AWS Certified Developer Associate

Lesson 2: IAM Overview

simplilearn

What You'll Learn

- Describe an overview of Identity and Access Management - IAM
- Define the elements of IAM: Identities, Polices, and Resources
- Identify IAM features and key terms
- Describe IAM's Policy Document
- Analyze a Sample JSON Script
- List IAM Best Practices and Important Facts

Basic Concepts of IAM

What is IAM?

IAM provides two main elements of secure access control:

Authentication

Authorization

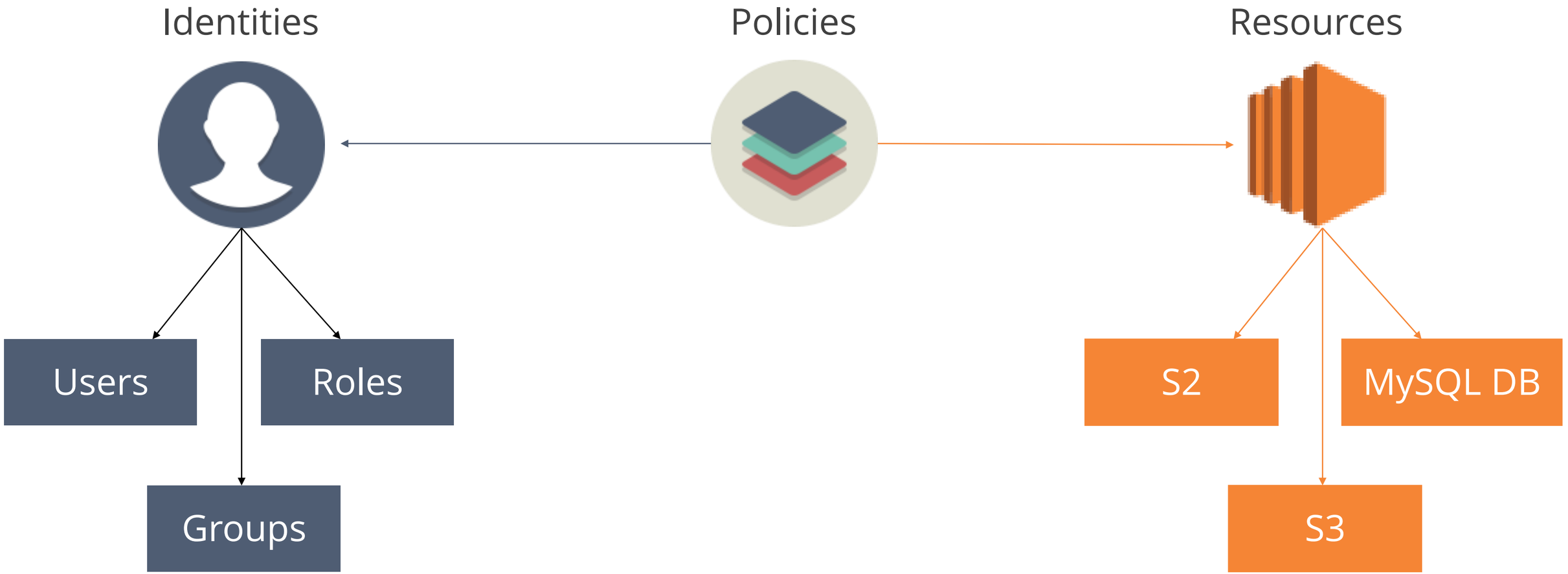
Amazon Web Services ecosystem is mainly built on three basic elements:

Identities

Policies

Resources

Elements of IAM



IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Central Control

Central Control is a single, centralized place to control and manage IAM users, groups, and permissions of various AWS resources.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Consolidated Billing

You can manage multiple AWS accounts from the cost perspective, and obtain a combined view of AWS charges incurred by all your child accounts.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Shared Access

Shared Access allows other users in your organization to perform administrative functions on your AWS resources without sharing your login credentials.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Granular Permissions

Granular Permissions lets you manage various permission levels for different resources. For example, you can provide S3 read and write permissions to a specific user, but not allow them to delete any object. Simultaneously, you can provide delete permissions to another user, and not provide them with read and write permissions.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Specified Networks

AWS heavily uses VPC, an isolated private network in the cloud. You may need to create multiple VPCs for your organization and provide access to only specific users. IAM is used for such access-control policies.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Management of Security Credentials

You may need to use different types of security credentials depending on the AWS resources that you are trying to access. These credentials, including access keys and key pairs, are managed by IAM.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Multifactor Authentication

Multifactor Authentication provides an additional security layer to your AWS account. You can set up and enable MFA using the IAM dashboard in the AWS management console.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

Central Control

Consolidated Billing

Shared Access

Granular Permissions

Specified Networks

Temporary Access

You can provide temporary access to users, services, and devices on a need-basis using IAM's security token service. These credentials are provided to users on short-term and expire after a specific time.

