

# Solution for the Hiring Assignment

---

**By Nitin Tiwari**

- [Solution for the Hiring Assignment](#)
  - [Overview](#)
  - [Problems](#)
  - [Solution for Problem a](#)
    - [Problem - 1](#)
    - [Problem - 2](#)
    - [Problem - 3](#)
    - [Problem - 4](#)
    - [Summary](#)
  - [Solution for Problem b](#)
    - [Proposed High Level Architecture](#)
    - [Technical Solution Brief](#)
  - [Deployment Best Practice](#)
  - [Alignment with AWS Well-Architected Framework](#)
    - [Operational Excellence](#)
    - [Security](#)
    - [Reliability](#)
    - [Performance Efficiency](#)
    - [Cost Optimization](#)

## Overview

This document describes the solution to the problem statement specified in the [AWS EMEA Solutions Architect Hiring Assignment](#).

Customer has provided a [cloudformation template](#) for the configuration they created with their current level of AWS Knowledge. Configuration is faulty and the website is not loading.

## Problems

Requirement is broken down to three problems:

- a) Troubleshoot the implementation by doing the minimum amount of work required to make the web site operational. Your customer expects detailed written troubleshooting instructions or scripts for the in-house team.
- b) Propose short term changes you could help them implement to improve the availability, security, reliability, cost and performance before the project goes into production. Your customer expects you to explain the business and technical benefits of your proposals, with artifacts such as a design or architecture document and diagrams.
- c) Optionally, propose high level alternative solution(s) for the longer term as their web application becomes more successful.

## Solution for Problem a

a) Troubleshooting steps executed for the current proof of concept:

- Cloud formation template was launched in the AWS, a stack is created
- As suggested in the problem statement, the loadbalancer link did not render the expected webpage.
- In such a case the first point to look is whether the Load Balancer is configured correctly, and routing correctly defined.

### Problem - 1

On inspection we found that the instance attached doesn't match the Availability Zones configured for the Load Balancer.

Load balancer: AWS-SA-Test1-SAelb-AWIPIG4F4WO0

Connection Draining: Disabled (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status
i-000c36d0b662a01a0	Instance1-Nitin Tiwari	eu-west-1a	OutOfService (1)

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
eu-west-1b	subnet-0a2b9b668d039dd33	10.0.1.0/24	0	No (Availability Zone contains no healthy targets)	-

**Solution:** Add the same AZ, instance is available in.

**Add and Remove Subnets**

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-0015ed771c8c481c8

Please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

**Available subnets**

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
+	eu-west-1a	subnet-08e5472e7c7306108	10.0.0.0/24	PublicSubnetA-Nitin Tiwari
+	eu-west-1a	subnet-06d06e71581605f96	10.0.2.0/24	PrivateSubnetA-Nitin Tiwari
+	eu-west-1b	subnet-08f2eaa5fe00959f2	10.0.3.0/24	PrivateSubnetB-Nitin Tiwari

**Selected subnets**

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
-	eu-west-1b	subnet-0a2b9b668d039dd33	10.0.1.0/24	PublicSubnetB-Nitin Tiwari

Add 'Public subnet in eu-west-1a' to Selected subnets

Cancel Save

- After doing the change above, we found the error on the Instance Status changed to Health Check configuration problem.

## Problem - 2

Load Balancer fails Health Check, it refers to TCP:443, while instance isn't configured on port 443, used commonly for secured HTTP connection.

Resource Groups

Create Load Balancer Actions

search : AWS-SA-Test1-SAelb-AWIPIG4F4WO0 Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
AWS-SA-Test1-SAelb-AWIP...	AWS-SA-Test1-SAelb-AWIP...		vpc-0015ed771c8c481c8	eu-west-1a, eu-west-1b	classic	June 22, 2020 at 11:05:19 A...

Load balancer: AWS-SA-Test1-SAelb-AWIPIG4F4WO0

Description Instances Health check Listeners Monitoring Tags Migration

Connection Draining: Disabled (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-000c36d0b662a01a0	Instance1-Nitin Tiwari	eu-west-1a	OutOfService	Remove from Load Balancer

Instance has failed at least the UnhealthyThreshold number of health checks consecutively.

