

SIEM for Your Multi Account AWS Organization

By Veselin Hristov

2021

Table of Contents

Intended for	3
Introduction	3
Log data to Amazon S3	4
AWS CloudTrail Organization logs to Amazon S3	4
AWS CloudWatch logs to Amazon S3	5
AWS VPC Flow logs to S3	6
AWS Web Application Firewall (WAF) to Amazon S3	7
AWS Security Hub to S3	8
Amazon S3 log data to Amazon Elasticsearch Service (ES)	9
Amazon S3 event notification to AWS Lambda	9
AWS Lambda to Elasticsearch	9
Kibana to analyze	10
Visualizations	10
Dashboards	15
Kibana to alert	18
Alerting / Dashboard	19
Alerting / Monitor	20
Trigger	20
Trigger Action	20
Kibana Authentication	21
Index snapshots (Backups)	22
Meta-monitoring	22
Conclusions	22
References	23
About HeleCloud	23

Intended for

Robust features, ease of use, and support are the standard items that make enterprise solutions stand apart from their free SIEM solution counterparts. However, enterprise SIEM solutions can be very different from one another. For example, a majority of SIEM tools are intended for huge organisations and would be far too complex and costly for smaller organisations.

We firmly believe organisations, no matter of their size, should pay as they go for the scale they need whilst retaining full control over every single event across their environments.

Introduction

The purpose of this whitepaper is to explain how the HeleCloud Security Information and Event Management (SIEM) solution is designed to support multi-account AWS workloads and ingests log data from your AWS Organizations accounts. The data is collected at source and aggregated into Amazon Simple Storage Service (Amazon S3) for security and compliance before being ingested into the AWS Elasticsearch service as its final destination. Also, this whitepaper will demonstrate how that data is used for visualisations and reporting, notifying and alerting.

SIEM is not any single technology or service, it is a system for real-time analysis of logs generated by applications, AWS services, anti-virus software (AV), image scanning tools and much more. The data ingested into the SIEM can be used to determine the security and compliance posture of AWS workloads by defining security rules matching certain criteria or thresholds. Of course, it would not be complete SIEM without the ability to trigger alarms, send alerts, raise tickets to someone and so on.

Areas covered by any SIEM system include:

- Log management: Simple collection and storage of log messages and audit trails.
- Security Information Management (SIM): Long-term storage as well as analysis and reporting of log data.
- Security Event Manager (SEM): Real-time monitoring, correlation of events, notifications and console views.
- Security Information and Event Management (SIEM): Combines SIM and SEM and provides real-time analysis of security alerts generated by network hardware and applications.

Log data to Amazon S3

One of the key components when using a multi-account architecture is centralised logging. It provides a single pane of glass to all salient logs generated across accounts and regions, and is critical for auditing, security and compliance. The HeleCloud Landing Zone solution deploys an AWS account specifically for archiving all logs inside Amazon S3 (S3) buckets for integrity and future analysis. Access to this account is highly regulated (and recorded via Amazon CloudTrail) and only teams with specific requirements to access this data are granted access (e.g., security personal only). The S3 buckets holding log data have least privilege policies applied to them, to minimize the risk of them being exposed to unauthorised team members and accidental, deliberate alteration, deletion. Where a service supports additional mechanisms to ensure the integrity of the log files (e.g., a digest file) these are also enabled.

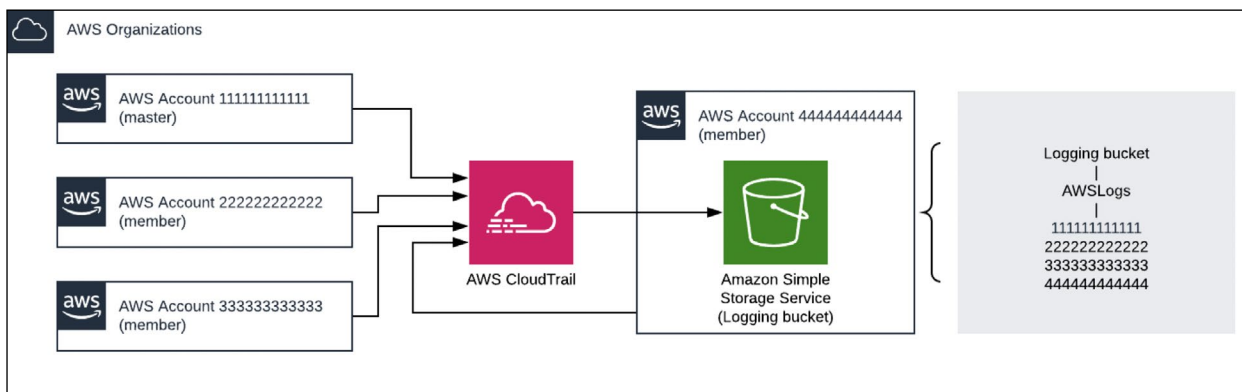
Amazon S3 allows organisations to keep files indefinitely at very low cost. By defining an Amazon S3 Lifecycle policy it is possible to move the objects to different storage classes such as Glacier to further reduce these costs. Keeping data for long periods of time inside Elasticsearch is comparatively very expensive compared to keeping them in Amazon S3. Therefore, from a cost and performance perspective it's good practice to keep the logs inside Elasticsearch for short period of time. The HeleCloud solution is optimised to ensure that only the required data is retained with ElasticSearch to reduce running costs and improve performance whilst ensuring it retains historical data securely and cost efficiently using services such as Amazon S3 and Glacier.

The following sub-sections focus on the different log sources, and how specific log sources are recorded to Amazon S3 buckets. The HeleCloud SIEM defines a different bucket for each specific log source to simplify the bucket structure, data governance and event management.

AWS CloudTrail Organization logs to Amazon S3

AWS recommend a multi-account environment as a security and compliance best practice. By isolating workloads or applications into individual accounts enables natural security, access and billing boundaries for the AWS resources. This approach enables the user to achieve resource independence and isolation. AWS Organizations can be used to centrally audit, monitor, and secure the AWS environment to ensure it is compliant with corporate policies. Defining an organisation-wide AWS CloudTrail trail to centrally log all actions performed across the environment will protect it from tampering at the account level. Organization trails are similar to regular trails in many ways; providing the ability to create multiple trails for the organization, choose single or multiple regions to trail, and choose the type of events to be logged in the organization trail, just like any other trail.

When creating the organization trail the user can specify which Amazon S3 bucket should receive the logs. Organization trails are automatically applied to all member accounts in the organization. Member accounts can see the organization trail but cannot modify or delete it. By default, member accounts do not have access to the log files for the organization trail in the Amazon S3 bucket. This helps the user uniformly apply and enforce the event logging strategy across the accounts in your organization. To enable that access the user must sign into the AWS Organizations master account to create a trail. If this is done through the AWS CloudTrail console, trusted access is configured automatically. This can also be created through AWS Command Line Interface (CLI) or AWS Application Programming Interface (API), with access manually configured.



The above diagram displays how API changes across all the AWS Accounts are being recorded to the organization CloudTrail, which is integrated with Amazon S3 to receive those events in the form of JSON files.

