

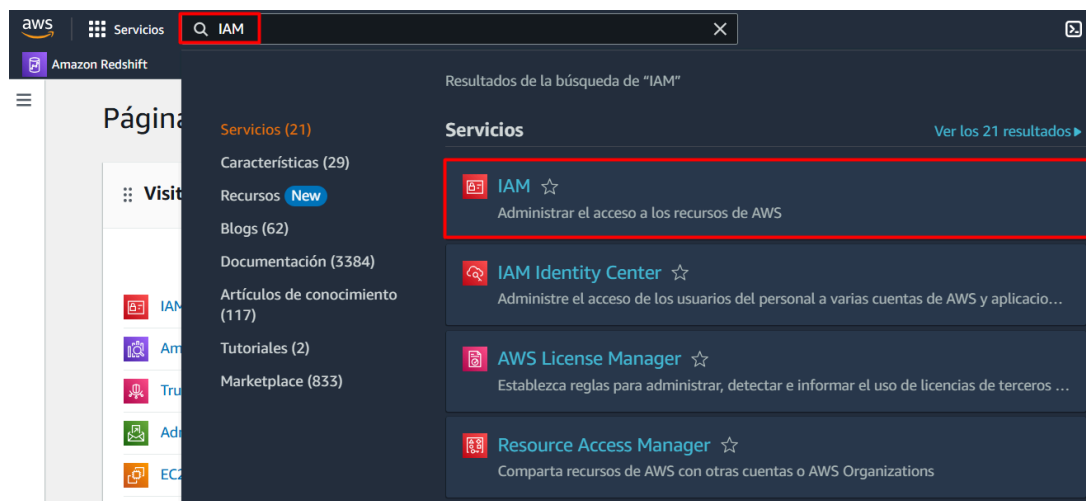
LABORATORIO DE AWS IAM, KMS Y SECRETS MANAGER

En este laboratorio vamos a crear usuarios y grupos, a agregar usuarios a los grupos y a asignar políticas a los usuarios y a los grupos.

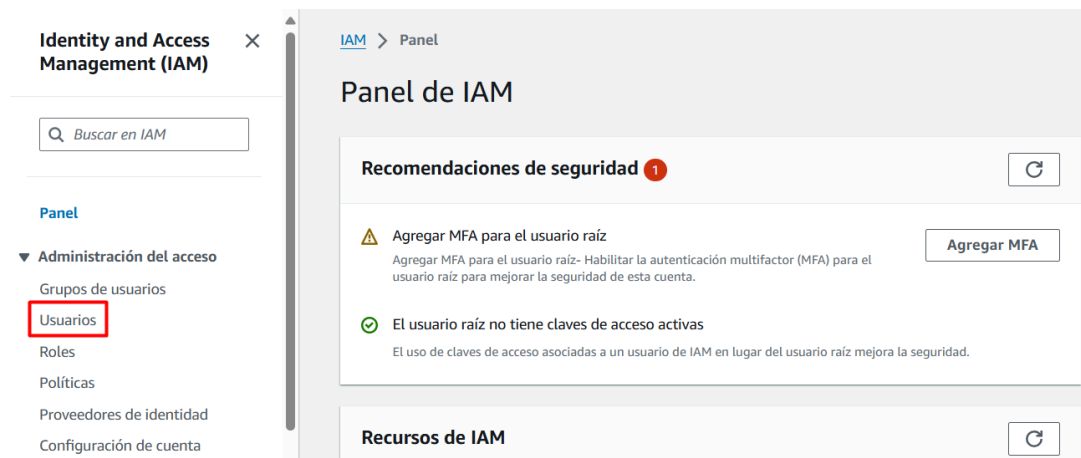
Además, vamos a crear una key con KMS y a proteger secretos con Secrets Manager.

USUARIOS

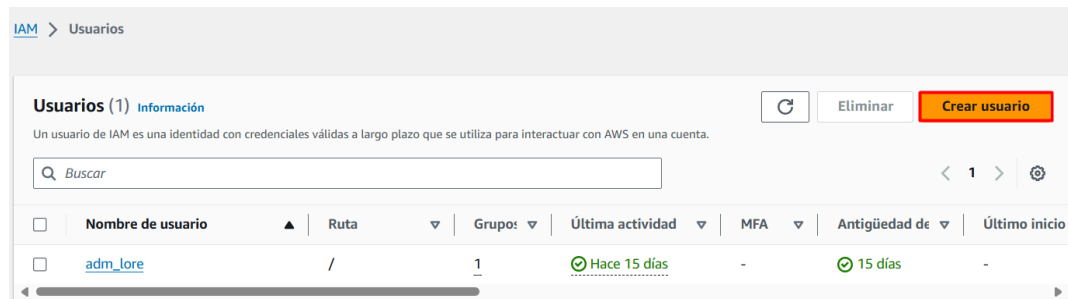
1. Iniciamos sesión en la consola de administración de AWS e ingresamos a la consola de IAM:



2. En el menú lateral, seleccionamos **Usuarios**:



3. Damos clic en **Crear usuario**:



4. Especificar los detalles del usuario:

- Nombre de usuario: El nombre de usuario puede tener un máximo de 64 caracteres. Caracteres válidos: A-Z, a-z, 0-9, and + = , . @ _ - (guion). En este ejemplo usamos **test_betek1**, puedes usar tu nombre si deseas.
- Proporcione acceso de usuario a la consola de administración de AWS: **Check (marcar)**.
La consola de administración es la interfaz gráfica de AWS, por la que accedemos a los servicios a través del navegador del web.
- Seleccionar **Quiero crear un usuario de IAM**.
- Cuando vamos a crear la contraseña de la consola, tenemos dos opciones, Contraseña generada automáticamente y Contraseña personalizada. Para este laboratorio vamos seleccionar **Contraseña generada automáticamente**.
- Podemos determinar que los usuarios deban crear una nueva contraseña en el siguiente inicio de sesión, esta es una buena práctica recomendada de seguridad: **Marcar (check)**.
- Clic en **Siguiente**.

Nombre de usuario

1 test_betek1

El nombre de usuario puede tener un máximo de 64 caracteres. Caracteres válidos: A-Z, a-z, 0-9, and +, -, @, _ - (guion)

2 ☒ Proporcionar acceso de usuario a la consola de administración de AWS: *opcional*
Si proporciona acceso a la consola a una persona, se trata de un [práctica recomendada](#) para administrar su acceso en IAM Identity Center.

3 ☒ ¿Está proporcionando acceso a la consola a una persona?

Tipo de usuario

☐ Especificar un usuario en Identity Center: **recomendado**
Le recomendamos que utilice Identity Center para proporcionar acceso a la consola a una persona. Con Identity Center, puede administrar de forma centralizada el acceso de los usuarios a sus cuentas de AWS y aplicaciones en la nube.

☒ Quiero crear un usuario de IAM
Le recomendamos que cree usuarios de IAM solo si necesita habilitar el acceso mediante programación a través de claves de acceso, credenciales específicas de servicios para AWS CodeCommit o Amazon Keyspaces o una credencial de copia de seguridad para el acceso a la cuenta de emergencia.

4 Contraseña de la consola

☒ Contraseña generada automáticamente
Puede ver la contraseña después de crear el usuario.

☐ Contraseña personalizada
Ingrese una contraseña personalizada para el usuario.

5 ☐ Mostrar contraseña

☒ Los usuarios deben crear una nueva contraseña en el siguiente inicio de sesión (recomendado).
Los usuarios obtienen automáticamente la [IAMUserChangePassword](#) política para poder cambiar su propia contraseña.

6 Si está creando acceso mediante programación a través de claves de acceso o credenciales específicas de servicios para AWS CodeCommit o Amazon Keyspaces, puede generarlos después de crear este usuario de IAM. [Más información](#)

Cancelar **6** **Siguiente**

5. Establecer permisos: En este paso vamos a indicar qué acciones podrá realizar este usuario sobre los recursos de AWS:

- Podemos agregar el usuario a un grupo, de esta manera el usuario heredará los permisos del grupo. Se recomienda utilizar grupos para administrar los permisos de usuario según las funciones laborales.
- También podemos copiar los permisos de otro usuario ya existente.
- O adjuntar una política administrada a un usuario de manera directa. Como práctica recomendada, se sugiere adjuntar políticas a un grupo y luego agregar el usuario al grupo adecuado.

