

AWS VPC last part:



Lets say we have two VPCs with VMs. These VPCs could be in the same AWS account or different AWS account.

Open VPCs

Your VPCs (2) [Info](#)

Search

<input type="checkbox"/>	Name	VPC ID	State
<input type="checkbox"/>	MyVPC	vpc-001c2f84899f4c3ea	Available
<input type="checkbox"/>	default_VPC	vpc-0a752647f0a021f2e	Available

We have 2 VPCs here
Click on Peering connections:
No Peering connections found

VPC dashboard <

EC2 Global View [↗](#)

Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections**

Peering connections [Info](#)

Find resources by attribute or tag

Name	Peering
------	---------

Select a peering connection above

What's the use of Peering connections?

VPC Peering is a networking connection that allows two Amazon Virtual Private Clouds (VPCs) to communicate privately as if they were in the same network.

Select Requester as MyVPC and I want to connect with default-VPC

[Connections](#) > Create peering connection

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traf

Peering connection settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

My-VPC-PC

Select a local VPC to peer with

VPC ID (Requester)

vpc-001c2f84899f4c3ea (MyVPC)

VPC CIDRs for vpc-001c2f84899f4c3ea (MyVPC)

CIDR	Status	Status reason
10.0.0.0/16	✔ Associated	-

Select another VPC to peer with

Account

- ☒ My account
☐ Another account

Region

- ☒ This Region (ca-central-1)
☐ Another Region

VPC ID (Acceptor)

vpc-0a752647f0a021f2e (default_VPC)

VPC CIDRs for vpc-0a752647f0a021f2e (default_VPC)

Select another VPC to peer with

Account
☒ My account
☐ Another account

Region
☒ This Region (ca-central-1)
☐ Another Region

VPC ID (Acceptor)
 vpc-0a752647f0a021f2e (default_VPC)

VPC CIDRs for vpc-0a752647f0a021f2e (default_VPC)

CIDR	Status	Status reason
172.31.0.0/16	Associated	-

Tags
 A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Value - optional**

You can add 49 more tags.

Click Create Peering connection

Actions ----> Accept request

Accept VPC peering connection request [Info](#)

Are you sure you want to accept this VPC peering connection request? (pcx-03037da873e7823fd / My-VPC-PC)

Requester VPC vpc-001c2f84899f4c3ea / MyVPC	Acceptor VPC vpc-0a752647f0a021f2e / default_VPC	Requester CIDRs <input type="checkbox"/> 10.0.0.0/16
Acceptor CIDRs -	Requester Region Central (ca-central-1)	Acceptor Region Central (ca-central-1)
Requester owner ID <input type="checkbox"/> 577638386543 (This account)	Acceptor owner ID <input type="checkbox"/> 577638386543 (This account)	

✓ Your VPC peering connection (pcx-03037da873e7823fd | My-VPC-PC) has been established.
 To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables.

We have to go and enable in the route tables that VMs can interact with each other

Go to Route tables

Route tables (3) [Info](#)

<input type="checkbox"/>	Name	
<input type="checkbox"/>	MyVPC-RT-private	
<input type="checkbox"/>	MyRT-public	
<input type="checkbox"/>	Default-RT	

<input checked="" type="checkbox"/>	Default-RT	rtb-0167102c60028bf78	-
-------------------------------------	------------	---------------------------------------	---

rtb-0167102c60028bf78 / Default-RT

Details	Routes	Subnet associations	Edge associations	Route propagation
Routes (2)				
<input type="text" value="Filter routes"/>				
Destination		▼	Target	
0.0.0.0/0			igw-07e4ffa1bfe1a8d7f	
172.31.0.0/16			local	

Go back to My_VPC

<input checked="" type="checkbox"/>	Name	▼	VPC ID	▼	State	▼
<input checked="" type="checkbox"/>	MyVPC		vpc-001c2f84899f4c3ea		✓ Available	
<input type="checkbox"/>	default_VPC		vpc-0a752647f0a021f2e		✓ Available	

vpc-001c2f84899f4c3ea / MyVPC

Details	Resource map	CIDRs	Flow logs	Tags	Integrations
IPv4 CIDRs Info					
Address family			▲	CIDR	
IPv4				10.0.0.0/16	

Copy CIDR 10.0.0.0/16

Go back to Default-RT

rtb-0167102c60028bf78 / Default-RT

Actions

Details

Route table ID
rtb-0167102c60028bf78

VPC
vpc-0a752647f0a021f2e | default_VPC

Main

Yes

Owner ID
577638386543

Explicit subnet associations

-

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

Both

Edit routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-07e4ffa1bfe1a8d7f	Active	No
172.31.0.0/16	local	Active	No

What it means is, allow requests from this IP: 10.0.0.0/16. there will be a request from MyVPC into default-VPC that's the meaning
From this particular CIDR block there will be a request

Add route

tables

>

rtb-0167102c60028bf78

>

Edit routes

Edit routes

Destination

172.31.0.0/16

Target

local

Status

Active

0.0.0.0/0

Internet Gateway

Status

Active

10.0.0.0/16

Peering Connection

-

pcx-03037da873e7823fd

Add route

For two way, update default-VPC CIDR in My-Route table

<input checked="" type="checkbox"/>	default_VPC	vpc-0a752647f0a021f2e	<input checked="" type="checkbox"/> Available
-------------------------------------	-------------	-----------------------	---

vpc-0a752647f0a021f2e / default_VPC

Details

Resource map

CIDRs

Flow logs

Tags

Integrations

IPv4 CIDRs

Info

Address family

▲

CIDR

IPv4

172.31.0.0/16

Click on public Route table

Route tables (1/3) Info				
<input type="text" value="Find resources by attribute or tag"/>				
<input type="checkbox"/>	Name	▼	Route table ID	▼ Ex
<input type="checkbox"/>	MyVPC-RT-private		rtb-04905a360f41f4742	sul
<input checked="" type="checkbox"/>	MyRT-public		rtb-0cdb1ce0fcf96d17f	sul
<input type="checkbox"/>	Default-RT		rtb-0167102c60028bf78	-

rtb-0cdb1ce0fcf96d17f / MyRT-public

Details	Routes	Subnet associations	Edge associations	ROI
Details				
Route table ID		Main		
<input type="checkbox"/> rtb-0cdb1ce0fcf96d17f		<input type="checkbox"/> No		
VPC		Owner ID		
vpc-001c2f84899f4c3ea MyVPC		<input type="checkbox"/> 577638386543		

Routes ---> Edit route --> Add route
Added default-VPC CIDR into Public-RT

> [rtb-0cdb1ce0fc96d17f](#) > Edit routes

Edit routes

Destination	Target	Status
10.0.0.0/16	local	✓ Active
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="local"/>	
	Internet Gateway	✓ Active
<input type="text" value="172.31.0.0/16"/>	<input type="text" value="igw-0ba79e620177b3bc1"/>	
	Peering Connection	-
	<input type="text" value="pcx-03037da873e7823fd"/>	
	Use: "pcx-03037da873e7823fd"	
	pcx-03037da873e7823fd (My-VPC-PC)	

[Add route](#)

Launch an EC2 instance

Key pair name - required

[Create new](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

[10.0.0.0/16](#) [Create new](#)

Subnet [Info](#)

[subnet-public](#) [Create new](#)

VPC: vpc-001c2f84899f4c3ea Owner: 577638386543 Availability Zone: ca-central-1a
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.0.0/24

Auto-assign public IP [Info](#)

[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](#)

[Compare s...](#)

VPC: vpc-001c2f84899f4c3ea

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

subnet-public is there

Create another VM into default-VPC

DevOpsMar30 ▼ [Create new key pair](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0a752647f0a021f2e (default_VPC) (default) ▼ [↻](#)

172.31.0.0/16

Subnet [Info](#)

No preference ▼ [↻](#) [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 ▼

default sg-0483bbb02e36e7efe ✕

VPC: vpc-0a752647f0a021f2e

[↻ Compare security group rules](#)

