

Lab: Final Design Implementation

Submitted by : Junie Mariam Varghese

Github link: <https://github.com/juniemariam/terraform-ecommerce-aws-hosting>

1. Terraform Plan

```
rohan@junie:~/Final % terraform plan
...
aws_sns_topic_scaling_notifications: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications]
aws_iam_role_backend_ec2_role: Refreshing state... [id=backend-ec2-role]
aws_vpc.ecommerce_vpc: Refreshing state... [id=vpc-06e352ee7d7b56c3f]
aws_iam_policy.secrets_access_policy: Refreshing state... [id=arn:aws:iam::640168430849:policy/secrets-access-policy]
aws sns topic subscription_email_subscription: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications:031d82e1-095d-4778-98ca-7e2a27f5b086]
aws iam role_policy_attachment.backend_ec2_role_attachment: Refreshing state... [id=backend-ec2-role-20241203051427890800000001]
aws iam instance_profile.backend_instance_profile: Refreshing state... [id=backend-instance-profile]
aws internet_gateway.ecommerce_igw: Refreshing state... [id=igw-017354e407aaea4eb]
aws subnet.public_subnet: Refreshing state... [id=subnet-02897fc1ef8e080db]
aws route_table.public_rt: Refreshing state... [id=rtb-07897d494885fa05f1]
aws subnet.private_subnet: Refreshing state... [id=subnet-0feb761d0f2578473]
aws subnet.public_subnet_3: Refreshing state... [id=subnet-0b31e0a6cccc1846dd]
aws security_group.rds_sg: Refreshing state... [id=sg-06f3bc816c6af7518]
aws security_group.backend_sg: Refreshing state... [id=sg-0410d7e375aada8e]
aws_lb_target_group.backend_tg: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:targetgroup/backend-tg/61a445cd4e6827a3]
aws subnet.private_subnet_3: Refreshing state... [id=subnet-0fc03558753d97af]
aws route.default_route: Refreshing state... [id=rtrt-07897d494885fa05f11080289494]
aws security_group_rule.rds_backend_ingress: Refreshing state... [id=sgrule-887591629]
aws_launch_template.backend_lt: Refreshing state... [id=lt-0a867357a7a8c77d4]
aws_db_subnet_group.ecommerce_db_subnet_group: Refreshing state... [id=ecommerce-db-subnet-group]
aws_route_table_association.public_subnet_assoc[0]: Refreshing state... [id=rtbassoc-003eef83eccb03a6f]
aws_route_table_association.private_subnet_assoc[1]: Refreshing state... [id=rtbassoc-00a14e691be7cc5f0]
aws_lb.backend_lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:loadbalancer/app/backend-lb/6a91b983516f84b7]
aws_autoscaling_group.backend_asg: Refreshing state... [id=auto-terraform-202412030413373245000000004]
aws_db_instance.ecommerce_rds: Refreshing state... [id=db-VTHIRWANAWL6AJC02EXNNW6A]
aws_lb_listener.backend_listener: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:listener/app/backend-lb/6a91b983516f84b7/64e6841f2c4bb914]
aws_autoscaling_notification.backend_asg_notifications: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
- update in-place

Terraform will perform the following actions:

# aws_autoscaling_group.backend_asg will be updated in-place
- resource "aws_autoscaling_group" "backend_asg" {
  - desired_capacity          = 1 -> 2
  - id                         = "terraform-202412030413373245000000004"
  - min_size                   = 1 -> 2
  - name                       = "terraform-202412030413373245000000004"
  - termination_policies      = [
    - "OldestInstance",
  ]
  # (28 unchanged attributes hidden)
}

# aws_launch_template.backend_lt will be updated in-place
- resource "aws_launch_template" "backend_lt" {
  - id                         = "lt-0a867357a7a8c77d4"
  - latest_version              = 3 -> (known after apply)
}
```

```
        },
        # (1 unchanged element hidden)
    ]
    name          = "rds-sg"
    tags          = {
        "Name" = "rds-sg"
    }
    # (8 unchanged attributes hidden)
}

# aws_security_group_rule.rds_backend_ingress will be created
resource "aws_security_group_rule" "rds_backend_ingress" {
    + from_port      = 3386
    + id             = (Known after apply)
    + protocol       = "tcp"
    + security_group_id = "sg-00f3bc816c6af7518"
    + security_group_rule_id = (Known after apply)
    + self           = false
    + source_security_group_id = "sg-0410d7e375eada8e"
    + to_port        = 3386
    + type           = "ingress"
}

Plan: 1 to add, 3 to change, 0 to destroy.



---



Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.


