



ISSN: 2321-2152

IJMECE

*International Journal of modern
electronics and communication engineering*

E-Mail

editor.ijmece@gmail.com

editor@ijmece.com

www.ijmece.com

Implementation of Route 53 and load balancer with associated security in Amazon Web services

G. POSHAMALLU

Electronics and Communication Engineering, St.Martin's Engineering College,Dhulapally,Secundrabad,India

Email: gposhamalluece@smec.ac.in

Abstract— Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS

With AWS Certificate Manager, there is no additional charge for provisioning public or private SSL/TLS certificates you use with ACM-integrated services, such as Elastic Load Balancing and API Gateway. You pay for the AWS resources you create to run your application. For private certificates, AWS Private CA provides you the ability to pay monthly for the service and certificates you create. You pay less per certificate as you create more private certificates

using GoDaddy a domain name is Poshamallu.cloud the main aim of my project I want give security for my domain for that i used route 53 from that route 53 added ns records to GoDaddy added the ns records .created a ec2 instance and installed Apache server added some content in my domain name .and created SSL certification for giving security to my domain name using a amazon certification manger ,created an application load balancer for giving security to my domain naming .created a record for a adding loadbalncer in route 53.whatever the traffic will come to HTTP routed to https port number 443

With AWS Certificate Manager, there is no additional charge for provisioning public or private SSL/TLS certificates you use with ACM-integrated services, such as Elastic Load Balancing and API Gateway. You pay for the AWS resources you create to run your application. For private certificates, AWS Private CA provides you the ability to pay monthly for the service and certificates you create. You pay less per certificate as you create more private certificates With a few clicks in the AWS Management Console, you can request a trusted SSL/TLS certificate

from AWS. Once the certificate is created, AWS Certificate Manager takes care of deploying certificates to help you enable SSL/TLS for your website or application.

Keywords: Virtual private cloud, EC2, GoDaddy. Amazon certification manger. rout53. Application load balancer.

I. INTRODUCTION

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

1.1 Features

The following features help you configure a VPC to provide the connectivity that your applications need:

1.1.1 Virtual private clouds (VPC)

A VPC is a virtual network that closely resembles a traditional network that you'd operate in your own data center. After you create a VPC, you can add subnets.

1.1.2 Subnets

A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.

1.1.3 IP addressing

You can assign IPv4 addresses and IPv6 addresses to your VPCs and subnets. You can also bring your public IPv4 and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.

1.1.4 Routing

Use route tables (p. 78) to determine where network traffic from your subnet or gateway is directed

1.1.5 Gateways and endpoints

A gateway (p. 131) connects your VPC to another network. For example, use an internet gateway (p. 131) to connect your VPC to the internet. Use a VPC endpoint to connect to AWS services privately, without the use of an internet gateway or NAT device.

1.2 Peering connections

Use a VPC peering connection to route traffic between the resources in two VPCs.

1.2.1 Traffic Mirroring

Copy network traffic from network interfaces and send it to security and monitoring appliances for deep packet inspection.

1.2.2 Transit gateways

Use a transit gateway which acts as a central hub, to route traffic between your VPCs, VPN connections, and AWS Direct Connect connections.

1.2.3 VPC Flow Logs

A flow log captures information about the IP traffic going to and from network interfaces in your VPC.

1.2.4 VPN connections

Connect your VPCs to your on-premises networks using AWS Virtual Private Network (AWS VPN)

1.3 Getting started with Amazon VPC

Your AWS account includes a default VPC in each AWS Region. Your default VPCs are configured such that you can immediately start launching and connecting to EC2 instances

You can choose to create additional VPCs with the subnets, IP addresses, gateways and routing that you need.

1.3.1 Working with Amazon VPC

You can create and manage your VPCs using any of the following interfaces:

AWS Management Console — Provides a web interface that you can use to access your VPCs.

AWS Command Line Interface (AWS CLI) — Provides commands for a broad set of AWS services, including Amazon VPC, and is supported on Windows, Mac, and Linux.

AWS SDKs — Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, handling request retries, and error handling.

Query API — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Amazon VPC, but it requires that your application handle low-level details such as generating the hash to sign the request, and error handling.

Pricing for Amazon VPC

There's no additional charge for using a VPC. There are charges for some VPC components, such as NAT gateways, IP Address Manager, traffic mirroring, Reachability Analyzer, and Network Access Analyzer.

1.4. How Amazon VPC works

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in

your own data center, with the benefits of using the scalable infrastructure of AWS.

Concepts

- VPCs and subnets
- Default and nondefault VPCs
- IP addressing
- Route tables
- Access the internet
- Access a corporate or home network
- Connect VPCs and networks
- AWS private global network considerations

1.4.1 VPCs and subnets

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can specify an IP address range for the VPC, add subnets, add gateways, and associate security groups.

A subnet is a range of IP addresses in your VPC. You launch AWS resources, such as Amazon EC2 instances, into your subnets. You can connect a subnet to the internet, other VPCs, and your own data centers, and route traffic to and from your subnets using route tables.

1.4.2 Default and nondefault VPCs

If your account was created after 2013-12-04, it comes with a default VPC in each Region. A default VPC is configured and ready for you to use. For example, it has a default subnet in each Availability Zone in the Region, an attached internet gateway, a route in the main route table that sends all traffic to the internet gateway, and DNS settings that automatically assign public DNS hostnames to instances with public IP addresses and enable DNS resolution through the Amazon-provided DNS server. Therefore, an EC2 instance that is launched in a default subnet automatically has access to the internet.

If you have a default VPC in a Region and you don't specify a subnet when you launch an EC2 instance into that Region, we choose one of the default subnets and launch the instance into that subnet.

1.4.3 IP addressing

IP addresses enable resources in your VPC to communicate with each other, and with resources over the internet.

Classless Inter-Domain Routing (CIDR) notation is a way of representing an IP address and its network mask. The format of these addresses is as follows:

- An individual IPv4 address is 32 bits, with 4 groups of up to 3 decimal digits. For example, 10.0.1.0.
- An IPv4 CIDR block has four groups of up to three decimal digits, 0-255, separated by periods, followed by a slash and a number from 0 to 32. For example, 10.0.0.0/16.

