AWS Certified Security – Specialty Exam Guide

Build your cloud security knowledge and expertise as an AWS Certified Security Specialist (SCS-C01)

Stuart Scott

Packt>
www.packt.com

13
Auditing and Governance

Every organization is subjected to a level of governance and compliance, and this governance comes hand in hand with auditing. As an organization, you must be able to trace, track, and record different processes and operational steps to ensure that you as a company are following set guidelines, laws, and regulations. You have to able to prove at any given point that you as an organization are adhering to this level of governance, and this is referred to as auditing. Therefore, auditing provides a means and method of providing evidence to a third-party auditor that you meet specific criteria through implemented controls and processes. Failure to comply with governance controls, depending on the audit itself, can result in financial fines and/or criminal charges.

This chapter will look at some of the methods and AWS services that can play key parts in maintaining this governance and how a level of auditing can be achieved. As a result, we will be focusing on the following:

- What is an audit?
- Understanding AWS Artifact
- Securing AWS CloudTrail
- Understanding your AWS environment through AWS Config
- Maintaining compliance with Amazon Macie

Technical requirements

To follow the examples and demonstrations, you must have access to an AWS environment with elevated privileges to the following:

- AWS Artifact
- AWS CloudTrail
- Amazon S3
- AWS Config
- Amazon Macie

For more information on how to access these privileges, please refer to `Chapter 4`, *Working with Access Policies*.

What is an audit?

There are many different types of audits that a third-party external auditor can assess your organization on, for example, network security, data management, remediation management, and change management are just a few. The auditors may conduct the assessment by collecting evidence in a variety of ways, such as analyzing logs, physical inspection, reviewing procedures, and general inquiries. Before we begin this chapter, let's highlight some common audit compliance programs that you might see.

AWS complies with global compliance programs to meet the needs of its customers. A full breakdown of the compliance programs that they are subjected to and have certification for can be found at `https://aws.amazon.com/compliance/programs/`.

The following are the global compliance programs that AWS adheres to:

	ISO 9001	**ISO 27001**	**ISO 27017**	**ISO 27018**
CSA controls	Global Quality Standard	Security Management Controls	Cloud Specific Controls	Personal Data Protection
Report type	SOC 1: Audit Controls Report	SOC 2: Security, Availability, and Confidentiality Report	SOC 3: General Controls Report	

As you can see, many of these compliance programs focus on specific aspects of an organization's processes, such as ISO 27017, which looks at Cloud Specific Controls, and SOC 2, focusing on Security, Availability, and Confidentiality.

AWS also adheres to many other region-specific compliance programs, such as FedRAMP and HIPAA in the US, G-Cloud (UK) in the United Kingdom, and many more!

In each of these compliance programs, there is a set of audit controls that need to be met. For each control, evidence will need to be seen and collected to ensure that your organization meets the criteria of that control. Once the auditor is satisfied with the evidence, the audit control is classed as met. After each audit, a full report is carried out stating the controls that were passed and the ones that were failed, highlighting any certifications and accreditation achieved as a result of the audit.

Differences businesses, depending on the industry, need to follow different compliance controls, and just because AWS adheres to certain controls does not mean that you as a business running in AWS also meets those controls. You will be assessed on how you are running your infrastructure, processes, security, management, and more, to also ensure that you meet the controls required for your own audits conducted by external auditors.

With this basic understanding of what auditing is all about, let's now go one step further and look at some of the methods and AWS services that can play a key role in achieving this audit. Let's begin with AWS Artifacts.

Understanding AWS Artifact

Unlike other AWS services, AWS Artifact is not a service that you use to create a resource, such as an EC2 instance, a database, or a VPC. Instead, AWS Artifact is an on-demand portal to allow you to view and download AWS security and compliance **reports**, in addition to any online **agreements**. But what are these reports and agreements exactly?

- **Reports**: These are the reports that have been undertaken by external auditors of AWS, who have issued the relevant reports, certifications, accreditations, and other third-party attestations.
- **Agreements**: These allow customers to review and then accept any agreements that have been made with AWS that relate to your own individual account. If your account is a part of an AWS Organization, then all of the agreements for all accounts can be reviewed and accepted here in a central location, simplifying the management of AWS agreements.

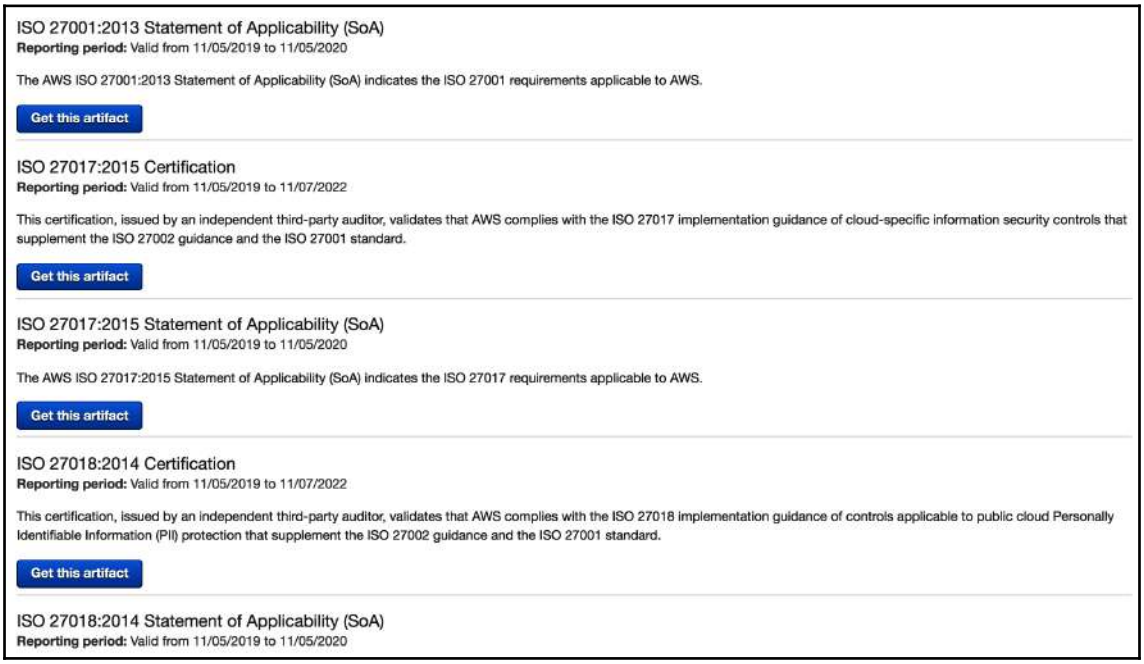
You can access these reports and agreements from the AWS Management Console itself. Let's see how to do it.

Accessing reports and agreements

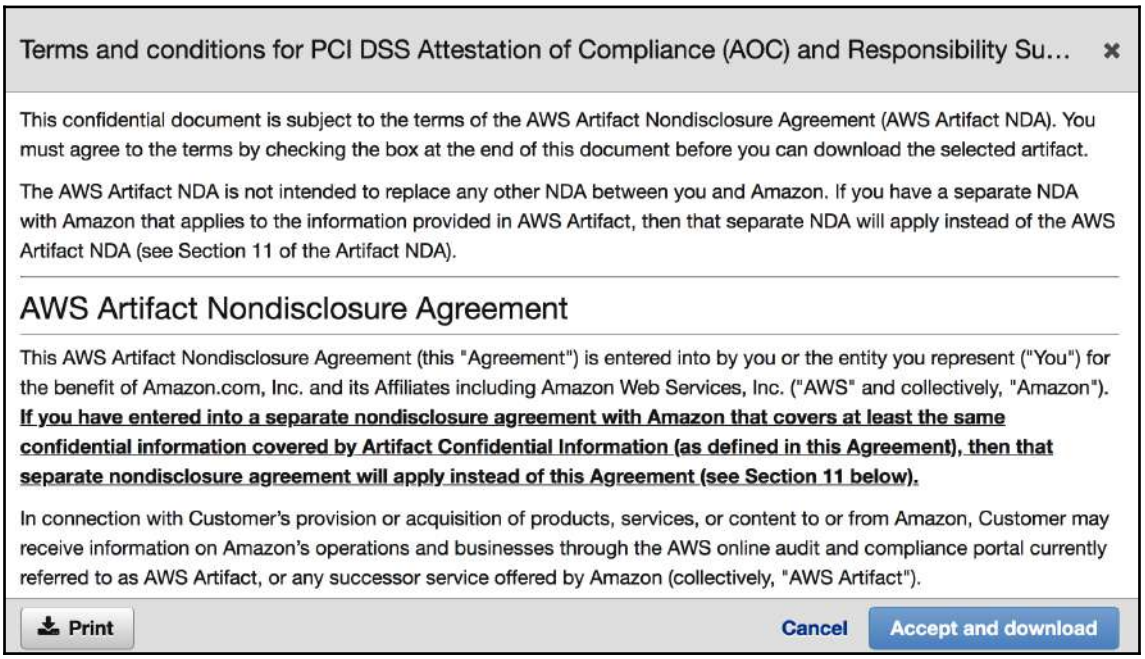
Accessing these reports is simple and you can do it from the Management Console. Follow these steps:

1. Log in to your AWS account and, from the AWS Management Console, click on the **Security, Identity and Compliance** category.
2. Now navigate to the **Artifact** dashboard. You are presented with two options, which help organize Artifact documents, these being **Reports** and **Agreements**:

3. We'll look at **Reports** first, so click on it. Within the **Reports** section, you can find artifacts that relate to specific compliance and security requirements, such as SOC, PCI-DSS, IS27000 series, among many others. Here is a screenshot of the list of artifacts available:

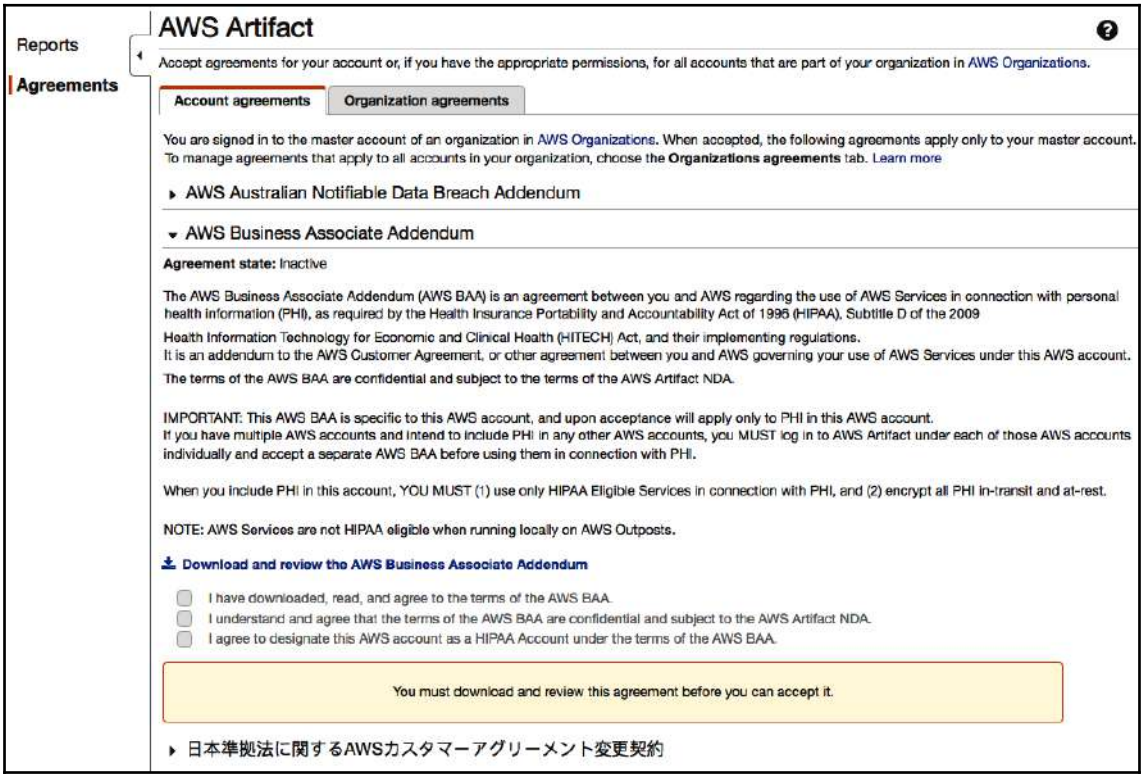


4. As you can see, for each artifact there is a blue button that allows you to **Get this artifact**. To view the artifact and download the document, you must select the blue button and then sign a **non-disclosure agreement** (**NDA**) with AWS relating to the information contained within the artifact. This NDA screen will look something like the following:



5. Once downloaded, these compliance artifacts can then be submitted to your own auditors to help achieve the level of governance and compliance that you need for your infrastructure.

