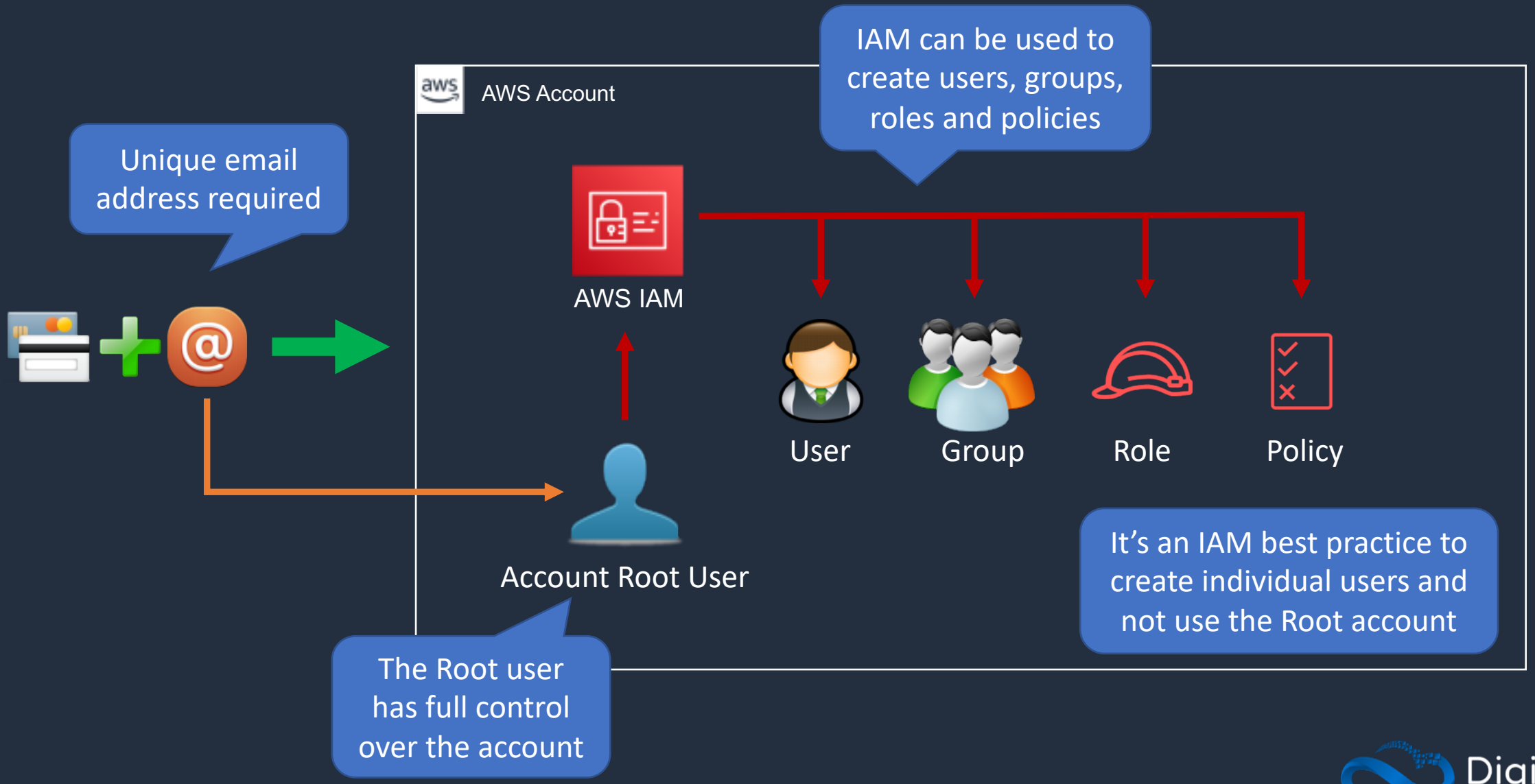
SECTION 2

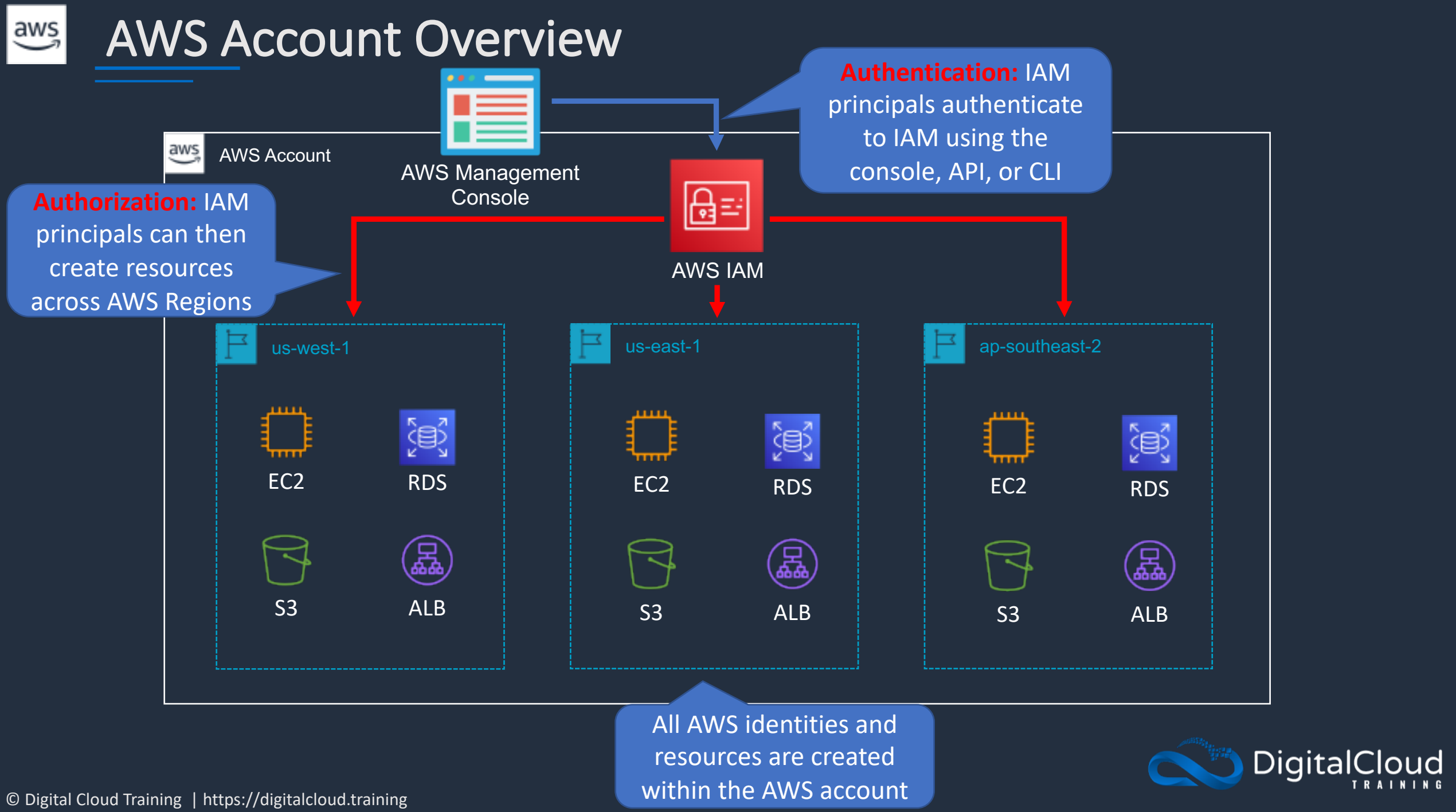
Getting Started - AWS Accounts

AWS Account Overview

AWS Account Overview



AWS Account Overview



DigitalCloud TRAINING

Create Management AWS Account

What you need…

Credit card for setting up the account and paying any bills

Unique email address for this account

Check if you can use an alias with an existing email address (e.g dynamic aliases in Gmail / O365)

john@example.com

john+dctmanagement@example.com

john+dctprod@example.com

AWS account name – mine will be DCT-MANAGEMENT

Phone to receive an SMS verification code

Configure Account and Setup Billing Alarms

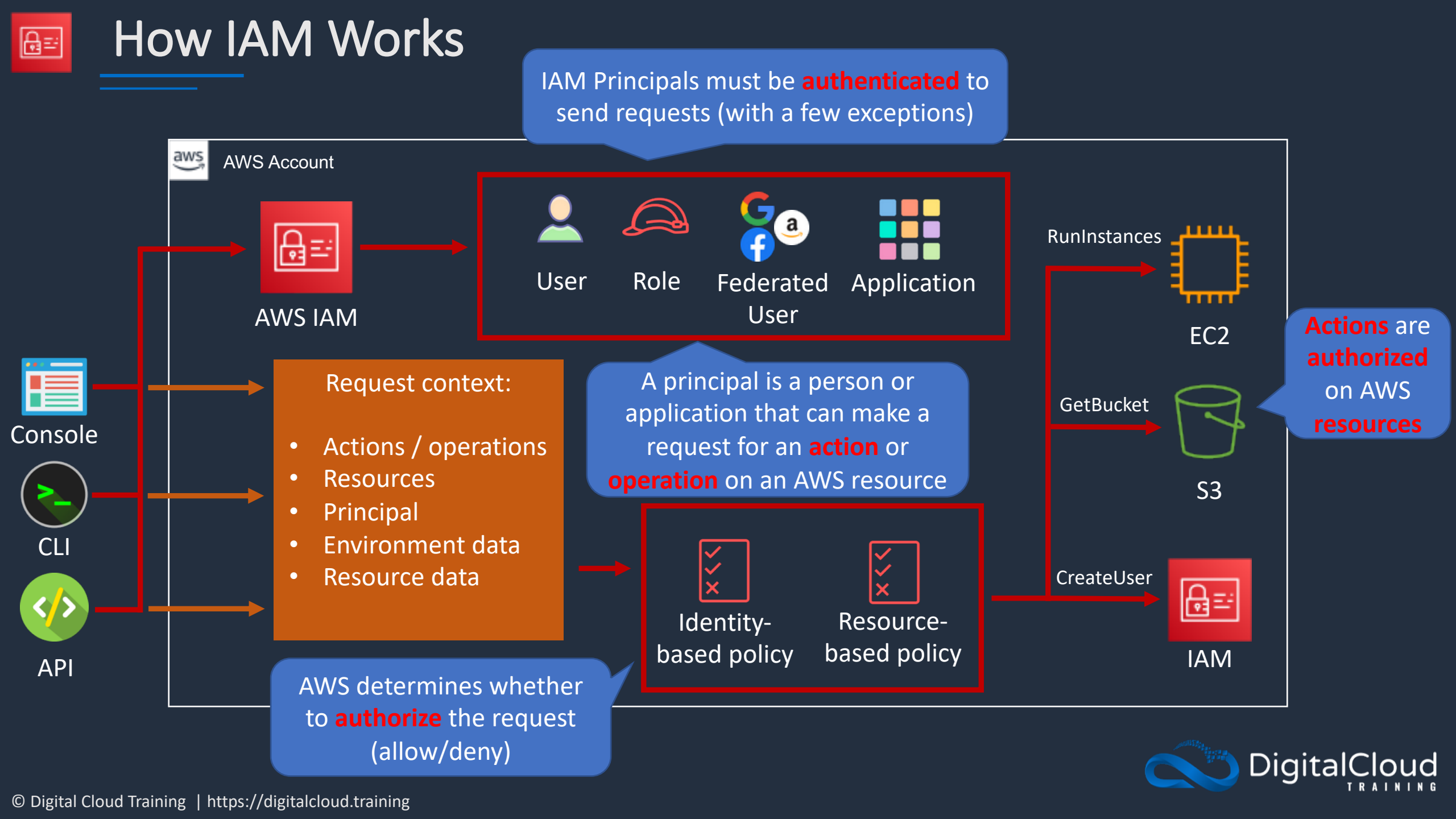
Install Tools (AWS CLI and VS Code)

