AWS Solution Architect

TECHNOLOGY

Security and IAM

simplilearn

A Day in the Life of a Cloud Architect

Mr. William works for an IT company as a cloud architect. The company has several cloud-based applications. Recently, the company has been facing security and data privacy issues. The company is facing a huge loss due to this.

Now, the company wants Mr. William to certify that these types of failures should not occur in the future.

In this lesson, he will get a brief idea about how to protect AWS accounts with threat detection, management of security, and data privacy.

Learning Objectives

By the end of this lesson, you will be able to:

- Illustrate and configure models in AWS
- Define Identity and Access Management in AWS
- Protect the AWS accounts with intelligent threat detection using Amazon GuardDuty
- Manage data security and data privacy using AWS Macie

TECHNOLOGY

Responsibility Models in AWS

Introduction to Shared Responsibility Model

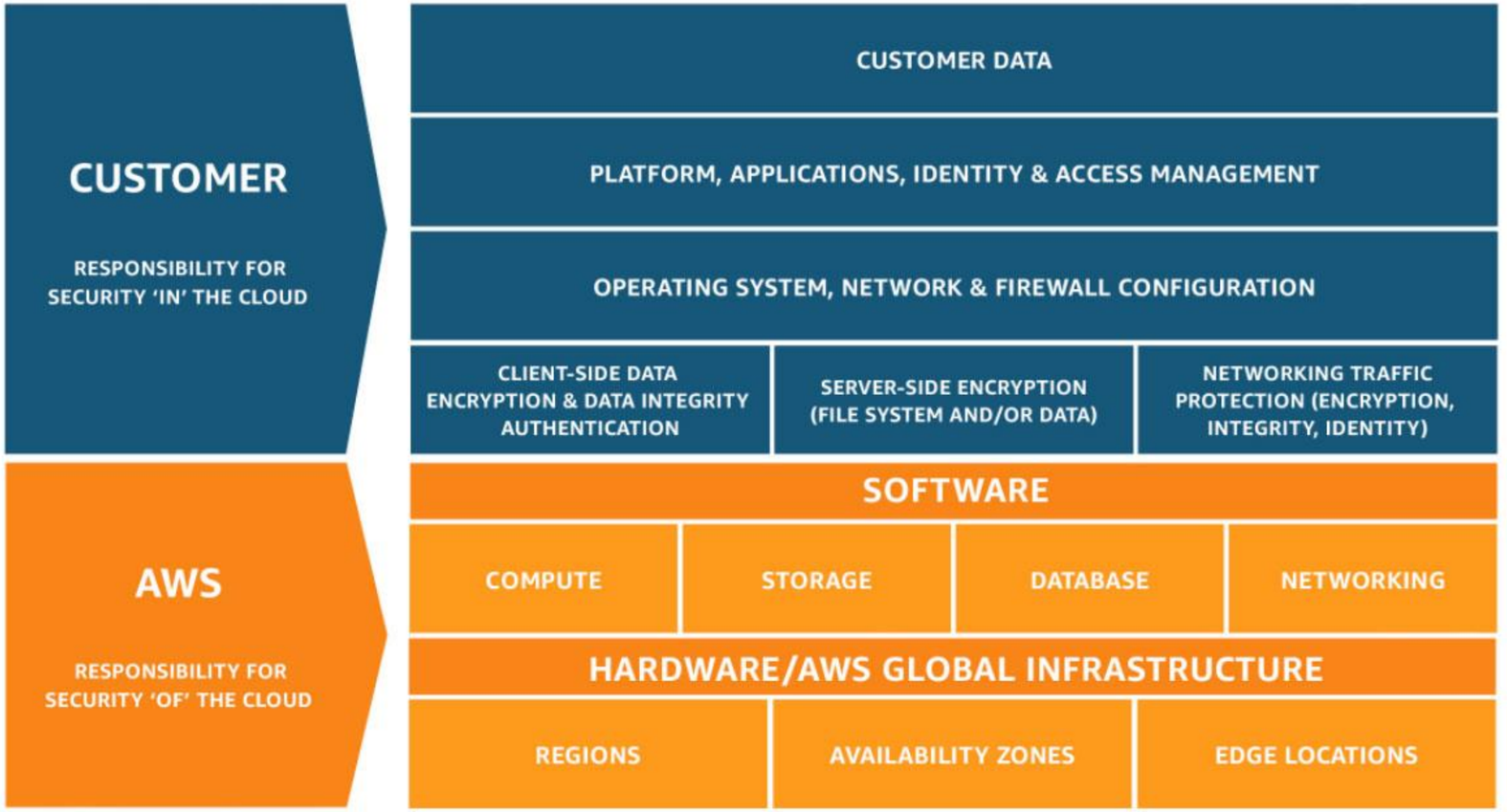
AWS and the customer share responsibility for security and compliance.

This shared approach can reduce the operational load placed on the consumer.

The host operating system, virtualization layer, and the physical security of the buildings are all managed.

Introduction to Shared Responsibility Model

AWS shared responsibility model:



https://docs.aws.amazon.com/whitepapers/latest/aws-risk-and-compliance/images/image2.png

Compliance Program

AWS and clients share compliance responsibilities when systems are built in the AWS Cloud.

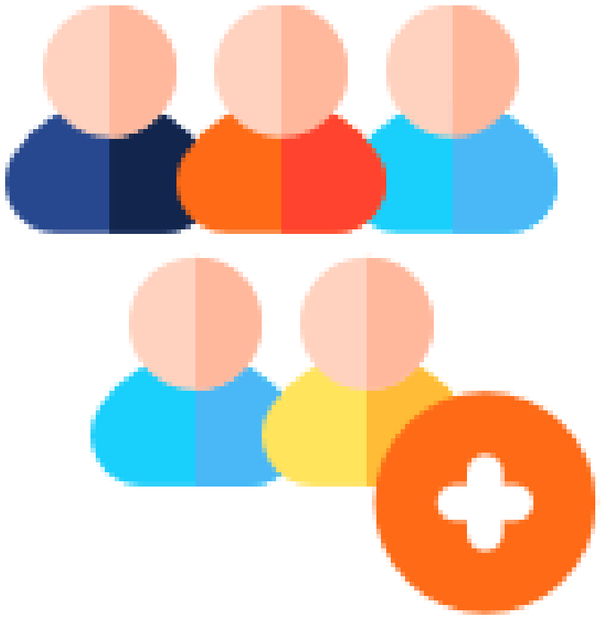
- Customers can learn more about the strict measures that AWS has in place to maintain security.
- AWS Compliance Enablers expand on conventional programs by connecting governance-focused, audit-friendly service features with relevant compliance.

Compliance Reports

Following are the AWS artifact:

- On-demand access to AWS
- Compliance reports
- Globally available
- Easy identification
- Quick assessments
- Continuous monitoring
- Enhanced transparency

Delegation

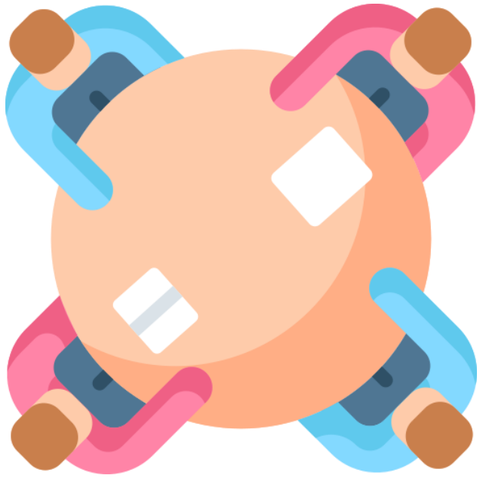


A registered member account can be used for administrative AWS SSO activities.

The management account in AWS Organizations automatically creates an SSO instance.

Roles can be created, deleted, and updated across all of the organization's member accounts using AWS SSO.

Benefits of Delegation



Minimizes the number of individuals who need access to the management account to assist reduce security risks.

Enables a restricted number of administrators to allocate users and groups to applications and user accounts for organization members.

Federation

Identity federation is a trust-based system that links two parties together to authenticate users and transmit the data necessary to grant them access to resources.

Enabling Federated AWS Access

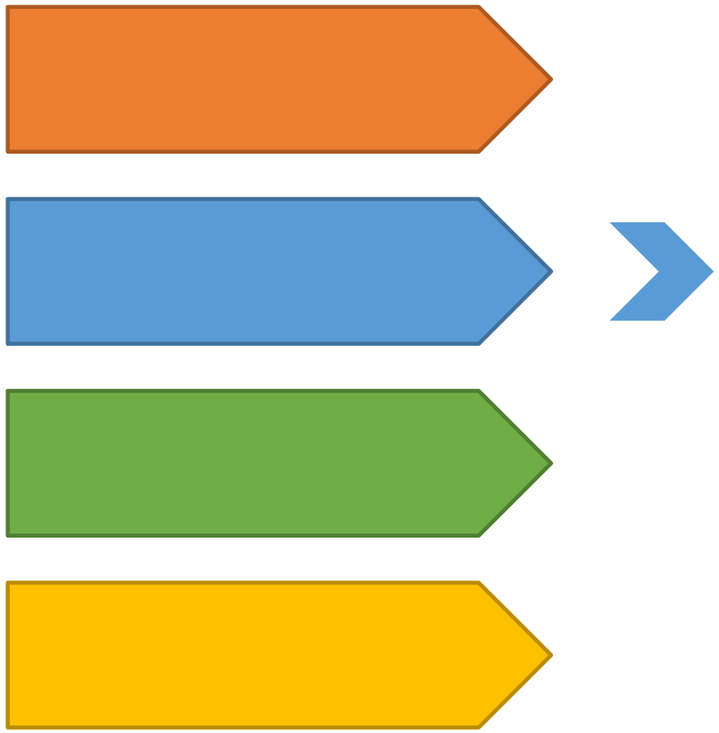
- Using federation, users can access their AWS accounts using single sign-on (SSO).
- Utilizing AWS IAM to control, federated, fine-grained access to AWS accounts.
- Enabling federated access to user-facing mobile and online applications.

Security Is Job Zero-AWS Security Process

Physical security

- Perimeter Layer security is the first layer of physical protection for AWS data centers.
- Depending on the area, this layer can contain security guards, fencing, security feeds, and intrusion detection systems.

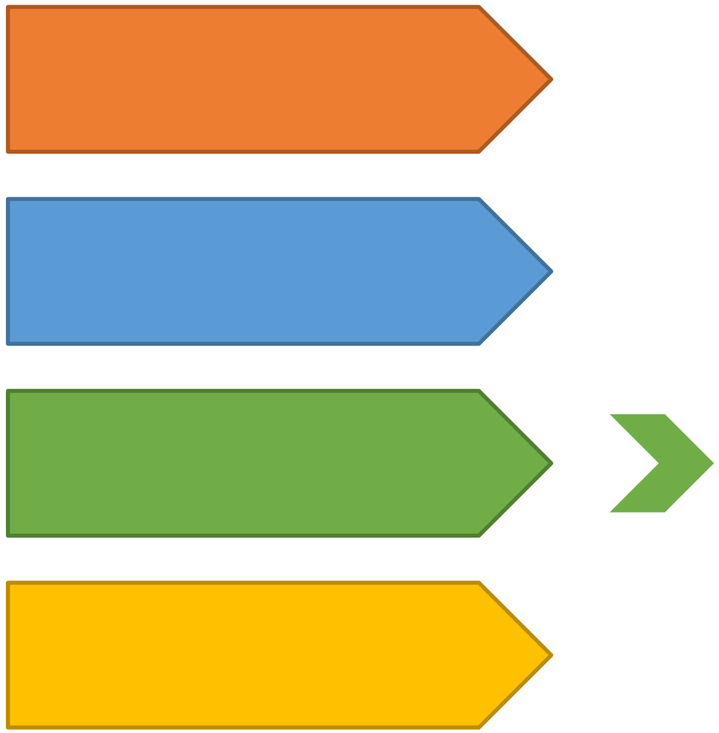
Security Is Job Zero-AWS Security Process



Network security

AWS's Network and Application Protection services give the user the power to apply fine-grained security controls to each network control point in their company.

Security Is Job Zero-AWS Security Process



Platform security

- **Compliance**: Users will inherit the most effective compliance controls with AWS.
- **Encryption:** AWS provides APIs that enable users to integrate encryption and data security into any service they design or deploy in an AWS environment.

Security Is Job Zero-AWS Security Process

People & Procedures

- Identity and Access Management (IAM)
- SSO
- MFA
- Trusted identities/ federated identities

TECHNOLOGY

AWS IAM

Security Credentials

AWS offers a variety of options for granting users secure access to their AWS resources:

Email address and password:
They are used to sign in to the AWS Console.

IAM username and password:
They enable access to an AWS account for various users and programs.

Access keys:
Access keys can be used to grant access to programmatic requests.

AWS Multi Factor Authentication

Multi Factor Authentication or MFA adds an extra layer of security to the signing in process. It requires users to authenticate from an AWS supported MFA mechanism in addition to their sign in credentials when they access AWS services.

Multi Factor Authentication

AWS Multi Factor Authentication

The following are the AWS supported MFA mechanisms:

Virtual MFA Devices

This is a type of MFA where an application running on a phone or other device generates a six-digit numeric code.

U2F Security Key

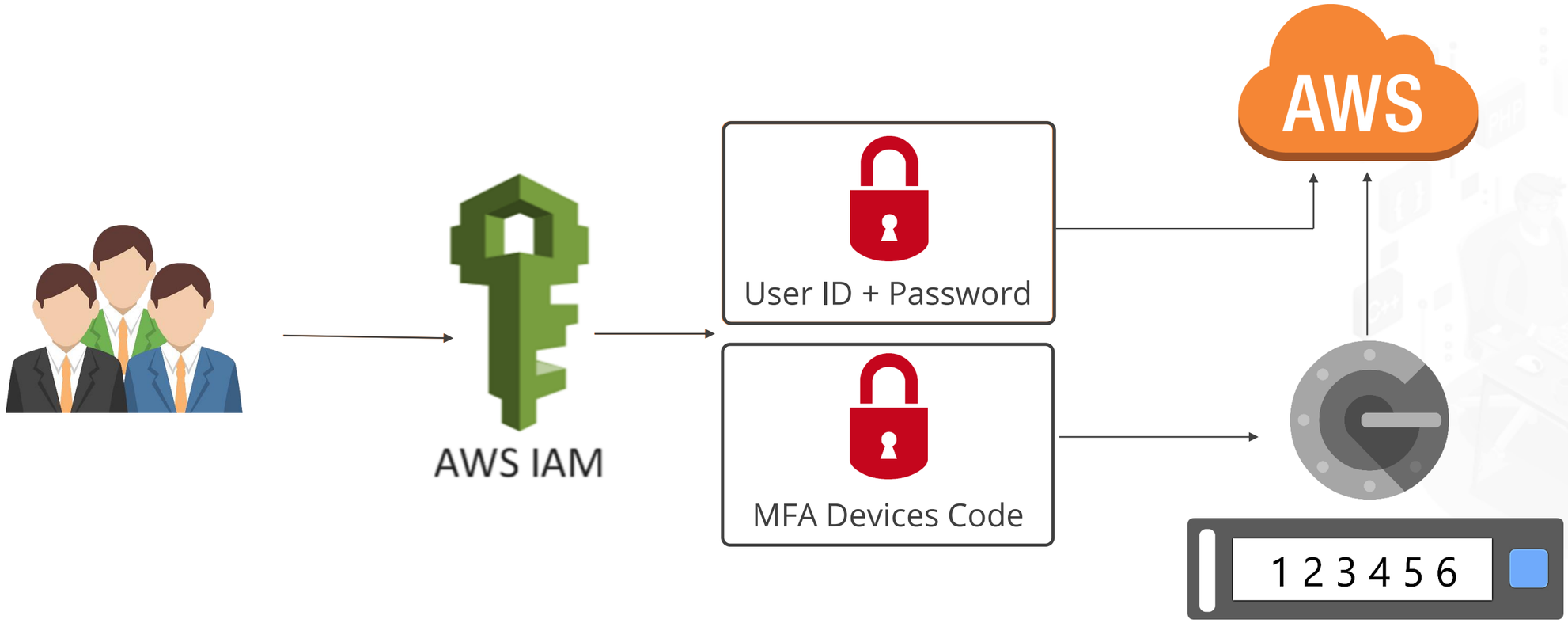
When a device is inserted into a USB port on the user's computer, the U2F security key is enabled as a form of MFA.

SMS Text Message-Based MFA

This form of MFA includes the user's SMS-compatible mobile phone number in the IAM user settings.

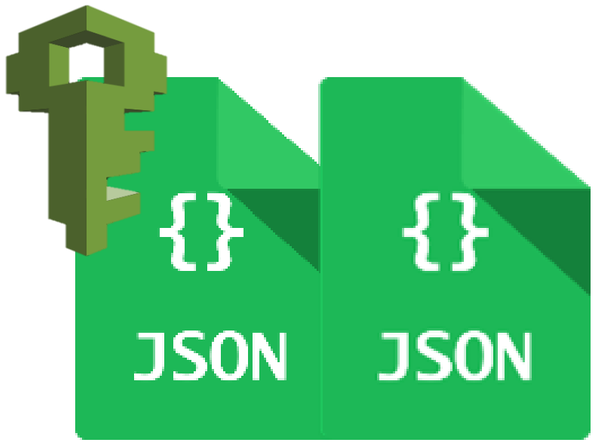
Multi Factor Authentication

AWS IAM supports Multi Factor Authentication (MFA) for users and resources to ensure absolute security by using MFA devices.



What Is IAM Policy?

An IAM policy is a document that defines one or more permissions. IAM policies can be attached to users, groups, roles, and AWS resources. They are written in JSON format.



AWS IAM Policies

More about IAM policies:

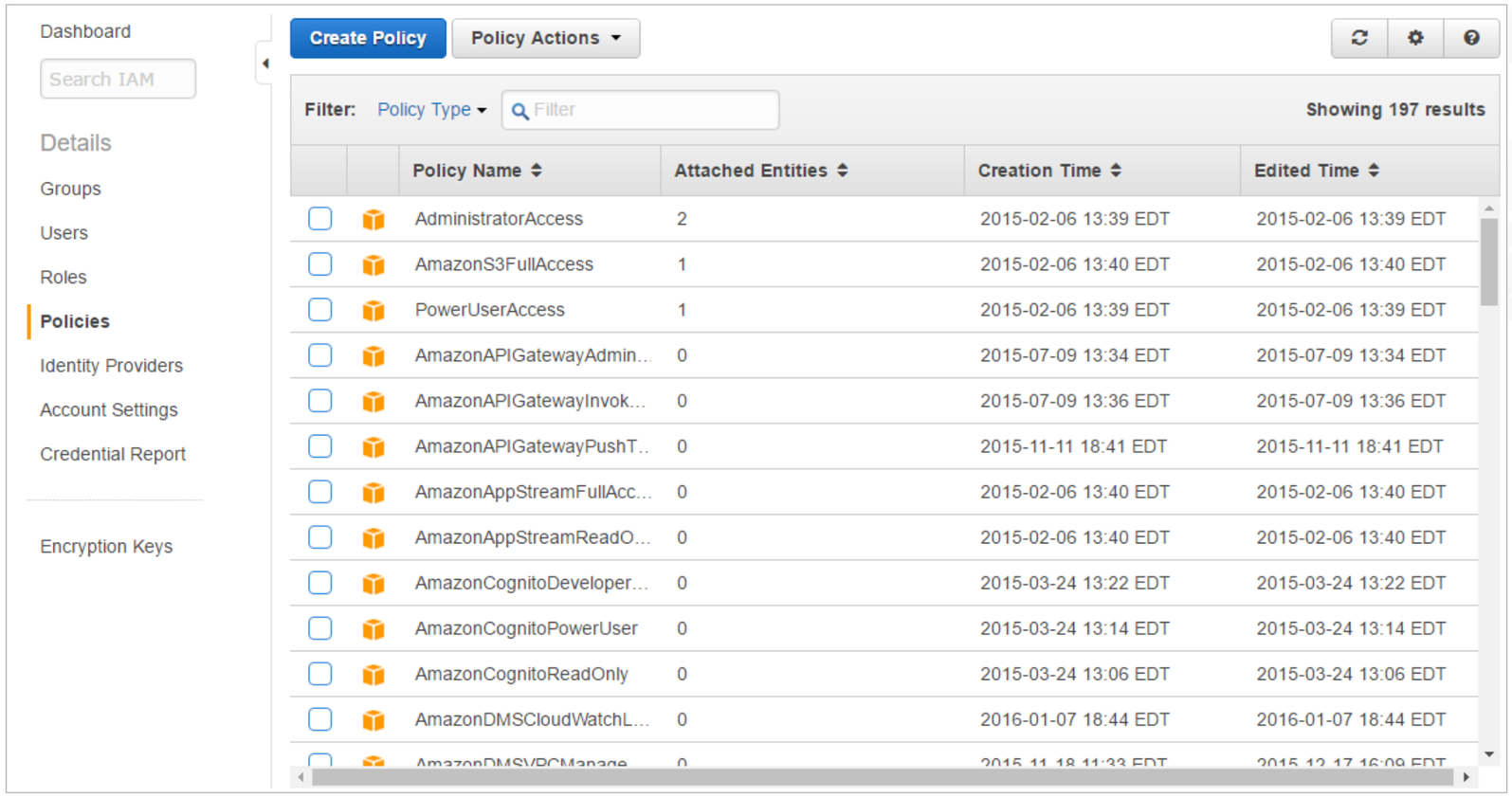
01 IAM Policies can be preselected from the AWS list of predefined policies.

02 Root users can edit the predefined policies to make customizations.

03 Root users can create and define a custom IAM policy from scratch.

What Is IAM Policy?

AWS has many predefined policies that allow users to define granular access to AWS resources. There are around 200 predefined policies available for users to choose from.



