

AWS Academy Introduction to Cloud

Administración y gobernanza

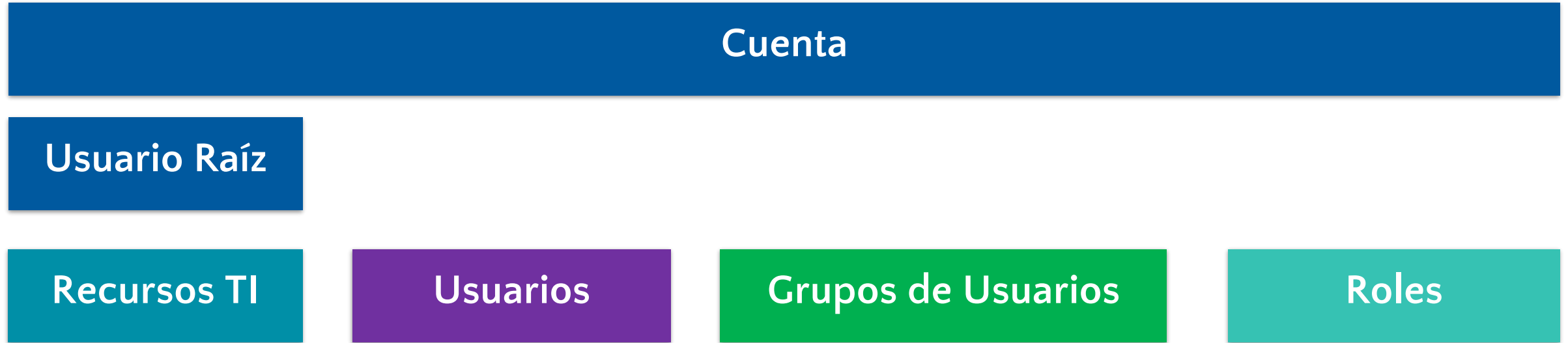
Temas

- Estructura de una cuenta de AWS
 - Credenciales de seguridad
 - Identidad (Usuario raíz, Usuario, Grupo, Rol)
- AWS Organizations
 - Estrategia multicuenta
 - Políticas de control de servicio
- AWS Control Tower
- AWS Identity and Access Management (IAM)
 - Políticas de acceso.
 - Usuario, Grupo, Rol.

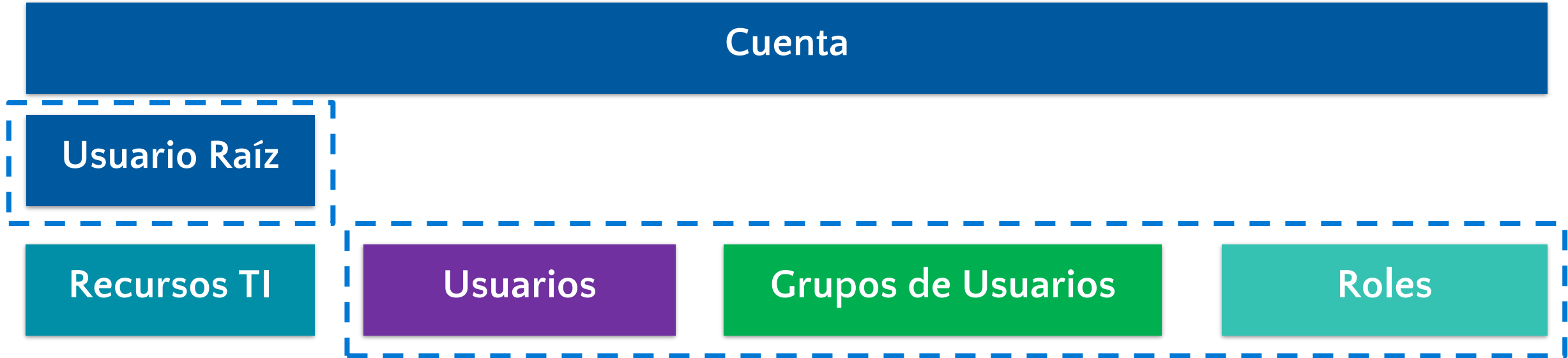
Administración y gobernanza

Estructura de una cuenta de AWS

Estructura de una cuenta de AWS

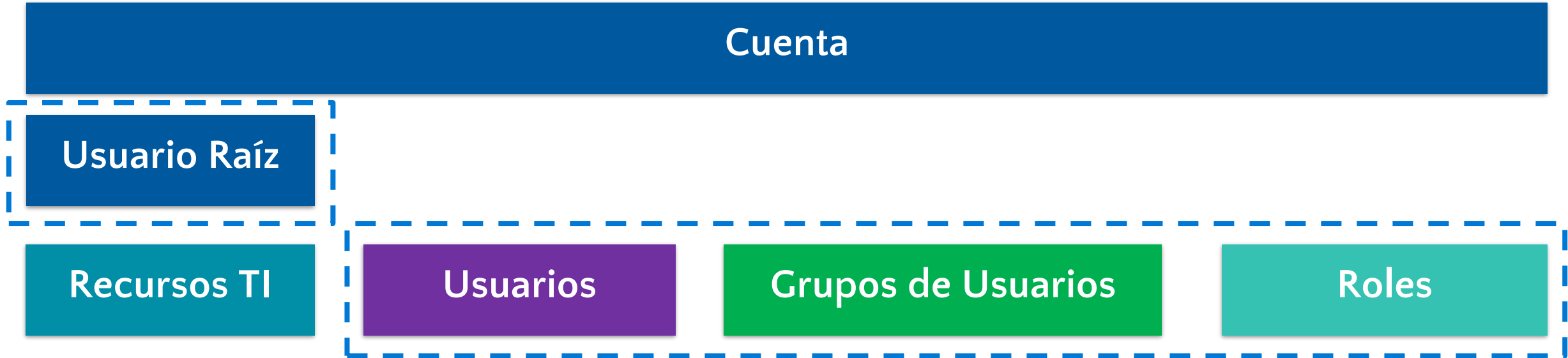


Estructura de una cuenta de AWS



Las **identidades** permiten identificar **quién** accede al sistema y **a qué recursos** tiene acceso.

Estructura de una cuenta de AWS



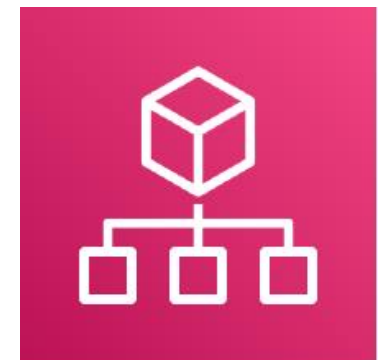
Las **identidades** proveen **credenciales de acceso** (usuario, contraseña, clave de acceso, etc) a los recursos.

Administración y gobernanza

AWS Organizations

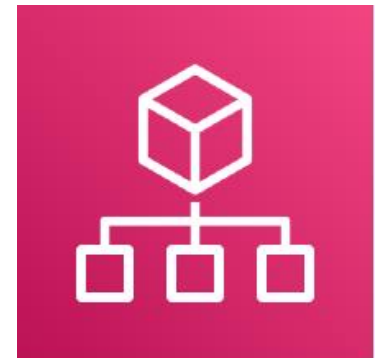
AWS Organizations

- Servicio global
- Permite gestionar **múltiples cuentas AWS**
- La cuenta principal se denomina **Cuenta maestra**
- La API del servicio permite automatizar la creación de cuentas AWS
- Se pueden implementar controles en las cuentas usando **Políticas de control de servicios(SCP)**
- **Beneficios:**
 - **Facturación consolidada** de todas las cuentas permitiendo tener un único medio de pago
 - Beneficio de **precios por uso agregado** (descuento por volumen para servicios como EC2, S3, entre otros)
 - **Agrupación de instancias EC2 reservadas** para un ahorro óptimo



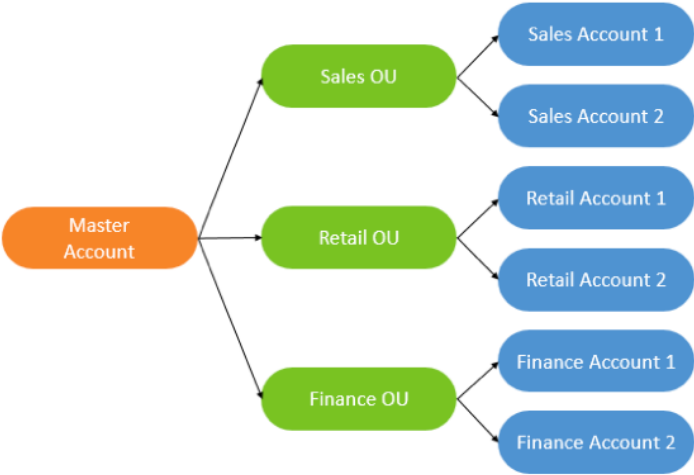
Estrategia Multicuenta

- Crear cuentas por **departamento**, por **centro de costes**, por **ambiente**, en función de **restricciones normativas(SCP)**
- El objetivo es tener un mejor aislamiento de recursos(VPC por cuenta), tener limites de servicios separados por cuenta(Budgets, estrategias de ahorro)
- Recomendaciones:
 - Manejar múltiples cuentas AWS, en lugar de una única con múltiples VPCs
 - Utilizar normas de etiquetado para facturación
 - Activas Cloudtrail en todas las cuentas y enviar logs a una cuenta centralizada
 - Enviar logs de Cloudwatch a cuenta central de logs