- An individual IPv6 address is 128 bits, with 8 groups of 4 hexadecimal digits. For example, 2001:0db8:85a3:0000:0000:8a2e:0370:7334.
- An IPv6 CIDR block has four groups of up to four hexadecimal digits, separated by colons, followed by a double colon, followed by a slash and a number from 1 to 128. For example, 2001:db8:1234:1a00::/56.

When you create a VPC, you assign it an IPv4 CIDR block (a range of private IPv4 addresses), an IPv6 CIDR block, or both IPv4 and IPv6 CIDR blocks (dual-stack).

Private IPv4 addresses are not reachable over the internet. IPv6 addresses are globally unique and can be configured to remain private or be reachable over the internet.

Your VPC can operate in dual-stack mode. This means that your resources can communicate over IPv4, IPv6, or both IPv4 and IPv6. IPv4 and IPv6 addresses are independent of each other; you must add separate routes and security group rules for IPv4 and IPv6.

1.4.4 Private IPv4 addresses

Private IPv4 addresses (also referred to as private IP addresses in this topic) are not reachable over the internet, and can be used for communication between the instances in your VPC. When you launch an instance into a VPC, a primary private IP address from the IPv4 address range of the subnet is assigned to the default network interface (eth0) of the instance. Each instance is also given a private (internal) DNS hostname that resolves to the private IP address of the instance. The hostname can be of two types: resource-based or IP-based. For more information, see EC2 instance naming. If you don't specify a primary private IP address, we select an available IP address in the subnet range for you. For more information about network interfaces, see Elastic Network Interfaces in the Amazon EC2 User Guide for Linux Instances.

You can assign additional private IP addresses, known as secondary private IP addresses, to instances that are running in a VPC. Unlike a primary private IP address, you can reassign a secondary private IP address from one network interface to another. A private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is

terminated. For more information about primary and secondary IP addresses, see Multiple IP Addresses in the Amazon EC2 User Guide for Linux Instances.

We refer to private IP addresses as the IP addresses that are within the IPv4 CIDR range of the VPC. Most VPC IP address ranges fall within the private (non-publicly routable) IP address ranges specified in RFC 1918; however, you can use publicly routable CIDR

blocks for your VPC. Regardless of the IP address range of your VPC, we do not support direct access to the internet from your VPC's CIDR block, including a publicly-routable CIDR block. You must set up internet access through a gateway; for example, an internet gateway, virtual private gateway, a AWS Site-to-Site VPN connection, or AWS Direct Connect.

1.4.5 Public IPv4 addresses

All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address (also referred to as a public IP address in this topic).

Therefore, when you launch an instance into a subnet that has this attribute enabled, a public IP address is assigned to the primary network interface (eth0) that's created for the instance. A public IP address is mapped to the primary private IP address through network address translation (NAT).

You can control whether your instance receives a public IP address by doing the following:

- Modifying the public IP addressing attribute of your subnet.
- Enabling or disabling the public IP addressing feature during instance launch, which overrides the subnet's public IP addressing attribute.

A public IP address is assigned from Amazon's pool of public IP addresses; it's not associated with your account. When a public IP address is disassociated from your instance, it's released back into the pool, and is no longer available for you to use. You cannot manually associate or disassociate a public IP address. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one. For more information, see Public IP addresses in the Amazon EC2 User Guide for Linux Instances.

If you require a persistent public IP address allocated to your account that can be assigned to and removed from instances as you require, use an Elastic IP address instead.

If your VPC is enabled to support DNS hostnames, each instance that receives a public IP address or an Elastic IP address is also given a public DNS hostname. We resolve a public DNS hostname to the public IP address of the instance outside the instance network, and to the private IP address of the instance from within the instance network.

1.4.6 IPv6 addresses

You can optionally associate an IPv6 CIDR block with your VPC and subnets.

Your instance in a VPC receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and your subnet, and if one of the following is true:

Your subnet is configured to automatically assign an IPv6 address to the primary network interface of an instance during launch.

You manually assign an IPv6 address to your instance during launch.

You assign an IPv6 address to your instance after launch.

You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can disassociate the IPv6 address from the primary network interface. We do not support IPv6 DNS hostnames for your instance.

1.4.7 Route tables

A route table contains a set of rules, called routes, that are used to determine where network traffic from your VPC is directed. You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table.

Each route in a route table specifies the range of IP addresses where you want the traffic to go (the destination) and the gateway, network interface, or connection through which to send the traffic (the target).

1.4.8 Access the internet

You control how the instances that you launch into a VPC access resources outside the VPC.

A default VPC includes an internet gateway, and each default subnet is a public subnet. Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address. These instances can communicate with the internet through the internet gateway. An internet gateway enables your instances to connect to the internet through the Amazon EC2 network edge.

By default, each instance that you launch into a nondefault subnet has a private IPv4 address, but no public IPv4 address,

unless you specifically assign one at launch, or you modify the subnet's public IP address attribute. These instances can communicate with each other, but can't access the internet.

You can enable internet access for an instance launched into a nondefault subnet by attaching an internet gateway to its VPC (if its VPC is not a default VPC) and associating an Elastic IP address with the instance.

Alternatively, to allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the

internet, you can use a network address translation (NAT) device. NAT maps multiple private IPv4 addresses to a single public IPv4 address. You can configure the NAT device with an Elastic IP address and connect it to the internet through an internet gateway. This makes it possible for an instance in a private subnet to connect to the internet through the NAT device, routing traffic from the instance to the internet gateway and any responses to the instance

If you associate an IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances, instances can connect to the internet over IPv6 through an internet gateway. Alternatively, instances can initiate outbound connections to the internet over IPv6 using an egress-only internet gateway. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.

1.4.9 Connect VPCs and networks

You can create a VPC peering connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can also create a transit gateway and use it to interconnect your VPCs and on-premises networks. The transit gateway acts as a Regional virtual router for traffic flowing between its attachments, which can include VPCs, VPN connections, AWS Direct Connect gateways, and transit gateway peering connections.

1.4.10 Get started with Amazon VPC

You can create AWS resources in the subnets of your virtual private cloud (VPC). For example, to get started with Amazon EC2 quickly, you can launch an EC2 instance into the default subnets of a default VPC. Alternatively, you can create subnets in a custom VPC for your AWS resources. For more information, you'll launch an EC2 instance into a default subnet, connect to the instance, and then terminate the instance. If you are within the AWS Free Tier, there is no charge for launching an On-Demand instance.

