

Auto-Healing Multi-Region AWS Web Application using Terraform

Introduction:

Modern web applications must be highly available, fault tolerant, and capable of recovering automatically from infrastructure failures. To achieve this, cloud platforms such as Amazon Web Services (AWS) provide managed services that support high availability, scalability, and disaster recovery across multiple regions.

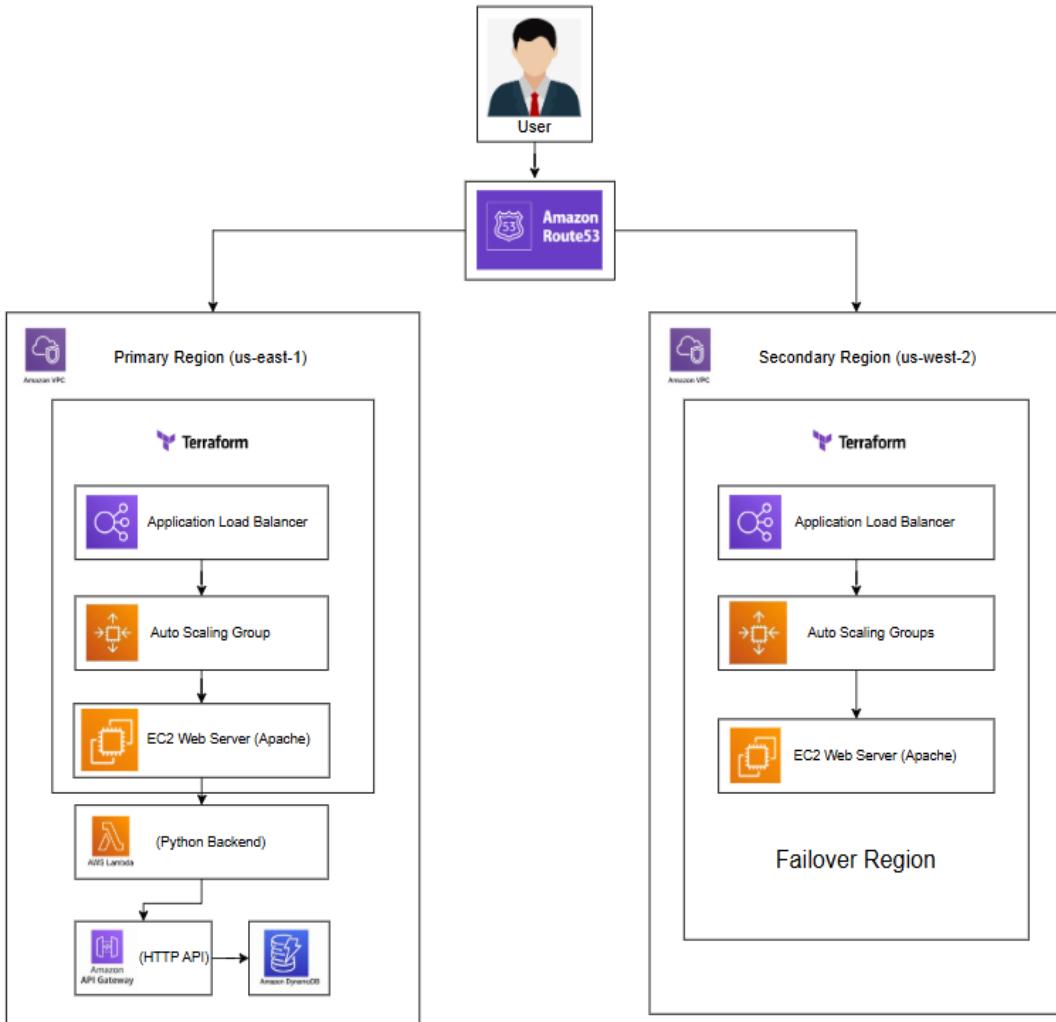
This project focuses on the design and implementation of an **auto-healing, multi-region highly available web application architecture on AWS**, built using **Infrastructure as Code (IaC) with Terraform**. The application leverages **Elastic Load Balancing (ALB)** and **Auto Scaling Groups (ASG)** to ensure automatic recovery from instance-level failures.

The primary region hosts the complete application stack, including the web layer and backend services implemented using **API Gateway, AWS Lambda, and Amazon DynamoDB**. The secondary region is configured as a **disaster recovery (DR) region**, providing high availability for the web and compute layer. This design ensures continuous access to the application during regional outages while keeping backend services centralized to maintain data consistency and control operational cost.

