

Arquitecturas de nube con AWS

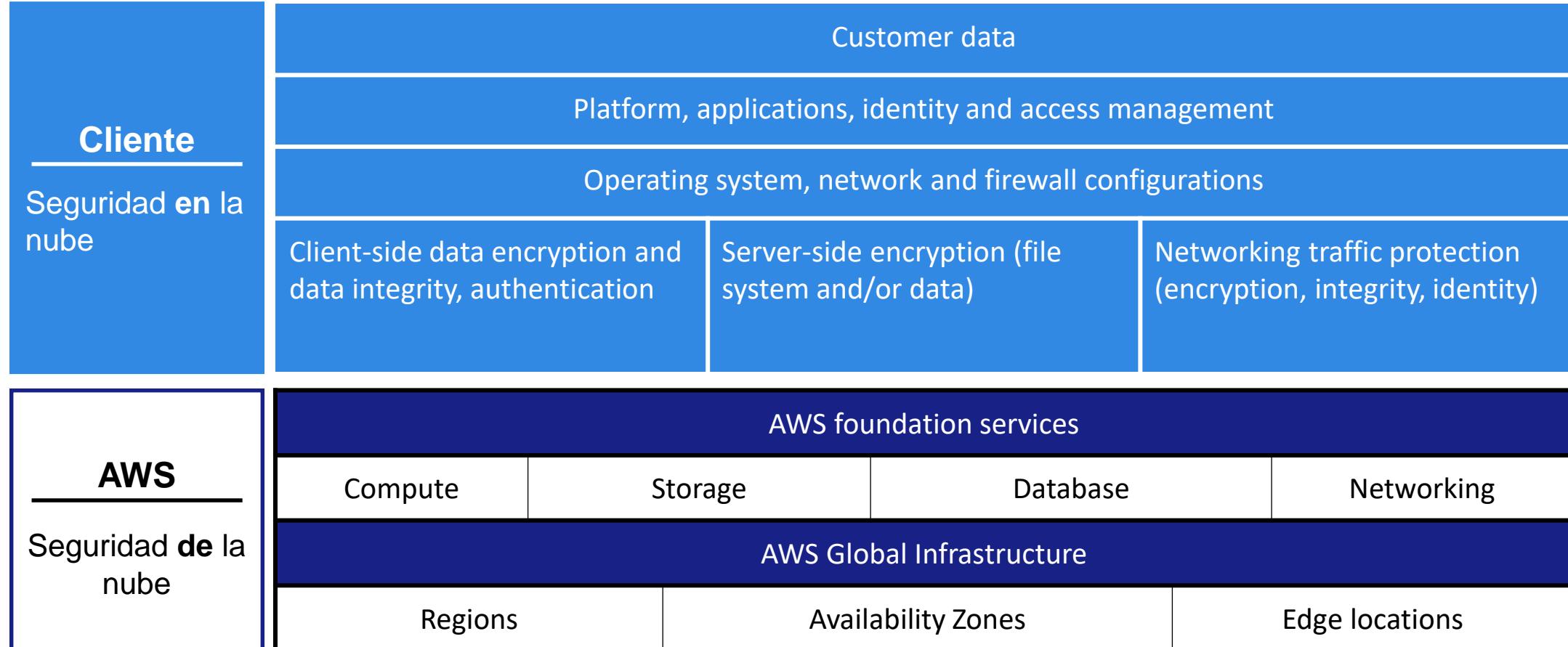
Ing. Fernando Lichtschein

Ing. Mora Villa Abrille

2. Seguridad de acceso

Conceptos fundamentales

Modelos de responsabilidad compartida



Well-architected framework

Pilares



Operational
Excellence



Security



Reliability



Performance
Efficiency



Cost
Optimization



Sustainability

Pilar de seguridad

Principios de diseño



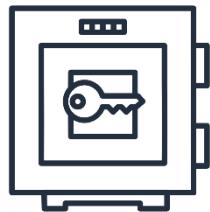
Gestión de identidades



Proteger los datos en tránsito y en reposo



Asegurar todas las capas



Restringir accesos



Mantener la trazabilidad



Prepararse para eventos



Automatizar buenas prácticas

Gestión de identidades

AWS Identity and Access Manager (IAM)



- Control de accesos a recursos de AWS para individuos o grupos
- Integración con otros servicios
- Federación de identidades
- Autenticación multifactor
- Permisos granulares

Gestión de identidades



Autenticar

¿Quién solicita acceso a la cuenta de AWS y a los recursos que contiene?

