DCA Digital Cloud Adviso

Quickest Way(s) To Fix Most Common Misconfiguration Issues in few hours And Outline a Security Strategy Going Forward

An AWS (Amazon Web Services) Security Guide for SMBs And Startups

Powered by aws

Did you know that 19% of security breaches and serious security incidents are caused by Cloud misconfiguration? According to Gartner surveys, these issues cause 80% of all Data Security breaches and until 2025, up to 99% of cloud environment failures will be attributed to human errors. You can see the full report here: Is The Cloud Secure (gartner.com)

Misconfiguration poses a big issue, particularly in terms of access permissions setup. According to insights from expertinsights.com, approximately 83% of organizations have reported that at least one of their cloud data breaches can be attributed to access-related issues, with around 50% stating that at least a quarter of their cloud breaches were access-related. This is largely because 52% of organizations lack visibility into the resources accessible to users and the granted level of permission or privilege.

When embarking on your organization's cloud transformation, it is crucial to have a well-defined and comprehensive strategy in place for implementing security controls. As your business expands and your IT cloud infrastructure becomes more complex, managing access controls for users and resources will become increasingly difficult. Establishing clear and robust security policies from the very beginning is essential to ensure effective security management throughout your journey. These policies should be followed by all members of your organization.

Always keep in mind that Security should be a top priority for any business and should be considered at every stage of its growth. Remember: SECURITY is “JOB "ZERO."

How would you like to reduce the probability of this happening dramatically in less than a few hours with minimum effort?

In this E-book we will go step by step through the of basic Security controls in AWS that any Start Up and SMB company should adopt to avoid human errors. We will try and keep it as simple as possible for you companies to be able to adopt it.

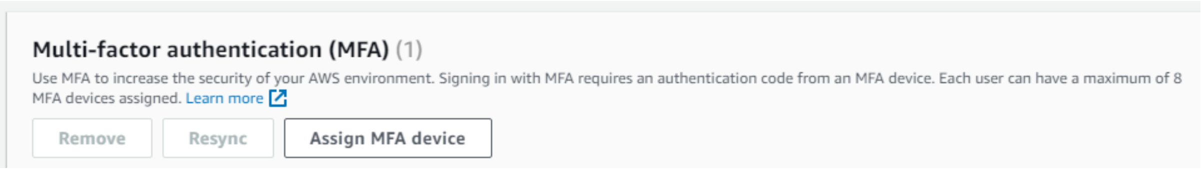
You should always keep in your mind that the Security and Compliance are a shared responsibility between the AWS itself and the customer, which is also known as the Shared Responsibility model. Very often, under the Security pillar of the Well Architected Framework, you will hear that the AWS is responsible for the Security **OF** the Cloud and the customer is responsible for the Security **IN** the Cloud.

You can adopt this guide yourself, ask our expert team to help you implement it, or try our new subscription service from https://digitalcloudadvisor.com.

AWS Account Set up

Every newly created AWS Account automatically includes a Root User profile upon setup. This Root User profile is commonly referred to as the God User due to its unrestricted privileges, making it not suitable for day-to-day operations and warranting its secure isolation.

It is imperative to mandate the utilization of a Multi-Factor Authentication (MFA) device for your Root User profile. Following the industry best practices, it is recommended that all users, including the Root User, are equipped with MFA. Enabling MFA necessitates users to provide an additional layer of authentication, such as a one-time password generated by a mobile application or a physical token, alongside their standard username and password. This added layer of security ensures that even if a malicious actor gains access to user credentials, they would still require possession of the secondary authentication factor to gain unauthorized entry.

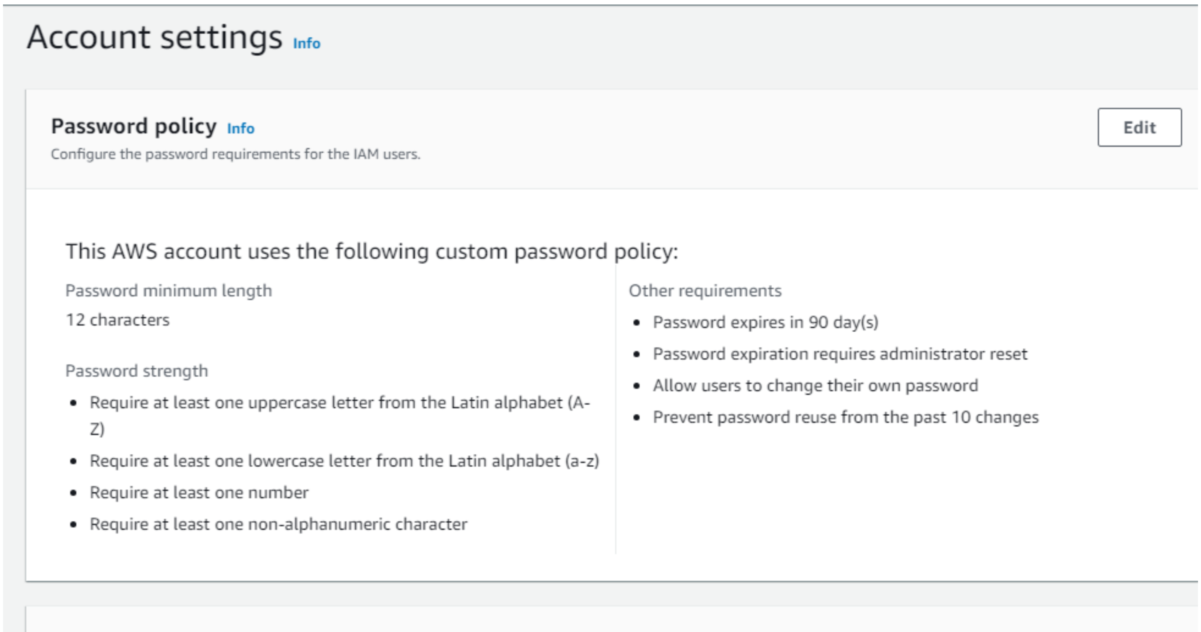


Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

As depicted in the image provided within the IAM settings, we recommend that you allocate your preferred Multi-Factor Authentication (MFA) device.

Considering that a considerable number of security breaches stem from credential hacking, it is crucial to establish a robust password policy, as outlined in AWS best practices. You can configure your password policy directly from the IAM console. The degree of granularity for your password policy can be tailored to your specific requirements, as exemplified below. Following these guidelines, it is advisable to ensure that passwords are a minimum of 8 characters long and encompass a combination of lowercase and uppercase letters, at least one numeric character, and one non-alphanumeric character. Furthermore, within the image's right-hand section, additional customizable requirements can be observed. We recommend the implementation of a policy where passwords expire, as depicted in the image below.



Account settings Info

Password policy Info

Configure the password requirements for the IAM users.

This AWS account uses the following custom password policy:

Password minimum length

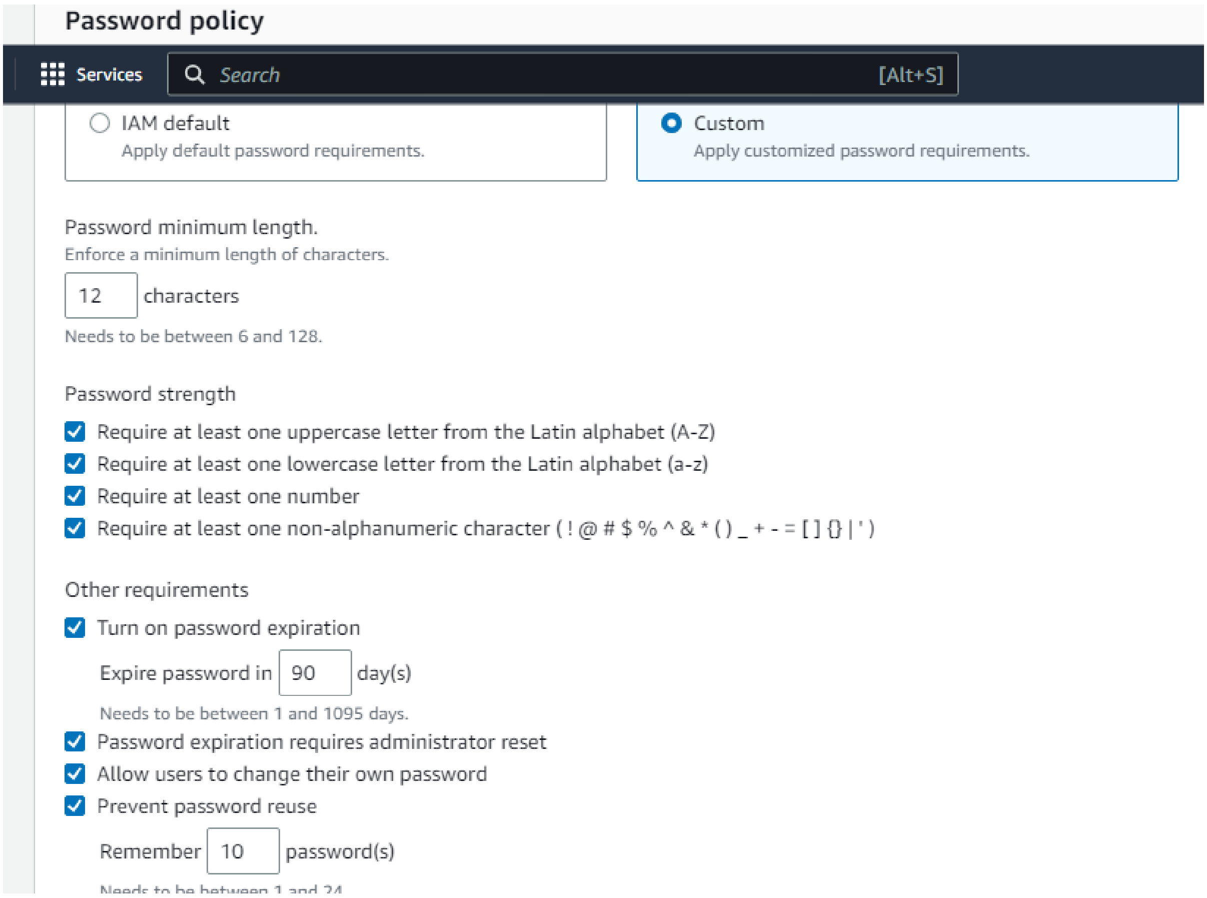
12 characters

Password strength

- Require at least one uppercase letter from the Latin alphabet (A-Z)
- Require at least one lowercase letter from the Latin alphabet (a-z)
- Require at least one number
- Require at least one non-alphanumeric character

Other requirements

- Password expires in 90 day(s)
- Password expiration requires administrator reset
- Allow users to change their own password
- Prevent password reuse from the past 10 changes



Password policy

○ IAM default
Apply default password requirements.

● Custom
Apply customized password requirements.

Password minimum length.
Enforce a minimum length of characters.

12 characters

Needs to be between 6 and 128.

Password strength

☑ Require at least one uppercase letter from the Latin alphabet (A-Z)
☑ Require at least one lowercase letter from the Latin alphabet (a-z)
☑ Require at least one number
☑ Require at least one non-alphanumeric character (! @ # $ % ^ & * ( ) _ + - = [ ] {} | ')

Other requirements

☑ Turn on password expiration

Expire password in 90 day(s)

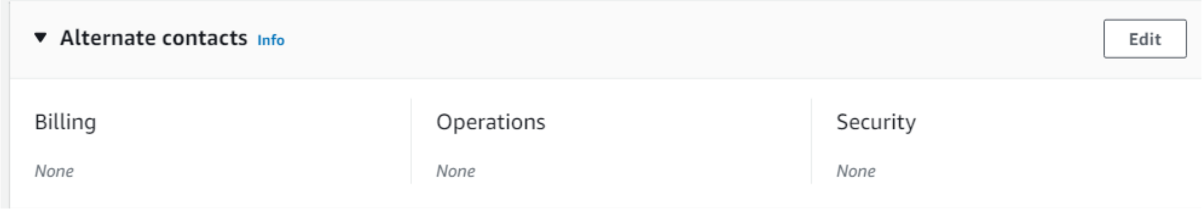
Needs to be between 1 and 1095 days.

☑ Password expiration requires administrator reset
☑ Allow users to change their own password
☑ Prevent password reuse

Remember 10 password(s)

Needs to be between 1 and 24

We strongly recommend configuring an account-level contact using a valid email distribution list. In the process of establishing primary and alternate contacts for your AWS Account, opt for utilizing an email distribution list instead of individual email addresses. This approach safeguards the continuity of ownership and accessibility even as personnel transitions occur within your organization. Equally vital is the establishment of alternate contacts for billing, operational, and security notifications, with the corresponding email distribution lists employed judiciously. AWS employs these designated email addresses to communicate with you, underscoring the significance of ensuring continuous access. Navigate to the account settings page to appropriately update and manage these email addresses.

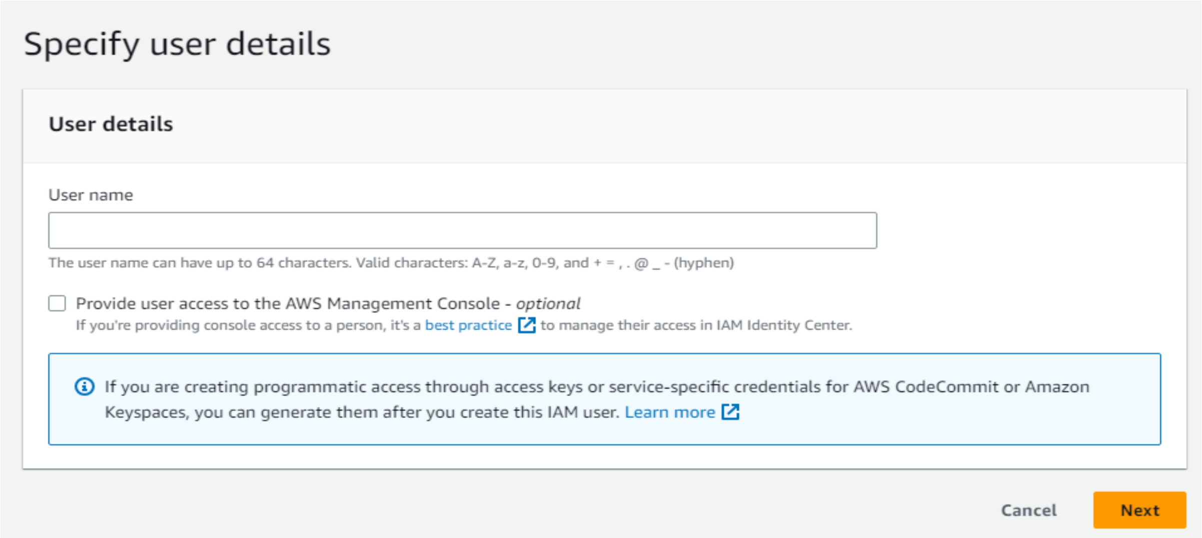


▼ Alternate contacts Info

Billing	Operations	Security
None	*None*	*None*

Moving forward, your subsequent action should involve the creation of an Admin User, who wields the authority to regulate access to AWS resources by associating permission policies with IAM identities, encompassing users, groups, and roles. To forge an IAM user, proceed through the IAM console, navigating to the 'Users' section in the menu on the left. Once there, select the 'Add user' button, effectively initiating the user addition process. As illustrated in the accompanying image, you will designate a distinctive name for the user and indicate their eligibility for AWS Management Console access.

Subsequently, upon advancing to the subsequent screen, you will be presented with a trio of options to choose from. Optimal practice entails grouping users, thereby facilitating streamlined management as your organization expands. Thus, a judicious approach would be to establish a new Administrator group through group creation. On the permissions screen, delineate the group's nomenclature and proceed to select the 'Administrator Access' policy. With these steps executed, the only remaining action is to validate the information on the final screen.



Specify user details

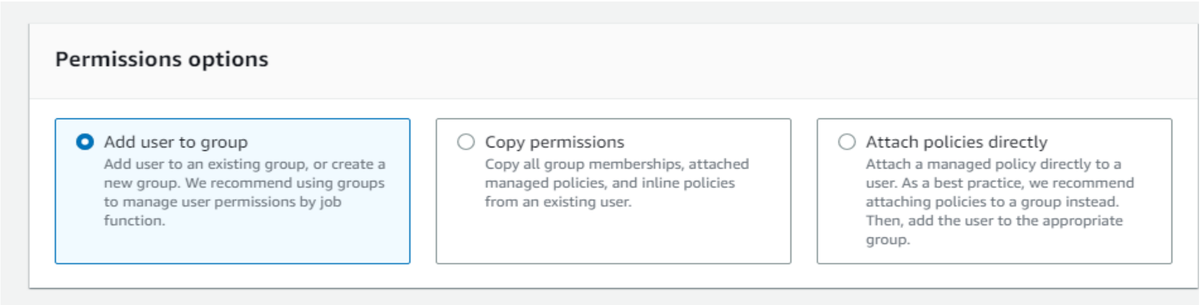
User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

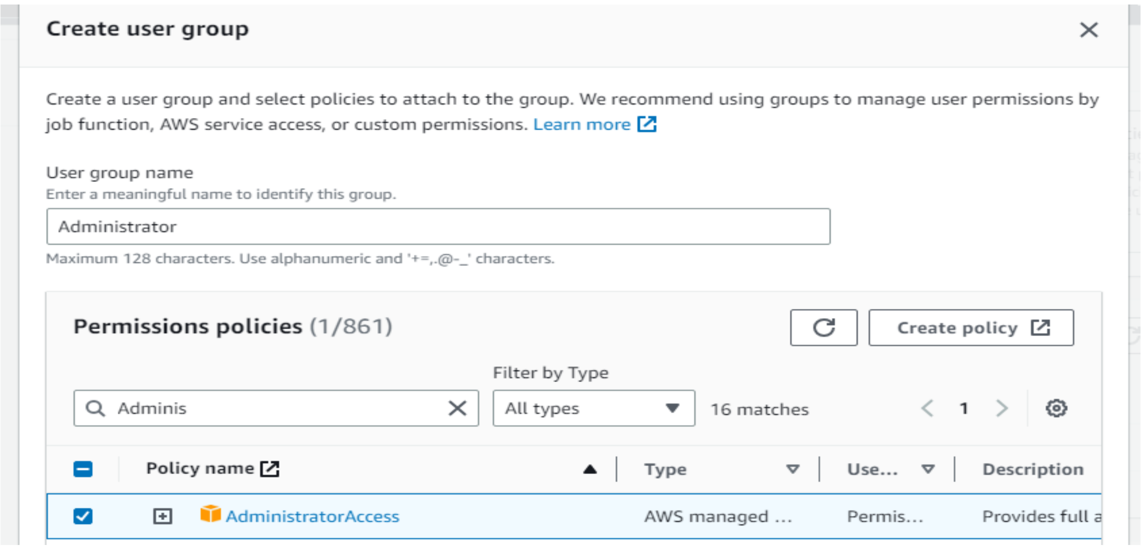


Permissions options

● Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.



Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

User group name
Enter a meaningful name to identify this group.
Administrator
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions policies (1/861)

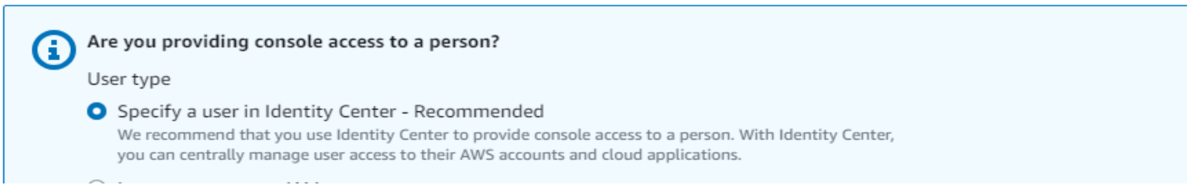
Filter by Type

Adminis | All types | 16 matches | 1

	Policy name	Type	Use...	Description
☑	AdministratorAccess	AWS managed ...	Permis...	Provides full a

Now that you have your Admin User defined, you can logout of your root user and login as the new Admin User. As you did with the same as with the Root User, you should implement MFA for your Admin User.
Next, when you think of creating your subsequent following users, and following best practices you should use the IAM Identity Center, previously known as Single Sign On (SSO).

Search for IAM Identity Centre’ in the console main search bar.



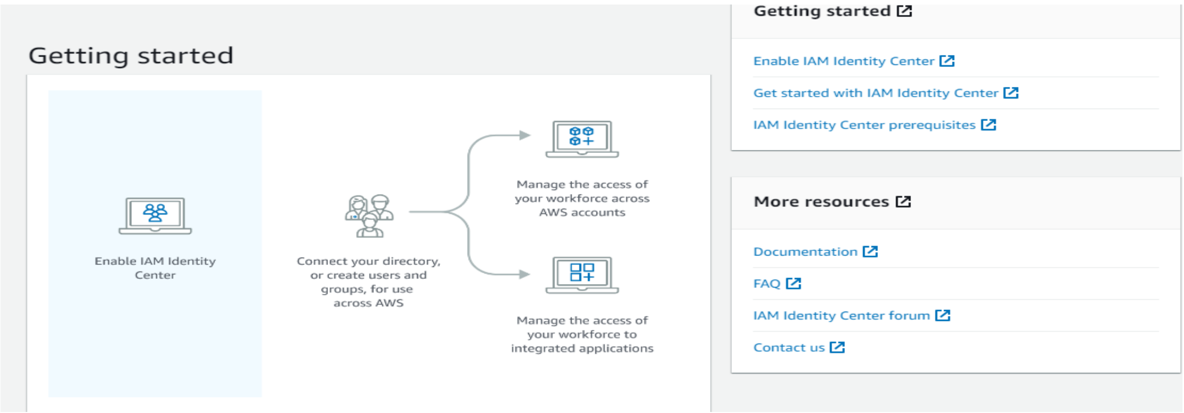
Are you providing console access to a person?

User type

● Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

This will take you to a new screen where you need to set up your Identity Centre.



Getting started

Enable IAM Identity Center
Connect your directory, or create users and groups, for use across AWS
Manage the access of your workforce across AWS accounts
Manage the access of your workforce to integrated applications

Getting started

- Enable IAM Identity Center
- Get started with IAM Identity Center
- IAM Identity Center prerequisites

More resources

- Documentation
- FAQ
- IAM Identity Center forum
- Contact us

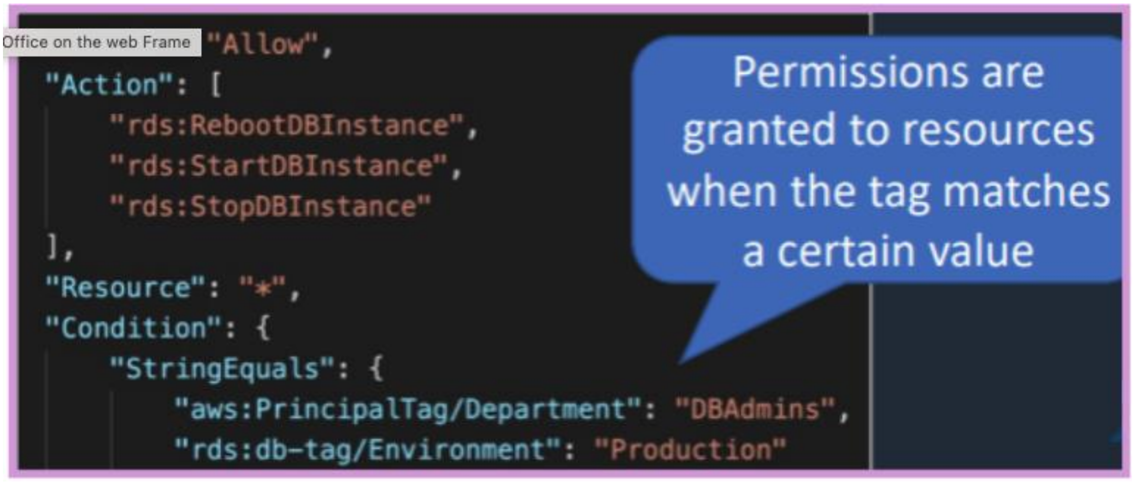
Varied users necessitate distinct access permissions to AWS services and resources, aligning with their specific requirements and designated roles. Furthermore, the utilization of the Identity Center serves to mitigate the risk associated with prolonged credentials use across workloads.

AWS Identity Center offers a comprehensive solution for orchestrating user identities, governing access permissions, and fortifying authentication mechanisms throughout your AWS framework. This enables you to establish a secure and adeptly controlled environment. Moreover, the provision to generate reports empowers you to scrutinize the permissions granted to each entity, whether internal or external, within your account.

In the realm of IAM Account strategies, we've observed that organizations, particularly small to medium-sized businesses (SMBs), can occasionally find themselves daunted by the intricate layers involved. A primary challenge lies in effectuating resource access management in a manner that is both secure, compliant, and scalable. Two prominent strategies are Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC). RBAC is apt for scenarios where access hinges on a user's role or responsibilities (e.g., Admin, Developer). On the other hand, ABAC is well-suited to regulating resource access based on user attributes like location, device type, IP, and other contextual factors. ABAC is optimal for scenarios contingent on the user's context.

For the implementation of an RBAC strategy, AWS offers an array of predefined policies tailored to various job functions, encompassing roles such as Administrator, Billing, Database Administrator, Data Scientist, Developer, Power User, Network Administrator, Security Auditor, Support User, System Administrator, and View-Only User. Of course, these policies can be fine-tuned in accordance with best practices to assign the least necessary permissions for the tasks at hand.

To illustrate an ABAC strategy, consider a hypothetical user named John who functions as a Database Administrator (DB Admin). However, you wish to restrict his activities within the Development department. In this scenario, tags can be employed. For instance, a Tag Value denoting "DB Admins" can be attributed to John, and a corresponding policy can be linked to him, confining his scope of action within defined parameters.



```
Office on the web Frame "Allow",
"Action": [
    "rds:RebootDBInstance",
    "rds:StartDBInstance",
    "rds:StopDBInstance"
],
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:PrincipalTag/Department": "DBAdmins",
        "rds:db-tag/Environment": "Production"
```

Permissions are granted to resources when the tag matches a certain value

As you can see, you have two strings in the policy which must that have to be met for access to be granted. One is the Tag/Department and the other one is the Environment. So according to this policy, John will not be able to carry out do any operations on resources tagged as the ‘Development’ department, even though he is an “DB Admin”.

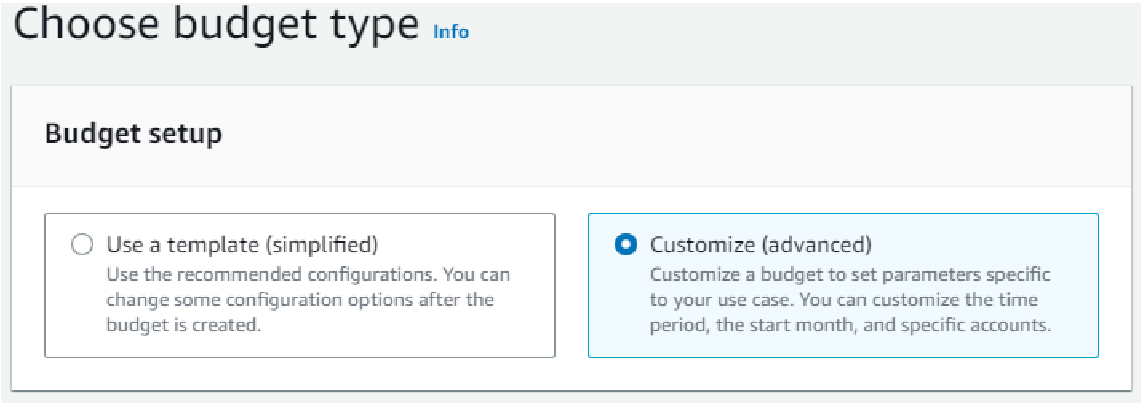
It is important to mention that all the above resources and controls are free within from AWS, so it makes sense to why not use a robust configuration on your account when all comes at no cost?

AWS Budgets

AWS Budgets provide a means to monitor utilization, expenses, and reservations of resources. Alerts can be configured for budgets, triggering notifications upon reaching predefined thresholds. Each budget can accommodate up to five alerts, and each alert can have up to ten subscribers. This functionality leverages the underlying Simple Notification Service (SNS). And the best part? All of this functionality is available free of charge!

Whether you have distinct workloads or even more finely defined resources within your AWS ecosystem, budgets can be tailored accordingly. For instance, you can establish budgets for overall workloads or delve deeper, creating budgets for specific resources like EC2, Lambda, RDS, EBS, S3, or even designated User Groups. The utilization of AWS Budgets extends beyond mere alerting; it furnishes insights into resource cost distribution within your budget allocation. This vantage point can facilitate strategic decision-making, guiding resource allocation, enhancing cost efficiency, and identifying avenues for cost optimization. A preview of a budget layout is illustrated below.

Now that we've grasped the rationale behind AWS Budgets, let's delve into the setup process. Within the AWS Billing and Cost Management Console, navigate to the 'Budgets' section located on the right-hand side. Upon selecting the 'Create Budget' button, two options will be presented: "Use a template" and "Advanced." Opting for the "Advanced" approach empowers you with the ability to finely configure your budgets, catering to specific needs with granular precision.



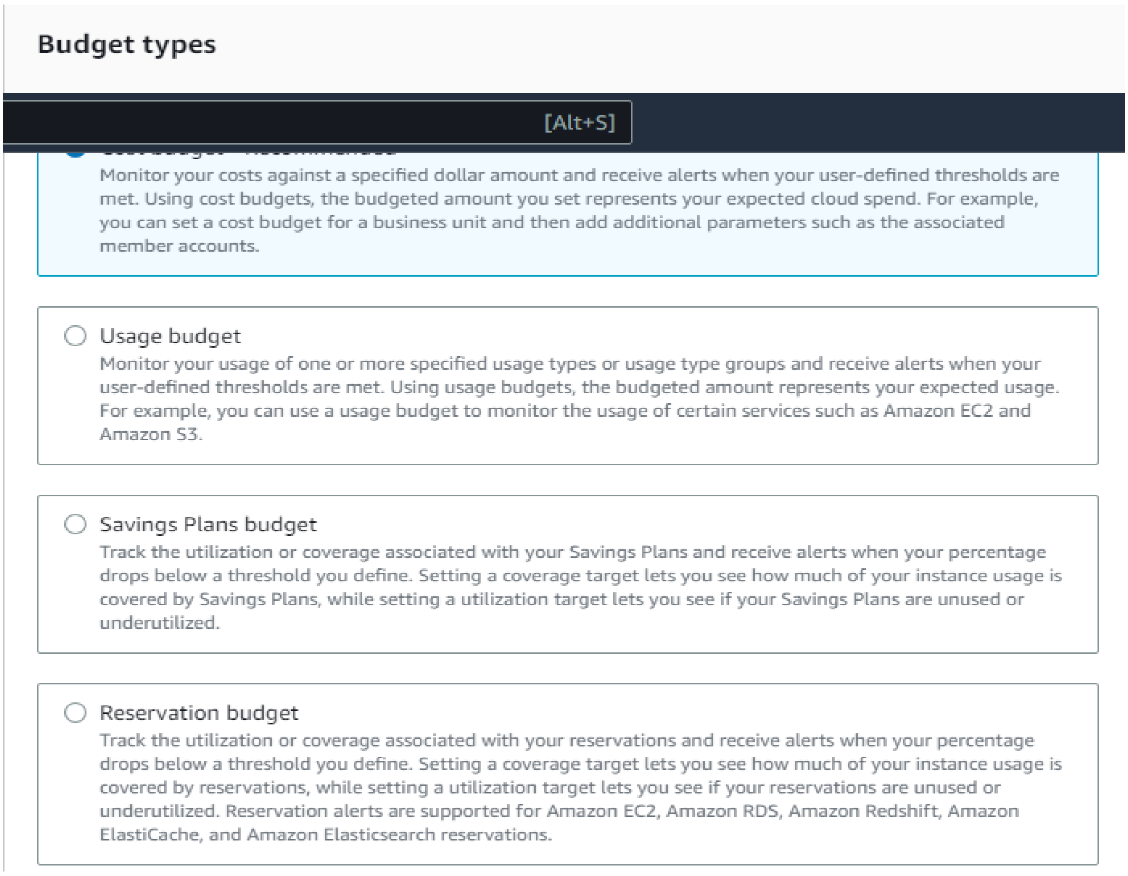
Choose budget type

Budget setup

○ Use a template (simplified)
Use the recommended configurations. You can change some configuration options after the budget is created.

● Customize (advanced)
Customize a budget to set parameters specific to your use case. You can customize the time period, the start month, and specific accounts.

You then have the option to choose from one out of the following four4 options: Cost Budget, Usage Budget, Saving Plans Budget or Reservation Budget. Without going into a lot of details, the last two options cover any compute resource that is associated with a Saving Plan or has been Reserved.



Budget types

Monitor your costs against a specified dollar amount and receive alerts when your user-defined thresholds are met. Using cost budgets, the budgeted amount you set represents your expected cloud spend. For example, you can set a cost budget for a business unit and then add additional parameters such as the associated member accounts.

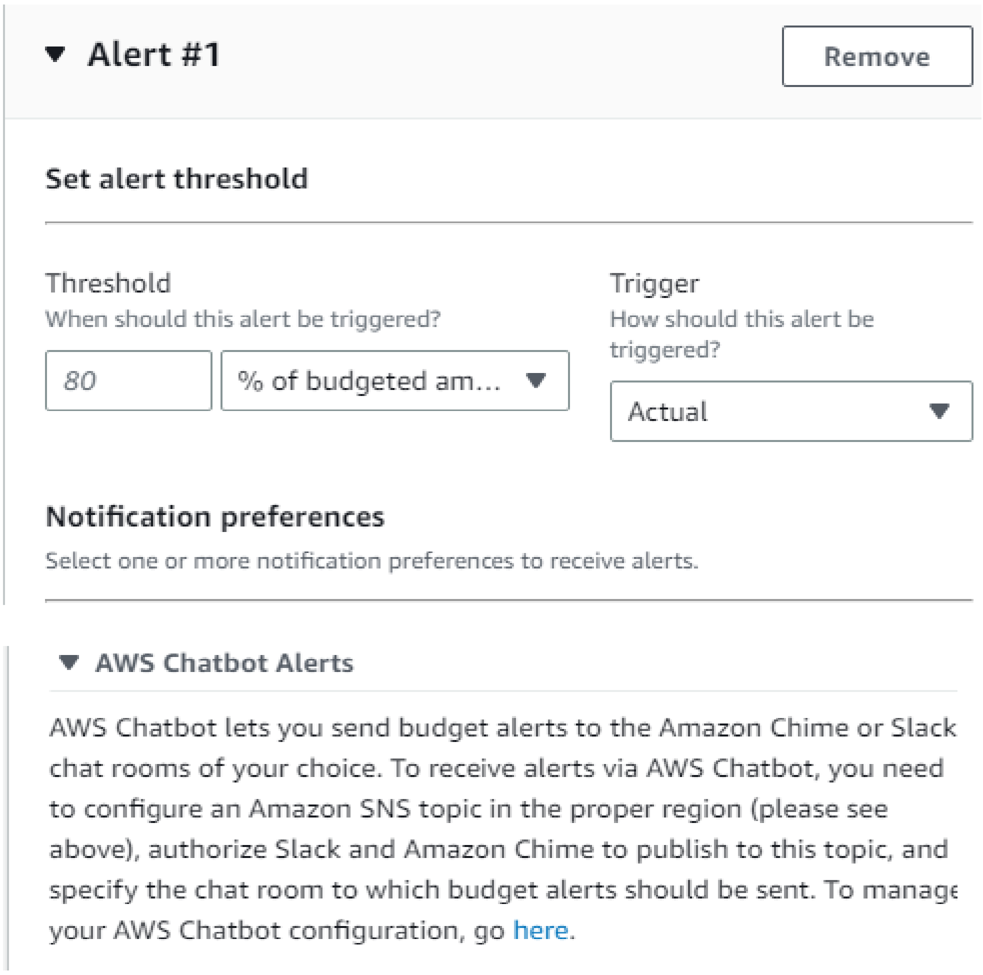
○ Usage budget
Monitor your usage of one or more specified usage types or usage type groups and receive alerts when your user-defined thresholds are met. Using usage budgets, the budgeted amount represents your expected usage. For example, you can use a usage budget to monitor the usage of certain services such as Amazon EC2 and Amazon S3.

