Reference: https://dev.to/aws-builders/aws-importexport-part-1-import-1kpj

1. Export Virtual Machine from On-premise

I have already done this.

https://neu-cy5150-2023-fall-pentest.s3.us-east-2.amazonaws.com/Pentest_LAB_new-disk1.vmdk

Upload virtual machine to AWS

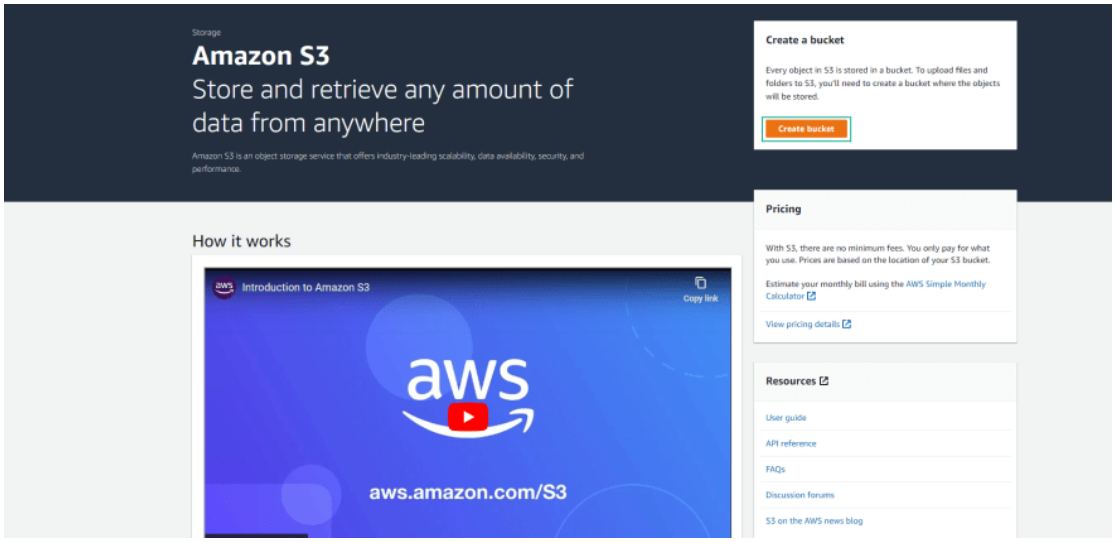
In this step, we will use Amazon S3 to store the virtual machine file that has been exported from the virtualized environment.

2. Create S3 bucket to store virtual machines

To create an S3 bucket, we perform the following steps:

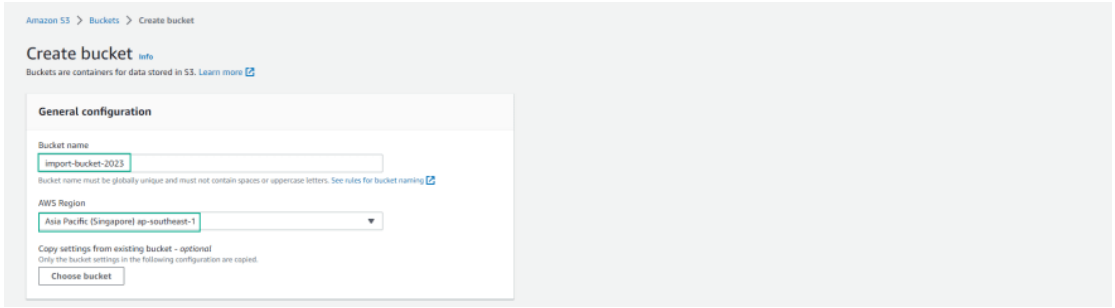
Access the Amazon S3 Management console.

- In the navigation bar, select Buckets.
- Select **Create bucket** to create a new S3 bucket.

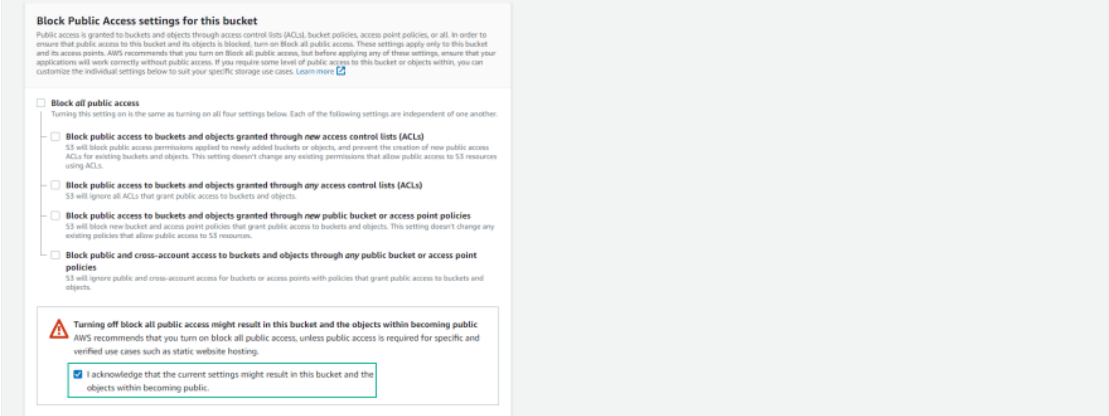


On the Create bucket page, set the parameters for the S3 bucket.

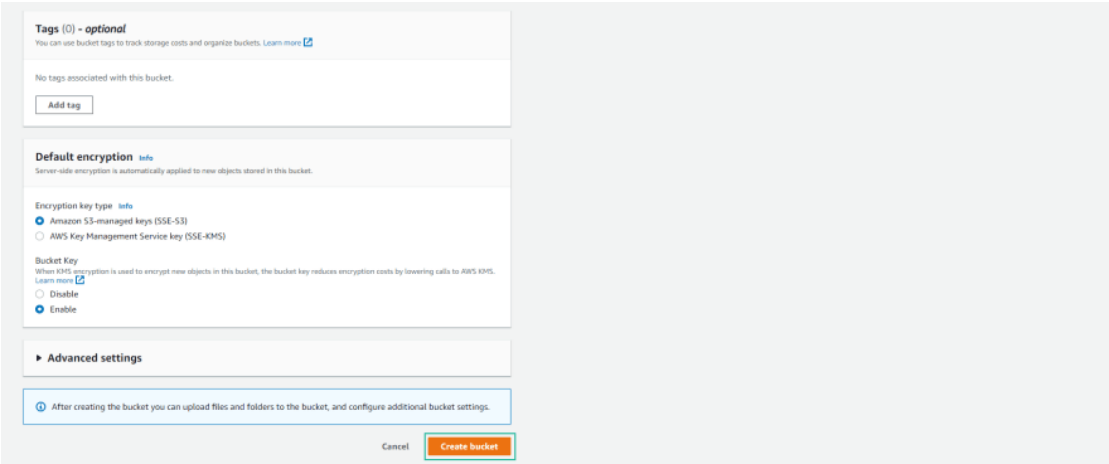
- **Bucket name**: Enter the bucket name. This name must be unique and not duplicate. (Example: import-bucket-2023)
- **Region**: Select the storage region of the bucket.