SECTION 3

AWS IAM Fundamentals

How IAM Works

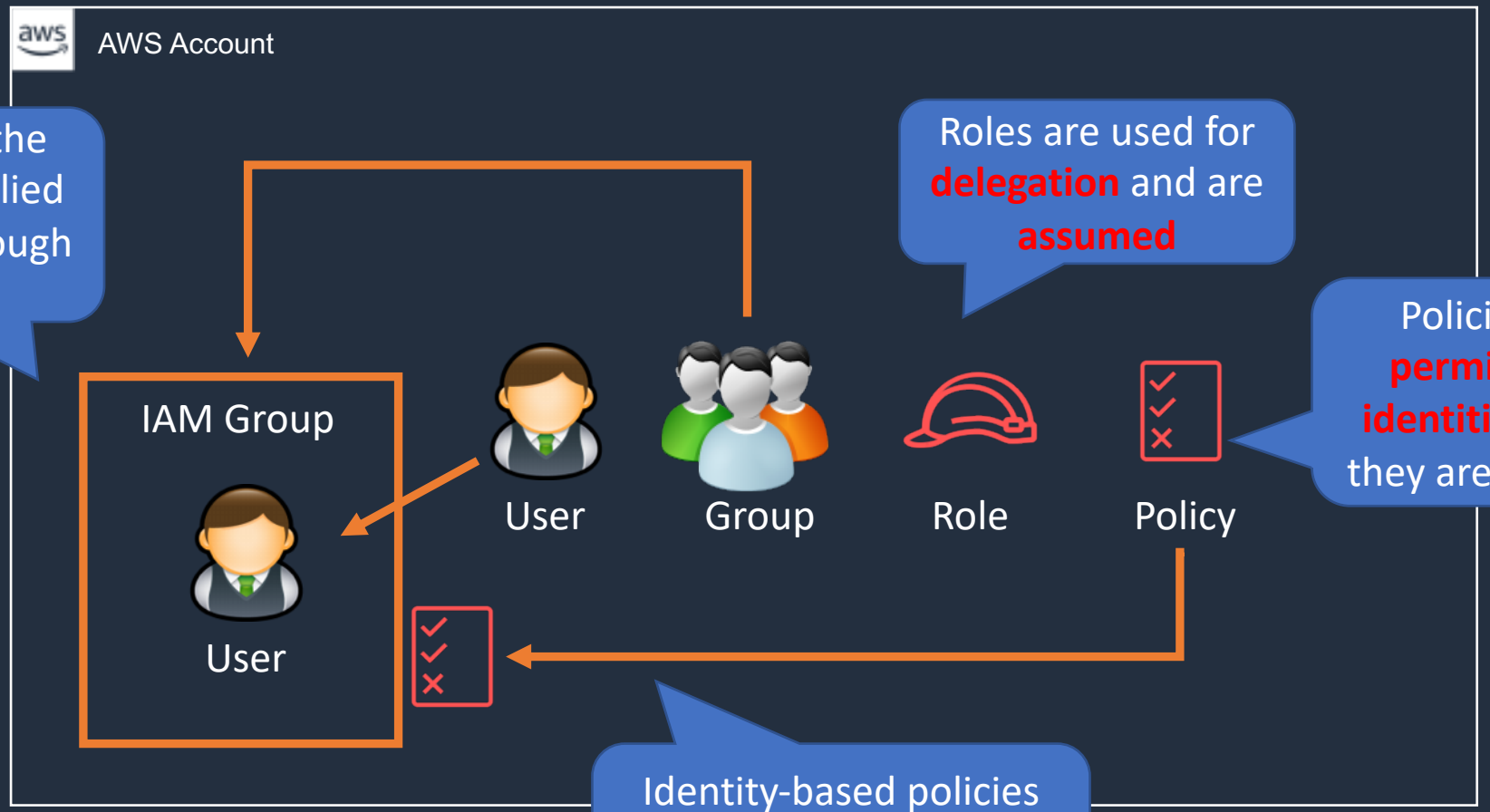
How IAM Works



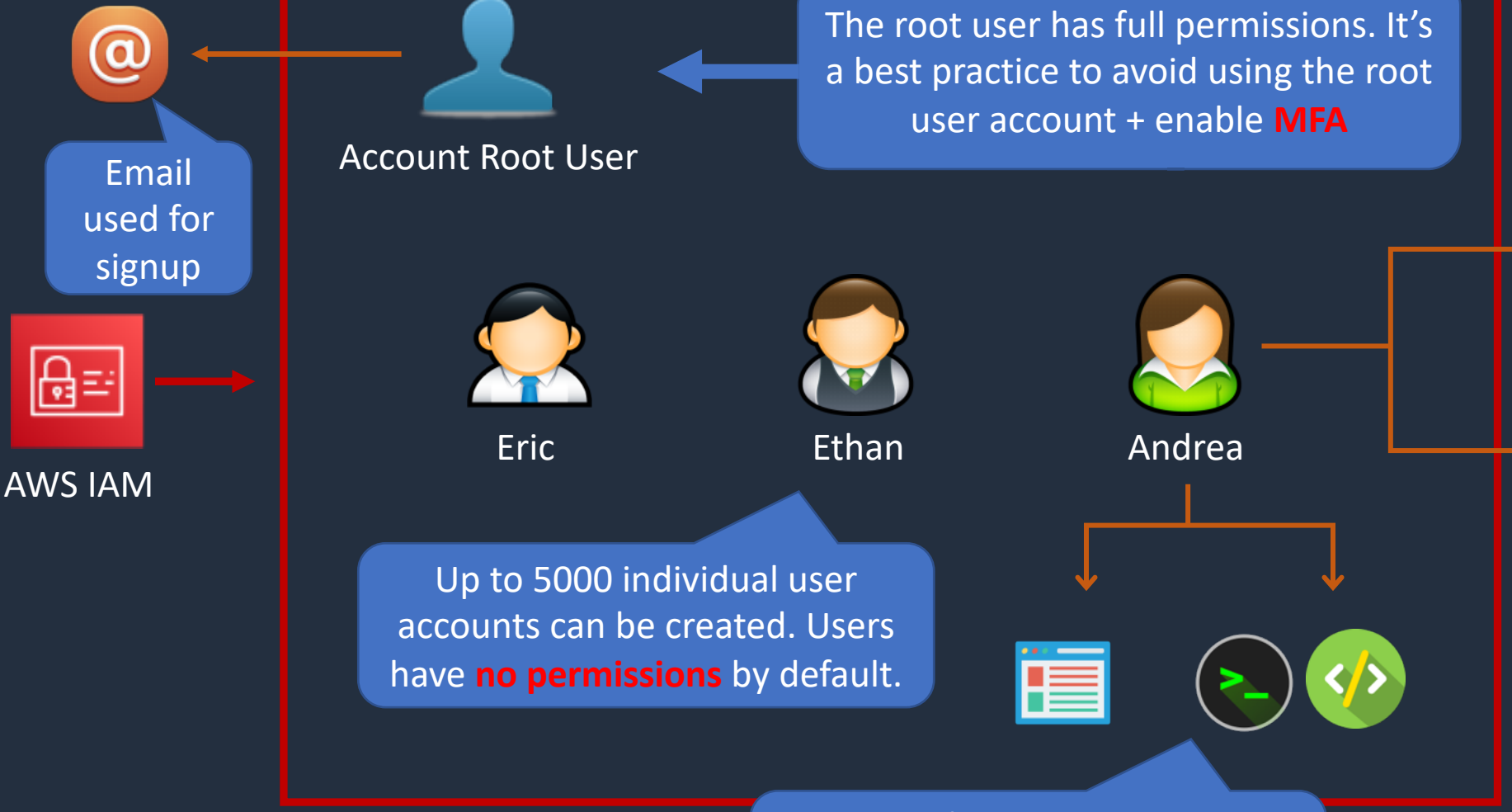
Overview of Users, Groups, Roles and Policies

DigitalCloud TRAINING

Users, Groups, Roles and Policies

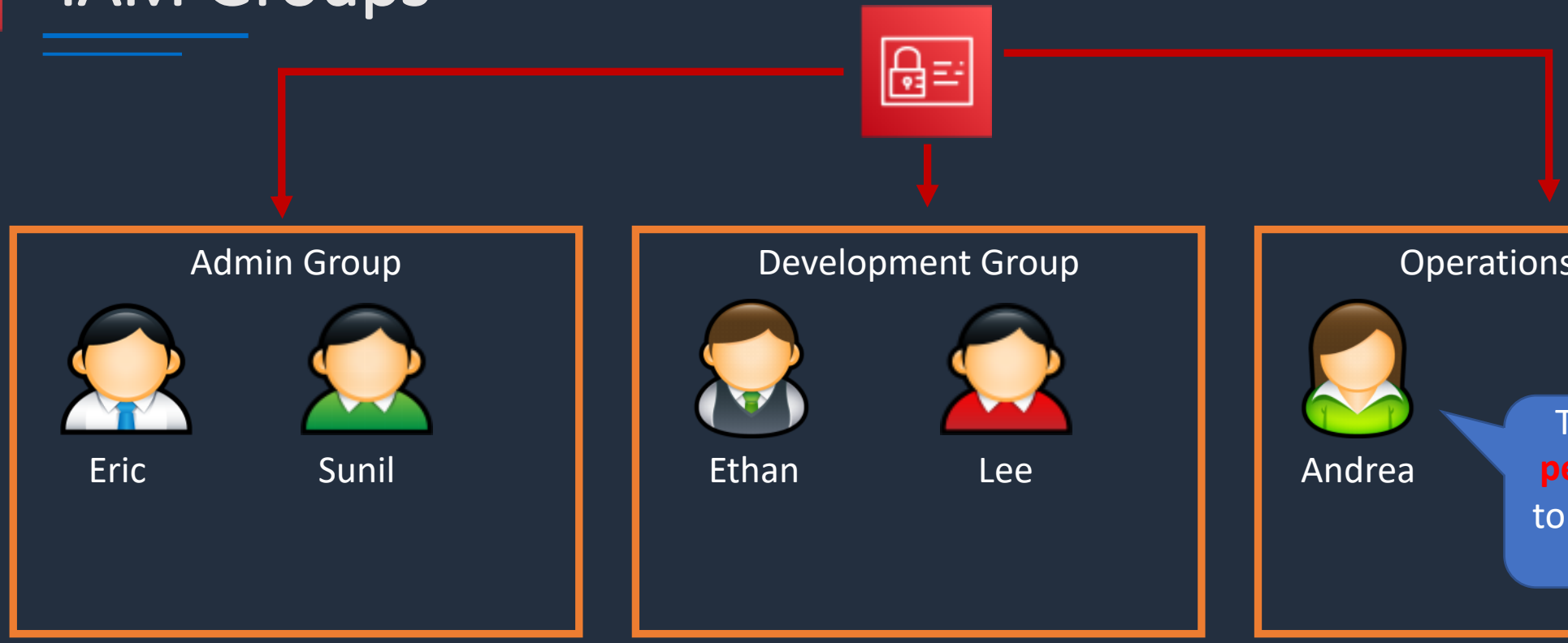


IAM Users



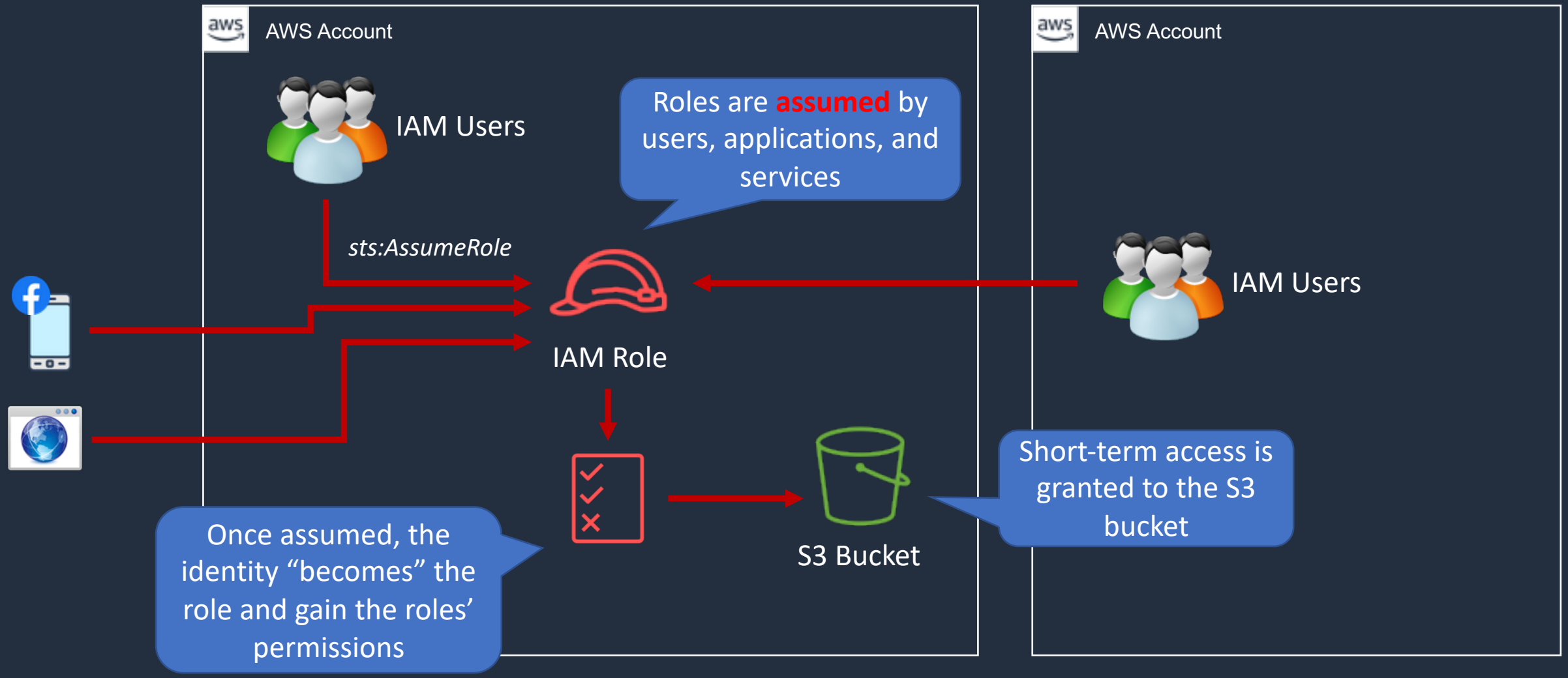
DigitalCloud TRAINING

IAM Groups



IAM Roles

An IAM role is an IAM identity that that has specific permissions



IAM Policies

Policies are documents that define permissions and are written in JSON

AdministratorAccess

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

All permissions are implicitly denied by default

User

Group

Role

Identity-based policies can be applied to users, groups, and roles

Bucket Policy

```
{
    "Version": "2012-10-17",
    "Id": "Policy1561964929358",
    "Statement":[
    {
        "Sid": "Stmt1561964454052",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::515148227241:user/Paul"
        },
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::dctcompany",
        "Condition": {
            "StringLike": {
                "s3:prefix": "Confidential/*"
            }
        }
    }
    ]
}
```

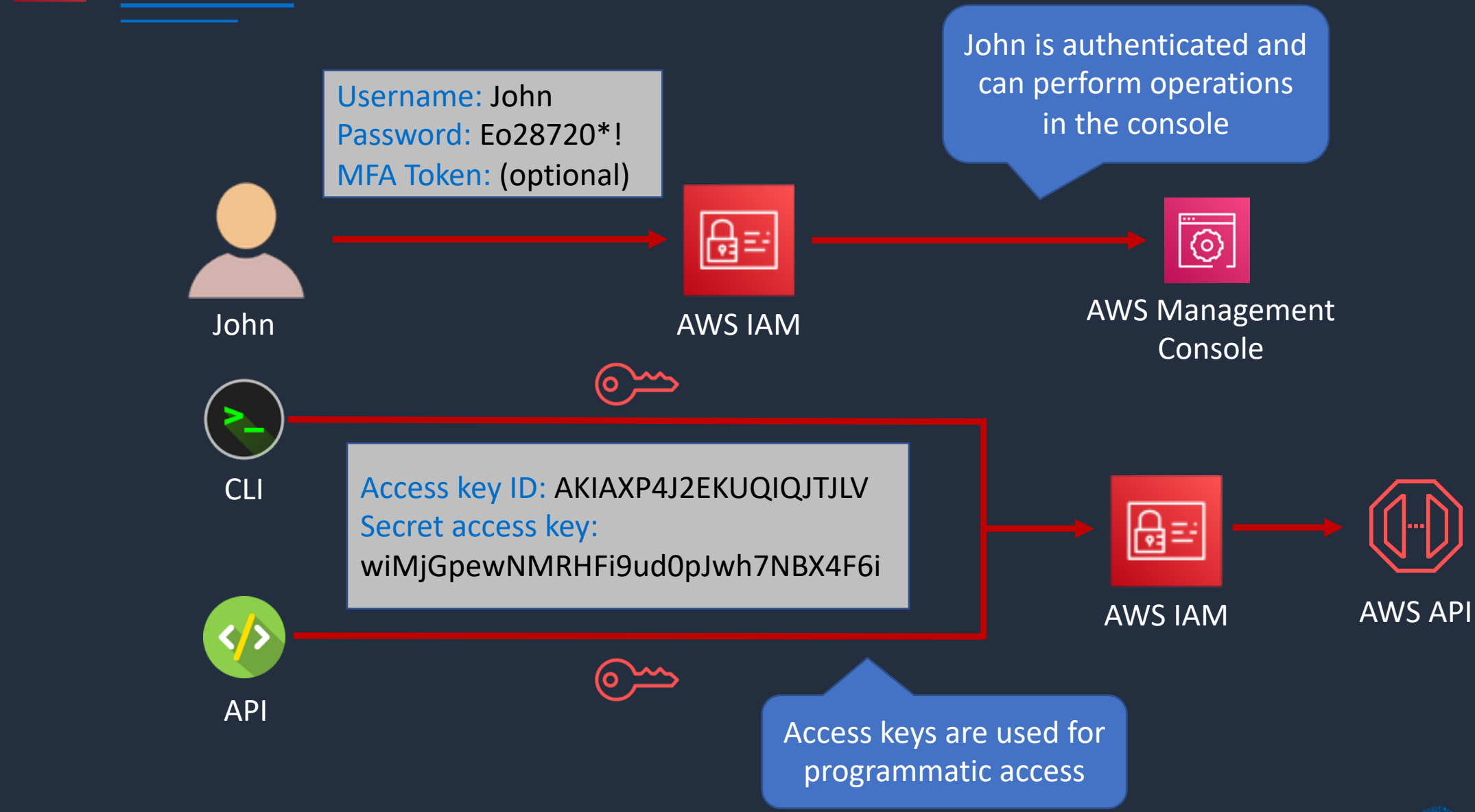
Resource-based policies apply to resources such as S3 buckets or DynamoDB tables

S3 Bucket

DigitalCloud TRAINING

IAM Authentication Methods

IAM Authentication Methods



IAM Authentication Methods

X.509 **certificate authentication** to **some** services. E.g. Amazon EC2 SOAP and CLI interfaces

Signing Certificate

AWS IAM

Amazon EC2

SSH Key / HTTPS Git Credentials

AWS IAM

AWS CodeCommit

Authentication to AWS CodeCommit

Keyspaces Credentials

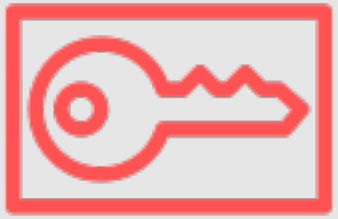
AWS IAM

Amazon Keyspaces

DigitalCloud TRAINING

Create User, Group, and Configure CLI

AWS Security Token Service (STS)



AWS Security Token Service (STS)

Credentials include:
- AccessKeyId
- Expiration
- SecretAccessKey
- SessionToken

EC2 Instance

Application

S3 Bucket

Temporary security credentials are returned

AWS STS

Instance Profile

IAM Role

EC2 attempts to assume role (sts:AssumeRole API call)

