

# SECTION 1

Let's Get Started!

# The SCS-C02 Exam





# The SCS-C02 Exam

---

**Level:** Specialty

**Length:** 170 minutes

**Format:** 65 questions

**Cost:** \$300 USD

**Delivery Method:** Testing center or online

**Scoring:**

- Scaled score between 100 – 1000
- Minimum passing score of 750



# The SCS-C02 Exam

---

---

## Recommended knowledge:

- Five years of IT security experience in designing and implementing security solutions and at least two years of hands-on experience in securing AWS workloads
- Working knowledge of AWS security services and features of services to provide a secure production environment and an understanding of security operations and risks
- Knowledge of the AWS shared responsibility model and its application; security controls for workloads on AWS; logging and monitoring strategies; cloud security threat models; patch management and security automation; ways to enhance AWS security services with third-party tools and services; and disaster recovery controls, including BCP and backups, encryption, access control, and data retention
- Understanding of specialized data classifications and AWS data protection mechanisms, data-encryption methods and AWS mechanisms to implement them, and secure internet protocols and AWS mechanisms to implement them
- Ability to make tradeoff decisions with regard to cost, security, and deployment complexity to meet a set of application requirements



## Question format:

- **Multiple-choice:** Has one correct response and three incorrect responses
- **Multiple-response:** Has two or more correct responses out of five or more options



# The SCS-C02 Exam

---

---

## Domain 1: Threat Detection and Incident Response

- Task Statement 1: Design and implement an incident response plan
- Task Statement 2: Detect security threats and anomalies by using AWS services
- Task Statement 3: Respond to compromised resources and workloads

## Domain 2: Security Logging and Monitoring

- Task Statement 1: Design and implement monitoring and alerting to address security events
- Task Statement 2: Troubleshoot security monitoring and alerting
- Task Statement 3: Design and implement a logging solution
- Task Statement 4: Troubleshoot logging solutions
- Task Statement 5: Design a log analysis solution



# The SCS-C02 Exam

---

---

## Domain 3: Infrastructure Security

- Task Statement 1: Design and implement security controls for edge services
- Task Statement 2: Design and implement network security controls
- Task Statement 3: Design and implement security controls for compute workloads
- Task Statement 4: Troubleshoot network security

## Domain 4: Identity and Access Management

- Task Statement 1: Design, implement, and troubleshoot authentication for AWS resources
- Task Statement 2: Design, implement, and troubleshoot authorization for AWS resources



# The SCS-C02 Exam

---

---

## Domain 5: Data Protection

- Task Statement 1: Design and implement controls that provide confidentiality and integrity for data in transit
- Task Statement 2: Design and implement controls that provide confidentiality and integrity for data at rest
- Task Statement 3: Design and implement controls to manage the lifecycle of data at rest
- Task Statement 4: Design and implement controls to protect credentials, secrets, and cryptographic key materials



# The SCS-C02 Exam

---

---

## Domain 6: Management and Security Governance

- Task Statement 1: Develop a strategy to centrally deploy and manage AWS accounts
- Task Statement 2: Implement a secure and consistent deployment strategy for cloud resources
- Task Statement 3: Evaluate the compliance of AWS resources
- Task Statement 4: Identify security gaps through architectural reviews and cost analysis



# The SCS-C02 Exam

---

---

Domain	% of Exam
Domain 1: Threat Detection and Incident Response	14%
Domain 2: Security Logging and Monitoring	18%
Domain 3: Infrastructure Security	20%
Domain 4: Identity and Access Management	16%
Domain 5: Data Protection	18%
Domain 6: Management and Security Governance	14%

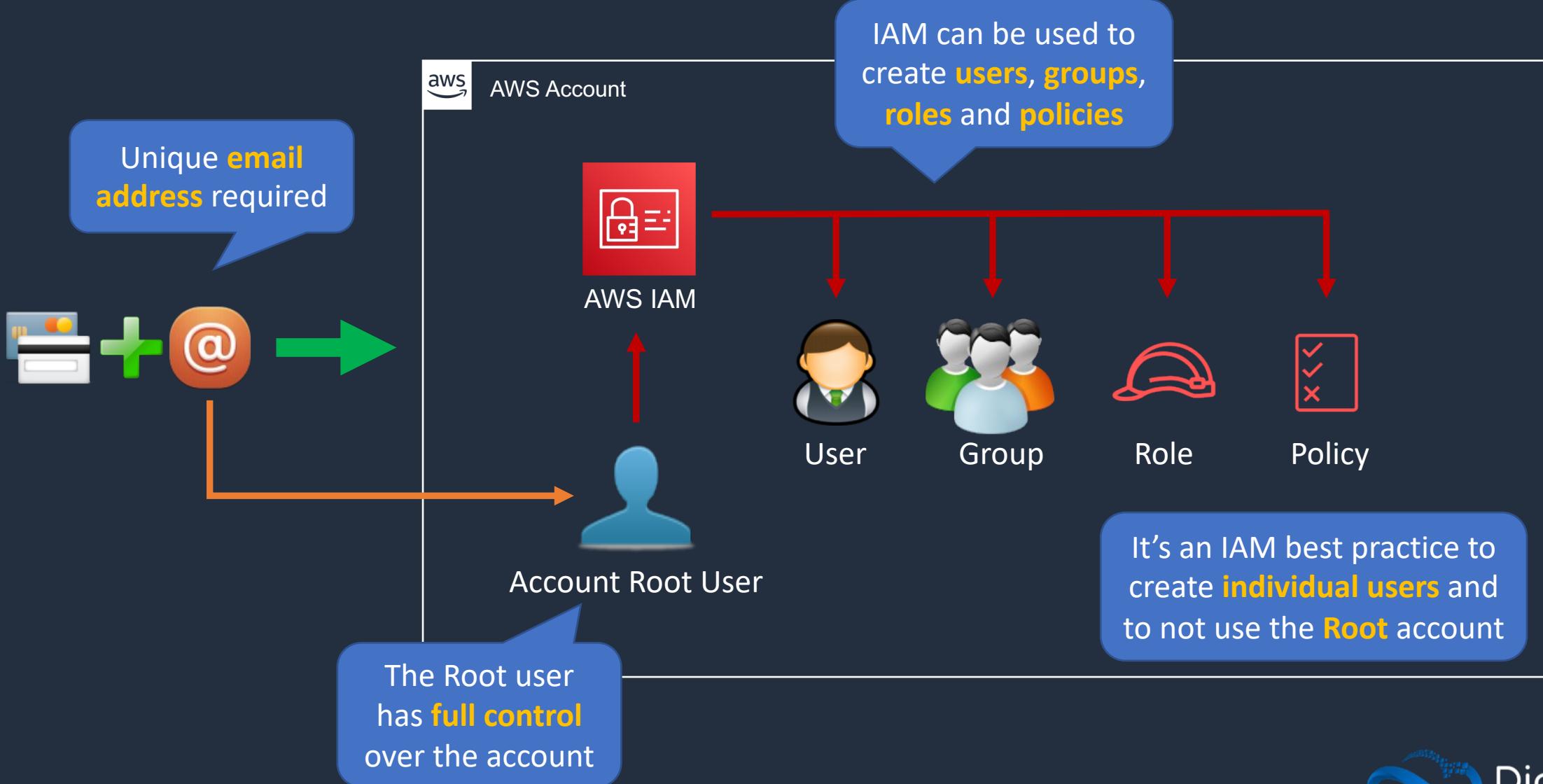
# SECTION 2

## Getting Started - AWS Accounts

# AWS Account Overview



# AWS Account Overview



# AWS Account Overview



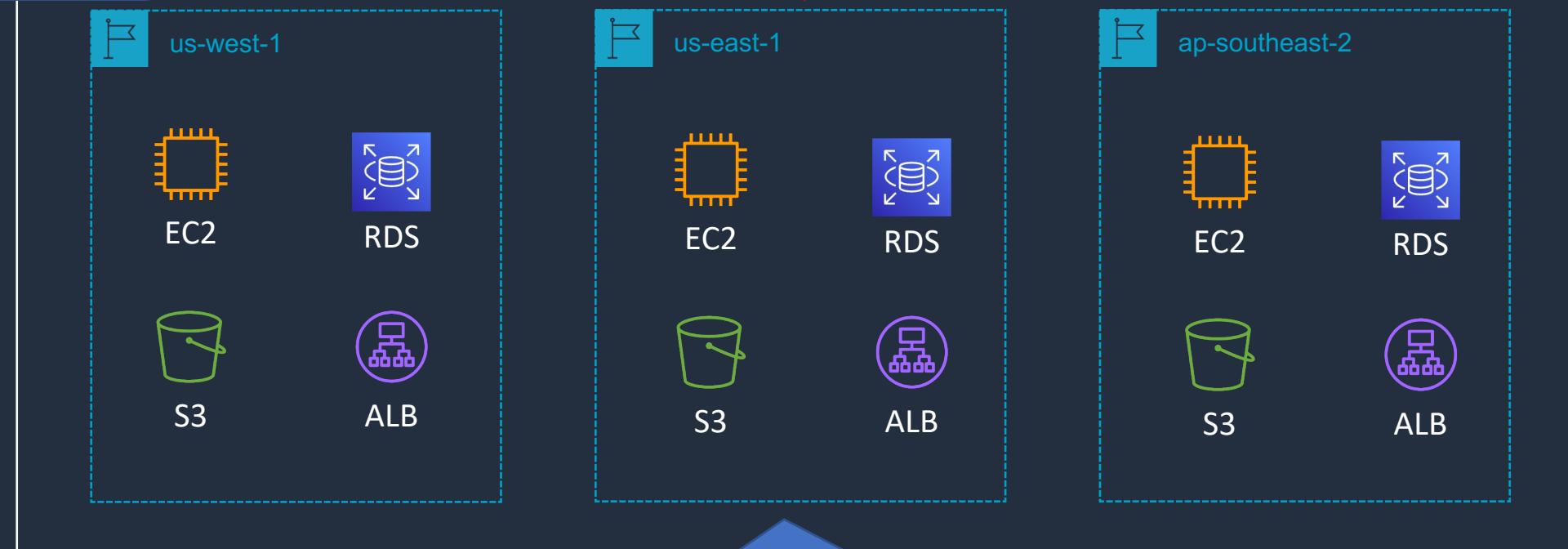
**Authentication:** IAM principals authenticate to IAM using the console, API, or CLI

**Authorization:** IAM principals can then create resources across AWS Regions

AWS Management  
Console



AWS IAM



All AWS **identities** and **resources** are created within the AWS account  
lucabigoni@gmail.com

# Create Your AWS Free Tier Account



# What you need...



Credit card for setting up the account and paying any bills



Unique email address for this account

john@gmail.com



Check if you can use a **dynamic alias** with an existing email address

john+ACCOUNT-ALIAS-1@gmail.com

john+ACCOUNT-ALIAS-2@gmail.com



AWS account name / alias



Phone to receive an **SMS** verification code

# Configure Account and Create a Billing Alarm



# Account Configuration

- Configure **Account Alias**
- Enable access to billing for **IAM users**
- Update **billing preferences**
- Create a **billing alarm**
- Confirm **SNS subscription**

# Install Tools (AWS CLI, VS Code, CloudShell)



# Install Tools

---

- Install the **AWS Command Line Interface (CLI)**
- Install **Visual Studio Code**
- Launch **AWS CloudShell**

# SECTION 3

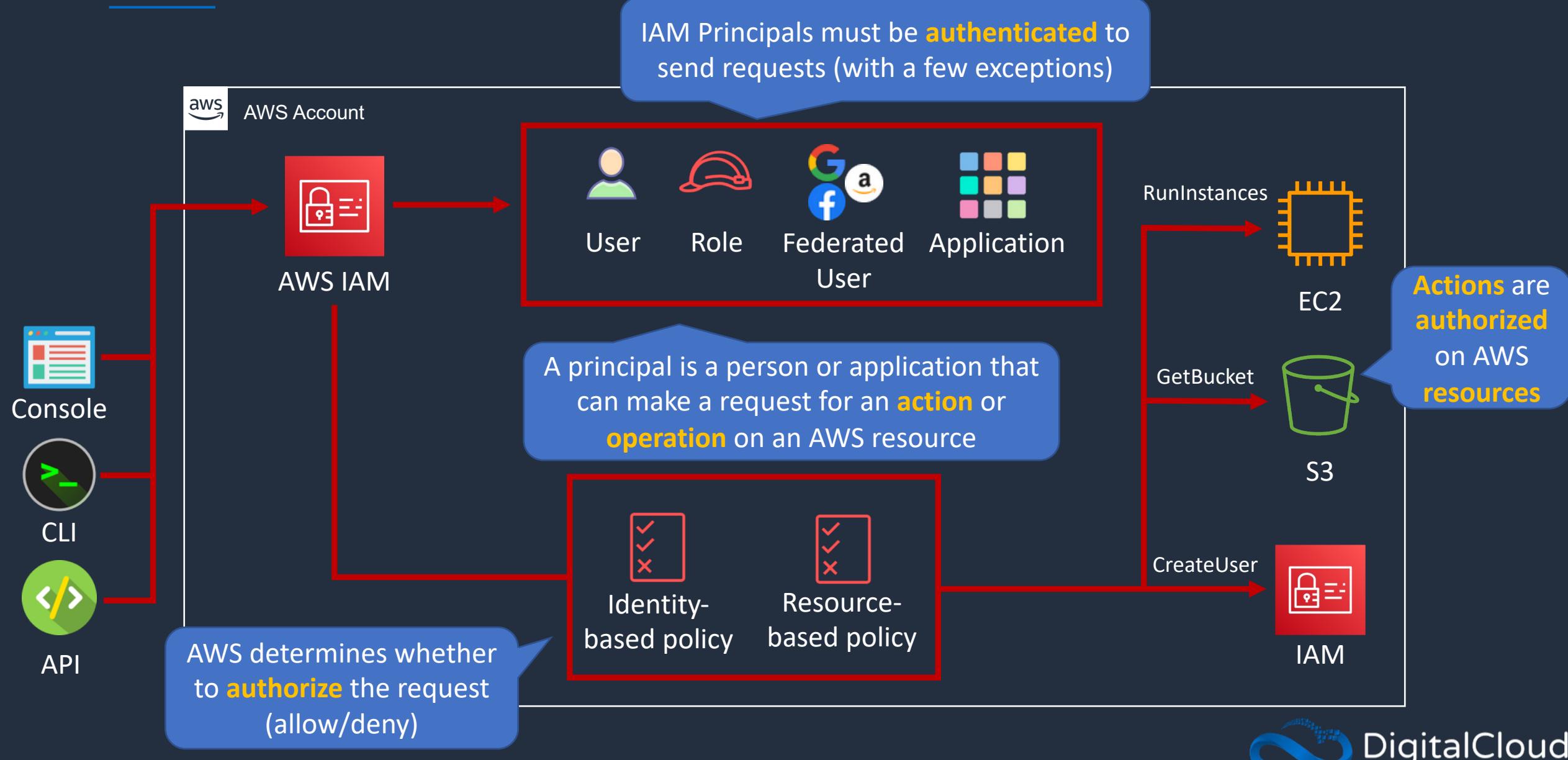
## AWS IAM Fundamentals

# AWS IAM Overview





# AWS Identity and Access Management (IAM)

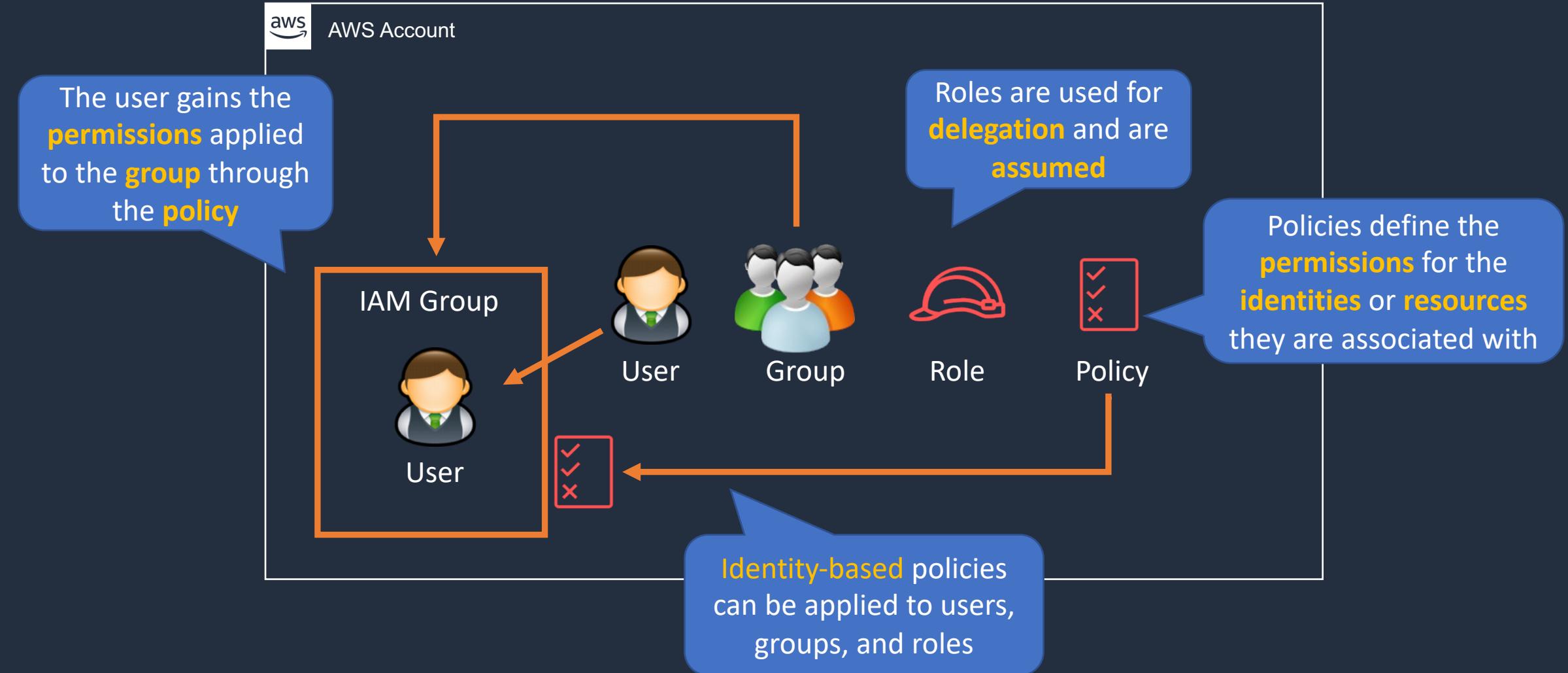


# IAM Users, Groups, Roles, and Policies



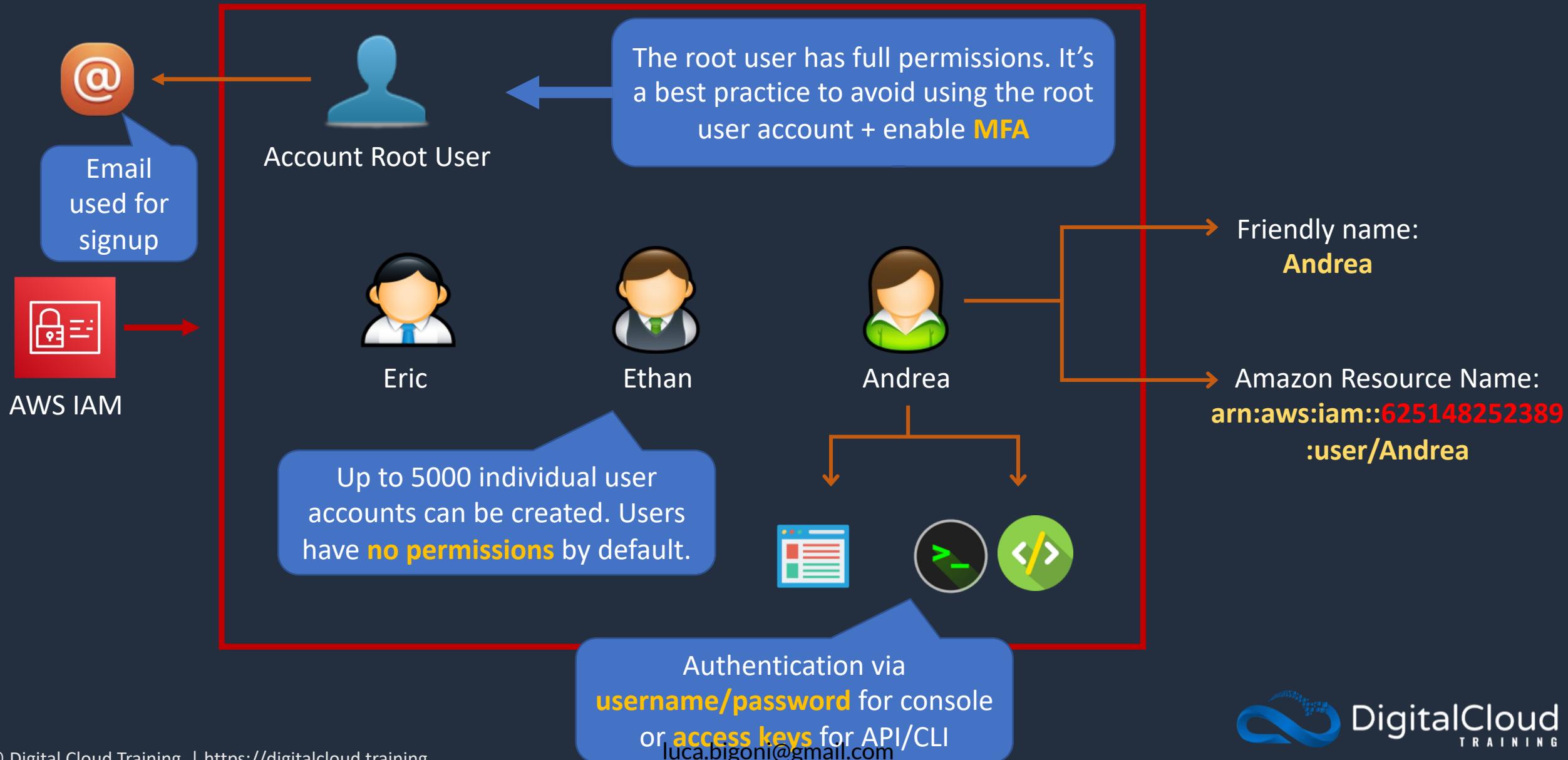


# Users, Groups, Roles and Policies



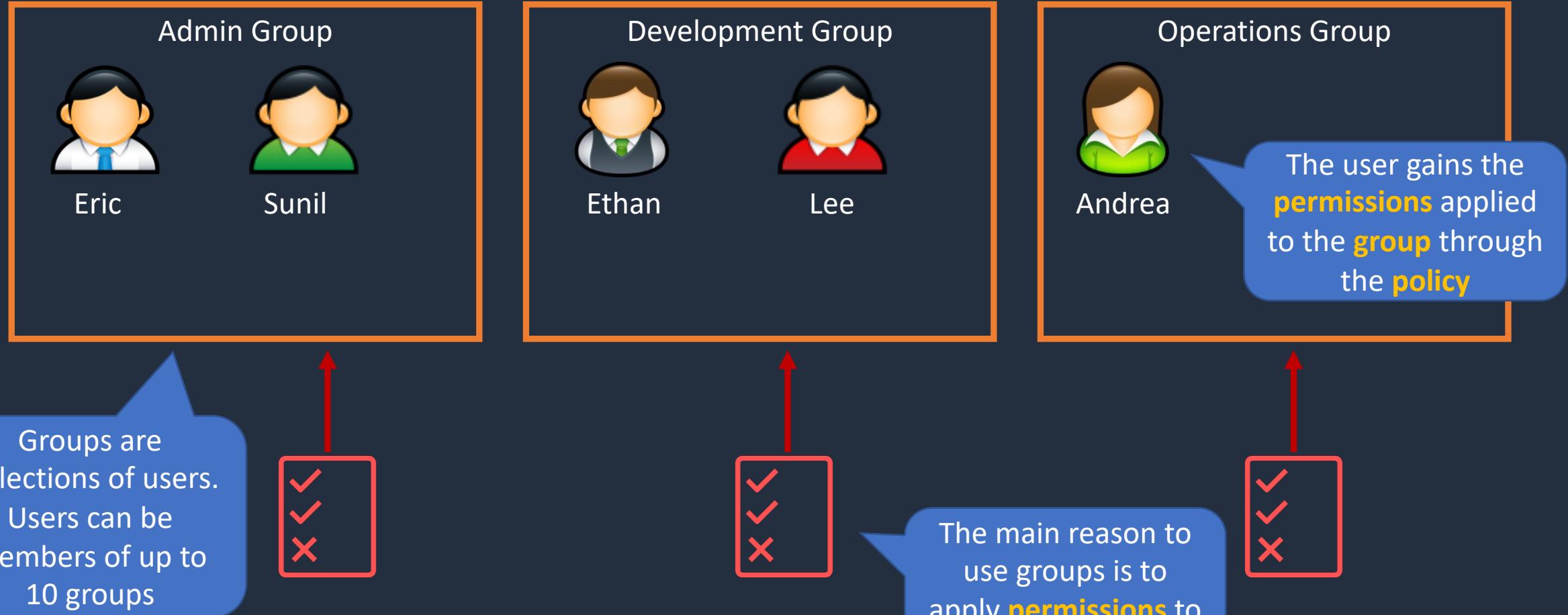


# IAM Users





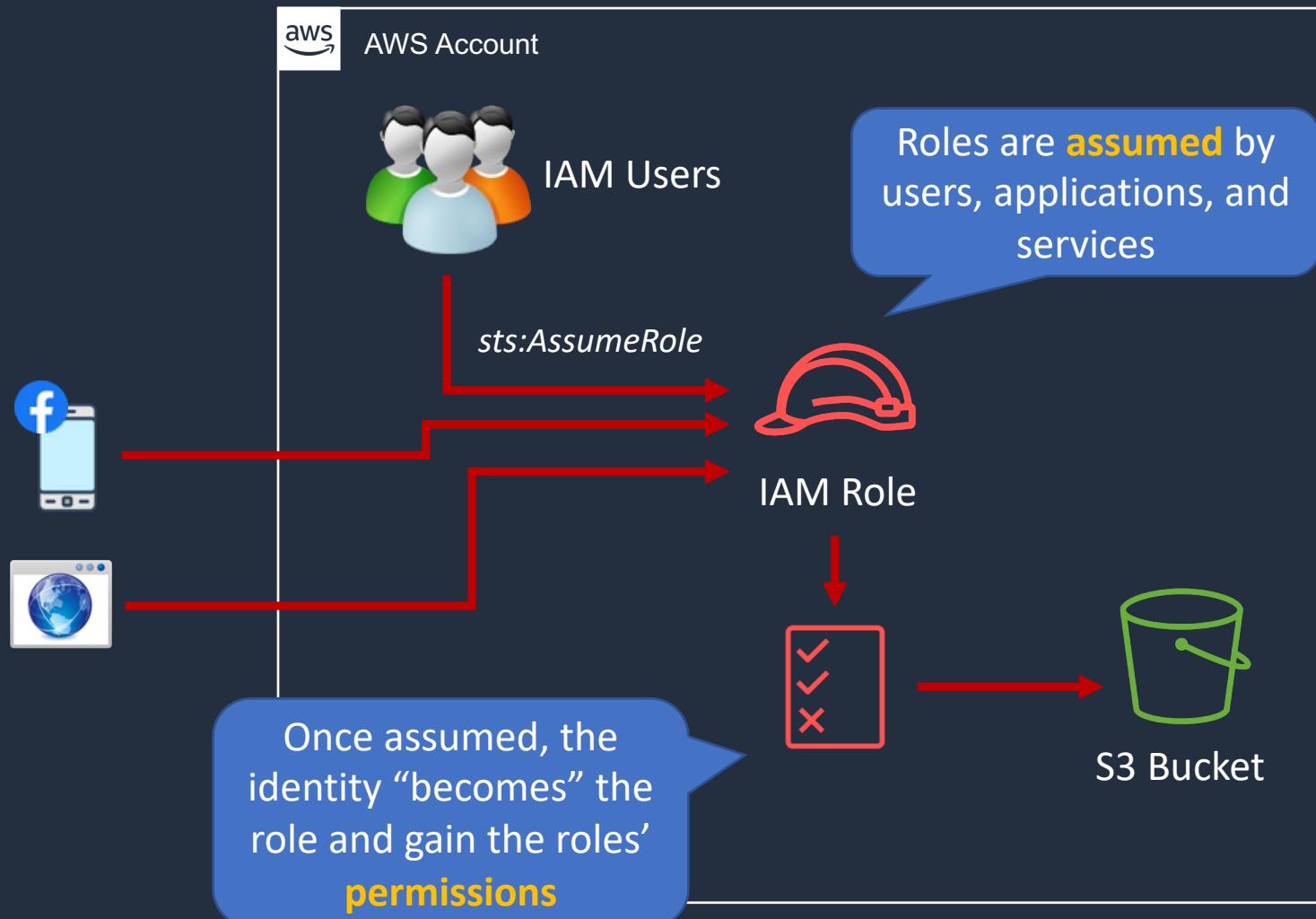
# IAM Groups





# IAM Roles

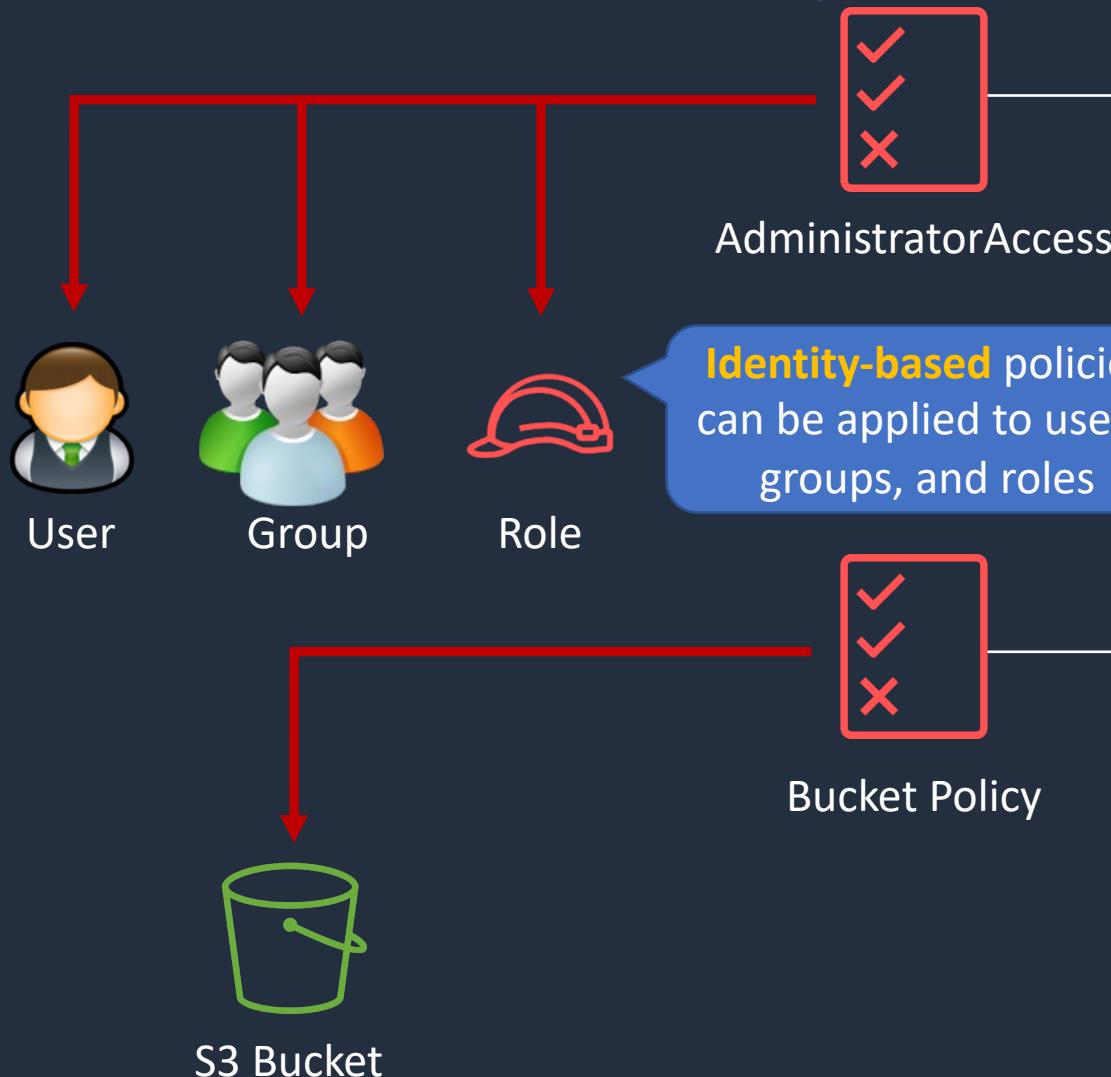
An **IAM role** is an IAM **identity** that that has specific **permissions**





# IAM Policies

Policies are **documents** that define **permissions** and are written in **JSON**



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

All permissions are **implicitly denied** by default

```
{  
  "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

# Create IAM User Account





# Root User vs IAM User

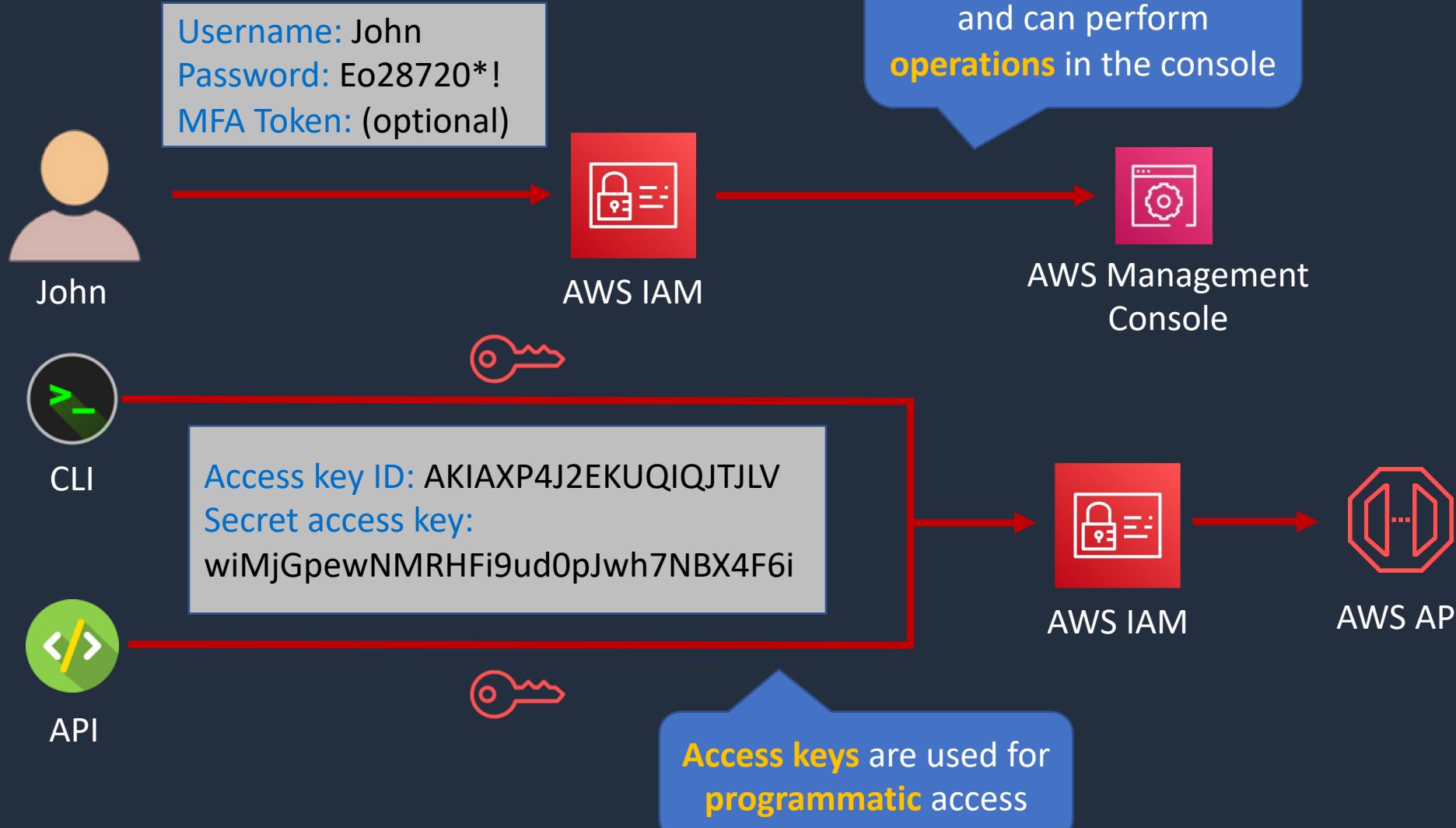
User	Login Details	Permissions
 Root User	 Email address	 Full - Unrestricted
 IAM User	Friendly name: <b>John</b> + AWS account ID or Alias	 IAM Permissions Policy

# IAM Authentication and MFA





# IAM Authentication Methods





# Multi-Factor Authentication

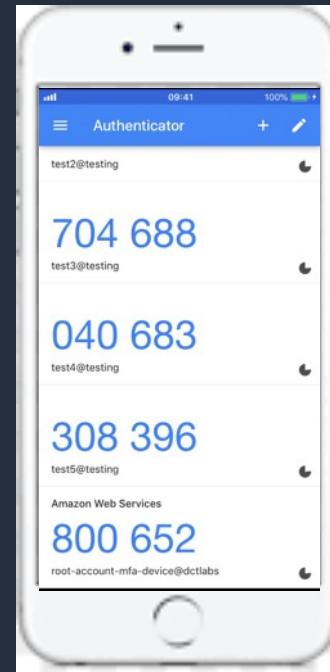
---

Something you **know**:

EJPx!\*21p9%

Password

Something you **have**:



Something you **are**:





# Multi-Factor Authentication

---

Something you **know**:



IAM User

EJPx!\*21p9%

Password

Something you **have**:



Virtual MFA

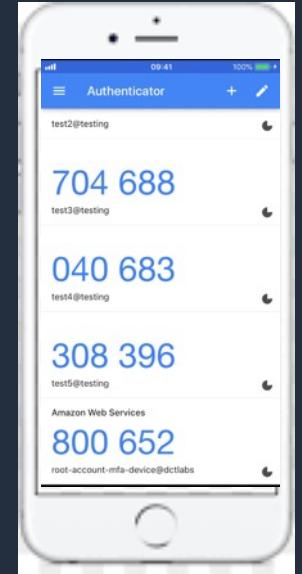


Physical MFA

e.g. Google Authenticator on  
your smart phone



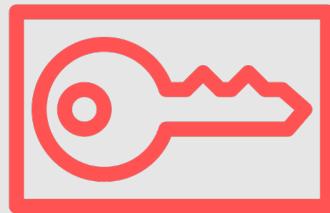
Physical tokens can  
be purchased from  
**third parties**



# Enable Multi-Factor Authentication (MFA)

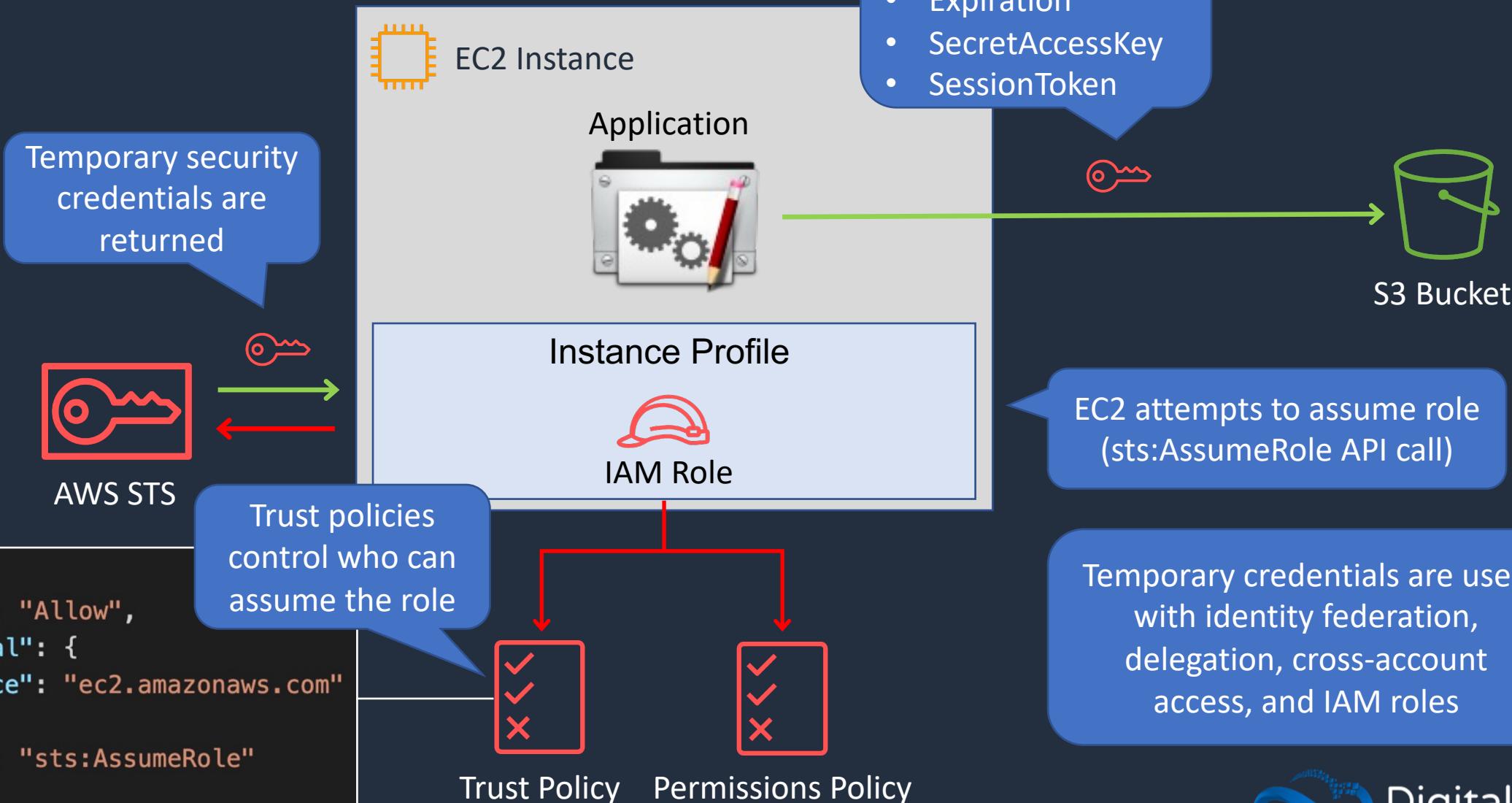


# AWS Security Token Service (STS)





# AWS Security Token Service (STS)



# IAM Password Policy



# SECTION 4

## IAM Access Control

# Identity-Based Policies and Resource-Based Policies





# 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



# Resource-Based Policies

---

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



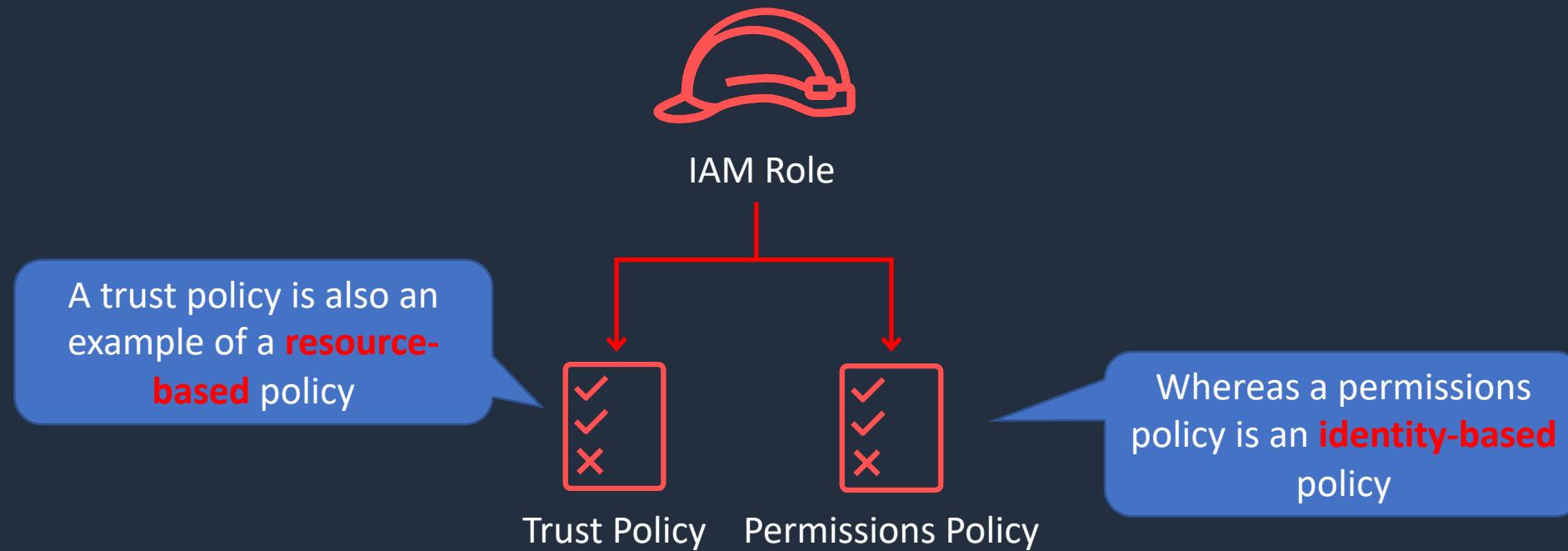
Resource-based policies grant the specified **principal** (Paul) **permission** to perform specific **actions** on the **resource**

```
{  
  "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

---

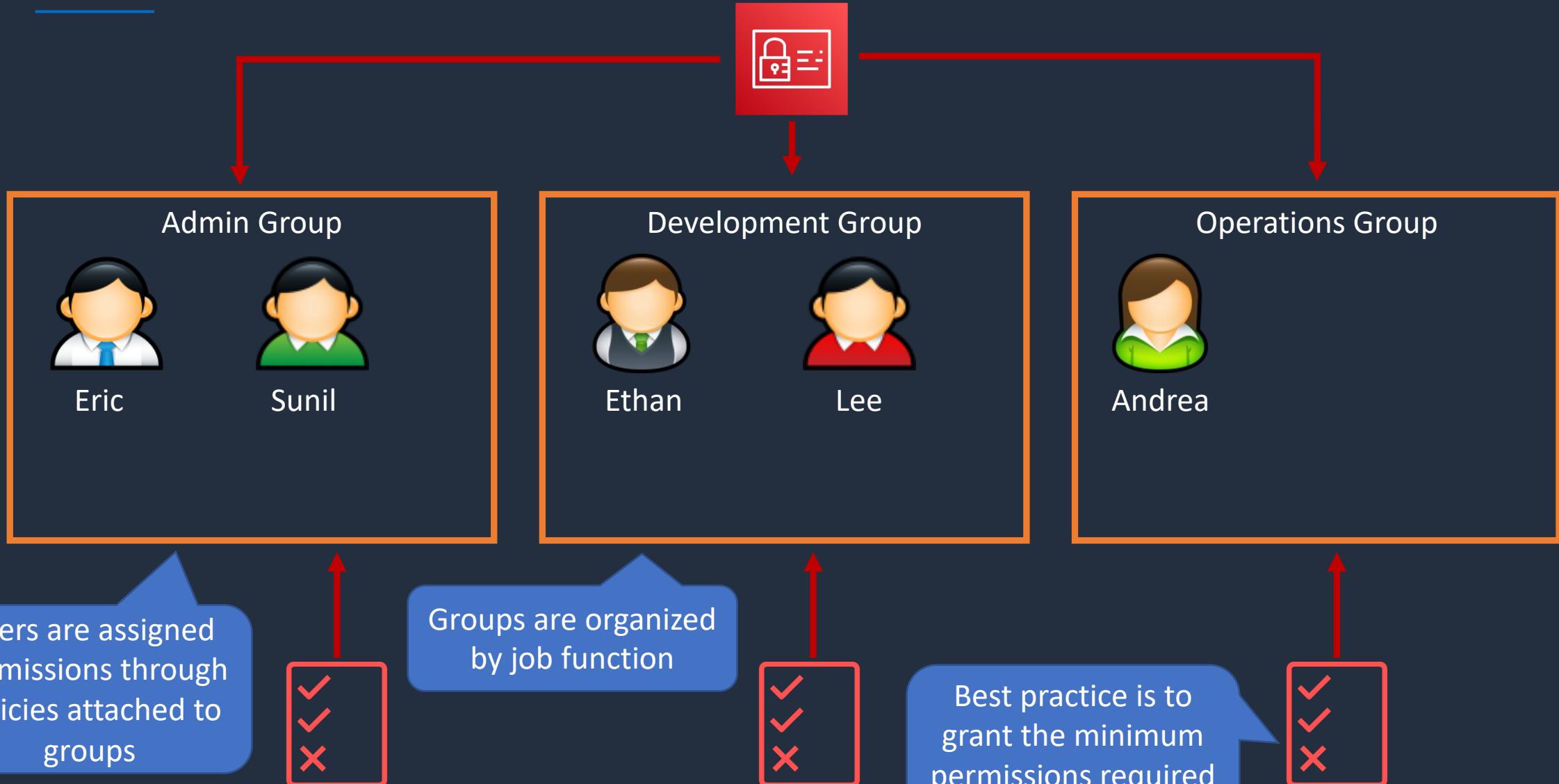


# Access Control Methods - RBAC & ABAC





# Role-Based Access Control (RBAC)



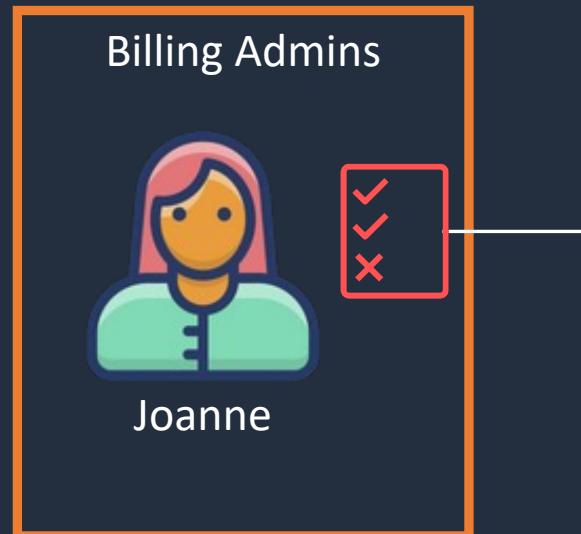


# Role-Based Access Control (RBAC)

## 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

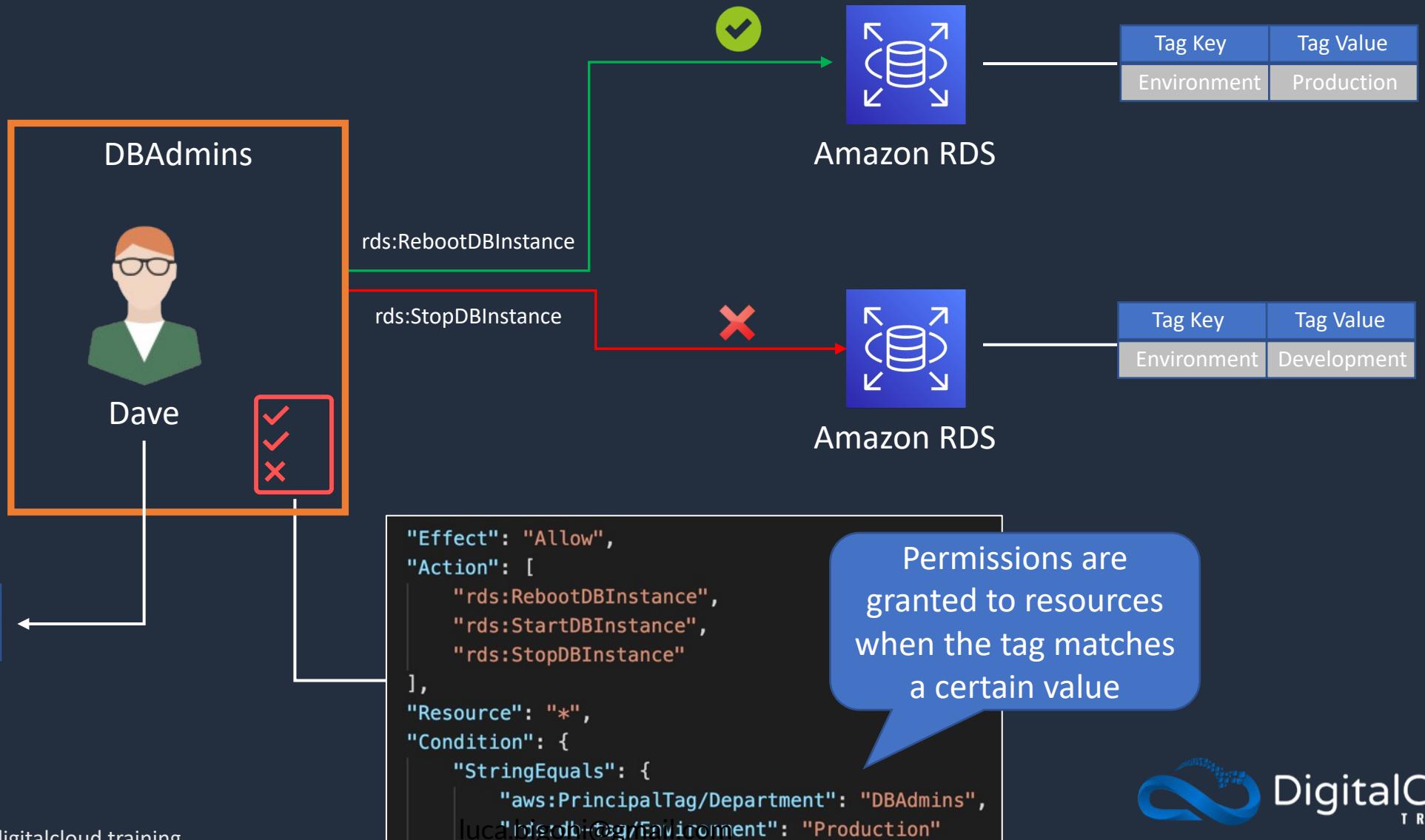


AWS managed policies for job functions are designed to closely align to common job functions in the IT industry

```
{  
    "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": "*"  
        }  
    ]  
}
```



# Attribute-Based Access Control (ABAC)

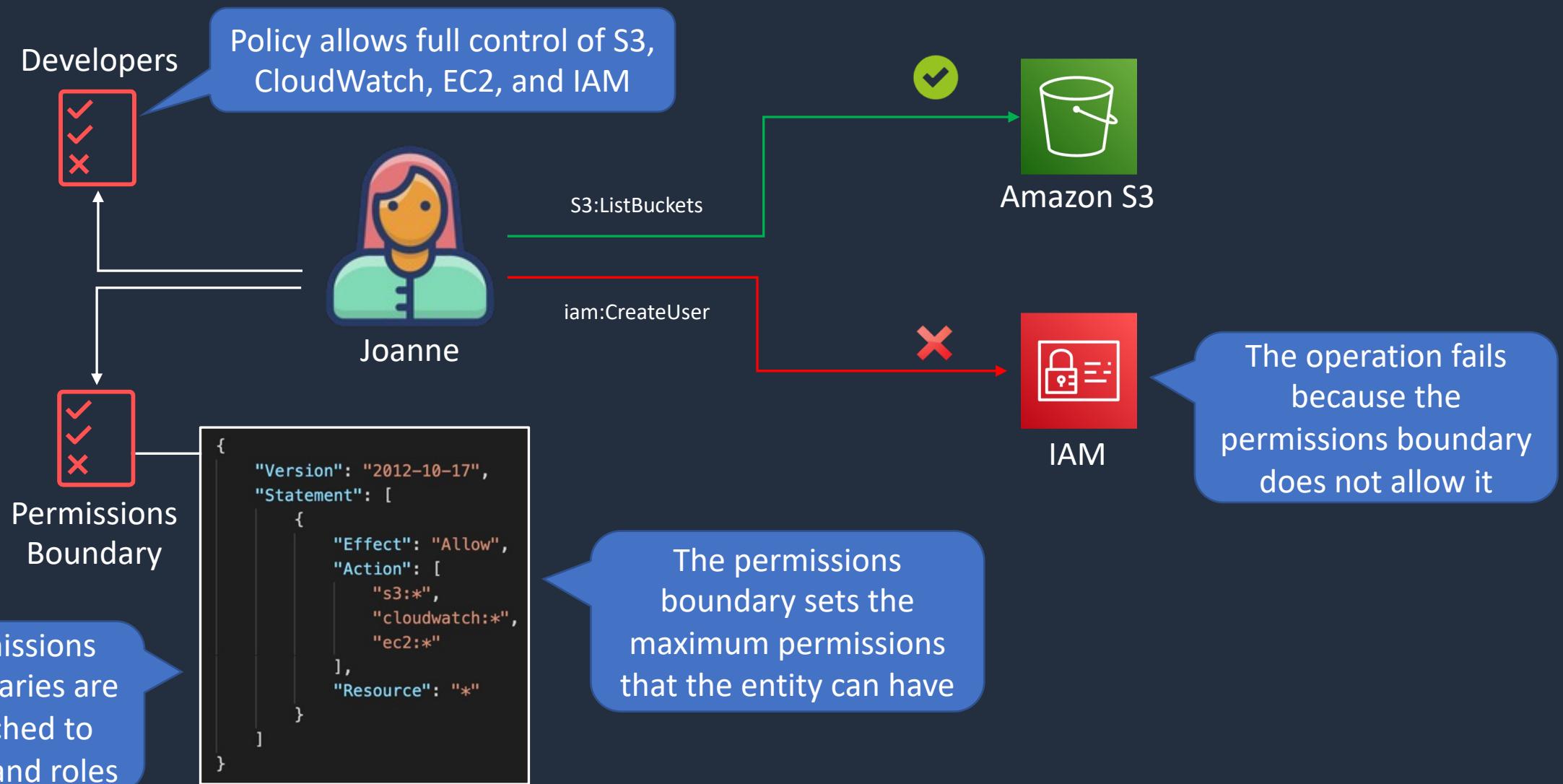


# Permissions Boundaries





# Permissions Boundaries





# Privilege Escalation

IAMFullAccess



Lindsay

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

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

luca.bigoni@gmail.com



# Preventing Privilege Escalation

IAMFullAccess



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

Permissions Boundary

The permissions boundary ensures that users created by Lindsay have the same or fewer permissions

iam:CreateUser



IAM

Lindsay applies the AdministratorAccess policy to the X-User account

AdministratorAccess



luca.bigoni@gmail.com

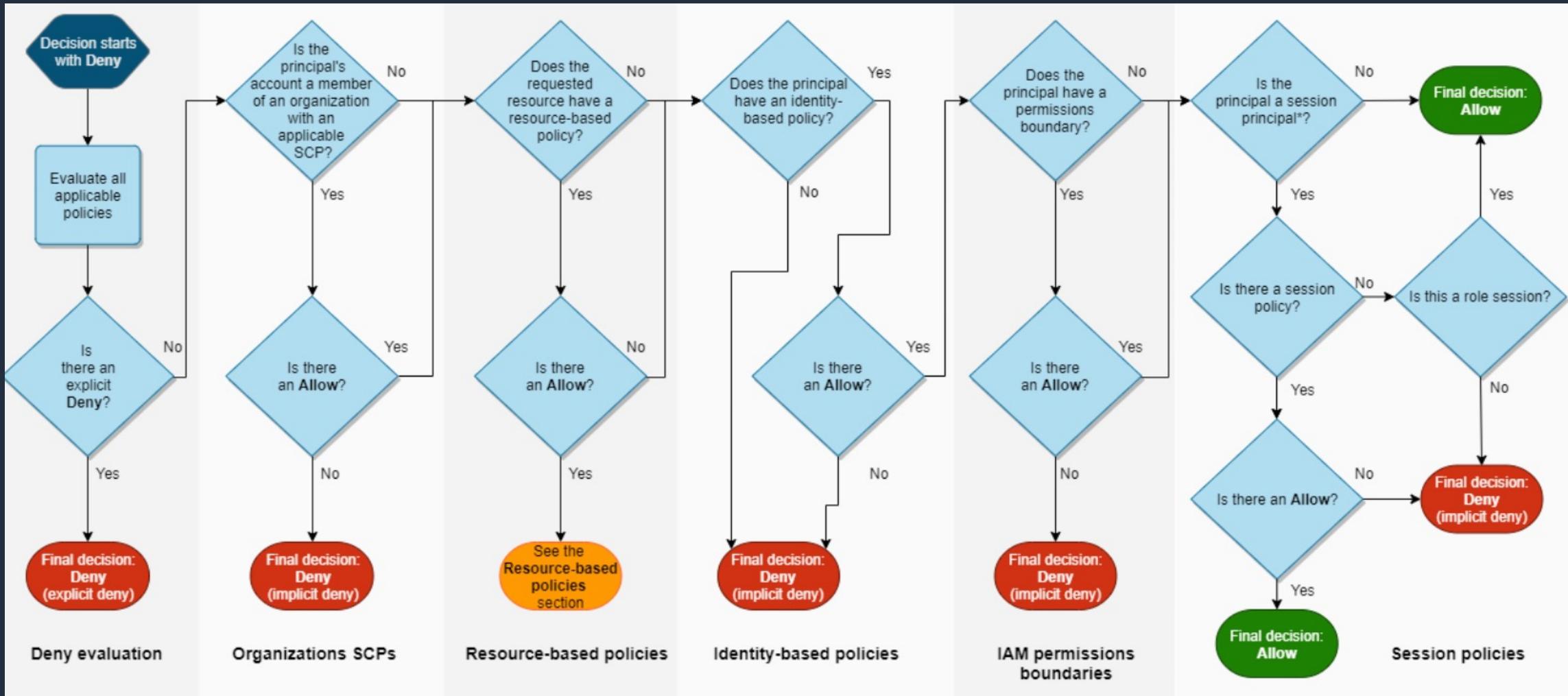
Lindsay does not have more privileges when logging in as X-User and cannot launch AWS resources

# IAM Policy Evaluation





# Evaluation Logic





# Steps for Authorizing Requests to AWS

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



AWS IAM

- 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

**2. Processing** the request context

luca.bigoni@gmail.com

**3. Evaluating** all policies within the account



Resource-based policy



S3 Bucket

**4. Determining** whether a request is **allowed** or **denied**

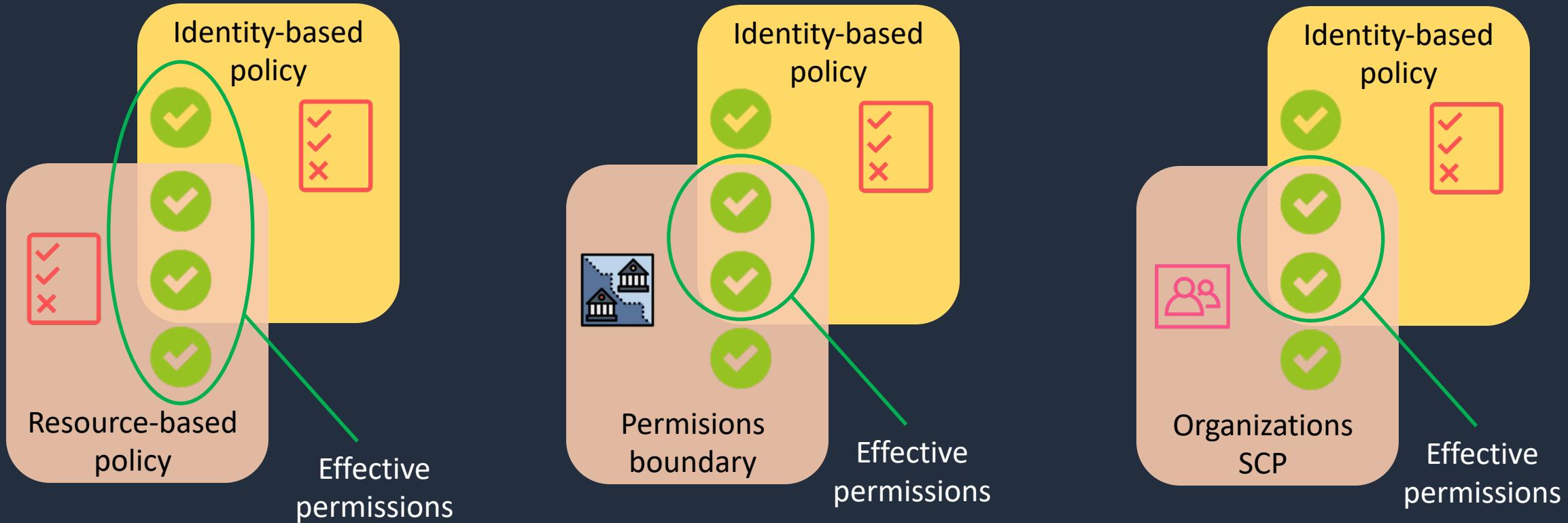


# Types of Policy

- **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

# IAM Policy Structure





# IAM Policy Structure

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

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

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

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

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": "ExampleStatement01",  
            "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

# Using Role-Based Access Control (RBAC)



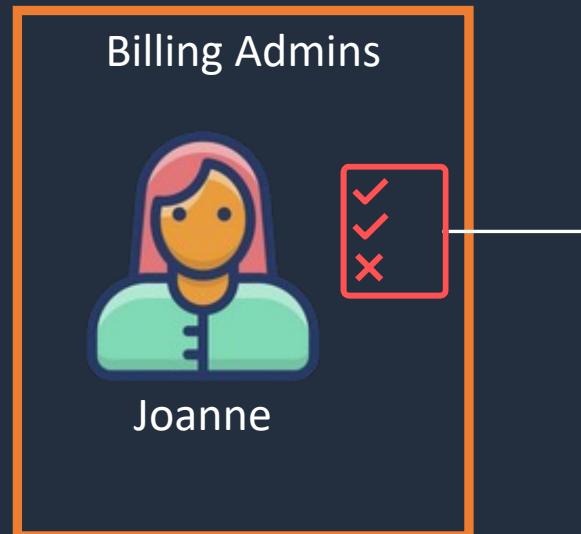


# Role-Based Access Control (RBAC)

## 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



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

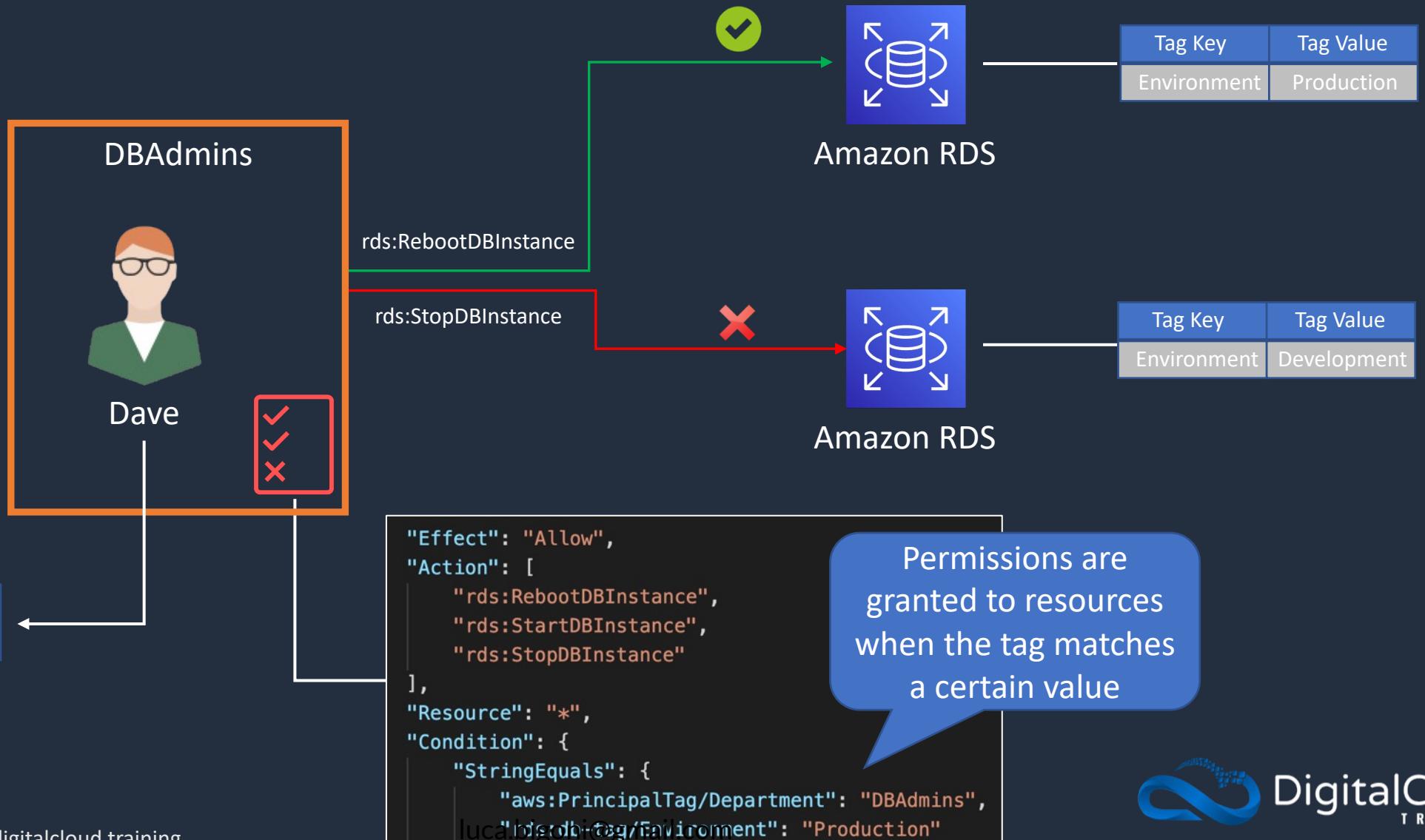
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "aws-portal:*Billing",  
                "aws-portal:*Usage",  
                "aws-portal:*PaymentMethods",  
                "budgets:ViewBudget",  
                "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": "*"  
        }  
    ]  
}
```

# Using Attribute-Based Access Control (ABAC)





# Attribute-Based Access Control (ABAC)



# Apply Permissions Boundary





# Permissions 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



# Privilege Escalation

IAMFullAccess



Lindsay

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

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

# AWS Policy Generator



luca.bigoni@gmail.com

# IAM Policy Simulator



luca.bigoni@gmail.com



# 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

# IAM Best Practices





# AWS IAM Best Practices

- Lock away your AWS account root user access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Get started using permissions with AWS managed policies
- Use customer managed policies instead of inline policies
- Use access levels to review IAM permissions
- Configure a strong password policy for your users
- Enable MFA



# AWS IAM Best Practices

---

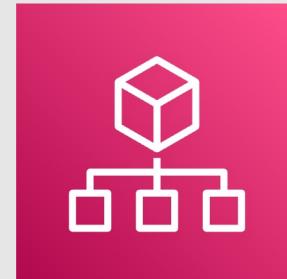
---

- Use roles for applications that run on Amazon EC2 instances
- Use roles to delegate permissions
- Do not share access keys
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

# SECTION 5

## AWS Organizations and Control Tower

# AWS Organizations





# AWS Organizations

---

---

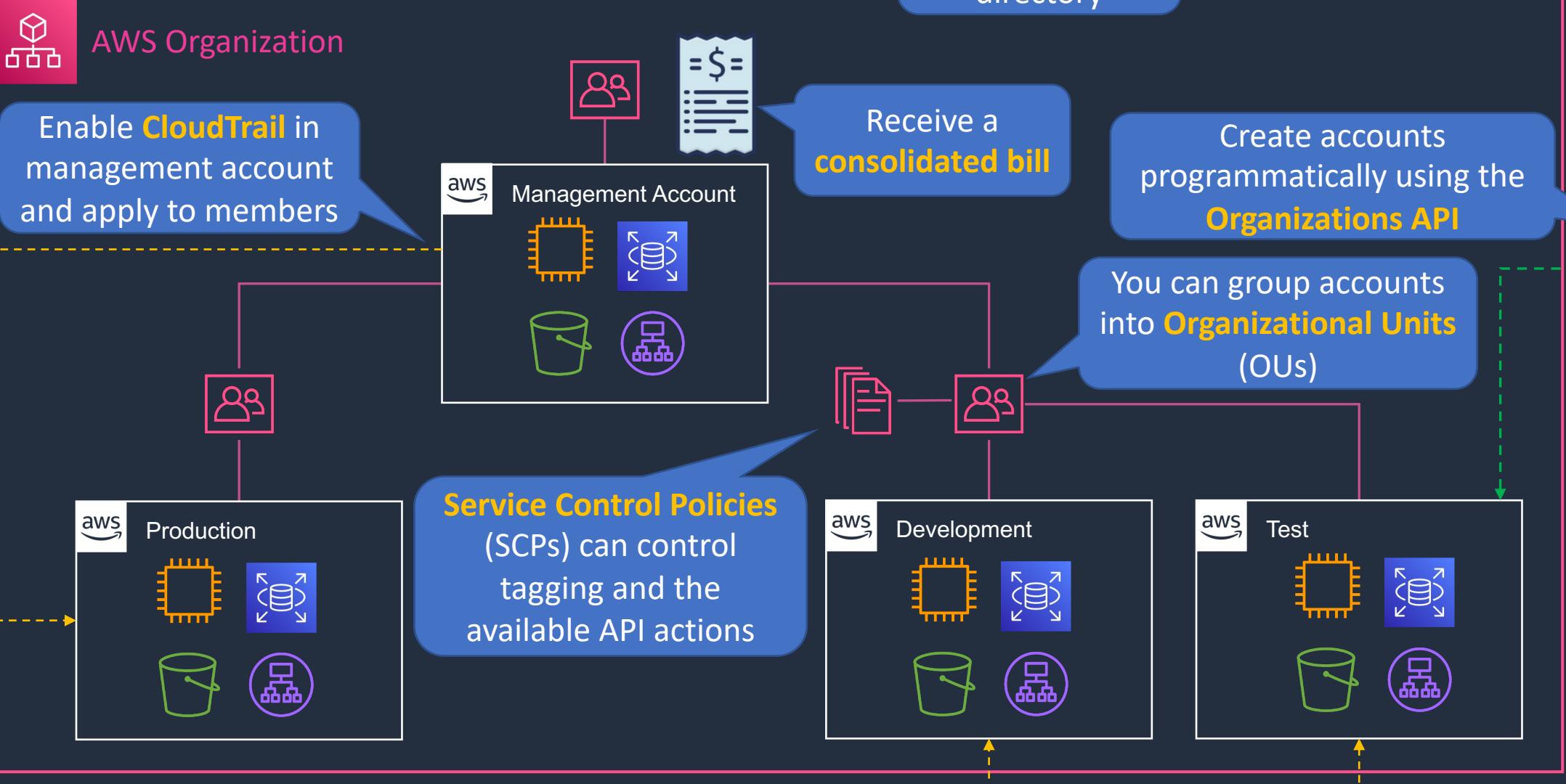
- AWS organizations allows you to consolidate multiple AWS accounts into an organization that you create and centrally manage
- Available in two feature sets:
  - **Consolidated Billing**
  - **All features**
- Includes root accounts and organizational units
- Policies are applied to root accounts or OUs
- Consolidated billing includes:
  - **Paying Account** – independent and cannot access resources of other accounts
  - **Linked Accounts** – all linked accounts are independent



# AWS Organizations



Enable AWS SSO  
using on-prem  
directory

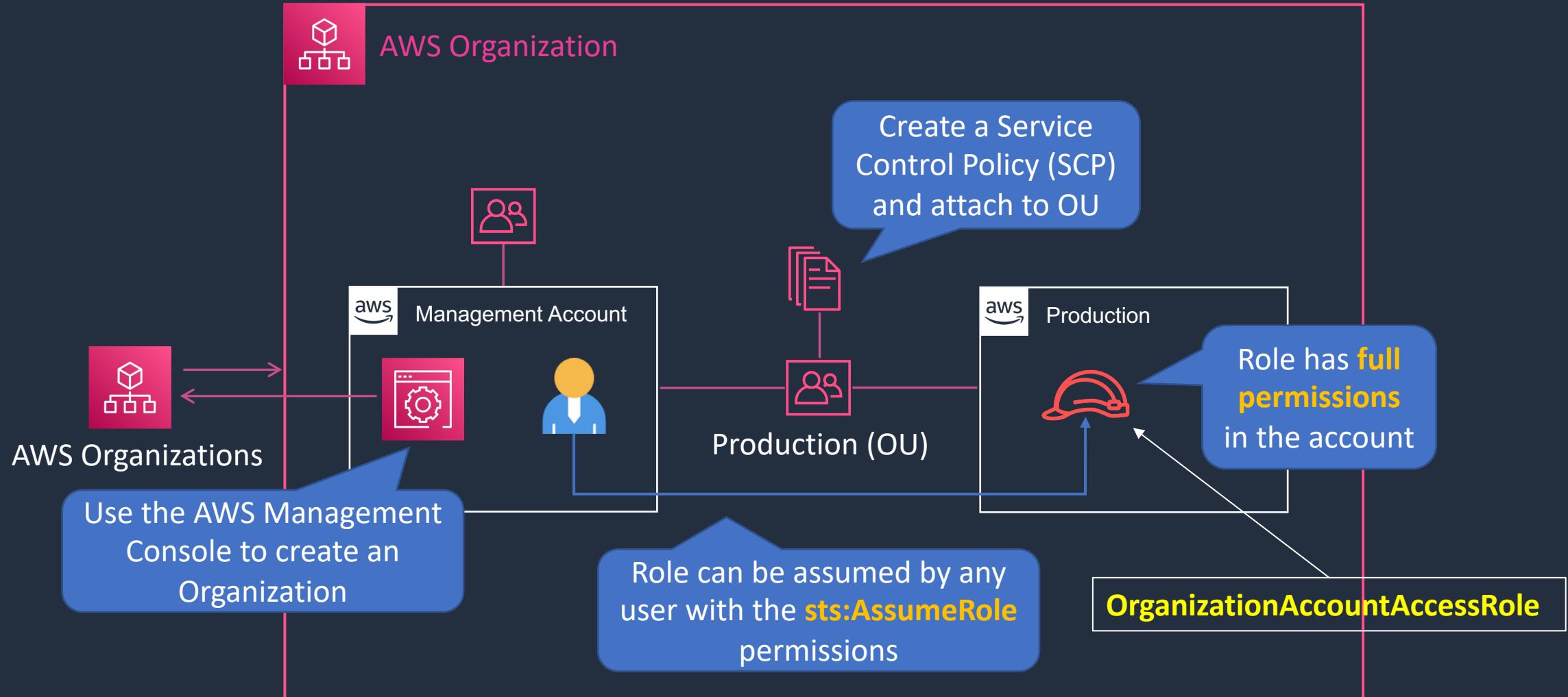


# Create AWS Organization and Add Account

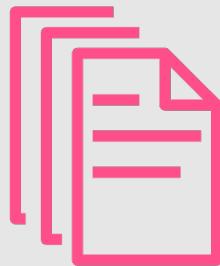




# Account Configuration

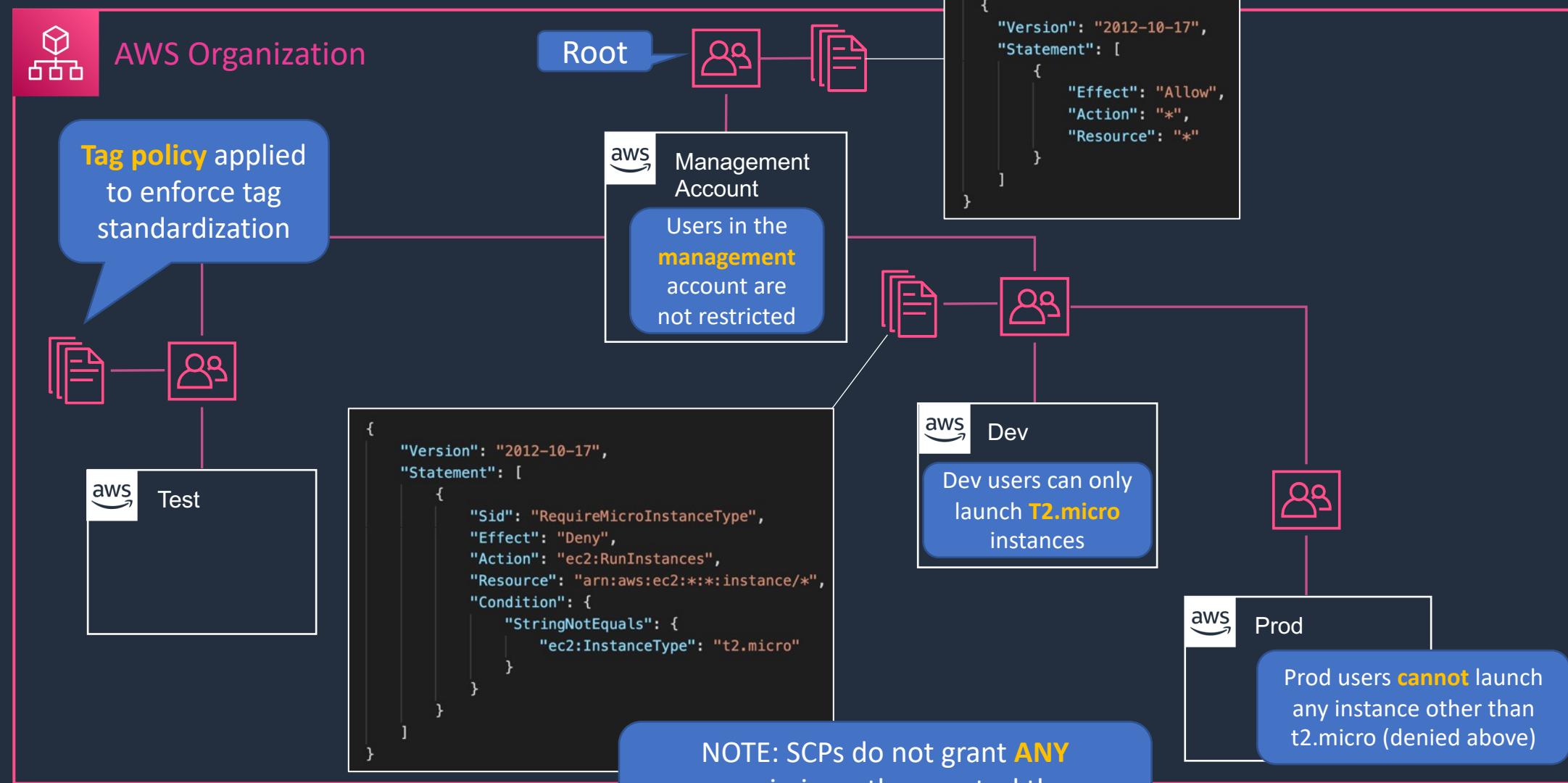


# Service Control Policies (SCPs)



# Service Control Policies

SCPs control the maximum available permissions



# Create Service Control Policy (SCP)

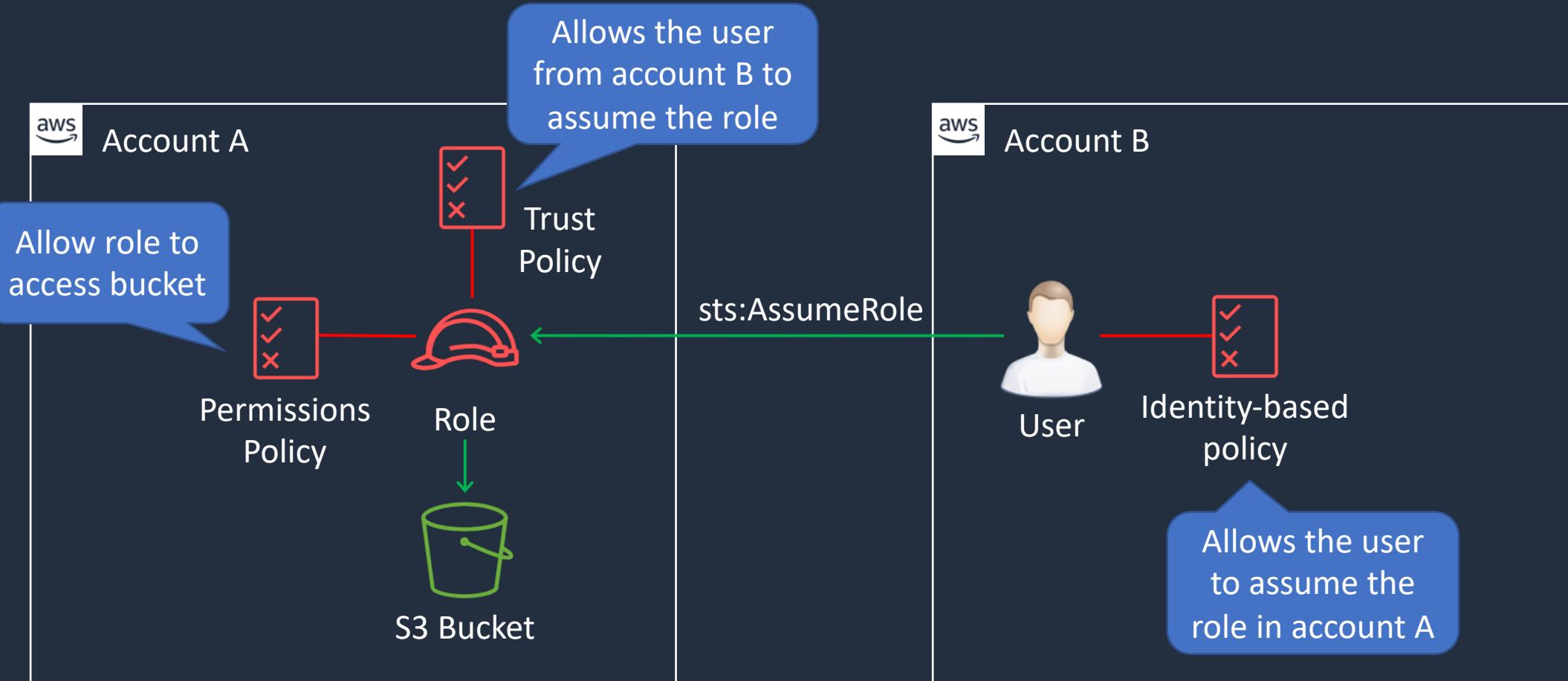


# Use Cases for IAM Roles





# Use Case: Cross Account Access

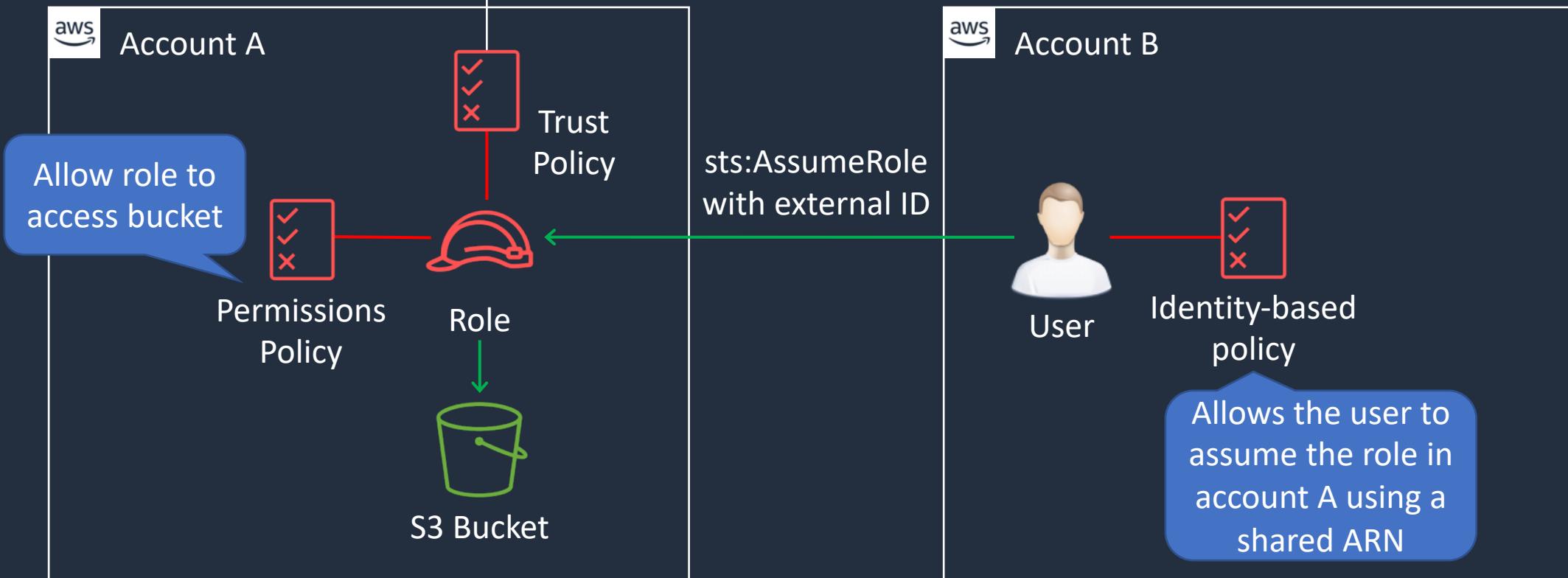




# Use Case: Cross Account Access (3<sup>rd</sup> Party)

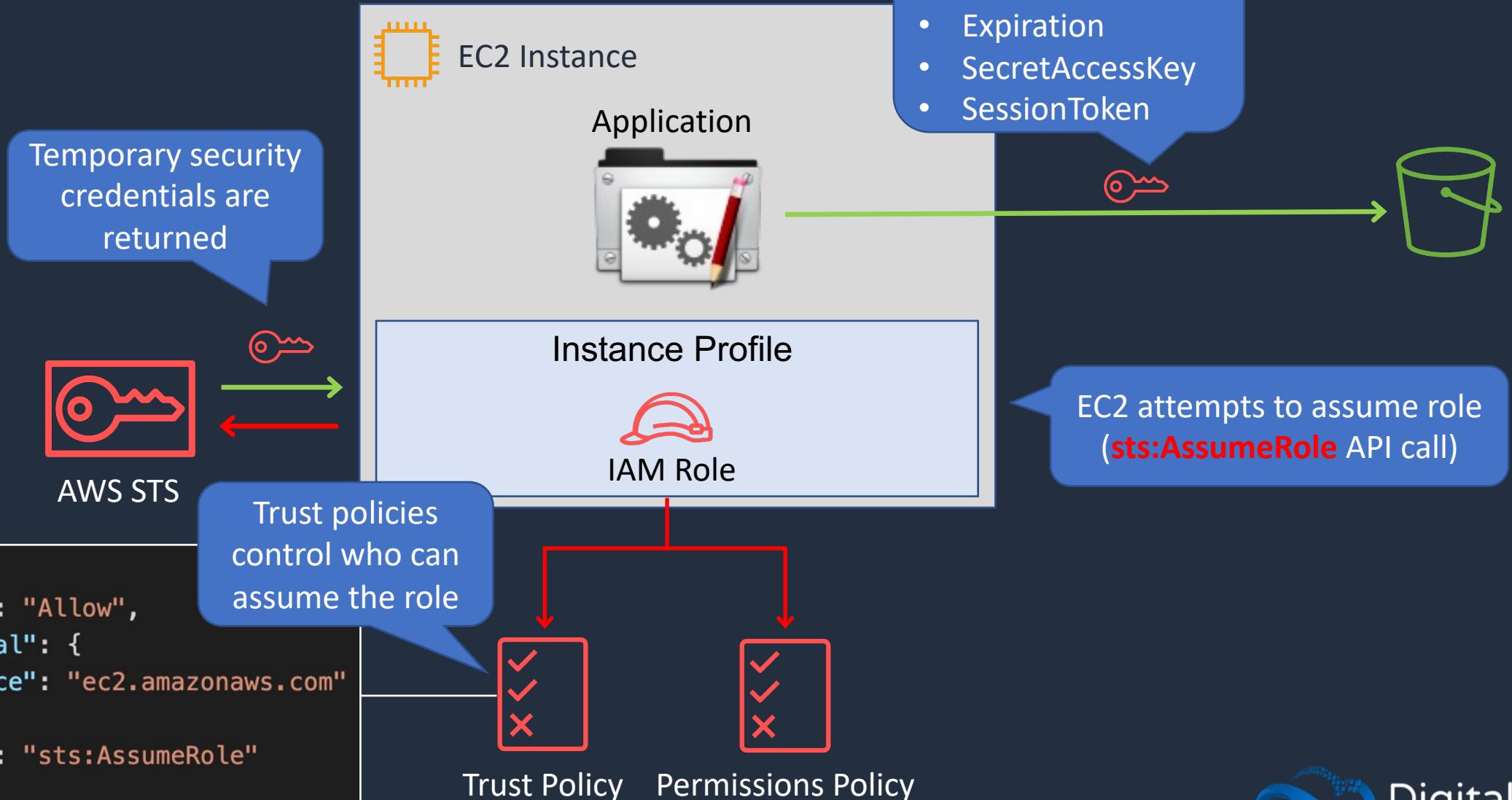
The trust policy condition requires the external ID

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





# Use Case: Delegation to AWS Services



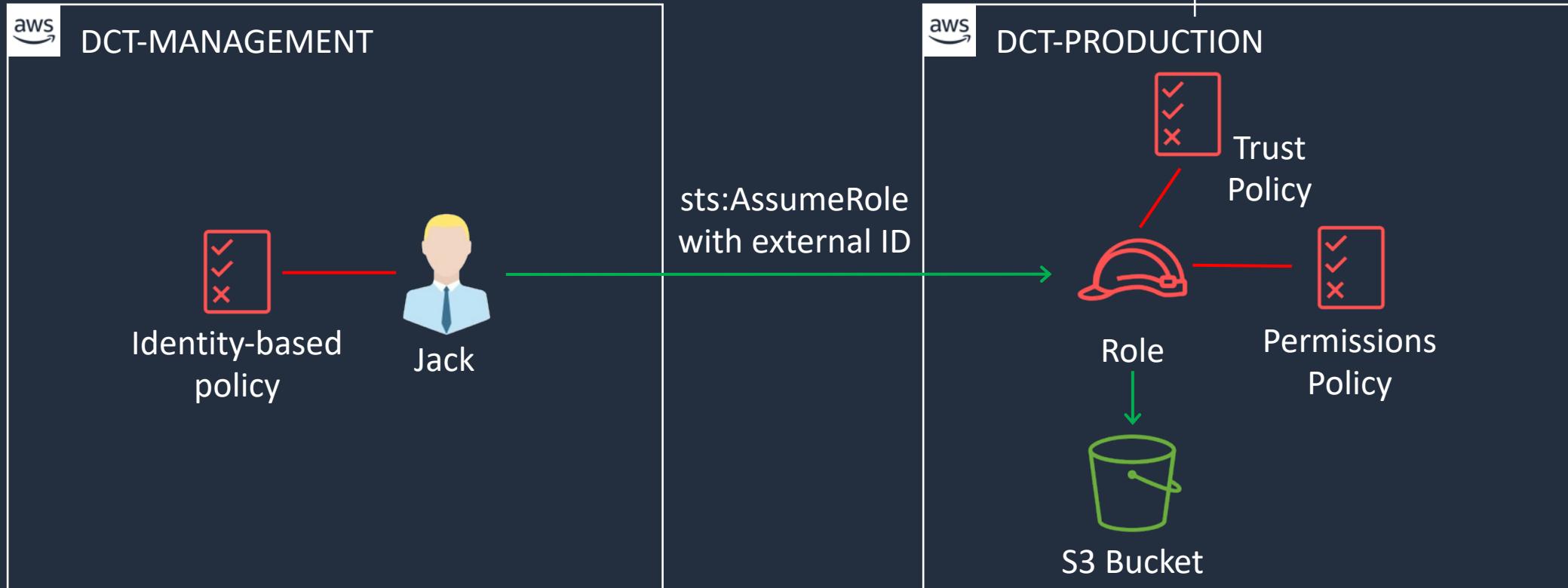
# Cross-Account Access to S3



luca.bigoni@gmail.com



# Cross Account Access (3<sup>rd</sup> Party) Hands-On



# AWS Control Tower

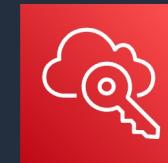


luca.bigoni@gmail.com



# AWS Control Tower

Directory source can be SSO, SAML 2.0 IdP, or Microsoft AD

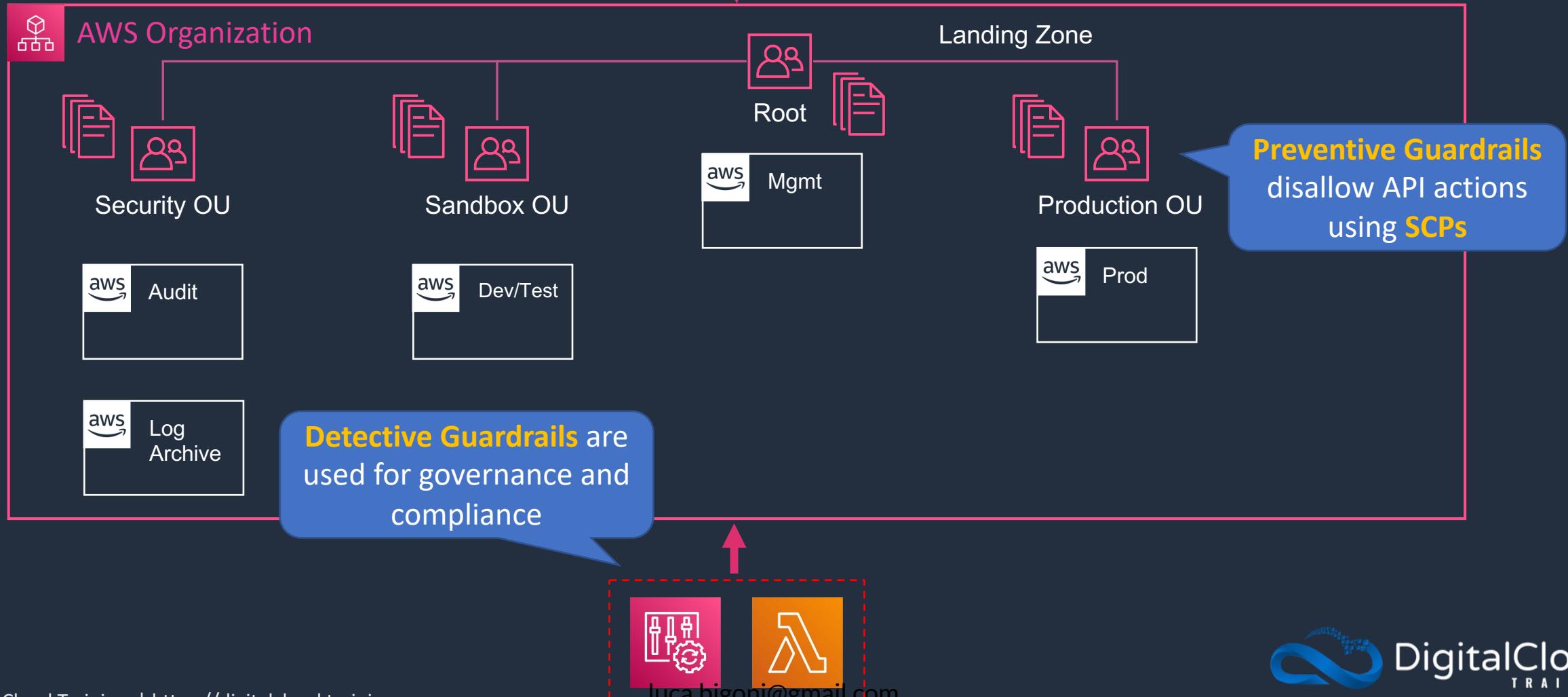


AWS Single Sign-On



AWS Control Tower

A **landing zone** is a well-architected multi-account baseline





# AWS Control Tower

- Control Tower creates a well-architected multi-account baseline based on best practices
- This is known as a landing zone
- Guardrails are used for governance and compliance:
  - **Preventive guardrails** are based on SCPs and disallow API actions
  - **Detective guardrails** are implemented using AWS Config rules and Lambda functions and monitor and govern compliance
- The root user in the management account can perform actions that guardrails would disallow

# Create a Landing Zone



luca.bigoni@gmail.com

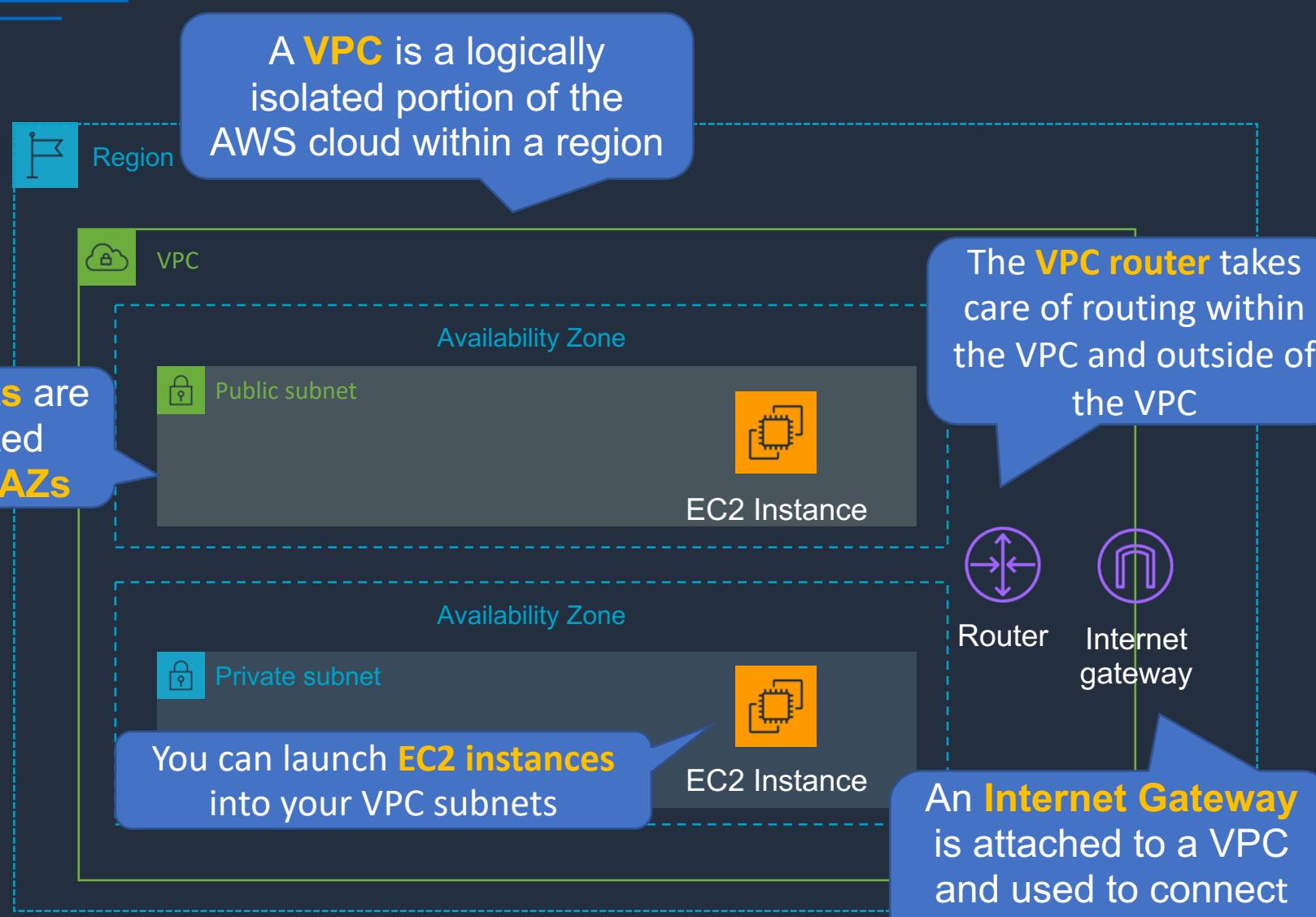
# SECTION 6

## Infrastructure Security

# Secure VPC Design



# Amazon VPC Refresher





# Using IPv4 in a VPC



Region

Each **VPC** has a different block of IPv4 addresses

**CIDR** stands for Classless Interdomain Routing



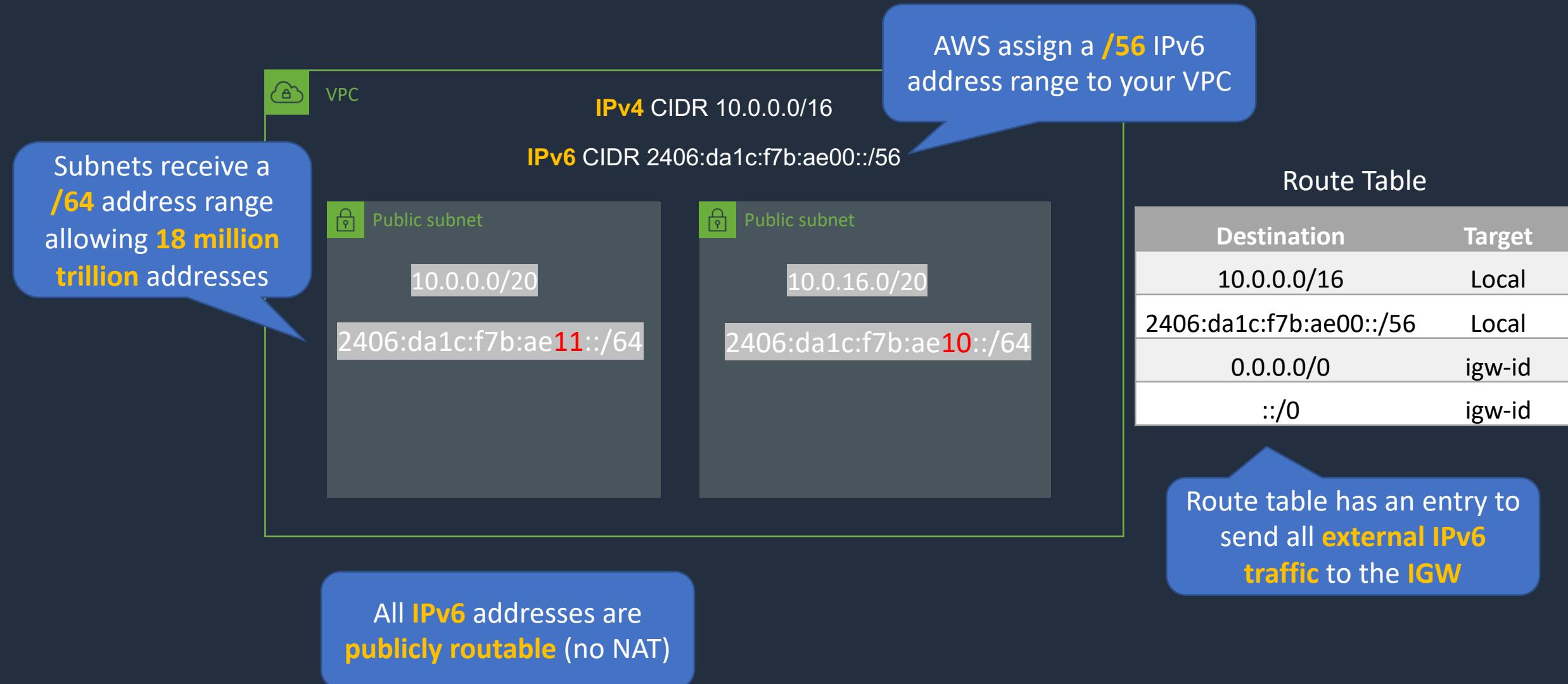
Each subnet has a block of **IP addresses** from the CIDR block



You can create **multiple VPCs** within each region



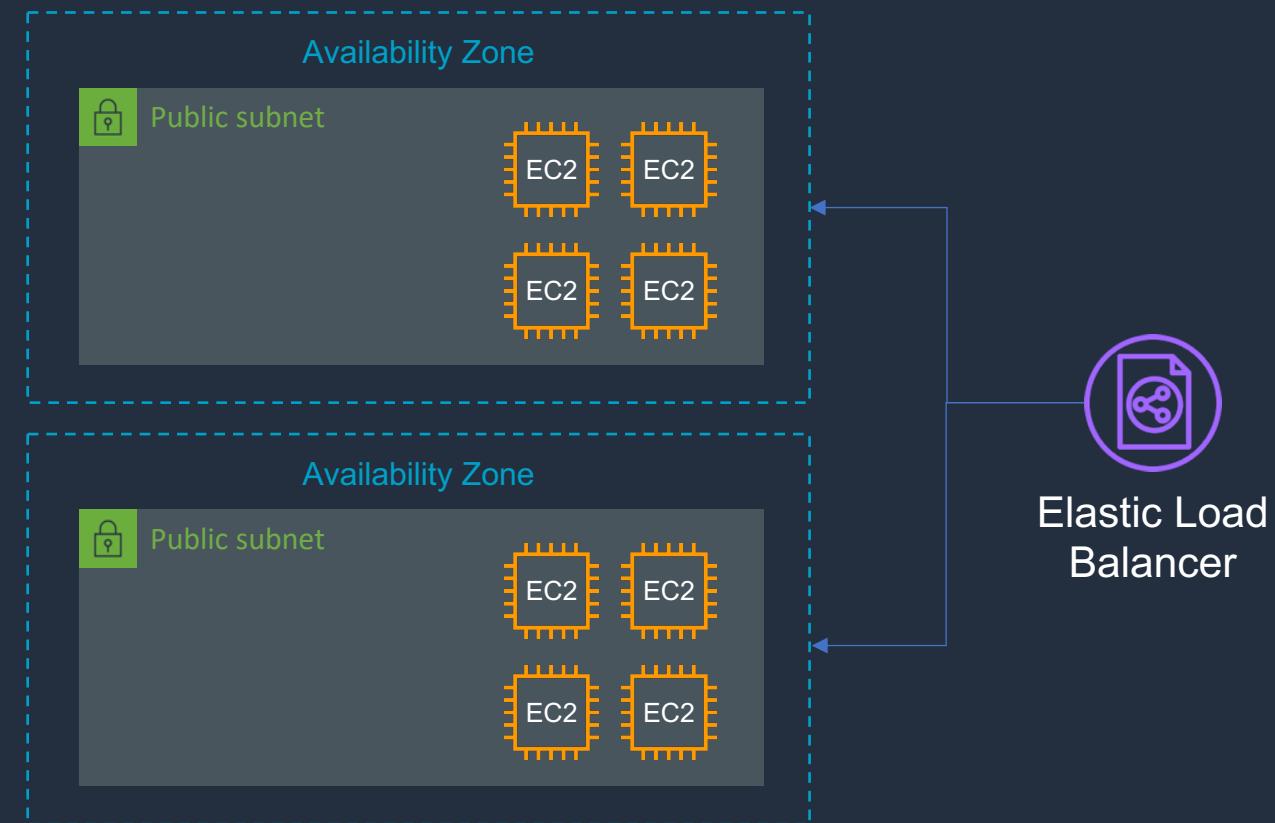
# Using IPv6 in a VPC





# Amazon VPC Best Practices

- Use multiple AZs for High Availability (HA)





# Amazon VPC Best Practices

---

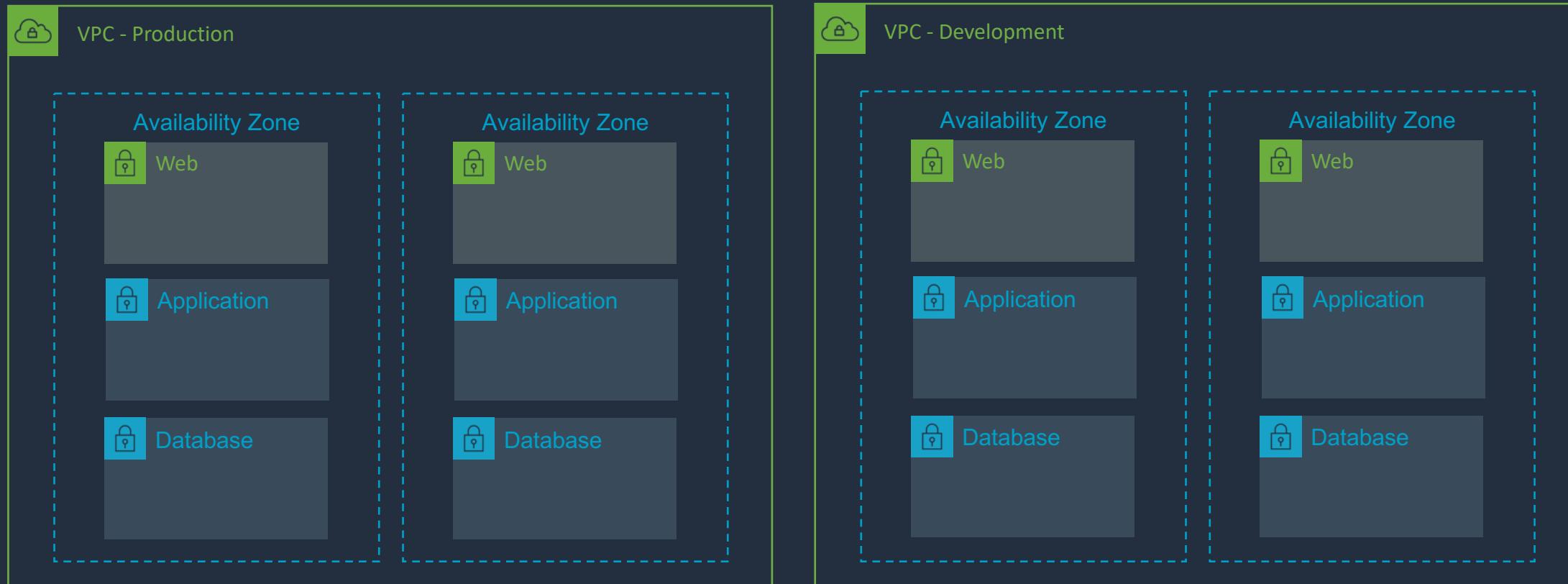
---

- Control traffic with **Security Groups** and **NACLs**
- Use IAM policies to control access
- Use Amazon CloudWatch to monitor VPC components
- Use VPC Flow Logs to capture IP traffic



# Amazon VPC Best Practices

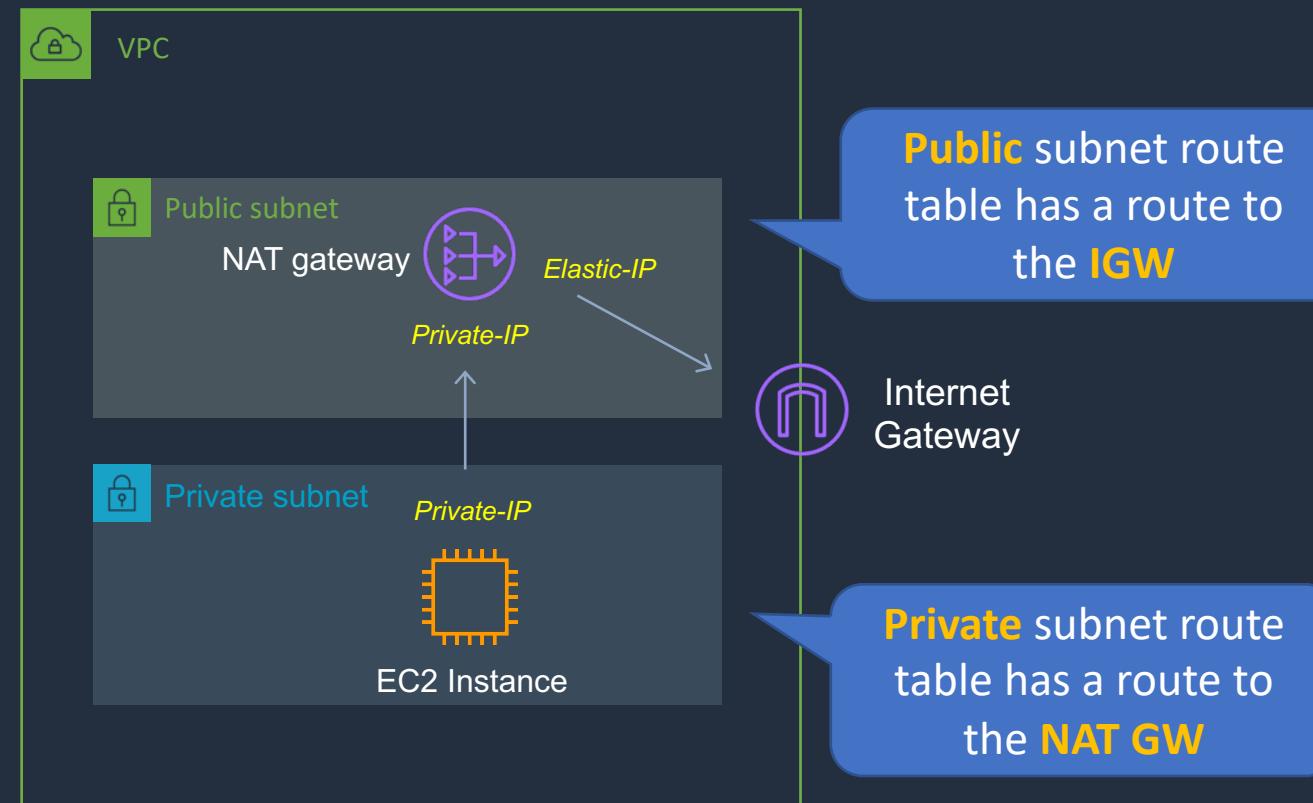
- Use separate VPCs to isolate infrastructure
- Use subnets to isolate the tiers of your application





# Amazon VPC Best Practices

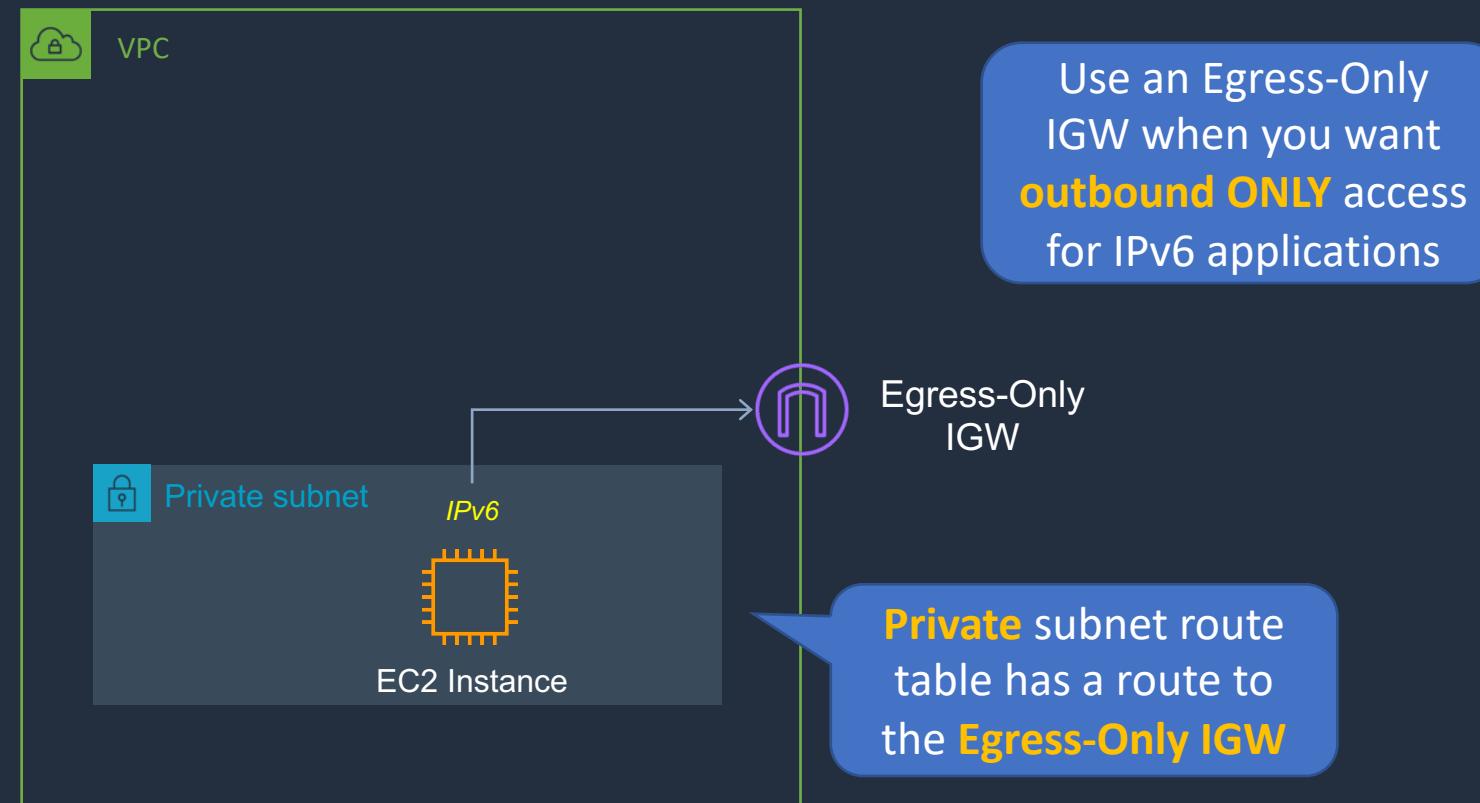
- Use AWS PrivateLink to keep traffic private
- Use private subnets for your instances if they should not be accessed directly from the internet





# Amazon VPC Best Practices

- Use AWS PrivateLink to keep traffic private
- Use private subnets for your instances if they should not be accessed directly from the internet

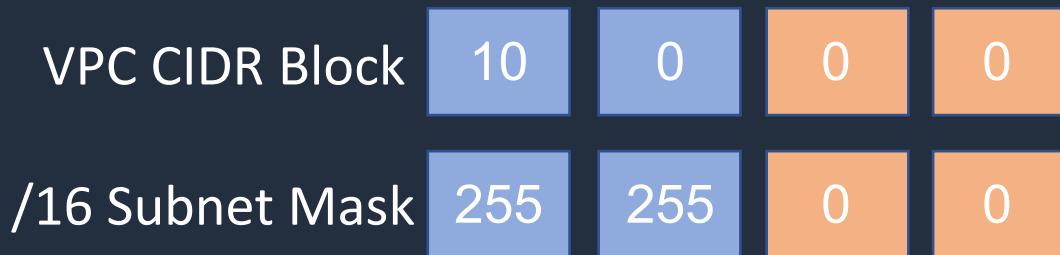


# Create a Custom VPC with Subnets





# VPC CIDR Block and Subnets



Subnet Name	IPv4 CIDR block	Availability Zone	Route Table	Auto-assign Public IPv4
private-1a	10.0.3.0/24	us-east-1a	Private-RT	No
private-1b	10.0.4.0/24	us-east-1b	Private-RT	No
public-1a	10.0.1.0/24	us-east-1a	MAIN	Yes
public-1b	10.0.2.0/24	us-east-1b	MAIN	Yes

Has a route to an  
**Internet Gateway**

Automatically assign  
**IPv4 Public  
addresses**

# Launch Instances and Test VPC



# Stateful and Stateless Firewalls





# Stateful vs Stateless Firewalls

PROTOCOL	SOURCE IP	DESTINATION IP	SOURCE PORT	DESTINATION PORT
HTTP	10.1.1.1	10.2.1.10	65188	80
HTTP	10.2.1.10	10.1.1.1	80	65188



A **stateful** firewall allows the return traffic automatically

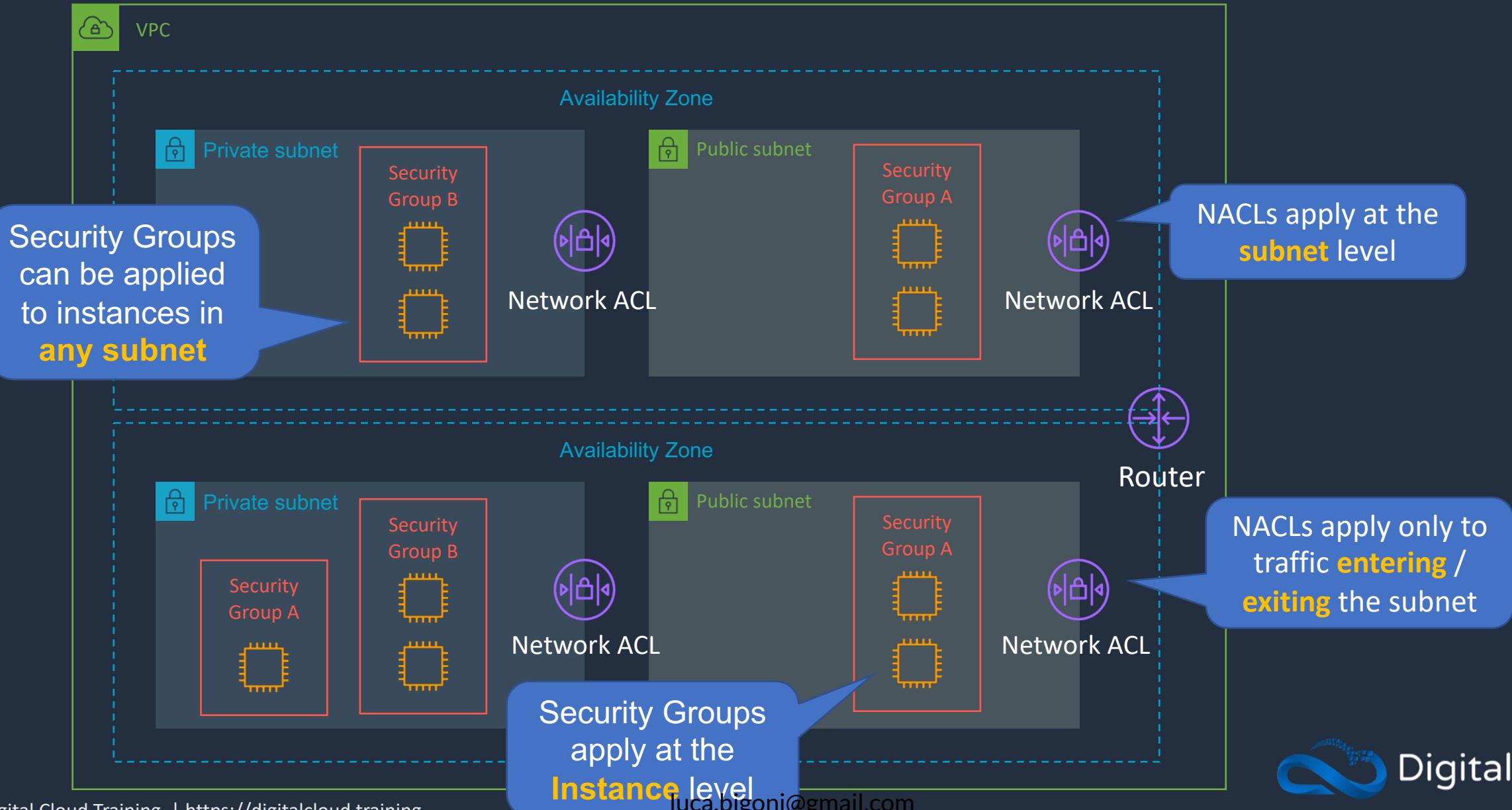
A **stateless** firewall checks for an allow rule for **both** connections

# Security Groups and Network ACLs





# Security Groups and Network ACLs





# Security Group Rules

Security groups support  
**allow** rules only

## Inbound rules

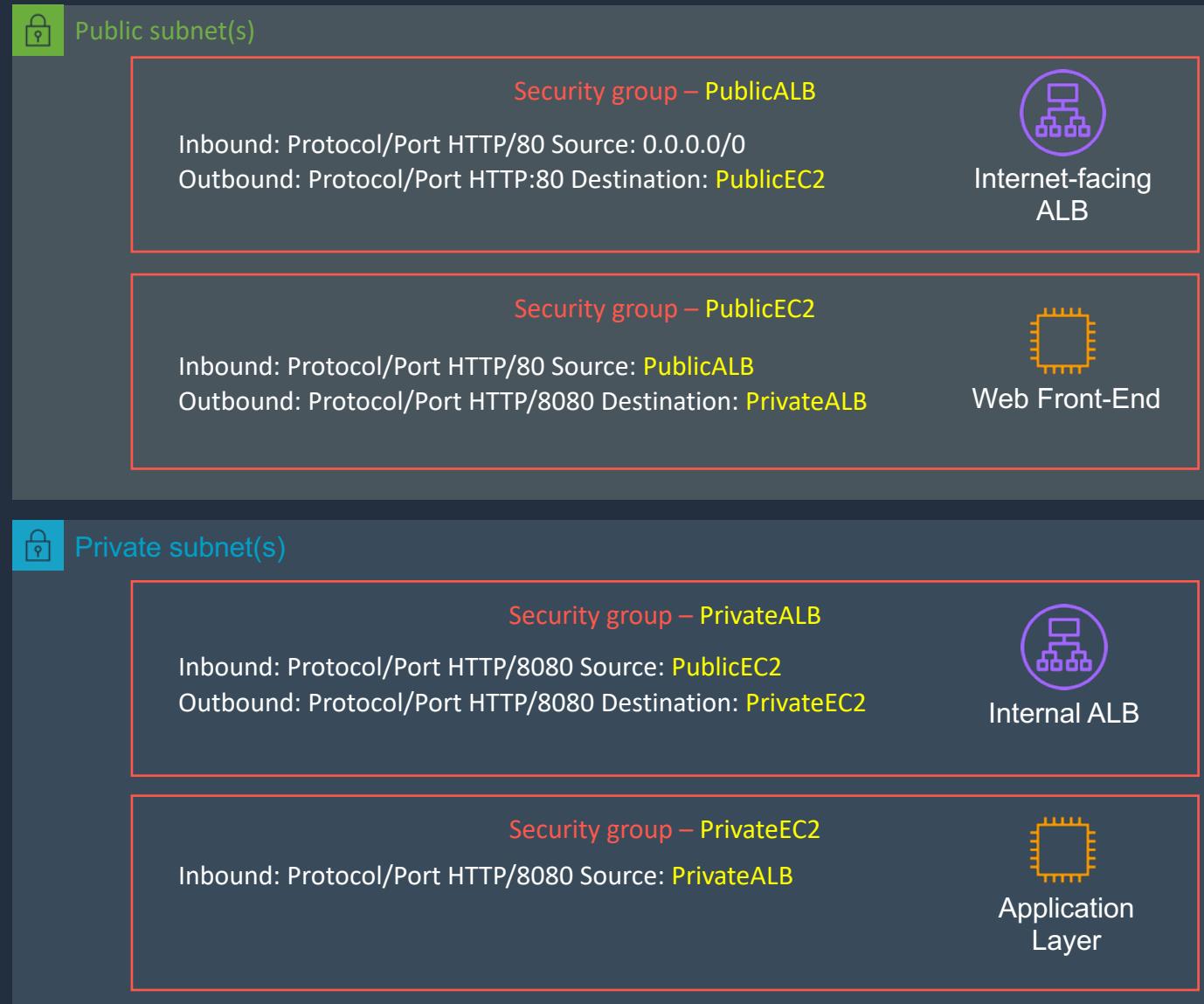
Type	Protocol	Port range	Source
SSH	TCP	22	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0
RDP	TCP	3389	::/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0
All ICMP - IPv4	ICMP	All	0.0.0.0/0

Separate rules  
are defined for  
**outbound** traffic

A source can be an **IP  
address or security  
group ID**



# Security Groups Best Practice





# Network ACLs

## Inbound Rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

## Outbound Rules

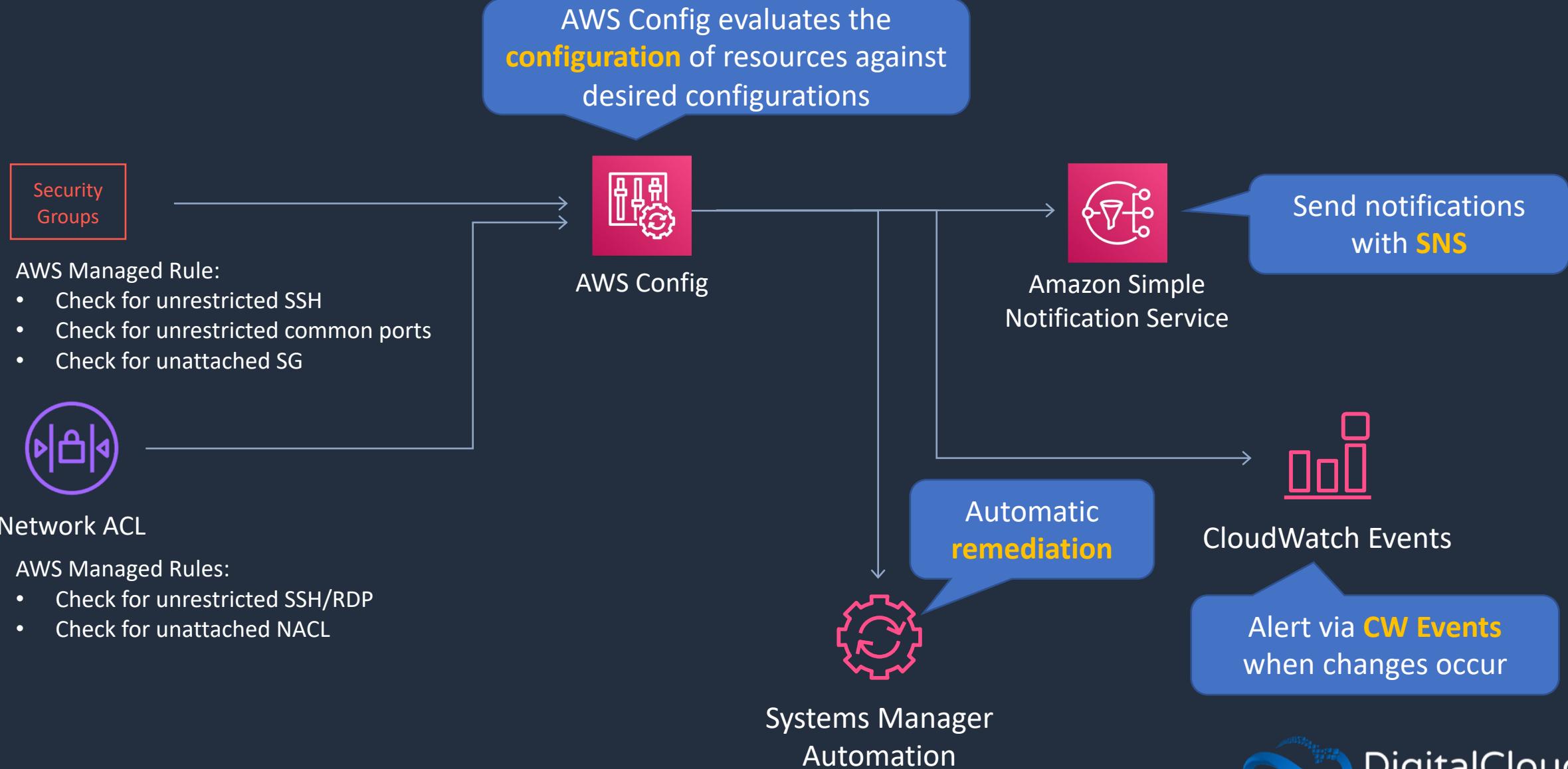
Rule #	Type	Protocol	Port Range	Destination	
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

Rules are processed  
in order

NACLs have an  
explicit deny

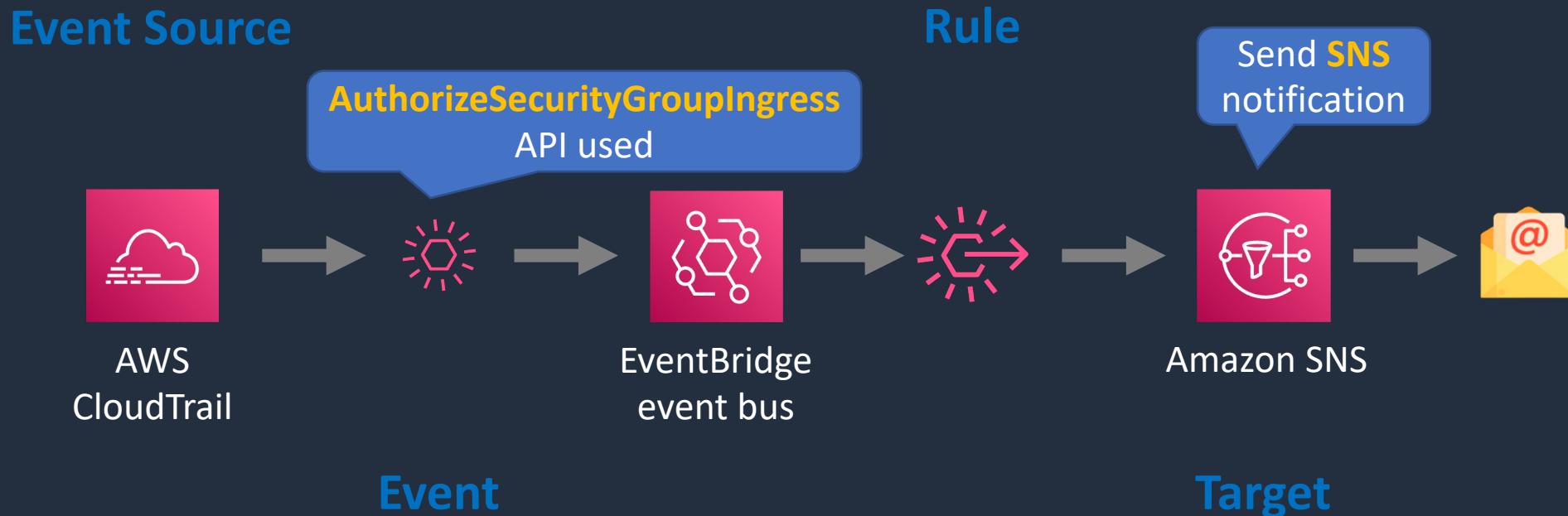


# Configuration Management





# Audit Security Group Changes



```
{  
  "source": ["aws.ec2"],  
  "detail-type": ["AWS API Call via CloudTrail"],  
  "detail": {  
    "eventSource": ["ec2.amazonaws.com"],  
    "eventName": ["AuthorizeSecurityGroupIngress", "AuthorizeSecurityGroupEgress",  
      "RevokeSecurityGroupIngress", "RevokeSecurityGroupEgress"]  
  }  
}
```

# Configure Security Groups and NACLs

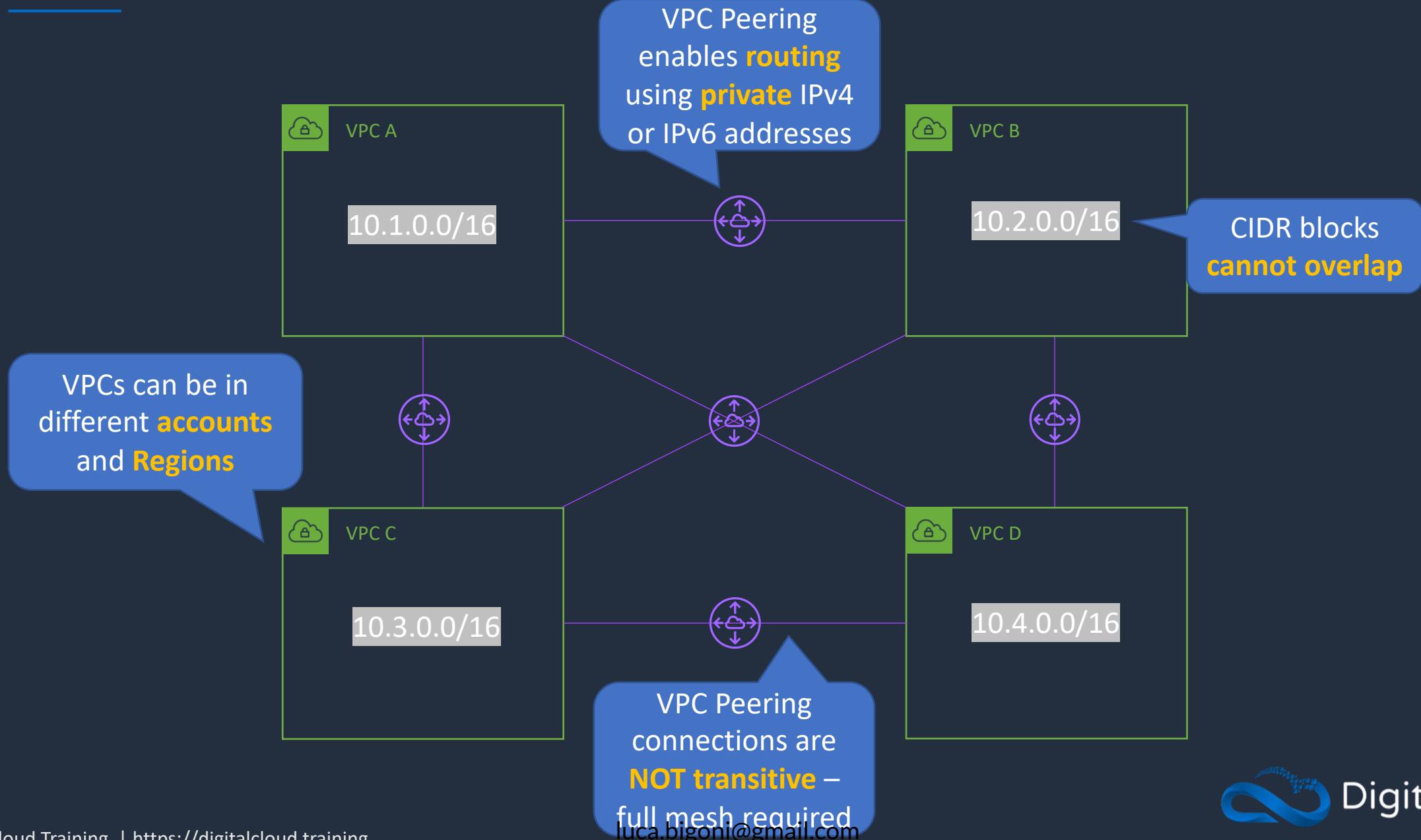


# VPC Peering





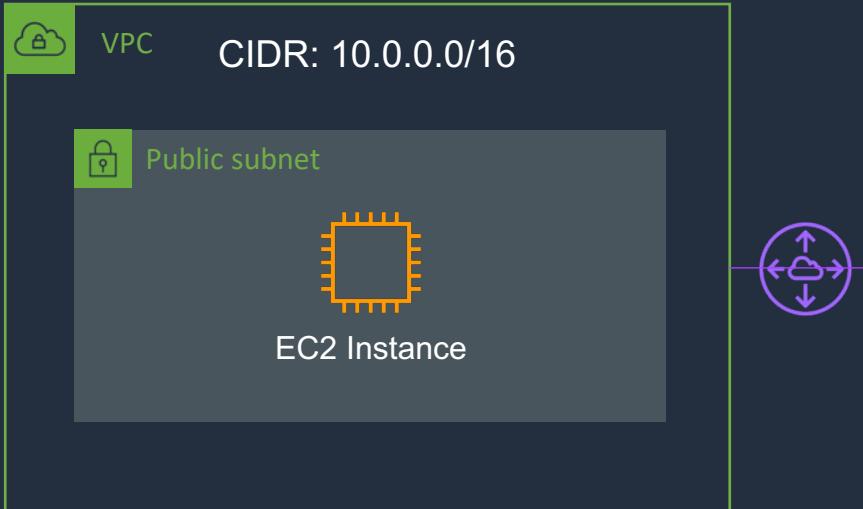
# VPC Peering



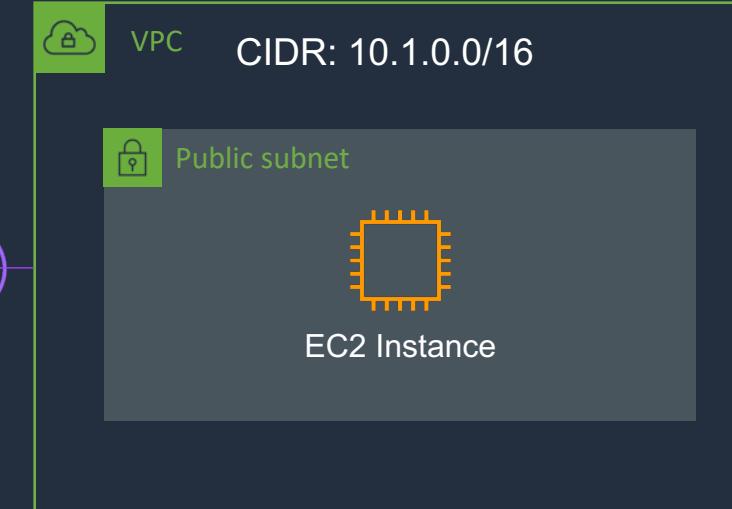


# VPC Peering

Management Account



Production Account



Security group (VPCPEER-MGMT)

Protocol	Port	Source
ICMP	All	10.1.0.0/16
TCP	22	0.0.0.0/0

Security group (VPCPEER-PROD)

Protocol	Port	Source
ICMP	All	10.0.0.0/16
TCP	22	0.0.0.0/0

Route Table

Destination	Target
10.1.0.0/16	peering-id

Route Table

Destination	Target
10.0.0.0/16	peering-id

# Configure VPC Peering

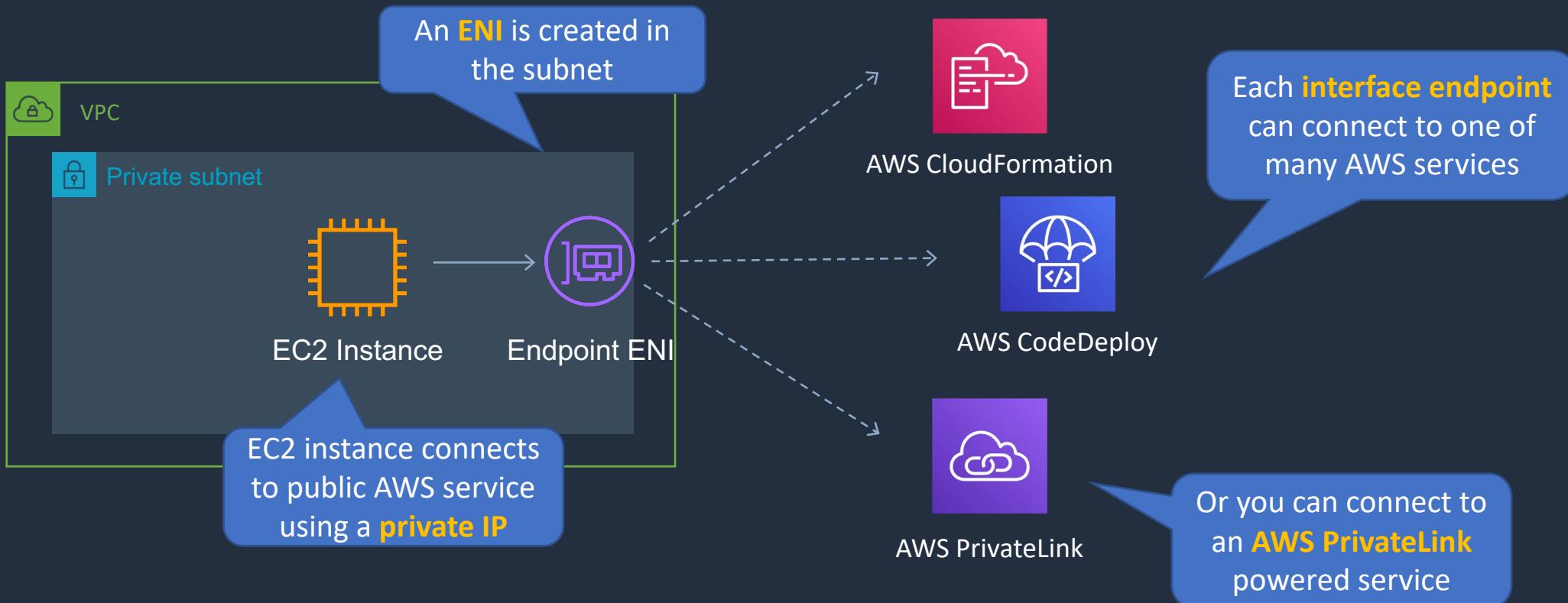


# VPC Endpoints



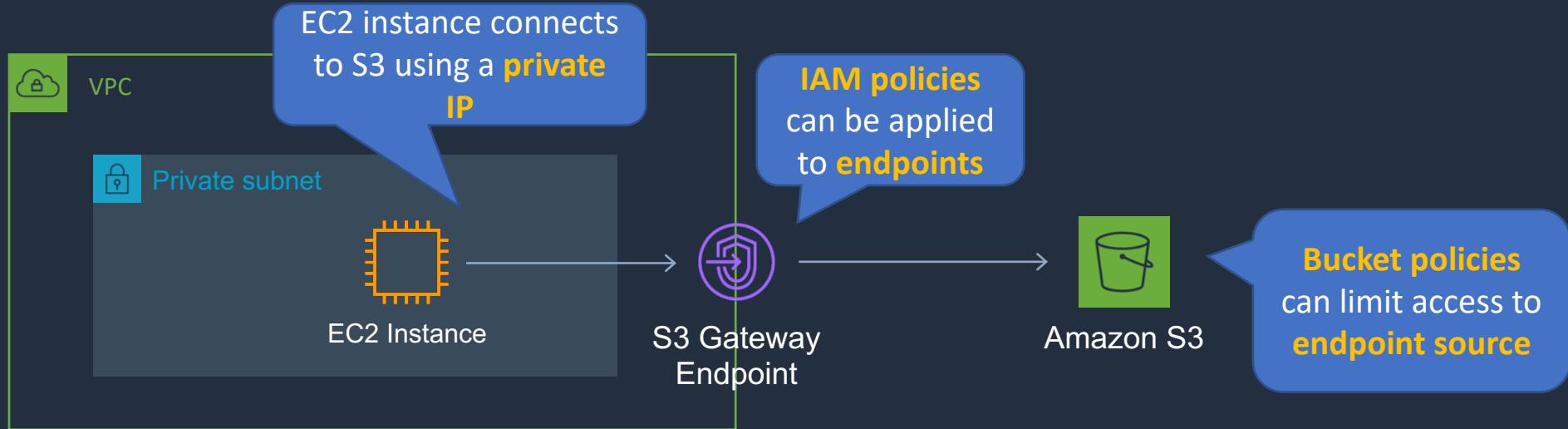


# VPC Interface Endpoints





# VPC Gateway Endpoints



Route Table

Destination	Target
<code>pl-6ca54005 (com.amazonaws.ap-southeast-2.s3, 54.231.248.0/22, 54.231.252.0/24, 52.95.128.0/21)</code>	<code>vpce-ID</code>

A **route table** entry is required with the prefix list for S3 and the **gateway ID**



# VPC Endpoints

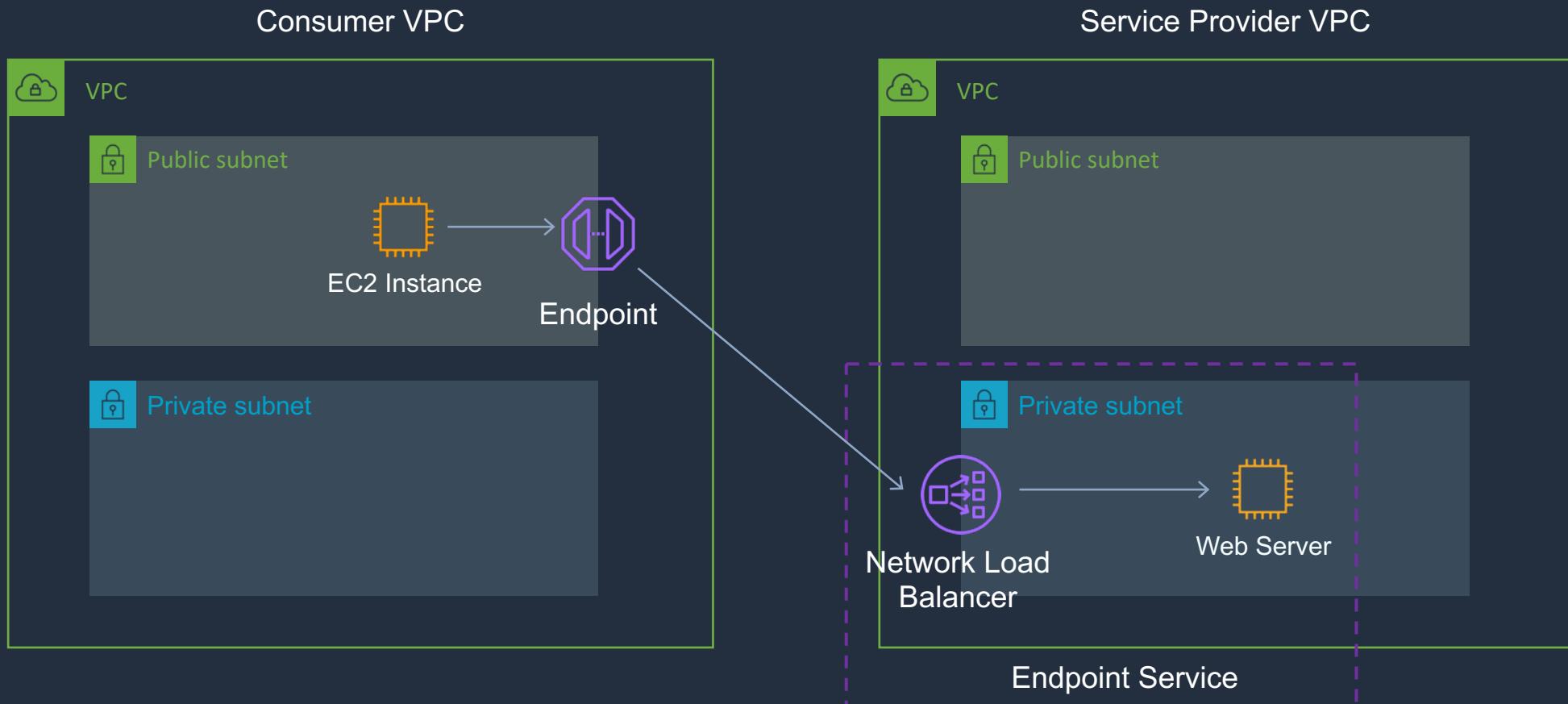
---

	Interface Endpoint	Gateway Endpoint
What	Elastic Network Interface with a Private IP	A gateway that is a target for a specific route
How	Uses DNS entries to redirect traffic	Uses prefix lists in the route table to redirect traffic
Which services	API Gateway, CloudFormation, CloudWatch etc.	Amazon S3, DynamoDB
Security	Security Groups	VPC Endpoint Policies



# Service Provider Model

---





# Some AWS Services that Support PrivateLink

---

---

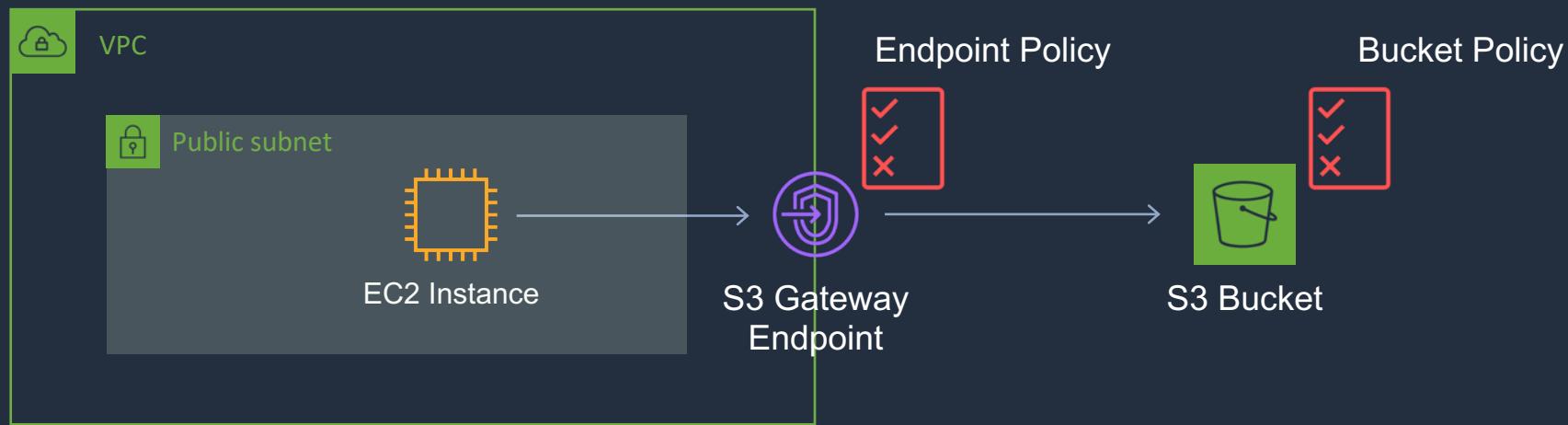
- [Amazon API Gateway](#)
- [Amazon Athena](#)
- [AWS Key Management Service](#)
- [AWS CloudHSM](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch Events](#)
- [Amazon CloudWatch Logs](#)
- [AWS Config](#)
- [Amazon EventBridge](#)
- [AWS Glue](#)
- [AWS Lambda](#)
- [Amazon S3](#)
- [AWS Secrets Manager](#)
- [Amazon SNS](#)
- [Amazon SQS](#)
- [AWS Step Functions](#)
- [AWS Systems Manager](#)
- [Full list here](#)

# Create VPC Endpoint





# VPC Gateway Endpoints



Route Table

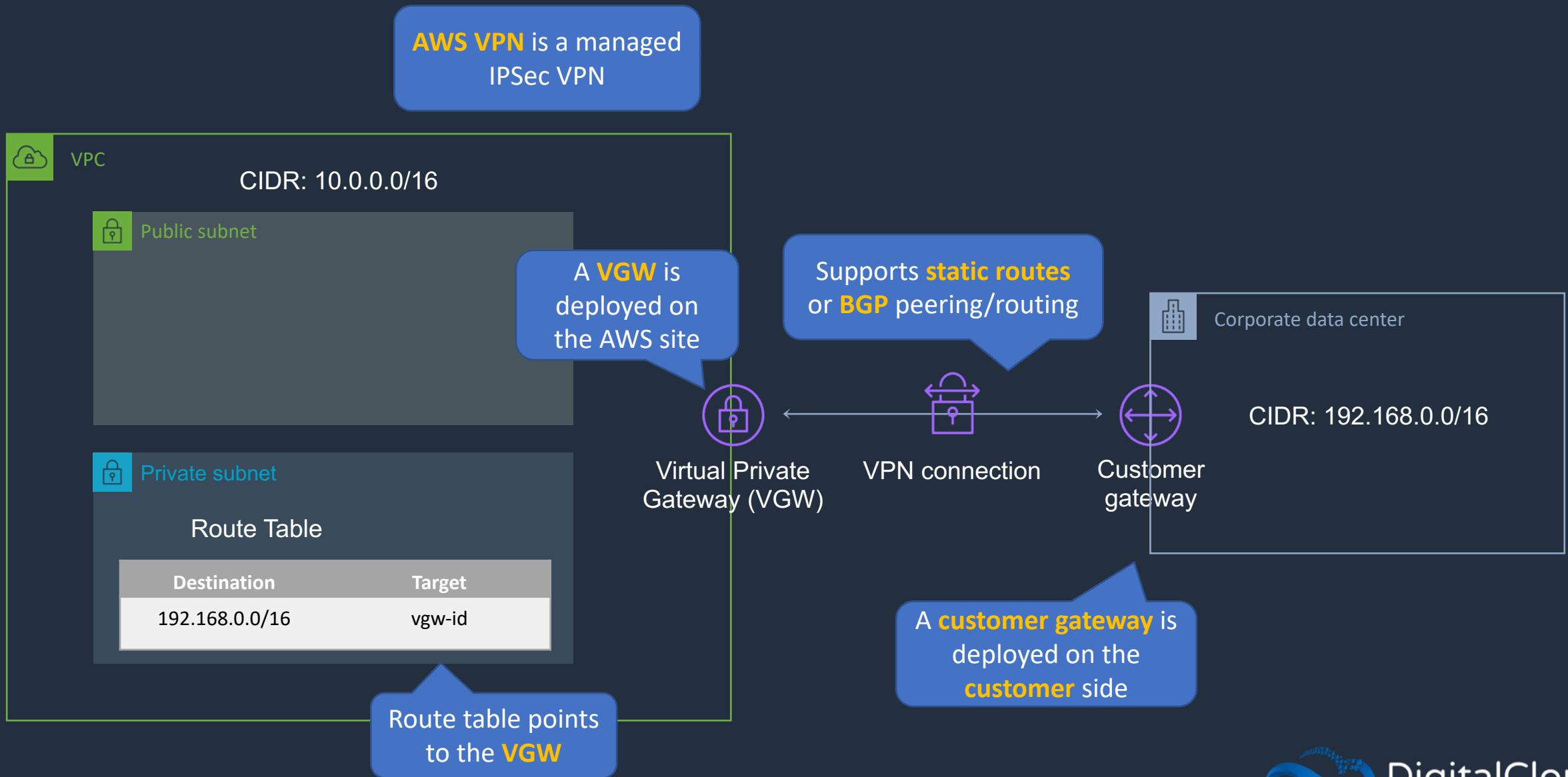
Destination	Target
<code>pl-6ca54005 (com.amazonaws.ap-southeast-2.s3, 54.231.248.0/22, 54.231.252.0/24, 52.95.128.0/21)</code>	<code>vpce-ID</code>

# AWS Site-to-Site VPN





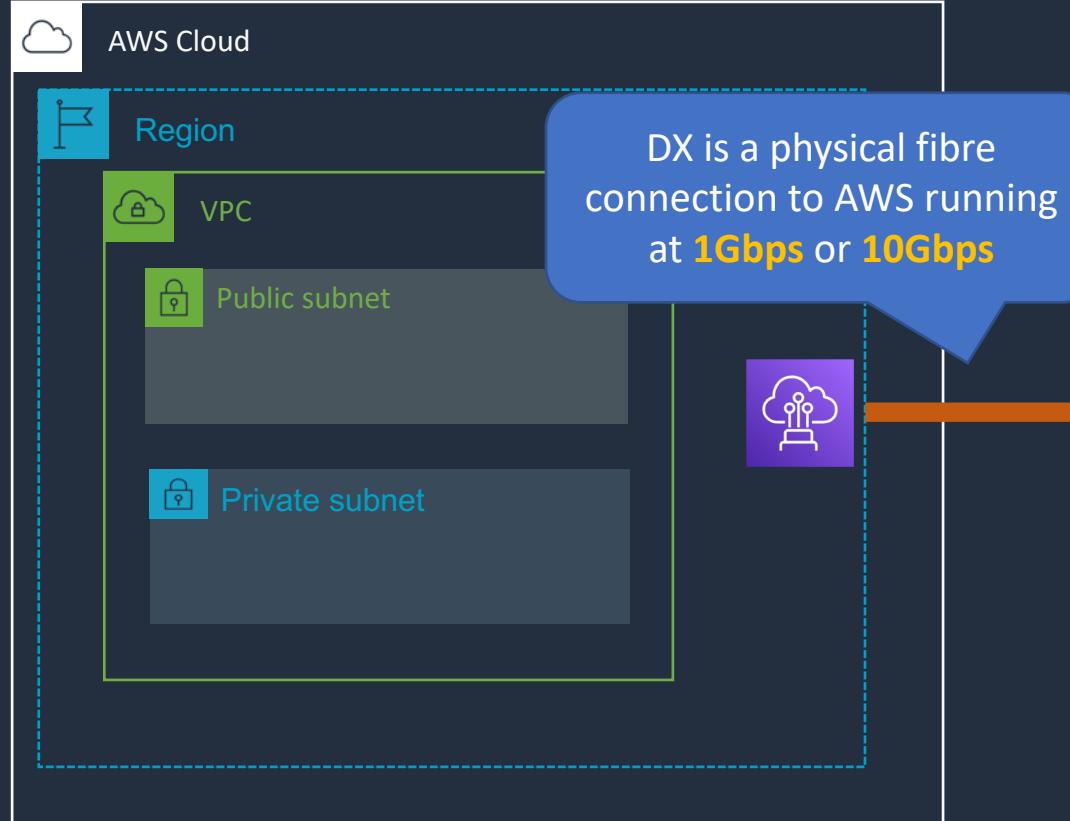
# AWS Site-to-Site VPN



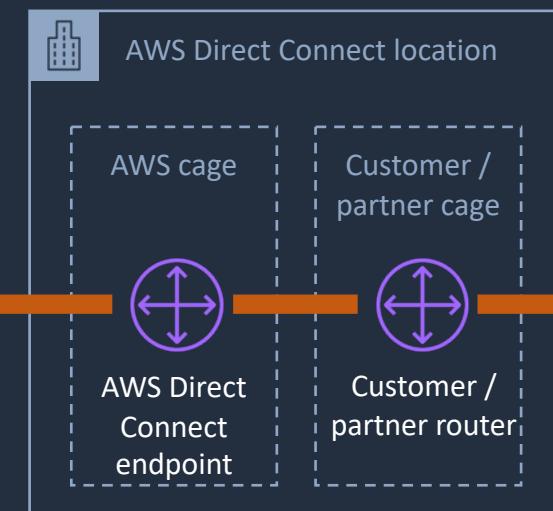
# Securing AWS Direct Connect



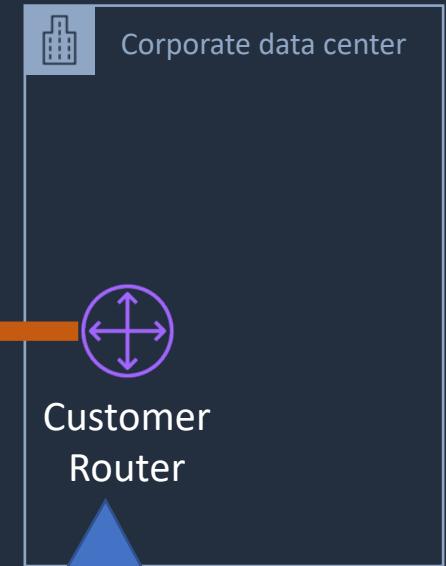
# AWS Direct Connect (DX)



A **cross-connect** between the AWS DX router and the customer/partner DX router



A **DX port** must be allocated in a **DX location**



The **customer router** is connected to the DX router in the DX location



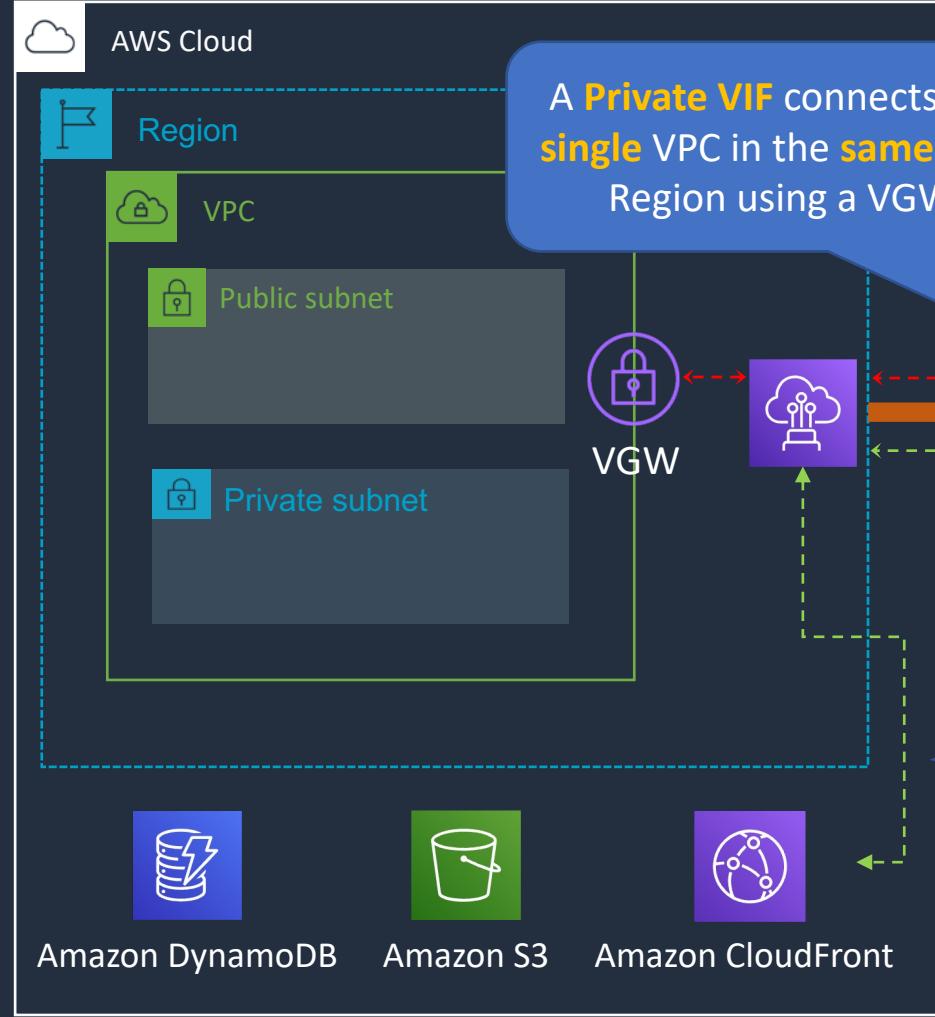
# AWS Direct Connect Benefits

---

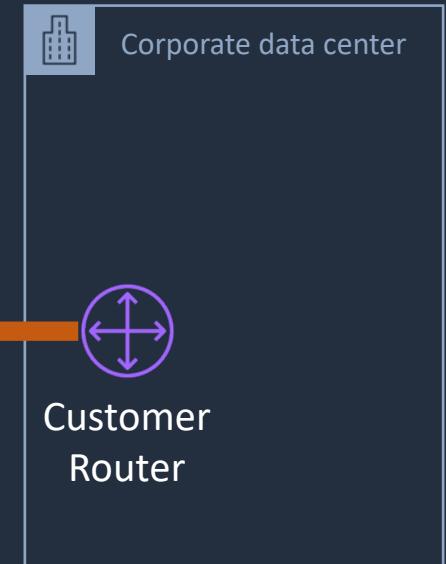
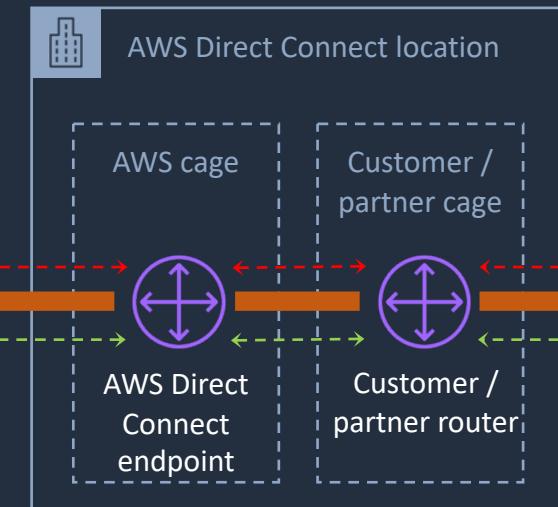
---

- **Private** connectivity between AWS and your data center / office
- Consistent network experience – increased **speed/latency** & **bandwidth/throughput**
- Lower costs for organizations that transfer **large** volumes of data

# AWS Direct Connect (DX)



A **VIF** is a virtual interface (802.1Q VLAN) and a **BGP** session





# Connectivity Options

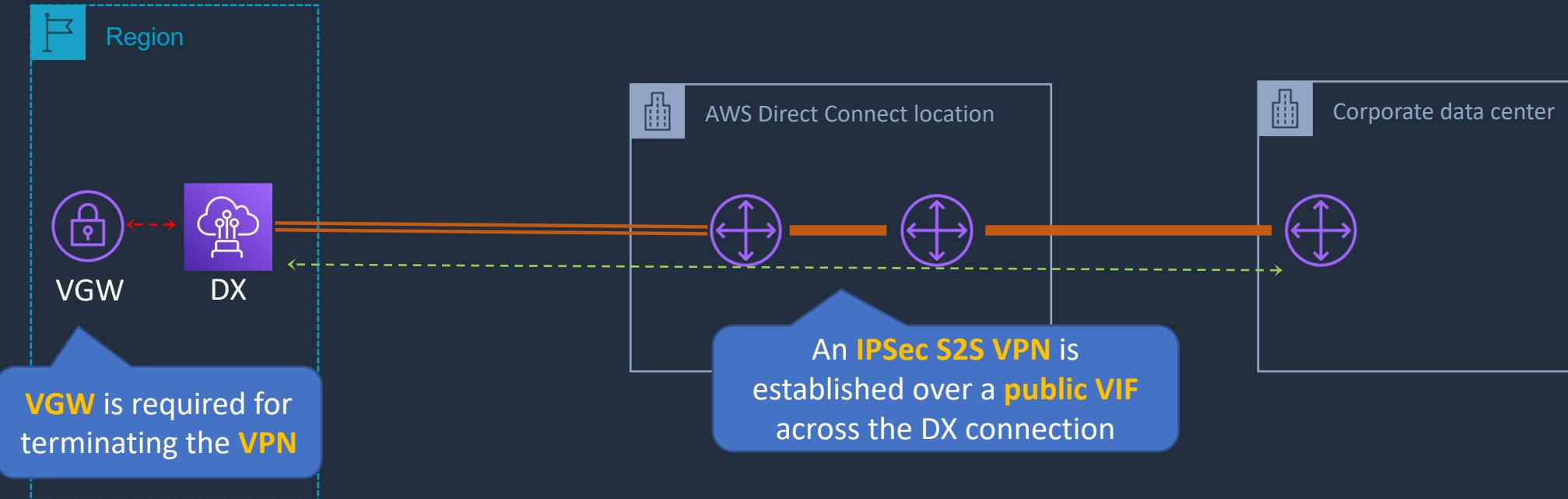
---

- Use public VIFs for accessing public AWS services
- Use private VIFs to connect to VPC resources in the same Region
- A Direct Connect Gateway can be used with a private VIF to connect multiple Regions
- A Transit Gateway can be used for a fully-meshed architecture within a Region



# Encryption – DX + IPSec VPN

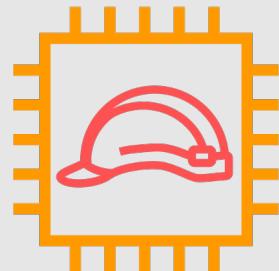
There's no native encryption  
for Direct Connect

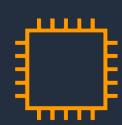


# VPC Flow Logs

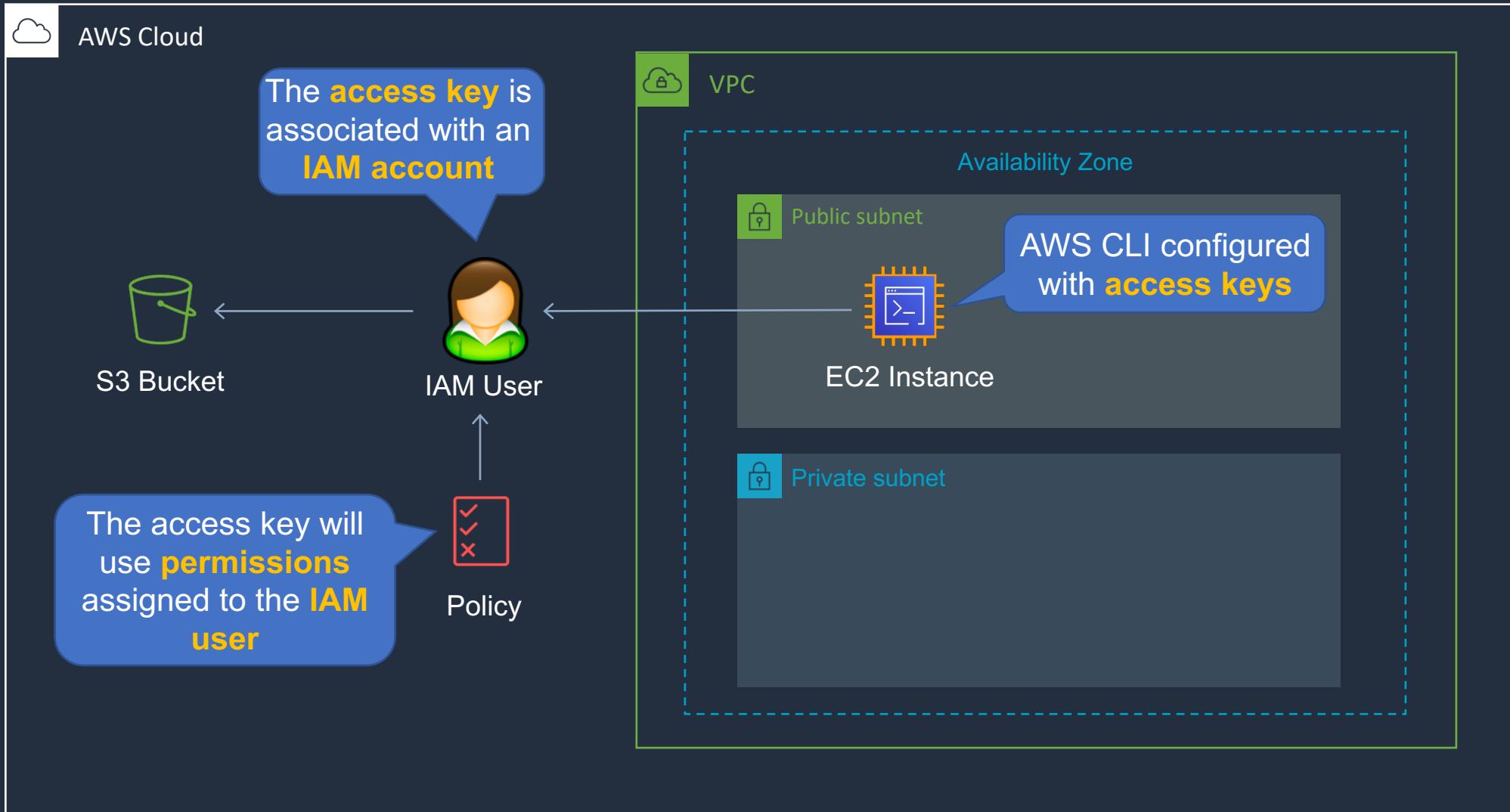


# Accessing Services – Access Keys and IAM Roles



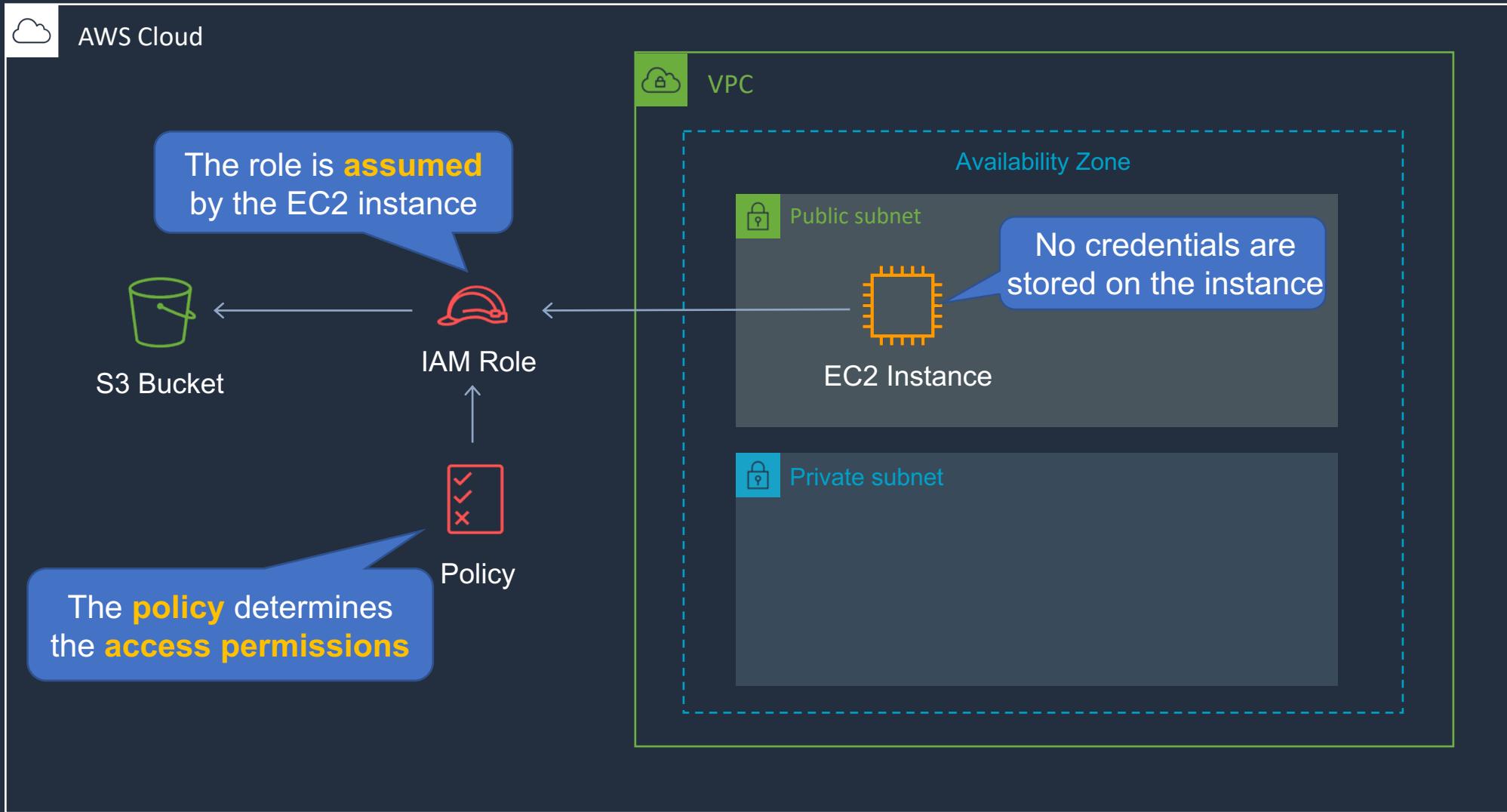


# Access Keys





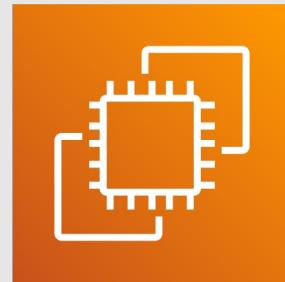
# Amazon EC2 Instance Profiles (IAM Roles for EC2)



# Access Keys and IAM Roles



# Managing Amazon EC2 Security



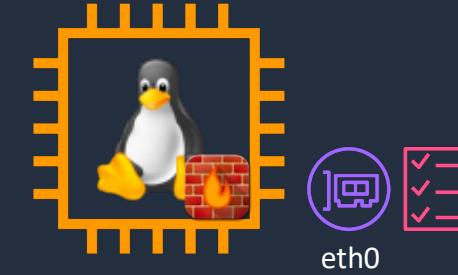


# Host-Based Firewalls

---

---

- Host-based firewall is configured within the operating system (OS)
- Can be used if complex rules exceed limits of SGs and NACLs
- Can be used with SGs and NACLs for defense in depth





# Amazon Inspector

---

---

- Runs assessments that check for security exposures and vulnerabilities in EC2 instances
- Can be configured to run on a schedule
- Agent must be installed on EC2 for host assessments
- Network assessments do not require an agent



## Network Assessments

- Assessments: Network configuration analysis to check for ports reachable from outside the VPC
- If the Inspector Agent is installed on your EC2 instances, the assessment also finds processes reachable on port
- Price based on the number of instance assessments



## Host Assessments

- Assessments: Vulnerable software (CVE), host hardening (CIS benchmarks), and security best practices
- Requires an agent (auto-install with SSM Run Command)
- Price based on the number of instance assessments



# AWS Systems Manager Patch Manager

- Helps you select and deploy operating system and software patches automatically across large groups of Amazon EC2 or on-premises instances
- Patch baselines:
  - Set rules to auto-approve select categories of patches to be installed
  - Specify a list of patches that override these rules and are automatically approved or rejected
- You can also schedule maintenance windows for your patches so that they are only applied during predefined times
- Systems Manager helps ensure that your software is up-to-date and meets your compliance policies



Patch Manager



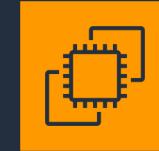
# AWS Systems Manager Session Manager

- Secure remote management of your instances at scale without logging into your servers
- Replaces the need for bastion hosts, SSH, or remote PowerShell
- Integrates with IAM for granular permissions
- All actions taken with Systems Manager are recorded by AWS CloudTrail
- Can store session logs in an S3 and output to CloudWatch Logs
- Requires IAM permissions for EC2 instance to access SSM, S3, and CloudWatch Logs

Doesn't require port 22,5985/5986



No need for bastion hosts



Amazon EC2  
(Linux)



Amazon EC2  
(Windows)

# EC2 Host-Based Security

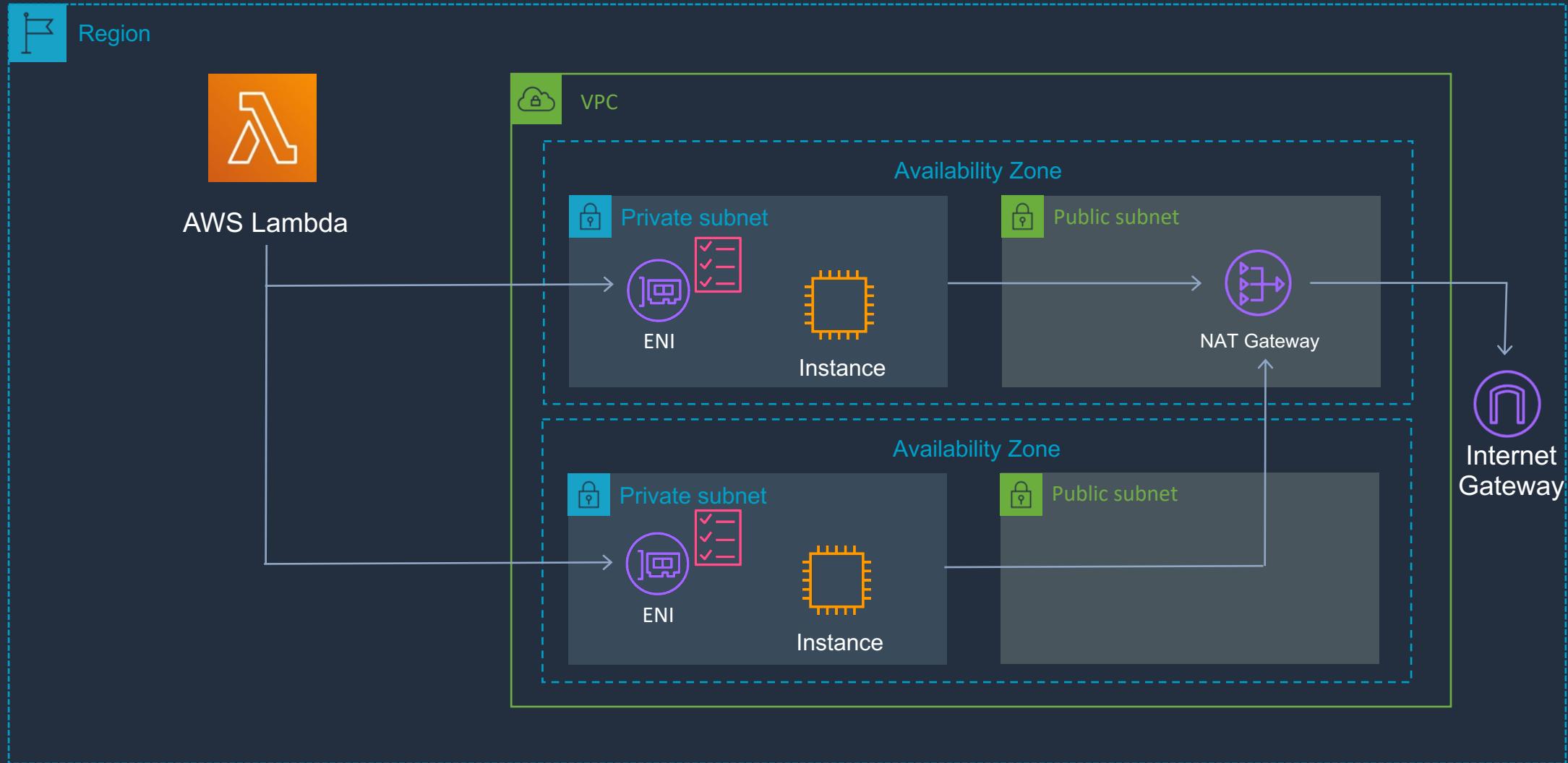


# AWS Services in Amazon VPC





# AWS Lambda in a VPC





# Amazon OpenSearch Service

---

---

- Distributed search and analytics suite
- Based on the popular open source Elasticsearch
- Clusters are created (Management Console, API, or CLI)
- Clusters are also known as OpenSearch Service domains
- You specify the number of instances and instance types



# OpenSearch in an Amazon VPC

---

- Clusters can be deployed in a VPC for secure intra-VPC communications
- VPN or proxy required to connect from the internet (public domains are directly accessible)



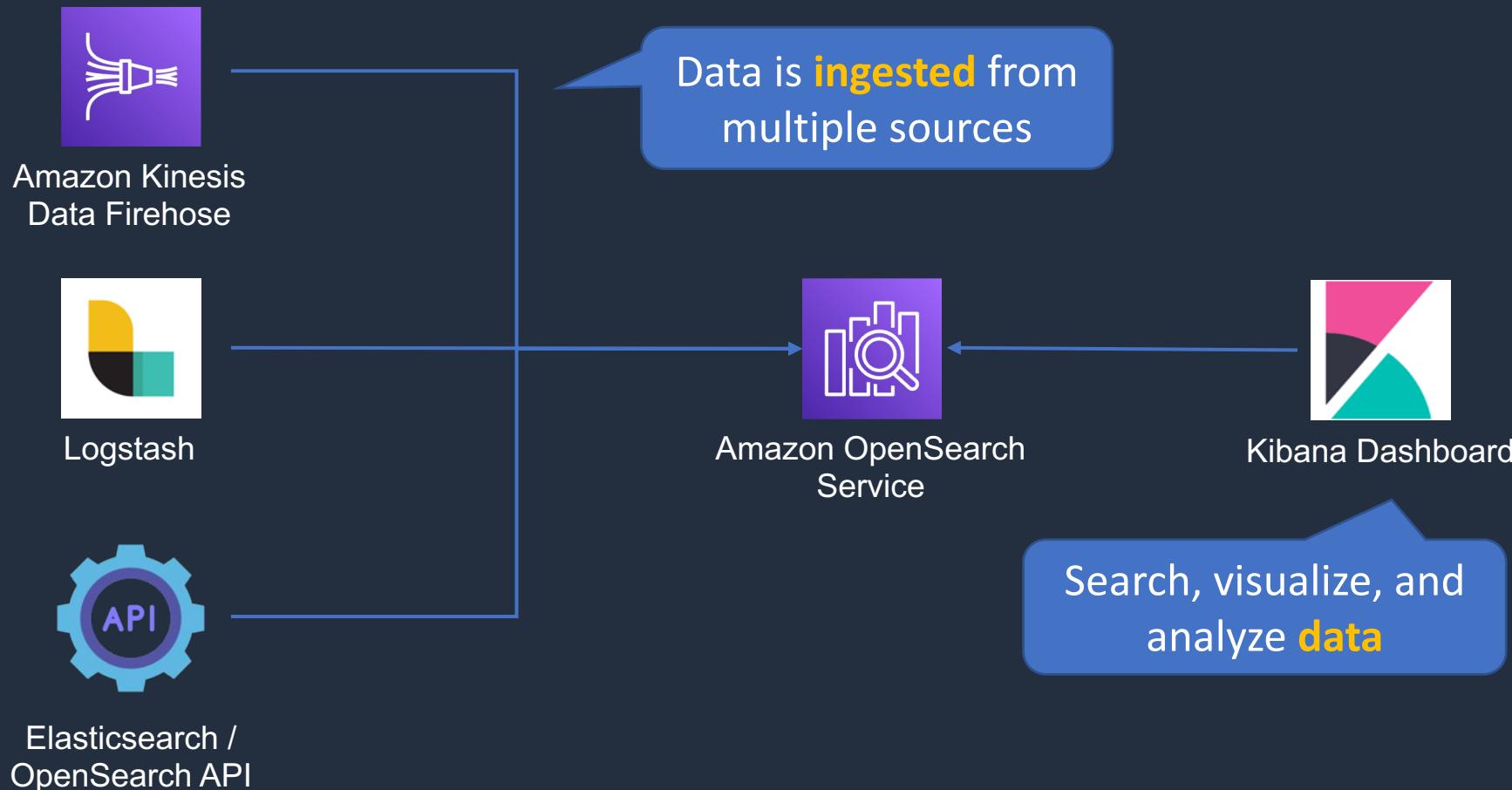
# OpenSearch in an Amazon VPC

---

- Limitations of VPC deployments:
  - You can't switch from VPC to a public endpoint. The reverse is also true
  - You can't launch your domain within a VPC that uses dedicated tenancy
  - After you place a domain within a VPC, you can't move it to a different VPC, but you can change the subnets and security group settings
  - Cannot use IP-based access policies



# Ingesting Data into OpenSearch Service Domains

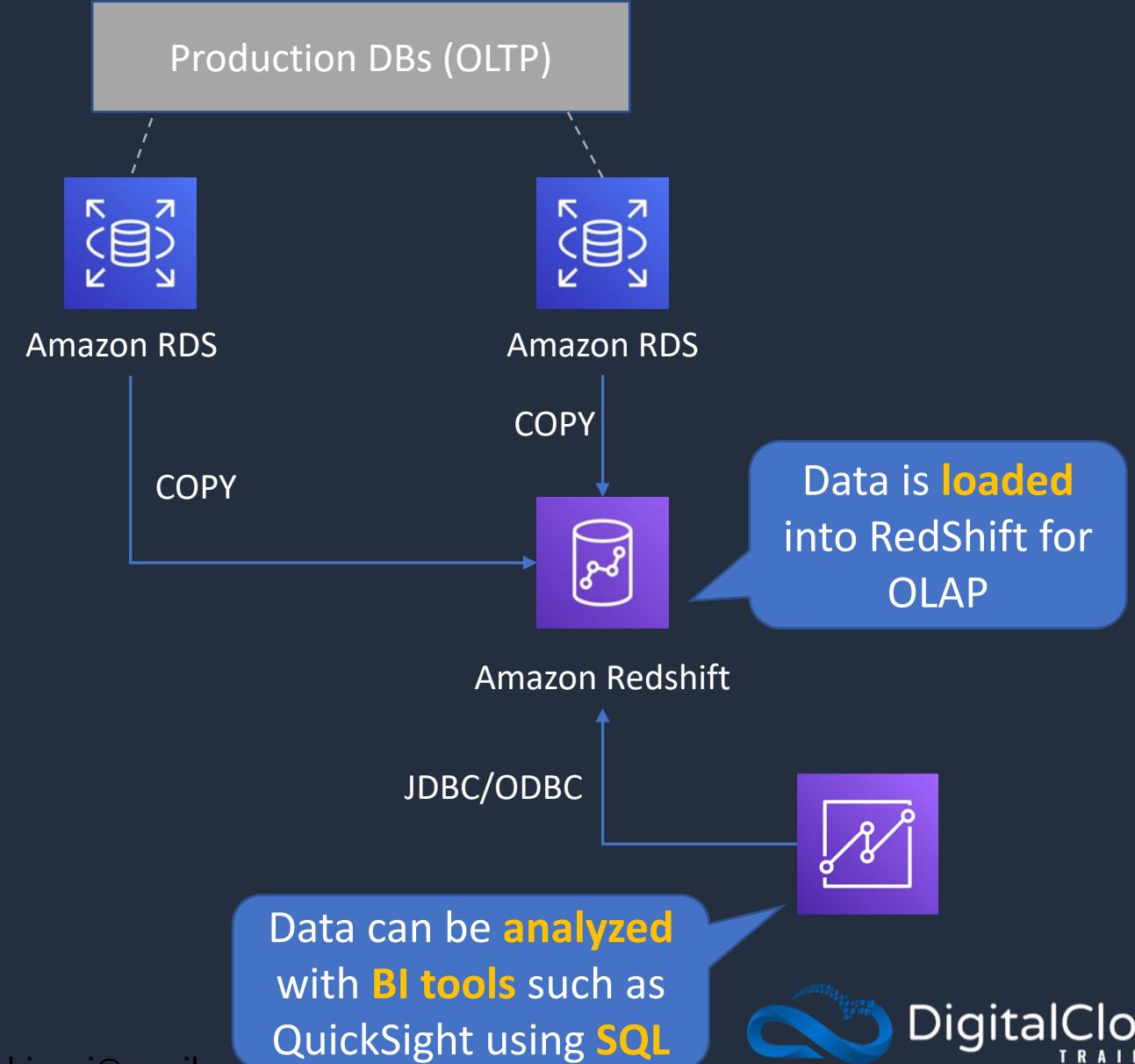




# Amazon RedShift

---

- Amazon Redshift is a fast, fully managed data warehouse
- Analyze data using standard SQL and Business Intelligence (BI) tools
- RedShift is an online analytics processing (OLAP) type of DB





# Deploying RedShift in a VPC

---

- Must create a cluster subnet group and provide VPC ID and a list of subnets in your VPC
- For publicly available clusters, you can specify an Elastic IP address to use
- Need to enable **DNS resolution** and **DNS hostnames** VPC settings to connect to a publicly available cluster using a private IP
- Use VPC security groups to control access to the database port

# Automating Infrastructure as Code





# AWS CloudFormation

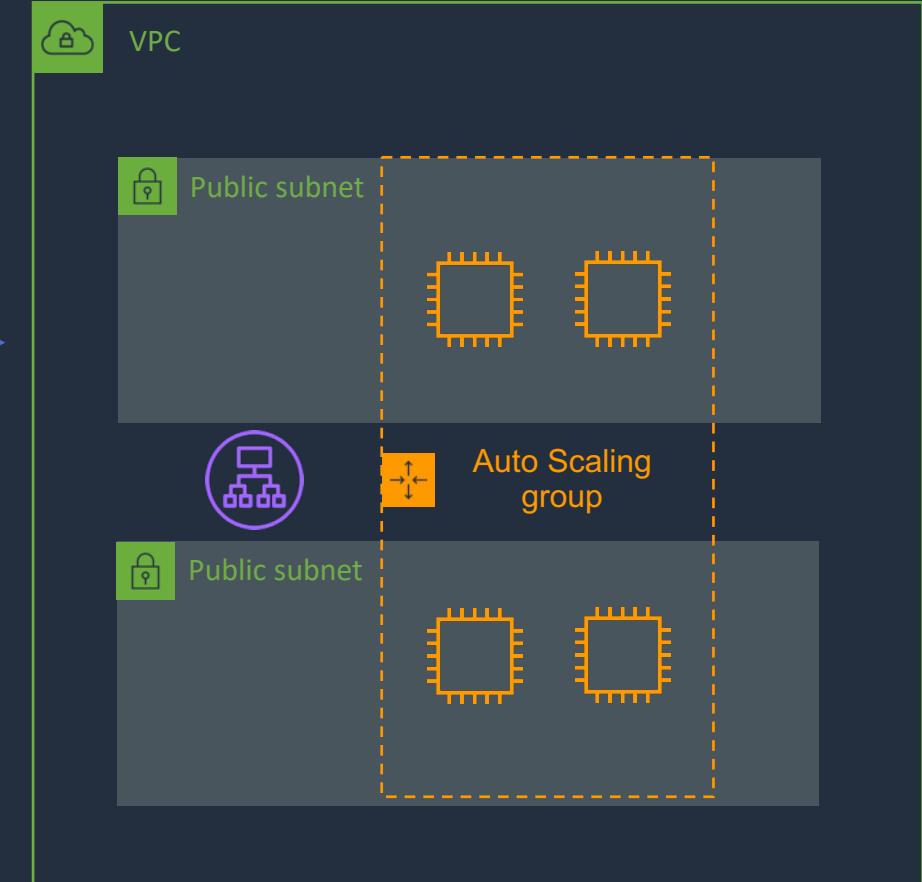
Infrastructure patterns are defined in a **template** file using **code**



CloudFormation **builds** your infrastructure according to the **template**

AWS CloudFormation

```
1 "AWSTemplateFormatVersion" : "2010-09-09",
2
3 "Description" : "AWS CloudFormation Sample Template WordPress_Multi_AZ: WordPress is web
4
5 "Parameters" : {
6   "VpcId" : {
7     "Type" : "AWS::EC2::VPC::Id",
8     "Description" : "VpcId of your existing Virtual Private Cloud (VPC)",
9     "ConstraintDescription" : "must be the VPC Id of an existing Virtual Private Cloud."
10 },
11
12 "Subnets" : {
13   "Type" : "List<AWS::EC2::Subnet::Id>",
14   "Description" : "The list of SubnetIds in your Virtual Private Cloud (VPC)",
15   "ConstraintDescription" : "must be a list of at least two existing subnets associated
16 },
```





# AWS CloudFormation - Benefits

- Infrastructure is provisioned consistently, with fewer mistakes (human error)
- Once templates are defined with secure settings, you can repeatedly deploy secure infrastructure
- Less time and effort than configuring resources manually
- You can use version control and peer review for your CloudFormation templates
- Can be used to rollback and delete the entire stack as well



# AWS CloudFormation

Component	Description
Templates	The JSON or YAML text file that contains the instructions for building out the AWS environment
Stacks	The entire environment described by the template and created, updated, and deleted as a single unit
StackSets	AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation
Change Sets	A summary of proposed changes to your stack that will allow you to see how those changes might impact your existing resources before implementing them



# AWS CloudFormation - Templates

---

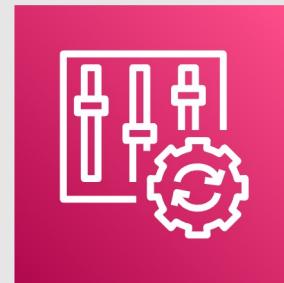
---

- A template is a YAML or JSON template used to describe the end-state of the infrastructure you are either provisioning or changing
- After creating the template, you upload it to CloudFormation directly or using Amazon S3
- CloudFormation reads the template and makes the API calls on your behalf
- The resulting resources are called a "Stack"

# Deploy AWS CloudFormation Stack



# AWS Config



luca.bigoni@gmail.com



# AWS Config

---

---

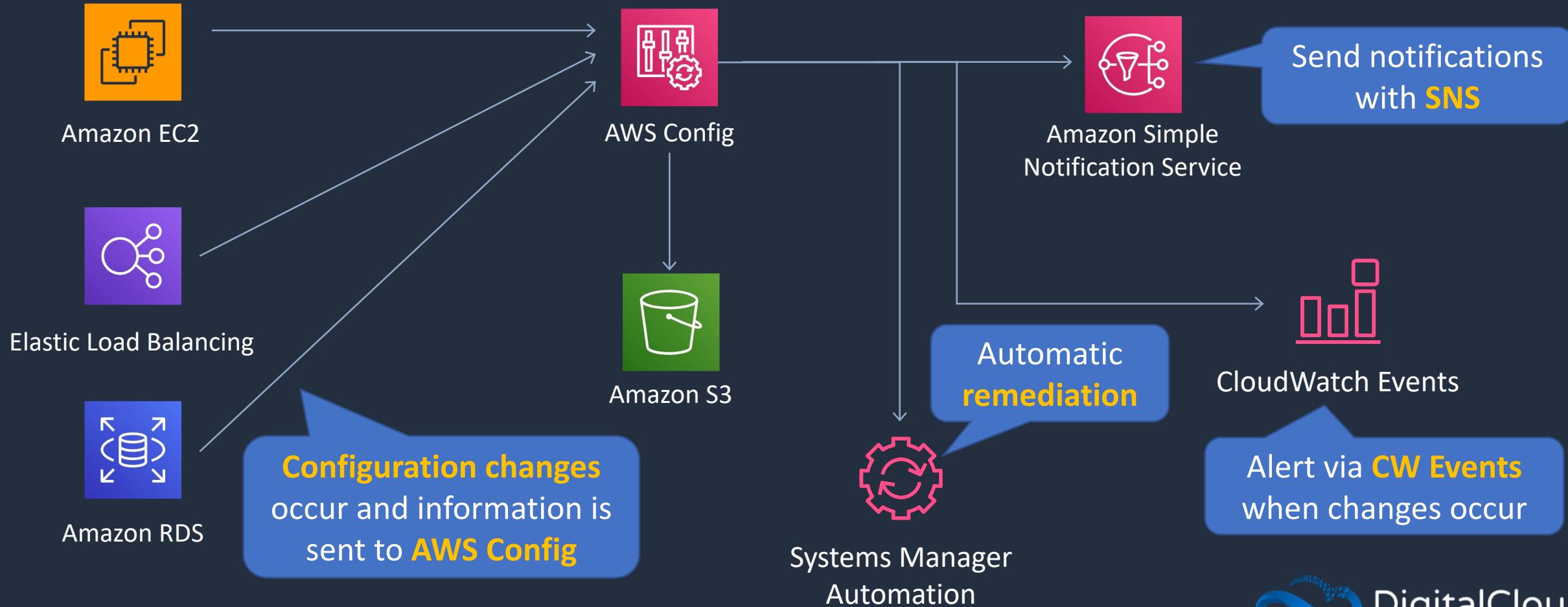
- Evaluate your AWS resource configurations for desired settings
- Get a snapshot of the current configurations of resources that are associated with your AWS account
- Retrieve configurations of resources that exist in your account
- Retrieve historical configurations of one or more resources
- Receive a notification whenever a resource is created, modified, or deleted
- View relationships between resources



# AWS Config

AWS Config evaluates the **configuration** against desired configurations

Example Services:





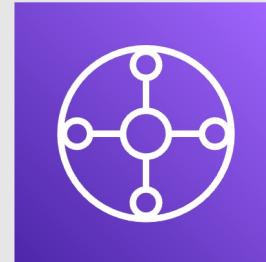
# AWS Config

Example Rule	Description
s3-bucket-server-side-encryption-enabled	Checks that your Amazon S3 bucket either has S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server side encryption
restricted-ssh	Checks whether security groups that are in use disallow unrestricted incoming SSH traffic
rds-instance-public-access-check	Checks whether the Amazon Relational Database Service (RDS) instances are not publicly accessible
cloudtrail-enabled	Checks whether AWS CloudTrail is enabled in your AWS account

# AWS Config Rule with Remediation



# AWS Transit Gateway



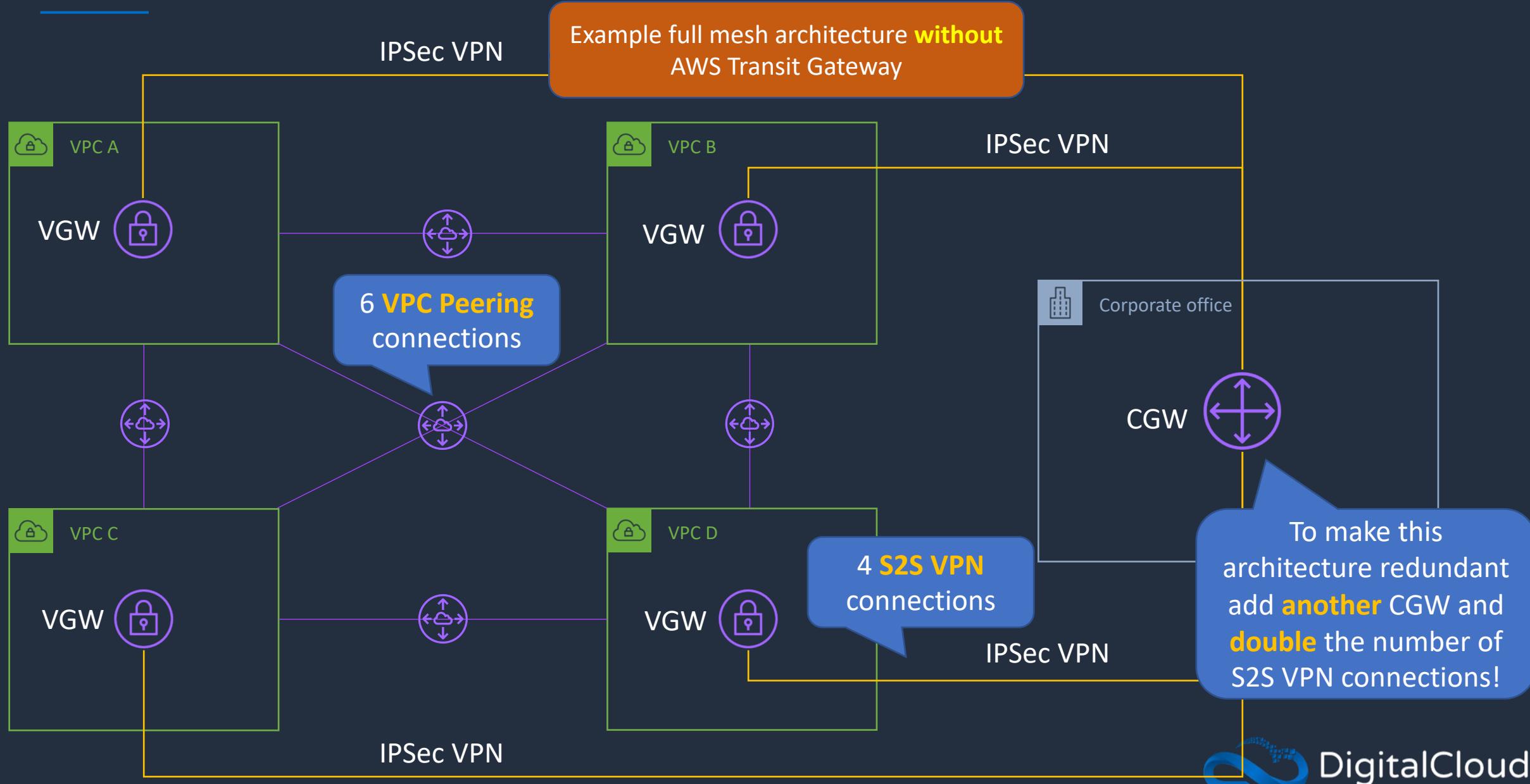


# AWS Transit Gateway

- Acts as a cloud router simplifying network architecture
- Supports dynamic and static layer 3 routing between Amazon VPCs and VPN
- Supports Equal Cost Multipath (ECMP) between multiple VPN connections
- Transit gateway is a Regional service
- Peering connections can be established between transit gateways in the same AWS region or across regions

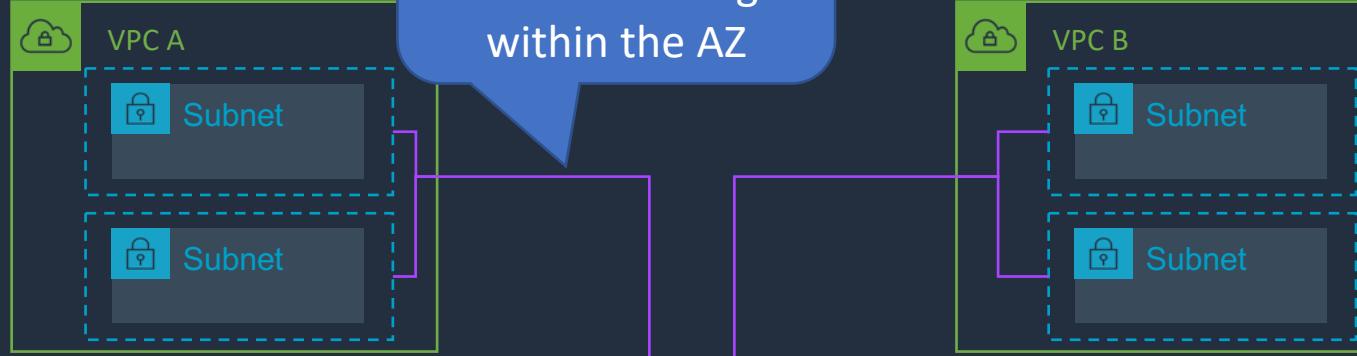


# AWS Transit Gateway



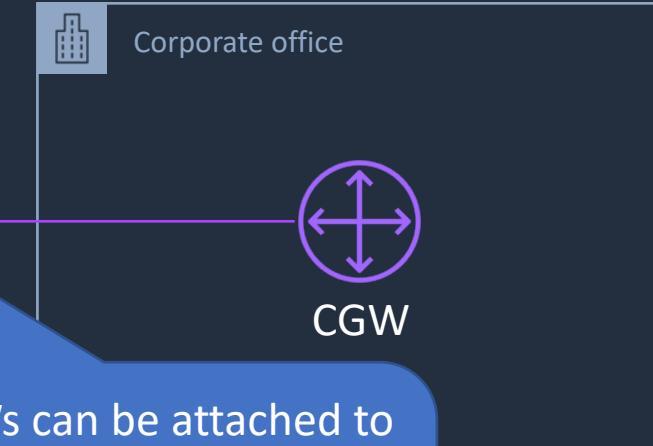
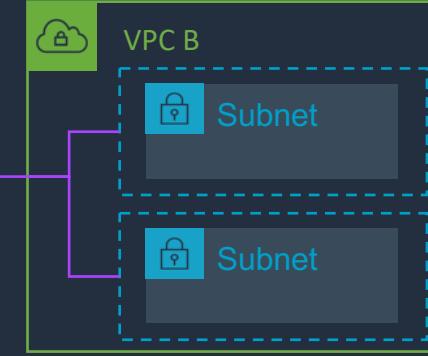
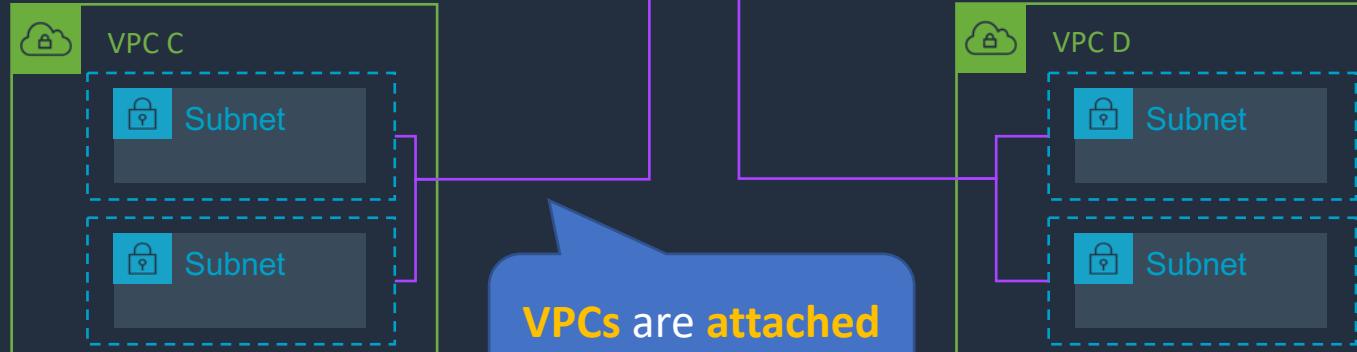


# AWS Transit Gateway



Example full mesh architecture **with** AWS Transit Gateway

**Transit Gateway** is a network transit hub that interconnects **VPCs** and **on-premises** networks

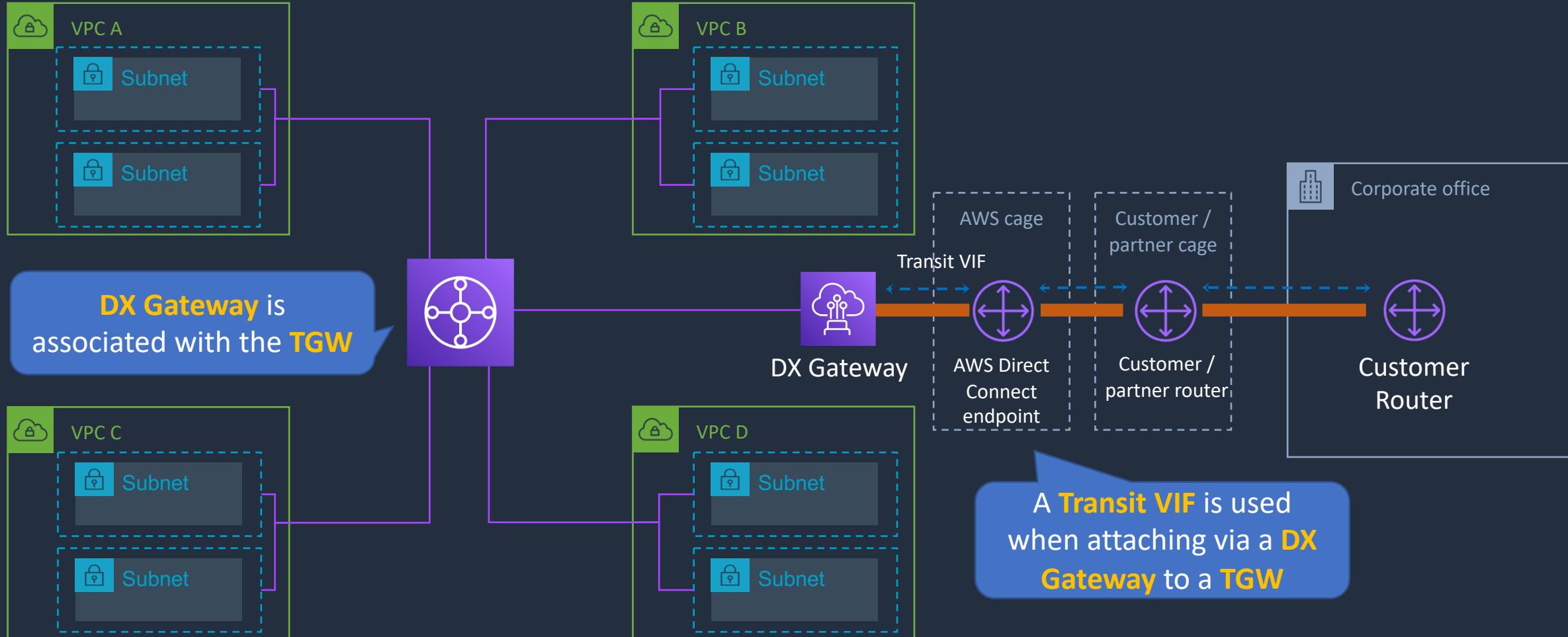


TGWs can be attached to **VPNs, Direct Connect Gateways, 3<sup>rd</sup> party appliances** and **TGWs** in other Regions/accounts



# AWS TGW + DX Gateway

This architecture supports **full transitive** routing between **on-premises**, **TGW** and **VPCs**





# Transit Gateway Connect

---

- Transit gateway connect can be used to connect to SD-WAN appliances running in a VPC
- Connect attachments support Generic Routing Encapsulation (GRE) and BGP
- Can establish one or more GRE tunnels
- You establish two BGP sessions over the GRE tunnel to exchange routing information



# Multicast Routing

---

- Transit gateway supports multicast routing between subnets of attached VPCs
- TGW routes traffic for instances sending to multiple receiving instances
- Multicast domain membership is defined at the subnet level
- Multicast groups identify hosts that send and receive multicast traffic by groups of IP addresses
- Multicast group membership is defined by individual ENI
- Internet Group Management Protocol (IGMP) is used for managing multicast group membership
- Multicast routing is not supported over AWS Direct Connect, Site-to-Site VPN, or peering attachments



# Transit Gateway Monitoring

---

- You can monitor with Amazon CloudWatch:
  - Transit gateway metrics ([AWS/TransitGateway](#))
  - Attachment-level metrics
- Amazon VPC Flow Logs to capture information on the IP traffic routed through the TGW
- AWS Transit Gateway Network Manager enables global monitoring for AWS and on-premises
  - **Centralized Network Monitoring** – including alerting on changes to topology, routing, and connection status
  - **Global Network Visibility** – visualize your entire global network
  - **SD-WAN Integration** – seamless integration with SD-WAN solutions from Cisco, Aruba, Silver Peak, Aviatrix, and Versa Networks

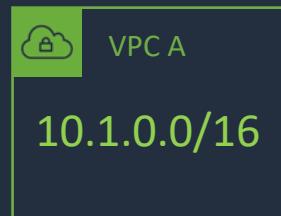


# Isolated VPCs and Shared Services Architecture

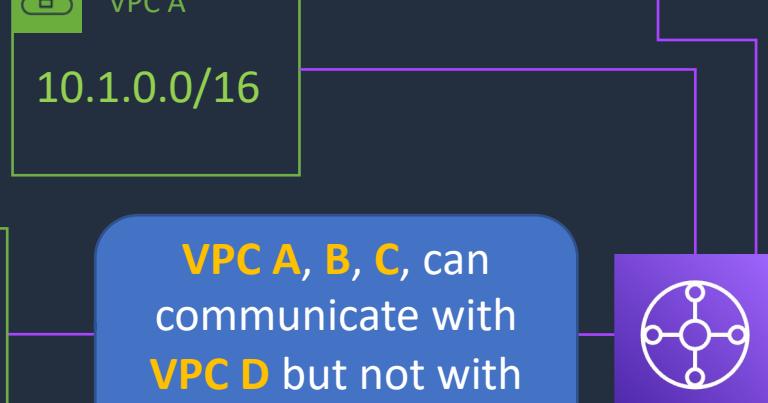
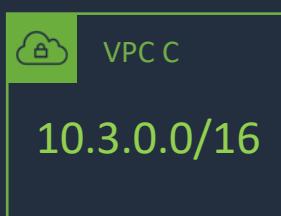
VPC A Route Table

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	tgw-id

The route tables for **VPC A, B, C, and D** will have similar routes



**VPC A, B, C,** can communicate with **VPC D** but not with each other



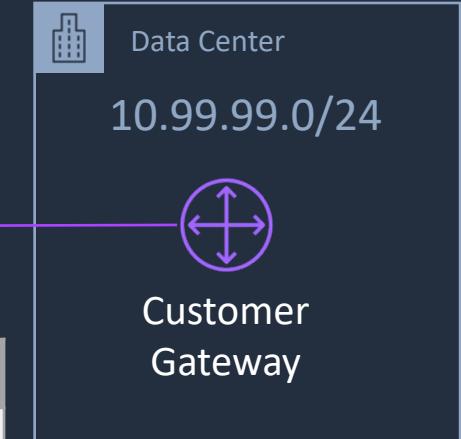
**Transit GW** functions as multiple isolated routers

TGW RT1 – Associated with VPC A, B, and C

Destination	Target	Route type
10.99.99.0/24	VPN attachment	propagated
10.4.0.0/16	VPC D attachment	propagated

TGW RT2 - Associated with VPN and VPC D

Destination	Target	Route type
10.1.0.0/16	VPC A attachment	propagated
10.2.0.0/16	VPC B attachment	propagated
10.3.0.0/16	VPC C attachment	propagated
10.4.0.0/16	VPC D attachment	propagated





# Isolated VPCs and Shared Services Architecture

VPC A Route Table

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	tgw-id



VPC A, B, C, can communicate with VPC D but not with each other

Attachments associated with **one isolated router** can route packets to **each other**, but **cannot** route packets to or receive packets from the attachments for **another isolated router**



TGW RT1 – Associated with VPC A, B, and C

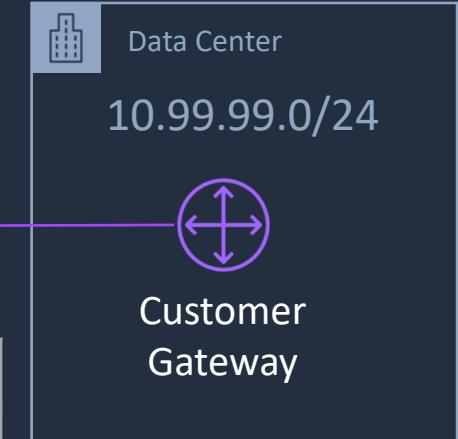
Destination	Target	Route type
10.99.99.0/24	VPN attachment	propagated
10.4.0.0/16	VPC D attachment	propagated

Transit GW functions as multiple isolated routers

S2S VPN

TGW RT2 - Associated with VPN and VPC D

Destination	Target	Route type
10.1.0.0/16	VPC A attachment	propagated
10.2.0.0/16	VPC B attachment	propagated
10.3.0.0/16	VPC C attachment	propagated
10.4.0.0/16	VPC D attachment	propagated





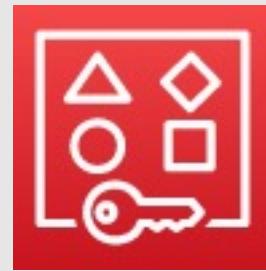
# Transit Gateway Best Practices

---

---

- Use a separate subnet for each transit gateway VPC attachment
- Create one network ACL and associate it with all of the subnets that are associated with the transit gateway
- Associate the same VPC route table with all subnets that are associated with the transit gateway, unless your network design requires multiple VPC route tables
- Use Border Gateway Protocol (BGP) Site-to-Site VPN connections
- Enable route propagation for AWS Direct Connect gateway attachments and BGP Site-to-Site VPN attachments
- More here:  
<https://docs.aws.amazon.com/vpc/latest/tgw/tgw-best-design-practices.html>

# VPC Sharing





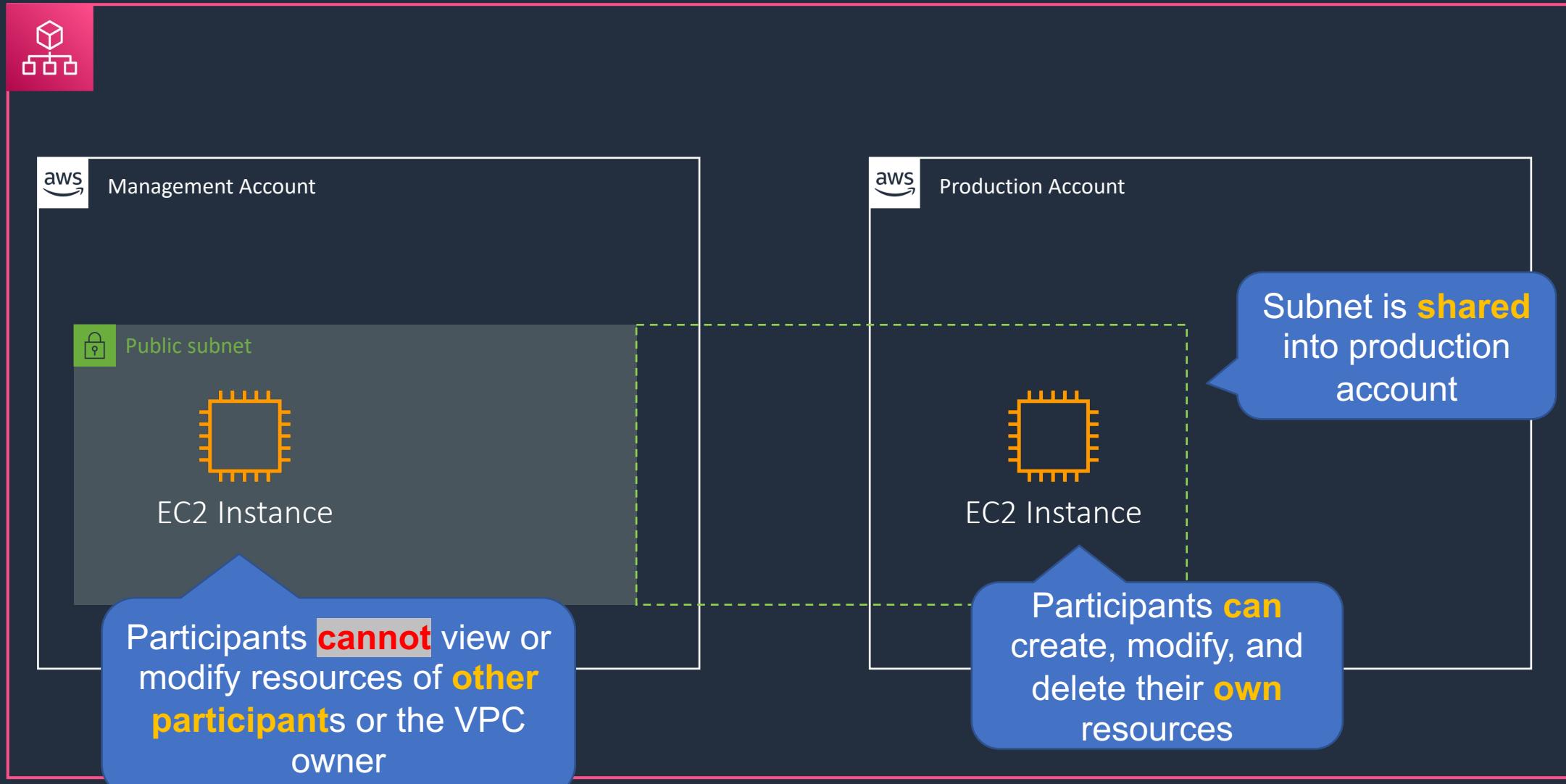
# Sharing VPCs with AWS Organizations and RAM

---

- VPCs sharing allows sharing of subnets with other accounts
- Uses AWS Resource Access Manager (AWS RAM)
- Integrates with AWS Organizations
- RAM can be used to share many other resources including:
  - Transit gateways
  - AWS Network Firewalls firewalls
  - Amazon Route 53 Resolver rules
  - AWS Cloud WAN



# Sharing VPCs with AWS Organizations and RAM



# Network Access Analyzer

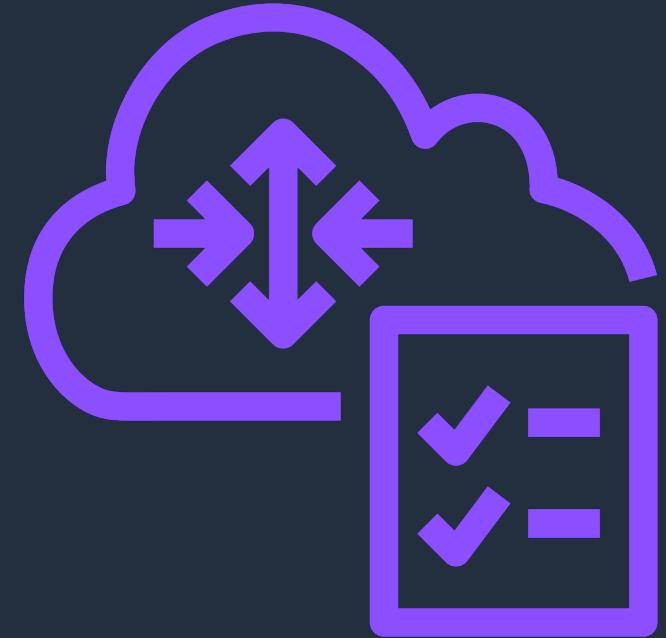




# Overview

---

- Identify unintended network access
- Verify compliance

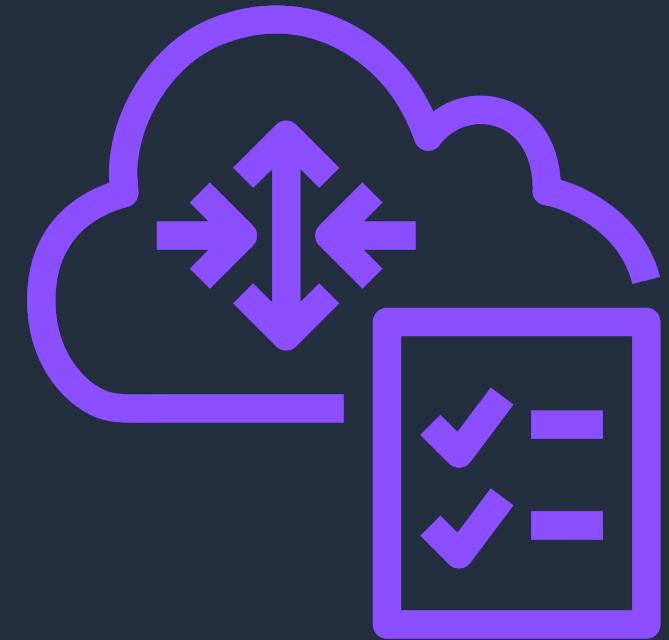




# Overview

---

- Does not actually send traffic
- Evaluates paths based on configuration





# Limitations

---

---



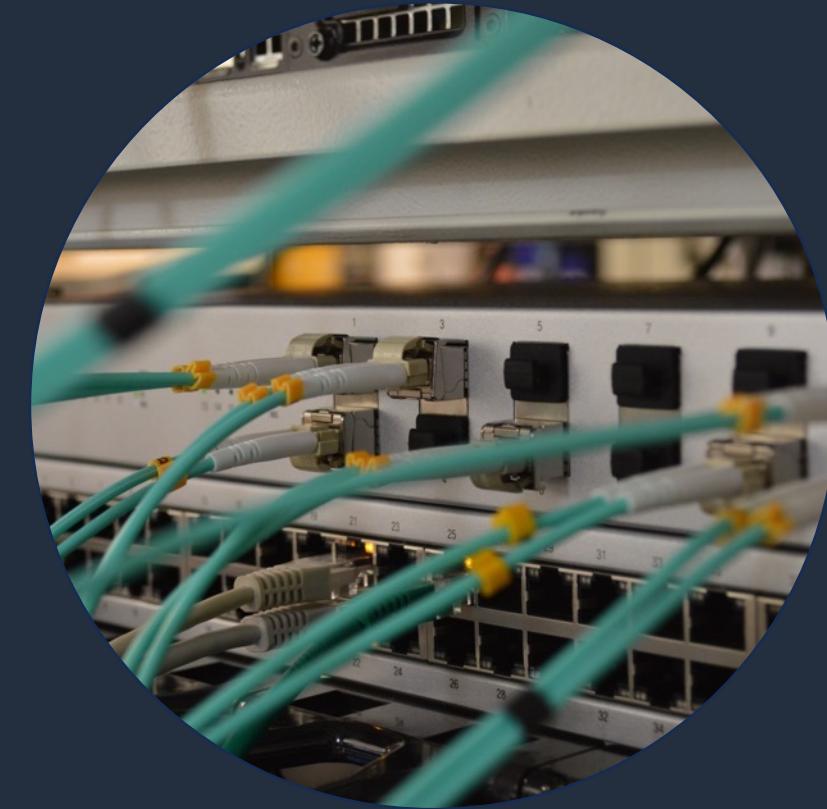
- Limited to AWS networks
- IPv4 TCP/UDP
- Cannot cross region or account boundaries



# Supported Resources

---

- Path Source/Destination
  - Network Interfaces
  - Endpoint objects
  - Internet & Virtual Private Gateways
  - Transit Gateway attachments
- Can trace paths through other VPC resources
- Not all configurations are supported





# Scopes

---



- Contain conditions to identify paths of interest
- Match Conditions
  - Identify non-compliant paths
  - Matches reported as **findings**
- Exclude Conditions
  - Optional
  - Identify compliant paths



# Scopes

---



- Conditions can evaluate by
  - Resource type or ID
  - IP configuration
  - Source/Destination
- Pre-created scopes & templates are provided.



# Analysis & Findings

---

---

- Performed per Scope
- All NICs in region included
  - Narrower conditions will reduce analysis time
  - 90 minute timeout
- Paths found by match condition are displayed
  - General source/destination info
  - Detailed path resources





# Quotas & Billing



- 1000 scopes
- 10,000 reports
- 25 concurrent analyses
- 1000 findings/analysis
- \$0.002/NIC



# Recap

---

---

Verify network access compliance.

---

**Scopes** define conditions for paths of interest, and can exclude compliant paths.

---

Analysis findings show paths of interest.

---

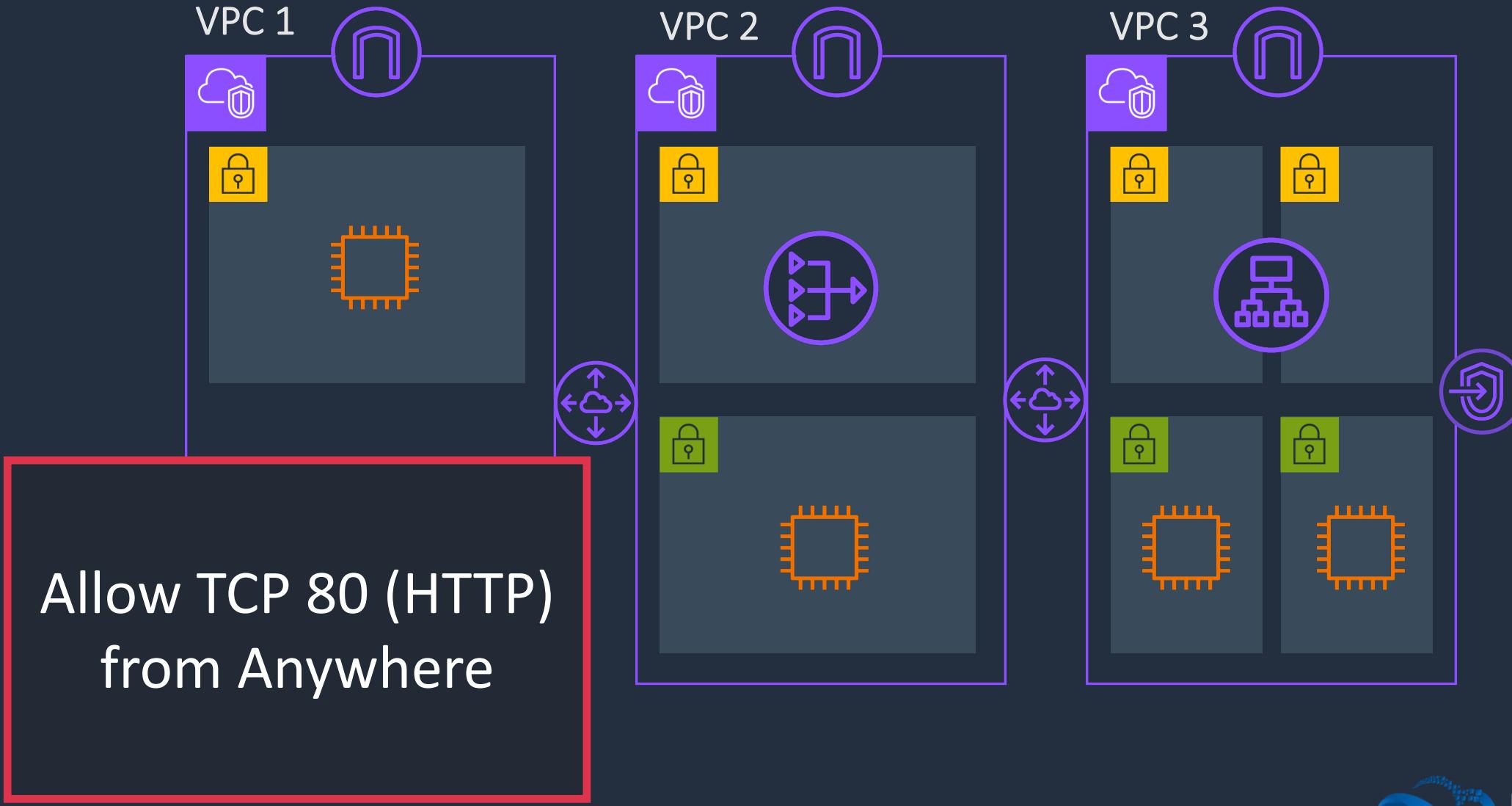
All NICs in a region are analyzed.

# Reviewing Findings with Network Access Analyzer





# Environment



# SECTION 7

## Edge Security

# DNS Name Resolution and Routing



# Amazon Route 53 Hosted Zones

Name	Type	Value	TTL
example.com	A	8.1.2.1	60
dev.example.com	A	8.1.2.2	60



Amazon Route 53

This is an example of a  
**public hosted zone**

What's the address for  
example.com?

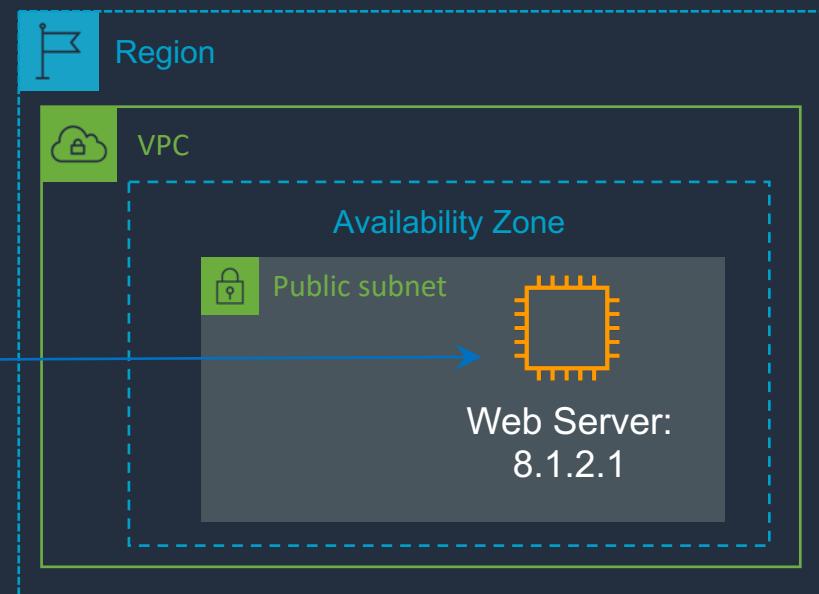
example.com

Address is 8.1.2.1



HTTP GET to 8.1.2.1

A **hosted zone** represents  
a set of records belonging  
to a domain



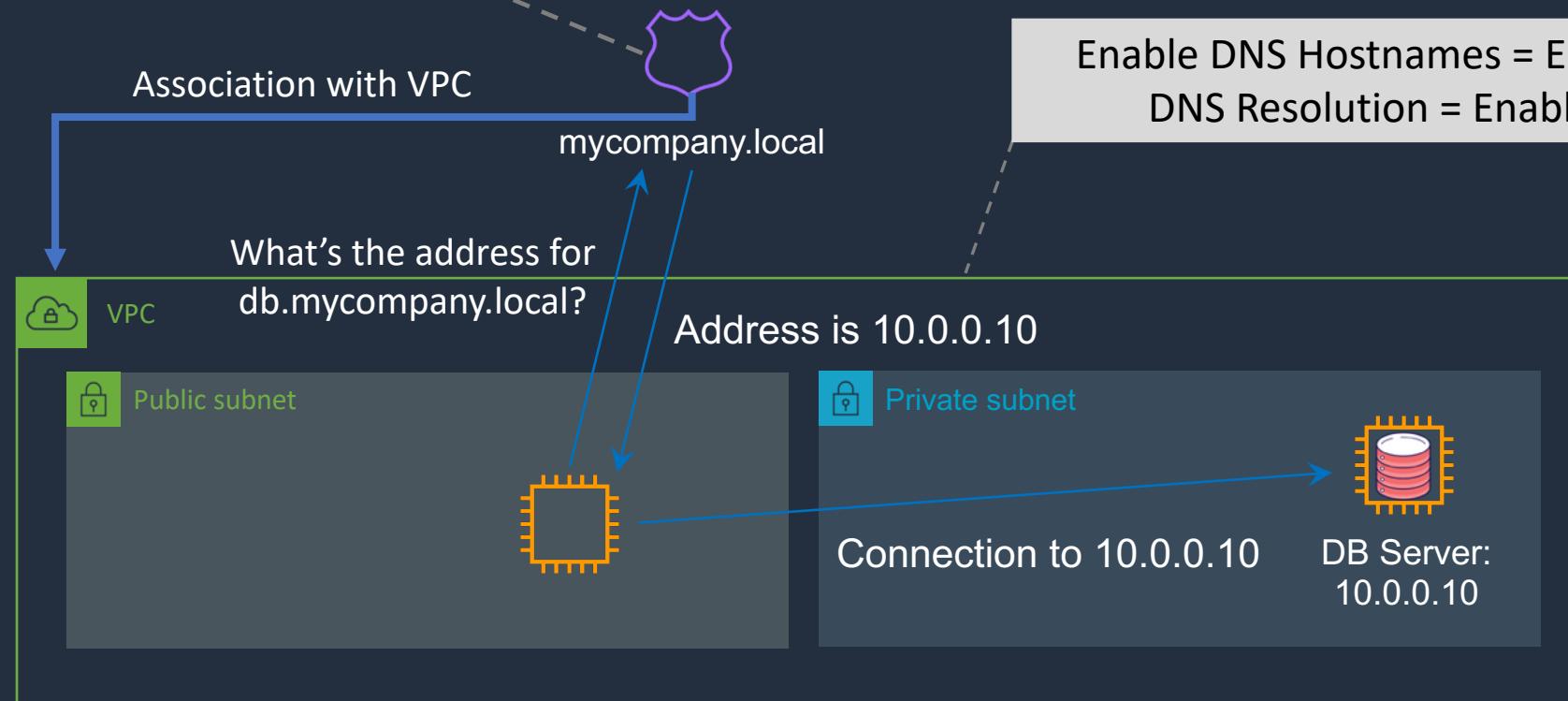
# Amazon Route 53 Hosted Zones

Name	Type	Value	TTL
db.mycompany.local	A	10.0.0.10	60
app.mycompany.local	A	10.0.0.11	60



This is an example of a  
**private hosted zone**

Amazon Route 53





# Amazon Route 53 Routing Policies

Routing Policy	What it does
Simple	Simple DNS response providing the IP address associated with a name
Failover	If primary is down (based on health checks), routes to secondary destination
Geolocation	Uses geographic location you're in (e.g. Europe) to route you to the closest region
Geoproximity	Routes you to the closest region within a geographic area
Latency	Directs you based on the lowest latency route to resources
Multivalue answer	Returns several IP addresses and functions as a basic load balancer
Weighted	Uses the relative weights assigned to resources to determine which to route to

# Amazon Route 53 Record Types

---

- A (address record)
- AAAA (IPv6 address record)
- **CNAME** (canonical name record)
- **Alias** (an Amazon Route 53-specific virtual record)
- CAA (certification authority authorization)
- MX (mail exchange record)
- NAPTR (name authority pointer record)
- NS (name server record)
- PTR (pointer record)
- SOA (start of authority record)
- SPF (sender policy framework)
- SRV (service locator)
- TXT (text record)

# Amazon Route 53 Record Types

---

CNAME Records	Alias Records
Route 53 charges for CNAME queries	Route 53 doesn't charge for alias queries to AWS resources
You can't create a CNAME record at the top node of a DNS namespace (zone apex)	You can create an alias record at the zone apex (however you can't route to a CNAME at the zone apex)
A CNAME record redirects queries for a domain name regardless of record type	Route 53 follows the pointer in an alias record only when the record type also matches
A CNAME can point to any DNS record that is hosted anywhere	An alias record can only point to a CloudFront distribution, Elastic Beanstalk environment, ELB, S3 bucket as a static website, or to another record in the same hosted zone that you're creating the alias record in
A CNAME record is visible in the answer section of a reply from a Route 53 DNS server	An alias record is only visible in the Route 53 console or the Route 53 API
A CNAME record is followed by a recursive resolver	An alias record is only followed inside Route 53. This means that both the alias record and its target must exist in Route 53

# CNAME and Alias Records



# Secure Content Delivery with CloudFront





# Amazon CloudFront Caching



There are 12+  
Regional Edge Caches

Regional  
Edge Cache

Edge location



Global  
Users

Regional  
Edge Cache

Edge location



Global  
Users

CloudFront improves  
performance for  
global users

There are 210+  
Edge locations



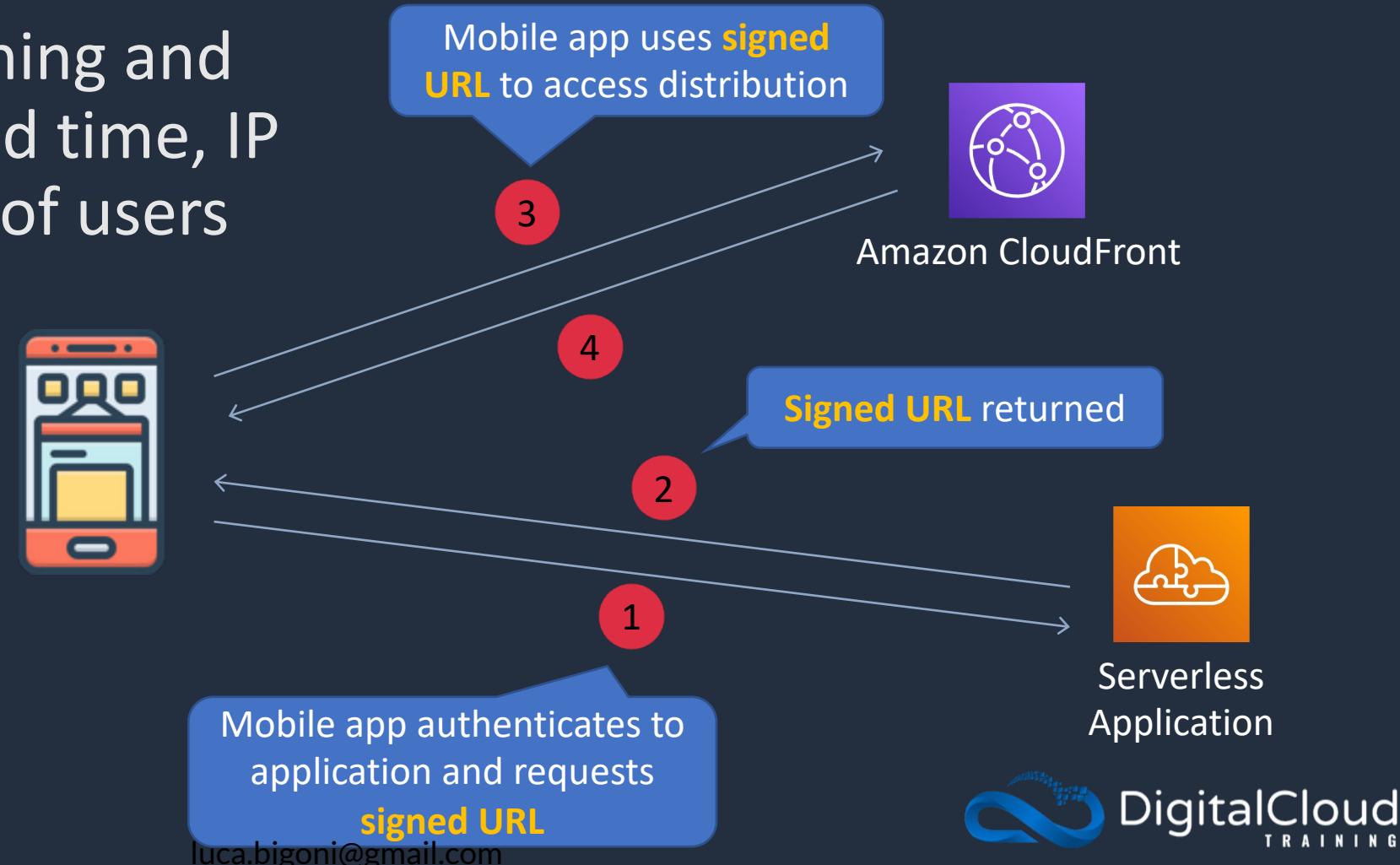
Global  
Users



# CloudFront Signed URLs

- Signed URLs provide more control over access to content
- Can specify beginning and expiration date and time, IP addresses/ranges of users

Signed URLs should be used for **individual files** and clients that don't support **cookies**





# CloudFront Signed Cookies

---

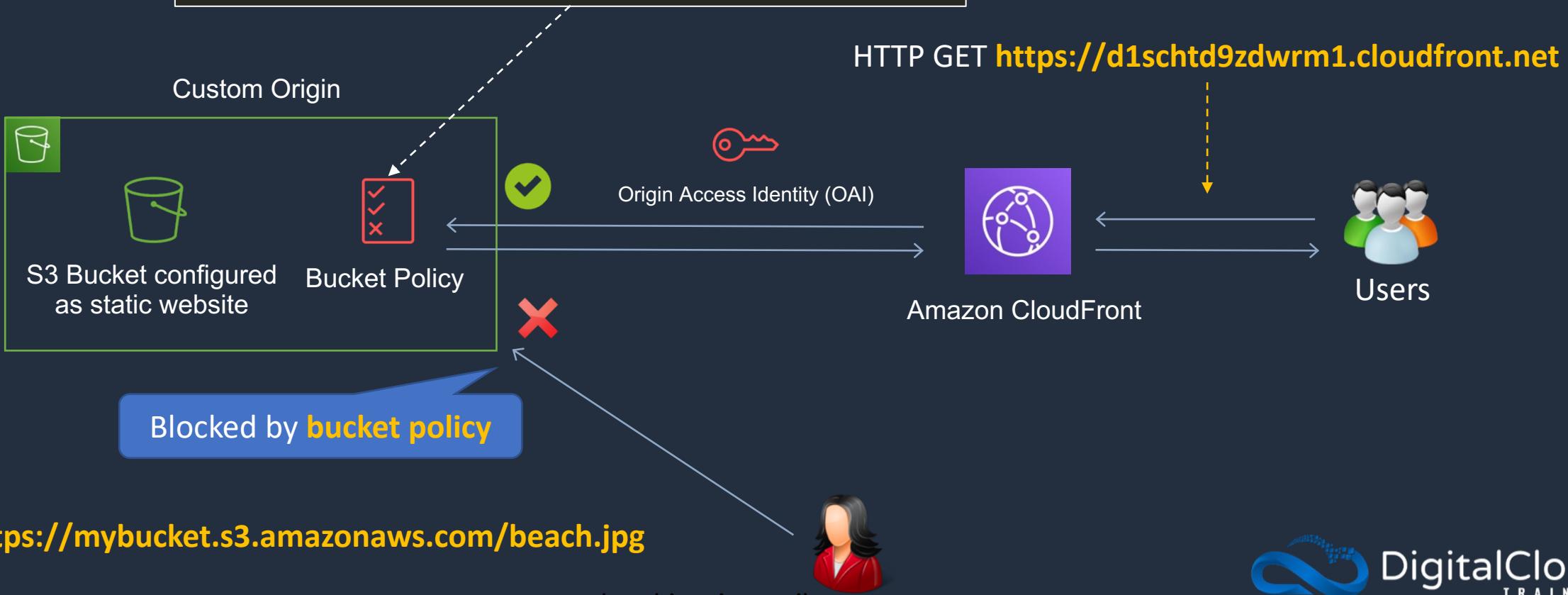
- Similar to Signed URLs
- Use signed cookies when you don't want to change URLs
- Can also be used when you want to provide access to **multiple restricted files** (Signed URLs are for individual files)



# Restrict Access to S3 Bucket

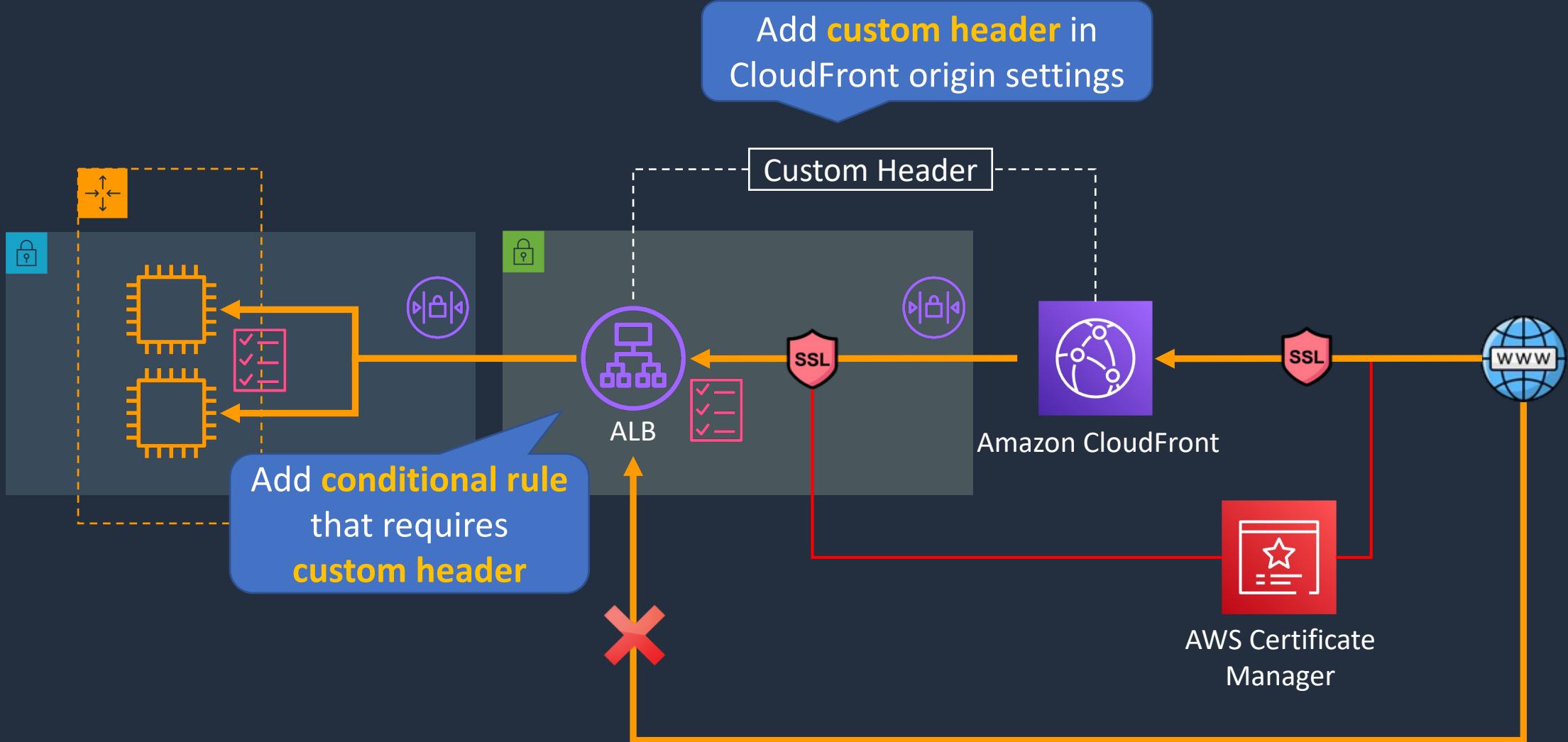
```
{  
    "Version": "2008-10-17",  
    "Id": "PolicyForCloudFrontPrivateContent",  
    "Statement": [  
        {  
            "Sid": "1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity E11A2JL2H306JJ"  
            },  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::mybucket/*"  
        }  
    ]  
}
```

Policy restricts access  
to the **OAI**





# Restrict Access to an ALB





# Additional Security Features

---

- AWS WAF web ACLs can be attached to CloudFront distributions
- Custom errors can be returned for blocked requests
- Field-level encryption protects sensitive data through the entire application stack
- Geo restriction / blocking can be used to prevent users in specific geographic locations from accessing content

# Configure Distribution Settings

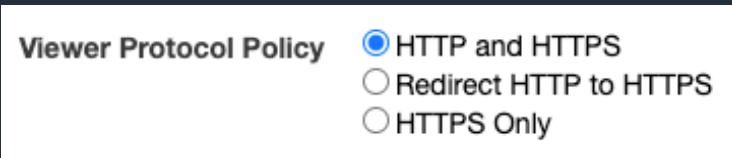


# CloudFront SSL/TLS and SNI



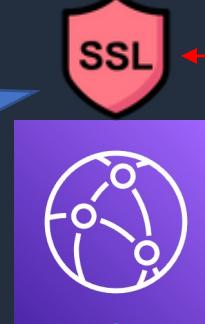


# CloudFront SSL/TLS



## Viewer Protocol

Certificate can be **ACM** or a trusted **third-party CA**



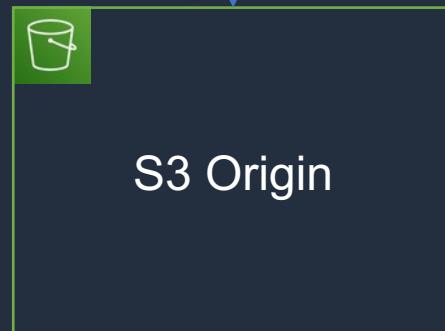
For CloudFront certificate must be issued in **us-east-1**



AWS Certificate Manager

Default CF **domain name** can be changed using **CNAMEs**

S3 has its **own** certificate (can't be changed)



S3 Origin

## Origin Protocol

Origin certificates must be **public certificates**



Custom Origin

Certificate can be **ACM** (ALB) or **third-party** (EC2)



# CloudFront Server Name Indication (SNI)



HTTP GET: <https://mypublicdomain.com>



HTTP GET: <https://myotherdomain.com>

Note: SNI works with  
browsers/clients released  
**after 2010** – otherwise  
need **dedicated IP**

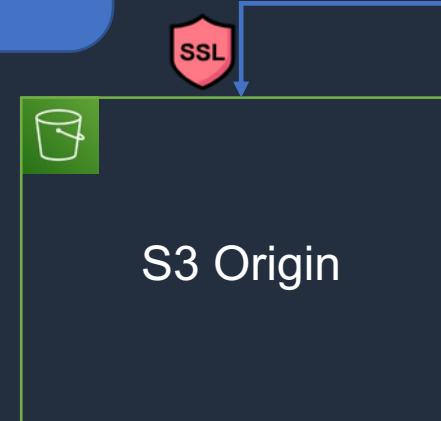
Name: [myotherdomain.com](https://myotherdomain.com)



Request URL includes  
domain name which  
**matches** certificate

Name: [mypublicdomain.com](https://mypublicdomain.com)

**Multiple** certificates  
share the **same IP**  
with SNI



# Lambda@Edge

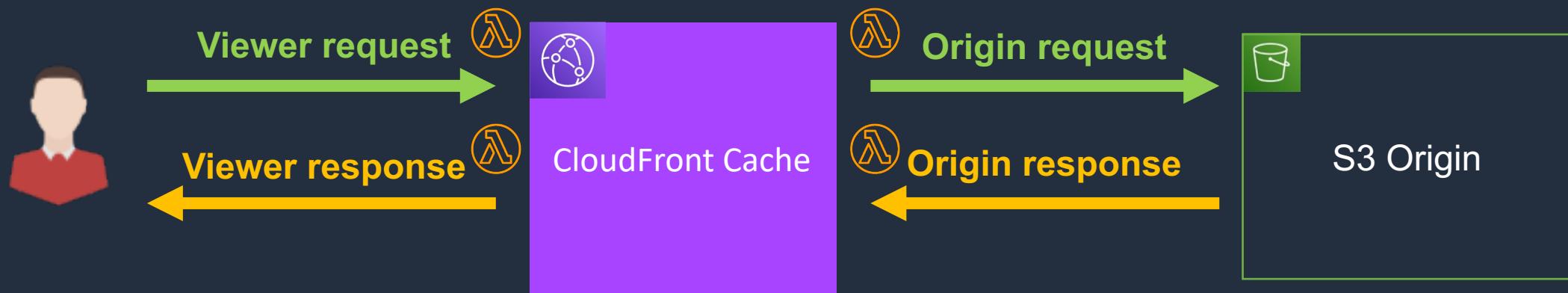


luca.bigoni@gmail.com



# Lambda@Edge

- Run Node.js and Python Lambda functions to customize the content CloudFront delivers
- Executes functions closer to the viewer
- Can be run at the following points
  - After CloudFront receives a request from a viewer (viewer request)
  - Before CloudFront forwards the request to the origin (origin request)
  - After CloudFront receives the response from the origin (origin response)
  - Before CloudFront forwards the response to the viewer (viewer response)



# AWS Web Application Firewall (WAF)





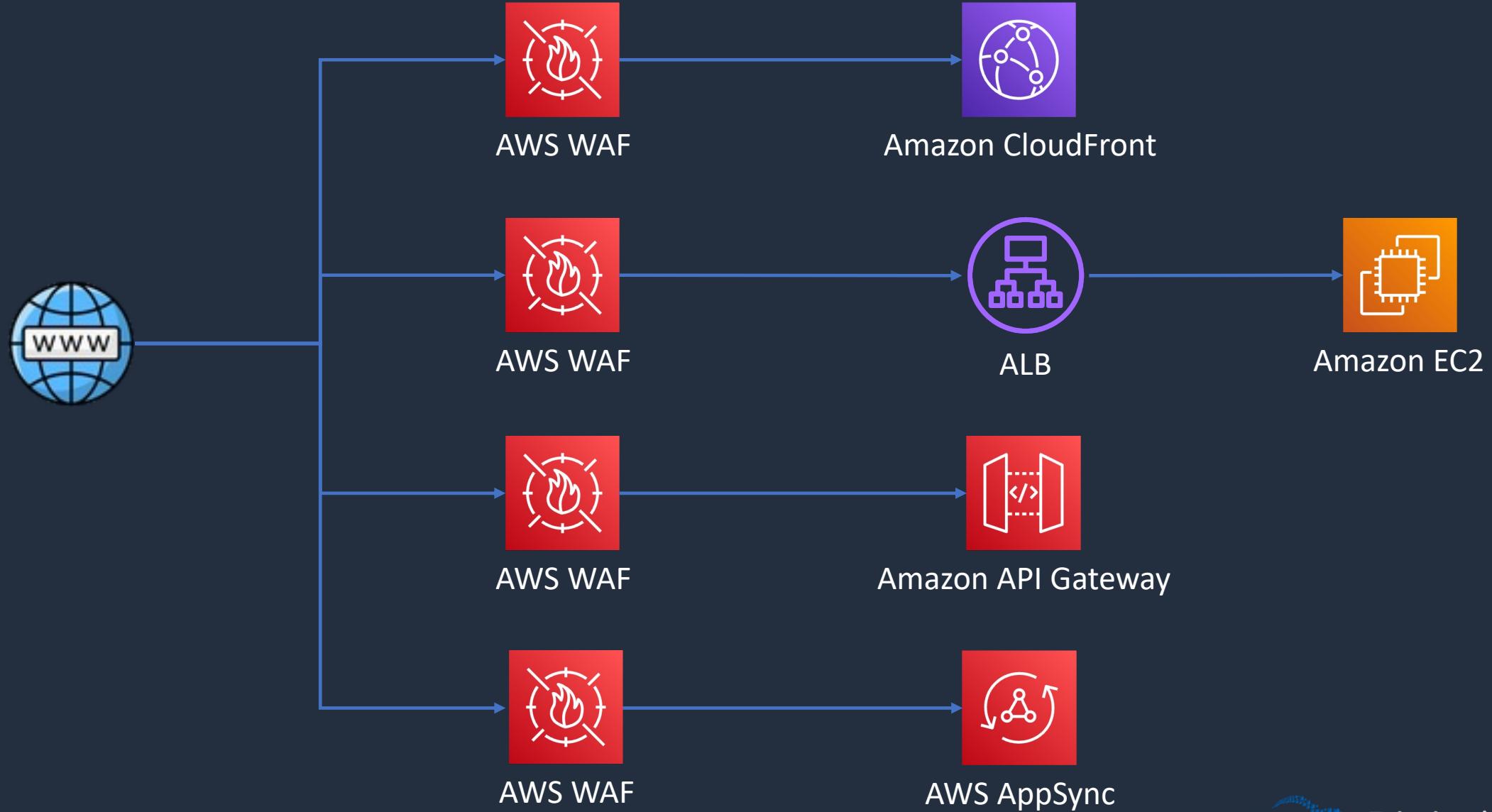
# AWS WAF

---

- AWS WAF is a web application firewall
- WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs
- WAF makes it easy to create rules that block common web exploits like SQL injection and cross site scripting



# AWS WAF





- **Web ACLs** – You use a web access control list (ACL) to protect a set of AWS resources
- **Rules** – Each rule contains a statement that defines the inspection criteria, and an action to take if a web request meets the criteria
- **Rule groups** – You can use rules individually or in reusable rule groups

Rule type

IP set  
Use IP sets to identify a specific list of IP addresses.

Rule builder  
Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.

Rule group  
Use a rule group to combine rules into a single logical set.



# AWS WAF

---

---

- **IP Sets** - An IP set provides a collection of IP addresses and IP address ranges that you want to use together in a rule statement
- **Regex pattern set** - A regex pattern set provides a collection of regular expressions that you want to use together in a rule statement



A **rule action** tells AWS WAF what to do with a web request when it **matches** the criteria defined in the rule:

- **Count** – AWS WAF counts the request but doesn't determine whether to allow it or block it. With this action, AWS WAF continues processing the remaining rules in the web ACL
- **Allow** – AWS WAF allows the request to be forwarded to the AWS resource for processing and response
- **Block** – AWS WAF blocks the request and the AWS resource responds with an HTTP 403 (Forbidden) status code



**Match** statements compare the web request or its origin against conditions that you provide

Match Statement	Description
Geographic match	Inspects the request's country of origin
IP set match	Inspects the request against a set of IP addresses and address ranges
Regex pattern set	Compares regex patterns against a specified request component
Size constraint	Checks size constraints against a specified request component
SQLi attack	Inspects for malicious SQL code in a specified request component
String match	Compares a string to a specified request component
XSS scripting attack	Inspects for cross-site scripting attacks in a specified request component

# AWS Shield



luca.bigoni@gmail.com

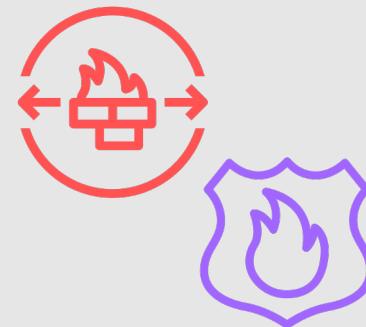


# AWS Shield

---

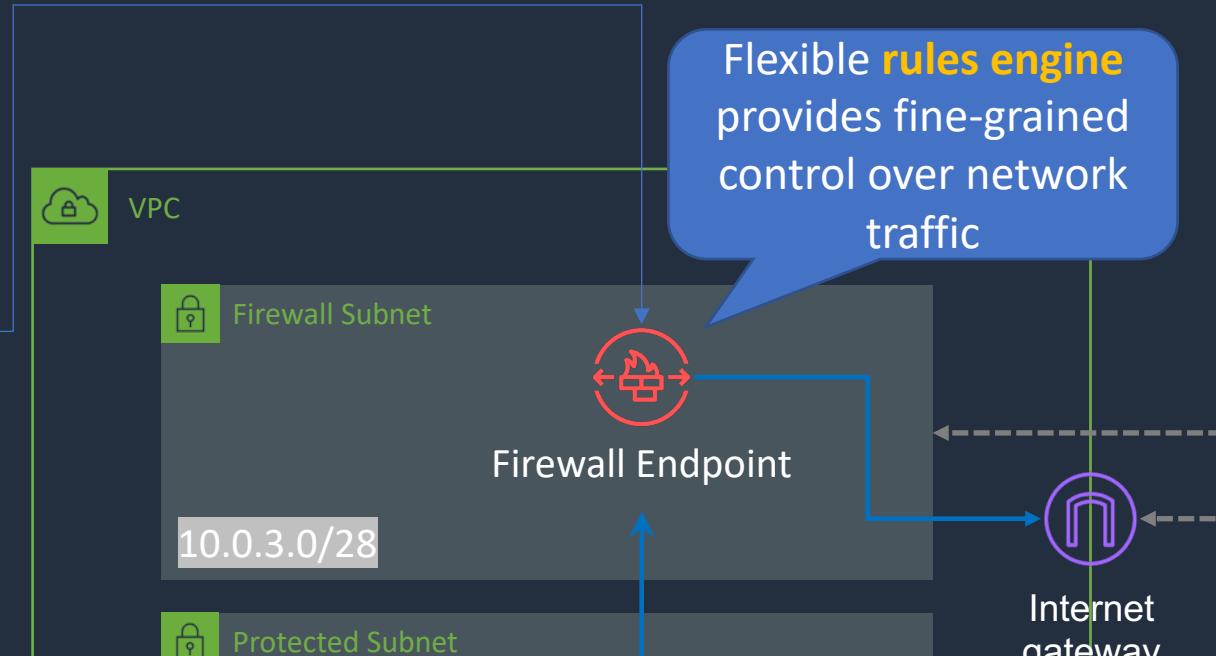
- AWS Shield is a managed Distributed Denial of Service (DDoS) protection service
- Safeguards web application running on AWS with always-on detection and automatic inline mitigations
- Helps to minimize application downtime and latency
- Two tiers –
  - **Standard** – no cost
  - **Advanced** - \$3k USD per month and 1 year commitment
- Integrated with Amazon CloudFront (standard included by default)

# Network Firewall and DNS Firewall





# AWS Network Firewall



Manage multiple AWS Network Firewall deployments

Traffic for resources in **protected** subnets is routed via **firewall** subnets

Firewall Subnet RT	
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

IGW Ingress RT	
Destination	Target
10.0.0.0/16	Local
10.0.0.0/24	vpce-id-az-a

Protected Subnet RT	
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	vpce-id-az-a



# AWS Network Firewall

---

---

- Managed service for **VPC network protection**
- **Includes:**
  - Stateful & Stateless firewall
  - Intrusion Prevention System (IPS)
  - Web filtering
- Works with **AWS Network Firewall** manager for centrally applying policies across VPCs / accounts
- Uses a **VPC endpoint** and **Gateway Load Balancer**
- Do not deploy resources in the firewall subnet
- For HA, allocate a subnet per AZ



# Route 53 Resolver DNS Firewall

---

- Filter and regulate outbound **DNS traffic for VPCs**
- Requests route through Route 53 Resolver for DNS
- Helps prevent DNS exfiltration of data
- Monitor and control the domains applications can query
- Can use AWS Firewall Manager to centrally configure and manage DNS Firewall
- Central management can span VPCs and accounts in AWS Organizations

# AWS Firewall Manager



# SECTION 8

## Data and Application Protection

# Encryption Primer





# Encryption In Transit vs At Rest



User

## Encryption In Transit

HTTPS Connection

Data is protected by  
**SSL/TLS** in transit



ALB

## Encryption At Rest

Amazon S3 **encrypts** the object as it is **written** to the bucket it



Unencrypted  
Object



Data encryption key



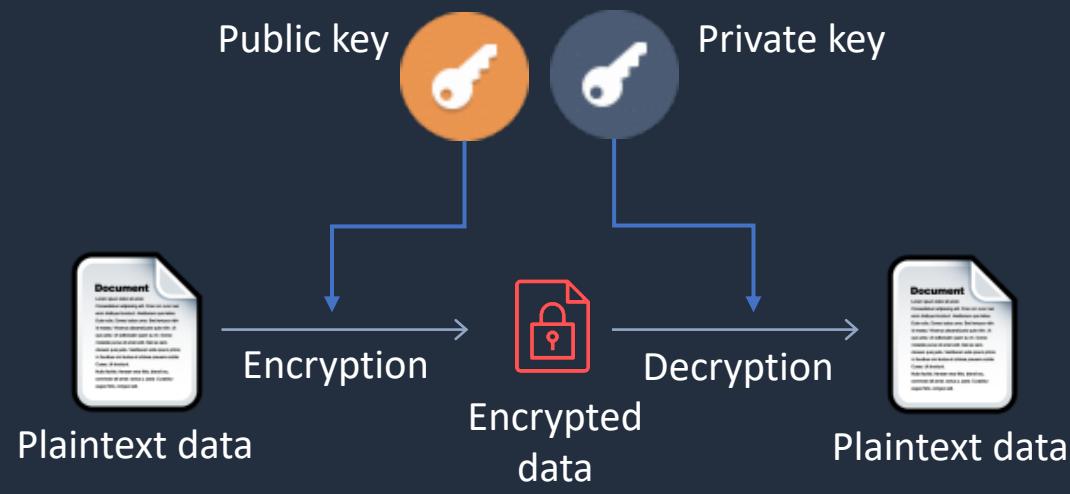
Encryption process



Encrypted  
bucket

# Asymmetric Encryption

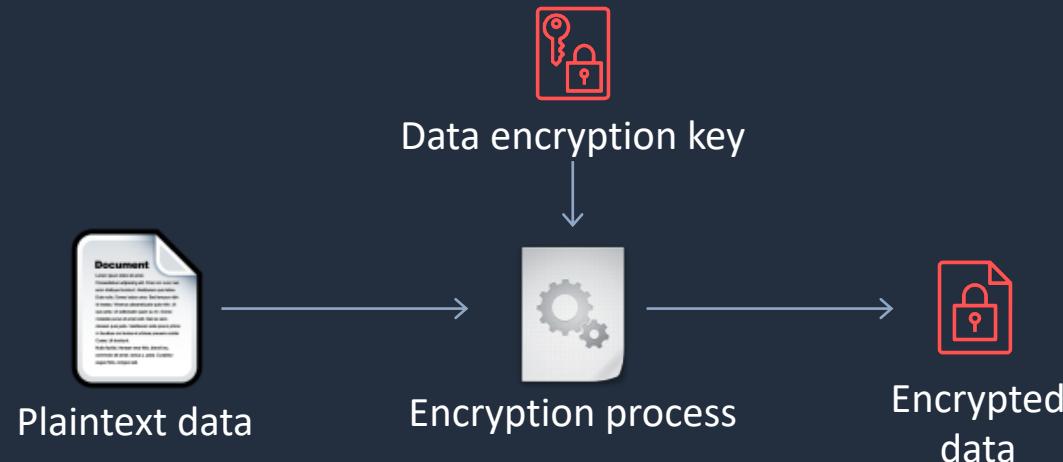
- Asymmetric encryption is also known as public key cryptography
- Messages encrypted with the public key can only be decrypted with the private key
- Messages encrypted with the private key can be decrypted with the public key
- Examples include SSL/TLS and SSH



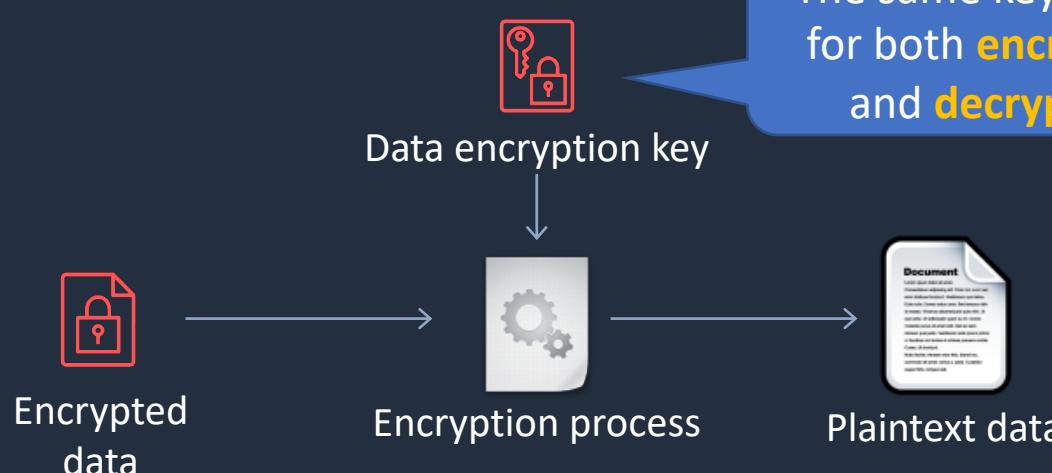


# Symmetric Encryption

## Encryption



## Decryption



# AWS Certificate Manager (ACM)



luca.bigoni@gmail.com



# AWS Certificate Manager (ACM)

---

---

- Create, store and renew SSL/TLS X.509 certificates
- Single domains, multiple domain names and wildcards
- Integrates with several AWS services including:
  - **Elastic Load Balancing**
  - **Amazon CloudFront**
  - **AWS Elastic Beanstalk**
  - **AWS Nitro Enclaves**
  - **AWS CloudFormation**



# AWS Certificate Manager (ACM)

- **Public certificates** are signed by the AWS public Certificate Authority
- You can also create a Private CA with ACM
- Can then issue private certificates
- You can also import certificates from third-party issuers

# Create SSL/TLS Certificate



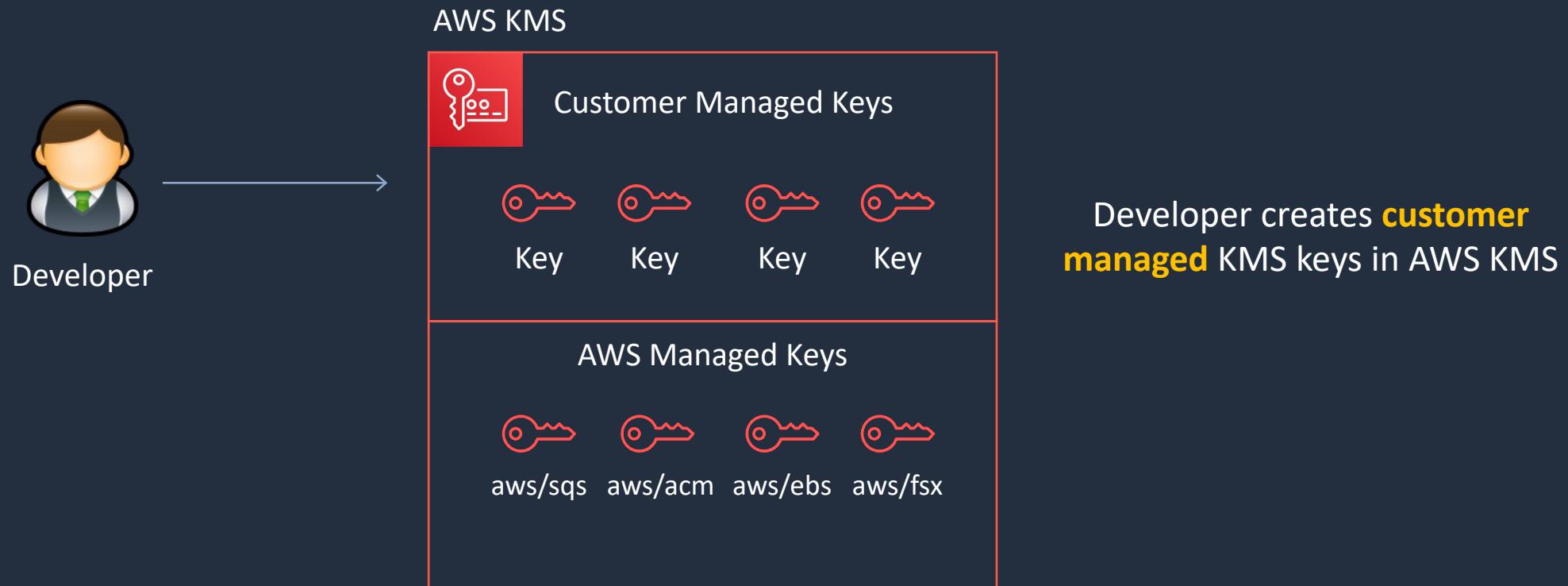
# AWS Key Management Service (KMS)





# AWS Key Management Service (KMS)

- Create and managed **symmetric** and **asymmetric** encryption keys
- The **KMS keys** are protected by hardware security modules (HSMs)





# KMS Keys

---

- KMS keys are the primary resources in AWS KMS
- Used to be known as “customer master keys” or CMKs
- The KMS key also contains the key material used to encrypt and decrypt data
- By default, AWS KMS creates the key material for a KMS key
- You can also import your own key material
- A KMS key can encrypt data up to 4KB in size
- A KMS key can generate, encrypt and decrypt Data Encryption Keys (DEKs)





# Alternative Key Stores

---

---

## External Key Store

- Keys can be stored outside of AWS to meet regulatory requirements
- You can create a KMS key in an AWS KMS external key store (XKS)
- All keys are generated and stored in an external key manager
- When using an XKS, key material never leaves your HSM

## Custom Key Store

- You can create KMS keys in an AWS CloudHSM custom key store
- All keys are generated and stored in an AWS CloudHSM cluster that you own and manage
- Cryptographic operations are performed solely in the AWS CloudHSM cluster you own and manage
- Custom key stores are not available for asymmetric KMS keys



# AWS Managed KMS Keys

- Created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS
- You cannot manage these KMS keys, rotate them, or change their key policies
- You also cannot use AWS managed KMS keys in cryptographic operations directly; the service that creates them uses them on your behalf

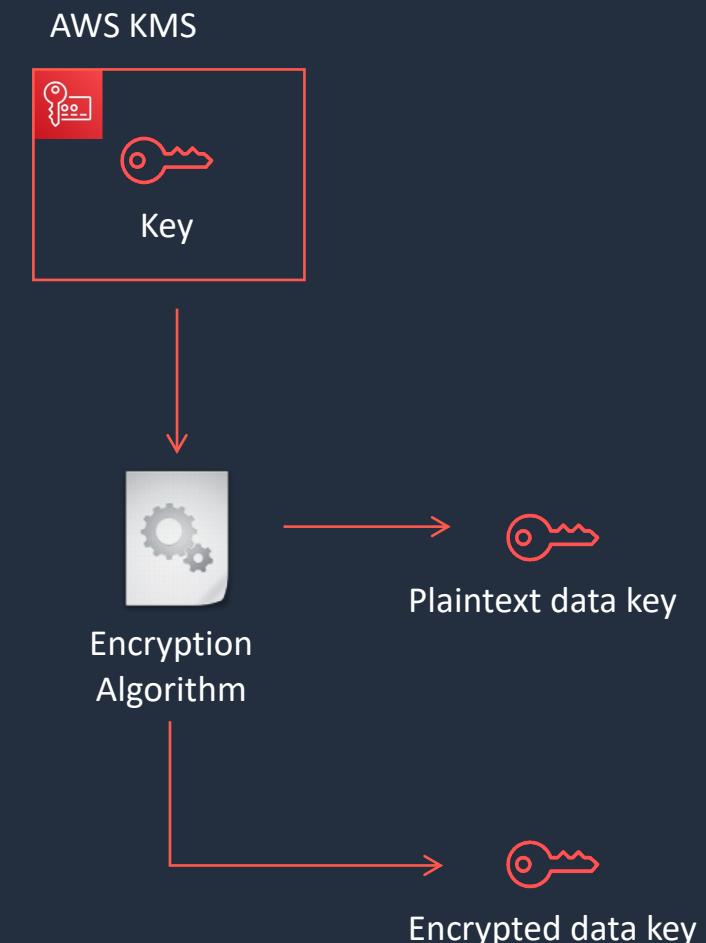
Alias	Key ID
aws/sqs	025b9386-b1f8-4fa9-84e2-ac3220b1de59
aws/acm	2d604e85-c2d4-42dc-ab0b-0b356f5fe26e
aws/codecommit	41fea9df-e447-4992-8af7-6ddec6d81175
aws/elasticfilesystem	460c4f05-fe98-4a35-b940-3e1992f04314
aws/glue	617516fe-bf19-4da4-a743-2b13c41973e1
aws/lambda	7f513d01-784b-41b9-9c51-51621db7b5e1
aws/ebs	b9baa4f6-3e87-4256-af6a-d181940df286
aws/lightsail	bc7ba666-8e17-444a-800c-d3d4be303a97
aws/fsx	cebc434c-ee2b-4a61-9b5a-f63be9fdb068
aws/kinesis	d99014b5-09d4-480d-9b2a-3a7d7e3e9c5b



# Data Encryption Keys

---

- Data keys are encryption keys that you can use to encrypt large amounts of data
- You can use AWS KMS keys to generate, encrypt, and decrypt data keys
- AWS KMS does not store, manage, or track your data keys, or perform cryptographic operations with data keys
- You must use and manage data keys outside of AWS KMS





# KMS Keys and Automatic Rotation

---

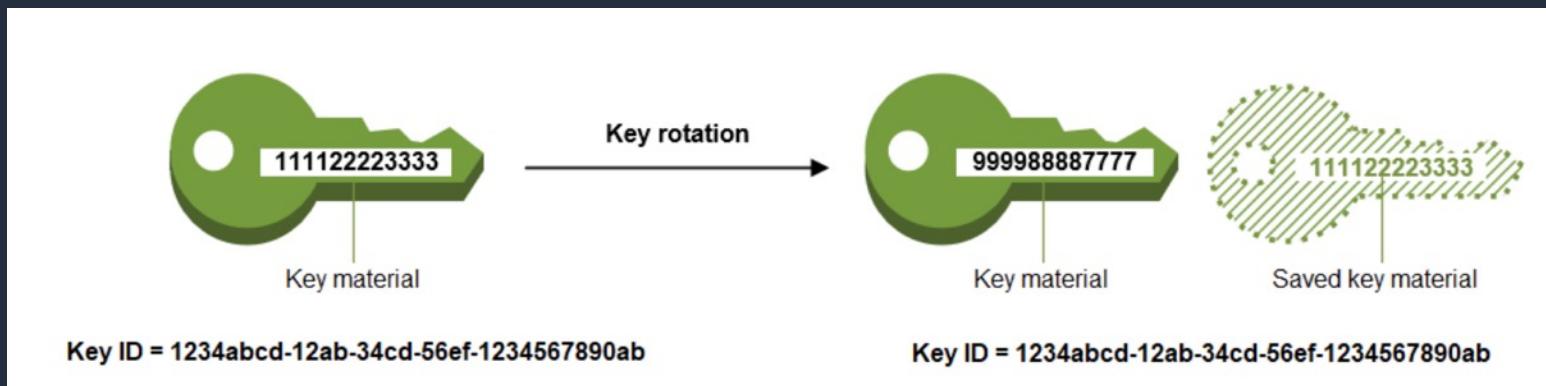
Type of KMS Key	Can view	Can manage	Used only for my AWS account	Automatic rotation
<b>Customer managed key</b>	Yes	Yes	Yes	Optional. Every 365 days
<b>AWS managed key</b>	Yes	No	Yes	Required. Every 365 days
<b>AWS owned key</b>	No	No	No	Varies

- You cannot enable or disable key rotation for AWS owned keys
- Automatic key rotation is supported only on symmetric encryption KMS keys with key material that AWS KMS generates (**Origin = AWS\_KMS**)



# KMS Keys and Automatic Rotation

- Automatic rotation generates new key material every year  
*(optional for customer managed keys)*



Rotation only changes the **key material** used for encryption, the KMS key remains the same



# KMS Keys and Automatic Rotation

---

---

## With automatic key rotation:

- The properties of the KMS key, including its key ID, key ARN, region, policies, and permissions, do not change when the key is rotated
- You do not need to change applications or aliases that refer to the key ID or key ARN of the KMS key
- After you enable key rotation, AWS KMS rotates the KMS key automatically every year

## Automatic key rotation is not supported on the following types of KMS keys:

- Asymmetric KMS keys
- HMAC KMS keys
- KMS keys in custom key stores
- KMS keys with imported key material

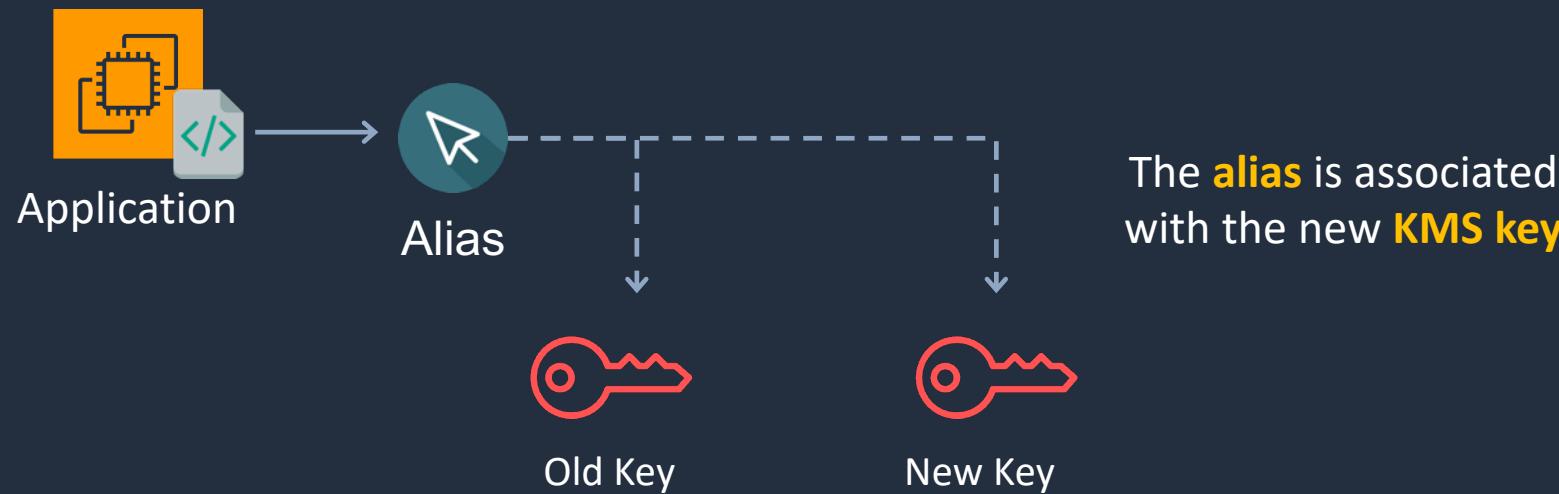
**Note:** You can rotate these KMS keys **manually**



# Manual Rotation

---

- Manual rotation is creating a new KMS key with a different key ID
- You must then update your applications with the new key ID
- You can use an **alias** to represent a KMS key so you don't need to modify your application code





# KMS Key Policies

- Key policies define management and usage permissions for KMS keys