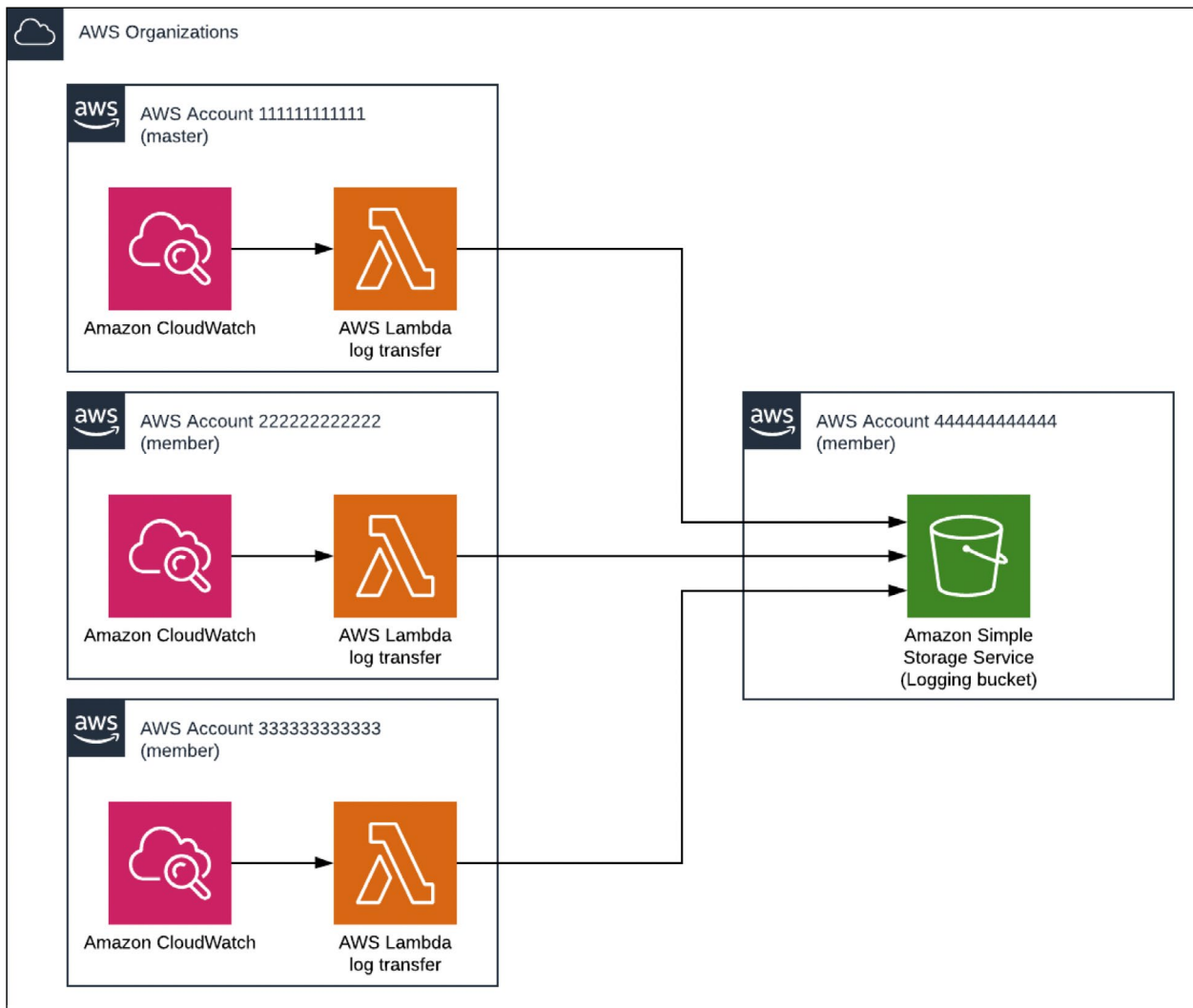
AWS CloudWatch logs to Amazon S3

The Amazon CloudWatch service allows you to collect and store logs from resources, applications, and services in near real-time. CloudWatch allows collection and access to all the performance and operational data in the form of logs and metrics from a single platform. There are three main categories of logs:

- AWS native, published by services on the customers behalf. Currently Amazon Virtual Private Cloud VPC (VPC)
- Flow logs and Amazon Route 53 logs are the two supported types.
- AWS logs published by AWS services. Currently over 30 AWS services publish logs to CloudWatch. These services include Amazon API Gateway, AWS CloudTrail, AWS Lambda, Amazon GuardDuty and many others.
- Custom logs that originate from the owned application and on-premises resources.

All these logs are sent to a log stream. A log stream is a sequence of log events that share the same source. Each separate source of logs in CloudWatch Logs make up a separate log stream. A log group is a group of log streams that share the same retention, monitoring, and access control settings. The user can define log groups and specify which streams to put into each group.

Real-time processing of log groups and subscriptions is important. Subscriptions can be used to get access to a real-time feed of log events from CloudWatch Logs and be delivered to other services such as an Amazon Kinesis stream, an Amazon Kinesis Data Firehose stream or in this case to AWS Lambda for custom processing, analysis, or loading to other systems such as Amazon S3. In the HeleCloud SIEM the Lambda function's only job is to transfer log data from the received event to the S3 Logging bucket. Each CloudWatch Log Group in each AWS Account under the organization is subscribed to a Lambda function that is responsible for transferring log data in Amazon S3. CloudWatch Log data is stored in JSON format and HeleCloud retain this format when writing the log files to Amazon S3.



Another common scenario could be that a new AWS service is enabled that creates a new CloudWatch Log Group; this would not automatically subscribe itself to the *log-transferring* Lambda. This issue is addressed by a second Lambda function, called '*log-subscribing*'. When a new CloudWatch Log Group is created in any AWS account, a CloudWatch Event is automatically generated. That event is sent to the '*log-subscribing*' Lambda, which extracts the name of the new CloudWatch log group and subscribes it to the *log-transferring* Lambda.

AWS VPC Flow logs to S3

Amazon VPC Flow Logs enable the capture of information about the network traffic moving to and from network interfaces within a VPC. VPC Flow Logs can be used as a centralised, single source of information to monitor different network aspects of a VPC. VPC Flow logging gives security engineers a history of high-level network traffic flows within entire VPCs, subnets, or specific network interfaces (ENIs). This makes VPC Flow Logs a useful source of information for detection teams focused on collecting network instrumentation across large groups of instances.

Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. HeleCloud chose to send the logs directly to S3. If sent to CloudWatch, those will eventually be transferred to S3 by '*log-transferring*' Lambda. After the flow log is created, its data can be retrieved and viewed in the chosen destination. VPC

Flow Logs are the only log that HeleCloud do not get in JSON format making them an edge case requiring special attention when processing the data. More details about this can be found in the, "*AWS Lambda to Elasticsearch*" section.

AWS Web Application Firewall (WAF) to Amazon S3

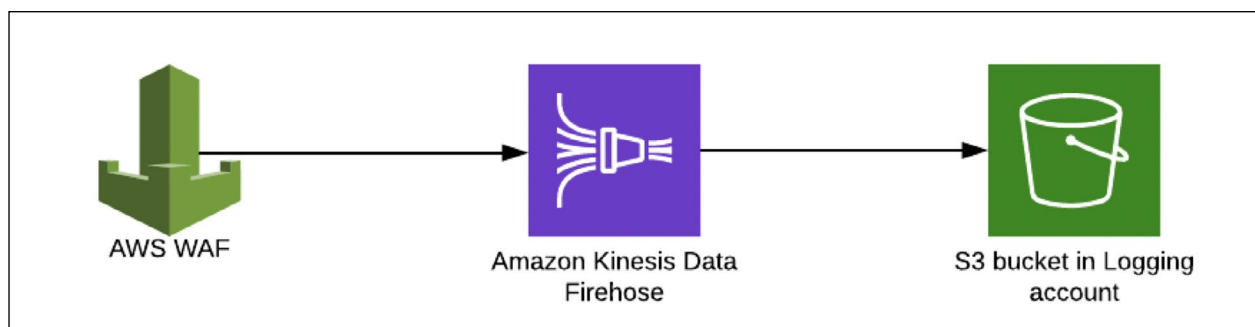
AWS provides services such as Amazon Elastic Compute Cloud (EC2), Elastic Load Balancers (ELB), Amazon S3 and Amazon Elastic Block Store (EBS) to create reliable and scalable applications quickly for less CAPital EXPenditure (CAPEX) than traditional data center approaches. Regardless of these benefits, it is equally important to secure the application and protect its data which is where services like the AWS Web Application Firewall can assist. If not properly secured, the system could be compromised, and sensitive data might get into the wrong hands. Such cases regularly make the front pages of the national and security press. This section is not focused on how to configure the AWS Web Application Firewall (WAF), but rather how best to get the logs into a centralised secure bucket location, that no one except restricted personnel has access to.

