

AWS Cloud Hardening & Monitoring (Free Tier Edition)

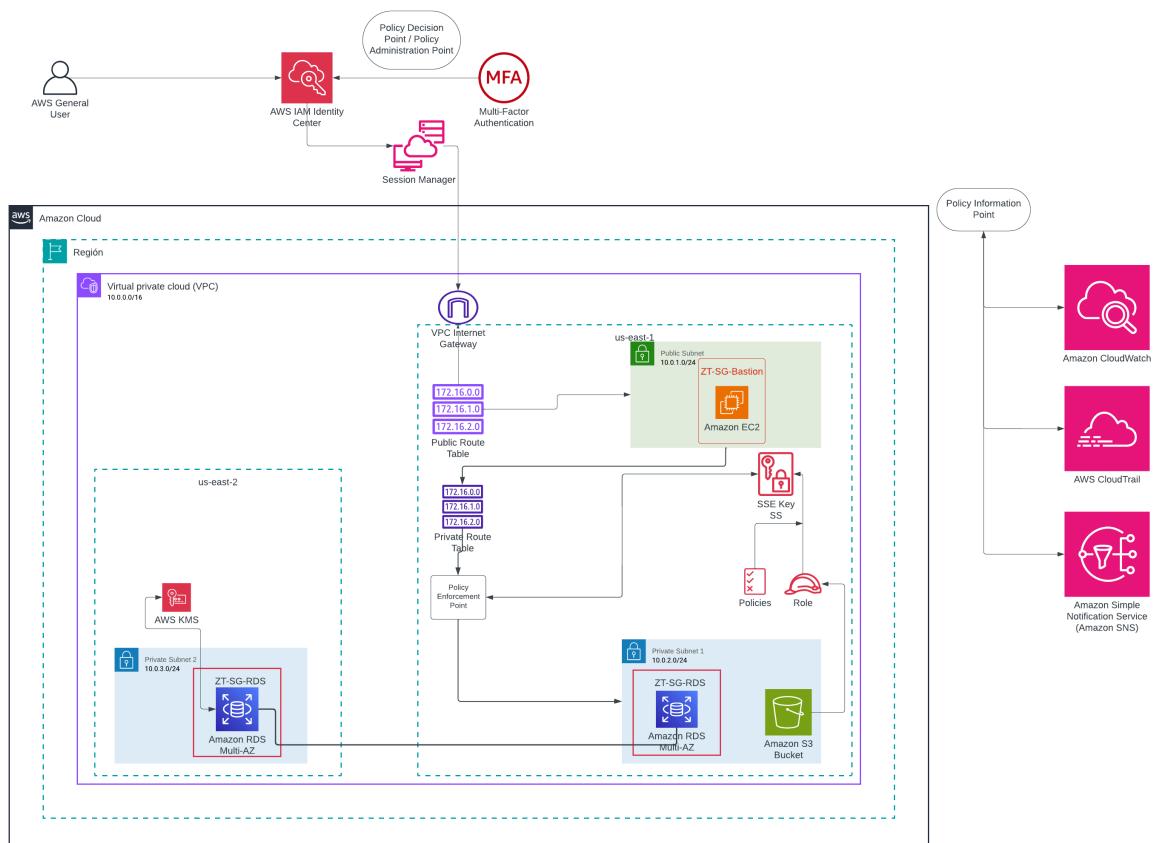
Por: Cristian Jiménez

Objetivos

Implementar una arquitectura segura en AWS aplicando los **principios de Zero Trust Architecture (ZTA)** según el marco **NIST SP 800-207**, utilizando exclusivamente servicios disponibles en el **Free Tier de AWS**. El Proyecto demuestra cómo aplicar controles de acceso estrictos, segmentación de red, cifrado y monitoreo continuo sin depender de servicios premium como GuardDuty o Security Hub.

[Expandir un poco mas los objetivos y si es posible agregar una introducción]

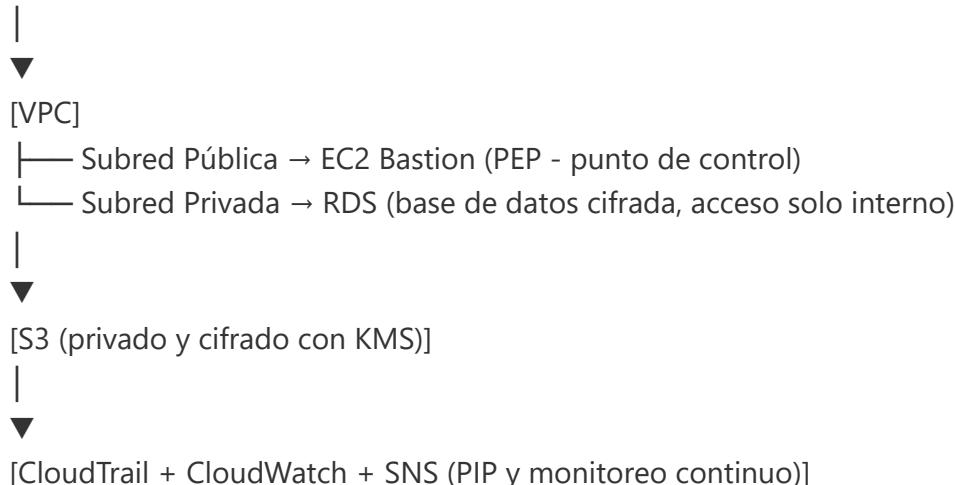
Arquitectura General



[Usuario / Administrador]



[IAM + MFA (PAP & PDP)]



Servicios utilizados

Servicio	Rol / Propósito
IAM	Control de identidades y políticas (PAP / PDP).
S3	Almacenamiento seguro con cifrado y acceso controlado.
KMS	Cifrado de datos en reposo para S3, RDS y EBS.
VPC	Microsegmentación de red (Zonas de confianza reducida).
EC2 (Bastion)	Punto de Acceso seguro a recursos internos (PEP).
RDS	Base de datos privada cifrada (recurso protegido)
CloudTrail	Auditoría de acciones (PIP: información de políticas).
CloudWatch	Detección de anomalías y generación de alertas
SNS	Notificaciones automáticas de seguridad.

Pasos de creación del Entorno

1 Configuración de seguridad de identidad (IAM)

 Objetivo

Configurar el entorno inicial con **Identity Center (antes AWS SSO)** para manejar acceso seguro basado en identidad y MFA.

Esta fase establece el **PDP (Policy Decision Point)** y **PAP (Policy Administration Point)** del modelo Zero Trust de nuestro Proyecto.

- Habilita **MFA (Multi-Factor Authentication)** para la cuenta root.

The screenshot shows the 'Multi-factor authentication (MFA)' configuration page. It includes a table with columns for Type (Virtual), Identifier, Certifications, and Created on (Sun Sep 08 2024). Buttons for Remove, Resync, and Assign MFA device are at the top right.

Type	Identifier	Certifications	Created on
Virtual		Not Applicable	Sun Sep 08 2024

- Paso 1: Crear un usuario administrador separado en Identity Center.
1. En el panel lateral, seleccionamos **Users** → **Add user**.
 2. Introducimos los datos:
 - a. User Name: el de nuestra preferencia
 - b. Email Address: en mi caso utilicé mi correo personal (para recibir invitaciones de inicio de sesión)
 - c. Password Options, aquí se dejó la opción "Send an email to the user to set their password".
 3. (Opcional) Podemos asignar este usuario a un grupo llamado ZT-Admins.
 4. Click en **Add User**.

A green confirmation box displays: **✓ The user "ICDD" was successfully added.** It explains that the user will receive an email with a link to set up a password and instructions to connect to the AWS access portal. The link will be valid for up to 7 days. You can grant this user permissions to accounts or applications so that they can access their assigned AWS accounts and cloud applications when they sign in to the AWS access portal.

The 'Users' list page shows one user entry:

Username	Display name	Status	MFA
ICDD	Admin ZT	Enabled	None

Resultado esperado:

El usuario recibe un **correo con un enlace de activación** para iniciar sesión en el portal Identity Center.

no-reply@login.awsapps.com
para mí ▾

2:30 (hace 1 minuto) ☆ ☺ ↶ ⋮



Hello Admin ZT,

Your administrator for AWS Account #948531372386 has invited you to AWS IAM Identity Center. Accepting this invitation activates your user account in IAM Identity Center so that you can access assigned AWS resources. Choose the link below to accept this invitation.

[Accept invitation](#)

This invitation will expire in 7 days.

Accessing the AWS access portal

After you've accepted the invitation, you can sign in to the AWS access portal by using the information below.

Your AWS access portal URL:

<https://d->

Your Username:

ICDD

Una vez se acepta la invitación, el mismo portal de AWS le pedirá al nuevo usuario registrar un dispositivo MFA:



Registrar dispositivo MFA

Nombre de usuario:

ICDD ([¿No es usted?](#))

La organización requiere la autenticación multifactor (MFA) para mayor seguridad durante el inicio de sesión. Cada vez que inicie sesión, se le pedirá que indique su contraseña y un dispositivo MFA. [Más información](#)

Seleccione una de las siguientes opciones para comenzar:



Aplicación de autenticación

Autenticar con un código generado por una aplicación instalada en su dispositivo móvil o equipo.



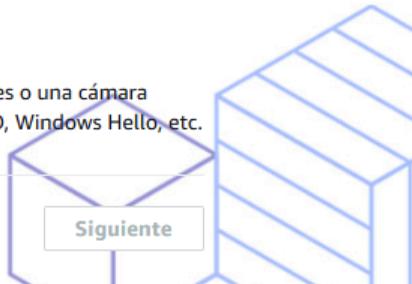
Clave de seguridad

Para autenticar, pulse una llave de seguridad de hardware como YubiKey, Feitian, etc.



Autenticador integrado

Autenticar con un escáner de huellas digitales o una cámara integrada en su equipo, como Apple TouchID, Windows Hello, etc.



En este caso se usará la Aplicación de autenticación.

- Luego de seleccionarlo debemos tener instalada una aplicación de autenticación (Ejemplo: Authy, Duo Mobile o Google Authenticator).
- Luego escaneamos el código QR que nos aparece.
- Una vez escaneado colocamos el código de seis dígitos de la aplicación de autenticación.

Una vez seguidos los pasos mencionados anteriormente deberíamos poder tener acceso a la consola AWS



Aplicación de autenticación registrado

✓ Su aplicación de autenticación se ha registrado correctamente. Ahora puede utilizarlo cuando se le solicite verificación adicional al iniciar sesión.

ICDD's MFA 1 Cambiar nombre

Tipo y descripción: Aplicación de autenticación

Listo

The screenshot shows a dark-themed AWS Identity Center interface. At the top, it says "aws portal de acceso". On the right, there are navigation icons and a "Admin" dropdown. In the center, a green box contains the message: "Su aplicación de autenticación se ha registrado correctamente. Ahora puede utilizarlo cuando se le solicite verificación adicional al iniciar sesión." Below this, there's a section titled "ICDD's MFA 1 Cambiar nombre" with a "Listo" button. At the bottom, a message says "Accedió al portal de acceso de AWS" and "Una vez que el administrador le conceda acceso a las aplicaciones y las cuentas de AWS, podrá encontrarlas aquí."

- Paso 2: Asignar permisos al usuario
- 1. En el panel de Identity Center, seleccionamos **AWS Accounts → Assign users or groups.**
- 2. Escogemos nuestra cuenta AWS.
- 3. Hacemos click en Assign users, y seleccionamos el usuario que recién creamos.

AWS accounts

2

Organization o-egn2c6ef5v

Select one or more AWS accounts in your organization to provide multi-account access to users and groups in IAM Identity Center. [Learn more](#)

Search by name, email, account ID or OU ID.

Hierarchy | **List**

Organizational structure | Permission sets

▼ Root
r-4ewu

1 ►
 CI
948531372386 | cristian.jimenez2@utp.ac.pa

Select users and groups

Assign users and groups to "Cristian Jimenez"

Select one or more users or groups in IAM Identity Center that you want to give multi-account access to.

1.

Users Groups

Users (1/1)

Find by: Username ▼

Search users in IAM Identity Center by username or display name

2

<input checked="" type="checkbox"/> Username	Display name	Status
<input checked="" type="checkbox"/> ICDD	Admin ZT	<input checked="" type="checkbox"/> Enabled

► Selected users and groups (Users: 1) 3

4. Elegimos un **permission set**:

- Seleccionamos "AdministratorAccess" (Se puede crear un conjunto nuevo si se quiere probar personalización).

Select permission set type

A permission set contains policies that determine a user's permissions to access an AWS account. When you assign a user or group to a permission set in an AWS account, IAM Identity Center creates an IAM role in the account and attaches the policies specified in the permission set to that role. Select an option to specify the permission set type. [Learn more](#)

Permission set type

Types

Predefined permission set
Create a predefined permission set by choosing an AWS-defined template. This template enables you to select a single AWS managed policy. For example, you can select a policy that grants permissions for a common job function, such as Billing, or a specific level of access to AWS services and resources, such as ViewOnlyAccess. You can update the permission set as your needs evolve.

Custom permission set
Create a custom permission set by selecting AWS managed policies and creating an inline policy (recommended). You can also attach customer managed policies and set a permissions boundary (advanced).

Review and submit

Review and submit assignments to "Cristian Jimenez"

Step 1: Select users and groups [Edit](#)

Users and groups (1)

Display name / group name	Type
ICDD	User

Step 2: Select permission sets [Edit](#)

Permission sets (1)

Permission set	Description	ARN	Create time
AdministratorAccess	-	arn:aws:sso:::permissionSet/ssoins-72235c15d4dc8783/ps	Now b54be2765fde8a25

[Cancel](#) [Previous](#) [Submit](#)

5. Confirmamos la asignación.

Resultado esperado:

El usuario tiene permisos de administrador dentro de la cuenta, pero autenticación controlada por Identity Center (sin credenciales permanentes).

- Paso 3: Habilitamos MFA (Multi-Factor Authentication)
 1. En el panel lateral de **IAM Identity Center**, entramos a **Settings**→ **Multi-factor authentication**.

The screenshot shows the IAM Identity Center interface. On the left, there's a sidebar with 'Managing instance' at the top, followed by 'Dashboard', 'Users', 'Groups', 'Settings' (which is selected), 'Multi-account permissions' (with 'AWS accounts' and 'Permission sets'), and 'Application assignments' (with 'Applications'). Below these are 'Related consoles' for 'CloudTrail' (Recommended) and 'AWS Organizations'. The main content area has tabs for 'Identity source', 'Authentication' (which is selected), 'Management', and 'Tags'. Under 'Authentication', there are two sections: 'Standard authentication' (with a 'Configure' button) and 'Multi-factor authentication' (also with a 'Configure' button). The 'Multi-factor authentication' section contains four items: 'Prompt users for MFA' (set to 'Only when their sign-in context changes (context-aware)'), 'When prompted for MFA' (describing authenticators like Authenticator apps, security keys, and built-in authenticators), 'If this user does not have a registered MFA device' (requiring them to register an MFA device at sign in), and 'Who can manage MFA devices' (allowing users and administrators to manage MFA devices).

