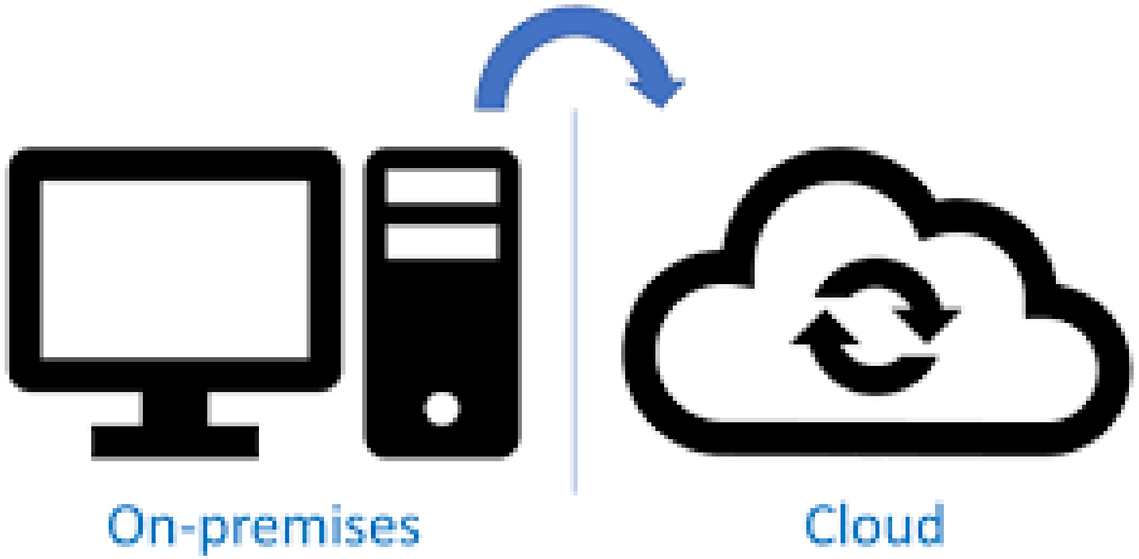
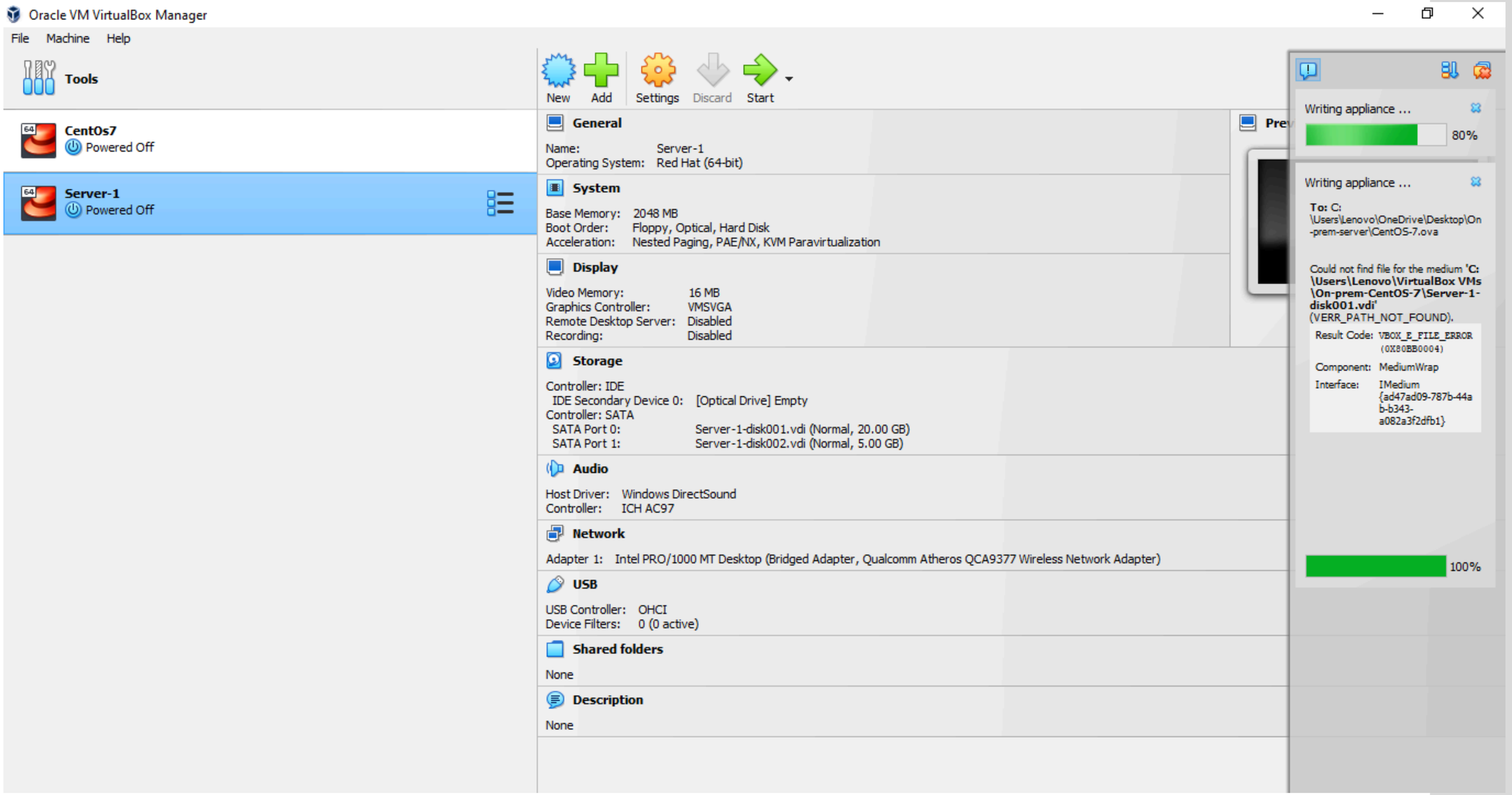
SERVER MIGRATION

REPORT

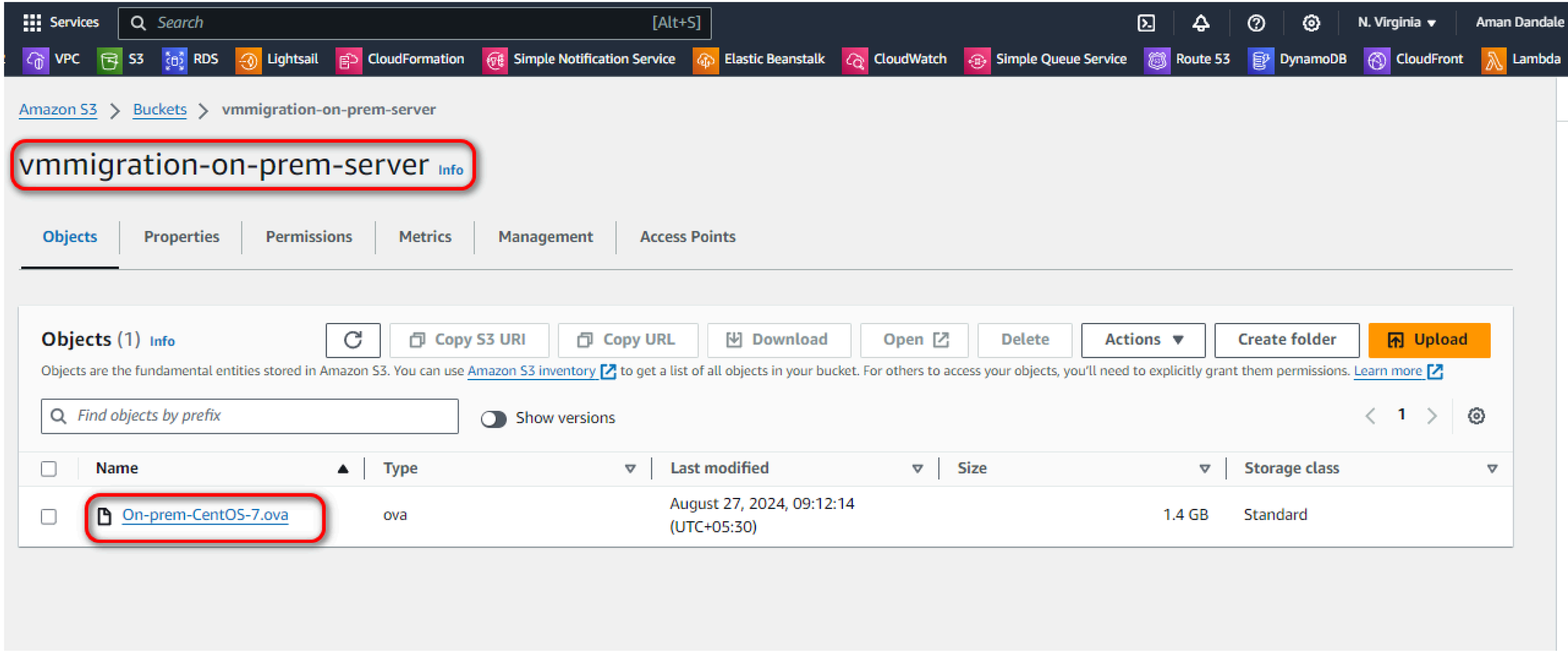
FROM ON-PREM TO AWS CLOUD



Step-1 : Extract On-Prem server

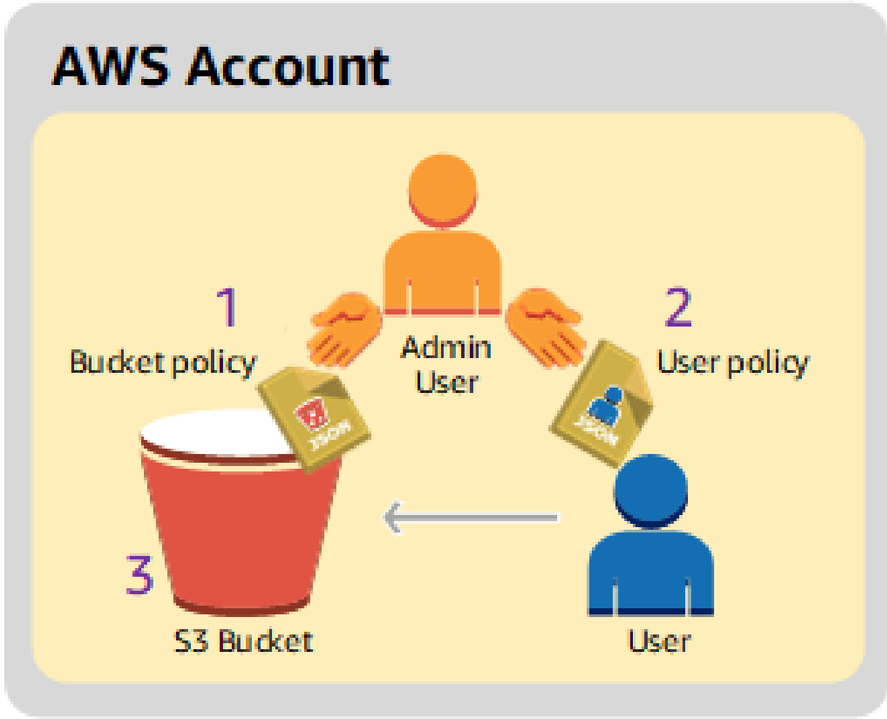


Step-2 : Create s3 bucket with public access and upload the on-prem server in bucket