6. Now let's move on to the **Agreements** dashboard. The **Agreements** section of AWS Artifact allows you to review, accept, and manage agreements for your account or organization such as the **Business Associate Addendum** (**BAA**) used for HIPAA compliance:



Account agreements shows any agreements that have been associated directly with your AWS account that you are currently viewing. If your account is the master account in an AWS Organization, then you will be able to view all of the account agreements for all member accounts using **Organization agreements**.

In this section, I explained how AWS Artifact can be used to help you provide reporting and agreement evidence from AWS to your auditors as and when required. This can be used in conjunction with your own evidence and reports gathered across your own infrastructure, which is what we will move on to next, starting with AWS CloudTrail.

Securing AWS using CloudTrail

In the previous chapter, I explained how you can create a new trail and configure logging mechanisms for AWS CloudTrail, in addition to diving into detail about the information captured, which provides great insight from an auditing perspective. However, here I just want to look at and highlight some of the best practices from a security perspective when configuring CloudTrail.

As we know, AWS CloudTrail is a great service to track and record all API activity on your accounts, which, as expected, can contain some very sensitive information that you would want to restrict access to. CloudTrail stores its logs in Amazon S3 by default, but as discussed previously, these can also be sent to CloudWatch Logs.

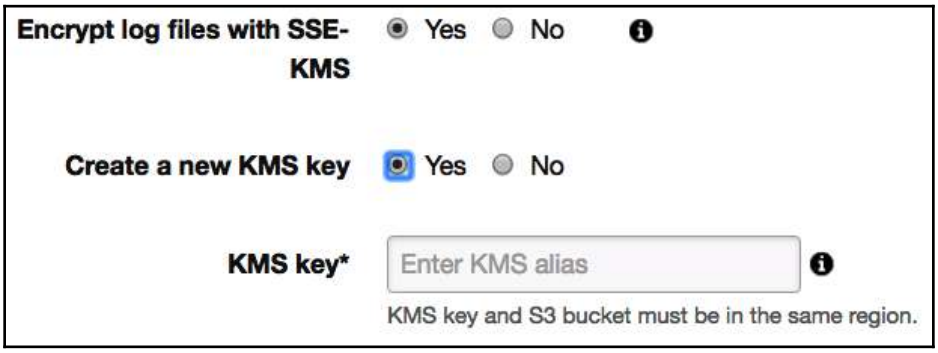
You may have heard over the past few years a lot of emphasis on Amazon S3 security controls, largely due to a string of data breaches where sensitive information had been exposed and was accessible to public users with malicious intent. However, much of this exposure was simply down to a lack of understanding from users of S3 about utilizing the available security controls that Amazon S3 offers, rather than a security design fault in the service itself. As a result, extra care should always be taken when storing sensitive information in any cloud storage. Thankfully, AWS has implemented further security enhancements, and in addition to an increase in user awareness of the service, it has helped prevent such accidents from happening since.

With this in mind, extra security controls on your CloudTrail data should be implemented. For example, you should always choose **Encrypt log files with SSE-KMS** in addition to selecting **Enable log file validation**. Let's see why these selections are important over the next two subsections.

The screenshots in the next section are taken from the configuration screen of an AWS CloudTrail trail. For more information on how to set up a trail, please see `Chapter 12`, *Implementing Logging Mechanisms*.

Encrypting log files with SSE-KMS

I will cover the full extent of the different S3 encryption options, in addition to how KMS works, in `Chapter 16`, *Managing Key Infrastructure*. However, at this stage, all we need to be concerned with is that it's possible to encrypt our CloudTrail logs files using either an existing or new KMS key. This is a very easy feature to enable as it's simply a checkbox and a KMS key selection:



By doing so, all of your CloudTrail log data at rest will be encrypted unless you have access to the `kms:decrypt` action for the selected KMS key, in addition to access to the S3 bucket where your logs are stored. Adding this level of encryption ensures that only someone with access to decrypt the file can access the sensitive information that can be found within your log files. Due to the amount of information that can be contained in your CloudTrail log files, you will want to restrict access to them as much as possible, and this level of restriction might even be required to pass an audit control during an assessment.

Enabling log file validation

This checkbox is especially useful when you need to perform some forensic investigation into a security threat as it ensures that your log files have not been tampered with or modified at all from when they were written to your bucket in Amazon S3. To enforce this validation, CloudTrail uses algorithms such as SHA-256 for hashing and SHA-256 with RSA for digital signing:

Every time a new log file is delivered to S3 with validation enabled, CloudTrail will create a hash for it. In addition to this, and once an hour, CloudTrail will also create another file called a digest file that references each and every log file that was delivered within that hour, along with the associated hash. These digest files are signed using a private key of a public/private key pair used by CloudTrail for that region. Using the associated public key, you can then validate the digest files, which are stored in the same S3 bucket as your logs but in a different folder.

If you have read access to the bucket and you have *not* moved the files from their original bucket, then you can perform the validation by entering the following command using the AWS CLI:

```
aws cloudtrail validate-logs --trail-arn <trailARN> --start-time <start-time>
```

Additionally, you can add the following options to this command:

```
[--end-time <end-time>] [--s3-bucket <bucket-name>] [--s3-prefix <prefix>] [--verbose]
```

These allow you to narrow your selection using specific parameters such as a set bucket name, prefixes, and time constraints.

For a full listing of all the results and what they mean, please refer to the table within the AWS documentation at: `https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-cli.html`.

In addition to the aforementioned controls, you should also limit and control access to the S3 bucket in which the CloudTrail logs are stored. In addition to IAM permissions, limit access using bucket policies to only those who require access due to the sensitive information that can be found in the logs, which could, if accessed by a malicious user, provide a lot of information about your infrastructure, and potentially help to identify weak spots in your environment that could be used against you.

Auditing and governance do not simply require you to implement a level of logging, such as you can with AWS CloudTrail, but some audits will require you to demonstrate that you are restricting access to these logs as to ensure they do not become compromised and used as a method of gaining intelligence of intrusion. So, this section will be extremely helpful as you try and enforce additional control of your AWS CloudTrail logs.

Next, I want to move focus on to another service, AWS Config, which works in conjunction with AWS CloudTrail.

Understanding your AWS environment through AWS Config

With the number of services rising each year in AWS (currently at 168 services at the time of writing), it's easy to comprehend how difficult it can be to have an understanding of what resources you might be running within your environment. How can you keep up with what instances you have running and where, what are they running, and the resources still needed? You might be running infrastructure that's no longer required that got overlooked in among the thousands of virtual devices that are in production.

With the huge network of resources running within your account, do you have a clear understanding of which resource is connected to which? What ENI is connected to which instance? Which subnet is that instance running in? Which subnets are connected to which VPCs? Do you have a logical mapping of infrastructure that quickly and easily allows you to identify a blast radius should an incident occur, or visibility into resource dependencies should you change your configuration?

On top of that, do you know their current state of configuration? Are you certain they are running the latest patches, or is there a chance that some of your infrastructure is exposed and has been left vulnerable to potential security threats?

If someone makes a change to your infrastructure and environment, do you have an accurate record of that change, what changed, and when it changed?

Going back to compliance, how can you be assured that the resources that you are deploying and keeping meet compliance needs as dictated by both your internal and external controls and processes?

Answers to all of the above questions are generally required when performing audits, but gaining this information can be very cumbersome in traditional IT deployments, let alone cloud environments, which by their very nature are far more dynamic and are subject to a far higher rate of change. However, AWS is aware of these audit and compliance requirements and has an AWS service called AWS Config to help you address many of these questions in an automated, auditable, and compliant way.

For a comprehensive walk through of how to configure AWS Config, please see: `https://docs.aws.amazon.com/config/latest/developerguide/gs-console.html`. In this book, I want to focus more on the different components of AWS Config, and how they operate to help you understand how the service works and provides a level of auditing and governance checks. So, once you are set up, come back here to explore the various components of AWS Config.

To understand how AWS Config can help you achieve these results, let me explain some of the components of the service, which include the following:

- Configuration items
- Configuration streams
- Configuration history
- Configuration snapshots
- Configuration recorder
- Config rules
- Resource relationships
- Config role

Let's begin with our first component – **configuration items** (**CIs**).

Configuration items

This is a fundamental element of AWS Config and is essentially a JSON file containing point-in-time snapshot information on the configuration data of attributes of a specific AWS resource within your environment that is supported by AWS Config (a full list of supported resources can be found at: `https://docs.aws.amazon.com/config/latest/developerguide/resource-config-reference.html`).

These attributes include its current configuration, any direct relationships the resource has with other resources, metadata, and events. A new CI is updated every time a change is made on that resource, for example, when a create, update, or delete API call is made against the resource.

To understand more about the construct of a CI, a table containing a list of components of a configuration item can be found within the AWS documentation at: `https://docs.aws.amazon.com/config/latest/developerguide/config-item-table.html`. Let's go over the components one by one:

- The **Metadata** section contains information and data about the configuration item itself.
- The **Attributes** section focuses on the data of the actual resource that the CI relates to.

- The **Relationship** section holds data related to any connected resource, for example, if the CI related to a subnet, the relationship could contain data related to the associated VPC the subnet was a part of.
- The **Current Configuration**, as the table explains, shows the same information that would be generated if you were to perform a `describe` or `list` API call made by the AWS CLI.

These CIs are effectively building blocks of AWS Config and are used by many other components of the service. Let's continue to see how these work together.

Configuration streams

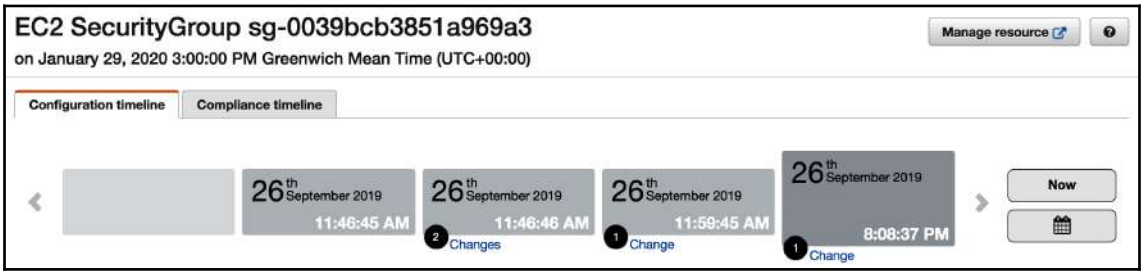
When a change against a resource occurs in your environment, and as a result a new CI is created, then the CI is automatically added to a configuration stream, which is essentially an SNS topic. During the configuration of AWS Config, you can specify the SNS topic to be used for your stream:

This enables you to monitor the stream and customize notifications for changes occurring to resources, helping you to identify potential issues or security incidents that are unexpected.

Configuration history

This is especially useful when it comes to audits and provides a complete history of all the changes made to a resource. By collating the CIs for a resource, AWS Config is able to assemble a history of modifications to that resource. The history of your resource can be accessed via the AWS CLI or via the AWS Management Console as a timeline of events. Also, as a part of the process, AWS Config will store a configuration history file of each resource type in an S3 bucket that is selected during the configuration of AWS Config.

