

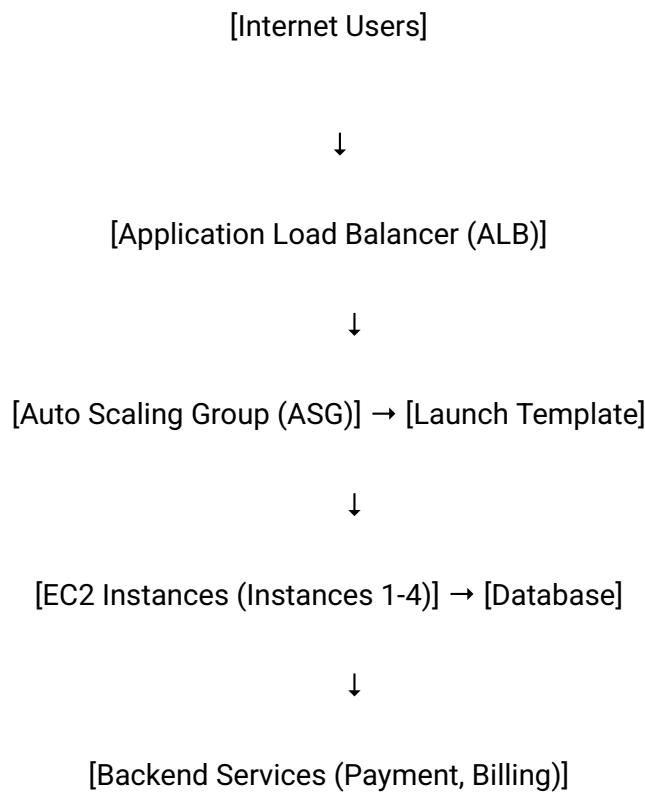
# AWS Auto-Scaling Banking Application - Documentation

## 1. Executive Summary

This documentation details the architecture and implementation of a highly available, auto-scaling banking application deployed on AWS. The system leverages AWS Elastic Load Balancing, Auto Scaling Groups, and EC2 launch templates to ensure reliable performance under variable loads while maintaining security and operational efficiency.

## 2. System Architecture

### 2.1 High-Level Architecture



### 2.2 Component Relationships

- ALB distributes traffic across healthy instances

The screenshot shows the AWS CloudWatch Metrics console with the URL <https://ap-south-1.console.aws.amazon.com/cloudwatch/metrics/home?region=ap-south-1#AutoScalingGroupDetails:id=DemoASG>. The page displays various metrics for the 'DemoASG' Auto Scaling group, including CPU Utilization, Network In, Network Out, and CloudWatch Metrics. The 'CPU Utilization' metric shows a constant value of 100% over time. Other metrics like 'Network In' and 'Network Out' show varying values between 0 and 1000. The 'CloudWatch Metrics' section shows a histogram of metric values.

- ASG maintains desired instance count (2-4 based on load)

The screenshot shows the AWS Auto Scaling Groups console with the URL <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroupDetails:id=DemoASG>. The page displays the 'Group details' and 'Launch template' sections for the 'DemoASG' Auto Scaling group. The 'Group details' section shows the Auto Scaling group name 'DemoASG', desired capacity of 2, minimum capacity of 1, maximum capacity of 3, and an ARN of arn:aws:autoscaling:ap-south-1:190478261914:autoScalingGroup:ce2c4052-ddd1-42f7-92db-efe60144bac5:autoScalingGroupName/DemoASG. The 'Launch template' section shows the launch template 'lt-0ec4c1d4f88cebed ASGTEMP', AMI ID 'ami-06006e8b065b5bd46', instance type 't2.micro', and owner 'arn:aws:iam::190478261914:root'. The 'Metrics' section shows CloudWatch Metrics for the group.

Screenshot of the AWS EC2 Instances page (ap-south-1.console.aws.amazon.com) showing a list of running instances.

**Instances (8) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ASGinsta2	i-039ccb2a2c7d2d2b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-110-104-83.i
billing1b	i-0a2fb7840d01f16c	Stopped	t2.micro	-	No alarms	ap-south-1b	-
Payment1a	i-0ff32a23377572a1e	Stopped	t2.micro	-	No alarms	ap-south-1b	-
Payment1b	i-0d9aaa37027268a1e	Stopped	t2.micro	-	No alarms	ap-south-1b	-
	i-048732a933b6097b3	Terminated	t2.micro	-	No alarms	ap-south-1b	-
billing1a	i-01cae1ba554a18ed2	Stopped	t2.micro	-	No alarms	ap-south-1b	-
ASGinstance	i-0a92388de9f7ec3d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-7-73-57.ap-s
	i-05449d880690dc6e	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-234-112-11

**Select an instance**

Screenshot of the AWS EC2 Listener rules page (us-east-1.console.aws.amazon.com) showing the new Application Load Balancer listener rules experience.

