

Aws Two Tier using Load balancers

1. Project Title:

Two tier architecture we make communication between Web server to App Server by using load balancers.

2. Name of the Person:

Vamshi Mamidi

3. Objective of the Project:

The objective of this project is to make the communication between Web server to App Server by using load balancers.

4. Steps Done in the Project:

- Create the VPC Eg: “10.0.0.0/16” and then create Two Subnets one is Public and Another is Private.
- And then Create Internet Gateway and Attach to VPC.
- And then Edit the Default Route table to as Public Route table, then add internet gateway to the route table.
- Then all traffic goes via internet gateway to access the internet.
- Now Edit the Security Group inbound rules of web server.
- Create the VPC.

vpc-0ce0cfffe918a95c3a / 2-Tier

Actions ▾

Details		Info	
VPC ID	vpc-0ce0cfffe918a95c3a	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-04d14ecc3b802a1a1	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
Encryption control ID	-	Encryption control mode	-
		Block Public Access	Off
		DHCP option set	dopt-0c42105c942ae5ea4
		IPv4 CIDR	10.0.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-07d8d1e7df8386655 / RT
		IPv6 pool	-
		Owner ID	626747814259

- Create the Public and Private Subnets.

Subnets (2) Info

Last updated about 3 hours ago

Actions ▾ Create subnet

Name	Subnet ID	State	VPC	Block Public...
Private-Subnet	subnet-05c1f08ad61fc1278	Available	vpc-0ce0cfffe918a95c3a 2-Tier	Off
Public-Subnet	subnet-06b1859fe05e47419	Available	vpc-0ce0cfffe918a95c3a 2-Tier	Off

- Create Internet Gateway and Attach to VPC

Internet gateways (1) Info					
<input type="text"/> Find internet gateways by attribute or tag				Actions ▼ Create internet gateway	
<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	
<input type="checkbox"/>	IGW	igw-0ecf6bef4b41cccd6d	Attached	vpc-0ce0cffef18a95c3a 2-Tier	6:

- Create the Route Tables and Configure internet gateway.

Route tables (1) Info					
<input type="text"/> Find route tables by attribute or tag				Actions ▼ Create route table	
<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main
<input type="checkbox"/>	RT	rtb-07d8d1e7df8386655	-	-	Yes vpc-0ce0cffef18a95c3a

- Create the Security Group and write the Rules.

Edit inbound rules Info					
Security group rule ID	Type Info	Protocol Info	Port range	Source Info	Description - optional Info
sgr-0a88f4db6dd3d1b8a	Custom TCP ▼	TCP ▼	8080	Cust... ▼	<input type="text"/> Delete
				0.0.0.0/0 X	
sgr-09603657ea8fb6810	All traffic ▼	All ▼	All	Cust... ▼	<input type="text"/> Delete
				0.0.0.0/0 X	
sgr-02235b22e57766a43	SSH ▼	TCP ▼	22	Cust... ▼	<input type="text"/> Delete
				0.0.0.0/0 X	
Add rule					

- Create the Web Server and App server EC2 machine and login into the machine from mobaXterm.

Instances (2) Info					
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		Actions ▼ Launch instances		All states ▼	
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check
<input type="checkbox"/>	Web-Server	i-0c16866e0938e1e0c	Running Q Q	t3.micro	3/3 checks passed View alarms +
<input type="checkbox"/>	App-Server	i-036300ed323f4b5c1	Running Q Q	t3.micro	3/3 checks passed View alarms +

- Now Create the Two Network Load balancer one is Public load balancer using internet-facing and other is the Private load balancer using internal and allow the port 8080 while creating private LB.
- While creating the load balancers the target groups also created and for private target group allow 8080 port.

Load balancers (2) What's new?																											
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.																											
<input type="button" value="Actions"/> Create load balancer																											
<input type="text" value="Filter load balancers"/>																											
<table border="1"> <thead> <tr> <th>Name</th><th>State</th><th>Type</th><th>Scheme</th><th>IP address type</th><th>VPC ID</th><th></th></tr> </thead> <tbody> <tr> <td>Public-LB</td><td>Active</td><td>network</td><td>Internet-facing</td><td>IPv4</td><td>vpc-0ce0cffea918a95c3a</td><td>Edit</td></tr> <tr> <td>private</td><td>Active</td><td>network</td><td>Internal</td><td>IPv4</td><td>vpc-0ce0cffea918a95c3a</td><td>Edit</td></tr> </tbody> </table>							Name	State	Type	Scheme	IP address type	VPC ID		Public-LB	Active	network	Internet-facing	IPv4	vpc-0ce0cffea918a95c3a	Edit	private	Active	network	Internal	IPv4	vpc-0ce0cffea918a95c3a	Edit
Name	State	Type	Scheme	IP address type	VPC ID																						
Public-LB	Active	network	Internet-facing	IPv4	vpc-0ce0cffea918a95c3a	Edit																					
private	Active	network	Internal	IPv4	vpc-0ce0cffea918a95c3a	Edit																					

Target groups (2) Info What's new?																											
<input type="button" value="Actions"/> Create target group																											
<input type="text" value="Filter target groups"/>																											
<table border="1"> <thead> <tr> <th>Name</th><th>ARN</th><th>Port</th><th>Protocol</th><th>Target type</th><th>Load balancer</th><th></th></tr> </thead> <tbody> <tr> <td>TG2</td><td>arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:1</td><td>80</td><td>TCP</td><td>Instance</td><td>None associated</td><td>Edit</td></tr> <tr> <td>TG1</td><td>arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:2</td><td>80</td><td>TCP</td><td>Instance</td><td>Public-LB</td><td>Edit</td></tr> </tbody> </table>							Name	ARN	Port	Protocol	Target type	Load balancer		TG2	arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:1	80	TCP	Instance	None associated	Edit	TG1	arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:2	80	TCP	Instance	Public-LB	Edit
Name	ARN	Port	Protocol	Target type	Load balancer																						
TG2	arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:1	80	TCP	Instance	None associated	Edit																					
TG1	arn:aws:elasticloadbalancing:us-east-1:45972fc6f2593d0d:elb-ap-south-1:2	80	TCP	Instance	Public-LB	Edit																					

- After login into the MobaXterm with public IP of Web Server EC2 machine then follow the commands.
- Sudo su
- apt update
- apt install nginx -y
- check it in browser with public IP.



- Then go to the public instance of web server and add key in the instance like "key.pem"
- Add permission to the key chmod 400 key.pem (it says that owner can only read the key)
- Then login the private instance using ssh command: ssh -i key.pem ubuntu@<pvtip>
- Then it was login into the App server then follow the commands
- sudo su
- apt update
- Then install tomcat on the app server
- apt install default-jdk -y
- wget <https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.18/bin/apache-tomcat-11.0.18.tar.gz> (Check latest)
- tar -xvf apache-tomcat-11.0.15.tar.gz (unzip the folder)
- mv apache-tomcat-11.0.15 tomcat (after unzip move the folder by creating tomcat folder)
- cd tomcat/bin

- ./startup.sh
- After giving Above commands the Tomcat is started.
- Then check the connection web server to app server.

```
apache-tomcat-10.1.52.tar.gz vamshi-tomcat
root@ip-10-0-1-91:/home/ubuntu# cd vamshi-tomcat/
root@ip-10-0-1-91:/home/ubuntu/vamshi-tomcat# ls
BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work
root@ip-10-0-1-91:/home/ubuntu/vamshi-tomcat# cd bin/
root@ip-10-0-1-91:/home/ubuntu/vamshi-tomcat/bin# ls
bootstrap.jar ciphers.bat configtest.bat digest.sh migrate.sh shutdown.sh tomcat-native.tar.gz version.sh
catalina-tasks.xml ciphers.sh configtest.sh makebase.bat setclasspath.bat startup.bat tool-wrapper.bat
catalina.bat commons-daemon-native.tar.gz daemon.sh makebase.sh setclasspath.sh startup.sh tool-wrapper.sh
catalina.sh commons-daemon.jar digest.bat migrate.bat shutdown.bat tomcat-juli.jar version.bat
root@ip-10-0-1-91:/home/ubuntu/vamshi-tomcat/bin# ./startup.sh
Using CATALINA_BASE: /home/ubuntu/vamshi-tomcat
Using CATALINA_HOME: /home/ubuntu/vamshi-tomcat
Using CATALINA_TMPDIR: /home/ubuntu/vamshi-tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/ubuntu/vamshi-tomcat/bin/bootstrap.jar:/home/ubuntu/vamshi-tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@ip-10-0-1-91:/home/ubuntu/vamshi-tomcat/bin# ls
```

- Check the connection from Web Server to App Server by using private ip of App Server and as well as private Load balancer DNS name.
- It shows as connected

```
root@ip-10-0-0-100:/home/ubuntu# telnet 10.0.1.91 8080
Trying 10.0.1.91...
Connected to 10.0.1.91.
Escape character is '^>'.
Connection closed by foreign host.
root@ip-10-0-0-100:/home/ubuntu# telnet private-959d1c4cffc42f52.elb.ap-south-1.amazonaws.com 8080
Trying 10.0.1.146...
Connected to private-959d1c4cffc42f52.elb.ap-south-1.amazonaws.com.
Escape character is '^>'.
Connection closed by foreign host.
root@ip-10-0-0-100:/home/ubuntu# telnet private-959d1c4cffc42f52.elb.ap-south-1.amazonaws.com 8080
Trying 10.0.1.146...
Connected to private-959d1c4cffc42f52.elb.ap-south-1.amazonaws.com.
Escape character is '^>'.
```

7. Conclusion:

The two tier architecture we make communication between Web server to App Server by using load balancers has successfully executed.