AWS Capstone Project

Problem Statement

ABC is India based entertainment production company with a focus on Northeast and East Indian cinema. They wanted a highly available and reliable storage solution for their onprem data. The client wanted the data to be reliable, highly available, secure, and persistent and also wanted to have data on the storage to be accessible from all the servers.

The issues with the existing infrastructure are mentioned below:

1) They were using NAS storage which had limitations in scalability.

2) The client wanted a complete application to run in Aws cloud with storage in the cloud to keep his files in sync with servers so that he has centralized storage of data

3) There were security issues in the existing infrastructure and no encryption at rest or transit.

4) Manual intervention was needed to change the storage type and transfer files to infrequent storage.

5) The client is unable to scale up the infrastructure due to high capital costs for new hardware.

6) Need for low-cost storage options for both frequent and infrequent data.

7) Highly available, secure, and persistent shared File system in AWS cloud with EFS.

We will make use of Amazon EFS.

Amazon Elastic File System (EFS) is ideal for ABC's storage needs:

1. Scalability: EFS automatically scales with usage, unlike NAS.

2. Centralized Cloud Storage: Accessible by multiple servers, ensuring data consistency.

3. Security: Provides built-in encryption at rest and in transit.

4. Automated Storage Management: Automatically transitions files to cost-effective storage tiers.

5. Cost-Efficiency: Eliminates hardware costs with a pay-as-you-go model.

6. Low-Cost Options: Offers tiered storage for frequent and infrequent data.

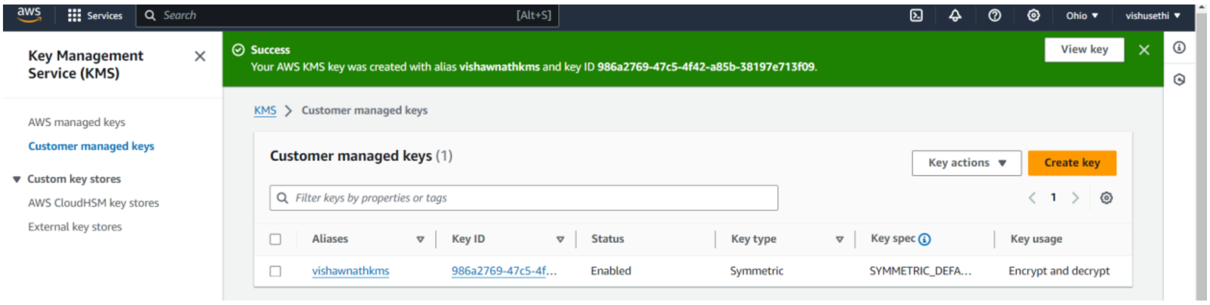
7. High Availability: Stores data across multiple Availability Zones for reliability.

EFS integrates seamlessly with AWS, providing scalable, secure, and cost-effective storage for ABC's operations.

Step 1: Create EFS

Created KMS in the process of creating EFS

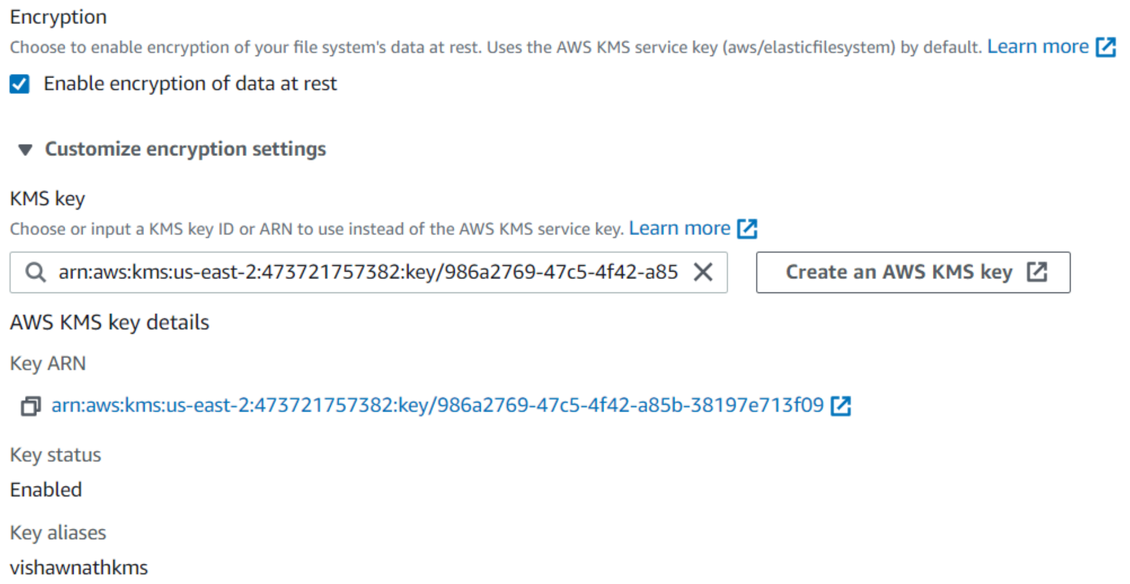
KMS ensures data security and compliance by managing encryption keys and controlling access to them.



Success
Your AWS KMS key was created with alias **vishawnathkms** and key ID **986a2769-47c5-4f42-a85b-38197e713f09**.

Customer managed keys (1)

☐	Aliases	Key ID	Status	Key type	Key spec	Key usage
☐	vishawnathkms	986a2769-47c5-4f...	Enabled	Symmetric	SYMMETRIC_DEFA...	Encrypt and decrypt



Encryption

Choose to enable encryption of your file system's data at rest. Uses the AWS KMS service key (aws/elasticfilesystem) by default. Learn more

☑ Enable encryption of data at rest

▼ Customize encryption settings

KMS key

Choose or input a KMS key ID or ARN to use instead of the AWS KMS service key. Learn more

arn:aws:kms:us-east-2:473721757382:key/986a2769-47c5-4f42-a85

AWS KMS key details

Key ARN

arn:aws:kms:us-east-2:473721757382:key/986a2769-47c5-4f42-a85b-38197e713f09

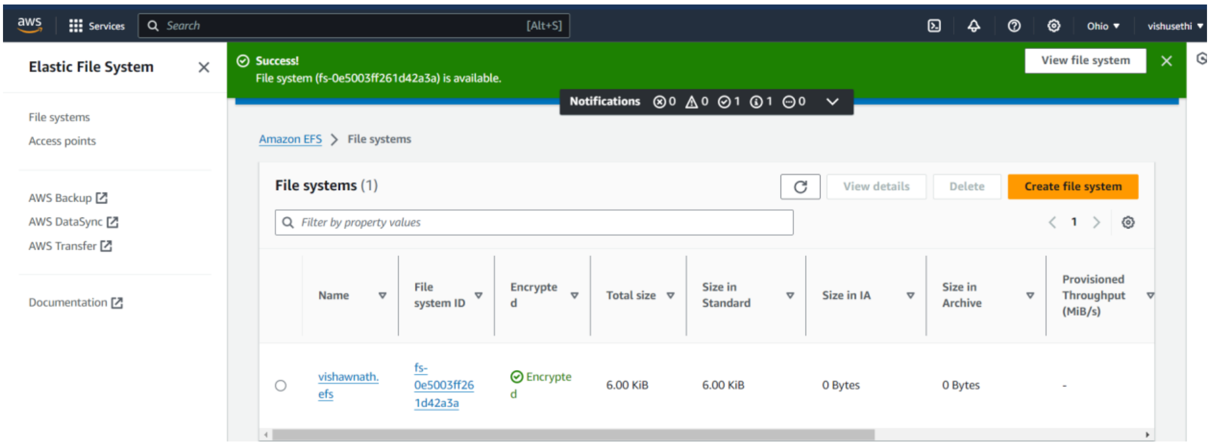
Key status

Enabled

Key aliases

vishawnathkms

EFS Created



Success!
File system (fs-0e5003ff261d42a3a) is available.

File systems (1)

	Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive	Provisioned Throughput (MiB/s)
○	vishawnath.efs	fs-0e5003ff261d42a3a	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes	-

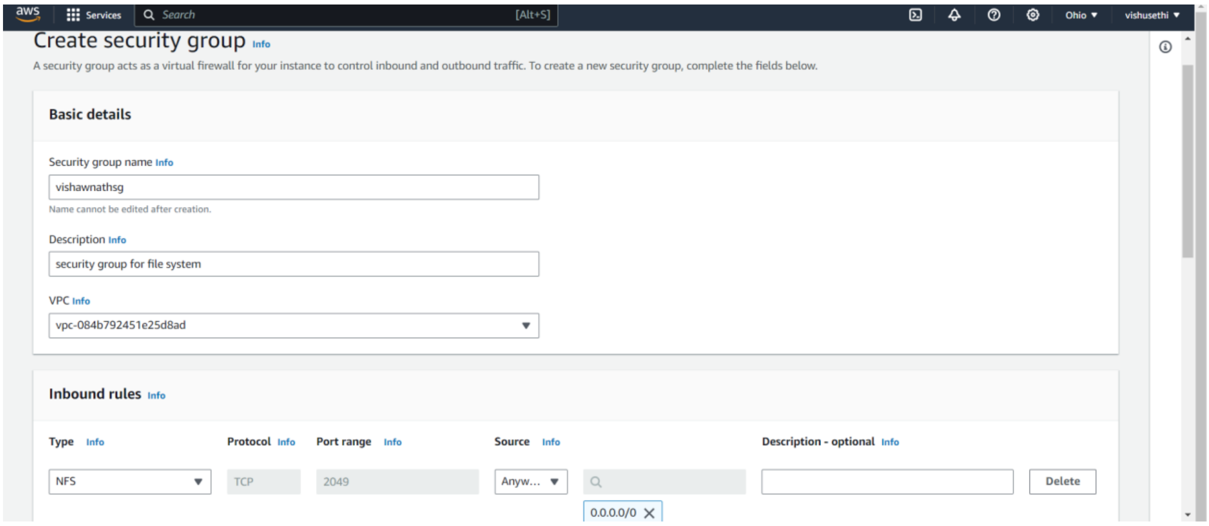
Creating security group and adding inbound rule in EC2

Security Groups in Amazon EC2 control traffic to instances.

Inbound Rules

- Allow specified incoming traffic

- Set rules for protocols, ports, and source IPs



Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: vishawnathsg

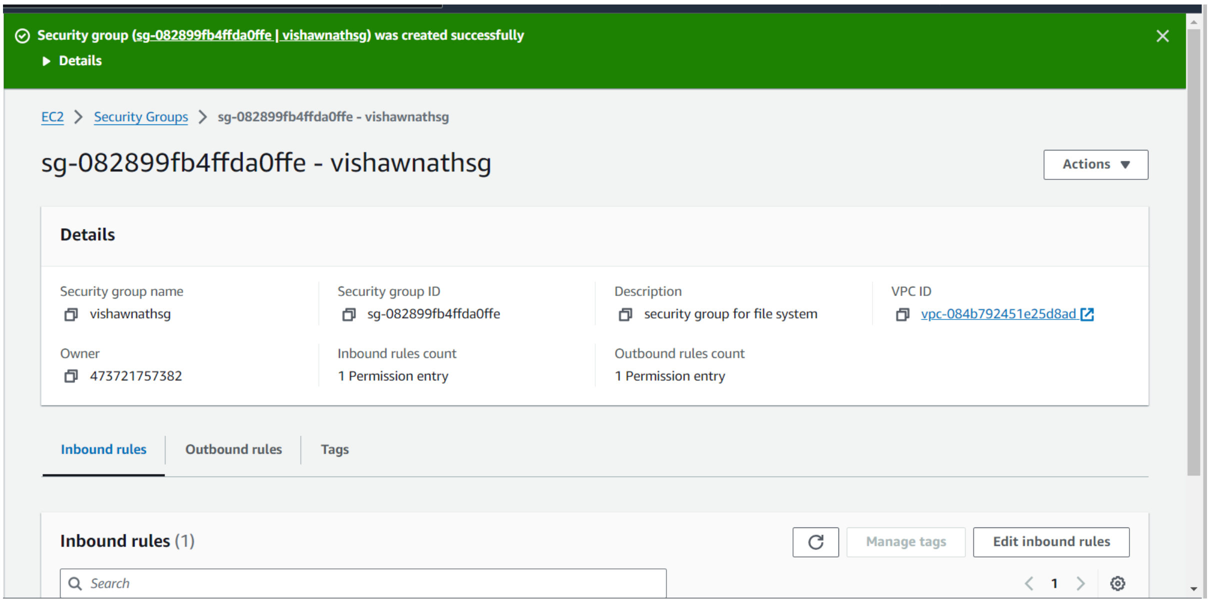
Name cannot be edited after creation.

Description: security group for file system

VPC: vpc-084b792451e25d8ad

Inbound rules

Type	Protocol	Port range	Source	Description - optional
NFS	TCP	2049	Anyw... 0.0.0.0/0	



Security group (sg-082899fb4ffda0ffe | vishawnathsg) was created successfully

Details

EC2 > Security Groups > sg-082899fb4ffda0ffe - vishawnathsg

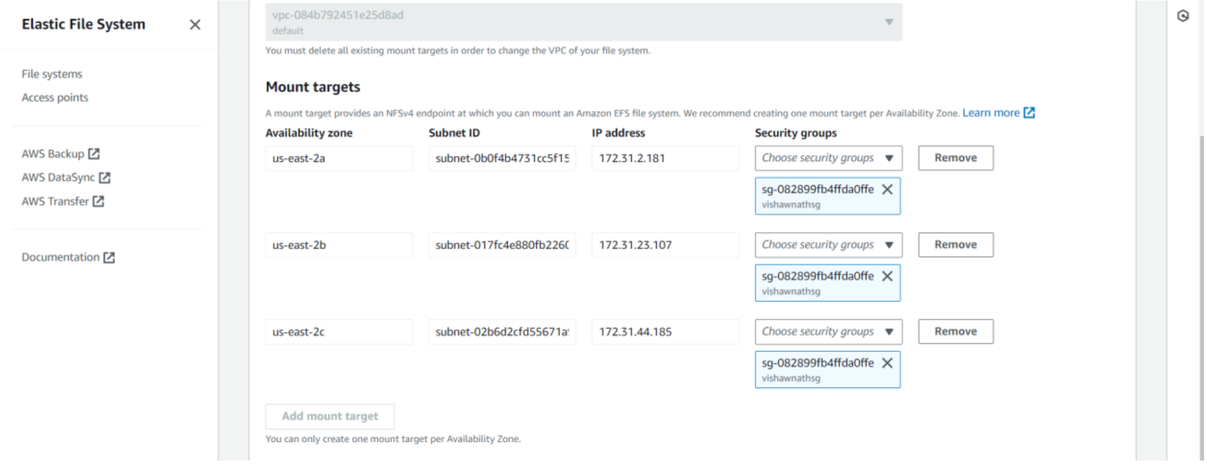
sg-082899fb4ffda0ffe - vishawnathsg

Details

Security group name	Security group ID	Description	VPC ID
vishawnathsg	sg-082899fb4ffda0ffe	security group for file system	vpc-084b792451e25d8ad
Owner	**Inbound rules count**	**Outbound rules count**	
473721757382	1 Permission entry	1 Permission entry	

Inbound rules (1)

Removing default security group and associating newly created security group to EFS



vpc-084b792451e25d8ad
default

You must delete all existing mount targets in order to change the VPC of your file system.