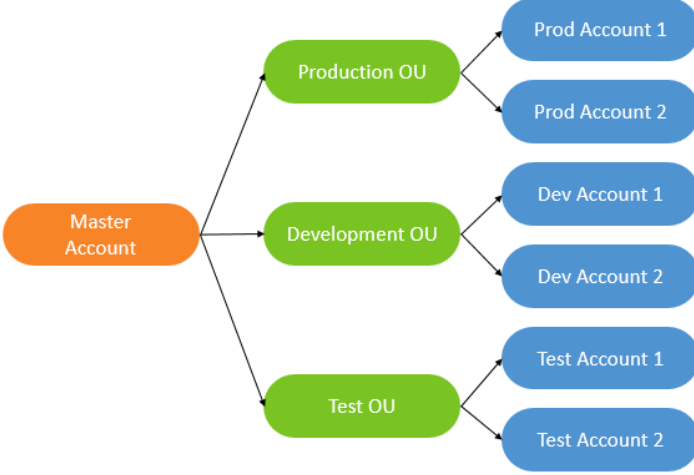


Ejemplos de Unidades Organizativas (UO)

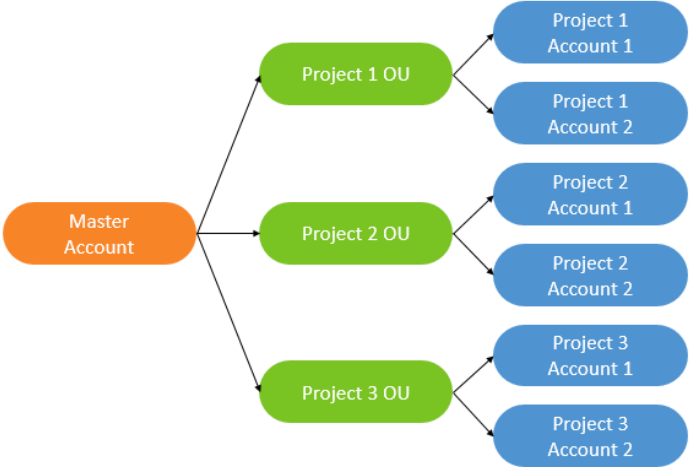
Unidad de negocio



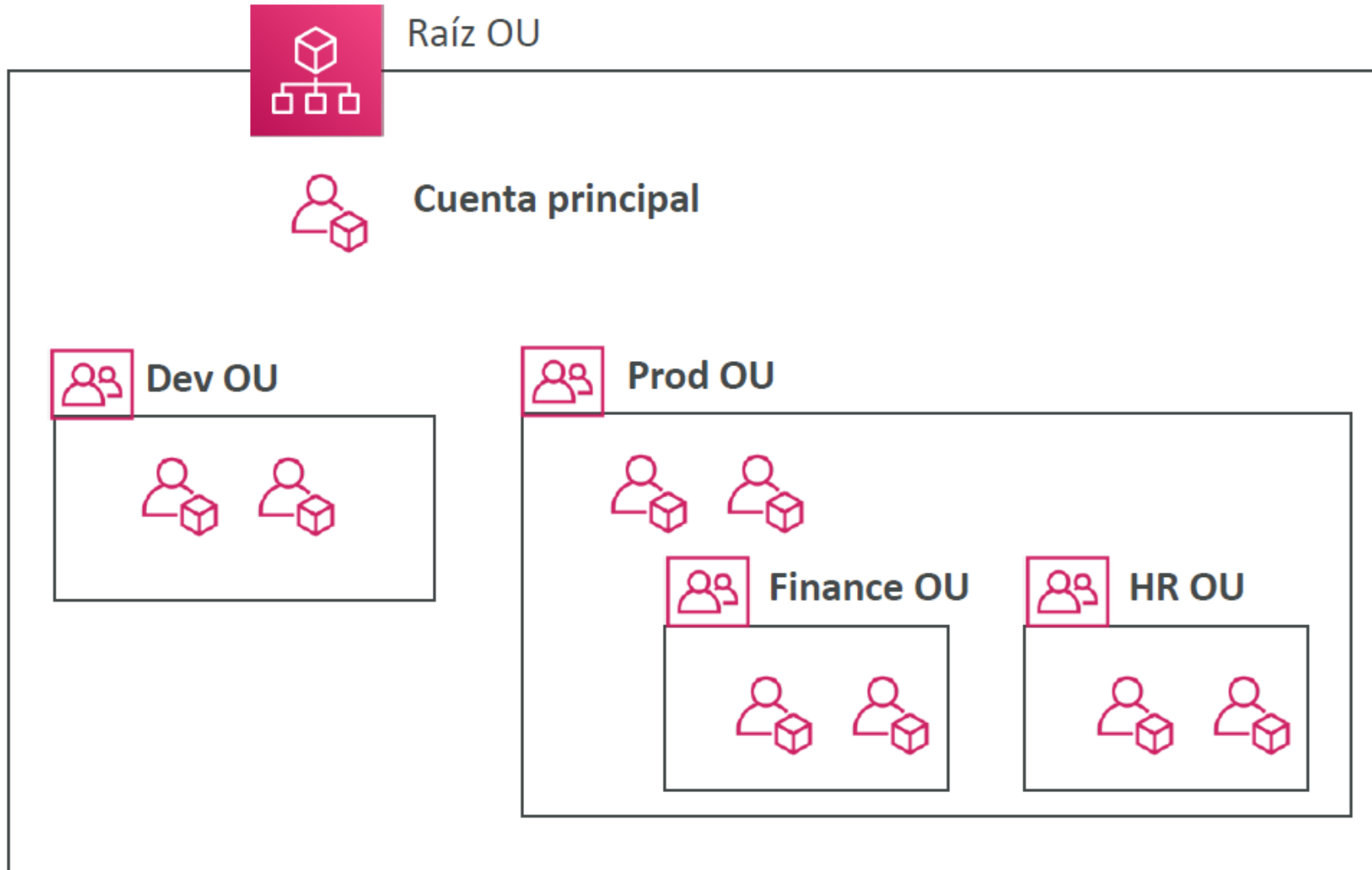
Ciclo de vida



Basado en proyectos

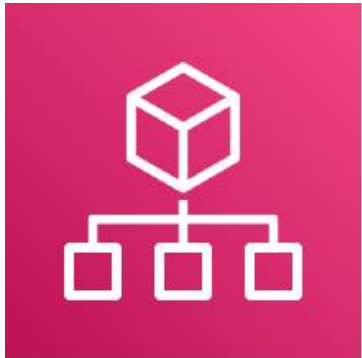


AWS Organization



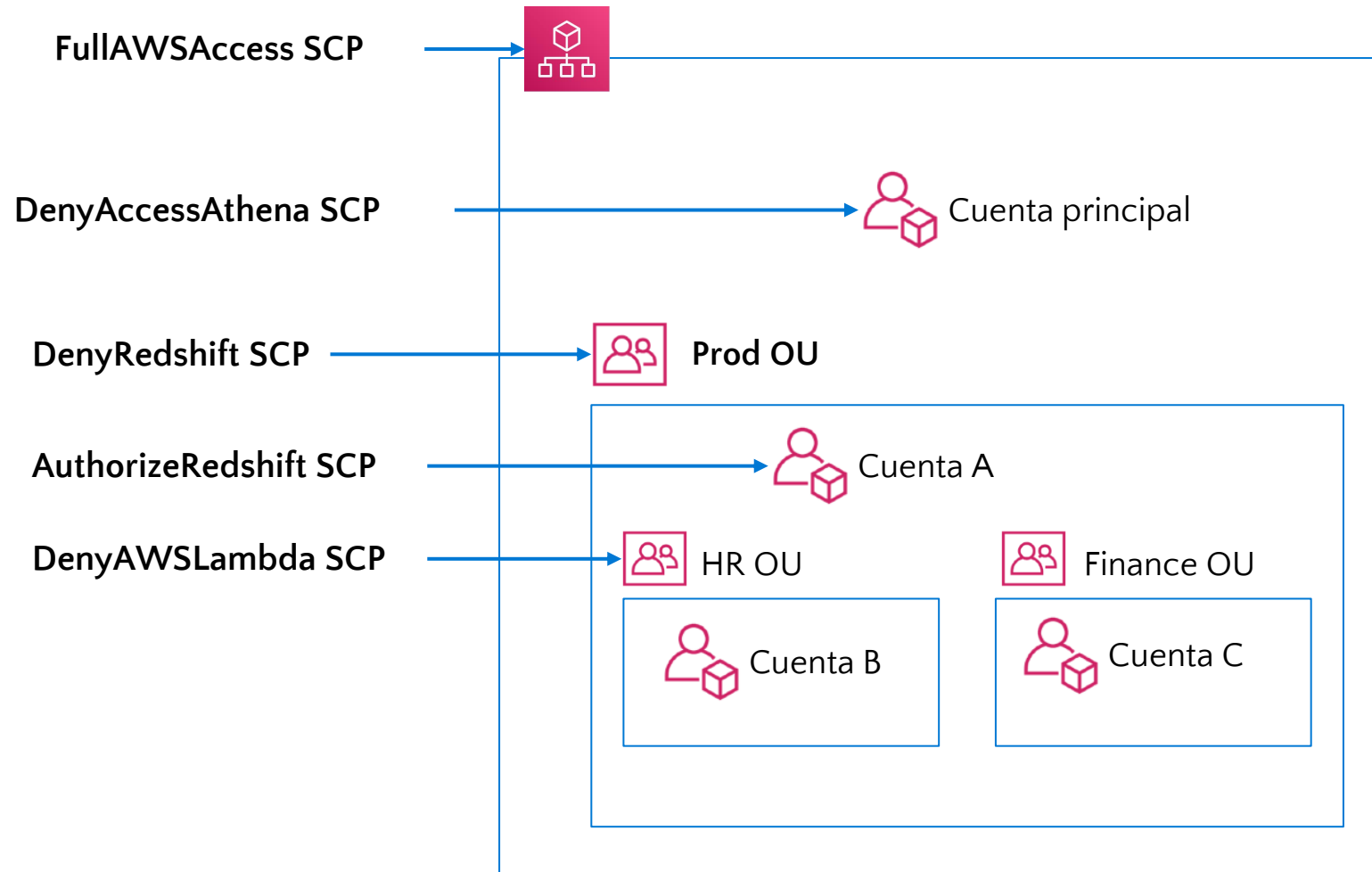
Políticas de control de servicios(SCP)

- Lista blanca o negra de acciones IAM
- Se pueden aplicar a nivel de OU o cuenta AWS
- No se puede aplicar a la cuenta maestra
- La política aplica a todos los usuarios y roles de la cuenta, incluido el usuario root
- La política debe tener un allow explícito