Establecer la **identidad** mediante credenciales



Autorizar

¿Qué puede hacer un usuario o una aplicación que ya está autenticado?

Permitir o denegar la solicitud.

AWS Identity and Access Management (IAM)

Definiciones

Recurso	Entidad	Identidad	Principal
Objeto almacenado en IAM.	Recurso de IAM que AWS usa para autenticación	Recurso de IAM que puede recibir una autorización en una política de acceso	Persona o aplicación que puede ingresar (sign in) y generar requests en AWS
Usuario, grupo, rol o política	Roles y usuarios	Usuarios, roles y grupos	

Acceso a recursos en AWS



Usuario

Una persona o aplicación que puede autenticarse con una cuenta de AWS



Grupo

Un grupo de usuarios IAM que tienen permisos idénticos



Rol

Identidad que se utiliza para otorgar un conjunto **temporal** de permisos



Política

Documento que define los recursos accesibles y los niveles de acceso de cada uno

Autenticación

Credenciales

Credenciales de usuario

Autenticación en la consola

User email + password

AWS Access key

AWS CLI

SDK

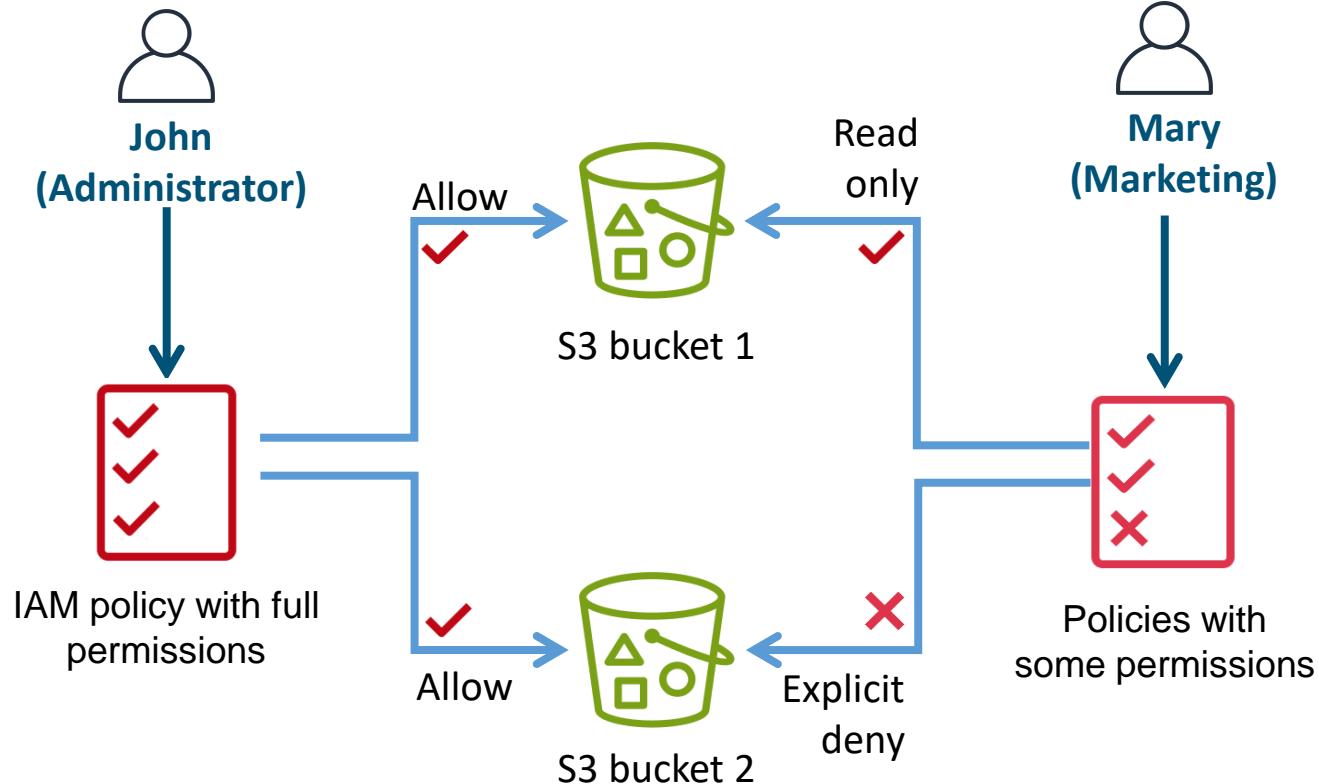
Llamadas directas a API

Access key ID + secret access key

Buenas prácticas de control de accesos

- Aplicar el principio de mínimo privilegio
- Habilitar la autenticación multifactor (MFA)
- Implementar el uso de credenciales temporales siempre que se pueda
- Rotar las claves de acceso para el uso de credenciales de largo plazo
- Usar contraseñas fuertes y complejas
- Proteger las credenciales locales
- Usar AWS Organizations

Principio del mínimo privilegio

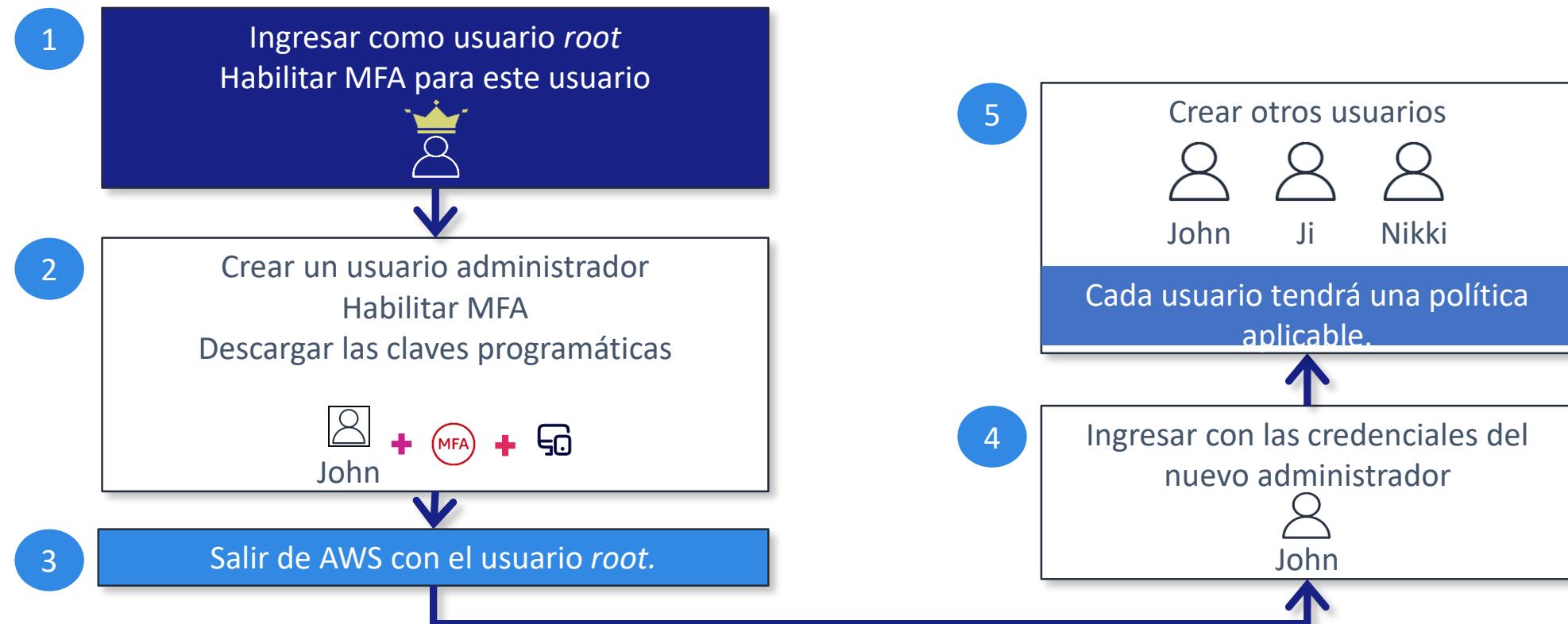


Buenas prácticas

- Asignar únicamente los permisos necesarios para ejecutar la tarea.
- Dar un conjunto mínimo de permisos y luego ir asignando otros en la medida que se necesitan.
- Revisar y revocar los permisos innecesarios.

