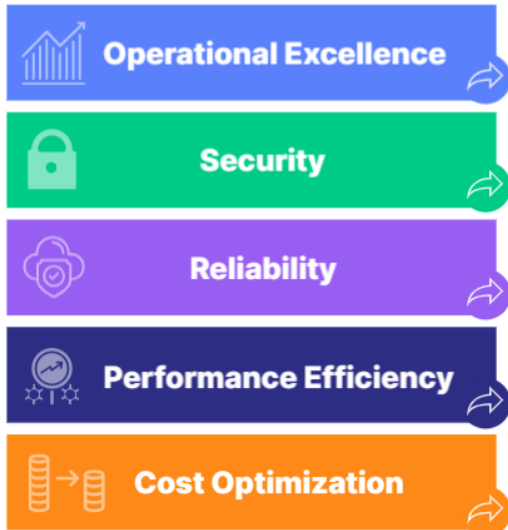# Well-Architected Framework

The **5 pillars** of the **Well-Architected Framework** describe design principles and best practices for running workloads in the cloud.

Hover to see more about each pillar.

Operational Excellence

Security

Reliability

Performance Efficiency

Cost Optimization

5 Pillars

5 Pillars in the Real World

Here are a **few examples** of applying best practices and design principles from the 5 pillars of the **Well-Architected Framework** in the real world.

1 Operational Excellence

CodeCommit

You can use AWS CodeCommit for version control to enable tracking of code changes and to version-control CloudFormation templates of your infrastructure.

2 Security

CloudTrail

You can configure central logging of all actions performed in your account using CloudTrail.

3 Reliability

RDS

You can use Multi-AZ deployments for enhanced availability and reliability of RDS databases.

4 Performance Efficiency

Lambda

You can use AWS Lambda to run code with zero administration.
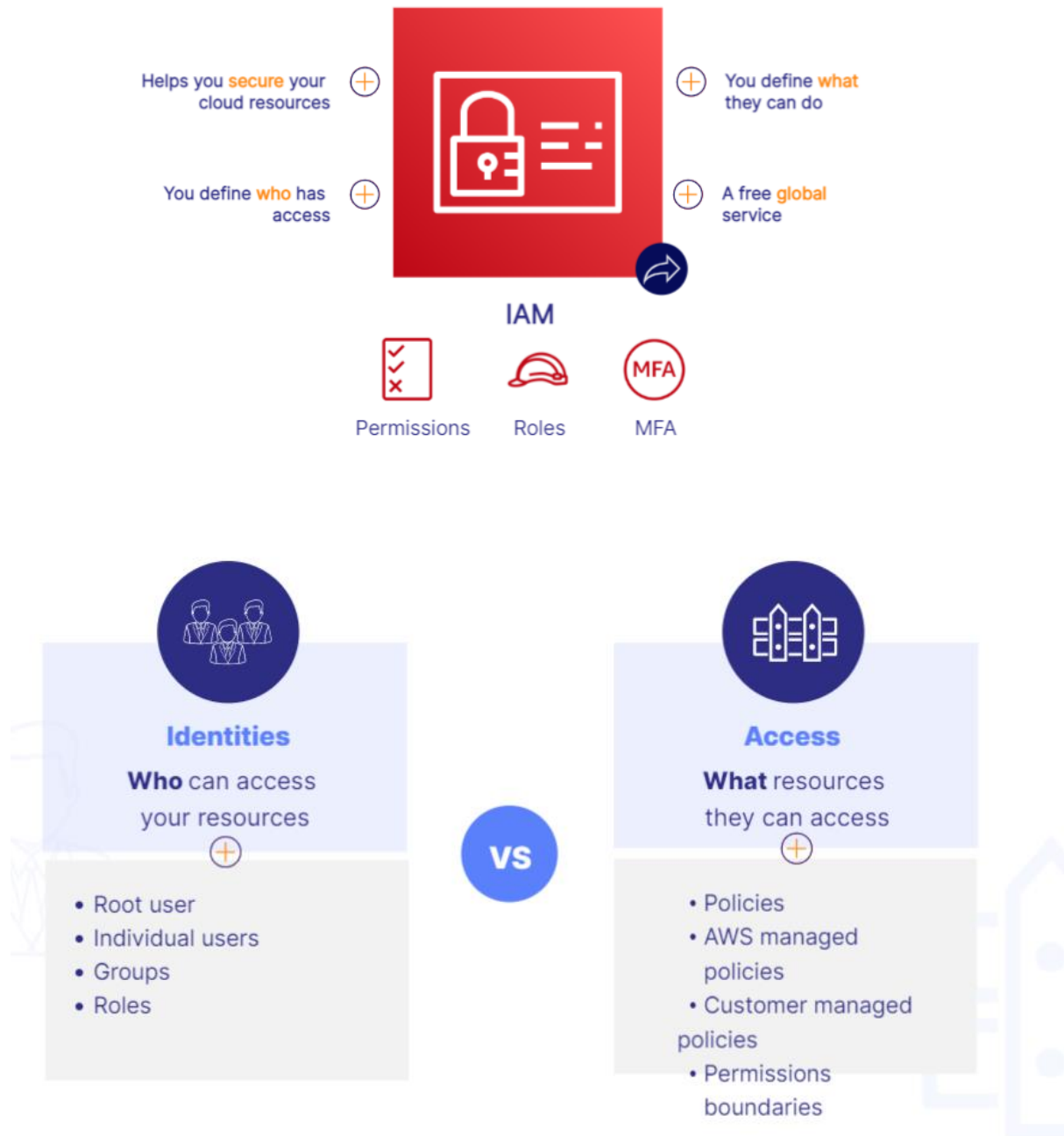
5 Cost Optimization

S3

You can use S3 Intelligent-Tiering to automatically move your data between access tiers based on your usage patterns.

