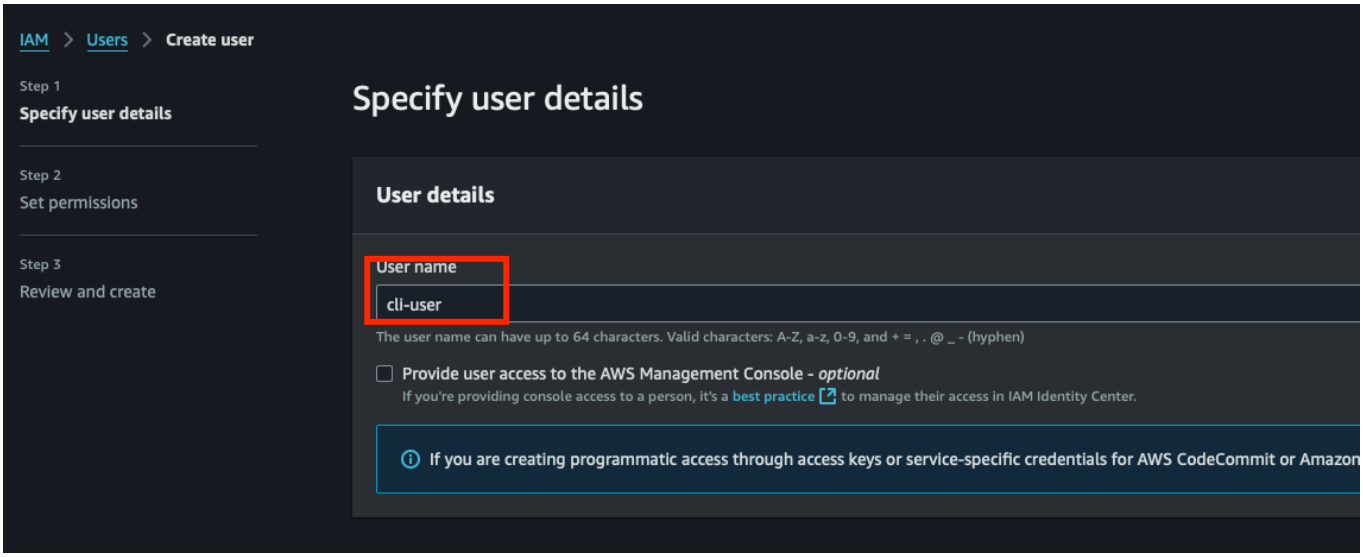
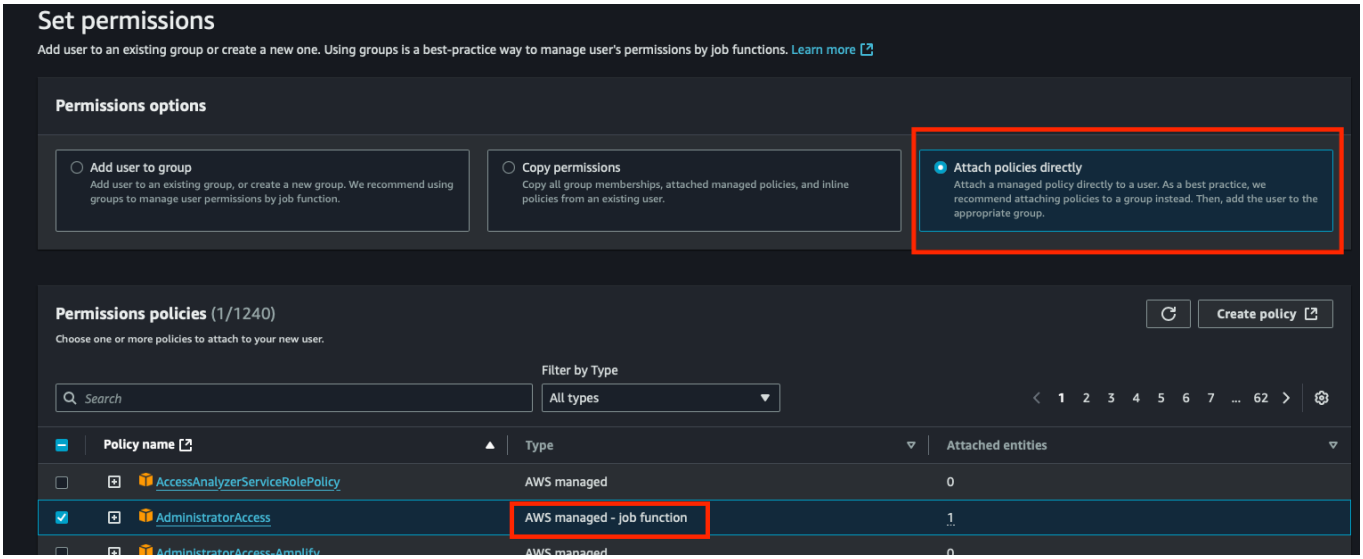
05. AWS SecOnion Lab Setup - Installing and Running Windows 11 Agent in EC2 Instance

