

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

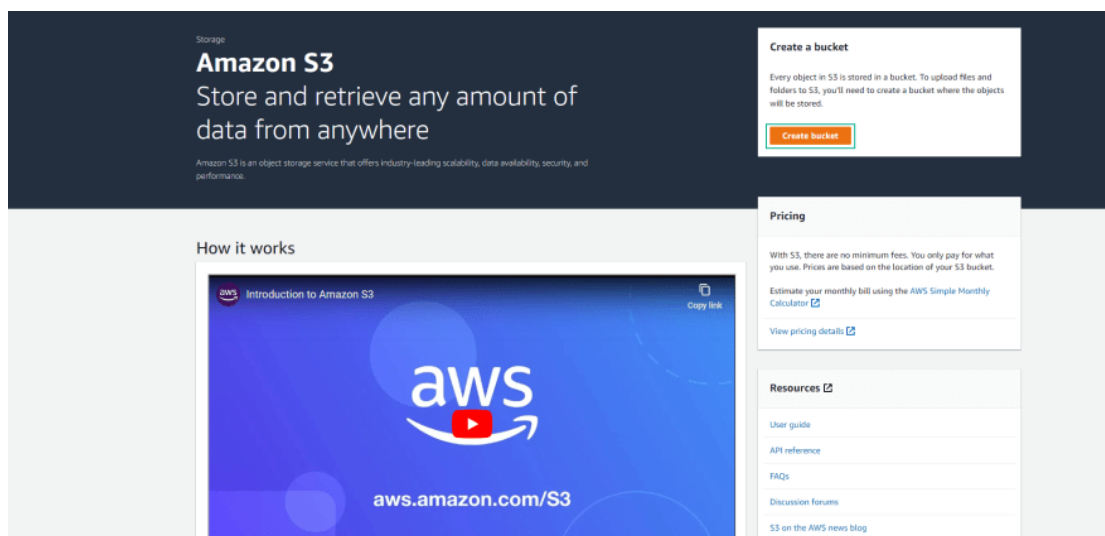
In this step, we will use Amazon S3 to store the virtual machine file that has been exported from the virtualized environment.

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.

Amazon S3 > Buckets > Create bucket

Create bucket info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be globally unique and must not contain spaces or uppercase letters. See [rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

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.**

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **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.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ **I acknowledge that the current settings might result in this bucket and the objects within becoming public.**

Select **Create bucket**.

Tags (0) - optional
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Default encryption info
Server-side encryption is automatically applied to new objects stored in this bucket.

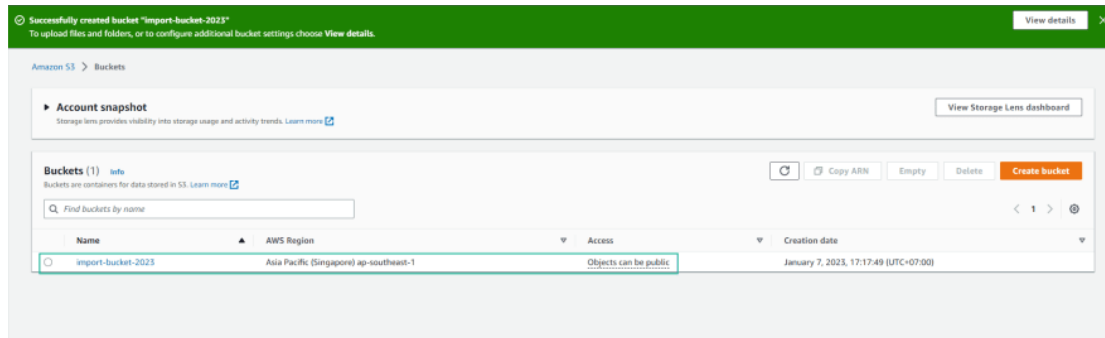
Encryption key type info
☒ Amazon S3-managed keys (SSE-S3)
☐ AWS Key Management Service key (SSE-KMS)

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. [Learn more](#)
☐ Disable
☒ Enable

