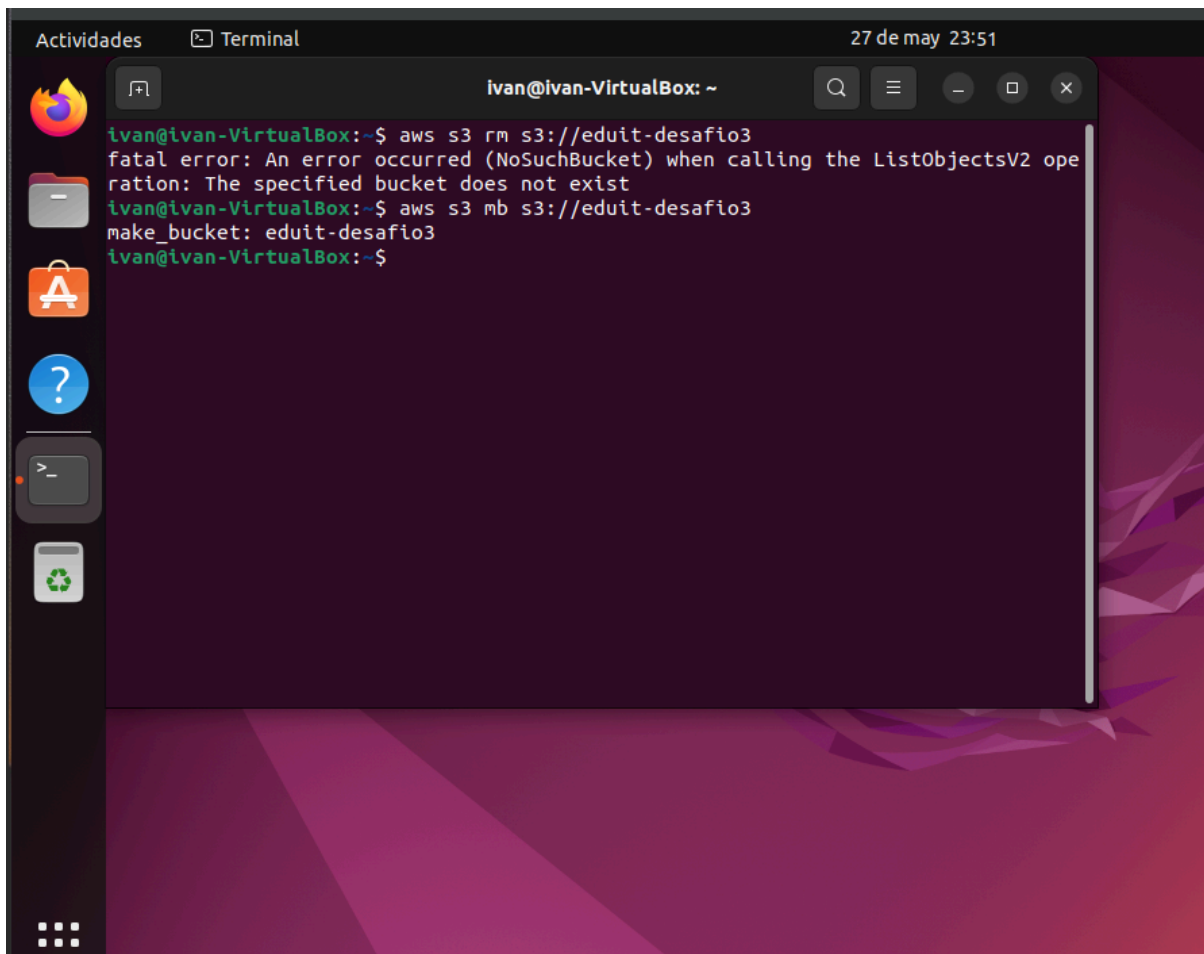


Desafío 3

Introducción

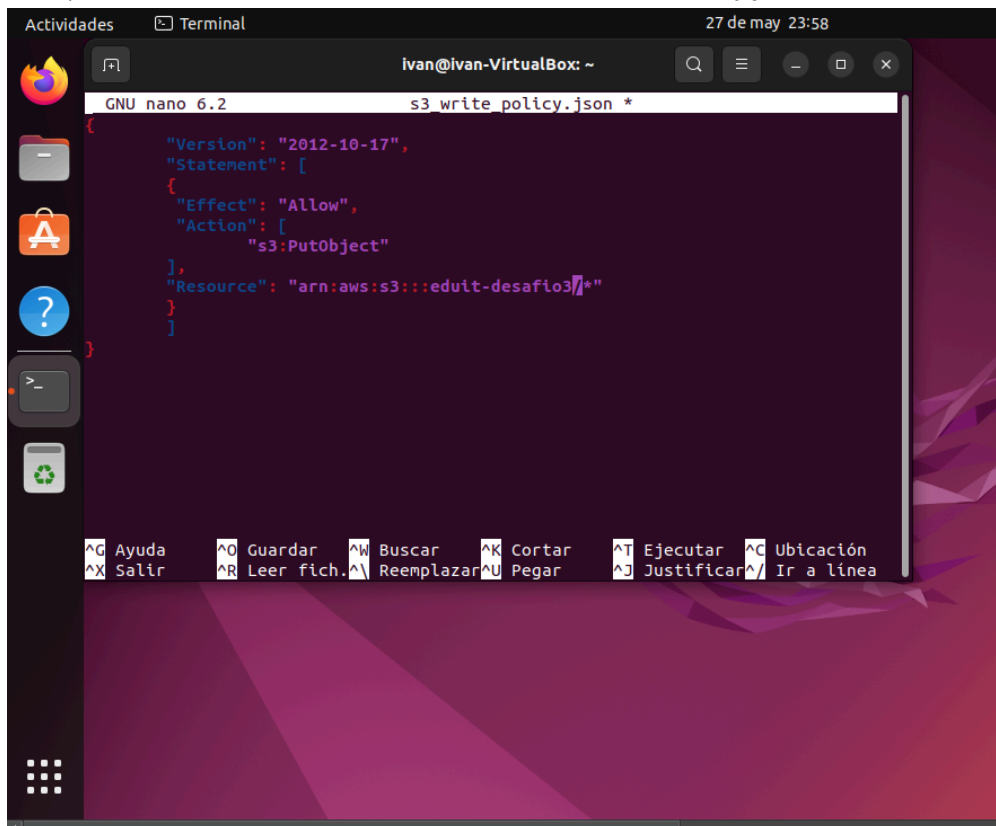
El desafío se realizó a través de AWS CLI corriendo en VBox con SO Ubuntu, previamente creamos una cuenta en AWS para utilizar la capa gratuita siguiendo las recomendaciones de agregar 2FA, Crear un usuario IAM para no utilizar el usuario raíz y agregar 2FA también para los usuarios IAM. Luego conectamos AWS CLI con las credenciales para el usuario IAM que creamos (admin).

