



PROJECT-1

Name: A Bharath

Course: Aws, Devops

Project: Aws 3- tier Architecture

Mobile: 9391939511

Email: bharathraising91@gmail.com

INTRODUCTION OF 3-TIER ARCHITECTURE

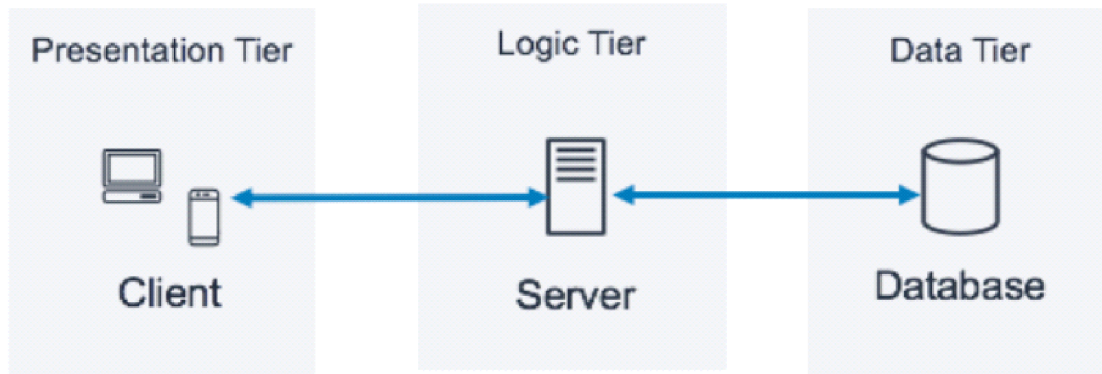
The three-tier architecture is the most popular implementation of a multi-tier architecture and consists of a single presentation tier (or) web tier, Application tier (or) logic tier, and database tier.

Presentation Tier: Its main purpose is to display information to and collect information from the user.

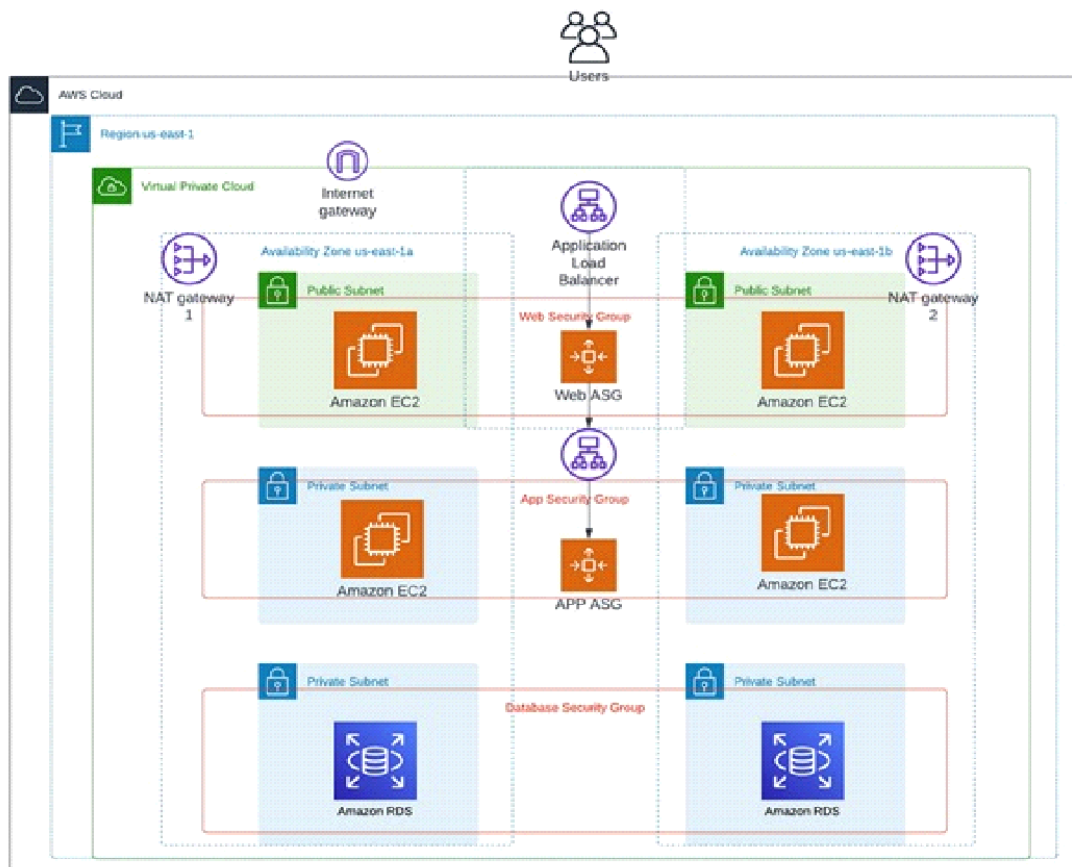
Application Tier: Information that is collected in the presentation tier is processed - sometimes against other information in the data tier. The application tier can also add, delete, or modify data in the data tier.

Database Tier: The data tier, sometimes called database tier, data access tier or back-end, is where the information that is processed by the application is stored and managed.

Benefits of 3-Tier Architecture: Scalability, Reliability, Security, Flexibility.



ARCHITECTURE:



VPC :

VPC stands for virtual private cloud, it gives you full control over your virtual networking environment, including resource placement, connectivity, and security

The purpose of VPC is a secure, isolated private cloud hosted within a public cloud.

- Creating a VPC and its components
- VPC components:
 - 1.subnets-> public, private
 - 2.internet gateway
 - 3.route Tables
 - 4.NAT gateway
- Using above architecture we need to create a VPC and 2 public subnets in 2 availability zones and 4 private subnets in 2 availability zones
- Login to the AWS management console and in search bar search VPC
- Click on VPC and Create VPC
- Select vpc only and give name
- Enter the IPV4 CIDR
- Click on create VPC

The screenshot displays the AWS VPC dashboard. On the left, a sidebar lists navigation options under 'Virtual private cloud', including 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', and 'Elastic IPs'. The main area is titled 'Your VPCs (1/1)' and shows a table with one VPC:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
Project	vpc-03cef1371c02b777d	Available	150.0.0.0/16	-

Below the table, the selected VPC 'vpc-03cef1371c02b777d / Project' is shown with tabs for 'Details', 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. The 'Details' tab is currently active.

VPC > Your VPCs > vpc-03cef1371c02b777d

vpc-03cef1371c02b777d / Project Actions ▼

Details [Info](#)

VPC ID vpc-03cef1371c02b777d	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default VpcDetails	DHCP option set dopt-0c64d69787bef3a13	Main route table rtb-085c1d88768e28fa2	Main network ACL acl-0712dd4941b3b4f0e
Default VPC No	IPv4 CIDR 150.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 326011653143	

[Resource map](#) | [CIDRs](#) | [Flow logs](#) | [Tags](#) | [Integrations](#)

Resource map [Info](#)

VPC [Show details](#)
Your AWS virtual network

Subnets (6)
Subnets within this VPC

Route tables (1)
Route network traffic to resources

- Here VPC is successfully created

Subnet :

A subnet is a range of IP addresses in your VPC. You launch AWS resources, such as Amazon EC2 instances, into your subnets. You can connect a subnet to the internet, other VPCs and your own data centers.

- We need to create 6 subnets, that are 2 public subnets in 2 zones and 4 private subnets, 2 private subnets are 1a zone for autoscaling, another 2 private subnets are 1b zone for database
- Go to subnets and click on create subnets
- Select your VPC, in subnet settings give subname
- Select availability zone and give IPV4 VPC CIDR block
- Now click on Add subnets
- Click on add subnets we have to create 5 more subnets
- Click on create subnets

Subnets (6) Info						Last updated 4 minutes ago	Actions	Create subnet
Find resources by attribute or tag								
<input type="checkbox"/> Name	Subnet ID	State	VPC	IPv4				
<input type="checkbox"/> Public-subnet2	subnet-017dcd862286e6b4b	Available	vpc-03cef1371c02b777d Project	150				
<input type="checkbox"/> Public-subnet-1	subnet-08a1fc516335dea57	Available	vpc-03cef1371c02b777d Project	150				
<input type="checkbox"/> Private-subnet1	subnet-0ef7e002a779f612f	Available	vpc-03cef1371c02b777d Project	150				
<input type="checkbox"/> Private-subnet2	subnet-0896c0436704b8b8d	Available	vpc-03cef1371c02b777d Project	150				
<input type="checkbox"/> Private-subnet3	subnet-0d6f69fc5cf5bc84e	Available	vpc-03cef1371c02b777d Project	150				
<input type="checkbox"/> Private-subnet4	subnet-02bdd52c17fba15f9	Available	vpc-03cef1371c02b777d Project	150				

Internet gateway:

Internet gateways provide two-way public connectivity to applications running in AWS Regions or in Local Zones. It is used to enable you to connect to an EC2 instance in AWS using your local computer.

- Next we need to create one internet gateway
- Click on create internet gateway and attach to your VPC to internet gateway

Internet gateways (1) Info					Last updated 4 minutes ago	Actions	Create internet gateway
Search <input type="text"/>							
<input type="checkbox"/> Name	Internet gateway ID	State	VPC ID				
<input type="checkbox"/> Project-igw	igw-012498e3baf0b95c2	Attached	vpc-03cef1371c02b777d Project				

NAT:

NAT stands for Network Address Translator. It is a highly available AWS managed service that makes it easy to connect to the Internet from instances within a private subnet in an Amazon VPC.

- According to 3-tier architecture we need to create two NAT gateways
- NAT gateway
- Create NAT gateway and give the name, select your public subnet

- Click on Allocate Elastic IP now you can create your NAT gateway

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

Public-NAT1

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

subnet-08a1fc516335dea57 (Public-subnet-1)

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-00bf4aa632a5a0260

[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 Value - optional

Q Name X Q Public-NAT1 X Remove

Add new tag

- Also create one more NAT gateway, select public subnet2
- Click on create NAT gateway

Find resources by attribute or tag

	Name	NAT gateway ID	Connectivity...	State	State message	Primary pu
<input type="radio"/>	Private-NAT1	nat-0d4747462f1910668	Public	Pending	–	–
<input type="radio"/>	Public-NAT1	nat-04dc908995b39d290	Public	Pending	–	–

Route Table:

A route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed. It is used to determine which way to forward traffic.

- Create 3 route tables, 1 is for public subnets, next one is private subnets and next one is database subnets
- Give the name and select your VPC
- Click on create route table
- Create 2 more route tables

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 type="text" value="Name"/>	<input type="text" value="public-route"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

- Select public route table and go to actions
- Click on edit routes and add routes
- Attach internet gateway and click on save changes
- Now go to subnet associations and click on edit subnet associations
- Select public subnets and click on save changes
- For private route tables you can attach NAT gateways

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
<input checked="" type="checkbox"/> private-route2	rtb-0ae63c441a85047e6	2 subnets	-	No	vpc-0e00b3fbd7ee5bd7a 3-tie...	730335267904
<input type="checkbox"/> -	rtb-00ef741bd71604e53	-	-	Yes	vpc-0e00b3fbd7ee5bd7a 3-tie...	730335267904
<input type="checkbox"/> -	rtb-006ab3cd8d6538f50	-	-	Yes	vpc-0186fabe084a126fa	730335267904
<input type="checkbox"/> private-route1	rtb-06ccf156700c6ad0f	subnet-0fbf999bbb6f8bb...	-	No	vpc-0e00b3fbd7ee5bd7a 3-tie...	730335267904
<input type="checkbox"/> public-route	rtb-0030f3b55f8935a16	2 subnets	-	No	vpc-0e00b3fbd7ee5bd7a 3-tie...	730335267904

EC2:

- Go to Ec2 now you have to launch 2 instances manually
- Click on launch instance and give the naming and create key pair
- Edit network settings, select your VPC and subnet
- Enable Auto-assign public IP

- Create security group, creation of security group you have to add security groups i.e HTTP, MYSQL/Aurora
- Now click on launch instance

Instances (4) Info								
<div> <div>Find Instance by attribute or tag (case-sensitive)</div> <div>All states ▾</div> </div>			<div> <div>Last updated less than a minute ago</div> <div>Refresh</div> </div>		Connect	Instance state ▾	Actions ▾	Launch instances ▾
<input type="checkbox"/>	Name ↗ ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IP
<input type="checkbox"/>	Public2	i-Of08b72e80bb3a89c	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	-
<input type="checkbox"/>	Private2	i-0bd7ad8c0180b316a	Running 🔍 🔍	t2.micro	Initializing	View alarms +	ap-south-1b	-
<input type="checkbox"/>	Public1	i-0044204f63bb14641	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-
<input type="checkbox"/>	Private1	i-05104d70a312b46a1	Running 🔍 🔍	t2.micro	Initializing	View alarms +	ap-south-1a	-

- Now you to create launch Template
- Select your instance go with image and templates
- Click on create template from instance
- Give naming and create launch template

Launch Templates (2) Info						
<div> <div>Search</div> <div>1</div> </div>			Refresh	Actions ▾	Create launch template	
<input type="checkbox"/>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Creation ID
<input type="checkbox"/>	lt-08174cb4b63eb4b5f	Public-temp	1	1	2024-08-21T10:34:26.000Z	arn...
<input type="checkbox"/>	lt-0322c8eb830025089	Private-temp	1	1	2024-08-21T10:37:49.000Z	arn...

- Create Target group and click on it
- Give the name and choose your VPC nad click on next
- Select your instance ,click on create Target group

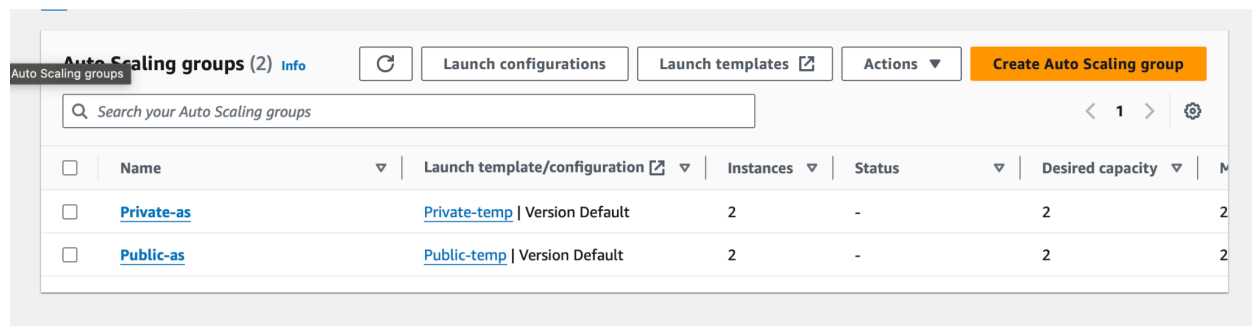
Load balancer:

Load balancer distribute incoming application traffic across multiple targets, such as Ec2 instance, in multiple availability Zones

- Now create load balancer and go with Application load balancer
- Give name and choose your VPC, In mappings select public subnets
- select your security group and also select Target group
- click on create load balancer

- Now go to Ec2 service and select the Auto Scaling group.
- Click on Create Auto Scaling group.
- Give the name tag Auto Scaling group name to the public.
- Select your public launch template and click on next
- In network settings select your VPC and select two public subnets click on next
- In configure group size and scaling enter desired capacity, click on next
- click on create Auto scaling

Auto Scaling:



<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	
<input type="checkbox"/>	Private-as	Private-temp Version Default	2	-	2	2
<input type="checkbox"/>	Public-as	Public-temp Version Default	2	-	2	2

- Same as it is, you need launch another instance which is "private"
- Create another template "private" and attach to private instance
- Create another Target group "private"
- Create another load balancer "private"
- Create another Auto-scaling
- Attach private subnets 2

RDS:

RDS stands for Relational database service and it is an easy-to-manage relational database service optimized for total cost of ownership.

RDS DB subnet group is a collection of subnets that are associated with an Elastic Compute Cloud(Ec2) and Virtual private cloud(VPC).

- Now go to search bar and search RDS
- Now you need to create subnet group

- An RDS Subnet Group is a collection of subnets that you can use to designate for your RDS database instance in a VPC.
- Click on create subnet group and enter the name, choose your VPC
- Select your Availability Zones
- Add subnets using private subnets and select private subnets
- Click on create
- Go to database and click on create database
- Select a database creation method is "standard create"
- Select MYSQL engine type and select free tier template
- Enter name for your DB cluster and enter master username
- Select credentials management is "self managed" and enter your own password
- In connectivity select your VPC and security group and click on create database.
- Go to ec2 instance click on public instance and connect it
- Now in public instance you have to connect private instance

Output:

```
5 rows in set (0.00 sec)

mysql> exit
Bye
root@ip-101-0-6-176:~# ls
Project.pem  snap
root@ip-101-0-6-176:~# rm Project.pem
root@ip-101-0-6-176:~# vi Project.pem
root@ip-101-0-6-176:~# chmod 400 Project.pem
root@ip-101-0-6-176:~# ssh -i "Project.pem" ubuntu@ec2-43-205-240-251.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-43-205-240-251.ap-south-1.compute.amazonaws.com (101.0.130.74)' can't be established.
ED25519 key fingerprint is SHA256:eyUbxjT/L6Vhqs1z0dg/DOlTTo16quQPGpH+vuxvPmg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-43-205-240-251.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Aug 21 11:31:54 UTC 2024

System load:  0.0           Processes:    105
Usage of /:   22.7% of 6.71GB Users logged in: 0
Memory usage: 19%          IPv4 address for enX0: 101.0.130.74
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.
```

```
System information as of Wed Aug 21 11:31:54 UTC 2024

System load:  0.0                Processes:           105
Usage of /:   22.7% of 6.71GB    Users logged in:    0
Memory usage: 19%               IPv4 address for enX0: 101.0.130.74
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- You have install MYSQL
- First you need update for that enter "apt update -y"
- To install MYSQL enter the command is "apt install mysql-server"
- To start the MYSQL enter the command is "systemctl start mysql.service"
- To connect database enter the command is mysql -h <paste database endpoint> -u <master username> -p
- Enter your password
- So finally SQL server is connected

```

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> show databases
-> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show databases' at line 2
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database meghs;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| meghs |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>

```

conclusion: 3-tier architecture provides a robust and scalable approach to building modern applications. By dividing an application into presentation, application, and data layers, we successfully designed and implemented a scalable, secure, and efficient 3-tier architecture in Amazon Web Services for web applications.