**8**

**Best Practices for Application Security, Identity, and Compliance**

In the past, a common refrain from companies was that they were hesitant to move to the cloud because they believed the cloud was not secure. A big part of this pushback was that companies didn’t understand the cloud or its capabilities. It is possible to have security vulnerabilities even if you use cloud infrastructure. However, as we will see in this chapter, AWS provides a comprehensive catalog of services enabling you to create highly secure sites and applications.

When creating applications and implementing workflows, it is imperative to consider security from the start of your design and not as an afterthought. First, you will understand why security is essential in any system – not just in the cloud. Next, you will learn how AWS, in general, and IAM, in particular, can help us design and build robust and secure cloud applications. Also, as you will see in this chapter, AWS provides a veritable cornucopia of other security services.

In this chapter, we will cover the following topics:

* Understanding the importance of security, identity, and compliance in AWS
* AWS’s shared responsibility model
* Getting familiar with identity and access management
* Managing resources, permissions, and identities using IAM
* Applying security controls and infrastructure protection in AWS
* Building data protection
* Adhering to compliances
* Learning about other AWS security services
* AWS security best practices

By the end of this chapter, you will have learned about how to secure your AWS cloud environment and the AWS services available to make your environment secure. AWS offers a plethora of security services to assist with these requirements. In this chapter, we will review these services in detail.

**Understanding the importance of security, identity, and compliance in AWS**

Many organizations face challenges in maintaining and managing the security of their on-premises infrastructure. In an on-premises environment, it can be challenging to know what resources and data are out there at any given time, where they are moving, and who is utilizing/accessing them. Accurate, real-time asset inventory requires expensive and complex tooling, making it inaccessible for most organizations. This lack of visibility in their on-premises environment hinders their ability to ensure adequate security and compliance of infrastructure and data. With AWS, you can see all your infrastructure and application resources in one place and maintain servers, storage, and database inventory records and access patterns.

AWS enhances your capacity to adhere to key security and compliance standards, such as data locality, protection, and confidentiality, through its extensive services and features. Boasting the largest network of security partners and solutions, the capabilities of AWS can be further extended through familiar security technology and consulting providers. AWS is compliant with major security standards and certifications, such as PCI-DSS, HIPAA/HITECH, FedRAMP, SEC Rule 17a-4, the EU Data Protection Directive, and FISMA, thereby enabling organizations to meet compliance requirements globally.

Another common problem is the reliance on manual processes for remediation. This may involve the manual copying of access information from one tool to another or the manual application of security patches. Automating key security tasks has been challenging due to the lack of interoperability between third-party and custom-made tools. These manual processes result in inconsistent execution and longer wait times to address all systems, and often negatively impact the customer experience. Automation aims to solve these issues by programmatically managing security tasks, such as checking if access to an application server is exposed to the internet or ensuring that an S3 bucket is not left public unintentionally.

As with any computer system, ensuring that it’s secure and that only authorized users access the system is paramount. Security should be incorporated at the beginning of your application design and not as an afterthought. AWS provides a broad set of security offerings that can ensure your application is secure and that your application data is safe. Notice that we used the word *assist*. Just because you are using AWS does not mean that your applications will be instantly secure.

A quick example of how easy it is to expose your data: There is nothing barring you from creating a bucket in AWS that is both unencrypted and public. You may get some warnings asking you if you are certain that you want to proceed, but AWS won’t disallow it. You could then put a client file in that bucket that may contain emails, passwords, names, addresses, and so on. This combination would immediately make this data accessible to anyone in the world with an internet connection (including any threat actors).

Even though you may not have published the URL for the bucket, please don’t assume it is secure. Threat actors know these mistakes happen sometimes. They constantly send requests to random AWS S3 buckets, trying to guess if they were created unsecured and exposed. And occasionally, they get lucky and hit pay dirt.

In the following section, we will learn more about what security AWS provides without involving the user and what services and tools AWS provides to its users to keep their systems secure.

**Understanding the shared responsibility model**

AWS uses the **shared responsibility model**, which means that AWS and its users are responsible for keeping applications secure. However, the lines of responsibility are pretty clear. AWS is solely responsible for some aspects (for example, physical data center security), and users are solely responsible for other aspects (for example, making sure that Amazon S3 buckets that will contain sensitive information are private, accessible to only authorized users, and encrypted).

This model enables AWS to reduce the burden of securing some components needed to create applications while enabling users to customize the applications to suit their clients’ needs and budgets.

Depending on the service chosen, some responsibilities may fall on AWS or the user. For example, if you use Amazon RDS to stand up an instance of MySQL, the patching of the database and the underlying operating system would be performed by AWS.

Suppose you instead decide to install MySQL directly into an Amazon EC2 instance. In that case, you will still be able to use the MySQL functionality. But in this case, the responsibility to patch the operating system and the database would fall on you.

One quick note: If your use case requires you to deploy MySQL manually into an EC2 instance, there is another option rather than deploying it yourself and risking the database not being deployed properly and securely. It is better to work with an AWS Consulting Partner. AWS has a list of trusted Consulting Partners that they recommend, and that can assist AWS customers. AWS ranks these partners by the level of service that they can provide. They have changed the ranking names in the past, but as of December 2022, the rankings are as follows:

* Select Partner
* Advanced Partner
* Premier Partner

Where Premier Partner is at the highest level. The current list of Consulting Partners can be found here: <https://partners.amazonaws.com/>.

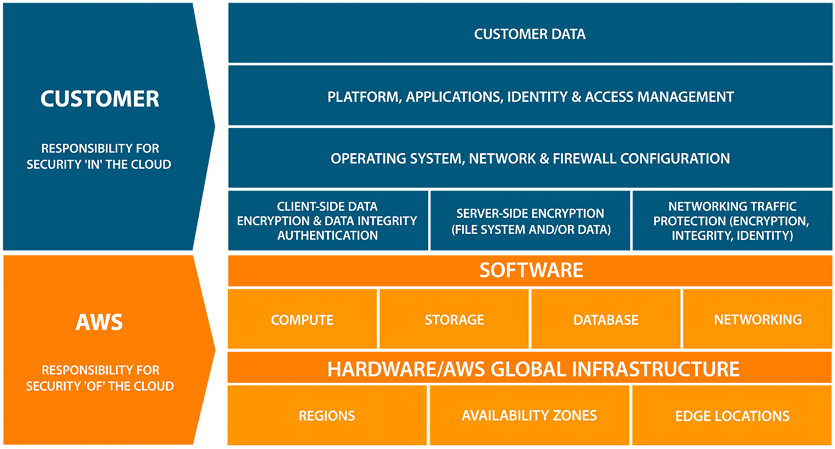
Leverage the power of AWS by incorporating security technology and consulting services from reputable and trusted providers. AWS has handpicked these providers to ensure they have extensive experience in securing every aspect of cloud adoption, from initial migration to ongoing management. The **AWS Partner Network** (**APN**) is a worldwide program of technology and Consulting Partners, many of whom specialize in delivering security solutions tailored to your specific needs and use cases. APN partner solutions promote automation, agility, and scalable growth with your workloads. You can easily access, purchase, implement, and manage these cloud-optimized software solutions, including SaaS products, within minutes on AWS Marketplace. You can find AWS security partners here: <https://aws.amazon.com/security/partner-solutions/>.

Deciding whether to use managed services versus deploying applications yourself is an important decision. Both approaches have advantages and disadvantages and the decision to use one method or another will be dictated by your business needs. For example, using Amazon RDS will require less maintenance since AWS performs the patching. Still, your organization may require you to own complete control of what changes happen to the software (perhaps because of regulatory reasons), in which case, using the approach to install MySQL on your own would make more sense. Now AWS provides Amazon RDS Custom, with which you can manage the underlying operating system and database settings as per your need while taking advantage of the scale that comes with RDS.

One common refrain heard to distinguish which components are the responsibility of AWS and which are the responsibility of the customer is as follows:

* AWS is responsible for the security **OF** the cloud.
* The AWS customer is responsible for security **IN** the cloud.

The following diagram illustrates the separation of duties:

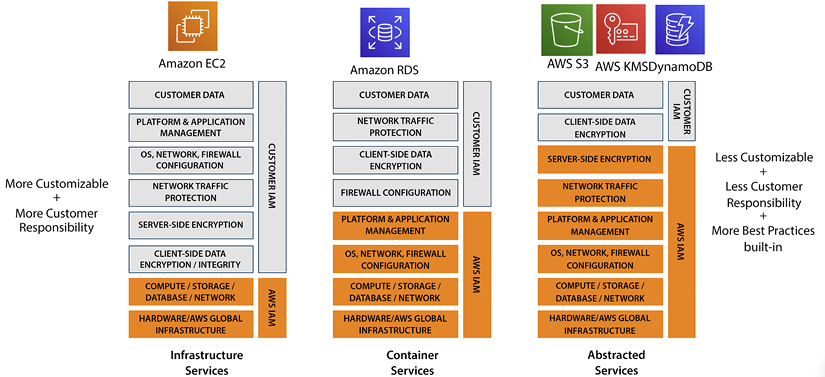


*Figure 8.1: Shared responsibility model*

The preceding figure shows in broad strokes how the responsibilities are broken down. For example, it clearly shows that the responsibility of AWS is for infrastructure elements such as regions, edge locations, and Availability Zones. This includes the physical security of the data centers. You may have passed an AWS data center and not noticed; AWS data centers are always unmarked buildings. On the other hand, customer data is the customer’s responsibility. When it comes to customer data, the encryption of the data is also the customer’s responsibility.