2. Clickeamos en Configure, y una vez dentro seleccionamos la casilla que dice **Every time they sign in (always on)**

MFA Settings

Prompt users for MFA

- Only when their sign-in context changes (context-aware)

Users with a registered MFA device are only prompted when their sign-in context changes (for example, they sign in from a new device or browser, or from an unknown IP address). Users can remember devices when this mode is selected.

- Every time they sign in (always-on)

Users with a registered MFA device are prompted every time they sign in.

- Never (disabled)

All users sign in with their standard user name and password only. Choosing this option disables MFA.

Users can authenticate with these MFA types

- Security keys and built-in authenticators

Users can verify their identity by using any FIDO2 or U2F capable device such as an external physical security key (for example, YubiKey or Feitian devices) or a built-in authenticator (for example, Apple TouchID or Windows Hello).

- Authenticator apps

Users can verify their identity by entering a code generated from a time-based one-time password authenticator app (for example, Authy, Google Authenticator, Microsoft Authenticator).

If a user does not yet have a registered MFA device

- Require them to register an MFA device at sign in
- Require them to provide a one-time password sent by email to sign in
- Block their sign-in
- Allow them to sign in

Who can manage MFA devices

- Users can add and manage their own MFA devices

1.

2.

Cancel

Save changes

3. Guardamos los cambios.

 *Resultado esperado:*

Cada inicio de sesión al portal de AWS requerirá MFA, siguiendo uno de los principios fundamentales de Zero Trust “continuous verification”.

- Paso 4: Probar el inicio de sesión del nuevo usuario

1. Vamos al **User Portal URL** (Identity Center muestra algo como [https\[:\]//d-xxxxxx.awsapps\[.\]com/start](https://d-xxxxxx.awsapps.com/start)

The screenshot shows the IAM Identity Center dashboard. A green notification bar at the top right says "Multi-factor authentication settings have been updated." The main area has several sections: "Central management" (with a "Prevent account instances" button), "Monitor activities in your instances of IAM Identity Center" (with a "Learn about monitoring IAM Identity Center" link), "IAM Identity Center setup" (with a "Confirm your identity source" section and a "Confirm identity source" button), and "Settings summary" (with fields for "Instance name", "Identity source", "Region", and "Organization ID"). A red box highlights the "AWS access portal URL - Edit" field, which contains the URL <https://d-533c15d4dc8783.ssoinsservice.identitycenter.amazonaws.com>. The bottom right corner shows copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

2. Iniciamos sesión con el correo y la contraseña creados anteriormente.
3. Colocamos el código de 6 dígitos generado en nuestra app de autenticación.
 - Nota: Si llegas a este punto y aún no has iniciado sesión en la cuenta de administrador que acabamos de crear, entonces tienes que configurarla desde 0, crearle una contraseña y hacer set up del MFA.
4. Una vez dentro, veremos un panel con nuestros accesos asignados: "AWS Account - Administrator Access".

The screenshot shows the AWS Access Portal. At the top, there's a green notification bar with the message "Las preferencias se han actualizado correctamente." Below it, the title "AWS access portal" is displayed. There are two tabs: "Accounts" (which is selected) and "Applications". Under the "Accounts" tab, there's a section titled "AWS accounts (1)". It shows a list with one item: "AdministratorAccess". To the right of this item is a "Create shortcut" button. Below the list is a search bar with the placeholder "Filter accounts by name, ID, or email address". At the bottom of the list, there are two buttons: "AdministratorAccess" and "Access keys". The "Access keys" button is highlighted with a red box. The top right corner of the screen shows the user "Admin" and some other interface elements.

The screenshot shows the AWS Home Page. At the top right, there is a red box highlighting the account information: "ID de cuenta: 911111111111" and "AdministratorAccess/ICDD". Below this, the page displays various sections: "Visitados recientemente", "Aplicaciones (0)", "Le damos la bienvenida a AWS", "AWS Health", and "Costo y uso".

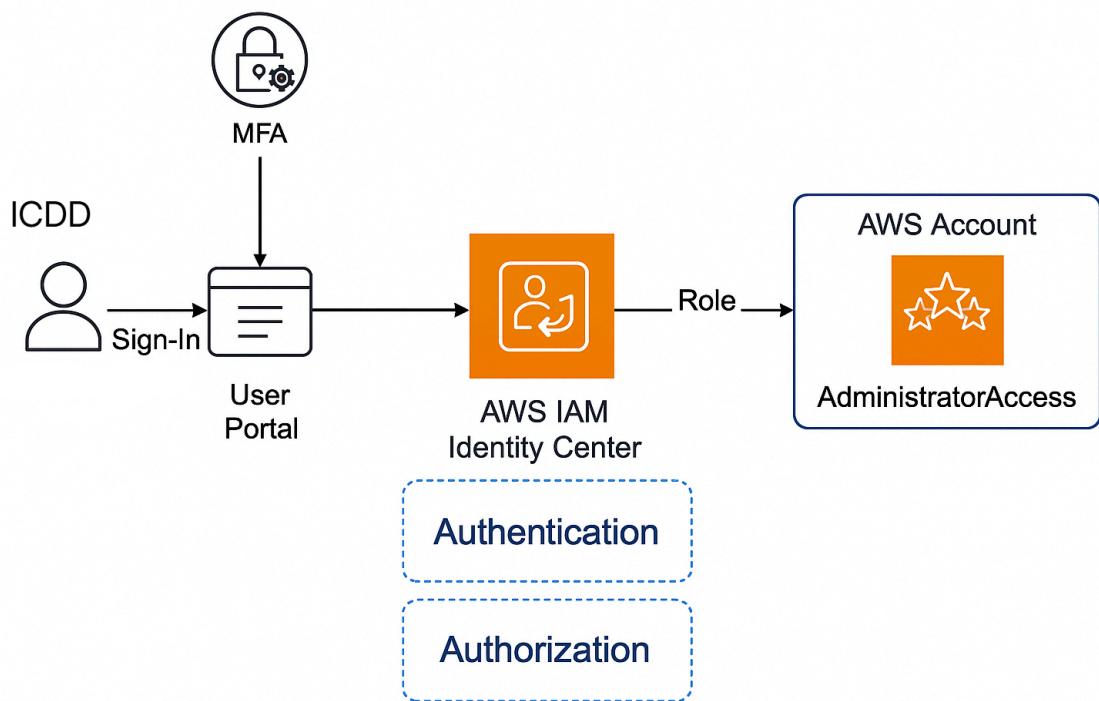
Nota: Asegurarnos de estar en la us-east-1 o la de tu preferencia.

Resultado esperado:

El usuario ICDD inicia sesión mediante MFA y accede a la consola sin usar claves permanentes.

Resultados finales de la Fase 1

Elemento configurado	Propósito en Zero Trust
IAM Identity Center	Punto de administración de políticas (PAP)
MFA Habilitado	Autenticación continua
Roles Temporales	Sin credenciales permanentes
Permission sets	Aplicación granular de privilegios



Este diagrama muestra el flujo completo de autenticación y autorización basado en Zero Trust:

- 1. Usuario: "ICDD"** → inicia sesión en el **User Portal**.
- 2. MFA** → se requiere autenticación multifactor para validar identidad.
- 3. AWS IAM Identity Center** → actúa como **PAP (Policy Administrator Point)** y **PDP (Policy Decision Point)**
- 4. Role Assignment: AdministrationAccess** → se concede acceso temporal a la cuenta AWS.
- 5. Se indican claramente los procesos de Authentication y Authorization**, que conforman la base del plano de identidad del proyecto Zero Trust.

Próximo Paso

Con la **identidad segura y MFA habilitado**, ya tenemos el **punto de entrada** al modelo Zero Trust.

El siguiente paso será entrar a la Fase 2: Auditoría y configuración inicial de identidad (IAM & roles), en donde vamos a:

- Auditar permisos efectivos.
- Definir políticas de acceso mínimo
- Preparar la base para la segmentación de recursos

Fase 2 - Identidad y Control de Acceso (IAM)

🎯 Objetivo

El objetivo en esta fase es aplicar el **principio de mínimo privilegio (Least Privilege principle)** y habilitar visibilidad completa sobre las identidades y roles de acceso dentro de nuestra cuenta AWS.

Esta fase representa la consolidación de los componentes **PAP** (Policy Administration Point) y **PDP** (Policy Decision Point) del modelo Zero Trust.

- Paso 1: Revisar la configuración de Identity Center
 1. Vamos al panel de **IAM Identity Center** → **AWS Accounts** → **Assignments**.
 2. Verificamos que nuestro usuario ICDD tenga asignado el permiso:
 - Permission set: **AdministratorAccess**

The screenshot shows the IAM Identity Center interface with the 'Users and groups' tab selected. It displays one assigned user, 'ICDD', who has been assigned the 'AdministratorAccess' permission set. A red box highlights the 'Permission sets' column for the user, showing the specific assignment.

Username / group ...	Permission sets	Type
ICDD	• AdministratorAccess	User

3. Si queremos probar distintos niveles de acceso (como por ejemplo, un usuario sin permisos de escritura), podemos crear otro permission set:
 - **Permission set name:** ZTA-ReadOnly
 - **Policies:** seleccionamos ReadOnlyAccess.

Review and create

Step 1: Select permission set type

Edit

Permission set type

Type

Predefined permission set

AWS managed policy

ReadOnlyAccess

Step 2: Define permission set details

Edit

Permission set details

Permission set name

ZTA-ReadOnly

Session duration

1 hour

Description

-

Relay state

-

Resultado esperado:

Tenemos acceso a al menos dos roles: uno con acceso administrativo y otro limitado (útil por si queremos simular un **PEP** más adelante).

- Paso 2: Crear roles y políticas específicas (desde AWS IAM)

Por mas que usemos Identity Center, el servicio IAM sigue siendo donde definimos las políticas detalladas.

Estas políticas son la base del Enforcement Point (PEP) dentro de nuestra arquitectura.

- En la consola, abrimos **IAM** → **Policies** → **Create policy**.
- Creamos una política personalizada llamada ZT-Restricted-S3-Access:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowReadAccessToSpecificBucket",  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3>ListBucket"  
      ]  
    }  
  ]  
}
```

```

],
"Resource": [
"arn:aws:s3::::zt-secure-bucket",
"arn:aws:s3::::zt-secure-bucket/*"
],
"Condition": {
"Bool": {
"aws:SecureTransport": "true"
}
}
}
]
}

```

3. Guardamos la política.

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and a search bar. The main area has a title 'Policies (1399)'. Below it, a sub-header says 'A policy is an object in AWS that defines permissions.' There's a 'Filter by Type' dropdown set to 'All types'. A pagination bar shows pages 1 through 70. A table lists policies, with one row highlighted by a red box: 'Policy name: ZT-Restricted-S3-Access', 'Type: Customer man...', and 'Used as: None'.

Resultado esperado:

Tenemos una política IAM que aplica **acceso condicional** (HTTPS obligatorio), alineado con Zero Trust.

- Paso 3: Crear un rol IAM para acceso controlado

1. En **IAM → Roles → Create role**.
2. Tipo de entidad: **AWS Account** → selecciona "This account".
3. Asignamos la política que creamos anteriormente "ZT-Restricted-S3-Access).
4. Le nombraremos ZT-S3-Auditor.

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+=_.,@-_-' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '_+=_,@-_/'
[{}!#\$%^&*();~`]

Step 1: Select trusted entities Edit

Trust policy

```

1 [{}]
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole",
7       "Principal": {
8         "AWS": "94
9       },
10      "Condition": {}
11    }
12  ]
13 ]

```

Step 2: Add permissions Edit

Permissions policy summary

Policy name	Type	Attached as
ZT-Restricted-S3-Access	Customer managed	Permissions policy

Resultado esperado:

Creamos un rol con permisos mínimos que servirá como **Policy Enforcement Point (PEP)**

- Paso 4: Asignar ese rol a un usuario del Identity Center

- Volvemos a **IAM Identity Center** → **AWS Accounts** → **Assign users**.
- Escogemos nuestra cuenta AWS.
- Seleccionamos a nuestro usuario ICDD
- Asignamos un nuevo **Permission set** y elegimos el rol ZT-S3-Auditor

Users and groups (1) **Permission sets (1)**

Assigned users and groups (1)

Change permission sets Remove access **Assign users or groups**

The following users and groups in IAM Identity Center can select this AWS account from within their AWS access portal. [Learn more](#)

Search users by username, search groups by group name

< 1 > |

Username / group ...	Permission sets	Type
<input type="radio"/> ICDD	• AdministratorAccess • ZTA-ReadOnly	User

Resultado esperado: Nuestro usuario ahora puede cambiar entre dos roles:

- AdministratorAccess
- ZT-S3-Auditor (acceso restringido a S3 bajo Zero Trust).

Esto refleja el modelo Zero Trust de separación de privilegios por contexto.

- Paso 5: Auditar usuarios, roles y permisos activos

Con AWS CLI, ejecutamos los siguientes comandos:

```
aws iam list-users  
aws iam list-roles  
aws iam list-policies --scope Local  
aws iam generate-credential-report
```

Esto nos da una visión de:

- Qué usuarios existen.
- Qué roles están definidos.
- Qué políticas están activas.

```
aws iam list-users
```

```
CloudShell
us-east-1 + 

~ $ aws iam list-users
{
    "Users": []
}
~ $
```

Esta salida indica que no hay usuarios IAM clásicos creados en la cuenta. Esta salida es correcta si estamos usando AWS Identity Center (SSO), porque:

- Los usuarios de Identity Center no se gestionan en IAM clásico, sino en su propio directorio interno.
- Por eso el comando aws iam list-users solo muestra usuarios IAM tradicionales, no los de Identity Center.

Podemos decir que nuestro entorno está limpio, sin usuarios IAM permanentes, lo cual es ideal para Zero Trust (evita credenciales estáticas).