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowsAllActions",  
      "Effect": "Allow",  
      "Action": "*",  
      "Resource": "*"   
    },  
    {  
      "Sid": "DenyDynamoDB",  
      "Effect": "Deny",  
      "Action": "dynamodb:*",  
      "Resource": "*"   
    }   
  ]  
}
```

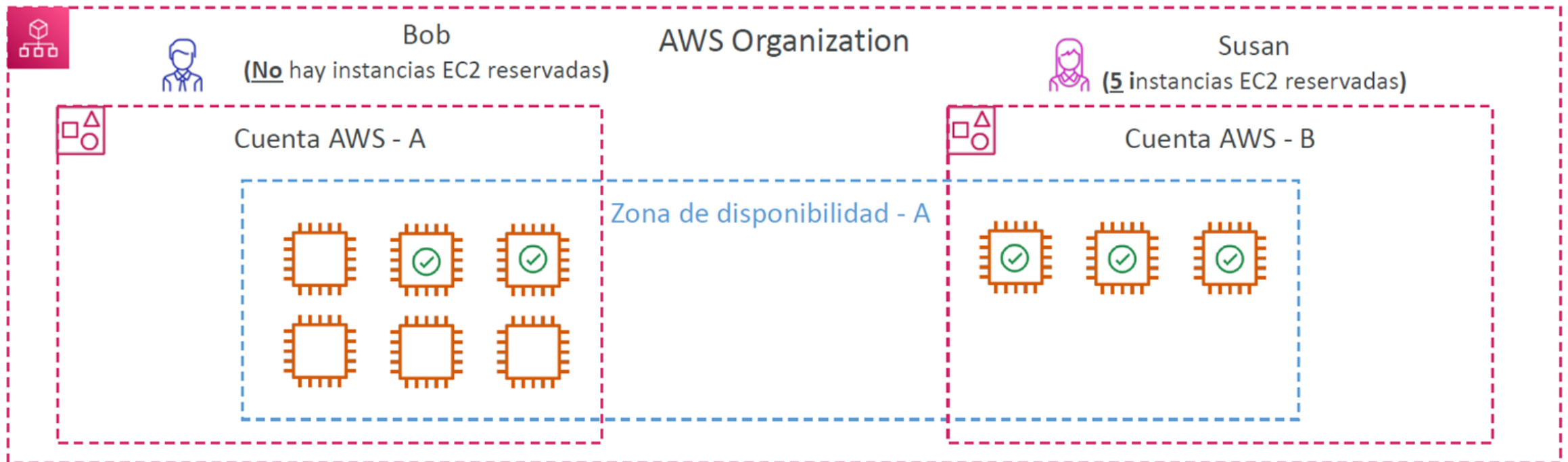
Jerarquía de una política SCP



- **Cuenta maestra**
 - Privilegios máximos
- **Cuenta A**
 - Puede hacer cualquier acción
 - Excepto acceder a redshift (Denegación explícita desde Prod OU)
- **Cuenta B**
 - Puede hacer cualquier acción
 - Excepto acceder a redshift (Denegación explícita desde Prod OU)
 - Excepto acceder al servicio Lambda (Denegación explícita de OU de HR)
- **Cuenta C**
 - Puede hacer cualquier acción
 - Excepto acceder a redshift (Denegación explícita desde Prod OU)

AWS Organization – Facturación consolidada

- Cuando se activa, proporciona:
 - Uso combinado – combina el uso en todas las cuentas AWS en la organización de AWS para compartir los precios por volumen, instancias reservadas y descuentos de planes de ahorro
 - Facturación consolidada: Se obtiene una factura para todas las cuentas de AWS en la organización
- La cuenta maestra puede desactivar el uso compartido de los descuentos de instancias reservadas para cualquier cuenta de la organización de AWS, incluida ella misma.



Administración y gobernanza

AWS Control Tower

AWS Control Tower

- Brinda facilidad para **configurar y gobernar un entorno AWS multicuenta y seguro** conforme a las mejores prácticas
- Ventajas
 - Automatiza la configuración de nuevas cuentas AWS en pocos pasos
 - Automatiza la gestión continua de políticas
 - Detecta infracciones de políticas y las corrige
 - Supervisa el cumplimiento por medio de un dashboard interactivo
- AWS Control Tower se ejecuta sobre organizaciones de AWS.



Administración y gobernanza

AWS Identity and Access Management (IAM)

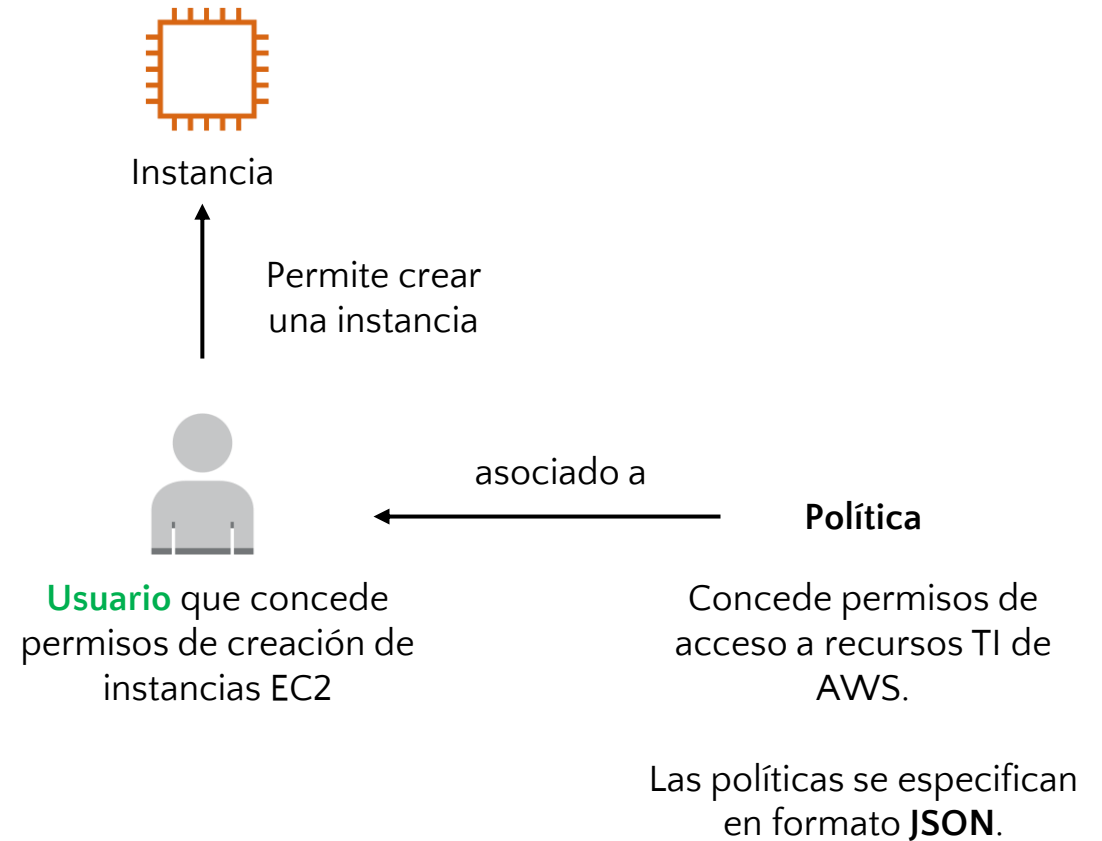
AWS Identity and Access Management (IAM)

¿Qué es?

- Servicio que administra el **acceso** a los recursos de TI de AWS a través de **políticas**.
- Las políticas son posteriormente asociadas a **identidades**:
 - **Usuarios**, grupos de usuarios, roles.



AWS Identity and Access
Management (IAM)



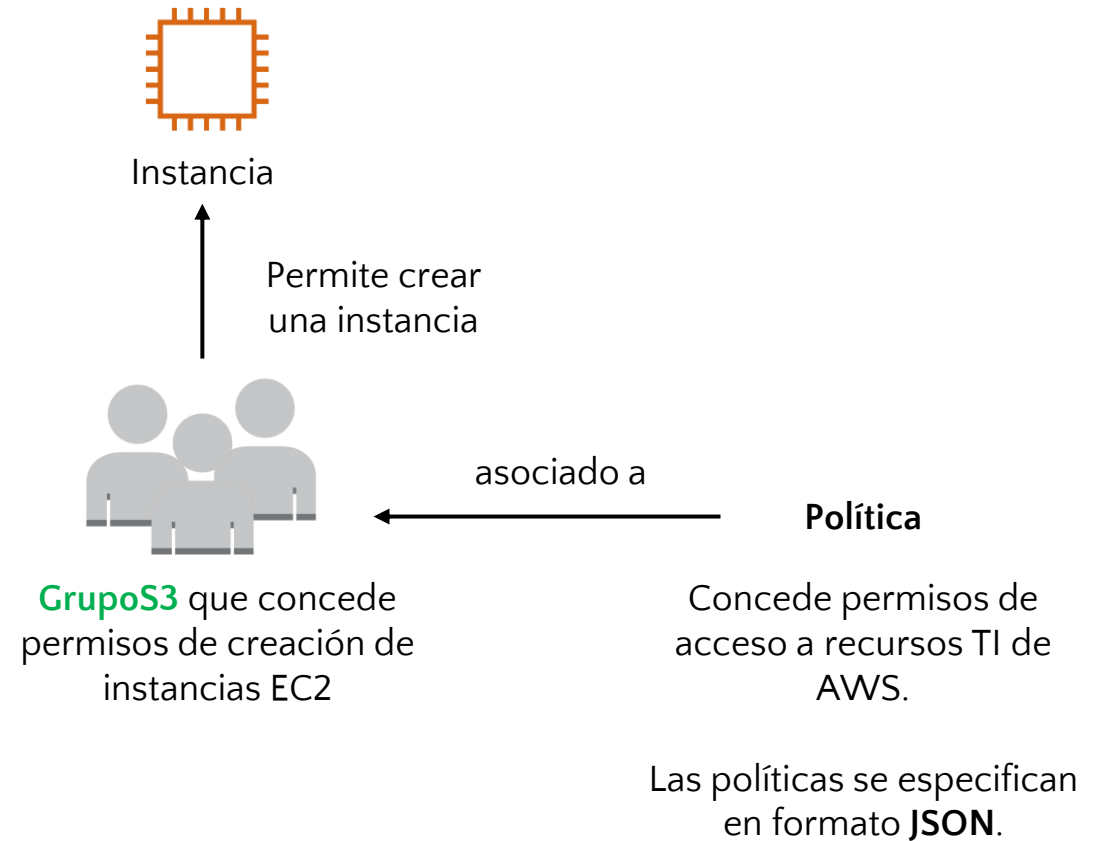