These areas of responsibility can be fuzzy depending on how a certain functionality is implemented. We see in the chart that databases fall under the purview of AWS, but as we saw previously, the customer can install a database, in which case they would be responsible for its management. Similarly, the chart in *Figure 8.2* shows the customer’s responsibility for operating systems, the network, and firewall configuration. But the shared responsibility model varies depending on the services provided to you by AWS.

The following diagram shows various levels of security responsibilities shared by AWS:



*Figure 8.2: Shared responsibility model for different AWS service categories*

As shown in the diagram above, in some cases, for example, when using AWS S3, the management of most items is the responsibility of AWS. For EC2, AWS only handles infrastructure security while RDS security is managed at the platform level. For DynamoDB, you just need to manage data and its access while everything else in the layer up until network traffic and server encryption is managed by AWS.

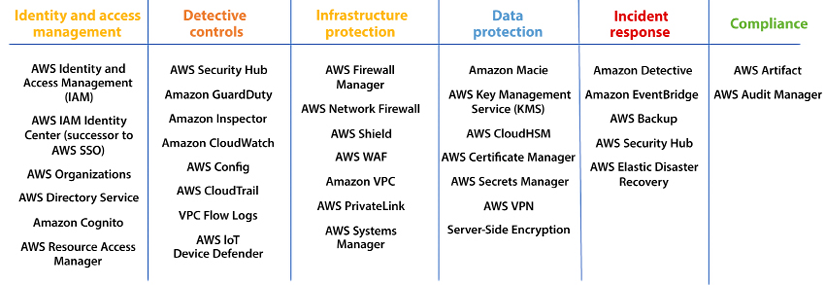
Another way to understand how security in AWS works is by using the analogy of locks and doors. AWS provides you with the doors and the locks to secure your applications and data, but you can still leave the door open and not secure the lock, leaving the contents of your home exposed to the world.

For example, Amazon RDS is a managed service. AWS does much of the heavy lifting to make a database secure. However, you can still publish the credentials to access your Amazon RDS instance on GitHub and let anyone who views these credentials access your database.

Overall, with AWS, you own your data and applications, and under the shared responsibility model, it becomes your responsibility to secure them by using various security services provided by AWS, from access management to encryption.

**AWS security, identity, and compliance solutions**

AWS has a broad range of security services available to fulfill every protection need of their customers. AWS is built to support the creation of secure, high-performing, resilient, and efficient infrastructure for your applications. The following AWS security services and solutions are designed to provide critical benefits that are crucial in helping you attain the best security posture for your organization:



*Figure 8.3: AWS security services*

As shown in the above table, AWS divides security services into the following pillars:

* **Identity and access management**: Establish, enforce, and monitor user access to AWS services, actions, and resources.
* **Detective controls**: Obtain the necessary visibility to identify potential issues before they affect your business, enhance your security posture, and minimize the risk to your environment.
* **Infrastructure protection**: Minimize the surface area to manage and enhance the privacy, control, and security of your overall infrastructure on AWS.
* **Data protection**: A collection of services that automate and simplify various data protection and security tasks, such as key management and storage, and credential management.
* **Incident response**: During a security incident, containing the event and restoring to a secure state are critical steps in a response plan. AWS offers tools to automate parts of this best practice.
* **Compliance**: Provide audit traces and artifacts in order to meet compliance requirements.

Let’s look into individual services belonging to these security pillars.

**Getting familiar with identity and access management**

Identity and access management is the most fundamental security posture for any organization, and AWS provides the following services in this category:

* **AWS Identity and Access Management** (**IAM**) – Securely manage access to AWS services and resources
* **AWS Organizations** – Policy-based management for multiple AWS accounts
* **AWS Directory Service** – Managed Microsoft Active Directory in AWS
* **AWS IAM Identity Center** (successor to AWS SSO) – Centrally manage **single sign-on** (**SSO**) access to multiple AWS accounts and business apps
* **AWS Resource Access Manager** – A simple, secure service for sharing AWS resources
* **Amazon Cognito** – Add user sign-up, sign-in, and access control to your web and mobile apps

Let’s look into each of the above services in detail.

**AWS Identity and Access Management (IAM)**

Perhaps the most fundamental and important service in AWS is **Identity and Access Management** (**IAM**), which can secure every single other software service offered by AWS. AWS IAM offers precise access control across all AWS services. This level of control allows you to define who can access specific services and resources and under what conditions. By creating IAM policies, you can manage access permissions for your users or applications to ensure minimal privilege access. IAM is a complimentary service provided by AWS at no extra cost. More specifically, AWS IAM can be used to do the following:

* Grant others shared access to your AWS account without sharing passwords or access keys.
* Provide granular permissions for different people and resources.
* Securely access AWS resources using IAM credentials for applications running on EC2 instances and other resources.
* Enhance security with multi-factor authentication.
* Facilitate identity federation for temporary access to your AWS account with external passwords.
* Be eventually consistent, with changes replicated globally. However, it is recommended to not include IAM changes in high-availability code paths and to verify changes have propagated before usage.

Here are a few use cases for AWS IAM:

* **Access control for AWS services**: IAM can be used to control who has access to various AWS services, such as EC2, S3, and DynamoDB, as well as what actions they can perform on those services.
* **Multi-factor authentication**(**MFA**): IAM supports MFA, which can be used to add an extra layer of security to an AWS account.
* **Secure application credentials**: IAM can be used to securely provide credentials for applications running on EC2 instances and other resources so that those applications can access other AWS resources.
* **Identity federation**: IAM can be used to grant temporary access to AWS resources for users with existing passwords, for example, in a corporate network or with an internet identity provider.
* **Billing and cost allocation**: IAM can be used to manage access to billing and cost allocation information for an AWS account.

Suppose you have a company that uses AWS to host its applications. You can use IAM to create a group for your developers, granting them access to EC2 instances, S3 buckets, and DynamoDB tables, while only allowing them read-only access to your billing information. You can also use IAM to set up MFA for the root account and individual IAM users to add an extra layer of security.

**Managing resources, permissions, and identities using IAM**

To understand AWS IAM, we must first understand how authentication and identity management work. Users, groups, roles, permissions, and policies are fundamental concepts that need to be fully understood to grasp how resources are secured using AWS IAM. The purpose of using IAM is to regulate the authentication and authorization of individuals who wish to utilize resources. This is achieved by establishing precise permissions through IAM, thereby determining who has access to what. IAM consistently implements these permissions for every request made. By default, all requests are denied (except for the root user, which is allowed by default) unless an explicit “allow” is specified. An explicit “deny” overrides any allows.

In the following sections, you will learn AWS IAM terms.

**IAM users**

An IAM user is an IAM principal you create in AWS to represent the person or application that uses it to interact with AWS. An IAM principal is a user, group, or service that is authenticated and authorized to access resources in an AWS account. An IAM principal can be an AWS account root user, an IAM user, an IAM role, or a federated user.

An AWS user comprises a username and associated credentials. Take, for instance, a user named John. Upon creating an IAM user account for John, you’ll need to establish a password for that user. You have the option to assign IAM user-specific permissions, such as the ability to start a particular Amazon EC2 instance.

An IAM user is an individual that needs to access, interact with, and potentially modify data and AWS resources. Users can interact through one of three ways:

* The AWS Management Console
* The AWS **Command-Line Interface** (**CLI**)
* The AWS API

Other than the root user, no implicit permissions or credentials are given when a new user is set up. That new user cannot access any resources until permission is explicitly assigned.

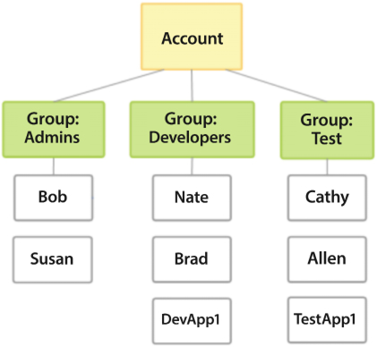
The IAM service in AWS enables you to securely control access to AWS resources and the actions that can be performed on those resources. You can use IAM to create and manage IAM principals, as well as to assign permissions to these principals to allow or deny access to AWS resources. For example, you can use IAM to create an IAM user for a person in your organization, and then grant that user permissions to access specific AWS resources or perform certain actions. Overall, using IAM helps you to securely and effectively manage access to your AWS resources, and helps you to enforce the principle of least privilege by granting only the necessary permissions to IAM principals.