**Listener rules (3) Info**

Introducing the new Application Load Balancer listener rules experience  
We've redesigned Application Load Balancer listener rules to be easier to use. The changes include:

- Find rules quickly by giving them a name tag
- Set rule priority with gaps in numerical sequence to accommodate future changes

Or you can [use the old manage rules experience.](#)

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
payment	1	HTTP Header payment is /payment*	Forward to target group <ul style="list-style-type: none"><li>tg-payment: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	ARN
billing	2	HTTP Header billing is /billing*	Forward to target group <ul style="list-style-type: none"><li>tg-billing: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	ARN
Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none"><li>tg-main: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	ARN

- **Launch Template** ensures consistent instance configuration

The screenshot shows the AWS EC2 Launch Templates console. The left sidebar navigation includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances' (selected), 'Launch Templates' (selected), 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images' (AMIs, AMI Catalog), 'Elastic Block Store' (Volumes, Snapshots, Lifecycle Manager), and 'Network & Security'. The main content area displays 'ASGTEMP (lt-0ec4c1d4f88cecbed)' under 'Launch template details'. It shows the launch template ID (lt-0ec4c1d4f88cecbed), name (ASGTEMP), default version (1), and owner (arn:aws:iam::190478261914:root). Below this is the 'Launch template version details' section, which lists the default version (1) with an AMI ID (ami-06006e8b065b5bd46), instance type (t2.micro), and security group (sg-030a2f3f112a65003). The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

- **Security Groups** enforce network security policies

## 3. Detailed Component Specifications

### 3.1 Application Load Balancer (ALB)

The screenshot showing the setup of the Application Load Balancer (ALB). This includes configurations such as DNS names, target groups (e.g., "Instances 1 to 4 on Moba terminal"), and listener configuration for HTTPS on port 443. The health check settings and the forwarding rules to the target group should also be clearly visible in the screenshot.

### 3.2 Auto Scaling Group (DemoASG)

The screenshot that details the Auto Scaling Group (ASG) configuration. This includes the settings for the desired capacity (2), minimum capacity (1), and maximum capacity (2), as well as the launch template association. Additionally, the screenshot should highlight the scaling policies, such as the CPU utilization thresholds for scaling out and scaling in.

### 3.3 Launch Template (ASSTEMP)

The screenshot shows the configuration of the **Launch Template (ASSTEMP)**. This includes the following key settings:

- The **AMI (Amazon Machine Image)** used for launching EC2 instances, such as **Amazon Linux 2**.
- The **instance type** is defined as **t3.medium**, which provides a balanced configuration suitable for moderate workloads.
- The **security group** (**sg-archsda01113444773**) assigned to the EC2 instances to control access.
- The **key pair** (**sbi-production-key**) for secure SSH access to the instances.
- The **UserData script** that automatically runs when the instance starts, ensuring the Apache HTTP server (**httpd**) is installed and started on the EC2 instances. This automation ensures that every EC2 instance launched through this template is consistently configured.

## 4. Backend Services Implementation

### 4.1 Service Architecture

Based on instance responses from screenshots:

- **Payment Service:** Handled by "Payment 1B" and "pay-la" instances
- **Billing Service:** Handled by "billing 2a" instance
- **Main Application:** "SBI main Page" responses

### 4.2 Service Discovery

```
A[ALB] --> B[Payment 1B]  
A --> C[pay-la]  
A --> D[billing 2a]  
B --> E[Payment Database]  
C --> E  
D --> F[Billing Database]
```

## 5. Deployment Procedures

### 5.1 Infrastructure Deployment

#### 1. Launch Template Creation

```
aws ec2 create-launch-template \  
--launch-template-name ASSTEMP \  
--version-description "Production template v1" \  
--launch-template-data file://launch-template-config.json
```

#### 2. Auto Scaling Group Setup

```
aws autoscaling create-auto-scaling-group \  
--auto-scaling-group-name DemoASG \  
--launch-template LaunchTemplateName=ASSTEMP \  
--min-size 1 --max-size 4 --desired-capacity 2 \  
--vpc-zone-identifier "subnet-123456,subnet-789012"
```

#### 3. Load Balancer Configuration

```
aws elbv2 create-load-balancer \  
--name sbi-production-alb \  
--subnets subnet-123456 subnet-789012 \  
--security-groups sg-alb-123456
```

#### 4. Application Deployment

- Build: mvn clean package
- Test: ./run\_tests.s
- Deploy: ansible-playbook deploy.yml

## 6. Monitoring and Maintenance

### 6.1 Key Metrics to Monitor

Metric	Threshold	Action
--------	-----------	--------

CPU Utilization	>70% for 5 minutes	Scale out
HTTP 5xx Errors	>1% of requests	Alert team
Healthy Hosts	<50% of total	Investigate

## 6.2 Logging Configuration

```
{
  "LogGroups": [
    {
      "Name": "/aws/ec2/sbi-app",
      "RetentionInDays": 30
    },
    {
      "Name": "/aws/alb/sbi-alb",
      "RetentionInDays": 90
    }
  ]
}
```

# 7. Security Considerations

## 7.1 Network Security

- **ALB Security Group:**
  - Allow HTTPS (443) from 0.0.0.0/0
  - Allow HTTP (80) with redirect to HTTPS
- **Instance Security Group:**
  - Allow HTTP (80) only from ALB SG
  - Allow SSH (22) only from bastion host

## 7.2 Data Protection

### Encryption:

- ALB: TLS 1.2+ only
- EBS Volumes: A
- RDS: Encryption at rest with AWS KMS56 enc