► **Advanced settings**

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

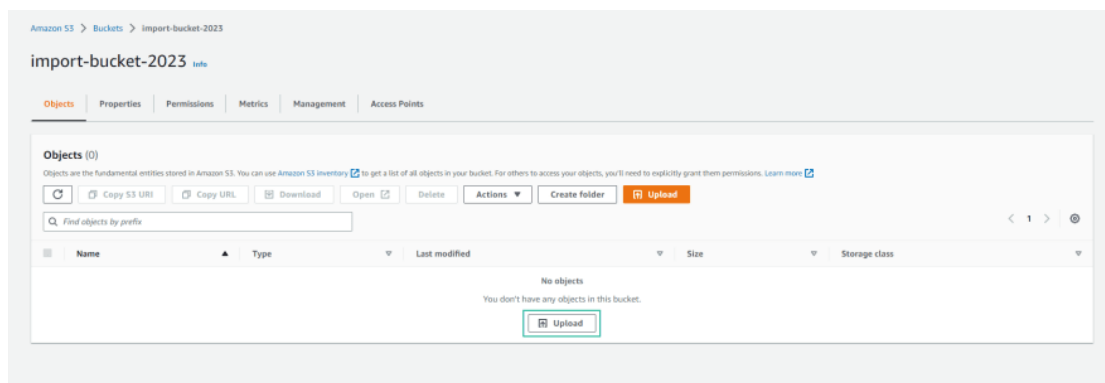
Successful bucket creation



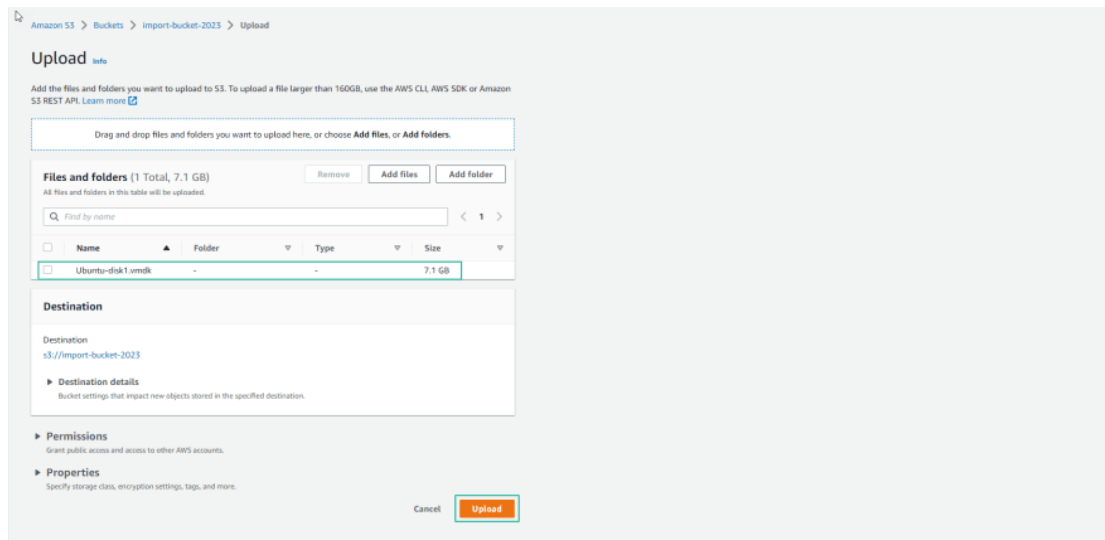
3. Upload virtual machine to S3 Bucket

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

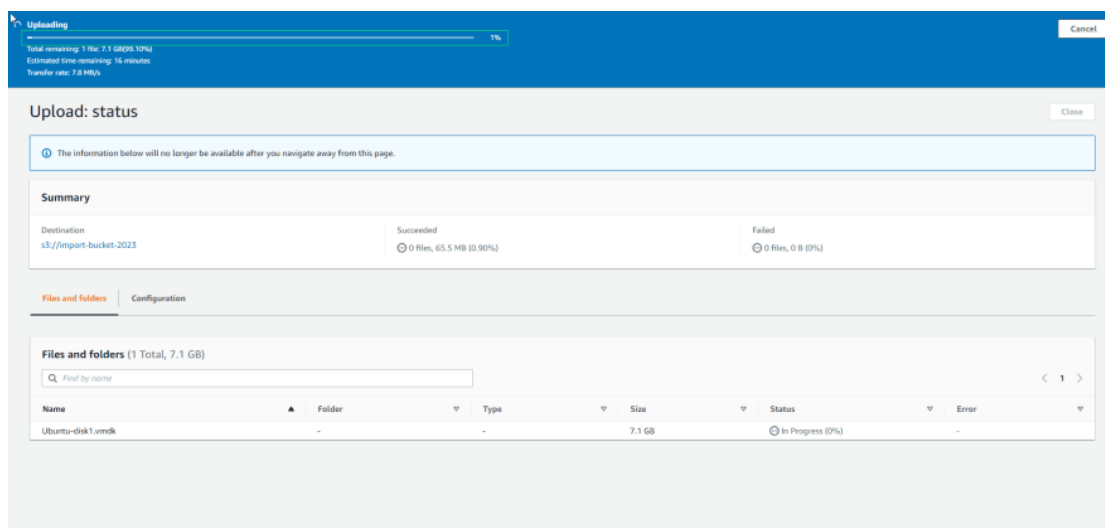
- Access to the S3 bucket you created above. (Example: import-bucket-2023)
- In the **Objects** section, select **Upload**



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.



