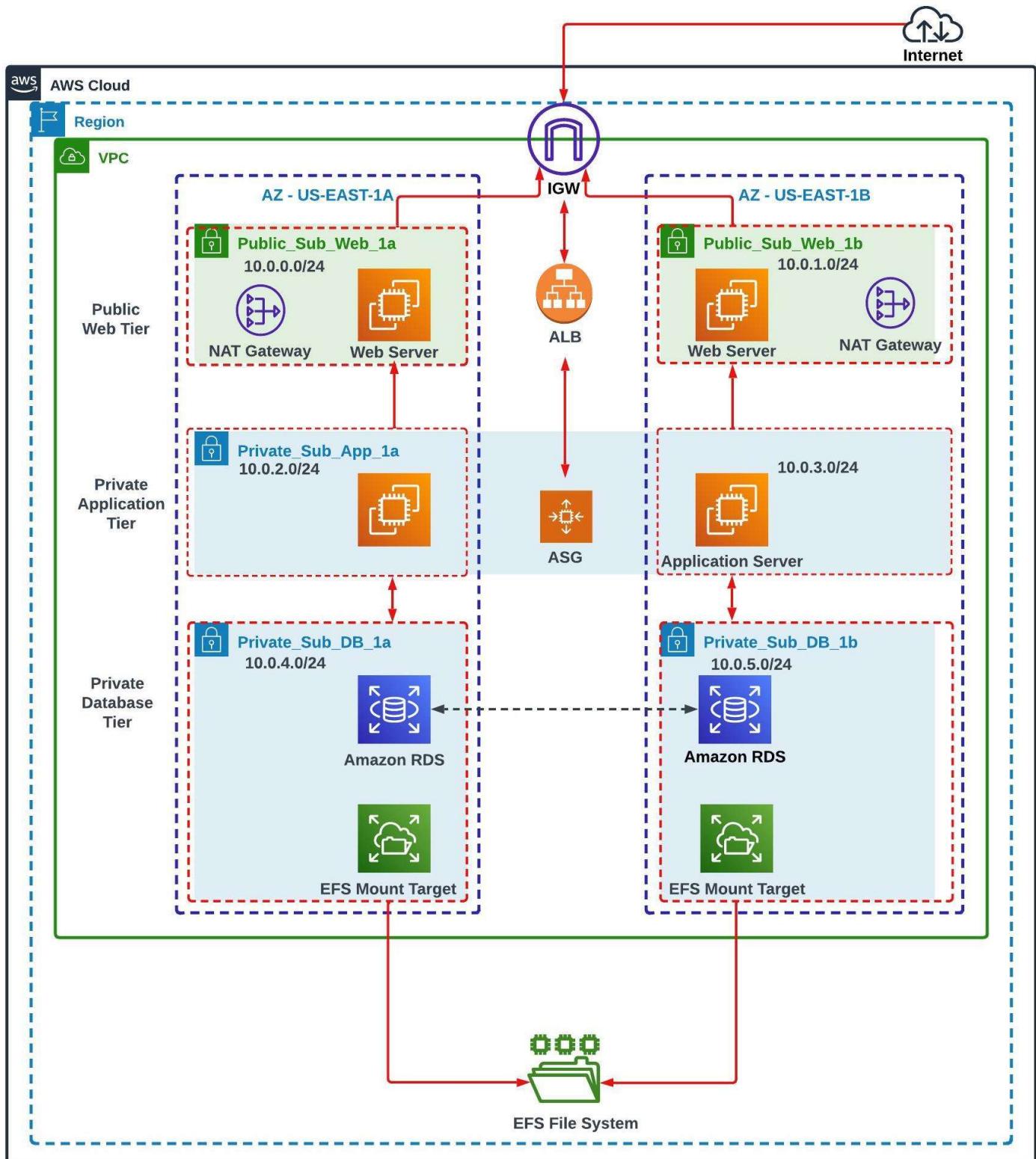


DEPLOYMENT OF A WORDPRESS WEBSITE ON AWS



Deploying a WordPress website on AWS involves several steps setting up the necessary infrastructure and configuring the environment for the various AWS services to host and run the WordPress application.

The specific steps and tools used in the configuration may vary depending on the hosting option chosen and your specific requirements. Detailed documentation and tutorials are available from AWS and the WordPress community to assist you in each stage of the deployment process.

With AWS as our launchpad, let's navigate the vast AWS cosmos together to successfully deploy our WordPress website and have it running seamlessly.

REQUIRED AWS SERVICES

1. VPC WITH PRIVATE AND PUBLIC SUBNETS 10.0.0.0/16
2. EC2
3. SECURITY GROUPS
4. RDS
5. NAT GATEWAY
6. ELASTIC FILE SYSTEM (EFS)
7. APPLICATION LOAD BALANCER (ALB)
8. AUTO SCALING GROUP (ASG)
9. AMAZON CERTIFICATE MANAGER
10. ROUTE 53
11. CLOUDFRONT

However, here is a brief introduction to the process step by step:

STEP 1:

CREATE AN AWS ACCOUNT: If you don't have an AWS account, you'll need to sign up for one at aws.amazon.com. This will provide you with access to the AWS Management Console.

STEP 2:

WE HAVE TO CREATE A VIRTUAL PRIVATE CLOUD (VPC)

First, let's create a VPC.

We also must specify the range of IPv4 CIDR block address for the VPC, and my specified CIDR range will be **10/0.0.0/16**.

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

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

IPv4 CIDR block Info
 IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR

IPv6 CIDR block Info
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info

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 X

Value - optional

VPC_My_Project X

Remove

Add new tag

You can add 49 more tags.

Cancel

Create VPC

VPC > Your VPCs > vpc-09e54287b0cadcace

vpc-09e54287b0cadcace / VPC_My_Project

Actions ▾

Details Info

VPC ID
vpc-09e54287b0cadcace

Available

DNS hostnames
Disabled

DNS resolution
Enabled

Tenancy
Default

DHCP option set
dopt-0df130f81ab802fe7

Main route table
rtb-03e6603061628bc6

Main network ACL
acl-0afb18cb59013b941

Default VPC
No

IPv4 CIDR
10.0.0.0/16

IPv6 pool
-

IPv6 CIDR (Network border group)
-

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
-

Owner ID
763176333159

My VPC for this project (**VPC_My_Project**) has been successfully created.

STEP 3:

NOW, WE HAVE TO CREATE AN INTERNET GATEWAY (IGW)

Let's create an Internet Gateway (**IGW**), after that, we must select my VPC and then attach the internet gateway to the VPC.

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag

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

Tags - optional

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 X**Value - optional** XRemoveAdd new tag

You can add 49 more tags.

CancelCreate internet gateway

Attach to VPC (igw-0e1e2f6ca9c065b82) 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.

 X▶ AWS Command Line Interface commandCancelAttach internet gateway

The screenshot shows the AWS VPC Internet gateways page. A specific Internet gateway, 'igw-0e1e2f6ca9c065b82 / IGW_My_Project', is selected. The 'Details' tab is active. Key information displayed includes:

- Internet gateway ID:** igw-0e1e2f6ca9c065b82
- State:** Attached
- VPC ID:** vpc-09e54287b0cadace | VPC_My_Project
- Owner:** 763176333159

In the 'Tags' section, there is one tag entry:

Key	Value
Name	IGW_My_Project

The Internet gateway (**IGW_My_Project**) was successfully created and attached to our VPC.

STEP 4:

NEXT WE HAVE TO CONFIGURE TWO (2) NAT GATEWAYS

Let's create two (2) NAT Gateways for redundancy.

The main function of the NAT Gateway is to allow our EC2 instances in the private subnets to have access to the internet for software updates, package installation, downloads etc.

Navigate to VPC, scroll down to select NAT gateways, and click on create NAT gateway.

Let's specify the public subnets where we want the NAT gateways to reside when created. Let's select our first public subnet "**Public_Sub_Web_1a**" for this configuration, and then select connectivity type as public.

We must click on "Allocate Elastic IP" to allocate an Elastic IP address to the NAT gateway.

Click on create NAT gateway and repeat same process to create the second NAT gateway in the second public subnet "**Public_Sub_Web_1b**".

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional

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

The name can be up to 256 characters long.

Subnet

Select a subnet in which to create the NAT gateway.

Connectivity type

Select a connectivity type for the NAT gateway.

- Public
- Private

Elastic IP allocation ID Info

Assign an Elastic IP address to the NAT gateway.

► Additional settings Info

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.

nat-061e6f27aa751aaaf / NATGW1_My_Project

Actions ▾

Details [Info](#)

NAT gateway ID
 nat-061e6f27aa751aaaf

Connectivity type
 Public

State
 Pending

State message [Info](#)
 -

NAT gateway ARN
 arn:aws:ec2:us-east-1:765176533159:natgateway/nat-061e6f27aa751aaaf

Primary public IPv4 address
 -
 Subnet
 subnet-0552f97133c2ed30b /
 Public_Sub_Web_1a

Primary private IPv4 address
 -
 Created
 Sunday, June 4, 2023 at 15:57:35 CDT

Primary network interface ID
 -
 Deleted
 -

VPC
 vpc-09e54287b0cadcae / VPC_My_Project

The first NAT Gateway (**NATGW1_My_Project**) was successfully created.

Now let's create the second NAT Gateway.

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - *optional*

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

NATGW2_My_Project

The name can be up to 256 characters long.

Subnet

Select a subnet in which to create the NAT gateway.

subnet-0a00c31de3ca2a263 (Public_Sub_Web_1b) ▾

Connectivity type

Select a connectivity type for the NAT gateway.

- Public
- Private

Elastic IP allocation ID [Info](#)

Assign an Elastic IP address to the NAT gateway.

eipalloc-0e9530603c2563dfe ▾

Allocate Elastic IP

 Additional settings [Info](#)

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 X

Value - optional

NATGW2_My_Project X

Remove

Add new tag

You can add 49 more tags.

Cancel

Create NAT gateway

nat-01ced9116340b26ed / NATGW2_My_Project			
Actions ▾			
Details Info			
NAT gateway ID nat-01ced9116340b26ed	Connectivity type Public	State Pending	State message Info -
NAT gateway ARN arn:aws:ec2:us-east-1:763176333159:natgateway/nat-01ced9116340b26ed	Primary public IPv4 address -	Primary private IPv4 address -	Primary network interface ID -
VPC vpc-09e54287b0cadcace / VPC_My_Project	Subnet subnet-0a00c31de3ca2a263 / Public_Sub_Web_1b	Created Sunday, June 4, 2023 at 16:00:03 CDT	Deleted -

The second NAT Gateway (**NAT2_My_Project**) has been successfully created.

STEP 5:

NEXT WE NEED TO CREATE TWO PUBLIC SUBNETS

Next, we need to create 2 Subnets, we can do this by specifying a subnet name, the availability zone and the IPv4 CIDR block for each subnet.

The CIDR blocks assigned to the subnets must be derived from my main VPC CIDR block **10.0.0.0/16**.

Create subnet Info

VPC

VPC ID

Create subnets in this VPC.

vpc-09e54287b0cadcace (VPC_My_Project) ▾

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 2

Subnet name

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

Public_Sub_Web_1a

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.

US East (N. Virginia) / us-east-1a ▾

IPv4 CIDR block Info

Q 10.0.0.0/24 X

▼ Tags - optional

Key

Value - optional

Q Name X

Q Public_Sub_Web_1a X

Remove

Add new tag

You can add 49 more tags.

Remove

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



You can add 49 more tags.

VPC > Subnets > subnet-0552f97133c2ed30b > Edit subnet settings

Edit subnet settings [Info](#)

Subnet

Subnet ID

Name

Auto-assign IP settings [Info](#)

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

 Enable auto-assign public IPv4 address [Info](#)

Edit subnet settings Info

Subnet	
Subnet ID subnet-0a00c31de3ca2a263	Name Public_Sub_Web_1b

Auto-assign IP settings <small>Info</small>
Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.
<input checked="" type="checkbox"/> Enable auto-assign public IPv4 address <small>Info</small>

You have successfully created 2 subnets: subnet-0552f97133c2ed30b, subnet-0a00c31de3ca2a263

Subnets (2) <small>Info</small>							
<input type="text"/> Filter subnets							
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available
<input type="checkbox"/>	Public_Sub_Web_1a	subnet-0552f97133c2ed30b	Available	vpc-09e54287b0cadace VPC...	10.0.0.0/24	-	251
<input type="checkbox"/>	Public_Sub_Web_1b	subnet-0a00c31de3ca2a263	Available	vpc-09e54287b0cadace VPC...	10.0.1.0/24	-	251

The first two public subnets have been successfully created as we can see from the above screenshots.

It is important to note that the public subnets (**Public_Sub_Web_1a**) and (**Public_Sub_Web_1b**) will be used in the Web Tier.

Now let's modify auto-assign IP settings of the newly created public subnets so that when we create EC2 instances, public IPV4 addresses will automatically be assigned to the EC2 instances.

The auto assign IPV4 address has been successfully modified and saved.

STEP 6:

CREATION OF PUBLIC ROUTE TABLES

(a) Create Public Route Table (**RouteT_Public_Sub_Web**)

The screenshot shows the 'Create route table' wizard in the AWS VPC console. It has two main sections: 'Route table settings' and 'Tags'.

Route table settings:

- Name - optional:** A text input field containing 'RouteT_Public_Sub_Web' is highlighted with a red box.
- VPC:** A dropdown menu showing 'vpc-09e54287b0cadcace (VPC_My_Project)' is highlighted with a red box.

Tags:

- A table with columns 'Key' and 'Value - optional'. One row is shown with 'Name' in the key field and 'RouteT_Public_Sub_Web' in the value field, both highlighted with red boxes.
- A button 'Add new tag' is visible.
- A note says 'You can add 49 more tags.'

At the bottom right are 'Cancel' and 'Create route table' buttons, with 'Create route table' also highlighted with a red box.

(b) Edit the route table (**RouteT_Public_Sub_Web**) to route traffic to Internet Gateway (**IGW_My_Project**)

For this configuration, simply click on the Route table (**RouteT_Public_Sub_Web**), select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select Internet gateway (**IGW_My_Project**) as target, then click "Save changes".

VPC > Route tables > rtb-0404eb6be8b54a1

rtb-0404eb6be8b54a1 / RouteT_Public_Sub_Web

[Actions ▾](#)

You can now check network connectivity with Reachability Analyzer

[Run Reachability Analyzer](#) [X](#)

Details		Info	
Route table ID	rtb-0404eb6be8b54a1	Main	No
VPC	vpc-09e54287b0cadace VPC_My_Project	Owner ID	763176333159
Explicit subnet associations	-		
Edge associations	-		

[Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Routes (1)

[Edit routes](#)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

VPC > Route tables > rtb-0404eb6be8b54a1 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
Q 0.0.0.0/0	igw-igw-0e1e2f6ca9c065b82 (IGW_My_Project)	-	No

[Add route](#)

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

(c) Edit subnet association

To edit the subnet association of the route table (**RouteT_Public_Sub_Web**) associated with the two public subnets (**Public_Sub_Web_1a**) and (**Public_Sub_Web_1b**), simply click on "Subnet associations", proceed to click on "Edit subnet associations", select the public subnets (**Public_Sub_Web_1a**) and (**Public_Sub_Web_1b**), then click on "save associations"

NOTE: The Public Subnets (**Public_Sub_Web_1a**) and (**Public_Sub_Web_1b**) that are linked with my route table (**RouteT_Public_Sub_Web**) will now have access to the internet.

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/2)

[Filter subnet associations](#)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> Public_Sub_Web_1a	subnet-0552f97133c2ed30b	10.0.0.0/24	-	Main [rtb-0c3e6603061628bc6]
<input checked="" type="checkbox"/> Public_Sub_Web_1b	subnet-0a00c31de3ca2a263	10.0.1.0/24	-	Main [rtb-0c3e6603061628bc6]

Selected subnets

[subnet-0a00c31de3ca2a263 / Public_Sub_Web_1b X](#) [subnet-0552f97133c2ed30b / Public_Sub_Web_1a X](#)

[Cancel](#) [Save associations](#)

✓ You have successfully updated subnet associations for rtb-0404eb6be8b54a1 / RouteT_Public_Sub_Web.

X

VPC > Route tables > rtb-0404eb6be8b54a1

rtb-0404eb6be8b54a1 / RouteT_Public_Sub_Web

Actions ▾

ⓘ You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

X

Details Info

Route table ID

rtb-0404eb6be8b54a1

Main

No

Explicit subnet associations

2 subnets

Edge associations

–

VPC

vpc-09e54287b0cadcace | VPC_My_Project

Owner ID

763176333159

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (2)

Find subnet association

Edit subnet associations

< 1 > ⚙

Name

Subnet ID

IPv4 CIDR

IPv6 CIDR

Public_Sub_Web_1b

subnet-0a00c31de3ca2a263

10.0.1.0/24

–

Public_Sub_Web_1a

subnet-0552f97133c2ed30b

10.0.0.0/24

–

Subnets without explicit associations (0)

Edit subnet associations

Subnets were successfully associated with route table.

STEP 7:

NEXT WE NEED TO CREATE FOUR (4) PRIVATE SUBNETS

Next, we need to create 4 private Subnets, we can do this by specifying a subnet name, the availability zone and the IPv4 CIDR block for each subnet.

The CIDR blocks assigned to the subnets must be derived from my main VPC CIDR block **10.0.0.0/16**.

VPC > Subnets > Create subnet

Create subnet

Info

VPC

VPC ID

Create subnets in this VPC.

vpc-09e54287b0cadcace (VPC_My_Project)

▼

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet 1 of 4

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



You can add 49 more tags.

Subnet 2 of 4

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



You can add 49 more tags.

Subnet 3 of 4

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



You can add 49 more tags.

Subnet 4 of 4

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



You can add 49 more tags.

You have successfully created 4 subnets: subnet-0080e1bfd672c0e4d, subnet-077941617c18c49d8, subnet-05915ba82b2a4016b, subnet-060e55a78d5518d83

Subnets (4) Info						
Filter subnets				Actions Create subnet		
Subnet ID: subnet-0080e1bfd672c0e4d X				Subnet ID: subnet-077941617c18c49d8 X		
Subnet ID: subnet-05915ba82b2a4016b X				Subnet ID: subnet-060e55a78d5518d83 X		
Clear filters						
<input type="checkbox"/>	Private_Sub_App_1a	subnet-0080e1bfd672c0e4d	<input checked="" type="radio"/> Available	vpc-09e54287b0cadace VPC...	10.0.2.0/24	-
<input type="checkbox"/>	Private_Sub_App_1b	subnet-077941617c18c49d8	<input checked="" type="radio"/> Available	vpc-09e54287b0cadace VPC...	10.0.3.0/24	-
<input type="checkbox"/>	Private_Sub_DB_1a	subnet-05915ba82b2a4016b	<input checked="" type="radio"/> Available	vpc-09e54287b0cadace VPC...	10.0.4.0/24	-
<input type="checkbox"/>	Private_Sub_DB_1b	subnet-060e55a78d5518d83	<input checked="" type="radio"/> Available	vpc-09e54287b0cadace VPC...	10.0.5.0/24	-

All 4 private subnets have been successfully created.

STEP 8:

CREATION OF PRIVATE ROUTE TABLES THE APP TIER SUBNETS

(a) Create the private Route Table (**RouteT_Private_Sub_App**)

Create route table [Info](#)

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.

VPC
The VPC to use for this route table.

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 - <i>optional</i>	
<input style="width: 100%; height: 30px; border: 1px solid #ccc; margin-bottom: 5px;" type="text" value="Name"/>	<input style="width: 100%; height: 30px; border: 2px solid red; margin-bottom: 5px;" type="text" value="RouteT_Private_Sub_App"/>	Remove
Add new tag		
You can add 49 more tags.		

[Cancel](#) [Create route table](#)

(b) Edit the route table (**RouteT_Private_Sub_App**) to route traffic to the first NAT gateway (**NATGW1_My_Project**) that resides in the public subnet (**Public_Sub_Web_1a**).

For this configuration, simply click on the Route table (**RouteT_Private_Sub_App**), select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select NAT gateway (**NATGW1_My_Project**) as target, then click save changes.

The screenshot shows the AWS VPC Route Tables page. A specific route table, **rtb-0e76d5fdf2569b4bc / RouteT_Private_Sub_App**, is selected. In the 'Routes' section, there is one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

To edit this route, a modal dialog titled "Edit routes" is open. Inside, there are two rows of route entries:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	nat-061e6f27aa751aaaf	-	No

The second row has a "Remove" button next to it. At the bottom of the dialog are "Cancel", "Preview", and "Save changes" buttons.

(c) Edit subnet association

To edit the subnet association of the route table (**RouteT_Private_Sub_App**) associated to the 2 private subnets (**Private_Sub_App_1a**) and (**Private_Sub_DB_1a**), simply click on "Subnet associations", proceed to click on **Edit subnet associations**, **select** the private subnets (**Private_Sub_App_1a**) and (**Private_Sub_DB_1a**) then click on **save associations**.

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/6)					
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	Private_Sub_App_1a	subnet-0080e1bfd672c0e4d	10.0.2.0/24	-	Main (rtb-0c3e6603061628bc6)
<input type="checkbox"/>	Private_Sub_App_1b	subnet-077941617c18c49d8	10.0.3.0/24	-	Main (rtb-0c3e6603061628bc6)
<input checked="" type="checkbox"/>	Private_Sub_DB_1a	subnet-05915ba82b2a4016b	10.0.4.0/24	-	Main (rtb-0c3e6603061628bc6)
<input type="checkbox"/>	Private_Sub_DB_1b	subnet-060e55a78d518d83	10.0.5.0/24	-	Main (rtb-0c3e6603061628bc6)
<input type="checkbox"/>	Public_Sub_Web_1a	subnet-0552f97133c2ed30b	10.0.0.0/24	-	rtb-0404eb6be8b54a1 / RouteT_Publi...
<input type="checkbox"/>	Public_Sub_Web_1b	subnet-0a00c31de3ca2a263	10.0.1.0/24	-	rtb-0404eb6be8b54a1 / RouteT_Publi...

Selected subnets	
subnet-0080e1bfd672c0e4d / Private_Sub_App_1a	X
subnet-05915ba82b2a4016b / Private_Sub_DB_1a	X

Cancel
Save associations

STEP 9:

CREATION OF PRIVATE ROUTE TABLES FOR THE DATABASE TIER SUBNETS

(d) Create the private Route Table (**RouteT_Private_Sub_DB**)

Create route table Info

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.

VPC
The VPC to use for this route table.

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 style="width: 100%; border: 1px solid #ccc;" type="text" value="Name"/>	<input style="width: 100%; border: 2px solid red;" type="text" value="RouteT_Private_Sub_DB"/>	Remove
Add new tag		

You can add 49 more tags.

Cancel
Create route table

(e) Edit the route table (**RouteT_Private_Sub_DB**) to route traffic to the second NAT gateway (**NATGW2_My_Project**) that resides in the public subnet (**Public_Sub_Web_1b**). For this configuration, simply click on the Route table (**RouteT_Private_Sub_DB**), select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select NAT gateway (**NATGW2_My_project**) as target, then click **save changes**.

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

(f) Edit subnet association

To edit the subnet association of the route table (**RouteT_Private_Sub_DB**) linked to the 2 private subnets (**Private_Sub_App_1b**) and (**Private_Sub_DB_1b**), simply click on "Subnet

associations", proceed to click on "Edit subnet associations", select the private subnets **Private_Sub_App_1b** and **Private_Sub_DB_1b** then click on **save associations**.

The screenshot shows the 'Edit subnet associations' page in the AWS VPC Route Tables interface. At the top, it says 'Available subnets (2/6)'. Below is a table with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. Two subnets are selected: 'Private_Sub_App_1b' and 'Private_Sub_DB_1b'. In the 'Selected subnets' section, these two subnets are listed. At the bottom right are 'Cancel' and 'Save associations' buttons.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private_Sub_App_1a	subnet-0080e1bfd672c0e4d	10.0.2.0/24	-	rtb-0e76d5fd2569b4bc / RouteT_Private...
<input checked="" type="checkbox"/> Private_Sub_App_1b	subnet-077941617c18c49d8	10.0.3.0/24	-	Main (rtb-0c3e6603061628bc6)
Private_Sub_DB_1a	subnet-05915ba82b2a4016b	10.0.4.0/24	-	rtb-0e76d5fd2569b4bc / RouteT_Private...
<input checked="" type="checkbox"/> Private_Sub_DB_1b	subnet-060e55a78d5518d83	10.0.5.0/24	-	Main (rtb-0c3e6603061628bc6)
Public_Sub_Web_1a	subnet-0552f97133ced30b	10.0.0.0/24	-	rtb-0404eb6be8b54a1 / RouteT_Publi...
Public_Sub_Web_1b	subnet-0a0c31de3ca2a263	10.0.1.0/24	-	rtb-0404eb6be8b54a1 / RouteT_Publi...

Private Subnets (**Private_Sub_App_1b** and **Private_Sub_DB_1b**) successfully associated to the Route table (**RouteT_Private_Sub_DB**) as shown above.

STEP 10:

CREATION OF SECURITY GROUPS

Let's confirm that we are in the right region (**us-east-1**). Now let's navigate to the EC2 dashboard, scroll down to select Security Groups.

Click on create security group, give it a name, add description, and specify your VPC.

Choose the security group rule that will control our web server inbound and outbound traffic.

When traffic flows in from the internet, it must flow through the Application load balancer (ALB) before entering the web servers.

IMPORTANT: It is important to note that the Web Servers will only accept traffic that flows in from the internet through the Application Load Balancer and to the Web Server.

IMPORTANT: It is also very important to know that the Database servers will only accept traffic that flows in from the Web Servers.

IMPORTANT: Finally, it is important to know that the EFS file system servers will only accept traffic that flows in from the Web Servers as well.

Now, let's dive in to create our security groups.

(a) Lets create the security group for our Application Load balancer (**ALB_security_group**)

Create security group [Info](#)

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 [Info](#)
ALB_security_group
Name cannot be edited after creation.

Description [Info](#)
ALB Security Group

VPC [Info](#)
vpc-09e54287b0cadcae X

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
HTTP	TCP	80	Anywhere... <input type="button" value="▼"/>	<input type="text"/> 0.0.0.0/0 X <input type="button" value="Delete"/>
HTTPS	TCP	443	Anywhere... <input type="button" value="▼"/>	<input type="text"/> 0.0.0.0/0 X <input type="button" value="Delete"/>

[Add rule](#)

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Custom <input type="button" value="▼"/>	<input type="text"/> 0.0.0.0/0 X <input type="button" value="Delete"/>

[Add rule](#)

⌚ Security group (sg-0fadcc2ee9bd9bd01b | ALB_security_group) was created successfully

► [Details](#)

EC2 > Security Groups > sg-0fadcc2ee9bd9bd01b - ALB_security_group [Actions ▾](#)

sg-0fadcc2ee9bd9bd01b - ALB_security_group

Details

Security group name ALB_security_group	Security group ID sg-0fadcc2ee9bd9bd01b	Description ALB Security Group	VPC ID vpc-09e54287b0cadcae
Owner 763176333159	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

ALB_Security_group was successfully created.

(b) Lets create the security group for our web servers (**Web_server_SG**)

Create security group [Info](#)

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 [Info](#)
Web_server_SG
Name cannot be edited after creation.

Description [Info](#)
Web server security group

VPC [Info](#)
vpc-09e54287b0cadcae

⌚ Security group (sg-0d3f1230589c7b51f | Web_server_SG) was created successfully X

▶ Details

EC2 > Security Groups > sg-0d3f1230589c7b51f - Web_server_SG Actions ▾

Details

Security group name Web_server_SG	Security group ID sg-0d3f1230589c7b51f	Description Web server security group	VPC ID vpc-09e54287b0cadcae
Owner 763176333159	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Web_Server_SG was successfully created.

(c) Lets create security group for our database (**Database_SG**)

EC2 > Security Groups > Create security group

Create security group [Info](#)

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 [Info](#)
Database_SG
Name cannot be edited after creation.

Description [Info](#)
database security group

VPC [Info](#)
vpc-09e54287b0cadcae

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0ad6a6244db4a0686	MySQL/Aurora	TCP	3306	Custom	<input type="text" value="sg-0d3f1230589c7b51f"/> X

[Add rule](#)

⌚ Security group (sg-0e8857da8b9f6cdb4 | Database_SG) was created successfully
 ► Details

EC2 > Security Groups > sg-0e8857da8b9f6cdb4 - Database_SG

sg-0e8857da8b9f6cdb4 - Database_SG [Actions ▾](#)

Details			
Security group name Database_SG	Security group ID sg-0e8857da8b9f6cdb4	Description database security group	VPC ID vpc-09e54287b0cadace
Owner 763176333159	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Database_SG was successfully created.

(d) Lets create security group for our EFS File system (**EFS_File_SG**)

Create security group [Info](#)

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 Info	<input type="text" value="EFS_File_SG"/>
Name cannot be edited after creation.	
Description Info	<input type="text" value="EFS file system security group"/>
VPC Info	<input type="text" value="vpc-09e54287b0cadace"/> X

Inbound rules Info				
Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
NFS	TCP	2049	Custom	<input type="text" value="sg-0d3f1230589c7b51f"/> X

[Add rule](#)

⌚ Security group (sg-056dc5268d7fcc18b | EFS_File_SG) was created successfully

► Details

EC2 > Security Groups > sg-056dc5268d7fcc18b - EFS_File_SG

sg-056dc5268d7fcc18b - EFS_File_SG

Actions ▾

Details			
Security group name EFS_File_SG	Security group ID sg-056dc5268d7fcc18b	Description EFS file system security group	VPC ID vpc-09e54287b0cadace
Owner 763176333159	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

EFS_File_SG has been successfully created.

Let's edit the inbound rule of the NFS security group to add EFS security group (**EFS_File_SG**) as source for the NFS security group.

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-07f79d5ad21350b4e	NFS	TCP	2049	Custom	<input type="text"/> sg-0d3f1230589c7b51f X
-	NFS	TCP	2049	Custom	<input type="text"/> Q sg-0d3f1230589c7b51f X

Add rule

Cancel Preview changes Save rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-07f79d5ad21350b4e	NFS	TCP	2049	Custom	<input type="text"/> sg-0d3f1230589c7b51f X
-	NFS	TCP	2049	Custom	<input type="text"/> Q sg-056dc5268d7fcc18b X

Add rule

Cancel Preview changes Save rules

⌚ Inbound security group rules successfully modified on security group (sg-056dc5268d7fcc18b | EFS_File_SG)

▶ Details

EC2 > Security Groups > sg-056dc5268d7fcc18b - EFS_File_SG

sg-056dc5268d7fcc18b - EFS_File_SG

Actions ▾

Details			
Security group name EFS_File_SG	Security group ID sg-056dc5268d7fcc18b	Description EFS file system security group	VPC ID vpc-09e54287b0cadcace
Owner 763176333159	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Finally, the inbound rule for the NFS security group has been edited and added to EFS security group (**EFS_File_SG**) as traffic source for the NFS security group.

STEP 11:

CREATE THE DATABASE SUBNET GROUP THAT WOULD BE USED TO CREATE THE RDS DB.

First, we must confirm that we are in the right **region (N. Virginia us-east-1)** where our RDS databases will be launched. Let's carefully follow the steps below to create our RDS Databases.

RDS Engine version 5.7.42 is what we are going to use for this project.

IMPORTANT: First thing to do is to RDS > Subnet groups > Create DB subnet group

RDS > Subnet groups > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
 Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-east-1a X

us-east-1b X

Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-060e55a78d5518d83 (10.0.5.0/24) X

subnet-05915ba82b2a4016b (10.0.4.0/24) X

i For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-060e55a78d5518d83	10.0.5.0/24
us-east-1a	subnet-05915ba82b2a4016b	10.0.4.0/24

Cancel

Create

Subnet groups (3)

Subnet groups (3)				
C Edit Delete Create DB subnet group				
Filter by subnet group < 1 > @				
Name	Description	Status	VPC	
<input type="checkbox"/> default-vpc-08328352666abbbb8	Created from the RDS Management Console	✓ Complete	vpc-08328352666abbbb8	⋮
<input type="checkbox"/> default-vpc-09e54287b0cadcace	Created from the RDS Management Console	✓ Complete	vpc-09e54287b0cadcace	⋮
<input checked="" type="checkbox"/> my-project-rds-db	My project Subnet group	✓ Complete	vpc-09e54287b0cadcace	⋮

Successfully created Database subnet groups.

Now, let's navigate to **RDS**, click on create database, select "**Standard create**" and Choose **MySQL**.

RDS > Create database

Create database

Choose a database creation method Info

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 Info

Aurora (MySQL Compatible)


Aurora (PostgreSQL Compatible)


MySQL


MariaDB


PostgreSQL


Oracle


Templates

Choose a sample template to meet your use case.

Production

Use defaults for high availability and fast, consistent performance.

Dev/Test

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

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

[Info](#)

Availability and durability

We want to leave Engine Version as default MySQL 5.7.42

Engine Version

MySQL 5.7.42



Then entering the name (It is important to note that First character must be a letter. Can't have two consecutive hyphens (- -), it can't contain underscores (_), and it can't end with a hyphen.)

Next step will be to scroll down to settings, enter DB identifier and master password.

Settings

DB instance identifier [Info](#)

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.

My-Project-RDS-DB

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.

admin

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

Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

ⓘ If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
[Learn more](#)

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).

Scroll down to networking, select Burstable classes, and select db.t2.micro

Instance configuration

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



Amazon RDS Optimized Writes - new [Info](#)

Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t2.micro

1 vCPUs 1 GiB RAM Not EBS Optimized

Include previous generation classes

Connectivity [Info](#)



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.

Network type [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4

Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode

Your resources can communicate over IPv4, IPv6, or both.

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

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

VPC_My_Project (vpc-09e54287b0cadcace)

6 Subnets, 2 Availability Zones



Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

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.

my-project-rds-db

2 Subnets, 2 Availability Zones



Scroll down to Additional configuration > Database options.

All we need to do, is give a database name and maintain all other default values as they are.

IMPORTANT: Note that the "Initial database name" field accepts letters, numbers, underscores (_)

Navigate to the end of the page and then **create** a database.

▼ **Additional configuration**

Database options, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)
My_Project_RDS_DB

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)
default.mysql5.7

Option group [Info](#)
default:mysql-5-7

Backup

Enable automated backups
Creates a point-in-time snapshot of your database

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Backup retention period [Info](#)
The number of days (1-35) for which automatic backups are kept.
7 days

Backup window [Info](#)

Successfully created database my-project-rds-db
You can use settings from my-project-rds-db to simplify configuration of [suggested database add-ons](#) while we finish creating your DB for you.
How was your experience creating an Amazon RDS database? [Provide feedback](#)

[View connection details](#)

RDS > Databases > my-project-rds-db

my-project-rds-db

Modify Actions ▾

Summary			
DB identifier my-project-rds-db	CPU -	Status ⌚ Backing-up	Class db.t2.micro
Role Instance	Current activity <div style="width: 100%; height: 10px; background-color: #ccc; margin-bottom: 5px;"></div> 0 Connections	Engine MySQL Community	Region & AZ us-east-1a

Database successfully created.

STEP 12:

CREATE EFS FILE SYSTEM

We are going to create an Elastic File System (EFS) and create mount targets in each of the Database subnets (**Private_Sub_DB_1a**) and (**Private_Sub_DB_1b**) in each availability zone.

Then, we will put our application code in the created File System so that our Web Servers can pull our application code from the same location.

To create our EFS file system, navigate to:

services > storage > EFS > create file system > select VPC > customize.

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 recommended settings, including Elastic Throughput, Lifecycle Management, and Automatic Backups. These settings are designed to optimize the price-performance of your file system.

[Create file system](#)

Create file system

X

Create an EFS file system with recommended settings, including Elastic Throughput, Lifecycle Management, and Automatic Backups. These settings are designed to optimize the price-performance of your file system. [Learn more](#)

Name - optional

Name your file system.

Optional. Apply a name to your file system

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-09e54287b0cadcace
VPC_My_Project



Cancel

Customize

Create

File system settings

General

Name - optional

Name your file system.

My_Project_EFS

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

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](#)

Transition into IA

Transition files from Standard to Standard-Infrequent Access.

30 day(s) since last access

Transition out of IA

Transition files from Standard-Infrequent Access to Standard.

None

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

At this point, the file system name is optional, so we are not going to include any at this point.

Performance settings

Throughput mode

Choose a method for your file system's throughput limits. [Learn more](#)

Enhanced

Provides more flexibility and higher throughput levels for workloads with a range of performance requirements.

Bursting

Provides throughput that scales with the amount of storage for workloads with basic performance requirements.

Elastic (Recommended)

Use this mode for workloads with unpredictable I/O. With Elastic mode, your throughput scales automatically and you only pay for what you use.

Provisioned

Use this mode if you can estimate your workload's throughput requirements. With Provisioned mode, you configure your file system's throughput and pay for throughput provisioned.

► Additional settings

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

Network

Virtual Private Cloud (VPC) [Learn more](#)
Choose the VPC where you want EC2 instances to connect to your file system.
vpc-09e54287b0cadace
VPC_My_Project

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-1a	subnet-05915ba82b2a4...	Automatic	<input type="button" value="Choose security groups"/> Remove sg-056dc5268d7fcc18b X EFS_File_SG
us-east-1b	subnet-060e55a78d5518...	Automatic	<input type="button" value="Choose security groups"/> Remove sg-056dc5268d7fcc18b X EFS_File_SG

Cancel Previous Next

Review and create file system.

Success! File system (fs-01288959068f268f3) is available. [View file system](#)

Amazon EFS > File systems

File systems (1)

Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time	Availability Zone
My_Project_EFS	fs-01288959068f268f3	Unencrypted	6.00 KiB	6.00 KiB	0 Bytes	-	Available	Mon, 05 Jun 2023 22:29:44 GMT	Standard

File system successfully created.

The file system state is **available** as shown in the above screenshot.

Click on **My_Project_EFS**

Click on **Attach**

Amazon EFS > File systems > fs-01288959068f268f3

My_Project_EFS (fs-01288959068f268f3)

General

Performance mode: General Purpose

Throughput mode: Elastic

Lifecycle management:

- Transition into IA: 30 day(s) since last access
- Transition out of IA: None

Availability zone: Standard

Automatic backups: Enabled

Encrypted: No

File system state: Available

DNS name: fs-01288959068f268f3.efs.us-east-1.amazonaws.com

Delete Attach Edit

This will display the code that will be used to attach the file system to the web servers

Attach

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

Mount via DNS

Mount via IP

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-01288959068f268f3:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-01288959068f268f3.efs.us-east-1.amazonaws.com:/ efs
```

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

[Close](#)

Now, the next step will be to mount the NFS file system.

STEP 13: HOW TO USE AMAZON LINUX 2 TO INSTALL WORDPRESS AND MOVE FILES TO EFS

Let's navigate to our EC2 console to create a new SECURITY GROUP to enable us SSH into our EC2 web servers.

EC2 Dashboard > Security group > Create Security group.

Create security group Info

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 Info

Name cannot be edited after creation.

Description Info

VPC Info

 X

Inbound rules Info

Type Info

Protocol Info

Port range Info

Source Info

Description - optional Info

 ▼ X

47.185.111.40/32 X

Delete

Add rule

Outbound rules Info

Type Info

Protocol Info

Port range Info

Destination Info

Description - optional Info

 ▼ ▼ X

0.0.0.0/0 X

Delete

Add rule

Tags - optional

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Cancel

Create security group

⌚ Security group (sg-0bad8be8b4cf717b2 | SSH_Security_grp) was created successfully

► Details

EC2 > Security Groups > sg-0bad8be8b4cf717b2 - SSH_Security_grp

Actions ▾

sg-0bad8be8b4cf717b2 - SSH_Security_grp

Details

Security group name

Security group ID

Description

VPC ID

🔗

Owner

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

SSH Security Group was successfully created.

Now let's edit the inbound rules of our Web Servers (**Web_server_SG**) security group.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info		Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-03e9b442e5cc1951	SSH	TCP	22	Custom	<input type="text" value="Custom"/> <input type="button" value="Search"/> <input type="button" value="X"/>	<input type="text" value="sg-0bad8be8b4cf717b2"/> <input type="button" value="Delete"/>	
sgr-03adfcad974913028	HTTP	TCP	80	Custom	<input type="text" value="Custom"/> <input type="button" value="Search"/> <input type="button" value="X"/>	<input type="text" value="sg-0fadcc2ee9bd9bd01b"/> <input type="button" value="Delete"/>	
sgr-0664bbbe1137a53c9	HTTPS	TCP	443	Custom	<input type="text" value="Custom"/> <input type="button" value="Search"/> <input type="button" value="X"/>	<input type="text" value="sg-0fadcc2ee9bd9bd01b"/> <input type="button" value="Delete"/>	

[Add rule](#)

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

Now let's edit the inbound rules of our (**EFS_File_SG**) security group.

SSH was added to the inbound rules of (**Web_server_SG**) and EFS (**EFS_File_SG**) security groups respectively.

STEP 14: CREATION OF TWO EC2 WEB SERVERS

Now, let's create two (2) EC2 Linux Servers in our Public Web Tier Subnets:

(**Public_Sub_Web_1a**) and (**Public_Sub_Web_1b**).

Name [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

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

[Recents](#) [Quick Start](#)

Amazon Linux 	macOS 	Ubuntu 	Windows 	Red Hat 	S >
--	---	--	---	--	-----

Amazon Machine Image (AMI)

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Network Settings for Web Server (**My_Project_Server_01**) launched in Public Web Subnet.
(Public_Sub_Web_1a)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-09e54287b0cadcace (VPC_My_Project)
10.0.0.0/16

Subnet [Info](#)

subnet-0552f97133c2ed30b Public_Sub_Web_1a
VPC: vpc-09e54287b0cadcace Owner: 763176333159 Availability Zone: us-east-1a
IP addresses available: 249 CIDR: 10.0.0.0/24

Create new subnet [\[+\]](#)

Auto-assign public IP [Info](#)

Enable

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

SSH_Security_grp sg-0bad8be8b4cf717b2 X
VPC: vpc-09e54287b0cadcace

ALB_security_group sg-0fadcc2ee9bd9bd01b X
VPC: vpc-09e54287b0cadcace

Web_server_SG sg-0d3f1230589c7b51f X
VPC: vpc-09e54287b0cadcace

Compare security group rules

Hide all selected

Network Settings for Web Server (**My_Project_Server_02**) is launched in Public Web Subnet.
(Public_Sub_Web_1b)

Name

My_Project_Server_02

Add additional tags

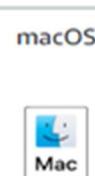
▼ Application and OS Images (Amazon Machine Image) Info

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

Recents

Quick Start



S
...



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

▼ Network settings Info

VPC - required Info

vpc-09e54287b0cadcace (VPC_My_Project)
10.0.0.0/16



Subnet Info

subnet-0a00c31de3ca2a263 Public_Sub_Web_1b
VPC: vpc-09e54287b0cadcace Owner: 763176333159 Availability Zone: us-east-1b
IP addresses available: 250 CIDR: 10.0.1.0/24



Create new subnet

Auto-assign public IP Info

Enable



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



SSH_Security_grp sg-0bad8be8b4cf717b2 X
VPC: vpc-09e54287b0cadcace



Compare security group rules

Web_server SG sg-0d3f1230589c7b51f X
VPC: vpc-09e54287b0cadcace

ALB_security_group sg-0fadcc2ee9bd9bd01b X
VPC: vpc-09e54287b0cadcace

Hide all selected

The 2 Web servers (**My_Project_Server_01**) and (**My_Project_Server_02**) have been successfully launched and they are running as you can see below.

Instances (2) <small>Info</small>				Connect	Instance state	Actions	Launch instances	
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
My_Project_Server_01	i-0702ea47e898c96c9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-167-61-145.co...	
My_Project_Server_02	i-0ca9c9c0cc1c2a8b7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-52-203-10-12.com...	

STEP 15:

CONNECTION TO OUR EC2 WEB SERVERS VIA SSH

I'm using a Windows 11 machine for this Project, so I will be connecting to my EC2 Web Servers using putty.

Established a connection to the Server using Putty.

Create the html directory and mount the efs to it

Ran the **sudo su** to switch user to root and then ran **yum update -y** to update the Web Server.

mkdir -p /var/www/html to create a new directory.

```
root@ip-10-0-0-77:/home/ec2-user
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
.
.
.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Wed Jun  7 07:22:55 2023 from 47.185.111.40
[ec2-user@ip-10-0-0-77 ~]$ 
[ec2-user@ip-10-0-0-77 ~]$ sudo su
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# yum update -y
Last metadata expiration check: 10:48:42 ago on Wed Jun  7 06:31:
30 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# mkdir -p /var/www/html
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsiz...ard,timeo=600,retrans=2,noresvport fs-01288959068f268f3.efs.us-east-1.amazonaws.com:/ /var/www/html
[root@ip-10-0-0-77 ec2-user]# 
```

```
sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-  
01288959068f268f3.efs.us-east-1.amazonaws.com:/ /var/www/html
```

Installed Apache

```
sudo yum install httpd -y httpd-tools mod_ssl.
```

```
sudo systemctl enable httpd to enable the httpd service.
```

```
sudo systemctl start httpd to start the service.
```

```
sudo systemctl status httpd to check the status of the httpd daemon if it is running or not.
```

```
[root@ip-10-0-0-77 ec2-user]#  
[root@ip-10-0-0-77 ec2-user]# yum install httpd -y httpd-tools mod_ss  
Last metadata expiration check: 11:05:21 ago on Wed Jun 7 06:31:30 2023.  
Package httpd-2.4.56-1.amzn2023.x86_64 is already installed.  
Package httpd-tools-2.4.56-1.amzn2023.x86_64 is already installed.  
No match for argument: mod_ss  
Error: Unable to find a match: mod_ss  
[root@ip-10-0-0-77 ec2-user]#  
[root@ip-10-0-0-77 ec2-user]# systemctl enable http  
Failed to enable unit: Unit file http.service does not exist.  
[root@ip-10-0-0-77 ec2-user]#  
[root@ip-10-0-0-77 ec2-user]# systemctl enable httpd  
[root@ip-10-0-0-77 ec2-user]# systemctl start httpd  
[root@ip-10-0-0-77 ec2-user]#  
[root@ip-10-0-0-77 ec2-user]# systemctl status httpd  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)  
   Active: active (running) since Wed 2023-06-07 07:56:58 UTC; 9h ago  
     Docs: man:httpd.service(8)  
 Main PID: 27811 (httpd)  
    Status: "Total requests: 269; Idle/Busy workers 100/0;Requests/sec: 0.0077; Bytes served:  
      Tasks: 230 (limit: 1108)  
     Memory: 24.1M  
       CPU: 24.555s  
      CGroup: /system.slice/httpd.service  
              ├─27811 /usr/sbin/httpd -DFOREGROUND  
              ├─27823 /usr/sbin/httpd -DFOREGROUND  
              ├─27825 /usr/sbin/httpd -DFOREGROUND  
              ├─27826 /usr/sbin/httpd -DFOREGROUND  
              ├─27827 /usr/sbin/httpd -DFOREGROUND  
              └─30502 /usr/sbin/httpd -DFOREGROUND
```

STEP 16: INSTALLATION OF WORDPRESS

Now to need to run the following commands step by step for the installation of WordPress.

```
# Install php 7.4  
sudo Amazon-Linux-extras enable php7.4  
sudo yum clean metadata  
sudo yum install php php-common php-pear -y  
sudo yum install php-{cgi,curl,mbstring,gd,mysqlnd,gettextjson,xml,fpm,intl,zip} -y
```

```
Installed:  
libxslt-1.1.34-5.amzn2023.0.2.x86_64  
nginx-filesystem-1:1.22.1-1.amzn2023.0.3.noarch  
php-pear-1:1.10.13-2.amzn2023.0.4.noarch  
php8.1-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-cli-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-common-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-fpm-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-mbstring-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-opcache-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-pdo-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-process-8.1.16-1.amzn2023.0.1.x86_64  
php8.1-xml-8.1.16-1.amzn2023.0.1.x86_64  
  
Complete!  
Last metadata expiration check: 0:00:05 ago on Wed Jun  7 17:56:54 2023.  
Package php8.1-cli-8.1.16-1.amzn2023.0.1.x86_64 is already installed.  
Package php8.1-common-8.1.16-1.amzn2023.0.1.x86_64 is already installed.  
Package php8.1-mbstring-8.1.16-1.amzn2023.0.1.x86_64 is already installed.  
No match for argument: php-gettextjson  
Package php8.1-xml-8.1.16-1.amzn2023.0.1.x86_64 is already installed.  
Package php8.1-fpm-8.1.16-1.amzn2023.0.1.x86_64 is already installed.  
No match for argument: php-zip  
Error: Unable to find a match: php-gettextjson php-zip  
[root@ip-10-0-0-77 ec2-user]#
```

```
# Install mysql 5.7  
sudo rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm  
sudo yum install mysql-community-server -y  
sudo systemctl enable mysqld
```

```
sudo systemctl start mysqld
```

```
sudo systemctl status mysqld
```

```
root@ip-10-0-0-77:/home/ec2-user
mysql-community-common x86_64 5.7.42-1.el7           mysql57-community 311 k
mysql-community-libs   x86_64 5.7.42-1.el7           mysql57-community 2.6 M
ncurses-compat-libs   x86_64 6.2-4.20200222.amzn2023.0.3
                                         amazonlinux      322 k

Transaction Summary
=====
Install 6 Packages

Total download size: 210 M
Installed size: 896 M
Downloading Packages:
(1/6) : libxcrypt-compat-4.4.33-7.amzn2023.x86_64 1.2 MB/s | 92 kB    00:00
(2/6) : ncurses-compat-libs-6.2-4.20200222.amzn2023.0.3.x86_64 2.9 MB/s | 322 kB   00:00
(3/6) : mysql-community-common-5.7.42-1.el7.x86_64 4.9 MB/s | 311 kB   00:00
(4/6) : mysql-community-libs-5.7.42-1.el7.x86_64 9.1 MB/s | 2.6 MB   00:00
(5/6) : mysql-community-client-5.7.42-1.el7.x86_64 45 MB/s | 28 MB   00:00
(6/6) : mysql-community-server-5.7.42-1.el7.x86_64 52 MB/s | 179 MB  00:03

Total                                         58 MB/s | 210 MB  00:03

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : mysql-community-common-5.7.42-1.el7.x86_64 1/6
Installing : mysql-community-libs-5.7.42-1.el7.x86_64 2/6
Running scriptlet: mysql-community-libs-5.7.42-1.el7.x86_64 2/6
Installing : ncurses-compat-libs-6.2-4.20200222.amzn2023.0.3.x86_64 3/6
Installing : mysql-community-client-5.7.42-1.el7.x86_64 4/6
Installing : libxcrypt-compat-4.4.33-7.amzn2023.x86_64 5/6
Running scriptlet: mysql-community-server-5.7.42-1.el7.x86_64 6/6
Installing : mysql-community-server-5.7.42-1.el7.x86_64 ] 6/6
```

```
# Set the right permissions
```

```
sudo usermod -a -G apache ec2-user
```

```
sudo chown -R ec2-user:apache /var/www
```

```
sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
```

```
sudo find /var/www -type f -exec sudo chmod0664 {} \;
chown apache:apache -R /var/www/html
# Download of wordpress files
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
```

Create the wp-config.php file

```
cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
```

Edit the wp-config.php file

```
vi /var/www/html/wp-config.php
```

Navigate to your database, click on configuration to get information to edit the file

We need to configure 4 parameters:

DB_NAME

DB_USER

DB_PASSWORD

DB_HOST

Now after the configuration lets save by :wq! + Enter

Restart the Webserver

```
sudo systemctl restart httpd or service httpd restart
```

Navigate to the EC2 console

Select the EC2, select (My_Project_Server_01)

Copy the public IPV4 address

Paste the copied IP address on a new tab and boooooom

WordPress User Interface is ready!

In case of any extension error message, run the following command to install extension:

sudo yum install php-mysqli or sudo yum install php-pdo-mysql

```
root@ip-10-0-0-77:/home/ec2-user
Verifying : mysql-community-common-5.7.42-1.el7.x86_64          4/6
Verifying : mysql-community-libs-5.7.42-1.el7.x86_64           5/6
Verifying : mysql-community-server-5.7.42-1.el7.x86_64         6/6

Installed:
libxcrypt-compat-4.4.33-7.amzn2023.x86_64
mysql-community-client-5.7.42-1.el7.x86_64
mysql-community-common-5.7.42-1.el7.x86_64
mysql-community-libs-5.7.42-1.el7.x86_64
mysql-community-server-5.7.42-1.el7.x86_64
ncurses-compat-libs-6.2-4.20200222.amzn2023.0.3.x86_64

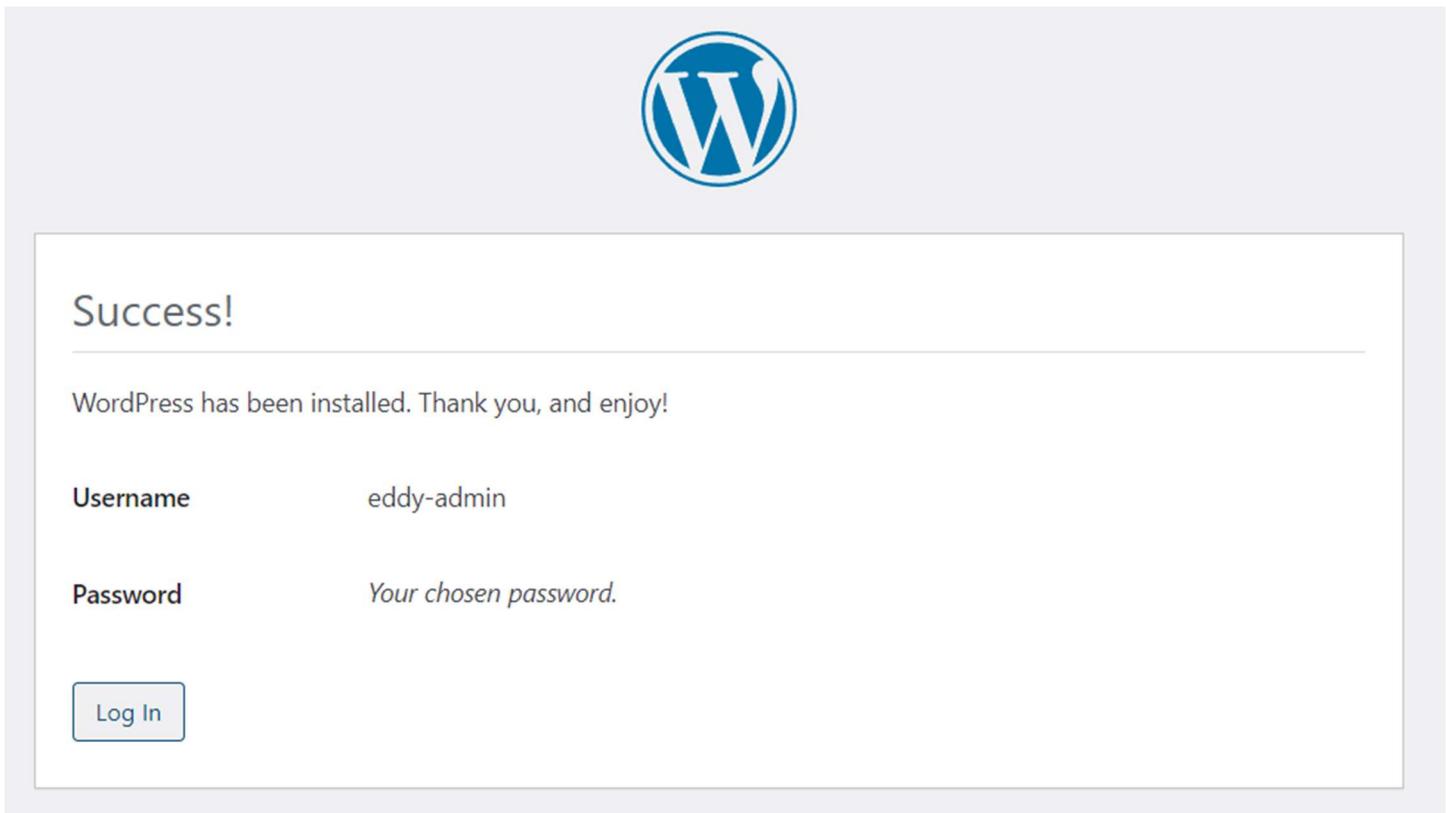
Complete!
[root@ip-10-0-0-77 ec2-user]# sudo systemctl enable mysqld
sudo systemctl start mysqld
sudo systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: >
   Active: active (running) since Wed 2023-06-07 19:21:14 UTC; 77ms ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 62583 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, stat>
   Process: 62635 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/>
 Main PID: 62637 (mysqld)
    Tasks: 27 (limit: 1108)
   Memory: 324.6M
      CPU: 2.413s
     CGroup: /system.slice/mysqld.service
             └─62637 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/m>

Jun 07 19:21:07 ip-10-0-0-77.ec2.internal systemd[1]: Starting mysqld.service >
Jun 07 19:21:09 ip-10-0-0-77.ec2.internal mysqld_pre_systemd[62607]: mysqld: O>
Jun 07 19:21:14 ip-10-0-0-77.ec2.internal systemd[1]: Started mysqld.service ->
lines 1-17/17 (END)
```

Ln 351, Col 29

In case of any extension error message, run the following command to install extension:

sudo yum install php-mysqli or sudo yum install php-pdo-mysql



A screenshot of the WordPress dashboard. The top navigation bar includes 'Dashboard', 'Screen Options ▾', and 'Help ▾'. A prominent blue banner at the top says 'Welcome to WordPress!' and provides a link to 'Learn more about the 6.2.2 version.' Below the banner, there are three main sections: 1) 'Author rich content with blocks and patterns' featuring a pen icon, a brief description, and a 'Add a new page' button. 2) 'Customize your entire site with block themes' featuring a theme icon, a brief description, and a 'Open site editor' button. 3) 'Switch up your site's look & feel with Styles' featuring a style icon, a brief description, and a note about tweaking colors and fonts. An 'X Dismiss' button is located in the top right corner of the banner area.

We have successfully installed WordPress application on Amazon Linux 2 AMI, and we have moved the file to EFS.

STEP 17:

CREATION OF 2 EC2 INSTANCES FOR TARGET GROUPS

Launch EC2 instances

IMPORTANT: Here is the user data script for the installation of the EC2 instances

```
#!/bin/bash

yum update -y

sudo yum install httpd -y httpd-tools mod_ssl

sudo systemctl enable httpd

sudo systemctl start httpd

sudo systemctl status httpd

sudo amazon-linux-extras enable php7.4

sudo yum clean metadata

sudo yum install php php-common php-pear -y

sudo yum install php-{cgi,curl,mbstring,gd,mysqlnd,gettextjson,xml,fpm,intl,zip} -y

sudo rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

sudo yum install mysql-community-server -y

sudo systemctl enable mysqld

sudo systemctl start mysqld

echo "fs-01288959068f268f3.efs.us-east-1.amazonaws.com:/var/www/html nfs4
nfsvers=4.1,rsize=1048576 mount -a

service httpd restart
```

My two EC2 instances (EC2-For-ALB-App-Sub-1a) and (EC2-For-ALB-App-Sub-1b) were successfully created and running.

STEP 18:

CREATION OF ALB TARGET GROUPS

Go to the EC2 service.

In the navigation pane, click on "Target Groups" under "Load Balancing."

Click on the "Create target group" and specify a name for the target group,

Configure the health checks for your target group. Health checks help the load balancer determine the availability of targets. Set the protocol, path, and interval for the health checks.

Specify the targets for your target group. This depends on the target type you selected earlier. If you choose "Instance," you can select one or more instances to include in the target group.

Once the target group is created, you can associate it with your Application Load Balancer to route traffic to the targets based on the specified rules.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol**Port**

:

1-65535

VPC

Select the VPC with the instances that you want to include in the target group.

**Protocol version** **HTTP1**

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

 HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

 gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

 Traffic port **Override****Healthy threshold**

The number of consecutive health checks successes required before considering an unhealthy target healthy.

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

seconds

2-120

Interval

The approximate amount of time between health checks of an individual target

seconds

5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

TargetGrp-My-Project

Actions ▾

Details

arn:aws:elasticloadbalancing:us-east-1:763176333159:targetgroup/TargetGrp-My-Project/460faed089629ef

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-09e54287b0cadace		
IP address type IPv4	Load balancer None associated				
Total targets 2	Healthy 0	Unhealthy 0	Unused 2	Initial 0	Draining 0

► Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Target group successfully created and registered with the ALB.

STEP 19:

CREATION OF APPLICATION LOAD BALANCER (ALB)

Now, let's proceed to create our Application Load Balancer

IMPORTANT: Very important that the Application Load balancer (ALB) should be associated with Public Subnets (**Public_Sub_Web-1a**) and (**Public_Sub_Web_1b**).

The reason for this is simple, this is because Internet traffic flows into both subnets through the internet gateway (**IGW**), the ALB being associated with the public subnets, with then be contaminated with the traffic. The ALB will then wire this traffic from both Public Subnet straight to the Private Application Subnets and to the Private Database Subnets as well.

Create Application Load Balancer

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Elastic Load Balancing works

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

ALB-My-Project

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme

Scheme can't be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)

For internet-facing load balancers, the IPv4 addresses of the nodes are assigned by AWS. For internal load balancers, the IPv4 addresses are assigned from the subnet CIDR.

VPC

vpc-09e54287b0cadace [Edit](#)

IP address type

IPv4

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az4)

Subnet

subnet-0552f97133c2ed30b

Public_Sub_Web_1a ▾

IPv4 address

Assigned by AWS

us-east-1b (use1-az6)

Subnet

subnet-0a00c31de3ca2a263

Public_Sub_Web_1b ▾

IPv4 address

Assigned by AWS

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups



ALB_security_group sg-0fadcc2ee9bd9bd01b X
VPC: vpc-09e54287b0cadace

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol

Port

Default action [Info](#)

HTTP

: 80

1-65535

Forward to

TargetGrp-My-Project
Target type: Instance, IPv4

HTTP

Create target group [Edit](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Load balancers (1)						
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.						
<input type="text"/> Find resources by attribute or tag						
<input checked="" type="checkbox"/>	<input type="text"/> ALB-My-Project X <input type="button" value="Clear filters"/>					
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	
<input type="checkbox"/>	ALB-My-Project	ALB-My-Project-19543104...	Provisioning	vpc-09e54287b0cadcae	2 Availability Zones	

Load balancer successfully created.

PROJECT SUMMARY

Deployed a WordPress Website on AWS using some of the AWS core services such as VPC, Public and Private Subnets, EC2, Security Groups, Route Tables, Internet Gateway, NAT Gateway, MySQL, Elastic File System (EFS), Target groups, Application Load Balancer, Auto Scaling Group and Route 53.

Installed and configured WordPress, Apache Web Server, PHP, and MySQL on my EC2 instance and provided some necessary details, such as database name, username, and password.

Set up a domain name for my WordPress website with Route 53 and mapped my domain DNS record to the IP address of my EC2 instance.

Tested the WordPress website by accessing the domain in a web browser and was able to see the WordPress installation default page and WordPress ran perfectly.

Launch an EC2 Instance: In the AWS Management Console, navigate to EC2 (Elastic Compute Cloud) and launch a new EC2 instance. Select an appropriate Amazon Machine Image (AMI) based on your requirements. It's common to choose an AMI with a pre-installed LAMP (Linux, Apache, MySQL, PHP) stack.

For this project, we would be using our desired Amazon Linux 2 (AMI) type t2.micro, and configuring the necessary settings, as well as installing Apache, MySQL and PHP based on the WordPress requirements.