Por ahora no le vamos a dar ningún tipo de permiso a este usuario y damos clic en **Siguiente**:

Opciones de permisos

☒ Agregar usuario al grupo
Agregue el usuario a un grupo existente o cree uno nuevo. Le recomendamos que utilice grupos para administrar los permisos de usuario según las funciones laborales.

☐ Copiar permisos
Cople todas las suscripciones a grupos, las políticas administradas adjuntas y las políticas insertadas de un usuario existente.

☐ Adjuntar políticas directamente
Adjunte una política administrada a un usuario de manera directa. Como práctica recomendada, le sugerimos, en cambio, adjuntar políticas a un grupo. A continuación, agregue el usuario al grupo adecuado.

Grupos de usuarios (1)

Buscar

<input type="checkbox"/>	Nombre del grupo	Usuarios	Políticas adjuntas	Creado
<input type="checkbox"/>	admin_users	1	AmazonEC2FullAccess, Amaz...	2024-07-07 (Hace 15 días)

► Establecer límite de permisos: *opcional*

Cancelar Anterior **Siguiente**

- Revisar y crear: En este paso revisamos la configuración del usuario que vamos a crear, el nombre es **test_betek1**, el tipo de contraseña es autogenerada, sí requiere un restablecimiento de la contraseña y, el único permiso que tiene es el de cambiar su propia contraseña, esto es para que él pueda cambiar su contraseña la primera vez que ingrese a AWS.

Después de revisar, damos clic en **Crear usuario**:

Revisar y crear

Revise las opciones seleccionadas. Después de crear el usuario, puede ver y descargar la contraseña autogenerada, si está habilitada.

Detalles del usuario

Nombre de usuario test_betek1	Tipo de contraseña de consola Autogenerated	Exigir el restablecimiento de la contraseña Sí
----------------------------------	--	---

Resumen de permisos

Nombre	Tipo	Usado como
IAMUserChangePassword	Administrada por AWS	Política de permisos

Etiquetas : opcional

Las etiquetas son pares clave-valor que puede agregar a los recursos de AWS para ayudar a identificar, organizar o buscar recursos. Elija las etiquetas que desee asociar a este usuario.

No hay etiquetas asociadas al recurso.

[Agregar nueva etiqueta](#)

Puede agregar hasta 50 etiqueta más.

Cancelar Anterior **Crear usuario**

- Una vez tengamos el mensaje de que el usuario se creó correctamente, descargamos las credenciales de este nuevo usuario:

El usuario se ha creado correctamente [Ver usuario](#)

Puede ver y descargar la contraseña del usuario y las instrucciones de correo electrónico para iniciar sesión en la Consola de administración de AWS.

[IAM](#) > [Usuarios](#) > [Crear usuario](#)

Paso 1
[Especificar los detalles del usuario](#)

Paso 2
[Establecer permisos](#)

Paso 3
[Revisar y crear](#)

Paso 4
Recuperar contraseña

Recuperar contraseña

Puede ver y descargar la contraseña del usuario a continuación o enviar por correo electrónico instrucciones a los usuarios para iniciar sesión en la consola de administración de AWS. Esta es la única vez que puede ver y descargar esta contraseña.

Detalles de inicio de sesión en la consola

Instrucciones de inicio de sesión por correo electrónico [✉](#)

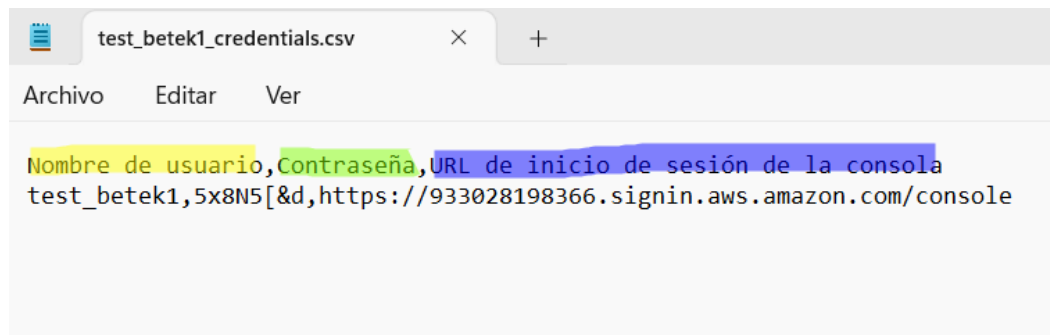
URL de inicio de sesión de la consola
<https://933028198366.signin.aws.amazon.com/console>

Nombre de usuario
test_betek1

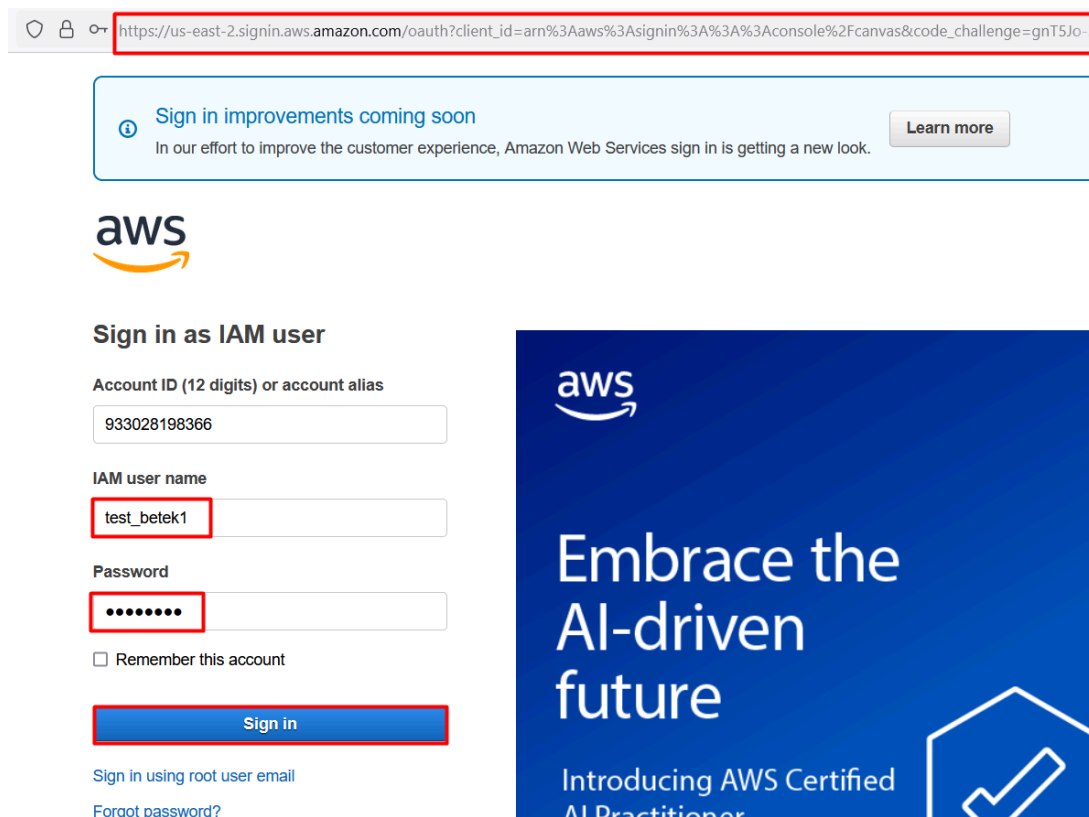
Contraseña de la consola
***** [Mostrar](#)

Cancelar **Descargar archivo.csv** [Volver a la lista de usuarios](#)


- El archivo que descargamos, en formato CSV, tiene tres parámetros separados por coma: Nombre de usuario, contraseña y URL del inicio de sesión, con estos tres datos nuestro usuario ya puede ingresar a la consola de AWS:



Para evitar conflictos entre las sesiones, pegamos la URL en otro navegador e ingresamos con los datos del CSV descargado:



Como se lo indicamos en la creación del usuario, nos va a pedir cambio de contraseña, ingresamos la actual (del CSV) y establecemos la nueva:



You must change your password to continue

AWS account 933028198366

IAM user name test_betek1

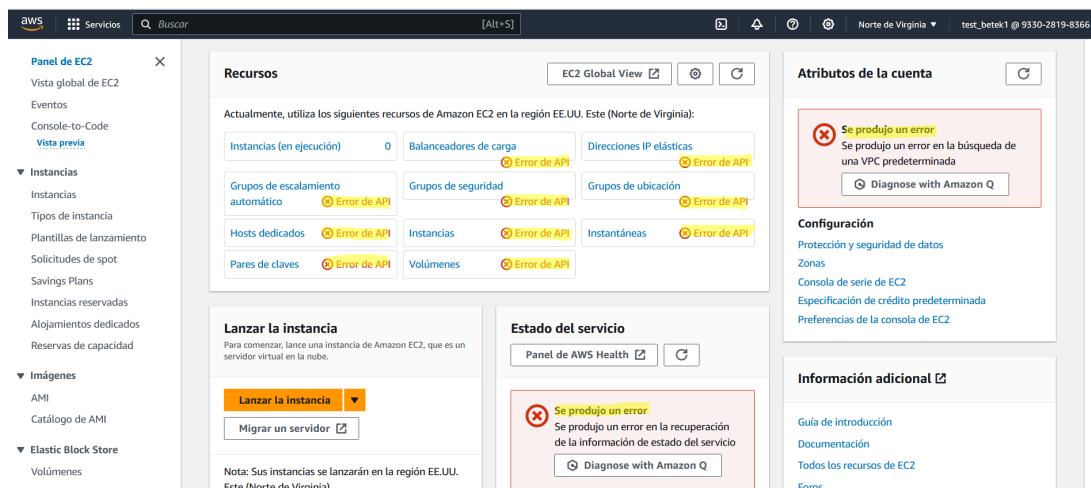
1 Old password

2 New password

3 Retype new password

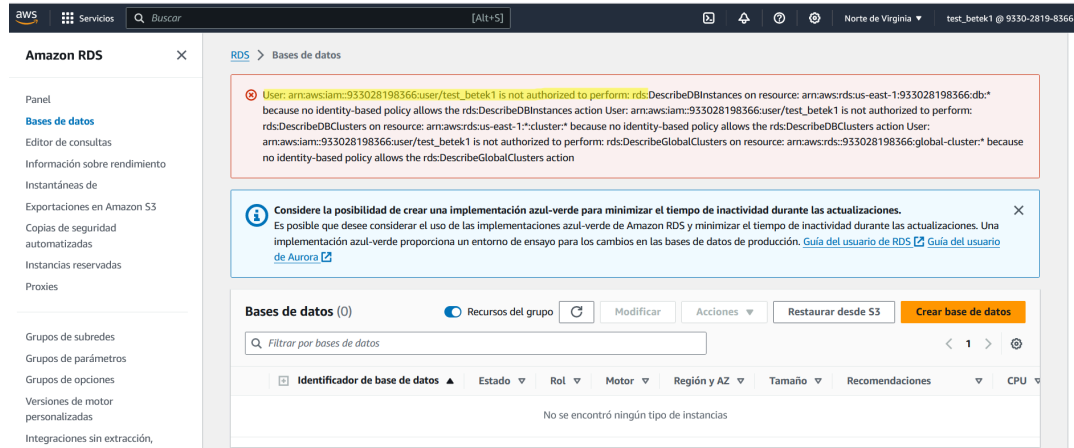
4 Confirm password change

Si navegamos a la consola de EC2, nos va a salir error, no podemos ver las instancias, los balanceadores de carga, nada:



The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for services like EC2, IAM, and VPC. The main content area displays several error messages, all indicating 'Error de API' (API Error). These errors are present in the 'Recursos' (Resources) section, the 'Lanzar la instancia' (Launch Instance) section, and the 'Estado del servicio' (Service Status) section. The 'Atributos de la cuenta' (Account Attributes) section also shows an error related to VPC search. The 'Información adicional' (Additional Information) section provides links to guides, documentation, and forums.

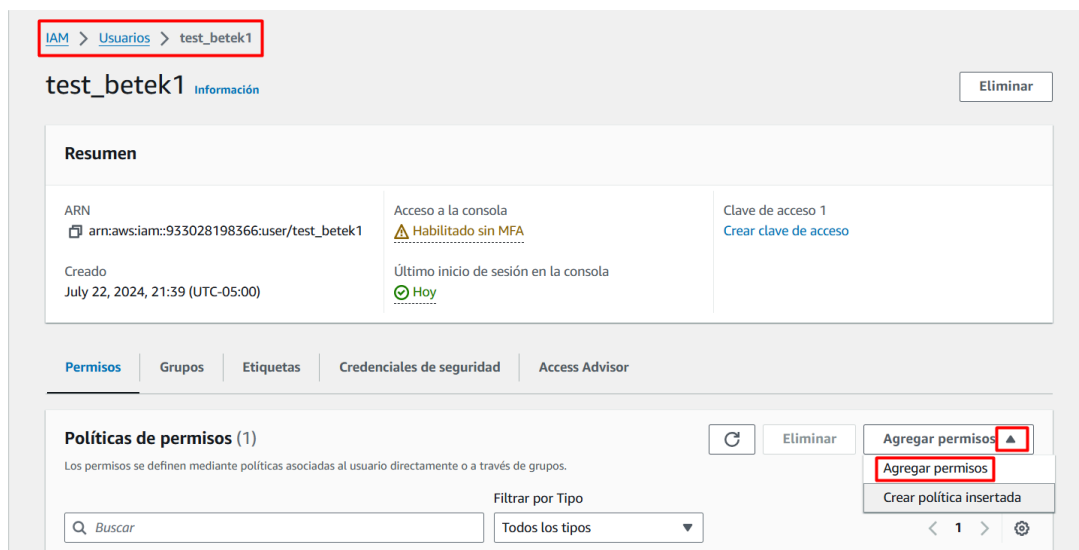
Lo mismo sucede si navegamos en la consola de RDS, o en cualquier otro servicio, no podemos ver nada:



Esto se debe a que el usuario no tiene permiso en EC2, ni en RDS, ni en ningún recurso de AWS, el único permiso que tiene el usuario es el de cambiar su propia contraseña.

Para solucionar esto, debemos ir a la cuenta administradora y agregar los permisos al usuario para poder realizar estas acciones.

- En la consola del usuario root, vamos al servicio **IAM**, ingresamos a la opción **Usuarios**, buscamos el usuario que creamos (en este ejemplo **test_betek1**), desplegamos el menú Agregar permisos y seleccionamos **Agregar permisos**:



Vamos a adjuntarle una política que ya existe con los permisos requeridos: Seleccionamos **Adjuntar políticas directamente**, seleccionamos **AdministratorAccess**, esta política le va a dar acceso al usuario a todos los servicios de AWS y va a poder realizar cualquier acción:

Agregar permisos
Agregue un usuario a un grupo existente o cree uno nuevo. El uso de grupos es una práctica recomendada para administrar los permisos de usuario según las funciones laborales. [Más información](#)

Opciones de permisos

☐ **Agregar usuario al grupo**
Agregue el usuario a un grupo existente o cree uno nuevo. Le recomendamos que utilice grupos para administrar los permisos de usuario según las funciones laborales.

☐ **Copiar permisos**
Copie todas las suscripciones a grupos, las políticas administradas adjuntas, las políticas insertadas y todos los límites de permisos existentes de un usuario existente.

☒ **Adjuntar políticas directamente**
Adjunte una política administrada a un usuario de manera directa. Como práctica recomendada, le sugerimos, en cambio, adjuntar políticas a un grupo. A continuación, agregue el usuario al grupo adecuado.

Políticas de permisos (1/1227)

Buscar Filtrar por Tipo Todos los tipos

	Nombre de la política	Tipo	Entidades asociadas
<input type="checkbox"/>	AccessAnalyzerServiceRolePolicy	Administrada por AWS	0
<input checked="" type="checkbox"/>	AdministratorAccess	Administrada por AWS: función de trabajo	1
<input type="checkbox"/>	AdministratorAccess-Amplify	Administrada por AWS	0
<input type="checkbox"/>	AdministratorAccess-AWSFaaS...	Administrada por AWS	0

Al final de la página, damos clic en **Siguiente** y luego en **Agregar permisos**:

Revisar
Las siguientes políticas se asociarán a este usuario. [Más información](#)

Detalles del usuario

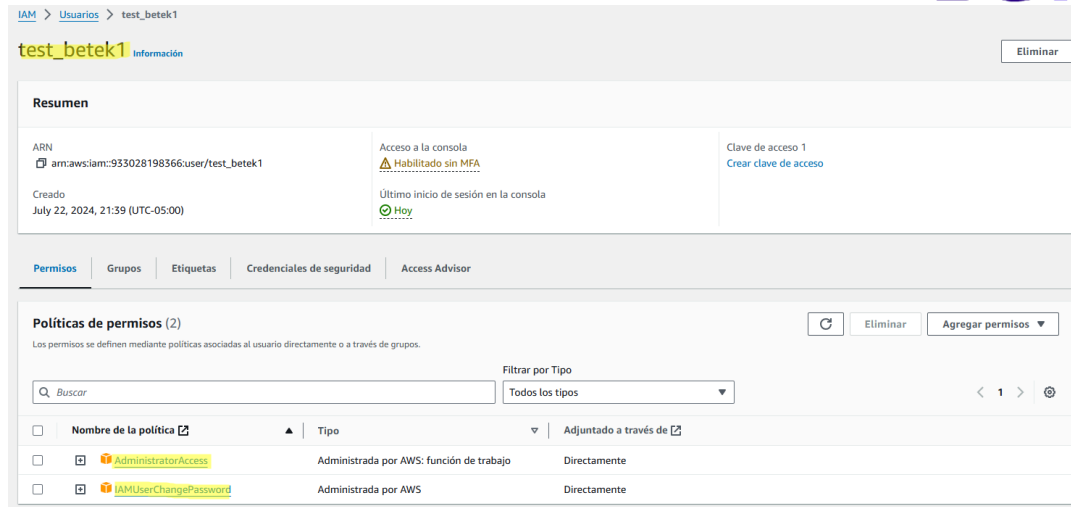
Nombre de usuario
test_betek1

Resumen de permisos (1)

Nombre	Tipo	Usado como
AdministratorAccess	Administrada por AWS: función de trabajo	Política de permisos

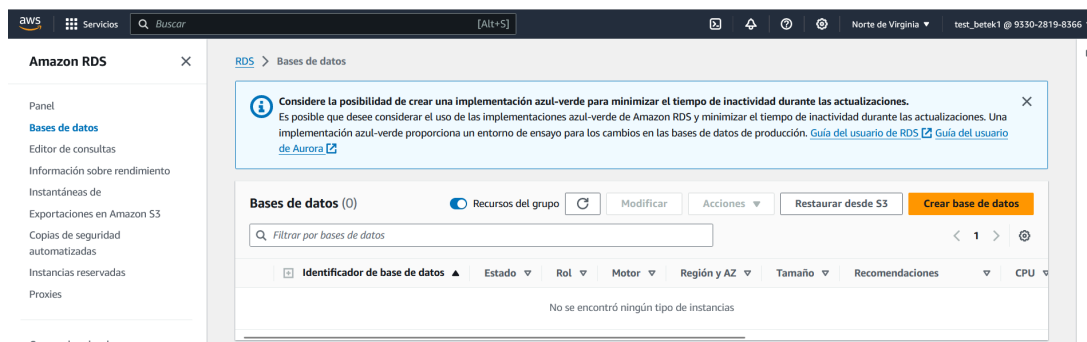
Cancelar Anterior **Agregar permisos**

Ahora vemos que el usuario tiene el permiso FullAccess y el permiso de cambiar su contraseña:

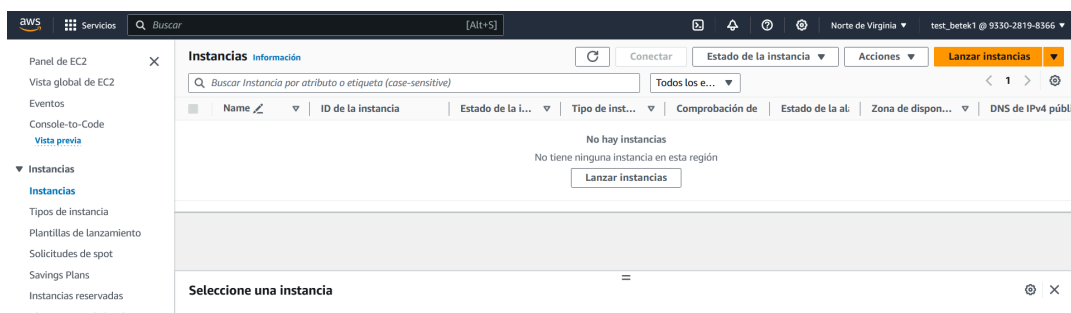


The screenshot shows the AWS IAM console for the user **test_betek1**. The **Resumen** (Summary) section displays the user's ARN, creation date (July 22, 2024), and access status. The **Políticas de permisos** (Permissions policies) section shows two policies: **AdministratorAccess** and **IAMUserChangePassword**, both administered by AWS.

10. Regresamos a la sesión del usuario **test_betek1**, damos F5 o refrescamos la página y ahora no tenemos los mensajes de error que teníamos antes de asignar los permisos:



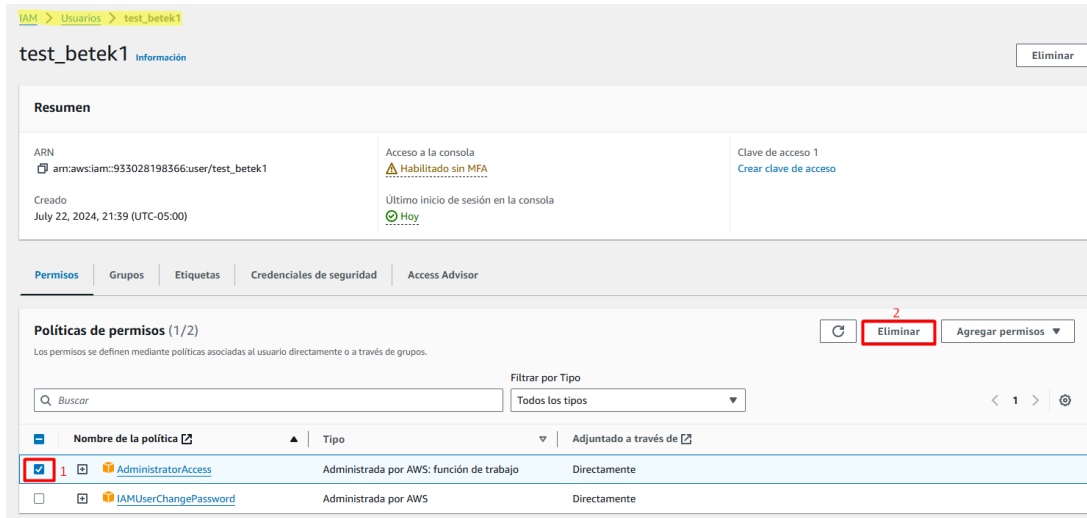
The screenshot shows the AWS RDS console. A message at the top suggests creating a blue-green implementation for Amazon RDS. The **Bases de datos** (Databases) section shows a table with no instances, indicating that the user now has the necessary permissions to view the RDS instances.



