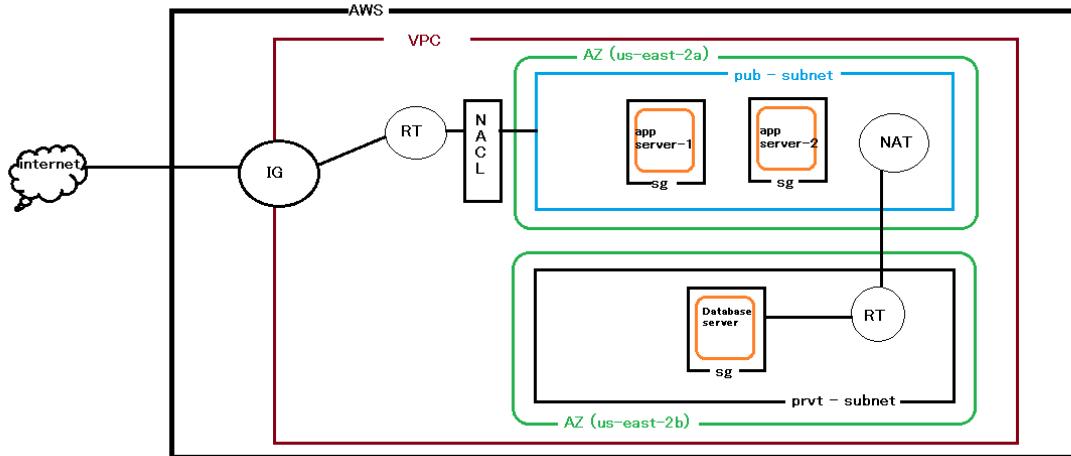


## Custom VPC with EC2 to S3, EFS and EC2 to RDS



VPC - Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Virtual Private Cloud (VPC) lets you have a virtual network, so that you can select your own IP address range, create your own subnets and configure your route tables and network gateways. Here is a quick guide on how to setup VPC on Amazon Web Services (AWS).

### **PART – 1 : VPC and subnets creation**

#### **Step 1 : Creating VPC**

- Goto aws console vpc dashboard and select Create VPC.
- VPC settings --> Resource to create – VPC only, Name tag – add a name to Custom VPC, IPV4 CIDR – 10.0.0.0/16 ( to the 16 mask bit value there will be 65536 IP's are generating into this VPC), Tenancy – Default.
- And create VPC

The screenshot shows a step-by-step process for creating a VPC in the AWS Management Console.

**Step 1: Your VPCs**

The first window shows the "Your VPCs" list. It displays one VPC entry:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-04b0d41543300892e	Available	172.31.0.0/16	-

**Step 2: Create VPC**

The second window is the "Create VPC" configuration page. It includes the following fields:

- VPC settings:**
  - Resources to create:** "VPC only" (selected) and "VPC and more". A red checkmark is placed on "VPC only".
  - Name tag - optional:** "custom-vpc". A red checkmark is placed next to it.
  - IPv4 CIDR block:** "10.0.0.0/16". A red checkmark is placed next to it.
  - IPv6 CIDR block:** "No IPv6 CIDR block" (selected). A red checkmark is placed next to it.
  - Tenancy:** "Default". A red checkmark is placed next to it.
- Tags:** A key-value pair "Name: custom-vpc". A red checkmark is placed next to the "Remove" button.

**Step 3: Confirmation**

The third window shows a green success message: "You successfully created vpc-0d05e0208e0ec5714 / custom-vpc".

**Step 4: VPC Details**

The fourth window displays the details of the newly created VPC:

VPC ID	State	DNS hostnames	DNS resolution
vpc-0d05e0208e0ec5714	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-0a4a3fe36e9a1a955	rtb-0d882d7b4bc906647	acl-0a5f5649ea60d42f0
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR
No	10.0.0.0/16	-	-
Route 53 Resolver DNS Firewall rule groups	Owner ID		
=	881832161071		

## Step 2 : Creating private and public Subnets and attach to custom VPC.

- In the VPC dashboard click subnet - > create subnet.
- VPC ID – select the custom vpc which are previously created.
- Subnet settings --> subnet name – public subnet, Availability Zone – select one Az, ipv4 cidr block – 10.0.0.0/24 (to the 24 mask bit value there will be 256 IP's are generating into this subnet )
- Add new subnet --> subnet name – private subnet, Availability Zone – select different Az compare to public subnet, ipv4 cidr block – 10.0.1.0/24 (to the 24 mask bit value there will be 256 IP's are generating into this subnet , make sure the IP's are not overlap with other subnet)
- And create the subnets.

The screenshots illustrate the process of creating a new subnet. The top screenshot shows the existing subnets in the 'Subnets (3) Info' table. The bottom screenshot shows the 'Create subnet' wizard, where the user has selected the VPC ID 'custom-vpc' and is choosing the CIDR block for the new subnet. The '10.0.0.0/16' block is selected and highlighted with a red checkmark.

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

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

The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 CIDR block** [Info](#)

**Tags - optional**

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Q public-subnet"/>

**Add new tag**  
You can add 49 more tags.

**Remove**

**Activate Windows**  
Go to Settings to activate Windows.

**Feedback** Looking for language selection? Find it in the new [Unified Settings](#).

**Subnet 2 of 2**

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

The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 CIDR block** [Info](#) ✓

**Tags - optional**

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Q private-subnet"/>

**Add new tag**  
You can add 49 more tags.

**Remove**

**Activate Windows**  
Go to Settings to activate Windows.

**Feedback** Looking for language selection? Find it in the new [Unified Settings](#).

**Subnets (2) [Info](#)**

**Actions** Create subnet

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	public-subnet	subnet-061e8a4e28673b206	<span style="color: green;">Available</span>	vpc-0d05e0208e0ec5714   cus...	10.0.0.0/24	-
<input type="checkbox"/>	private-subnet	subnet-0c7a2470ca84e97b8	<span style="color: green;">Available</span>	vpc-0d05e0208e0ec5714   cus...	10.0.1.0/24	-

**Step 3 :** Access the internet from the outside world to the VPC, Internet gateway are used and Public subnets are associated with a route table that directs subnet traffic to the internet using an Internet Gateway.

Internet gateway,

- From the VPC dashboard select Internet gateway and click create internet gateway.
- Give the name of IG for identification and create internet gateway.
- Attach this Internet gateway to the custom VPC.

Route tables,

- From the VPC dashboard select Route tables and click create Route tables.
- Name – public RT, VPC – select the custom VPC and create it.
- Associate this Route tables with the public subnet from the subnet association --> edit subnet associations --> select public subnet and save associates.
- Now goto the Routes --> Edit routes --> destination – 0.0.0.0/0 and Target – internet gateway select it which are previously created and save changes.

The image consists of three vertically stacked screenshots of the AWS VPC console interface.

**Screenshot 1: Internet gateways (1/1) - List View**

This screenshot shows a list of existing Internet Gateways. One gateway is listed with the ID **igw-04a6bb00db2fa2531**, State **Attached**, and VPC ID **vpc-04b0d41543300892e**. A red arrow points to the "Create Internet gateway" button at the top right of the list view.

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-04a6bb00db2fa2531	Attached	vpc-04b0d41543300892e	881832161071

**Screenshot 2: Create internet gateway - Settings**

This screenshot shows the "Create internet gateway" wizard. In the "Internet gateway settings" step, a "Name tag" is being added with the value **myIG**. Below this, under "Tags - optional", a single tag is defined with Key **Name** and Value **myIG**.

**Screenshot 3: Create internet gateway - Confirmation**

This screenshot shows the confirmation step of the wizard. It displays a message: "The following internet gateway was created: igw-08cba379d52dc44 - myIG. You can now attach to a VPC to enable the VPC to communicate with the internet." The gateway ID **igw-08cba379d52dc44 / myIG** is shown. On the right, a context menu for the gateway is open, with the "Attach to a VPC" option highlighted.

**Attach to VPC (igw-08cba379d52dc44) Info**

VPC  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs  
Attach the internet gateway to this VPC.

Select a VPC  
vpc-0d05e0208e0ec5714 - custom-vpc  
▶ AWS Command Line Interface command

Cancel **Attach internet gateway**

**Internet gateway igw-08cba379d52dc44 successfully attached to vpc-0d05e0208e0ec5714**

**igw-08cba379d52dc44 / myIG**

**Details Info**

Internet gateway ID igw-08cba379d52dc44	State <span style="color: green;">Attached</span>	VPC ID vpc-0d05e0208e0ec5714   custom-vpc	Owner 881832161071
--	--	--	-----------------------

**Create route table**

**Route tables (2) Info**

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
-	rtb-00b0630c85fe35ad6	-	-	Yes	vpc-04b0d41543300892e
-	rtb-0d882d7b4bc906647	-	-	Yes	vpc-0d05e0208e0ec5714   cus...

**Create route table**

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

Name - optional  
Create a tag with a key of 'Name' and a value that you specify.  
**public-RT**

VPC  
The VPC to use for this route table.  
**vpc-0d05e0208e0ec5714 (custom-vpc)**

**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 Name	Value - optional public-RT	Remove
-------------	-------------------------------	--------

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

**Screenshot 1: AWS VPC Route Table Details**

The screenshot shows the AWS VPC Route Tables page. A specific route table, "rtb-098243f3b291ba7f3 / public-RT", is selected. The "Details" tab is active, showing the following information:

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-098243f3b291ba7f3	No	-	-
VPC	Owner ID		
vpc-0d05e0208e0ec5714   custom-vpc	881832161071		

Below the details, there are tabs for Routes, Subnet associations, Edge associations, Route propagation, and Tags. The Subnet associations tab is selected, displaying a message: "You can now check network connectivity with Reachability Analyzer" and a "Run Reachability Analyzer" button.

**Screenshot 2: Edit Subnet Associations**

This screenshot shows the "Edit subnet associations" interface for the selected route table. It lists available subnets:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
public-subnet	subnet-061e8a4e28673b206	10.0.0.0/24	-	Main (rtb-0d882d7b4bc906647)
private-subnet	subnet-0c7a2470ca84e97b8	10.0.1.0/24	-	Main (rtb-0d882d7b4bc906647)

The "public-subnet" checkbox is checked. Below this, the "Selected subnets" section shows the chosen subnet: "subnet-061e8a4e28673b206 / public-subnet". At the bottom are "Cancel" and "Save associations" buttons, with a red arrow pointing to the "Save associations" button.

**Screenshot 3: AWS VPC Route Table Details (After Changes)**

This screenshot shows the route table after changes have been made. The "Routes" tab is active, displaying one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

A red arrow points to the "Edit routes" button at the top right of the routes table.

The screenshot shows two consecutive screenshots of the AWS VPC Route Tables interface.

**Screenshot 1: Edit routes**

- Route Table:** rtb-098243f3b291ba7f3
- Destination:** 10.0.0.0/16
- Target:** local
- Status:** Active
- Propagated:** No
- Actions:** Remove
- Search:** in
- Options:** Add route

**Screenshot 2: Updated routes for rtb-098243f3b291ba7f3 / public-RT successfully**

- Route Table:** rtb-098243f3b291ba7f3
- Owner ID:** 881832161071
- Subnet:** subnet-061e8a4e28673b206 / public-subnet
- Routes:**
  - Destination: 0.0.0.0/0 Target: igw-08cba379d52cd44 Status: Active Propagated: No
  - Destination: 10.0.0.0/16 Target: local Status: Active Propagated: No
- Actions:** Edit routes

**Step 4:** Create the NACL an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

- From the VPC dashboard select network ACL and click create network ACL.
- Name – publicnacl, VPC – select the custom VPC. And create.
- Goto the NACL, select public nacl --> actions--> Edit subnet association --> select public subnet and save changes.
- Similarly create one more NACL for private subnet and give the subnet association as private subnet.
- As per present on going task, give the allow all traffic of inbound and outbound rules to the public NACL and Deny the all traffic of inbound and outbound rules to the private 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.