○ Savings Plans budget
Track the utilization or coverage associated with your Savings Plans and receive alerts when your percentage drops below a threshold you define. Setting a coverage target lets you see how much of your instance usage is covered by Savings Plans, while setting a utilization target lets you see if your Savings Plans are unused or underutilized.

○ Reservation budget
Track the utilization or coverage associated with your reservations and receive alerts when your percentage drops below a threshold you define. Setting a coverage target lets you see how much of your instance usage is covered by reservations, while setting a utilization target lets you see if your reservations are unused or underutilized. Reservation alerts are supported for Amazon EC2, Amazon RDS, Amazon Redshift, Amazon ElastiCache, and Amazon Elasticsearch reservations.

If you intend to establish a predefined monetary value for triggering alerts, the "Cost Budget" option is the suitable choice. For those desiring a more intricate analysis of their resource consumption, the "Usage Budget" option should be selected. Following this, on the subsequent screen, you will have the opportunity to designate a time frame for the budget—be it Daily, Monthly, Quarterly, or Annually. You can set the budget to recur or have it expired after a certain duration, all while specifying the commencement month. You also have the option to either monitor all AWS Services in use or apply filters to target specific dimensions.

Upon proceeding to the subsequent page, you'll be tasked with configuring your alerts. This involves setting an alert threshold, such as 80% of the budget, and indicating the email addresses to which alert notifications should be dispatched. Additionally, AWS Chatbots Alerts can be leveraged to forward alerts to your Slack Channel, enhancing communication and visibility within your team.



▼ Alert #1

Set alert threshold

Threshold
When should this alert be triggered?

80 | % of budgeted am... ▼

Trigger
How should this alert be triggered?

Actual ▼

Notification preferences
Select one or more notification preferences to receive alerts.

▼ AWS Chatbot Alerts

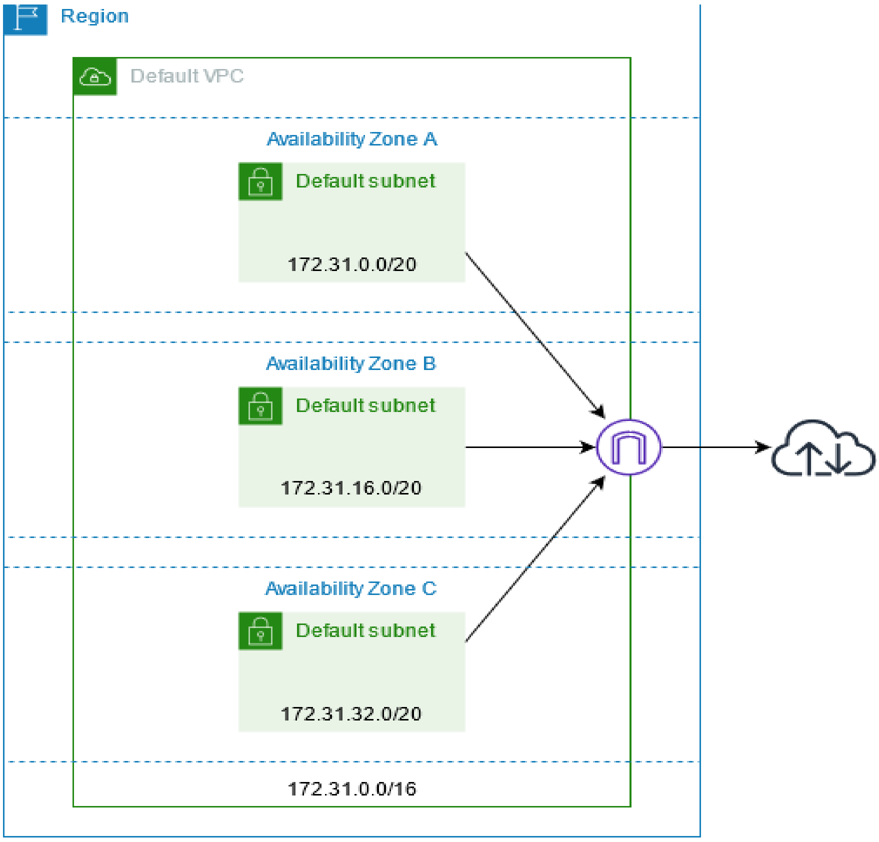
AWS Chatbot lets you send budget alerts to the Amazon Chime or Slack chat rooms of your choice. To receive alerts via AWS Chatbot, you need to configure an Amazon SNS topic in the proper region (please see above), authorize Slack and Amazon Chime to publish to this topic, and specify the chat room to which budget alerts should be sent. To manage your AWS Chatbot configuration, go here.

The last step is to All that will be left is just to review and confirm your Budget.

Delete unused VPCs, Subnets and Security Groups

When an account is initially established, an automatic creation of a Virtual Private Cloud (VPC) takes place in each AWS Region. This VPC facilitates the allocation of public IP addresses within its public subnets. Nevertheless, adhering to best practices entails removing these resources if they are not actively utilized. Retaining these deployments introduces the potential for inadvertent exposure of resources. Deleting a VPC results in the removal of its constituent elements, including subnets and security groups. Consequently, with such resources no longer present, the risk of unintentional utilization for subsequent workloads is mitigated.

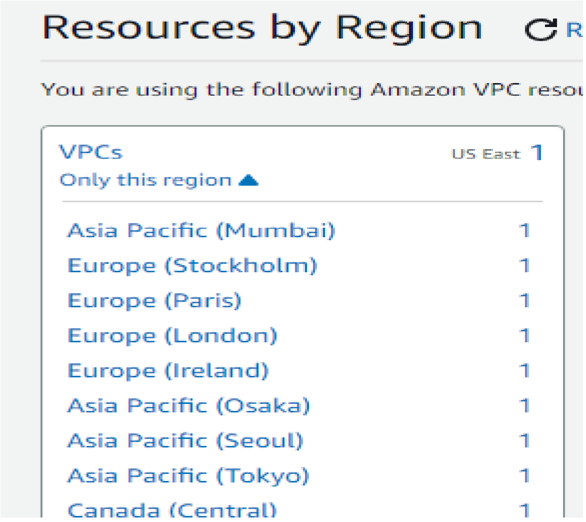
The subsequent diagram illustrates the fundamental components constituting a default VPC.



Understanding that if you choose to eliminate a default VPC, it's crucial to note that while you can establish a new one, the restoration of a previously deleted default VPC is not feasible.

Adhering to best practices within the AWS environment involves the removal of any unused resources. This approach aligns with two key aspects of the Well-Architected Framework: 'Security' and 'Sustainability'. Alongside these, the framework also encompasses Operational Excellence, Cost Optimization, Performance Efficiency, and Reliability. This emphasis on security and sustainability resonates not only from a security perspective but also from the viewpoint of sustainability, a core tenet upon which AWS itself is constructed.

Upon navigating to your VPC Management Console, you will observe that each region features at least one VPC, resembling the depiction in the image below.

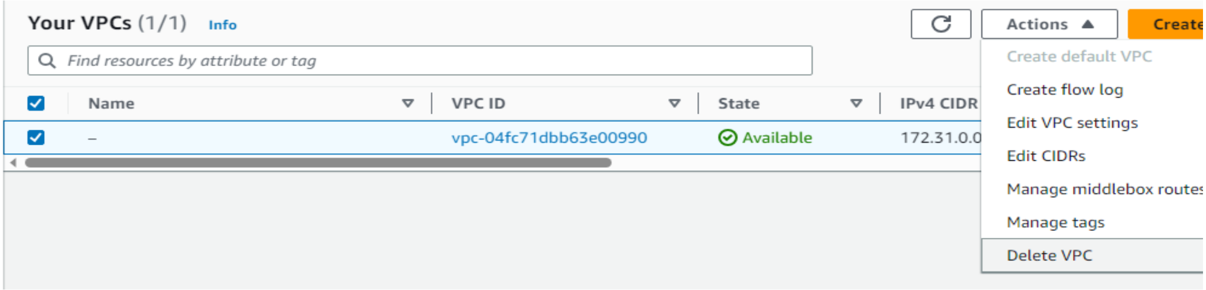


Resources by Region R

You are using the following Amazon VPC resou

VPCs US East 1
Only this region

Asia Pacific (Mumbai)	1
Europe (Stockholm)	1
Europe (Paris)	1
Europe (London)	1
Europe (Ireland)	1
Asia Pacific (Osaka)	1
Asia Pacific (Seoul)	1
Asia Pacific (Tokyo)	1
Canada (Central)	1



Your VPCs (1/1) Info

☑	Name	VPC ID	State	IPv4 CIDR
☑	–	vpc-04fc71dbb63e00990	Available	172.31.0.0

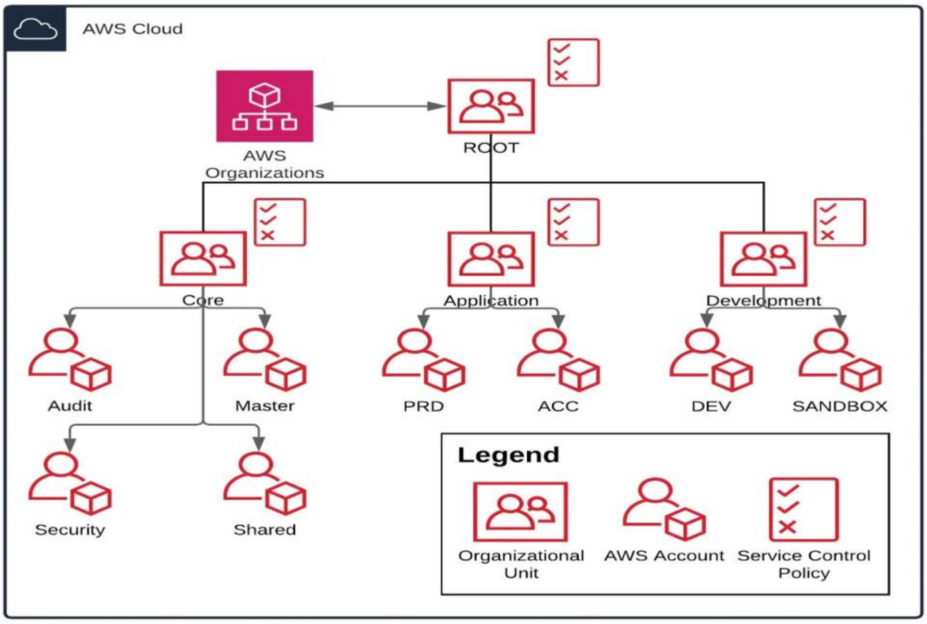
AWS Organisations

One of the primary mechanisms by which AWS ensures the security of your applications and data revolves around the concept of the AWS account. An AWS account establishes inherent security, access, and billing boundaries for your AWS resources, fostering resource autonomy and isolation. While your journey might commence with a single account, AWS strongly recommends the establishment of multiple accounts as your workloads expand in both scale and intricacy. The adoption of a multi-account environment aligns with AWS's best practices, delivering several advantages including:

- Streamlined billing management
- Adaptable security controls
- Seamless alignment with evolving business processes

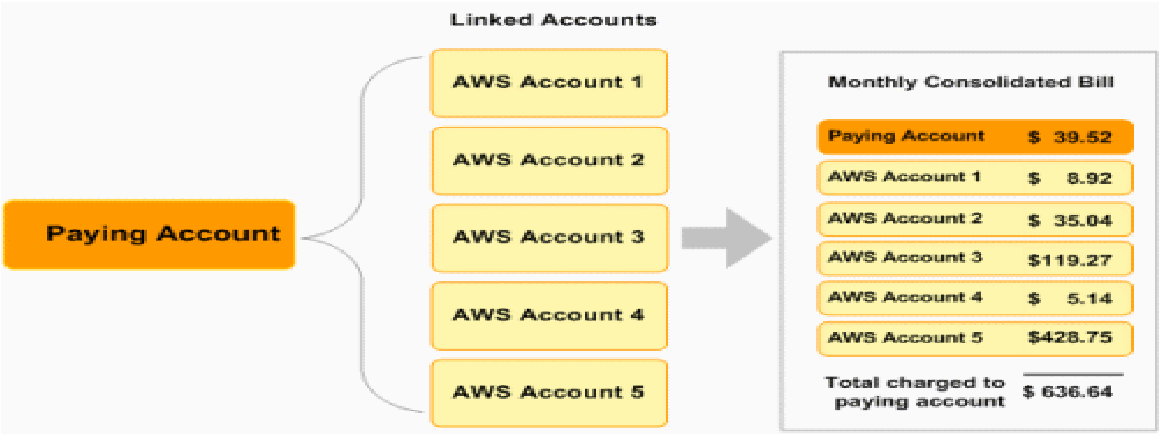
Many of these benefits come at no additional cost for AWS services, for instance, features like Service Control Policies (SCPs) and Single Sign-On (SSO) are available without incurring any charges.

AWS Organizations serves as a service that empowers you to centrally oversee and govern numerous AWS accounts. It introduces a hierarchical structure for organizing and administering these accounts, simplifying the implementation of security and governance protocols, especially valuable for SMBs. Here's a breakdown of how you can harness AWS Organizations to enhance the security of SMB accounts:

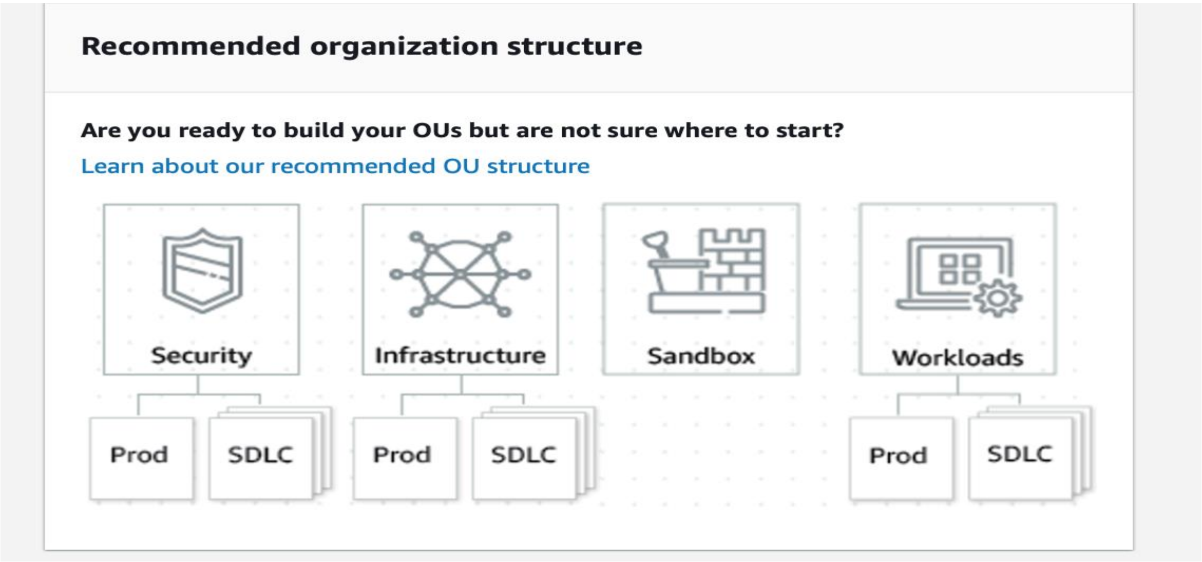


- Consolidated Billing: AWS Organizations offers the capability to centralize billing for all your AWS accounts under a single master account, often referred to as the Root Account. This streamlines financial oversight and grants a clear, comprehensive perspective of expenditures.

You receive a consolidated bill that can be examined in detail, allowing you to break down or segment costs into various resource-specific categories. This level of granularity is customizable to your preferences. By assigning tags to resources and services within your account, including all resources initiated within the account, you gain the ability to scrutinize spending based on these tags. This enables you to monitor individual departments, expenses, accounts, projects, or development initiatives, tailoring the analysis to your specific needs. Moreover, through the application of Service Control Policies (SCPs), you can enforce the launch of new resources only if they adhere to a designated tagging policy, one that you design and enforce.