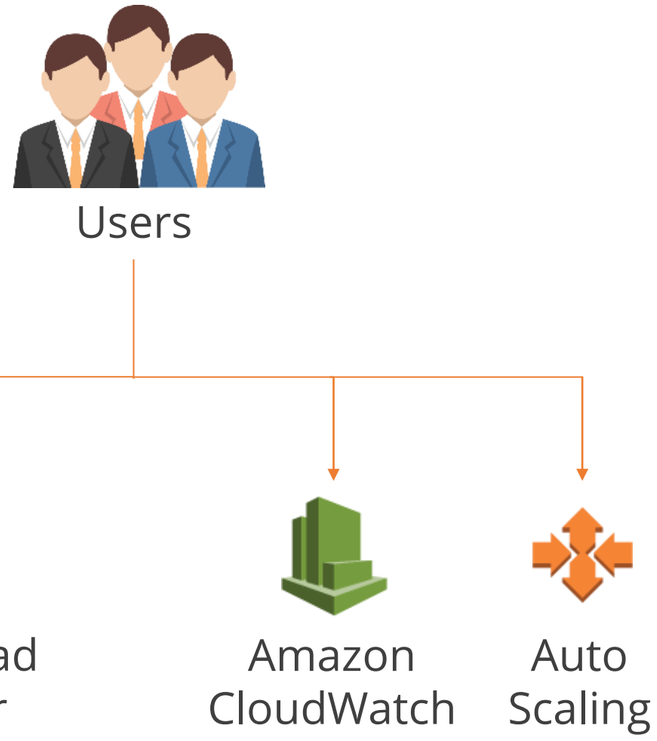
AdministratorAccess Policy

AdministratorAccess policy provides full access to AWS services and resources.

Admin User

AmazonEC2FullAccess Policy

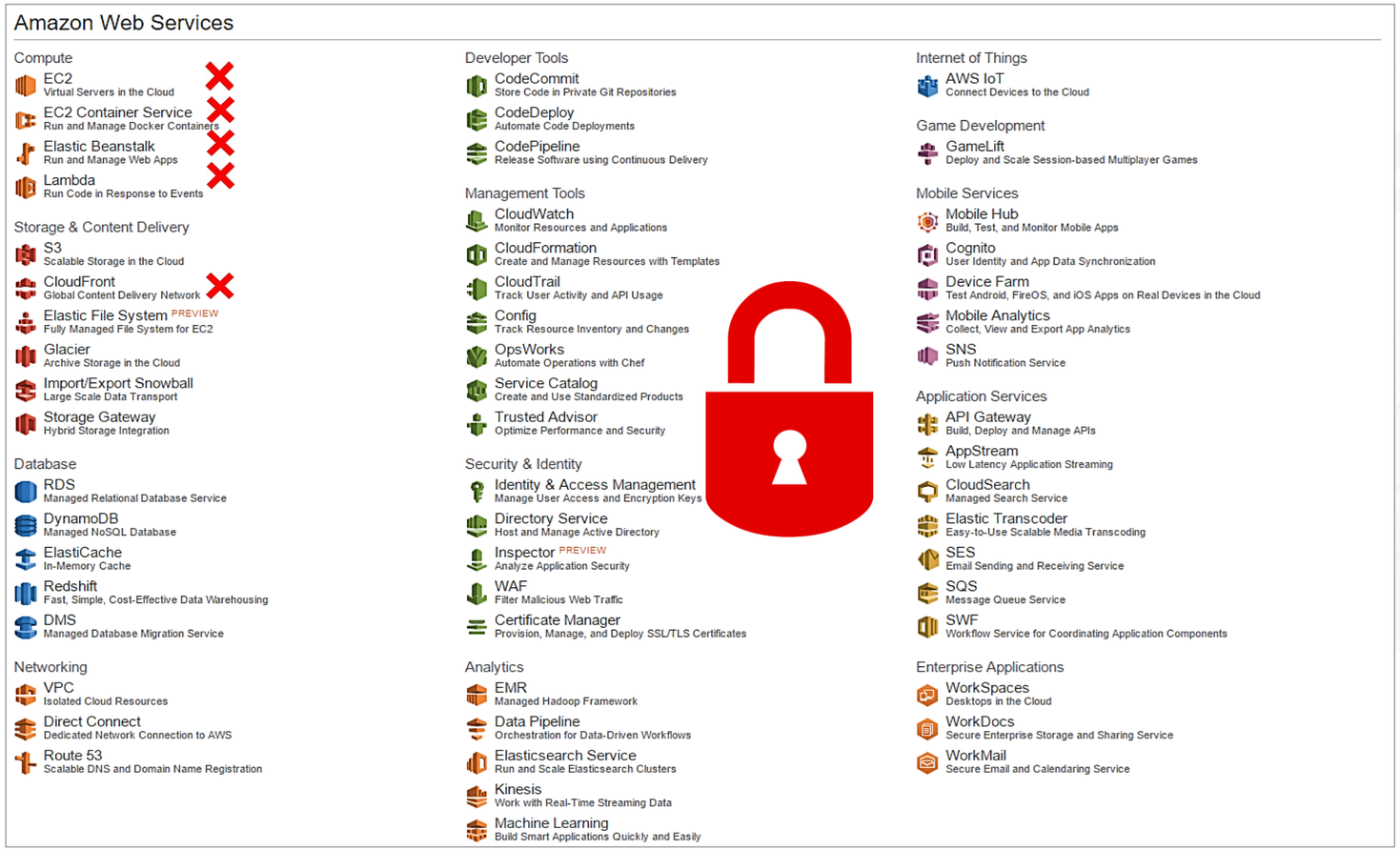
AmazonEC2FullAccess policy provides users or groups full access to the Amazon EC2 services and resources.



AmazonS3ReadOnlyAccess Policy

AmazonS3ReadOnlyAccess policy provides read-only access to all the buckets using AWS Management Console.

Users



Types of IAM Policies

There are two types of IAM policies:

Identity-based policies

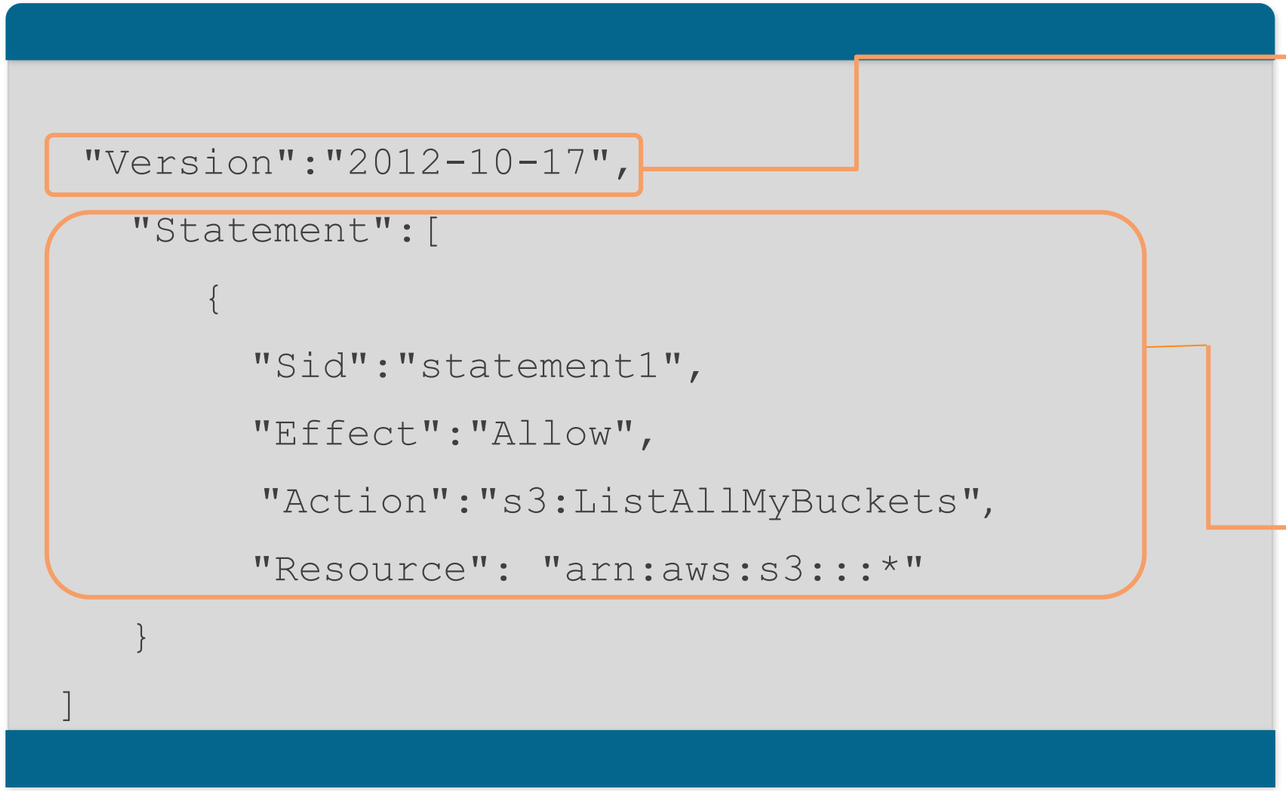
Identity-based policies can be attached directly to identities such as users, groups, and roles.

Resource-based policies

Resource-based policies are attached to AWS resources such as Amazon S3, Amazon EC2, and more.

Syntax of Writing AWS IAM Policies

AWS policies are written using JavaScript Object Notation (JSON).



Policy-wide information:

Version: Date when the policy was created

One or more individual statements:

Effect: Allows permission
Action: Lists all the S3 buckets
Resource: Name of the S3 bucket

Example of an Identity-Based Policy

Allowing a user to access Amazon EC2 from us-east-1:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEC2AccessUSEast1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances",
                "ec2:TerminateInstances"
            ],
```

Example of a Resource-based Policy

Allowing EC2 to access Amazon S3 from its public-IP:

```
"Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-1"
                }
            }
        }
    ]
}
```

Example of a Resource-based Policy

Allowing EC2 to access Amazon S3 from its public-IP:

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::YOUR_BUCKET_NAME",
        "arn:aws:s3:::YOUR_BUCKET_NAME/*"
      ],
```

Example of a Resource-based Policy

Allowing EC2 to access Amazon S3 from its public-IP:

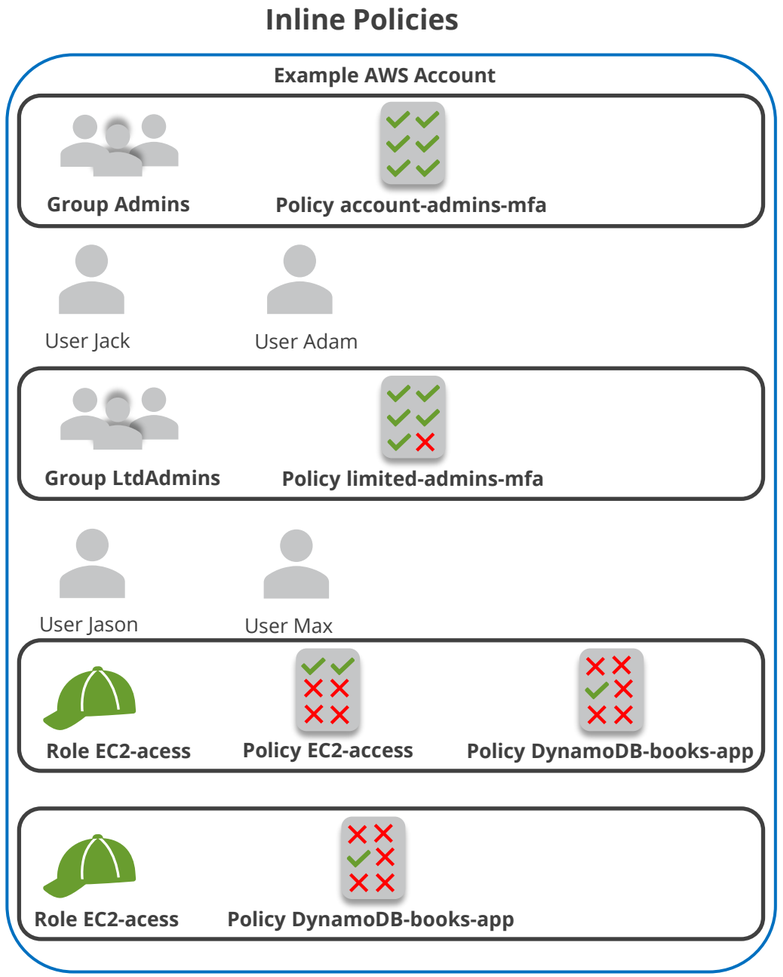
```
"Condition": {
        "IpAddress": {
          "aws:SourceIp": "YOUR_EC2_INSTANCE_PUBLIC_IP/32"
        }
      }
    }
  ]
}
```

Inline Policy

A policy that is included in an IAM identity is known as an inline policy (a user, group, or role).

Inline Policy

The following diagram explains inline policies:



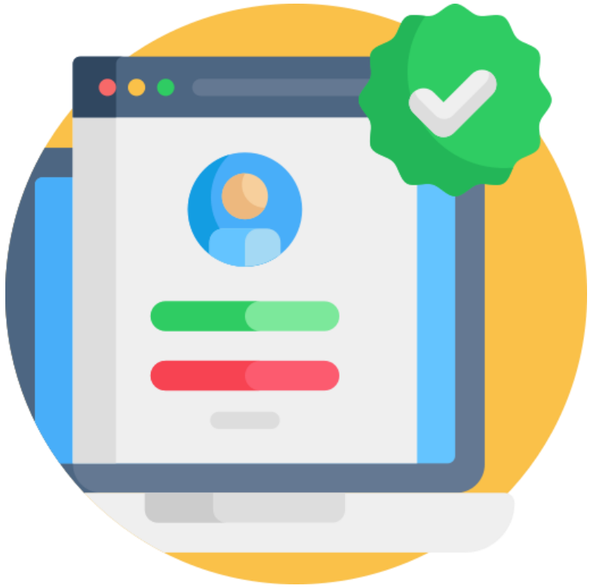
Managed Policy

AWS creates and manages AWS managed policy, which is a standalone policy.

Example:

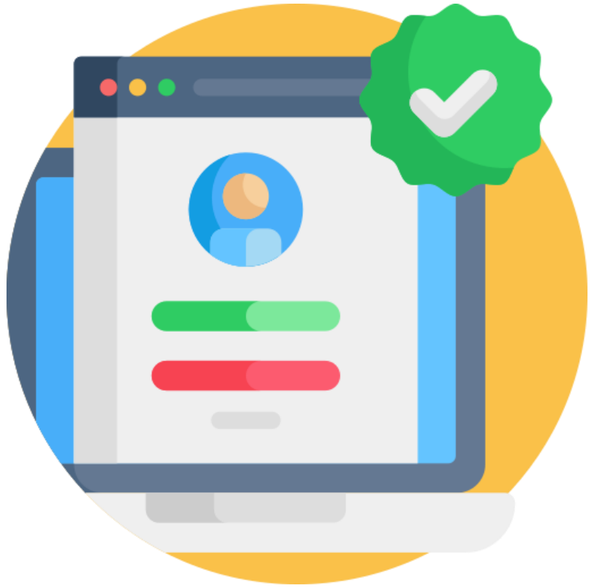
```
arn:aws:iam::aws:policy/IAMReadOnlyAccess
```

Managed Policy



Full access managed policies that provide complete access to a service include **AmazonDynamoDBFullAccess** and **IAMFullAccess**, which establish permissions for service administrators. An example of the same is common use cases.

Managed Policy



Assigning the proper permissions to individuals, groups, and roles is simpler with managed policies than it would be with user-written policies.

Amazon Resource Naming (ARN)

- AWS resources are identified uniquely by ARNs (Amazon Resource Names).
- When a resource needs to be specified clearly throughout all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls, the user needs to provide an ARN.

(ARN) Format

The standard formats for ARNs are as follows. The resource will specify the formats.

```
arn:partition:service:region:account-id:resource-id
arn:partition:service:region:account-id:resource-type/resource-id
arn:partition:service:region:account-id:resource-type:resource-id
```

AWS Service Principal

- A service's identification is called a service principal.
- Service roles are IAM roles that can be used by an AWS service.
- A trust policy is a requirement for service roles.

AWS Service Principal

An example of a policy that can be related to a service role is shown below.

```
"Principal": { "Service": [ "ecs.amazonaws.com",
"elasticloadbalancing.amazonaws.com" ] }
```

AWS Service Principal

The following example policy specifies permissions for the accounts 555555555555 or 123456789012:

```
"Principal" : {
       "AWS": [
                    "123456789012",
                     "555555555555"
              ]              }
```

AWS Account Principals

The principal element of a resource-based policy or condition keys that support principals are two places where users can specify AWS account identifiers.

```
"Principal": { "AWS":"arn:aws:iam::123456789012:root" }

      "Principal": { "AWS": "123456789012" }
```

IAM Role Principals

Identity resources in IAM are resources to which users can grant permission. Roles rely on an additional authenticated identity to act in that role.

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:role/role-name" }
```

Role Session Principals

A principal or identity that adopts a role is given temporary security credentials with the privileges of the assumed role.

Example

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:assumed-
            role/role-name/role-session-name" }
```

Web Identity Session Principal

Due to the web identity federation feature in AWS STS, users can develop applications where they may log in using a web-based identity provider like Login with Amazon, Facebook, or Google.

The web identity session principal is created when the AWS STS AssumeRoleWithWebIdentity action is used.

The user can log in using an external web identity provider (IdP) and then utilize this action to take on an IAM role.

Format of Web Identity Session Principal

Use the format shown below as the main component of a role trust policy:

```
"Principal": { "Federated": "cognito-identity.amazonaws.com" }

"Principal": { "Federated": "www.amazon.com" }

"Principal": { "Federated": "graph.facebook.com" }

"Principal": { "Federated": "accounts.google.com" }
```

SAML Session Principal

When the AWS STS AssumeRoleWithSAML operation is used, a SAML session principal is created.

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-
ID:saml-provider/provider-name" }
```

IAM User Principal

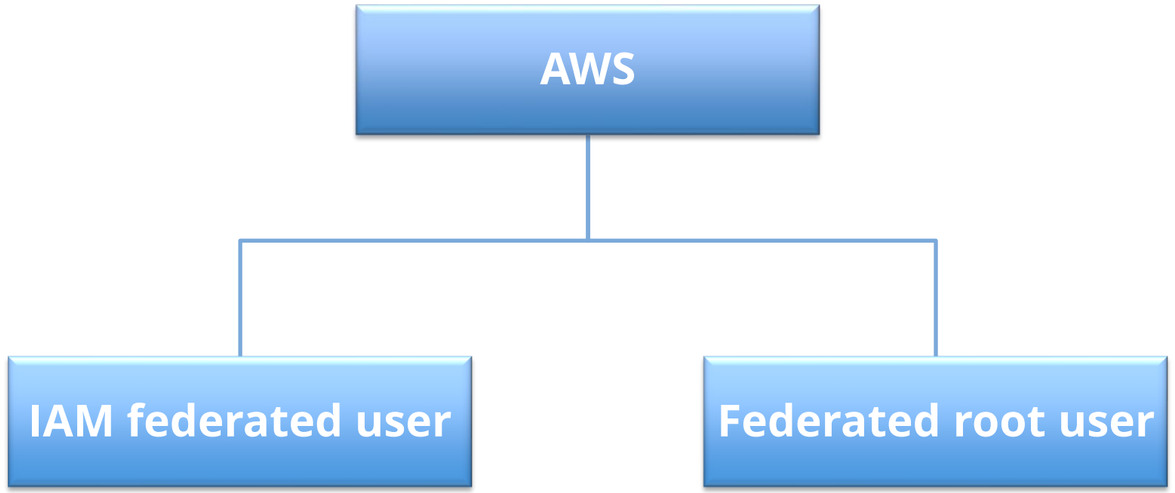
IAM users can be specified in condition keys that support principals or in the principal element of a resource-based policy.

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:user/user-name" }

"Principal": {
  "AWS": [
    "arn:aws:iam::AWS-account-ID:user/user-name-1",
    "arn:aws:iam::AWS-account-ID:user/user-name-2"
  ]
}
```

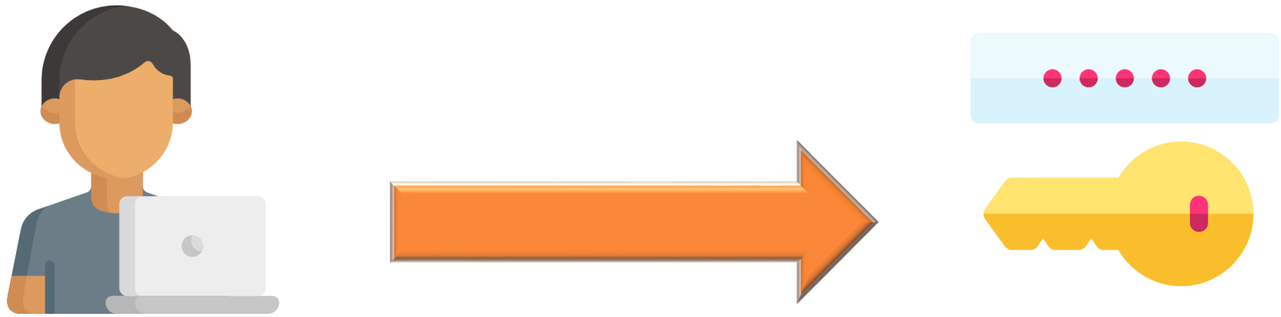
AWS STS Federated User Session Principal

A session principal created by using the AWS STS GetFederationToken procedure is known as an AWS STS federated user session principal.



AWS STS Federated User Session Principal

Use the following format in the Principal element to specify the federated user session ARN:



```
"Principal": { "AWS": "arn:aws:sts::AWS-account-
ID:federated-user/user-name" }
```

Principal

The following principles are used in AWS:

01	AWS account and root user
02	IAM roles
03	Role sessions
04	IAM users

Principal

The following principles are used in AWS:

05	Federated user sessions
06	AWS services
07	All principals

IAM JSON Policy Elements: Effect

The Effect element indicates whether the statement results in an allow or an explicit deny.