**IAM user groups**

An IAM user group is an assembly of IAM users. By organizing IAM users into groups, you can efficiently manage their permissions as a collective. As an illustration, consider a user group named Dev, to which you have assigned the typical permissions required for developers. Any IAM user belonging to this group will automatically inherit the permissions assigned to the Dev user group.

When a new member joins your organization and requires developer privileges, you can grant the necessary permissions by adding them to the relevant user group.

On the other hand, if an individual changes their role within your organization, you can simply transfer them from their current user group to the appropriate new user group, rather than modifying their individual permissions. The following diagram shows IAM users assigned to different user groups:



*Figure 8.4: AWS IAM user groups and IAM user*

The above diagram shows three user groups, Admins, Developers, and Test, and IAM users assigned to those groups with the same credentials set. Putting users into groups facilitates permission management and gives system administrators a more efficient way to administer permissions. Users that have similar profiles are grouped. They could be grouped based on similar characteristics and on having similar needs, such as the following:

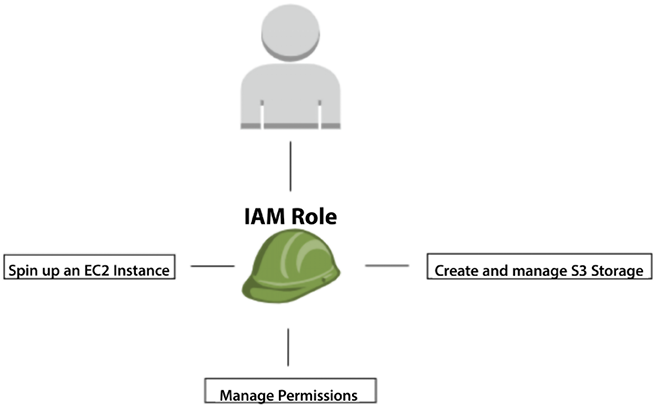
* Job function or role
* Department
* Persona

Then, permissions for users that belong to one group can be managed all at once through the group. It is recommended to put all users in one group that need the same access level. Often organizations use Active Directory to group employees, and in that case, you can map IAM groups to your Active Directory groups. If you have been around technology for a while, the idea of users and groups should not be new. However, IAM roles may require a little more explanation. Let’s continue discussing them in the next section.

**IAM roles**

An IAM role is a way to grant permission to access AWS resources to users or processes that do not have their own AWS credentials. The major difference is that, unlike users, IAM roles have no long-term credentials (i.e., passwords or access keys).

As shown in the diagram below, you can use an IAM role to allow a user to access an S3 bucket. For that, first, you need to create an IAM role that has the necessary permissions to access the S3 bucket. This can be done through the AWS Management Console or using the AWS CLI. For example, you might create a role that has the AmazonS3FullAccess policy attached to it. Next, create a user and associate the IAM role with the user. The user can then access the S3 bucket using the AWS Management Console or the AWS SDKs by assuming the IAM role. This will allow the user to use the permissions of the IAM role to access the S3 bucket, without the need for the user to have their own AWS credentials.



*Figure 8.5: AWS IAM role*

In IAM, a role is an object definition configuring a set of permissions assigned to that role. The role can be assigned to other entities, such as a user. A role is not directly connected to a person or a service. Instead, the role can be assumed by an entity that is given the role. Role credentials are always only temporary and rotated on a schedule defined by the AWS **Session Token Service** (**STS**). It is best practice to use roles whenever possible instead of granting permissions directly to a user or group.

STS allows you to request short-lived, restricted credentials for both AWS IAM users and federated users. This service is frequently utilized to grant temporary access to resources for trusted users, such as by granting them an IAM role that has a more limited set of permissions compared to their standard IAM user or federated user permissions.

STS enables you to grant trusted users temporary permissions to resources without having to share long-term AWS access keys. For example, you can use STS to grant temporary access to an IAM role that allows users to perform specific tasks in your AWS account, such as creating and managing Amazon EC2 instances or uploading objects to Amazon S3. STS can also be used to provide federated users with temporary credentials to access resources in the AWS cloud.

You can use STS to grant temporary credentials in several ways:

* AssumeRole: This operation enables you to grant a trusted user temporary access to an IAM role.
* GetFederationToken: This operation enables you to grant a trusted user temporary access to AWS resources that you specify in the permissions policy associated with the token.
* GetSessionToken: This operation enables you to obtain temporary credentials for an IAM user or for a federated user.

Using STS helps you to secure your AWS resources and provides flexibility for granting temporary access to your resources. In Python, the user can use the boto3 library to assume the IAM role and then access the S3 bucket like this:

import boto3

*# Assume the IAM role*

sts\_client = boto3.client('sts')

assumed\_role\_object = sts\_client.assume\_role(

RoleArn='arn:aws:iam::123456789012:role/my-iam-role',

RoleSessionName='my\_session'

)

*# Use the temporary credentials provided by the assume\_role method to access S3*

s3\_client = boto3.client('s3', aws\_access\_key\_id=assumed\_role\_object['Credentials']['AccessKeyId',aws\_secret\_access\_key=assumed\_role\_object['Credentials']['SecretAccessKey'],

aws\_session\_token=assumed\_role\_object['Credentials']['SessionToken'])

*# List the objects in the S3 bucket*

objects = s3\_client.list\_objects(Bucket='my-s3-bucket')

print(objects)

Furthermore, roles enable you to grant multi-account access to users, services, and applications. Assigning a role to users not part of your organization is possible. Obviously, this has to be done judiciously and with flexibility as required.

IAM roles carry out a fundamental task in the security access landscape. By assigning permissions to a role instead of directly to a user or group, roles facilitate and simplify system administration and allow these permissions to only be given temporarily.

**Policies and permissions**

Access control in AWS is achieved through the creation and attachment of policies to IAM identities (such as users, groups, or roles) or AWS resources. These policies, which are objects in AWS, define the permissions of the associated identity or resource when they are attached. When an IAM principal, such as a user or role, makes a request, AWS evaluates the relevant policies to determine whether the request should be granted or denied. The majority of these policies are stored in AWS in the form of JSON documents.

A policy is a named document with a set of rules that specify what actions can be performed. Each policy laid out in the document gives a set of permissions. These policies can then be assigned to the IAM principals covered previously—users, groups, and roles. The syntax for AWS policy documents comes in two flavors:

* JSON
* YAML

The following is the syntax for defining policy and permissions:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

- ec2:DescribeInstances

Resource: "\*"

The above policy allows the ec2:DescribeInstances action to be performed on all resources. The Version field specifies the version of the policy language being used. The Statement field is a list of individual statements that together make up the policy.

Each statement consists of an Effect field (either Allow or Deny), an Action field that lists the actions that are allowed or denied by the Effect field, and a Resource field that specifies the resources that the actions apply to. IAM policies can be attached to IAM users, groups, and roles to grant permissions to perform various actions on AWS resources.

Policies can be defined in the two following ways:

* **Managed policies**: When policies are defined as managed policies, they are created as standalone policies, and therefore they can be attached to multiple entities. Out of the box, AWS provides a set of predefined managed policies that can be used in many use cases. Managed policies can be combined to deliver additional access to roles, users, or groups. Finally, AWS users can define and customize their own managed policies.
* **Inline policies**: Inline policies are created within the definition of an IAM entity and can only be assigned to the entity to which they are attached. They do not have their own **Amazon Resource Name** (**ARN**). Since inline policies are related to a specific entity, they are not reusable. An ARN is an identifier used to uniquely identify AWS resources. It is a string that consists of several different parts, including the service, region, and resource identifier. Here is an example of an AWS ARN: arn:aws:s3:::my\_bucket/example.jpg. In this example, arn:aws:s3 indicates that the resource is an S3 bucket, my\_bucket is the name of the bucket, and example.jpg is the name of a file stored in the bucket.

It is best practice to use managed policies whenever possible and use inline policies only when there is a good reason to do so.

Permissions are lists of actions that can be taken on AWS resources. When a user or group is created, initially, they have no permissions. One or more policies can be attached to the new user or group to enable access to resources.

When creating policies, it is a good idea to abide by the principle of least privilege. In simple terms, this means that entities should be given a high enough level of access to perform assigned tasks but nothing more. For example, suppose an Amazon EC2 instance is created, and we know that only five users with five different IPs will access it. In that case, we should use allowlist for those IPs and only give them access instead of opening the Amazon EC2 instance to the whole world.

Here is an example IAM policy that allows an EC2 instance to perform certain actions on S3 and EC2 resources, but only from a specific IP address range:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:ListBucket",

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::my-s3-bucket"

],

"Condition": {

"IpAddress": {

"aws:SourceIp": "10.0.0.0/16"

}

}

},

{

"Effect": "Allow",

"Action": [

"ec2:StartInstances",

"ec2:StopInstances"

],

"Resource": "\*",

"Condition": {

"IpAddress": {

"aws:SourceIp": "10.0.1.1/16"

}

}

}

]

}

This policy allows the EC2 instance to list the contents of the my-s3-bucket S3 bucket and retrieve objects from it, as well as to start and stop other EC2 instances, but only if the request originates from an IP address in the range 10.0.1.1/16. You can attach this policy to an IAM role and then associate the role with an EC2 instance to apply the permissions to the instance.