AWS Identity and Access Management (IAM)

¿Qué es?

- Servicio que administra el **acceso** a los recursos de TI de AWS a través de **políticas**.
- Las políticas son posteriormente asociadas a **identidades**:
 - Usuarios, **grupos de usuarios**, roles.



AWS Identity and Access
Management (IAM)



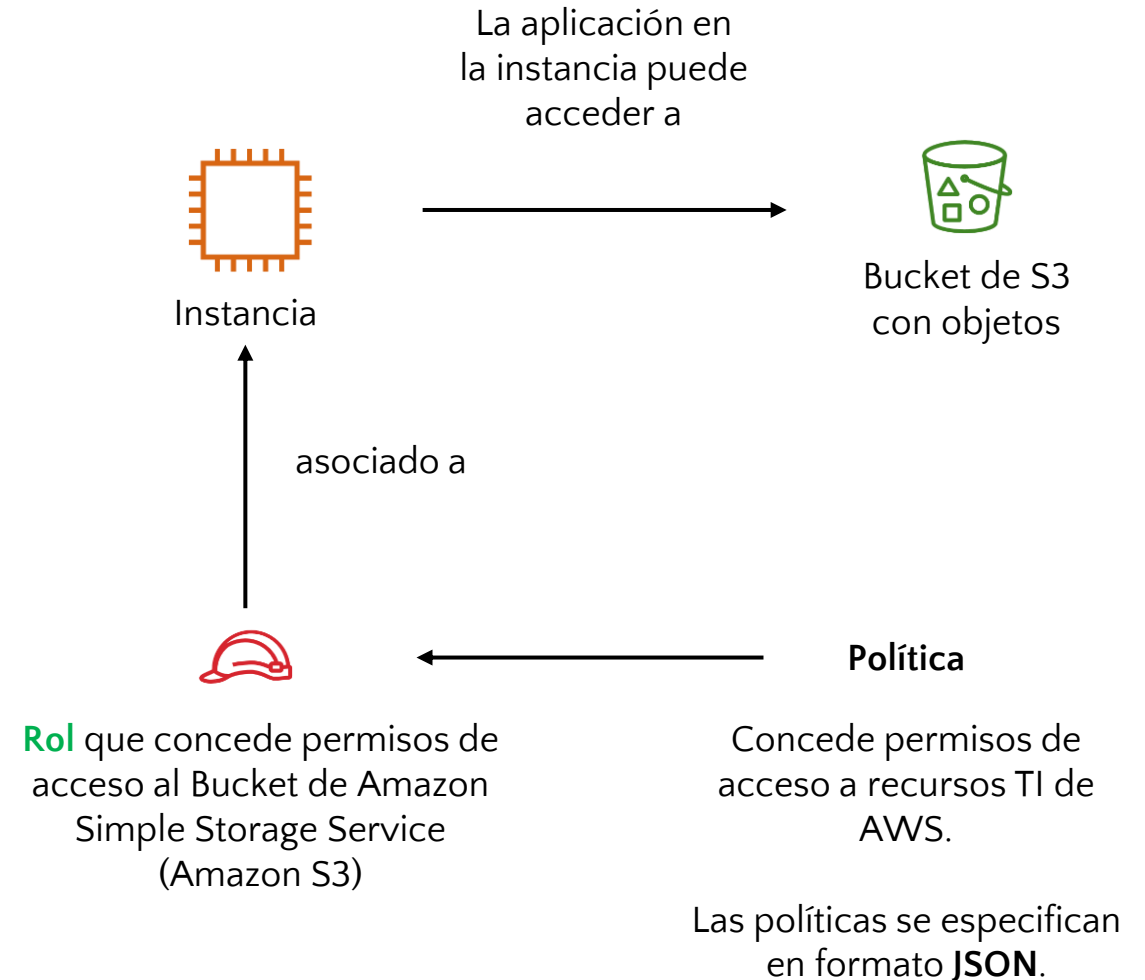
AWS Identity and Access Management (IAM)

¿Qué es?

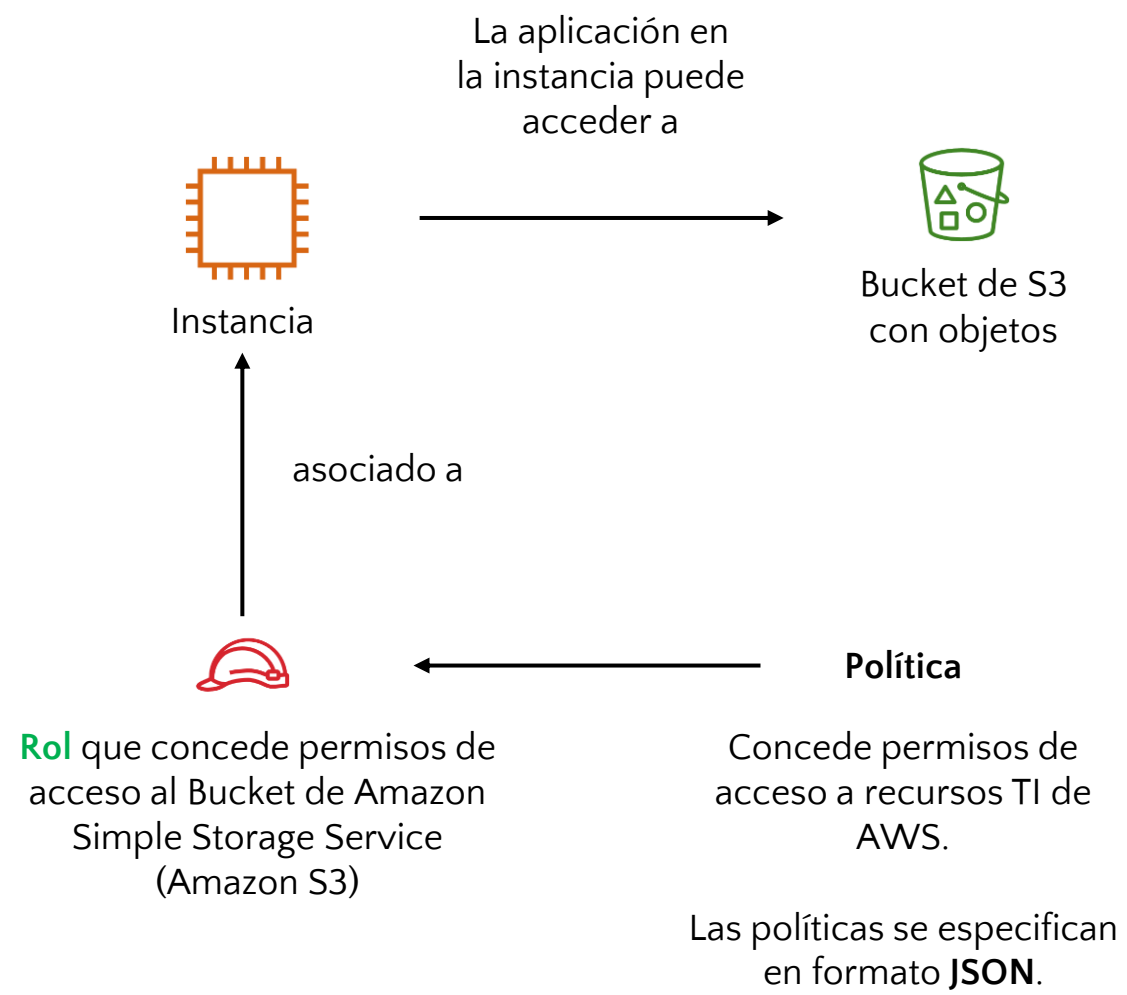
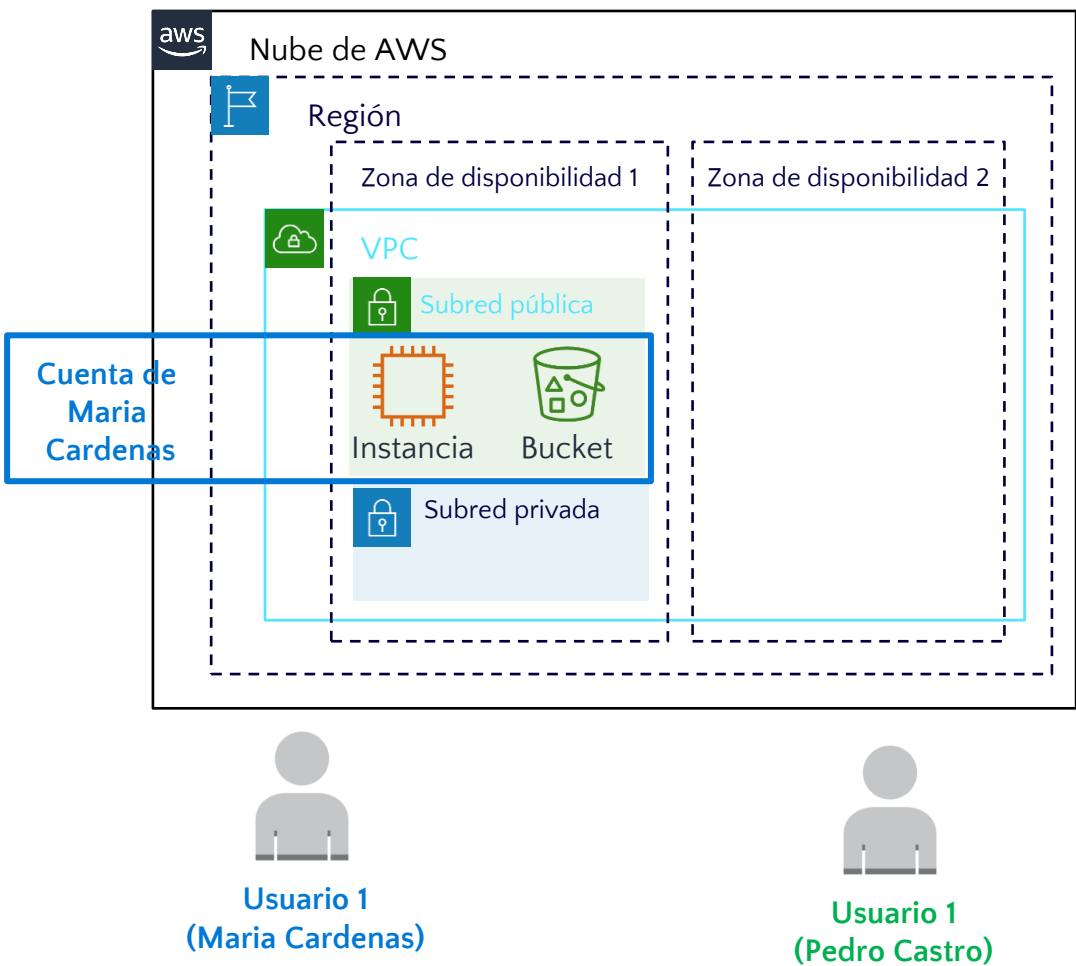
- Servicio que administra el **acceso** a los recursos de TI de AWS a través de **políticas**.
- Las políticas son posteriormente asociadas a **identidades**:
 - Usuarios, grupos de usuarios, **roles**.



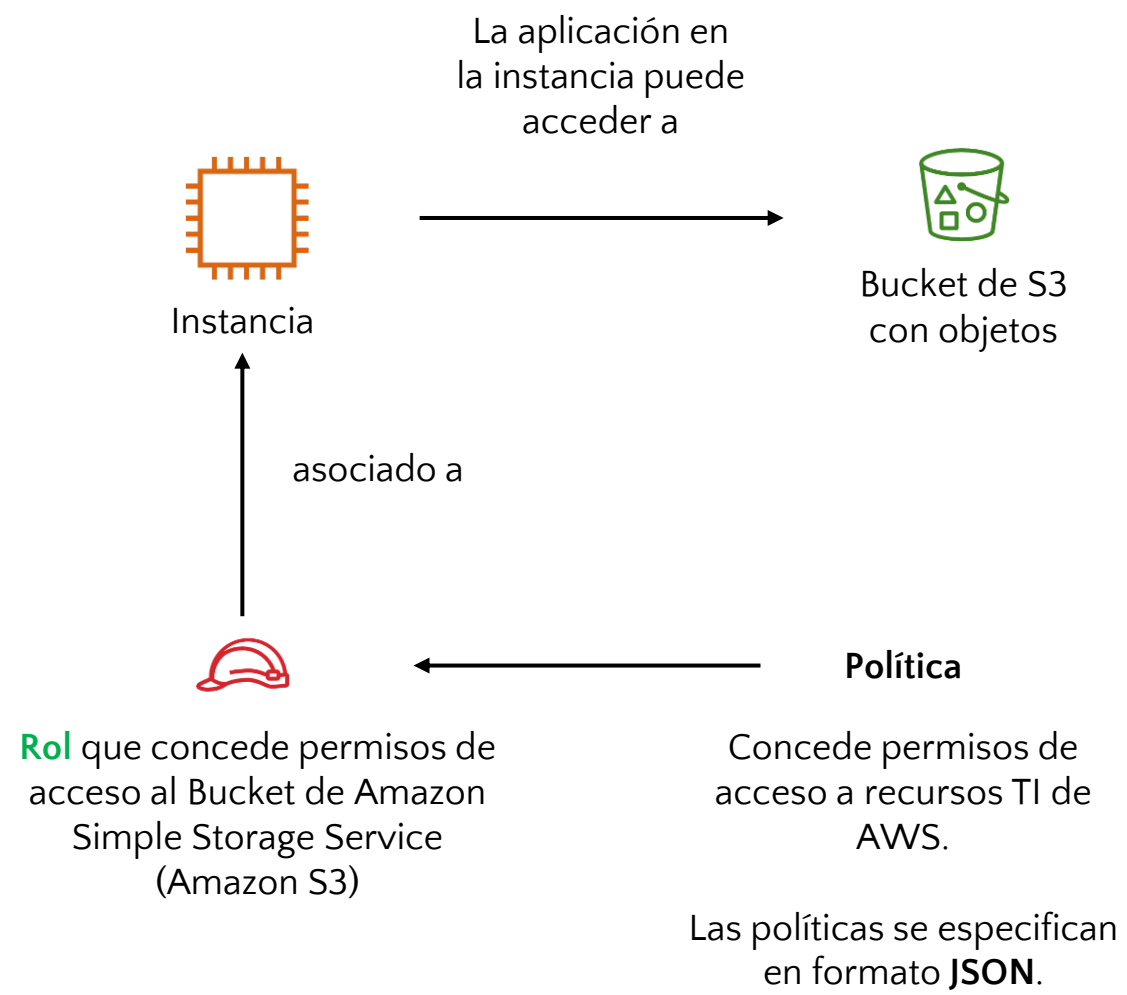
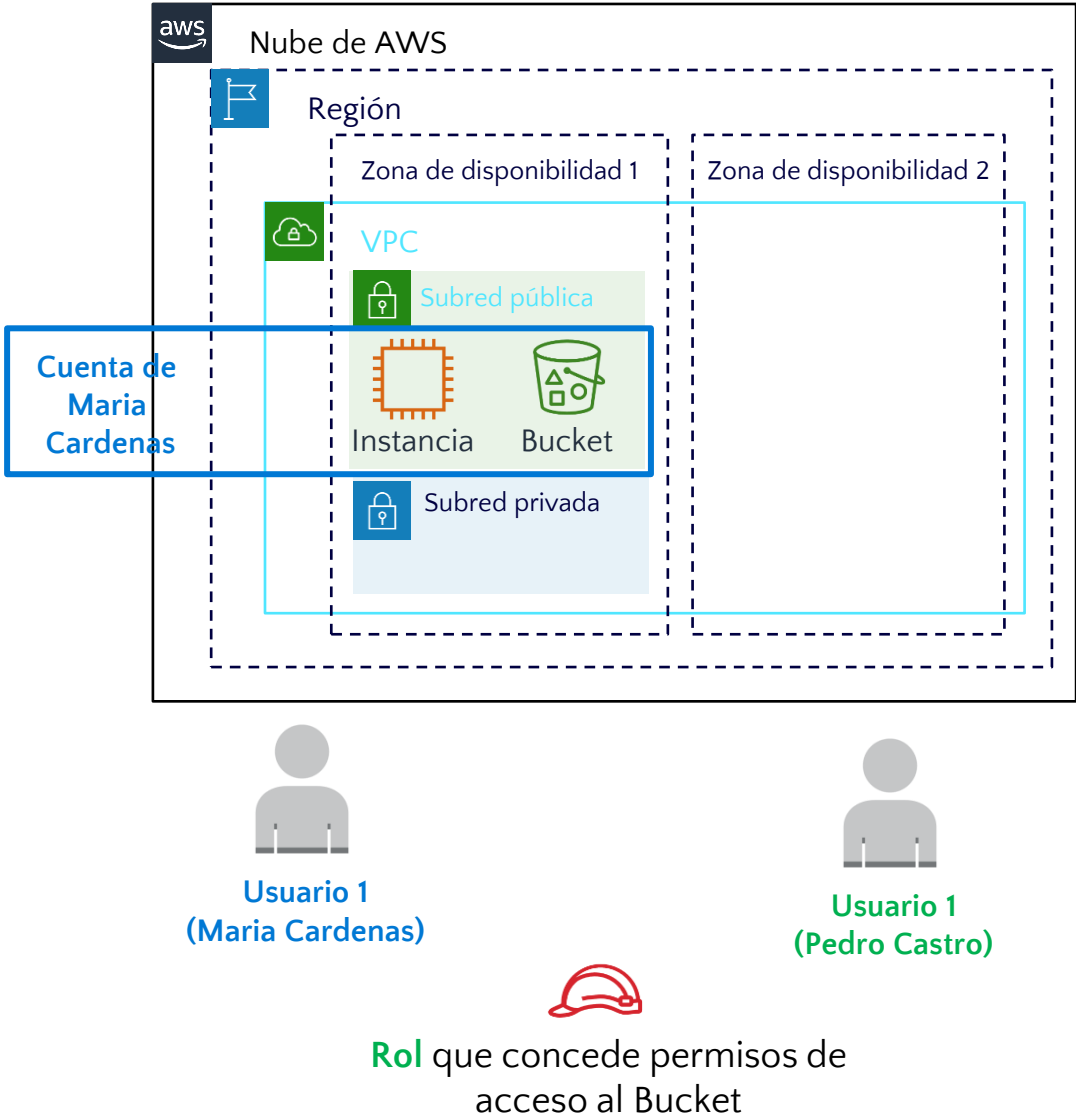
