Solution for the Hiring Assignment

By Nitin Tiwari

Overview

This document describes the solution to the problem statement specified in the AWS EMEA Solutions Architect Hiring Assignmnet.

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.

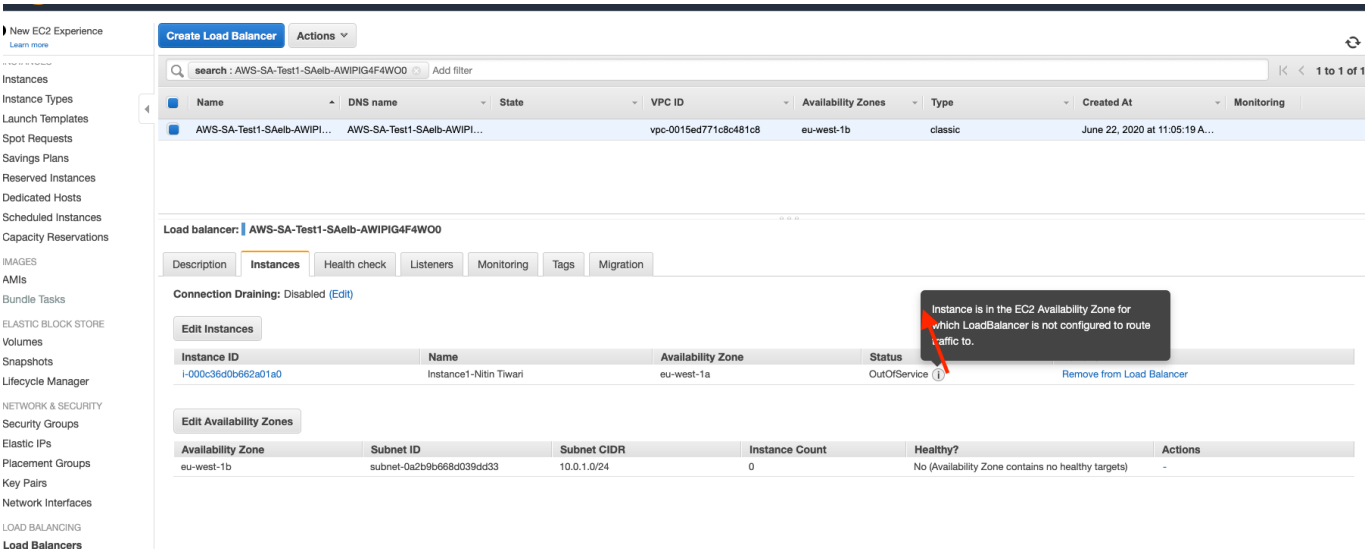
Solution for Peoblem a

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

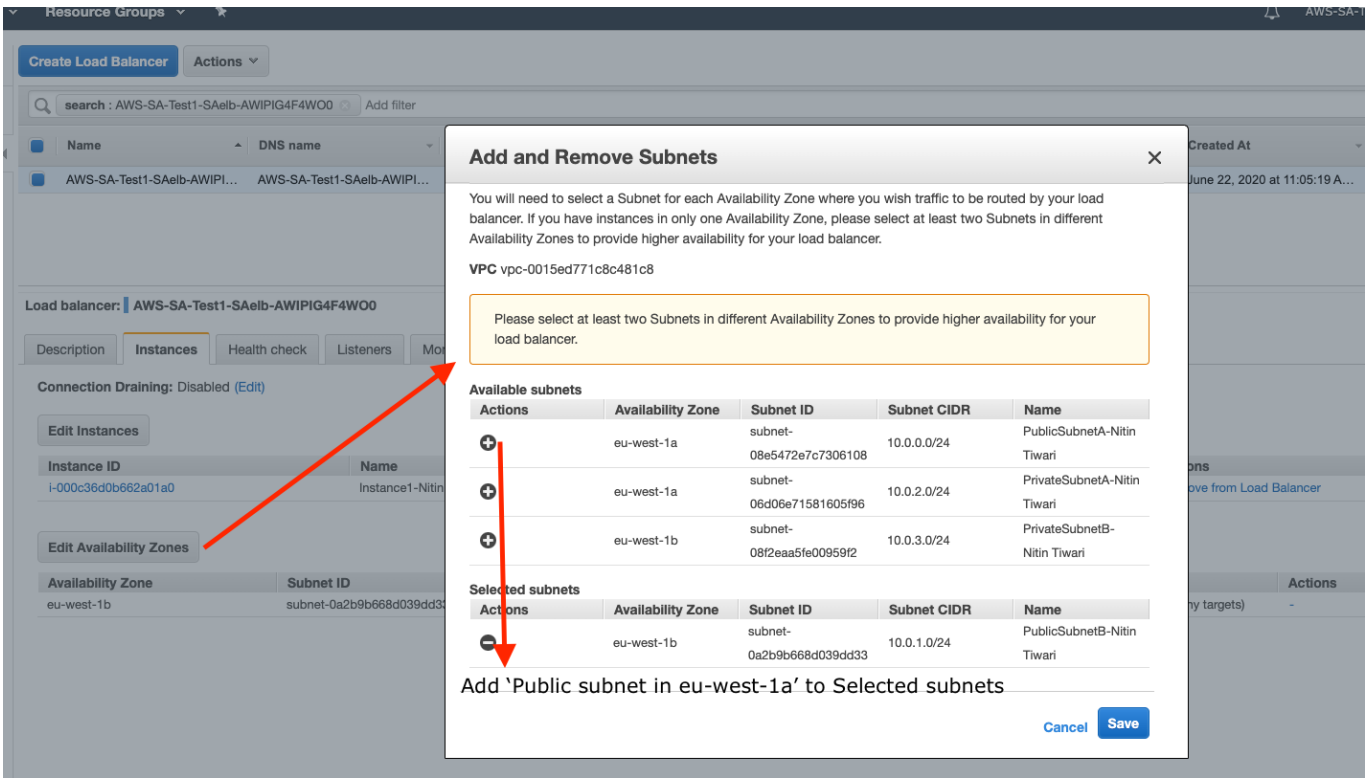