Paying Account	$ 39.52
AWS Account 1	$ 8.92
AWS Account 2	$ 35.04
AWS Account 3	$119.27
AWS Account 4	$ 5.14
AWS Account 5	$428.75
Total charged to paying account	$ 636.64

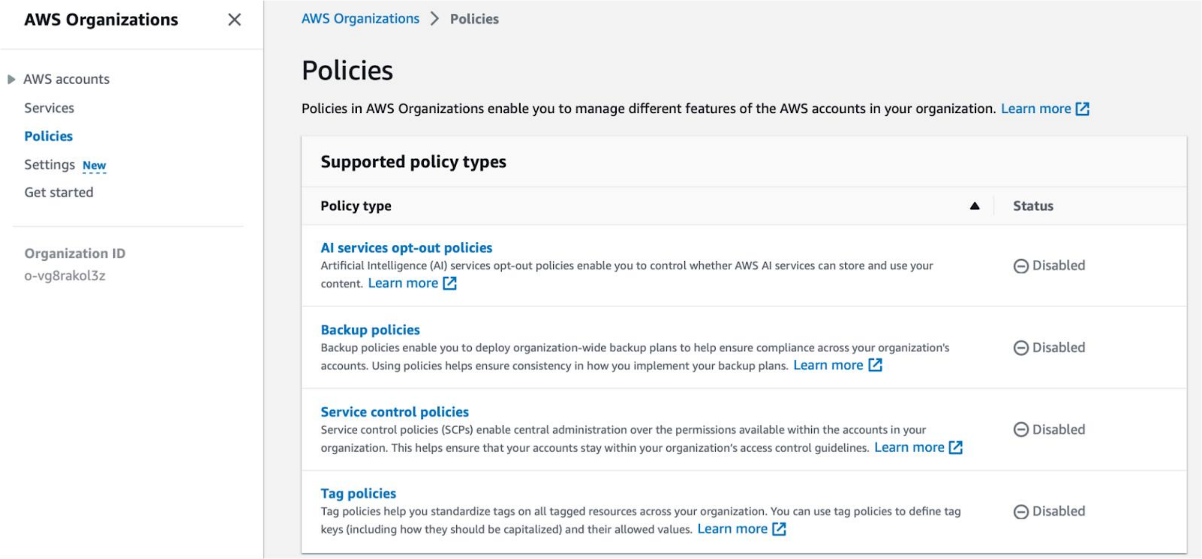


Recommended organization structure

Are you ready to build your OUs but are not sure where to start?

Learn about our recommended OU structure

- Implement a Multi-Account Strategy: With AWS Organizations, you can create multiple AWS accounts for different purposes, such as development, production, testing, or specific projects. This helps isolate resources and mitigate potential risks.
- Apply Service Control Policies (SCPs): SCPs are security policies which that you can apply at the organizational level to control the services and actions available to the member accounts. By defining SCPs, you can enforce security best practices, limit access to certain AWS services, and prevent accidental or unauthorized changes.



Organization ID
o-vg8rakol3z

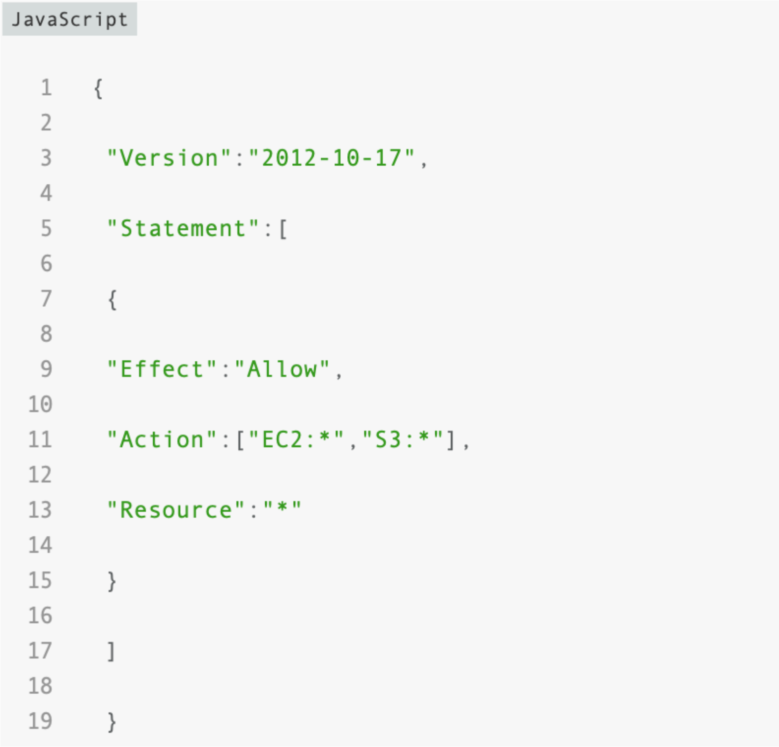
Policies

Policies in AWS Organizations enable you to manage different features of the AWS accounts in your organization. Learn more

Supported policy types

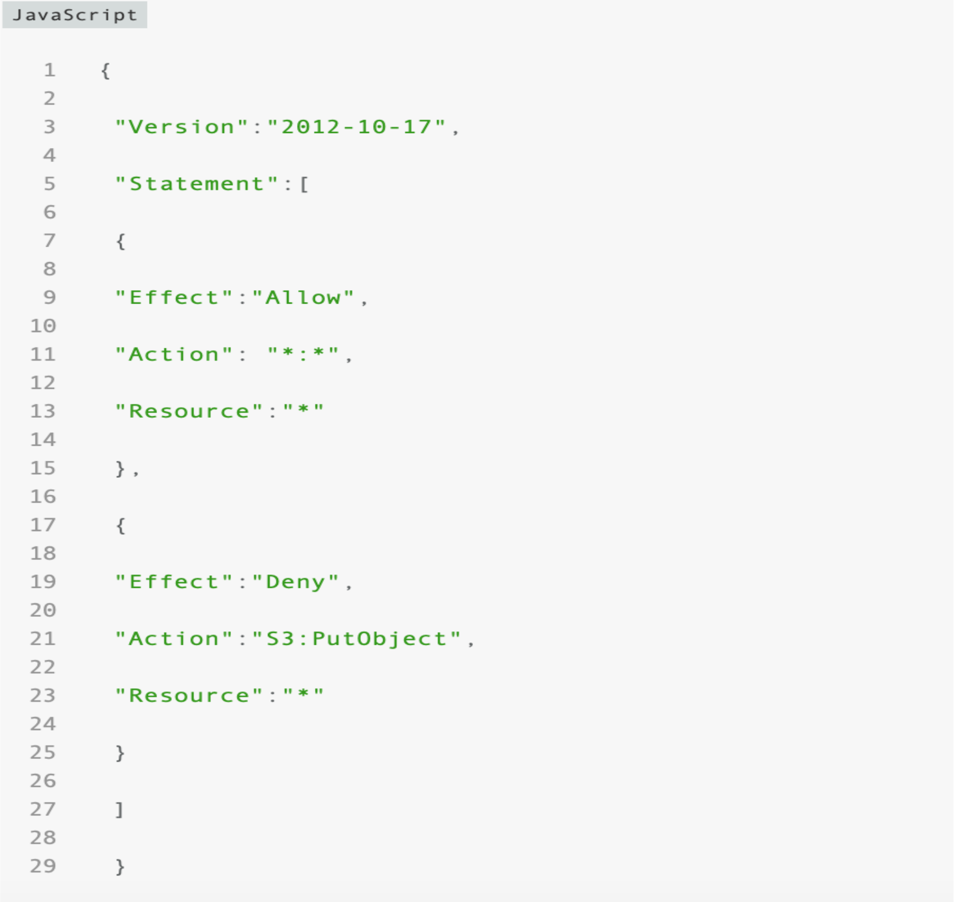
Policy type ▲	Status
AI services opt-out policies Artificial Intelligence (AI) services opt-out policies enable you to control whether AWS AI services can store and use your content. Learn more	⊖ Disabled
Backup policies Backup policies enable you to deploy organization-wide backup plans to help ensure compliance across your organization's accounts. Using policies helps ensure consistency in how you implement your backup plans. Learn more	⊖ Disabled
Service control policies Service control policies (SCPs) enable central administration over the permissions available within the accounts in your organization. This helps ensure that your accounts stay within your organization's access control guidelines. Learn more	⊖ Disabled
Tag policies Tag policies help you standardize tags on all tagged resources across your organization. You can use tag policies to define tag keys (including how they should be capitalized) and their allowed values. Learn more	⊖ Disabled

All SCPs (Service Control Policies) are written in JSON format, as shown in below examples.



```
{

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

 "Statement":[

 {

 "Effect":"Allow",

 "Action":["EC2:*","S3:*"],

 "Resource":"*"

 }

 ]

 }
```

This is an example of for an ‘Allow Strategy’ policy., Everything is implicitly denied unless you specifically allow it as in this policy example,

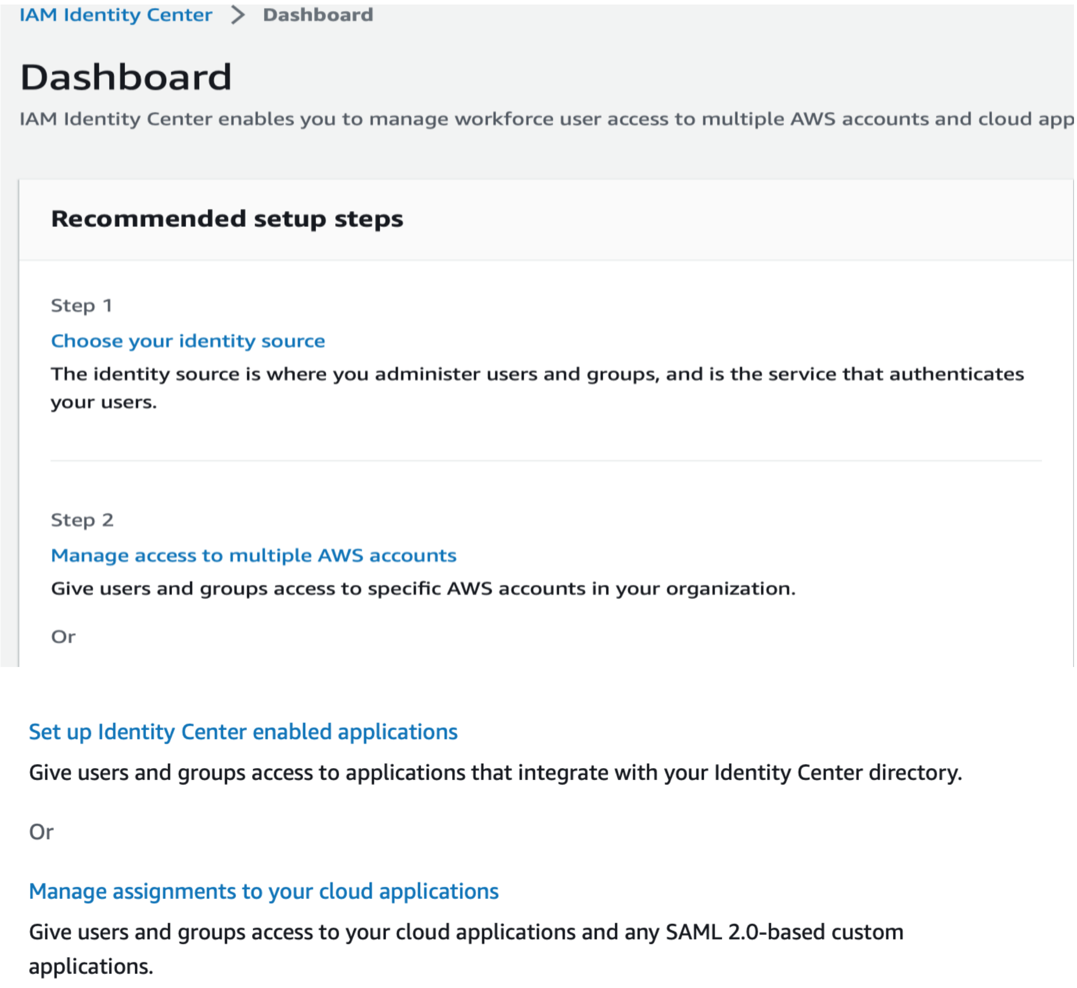


JavaScript

```
{

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

 "Statement":[

 {

 "Effect":"Allow",

 "Action":  "*:*",

 "Resource":"*"

 },

 {

 "Effect":"Deny",

 "Action":"S3:PutObject",

 "Resource":"*"

 }

 ]

 }
```

This example represents a ‘Deny Strategy’ Policy, where everything is implicitly allowed, and you explicitly deny specific actions that you do not want to permit within this account. This approach employs explicit denial, which provides stronger control; however, it requires significant effort to manage when dealing with numerous services that need to have deny rules added. As a result, Consequently, the policy can become large and cumbersome intricate.

- Enable Cross-Account Access: AWS Organizations allows you to grant cross-account access to resources. You can set up IAM roles in the member accounts and establish trust relationships with the master account, allowing centralized management while maintaining separation of permissions.



IAM Identity Center > Dashboard

Dashboard

IAM Identity Center enables you to manage workforce user access to multiple AWS accounts and cloud app

Recommended setup steps

Step 1

Choose your identity source

The identity source is where you administer users and groups, and is the service that authenticates your users.

Step 2

Manage access to multiple AWS accounts

Give users and groups access to specific AWS accounts in your organization.

Or

Set up Identity Center enabled applications

Give users and groups access to applications that integrate with your Identity Center directory.

Or

Manage assignments to your cloud applications

Give users and groups access to your cloud applications and any SAML 2.0-based custom applications.

- Streamline Security Controls: With AWS Organizations, you can apply security settings and configurations consistently across multiple accounts. This includes enabling AWS CloudTrail for centralized logging, configuring AWS Config rules for compliance checks, and implementing AWS Shield and AWS WAF for DDoS protection and web application firewall capabilities.
- Simplify Identity and Access Management: AWS Organizations integrates with AWS Identity and Access Management (IAM), allowing you to manage user access and permissions centrally. You can create IAM roles, groups, and policies in the master account and grant appropriate access to member accounts.
- Implement Account Vending Machine (AVM): AVM is an AWS Solutions Implementation that simplifies the process of creating new AWS accounts within your organization. It provides a self-service portal for users to request new accounts while enforcing standard security configurations.

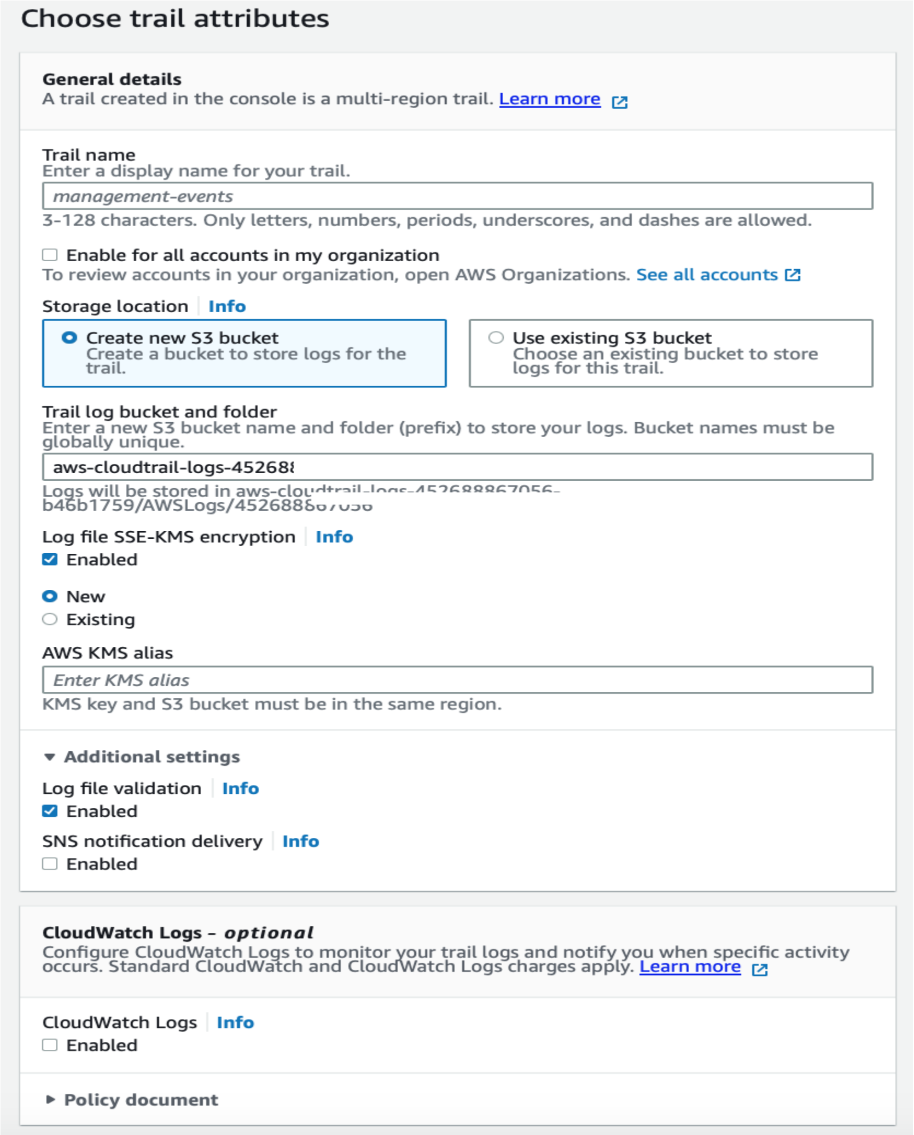
By leveraging AWS Organizations, SMBs can effectively manage and secure their AWS accounts, enforce security policies, control access, and streamline operations across their organization. It offers a scalable and flexible solution for maintaining a secure and well-governed AWS environment.

