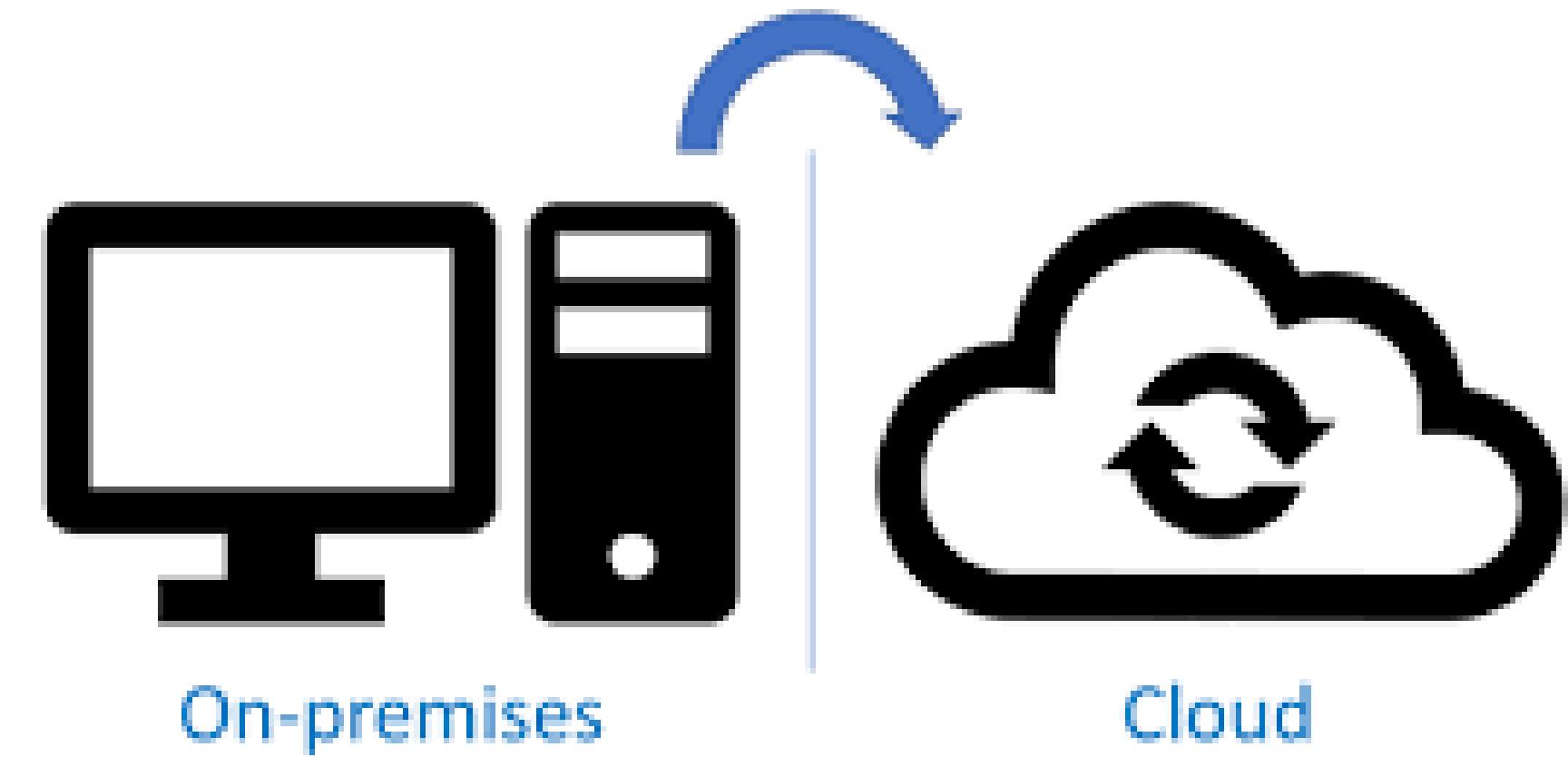


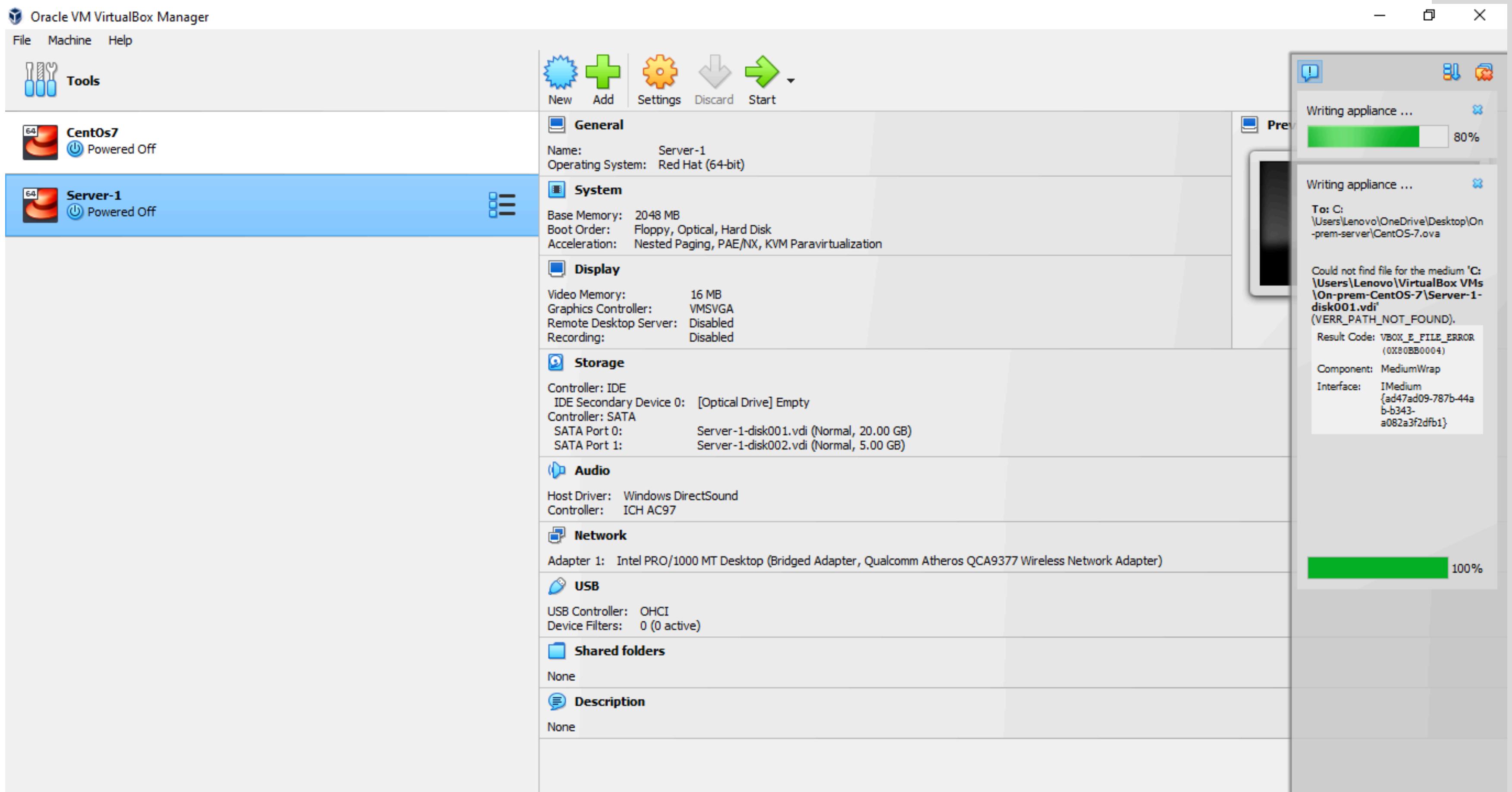
SERVER MIGRATION

REPORT

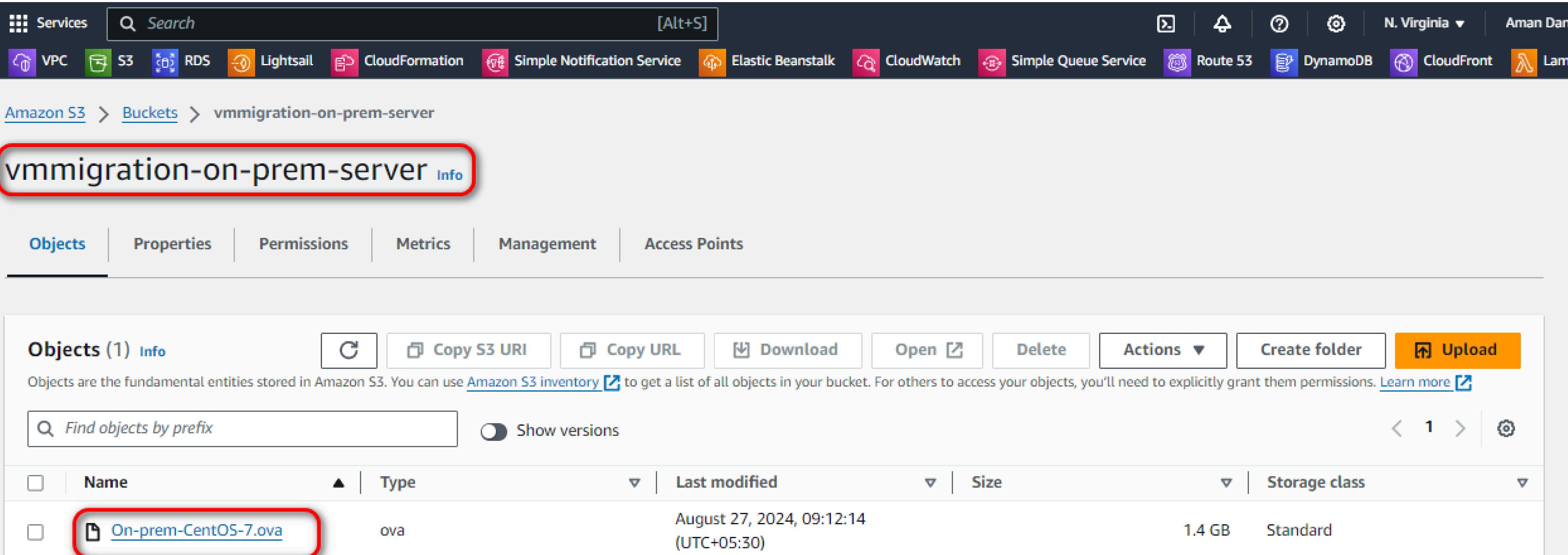
FROM ON-PREM TO AWS CLOUD



Step-1 : Extract On-Prem server



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

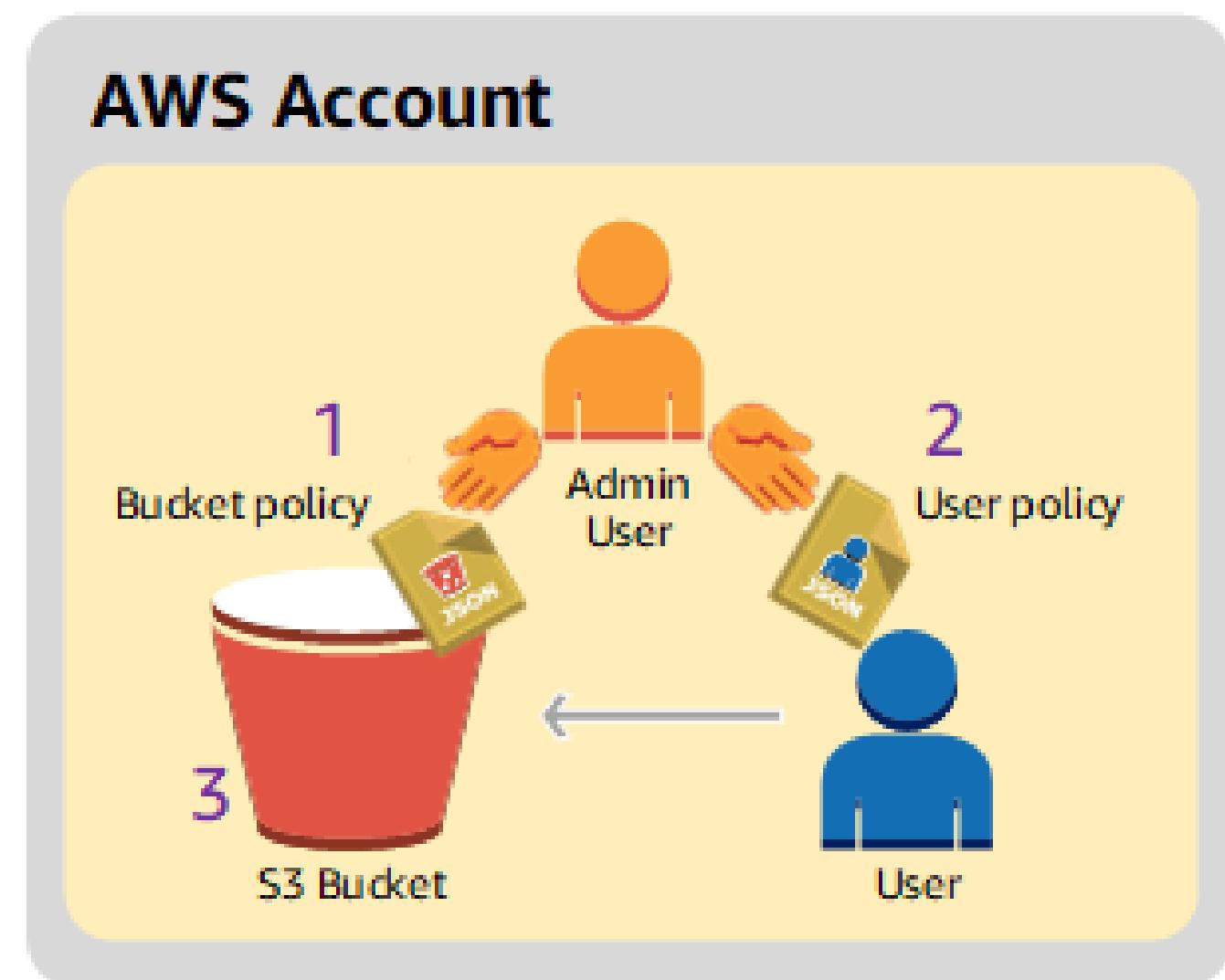


The screenshot shows the Amazon S3 console interface. At the top, there's a navigation bar with various services like VPC, S3, RDS, Lightsail, CloudFormation, Simple Notification Service, Elastic Beanstalk, CloudWatch, Simple Queue Service, Route 53, DynamoDB, CloudFront, and Lambda. Below the navigation bar, the path 'Amazon S3 > Buckets > vmmigration-on-prem-server' is displayed. The main title 'vmmigration-on-prem-server' is highlighted with a red box. Below the title, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is selected. On the left, there's a summary section for 'Objects (1) Info' with buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A note says 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' Below this, there's a search bar 'Find objects by prefix' and a 'Show versions' button. The main table lists one object: 'On-prem-CentOS-7.ova' (Type: ova, Last modified: August 27, 2024, 09:12:14 (UTC+05:30), Size: 1.4 GB, Storage class: Standard). The file name 'On-prem-CentOS-7.ova' is also highlighted with a red box.

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	On-prem-CentOS-7.ova	ova	August 27, 2024, 09:12:14 (UTC+05:30)	1.4 GB	Standard

