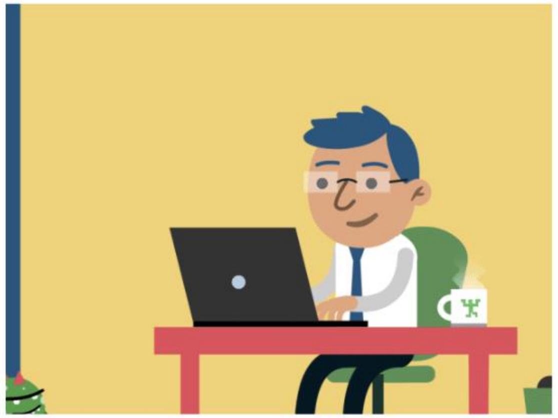
Fighting Financial Crimes As An SRE

Presented By
Anisha Manoharan

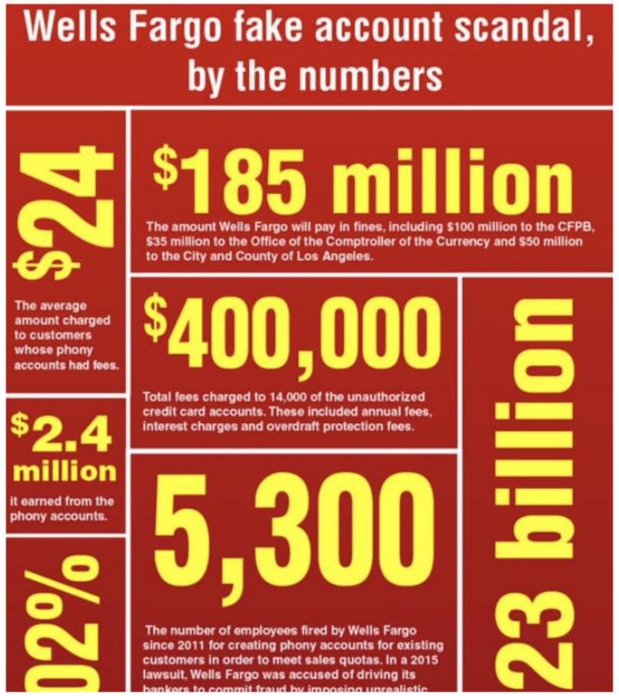
What do you think could've happened to these people & their funds ?

Wells Fargo 3.5 MILLION FAKE accounts - Insider Threat !!!



WireCard faced Accounting Fraud €1.9 BILLION was missing from its balance sheet !!!

JPMorgan encountered Security Breach - compromised 76 Million Accounts !!!



Wells Fargo fake account scandal, by the numbers

$24
The average amount charged to customers whose phony accounts had fees.

$185 million
The amount Wells Fargo will pay in fines, including $100 million to the CFPB, $35 million to the Office of the Comptroller of the Currency and $50 million to the City and County of Los Angeles.

$400,000
Total fees charged to 14,000 of the unauthorized credit card accounts. These included annual fees, interest charges and overdraft protection fees.

$2.4 million
it earned from the phony accounts.

5,300
The number of employees fired by Wells Fargo since 2011 for creating phony accounts for existing customers in order to meet sales quotas. In a 2015 lawsuit, Wells Fargo was accused of driving its bankers to commit fraud by imposing unrealistic

02%

23 billion

Subsequent investigations that revealed the root cause for these threats :

1. Failed to adequately monitor.

2. Unable to Report suspicious transactions.

3. Allowed illicit funds to pass through its systems!!!

4. Failed to detect fraud and non-transparent transactions.

5. Failed to trace the origin and purpose of the funds.

Financial Crime

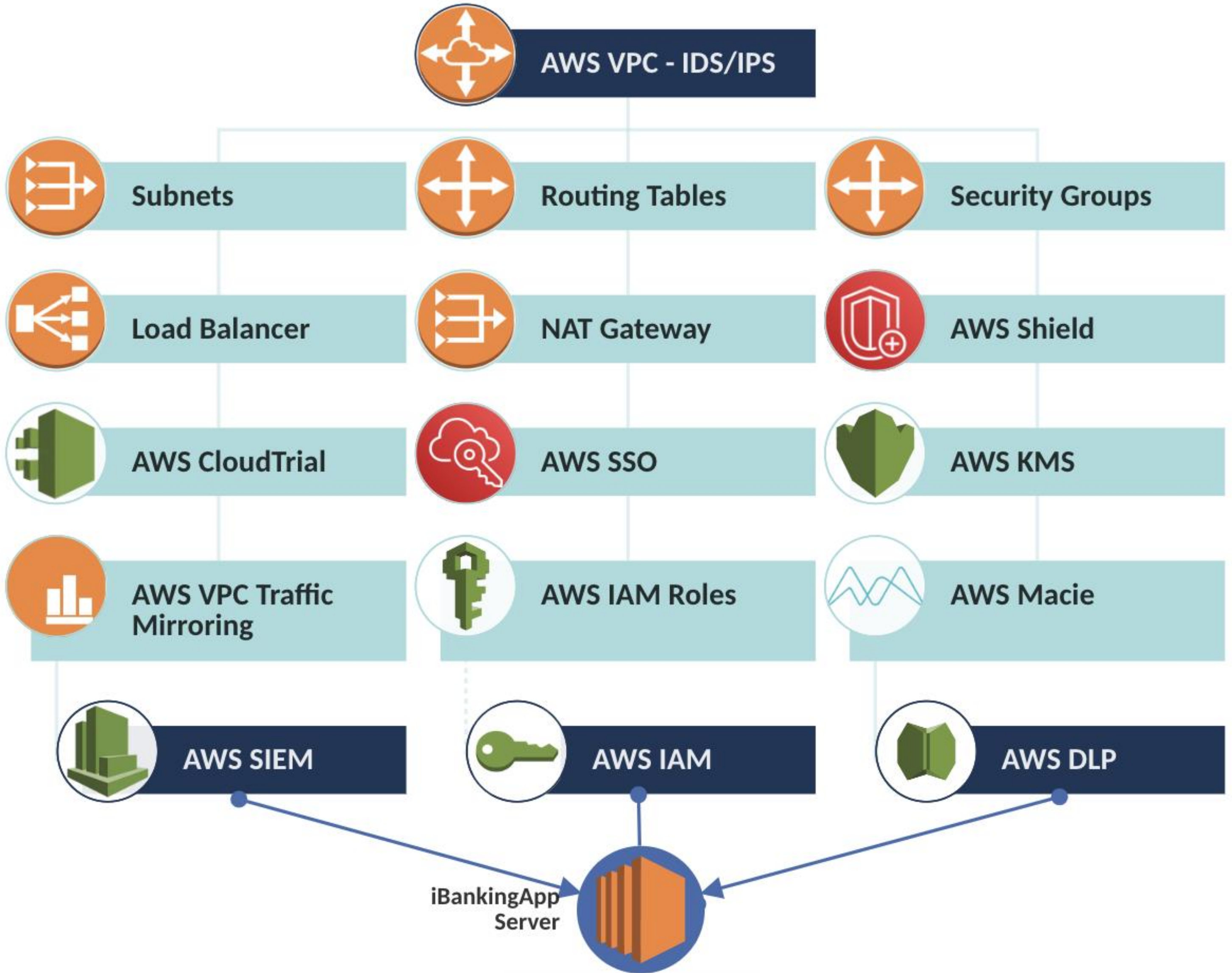
is a broad term that encompasses a variety of ***illegal activities*** that are often associated ***insider threat*** , ***external threat*** & ***security breach***.