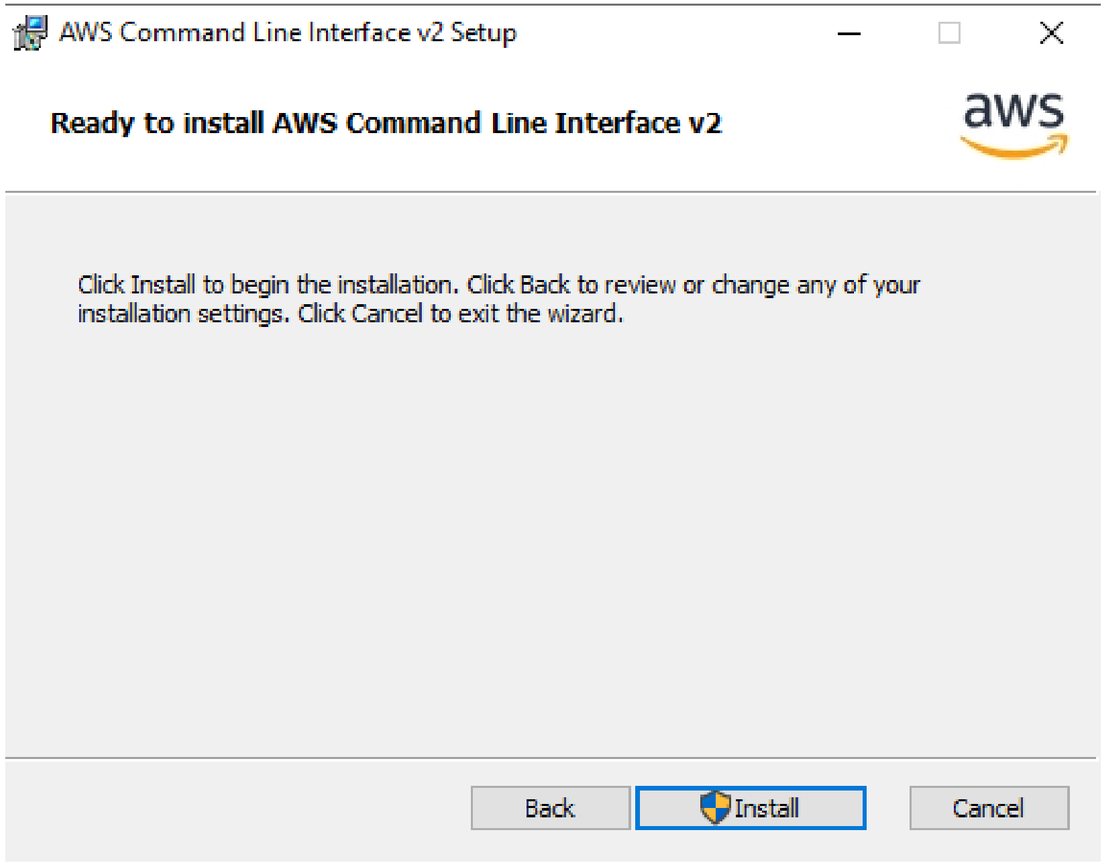
Step-3 : Attached s3 bucket policy to s3 bucket for public access.

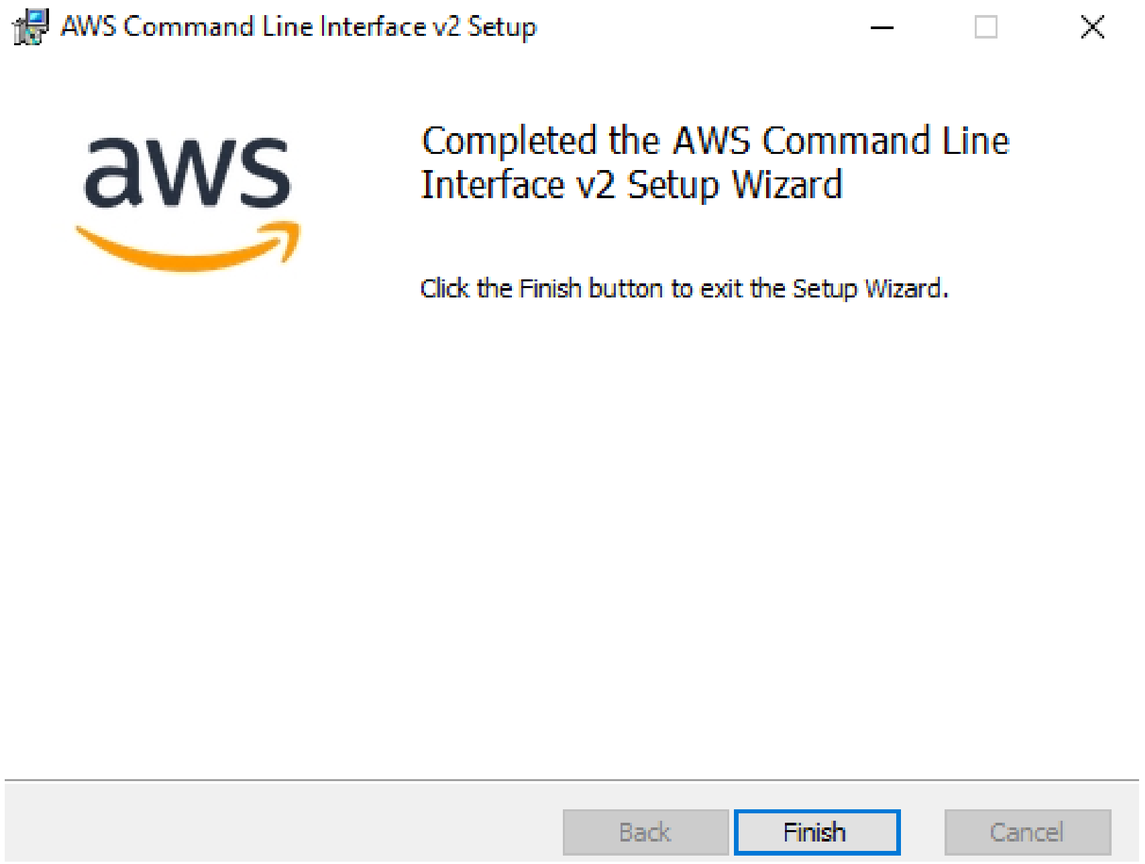
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::/vmmigration-on-prem-server*"
        }
    ]
}
```



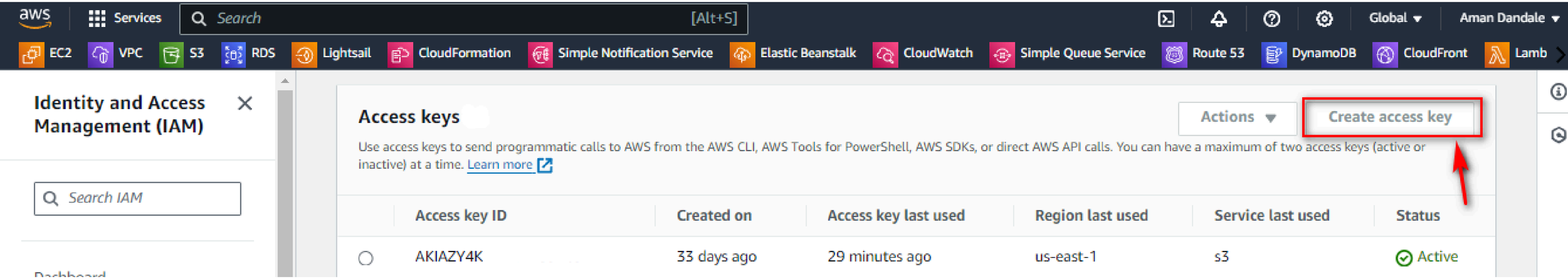
Step-4 : Download and install AWS CLI

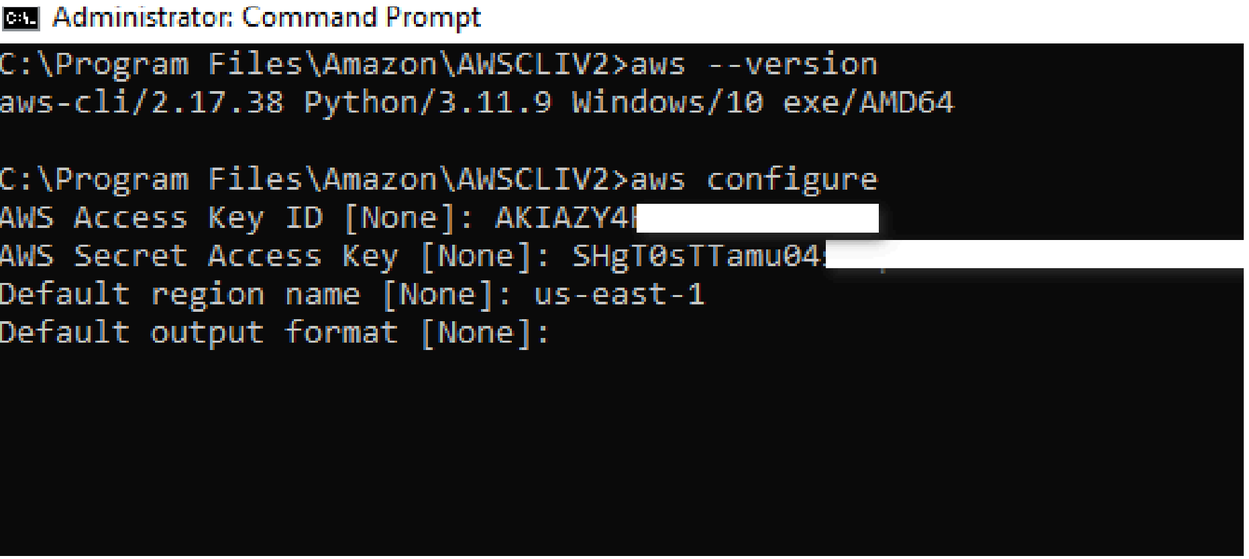
Download AWS CLI: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

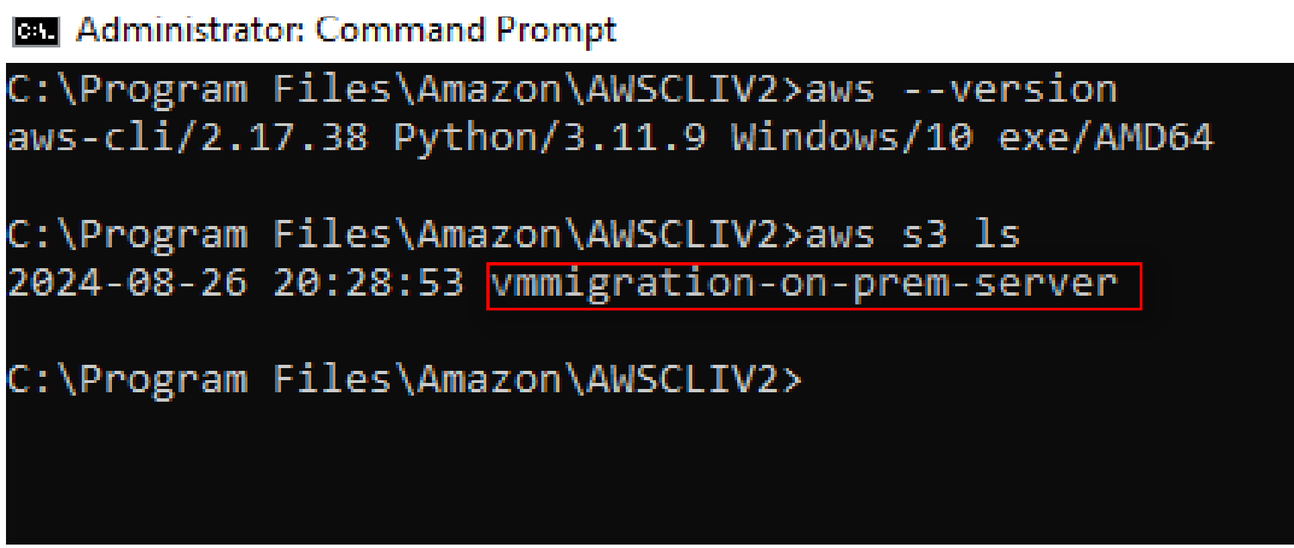




Step-5 : Now configure AWS Cli with aws account throught command prompt







- Use command to sync aws account: aws configure
- Provide your AWS account's Access key ID and Secret Access key ID
- Select your Default region and output format as Json

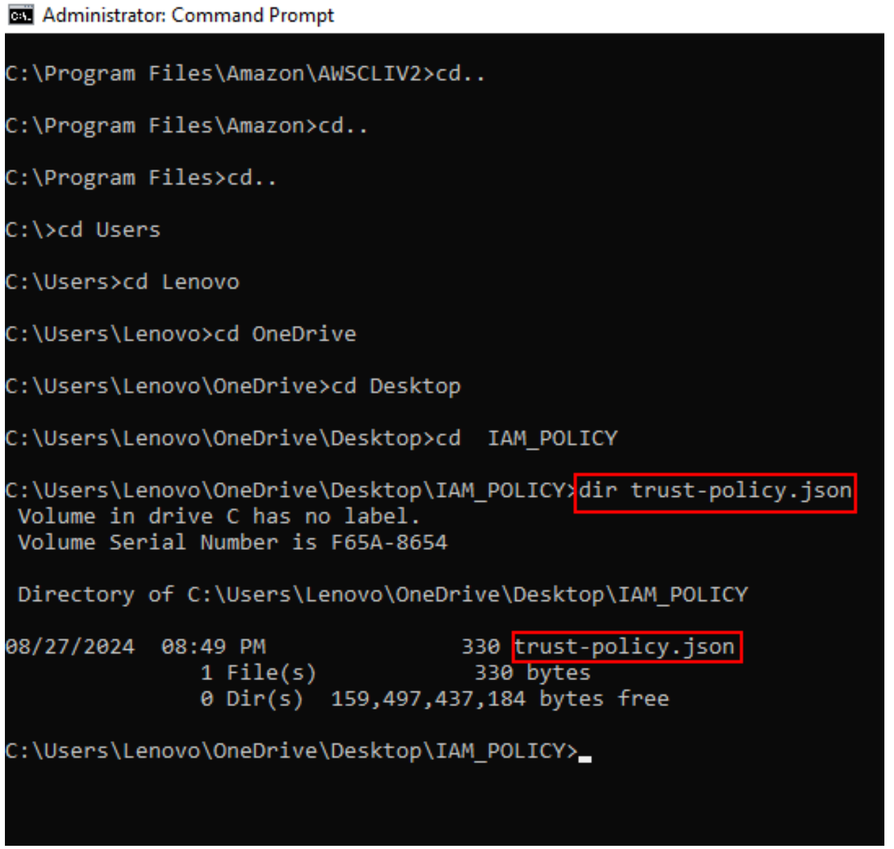
Step-6 : Now create a IAM role for importing the on-prem vm to AWS

- Search on google “Required permissions for vm import/export” you will get
- Documentation: https://docs.aws.amazon.com/vm-import/latest/userguide/required-permissions.html
- Save IAM policy on Desktop with name “trust-policy.json”

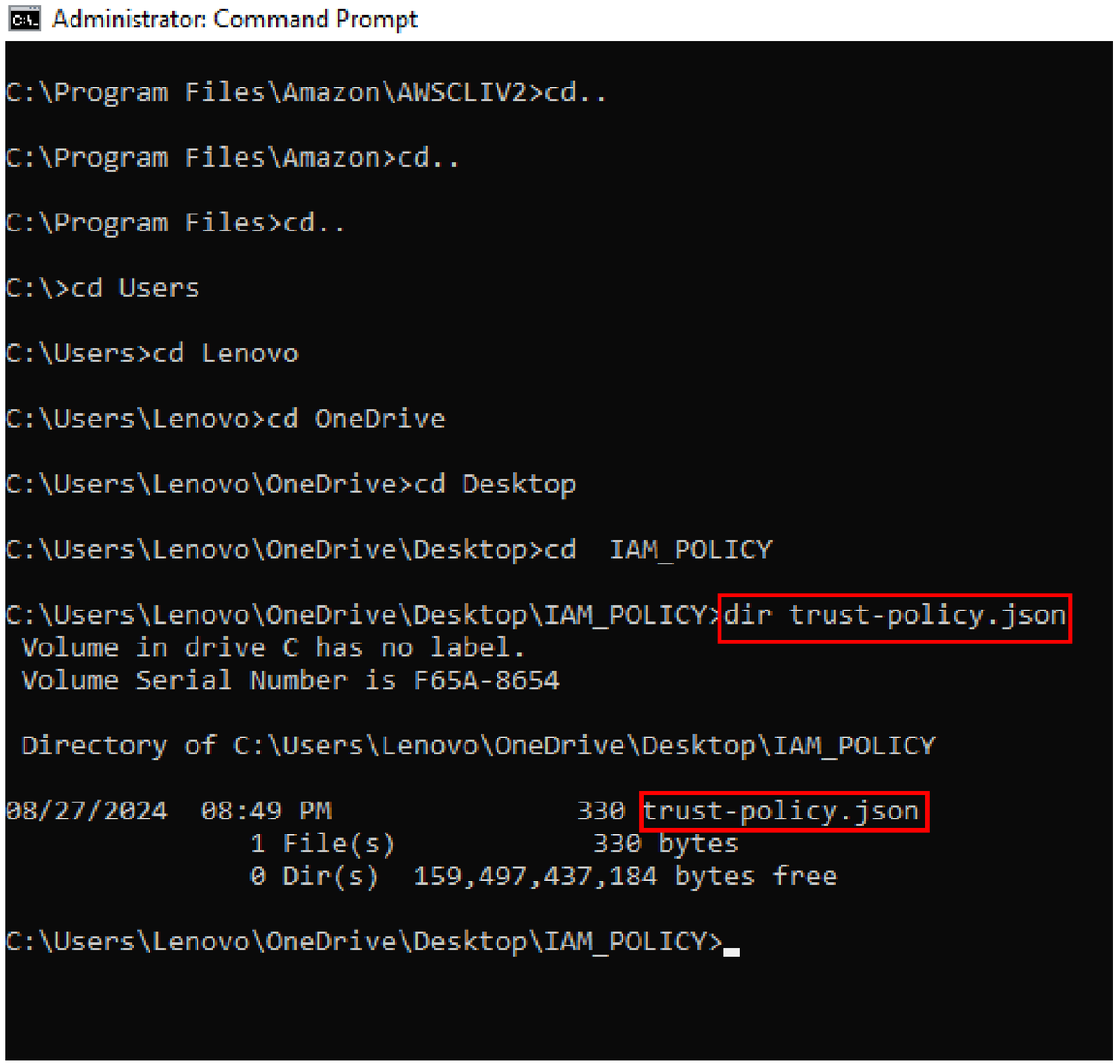
To create the service role

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"
            }
         }
      }
   ]
}
```



