**I.- OBJETIVOS:**

* Crear una instancia en la plataforma de AWS LightSail

**II.- SEGURIDAD:**

|  |  |
| --- | --- |
|  | **Advertencia:**  **En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.** |

**III.- FUNDAMENTO TEÓRICO:**

* Revisar el texto guía que está en el campus Virtual.

**IV.- NORMAS EMPLEADAS:**

* No aplica

**V.- RECURSOS:**

* En este laboratorio cada alumno trabajará con un equipo con conexión a Internet y una cuenta de AWS

**VI.- METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:**

* El desarrollo del laboratorio es individual.

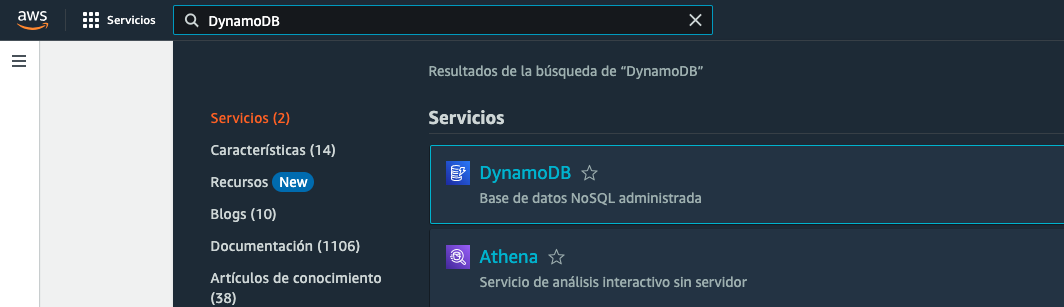
**VII.- PROCEDIMIENTO:**

**Paso 1: Creación de servicio DynamoDB**

1. Acceda a la consola de Amazon EC2 con el usuario raiz

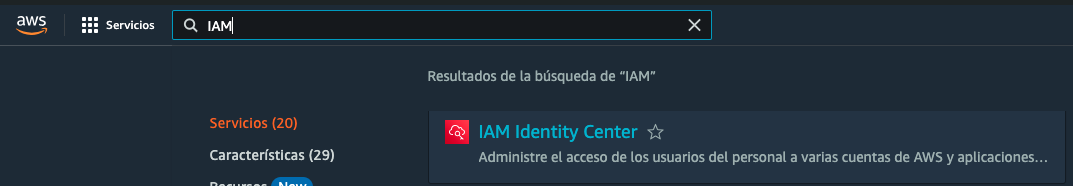
|  |
| --- |
| <https://console.aws.amazon.com/ec2/> |

1. En el panel de búsqueda de servicios, busque el servicio **DynamoDB**

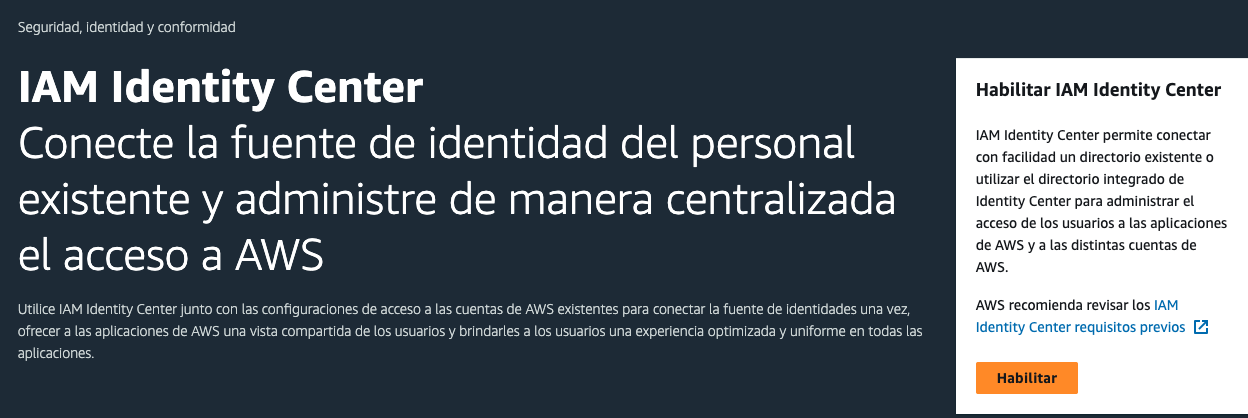


**Paso 2: Habilitación de servicio IAM Identity Center y creación de usuario, grupo y permisos**

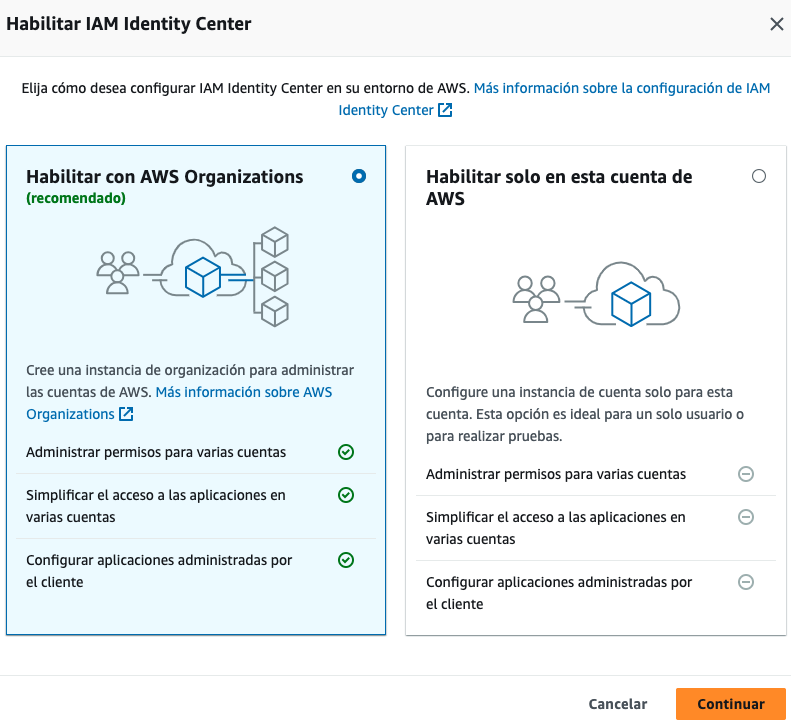
1. En una nueva pestana acceda al portal de su consola AWS y en el panel de búsqueda de servicios, busque el servicio **IAM Identity Center**