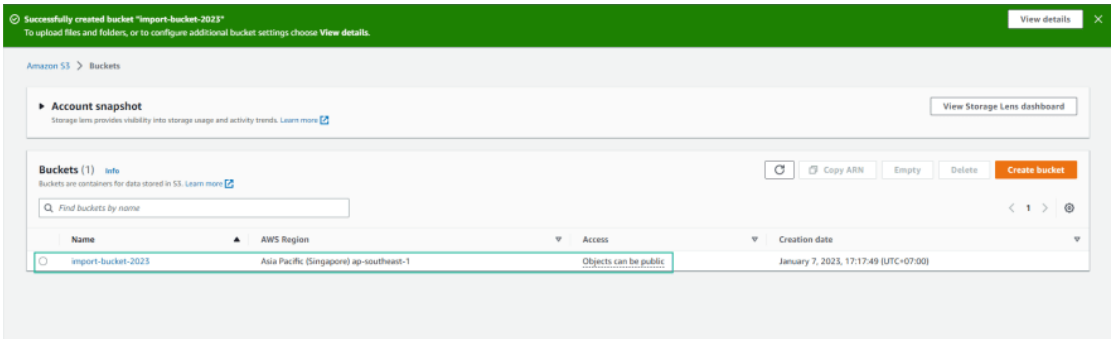
Uncheck **Block all public access** to allow public access. AWS will then issue a warning, and you select **I acknowledge that the current settings might result in this bucket and the objects within becoming public**.



Select **Create bucket**.



Successful bucket creation

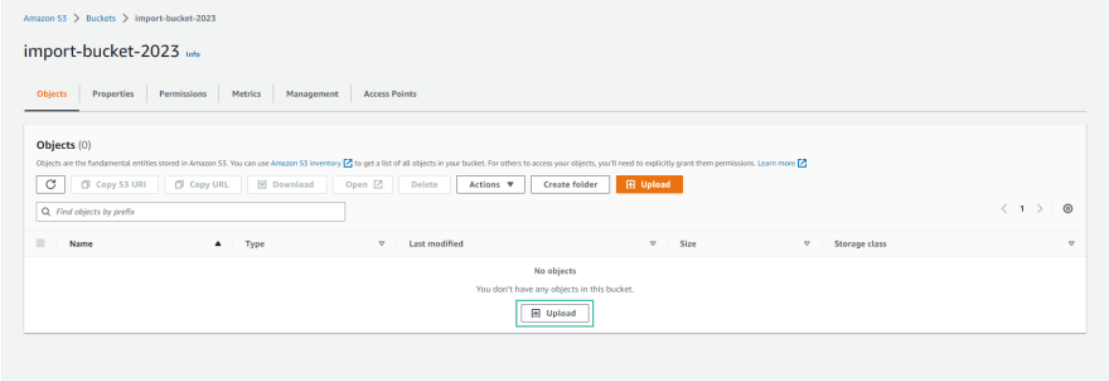


3. Upload virtual machine to S3 Bucket

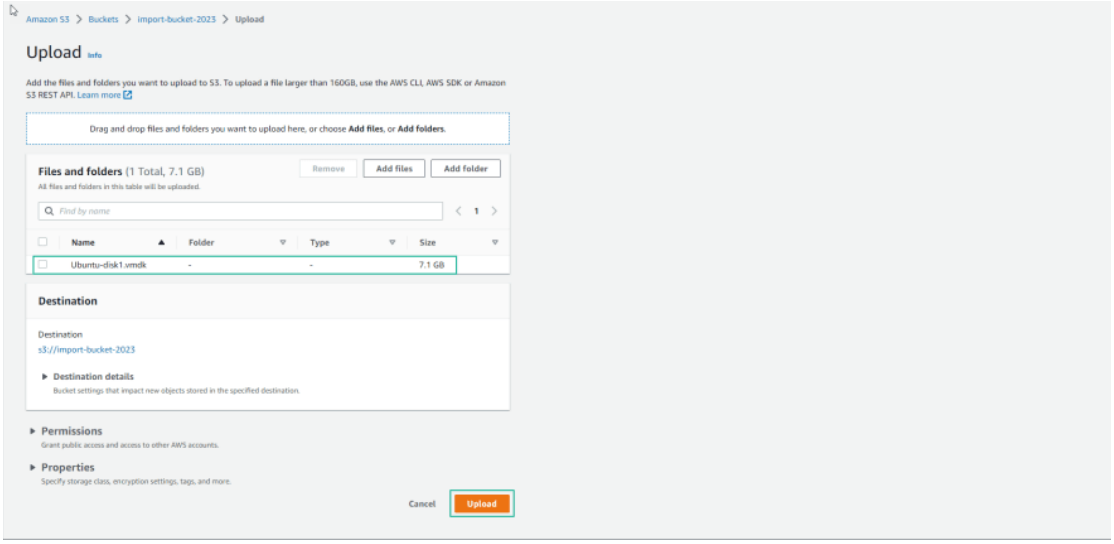
After creating the bucket, we will proceed to upload the virtual machine file that we exported in the previous section.

- Access to the S3 bucket you created above. (Example: import-bucket-2023)

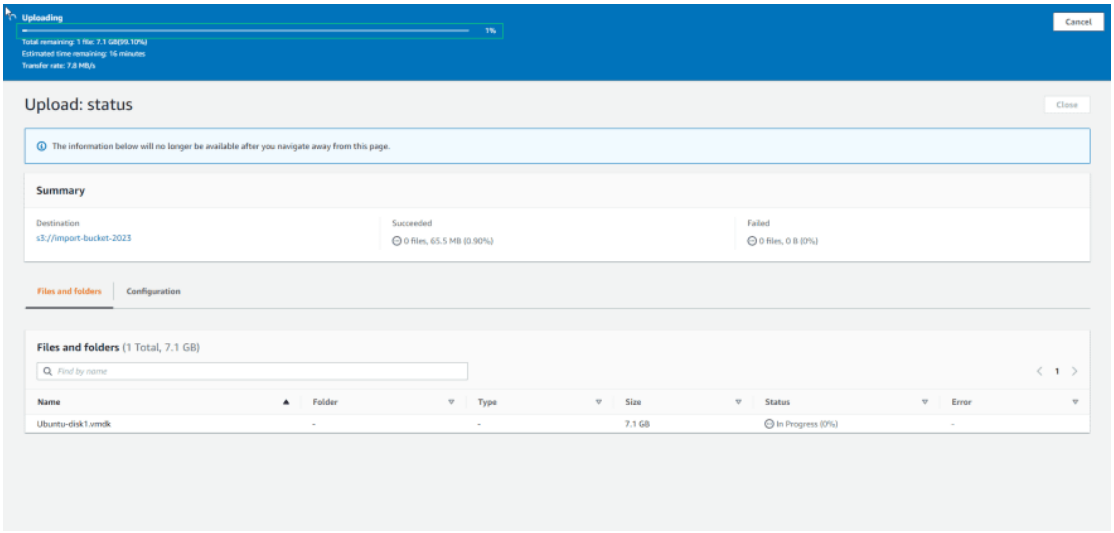
- In the **Objects** section, select **Upload**



Drag and drop the exported virtual machine file from the on-prem virtualization environment into the window or select Add files to select the virtual machine file. Then select **Upload**.
You create a virtual machine using VMWare Workstation, the virtual machine file in the example is Ubuntu-disk1.vmdk.



It will take some time for the file to be uploaded to the S3 bucket.