AWS CloudTrail and CloudWatch

Securing your SMB’s AWS cloud accounts is crucial for protecting sensitive data and mitigating potential risks such as data leaks or unwanted account access. AWS provides services like AWS CloudTrail and AWS CloudWatch, which can be utilized to enhance the security posture of your accounts. Here's a guide on how to leverage CloudTrail and CloudWatch for SMBs:

Taking into consideration that both services are provided at no cost, it is important to note that subscribing to the CloudWatch service and enabling CloudTrail or creating your own trail are required. It is worth mentioning that as the log files accumulate, storage charges will be incurred based on the file size. However, you can effectively manage storage using lifecycle policies, allowing you to archive logs after a certain period to deep archive or delete them if they are no longer needed. Additionally, CloudWatch generates reports every 5 minutes as the standard frequency, but if you require more precise and detailed reports and information monitoring, enabling detailed monitoring with a frequency of every minute is available at an additional cost.

- Enable AWS CloudTrail: AWS CloudTrail captures detailed logs of all API activities performed within your AWS account. Enable CloudTrail in all relevant regions and configure it to store logs in an S3 bucket. This ensures that any changes made to resources or actions taken by users are recorded, allowing you to monitor and investigate any suspicious or unauthorized activities.
- Set Up CloudTrail Log File Integrity Validation: Enable CloudTrail Log File Integrity Validation, which uses cryptographic hashes to validate the integrity of your log files. This ensures that the log files are not tampered with or modified, providing an extra layer of assurance for the audit trail.
- Configure CloudTrail Event Notifications: Use CloudTrail to set up event notifications for specific API activities or resource changes that are critical to your SMB's security. By doing so, you can receive real-time notifications via Amazon Simple Notification Service (SNS) or trigger automated actions through AWS Lambda functions, enabling proactive response to security events.
- Leverage AWS CloudWatch Logs: AWS CloudWatch allows you to collect, monitor, and store logs from various AWS services, including CloudTrail. Create CloudWatch Log Groups and Log Streams to centralize CloudTrail logs and retain them for an appropriate period. This ensures that you have a centralized repository of logs for analysis, compliance, and security investigations.



Choose trail attributes

General details
A trail created in the console is a multi-region trail. Learn more

Trail name
Enter a display name for your trail.
management-events
3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

□ Enable for all accounts in my organization
To review accounts in your organization, open AWS Organizations. See all accounts

Storage location | Info

● Create new S3 bucket
Create a bucket to store logs for the trail.

○ Use existing S3 bucket
Choose an existing bucket to store logs for this trail.

Trail log bucket and folder
Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.
aws-cloudtrail-logs-45268
Logs will be stored in aws-cloudtrail-logs-452688867056-b46b1759/AWSLogs/452688867056

Log file SSE-KMS encryption | Info
☑ Enabled

● New
○ Existing

AWS KMS alias
Enter KMS alias
KMS key and S3 bucket must be in the same region.

▼ Additional settings

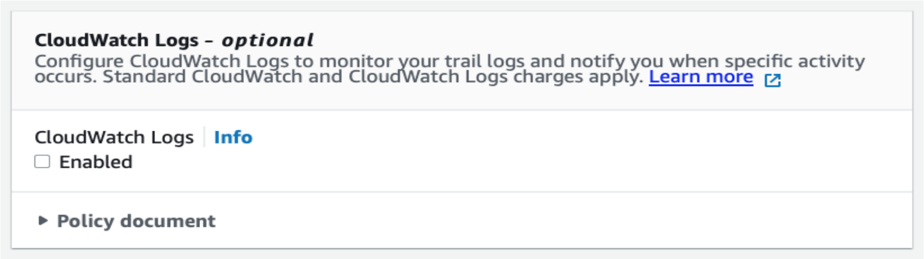
Log file validation | Info
☑ Enabled

SNS notification delivery | Info
□ Enabled

CloudWatch Logs – *optional*
Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. Learn more

CloudWatch Logs | Info
□ Enabled

▸ Policy document

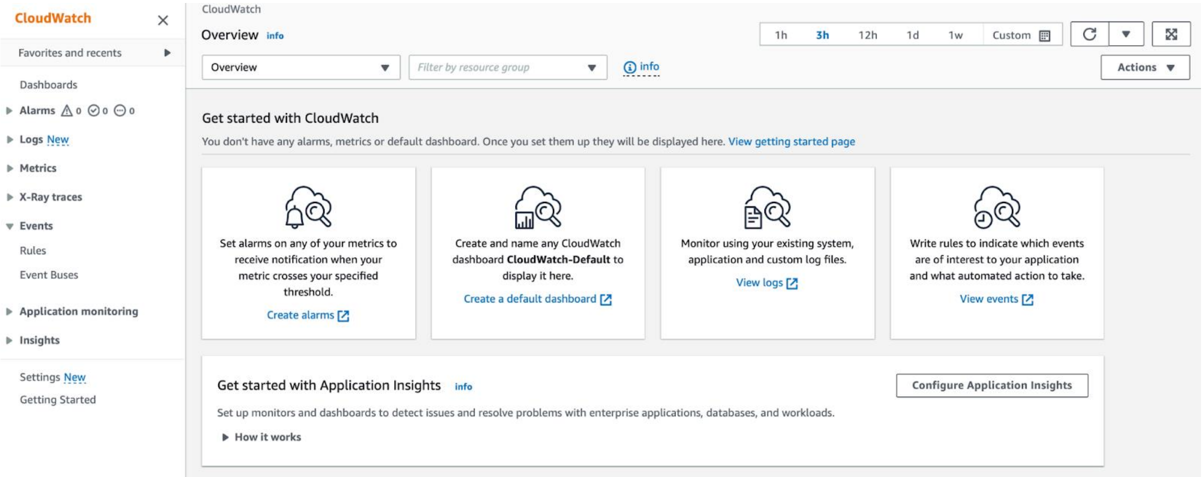


CloudWatch Logs – *optional*
Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. Learn more

CloudWatch Logs | Info
□ Enabled

▸ Policy document

Implement CloudWatch Alarms: Utilize CloudWatch Alarms to set up threshold-based notifications for specific security-related events. For example, you can create an alarm to trigger when unauthorized API calls are detected or when resource usage exceeds defined limits. This helps in identifying potential security breaches or anomalies and allows for timely response.



CloudWatch

Favorites and recents

Dashboards

Alarms ⚠ 0 ⊘ 0 ⊖ 0

Logs New

Metrics

X-Ray traces

Events

Rules

Event Buses

Application monitoring

Insights

Settings New

Getting Started

CloudWatch

Overview info

1h 3h 12h 1d 1w Custom

Overview

Filter by resource group

ⓘ info

Actions

Get started with CloudWatch

You don't have any alarms, metrics or default dashboard. Once you set them up they will be displayed here. View getting started page

Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold.

Create alarms

Create and name any CloudWatch dashboard **CloudWatch-Default** to display it here.

Create a default dashboard

Monitor using your existing system, application and custom log files.

View logs

Write rules to indicate which events are of interest to your application and what automated action to take.

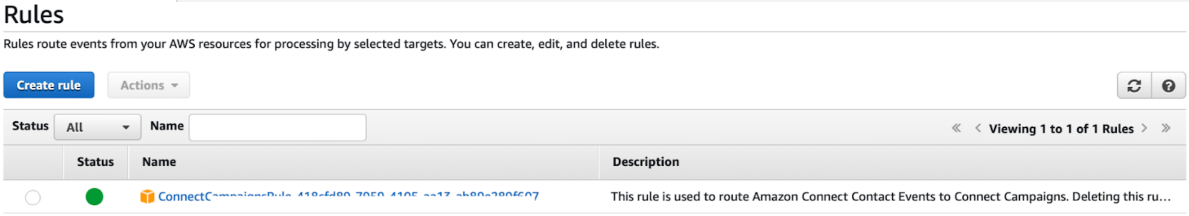
View events

Get started with Application Insights info

Configure Application Insights

Set up monitors and dashboards to detect issues and resolve problems with enterprise applications, databases, and workloads.

▶ How it works



Rules

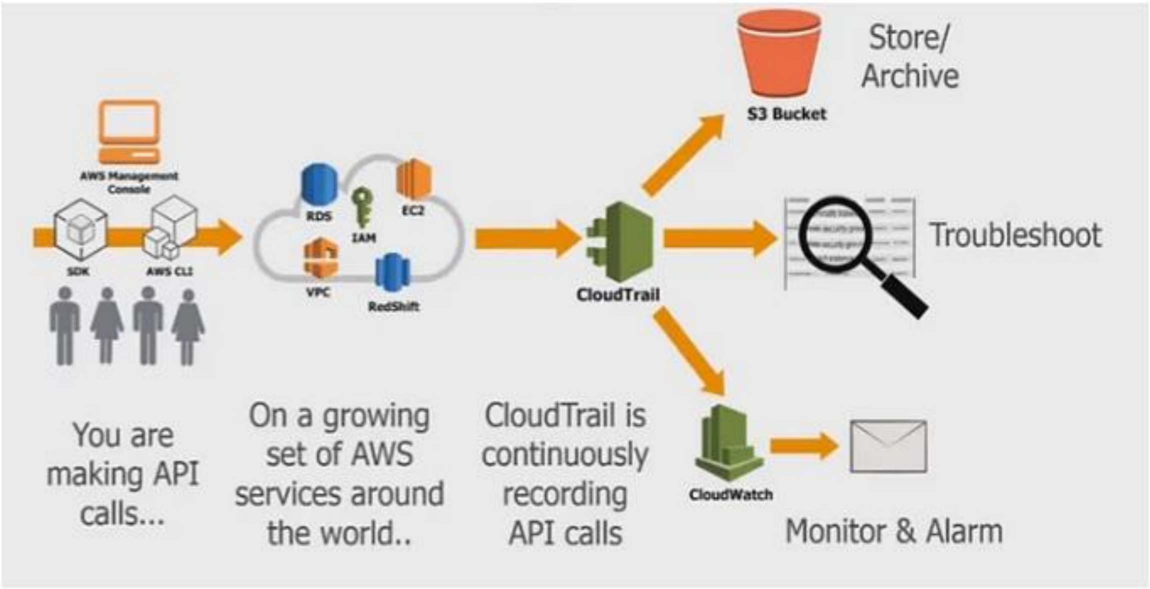
Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Create rule   Actions

Status All   Name   « ‹ Viewing 1 to 1 of 1 Rules › »

	Status	Name	Description
○	●	ConnectCampaignsRule-418cfd80-7059-4105-aa17-ab80a280f607	This rule is used to route Amazon Connect Contact Events to Connect Campaigns. Deleting this ru...

Enable CloudWatch Event Rules: CloudWatch Event Rules enable you to automate security-related actions in response to specific events or patterns. For instance, you can create a rule to automatically terminate an EC2 instance if it is detected that it has been compromised or to trigger a Lambda function to disable access keys associated with suspicious API activities.


AWS Management Console
SDK
AWS CLI
You are making API calls...
RDS
EC2
IAM
VPC
RedShift
On a growing set of AWS services around the world..
CloudTrail
CloudTrail is continuously recording API calls
S3 Bucket
Store/ Archive
Troubleshoot
CloudWatch
Monitor & Alarm

AWS CloudTrail

By effectively configuring and utilizing AWS CloudTrail and AWS CloudWatch, SMBs can establish a robust security framework within their AWS cloud accounts. These services provide comprehensive visibility into account activities, allow for real-time monitoring and detection of security incidents, and enable proactive response to potential threats or unauthorized actions. Regularly review and analyze the logs and metrics provided by these services to identify security gaps, improve incident response, and ensure a secure AWS environment for your SMB.

Security Scanning Assessments tools

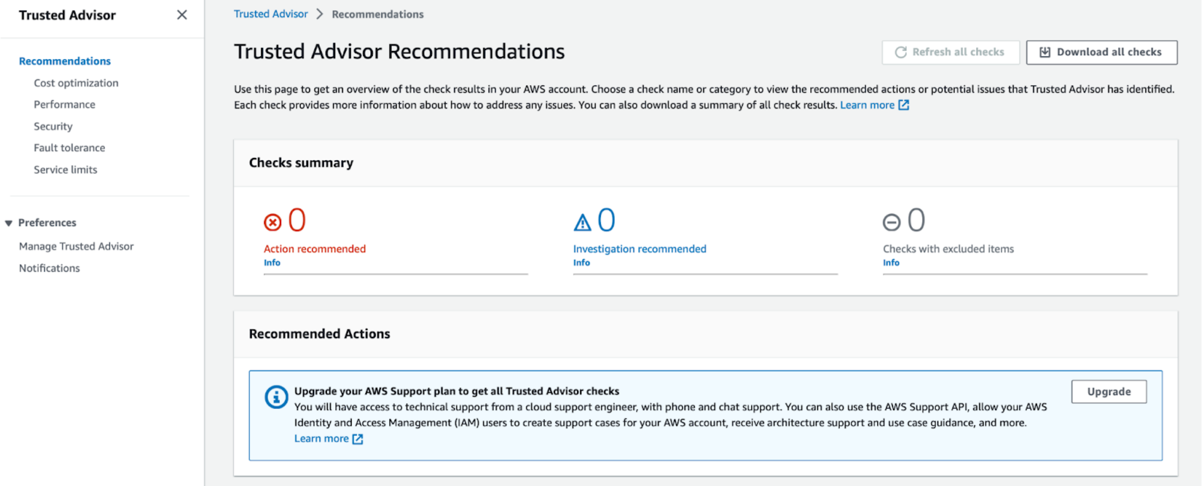
Step 1: Select a Security Scanning Assessment Tool

Numerous security scanning assessment tools are available for AWS, allowing you to select the one that best suits your requirements and aligns with your security objectives. Trusted Advisor and AWS Inspector are popular options in this regard. Both services offer free features, with Trusted Advisor providing 7 basic security checks for your account at no cost. However, additional checks are available through higher paid support plans like Business or Enterprise. On the other hand, AWS Inspector itself is free, but charges are incurred based on the number of instances used for scanning and the frequency of scans. You can find detailed pricing information for AWS Inspector at this link: https://aws.amazon.com/inspector/pricing/. It's important to note that there are other services, such as Prowler, that can perform security scans on your behalf. Prowler covers over 200 security best practices across all your AWS regions and most AWS services, including IAM, Logging, Monitoring, Networking, CIS lvl 1 & 2, GDPR, HIPAA, Trust boundaries, Secrets, PCI-DSS, ISO 27001, and SOC2 checks. This service can be installed on one of your EC2 instances or even on your on-premises machine for a hybrid model, and it is available for free."

AWS Trusted Advisor is a service that draws upon best practices and inspects your AWS environment making recommendations for saving money, improving system performance, or closing security gaps. You can configure Trusted Advisor notifications to receive weekly emails about any changes. You can also subscribe to Business and Enterprise-level support to access the full suite of Trusted Advisor best-practice checks.

Many AWS services provide built-in access control audit trails and logs. You can enable Amazon VPC Flow Logs to capture information about the IP traffic going to and from network interfaces in your VPC. VPC flow logs can help you with number of tasks. For example, you can troubleshoot why specific traffic is not reaching an instance, which in turn helps you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance.

AWS Security Hub gives you a single pane of glass view of your high-priority security alerts and compliance status across AWS accounts. It provides you a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie, as well as from AWS Partner solutions.



Trusted Advisor Recommendations

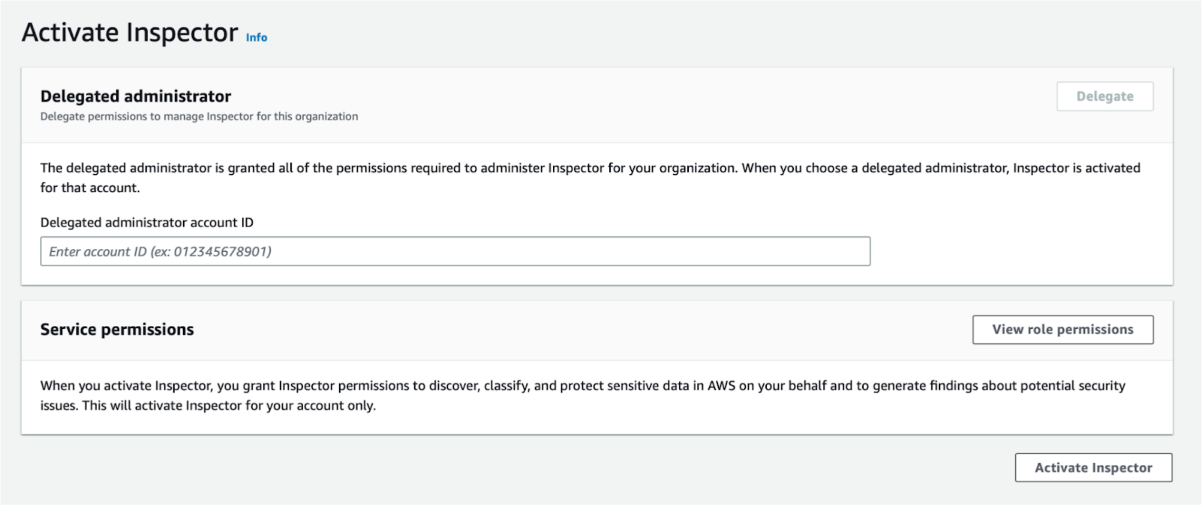
Use this page to get an overview of the check results in your AWS account. Choose a check name or category to view the recommended actions or potential issues that Trusted Advisor has identified. Each check provides more information about how to address any issues. You can also download a summary of all check results. Learn more

Checks summary

⊗ 0 Action recommended Info	⚠ 0 Investigation recommended Info	⊖ 0 Checks with excluded items Info

Recommended Actions

Upgrade your AWS Support plan to get all Trusted Advisor checks
You will have access to technical support from a cloud support engineer, with phone and chat support. You can also use the AWS Support API, allow your AWS Identity and Access Management (IAM) users to create support cases for your AWS account, receive architecture support and use case guidance, and more.
Learn more



Activate Inspector Info

Delegated administrator
Delegate permissions to manage Inspector for this organization

The delegated administrator is granted all of the permissions required to administer Inspector for your organization. When you choose a delegated administrator, Inspector is activated for that account.