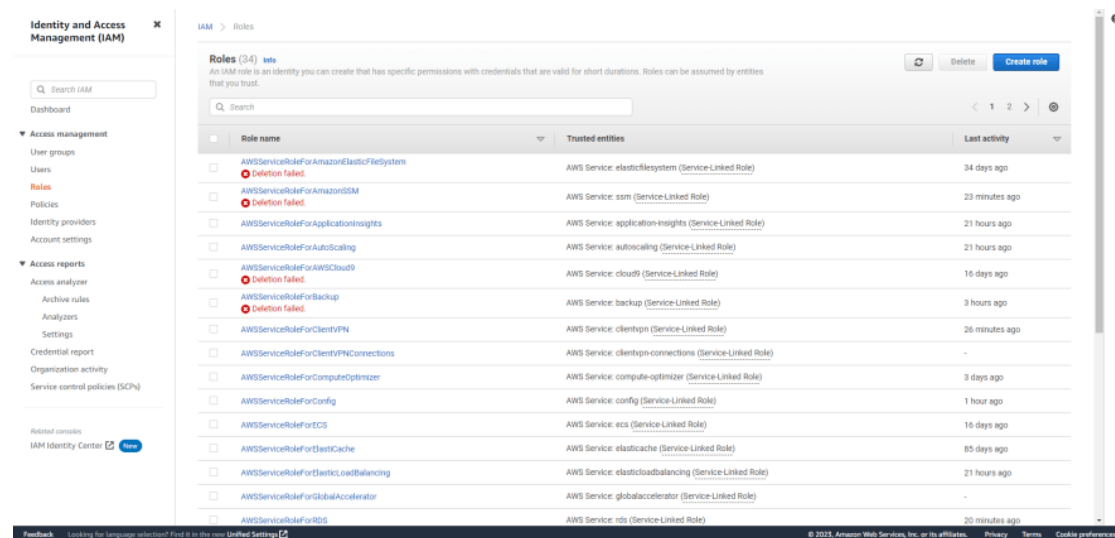
4. Import virtual machine to AWS

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



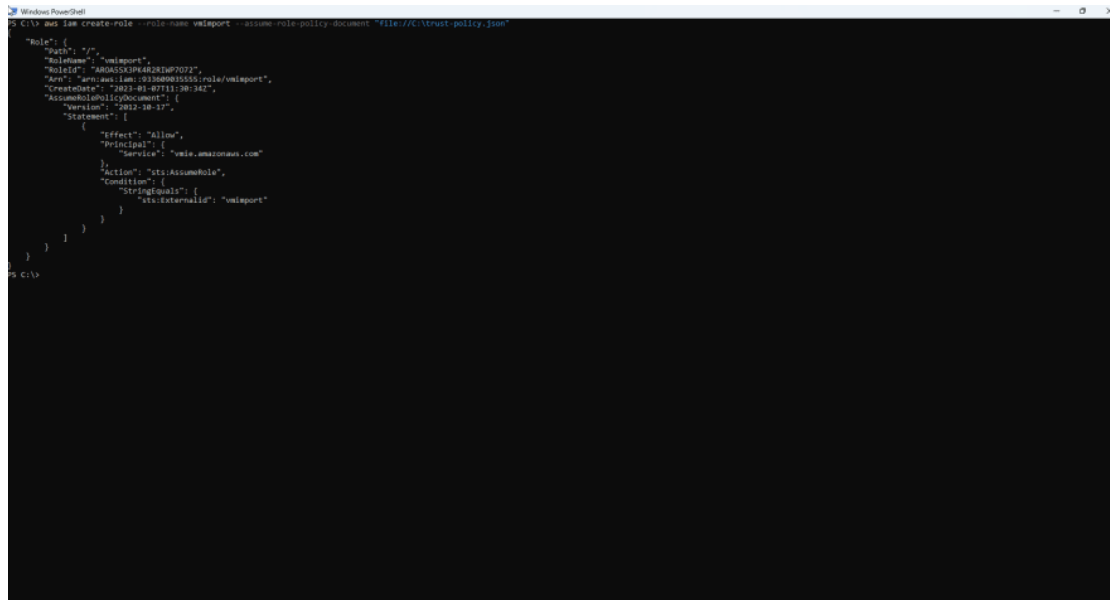
If you do not see the vmimport role, proceed to create the vmimport role.
Create a file named trust-policy.json to allow the VM Import/Export service to accept your upcoming vmimport role.

```
{
  "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

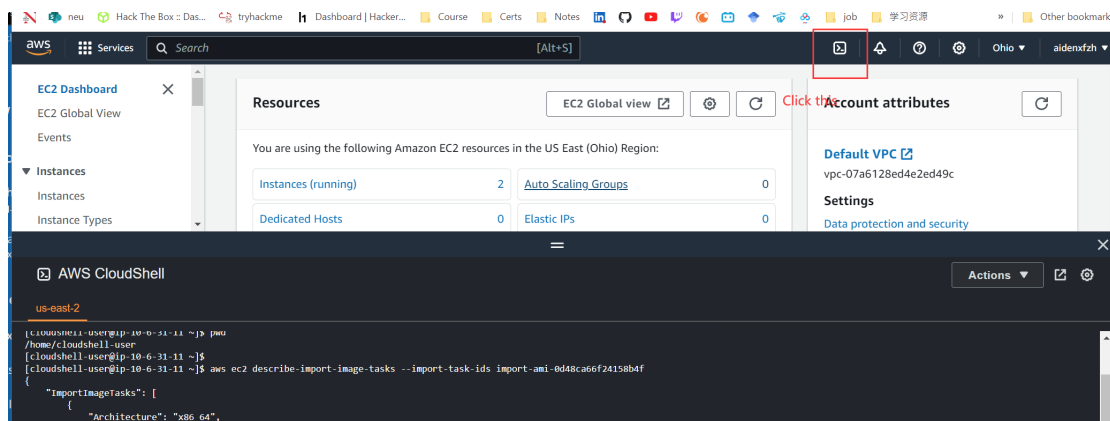
aws iam create-role --role-name vmimport --assume-role-policy-document "file://E:\trust-policy.json"



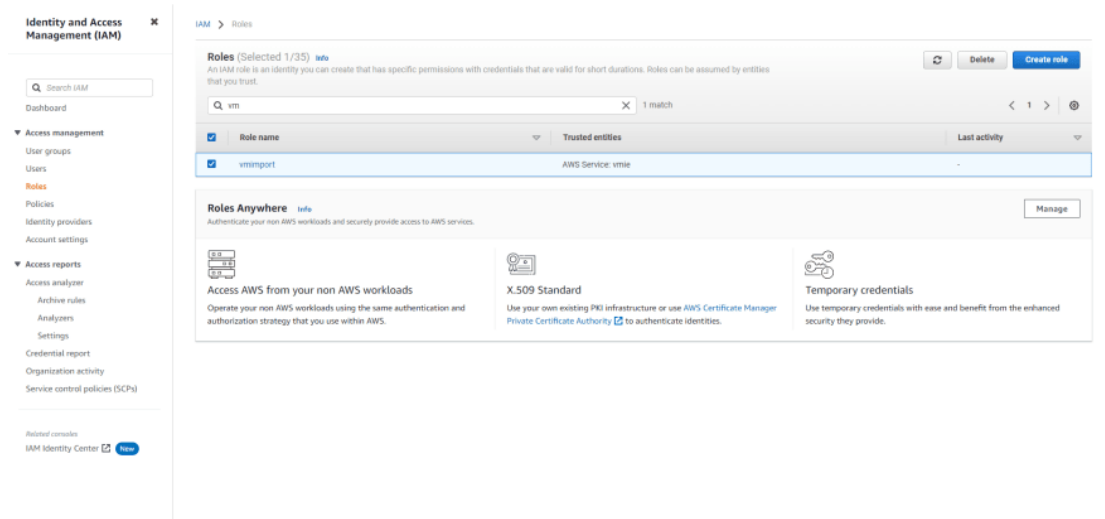
```
C:\> aws iam create-role --role-name vmimport --assume-role-policy-document "file://E:\trust-policy.json"