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

<input type="checkbox"/>	default-VM	i-04cf51658a03b7f44	✔ Running 🔍 🔍	t2.micro	⌚ Initializing
<input type="checkbox"/>	public-EC2	i-09132869fe2ce9fe2	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed

Click on Connect in public-EC2

Instances > [i-09132869fe2ce9fe2](#) > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-09132869fe2ce9fe2 (public-EC2) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

[🔍](#) i-09132869fe2ce9fe2 (public-EC2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is DevOpsMar30.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

[🔍](#) `chmod 400 "DevOpsMar30.pem"`
4. Connect to your instance using its Public IP:

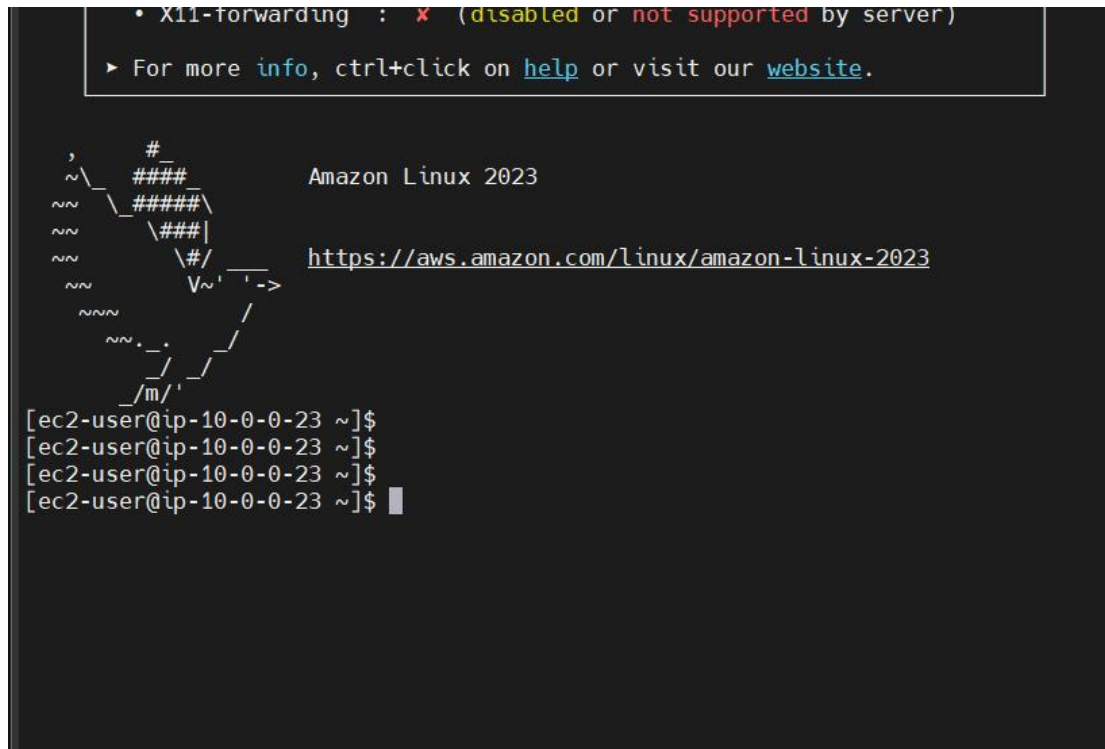
[🔍](#) 15.223.182.38

Example:

[🔍](#) `ssh -i "DevOpsMar30.pem" ec2-user@15.223.182.38`

[🔍](#) **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has

Connect to public-EC2



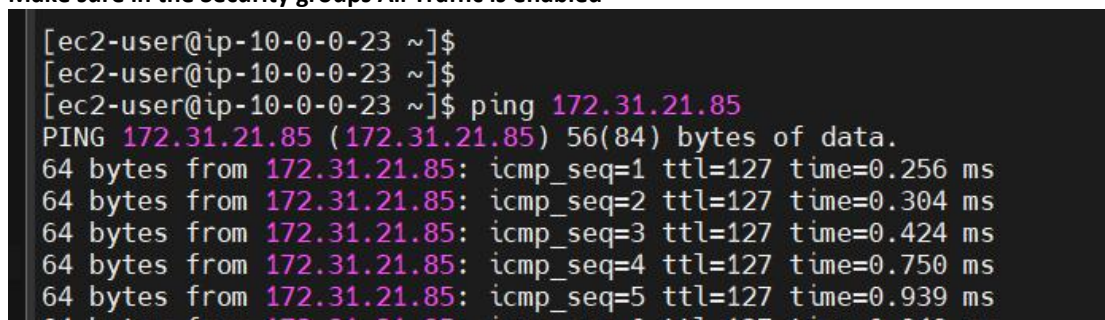
Copy Private IPv4 address from default-VM

i-04cf51658a03b7f44 (default-VM)			
Details Status and alarms Monitoring Security Networking Storage Tags			
▼ Instance summary <small>Info</small>			
Instance ID i-04cf51658a03b7f44	Public IPv4 address 3.96.221.195 open address	Private IPv4 addresses 172.31.21.85	
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-96-221-195.ca-central-1.compute.amazonaws.com	

Private IP: 172.31.21.85

Connect from Public-EC2 to default-VM

Make sure in the Security groups All Traffic is enabled



In this example, both VPCs are in the same account

Yes, we can establish connection between two VPCs in different accounts as well. This is called as VPC peering

Security Groups:

Why do we have security groups?

Acts like a firewall to secure our resources, decides incoming traffic and outgoing traffic

SG contains two important parts -> inbound rules and outbound rules

Inbound rules: incoming traffic to your VM
Outgoing rules: outgoing traffic from your VM

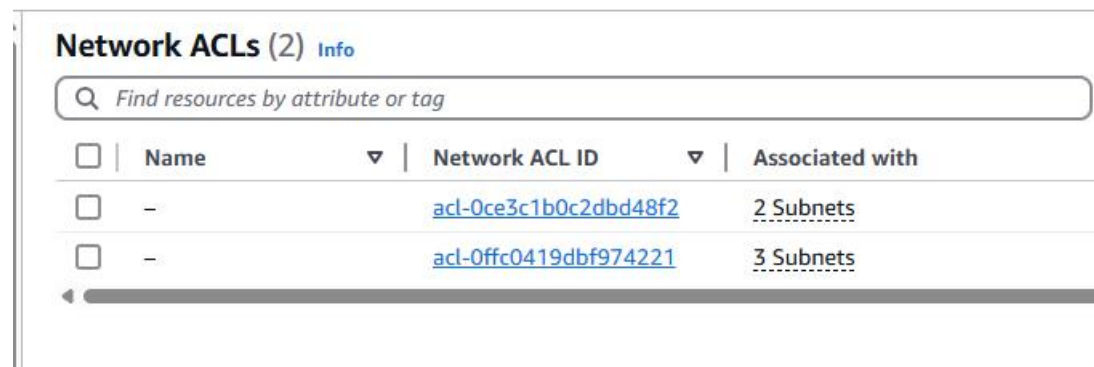
In One Security group, we can add 50 rules
By default, all traffic is denied in security groups, unless and until you make the request for what kind of traffic is allowed etc

- > Security group only allows rules, by default all rules are denied
- > We cannot configure deny rules in security group
- > Security groups are applicable at resource level (not at subnet or VPC) and manually we have to add Security group to a resource
- > Security groups are stateful (any changes we apply to the incoming rules it is applicable to outgoing rules also)

NACL: Network Access Control List / Network ACL

NACL acts as a Firewall for our subnets in VPC
It is applicable at the subnet level
NACL rules are applicable resources, which are part of a subnet
NACL is Stateless (Any rule for the incoming traffic must be manually added to outgoing traffic)
In NACL, we can configure both allow and deny rules
One Subnet can have only one NACL
However, one NACL can be added to multiple subnets
NACL acts as first level of defense for incoming traffic while security group acts as first level of defense for outgoing traffic
We have Subnet first then VM inside, so first level of defense for incoming is NACL but for outgoing is Security group

In VPC, we can find NACL



The screenshot shows the AWS Network ACLs console. At the top, it says "Network ACLs (2) Info". Below this is a search bar with the placeholder text "Find resources by attribute or tag". The main content is a table with the following columns: "Name", "Network ACL ID", and "Associated with". There are two rows of data:

<input type="checkbox"/>	Name	Network ACL ID	Associated with
<input type="checkbox"/>	-	acl-0ce3c1b0c2dbd48f2	2 Subnets
<input type="checkbox"/>	-	acl-0ffc0419dbf974221	3 Subnets

By default, 2 NACLs are created. We are using only 2 NACLs for all the subnets. Multiple NACLs cannot be used in one subnet but vice-versa is possible.