Delegated administrator account ID
Enter account ID (ex: 012345678901)

Service permissions

When you activate Inspector, you grant Inspector permissions to discover, classify, and protect sensitive data in AWS on your behalf and to generate findings about potential security issues. This will activate Inspector for your account only.

Step 2: Launch an EC2 Instance

- To deploy the security scanning assessment tool, you will need to launch an Amazon EC2 instance. Follow these steps:
- Sign in to the AWS Management Console.
- Open the Amazon EC2 console.
- Click on "Launch Instance."
- Choose an Amazon Machine Image (AMI) that is compatible with your chosen security scanning tool.
- Select an instance type that meets your needs.
- Configure the instance details, including network settings and storage.

- Add any necessary tags and specify a security group to allow inbound access for scanning purposes.
- Review your settings and launch the instance.

Step 3: Connect to the EC2 Instance

- After launching the EC2 instance, you need to connect to it via SSH or RDP, depending on the operating system used by the instance. Here is how:
- Obtain the key pair you selected during instance launch (if using SSH).
- Use an SSH client (e.g., PuTTY) or an RDP client (e.g., Microsoft Remote Desktop) to connect to the instance using the appropriate credentials and IP address.

Step 4: Install and Configure the Security Scanning Assessment Tool

Once connected to the EC2 instance, you can proceed to install and configure the security scanning assessment tool of your choice. The specific steps will depend on the tool you selected. The process involves downloading the tool, installing any required dependencies, and configuring it according to your needs. Refer to the tool's documentation for detailed instructions.

Step 5: Configure Access Permissions and Credentials

- To ensure the security scanning tool can access and scan your AWS resources, you will need to configure appropriate access permissions and credentials. Follow these steps:
- Create an IAM role or user with the necessary permissions to access the AWS resources you want to scan. The required permissions will vary based on the scanning tool's requirements. At a minimum, you will need permissions to read and scan AWS resources.
- Obtain the access key and secret key associated with the IAM role or user.
- Configure the scanning tool to use the obtained access and secret keys.

Step 6: Configure Scan Parameters

Next, you will need to configure the scan parameters based on your requirements. This includes specifying the target resources, scan frequency, scanning options, and any custom configurations. Refer to the documentation of your chosen scanning tool for guidance on configuring scan parameters.

Step 7: Run Security Scans

Once you have configured the scan parameters, you can initiate the security scans. Depending on the tool, you may be able to start scans manually or schedule them to run at specific intervals. Monitor the scan progress and review the generated reports for any security vulnerabilities or issues.

Step 8: Analyse and Remediate Findings

- After the scans are complete, carefully analyse the scan reports generated by the security scanning tool. Identify any security vulnerabilities or weaknesses in your AWS resources and develop a plan to remediate them. Address each finding according to best practices and industry standards to enhance the security posture of your AWS environment.

Remember to regularly update the security scanning tool and repeat the scanning process periodically to stay proactive in identifying and addressing security risks in your AWS infrastructure.

Please note that this guide provides a general overview, and the exact steps may vary depending on the specific security scanning tool you choose to deploy on AWS. Always refer to the tool's documentation for detailed instructions and best practices.

Encrypt EBS Volumes and RDS Databases

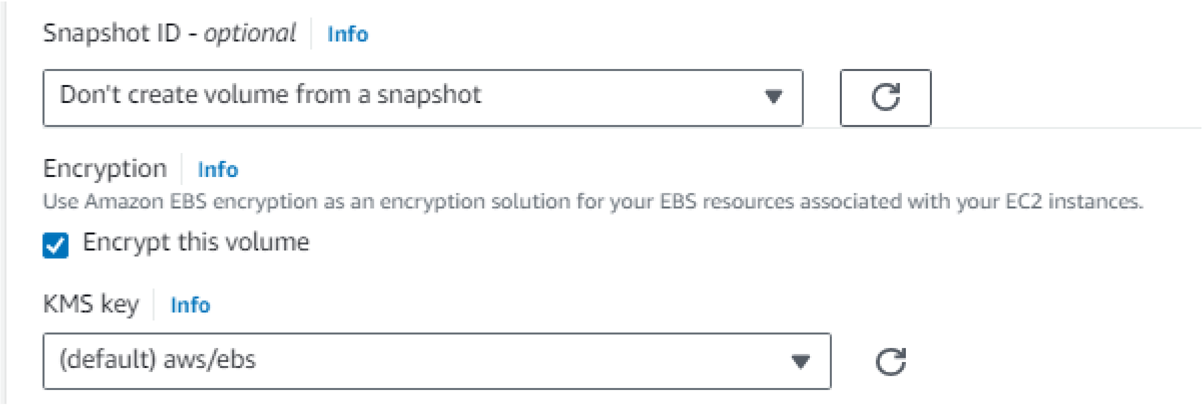
The Encryption of the EBS volumes is free and does not cost you anything in billing, but it has a very slight impact on the performance.

- Encrypting Elastic Block Store (EBS) volumes is a critical security measure to protect sensitive data stored in your AWS environment. This guide will walk you through the steps to encrypt EBS volumes effectively.
- Understand EBS Encryption:
- AWS offers built-in encryption capabilities for EBS volumes.
- EBS encryption uses AWS Key Management Service (KMS) to manage encryption keys securely.
- When you enable encryption for an EBS volume, the data is automatically encrypted before being written to disk and decrypted when read back.
- Choose the Encryption Option:
- Decide on the encryption option based on your requirements:

a. Encrypt New Volumes: Encrypt volumes when creating them.
b. Encrypt Existing Volumes: Encrypt volumes that are already in use.

Encrypting New EBS Volumes:

- Launch an EC2 instance or select an existing instance to attach the new EBS volume.
- Access the AWS Management Console or use AWS CLI/API to create a new EBS volume.
- During creation, choose the "Encrypt this volume" option and select a KMS key for encryption.
- Complete the creation process and attach the encrypted EBS volume to your instance.



Snapshot ID - *optional* Info

Don't create volume from a snapshot

Encryption Info

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

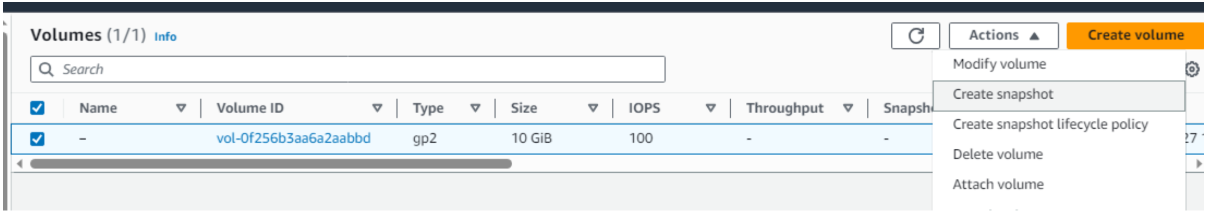
☑ Encrypt this volume

KMS key Info

(default) aws/ebs

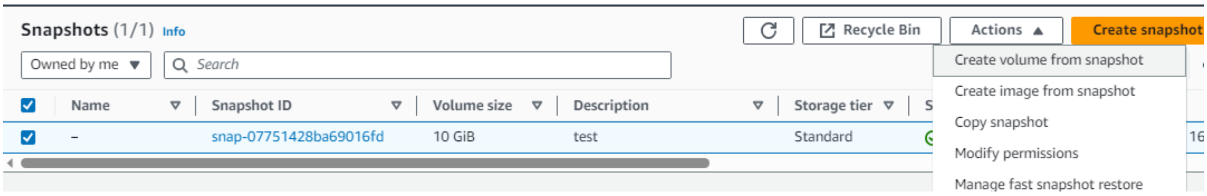
Encrypting Existing EBS Volumes:

- Identify the EBS volume(s) you want to encrypt.
- Create a snapshot of the existing unencrypted volume(s).
- From the snapshot, launch a new encrypted EBS volume. During creation, select a KMS key for encryption.
- Attach the new encrypted volume to your instance.
- Verify that data is accessible and functioning as expected on the new encrypted volume.
- If everything is working correctly, you can delete the unencrypted volume(s) and associated snapshot(s) to free up storage and save on cost too.



Volumes (1/1) Info

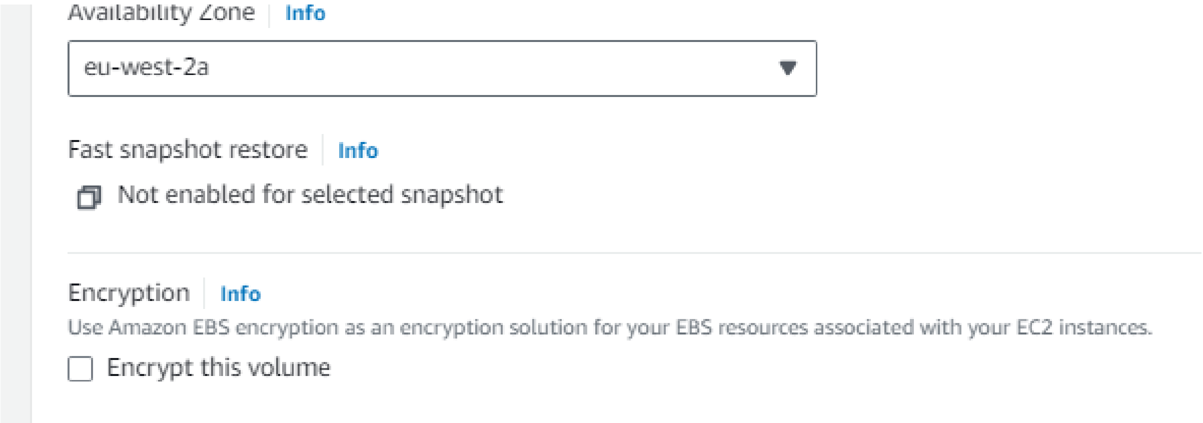
☑	Name	Volume ID	Type	Size	IOPS	Throughput	Snapsh
☑	–	vol-0f256b3aa6a2aabbd	gp2	10 GiB	100	-	-



Snapshots (1/1) Info

☑	Name	Snapshot ID	Volume size	Description	Storage tier	S
☑	–	snap-07751428ba69016fd	10 GiB	test	Standard	

Recycle Bin
Actions
Create snapshot
Owned by me
Search
Create volume from snapshot
Create image from snapshot
Copy snapshot
Modify permissions
Manage fast snapshot restore



Availability Zone Info

eu-west-2a

Fast snapshot restore Info

Not enabled for selected snapshot

Encryption Info

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

☐ Encrypt this volume

- Manage EBS Volume Encryption:
- Monitor and manage your encrypted EBS volumes:
- Backup and protect your KMS encryption keys to prevent data loss.
- Regularly rotate your encryption keys for enhanced security.
- Ensure that the appropriate IAM policies are in place to control access to KMS keys.
- Enable CloudTrail to capture relevant API calls related to EBS encryption for auditing purposes.

Testing and Verification:

- Test the functionality and performance of your encrypted EBS volumes to ensure they meet your requirements.
- Verify that applications, databases, and services dependent on the encrypted volumes are operating correctly.
- Monitor the performance impact of encryption and adjust instance types or configurations if necessary.
- Documentation and Compliance:
- Maintain proper documentation of your encrypted EBS volumes, including details on encryption keys used and any compliance requirements.
- Ensure compliance with industry standards and regulations that may mandate data encryption.

By following this guide, you can effectively encrypt your EBS volumes on AWS, protecting your data at rest and ensuring compliance with security best practices. Regularly review and update your encryption configurations as your infrastructure evolves to maintain a secure environment.

Like the procedure for encrypting an EBS volume, you can enable encryption on any RDS database. The encryption is performed at the underlying volume level and has the same IOPS performance as unencrypted volumes with a minimal effect on latency. Since any RDS database is running on an EC2 and the EBS is the raw block-level storage that is attached to the EC2 instances, there are no differences between the way you will encrypt your RDS database either at the creation or after the creation.

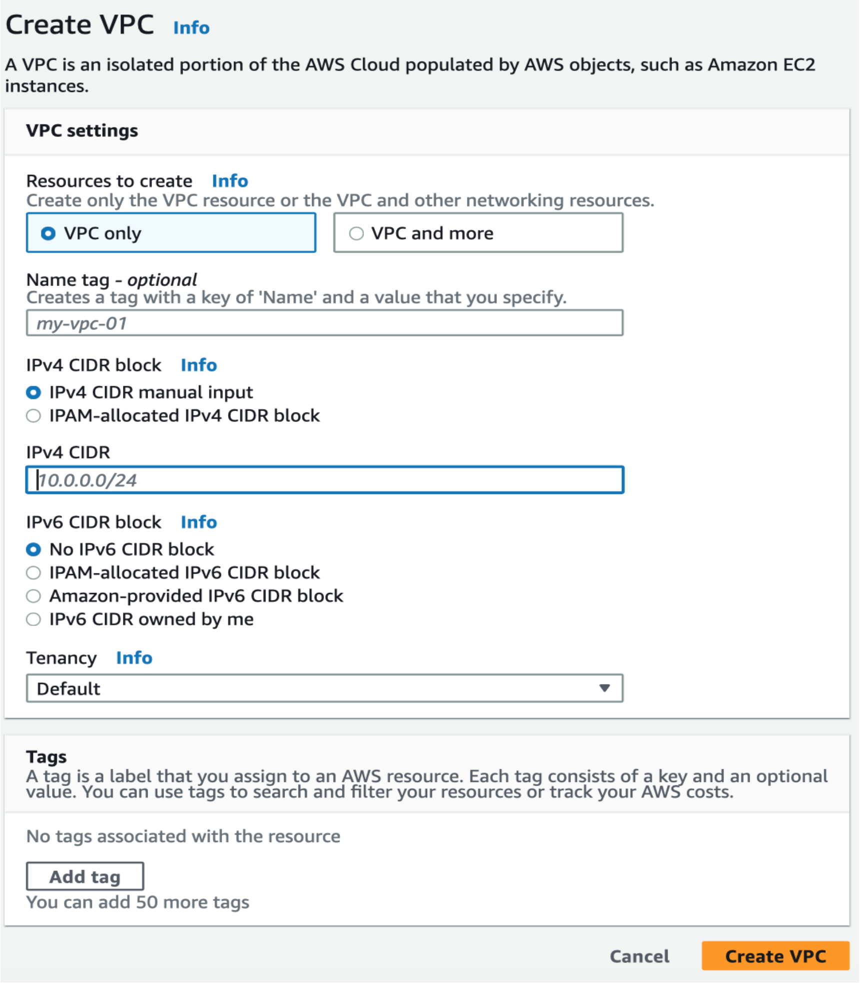
It is important to note that encrypting an EBS volume or an RDS database directly after creation is not possible. However, as mentioned earlier, you can create a snapshot, encrypt it, and then restore an RDS instance from the encrypted snapshot. This process ensures that the new instance is encrypted, serving any purposes requiring encryption.

Deploy private resources in private subnets

Step 1: Create a VPC (Virtual Private Cloud)

- Sign in to the AWS Management Console.
- Open the Amazon VPC console.
- Click on "Create VPC."
- Provide a name and CIDR block for your VPC.

- Configure other options such as IPv6 settings and DNS support.
- Click on "Create VPC" to create the VPC.



Create VPC

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create
Create only the VPC resource or the VPC and other networking resources.

● VPC only ○ VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
my-vpc-01

IPv4 CIDR block

● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24

IPv6 CIDR block

● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy
Default ▾

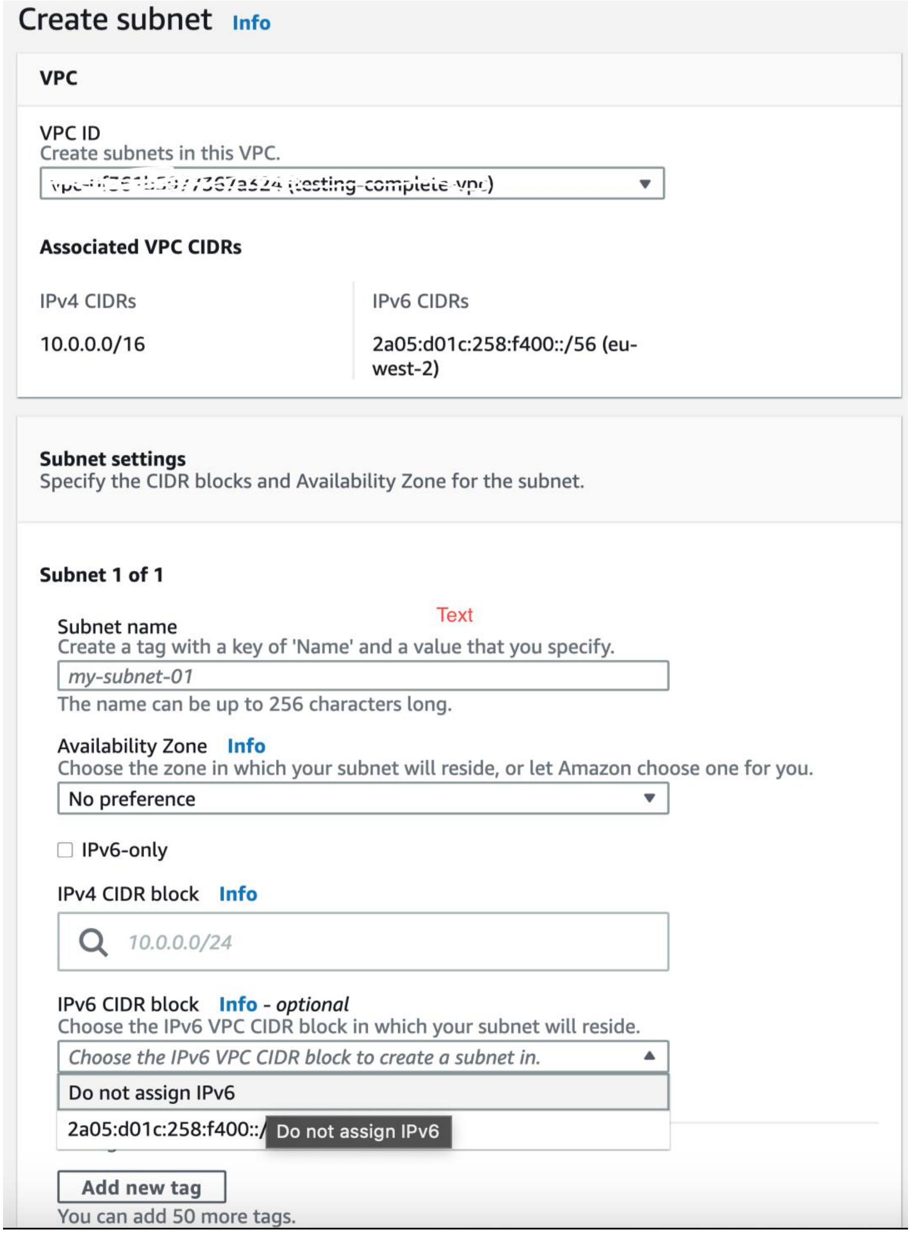
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource

You can add 50 more tags

Step 2: Create Private Subnets

- In the VPC console, click on "Subnets" in the sidebar.
- Click on "Create subnet."
- Select the VPC you created in Step 1.
- Provide a name for the subnet.
- Choose an availability zone for the subnet.
- Specify the subnet's IPv4 CIDR block.
- Make sure you select Do not assign IPV6 option.
- Click on "Create subnet" to create the private subnet.
- Repeat this step to create additional private subnets in different availability zones if desired.