Creating a CLI User

Navigate to the AWS IAM tab and select Users:
For cli access, first you must create a cli-user
NOTE: For security purposes, this should only be the user that's allowed to access the cli-interface and not the management gui

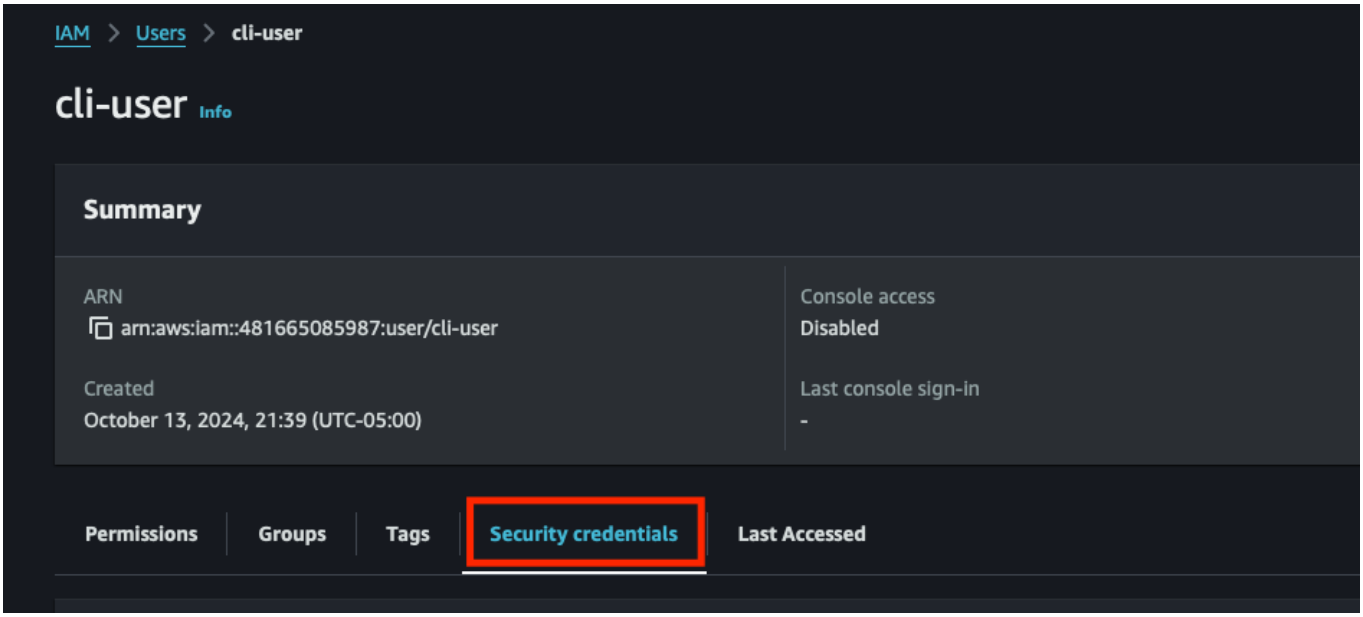


Attached the policies directly and provide admin access, for now:

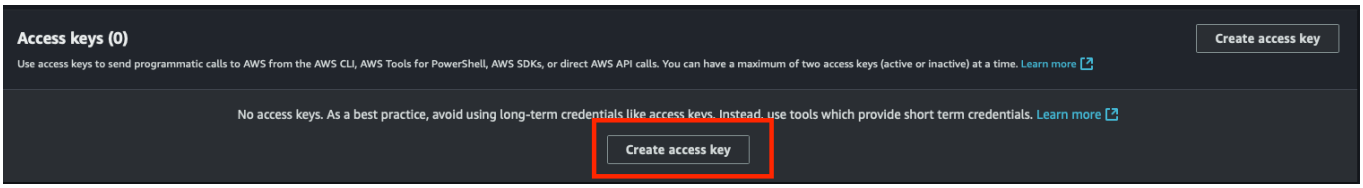


Click "Create user"

Next, click on the user you just create and click the tab in Security Credentials

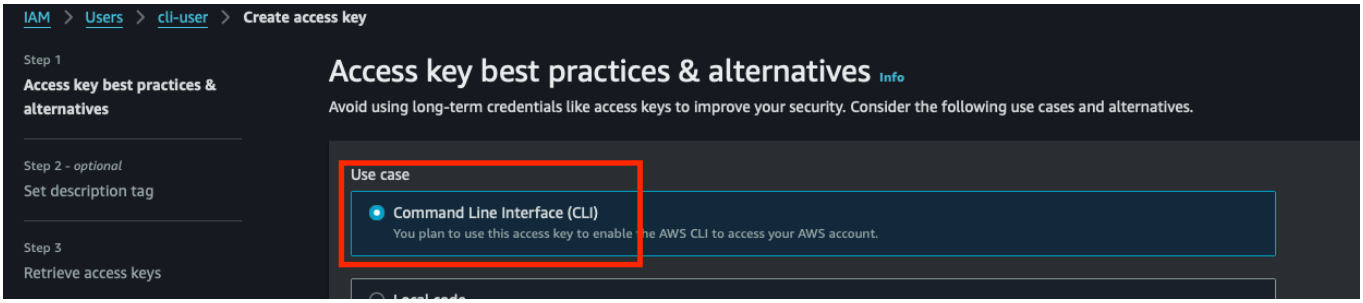


Scroll down to Create access key:

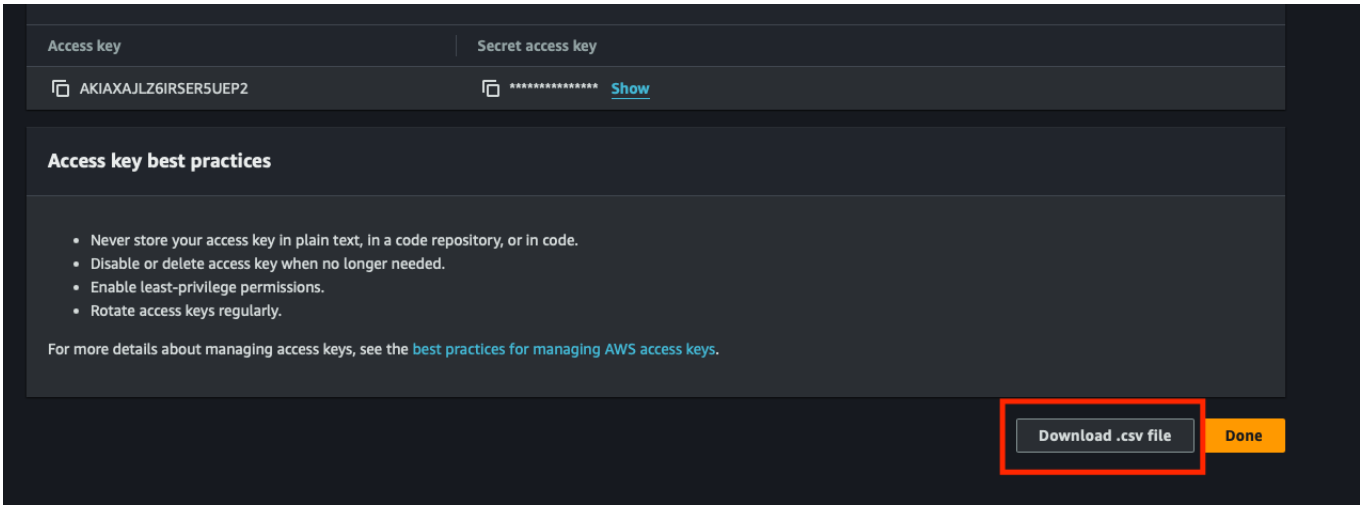


Create an access key

Select cli

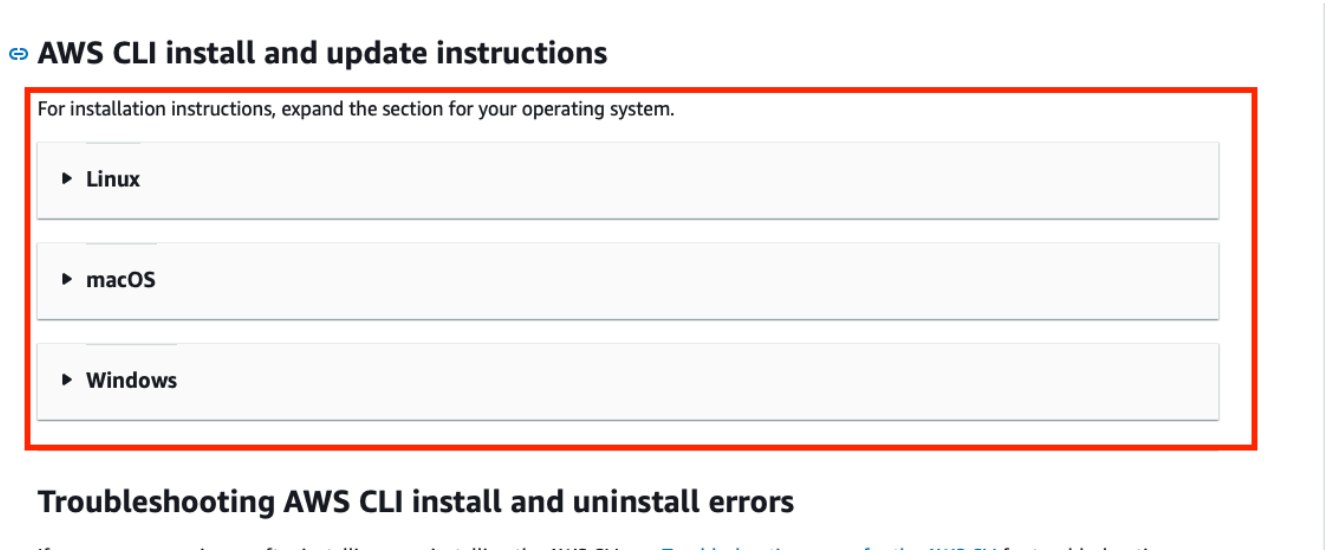


Download the csv and click done:



Install AWS CLI

Navigate to web link to download the appropriate version
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html



Select your choice, download the installer and install.
To ensure successful installation, check your aws cli version

```
aws --version
```

Configure AWS CLI

Next head to your terminal and connect to the aws console:

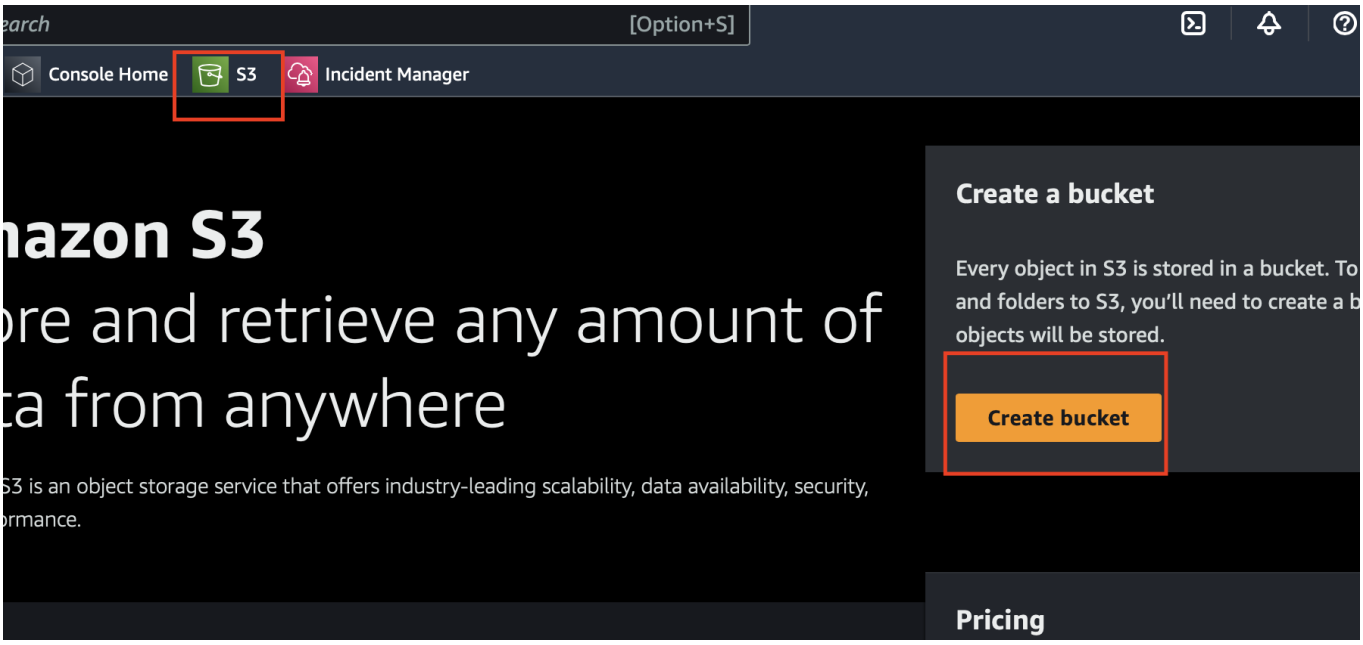
```
aws configure
```

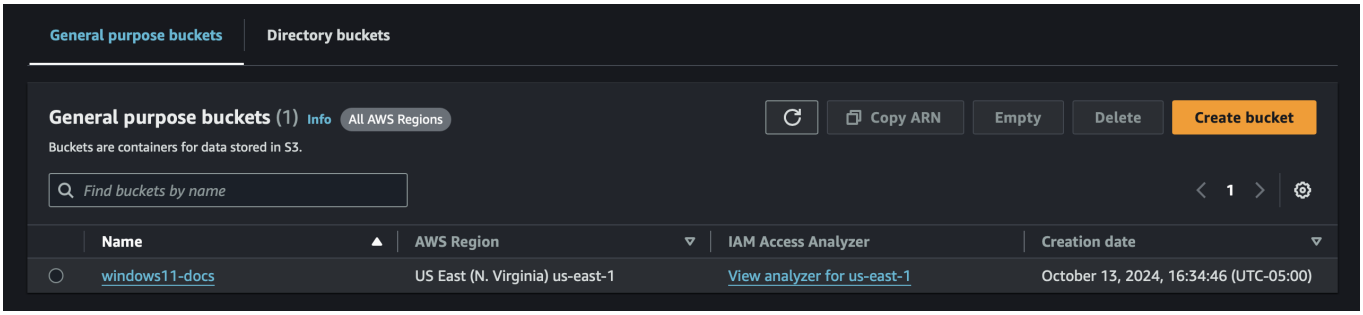
```
AWS Access Key ID [None]: AKIAXAJLZ6IRS
AWS Secret Access Key [None]: fhfx4Ydou
Default region name [None]: us-east-1
Default output format [None]: json
```