AWS WAF supports full logging of all web requests inspected by the service. These logs can be stored in Amazon S3 for compliance and auditing needs as well as used for debugging and additional forensics. Logs help to understand why certain rules are triggered and why certain web requests are blocked.

The two-step process below shares how to stream AWS WAF Logs in each account in the Organization as required:

- On the Amazon Kinesis console, create an instance of Amazon Kinesis Data Firehose. As part of this configuration, choose a destination for the data; Amazon S3, Amazon RedShift or Amazon Elasticsearch. HeleCloud's preferred option is Amazon S3 instead of Amazon Elasticsearch for the destination of the logs.
- Through the AWS WAF console, enable logging and select the Firehose instance.

For each web request, AWS WAF logs provide raw HTTP/S headers along with information on which AWS WAF rules are triggered. This is useful for troubleshooting custom WAF rules and Managed Rules for AWS WAF. These logs will be made available via Amazon Kinesis Data Firehose in JavaScript Object Notation (JSON) format.



AWS Security Hub to S3

AWS contains a variety of services designed to help organizations meet certain security and compliance goals, including:

- [Amazon GuardDuty](#) – threat detection that monitors for malicious activity and unauthorised behavior
- [Amazon Inspector](#) – assess applications for exposure, vulnerabilities, and deviations from best practices.

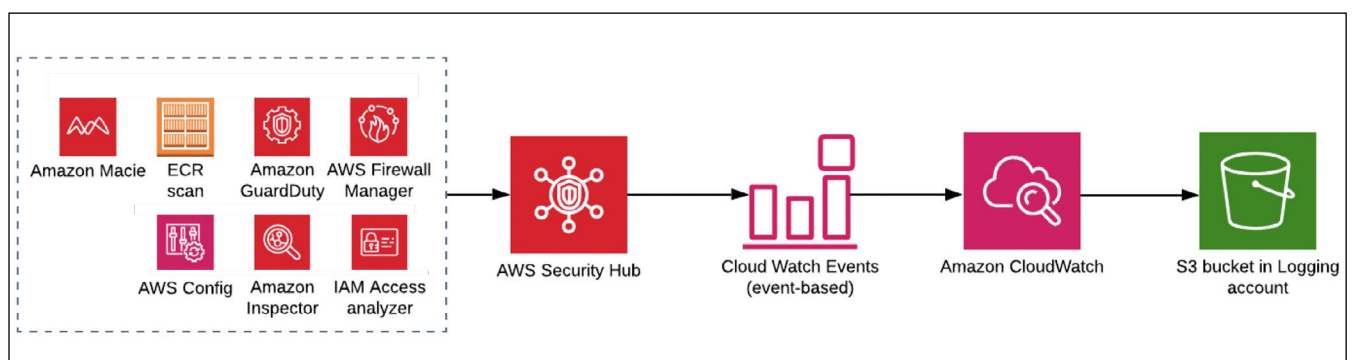
While GuardDuty and Inspector provide valuable assessment and detection capabilities, their use alone is by no means sufficient in meeting security and compliance needs. In addition to findings reported from these services and AWS Partner Integrations, it's also possible to perform continuous configuration and compliance checks based on industry standards, such as the [CIS AWS Foundations Benchmarks](#). The CIS AWS Foundations Benchmarks provide prescriptive guidance on how to configure a handful of AWS Services to meet various security and compliance best practices. These checks are enforced via a set of [AWS Config](#) rules.

In 2018, AWS launched its integrated security tool called AWS Security Hub, which can provide a comprehensive view of your security state in your AWS environments. Security Hub centralises and prioritises security findings from across AWS accounts, services, and supported third-party partners to help analyse security trends and identify the highest priority security issues.

Security Hub collects findings from the security services enabled across AWS accounts, such as intrusion detection findings from Amazon GuardDuty, vulnerability scans from Amazon Inspector, sensitive data identification findings from Amazon Macie and compliance checks associated with Centre of Internet Security (CIS) AWS Foundations Benchmarks. Security Hub also collects findings from partner security products using a standardised AWS Security Finding Format (JSON), eliminating the need for time-consuming data parsing and normalisation efforts. Customers can designate a primary account that can see all findings across their accounts.

Security Hub generates its own findings by running continuous and automated account and resource-level configuration checks against the rules in the supported industry best practices and standards (for example, the [CIS AWS Foundations Benchmark](#)).

AWS Security Hub supports workflow options by enabling the export of findings via CloudWatch events. The HeleCloud SIEM has a CloudWatch rule which is monitoring for any events coming from Security Hub. The target for that rule is configured to be AWS CloudWatch Logs. As discussed earlier the creation of a new log group triggers a Lambda function that subscribes and transfers the data to Amazon S3. AWS Security Hub findings pipeline is visualised in below diagram.



Amazon S3 log data to Amazon Elasticsearch Service (ES)

Elasticsearch is a distributed document store. Instead of storing information as rows of columnar data, Elasticsearch stores complex data structures that have been serialised as JSON documents. When multiple Elasticsearch nodes are in a cluster; stored documents are distributed across the cluster and can be accessed immediately from any node.

When a document is stored, it is indexed and fully searchable in near real-time, within 1 second. Elasticsearch uses a data structure called an inverted index that supports very fast full-text searches. An inverted index lists every unique word that appears in any document and identifies all the documents each word occurs in.

As previously covered, different log data can be transferred from its origin to Amazon S3 for safe keeping. Whether it was AWS API calls, logs published by AWS services, AWS native logs, custom logs or even on-prem logs. All the log data flows through different data pipelines but ultimately it is all stored in an Amazon S3 bucket in a centralised account. The next section continues to follow the data's journey and examine how it reaches Elasticsearch.

Amazon S3 event notification to AWS Lambda

The Amazon S3 notification feature enables the user to receive notifications when certain events happen in their bucket. For example, whenever an action is applied to an S3 object, an event is created. This event can send notifications to Simple Queue Services (SQS), Simple Notification Service

(SNS) or the AWS Lambda service. These events can be used to enable event-driven workflows.

AWS Lambda can run custom code in response to Amazon S3 bucket events. Custom code can be uploaded to AWS Lambda and create what is called a Lambda function. When Amazon S3 detects an event of a specific type (for example, an object created event), it can publish the event to AWS Lambda and invoke the function in Lambda. This is called event driven processing – the Lambda function is invoked in response to the object create event. In response, AWS Lambda executes the function, parses the received event to determine which S3 object triggered it and reads it for further processing.

Decoupling with a message queue such as SQS before log processing by AWS Lambda service will bring some stability and redundancy. Having logs in a queue guarantees to some extent that they will eventually reach their final stop even when that Lambda is undergoing maintenance to that Lambda.

AWS Lambda to Elasticsearch

Using event-driven processing as explained, and how the Lambda function is triggered by an S3 event, that event contains information about which S3 object the event is for. The next step is to read that object and start processing it.