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-[illegible]7367a524 (testing-complete-vpc) ▼

Associated VPC CIDRs

IPv4 CIDRs	IPv6 CIDRs
10.0.0.0/16	2a05:d01c:258:f400::/56 (eu-west-2)

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
my-subnet-01
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference ▼

□ IPv6-only

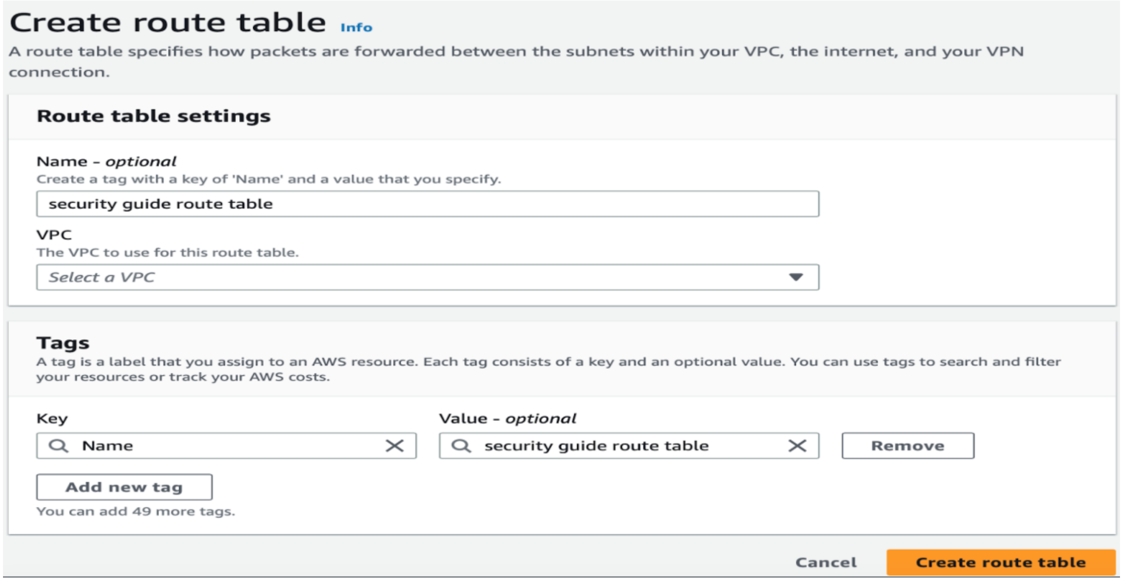
IPv4 CIDR block Info
10.0.0.0/24

IPv6 CIDR block Info - *optional*
Choose the IPv6 VPC CIDR block in which your subnet will reside.
Choose the IPv6 VPC CIDR block to create a subnet in. ▲
Do not assign IPv6
2a05:d01c:258:f400::/ Do not assign IPv6

You can add 50 more tags.

Step 3: Create Route Tables

- In the VPC console, click on "Route Tables" in the sidebar.
- Click on "Create route table."
- Provide a name for the route table.
- Select the VPC you created in Step 1.
- Click on "Create route table" to create the route table.



Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
security guide route table

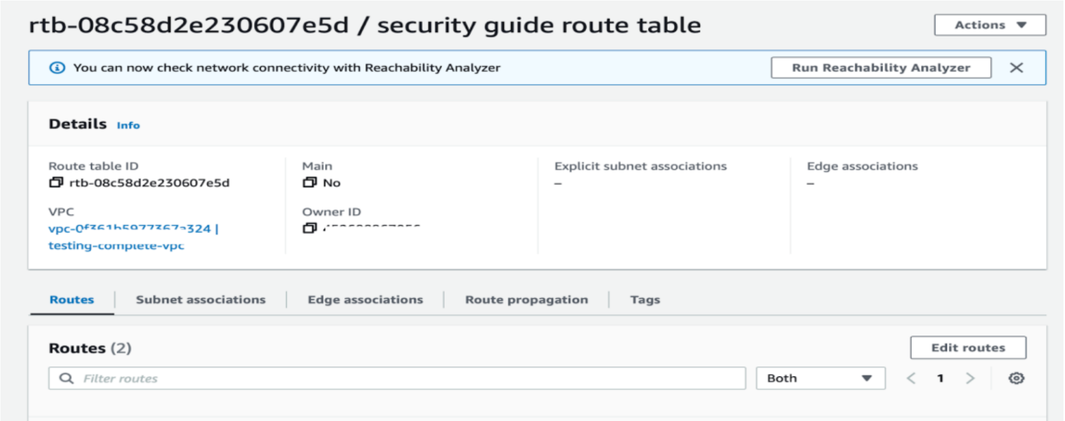
VPC
The VPC to use for this route table.
Select a VPC

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	security guide route table	Remove

You can add 49 more tags.

- In the route table details, click on the "Subnet Associations" tab.
- Click on "Edit subnet associations" and associate the private subnets created in Step 2 with the route table.



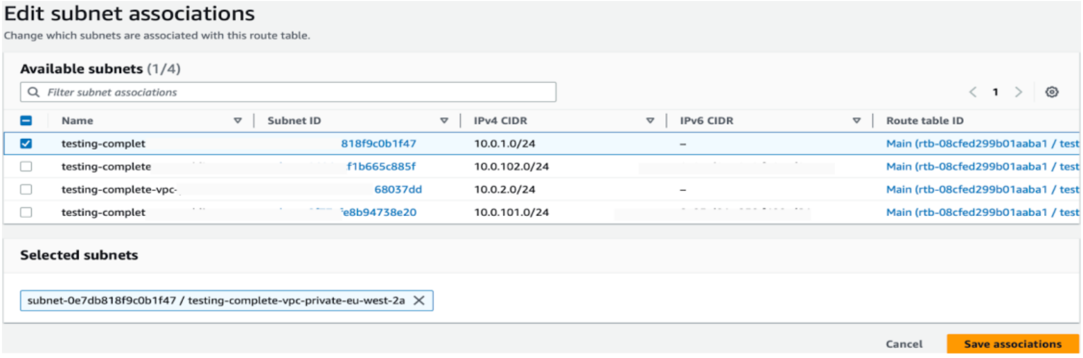
rtb-08c58d2e230607e5d / security guide route table

You can now check network connectivity with Reachability Analyzer

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-08c58d2e230607e5d	No	–	–
VPC vpc-0f[redacted]324 | testing-complete-vpc	Owner ID [redacted]		

Routes (2)



Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/4)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	testing-complet	818f9c0b1f47	10.0.1.0/24	–	Main (rtb-08cfed299b01aaba1 / test
□	testing-complete	f1b665c885f	10.0.102.0/24		Main (rtb-08cfed299b01aaba1 / test
□	testing-complete-vpc-	68037dd	10.0.2.0/24	–	Main (rtb-08cfed299b01aaba1 / test
□	testing-complet	e8b94738e20	10.0.101.0/24		Main (rtb-08cfed299b01aaba1 / test

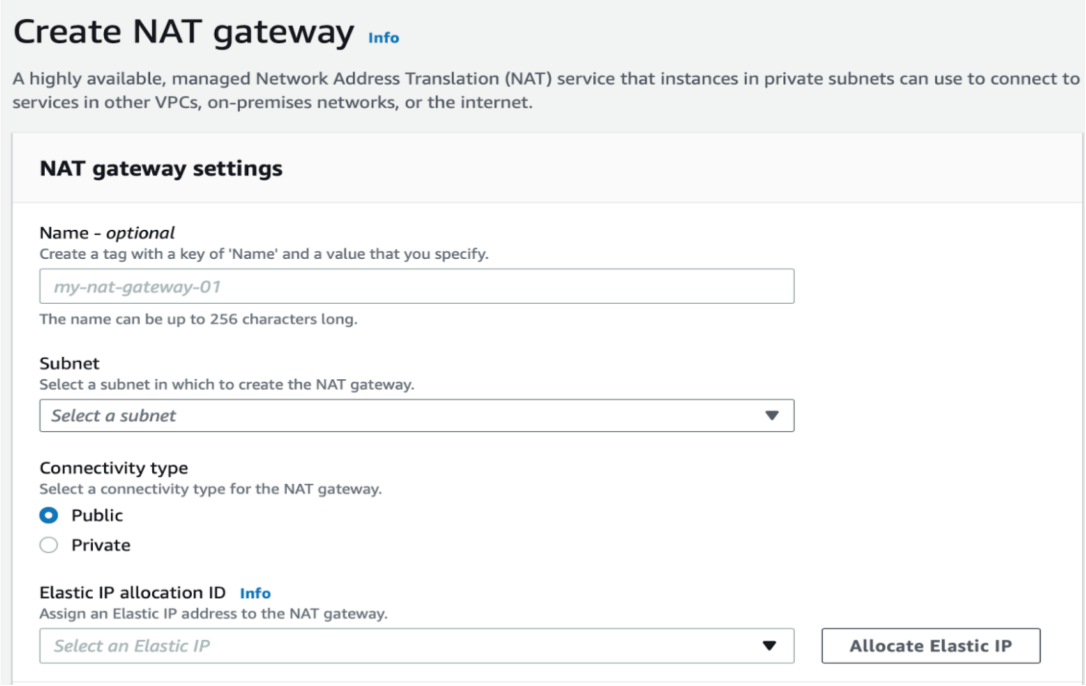
Selected subnets

subnet-0e7db818f9c0b1f47 / testing-complete-vpc-private-eu-west-2a

- Save the changes.

Step 4: Create NAT Gateways (if required)

- In the VPC console, click on "NAT Gateways" in the sidebar.
- Click on "Create NAT Gateway."



Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
my-nat-gateway-01
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
Select a subnet

Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
Select an Elastic IP

- Select the subnet corresponding to your public subnet (typically associated with an Internet Gateway).
- Ensure that an Elastic IP address is allocated to the NAT Gateway.
- Click on "Create NAT Gateway" to create the NAT Gateway.
- Associate the private subnets with the NAT Gateway for internet access (optional but required if private resources need outbound internet access).

Step 5: Launch Instances in Private Subnets

- Open the EC2 console.
- Click on "Launch Instance" to launch a new EC2 instance.
- Choose the desired AMI and instance type.
- In the "Configure Instance" section, select the VPC and the private subnet where you want to launch the instance.
- Configure other options such as security groups, storage, and tags as needed.
- Proceed to launch the instance.

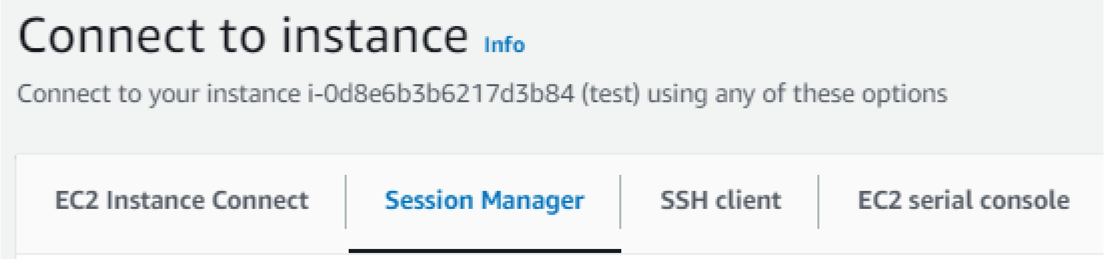
Step 6: Access Private Resources

- To access private resources in the private subnets, you have a few options:
- Connect to an instance in the private subnet using Nat Gateway or Instance
- Establish a VPC peering connection between the VPC with the private subnet and another VPC that has resources with public accessibility.
- Set up AWS Direct Connect for dedicated private connectivity to your VPC.
- Remember to configure appropriate security groups and network ACLs to control inbound and outbound traffic to your private resources.

Please note that this guide provides a general overview, and the exact steps may vary depending on your specific requirements and configurations. Always refer to the AWS documentation for detailed instructions and best practices.

Use SSM instead of SSH or RDP

You should connect remotely on your private subnets to your web servers using SSM (Session Manager), which is a functionality of System Manager. You can use the AWS CLI or the Management Console to start a session that connects into the instance through a secured tunnel, preventing the need to manage additional credentials used for Secure Shell (SSH) or Windows Remote Desktop Protocol (RDP).



Connect to instance Info

Connect to your instance i-0d8e6b3b6217d3b84 (test) using any of these options

To set up Session Manager you need to make sure that you have the following:

- The EC2 instance is using the latest Operating System (Linux, macOS or Windows) AMI's (Amazon Machine Image). On all the other instances you will have to manually install the SSM Agent, where on all the latest AMI's the SSM Agent come pre-install
- Attach the AWS managed policy AmazonSSMManagedInstanceCore to the IAM role that is associated with the instance
- The managed Instance or nodes you connect to must also allow HTTPS (port 443) outbound traffic to the following endpoints:
- EC2messages.region.amazonaws.com (where region refers to the region your resources are deployed in)
- ssm.region.amazonaws.com
- Ssmmessaged.region.amazonaws.com

Adopt a Defence in Depth approach on your firewalls

Securing resources in the AWS Virtual Private Cloud (VPC) involves utilizing two key network security features: AWS Network Access Control Lists (NACLs) and Security Groups (SGs). While they share a common objective of protecting network traffic, NACLs and SGs differ significantly in their functionality.

Firstly, AWS NACLs operate at the subnet level, providing security measures for traffic flow. They employ both allow and deny rules and apply these rules based on their numerical order. Unlike SGs, NACLs are stateless, which means that inbound and outbound rules must be explicitly defined to allow traffic. Each rule must be specified for both directions of traffic.

On the other hand, SGs are applied at the resource level, allowing for security control at a more granular level. They can be associated with various resources such as Application Load Balancers (ALBs), servers, containers, databases, and more. SGs only support allow rules, meaning that traffic is allowed based on defined rules, and all other traffic is implicitly denied. Unlike NACLs, SGs are stateful, enabling them to track the state of connections. Once a rule allowing inbound traffic to a specific port is defined, outbound traffic on the same port is automatically permitted.

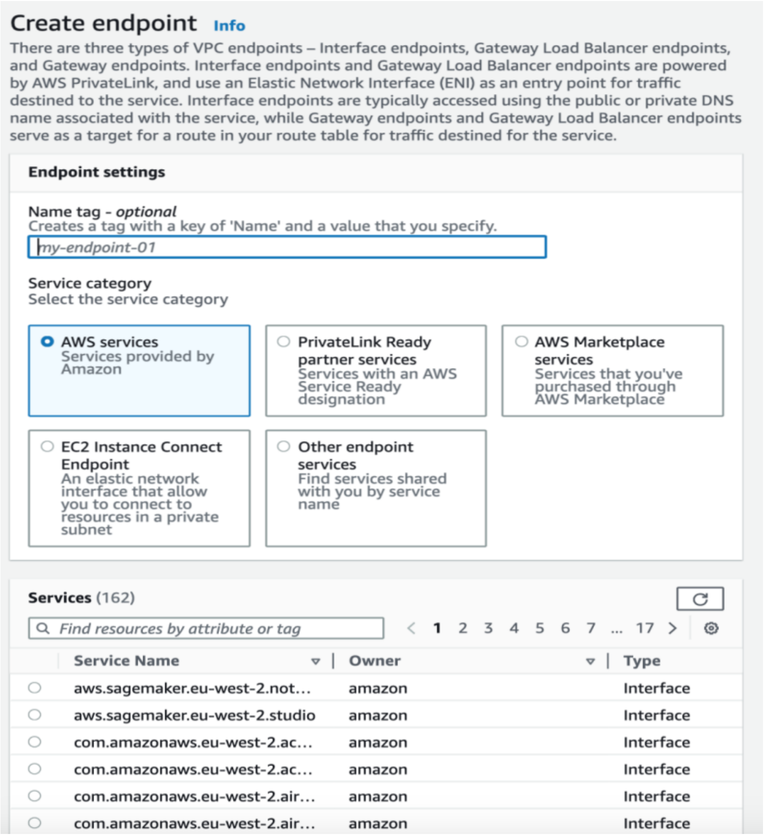
By leveraging NACLs and SGs, SMBs can establish robust network security measures within their AWS VPC. NACLs provide subnet-level protection with both allow and deny rules, while SGs offer resource-level security with allow rules and stateful connection tracking. Together, these security features help create a layered defence strategy, ensuring that only authorized traffic is allowed into and out of resources, enhancing the overall security posture of the AWS environment.

