20 IAM Access Key Best Practices: that every **Cloud Engineer** should know

1. IAM Access Key Creation

What is it?

IAM access keys are a pair of security credentials (access key ID and secret access key) used to interact with AWS services programmatically.

Why use it?

If you need to use AWS SDKs, command-line tools, or direct HTTPS calls to AWS APIs, you will need an IAM access key.

How to use it?

You can create an IAM access key through the AWS Management Console, CLI, or SDKs.

Not using?

If you don't use IAM access keys, you won't be able to make programmatic calls to AWS services.

2. Access Key Rotation

What is it?

Rotating IAM access keys means generating new keys and retiring old ones periodically.

Why use it?

Regular key rotation reduces the risk of unauthorized use of old or compromised keys.

How to use it?

In AWS Management Console, you can create a new access key and then disable or delete the old one.

Not using?

If you don't regularly rotate your keys, you could be at higher risk of security breaches.

3. Securing IAM Access Keys

What is it?

Securing IAM access keys involves storing them securely and never exposing them in any publicly accessible areas.

Why use it?

IAM access keys can be used to gain programmatic access to your AWS resources. It's critical to keep them secure.

How to use it?

Avoid hard-coding keys into your applications, and use secure methods such as AWS Secrets Manager or environment variables to store them.

Not using?

If you don't secure your keys properly, they could be leaked or stolen, leading to unauthorized access to your AWS resources.

4. Access Key Limitation

What is it?

AWS limits each IAM user to have two access keys at any given time.

Why use it?

This limit ensures control over access key issuance and promotes regular key rotation.

How to use it?

If you already have two keys, you'll need to delete an existing one before you can create a new one.

Not using?

If you're not aware of this limitation, you might encounter issues when trying to create new keys.

5. Disabling IAM Access Keys

What is it?

IAM access keys can be disabled when not in use.

Why use it?

Disabling keys that aren't currently needed minimizes potential security risks.

How to use it?

You can disable a key through the AWS Management Console, AWS CLI, or AWS SDKs.

Not using?

If you leave unnecessary keys enabled, you may expose your AWS resources to unnecessary risk.

6. Deleting IAM Access Keys

What is it?

IAM access keys can be deleted when they're no longer needed.

Why use it?

Deleting unneeded keys permanently removes their access to AWS APIs, enhancing your security.

How to use it?

You can delete a key through the AWS Management Console, AWS CLI, or AWS SDKs.

Not using?

If you don't delete keys that are no longer needed, you're unnecessarily increasing potential security risks.

7. Using Access Keys with AWS CLI

What is it?

IAM access keys can be used with the AWS Command Line Interface to make programmatic calls to AWS.

Why use it?

The AWS CLI provides a unified way to interact with AWS services from a command line.

How to use it?

After installing the AWS CLI, you can configure it with your access keys using the `aws configure` command.

Not using?

If you don't use access keys with the AWS CLI, you won't be able to interact with AWS services programmatically from the command line.

8. Using IAM Roles Instead of Access Keys

What is it?

IAM roles can be assumed by entities (like EC2 instances), which are then given temporary permissions to carry out AWS tasks.

Why use it?

IAM roles enhance security by eliminating the need for long-term, user-specific credentials.

How to use it?

You can create an IAM role with specific permissions and then associate it with an EC2 instance or other AWS service.

Not using?

If you don't use IAM roles, you may need to manage long-term credentials for various services, increasing security risks.

9. Access Keys and AWS SDKs

What is it?

AWS Software Development Kits (SDKs) also use IAM access keys for authentication.

Why use it?

SDKs allow developers to interact with AWS services using various programming languages.

How to use it?

You can configure AWS SDKs with your access keys, similar to how you do it with the AWS CLI.

Not using?

If you don't use access keys with AWS SDKs, you won't be able to use the SDKs to interact with AWS services programmatically.

10. Tracking Access Key Usage

What is it?

AWS CloudTrail allows you to monitor and retain account activity related to actions across your AWS infrastructure, including access key usage.

Why use it?

Tracking access key usage helps you audit and review actions performed with access keys, which is essential for security.

How to use it?

Enable AWS CloudTrail and check the logs regularly for activity involving your access keys.

Not using?

If you don't track key usage, it may be difficult to investigate and understand activities in your AWS environment.

11. Understanding Access Key Visibility

What is it?

IAM access keys are only visible when they are first created.

Why use it?

For security reasons, AWS does not allow access keys to be retrieved later.

How to use it?

When you create a key pair, make sure to securely save the keys in a safe place.

Not using?

If you fail to record your keys when you create them, you'll need to delete and recreate them to get access again.

12. IAM Access Keys for Root Account

What is it?

Root account credentials have full access to all resources in your AWS account.

Why use it?

AWS strongly discourages the use of root account credentials for daily interactions. IAM user access keys should be used instead.

How to use it?

Create individual IAM users with appropriate permissions for everyday tasks and limit the use of your root account.

Not using?

If you regularly use your root account access keys, you're exposing your entire AWS account to unnecessary risk.

13. IAM User Access Keys

What is it?

IAM User Access Keys are unique to each user and should be used for day-to-day interactions.

Why use it?

These keys provide the principle of least privilege, only granting access to the services a user needs.

How to use it?

Create an IAM user, grant necessary permissions, and then create access keys for that user.

Not using?

If not used, you risk exposing your root access keys or over-granting privileges.

14. Access Keys and STS

What is it?

AWS Security Token Service (STS) allows you to request temporary, limited-privilege credentials for IAM users.

Why use it?

This helps manage permissions and reduce the risks associated with long-term keys.

How to use it?

Request a set of temporary security credentials using the `AssumeRole` or `GetSessionToken` APIs.

Not using?

If you don't use STS with your access keys, you might not be effectively managing their lifespan and associated risks.

15. Key Age Monitoring

What is it?

Monitoring the age of your IAM access keys helps you manage key rotation effectively.

Why use it?

Old keys are at a higher risk of being compromised, so it's best practice to rotate them regularly.

How to use it?

Use the AWS Management Console or AWS CLI to check the age of your access keys regularly.

Not using?

If not monitored, you may be using outdated keys, increasing your security risks.

16. Access Key Metadata

What is it?

Metadata for IAM access keys includes information like the creation date and the last time the key was used.

Why use it?

This information helps track key usage and decide when to rotate keys.

How to use it?

You can view access key metadata in the IAM console or via AWS CLI.

Not using?

If you don't keep track of key metadata, it might be more challenging to manage your keys effectively.

17. Access Keys and AWS Regions

What is it?

IAM access keys are not tied to a specific region and can be used to access resources in any AWS region.

Why use it?

This provides flexibility in managing resources across multiple regions.

How to use it?

You can use the same access keys to make requests to different regions.

Not using?

If you are not aware of this, you might unnecessarily create additional access keys for different regions.

18. Access Key Permissions

What is it?

The permissions for an IAM access key are determined by the policies attached to the IAM user or group that the key belongs to.

Why use it?

This allows granular control over what AWS services and resources the access key can interact with.

How to use it?

Create IAM policies that define the necessary permissions and attach them to the IAM user or group.

Not using?

If you don't properly manage access key permissions, you could either over-provision or under-provision access to AWS resources.

19. Understanding Access Key IDs

What is it?

An Access Key ID is a part of the access key pair and is not a secret.

Why use it?

While it is not a secret, it is unique to the user and is used to identify the access key pair in AWS interactions.

How to use it?

The Access Key ID is used in conjunction with the Secret Access Key when making AWS API requests.

Not using?

If not used properly, AWS API requests will fail authentication.

20. Understanding Secret Access Keys

What is it?

The Secret Access Key is also part of the access key pair and is a secret.

Why use it?

It is used to calculate the signature to include in AWS API requests for authentication.

How to use it?

The Secret Access Key is used in conjunction with the Access Key ID when making AWS API requests.

Not using?

If not used or used improperly, AWS API requests will fail authentication.