The Effect has two possible values: Allow and Deny.

```
"Effect":"Allow"
```

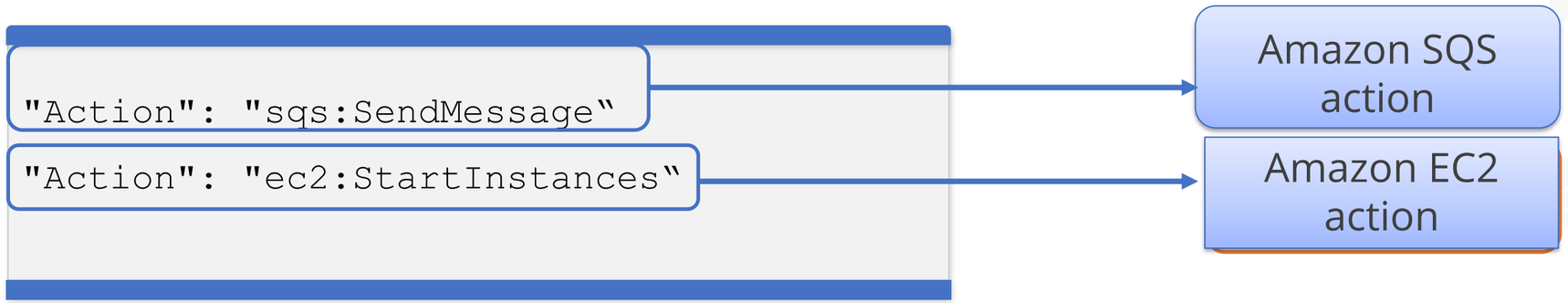
IAM JSON Policy Elements: Action

The Action element specifies which specific actions will be **allowed** or **denied**.

Every AWS service has a unique set of actions that outlines the tasks that the user can complete with that service.

IAM JSON Policy Elements: Action

Some examples of various services' Action elements are:



Resources

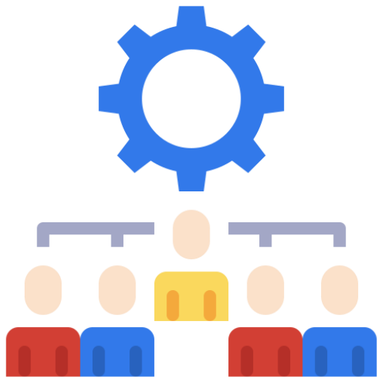
A resource in AWS is an entity with which a user can interact.

The following is the format for resource type identifiers:

```
service-provider::service-name::data-type-name
```

Resource

Amazon Simple Storage Service resource reference AWS::S3::AccessPoint:



The AWS::S3::AccessPoint resource is a sort of Amazon S3 resource that allows users to access buckets.

AWS::S3::AccessPoint

Use the following syntax to declare this entity in the user's AWS CloudFormation template JSON:

```
{
 "Type" : "AWS::S3::AccessPoint",
 "Properties" : { "Bucket" : String,
 "Name" : String,
 "Policy" : Json,
 "PolicyStatus" : Json,
 "PublicAccessBlockConfiguration" : PublicAccessBlockConfiguration,
 "VpcConfiguration" : VpcConfiguration
 }
}
```

AWS::S3::AccessPoint

Use the following syntax to declare this entity in the user's AWS CloudFormation template YAML:

```
Type: AWS::S3::AccessPoint
Properties:
 Bucket: String
 Name: String
 Policy: Json
 PolicyStatus: Json
 PublicAccessBlockConfiguration:
 PublicAccessBlockConfiguration
 VpcConfiguration:
 VpcConfiguration
```

Not Clause

The NOT rule statement logically negates the outcomes of a single nested statement, the nested statements must not match for the NOT statement to match, and vice versa.

Conditional Access

- Users can provide conditions for when a policy is in effect using the condition element (or condition block).
- The condition element is optional.

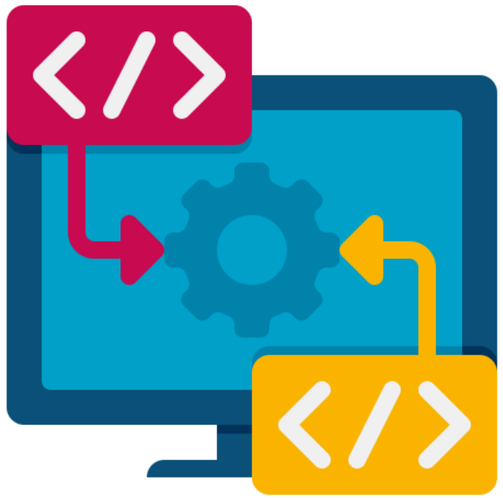
```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

Conditional Access

The following condition matches users with the names Anna or Ana by using the StringEqualsIgnoreCase operator.

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "johndoe" }}
```

Implicit Deny



- By default, each request is denied. This is called an implicit deny.
- An implicit denial occurs when there is no applicable deny statement and no applicable allow statement.

Explicit Allow and Deny

A request is explicitly denied if the applicable policy contains a deny statement.

Explicit Allow and Deny

The AWS service decides whether to approve or reject a request after it is sent. **Implicit deny** is the default setting, where all requests are rejected. **Explicit allow** is a type of allowing that can overwrite this option.

Explicit Allow and Deny

The following policy, which includes allowed actions, acts that are implicitly denied, and actions that are expressly denied, can be created.

```
{
"Version": "2012-10-17",
"Statement": [
        {
                "Sid": "AllowGetList",
                "Effect": "Allow",
                "Action": [
                                "iam:Get*", "iam:List*"
                                ],
                "Resource": "*"
```

Explicit Allow and Deny

The following policy, which includes allowed actions, acts that are implicitly denied, and actions that are expressly denied, can be created.

```
},
 {
                "Sid": "DenyReports",
                "Effect": "Deny",
                "Action": "iam:*Report",
                "Resource": "*"
                }
]
}
```

Permission Boundary

An advanced feature of managed policies is the ability to set the maximum number of permissions that an identity-based policy can provide to an IAM object.

Permission Boundary

Assume that only Amazon S3, Amazon CloudWatch, and Amazon EC2 should be accessible to the IAM user anna. The permissions limit for the user can be configured using the following policy to enforce this rule:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
```

Permission Boundary

Assume that only Amazon S3, Amazon CloudWatch, and Amazon EC2 should be accessible to the IAM user anna. The permissions limit for the user can be configured using the following policy to enforce this rule:

```
"cloudwatch:*",
                    "ec2:*"
            ],
            "Resource": "*"
            }
        ]
    }
```

IAM Roles

IAM Roles are permissions and policies that determine the access available to the AWS identities.

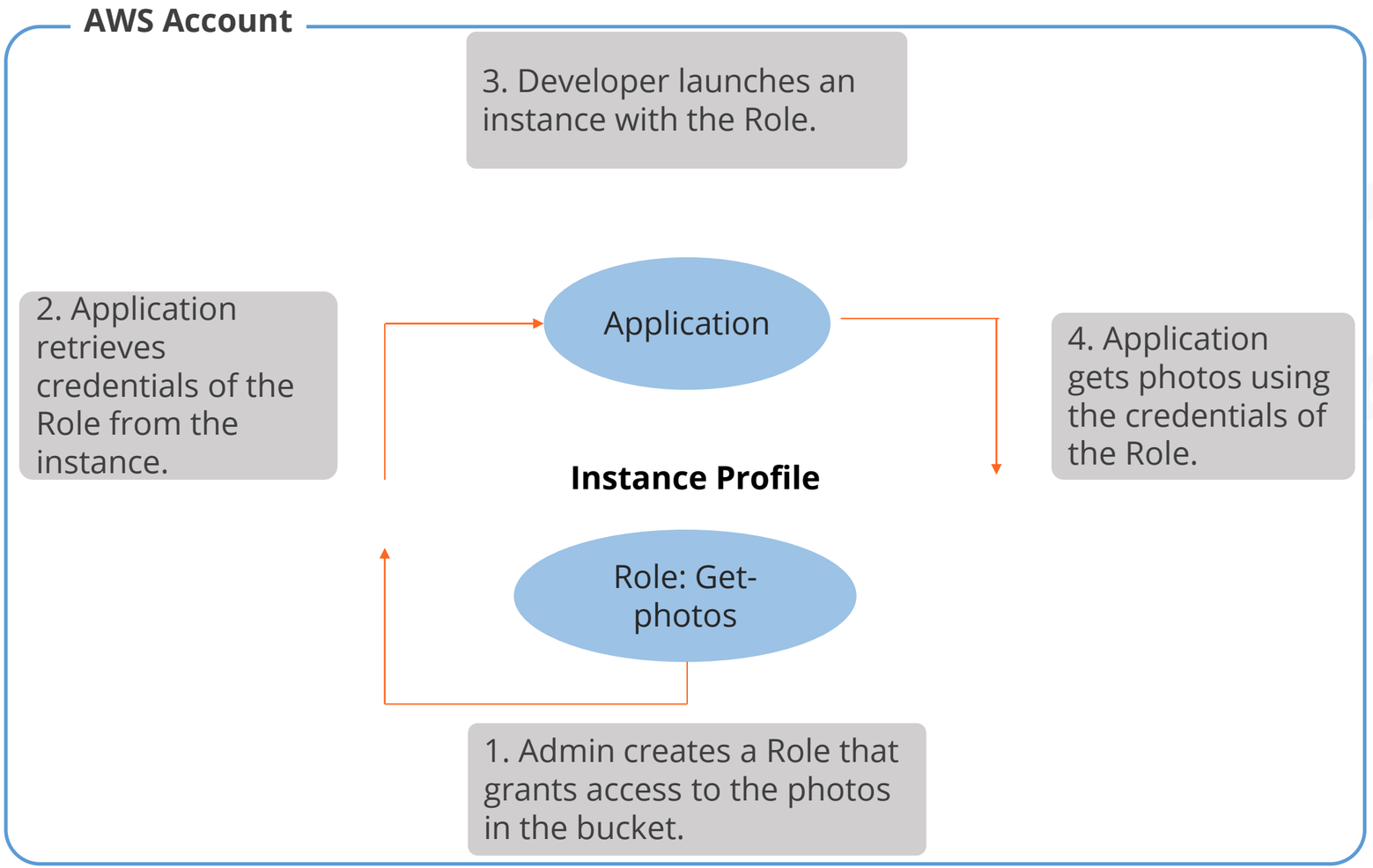
IAM roles can be further discussed as follows:

IAM Roles

01	IAM Roles function in a way similar to that of IAM users.
02	They are not password protected and do not require access keys.
03	These roles can be used by anyone who requires them.

Various Functions of IAM Roles

IAM Roles provide access to users, applications, and services that do not have permission to use AWS resources.



Cross-Account Policy

- A principal with access to one account that can also use the resources in another is called cross-Account access.
- When users enable cross-account access, the principal's account is referred to as the trusted account.
- The resource's account performs as the trusting account.

Cross-Account Policy

The following example shows the scenario where a user in one account receives permissions from a resource-based policy in a different account:

```
{
        "Version": "2012-10-17",
       "Statement": [ {
       "Sid": "AllowS3ListRead",
       "Effect": "Allow",
       "Action": "s3:ListAllMyBuckets",
       "Resource": "*" },
       {
       "Sid": "AllowS3ProductionObjectActions",
       "Effect": "Allow",
```

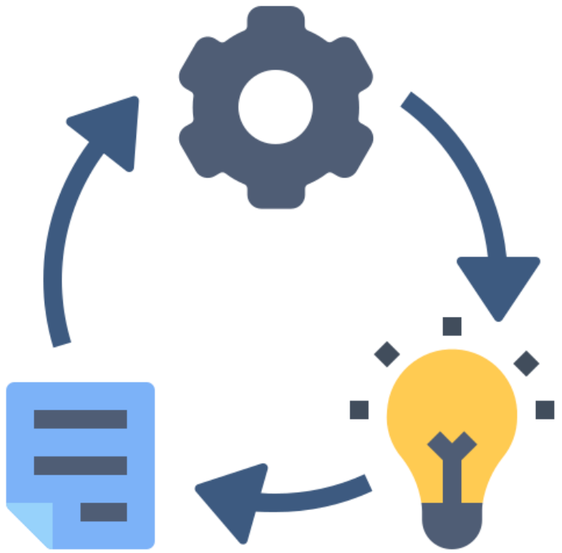
Cross-Account Policy