- Cloud formation template was laucnhed in the AWS, a stack is created
- As suggested in the problem statement, the loadbalancer link did not render the expected webpage.
- In such a case the first point to look is whether the Load Balancer is configured correctly, and routing correctly defined.

Problem - 1

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



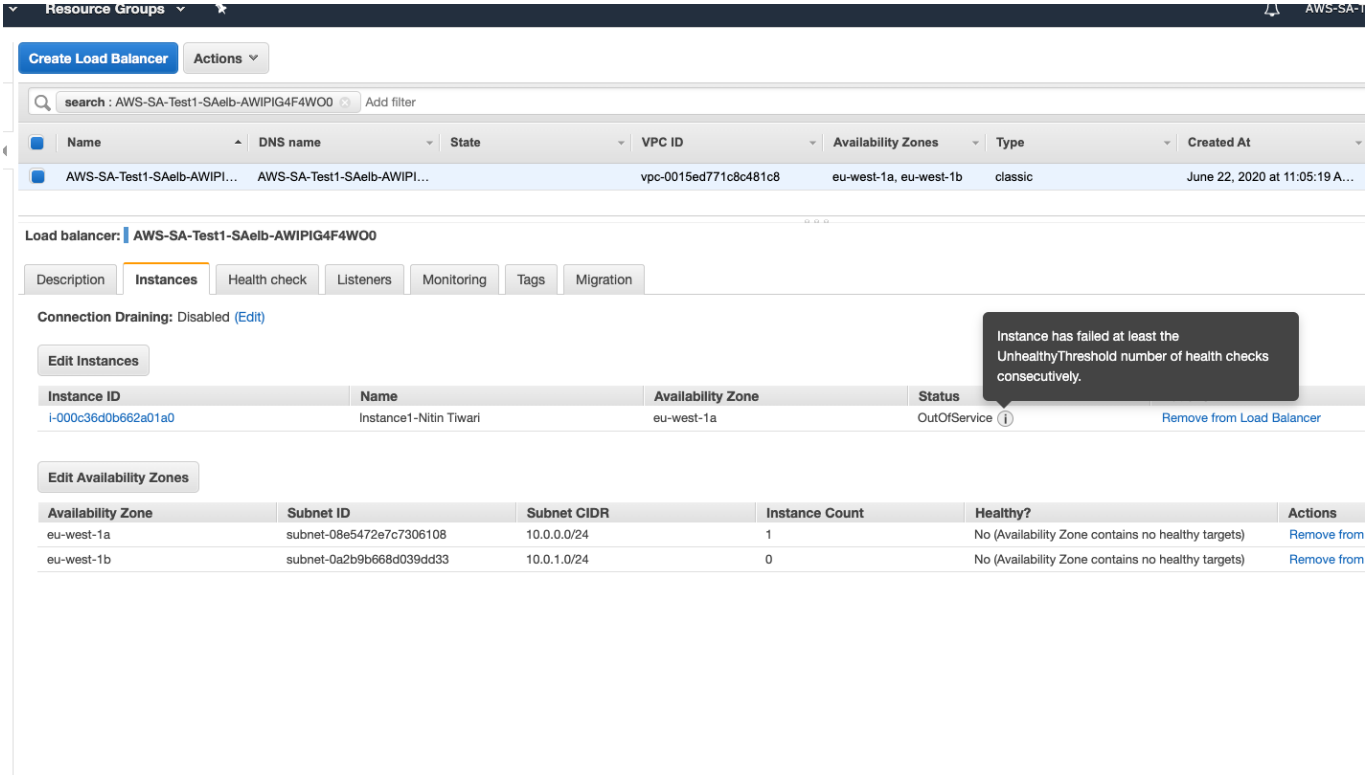
Solition: Add the same AZ, instance is available in.