The screenshot shows the AWS EC2 console. The **Instancias** (Instances) section shows a table with no instances, indicating that the user now has the necessary permissions to view the EC2 instances.

El usuario no ve bases de datos RDS ni instancias EC2 porque aún no ha creado ninguna, pero ya no tiene mensajes de error por falta de permisos, como antes.

11. Ahora vamos a regresar a la cuenta del usuario root y le vamos a quitar los permisos de administrador al usuario **test_betek1**:



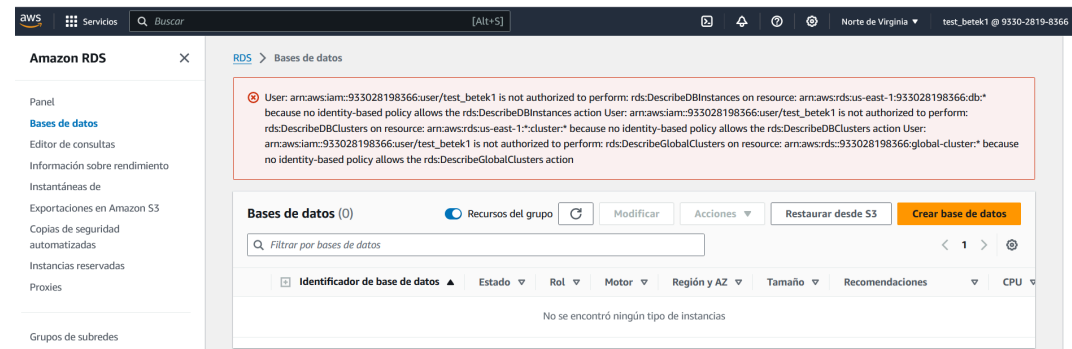
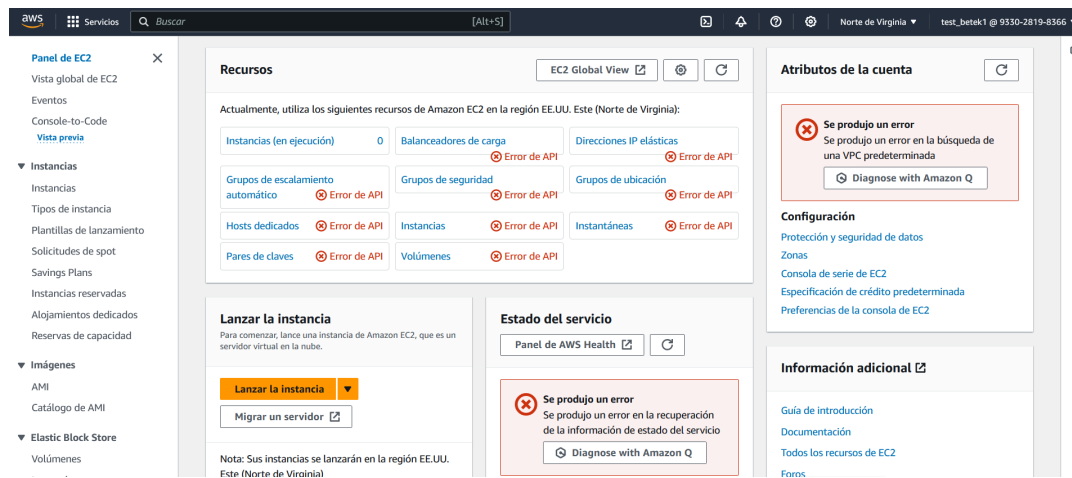
The screenshot shows the AWS IAM console for user **test_betek1**. The 'Permisos' tab is active, displaying a table of policies. The first policy, **AdministratorAccess**, is selected with a checkbox. A red box highlights the 'Eliminar' button in the top right corner of the policy list, with a red number '2' above it.

Confirmamos que queremos eliminar la política:



The screenshot shows a confirmation dialog box titled '¿Desea eliminar la política para el usuario?'. The dialog asks '¿Desea eliminar la política **AdministratorAccess**?'. At the bottom right, the 'Eliminar política' button is highlighted with a red box.

12. Si regresamos a la sesión del usuario **test_betek1**, damos F5 o refrescamos la página ya no debería tener acceso a ver los recursos de EC2, ni de RDS, ni ningún otro recurso:

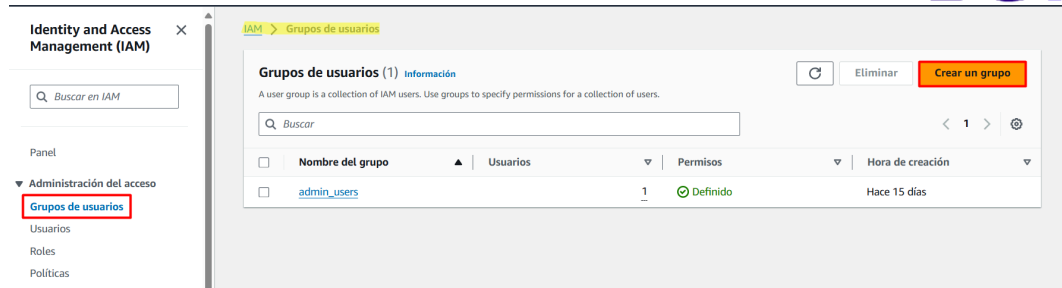


GRUPOS

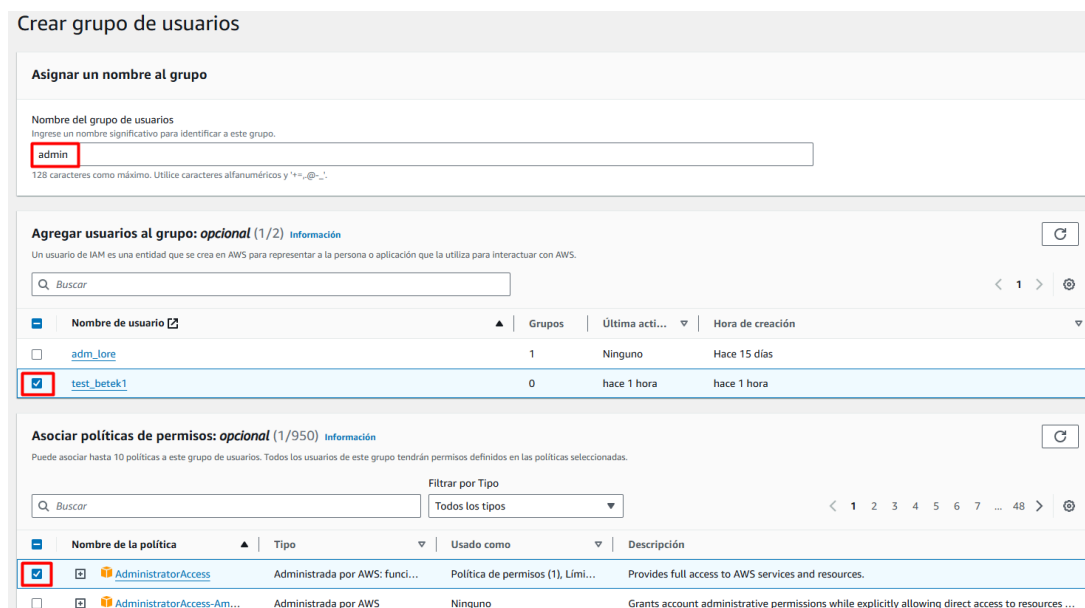
Para evitar esto de estar asignando y quitando permisos a los usuarios, lo más recomendable es, primero crear grupos de usuarios y luego agregar los usuarios al grupo que le corresponda, según los permisos que requiera.

En este caso, creamos un grupo de administradores y a todos los usuarios que sean administradores dentro de nuestra empresa, los agregamos a este grupo.

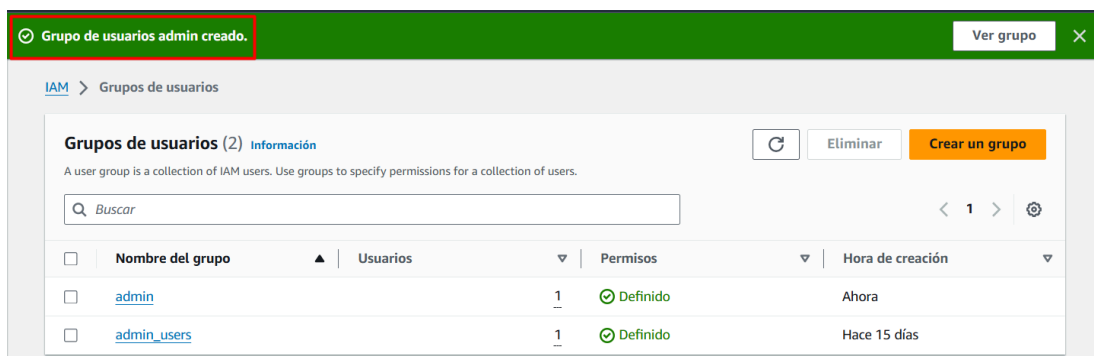
13. En la sesión del usuario root, vamos a la consola de **IAM**, seleccionamos **Grupos de usuarios** y damos clic en **Crear un grupo**:



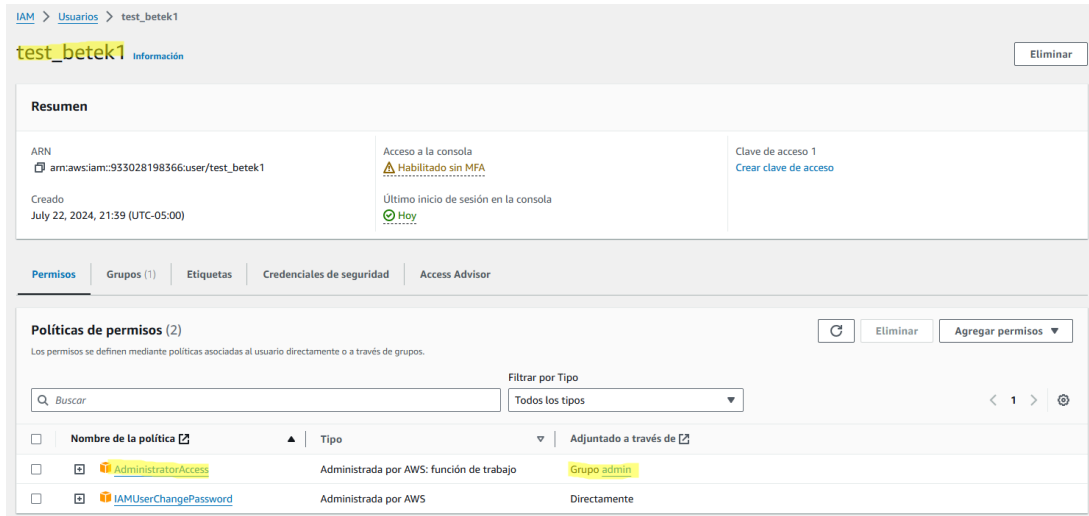
Asignamos un nombre al grupo de usuarios, en este caso será **admin**, si ya tenemos usuarios creados para agregar al grupo, los seleccionamos, asignamos la política **AdministratorAccess** y al final de la página damos clic en **Crear grupo de usuarios**:



Y así de fácil ya tenemos creado el grupo de administradores, donde tenemos el usuario **test_betek1**:

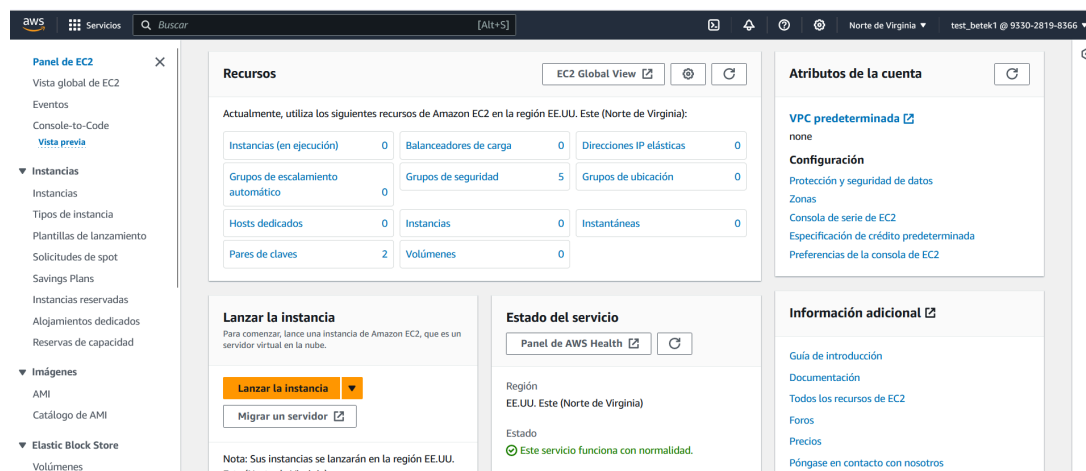


14. Regresamos al usuario **test_betek1** y encontramos que nuevamente el usuario tiene la política de administrador, pero esta vez no se le agregó directamente sino a través de un grupo:

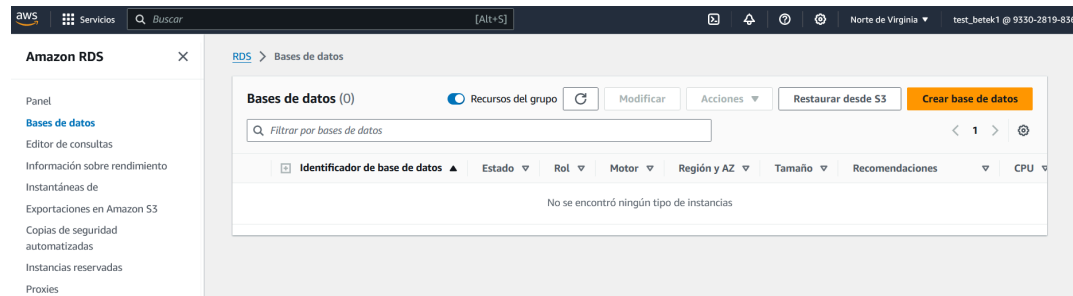


The screenshot shows the AWS IAM console for the user **test_betek1**. The user's ARN is `arn:aws:iam::933028198366:user/test_betek1`. The user is associated with the **Grupo admin** group, which has the **AdministratorAccess** policy. The user's last sign-in was on July 22, 2024, at 21:39 (UTC-05:00).

Si regresamos a la sesión del usuario **test_betek1**, damos F5 y refrescamos la página, vamos a ver cómo este usuario otra vez puede ver todos los recursos (EC2, RDS y demás servicios de AWS) porque nuevamente tiene los permisos para ello:



The screenshot shows the AWS Management Console for the user **test_betek1**. The user is logged in, and the console displays various AWS resources including EC2 instances, S3 buckets, and IAM roles. The 'Recursos' section shows a list of resources with their status and details. The 'Estado del servicio' section shows the status of the AWS services, indicating that the services are functioning normally.

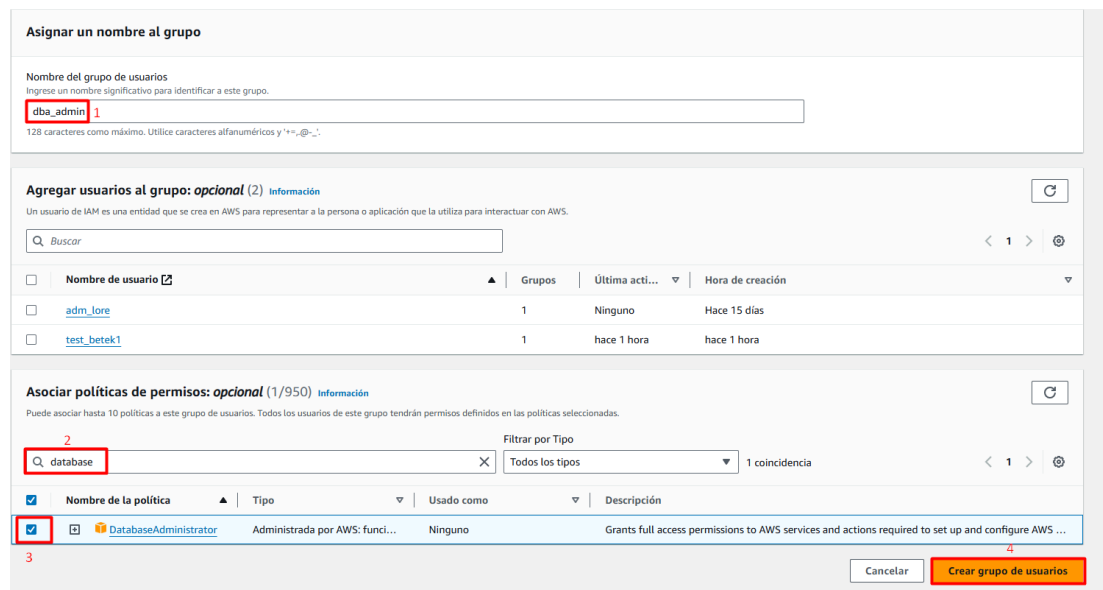


15. Regresemos a la consola de IAM y vamos a crear otro grupo.

Supongamos que tenemos algunos administradores de bases de datos y queremos crear un grupo para ellos.

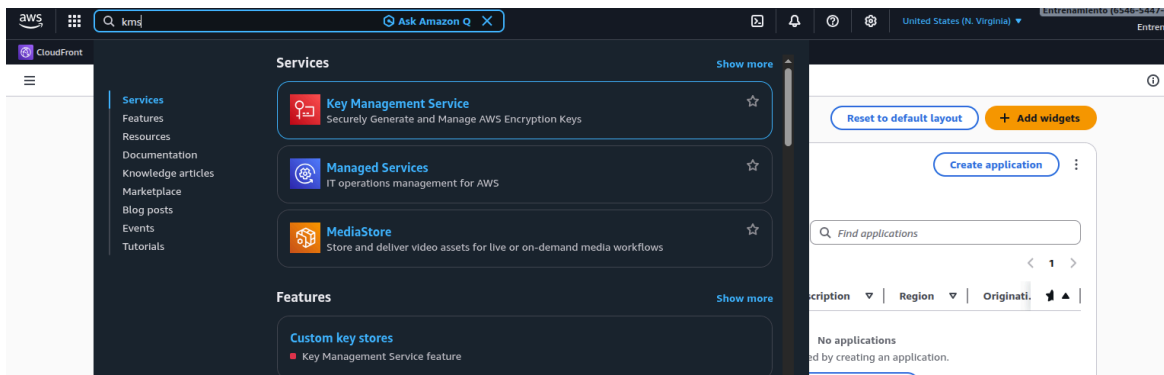
El grupo se va a llamar **dba_admin**, agregamos los usuarios (en este caso no tenemos usuarios para agregar porque el usuario test_betek1 ya tiene full access y no tiene sentido agregarlo a este grupo), buscamos la política **DatabaseAdministrator**, la seleccionamos y automáticamente se agregan los permisos necesarios para administrar bases de datos en la cuenta de AWS.

Damos clic en **Crear grupo de usuarios**:

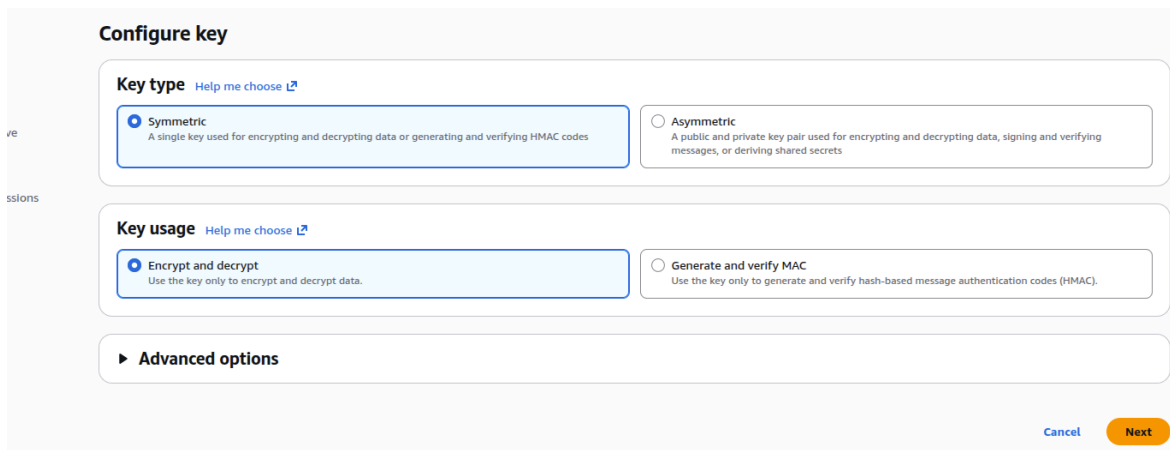


CREACIÓN DE KMS KEY

1. Ir la a consola
2. Buscamos KMS en el explorador de la consola:

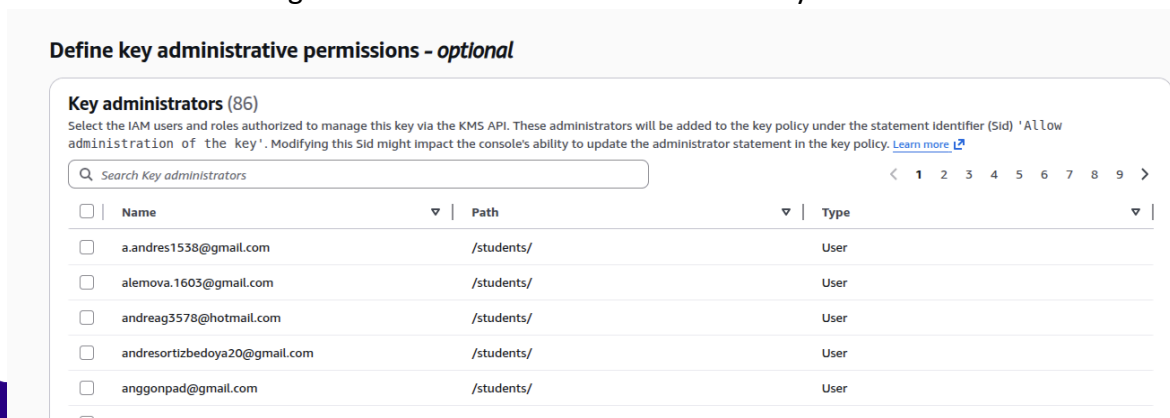


3. Creamos nuestra primera key



The screenshot shows the 'Configure key' wizard in the AWS Management Console. The 'Key type' section has two options: 'Symmetric' (selected) and 'Asymmetric'. The 'Key usage' section has two options: 'Encrypt and decrypt' (selected) and 'Generate and verify MAC'. The 'Advanced options' section is collapsed. At the bottom right, there are 'Cancel' and 'Next' buttons.

4. Nos asignamos como administradores de la key



The screenshot shows the 'Define key administrative permissions - optional' step in the AWS Management Console. The section is titled 'Key administrators (86)'. Below the title, there is a search bar and a table listing the administrators. The table has columns for 'Name', 'Path', and 'Type'. The administrators listed are:

Name	Path	Type
a.andres1538@gmail.com	/students/	User
alemova.1603@gmail.com	/students/	User
andrea3578@hotmail.com	/students/	User
andresortizbedoya20@gmail.com	/students/	User
anggonpad@gmail.com	/students/	User
andresortizbedoya20@gmail.com	/students/	User

5. Definimos quién puede usar esa key:

Define key usage permissions - optional

Key users (86)
Select the IAM users and roles authorized to use this key in cryptographic operations. These users will be added to the key policy under the statement identifiers (Sid) 'Allow use of the key' and 'Allow attachment of persistent resources'. Modifying these Sids might impact the console's ability to update the user statements in the key policy. [Learn more](#)

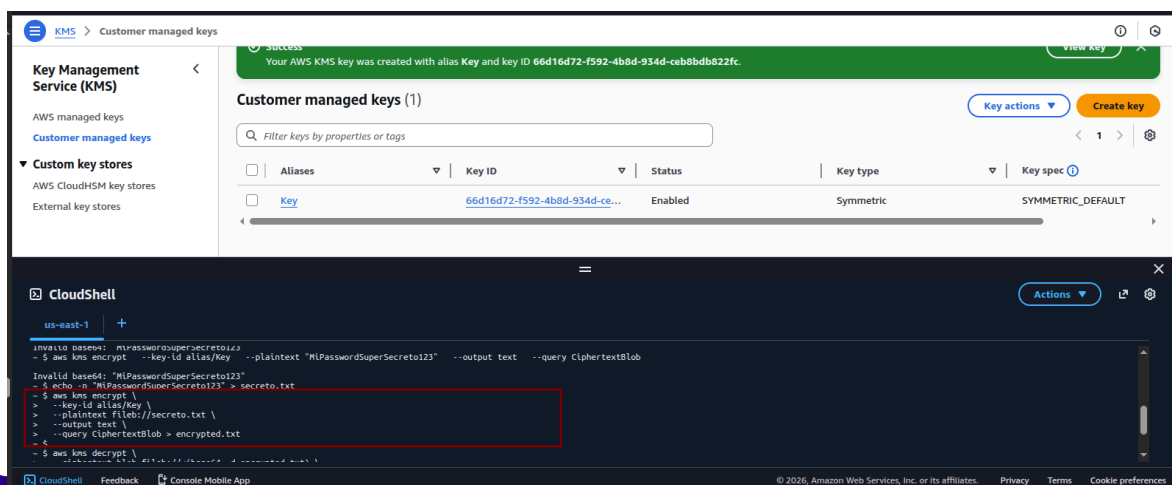
Search Key users

<input type="checkbox"/>	Name	Path	Type
<input type="checkbox"/>	GithubActionsAwsOrganizationsCICD	/	Role
<input type="checkbox"/>	gluetos3	/	Role
<input type="checkbox"/>	lambda-s3-read-only	/	Role
<input type="checkbox"/>	lambda_function-role-pl9r143a	/service-role/	Role
<input type="checkbox"/>	lambda_test-role-zuscaam	/service-role/	Role
<input type="checkbox"/>	OrganizationStack-AW5679f53fac002430cb0da5b7...	/	Role

Desde la consola desde la cloudshell, podemos simular el cifrado.

- Con este comando, solo crea un archivo con un texto:
echo -n "PasswordProduccion2026!" > secret.txt
- Para cifrar corremos este comando, ajustando el valor al de nuestra llave:

```
aws kms encrypt \
  --key-id alias/app-prod-key \
  --plaintext fileb://secret.txt \
  --query CiphertextBlob \
  --output text > secret.encrypted
```



The screenshot shows the AWS KMS console with a success message: "Your AWS KMS key was created with alias Key and key ID 66d16d72-f592-4b8d-934d-ceb8db822fc." Below this, the "Customer managed keys (1)" section shows a table with one key: "66d16d72-f592-4b8d-934d-ceb8db822fc" with status "Enabled", key type "Symmetric", and key spec "SYMMETRIC_DEFAULT".

Below the console, a CloudShell terminal window shows the following commands and output:

```
us-east-1
Invalid base64: "MlPasswordSuperSecret0123"
$ aws kms encrypt --key-id alias/app-prod-key --plaintext fileb://secret.txt --query CiphertextBlob --output text
Invalid base64: "MlPasswordSuperSecret0123"
$ echo -n "MlPasswordSuperSecret0123" > secret.txt
$ aws kms encrypt \
  --key-id alias/app-prod-key \
  --plaintext fileb://secret.txt \
  --output text \
  --query CiphertextBlob > secret.encrypted
$ aws kms decrypt \
  --key-id alias/app-prod-key --ciphertextBlob fileb://secret.encrypted
```


Ahora, tenemos un secreto cifrado en base64.

¿Qué ocurrió?

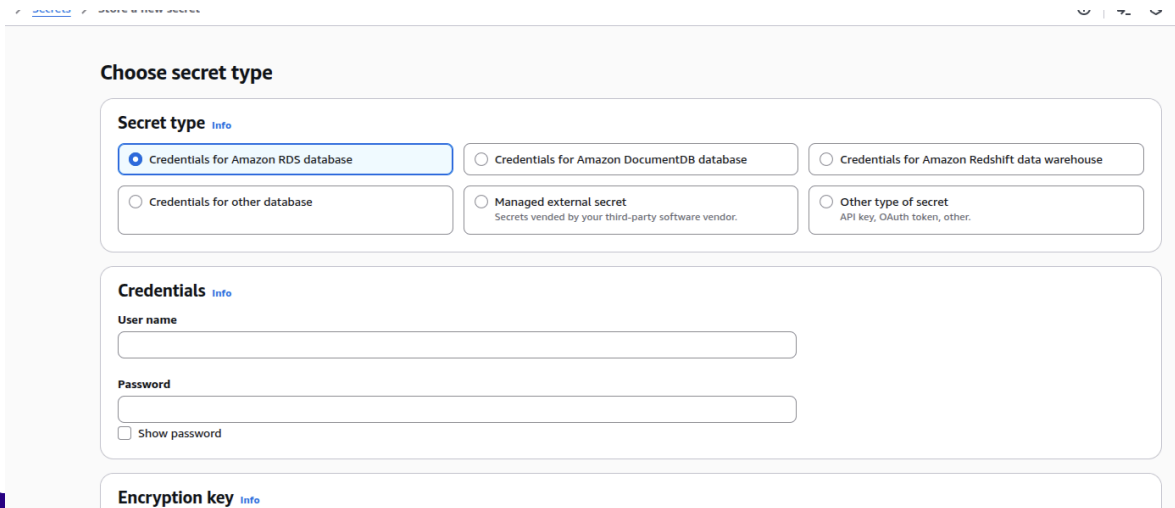
- ☐ CLI envió binario
 - ☐ KMS generó una data key
 - ☐ Cifró el texto
 - ☐ Devolvió el blob cifrado
- Para desencriptar

```
aws kms decrypt \  
  --ciphertext-blob fileb://<(base64 -d secret.encrypted) \  
  --query Plaintext \  
  --output text | base64 --decode
```

Solo quien tenga permiso **kms:Decrypt** puede hacerlo.

USUARIOS SECRETS MANAGER

1. En la consola, vamos al servicio Secrets Manager y luego a Store new secret



The screenshot shows the 'Store new secret' form in the AWS Secrets Manager console. The form is titled 'Choose secret type' and contains three main sections: 'Secret type', 'Credentials', and 'Encryption key'. In the 'Secret type' section, 'Credentials for Amazon RDS database' is selected. The 'Credentials' section has fields for 'User name' and 'Password', with a 'Show password' checkbox. The 'Encryption key' section is partially visible at the bottom.

2. Ponemos en texto plano:

```
{
  "username": "app_user",
  "password": "PasswordFuerte2026!"
}
```

3. Necesitamos una política de IAM que permita usar el secreto

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:654654478122:secret:prod/app/db-credentials-*"
    }
  ]
}
```

4. Para ejemplo práctico crearemos un rol, el cual podrá usar dicho secreto desde una instancia:

- Creamos una instancia
- Asociamos el rol que permite el secreto
- Nos conectamos via ssh a la ec2
- Ejecutamos:

```
aws secretsmanager get-secret-value \
  --secret-id prod/app/db-credentials \
  --query SecretString \
  --output text
```

- Nos permitirá y devolverá el secreto en formato json

RECUERDA: Eliminar todos los recursos desplegados en esta práctica