Step-7 : Redirect the directory path to the location of file “trust-policy.json”

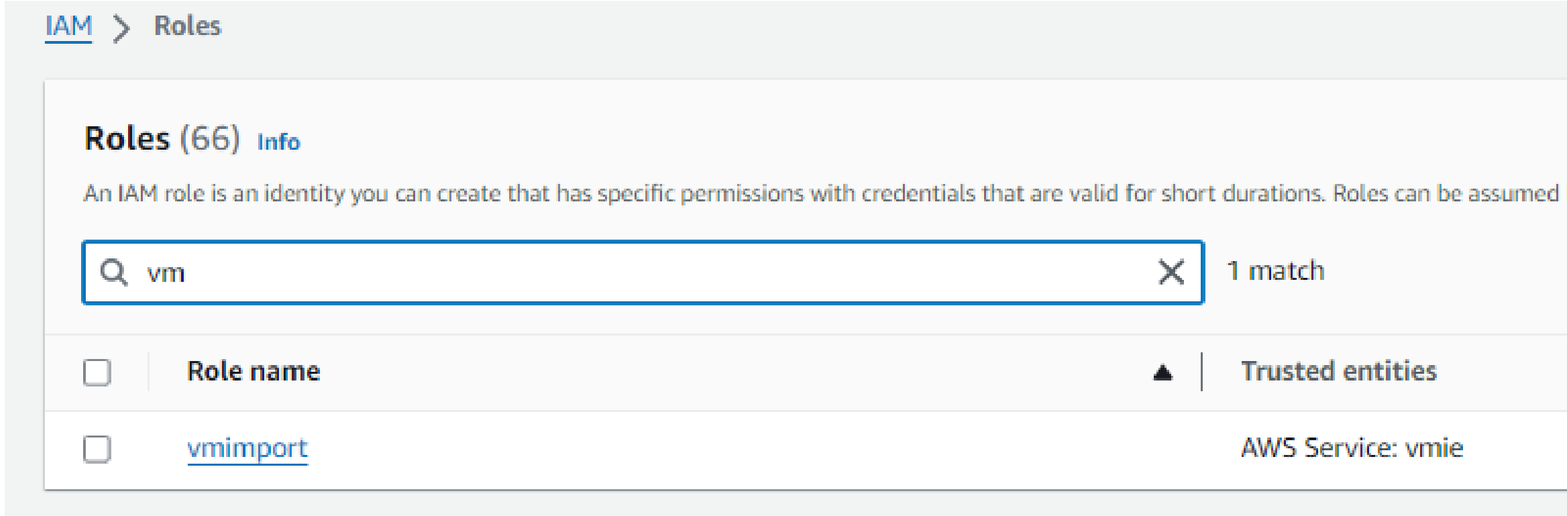


- Then use below command.

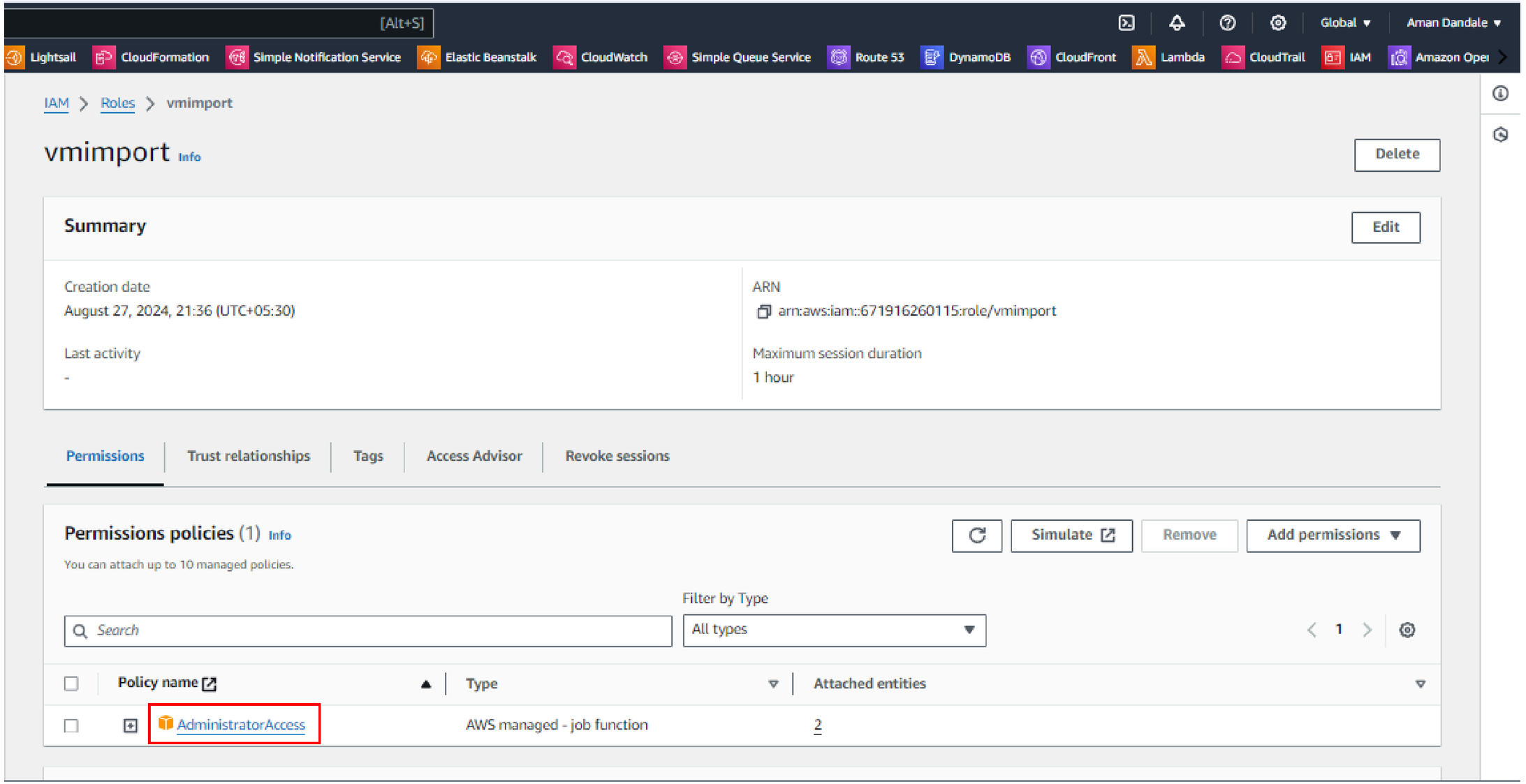
aws iam create-role --role-name vmimport --assume-role-policy-document "file://trust-policy.json"

- It will automatically creates one IAM role for importing on-prem VM to cloud.
- This command only creates IAM role without policy.
- We need to attached required policy to IAM role.

```
C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>dir trust-policy.json
 Volume in drive C has no label.
 Volume Serial Number is F65A-8654

 Directory of C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY

08/27/2024  08:49 PM               330 trust-policy.json
               1 File(s)            330 bytes
               0 Dir(s)  159,495,172,096 bytes free

C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>aws iam create-role --role-name vmimport --assume-role-policy-document "file://trust-policy.json"
{
    "Role": {
        "Path": "/",
        "RoleName": "vmimport",
        "RoleId": "AROAZY4KUPMJW4IRVQV4Q",
        "Arn": "arn:aws:iam::671916260115:role/vmimport",
        "CreateDate": "2024-08-27T16:06:50+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "vmie.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {
                        "StringEquals": {
                            "sts:Externalid": "vmimport"
                        }
                    }
                }
            ]
        }
    }
}
```



