TALLER DE ARQUITECTURA SERVELESS EN AWS

ARQUITECTURA DE SOFTWARE

De: Jhon Sebastián Caviedes Guevara

Jhon Sebastián Molina Fierro

Docente: Luis Ángel Vargas Narváez

CORPORACION UNIVARSITARIA DEL HUILA – CORHUILA

1 de octubre de 2025

# Análisis Teórico:

1. ¿ Cómo garantizarías la **disponibilidad del sistema** ante picos de tráfico o fallos en una zona de disponibilidad?

Implementaría una arquitectura Multi-AZ en AWS con Auto Scaling Group (ASG) y Application Load Balancer (ALB) para mantener el sistema activo ante picos de tráfico o fallos. Usar RDS en modo Multi-AZ para base de datos con failover automático, ElactiCache para mejorar el rendimiento y CloudFront para entregar contenido rápido. Monitorear con CloudWatch y usar Route 53 para redirigir tráfico en caso de caída.

1. ¿ Qué tipo de **escabilidad** aplicarías y cómo se coordina con el balanceo de carga?

Podría aplicar escabilidad horizontal mediante Auto Scaling, que añade o quita servidores según la demanda. El ALB distribuye el tráfico entre las instancias y mantiene solo las saludables. En la base de datos, usar read replicas o Aurora para lecturas. Si es posible, incluir componentes serveless (Lambda + API Gateway) para escalar automáticamente sin gestión de servidores.

1. ¿ Qué **configuraciones IAM** definirás para separar accesos entre personal clínico, técnico y administrativo?

Definiría los roles según el tipo de usuario:

* **Clínico:** acceso solo a la app o API, sin permisos directos a AWS
* **Administrativo:** solo lectura de informes o datos anonimizados.
* **Técnico:** gestión de infraestructura (EC2, RDS, etc.) sin acceso a datos sensibles, con MFA y roles temporales. Usar políticas IAM, S3 Access points y CloudTrail para registrar y controlar todos los accesos.

1. ¿ Cómo se reflejarían los principios de **seguridad, privacidad y trazabilidad** en la arquitectura propuesta?