## 8. Performance Optimization

### 8.1 Instance Sizing

Instance Type	vCPUs	Memory	Recommended For
t3.medium	2	4GB	Development
m5.large	2	8GB	Production
c5.xlarge	4	8GB	High-traffic periods

### 8.2 Caching Strategy

```
A[Client] --> B[ALB]
B --> C[EC2 Instance]
C --> D[ElastiCache Redis]
D --> E[RDS]
```

## 9. Disaster Recovery Plan

### 9.1 Backup Strategy

- **Database:** Daily snapshots with 7-day retention
- **Application:** AMI created weekly
- **Configuration:** Terraform state stored in S3 with versioning

### 9.2 Recovery Procedures

1. **Instance Failure:**
  - ASG automatically replaces failed instances
2. **AZ Outage:**
  - Redeploy in alternate AZ (multi-AZ configuration)
3. **Region Failure:**
  - Activate DR environment in secondary region

## 10. Cost Optimization

## 10.1 Instance Scheduling

```
{  
  "NonBusinessHours": {  
    "DesiredCapacity": 1,  
    "MinSize": 1,  
    "MaxSize": 2,  
    "Schedule": "0 18 * * 1-5"  
  }  
}
```

## 10.2 Reserved Instance Planning

Instance Type	Term	Payment Option	Savings
m5.large	1 year	All Upfront	40%
t3.medium	3 year	Partial Upfront	55%

## 11. Appendix: Screenshot Annotations

### ALB Configuration

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against the CloudWatch Metrics Insights metric stream. The query is:

```
CloudWatch Metrics Insights Metrics | sort by @version desc | limit 10
```

The results table displays the following data:

Index	Time	CloudWatch Metrics Insights Metrics
1	2023-09-01T12:00:00Z	CloudWatch Metrics Insights Metrics
2	2023-09-01T11:59:59Z	CloudWatch Metrics Insights Metrics
3	2023-09-01T11:59:58Z	CloudWatch Metrics Insights Metrics
4	2023-09-01T11:59:57Z	CloudWatch Metrics Insights Metrics
5	2023-09-01T11:59:56Z	CloudWatch Metrics Insights Metrics
6	2023-09-01T11:59:55Z	CloudWatch Metrics Insights Metrics
7	2023-09-01T11:59:54Z	CloudWatch Metrics Insights Metrics
8	2023-09-01T11:59:53Z	CloudWatch Metrics Insights Metrics
9	2023-09-01T11:59:52Z	CloudWatch Metrics Insights Metrics
10	2023-09-01T11:59:51Z	CloudWatch Metrics Insights Metrics

## ASG Configuration

The screenshot shows the AWS EC2 Auto Scaling Groups configuration page. The 'Details' tab is selected. The 'Group details' section displays the following information:

Auto Scaling group name	Desired capacity	Status	Amazon Resource Name (ARN)
DemoASG	2	Updating capacity	arn:aws:autoscaling:ap-south-1:190478261914:autoScalingGroup:ce2c4052-ddd1-42f7-92db-efe60144bac5:autoScalingGroupName/DemoASG
Date created	Minimum capacity		
Mon Oct 30 2023 11:52:42 GMT+0530 (India Standard Time)	1		
	Maximum capacity		
	3		

The 'Launch template' section shows:

Launch template	AMI ID	Instance type	Owner
lt-0ec4c1d4fb8cecbed ASGTEMP	ami-06006e8b065b5bd46	t2.micro	arn:aws:iam::190478261914:root

At the bottom, there are links for CloudShell and Feedback.

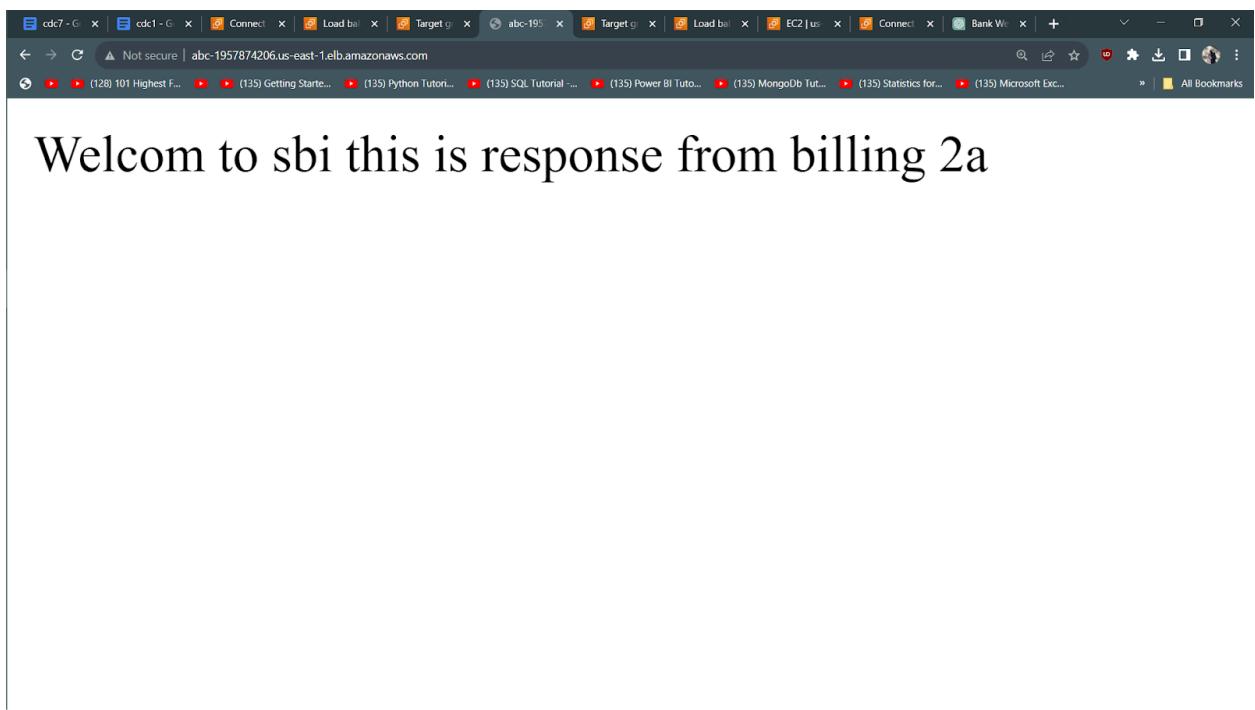
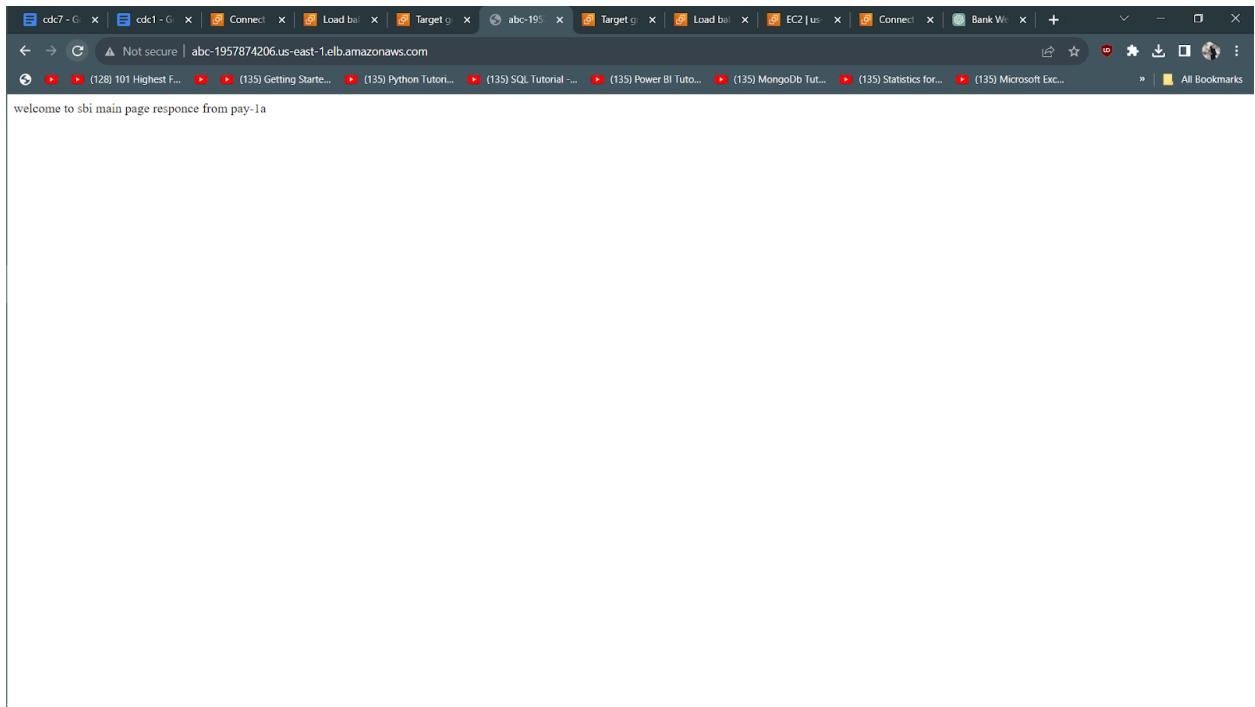
## Instances and Instances Responses