VPC  
VPC to use for this network ACL.

**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
<input type="text" value="Name"/>	<input type="text" value="publicacl"/>
<a href="#">Remove</a>	
<a href="#">Add new tag</a>	

You can add 49 more tags.

**Feedback** Looking for language selection? Find it in the new [Unified Settings](#).

**Network ACLs (1/3)** Info

You successfully created acl-07567ff648e10bb5b / publicacl.

**Actions**

- [View details](#)
- [Edit inbound rules](#)
- [Edit outbound rules](#)
- [Edit subnet associations](#) (highlighted)
- [Manage tags](#)
- [Delete network ACLs](#)
- [Troubleshoot](#)
- [Trace network reachability](#)

Name	Network ACL ID	Associated with	Default
-	acl-041e082340def1328	3 Subnets	Yes
<input checked="" type="checkbox"/> publicacl	acl-07567ff648e10bb5b	-	No
-	acl-0a5f5649ea60d42f0	2 Subnets	Yes

**acl-07567ff648e10bb5b / publicacl**

[Details](#) [Inbound rules](#) [Outbound rules](#) [Subnet associations](#) [Tags](#)

**Available subnets (1/2)**

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> public-subnet	subnet-061e8a4e28673b206	acl-0a5f5649ea60d42f0	us-east-2a	10.0.0.0/24	-
<input type="checkbox"/> private-subnet	subnet-0c7a2470ca84e97b8	acl-0a5f5649ea60d42f0	us-east-2b	10.0.1.0/24	-

**Selected subnets**

[Cancel](#) [Save changes](#)

The first screenshot shows the 'Create network ACL' wizard with 'Network ACL settings' and 'Tags' sections. The second screenshot shows the 'Network ACLs (1 / 4)' list with details for 'privatenacl'. The third screenshot shows the 'Edit subnet associations' page for 'privatenacl'.

**Create network ACL**

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.  
privatenacl

VPC to use for this network ACL.  
vpc-0d05e02080ec5714 (custom-vpc)

**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  
Name privatenacl Remove Add new tag

**Network ACLs (1 / 4) Info**

privatenacl acl-0e567442a8d8b3c55 – No Edit subnet associations stom-vpc

– acl-041e082340def1328 3 Subnets Yes Manage tags

publicnac1 acl-07567ff648e10bb5b subnet-061e8a4e28673b206 / public-subnet No Delete network ACLs stom-vpc

– acl-0a5f5649ea60d42f0 subnet-0c7a2470ca84e97b8 / private-subnet Yes Troubleshoot

Trace network reachability

**Edit subnet associations**

Change which subnets are associated with this network ACL.

**Available subnets (1 / 2)**

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
public-subnet	subnet-061e8a4e28673b206	acl-07567ff648e10bb5b / publicnac1	us-east-2a	10.0.0.0/24	–
<input checked="" type="checkbox"/> private-subnet	subnet-0c7a2470ca84e97b8	acl-0a5f5649ea60d42f0	us-east-2b	10.0.1.0/24	–

**Selected subnets**

subnet-0c7a2470ca84e97b8 / private-subnet

Cancel Save changes

## PART – 2 : Access S3 from EC2 Instance with in custom VPC

- Launch amazon linux EC2 instance as configure like, Network – select custom VPC, subnet – select public subnet , Auto-assign public IP – Enable it. And launch it.
- To get access S3 to the EC2 instance , Create a IAM role of S3 and attach it to the EC2 instance(app-server-1) as , select machine --> action --> security --> Modify IAM role --> select the role of S3 --> update IAM role.

- Connect app-server-1, and switch to root user the give the command as “aws s3 ls” we can get the list buckets which are available in the Amazon S3.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-0d05e0208e0ec5714 | custom-vpc Create new VPC

Subnet: subnet-061e8a4e28673b206 | public-subnet | us-east-1 Create new subnet  
251 IP Addresses available

Auto-assign Public IP: Enable

Hostname type: Use subnet setting (IP name)

DNS Hostname:

- Enable IP name IPv4 (A record) DNS requests
- Enable resource-based IPv4 (A record) DNS requests
- Enable resource-based IPv6 (AAAA record) DNS requests

Placement group: Add instance to placement group

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: app-server-1

Description: app-server-1

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

i-0104b1f2fda0cbcf1 (app-server-1)  
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

Amazon Linux 2 AMI

http://aws.amazon.com/amazon-linux-2/ [ec2-user@ip-10-0-0-201 ~]\$

**Instances (1/1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Actions
app-server-1	i-0104b1f2fda0cbcf1	Running	t2.micro	2/2 checks passed	Actions ▾

Actions ▾

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security**
- Image and templates
- Monitor and troubleshoot

**Instance: i-0104b1f2fda0cbcf1 (app-server-1)**

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

### PART – 3 : Elastic File System with in custom VPC

#### Step 1:

- Goto to aws console launch one more amazon linux instance assign name as app-server-2 , app-server-1 are already created. 2 instances launch from us-east-2a public subnet from custom vpc.
- Connect the app-server-1 (instance) and run below commands and instructions,

sudo su - //switch root user

yum update -y //update packages

```
sudo yum install -y amazon-efs-utils //to install EFS utility
```

```
sudo service nfs status //to check the status of nfs at initial it will be inactive state we need to active it by below command
```

```
sudo service nfs start
```

```
sudo service nfs status //now the status of nfs has been activated.
```

```
mkdir efs //create a directory of efs is mandatory
```

- Continue and do the same process to the app-server-2 machine.

