows and Linux machines, in Azure and non-Azure environments, including on-premises and third-party clouds. The agent introduces a simplified, flexible method of configuring data collection using data collection rules (DCRs). This article provides guidance on how to implement a successful migration from the Log Analytics agent to Azure Monitor Agent.
- Important note: The Log Analytics agent will be retired on August 31, 2024. After this date, Microsoft will no longer provide any support for the Log Analytics agent. If you're currently using the Log Analytics agent with Azure Monitor or other supported features and services, start planning your migration to Azure Monitor Agent by using the information in this article.

The image is a collage of several Microsoft-related posts and a GitHub profile.

Top Left: Get insights on PostgreSQL data with Azure Sentinel

Top Right: Ingest DCR-based custom logs in Microsoft Sentinel with Logstash

Middle Left: Secure your Logstash connections to Microsoft Sentinel

Middle Right: Koos Goossens (9 min read - Jan 26) - Me and my colleague Dennis van den Akker wrote a blog about implementing automatic key rotation for your ARM service connections in Azure DevOps. Stop not letting your app registration secrets expire! Automate as much as you can and embrace a DevOps mindset. This leads to a higher security posture and less shadow IT. <https://lnkd.in/gQvqa>

Bottom Left: Koos Goossens (TheCloudScout) - PowerShell · 21 · 8 · Updated on Dec 1, 2022

Bottom Right: TheCloudScout / sentinel-pricing · PowerShell · 21 · 8 · Updated 12 hours ago

Bottom Center: WHERE'S THE PASSWORD (image of Jeff Bridges as The Dude from The Big Lebowski)

Bottom Right Logo: DUTCH MICROSOFT SECURITY MEETUP







Are
there
any
questions? _