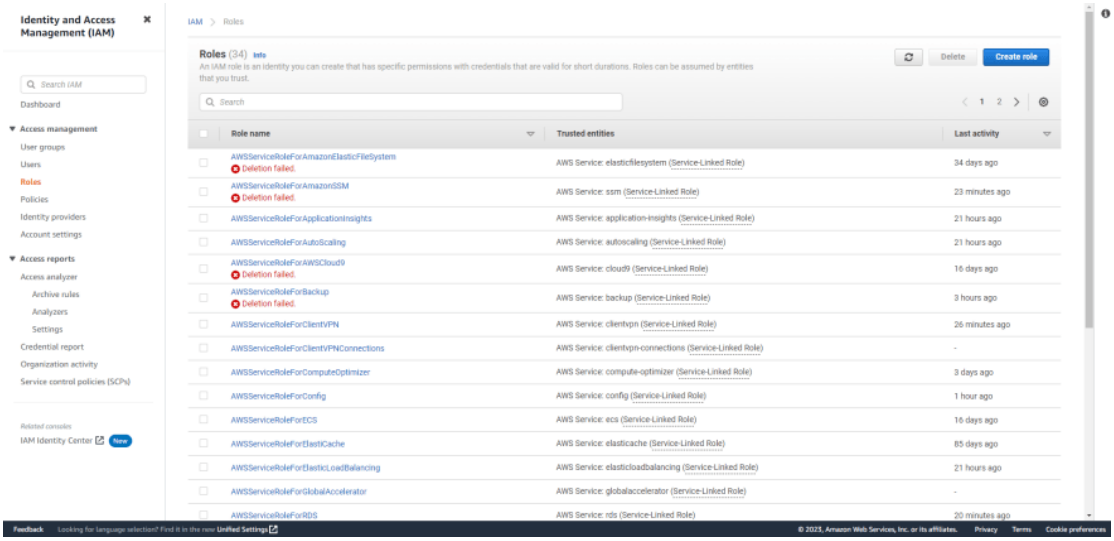
4. Import virtual machine to AWS

In this step, you will create a role named **vmimport** and import the virtual machine that was uploaded to the S3 Bucket in the previous step into an AMI. The entire process will be handled with the AWS CLI.

Create vmimport role

Before performing the Import of virtual machines into AWS. You need to check the role required for this implementation.

Access the IAM Management console.
In the navigation bar, select Roles

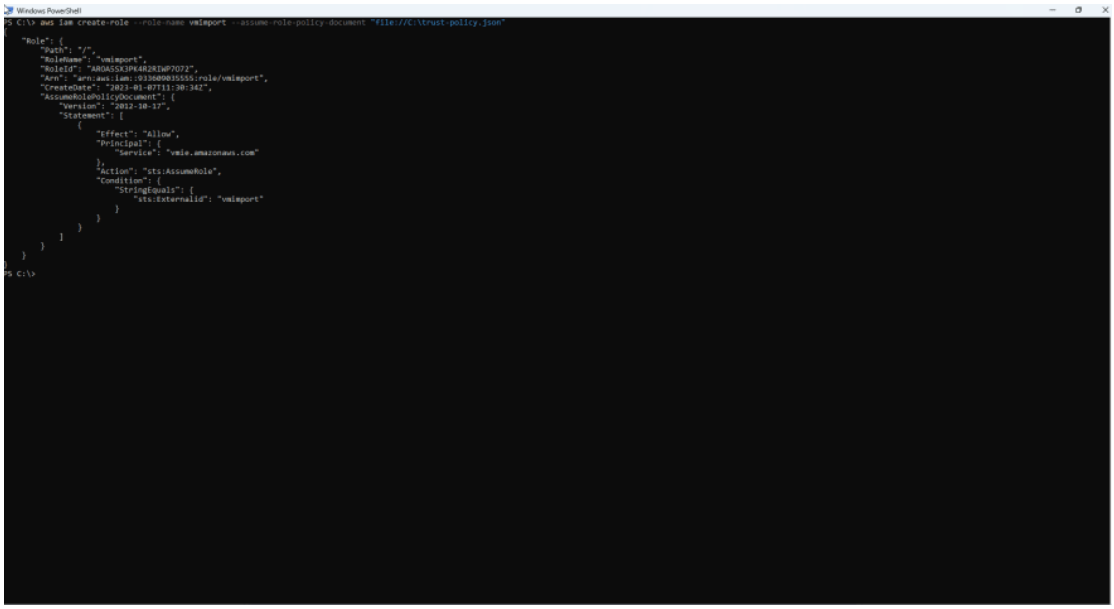


If you do not see the vmimport role, proceed to create the vmimport role.
Create a file named trust-policy.json to allow the VM Import/Export service to accept your upcoming vmimport role.

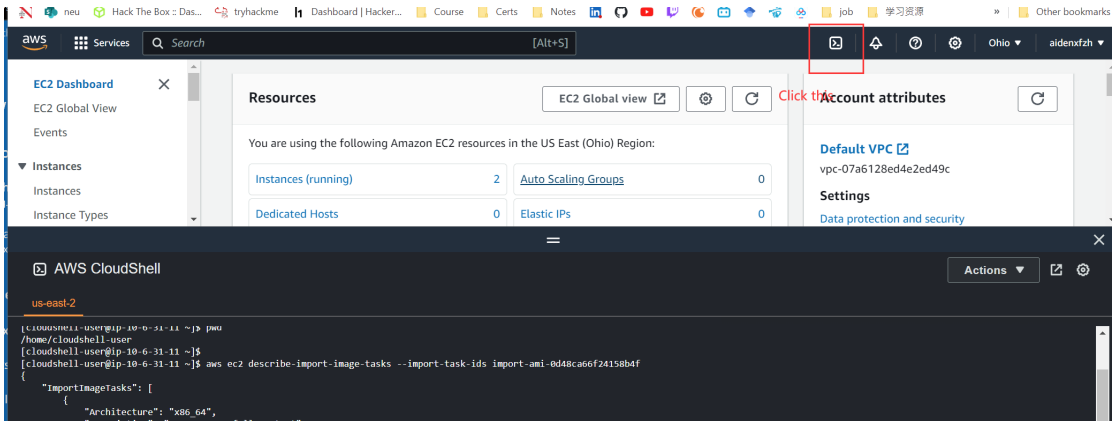
```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": { "Service": "vmie.amazonaws.com" },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals":{
               "sts:Externalid": "vmimport"
            }
         }
      }
   ]
}
```

Use the create-role command to create an IAM role named vmimport and assign **trust-policy.json** to the parameter **--assume-role-policy-document**

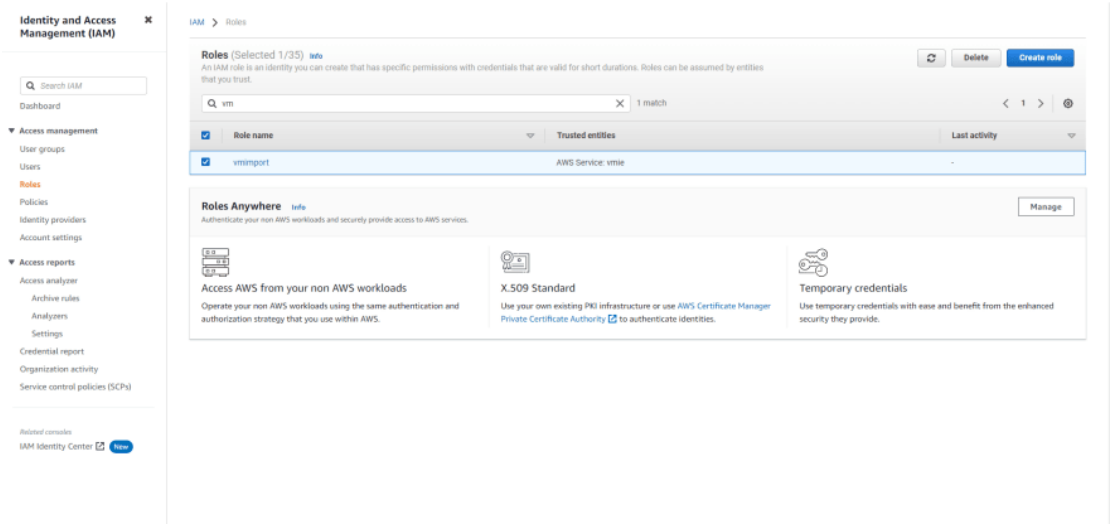
```
# replace "E:\trust-policy.json" with the path to the trust-policy.json file on your environment

aws iam create-role --role-name vmimport --assume-role-policy-document "file://E:\trust-policy.json"
```