The screenshot shows two terminal sessions in the AWS CloudWatch interface. The top session is for the app-server-1 instance, and the bottom session is for another instance. Both sessions show the execution of commands to install the Amazon EFS utilities and start the NFS service. The output includes package details, dependency information, and logs from the system and NFS services.

```
Last login: Thu Sep 15 03:14:15 2022 from ec2-3-16-146-4.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo yum install amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-10-0-0-201 ~]$ sudo yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-10-0-0-201 ~]$ sudo service nfs start
[ec2-user@ip-10-0-0-201 ~]$ ls
[ec2-user@ip-10-0-0-201 ~]$ cd /var/nfs
[ec2-user@ip-10-0-0-201 ~]$ touch testfile
[ec2-user@ip-10-0-0-201 ~]$ exit
i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

Installing : amazon-efs-utils-1.33.3-1.amzn2.noarch
Verifying : amazon-efs-utils-1.33.3-1.amzn2.noarch
Verifying : stunnel-4.56-6.amzn2.0.3.x86_64
2/2
1/2
2/2

Installed:
amazon-efs-utils.noarch 0:1.33.3-1.amzn2

Dependency Installed:
stunnel.x86_64 0:4.56-6.amzn2.0.3

Complete!
[ec2-user@ip-10-0-0-201 ~]$ sudo service nfs start
Redirecting to /bin/systemctl start nfs.service
[ec2-user@ip-10-0-0-201 ~]$ sudo service nfs status
Redirecting to /bin/systemctl status nfs.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset: disabled)
     Active: active (exited) since Thu 2022-09-15 03:23:59 UTC; 5s ago
       Process: 4766 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
      Process: 4762 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid` (code=exited, status=0/SUCCESS)
      Process: 4760 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Main PID: 4766 (code=exited, status=0/SUCCESS)
    CGroup: /system.slice/nfs-server.service

Sep 15 03:23:58 ip-10-0-0-201.us-east-2.compute.internal systemd[1]: Starting NFS server and services...
Sep 15 03:23:59 ip-10-0-0-201.us-east-2.compute.internal systemd[1]: Started NFS server and services.
[ec2-user@ip-10-0-0-201 ~]$
```

```

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Installed: amazon-efs-utils.noarch 0:1.33.3-1.amzn2
Dependency Installed: stunnel.x86_64 0:4.56-6.amzn2.0.3
Complete!
[ec2-user@ip-10-0-0-201 ~]$ sudo service nfs start
Redirecting to /bin/systemctl start nfs.service
[ec2-user@ip-10-0-0-201 ~]$ sudo service nfs status
Redirecting to /bin/systemctl status nfs.service
● nfs-server.service - NFS server and services
    Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset: disabled)
    Active: active (exited) since Thu 2022-09-15 03:23:59 UTC; 5s ago
      Process: 4766 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
     Process: 4762 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid` (code=exited, status=0/SUCCESS)
    Process: 4760 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Main PID: 4766 (code=exited, status=0/SUCCESS)
     CGroup: /system.slice/nfs-server.service

Sep 15 03:23:58 ip-10-0-0-201.us-east-2.compute.internal systemd[1]: Starting NFS server and services...
Sep 15 03:23:59 ip-10-0-0-201.us-east-2.compute.internal systemd[1]: Started NFS server and services.
[ec2-user@ip-10-0-0-201 ~]$ mkdir efs
[ec2-user@ip-10-0-0-201 ~]$ ll
total 0
drwxrwxr-x 2 ec2-user ec2-user 6 Sep 15 03:24 efs ✓
[ec2-user@ip-10-0-0-201 ~]$ 

i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Installed: amazon-efs-utils.noarch 0:1.33.3-1.amzn2
Dependency Installed: stunnel.x86_64 0:4.56-6.amzn2.0.3
Complete!
[root@ip-10-0-0-163 ~]# sudo service nfs start
Redirecting to /bin/systemctl start nfs.service
[root@ip-10-0-0-163 ~]# sudo service nfs status
Redirecting to /bin/systemctl status nfs.service
● nfs-server.service - NFS server and services
    Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset: disabled)
    Active: active (exited) since Thu 2022-09-15 03:27:09 UTC; 4s ago
      Process: 10137 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
     Process: 10133 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid` (code=exited, status=0/SUCCESS)
    Process: 10131 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Main PID: 10137 (code=exited, status=0/SUCCESS)
     CGroup: /system.slice/nfs-server.service

Sep 15 03:27:08 ip-10-0-0-163.us-east-2.compute.internal systemd[1]: Starting NFS server and services...
Sep 15 03:27:09 ip-10-0-0-163.us-east-2.compute.internal systemd[1]: Started NFS server and services.
[root@ip-10-0-0-163 ~]# mkdir efs
[root@ip-10-0-0-163 ~]# ll
total 0
drwxr-xr-x 2 root root 6 Sep 15 03:27 efs
[root@ip-10-0-0-163 ~]# 

i-0909e94ff5b863647 (app-server-2)
PublicIPs: 18.222.33.174 PrivateIPs: 10.0.0.163

```

## Step 2: Now goto the Amazon EFS >> create filesystem

- Name > as per requirement give the name
- Storage class > standard.
- For the practice purpose uncheck the Enable the automatic backups.
- And remaining things are kept default , move to 2<sup>nd</sup> step network access.
- Network access , make sure the instances and efs are in same custom vpc, to the mount targets add the security group which consists of NFS protocol. and create file system.

**Elastic File System**

File systems  
Access points

AWS Backup   
AWS DataSync   
AWS Transfer

Documentation

# 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 system**

Create an EFS file system with service recommended settings.

**Create file system**

Amazon EFS > File systems > Create

Step 1  
**File system settings**

Step 2  
Network access

Step 3 - optional  
File system policy

Step 4  
Review and create

### File system settings

#### General

Name - optional  
Name your file system.  
 Name can include letters, numbers, and +-=\_.:/ symbols, up to 256 characters.

Storage class [Learn more](#)

Standard  
Stores data redundantly across multiple AZs

One Zone  
Stores data redundantly within a single AZ

Automatic backups

Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. [Learn more](#)

Enable automatic backups

We recommend that you create a backup policy for your file system

Lifecycle management

EFS Intelligent-Tiering uses Lifecycle Management to automatically achieve the right price and performance blend for your application by moving your files between the Standard and Standard-Infrequent Access storage classes. [Learn more](#)

Activate Windows

Go to Settings to activate Windows.