Step-8: Attach the required policy to IAM role



Step-9: Importing a VM as an AWS AMI machine using VM Import/Export

- Import your VM as an image : https://docs.aws.amazon.com/vm-import/latest/userguide/import-vm-image.html
- Save policy on Desktop with name “containers.json”

The following is an example `containers.json` file that specifies the image using an S3 bucket.

```
[
  {
    "Description": "My Server OVA",
    "Format": "ova",
    "UserBucket": {
        "S3Bucket": "my-import-bucket",
        "S3Key": "vms/my-server-vm.ova"
    }
  }
]
```

"vmmigration-on-prem-server"

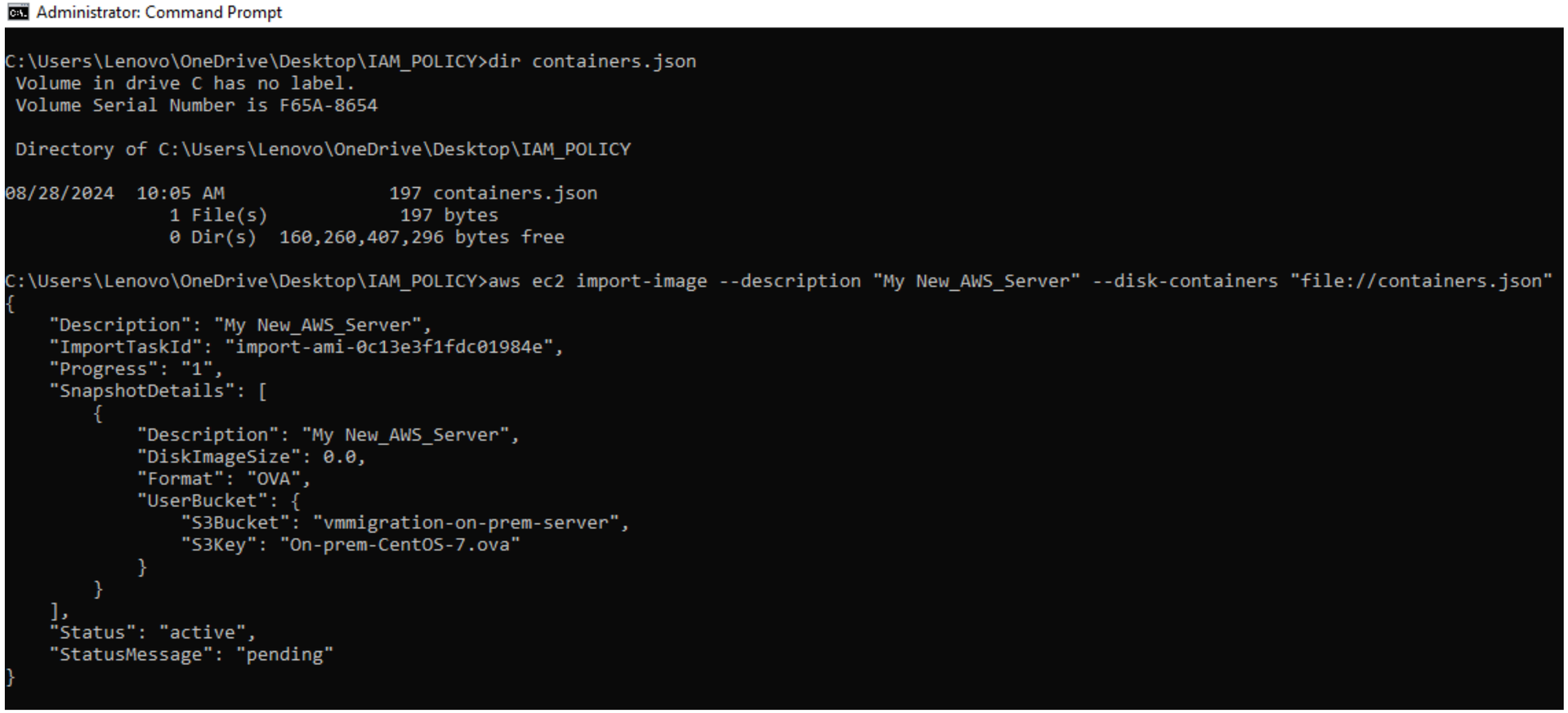
On-prem-CentOS-7.ova

```
[
  {
    "Description": "My New_AWS_Server",
    "Format": "ova",
    "UserBucket": {
        "S3Bucket": "vmmigration-on-prem-server",
        "S3Key": "On-prem-CentOS-7.ova"
    }
  }
]
```

Step-10: Import an image with a single disk

- Run the following command on cmd to import an image with a single disk.

aws ec2 import-image --description "My New_AWS_Server" --disk-containers "file://containers.json"



Step-11: Monitor an import image task

- Use the describe-import-image-tasks command to return the status of an import task.

aws ec2 describe-import-image-tasks --import-task-ids import-ami-0c13e3f1fdc01984e

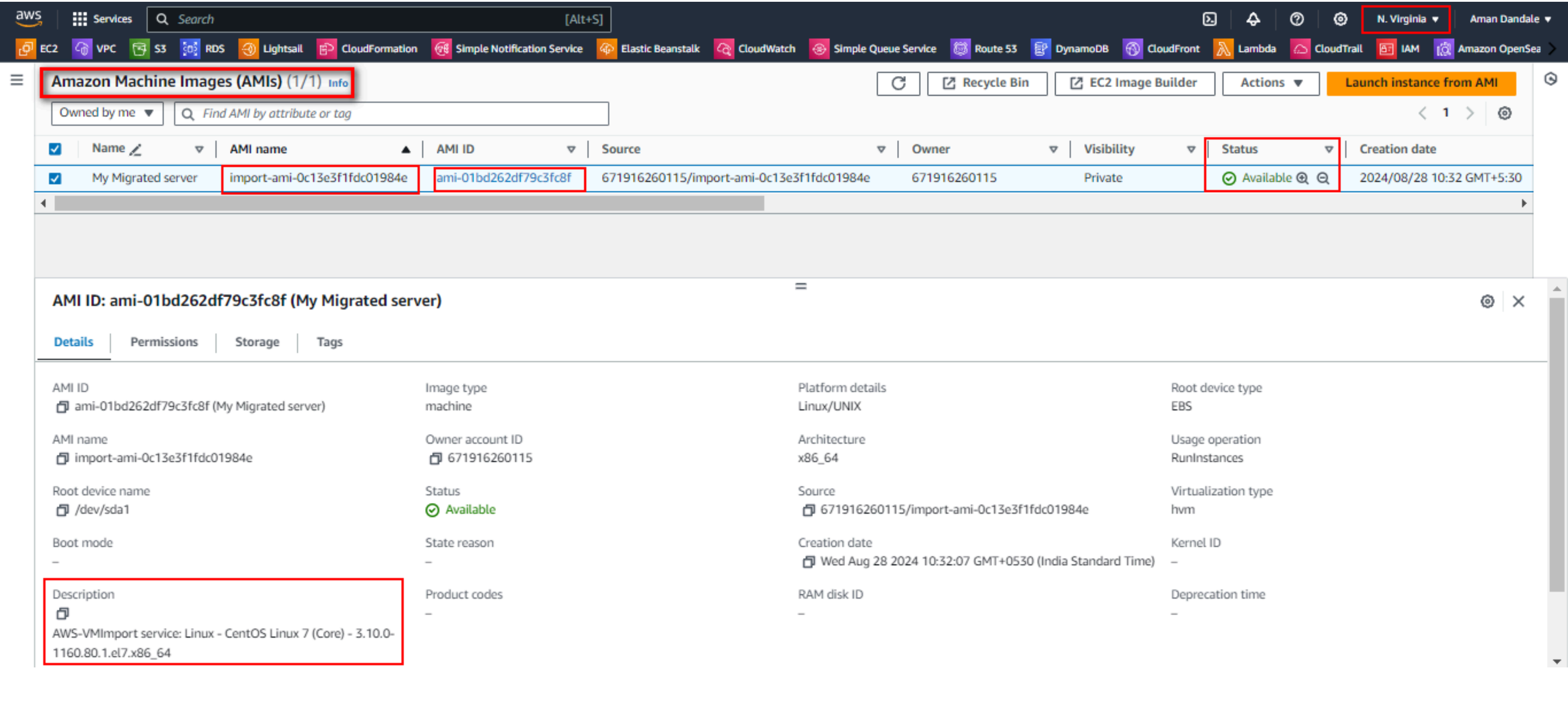
```
C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>aws ec2 describe-import-image-tasks --imp
{
    "ImportImageTasks": [
        {
            "Description": "My New_AWS_Server",
            "ImportTaskId": "import-ami-0c13e3f1fdc01984e",
            "Progress": "20",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 1531146240.0,
                    "Format": "VMDK",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "vmmigration-on-prem-server",
                        "S3Key": "On-prem-CentOS-7.ova"
                    }
                },
                {
                    "DiskImageSize": 330240.0,
                    "Format": "VMDK",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "vmmigration-on-prem-server",
                        "S3Key": "On-prem-CentOS-7.ova"
                    }
                }
            ],
            "Status": "active",
            "StatusMessage": "updating",
            "Tags": []
        }
    ]
}
```

```
C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>aws ec2 describe-import-image-tasks --import-task-ids import-ami-0c13e3f1fdc01984e
{
    "ImportImageTasks": [
        {
            "Architecture": "x86_64",
            "Description": "My New_AWS_Server",
            "ImageId": "ami-01bd262df79c3fc8f",
            "ImportTaskId": "import-ami-0c13e3f1fdc01984e",
            "LicenseType": "BYOL",
            "Platform": "Linux",
            "SnapshotDetails": [
                {
                    "DeviceName": "/dev/sda1",
                    "DiskImageSize": 1531146240.0,
                    "Format": "VMDK",
                    "SnapshotId": "snap-02d07ec8f179f402f",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "vmmigration-on-prem-server",
                        "S3Key": "On-prem-CentOS-7.ova"
                    }
                },
                {
                    "DeviceName": "/dev/sdf",
                    "DiskImageSize": 330240.0,
                    "Format": "VMDK",
                    "SnapshotId": "snap-09978c53efd258304",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "vmmigration-on-prem-server",
                        "S3Key": "On-prem-CentOS-7.ova"
                    }
                }
            ],
            "Status": "completed",
            "Tags": []
        }
    ]
}
```

Launching EC-2

- The process involved creating an Amazon Machine Image (AMI) of the original server and launching a new EC2 instance from this AMI.
- Post-migration, I thoroughly compared the specifications of the original server with the newly created EC2 instance, confirming that all configurations matched perfectly with no data loss or discrepancies.