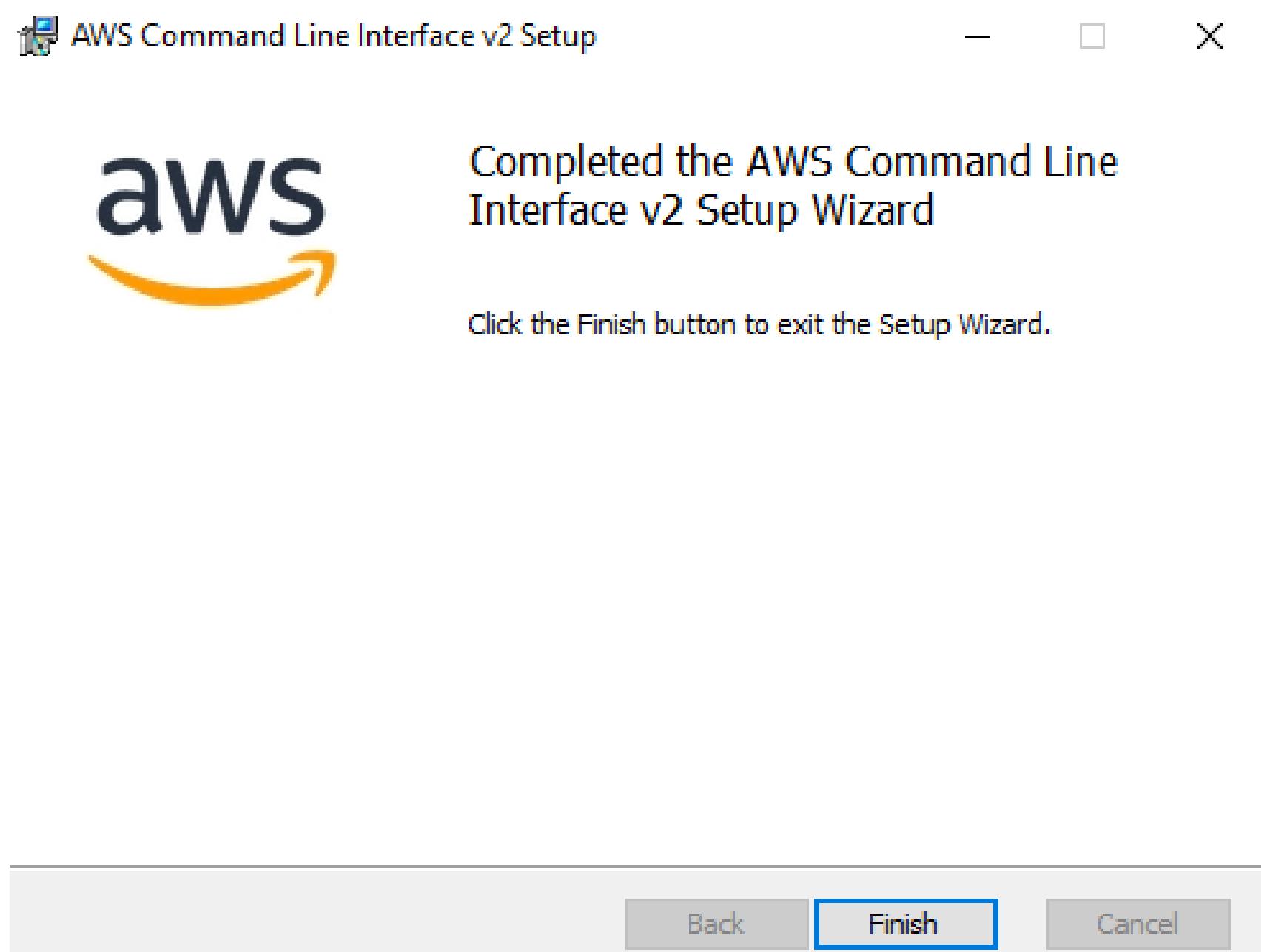
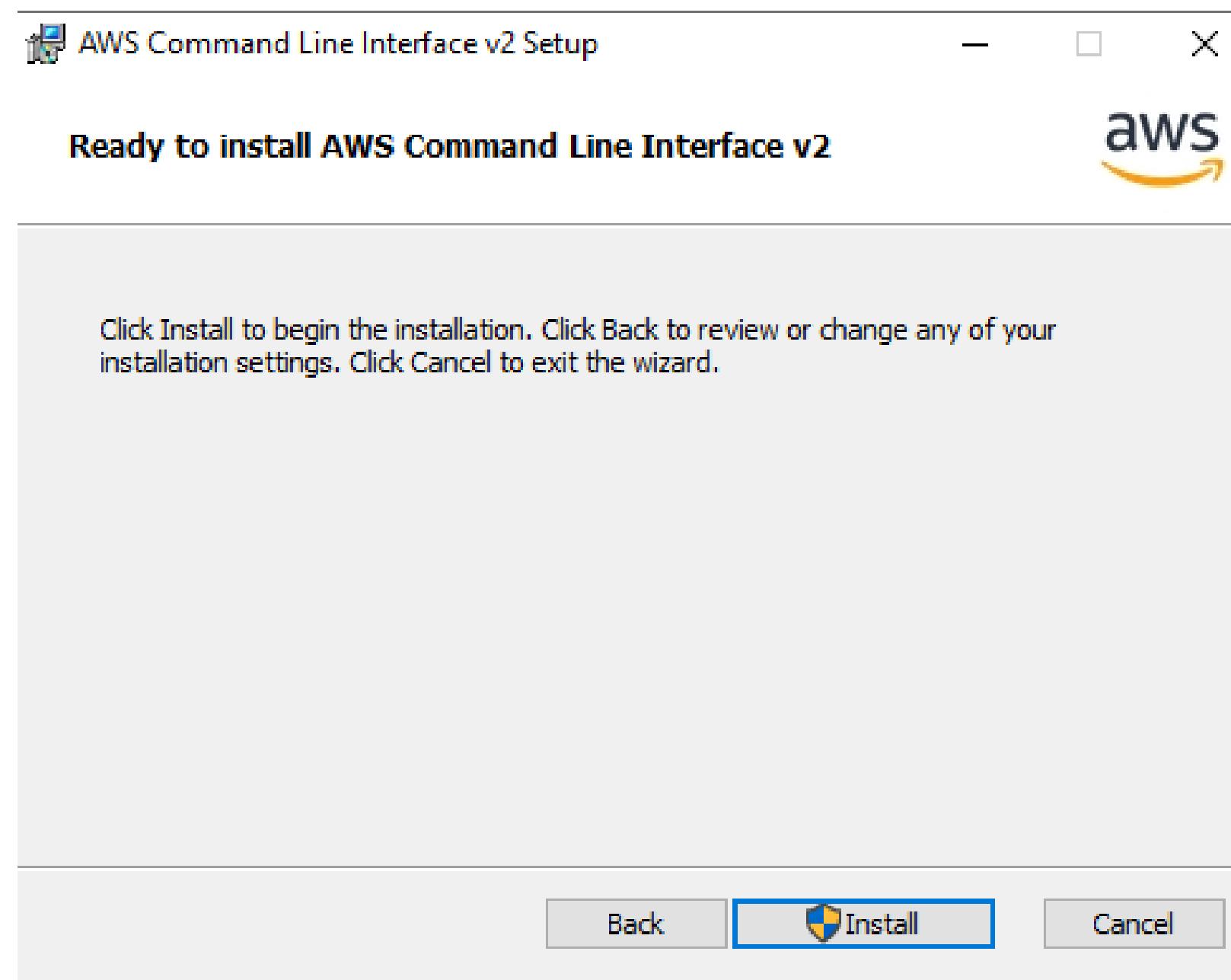
Step-3 : Attached s3 bucket policy to s3 bucket for public access.

```
{  
  "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 through command prompt

The screenshot shows the AWS IAM Access Keys page and two terminal windows for AWS CLI configuration.

IAM Access Keys Page:

- Shows a table of access keys, including one active key (AKIAZY4K) created 33 days ago, last used 29 minutes ago in us-east-1 for s3 service.
- A red box highlights the "Create access key" button in the top right corner.

Terminal 1 (Administrator: Command Prompt):

```
C:\Program Files\Amazon\AWSCLIv2>aws --version
aws-cli/2.17.38 Python/3.11.9 Windows/10 exe/AMD64
```

Terminal 2 (Administrator: Command Prompt):

```
C:\Program Files\Amazon\AWSCLIv2>aws --version
aws-cli/2.17.38 Python/3.11.9 Windows/10 exe/AMD64
```

Terminal 3 (Administrator: Command Prompt):

```
C:\Program Files\Amazon\AWSCLIv2>aws configure
AWS Access Key ID [None]: AKIAZY4K
AWS Secret Access Key [None]: SHgT0sTTamu04
Default region name [None]: us-east-1
Default output format [None]:
```

Terminal 4 (Administrator: Command Prompt):