Contents

- Prerequisites
- Step 1: Get to know your default VPC
- Step 2: Launch an instance into your VPC
- Step 3: Connect to an EC2 instance in your public subnet
- Step 4: Clean up
- Next steps

Prerequisites

If this is your first time using AWS, you must sign up for an account. When you sign up, your AWS account is automatically signed up for all services in AWS, including Amazon VPC. If you haven't created an

AWS account already, use the following procedure to create one.

1.5 Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open

<https://portal.aws.amazon.com/billing/signup>.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an AWS account root user is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to an administrative user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/>

Who Uses Cloud Computing?

Organizations of different types, sizes, and industries are using the cloud for a wide variety of use cases, such as building customer-facing web applications, data backup, sending email/SMS notifications, virtual desktops, software development and testing, big data analytics, and disaster recovery. For example, Telecom companies are using cloud services to connect with their customers by sending different types of communications. Financial services companies are using the cloud to power real-time fraud detection and prevention

2.2 Cloud computing services

IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service) and SaaS (Software-as-a-Service) are the three most common models of cloud services, and it's not uncommon for an organization to use all three.

SaaS (Software-as-a-Service)

SaaS also known as cloud-based software or cloud applications—is application software that's hosted in the cloud, and that users access via a web browser, a

dedicated desktop client, or an API that integrates with a desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer 'pay-as-you-go' pricing based on your actual usage.

In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:** With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.
- **Protection from data loss:** Because SaaS stores application data in the cloud with the application, users don't lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today—there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications, to powerful enterprise software database and AI (artificial intelligence) software

PaaS (Platform-as-a-Service)

PaaS provides software developers with on-demand platform hardware, complete software stack, infrastructure, and even development tools for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything servers, networks, storage, operating system software, middleware, databases at their data center. Developers simply pick from a menu to 'spin up' servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around containers, a virtualized compute model one step removed

from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

Red Hat OpenShift is a popular PaaS built around Docker containers and Kubernetes, an open source container orchestration solution that automates deployment, scaling, load balancing, and more for container-based applications

IaaS (Infrastructure-as-a-Service)

IaaS provides on-demand access to fundamental computing resources physical and virtual servers, networking, and storage over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage.

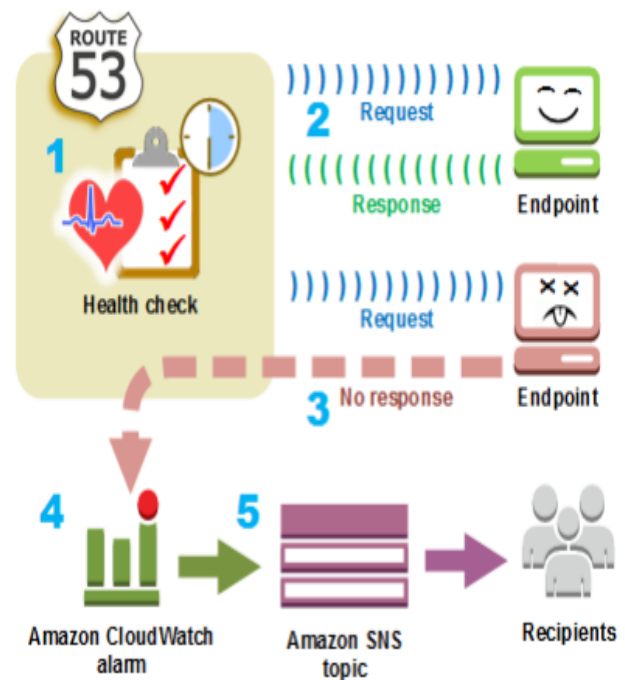
In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud.

IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate

IaaS	PaaS	SaaS
Data Access Security	Data Access Security	Data Access Security
Application Security	Application Security	Application Security
Middleware Security	Middleware Security	Middleware Security
Operating System Security	Operating System Security	Operating System Security
Network Security	Network Security	Network Security
Virtualized Infrastructure Security	Virtualized Infrastructure Security	Virtualized Infrastructure Security
Physical Security	Physical Security	Physical Security
Customer Responsibility	Shared Responsibility	Vendor's Responsibility

How Amazon Route 53 checks the health of your resources

Amazon Route 53 health checks monitor the health of your resources such as web servers and email servers. You can optionally configure Amazon CloudWatch alarms for your health checks, so that you receive notification when a resource becomes unavailable. Here's an overview of how health checking works if you want to be notified when a resource becomes unavailable



1. You create a health check and specify values that define how you want the health check to work, such as the following:
 - The IP address or domain name of the endpoint, such as a web server, that you want Route 53 to monitor. (You can also monitor the status of other health checks, or the state of a CloudWatch alarm.)
 - The protocol that you want Amazon Route 53 to use to perform the check: HTTP, HTTPS, or TCP.
 - How often you want Route 53 to send a request to the endpoint. This is the request interval.

- How many consecutive times the endpoint must fail to respond to requests before Route 53 considers it unhealthy. This is the failure threshold.
 - Optionally, how you want to be notified when Route 53 detects that the endpoint is unhealthy. When you configure notification, Route 53 automatically sets a CloudWatch alarm. CloudWatch uses Amazon SNS to notify users that an endpoint is unhealthy.
2. Route 53 starts to send requests to the endpoint at the interval that you specified in the health check.
If the endpoint responds to the requests, Route 53 considers the endpoint to be healthy and takes no action.
 3. If the endpoint doesn't respond to a request, Route 53 starts to count the number of consecutive requests that the endpoint doesn't respond to:
 - If the count reaches the value that you specified for the failure threshold, Route 53 considers the endpoint unhealthy.
 - If the endpoint starts to respond again before the count reaches the failure threshold, Route 53 resets the count to 0, and CloudWatch doesn't contact you.
 4. If Route 53 considers the endpoint unhealthy and if you configured notification for the health check, Route 53 notifies CloudWatch.
 5. If you configured notification for the health check, CloudWatch triggers an alarm and uses Amazon SNS to send notification to the specified recipients. In addition to checking the health of a specified endpoint, you can configure a health check to check the health of one or more other health checks so that you can be notified when a specified number of