AWS Identity and Access
Management (IAM)



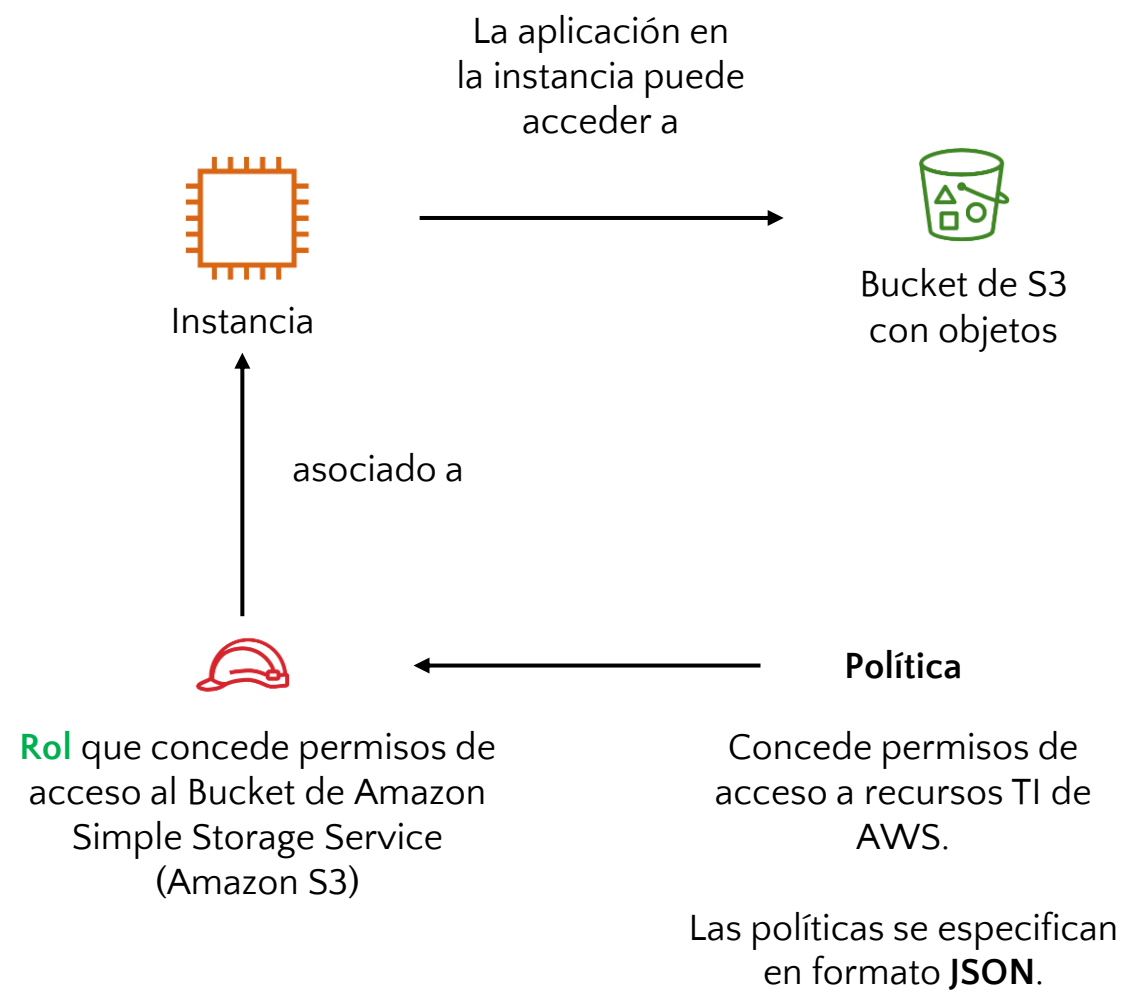
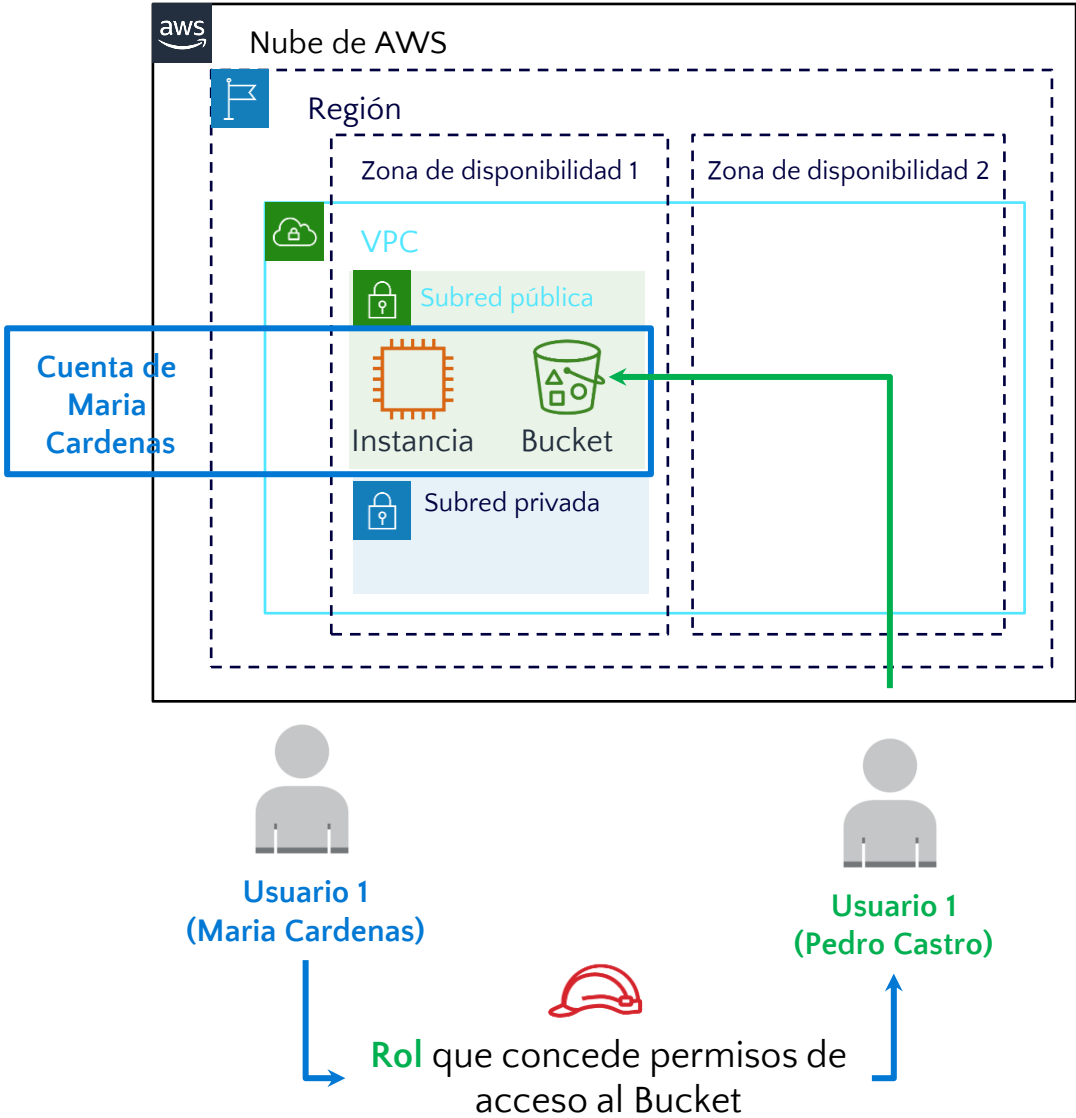
AWS Identity and Access Management (IAM)



AWS Identity and Access Management (IAM)



AWS Identity and Access Management (IAM)



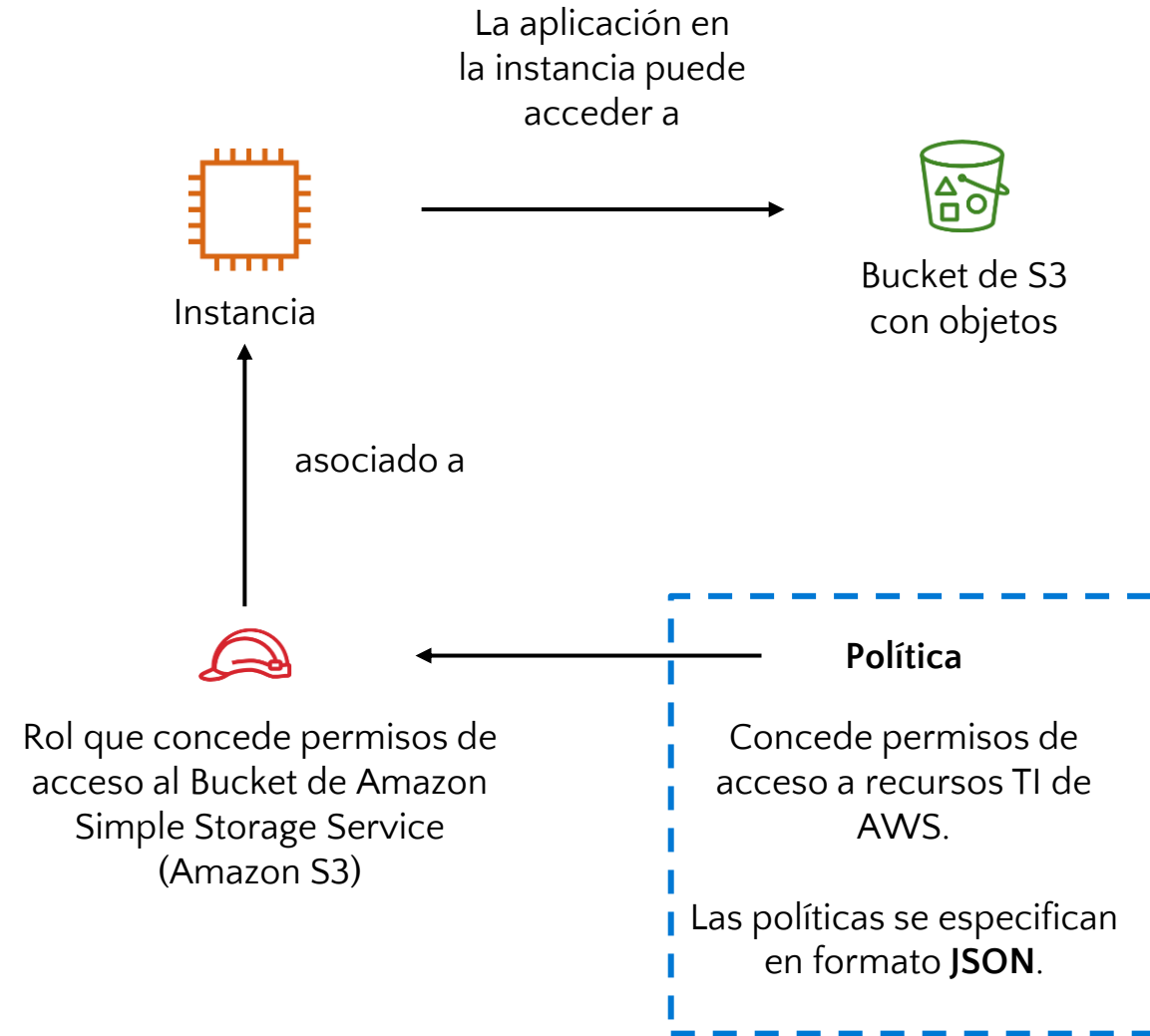
■ Políticas de acceso

¿Qué son las políticas de acceso?

Política de acceso

¿Qué es?

- Un documento que **define permisos**.
- Concede permisos de acceso a recursos TI de AWS.



Política de acceso

¿Cómo se construye una política?

- Las políticas se construyen en base a la **Notación de objetos JavaScript (JSON)**


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject" ],
      "Resource": [ "arn:aws:s3:::example-bucket/*" ]
    }
  ]
}
```