```
"Action": "s3:*Object*",
      "Resource": "arn:aws:s3:::Production/*" },
{
      "Sid": "DenyS3Logs",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": [ "arn:aws:s3:::*log*",
                    "arn:aws:s3:::*log*/*" ]
}
]
}
```

External ID

- The External ID is a piece of information that can be supplied to the Security Token Service's AssumeRole API (STS).
- When a certain value is present in the external ID, the user can utilize the external ID in the condition element of a role's trust policy to restrict when the role can be assumed.

What Is the Purpose of the External ID?



The main goal of the external ID is to solve the issue of the confused deputy problem.

The external ID enables the person who is assuming the role to declare their operating environment.

It gives the account owner a way to restrict the role's assumption to certain situations.

Example of External ID

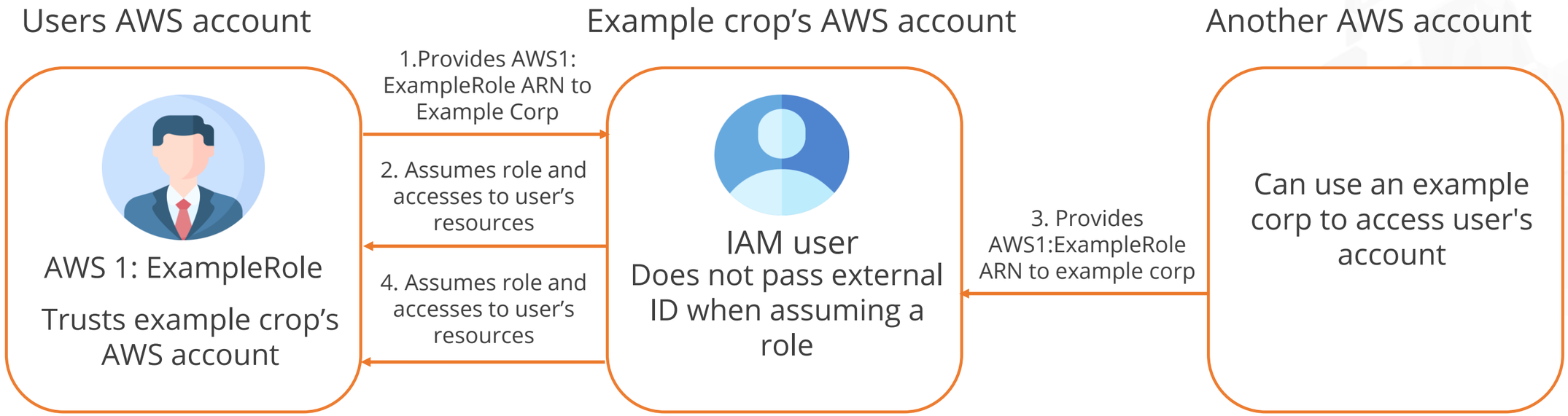
This is a scenario where a partner needs access to specific resources to carry out AWS operations on their behalf. The partner requires a mechanism to access the AWS resources belonging to all of its multiple customers.

Customers

AWS resources

The Confused Deputy

In this example, the confused deputy problem is utilized to show how an unauthorized person could access users' AWS services. Given this situation:

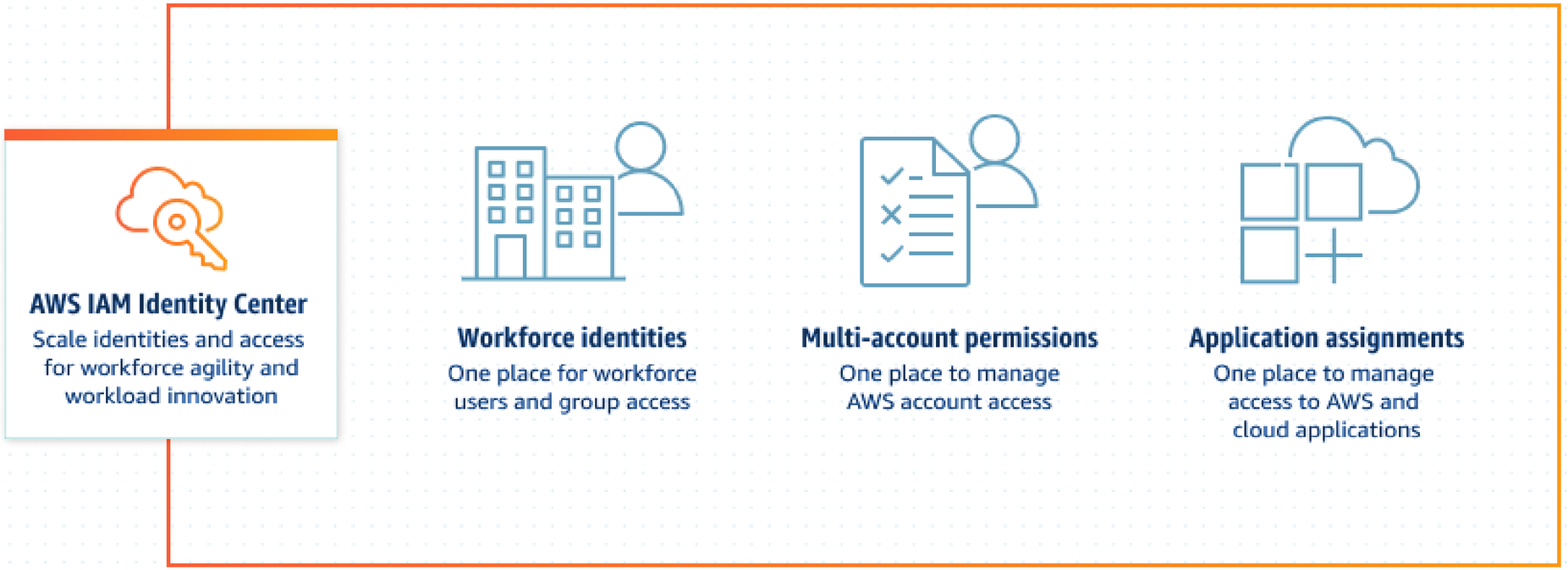


AWS Cognito

SSO

The cloud-based single sign-on (SSO) solution, AWS Single Sign-On, makes it simple to manage SSO access to all user's AWS accounts and cloud applications from a single location.

Users can use AWS Organizations to manage SSO access and user rights for all their AWS accounts.



SAML

- An identity provider (IdP) can authenticate users and send identification and security information about them to a service provider (SP).
- Generally, an application or service uses the open federation standard Security Assertion Markup Language 2.0 (SAML).

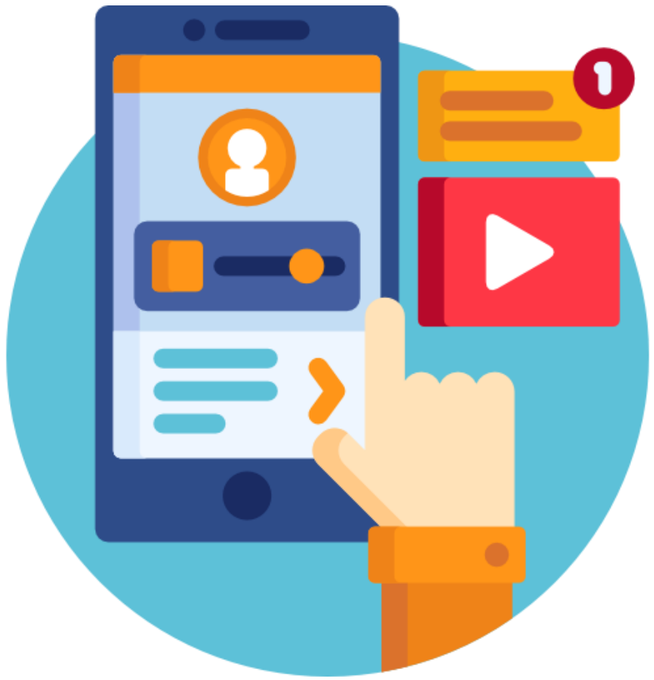
Active Directory

Users' directory-aware workloads and AWS resources can use managed Active Directory (AD) in AWS with the help of AWS Directory Service for Microsoft Active Directory, commonly known as AWS Managed Microsoft Active Directory (AD).

AWS Organizations

- As users' AWS resources expand and scale, AWS organizations enable users to manage and control their environment centrally.
- Users can group accounts to organize their workflows, apply policies to accounts or groups for governance, and organize billing by using a single payment method for all their accounts using AWS organizations.

Cognito



Amazon Cognito handles web and mobile app authentication, authorization, and user management.

Users can sign in directly using a username and password or via a third-party service such as Facebook, Amazon, Google, or Apple.

User pools and identity pools are the two fundamental components of Amazon Cognito.

Cognito

This example is a combination of an Amazon Cognito user pool and an identity pool.

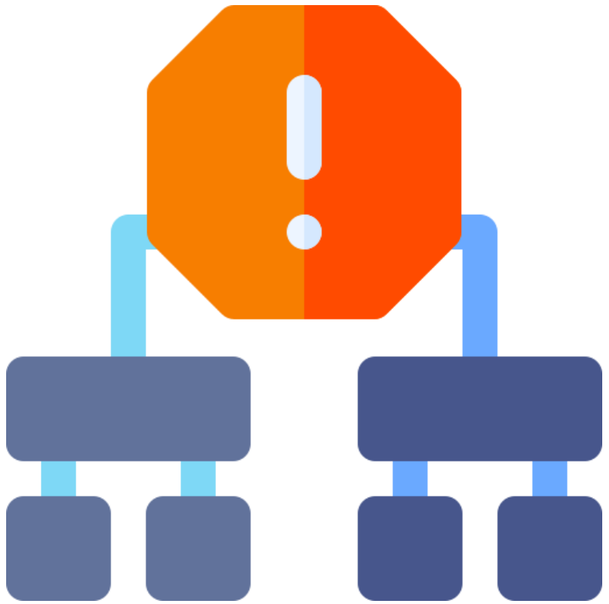
In the first step, the app user registers via a user pool and receives user pool tokens after successful authentication.

Following that, the user app uses an identity pool to exchange user pool tokens for AWS credentials.

Finally, app users can use their AWS credentials to access other AWS services like Amazon S3 and DynamoDB.

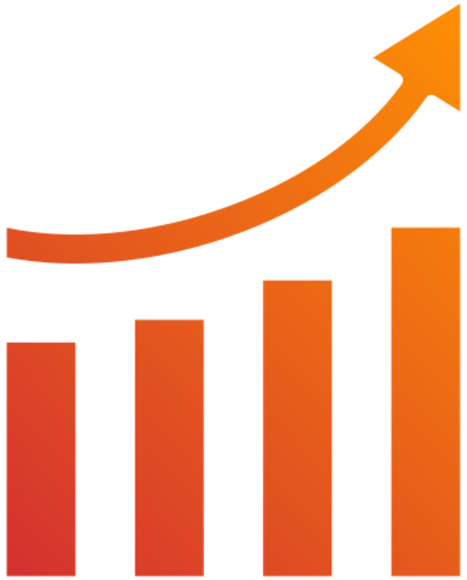
Cloud Tower

AWS Control Tower



- AWS Control Tower is the simplest method to set up and manage a secure, multi-account AWS environment known as a landing zone.
- Provides user’s landing zone with AWS Organizations, providing continuous account management and governance as well as best practices for deployment based on AWS's expertise in working with hundreds of clients as they migrate to the cloud.

Benefits



- **Set up and create a new AWS environment quickly.**
- Automate continuous policy management.
- View policy-level summaries of users' AWS environment.

Cloud Security

Users don't need to handle actual servers or storage components when using the cloud. As an alternative, users employ software-based security measures to keep an eye on and secure the information flowing into and out of their cloud resources.

Identity-based policy, Implicit Deny, Explicit Allow

Duration: 13 mins

Problem Statement:

You have been assigned a task to implement identity-based policies, implicit deny, and explicit allow.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Configuring identity-based policy
2. Performing implicit denial on the bucket
3. Performing explicit allow

Policy Generator, Managed Policy, Versions, Groups

Duration: 8 mins

Problem Statement:

You have been assigned a task to perform policy generator, managed policy, versions, and groups.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Creating and managing policy
2. Using group users can attach policy and permissions directly to the group
3. Creating and managing S3 versioning

Resource-based policy, Policy Generator, Principals

Duration: 8 mins

Problem Statement:

You have been assigned a task to create a resource-based policy using a policy generator and principals.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Creating a resource-based policy
2. Policy generator using principals

Conditional Variables Restricted Access By IP

Duration: 8 mins

Problem Statement:

You have been assigned a task to allow or block conditional variables restricted access by IP.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Creating a security group
2. Changing the security group

Restrict Access Using VPC Endpoint

Duration: 8 mins

Problem Statement:

You have been assigned a task to perform restricted access using the VPC endpoint.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Create a custom VPC and enable DNS hostname
2. Create an Internet Gateway
3. Create 2 subnets
4. Create a Route table and attach them to subnets
5. Launch 3 instances
6. Launch public and private VM
7. Install DB on a private DB server
8. Create a bucket and upload a document
9. Create an Endpoint

Cross-account Management Console access using IAM Roles

Duration: 8 mins

Problem Statement:

You have been assigned a task to create and configure cross-account management console access using IAM roles.

ASSISTED PRACTICE

Assisted Practice: Guidelines

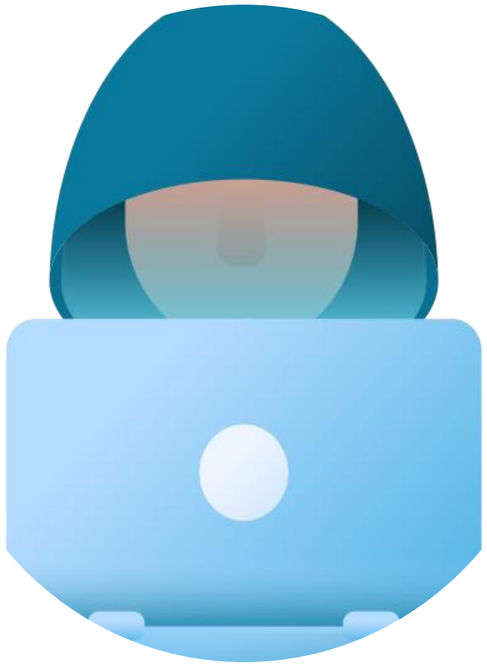
Steps to be followed:

1. Creating a role in the production account
2. Granting access to the role
3. Testing the access by switching roles

Cloud Security

Types of Cyber Attacks

Following are some common threats:



- Phishing
- Password management
- Credential leaks
- Network security
- Insider threat
- Security incident recovery planning

Case Study: Netflix

Location: United States

Industry: On-demand media industry

Challenge

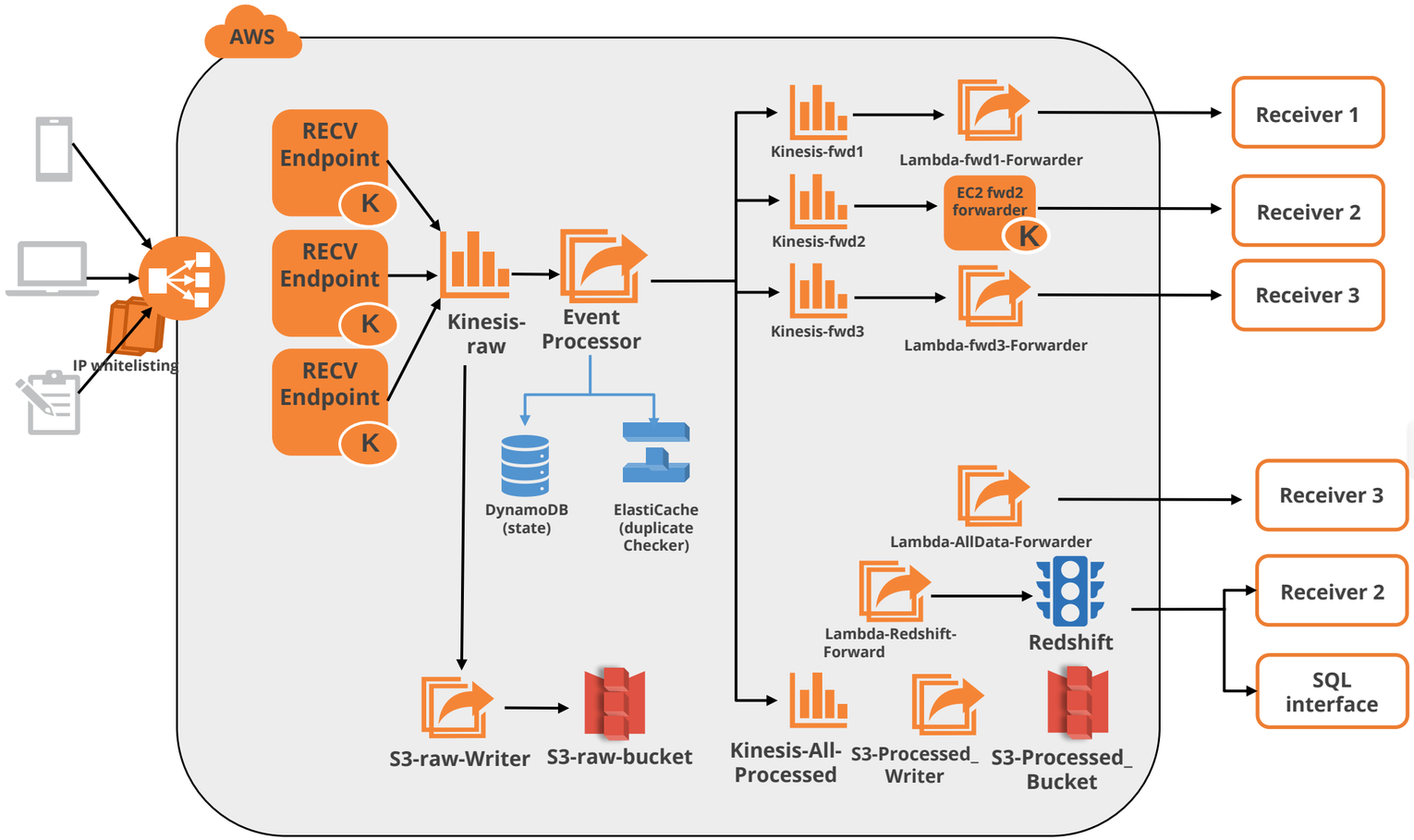
In 2008, Netflix was highly focused on DVD-by-mail services. Due to the above-mentioned database corruption problem, DVD shipping was halted for three days.

Solution

Netflix management decided to shift away from relational systems in their data centers and toward the cloud. The cloud was AWS (Amazon Web Services), which allowed the organization to scale as much as it required.

Architecture Diagram

Following is a F-secure Data Pipeline:



Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides the most comprehensive and deep compute platform, with over 500 instances and a selection of the most recent processor, storage, networking, and operating system.

Windows / Linux
Amazon EC2 instance

Use Cases:

- Run cloud-native and enterprise applications
- Scale for HPC applications
- Develop for Apple platforms
- Train and deploy ML applications

Amazon Kinesis

Amazon Kinesis simplifies the collection, processing, and analysis of real-time, streaming data, allowing the user to gain timely insights and respond quickly to new information.

NETFLIX

Use Cases:

- Build video analytics applications
- Evolve from batch to real-time analytics
- Build real-time applications
- Analyze IoT device data

Amazon DynamoDB

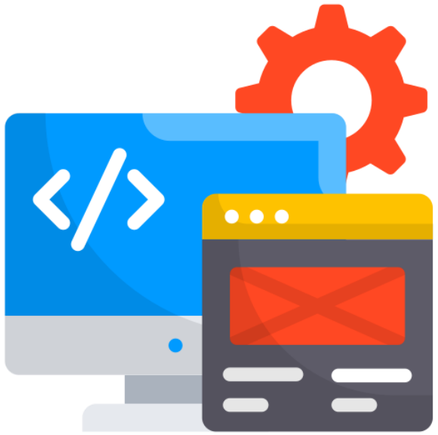
A serverless, fully managed, key-value NoSQL database called Amazon DynamoDB is made to support high-performance software at any scale.

Use Cases:

- Develop software applications
- Create media metadata stores
- Deliver seamless retail experiences
- Scale gaming platforms

Amazon RDS

A group of managed services known as an Amazon Relational Database Service (Amazon RDS) makes it simple to set up, operate, and scale databases in the cloud.



Use Cases:

- Build web and mobile applications
- Move to managed databases
- Break free from legacy databases

Amazon Lambda

Users can run code for practically any kind of application or backend service using the serverless, event-driven compute service provided by AWS Lambda, which eliminates the need for installing or managing servers.

Use Cases:

- Process data at scale
- Run interactive web and mobile backends
- Enable powerful ML insights

Amazon S3

A leading provider of scalability, data availability, security, and performance in the object storage space is Amazon Simple Storage Service (Amazon S3).

Use Cases:

- Build a data lake
- Back up and restore critical data
- Archive data at the lowest cost

Amazon VPC

Amazon VPC allows users to have complete control over their virtual networking environment, including resource placement, connection, and security.

Use Cases:

- Launch a simple website or blog
- Host multi-tier web applications
- Create hybrid connections

Amazon CloudFront

Amazon CloudFront is a CDN solution designed for great performance, security, and developer simplicity.

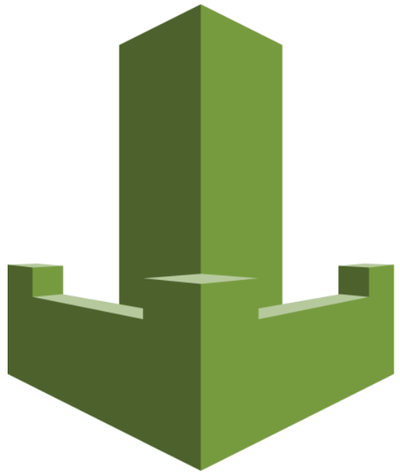
Use Cases:

- Deliver fast, secure websites
- Accelerate dynamic content delivery and APIs
- Stream live and on-demand video

WAF

What Is AWS WAF?

AWS WAF is a web application firewall that lets users monitor the HTTP and HTTPS requests that are forwarded to an Amazon CloudFront distribution, an Amazon API Gateway REST API, or an Application Load Balancer.



AWS WAF

AWS WAF

AWS WAF enables users to:

01	Allow all the requests except the ones that they specify
02	Block all the requests except the ones that they specify
03	Count the requests that match the properties that they specify

How to Handle False Positives in WAF?

Users have to use a web browser. Then, to check for a false positive of **style==xxx** on their **example.com** domain, they enter **example.com/style==xxx** in the web browser. The response is error code **403 Forbidden**.

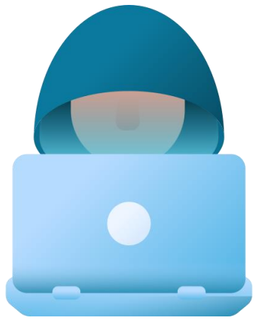
```
$ curl -ikv http://example.com/[false positive]
```

Note

Users apply curl and replace **users' false positive** for **false positive**.

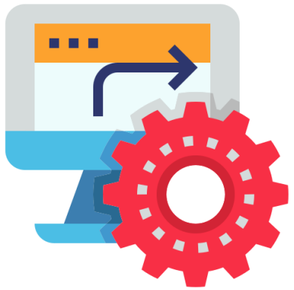
Benefits of AWS WAF

The following are the benefits of AWS WAF:



WAF offers additional protection against web attacks.

It provides real-time information and samples of web requests.



It offers automated administration.

AWS WAF

Duration: 8 mins

Problem Statement:

You have been assigned a task to create and configure AWS WAF.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Creating and configuring IP sets
2. Creating and configuring web ACL
3. Creating a Custom rule in web ACLs

Shield

What Is AWS Shield?

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency.

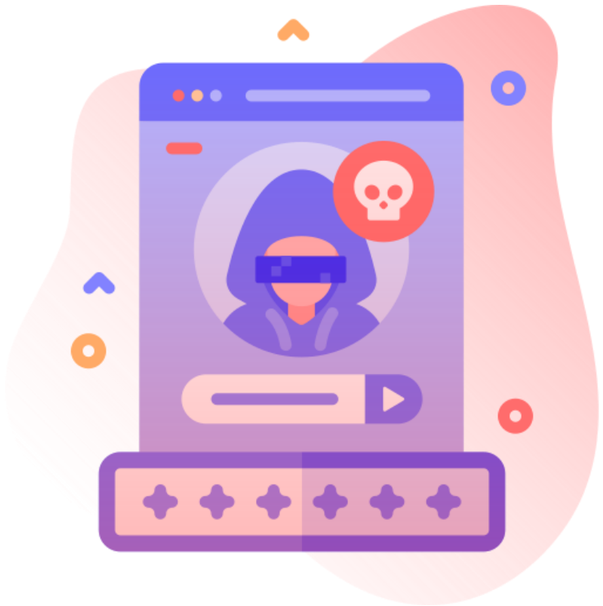
AWS Shield

Network and Application Protection

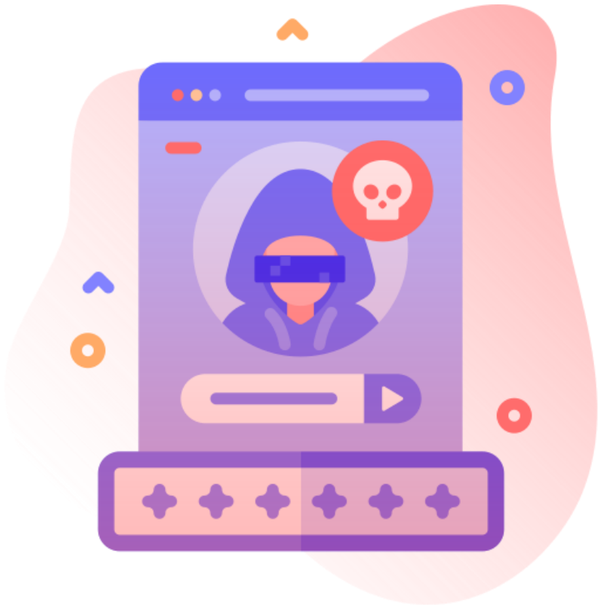
- AWS offers services to the user's network and application security teams that address their specific protection needs and regulatory requirements.
- Amazon VPC security groups secure resources in user’s AWS workloads at the host level.

AWS Shield Advance

The advanced subscription service adds DDoS mitigation capabilities, intelligent attack detection, and attack mitigation at the application (AWS WAF included) and network layers.



AWS Shield Advance



- Application-layer Attacks (layer 7):
  Consists of requests like HTTP Gets designed to consume application resources.

- Other Application-layer Attacks:
  The OWASP Top 10 publication includes SQL injection (SQLi), cross-site scripting (XSS), remote file inclusion (RFI), and other online application attacks and threats.

Benefits of AWS Shield

The following are some of the benefits of AWS Shield:

01 Seamless integration and deployment

02 Customizable protection

03 Managed protection and attack visibility

04 Cost-efficiency

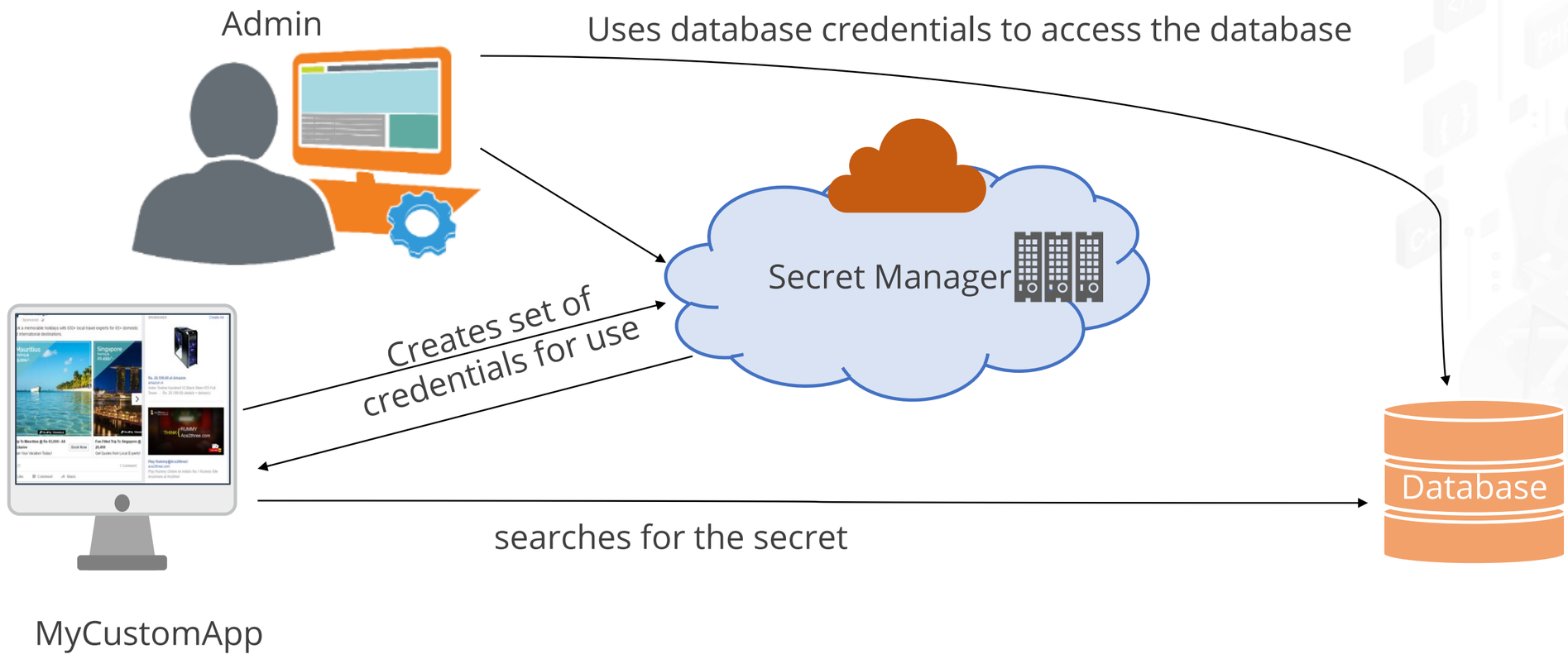
Secrets Manager

AWS Secrets Manager

Secrets Manager allows users to replace hardcoded credentials, such as passwords, in the code with an API call to Secrets Manager to retrieve the secret programmatically.

AWS Secrets Manager Scenario

The diagram below depicts how the users can save database credentials in Secrets Manager and then use those credentials in an application to access the database.



AWS Secrets Manager Features

The features of AWS Secret Manager can be listed below:

01 It allows the users to replace stored credentials with a runtime call to the Secrets Manager Web service dynamically.

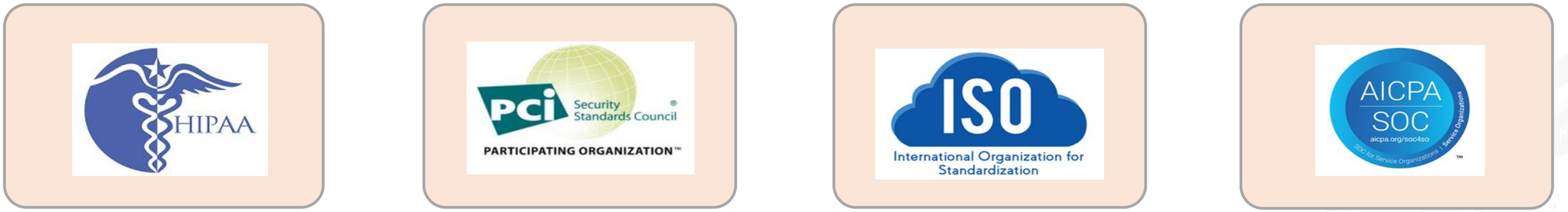
02 It is easy to create, leaving users time to focus on creating the applications.

03 It always returns the most recently encrypted secret value version.

04 It provides users with the scalability and reliability provided by AWS.

Standards for AWS Secrets Manager

AWS Secrets Manager has a few standards which can be listed below:



A list of the cost of each activity in a process is totaled to understand the cost associated with a product or service.

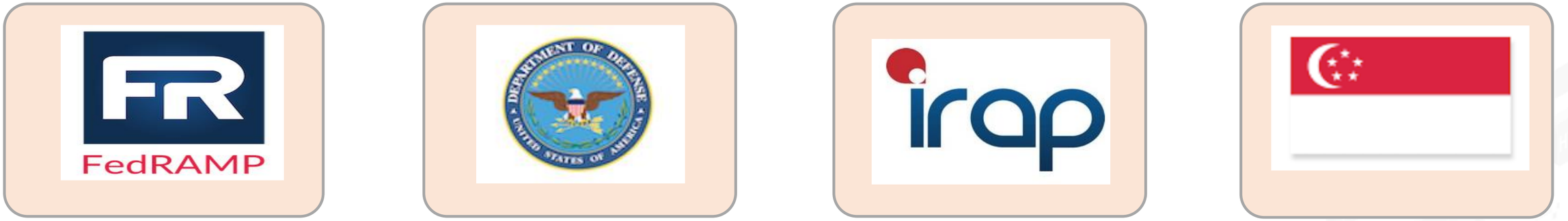
A set of diagrams used for aligning process improvement efforts to customer requirements.

An analysis of time taken by each activity within the process.

Used to develop new processes or improve existing processes.

Standards for AWS Secrets Manager

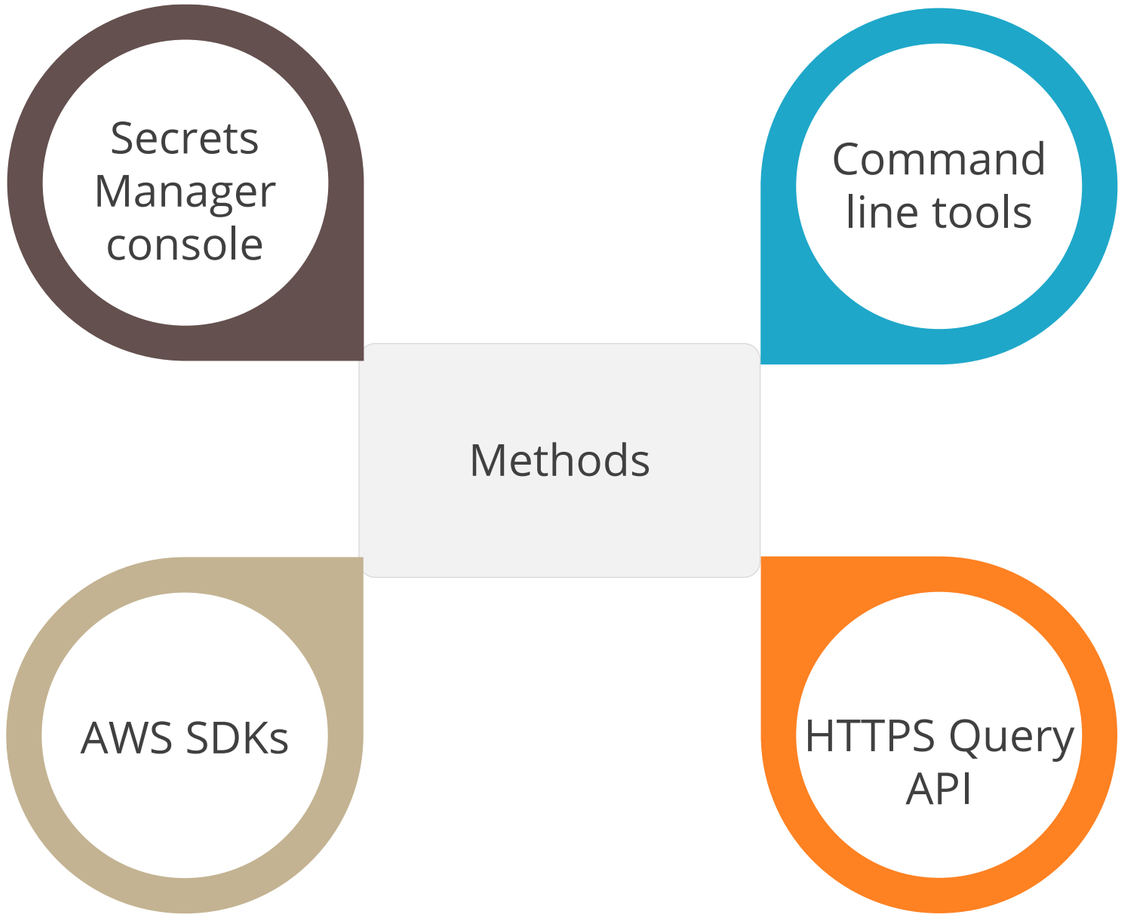
AWS Secrets Manager has a few standards which can be listed below:



- It provides an approach to security assessment and continuous monitoring for cloud products.
- It validates that appropriate controls are in place and determines the appropriate responsibility model.
- It provides process improvement effort to customer requirements.
- It demonstrates to customers AWS's commitment to meeting the high expectations.

Working with Access Secrets Manager

Any of the following methods can be used to interact with Secrets Manager:



Working with Access Secrets Manager

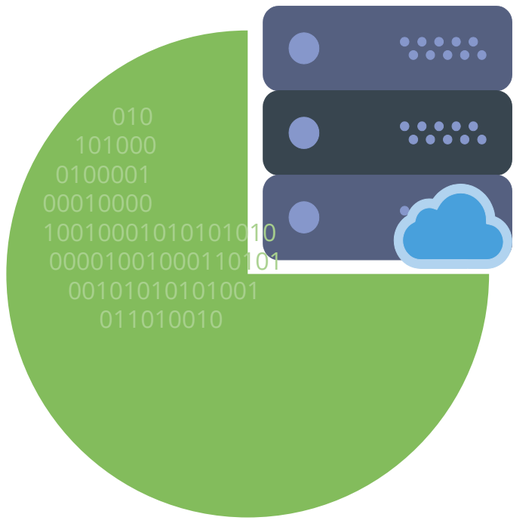
Secrets Manager console

The browser-based Secrets Manager console allows users to manage the secrets and perform almost any task related to the secrets.

Working with Access Secrets Manager

Command line tools

The AWS command line tools enable users to perform Secrets Manager and other AWS tasks by issuing commands from the system command line.



Working with Access Secrets Manager

AWS SDKs

The AWS SDKs include libraries and sample code for a variety of programming languages and platforms, including Java, Python, Ruby, .NET, and others.

Working with Access Secrets Manager

HTTPS Query API

The HTTPS Query API allows users to access Secrets Manager and AWS programmatically.

API

Secrets Manager Administrator Permissions

End users should not be granted administrator permissions that helps them to carry out the following tasks:

- Focusing on how an organization performs work
- Viewing the value delivery of an organization

Permissions to Access Secrets

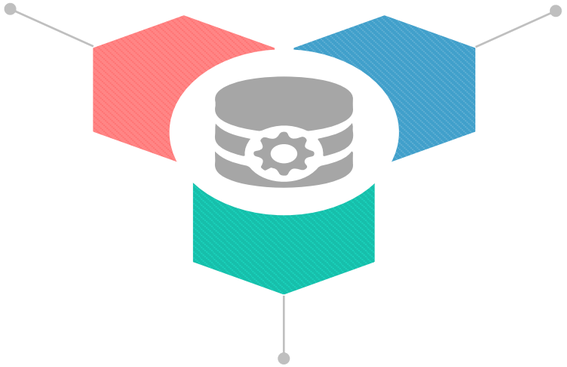
Users may manage which people or services have access to the secrets by using IAM authorization policies. A permissions policy specifies who can do what on which resources. Users may:

- Attach a permissions policy to an identity
- Attach a permissions policy to a secret

Attach a Permissions Policy to an Identity

Identity-based policies can be used to:

Give a user access to several secrets

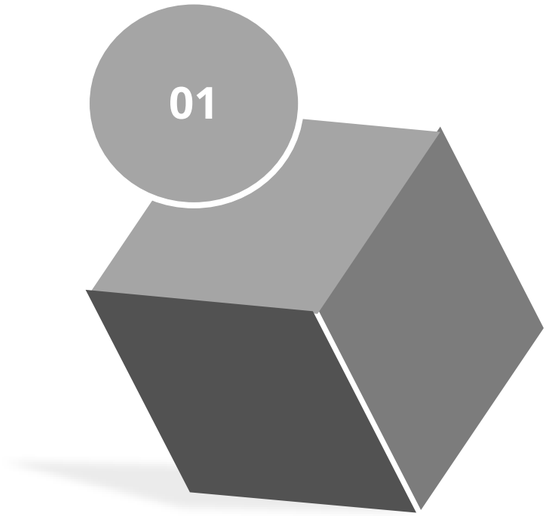


Allow an IAM group to access secrets

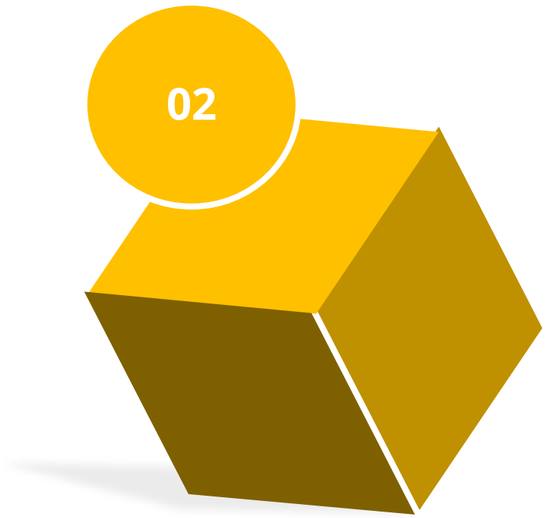
Control who can create new secrets and who can access already generated secrets

Attach a Permissions Policy to a Secret

In a resource-based policy, users determine who will have access to the secret and what actions they may do with it. Resource-based policies can be used to:



Grant access to a single secret to multiple users and roles



Grant access to users or roles in other AWS accounts

AWS CLI

Use get-resource-policy to obtain the policy associated with the secret. The policy related to the secret is retrieved using the CLI command below:

```
aws secretsmanager get-resource-policy --secret-id production/MyAwesomeAppSecret
{
  "ARN": "arn:aws:secretsmanager:us-east-
2:123456789012:secret:production/MyAwesomeAppSecret-a1b2c3",
  "Name": "MyAwesomeAppSecret",
  "ResourcePolicy": "{\"Version\":\"2012-10-
17\",\"Statement\":{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::1111
22223333:root\",\"arn:aws:iam::444455556666:root\"]},\"Action\":[\"secretsmanager:Ge
tSecret\",\"secretsmanager:GetSecretValue\"],\"Resource\":\"*\"}}"
}
```

AWS CLI

Use delete-resource-policy to remove the policy associated with the secret. The following CLI command removes the policy from the secret:

```
$ aws secretsmanager delete-resource-policy --secret-id
production/MyAwesomeAppSecret
{
  "ARN": "arn:aws:secretsmanager:us-east-
2:123456789012:secret:production/MyAwesomeAppSecret-a1b2c3",
  "Name": "production/MyAwesomeAppSecret"
}
```

AWS SDK

Use Get-Resource-Policy to obtain the policy that is associated with a secret. Use Delete-Resource-onePolicy to remove a policy that is associated with a secret.

Permissions for Users in a Different Account

Step 1: Attach a resource policy to the secret in Account1

The following policy allows ApplicationRole in Account2 to access the secret in Account1.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Account2:role/ApplicationRole"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Permissions for Users in a Different Account

Step 2: Add a statement to the key policy for the KMS key in Account1

The ApplicationRole in Account2 is permitted to use the KMS key in Account1 to decode the secret by the following key policy clause. Add this clause to the key policy for the KMS key in order to use it.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::Account2:role/ApplicationRole"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey"
  ],
  "Resource": "*"}
```

Permissions for Users in a Different Account

Step 3: Attach an identity policy to the identity in Account2

The following policy permits ApplicationRole in Account2 to access the Account1 secret and decrypt the secret value using the Account1 encryption key.

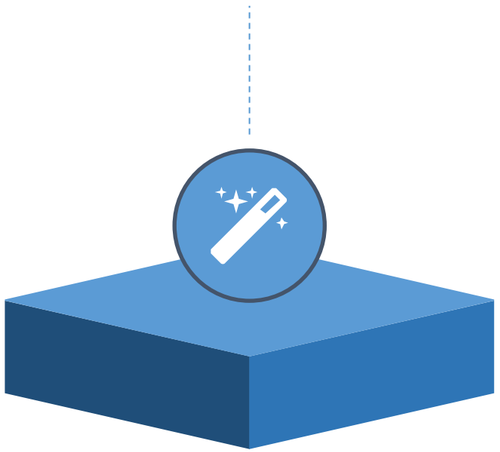
```
{
 "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "SecretARN"
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws:kms:Region:Account1:key/EncryptionKey"
    }
  ]
}
```

Permissions Policy

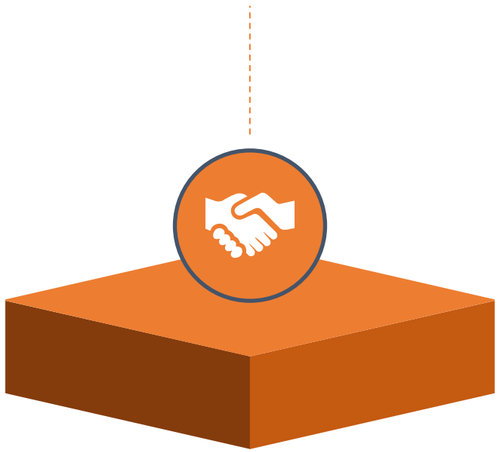
There are many similarities between the permissions policies that users apply to resources and identities. The users might include the following things in a secret access policy:

Example

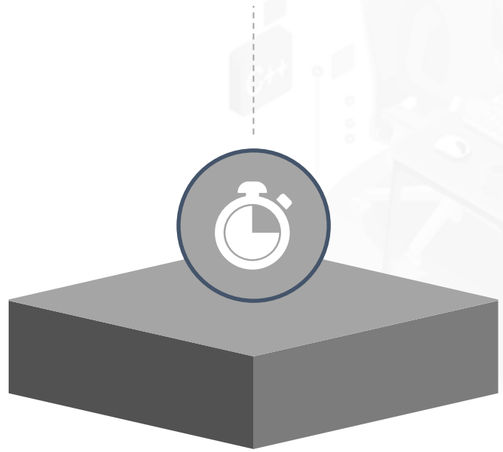
Principal



Action



Resource



Secrets Manager Concepts

A secret in Secrets Manager is made up of secret information, the secret value, and secret-related metadata. A string or binary can be used as a secret value.
The following concepts are important for understanding how Secrets Manager works.

A secret in Secrets Manager is made up of secret information, the secret value, and secret-related metadata. A string or binary can be used as a secret value.

Secrets Manager Concepts

To make it more challenging for an attacker to access the credentials, a secret is rotated on a regular basis.

Rotation

Secrets Manager Concepts

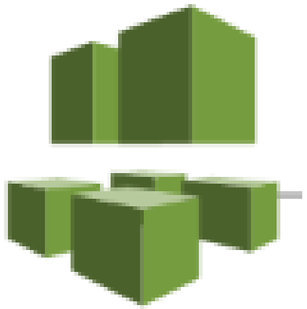
Versions of a secret store copies of the secret value that has been encrypted.

Version

Systems Manager

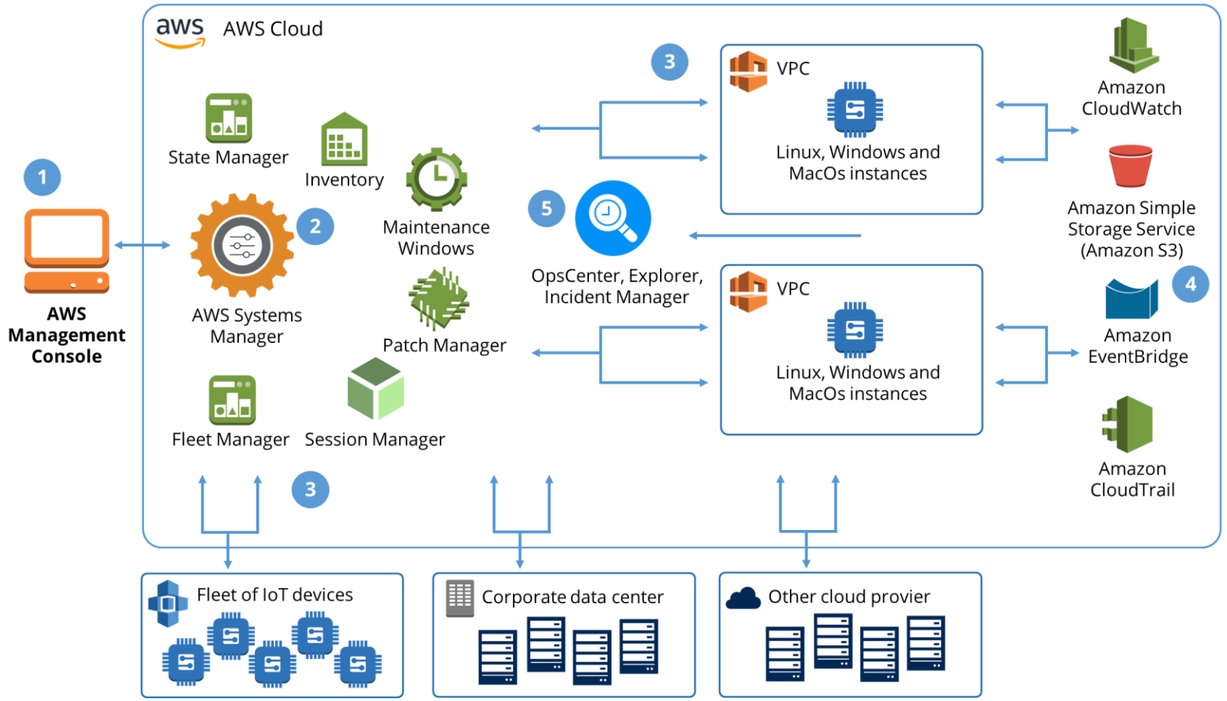
AWS Systems Manager

AWS Systems Manager is a set of tools that can assist users in managing the infrastructure and applications that are hosted on the AWS Cloud.



How Systems Manager Works?

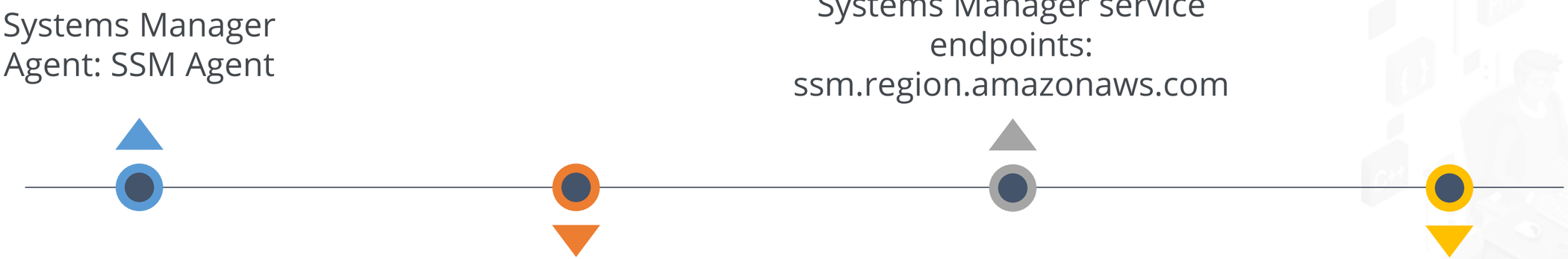
The activities performed by the Systems Manager on the resources are shown in the following diagram with a detailed description of each interaction between different components:



Source:https://docs.aws.amazon.com/systems-manager/latest/userguide/what-is-systems-manager.html

Systems Manager Service Name History

The previous names for AWS Systems Manager are listed below:



Systems Manager Service Name History

The previous names for AWS Systems Manager are listed below:

AWS Config rule identifier: EC2_INSTANCE _MANAGED_BY_SSM

AWS Command Line Interface (AWS CLI) commands: aws ssm describe-patch-baselines

AWS Identity and Access Management (IAM) managed policy names: AmazonSSMReadOnlyAccess

Systems Manager resource ARNs: arn:aws:ssm:region:account-id:patchbaseline/pb-07d8884178EXAMPLE

TECHNOLOGY

AWS Config

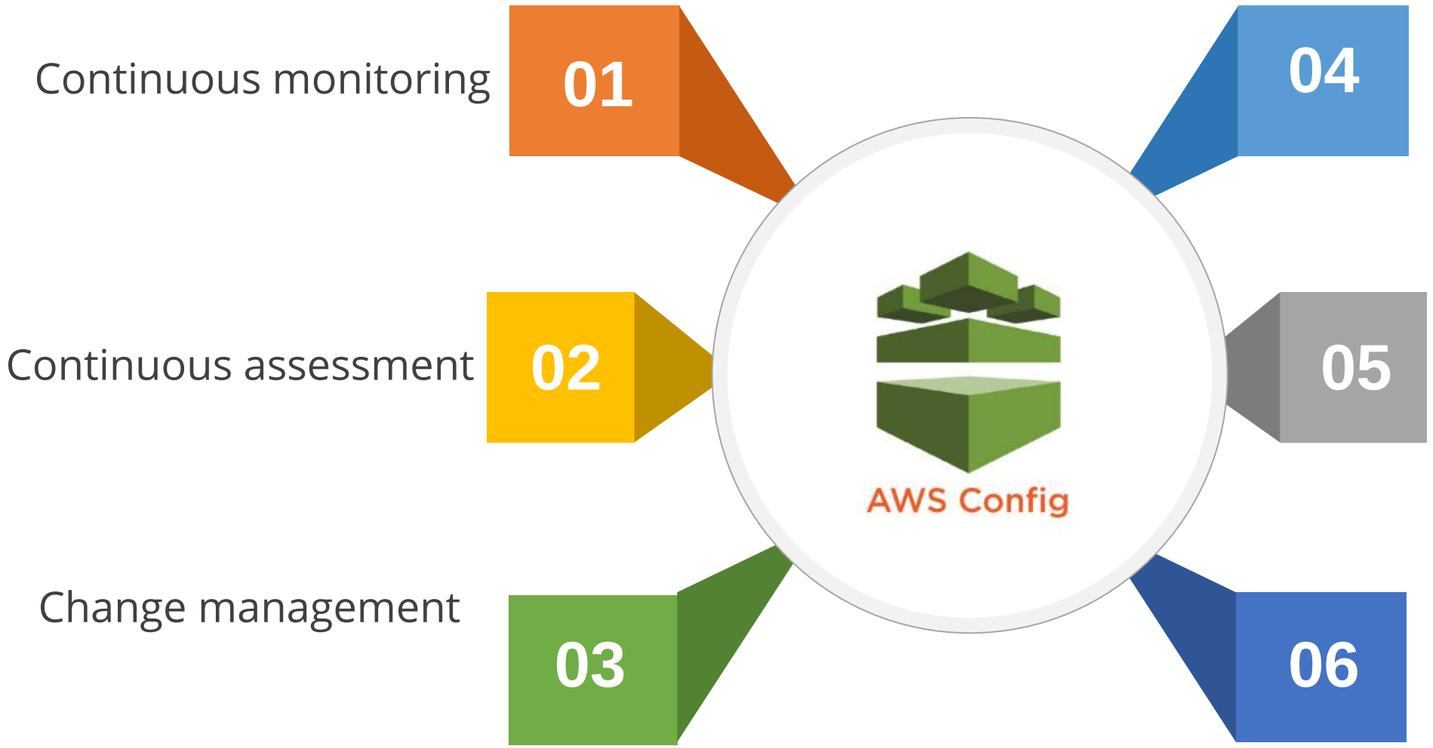
simplilearn

AWS Config

AWS Config is a service that allows users to inspect, audit, and review the AWS resource setups.