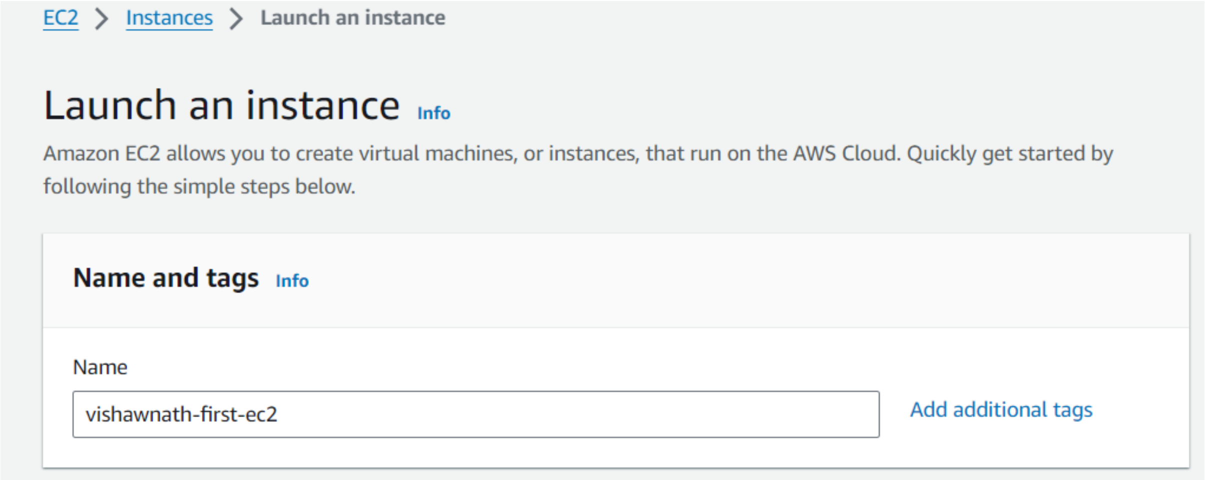
Mount targets

A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. Learn more

Availability zone	Subnet ID	IP address	Security groups	
us-east-2a	subnet-0b0f4b4731cc5f15	172.31.2.181	Choose security groups sg-082899fb4ffda0ffe vishawnathsg	Remove
us-east-2b	subnet-017fc4e880fb226(	172.31.23.107	Choose security groups sg-082899fb4ffda0ffe vishawnathsg	Remove
us-east-2c	subnet-02b6d2cfd55671a'	172.31.44.185	Choose security groups sg-082899fb4ffda0ffe vishawnathsg	Remove

You can only create one mount target per Availability Zone.

Creating first ec2 instance with "amazon Linux image" and "T2. micro instance"



EC2 > Instances > Launch an instance

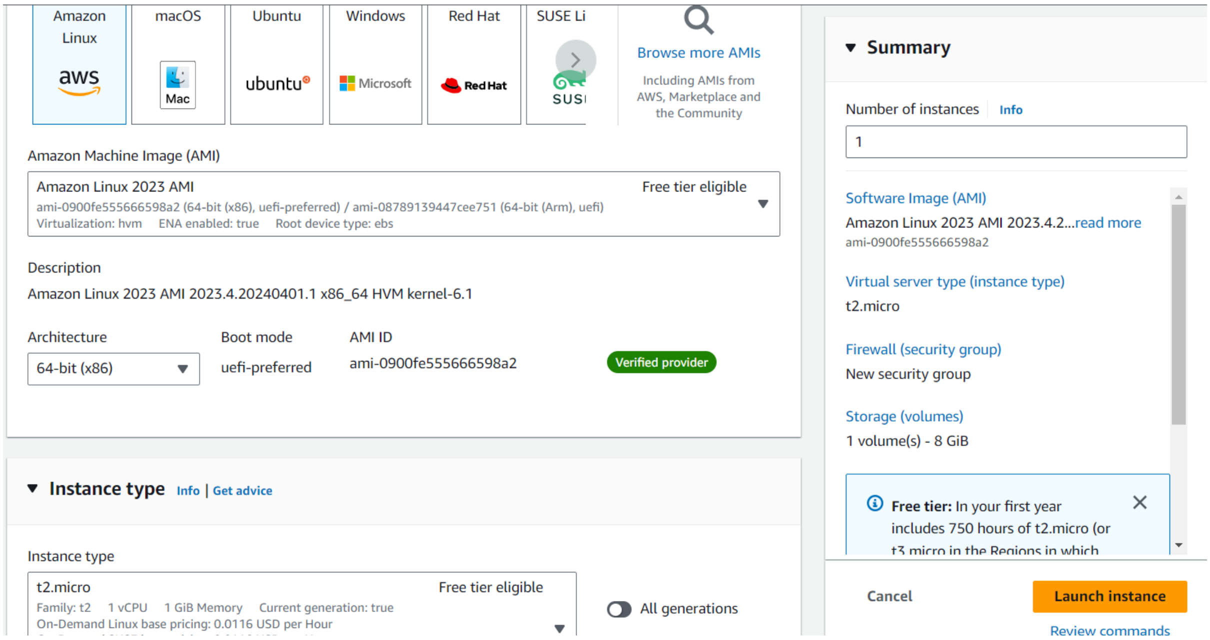
Launch an instance Info

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: vishawnath-first-ec2

Add additional tags



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0900fe555666598a2 (64-bit (x86), uefi-preferred) / ami-08789139447cee751 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 AMI 2023.4.20240401.1 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	
64-bit (x86)	uefi-preferred	ami-0900fe555666598a2	Verified provider

Instance type Info | Get advice

Instance type
t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour

○ All generations

Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-0900fe555666598a2

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

Key pair creation

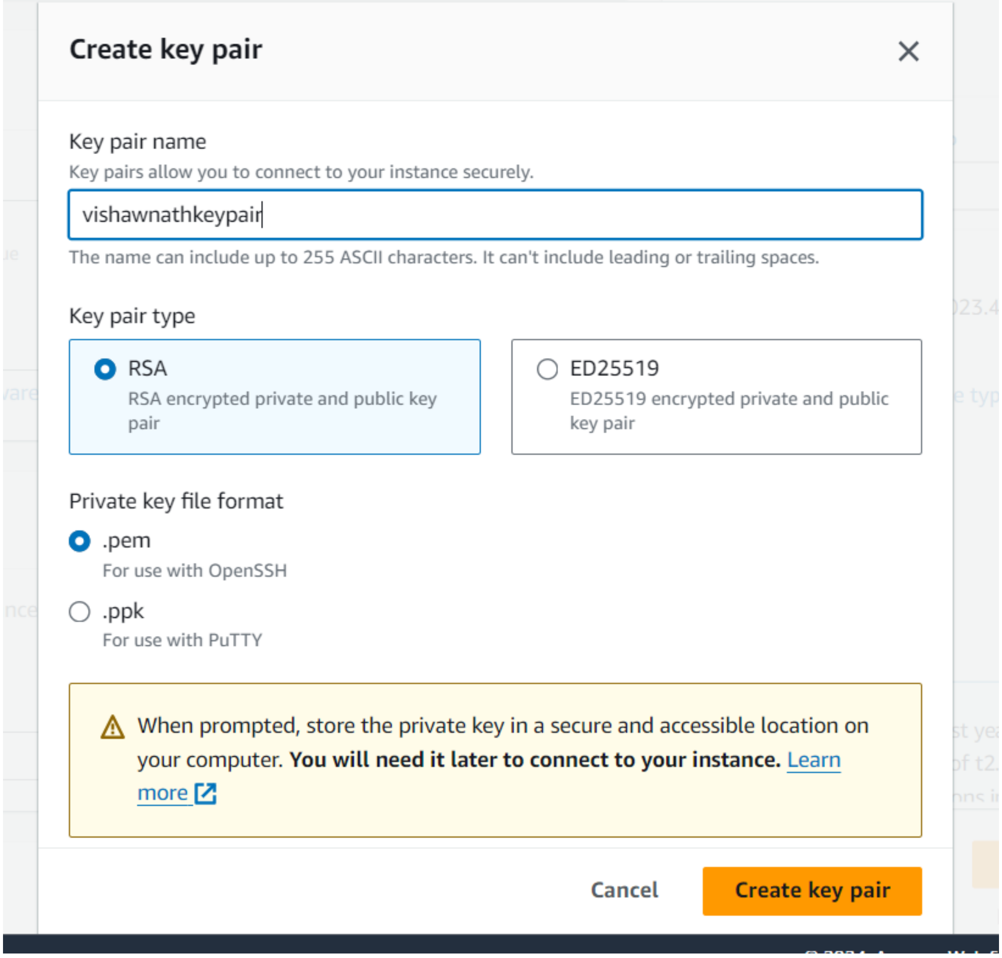
Use of key pair:

- Authenticate access to instances securely.

- Instances use a public key for access.

- Only holders of the private key can access instances.

- Enable encrypted communication for data security.



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

vishawnathkeypair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

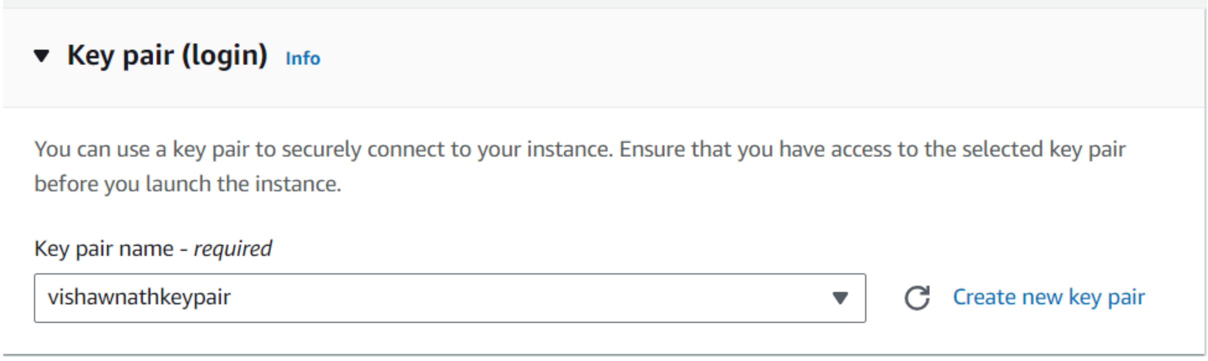
○ ED25519
ED25519 encrypted private and public key pair

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more



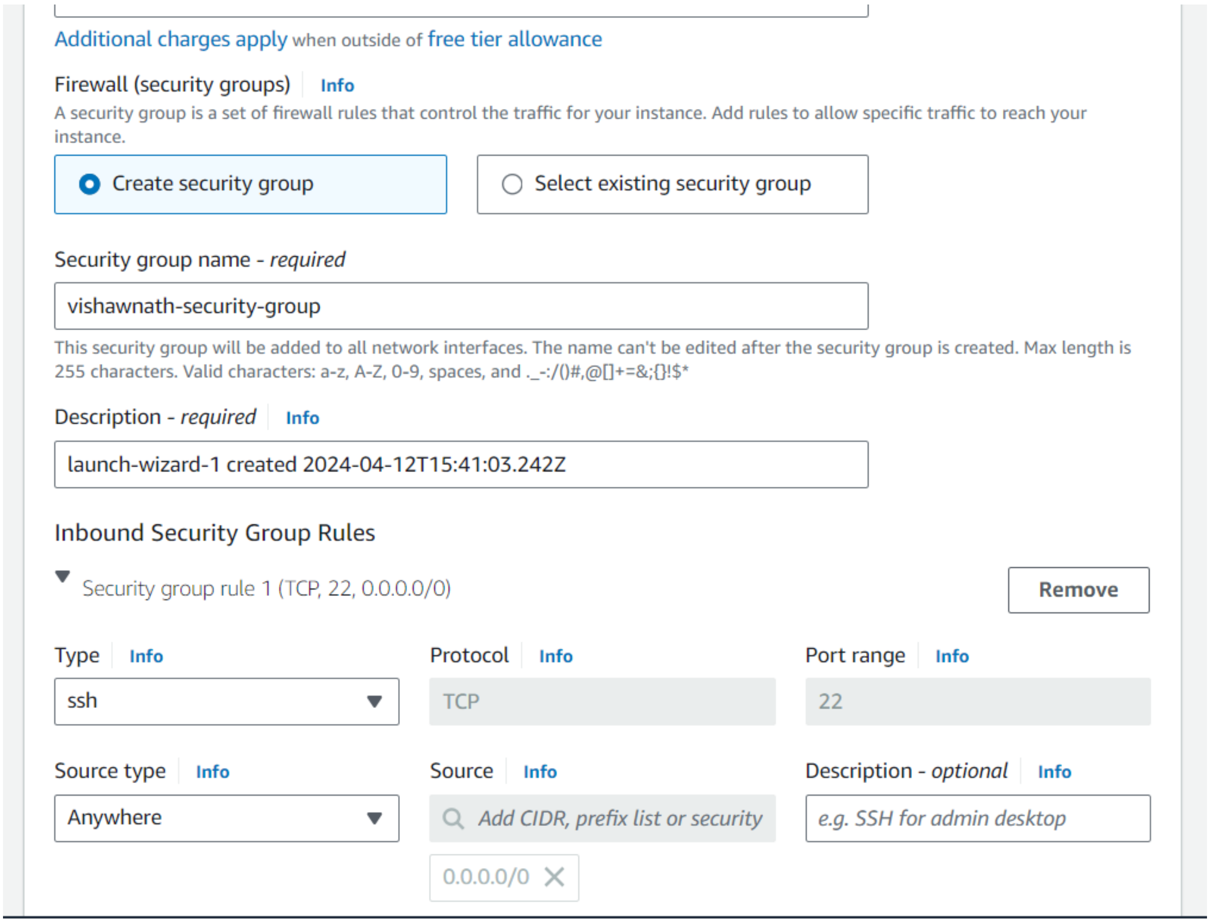
Key pair (login) Info

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*

vishawnathkeypair

Created new security group while creating a first EC2 instance



Additional charges apply when outside of free tier allowance

Firewall (security groups)

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

Security group name - *required*

vishawnath-security-group

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required*

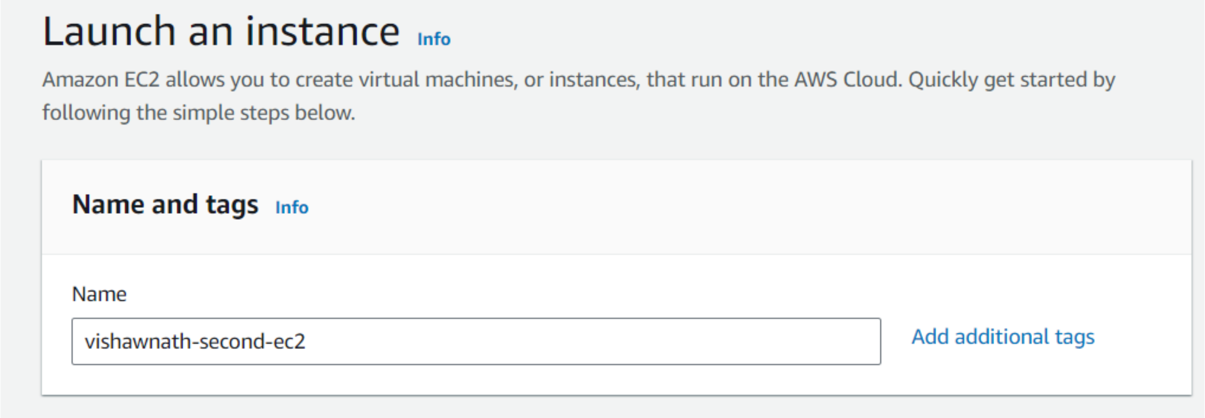
launch-wizard-1 created 2024-04-12T15:41:03.242Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22
Source type	Source	Description - *optional*
Anywhere	*Add CIDR, prefix list or security* 0.0.0.0/0 ✕	*e.g. SSH for admin desktop*

Creating second EC2 instance with “amazon Linux image” and “T2. micro instance”