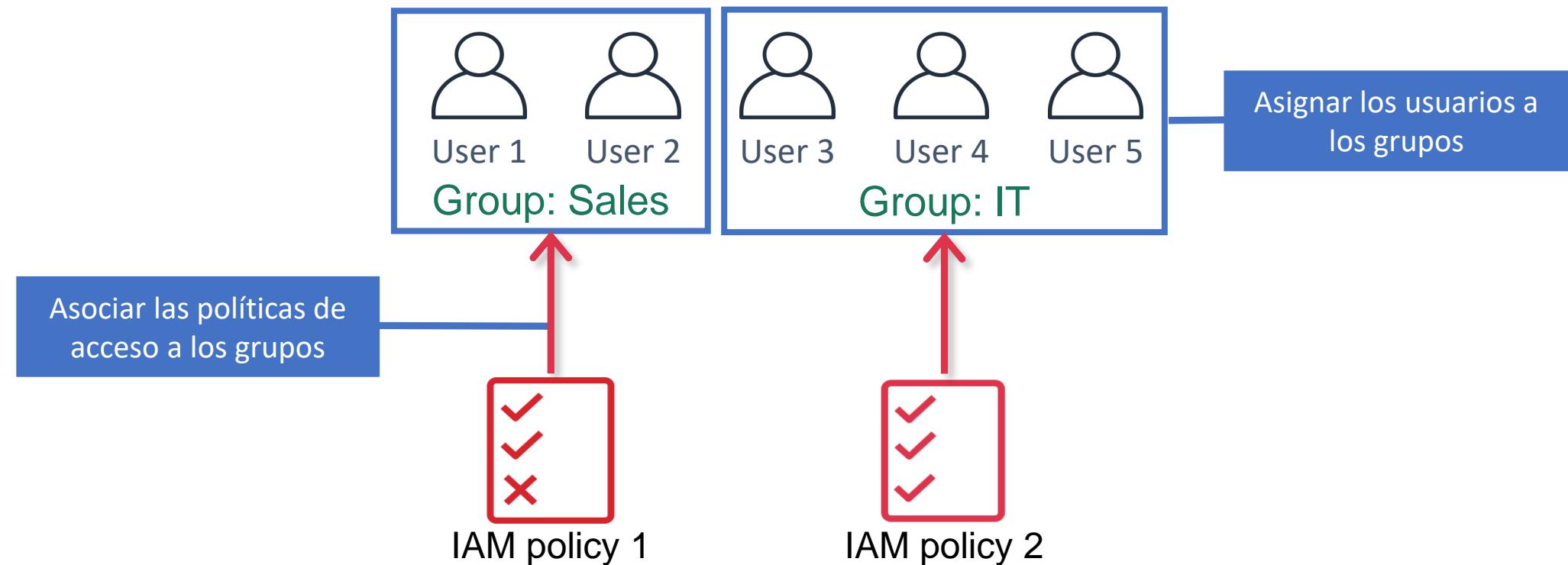
Buenas prácticas de control de accesos

Proteger el usuario administrador



Control de accesos

Usuarios y grupos



Roles IAM en AWS

Definiciones

¿Qué es?

Un medio para obtener **credenciales transitorias**.

No está asociado de manera única con una persona.

Puede ser asumido por una persona, una aplicación o un servicio.

¿Cuándo se usa?

Cuando queremos delegar un acceso sin asignar permisos permanentes a un *principal*.