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

**Step 3:** Into the file system > attach, in the attach to Mount the EFS file system which are created are on a linux instance by using the bellow steps, connect the app-server-2 instance ,

- Sudo su -
- sudo mount -t nfs4 -o -----:/efs
- cd efs
- touch file
- ll

//this taken from the EFS attach Mount  
 via IP > using the NFS client  
 //open the efs directory  
 //create the new file  
 //check the list

Amazon EFS > File systems > fs-0ae692aee46d8c754

### my-efs (fs-0ae692aee46d8c754)

**General**

Performance mode	Automatic backups
General Purpose	Disabled
Throughput mode	Encrypted
Bursting	No
Lifecycle management	File system state
Transition into IA: None	Available
Transition out of IA: None	DNS name
Availability zone	No mount targets available
Standard	

**Attach**

Mount your Amazon EFS file system on a Linux instance. [Learn more](#)

Mount via DNS       Mount via IP

Availability zone: us-east-2a

Using the NFS client:

```
nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.131:/ efs
```

See our user guide for more information. [User guide](#)

**Terminal Output:**

```
[root@ip-10-0-0-163 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.131:/ efs ✓
[root@ip-10-0-0-163 ~]# cd efs
[root@ip-10-0-0-163 efs]# touch file ✓
[root@ip-10-0-0-163 efs]# ll
total 4
-rw-r--r-- 1 root root 0 Sep 15 03:40 file ✓
[root@ip-10-0-0-163 efs]#
```

i-0909e94ff5b863647 (app-server-2)  
PublicIPs: 18.222.33.174 PrivateIPs: 10.0.0.163

**Step 4:** connect the another instance (app-server-1) follow the below commands.

- sudo su -
- sudo mount -t nfs4 -o -----:/ efs //this taken from the EFS attach Mount via IP > using the NFS client
- cd efs //open the efs directory
- ll // check the list and we will get the file which are created in the app-server-2 instance.

```
[root@ip-10-0-0-201 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.131:/ efs ✓
[root@ip-10-0-0-201 ~]# cd efs
[root@ip-10-0-0-201 efs]# ll
total 4
-rw-r--r-- 1 root root 0 Sep 15 03:40 file ✓
[root@ip-10-0-0-201 efs]#
```

i-0104b1f2fda0cbcfc1 (app-server-1) ✓  
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

This part of exercise will provide some details about the how the Amazon EFS are works on instances. Since the EFS file system supports accessing files mounted on EC2 instances simultaneously.

#### PART – 4: Setup RDS and connect with EC2 in AWS custom VPC

Setup RDS and connect with EC2, this complete setup are into the creating on private subnet for secure the Database.

##### Step 1:

- Launch amazon linux EC2 instance (Database-server) as configure like, Network – select custom VPC, subnet – select private subnet , Auto-assign public IP – Disable it, Security group give the Mysql/aurora 3306 port and protocol And launch it.
- The Database-server are in private subnet which doesn't assign the public IP. We are unable to connect it direct. We need to take the access of Database-server from the app-server-1.
- Then connect the "app-server-1" , switch root user and create a file and add the pem key encrypted data into it.

```
#vim ec2-key.pem //create a file and it on edit mode add the data of  
Database-server pem file (which are used to launch  
instance).save it
```

```
#chmod 400 ec2-key.pem // To protect a file against accidental overwriting
```

```
#ssh ec2-user@10.0.1.77 -i ec2-key.pem //10.0.1.77 is the private IP of Database-server
```

Now enter “yes” command then we will login into the Database-server. Switch to root user, ping google.com check there is no response from google website due to the internet access are not access into this server.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-0d05e0208e0ec5714   custom-vpc <input type="button" value="Create new VPC"/>	
Subnet	subnet-0c7a2470ca84e97b8   private-subnet   us-east-1 <input type="button" value="Create new subnet"/> 251 IP Addresses available	
Auto-assign Public IP	<input type="button" value="Use subnet setting (Disable)"/>	
Hostname type	<input type="button" value="Use subnet setting (IP name)"/>	
DNS Hostname	<input type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	<input type="button" value="Open"/>	

**Review and Launch** Next: Add Storage

**Step 5: Add Tags**

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name	<input type="text" value="Database-server"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add another tag	(Up to 50 tags maximum)					

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:	<input checked="" type="radio"/> Create a new security group	<input type="radio"/> Select an existing security group		
Security group name:	<input type="text" value="prvt"/>			
Description:	<input type="text" value="prvt"/>			
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0
MySQL/Aurora	TCP	3306	Anywhere	0.0.0.0/0, ::/0