Combining multi-region deployment, health-based failover, and auto-healing mechanisms, this project demonstrates a production-grade cloud architecture aligned with real-world industry practices. It also highlights how Terraform can be used to manage and reproduce cloud infrastructure reliably and consistently.

Architecture Diagram:

Auto-Healing Multi-Region AWS Web Application using Terraform



Detailed Steps:

- 1) Begin with create **IAM roles** to restrict permission
 - # Create EC2 service
 - 1) AmazonDynamoDBFullAccess
 - 2)AmazonEC2RoleforSSM
 - 3)CloudWatchAgentServerPolicy

Step 1
 Select trusted entity
 Step 2
 Add permissions
 Step 3
 Name, review, and create

Select trusted entity Info

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Cancel Next

Step 1
 Select trusted entity
 Step 2
 Add permissions
 Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+-_.' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: +-_@/{}#\$%^&`-

Step 1: Select trusted entities Edit

Trust policy

```

1+ [
2+   "Version": "2012-10-17",
3+   "Statement": [
4+     {
5+       "Effect": "Allow",
6+       "Action": "sts:AssumeRole"
7+     },
8+     "Principal": [
9+       "Service": [
10+         "ec2.amazonaws.com"
11+       ]
12+

```

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonEC2RoleforSSM	AWS managed	Permissions policy
CloudWatchAgentServerPolicy	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role
W U

Role WebApp-EC2-Role created. View role X

Roles (13) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

<input type="checkbox"/> Role name	Trusted entities	Last activity
<input type="checkbox"/> AWSRoleForAPIGateway	AWS Service: ops.apigateway (Service-Li	-
<input type="checkbox"/> AWSRoleForAutoScaling	AWS Service: autoscaling (Service-Li	3 days ago
<input type="checkbox"/> AWSRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	5 days ago

< 1 > ⌘

2) Create VPC and choose VPC and more option in which you can choose all settings as required.

[VPC](#) > [Your VPCs](#) > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.
 Auto-generate

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.
 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy [Info](#)
 Default Dedicated

Encryption settings - optional

Preview

VPC [Show details](#)
Your AWS virtual network

WebApp-VPC-vpc

Subnets (2)
Subnets within this VPC

- us-east-1a WebApp-VPC-subnet-public1-us-
- us-east-1b WebApp-VPC-subnet-public2-us-

Route tables (1)
Route network traffic to resources

WebApp-VPC-rtb-public

[VPC](#) > [Your VPCs](#) > Create VPC

secure backend resources that don't need public access.

Customize subnets CIDR blocks

NAT gateways (\$ - updated) [Info](#)
NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

None Regional - new Zonal

Introducing regional NAT gateway [Info](#)
AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones.

Endpoint [Info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

DNS options [Info](#)
 Enable DNS hostnames Enable DNS resolution

Additional tags

[Cancel](#) [Preview code](#) [Create VPC](#)

Create VPC workflow

Associate route table 77%

▼ Details

- ✓ Create VPC: vpc-0e3848fb897ad3fa6 ↗
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-0e3848fb897ad3fa6 ↗
- ✓ Create subnet: subnet-0b593eaff3ff5a525 ↗
- ✓ Create subnet: subnet-04a963ee334de1b69 ↗
- ✓ Create internet gateway: igw-05e946f14b4c08721 ↗
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-00982d640f058fdad ↗
- ✓ Create route
 - ↳ Associate route table
 - ↳ Associate route table
 - ↳ Verifying route table creation

3) Create **Security Groups** with inbound and outbound rules.

≡ VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name **Info**
WebApp-ALB-SG
Name cannot be edited after creation.

Description **Info**
ALB security group for public access

VPC **Info**
vpc-0e3848fb897ad3fa6 (WebApp-VPC-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere... 0.0.0.0/0	<input type="button" value="Delete"/>
HTTPS	TCP	443	Anywhere... 0.0.0.0/0	<input type="button" value="Delete"/>

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
WebApp-EC2-SG
Name cannot be edited after creation.

Description Info
EC2-SG allow traffic only from ALB

VPC Info
vpc-0e384fb897ad5fa6 (WebApp-VPC-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Custom	Q sg-0227db797abe6605f X sg-0227db797abe6605f X

[Add rule](#)

Outbound rules Info

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	Q 0.0.0.0 X

4) Create **Launch template** choose key pair, OS image, instance type and so on.