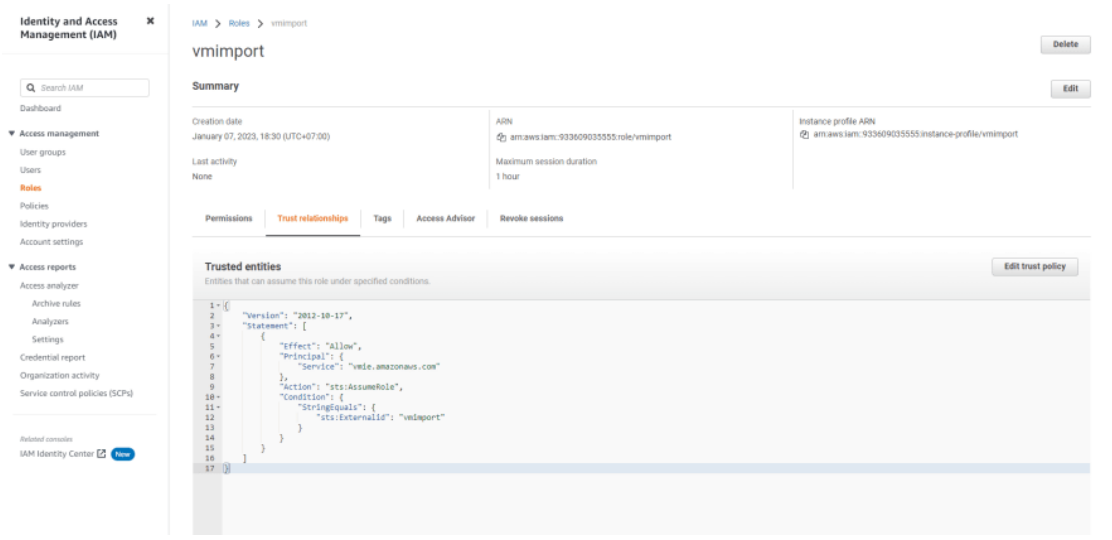
If you haven't configured aws cli on your computer, you can instead use aws cloudshell to run the commands and put your json file in it.



Check the created role.



See **Trust relationships**



Create a file role-policy.json containing the following policies to allow the IAM role to access buckets containing virtual machines to exercise the permissions in the "Action" section:. Inside:

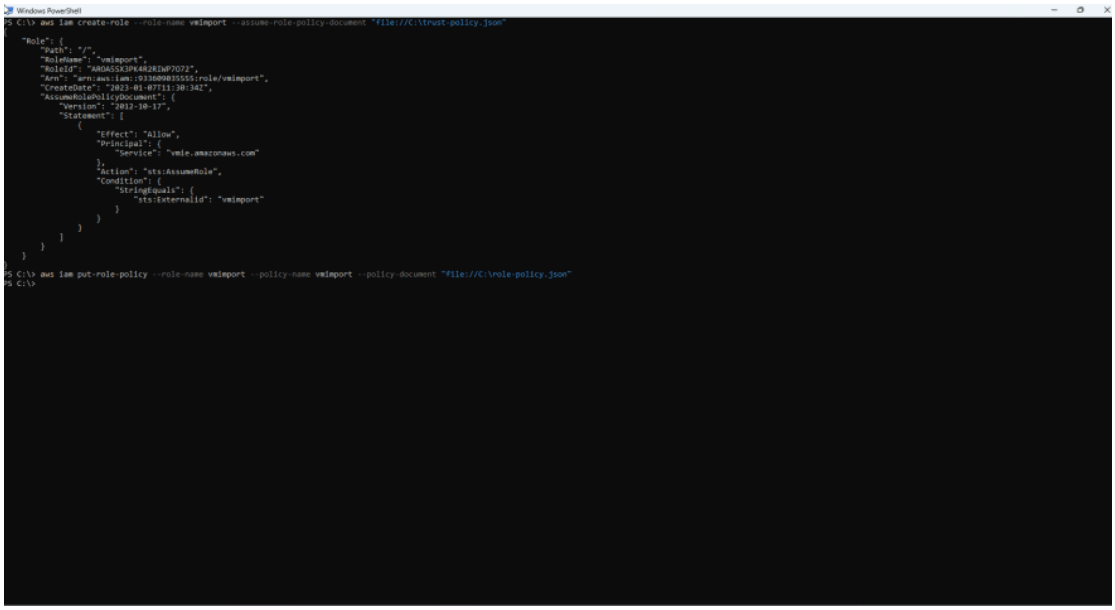
- disk-image-file-bucket is the name of the S3 bucket used to store the exported files from onpremise (import-bucket-2023 in this example).
- export-bucket is the name of the S3 bucket used to export the ec2 instance that will be used for the Export VM from AWS later.

{

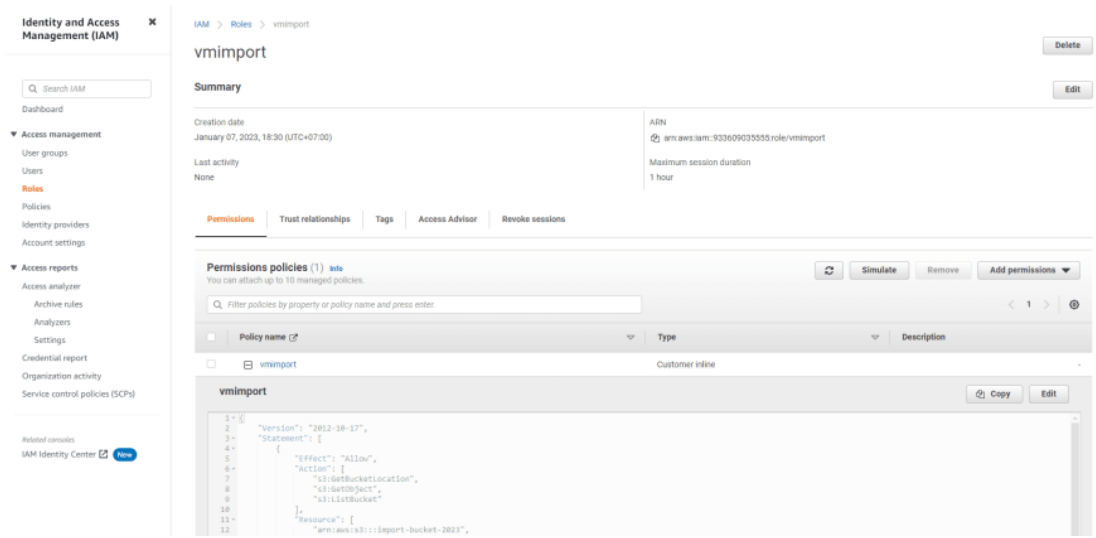
```
   "Version":"2012-10-17",
   "Statement":[
     {
       "Effect": "Allow",
       "Action": [
         "s3:GetBucketLocation",
         "s3:GetObject",
         "s3:ListBucket"
       ],
       "Resource": [
         "arn:aws:s3:::disk-image-file-bucket",
         "arn:aws:s3:::disk-image-file-bucket/*"
       ]
     },
     {
       "Effect": "Allow",
       "Action": [
         "s3:GetBucketLocation",
         "s3:GetObject",
         "s3:ListBucket",
         "s3:PutObject",
         "s3:GetBucketAcl"
       ],
       "Resource": [
         "arn:aws:s3:::export-bucket",
         "arn:aws:s3:::export-bucket/*"
       ]
     },
     {
       "Effect": "Allow",
       "Action": [
         "ec2:ModifySnapshotAttribute",
         "ec2:CopySnapshot",
         "ec2:RegisterImage",
         "ec2:Describe*"
       ],
       "Resource": "*"
     }
   ]
}
```

- Use the following command to assign the roles described in the role-policy.json file to the created vmimport role

aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document "file://E:\role-policy.json"



Check permissions. You can also check to see if the vmimport role has been successfully created by going to the IAM Management Console and selecting the role. You can also edit the role policy directly by selecting Edit policy.