- 1) creamos el bucket



```
ivan@ivan-VirtualBox: ~  
ivan@ivan-VirtualBox:~$ aws s3 rm s3://eduit-desafio3  
fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist  
ivan@ivan-VirtualBox:~$ aws s3 mb s3://eduit-desafio3  
make_bucket: eduit-desafio3  
ivan@ivan-VirtualBox:~$
```

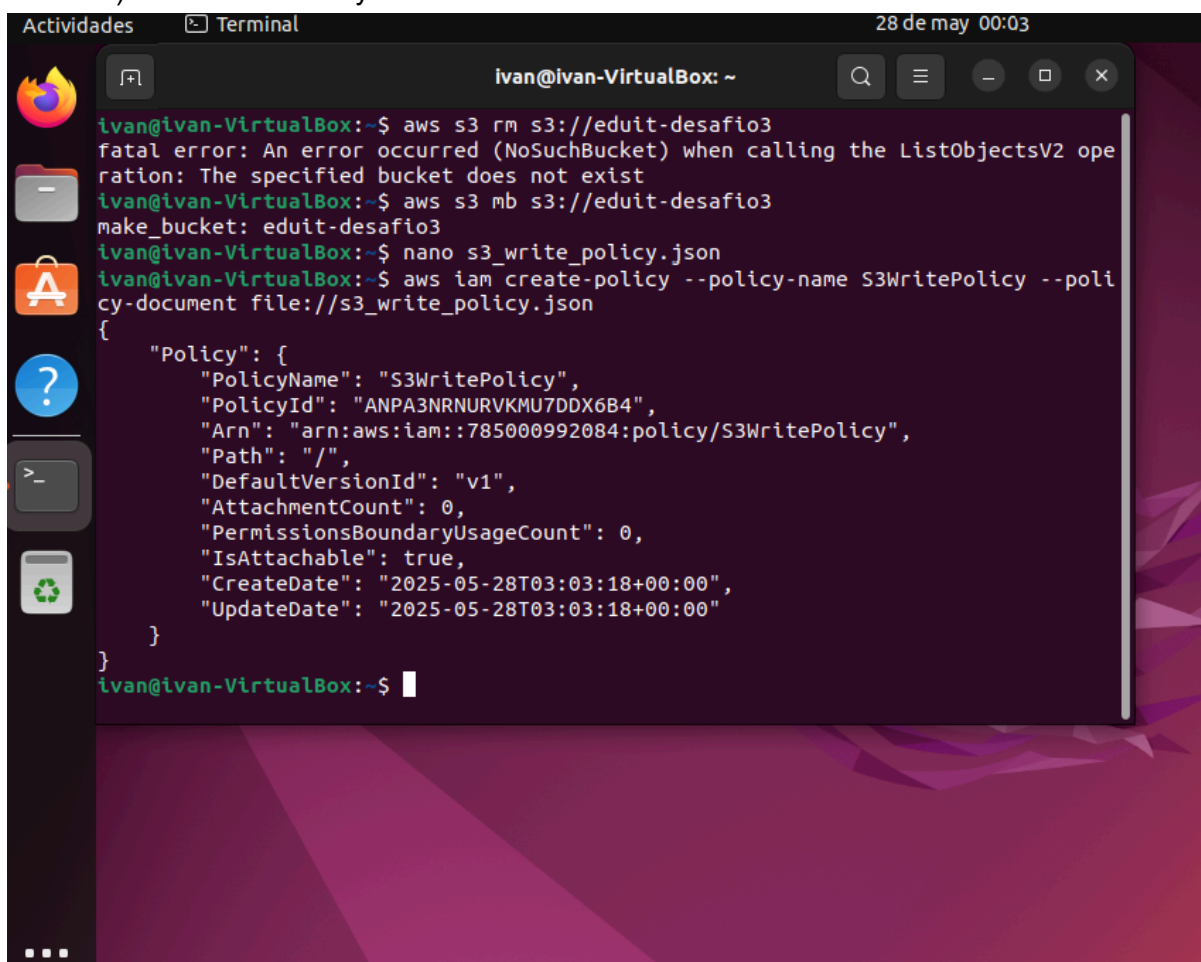
2) Creamos el archivo con la política: s3_write_policy.json



The screenshot shows a terminal window titled 'ivan@ivan-VirtualBox: ~' with a timestamp of '27 de may 23:58'. The user is editing a file named 's3_write_policy.json' using the 'GNU nano 6.2' editor. The file content is a JSON policy document. The terminal window has a sidebar with icons for 'Actividades', 'Terminal', and a file manager. At the bottom, there is a keyboard shortcuts menu.

```
ivan@ivan-VirtualBox: ~  
GNU nano 6.2 s3_write_policy.json *  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:PutObject"  
      ],  
      "Resource": "arn:aws:s3:::eduit-desafio3/*"  
    }  
  ]  
}
```

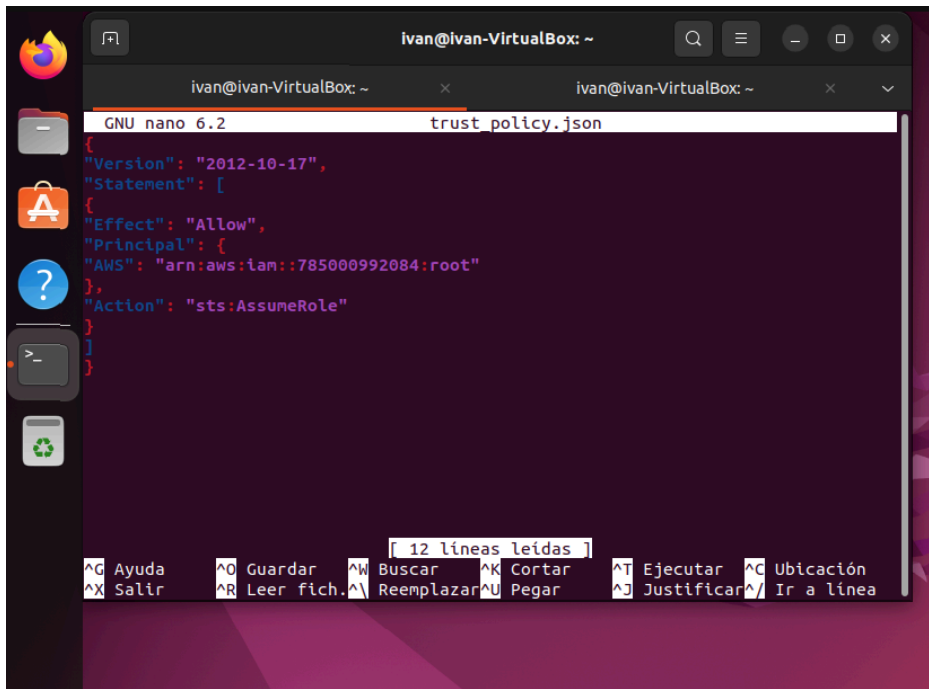
3) Creamos la Policy en aws.