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
eu-west-1a	subnet-08e5472e7c7306108	10.0.0.0/24	1	No (Availability Zone contains no healthy targets)	Remove from
eu-west-1b	subnet-0a2b9b668d039dd33	10.0.1.0/24	0	No (Availability Zone contains no healthy targets)	Remove from

**Solution:** Update the health check page, as below:

Resource Groups

Create Load Balancer Actions

search : AWS-SA-Test1-SAelb-AWIPIG4F4WO0 Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
AWS-SA-Test1-SAelb-AWIP...	AWS-SA-Test1-SAelb-AWIP...		vpc-0015ed771c8c481c8	eu-west-1a, eu-west-1b	classic	June 22, 2020 at 11:05:19 A...

Load balancer: AWS-SA-Test1-SAelb-AWIPIG4F4WO0

Description Instances Health check Listeners Monitoring Tags

Ping Target TCP:443

Timeout 5 seconds

Interval 15 seconds

Unhealthy threshold 2

Healthy threshold 2

Edit Health Check

Add correct health check page

**Configure Health Check**

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol HTTP

Ping Port 80

Ping Path /demo.html

**Advanced Details**

Response Timeout 5 seconds

Interval 15 seconds

Unhealthy threshold 2

Healthy threshold 2

Cancel Save

- This step resolves the Instance configuration service, however the website still doesn't load.

Resource Groups

Create Load Balancer Actions

search : AWS-SA-Test1-SAelb-AWIPIG4F4W00 Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
AWS-SA-Test1-SAelb-AWIPI...	AWS-SA-Test1-SAelb-AWIPI...		vpc-0015ed771c8c481c8	eu-west-1a, eu-west-1b	classic	June 22, 2020 at 11:05:19 A...	

Load balancer: AWS-SA-Test1-SAelb-AWIPIG4F4W00

Description Instances Health check Listeners Monitoring Tags Migration

Connection Draining: Disabled (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-000c36d0b662a01a0	Instance1-Nitin Tiwari	eu-west-1a	InService (i)	Remove from Load Balancer

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
eu-west-1a	subnet-08e54726c7306108	10.0.0.0/24	1	Yes	Remove from Load Balancer
eu-west-1b	subnet-0a2b9b668d039dd33	10.0.1.0/24	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer

Next step is to check the Security group configurations.

### Problem - 3

On inspection we found that Security group for Load balancer was not configured to allow Inbound traffic.

Resource Groups

Create Load Balancer Actions

search : AWS-SA-Test1-SAelb-AWIPIG4F4W00 Add filter

AWS-SA-Test1-SAelb-AWIPI...

type Classic (migrate view)

Scheme Internet-facing

Availability Zones subnet-0a2b9b668d039dd33 - eu-west-1

Port Configuration

Port Configuration 80 (HTTP) forwarding to 80 (HTTP)

Stickiness: Disabled

Edit stickiness

Security

Source Security Group sg-09cbe8dae55db4d4e, AWS-SA-Test1-SAelb-AWIPIG4F4W00

Edit security groups

Attributes

Security groups | EC2 Management Console

eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1...

Create Security Group Actions

search : sg-09cbe8dae55db4d4e Add filter

Name	Group ID	Group Name	VPC ID	Owner
ELBSecurity...	sg-09cbe8dae55db4d4e	AWS-SA-Test1-SASGELB-1...	vpc-0015ed771c8c481c8	491

Security Group: sg-09cbe8dae55db4d4e

Description Inbound Outbound Tags

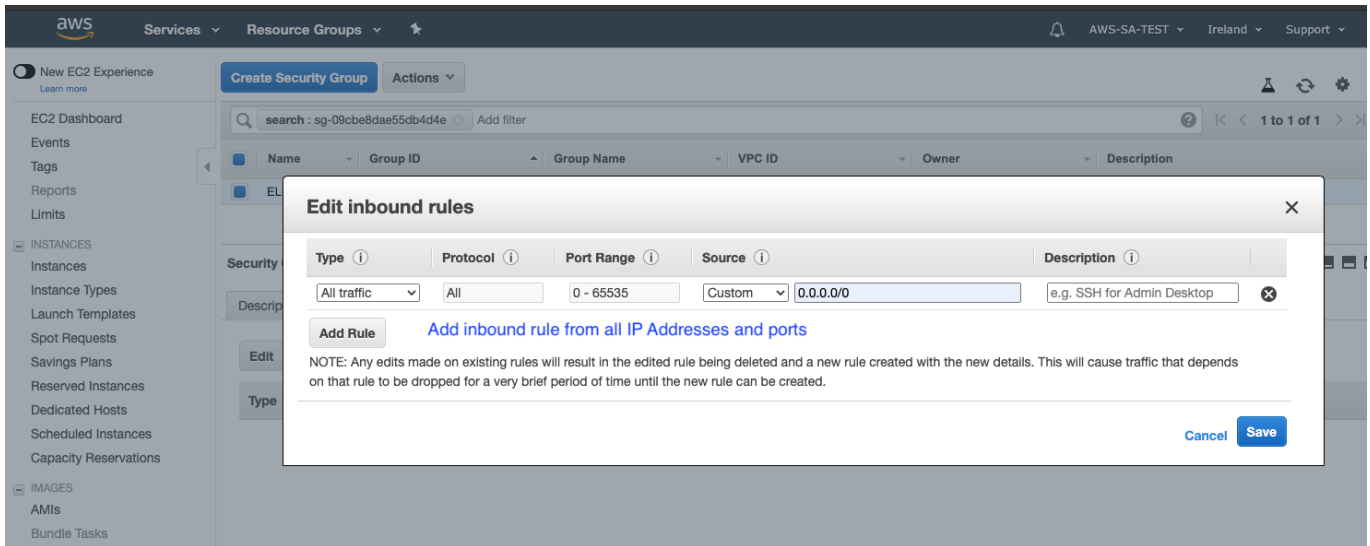
Edit

Type	Protocol	Port Range	Description
------	----------	------------	-------------

This security group has no rules

No Inbound rules

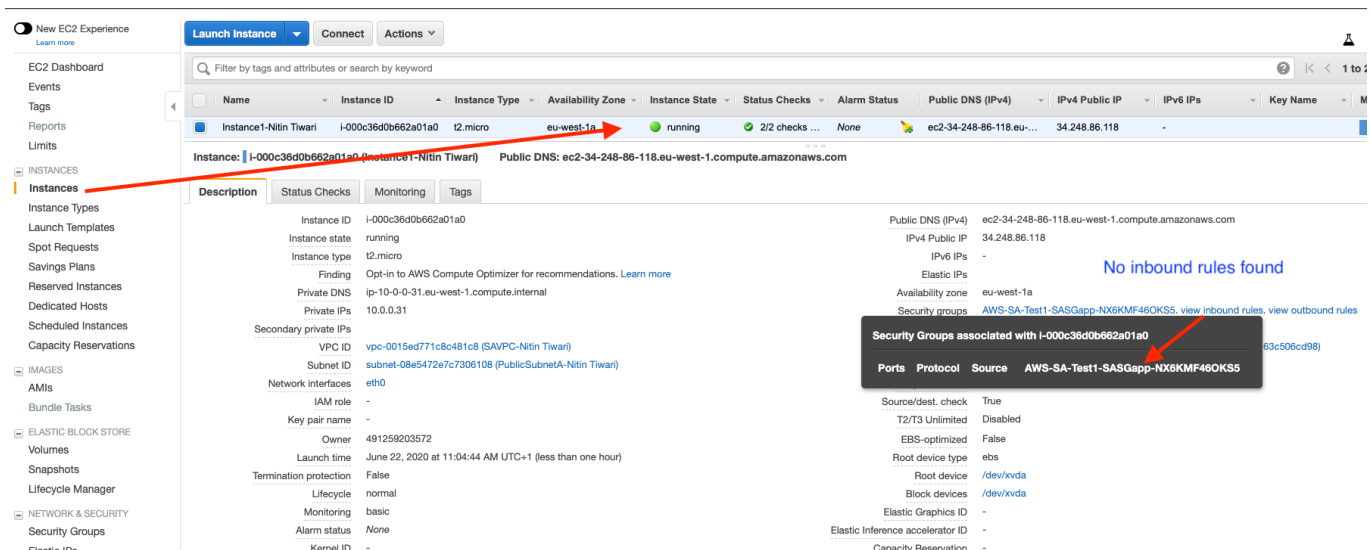
**Solution:** Add a new inbound rule to allow incoming traffic from All IP and Ports.



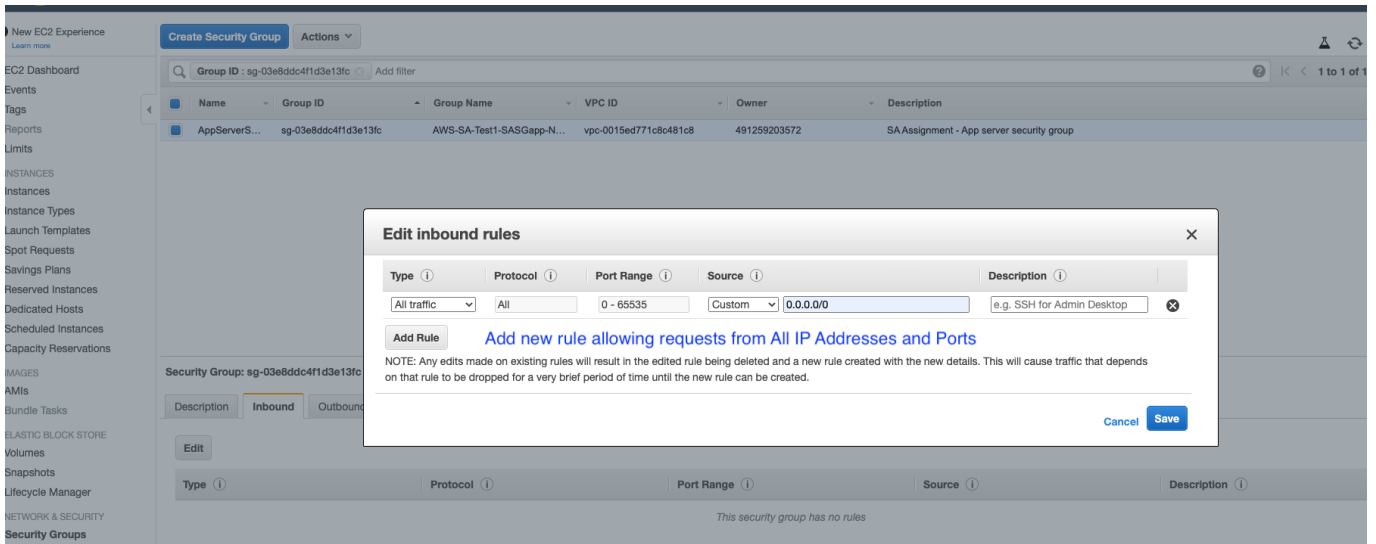
- Next step is to check the rules for the EC2 instance and same problem as above was encountered - No inbound traffic rule was found

## Problem - 4

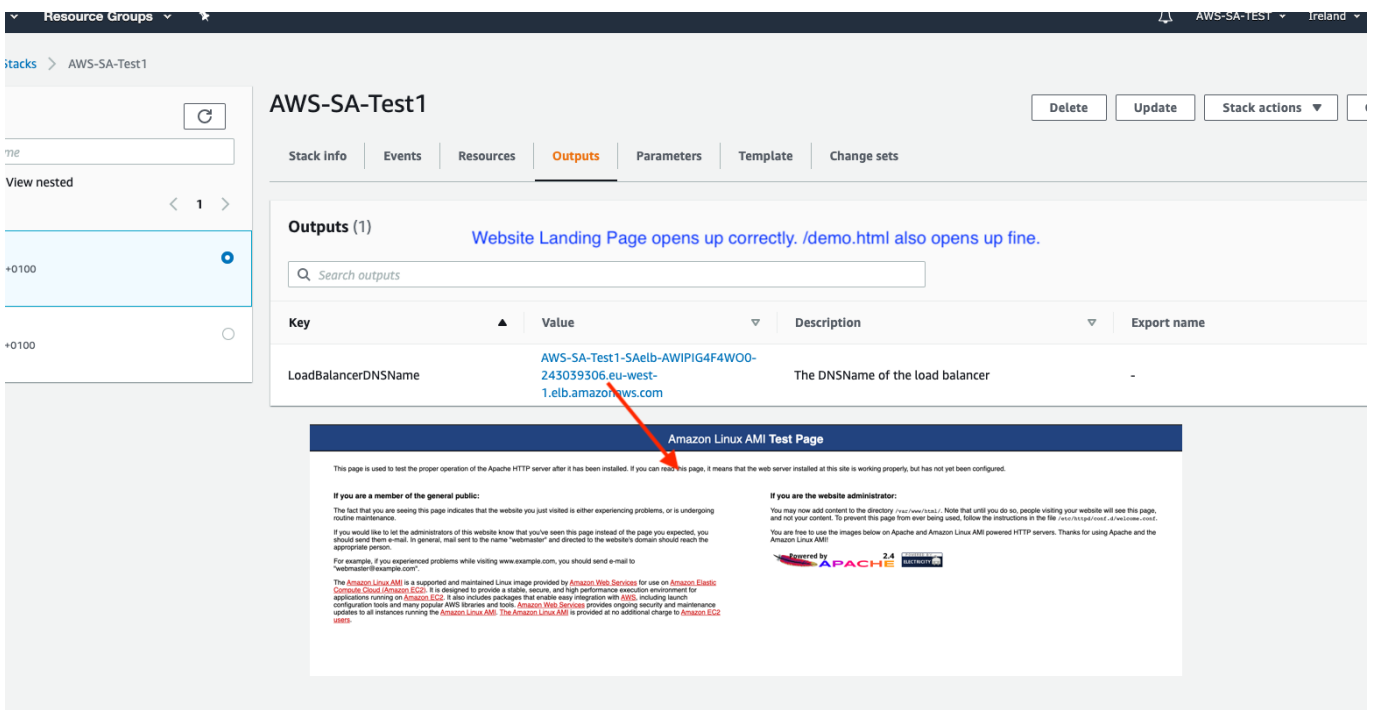
Instance doesn't allow inbound requests.



**Solution:** Add a new inbound rule to allow incoming traffic from All IP and Ports.



At this stage, the website starts to load correctly.



## Summary

To fix the problems customer is facing to launch their website can be resolved by doing following changes to Cloudformation Template:

1. New ELB Configuration with correct subnet, and healthcheck page;

SAelb:

```
Type: AWS::ElasticLoadBalancing::LoadBalancer
Properties:
  Subnets: [!Ref 'PublicSubnetA']
  Instances: [!Ref 'SAInstance1']
  SecurityGroups: [!Ref 'SASGELB']
  Listeners:
    - LoadBalancerPort: '80'
```

```

    InstancePort: '80'
    Protocol: HTTP
    HealthCheck:
      HealthyThreshold: '2'
      Interval: '15'
      Target: HTTP:80/demo.html
      Timeout: '5'
      UnhealthyThreshold: '2'
    Tags:
      - Key: environment
        Value: sa-assignment
      - Key: Name
        Value: !Join ['-', [ELB, !Ref 'CandidateName']]

```

## 2. Add Inbound traffic rule for Load Balancer

```

SASGELBINGRESS:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    CidrIp: 0.0.0.0/0
    Description: Inbound rule
    FromPort: '-1'
    GroupId: !GetAtt SASGELB.GroupId
    IpProtocol: '-1'
    ToPort: 80

```

## 3. Add Inbound traffic rule for the EC2 Instance

```

SASGAPPINGRESS:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    CidrIp: 0.0.0.0/0
    Description: Inbound rule
    FromPort: '-1'
    GroupId: !GetAtt SASGapp.GroupId
    IpProtocol: '-1'
    ToPort: 80

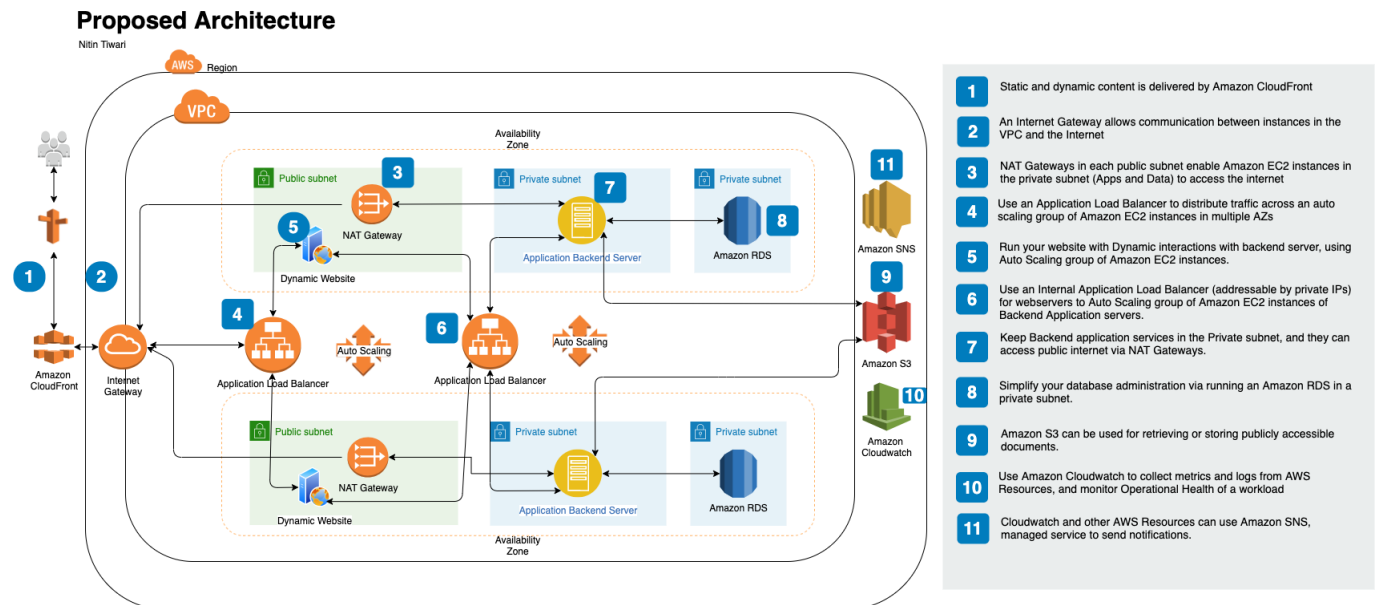
```

Final cloudformation template with above resolutions can be found [here](#).

## Solution for Problem b

Proposal for short term changes to help the customer improve the availability, security, reliability, cost and performance before the project goes into production can be find in a separate document below.

## Proposed High Level Architecture



## Technical Solution Brief

- Solution proposes a clear separation of various components of an application. With the current knowledge of Customer's future requirements, I propose provisioning of 2 AWS Availability zones.
- Each with 1 Public Subnet and 2 Private Subnets. This ensures the Network level Security of the protected resources, like core backend application, and database(s).
- Provision AWS Route 53 for DNS Resolution of the publicly available resources, i.e. static website and content.
- I propose usage of Amazon CloudFront for static content delivery.
- Public subnet(s) host a web server for static and dynamic content, fronted by an Amazon Load Balancer in auto scaling group to facilitate performance in the peak loads. Restrict the Web servers accessible only via Load Balancer.
- Public Subnet(s) also host a NAT Gateway to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances.
- I propose setting up an Internal Elastic Load Balancer for Application backend server access, again in an Auto Scaling group and servers accessible via Load Balancers only.
- Solution also includes 3 more services:
  - Amazon S3 - Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This service must be used for storage of any publicly accessible document.
  - Amazon Cloudwatch - it is a monitoring and observability service, it collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.
  - Amazon SNS - primarily used for notifications, email or SMS etc.

## Deployment Best Practice



All of the above configuration shall be built via Cloudformation template, to be able to define the whole infrastructure as code. This gives us flexibility to replicate the same environment or setup in another AWS region programmatically in future.

## Alignment with AWS Well-Architected Framework

Solution adheres to the following AWS Well-Architected framework and the document later describes how these principles are met with the proposed design:

- Operational Excellence
- Security
- Reliability
- Performance Efficiency
- Cost Optimization

### Operational Excellence

Provisioning of Amazon Cloudwatch, Auto scaling, AWS managed services for Database like RDS and Notification enables the system to run and deliver business value and continually improve supporting processes and procedures.

### Security

- Solution must ensure secure connection to the Web servers only allowing traffic on https. This would involve certificate management, Route 53 DNS configuration, and enabling Load Balancer listener to listen to only https incoming traffic, and deny others.
- Provision IAM (Identity and Access Management) policy for credentials and authentication mechanism.
- Provision controls to capture and analyze logs and events, to secure from unauthorized access, or threats.
- Provision of public and private subnets and corresponding Access Control Layer protects the network.
- Security groups protect EC2 instances

### Reliability

- Establish Auto scaling policies to scale the availability of instances to serve requests reliably according to demand.
- Configuring the Amazon CloudWatch to monitor runtime metrics, and aggregate logs.
- Use of Amazon S3 for backups.

### Performance Efficiency

- Solution is based on the brief knowledge of Customer requirements, however it can be improved for performance, with more clarity on Compute, and Storage requirements.
- For example, AWS Lambda can be suggested in the solution for some of the Business functions to optimize resource utilization and eventually provide better performance.

### Cost Optimization

- Using managed services, you can reduce or remove much of your administrative and operational overhead, freeing you to work on applications and business-related activities.

- Be expenditure aware using AWS Cost Explorer, and AWS Budgets that notify you if your usage or spend exceeds actual or forecast budgeted amounts.
- Optimize resources over time.