```
aws iam list-roles
```

```
CloudShell
us-east-1 + 

~ $ aws iam list-roles
{
    "Roles": [
        {
            "Path": "/aws-reserved/sso.amazonaws.com/",
            "RoleName": "AWSReservedSSO_AdministratorAccess_17ef4cddad00e978",
            "RoleId": "AROA52WHAEFRJ0FPFR4FT",
            "Arn": "arn:aws:iam::948531372386:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_AdministratorAccess_17ef4cddad00e978",
            "CreateDate": "2025-10-21T09:35:35+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Federated": "arn:aws:iam::948531372386:saml-provider/AWSSSO_cfc9d23a14387379_DO_NOT_DELETE"
                        },
                        "Action": [
                            "sts:AssumeRoleWithSAML",
                            "sts:TagSession"
                        ],
                        "Condition": {
                            "StringEquals": {
                                "SAML:aud": "https://signin.aws.amazon.com/saml"
                            }
                        }
                    }
                ]
            },
            "MaxSessionDuration": 43200
        },
        {
            "Path": "/aws-reserved/sso.amazonaws.com/",
            "RoleName": "AWSReservedSSO_ZTA-ReadOnly_801ea194c0c00be1",
            "RoleId": "AROA52WHAEFREZOKKNXF5",
            "Arn": "arn:aws:iam::948531372386:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_ZTA-ReadOnly_801ea194c0c00be1",
            "CreateDate": "2025-10-21T11:18:42+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Federated": "arn:aws:iam::948531372386:saml-provider/AWSSSO_cfc9d23a14387379_DO_NOT_DELETE"
                        },
                        "Action": [
                            "sts:AssumeRoleWithSAML"
                        ],
                        "Condition": {
                            "StringEquals": {
                                "SAML:aud": "https://signin.aws.amazon.com/saml"
                            }
                        }
                    }
                ]
            },
            "MaxSessionDuration": 43200
        }
    ]
}
```

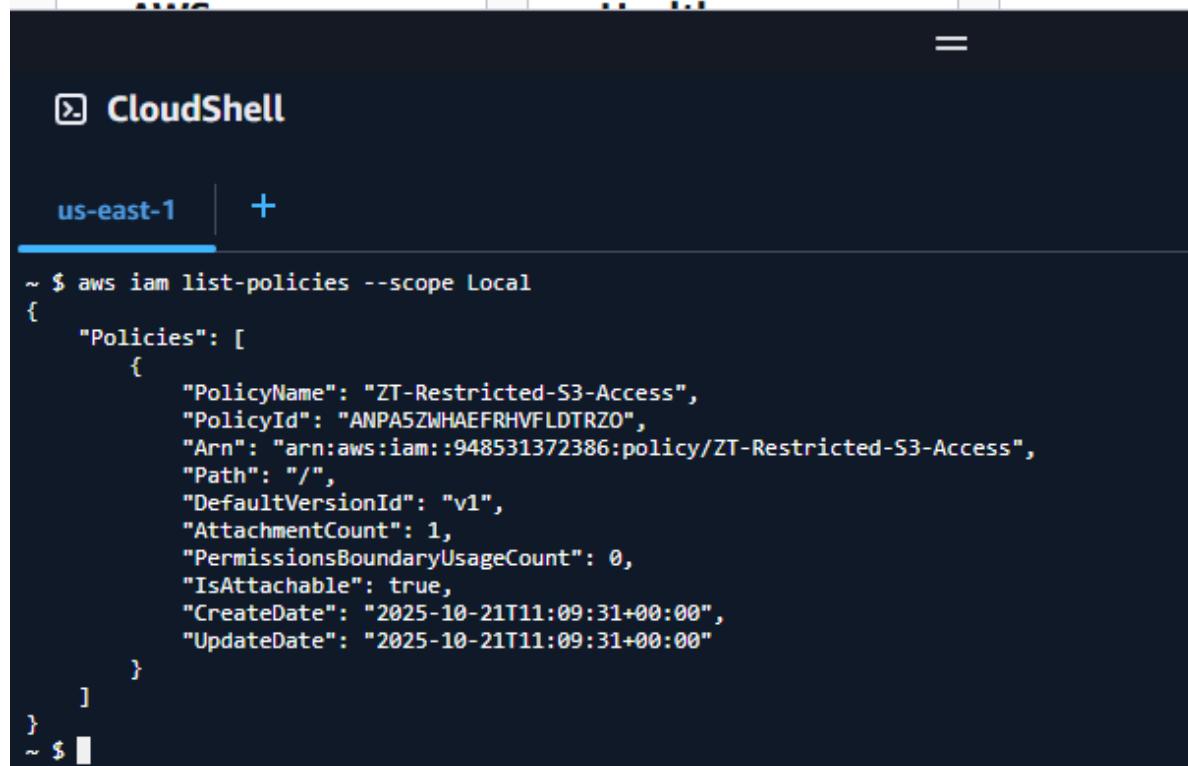
Este comando devuelve todos los roles IAM definidos en la cuenta, cada rol representa un conjunto de permisos que pueden asumir usuarios o servicios de AWS. Buscando bien en

la salida que nos dió podemos encontrar el **Permission set** que creamos hace un momento "ZTA-ReadOnly"

```
},  
{  
    "Path": "/aws-reserved/ss0.amazonaws.com/",  
    "RoleName": "AWSReservedSSO_ZTA-ReadOnly_801ea194c0c00be1",  
    "Arn": "arn:aws:iam::948531372386:role/aws-reserved/ss0.amazonaws.com/AWSReservedSSO_ZTA-ReadOnly_801ea194c0c00be1",  
    "CreateDate": "2025-10-21T11:18:42+00:00",  
    "AssumeRolePolicyDocument": {  
        "Version": "2012-10-17",  
        "Statement": [  
            {  
                "Effect": "Allow",  
                "Principal": {  
                    "Federated": "arn:aws:iam::948531372386:saml-provider/AWSSSO_cfc9d23a14387379_DO_NOT_DELETE"  
                },  
                "Action": [  
                    "sts:AssumeRoleWithSAML",  
                    "sts:TagSession"  
                ],  
                "Condition": {  
                    "StringEquals": {  
                        "SAML:aud": "https://signin.aws.amazon.com/saml"  
                    }  
                }  
            }  
        ]  
    },  
    "MaxSessionDuration": 43200  
},  
"MaxSessionDuration": 43200
```

```
aws iam list-policies --scope Local
```

Este comando muestra solo las políticas personalizadas que hemos creado en esta cuenta (no las políticas predeterminadas de AWS). Este comando es super útil para auditar nuestras propias reglas de seguridad y asegurarnos de que solo existen las necesarias.



```
CloudShell  
us-east-1 +  
~ $ aws iam list-policies --scope Local  
{  
    "Policies": [  
        {  
            "PolicyName": "ZT-Restricted-S3-Access",  
            "PolicyId": "ANPA5ZWHAERHVFLDTRZ0",  
            "Arn": "arn:aws:iam::948531372386:policy/ZT-Restricted-S3-Access",  
            "Path": "/",  
            "DefaultVersionId": "v1",  
            "AttachmentCount": 1,  
            "PermissionsBoundaryUsageCount": 0,  
            "IsAttachable": true,  
            "CreateDate": "2025-10-21T11:09:31+00:00",  
            "UpdateDate": "2025-10-21T11:09:31+00:00"  
        }  
    ]  
}  
~ $ |
```

```
aws iam generate-credential-report
```

Este comando genera un reporte de credenciales con el estado de seguridad de todos los usuarios IAM (si tienen MFA, claves activas, etc).

```
CloudShell
```

us-east-1 +

```
~ $ aws iam generate-credential-report
{
    "State": "COMPLETE"
}
~ $
```

En nuestro caso, nuestra salida significa que AWS generó correctamente el reporte de credenciales, pero aún no lo descargamos. Lo podemos ver ejecutando el comando:
`aws iam get-credential-report`

Este comando devolverá un CSV codificado en Base64, algo como esto:

```
{  
    "Content": "QWNjb3VudCBOYW1lLENyZWF0aW9uIERhdGU...==",  
    "ReportFormat": "text/csv",  
    "GeneratedTime": "2025-10-21T12:34:56+00:00"  
}
```

Podemos decodificarlo con el siguiente comando:

```
aws iam get-credential-report --query 'Content' --output text | base64 --decode > iam-credential-report.csv
```

Luego de correr ese comando obtenemos lo siguiente:

Resultado de la Fase 2

Elemento Configurado	Componente de Zero Trust	Descripción
Identity Center	PAP/PDP	Administra políticas de acceso e identidad.
IAM Policies	PAP	Define permisos mínimos y condicionales.
IAM Roles	PEP	Aplica las políticas en recursos específicos.
MFA	PIP	Aporta información de autenticación contextual.
CLI Audit	CA-7 (NIST 800-53)	Valida configuración y cumplimiento.

Fase 3 - Protección de Datos (S3 + SSE-SS)

🎯 Objetivo

Configurar un almacenamiento seguro (S3) con:

- Acceso restringido bajo políticas Zero Trust.
- Cifrado en reposo mediante KMS o SSE-S3
- Bloqueo total del acceso público.
- Auditoría del uso mediante CloudTrail
- Paso 1: Crear el Bucket Seguro

1. Desde CloudShell o desde la consola podemos crear un bucket nuevo, en este caso vamos a nombrarlo zta-secure-bucket, y lo crearemos con el siguiente comando:

```
aws s3 mb s3://zta-secure-bucket --region us-east-1
```

O desde la consola vamos a S3 → **General purpose buckets** → **Create bucket**.

⌚ Successfully created bucket "zta-secure-bucket"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

[General purpose buckets](#) [All AWS Regions](#) [Directory buckets](#)

General purpose buckets (1) [Info](#)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
zta-secure-bucket	US East (N. Virginia) us-east-1	October 21, 2025, 07:18:13 (UTC-05:00)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

- Paso 2: Bloquear acceso público

1. Vamos a la consola, → **Amazon S3** → **Nuestro bucket** → **Permissions** → **Block public access**.
2. Activamos todas las opciones de bloqueo:
 - Block all public access
 - Block ACLs
 - Block New public policies.

[Amazon S3](#) > [Buckets](#) > [zta-secure-bucket](#)

⌚ Successfully edited Block Public Access settings for this bucket.

Permissions overview

Access finding
Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#). [View analyzer for us-east-1](#)

Block public access (bucket settings)

[Edit](#)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
 On

▼ Individual Block Public Access settings for this bucket

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Resultado:

Ningún objeto será accesible desde fuera de AWS (PEP: Enforcement de acceso).

- Paso 3: Habilitar Versioning y cifrado

1. Vamos nuevamente a la consola → **Amazon S3** → **Nuestro bucket** → **Properties**.
2. Bucket Versioning → Edit → Enable → Save Changes.

The screenshot shows the AWS S3 Bucket Properties page for the 'zta-secure-bucket'. At the top, there are tabs for EC2, S3, and VPC. Below the tabs, the breadcrumb navigation shows 'Amazon S3 > Buckets > zta-secure-bucket'. A green success message box is displayed with the text: 'Successfully edited Bucket Versioning' and 'To transition, archive, or delete older object versions, configure lifecycle rules for this bucket.' Below the message, the 'Bucket Versioning' section is shown with the status 'Enabled'. There is also a 'Multi-factor authentication (MFA) delete' section with the status 'Disabled'. The bottom of the page shows a note about tags: 'No tags associated with this resource.'

3. Default encryption → Edit → Server-side encryption with Amazon S3 managed keys (SSE-S3) → Save Changes

The screenshot shows the AWS S3 Bucket Properties page for the 'zta-secure-bucket'. A green success message box is displayed with the text: 'Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.' Below the message, the 'Default encryption' section is shown with the status 'Server-side encryption is automatically applied to new objects stored in this bucket.' It also shows the 'Encryption type' as 'Info' and 'Server-side encryption with Amazon S3 managed keys (SSE-S3)'. There is a 'Bucket Key' section with the status 'Enabled'.

Nota: Opcional podemos activar el Server-side encryption with AWS Key Management Service keys (SSE-KMS), pero debemos de tener en cuenta de que cambiar las configuraciones de cifrado puede causar problemas de replicación y Batch replication. Eso puede fallar por la falta de permisos de parte de AWS KMS en los roles IAM, hay que tener en cuenta y verificar de que los roles IAM tengan los permisos de AWS KMS necesarios.

Resultado:

Cada archivo que subamos se cifrará automáticamente con AES-256 en el lado del servidor.

- Paso 4: Subir un archivo de prueba y verificar su cifrado

1. Creamos un archivo local en AWS CloudShell:

```
echo "Prueba Zero Trust S3" > prueba.txt
```

```
us-east-1 | +  
~ $ echo "Prueba Zero Trust S3" > prueba.txt  
~ $ ls  
iam-credential-report.csv  iam-policies.json  iam-roles.json  iam-users.json  prueba.txt  
~ $
```

2. Subimos el archivo:

```
aws s3 cp prueba.txt s3://zta-secure-bucket/
```

```
~ $ aws s3 cp prueba.txt s3://zta-secure-bucket/  
upload: ./prueba.txt to s3://zta-secure-bucket/prueba.txt  
~ $
```

3. Verificamos su cifrado:

```
aws s3api head-object --bucket zta-secure-bucket --key prueba.txt
```

```
us-east-1 +  
~ $ aws s3api head-object --bucket zta-secure-bucket --key prueba.txt  
{  
    "AcceptRanges": "bytes",  
    "LastModified": "2025-10-21T12:29:54+00:00",  
    "ContentLength": 21,  
    "ETag": "\"a0377b2545bd8176bec37a2b7c99730b\"",  
    "VersionId": "Ayvo9J16ctsvQatbGN_8jPVoQGtnBYRq",  
    "ContentType": "text/plain",  
    "ServerSideEncryption": "AES256",  
}  
~ $
```

Resultado esperado:

El archivo se almacena cifrado y versionado.

- Paso 5: Crear Política de acceso Zero Trust para S3

1. Crearemos un archivo zt-s3-policy.json con el siguiente contenido:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "DenyUnsecuredTransport",  
      "Effect": "Deny",  
      "Principal": "*",  
      "Action": "s3:*",  
      "Resource": [  
        "arn:aws:s3:::zta-secure-bucket",  
        "arn:aws:s3:::zta-secure-bucket/*"  
      ],  
      "Condition": {  
        "Bool": {"aws:SecureTransport": "false"}  
      }  
    },  
    {  
      "Sid": "DenyUnencryptedUploads",  
      "Effect": "Deny",  
      "Principal": "*",  
      "Action": "s3:PutObject",  
      "Resource": "arn:aws:s3:::zta-secure-bucket/*",  
      "Condition": {  
        "StringNotEquals": {"s3:x-amz-server-side-encryption": "AES256"}  
      }  
    }  
  ]  
}
```

2. Aplicamos la política con el siguiente comando:

```
aws s3api put-bucket-policy --bucket zta-secure-bucket --policy file://zta-s3-policy.json
```

■ Resultado:

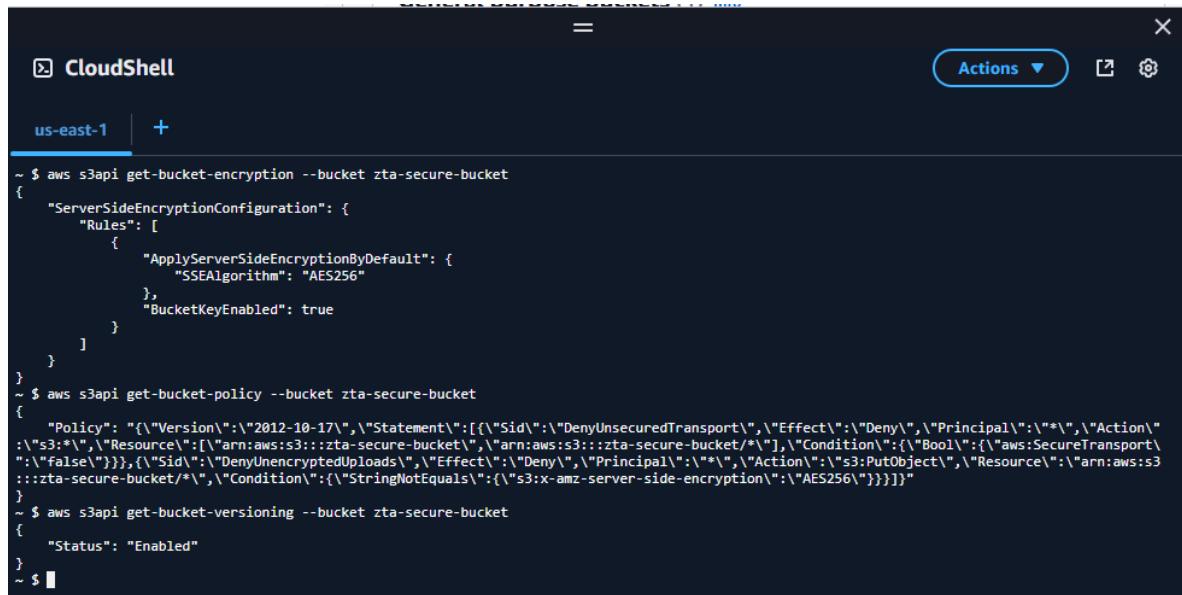
Solo se permiten conexiones seguras y objetos cifrados. Esto en definitiva es el **Zero Trust Enforcement Point (PEP)** en acción.

- Paso6: Validar la configuración

1. Verificamos que todo esté activo con los siguientes comandos:

```
aws s3api get-bucket-encryption --bucket zta-secure-bucket
aws s3api get-bucket-policy --bucket zta-secure-bucket
aws s3api get-bucket-versioning --bucket zta-secure-bucket
```

2. Salidas:



The screenshot shows a terminal window titled "CloudShell" with the region set to "us-east-1". The session history displays three AWS CLI commands run sequentially:

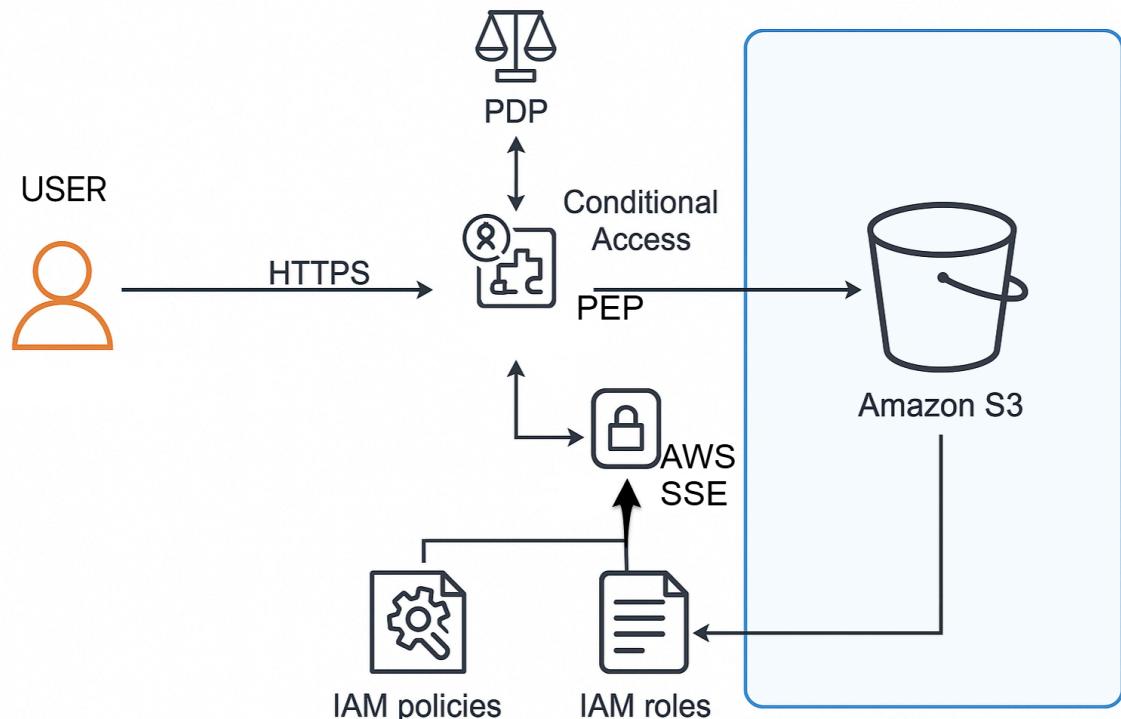
```
~ $ aws s3api get-bucket-encryption --bucket zta-secure-bucket
{
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {
          "SSEAlgorithm": "AES256"
        },
        "BucketKeyEnabled": true
      }
    ]
  }
}
~ $ aws s3api get-bucket-policy --bucket zta-secure-bucket
{
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"DenyUnsecuredTransport\",\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"s3:*\",\"Resource\": [\"arn:aws:s3:::zta-secure-bucket\", \"arn:aws:s3:::zta-secure-bucket/*\"], \"Condition\":{\"Bool\":{\"aws:SecureTransport\":\"false\"}}, {\"Sid\":\"DenyUnencryptedUploads\",\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"s3:PutObject\",\"Resource\": \"arn:aws:s3:::zta-secure-bucket/*\", \"Condition\":{\"StringNotEquals\":{\"s3:x-amz-server-side-encryption\":\"AES256\"}}}]}"
}
~ $ aws s3api get-bucket-versioning --bucket zta-secure-bucket
{
  "Status": "Enabled"
}
~ $
```

Resultados de la Fase 3

Control Aplicado	Propósito	Componente Zero Trust
Bloqueo Público	Impide acceso implícito	PEP
Cifrado SSE/KMS	Protege datos en reposo	PAP/PDP
Políticas condicionales	Exige HTTPS y cifrado	PDP/PEP
Versioning	Control de integridad	PIP
Auditoría con CLI	Evidencia de cumplimiento	CA-7 (NIST)

Hasta ahora tenemos:

- Identidad Segura.
- Almacenamiento cifrado y segmentado.



Este diagrama refleja lo siguiente:

- El usuario accediendo mediante HTTPS y autenticado con MFA (Aunque no se muestre en el diagrama)
- IAM Policies & Roles actuando como PAP (Policy Administrator Point) y PEP (Policy Enforcement Point)
- PDP (Policy Decision Point) tomando decisiones de acceso condicional.

- AWS SS3/KMS Proporciona cifrado gestionado de datos.
- Amazon S3 como almacenamiento protegido, con políticas de acceso cifrado y auditoría continua.

Fase 4: Segmentación de red con VPC y Security Groups

🎯 Objetivos

En esa fase vamos a construir y aislar recursos en una subred pública (bastion) y una subred privada (DB), y aplicaremos controles de red mínimamente necesarios y preparar todo para colocar una RDS.

Básicamente crearemos una red segura basada en Zero Trust: microsegmentación, acceso mínimo y controlado.

¿Qué vamos a construir?

- VPC (10.0.0.0/16)
 - Subred pública 10.0.1.0/24 → EC2 Bastion (o podemos acceder mediante SSM Session Manager para evitar abrir SSH)
 - Subred privada 10.0.2.0/24 → RDS (Sin IP Pública)
 - Un Internet Gateway solo para la subred pública.
 - Route tables separadas (Una con ruta pública a Internet, y otra privada sin salida pública).
 - Security Groups (Este actuará como un PEP):
 - sg-bastion: este security group solo dará acceso desde nuestra IP pública.
 - sg-rds: este security group solo permitirá conexiones a los puertos 3306 y/o 5432 desde sg-bastion.
 - Paso 1: Crear la VPC
1. En nuestra consola AWS, abrimos el servicio VPC.
 2. En el menú lateral, elegimos **Your VPCs** → **Create VPC**
 3. Marcamos la "opción VPC only"
 4. Y configuramos:
 - **Name tag:** ZT-VPC
 - **IPv4 CIDR block:** 10.0.0.0/16
 - **Tenancy:** Default
5. Luego de colocar la información, creamos la VPC haciendo click en **Create VPC**.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only **VPC and more**

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
ZT-VPC

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
IPv4 CIDR **10.0.0.0/16**
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text"/> Name	<input type="text"/> ZT-VPC X Remove tag

Add tag
You can add 49 more tags

[Cancel](#) [Preview code](#) **Create VPC**

■ Nota: Esta será nuestra red principal, en otras palabras podemos verla como una "caja de seguridad" que es donde se aislarán todos los recursos.

- Paso 2: Crear las subredes
 - Subred pública
1. En el menú lateral, clickeamos en **Subnets** → **Create subnet**
 2. Elegimos la VPC que acabamos de crear: ZT-VPC
 3. Llenamos la siguiente información:
 - **Name tag:** ZT-Public-Subnet
 - **Availability Zone:** en este caso usaremos us-east-1a
 - **IPv4 CIDR block:** 10.0.1.0/24
 4. Luego clickeamos en Create Subnet.

The screenshot shows the AWS VPC Subnets Details page. The subnet ID is subnet-09f84443ed22e8f1a. The 'IPv4 CIDR' is 10.0.1.0/24 and the 'Availability Zone' is us-east-1a. Other details include:

Setting	Value
Subnet ID	subnet-09f84443ed22e8f1a
Subnet ARN	arn:aws:ec2:us-east-1:000000000000:subnet/09f84443ed22e8f1a
State	Available
IPv6 CIDR	-
VPC	vpc-000000000000
Network border group	us-east-1
Auto-assign public IPv4 address	No
Default subnet	No
Customer-owned IPv4 pool	-
IPv6-only	No
Resource name DNS AAAA record	Disabled
DNS64	Disabled
Outpost ID	-
Hostname type	IP name
Owner	94
Block Public Access	Off
IPv6 CIDR association ID	-
Route table	-
Auto-assign IPv6 address	No
IPv4 CIDR reservations	-
Resource name DNS A record	Disabled

- Subred privada
1. Repetimos todos los pasos anteriores, pero esta vez colocamos la siguiente información:
 - **Name tag:** ZT-Private-Subnet
 - **Availability Zone:** us-east-1a
 - **CIDR block:** 10.0.2.0/24
 2. Y hacemos click en **Create Subnet**.

subnet-0c85456c86c3d2b04 / ZT-Private-Subnet

Actions ▾

Details

Subnet ID subnet-0c85456c86c3d2b04	Subnet ARN arn:aws:ec2:us-east-1:123456789012:subnet/0c85456c86c3d2b04	State Available	Block Public Access <input checked="" type="radio"/> Off
IPv4 CIDR 10.0.2.0/24	6c86c3d2b04	IPv6 CIDR	IPv6 CIDR association
Availability Zone use1-az2 (us-east-1a)	Available IPv4 addresses 251	VPC vpc- ZT-VPC	ID Route table -
Network ACL -	Network border group us-east-1	Auto-assign public IPv4 address No	Auto-assign IPv6 address No
Auto-assign customer-owned IPv4 address No	Default subnet No	Outpost ID	IPv4 CIDR reservations
IPv6 CIDR reservations -	Customer-owned IPv4 pool -	Hostname type IP name	Resource name DNS A record Disabled
Resource name DNS AAAA record Disabled	IPv6-only No	Owner 948	
	DNS64 Disabled		

[Flow logs](#) | [Route table](#) | [Network ACL](#) | [CIDR reservations](#) | [Sharing](#) | >

La subred pública permitirá conexiones controladas, mientras que la subred privada servirá para nuestra base de datos, sin acceso a internet.

- Paso 3: Crear y asociar un Internet Gateway
1. En el panel lateral, clickeamos en **Internet Gateways** → **Create Internet gateway**
 2. Configuramos:
 - Name tag: ZT-IGW

The following internet gateway was created: igw-04abe18162ef26926 - ZT-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

[Attach to a VPC](#) [X](#)

igw-04abe18162ef26926 / ZT-IGW

[Actions ▾](#)

Details [Info](#)

Internet gateway ID igw-04abe18162ef26926	State Detached	VPC ID -	Owner 948531372386
--	-------------------	-------------	-----------------------

Tags [Manage tags](#)

|

Key		Value
Name		ZT-IGW

3. Clickeamos en **Create internet gateway**.
4. Luego lo seleccionamos nuevamente → **Actions** → **Attach to VPC** → **ZP-VPC**.

The screenshot shows two consecutive steps in the AWS CloudFormation console:

Step 1: Creating the Internet Gateway

- A green success message at the top states: "The following internet gateway was created: igw-04abe18162ef26926 - ZT-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet."
- The Internet gateway ID is listed as `igw-04abe18162ef26926`.
- The State is `Detached`.
- The VPC ID is listed as `-`.
- An "Actions" dropdown menu includes "Attach to VPC" (which is highlighted with a red box).
- Other options in the Actions menu are "Manage tags" and "Delete".

Step 2: Attaching the Internet Gateway to a VPC

- A green success message at the top states: "Internet gateway igw-04abe18162ef26926 successfully attached to vpc-009bc7d30afdeb192".
- The Internet gateway ID is listed as `igw-04abe18162ef26926`.
- The State is `Attached`.
- The VPC ID is listed as `vpc-009bc7d30afdeb192`, which is highlighted with a red box.
- The Owner is listed as `aws`.
- An "Actions" dropdown menu includes "Manage tags".
- Below the gateway details, there is a "Tags" section with a search bar and a table showing one tag: `Name: ZT-IGW`.

■ Esto permitirá que los recursos de la subred pública puedan comunicarse con Internet (Todo esto depende de la configuración de los Security Group, podemos crearlos de tal forma que permitan o nieguen la comunicación con Internet).

- Paso 4: Configurar las Route Tables
 - Route Table pública
1. En el menú lateral, clickeamos en **Route Tables** → **Create route table**.
 2. Configuramos:
 - **Name tag:** ZT-Public-RT
 - **VPC:** ZT-VPC

VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

ZT-Public-RT

VPC
The VPC to use for this route table.

vpc-009bc7d30afdeb192 (ZT-VPC)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	Remove
<input type="text" value="Name"/> X	<input type="text" value="ZT-Public-RT"/> X	Remove

Add new tag

You can add 49 more tags.

Cancel **Create route table**

The screenshot shows the 'Create route table' wizard. In the 'Route table settings' section, a tag 'Name' is set to 'ZT-Public-RT'. In the 'Tags' section, one tag is defined with key 'Name' and value 'ZT-Public-RT'. At the bottom, there are 'Cancel' and 'Create route table' buttons.

3. Clickeamos en **Create route table**.
4. Una vez creada, la seleccionamos → pestaña de **Routes** → **Edit routes** → **Edit subnet associations**
 - Asociamos la subred **ZT-Public-Subnet**

AWS Search [Alt+S] United States (N. Virginia) Account ID: Cristian Jimenez

EC2 S3 VPC

VPC Route tables

VPC dashboard

AWS Global View Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers New

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started
- Endpoints
- Endpoint services

Route tables (1/3) Last updated 4 minutes ago Actions Create route table

Name	Route table ID	Explicit subnet assoc...
-	rtb-0d464419dba41393f	-
-	rtb-0d464419dba41393f	-
<input checked="" type="checkbox"/> ZT-Public-RT	rtb-0d464419dba41393f	-

1.

rtb-0d464419dba41393f / ZT-Public-RT 2.

Details Routes Subnet associations Edge associations Routes

Explicit subnet associations (0) Edit subnet associations

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
ZT-Public-Subnet	subnet-09f84443ed22e8f1a	10.0.1.0/24	-	Main (rtb-06at)
ZT-Private-Subnet	subnet-09f84443ed22e8f1a	10.0.2.0/24	-	Main (rtb-06at)

No subnet associations

3.

EC2 S3 VPC

VPC Route tables rtb-0d464419dba41393f / ZT-Public-RT Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

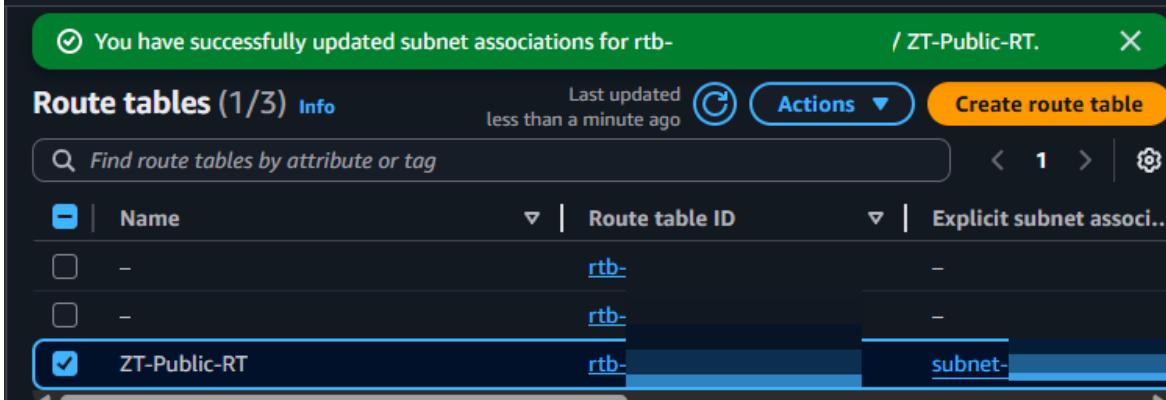
Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> ZT-Public-Subnet	subnet-09f84443ed22e8f1a	10.0.1.0/24	-	Main (rtb-06at)
<input type="checkbox"/> ZT-Private-Subnet	subnet-09f84443ed22e8f1a	10.0.2.0/24	-	Main (rtb-06at)

Selected subnets

subnet-09f84443ed22e8f1a / ZT-Public-Subnet X

Cancel Save associations



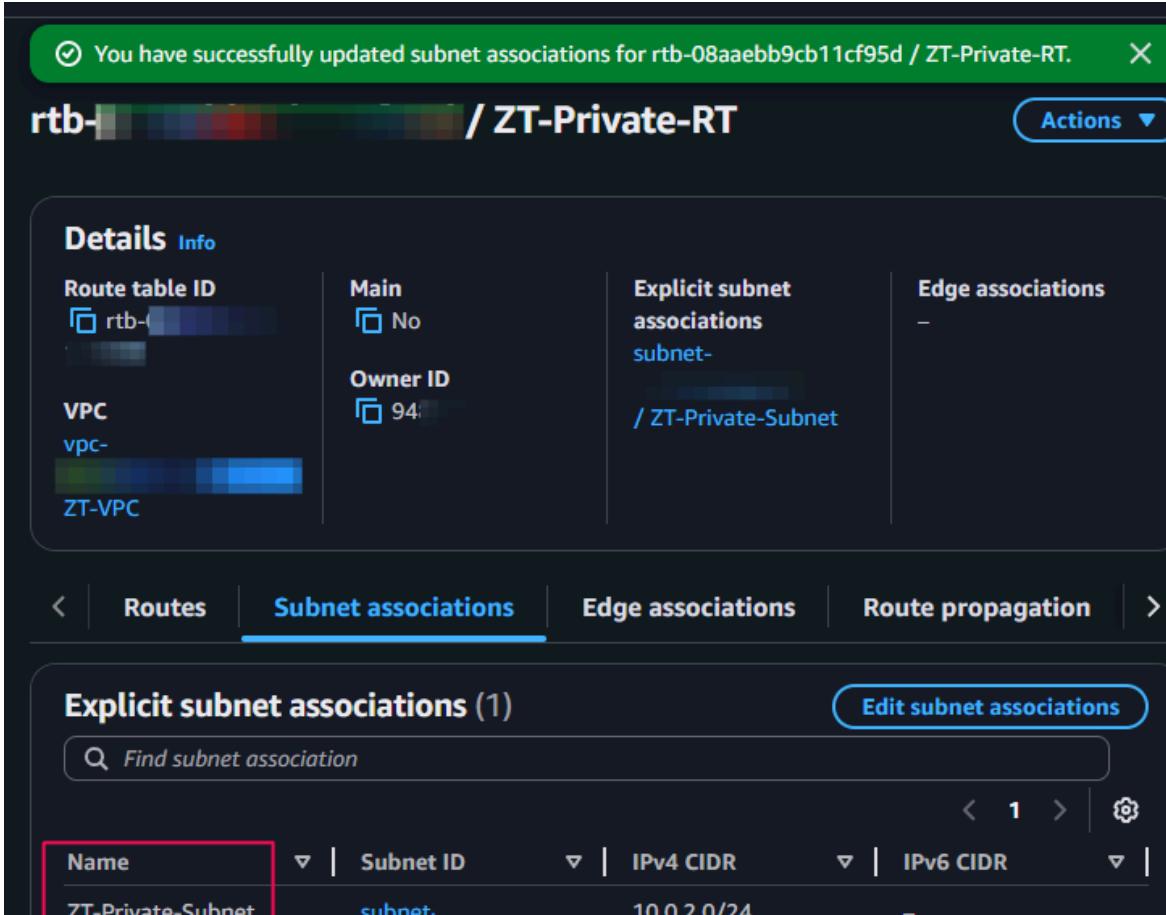
You have successfully updated subnet associations for rtb-08aaebb9cb11cf95d / ZT-Public-RT.

Route tables (1/3) Info Last updated less than a minute ago **Actions** **Create route table**

Find route tables by attribute or tag

Name	Route table ID	Explicit subnet associ...
-	rtb-	-
-	rtb-	-
<input checked="" type="checkbox"/> ZT-Public-RT	rtb-	subnet-

- Crear Route Table privada
1. Para crear la nueva Route Table seguimos los mismos pasos que hicimos anteriormente, pero con la siguiente información:
 - **Name tag:** ZT-Private-RT
 - **VPC:** ZT-VPC
 2. En esta route table no agregaremos ninguna ruta pública.
 3. En Subnet associations, vamos a asociar ZT-Private-Subnet.



You have successfully updated subnet associations for rtb-08aaebb9cb11cf95d / ZT-Private-RT.

rtb-08aaebb9cb11cf95d / ZT-Private-RT Actions

Details Info

Route table ID <input type="checkbox"/> rtb-08aaebb9cb11cf95d	Main <input type="checkbox"/> No	Explicit subnet associations subnet-08aaebb9cb11cf95d / ZT-Private-Subnet	Edge associations -
VPC vpc-08aaebb9cb11cf95d	Owner ID <input type="checkbox"/> 94		
ZT-VPC			

Routes **Subnet associations** **Edge associations** **Route propagation**

Explicit subnet associations (1) **Edit subnet associations**

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
ZT-Private-Subnet	subnet-08aaebb9cb11cf95d	10.0.2.0/24	-

Como resultado tenemos que la subred pública es la única que tiene salida a Internet, mientras que la subred privada está completamente aislada.

- Paso 5: Crear los Security Groups
 - Security Group para Bastion Host
1. En **VPC** → nos ubicamos en el panel de la izquierda y clickeamos **Security Groups** → **Create security group**.
 2. Configuramos:
 - **Name:** ZT-SG-Bastion
 - **Descripción:** Bastion host control
 - **VPC:** ZT-VPC

The screenshot shows the 'Create security group' wizard. At the top, the breadcrumb navigation shows 'VPC > Security Groups > Create security group'. The main section is titled 'Create security group' with a 'Info' link. A descriptive text explains that a security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Below this, the 'Basic details' section contains three fields: 'Security group name' (set to 'ZT-SG-Bastion'), 'Description' (set to 'Bastion host control'), and 'VPC' (set to 'vpc- (ZT-VPC)').

3. Justo abajo, en la sección **Inbound rules** → clickeamos en **Add rule**:
 - **Type:** SSH
 - **Port:** 22
 - **Source:** My IP (Nuestra IP Pública actual)

The screenshot shows the 'Inbound rules' section of the security group configuration. It displays a single existing rule ('Inbound rule 1') and a form for adding a new rule. The new rule configuration includes: 'Type' set to 'SSH', 'Protocol' set to 'TCP', 'Port range' set to '22', 'Source type' set to 'My IP', and an empty 'Description - optional' field. At the bottom left, there is a blue 'Add rule' button.

4. Luego, en **Outbound rules**, dejamos el tráfico abierto (0.0.0.0/0)

The screenshot shows the 'Outbound rules' section of a cloud provider's interface. It displays a single rule named 'Outbound rule 1'. The rule configuration is as follows:

- Type: All traffic
- Protocol: All
- Port range: All
- Destination type: Custom
- Destination: 0.0.0.0/0
- Description - optional: (empty)

A prominent warning message at the bottom states: "⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses." with a close button.

5. Guardamos.

The screenshot shows the 'ZT-SG-Bastion' security group details page. At the top, a success message says: "✓ Security group (sg-[ZT-SG-Bastion](#)) was created successfully". The main details section includes:

Security group name ZT-SG-Bastion	Security group ID sg- [REDACTED]	Description Bastion host contr ol	VPC ID vpc- [REDACTED]
Owner 94	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Below this, a navigation bar allows switching between Inbound rules, Outbound rules, Sharing, and VPC associations. The 'Inbound rules' tab is selected, showing one rule:

Name	Security group rule ID	IP version	Type
-	sgr- [REDACTED]	IPv4	SSH

- └─ Esta configuración permitirá únicamente conexión SSH desde nuestra IP, evitando accesos externos.

- Security Group para la Base de Datos (RDS)

1. Creamos otro Security Group:

- **Name:** ZT-SG-RDS
- **Descripción:** RDS Private access
- **VPC:** ZT-VPC

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a security group, follow the steps below.

Basic details

Security group name Info

ZT-SG-RDS

Name cannot be edited after creation.

Description Info

RDS Private Access

VPC Info

vpc- (ZT-VPC) ▾

2. En **Inbound Rules** → click en **Add rule**:

- **Type:** MySQL/Aurora
- **Port:** 3306 o 5432 (en nuestro caso usaremos el 3306)
- **Source:** seleccionamos ZT-SG-Bastion (NO CIDR)

Inbound rules Info

Inbound rule 1 Delete

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>
MYSQL/Aurora	TCP	3306
Source type <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
Custom	<input type="text" value="sg-l"/> X <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> Use: "sg-l" [Color Swatches] CIDR blocks Security Groups ZT-SG-Bastion sg-l </div>	

Add rule

3. En **Outbound**, podemos dejarlo por defecto.

Outbound rules [Info](#)

Outbound rule 1

Type [Info](#) Protocol [Info](#) Port range [Info](#)

All traffic All All

Destination type [Info](#) Destination [Info](#) Description - optional [Info](#)

Custom Q 0.0.0.0/0 X

[Add rule](#)

⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses. [X](#)

- La base de datos solo aceptará conexiones desde el bastion host, no desde Internet.

sg-0b33fe0b22cf73dcb - ZT-SG-RDS

✓ Security group (sg-0b33fe0b22cf73dcb) was created successfully [X](#)

► Details

sg-0b33fe0b22cf73dcb - ZT-SG-RDS [Actions](#)

Details

Security group name ZT-SG-RDS	Security group ID sg-0b33fe0b22cf73dcb	Description RDS Private Access	VPC ID vpc-0b33fe0b22cf73dcb
Owner 94	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

< [Inbound rules](#) [Outbound rules](#) Sharing - new VPC associations - new >

Inbound rules (1) [Manage tags](#) [Edit inbound rules](#)

Search

<input type="checkbox"/> Name	▼ Security group rule ID	▼ IP version	▼ Type
<input type="checkbox"/> -	sg-0b33fe0b22cf73dcb	-	MySQL/AU...

- Paso 6: Crear instancia EC2 Bastion (dentro de la subred pública)
 - Primero abrimos el servicio **EC2** → **Instances** → **Launch instance**.
 - Configuramos así:

- **Name:** ZT-Bastion
- **AMI:** Amazon Linux 2 (64-bit x86)
- **Instance Type:** t2.micro
- **Key pair:** Seleccionamos o creamos una nueva (en nuestro caso crearemos un nuevo par).

The screenshot shows the AWS EC2 'Launch an instance' wizard. The first step, 'Name and tags', has 'ZT-Bastion' entered in the 'Name' field. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a grid of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. An 'Amazon Machine Image (AMI)' card for 'Amazon Linux 2023 kernel-6.1 AMI' is selected, showing details like AMI ID, Publish Date (2025-10-09), and Username (ec2-user). The third step, 'Instance type', shows the 't2.micro' instance type selected. The fourth step, 'Key pair (login)', shows 'ZT-KeyPair' selected as the key pair.

- Network settings:
 - **VPC:** ZT-VPC
 - **Subnet:** ZT-Public-Subnet

- **Auto-assign Public IP:** Enable
- **Security Group:** ZT-SG-Bastion

Network settings

VPC - required | Info
vpc-10.0.0.0/16 (ZT-VPC)

Subnet | Info
subnet-C... Owner: ZT-Public-Subnet
VPC: vpc-10.0.0.0/16 Availability Zone: us-east-1a (use1-az2) Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.1.0/24

Create new subnet

Auto-assign public IP | Info
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups | Info
Select security groups

ZT-SG-Bastion sg-0
X
VPC: vpc-10.0.0.0/16

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

3. En **Advanced details** podemos dejar todo por defecto.
4. Hacemos click en **Launch instance**.

Con lo que acabamos de hacer, creando el Bastion host, nos permitirá conectarnos de forma segura al entorno interno.

- Paso 7: Habilitar Session Manager (Sin Abrir SSH) (Recomendado)

El objetivo de Habilitar el SSM es para conectarnos a nuestra instancia EC2 sin abrir el puerto 22 ni usar claves SSH.

El acceso ocurre mediante AWS Systems Manager (SSM), que usa autenticación IAM + MFA, cumpliendo el principio "nunca confíes, siempre verifica" de Zero Trust.

1. Vamos a **EC2** → **Instances** → Seleccionamos nuestra instancia → **Actions** → **Security** → **Modify IAM role**.

Instances (1/1) Info

Last updated 16 minutes ago

Actions ▾ Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

Name ↴ Instance ID

ZT-Bastion

Change security groups
Get Windows password
Modify IAM role

Instance diagnostics
Instance settings
Networking
Security
Image and templates
Monitor and troubleshoot

2. Asignamos el rol administrado **AmazonSSMManagedInstanceCore**.

Nota: Si no tenemos el rol, podemos crearlo directamente desde el servicio IAM. Para tomar en cuenta, en tipo de identidad elegimos AWS Service, en Use case, EC2, y luego asignamos la política AmazonSSMManagedInstanceCore, le damos un nombre y lo asociamos con la instancia EC2.

EC2 S3 VPC

EC2 > Instances > i-0 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID: i-0...

IAM role: Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

ZT-Bastion-SSM-Role Create new IAM role

Cancel Update IAM role

Successfully attached ZT-Bastion-SSM-Role to instance i...

Instances (1/1) Info

Last updated 30 minutes ago

Actions ▾ Launch instances ▾

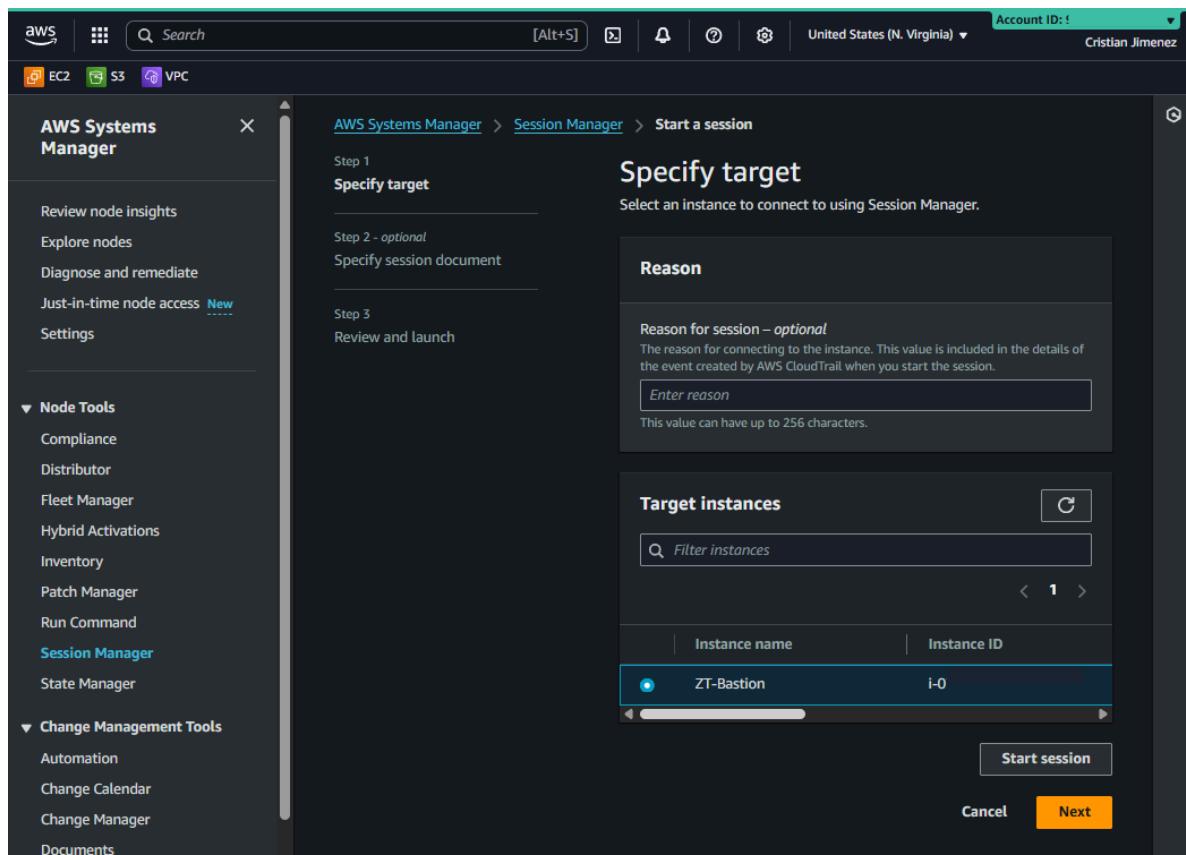
Find Instance by attribute or tag (case-sensitive)

All states ▾

Name ↴ Instance ID Instance state Instance type Status check

ZT-Bastion i-0... Running t2.micro 2/2 checks passed

3. Abrimos el **Systems Manager** → **Session Manager** → **Start Session** → Seleccionamos nuestra instancia.



4. Hacemos click en **Start Session**. Esto nos abre otra pestaña en nuestro navegador y deberíamos poder ver la terminal.

■ Ahora podemos acceder sin SSH, y tampoco sin IP expuesta, perfecto para Zero Trust.

- Paso 8: Crear un grupo de subredes para RDS

1. Abrimos el servicio RDS → Subnet groups → Create DB subnet group.

2. Configuramos:

- **Name:** ZT-DB-Subnet-Group
- Descripción: Private DB subnets
- VPC: ZT-VPC
- Subnets: seleccionamos ZT-Private-Subnet

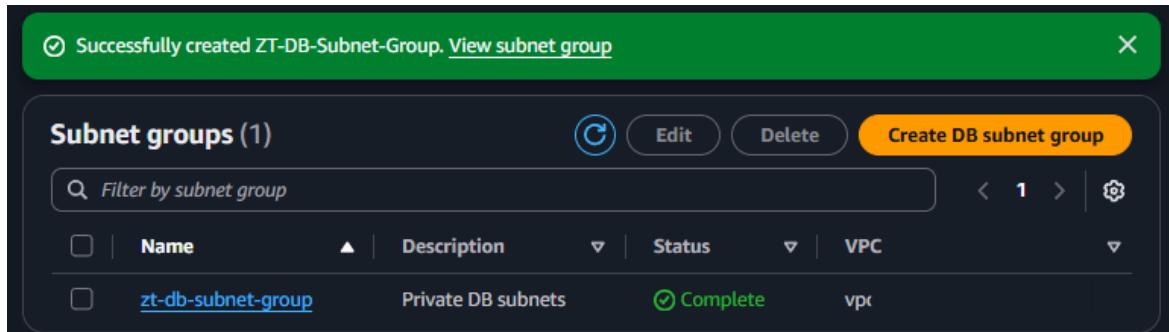
The screenshot shows the AWS RDS 'Create DB subnet group' wizard. The top navigation bar includes the AWS logo, search bar, and United States (N. Virginia) region. The breadcrumb trail shows 'Aurora and RDS > Subnet groups > Create DB subnet group'. The main section is titled 'Create DB subnet group' with a sub-section 'Subnet group details'. It contains fields for 'Name' (ZT-DB-Subnet-Group), 'Description' (Private DB subnets), and 'VPC' (ZT-VPC). Below this is the 'Add subnets' section, which includes 'Availability Zones' (us-east-1a, us-east-1b selected) and 'Subnets' (ZT-Private-Subnet-2 and ZT-Private-Subnet selected). A note states 'For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.' The 'Subnets selected (2)' table lists the chosen subnets. At the bottom are 'Cancel' and 'Create' buttons.

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1b	ZT-Private-Subnet-2	subnet-1	10.0.3.0/24
us-east-1a	ZT-Private-Subnet	subnet-2	10.0.2.0/24

Nota: A la hora de crear esta subnet group, el mismo servicio nos pedía explicitamente tener más de una AZ. Aunque no utilicemos Multi-AZ, AWS reserva la capacidad de mover la instancia si hay una falla en la AZ, por eso siempre pide al menos 2 subredes privadas de diferentes AZs para alta disponibilidad.

Por eso, en este caso, creamos una nueva subred con CIDR 10.0.3.0/24, luego la asociamos la subred a nuestra private route table (ZT-Private-RT) y una vez creado volvemos a crear el subnet group en RDS.

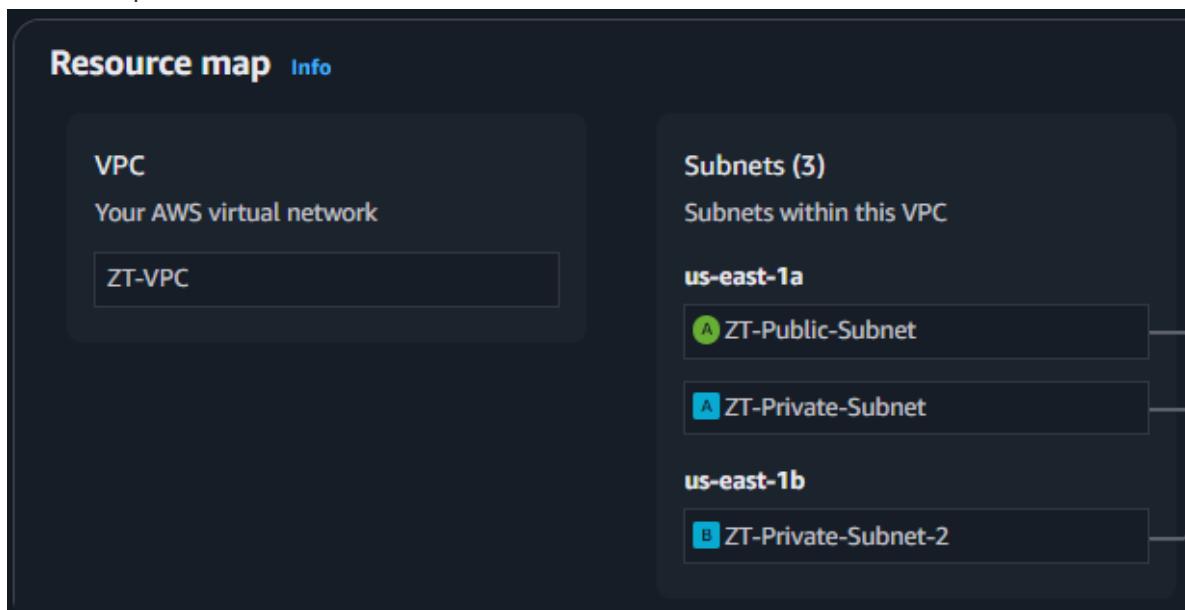
3. Clickeamos en **Create**.



- Paso 9: Validar configuración

Vamos a revisar visualmente lo que hemos hecho; VPC, Route Tables, Security Groups, EC2, y SSM.

- En el panel de VPC → Your VPCs: ZT-VPC con sus subredes



- Route Tables:
 - Route Table Pública con ruta 0.0.0.0/0 → IGW
 - Route Table Privada sin rutas externas

Route tables (1/4) [Info](#)

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
-	rtb-	-	-	Yes	vpc-	ZT-VPC
-	rtb-	-	-	Yes	vpc-	ZT-VPC
ZT-Private-RT	rtb-	2 subnets	-	No	vpc-	ZT-VPC
ZT-Public-RT	rtb-	subnet-	-	No	vpc-	ZT-VPC

rtb- / ZT-Public-RT

- Details
- Routes** (2)
- Subnet associations
- Edge associations
- Route propagation
- Tags

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-C	Active	No
10.0.0.0/16	local	Active	No

[Both](#) [Edit routes](#)

[Create Route](#) [Create Route Table](#)

Last updated 5 minutes ago [Actions](#) [Create route table](#)

Route tables (1/4) [Info](#)

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
-	rtb-	-	-	Yes	vpc-	9z
-	rtb-	-	-	Yes	vpc-	9z
ZT-Private-RT	rtb-	2 subnets	-	No	vpc-	9z
ZT-Public-RT	rtb-	subnet-	-	No	vpc-	9z

rtb- / ZT-Private-RT

- Details
- Routes** (1)
- Subnet associations
- Edge associations
- Route propagation
- Tags

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

[Both](#) [Edit routes](#)

[Create Route Table](#)

Last updated 5 minutes ago [Actions](#) [Create route table](#)

Resource map



- En Security Groups:
 - ZT-SG-Bastion: Solo nuestra IP
 - ZT-SG-RDS: Solo acepta conexiones del bastion.

ZT-SG-Bastion

Inbound rules (1)

Type	Protocol	Port range	Source
SSH	TCP	22	1 [Private IP]

ZT-SG-RDS

Inbound rules (1)

Protocol	Port range	Source	Description
TCP	3306	sg-ZT-SG-Bastion	ZT-SG-Bastion

- En EC2 → Instances: Verificamos que nuestra instancia esté corriendo.