{
  "Role": {
    "Path": "/",
    "RoleName": "vmimport",
    "RoleId": "AROAS3PK4R2IM7072",
    "Arn": "arn:aws:iam::0108001555:role/vmimport",
    "CreateDate": "2018-07-11T18:34Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "vmie.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
            "StringEquals": {
              "sts:ExternalId": "vmimport"
            }
          }
        }
      ]
    }
  }
}
```

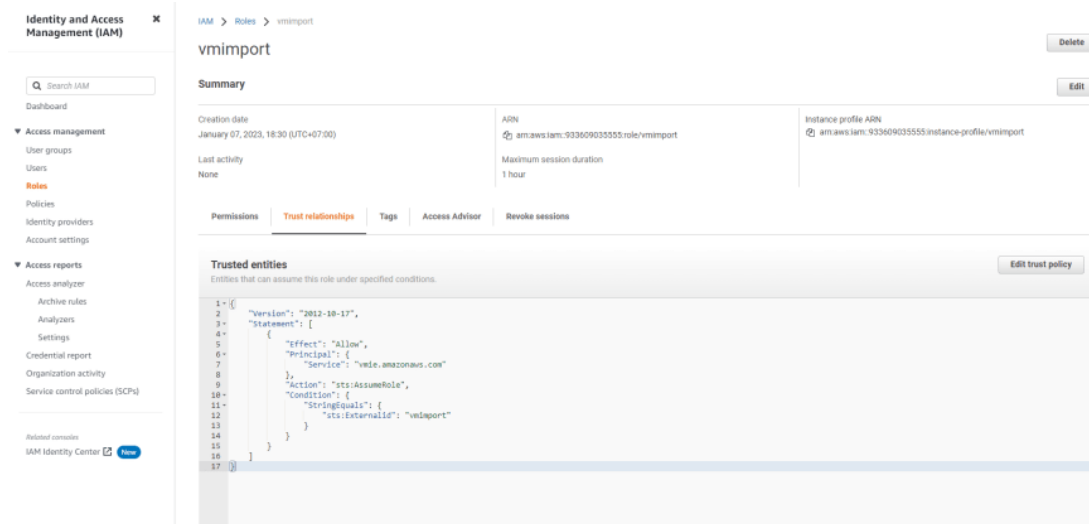
If you haven't configured aws cli on your computer, you can instead use aws cloudshell to run the commands and put your json file in it.



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

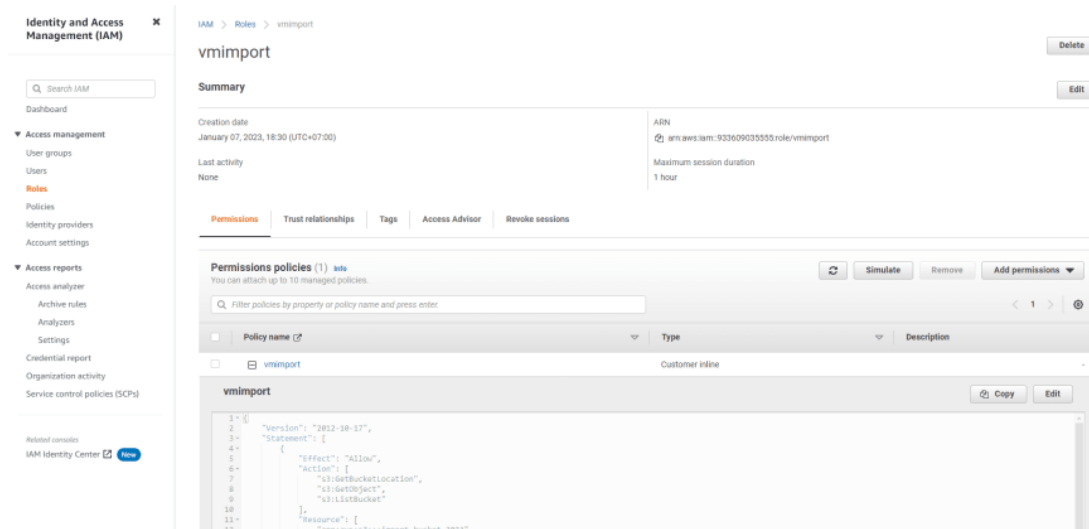
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document "file://E:\role-policy.json"