The screenshot shows a terminal window titled 'ivan@ivan-VirtualBox: ~' with a timestamp of '28 de may 00:03'. The user is performing several AWS CLI commands. First, they attempt to remove a bucket 's3://eduit-desafio3' but receive a 'fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist'. Then, they create the bucket 's3://eduit-desafio3'. Next, they edit the 's3_write_policy.json' file using 'nano'. Finally, they create the policy 'S3WritePolicy' using 'aws iam create-policy --policy-name S3WritePolicy --policy-document file://s3_write_policy.json'. The output shows the details of the created policy.

```
ivan@ivan-VirtualBox:~$ aws s3 rm s3://eduit-desafio3  
fatal error: An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist  
ivan@ivan-VirtualBox:~$ aws s3 mb s3://eduit-desafio3  
make_bucket: eduit-desafio3  
ivan@ivan-VirtualBox:~$ nano s3_write_policy.json  
ivan@ivan-VirtualBox:~$ aws iam create-policy --policy-name S3WritePolicy --policy-document file://s3_write_policy.json  
{  
  "Policy": {  
    "PolicyName": "S3WritePolicy",  
    "PolicyId": "ANPA3NRNURVKMU7DDX6B4",  
    "Arn": "arn:aws:iam::785000992084:policy/S3WritePolicy",  
    "Path": "/",  
    "DefaultVersionId": "v1",  
    "AttachmentCount": 0,  
    "PermissionsBoundaryUsageCount": 0,  
    "IsAttachable": true,  
    "CreateDate": "2025-05-28T03:03:18+00:00",  
    "UpdateDate": "2025-05-28T03:03:18+00:00"  
  }  
}
```

4) Creamos el archivo con el rol: trust-policy.json

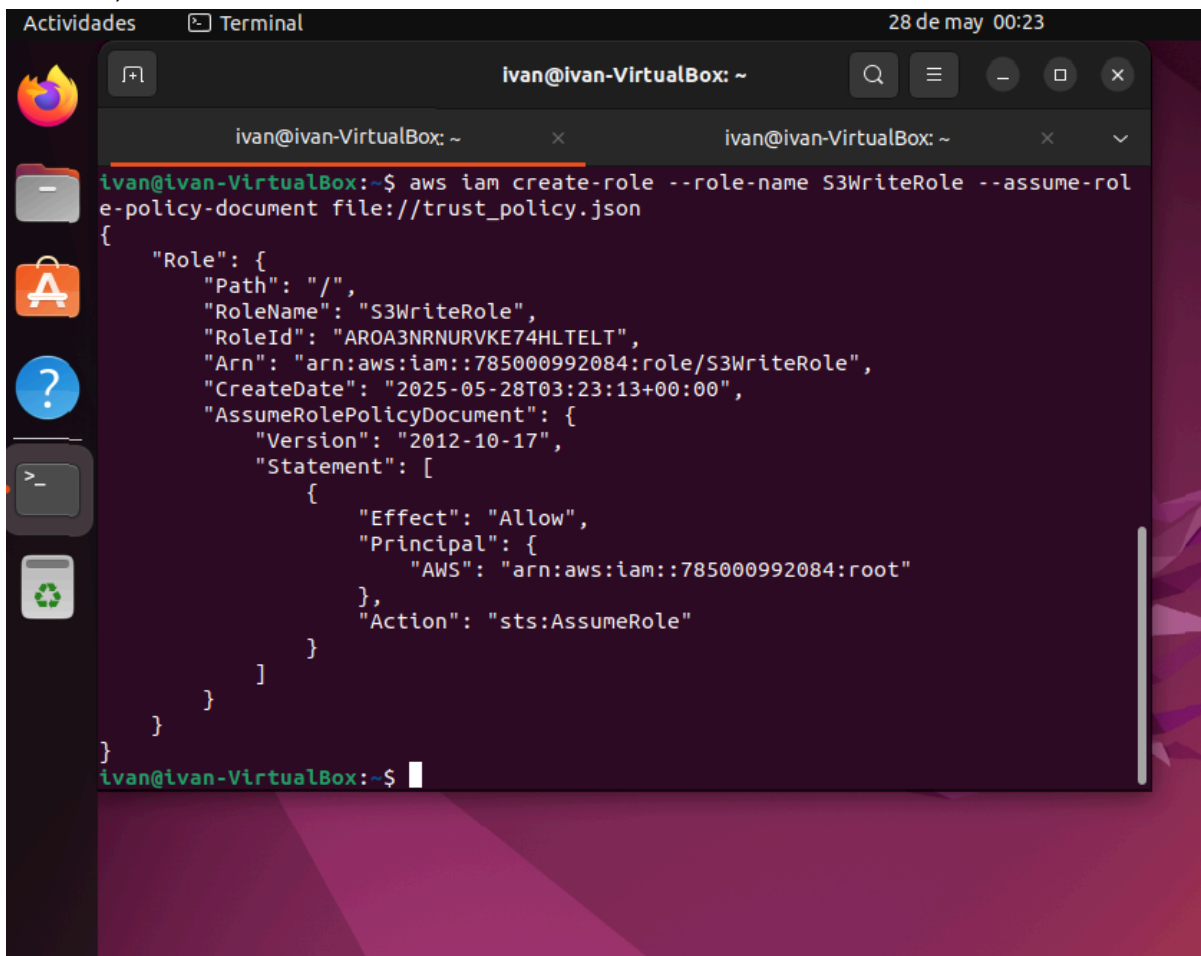


The screenshot shows a terminal window with the GNU nano 6.2 editor open. The file being edited is named 'trust_policy.json'. The content of the file is a JSON object representing an AWS IAM trust policy. The JSON structure is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::785000992084:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The terminal window also shows a status bar at the bottom with various keyboard shortcuts like 'Ayuda', 'Salir', 'Guardar', 'Leer fich.', 'Buscar', 'Reemplazar', 'Cortar', 'Pegar', 'Ejecutar', 'Justificar', 'Ubicación', and 'Ir a línea'.

5) Creamos el rol en aws.