Elasticsearch stores data structures in the form of serialised JSON documents which helps with next steps. At this point get the file associated with the S3 event, open it and read a multi-dimensional dictionary, then flatten it, separating the nested keys with a separator. The example below illustrates the process of flattening the nested keys in the log data:

```

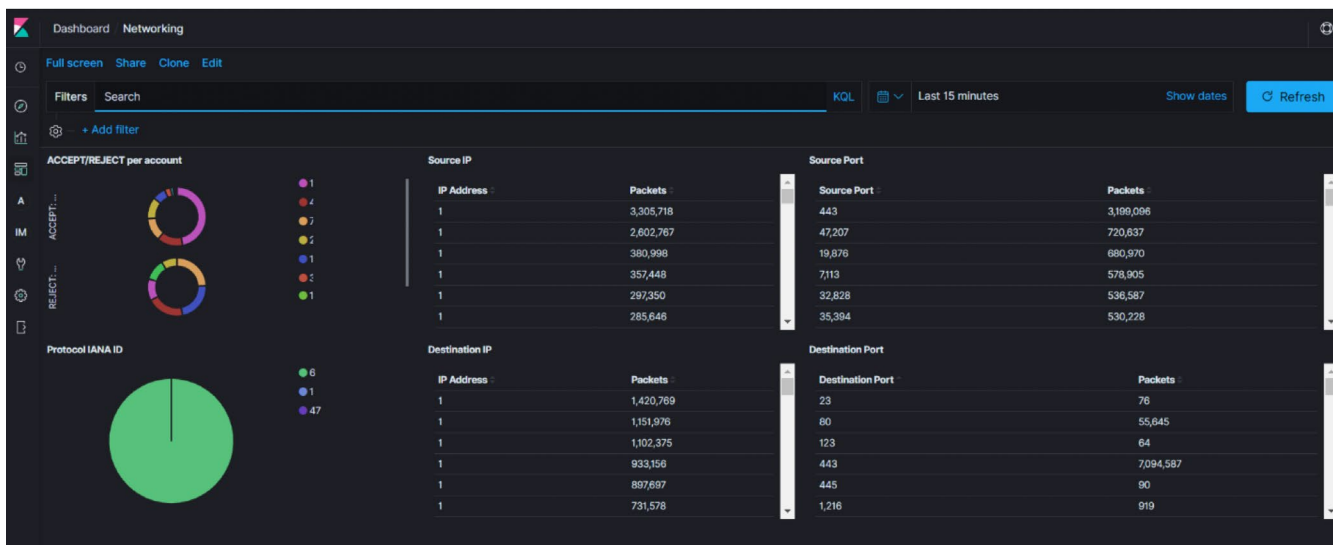
1  {
2      a1: "Level1",
3      b1: {
4          a2: "Level2",      = >
5          c2: {
6              c2: {
7                  a3: "Level3",
8                  b3: [
9                      "Level4",
10                     "Level4"
11                 ]
12             }
13         }
14     }

```

Sending logs to Elasticsearch happens through an API. The Elasticsearch documentation reveals the [Index API](#), which is used for adding a document in order to make it searchable. The only drawback of the Index API is that it is synchronous. This means that when it receives a request for storing information, it will process it first and then return a status code indicating the outcome of the action (i.e., it successfully completed the request). Imagine that the Lambda is processing 1,000,000 JSON records in 1 execution. For each of those an Elasticsearch request will be sent and kept waiting for a response after which will move on to the next one. This creates both an unwanted CPU load on Elasticsearch and increases the run time for the Lambda function as it sits waiting for Elasticsearch to return the outcome of each request. That is why a way to perform multiple Elasticsearch operations in a single API request is needed. The answer to this problem is to use the [Bulk API](#) which dramatically reduces the processing overhead and greatly increases the indexing speed.

As previously noted, VPC flow logs and how they are generated by default in a non-JSON format, in other words plain-text logs. This creates some additional complexity because this data needs to be processed by a separate Lambda function. By default, the log line format for a flow log record is a space-separated string that has a predefined set of fields supplied in order. The details of flow log records can be seen here. These fields are used to define the keys for the dictionary. Each line from the VPC flow log contains the values for the dictionary. When JSON is ready, data is sent to Elasticsearch again in bulk (because each line would do a request to the cluster).

Kibana to analyze



Previous sections explain how to get the data to Elasticsearch. Once it is in Elasticsearch the log data can be analysed and detect if something has breached a threshold, or an unusual pattern of behavior has occurred.

Kibana is a popular open-source dashboard creation tool designed to work with Elasticsearch Service (ES). Amazon ES provides an installation of Kibana with every Amazon ES domain. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data. Kibana is designed to use Elasticsearch as a data source. Think of Elasticsearch as the engine that stores and processes the data, with Kibana sitting on top to provide a fully customisable visual representation of key aspects of the system being monitored.

The following sections will cover enabling visualizations when those thresholds are met.

Visualizations

The Visualize tab enables the user to create visualizations of the data from the Elasticsearch indices, which can then be added to dashboards for analysis.

Kibana visualizations are based on Elasticsearch queries. By using a series of Elasticsearch aggregations to extract and process the data, create charts that showing the all-important trends, spikes and dips.

The HeleCloud SIEM provides a series of queries based upon AWS best practice to help businesses. Kibana can be used to build visualizations from, enabling the user to look at a certain visualization and includes a link to the saved query highlighting the details behind that visualization. Custom visualizations are built by HeleCloud Managed Service team to simplify the process.

Here is an example that queries the CloudTrail log data for all information pertaining to Amazon S3.

Under the *Discover* tab, select the CloudTrail index pattern and then add a filter for the *event Source* field to have the value, *s3.amazonaws.com*. When this filter is saved, Elasticsearch will run through its indices to retrieve the documents which meet that criterion.

+ Add filter

EDIT FILTER
Edit as Query DSL

Field
Operator

eventSource
is

Value

s3.amazonaws.com

☐ Create custom label?

Cancel Save

The output of the above filter will be every single event that was made in the respective account for the Amazon S3 service – this will probably be quite a lot of data!

To be more specific and filter out everything except the, *AccessDenied* messages, simply filter out by the *errorCode* field and set the value to, *AccessDenied*.

+ Add filter

EDIT FILTER
Edit as Query DSL

Field
Operator

errorCode
is

Value

AccessDenied

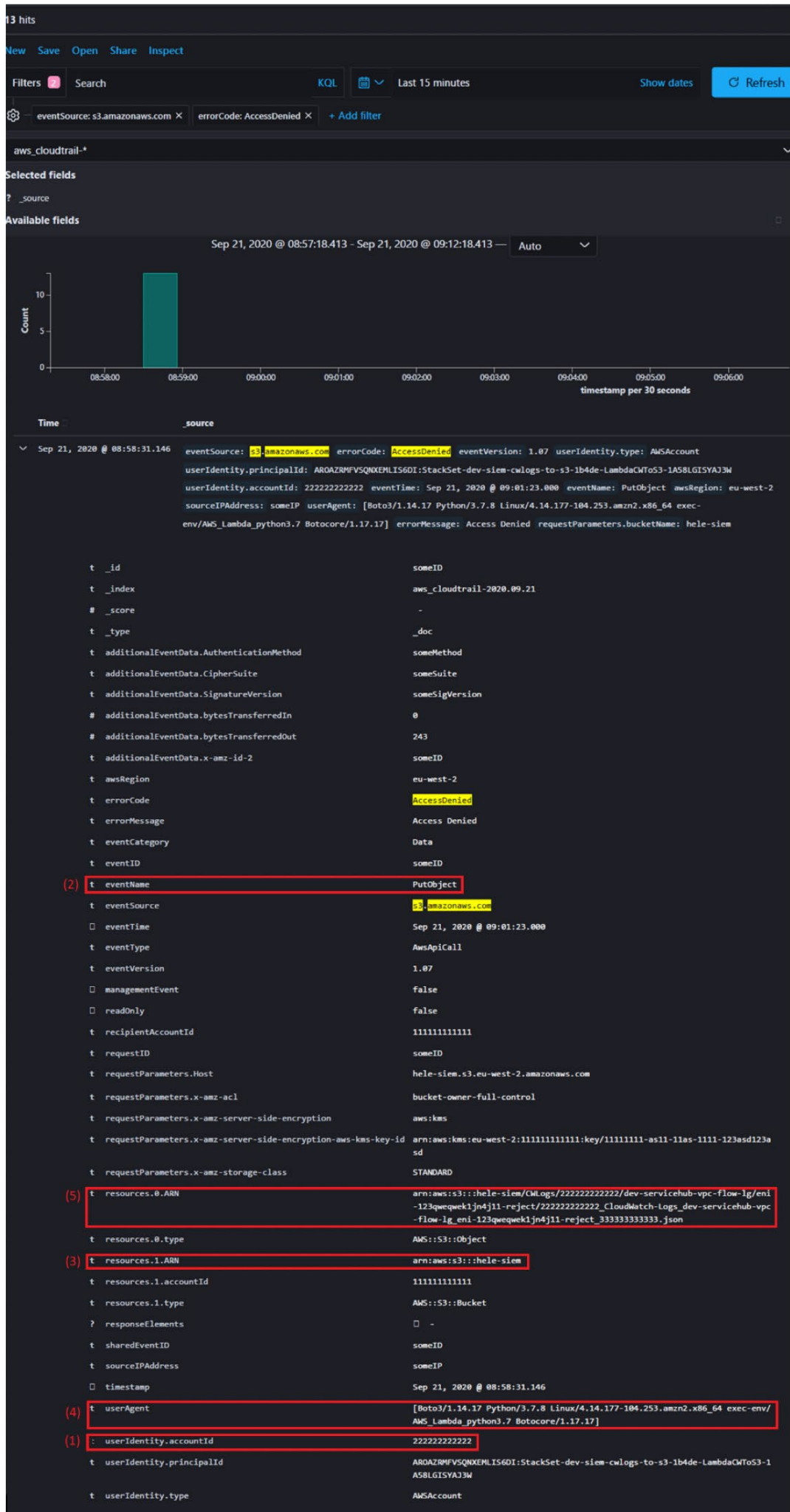
☐ Create custom label?