Note: This is just one example of how IAM policies can be used to allowlist EC2 IP addresses. There are many other ways to write IAM policies and you should carefully consider the specific needs of your use case when writing your own policies. AWS provides a policy simulator. This policy simulator can test new policies you may create and ensure they have the correct syntax. You can learn more here: <https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_testing-policies.html>.

Permissions can be assigned to AWS users, groups, and roles via policies. These permissions can be given with policies in one of two ways:

* **Identity-based policies**: In this case, the policies are attached to users, groups, or roles.
* **Resource-based policies**: In this case, the policies are attached to AWS resources such as Amazon S3 buckets, Amazon EC2 instances, and AWS Lambda functions.

Identity-based policies are attached to an AWS identity and grant permissions to the identity. AWS identities include IAM users, IAM roles, and AWS service accounts. Identity-based policies can be used to grant permissions to IAM users or roles in your AWS account or to grant permissions to other AWS accounts.

Resource-based policies are attached to a resource, such as an Amazon S3 bucket or an Amazon SNS topic, and grant permissions for the resource. These permissions can be used by any AWS identity that has access to the resource.

Both types of policies use the same syntax and structure, and you can use them together to fine-tune access to your resources. It is important to choose the right type of policy for your use case and to design your policies carefully to ensure that they provide the right level of access to your resources. Now, let’s learn about how to manage multiple AWS accounts using AWS Organizations.

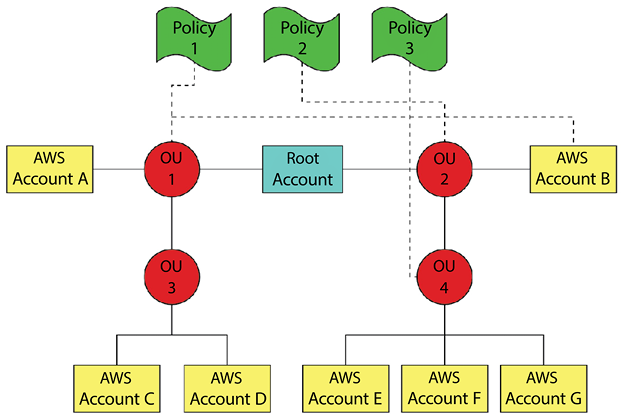
**AWS Organizations**

AWS Organizations is a service that can be used to manage multiple AWS accounts in a consolidated manner. It provides a centralized location where you can see all your organization’s bills and manage all your AWS accounts from one place. This central location makes it much easier to establish, manage, and enforce your organization’s security policies. This central control ensures that security administrators and auditors can perform their jobs more efficiently and confidently.

These are the most important and relevant concepts when working with the AWS Organizations service:

* **Organization**: The overarching owner that will control all AWS accounts.
* **Root account**: The owner account for all other AWS accounts. Only one root account can exist across the organization. The root account needs to be created when the organization is created.
* **Organizational unit**: A grouping of underlying AWS accounts and/or other organizational units. An organizational unit can be the parent of other organizational units. This enables the potential creation of a hierarchy of organizational units that resembles a family tree. See the following diagram for more clarity.
* **AWS account**: A traditional AWS account that manages AWS resources and services. AWS accounts reside under an organizational unit or the root account.
* **Service Control Policy (SCP)**: An SCP specifies the services and permissions for users and roles. An SCP can be associated with an AWS account or organizational unit.

The following figure illustrates how these components interact with each other:

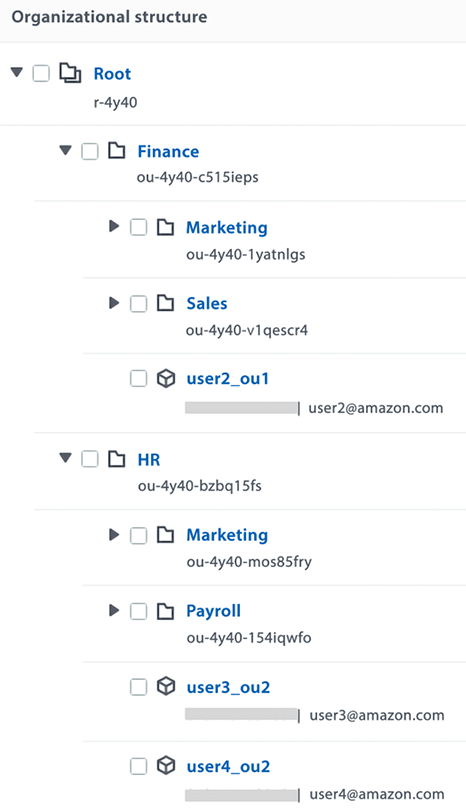


*Figure 8.6: Sample organizational unit hierarchy*

As you can see in the diagram above, **Policy 1** is associated with **Organizational Unit** (**OU**)**1** and with **AWS Account B**. **Policy 1** is also applied to all children of **OU 1** (**OU 3**, **AWS Account C**, and **AWS Account D**).

Since **Policy 1** is associated with **AWS Account B** directly, it overrides **Policy 2**, which is associated with **OU 2** and all its children except for **AWS Account B**. **Policy 3** is associated with **OU 4** and all its children (**AWS Accounts E**, **F**, and **G**).

The following diagram shows an AWS organizational structure created in the AWS console:



*Figure 8.7: AWS organizational unit hierarchy in an AWS account*

As you can see in the preceding diagram, two OUs are under the root account, and each unit has its sub-unit and AWS accounts.

The following are the key benefits of AWS Organizations:

* It provides tools to centrally govern and manage your cloud environment. You can quickly scale by creating accounts and allocating resources. You can provision common resources and permissions to new and existing accounts using AWS CloudFormation StackSets.
* You can customize your environment by applying governance policies. You can simplify access management for users by providing cross-application permissions with your identity source and AWS’s SSO service.
* You can secure and audit your environment by logging all events in AWS CloudTrail.
* You can apply scheduled backups with AWS Backup for all accounts in the organization.
* You can manage costs and identify cost-saving measures. You can consolidate billing charges for your organization, and take advantage of volume discounts for qualifying services.

Without AWS Organization’s SCP, all these policies would have to be repeated individually for each account. Every time there was a change to a policy, it would have to be changed individually in each account. This old approach had a high likelihood of policies that were supposed to be identical getting out of sync. You can learn more about AWS Organizations by visiting the AWS page here: <https://aws.amazon.com/organizations/>.

Managing multiple accounts could be complicated. If you’d like to start your AWS environment using a simple UI and built-in best practices it’s better to use AWS Control Tower.

When you assign permission to a user or resource, you want to see the policy evaluation and how it will work. You can refer to details on IAM policy evaluation here: <https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html>.

Hopefully, the concepts of users, groups, roles, permissions, and policies are clearer now. IAM is far from the only security service that AWS offers. AWS IAM is a vast topic that warrants a book in itself. You can find more detail about AWS IAM here: <https://aws.amazon.com/iam/>. Let’s learn about the next service in the IAM category: AWS Directory Service.

**AWS Directory Service**

Microsoft Active Directory has been a popular choice for user and role management for decades, which, in computer years, is a long time. Given this popularity, AWS offers a fully managed implementation of Microsoft Active Directory. AWS Directory Service for Microsoft AD, also known as **AWS Managed Microsoft Active Directory**, allows AWS services that require directory services to integrate with Microsoft Active Directory.

AWS Managed Microsoft AD uses the actual Microsoft AD. It does not need to stay in sync because it does not copy the contents of existing ADs to the cloud. For this reason, the standard Microsoft AD administration tools can be used, and you can leverage the built-in AD capabilities, such as group policies and single sign-on. Using AWS Managed Microsoft AD, you can integrate Amazon EC2 and Amazon RDS for SQL Server instances with Microsoft AD. You can learn more about AWS Directory Service here: <https://aws.amazon.com/directoryservice/>. Let’s learn about how AWS offers support for single sign-ons.

**AWS IAM Identity Center (successor to AWS SSO)**

Being able to sign on to multiple enterprise applications using a user’s network login ID has been a pervasive way to manage application access for a while now.

AWS IAM Identity Center has replaced AWS Single Sign-On, and offers a secure way to establish and link workforce identities, as well as to centrally manage their access across AWS accounts and applications. With AWS IAM Identity Center, SSO can be implemented without too much effort, and it can also be centrally managed, even in a multi-account AWS environment. It can be used to manage user access and permissions for multiple AWS accounts in one central place by leveraging AWS Organizations. IAM Identity Center can be used to configure and maintain all account permissions automatically. It does not need additional configuration for each account. User permissions can be assigned using roles. The following are the benefits of IAM Identity Center:

* Manage users and groups where you want; connect to AWS once
* Centrally assign and manage access to AWS accounts; AWS SSO-integrated and cloud-based business applications
* Provide an SSO user portal to assigned AWS accounts; AWS and business applications
* It works with existing processes for joiners, leavers, and movers
* Increase developer productivity with the AWS CLI v2

You can use the following simple steps to set up application access through single sign-on:

1. Choose an identity resource, which could be IAM users, groups, or resources
2. Define permission sets for each role
3. Assign groups/users to permission sets in selected accounts
4. Connect cloud apps with SAML
5. Assign groups/users to apps

To manage user identities, IAM Identity Center provides an identity store or can connect with an existing identity store. Some of the supported identity stores are the following:

* Microsoft Active Directory
* Azure Active Directory
* Okta Universal Directory

Any activity that occurs when using IAM Identity Center will be recorded using AWS CloudTrail. You can find more details about configuring SSO using AWS IAM Identity Center here: <https://aws.amazon.com/iam/identity-center/>. You have now learned how to manage access for multiple users.

**AWS Control Tower**

If you have a simple AWS setup with a few servers and only one AWS account, then you don’t need AWS Control Tower. But if you are part of an environment with hundreds or thousands of resources and multiple AWS accounts and teams, then you will want to learn about and leverage AWS Control Tower. AWS Control Tower simplifies a multi-account environment’s administration, governance, and security setup.

Control Tower helps you quickly set up and govern multi-account environments securely. It automatically applies management features from existing AWS services, such as Organizations, AWS Config, and IAM Identity Center, and implements default account structure and governance policies based on AWS best practices from thousands of customers.

You can continue to use native features from Organizations, such as tag or backup policies, and integrated AWS services. AWS Control Tower enables you to set up company-wide policies and apply them across multiple AWS accounts. Without AWS Control Tower, you would have to apply the individual files to each account, opening up the possibility of having inconsistencies in your accounts. You can learn more about AWS Control Tower by visiting the AWS page here: <https://aws.amazon.com/controltower/>. Now let’s learn how to manage multiple resources across organizational units using AWS Resource Access Manager.

**AWS Resource Access Manager**

The AWS **Resource Access Manager** (**RAM**) service allows you to share AWS resources with other AWS accounts or within your own organization. RAM allows you to share resources such as Amazon EC2 instances, Amazon RDS database instances, and **Amazon Virtual Private Clouds** (**Amazon VPCs**) with other AWS accounts or within your organization.

You can use RAM to manage resource sharing by creating resource shares, which are collections of resources that you want to share with specific AWS accounts or within your organization. You can specify the accounts or **organizational units** (**OUs**) that you want to share the resources with, and set permissions to control how the resources can be accessed.

RAM is useful for scenarios where you want to share resources with other teams or organizations, or when you want to centralize the management of resource sharing within your organization. It helps you to simplify resource sharing, reduce the complexity of resource management, and maintain control over the resources that you share. Here are the steps you can follow to use AWS RAM:

1. Sign in to the AWS Management Console and open the AWS RAM console at <https://console.aws.amazon.com/ram/>.
2. In the left navigation pane, choose **Resource Shares**.
3. Choose **Create resource share**.
4. On the **Create resource share** page, enter a name and optional description for your resource share.
5. Select the resources that you want to share and specify the accounts or OUs that you want to share the resources with.
6. Choose **Create resource share**.
7. To view the status of the resource share, choose the resource share in the list and then choose the **Status** tab.
8. To modify the resource share, choose the resource share in the list and then choose the **Modify** tab.

Note that you can only share resources that support sharing, and some resources have additional sharing requirements. For example, you can’t share an EC2 instance unless it’s in a VPC. You can learn more about AWS RAM by visiting the AWS page here: <https://aws.amazon.com/ram/>.

As of now, you have learned that managing users’ security is the responsibility of your organization, but what if you are developing a web or mobile app open to the world? In those scenarios, you must manage millions of users, secure their credentials, and provide the required access. Amazon Cognito fulfills these needs. Let’s learn more about it.

**Amazon Cognito**

Amazon Cognito enables developers to add user sign-in, sign-up, and access control to their web and mobile apps. It provides granular APIs and SDKs to manage end-user authentication, and authorization workflows that can be customized using out-of-the-box integration with AWS Lambda.

Cognito is fully managed with a built-in hosted UI and provides out-of-the-box support for open standards authentication protocols such as OAuth 2.

You can easily integrate your app to authenticate users using federation with Facebook or login with Amazon, Google, and custom OpenID Connect or SAML providers. It provides a serverless fully managed directory to store and securely manage user information using MFA authentication through SMS and email. Amazon Cognito offers authentication, authorization, and user management services for web and mobile applications. Here are some of the security features of Amazon Cognito:

* **MFA**: Amazon Cognito supports MFA to help protect against unauthorized access to user accounts. MFA can be configured to require a one-time code sent to the user’s phone or email, or a hardware token.
* **Password policies**: Amazon Cognito allows you to set password policies to ensure that users choose strong passwords.
* **Encryption**: Amazon Cognito stores user data and passwords in an encrypted format, using AES-256 encryption.
* **Access control**: Amazon Cognito provides fine-grained access control to resources using IAM policies.
* **Activity tracking**: Amazon Cognito tracks user sign-in and sign-out activity, as well as changes to user attributes. This information can be used to monitor for suspicious activity and alert administrators.
* **Security tokens**: Amazon Cognito issues **JSON Web Tokens** (**JWTs**) to authenticated users, which can be used to access authorized resources. The JWTs have a limited lifespan and can be easily invalidated if a user’s security is compromised.
* **Account recovery**: Amazon Cognito provides options for users to recover their accounts if they forget their passwords or lose access to their MFA devices.

You can learn more about Amazon Cognito by visiting the AWS page here: <https://aws.amazon.com/cognito/>.

In this section about the security services in AWS’s IAM pillar you learned about managing user security in AWS. As AWS security is a vast topic that would require multiple books to cover in detail, in the upcoming section, you will learn a bit about each AWS service belonging to different security pillars with resources to learn more. Let’s learn about the next security pillar, which helps you detect and control security threats.

**Applying security controls**

Security is more about preventive gestures than reactive as any security incident can cause significant damage to organizations, so it’s better to detect and fix incidents before a security leak can cause damage. AWS provides an array of services to help you to monitor, detect, and mitigate security threats. The following is a summary of services to apply proactive security control:

* **Amazon GuardDuty**: Intelligent threat detection and continuous monitoring to protect your AWS accounts and workloads.
* **Amazon Inspector**: Automates security assessments to help improve the security and compliance of applications deployed on AWS.
* **AWS Security Hub**: Centrally view and manage security alerts and automate compliance checks.

The following are common security audit services:

* **AWS Config**: AWS Config enables you to assess, track, and evaluate the configurations of your AWS resources. With AWS Config, you can monitor the changes to your resources and assess their compliance with internal policies and regulatory standards. Additionally, you can use AWS Config to conduct security analysis and improve the visibility of your resource configurations, making it easier to identify potential security risks and respond to any incidents. Overall, AWS Config provides a centralized and automated way to manage the configuration of your AWS resources, ensuring that they remain compliant and secure over time. You can learn more about Config by visiting the AWS page here: <https://aws.amazon.com/config/>.
* **AWS CloudTrail**: AWS CloudTrail provides a record of all API calls made to your AWS account. This service enables you to monitor and audit your AWS resource activity, including changes to your resources and the actions of your users and applications. With AWS CloudTrail, you can gain greater visibility of your resource usage and security posture, as well as quickly identify and respond to any suspicious or unauthorized activity. AWS CloudTrail supports multiple platforms, including the AWS Management Console, AWS CLI, and AWS SDKs, making it easy to track and manage your AWS resource activity from a variety of sources. You can learn more about CloudTrail by visiting the AWS page here: <https://aws.amazon.com/cloudtrail/>.
* **Amazon VPC Flow Logs**: VPC Flow Logs enables you to capture information about the IP traffic going to and from network interfaces in your VPC.

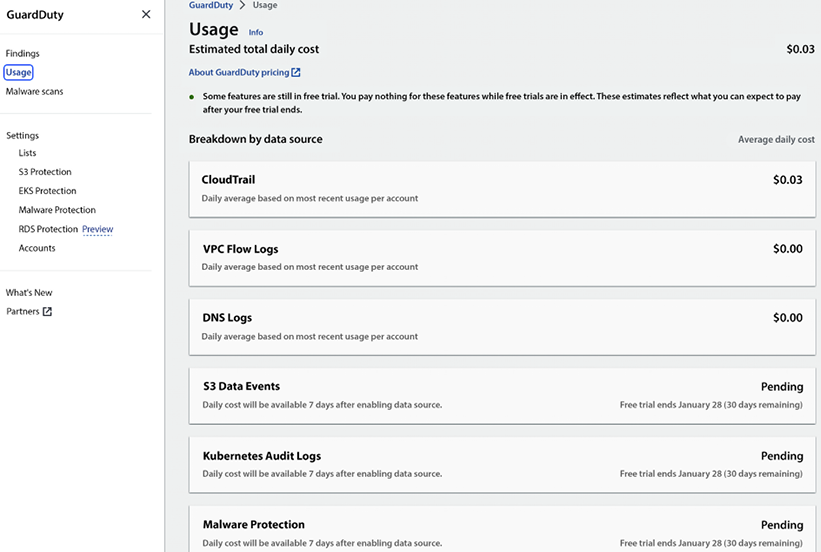
This service allows you to track the traffic flow and troubleshoot network connectivity issues, as well as improve the security of your network by identifying any potential network threats or unauthorized access. You can learn more about Flow Logs by visiting the AWS page here: <https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html>.

* **Amazon CloudWatch**: CloudWatch is a monitoring service that enables you to monitor your AWS resources and applications in real time. With CloudWatch, you can monitor metrics, logs, and events generated by your resources and applications, and set alarms to be notified of any issues. This service provides a centralized and automated way to monitor your environment, making it easier to detect and resolve issues, improve the performance of your resources, and ensure the availability of your applications. CloudWatch supports a variety of resources and services, including Amazon EC2 instances, Amazon RDS databases, AWS Lambda functions, and many more. You can also use CloudWatch to monitor custom metrics, such as the number of requests made to an application, the response time of a database, or the disk usage of an EC2 instance. You can learn more about CloudWatch by visiting the AWS page here: <https://aws.amazon.com/cloudwatch/>.

Let’s learn about security control services in detail.

**Amazon GuardDuty**

Amazon GuardDuty is an AWS service that can detect unauthorized actors’ threats, malicious behavior, and activity. It protects all other AWS resources and your enterprise’s data. Getting more traffic in the application is usually good news because it typically means more business. But additional traffic requires more work to track and monitor additional logs and activity. The following screenshot shows services being monitored by GuardDuty:



*Figure 8.8: Amazon GuardDuty data source list*

Amazon GuardDuty enables and simplifies the monitoring of this activity. Amazon GuardDuty leverages machine learning and advanced anomaly detection to compile, process, and prioritize potential malicious activity. GuardDuty can analyze billions of real-time events across various AWS real-time and near-real-time streams such as AWS CloudTrail logs, Amazon VPC Flow Logs, and DNS logs.

However, keep in mind that Amazon GuardDuty doesn’t do anything with the analysis. It is an intrusion detection system, not an intrusion prevention system. If you need enforcement for malicious IPs, you will need a third-party solution such as Aviatrix GuardDuty Enforcement. You can learn more about GuardDuty by visiting the AWS page here: <https://aws.amazon.com/guardduty/>.

**Amazon Inspector**

Amazon Inspector is a service that automatically checks application compliance against certain predefined policies and is used to increase compliance. Amazon Inspector can identify vulnerabilities, exposures, and deviations from predefined best practices. Once the assessment has been completed, the service generates a comprehensive report of security flaws and issues sorted by severity level. These findings can then be used to close these security gaps.

Amazon Inspector security assessments enable users to look for unauthorized network access to Amazon EC2 instances. It can find vulnerabilities in EC2 instances and containers. Amazon Inspector assessments are available as predefined rule components that can map to security best practices and vulnerability definitions. Some samples of predefined rules are as follows:

* Someone trying to access EC2 instances from outside your network
* If someone turns on the remote root login
* Identify software and operating system versions that are due for patching

These rules are constantly monitored and enhanced by the AWS security team. You can learn more about Amazon Inspector by visiting the AWS page here: <https://aws.amazon.com/inspector/>.

**Building infrastructure protection**

Infrastructure protection is the first line of defense when it comes to security. AWS secures physical infrastructure with multi-layered security in its data centers. However, securing the logical infrastructure boundary and network traffic becomes your responsibility as AWS provides you with more control to manage boundaries for your logical cloud infrastructure. AWS provides the following services to protect your cloud infrastructure:

* **AWS Web Application Firewall**: Web Application Firewall is a security service that helps protect your web applications from common web exploits, such as SQL injection and **cross-site scripting** (**XSS**) attacks. Web Application Firewall is a firewall that allows you to monitor HTTP and HTTPS requests made to your applications and block, allow, or count requests based on conditions that you define, such as IP addresses, headers, and content.
* **AWS Firewall Manager**: Firewall Manager is a security service that makes it easier to manage the firewall rules for your Amazon VPC security groups. With Firewall Manager, you can centrally manage the firewall rules across multiple AWS accounts and VPCs, ensuring that your security policies are consistent and up to date.
* **AWS Shield**: Shield is a managed **Distributed Denial of Service** (**DDoS**) protection service for AWS customers. DDoS attacks are a common type of cyber attack that attempt to overwhelm a website or online service with an excessive amount of traffic, making it unavailable to users.

Let’s learn about AWS infrastructure security services in detail.

**AWS Web Application Firewall (WAF)**

AWS **Web Application Firewall** (**WAF**), as the name implies, is a firewall for your web applications. It can create a layer of protection around your web applications and RESTful APIs. It guards against the most well-known web exploits. AWS WAF can be used to control network traffic. This traffic is controlled by creating rules. These rules can target well-known exploits such as SQL injections or XSS attacks.

Furthermore, these rules can be customized to filter transactions that meet user-defined patterns. AWS WAF has Managed Rules, which simplifies management. AWS can manage these rules, and AWS Marketplace sellers also offer preconfigured rules. AWS and Marketplace rules are constantly modified as new threats are identified. AWS WAF also provides an API to assist in developing, deploying, and maintaining these security rules.

AWS WAF can be deployed on the following:

* Amazon CloudFront
* Application Load Balancer
* Origin servers running on EC2
* Amazon API Gateway
* AWS AppSync

AWS WAF pricing depends on the number of rules deployed and the number of requests that applications receive. You can learn more about WAF by visiting the AWS page here: <https://aws.amazon.com/waf/>.

**AWS Firewall Manager**

AWS Firewall Manager makes setting up firewalls simple. It enables users to administer firewall rules in a central dashboard. This can be achieved even across multiple AWS accounts and applications.

Cloud environments are dynamic. This can create maintenance headaches as new applications come online. AWS Firewall Manager simplifies the process of provisioning new applications and ensuring they comply with an enterprise’s security policies by enabling users to manage firewall settings from one location.

If new security rules need to be created or if existing rules need to be modified, they can also be changed only once. Some of the services in AWS that can benefit from AWS Firewall Manager are the following:

* Application Load Balancer
* API Gateway
* Amazon CloudFront distributions
* Amazon EC2

AWS Firewall Manager allows adding AWS WAF rules, AWS Shield Advanced protection, security groups, and AWS Network Firewall rules to VPCs across accounts and resources using a centralized dashboard. You can learn more about Firewall Manager by visiting the AWS page here: <https://aws.amazon.com/firewall-manager/>.

**AWS Shield**

AWS Shield is an AWS-managed DDoS protection service used to protect systems and data. AWS Shield delivers automatic attack detection and resolution that can keep your application running, or at least reduce the amount of downtime. Since AWS Shield Standard comes with all AWS accounts, you normally have to contact AWS support to assist you if you suffer a DDoS attack. AWS Shield comes in two flavors:

* **AWS Shield Standard**: Provided at no additional charge to all AWS customers
* **AWS Shield Advanced**: Provides a higher level of protection but at an additional cost

AWS Shield Standard can protect against and handle the more common types of attacks. The more common DDoS attacks happen at the network and transport layer. AWS Shield Standard can help you protect Amazon CloudFront and Amazon Route 53 against (Layer 3 and Layer 4) attacks.

AWS Shield Advanced provides higher protection for more services. AWS Shield Advanced can be used to defend against attacks targeting the following:

* Amazon EC2 instances
* Elastic Load Balancing
* Amazon CloudFront
* AWS Global Accelerator
* Amazon Route 53

To get this level of protection, you will need to subscribe to AWS Shield Advanced and pay an additional fee. AWS Shield Advanced not only protects against network and transport layer attacks, but also delivers additional monitoring and resolution, protecting against large and sophisticated DDoS attacks and providing real-time reporting when attacks occur. It integrates with AWS WAF. AWS Shield Advanced provides 24-hour support from AWS’s DDoS response team as an additional feature. Finally, with AWS Shield Advanced, AWS will cover any charges your account incurs for certain services that can be attributed to an attack. You can learn more about Shield by visiting the AWS page here: <https://aws.amazon.com/shield/>.

Data is the essential thing that any organization wants to protect. Let’s learn about AWS services available for data protection.

**Building data protection**

Data is key for any application or organization. Most hacking attempts made to steal data, and leakage of your customer data, can be very harmful to your organization in terms of customer trust and financial damage. You need to have multi-layer security to protect your customer data. As you are the owner of the data, most of the time the responsibility for data protection lies with you. AWS provides a number of services to protect data. Let’s look at them below:

* **Amazon Macie**: Amazon Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Macie provides visibility of how your data is being stored and accessed, helping you to identify and secure sensitive data, such as credit card numbers, Social Security numbers, and intellectual property.
* **AWS Key Management Service** (**KMS**): KMS is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. KMS provides a central repository for your encryption keys, making it easier to manage their lifecycle, such as rotating your keys on a regular basis and enforcing security and compliance policies.
* **AWS CloudHSM**: CloudHSM is a **hardware security module** (**HSM**) service that provides secure key storage for cryptographic operations within the AWS cloud. An HSM is a dedicated, tamper-resistant hardware device that provides secure storage for encryption keys and performs encryption and decryption operations.
* **AWS Certificate Manager** (**ACM**): **ACM** is a service that makes it easy for you to manage SSL/TLS certificates for your AWS-hosted websites and applications. ACM provides a simple and cost-effective way to manage SSL/TLS certificates, eliminating the need to manage and maintain your own certificate infrastructure.
* **AWS Secrets Manager**: Secrets Manager is a secure and scalable service that enables you to store, manage, and rotate your secrets, such as database credentials, API keys, and SSH keys. With Secrets Manager, you can store your secrets in a centralized and encrypted manner, making it easier for you to manage and secure access to your secrets.
* **Server-Side Encryption** (**SSE**): SSE is a security feature for encrypting data stored in AWS storage services, such as Amazon S3 and Amazon EBS. SSE helps protect your data from unauthorized access by automatically encrypting your data at rest, which means that the data is encrypted when it is stored on disk. You learned about SSE in *Chapter 5*, *Storage in AWS – Choosing the Right Tool for the Job*. You can explore more about SSE by visiting the SSE page here: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html>.

Let’s learn about some of these services in more detail.

**Amazon Macie**

Amazon Macie is another fully managed security service. It can be used to protect your data and its privacy. It leverages artificial intelligence and machine learning to find and protect sensitive data in AWS environments.

In today’s enterprises, data comes in at an ever-increasing speed. Handling those growing volumes of data creates scalability issues with more data and complexity, making expenses increase. Amazon Macie enables the automation of sensitive data discovery. Since it leverages machine learning, it can scale and handle petabyte-sized datasets. Macie creates a list of Amazon S3 buckets in a user’s account. It can flag which ones are unencrypted, which ones can be accessed publicly, and buckets that are being shared with other AWS accounts that are not defined in AWS Organizations.

Amazon Macie uses machine learning and pattern matching on these buckets. Amazon Macie can be configured to identify sensitive data such as personally identifiable information and deliver alerts to a predefined user base. Once these alerts and issues are generated, they can be quickly sorted and filtered in the AWS Management Console. It can then be integrated with other AWS services using workflow or event management systems. It can also be used together with other AWS services.

An example is AWS Step Functions. AWS Step Functions can leverage automated remediation actions. This can assist with compliance with rules and regulations, such as the **Health Insurance Portability and Accountability Act** (**HIPAA**) and **General Data Privacy Regulation** (**GDPR**). You can learn more about Macie by visiting the AWS page here: <https://aws.amazon.com/macie/>.

**AWS Key Management Service**

KMS, or Key Management Service, is a service offered by AWS that simplifies the process of creating and managing encryption keys. KMS provides a central, secure location for storing and managing your encryption keys, and it integrates with other AWS services to help you easily encrypt and decrypt data in the cloud. It is a secure and fault-tolerant service. AWS KMS can be used to assist in the management of encryption of data at rest. KMS provides the ability to create and manage cryptographic keys. It can also be used to manage which users, services, and applications have access to them.

Behind the scenes KMS uses HSMs to protect your encryption keys, ensuring that they are kept secure even if an attacker gains access to your systems. It also provides auditing and logging capabilities to help you track the use of your keys and meet compliance requirements.

You can use KMS to encrypt data in a number of different ways, including:

* Encrypting data at rest, such as data stored in Amazon S3 or Amazon EBS.
* Encrypting data in transit, such as data transmitted over the network or data transmitted between AWS regions.
* Encrypting data in use, such as data stored in memory or data being processed by an application.

The HSMs that KMS uses comply with Federal Information Processing Standard 140-2. AWS KMS integrates with AWS CloudTrail so that it is simple to see who has used the keys and when. You can learn more about KMS by visiting the AWS page here: <https://aws.amazon.com/kms/>.

**AWS CloudHSM**

AWS makes encrypting data at rest quite simple if you use encryption keys provided by AWS through KMS. However, KMS works under the shared tenancy model, which means that behind the scenes a single HSM may be storing keys from different customers. In some instances, such as in the finance industry, you cannot use shared storage to store encryption keys in order to comply with regulations, which state you have to have your own dedicated HSM for key storage.

For that, AWS provides a service called AWS CloudHSM.

AWS CloudHSM is a **hardware security module** (**HSM**) that empowers users to generate their own encryption keys. CloudHSM provides the ability to create encryption keys using Federal Information Processing Standard 140-2 Level 3 validated HSMs. AWS CloudHSM can be integrated with other AWS services via well-defined industry-standard APIs. Some of the APIs supported are as follows:

* PKCS#11
* Microsoft **CryptoNG** (**CNG**) libraries
* **Java Cryptography Extensions** (**JCEs**)

AWS CloudHSM complies with many security standards. It is also possible to export the generated keys to various third-party HSMs. Like many of the other security services we have learned about in this section, AWS CloudHSM is fully managed by AWS, enabling you to focus on your applications and not the administration of your key management service. Some of the tasks that AWS handles when using this service are as follows:

* Provisioning the required hardware to run the service
* Applying software patching
* Making sure the service is highly available
* Performing backups

CloudHSM has a serverless architecture that allows users to seamlessly and effortlessly scale. Like other cloud services, you can use CloudHSM with an on-demand, pay-as-you-go model. You can learn more about CloudHSM by visiting the AWS page here: <https://aws.amazon.com/cloudhsm/>.

**AWS Certificate Manager**

AWS Certificate Manager is another security service. It can create, maintain, and deploy public and private SSL/TLS certificates that can be added to other AWS services and applications. SSL/TLS certificates can secure network communications by enabling encryption. They can also be used to authenticate a website’s identity in public and private networks. AWS Certificate Manager streamlines and automates the management process for certificate management.

AWS Certificate Manager can be used to provision and renew a certificate and install it on another AWS service such as Elastic Load Balancing, Amazon CloudFront, and APIs on API Gateway. It can also be used to create private certificates for internal applications. These certificates can then be centrally managed.

There is no charge when you provision public and private certificates using AWS Certificate Manager. The cost is bundled with spinning up the underlying resources (like an EC2 instance). When you use AWS Certificate Manager Private Certificate Authority, there is a monthly charge for the use of Private Certificate Authority and for the private certificates that are issued. You can learn more about Certificate Manager by visiting the AWS page here: <https://aws.amazon.com/certificate-manager/>.

**AWS Secrets Manager**

AWS Secrets Manager is a security service that can be used to protect secrets. These secrets may be strings such as passwords that can be used to access services, applications, and IT resources. AWS Secrets Manager facilitates the rotation, management, and retrieval of API keys, database credentials, passwords, and other secrets. These secrets can be retrieved using the Secrets Manager APIs. The need to store passwords clearly in plain text files is obviated by using AWS Secrets Manager. Some of the services that can integrate with AWS Secrets Manager are as follows:

* Amazon RDS
* Amazon Redshift
* Amazon DynamoDB
* Amazon Neptune
* Amazon DocumentDB

AWS Secrets Manager can be customized to support additional types of secrets. Some examples of use cases are the following:

* API keys
* OAuth authentication tokens

Another feature of AWS Secrets Manager is that it allows secrets to be rotated periodically without impacting applications that use them for password management and other uses.

You can learn more about Secrets Manager by visiting the AWS page here: <https://aws.amazon.com/secrets-manager/>.

There are so many AWS services that collect data, and security needs to look across all the logs and data collected. However, collecting data across services such as VPC Flow Logs, AWS CloudTrail logs, audit logs, GuardDuty, and so on, could be very tedious. You need a unified view of logs to understand any security issues. AWS provides a way to quickly analyze the issue through a service called Amazon Detective. Let’s learn about Amazon Detective in more detail.

**Amazon Detective**

Amazon Detective is a security solution that employs machine learning, statistical analysis, and graph theory to help customers identify and investigate security issues in their AWS accounts. This service provides a powerful tool for analyzing and understanding security-related activity in your AWS environment, making it easier for you to quickly identify the root cause of any suspicious activity or security incidents. It can be used to identify unusual activity or suspicious behavior in your account, such as resource provisioning or access patterns that deviate from normal behavior.

To use Amazon Detective, you first need to enable the service in your AWS account and then connect your AWS resources, such as Amazon EC2 instances and Amazon RDS databases, to it. Amazon Detective then analyzes log data from your AWS resources, creates a linked set of data that provides you with a comprehensive view of your security posture, and builds a graph of the interactions and relationships between them. It uses machine learning algorithms to identify patterns and anomalies in the data that may indicate security issues or suspicious activity.

Once Amazon Detective has identified a potential issue, it provides a detailed investigation summary that includes a timeline of events, relevant log data, and recommended actions for further investigation or remediation. You can use this summary to quickly understand the issue and take the appropriate action to resolve it.

With Amazon Detective, you can use advanced algorithms to analyze security-related activity in your AWS environment and gain insights into potential security risks. The service also provides visualizations and summaries that help you triage security findings and prioritize your investigations, making it easier for you to focus on the most critical issues.

By automating the collection and analysis of security data, Amazon Detective helps you streamline your security investigations and resolve security incidents more quickly. This helps you reduce the risk of security breaches and ensure that your AWS environment remains secure and compliant. You can learn more about Amazon Detective by visiting the AWS page here: <https://aws.amazon.com/detective/>.

**AWS Security Hub**

AWS Security Hub is a security management service that provides a central place to manage security alerts and findings from multiple AWS services, as well as from other **AWS Partner Network** (**APN**) security solutions. It provides a comprehensive view of your security posture across your AWS accounts, making it easier to identify and prioritize security issues.

Security Hub integrates with a number of AWS services, including Amazon GuardDuty, Amazon Inspector, and Amazon Macie, as well as third-party security solutions from APN partners. It also provides APIs that enable you to automate the process of responding to security findings. With Security Hub, you can:

* Consolidate security alerts and findings from multiple sources into a single view.
* Prioritize and triage security issues based on their severity and the likelihood of impact.
* Automate the process of responding to security findings, using AWS Config rules and AWS Lambda functions.
* Collaborate with your team to investigate and resolve security issues.

Security Hub helps you improve your organization’s security posture by providing a central place to manage security alerts and findings, and by enabling you to automate the process of responding to security issues. You can learn more about Security Hub by visiting the AWS page here: <https://aws.amazon.com/security-hub/>.

While security is essential, multiple compliances are defined by the local governing body to which your application must adhere. Let’s learn about AWS-provided services to fulfill your compliance needs.

**Adhering to compliances**

AWS offers a variety of services and tools to help organizations comply with a wide range of regulations and standards, such as the **General Data Protection Regulation** (**GDPR**), the **Health Insurance Portability and Accountability Act** (**HIPAA**), and the **Payment Card Industry Data Security Standard** (**PCI DSS**).

AWS provides broad support for security standards and compliance certifications, including HITECH, FedRAMP, GDPR, FIPS 140-2, and NIST 800-171, to help meet the compliance requirements of regulatory agencies around the world. This extensive coverage makes it easier for organizations to achieve and maintain compliance with a wide range of security standards and regulations, regardless of their location or the specific regulatory requirements they must adhere to.

By supporting these security standards and certifications, AWS helps organizations ensure that their data and applications are protected by rigorous security controls and processes, reducing the risk of data breaches and security incidents. This helps organizations meet their regulatory obligations and maintain the trust of their customers, employees, and stakeholders.

The following are the services provided by AWS to audit your compliance needs:

* **AWS Artifact**: No-cost, self-service portal for on-demand access to AWS compliance reports.
* **AWS Audit Manager**: Amazon Audit Manager is a security and compliance auditing service that makes it easy for customers to automate the process of auditing their AWS accounts. With Audit Manager, customers can assess their compliance with AWS security best practices, industry standards and regulations, and internal policies. You can learn more about AWS Audit Manager by visiting the AWS page here: <https://aws.amazon.com/audit-manager/>.
* **AWS IAM**: AWS IAM allows you to set up and manage users and their permissions within your AWS account. This can help you ensure that only authorized users have access to sensitive resources and data.
* **AWS Config**: Config provides visibility of resource configurations and changes in your AWS environment. It can help you track changes to resources, ensure that resources are compliant with your internal policies, and audit resource configurations for compliance with external regulations.
* **AWS Encryption SDK**: SDK is a set of libraries that you can use to build encryption into your applications. It helps you protect data in transit and at rest, and can be used to meet compliance requirements for data encryption.
* **AWS PrivateLink**: PrivateLink is a network interface that you can use to connect your on-premises data centers to AWS services, without the data traversing the public internet. This can help you comply with regulations that require the use of private networks for certain types of data.

There are many other AWS services and features that can help with compliance, depending on your specific needs. If you have specific questions about how AWS can help with compliance in your organization, you can contact AWS Support or consult with a security and compliance specialist.

Let’s learn about AWS Artifact in detail.

**AWS Artifact reports**

AWS Artifact is a portal that provides on-demand access to AWS’s security and compliance documents. It includes AWS compliance reports, Service Organization Control reports, and other documents that can be used to demonstrate compliance with various regulations and standards.

AWS Artifact Reports deliver a centralized repository to store, manage, and access a variety of compliance reports from third-party auditors who have audited and certified that a given standard or regulation is met by the AWS infrastructure or by a given service. These rules, standards, and regulations may be global, regional, or industry-specific. As these rules and regulations change, AWS is constantly engaging third parties to ensure that compliance is up to date.

The AWS Artifact Agreements service provides the ability to access, approve, terminate, and manage agreements with AWS. It can be used to manage one AWS account or leverage AWS Organizations to manage multiple AWS accounts.

Some of the types of reports that can be managed with AWS Artifact are as follows:

* Service Organization Control reports
* Payment Card Industry reports
* Certifications from accreditation agencies around the world
* Industry-specific compliance reports
* Compliance reports about AWS security controls
* Business Associate Addendums
* Nondisclosure agreements

You can learn more about AWS Artifact by visiting the AWS page here: <https://aws.amazon.com/artifact/>.

In this section, you have learned about AWS security and compliance services. Let’s look at the “best of the best” tips for security in the AWS cloud.

**Best practices for AWS security**

While AWS provides a number of security services, it’s essential to understand how to apply them to secure your application. AWS offers a wide range of security features and services, but customers are responsible for properly configuring and managing these features to meet their specific security requirements. Here are some best practices for AWS security:

1. Implement a strong IAM policy:
   * Use MFA for privileged users.
   * Grant least privilege, meaning only provide access to the resources and actions that users need to do their jobs.
   * Regularly rotate AWS access keys and passwords to reduce the risk of compromise.
2. Encrypt data at rest and in transit:
   * Encrypt sensitive data, such as databases and backups, using AWS KMS or SSE.
   * Use SSL or TLS to encrypt data in transit.
3. Monitor and log activity:
   * Use Amazon CloudWatch Logs to monitor and store log data from EC2 instances, ELB, and other AWS services.
   * Use Amazon CloudTrail to track API activity and changes to your AWS resources.
   * Enable VPC Flow Logs to monitor network traffic in your VPC.
4. Implement security groups and **network access control lists** (**NACLs**):
   * Use security groups to control inbound and outbound network traffic to your EC2 instances.
   * Use NACLs to control traffic to and from subnets within your VPC.
5. Use Amazon VPC to segment your network:
   * Use a VPC to create isolated networks and control access to your AWS resources.
   * Use AWS VPC peering to connect multiple VPCs for communication between resources.
6. Use AWS Shield for DDoS protection:
   * AWS Shield provides protection against DDoS attacks.
   * Consider using AWS WAF to protect against web-based attacks.
7. Implement Amazon Macie for data protection:
   * Use Macie to discover, classify, and protect sensitive data in AWS.
   * Use Macie to monitor access to your data and detect any unauthorized access or data leaks.
8. Use AWS Config to track resource configurations:
   * AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources.
   * Use AWS Config to monitor changes to your resources and ensure that configurations remain compliant with your policies.

In conclusion, security is a top priority for AWS, and there are many best practices that customers can follow to secure their AWS environments. By implementing these best practices, customers can ensure that their AWS resources are secure and protected against a variety of security threats.

Overall, more automation improves security outcomes. You should minimize human intervention and always make smaller changes and do these more often to stay on top of vulnerabilities as quickly as they are discovered.

**Summary**

In this chapter, we laid the groundwork to enable you to understand how security is implemented in AWS. As we saw in the chapter, the shared responsibility model is a fundamental pillar. You saw how some components of security are the responsibility of AWS, and some parts are the customer’s responsibility. You learned about the six security pillars into which AWS security and compliance services are divided.

You then looked at the most basic and fundamental security service in IAM. You dove deep into AWS IAM and reviewed concepts such as users, groups, permissions, roles, and policies and how they are connected to each other. Further, you briefly learned about security services in each security pillar with available resources to explore. Finally, in the last section of the chapter, you learned about security best practices and how to make your cloud environment even more secure.

Hopefully, after completing this chapter, you feel more confident about how AWS can be leveraged to write world-class applications offering the highest levels of security.

In the next chapter, you will further explore some more elements of cloud automation.

w