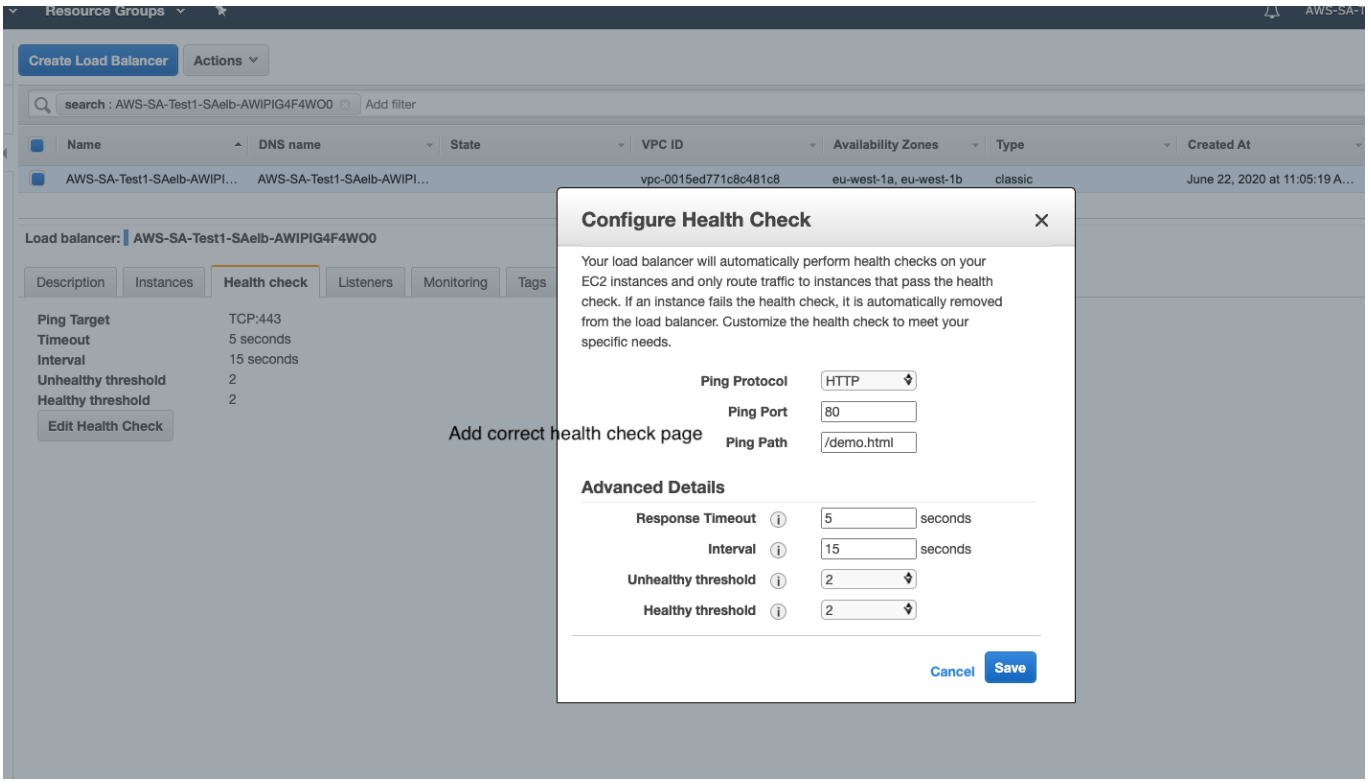
- After doing the change above, we found the error on the Instance Status changed to Health Check configuration problem.