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

```

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 03:42:25 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ ls
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 11 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-0-201 ~]$ s
i-0104b1f2fd00cbcf1 (app-server-1)
Public IPs: 3.128.255.66 Private IPs: 10.0.0.201

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
i-0104b1f2fd00cbcf1 (app-server-1)
Public IPs: 3.128.255.66 Private IPs: 10.0.0.201

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ cat ec2-key.pem
-----BEGIN RSA PRIVATE KEY-----  

MIIIPcwIBAAKCAQEAwt90J1NNRCo2DyXnL8A1SS91p0oIg44x1nUvqg5d7GnNYSU  
Sc51snbpm9m2mzmaW$/FEBXqpWSkzxS0gozbh:xVm0uBcctIAGe0ufiCS2gjzJNw  
GNY2zfPiEyo+TVY5uJAB8Gbxxhunpuhn1lwtx1TThpYo!Hv1Vxh1lxv1l2o0igrcY4  
e1HmaF80UIlpCmnc25LXPQa3xHo/r8T/Fozdv9u19P49ehub2l1mna4g33LJwqTq7H  
wM225Gu0R5grhc++NR5gvSLP4uhUtovBTtjwRAiCCt7hdByhlyex+pNC2lMjSzga  
ymQfz1thYc07saWg5nwyywC51L7gbijju1Agw1DQRABao1BAEcdJLJtngluuRaPEX  
1QeToqhpOnTAYB3x3pV++oh1lQvRCODNakfml7ledahwv9.13andh0S6sJcv6j7Ct  
LzhvHmW/JkuwuCrj2kbDKFrh63m4KwNwGh4LlgKu44D0+R87ofY177c7XW  
04:mcG1KRG0X18sp4g2ndN25mFRFKTS-8nzo981lc0Qg1Yr3v-Q1/54:SR1FHc0mYL  
4dv1lcuQyQbko/j8b3bnVsLzCgcCgkmtetzuUca0ldefj1lKV/CYVfHOMg1D3K8  
ek8Thj:6fkVlozAy9eKMF13+*RHj2AOXhoiHDK8E6E6w41L751585xfPKEDX  
pdZs+6EcgiE277+CWSbtXftm47bZAK6KFmrxxfrD8kkuCaW2RtsXdwvssw0N2  
Mie8tmsYJhQs92pOXb2meQltaa0o5N-H615WXRmEfgrs9y8+845wZ1pbc  
zm41gleiy2Ghpgy1/SzuuYhoXu1GJ5saJ1DRehJgyV/KJMaDRVCEgiEAdhnt  
Dr4m1gJg1gzz4V+xsRoP4Fc1FMnfPQUn1h2zouQOsD001pc2/PavCb4KzaOch  
14pxXGhkvz5...  
sem5dUMq/kphWMjUNsr:MGXZKzH19+bWukTsCgYYbaelyhvXHURKjH14Md  
+NHukzWm0fb7czTkjoubxERpFgh1KkqdRTvRNsceznf10Bsc+sawNQGdkER012C  
WXEavvsklh4:Ha/SiYxvhMewYDBtPbc7PgMmAngbTtQXJXWTTYFJGxXm9VfGxq8  
Lol1Mopcschy0f6fc1BeqcQK8gFl+t+NmZy/TOBVXJ/bf3kelyvDQq1i9c4g3hbD  
Ex0h0Lk9tDzdr/svqB8SFzPdln9c1UaLi1FFTz9C3btFaM+mqAr6gVo6sIU  
MZTnE7C1L1lDbY1Wbm11fp1BGEM0utw01dcZ2z3t:EkxJQ9Wza5mW+kQ/GG1  
nu4xAcGBA/BPrgnqW7d+gwPg/5ca9k9GeogAZZXCGmJ/K62BCMMUKgbfqRnAkGn  
cU3NrfitgO9v8/+hdWv+C9aSiRK8HFWzleis1WJVJyyk2i17k9eP87WH02Hb  
xip1k+owizW9rvK7chNx1MtwX61UhxSyu0FtbTCSJtXki9Jk5  
----END RSA PRIVATE KEY----
```

```
AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
```

i-0104b1f2fd0cbcf1 (app-server-1)

PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```
AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
[root@ip-10-0-0-201 ~]# ll
total 8
-r----- 1 root root 1675 Sep 15 08:08 ec2-key.pem
drwxr-xr-x 2 root root 6144 Sep 15 03:44 efs
[root@ip-10-0-0-201 ~]#
```

i-0104b1f2fd0cbcf1 (app-server-1)

PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```
AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
[root@ip-10-0-0-201 ~]# ll
total 8
-r----- 1 root root 1675 Sep 15 08:08 ec2-key.pem
drwxr-xr-x 2 root root 6144 Sep 15 03:44 efs
[root@ip-10-0-0-201 ~]# ssh ec2-user@10.0.1.77 -i ec2-key.pem
```

i-0104b1f2fd0cbcf1 (app-server-1)

PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Last login: Thu Sep 15 08:01:43 2022 from ec2-3-16-146-5.us-east-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
[root@ip-10-0-0-201 ~]# ll
total 8
-r----- 1 root root 1675 Sep 15 08:08 ec2-key.pem
drwxr-xr-x 2 root root 6144 Sep 15 03:44 efs
[root@ip-10-0-0-201 ~]# ssh ec2-user@10.0.1.77 -i ec2-key.pem
The authenticity of host '10.0.1.77' (10.0.1.77) can't be established.
ECDSA key fingerprint is SHA256:blEu+RzXRMGAJMcBFcF3iag7eUOkKQpeXUVdOmhC3+Y.
ECDSA key fingerprint is MD5:26:90:f0:bf:ee:12:b1:ce:94:ad:11:50:a1:cc:83:83.
Are you sure you want to continue connecting (yes/no)? yes

```

i-0104b1f2fd0cbcf1 (app-server-1)  
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
[ec2-user@ip-10-0-0-201 ~]$ sudo su -
Last login: Thu Sep 15 08:04:39 UTC 2022 on pts/0
[root@ip-10-0-0-201 ~]# vim ec2-key.pem
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
[root@ip-10-0-0-201 ~]# ll
total 8
-r----- 1 root root 1675 Sep 15 08:08 ec2-key.pem
drwxr-xr-x 2 root root 6144 Sep 15 03:44 efs
[root@ip-10-0-0-201 ~]# ssh ec2-user@10.0.1.77 -i ec2-key.pem
The authenticity of host '10.0.1.77' (10.0.1.77) can't be established.
ECDSA key fingerprint is SHA256:blEu+RzXRMGAJMcBFcF3iag7eUOkKQpeXUVdOmhC3+Y.
ECDSA key fingerprint is MD5:26:90:f0:bf:ee:12:b1:ce:94:ad:11:50:a1:cc:83:83.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.77' (ECDSA) to the list of known hosts.

```

i-0104b1f2fd0cbcf1 (app-server-1)  
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```