```

2. Terraform apply

```
rohan@junie Final % terraform apply
aws_vpc.ecommerce_vpc: Refreshing state... [id=vpc-06a352ee7d7b56c3f]
aws_iam_role.backend_ec2_role: Refreshing state... [id=backend-ec2-role]
aws_iam_policy.secrets_access_policy: Refreshing state... [id=arn:aws:iam::640168430849:policy/secrets-access-policy]
aws sns topic.scaling_notifications: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications]
aws sns topic.subscription.email_subscription: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications:031d82e1-095d-4778-98ca-7e2a27f3b086]
aws iam role_policy_attachment.backend_ec2_role_attachment: Refreshing state... [id=backend-ec2-role-20241203051427890880000001]
aws iam instance_profile.backend_instance_profile: Refreshing state... [id=backend-instance-profile]
aws internet_gateway.ecommerce_igw: Refreshing state... [id=igw-017354e4a7aae3eb]
aws subnet.private_subnet_3: Refreshing state... [id=subnet-01c03558753d67af5]
aws security_group.rds_sg: Refreshing state... [id=sg-00f30c81c16c6af7518]
aws subnet.public_subnet_3: Refreshing state... [id=subnet-0b31e0a4cce184dd]
aws subnet.private_subnet: Refreshing state... [id=subnet-0fe6751d0ff2578473]
aws route_table.public_rt: Refreshing state... [id=rtb-07897d494985f4fb5f1]
aws security_group.backend_sg: Refreshing state... [id=sg-0418d7e375aae0d8e]
aws_lb_target_group.backend_tg: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:targetgroup/backend-tg/61a445cd4e6827a3]
aws subnet.public_subnet: Refreshing state... [id=subnet-028907fc1ef8e800]
aws route.default_route: Refreshing state... [id=r-rtb-07897d494985f4fb5f11880289494]
aws security_group_rule.rds_backend_ingress: Refreshing state... [id=sgrule-887591629]
aws_db_subnet_group.ecommerce_db_subnet_group: Refreshing state... [id=ecommerce-db-subnet-group]
aws_route_table_association.public_subnet_assoc[1]: Refreshing state... [id=rtbassoc-08a14e691be7cc5f8]
aws_route_table_association.public_subnet_assoc[0]: Refreshing state... [id=rtbassoc-003eaf83ecb03a6f]
aws_lb.backend_lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:loadbalancer/app/backend-lb/6a91b983516f84b7]
aws_launch_template.backend_lt: Refreshing state... [id=lt-0a0673a7a8c77d4]
aws_launch_template.backend_lt: Refreshing state... [id=lt-0a0673a7a8c77d4]
aws_db_instance.ecommerce_rds: Refreshing state... [id=d-0-VTHIRWANAWL6A.JC02EXNNW6A]
aws_lb_listener.backend_listener: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:640168430849:listener/app/backend-lb/6a91b983516f84b7/64e6841f2c4bb914]
aws_autoscaling_notification.backend_asg_notifications: Refreshing state... [id=arn:aws:sns:us-west-1:640168430849:scaling-notifications]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
~ update in-place

Terraform will perform the following actions:

# aws_autoscaling_group.backend_asg will be updated in-place
~ resource "aws_autoscaling_group" "backend_asg" {
    ~ desired_capacity           = 1 -> 2
    ~ id                          = "terraform-202412030413373245000000004"
    ~ min_size                    = 1 -> 2
    ~ name                        = "terraform-202412030413373245000000004"
    ~ termination_policies       = [
        ~ "OldestInstance",
    ]
    # (28 unchanged attributes hidden)
    ~ # (3 unchanged blocks hidden)
    ~
}

# aws_launch_template.backend_lt will be updated in-place
~ resource "aws_launch_template" "backend_lt" {
```



```

        }
        - self          = false
        - to_port       = 3306
        # (1 unchanged attribute hidden)
    },
    # (1 unchanged element hidden)
]
name           = "rds-sg"
tags           = [
    "Name" = "rds-sg"
]
# (8 unchanged attributes hidden)
}

# aws_security_group_rule.rds_backend_ingress will be created
resource "aws_security_group_rule" "rds_backend_ingress" {
    + from_port      = 3306
    + id             = (known after apply)
    + protocol       = "tcp"
    + security_group_id = "sg-00f3bc816c6af7518"
    + security_group_rule_id = (known after apply)
    + self           = false
    + source_security_group_id = "sg-0410d7e375aeada8e"
    + to_port         = 3306
    + type            = "ingress"
}

Plan: 1 to add, 3 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_security_group.rds_sg: Modifying... [id=sg-00f3bc816c6af7518]
aws_launch_template.backend_lt: Modifying... [id=lt-0a067367a7a8c77d4]
aws_security_group.rds_sg: Modifications complete after 1s [id=sg-00f3bc816c6af7518]
aws_security_group_rule.rds_backend_ingress: Creating...
aws_security_group_rule.rds_backend_ingress: Creation complete after 0s [id=rdsrule-887591629]
aws_launch_template.backend_lt: Modifications complete after 0s [id=lt-0a067367a7a8c77d4]
aws_autoscaling_group.backend_asg: Modifying... [id=terraform-20241203041337324500000004]
aws_autoscaling_group.backend_asg: Still modifying... [id=terraform-20241203041337324500000004, 10s elapsed]
aws_autoscaling_group.backend_asg: Modifications complete after 15s [id=terraform-20241203041337324500000004]

① Apply complete! Resources: 1 added, 3 changed, 0 destroyed.
99 rohan@junie Final %