Cancel Save

In this hypothetical example 13 hits in the results are revealed. This means that in the past 15 minutes (the default cadence of log updates) something or someone has tried to access an S3 bucket and received *AccessDenied* response 13 times. By expanding one of the documents, the cause of those responses can be identified.

This example shows another account (field: *userIdentity.accountId*)(1) was trying to upload (field: *eventName*)(2) a file to a bucket with ARN (Amazon Resource Name) *arn:aws:s3:::hele-siem* (field: *resources.1.ARN*)(3).

The *UserAgent*(4) field shows Python SDK (Software Development Kit) is being used to make that request. This can be concerning as the user may think that someone is probing their access policies. In fact, when checking the field *resources.0.ARN*(5) it shows the destination file is a VPC flow log and it turns out that this is just a misconfigured access policy between both accounts.



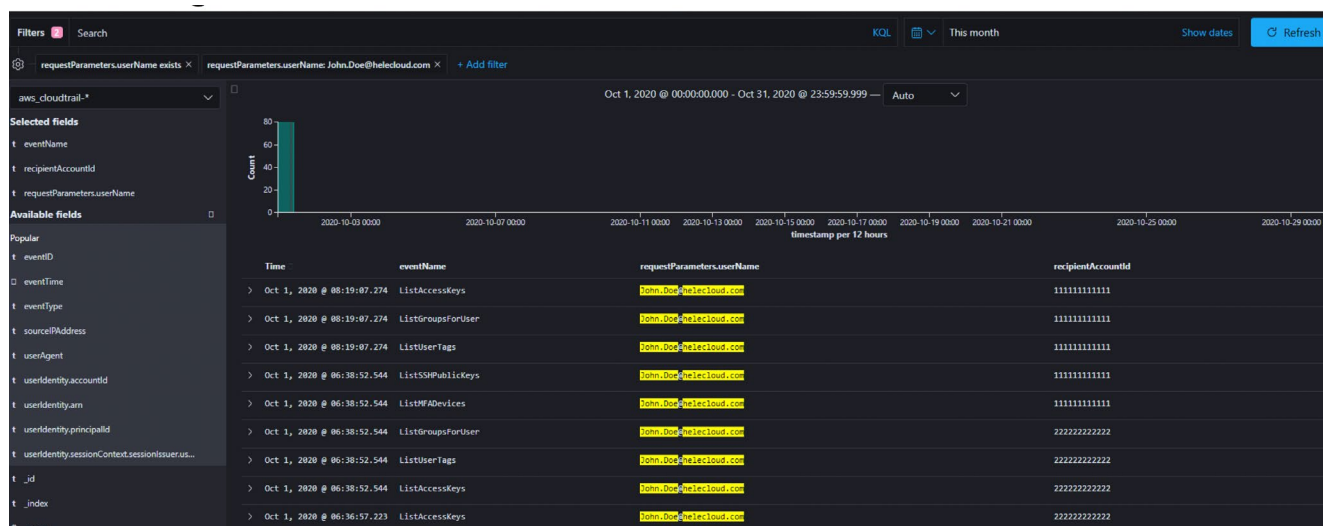
Another common scenario is to monitor for events coming from the EC2 service for example, when detecting illegal EC2 instances being spun up for the purposes of crypto mining. This can be done with a combination of monitoring for events that are, *RunInstance* and the instance type is not on the white list. An alarm can be triggered in response to a positive result from the query so that someone can check the details of the EC2 instance(s) and ascertain whether they are legitimately provisioned or are an indication of suspicious activity.

Another important example is how to detect tampering with CloudTrail since that is one of the key data sources in the SIEM system. In this example, watch for any events coming from *cloudtrail.amazonaws.com* with any of the following values for the *EventName* field:

- *DeleteTrail*
- *RemoveTags*
- *StopLogging*
- *UpdateTrail*

Later in this section shares how and why any attempt to invoke any of the above events should result in the security team being notified.

A common scenario is to use the SIEM to extract all the actions carried out by a specific user over a given timeframe (e.g., today, yesterday, past month etc.). To do this simply identify the correct field to search by, set the value to the user's email and set the time range. The screenshot below lists every action the user *John.Doe@helecloud.com* did across all accounts.



This query immediately returns both the accounts and events initiated by Mr. John Doe. In this hypothetical scenario none of the actions carried out (e.g., *ListAccessKeys*, *ListGroupsForUser* etc..) are considered restricted or suspicious and no further investigation is required.

The HeleCloud SIEM solution comes provisioned with around 100 individual Kibana Visualizations, each backed up with a search query that is monitors the AWS multi account security. The next section will discuss how to aggregate those visualizations into purpose-built dashboards.

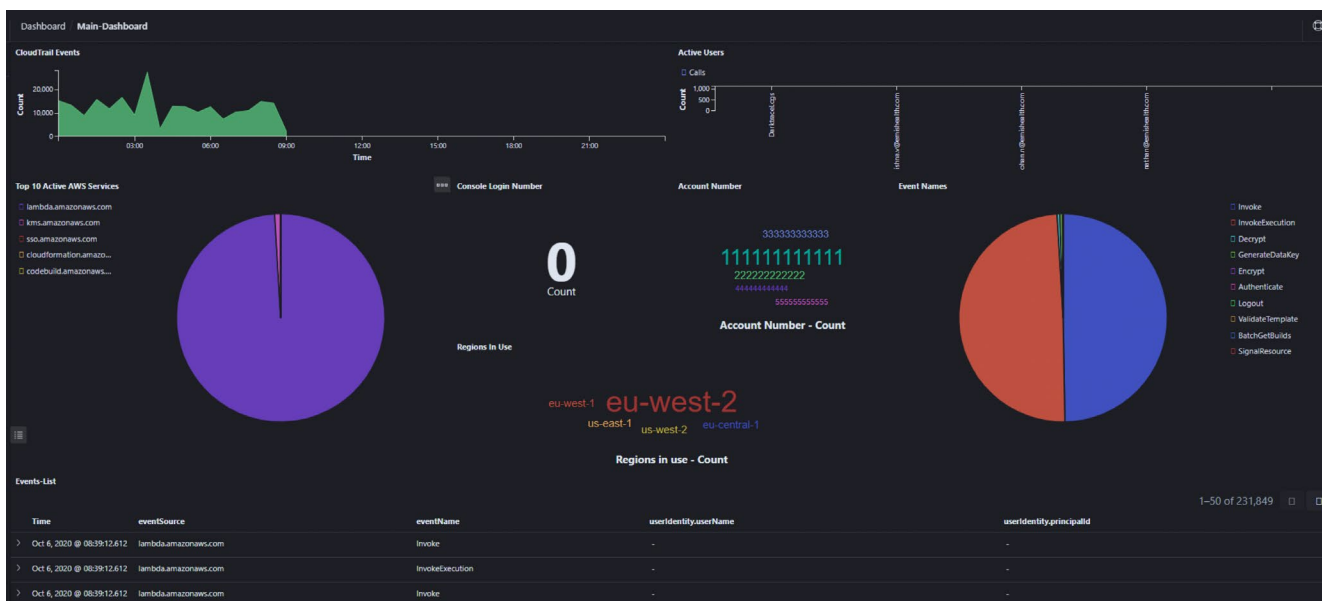
Dashboards

Dashboards allow the user to create a collection of visualizations, searches, and maps, typically in real-time. Dashboards provide at-a-glance insights into the data and enable the user to drill down into details.