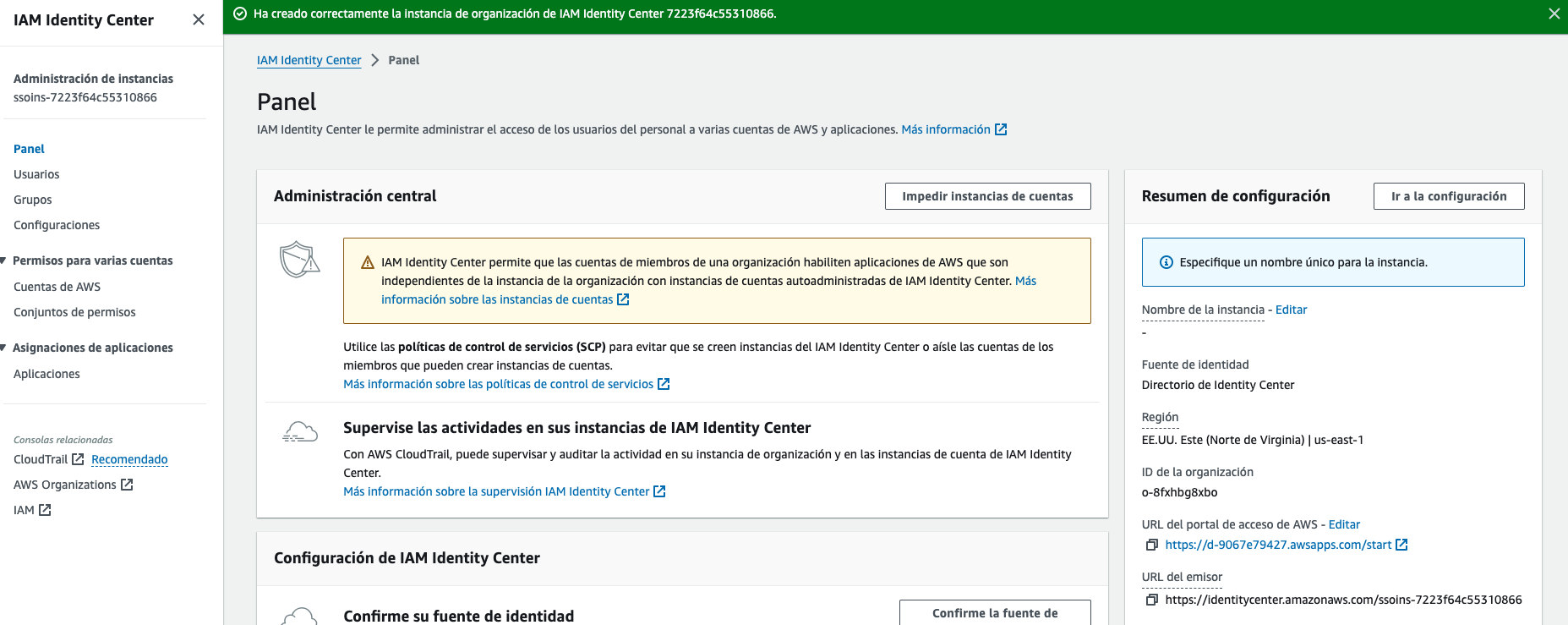
1. Haga clic en el botón de **Habilitar** de **IAM Identity Center**



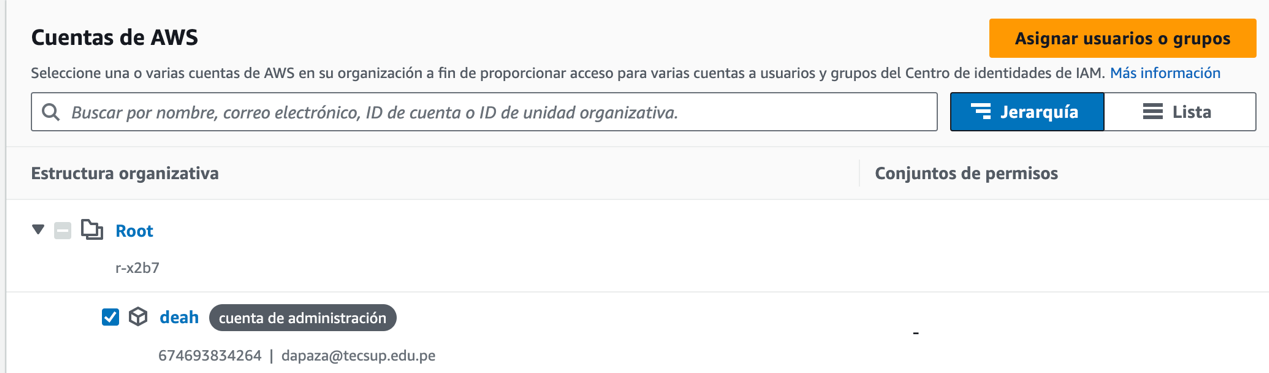
1. Seleccione **Habilitar con AWS Organizations** y haga clic en **Continuar**



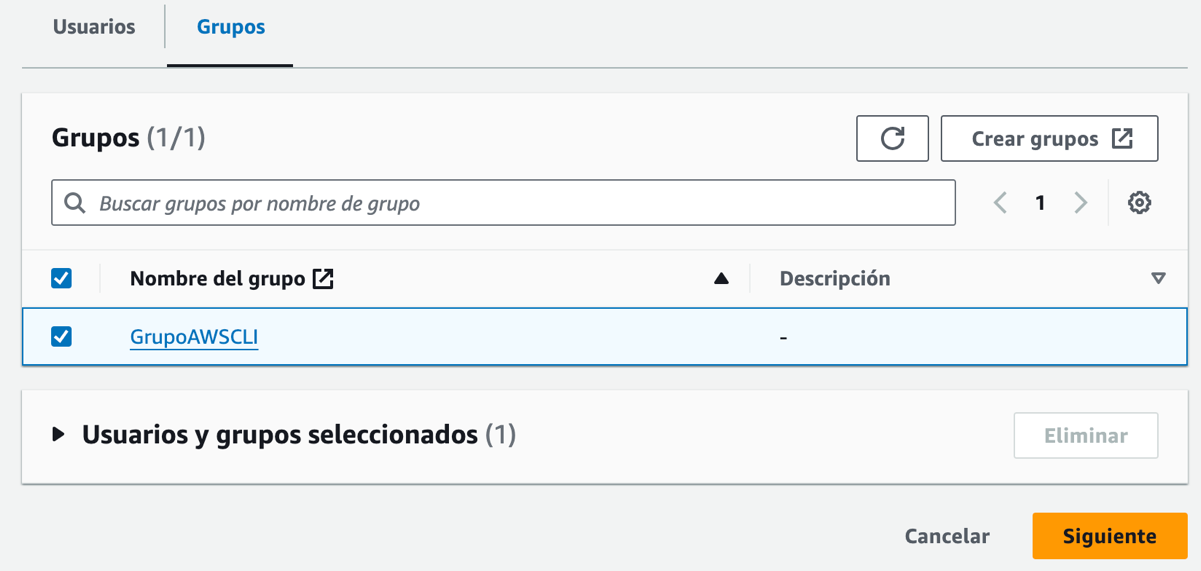
1. Una vez que habilite los servicios verificara que en el lado izquierdo se mostrara el resultado de las configuraciones habilitadas