```
Windows PowerShell
C:\> aws iam create-role --role-name vmimport --assume-role-policy-document "file://C:\test-policy.json"

{
  "Role": {
    "Path": "/",
    "RoleName": "vmimport",
    "RoleId": "AROAS3PK4R2HM7D72",
    "Arn": "arn:aws:iam::93309035553:role/vmimport",
    "CreateDate": "2023-01-07T11:30:34Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
            "StringEquals": {
              "sts:ExternalId": "vmimport"
            }
          }
        }
      ]
    }
  }
}
```

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

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.



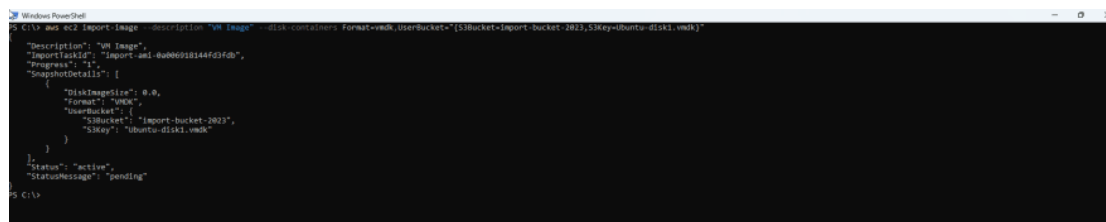
Import virtual machine to AMI

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:

- **--description:** Set description for AMI
- **--disk-containers:** 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}"
```



```
aws ec2 import-image --description "VM Image" --disk-containers Format=vhdx,UserBucket="{S3Bucket=import-bucket-2021,S3Key=Ubuntu-disk1.vmdk}"
{"Description": "VM Image",
 "ImportTaskId": "import-ami-4a060918144f2f2b",
 "Progress": 0,
 "SnapshotDetails": [
  {
    "DiskImageSize": 80,
    "Format": "VHDX",
    "UserBucket": {
      "S3Bucket": "import-bucket-2023",
      "S3Key": "Ubuntu-disk1.vmdk"
    }
  }
 ],
 "Status": "active",
 "StatusMessage": "pending"
}
```