Role of an SRE in Fighting Financial Crime

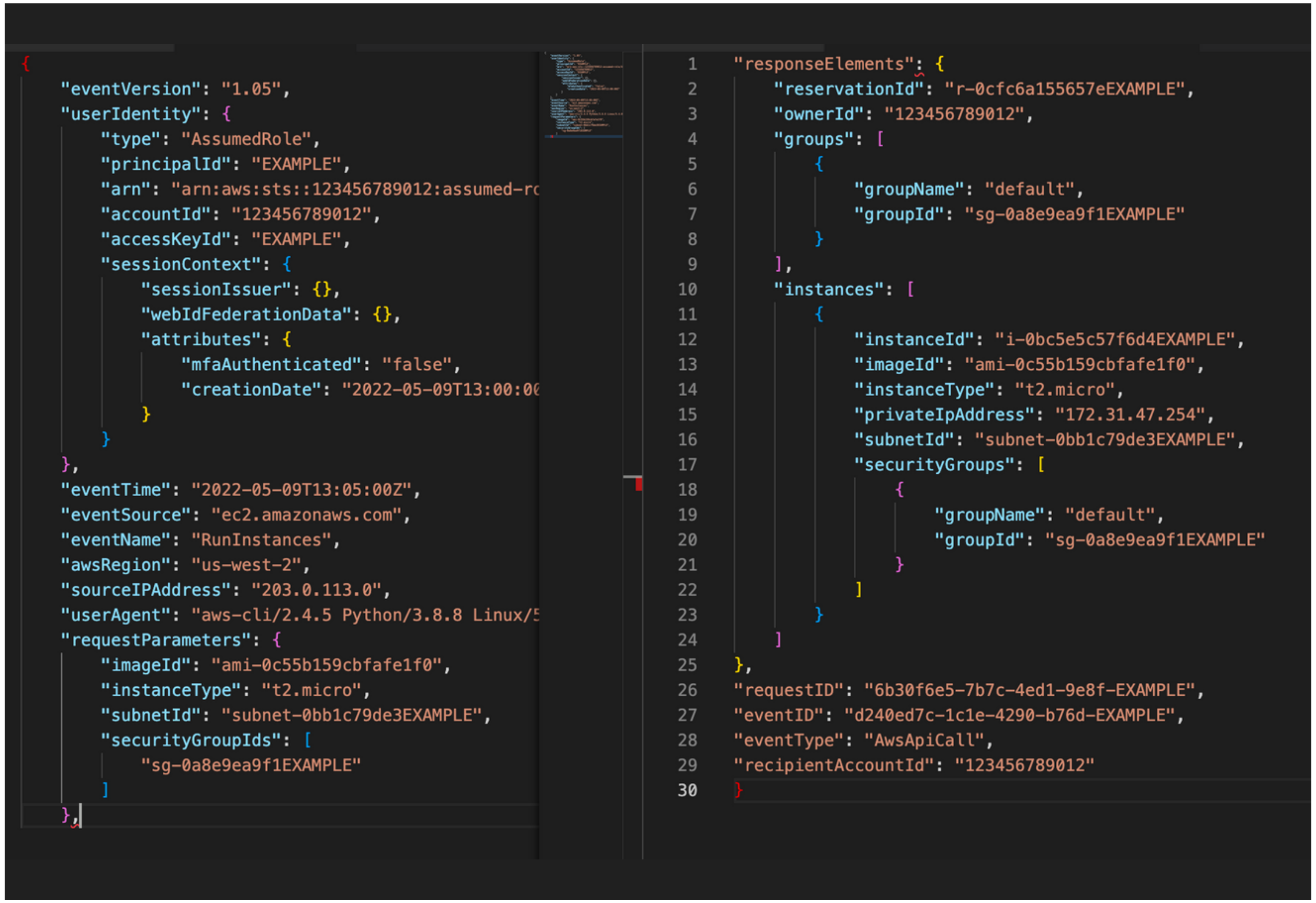
How do you implement these tools in Cloud to build a Secure Infrastructure ?

aws
iBankingApp
DLP
IAM
SIEM
IDS/IPS
? ? ?

Architecture utilises the multi-tier approach to help us identify the patterns of potential Vulnerabilities & threats by analysing the logs from these AWS equivalent services.