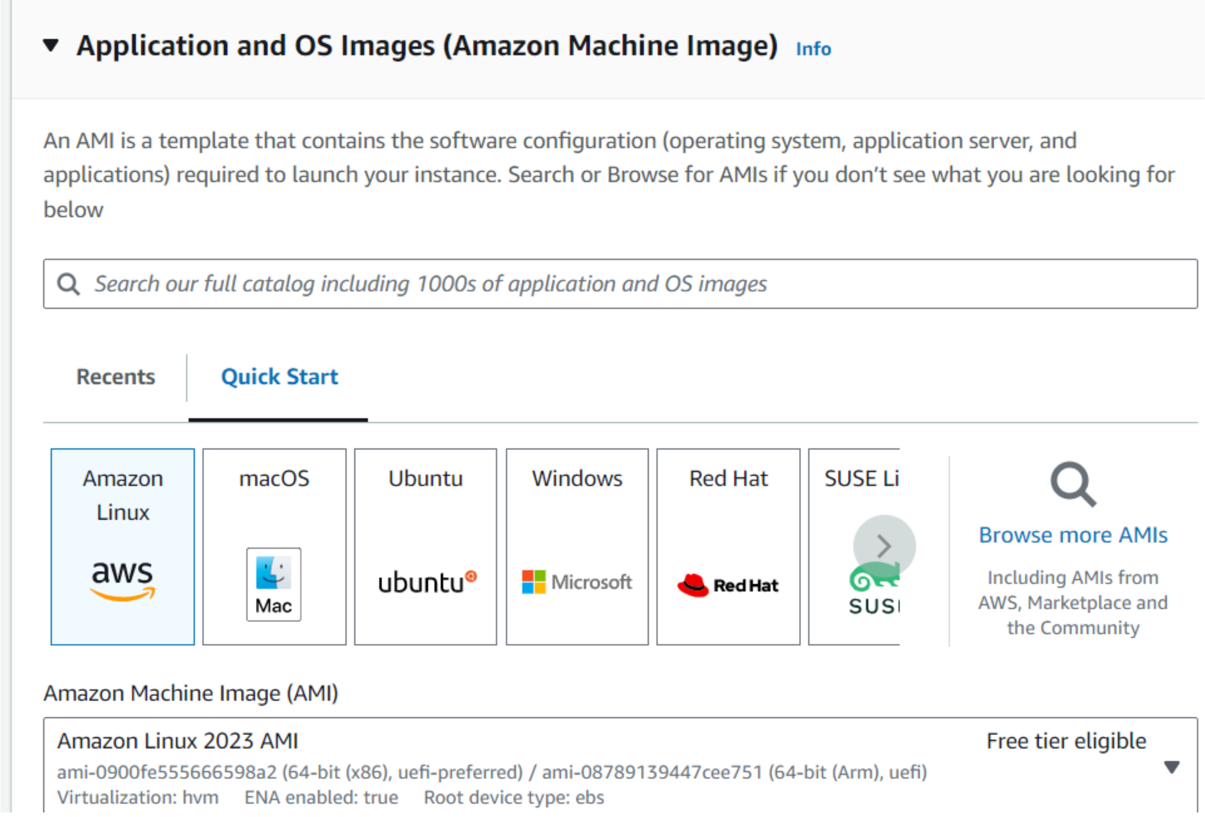


Launch an instance

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

Name: vishawnath-second-ec2



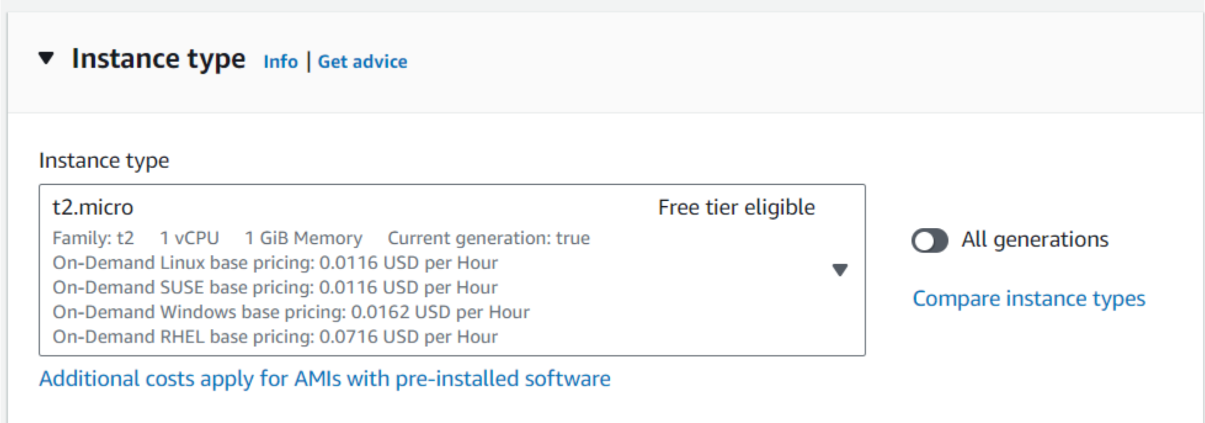
▼ Application and OS Images (Amazon Machine Image)

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

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0900fe555666598a2 (64-bit (x86), uefi-preferred) / ami-08789139447cee751 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs



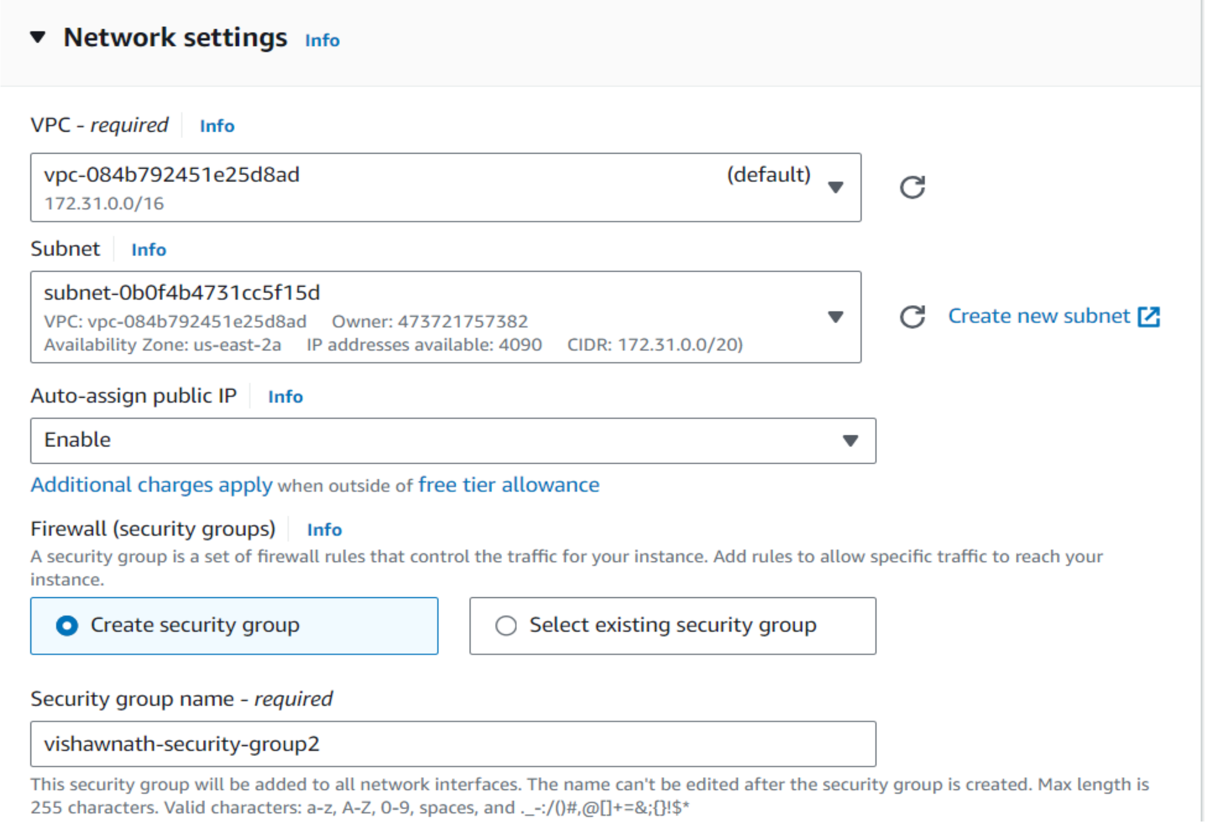
▼ Instance type

Instance type

t2.micro — Free tier eligible
Family: t2   1 vCPU   1 GiB Memory   Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour

Additional costs apply for AMIs with pre-installed software

Created new security group while creating a second EC2 instance



▼ **Network settings** Info

VPC - *required* Info

vpc-084b792451e25d8ad (default)
172.31.0.0/16

Subnet Info

subnet-0b0f4b4731cc5f15d
VPC: vpc-084b792451e25d8ad Owner: 473721757382
Availability Zone: us-east-2a IP addresses available: 4090 CIDR: 172.31.0.0/20)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