Trust policies control who can assume the role

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "ec2.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
```

Trust Policy

Permissions Policy

Temporary credentials are used with identity federation, delegation, cross-account access, and IAM roles

DigitalCloud TRAINING

Multi-Factor Authentication (MFA)

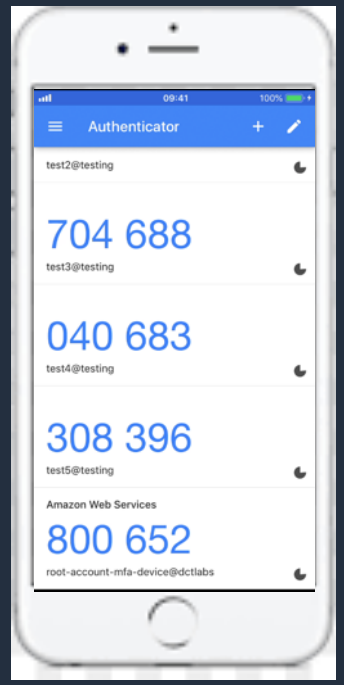
Multi-Factor Authentication

Something you know:

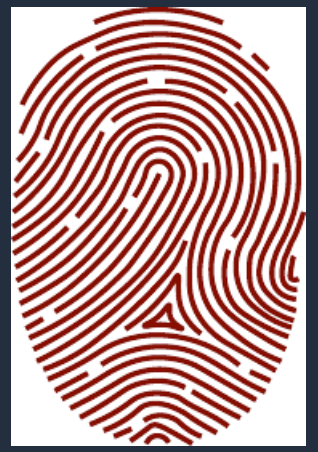
EJPx!*21p9%

Password

Something you have:



Something you are:



DigitalCloud TRAINING

Multi-Factor Authentication

Something you know:

IAM User

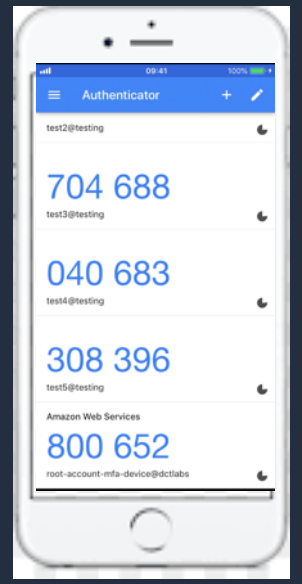
EJPx!*21p9%

Password

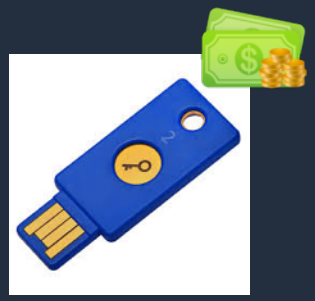
Something you have:

MFA Virtual MFA

e.g. Google Authenticator on your smart phone



MFA Physical MFA



DigitalCloud TRAINING

Secure the AWS Account

SECTION 4

IAM Access Control

Identity-Based Policies and Resource-Based Policies

DigitalCloud TRAINING

Identity-Based IAM Policies

Identity-based policies are JSON permissions policy documents that control what actions an identity can perform, on which resources, and under what conditions

Managed policy. Either AWS managed or customer managed

AWS managed are created and managed by AWS; customer managed are created and managed by you

Managed policies are standalone policies that can be attached to multiple users, groups, or roles

User

Inline policy

Group

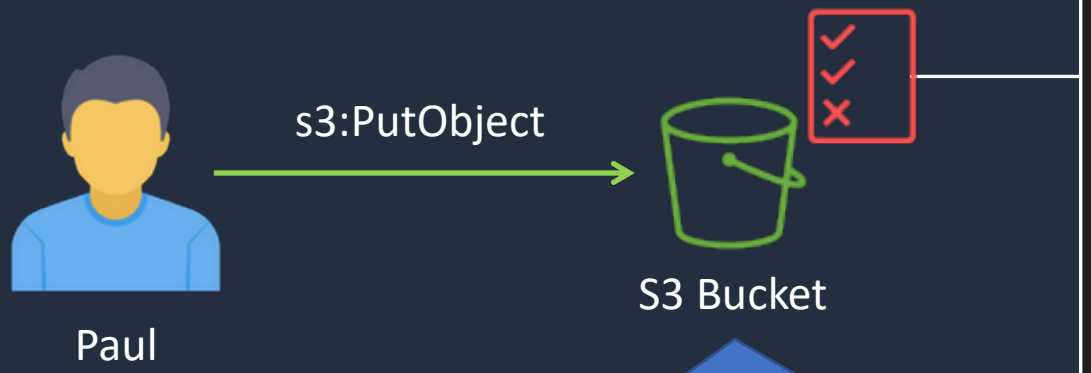
Role

Inline policies have a 1-1 relationship with the user, group, or role

DigitalCloud TRAINING

Resource-Based Policies

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket

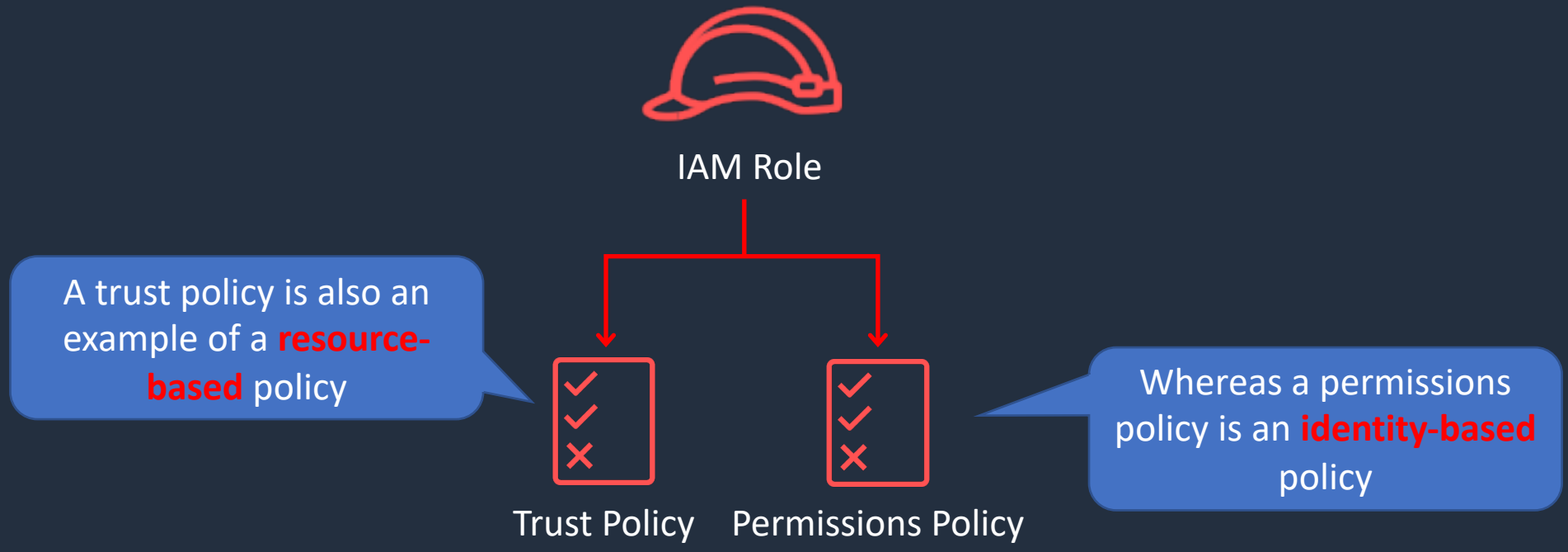