In the example below, the following visualizations are displayed:

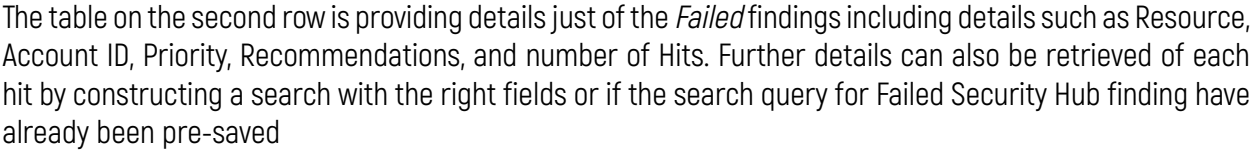
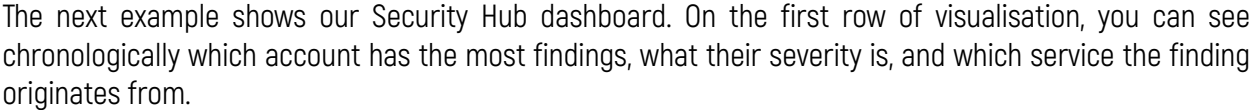
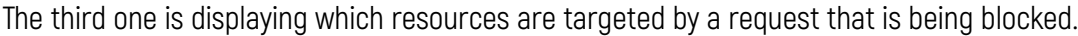
- Currently active users,
- Top 10 active AWS services,
- Event actions,
- Monitor active regions

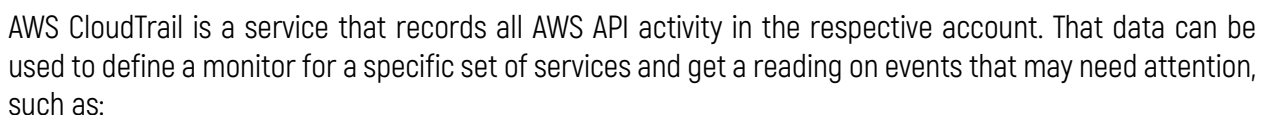
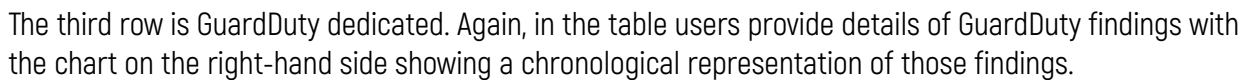
All these visualizations can be displayed on one Dashboard which in this example is called the "Main Dashboard":



The Kibana dashboard is interactive and clicking on the visualizations will set up a filter for the user to see events specific to that one. For example, by clicking under the "Account number" visualization on any of the accounts displayed, Kibana will filter out the rest of the visualizations to display data only for that account. This is an extremely fast way to zoom into data focusing on a specific aspect of the system. The same applies for every other field displayed in the visualization.

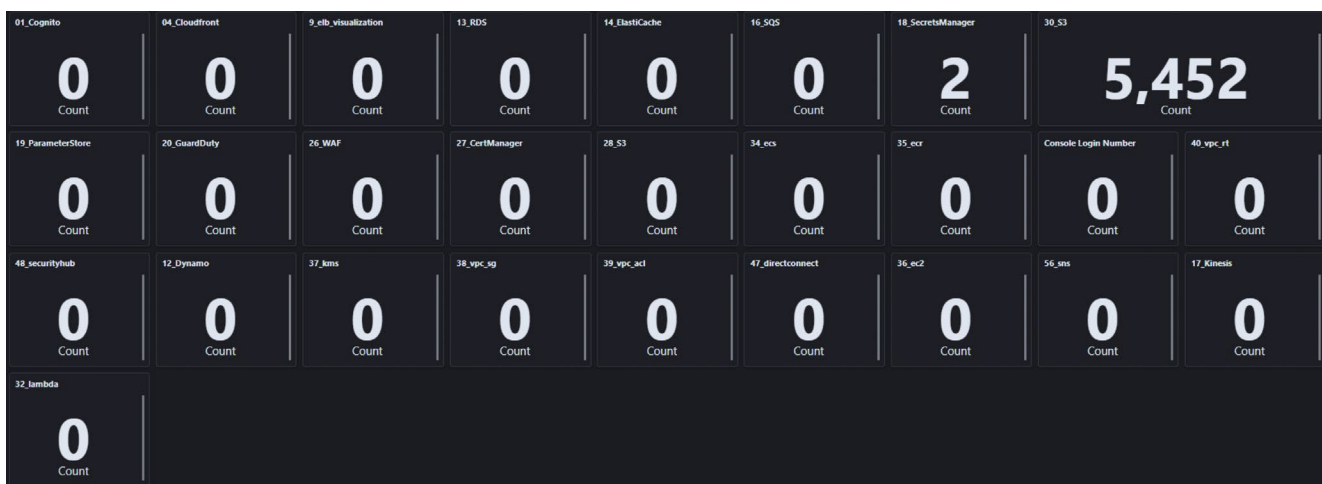
The next example Kibana dashboard displays visualizations from the AWS WAF event source. The multiple rows of visualizations shown below. The first one displays which countries the requests are coming from as well as the ALLOW/BLOCK request count over time.





- Too many EC2 machines being created in a short period of time, this can mean someone is trying to do crypto mining in an account.
- Exceeding a threshold for the amount of Lambda executions may mean also crypto mining.
- A high volume of S3 access denied attempts may mean someone is probing S3 buckets.
- Someone updating security groups of a resource in an account.
- Create/Update/Delete tables in services like DynamoDB and RDBs.
- Update/Put/Create/Delete events on services like AWS SecretsManager and Parameter store.
- For AWS CertificateManager you will want to get a reading on who did Exports, Imports, Delete certificates and so on.

There are around 60 different CloudTrail metrics, like the ones described above, that are constantly being monitored. For each metric there is a separate visualization, which is included in the CloudTrail Dashboard. The following screenshot shows part of those visualizations:



It is time to dig deeper into the CloudTrail dashboard. In the visualization *18_SecretsManager* is monitored for any of the following events:

- *Create**,
- *Delete**,
- *Update**,
- *Put**,
- *Tag*,
- *TagResource*,
- *UntagResource*,
- *GetSecretValue*.

As shown, two events that fall into the above categories. The next step is to open the Search Query behind the visualization and check what or who is responsible for this reading.

Next to that visualization is "30_S3" which displays a Search Query that checks for S3 Access Denied responses i.e., when someone or something tried to access any of the S3 buckets in one of our AWS accounts. In this example we have a very high hit count metric of, '5,432'. This does not necessarily mean someone is probing the bucket as it could also mean that there is a service or an application that might be misconfigured and we need to grant access to that bucket. However, it does mean further investigation is required to determine the actual cause.