Create NACL

Create network ACL [Info](#)

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Network ACL settings

Name - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

My-NACL

VPC

VPC to use for this network ACL.

vpc-001c2f84899f4c3ea (MyVPC)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track costs.

Key

Q Name

Value - *optional*

Q My-NACL

Add tag

You can add 49 more tags

In NACL, we have both options: Allow and Deny

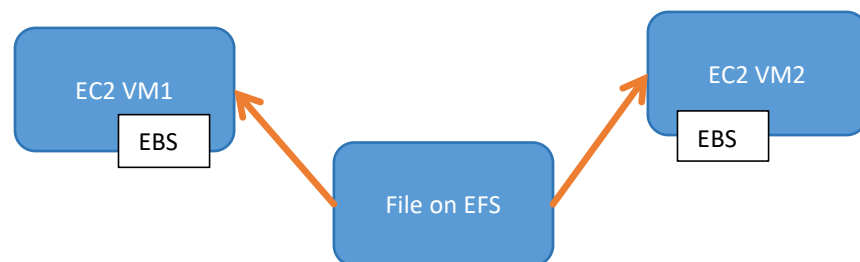
Inbound rules (2)								Edit inbound rules	
Filter inbound rules								< 1 > ⚙	
Rule number	Type	Protocol	Port range	Source	Allow/Deny				
1	Custom TCP	TCP (6)	0	0.0.0.0/0	Allow				
*	All traffic	All	All	0.0.0.0/0	Deny				

Elastic File System (EFS):

EBS --> Elastic Block Store

S3 --> unlimited storage

EFS --> File system storage --> shared that means this particular file can work with multiple resources



This file on EFS can be accessed by multiple resources at the same time

Amazon Elastic File System

Scalable, elastic, cloud-native NFS file system


Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for general purpose workloads for use with AWS Cloud services and on-premises resources.

Create file sy

Create an EFS fil

Create file sy

What is Amazon Elastic File System?

 Amazon Elastic File System - Scalable, Elastic, Cloud-Nativ...[Copy link](#)

Pricing

With EFS, there a
the storage that
write, and any ac
Estimate your co

Advantage:

Create scalable file storage to be used on EC2

Fully managed by AWS, Low cost, pay for what you use, highly available and scalable performance

Create file system

Create a file system with the recommended settings shown below by choosing Create file system. To view all settings or to customize your file system, choose Customize. [Learn more](#)

Name - optional

Name your file system.

MyTestEFS

Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)

Choose the VPC where you want EC2 instances to connect to your file system.

vpc-0a752647f0a021f2e
default

Recommended settings

Your file system is created with the following recommended settings unless you choose to customize the file system. You will be charged for storage and throughput. We recommend reviewing pricing for these features using the [AWS Pricing Calculator](#).

Setting	Value	Editable after creation
Throughput mode Learn more	Elastic	Yes
Transition into Infrequent Access (IA)	30 day(s) since last access	Yes
Transition into Archive	90 day(s) since last access	Yes
Transition into Standard	None	Yes
Automatic backups	Enabled	Yes
Encryption	Enabled	No