```

AWS Resources:

1. EC2

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 MAC	Elastic IP	IPv6 IPs
backend-asg	i-0da122e5604f474cf	Running	t2.micro	2/2 checks passed	View alarms	us-west-1a	ec2-52-53-200-184.us.w...	52.53.200.184	-	-
backend-asg	i-0fd0146284ef8211	Running	t2.micro	2/2 checks passed	View alarms	us-west-1a	ec2-54-177-248-99.us.w...	54.177.248.99	-	-
backend-asg	i-0cf2f2872d357a2b1	Running	t2.micro	2/2 checks passed	View alarms	us-west-1a	ec2-13-56-247-118.us.w...	13.56.247.118	-	-

2. ASG

Screenshot of the AWS EC2 Auto Scaling Groups console showing the configuration for an Auto Scaling group named "terraform-20241203041337324500000004".

Capacity overview:

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	2 - 5	Units (number of instances)	-

Launch template:

AMI ID	Instance type	Owner
ami-0d53d72369335ad96	t2.micro	arnawsiam:540168430849:user/JunieMV

Network:

Availability Zones	Subnet ID	Availability Zone distribution
us-west-1a	subnet-028907c1ef8e80db	Balanced best effort

Instance type requirements:

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Health checks:

Load balancing and VPC lattice options have moved to the new integrations tab.

3. VPC

Screenshot of the AWS VPC dashboard showing the configuration for a VPC named "vpc-06a352ee7d7b56c3f / ecommerce-vpc".

Details:

VPC ID: vpc-06a352ee7d7b56c3f	State: Available	Block Public Access: Off	DNS hostnames: Enabled
DNS resolution: Enabled	Tenancy: Default	DHCP option set: dopt-024a2088ed99c98e1	Main route table: rtb-0a65950baa51af60b
Main network ACL: ad-09d2c0660ba91a54f	Default VPC: No	IPv4 CIDR: 10.0.0.0/16	IPv6 pool: -
IPv6 CIDR: -	Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: -	Owner ID: 640168430849

Resource map:

```

graph LR
    VPC[VPC Show details] --- Subnets[Subnets (4)]
    VPC --- RouteTables[Route tables (2)]
    VPC --- NetworkConnections[Network connections (1)]
    
    Subnets --- uswest1a[us-west-1a]
    Subnets --- uswest1c[us-west-1c]
    uswest1a --- publicsubnet1[public-subnet]
    uswest1a --- privatesubnet1[private-subnet]
    uswest1c --- publicsubnet2[public-subnet-3]
    uswest1c --- privatesubnet2[private-subnet-3]
    
    RouteTables --- rtb1[rtb-0a65950baa51af60b]
    RouteTables --- rtb2[rtb-0a65950baa51af60b]
    
    NetworkConnections --- igw[ecommerce-igw]
  
```

The diagram illustrates the VPC structure with four subnets (two in us-west-1a and two in us-west-1c) connected to two route tables (rtb-0a65950baa51af60b) and one internet gateway (ecommerce-igw).