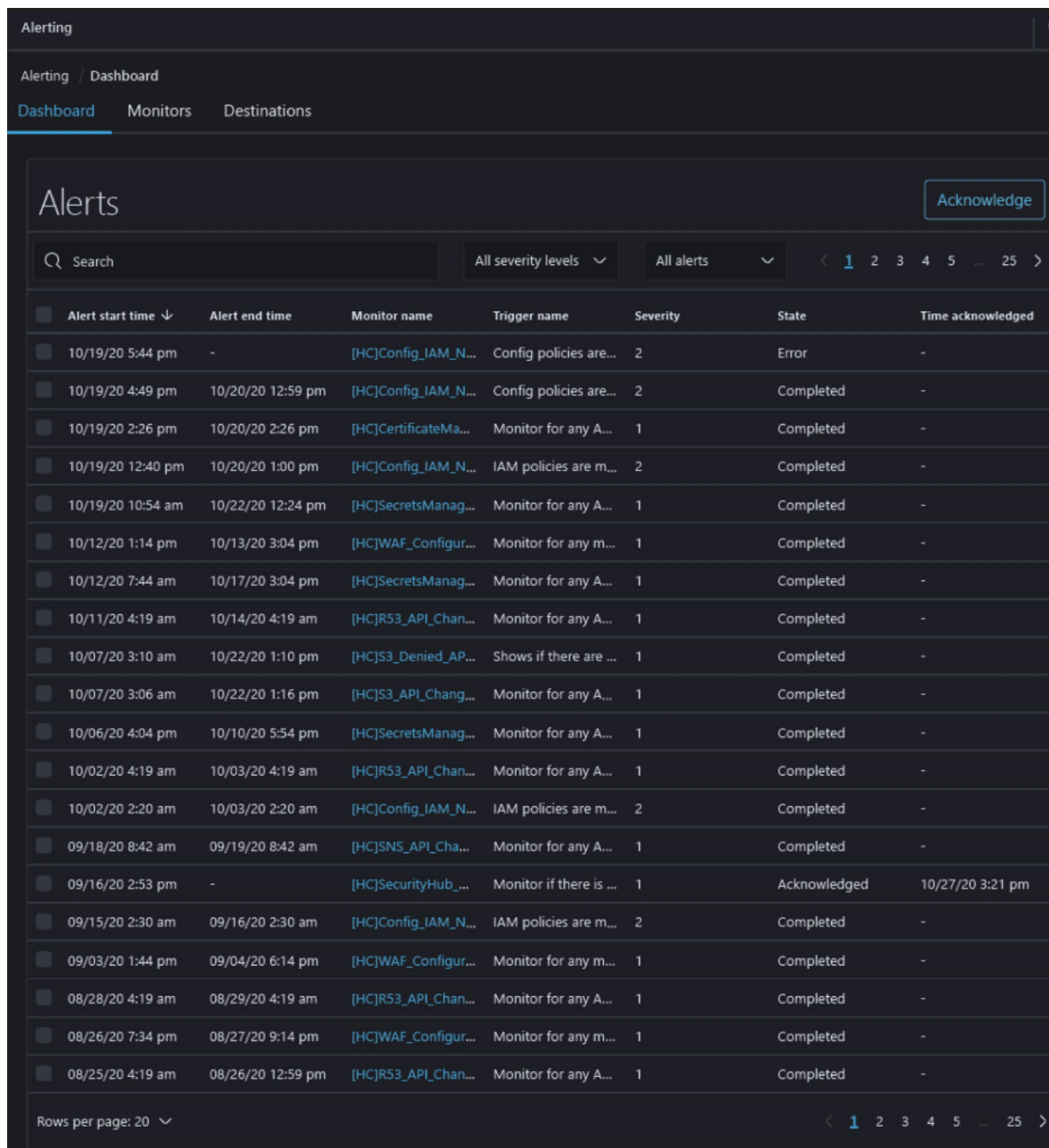
Kibana to alert

In the previous sections, the process of first creating Search Queries, second creating Visualizations and finally assembling Dashboards have been discussed. In this part, Event management is covered, which is the "E" in SIEM acronym.

Fortunately, Kibana has a handy plugin to help with the task, which (un)surprisingly is called "*Alerting*". Its purpose is to send a notification when certain conditions are met in Elasticsearch indices data.

Alerting / Dashboard

The following screenshot shows an example of the alerting tab in Kibana.



The screenshot displays the Kibana Alerting Dashboard. At the top, there's a navigation bar with 'Alerting / Dashboard' and tabs for 'Dashboard', 'Monitors', and 'Destinations'. The 'Alerts' section is active, showing a table of alerts. The table has columns for 'Alert start time', 'Alert end time', 'Monitor name', 'Trigger name', 'Severity', 'State', and 'Time acknowledged'. The alerts are sorted by start time, showing a list of alerts with various states like 'Error', 'Completed', and 'Acknowledged'. A search bar and filters for severity levels and alert types are visible at the top of the table. A pagination bar at the bottom shows 'Rows per page: 20' and a range of page numbers from 1 to 25.

Alert start time ↓	Alert end time	Monitor name	Trigger name	Severity	State	Time acknowledged
10/19/20 5:44 pm	-	[HC]Config_IAM_N...	Config policies are...	2	Error	-
10/19/20 4:49 pm	10/20/20 12:59 pm	[HC]Config_IAM_N...	Config policies are...	2	Completed	-
10/19/20 2:26 pm	10/20/20 2:26 pm	[HC]CertificateMa...	Monitor for any A...	1	Completed	-
10/19/20 12:40 pm	10/20/20 1:00 pm	[HC]Config_IAM_N...	IAM policies are m...	2	Completed	-
10/19/20 10:54 am	10/22/20 12:24 pm	[HC]SecretsManag...	Monitor for any A...	1	Completed	-
10/12/20 1:14 pm	10/13/20 3:04 pm	[HC]WAF_Configur...	Monitor for any m...	1	Completed	-
10/12/20 7:44 am	10/17/20 3:04 pm	[HC]SecretsManag...	Monitor for any A...	1	Completed	-
10/11/20 4:19 am	10/14/20 4:19 am	[HC]R53_API_Chan...	Monitor for any A...	1	Completed	-
10/07/20 3:10 am	10/22/20 1:10 pm	[HC]S3_Denied_AP...	Shows if there are ...	1	Completed	-
10/07/20 3:06 am	10/22/20 1:16 pm	[HC]S3_API_Chang...	Monitor for any A...	1	Completed	-
10/06/20 4:04 pm	10/10/20 5:54 pm	[HC]SecretsManag...	Monitor for any A...	1	Completed	-
10/02/20 4:19 am	10/03/20 4:19 am	[HC]R53_API_Chan...	Monitor for any A...	1	Completed	-
10/02/20 2:20 am	10/03/20 2:20 am	[HC]Config_IAM_N...	IAM policies are m...	2	Completed	-
09/18/20 8:42 am	09/19/20 8:42 am	[HC]SNS_API_Cha...	Monitor for any A...	1	Completed	-
09/16/20 2:53 pm	-	[HC]SecurityHub_...	Monitor if there is ...	1	Acknowledged	10/27/20 3:21 pm
09/15/20 2:30 am	09/16/20 2:30 am	[HC]Config_IAM_N...	IAM policies are m...	2	Completed	-
09/03/20 1:44 pm	09/04/20 6:14 pm	[HC]WAF_Configur...	Monitor for any m...	1	Completed	-
08/28/20 4:19 am	08/29/20 4:19 am	[HC]R53_API_Chan...	Monitor for any A...	1	Completed	-
08/26/20 7:34 pm	08/27/20 9:14 pm	[HC]WAF_Configur...	Monitor for any m...	1	Completed	-
08/25/20 4:19 am	08/26/20 12:59 pm	[HC]R53_API_Chan...	Monitor for any A...	1	Completed	-

Whenever a pre-determined threshold is reached in the SIEM a certain alert is raised. The different states can be checked within this panel. Defining severity, execution schedule and name will be covered in the next section. There are a few steps that need to be taken care of before the above can be viewed.

Alerting / Monitor

The first step is to create the *monitors*. Monitors define the thresholds, which when breached raise an alert like the ones shared in the previous section. When creating a new monitor, the following details need to be provided:

- The monitor name.
- The time interval defined as a *cron* expression on how often the monitor will run.
- Which index to run against.
- Whether to use a visual graph or an extraction query to define that monitor.