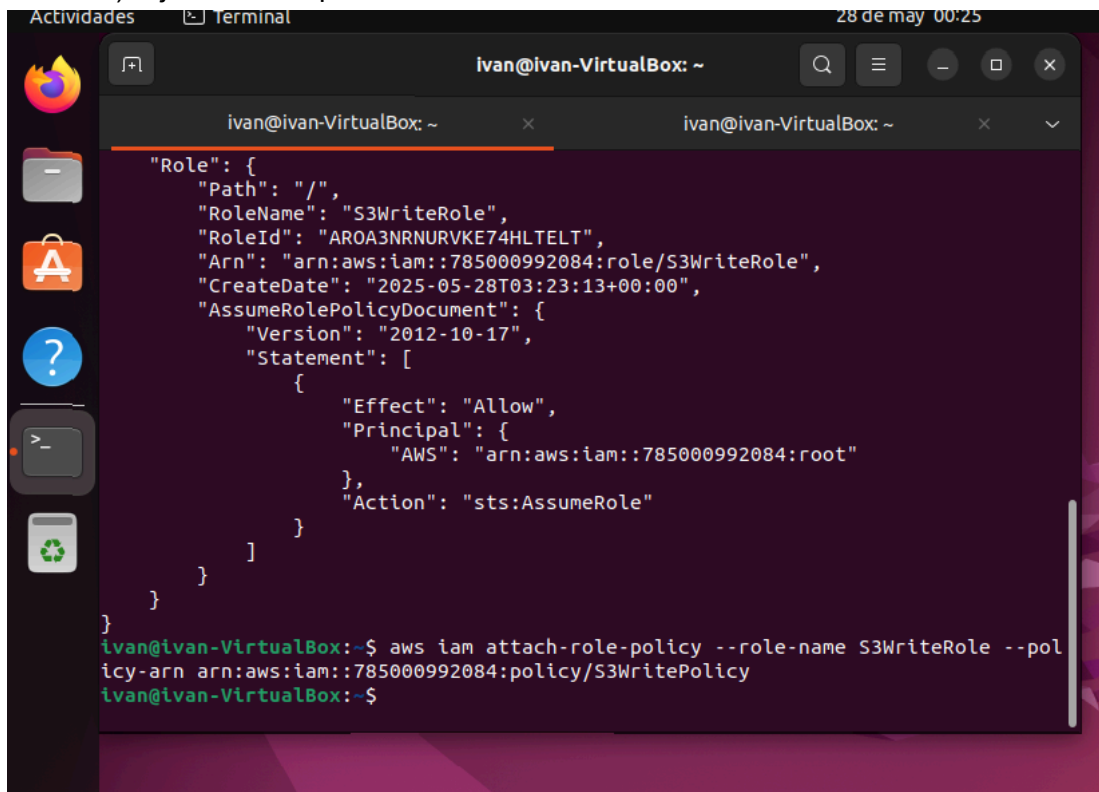


The screenshot shows a terminal window with the command prompt 'ivan@ivan-VirtualBox:~\$'. The command entered is 'aws iam create-role --role-name S3WriteRole --assume-role-policy-document file://trust_policy.json'. The output of the command is a JSON object representing the created role:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "S3WriteRole",
    "RoleId": "AROA3NRNURVKE74HLELT",
    "Arn": "arn:aws:iam::785000992084:role/S3WriteRole",
    "CreateDate": "2025-05-28T03:23:13+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::785000992084:root"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  }
}
```

The terminal window also shows the date and time '28 de may 00:23' in the top right corner.

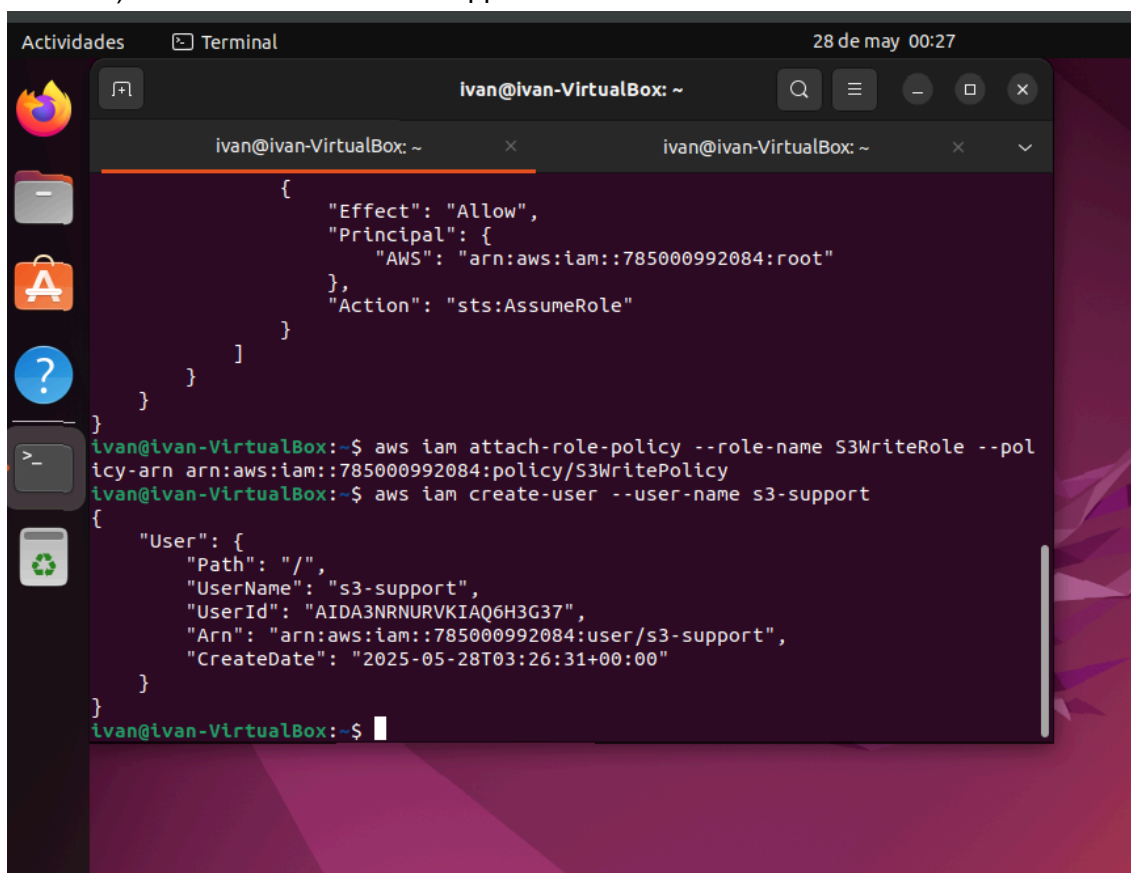
6) Adjuntamos la política de escritura en el bucket al Role.



```
ivan@ivan-VirtualBox: ~
"Role": {
  "Path": "/",
  "RoleName": "S3WriteRole",
  "RoleId": "AROA3NRNURVKE74HLTELT",
  "Arn": "arn:aws:iam::785000992084:role/S3WriteRole",
  "CreateDate": "2025-05-28T03:23:13+00:00",
  "AssumeRolePolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::785000992084:root"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