To evolve that idea, let us create an example of a SG's chaining. Let us say we want to allow traffic on port 443(HTTPS) that wants to reach our server in a private subnet. Let us assume that we also have an ALB deployed as a point of entry in our environment. Therefore, at your point of entry, it is recommended you protect your ALB with a SG. The SG's rules, at your ALB level, should have an inbound rule that allows any traffic that comes on port 443, where "any" refers to 0.0.0.0/0. Also, there should be an outbound rule that redirects the incoming traffic to the web server SG. On the Web server SG (to make it clear, DO NOT USE THE SAME SG FOR ALL THE RESOURCES YOU ARE DEPLOYING!!!), you will have an inbound rule that only allows traffic from the ALB SG, so any other traffic will be denied implicitly. As we mentioned above, the Security Groups SG's are stateful, so you do not have to allow any outbound rule back towards the ALB. However, if, let's say your web server communicates with a Database, you will have to define an outbound rule that will allow traffic from your web server to the Database SG.

As we said previously, from the NACLs perspective, best practice is to allow what traffic you need as a lower numbered rule, followed by a rule that denies everything. The rules inside the NACLs are processed in order, so any incoming traffic is filtered through them; if the traffic does not get allowed, then will be denied. Also remember, on the NACLs you will have to define that outbound traffic as well, as we mentioned above the NACLs are stateless.

Use VPC Endpoints to access supported services

- Step 1: Determine the Endpoint Service and Region
- Identify the AWS service for which you want to create a VPC endpoint.
- Determine the region in which the service is located.
- Step 2: Create an Endpoint Policy (if required)
- If the chosen AWS service requires an endpoint policy, create a JSON policy document that defines the permissions for accessing the service.
- Include the necessary permissions and restrictions based on your requirements.
- Save the policy document for later use.
- Step 3: Create a VPC Endpoint
- Sign in to the AWS Management Console.
- Open the Amazon VPC console.
- Click on "Endpoints" in the sidebar.
- Click on "Create Endpoint."
- Select the service category for the endpoint, such as AWS services or Marketplace partner services.
- Choose the AWS service for which you want to create the VPC endpoint.



Create endpoint Info

There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints. Interface endpoints and Gateway Load Balancer endpoints are powered by AWS PrivateLink, and use an Elastic Network Interface (ENI) as an entry point for traffic destined to the service. Interface endpoints are typically accessed using the public or private DNS name associated with the service, while Gateway endpoints and Gateway Load Balancer endpoints serve as a target for a route in your route table for traffic destined for the service.

Endpoint settings

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
my-endpoint-01

Service category
Select the service category

● AWS services
Services provided by Amazon

○ PrivateLink Ready partner services
Services with an AWS Service Ready designation

○ AWS Marketplace services
Services that you've purchased through AWS Marketplace

○ EC2 Instance Connect Endpoint
An elastic network interface that allow you to connect to resources in a private subnet

○ Other endpoint services
Find services shared with you by service name

Services (162)

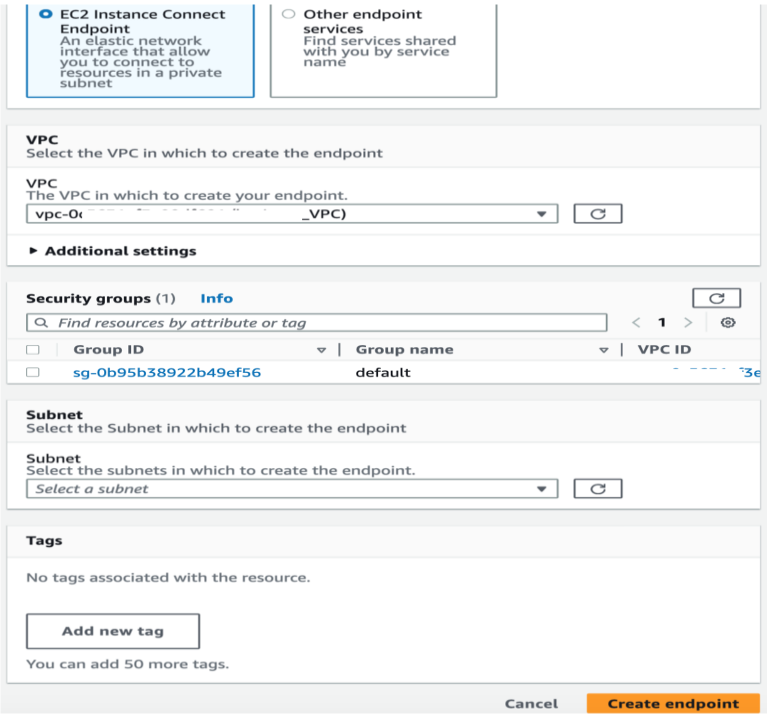
Find resources by attribute or tag

< 1 2 3 4 5 6 7 ... 17 >

	Service Name	Owner	Type
○	aws.sagemaker.eu-west-2.not...	amazon	Interface
○	aws.sagemaker.eu-west-2.studio	amazon	Interface
○	com.amazonaws.eu-west-2.ac...	amazon	Interface
○	com.amazonaws.eu-west-2.ac...	amazon	Interface
○	com.amazonaws.eu-west-2.air...	amazon	Interface
○	com.amazonaws.eu-west-2.air...	amazon	Interface

- Select your VPC and specify the subnet(s) in which you want to create the endpoint.

- If an endpoint policy is required, click on "Policy" and attach the policy document you created in Step 2.



● EC2 Instance Connect Endpoint
An elastic network interface that allow you to connect to resources in a private subnet

○ Other endpoint services
Find services shared with you by service name

VPC
Select the VPC in which to create the endpoint

VPC
The VPC in which to create your endpoint.
vpc-0[illegible]_VPC)

▸ **Additional settings**

Security groups (1) Info

Find resources by attribute or tag

□	Group ID	Group name	VPC ID
□	sg-0b95b38922b49ef56	default	[illegible]3e

Subnet
Select the Subnet in which to create the endpoint

Subnet
Select the subnets in which to create the endpoint.
Select a subnet

Tags

No tags associated with the resource.

You can add 50 more tags.

- Click on "Create endpoint" to create the VPC endpoint.
- Step 4: Verify the VPC Endpoint Status
- After creating the VPC endpoint, wait for the status to change to "Available."
- Refresh the VPC Endpoints page to see the status.
- Step 5: Test Connectivity to the Endpoint
- To ensure the VPC endpoint is functioning correctly, launch an EC2 instance within the same VPC and subnet as the endpoint.
- Connect to the EC2 instance via SSH or RDP.
- Attempt to access the AWS service associated with the VPC endpoint.
- For example, if it is an S3 VPC endpoint, try accessing an S3 bucket or object.
- If it is a DynamoDB VPC endpoint, attempt to query or write data to the DynamoDB table.
- If the access is successful, it confirms the connectivity to the VPC endpoint.
- Step 6: Update Security Groups and Network ACLs (if required)
- If necessary, update your security groups and network ACLs to allow traffic to and from the VPC endpoint.
- Adjust the inbound and outbound rules to permit the required access based on your application's needs.
- Ensure that the necessary ports and protocols are open to establish connectivity.
- Step 7: Repeat Steps for Additional VPC Endpoints (if needed)
- If you need to create VPC endpoints for other AWS services or in different VPCs or subnets, repeat the steps outlined above.

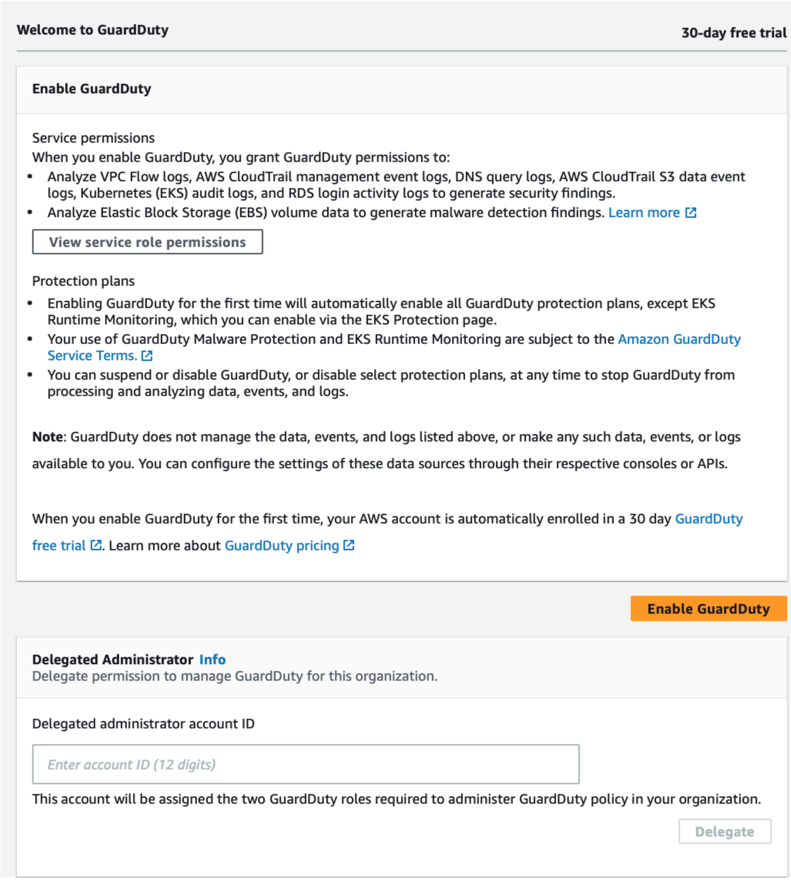
- Customize the endpoint settings and policies based on the specific requirements of each service.

Please note that the specific steps and options may vary depending on the AWS Management Console interface version or updates made to AWS services. Always refer to the AWS documentation for the latest instructions and best practices when setting up VPC endpoints.

Enable and respond to GuardDuty notifications

Amazon GuardDuty is an intelligent threat detection service that provides customers with a way to continuously monitor and protect their AWS accounts and workloads. GuardDuty identifies suspected attackers through integrated threat intelligence feeds and uses machine learning to detect anomalies in account and workload activity. It monitors for activity such as unusual API calls or unauthorized deployments that indicate that a customer's accounts may have been compromised, as well as direct threats like compromised instances or reconnaissance by attackers.

- Enable GuardDuty: First, you need to enable GuardDuty in your AWS account. This can be done through the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. Enable GuardDuty on the regions where you want to monitor for threats.



Welcome to GuardDuty — 30-day free trial

Enable GuardDuty

Service permissions
When you enable GuardDuty, you grant GuardDuty permissions to:

- Analyze VPC Flow logs, AWS CloudTrail management event logs, DNS query logs, AWS CloudTrail S3 data event logs, Kubernetes (EKS) audit logs, and RDS login activity logs to generate security findings.
- Analyze Elastic Block Storage (EBS) volume data to generate malware detection findings. Learn more

View service role permissions

Protection plans

- Enabling GuardDuty for the first time will automatically enable all GuardDuty protection plans, except EKS Runtime Monitoring, which you can enable via the EKS Protection page.
- Your use of GuardDuty Malware Protection and EKS Runtime Monitoring are subject to the Amazon GuardDuty Service Terms.
- You can suspend or disable GuardDuty, or disable select protection plans, at any time to stop GuardDuty from processing and analyzing data, events, and logs.

Note: GuardDuty does not manage the data, events, and logs listed above, or make any such data, events, or logs available to you. You can configure the settings of these data sources through their respective consoles or APIs.

When you enable GuardDuty for the first time, your AWS account is automatically enrolled in a 30 day GuardDuty free trial. Learn more about GuardDuty pricing

Enable GuardDuty

Delegated Administrator Info
Delegate permission to manage GuardDuty for this organization.

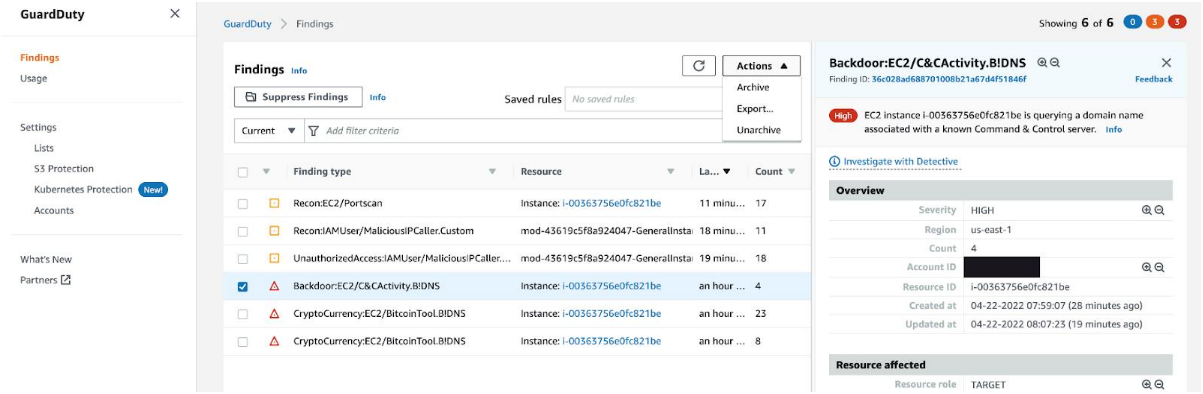
Delegated administrator account ID

Enter account ID (12 digits)

This account will be assigned the two GuardDuty roles required to administer GuardDuty policy in your organization.

Delegate

- Set up SNS Topic: Create an Amazon Simple Notification Service (SNS) topic to receive notifications from GuardDuty. This topic acts as a communication channel for receiving alerts.
- Configure GuardDuty to send notifications: In the GuardDuty settings, configure it to send findings to the SNS topic you created. You can specify the types of findings you want to receive notifications for, such as unauthorized access, malware, or data exfiltration.
- Subscribe to the SNS Topic: Subscribe to the SNS topic using your preferred method (email, SMS, Lambda function, etc.) to receive the GuardDuty notifications.
- Configure and prioritize notification responses: Determine how you want to respond to different types of GuardDuty findings based on their severity. For example, you might have different response actions for low, medium, or high severity findings.
- Implement automated responses: You can use AWS Lambda functions or other automation tools to trigger automated response actions based on the notifications received. These actions can include blocking IP addresses, terminating instances, or triggering incident response workflows.
- Monitor and investigate: Regularly monitor the GuardDuty findings and notifications. When a notification is received, investigate the finding to understand the nature of the threat and assess its impact. Determine the appropriate mitigation or response steps based on your security policies.



□		Finding type	Resource	La... ▼	Count
□	□	Recon:EC2/Portscan	Instance: i-00363756e0fc821be	11 minu...	17
□	□	Recon:IAMUser/MaliciousIPCaller.Custom	mod-43619c5f8a924047-GeneralInstai	18 minu...	11
□	□	UnauthorizedAccess:IAMUser/MaliciousIPCaller....	mod-43619c5f8a924047-GeneralInstai	19 minu...	18
☑	△	Backdoor:EC2/C&CActivity.B!DNS	Instance: i-00363756e0fc821be	an hour ...	4
□	△	CryptoCurrency:EC2/BitcoinTool.B!DNS	Instance: i-00363756e0fc821be	an hour ...	23
□	△	CryptoCurrency:EC2/BitcoinTool.B!DNS	Instance: i-00363756e0fc821be	an hour ...	8

Backdoor:EC2/C&CActivity.B!DNS

Finding ID: 36c028ad688701008b21a67d4f51846f

High EC2 instance i-00363756e0fc821be is querying a domain name associated with a known Command & Control server. Info

Overview	
Severity	HIGH
Region	us-east-1
Count	4
Account ID	[redacted]
Resource ID	i-00363756e0fc821be
Created at	04-22-2022 07:59:07 (28 minutes ago)
Updated at	04-22-2022 08:07:23 (19 minutes ago)

Resource affected	
Resource role	TARGET

- Take necessary actions: Based on the severity and type of finding, take appropriate actions to mitigate the threat. This may involve remediating the affected resources, escalating to your security team, or engaging AWS Support for assistance.
- Review and refine: Continuously review your GuardDuty findings and notifications to identify any patterns or recurring issues. Refine your response actions and security controls based on the insights gained from these reviews.

It is important to note that the specific steps for enabling and responding to GuardDuty notifications may vary depending on your AWS environment and the tools you use for automation and incident response. The above steps provide a general framework to help you get started with GuardDuty notifications and responses.

In conclusion, security remains a dynamic priority in technology, IT, and cloud infrastructure. Whether you opt for cloud migration, maintain on-premises systems, or embrace a hybrid setup, the approach to securing your infrastructure evolves continuously to counter emerging threats. The guidelines provided here serve as a starting point for securing your cloud adoption journey. Should you encounter specific requirements or need expert assistance navigating the complexities of cloud security, our team and AWS experts are readily available to help.

For further assistance, visit our website at www.digitalcloudadvisor.com or email us at support@digitalcloudadvisor.com to connect with one of our knowledgeable security experts who can provide tailored support and guidance.