AWS Services Search for services, features, blogs, docs, and more [Alt+S]
[root@ip-10-0-0-201 ~]# chmod 400 ec2-key.pem
[root@ip-10-0-0-201 ~]# ll
total 8
-r----- 1 root root 1675 Sep 15 08:08 ec2-key.pem
drwxr-xr-x 2 root root 6144 Sep 15 03:44 efs
[root@ip-10-0-0-201 ~]# ssh ec2-user@10.0.1.77 -i ec2-key.pem
The authenticity of host '10.0.1.77' (10.0.1.77) can't be established.
ECDSA key fingerprint is SHA256:blEu+RzXRMGAJMcBFcF3iag7eUOkKQpeXUVdOmhC3+Y.
ECDSA key fingerprint is MD5:26:90:f0:bf:ee:12:b1:ce:94:ad:11:50:a1:cc:83:83.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.77' (ECDSA) to the list of known hosts.

```

[root@ip-10-0-0-201 ~]# ping google.com
PING google.com (142.250.191.206) 56(84) bytes of data.

^C
--- google.com ping statistics ---
63 packets transmitted, 0 received, 100% packet loss, time 63488ms

[root@ip-10-0-0-201 ~]#

i-0104b1f2fd0cbcf1 (app-server-1)  
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

**Step 2:** NAT Gateway is a device used to enable the Database-server instance in a private subnet to connect to the internet. Create natgateway and route table as shown below,

NAT gateway,

- From the VPC dashboard select NAT gateway and click create NAT gateway.
- Give the name of NAT gateway for identification.
- Subnet – public subnet, connectivity type public, allocate elastic IP and create NAT gateway.

Route tables,

- From the VPC dashboard select Route tables and click create Route tables.
- Name – private RT, VPC – select the custom VPC and create it.
- Associate this Route tables with the private subnet from the subnet association --> edit subnet associations --> select private subnet and save associates.
- Now goto the Routes --> Edit routes --> destination – 0.0.0.0/0 and Target – NAT gateway select it which are previously created and save changes.

Now get back to the Database-server which are access from the app-server-1, then ping google.com we can get the response. Since we build a successful internet connection to the private subnet's Database-server.

The screenshot displays two windows from the AWS VPC service. The top window is titled 'NAT gateways' and shows a list of existing NAT gateways. A red arrow points to the 'Create NAT gateway' button at the top right. The bottom window is titled 'Create NAT gateway' and contains the following fields:

- NAT gateway settings**
- Name - optional**: my-nat-gw
- Subnet**: subnet-061e8a4e28673b206 (public-subnet)
- Connectivity type**: Public (radio button selected)
- Elastic IP allocation ID**: eipalloc-002f7017cd1d0fd5b
- Allocate Elastic IP** button

Screenshot of the AWS VPC NAT gateway configuration page.

**Details**

NAT gateway ID nat-02e5c0845e624ed0d	Connectivity type Public	State <b>Available</b>	State message Info
NAT gateway ARN arn:aws:ec2:us-east-2:881832161071:natgateway/nat-02e5c0845e624ed0d	Elastic IP address 3.17.8.225	Private IP address 10.0.0.125	Network interface ID eni-07e276c82ae2ca4bd
VPC vpc-0d05e0208e0ec5714 / custom-vpc	Subnet subnet-061e8a4e28673b206 / public-subnet	Created Thursday, September 15, 2022 at 13:57:51 GMT+5:30	Deleted -

**Monitoring**

**Tags**

Screenshot of the AWS Route tables management page.

**Route tables (3) Info**

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-00b0630c85fe35ad6	-	-	Yes	vpc-04b0d41543300892e
public-RT	rtb-098243f3b291ba7f3	subnet-061e8a4e28673b206	-	No	vpc-0d05e0208e0ec5714   cus...
-	rtb-0d882d7b4bc906647	-	-	Yes	vpc-0d05e0208e0ec5714   cus...

**Create route table**

Screenshot of the AWS Create route table configuration page.

**Create route table**

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.  
**private-RT**

**VPC**  
The VPC to use for this route table.  
vpc-0d05e0208e0ec5714 (custom-vpc)

**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**  
Name private-RT

**Add new tag**

You can add 49 more tags.

Screenshot of the AWS VPC Route Table Details page.

**Route table ID:** rtb-0c74dfeb375f1a65c

**Main:** No

**Owner ID:** 881832161071

**Explicit subnet associations:** -

**Edge associations:** -

**Subnet associations:** (Selected)

**Explicit subnet associations (0):**

**Edit subnet associations** button highlighted with a red arrow.

Screenshot of the "Edit subnet associations" page.

**Available subnets (1/2):**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
public-subnet	subnet-061e8a4e28673b206	10.0.0.0/24	-	rtb-098243f3b291ba7f3 / public-RT
<b>private-subnet</b>	subnet-0c7a2470ca84e97b8	10.0.1.0/24	-	Main (rtb-0d882d7b4bc906647)

**Selected subnets:**

- subnet-0c7a2470ca84e97b8 / private-subnet

**Save associations** button highlighted with a red arrow.

Screenshot of the AWS VPC Route Table Details page after saving subnet associations.

**Route table ID:** rtb-0c74dfeb375f1a65c

**Main:** No

**Owner ID:** 881832161071

**Explicit subnet associations:** subnet-0c7a2470ca84e97b8 / private-subnet

**Edge associations:** -

**Routes (1):**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

**Edit routes** button highlighted with a red arrow.

Screenshot of the AWS VPC Route Tables interface showing the creation of a route from 0.0.0.0/0 to a NAT gateway target.

**Edit routes**

Destination	Target	Status	Propagated
10.0.0.16	local	Active	No
0.0.0.0/0	nat- nat-02e5c0845e624ed0d (my-nat-gw)	-	No

Add route

Cancel Preview Save changes

Screenshot of the AWS VPC Route Tables interface showing the successful update of the route table.

Updated routes for rtb-0c74dfb375f1a65c / private-RT successfully

Route	Status	Propagation
rtb-0c74dfb375f1a65c	No	subnet-0c7a2470ca84e97b8 / private-subnet
VPC	Owner ID vpc-0d05e0208e0ec5714   custom-vpc	

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-02e5c0845e624ed0d	Active	No
10.0.0.16	local	Active	No

Activate Windows Go to Settings to activate Windows

Screenshot of a terminal window showing ping results between the EC2 instance and Google's public IP.

```
[root@ip-10-0-1-77 ~]# ping google.com
PING google.com (142.250.191.206) 56(84) bytes of data.