```
{  
    "Sid": "Allow access for Key Administrators",  
    "Effect": "Allow",  
    "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},  
    "Action": [  
        "kms:Describe*",  
        "kms:Put*",  
        "kms>Create*",  
        "kms:Update*",  
        "kms:Enable*",  
        "kms:Revoke*",  
        "kms>List*",  
        "kms:Disable*",  
        "kms:Get*",  
        "kms>Delete*",  
        "kms:ScheduleKeyDeletion",  
        "kms:CancelKeyDeletion"  
    ],  
    "Resource": "*"
```

This key policy defines the **administrative actions** that are permitted for a key administrator



# KMS Key Policies

- Multiple policy statements can be combined to specify separate administrative and usage permissions

```
{  
  "Sid": "Allow use of the key",  
  "Effect": "Allow",  
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},  
  "Action": [  
    "kms:DescribeKey",  
    "kms:GenerateDataKey*",  
    "kms:Encrypt",  
    "kms:ReEncrypt*",  
    "kms:Decrypt"  
,  
  "Resource": "*"  
}
```

This key policy defines the **cryptographic** actions for encrypting and decrypting data with the KMS key



# KMS Key Policies

---

---

- Permissions can be specified for delegating use of the key to AWS services

```
{  
  "Sid": "Allow attachment of persistent resources",  
  "Effect": "Allow",  
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},  
  "Action": [  
    "kms>ListGrants",  
    "kms>CreateGrant",  
    "kms>RevokeGrant"  
  ],  
  "Resource": "*",  
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}  
}
```

Grants are useful for **temporary permissions** as they can be used without modifying key policies or IAM policies



# Additional Exam Tips

---

---

- To share snapshots with another account you must specify Decrypt and CreateGrant permissions
- The kms:ViaService condition key can be used to limit key usage to specific AWS services
- For example:

```
"Condition": {  
    "StringEquals": {  
        "kms:ViaService": [  
            "ec2.us-west-2.amazonaws.com",  
            "rds.us-west-2.amazonaws.com"  
        ]  
    }  
}
```



# Additional Exam Tips

---

---

- Cryptographic erasure means removing the ability to decrypt data and can be achieved when using **imported key material** and deleting that key material
- You must use the **DeleteImportedKeyMaterial** API to remove the key material
- An **InvalidKeyId** exception when using SSM Parameter Store indicates the KMS key is not enabled
- Make sure you know the differences between AWS managed and customer managed KMS keys and automatic vs manual rotation

# Create Custom KMS Keys



# AWS CloudHSM





# AWS CloudHSM

---

---

- AWS CloudHSM is a cloud-based hardware security module (HSM)
- Generate and use your own encryption keys on the AWS Cloud
- CloudHSM runs in your Amazon VPC
- Uses FIPS 140-2 level 3 validated HSMs
- Managed service and automatically scales
- Retain control of your encryption keys - you control access (and AWS has no visibility of your encryption keys)



# AWS CloudHSM Use Cases

---

---

- Offload SSL/TLS processing from web servers
- Protect private keys for an issuing certificate authority (CA)
- Store the master key for Oracle DB Transparent Data Encryption
- Custom key store for AWS KMS – retain control of the HSM that protects the master keys



# AWS CloudHSM vs KMS

---

	CloudHSM	AWS KMS
<b>Tenancy</b>	Single-tenant HSM	Multi-tenant AWS service
<b>Availability</b>	Customer-managed durability and available	Highly available and durable key storage and management
<b>Root of Trust</b>	Customer managed root of trust	AWS managed root of trust
<b>FIPS 140-2</b>	Level 3	Level 2 / Level 3
<b>3<sup>rd</sup> Party Support</b>	Broad 3 <sup>rd</sup> Party Support	Broad AWS service support

# Protecting Data on S3, EBS, and EFS





# S3 Encryption

## Server-side encryption with S3 managed keys (SSE-S3)



- S3 managed keys
- Unique object keys
- Master key
- AES 256



Encryption / decryption



## Server-side encryption with AWS KMS managed keys (SSE-KMS)



- KMS managed keys
- KMS key can be customer generated



Encryption / decryption



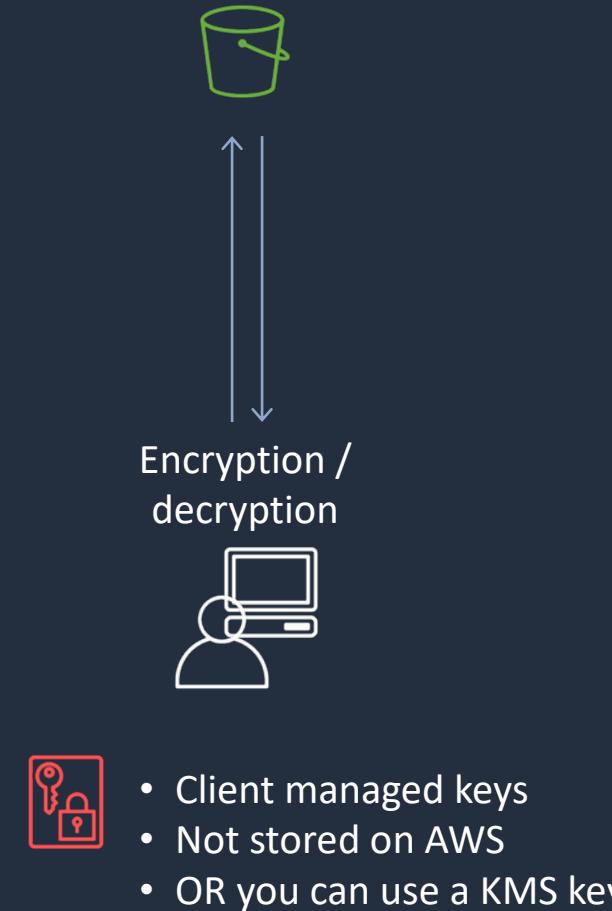


# S3 Encryption

## Server-side encryption with client provided keys (SSE-C)



## Client-side encryption





# S3 Default Encryption

---

- Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket
- You can set default encryption on a bucket so that all new objects are encrypted when they are stored in the bucket
- The objects are encrypted using server-side encryption
- Amazon S3 encrypts objects before saving them to disk and decrypts them when the objects are downloaded
- There is no change to the encryption of objects that existed in the bucket before default encryption was enabled



# Prevent uploads of unencrypted objects

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjPolicy",  
    "Statement": [  
        {  
            "Sid": "DenyIncorrectEncryptionHeader",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::<bucket_name>/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-server-side-encryption": "AES256"  
                }  
            }  
        },  
        {  
            "Sid": "DenyUnEncryptedObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::<bucket_name>/*",  
            "Condition": {  
                "Null": {  
                    "s3:x-amz-server-side-encryption": true  
                }  
            }  
        }  
    ]  
}
```

Enforces encryption  
using SSE-S3

For SSE-KMS use  
"aws:kms"

## Example PUT request

```
PUT /example-object HTTP/1.1  
Host: myBucket.s3.amazonaws.com  
Date: Wed, 8 Jun 2016 17:50:00 GMT  
Authorization: authorization string  
Content-Type: text/plain  
Content-Length: 11434  
x-amz-meta-author: Janet  
Expect: 100-continue  
x-amz-server-side-encryption: AES256  
[11434 bytes of object data]
```



# Glacier Vault Lock and Vault Access Policies

---

---

## S3 Glacier Vault Lock:

- S3 Glacier Vault Lock enforces compliance controls for S3 Glacier vaults with a vault lock policy
- Can specify controls such as “write once read many” (WORM) in a vault lock policy and lock the policy from future edits
- Once locked, the policy can no longer be changed

## S3 Glacier Vault Access Policy:

- Resource-based policy that you can use to manage permissions to your vault
- You can create one vault access policy for each vault to manage permissions
- You can modify permissions in a vault access policy at any time

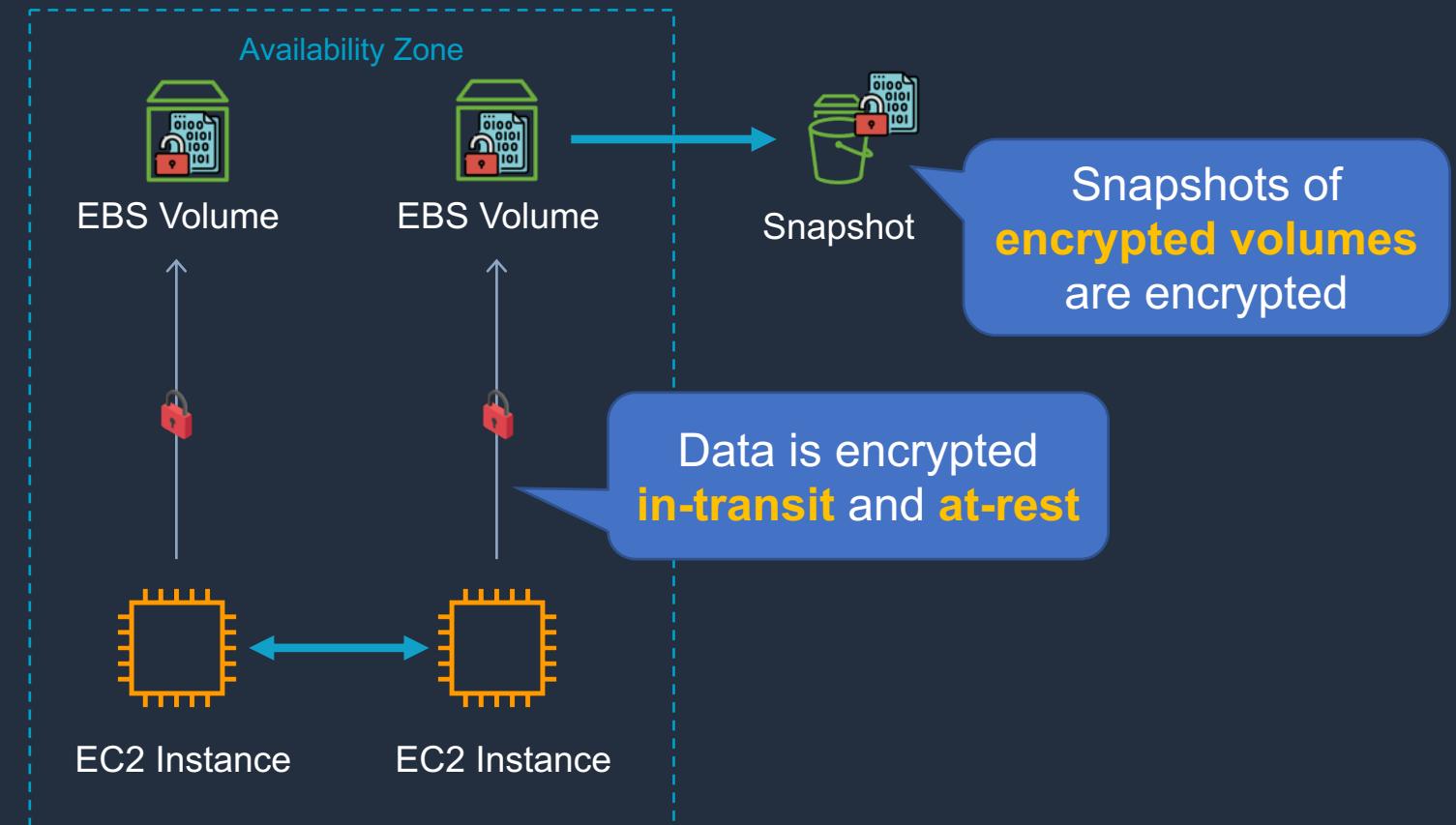


# Amazon EBS Encryption

EBS encryption affects:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

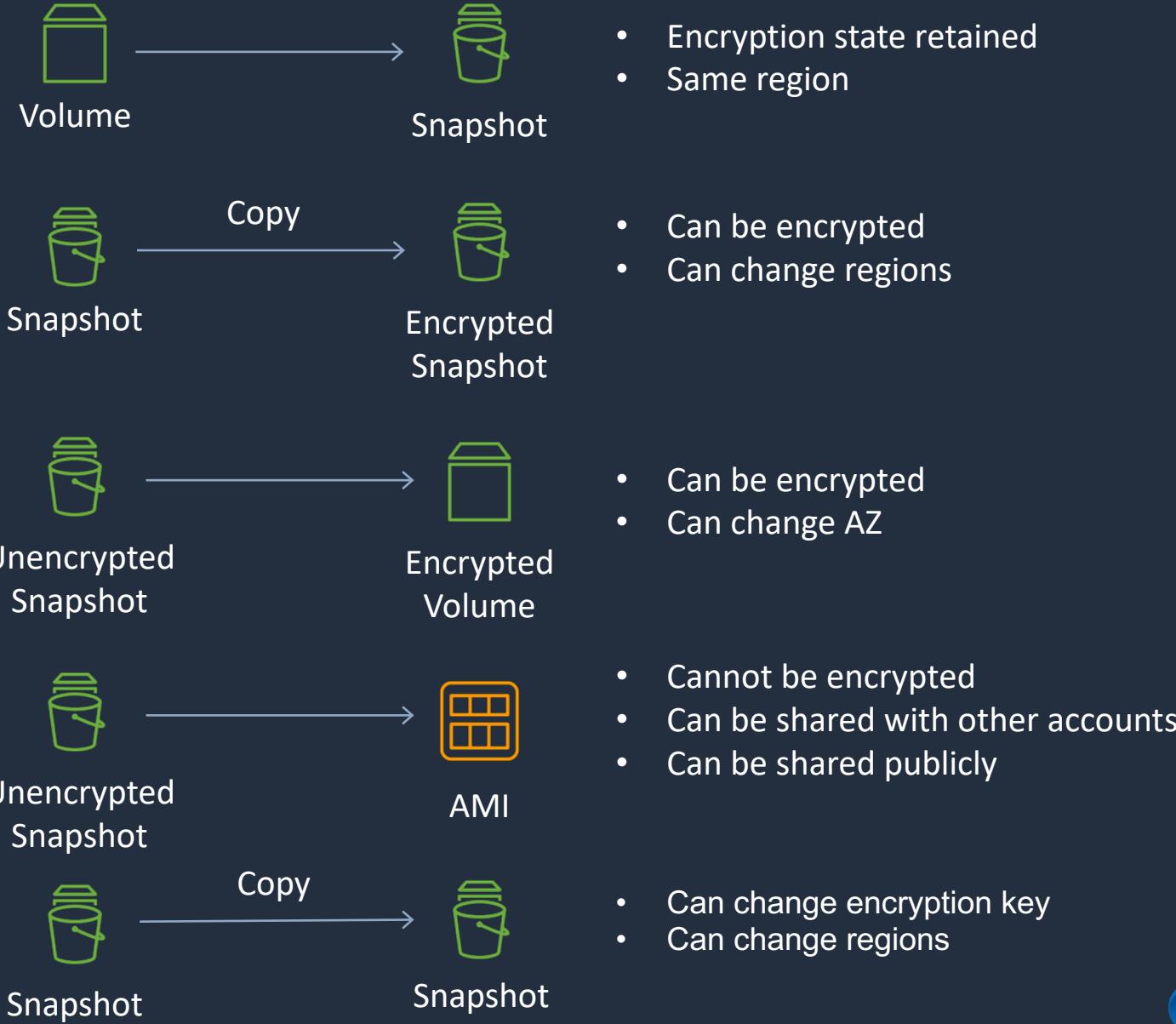
Traffic between AZs is **encrypted**



Traffic between instances is **encrypted** in transit for some instance types  
luca.bigoni@gmail.com

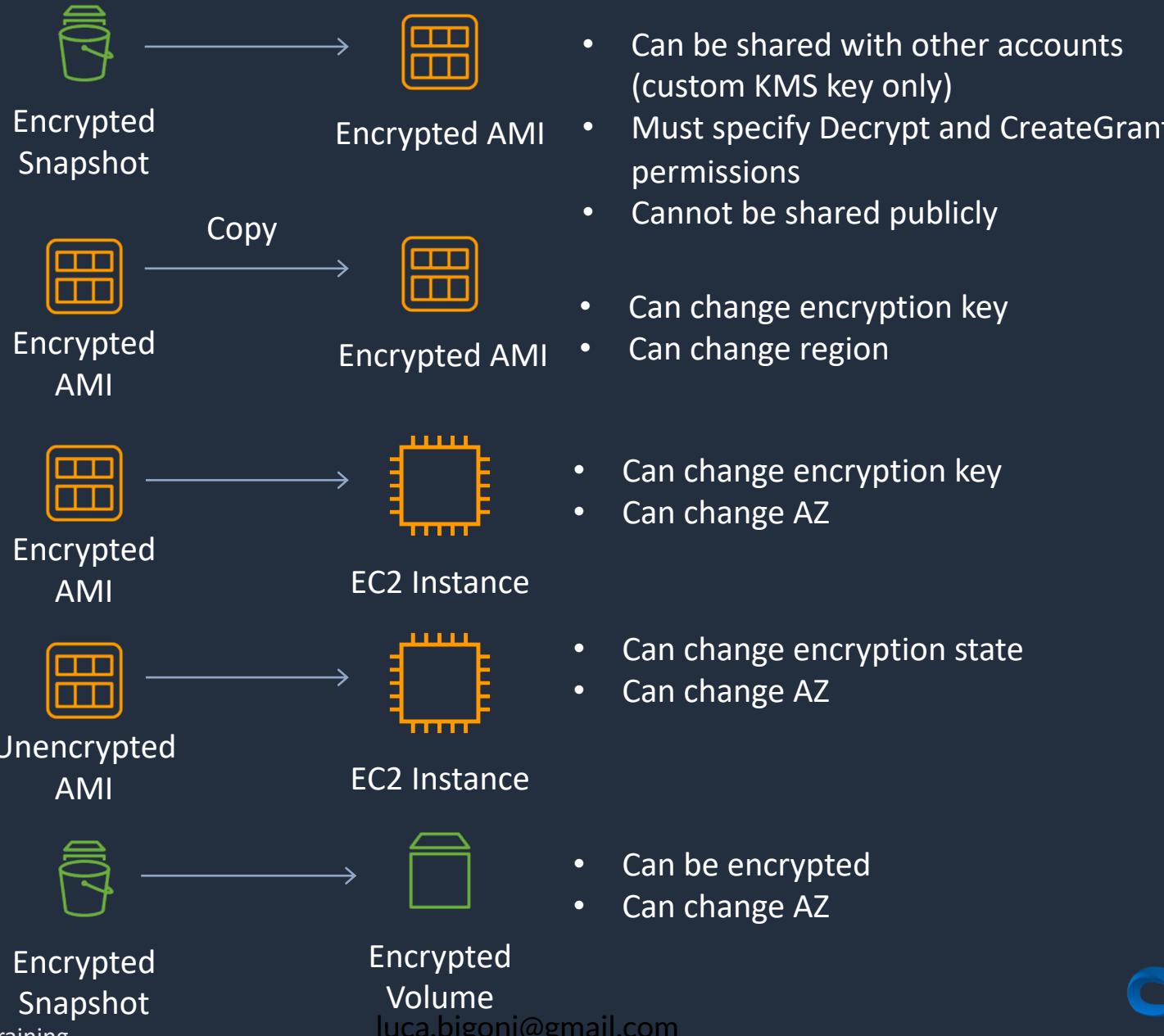


# Copying and Sharing AMIs and Snapshots





# Copying and Sharing AMIs and Snapshots

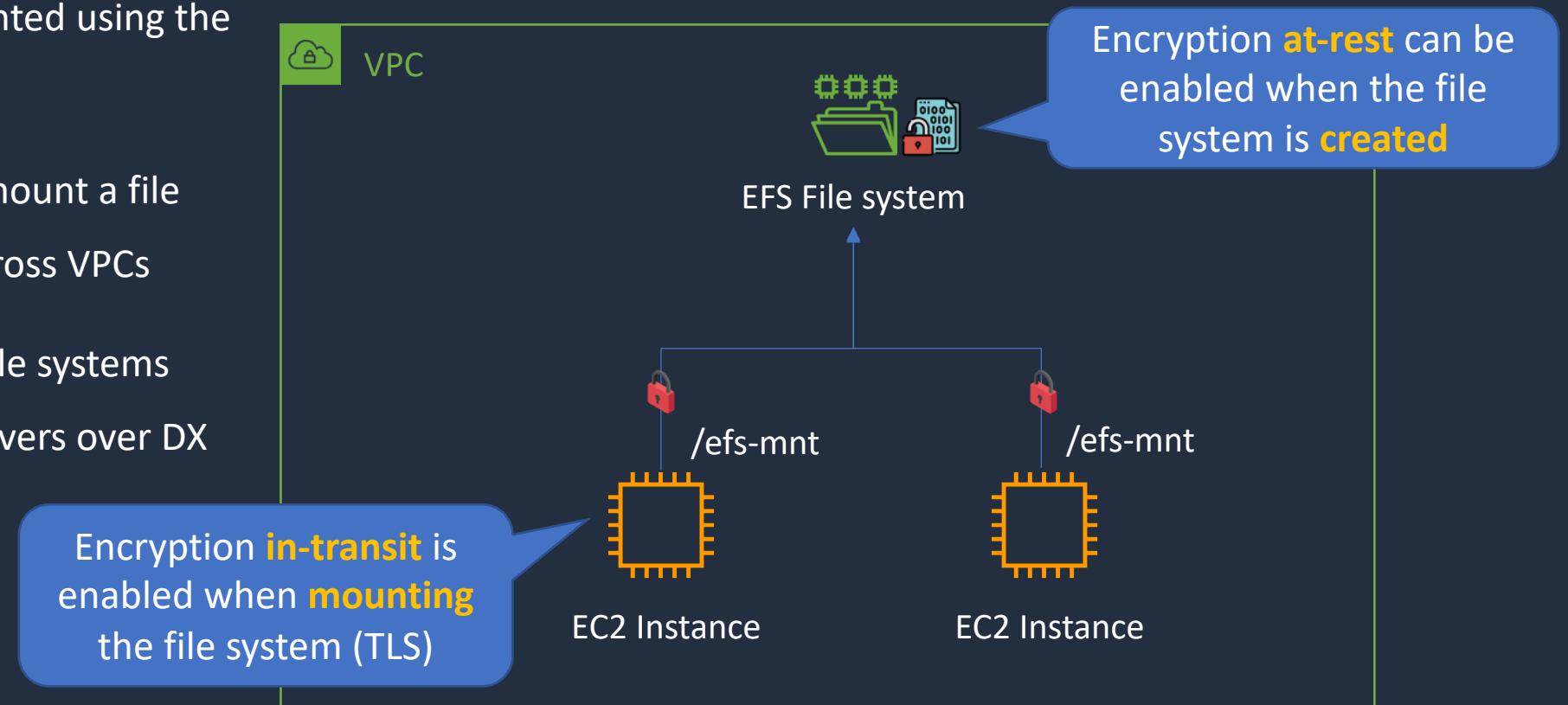




# Amazon EFS Encryption

---

- EFS is the Amazon Elastic File System
- File systems are mounted using the NFS protocol
- Many instances can mount a file system within and across VPCs
- You can also mount file systems from on-premises servers over DX or VPN



# Enforce KMS Encryption for S3 Bucket



# Copy Encrypted Snapshot Across Accounts



# Database Protection – DynamoDB and RDS

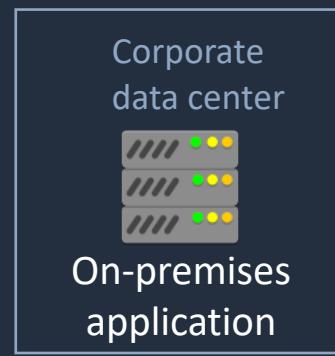




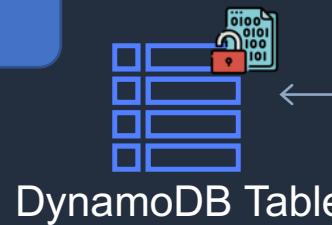
# Amazon DynamoDB

DynamoDB data is encrypted **at rest**

EC2 instances connect using **private** addresses



VPN or Direct Connect

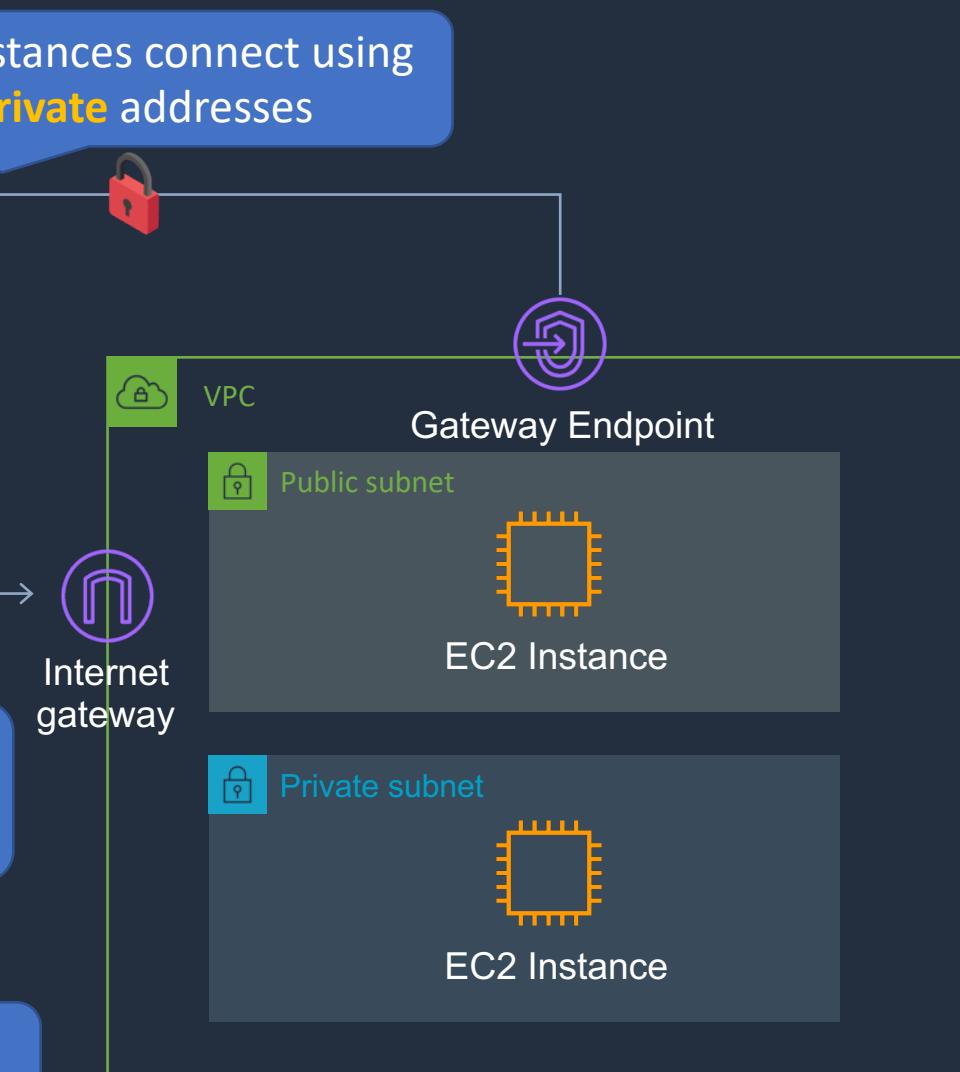


KMS key can be:

- AWS owned (default/free)
- AWS managed
- Customer managed

DynamoDB supports **identity-based** policies

luca.bigoni@gmail.com





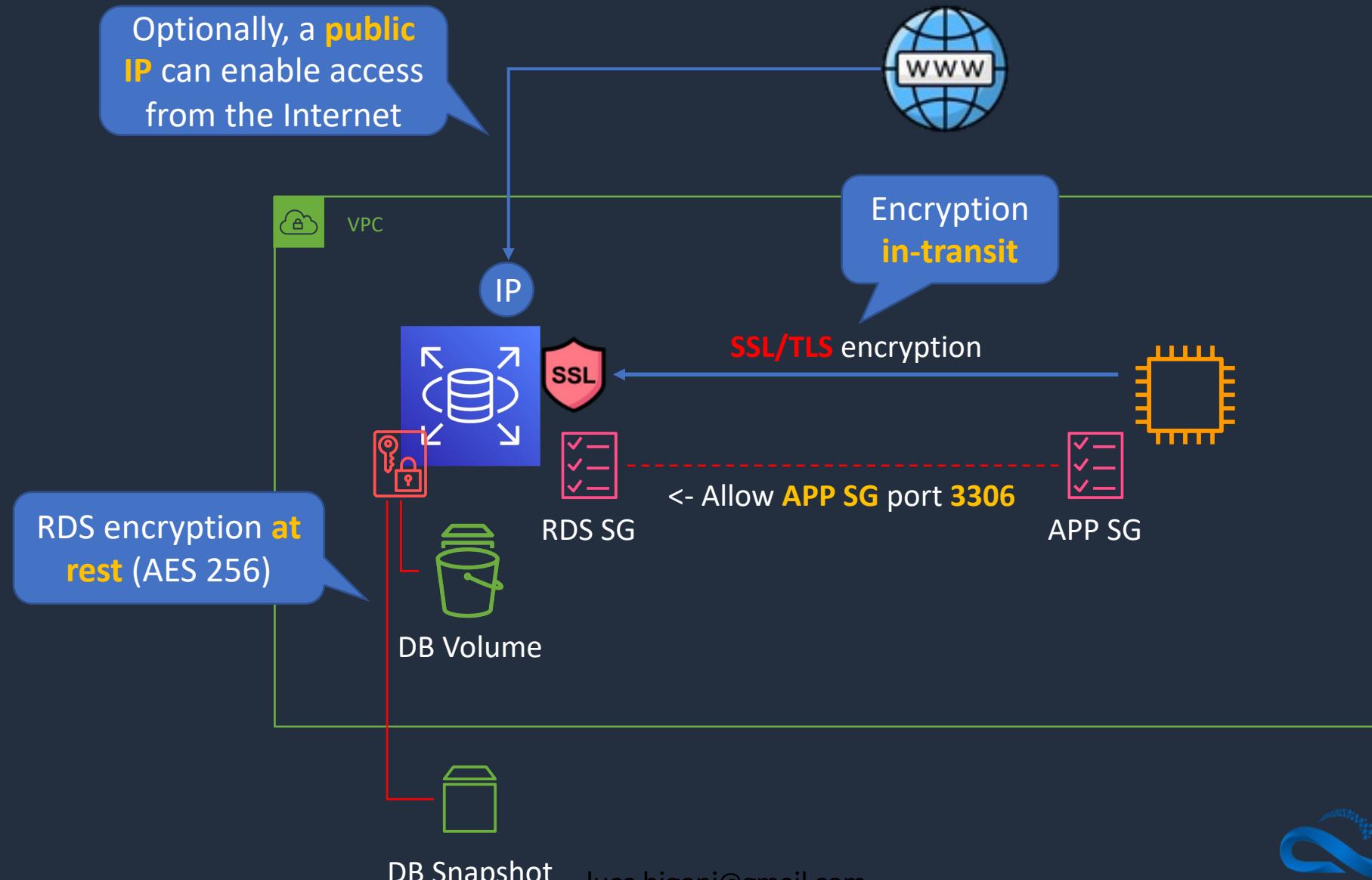
# Amazon RDS Security

---

- The Amazon Relational Database service runs on instances within a VPC
- Encryption **at rest** can be enabled
- Encryption includes DB storage, backups, read replicas and snapshots
- You can only enable encryption for an Amazon RDS DB instance when you create it
- DB instances that are encrypted can't be modified to disable encryption
- Uses AES 256 encryption
- RDS for Oracle and SQL Server is supported using Transparent Data Encryption (TDE)
- AWS KMS is used for managing encryption keys



# Amazon RDS Security





# Amazon RDS Security

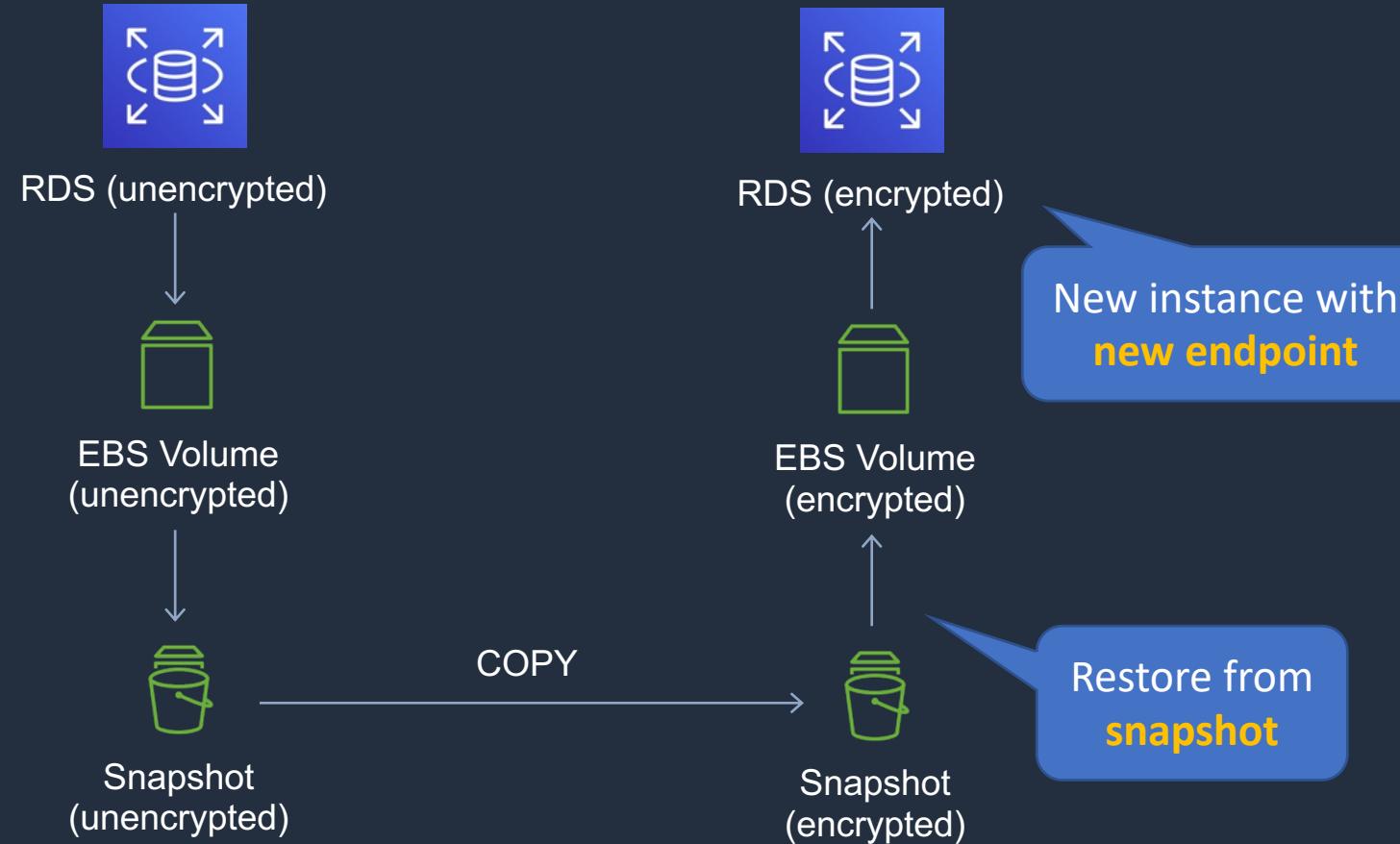
---

- You can't have:
  - An **encrypted** read replica of an **unencrypted** DB instance
  - An **unencrypted** read replica of an **encrypted** DB instance
- Read replicas of encrypted primary instances are encrypted
- The same KMS key is used if in the same Region as the primary
- If the read replica is in a different Region, a different KMS key is used
- You can't restore an unencrypted backup or snapshot to an encrypted DB instance



# Amazon RDS Security

---



# Encryption Options for AWS Databases



# KMS Keys - Schedule Key Deletion



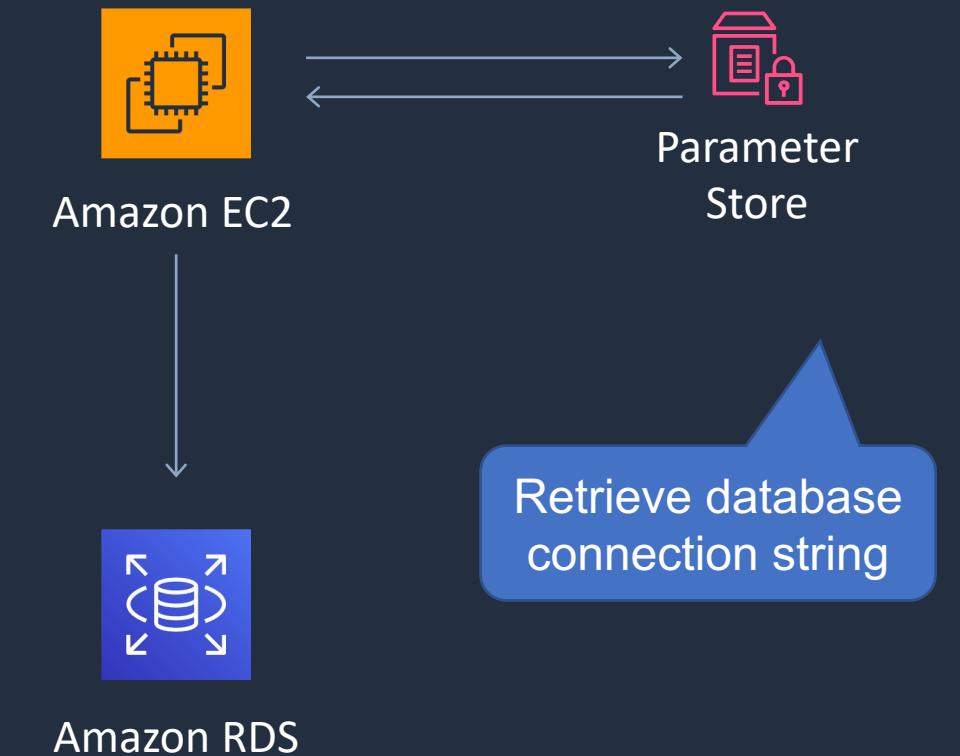
# Storing Secrets





# AWS SSM Parameter Store

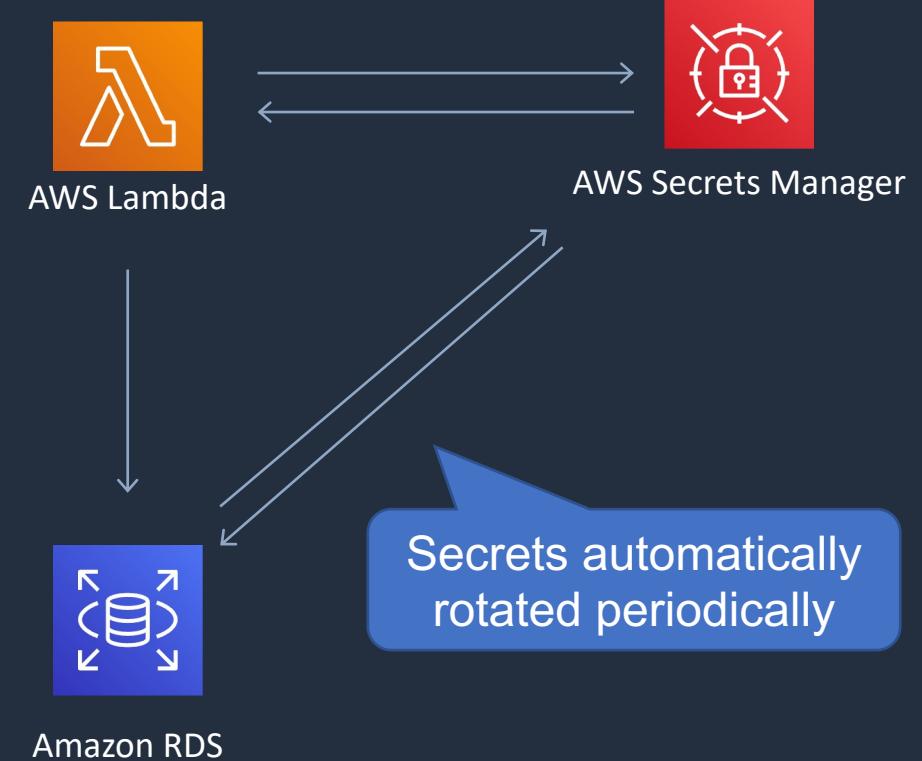
- Parameter Store provides secure, hierarchical storage for configuration data and secrets
- Highly scalable, available, and durable
- Store data such as passwords, database strings, and license codes as parameter values
- Store values as plaintext (unencrypted data) or ciphertext (encrypted data)
- Reference values by using the unique name that you specified when you created the parameter
- No native rotation of keys (difference with AWS Secrets Manager which does it automatically)





# AWS Secrets Manager

- Stores and rotate secrets safely without the need for code deployments
- Secrets Manager offers automatic rotation of credentials (built-in) for:
  - Amazon RDS (MySQL, PostgreSQL, and Amazon Aurora)
  - Amazon Redshift
  - Amazon DocumentDB
- For other services you can write your own AWS Lambda function for automatic rotation





# AWS Secrets Manager vs SSM Parameter Store

---

	Secrets Manager	SSM Parameter Store
<b>Automatic Key Rotation</b>	Yes, built-in for some services, use Lambda for others	No native key rotation; can use custom Lambda
<b>Key/Value Type</b>	String or Binary (encrypted)	String, StringList, SecureString (encrypted)
<b>Hierarchical Keys</b>	No	Yes
<b>Price</b>	Charges apply per secret	Free for standard, charges for advanced

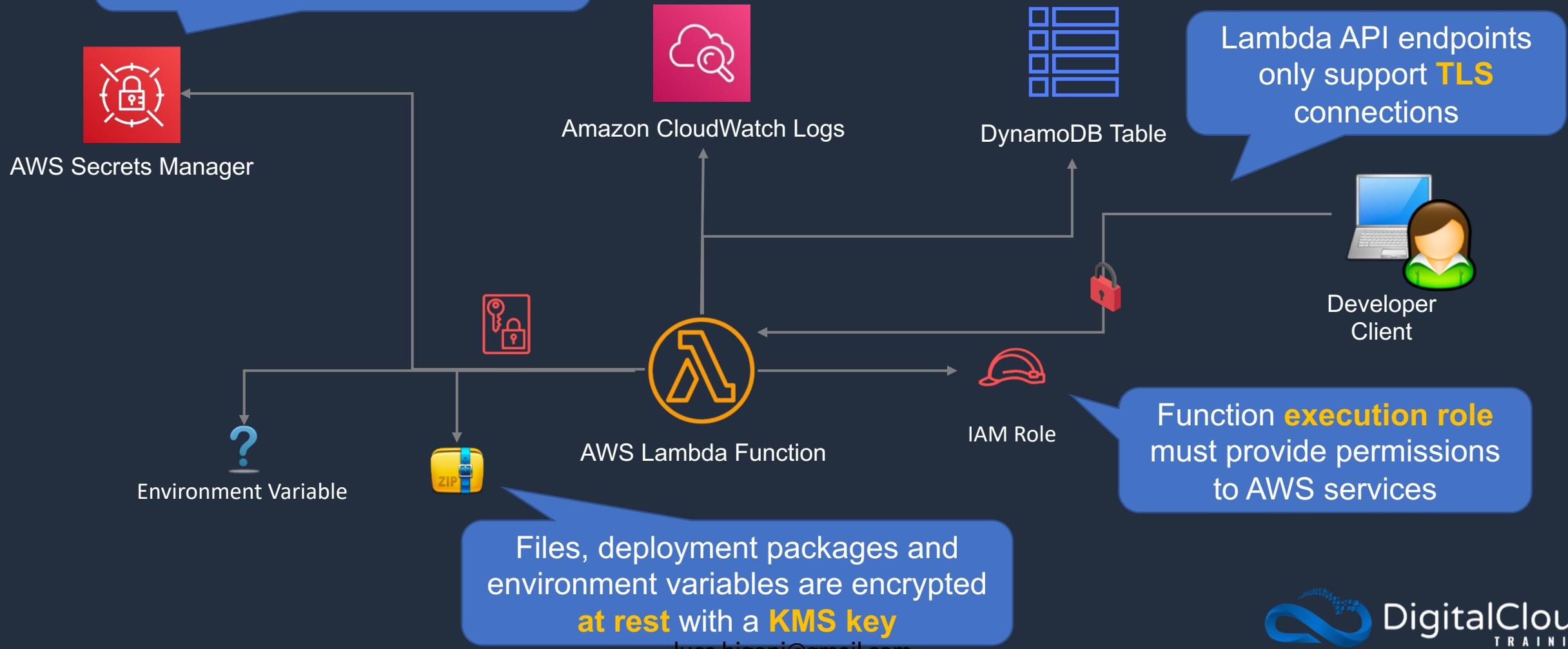
# Security for Lambda Functions





# Security for AWS Lambda Functions

AWS recommend to use **Secrets Manager** instead of environment variables





# AWS Signer

---

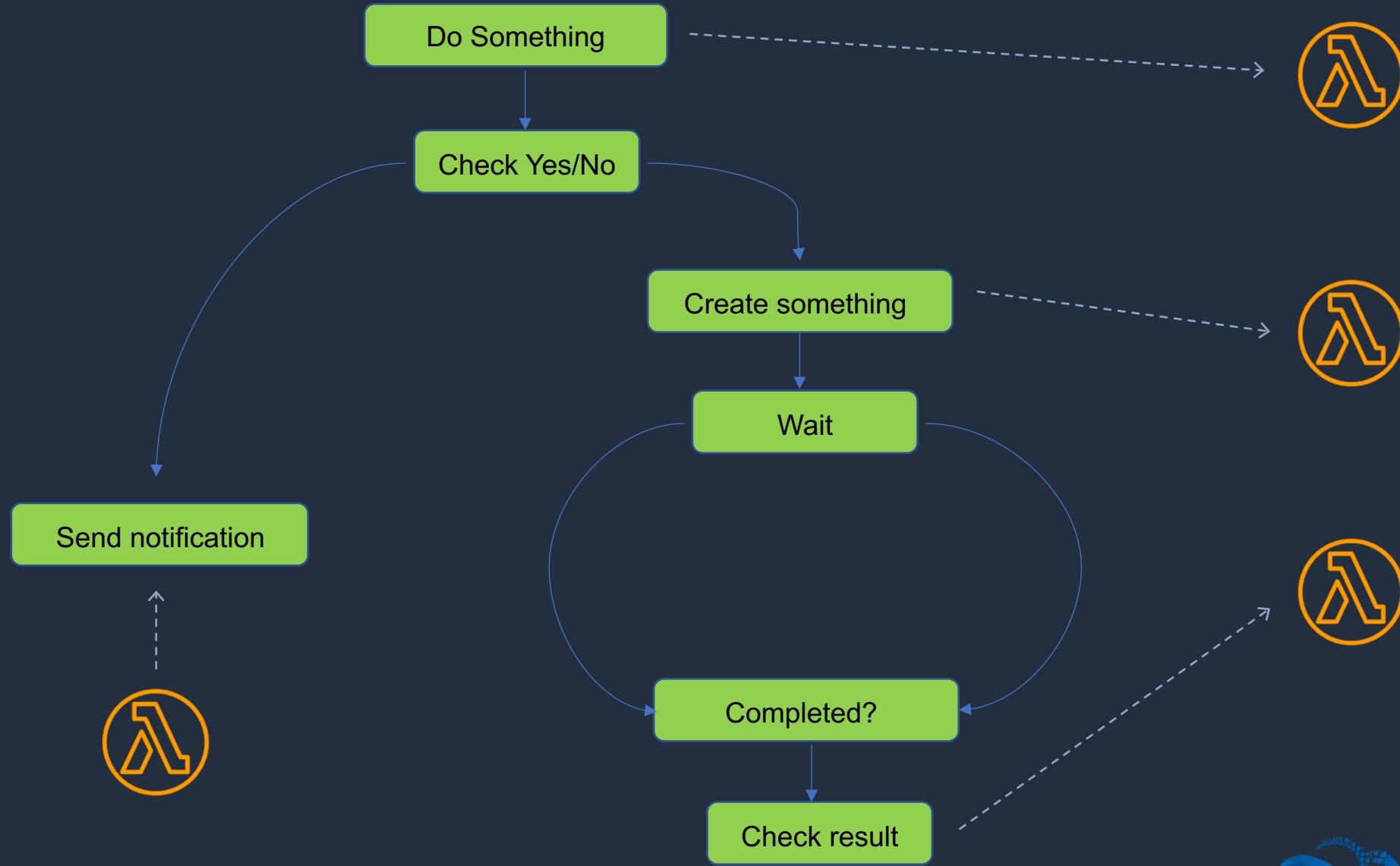
- AWS Signer is a fully managed code-signing service
- Used to ensure the trust and integrity of code
- Code is validated against a digital signature
- With Lambda you can ensure only trusted code runs in Lambda functions
- Signer is used to create digitally signed packages for deployment
- IAM policies can enforce that functions can be created only if they have code signing enabled
- If a developer leaves you can revoke all versions of the signing profile so the code cannot run

# AWS Step Functions





# AWS Step Functions





# AWS Step Functions

---

---

- AWS Step Functions is used to build distributed applications as a series of steps in a visual workflow
- You can quickly build and run state machines to execute the steps of your application

## How it works:

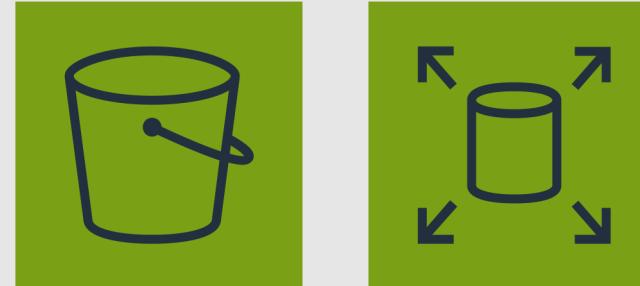
1. Define the steps of your workflow in the **JSON-based Amazon States Language**  
The visual console automatically graphs each step in the order of execution
2. Start an execution to visualize and verify the steps of your application are operating as intended. The console highlights the real-time status of each step and provides a detailed history of every execution
3. AWS Step Functions **operates and scales** the steps of your **application** and **underlying compute** for you to help ensure your application executes reliably under increasing demand



## Security related use case:

- Use CloudFormation to create infrastructure for a forensic environment
- Use Step Functions to orchestrate the processes of forensic analysis (with Lambda)

# AWS Data Lifecycle Management Features





# What's a Data Lifecycle?

---

---

How an organization  
handles stored data  
as it ages.





# Who Wants to Store Forever?

---

*Keep data only as long as it is needed.*



Reduce **storage costs**



Avoid **compliance violations**



# Who Wants to Store Forever?

---

---

- Retention planning the customer's responsibility
  - Know compliance requirements
  - Identify and categorize data





- Object retention periods
- S3 Lifecycle Rules
- Amazon Data Lifecycle Manager



# S3 Lifecycle Rules

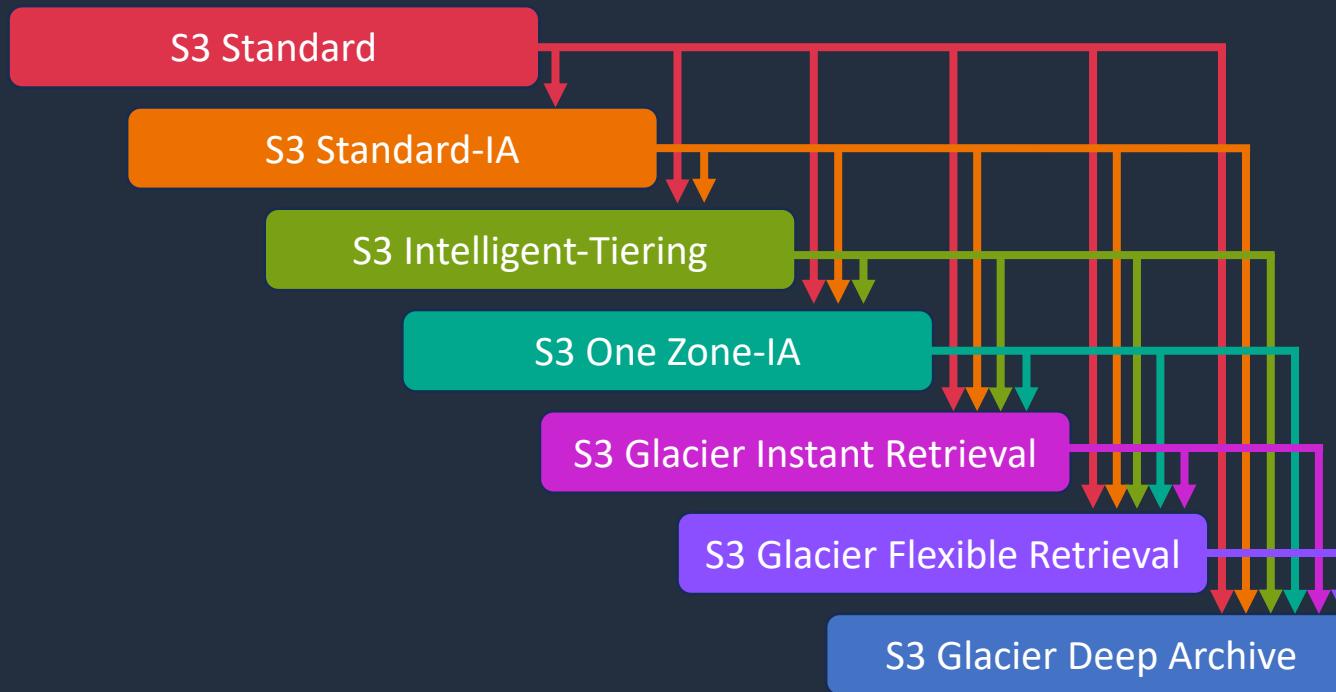
- Apply actions to bucket objects after X time-period
  - Transition** – change object's storage class
  - Expiration** – permanently delete objects
- Configured at bucket level
- Filter affected objects by:
  - Prefix
  - Tag
  - Size





# S3 Lifecycle Rules: Transitions

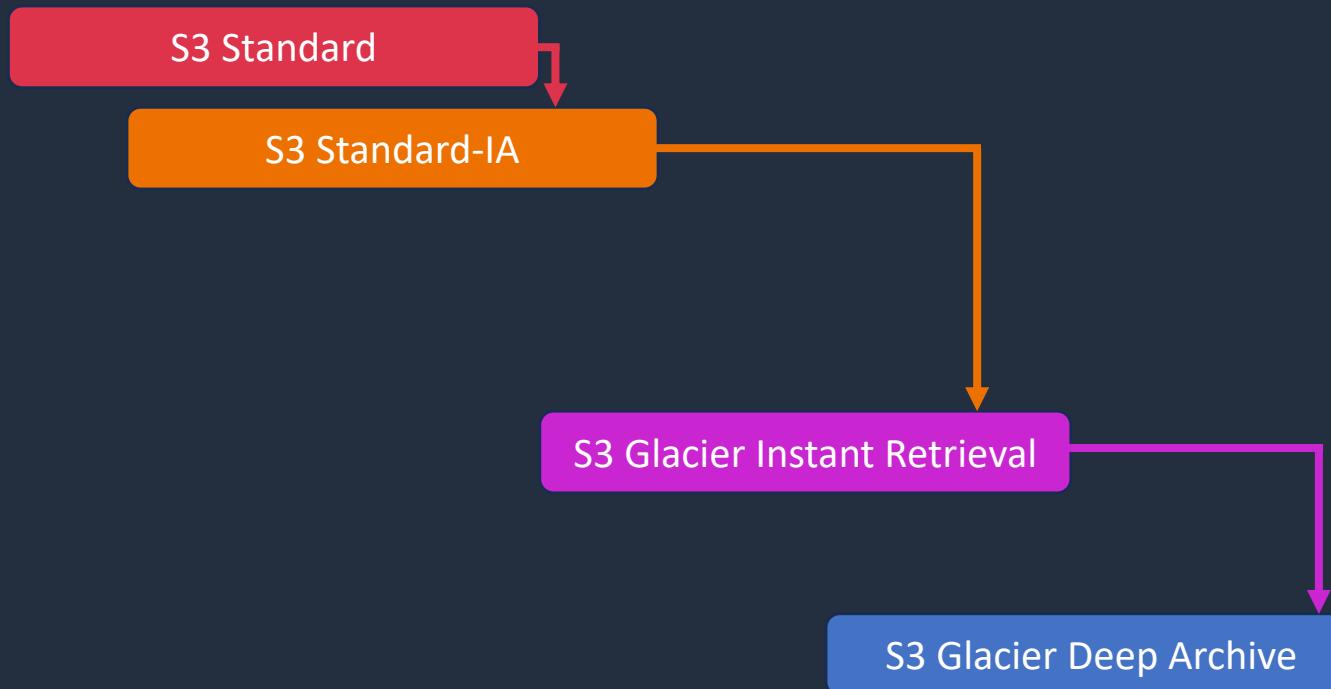
- Can only transition “down”





# S3 Lifecycle Rules: Transitions

- Can only transition “down”



- Transitions may be chained



# S3 Lifecycle Rules

---

- Current & previous object versions are handled independently
- Specify number of newer, non-current versions to retain





# Intelligent-Tiering

---

- Moves infrequently used resources to more cost-efficient storage class
- Supported by S3 and EFS
- Does not handle expiration





# Amazon Data Lifecycle Manager

---



Create & delete snapshots of

- EBS volumes
- EBS-backed AMIs



# Amazon Data Lifecycle Manager: Policies

---



## EBS snapshot & EBS-backed AMI

- Protects resources with matching tags
- Up to three different schedules
  - Snapshot frequency
  - Retention method
  - Migration to archive tier
  - Cross-region copy



# Amazon Data Lifecycle Manager: Policies

---



- EBS policies support cross-account copy
- AMI policies support source reboot and AMI depreciation



# Amazon Data Lifecycle Manager: Policies

---



## Cross-Account copy

- Created in target AWS account
- Triggered by snapshot sharing from source account
  - Snapshot description must match regex defined in policy



## Recap

---

---

Use DLM to reduce storage costs and meet compliance requirements.

---

Many AWS objects can manage their own data expiration.

---

S3 Lifecycle Rules and EC2 Data Lifecycle manager can delete objects.

---

Intelligent-Tiering cannot delete objects.

# AWS Data Integrity Features





# No, It's NOT Encryption

---

---

Prevent *accidental* or *malicious*  
**modification** or **deletion** of data

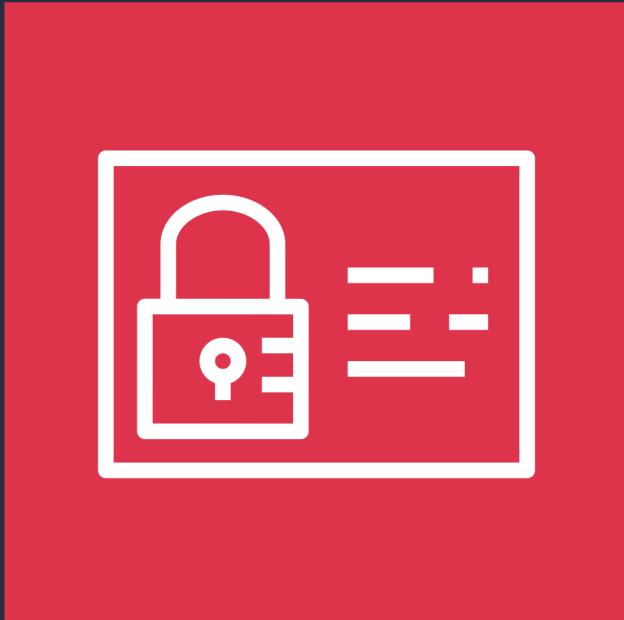
**WORM** – Write Once Read Many



# Always Start with IAM

---

---



- Grant least privilege
- Implement strong authentication
- Secure root account access



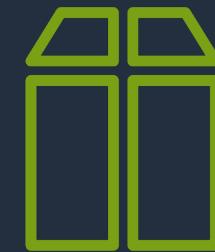
# AWS Data Integrity Options

---

- S3 Object Lock



- S3 Glacier Vault Lock



- AWS Backup Vault Lock





# S3 Object Lock

---

- Prevents **deletion** of objects for configured retention period
- Option for new buckets
- Automatically enables versioning
  - New versions may be created
  - Older versions may not be deleted
- New objects inherit bucket settings





## Governance

- Users may be authorized to delete objects
- *BypassGovernanceRetention*



## Compliance

- No one may delete object until retention period expires

*...not even account root!*



# S3 Glacier Vault Lock

---



- Resource policy that controls when vault objects may be deleted
- Denies *DeleteArchive* unless conditions are met.
- Used in addition to IAM or vault access policies



# S3 Glacier Vault Lock

---



- “In-progress” state for 24 hours.
  - Test lock policy
  - Policy deleted if not confirmed.
- Vault lock policy cannot be modified or deleted after confirmation.



# AWS Backup Vault Lock

---

- Prevents backups in vault from being deleted
- Prevents backup vault from being deleted or modified





## Governance

- Users may be authorized to delete objects or vault

## Compliance

- After start date, no one can delete backups or delete/manage vault...*forever*
- Vault may only be deleted by terminating the AWS account.





# Legal Holds

---



- Manually applied locks
- Applied or removed at any time
- Independent of other lock retention periods



# Legal Holds

---

- S3 Object lock legal hold
  - Requires S3 Object Lock
  - Object-level setting
- AWS Backup legal hold
  - Does not require vault lock
  - Prevents backups from being deleted for duration of lock
  - Applied to backups of resources or to all backups in a vault





## Recap

---

---

Always follow best IAM practices.

---

Governance mode allows authorized users to delete objects.

---

Compliance mode prevents all object deletions.

---

Legal Holds are manually applied and removed.

# SECTION 9

## Logging, Monitoring, and Auditing

# Amazon CloudWatch & EventBridge





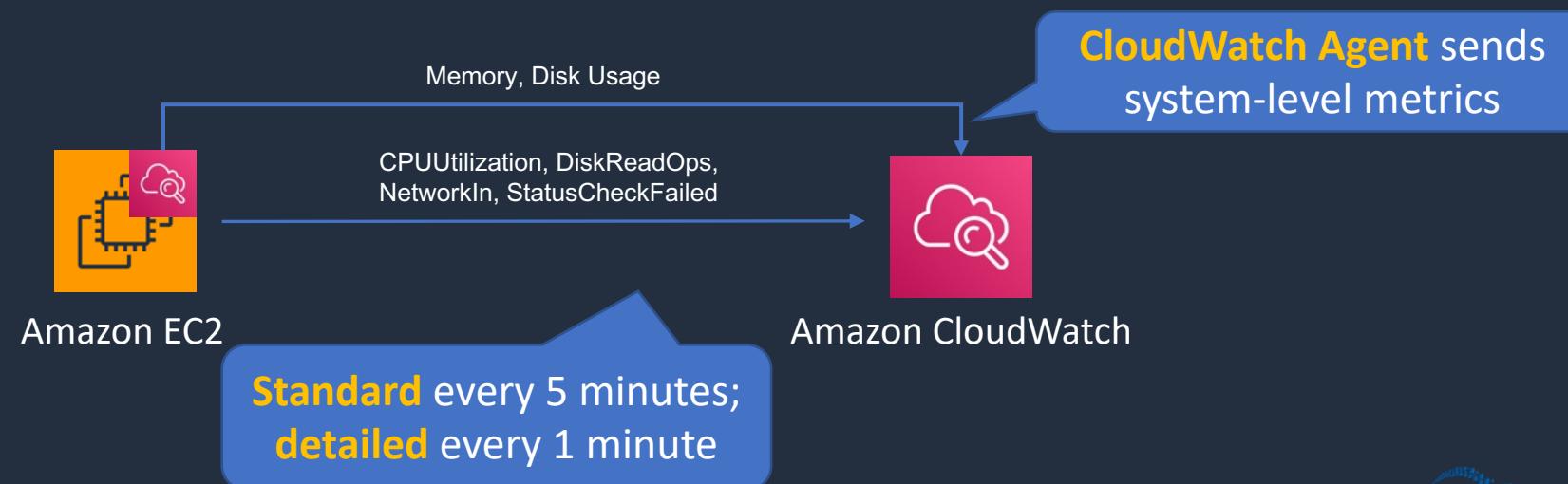
# Amazon CloudWatch

- **CloudWatch Metrics** – services send time-ordered data points to CloudWatch
- **CloudWatch Alarms** – monitor metrics and initiate actions
- **CloudWatch Logs** – centralized collection of system and application logs
- **CloudWatch Events** – stream of system events describing changes to AWS resources and can trigger actions



# Amazon CloudWatch Metrics

- Metrics are sent to CloudWatch for many AWS services
- EC2 metrics are sent every **5 minutes** by default (free)
- Detailed EC2 monitoring sends every **1 minute** (chargeable)
- Unified CloudWatch Agent sends system-level metrics for EC2 and on-premises servers
- System-level metrics include memory and disk usage

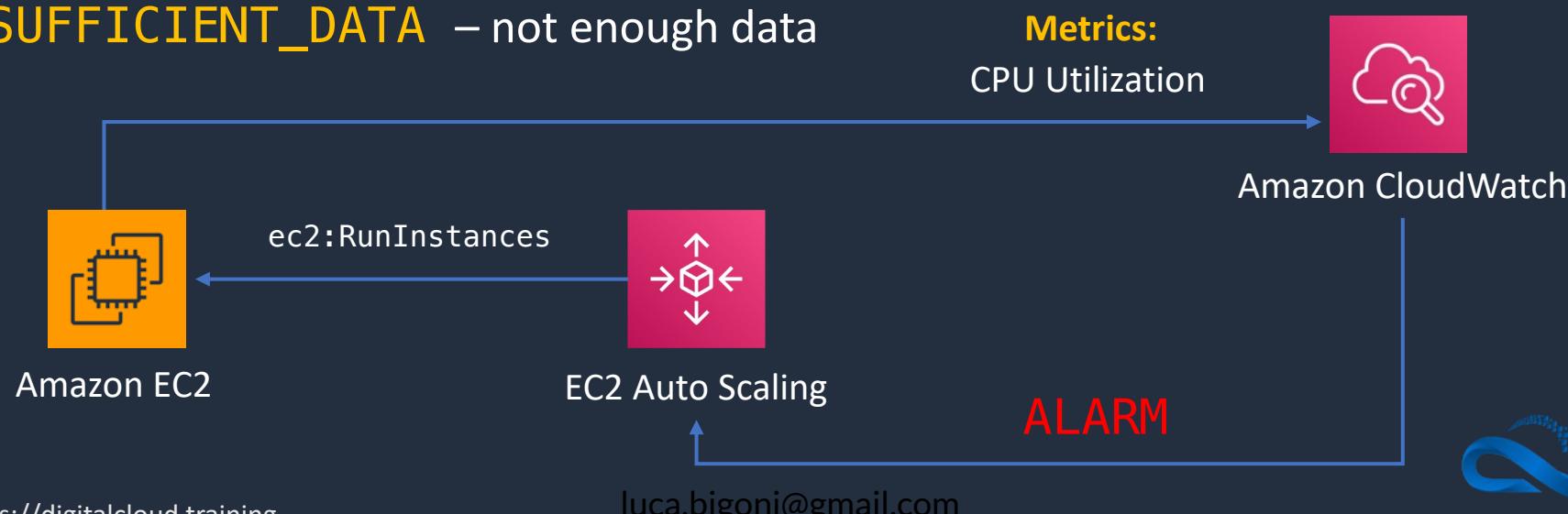




# Amazon CloudWatch Alarms

Two types of alarms

- **Metric alarm** – performs one or more actions based on a single metric
- **Composite alarm** – uses a rule expression and takes into account multiple alarms
- Metric alarm states:
  - **OK** – Metric is within a threshold
  - **ALARM** – Metric is outside a threshold
  - **INSUFFICIENT\_DATA** – not enough data

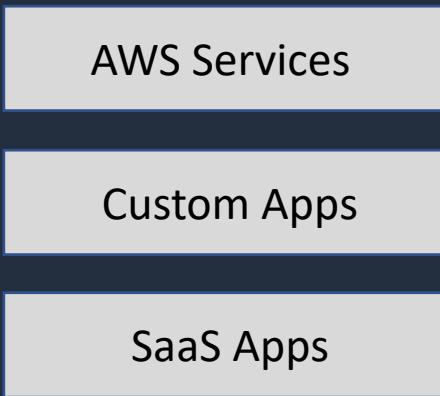




# Amazon CloudWatch Events / EventBridge

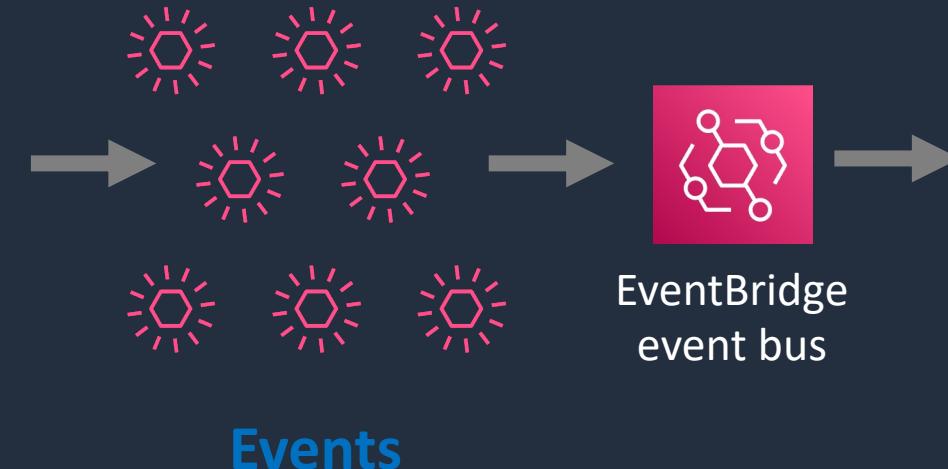
---

## Event Sources

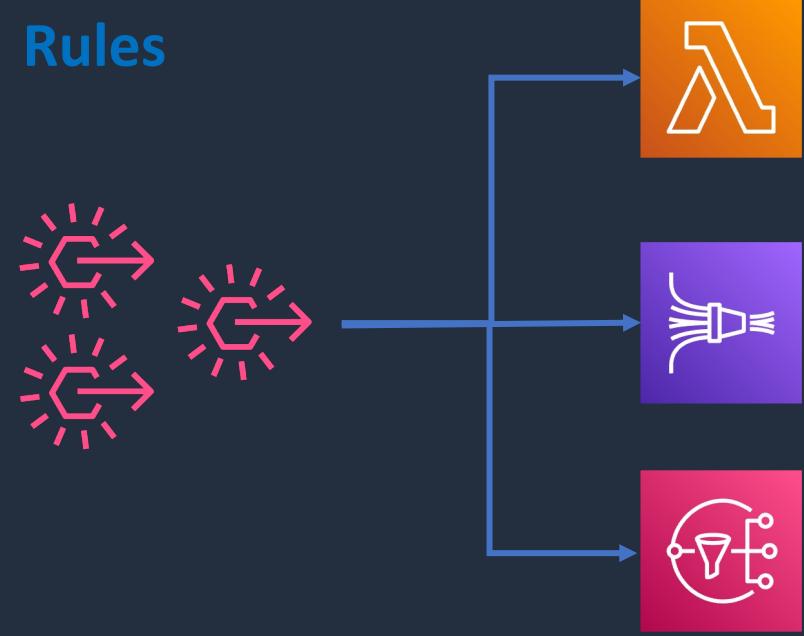


EventBridge used to be known as CloudWatch

## Events



## Rules

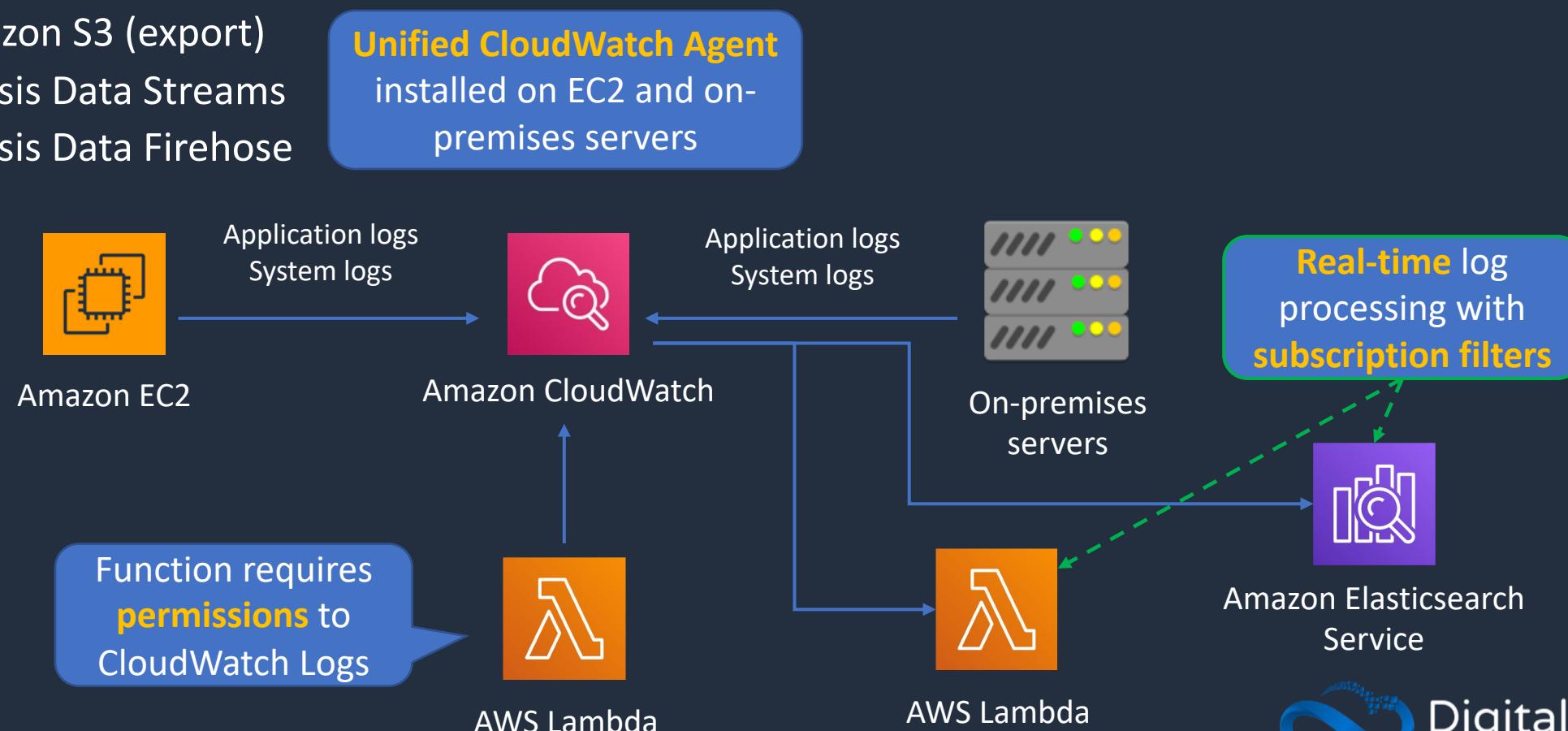


## Targets



# Amazon CloudWatch Logs

- Gather application and system logs in CloudWatch
- Defined expiration policies and KMS encryption
- Send to:
  - Amazon S3 (export)
  - Kinesis Data Streams
  - Kinesis Data Firehose





# The Unified CloudWatch Agent

The unified CloudWatch agent enables you to do the following:

- Collect internal system-level metrics from Amazon **EC2 instances** across operating systems
- Collect system-level metrics from **on-premises servers**
- Retrieve custom metrics from your applications or services using the StatsD and collectd protocols
- Collect logs from Amazon EC2 instances and on-premises servers (Windows / Linux)



# The Unified CloudWatch Agent

- Agent must be installed on the server
- Can be installed on:
  - Amazon EC2 instances
  - On-premises servers
  - Linux, Windows Server, or macOS

# Create a Custom Metric



# Configure Logging for Lambda



# Logging for Other AWS Services





# VPC Flow Logs vs ELB Access Logs

## VPC Flow Log

version	account-id	interface-id	srcaddr	dstaddr	srcport	dstport	protocol	packets	bytes	start	end	action	log-status
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	52933	22	6	1	401599...	1599...	401599...	ACCEPT	OK
2	55112233445	eni-0f5...	10.0.1.15	11.200.185.200	22	52933	6	1	401599...	1599...	401599...	ACCEPT	OK
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...	1599...	441599...	REJECT	OK
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...	1599...	441599...	REJECT	OK