Here, you can see the configuration history of an EC2 security group. It shows the date and time of any changes to the resource:



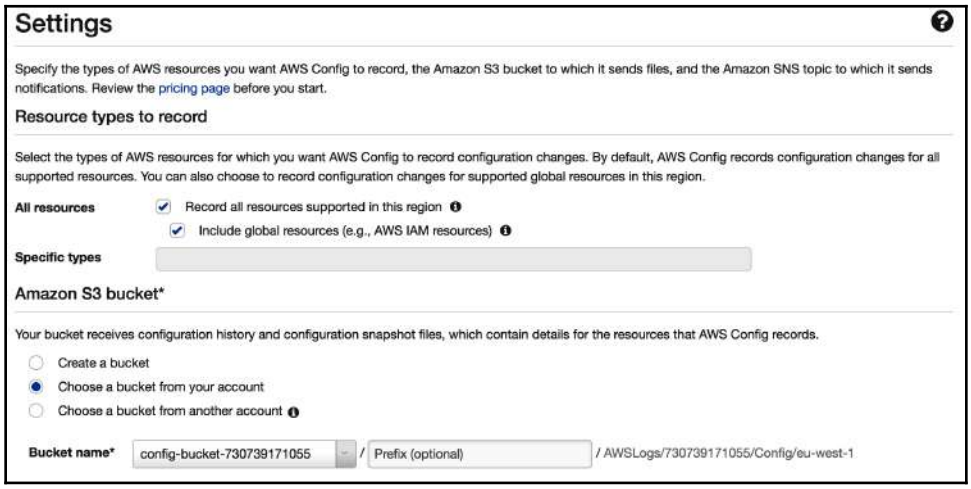
Using the AWS Management Console, you can select these changes and dive deeper to understand what element changed. Also, following a security incident or an outage, this history can be very useful to determine the timeline of events that led to the incident and can help you resolve it quickly and effectively.

Configuration snapshot

Again, using the building blocks of AWS Config, new CIs will be created allowing for a configuration snapshot to be constructed to get a point-in-time image of your AWS environment of all supported AWS resources with AWS Config in a particular region. This snapshot can be initiated by running the AWS CLI `deliver-config-snapshot` command and the results will be sent to your predefined Amazon S3 bucket.

Configuration recorder

You can think of the configuration recorder as the on and off switch for the AWS Config service. You must first enable the configuration recorder before the service can start creating your **configuration items** (**CIs**). When you first configure AWS Config, the configuration recorder is automatically started, but once started, you can stop and re-enable it at a later date:



This shows the initial configuration screen and allows you to select the resource types that you want AWS Config to record. If you uncheck the **Record all resources supported in the region**, then you will be able to select from a drop-down list of **Specific types**, an example of which can be seen here:

With your chosen resources selected and the destination S3 bucket selected to store your configuration history and snapshot files, the configuration recorder can begin resource changes.

AWS Config rules

From a compliance perspective, AWS Config rules are a great feature and should be implemented whenever you use AWS Config. Backed by AWS Lambda functions performing simple logic, Config rules automatically monitor your resources to ensure they are meeting specific compliance controls that you might need to introduce within your AWS environment. If a resource is found not to be compliant, you will be notified via SNS and the configuration stream, allowing you to take corrective action.

With Config rules, you can enforce a consistent deployment and configuration approach, ensuring all resource types are following set criteria, regardless of who or when the resource was deployed.

There are two types of Config rules available: those that can be *custom defined*, and those that are *predefined and managed by AWS*. These AWS rules are ready and available to use to save you having to create your own from scratch.

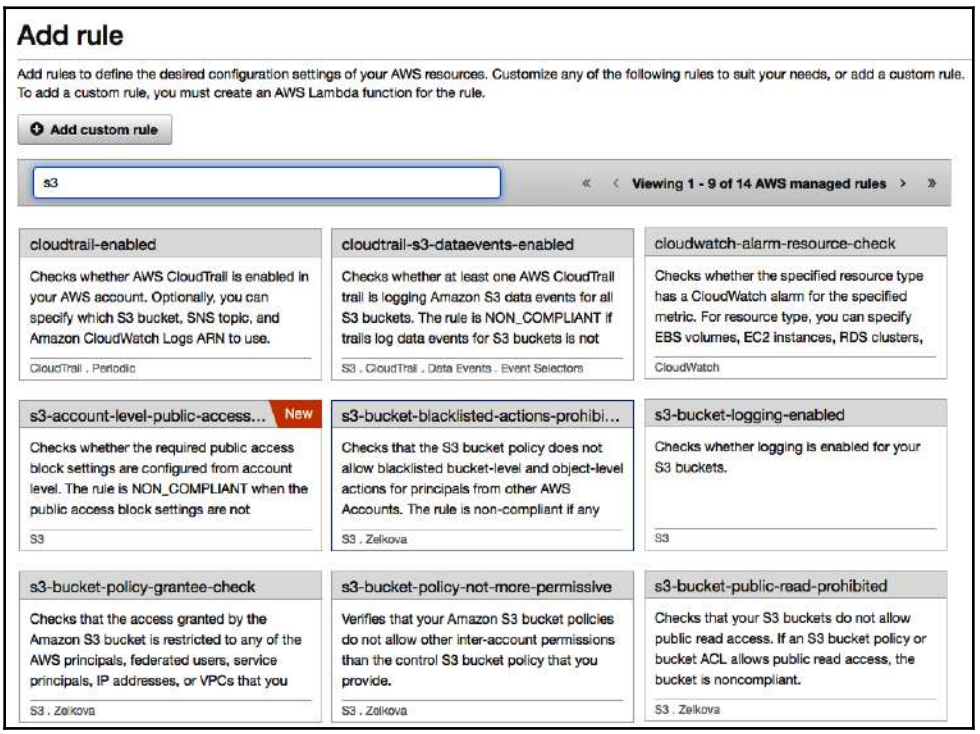
Let's take a look at how you can set up AWS Config rules to help with the compliance of your infrastructure.

Please note, you must have already set up and configured AWS Config (see the *Configuration recorder* section for more information).

1. Log in to the AWS Management Console and navigate to AWS Config under the **Management and Governance** category.
2. From the menu on the left-hand side, select **Rules**:

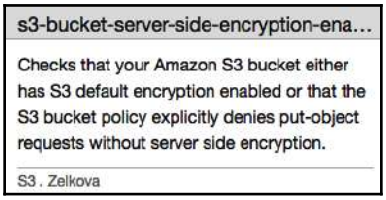
3. To add a new rule, select **Add Rule**:

4. From here, you will be presented with a list of pre-configured AWS-managed rules. You can filter on these rules using the search box. As shown in the screenshot, I have searched for S3 and it has returned all compliance checks that relate to S3, of which there are 14. So it's likely that there is a compliance check that already exists that you are looking for, or at least a very similar one:

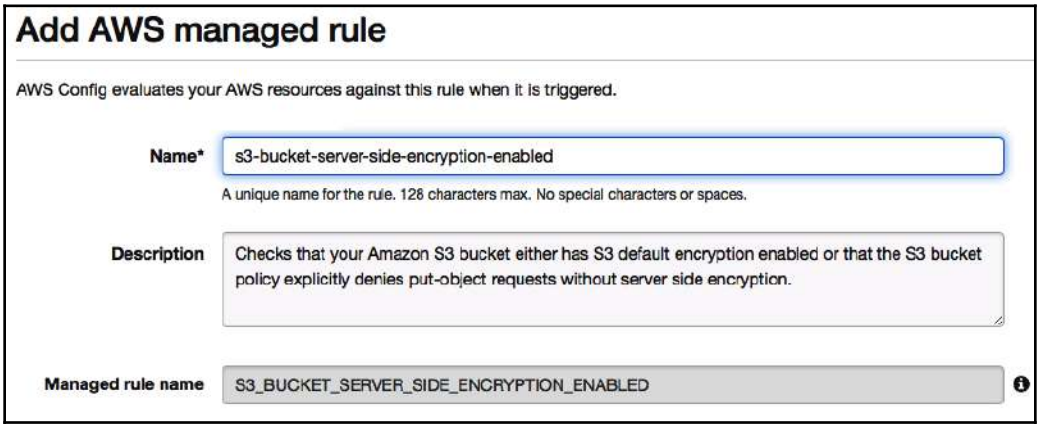


If there isn't a rule that matches your requirements, then you can select the **Add custom rule** button, which will allow you to select a Lambda function that will perform the logic of your compliance check. So, if you are familiar with AWS Lambda and are confident in the creation of a function, then you can create your own checks to fit your exact compliance needs.

5. For this demonstration, I am going to select the existing pre-configured AWS-managed Config rule **s3-bucket-server-side-encryption-enabled**. This will evaluate my S3 buckets to check which ones do not provide default server-side encryption:

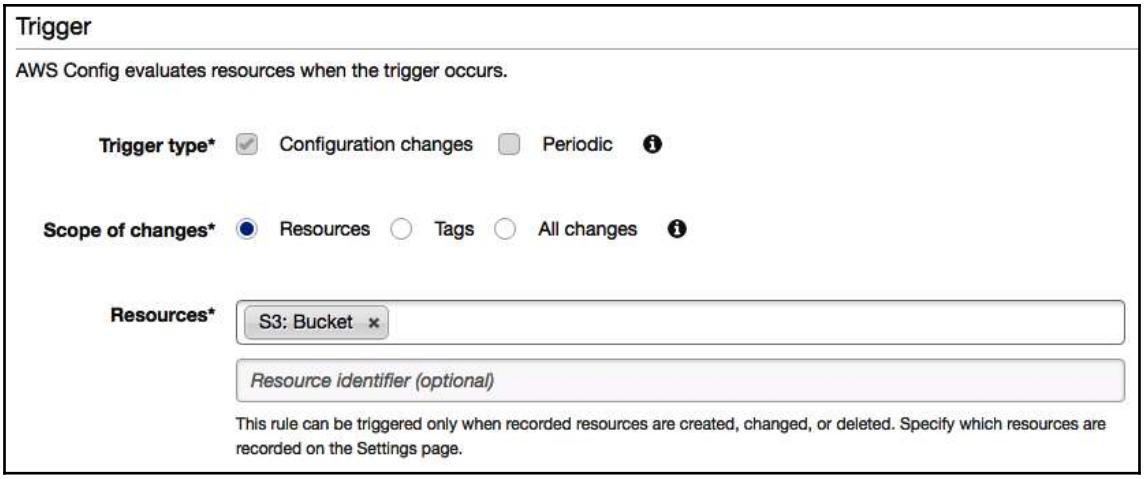


6. Once you have selected your Config rule, you are able to view its configuration and make additional configurational changes:



Firstly, you will be able to change the **Name** and **Description** to something more meaningful if desired. The **Managed rule name** section is protected and can't be changed. This is the actual Lambda function that is involved when the compliance check is run.

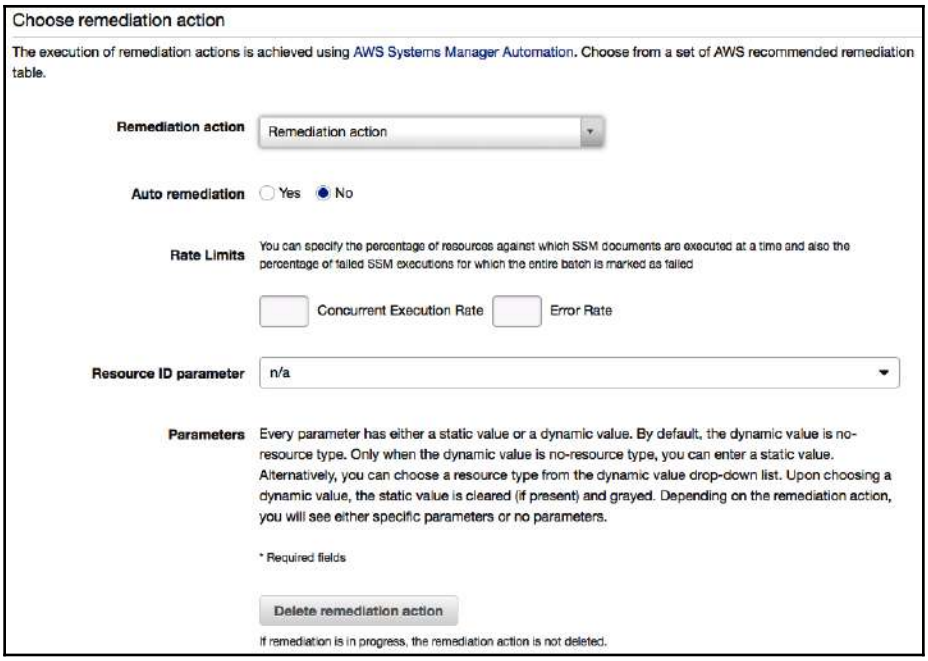
7. Next, you can configure the **Trigger** section:



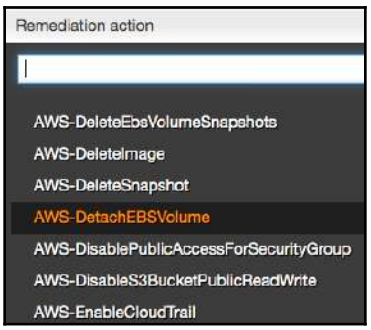
Here, you can make changes to affect how and when the compliance check is invoked. As this is a managed rule, the **Trigger type** is automatically set, however, if you were to create your own compliance check, you would be able to specify whether you wanted the check to trigger for every configuration change relating to the associated resources, or on a periodic basis regardless of any configuration changes in your environment.

Scope of changes allows you to specify the resources that you want the check to be associated with. As this is a check specifically to check the encryption status of S3 buckets, the scope of changes has been restricted to S3 buckets only.

8. Next, you have the ability to configure remediation options:



The **Remediation action** is a drop-down list of pre-configured automated remediation actions carried out by AWS Systems Manager Automation. An example use of this could be if you were checking to make sure EBS volumes were encrypted, and your compliance check found out that a volume wasn't, then you could select the following remediation action of **AWS-DetachEBSVolume**. This would prevent anyone from having access to the insecure EBS volume, allowing you to rectify the problem:



For this demonstration, I will leave the remediation action empty and will set the **Auto remediation** option as **No**, and all other options will be left as their default.

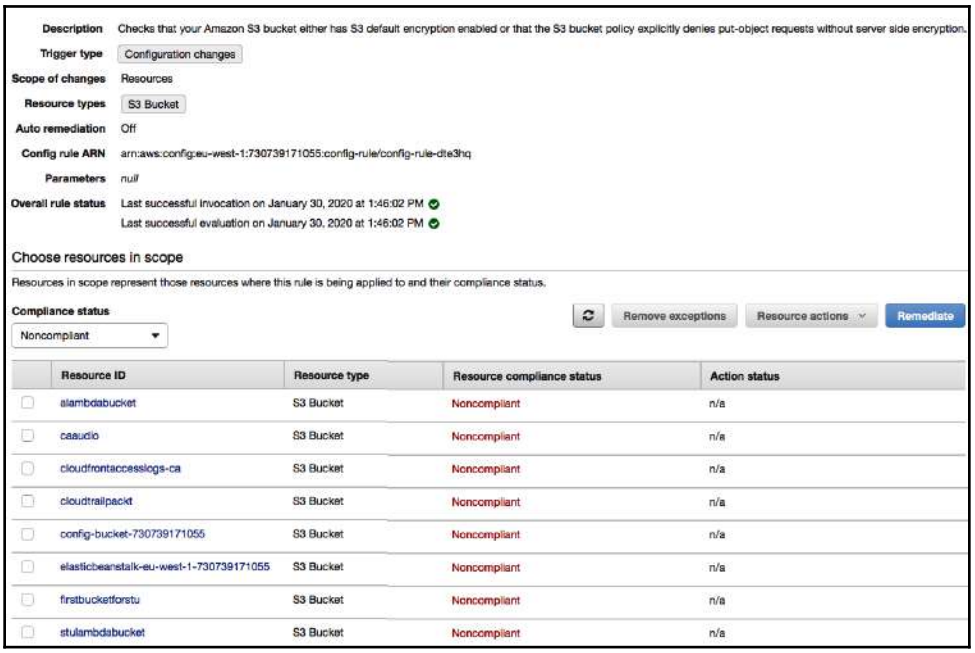
9. Once your configuration changes have been made, select the blue **Save** button. Your new rule will then begin its evaluation based on your configuration settings:

	Rule name	Compliance
○	s3-bucket-server-side-encryption-enabled	Evaluating...

10. If any non-compliant resources are found, it displays its findings as shown. As you can see, this rule has found eight non-compliant S3 buckets:

	Rule name	Compliance
○	s3-bucket-server-side-encryption-enabled	8 noncompliant resource(s)

11. By selecting the **Rule**, we can dive deeper to find out exactly which resources do not provide the level of encryption required:



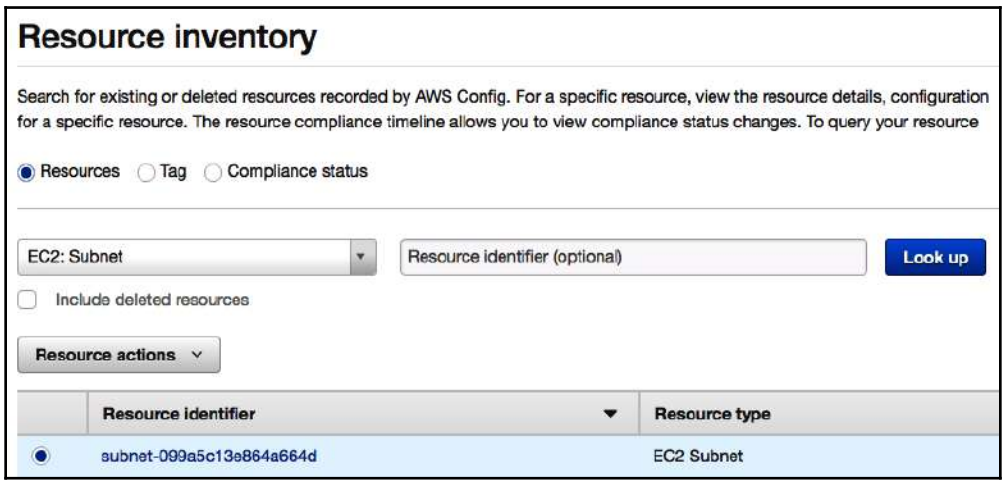
12. Using this information, we can easily uncover which buckets are affected and can choose to take any action that might be necessary.

Using AWS Config rules gives you the ability to very quickly and easily have a level of automated checks to help with your overall security posture and compliance. This also helps protect against any accidental human errors that have been made that might otherwise have gone unnoticed and could potentially have led to exposure or a vulnerability of some sort.

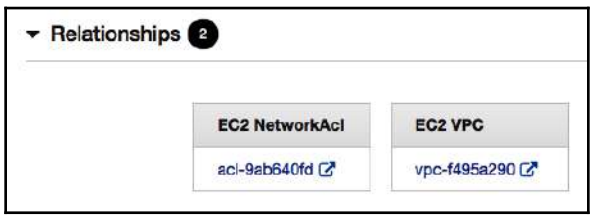
Resource relationships

The resource relationship component builds a logical mapping of your resources and their links to one another. Let's run through a quick example:

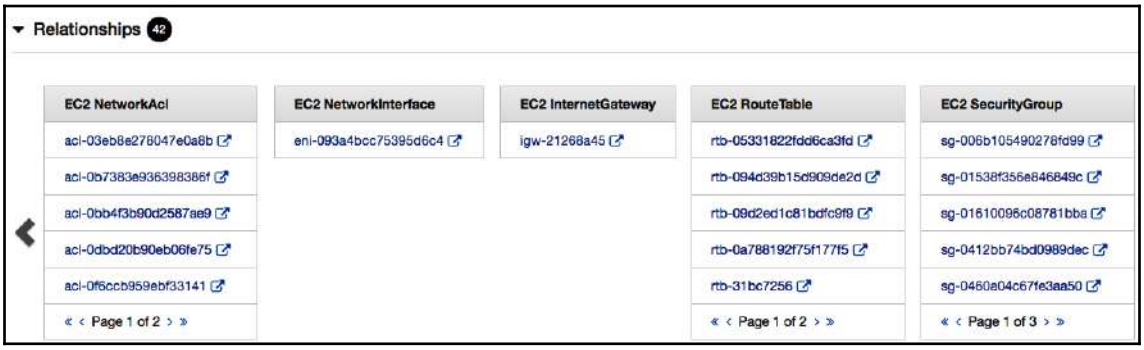
1. Select one of the resources from within AWS Config, for example, a particular subnet:



2. Now navigate through to the configuration timeline. From here, you can see the **Relationships** section:



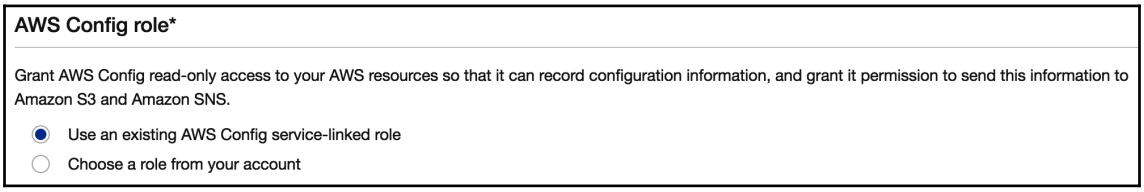
3. You can now clearly see which EC2 Network ACL and which VPC this subnet belongs to, and can quickly and easily select either of these resources to then navigate to the configuration timeline of that resource, for example, the EC2 VPC:



Here, we can see all the relationships that this VPC has, and so it becomes a very powerful tool to help you understand your logical mapping of resources very quickly and easily.

AWS Config role

During the configuration of AWS Config, you will be asked to specify a role:



As stated in the screenshot, this is used to allow AWS Config to gain read-only access to your resources to record CI data. It will also be used to publish data to both your selected SNS topic for your configuration stream and your S3 bucket to store configuration history and snapshots. You can use an existing role within your AWS account, or allow AWS Config to select an existing service-linked role.

Now that we have seen the various components of AWS Config, let's briefly understand the Config process, which will help us understand how these components work together to provide the information necessary for the audit.

The AWS Config process

Before moving on to the next section in this chapter, let's just review the AWS Config process:

1. You must first configure elements of AWS Config, which in turn enables the **configuration recorder** to begin capturing and recording changes within your environment.
2. AWS Config will then actively identify and discover resources based upon your configuration.
3. For any changes, creations, and deletions of supported resources made within your environment where AWS Config is running, a **configuration item** will be created.
4. This change will be sent to the notification stream (**SNS topic**).
5. If you have any **Config rules** configured for managing compliance, AWS Config will evaluate any changes made to your environment and assess that change for any non-compliance. If a change of compliance state occured, a notification will be sent to the notification stream.
6. Should a **configuration snapshot** be initiated, a point-in-time snapshot of your environment will be captured and the output delivered to your predefined S3 bucket.
7. AWS Config will periodically update its **configuration history** for each resource type, which can be viewed via the AWS Config dashboard within the Management Console, in addition to reviewing the output stored in your designated S3 bucket.

As you have seen, AWS Config is a great service to help you manage your resources from an auditing perspective. It can provide you with a timeline of events, saving resource history, changes, and relationships. As an auditing tool, this is highly effective and can be used to provide a significant amount of auditing evidence during an audit being performed by a third party.

Now let's move on to another service that can be used to help you with your compliance program by protecting **personally identifiable information** (**PII**) – Amazon Macie, which tightly integrates with Amazon S3.

Maintaining compliance with Amazon Macie

Amazon Macie is a managed service, backed by machine learning, that provides an automatic way of detecting, protecting, and classifying data within your S3 buckets. By reviewing and continuously monitoring data object access patterns in S3 and associated CloudTrail log data, Amazon Macie can identify and spot any irregular or suspicious activity that sits outside of what Macie would consider familiar operations, potentially identifying a new security threat.

Some useful features of Amazon Macie include the following:

- The ability to use **natural language processing** (**NLP**) techniques to interpret data stored in S3, helping to classify it. To learn more about NLP, please visit `https://en.wikipedia.org/wiki/Natural_language_processing`.
- The ability to spot changes to specific security policies and ACLs that might affect who has access to your S3 bucket.
- The ability to categorize information, including sensitive security data such as **personally identifiable information** (**PII**), **protected health information** (**PHI**) data, access keys, and API keys.
- Customized configuration, allowing you to set and assign business values to certain types of data using a risk score. Depending on your business and what's considered critical, you can set your own values on what you consider a risk.

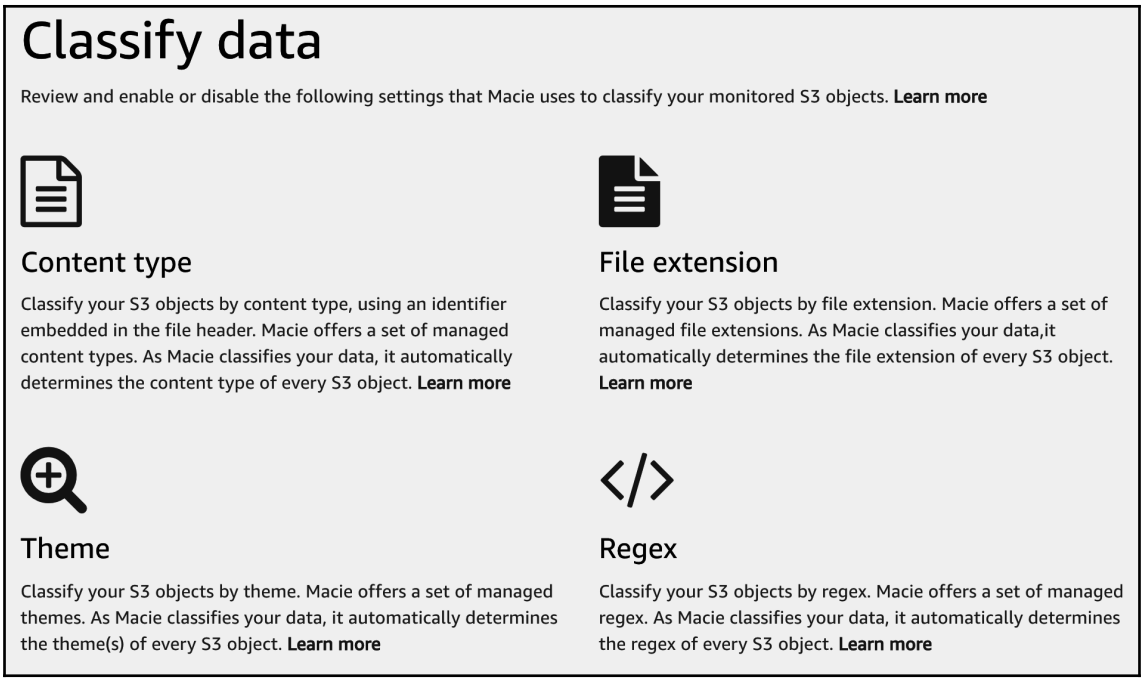
One of the key components of Amazon Macie is how it classifies data to help determine its level of sensitivity and criticality to your business. I now want to explain more about these classifications next.

If you would like to enable Amazon Macie, please refer to: `https://docs.aws.amazon.com/macie/latest/userguide/macie-setting-up.html`.

Classifying data using Amazon Macie

Amazon Macie classifies data through a series of automatic content classification mechanisms. It performs its classification using the object-level API data events collated from CloudTrail logs.

There are currently five different levels of classification at the time of writing this book. Four of them can be seen from the Amazon Macie console from within the AWS Management Console, which can be enabled/disabled, and are shown as follows:



First, we will go over the fifth type, which cannot be managed from the console.

Support vector machine-based classifier

The fifth type, **support vector machine-based classifier** is managed entirely by AWS with no modifications of any type and so is hidden from the console. This classifies each of your objects stored in S3 by analyzing the content within each object, in addition to its metadata, such as document length. Examples of these classifications include the following:

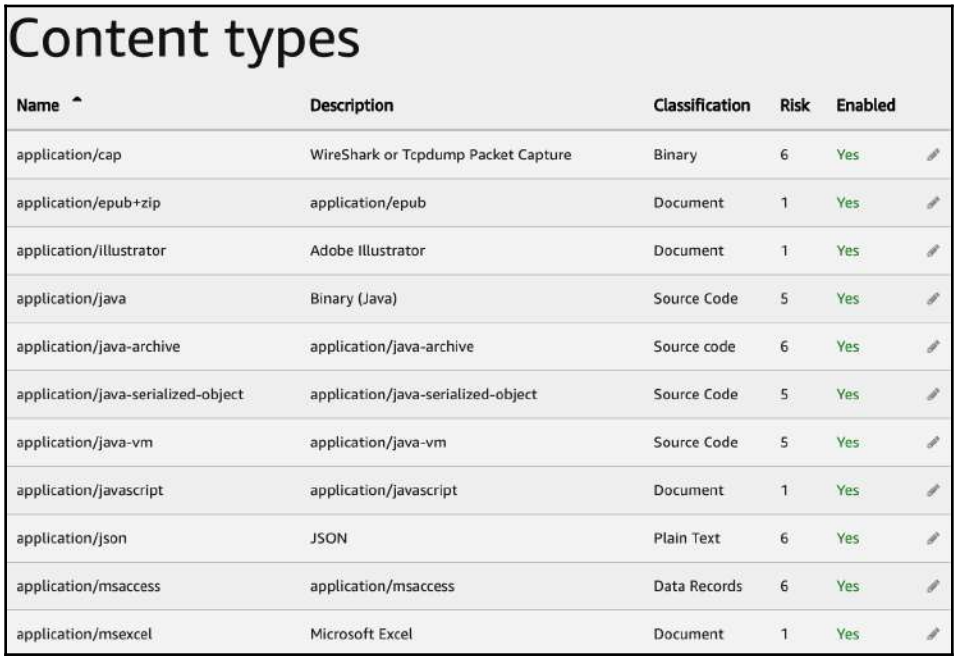
- Financial
- Application logs
- Web languages
- Generic encryption keys

For a full list and breakdown of the full list of classifications, please see the AWS documentation found: `https://docs.aws.amazon.com/macie/latest/userguide/macie-classify-objects-classifier.html`.

Content type

This classification looks at the classification of the actual file that is being stored, using an identifier that's embedded in the file header of the object. As an example, this classification could be identified as a *document* or *source code*. Each object stored is associated against a set of predefined content types, which would also indicate its risk factor.

A sample of these content types can be seen here, taken from the Management Console:



Content types

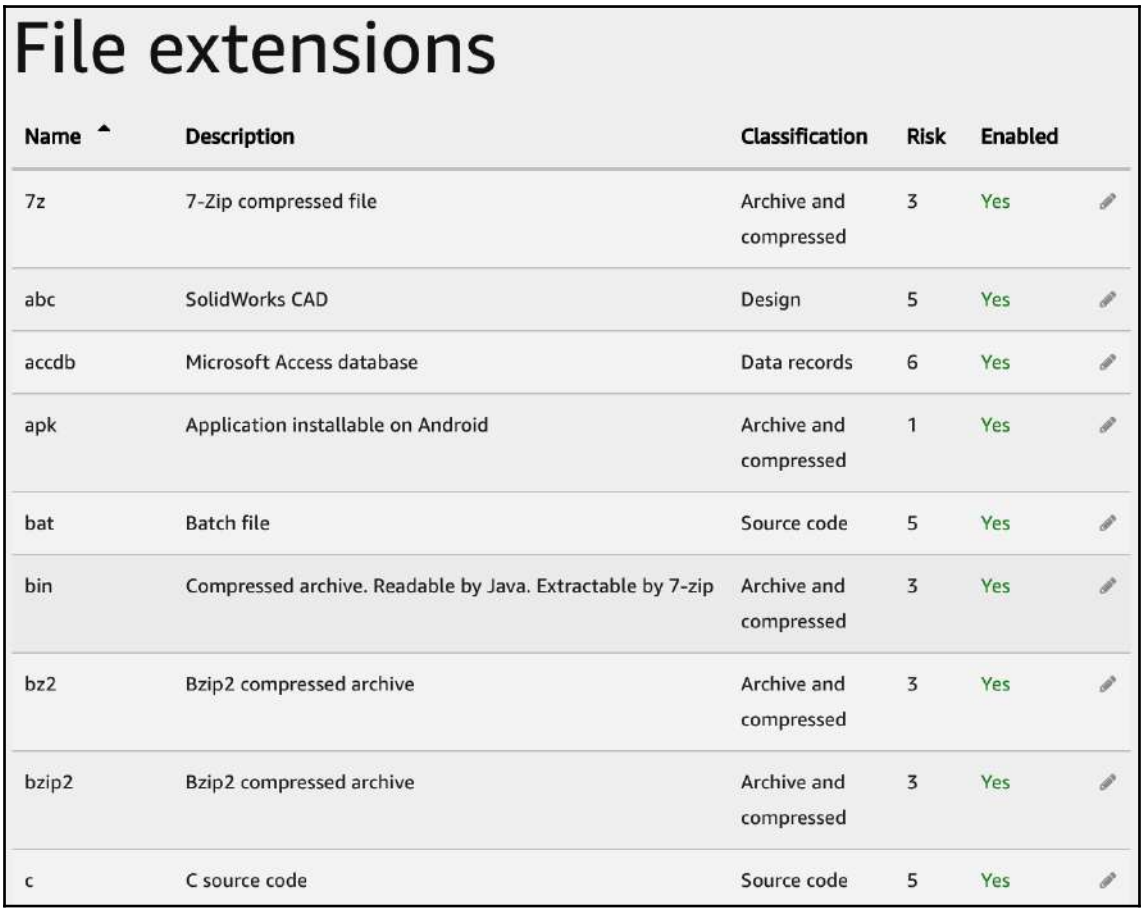
Name	Description	Classification	Risk	Enabled
application/cap	WireShark or Tcpdump Packet Capture	Binary	6	Yes
application/epub+zip	application/epub	Document	1	Yes
application/illustrator	Adobe Illustrator	Document	1	Yes
application/java	Binary (Java)	Source Code	5	Yes
application/java-archive	application/java-archive	Source code	6	Yes
application/java-serialized-object	application/java-serialized-object	Source Code	5	Yes
application/java-vm	application/java-vm	Source Code	5	Yes
application/javascript	application/javascript	Document	1	Yes
application/json	JSON	Plain Text	6	Yes
application/msaccess	application/msaccess	Data Records	6	Yes
application/msexcel	Microsoft Excel	Document	1	Yes

As you can see, there are five different fields that are used in this classification method. The **Name** and **Description** fields define the file type that's associated with the object, such as a Microsoft Excel file, as seen in the last row of the excerpt of the preceding screenshot. The **Classification** is what Amazon Macie determines the file classification to be, such as binary, document, plain text, and so on. In the Microsoft Excel example, the classification is identified as **Document**.

When looking at the **Risk** column, it is rated from 1 to 10, with 10 carrying the highest risk. Again, our Excel document carries a low risk of just **1**. For each of the content types, you can choose to have these enabled or disabled, and it's worth pointing out that this is the only value that you can change for each classification type by clicking on the *pencil* symbol.

File extensions

Quite simply, instead of looking at the content type of the object, as in the previous classification, this determines a classification based upon the actual file extension of the object. A sample of these file extensions can be seen here:



File extensions

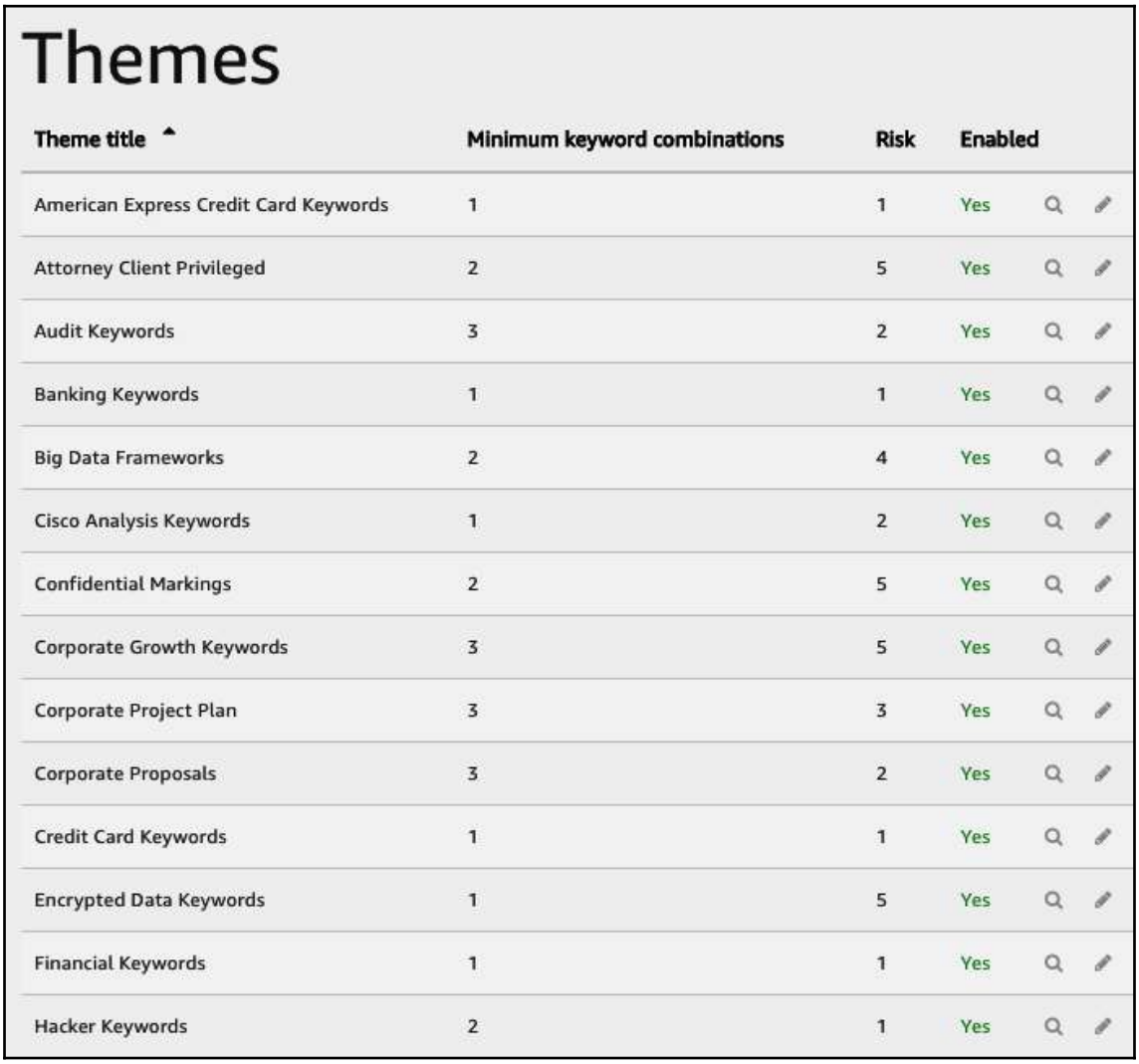
Name	Description	Classification	Risk	Enabled
7z	7-Zip compressed file	Archive and compressed	3	Yes
abc	SolidWorks CAD	Design	5	Yes
accdb	Microsoft Access database	Data records	6	Yes
apk	Application installable on Android	Archive and compressed	1	Yes
bat	Batch file	Source code	5	Yes
bin	Compressed archive. Readable by Java. Extractable by 7-zip	Archive and compressed	3	Yes
bz2	Bzip2 compressed archive	Archive and compressed	3	Yes
bzip2	Bzip2 compressed archive	Archive and compressed	3	Yes
c	C source code	Source code	5	Yes

The first column, **Name**, shows the actual file extensions that are supported by Macie, for example, we can see that a **bat** file is a **Batch file** (as per the **Description** field). The **classification** is then determined for each file extension, for example, **Archive and compressed**, **Design**, **Data records**, **Source code**, **Keychain**, **Executable**, and **Email data**.

Risk, again, ranges from 1 to 10 (10 being the highest risk) and each classification can either be enabled or disabled.

Themes

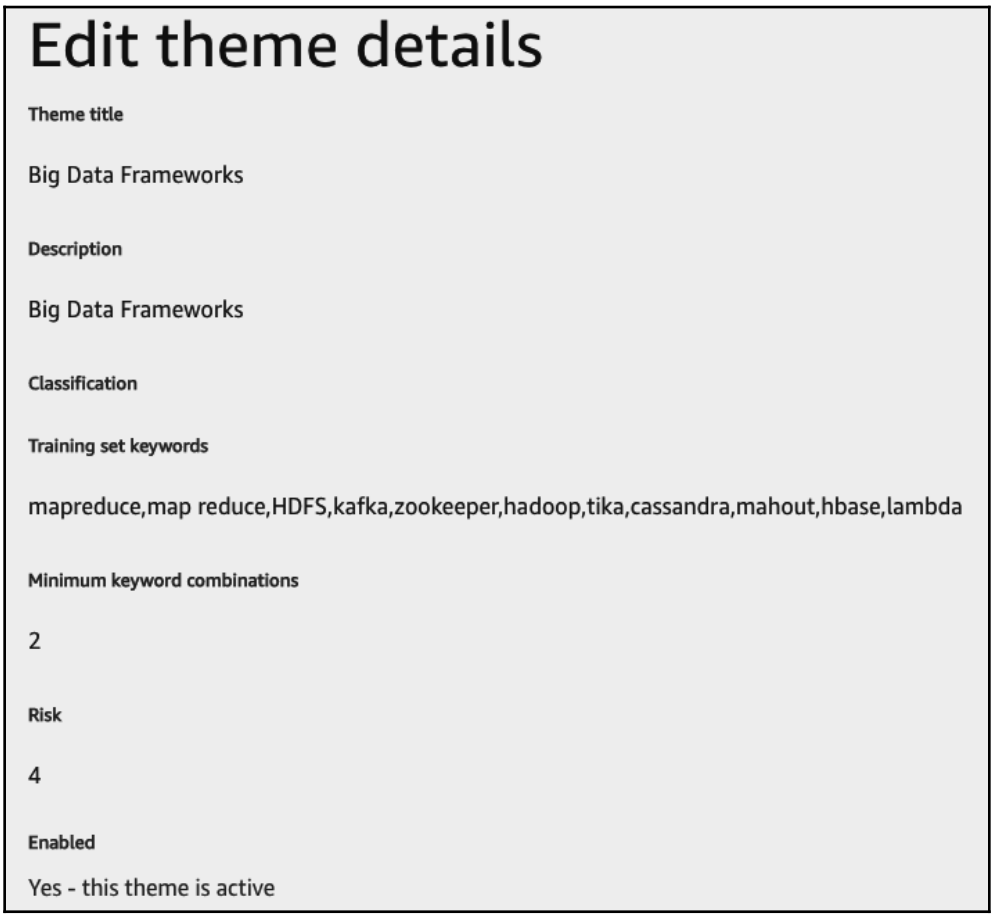
Classification by themes is very different compared to both content type and file extension. With themes, a classification is made using a list of predefined keywords that exist within the actual content of the object being stored. Using themes, Amazon Macie can assign more than one theme if multiple themes are detected within the object:



Themes

Theme title	Minimum keyword combinations	Risk	Enabled
American Express Credit Card Keywords	1	1	Yes
Attorney Client Privileged	2	5	Yes
Audit Keywords	3	2	Yes
Banking Keywords	1	1	Yes
Big Data Frameworks	2	4	Yes
Cisco Analysis Keywords	1	2	Yes
Confidential Markings	2	5	Yes
Corporate Growth Keywords	3	5	Yes
Corporate Project Plan	3	3	Yes
Corporate Proposals	3	2	Yes
Credit Card Keywords	1	1	Yes
Encrypted Data Keywords	1	5	Yes
Financial Keywords	1	1	Yes
Hacker Keywords	2	1	Yes

Theme title identifies the type of keywords that are associated, for example, **Encrypted Data Keywords**. **Minimum keyword combinations** shows how many words must exist within the object from the associated theme title for it to be classified with the related **Risk**. So, for example, there must be two keyword combinations from within the **Big Data Frameworks** theme title for it to be associated with that theme and associated a **Risk** level of **4**. If you select **Theme Title**, you can view the keywords that are searched for that theme:

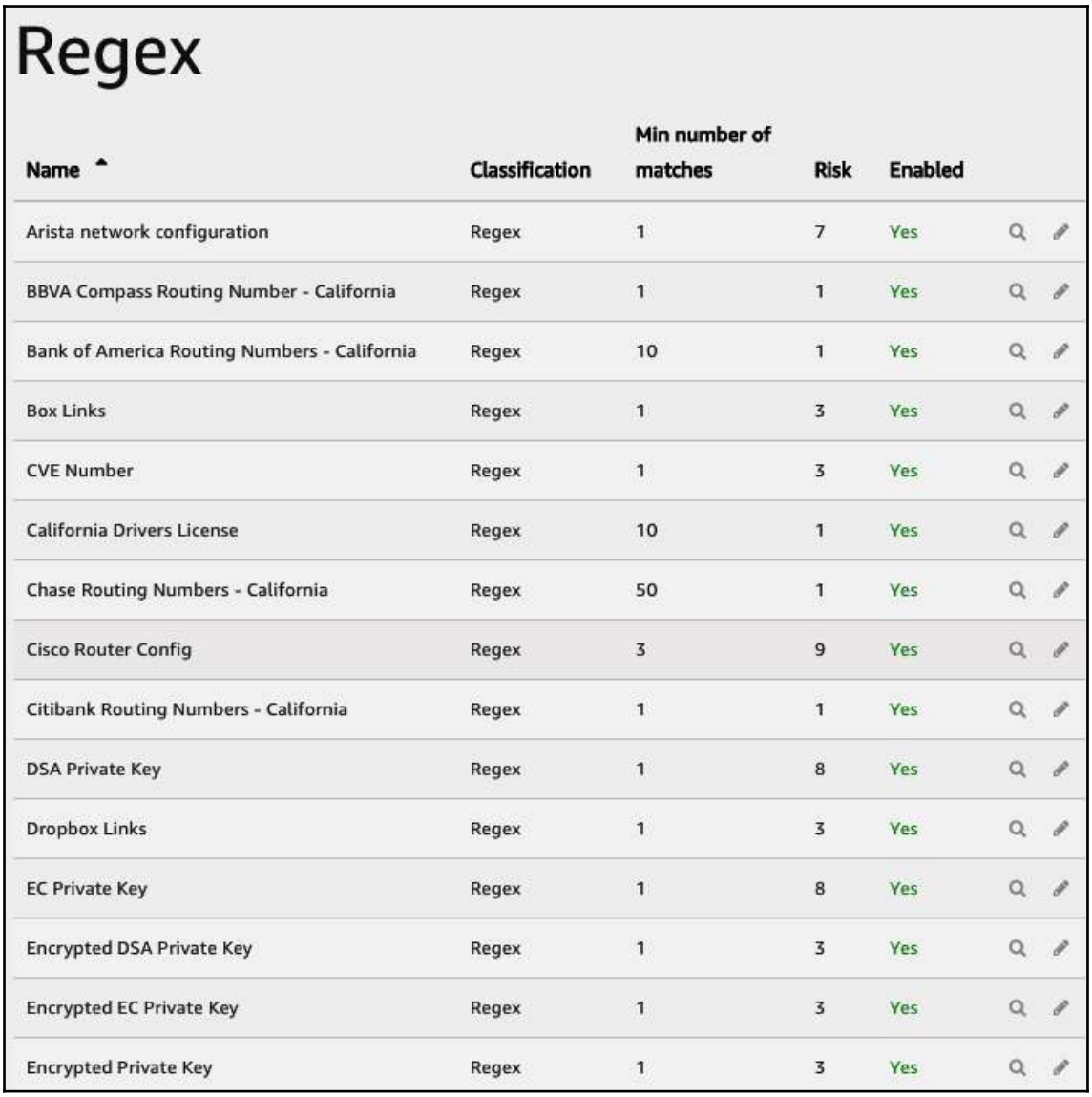


As you can see, the training set keywords that exist for the **Big Data Frameworks** theme include **mapreduce**, **map reduce**, **HDFS**, **kafka**, and **zookeeper** to name a few.

Again, the **Risk** level ranges from 1 to 10 (10 being the highest risk) and each theme title can be enabled or disabled.

Regex

Regex classifies an object based upon regular expressions, again found within the content of the object itself. It looks for a string of specific data or data patterns before associating a risk level:



Regex

Name	Classification	Min number of matches	Risk	Enabled
Arista network configuration	Regex	1	7	Yes
BBVA Compass Routing Number - California	Regex	1	1	Yes
Bank of America Routing Numbers - California	Regex	10	1	Yes
Box Links	Regex	1	3	Yes
CVE Number	Regex	1	3	Yes
California Drivers License	Regex	10	1	Yes
Chase Routing Numbers - California	Regex	50	1	Yes
Cisco Router Config	Regex	3	9	Yes
Citibank Routing Numbers - California	Regex	1	1	Yes
DSA Private Key	Regex	1	8	Yes
Dropbox Links	Regex	1	3	Yes
EC Private Key	Regex	1	8	Yes
Encrypted DSA Private Key	Regex	1	3	Yes
Encrypted EC Private Key	Regex	1	3	Yes
Encrypted Private Key	Regex	1	3	Yes

These tables are probably familiar to you by now and follow the same pattern as the previous classifications. Again, **Risk** ranges from 1 to 10, and each Regex can be enabled or disabled, and Macie can associate more than one Regex with each object.

Much like the themes, you can select a specific Regex name to gain additional information. The following is taken from the **Cisco Router Config** regex:

Through the five different types of object classification, Amazon Macie will collate and gather all risk scores from each to identify its overall risk factor. The result is defined by the highest risk found from any of the classification types. Let's work this out with the help of an example.

So let's assume you had a *document* that returned the following values from its object classification:

Object classification	**Risk values**
Support vector machine-based	2
Content type	1
File extension	1
Theme	2
Regex	5

The overall risk associated with this object would be identified as 5 as that is the highest value received.

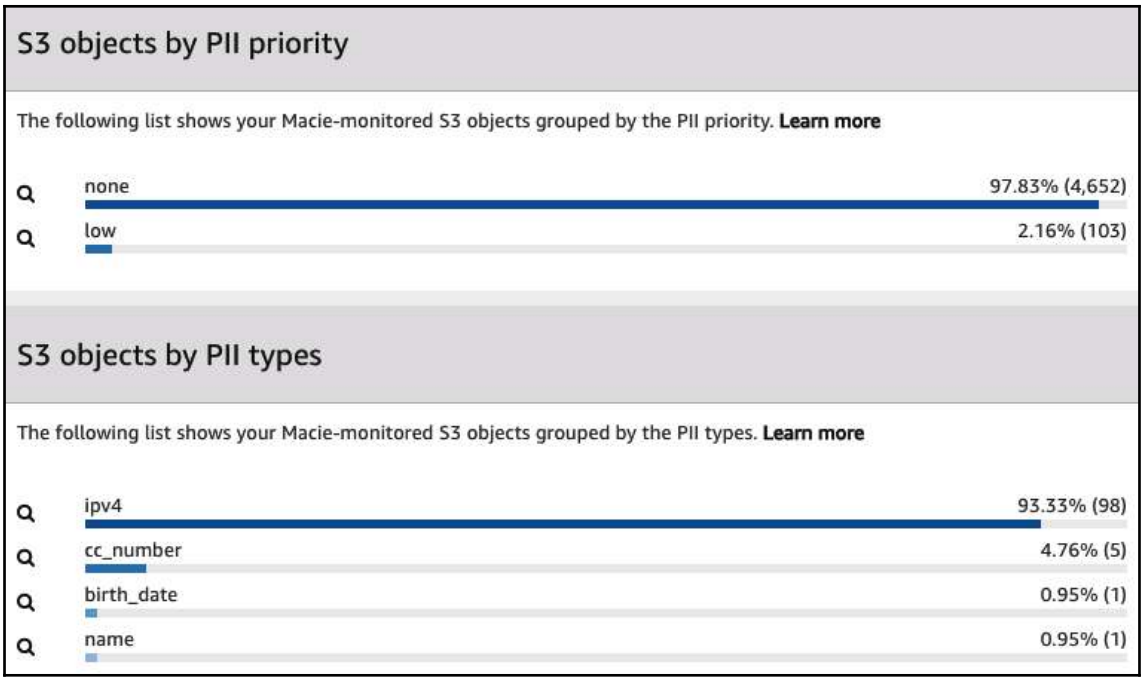
In addition to this object classification, Amazon Macie will also scan and perform the automatic detection of any **personally identifiable information** (**PII**) data found within each of the objects, which is based on industry standards (NIST-80-122 / FIPS 199). These automatic detections can identify the following PII data and assign a PII impact of either high, moderate, or low:

- Full names
- Mailing addresses
- Email addresses
- Credit card numbers
- IP addresses (IPv4 and IPv6)
- Driver's license IDs (USA)
- National identification numbers (USA)
- Birth dates

The PII impact is categorized as follows:

High	>= 1 full name and credit card >= 50 names or emails and any combination of other PII
Moderate	>= 5 names or emails and any combination of other PII
Low	1 to 5 names or emails and any combination of PII Any quantity of the PII attributes above (without names or emails)

The following screenshot shows the output of the data classification of PII data:



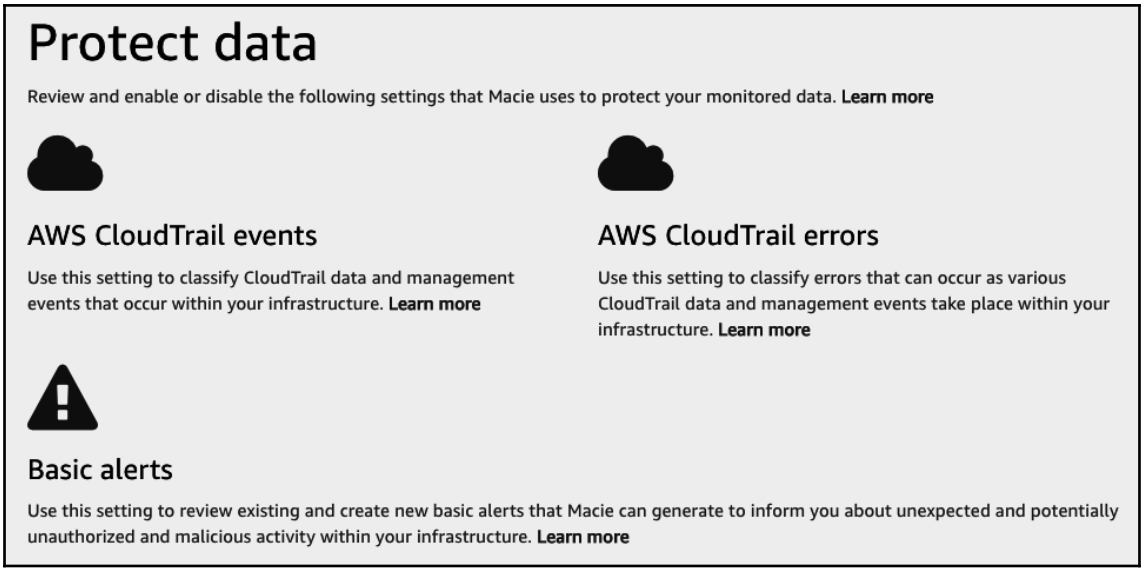
In this screenshot, we can see that Amazon Macie detected that 2.16% (103) of the objects contained a PII priority of **low**. Also, when grouping the classification by PII types, it detected that 93.33 % (98) of the objects contained IPv4 data.

In this section, we looked at how Amazon Macie classifies data by content type, file extensions, themes, and regex to help determine a risk factor of the data being stored. By identifying high-risk data, we can put in additional measures to ensure it is being protected through encryption, for example.

Amazon Macie data protection

Earlier, I mentioned that Amazon Macie can identify and spot any irregular or suspicious activity sitting outside of what Macie would consider normal boundaries of operations, potentially identifying a new security threat using AWS CloudTrail logs. Using historical data to review access patterns, Amazon Macie uses AI/ML to identify potential security weaknesses and threats from different users, applications, and service accounts.

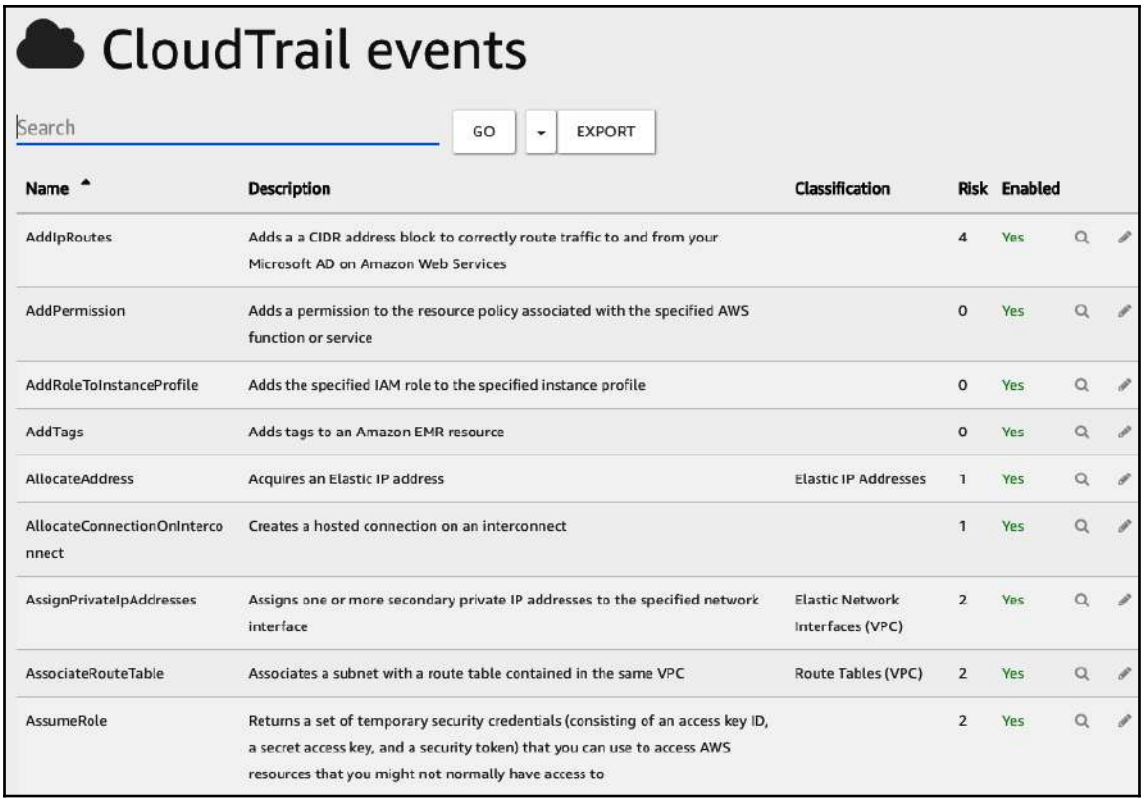
As you can see, Amazon Macie offers two features to identify threats – **AWS CloudTrail events** and **AWS CloudTrail errors**:



Let's go over these features one by one.

AWS CloudTrail events

Amazon Macie will review API calls gathered from CloudTrail logs stored on Amazon S3 and depending on the type of API and the impact it can have on your security posture, it will associate a risk score, again ranging from 1 to 10 (with 10 being the highest). If we look at a sample of these API calls, we can see what it looks for:



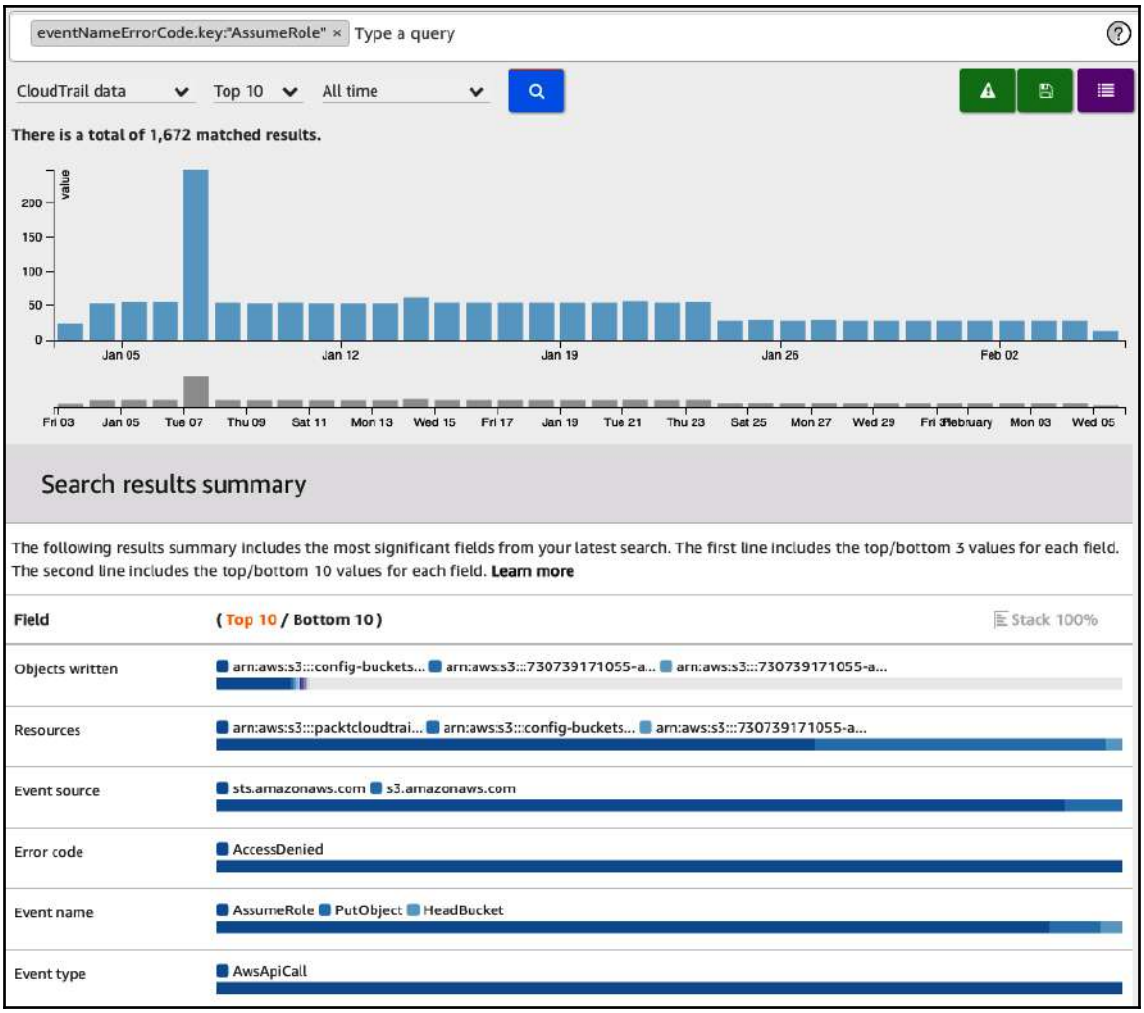
CloudTrail events

Search GO EXPORT

Name	Description	Classification	Risk	Enabled
AddIpRoutes	Adds a a CIDR address block to correctly route traffic to and from your Microsoft AD on Amazon Web Services		4	Yes
AddPermission	Adds a permission to the resource policy associated with the specified AWS function or service		0	Yes
AddRoleToInstanceProfile	Adds the specified IAM role to the specified instance profile		0	Yes
AddTags	Adds tags to an Amazon EMR resource		0	Yes
AllocateAddress	Acquires an Elastic IP address	Elastic IP Addresses	1	Yes
AllocateConnectionOnInterconnect	Creates a hosted connection on an interconnect		1	Yes
AssignPrivateIpAddresses	Assigns one or more secondary private IP addresses to the specified network interface	Elastic Network Interfaces (VPC)	2	Yes
AssociateRouteTable	Associates a subnet with a route table contained in the same VPC	Route Tables (VPC)	2	Yes
AssumeRole	Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) that you can use to access AWS resources that you might not normally have access to		2	Yes

To make it easier to navigate, a search option is available to search for a specific API call. The table is self-explanatory and follows a similar approach to the object classifications we looked at previously.

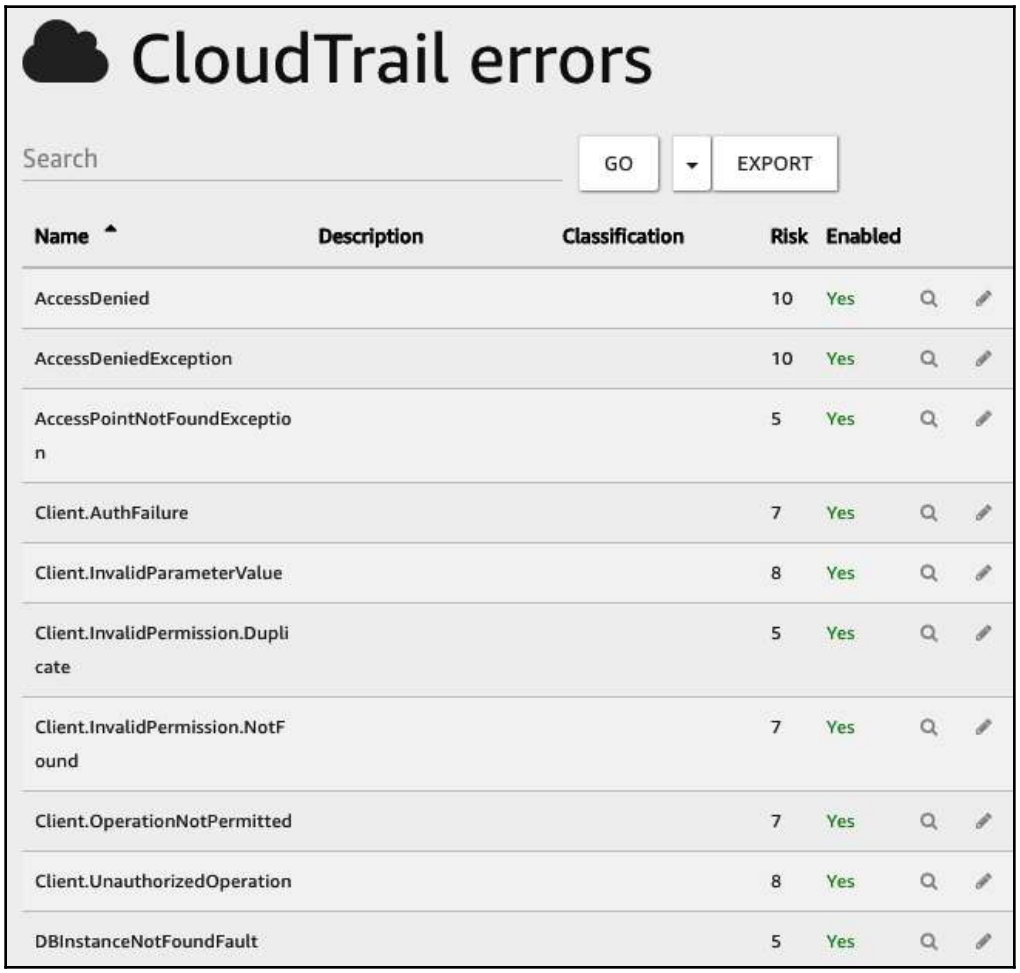
By using the magnifying search option associated with each API call, you can query the CloudTrail data to understand where and how this API is being used. For example, here is a screenshot of the **AssumeRole** API, allowing you to dive deeper into all the details captured by CloudTrail:



As you can see from the top section, this API was used multiple times and you can quickly see there was a peak on Tuesday 7th. If this was unexpected for a particular API, you could investigate the reason behind it further. At the bottom of the screenshot is a summary of the results that were used in conjunction with this API.

CloudTrail errors

This is an extremely useful feature as it captures any errors generated by an API call that CloudTrail has then captured, such as an **AccessDenied** response. This could be a sure sign that someone is trying to access something that they shouldn't be and could be the sign of a potential security attack or breach. There are many different errors that Amazon Macie looks for, which are assigned a risk value between 1 and 10 (10 being the highest risk). The following screenshot shows some of these errors:



Using the preceding example of **AccessDenied,** you can see this carries the highest risk factor or 10.

All of the results of the classification types and personally identifiable information and data risk values, along with any potential security problems found, are presented in a series of graphs and tables accessed via the Amazon Macie dashboard, which can be drilled down into to find further information. If you then couple this information with the ability to configure alerts for Amazon Macie's findings, it allows you to implement a series of strong compliance controls to meet stringent security controls that you need to adhere to.

Summary

In this chapter, we looked at some of the different services and features that we hadn't already covered thus far that can be used to help you meet your business audit requirements, certifications, and compliance controls. The services we looked at included AWS Artifact, AWS CloudTrail, AWS Config, and Amazon Macie.

Remaining compliant in the cloud can be a daunting task, but having an understanding of some of the logging capabilities and the services that can help you capture this information is vital. Being aware of the features of these tools will help you maintain a level of compliance and maintain full audit awareness.

In the next chapter, we are going to be looking at how you can utilize automation to quickly identify, record, and remediate security threats as and when they occur.

Questions

As we conclude this chapter, here is a list of questions for you to test your knowledge regarding its material. You will find the answers in the *Assessments* section of the *Appendix*:

1. Which AWS service is an on-demand portal to allow you to view and download AWS security and compliance **reports**, in addition to any online **agreements?**
2. Which security feature of AWS CloudTrail ensures that your log files have not been tampered with or modified after they have been written to your bucket in Amazon S3?
3. Which feature in AWS Config automatically monitors your resources to ensure they are meeting specific compliance controls?
4. Which service is backed by machine learning and provides an automatic way of detecting, protecting, and classifying data within your S3 buckets?
5. True or false: Amazon Macie classifies data through a series of automatic content classification mechanisms. It performs its classification using the object-level API data events collated from CloudTrail logs.