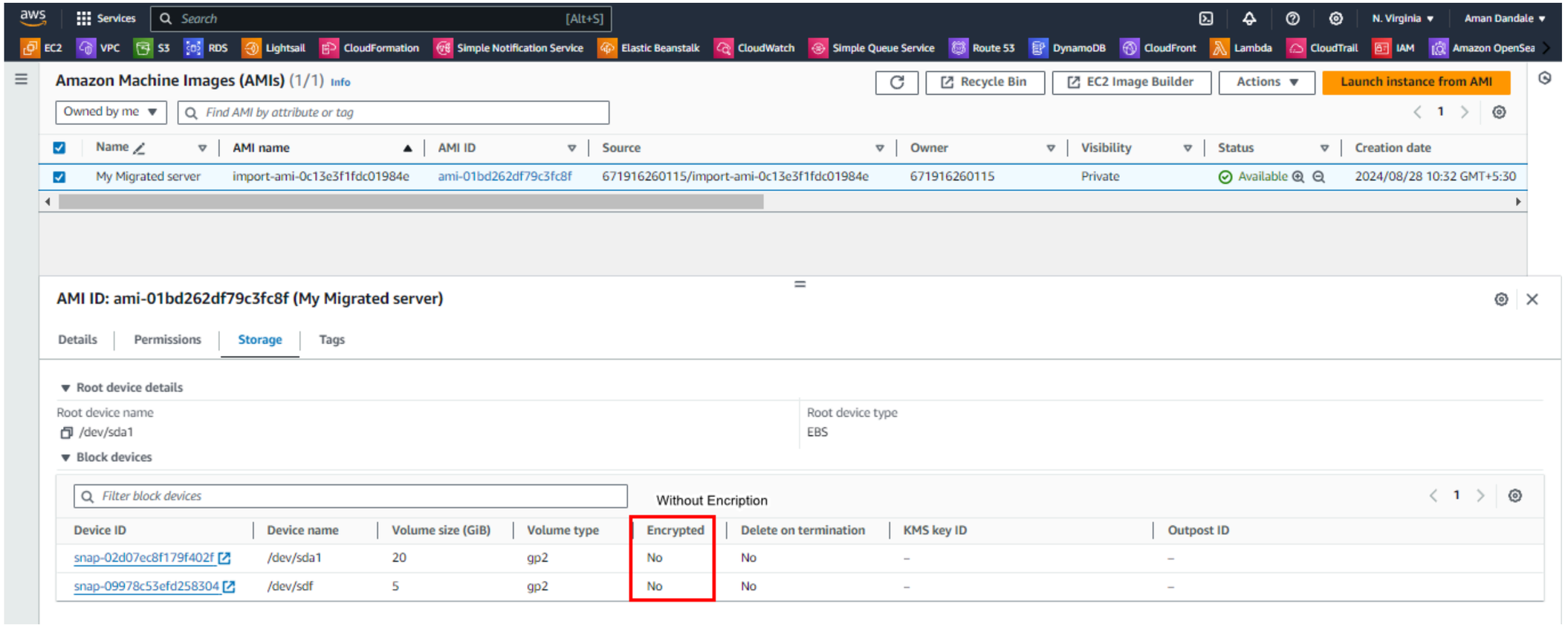
Step-12: Check our imported IAM to respective region with proper IAM ID and Name



Step-13: Import with the encrypted option enabled

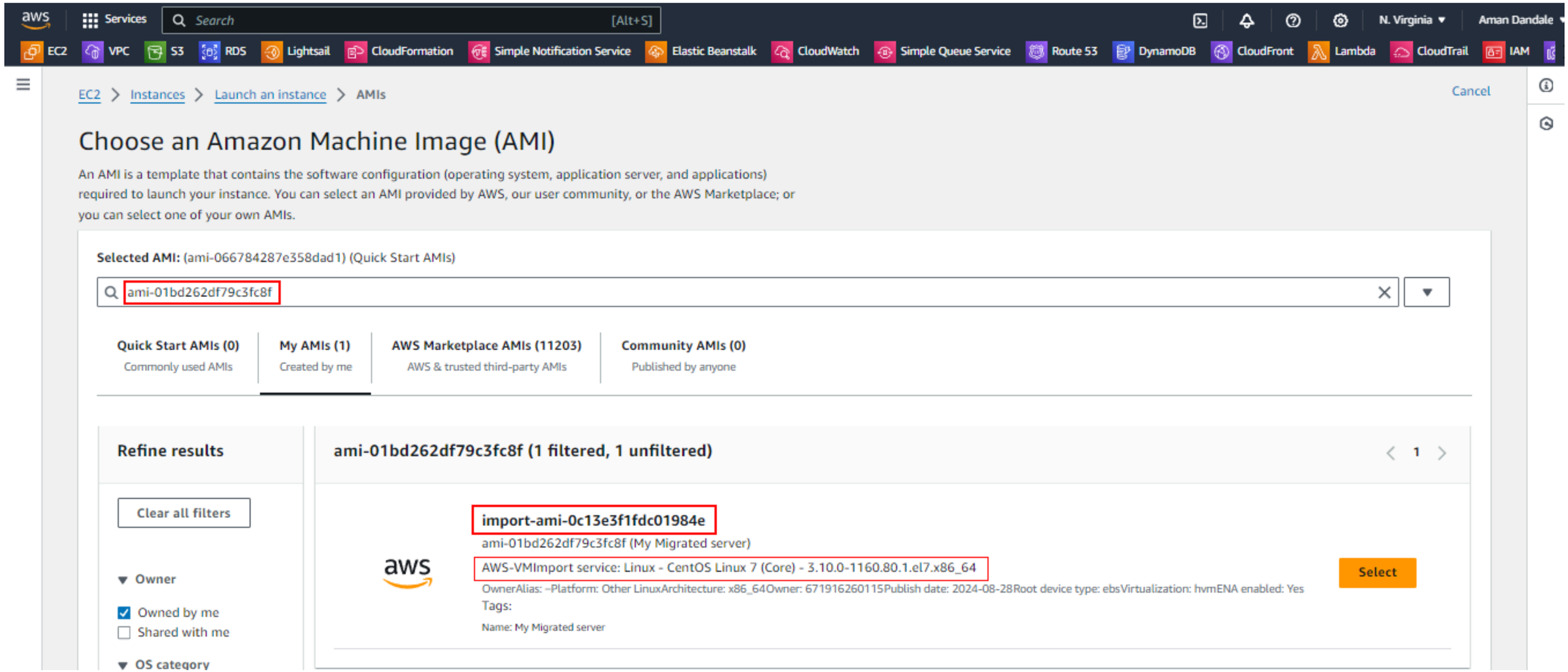
- Use the following command to import an image with an encrypted root volume.

aws ec2 import-image --description "My server disks" --encrypted --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE --disk-containers "file://C:\import\containers.json"

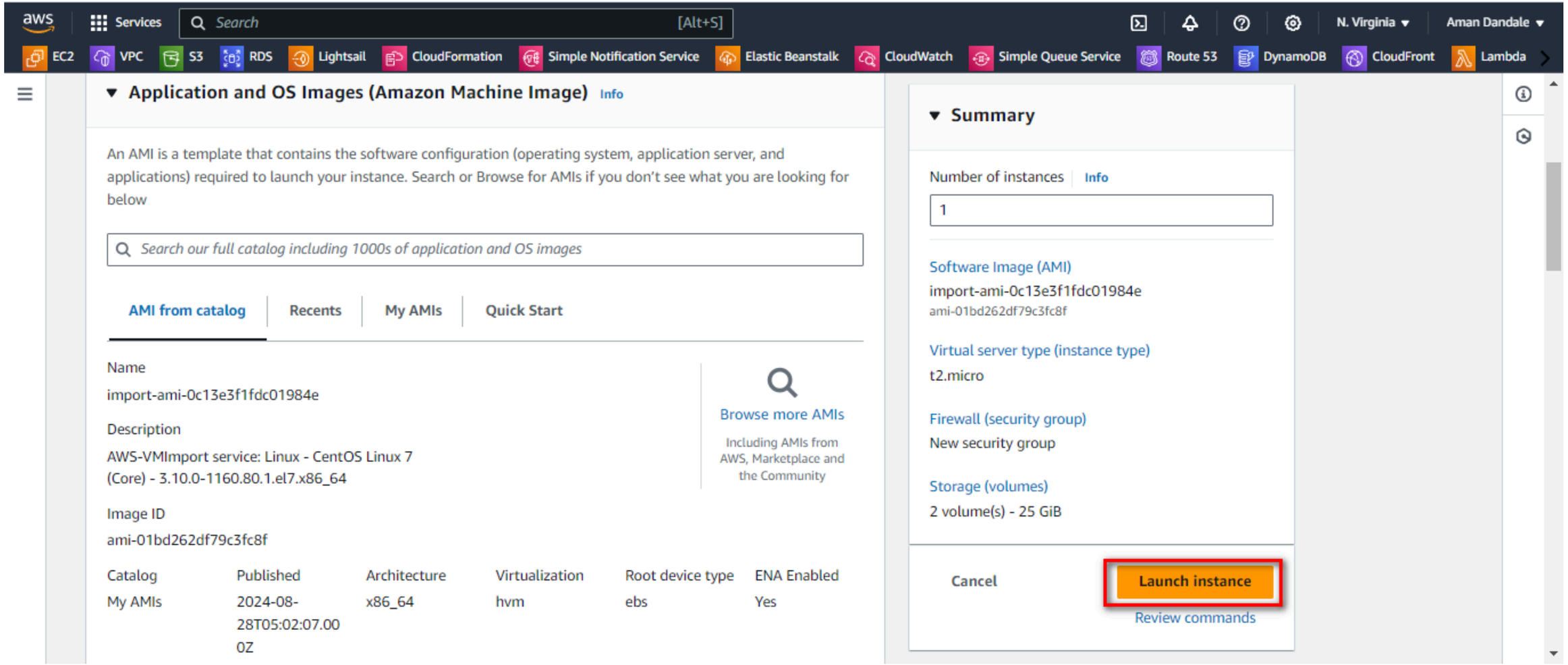


Step-14 Now launch EC-2 from our AMI

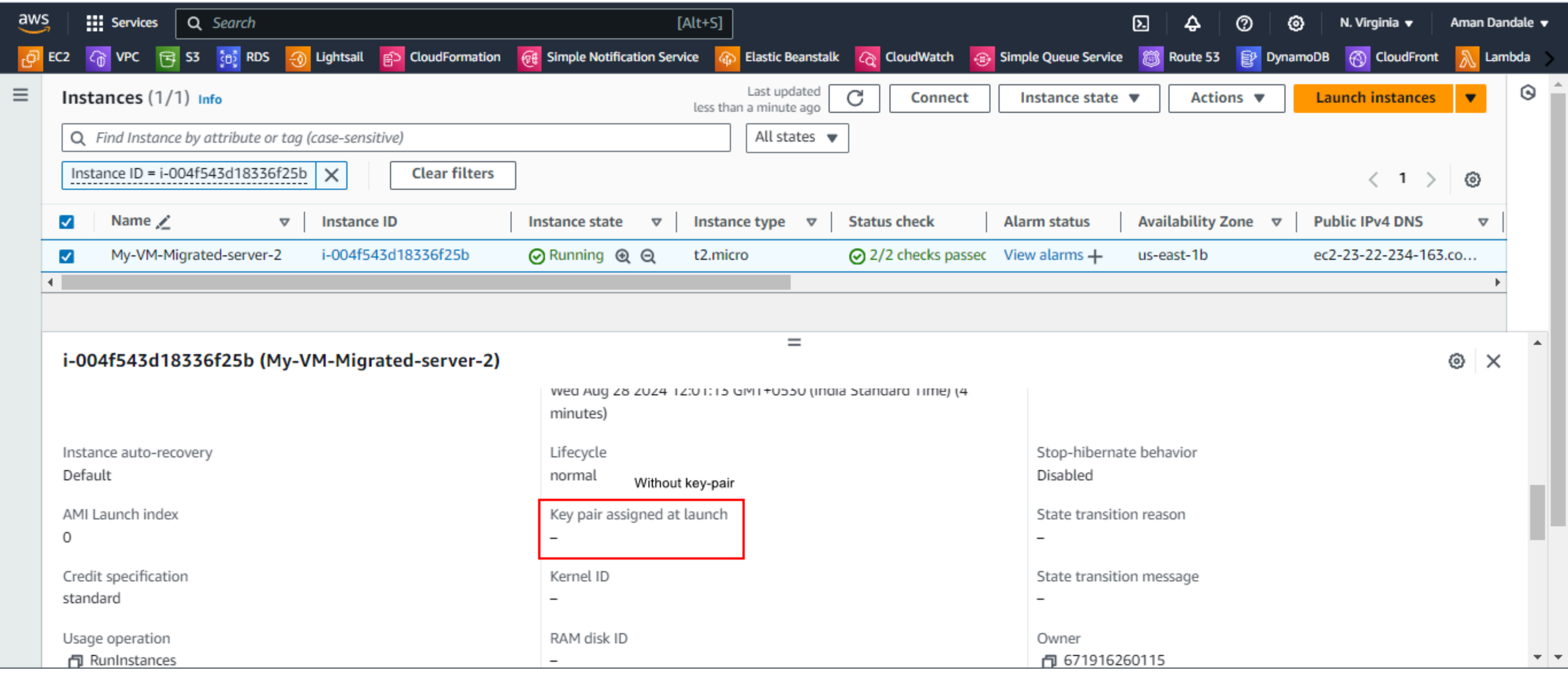
- Search our AMI with AMI ID to lauch EC-2 Instance



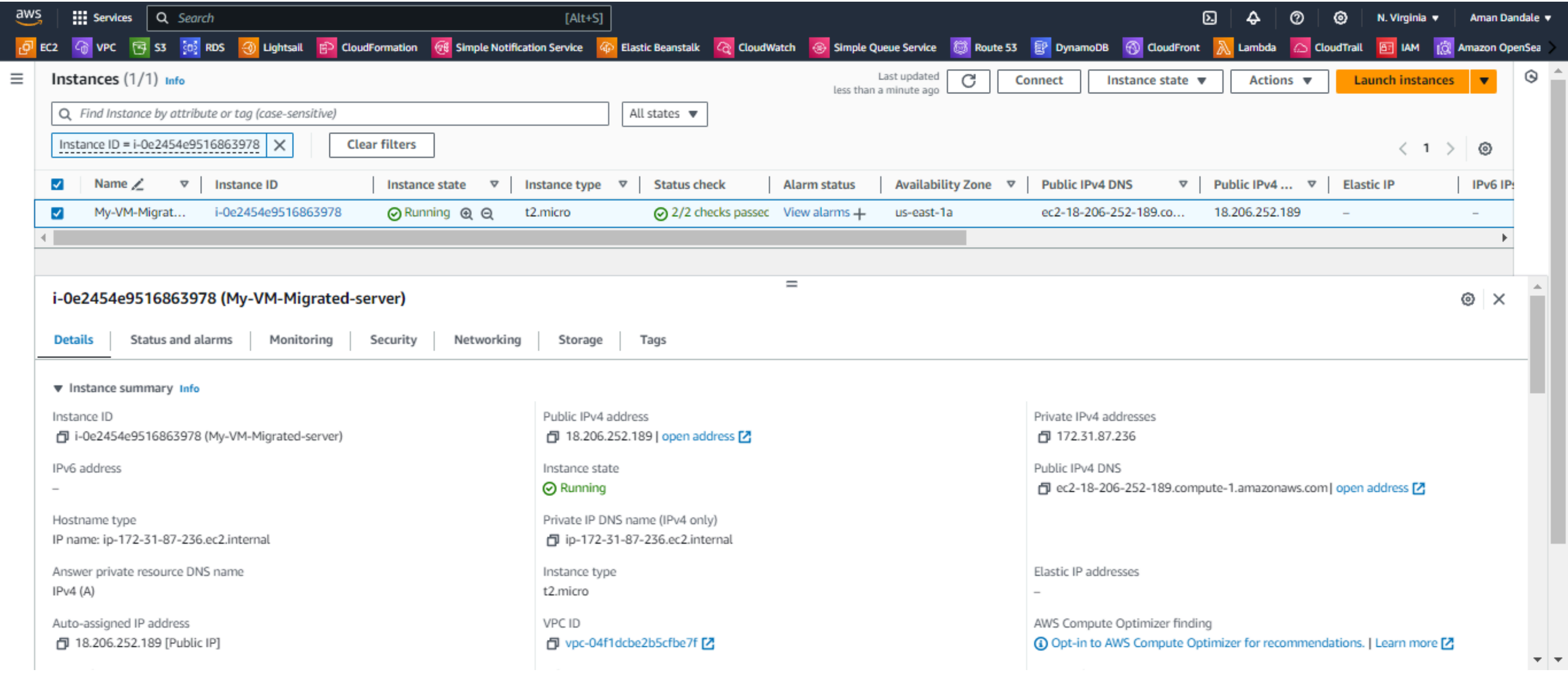
Step-15 Now launch EC-2 from our AMI



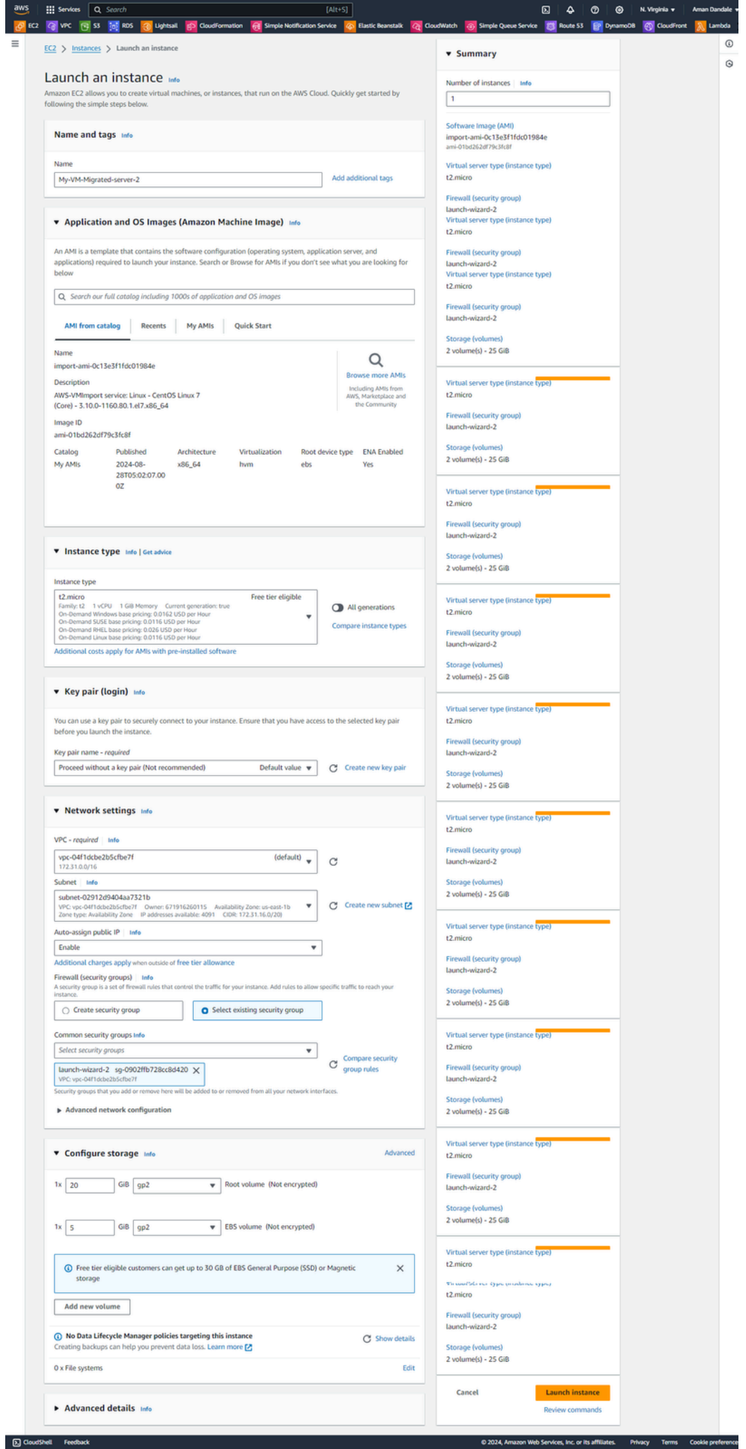
Step-16: EC-2 Instance launch without key-pair

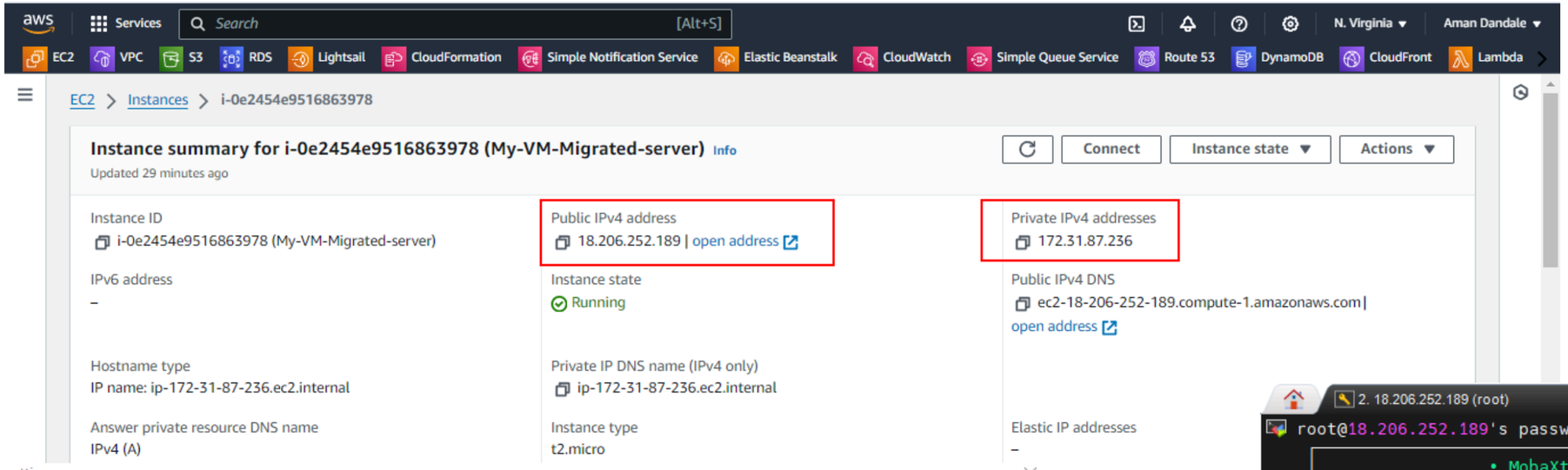


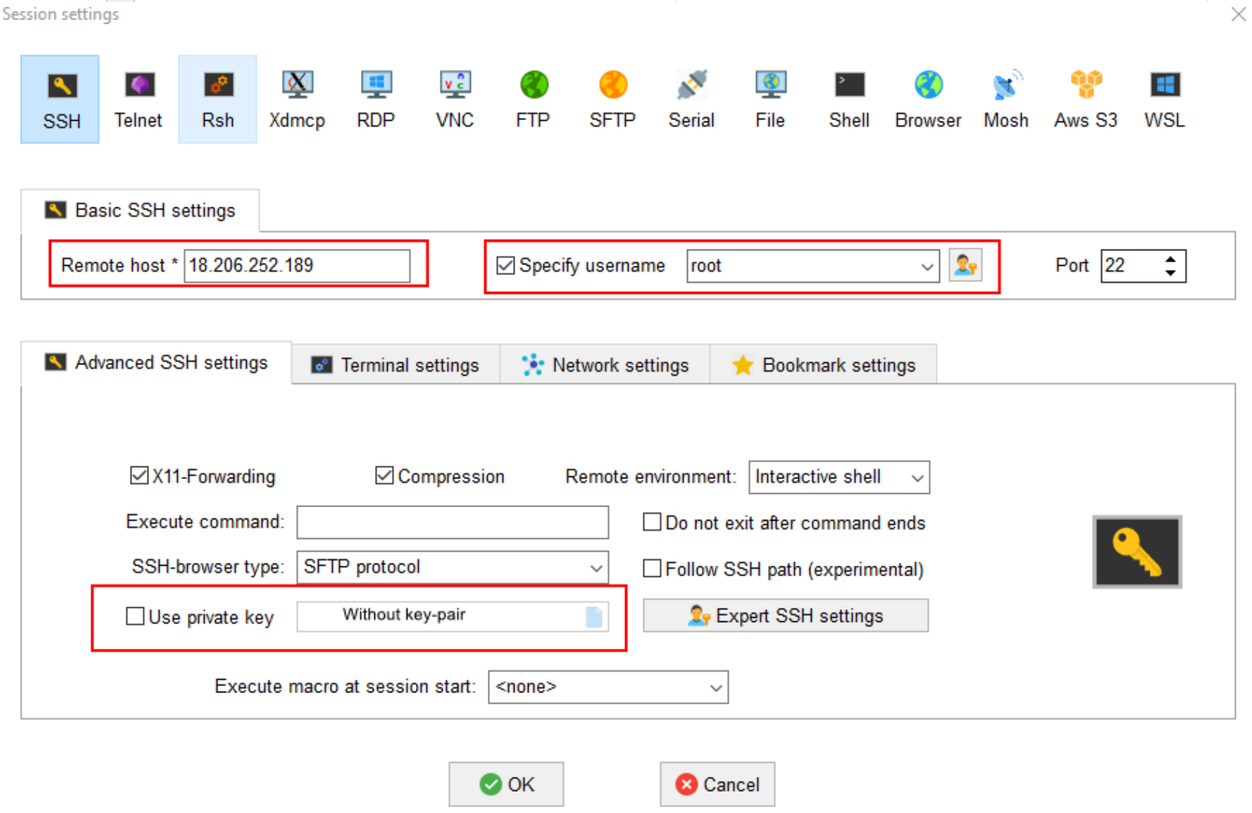
Step-17 Successfully created and launch EC-2 Instance from our on-prem VM-machine

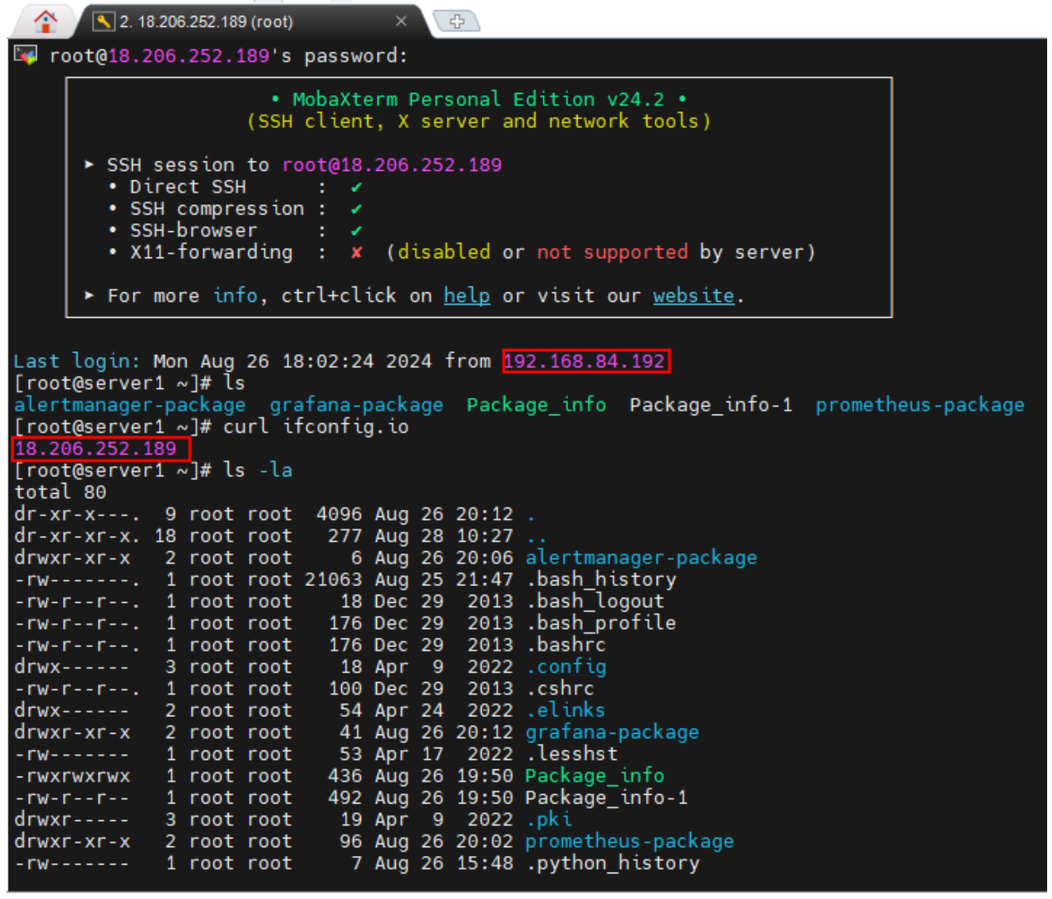


Launch without key-pair in AZ-1b




aws
Services
Search
[Alt+S]
N. Virginia
Aman Dandale
EC2
VPC
S3
RDS
Lightsail
CloudFormation
Simple Notification Service
Elastic Beanstalk
CloudWatch
Simple Queue Service
Route 53
DynamoDB
CloudFront
Lambda
EC2 > Instances > i-0e2454e9516863978
Instance summary for i-0e2454e9516863978 (My-VM-Migrated-server) Info
Updated 29 minutes ago
Connect
Instance state
Actions
Instance ID
i-0e2454e9516863978 (My-VM-Migrated-server)
Public IPv4 address
18.206.252.189 | open address
Private IPv4 addresses
172.31.87.236
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-18-206-252-189.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-87-236.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-87-236.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–


Session settings
SSH
Telnet
Rsh
Xdmcp
RDP
VNC
FTP
SFTP
Serial
File
Shell
Browser
Mosh
Aws S3
WSL
Basic SSH settings
Remote host * 18.206.252.189
Specify username
root
Port 22
Advanced SSH settings
Terminal settings
Network settings
Bookmark settings
X11-Forwarding
Compression
Remote environment: Interactive shell
Execute command:
Do not exit after command ends
SSH-browser type: SFTP protocol
Follow SSH path (experimental)
Use private key
Without key-pair
Expert SSH settings
Execute macro at session start: <none>
OK
Cancel


2. 18.206.252.189 (root)
root@18.206.252.189's password:
• MobaXterm Personal Edition v24.2 •
(SSH client, X server and network tools)
▸ SSH session to root@18.206.252.189
• Direct SSH : ✔
• SSH compression : ✔
• SSH-browser : ✔
• X11-forwarding : ✘ (disabled or not supported by server)
▸ For more info, ctrl+click on help or visit our website.
Last login: Mon Aug 26 18:02:24 2024 from 192.168.84.192
[root@server1 ~]# ls
alertmanager-package grafana-package Package_info Package_info-1 prometheus-package
[root@server1 ~]# curl ifconfig.io
18.206.252.189
[root@server1 ~]# ls -la
total 80
dr-xr-x---. 9 root root 4096 Aug 26 20:12 .
dr-xr-xr-x. 18 root root 277 Aug 28 10:27 ..
drwxr-xr-x 2 root root 6 Aug 26 20:06 alertmanager-package
-rw-------. 1 root root 21063 Aug 25 21:47 .bash_history
-rw-r--r--. 1 root root 18 Dec 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 Dec 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
drwx------ 3 root root 18 Apr 9 2022 .config
-rw-r--r--. 1 root root 100 Dec 29 2013 .cshrc
drwx------ 2 root root 54 Apr 24 2022 .elinks
drwxr-xr-x 2 root root 41 Aug 26 20:12 grafana-package
-rw------- 1 root root 53 Apr 17 2022 .lesshst
-rwxrwxrwx 1 root root 436 Aug 26 19:50 Package_info
-rw-r--r-- 1 root root 492 Aug 26 19:50 Package_info-1
drwxr----- 3 root root 19 Apr 9 2022 .pki
drwxr-xr-x 2 root root 96 Aug 26 20:02 prometheus-package
-rw------- 1 root root 7 Aug 26 15:48 .python_history

Step-18: Comparison

On-prem-VM-server

AWS EC-2 Instance (After migration to AWS)

Server-1 [Running] - Oracle VM VirtualBox

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.80.1.el7.x86_64 on an x86_64

server1 login: root
Password:
Last login: Wed Aug 28 20:10:55 on tty1
[root@server1 ~]# ls
alertmanager-package  grafana-package  Package_info  Package_info-1  prometheus-package
[root@server1 ~]# ls -l
total 8
drwxr-xr-x 2 root root   6 Aug 26 20:06 alertmanager-package
drwxr-xr-x 2 root root  41 Aug 26 20:12 grafana-package
-rwxrwxrwx 1 root root 436 Aug 26 19:50 Package_info
-rw-r--r-- 1 root root 492 Aug 26 19:50 Package_info-1
drwxr-xr-x 2 root root  96 Aug 26 20:02 prometheus-package
[root@server1 ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.84.196  netmask 255.255.255.0  broadcast 192.168.84.255
        inet6 2401:4900:5618:98ed:2ad:50d2:c143:cdf4  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::703f:b1d1:cadf:3f7e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:da:a2:19  txqueuelen 1000  (Ethernet)
        RX packets 6  bytes 742 (742.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 1344 (1.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2  bytes 112 (112.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2  bytes 112 (112.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@server1 ~]#
```

11. 18.206.252.189 (root)

```
root@18.206.252.189's password:

                • MobaXterm Personal Edition v24.2 •
              (SSH client, X server and network tools)

▸ SSH session to root@18.206.252.189
  • Direct SSH      :  ✔
  • SSH compression :  ✔
  • SSH-browser     :  ✔
  • X11-forwarding  :  ✘  (disabled or not supported by server)

▸ For more info, ctrl+click on help or visit our website.


Last login: Wed Aug 28 20:17:09 2024 from 117.99.255.180
[root@server1 ~]# ls
alertmanager-package  grafana-package  mybackup.tar.gz  Package_info  Package_info-1  prometheus-package
[root@server1 ~]# curl ifconfig.io
18.206.252.189
[root@server1 ~]# ls -l
total 11860
drwxr-xr-x 2 root root        6 Aug 26 20:06 alertmanager-package
drwxr-xr-x 2 root root       41 Aug 26 20:12 grafana-package
-rw-r--r-- 1 root root 12134781 Aug 28 18:05 mybackup.tar.gz
-rwxrwxrwx 1 root root      436 Aug 26 19:50 Package_info
-rw-r--r-- 1 root root      492 Aug 26 19:50 Package_info-1
drwxr-xr-x 2 root root       96 Aug 26 20:02 prometheus-package
[root@server1 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.87.236  netmask 255.255.240.0  broadcast 172.31.95.255
        inet6 fe80::1064:eeff:fe70:da5  prefixlen 64  scopeid 0x20<link>
        ether 12:64:ee:70:0d:a5  txqueuelen 1000  (Ethernet)
        RX packets 2809  bytes 355310 (346.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2678  bytes 486706 (475.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Comparison: On-prem-VM-Server AWS EC-2 Server

General

Name: Server-1
Operating System: Red Hat (64-bit)

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Server-1-disk001.vdi (Normal, 20.00 GB)
SATA Port 1: Server-1-disk002.vdi (Normal, 5.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Qualcomm Atheros QCA9377 Wireless Network Adapter)

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

None

Description

None

(My-VM-Migrated-server-1)