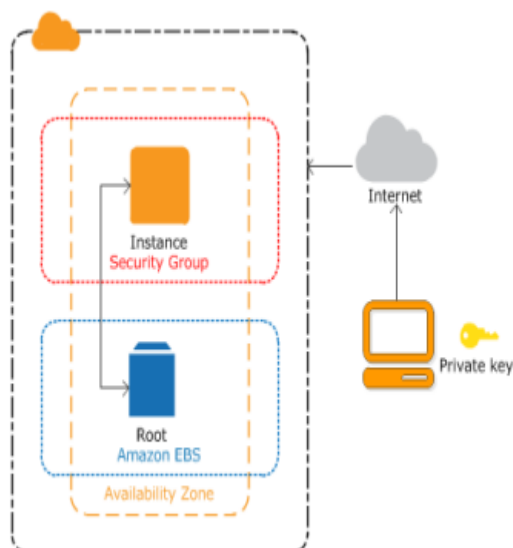
resources, such as two web servers out of five, are unavailable. You can also configure a health check to check the status of a CloudWatch alarm so that you can be notified on the basis of a broad range of criteria, not just whether a resource is responding to requests

6. If you have multiple resources that perform the same function, for example, web servers or database servers, and you want Route 53 to route traffic only to the resources that are healthy, you can configure DNS failover by associating a health check with each record for that resource. If a health check determines that the underlying resource is unhealthy, Route 53 routes traffic away from the associated record
7. **Get started with Amazon EC2 Linux instances**
8. Use this tutorial to get started with Amazon Elastic Compute Cloud (Amazon EC2). You'll learn how to launch, connect to, and use a Linux instance. An instance is a virtual server in the AWS Cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance. When you sign up for AWS, you can get started with Amazon EC2 using the AWS Free Tier. If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it won't cost you anything to complete this tutorial because we help you select options that are within the free tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance

(which is the final task of this tutorial), even if it remains idle

Overview

9. The instance launched in this tutorial is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. Availability Zones are multiple, isolated locations within each Region. You can think of an Availability Zone as an isolated data center. When you launch your instance, you secure it by specifying a key pair (to prove your identity) and a security group (which acts as a virtual firewall to control ingoing and outgoing traffic). When you connect to your instance, you must provide the private key of the key pair that you specified when you launched your instance.



10. Launch an instance

You can launch a Linux instance using the AWS Management Console as described in the

following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options. For information about advanced options, see Launch an instance using the new launch instance wizard For information about other ways to launch your instance, see Launch your instance To launch an instance

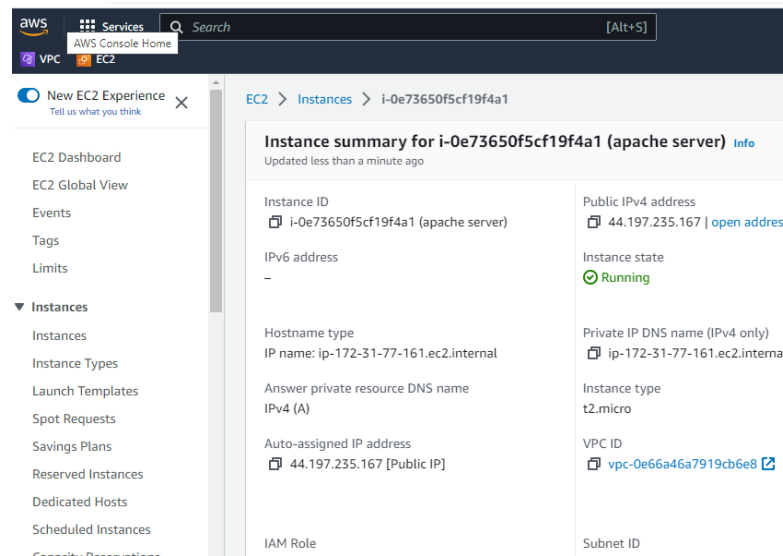
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the Launch instance box, choose Launch instance, and then choose Launch instance from the options that appear.
3. Under Name and tags, for Name, enter a descriptive name for your instance.
4. Under Application and OS Images (Amazon Machine Image), do the following:
 - a. Choose Quick Start, and then choose Amazon Linux. This is the operating system (OS) for your instance.
 - b. From Amazon Machine Image (AMI), select an HVM version of Amazon Linux 2. Notice that these AMIs are marked Free tier eligible. An Amazon Machine Image (AMI) is a basic configuration that serves as a template for your instance.
5. Under Instance type, from the Instance type list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. For more information, see AWS Free Tier.
6. Under Key pair (login), for Key pair name, choose the key pair that you created when getting set up.

Warning Do not choose Proceed without a key pair (Not recommended). If you launch your instance without a key pair, then you can't connect to it.

IMPLEMENTATION

Launching EC2 Instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**. In **Step 1: Choose an Amazon Machine Image (AMI)**, find an Amazon Linux 2 AMI at the top of the list and choose **Select**. In **Step 2: Choose an Instance Type**, choose **Next: Configure Instance Details**. In **Step 3: Configure Instance Details**, provide the following information:
 - Leave **Number of instances** at one.
 - Leave **Purchasing option** at the default setting.



Installing Apache server and added some content:

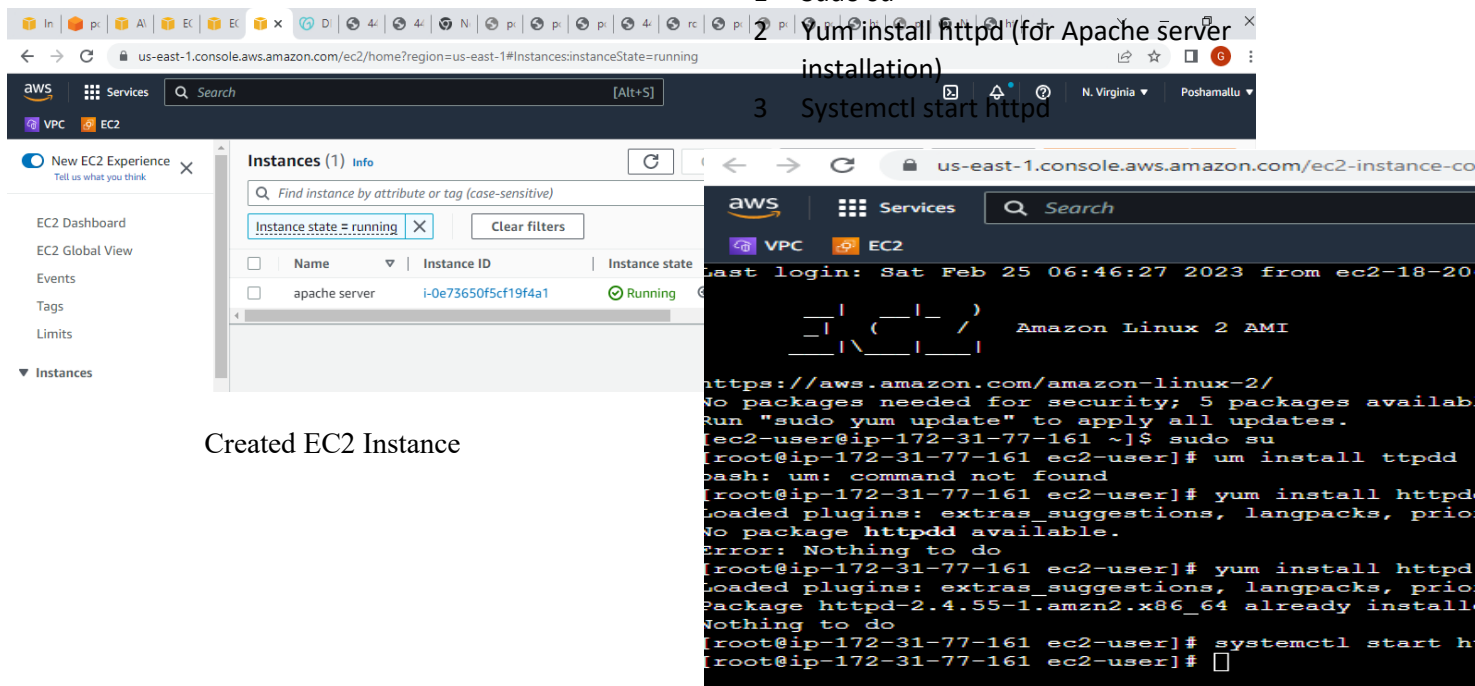
Before you install Apache server for the first time on a new host machine, you need to set up the EC2 Instance type. After that we can install ec2 instance Connected with EC2 INSTANCE The command prompt window

First command

1 Sudo su

2 Yum install httpd (for Apache server installation)

3 Systemctl start httpd



Created EC2 Instance

APACHE SERVER ADDING SOME CONTENT

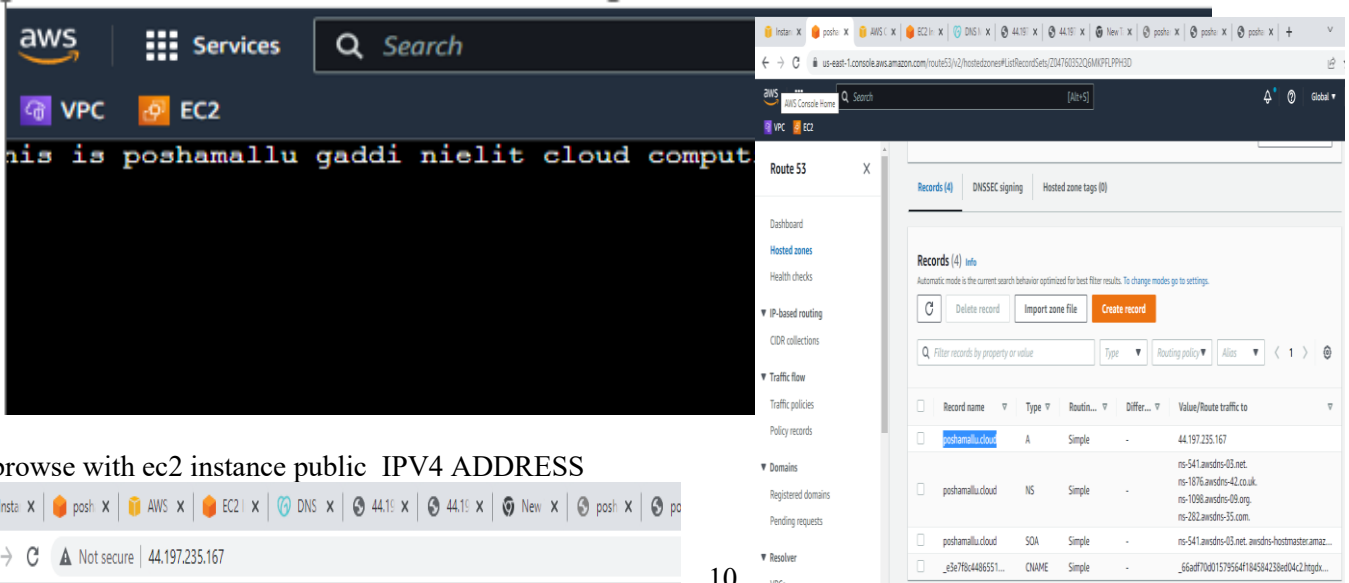
After installing the Apache server added some content (This is Poshamalla gaddi nielit cloud computing engineer using below commands

4.cd /var/www/html

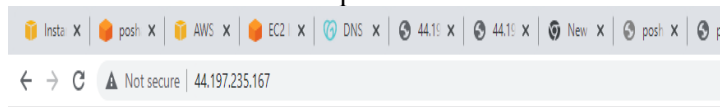
vi index.html

Html page opened added some content

9. Created a route 53 instance



browse with ec2 instance public IPV4 ADDRESS



10.

11.

III. RESULTS AND DISCUSSIONS

Cell studies:

Load balancer creation

Godaddy account with domain name Poshamallu.cloud

1. Go to godaddy.com.
2. Click Sign In, and then in the New Customer area, click Create My Account.
3. Complete the onscreen fields, and then click Create Account.
4. Search with your domain name
5. Go to your GoDaddy Payment Methods page. You might be prompted to sign in.
6. Select Add Payment Method.
7. Next to Billing Information, select Edit.

Register domain names route 53

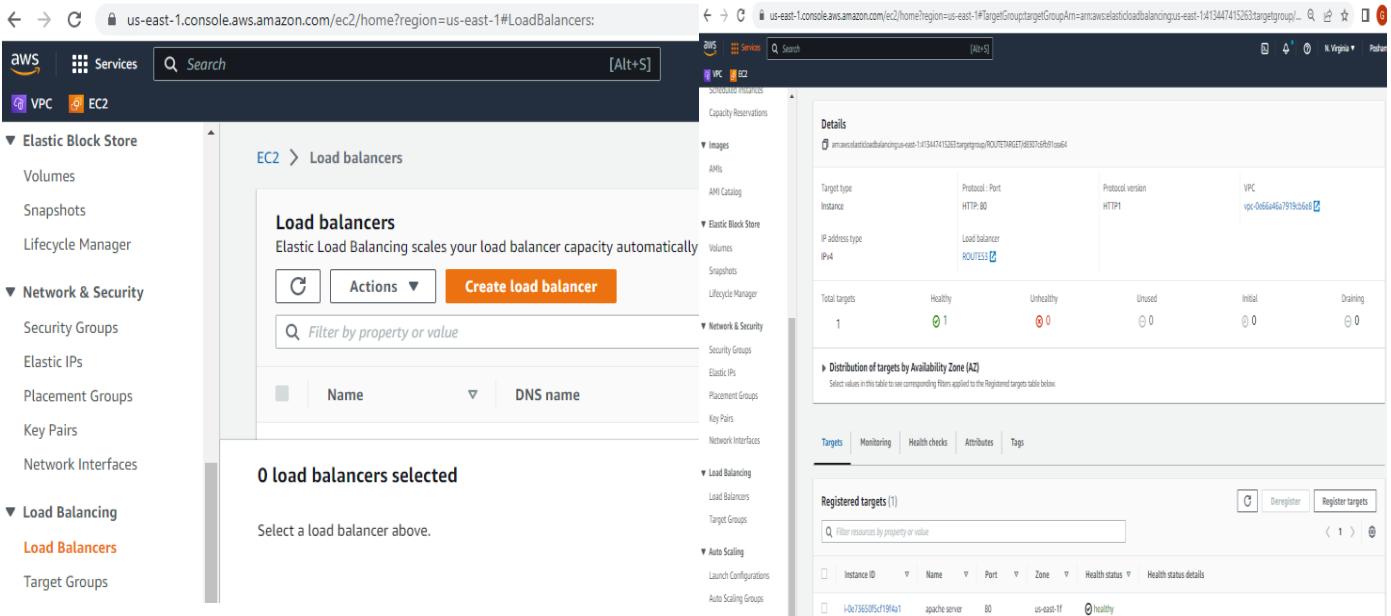
Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises.

8. Your website needs a name, such as example.com. Route 53 lets you register a name for your website or web application, known as a *domain name*.

Elastic Load Balancing supports different types of load balancers. For this tutorial, you create a Classic Load Balancer.

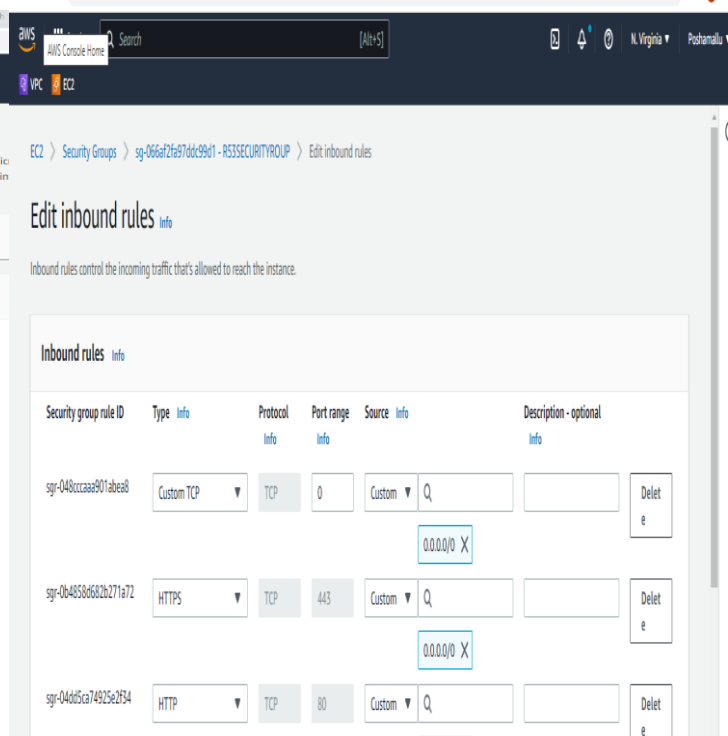
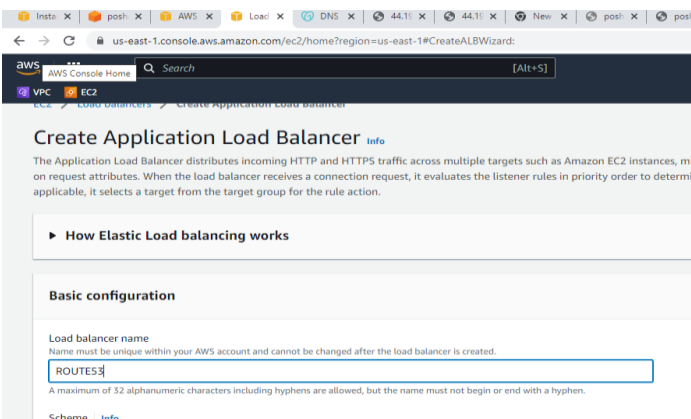
To create a Classic Load Balancer

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar, choose a Region for your load balancer. Be sure to select the same Region that you selected for your EC2 instances.
3. On the navigation pane, under LOAD BALANCING, choose Load Balancers.
4. Choose Create Load Balancer.
5. For application Load Balancer, choose Create.