```
C:\Program Files\Amazon\AWSCLIv2>aws s3 ls
2024-08-26 20:28:53 vmmigration-on-prem-server
```

- 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

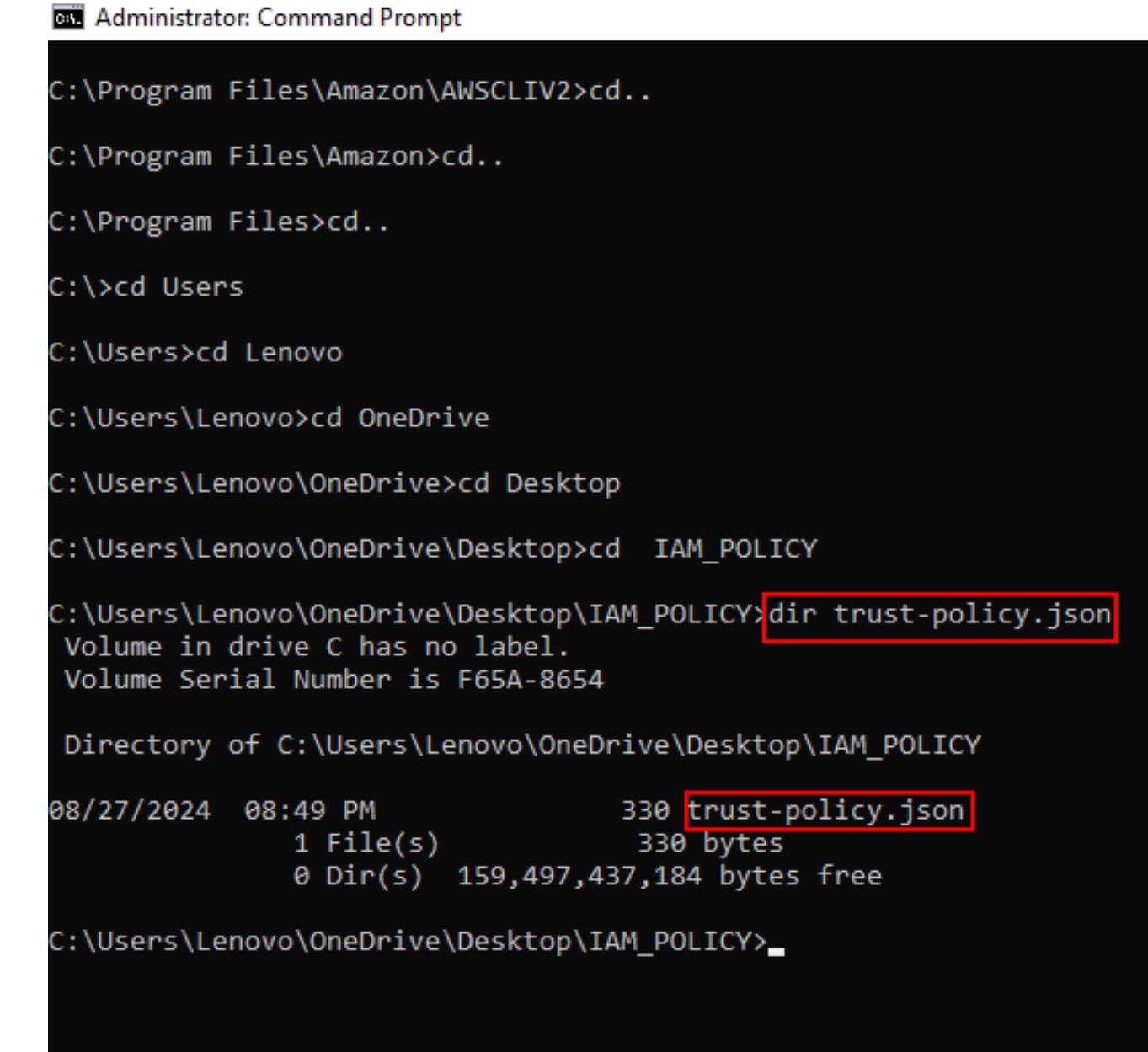
Step-6 : Now create a IAM role for importing the on-prem vm to AWS

- 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"  
                }  
            }  
        }  
    ]  
}
```



The screenshot shows an Administrator Command Prompt window with the following command history and output:

```
C:\Program Files\Amazon\AWSCLIV2>cd..  
C:\Program Files\Amazon>cd..  
C:\Program Files>cd..  
C:\>cd Users  
C:\Users>cd Lenovo  
C:\Users\Lenovo>cd OneDrive  
C:\Users\Lenovo\OneDrive>cd Desktop  
C:\Users\Lenovo\OneDrive\Desktop>cd IAM_POLICY  
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) 159,497,437,184 bytes free  
C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>
```

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

```
Administrator: Command Prompt

C:\Program Files\Amazon\AWSCLIV2>cd..

C:\Program Files\Amazon>cd..

C:\Program Files>cd..

C:\>cd Users

C:\Users>cd Lenovo

C:\Users\Lenovo>cd OneDrive

C:\Users\Lenovo\OneDrive>cd Desktop

C:\Users\Lenovo\OneDrive\Desktop>cd IAM_POLICY

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)  159,497,437,184 bytes free

C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>
```