IAM Users:

Identity and Access Management (IAM)

IAM allows you to control access to your AWS services and resources.

Helps you **secure** your cloud resources ⊕

You define **who** has access ⊕

You define **what** they can do ⊕

A free **global** service ⊕

IAM

Permissions   Roles   MFA

Identities

Who can access your resources ⊕

- Root user
- Individual users
- Groups
- Roles

VS

Access

What resources they can access ⊕

- Policies
- AWS managed policies
- Customer managed policies
- Permissions boundaries

What Is Authentication?

⊖

CLICK HERE

Authentication is where you present your identity (**username**) and provide verification (**password**).

What Is Authorization?

⊖

CLICK HERE

Authorization determines which services and resources the authenticated identity has **access** to.

WHO

POLICY

```
{
  "Version": ". . .",
  "Statement": [
    {
      "Action": [ . . .],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

aws AWS Cloud

DynamoDB

Lambda

S3

API Gateway

WHAT

Users are entities you create in **IAM** to represent the **person** or **application** needing to access your **AWS resources**.

ROOT USER

The root user is created when you first open your AWS account.

What can only the root user do?
- Close your account
- Change email address
- Modify your support plan

READ ENTIRE LIST

USERS

Individual users are created in IAM and are used for everyday tasks.

What can individual users do?
- Perform administrative tasks
- Access application code
- Launch EC2 instances
- Configure databases

Don't forget activity performed by users in your account is billed to your account!
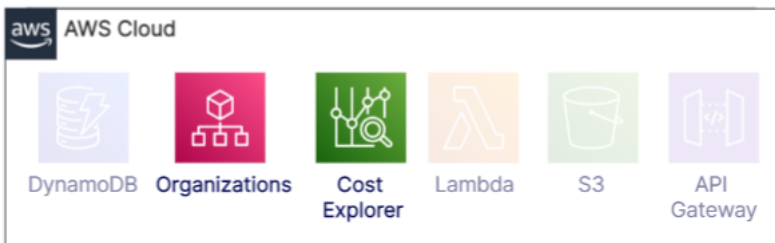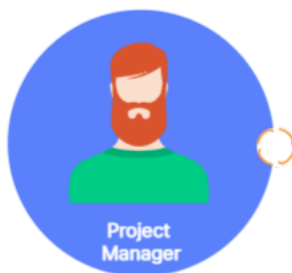
APPLICATIONS

Did you know applications can be users?

You'll create a user in IAM so you can generate access keys for an application running on-premises that needs access to your cloud resources.

The **principle of least privilege** involves giving a user the **minimum** access required to get the job done.
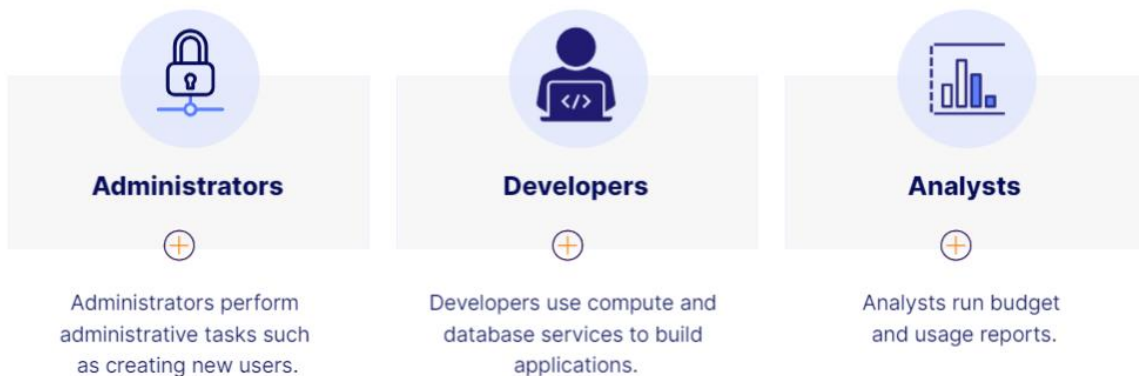

Developer

AWS Cloud

DynamoDB · Organizations · Cost Explorer · Lambda · S3 · API Gateway

Developers are responsible for building applications.

Project Manager

AWS Cloud

DynamoDB · **Organizations** · **Cost Explorer** · Lambda · S3 · API Gateway

Project managers are responsible for managing the budget.

Groups

A group is a **collection of IAM users** that helps you apply common access controls to all group members.

Administrators

⊕

Administrators perform administrative tasks such as creating new users.

Developers

⊕

Developers use compute and database services to build applications.

Analysts

⊕

Analysts run budget and usage reports.

Things to Remember When
Studying for the Exam

✓ **Users and groups**

Going into the exam, understand the differences between users and group.

✓ **Root user tasks**

Remember the tasks that only the root user can do.

✓ **Principle of least privilege**

Don't forget about the principle of least privilege.

✓ **Real-world use cases**

Don't forget the real-world use cases for IAM.