```
{
    "Version": "2012-10-17",
    "Id": "Policy1561964929358",
    "Statement":[
    {
        "Sid": "Stmt1561964454052",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::515148227241:user/Paul"
        },
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::dctcompany"
    }
    ]
}
```

Resource-based policies grant the specified **principal** (Paul) **permission** to perform specific **actions** on the **resource**

DigitalCloud TRAINING

Resource-Based Policies



Access Control Methods - RBAC & ABAC

Role-Based Access Control (RBAC)

Users are assigned permissions through policies attached to groups

Groups are organized by job function

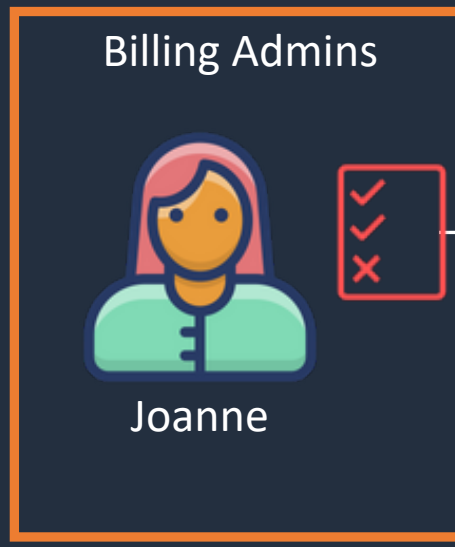
Best practice is to grant the minimum permissions required to perform the job

Role-Based Access Control (RBAC)

Job function policies:

- Administrator
- Billing
- Database administrator
- Data scientist
- Developer power user
- Network administrator
- Security auditor
- Support user
- System administrator
- View-only user

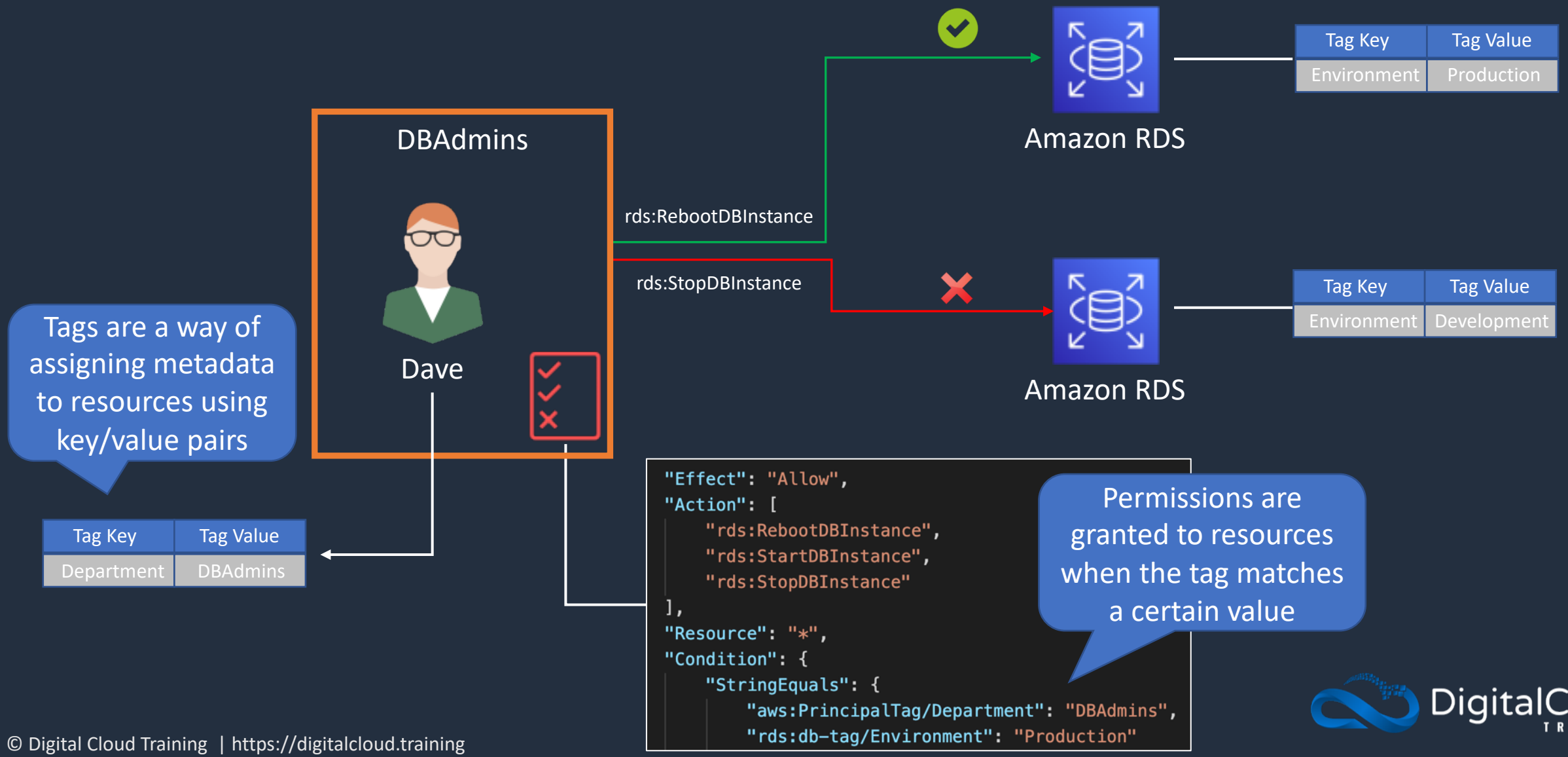
The Billing managed policy is attached to the group




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-portal:*Billing",
                "aws-portal:*Usage",
                "aws-portal:*PaymentMethods",
                "budgets:ViewBudget",
                "budgets:ModifyBudget",
                "ce:UpdatePreferences",
                "ce:CreateReport",
                "ce:UpdateReport",
                "ce:DeleteReport",
                "ce:CreateNotificationSubscription",
                "ce:UpdateNotificationSubscription",
                "ce:DeleteNotificationSubscription",
                "cur:DescribeReportDefinitions",
                "cur:PutReportDefinition",
                "cur:ModifyReportDefinition",
                "cur:DeleteReportDefinition",
                "purchase-orders:*PurchaseOrders"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policies for job functions are designed to closely align to common job functions in the IT industry

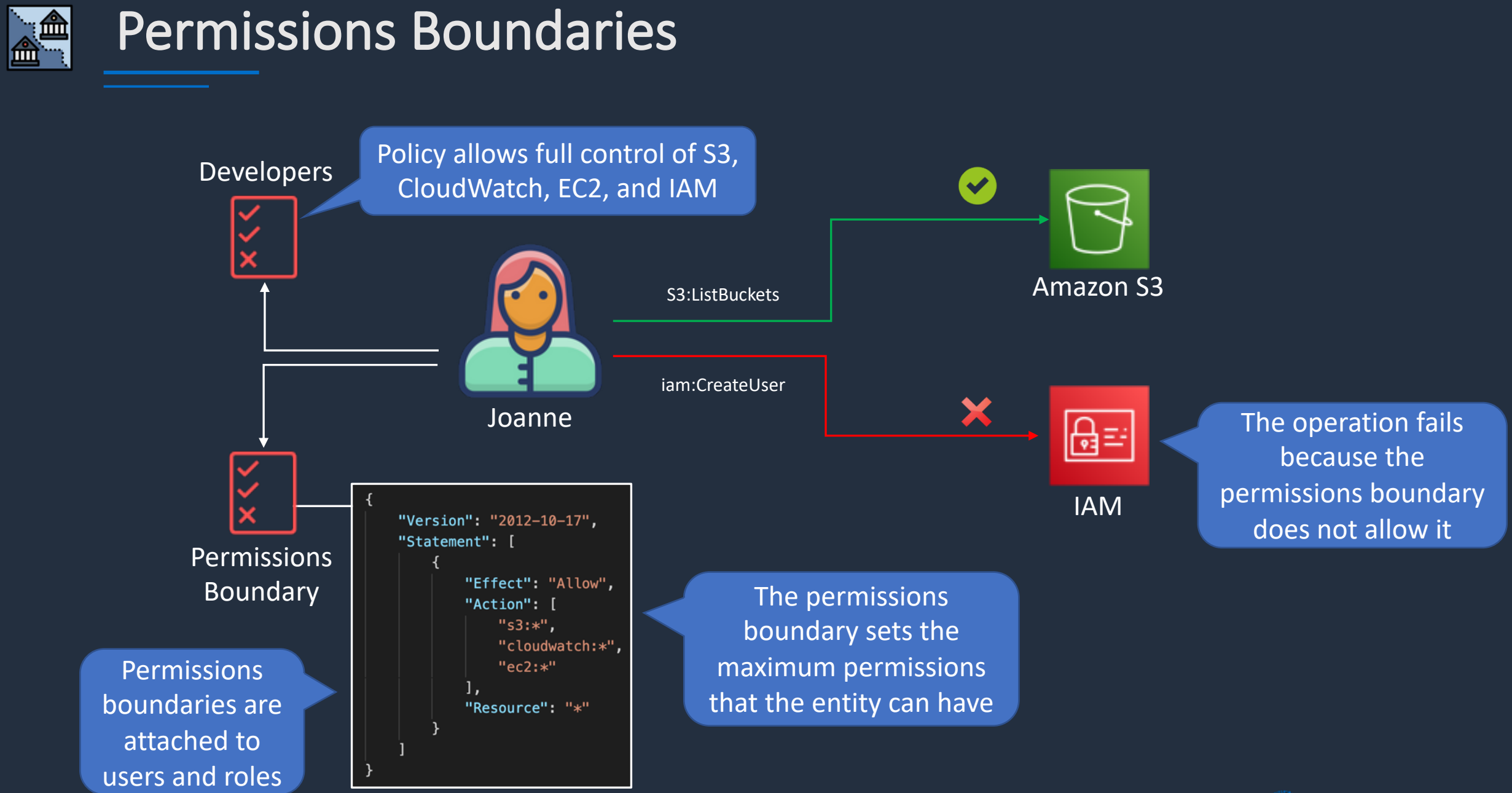
Attribute-Based Access Control (ABAC)



DigitalCloud TRAINING

Permissions Boundaries

Permissions Boundaries



Privilege Escalation

IAMFullAccess

Lindsay is assigned permissions to AWS IAM only and cannot launch AWS resources

Lindsay

iam:CreateUser

IAM

Lindsay applies the AdministratorAccess policy to the X-User account

AdministratorAccess

X-User

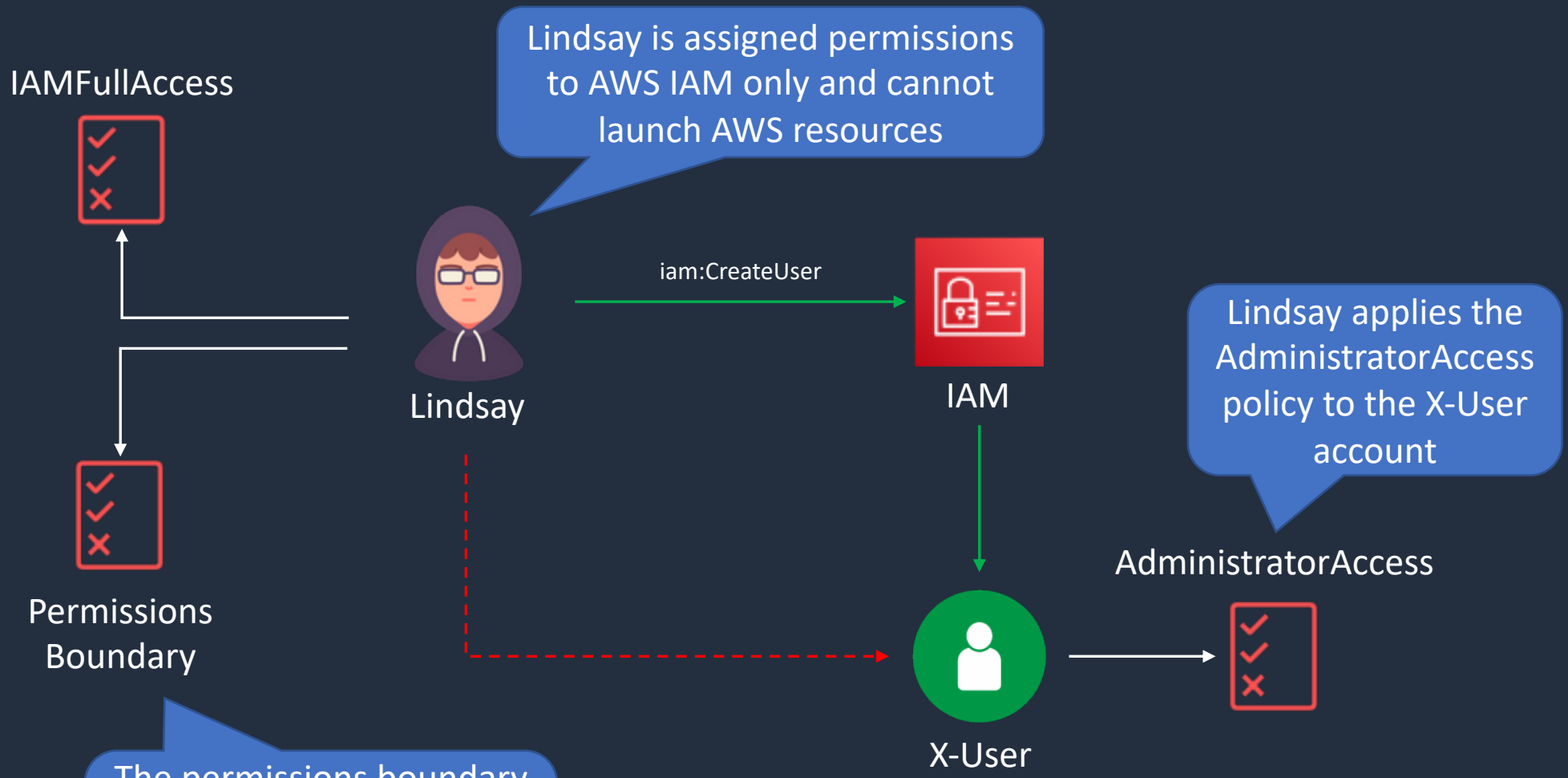
Lindsay is now able to login with the X-User account and gain full privileges to the AWS account

AWS Batch

Lindsay mines bitcoins

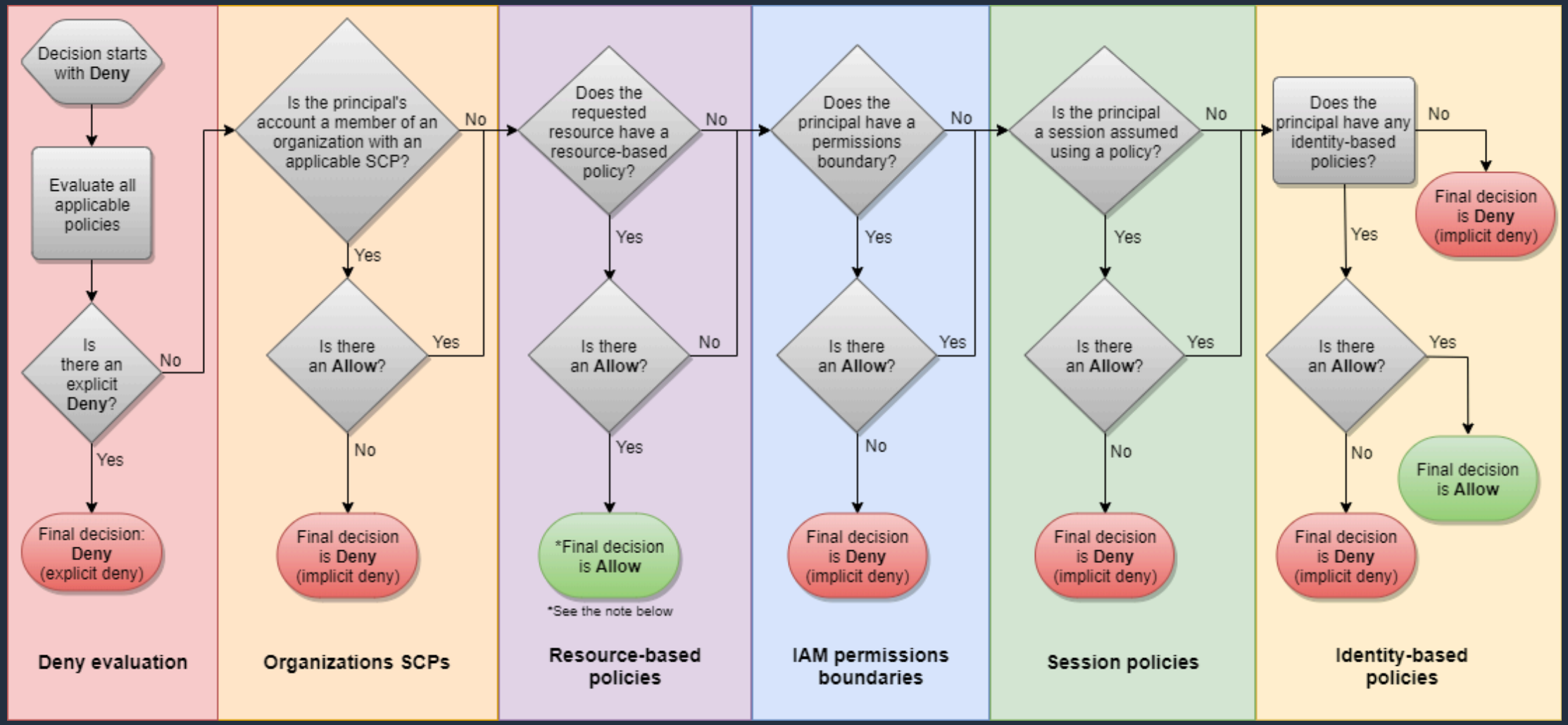
DigitalCloud TRAINING

Preventing Privilege Escalation



IAM Policy Evaluation Logic

Evaluation Logic



Steps for Authorizing Requests to AWS

1. Authentication – AWS authenticates the principal that makes the request

AWS IAM

User

Identity-based policy

3. Evaluating all policies within the account

Resource-based policy

Request context:

- **Actions** – the actions or operations the principal wants to perform
- **Resources** – The AWS resource object upon which actions are performed
- **Principal** – The user, role, federated user, or application that sent the request
- **Environment data** – Information about the IP address, user agent, SSL status, or time of day
- **Resource data** – Data related to the resource that is being requested

Console

CLI

API

s3:GetObject

S3 Bucket

4. Determining whether a request is allowed or denied

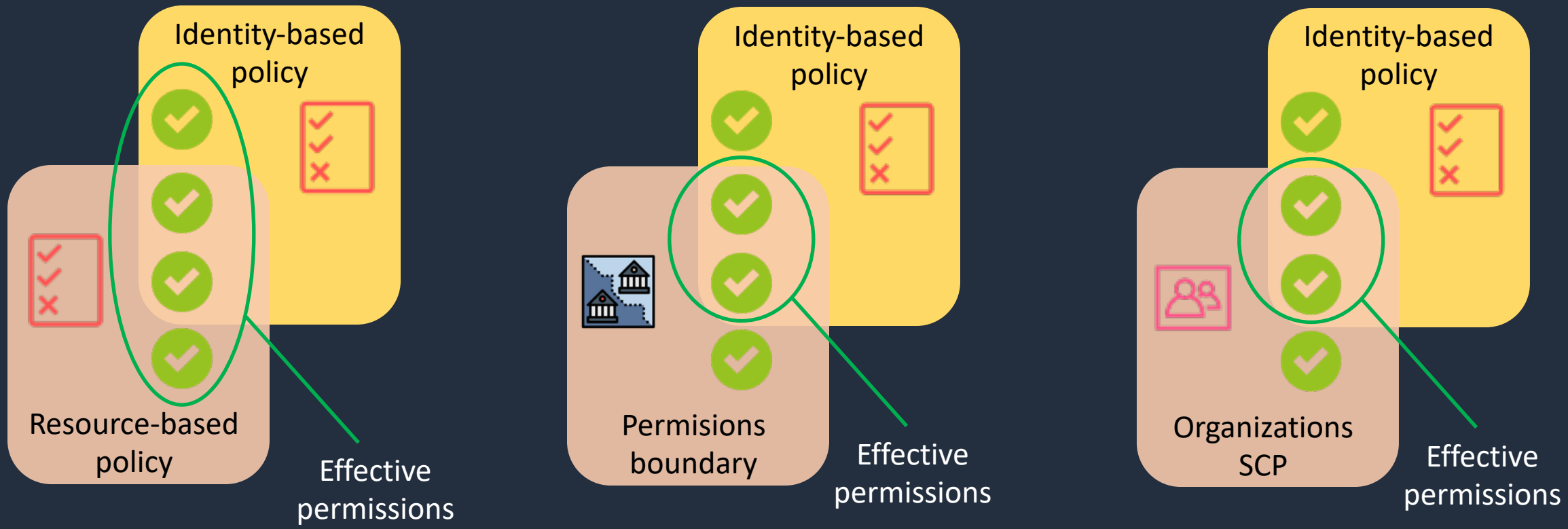
2. Processing the request context

DigitalCloud TRAINING

Types of Policy

- **Identity-based policies** – attached to users, groups, or roles
- **Resource-based policies** – attached to a resource; define permissions for a principal accessing the resource
- **IAM permissions boundaries** – set the maximum permissions an identity-based policy can grant an IAM entity
- **AWS Organizations service control policies (SCP)** – specify the maximum permissions for an organization or OU
- **Session policies** – used with AssumeRole* API actions

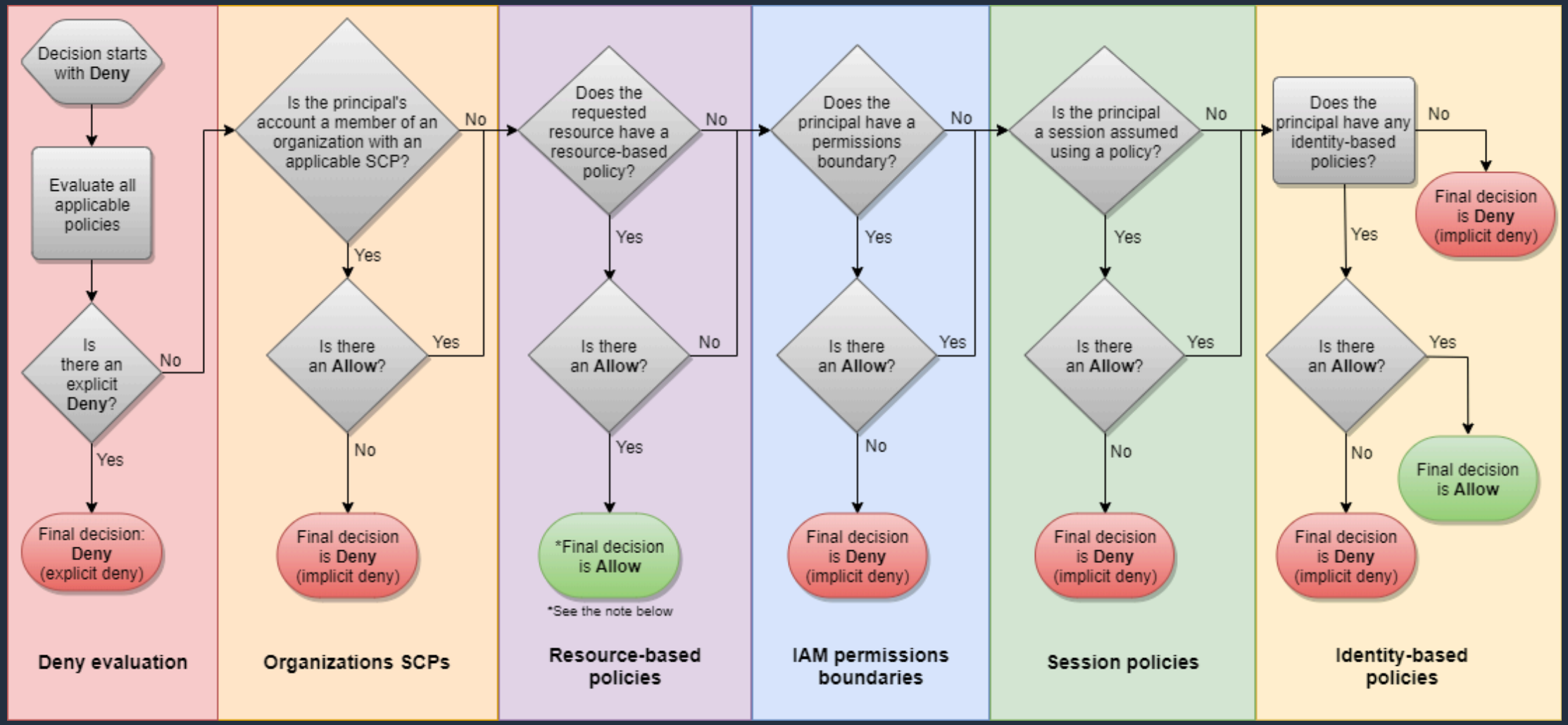
Evaluating Policies within an AWS Account



Determination Rules

1. By default, all requests are implicitly denied (though the root user has full access)
2. An explicit allow in an identity-based or resource-based policy overrides this default
3. If a permissions boundary, Organizations SCP, or session policy is present, it might override the allow with an implicit deny
4. An explicit deny in any policy overrides any allows

Evaluation Logic



IAM Policy Structure

IAM Policy Structure

An IAM policy is a JSON document that consists of one or more statements

```
{
    "Statement":[{
      "Effect":"effect",
      "Action":"action",
      "Resource":"arn",
      "Condition":{
        "condition":{
          "key":"value"
          }
        }
      }
    ]
}
```

The **Effect** element can be Allow or Deny

The **Action** element is the specific API action for which you are granting or denying permission

The **Resource** element specifies the resource that's affected by the action

The **Condition** element is optional and can be used to control when your policy is in effect

IAM Policy Example 1

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

The AdministratorAccess policy uses wildcards (*) to allow all actions on all resources

IAM Policy Example 2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

The specific API action is defined

The effect is to deny the API action if the IP address is not in the specified range

IAM Policy Example 3

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleSatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "elasticfilesystem:ClientRootAccess",
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

You can tell this is a resource-based policy as it has a principal element defined

The policy grants read and write access to an EFS file systems to all IAM principals ("AWS ": "*")

Additionally, the policy condition element requires that SSL/TLS encryption is used

IAM Policy Example 4

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": ["s3:ListBucket"],
        "Effect": "Allow",
        "Resource": ["arn:aws:s3:::mybucket"],
        "Condition": {"StringLike": {"s3:prefix": ["${aws:username}/*"]}}
      },
      {
        "Action": [
          "s3:GetObject",
          "s3:PutObject"
        ],
        "Effect": "Allow",
        "Resource": ["arn:aws:s3:::mybucket/${aws:username}/*"]
      }
    ]
  }
```