Search results

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required
WebApp-LT
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '"', '@'.

Template version description
Initial version for ASG
Max 255 chars

Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

[Template tags](#)
[Source template](#)

Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Summary

Software Image (AMI)
Amazon Linux 2023 kernel-6.1 A... [read more](#)
ami-068c0051b15cd816

Virtual server type (instance type)
t2.micro

Firewall (security group)
WebApp-EC2-SG

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.

5) Create **Target Group** with HTTP/80.

EC2 > Target groups > Create target group

Create target group

A target group can be made up of one or more targets. Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Settings - immutable

Choose a target type and the load balancer and listener will route traffic to your target. These settings can't be modified after target group creation.*

Target type

Indicate what resource type you want to target. Only the selected resource type can be registered to this target group.

- Instances

Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.

Suitable for: **ALB** **NLB** **GWLB**
- IP addresses

Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.

Suitable for: **ALB** **NLB** **GWLB**
- Lambda function

Supports load balancing to a single Lambda function. ALB required as traffic source.

Suitable for: **ALB**
- Application Load Balancer

Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.

Suitable for: **NLB**

Target group name

Name must be unique per Region per AWS account.

WebApp-TG

Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 9/32

Protocol

Protocol for communication between the load balancer and targets.

HTTP

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

Target group details

Name WebApp-TG	Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1
VPC vpc-0e384fb897ad3fa6	IP address type IPv4		

Health check details

Health check protocol HTTP	Health check path /	Health check port traffic-port	Interval 30 seconds
Timeout 10 seconds	Healthy threshold 2	Unhealthy threshold 2	Success codes 200

Step 2: Register targets

Targets (0)

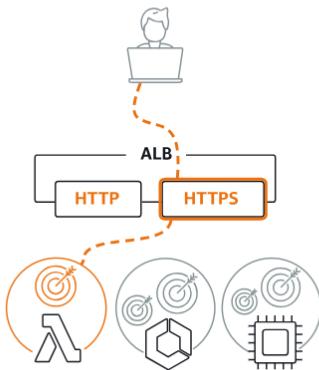
Instance ID	Name	Port	Zone
No targets added			

Create target group

6) Choose Application Load Balancer from **Load Balancer** and create load balancer with the created Target Group.

Load balancer types

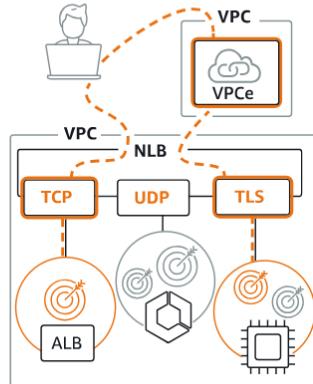
Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Create](#)

Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

[Create](#)

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

[Create](#)

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

WebApp-ALB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type [Info](#)

Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4

Includes only IPv4 addresses.

Dualstack

Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4

Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80

Protocol	Port	80 1-65535
----------	------	---------------

Default action | Info
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Routing action

<input checked="" type="radio"/> Forward to target groups	<input type="radio"/> Redirect to URL	<input type="radio"/> Return fixed response
---	---------------------------------------	---

Forward to target group | Info
Choose a target group and specify routing weight or [create target group](#).

Target group

WebApp-TG Target type: Instance, IPv4 Target stickiness: Off	HTTP	Weight <input type="text" value="1"/> 0-999
---	------	---

+ Add target group
You can add up to 4 more target groups.

Target group stickiness | Info
Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

Turn on target group stickiness

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

7) Go to **Auto Scaling Groups** and choose the created template and settings accordingly.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1

Choose launch template

- Step 2
- Choose instance launch options
- Step 5 - optional
- Integrate with other services
- Step 4 - optional
- Configure group size and scaling
- Step 5 - optional
- Add notifications
- Step 6 - optional
- Add tags
- Step 7
- Review

Choose launch template | Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

Launch template | Info

(i) For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Version

Description
Initial version for ASG

Launch template
WebApp-LT
ID-0631880ab255526e7

Instance type
t2.micro

Step 2: Set instance type

Additional settings

Instance scale-in protection Disabled	Monitoring Disabled	Default instance warmup Disabled
---	-------------------------------	--

Capacity Reservation preference

Preference Default	Capacity Reservation IDs -	Resource Groups -
------------------------------	--------------------------------------	-----------------------------

Step 3: Set scaling policy

Step 4: Set scaling metrics

Step 5: Add notifications

Notifications
No notifications

Step 6: Add tags

Tags (0)

Key	Value	Tag new instances
------------	--------------	--------------------------

No tags

Preview code

The screenshot shows the AWS Auto Scaling Groups page. At the top, a green banner displays the message "WebApp-ASG created successfully". Below this, the "Auto Scaling groups (1) Info" section is shown. A search bar is at the top left. To the right are buttons for "Launch configurations", "Launch templates", "Actions", and "Create Auto Scaling group". The main table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, Availability Zones, and Creation time. One row is visible for "WebApp-ASG" with "WebApp-LT" as the launch template.