^C
--- google.com ping statistics ---
63 packets transmitted, 0 received, 100% packet loss, time 63488ms

[root@ip-10-0-1-77 ~]# ping google.com
PING google.com (172.217.0.174) 56(84) bytes of data.
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=1 ttl=93 time=19.7 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=2 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=3 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=4 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=5 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=6 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=7 ttl=93 time=18.9 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=8 ttl=93 time=18.8 ms

64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=9 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=10 ttl=93 time=18.8 ms
64 bytes from yzz08s10-in-f174.le100.net (172.217.0.174): icmp_seq=11 ttl=93 time=18.8 ms
^C
--- google.com ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10015ms
rtt min/avg/max/mdev = 18.803/18.944/19.735/0.265 ms
[root@ip-10-0-1-77 ~]# i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201
```

### Step 3 : Creating MySql Database and setup to EC2 (Database-server).

- Goto aws RDS dashboard then click “create database” .
- Engine options – MySql , version – select updated version, Templates – free tier.
- Credential settings --> Master username – admin, Master password – give the own password.

- Connectivity --> VPC – custom vpc, public access – no , VPC security group – add security group which are contain MySQL – 3306 port and protocol, Availability zone – private az.
- Create It. Copy the End point of database which will used to take this into the ec2.

**Amazon RDS**

**Dashboard**

Databases  
Query Editor  
Performance insights  
Snapshots  
Automated backups  
Reserved instances  
Proxies

Subnet groups  
Parameter groups  
Option groups  
Custom engine versions

**Create database**

Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL  
For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies by 2x, experience faster failover typically less than 35 seconds and, get read scalability with two readable standby DB instances by deploying the Multi-AZ DB cluster [Learn more](#)

**Create database**

Or, Restore Multi-AZ DB Cluster from Snapshot

**Resources**

You are using the following Amazon RDS resources in the US East (Ohio) region (used/quota)

DB Instances (0/40)	Parameter groups (0)
Allocated storage (0 TB/100 TB)	Default (0)
Increase DB instances limit <input checked="" type="checkbox"/>	Custom (0/100)
DB Clusters (0/40)	Option groups (0)
Reserved instances (0/40)	Default (0)
Snapshots (0)	Custom (0/20)
Manual Subnet groups (0/50)	

**Additional information**

Getting started with RDS  
Overview and features  
Documentation  
Articles and tutorials  
Data import guide for MySQL  
Data import guide for Oracle  
Data import guide for SQL Server

**Create database**

**Choose a database creation method**

Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

Engine type

- Amazon Aurora
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

Activate Windows  
Go to Settings to activate Windows.

**Known issues/limitations**

Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Version

MySQL 8.0.28

**Templates**

Choose a sample template to meet your use case.

- Production
- Dev/Test
- Free tier

Free tier

This instance is intended for development use outside of a production environment.

**Availability and durability**

Deployment options

The deployment options below are limited to those supported by the engine you selected above.

Multi-AZ DB Cluster - new

Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

Activate Windows  
Go to Settings to activate Windows.

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

**DB instance identifier**

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Credentials Settings**

**Master username** [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter.

**Auto generate a password**

Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

**Confirm password** [Info](#)

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**Compute resource**

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

**Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

**Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** [Info](#) ✓

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Only VPCs with a corresponding DB subnet group are listed.

**DB Subnet group** [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

**Public access** [Info](#)

**Yes**  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to the database.

**Activate Windows**  
Go to Settings to activate Windows.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

**VPC security group (firewall) Info**  
 No  
 RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**Existing VPC security groups**  
 Choose existing  
 Choose existing VPC security groups  
 Create new  
 Create new VPC security group

**Availability Zone Info**  
 us-east-2b

**Additional configuration**

**Database authentication**  
 database-1

**Summary**

DB identifier database-1	CPU <div style="width: 5.42%;">5.42%</div>	Status <span style="color: green;">Available</span>	Class db.t2.micro
Role Instance	Current activity <div style="width: 0%;">0 Connections</div>	Engine MySQL Community	Region & AZ us-east-2b

**Connectivity & security**

Endpoint & port Endpoint <a href="#">database-1.ci0rfzkqjpubs.us-east-2.rds.amazonaws.com</a>	Networking Availability Zone us-east-2b	Security VPC security groups <a href="#">prvt (sg-08afca5a196ab92c1)</a>
Port 3306	VPC <a href="#">custom-vpc (vpc-0d05e0208e0ec5714)</a>	Publicly accessible No
Subnet group		

**Step 4 :** to get the access of MySql Database which are previously created to the Ec2 instance (Database-server) follow the below instructions,

- First connect the app-server-1 and login to the Database-server, by this command- `#ssh ec2-user@10.0.1.77 -i ec2-key.pem`

```
#sudo su - //switch to root user
#yum update -y //update the all packages
#yum install mysql -y //install the myql appliation
#mysql --version //check the version of mysql
#mysql -h databaseendpoint -P 3306 -u admin -p
```

Give password and hit enter, then we get the access of database.

```

Complete!
[root@ip-10-0-1-77 ~]# yum install mysql -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

Activate Windows
AWS Services Search for services, features, blogs, docs, and more [Alt+S]
Downloading packages:
mariadb-5.5.68-1.amzn2.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.x86_64
  Verifying   : 1:mariadb-5.5.68-1.amzn2.x86_64

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2

Complete!
[root@ip-10-0-1-77 ~]# mysql --version
mysql Ver 15.1 Distrib 5.5.68-MariaDB, for Linux (x86_64) using readline 5.1
[root@ip-10-0-1-77 ~]# mysql -h database-1.cxxfzkqipubs.us-east-2.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> 

i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

Activate Windows
Feedback Looking for language selection? Find it in the new Unified Settings [ ]
AWS Services Search for services, features, blogs, docs, and more [Alt+S]
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Go to Settings to activate Windows. Privacy Terms Cookie preferences
Complete!
[root@ip-10-0-1-77 ~]# mysql --version
mysql Ver 15.1 Distrib 5.5.68-MariaDB, for Linux (x86_64) using readline 5.1
[root@ip-10-0-1-77 ~]# mysql -h database-1.cxxfzkqipubs.us-east-2.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.01 sec)

MySQL [(none)]> 

i-0104b1f2fd0cbcf1 (app-server-1)
PublicIPs: 3.128.255.66 PrivateIPs: 10.0.0.201

```

**Conclusion:** completed all the three exercises with in the custom VPC.