Management Security Credentials

Multifactor Authentication

Temporary Access

Identify Federation

Support Roles

IAM Features

- Central Control
- Consolidated Billing
- Shared Access
- Granular Permissions
- Specified Networks

Identity Federation

IAM provides identity federation for active directories, such as Facebook, by use of SAML for single sign-on. This function allows you to manage identities outside of AWS and grants permission or access to use AWS resources using those identities.

- Management Security Credentials
- Multifactor Authentication
- Temporary Access
- Identify Federation
- Support Roles

IAM Features

Central Control

Consolidated Billing

Shared Access

Granular Permissions

Specified Networks

Roles Support

IAM supports roles that can be consumed by any user or resource. You can use roles to delegate access controls to users, applications or resources. You will learn more about this in the next few slides.

Management Security Credentials

Multifactor Authentication

Temporary Access

Identify Federation

Roles Support

Knowledge Check

KNOWLEDGE CHECK

Which is not an element of IAM?

a. Identities

b. Regions

c. Data Center

d. Policies

KNOWLEDGE CHECK

Which is not an element of IAM?

a. Identities

b. Regions

c. Data Center

d. Policies

The correct answer is **Regions and Data Center**

Explanation : Here are the three basic elements of IAM: Identities, policies, and resources.

IAM Terminology

IAM Key Terms: Users & Groups

USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS

- A user is an entity created in IAM. An IAM user can be an individual, system, or application requiring access to AWS services.
- A user will require a name and password to sign into the AWS management console, but it will need up to two access keys which will be required to sign into AWS using the API or CLI.
- AWS resources for users managed outside of AWS in your corporate directory are referred to as "federated users."

IAM Key Terms: Users & Groups

USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS

- A group is a collection of IAM users combined together. You can use groups when you need to specify permissions for a collection of users. It becomes easier to manage these users together.

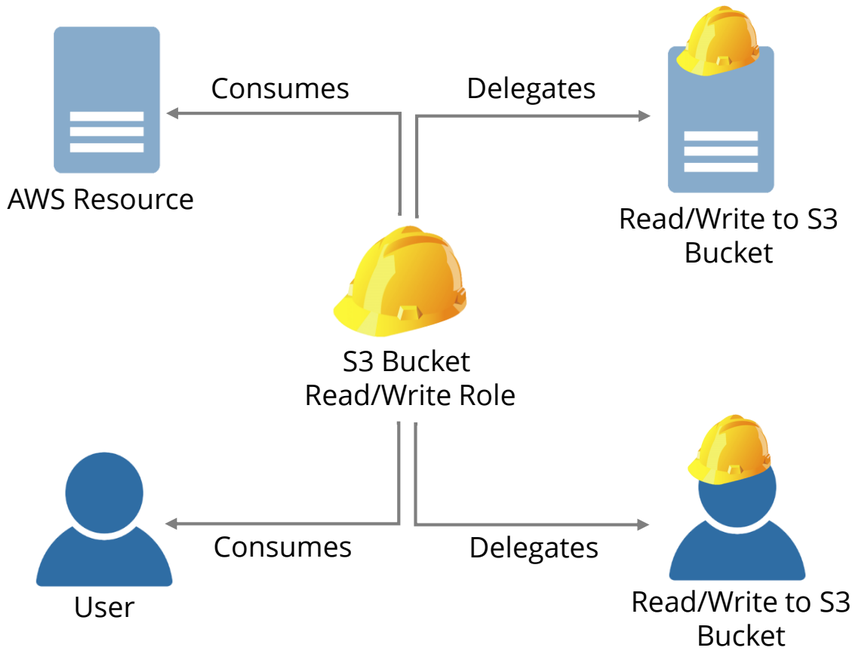
- If a new user joins the group all privileges and permissions assigned to that group is automatically assigned and available for the new user.

IAM Key Terms: Users & Groups

USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS

Roles

- Roles work in a similar way to users
- It is an identity by permission with policies attached to it
- It is intended to be assumable by anyone who needs it



IAM Key Terms: Users & Groups

USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS

Temporary credentials

- Is used with IAM roles
- Has more restricted set of permissions than any standard IAM user
- Protects from accidentally performing unauthorized tasks or limiting user access

IAM Key Terms: Users & Groups

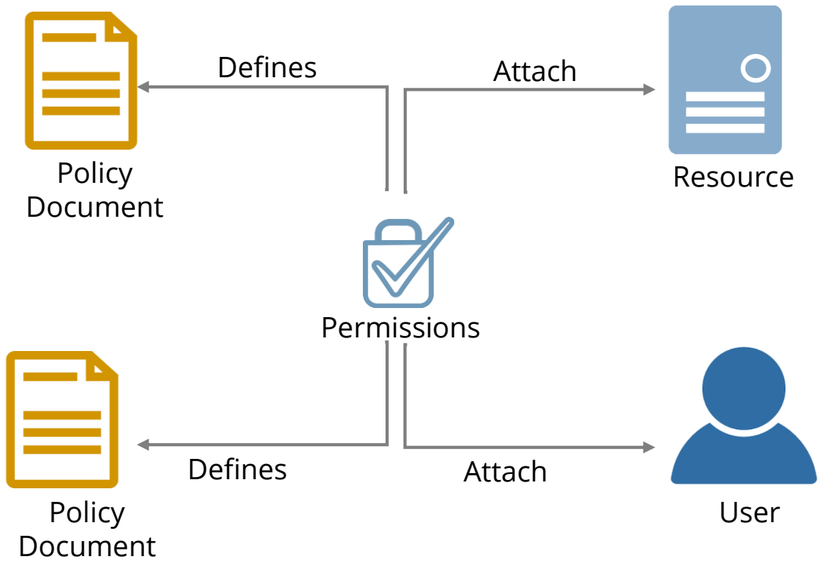
USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS

Policies

- Assigns specific permissions to a user, group, role, or resource
- Documents explicit lists of permissions and their access levels
- Defines permissions

IAM Key Terms: Users & Groups

USERS | GROUPS | ROLES | CREDENTIALS | POLICIES | PERMISSIONS



Permissions

- Specifies extent of access to a user for AWS resources, and actions that can be performed on those resources
- IAM users have NO permissions by default

JSON Policy Document

```
{
        "Version": "2014-11-13",
        "Statement":
        { "Effect": "Allow",
          "Action": "s3:ListBucket",
          "Resource": "arn:aws:s3:::simplylearn_bucket"
        }
}
"Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-
WITHOUT-HYPHENS:root"]},
```

Knowledge Check

KNOWLEDGE CHECK

What are the differences between IAM user and role?

a. Role is intended to be assumable by anyone who needs it

b. User is intended to be assumable by anyone who needs it

c. User can be attached to AWS Resources

d. Role can be attached to AWS Resources

KNOWLEDGE CHECK

What are the differences between IAM user and role?

a. Role is intended to be assumable by anyone who needs it

b. User is intended to be assumable by anyone who needs it

c. User can be attached to AWS Resources

d. Role can be attached to AWS Resources

The correct answer is **A and D**

Explanation : Users, as identities, are owned by a single individual, but Roles are intended to be consumed by anyone who needs them. However, roles can't act by themselves and should be applied to a user, group, or resource to consume their capabilities.

IAM Best Practices and Important Facts

IAM Best Practices

1

2

3

4

5

6

7

IAM Best Practices

1

Do not use access keys of the root account

Root account has full permissions to all your AWS resources including billing and payment

2

Avoid using root account

Create individual IAM users for people accessing the root account

3

Use IAM role with EC2 instance

Instead of using or storing access keys, attach a role when you launch an EC2 instance

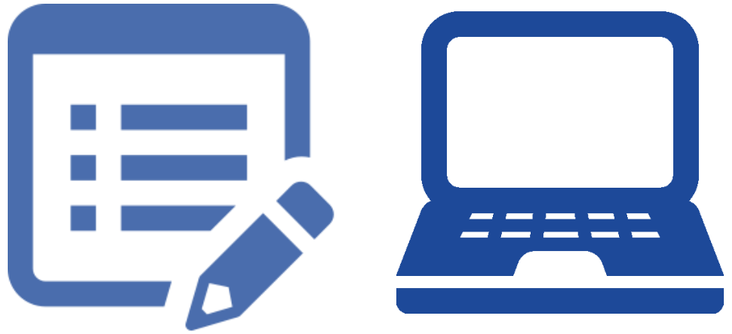
IAM Best Practices

4

Use groups to manage user permissions

Create a group and assign a set of permissions collectively, instead of assigning them to individual users

A user will initially always have an explicit "deny" for all AWS services



5

Enable multifactor authentication

Multifactor authentication adds an extra layer of protection, on top of your user name and password

Multifactor provides increased security, for your AWS account access and resources

IAM Best Practices

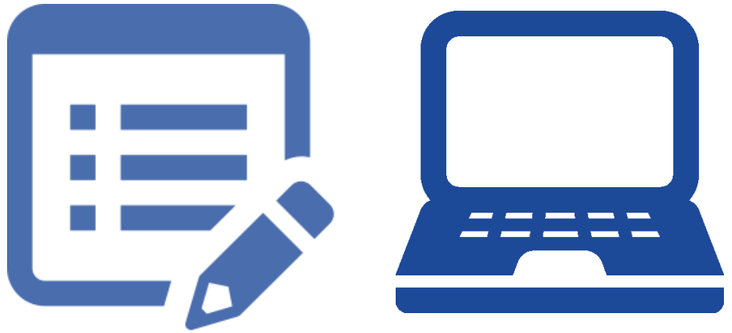
6

Make use of password policies

Change your AWS account passwords and access keys regularly

Enforce all IAM users to do the same

An IAM user can have only one valid password at any given time



7

Grant least privilege

Provide minimum set of permissions needed to perform an task or work

IAM – Important Facts

Users are global entities

New users have NO permissions

Root account has admin access by default

The best practice of using the root user is only to create your first IAM user

Access keys are not to be used for console login

Use IAM access keys instead of AWS root account access keys

IAM lets you securely control access to AWS services and resources

Validate policy documents created by JSON using policy validator

Knowledge Check

KNOWLEDGE CHECK

When you create a new IAM user, what permissions does he or she have?

a. EC2 access

b. Admin

c. S3 access

d. None

KNOWLEDGE CHECK

When you create a new IAM user, what permissions does he or she have?

a. EC2 access

b. Admin

c. S3 access

d. None

The correct answer is **d**

Explanation : Every IAM user starts with no permissions. By default, users can do nothing, not even view their own access keys.