- 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"
            }
          }
        }
      ]
    }
  }
}
```

IAM > Roles

Roles (66) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by other AWS services or by users and applications.

Q vm X 1 match

<input type="checkbox"/>	Role name	▲	Trusted entities
<input type="checkbox"/>	vmimport		AWS Service: vmie

Step-8: Attach the required policy to IAM role

The screenshot shows the AWS IAM service interface. The top navigation bar includes links for Lightsail, CloudFormation, Simple Notification Service, Elastic Beanstalk, CloudWatch, Simple Queue Service, Route 53, DynamoDB, CloudFront, Lambda, CloudTrail, IAM, and Amazon Open. The user is signed in as Aman Dandale.

The current view is the 'vmimport' role's details page. The 'Summary' section displays the creation date (August 27, 2024, 21:36 (UTC+05:30)), ARN (arn:aws:iam::671916260115:role/vmimport), and maximum session duration (1 hour). The 'Edit' button is located in the top right corner of this section.

The 'Permissions' tab is selected, showing one attached policy: 'AdministratorAccess'. This policy is highlighted with a red border. Other tabs available include Trust relationships, Tags, Access Advisor, and Revoke sessions.

The 'Permissions policies (1)' section lists the attached policy. It includes a search bar, a filter dropdown set to 'All types', and a page indicator showing 1 of 1. The policy listed is 'AdministratorAccess', which is an AWS managed - job function policy.

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.

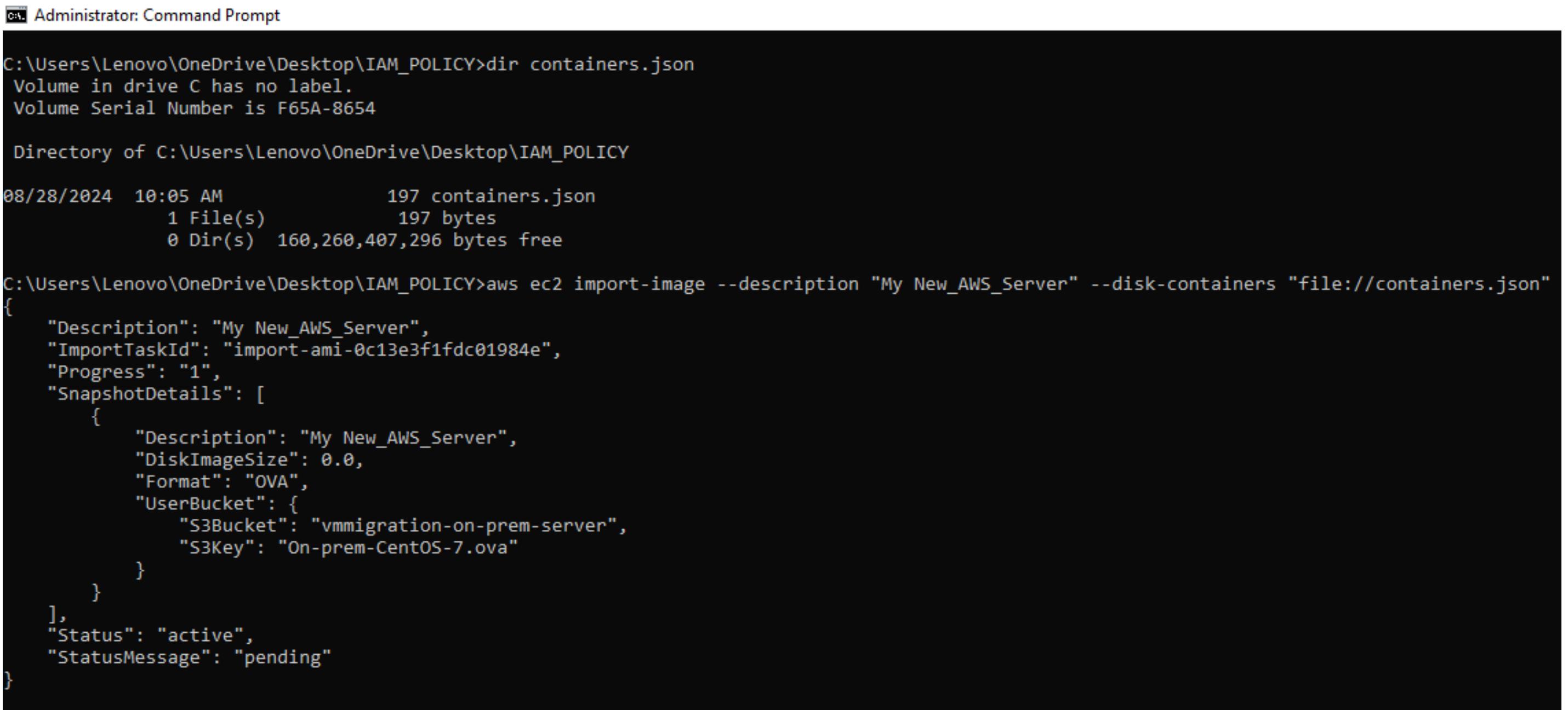
```
[  
  {  
    "Description": "My Server OVA",  
    "Format": "ova",  
    "UserBucket": {  
      "S3Bucket": "my-import-bucket",  
      "S3Key": "vms/my-server-vm.ova"  
    }  
  }  
]
```

```
[  
  {  
    "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"
```



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command entered is:

```
C:\Users\Lenovo\OneDrive\Desktop\IAM_POLICY>aws ec2 import-image --description "My New_AWS_Server" --disk-containers "file://containers.json"
```

The output shows the creation of a file named "containers.json" in the current directory:

```
{  
  "Description": "My New_AWS_Server",  
  "ImportTaskId": "import-ami-0c13e3f1fdc01984e",  
  "Progress": "1",  
  "SnapshotDetails": [  
    {  
      "Description": "My New_AWS_Server",  
      "DiskImageSize": 0.0,  
      "Format": "OVA",  
      "UserBucket": {  
        "S3Bucket": "vmmigration-on-prem-server",  
        "S3Key": "On-prem-CentOS-7.ova"  
      }  
    }  
  ],  
  "Status": "active",  
  "StatusMessage": "pending"  
}
```

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 --import-task-ids import-ami-0c13e3f1fdc01984e
{
    "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

The screenshot shows the AWS Management Console interface for the Amazon Machine Images (AMIs) service. The top navigation bar includes the AWS logo, a search bar, and a user dropdown for "Aman Dandale". The "N. Virginia" region is selected. The main menu bar lists various services: EC2, VPC, S3, RDS, Lightsail, CloudFormation, Simple Notification Service, Elastic Beanstalk, CloudWatch, Simple Queue Service, Route 53, DynamoDB, CloudFront, Lambda, CloudTrail, IAM, and Amazon OpenSea.

The main content area displays the "Amazon Machine Images (AMIs) (1/1)" list. A single item is listed:

Name	AMI ID	Source	Owner	Visibility	Status	Creation date	
My Migrated server	import-ami-0c13e3f1fdc01984e	ami-01bd262df79c3fc8f	671916260115/import-ami-0c13e3f1fdc01984e	671916260115	Private	Available	2024/08/28 10:32 GMT+5:30

Below the list, a modal window provides detailed information about the selected AMI:

AMI ID: ami-01bd262df79c3fc8f (My Migrated server)

Details | Permissions | Storage | Tags

AMI ID ami-01bd262df79c3fc8f (My Migrated server)	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name import-ami-0c13e3f1fdc01984e	Owner account ID 671916260115	Architecture x86_64	Usage operation RunInstances
Root device name /dev/sda1	Status Available	Source 671916260115/import-ami-0c13e3f1fdc01984e	Virtualization type hvm
Boot mode -	State reason -	Creation date Wed Aug 28 2024 10:32:07 GMT+0530 (India Standard Time)	Kernel ID -
Description AWS-VMImport service: Linux - CentOS Linux 7 (Core) - 3.10.0-1160.80.1.el7.x86_64	Product codes -	RAM disk ID -	Deprecation time -

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 oea3fef3-80a7-4778-9d8c-1coc6EXAMPLE --disk-containers "file://C:\import\containers.json"
```