1. En la consola **de IAM Identity Center** en el lado derecho haga clic en **Usuarios** y luego clic en el botón **Agregar usuario**. Configure como se especifica:
   1. Nombre de usuario: **UsuarioAWSCLI**
   2. Contraseña: Envíe un correo electrónico a este usuario con las instrucciones de configuración de la contraseña.
   3. Dirección de correo electrónico: Coloque algún correo personal que no corresponda al correo del usuario raíz
   4. Llene los demás datos según su criterio y luego haga clic en el botón de **Siguiente**
   5. En la ventana mostrada ahí clic en el botón de **Crear grupo**. Notara que se apertura una nueva pestaña en el navegador.
      1. Nombre de grupo: **GrupoAWSCLI**
      2. Deje las demás opciones por defecto y luego haga clic en el botón **Crear grupo**
   6. En la ventana de creación de usuario haga clic en el botón de **Actualizar** para mostrar el nuevo grupo creado
   7. 
   8. Seleccione el grupo creado **GrupoAWSCLI** y haga clic en el botón de **Siguiente**
   9. Se mostrar una pantalla con detalles del nuevo usuario a crear haga clic en el botón de **Agregar usuario**
   10. 
2. Se procederá a asignar permisos al usuario **UsuarioAWSCLI** para que se le permita tener determinados privilegios de administración. En su plataforma **de IAM Identity Center** en el lado izquierdo haga clic en **Panel**. Configure como se indica:
   1. En el lado izquierdo en la sección **Permisos para varias cuentas** haga clic en **Cuentas AWS.** Se mostrara en la estructura organizativa su usuario



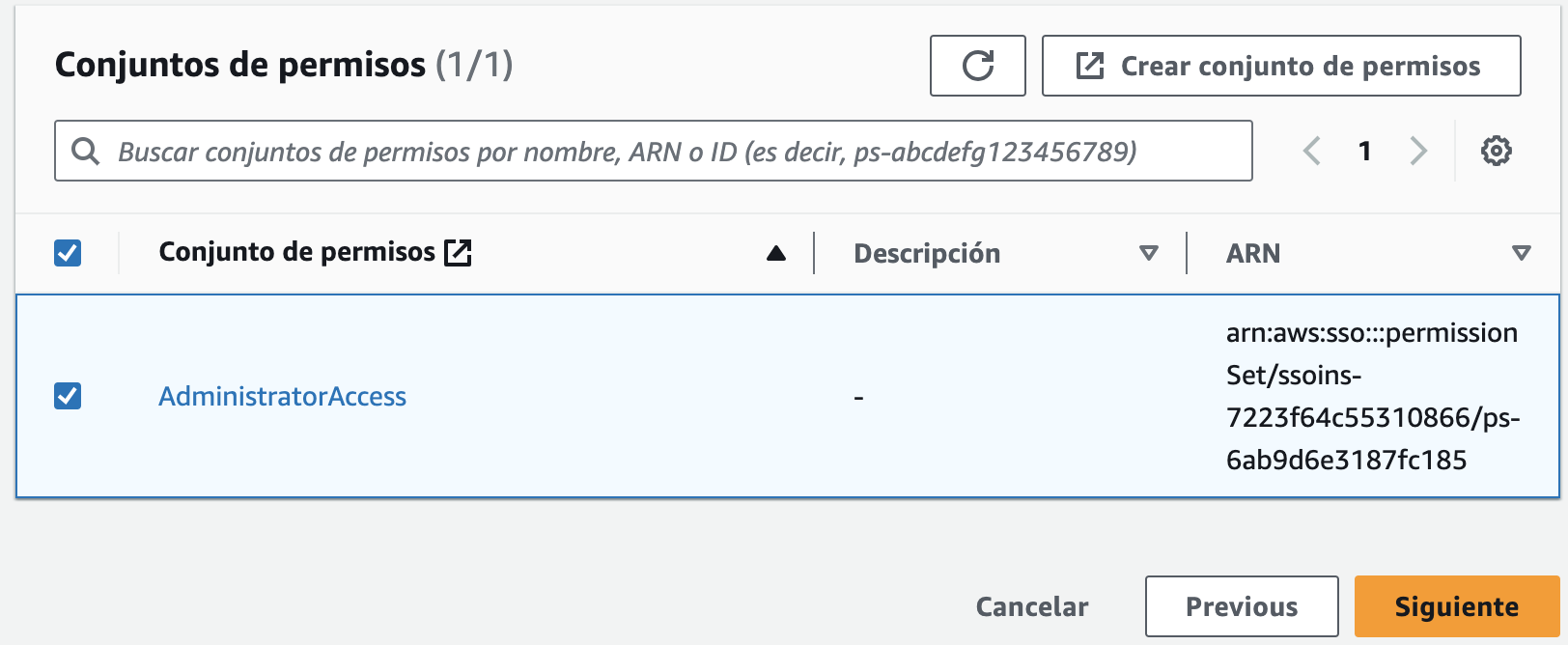
* 1. Seleccione su usuario y haga clic en **Asignar usuarios o grupos** y configure como se indica:
     1. Seleccione su usuario **UsuarioAWSCLI** y haga clic en Siguiente



* + 1. Haga clic en el botón de **Crear conjunto de permisos**, este enlace se habilitara en una nueva pestaña
       1. En este primer paso de asistente , en **tipos** seleccione **Conjunto de permisos predefinido**. En **Seleccione una política administrada por AWS** elija **AdministratorAccess**. Haga clic en **Siguiente**.
       2. En el segundo paso del asistente deje todo por defecto y haga clic **en Siguiente**
       3. En el tercer paso del asistente deje todo por defecto y haga clic en **Crear**



* + 1. En la ventana previa de **Asignar usuarios y grupos** haga clic en **actualizar** y seleccione el nuevo permiso creado(AdministratorAccess) y haga clic en **Siguiente**