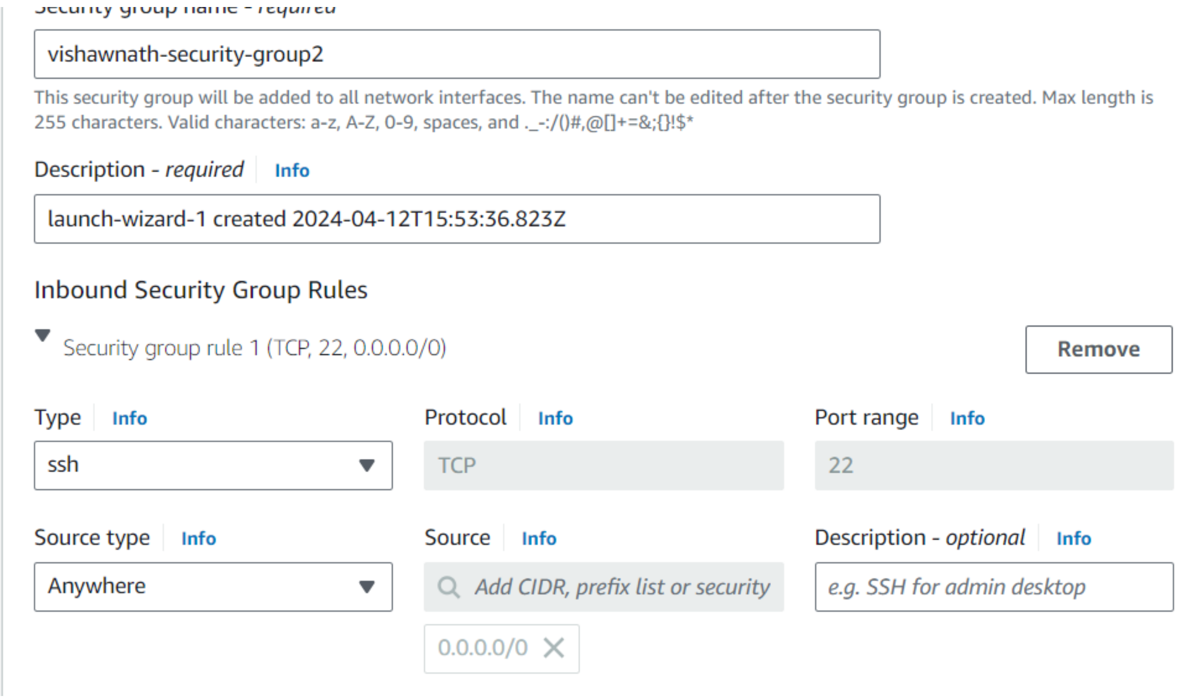
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

vishawnath-security-group2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*



Security group name - *required*

vishawnath-security-group2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

launch-wizard-1 created 2024-04-12T15:53:36.823Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type Info: ssh

Protocol Info: TCP

Port range Info: 22

Source type Info: Anywhere

Source Info: Add CIDR, prefix list or security
0.0.0.0/0 ✕

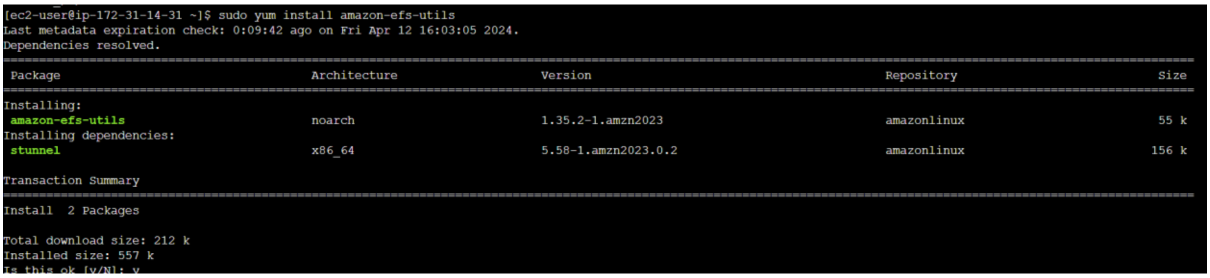
Description - *optional* Info: e.g. SSH for admin desktop

Connected to first EC2 instance and ran following commands:

sudo yum install amazon-efs-utils- to manually create the EFS client

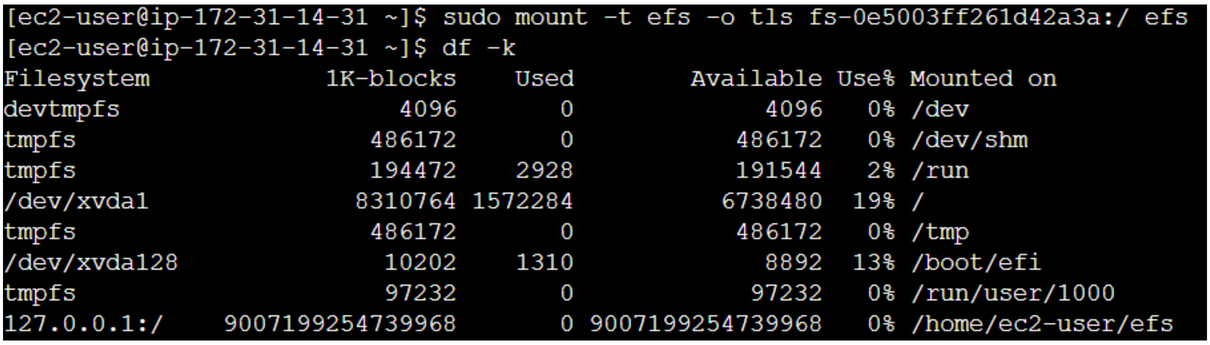
mkdir efs : To create directory

Perform following steps to mount file system on /home/ec2-user/efs directory



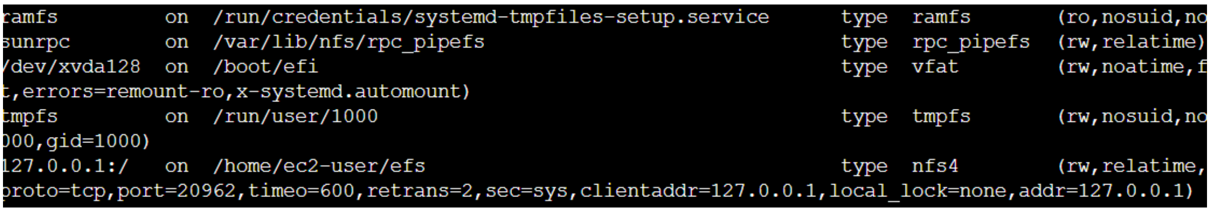
```
[ec2-user@ip-172-31-14-31 ~]$ sudo yum install amazon-efs-utils
Last metadata expiration check: 0:09:42 ago on Fri Apr 12 16:03:05 2024.
Dependencies resolved.
================================================================================================================================
 Package                        Architecture          Version                                Repository                   Size
================================================================================================================================
Installing:
 amazon-efs-utils               noarch                1.35.2-1.amzn2023                      amazonlinux                  55 k
Installing dependencies:
 stunnel                        x86_64                5.58-1.amzn2023.0.2                    amazonlinux                 156 k

Transaction Summary
================================================================================================================================
Install  2 Packages

Total download size: 212 k
Installed size: 557 k
Is this ok [y/N]: y
```

Under EFS > view details > attach> mount via DNS> using the EFS mount helper > copy the link and run it in the AWS First EC2 CLI to mount the file system to the EFS directory



```
[ec2-user@ip-172-31-14-31 ~]$ sudo mount -t efs -o tls fs-0e5003ff261d42a3a:/ efs
[ec2-user@ip-172-31-14-31 ~]$ df -k
Filesystem              1K-blocks    Used        Available Use% Mounted on
devtmpfs                     4096       0             4096   0% /dev
tmpfs                      486172       0           486172   0% /dev/shm
tmpfs                      194472    2928           191544   2% /run
/dev/xvda1                8310764 1572284          6738480  19% /
tmpfs                      486172       0           486172   0% /tmp
/dev/xvda128                10202    1310             8892  13% /boot/efi
tmpfs                       97232       0            97232   0% /run/user/1000
127.0.0.1:/     9007199254739968       0 9007199254739968   0% /home/ec2-user/efs
```