A variable is used for the s3:prefix that is replaced with the user's friendly name

The actions are allowed only within the user's folder within the bucket

DigitalCloud TRAINING

Using Role-Based Access Control (RBAC)

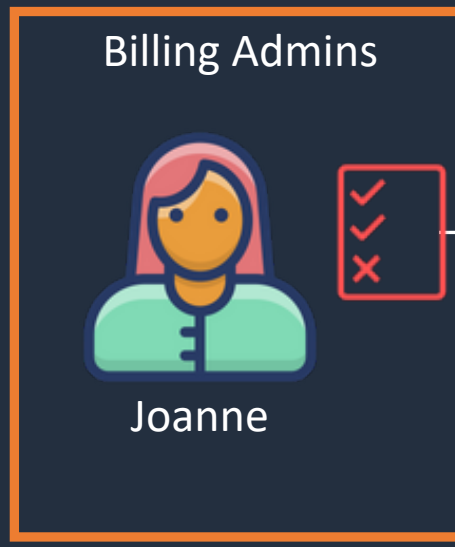
DigitalCloud TRAINING

Role-Based Access Control (RBAC)

Job function policies:

- Administrator
- Billing
- Database administrator
- Data scientist
- Developer power user
- Network administrator
- Security auditor
- Support user
- System administrator
- View-only user

The Billing managed policy is attached to the group

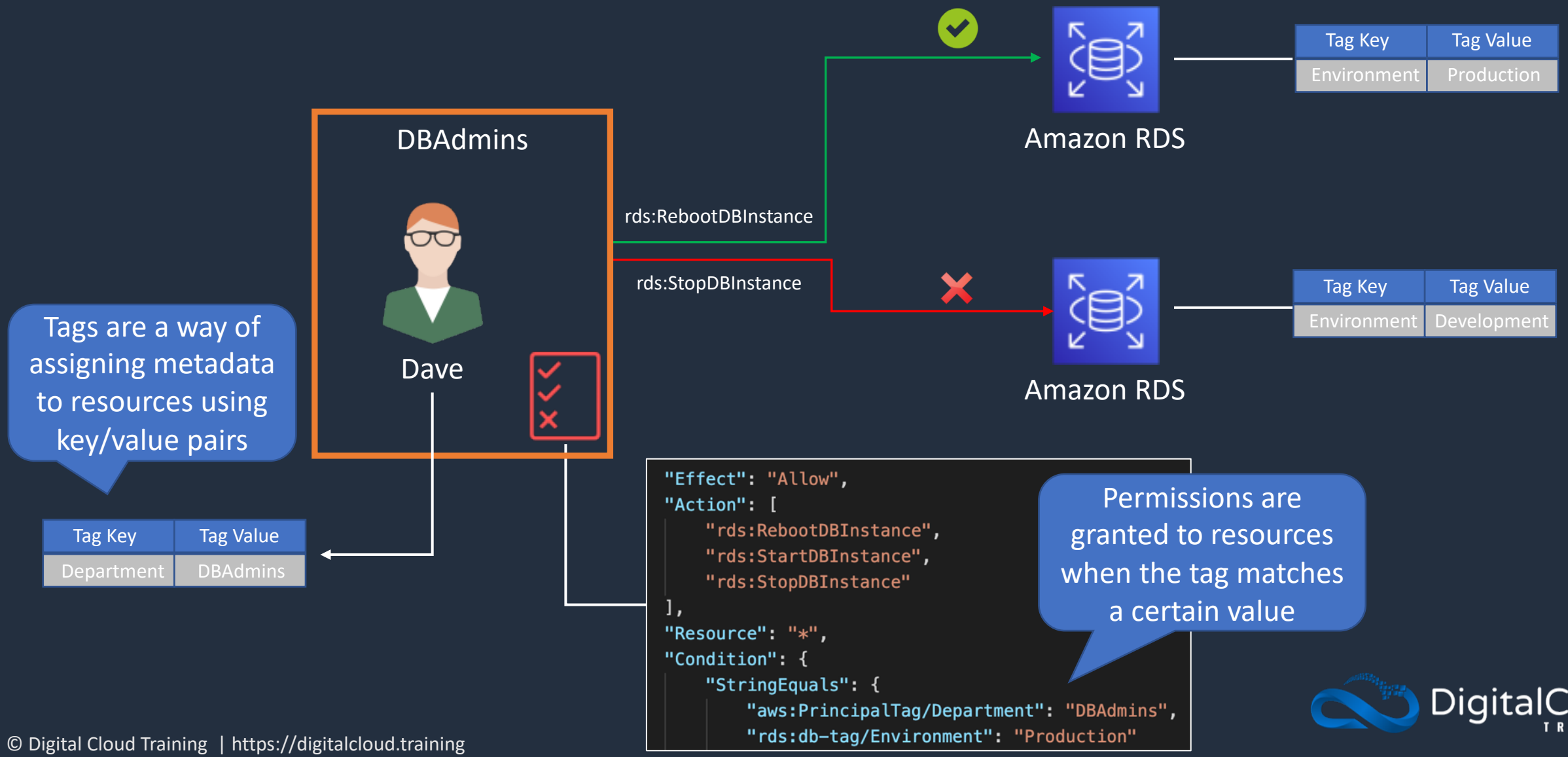



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-portal:*Billing",
                "aws-portal:*Usage",
                "aws-portal:*PaymentMethods",
                "budgets:ViewBudget",
                "budgets:ModifyBudget",
                "ce:UpdatePreferences",
                "ce:CreateReport",
                "ce:UpdateReport",
                "ce:DeleteReport",
                "ce:CreateNotificationSubscription",
                "ce:UpdateNotificationSubscription",
                "ce:DeleteNotificationSubscription",
                "cur:DescribeReportDefinitions",
                "cur:PutReportDefinition",
                "cur:ModifyReportDefinition",
                "cur:DeleteReportDefinition",
                "purchase-orders:*PurchaseOrders"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policies for job functions are designed to closely align to common job functions in the IT industry

Using Attribute-Based Access Control (ABAC)

Attribute-Based Access Control (ABAC)



DigitalCloud TRAINING

Apply Permissions Boundary

Permisions Boundary Hands-On Practice

*** Use the PermissionsBoundary.json file from the course download ***

The policy will enforce the following:

- IAM principals can't alter the permissions boundary to allow their own permissions to access restricted services
- IAM principals must attach the permissions boundary to any IAM principals they create
- IAM admins can't create IAM principals with more privileges than they already have
- The IAM principals created by IAM admins can't create IAM principals with more permissions than IAM admins

DigitalCloud TRAINING

Privilege Escalation

IAMFullAccess

Lindsay is assigned permissions to AWS IAM only and cannot launch AWS resources

Lindsay

iam:CreateUser

IAM

Lindsay applies the AdministratorAccess policy to the X-User account

AdministratorAccess

X-User

Lindsay is now able to login with the X-User account and gain full privileges to the AWS account

AWS Batch

Lindsay mines bitcoins

DigitalCloud TRAINING

AWS Policy Generator

IAM Policy Simulator

IAM Access Analyzer

IAM Access Analyzer

- AWS IAM Access Analyzer helps you identify the resources in your organization and accounts that are **shared** with an **external** entity

Access Analyzer analyzes the following resource types:

- Amazon Simple Storage Service buckets
- AWS Identity and Access Management roles
- AWS Key Management Service keys
- AWS Lambda functions and layers
- Amazon Simple Queue Service queues

SECTION 5

AWS Organizations

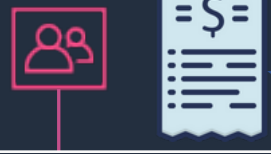
AWS Organizations

AWS Organizations

Enable AWS SSO using on-prem directory

AWS Organization

Enable CloudTrail in management account and apply to members



Receive a consolidated bill

Create accounts programmatically using the Organizations API

Management Account

You can group accounts into Organizational Units (OUs)

Service Control Policies (SCPs) can control tagging and the available API actions

Production

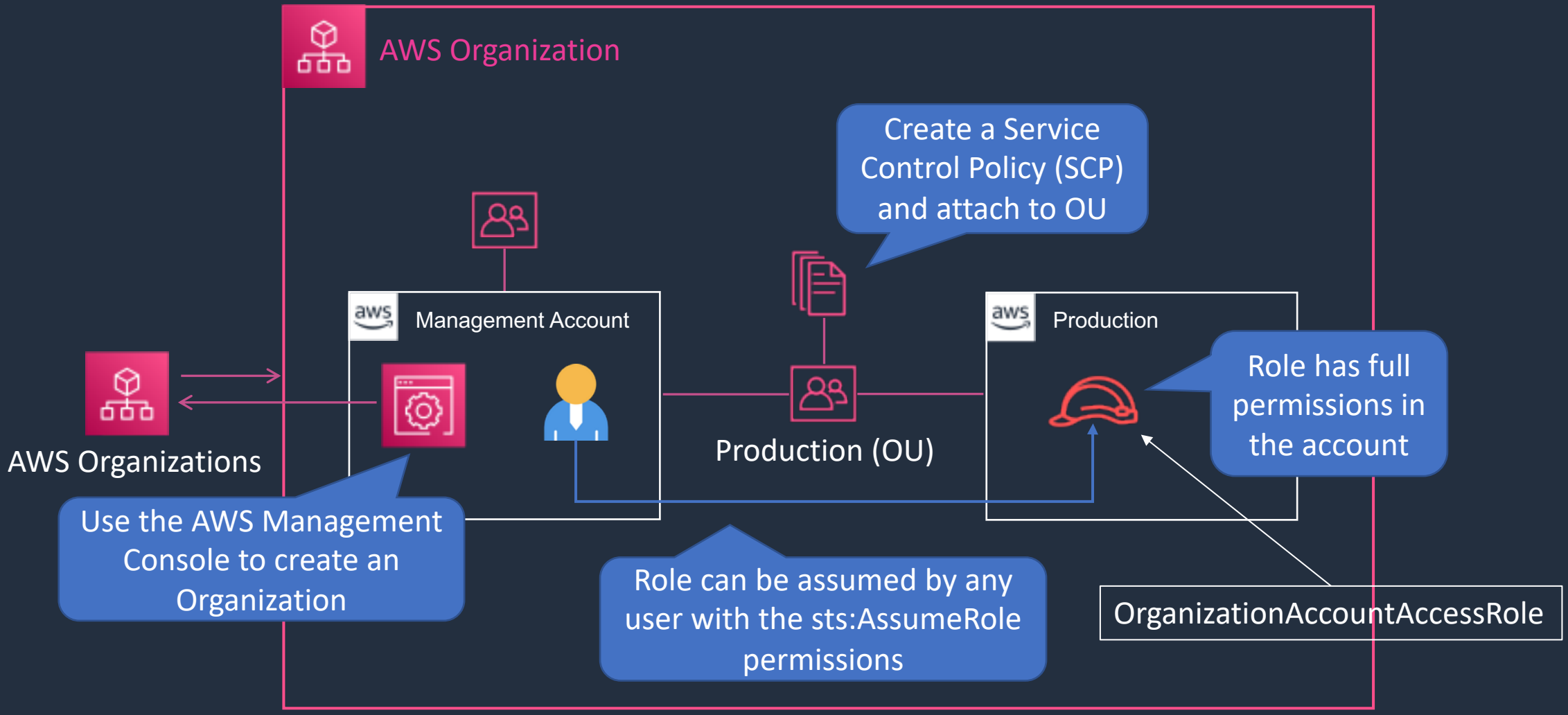
Development

Test

DigitalCloud TRAINING

Account Configuration

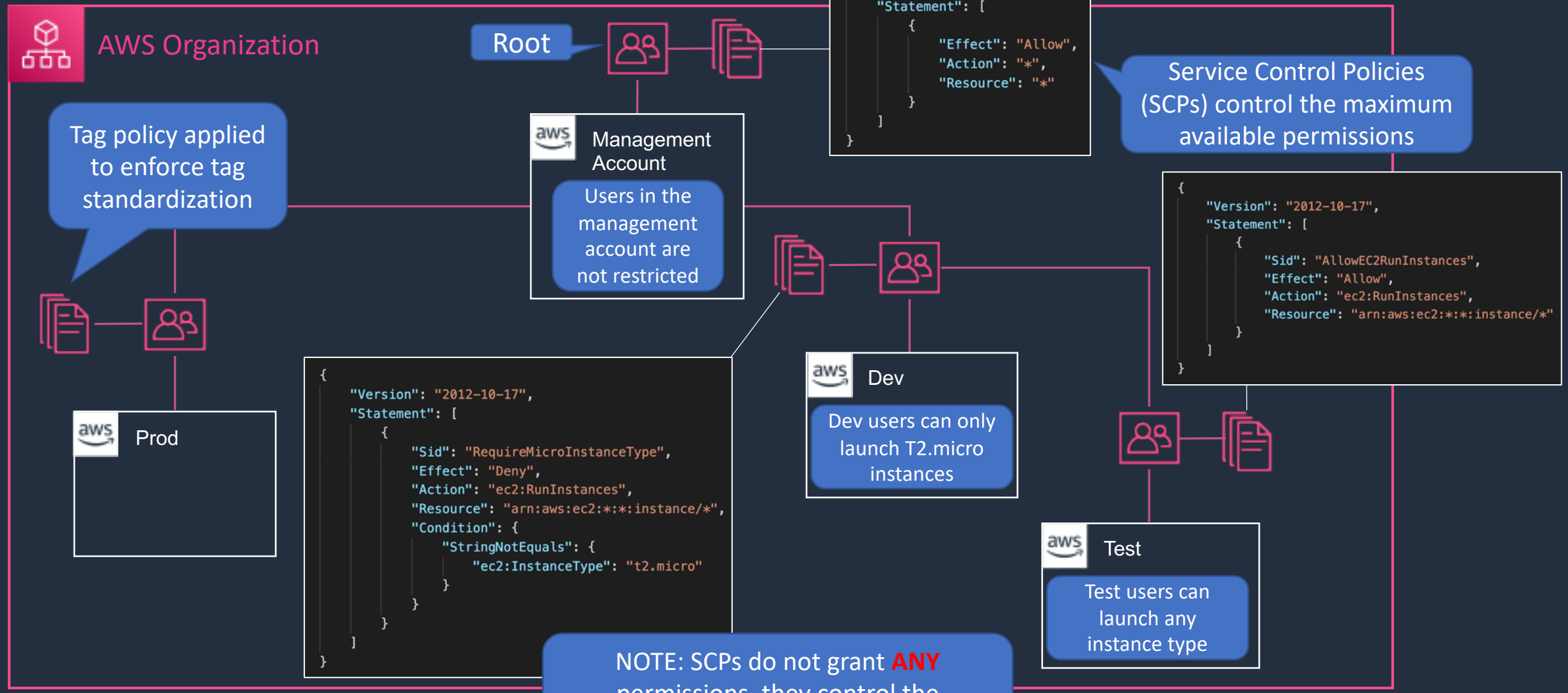
Account Configuration



Create AWS Organization and Add Account

Service Control Policies (SCPs)