ivan@ivan-VirtualBox:~$ aws iam attach-role-policy --role-name S3WriteRole --pol
icy-arn arn:aws:iam::785000992084:policy/S3WritePolicy
ivan@ivan-VirtualBox:~$
```

7) Creamos el usuario s3-support.

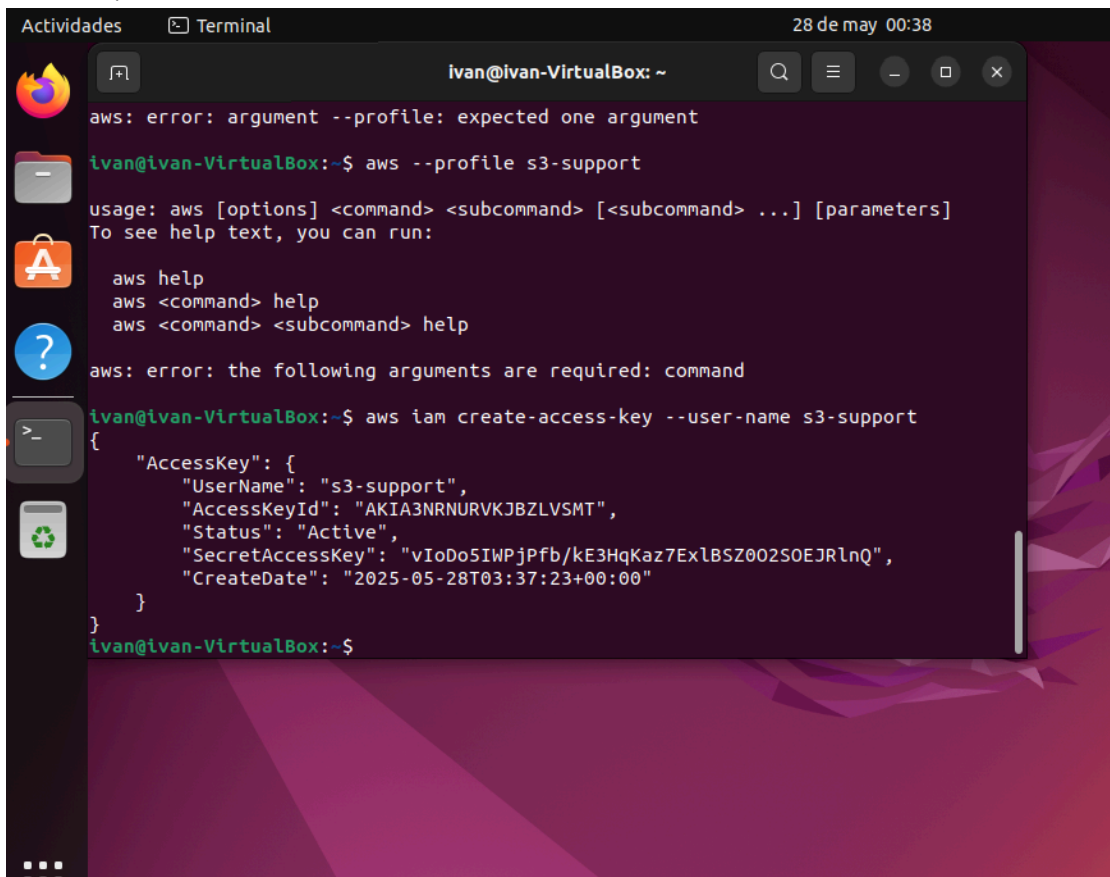


```
ivan@ivan-VirtualBox: ~
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::785000992084:root"
  },
  "Action": "sts:AssumeRole"
}
}
}
}
}

ivan@ivan-VirtualBox:~$ aws iam attach-role-policy --role-name S3WriteRole --pol
icy-arn arn:aws:iam::785000992084:policy/S3WritePolicy
ivan@ivan-VirtualBox:~$ aws iam create-user --user-name s3-support
{
  "User": {
    "Path": "/",
    "UserName": "s3-support",
    "UserId": "AIDA3NRNURVKIAQ6H3G37",
    "Arn": "arn:aws:iam::785000992084:user/s3-support",
    "CreateDate": "2025-05-28T03:26:31+00:00"
  }
}

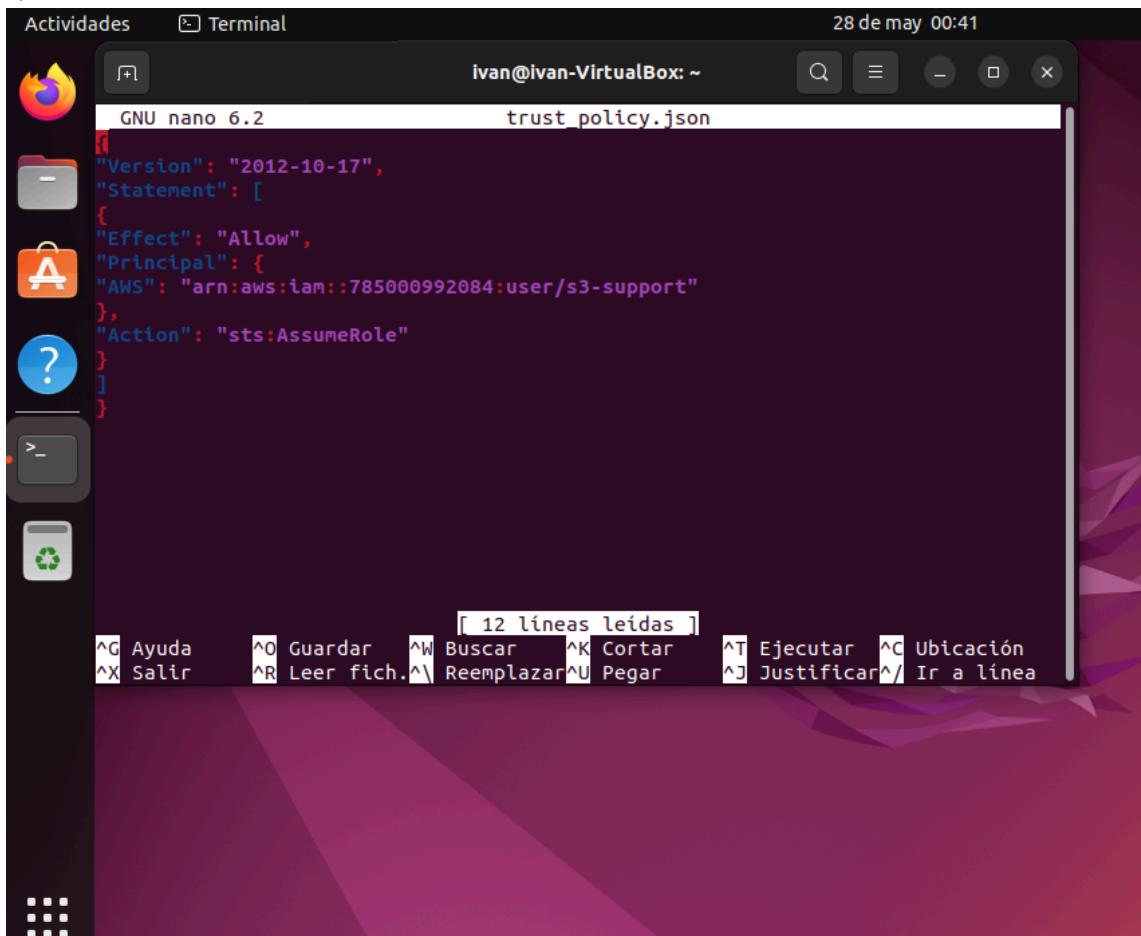
ivan@ivan-VirtualBox:~$
```

8) Obtenemos las credenciales del usuario s3-support.



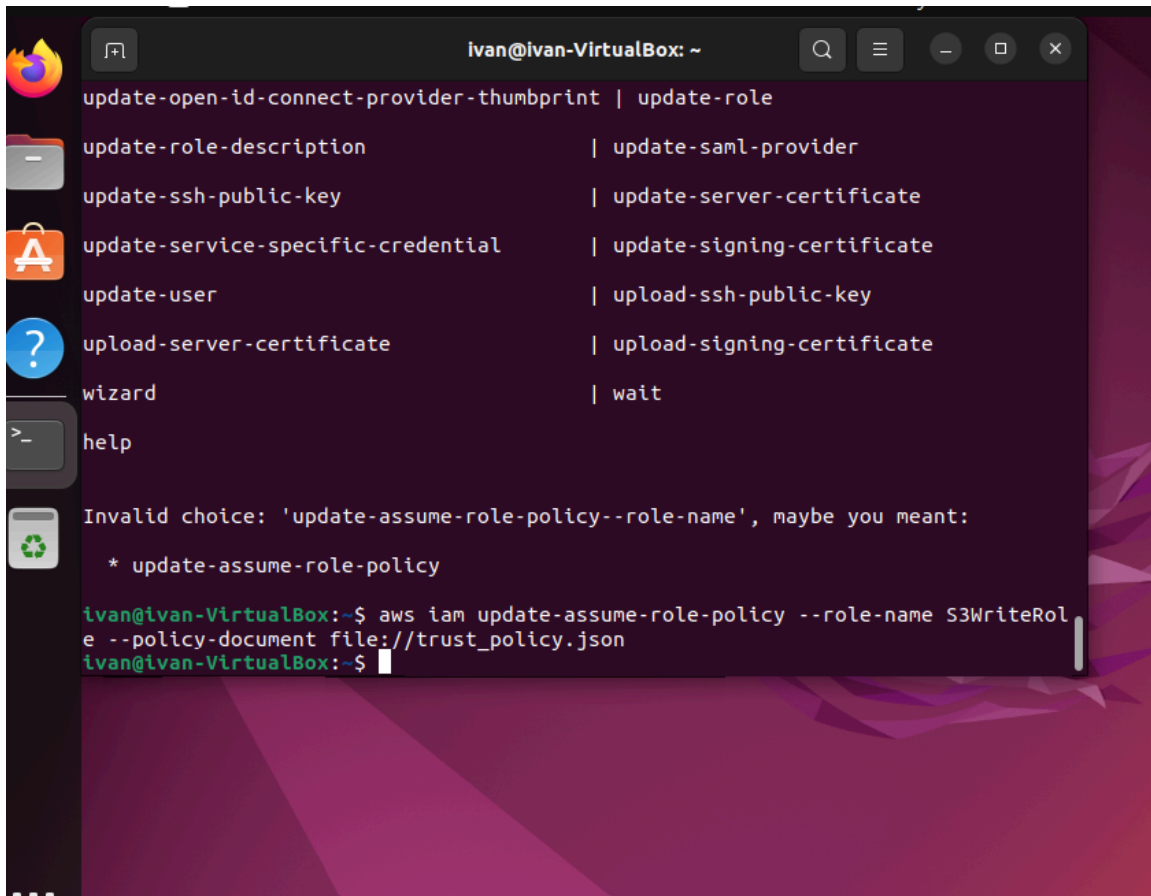
```
ivan@ivan-VirtualBox: ~  
aws: error: argument --profile: expected one argument  
ivan@ivan-VirtualBox:~$ aws --profile s3-support  
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]  
To see help text, you can run:  
  