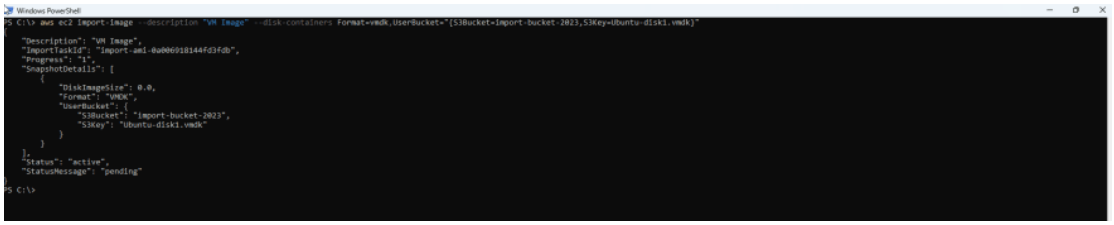
Import virtual machine to AMI

We will use the AWS CLI to launch the Import virtual machine to AMI process.

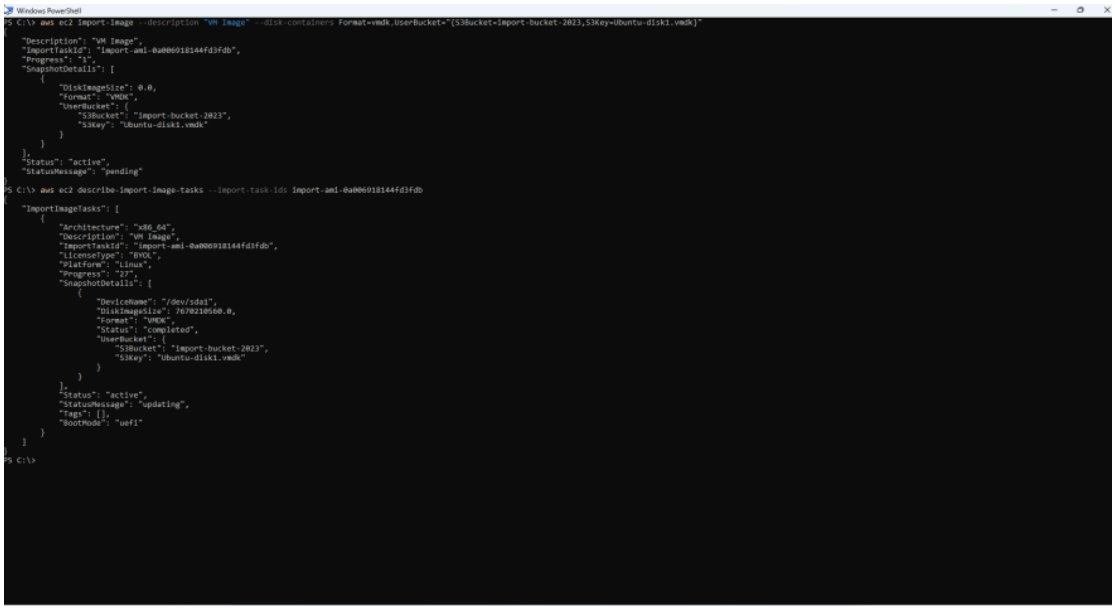
In Terminal on Linux (or Command Prompt/Power Shell on Windows), run the command aws ec2 import-image to start importing the exported virtual machine and convert it to AMI. The following settings are relevant:

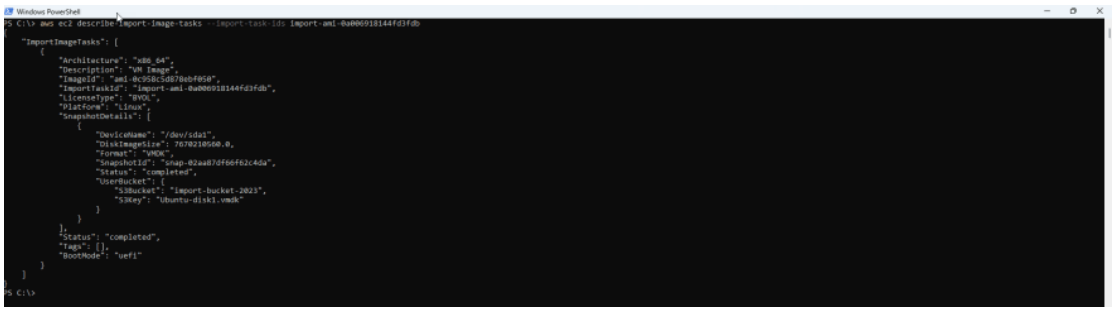
- **--deescription**: Set description for AMI
- **--disk-ccontainers**: Contains information identifying virtual machine files such as:
- Format format (eg: vhdx or vmdk)
- Storage bucket (eg import-bucket-2023)
- File path (e.g. Ubuntu.vhdx or Ubuntu-disk1.vmdk)

aws ec2 import-image --description "VM Image" --disk-containers Format=vhdx,UserBucket="{S3Bucket=import-bucket-2021,S3Key=Ubuntu.vhdx}"

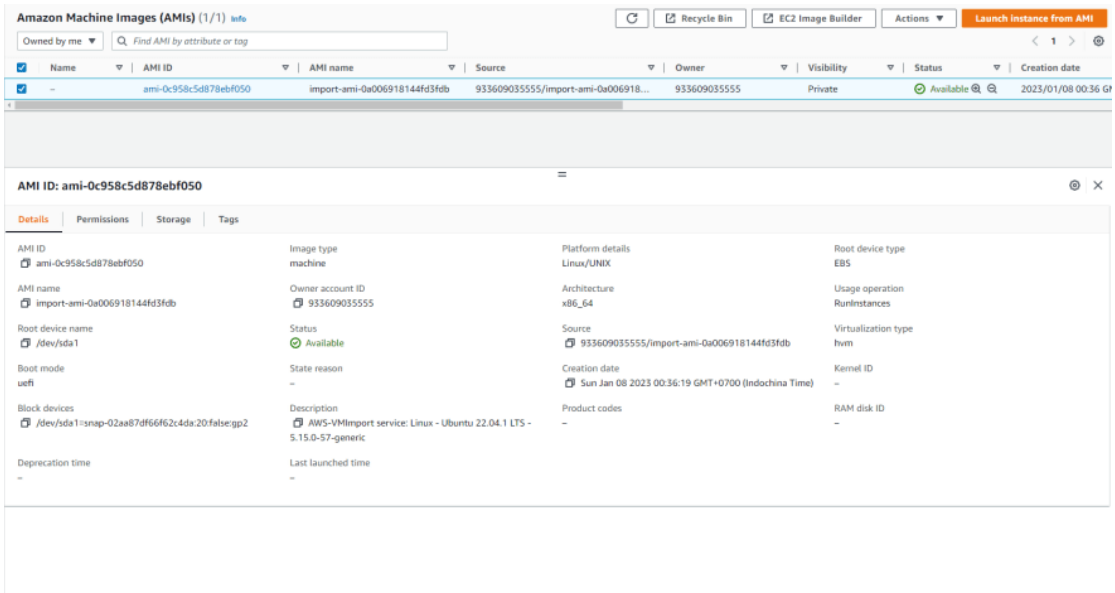


It will take 5-10 minutes depending on the size of the virtual machine for AWS to convert the virtual machine into an AMI.