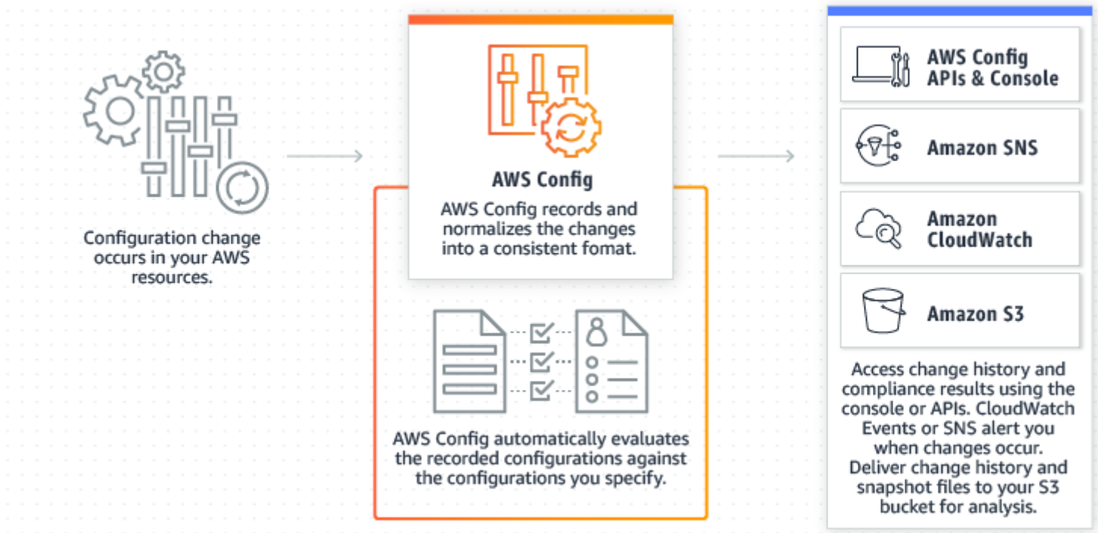
Config monitors and records the AWS resource configurations in real time and allows users to manage the comparison of recorded configurations to desired configurations.

Features of AWS Config



How does it work?

When a resource's configuration changes, AWS Config creates configuration items, and starting with the moment users start the configuration recorder, it keeps historical recordings of the configuration items of the resources.



Source:https://aws.amazon.com/config/

Use cases

The following use cases of AWS Config can be listed as follows:

- Discovery
- Change management
- Continuous audit and compliance
- Compliance-as-code framework
- Troubleshooting
- Security analysis

AWS Config S3 Bucket Encryption Compliance

Duration: 8 mins

Problem Statement:

You have been assigned a task to configure S3 bucket encryption compliance.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Setting up the AWS config
2. Create rules in AWS Config Console

AWS Config Automated Remediation

Duration: 8 mins

Problem Statement:

You have been assigned a task to configure automated remediation.

ASSISTED PRACTICE

Assisted Practice: Guidelines

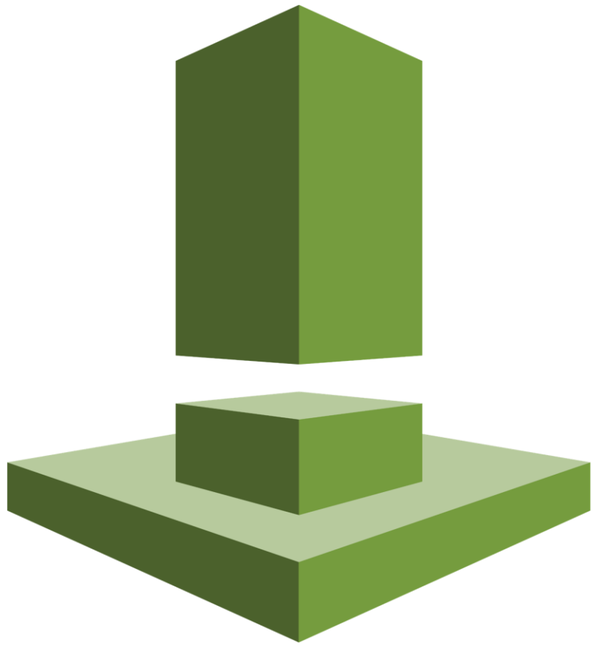
Steps to be followed:

1. Open the AWS Config console by logging into the AWS Management Console at https://console.aws.amazon.com/config/
2. From the navigation pane, choose Preferences
3. Choose Disable Trusted Advisor

AWS Inspector, Trusted Advisor

AWS Inspector

Amazon Inspector is an automated vulnerability management program that continuously checks for software flaws and accidental network exposure in an AWS workload.



It looks for vulnerabilities in an Amazon Elastic Compute Cloud(EC2) instance and creates reports.

source:https://aws.amazon.com/config/

AWS Inspector

Some features of Amazon Inspector are as follows:

01

Central Management
It can be managed by a single administrator.

02

Real Time Scans
Eligible objects and resources are scanned automatically.

03

AWS Inspector Risk Score
A report mentioning the weakness and severity.

04

Amazon Inspector Dashboard
It displays a detailed list of findings.

05

Manage Findings
All vulnerabilities are listed according to their type.

06

Monitor Vulnerability
EventBridge can be used to process them in almost Real-time.

AWS Inspector

- AWS inspector provides recommendations following the AWS best practices.
- It performs searches to optimize the security of the AWS infrastructure.
- It aids in reducing cost, boosting performance, and enhancing security.

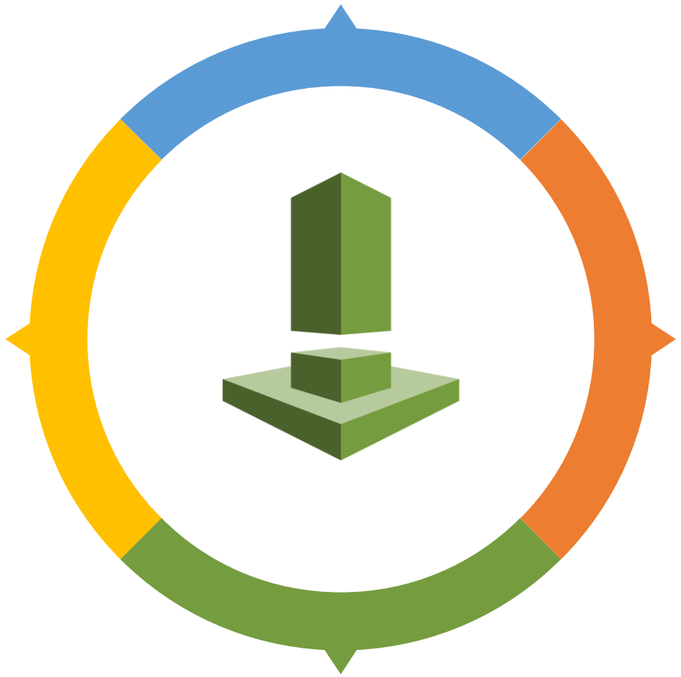
AWS Inspector: Access

AWS Management Console

AWS Management Console can be accessed through a browser.

AWS Command Line

It is a faster and more convenient approach to performing tasks.



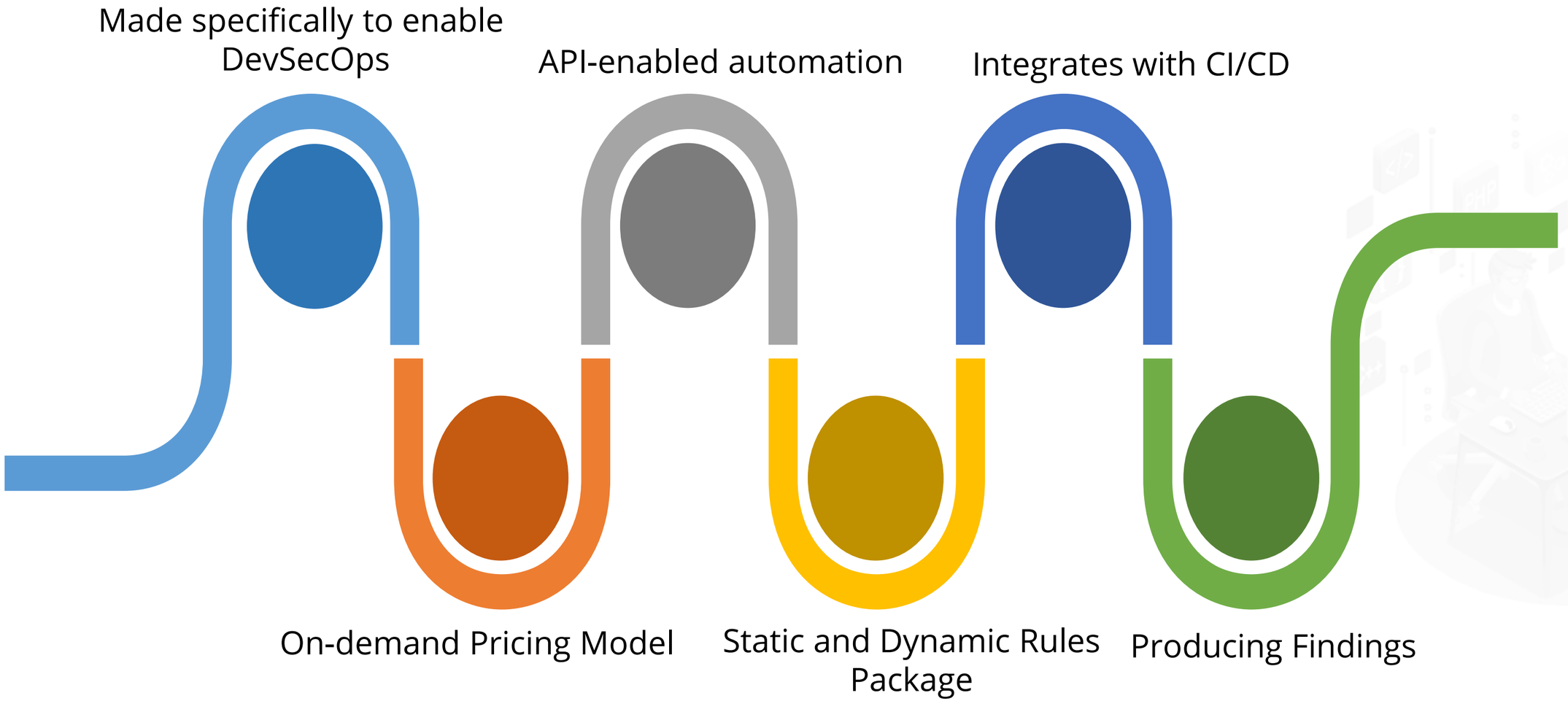
AWS SDKs

They provide an easy and convenient way to manage errors.

AWS Inspector REST API

It can be used to send web requests to the AWS Inspector.

AWS Inspector: Vulnerability Assessment



AWS Inspector: Use Case

Some common use cases of AWS inspector are:

- Before Deployment
- Migration
- Production

Network Reachability

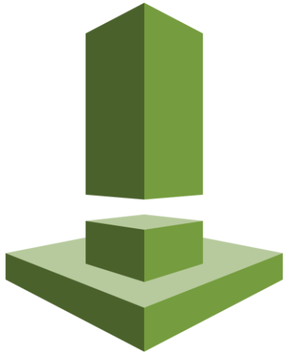
- Validate Network; Find Vulnerability.
- VPC Configuration Mistake.
- Check that no exposure has opened.

Host Assessment

- Check Gold AMIs; DevOps Pipeline.
- Check for Software Changes;New(Zero Day) Vulnerabilities
- Check for Software Changes;New(Zero Day) Vulnerabilities

Working of AWS Inspector

AWS Inspector works in the following ways:

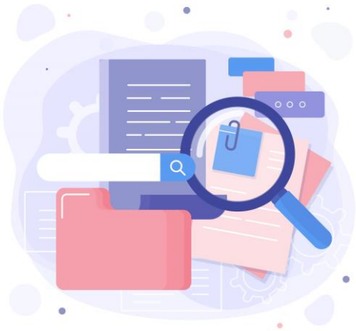


Configure Assessment

- Automated Scans
- Configure assessment

Run Assessment

- One Click for network reachability
- Deep Scans of eligible resources



Findings

- Vulnerability
- Resource Affected
- Recommendation

Action

- Remediation
- Inspector Partners;

1. SIEM
2. Reporting
3. Ticketing

- Store in Database

Three Levels of Warning by AWS Inspector

There are three levels of warning given by the AWS Inspector.

Red

- Critical error in one or more resources.
- Recommended action is “action recommended”

Yellow

- Warning condition for one or more resources
- Recommended action is “investigation recommended”.

Green

- No vulnerability.
- No recommendation. All resources are fine.

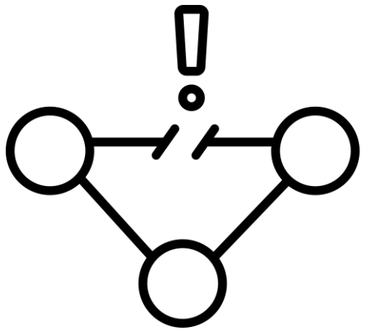
Benefits of AWS Inspector

Cost Optimization

Performance

Security

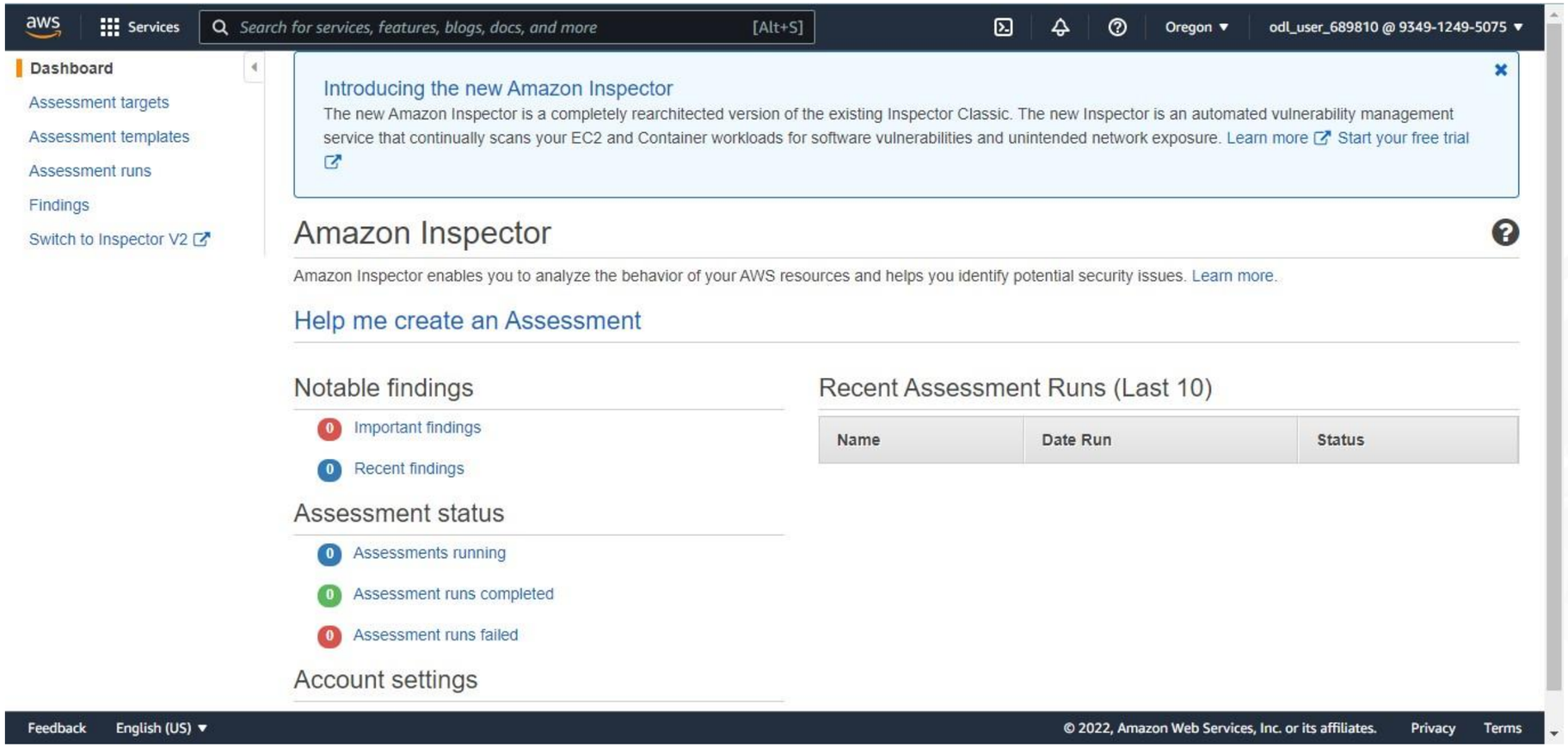
Services



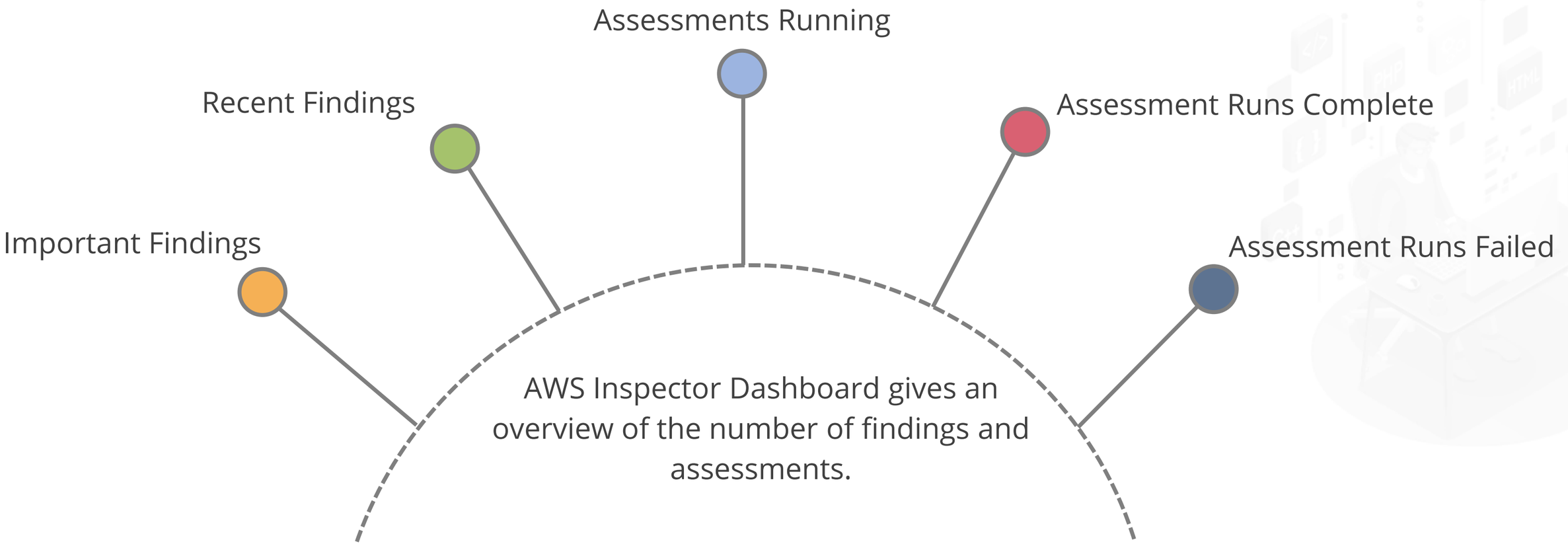
Fault Tolerance

Integrated Usage Walk-through

AWS Inspectors Dashboard gives out information regarding the findings.