CloudTrail Logs - SIEM & IAM



```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-rc
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2022-05-09T13:00:0(
            }
        }
    },
    "eventTime": "2022-05-09T13:05:00Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "RunInstances",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.4.5 Python/3.8.8 Linux/5
    "requestParameters": {
        "imageId": "ami-0c55b159cbfafe1f0",
        "instanceType": "t2.micro",
        "subnetId": "subnet-0bb1c79de3EXAMPLE",
        "securityGroupIds": [
            "sg-0a8e9ea9f1EXAMPLE"
        ]
    },
```

```
"responseElements": {
    "reservationId": "r-0cfc6a155657eEXAMPLE",
    "ownerId": "123456789012",
    "groups": [
        {
            "groupName": "default",
            "groupId": "sg-0a8e9ea9f1EXAMPLE"
        }
    ],
    "instances": [
        {
            "instanceId": "i-0bc5e5c57f6d4EXAMPLE",
            "imageId": "ami-0c55b159cbfafe1f0",
            "instanceType": "t2.micro",
            "privateIpAddress": "172.31.47.254",
            "subnetId": "subnet-0bb1c79de3EXAMPLE",
            "securityGroups": [
                {
                    "groupName": "default",
                    "groupId": "sg-0a8e9ea9f1EXAMPLE"
                }
            ]
        }
    ]
},
"requestID": "6b30f6e5-7b7c-4ed1-9e8f-EXAMPLE",
"eventID": "d240ed7c-1c1e-4290-b76d-EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

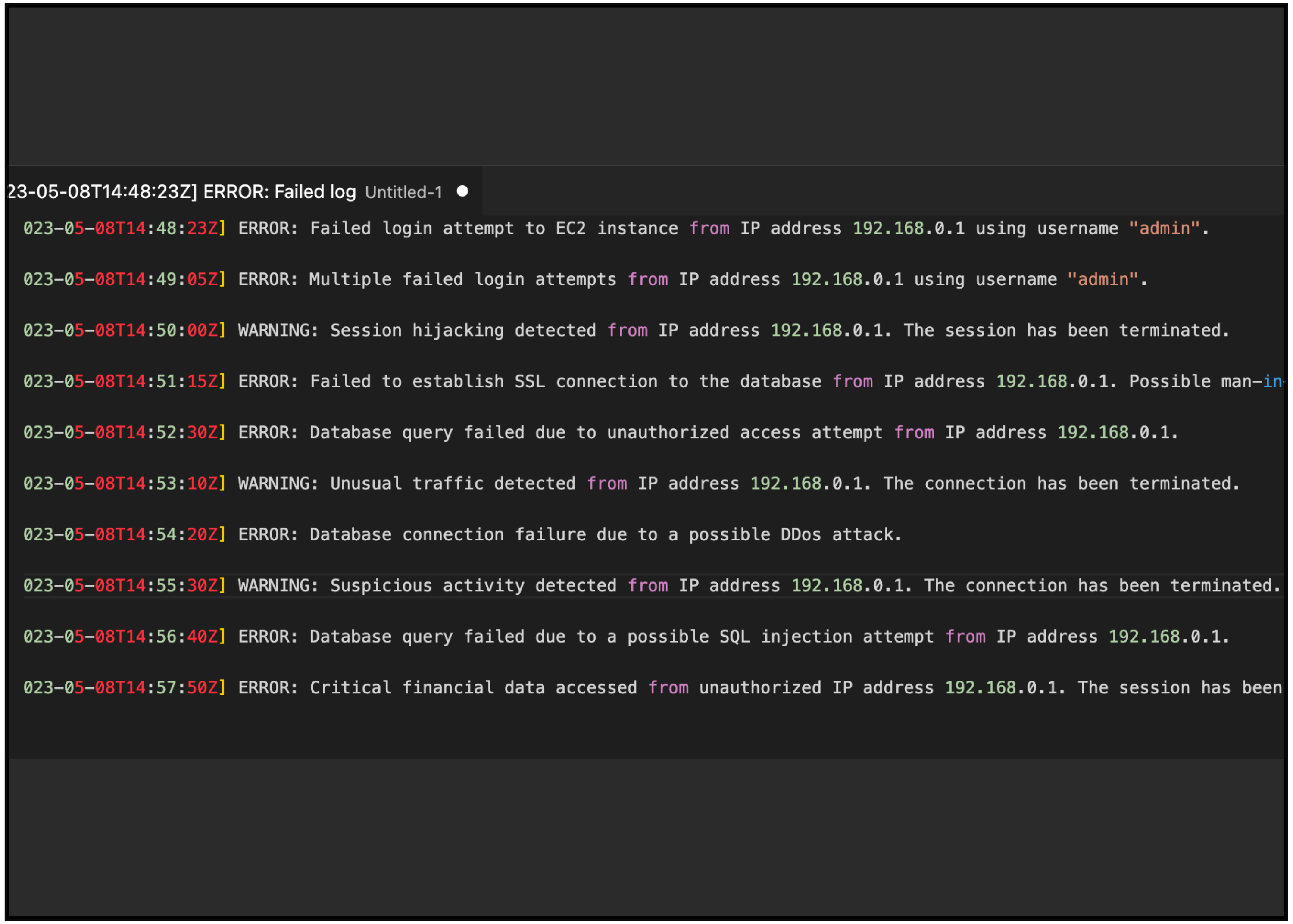
Scenario - Insider Threat Unauthorised employee attempted to login to Server :

The **event** was **triggered by** an **"AssumedRole" identity type**, which would have had a **set of permissions** that allowed it **to run the EC2 instances**, which is the **"RunInstances" event** mentioned in the logs.

What is an "AssumedRole" identity type ?

It's a security mechanism that **allows a user to assume a specific set of permissions** to access AWS resources that **they wouldn't normally have access to.**

This is **done by Temporarily granting users** to **access application** by **providing Access Key & Secret Key** that **they use to make API request** to AWS services.



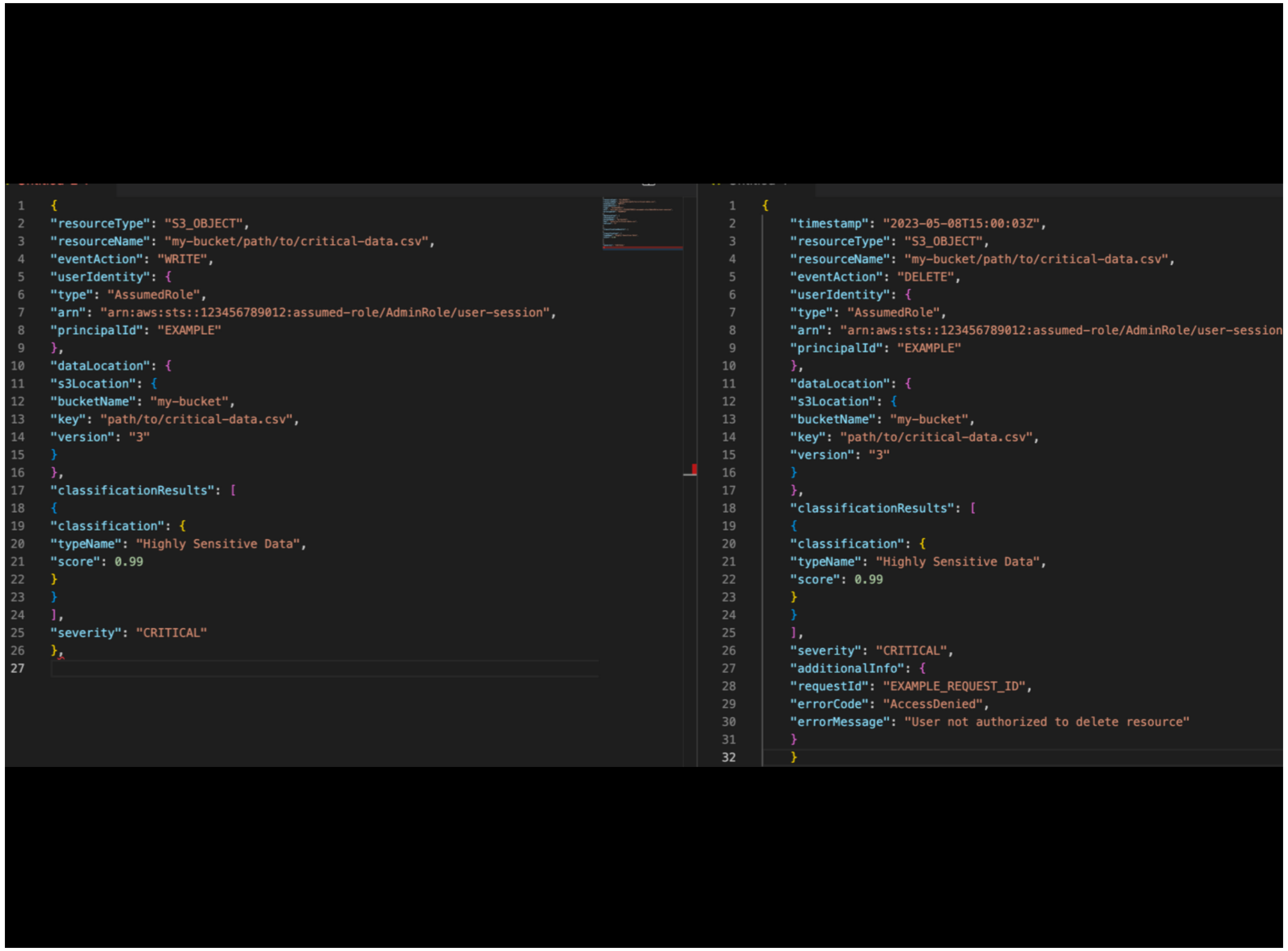
```
23-05-08T14:48:23Z] ERROR: Failed log Untitled-1 ●
023-05-08T14:48:23Z] ERROR: Failed login attempt to EC2 instance from IP address 192.168.0.1 using username "admin".