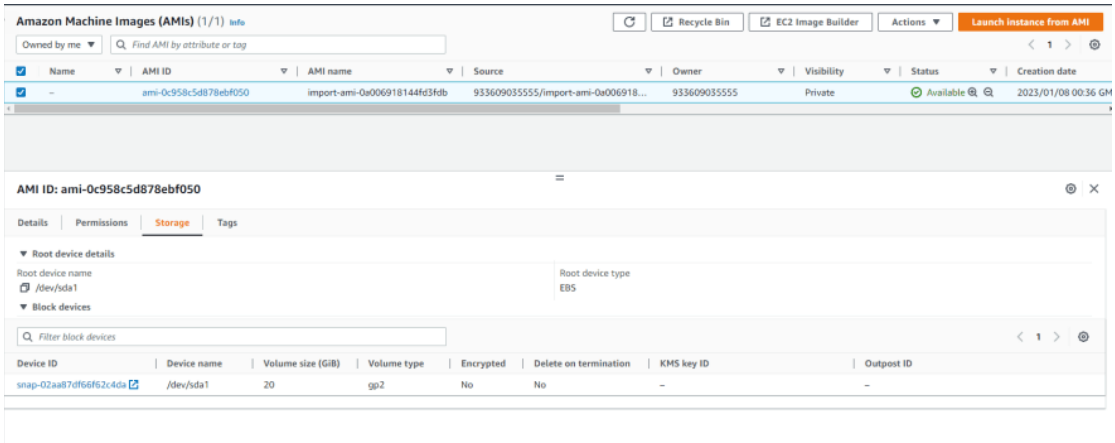




Once completed, we will see in the AMI list there will be one more AMI with the AMI name being the task id we created above.



You must check that EBS is not **Encrypted**

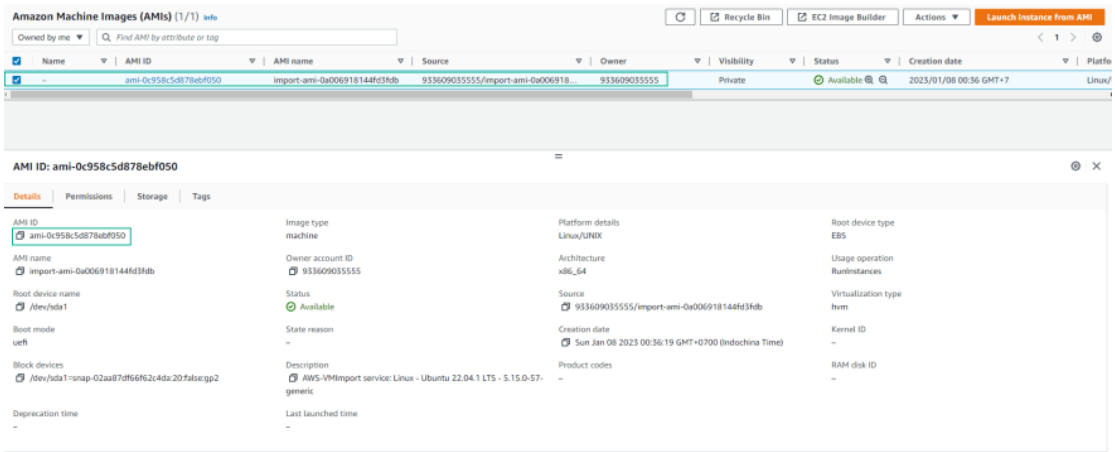


5. Deploy EC2 Instance from AMI

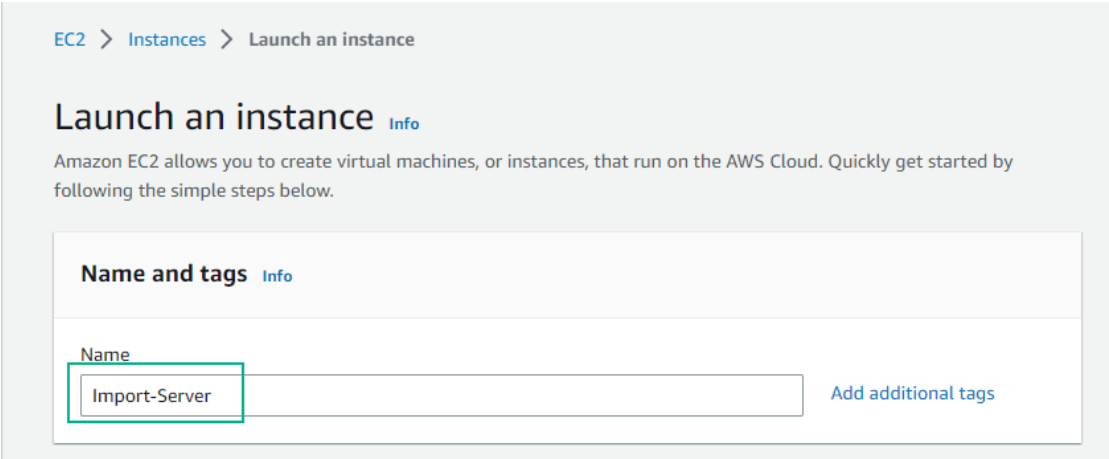
To deploy the virtual machine from the imported AMI, we perform the following steps:

To deploy the virtual machine from the imported AMI, we perform the following steps:

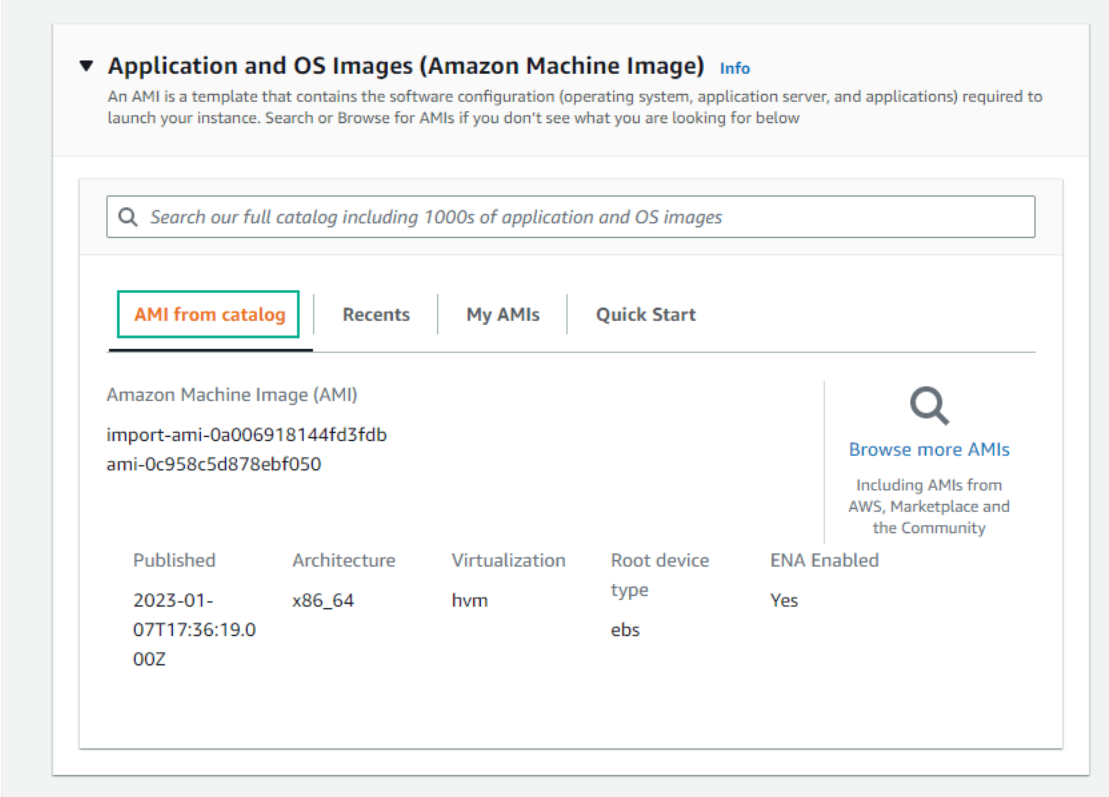
- Access to EC2 Management console.
- In the navigation bar, select AMIs.
- Select the AMI you just imported from the virtual machine (eg import-ami-08a9efac866dfcb04). Then select Launch.