Integrated Usage Walk-through



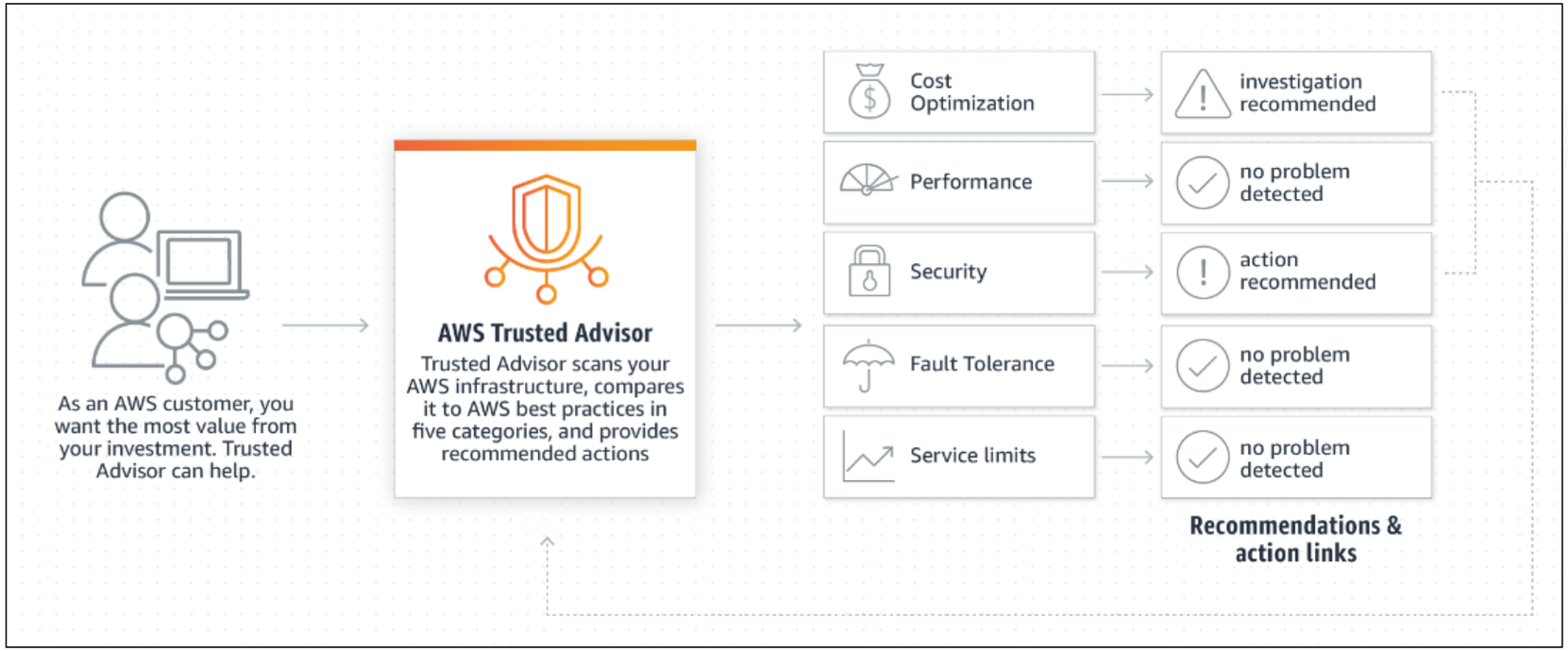
Trusted Advisor

AWS Trusted Advisor provides recommendations that help users follow AWS best practices.

- AWS Trusted Advisor provides recommendations that help users follow AWS best practices.
- Trusted Advisor evaluates a user's account by using checks.
- These audits help users optimize their AWS infrastructure, boost performance and security, reduce expenses, and keep an eye on service quotas.

How Trusted Advisor works?

AWS Trusted Advisor offers recommendations to help users adhere to AWS best practices.



Trusted Advisor uses checks to analyze the account.

Source:https://aws.amazon.com/premiumsupport/technology/trusted-advisor//

AWS Inspector for Network Reachability, Vulnerability, and Host Hardening

Duration: 8 mins

Problem Statement:

You have been assigned a task to configure AWS inspector for network reachability, vulnerability, and host hardening.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Developing the AWS Inspector
2. Creating the AWS Inspector dashboard

Trusted Advisor

Duration: 8 mins.

Problem Statement:

You have been assigned a task to create a trusted advisor.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Open the Trusted Advisor console
2. From the navigation pane, choose Preferences
3. Choose Email Save Preferences

GuardDuty

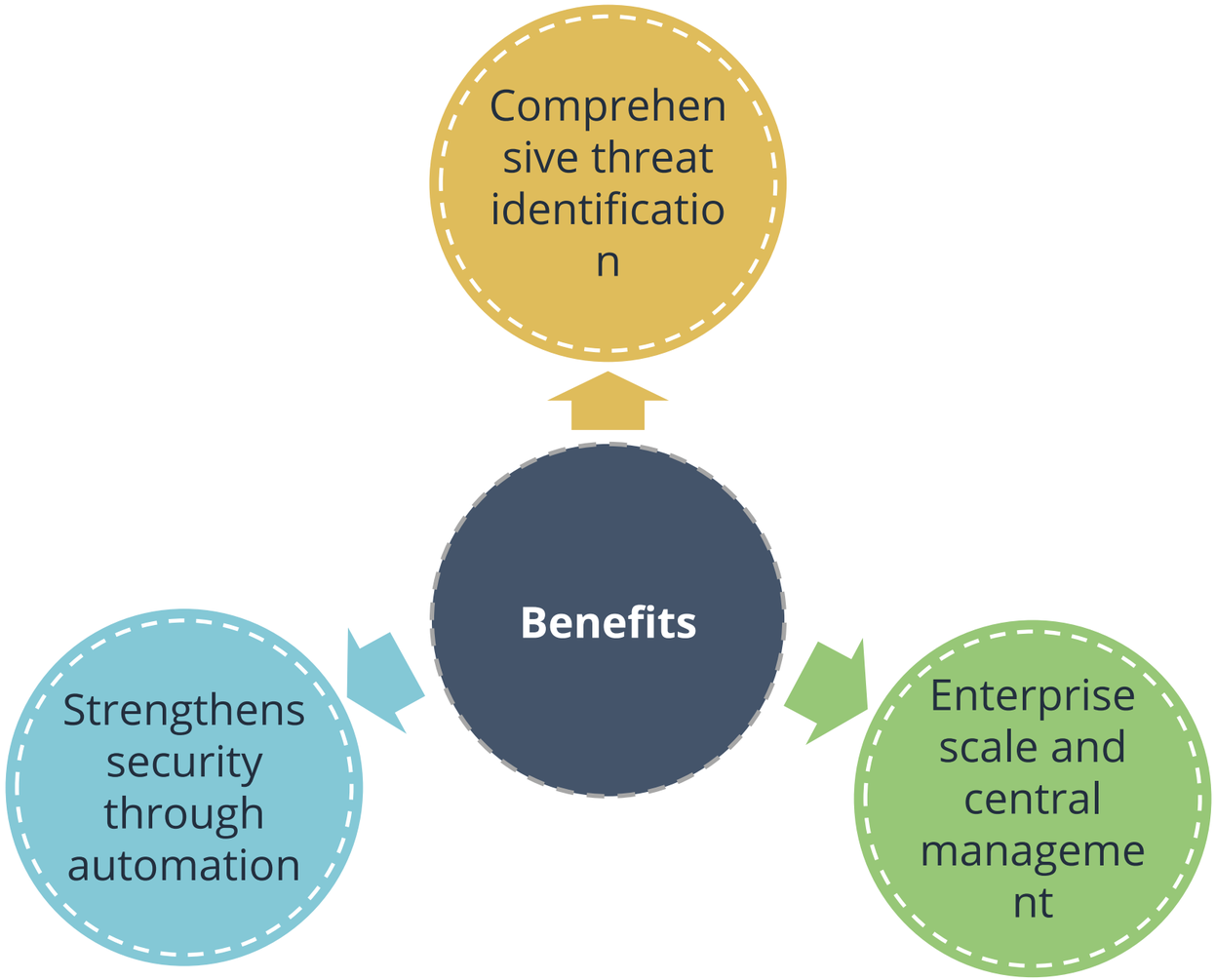
Amazon GuardDuty

Amazon GuardDuty is a threat detection service that delivers in-depth security findings for visibility and remediation.

It continuously scans your AWS accounts and workloads for malicious activity.

Source: twitter.com

Amazon GuardDuty Benefits



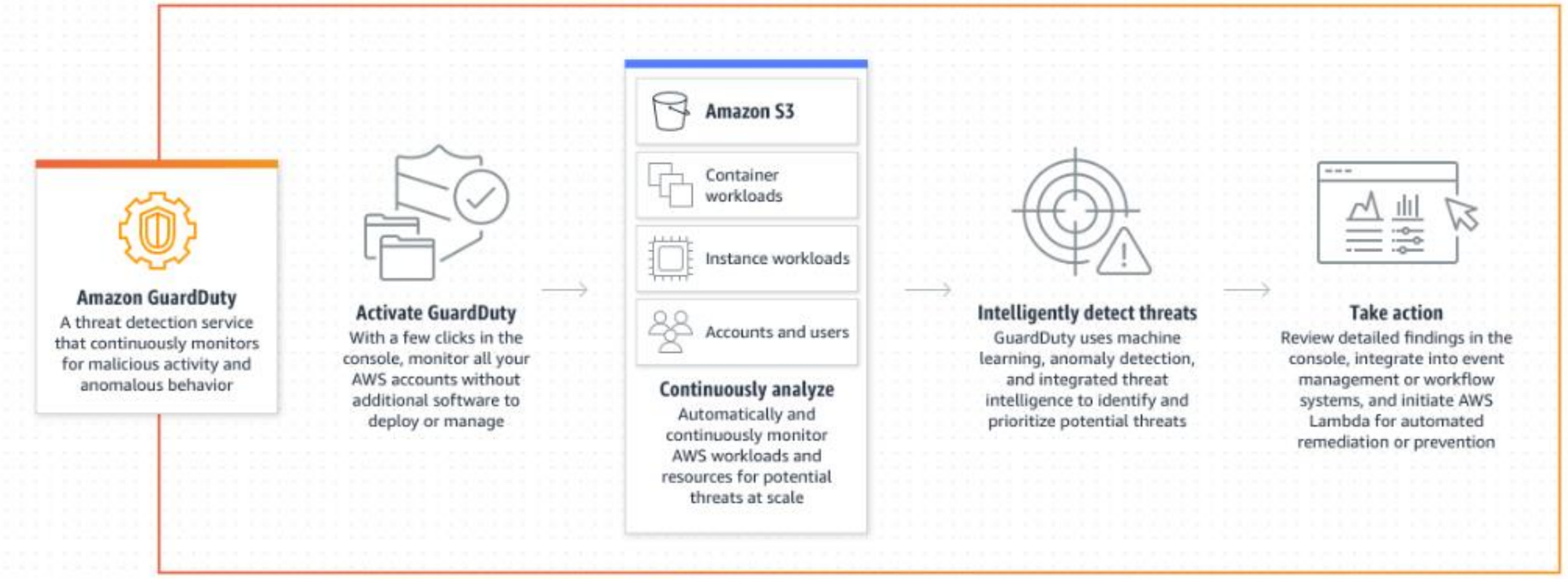
Accessing GuardDuty

With GuardDuty, users can cooperate in the following ways:

- GuardDuty Console
- SDKs for AWS
- HTTPS API for GuardDuty

How GuardDuty works?

AWS CloudTrail, Amazon VPC Flow Logs, and domain name system (DNS) logs are being continuously examined for any malicious activities by Amazon GuardDuty.
The service performs analysis in close to real time by utilizing built-in threat intelligence, anomaly detection, and machine learning capabilities created by the AWS security team.



Source:https://aws.amazon.com/guardduty/

GuardDuty - Managing Findings from Multiple Accounts

Managing multiple accounts with AWS Organizations

Users can designate a particular account as the organization's delegated administrator for GuardDuty if that account is a member of an organization in AWS Organizations

Managing multiple accounts by invitation

Users can specify an administrator account in GuardDuty and then use the administrator account to invite additional AWS accounts to become member accounts

GuardDuty administrator and member accounts

The administrator account can control specific GuardDuty features on behalf of the member accounts when using GuardDuty in a multi-account environment.

Use cases

The following use cases of AWS GuardDuty can be listed as follows:

- Stop unauthorized activity
- Enable continuous monitoring and analysis
- Simplify forensics
- Intelligent threat detection

Three Main Types of Threats

- Compromised instances
- Reconnaissance
- Compromised accounts

Available Regions

Currently, the following AWS regions offer support for Amazon GuardDuty:

Detecting Known Threats

Known malware infected hosts

Anonymizing proxies

Sites hosting malware and hacker tools

Crypto-currency mining pools & wallets

Great catch-all for suspicious & malicious activity

Detecting Unknown Threats

Algorithms to detect unusual behaviour

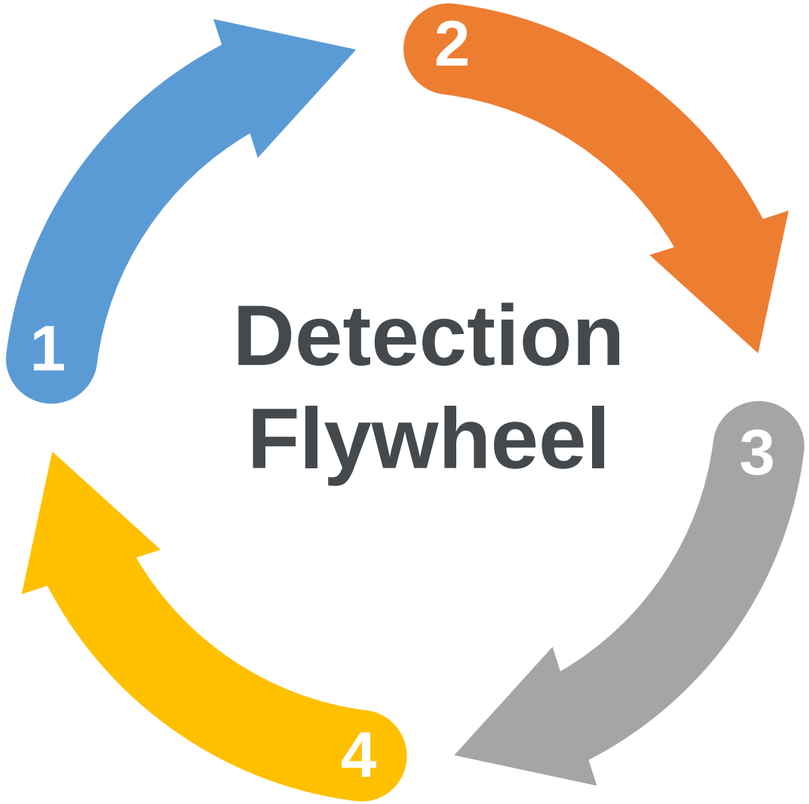
- Analyzing signaling patterns for signatures.
- Profiling normal and looking at deviations.
- Machine Learning Classifiers

Larger R&D Effort

- Highly skilled data scientists to study data.
- Develop theoretical model
- Experiment with implementations
- Testing,tuning and validations

Detection Flywheel

Amazon GuardDuty learns new threats, consumes threat intelligence, develops new analytics and new detections



Customer and partners receive a steady flow of detection streams and develop new detections.

AWS Security investigates threats and patterns, produces threat intelligence and provides guidance to all service teams

AWS Field Organizations consists of AWS Support/TAM Solutions Architect and Professional services

CloudWatch Events

Amazon CloudWatch Events provides a near-real-time stream of system events describing changes to Amazon Web Services (AWS) resources.

Source:https://www.slideshare.net/AmazonWebServices/amazon-guardduty-intelligent-threat-detection-and-continuous-monitoring-to-protect-your-aws-accounts-and-workloads?qid=6b0a46cf-9937-42c7-a81d-dc09dca1e939&v=&b=&from_search=2

Languages Supported

The information regarding GuardDuty is available in eight languages including **Chinese,English,French,German,Japanese Korean,Portugese and Spanish**.

Macie

Amazon Macie

Amazon Macie is a fully managed data security and data privacy solution that discovers and protects sensitive data in AWS using machine learning and pattern matching.

Benefits of Macie

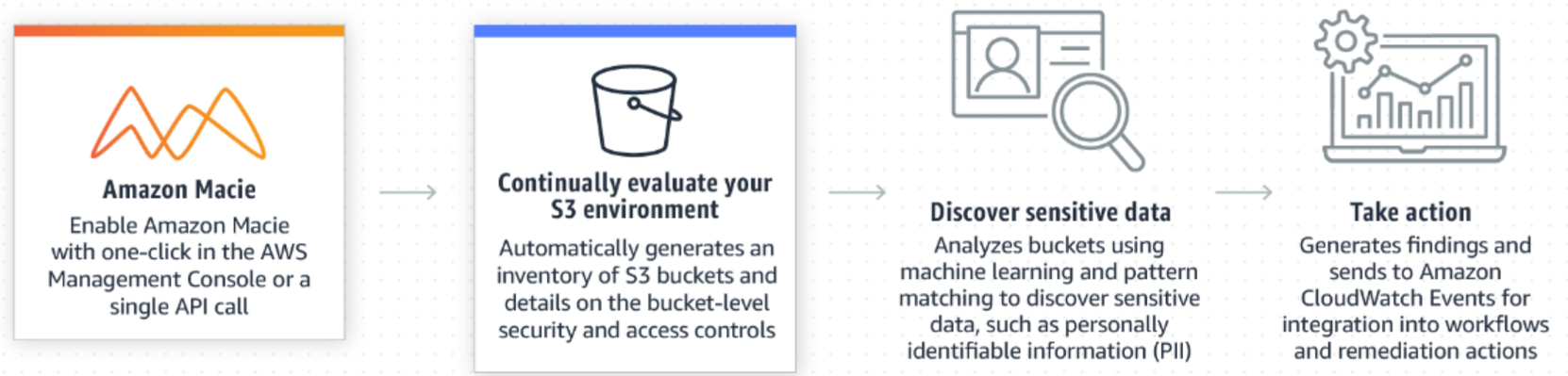
01 Discover your sensitive data at scale

02 Visibility of your data security posture

03 Easy to setup and manage

How does it work?

After activating Macie for the AWS account, the S3 bucket list will be created in the region where user activated it. Macie will also start monitoring the access control and security of the buckets.



Source:https://aws.amazon.com/macie/

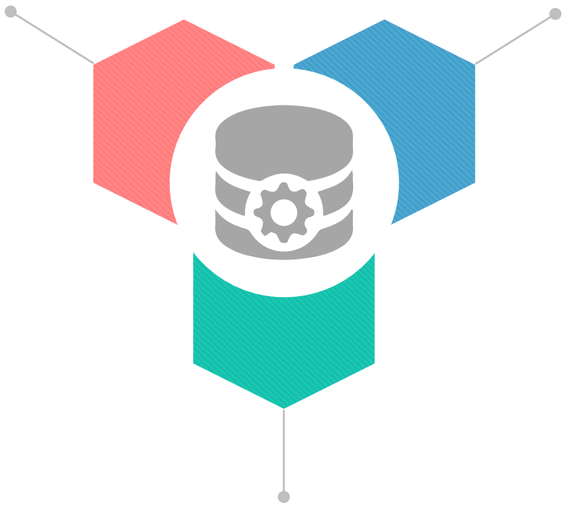
Use Cases

The following use cases of AWS Macie can be listed as follows:

- Assessing your data privacy and security
- Maintaining regulatory compliance
- Identifying sensitive data in data migrations

Features Of Macie

Macie summary dashboard



Macie Jobs

Macie Findings

Other Security Services

Security Hub

AWS Security Hub is a cloud security posture management service that checks for quality standards, collects alarms, and allows automated repair.

Source :https://dev.classmethod.jp/articles/aws-security-hub-check-count-use-aws-cli/

Security in AWS Security Hub

Both AWS and users share responsibility for security. This is referred to as cloud security and security in the shared responsibility model of the cloud.

Security of the cloud

Security in the cloud

Setting Up AWS Security Hub

The way the accounts are managed determines whether an account needs to manually enable AWS Security Hub. Users have two options for managing accounts:

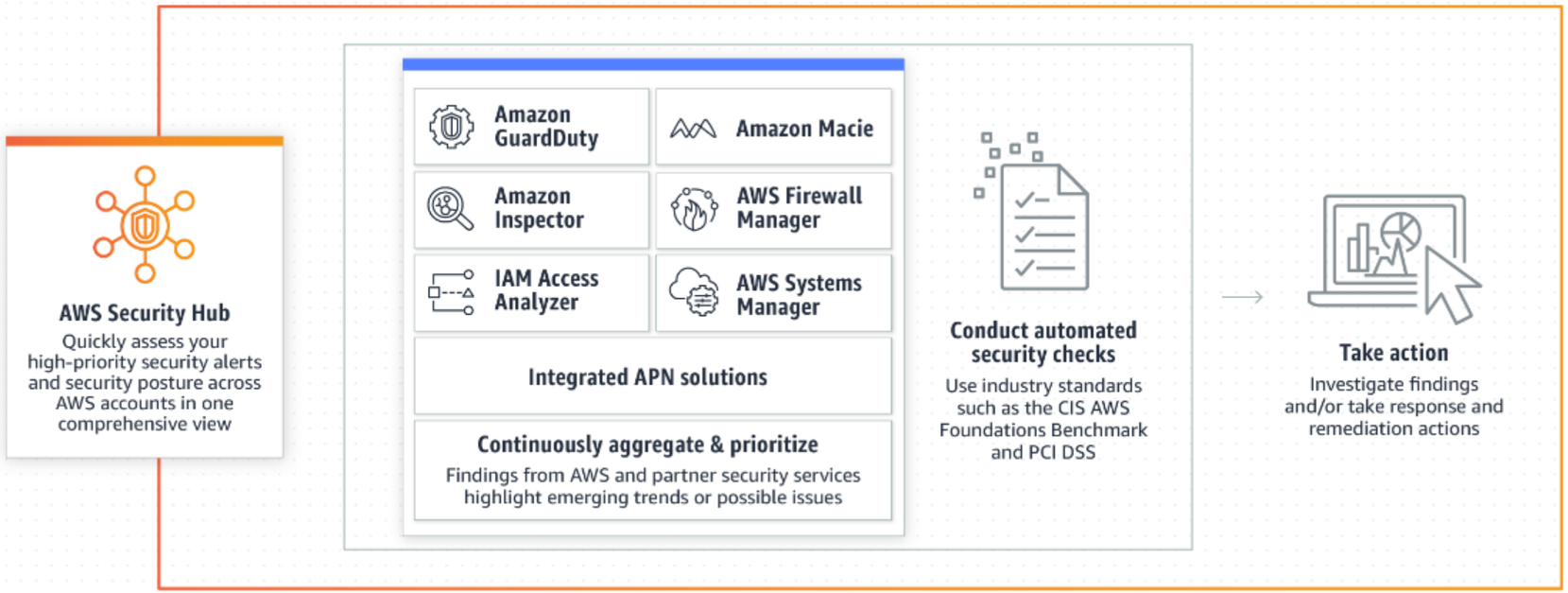
Integration with organizations

Manual account management

Setting Up

How does It work?