023-05-08T14:49:05Z] ERROR: Multiple failed login attempts from IP address 192.168.0.1 using username "admin".

023-05-08T14:50:00Z] WARNING: Session hijacking detected from IP address 192.168.0.1. The session has been terminated.

023-05-08T14:51:15Z] ERROR: Failed to establish SSL connection to the database from IP address 192.168.0.1. Possible man-in

023-05-08T14:52:30Z] ERROR: Database query failed due to unauthorized access attempt from IP address 192.168.0.1.

023-05-08T14:53:10Z] WARNING: Unusual traffic detected from IP address 192.168.0.1. The connection has been terminated.

023-05-08T14:54:20Z] ERROR: Database connection failure due to a possible DDos attack.

023-05-08T14:55:30Z] WARNING: Suspicious activity detected from IP address 192.168.0.1. The connection has been terminated.

023-05-08T14:56:40Z] ERROR: Database query failed due to a possible SQL injection attempt from IP address 192.168.0.1.

023-05-08T14:57:50Z] ERROR: Critical financial data accessed from unauthorized IP address 192.168.0.1. The session has been
```

CloudWatch Logs : IDS/IPS

Scenario - ExternalThreat : Hacker attempted session hijacking / DDos Attack.

1. **CloudWatch records** various types of **errors and warnings** related to **login attempts from a specific IP address**.

2. During a **DDoS attack**, **CloudFront leverages** its distributed **edge location network** to **distribute** and handle **incoming traffic** across **multiple edge locations**. This **distribution helps** absorb and **mitigate the impact of the attack** by **spreading the load** and **preventing a single point of failure.**



```
{
"resourceType": "S3_OBJECT",
"resourceName": "my-bucket/path/to/critical-data.csv",
"eventAction": "WRITE",
"userIdentity": {
"type": "AssumedRole",
"arn": "arn:aws:sts::123456789012:assumed-role/AdminRole/user-session",
"principalId": "EXAMPLE"
},
"dataLocation": {
"s3Location": {
"bucketName": "my-bucket",
"key": "path/to/critical-data.csv",
"version": "3"
}
},
"classificationResults": [
{
"classification": {
"typeName": "Highly Sensitive Data",
"score": 0.99
}
}
],
"severity": "CRITICAL"
},
```

```
{
    "timestamp": "2023-05-08T15:00:03Z",
    "resourceType": "S3_OBJECT",
    "resourceName": "my-bucket/path/to/critical-data.csv",
    "eventAction": "DELETE",
    "userIdentity": {
    "type": "AssumedRole",
    "arn": "arn:aws:sts::123456789012:assumed-role/AdminRole/user-session
    "principalId": "EXAMPLE"
    },
    "dataLocation": {
    "s3Location": {
    "bucketName": "my-bucket",
    "key": "path/to/critical-data.csv",
    "version": "3"
    }
    },
    "classificationResults": [
    {
    "classification": {
    "typeName": "Highly Sensitive Data",
    "score": 0.99
    }
    }
    ],
    "severity": "CRITICAL",
    "additionalInfo": {
    "requestId": "EXAMPLE_REQUEST_ID",
    "errorCode": "AccessDenied",
    "errorMessage": "User not authorized to delete resource"
    }
    }
```

Amazon Macie Logs : DLP

Scenario - Accounting Fraud

An attempts to Modify sensitive data.

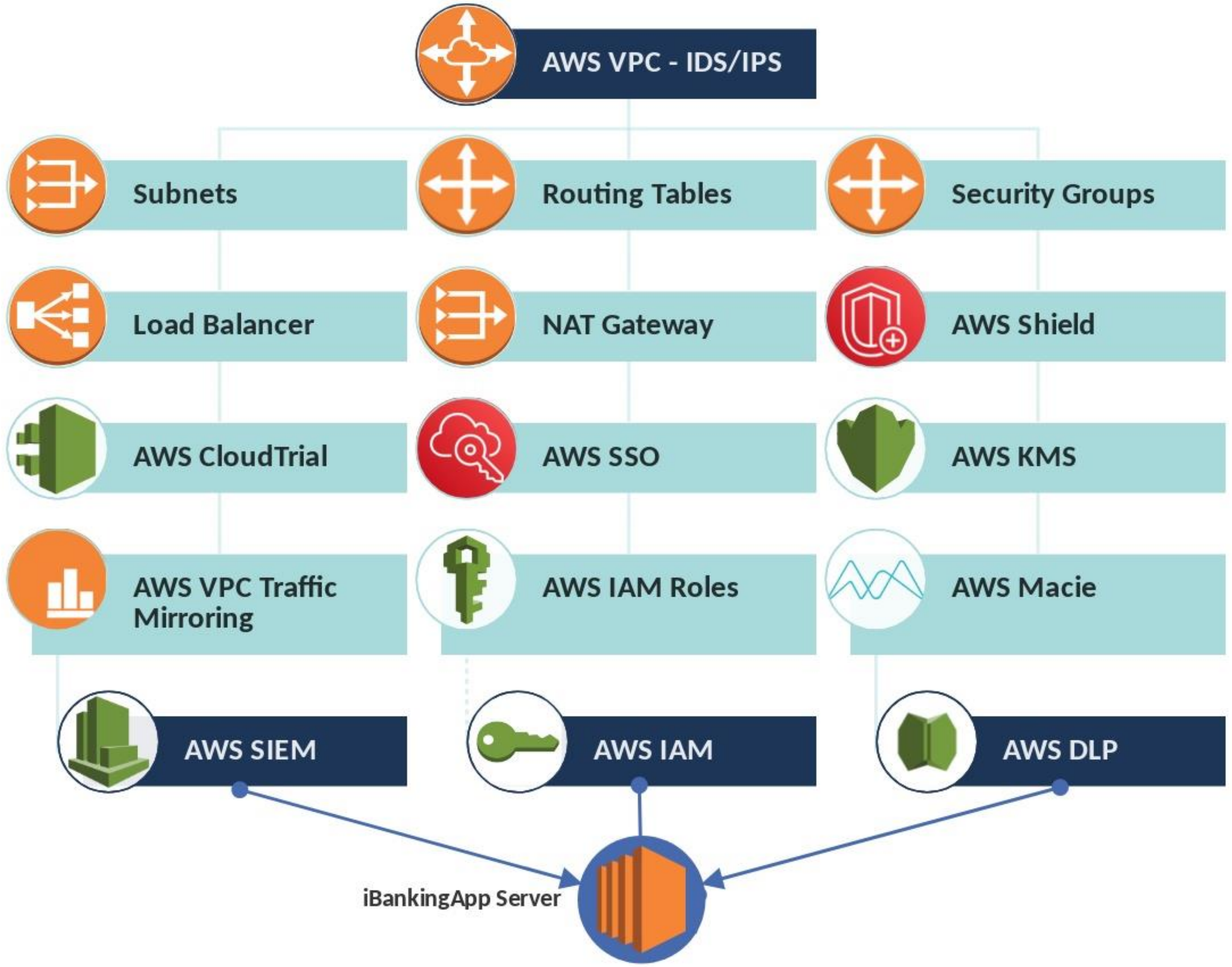
1. First log entry:
- **"eventAction"**: **WRITE** - Indicates an **attempt to modify the file.**
- "dataLocation" field provides details about the specific version of the file accessed.

2. Second log entry:
- **"eventAction"**: **DELETE** - Indicates an **attempt to delete the file.**
- "dataLocation" field provides details about the specific version of the deleted file.

3. Classification and severity:
- **"classificationResults"** field : **"Highly Sensitive Data"**
- "severity": CRITICAL

4. User information:
- **"userIdentity"** field : User **assumed AdminRole**.
- **"ErrorCode"** : **"AccessDenied"**
- **"ErrorMessage"** : ***User was not authorised to delete the resource.***

Architecture utilises the multi-tier approach to **help us identify the patterns of potential Vulnerabilities & Threats by analysing the logs** from these AWS equivalent Tools..

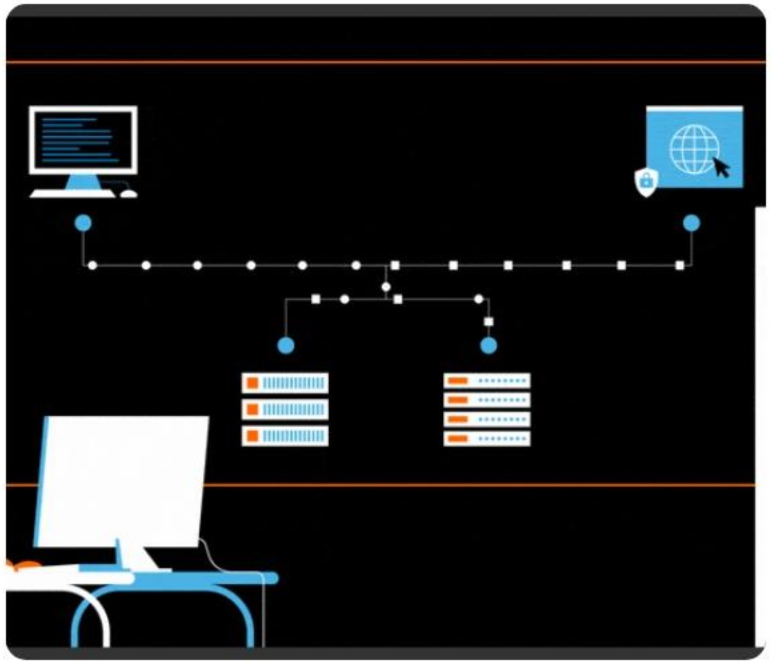


What is Cap's Shield & Thor's Mjolnir for an SRE ?

Aren't We the Worthy Ones ⍰

Data Analytics and Automation

Unleashing the AWS capabilities to make our job easier



Data storage

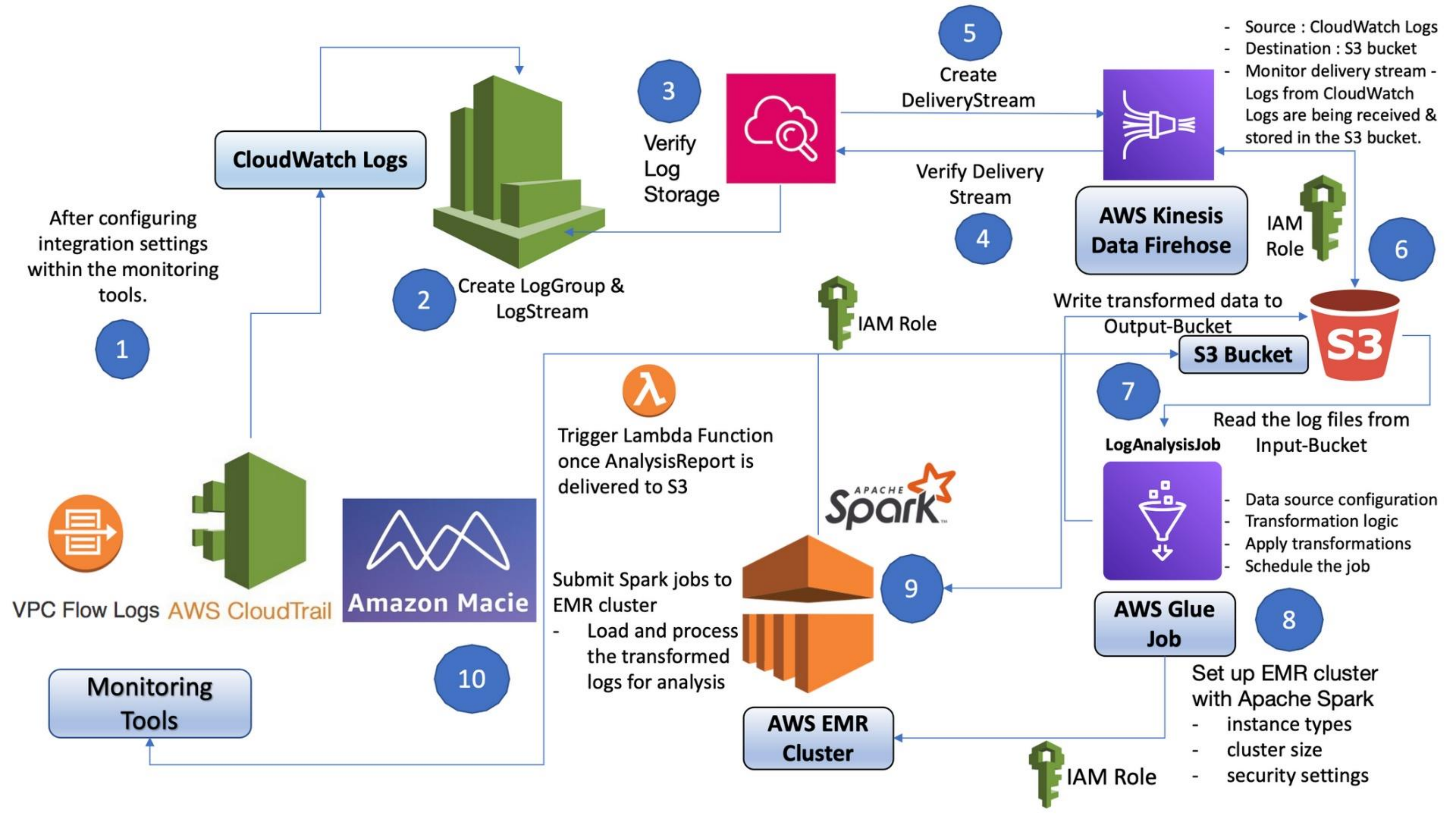
Can be used to store large amounts of dataset, including vpc logs, Maice events & CloudTrial logs.

Data analysis

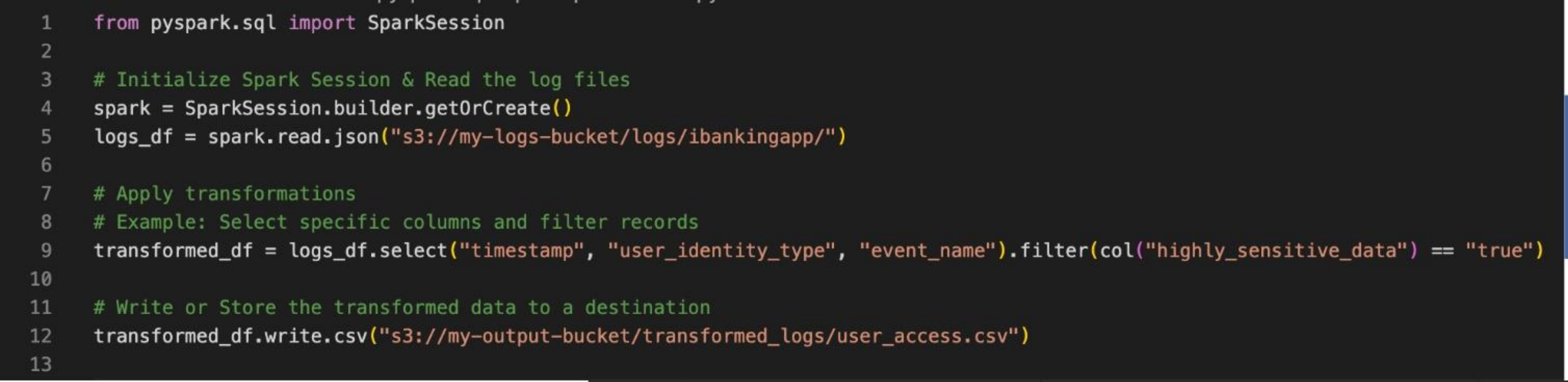
Use cluster algorithms to group similar log entries together and identify outliers that could be threat or Vulnerabilities.

Automation

Simple schedulers and cron jobs will do the magic.

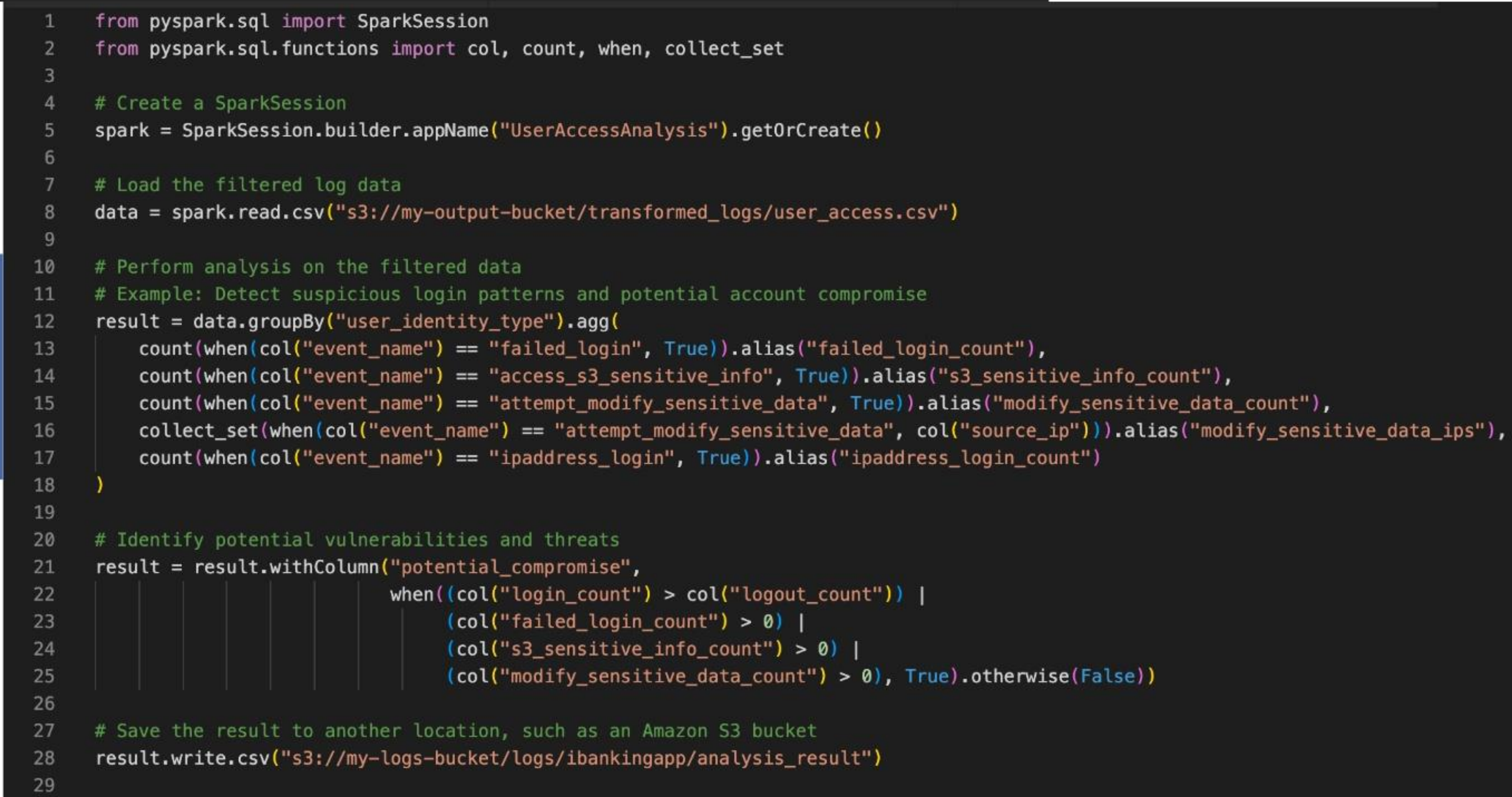

Source : CloudWatch Logs
Destination : S3 bucket
Monitor delivery stream - Logs from CloudWatch Logs are being received & stored in the S3 bucket.
5
Create DeliveryStream
3
Verify Log Storage
CloudWatch Logs
Verify Delivery Stream
4
AWS Kinesis Data Firehose
IAM Role
6
After configuring integration settings within the monitoring tools.
1
2
Create LogGroup & LogStream
IAM Role
Write transformed data to Output-Bucket
S3 Bucket
S3
7
Read the log files from Input-Bucket
LogAnalysisJob
Trigger Lambda Function once AnalysisReport is delivered to S3
Apache Spark
Data source configuration
Transformation logic
Apply transformations
Schedule the job
VPC Flow Logs
AWS CloudTrail
Amazon Macie
Submit Spark jobs to EMR cluster
- Load and process the transformed logs for analysis
9
AWS Glue Job
8
10
Set up EMR cluster with Apache Spark
- instance types
- cluster size
- security settings
Monitoring Tools
AWS EMR Cluster
IAM Role

AWS Glue - LogAnalysisJob

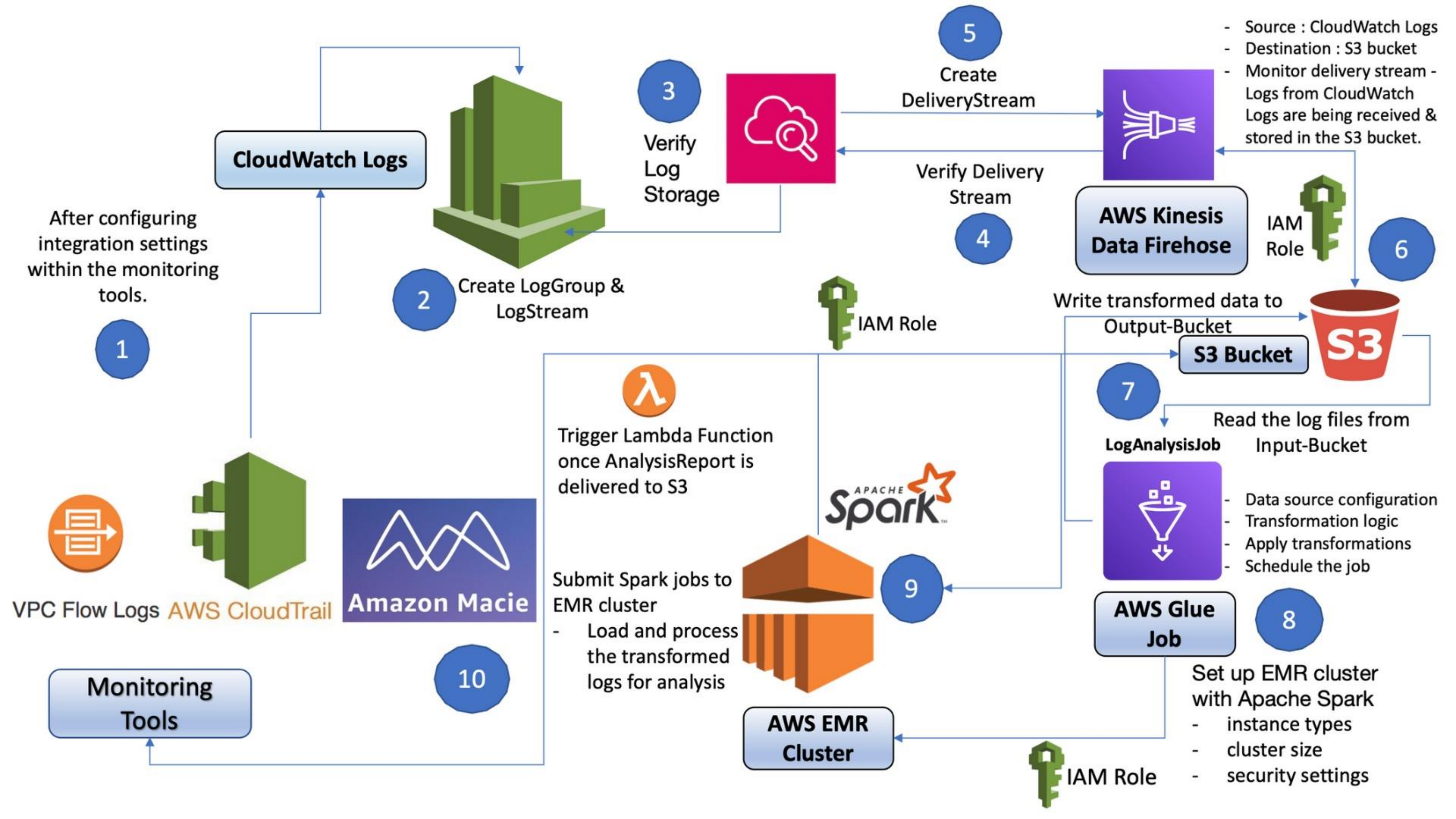


```python
from pyspark.sql import SparkSession

# Initialize Spark Session & Read the log files
spark = SparkSession.builder.getOrCreate()
logs_df = spark.read.json("s3://my-logs-bucket/logs/ibankingapp/")

# Apply transformations
# Example: Select specific columns and filter records
transformed_df = logs_df.select("timestamp", "user_identity_type", "event_name").filter(col("highly_sensitive_data") == "true")

# Write or Store the transformed data to a destination
transformed_df.write.csv("s3://my-output-bucket/transformed_logs/user_access.csv")

```

AWS EMR - Spark job code using Apache Spark APIs



```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, count, when, collect_set

# Create a SparkSession
spark = SparkSession.builder.appName("UserAccessAnalysis").getOrCreate()

# Load the filtered log data
data = spark.read.csv("s3://my-output-bucket/transformed_logs/user_access.csv")

# Perform analysis on the filtered data
# Example: Detect suspicious login patterns and potential account compromise
result = data.groupBy("user_identity_type").agg(
    count(when(col("event_name") == "failed_login", True)).alias("failed_login_count"),
    count(when(col("event_name") == "access_s3_sensitive_info", True)).alias("s3_sensitive_info_count"),
    count(when(col("event_name") == "attempt_modify_sensitive_data", True)).alias("modify_sensitive_data_count"),
    collect_set(when(col("event_name") == "attempt_modify_sensitive_data", col("source_ip"))).alias("modify_sensitive_data_ips"),
    count(when(col("event_name") == "ipaddress_login", True)).alias("ipaddress_login_count")
)

# Identify potential vulnerabilities and threats
result = result.withColumn("potential_compromise",
                           when((col("login_count") > col("logout_count")) |
                                (col("failed_login_count") > 0) |
                                (col("s3_sensitive_info_count") > 0) |
                                (col("modify_sensitive_data_count") > 0), True).otherwise(False))

# Save the result to another location, such as an Amazon S3 bucket
result.write.csv("s3://my-logs-bucket/logs/ibankingapp/analysis_result")

```



5
Create DeliveryStream
- Source : CloudWatch Logs
- Destination : S3 bucket
- Monitor delivery stream - Logs from CloudWatch Logs are being received & stored in the S3 bucket.
3
Verify Log Storage
CloudWatch Logs
Verify Delivery Stream
4
AWS Kinesis Data Firehose
IAM Role
6
After configuring integration settings within the monitoring tools.
1
2
Create LogGroup & LogStream
IAM Role
Write transformed data to Output-Bucket
S3 Bucket
S3
7
Read the log files from Input-Bucket
LogAnalysisJob
Trigger Lambda Function once AnalysisReport is delivered to S3
APACHE Spark
- Data source configuration
- Transformation logic
- Apply transformations
- Schedule the job
VPC Flow Logs
AWS CloudTrail
Amazon Macie
Submit Spark jobs to EMR cluster
- Load and process the transformed logs for analysis
9
AWS Glue Job
8
10
Monitoring Tools
Set up EMR cluster with Apache Spark
- instance types
- cluster size
- security settings
AWS EMR Cluster
IAM Role

THANK YOU!

in @linkedin.com/in/anisha-manoharan/