The screenshot shows the AWS EC2 console with the 'Amazon Machine Images (AMIs)' page. A single AMI named 'My Migrated server' is listed. The 'Storage' tab is selected in the navigation bar. Under 'Root device details', it shows a root device named '/dev/sda1' with an EBS type. In the 'Block devices' section, there are two entries: one for '/dev/sda1' (Volume size: 20 GiB, Volume type: gp2) and another for '/dev/sdf' (Volume size: 5 GiB, Volume type: gp2). Both entries in the 'Encrypted' column are marked as 'No'. A red box highlights the 'Encrypted' column header and the first row of the table.

Device ID	Device name	Volume size (GiB)	Volume type	Encrypted	Delete on termination	KMS key ID	Outpost ID
snap-02d07ec8f179f402f	/dev/sda1	20	gp2	No	No	-	-
snap-09978c53efd258304	/dev/sdf	5	gp2	No	No	-	-

Step-14 Now launch EC-2 from our AMI

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

The screenshot shows the AWS Management Console with the EC2 service selected. The breadcrumb navigation indicates: EC2 > Instances > Launch an instance > AMIs. The main title is "Choose an Amazon Machine Image (AMI)". A descriptive text explains that an AMI is a template containing software configuration required to launch an instance. It lists categories for selecting AMIs: Quick Start AMIs (0), My AMIs (1), AWS Marketplace AMIs (11203), and Community AMIs (0). The "My AMIs" tab is selected, showing a single result: "ami-01bd262df79c3fc8f". This result is highlighted with a red box. Below the search bar, there are filters for "Refine results": "Clear all filters", "Owner" (with "Owned by me" checked and "Shared with me" unchecked), and "OS category". The detailed view for the selected AMI shows its name, owner, and tags. A "Select" button is visible on the right.

Selected AMI: (ami-066784287e358dad1) (Quick Start AMIs)

Search: ami-01bd262df79c3fc8f

Quick Start AMIs (0)
Commonly used AMIs

My AMIs (1)
Created by me

AWS Marketplace AMIs (11203)
AWS & trusted third-party AMIs

Community AMIs (0)
Published by anyone

Refine results: ami-01bd262df79c3fc8f (1 filtered, 1 unfiltered)

Clear all filters

Owner: Owned by me Shared with me

OS category

import-ami-0c13e3f1fdc01984e
ami-01bd262df79c3fc8f (My Migrated server)
AWS-VMImport service: Linux - CentOS Linux 7 (Core) - 3.10.0-1160.80.1.el7.x86_64
OwnerAlias: -Platform: Other LinuxArchitecture: x86_64Owner: 671916260115Publish date: 2024-08-28Root device type: ebsVirtualization: hvmENA enabled: Yes
Tags:
Name: My Migrated server

Select

Step-15 Now launch EC-2 from our AMI

The screenshot shows the AWS EC2 console interface. On the left, the 'Application and OS Images (Amazon Machine Image)' section displays details for an imported AMI named 'import-ami-0c13e3f1fdc01984e'. This AMI is based on AWS-VMImport service: Linux - CentOS Linux 7 (Core) - 3.10.0-1160.80.1.el7.x86_64. It has an Image ID of 'ami-01bd262df79c3fc8f'. The catalog information shows it was published on 2024-08-28T05:02:07.000Z by 'My AMIs'. On the right, the 'Summary' section of the launch wizard is shown, where the user has selected 1 instance. The 'Software Image (AMI)' is set to 'import-ami-0c13e3f1fdc01984e' and the 'Virtual server type (instance type)' is 't2.micro'. The 'Launch instance' button is highlighted with a red box.

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

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

AMI from catalog Recents My AMIs Quick Start

Name: import-ami-0c13e3f1fdc01984e

Description: AWS-VMImport service: Linux - CentOS Linux 7 (Core) - 3.10.0-1160.80.1.el7.x86_64

Image ID: ami-01bd262df79c3fc8f

Catalog: My AMIs Published: 2024-08-28T05:02:07.000Z Architecture: x86_64 Virtualization: hvm Root device type: ebs ENA Enabled: Yes

Summary

Number of instances: 1

Software Image (AMI): import-ami-0c13e3f1fdc01984e
ami-01bd262df79c3fc8f

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 2 volume(s) - 25 GiB

Cancel Launch instance Review commands

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

The screenshot shows the AWS Management Console with the EC2 service selected. The main view displays a single EC2 instance named "My-VM-Migrated-server-2" with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
My-VM-Migrated-server-2	i-004f543d18336f25b	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-23-22-234-163.co...

Below the main table, a detailed view for the selected instance is shown:

Attribute	Value	Notes
Instance auto-recovery	Default	
AMI Launch index	0	
Credit specification	standard	
Usage operation	RunInstances	
Lifecycle	normal	Without key-pair
Key pair assigned at launch	-	(This field is highlighted with a red box)
Kernel ID	-	
RAM disk ID	-	
Stop-hibernate behavior	Disabled	
State transition reason	-	
State transition message	-	
Owner	671916260115	