Service Control Policies



Apply SCP to Restrict EC2 Instance Types

DigitalCloud TRAINING

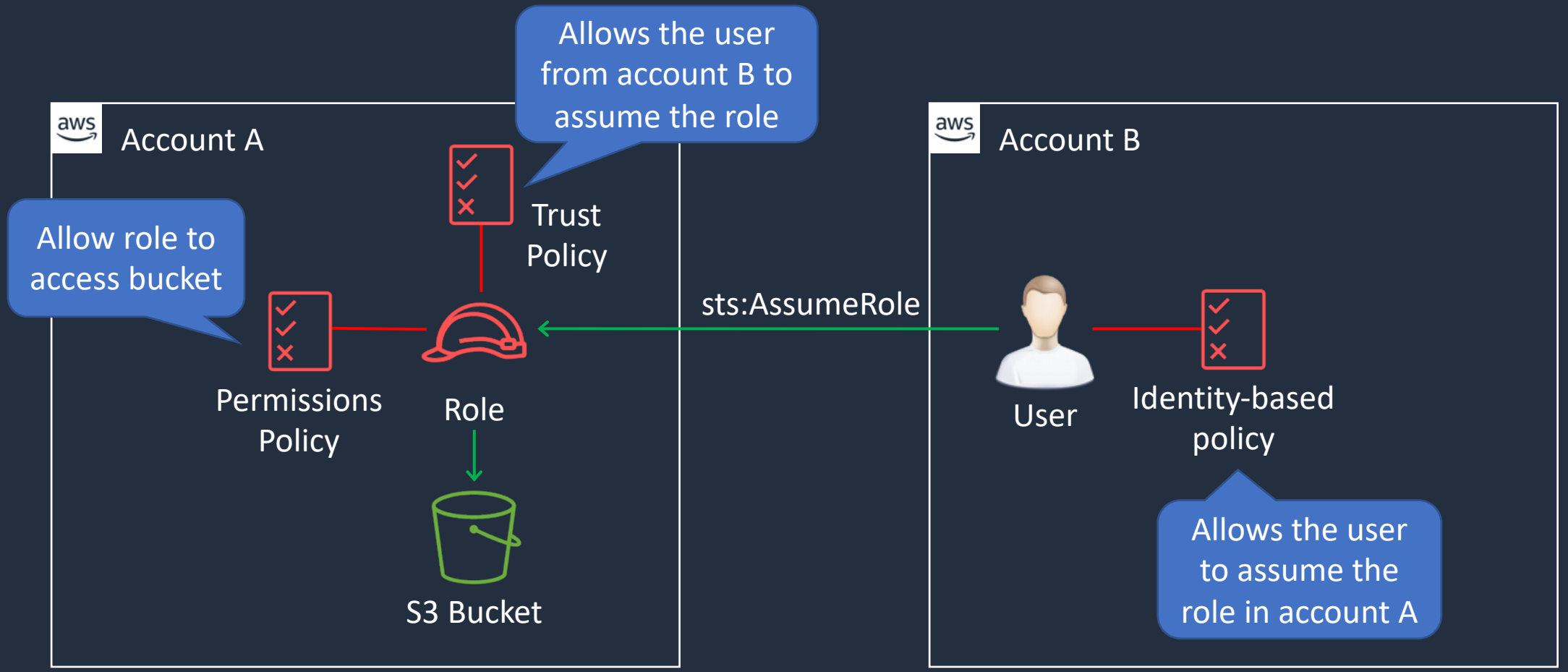
Apply SCP to Prevent S3 Bucket Deletion

SECTION 6

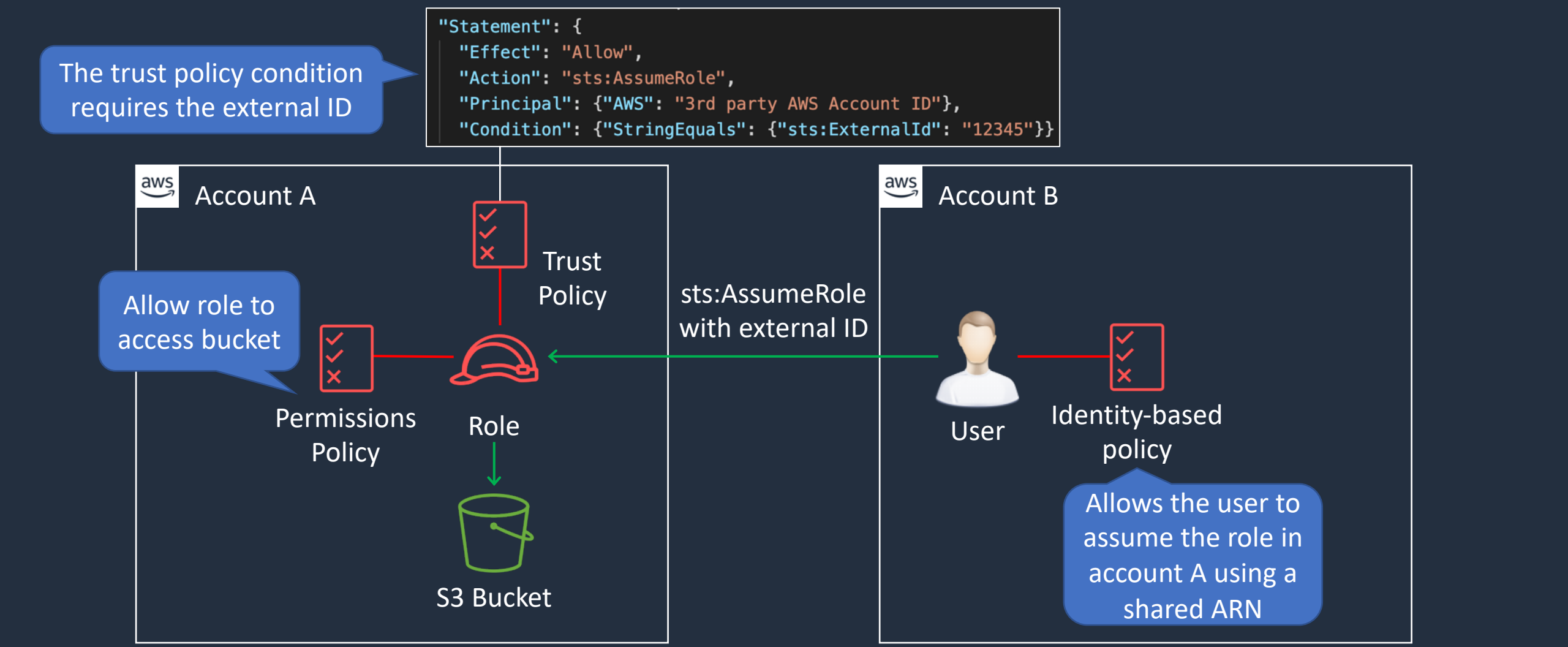
Working with IAM Roles

Use Cases for IAM Roles

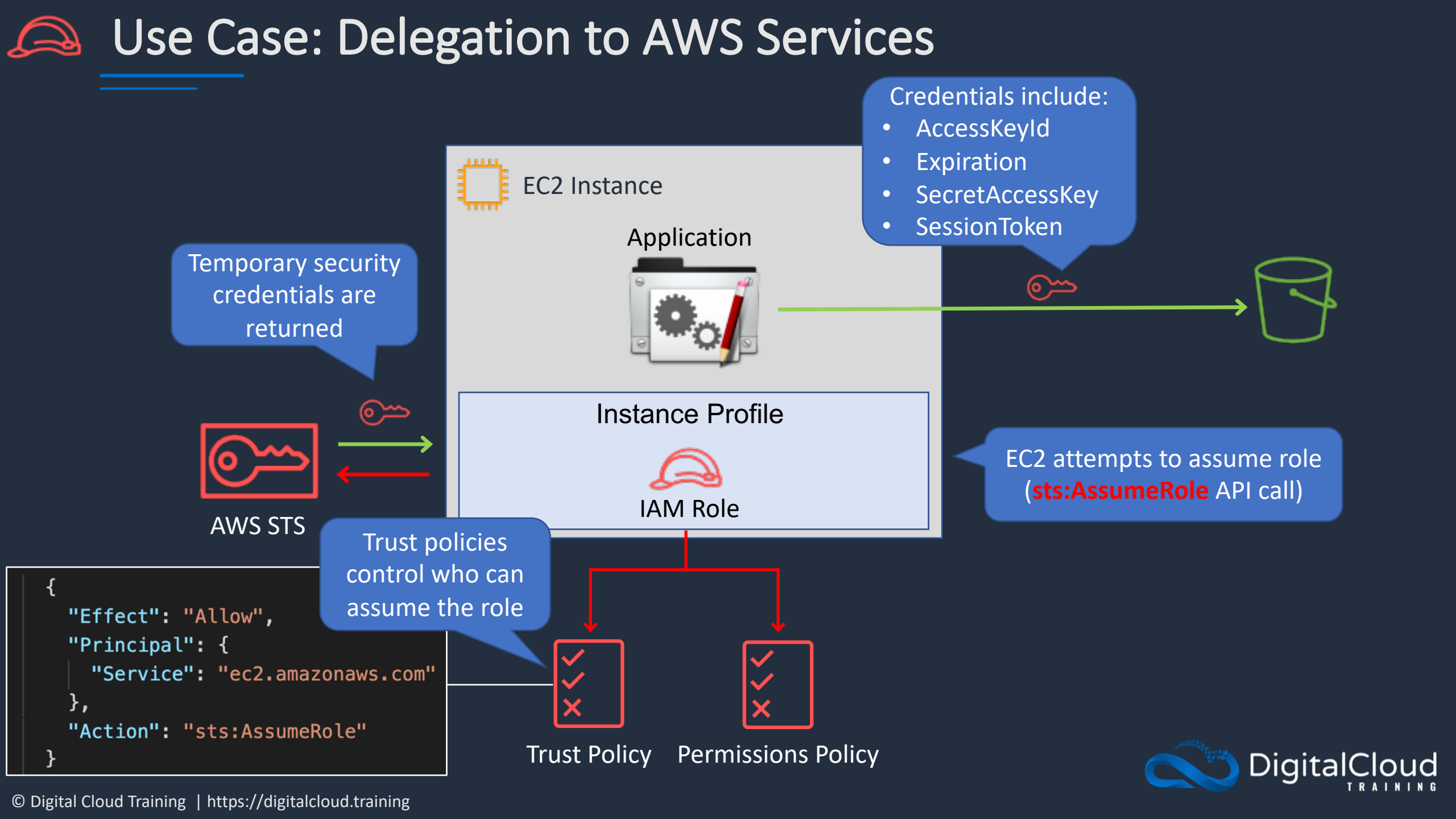
Use Case: Cross Account Access



Use Case: Cross Account Access (3rd Party)



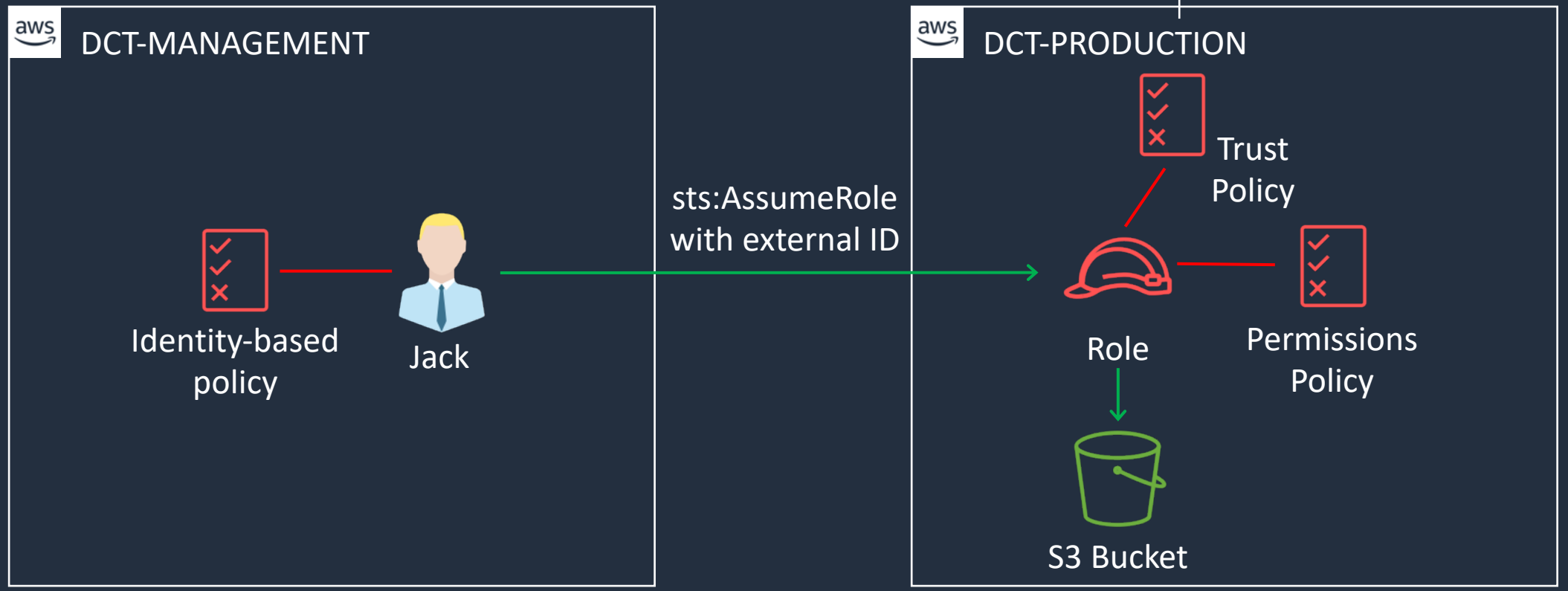
Use Case: Delegation to AWS Services



Cross Account Access to Amazon S3

Cross Account Access (3rd Party) Hands-On

```
"Statement": {
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Principal": {"AWS": "3rd party AWS Account ID"},
  "Condition": {"StringEquals": {"sts:ExternalId": "12345"}}
```



Amazon EC2 Instance Profile

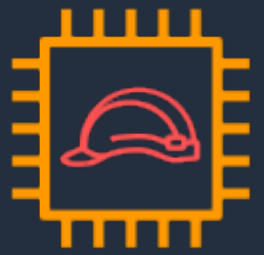
Attach Role to EC2 Instance

Jack

iam:PassRole

S3ReadOnly

The role is attached to the EC2 instance



s3:ListBuckets

Permissions Policy

The user needs permissions to pass the role

```
"Effect": "Allow",
"Action": [
    "iam:GetRole",
    "iam:PassRole"
],
"Resource": "arn:aws:iam::<1132255678:role/s3ReadOnly"
```

Trust Policy

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "ec2.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
```

Permissions Policy