Instances (1) [Info](#)

Instance state: Running

Name	Instance ID	Instance state	Instance type
ZT-Bastion	i-... Details	Running	t2.micro

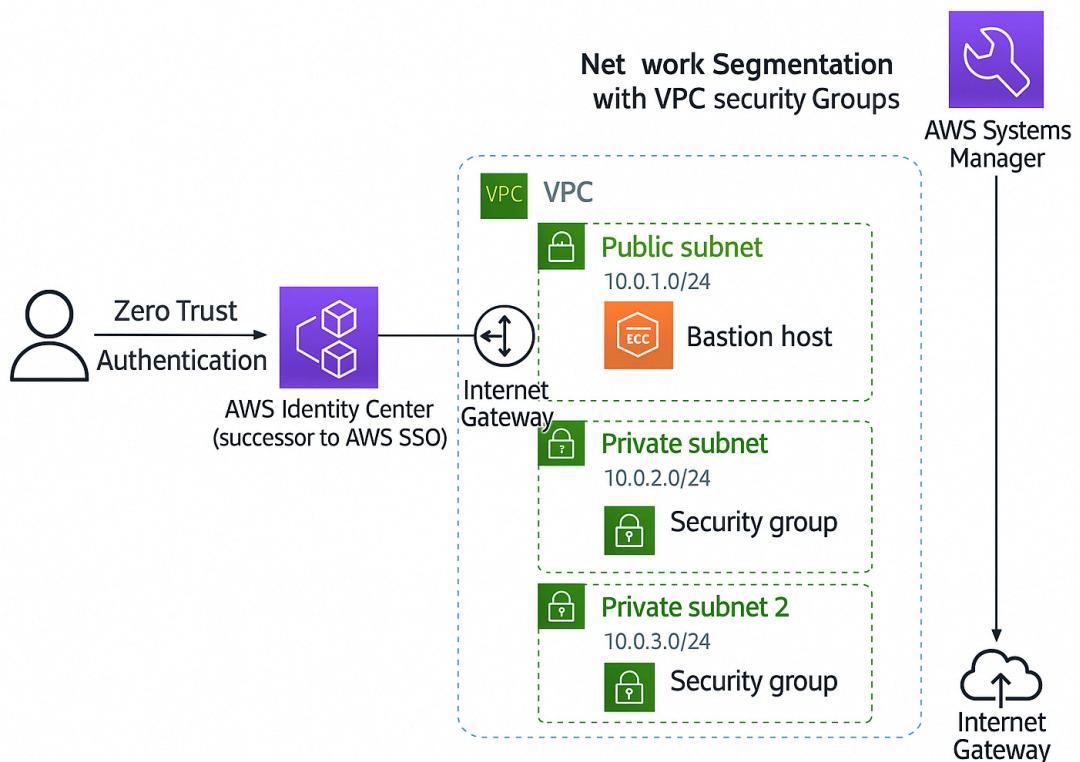
- En SSM: Verificamos que podemos abrir la sesión segura sin SSH.

Session ID: root-y62f77yfhorlunx3b7atanghga [Shortcuts](#) Instance ID: i-078f58cd118bbc93 [Terminate](#)

sh-5.2\$ pudimos acceder sin problemas a la instancia utilizando Systems Manager.

Zero Trust aplicado en esta fase

Principio	Implementación
Microsegmentación	Subredes públicas y privadas separadas
Mínimo privilegio	SG del RDS solo permite acceso del bastion (EC2)
No implicit trust	RDS sin IP Pública, acceso remoto interno
Verificación continua	MFA + Session Manager aseguran identidad



Los elementos principales de este diagrama, basado en lo que se ha hecho hasta ahora:

- **AWS Identity Center (PAP/PDP):** Autentica usuarios antes de cualquier acceso.
- **VPC con segmentación:**
 - Subred pública con la instancia Bastion Host.
 - Dos subredes privadas para bases de datos u otros recursos internos.
- **Internet Gateway (IGW):** Solo la subred pública tiene salida a internet.
- **AWS Systems Manager:** Es el canal seguro de gestión bajo políticas Zero Trust.
- **Security Groups:** Actúan como Policy Enforcement Points (PEP) que validan las conexiones entre capas.

FASE 5: Base de datos RDS Privada + KMS + IAM bajo Zero Trust

🎯 Objetivo

El objetivo en esta fase es implementar una base de datos Amazon RDS en la subred privada que creamos en la fase anterior, cifrada con AWS KMS, y accesible solo a través de la instancia bastion mediante políticas y security groups restrictivos.

Esta fase refuerza la **confidencialidad, integridad y control de acceso granular** de los datos en reposo, alineado al modelo Zero Trust.

- Paso 1: Crear instancia RDS Privada
1. Vamos al panel **RDS** → **Databases** → **Create databases**
 2. Seleccionamos lo siguiente:
 - **Engine type:** MySQL (o PostgreSQL)
 - **Templates:** Free Tier

Screenshot of the AWS RDS 'Create database' wizard.

Create database Info

Choose a database creation method

- Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

- Aurora (MySQL Compatible) 
- Aurora (PostgreSQL Compatible) 
- MySQL 
- PostgreSQL 
- MariaDB 
- Oracle 
- Microsoft SQL Server 
- IBM Db2 

Edition

- MySQL Community

Templates

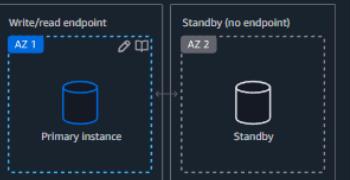
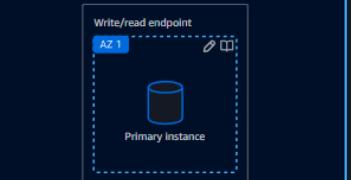
Choose a sample template to meet your use case.

- Production
Use defaults for high availability and fast, consistent performance.
- Dev/Test
This instance is intended for development use outside of a production environment.
- Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

Availability and durability

Deployment options Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

- Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability Zones
 - Increased read capacity
 - Reduced write latency
- Multi-AZ DB instance deployment (2 instances)
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability Zones
- Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:
 - 99.5% uptime
 - No data redundancy

3. Luego configuramos:

- DB instance identifier: ZT-DB-Instance

- Master username: admin
- Password: creamos una contraseña segura y la guardamos.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Password strength Strong
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " @

Confirm master password [Info](#)

4. En la DB instance size, elegimos, db.t3.micro (free tier elegible)

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

Hide filters

Show instance classes that support Amazon RDS Optimized Writes

Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

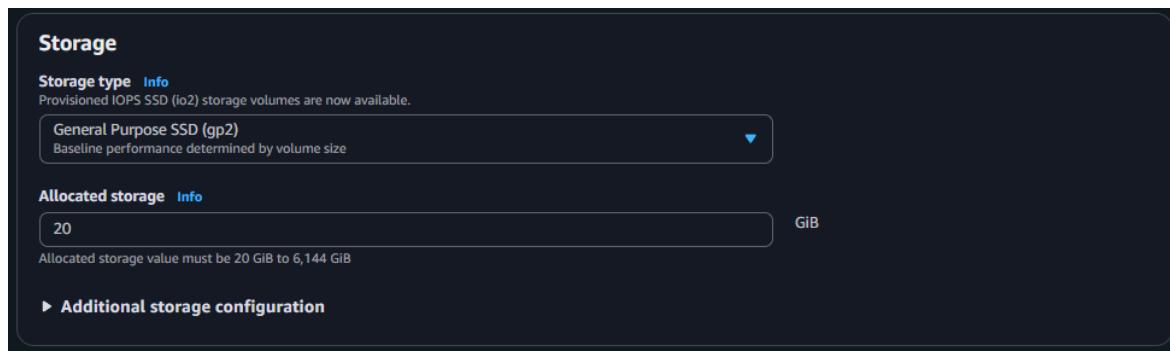
Include previous generation classes

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM EBS Bandwidth: Up to 2,085 Mbps Network: Up to 5 Gbps

5. En Storage, seleccionamos:

- **Type:** General Purpose (SSD)
- **Size:** 20GB



- Paso 2: Configurar red y seguridad
1. En la sección Connectivity:
 - **VPC:** seleccionamos ZT-VPC
 - **Subnet Group:** ZT-DB-Subnet-Group (Creado en fase 3)
 - **Public access:** No
 - **VPC Security group:** seleccionamos o creamos uno llamado ZT-RDS-SG
 - Reglas:
 - **Inbound:** Permitir tráfico TCP al puerto 3306, que es el puerto que seleccionamos.
 - **Source:** el security group del bastion (ZT-Bastion-SG)

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info
Choose the VPC. This VPC defines the virtual networking environment for this DB instance.

ZT-VPC (vpc-0000000000000000)
3 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

zt-db-subnet-group
2 Subnets, 2 Availability Zones

Public access Info
 Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
 No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups
Choose one or more options
ZT-SG-RDS X

Availability Zone Info
No preference

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-1 (default)
Expiry: May 25, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Database port Info
TCP/IP port that the database will use for application connections.
3306

2. Availability Zone: Esta parte no importa mucho, ya que se distribuirá en las dos subredes privadas.

- Paso 3: Cifrar con KMS
 1. En la sección Encryption:
 - Enable encryption: marcamos la casilla.
 - AWS KMS Key: Elegimos aws/rds o creamos una llave nueva asignando los permisos únicamente a nuestro usuario. En nuestro caso solo usaremos la llave default aws/rds.

2. Seleccionamos la clave aws/rds.

Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console.
[Info](#)

AWS KMS key Info
(default) aws/rds

- Paso 4: Finalizar la creación de la base de datos.

Solo hacemos click en **Create database** y esperamos a que cambio de estado a Available.

The screenshot shows a green success message at the top: "Successfully created database zt-db-instance". Below it, a note says: "You can use settings from zt-db-instance to simplify configuration of suggested database add-ons while we finish creating your DB for you." A "View connection details" button is present. The main interface is titled "Databases (1)". It includes a toolbar with "Group resources" (radio button), "Modify" (button), "Actions" (dropdown), and "Create database" (button). A search bar with "Filter by databases" placeholder is also shown. The table lists one database entry:

	DB identifier	Status	Role	
<input type="radio"/>	zt-db-instance	Available	Instance	M

- Paso 5: Probamos la conectividad segura
1. Desde el Session Manager, en Systems Manager, abrimos sesión en la instancia Bastion.
 2. y ejecutamos el siguiente comando:

```
sudo yum install -y mysql
```

Session ID: root-krtan4atclay97byyjrfyDyi Shortcuts Instance ID: i-078f58cd118bbcf93 Terminate

```

sh-5.2$ sudo yum install -y mysql
MySQL 8.0 Community Server
MySQL Connectors Community
MySQL Tools Community
Dependencies resolved.

=====
Package           Architecture Version      Repository   Size
=====
Installing:
mysql-community-client    x86_64      8.0.44-1.el9  mysql80-community  3.3 M
Installing dependencies:
mysql-community-client-plugins x86_64      8.0.44-1.el9  mysql80-community  1.4 M
mysql-community-common     x86_64      8.0.44-1.el9  mysql80-community  557 k
mysql-community-libs       x86_64      8.0.44-1.el9  mysql80-community  1.5 M

Transaction Summary
=====
Install 4 Packages

Total download size: 6.7 M
Installed size: 96 M
Downloading Packages:
(1/4): mysql-community-common-8.0.44-1.el9.x86_64.rpm          30 MB/s | 557 kB  00:00
(2/4): mysql-community-client-plugins-8.0.44-1.el9.x86_64.rpm  36 MB/s | 1.4 MB  00:00
(3/4): mysql-community-libs-8.0.44-1.el9.x86_64.rpm          45 MB/s | 1.5 MB  00:00
(4/4): mysql-community-client-8.0.44-1.el9.x86_64.rpm        13 MB/s | 3.3 MB  00:00

Total                                         27 MB/s | 6.7 MB  00:00
MySQL 8.0 Community Server                     3.0 MB/s | 3.1 kB  00:00

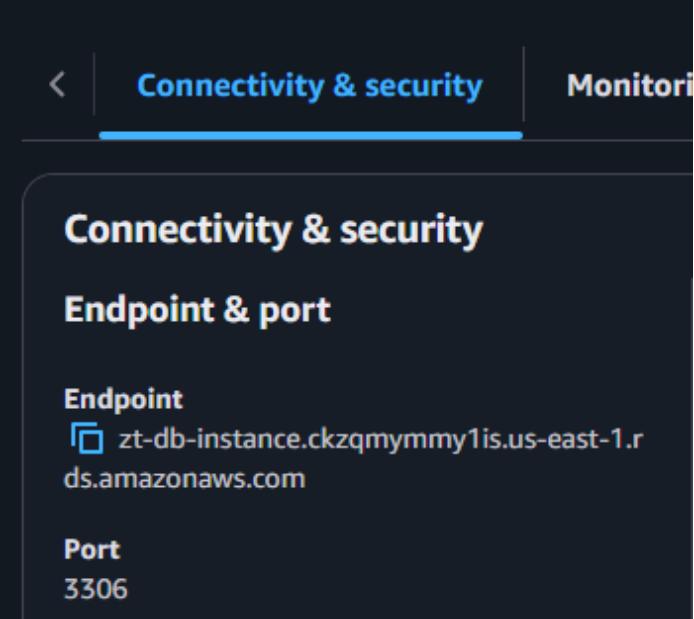
Importing GPG key 0x3A79BD29:
  Userid : "MySQL Release Engineering <mysql-build@oss.oracle.com>"
  Fingerprint: 859B EBD7 C586 F538 430B 19C2 467B 942D 3A79 BD29
  From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Key imported successfully
Import of key(s) didn't help, wrong key(s)?
Public key for mysql-community-client-8.0.44-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-client-8.0.44-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-client-plugins-8.0.44-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-client-plugins-8.0.44-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-common-8.0.44-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-common-8.0.44-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-libs-8.0.44-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-libs-8.0.44-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
The downloaded packages were saved in cache until the next successful transaction.
You can remove cached packages by executing 'yum clean packages'.
Error: GPG check FAILED
sh-5.2$ 

```

3. Luego nos conectamos ejecutando el siguiente comando:

```
mysql -h zt-db-instance.ckzqymmy1is.us-east-1.rds.amazonaws.com -u admin -p
```

(Encuentramos el <endpoint_RDS> en RDS → Databases → tu instancia → Connectivity & security)



```

sh-5.2$ mysql -h zt-db-instance.ckzqymmy1is.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

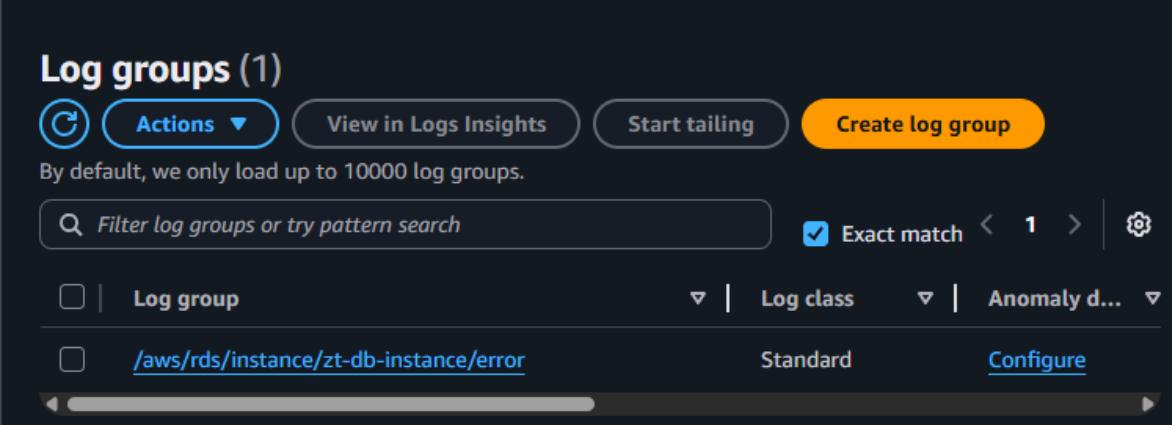
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>

```

Una vez conectados podemos verificar de que tenemos acceso seguro, cifrado y solo desde el bastión (EC2).