Problem - 2

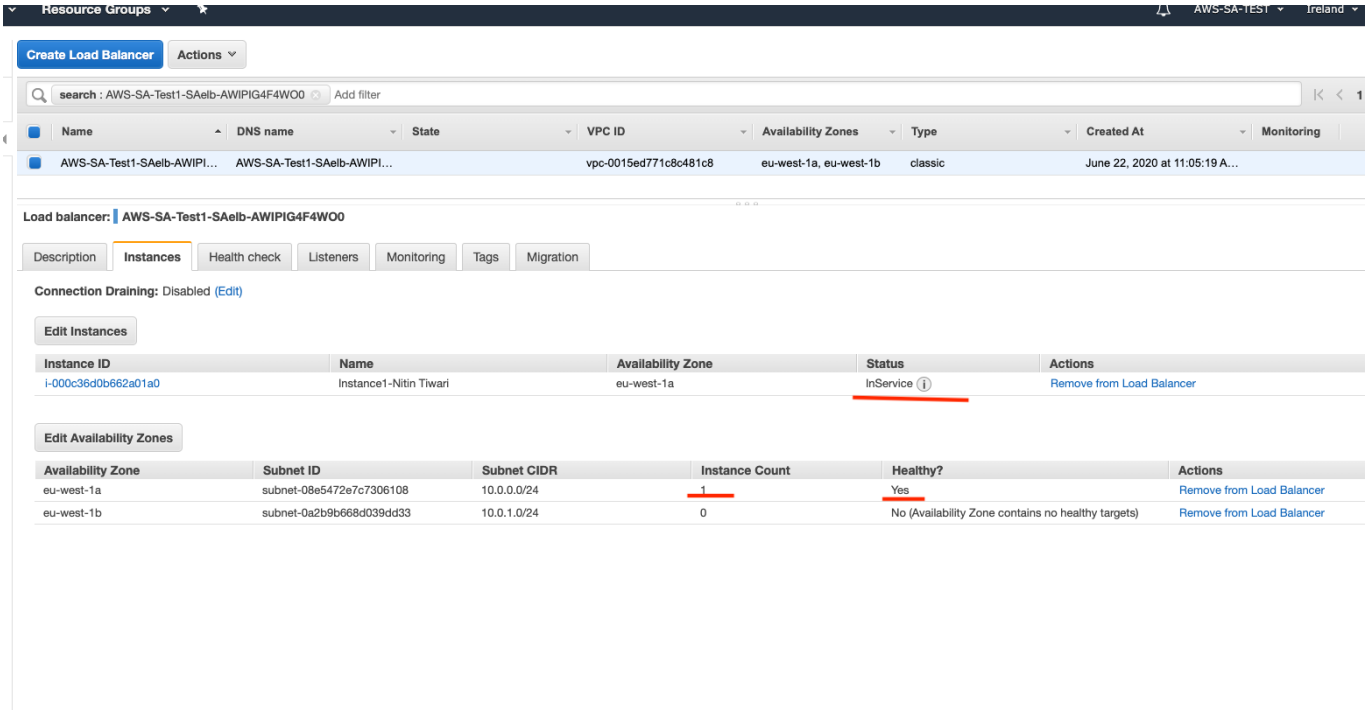
Load Balancer fails Health Check, it refers to TCP:443, while instance isn't configured on port 443, used commonly for secured HTTP connection.



Solution: Update the health check page, as below:



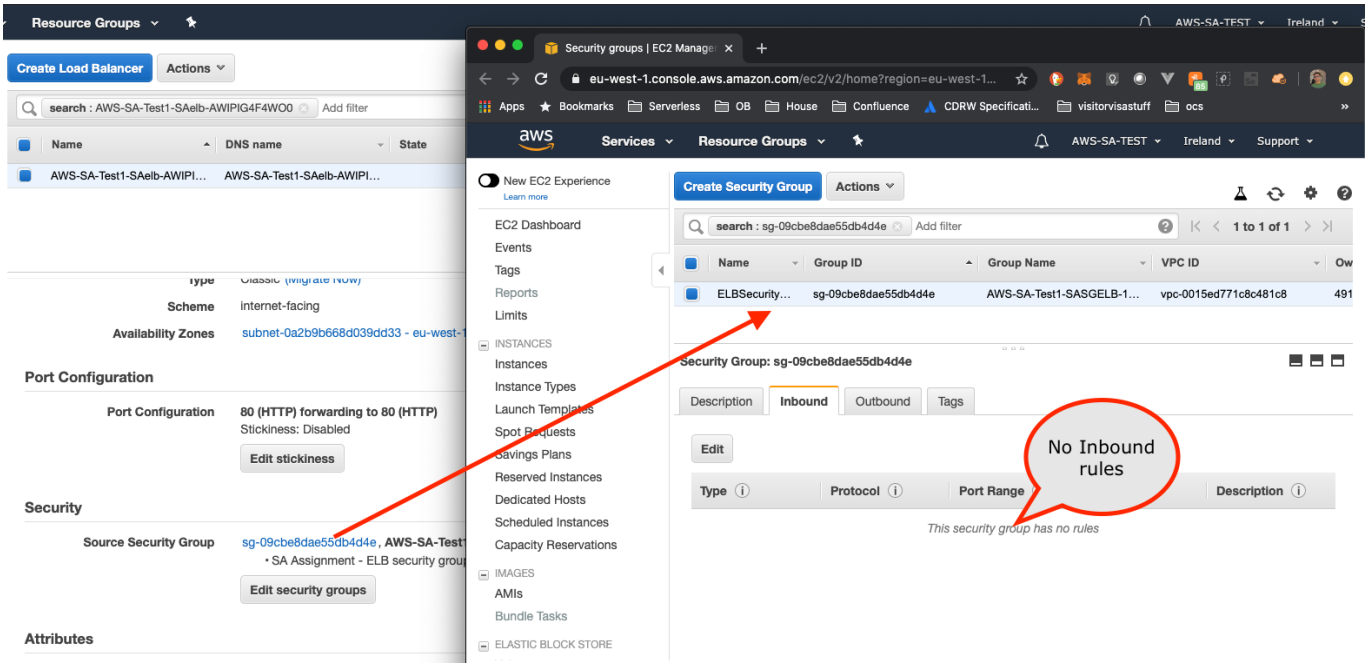
- This step resolves the Instance configuration service, however the website still doesn't load.



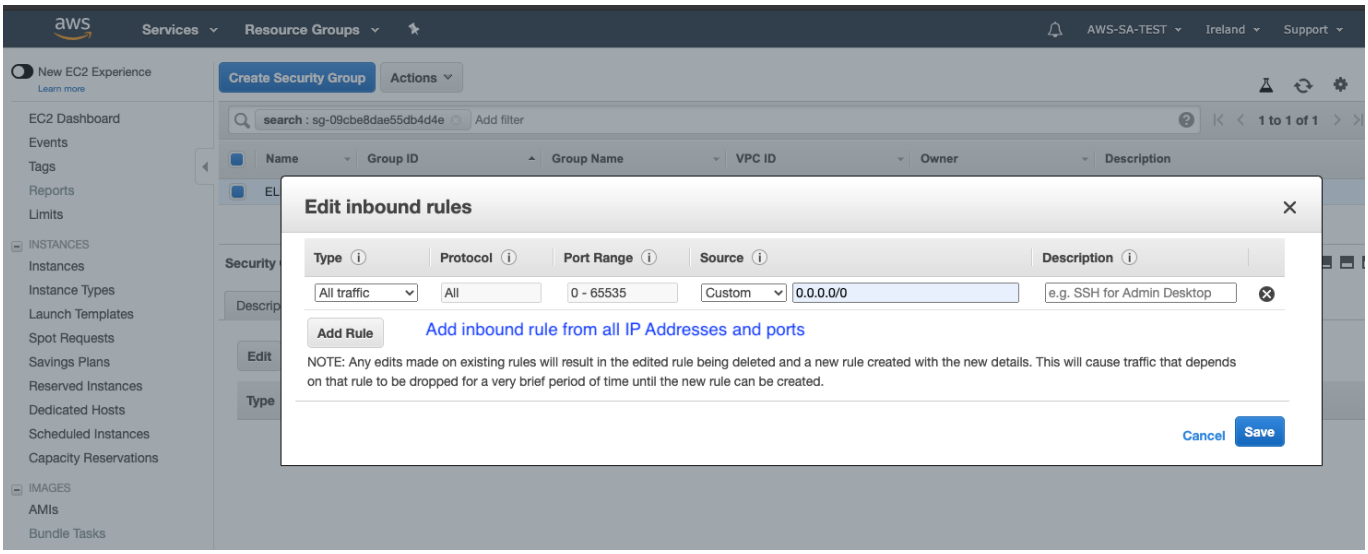
Next step is to check the Security group configurations.

Problem - 3

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



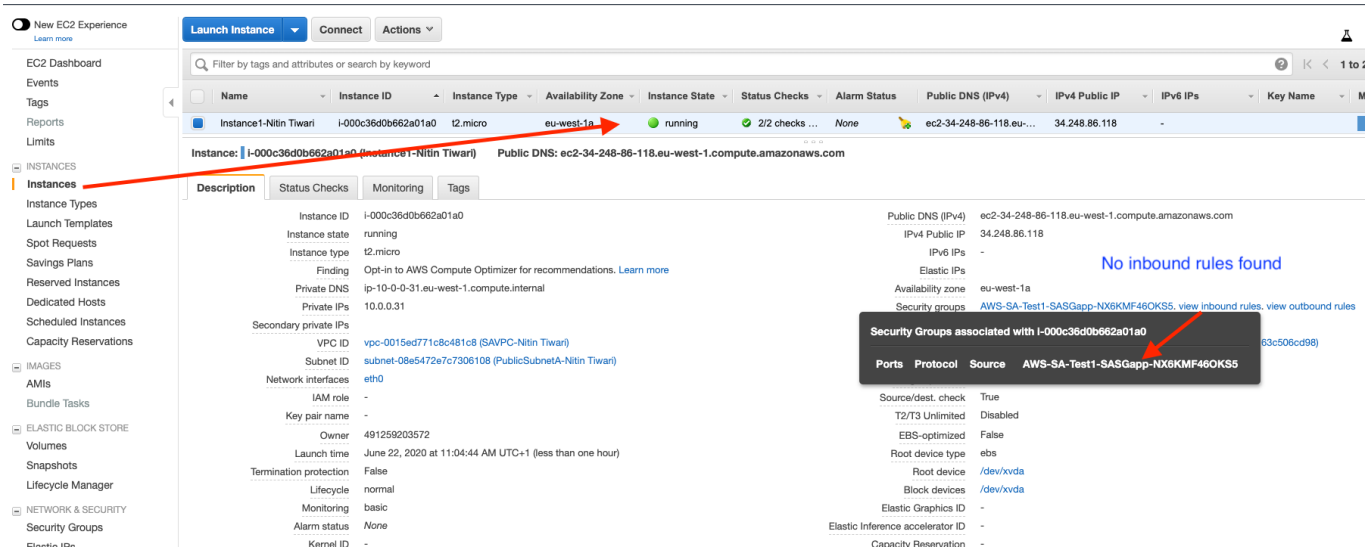
Solution: Add a new inbound rule to allow incoming traffic from All IP and Ports.



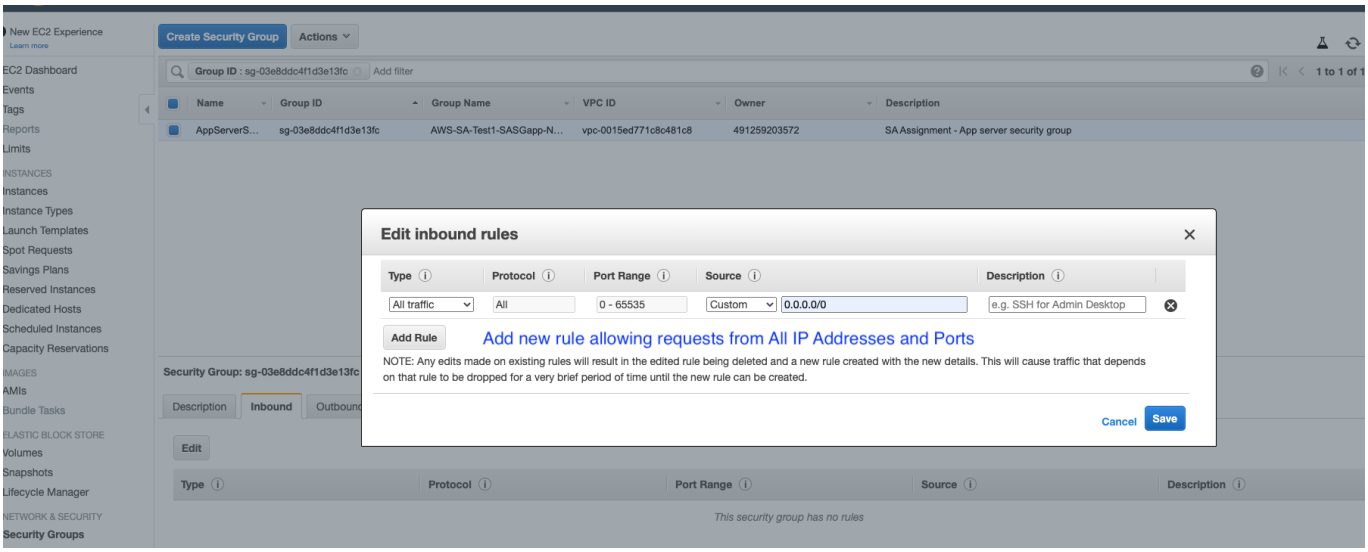
- Next step is to check the rules for the EC2 instance and same problem as above was encountered - No inbound traffic rule was found

Problem - 4

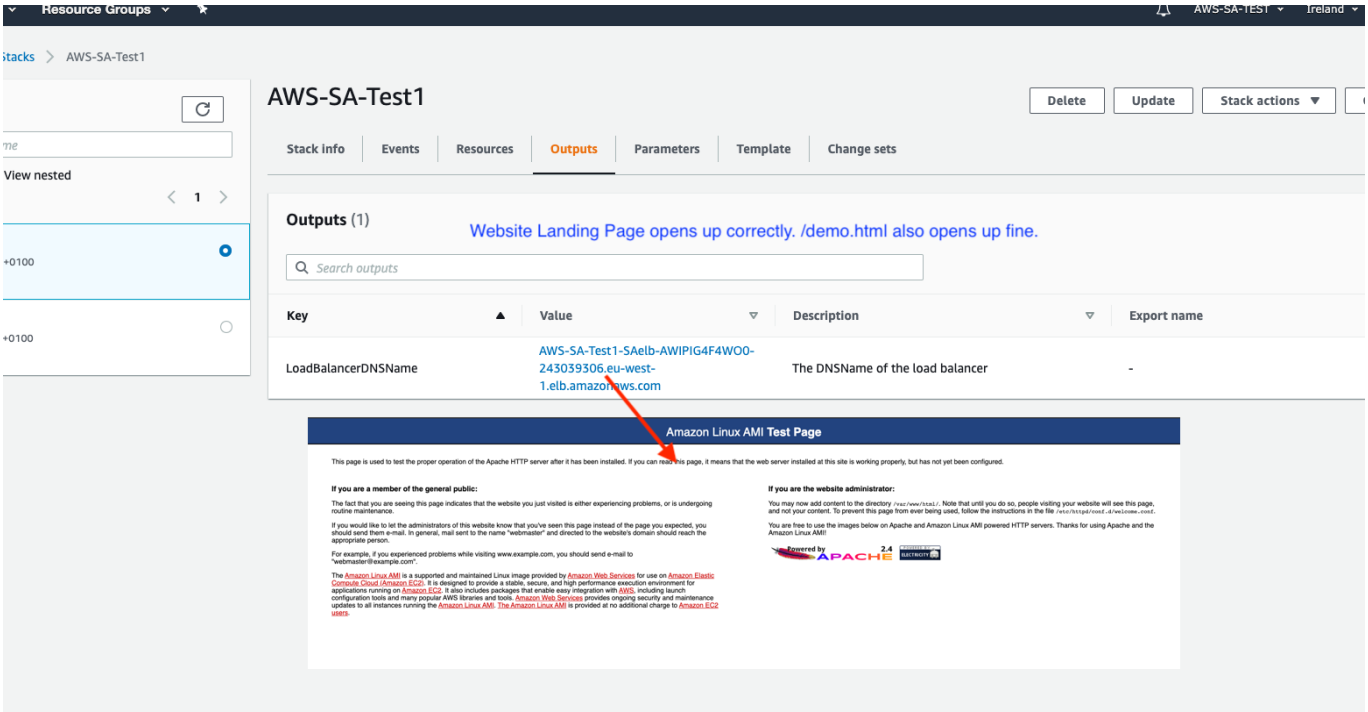
Instance doesn't allow inbound requests.



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

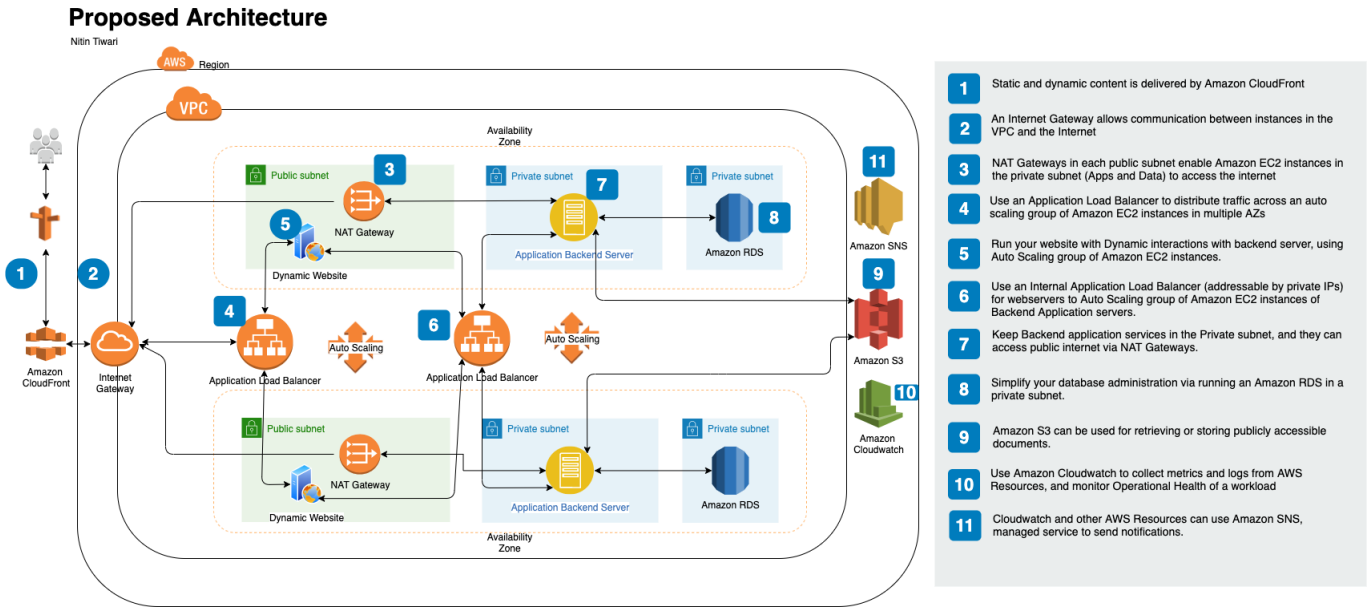
```
    SASGAPPINGRESS:
      Type: AWS::EC2::SecurityGroupIngress
      Properties:
        CidrIp: 0.0.0.0/0
        Description: Inbound rule
        FromPort: '-1'
        GroupId: !GetAtt SASGapp.GroupId
        IpProtocol: '-1'
        ToPort: 80
```

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 Networl lebel 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 Ibsite 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 Webservers 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 pubicly accessible document.
    - Amzon 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 - primarly 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 princples 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 sustem to run and deliover business value and continually improve supporting processes and proecedures.

Security

- Solution must enure secure connection to the Webservers only allowing traffic on http. 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 unauthorize access, or threats.
- Provision of public and private subnets and corresponding Access Control Layer protects the network.
- Security groups protects 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 oprimize resource utilization and eventurally provide better perfiornabce.

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 Eplorer, and AWS Budgets that notify you if your usage or spend exceeds actual or forecast budgeted amounts.
- Optimize resources over time.