

## AWS CLOUD SOLUTION FOR 2 COMPANY WEBSITES USING A REVERSE PROXY TECHNOLOGY

It is important to note that this infrastructure set up would not be covered in the AWS free tier as we would be using resources that would come at a cost .it is therefore advisable to set up a budget cost and configure notification to a pre-defined limit. Once done with the infrastructures we should ensure to delete all resources once the project is completed .

Please also note these overall principles that are used by the project are AWS concepts but they are common across most of the major Cloud Providers (e.g., [Microsoft Azure](#) and [Google Cloud Platform](#)).

This project would be implemented manually as it is important to understand the manual concepts before we begin to automate and this is the best way for to start because automation might become very frustrating if you don't understand these AWS specific concepts manually.

We will build a secure infrastructure inside AWS VPC (Virtual Private Cloud) network for a fictitious company named “PERFECT” that uses **WordPress CMS** for its main business website, and a **Tooling Website**

<https://github.com/eyewande2022/perfect-project-config.git>

This would be used for their DevOps team. As part of the PERFECT company's desire for improved security and performance, a decision has been made to use a reverse proxy technology from **NGINX** to achieve this.

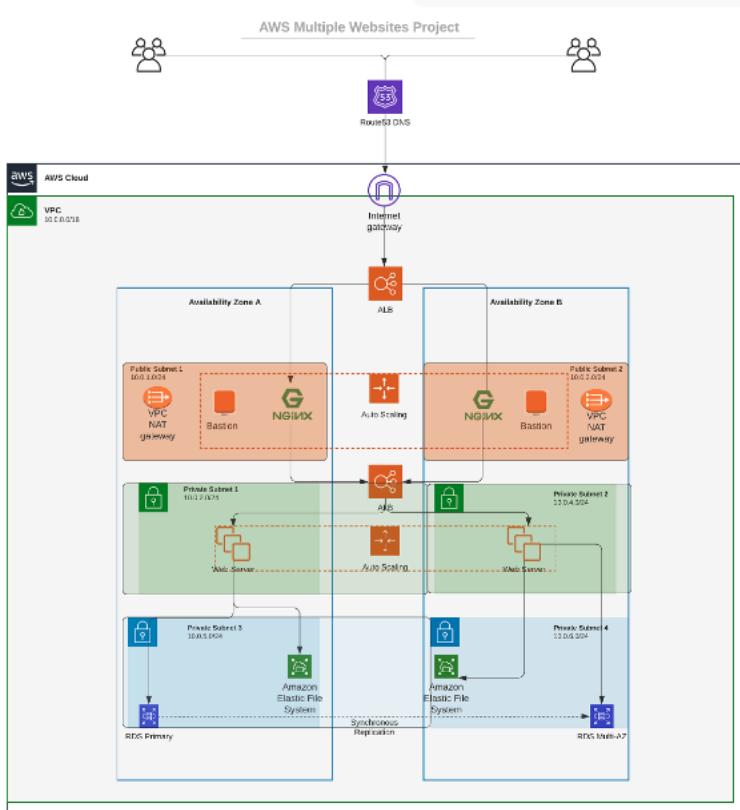
Pre-requisites for this project are:

Extensive Knowledge of AWS infrastructure resources namely is required for this project:

Route53 ,Application Load Balancer Nginx Reverse Proxy

VPC, Private and Public Subnets, Bastion Server ,DNS Elastic File System, RDS, Key Management Systems and many more:

Cost, Security, and Scalability are the major requirements for this project. Hence, implementing the architecture designed below, ensure that infrastructure for both websites, WordPress and Tooling, is resilient to Web Server's failures, can accommodate to increased traffic and, at the same time, has reasonable cost.



## AWS MULTIPLE WEBSITE PROJECT

### EXPLAINING EACH COMPONENT IN THIS INFRASTRUCTURE

- 1) Route 53 DNS server : provided by AWS helps us to resolve our ip address to a hostname. e.g mapping out our IP address to pftd.com we are using in this project .The DNS server helps us achieves that
- 2) Internet Gateway is what connected the internet or external network to our VPC (Virtual Private Cloud).Entry point to our VPC.
- 3) ALB: A single point of contact that helps us distribute traffic to multiple target server and this is where high

availability concept is used .we are using an internet facing ALB and an internal facing ALB for this project

- 4)VPC: Virtual Private Cloud Is an isolated environment in this cloud that has its own network pool addresses and can share its IP address with its component e.g 10.0.0.0/16
- 5)NGINX Reverse PROXY Server: It's a public server that sits behind a firewall in a private network and directs clients request to the appropriate backend server (2<sup>nd</sup> ALB internal). It gives an additional level ion abstraction and control to ensure smooth flow of network traffics between clients and servers
- 6)Autoscaling group: Helps us in scaling up and down our instances:

Please note in this project we would be using the autoscaling group to create the following

- a) At public subnet level our Bastion and NGINX reverse proxy has been placed in an ASG.
- b) At the Private subnet level, we use it to create the servers for the tooling and WordPress

- 7)NAT Gateway: Helps our private subnet to communicate with the internet. Without this our internet cannot communicate with our private subnet .

- 8) Public Subnet allows incoming traffic into it through the internet gateway
- 9) Private subnet: does not allow incoming traffic from the public but can only accept through the NAT GATEWAY.
- 10) Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth. It has a dynamic elasticity and instantly scales your file system storage capacity up or down as you add or remove files without disrupting your applications, giving you the storage, you need when you need it. It is fully managed by AWS
- 11) RDS (Relational Database Service) : This is a managed database system by AWS. It is easy set up ,operate and scale up RDS databases.
- 12) Bastion Server : It Is the entry point of all of our resources.

## CONNECTIVITY OF ALL INFRASTRUCTURES EXPLAINED:

Steps:

- a) Our clients on the internet hits that endpoint Wordpress.pftd.com.
- b) It gets to the route53 which then resolves that name to the “DNS name of our Application Load Balancer. This internet traffic comes in via the internet gateway to our ALB which is facing the internet and receiving traffic from anywhere.
- c) Once External ALB receives the internet traffic it forwards it to the NGINX reverse proxy which have its own special configuration with instructions to forward all traffics received from the External ALB to the INTERNAL ALB
- d) Internal ALB is expected to send this traffic to the webservers which in this case is WordPress server and the Tooling server. But how does it send it to the right one without making any mistake?

A rule has to be configured in our nginx reverse proxy server which would require the host header of

Default rule: Forward the host header of the webserver to the WordPress server via the Internal ALB

**Conditional Rule: Check host header if “tooling” Route specific traffic to Internal ALB which in turn sends to the WordPress tooling server.**

- e) Our webservers would mount all their file system to the Elastic File System
- f) The data would be stored in the RDS database
- g) Bastion Server which is the entry point of all our resources but to effectively perform its role, the nginx concept needs to be considered .Nginx reverse proxy is the one that routes traffic from the External ALB downwards because of its access to the internet (public subnet ). The Bastion needs to connect to the Nginx reverse proxy primarily to carry out administrative and management tasks which is the only way to connect to the private subnet, WordPress server as well as the WordPress tooling website .

Its time to start creating all resources to kickstart the project. Please note that we are using 2 different availability zone to ensure high availability .

**1)CREATE THE VPC:**

**Billing**  
Careers by sector | Capita  
www.capita.com/careers/careers-by-sector

**S3**  
Scalable Storage in the Cloud

**VPC**  
Isolated Cloud Resources

**Launch Wizard**  
Guided deployment for Enterprise applications and complex workloads

**Your VPCs (1) [Info](#)**

[Actions ▾](#) [Create VPC](#)

[Search](#) < 1 > [⚙️](#)

### 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.  
**PERFECT-vpc**

**IPv4 CIDR block [Info](#)**  
 **IPv4 CIDR manual input**  IPAM-allocated IPv4 CIDR block

**IPv4 CIDR**  
**10.0.0.0/16**

CIDR block size must be between /16 and /28.

**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](#)**  
**Default**

**Key** **Value - optional**

[Remove tag](#)

[Add tag](#)  
You can add 49 more tags

[Cancel](#) **Create VPC**

**YouTube** successfully created [vpc-02f4a7b74bf2b7029 / PERFECT-vpc](#)

[VPC](#) > [Your VPCs](#) > [vpc-02f4a7b74bf2b7029](#)

**vpc-02f4a7b74bf2b7029 / PERFECT-vpc**

[Actions ▾](#)

VPC has been created .

## 2) Edit and enable DNS Hostname

The screenshot shows the 'Edit VPC' configuration page for a VPC named 'PERFECT-vpc'. The 'DNS settings' section is highlighted with a green box, specifically the checkbox for 'Enable DNS hostnames'. A green arrow points from this section to the 'Save' button at the bottom right of the modal. The 'Save' button is also highlighted with a green box. Below the modal, the main VPC details page shows the 'DNS hostnames' status as 'Enabled'.

**DHCP settings**

DHCP option set [Info](#)  
dopt-0c65dd1678901b887 ▾

**DNS settings**

Enable DNS resolution [Info](#)

Enable DNS hostnames [Info](#)

**Network Address Usage metrics settings**

Enable Network Address Usage metrics [Info](#)

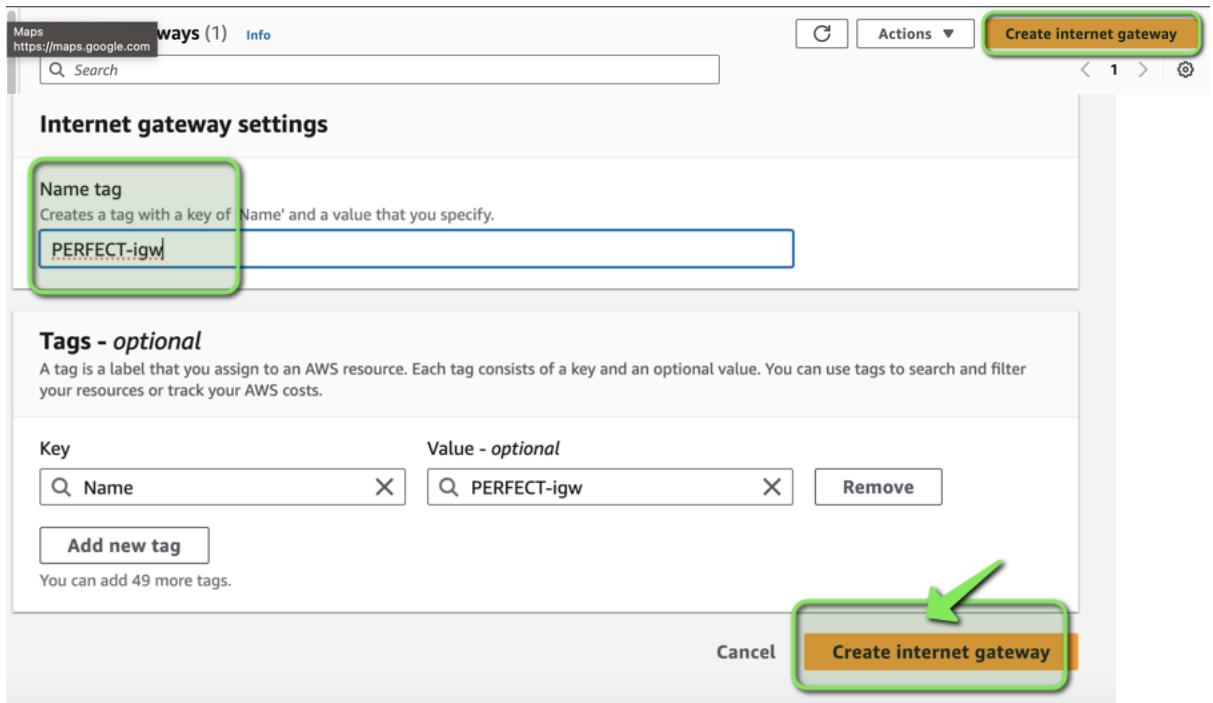
Cancel Save

You have successfully modified the settings for vpc-02f4a7b74bf2b7029 / PERFECT-vpc.

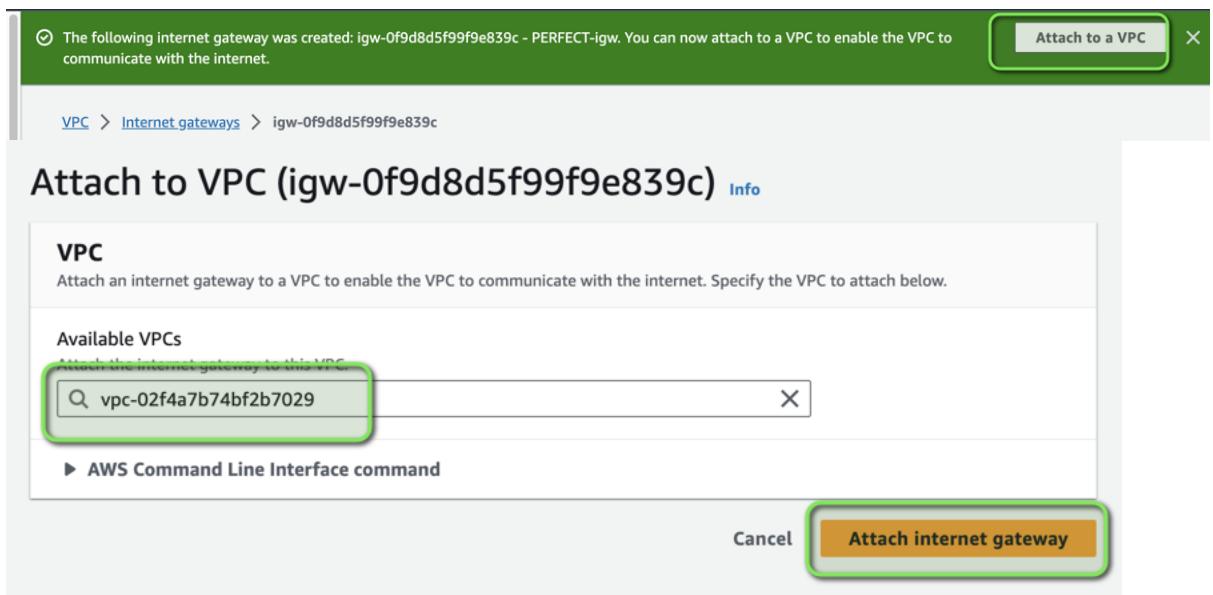
Actions

Details <a href="#">Info</a>			
VPC ID vpc-02f4a7b74bf2b7029	State Available	DNS hostnames Enabled	DNS resolution Enabled
<input checked="" type="checkbox"/> PERFECT-vpc	vpc-02f4a7b74bf2b7029	<input checked="" type="checkbox"/> Available	10.0.0.0/16

## 3) Create an Internet Gateway



#### 4) Attach Internet Gateway to the VPC



Attached successfully.

#### 5) Create Private and Public Subnets.

Please Note: All public subnets are even numbers while private subnets are odd numbers

The screenshot shows the AWS Subnets page with three main sections:

- Subnets (2) Info:** Shows two subnets: "perfect-public-subnet-2" and "perfect-public-subnet-1". Both are listed as "Available" in the VPC column.
- Subnets (4) Info:** Shows four subnets: "perfect-private-subnet-1", "perfect-private-subnet-2", "perfect-private-subnet-3", and "perfect-private-subnet-4". All are listed as "Available" in the VPC column.
- Instances (1/2) Info:** Shows one instance: "london-server" (ID i-09680294e021a9960). It is currently "Running".

A green banner at the top of the first section indicates: "You have successfully created 2 subnets: subnet-0ba79ccabddc2d174, subnet-069b4d1c58d3da588". A green banner at the top of the second section indicates: "You have successfully created 4 subnets: subnet-014a296e21ec4f399, subnet-01db88160dba0a1ee, subnet-0b4318fd34f0f7016, subnet-055865b7d7c9a174d".

All subnets are created .

Subnets (6) <a href="#">Info</a>			
<input type="text"/> Find resources by attribute or tag			
<input type="checkbox"/>	Name	Subnet ID	State
<input type="checkbox"/>	perfect-private-subnet-1	subnet-014a296e21ec4f399	<input checked="" type="checkbox"/> Available
<input type="checkbox"/>	perfect-private-subnet-2	subnet-01db88160dba0a1ee	<input checked="" type="checkbox"/> Available
<input type="checkbox"/>	perfect-private-subnet-3	subnet-0b4318fd34f0f7016	<input checked="" type="checkbox"/> Available
<input type="checkbox"/>	perfect-public-subnet-2	subnet-069b4d1c58d3da588	<input checked="" type="checkbox"/> Available
<input type="checkbox"/>	perfect-public-subnet-1	subnet-0ba79ccabddc2d174	<input checked="" type="checkbox"/> Available
<input type="checkbox"/>	perfect-private-subnet-4	subnet-055865b7d7c9a174d	<input checked="" type="checkbox"/> Available

## 6) Create route table for the public subnet :

Route tables (2) <a href="#">Info</a>			
<input type="text"/> Find resources by attribute or tag			
<input type="checkbox"/>	Name	Route table ID	Explicit subnet associati...   Edge associations   Main   VPC

**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%; height: 30px; border: 1px solid #ccc; margin-bottom: 5px;" type="text" value="Name"/>	<input style="width: 100%; height: 30px; border: 1px solid #ccc; margin-bottom: 5px;" type="text" value="perfect-public-rtb"/>
<input style="border: 1px solid #ccc; padding: 2px 10px; margin-bottom: 5px;" type="button" value="Add new tag"/>	
You can add 49 more tags.	

## 7) Create route table for the private subnet :

**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 type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="perfect-private-rtb"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

You can add 49 more tags.

## Route tables created

**Route tables (3)** Info

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associations	Edge associations
<input type="checkbox"/>	-	rtb-09bf33c7152a53f99	-	-
<input type="checkbox"/>	perfect-public-rtb	rtb-04b5962597ee2abea	-	-
<input type="checkbox"/>	perfect-private-rtb	rtb-034f5bcc7cf15c6c0	-	-

## 8) Perform subnet association to your created route tables.

Route Tables:

Name	ID	Description	Associations
<input checked="" type="checkbox"/> perfect-public-rtb	rtb-04b5962597ee2abea	-	No vpc-02f4a:
<input type="checkbox"/> perfect-private-rtb	rtb-034f5bcc7cf15c6c0	-	No vpc-02f4a:

**rtb-04b5962597ee2abea / perfect-public-rtb**

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**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 type="checkbox"/> perfect-private-subnet-1	subnet-014a296e21ec4f399	10.0.1.0/24	-	Main (rtb-09bf33c7152a53f99)
<input type="checkbox"/> perfect-private-subnet-2	subnet-01db88160dba0a1ee	10.0.3.0/24	-	Main (rtb-09bf33c7152a53f99)
<input type="checkbox"/> perfect-private-subnet-3	subnet-0b4318fd34f0f7016	10.0.5.0/24	-	Main (rtb-09bf33c7152a53f99)
<input checked="" type="checkbox"/> perfect-public-subnet-2	subnet-069b4d1c58d3da588	10.0.2.0/24	-	Main (rtb-09bf33c7152a53f99)
<input checked="" type="checkbox"/> perfect-public-subnet-1	subnet-0ba79ccabddc2d174	10.0.0.0/24	-	Main (rtb-09bf33c7152a53f99)
<input type="checkbox"/> perfect-private-subnet-4	subnet-055865b7d7c9a174d	10.0.7.0/24	-	Main (rtb-09bf33c7152a53f99)

**Selected subnets**

subnet-069b4d1c58d3da588 / perfect-public-subnet-2 | subnet-0ba79ccabddc2d174 / perfect-public-subnet-1

Cancel | **Save associations**

Public route table associated.

Route Tables:

Name	ID	Description	Associations
<input checked="" type="checkbox"/> perfect-private-rtb	rtb-034f5bcc7cf15c6c0	-	No vpc-02f4a:

**rtb-034f5bcc7cf15c6c0 / perfect-private-rtb**

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**Available subnets (4/6)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> perfect-private-subnet-1	subnet-014a296e21ec4f399	10.0.1.0/24	-	Main (rtb-09bf33c7152a53f99)
<input checked="" type="checkbox"/> perfect-private-subnet-2	subnet-01db88160dba0a1ee	10.0.3.0/24	-	Main (rtb-09bf33c7152a53f99)
<input checked="" type="checkbox"/> perfect-private-subnet-3	subnet-0b4318fd34f0f7016	10.0.5.0/24	-	Main (rtb-09bf33c7152a53f99)
<input type="checkbox"/> perfect-public-subnet-2	subnet-069b4d1c58d3da588	10.0.2.0/24	-	rtb-04b5962597ee2abea / perfect-public-rtb
<input type="checkbox"/> perfect-public-subnet-1	subnet-0ba79ccabddc2d174	10.0.0.0/24	-	rtb-04b5962597ee2abea / perfect-public-rtb
<input checked="" type="checkbox"/> perfect-private-subnet-4	subnet-055865b7d7c9a174d	10.0.7.0/24	-	Main (rtb-09bf33c7152a53f99)

**Selected subnets**

subnet-014a296e21ec4f399 / perfect-private-subnet-1 | subnet-01db88160dba0a1ee / perfect-private-subnet-2 | subnet-0b4318fd34f0f7016 / perfect-private-subnet-3 |  
subnet-055865b7d7c9a174d / perfect-private-subnet-4

Cancel | **Save associations**

Private route table associated

rtb-034f5bcc7cf15c6c0 / perfect-private-rtb

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**Explicit subnet associations (0)**

Find subnet association

Subnet	Route Table ID	Subnets
perfect-public-rtb	rtb-04b5962597ee2abea	2 subnets
perfect-private-rtb	rtb-034f5bcc7cf15c6c0	4 subnets

**Edit subnet associations**

All route table associated.

- 9) Editing route of the route table for public subnet to have access to the internet through the internet gateway. This would allow our public subnet be accessible to the internet .

Route tables (1/4) [Info](#)

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associati...	Edge...	Action
-	rtb-09bf33c7152a53f99	-	-	<a href="#">View details</a>
-	rtb-0165a714670aaabd2	-	-	<a href="#">Set main route table</a>
<b>perfect-public-rtb</b>	<b>rtb-04b5962597ee2abea</b>	<b>2 subnets</b>	-	<a href="#">Edit subnet associations</a>
perfect-private-rtb	rtb-034f5bcc7cf15c6c0	4 subnets	-	<a href="#">Edit edge associations</a>

[Create route](#)

[Edit route propagation](#)

[Edit routes](#)

[Manage tags](#)

**Edit routes**

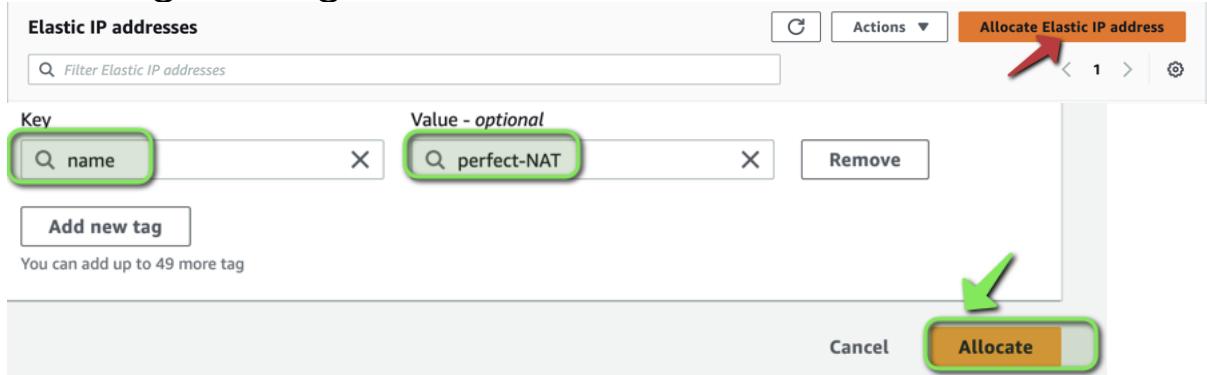
Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	No	<a href="#">Remove</a>

[Add route](#)

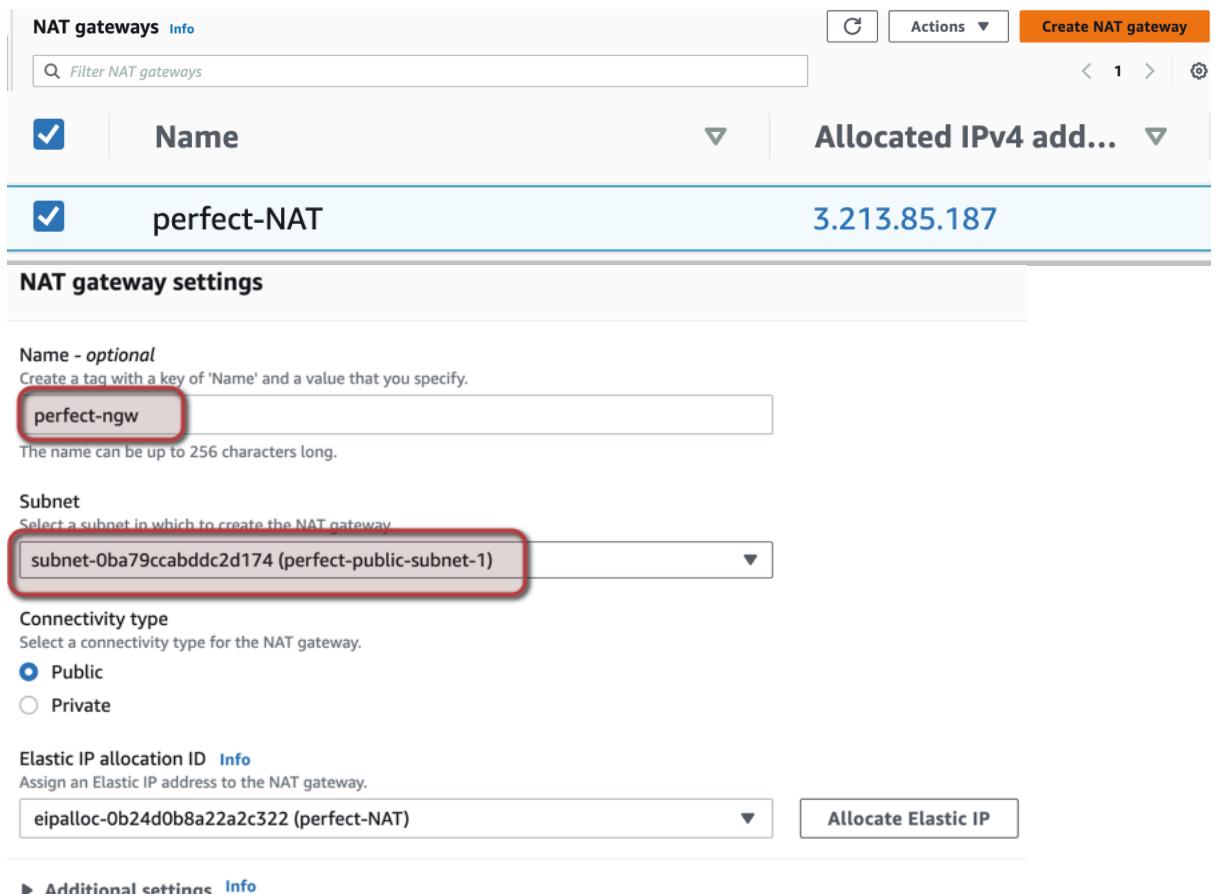
[Cancel](#) [Preview](#) **Save changes**

- 10) Create an Elastic IP address (stable IP ADDRESS ) that does not change .This would be needed by the NAT GATEWAY which is needed for private subnet access .

i) Elastic IP address allocated .Please note all Elastic IP must be allocated and you wont get charged but if you do not allocate it you would get charged .



11)Create NATGATEWAY. Please note they are always created on a public subnet 1 as shown below .



**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 gateway nat-0ba8f0b6d8c601cdb | perfect-ngw was created successfully.

VPC > NAT gateways > nat-0ba8f0b6d8c601cdb

NAT GATEWAY created successfully .

## 11) Navigate to the private route table and edit route.

Name	Route table ID	Explicit subnet associations	Edge associations
<input checked="" type="checkbox"/> perfect-private-rtb	rtb-034f5bcc7cf15c6c0	4 subnets	-
<input type="checkbox"/> -	rtb-09bf33c7152a53f99	-	-
<input type="checkbox"/> -	rtb-0165a714670aaabd2	-	-
<input type="checkbox"/> perfect-public-rtb	rtb-04b5962597ee2abea	2 subnets	-

Set main route table  
 Edit subnet associations  
 Edit edge associations  
 Edit route propagation  
 Edit routes  
 Manage tags  
 Delete route table

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

We have configured our route table and then we proceed to create security groups for other resources

.

**12) Create security group for external ALB. Open traffic for (HTTP AND HTTPS) for inbound rules .**

Security Groups (92) [Info](#) [Actions](#) [Export security groups to CSV](#) [Create security group](#)

Filter security groups

**Basic details**

Security group name [Info](#)  
perfect-ExternalALB-sg

Name cannot be edited after creation.

Description [Info](#)  
perfect-ExternalALB-sg

VPC [Info](#)  
vpc-02f4a7b74bf2b7029

**Inbound rules** [Info](#)

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

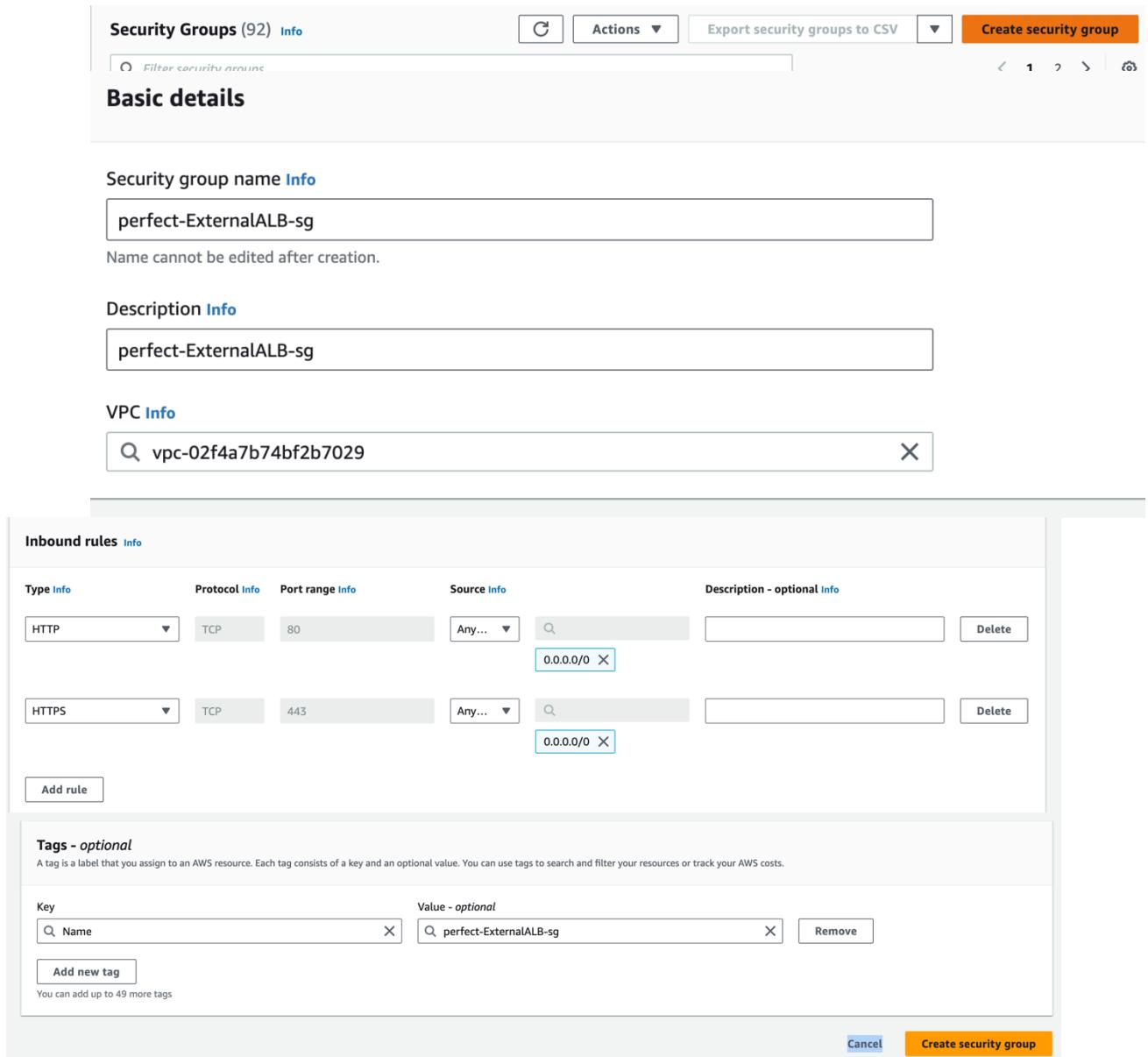
[Add rule](#)

**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.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="perfect-ExternalALB-sg"/> <input type="button" value="Remove"/>

[Add new tag](#)  
You can add up to 49 more tags

[Cancel](#) [Create security group](#)



- 13) Create security group for Bastion . Open traffic for (SSH traffic only to our computer) for inbound rules .

**Basic details**

**Security group name** [Info](#)  
perfect-Bastion-sg  
Name cannot be edited after creation.

**Description** [Info](#)  
Bastion server sec group

**Inbound rules** [Info](#)

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Cust... <input type="text" value="Q"/> <input type="button" value="Delete"/>	For local ip use only <input type="text" value="81.152.238.176/32"/> <input type="button" value="X"/>

**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.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="perfect-Bastion-sg"/> <input type="button" value="Remove"/>

**Add new tag**  
You can add up to 49 more tags

**Success message:** Security group (sg-0de5bf2965ef505e0 | perfect-Bastion-sg) was created successfully

- 14) Create security group for Nginx. Open traffic for (SSH from Bastion , HTTP AND HTTPS from External ALB) for inbound rules .

**Security group name** [Info](#)

Name cannot be edited after creation.

**Description** [Info](#)

**VPC** [Info](#)

Q vpc-02f4a7b74bf2b7029 X

Type	Protocol	Port range	Source	Description - optional	Action
HTTP	TCP	80	Cust...	http traffic from ALB	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>
HTTPS	TCP	443	Cust...	http traffic fromALB	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>
SSH	TCP	22	Cust...	ssh from Bastion	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>

**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	Value - optional	Action
<input type="text" value="Name"/>	<input type="text" value="perfect-Nginx-sg"/>	<span style="border: 1px solid #ccc; padding: 2px;">Remove</span>

Add new tag

You can add up to 49 more tags

Cancel
Create security group

✓ Security group (sg-0289118c69ab598aa | perfect-Nginx-sg) was created successfully
   
▶ Details



- 15) Create security group for internal ALB. Open traffic for (HTTP AND HTTPS) for inbound rules.

## Basic details

Security group name [Info](#)

perfect-InternalALB-sg

Name cannot be edited after creation.

Description [Info](#)

Sec group for perfect-InternalALB

VPC [Info](#)

vpc-02f4a7b74bf2b7029

X

### Inbound rules [Info](#)

Type Info

Protocol Info

Port range Info

Source Info

Description - optional info

HTTP

TCP

80

Any... ▾

0.0.0.0/0 X

HTTP access for Internal ALB

Delete

HTTP access for Internal ALB

HTTPS

TCP

443

Any... ▾

0.0.0.0/0 X

HTTPS access for Internal ALB

Delete

HTTPS access for Internal ALB

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.

Key

Q Name

Value - optional

Q perfect-InternalALB-sg

X Remove

Add new tag

You can add up to 49 more tags

Cancel

Create security group

✓ Security group ([sg-07fda2e31f7b82e77](#) | perfect-InternalALB-sg) was created successfully  
▶ Details

16) Create security group for Webserver. Open traffic for (SSH from Bastion , HTTP AND HTTPS) for inbound rules .

**Security group name [Info](#)**

perfect-Webservers-sg

Name cannot be edited after creation.

**Description [Info](#)**

Sec group for perfect-Webservers

**VPC [Info](#)**

vpc-02f4a7b74bf2b7029 X

Key	Value - optional
<input type="text"/> Name	<input type="text"/> perfect-Webserver-sg <span style="float: right;">X</span> <span style="margin-left: 10px;">Remove</span>
<span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 10px;">Add new tag</span> <small>You can add up to 49 more tags</small>	

[Cancel](#)

**Create security group**

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
SSH	TCP	22	Cust... <span style="float: right;">▼</span>	<input type="text"/> ssh access from bastion <span style="float: right;">Delete</span>
HTTPS	TCP	443	Cust... <span style="float: right;">▼</span>	<input type="text"/> https from the internal ALB <span style="float: right;">Delete</span> sg-0560f0ac11be8e6b0
HTTP	TCP	80	Cust... <span style="float: right;">▼</span>	<input type="text"/> http from the internal ALB <span style="float: right;">Delete</span> sg-0560f0ac11be8e6b0

<b>Tags - optional</b> <small>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.</small>				
Key	Value - optional			
<input type="text"/> Name	<input type="text"/> perfect-Webserver-sg <span style="float: right;">X</span> <span style="margin-left: 10px;">Remove</span>			
<span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 10px;">Add new tag</span> <small>You can add up to 49 more tags</small>				

[Cancel](#)

**Create security group**

⌚ Security group ([sg-0ee3978da16f5f6fb](#) | Perfect-Webservers-sg) was created successfully

▶ Details

- 16) Create security group for Data Layer. Our data layer has the Amazon EFS and RDS
- Our webserver needs to mount the file system and needs an NFS access.
  - Connect the webserver to the database (MySQL Access)
  - Bastion needs administrative access to the data base (MySQL Access)

### Basic details

Security group name [Info](#)  
 Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
MySQL/Aurora	TCP	3306	Cust... <input type="text" value="From Bastion"/>	
NFS	TCP	2049	Cust... <input type="text" value="From Webserver"/>	
MySQL/Aurora	TCP	3306	Cust... <input type="text" value="From Webserver"/>	

**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	Value - optional
<input type="text" value="Name"/>	<input type="text" value="perfect-datalayer-sg"/> Remove

Add new tag  
You can add up to 49 more tags

Cancel Create security group

⌚ Security group ([sg-04c6739bdff4d62eb](#) | [perfect-datalayer-sg](#)) was created successfully

▶ Details

All security groups are completed as shown below.

Name	Security group ID	Security group name	VPC ID	Description
perfect-DataLayer-sg	sg-0a1045ebc57645e34	perfect-DataLayer-sg	vpc-0c3c371436c0dc9d	for data layer
perfect-Nginx-sg	sg-0289118c69ab598aa	perfect-Nginx-sg	vpc-0c3c371436c0dc9d	Nginx allows traffic fro...
perfect-Bastion-sg	sg-0de5bf2965ef505e0	perfect-Bastion-sg	vpc-0c3c371436c0dc9d	All bastion sg
Perfect-Webservers...	sg-0ee3978da16f5f6fb	Perfect-Webservers-sg	vpc-0c3c371436c0dc9d	Sec Group for Webserv...
perfect-InternalALB...	sg-07fda2e31f7b82e77	perfect-InternalALB-sg	vpc-0c3c371436c0dc9d	Sec group for InternalLB
perfect-external-AL...	sg-06eec86a514f74e67	perfect-external-ALB-sg	vpc-0c3c371436c0dc9d	Sec group for external ...

17) The next steps would be to create a new domain name and register it putting it on route53 DNS.

Domain name created: “ptfd.link”

Request for a certificate:



## Request certificate

**Certificate type** [Info](#)  
ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

Request a public certificate  
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

Request a private certificate  
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit [AWS Private Certificate Authority](#).

[Cancel](#) [Next](#)

## Request public certificate

**Domain names**  
Provide one or more domain names for your certificate.

Fully qualified domain name [Info](#)

[Add another name to this certificate](#)  
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

**Validation method** [Info](#)  
Select a method for validating domain ownership.

DNS validation - recommended  
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Please note We put an asterisk \* to let our domain name have a wider range of choices and not restrict it.

**Tags** [Info](#)  
To help you manage your certificates, you can optionally assign your own metadata to each resource in the form of tags.

Tag key	Tag value - optional	Remove tag
<input type="text" value="Name"/>	<input type="text" value="perfect-cert"/>	<a href="#">Remove tag</a>

[Add tag](#)  
You can add 49 more tag(s).

[Cancel](#) [Previous](#) [Request](#)

Certificates (1)						
	<a href="#">C</a>	<a href="#">Delete</a>	<a href="#">Manage expiry events</a>	<a href="#">Import</a>	<a href="#" style="background-color: orange; color: white; border-radius: 5px; padding: 2px 10px;">Request</a>	
	Certificate ID	Domain name	Type	Status	In use	Renewal eligibility
<input type="checkbox"/>	<a href="#">1a17f000-0000-4000-a002-0919a0000000</a>	<a href="#">*.pftd.link</a>	Amazon Issued	Pending validation	No	Ineligible

**Success** Successfully requested certificate with ID [1a17f000-0000-4000-a002-0919a0000000](#)  
A certificate request with a status of pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

Certificate requested successfully but pending validation.

We create the records in Route DNS.

The screenshot shows the AWS Route 53 'Domains' page with one entry. At the top right, there are buttons for 'Create records in Route 53' (with a green arrow pointing to it) and 'Export to CSV'. Below the table, a green banner displays a success message: 'Successfully created DNS records' followed by a detailed log entry: 'Successfully created DNS records in Amazon Route 53 for certificate with ID 1ddc711660-8e196-4f99b-aed2-c953bea071af40'.

Once created ,on refresh ,Certificate issued below.

The screenshot shows the 'Certificate status' section of the AWS Certificate Manager. It displays a single certificate entry. The 'Identifier' column shows the ARN: 'arn:aws:acm:1ddc711660-8e196-4f99b-aed2-c953bea071af40'. The 'Status' column shows 'Issued' with a green checkmark icon.

18) We then proceed to create and customise our Amazon Elastic File System.

The screenshot shows the 'Elastic File System' console. On the left, a sidebar lists 'File systems', 'Access points', and links to 'AWS Backup', 'AWS DataSync', and 'AWS Transfer'. Below the sidebar, there's a 'Documentation' link. The main area is titled 'Amazon Elastic File System' and describes it as a 'Scalable, elastic, cloud-native NFS file system'. A sub-section titled 'Create file system' with the sub-instruction 'Create an EFS file system with recommended settings.' has an orange 'Create file system' button. A modal window titled 'Create file system' is open at the bottom, containing fields for 'Name - optional' (set to 'perfect-filesystem') and 'Virtual Private Cloud (VPC)' (set to 'vpc-02f4a7b74bf2b7029 PERFECT-vpc'). At the bottom of the modal are 'Cancel', 'Customize' (with a green arrow pointing to it), and 'Create' buttons.

▼ Tags optional

Add tags to associate key-value pairs to your resource. [Learn more](#)

Tag key	Tag value - optional
<input type="text" value="Name"/>	<input type="text" value="perfect-filesystem"/> <a href="#">X</a>
<a href="#">Remove tag</a>	
Use: "Name"	
<a href="#">Add tag</a>	

You can add 49 more tag(s)

[Cancel](#) [Next](#)

Specify the private subnets by routing into the file system as our webservers exists there. Once done the EFS becomes available to the webservers.

Fill in the network access .(Mount targets to the file system)

Virtual Private Cloud (VPC) [Learn more](#)

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

vpc-02f4a7b74bf2b7029	<a href="#">▼</a>
PERFECT-vpc	

**Mount targets**

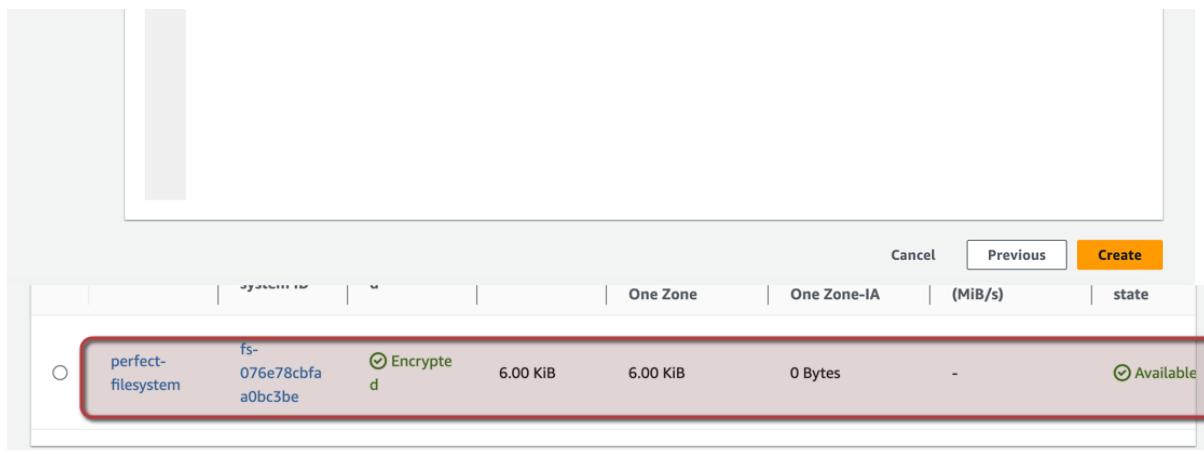
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)

Availability zone	Subnet ID	IP address	Security groups
us-east-1a	subnet-014a296e...	Automatic	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <a href="#">Choose security gr...</a> <a href="#">X</a>            sg-04c6739bdff4d62e            b            perfect-datalayer-sg         </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <a href="#">Choose security gr...</a> <a href="#">X</a>            sg-04c6739bdff4d62e            b            perfect-datalayer-sg         </div>
us-east-1b	subnet-01db8816...	Automatic	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <a href="#">Choose security gr...</a> <a href="#">X</a>            sg-04c6739bdff4d62e            b            perfect-datalayer-sg         </div>

[Add mount target](#)

[Cancel](#) [Previous](#) [Next](#)

Click Next and Create the EFS. It is also important to note that EFS file system is created and encrypted.



## 19) Create an access point for WordPress webserver.

The screenshot shows the 'Access points' section of the AWS EFS console. The top navigation bar includes tabs for Metered size, Monitoring, Tags, File system policy, **Access points**, Network, and Replication. The 'Access points' tab is selected, showing 'Access points (0)' and a 'Create access point' button. A green arrow points to this button. Below this, there's a search bar and a table with columns for Name, Access point ID, File system ID, Path, POSIX user, Creation info, and State. The table displays 'No resources'. Under the 'File system' section, there's a dropdown menu currently set to 'Select file system' and a note about choosing the file system associated with the access point. There are fields for 'Name - optional' (containing 'wordpress') and 'Root directory path - optional' (containing '/wordpress'). A note at the bottom says 'Example: "/foo/bar"'.

#### POSIX user - optional

The full POSIX identity on the access point that is used for all file operations by NFS clients. [Learn more](#)

##### User ID

POSIX user ID used for all file system operations using this access point.

Accepts values from 0 to 4294967295

##### Group ID

POSIX group ID used for all file system operations using this access point.

Accepts values from 0 to 4294967295

#### Root directory creation permissions - optional

EFS will automatically create the specified root directory with these permissions if the directory does not already exist.

##### Owner user ID

Owner user ID for the access point's root directory, if the directory does not already exist.

Accepts values from 0 to 4294967295

##### Owner group ID

Owner group ID for the access point's root directory, if the directory does not already exist.

Accepts values from 0 to 4294967295



Success!

Access point (fsap-072cbbfc6545251a0) is available.

## 20) Create an access point for tooling

**Tags - optional**

Add tags to associate key-value pairs to your resource. [Learn more](#)

Tag key Tag value - optional

You can add 49 more tag(s)

**Access points (2)**

< 1 >

Name	Access point ID	Path	POSIX user	Creation i...
wordpress	<a href="#">fsap-072cbbfc6545251a0</a>	/wordpress	0 : 0	0 : 0 (0755)
tooling	<a href="#">fsap-0291abab799a6e7f0</a>	/tooling	0 : 0	0 : 0 (0755)

We have completed our EFS creation. Next would be creating our RDS but we need to create a KMS key to be used to encrypt the database instance .

## 21)KMS Key and RDS instance creation

### AWS Key Management Service

Easily create keys and control encryption across AWS and beyond

AWS Key Management Service (KMS) is a managed service that makes it easy to create and manage keys and control the use of encryption across a wide range of AWS services. KMS is a secure and resilient service that uses FIPS 140-2 validated hardware security modules to isolate and protect your keys.

**Get started now**

You can create a key by clicking the button below.

**Create a key**

**Configure key**

**Key type** [Help me choose](#)

**Symmetric**  
A single key used for encrypting and decrypting data or generating and verifying HMAC codes.

**Asymmetric**  
A public and private key pair used for encrypting and decrypting data or signing and verifying messages.

**Key usage** [Help me choose](#)

**Encrypt and decrypt**  
Use the key only to encrypt and decrypt data.

**Generate and verify MAC**  
Use the key only to generate and verify hash-based message authentication codes (HMAC).

**Advanced options**

**Tags - optional**

perfect-rds

**Description - optional**  
You can change the description at any time.

Description  
for the RDS instances

**Tags - optional**

You can use tags to categorise and identify your KMS keys and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. [Learn more](#)

Tag key  Name  Tag value - optional  perfect-rds

## Success

Your AWS KMS key was created with alias **perfect-rds** and key ID b

Please note: Ensure you only designate database access to only one who needs access to it .

22) Next would be to create a DB subnet group for our private subnet

Subnet groups (0) C Edit Delete Create DB subnet group

Filter by subnet group

Name	Description	Status	VPC
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-055865b7d7c9a174d (10.0.7.0/24) X  
subnet-0b4318fd34f0f7016 (10.0.5.0/24) X

**Subnets selected (2)**

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-055865b7d7c9a174d	10.0.7.0/24
us-east-1a	subnet-0b4318fd34f0f7016	10.0.5.0/24

Cancel Create

⌚ Successfully created perfect-rds-subnet. [View subnet group](#)

RDS > Subnet groups

**Subnet groups (1)**

C Edit Delete Create DB subnet gro

Filter by subnet group < 1 >

<input type="checkbox"/>	Name	Description	Status	VPC
<input type="checkbox"/>	<a href="#">perfect-rds-subnet</a>	for RDS subnet	Complete	vpc-02f4a7b74bf2b7029

Subnet group successfully created .Next would be to create a database .

23) Create database .When creating it we would be choosing the free tier template as it won't give us access to be able to use the KMS key but if selecting the dev, test or production ,You would have access to the KMS and it comes with an expensive cost .

Databases (0)

Group resources C Modify Actions ▾ Restore from S3 Create database

Filter by databases < 1 > ⚙

## Choose username and password details .

### 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.

perfect-database

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.

PerfectAdmin

### Connectivity [Info](#)

#### Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting 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**

Set up a connr this database

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

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

PERFECT-vpc (vpc-02f4a7b74bf2b7029)

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 are selected.

perfect-rds-subnet

2 Subnets, 2 Availability Zones

Click No so that our RDS instance wouldn't be accessible . The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.

# Choose the data layer security group

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

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

PERFECT-vpc (vpc-02f4a7b74bf2b7029)

6 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

i 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.

perfect-rds-subnet

2 Subnets, 2 Availability Zones

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 your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

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.

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing

Choose existing VPC security groups

Create new

Create new VPC security group

Existing VPC security groups

Choose one or more options

perfect-datalayer-sg X

Availability Zone [Info](#)

us-east-1a

## ▼ Additional configuration

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

## Database options

Initial database name [Info](#)

test

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

Database created.

24) Our next task would be to create an auto scaling group which would need some hosts of resources to be created namely.

- a) AMI.
- b) Launch Template
- c) Target Group
- d) Load balancer
- e) Autoscaling group

We would need to create a launch template but to do that we need to create 3 instances for our bastion, nginx and web servers and then we would be able to create 3 AMI respectively and we have to have some software installed inside them. Then we would come up with their user data. With this 2 ,then our launch template would be created

	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Bastion	i-0e1a12f467cdb91a5	Running	t2.micro
<input type="checkbox"/>	Nginx	i-0876c40cfe5716499	Running	t2.micro
<input type="checkbox"/>	Webserver	i-0c4d3fe9f282b6b57	Running	t2.micro

25) For Bastion: The installation is below.

```
lec2-user@Bastion ~]$ sudo su -
[root@Bastion ~]# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
yum install -y dnf-utils http://rpms.remirepo.net/enterprise/remi-release-8.rpm
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs)
```

```
[root@Bastion ~]# yum install wget vim python3 telnet htop git mysql net-tools chrony -y
Updating Subscription Management repositories.
Unable to read consumer identity
```

## WE installed python, ntp, net tools, vim, wget, telnet, epel-release-stop

```
[root@Bastion ~]# yum install wget vim python3 telnet htop git mysql net-tools chrony -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:04:26 ago on Fri 13 Oct 2023 03:57:55 PM UTC.
Package python3-3.9.16-1.el9.x86_64 is already installed.
Package chrony-4.3-1.el9.x86_64 is already installed.
Dependencies resolved.

=====
 Package           Architecture   Version          Repository      Size
=====
Installing:
 git              x86_64         2.39.3-1.el9_2    rhel-9-appstream-rhui-rpms  66 k
 htop             x86_64         3.2.1-1.el9_2    epel            170 k
 mysql            x86_64         8.0.32-1.el9_2   rhel-9-appstream-rhui-rpms  2.8 k
 net-tools        x86_64         2.0-0.62.20140912git.el9   rhel-9-basesos-rhui-rpms  309 k
 telnet           x86_64         1:0.17-85.el9    rhel-9-appstream-rhui-rpms  66 k
 vim-enhanced    x86_64         2:8.2.2637-28.el9_1  rhel-9-appstream-rhui-rpms  1.8 k
 wget             x86_64         1.21-17.el9     rhel-9-appstream-rhui-rpms  70 k

=====

```

Complete!

```
[root@Bastion ~]# systemctl start chronyd
```

```
[systemctl enable chronyd
[root@Bastion ~]# ]
```

Once installed the chronyd is started we should create the AMI.

26)For Nginx: The installation is below.

We are carrying out the same installation but also adding the selinux configuration policies for the nginx server for the website to function properly.

We are also installing a self-signed certificate for our nginx and this is because the port must be set to 443 which is what nginx listens to.

```
[ec2-user@ip-172-31-21-30 ~]$ sudo hostname nginx
[ec2-user@ip-172-31-21-30 ~]$ bash
[ec2-user@nginx ~]$ ]
```

```
[ec2-user@nginx ~]$ sudo su -
[root@nginx ~]# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
yum install -y dnf-utils http://rpms.remirepo.net/enterprise/remi-release-8.rpm
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMS)
Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMS)          43 MB/s | 25 MB   0
                                                               38 MB/s | 14 MB   0

[root@nginx ~]# yum install wget vim python3 telnet htop git mysql net-tools chrony -y
systemctl start chronyd
systemctl enable chronyd
Updating Subscription Management repositories.
```

We are carry out the same installation but also adding the selinux configuration policies for the webservers. this is because the port must be set to 443 which is what webserver listens to .

```
[root@nginx ~]# setsebool -P httpd_can_network_connect=1
setsebool -P httpd_can_network_connect_db=1
setsebool -P httpd_execmem=1
[setsebool -P httpd_use_nfs 1
[root@nginx ~]# █
```

Please note that the reason we are performing it here is that our user-data exceed the 6kb limit .

```
[root@nginx ~]# git clone https://github.com/aws/efs-utils
cd efs-utils
yum install -y make
yum install -y rpm-build
make rpm
yum install -y ./build/amazon-efs-utils*rpm
Cloning into 'efs-utils'...
remote: Enumerating objects: 1449, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 1449 (delta 38)  reused 29 (delta 29)  pack-reused 1397
```

## Installing self-signed certificate for nginx

```
Country Name (2 letter code) [XX]:UK
State or Province Name (full name) []:Manchester
Locality Name (eg, city) [Default City]:Manchester
Organization Name (eg, company) [Default Company Ltd]:Devops
Organizational Unit Name (eg, section) []:Devops
Common Name (eg, your name or your server's hostname) []:ip-172-31-21-30.ec2.internal
Email Address []:mrincredible752@outlook.com
[root@nginx efs-utils]# █
```

Creating a dhparam and wait for a couple of minute to get it fully created .

```
[root@nginx efs-utils]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/perfect.key -out /etc/ssl/certs/perfect.crt
```

Please confirm the .key and .crt files are present.  
Nginx successfully done and we would create AMI next.

27)For Webserver : The installation is below  
We are carrying out the same installation but also adding the selinux configuration policies for the webserver for the website to function properly.

We are also installing a self-signed certificate for our Apache webserver and this is because the port must be set to 443 which is what webserver listens to.

```
[[ec2-user@ip-172-31-28-13 ~]$ sudo hostname webservers
[[ec2-user@ip-172-31-28-13 ~]$ bash
[ec2-user@webservers ~]$ █
```

After installation we would edit the Apache configuration file

```
[root@webservers ~]# vi /etc/httpd/conf.d/ssl.conf
[root@webservers ~]# █
```

Change from localhost.crt to perfect.cert

Change from localhost.key to perfect.key

```
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile /etc/pki/tls/certs/perfect.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file. Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile /etc/pki/tls/private/perfect.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
```

Once saved we would be creating an AMI from it .  
Hence we are creating AMI for each of the 3 server .

## 28)Webserver AMI

**Instances (1/3) Info**

Name	Instance ID	Instance state	Instance type	Status check
Bastion	i-0e1a12f467cdb91a5	Running	t2.micro	2/2 checks passed
Nginx	i-0876c40fce5716499	Running	t2.micro	2/2 checks passed
Webserver	i-0c4d3fe9f282b6b57	Running	t2.micro	2/2 checks passed

**Actions ▲** **Launch instances ▼**

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Create image** (highlighted with a green arrow)
- Image and templates (highlighted with a green box)
- Monitor and troubleshoot

**Instance: i-0c4d3fe9f282b6b57 (Webserver)**

⌚ Currently creating AMI ami-016d3cf73eabe8748 from instance i-0c4d3fe9f282b6b57. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

Webserver AMI successfully created

30) Bastion AMI successfully created

**Instances (1/3) Info**

Name	Instance ID	Instance state	Instance type	Status check
Bastion	i-0e1a12f467cdb91a5	Running	t2.micro	2/2 checks passed
Nginx	i-0876c40fce5716499	Running	t2.micro	2/2 checks passed
Webserver	i-0c4d3fe9f282b6b57	Running	t2.micro	2/2 checks passed

**Actions ▲** **Launch instances ▼**

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Create image** (highlighted with a red box)
- Image and templates (highlighted with a red box)
- Monitor and troubleshoot

**Instance: i-0e1a12f467cdb91a5 (Bastion)**

⌚ Currently creating AMI ami-0e5bf5ed7003b3a9b from instance i-0e1a12f467cdb91a5. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

Bastion AMI successfully created

31) NGINX AMI

**Instances (1/3) Info**

Name	Instance ID	Instance state	Instance type	Status check
Bastion	i-0e1a12f467cdb91a5	Running	t2.micro	2/2 checks passed
Nginx	i-0876c40fce5716499	Running	t2.micro	2/2 checks passed
Webserver	i-0c4d3fe9f282b6b57	Running	t2.micro	2/2 checks passed

**Actions ▲** **Launch instances ▼**

- Networking
- Security
- Create image** (highlighted with a blue box)
- Image and templates (highlighted with a blue box)
- Monitor and troubleshoot

**Instance: i-0876c40fce5716499 (Nginx)**

⌚ Currently creating AMI ami-0b7202de4916d7028 from instance i-0876c40fce5716499. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

NGINX AMI successfully created

### 32) Create Target group for the Nginx, WordPress Server and WordPress Tooling Webserver.

Please note we are not creating any target group for Bastion server because it is not behind any of the load balancers

All target groups created

	Name	ARN	Port	Protocol	Target type
<input type="checkbox"/>	perfect-nginx-TG	<a href="#">arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/perfect-nginx-TG/12345678901234567890</a>	443	HTTPS	Instance
<input type="checkbox"/>	perfect-tooling-TG	<a href="#">arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/perfect-tooling-TG/12345678901234567890</a>	443	HTTPS	Instance
<input type="checkbox"/>	perfect-wordpress-TG	<a href="#">arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/perfect-wordpress-TG/12345678901234567890</a>	443	HTTPS	Instance

All AMI created

Amazon Machine Images (AMIs) (3) <a href="#">Info</a>		
Owned by me	<input type="text"/> Find AMI by attribute or tag	
	Name	AMI ID
<input type="checkbox"/>	Nginx-ami	<a href="#">ami-0b7202de4916d7028</a>
<input type="checkbox"/>	Bastion-ami	<a href="#">ami-0e5bf5ed7003b3a9b</a>
<input type="checkbox"/>	webserver-ami	<a href="#">ami-016d3cf73eabe8748</a>

### 33)Create the external and internal load balancers and ensure they are on https :443

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

**perfect-external-alb**

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

Scheme | Info  
Scheme can't be changed after the load balancer is created.  
 Internet-facing

## 34) Create listener rules for the internal load balancer for WordPress and tolling

**Load balancer: perfect-Internal-alb**

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

<input type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy
<input type="checkbox"/>	HTTPS:443	Forward to target group <ul style="list-style-type: none"> <li><a href="#">perfect-wordpress-TG</a>: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	<a href="#">1 rule</a>	<a href="#">ARN</a>	ELBSecurityPolicy-TLS13-1-2-...

**Listener rules (2) [Info](#)**

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

<input type="checkbox"/>	Name tag	Priority	Conditions (If)	Actions (Then)
<input type="checkbox"/>	My Perfect Tooling	2	HTTP Host Header is tooling.pftd.link OR www.tooling.pftd.link	Forward to target group <ul style="list-style-type: none"> <li><a href="#">perfect-tolling-TG</a>: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none"> <li><a href="#">perfect-wordpress-TG</a>: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>

## 35) Create Launch Templates

Compute

### EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

Create launch template

Benefits and features

## a) For Bastion Launch Template

**Create launch template**

Creating a launch template allows you to create a saved instance configuration that can be reused, shared at a later time. Templates can have multiple versions.

**Launch template name and description**

Launch template name - *required*

perfect-bastion-LT

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description

for bastion

Max 255 chars

Select your Bastion AMI, Instance type ,Key Pair and in a public subnet

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

Recents | My AMIs | Quick Start

Don't include in launch template  Owned by me  Shared with me

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

perfect-Bastion-ami  
ami-0e5bf5ed7003b3a9b  
2023-10-13T17:11:21.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description

AMI for Bastion  Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name  [Create new key pair](#)

▼ Network settings [Info](#)

Subnet Info

subnet-069b4d1c58d3da588 perfect-public-subnet-2  
VPC: vpc-02f4a7b74bf2b7029 Owner: 416591745024 Availability Zone: us-east-1b  
IP addresses available: 250 CIDR: 10.0.2.0/24

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

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.

Select existing security group  Create security group

Common security groups [Info](#)

Select security groups

perfect-Bastion-sg sg-0d6e495529edd508d X  
VPC: vpc-02f4a7b74bf2b7029

Compare security group rules

## Add a network interface ,Add a security group and enable public IP

▼ Advanced network configuration

Network interface 1	Network interface 2	Network interface 3
Device index <a href="#">Info</a> 0	Network interface <a href="#">Info</a> New interface	Description <a href="#">Info</a>
Subnet <a href="#">Info</a> subnet-069b4d1c58d3da588 IP addresses available: 250	Security groups <a href="#">Info</a> Select security groups	Auto-assign public IP <a href="#">Info</a> Enable
	perfect-Bastion-sg X sg-0d6e495529edd508d vpc-02f4a7b74bf2b7029	
<input type="checkbox"/> Hide all selected		

## Add the user data and create the Launch template.

User data - *optional* [Info](#)

Upload a file with your user data or enter it in the field.

 Choose file

```
#!/bin/bash
yum install -y mysql
yum install -y git tmux
yum install -y ansible
```

```
#!/bin/bash
yum install -y mysql
yum install -y git tmux
yum install -y ansible
```

User data has already been base64 encoded

Storage (volumes)  
1 volume(s) - 10 GiB

 **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

 Create launch template

Subnet [Info](#)

Don't include in launch template

Not applicable for EC2 Auto Scaling

Security groups [Info](#)

Select security groups ▾

perfect-Bastion-sg   
sg-  
0d6e495529edd508d  
vpc-  
02f4a7b74bf2b7029

Hide all selected

Auto-assign public IP [Info](#)

Enable

 Success

Successfully created [perfect-bastion-LT](#)

Bastion Launch Template created.

b) For Nginx Launch Template, perform the same task but ensure the auto assign IP is disabled.

The screenshot shows the AWS Lambda function configuration interface. It displays two separate configurations for different launch templates:

**Nginx Configuration:**

- Subnet**: Info
- Don't include in launch template**
- Not applicable for EC2 Auto Scaling**
- Security groups**: Info
- Select security groups**: ▾
- perfect-Nginx-sg** X
- sg-06d581b44e0528942
- vpc-02f4a7b74bf2b7029
- Auto-assign public IP**: Info
- Enable**

**Web Servers Configuration:**

- Subnet**: Info
- Don't include in launch template**
- Not applicable for EC2 Auto Scaling**
- Security groups**: Info
- Select security groups**: ▾
- perfect-Webservers-** X
- sg-031c3ccf6991e37d2
- vpc-02f4a7b74bf2b7029
- Auto-assign public IP**: Info
- Disable**

Please note : For WordPress :It would require mounting on EFS and this can be gotten when you navigate to the EFS access point by clicking “attach” button

```
sudo mount -t efs -o tls, accesspoint=fsap-072cbbfc6545251a0 fs-076e78cbfaa0bc3be:/var/www/
```

Also navigate to the database and copy the endpoint to add into the user data .After making this adjustment then it can be created .

```

1  #!/bin/bash
2  mkdir /var/www/
3  sudo mount -t efs -o tls,accesspoint=fsap-072cbbfc6545251a0 fs-076e78cbfaa0bc3be:/ /var/www/
4  yum install -y httpd
5  systemctl start httpd
6  systemctl enable httpd
7  yum module reset php -y
8  yum module enable php:remi-7.4 -y
9  yum install -y php php-common php-mbstring php-opcache php-intl php-xml php-gd php-curl php-mysqlnd php-fpm php-json
10 systemctl start php-fpm
11 systemctl enable php-fpm
12 wget http://wordpress.org/latest.tar.gz
13 tar xzvf latest.tar.gz
14 rm -rf latest.tar.gz
15 cp wordpress/wp-config-sample.php wordpress/wp-config.php
16 mkdir /var/www/html/
17 cp -R /wordpress/* /var/www/html/
18 cd /var/www/html/
19 touch healthstatus
20 sed -i "s/localhost/perfect-database.firebaseio.com/g" wp-config.php
21 sed -i "s/username_here/PerfectAdmin/g" wp-config.php
22 sed -i "s/password_here/admin12345/g" wp-config.php
23 sed -i "s/database_name_here/wordpressdb/g" wp-config.php
24 chcon -t httpd_sys_rw_content_t /var/www/html/ -R
25 systemctl restart httpd

```

The 4 launch templates are created using their respective AMI and user data.

Launch Templates (1/4) <a href="#">Info</a>							<a href="#">Actions</a>	<a href="#">Create launch template</a>
							<a href="#">Search</a>	
Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By			
lt-008cccd8c47058a73	perfect-tooling-LT	1	1	2023-10-14T09:20:25.000Z	arn:aws:iam::416591745024:r...			
lt-02bede64f283a0d13	perfect-Nginx-LT	1	1	2023-10-14T08:58:13.000Z	arn:aws:iam::416591745024:r...			
lt-0e8255c3a032f5f91	perfect-wordpress-LT	1	1	2023-10-14T08:51:04.000Z	arn:aws:iam::416591745024:r...			
lt-0ab2cf61d36622724	perfect-bastion-LT	1	1	2023-10-14T07:44:19.000Z	arn:aws:iam::416591745024:r...			

- 36) Create Auto scaling group. For bastion and nginx first and then proceed to create the tooling and webserver. But we need to create the actual database first and then create it .

37) We would be using the SSH-AGENT to get into the instance as shown below to have access and create the tooling data base and the WordPress database.

```
[ec2-user@ip-10-0-0-190 ~]$ mysql -h perfect-database.cnfvqr8ijadp.us-east-1.rds.amazonaws.com -u PerfectAdmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

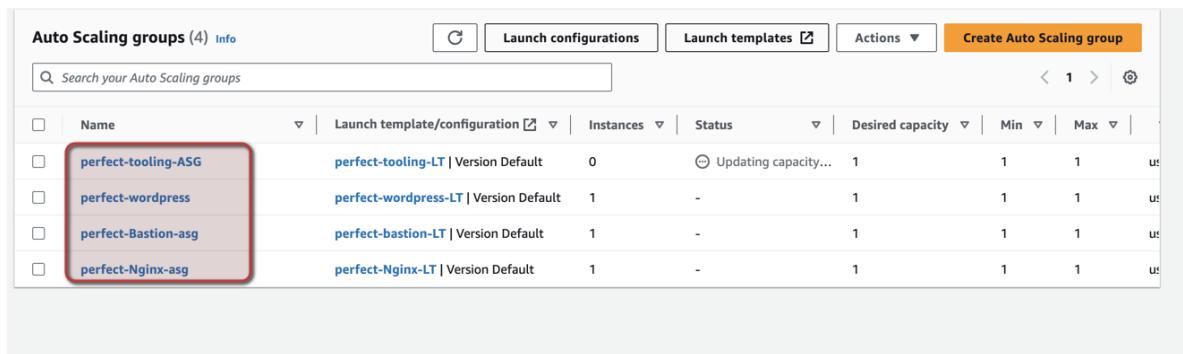
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database wordpressdb;
Query OK, 1 row affected (0.01 sec)

mysql> create database toolingdb;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
| toolingdb |
| wordpressdb |
+-----+
7 rows in set (0.01 sec)
```

After creating we now have the 4 Auto scaling groups



Auto Scaling groups (4) <a href="#">Info</a>							
<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	perfect-tooling-ASG	perfect-tooling-LT   Version Default	0	Updating capacity...	1	1	1
<input type="checkbox"/>	perfect-wordpress	perfect-wordpress-LT   Version Default	1	-	1	1	1
<input type="checkbox"/>	perfect-Bastion-asg	perfect-bastion-LT   Version Default	1	-	1	1	1
<input type="checkbox"/>	perfect-Nginx-asg	perfect-Nginx-LT   Version Default	1	-	1	1	1

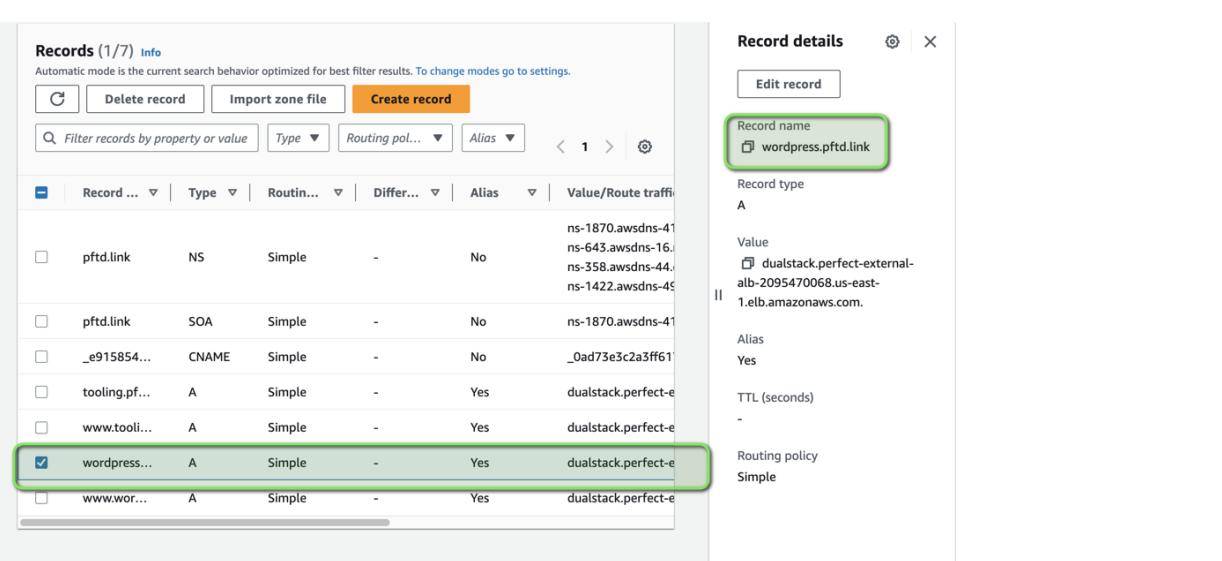
38) Create records for our tooling webserver and WordPress webserver as shown below. All traffic from the website should be forwarded to the external load balancer.

Add [www.tooling.pftd.link](http://www.tooling.pftd.link) or tooling.pftd.link

Add [www.wordpress.pftd.link](http://www.wordpress.pftd.link) or wordpress.pftd.link as shown below.

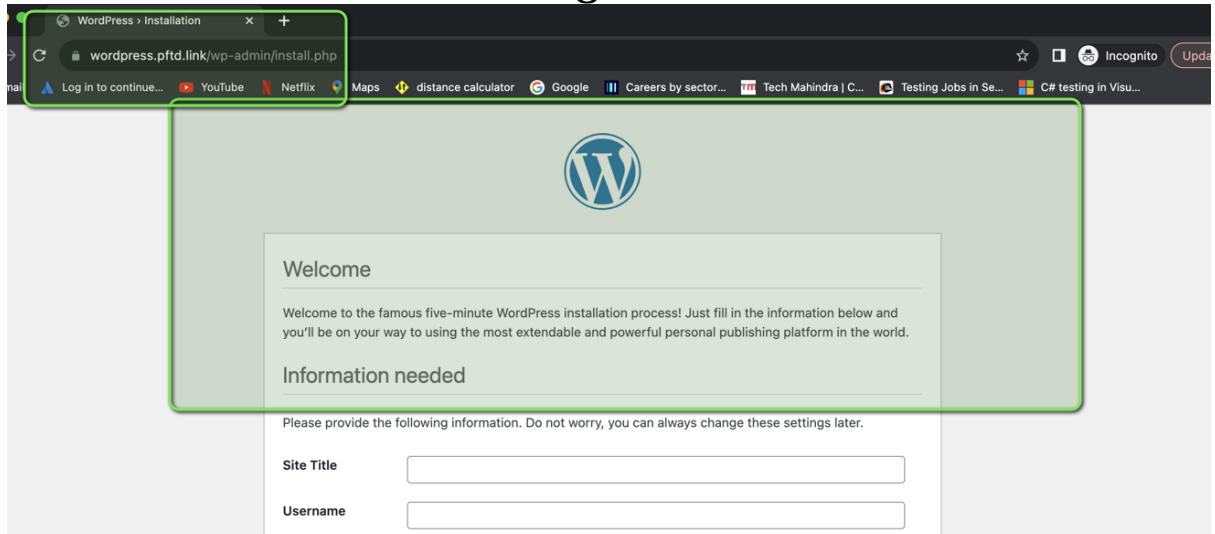


Record Name	Type	Behavior	TTL	Value
tooling.pftd.link	A	Simple	-	dualstack.perfect-external-alb-2095470068.us-east-1.elb.amazonaws.com.
www.tooling.pftd.link	A	Simple	-	dualstack.perfect-external-alb-2095470068.us-east-1.elb.amazonaws.com.
wordpress.pftd.link	A	Simple	-	dualstack.perfect-external-alb-2095470068.us-east-1.elb.amazonaws.com.
www.wordpress.pftd.link	A	Simple	-	dualstack.perfect-external-alb-2095470068.us-east-1.elb.amazonaws.com.

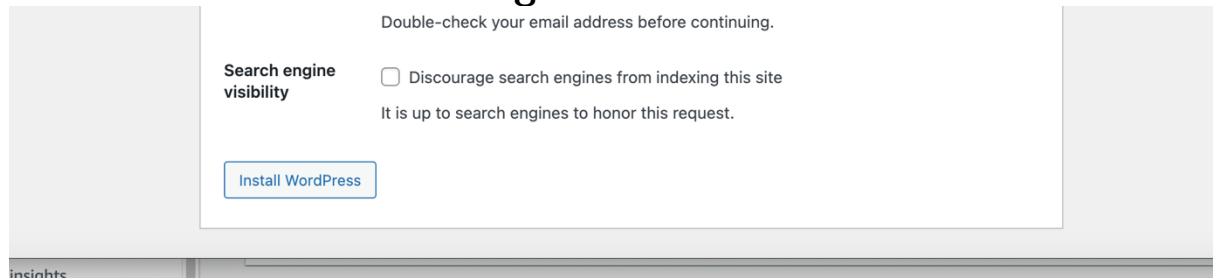
Record name	Value
wordpress.pftd.link	dualstack.perfect-external-alb-2095470068.us-east-1.elb.amazonaws.com.

39) Launch the website using the link for WordPress.

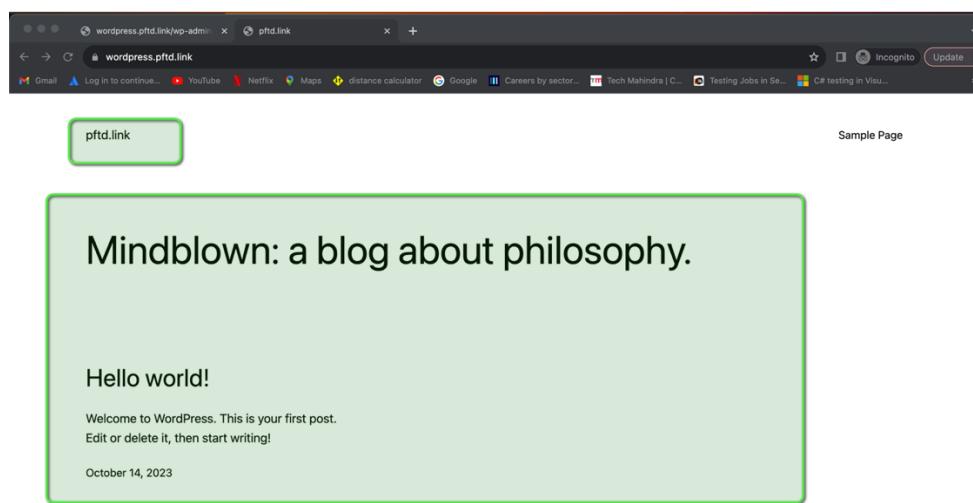


WordPress Launched successfully.

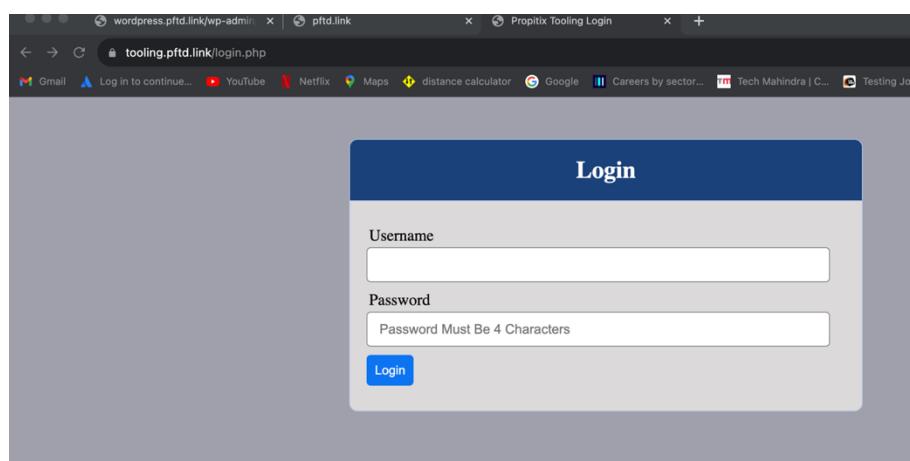
## 40)Install WordPress and log in the credentials.



Our WordPress is successfully launched.



## 41)Launch the website using the link for tooling.



42) Tooling website launched successfully.

In Conclusion, we have successfully launched a website. This process starts when the client (user or DevOps Engineer ) types the website on the browser which goes through the route53 DNS to the internet gateway into the External load balancer which sends the traffic to the Nginx reverse Proxy then routes into the Internal Load Balancer .

The internal Load balancer checks the host headers and if it recognizes it is default, it routes to the WORDPRESS but if otherwise it routes to the tooling website and then target renders the website for our view .

Congratulations We have successfully launched 2 company website using AWS Cloud Solution.

Please remember to delete all resources once the project is completed.

Thanks