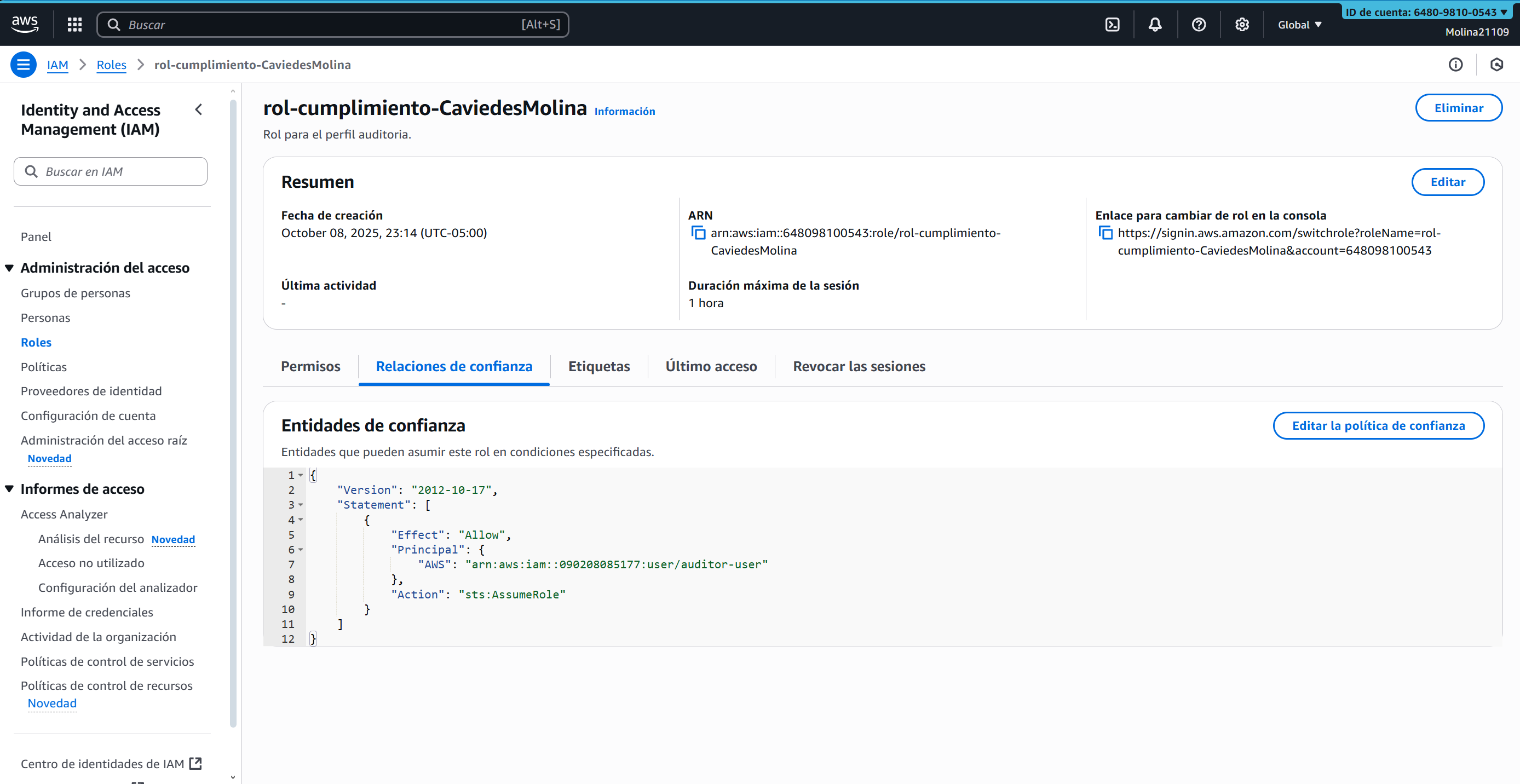
Cifrar los datos en tránsito (TLS) y en reposo (KMS). Usar WAF, Shield, Security Groups y NACLs para protección. Aplicar privacidad por diseño y pseudonimización en datos médicos, con políticas de retención en S3. Para trazabilidad, habilitar CloudTrail, Config y VPC Flow Logs para registrar todas las acciones y cumplir con la Ley 1581 de 2012.

DESARROLLO – PARTE 2

Se configuró una relación de confianza entre dos cuentas AWS con el objetivo de que el usuario auditor-user (cuenta 090208085177) pueda asumir el rol rol-cumplimiento-CaviedesMolina ubicado en la cuenta 648098100543.

1. **Configuración de la Relación de Confianza**

En la cuenta del rol (648098100543), se editó la política de confianza del rol rol-cumplimiento-CaviedesMolina para permitir el acceso desde el usuario externo:



1. **Política de Permisos del Usuario Auditor**

En la cuenta del auditor (090208085177), se creó y asignó la siguiente política:

Nombre: policy-assume-rol-cumplimiento



Esto habilita al usuario auditor-user para asumir el rol definido en la otra cuenta.

1. **Asunción del Rol desde la CLI**

*aws sts assume-role --role-arn arn:aws:iam::648098100543:role/rol-cumplimiento-CaviedesMolina --role-session-name auditor-prueba2*

**Resultado exitoso:**

*{*

*"Credentials": {*

*"AccessKeyId": "ASIA...",*

*"SecretAccessKey": "e1gY7wFaxkvmkKXBKkcbvK7pe...",*

*"SessionToken": "IQoJb3JpZ2luX2VjEDUaCXVzLWVhc3QtMSJG...",*

*"Expiration": "2025-10-09T06:01:42+00:00"*

*},*

*"AssumedRoleUser": {*

*"AssumedRoleId": "AROAZNZNIGU75DMRFK5PT:auditor-prueba1",*

*"Arn": "arn:aws:sts::648098100543:assumed-role/rol-cumplimiento-CaviedesMolina/auditor-prueba1"*