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

```
PS C:\> aws ec2 import-image --description "VM Image" --disk-containers Format=vmx,userBucket={s3bucket-import-bucket-2023,S3Key=ubuntu-disk1.vmx}
{"Description": "VM Image",
 "ImportTaskId": "import-ami-0a000918144fd3fdb",
 "Progress": "1",
 "SnapshotDetails": [
   {
     "DiskImageSize": 0.0,
     "Format": "VMDK",
     "UserBucket": {
       "S3Bucket": "import-bucket-2023",
       "S3Key": "ubuntu-disk1.vmx"
     }
   }
 ],
 "Status": "active",
 "StatusMessage": "pending"
}
PS C:\> aws ec2 describe-import-image-tasks --import-task-id import-ami-0a000918144fd3fdb
{"ImportImageTasks": [
  {
    "Architecture": "x86_64",
    "Description": "VM Image",
    "ImportTaskId": "import-ami-0a000918144fd3fdb",
    "LicenseType": "BYOL",
    "Platform": "Linux",
    "Progress": "27",
    "SnapshotDetails": [
      {
        "DeviceName": "/dev/sda1",
        "DiskImageSize": 7670210500.0,
        "Format": "VMDK",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "import-bucket-2023",
          "S3Key": "ubuntu-disk1.vmx"
        }
      }
    ],
    "Status": "active",
    "StatusMessage": "updating",
    "Tags": [],
    "BootMode": "uefi"
  }
 ]
}
PS C:\>
```

```
PS C:\> aws ec2 describe-import-image-tasks --import-task-id import-ami-0a000918144fd3fdb
{"ImportImageTasks": [
  {
    "Architecture": "x86_64",
    "Description": "VM Image",
    "ImageId": "ami-0c958c5d878ebf050",
    "ImportTaskId": "import-ami-0a000918144fd3fdb",
    "LicenseType": "BYOL",
    "Platform": "Linux",
    "SnapshotDetails": [
      {
        "DeviceName": "/dev/sda1",
        "DiskImageSize": 7670210500.0,
        "Format": "VMDK",
        "SnapshotId": "snap-02aa87df66f62c4da",
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "import-bucket-2023",
          "S3Key": "ubuntu-disk1.vmx"
        }
      }
    ],
    "Status": "completed",
    "Tags": [],
    "BootMode": "uefi"
  }
 ]
}
PS C:\>
```

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.

Amazon Machine Images (AMIs) (1/1) Info

Owned by me Find AMI by attribute or tag

Recycle Bin

EC2 Image Builder

Actions

Launch instance from AMI

Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date
-	ami-0c958c5d878ebf050	import-ami-0a000918144fd3fdb	933609035555/import-ami-0a000918...	933609035555	Private	Available	2023/01/08 00:36 G

AMI ID: ami-0c958c5d878ebf050

Details

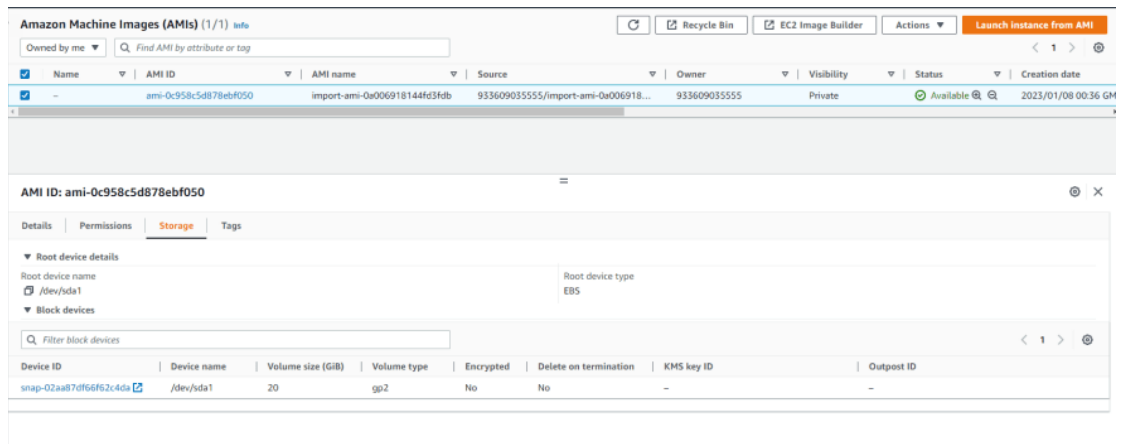
Permissions

Storage

Tags

AMI ID	ami-0c958c5d878ebf050	Image type	machine	Platform details	Linux/UNIX	Root device type	EBS
AMI name	import-ami-0a000918144fd3fdb	Owner account ID	933609035555	Architecture	x86_64	Usage operation	RunInstances
Root device name	/dev/sda1	Status	Available	Source	933609035555/import-ami-0a000918144fd3fdb	Virtualization type	hvm
Boot mode	uefi	State reason	-	Creation date	Sun Jan 08 2023 00:36:19 GMT+0700 (Indochina Time)	Kernel ID	-
Block devices	/dev/sda1::snap-02aa87df66f62c4da:20:false:gpg2	Description	AWS-VMImport service: Linux - Ubuntu 22.04.1 LTS - 5.15.0-57-generic	Product codes	-	RAM disk ID	-
Deprecation time	-	Last launched time	-				