Root de

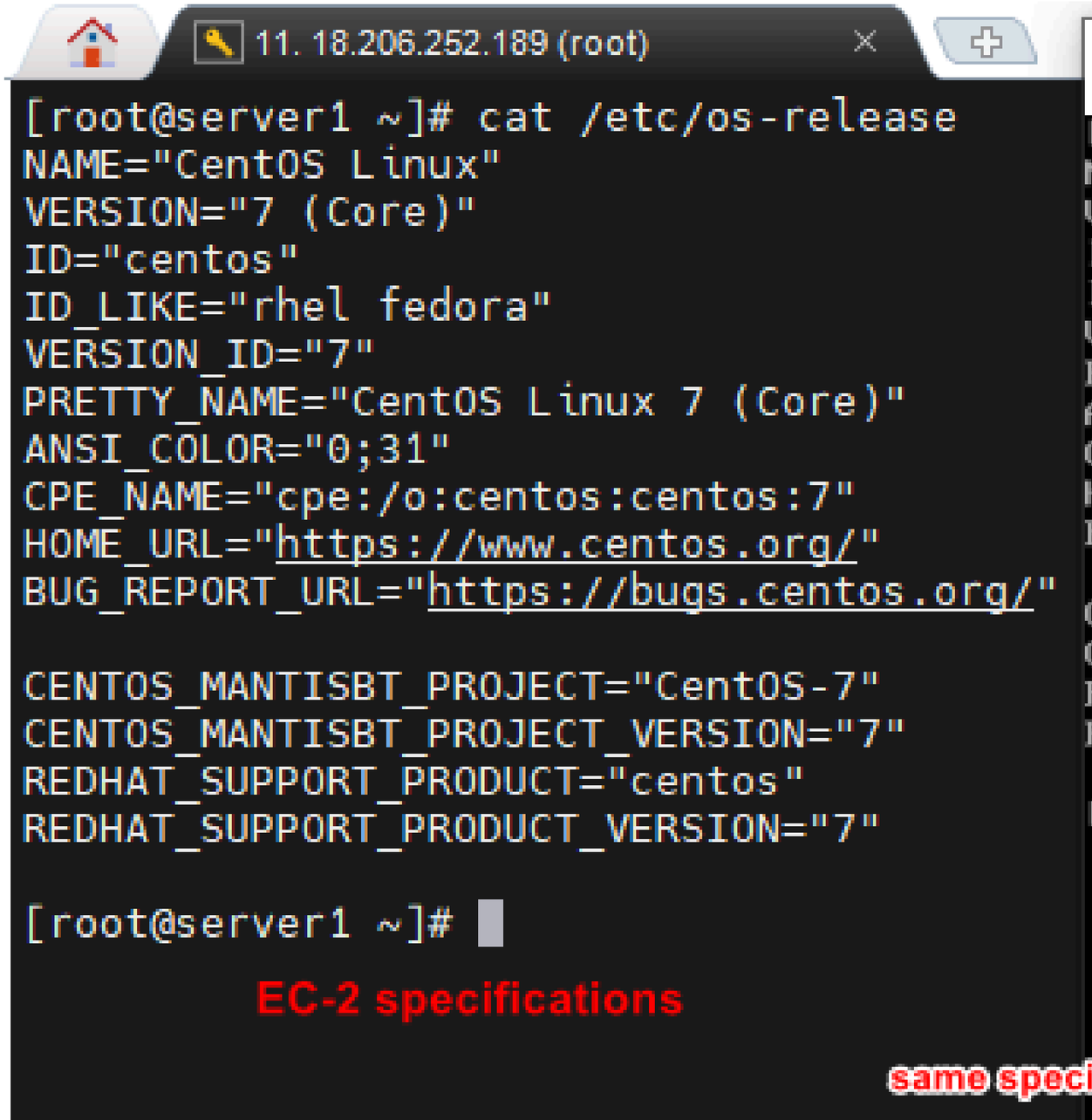
EBS

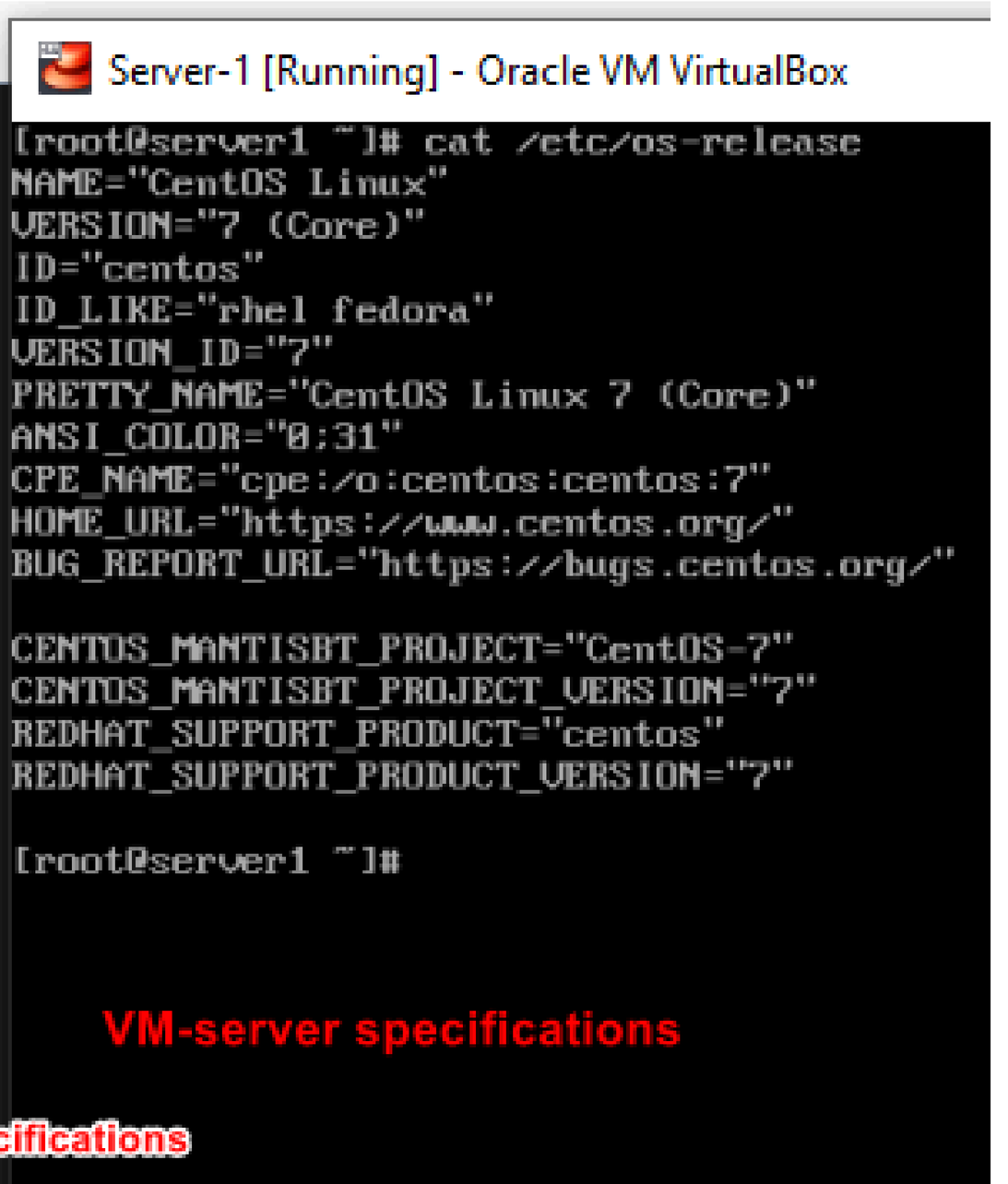
i-004f543d18336f25b (My-VM-Migrated-server-1)

Number of vCPUs
1

	Device name	Volume siz...	A
:216d	/dev/sda1	20	
1821e	/dev/sdf	5	

Operating System specifications of both server

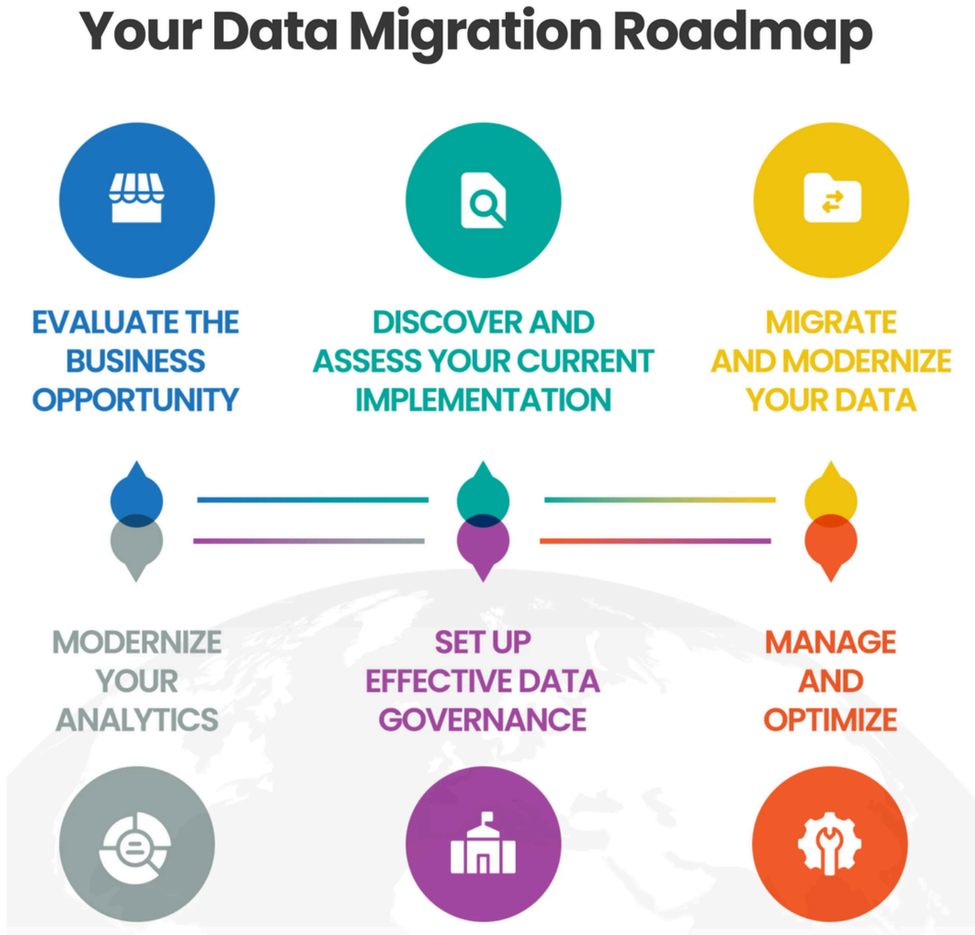




same specifications

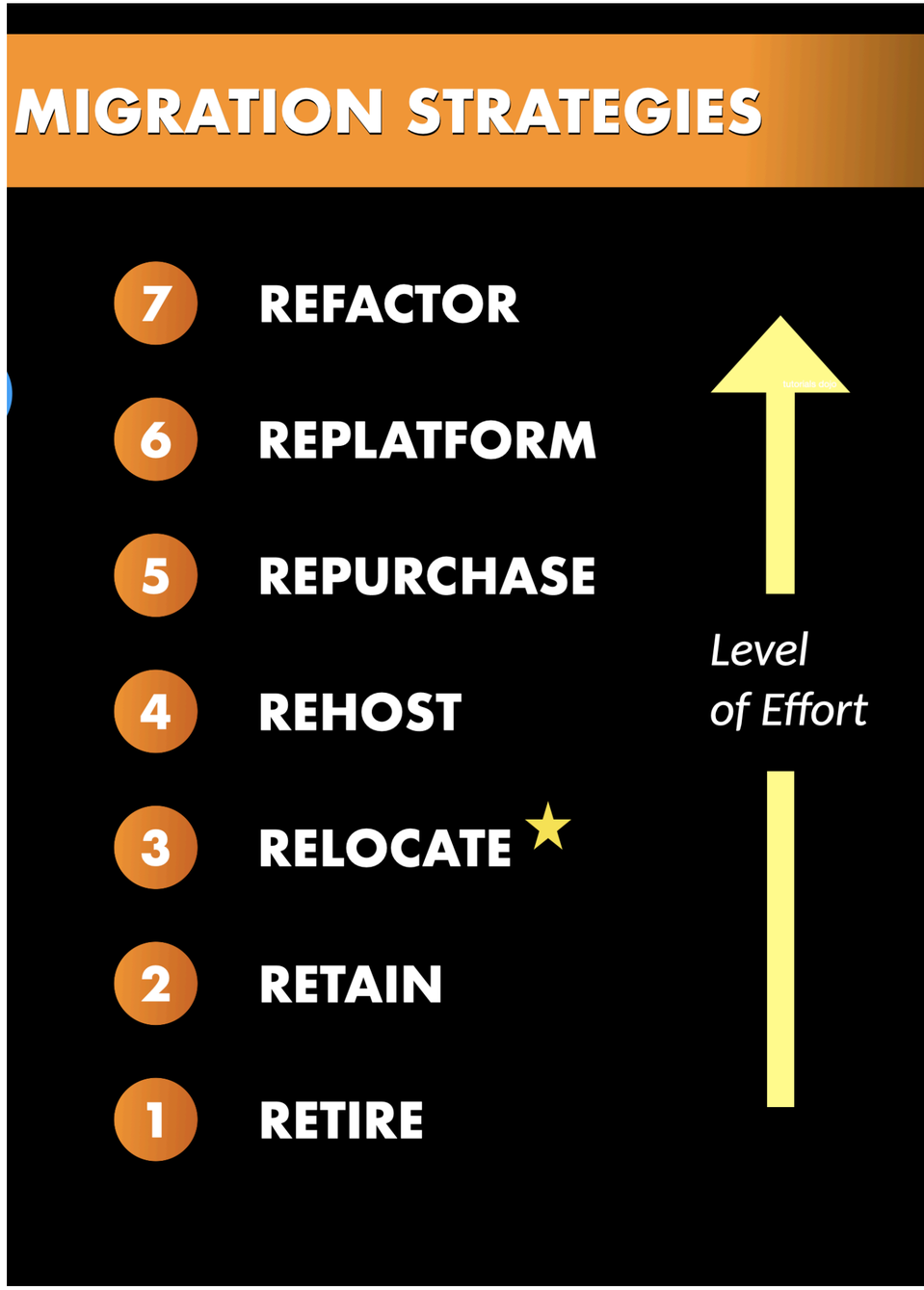
Data Analysis

- The migration process involved successfully transitioning a CentOS server from an on-premises virtual machine to an AWS EC2 instance using the Re-hosting (lift-and-shift) strategy.
- Post-migration analysis confirmed that the server specifications, including CPU, RAM, and storage, matched perfectly with the original environment, ensuring no data loss or performance degradation.
- Additionally, the migration process was completed efficiently, with minimal downtime and verified data integrity.
- Cost analysis indicated potential savings by leveraging AWS's scalable infrastructure.



Migration Strategies

- I have used Rehost migration strategies, also known as lift-and-shift.
- Re-hosting involves moving applications without making any changes to the architecture or the codebase of application.
- This strategy is typically chosen when the goal is to move quickly to the cloud with minimal modifications to the existing environment,



Conclusion

- In this project, I employed the Re-hosting (lift-and-shift) migration strategy,
- which involved migrating the CentOS server to AWS using the AWS CLI without any changes to the server's architecture or codebase.
- The process involved creating an Amazon Machine Image (AMI) of the original server and launching a new EC2 instance from this AMI.
- This approach allowed for a seamless and efficient transition to the cloud, ensuring that all server specifications remained intact and consistent with the original virtual machine environment.