AWS Root User and IAM

AWS Root User

Smart Internz

AWS Root User

- Has **full access** to all AWS services and resources, including billing and personal information.
- Should never be used for everyday tasks to minimize security risks.

Root User Credentials

- The root user has a unique password and access keys.
- Credentials should be securely stored and not shared.

Root Credentials

Username:

`john.doe_aws`

Password:

`A!9tD*2pW@8bF#0z`

Access keys

- **Access key ID:** for example, ***A2lAl5EXAMPLE***
- **Secret access key:** for example, ***wJalrFE/KbEKxE***

Smart Internz

AWS Root User Best Practices

1. Choose a Strong Password
2. Enable Multi-Factor Authentication (MFA)
3. Never Share Root User Credentials
4. Disable/Delete Access Keys
5. Use IAM Users for Admin Tasks
6. Monitor Root User Activity

Multi-Factor Authentication

Multi-Factor Authentication

- **Extra Security Layer:**

MFA requires both a password and a second factor (e.g., code from your phone).

- **How It Works:**

After entering your password, a time-sensitive code is provided to verify your identity.

- **AWS Support:**

MFA is supported for both root and IAM users, adding protection against unauthorized access.

AWS Identity and Access Management

Smart Internz

Authentication

- **Identity Verification:**
- Authentication is the process of verifying who a user is, typically through credentials like a username and password.
- **Methods:**
- Common methods include passwords, biometrics, and multi-factor authentication (MFA).
- **Purpose:**
- Ensures that only authorized users can access a system or service.

Authorization

- **Access Control:**
- Authorization determines what actions a user can perform on a system after authentication.
- **Permissions:**
- It involves setting permissions (read, write, execute) on resources to control user actions.
- **Role-Based:**
- Access is often managed by assigning users to roles or groups with specific permissions.

Smart Internz

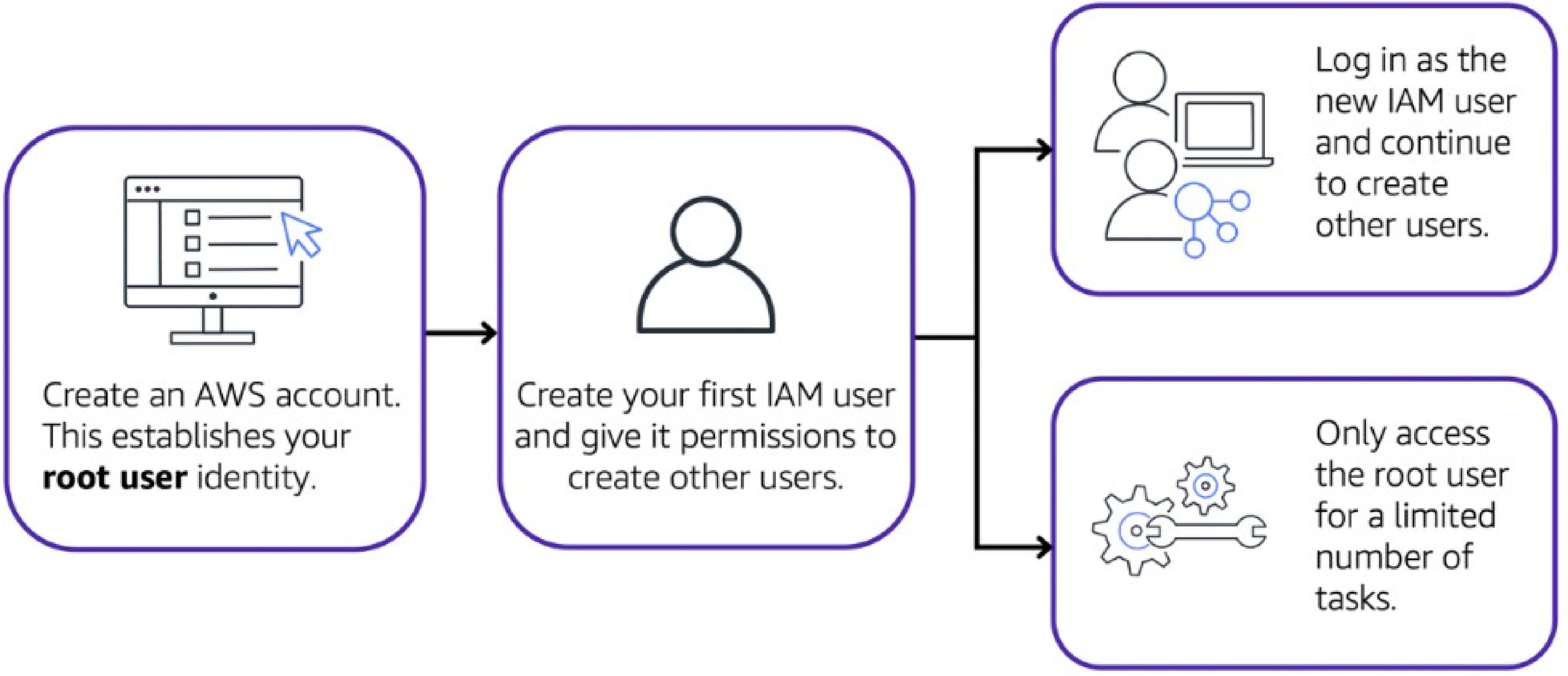
IAM (Identity and Access Management)

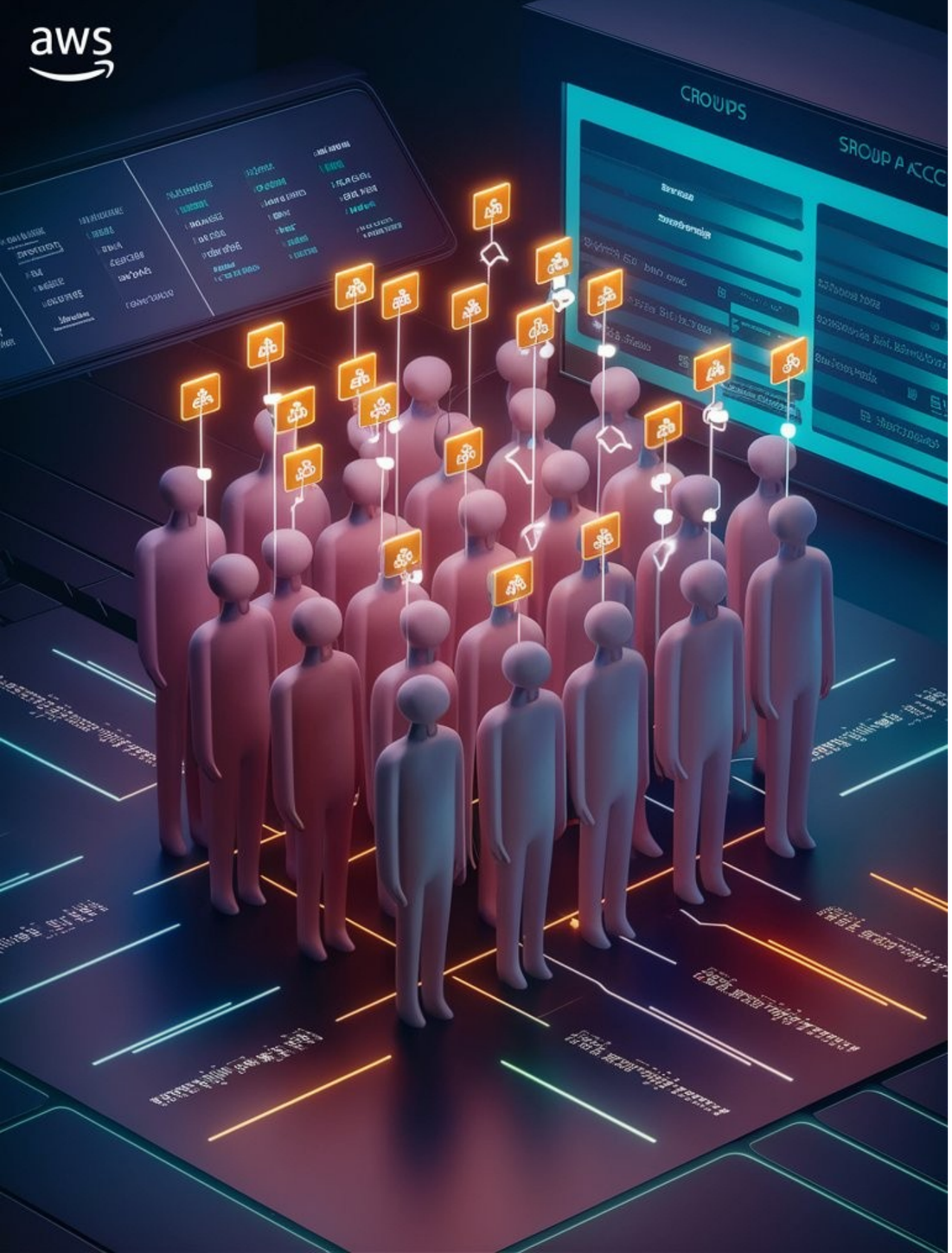
- AWS service to control access to AWS resources.
- Helps manage who can access your AWS account and what they can do.

Smart Internz

IAM User

- An individual identity in AWS with specific permissions.
- Each user has a unique set of credentials (username and password).

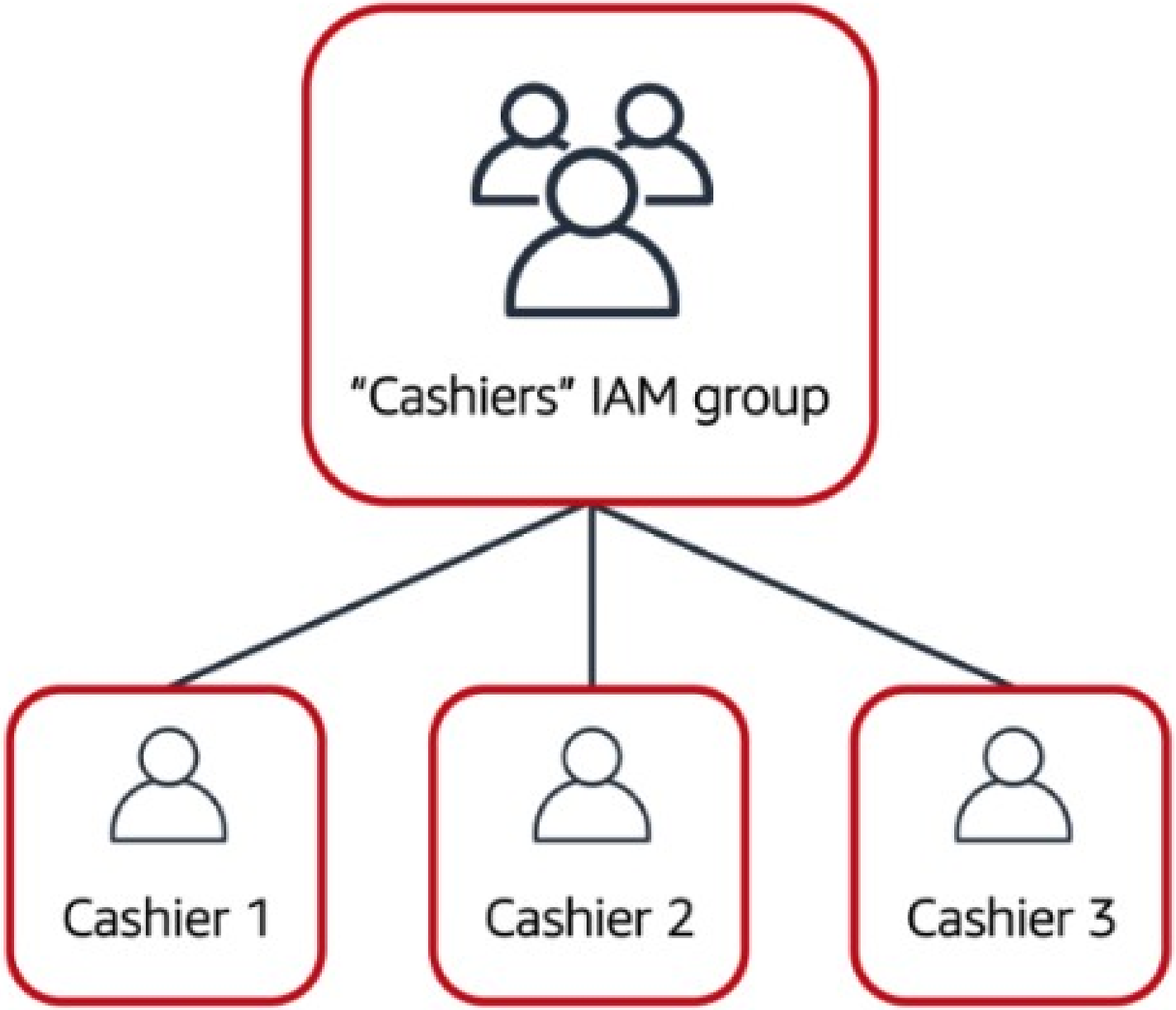

Create an AWS account. This establishes your **root user** identity.
Create your first IAM user and give it permissions to create other users.
Log in as the new IAM user and continue to create other users.
Only access the root user for a limited number of tasks.



Smart Internz

IAM Groups

- A collection of IAM users.
- You can assign permissions to a group, and all users in the group inherit those permissions.


"Cashiers" IAM group
Cashier 1
Cashier 2
Cashier 3

Smart Internz

IAM Policies

- Documents defining permissions for users, groups, or roles.
- Specifies what actions are allowed or denied on specific AWS resources.

IAM Policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListObject",
    "Resource": "arn:aws:s3:::
AWSDOC-EXAMPLE-BUCKET"
  }
}
```

This example IAM policy allows permission to access the objects in the Amazon S3 bucket with ID: *AWSDOC-EXAMPLE-BUCKET*.

Smart Internz

IAM Roles

- A set of permissions that define what actions are allowed.
- Can be assumed by users or services to perform specific tasks (e.g., EC2 instances accessing S3).

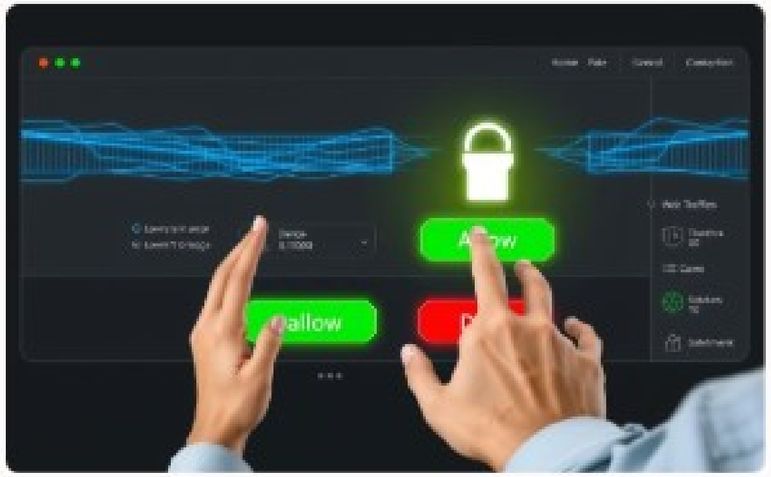
AWS Security Services Overview

Amazon GuardDuty

Continuous threat detection and analysis.

AWS Shield

Automatic DDoS attack protection.



AWS WAF

Web Application Firewall for web traffic control.

Smart Internz

IAM Best Practices

1. **Lock down the AWS root user -** Limit use and secure with MFA**.**
2. **Follow the principle of least privilege -** Grant only necessary permissions.
3. **Use IAM appropriately -** Assign users to groups, not individual permissions.
4. **Use IAM roles when possible -** Use roles for temporary access and automation.
5. **Consider using an identity provider -** Integrate with external identity systems.
6. **Regularly review and remove unused users, roles, and credentials -** Maintain clean, secure IAM configurations.