There are search queries that can be used in order to simplify this task. After the *monitor* is created, a *trigger* needs to be defined.

Trigger

A name and severity level need to be specified for a trigger. The severity level helps to manage alerts. Different severity level triggers can be configured to notify different channels. For example, severity level 3 can send a message to a chat room when severity level 1 can send an email to a wider audience.

Earlier, this whitepaper specified an extraction query for the monitor but for creating a trigger a *trigger condition* needs to be specified

Trigger condition scripts revolve around the `ctx.results[0]` variable, which corresponds to the extraction query response. For example, a script might reference `ctx.results[0].hits.total.value` or `ctx.results[0].hits.hits[i]._source.error_code`.

A return value of *true* means the trigger condition has been met, and the trigger should execute its actions.

Trigger Action

The final step when creating monitors is to define trigger actions. It is possible to add one or more actions when a Trigger condition is met, e.g., *true*.

Actions send notifications via any of the following channels:

- Amazon SNS (Simple Notification Service)
- Amazon Chime
- Slack
- Custom Webhook

HeleCloud's preferred choice of channel is to define an Amazon SNS (Simple Notification Service) topic with an email subscription. Upon receiving an email our Service Desk integration ensures that each Alert raised automatically creates an issue in our ticketing system.

Kibana Authentication

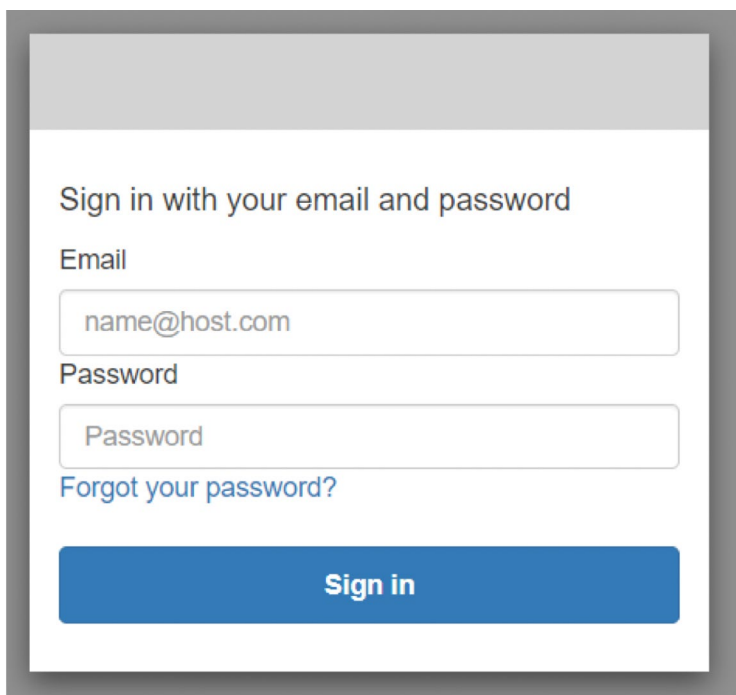
AWS Elasticsearch supports Amazon Cognito for authentication. Amazon Cognito provides authentication, authorization, and user management for web and mobile apps. Users can sign in directly with a username and password, or through a third party such as Facebook, Amazon, Google or Apple. Even if there is a cluster created through the console, Cognito for authentication can be selected. This setup will allow control for who has access to our Kibana instance, as a username/password will be required for access.

Before Amazon Cognito authentication for Kibana can be configured, prerequisites must be fulfilled. Amazon Cognito authentication for Kibana requires the following resources:

- Amazon Cognito [user pool](#)
- Amazon Cognito [identity pool](#)

A discussion on how to do this is outside the scope of this whitepaper however there is already a wealth of documentation available that describes the necessary steps. A good example can be found in this [blog post](#), for instance.

Once configured, when the Kibana URL is being hit in a browser, one will be redirected to a Cognito login page. Sign in with the user that has been created. After logging in, it redirects to the Kibana service.



Important to note that after enabling Cognito for authentication, AWS Elasticsearch service stops receiving anonymous requests. Before that Elasticsearch was restricting access via [resource-based policies](#), [identity-based policies](#) and [IP-based policies](#), now with Cognito authentication is needed in order to get access. Any Lambda functions sending data to our cluster need to be extended with a [python library](#) that handles authentication against AWS services.

Index snapshots (Backups)

Snapshots are backups of a cluster's indices and state. State includes cluster settings, node information, index settings, and shard allocation.

On the Amazon Elasticsearch Service, snapshots come in two forms, *automated* and *manual*:

- Automated snapshots are only for cluster recovery. They can be used to restore domains in the event of red cluster status or other data loss. Amazon ES stores automated snapshots in a preconfigured Amazon S3 bucket at no extra cost.
- Manual snapshots are for cluster recovery or moving data from one cluster to another. As the name suggests, initiate the manual snapshot. These snapshots are stored in the Amazon S3 bucket, and standard S3 charges apply. If there is a snapshot from a self-managed Elasticsearch cluster, that snapshot can even be used to migrate to an Amazon ES domain.

Every single Elasticsearch (v5.3 and above) cluster takes automatic snapshots on an hourly basis and retains them for 14 days, those are AWS managed and can be turned off thru console, cli or SDK. However, it is recommended to leave them running, because they are free of charge. As stated above, they find a good use when restoring a cluster from red cluster status.

With manual ones, snapshots can be kept for years, not just 14 days, which makes them perfect for restoring older data back. This, and knowing automated snapshots are free, enabling automated and manual snapshots together is recommended to support longer term storage.

For example, if an auditor requests an audit and requests one year-old data, it is desirable to retrieve that data quickly and ideally not spend hours searching manually through your S3 bucket for the data. Ideally a single query is run and gives the needed results. This is the second benefit of running manual backups because it simplifies loading data back in to Elasticsearch to make it searchable again. With a manual backup the day, month, year need to be identified to retrieve indexes and to be restored on the cluster.

Meta-monitoring

Meta-monitoring is a basic monitoring requirement to check whether the monitoring service is available and accessible. The Elasticsearch cluster is a valuable system asset and if it were to become inoperable, every single functionality described so far would be lost. Hence, it needs meta-monitoring to ensure to be immediately notified when the system is no longer functioning.

Fortunately, CloudWatch can be used to monitor Elasticsearch. There are built in Elasticsearch metrics in CloudWatch that can be used to create CloudWatch Alarms. Here is a [list of the recommended](#) ones. These alarms can trigger an action which in this case is again mail sent to Service Desk system.

Conclusions

Security information and event management tools allow businesses and organisations to obtain a wider view of their IT and network security throughout the entire organisation. With smart cyberattack monitoring and activity logs, combined with robust response management, organisations are better protected in a world in which new cyber threats arrive every day. It is in everyone's best interest to place a priority on protecting the organisation and its users. SIEM tools offer a comprehensive, streamlined solution to network security, and allow entrepreneurs to focus on nurturing their business.

References

- <https://aws.amazon.com/elasticsearch-service/>
- <https://opendistro.github.io/for-elasticsearch-docs/>
- <https://aws.amazon.com/cloudtrail/>
- <https://aws.amazon.com/cloudwatch/>
- <https://aws.amazon.com/cognito/>

About HeleCloud



HeleCloud™ provides strategic technology consultancy, engineering, and Cloud-based managed services for Retailers. By taking advantage of everything the Cloud has to offer, organisations can reduce operational costs, accelerate innovation, focus on their core business, and have more confidence around data security and compliance.

We take organisations on a complete journey into the Cloud environment. Right through from vision to implementation, we can help you transition into the Cloud future.

Our consultants will work with you to identify your digital needs in the context of your corporate vision and goals. As your Cloud competency centre, HeleCloud™ consultants have the knowledge and proven experience to guide and support you through the implementation of your Cloud strategy.