Ejemplos:

- Accesos entre distintas cuentas de AWS
- Aplicaciones móviles
- Aplicaciones que corren en EC2

Control de accesos

Políticas y permisos

Las políticas permiten asignar accesos granulares a los principals

Basadas en identidades

Se vinculan con un usuario, un grupo o un rol

Basadas en recursos

Se asocian a un recurso de AWS



Control de accesos

Políticas y permisos

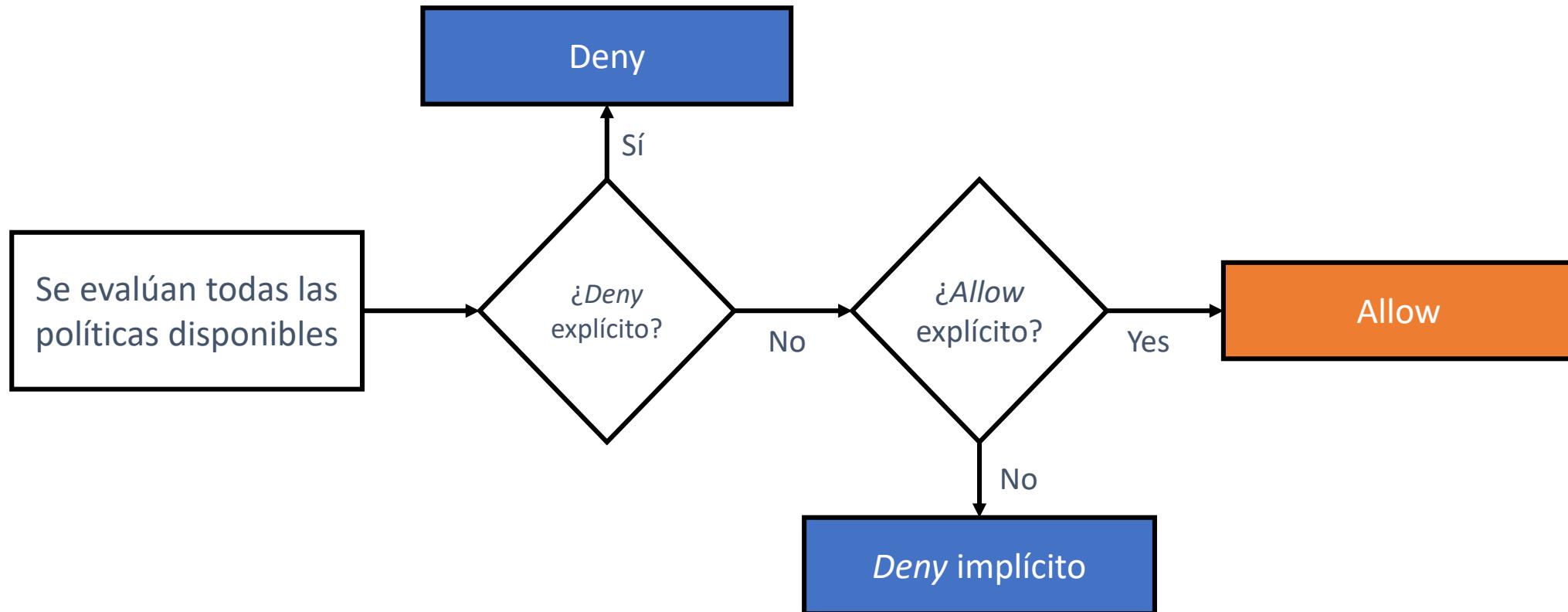
Definición de permisos en políticas IAM

- Formato: JSON
- La política define los recursos y operaciones permitidos y denegados.
- Siguen el principio de mínimo privilegio.



Buenas prácticas de control de accesos

Proteger el usuario administrador



Políticas

Ejemplos

Identity-based
(Asociada con un usuario, grupo o rol)

Juan

Recurso	Get	Put	List
Bucket 2	Allow	Allow	Allow
Bucket 3	N/A	N/A	Allow

Resource-based
(Attached to an AWS resource)

Bucket 1

User	Get	Put	List
Juan	Allow	Deny	Allow

Bucket 2

User	Get	Put	List
Juan	Allow	N/A	Allow

¿Qué puede hacer Juan en el *bucket 1*? ¿Y en el *bucket 2*?