4. Subnets

Subnets (4) Info											Last updated 1 minute ago	Actions	Create subnet
	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR		IPv6 CIDR		IPv6 CIDR association ID	Available		
<input type="checkbox"/>	public-subnet	subnet-028907fc1ef8e80db	Available	vpc-06a352ee7d7b56c3f eco...	<input type="radio"/> Off	10.0.1.0/24		-		-	248		
<input type="checkbox"/>	public-subnet-3	subnet-0b31e0a6ceef1846dd	Available	vpc-06a352ee7d7b56c3f eco...	<input type="radio"/> Off	10.0.6.0/24		-		-	249		
<input type="checkbox"/>	private-subnet	subnet-0feb761d0f2578473	Available	vpc-06a352ee7d7b56c3f eco...	<input type="radio"/> Off	10.0.2.0/24		-		-	251		
<input type="checkbox"/>	private-subnet-3	subnet-01cb3558753d87af5	Available	vpc-06a352ee7d7b56c3f eco...	<input type="radio"/> Off	10.0.5.0/24		-		-	251		

5. IGW

Details		Actions	
Internet gateway ID	<input type="checkbox"/> igw-017354e4a7aaea3eb	State	Attached
VPC ID	vpc-06a352ee2d7b56c3f ecommerce-vpc		
Owner	<input type="checkbox"/> 640168430849		
Tags		Manage tags	
<input type="text"/> Search tags		< 1 > ?	
Key	Value		
Name	ecommerce-igw		

6. RDS

ecommerce-db-instance

Summary

DB identifier ecommerce-db-instance	Status Available	Role Instance	Engine MySQL Community
CPU 	Class db.t3.micro	Current activity 	Region & AZ us-west-1c

Connectivity & security

Endpoint & port

Endpoint ecommerce-db-instance.cpewg24csbd6.us-west-1.rds.amazonaws.com	Networking	Security
Port 3306	Availability Zone us-west-1c	VPC security groups rds-sg (sg-00f3bc16c6af7518) Active
	VPC ecommerce-vpc (vpc-06a352ee7d7b56cf)	Publicly accessible No
	Subnet group ecommerce-db-subnet-group	Certificate authority rds-ca-rsa2048-g1
	Subnets subnet-028907fc1ef8e80db subnet-0b31e0a6cce1846dd	Certificate authority date May 19, 2061, 13:04 (UTC-07:00)
	Network type IPv4	DB instance certificate expiration date December 02, 2025, 20:15 (UTC-08:00)

Connected compute resources (0)

Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.

Filter by compute resources

Resource identifier	Resource type	Availability Zone	VPC security group	Compute resource security group	Connected proxy
No connected compute resources					

No connected compute resources that were created automatically to display.

7. SG

sg-0d0e9c62ce8e86879 - default

Inbound rules (3)

Name	Security group rule ID	IP version	Type	Protocol	Port
-	sgr-0dabe8e77867cf50	-	All traffic	All	All
-	sgr-08478a84972c10626	IPv4	SSH	TCP	22
-	sgr-0fb85af8198017f09	IPv4	Custom TCP	TCP	800

8. SNS

scaling-notifications

Subscriptions (2)

ID	Endpoint	Status	Protocol
775d9f71-086b-4d00-88b1-29b8f88381a0	junie.mariamvarghese@sjsu.edu	Confirmed	EMAIL
Pending confirmation	your-email@example.com	Pending confirmation	EMAIL

Auto Scaling: launch for group "terraform-20241203041337324500000004"

Service: AWS Auto Scaling
Time: 2024-12-04T07:37:28Z
RequestID: 4b564e42-6878-beef-7811-6ea396ec7404
Event: autoscaling:EC2_INSTANCE_LAUNCH
Account: 12345678901234567890
AutoScalingGroupName: terraform-20241203041337324500000004
AutoScalingGroupARN: arn:aws:autoscaling:us-west-1:640168430849:autoScalingGroup:90a47e8f-5e9e-4590-8310-20d9b707785:autoScalingGroupName:terraform-20241203041337324500000004
ActivityID: 4b564e42-6878-beef-7811-6ea396ec74b4
Description: Launching a new EC2 instance: i-0cd22f72d357a281
Cause: AI 2024-12-04T07:37:37.052 A user request update of AutoScalingGroup constraints to min: 2, max: 5, desired: 2 changing the desired capacity from 1 to 2. At 2024-12-04T07:37:18Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity by 1.
StartTime: 2024-12-04T07:37:19.651Z
EndTime: 2024-12-04T07:37:25.837Z
StatusCode: InProgress
StatusMessage:
ProjectID: 0
EC2InstanceID: i-0cd22f72d357a281
Details: {"Subnet ID": "subnet-029907fc1feff80db", "Availability Zone": "us-west-1a"}
Origin: EC2
Destination: AutoScalingGroup