Give some name for load balancer

Inbound rules changed to http and https changed to anywhere because the health status should be healthy

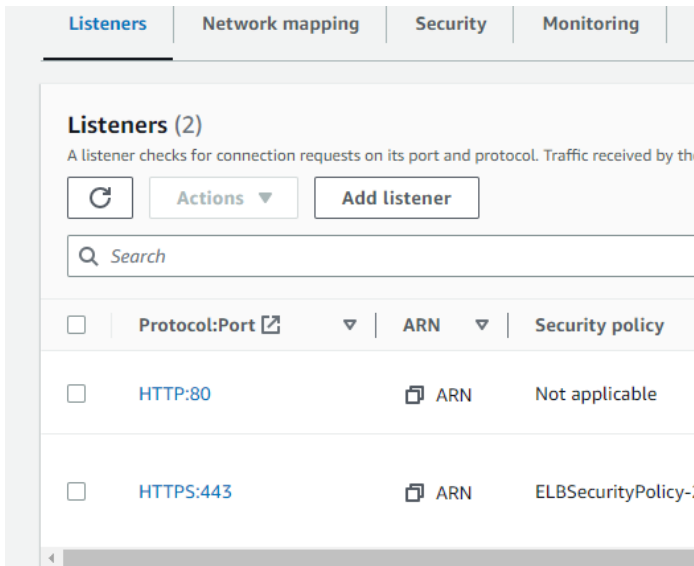


SECURITY GROUP

To update security groups using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Load Balancers**.
3. Select the load balancer.
4. On the **Security** tab, choose **Edit**.
5. To associate a security group with your load balancer, select it. To remove a security group association, choose the **X** icon for the security group.
6. Choose **Save changes**.

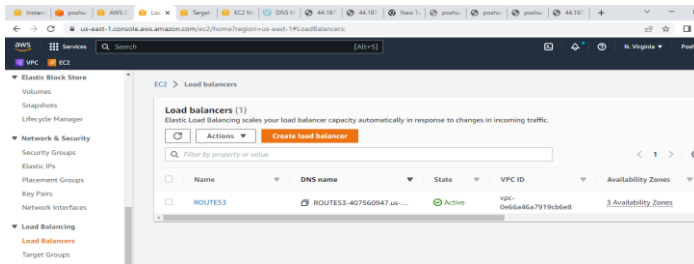
Added the listeners http and https



load balancer created but the status is provisioning

Traffic alias application and classic load balancer create a record in route 53 added a record type a route traffic to alias to application and classic load balancer and select your region and attach your load balancer and create a record

The load balancer status now active



Record name

Enter the domain or subdomain name that you want to use to route traffic to your ELB load balancer. The default value is the name of the hosted zone.

For example, if the name of the hosted zone is example.com and you want to

LOAD BALANCER DNS NAME

Open the [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) console.

2. Under **Load Balancing**, choose **Load Balancers** from the navigation pane.

3. Select the load balancer that you're finding the IP addresses for.

4. On the **Description** tab, copy the **Name**.

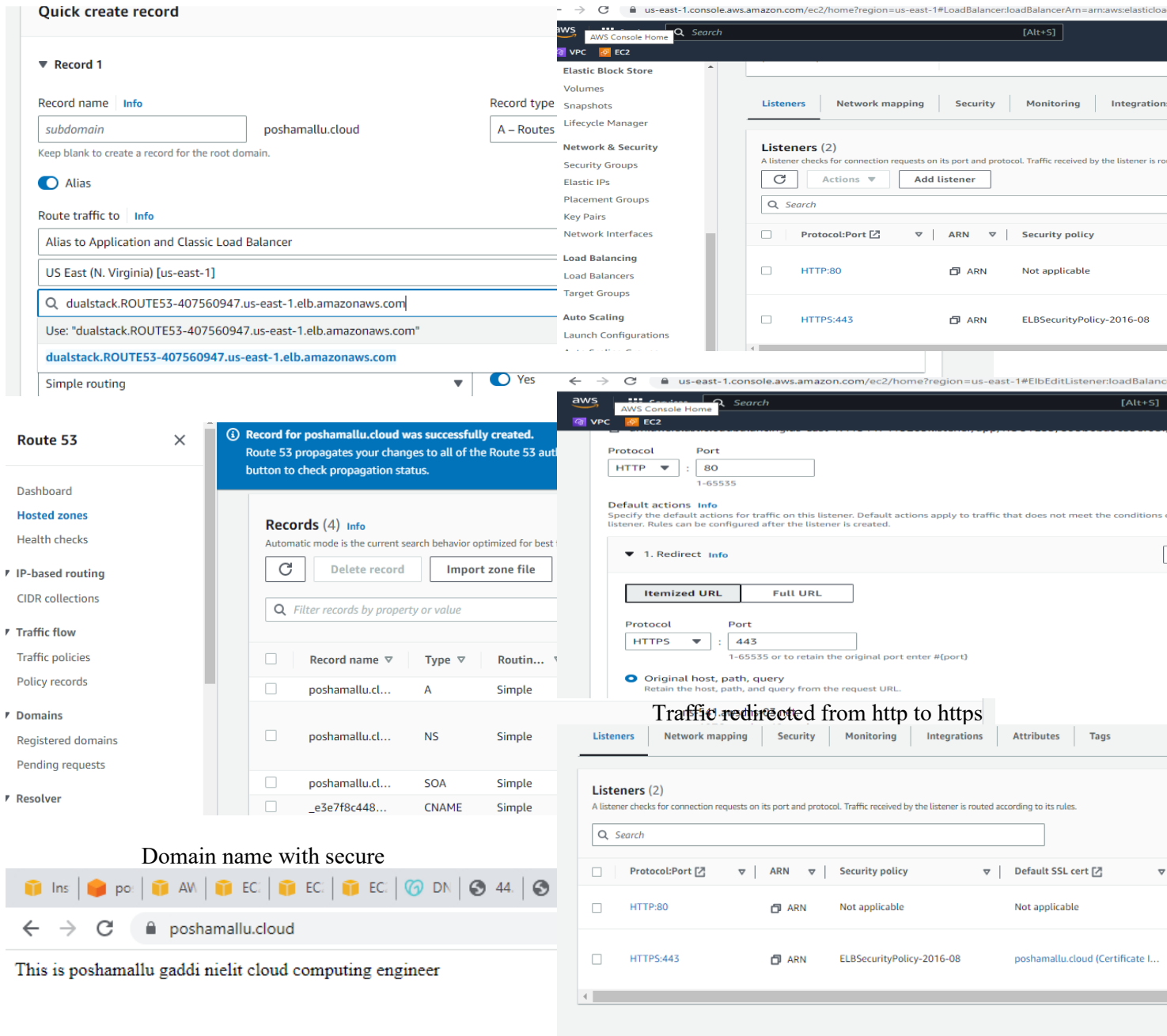
5. Under **Network & Security**, choose **Network Interfaces** from the navigation pane.