Políticas

Ejemplos

Identity-based
(Asociada con un usuario, grupo o rol)

Juan

Recurso	Get	Put	List
Bucket 2	Allow	Allow	Allow
Bucket 3	N/A	N/A	Allow

Resource-based
(Attached to an AWS resource)

Bucket 1

User	Get	Put	List
Juan	Allow	Deny	Allow

Bucket 2

User	Get	Put	List
Juan	Allow	N/A	Allow

¿Qué puede hacer Juan en el *bucket 1*? ¿Y en el *bucket 2*?

Estructura de una política

Elemento	Información
Version	Versión del lenguaje que queremos usar
Statement	Define qué se permite o deniega en función de ciertas condiciones
Effect	Allow o deny
Principal	<p>Política basada en recursos. La cuenta, usuario, rol o usuario federado al que se otorga o deniega el permiso.</p> <p>Política basada en identidades. Este dato es implícito, y corresponde al usuario o el rol al que se asocia la política.</p>
Action	<p>La acción sobre la cual se otorga o deniega el permiso.</p> <p>Ejemplo: "Action": "s3:GetObject"</p>
Resource	<p>Recurso o recursos a los que se aplica la acción.</p> <p>Por ejemplo: "Resource": "arn:aws:sqs:us-west-2:123456789012:queue1" (ARN = AWS resource name)</p>
Condition	Condiciones que deben cumplirse para que se aplique la regla

Ejemplos

Política basada en recursos

```
{  
    "version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["dynamodb:*", "s3:*"]  
        },  
        {  
            "Resource": [  
                "arn:aws:dynamodb:region:account-number-without-hyphens:table/course-notes",  
                "arn:aws:s3:::course-notes-web",  
                "arn:aws:s3:::course-notes-mp3/*"]  
        },  
        {  
            "Effect": "Deny",  
            "Action": ["dynamodb:*", "s3:*"]  
        },  
        {  
            "NotResource": [  
                "arn:aws:dynamodb:region:account-number-without-hyphens:table/course-notes",  
                "arn:aws:s3:::course-notes-web",  
                "arn:aws:s3:::course-notes-mp3/*"]  
        }  
    ]  
}
```

Ejemplos

Política basada en identidades

```
{  
    "version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": [  
             "iam:*LoginProfile",  
             "iam:*AccessKey*",  
             "iam:*SSHPublicKey*"  
         ],  
         "Resource": [  
             "arn:aws:iam::account-id-without-hyphens:user/${aws:username}"  
         ]  
    }  
}
```

[IAM API Reference](#)

Actividad

Análisis de políticas IAM

Caso 1

Revisemos esta política

```
{  
    "version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:Get*",  
            "iam>List*"  
        ],  
        "Resource": "*"  
    }  
}
```

1. ¿Sobre qué servicio de AWS asigna accesos?
2. ¿Permite la política crear un usuario, grupo, política o rol?
3. Identificar tres acciones específicas que permite la acción iam:Get*

[IAM API Reference](#)

Caso 2

¿Qué permite hacer esta política?

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

1. ¿Esta política permite eliminar cualquier instancia de EC2 sin restricciones?
2. ¿Permite ejecutar la acción desde cualquier lugar?
3. ¿Podría eliminar la instancia si me conecto desde la IP 192.0.2.243?

[IAM API Reference](#)

Caso 3

¿Qué permite hacer esta política?

```
{  
    "version": "2012-10-17",  
    "Statement": [  
        "Condition": {  
            "StringNotEquals": {  
                "ec2:InstanceType": [  
                    "t2.micro",  
                    "t2.small"]  
            }  
        },  
        "Resource": "arn:aws:ec2:*::instance/*",  
        "Action": [  
            "ec2:RunInstances",  
            "ec2:StartInstances"  
        ],  
        "Effect": "Deny"  
    ]  
}
```

1. ¿Qué acciones permite esta política?
2. ¿Qué pasaría si agregáramos este statement?

```
{  
    "Effect": "Allow",  
    "Action": "ec2:*"  
}
```

3. En ese caso, ¿el usuario podría eliminar una instancia m3.xlarge de la cuenta?

[IAM API Reference](#)

Actividades

AWS Academy



Material

Guía de estudio del
módulo 3

[WAF - Pilar de seguridad](#)

Laboratorio

Exploring IAM Lab

Vencimiento: 22/8

Muchas gracias.

www.austral.edu.ar