Ran to “Mount | Column -t” check encryption at rest is enabled or not



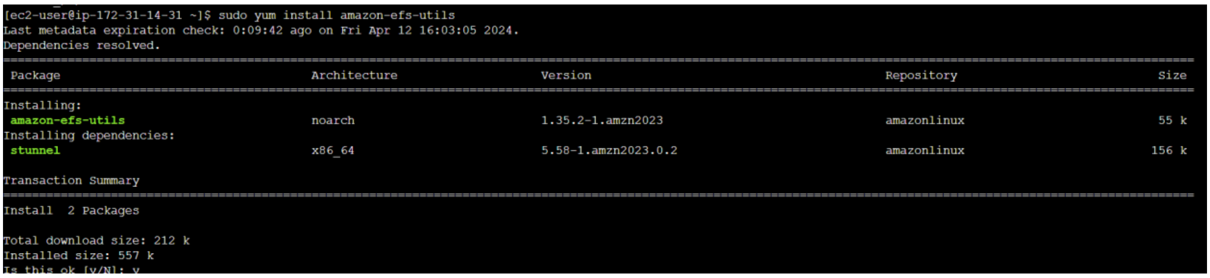
```
ramfs         on   /run/credentials/systemd-tmpfiles-setup.service      type   ramfs        (ro,nosuid,no
sunrpc        on   /var/lib/nfs/rpc_pipefs                              type   rpc_pipefs   (rw,relatime)
/dev/xvda128  on   /boot/efi                                            type   vfat         (rw,noatime,f
t,errors=remount-ro,x-systemd.automount)
tmpfs         on   /run/user/1000                                       type   tmpfs        (rw,nosuid,no
000,gid=1000)
127.0.0.1:/   on   /home/ec2-user/efs                                   type   nfs4         (rw,relatime,
proto=tcp,port=20962,timeo=600,retrans=2,sec=sys,clientaddr=127.0.0.1,local_lock=none,addr=127.0.0.1)
```

Connected to second EC2 instance and ran following commands:

sudo yum install amazon-efs-utils- to manually create the EFS client

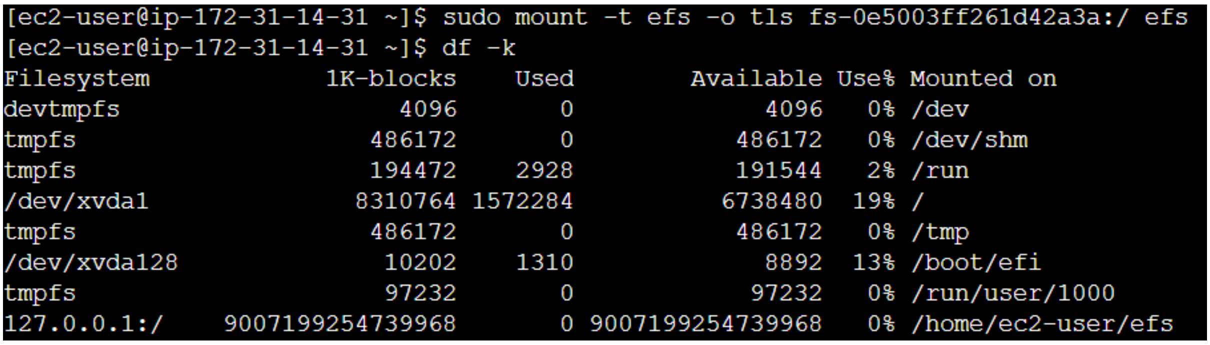
mkdir efs : To create directory

Perform following steps to mount file system on /home/ec2-user/efs directory



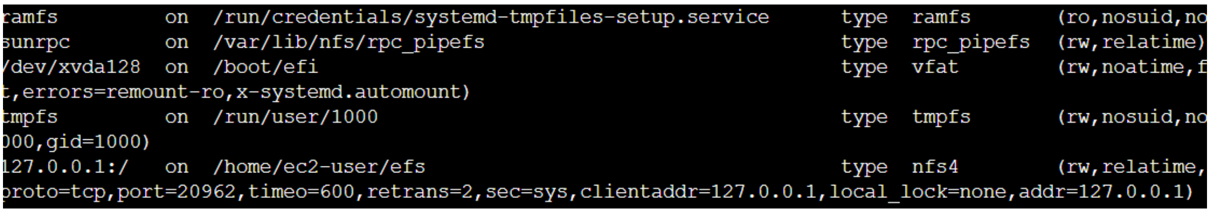
```
[ec2-user@ip-172-31-14-31 ~]$ sudo yum install amazon-efs-utils
Last metadata expiration check: 0:09:42 ago on Fri Apr 12 16:03:05 2024.
Dependencies resolved.
==========================================================================================================================================
 Package                            Architecture              Version                                       Repository               Size
==========================================================================================================================================
Installing:
 amazon-efs-utils                   noarch                    1.35.2-1.amzn2023                             amazonlinux              55 k
Installing dependencies:
 stunnel                            x86_64                    5.58-1.amzn2023.0.2                           amazonlinux             156 k

Transaction Summary
==========================================================================================================================================
Install  2 Packages

Total download size: 212 k
Installed size: 557 k
Is this ok [y/N]: y
```

Under EFS > view details > attach> mount via DNS> using the EFS mount helper > copy the link and run it in the AWS First EC2 CLI to mount the file system to the EFS directory



```
[ec2-user@ip-172-31-14-31 ~]$ sudo mount -t efs -o tls fs-0e5003ff261d42a3a:/ efs
[ec2-user@ip-172-31-14-31 ~]$ df -k
Filesystem               1K-blocks    Used        Available Use% Mounted on
devtmpfs                      4096       0             4096   0% /dev
tmpfs                       486172       0           486172   0% /dev/shm
tmpfs                       194472    2928           191544   2% /run
/dev/xvda1                 8310764 1572284          6738480  19% /
tmpfs                       486172       0           486172   0% /tmp
/dev/xvda128                 10202    1310             8892  13% /boot/efi
tmpfs                        97232       0            97232   0% /run/user/1000
127.0.0.1:/     9007199254739968       0 9007199254739968   0% /home/ec2-user/efs
```

Ran to “Mount | Column -t” check encryption at rest is enabled or not

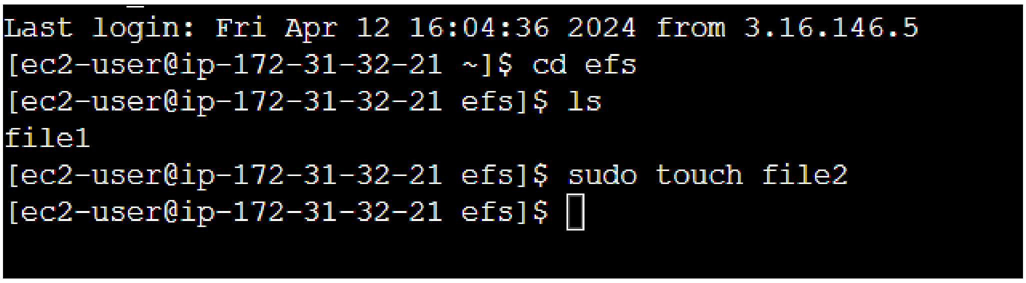


```
ramfs          on  /run/credentials/systemd-tmpfiles-setup.service      type  ramfs        (ro,nosuid,no
sunrpc         on  /var/lib/nfs/rpc_pipefs                              type  rpc_pipefs   (rw,relatime)
/dev/xvda128   on  /boot/efi                                            type  vfat         (rw,noatime,f
t,errors=remount-ro,x-systemd.automount)
tmpfs          on  /run/user/1000                                       type  tmpfs        (rw,nosuid,no
000,gid=1000)
127.0.0.1:/    on  /home/ec2-user/efs                                   type  nfs4         (rw,relatime,
proto=tcp,port=20962,timeo=600,retrans=2,sec=sys,clientaddr=127.0.0.1,local_lock=none,addr=127.0.0.1)
```

In first EC2 instance, went to the EFS directory and created a file “File2”

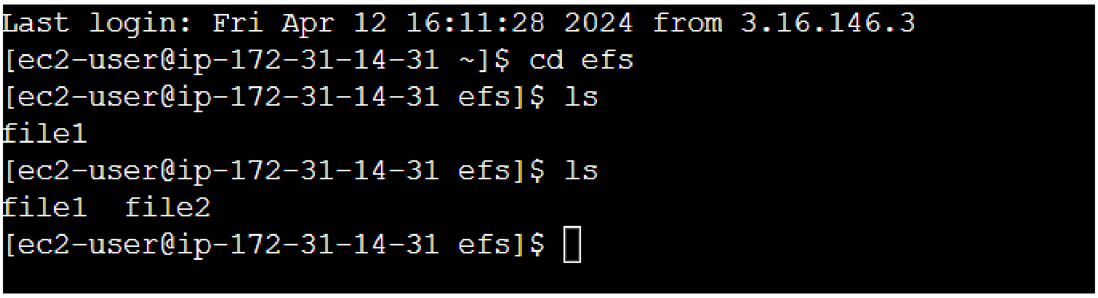
In Second instance, went to the EFS directory and listed the files and we are able to see the file created in First EC2 instance. Hence file sharing system is working fine.