```
"Effect": "Allow",
"Action": [
    "s3:Get*",
    "s3:List*"
],
"Resource": "*"
```

DigitalCloud TRAINING

SECTION 7

Directory Services and Federation

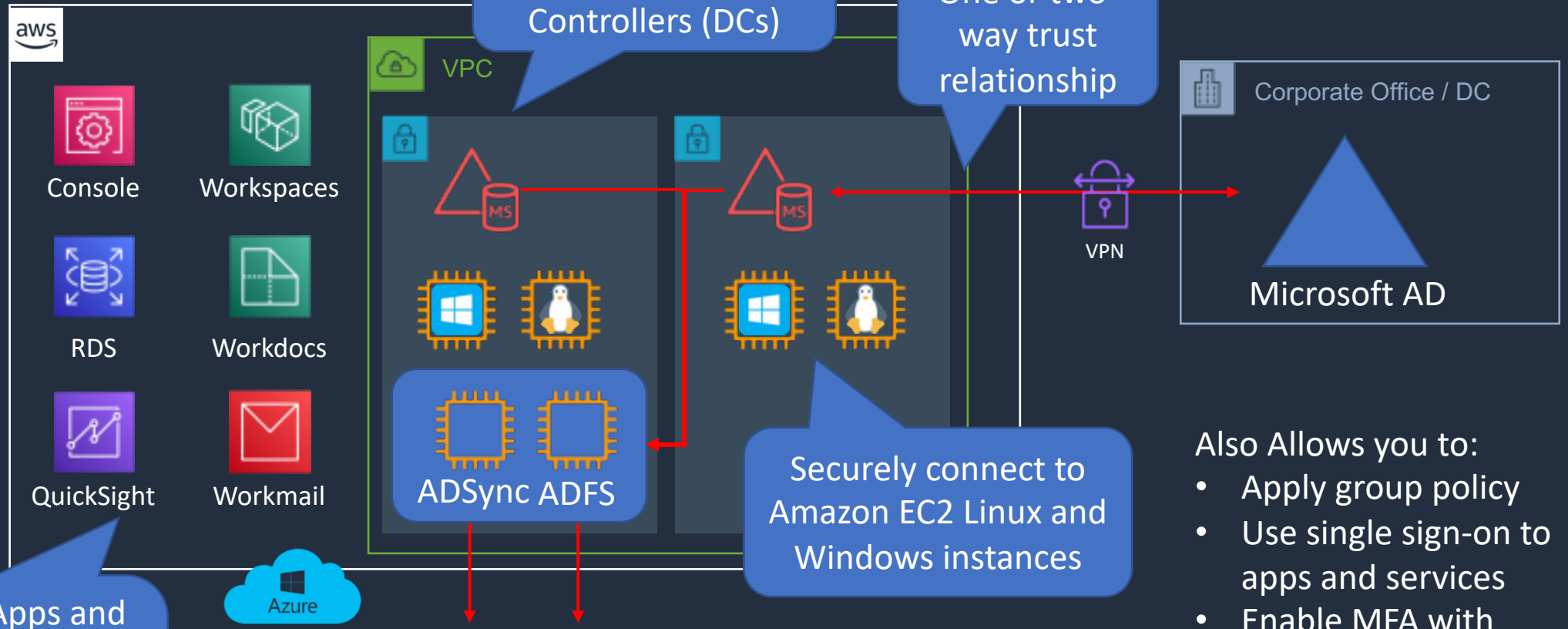
AWS Directory Services

AWS Managed Microsoft AD

Managed implementation of Microsoft Active Directory running on Windows Server 2012 R2

HA pair of Windows Server 2012 Domain Controllers (DCs)

One or two-way trust relationship



Securely connect to Amazon EC2 Linux and Windows instances

Also Allows you to:

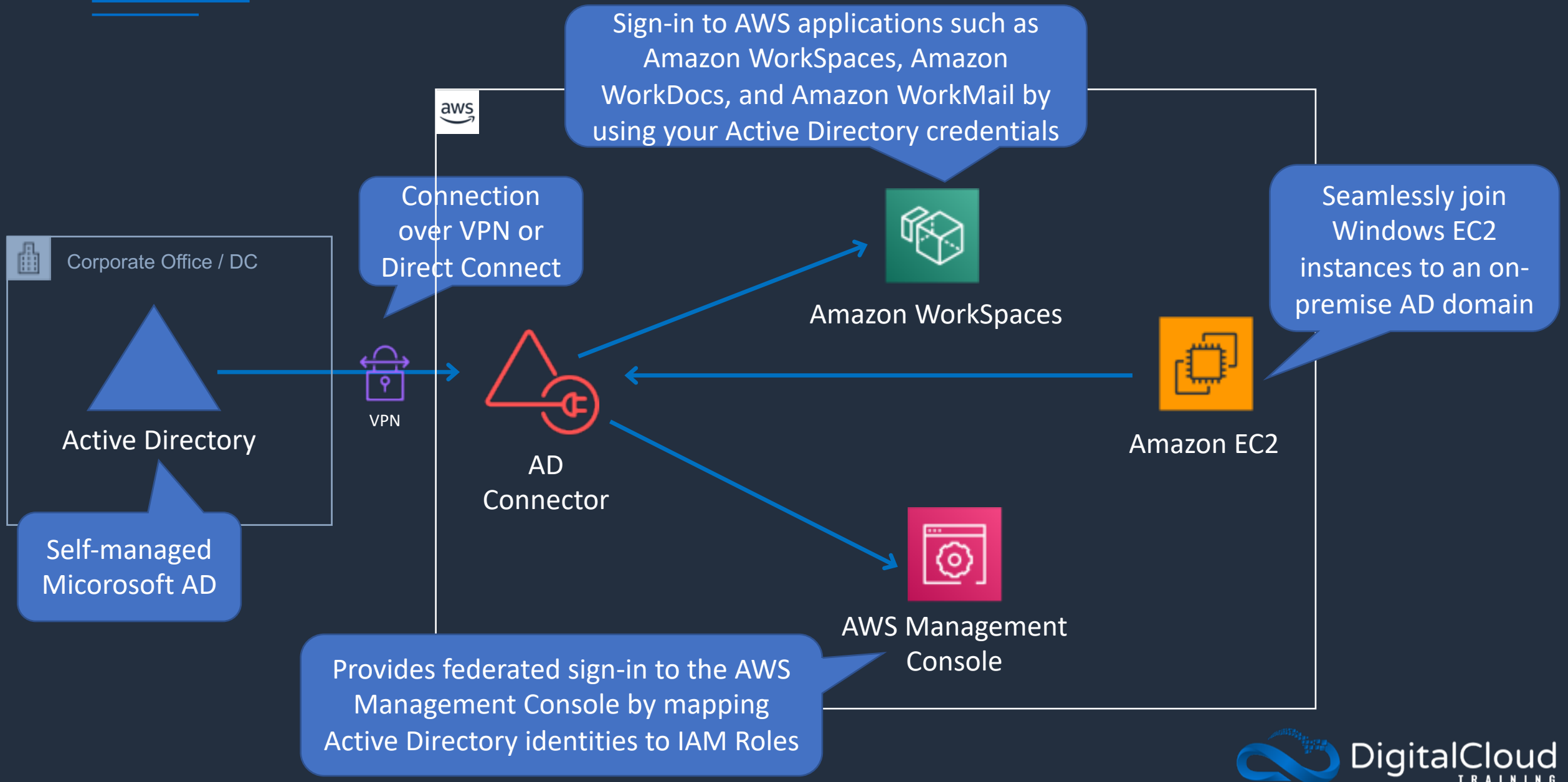
- Apply group policy
- Use single sign-on to apps and services
- Enable MFA with RADIUS

Some of the Apps and Services that support authentication and authorization using AWS Directory Services

Synchronize users and federate identities with Azure/O365

DigitalCloud TRAINING

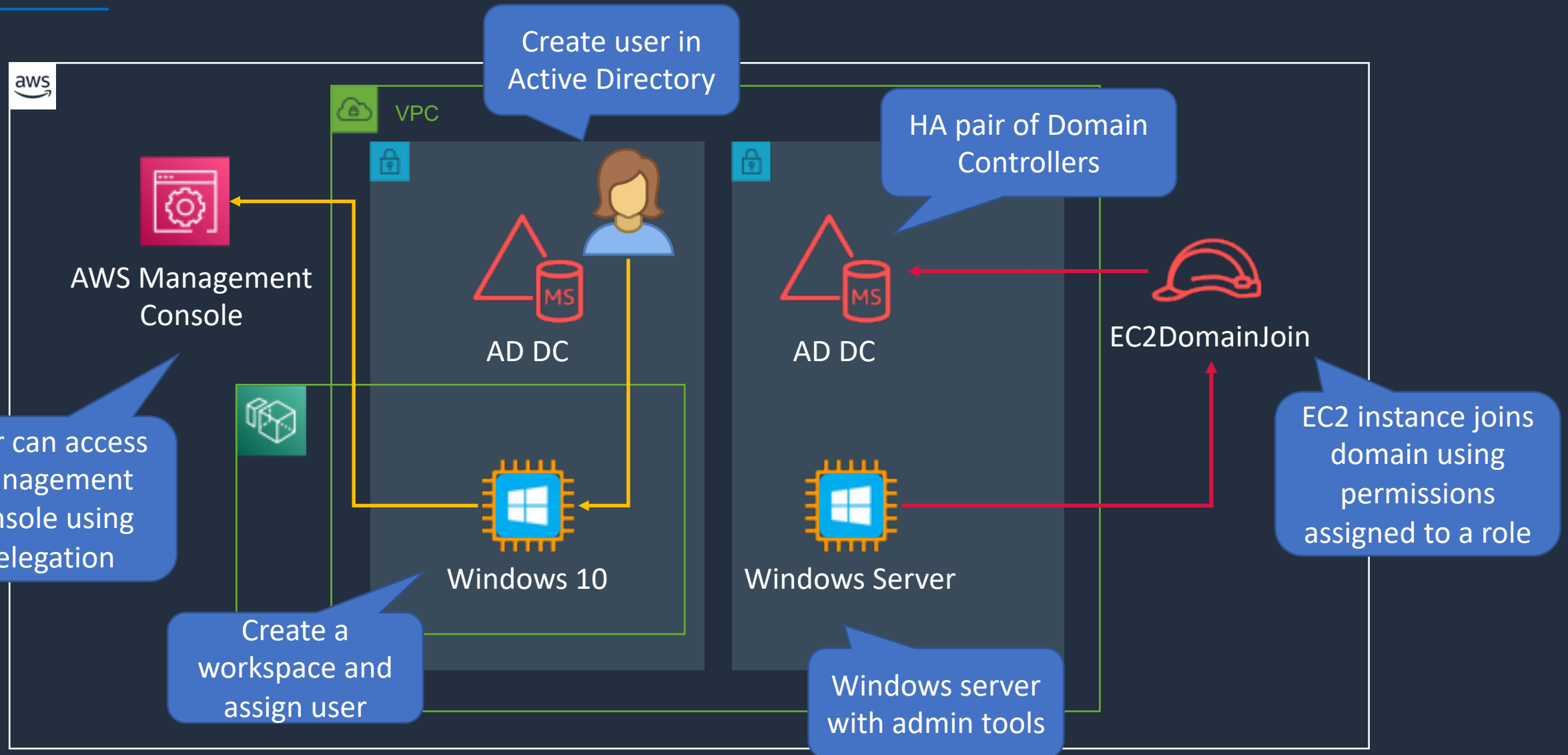
AD Connector



Create AWS Managed Microsoft AD

DigitalCloud TRAINING

AWS Managed Microsoft AD



Delegate Access to Management Console

Identity Federation

Identity Federation Services

AWS Identity & Access Management

- Can use separate SAML 2.0 or OIDC IdPs for each account
- Enables access control using federated user attributes
- User attributes can be cost center, job role etc.

AWS Single Sign-On

- Central management for federated access
- Attach multiple AWS accounts and business applications
- Identities can be in AWS SSO
- Works with many IdPs (e.g. Active Directory)
- Permissions assigned based on group membership in IdP

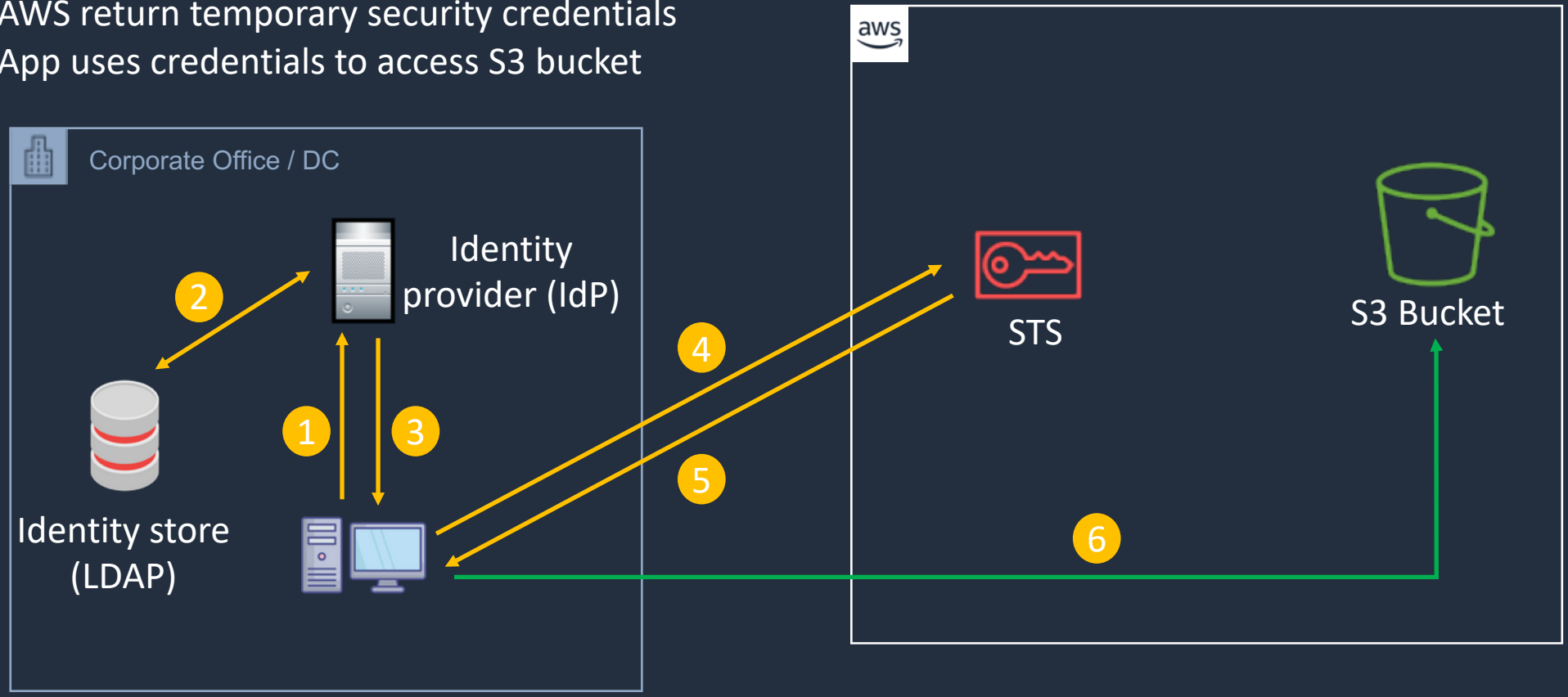
Amazon Cognito

- Federation support for web and mobile applications
- Provides sign-in and sign-up
- Supports sign-in with social IdPs such as Apple, Facebook, Google, and Amazon
- Supports IdPs using SAML 2.0

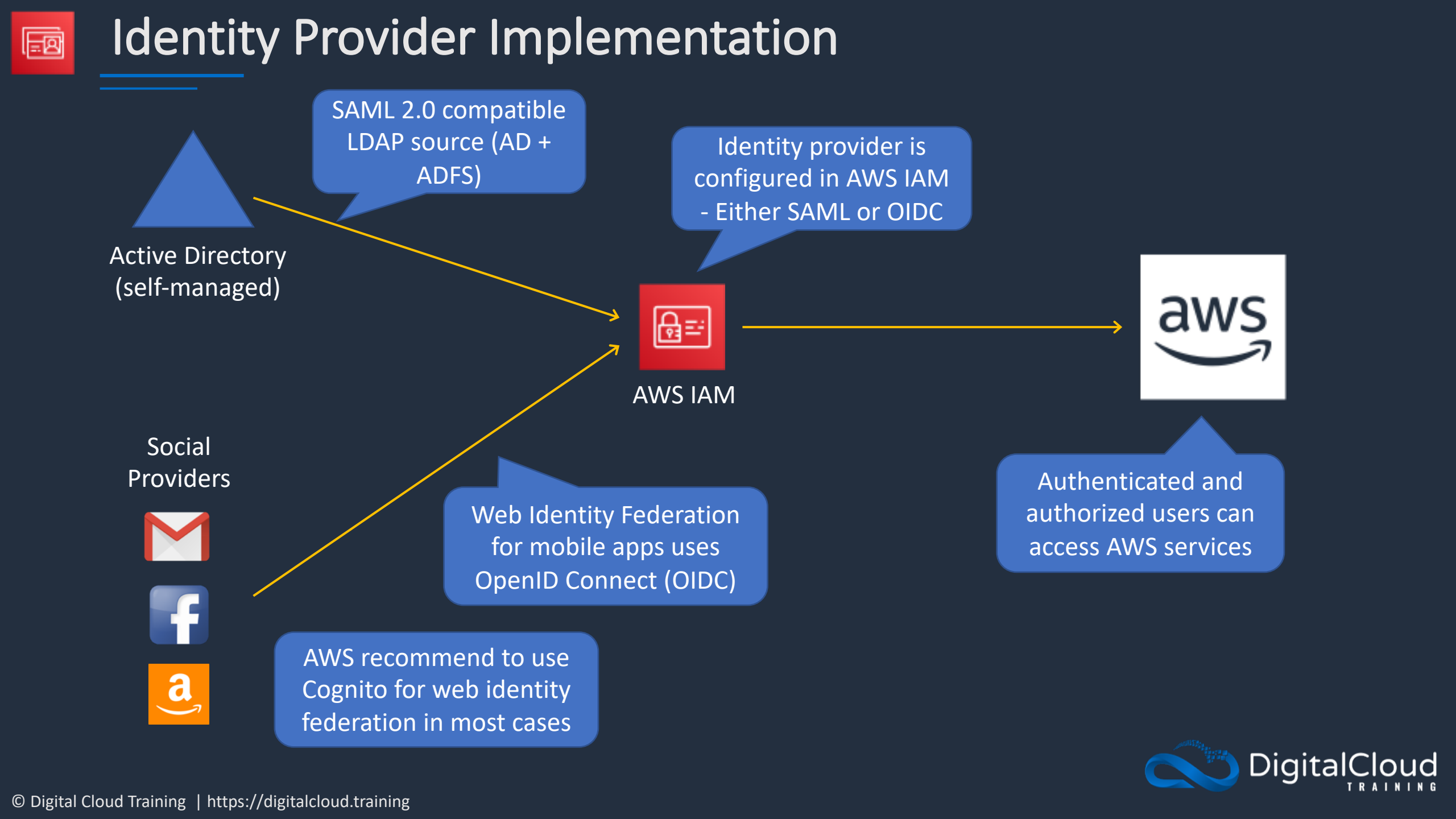
IAM Identity Federation

Identity Federation

1. Client application attempts to authenticate using IdP
2. IdP authenticates the user
3. IdP sends client SAML assertion
4. App calls sts:AssumeRoleWithSAML
5. AWS return temporary security credentials
6. App uses credentials to access S3 bucket



Identity Provider Implementation



AWS Single Sign-on (SSO)

AWS Single Sign-on (SSO)

Identity sources can be AWS SSO, Active Directory and standard providers using SAML 2.0

Enables centralized permissions management

Corporate Office / DC

Active Directory (self-managed)

AWS Directory Service

AWS Single Sign-On

AWS Organizations

AWS Account A

AWS Account B

Azure

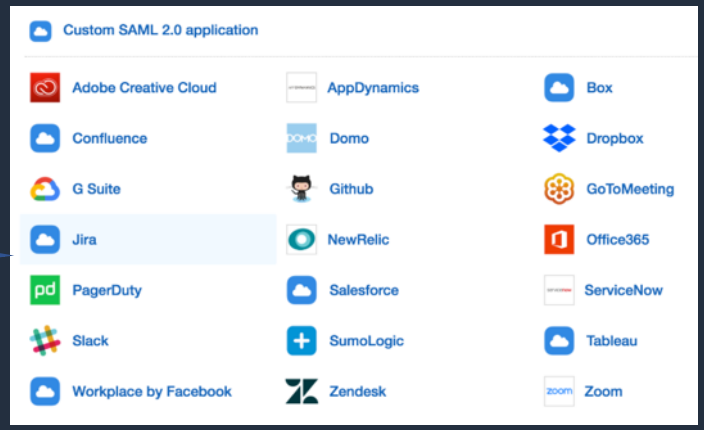
Azure AD

AWS Account C

AWS Account D

Connect AWS accounts and Organizations

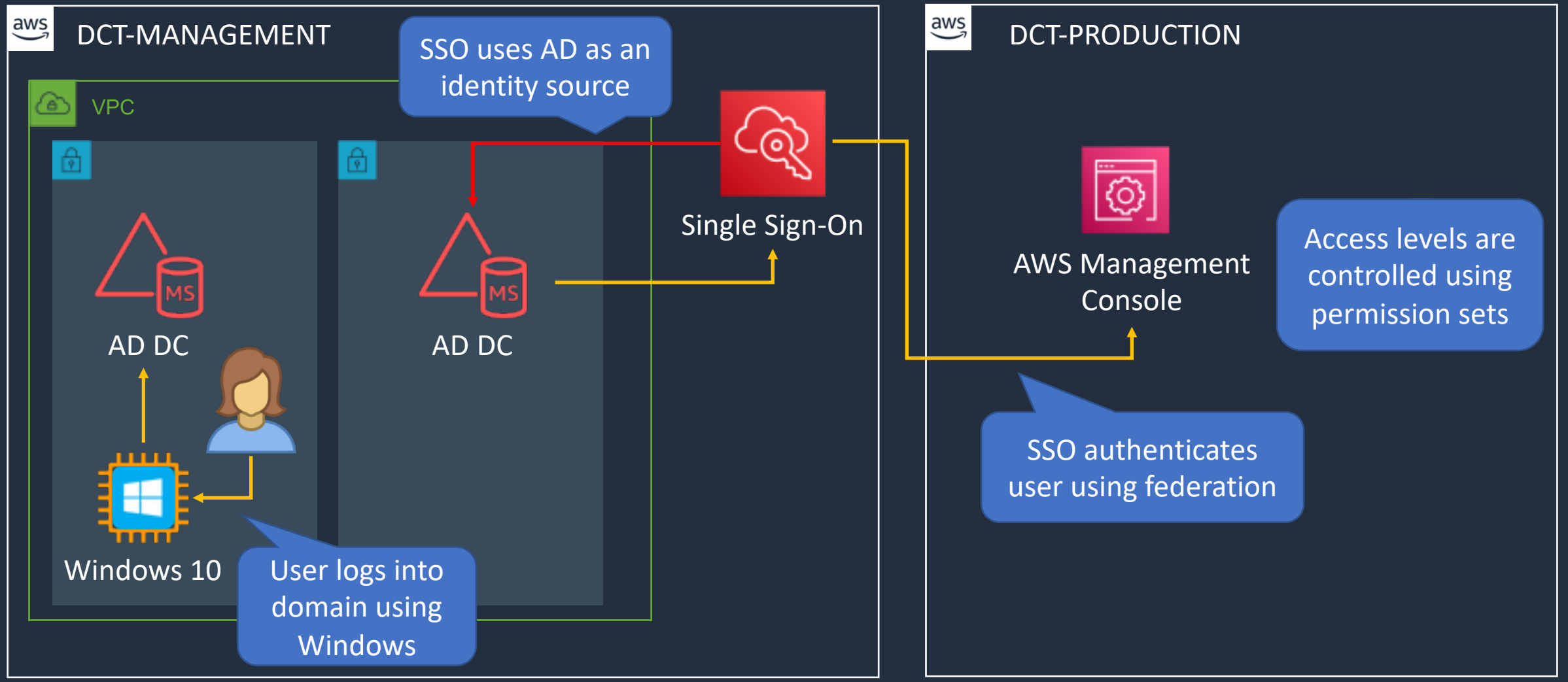
Built-in SSO integrations to business applications



Configure AWS SSO with AWS Managed AD

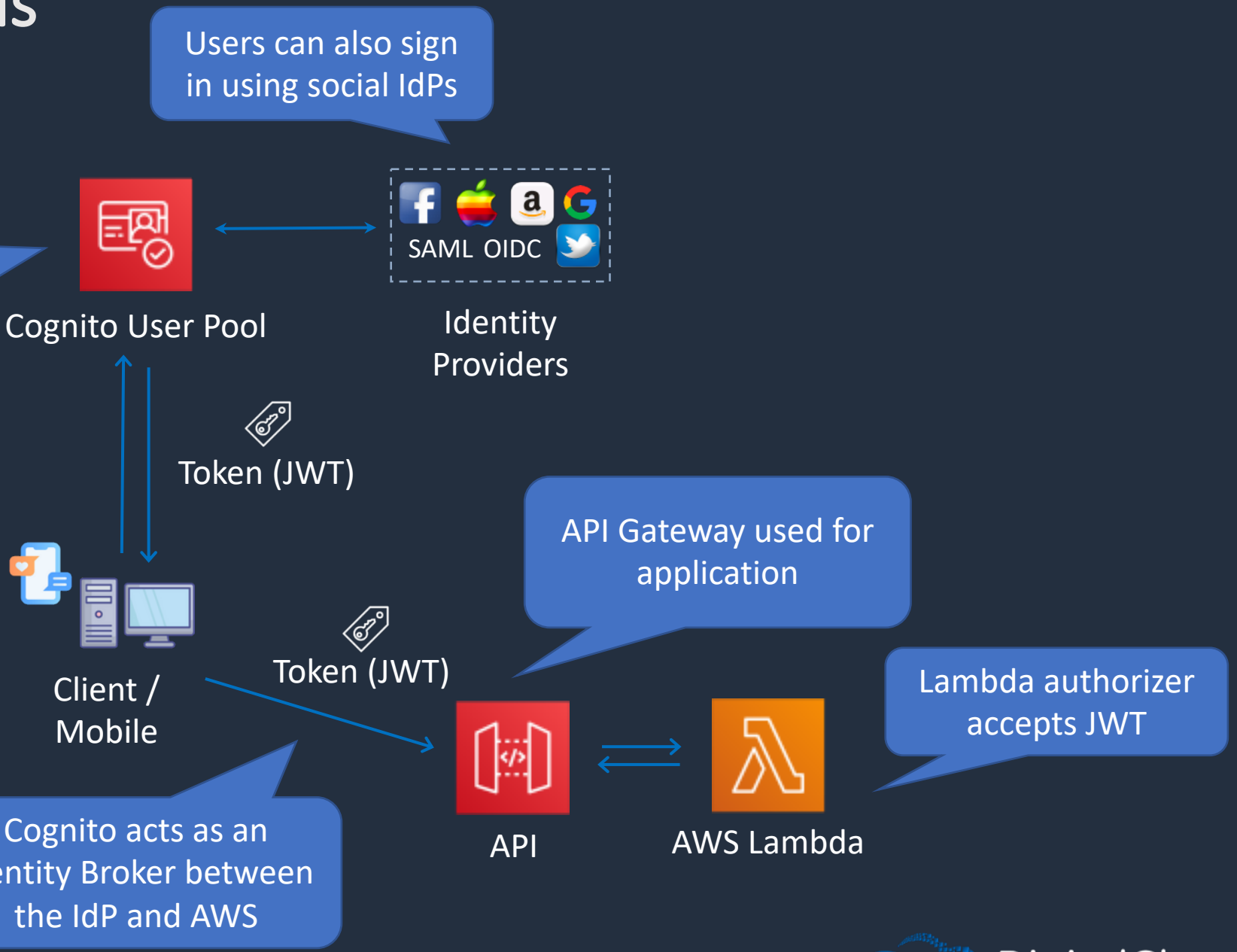
DigitalCloud TRAINING

AWS Managed Microsoft AD

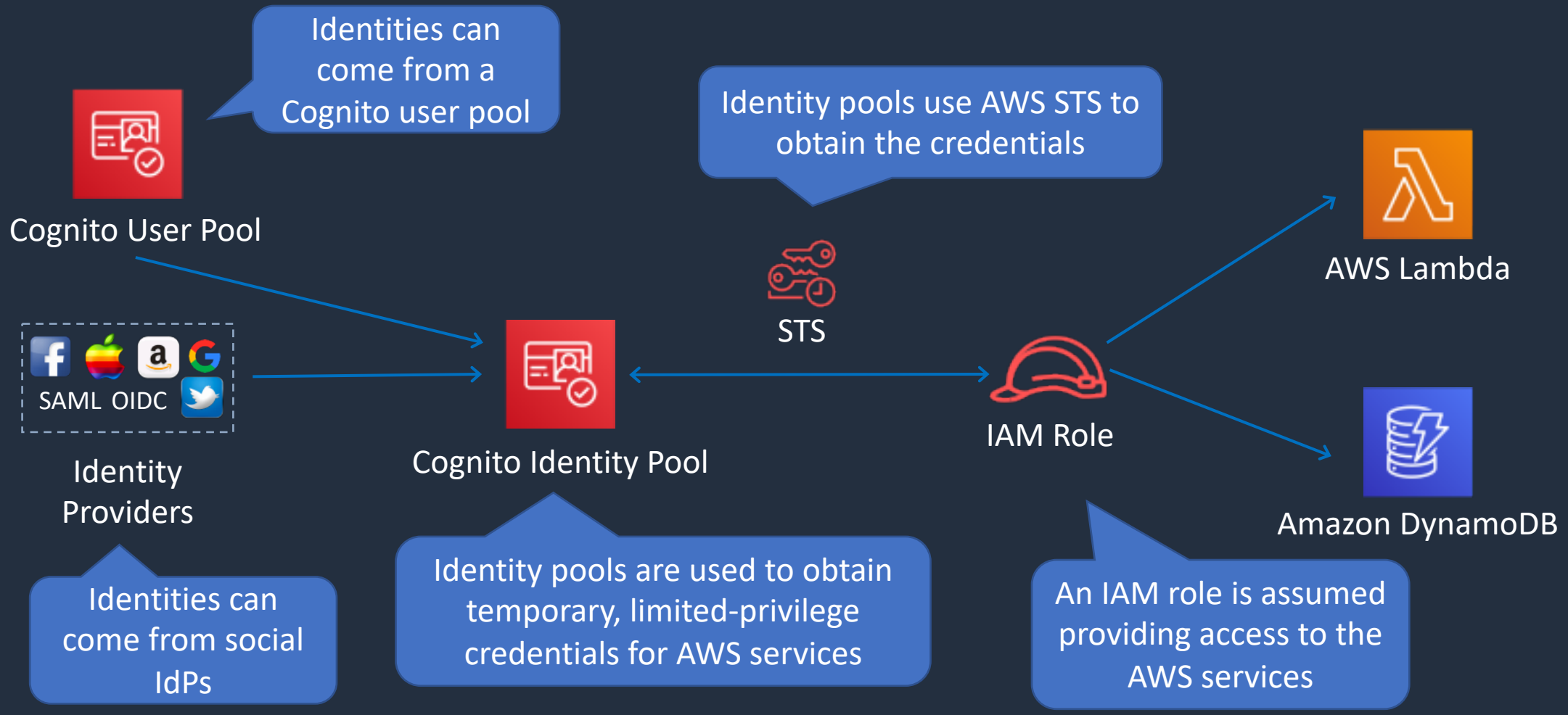


Amazon Cognito

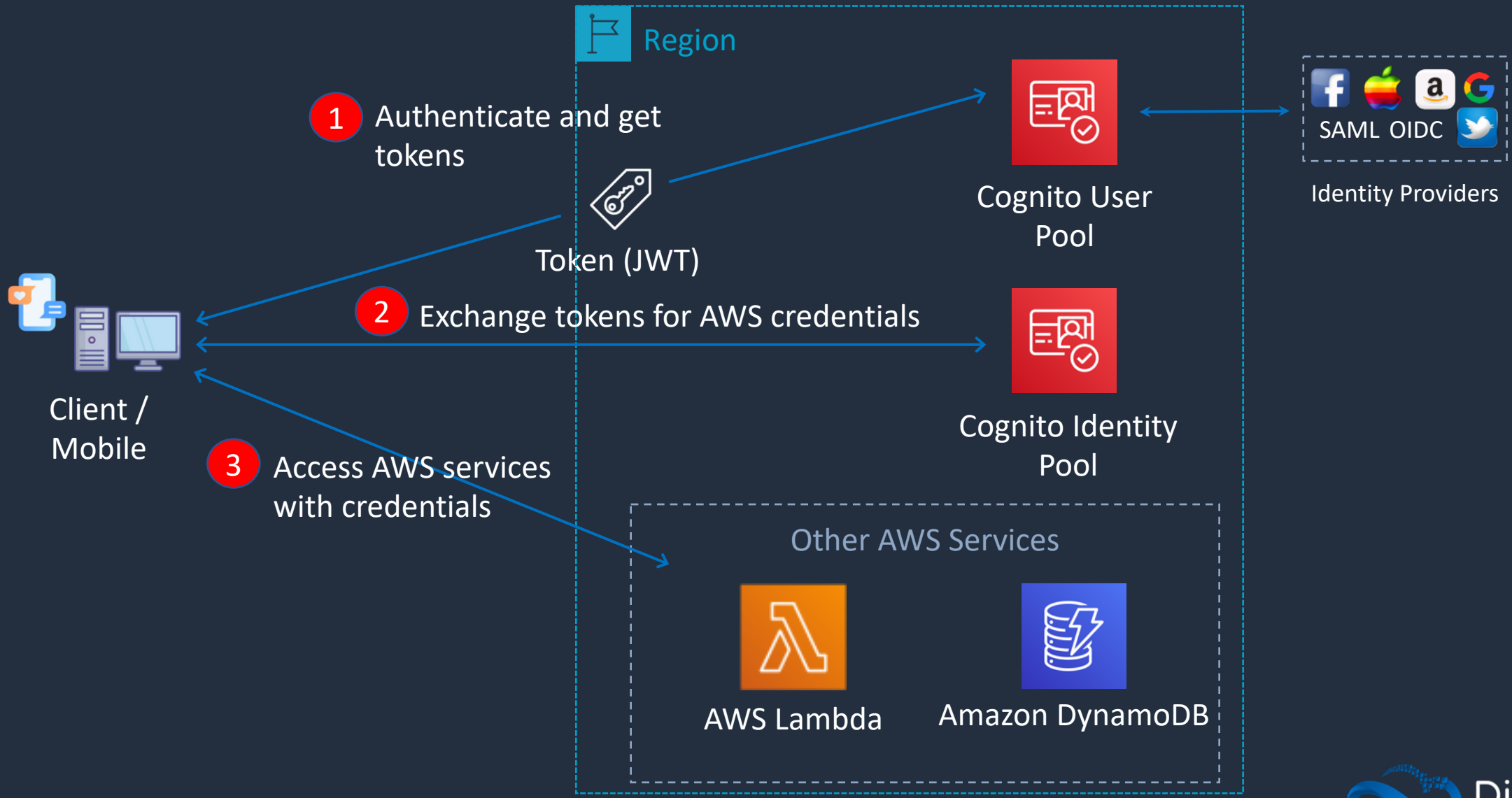
Cognito User Pools



Cognito Identity Pool



User Pools + Identity Pools



DigitalCloud TRAINING