Paste the load balancer name that you copied in step 4 in the search box. The filtered results show all elastic network interfaces associated with the load balancer.

Alias

If you are using the Quick create record creation method, turn on **Alias**. Value/Route traffic to

Choose **Alias to Application and Classic Load Balancer** or **Alias to Network Load Balancer**, then choose the Region that the endpoint is from.



Quick create record

Record name: poshamallu.cloud

Record type: A - Routes

Route traffic to: Alias to Application and Classic Load Balancer

Region: US East (N. Virginia) [us-east-1]

Endpoint:

Use: "dualstack.ROUTE53-407560947.us-east-1.elb.amazonaws.com"

Routing: dualstack.ROUTE53-407560947.us-east-1.elb.amazonaws.com

Simple routing: Yes

Route 53

Dashboard

Hosted zones

Health checks

IP-based routing

CIDR collections

Traffic flow

Traffic policies

Policy records

Domains

Registered domains

Pending requests

Resolver

Record for poshamallu.cloud was successfully created.

Route 53 propagates your changes to all of the Route 53 endpoints. Click the Refresh button to check propagation status.

Records (4) Info

Automatic mode is the current search behavior optimized for best results.

Filter records by property or value

Record name	Type	Routing
poshamallu.cl...	A	Simple
poshamallu.cl...	NS	Simple
poshamallu.cl...	SOA	Simple
_e3e7f8c448...	CNAME	Simple

Traffic redirected from http to https

Protocol: HTTP Port: 80

Default actions:

1. Redirect
 - Itemized URL
 - Full URL
 - Protocol: HTTPS Port: 443
 - Original host, path, query

Listeners (2)

Protocol:Port	ARN	Security policy	Default SSL cert
HTTP:80	ARN	Not applicable	Not applicable
HTTPS:443	ARN	ELBSecurityPolicy-2016-08	poshamallu.cloud (Certificate I...

Domain name with secure

poshamallu.cloud

This is poshamallu gaddi nielit cloud computing engineer

The domain name with security final result

CONCLUSION

With Amazon Route 53, We can create and manage your public DNS records. Like a phone book, Route 53 lets you manage the IP addresses listed for your domain names in the Internet's DNS phone book. Route 53 also answers requests to translate specific domain names like into their corresponding IP addresses like 192.0.2.1. You can use Route 53 to create DNS records for a new domain or transfer DNS

records for an existing domain. The simple, standards-based REST API for Route 53 allows you to easily create, update and manage DNS records. Route 53 additionally offers health checks to monitor the health and performance of your application as well as your web servers and other resources. You can also register new domain names or transfer in existing domain names to be managed by Route 53.

REFERENCES

- [1] American Council of Technology (2011), The Role of Enterprise Architecture in Federal Cloud Computing, Shared Interest Group: Enterprise Architecture, A White Paper, from the collaboration of ACT and IAC, Fairfax, VA, accessed via WWW and Retrieved on 07-02- 2012, Available @<http://www.actgov.org/knowledgebank/whitepapers/Documents/Shared%20Interest%20Groups/Enterprise%20Architecture%20SIG/Role%20of%20EA%20in%20Federal%20Cloud%20Computing%20EA%20SIG-%2001-2011.pdf>
- [2] Armbrust et al. (2011), A view of cloud computing, Communications of the ACM, 53(4), pp.50-58.
- [3] Brown, I. and Laurie, B. (2000), Security against compelled disclosure In Computer Security Applications 16th Annual Conference (ACSAC '00). IEEE, New Orleans, LA , USA,
 1. Accessed from WWW and Retrieved on 12-03-2012 and Available @ <http://www.apachessl.org/disclosure.pdf>
2. [4] Bruening and Treacy (2009), Privacy and Security Law Report, The Bureau Of National Affairs, Inc., Accessed from WWW and Retrieved on 12-03-2012 and Available @ http://www.hunton.com/files/Publication/6acf0d97-7c21-42d1-ab48-315a04601152/Presentation/PublicationAttachment/37dc2129-4f0c-45a0-8417-651e05dc423f/CloudComputing_Bruening-Treacy.pdf
3. [5] Clark, K., Warnier, M. and Brazier, F. M. T. (2011), Botclouds: The future of cloud-based Botnets? Closer Sci Te Press, p. 597-603, <http://homepage.tudelft.nl/68x7e/Papers/botclouds.pdf>.
- [6] Cavoukian, A. (1999), Privacy as a Fundamental Human Right vs Economic Right: An Attempt at Conciliation, Information and Privacy Commissioner, Ontario, Canada Available from <<http://www.ipc.on.ca>>
- [7] De Boni, M., and Prigmore, M. (2001), A Hegelian basis for information privacy as an economic right, Proceedings of the UKAIS conference, Portsmouth.
- [8] Edouard, N. & White, W. (eds.) (1999) UK PLC on the World Stage In 2010: Book 1: The Development of The Internet And The Growth Of E-Commerce Research Report, London, Management Consultancies Association.
- [9] Esteves, R.M. and Chunming Rong (2010), Social Impact of Privacy in Cloud Computing In 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), Nov. 30-Dec. 3 ,2010, pp. 593-596.
- [10] Garrie, D., Who has Legal Jurisdiction in the Cloud? , Accessed from WWW and Retrieved on 12-03-2012 and Available @ <https://www.gplus.com/legal-issues/insight/who-has-legal-jurisdiction-in-the-cloud-50084>
- [11] Gellman, R. (2009), Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud Computing, World Privacy Forum, USA.
- [12] Global Internet Liberty Campaign, Privacy and Human Rights: An International Survey of Privacy Laws and Practice, Accessed from WWW and Retrieved on 07-02-2012 and Available @ <http://gilc.org/privacy/survey/intro.html>