

# Project-4 - HAWA - CloudFormation

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Parameters:
```

```
LatestAmiId:
```

```
  Description: Region specific image from the Parameter Store
```

```
  Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
```

```
  Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

```
InstanceType:
```

```
  Description: Amazon EC2 instance type for the instances
```

```
  Type: String
```

```
  AllowedValues:
```

- t3.micro
- t3.small
- t3.medium

```
  Default: t3.micro
```

```
Resources:
```

```
# ----- VPC -----
```

```
HAWAVPC:
```

```
  Type: AWS::EC2::VPC
```

```
  Properties:
```

```
    CidrBlock: 10.0.0.0/16
```

```
    EnableDnsSupport: true
```

```
    EnableDnsHostnames: true
```

```
  Tags:
```

- Key: Name
- Value: HAWA VPC

```
# ----- Subnet -----
```

```
PublicSubnet1:
```

```
  Type: AWS::EC2::Subnet
```

```
  Properties:
```

```
    VpcId: !Ref HAWAVPC
```

```
    CidrBlock: 10.0.1.0/24
```

```
    AvailabilityZone: us-east-1a
```

```
    MapPublicIpOnLaunch: true
```

```
  Tags:
```

- Key: Name
- Value: PublicSubnet1

```
PrivateSubnet1:
```

```
  Type: AWS::EC2::Subnet
```

```
  Properties:
```

```
    VpcId: !Ref HAWAVPC
```

```
    CidrBlock: 10.0.2.0/24
```

```
    AvailabilityZone: us-east-1a
```

```
  Tags:
```

- Key: Name
- Value: PrivateSubnet1

```
PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC
    CidrBlock: 10.0.3.0/24
    AvailabilityZone: us-east-1b
    MapPublicIpOnLaunch : true
    Tags:
      - Key: Name
        Value: PublicSubnet2

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC
    CidrBlock: 10.0.4.0/24
    AvailabilityZone: us-east-1b
    Tags:
      - Key: Name
        Value: PrivateSubnet2

# ----- Gateway -----
NATGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP1.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: NATGateway1

EIP1:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

NATGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP2.AllocationId
    SubnetId: !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: NATGateway2

EIP2:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

InternetGateway:
  Type: AWS::EC2::InternetGateway

AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
```

```

    VpcId: !Ref HAWAVPC
    InternetGatewayId: !Ref InternetGateway

# ----- Public Route Table -----
PublicRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Public Route Table 1

PublicRoute1:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet1
    RouteTableId: !Ref PublicRouteTable1

PublicRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Public Route Table 2

PublicRoute2:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet2
    RouteTableId: !Ref PublicRouteTable2

# ----- Private Route Table -----
PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name

```

Value: Private Route Table 1

PrivateRoute1:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable1

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NATGateway1

PrivateSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet1

RouteTableId: !Ref PrivateRouteTable1

PrivateRouteTable2:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref HAWAVPC

Tags:

- Key: Name

Value: Private Route Table 2

PrivateRoute2:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable2

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NATGateway2

PrivateSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet2

RouteTableId: !Ref PrivateRouteTable2

# ----- Instance -----

KeyPair:

Type: AWS::EC2::KeyPair

Properties:

KeyName: hawaInstanceKey

MyInstanceSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security group for EC2 instances

VpcId: !Ref HAWAVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 80

ToPort: 80

```
    SourceSecurityGroupId: !Ref MyALBSG
  Tags:
    - Key: Name
      Value: HAWAInstanceSG

# ----- ELB -----
#https://dev.classmethod.jp/articles/cloudformation-template-for-creating-ec2-with-l
MyALBSG:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for ALB
    VpcId: !Ref HAWAVPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: HAWAALBSG

MyALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2
    SecurityGroups:
      - !Ref MyALBSG
    Scheme: internet-facing
    LoadBalancerAttributes:
      - Key: idle_timeout.timeout_seconds
        Value: '60'

MyTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: MyTargetGroup
    Port: 80
    Protocol: HTTP
    TargetType: instance
    VpcId: !Ref HAWAVPC

MyListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref MyTargetGroup
    LoadBalancerArn: !Ref MyALB
    Port: 80
    Protocol: HTTP
```

```

# ----- ASG -----
#https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-autoscaling-launch-template.html
#https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-autoscaling-launch-template.html

myLaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref MyInstanceSG
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          echo "<h1>Hello World from $(hostname -f)</h1>" > /var/www/html/index.html
          systemctl start httpd
          systemctl enable httpd

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: "3"
    MinSize: "1"
    DesiredCapacity: "2"
    TargetGroupARNs:
      - !Ref MyTargetGroup
    VPCZoneIdentifier:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2