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-west-1.amazonaws.com/v2/SubscriptionArn=arn:aws:sns:us-west-1:640168430849:scaling-notifications:775d9f71-086b-4d00-88b1-29b8f88381a0&Endpoint=junie.mariamvarghese@sjsu.edu>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

9. IAM

The screenshot shows the AWS IAM Role details page for 'backend-ec2-role'. The left sidebar includes sections for Identity and Access Management (IAM), Access management, Access reports, and Related consoles. The main content area displays the role's summary, ARN, instance profile ARN, and permissions policies. It also features a 'Generate policy based on CloudTrail events' section and a 'Listeners and rules' tab.

Summary	ARN	Instance profile ARN
Creation date: December 02, 2024, 21:14 (UTC-08:00)	arn:aws:iam::640168430849:role/backend-ec2-role	arn:aws:iam::640168430849:instance-profile/backend-instance-profile
Last activity: 1 hour ago	Maximum session duration: 1 hour	

Permissions policies (2)

Policy name	Type	Attached entities
secrets-access-policy	Customer managed	1
SecretsManagerReadWrite	AWS managed	1

Permissions boundary (not set)

Generate policy based on CloudTrail events

No requests to generate a policy in the past 7 days.

10. Load Balancer

The screenshot shows the AWS Load Balancer Details page for 'backend-lb-2'. The left sidebar includes sections for Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the load balancer's details, listeners, and rules. It also features a 'Listeners and rules' tab.

Status	VPC	Load balancer IP address type
Active	vpc-06a352ee7d7b56c3f	IPv4
Hosted zone: Z368ELLURRE2KJ0	Availability Zones: subnet-03390f1c1f8e80db [us-west-1a (usw1-az1)], subnet-0b31e0a6ce1846dd [us-west-1c (usw1-az3)]	Date created: December 3, 2024, 23:37 (UTC-08:00)

Listeners and rules (1)

Protocol:Port	Forward to target group	ARN	Security policy	Default SSL/TLS certificate
HTTP:80	backend-lb-2 [1 (100%)] • Target group stickiness: Off	1 rule	Not applicable	Not applicable

11. Target group

Screenshot of the AWS EC2 Target groups console showing a single target group named "backend-tg-2".

Target groups (1/1) [info]

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
backend-tg-2	arn:aws:elasticloadbalancing:us-west-2:7d7b56c3f	8000	HTTP	Instance	backend-lb-2	vpc-06a352ee7d7b56c3f

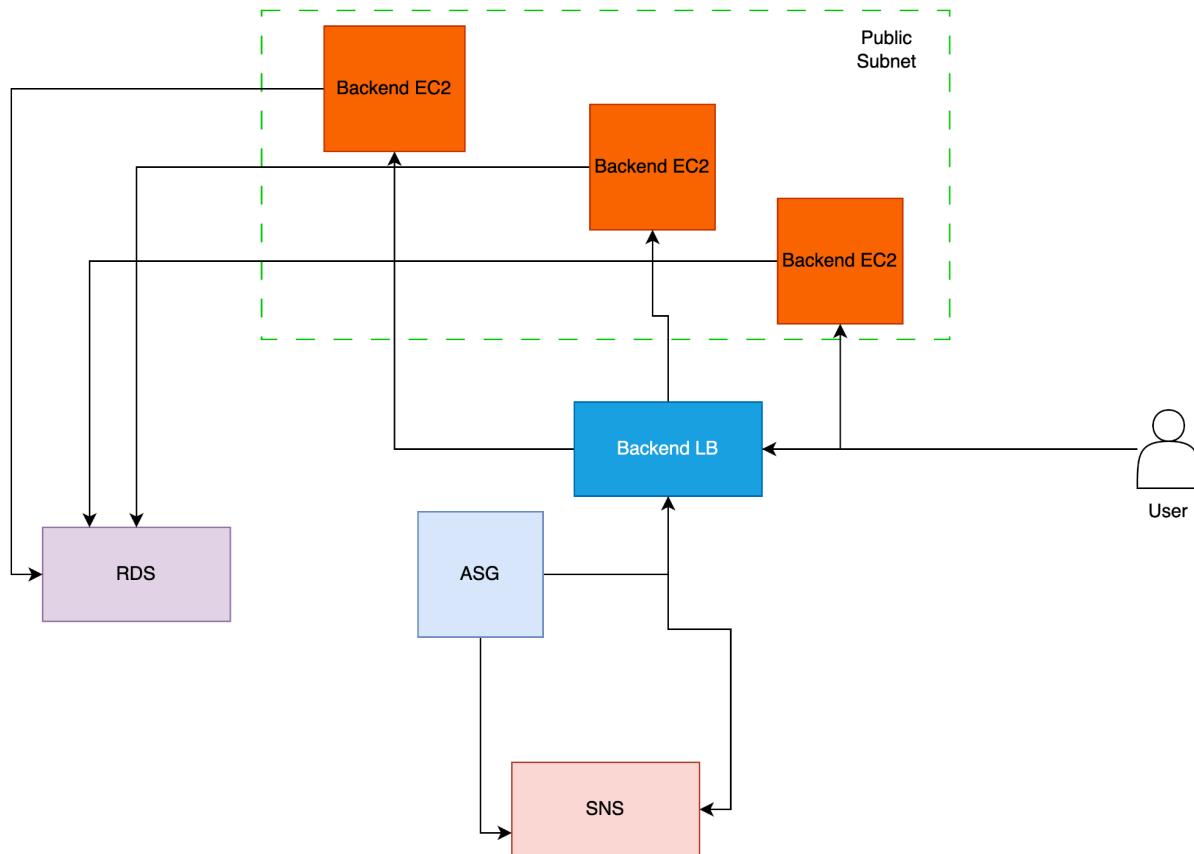
Target group: backend-tg-2

Registered targets (3) [info]

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative...	Override details	Launch...	Anomaly
i-0c02f2872d557a2b1	backend-asg	8000	us-west-1a (usw1-az1)	Draining	Target deregistration l...	<input type="radio"/> No override	No override is currently active on target	December...	Normal
i-0fdad0146284ef8211	backend-asg	8000	us-west-1a (usw1-az1)	Draining	Target deregistration l...	<input type="radio"/> No override	No override is currently active on target	December...	Normal
i-0da122e5604f474cf	backend-asg	8000	us-west-1a (usw1-az1)	Healthy	-	<input type="radio"/> No override	No override is currently active on target	December...	Normal