Created “File2” in first instance



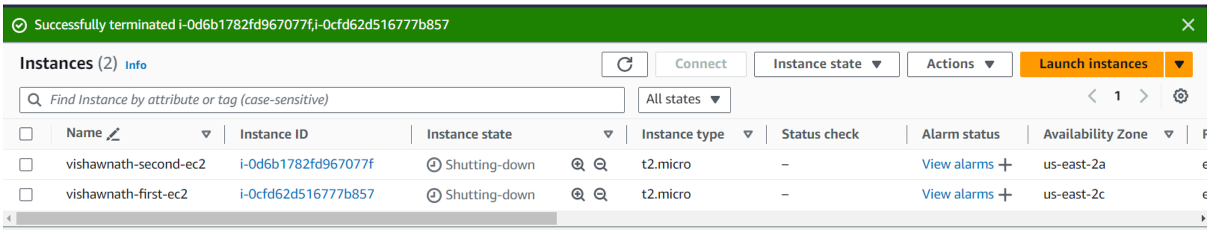
```
Last login: Fri Apr 12 16:04:36 2024 from 3.16.146.5
[ec2-user@ip-172-31-32-21 ~]$ cd efs
[ec2-user@ip-172-31-32-21 efs]$ ls
file1
[ec2-user@ip-172-31-32-21 efs]$ sudo touch file2
[ec2-user@ip-172-31-32-21 efs]$ 
```

“File2” is also showing up in second ec2 instance with the help of file sharing



```
Last login: Fri Apr 12 16:11:28 2024 from 3.16.146.3
[ec2-user@ip-172-31-14-31 ~]$ cd efs
[ec2-user@ip-172-31-14-31 efs]$ ls
file1
[ec2-user@ip-172-31-14-31 efs]$ ls
file1  file2
[ec2-user@ip-172-31-14-31 efs]$ 
```

First and Second Instance termination

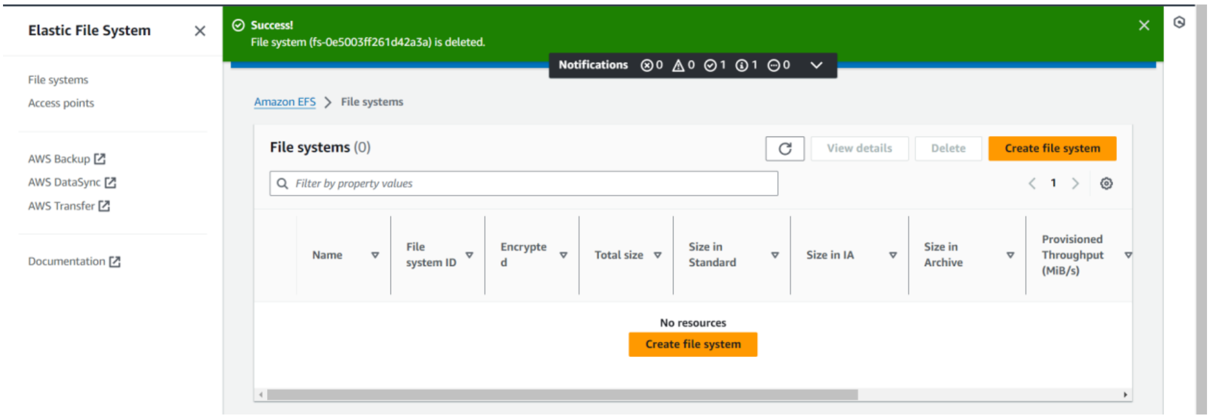


Successfully terminated i-0d6b1782fd967077f,i-0cfd62d516777b857

Instances (2)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	vishawnath-second-ec2	i-0d6b1782fd967077f	Shutting-down	t2.micro	–	View alarms +	us-east-2a
□	vishawnath-first-ec2	i-0cfd62d516777b857	Shutting-down	t2.micro	–	View alarms +	us-east-2c

File system deleted



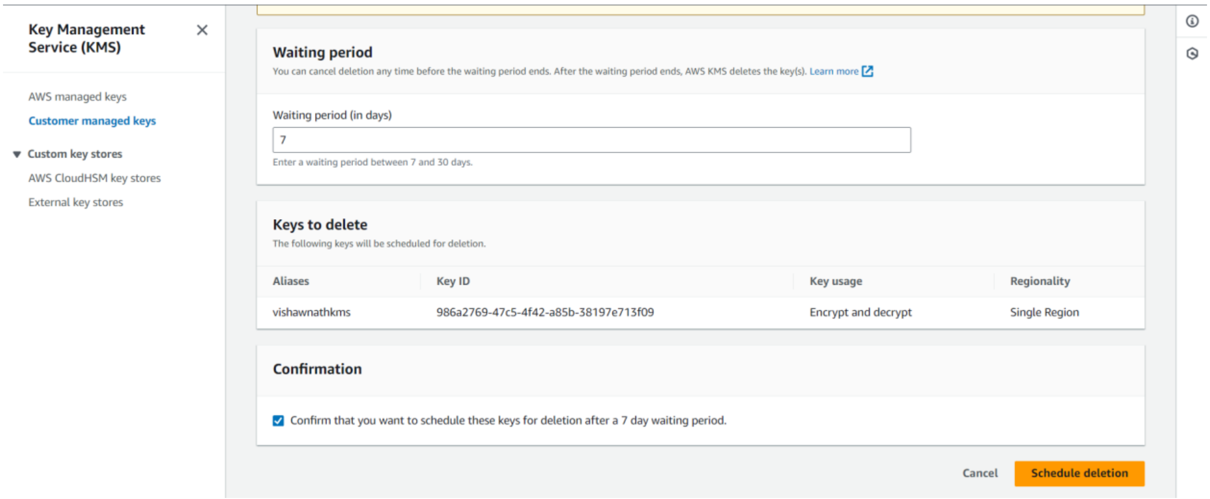
Success!
File system (fs-0e5003ff261d42a3a) is deleted.

File systems (0)

Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive	Provisioned Throughput (MiB/s)

No resources

In KMS Dashboard > select KMS key > select key actions > schedule key deletion & schedule key deletion



Waiting period

You can cancel deletion any time before the waiting period ends. After the waiting period ends, AWS KMS deletes the key(s). Learn more

Waiting period (in days)

7

Enter a waiting period between 7 and 30 days.

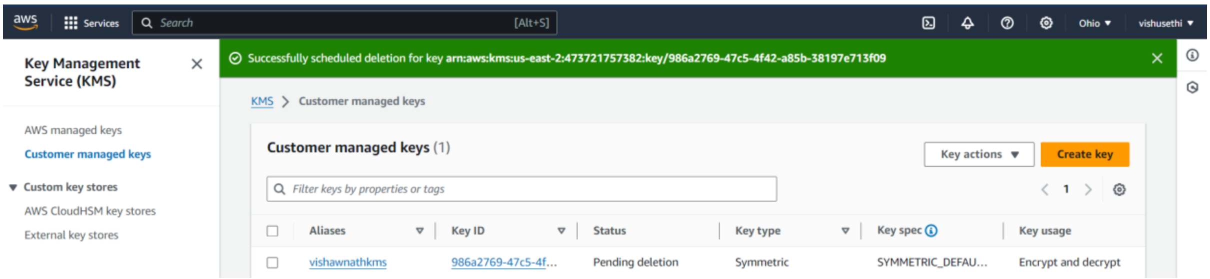
Keys to delete

The following keys will be scheduled for deletion.

Aliases	Key ID	Key usage	Regionality
vishawnathkms	986a2769-47c5-4f42-a85b-38197e713f09	Encrypt and decrypt	Single Region

Confirmation

☑ Confirm that you want to schedule these keys for deletion after a 7 day waiting period.



Successfully scheduled deletion for key arn:aws:kms:us-east-2:473721757382:key/986a2769-47c5-4f42-a85b-38197e713f09

KMS > Customer managed keys

Customer managed keys (1)

□	Aliases	Key ID	Status	Key type	Key spec	Key usage
□	vishawnathkms	986a2769-47c5-4f...	Pending deletion	Symmetric	SYMMETRIC_DEFAU...	Encrypt and decrypt