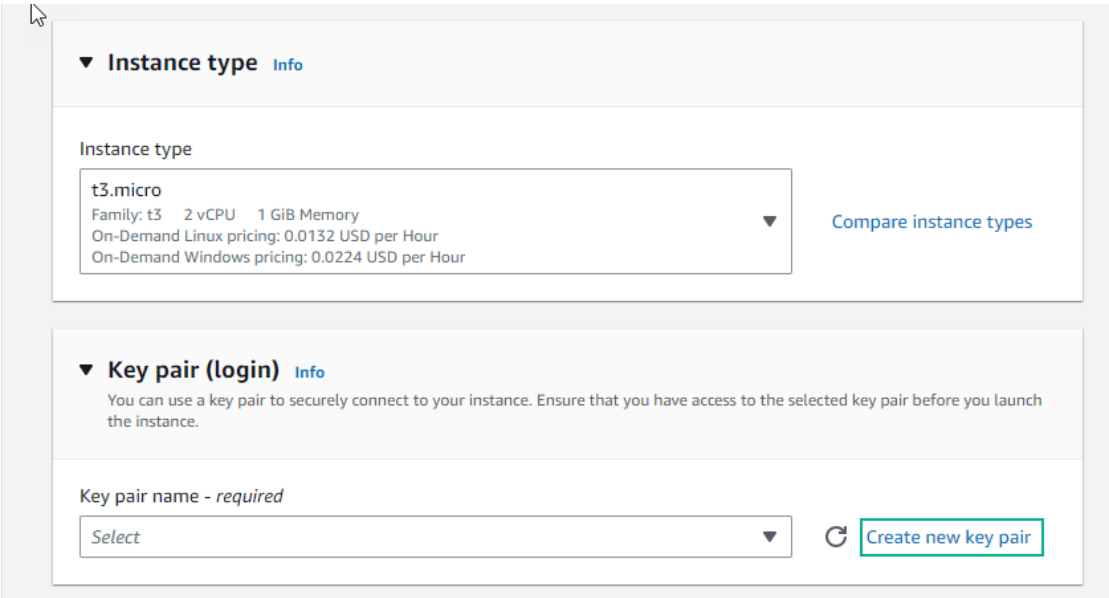
Name, enter **Import-Server**



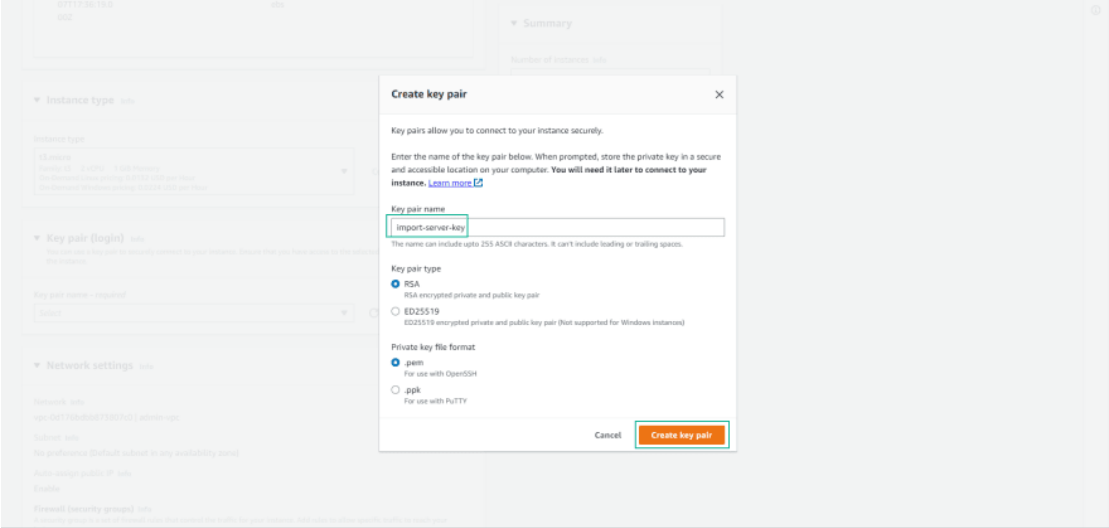
Keep the default AMI.



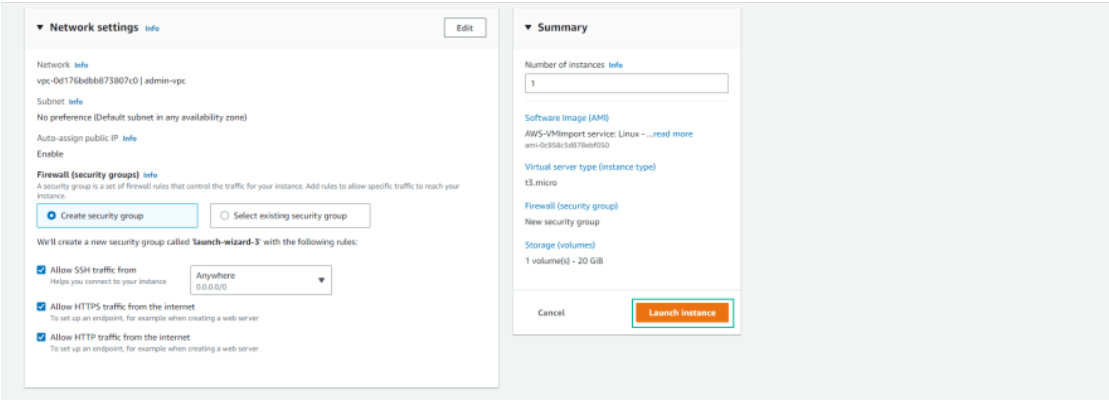
Keep **Instance type** and select **Create new key pair**



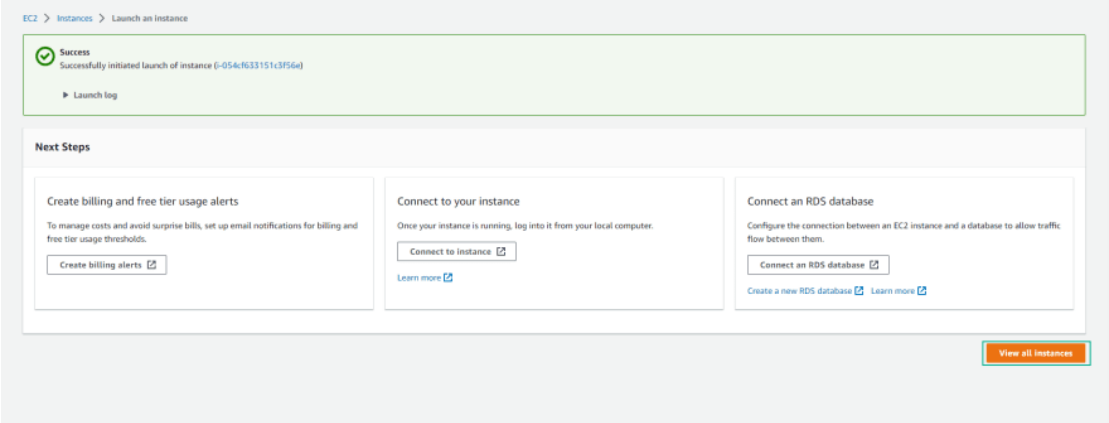
Fill in the key pair information and select **Create key pair**



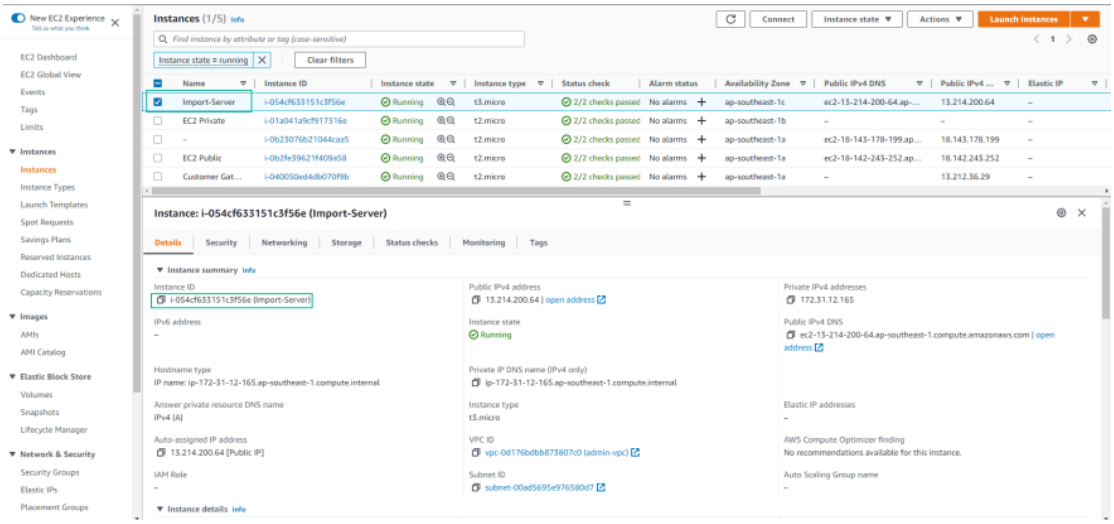
Leave the default **Network settings**



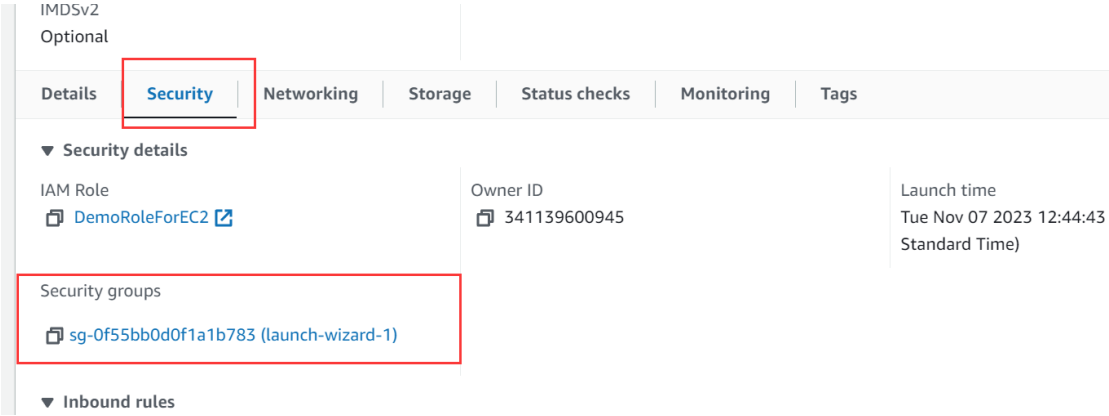
Select **View all instances**



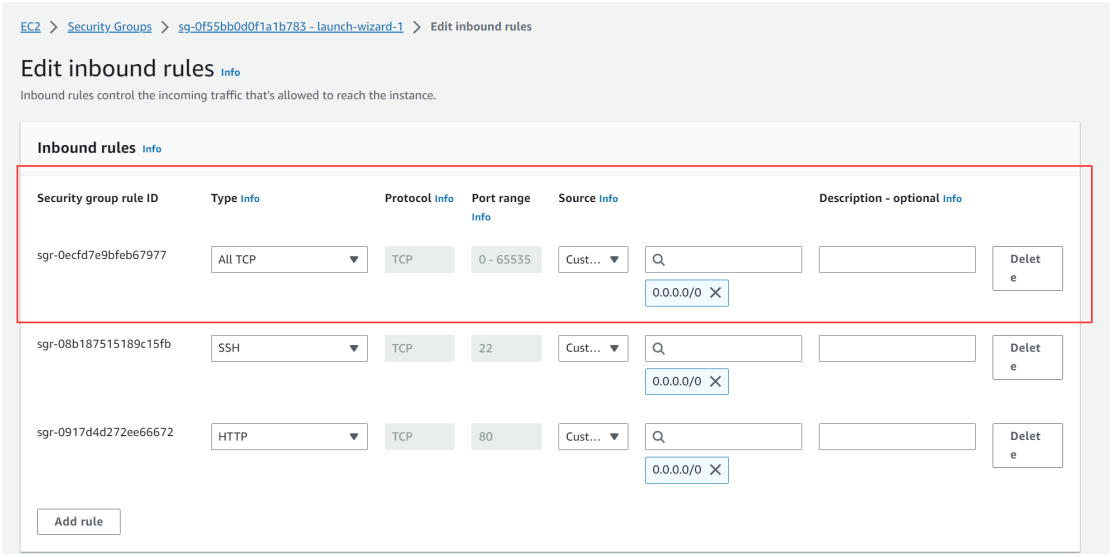
Check the created instance.

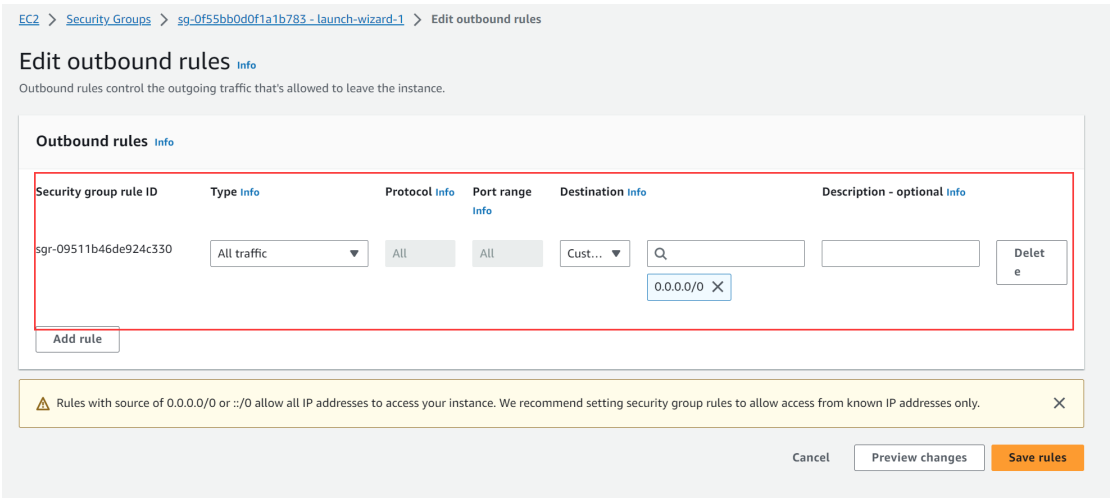


Click the instance ID. Then click security->security groups



Edit inbound and outbound rules to allow all tcp traffic





Then, happy pentesting.