aws help  
aws <command> help  
aws <command> <subcommand> help  
  
aws: error: the following arguments are required: command  
ivan@ivan-VirtualBox:~$ aws iam create-access-key --user-name s3-support  
{  
  "AccessKey": {  
    "UserName": "s3-support",  
    "AccessKeyId": "AKIA3NRNURVKJBZLVSM",  
    "Status": "Active",  
    "SecretAccessKey": "vIoDo5IWPjPfb/kE3HqKaz7ExlBSZ002S0EJRlnQ",  
    "CreateDate": "2025-05-28T03:37:23+00:00"  
  }  
}  
ivan@ivan-VirtualBox:~$
```

9) Modificamos el rol para que el usuario s3-support pueda asumirlo.



```
GNU nano 6.2 trust_policy.json  
{"Version": "2012-10-17",  
 "Statement": [  
   {  
     "Effect": "Allow",  
     "Principal": {  
       "AWS": "arn:aws:iam::785000992084:user/s3-support"  
     },  
     "Action": "sts:AssumeRole"  
   }  
 ]  
}  
[ 12 líneas leídas ]  
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea
```

10) Subimos el rol nuevamente.



```

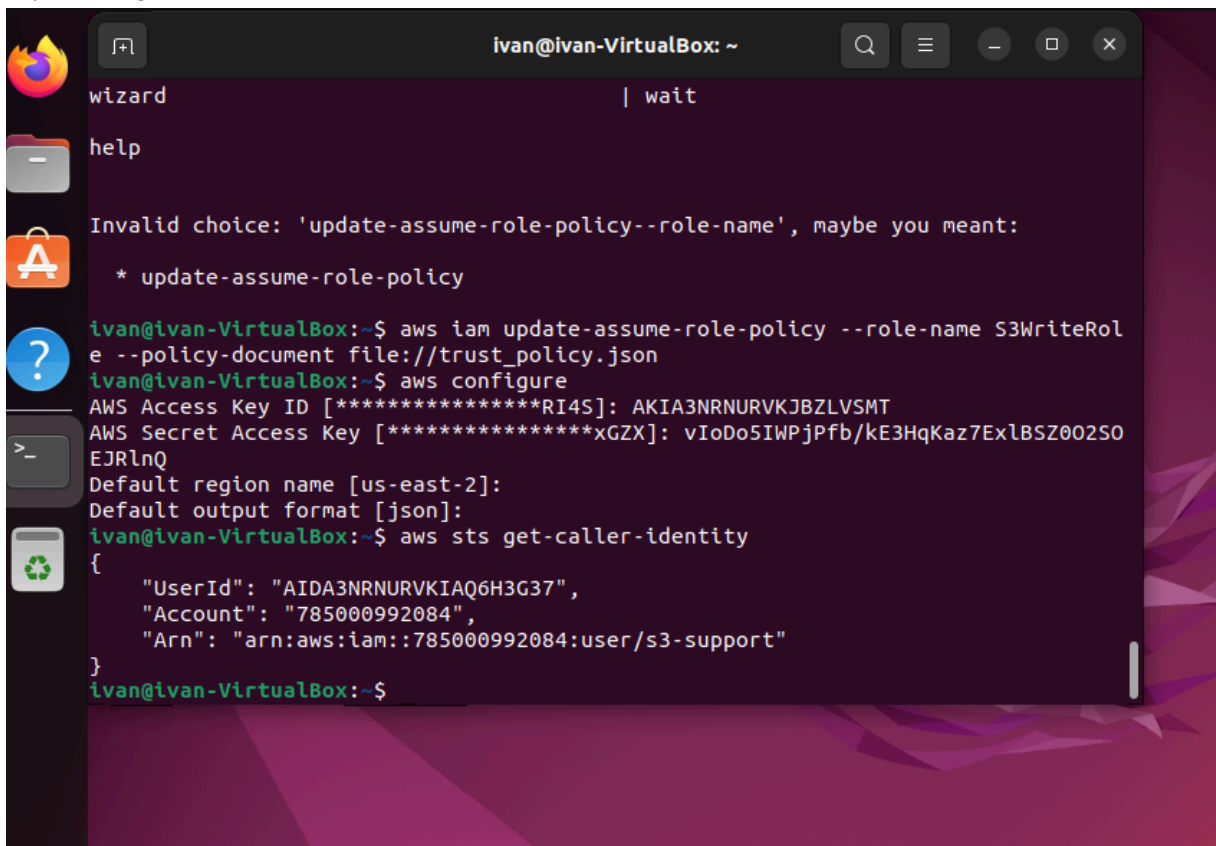
ivan@ivan-VirtualBox: ~
update-open-id-connect-provider-thumbprint | update-role
update-role-description | update-saml-provider
update-ssh-public-key | update-server-certificate
update-service-specific-credential | update-signing-certificate
update-user | upload-ssh-public-key
upload-server-certificate | upload-signing-certificate
wizard | wait
help