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

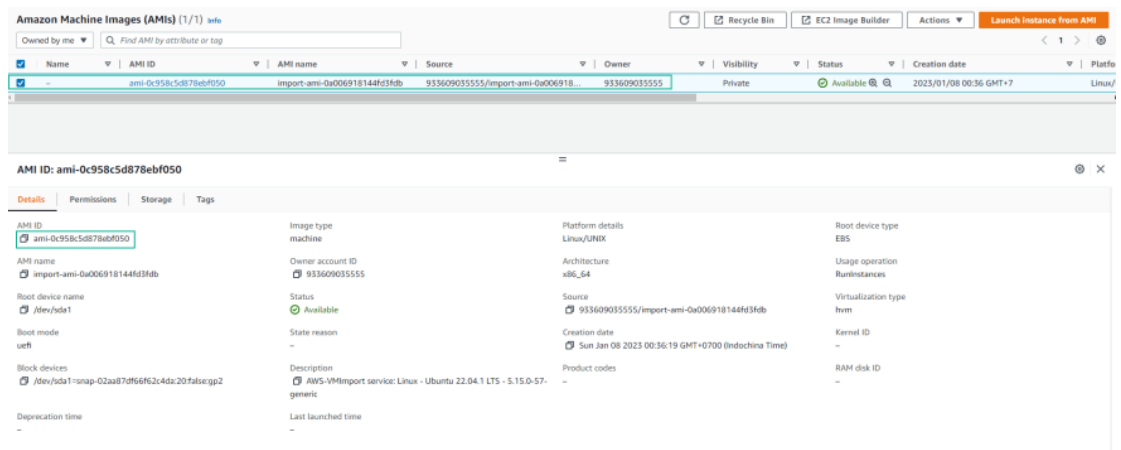


5. Deploy EC2 Instance from AMI

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**

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

Import-Server

Add additional tags

Keep the default AMI.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

AMI from catalog

Recents

My AMIs

Quick Start

Amazon Machine Image (AMI)

import-ami-0a006918144fd3fdb

ami-0c958c5d878ebf050

Published

2023-01-07T17:36:19.000Z

Architecture

x86_64

Virtualization

hvm

Root device type

ebs

ENA Enabled

Yes

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

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

▼ Instance type [Info](#)

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0132 USD per Hour
On-Demand Windows pricing: 0.0224 USD per Hour

▼

Compare instance types

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

▼

↺ Create new key pair

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

▼ Instance type [Info](#)

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0132 USD per Hour
On-Demand Windows pricing: 0.0224 USD per Hour

▼

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

▼

↺ Create new key pair

▼ Network settings [Info](#)

Network [Info](#)

vpc-0d1768dbb873807c0 | admin-vpc

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

☒ Create security group
☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

▼ Summary

Number of instances [Info](#)

1

Software image (AMI) [Info](#)

AWS-VMImport service: Linux - ...[read more](#)
ami-0c8dc5c87ba0f0902

Virtual server type (instance type) [Info](#)

t3.micro

Firewall (security group) [Info](#)

New security group

Storage (volumes) [Info](#)

1 volume(s) - 20 GiB

Cancel

Launch instance

Leave the default **Network settings**

▼ Network settings [Info](#)

Network [Info](#)

vpc-0d1768dbb873807c0 | admin-vpc

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

☒ Create security group
☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

▼ Summary

Number of instances [Info](#)

1

Software image (AMI) [Info](#)

AWS-VMImport service: Linux - ...[read more](#)
ami-0c8dc5c87ba0f0902

Virtual server type (instance type) [Info](#)

t3.micro

Firewall (security group) [Info](#)

New security group

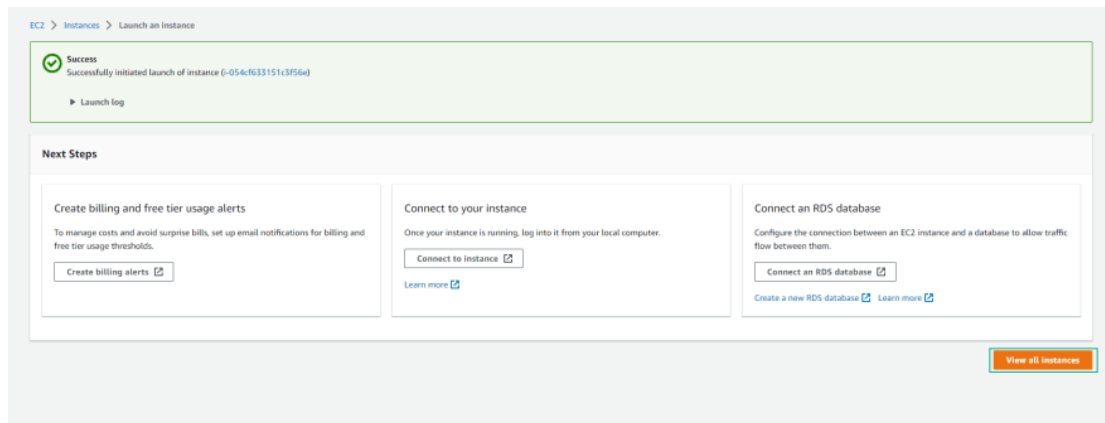
Storage (volumes) [Info](#)