- Paso 6: Activar Auditoría y trazabilidad
 1. Vamos a **CloudWatch** → **Logs** → **Log groups** → /aws/rds/instance/...



Log group	Log class	Anomaly d...
/aws/rds/instance/zt-db-instance/error	Standard	Configure

Nota: En esta primera captura vemos un solo Log group ya que ese /aws/rds/instance/zt-db-instance/error se activa con cualquier error interno del motor, para generar automáticamente otros Log Groups es ejecutando queries reales desde el bastion, como por ejemplo cuando empecemos a crear tablas o consultas).

En este caso, realizamos unas queries, nos conectamos a la instancia desde Systems Manager y entramos a la base de datos, una vez logeados, ejecutamos el siguiente query:

```
CREATE DATABASE prueba;
USE prueba;
CREATE TABLE test(id INT PRIMARY KEY, name VARCHAR(50));
INSERT INTO test VALUES (1, 'Cristian');
SELECT * FROM test;
```

```
MySQL [(none)]> CREATE DATABASE prueba;
Query OK, 1 row affected (0.019 sec)

MySQL [(none)]> USE prueba;
Database changed
MySQL [prueba]> CREATE TABLE test(id INT PRIMARY KEY, name VARCHAR(50));
Query OK, 0 rows affected (0.061 sec)

MySQL [prueba]> INSERT INTO test VALUES (1, 'Cristian');
Query OK, 1 row affected (0.008 sec)

MySQL [prueba]> SELECT * FROM test;
+---+-----+
| id | name   |
+---+-----+
| 1  | Cristian |
+---+-----+
1 row in set (0.001 sec)

MySQL [prueba]>
MySQL [prueba]>
```

Una vez realizado el query, esperamos unos 2-3 minutos y luego revisamos CloudWatch → Log Groups y deberíamos poder ver los grupos nuevos que se generen.

- y así obtenemos trazabilidad centralizada de eventos.

¿Qué hemos logrado hasta ahora?

Componente	Propósito
RDS en subred privada	Sin Acceso público, solo interno
KMS encryption	Cifrado en reposo gestionado por clave simétrica
IAM y SG	Control de acceso granular entre Bastion y DB
Session Manager	Acceso sin SSH, con trazabilidad
CloudWatch	Monitoreo y auditoría bajo Zero Trust

- **FASE 6: Monitoreo Continuo (CloudWatch + CloudTrail + SNS)**

Objetivo

El objetivo en esta fase es el habilitar la visibilidad total de eventos críticos (inicios de sesión, cambios IAM, accesos fallidos) y recibir alertas automáticas usando SNS.

- Paso 1: Habilitar CloudTrail

1. Vamos a CloudTrail -> Trails -> Create Trail

2. Configuramos:

- **Trail name:** ZT-CloudTrail
- **Storage location:** creamos un nuevo bucket S3 -> zt-cloudtrail-logs
- **Apply CloudTrail to all regions:** marcamos el check point para activarlo.
- **Log File SSE encryption:** Aquí podemos usar la clave aws/s3 o nuestra llave KMS por si queremos más control.
- **Log Events:**
 - **Management events** -> Read/Write
 - **Data events** -> S3
 - **Insights events** -> Esta opción es más opcional, por si queremos un análisis más anómalo.

3. Clickeamos en **Create CloudTrail**.

Step 1: Choose trail attributes

General details

- Trail name: ZT-CloudTrail
- Multi-region trail: Yes
- Apply trail to my organization: Not enabled
- Trail log location: zt-cloudtrail-logs/AWSLogs/948531372386
- Log file SSE-KMS encryption: Enabled
- AWS KMS key alias: zt-cloudtrail-logs-kms
- Log file validation: Enabled
- SNS notification delivery: aws-cloudtrail-logs-948531372386-0fdf6b17

Step 2: Choose log events

Management events

No additional charges apply to log management events on this trail because this is your first copy of management events.

API activity: All	Exclude AWS KMS events: No
	Exclude Amazon RDS Data API events: No

Data events

Data events: S3

Log selector template: Log all events	Selector name: --
---------------------------------------	-------------------

Trails

Name	Home region	Multi-region trail	ARN	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group	Status
ZT-CloudTrail	US East (N. Virginia)	Yes	arn:aws:cloudtrail:us-east-1:948531372386:trail/ZT-CloudTrail	Disabled	No	zt-cloudtrail-logs	-	-	Logging

• Paso 2: Enviar logs a CloudWatch

1. Luego de crear el Trail, podemos clickear en nuestro Trail recién creado -> CloudWatch logs -> Edit -> Activamos CloudWatch Logs
2. Configuramos:
 - **Log Group Name:** zt-cloudtrail-logs
 - **IAM Role:** CloudTrail se encarga de crear un nuevo rol automáticamente llamado CloudTrail_CloudWatchLogs_Role
 - Clickeamos en **Save Changes**

CloudWatch Logs | Info

Enabled

Log group info

New
 Existing

Log group name

zt-cloudtrail-logs

1-512 characters. Only letters, numbers, dashes, underscores, forward slashes, and periods are allowed.

IAM Role info

AWS CloudTrail assumes this role to send CloudTrail events to your CloudWatch Logs log group.

New
 Existing

Role name

CloudTrailRoleForCloudWatchLogs_(trail-name)

▶ Policy document

Cancel Save changes

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

• Paso 3: Crear filtro de métrica en CloudWatch

1. Vamos a **Cloudwatch -> Log groups -> /aws/cloudtrail/..**

2. En la pestaña Metric Filters, hacemos click en:

- **Create Metric Filter**

3. Pegamos este patrón:

```
{ ($.eventName = "ConsoleLogin") && ($.errorMessage = "Failed authentication") }
```

4. Hacemos click en Next y lo nombramos **FailedConsoleLogins**

5. Y creamos una métrica bajo el namespace: **SecurityMetrics**

- **Metric name:**
- **Metric Value: 1**

Metric details

Metric namespace

Namespaces let you group similar metrics. [Learn more](#)

SecurityMetrics Create new

Namespaces can be up to 255 characters long; all characters are valid except for colon(:) at the start of the name.

Metric name

Metric name identifies this metric, and must be unique within the namespace. [Learn more](#)

FailedConsoleLogins

Metric name can be up to 255 characters long; all characters are valid except for colon(:), asterisk(*), dollar(\$), and space().

Metric value

Metric value is the value published to the metric name when a Filter Pattern match occurs.

1

Valid metric values are: floating point number (1, 99.9, etc.), numeric field identifiers (\$1, \$2, etc.), or named field identifiers (e.g. \$requestSize for delimited filter pattern or \$.status for JSON-based filter patterns). All characters must be alphanumeric and/or underscore (_) characters.

Default value – optional

The default value is published to the metric when the pattern does not match. If you leave this blank, no value is published when there is no match. [Learn more](#)

Enter default value

Unit – optional

Select a unit ▾

6. En Dimensions marcamos ambas casillas como se muestra a continuación y clickeamos next.

Dimensions

A dimension is a name/value pair that is a part of the identity of a metric. You can assign up to 3 dimensions to a metric. Dimension names and values can be up to 255 characters, cannot contain blank spaces or non-ASCII characters. [Learn more](#)

System fields

Select system fields to include as dimensions

@aws.account

@aws.region

Custom fields

Add up to 3 custom dimension fields

No custom fields added yet

[Add new field](#)

Cancel Previous **Next**

Log groups (2)

By default, we only load up to 10,000 log groups.

Log group	Log class	Anomaly detecti...	D...	S...	Retention	Metric filters	Contrib...
/aws/rds/instance/zt-db-instance/error	Standard	Configure	-	-	Never expire	-	-
zt-cloudtrail-logs	Standard	Configure	-	-	Never expire	1 filter	-

- **Paso 4: Crear Alarma y SNS**

1. En **CloudWatch**, ubicamos el panel izquierdo → **Alarms** → **Create Alarm**.

CloudWatch > Alarms

CloudWatch

Favorites and recents

Dashboards

▶ AI Operations [New](#)

▼ Alarms [△ 0](#) [○ 0](#) [⋯ 0](#)

In alarm

All alarms [New](#)

Billing

▶ Logs

▶ Metrics [New](#)

▶ Application Signals (APM) [New](#)

Alarms (0)

Hide Auto Scaling alarms

[Create alarm](#)

Search

Alarm state: In alarm

Alarm type: Any

Actions status: Any

Name State Last state update (UTC) Conditions

No alarms

No alarms to display

[Read more about Alarms](#)

[Create alarm](#)

2. Seleccionamos la métrica FailedConsoleLogins

3. Condición:

- Threshold: ≥ 1
- Period: 5 Minutes

The screenshot shows the 'Specify metric and conditions' section of the AWS CloudWatch Metrics Insights interface. On the left, there is a graph titled 'Metric' showing 'FailedConsoleLogins' over time from 03:00 to 05:30. A blue line represents the metric, and a red line indicates the threshold at 1. On the right, configuration fields are shown:

- Namespace:** SecurityMetrics
- Metric name:** FailedConsoleLogins
- Statistic:** Sum (selected)
- Period:** 5 minutes

Below this, the 'Conditions' section is displayed:

- Threshold type:** Static (selected)
- Whenever FailedConsoleLogins is...** Greater/Equal (\geq threshold) (selected)
- than...** 1 (threshold value input field)

4. En notificación, creamos un nuevo SNS topic

- Name: security-alerts
- Email endpoint: nuestro correo

Configure actions

Notification

Alarm state trigger

Define the alarm state that will trigger this action.

[Remove](#)

In alarm

The metric or expression is outside of the defined threshold.

OK

The metric or expression is within the defined threshold.

Insufficient data

The alarm has just started or not enough data is available.

Send a notification to the following SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN to notify other accounts

Create a new topic...

The topic name must be unique.

security-alerts

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification...

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

user1@example.com, user2@example.com

[Create topic](#)

5. Confirmamos nuestro correo desde nuestra bandeja de entrada.

AWS Notification - Subscription Confirmation

AWS Notifications AN Para: CRISTIAN JIMENEZ Sáb 25/10/2025 0:52

No suele recibir correo electrónico de no-reply@sns.amazonaws.com. [Por qué es esto importante](#)

You have chosen to subscribe to the topic:
arn:aws: [REDACTED]

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws: [REDACTED]

If it was not your intention to subscribe, [click here to unsubscribe](#).

6. Click en Create alarm.

The screenshot shows the AWS CloudWatch Alarms page. At the top, there's a header with the title "Alarms (1/1)". Below the header are several buttons: "Hide Auto Scaling alarms" (unchecked), "Clear selection" (highlighted in blue), "Create composite alarm" (with a "C" icon), "Actions" (with a dropdown arrow), and "Create alarm". There are also search and filter fields for "Search", "Alarm state: Any", "Alarm type: Any", and "Actions status: Any". On the right side of the header, there are navigation icons for pages and a refresh symbol. The main table below has columns: "Name", "State", and "Last state update (UTC)". A single row is visible, showing "Security Alert" with an "Insufficient data" status and an update time of "2025-10-25 05:56:30".

- Paso 5: Simulación
 1. Intentamos iniciar sesión a AWS con una contraseña errónea.

The screenshot shows the AWS Root user sign in page. At the top, there's a red-bordered error message box containing a red circle with a white "X" icon and the text "Authentication failed: Your authentication information is incorrect. Please try again." Below the error message, the page title is "Root user sign in" with a help icon. The form asks for the password with the placeholder "(not you?)". There is a "Password" input field, a "Show password" checkbox, a "Forgot password?" link, and a large orange "Sign in" button at the bottom.

2. Esperamos de 2 a 5 minutos.

Alerta generada en Cloudwatch:



3. Verificamos que nos llega el correo de alerta.

Alerta en estado "In Alarm"

AN AWS Notifications<no-reply@sns.amazonaws.com>
Para: CRISTIAN JIMENEZ
Sáb 25/10/2025 16:48

[No suele recibir correo electrónico de no-reply@sns.amazonaws.com. Descubra por qué esto es importante en <https://aka.ms/LearnAboutSenderIdentification>]

You are receiving this email because your Amazon CloudWatch Alarm "Security Alert" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [1.0 (25/10/25 21:43:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Saturday 25 October, 2025 21:48:11 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/Security%20Alert>

Alarm Details:

- Name: Security Alert
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [1.0 (25/10/25 21:43:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Saturday 25 October, 2025 21:48:11 UTC
- AWS Account: [REDACTED]
- Alarm Arn: arn:aws:cloudwatch:us-east-1:alarm:Security%20Alert

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

- MetricNamespace: SecurityMetrics
- MetricName: FailedConsoleLogins
- Dimensions:
- Period: 300 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

Efectivamente verificamos que el correo nos llegó unos minutos después del evento.

- **Conclusión**

La implementación realizada permitió construir un entorno AWS seguro y trazable alineado a los principios del modelo Zero Trust. Durante las fases se configuraron los servicios fundamentales (VPC, EC2, RDS, IAM, CloudTrail, CloudWatch y SNS) garantizando control de acceso, visibilidad de eventos y respuesta ante incidentes. La arquitectura resultante integra administración de políticas (IAM y KMS), evaluación y monitoreo continuo (CloudWatch y CloudTrail) y mecanismos de alerta (SNS), cumpliendo los lineamientos del NIST SP 800-207. Este entorno sirve como base para futuras pruebas de detección y respuesta, así como para prácticas de automatización de políticas adaptativas y análisis de seguridad en la nube.