Invalid choice: 'update-assume-role-policy--role-name', maybe you meant:
* update-assume-role-policy

ivan@ivan-VirtualBox:~$ aws iam update-assume-role-policy --role-name S3WriteRole --policy-document file://trust_policy.json
ivan@ivan-VirtualBox:~$

```

11) Nos logueamos con el usuario s3-support.



```

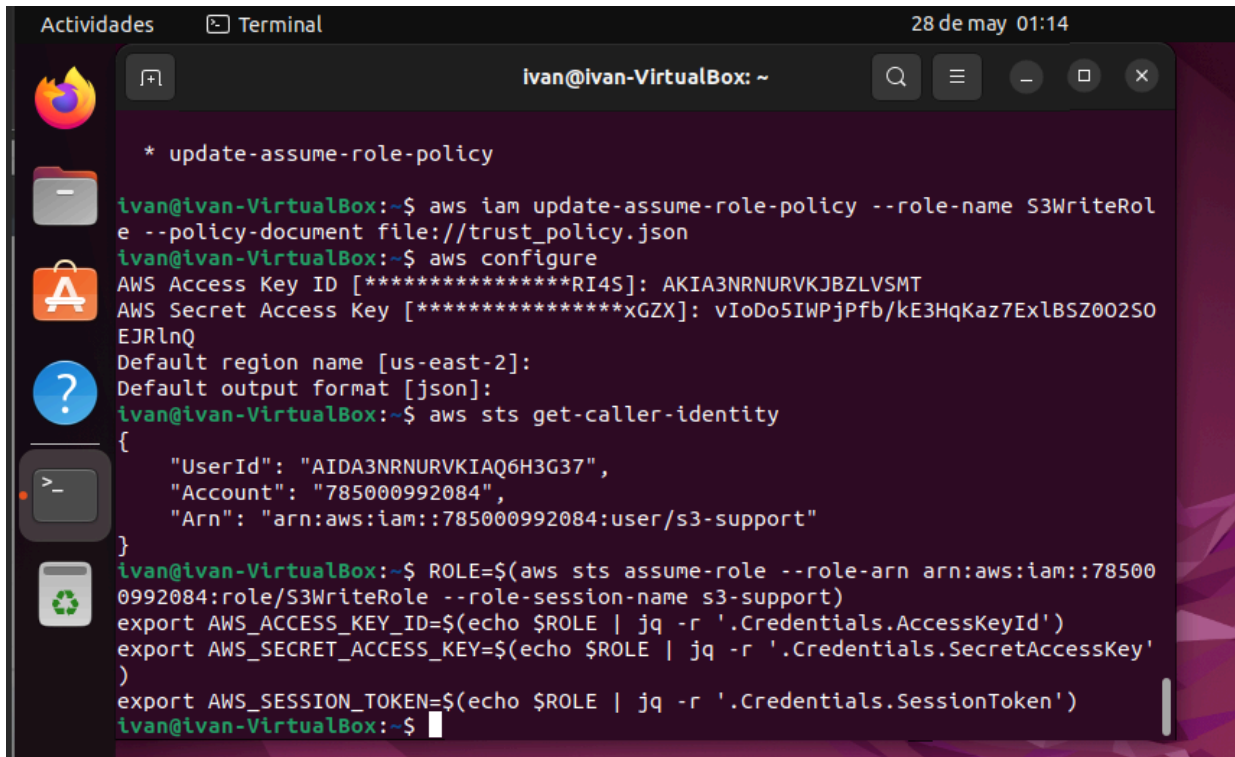
ivan@ivan-VirtualBox: ~
wizard | wait
help

Invalid choice: 'update-assume-role-policy--role-name', maybe you meant:
* update-assume-role-policy