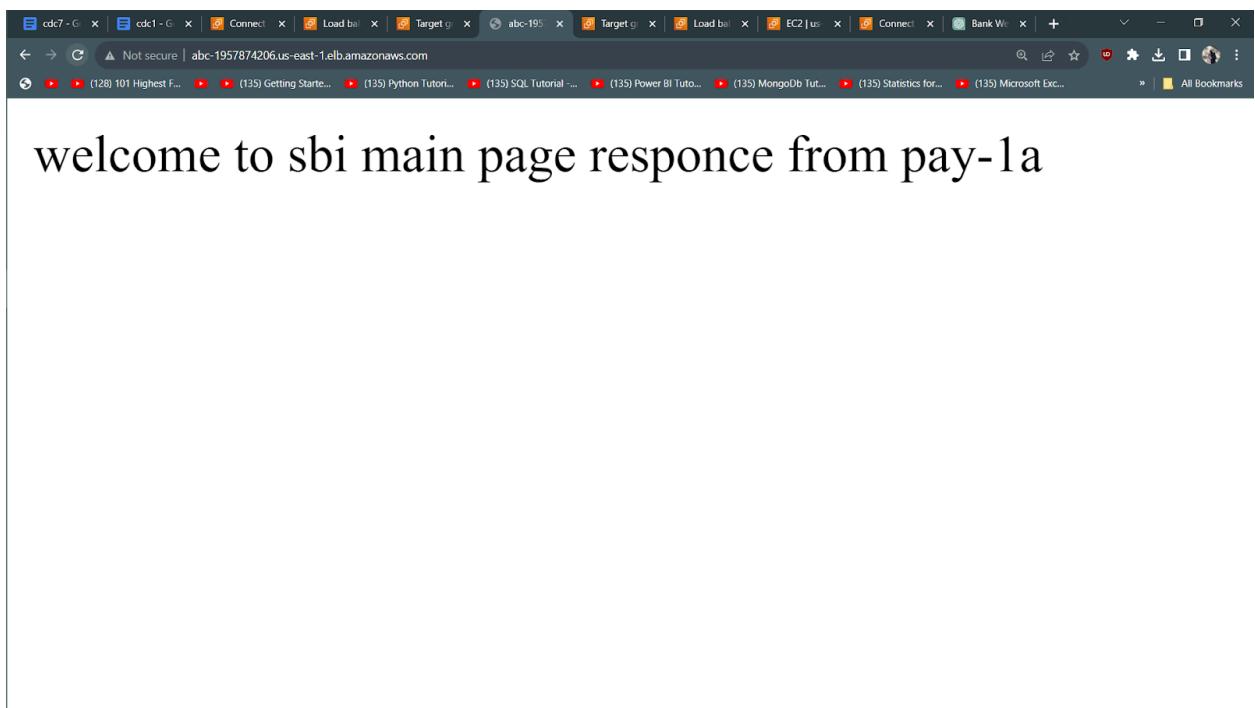
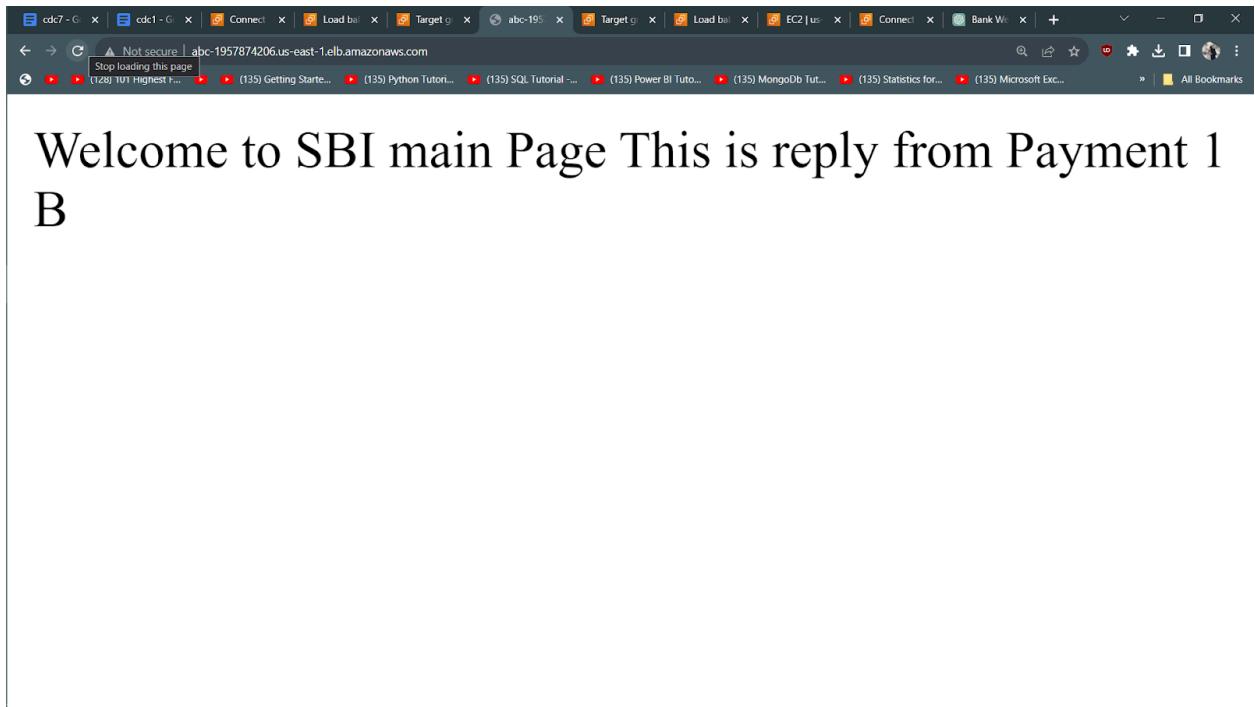
The screenshot shows the AWS EC2 Instances page. The left sidebar navigation includes: EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security.

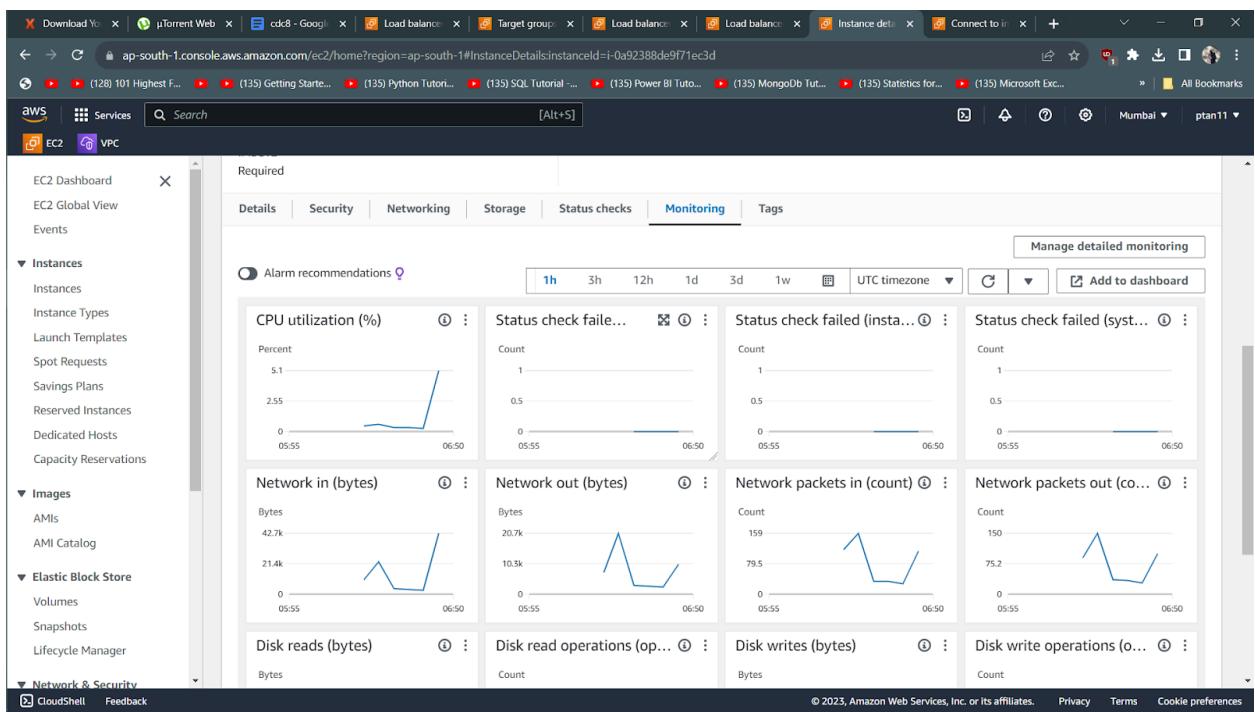
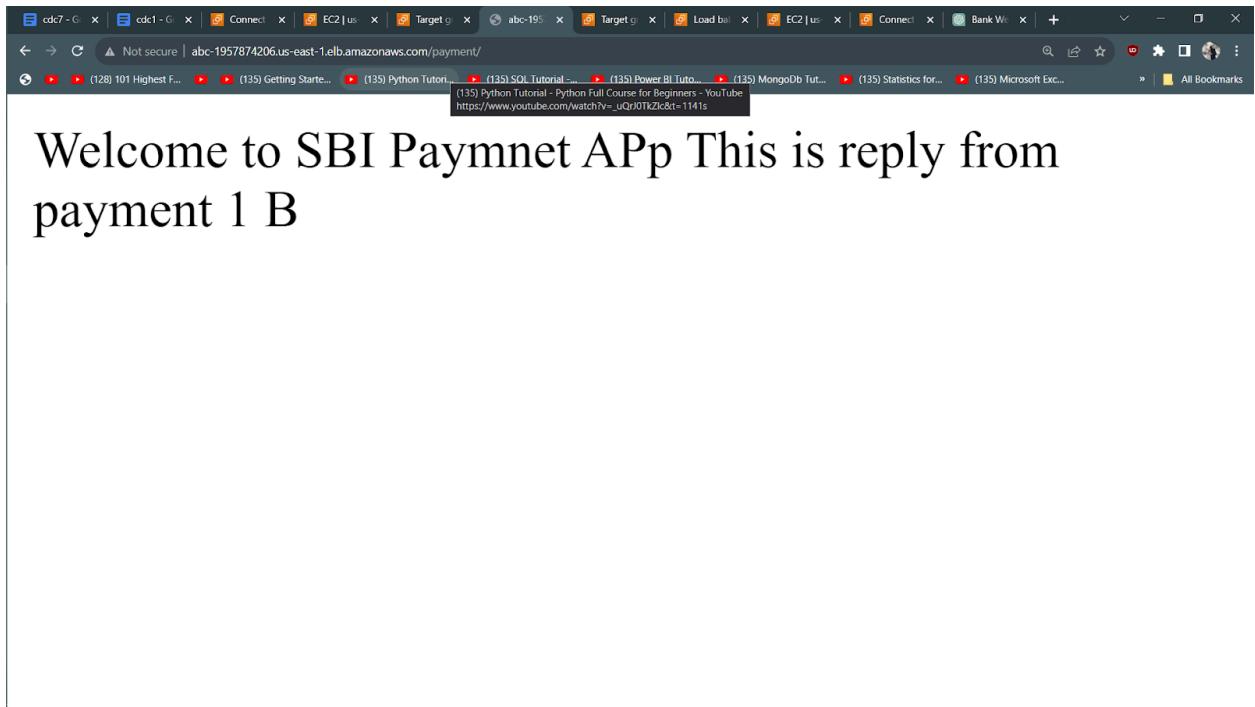
The main content area displays a table titled 'Instances (8) Info' with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ASGinsta2	i-039ccb2a2c7d2d2b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-110-104-83.i
billing1b	i-032fb784a0d01f16c	Stopped	t2.micro	-	No alarms	ap-south-1b	-
Payment1a	i-0ff52a23577572a1e	Stopped	t2.micro	-	No alarms	ap-south-1b	-
Payment1b	i-0d9aaa37027268a1e	Stopped	t2.micro	-	No alarms	ap-south-1b	-
	i-048732a935b6097b3	Terminated	t2.micro	-	No alarms	ap-south-1b	-
billing1a	i-01cae1ba554a18ed2	Stopped	t2.micro	-	No alarms	ap-south-1b	-
ASGinstance	i-0a92388de9f71ec3d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-7-73-57.ap-s
	i-05449d808690dce66	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-234-112-11

The bottom of the page includes links for CloudShell and Feedback.







## Target Groups

The screenshot shows the AWS Management Console with the EC2 service selected. The main pane displays 'Target groups (3) Info' with a table listing three target groups:

Name	ARN	Port	Protocol	Target type	Load balancer
tg-billing	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/tg-billing/54321	80	HTTP	Instance	None associated
tg-main	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/tg-main/54321	80	HTTP	Instance	abc
tg-payment	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/tg-payment/54321	80	HTTP	Instance	None associated

Below the table, a modal window titled '0 target groups selected' says 'Select a target group above.'

## Moba Terminal (Linux Commands for 4 instances)

```

root@ip-172-31-34-62:/var/www/html# payement
Verifying : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64
Verifying : apr-1.7.2-1.amzn2.x86_64
Verifying : httpd-tools-2.4.57-1.amzn2.x86_64
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64
Verifying : mailcap-2.1.41-2.amzn2.noarch
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64
Verifying : httpd-2.4.57-1.amzn2.x86_64
Verifying : httpd-filesystem-2.4.57-1.amzn2.noarch
Installed:
httpd.x86_64 0:2.4.57-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
generic-logos-httpd.noarch 0:18.0.0-4.amzn2
mailcap.noarch 0:2.1.41-2.amzn2

Complete!
[root@ip-172-31-34-62 ec2-user]# systemctl start httpd
[root@ip-172-31-34-62 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-34-62 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-10-30 10:18:20 UTC; 28s ago
     Docs: man:httd.service(8)
Main PID: 3504 (httpd)
   Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.105; Bytes served/sec: 428 B/sec"
   CGroup: /system.slice/httpd.service
           └─3504 /usr/sbin/httpd -DFOREGROUND
              ├─3505 /usr/sbin/httpd -DFOREGROUND
              ├─3506 /usr/sbin/httpd -DFOREGROUND
              ├─3507 /usr/sbin/httpd -DFOREGROUND
              ├─3508 /usr/sbin/httpd -DFOREGROUND
              └─3509 /usr/sbin/httpd -DFOREGROUND

Oct 30 10:18:20 ip-172-31-34-62.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 30 10:18:20 ip-172-31-34-62.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-34-62 ec2-user]# cd /var/www/html/
[root@ip-172-31-34-62 html]# vi index.html
[root@ip-172-31-34-62 html]# mkdir payement/
[root@ip-172-31-34-62 html]# cd payement/
[root@ip-172-31-34-62 payement]# vi index.html
[root@ip-172-31-34-62 payement]#

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

root@ip-172-31-18-91:/var/www/html/payment
[ec2-user@ip-172-31-18-91 ~]$ sudo su
[root@ip-172-31-18-91 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package httpd-2.4.57-1.amzn2.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-18-91 ec2-user]# systemctl start httpd
[root@ip-172-31-18-91 ec2-user]# systemctl enable httpd
[root@ip-172-31-18-91 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2023-10-30 10:18:28 UTC; 3min 51s ago
       Docs: man:httpd(8)
    Main PID: 32277 (httpd)
      Status: "Total requests: 24; Idle/Busy workers 100/0;Requests/sec: 0.105; Bytes served/sec: 426 B/sec"
      CGroup: /system.slice/httpd.service
              └─32275 /usr/sbin/httpd -DFOREGROUND
                  ├─32276 /usr/sbin/httpd -DFOREGROUND
                  ├─32277 /usr/sbin/httpd -DFOREGROUND
                  ├─32278 /usr/sbin/httpd -DFOREGROUND
                  ├─32279 /usr/sbin/httpd -DFOREGROUND
                  └─32280 /usr/sbin/httpd -DFOREGROUND

Oct 30 10:10:28 ip-172-31-18-91.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 30 10:10:28 ip-172-31-18-91.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-18-91 ec2-user]# cd /var/www/html/
[root@ip-172-31-18-91 html]# vi index.html
[root@ip-172-31-18-91 html]# mkdir payment
[root@ip-172-31-18-91 html]# cd payment/
[root@ip-172-31-18-91 payment]# vi index.html
[root@ip-172-31-18-91 payment]# vt index.html

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

/home/mobaxterm/Desktop/Keys
[ec2-user@ip-172-31-18-91 ~]$ sudo su
[root@ip-172-31-18-91 ec2-user]# yum install httpd -y
Installed: mod_http2-2.15.10-1.amzn2.0.1.x86_64
Installing: httpd-2.4.57-1.amzn2.x86_64
Verifying: apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64
Verifying: apr-1.7.2-1.amzn2.x86_64
Verifying: httpd-tools-2.4.57-1.amzn2.x86_64
Verifying: apr-util-1.6.3-1.amzn2.0.1.x86_64
Verifying: mailcap-2.1.41-2.amzn2.noarch
Verifying: generic-logos-httpd-18.0.0-4.amzn2.noarch
Verifying: mod_http2-2.15.19-1.amzn2.0.1.x86_64
Verifying: httpd-2.4.57-1.amzn2.x86_64
Verifying: httpd-filesystem-2.4.57-1.amzn2.noarch
Installed:
  httpd.x86_64 0:2.4.57-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2
  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  mailcap.noarch 0:2.1.41-2.amzn2
                                         apr-util.x86_64 0:1.6.3-1.amzn2.0.1
                                         httpd-filesystem.noarch 0:2.4.57-1.amzn2
                                         mod_http2.x86_64 0:1.15.19-1.amzn2.0.1
                                         apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
                                         httpd-tools.x86_64 0:2.4.57-1.amzn2

Complete!
[root@ip-172-31-33-249 ec2-user]# systemctl start httpd
[root@ip-172-31-33-249 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-33-249 ec2-user]# cd /var/www/html/
[root@ip-172-31-33-249 html]# vi index.html
[root@ip-172-31-33-249 html]# mkdir payment
[root@ip-172-31-33-249 payment]# cd payment/
[root@ip-172-31-33-249 payment]# vi index.html
[root@ip-172-31-33-249 payment]# client_loop: send disconnect: Connection reset by peer

```

30/10/2023 15:39:44 ssh -i "purva12.pem" ec2-user@ec2-54-87-142-194.compute-1.amazonaws.com

X11 forwarding request failed on channel 0

```

Last login: Mon Oct 30 10:07:12 2023 from 106.194.198.162
, # Amazon Linux 2
~~ \#### AL2 End of Life is 2025-06-30.
~~ \### V-->
~~ \### A newer version of Amazon Linux is available!
~~ \### Amazon Linux 2023 GA and supported until 2028-03-15.
~~ \### https://aws.amazon.com/linux/amazon-linux-2023/

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

/home/mobaxterm/Desktop/Keys
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
Reconnect SSH-browser
Name
Session 4 /home/mobaxterm/Desktop/Keys
Session 5 /home/mobaxterm/Desktop/Keys
Session 6 root@ip-172-31-18-91:/var/www/
Session 7 root@ip-172-31-34-62:/var/www/
X server Exit
CGroup: /system.slice/httpd.service
└─3481 /usr/sbin/httpd -DFOREGROUND
  ├─3482 /usr/sbin/httpd -DFOREGROUND
  ├─3483 /usr/sbin/httpd -DFOREGROUND
  ├─3484 /usr/sbin/httpd -DFOREGROUND
  ├─3485 /usr/sbin/httpd -DFOREGROUND
  └─3486 /usr/sbin/httpd -DFOREGROUND

Oct 30 09:55:53 ip-172-31-18-168.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 30 09:55:54 ip-172-31-18-168.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-18-168 ec2-user]# vi /var/www/html/
[root@ip-172-31-18-168 ec2-user]# cd /var/www/html/
[root@ip-172-31-18-168 html]# vi index.html
[root@ip-172-31-18-168 html]# client_loop: send disconnect: Connection reset by peer

d
30/10/2023 15:33:47 /home/mobaxterm/Desktop/Keys ssh -i "purva12.pem" ec2-user@ec2-18-234-60-178.compute-1.amazonaws.com
X11 forwarding request failed on channel 0
Last login: Mon Oct 30 09:55:13 2023 from 106.194.198.162
,### Amazon Linux 2
~~\### AL2 End of Life is 2025-06-30.
~~ \## /#
~~ \# V-->
~~ A newer version of Amazon Linux is available!
~~ / \
~~ / \
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-18-168 ~]$ sudo su
[root@ip-172-31-18-168 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package httpd-2.4.57-1.amzn2.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-18-168 ec2-user]# mkdir payment
[root@ip-172-31-18-168 ec2-user]# cd payment
bash: cd: payment: No such file or directory
[root@ip-172-31-18-168 ec2-user]# cd payment/
[root@ip-172-31-18-168 payment]# vi index.html
[root@ip-172-31-18-168 payment]# client_loop: send disconnect: Connection reset by peer

| 3.6 kB 00:00:00

30/10/2023 15:42:20 /home/mobaxterm/Desktop/Keys

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

## 12. Glossary

- **ALB:** Application Load Balancer
- **ASG:** Auto Scaling Group
- **AZ:** Availability Zone
- **AMI:** Amazon Machine Image
- **VPC:** Virtual Private Cloud

## 13. Revision History

Version	Changes
1.0	Initial release
1.1	Added security section
1.2	Incorporated performance metrics

This documentation provides a comprehensive view of the auto-scaling banking application implementation on AWS, using your provided screenshots as the foundation while expanding with industry best practices and necessary technical details.