Once you enter your information the AWS is configured to run commands

Creating an s3bucket

Log into AWS and create a bucket





Create VM Import Role

Navigate to page to create the vmimport polyabdrole
https://docs.aws.amazon.com/vm-import/latest/userguide/required-permissions.html

Scroll down to the **"Required service role" section and complete instruction 1 & 2**
NOTE: This document has been completed and is in the folder in the file named "trust-policy.json" - (Do not make any edits to this document)

1. Create a file named `trust-policy.json` on your computer. Add the following policy to the file:

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

2. Use the create-role command to create a role named `vmimport` and grant VM Import/Export access to it. Ensure that you specify the full path to the location of the `trust-policy.json` file that you created in the previous step, and that you include the `file://` prefix as shown the following example:

```
aws iam create-role --role-name vmimport --assume-role-policy-document "file://C:\import\trust-
```

Navigate to the folder where the file is stored and run he following command in the cli:

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

Next, in the same section, complete step 3 & 6 (4 and 5 are optional and not needed for basic uploads):
NOTE: The role-policy.json document has been provided:
You need to edit 4 lines: 12,13,26 & 27

replace **windows11-docs** with your bucket name:

3. Create a file named `role-policy.json` with the following policy, where *`amzn-s3-demo-import-bucket`* is the bucket for imported disk images and *`amzn-s3-demo-export-bucket`* is the bucket for exported disk images:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-import-bucket",
            "arn:aws:s3:::amzn-s3-demo-import-bucket/*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
```

```
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::windows11-docs",
            "arn:aws:s3:::windows11-docs/*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:GetBucketAcl"
         ],
         "Resource": [
            "arn:aws:s3:::windows11-docs",
            "arn:aws:s3:::windows11-docs/*"
         ]
      },
      {
         "Effect": "Allow",
```

After adding your bucket, run the following command in the directory where the role-policy.json file is located:

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --
```

```
policy-document "file://role-policy.json"
```

Creating an OVA from a VM Image:

On you Virtual application, select the virtual machine and export to ova.
If you're having an issue exporting to ova ensure you read the instruction from your virtual application documents.

Import The .OVA image to the s3bucket

NOTE: This process is bandwidth dependant. The upload could take anywhere from 2- 15 hours.

GUI takes long
CLI is quicker
An option to set a faster acceleration for large downloads is optional at a cost.

There are two ways to do this> We will use the AWS CLI.

In terminal, copy the .ova file to s3bucket you set up earlier

```
aws s3 cp win-aws.ova s3://windows11-docs/awswin.ova
```

Run the AWSCommand to Create an Image

Navigate to page on import instructions:
https://docs.aws.amazon.com/vm-import/latest/userguide/import-vm-image.html

NOTE: The containers.json file has been added to this file: Only edit the following sections:
Required: S3Bucket
Required: S3Key

Optional: Description

containers.json

```
[
  {
    "Description": "My Server OVA",
    "Format": "ova",
    "UserBucket": {
        "S3Bucket": "windows11-docs",
        "S3Key": "awswin.ova"
    }
  }
]
```

Run the following command in your terminal to begin the upload process:

```
aws ec2 import-image --description "My server VM" --disk-containers
"file://containers.json"
```

```
ile://containers.json"
{
    "Description": "My server VM",
    "ImportTaskId": "import-ami-04d16fac2b6f61fa0",
    "Progress": "1",
    "SnapshotDetails": [
        {
            "Description": "My Server OVA",
            "DiskImageSize": 0.0,
            "Format": "OVA",
            "UserBucket": {
                "S3Bucket": "windows11-docs",
                "S3Key": "awswin.ova"
            }
        }
    ],
    "Status": "active",
    "StatusMessage": "pending"
}
```

Record the **ImportTaskId** to check the status of the upload:

Grab your ami import id:

```
ile://containers.json"
{
    "Description": "My server VM",
    "ImportTaskId": "import-ami-04d16fac2b6f61fa0",
    "Progress": "1",
    "SnapshotDetails": [
        {
            "Description": "My Server OVA",
            "DiskImageSize": 0.0,
            "Format": "OVA",
            "UserBucket": {
                "S3Bucket": "windows11-docs",
                "S3Key": "awswin.ova"
            }
        }
    ],
    "Status": "active",
    "StatusMessage": "pending"
}
```

Run the command to check the status:

NOTE: Check the status occasionally to ensure no issues persist.

```
aws ec2 describe-import-image-tasks --import-task-ids import-ami-
04d16fac2b6f61fa0
```