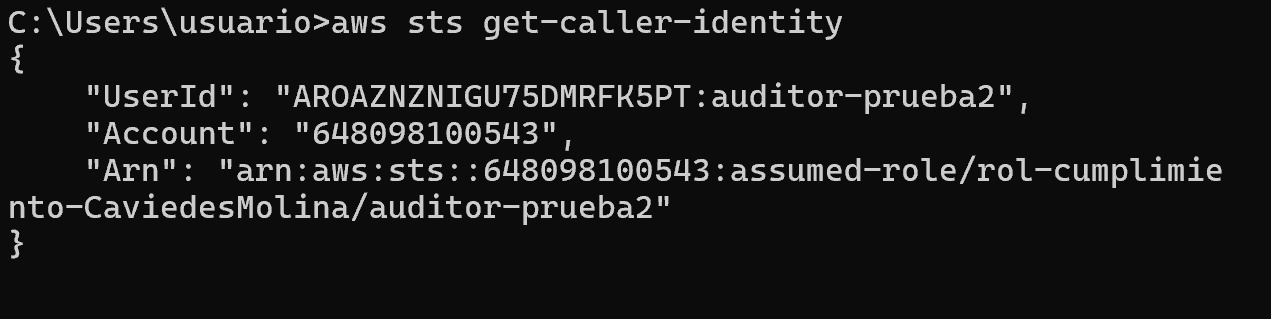
*}*

*}*

El usuario auditor-user ejecutó el siguiente comando desde su cuenta:

1. **Verificación de Identidad**

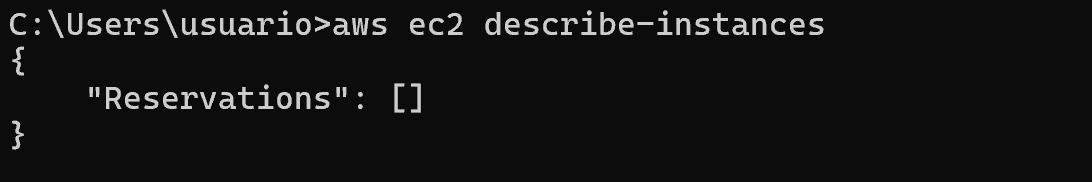
Luego de asumir el rol, se verificó que el usuario ahora pertenece a la cuenta del rol:



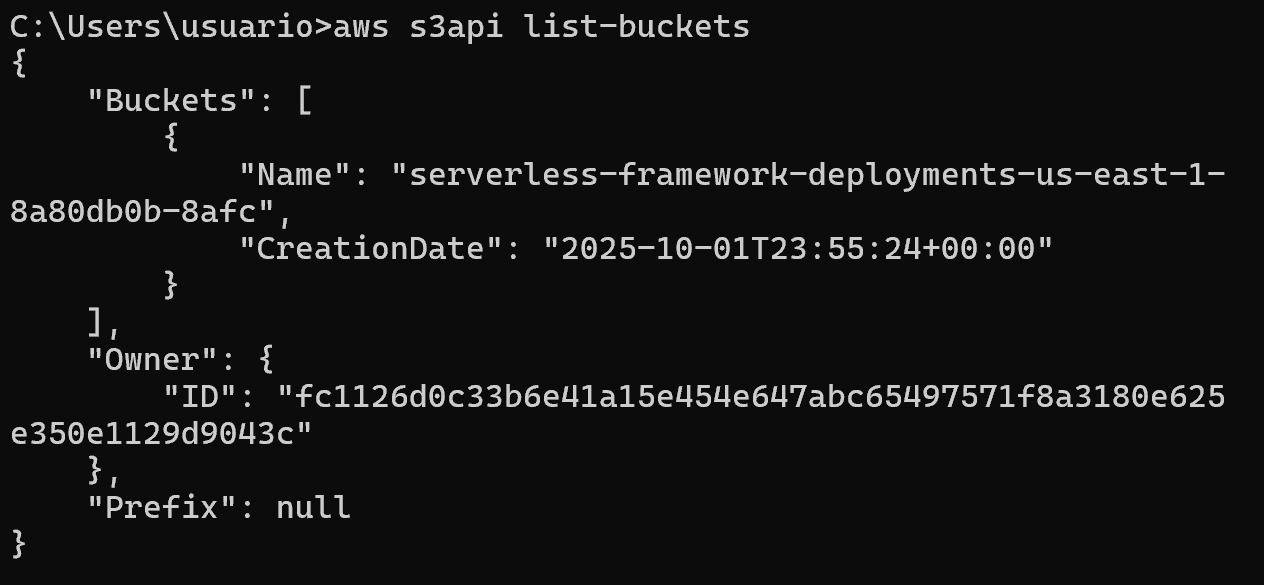
1. **Pruebas de Acceso a Recursos**

Con el rol asumido, el auditor realizó consultas sobre recursos de la cuenta del rol:

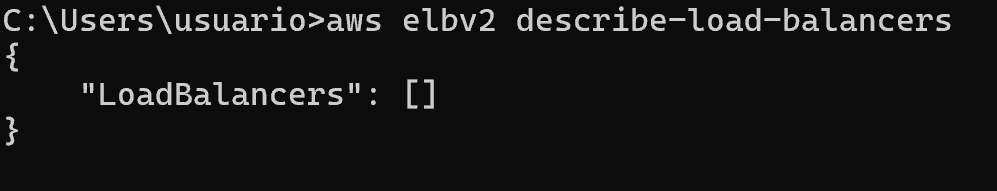
**EC2:**



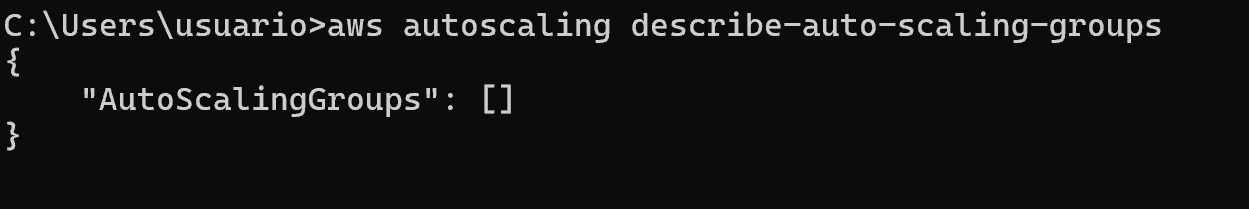
**S3:**



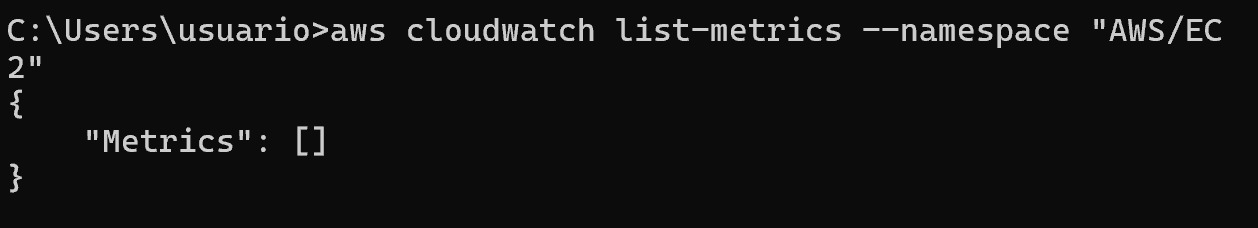
**ELB:**



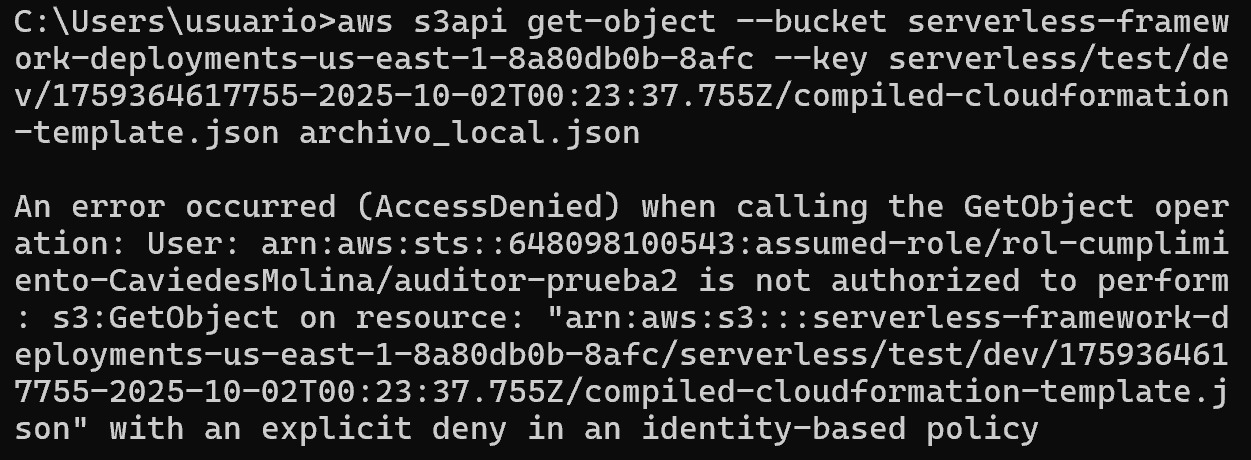
**ASG:**



**CloudWatch:**



1. **Prueba de acceso denegado**



1. **Descripción de Servicios Accesibles**

Durante la sesión asumida con el rol rol-cumplimiento-CaviedesMolina, el usuario auditor-user tuvo acceso de solo lectura a los siguientes servicios de la cuenta destino (648098100543):

| **Servicio AWS** | **Acción Permitida** | **Descripción del Acceso** |
| --- | --- | --- |
| **Amazon EC2** | ec2:DescribeInstances | Permite visualizar información sobre las instancias EC2 desplegadas, sin capacidad de modificarlas ni iniciar/detener instancias. |
| **Amazon S3** | s3:ListAllMyBuckets, s3:ListBucket, s3:GetObject (limitado) | Permite listar los buckets y objetos existentes. La lectura directa de archivos (GetObject) está restringida por políticas de confidencialidad. |
| **Elastic Load Balancer (ELBv2)** | elasticloadbalancing:DescribeLoadBalancers | Permite visualizar los balanceadores de carga existentes, sin permisos de modificación. |
| **Auto Scaling** | autoscaling:DescribeAutoScalingGroups | Permite consultar grupos de Auto Scaling para monitoreo y auditoría. |
| **CloudWatch** | cloudwatch:ListMetrics | Permite listar métricas de monitoreo, sin acceder a datos detallados de logs ni editar alarmas. |

Todos los accesos fueron de tipo lectura y observación (read-only), lo cual cumple el objetivo de auditoría sin comprometer la integridad de los recursos.

**Propósito del Rol rol-cumplimiento-CaviedesMolina**

El propósito del rol es facilitar auditorías de cumplimiento y verificación de configuración entre cuentas AWS, sin otorgar privilegios administrativos ni de modificación.

Este rol está diseñado para:

1. Permitir a auditores externos (como auditor-user) realizar verificaciones sobre la infraestructura y los recursos de la cuenta destino.
2. Garantizar el cumplimiento de políticas de seguridad, revisando el estado de recursos (instancias, buckets, balanceadores, etc.).
3. Evitar accesos indebidos, limitando las acciones del auditor únicamente a operaciones de inspección.