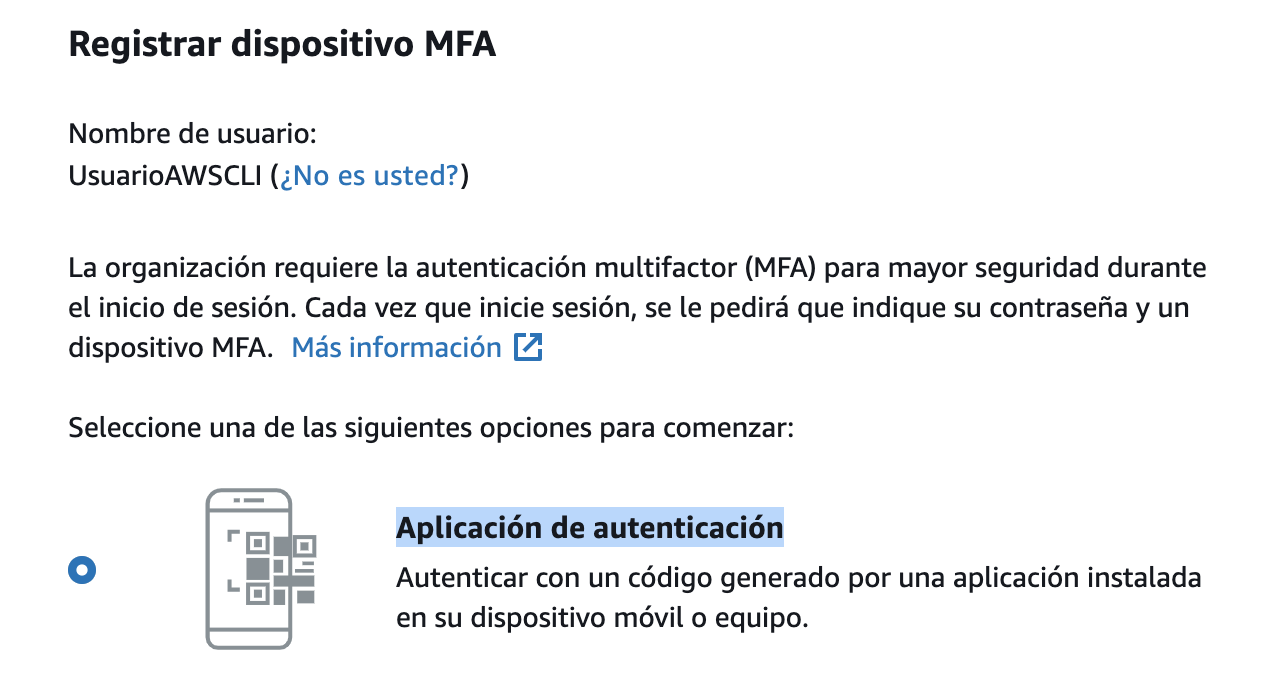


* + 1. Deje todo por defecto y haga clic en boto de **Enviar**

1. Se procederá a validar el acceso creado. Configure como se especifica:
   1. En la bandeja de su correo, verifique que recibió un mensaje. Y haga clic en el botón de **Accept Invitation**



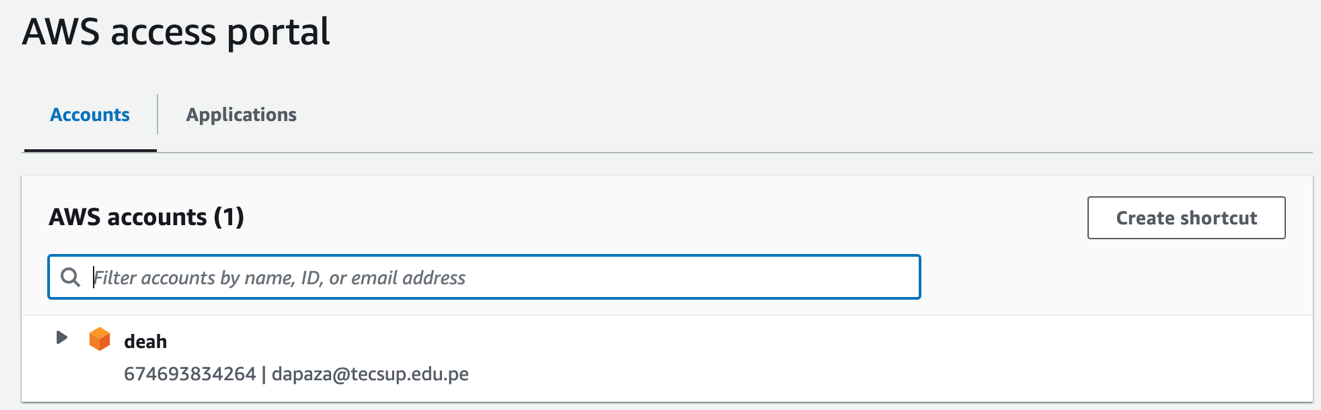
* 1. En la ventana mostrada asigne una contraseña para su usuario, como por ejemplo **1234@User** . Luego haga clic en establecer contraseña.
  2. Le pedirá que se autentifique que con el usuario y contraseña generada
  3. Se solicitará asignar un método de seguridad, seleccione la que dice **Aplicación de autenticación** y haga clic en el botón de **Siguiente**



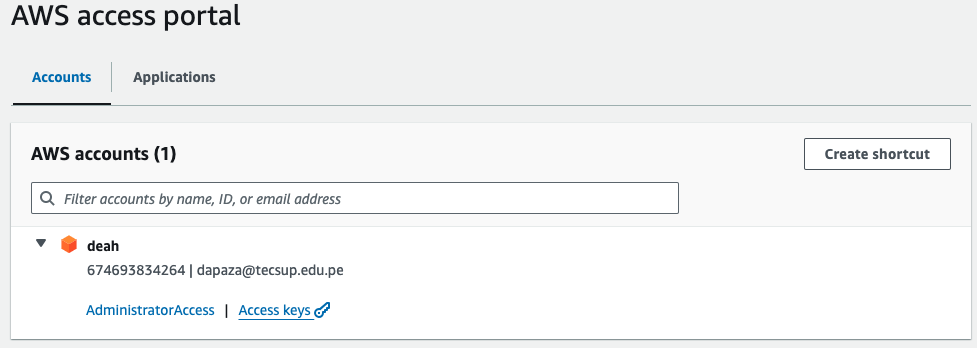
* 1. Siga los pasos del asistente para configurar su cuenta asociado a un dispositivo móvil



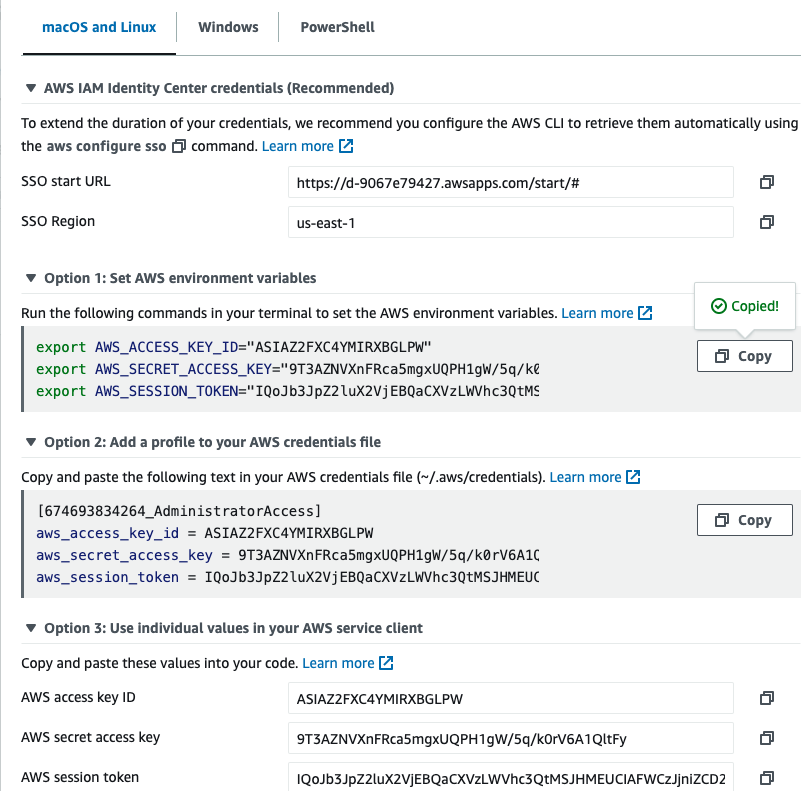
1. Debería en este punto tener acceso a **AWS Access portal**



1. Expanda el nodo del usuario y haga clic en el enlace de **Access keys**

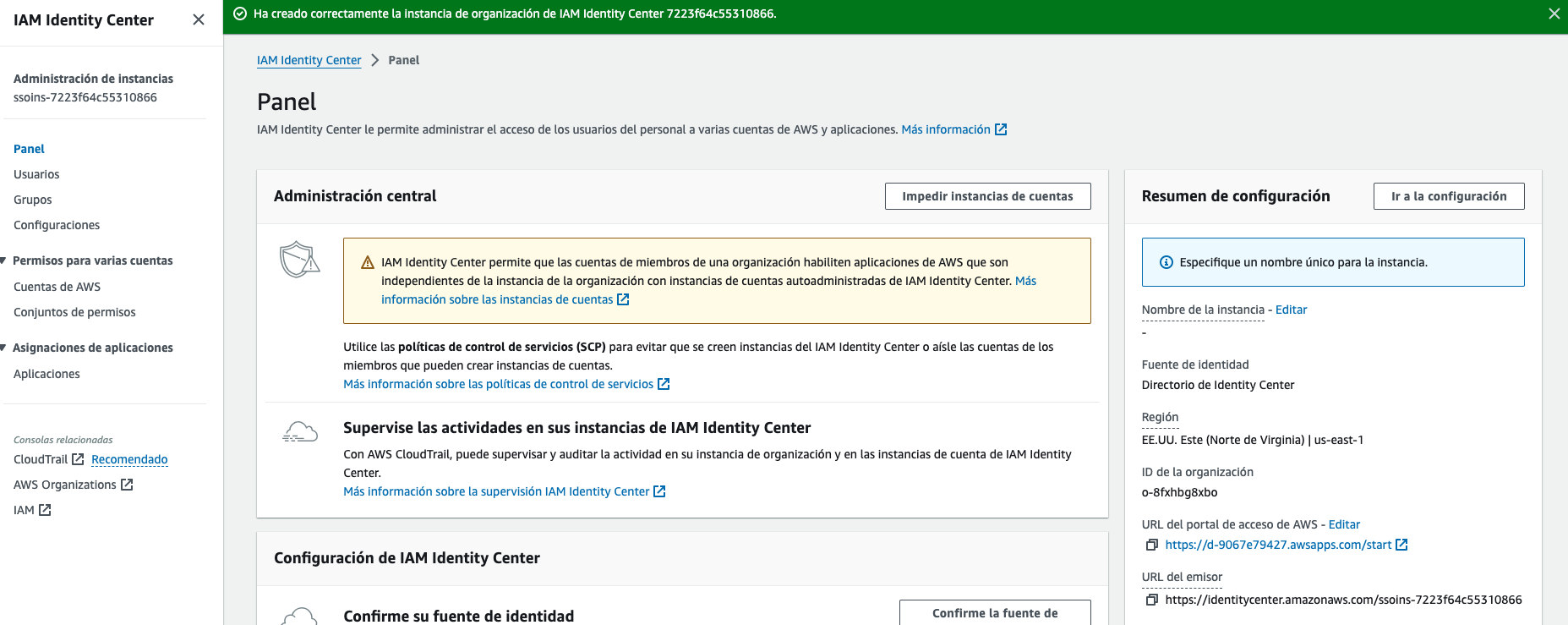


1. Se mostrara las claves de acceso generadas para el usuario. Utilizara estas credenciales pasos mas adelante(OJO: Estas credenciales tiene un tiempo dee vigencia de una hora por defecto, si desea puedee ampliar este tiempo al momento de configurar el permiso para el grupo creado)



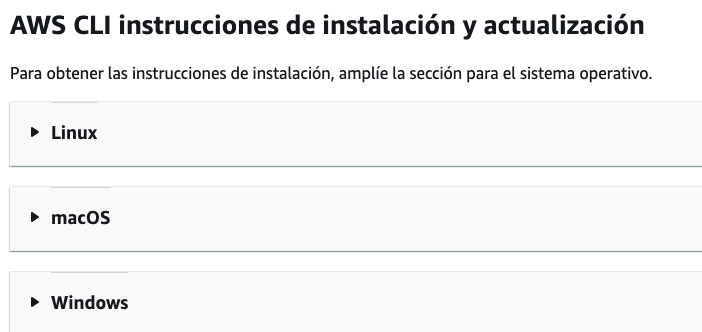
**Paso 3: Instalación de AWS CLI**

1. Cree un proyecto local en su computador denominado **UsoDynamoDB**
2. Vuelva al panel de administración de **IAM Identity Center** para verificar sus credenciales de acceso



1. Siga los pasos del siguiente asistente enlace que corresponde a la documentación oficial de AWS para instalar los servicios de AWS CLI. Realice los pasos en base al sistema operativo que utiliza

|  |
| --- |
| <https://docs.aws.amazon.com/es_es/cli/latest/userguide/getting-started-install.html> |



1. Una vez realizado los pasos según su sistema operativo, compruebe si se encuentra habilitado los servicios de AWS CLI

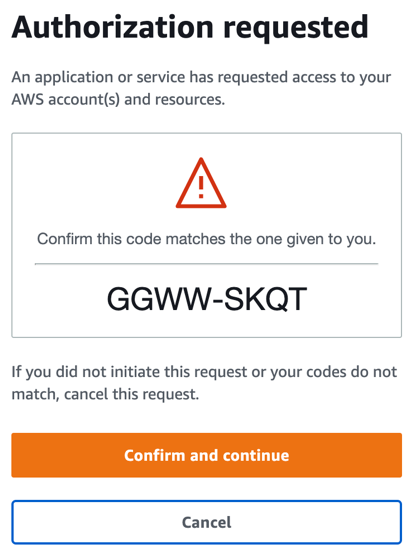
|  |
| --- |
| aws --version |



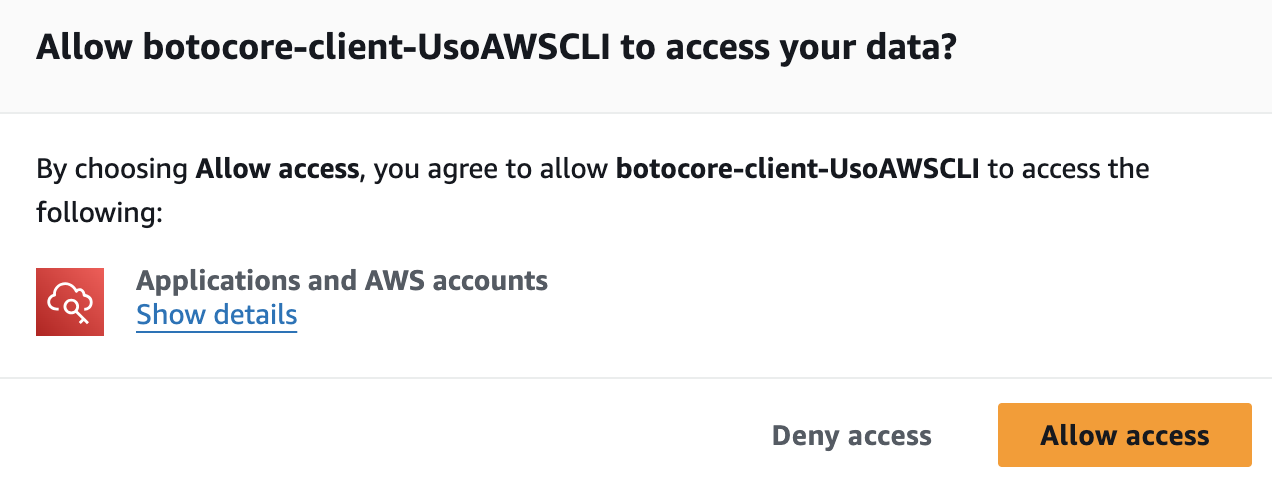
1. Ejecutaremos el asistente de AWS CLI para generar credenciales de autentificación. Configure como se indica a continuación:
   1. Ejecute el comando de asistente de configuración

|  |
| --- |
| aws configure sso |

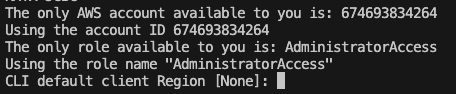
* 1. SSO Sesion Name: **UsoAWSCLI**
  2. SSO Start URL: Coloque del punto 12 el campo SSO Start URL
  3. SSO Region: Coloque del punto 12 el campo SSO Region
  4. SSO Registration scopes: **sso:account:access**
  5. Se mostrara una Ventana de autorizacion, haga clic en Confirm and continue



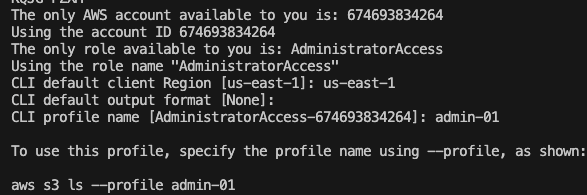
* 1. Acceda con su usuario y clave de Usuario creado en IAM Identity Center
  2. Para conceder los permisos de acceso haga clic en el boton de Allow Access



* 1. Una vez realizado el proceso puede cerrar la ventana
  2. En su terminal verifique que se ha validado su acceso



* 1. En CLI default client Region: coloque la region en la que esta trabajando su plataforma de AWS.
  2. En CLI default output format: deje por defecto y presione enter
  3. En CLI profile name: Coloque admin-01



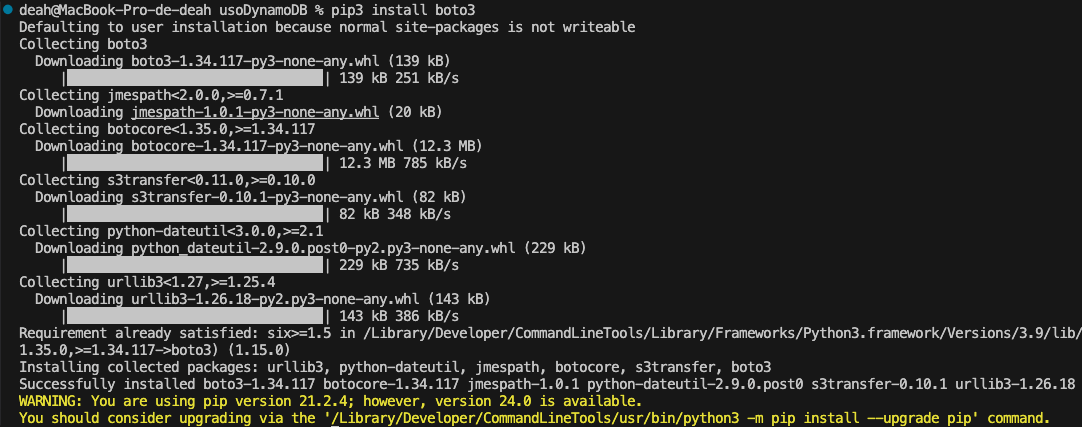
1. Ahora se definira los accesos generados para el usuario para poder enlazar a los servicios de AWS. Se va a definir variables de entorno para que el sistema operativo pueda utilizar las credenciales generadas. En las credenciales mostradas en el punto 12 elija el sistema operativo con el cual este trabajando y configure las variables de entorno desde su terminal(OJO: Si en algun momento se pierde al tiempo de acceso a la consola,, puede actualizar su pagina y estas credenciales cambiaran de valor para permitirle dar mas tiempo de acceso en la plataforma)



**Paso 4: Creación de tabla**

1. Acceda a la carpeta de su proyecto local creado denominado **UsoDynamoDB**
2. Instale la SDK de AWS para acceder a los servicios de **DynamoDB**. En una terminal ejecute el siguiente comando

|  |
| --- |
| pip3 install boto3 |



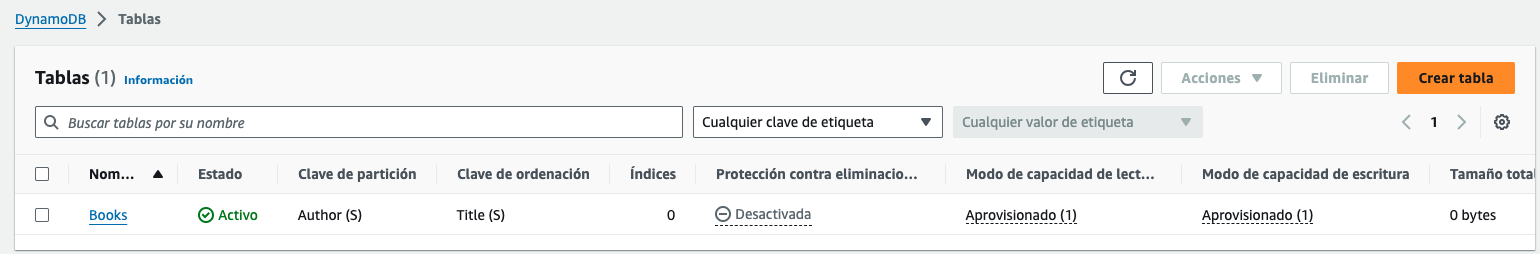
1. Cree un archivo denominado **crearTabla.py** con el siguiente contenido

|  |
| --- |
| import boto3  # boto3 is the AWS SDK library for Python.  # We can use the low-level client to make API calls to DynamoDB.  client = boto3.client('dynamodb', region\_name='us-east-1')  try:  resp = client.create\_table(TableName="Books",  # Declare your Primary Key in the KeySchema argument  KeySchema=[  {  "AttributeName": "Author",  "KeyType": "HASH"  },  {  "AttributeName": "Title",  "KeyType": "RANGE"  }  ],  # Any attributes used in KeySchema or Indexes must be declared in AttributeDefinitions  AttributeDefinitions=[  {  "AttributeName": "Author",  "AttributeType": "S"  },  {  "AttributeName": "Title",  "AttributeType": "S"  }  ],  # ProvisionedThroughput controls the amount of data you can read or write to DynamoDB per second.  # You can control read and write capacity independently.  ProvisionedThroughput={  "ReadCapacityUnits": 1,  "WriteCapacityUnits": 1  }  )  print("Table created successfully!")  except Exception as e:  print("Error creating table:")  print(e) |

1. Ejecute el archivo **crearTabla.py** . Verifique que no se muestra ningún error.



1. En el portal de **DynamoDB** en el lado izquierdo haga clic en **Tablas** y verifique que se muestra la base de datos **Books** creada



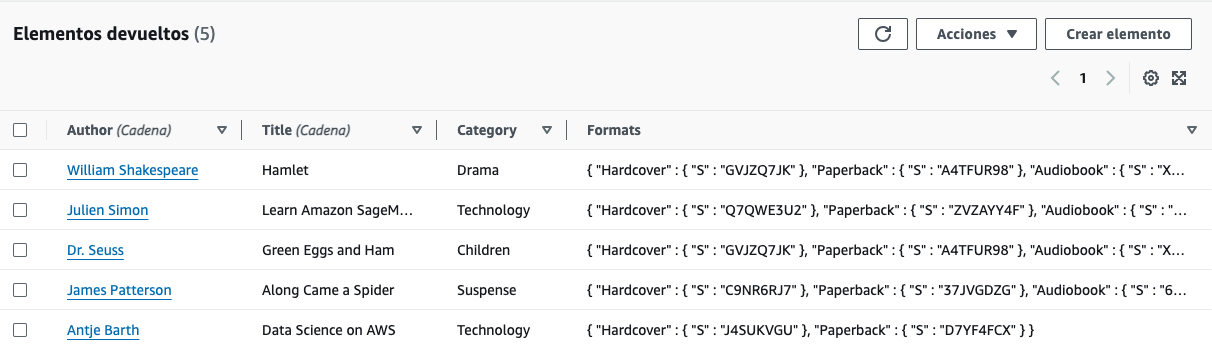
1. Cree un archivo denominado **insertarDatos.json** con el siguiente contenido

|  |
| --- |
| [  {"Author": "Antje Barth", "Title": "Data Science on AWS","Category": "Technology", "Formats": { "Hardcover": "J4SUKVGU", "Paperback": "D7YF4FCX" }},  {"Author": "Julien Simon", "Title": "Learn Amazon SageMaker","Category": "Technology", "Formats": { "Hardcover": "Q7QWE3U2","Paperback": "ZVZAYY4F", "Audiobook": "DJ9KS9NM" }},  {"Author": "James Patterson", "Title": "Along Came a Spider","Category": "Suspense", "Formats": { "Hardcover": "C9NR6RJ7","Paperback": "37JVGDZG", "Audiobook": "6348WX3U" }},  {"Author": "Dr. Seuss", "Title": "Green Eggs and Ham","Category": "Children", "Formats": { "Hardcover": "GVJZQ7JK","Paperback": "A4TFUR98", "Audiobook": "XWMGHW96" }},  {"Author": "William Shakespeare", "Title": "Hamlet", "Category": "Drama", "Formats": { "Hardcover": "GVJZQ7JK","Paperback": "A4TFUR98", "Audiobook": "XWMGHW96" }}  ] |

1. Cree un archivo denominado **insertarDatos.py** con el siguiente contenido. Ejecute el archivo.

|  |
| --- |
| import boto3  import json  # Configurar el cliente de DynamoDB  dynamodb = boto3.resource('dynamodb')  # Especificar el nombre de tu tabla  table\_name = 'Books'  table = dynamodb.Table(table\_name)  # Leer los datos del archivo JSON  with open('insertarDatos.json') as f:  data = json.load(f)  # Insertar cada item en la tabla de DynamoDB  for item in data:  table.put\_item(Item=item)  print("Datos insertados exitosamente.") |

1. Acceda a la plataforma **DynamoDB** y en el lado izquierdo elija **Explorar elementos** y verifique que se muestran los datos insertados



1. Cree un archivo denominado **consultaDatos.py** con el siguiente contenido. Este permitirá filtrar datos en base al autor y titulo.

|  |
| --- |
| import boto3  dynamodb = boto3.client('dynamodb', region\_name='us-east-1')  resp = dynamodb.execute\_statement(Statement='SELECT \* FROM Books WHERE Author = \'Antje Barth\' AND Title = \'Data Science on AWS\'')  print(resp['Items']) |

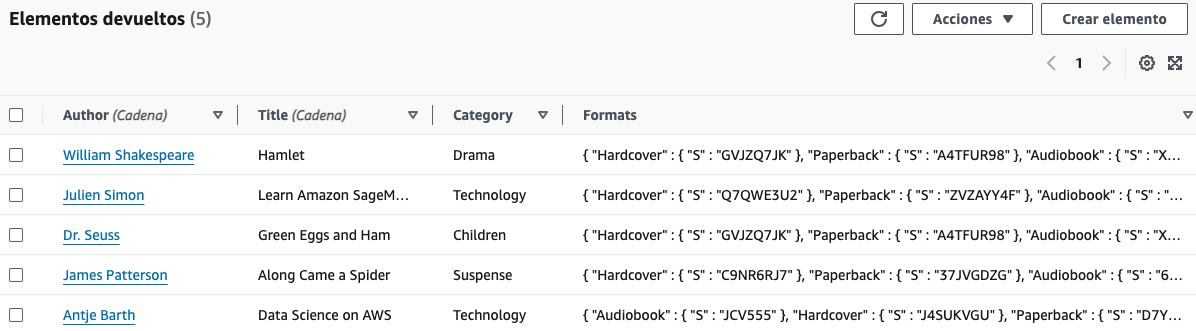
1. Ejecute el archivo consultaDatos.py y verifique que se lista la información en consola



1. Cree un archivo denominado **actualizarDatos.py** con el siguiente contenido.

|  |
| --- |
| import boto3  dynamodb = boto3.client('dynamodb', region\_name='us-east-1')  resp = dynamodb.execute\_statement(Statement='UPDATE Books SET Formats.Audiobook = \'JCV555\' WHERE Author = \'Antje Barth\' AND Title = \'Data Science on AWS\'')  print(resp['Items']) |

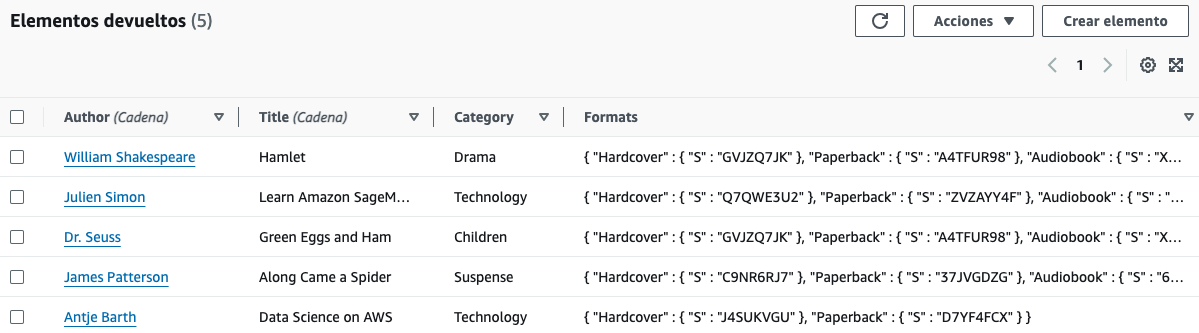
1. Ejecute el archivo **actualizarDatos.py** y verifique en la plataforma de **DynamoDB** que se agregó un nuevo valor al campo **Formats** perteneciente al usuario **Antje Barth**



1. Cree un archivo denominado **eliminarDatos.py** con el siguiente contenido

|  |
| --- |
| import boto3  dynamodb = boto3.client('dynamodb', region\_name='us-east-1')  resp = dynamodb.execute\_statement(Statement='UPDATE Books REMOVE Formats.Audiobook WHERE Author = \'Antje Barth\' AND Title = \'Data Science on AWS\'')  print(resp['Items']) |

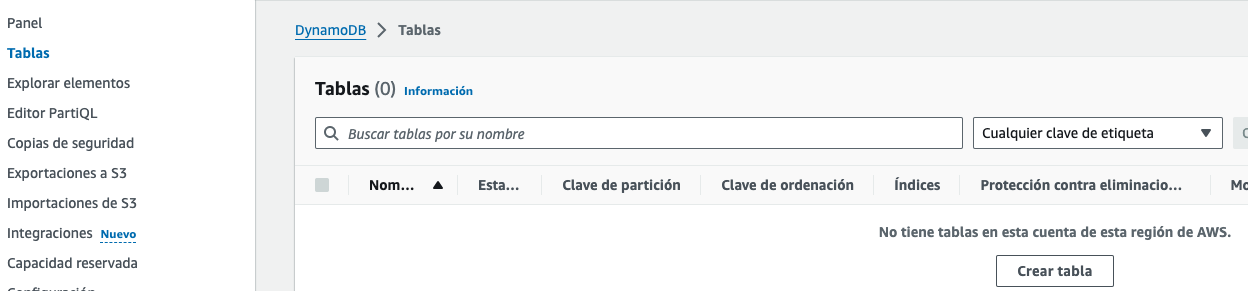
1. Ejecute el archivo **eliminarDatos.py** y verifique en la plataforma de **DynamoDB** que se elimino el elemento **AudioBook** del atributo **Formats**, perteneciente al usuario **Antje Barth**



1. Cree un archivo denominado **eliminarTabla.py** con el siguiente contenido

|  |
| --- |
| import boto3  client = boto3.client('dynamodb', region\_name='us-east-1')  try:  resp = client.delete\_table(  TableName="Books",  )  print("Table deleted successfully!")  except Exception as e:  print("Error deleting table:")  print(e) |

1. Ejecute el archivo **eliminarTabla.py** y verifique en la plataforma de **DynamoDB** que se elimino la tabla **Books**



**Caso propuesto**

* Implemente un sistema que simule una agenda de tareas que permita almacenar datos como: Nombre de tarea, encargado de tarea, fecha de inicio, fecha de finalización, estado de tarea. Debe crear la tabla en DynamoDB e implementar un CRUD completo.
* El sistema debe mostrar un entorno visual desde el cual el usuario podrá realizar el CRUD completo de los datos.
* Puede guiarse de la documentación oficial de aws que explica cómo utilizar los servicios de la SDK de AWS en distintos lenguajes de programación

|  |
| --- |
| <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.html> |

* Documente el proceso realizado, mostrando capturas de la funcionalidad del sistema implementado.

**CONCLUSIONES (5 mínimo):**

(Las conclusiones son una opinión personal sobre tu trabajo, explicar cómo resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión critica de lo realizado)