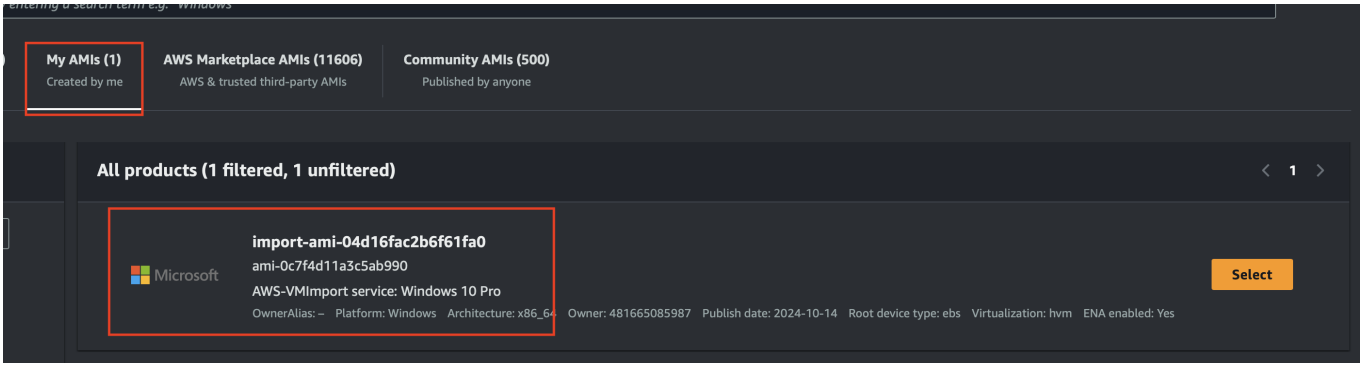
Notice the status:

```
                    "DiskImageSize": 29293367296.0,
                    "Format": "VMDK",
                    "Status": "active",
                    "UserBucket": {
                        "S3Bucket": "windows11-docs",
                        "S3Key": "awswin.ova"
                    }
                }
            ],
            "Status": "active",
            "StatusMessage": "converting",
            "Tags": []
        }
    ]
}
```

After a time you will see the status is complete:

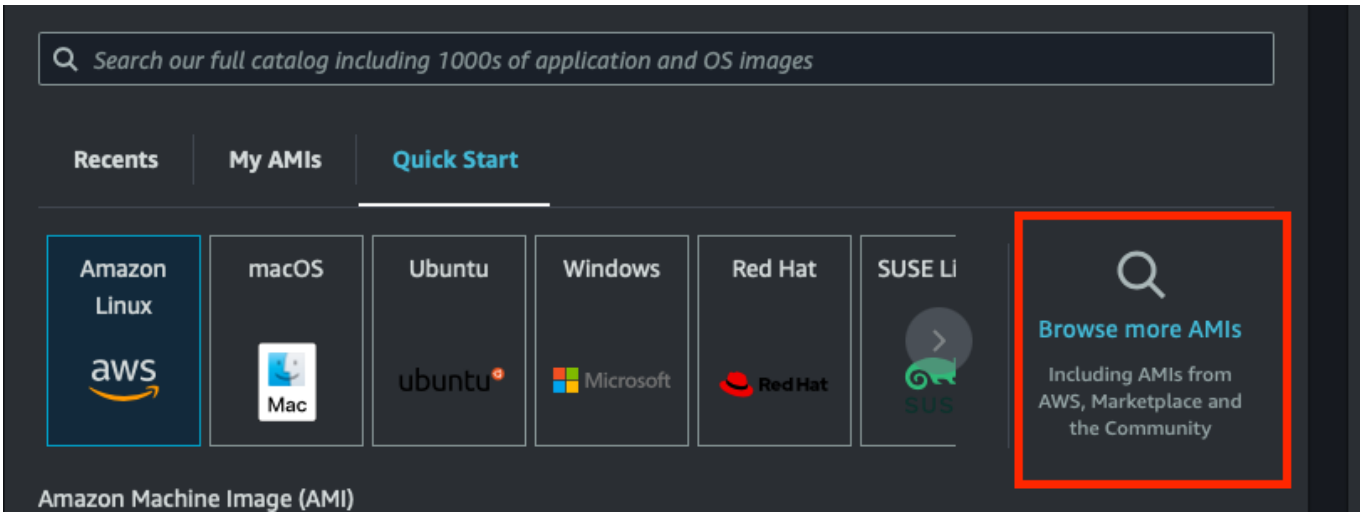

```
{
    "ImportImageTasks": [
        {
            "Architecture": "x86_64",
            "Description": "My server VM",
            "ImageId": "ami-0c7f4d11a3c5ab990",
            "ImportTaskId": "import-ami-04d16fac2b6f61fa0",
            "LicenseType": "BYOL",
            "Platform": "Windows",
            "SnapshotDetails": [
                {
                    "DeviceName": "/dev/sda1",
                    "DiskImageSize": 29293367296.0,
                    "Format": "VMDK",
                    "SnapshotId": "snap-078c6269e027134bc",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "windows11-docs",
                        "S3Key": "awswin.ova"
                    }
                }
            ],
            "Status": "completed",
            "Tags": []
        }
    ]
}
```


Proceed to EC2 > Launch instances to see if your AMI is properly uploaded:

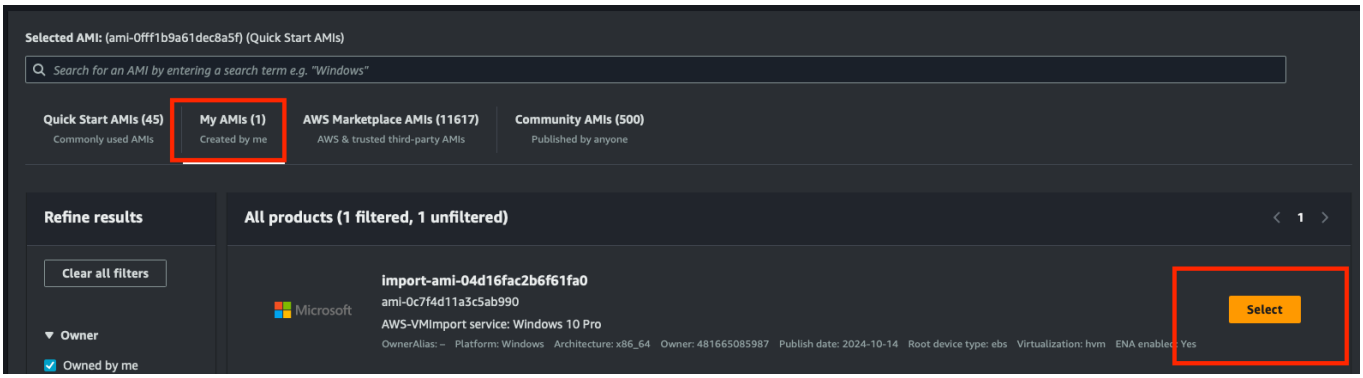


Launch The Uploaded AMI Instance

Head to launch instance and Browse more AMIs.



Click on My AMIs tab and select the AMI:



Name your new image:

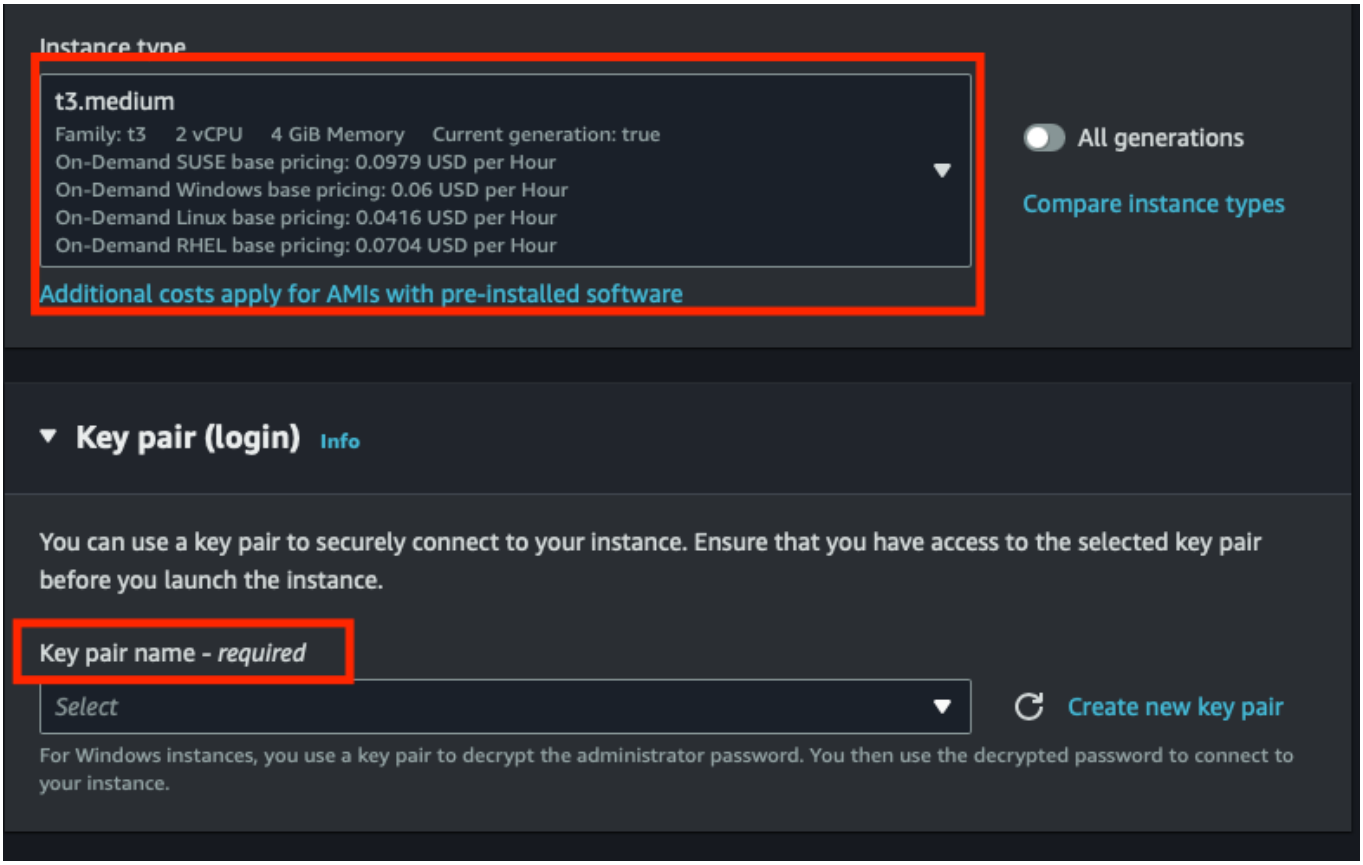
Select the appropriate storage:

WIndows requires a minimum (2vCPU 4GiB Memory)

Storage Selection

t2.medium or a t3.medium

Select or create a key pair:



Netowrking:

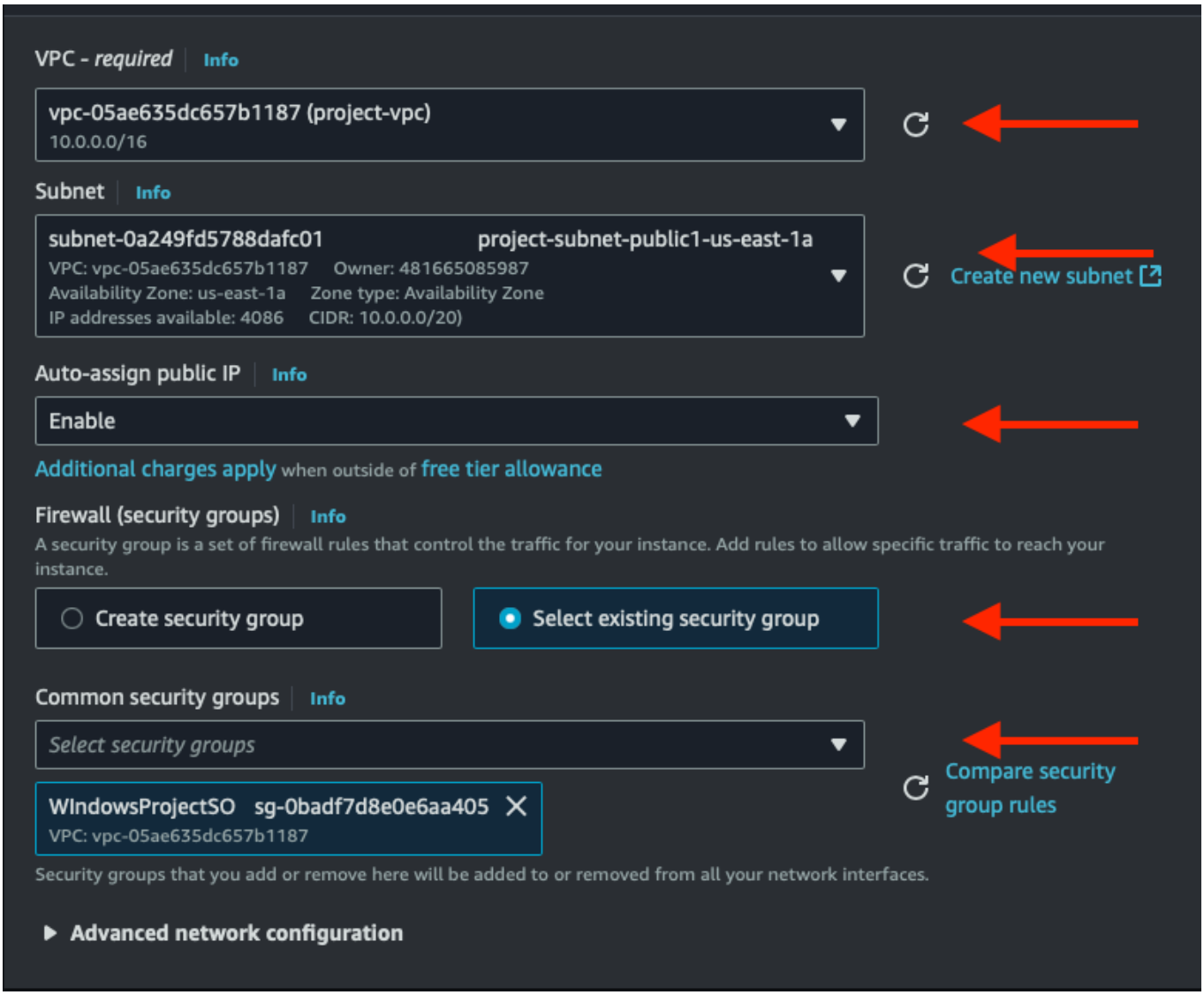
Enusre you adjust the following:

VPC: Same as the SOC Lab

Subnet: Same as the SOC Lab (Public1)

Auto-assign public IP: **Enabled**

Security Groups: Set Appropriate or Create New: Ensure rule added to allow traffic across VPC:



Launch the instance and wait: