

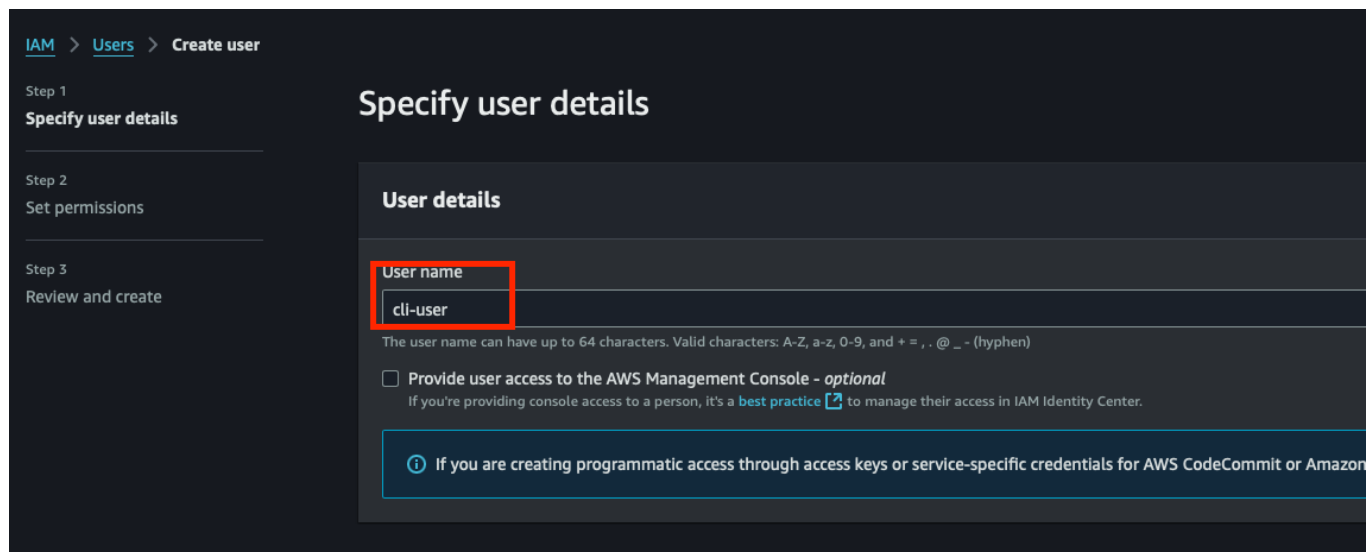
# 05. AWS SecOnion Lab Setup - Installing and Running Windows 11 Agent in EC2 Instance

## Creating a CLI User

Navigate to the AWS IAM tab and select Users:

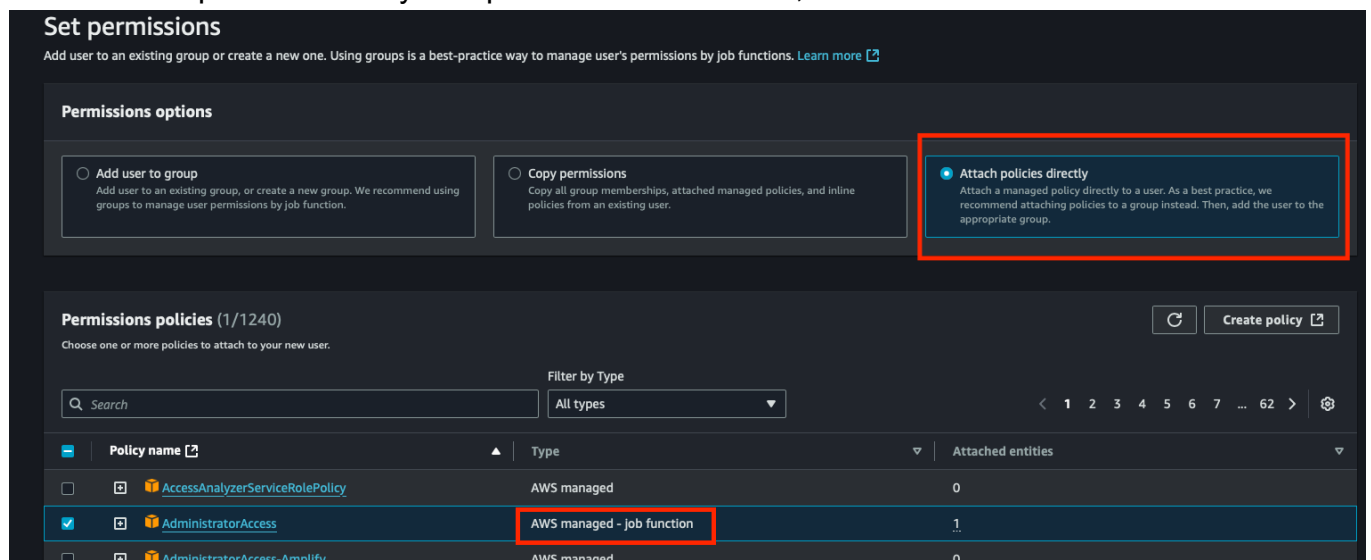
For cli access, first you must create a cli-user

NOTE: For security purposes, this should only be the user that's allowed to access the cli-interface and not the management gui



The screenshot shows the AWS IAM console's 'Create user' wizard. The 'Specify user details' step is active. The 'User name' field is highlighted with a red box and contains the text 'cli-user'. Below the field, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)'. There is an unchecked checkbox for 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.' At the bottom, a blue box contains an information icon and the text: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon'.

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

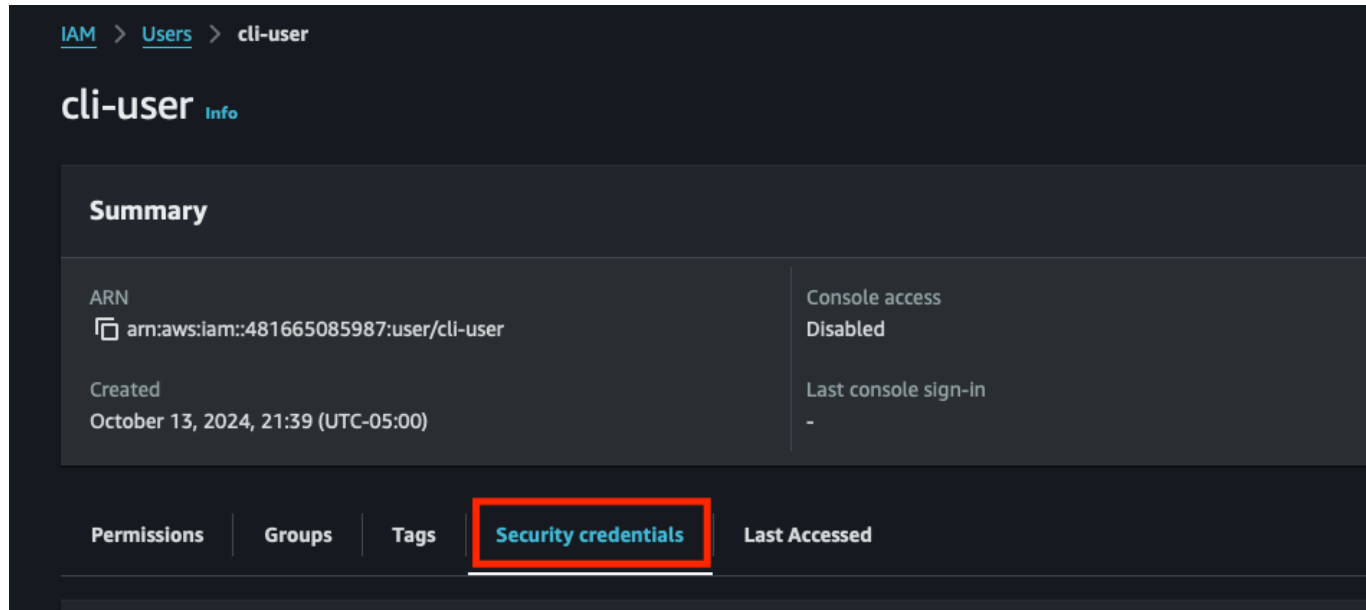


The screenshot shows the 'Set permissions' step in the AWS IAM console. The 'Permissions options' section has three radio buttons: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected and highlighted with a red box. Below this, the 'Permissions policies (1/1240)' section is shown. It includes a search bar, a 'Filter by Type' dropdown set to 'All types', and a table of policies. The table has columns for 'Policy name', 'Type', and 'Attached entities'. The 'AdministratorAccess' policy is selected with a checkbox and highlighted with a blue background. Its 'Type' is 'AWS managed - job function', which is highlighted with a red box. The 'Attached entities' column shows '1' for this policy.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	1
<input type="checkbox"/> AdministratorAccess-Amplicy	AWS managed	0

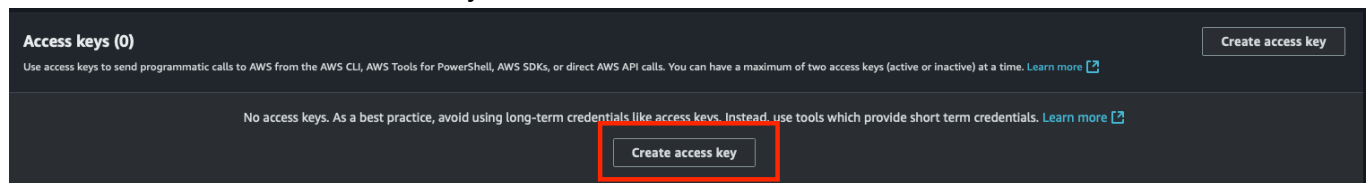
Click "Create user"

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



The screenshot shows the AWS IAM console interface for a user named 'cli-user'. The breadcrumb navigation at the top reads 'IAM > Users > cli-user'. Below the user name, there is a 'Summary' section with details: ARN 'arn:aws:iam::481665085987:user/cli-user', Console access 'Disabled', and Created date 'October 13, 2024, 21:39 (UTC-05:00)'. At the bottom, a horizontal tab bar contains 'Permissions', 'Groups', 'Tags', 'Security credentials' (which is highlighted with a red box), and 'Last Accessed'.

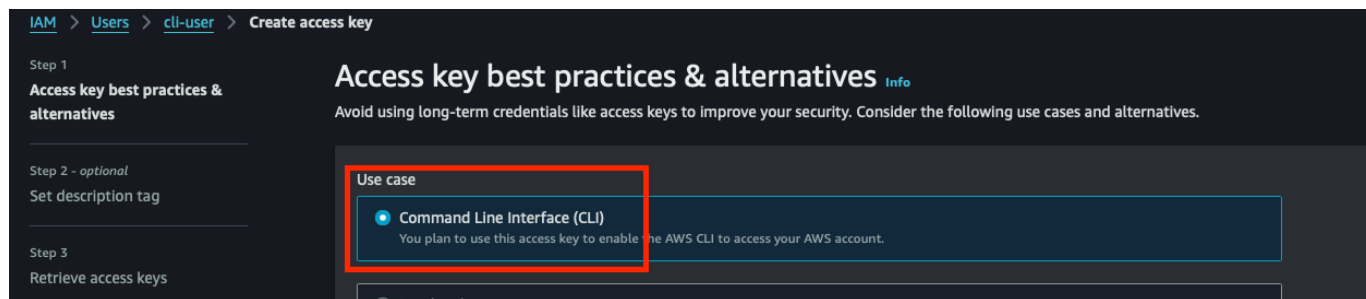
Scroll down to Create access key:



The screenshot shows the 'Create access key' page. At the top, it says 'Access keys (0)' and 'Create access key'. Below this, a message states: 'No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more'. At the bottom, the 'Create access key' button is highlighted with a red box.

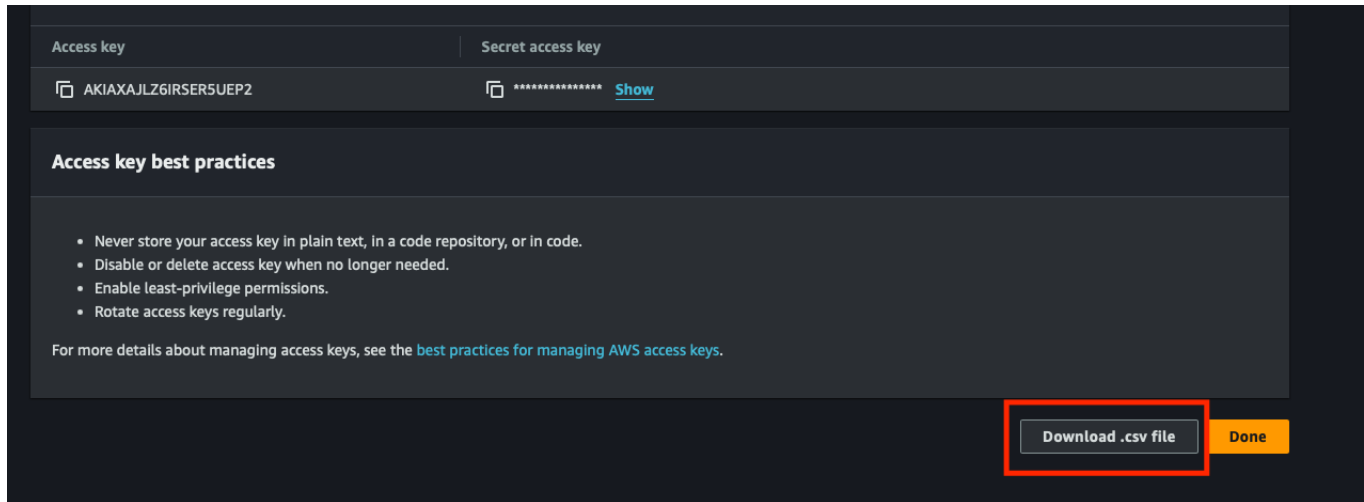
Create an access key

Select cli



The screenshot shows the 'Create access key' wizard. The breadcrumb navigation is 'IAM > Users > cli-user > Create access key'. On the left, there are steps: 'Step 1: Access key best practices & alternatives', 'Step 2 - optional: Set description tag', and 'Step 3: Retrieve access keys'. The main content area is titled 'Access key best practices & alternatives' and includes a warning about long-term credentials. Below this, a 'Use case' section is highlighted with a red box, showing 'Command Line Interface (CLI)' as the selected option with the description: 'You plan to use this access key to enable the AWS CLI to access your AWS account.'

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>

### AWS CLI install and update instructions

For installation instructions, expand the section for your operating system.

▸ Linux

▸ macOS

▸ Windows

### Troubleshooting AWS CLI install and uninstall errors

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:

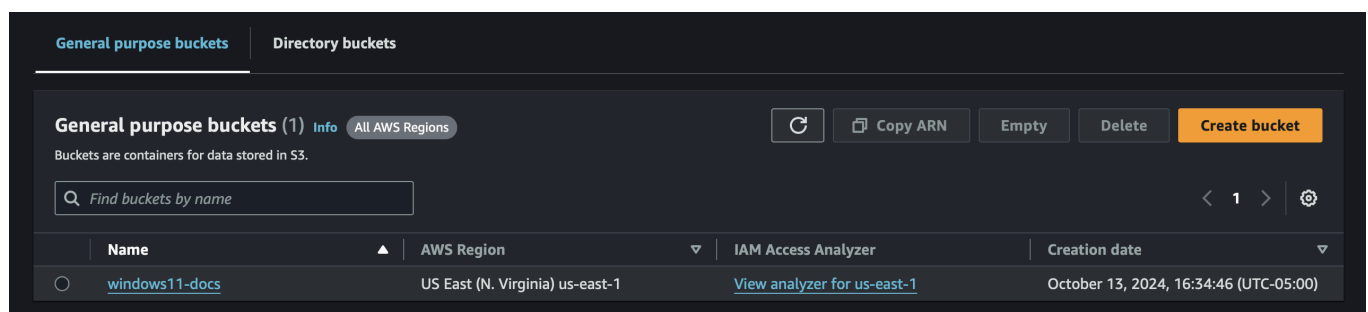
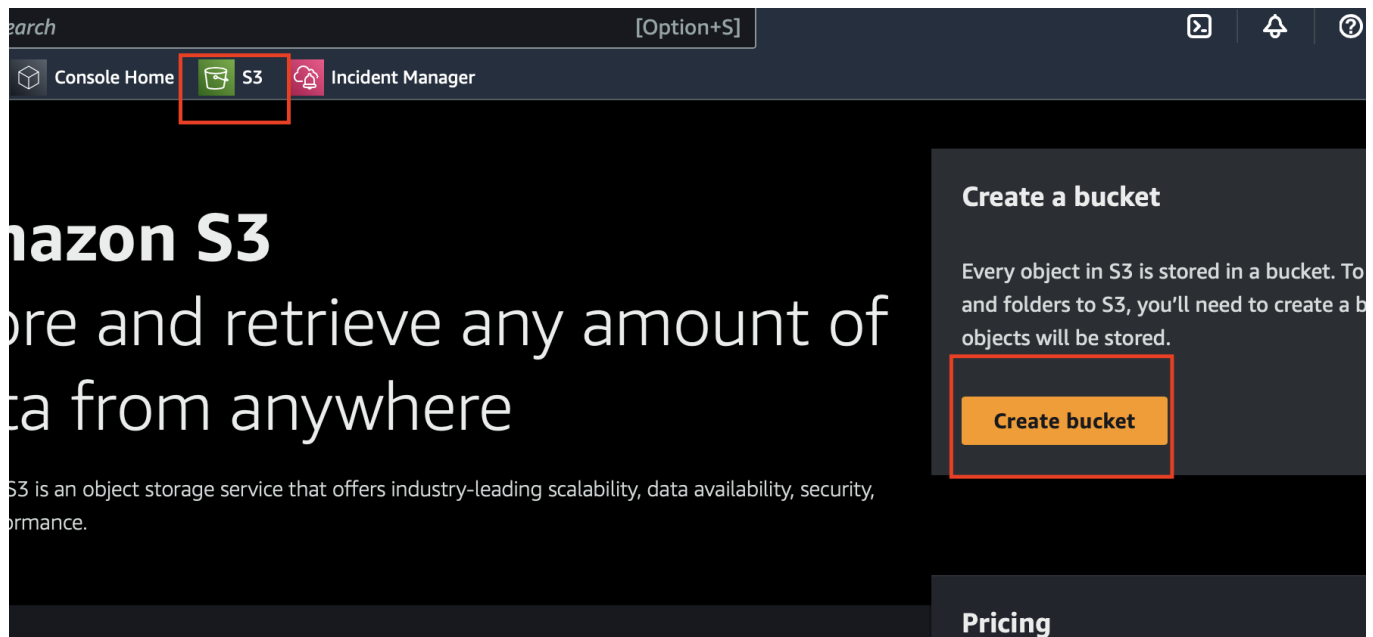
```
aws configure
```

```
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 polyabdrrole

<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)

To create the service role

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-policy.json"
```



Navigate to the folder where the file is stored and run the 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:GetBucketLocation",
```

```
8      "s3:GetObject",
9      "s3:ListBucket"
10    ],
11    "Resource": [
12      "arn:aws:s3:::windows11-docs",
13      "arn:aws:s3:::windows11-docs/*"
14    ]
15  },
16  {
17    "Effect": "Allow",
18    "Action": [
19      "s3:GetBucketLocation",
20      "s3:GetObject",
21      "s3:ListBucket",
22      "s3:PutObject",
23      "s3:GetBucketAcl"
24    ],
25    "Resource": [
26      "arn:aws:s3:::windows11-docs",
27      "arn:aws:s3:::windows11-docs/*"
28    ]
29  },
30  {
31    "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 x
1  [
2    {
3      "Description": "My Server OVA",
4      "Format": "ova",
5      "UserBucket": {
6        "S3Bucket": "windows11-docs",
7        "S3Key": "awswin.ova"
8      }
9    }
10 ]
```

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"
```

```
file://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:

```
file://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
```

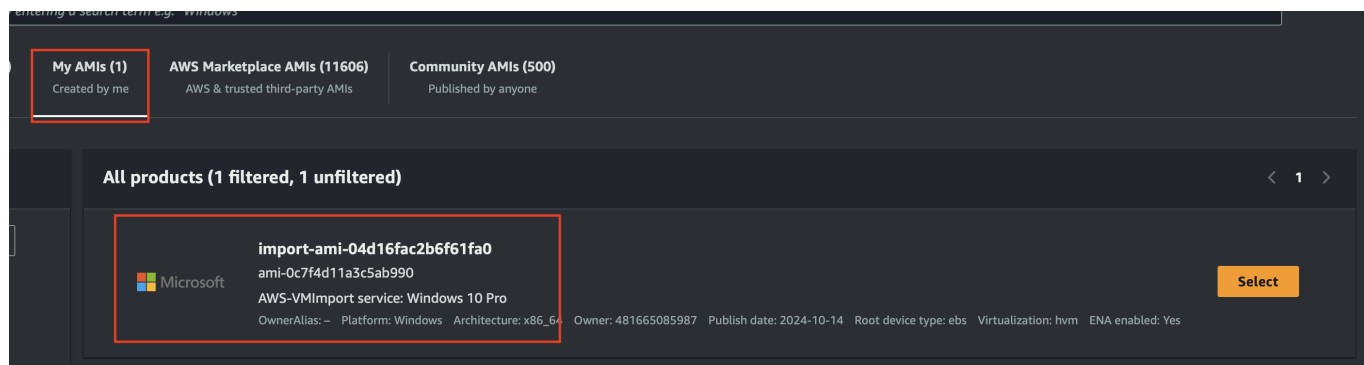
Notice the status:

```
      "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:

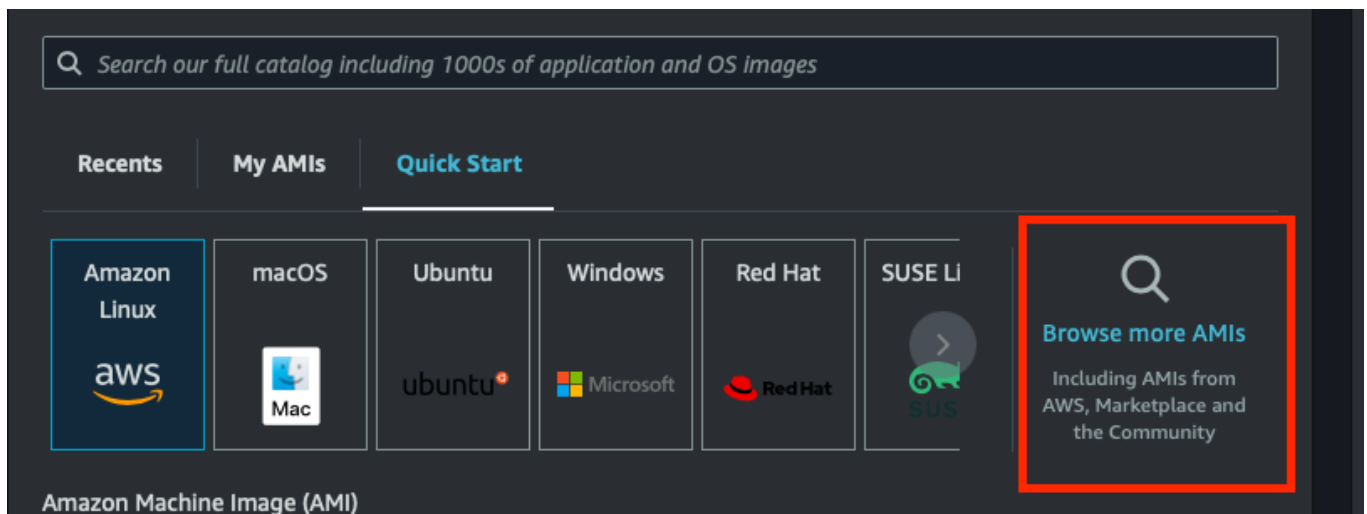
```
"ImportImageTasks": [
  {
    "Architecture": "x86_64",
    "Description": "My server VM",
    "ImageId": "ami-0c7f4d11a3c5ab990",
    "ImportTaskId": "import-ami-04d16fac2b6f61fa0",
    "LicenseType": "BYOL",
    "Platform": "Windows",
    "SnapshotDetails": [
      {
        "DeviceName": "/dev/sda1",
        "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:

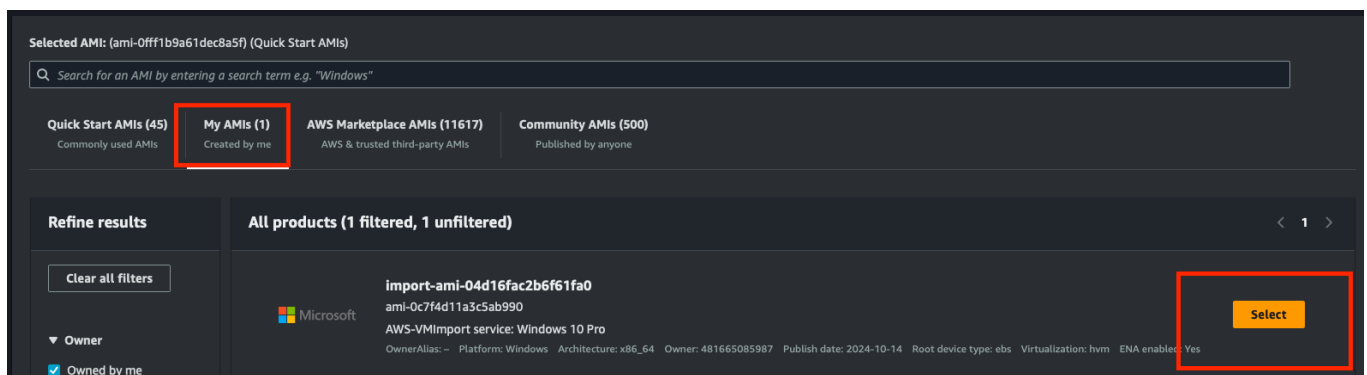


## Launch The Uploaded AMI Instance

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:

**Instance type**

**t3.medium**

Family: t3 2 vCPU 4 GiB Memory Current generation: true

On-Demand SUSE base pricing: 0.0979 USD per Hour

On-Demand Windows base pricing: 0.06 USD per Hour

On-Demand Linux base pricing: 0.0416 USD per Hour

On-Demand RHEL base pricing: 0.0704 USD per Hour

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

---

**▼ 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](#)

For Windows instances, you use a key pair to decrypt the administrator password. You then use the decrypted password to connect to your instance.

Networking:

Ensure 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:

**VPC - required** | [Info](#)

vpc-05ae635dc657b1187 (project-vpc)  
10.0.0.0/16

↻

←

**Subnet** | [Info](#)

subnet-0a249fd5788dafc01      project-subnet-public1-us-east-1a  
VPC: vpc-05ae635dc657b1187    Owner: 481665085987  
Availability Zone: us-east-1a    Zone type: Availability Zone  
IP addresses available: 4086    CIDR: 10.0.0.0/20

↻

← [Create new subnet](#)

**Auto-assign public IP** | [Info](#)

Enable

↻

←

**Additional charges apply** when outside of **free tier allowance**

**Firewall (security groups)** | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

←

**Common security groups** | [Info](#)

Select security groups

↻

← [Compare security group rules](#)

WindowsProjectSO sg-0badf7d8e0e6aa405 ✕  
VPC: vpc-05ae635dc657b1187

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► **Advanced network configuration**

Launch the instance and wait: