Hands-on Route 53-01

Overwiew

- Part 1 - Pre-Creating
- Part 2 - Getting familiar with Public Hosted Zone, SOA, NS.
- Part 3 - Creating A record set
- Part 4 - Creating CNAME record set
- Part 4 - Creating an Alias record set

Part 1 - Pre-Creating

STEP 1: Create Sec.Group:

```
Route 53 Sec: In bound : "SSH 22, HTTP 80 -------> anywhere(0:/00000)"
```

STEP 2: Create Instances:

- We'll totally create "2" instances.

1. Create EC2 that is installed httpd user data in default VPC named "N.virginia_1"

```
Region: "N.Virginia"
VPC: Default VPC
Subnet: PublicA
Sec Group: "Route 53 Sec"

user data:

#! /bin/bash
yum update -y
amazon-linux-extras install nginx1.12
yum install git -y
systemctl start nginx
cd /usr/share/nginx/html
git clone https://github.com/techproedu/designer.git
chmod -R 777 /usr/share/nginx/html
rm index.html
cp -R ./designer/. .
systemctl restart nginx
systemctl enable nginx
```

2. Create EC2 that is installed httpd user data in default VPC "N.virginia_2"

```
Region: "N.Virginia"
VPC: Default VPC
Subnet: PublicA
Sec Group: "Route 53 Sec"

user data:

#! /bin/bash
yum update -y
amazon-linux-extras install nginx1.12
yum install git -y
systemctl start nginx
cd /usr/share/nginx/html
git clone https://github.com/techproedu/norhvirginia.git
chmod -R 777 /usr/share/nginx/html
rm index.html
cp -R ./norhvirginia/. .
systemctl restart nginx
systemctl enable nginx
```

STEP 3: Create Static WebSite Hosting :

1. Create Static WebSite Hosting-1/ "www.[your sub-domain name].net"

- Go to S3 service and create a bucket with sub-domain name: "www.[your sub-domain name].net"
- Public Access "Enabled"
- Upload Files named "index.html" and "eagle.jpg" in "s3.bucket.www" folder
- Permissions>>> Bucket Policy >>> Paste bucket Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::don't forget to change me/*"
        }
    ]
}
```

STEP 4: Create Auto Scaling and ALB with CloudFormation Template :

- Go the Cloudformation console and create a stack based on the template seen below.
- Prefer not to copy code, instead use upload from your computer.

```
AWSTemplateFormatVersion: 2010-09-09
Description: |
  This is a demo template.
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
  Subnets:
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t1.micro
      - m1.small
      - m1.medium
      - m1.large
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: The EC2 Key Pair to allow SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
  PolicyTargetValue:
    Description: Please enter your Target value that triggers the Autoscaling
    Default: '60'
    Type: String

Mappings:
  RegionImageMap:
    us-east-1:
      AMI: ami-0c2b8ca1dad447f8a
    us-east-2:
      AMI: ami-0443305dabd4be2bc
    us-west-1:
      AMI: ami-04b6c97b14c54de18
    us-west-2:
      AMI: ami-083ac7c7ecf9bb9b0
    eu-west-1:
      AMI: ami-02b4e72b17337d6c1

Resources:
  myAutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      AvailabilityZones: !GetAZs
      LaunchConfigurationName: !Ref myLaunchConfig
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
      MinSize: '2'
      MaxSize: '3'
      TargetGroupARNs:
```

```
        - !Ref myALBTargetGroup

  myLaunchConfig:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      KeyName: !Ref KeyName
      ImageId: !FindInMap
        - RegionImageMap
        - !Ref AWS::Region
        - AMI
      UserData: !Base64 |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
          && PRIVATE_IP=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
http://169.254.169.254/latest/meta-data/local-ipv4`
          TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
          && PUBLIC_IP=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
http://169.254.169.254/latest/meta-data/public-ipv4`
          DATE_TIME=`date`
          chmod -R 777 /var/www/html
          echo "<html>
          <head>
              <title> Congratulations! You have created an instance from Launch
Template</title>
          </head>
          <body>
              <h1>This web server is launched from launch configuration by
YOUR_NAME</h1>
              <p>This instance is created at <b>$DATE_TIME</b></p>
              <p>Private IP address of this instance is <b>$PRIVATE_IP</b></p>
              <p>Public IP address of this instance is <b>$PUBLIC_IP</b></p>
          </body>
          </html>" > /var/www/html/index.html
          systemctl start httpd
          systemctl enable httpd
      SecurityGroups:
        - !Ref mySecurityGroup
      InstanceType: !Ref InstanceType

  myCPUPolicy:
    Type: "AWS::AutoScaling::ScalingPolicy"
    Properties:
      AutoScalingGroupName: !Ref myAutoScalingGroup
      PolicyType: TargetTrackingScaling
      TargetTrackingConfiguration:
        PredefinedMetricSpecification:
          PredefinedMetricType: ASGAverageCPUUtilization
        TargetValue: !Ref PolicyTargetValue
```

```
  myApplicationLoadBalancer:
    Type: "AWS::ElasticLoadBalancingV2::LoadBalancer"
    Properties:
      SecurityGroups:
        - !GetAtt mySecurityGroup.GroupId
      Subnets: !Ref Subnets



  myALBListener:
    Type: "AWS::ElasticLoadBalancingV2::Listener"
    Properties:
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myALBTargetGroup
      LoadBalancerArn: !Ref myApplicationLoadBalancer
      Port: 80
      Protocol: HTTP


  myALBTargetGroup:
    Type: "AWS::ElasticLoadBalancingV2::TargetGroup"
    Properties:
      HealthCheckIntervalSeconds: 25
      HealthCheckTimeoutSeconds: 5
      HealthyThresholdCount: 3
      Port: 80
      Protocol: HTTP
      UnhealthyThresholdCount: 3
      VpcId: !Ref VpcId


  mySecurityGroup:
    Type: "AWS::EC2::SecurityGroup"
    Properties:
      GroupDescription: Enables SSH and HTTP
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0



Outputs:
  URL:
    Description: The URL of the website
    Value: !Join
      - ''
```

```
        - - 'http://'
          - !GetAtt myApplicationLoadBalancer.DNSName
```

Part 2 - Getting familiar with Route 53 Public Hosted Zone, SOA, NS.

- Explain that Public hosted Zone and permanent records SOA nad NS.

Part 3 - Creating A Record Sets

STEP 1 : Create A Record with "www" subdomain:

- Go to Route 53 service
- Click hosted zones on the left hand menu
- click your Domain name's public hosted zone
- click "create record"
- select "simple routing" ---> Next
- click "Define simple record"
- Create A record with N. Virginia_1

```
Record Name:"www"
Value/Route traffic to:
  - select "IP address or another value depending the record type" option
      - enter IP of the "N.Virginia_1" EC2
Record Type : A
Type: "A – IPv4 address"
Alias:"No"
TTL:"1m"
```

- Select newly created record's flag and hit the "create record" tab seen bottom

STEP 2 : Create another "A record" with N. Virginia_1 with "info" subdomain

- Go to Route 53 service
- Click hosted zones on the left hand menu
- click your Domain name's public hosted zone
- click "create record"
- select "simple routing" ---> Next
- click "Define simple record"
- Create A record with N. Virginia_1

```
Record Name:"info"
Value/Route traffic to:
  - select "IP address or another value depending the record type" option
      - enter IP of the "N.Virginia_1" EC2
Record Type : A
TTL:"1m"
```

- Select newly created record's flag and hit the "create record" tab seen bottom
- After show "info.[your DNS name].net" on the browser, "Delete" this record

STEP 3: Add another IP (N. Virginia_2) to the existing "A record"

- select "www.[your DNS name].net" A-record ---> Edit

```
Name:"www"
Value/Route traffic to:
    "IP of N.Virginia_1" ,and
    "IP of N.Virginia_2"
```

- Check from local terminal nslookup www.[your DNS name].net an show two IP address

Part 4 - Creating a CNAME Record:

- Add CNAME record for "Domain Name"
- click your Domain name's public hosted zone
- click "create record"
- select "simple routing" ---> Next
- click "Define simple record"

```
Record Name:"showcname"
Value/Route traffic to:
  - "IP address or another value depending on the record type"
    - enter "www.[your DNS name].net"
Record Type : "CNAME"-Routes to another domain and some AWS resources
TTL:"1m"
```

- hit the define simple record
- Select newly created record's flag and hit the "create record" tab seen bottom
- After show "showcname.[your DNS name].net" on the browser. It will reflects the "www.[your DNS name].net". After that "Delete" this record

Part 5 - Creating a Alias Record:

STEP 1: Alias for S3 bucket

- click your Domain name's public hosted zone
- click "create record"
- select "simple routing" ---> Next
- click "Define simple record"

```
Record Name:"www"
Value/Route traffic to:
    - Alias to *****S3 Website Endpoint*******
    - US East (N.Virginia) [us-east-1]
    - choose your S3 bucket named "www.[your DNS name].net"
Record Type : A
```

- hit the define simple record
- Select newly created record's flag and hit the "create record" tab seen bottom
- go to the target domain name "www.[your DNS name].net" on browser
- show the content of web page. It is the same as S3 static web hosting page.
- ***Delete the Alias record

STEP 2: Alias for ELB bucket

- click your Domain name's public hosted zone
- click "create record"
- select "simple routing" ---> Next
- click "Define simple record"

```
Record Name:"www"
Value/Route traffic to:
    - Alias to ***** ELB Endpoint*******
    - US East (N.Virginia) [us-east-1]
    - choose your S3 bucket named "www.[your DNS name].net"
Record Type : A
```

- hit the define simple record
- Select newly created record's flag and hit the "create record" tab seen bottom

- go to the target domain name "www.[your DNS name].net" on browser
- show the content of web page. It is the same as S3 static web hosting page.

STEP 3: Cleaning

- Delete CloudFormation Stack
- Delete Instances.
- Delete bucket
- Delete A,CNAME and Alias record If they exist.
- Check the load balancer again