```

## VPC 2p2p

```

Resources:
# ----- VPC -----
HAWAVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: true
    EnableDnsHostnames: true
# ----- Subnet -----
PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC

```

```
CidrBlock: 10.0.1.0/24
AvailabilityZone: us-east-1a

PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC
    CidrBlock: 10.0.2.0/24
    AvailabilityZone: us-east-1a

PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC
    CidrBlock: 10.0.3.0/24
    AvailabilityZone: us-east-1b

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref HAWAVPC
    CidrBlock: 10.0.4.0/24
    AvailabilityZone: us-east-1b
# ----- Gateway -----
NATGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP1.AllocationId
    SubnetId: !Ref PublicSubnet1

EIP1:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

NATGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP2.AllocationId
    SubnetId: !Ref PublicSubnet2

EIP2:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

InternetGateway:
  Type: AWS::EC2::InternetGateway

AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId: !Ref HAWAVPC
    InternetGatewayId: !Ref InternetGateway
```

```

# ----- Public Route Table -----
PublicRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Public Route Table 1

PublicRoute1:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet1
    RouteTableId: !Ref PublicRouteTable1

PublicRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Public Route Table 2

PublicRoute2:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet2
    RouteTableId: !Ref PublicRouteTable2

# ----- Private Route Table -----
PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Private Route Table 1

PrivateRoute1:

```



```

Type: AWS::EC2::Route
Properties:
  RouteTableId: !Ref PrivateRouteTable1
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NATGateway1

PrivateSubnet1RouteTableAssociation:
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  SubnetId: !Ref PrivateSubnet1
  RouteTableId: !Ref PrivateRouteTable1

PrivateRouteTable2:
Type: AWS::EC2::RouteTable
Properties:
  VpcId: !Ref HAWAVPC
  Tags:
    - Key: Name
      Value: Private Route Table 2

PrivateRoute2:
Type: AWS::EC2::Route
Properties:
  RouteTableId: !Ref PrivateRouteTable2
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NATGateway2

PrivateSubnet2RouteTableAssociation:
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  SubnetId: !Ref PrivateSubnet2
  RouteTableId: !Ref PrivateRouteTable2

```

## HAWA Project - with init instances

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Parameters:
```

```
LatestAmiId:
```

```
  Description: Region specific image from the Parameter Store
```

```
  Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
```

```
  Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

```
InstanceType:
```

```
  Description: Amazon EC2 instance type for the instances
```

```
  Type: String
```

```
  AllowedValues:
```

- t3.micro
- t3.small
- t3.medium

```
  Default: t3.micro
```

Resources:

# ----- VPC -----

HAWAVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/16

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: HAWA VPC

# ----- Subnet -----

PublicSubnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref HAWAVPC

CidrBlock: 10.0.1.0/24

AvailabilityZone: us-east-1a

MapPublicIpOnLaunch : true

Tags:

- Key: Name

Value: PublicSubnet1

PrivateSubnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref HAWAVPC

CidrBlock: 10.0.2.0/24

AvailabilityZone: us-east-1a

Tags:

- Key: Name

Value: PrivateSubnet1

PublicSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref HAWAVPC

CidrBlock: 10.0.3.0/24

AvailabilityZone: us-east-1b

MapPublicIpOnLaunch : true

Tags:

- Key: Name

Value: PublicSubnet2

PrivateSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref HAWAVPC

CidrBlock: 10.0.4.0/24

AvailabilityZone: us-east-1b

Tags:

- Key: Name

Value: PrivateSubnet2

# ----- Gateway -----

```

NATGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP1.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: NATGateway1

EIP1:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

NATGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt EIP2.AllocationId
    SubnetId: !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: NATGateway2

EIP2:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc

InternetGateway:
  Type: AWS::EC2::InternetGateway

AttachGateway:
  Type: AWS::EC2::VPCEGatewayAttachment
  Properties:
    VpcId: !Ref HAWAVPC
    InternetGatewayId: !Ref InternetGateway

# ----- Public Route Table -----
PublicRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
    Tags:
      - Key: Name
        Value: Public Route Table 1

PublicRoute1:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:

```

```

Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  SubnetId: !Ref PublicSubnet1
  RouteTableId: !Ref PublicRouteTable1

PublicRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
  Tags:
    - Key: Name
      Value: Public Route Table 2

PublicRoute2:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet2
    RouteTableId: !Ref PublicRouteTable2

# ----- Private Route Table -----
PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
  Tags:
    - Key: Name
      Value: Private Route Table 1

PrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NATGateway1

PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PrivateSubnet1
    RouteTableId: !Ref PrivateRouteTable1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref HAWAVPC
  Tags:
    - Key: Name

```

Value: Private Route Table 2

PrivateRoute2:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable2

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NATGateway2

PrivateSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet2

RouteTableId: !Ref PrivateRouteTable2

# ----- Instance -----

KeyPair:

Type: AWS::EC2::KeyPair

Properties:

KeyName: hawaInstanceKey

MyInstanceSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security group for EC2 instances

VpcId: !Ref HAWAVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 80

ToPort: 80

SourceSecurityGroupId: !Ref MyALBSG

Tags:

- Key: Name

Value: HAWAInstanceSG

MyEC2Instance1:

Type: AWS::EC2::Instance

Properties:

InstanceType: t2.micro

ImageId: ami-06c68f701d8090592

SubnetId: !Ref PublicSubnet1

KeyName: hawaInstanceKey

BlockDeviceMappings:

- DeviceName: /dev/sdf

Ebs:

VolumeSize: 10

SecurityGroupIds:

- !Ref MyInstanceSG

UserData:

Fn::Base64: !Sub |

#!/bin/bash

```
    yum update -y
    yum install -y httpd
    echo '<h1>Hello World 1</h1>' > /var/www/html/index.html
    systemctl start httpd
    systemctl enable httpd
```

Tags:

- Key: Name
- Value: HAWAInstance1

MyEC2Instance2:

Type: AWS::EC2::Instance

Properties:

InstanceType: t2.micro  
ImageId: ami-06c68f701d8090592  
SubnetId: !Ref PublicSubnet2  
KeyName: hawaInstanceKey  
BlockDeviceMappings:

- DeviceName: /dev/sdf

Ebs:

VolumeSize: 10

SecurityGroupIds:

- !Ref MyInstanceSG

UserData:

Fn::Base64: !Sub |  
 *#!/bin/bash*  
 yum update -y  
 yum install -y httpd  
 echo '<h1>Hello World 2</h1>' > /var/www/html/index.html  
 systemctl start httpd  
 systemctl enable httpd

Tags:

- Key: Name
- Value: HAWAInstance2

# ----- ELB -----  
*#https://dev.classmethod.jp/articles/cloudformation-template-for-creating-ec2-with-l*

MyALBSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security group for ALB  
VpcId: !Ref HAWAVPC  
SecurityGroupIngress:

- IpProtocol: tcp  
FromPort: 80  
ToPort: 80  
CidrIp: 0.0.0.0/0

Tags:

- Key: Name
- Value: HAWAALBSG

MyALB:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Subnets:

- !Ref PublicSubnet1

```

    - !Ref PublicSubnet2
  SecurityGroups:
    - !Ref MyALBSG
  Scheme: internet-facing
  LoadBalancerAttributes:
    - Key: idle_timeout.timeout_seconds
      Value: '60'

```

```

MyTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: MyTargetGroup
    Port: 80
    Protocol: HTTP
    TargetType: instance
    VpcId: !Ref HAWAVPC
    Targets:
      - Id: !Ref MyEC2Instance1
        Port: 80
      - Id: !Ref MyEC2Instance2
        Port: 80

```

```

MyListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref MyTargetGroup
    LoadBalancerArn: !Ref MyALB
    Port: 80
    Protocol: HTTP

```

```

# ----- ASG -----
#https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-autoscc
#https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-autoscc

```

```

myLaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref MyInstanceSG
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          echo '<h1>Hello World 2</h1>' > /var/www/html/index.html
          systemctl start httpd
          systemctl enable httpd

```

```

AutoScalingGroup:

```

```
Type: AWS::AutoScaling::AutoScalingGroup
Properties:
  LaunchTemplate:
    LaunchTemplateId: !Ref myLaunchTemplate
    Version: !GetAtt myLaunchTemplate.LatestVersionNumber
  MaxSize: "3"
  MinSize: "1"
  DesiredCapacity: "2"
  TargetGroupARNs:
    - !Ref MyTargetGroup
  VPCZoneIdentifier:
    - !Ref PublicSubnet1
    - !Ref PublicSubnet2
```