**Políticas Aplicadas al Rol**

**Política de Permisos del Rol (rol-cumplimiento-molina-policy)**

Ejemplo de las acciones permitidas:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "EC2ReadOnly",

"Effect": "Allow",

"Action": [

"ec2:Describe\*"

],

"Resource": "\*"

},

{

"Sid": "ELBReadOnly",

"Effect": "Allow",

"Action": [

"elasticloadbalancing:Describe\*"

],

"Resource": "\*"

},

{

"Sid": "ASGReadOnly",

"Effect": "Allow",

"Action": [

"autoscaling:Describe\*"

],

"Resource": "\*"

},

{

"Sid": "CloudWatchReadOnly",

"Effect": "Allow",

"Action": [

"cloudwatch:Describe\*",

"cloudwatch:ListMetrics",

"cloudwatch:GetMetricData",

"cloudwatch:GetMetricStatistics"

],

"Resource": "\*"

},

{

"Sid": "CloudTrailReadOnly",

"Effect": "Allow",

"Action": [

"cloudtrail:DescribeTrails",

"cloudtrail:GetTrailStatus",

"cloudtrail:ListTrails"

],

"Resource": "\*"

},

{

"Sid": "LogsListOnly",

"Effect": "Allow",

"Action": [

"logs:Describe\*"

],

"Resource": "\*"

},

{

"Sid": "S3ListOnly",

"Effect": "Allow",

"Action": [

"s3:ListAllMyBuckets",

"s3:GetBucketLocation",

"s3:ListBucket"

],

"Resource": "arn:aws:s3:::\*"

},

{

"Sid": "DenySensitiveAccess",

"Effect": "Deny",

"Action": [

"s3:GetObject",

"s3:GetObjectAcl",

"s3:GetObjectVersion",

"s3:PutObject",

"s3:DeleteObject",

"s3:PutObjectAcl",

"logs:GetLogEvents",

"logs:FilterLogEvents",

"iam:\*",

"rds:\*",

"aws-portal:\*",

"pricing:\*"

],

"Resource": "\*"

},

{

"Sid": "DenyWriteOrModify",

"Effect": "Deny",

"Action": [

"ec2:RunInstances",

"ec2:StartInstances",

"ec2:StopInstances",

"ec2:TerminateInstances",

"autoscaling:CreateAutoScalingGroup",

"autoscaling:UpdateAutoScalingGroup",

"autoscaling:DeleteAutoScalingGroup",

"elasticloadbalancing:RegisterTargets",

"elasticloadbalancing:DeregisterTargets",

"elasticloadbalancing:CreateLoadBalancer",

"elasticloadbalancing:DeleteLoadBalancer",

"cloudwatch:PutMetricData",

"cloudwatch:PutDashboard",

"cloudwatch:DeleteAlarms",

"cloudwatch:SetAlarmState"

],

"Resource": "\*"

}

]

}

Esta política garantiza que el rol solo pueda visualizar información, sin posibilidad de alterarla, eliminarla o crear nuevos recursos.

**Justificación Técnica**

**Principio de Confidencialidad**

El diseño de este rol aplica el principio de confidencialidad al:

* Restringir la lectura de datos sensibles, permitiendo solo listar o describir recursos sin exponer contenido completo de objetos S3.
* Evitar operaciones de escritura o eliminación, previniendo fugas o alteraciones de información.
* Controlar la identidad de quien asume el rol, asegurando que únicamente el usuario auditor-user autorizado desde una cuenta específica pueda acceder mediante sts:AssumeRole.

**Principio de Mínimo Privilegio**

El principio de mínimo privilegio se cumple al:

* Otorgar exclusivamente los permisos necesarios para la tarea de auditoría (acciones Describe y List).
* Evitar incluir permisos amplios como \*:\*, PutObject, RunInstances o Delete\*.
* Permitir solo un tiempo de sesión temporal mediante STS (Security Token Service), eliminando el acceso permanente y reduciendo el riesgo de abuso.

Este enfoque minimiza el impacto de posibles errores o accesos indebidos, al limitar las capacidades del auditor estrictamente a las que necesita para cumplir su función.