Política de acceso

¿Cómo se construye una política?

- Las políticas se construyen en base a la **Notación de objetos JavaScript (JSON)**

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": [ "s3:GetObject" ],  
      "Resource": [ "arn:aws:s3:::example-bucket/*" ]  
    }  
  ]  
}
```



Efecto:

- Permitir: (Allow)
- Denegar: (Deny)

Política de acceso

¿Cómo se construye una política?

- Las políticas se construyen en base a la **Notación de objetos JavaScript (JSON)**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject" ],
      "Resource": [ "arn:aws:s3:::example-bucket/*" ]
    }
  ]
}
```

Acción:

- **sqs**:SendMessage
- **sqs**:ReceiveMessage
- **ec2**:StartInstances
- **iam**:ChangePassword
- **s3**:GetObject
- **s3**:PutObject
- **s3**:*

Política de acceso

¿Cómo se construye una política?

- Las políticas se construyen en base a la **Notación de objetos JavaScript (JSON)**

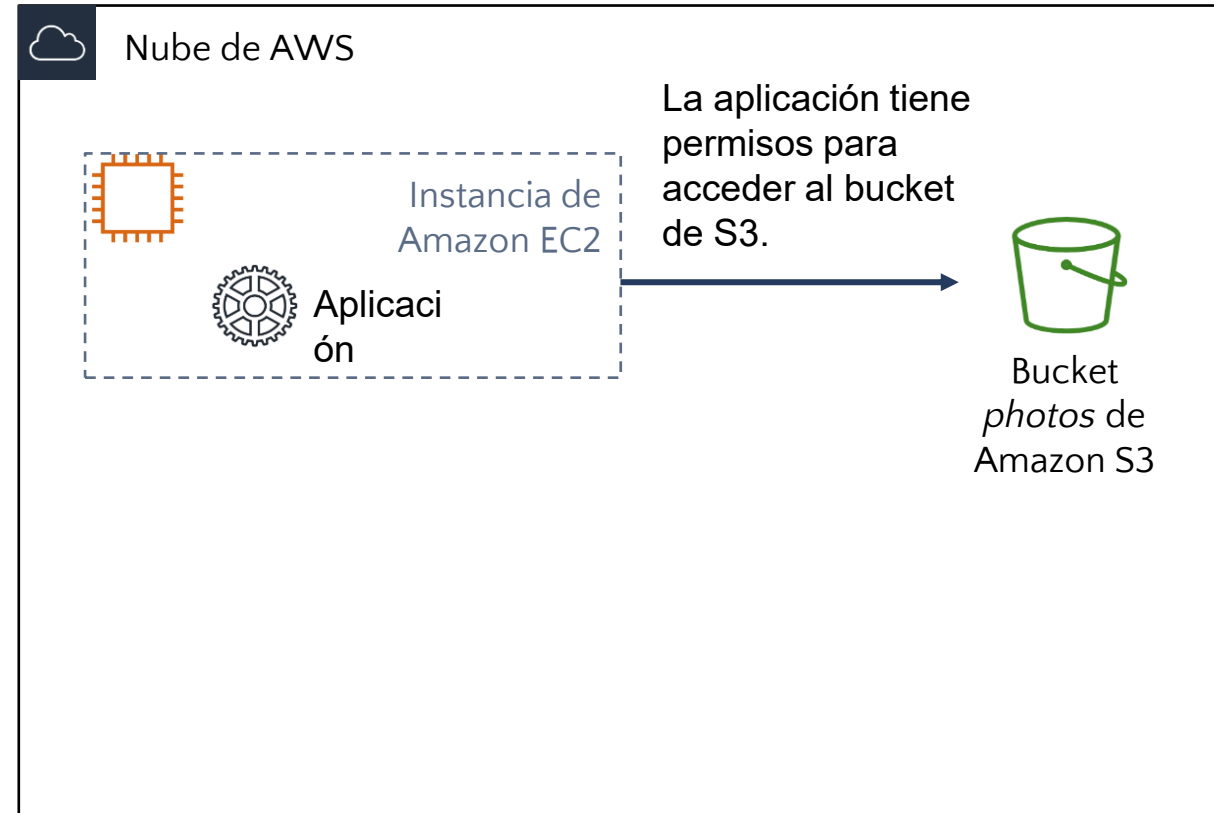
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject" ],
      "Resource": [ "arn:aws:s3:::example-bucket/*" ]
    }
  ]
}
```

Recurso:

- Buckets** de S3
- Objetos** dentro de un Bucket de S3.
- Instancia** de EC2.

Actividad

Situación: Una aplicación que se ejecuta en una instancia EC2 necesita acceso a un bucket de S3.

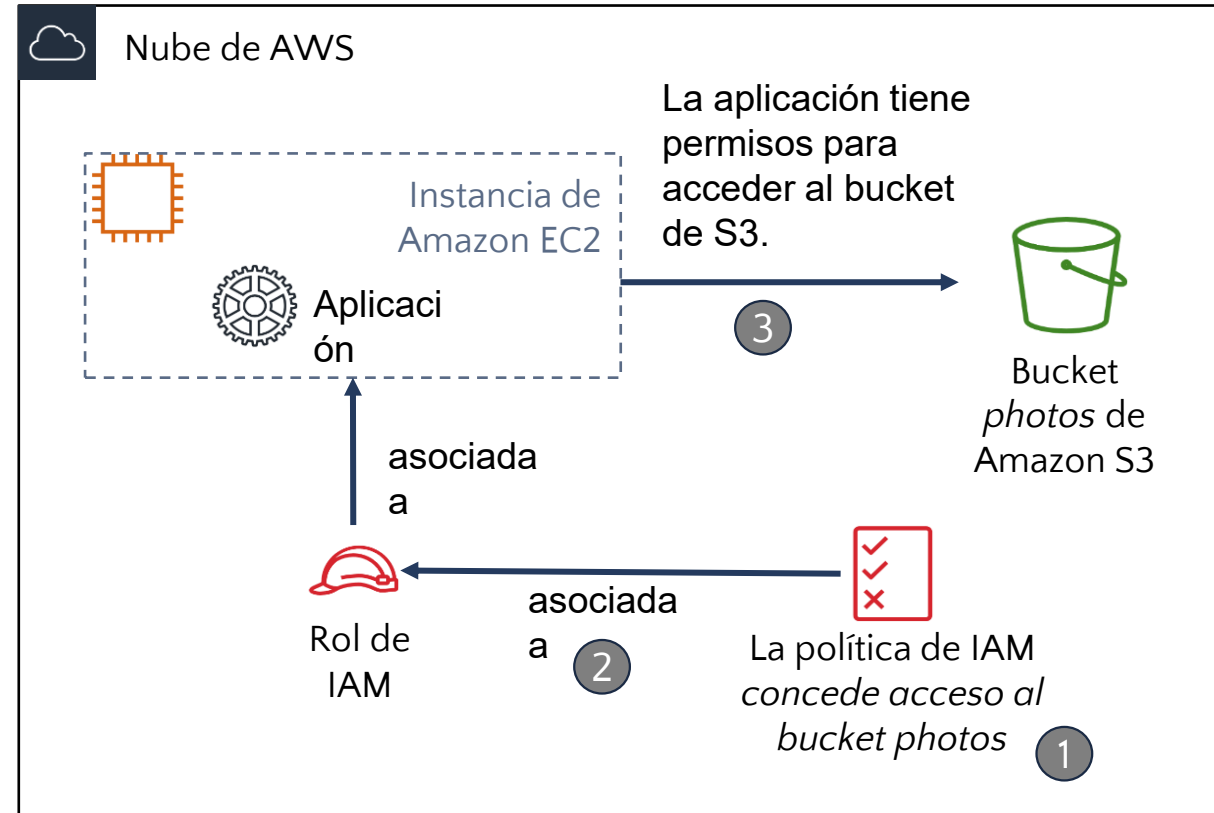


Actividad

Situación: Una aplicación que se ejecuta en una instancia EC2 necesita acceso a un bucket de S3.

Solución:

1. Definir una política de IAM que conceda acceso al bucket de S3
2. Asociar la política a un rol.
3. Permitir que la instancia EC2 asuma el rol.



Conceptos clave



- **Estructura de una cuenta de AWS** usuario raíz, usuarios, grupos de usuarios y roles.
- Las **identidades** permiten identificar **quién** accede al sistema y **a qué recursos** tiene acceso.
- Las identidades proveen **credenciales de acceso** (usuario, contraseña, clave de acceso, etc) a los recursos.
- **AWS IAM** servicio que administra el acceso a los recursos de TI de AWS a través de políticas.

Gracias