Cancel

Customize

Create file system

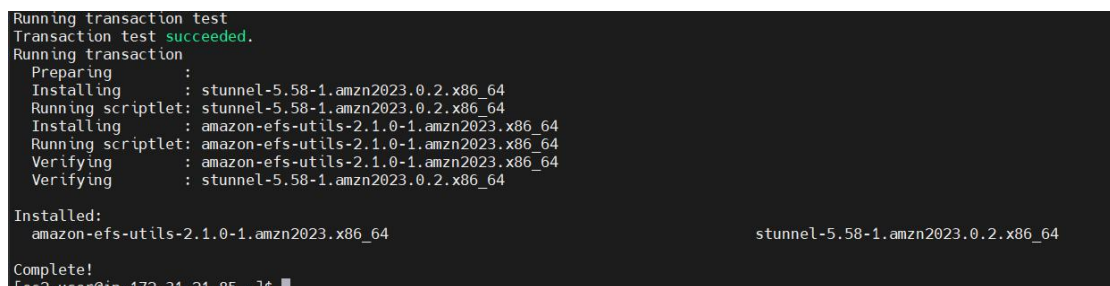
File ID that's required

File systems (1)					
<input type="text"/> Filter by property values					
	Name ▾	File system ID ▾	Encrypte d ▾	Total size ▾	Size in Standard ▾
<input type="radio"/>	MyTestEFS	fs-03ab33f2b0d3ad481	<input checked="" type="checkbox"/> Encrypted	6.00 KiB	6.00 KiB

I am using the defaultVM




[ec2-user@ip-172-31-21-85 ~]\$ sudo yum install -y amazon-efs-utils



```
[ec2-user@ip-172-31-21-85 ~]$
[ec2-user@ip-172-31-21-85 ~]$ sudo mkdir efsdir
[ec2-user@ip-172-31-21-85 ~]$
[ec2-user@ip-172-31-21-85 ~]$ ls
efsdir
[ec2-user@ip-172-31-21-85 ~]$
[ec2-user@ip-172-31-21-85 ~]$
```

Command for mounting
Copy the EFS file system ID

File systems (1)							
<input type="text" value="Filter by property values"/>							
	Name ▾	File system ID ▾	Encrypte d ▾	Total size ▾	Size in Standard ▾	Size in IA ▾	Size in Archive
<input type="radio"/>	MyTestEFS	fs-03ab33f2b0d3ad481	 Encrypte d	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes

Attach EFS File system ID to VM


```
[ec2-user@ip-172-31-21-85 ~]$
[ec2-user@ip-172-31-21-85 ~]$ sudo mount -t efs -o tls fs-03ab33f2b0d3ad481:/ efsdir
```

```
[ec2-user@ip-172-31-21-85 ~]$ sudo mount -t efs -o tls fs-03ab33f2b0d3ad481:/ efsdir
[ec2-user@ip-172-31-21-85 ~]$ cd efsdir/
[ec2-user@ip-172-31-21-85 efsdir]$ ls
```

```
[ec2-user@ip-172-31-21-85 efsdir]$ sudo touch demo.txt
[ec2-user@ip-172-31-21-85 efsdir]$ ls -l
total 4
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
```

We have attached EFS to default-VM

Now opening public-EC2

<input type="checkbox"/>	default-VM	i-04cf51658a03b7f44	 Running	 	t2.n
<input checked="" type="checkbox"/>	public-EC2	i-09132869fe2ce9fe2	 Running	 	t2.n

Open public-EC2 in a different terminal window


```

~\#####\
~\###|
~\#/
~V~'-'>
~~~~
~~~.
~~\_/
~~\_/
~~\m/'

Last login: Mon Mar 31 00:41:24 2025 from 99.228.11.52
[ec2-user@ip-10-0-0-23 ~]$
[ec2-user@ip-10-0-0-23 ~]$
[ec2-user@ip-10-0-0-23 ~]$

```

<https://aws.amazon.com/linux/amazon-linux-202>

```
[ec2-user@ip-10-0-0-23 ~]$ sudo yum install -y amazon-efs-utils
```

```
[ec2-user@ip-10-0-0-23 ~]$ sudo mount -t efs -o tls fs-03ab33f2b0d3ad481:/ efsdir
Failed to resolve "fs-03ab33f2b0d3ad481.efs.ca-central-1.amazonaws.com" - check that your file
system ID is correct, and ensure that the VPC has an EFS mount target for this file system ID.
```

Your EC2 instance and EFS must be in the same VPC.

Lets create one more EC2 in the same VPC

<input type="checkbox"/>	public-EC2	i-09132869fe2ce9fe2	Stopped	t2.micro
<input checked="" type="checkbox"/>	efs-ec2	i-0728f26f52d9746b8	Running	t2.micro

```
[ec2-user@ip-172-31-24-90 ~]$
[ec2-user@ip-172-31-24-90 ~]$ sudo yum install -y amazon-efs-utils
```

We mount EFS once again on this new EC2

```
[ec2-user@ip-172-31-24-90 ~]$ sudo mkdir efsdir
[ec2-user@ip-172-31-24-90 ~]$ sudo mount -t efs -o tls fs-03ab33f2b0d3ad481:/ efsdir
[ec2-user@ip-172-31-24-90 ~]$ cd efsdir
[ec2-user@ip-172-31-24-90 efsdir]$ ls -l
total 4
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
[ec2-user@ip-172-31-24-90 efsdir]$
```

We see the same demo.txt in this efsdir

```

b'mount.nfs4: mount point efsdir does not exist'
[ec2-user@ip-172-31-24-90 ~]$ sudo mkdir efsdir
[ec2-user@ip-172-31-24-90 ~]$ sudo mount -t efs -o tls fs-03ab33f2b0d3ad481:/ efsdir
[ec2-user@ip-172-31-24-90 ~]$ cd efsdir
[ec2-user@ip-172-31-24-90 efsdir]$ ls -l
total 4
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
[ec2-user@ip-172-31-24-90 efsdir]$

```

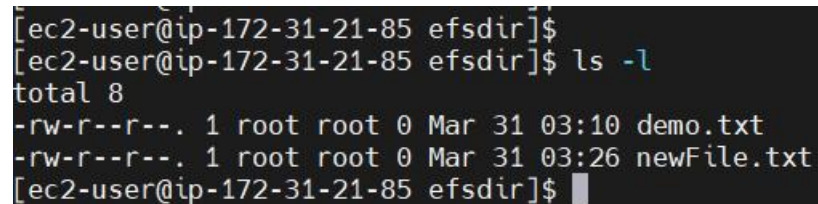
```
[ec2-user@ip-172-31-24-90 efsdir]$ sudo touch newFile.txt
[ec2-user@ip-172-31-24-90 efsdir]$ ls -l
total 8
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
-rw-r--r--. 1 root root 0 Mar 31 03:26 newFile.txt
```

We have two files on efs-ec2 new VM

Go to the other EC2:

```
[ec2-user@ip-172-31-21-85 efsdir]$ ls -l
total 8
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
-rw-r--r--. 1 root root 0 Mar 31 03:26 newFile.txt
```

We can see newFile.txt in the default-VM, which we created actually in efs-ec2

A terminal window screenshot showing the command 'ls -l' being executed in the directory '/efsdir' on an EC2 instance with IP 172.31.21.85. The output shows two files: 'demo.txt' and 'newFile.txt', both with permissions '-rw-r--r--', owned by 'root', and created on 'Mar 31' at '03:10' and '03:26' respectively. The prompt '[ec2-user@ip-172-31-21-85 efsdir]\$' is visible at the top and bottom of the terminal output.

```
[ec2-user@ip-172-31-21-85 efsdir]$
[ec2-user@ip-172-31-21-85 efsdir]$ ls -l
total 8
-rw-r--r--. 1 root root 0 Mar 31 03:10 demo.txt
-rw-r--r--. 1 root root 0 Mar 31 03:26 newFile.txt
[ec2-user@ip-172-31-21-85 efsdir]$
```

Steps to work with EFS practicals:

1. Login into AWS console ---> Services ---> EFS (it is in Storage also) --> Create FileSystem
2. Get File System ID
3. Create EC2 instances --> two different instances
4. Login into EC2 instance using Moba
5. Sudo yum install -y amazon-efs-utils
6. Create a folder/directory
7. sudo mkdir efsdir
8. Mounting File system:
9. sudo mount -t efs -o tls <file-id> ./ efsdir
10. cd efsdir
11. Create files in the directory
12. Then Connect to the second EC2 instance
13. Repeat the same steps
14. Check behavior of shared file system check efsdir