1 volume(s) - 20 GiB

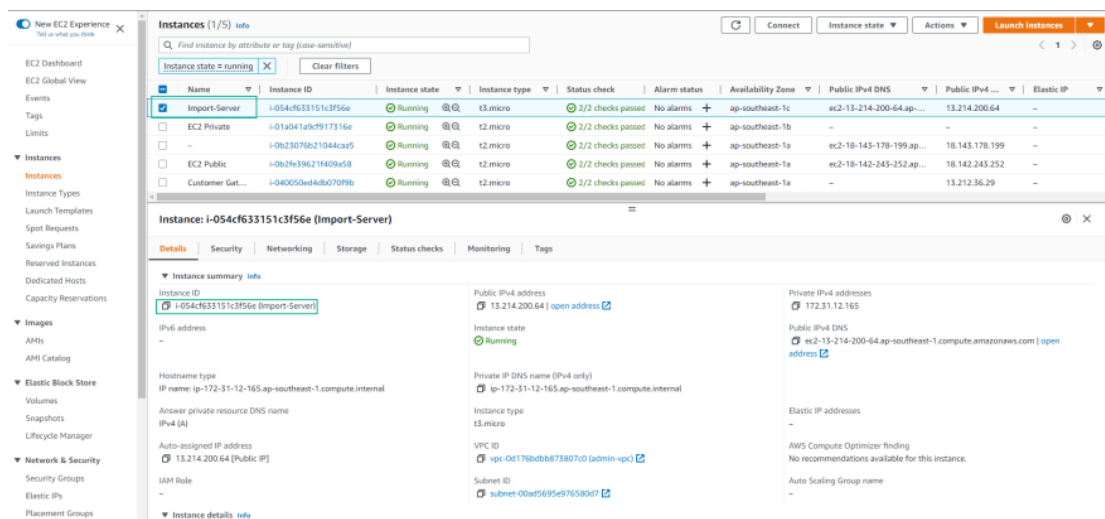
Cancel

Launch instance

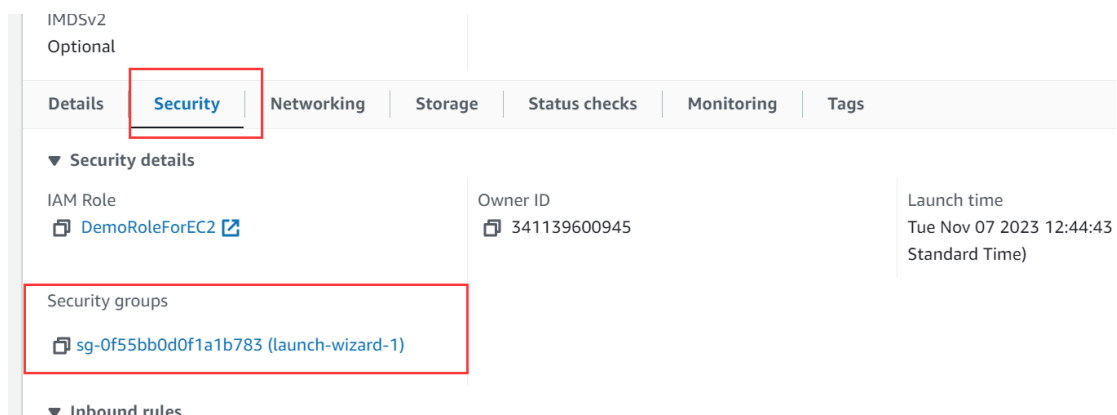
Select **View all instances**



Check the created instance.



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



Edit inbound and outbound rules to allow all tcp traffic

EC2 > Security Groups > sg-0f55bb0d0f1a1b783 - launch-wizard-1 > Edit inbound rules

Edit inbound rules [info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [info](#)

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional info	
sgr-0ecfd7e9bfeb67977	All TCP	TCP	0 - 65535	Cust... <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>		Delete
sgr-08b187515189c15fb	SSH	TCP	22	Cust... <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>		Delete
sgr-0917d4d272ee66672	HTTP	TCP	80	Cust... <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>		Delete

Add rule

EC2 > Security Groups > sg-0f55bb0d0f1a1b783 - launch-wizard-1 > Edit outbound rules

Edit outbound rules [info](#)

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules [info](#)

Security group rule ID	Type info	Protocol info	Port range info	Destination info	Description - optional info	
sgr-09511b46de924c330	All traffic	All	All	Cust... <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>		Delete

Add rule

Rules with source of 0.0.0.0/0 or ::0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Then, happy pentesting.

Reference: <https://dev.to/aws-builders/aws-importexport-part-1-import-1kpi>