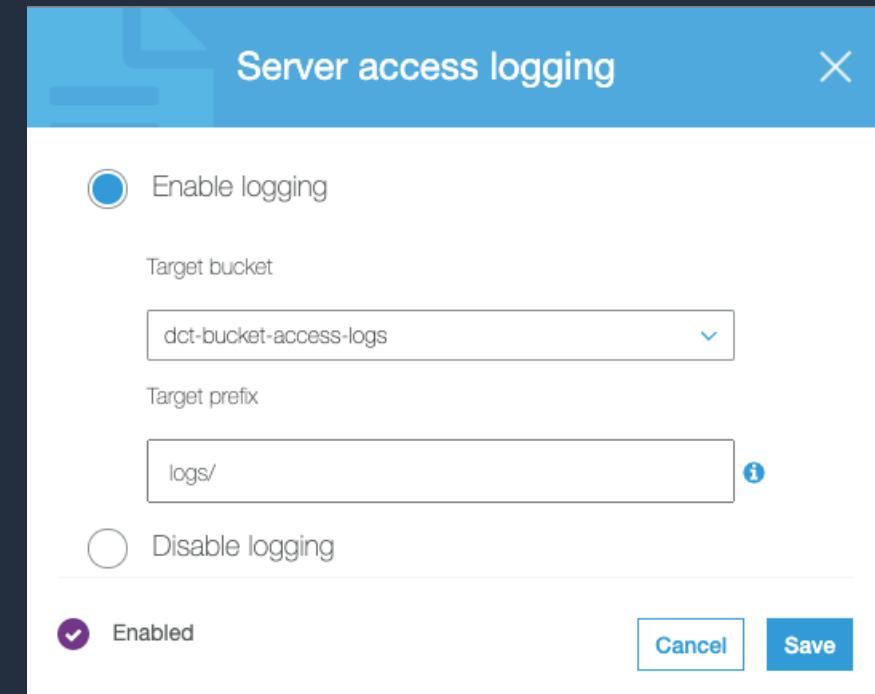
## ELB Access Log

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" --
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "--" "--"
0 2018-07-02T22:22:48.364000Z "forward" "--" "--" 10.0.0.1:80 200 "--" "--"
```



# Server Access Logging

- Provides detailed records for the requests that are made to a bucket
- Details include the requester, bucket name, request time, request action, response status, and error code (if applicable)
- Disabled by default
- Only pay for the storage space used
- Must configure a separate bucket as the destination (can specify a prefix)
- Must grant write permissions to the Amazon S3 Log Delivery group on destination bucket



# AWS CloudTrail





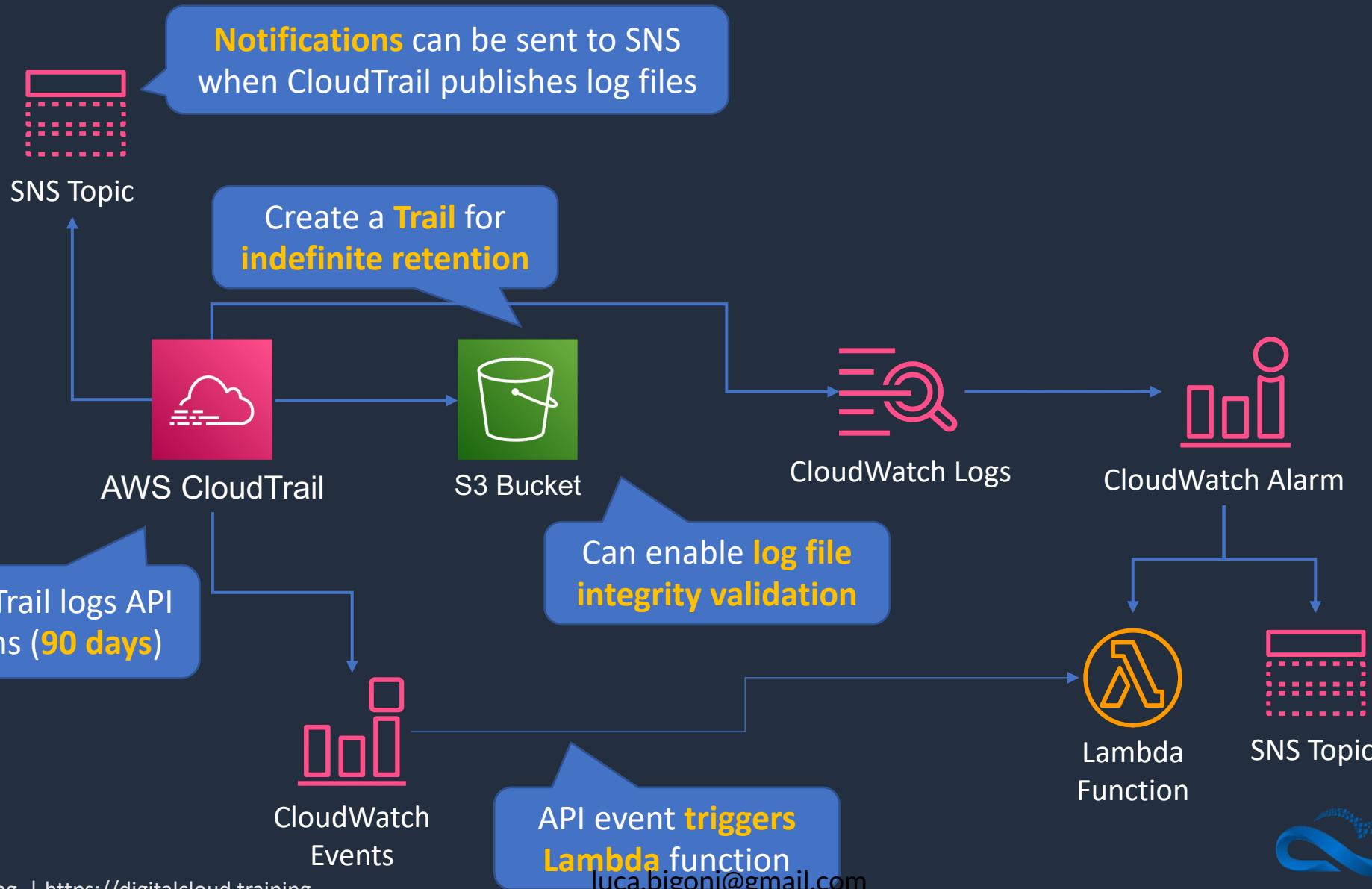
# AWS CloudTrail

---

- CloudTrail logs **API activity** for auditing
- By default, management events are logged and retained for 90 days
- A **CloudTrail Trail** logs any events to S3 for indefinite retention
- Trail can be within Region or all Regions
- CloudWatch Events can triggered based on API calls in CloudTrail
- Events can be streamed to CloudWatch Logs



# AWS CloudTrail





# CloudTrail – Types of Events

---

- **Management events** provide information about management operations that are performed on resources in your AWS account
- **Data events** provide information about the resource operations performed on or in a resource
- **Insights events** identify and respond to unusual activity associated with write API calls by continuously analyzing CloudTrail management events

# Create EventBridge rule for CloudTrail API calls



# SECTION 10

## Directory Services and Federation

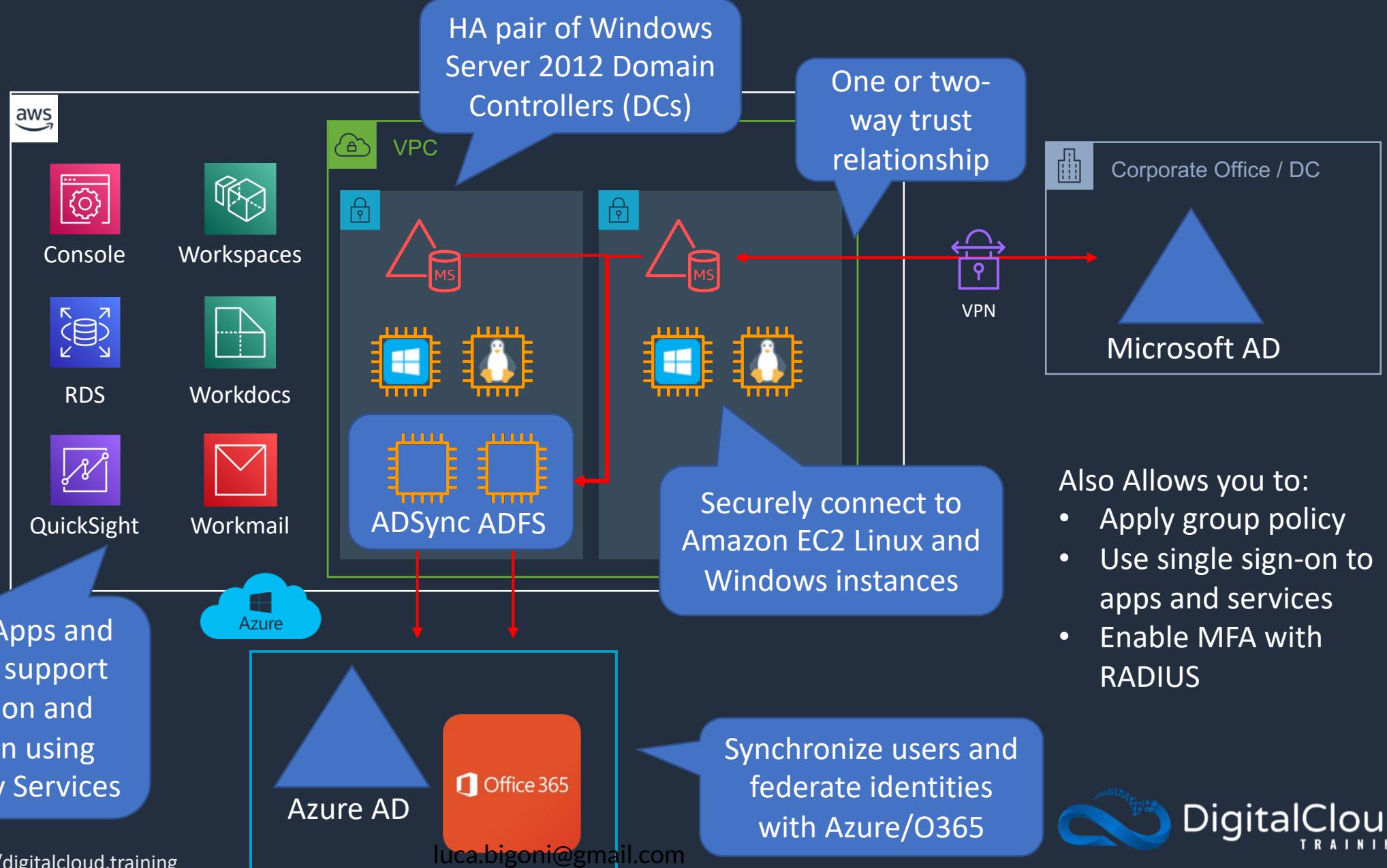
# AWS Directory Services



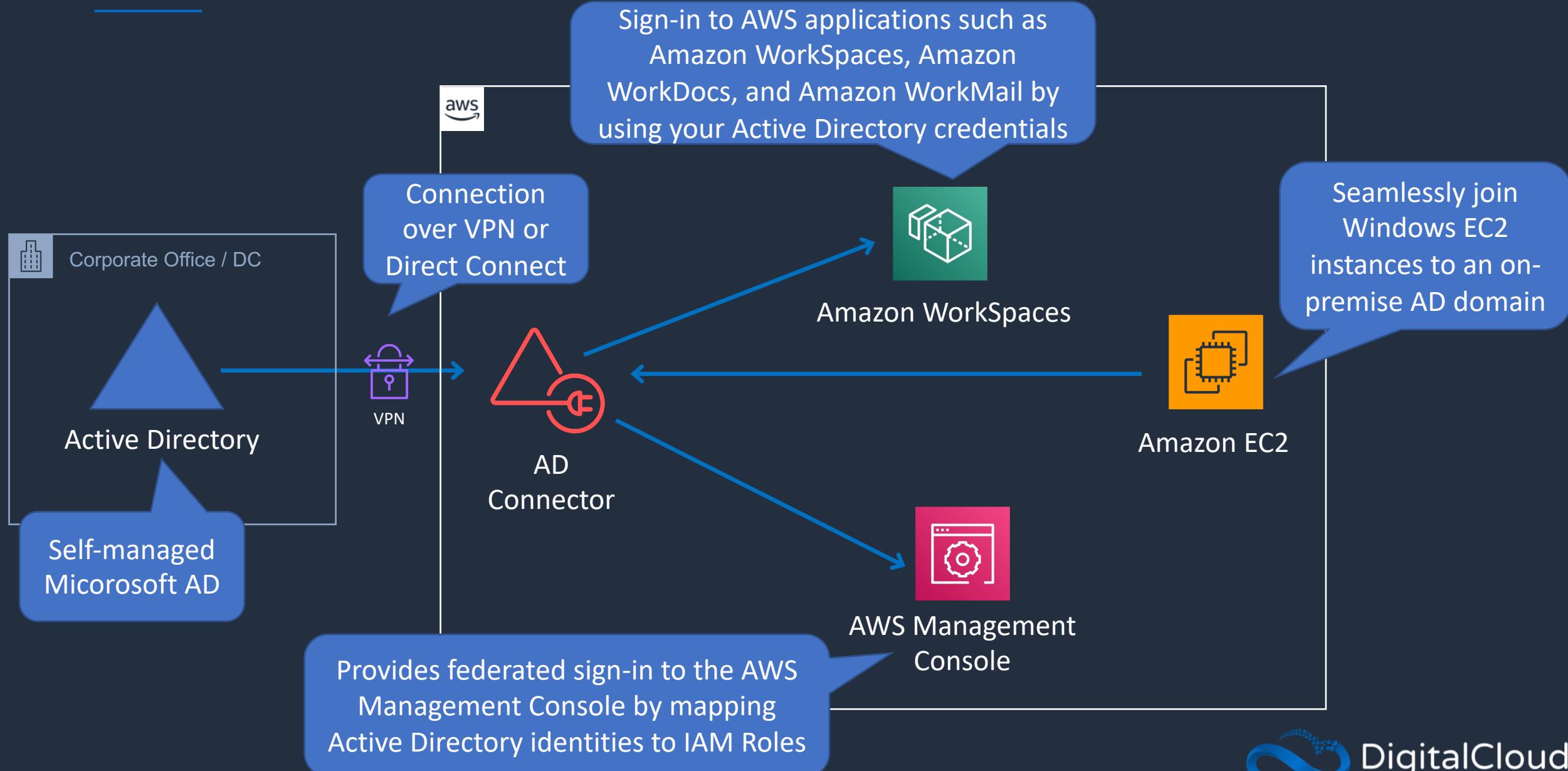
# AWS Managed Microsoft AD



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



# AD Connector



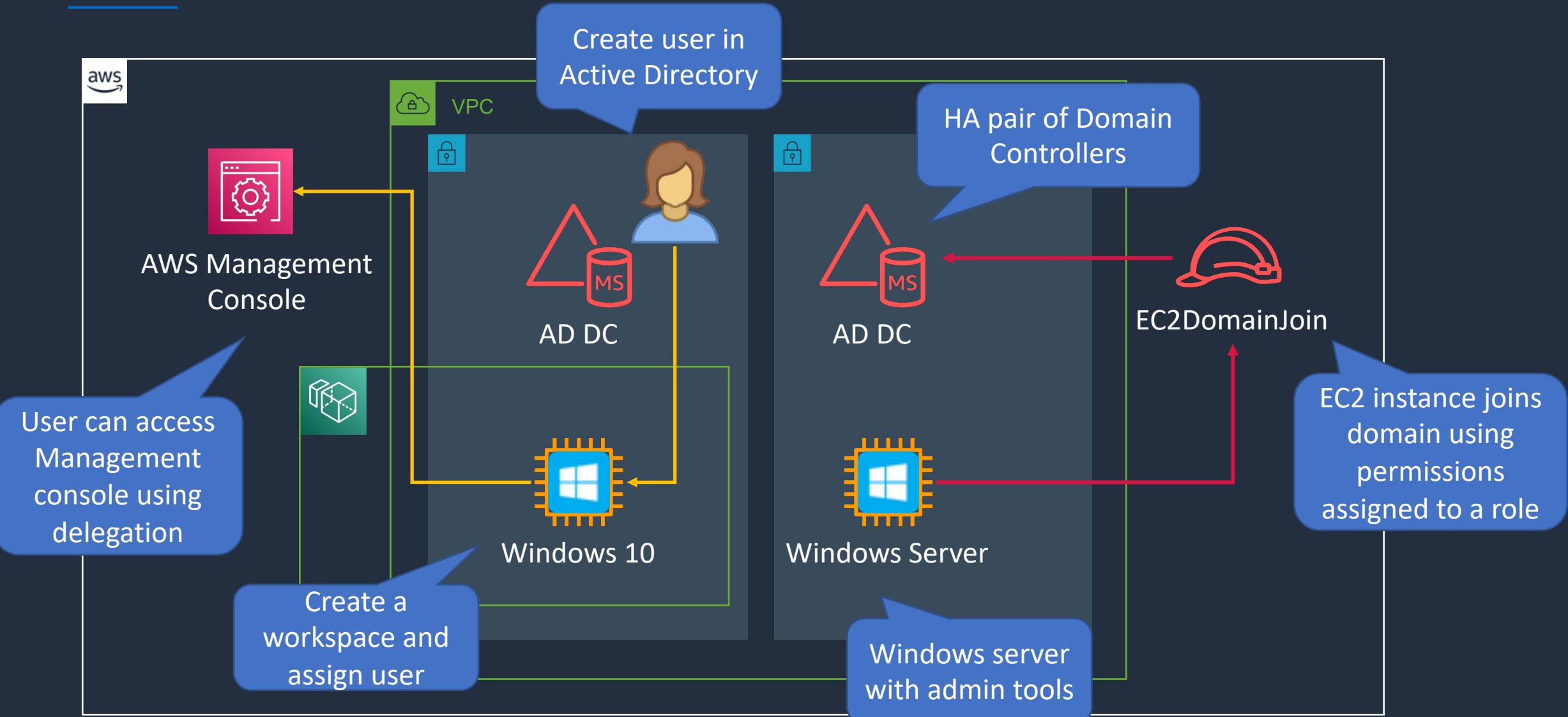
# Create AWS Managed Microsoft AD



luca.bigoni@gmail.com



# AWS Managed Microsoft AD



# 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

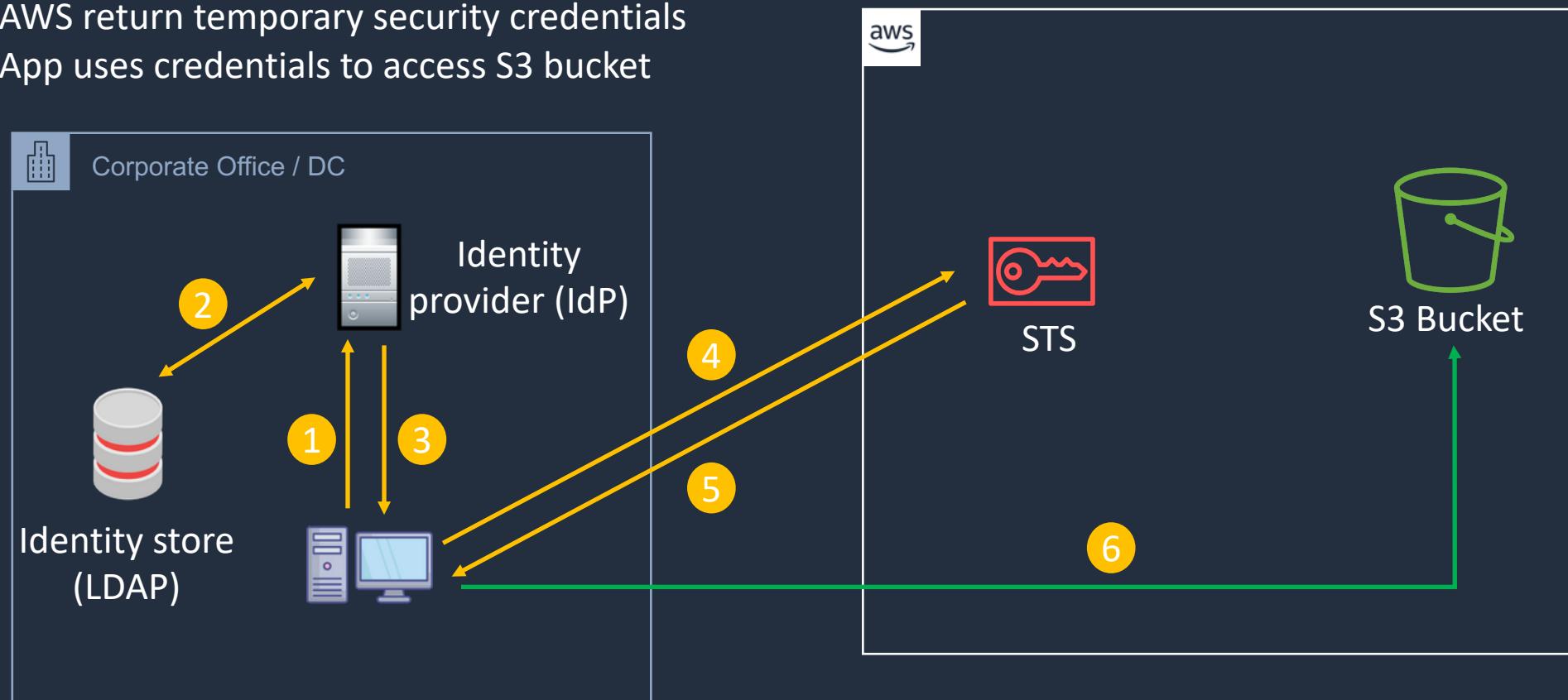
# IAM Identity Federation





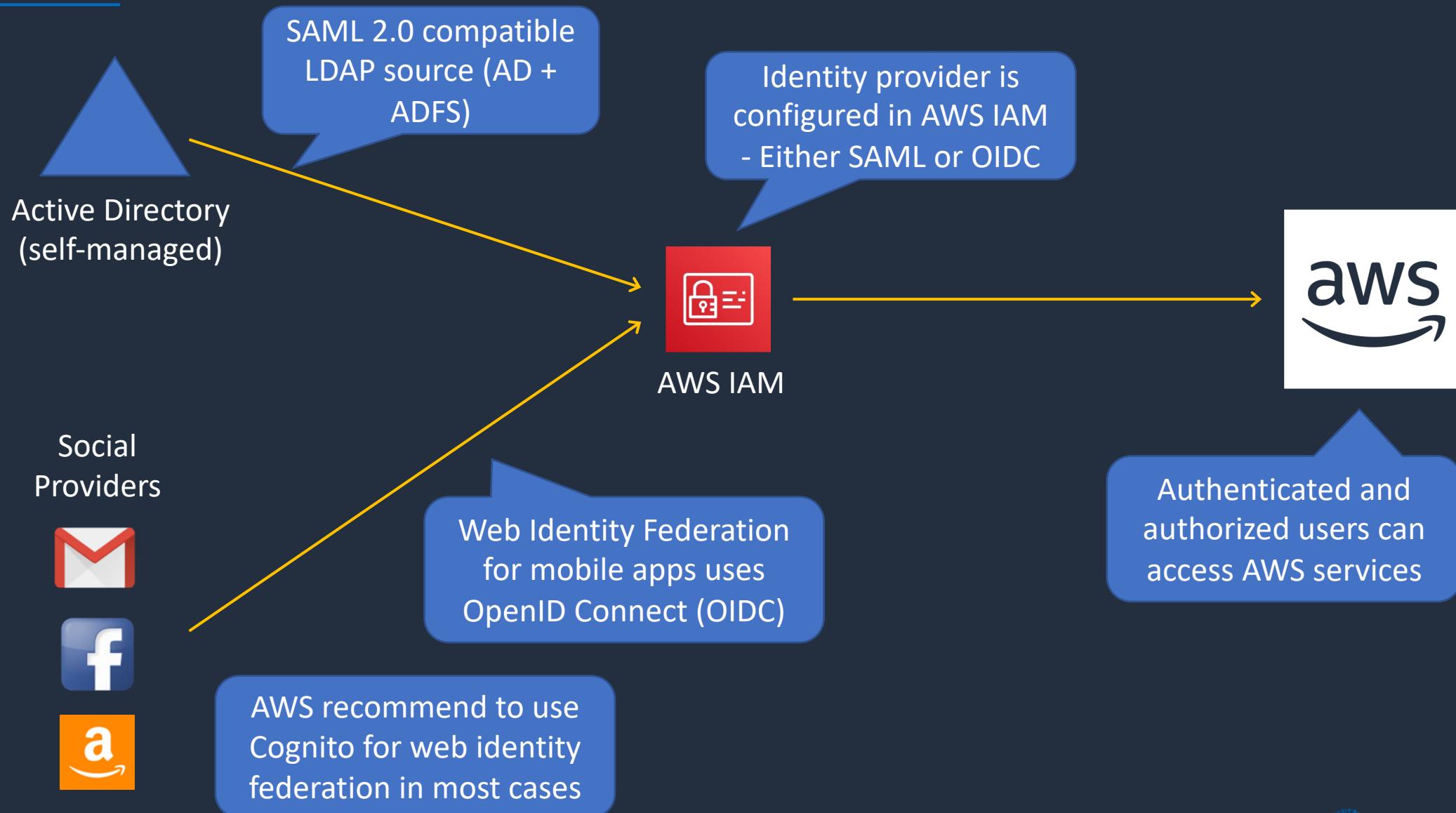
# Identity Federation

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

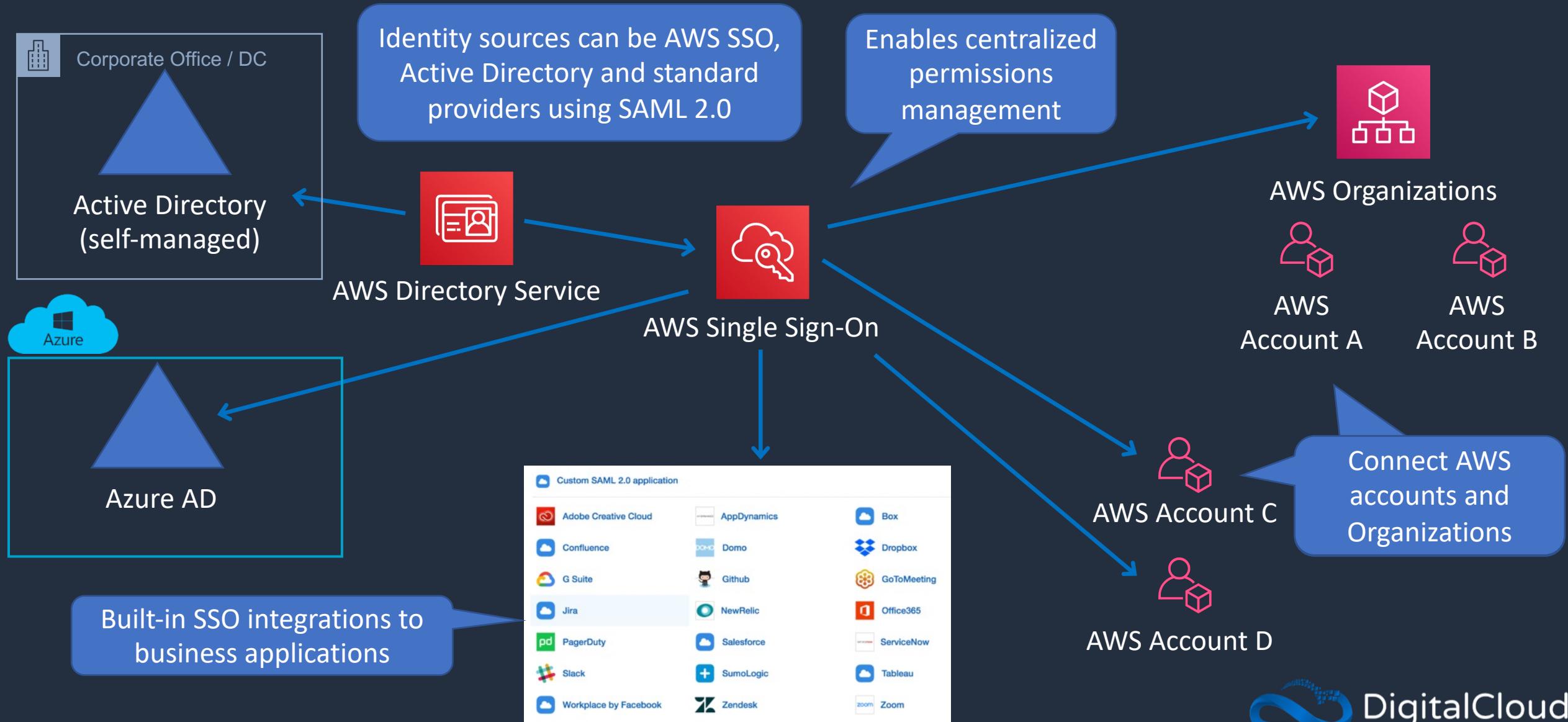


# AWS Single Sign-on (SSO)





# AWS Single Sign-on (SSO)

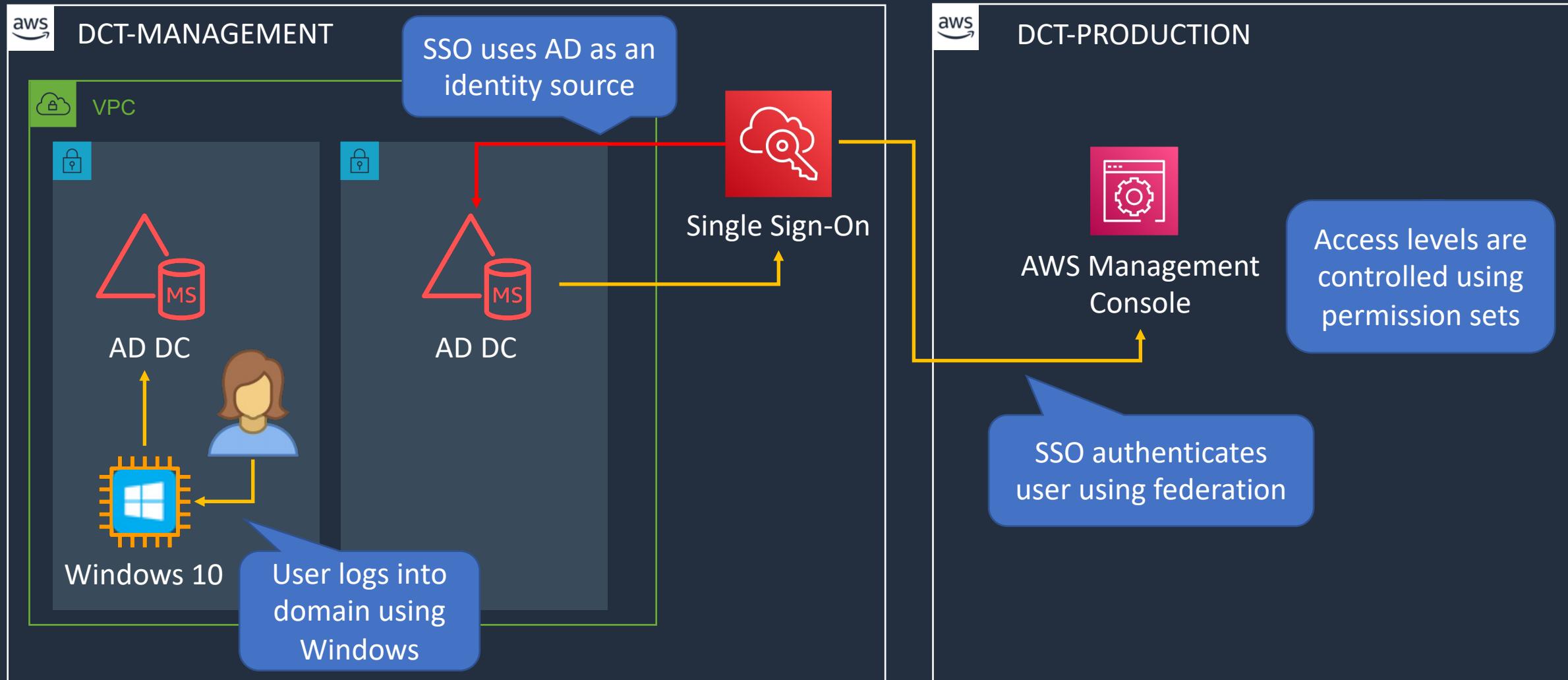


# Configure AWS SSO with AWS Managed AD

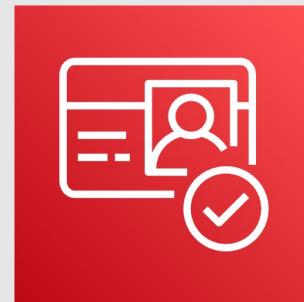




# AWS Managed Microsoft AD



# Amazon Cognito

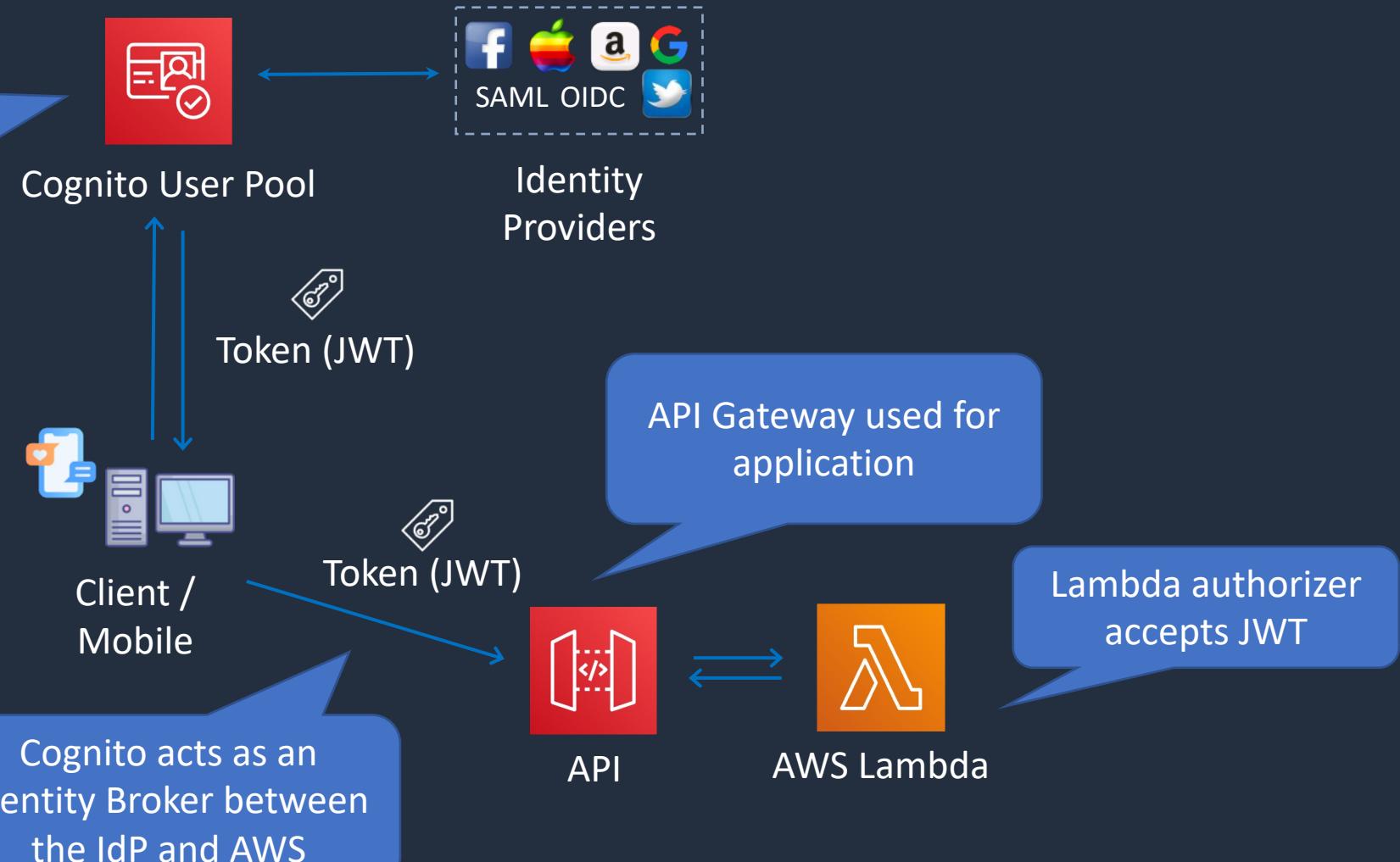




# Cognito User Pools

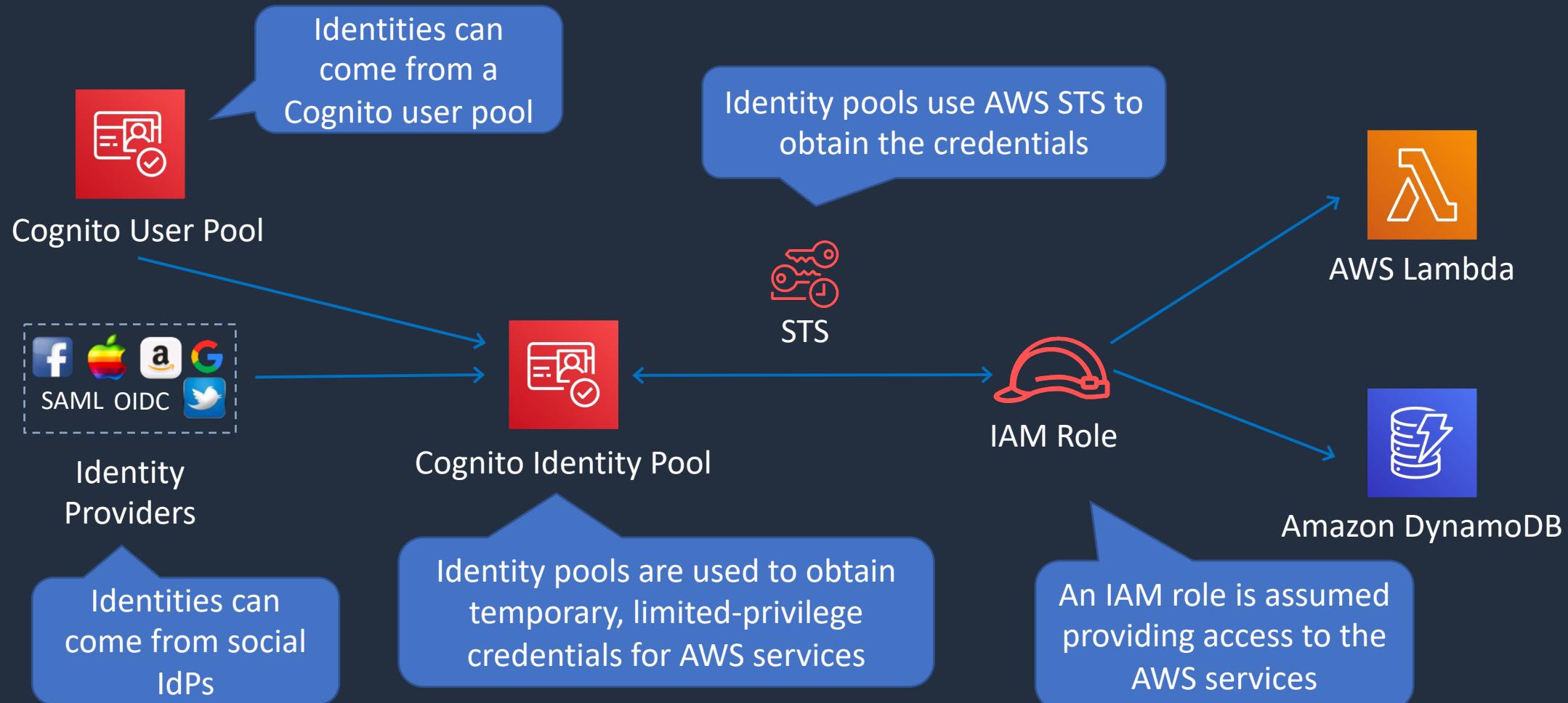
A User Pool is a directory for managing sign-in and sign-up for mobile applications

Users can also sign in using social IdPs



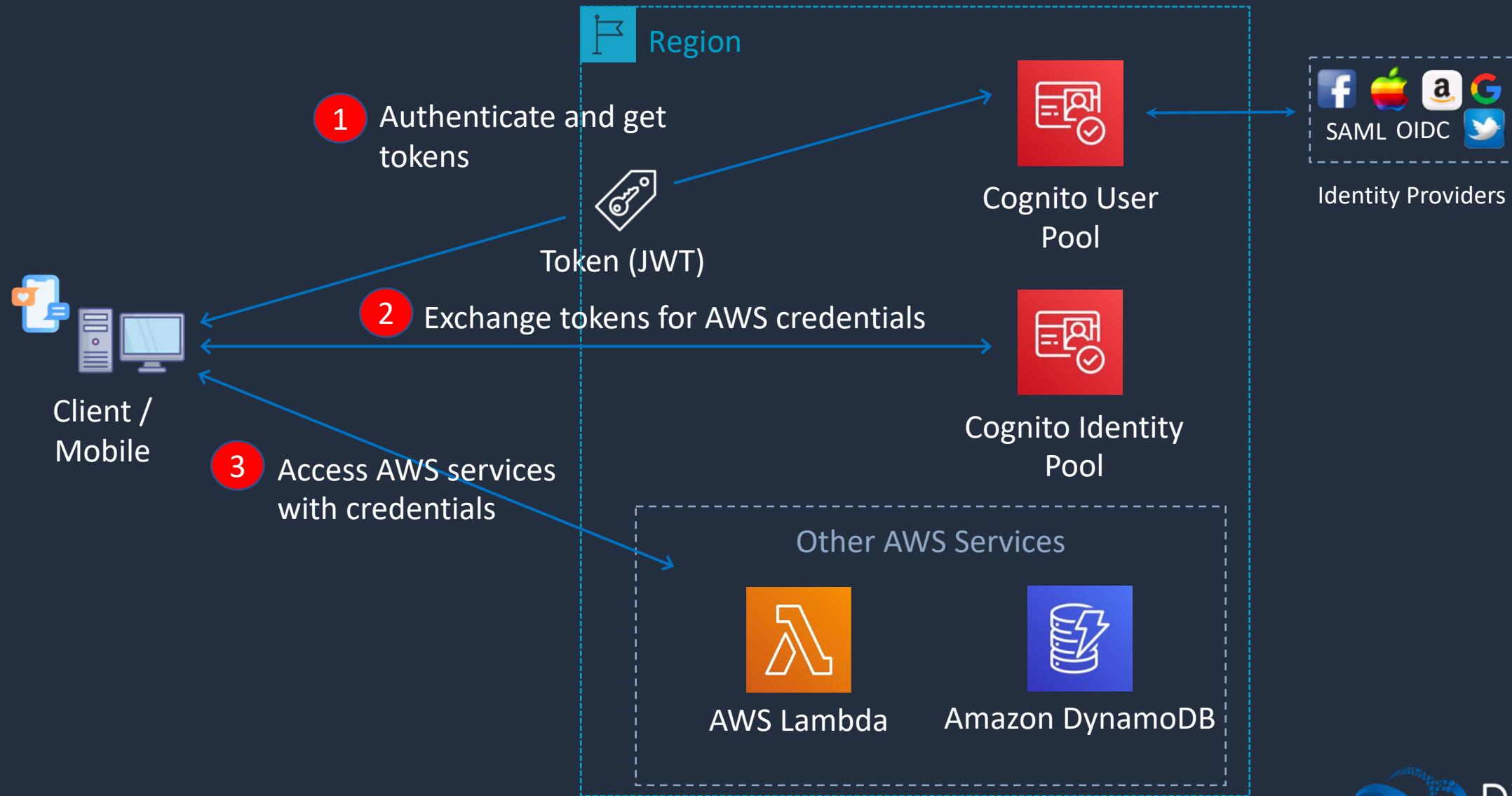


# Cognito Identity Pool





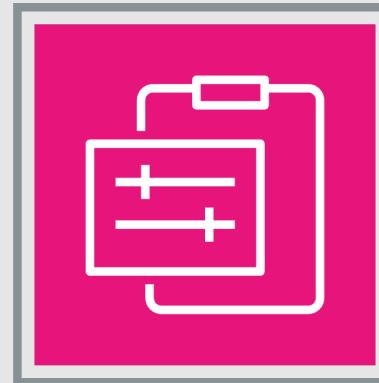
# User Pools + Identity Pools



# SECTION 11

## Data Analysis and Incident Response

# AWS Incident Response Guide Overview



luca.bigoni@gmail.com



# Overview

---

---

# Best practices for incident response planning in the cloud

---

AWS Security Incident  
Response Guide  
AWS Technical Guide





# Further Reading



- AWS Cloud Adoption Framework
- AWS Well Architected Framework
- NIST Computer Security Incident Handling Guide



# Cloud Differences

---

- Additional Incident Domain
  - Infrastructure
  - Application
  - Cloud Service
- Security is a shared responsibility
- API provisioned infrastructure
- Dynamic Environment
- Leverage IaC automation





# Aspects of Incidence Response Preparation



- People
  - Identify key personnel and responsibilities
  - Familiarity with environment, tools, and policies
  - Knowledge of AWS incident support options



# Aspects of Incidence Response- Preparation

---

---



- People
- Process
  - Document environment architecture
  - Develop response plans and playbooks
  - Perform regular simulations



# Aspects of Incidence Response Preparation



- People
- Process
- Technologies
  - Optimize AWS account structure
  - Devise a tagging strategy
  - Maintain threat landscape awareness
  - Establish logging and notification
  - Implement forensics support



# Aspects of Incidence Response Operations

---

---

- Detection
- Analysis
- Containment
- Eradication
- Recovery





# Aspects of Incidence Response - Post-Incident



- Review the incident
  - What was learned?
  - What worked well?
  - What could have gone better?
- Define success metrics
- Highlight Indicators of Compromise
- Continue learning



## Recap

---

---

Prepare personnel, processes, and environment.

---

Leverage cloud capabilities to streamline incident response.

---

Review incidents to identify successes and weaknesses.

---

Continue developing personnel, processes, and environment.

# Security Management and Support





# AWS Security Hub

- Provides a comprehensive view of security alerts and security posture **across AWS accounts**
- Aggregates, organizes, and prioritizes security alerts, or findings, from multiple AWS services
- Continuously monitors your environment using automated security checks
- Configure security standards to validate against
  - AWS Foundational Security Best Practices v1.0.0
  - CIS AWS Foundations Benchmark v1.2.0
  - PCI DSS v3.2.1



# AWS Security Bulletins

- Security and privacy events affecting AWS services are published (also has an RSS feed)

▼ Content Type

Important  
 Informational

▼ Year

2021  
 2020  
 2019  
 2018  
 2017  
 2016  
 2015  
 2014

<a href="#">Sudo Security Issue (CVE-2021-3156)</a> AWS-2021-001, 01/27/2021
<a href="#">Xen Security Advisory (XSA-286)</a> AWS-2020-005, 10/23/2020
<a href="#">Container Networking Security Issue (CVE-2020-8558)</a> AWS-2020-002v2, 07/09/2020
<a href="#">Minimum Version of TLS 1.2 Required for FIPS Endpoints by March 31, 2021</a> AWS-2020-001, 03/31/2020
<a href="#">Kubernetes Security Issue (CVE-2019-11249)</a> AWS-2019-007, 08/15/2019
<a href="#">Kubernetes Security Issue (CVE-2019-11246)</a> AWS-2019-006, 07/02/2019
<a href="#">Linux Kernel TCP SACK Denial of Service Issues</a> AWS-2019-005, 06/17/2019



# AWS Trust & Safety Team

- Contact the **AWS Trust & Safety** team if AWS resources are being used for:
  - Spam
  - Port scanning
  - Denial-of-service attacks
  - Intrusion attempts
  - Hosting of objectionable or copyrighted content
  - Distributing malware
- Email address is: [abuse@amazonaws.com](mailto:abuse@amazonaws.com)



**If you receive an abuse report from AWS, do the following:**

- Review the abuse notice to see what content or activity was reported. Logs that implicate abuse are included along with the abuse report, as provided by the reporter
- Reply directly to the abuse report and explain how you're preventing the abusive activity from recurring in the future

# Penetration testing





# Penetration Testing

---

---

- Penetration testing is the practice of testing one's own application's security for vulnerabilities by simulating an attack
- AWS allows penetration testing without prior approval for 8 AWS services



# Penetration Testing

---

---

## Permitted services

- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora
- Amazon API Gateways
- AWS Lambda and Lambda Edge functions
- Amazon Lightsail resources
- Amazon Elastic Beanstalk environments

## Prohibited Activities

- DNS zone walking via Amazon Route 53 Hosted Zones
- Denial of Service (DoS), Distributed Denial of Service (DDoS), Simulated DoS, Simulated DDoS Port flooding
- Protocol flooding
- Request flooding (login request flooding, API request flooding)

# Compliance Services



luca.bigoni@gmail.com



# AWS Artifact

- AWS Artifact provides on-demand access to AWS' security and compliance reports and select online agreements
- Reports available in AWS Artifact include:
  - Service Organization Control (SOC) reports
  - Payment Card Industry (PCI) reports
- Provides certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of **AWS security controls**
- Agreements available in AWS Artifact include the Business Associate Addendum (BAA) and the Nondisclosure Agreement (NDA)

# Incident Response Plans





# Incident Response Plans

---

- Review the AWS Security Incident Response Guide ([whitepaper](#))
- Based on the AWS Cloud Adoption Framework (CAF)
- Broken into four areas:
  1. Educate
  2. Prepare
  3. Simulate
  4. Iterate



# Incident Response Plans

---

---

## Educate

- Educate staff about cloud and cloud usage
- Topics:
  - [Shared Responsibility](#)
  - [Incident Response in the Cloud](#)
  - [Cloud Security Incidents](#)
  - [Understanding Cloud Capabilities](#)



## Prepare – People

- Prepare team to detect and respond to incidents
- Topics:
  - Define Roles and Responsibilities
  - Define Response Mechanisms
  - Create a Receptive and Adaptive Security Culture
  - Predicting Response



## Prepare – Technology

- Prepare tools, cloud services, runbooks etc.
- Topics:
  - [Prepare Access to AWS Accounts](#)
  - [Prepare Processes](#)
  - [Cloud Provider Support](#)



# Incident Response Plans

---

---

## Simulate

- Simulate expected and unexpected security events
- Topics:
  - [Security Incident Response Simulations](#)
  - [Simulation Steps](#)
  - [Simulation Examples](#)



# Incident Response Plans

---

---

## Iterate

- Iterate on the outcome of your simulation
- Topics:
  - [Runbooks](#)
  - [Automation](#)

# Detect and Respond





# Amazon Detective

---

---

- Analyze, investigate, and quickly identify the root cause of potential security issues or suspicious activities
- Automatically collects data from AWS resources
- Uses machine learning, statistical analysis, and graph theory
- Creates a unified, interactive view of resources, users and interactions between them
- Data sources include VPC Flow Logs, CloudTrail, and GuardDuty



# AWS GuardDuty

---

- Intelligent threat detection service
- Detects account compromise, instance compromise, malicious reconnaissance, and bucket compromise
- Continuous monitoring for events across:
  - **AWS CloudTrail Management Events**
  - **AWS CloudTrail S3 Data Events**
  - **Amazon VPC Flow Logs**
  - **DNS Logs**



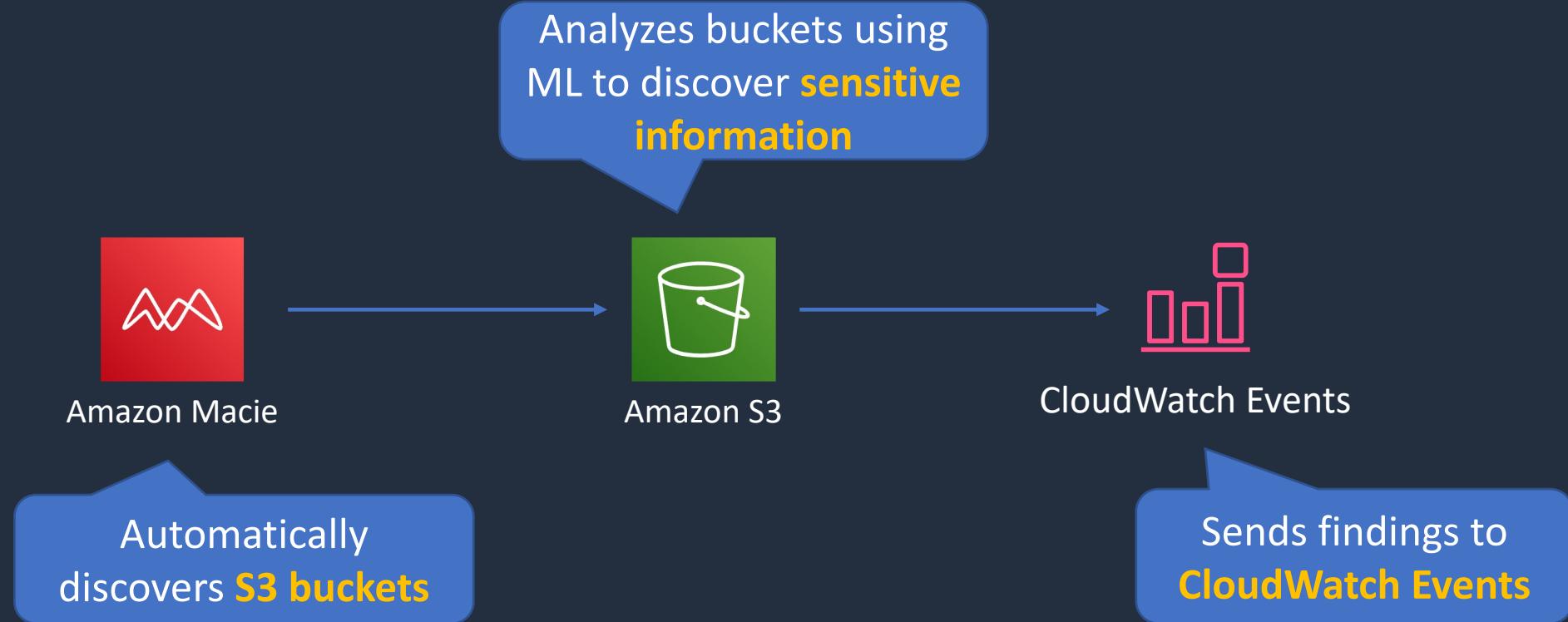
# Amazon Macie

---

- Macie is a fully managed data security and data privacy service
- Uses machine learning and pattern matching to discover, monitor, and help you protect your sensitive data on Amazon S3
- Macie enables security compliance and preventive security
- Can Identify a variety of data types, including PII, Protected Health Information (PHI), regulatory documents, API keys, and secret keys



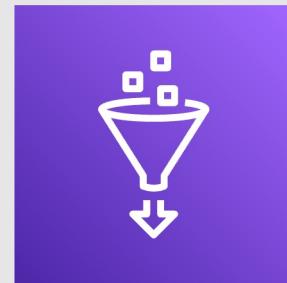
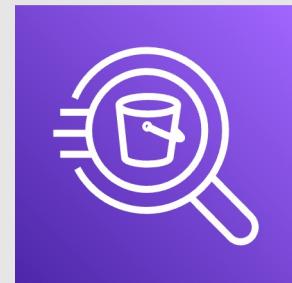
# Amazon Macie



# AWS Detective and GuardDuty

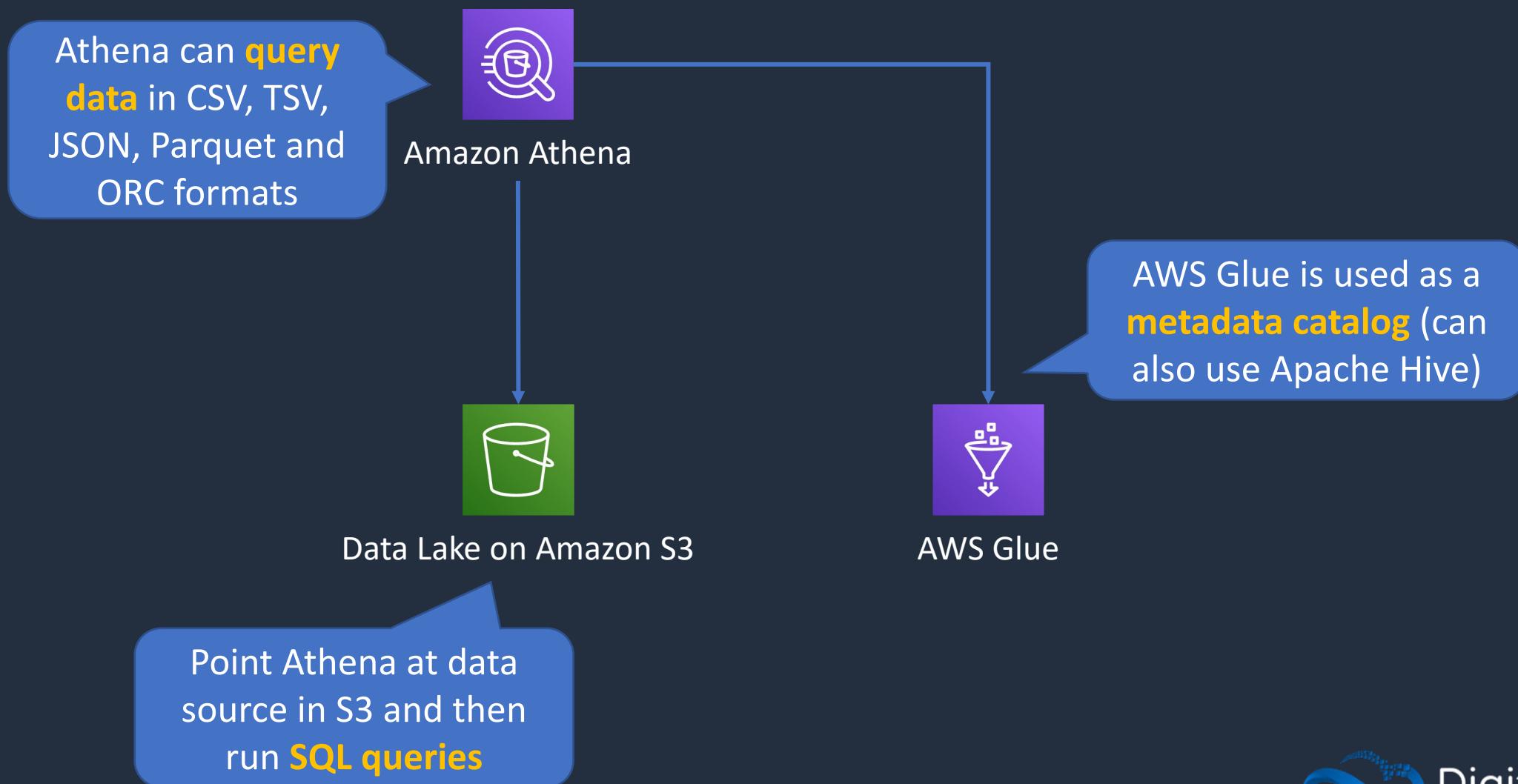


# Amazon Athena and AWS Glue



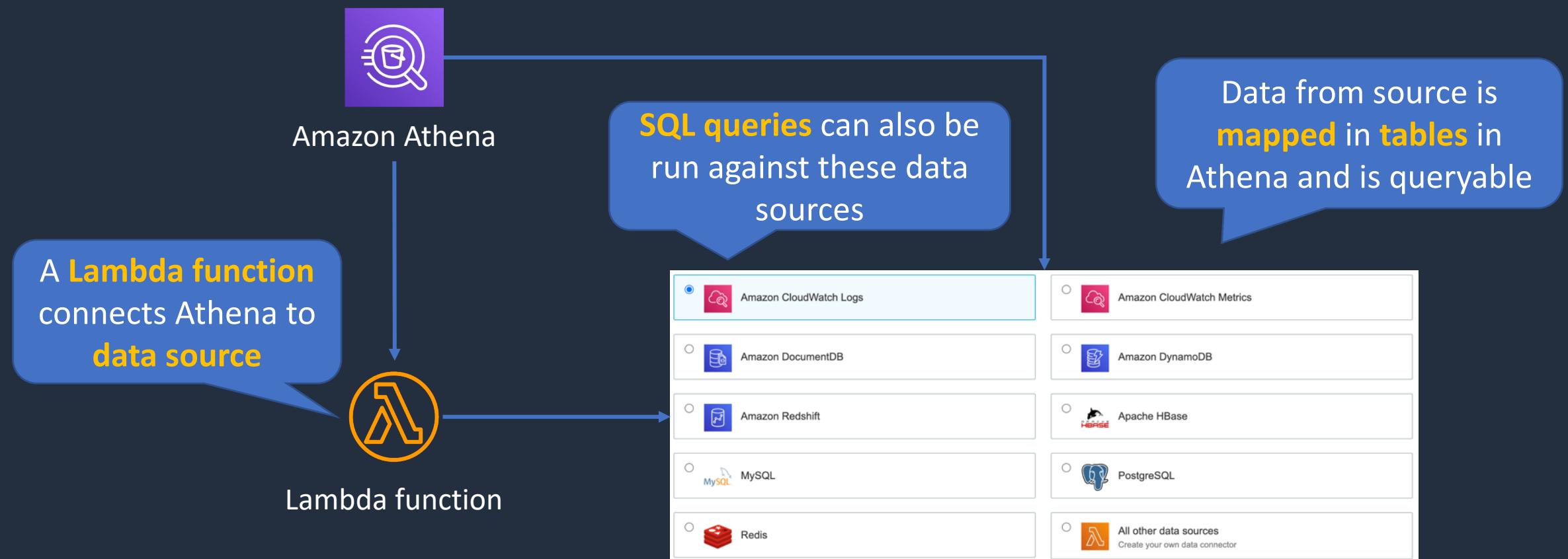


# Amazon Athena and AWS Glue





# Amazon Athena and AWS Glue





# Amazon Athena

---

- Athena queries data in S3 using SQL
- Can be connected to other data sources with Lambda
- Data can be in CSV, TSV, JSON, Parquet and ORC formats
- Uses a managed Data Catalog (AWS Glue) to store information and schemas about the databases and tables



# Optimizing Athena for Performance

- **Partition your data**
- **Bucket your data** – bucket the data within a single partition
- **Use Compression** – AWS recommend using either Apache Parquet or Apache ORC
- **Optimize file sizes**
- **Optimize columnar data store generation** – Apache Parquet and Apache ORC are popular columnar data stores
- **Optimize ORDER BY and Optimize GROUP BY**
- **Use approximate functions**
- **Only include the columns that you need**



# AWS Glue

---

---

- Fully managed extract, transform and load (ETL) service
- Used for preparing data for analytics
- AWS Glue runs the ETL jobs on a fully managed, scale-out Apache Spark environment
- AWS Glue discovers data and stores the associated metadata (e.g. table definition and schema) in the AWS Glue Data Catalog
- Works with data lakes (e.g. data on S3), data warehouses (including RedShift), and data stores (including RDS or EC2 databases)



# AWS Glue

---

- You can use a **crawler** to populate the AWS Glue Data Catalog with tables
- A crawler can crawl multiple data stores in a single run
- Upon completion, the crawler creates or updates one or more tables in your Data Catalog.
- ETL jobs that you define in AWS Glue use the Data Catalog tables as sources and targets

# SECTION 12

## Exam Cram

# Domain 1

# Incident Response





# Exam Cram: Domain 1 - Incident Response

---

---

- Domain 1 maps to section:
  - 11 - Data Analysis and Incident Response



# Exam Cram: Domain 1 - Incident Response

---

- VPC Flow Logs can be used to capture IP traffic and publish it to Amazon CloudWatch where metric filters can be used to search for specific event patterns such as connection attempts to a specific TCP port
- Automatic provisioning of a secure forensic environment can be achieved using AWS CloudFormation for infrastructure creation and AWS Step Functions to orchestrate the processes (using Lambda) required for forensic analysis
- EC2 instances with Elastic IPs must have the EIPs added to the ingress rules of security groups for connectivity using the public addresses across AZs
- You can collect memory dumps from EC2 instances that are unresponsive using the EC2Rescue CLI with the /offline mode and the device ID specified



# Exam Cram: Domain 1 - Incident Response

---

- To securely share encrypted snapshots with a separate AWS account, share the snapshot, use a customer managed KMS key, and allow the Decrypt and CreateGrant actions for the target account in the key policy
- If SSH keys for EC2 instances are compromised they can be removed by connecting to each instance and replacing the public key information in the authorized\_keys file
- If access key IDs and secret access keys are exposed, you should invalidate any temporary security credentials and delete the access key IDs and secret access keys
- If many access key IDs have been exposed, you can generate a credential report in each account in an organization to identify the users each access key ID belongs to and then rotate those keys
- If temporary security credentials issued by AWS STS are compromised you can revoke all active sessions for the IAM Role



# Exam Cram: Domain 1 - Incident Response

---

- You can block DDoS attacks where the User-Agent field of the request header has a certain value by using a web ACL with a string match condition that matches that value

# Domain 2

# Logging and Monitoring





# Exam Cram: Domain 2 - Logging and Monitoring

---

---

- Domain 2 maps to section:
  - 9 – Logging, Monitoring, and Auditing



# Exam Cram: Domain 2 - Logging and Monitoring

---

- Amazon CloudWatch is used for performance, logging, alarms, and Events
- Amazon EventBridge is based on Amazon CloudWatch Events
- The unified CloudWatch agent can be installed on EC2 instances to collect application log files and stream them to CloudWatch Logs
- The Unified CloudWatch Agent sends system-level metrics for EC2 and on-premises servers
- System-level metrics include memory and disk usage
- An EC2 instance in a private subnet running the unified CloudWatch agent can be configured to send logs to CloudWatch Logs securely via an interface VPC endpoint



# Exam Cram: Domain 2 - Logging and Monitoring

---

- You can also send CloudWatch Logs to S3, Kinesis Data Streams, and Kinesis Data Firehose
- If Lambda fails to write logs to CloudWatch Logs check the function execution role for permissions
- AWS CloudTrail logs API activity for auditing purposes and you must create a trail to store logs indefinitely (90 days otherwise)
- A CloudTrail trail can be configured in the management account of an AWS Organization with logging to a centralized S3 bucket. Administrators in child accounts cannot modify the trail
- Single region AWS CloudTrail trails can be reconfigured to apply to all regions and log API activity to a central Amazon S3 bucket
- To allow delivery of CloudTrail events to an S3 bucket you must ensure the S3 bucket policy grants CloudTrail the s3:PutObject permission and verify that the S3 bucket and prefix defined in CloudTrail exists



# Exam Cram: Domain 2 - Logging and Monitoring

- To automatically notify about security group changes use AWS CloudTrail with a CloudWatch Logs log group and use a metric filter to match security group changes and SNS to send notifications
- AWS Config can be used with the managed rule cloudtrail-enabled to check that CloudTrail is enabled. Systems Manager automation can be configured for automatic remediation if the rule returns a non-compliant state
- You can run ad-hoc SQL queries on ELB access logs in an S3 bucket using Amazon Athena
- The restricted-ssh managed rule in AWS Config can be configured to identify security groups that allow unrestricted traffic on port 22 (SSH)



# Exam Cram: Domain 2 - Logging and Monitoring

---

- Amazon Kinesis can be used for ingesting real-time streaming data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications
- Amazon OpenSearch can receive data from Kinesis and analyze that and store that data
- To enable Amazon Detective, you must have enabled Amazon GuardDuty more than 48 hours ago
- You can generate an alert if users add bucket policies by configuring an Amazon EventBridge rule that uses the “AWS API Call via CloudTrail” event source and the “s3:PutBucketPolicy” event pattern

# Domain 3

# Infrastructure Security





# Exam Cram: Domain 3 - Infrastructure Security

---

---

- Domain 3 maps to sections:
  - 6 – Infrastructure Security
  - 7 – Edge Security



# Exam Cram: Domain 3 - Infrastructure Security

---

- Use separate VPCs to isolate infrastructure, use multiple AZs for high availability, and use subnets to isolate the tiers of your application
- Control network traffic with Security Groups and Network ACLs. Security groups are stateful and NACLs are stateless
- Security groups support allow rules only, are applied at the ENI level, and all rules are evaluated. NACLs apply only to traffic **entering / exiting** the subnet and are applied in order until an allow or deny is reached
- You can audit security group changes by sending AWS API calls via CloudTrail to Amazon EventBridge and SNS for notifications
- Use private subnets for your instances if they should not be accessed directly from the internet and use VPC interface, gateway endpoints, and AWS PrivateLink to keep traffic private
- Use a NAT gateway or NAT instance for enabling internet access for EC2 instances in private subnets when using IPv4



# Exam Cram: Domain 3 - Infrastructure Security

---

- Outbound internet connectivity for instances using IPv6 can be enabled using an egress-only internet gateway which will only allow outbound and not inbound traffic
- A Lambda function can be configured to connect to DynamoDB using private IPs by configuring the function in a VPC and using a VPC endpoint for the DynamoDB table
- Lambda functions can be attached to VPCs by configuring the VPC and subnets and you can then use security groups
- For Lambda functions running in a VPC you will need a NAT gateway to reach the internet
- Amazon OpenSearch can be deployed in a VPC, and you will then have a private endpoint (cannot switch to public later)
- Amazon Route 53, AWS Shield, and Elastic Load Balancer can be used to protect EC2 instances against layer 4 DDoS attacks



# Exam Cram: Domain 3 - Infrastructure Security

---

- Alias records can be used at the zone apex whereas CNAME records can only be used on subdomains
- Internet connectivity to an EC2 instance relies on correct configuration of security groups, network ACLs, route table entries for an internet gateway, and host-based firewall settings (if applicable)
- EC2 instances can be scanned for known software vulnerabilities using Amazon Inspector with the “Common vulnerabilities and exposures” assessment and the “Center for Internet Security (CIS) Benchmarks” assessment
- You can improve security of the instance metadata service by requiring the use of IMDSv2 by setting the “--metadata-options HttpTokens” option to “required”



# Exam Cram: Domain 3 - Infrastructure Security

---

- EC2 proxy instances and NAT instances must have source/destination checks disabled to be able to forward traffic
- All traffic sent to EC2 instances can be captured for inspection with an intrusion detection appliance by configuring VPC traffic mirroring with a network load balancer
- To connect securely to the CLI on EC2 instances the SSM agent can be used, and the connection made through Session Manager, bypassing SSH ports, and with access controlled by IAM user policy
- Systems Manager Patch Manager can be used to scan EC2 instances, identify vulnerable software, and install patches
- AWS Config can be used to check that EC2 instances have been launched from approved AMIs by using the rule “approved\_ami\_by\_id”
- If health checks are failing for EC2 instances behind an NLB check that the instance security groups, and subnet network ACLs allow traffic from the NLB IP addresses



# Exam Cram: Domain 3 - Infrastructure Security

---

- To ensure that TLS traffic to an ALB is secure even if the certificate private key is compromised you should create an HTTPS listener with a predefined security policy that supports forward secrecy (FS)
- You can redirect HTTP connections to HTTPS on an ALB by creating both types of listener and a rule that redirects HTTP to HTTPS
- An origin access identity (OAI) can be used to restrict access to an S3 bucket so only CloudFront can access the bucket contents
- You can add a **custom header** in CloudFront origin settings to restrict access to a specific ALB with a conditional rule on the ALB looking for the header value
- You can use geo restriction in CloudFront to restrict users in specific locations from accessing the content
- To protect against DDoS attacks a static website can be migrated from EC2 to S3 behind a CloudFront distribution with an AWS WAF WebACL



# Exam Cram: Domain 3 - Infrastructure Security

---

- An AWS WAF web ACL can be associated with an ALB and configured with an IP set match rule statement with a block action to deny incoming requests from specific IP addresses or ranges of addresses
- AWS WAF can be configured with rate-based rules that protect against layer 7 DDoS attacks by putting a temporary block on requests from IP addresses that send excessive requests
- Encryption in transit can be added to an AWS Direct Connect (DX) connection by implementing a Virtual Private Gateway (VGW) and establishing an encrypted site-to-site VPN over the DX connection
- AWS Certificate Manager (ACM) can be used to create SSL/TLS certificates using public or private domain names and subdomains or wildcards

# Domain 4

# Identity and Access Management





# Exam Cram: Domain 4 - Identity and Access Management

---

---

- Domain 4 maps to sections:
  - 3 – AWS IAM Fundamentals
  - 4 – IAM Access Control
  - 5 – AWS Organizations and Control Tower
  - 10 – Directory Services and Federation



# Exam Cram: Domain 4 - Identity and Access Management

---

- IAM Roles are used for delegation and are assumed by IAM entities. IAM policies are attached to roles and define the permissions for the identities or resources they are associated with
- The trust policy associated with an IAM role determines who is allowed to assume the role
- Identity-based policies are JSON permissions policy documents that control what actions an identity can perform, on which resources, and under what conditions
- Identity-based policies can be applied to users, groups, and roles. Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket
- AWS managed policies are standalone policies that can be attached to multiple users, groups, or roles. Customer managed policies allow you to create your own custom policy permissions to apply to users, groups, or roles



# Exam Cram: Domain 4 - Identity and Access Management

---

- Groups are collections of users. Users can be members of up to 10 groups and you can assign permissions to all members of the group by attaching IAM policies
- Access authorization to an S3 bucket depends on the union of all the IAM policies, S3 bucket policies, and S3 ACLs that apply
- Permissions boundaries can be configured to specify the maximum available permissions that an identity-based policy can grant to an IAM entity
- S3 bucket policies can be configured to only allow access from the ID of a VPC endpoint
- The condition element in an IAM policy can be used to restrict access based on group membership, IP address etc.
- AWS Secrets Manager can be used for storing encrypted database connection strings and automatically rotating them for some AWS database services



# Exam Cram: Domain 4 - Identity and Access Management

---

- Automatic rotation of secrets is not available for Systems Manager Parameter Store or other database types in Secrets Manager. You would need to use Lambda to rotate the secrets
- To successfully issue the GetParameter API against secure string parameters in Systems Manager Parameter Store, the EC2 instance must have an IAM role assigned that has decrypt permissions on the KMS key and permissions to retrieve parameters
- To secure RDS database connection credentials store them in AWS Secrets Manager or Systems Manager Parameter Store in an encrypted state. Configure the EC2 instance role with credentials to retrieve the connection credentials
- Automatic rotation is available for Secrets Manager when using specific types of database such as RDS, DocumentDB and RedShift
- Host-based firewalls on EC2 instances may be used when complex rules are required to exceed the limits of security groups and network ACLs



# Exam Cram: Domain 4 - Identity and Access Management

---

- You can restrict that a KMS key is used only by specific services by using the kms:ViaService condition key and specifying the services that should be allowed to use the key
- To enforce that a role can only be assumed after authentication with MFA add an "aws:MultiFactorAuthPresent : true" condition to the role's trust policy
- You can identify the federated user that terminated an EC2 instance by searching CloudTrail for the TerminateInstances event and taking the IAM Role ARN and then looking for AssumeRoleWithSAML events that used that role
- AWS Organizations enables centralized management and governance of multiple AWS accounts, and you can apply service control policies (SCPs), tag policies, and use consolidated billing



# Exam Cram: Domain 4 - Identity and Access Management

---

- SCPs attached to member accounts control permissions for all users including the root user
- An explicit deny in an SCP cannot be overridden with an allow statement anywhere beneath it in the hierarchy
- The only way to get around the inheritance of SCP deny statements is to ensure that the SCP is not applied to any OUs with accounts you do not want to apply the deny statement to
- You can use the condition key aws:PrincipalOrgID in policies to require all principals accessing the resource to be from an account in an AWS Organization
- Use \* as a principal and aws:PrincipalOrgID as a condition key to restrict to users within the organization



# Exam Cram: Domain 4 - Identity and Access Management

---

- AWS Control Tower can be used for deploying and managing large numbers of AWS accounts. Control Tower creates a well-architected multi-account baseline known as a landing zone that is based on best practices
- With Control Tower you have preventive guardrails and detective guardrails. Preventive guardrails are based on SCPs and disallow API actions. Detective guardrails are implemented using AWS Config rules and Lambda functions and monitor and govern compliance
- AWS Managed Microsoft AD is a managed implementation of Microsoft Active Directory running on Windows Server 2012 R2. A HA pair of Windows Server 2012 Domain Controllers (DCs) are deployed in your VPC
- AD Connector is used to connect AWS to your on-premises Microsoft Active Directory. You can use it for sign-in to various AWS applications, seamlessly join Windows EC2 instances to the domain, and implement federated sign-in to the AWS management console



# Exam Cram: Domain 4 - Identity and Access Management

---

- When federating between a corporate IdP and IAM you can use IAM roles to manage access to AWS resources
- Federated sign in to AWS using ADFS and SAML can be used if you keep your identities in an on-premises directory such as Microsoft AD
- Password policies should be configured within Active Directory when using IAM federation
- Password policies can also be configured in IAM user pools and in IAM itself
- AWS SSO can be used for single sign on across many accounts using an existing on-premises identity provider
- Amazon Cognito is used for sign-in and sign-up for mobile applications. User pools are used for creating users or federating to social IdPs and identity pools are used for obtaining temporary, limited-privilege credentials for AWS services

# Domain 5

# Data Protection





# Exam Cram: Domain 5 - Data Protection

---

---

- Domain 5 maps to section:
  - 8 – Data and Application Protection



# Exam Cram: Domain 5 - Data Protection

---

- You can create and manage symmetric and asymmetric encryption keys with AWS KMS, and they are protected by hardware security modules (HSMs)
- KMS keys used to be known as “customer master keys” or CMKs and this terminology could still be used in the exam
- KMS keys can only encrypt data up to 4 KB in size and for anything larger you need to create data encryption keys
- AWS managed KMS keys are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS. You cannot manage these KMS keys, rotate them, or change their key policies
- Automatic rotation of KMS keys generates new key material every year (optional for customer managed keys) and is supported for symmetric keys with key material AWS KMS creates



# Exam Cram: Domain 5 - Data Protection

---

---

- You cannot use automatic rotation in the following situations:
  - You are using asymmetric KMS keys
  - You have KMS keys in custom key stores (AWS CloudHSM)
  - You are using KMS keys with imported key material
- Manual rotation is creating a new KMS key with a different key ID . You must then update your applications with the new key ID. You can use an alias to represent a KMS key, so you don't need to modify your application code
- KMS key policies define management and usage permissions for KMS keys. Multiple policy statements can be combined to specify separate administrative and usage permissions



# Exam Cram: Domain 5 - Data Protection

---

- When using AWS KMS customer managed keys with KMS key material you can enable automatic rotation for the key material every year. Automatic rotation of AWS managed KMS Keys is every 3 years and cannot be modified
- Temporary permissions to decrypt data encrypted with KMS keys can be provided using KMS grants. You must be granted permissions to use an AWS KMS key for encryption/decryption in the key policy
- If you require a managed service for encryption keys but must be able to immediately delete the keys use AWS KMS with imported key material and then use the DeleteImportedKeyMaterial API to remove the key material if necessary
- Cryptographic erasure within 15 minutes can be achieved using a KMS key with imported key material and deleting the key material if necessary



# Exam Cram: Domain 5 - Data Protection

- An InvalidKeyId error when trying to use an AWS KMS key may indicate that the key is disabled
- AWS CloudHSM should be used if you need hardware based HSMs within a VPC
- If you need to import your own key material into KMS use a default key store and a customer managed KMS key
- RDS databases cannot be encrypted after creation and you must instead take a snapshot, encrypt a copy of the snapshot, and create a new encrypted DB from the encrypted snapshot
- A multi-tier web application with an RDS database can be secured using encrypted EBS volumes, encrypted DB volumes, TLS connections to the DB, and connection strings stored in AWS Secrets Manager (with automatic rotation)



# Exam Cram: Domain 5 - Data Protection

---

- RDS database instances are configured with an SSL/TLS certificate, and you can download the AWS-provided root certificates for all Regions or specific Regions
- To identify and alert on unencrypted databases use AWS Config with an SNS notification
- Policy variables can be used to specify placeholders in a policy that are replaced with values from the context of the connection request. For example, you can replace the variable with the friendly name of the user making the connection
- The S3 Glacier vault lock operation can be aborted after the initiate vault lock operation
- To secure an S3 bucket with your own keys whilst not managing the encryption process, use server-side encryption with AWS KMS keys and supply your own key material



# Exam Cram: Domain 5 - Data Protection

- Client-side encryption means the entire encryption and decryption takes place outside of AWS and you manage the keys. You can optionally use KMS keys
- Take the following actions to enforce encryption for an S3 bucket:
  - Add a condition to the S3 bucket policy that allows actions if the request meets the condition "aws:SecureTransport": "true"
  - Configure default encryption for the S3 bucket. Add a condition to the S3 bucket policy that denies PUT requests that don't include the "x-amz-server-side-encryption" header
- Amazon EBS encryption affects:
  - Data at rest inside the volume
  - All data moving between the volume and the instance (in-transit).
  - All snapshots created from the volume
  - All volumes created from those snapshots



# Exam Cram: Domain 5 - Data Protection

---

- For Amazon EFS, encryption at-rest can be enabled when the file system is created but not after it has been created
- Encryption in-transit can be enabled for Amazon EFS file systems when mounting the file system using the TLS protocol
- The files, deployment packages, and environment variables in a Lambda function are encrypted at rest using an AWS KMS key. AWS recommend that you use Secrets Manager for storing sensitive data instead of environment variables
- To ensure that code written by a developer cannot be deployed to Lambda functions you can use AWS Signer and revoke all versions for the signing profile assigned to the developer
- To verify code integrity when deploying code to Lambda functions use AWS Signer and use IAM policies to enforce that developers can only create functions that have code signing enabled



# Exam Cram: Domain 5 - Data Protection

---

---

- Use a VPC endpoint for Kinesis Data Streams to keep the connection private between the VPC and the stream
- Data in Amazon DynamoDB tables is encrypted at rest, and you can choose to select an AWS KMS key or use the default key. Clients connect using published APIs over TLS
- All data flowing across AWS Regions over the AWS global network is automatically encrypted at the physical layer before it leaves AWS secured facilities. All traffic between AZs is also encrypted
- Traffic between instances may be encrypted in some circumstances. The instances must use a supported instance type and be within the same Region and VPC (or in a peered VPC)