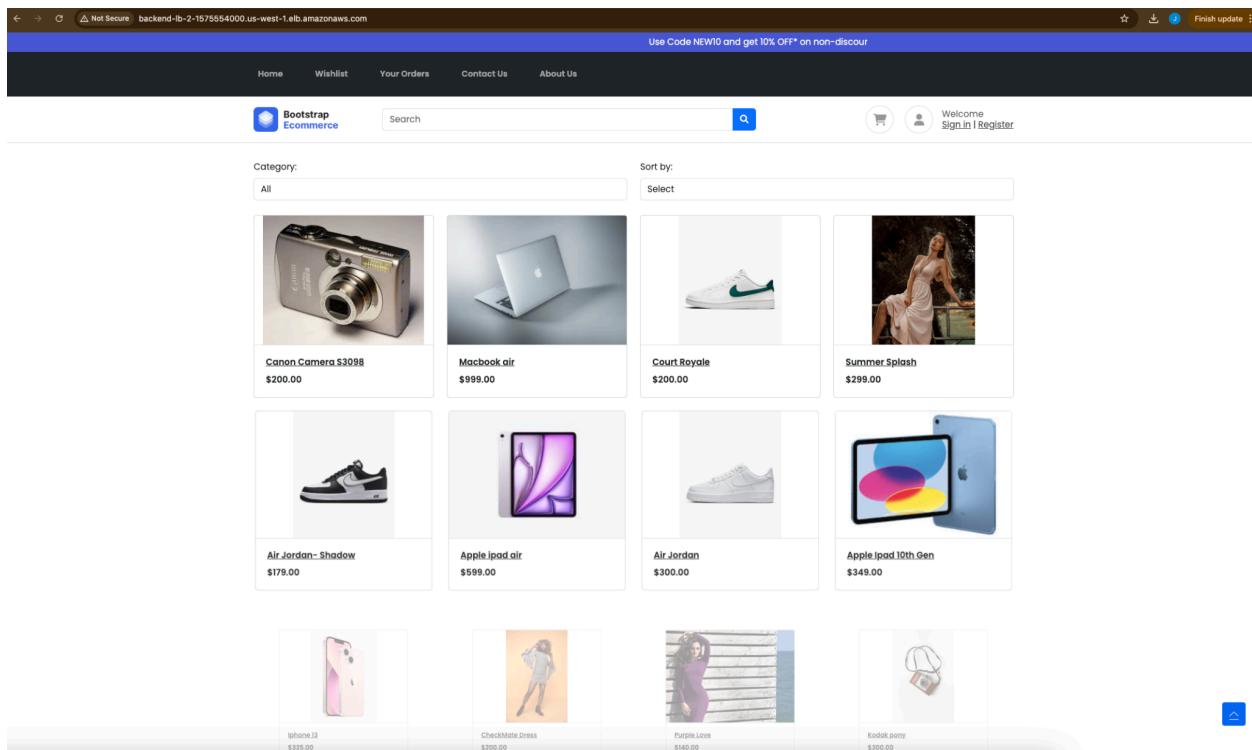
Deployment Architecture- Modified



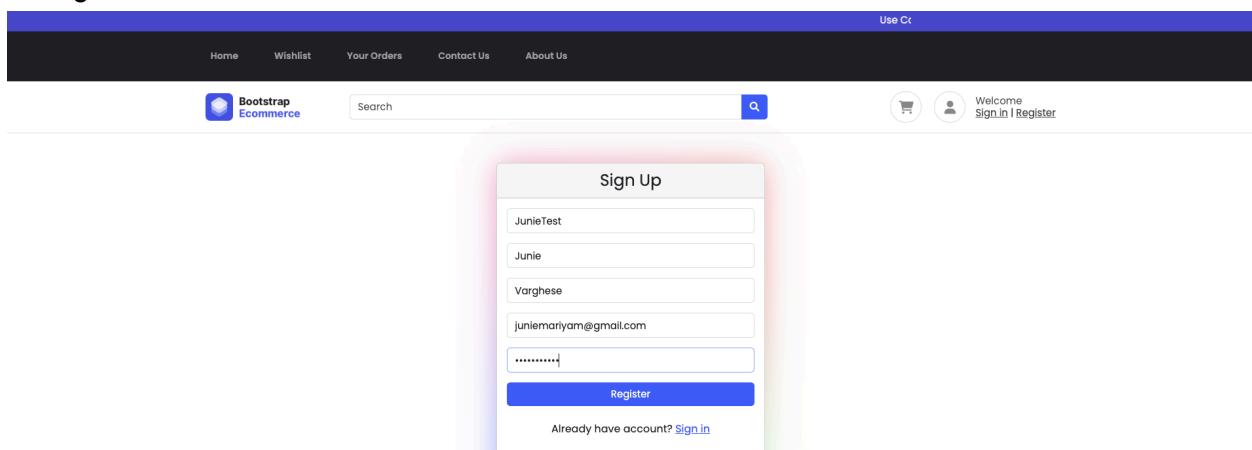
Ecommerce WebApp

Github Link: <https://github.com/juniemariam/ECommerceWebApp>

1. Application Welcome Page



2. Register



3. Login

The screenshot shows a web browser displaying a login page for 'Bootstrap Ecommerce'. The URL in the address bar is 'Not Secure backend-lb-2-1575554000.us-west-1.elb.amazonaws.com/accounts/login/?next=/accounts/logout/'. The page has a dark header with navigation links: Home, Wishlist, Your Orders, Contact Us, and About Us. A promotional banner at the top right says 'Use Code NEW10 and get ...' and 'Finish update'. Below the header is a search bar and a sign-in form. The sign-in form has fields for 'Email' (containing 'JunieTest') and 'Password' (containing '*****'). There are 'Remember' and 'Forgot password?' checkboxes, and a 'Login' button. Below the form is a link to 'Sign up'. The main content area features the 'Bootstrap Ecommerce' logo, a newsletter sign-up section with a 'Subscribe' button, and four footer sections: 'Get in Touch With Us', 'Our Mobile App', 'Information', and 'Shop Departments'.

JunieTest

Remember

Forgot password?

Login

Don't have account? [Sign up](#)

Or

Subscribe to our Newsletter
Get all the latest information, Sales and Offers.

Email address here... [Subscribe](#)

Get in Touch With Us

Phone: +1 (900) 33 169 7720
Monday to Sunday:
24x7 Available
support@ecommerce.com

Our Mobile App

[Download on the App Store](#)

[Download on the Google Play](#)

Information

About Us
Contact Us
Delivery
Terms and Conditions
Privacy Policy

Shop Departments

Clothes & Shoes
Smartphones & Tablets
TV, Video & Speaker
Cameras, Photo & Video
Headphones

4. Login Successful

The screenshot shows a web browser window for 'backend-lb-2-1575554000.us-west-1.elb.amazonaws.com'. The page title is 'Not Secure'. A blue header bar displays a 'Use Code NEW10 and get 10% OFF* on non-discounted Shoes over \$150. Maximum Discount: \$70. T&C Apply.' message. The main navigation menu includes Home, Wishlist (0), Your Orders, Contact Us, and About Us. The top right corner shows a user profile for 'JunieTest!' and a Logout link. The search bar contains the placeholder 'Search' with a magnifying glass icon. Below the header is a success message box: 'Login Successfull.' with an 'x' button. The main content area features a grid of products. The first row includes: 'Canon Camera S3098' (\$200.00), 'Macbook air' (\$999.00), 'Court Royale' (\$200.00), and 'Summer.Splash' (\$299.00). The second row includes: 'Air Jordan- Shadow' (\$179.00), 'Apple Ipod air' (\$599.00), 'Air Jordan' (\$300.00), and 'Apple Ipod 10th Gen' (\$349.00). Below these rows are three more product cards partially visible: 'iPhone 12 Pro Max' (\$1099.00), 'Dress' (\$250.00), 'Summer.Splash' (\$299.00), and 'Handbag' (\$129.00). The footer of the page includes a 'Bootstrap Ecommerce' logo and a 'Finish update' button.

5. Product Detail Page

The screenshot shows a product detail page for the 'Nike SB Force 58 - Black' shoe. The URL is 'backend-lb-2-1575554000.us-west-1.elb.amazonaws.com/product/nike-sb-force-58-black/'. The top navigation bar and user interface elements are identical to the previous screenshot. The main product image shows a black and white low-top sneaker. To the right of the image, the product name 'Nike SB Force 58 - Black' is displayed, along with a 'Shoes' category, a 5-star rating, 0 reviews, and 154 orders. The price is listed as '\$150.00'. Below this, a 'Brand Nike Special Price- Black Friday Deal' is mentioned. The product details section shows 'Brand: Nike', 'Color: Black White', and 'Delivery: All over the World!'. Below these details are quantity and size selection dropdowns, with 'Quantity: 1' and 'Select size: Large' selected. There are also 'Add to Wishlist' and 'Add to cart' buttons. The 'Related products' section below the main product shows three other Nike shoes: 'Court Royale' (\$200.00), 'Air Jordan- Shadow' (\$179.00), and 'Air Jordan' (\$300.00). The 'Reviews' section at the bottom indicates 'No reviews yet...'. A feedback form for 'Stars*' is partially visible. The footer of the page includes a 'Summary' button.

6. Add the item into cart

The screenshot shows a product page for a Nike SB Force 58 - Black shoe. The product details include Size UK 10, Color: Black/Black/White/White, and Brand: Nike. The quantity is set to 1, and the price is \$150. A 'Remove' button is visible. To the right, there's a 'Have coupon?' section with a 'Coupon code' input and an 'Apply' button. The total price is listed as \$150. Below the cart summary, there are buttons for 'Continue Shopping' and 'Make Purchase'. A green banner at the bottom states 'Free Delivery within 1-2 weeks'.

The footer of the site includes a newsletter sign-up form with fields for 'Email address here..', a 'Subscribe' button, and links for 'Get all the latest information, Sales and Offers.' and 'Bootstrap Ecommerce' logo. It also features sections for 'Get in Touch With Us', 'Our Mobile App', 'Information', and 'Shop Departments'.

7. Payment Page

The screenshot shows a payment options page from a Django Ecommerce Site. On the left, a sidebar displays a 'Price Summary' with '\$150'. The main content area is titled 'Payment Options' and lists several payment methods: 'Cards' (with icons for Visa, Mastercard, and American Express), 'Netbanking' (with icons for SBI, ICICI, and others), 'Wallet' (with icons for Paytm, PhonePe, and others), and 'Pay Later' (with icons for Afterpay and others). A sub-section titled 'Add a new card' contains fields for 'Card Number', 'MM / YY', 'CVV', and a name 'Junie Mariam Varghese'. A checkbox labeled 'Save this card as per RBI guidelines' is present. At the bottom is a large 'Continue' button.