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

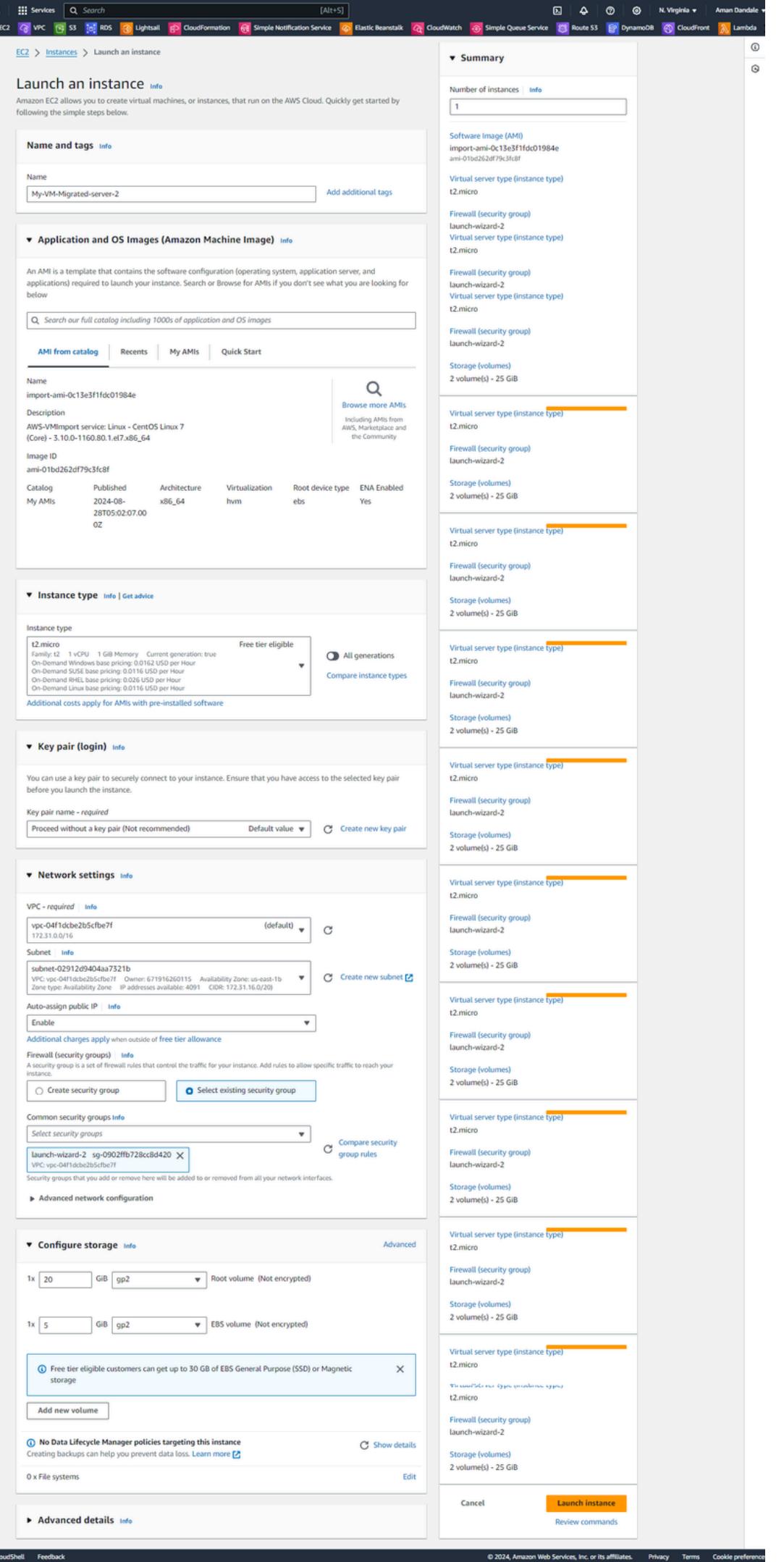
The screenshot shows the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and various service links like VPC, S3, RDS, CloudFormation, Simple Notification Service, Elastic Beanstalk, CloudWatch, Simple Queue Service, Route 53, DynamoDB, CloudFront, Lambda, CloudTrail, IAM, and Amazon OpenSea. The user is signed in as 'Aman Dandale'. The main content area displays the 'Instances' section with one item listed:

Instance ID	Name	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IP
i-0e2454e9516863978	My-VM-Migrated-server	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-18-206-252-189.co...	18.206.252.189	-	-

Below the table, the details for the selected instance (i-0e2454e9516863978) are shown. The 'Details' tab is active, displaying the following information:

Instance summary	Public IPv4 address	Private IPv4 addresses
Instance ID: i-0e2454e9516863978 (My-VM-Migrated-server)	18.206.252.189	172.31.87.236
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-18-206-252-189.compute-1.amazonaws.com
Hostname type: IP name: ip-172-31-87-236.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-87-236.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: IPv4 (A)	Instance type: t2.micro	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address: 18.206.252.189 [Public IP]	VPC ID: vpc-04f1dcbe2b5cfbe7f	

לעוני מיתר עלי-באי || אל-אלט



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) Updated 29 minutes ago

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 Xdmcp RDP VNC FTP 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:

- X11-Forwarding
- Compression
- Remote environment: Interactive shell
- Execute command:
- Do not exit after command ends
- SSH-browser type: SFTP protocol
- Use private key Without key-pair
- Follow SSH path (experimental)
- Expert SSH settings
- Execute macro at session start: <none>

OK Cancel

```

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
drwxr-xr-x  3 root root   18 Apr  9 2022 .config
-rw-r--r--  1 root root 100 Dec 29 2013 .cshrc
drwxr-xr-x  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 .lessht
-rw-rw-rwx  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  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

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 20:18:55 on ttys1
[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 192.168.04.196  netmask 255.255.255.0  broadcast 192.168.04.255
        inet6 2401:4980:5618:98ed:2ad:58d2:c143:cdf4  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::703f:b1d1:cadf:3f7e  prefixlen 64  scopeid 0x20<link>
      ether 00:00:27: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 ~]#
```

AWS EC-2 Instance (After migration to AWS)

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:eff: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
[root@server1 ~]#
```

Comparison: On-prem-VM-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)

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

Number of vCPUs
1

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

Operating System specifications of both server

```
11. 18.206.252.189 (root) × +  
[root@server1 ~]# cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"  
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"  
  
[root@server1 ~]#
```

EC-2 specifications

```
Server-1 [Running] - Oracle VM VirtualBox  
[root@server1 ~]# cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"  
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"  
  
[root@server1 ~]#
```

VM-server specifications

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.

Your Data Migration Roadmap



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,

MIGRATION STRATEGIES

- 1 RETIRE
- 2 RETAIN
- 3 RELOCATE ★
- 4 REHOST
- 5 REPURCHASE
- 6 REPLATFORM
- 7 REFACTOR



tutorials dojo

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.



On Premise to
Cloud Migration