To make it easier to respond to security alerts (or findings) from multiple AWS services and partner products, Amazon Security Hub compiles them in a consistent manner.



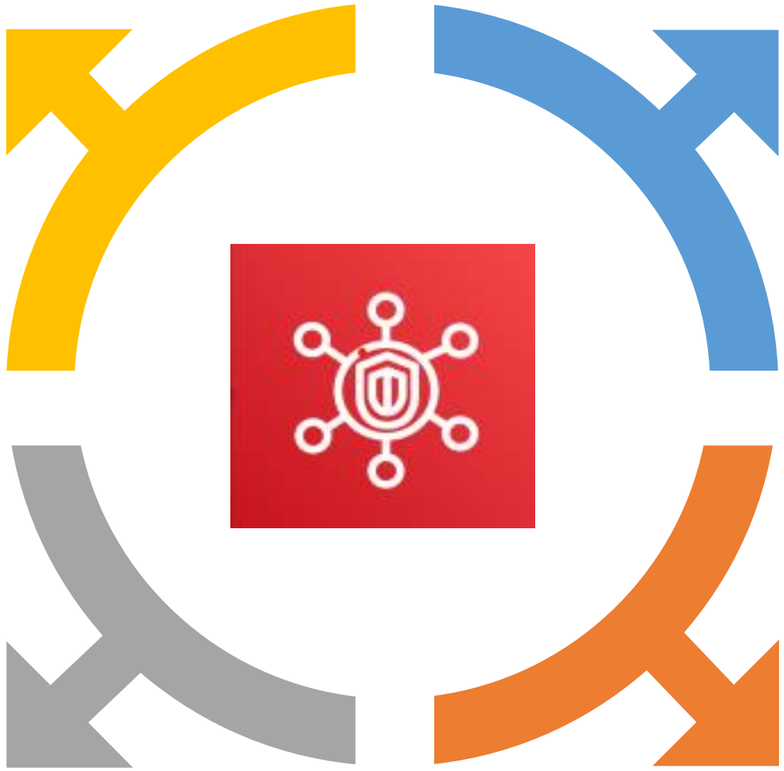
Security Hub makes it easier to analyze and enhance security posture with automatic security best practices powered by AWS Config rules.

Source:*https://aws.amazon.com/security-hub/*

Use Cases

The following use cases of AWS Security Hub can be listed as follows:

Conduct Cloud Security Posture Management (CSPM)



Initiate Security Orchestration, Automation, and Response (SOAR) workflows

Save time and money by simplifying integrations

Correlate the security findings to discover new insights

Security Hub and AWS Services Integration

The AWS services are integrated in the following ways in Security Hub:

Use Amazon GuardDuty to continuously identify threats to the AWS accounts, workloads, and data stored in Amazon S3 in order to lessen risks

Identify personally identifiable information in the S3 buckets with the aid of Amazon Macie

Use Amazon Inspector to check Amazon EC2 instances for common flaws and exposes

Security Hub and AWS Services Integration

These AWS services are integrated:

Examine the policies related to the AWS resources that IAM Access Analyzer use, such as S3 buckets, KMS keys,and Lambda functions

Automate any reaction to the discovered alerts using AWS Lambda, Amazon CloudWatch, and CloudWatch events

Manage web application firewalls and security groups across many AWS accounts using the solution known as AWS Firewall Manager

Security Standards Enabling Security Hub

Users can currently enable three automated checks :

- AWS Foundational Security Best Practices
- AWS Security Hub CIS Benchmark
- Payment Card Industry Data Security Standard (PCI DSS)

Setting Up Integration with AWS Security Hub

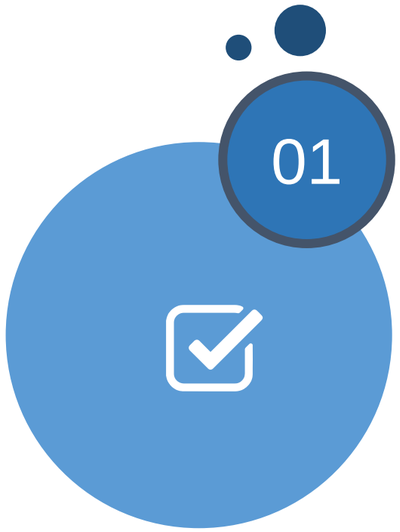
In the AWS environment, the integration builds AWS Lambda functions that transfer findings from AWS Security Hub to Sophos Cloud Optix.

Using a CloudFormation template provided by Sophos, it generates an IAM role.

With the help of CloudFormation StackSets offered by Sophos, it generates AWS Lambda functions that utilize this IAM role.

Top Available Security Standards

The three automated checks users can currently enable are as follows:



AWS Foundational Security Best Practices

AWS Security Hub CIS Benchmark

Payment Card Industry Data Security Standard (PCI DSS)

Detective

Amazon Detective simplifies the analysis, investigation, and rapid identification of the root cause of security findings or suspicious activity. It collects log data from the AWS resources automatically.

Souce: https://www.hava.io/blog/what-is-amazon-detective

How Does Detective Work?

Detective automatically pulls time-based events from AWS CloudTrail and Amazon VPC flow logs, such as login attempts, API calls, and network activity.

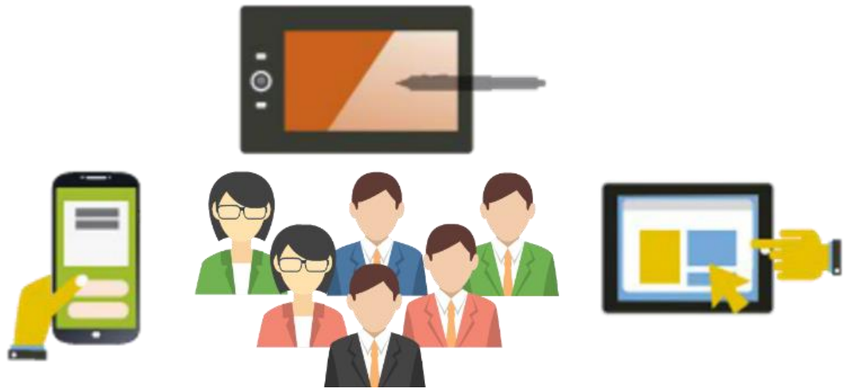
It combines machine learning and visualization to provide a unified, interactive view of the resource activities.

Detective-tailored graphics establish a baseline and summarize account data.

It eliminates the need to organize data or build algorithms.

Who Uses Detective?

An account becomes the administrator account for a behavior graph when it enables Detective. A behavior graph is a connected collection of data taken and examined from one or more AWS accounts.



Administrator accounts encourage member accounts to add their data to the behavior graph of the administrator account.

How to Use Amazon Detective?

Alert Analysis is the first use case for Detective. Users can prioritize security warnings and findings using Detective, which could prevent unneeded security escalations.

Another important task performed by Amazon Detective is incident investigation.

Having discovered a potential security concern, users can utilize Amazon Detective.

Amazon Detective Key Concepts

Behavior Graph	Data is ingested, and Detective employs machine learning to determine the typical behavior of traffic, roles, and resources in order to spot anomalies.
Investigation	Users can use Detective to archive the issue if the incidence turns out not to be a problem, such as a false-positive Guard Duty notice.
Account Structure	The behavior graph belongs to the management account, which also uses it to gather source data and carry out research.
Profile Panel Visualizations	Profile panels are designed to give an analyst the answers to a particular query regarding a result and supporting information.
Scope Time	The scope time is a span of time that includes the security event and often begins when suspicious activity is first noticed and ends when it stops.

Requirements to Begin Using Amazon Detective

An AWS Account

Given that Detective keeps an eye on AWS resources, this should be quite evident.

Amazon Guard Duty

To enable Detective, users must have a GuardDuty service operational for more than 48 hours.

Permissions Policy

Before users may enable detective, they must attach a permissions policy to the IAM principle.

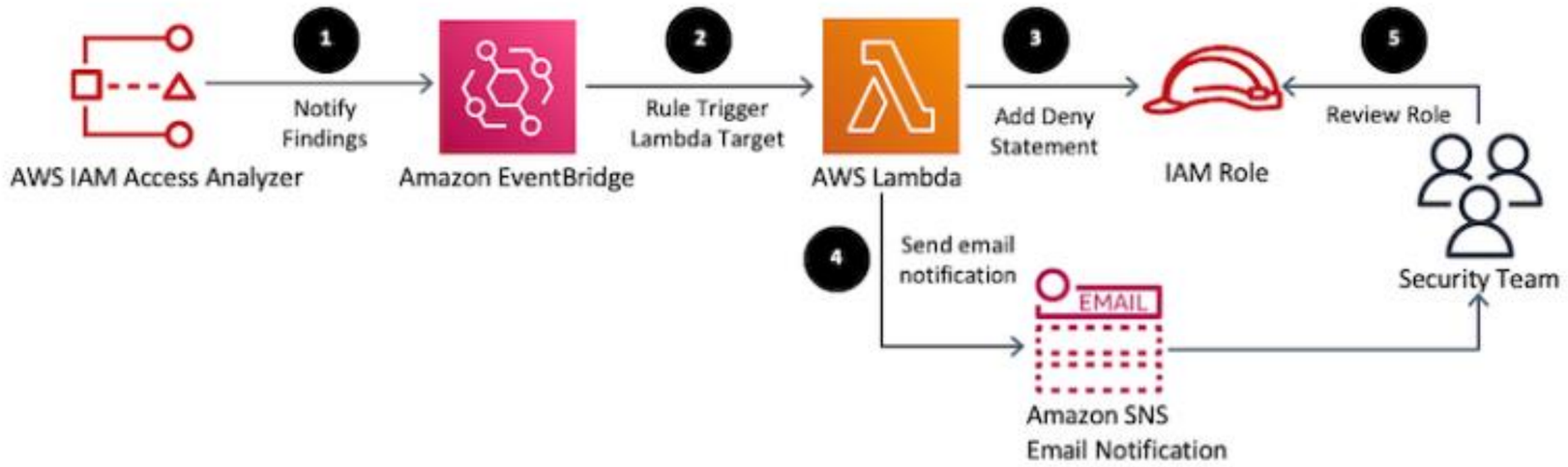
Pricing of Detective

Amazon Detective offers a free 30-day trial.
Let us look at the cost per account, region, and month for the Asia Pacific:

Cost	Cost	Cost	Cost
$2.50 per GB	$1.25 per GB	$0.63 per GB	$0.31 per GB
Usage	Usage	Usage	Usage
First 1,000 GB	**Next 4,000 GB**	**Next 5,000 GB**	**Next 10,000 GB**

IAM Access Analyzer

For each active discovery, Access Analyzer sends an event to Amazon EventBridge. By configuring an event rule in EventBridge to match an active finding, this solution starts an AWS Lambda function for resolution.



Source:https://aws.amazon.com/blogs/security/automate-resolution-for-iam-access-analyzer-cross-account-access-findings-on-iam-roles/

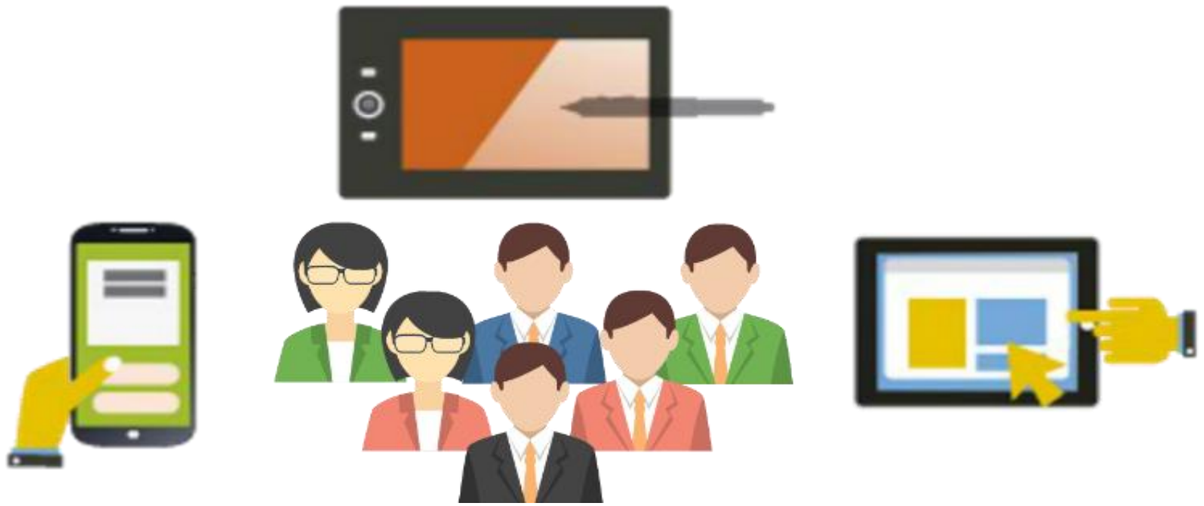
Patch Manager

Patch Manager, an AWS Systems Manager tool, automates the process of patching managed nodes with both security and non-security upgrades.

Patch Manager can be used to apply patches to both operating systems and apps.

Why Patch Manager?

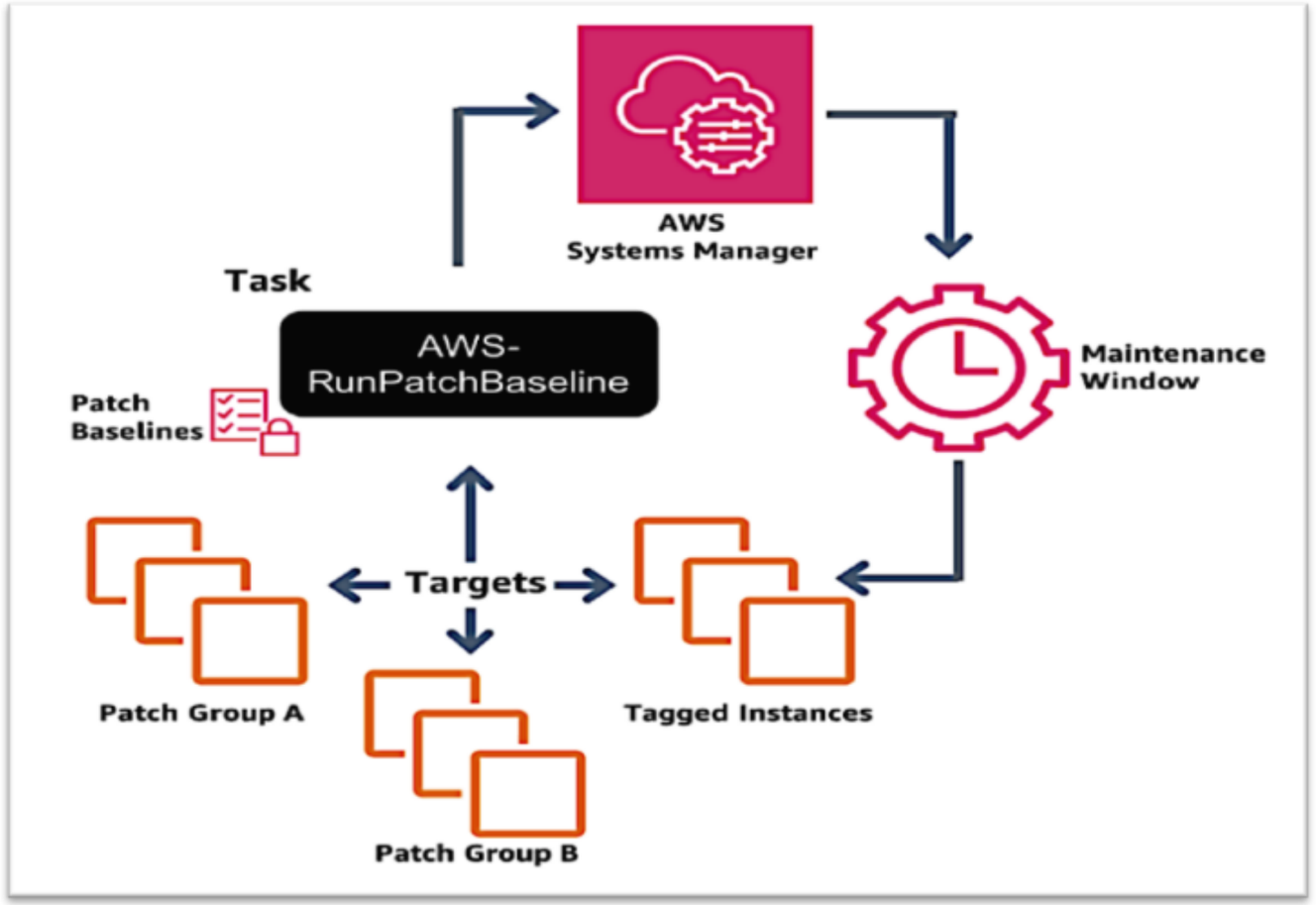
An automated application called AWS SSM Patch Manager can assist users to make a patch management procedure to save time and lower the risk of non-compliance.



By automating the patch maintenance procedure, users can save time and lower the possibility of non-compliance.

Patching Linux Instances with AWS SSM Patch Manager

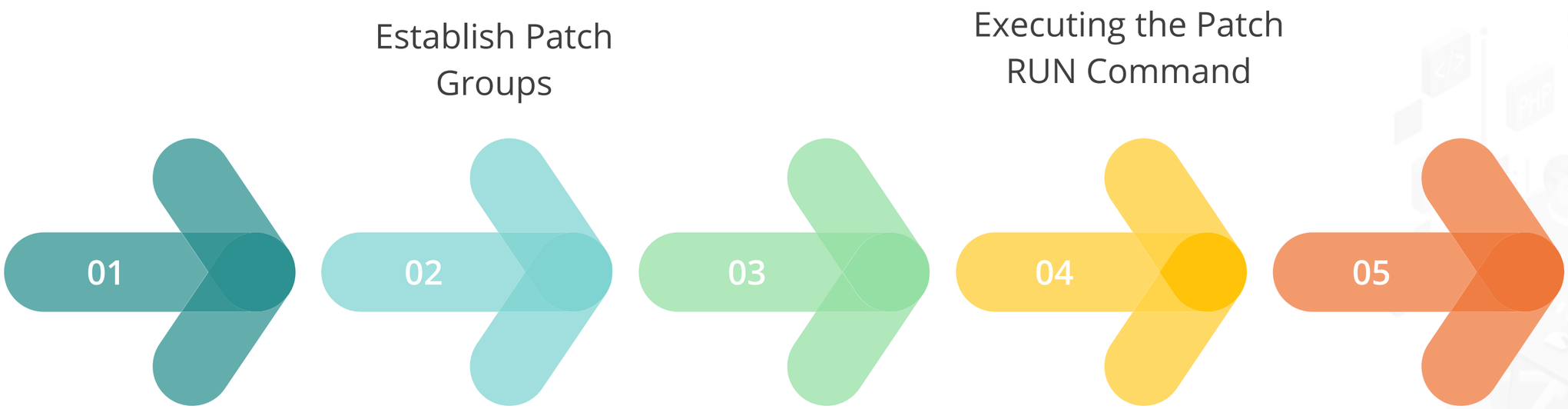
For managed instances, security-related upgrades are patched automatically using **Patch Manager.gh**.



Source:https://www.1cloudhub.com/automate-the-linux-patches-using-aws-system-manager-ssm/

The Patching Process

The five key phases in the Patch Manager patching process must be followed.



Key Takeaways

- IAM policies let users control employee and system access to ensure least-privilege access.
- For web and mobile apps, Amazon Cognito enables quick, secure user authentication, authorization, and management.
- AWS WAF gives control over how traffic enters the applications, allowing users to establish security rules to block common attack vectors.
- AWS Shield offers always-on detection and automatic inline mitigations that reduce application downtime and latency.

Key Takeaways

- Users can automate the process of creating and configuring numerous accounts made possible by AWS Control Tower.
- Users can secure, analyze, and manage secrets in the AWS Cloud, on external services, and on-premises using Secrets Manager.
- Amazon Macie continually checks the Amazon S3 environment and provides an S3 resource summary across all the accounts.
- User's security posture within Amazon Web Services is fully visible through AWS Security Hub (AWS).
- Users can compare their environment to security best practices and standards with the aid of AWS Security Hub.

Create and Configure Groups and Users Using Policies

Duration: 30 mins

Project agenda: To create and configure groups and users using policies

Description: The admin of a corporate decides to create 3 groups (each with any 2 policies) of IAM users (2 in each group), each with their own set of policies for the tasks in their respective groups.

Perform the following:

1. Creating a group
2. Configure the user for CLI operation

LESSON-END PROJECT

TECHNOLOGY

Thank You

simplilearn