Access Keys and Security Credentials

Access Keys – Deep Dive

- Used when accessing AWS resources using AWS CLI or SDK
- When an administrator creates a new IAM user, IAM generates access keys
- Access key is only available when you create a new IAM user
- It is a combination of the access key ID and secret access key
- Accessible only when created
- Deleted access keys can’t be retrieved
- Each user can have two access keys that can be rotated for higher security

Temporary Security Credentials

These credentials are valid for a specified duration with a specified set of permissions

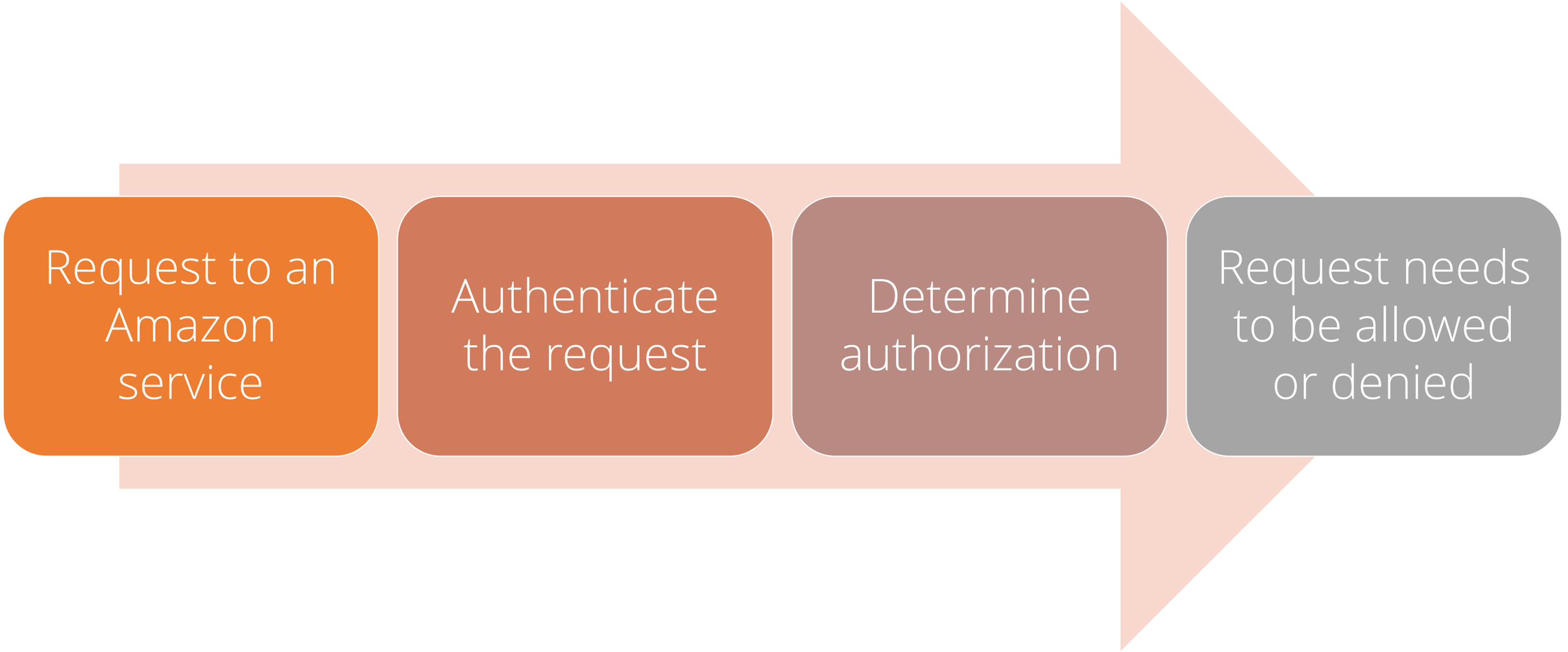
Distributing long-term access keys with application isn't necessary

Provide access to your AWS resources without using IAM user identities

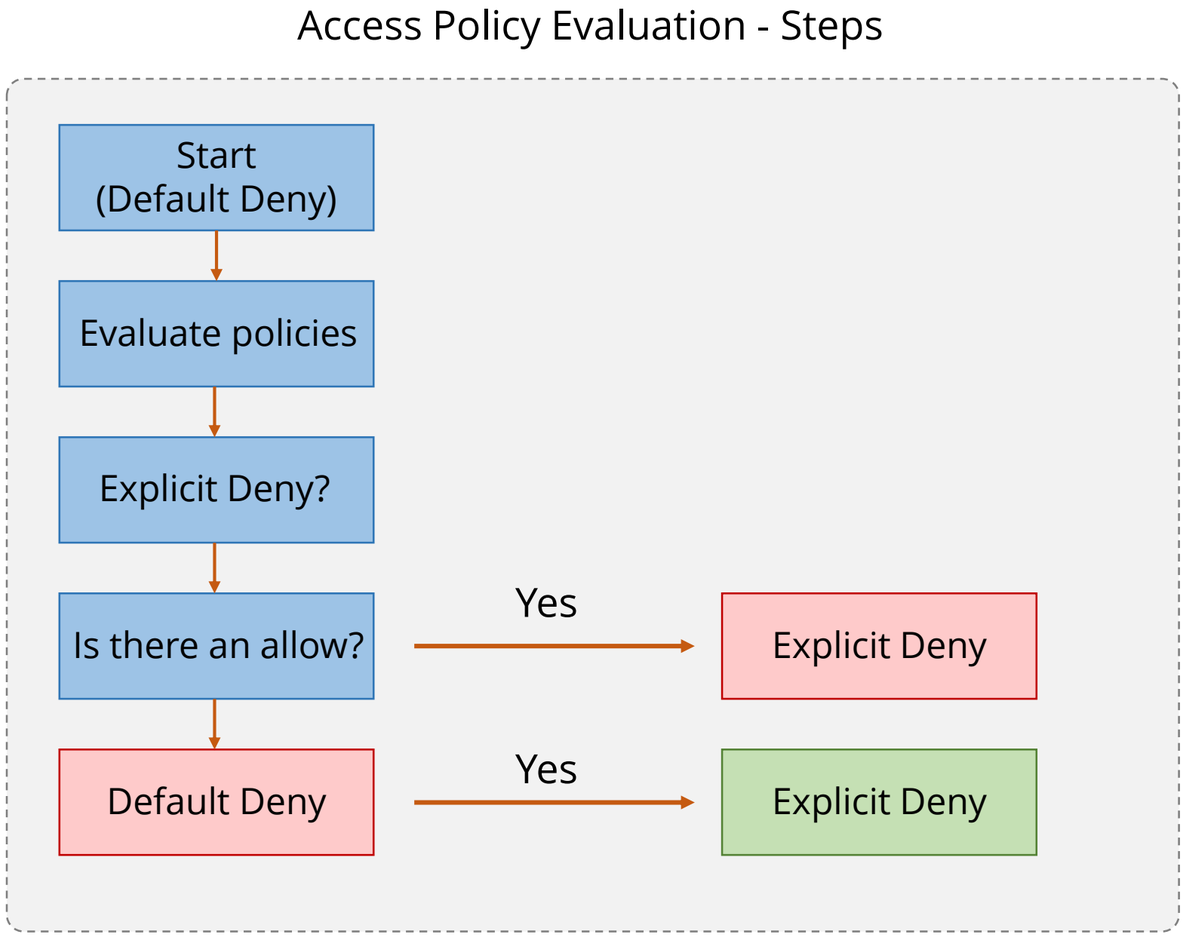
Rotating credentials isn't necessary since they expire and can't be reused

Used by API's such as GetFederationToken, AssumeRole, AssumeRoleWithSAML, or AssumeRoleWithWebIdentity

AWS Access Policy Evaluation



AWS Access Policy Evaluation (Cont.)



Knowledge Check

KNOWLEDGE CHECK

What are the advantages of temporary credentials?

a. No need to rotate

b. Can be used to create AWS accounts

c. They expire after 90 days

d. Provide access to your AWS resources without IAM user

KNOWLEDGE CHECK

What are the advantages of temporary credentials?

a. No need to rotate

b. Can be used to create AWS accounts

c. They expire after 90 days

d. Provide access to your AWS resources without IAM user

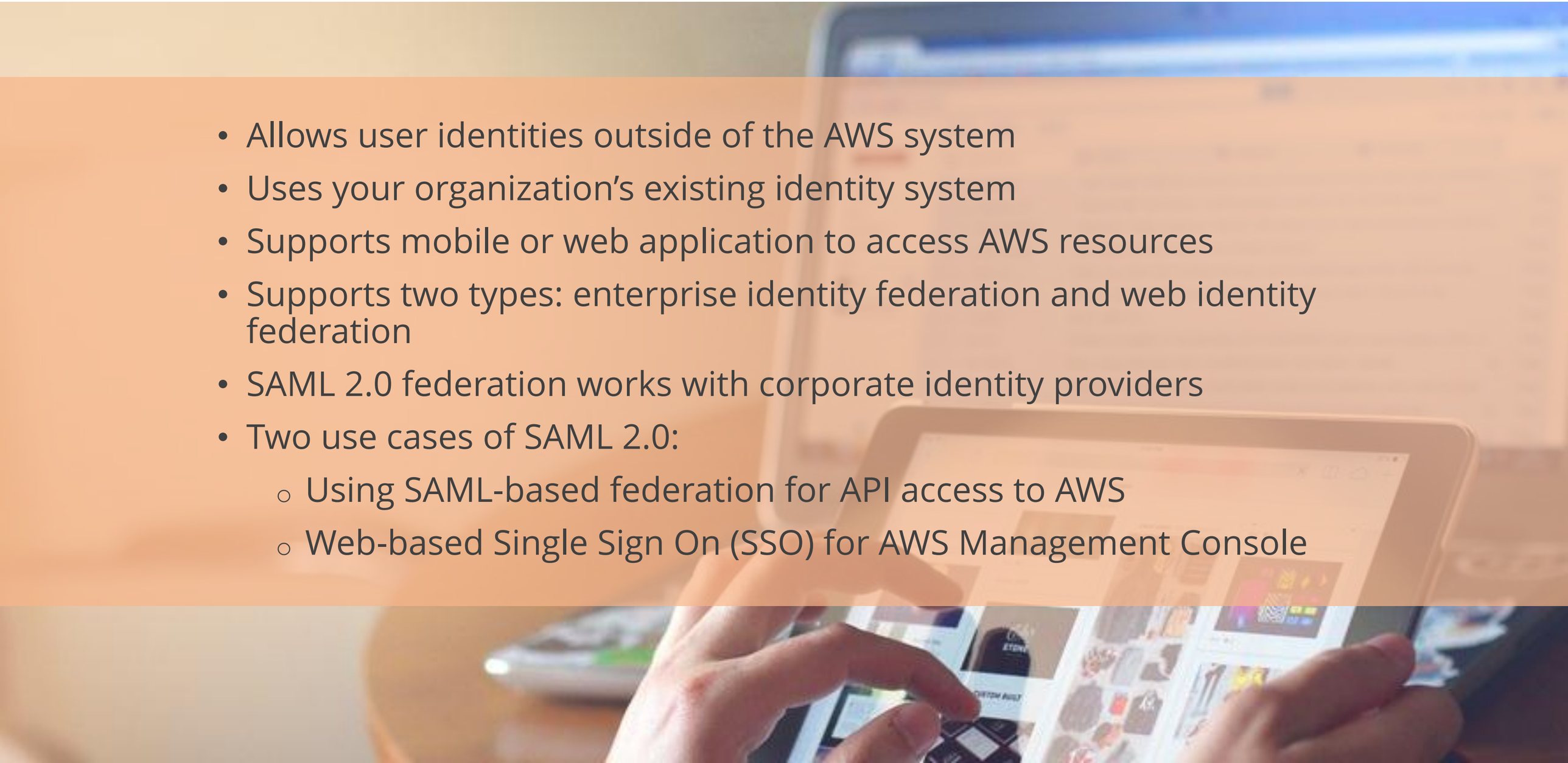
The correct answer is **Provide access to your AWS resources without IAM user and No need to rotate**

Explanation : Some of the advantages of using temporary credentials are:

1) No need to distribute long-term access keys with application. Instead you can use temporary credentials. 2) It provides access to your AWS resources without using IAM user identities, and 3) It doesn't require you to rotate credentials since they expire after a defined duration and can't be reused.

AWS Identity Federation

AWS Identity Federation

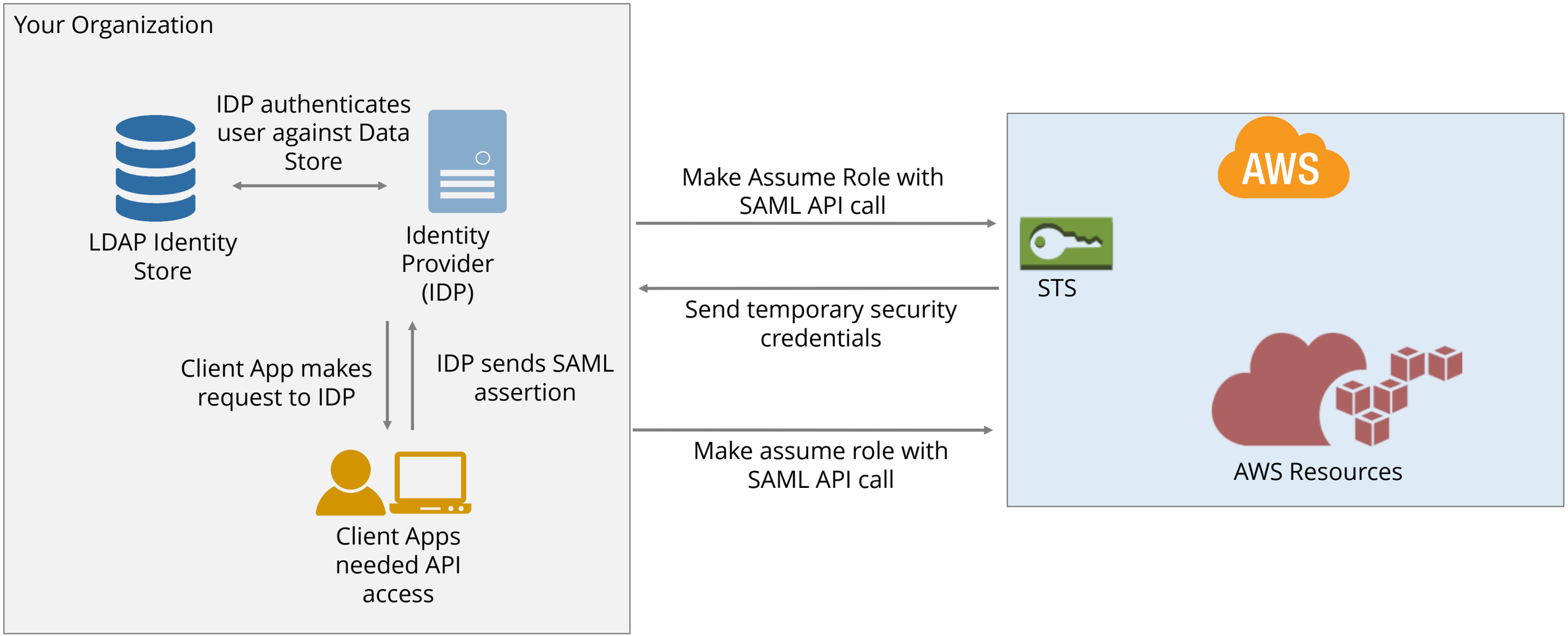


- Allows user identities outside of the AWS system
- Uses your organization's existing identity system
- Supports mobile or web application to access AWS resources
- Supports two types: enterprise identity federation and web identity federation
- SAML 2.0 federation works with corporate identity providers
- Two use cases of SAML 2.0:
  - Using SAML-based federation for API access to AWS
  - Web-based Single Sign On (SSO) for AWS Management Console

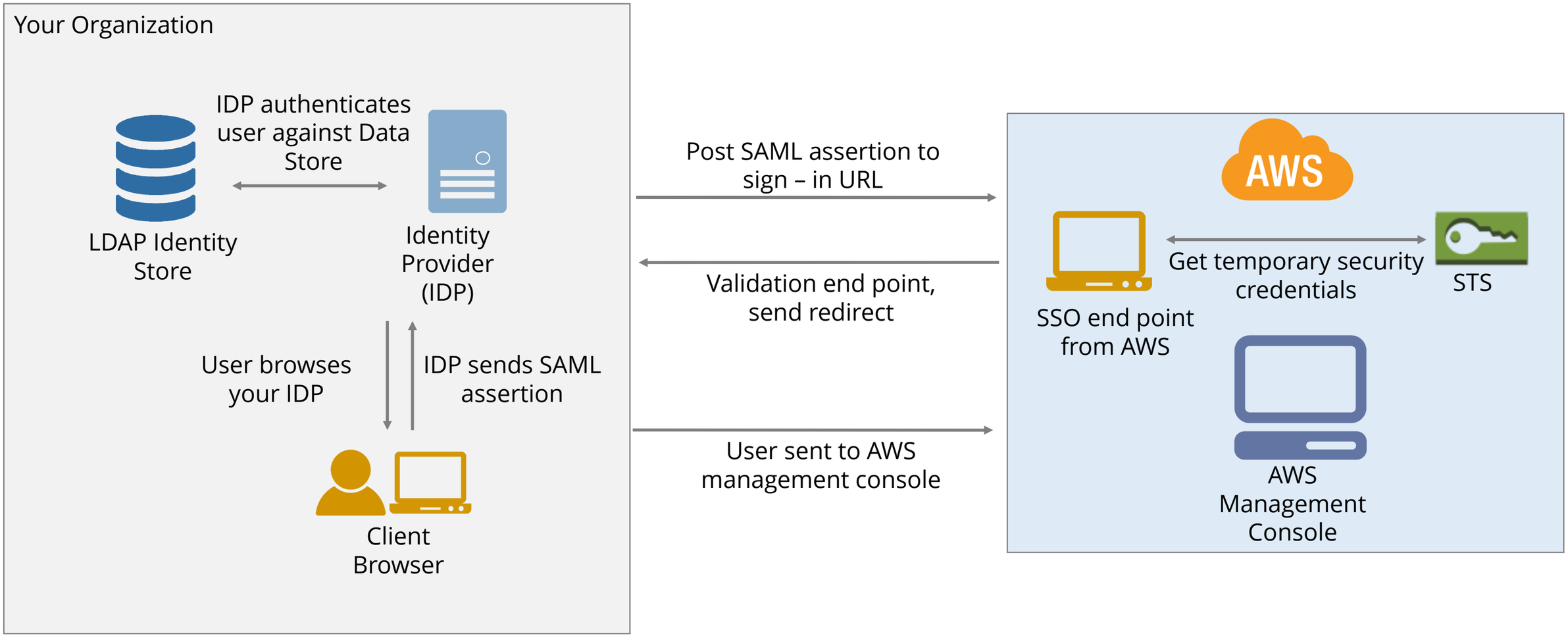
Web Identity Federation

- Uses third-party identity or any OpenID Connect (OIDC) 2.0 compatible providers
- Signup and configure your application with IdP
- Each IdP gives you an app ID that's unique to that IdP
- Create an entity in IAM for the selected identity provider
- Create an IAM role and define who can assume it with what permissions

Using SAML-Based Federation for API Access to AWS



Web Based Single Sign On (SSO) to AWS Management Console



Knowledge Check

KNOWLEDGE CHECK

Which of the following allow user identities outside of AWS system to access the resources of your AWS account?

a. Microsoft identity federation

b. Enterprise identity federation

c. Web identity federation

d. Office identity federation

KNOWLEDGE CHECK

Which of the following allow user identities outside of AWS system to access the resources of your AWS account?

a. Microsoft identity federation

b. Enterprise identity federation

c. Web identity federation

d. Office identity federation

The correct answer is **Enterprise identity federation & Web identity federation**

Explanation : Identity federation supports two types of external entities: enterprise identity federation, where those entities reside inside your data center, and web identity federation, where third-party identity providers, such as login with Amazon, Facebook, Google, or any OpenID Connect (OIDC) 2.0 compatible providers, are used.

Practice Assignment:

Use IAM roles to allow users to access resources from different AWS accounts.

Use IAM roles to access resources from different AWS accounts.

You have a Web-based social media company. It has multiple AWS accounts, you need to setup a role that will enable you to access resources from your company's different accounts. This will help your users to share resources with different AWS accounts. This can be done by setting up a cross-account access, which will eliminate the need for setting up individual IAM users in each account.

Prerequisites:

- Two separate AWS accounts have already been created: 1. Test account, and 2. Production account.
- Two users and 2 groups have already been created and configured in the test account:

  User: Joy Group: Testing Permissions: user can sign in the test account

  User: Symone Group: Design Permissions: user can sign in the test account
- No user or group is required in the production account
- Amazon S3 bucket is created in production account called "Prodresource."

Task:

To switch users between your company's AWS accounts to access resources from each other's accounts using IAM role.

.

Quiz

QUIZ

1

Which one is not an IAM identity?

a. User

b. Group

c. Role

d. Account

QUIZ 1

Which one is not an IAM identity?

a. User

b. Group

c. Role

d. Account

The correct answer is **Account**

Explanation: Identities include users, groups, and role.

QUIZ 2

What language do you use to write a policy document?

a. Java

b. SAML

c. JSON

d. SQL

QUIZ 2

What language do you use to write a policy document?

a. Java

b. SAML

c. JSON

d. SQL

The correct answer is **JSON**

Explanation: Policies are documents created using JSON. JSON - JavaScript Object Notation is a lightweight data-interchange format.

QUIZ 3	Can you use access keys to log into AWS console?

a. Yes, you can use access keys to login to AWS console

b. No, you need a user name and password

QUIZ 3	Can you use access keys to log into AWS console?

a. Yes, you can use access keys to login to AWS console

b. No, you need a user name and password

The correct answer is **No, you need a user name and password**

Explanation: Access keys consist of an access key ID and secret access key, which are used to programmatically access AWS services. You can't use them to log into AWS management console.

QUIZ 4	True or False: By default, all requests are denied during access policy evaluation.

a. True

b. False

QUIZ

4

True or False: By default, all requests are denied during access policy evaluation.

a. True

b. False

The correct answer is **True**

Explanation: By default, all requests are denied during access policy evaluation.

QUIZ 5

Which is the API call that you use in SAML based federation?

a. AssumeRole

b. AssumeSAMLwithRole

c. ConsumeRoleWithSAML

d. AssumeRoleWithSAML

QUIZ

5

Which is the API call that you use in SAML based federation?

a. AssumeRole

b. AssumeSAMLwithRole

c. ConsumeRoleWithSAML

d. AssumeRoleWithSAML

The correct answer is **AssumeRoleWithSAML**

Explanation: In the SAML-based federation, client app calls the AWS STS AssumeRoleWithSAML API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.

QUIZ 6

Does AssumeRoleWithWebIdentity API work with any of the following?

a. Direct Connect

b. ADFS

c. Facebook

d. AD Connect

QUIZ 6

Does AssumeRoleWithWebIdentity API work with any of the following?

a. Direct Connect

b. ADFS

c. Facebook

d. AD Connect

The correct answer is **Facebook**

Explanation: Web identity federation uses third-party identity provider, such as login with Amazon, Facebook, Google, or any OpenID Connect (OIDC) 2.0 compatible providers.

Key Takeaways

- Identity and Access Management (IAM) provides Authentication and Authorization support
- IAM has three basic elements: Identities, policies, and resources
- IAM provides identity federation for active directories, Facebook, and other Id providers
- IAM supports roles that can be consumed by any user or resource
- When working with IAM policies an explicit deny will happen by default
- Access keys consist of an access key ID and secret access key, which are used to programmatically access AWS services
- Temporary security credentials are valid for a specified duration with a specified set of permissions

This concludes “IAM Overview.”

The next lesson is “IAM LAB.”