ivan@ivan-VirtualBox:~$ aws iam update-assume-role-policy --role-name S3WriteRole --policy-document file://trust_policy.json
ivan@ivan-VirtualBox:~$ aws configure
AWS Access Key ID [*****RI4S]: AKIA3NRNURVKJBZLVSMTEJRLnQ
AWS Secret Access Key [*****xGZX]: vIoDo5IWPjPfb/kE3HqKaz7ExlBSZ002S0
Default region name [us-east-2]:
Default output format [json]:
ivan@ivan-VirtualBox:~$ aws sts get-caller-identity
{
  "UserId": "AIDA3NRNURVKIAQ6H3G37",
  "Account": "785000992084",
  "Arn": "arn:aws:iam::785000992084:user/s3-support"
}
ivan@ivan-VirtualBox:~$

```

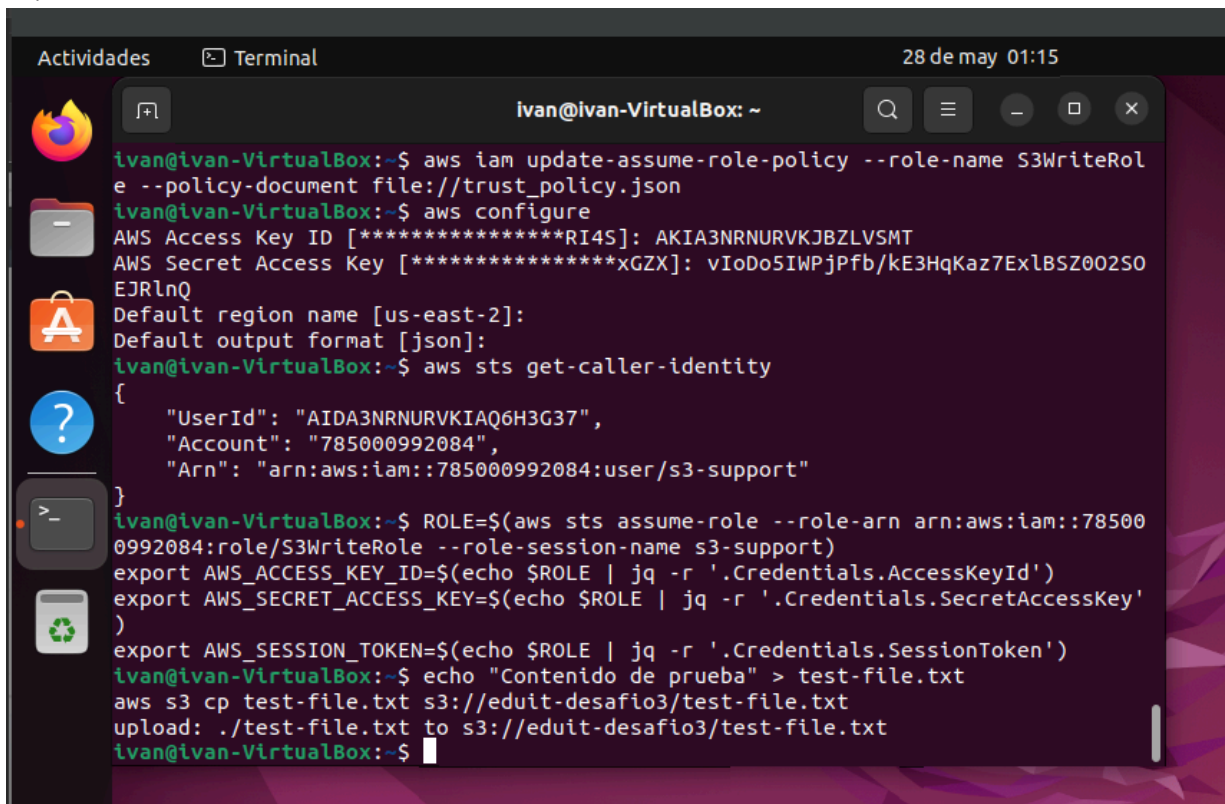

12) Asumimos el rol con el usuario s3-support y se obtienen las credenciales temporales.



```

Actividades Terminal 28 de may 01:14
ivan@ivan-VirtualBox: ~
* update-assume-role-policy
ivan@ivan-VirtualBox:~$ aws iam update-assume-role-policy --role-name S3WriteRole --policy-document file://trust_policy.json
ivan@ivan-VirtualBox:~$ aws configure
AWS Access Key ID [*****RI4S]: AKIA3NRNURVKJBZLVSM
AWS Secret Access Key [*****xGZX]: vIoDo5IWPjPfb/kE3HqKaz7ExlBSZ002SO
EJRLnQ
Default region name [us-east-2]:
Default output format [json]:
ivan@ivan-VirtualBox:~$ aws sts get-caller-identity
{
  "UserId": "AIDA3NRNURVKIAQ6H3G37",
  "Account": "785000992084",
  "Arn": "arn:aws:iam::785000992084:user/s3-support"
}
ivan@ivan-VirtualBox:~$ ROLE=$(aws sts assume-role --role-arn arn:aws:iam::785000992084:role/S3WriteRole --role-session-name s3-support)
export AWS_ACCESS_KEY_ID=$(echo $ROLE | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE | jq -r '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $ROLE | jq -r '.Credentials.SessionToken')
ivan@ivan-VirtualBox:~$
  
```

13) Subimos un archivo de prueba al bucket.



```

Actividades Terminal 28 de may 01:15
ivan@ivan-VirtualBox: ~
ivan@ivan-VirtualBox:~$ aws iam update-assume-role-policy --role-name S3WriteRole --policy-document file://trust_policy.json
ivan@ivan-VirtualBox:~$ aws configure
AWS Access Key ID [*****RI4S]: AKIA3NRNURVKJBZLVSM
AWS Secret Access Key [*****xGZX]: vIoDo5IWPjPfb/kE3HqKaz7ExlBSZ002SO
EJRLnQ
Default region name [us-east-2]:
Default output format [json]:
ivan@ivan-VirtualBox:~$ aws sts get-caller-identity
{
  "UserId": "AIDA3NRNURVKIAQ6H3G37",
  "Account": "785000992084",
  "Arn": "arn:aws:iam::785000992084:user/s3-support"
}
ivan@ivan-VirtualBox:~$ ROLE=$(aws sts assume-role --role-arn arn:aws:iam::785000992084:role/S3WriteRole --role-session-name s3-support)
export AWS_ACCESS_KEY_ID=$(echo $ROLE | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE | jq -r '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $ROLE | jq -r '.Credentials.SessionToken')
ivan@ivan-VirtualBox:~$ echo "Contenido de prueba" > test-file.txt
aws s3 cp test-file.txt s3://eduit-desafio3/test-file.txt
upload: ./test-file.txt to s3://eduit-desafio3/test-file.txt
ivan@ivan-VirtualBox:~$
  
```

14) Al intentar leer contenido del bucket con el usuario s3-support vemos que no posee permisos, ahora si cambiamos al profile de admin que es el usuario con el que empezamos el desafío vemos que el archivo "test-file.txt" se creó correctamente en el bucket.

```

ivan@ivan-VirtualBox: ~
ivan@ivan-VirtualBox: ~$ ROLE=$(aws sts assume-role --role-arn arn:aws:iam::785000992084:role/S3WriteRole --role-session-name s3-support)
export AWS_ACCESS_KEY_ID=$(echo $ROLE | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE | jq -r '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $ROLE | jq -r '.Credentials.SessionToken')
ivan@ivan-VirtualBox:~$ echo "Contenido de prueba" > test-file.txt
aws s3 cp test-file.txt s3://eduit-desafio3/test-file.txt
upload: ./test-file.txt to s3://eduit-desafio3/test-file.txt
ivan@ivan-VirtualBox:~$ aws s3 ls s3://eduit-desafio3/

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User:
arn:aws:sts::785000992084:assumed-role/S3WriteRole/s3-support is not authorized
to perform: s3:ListBucket on resource: "arn:aws:s3:::eduit-desafio3" because no
identity-based policy allows the s3:ListBucket action
ivan@ivan-VirtualBox:~$ aws s3 ls s3://eduit-desafio3

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User:
arn:aws:sts::785000992084:assumed-role/S3WriteRole/s3-support is not authorized
to perform: s3:ListBucket on resource: "arn:aws:s3:::eduit-desafio3" because no
identity-based policy allows the s3:ListBucket action
ivan@ivan-VirtualBox:~$ aws s3 ls s3://eduit-desafio3 --profile admin
2025-05-28 01:15:47          20 test-file.txt
ivan@ivan-VirtualBox:~$

```

15) verificamos ingresando por consola web,

eduit-desafio3 Información

Objetos | Metadatos | Propiedades | Permisos | Métricas | Administración | Puntos de acceso

Objetos (1)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	test-file.txt	txt	28 May 2025 1:15:47 AM -03	20.0 B	Estándar