8) Choose the template and modify the settings and use the updated version with User data.

The screenshot shows the AWS Launch Templates page. A context menu is open over a row in the table, specifically for the "lt-0631880ab255526e7" entry. The menu items include "Launch instance from template", "Modify template (Create new version)" (which is highlighted), "Delete template", "Delete template version", "Set default version", "Manage tags", "Create Spot Fleet", "Create Auto Scaling group", and "View details".

The screenshot shows the "Modify template (Create new version)" configuration page. The left panel contains fields for "Launch template name and version description", "Template version description" (with "updated version" entered), and "Auto Scaling guidance" (with "Provide guidance to help me set up a template that I can use with EC2 Auto Scaling" checked). The right panel is titled "Summary" and includes sections for "Software Image (AMI)", "Virtual server type (instance type)" (set to t2.micro), "Firewall (security group)" (set to WebApp-EC2-5G), and "Storage (volumes)" (1 volume(s) - 8 GiB). A note about the free tier is displayed in a box. At the bottom are "Cancel" and "Create template version" buttons.

User data - optional | Info

Upload a file with your user data or enter it in the field.

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>WebApp running via ALB + ASG</h1>" > /var/www/html/index.html
```

User data has already been base64 encoded

9) Choose the ASG and edit capacities.

The screenshot shows the AWS EC2 Auto Scaling groups page. On the left, there's a sidebar with navigation links like EC2 Global View, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area displays 'Auto Scaling groups (1/1)' with a single entry: 'WebApp-ASG' (Launch template: WebApp-LT, Version: Default). The 'Actions' dropdown menu for this group includes 'Edit' and 'Delete'. The status bar at the bottom indicates 'Last updated less than a minute ago'.

This screenshot shows the 'Edit' page for the 'WebApp-ASG' Auto Scaling group. It includes fields for 'Desired capacity' (set to 1), 'Scaling limits' (Min desired capacity: 1, Max desired capacity: 1), and a 'Launch template' section. The launch template is set to 'WebApp-LT'. Other settings shown include 'Description' (updated version), 'AMID' (ami-068c0051b15cdb816), 'Launch template' (WebApp-LT), 'Instance type' (t2.micro), and 'Request Spot Instances' (No).

10) Create DynamoDB tables

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. In the 'Table details' step, a 'Table name' is set to 'WebAppData'. Under 'Partition key', 'user_id' is selected as a String type. In the 'Sort key - optional' section, 'name' is selected as a String type. The 'Table settings' step shows 'Default settings' selected, with a note that it's the fastest way to create the table. A success message at the bottom indicates 'The WebAppData table was created successfully.'

Tables (1) Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode
WebAppData	Active	user_id (\$)	name (\$)	0	0	Off	☆	On-demand	On-demand

11) Set Lambda role

- 1) AmazonDynamoDBFullAccess
- 2) AWSLambdaBasicExecutionRole

The screenshot shows the 'Create role' wizard in the AWS IAM console. In the 'Select trusted entity' step, 'AWS service' is selected under 'Trusted entity type'. In the 'Use case' step, 'Lambda' is selected under 'Service or use case'. The 'Use case' dropdown also lists 'AWSAccount' and 'WebIdentity'.

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional [Info](#)
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create role](#)

12) Create a **Lambda Function** and python 3.12 as runtime.

[Lambda](#) > [Functions](#) > [Create function](#)

Create function [Info](#)
Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
WebAppBackend
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime [Info](#)
Choose the language to use for your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12 Last fetched 12/25/2025, 10:23:37 AM

Portable execution - new [Info](#)
Enable durable execution to simplify building resilient multi-step applications that checkpoint progress and resume after interruptions. Supports Python and Node.js runtimes. [View pricing](#)

Enable

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 arm64
 x86_64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

[Lambda](#) > [Functions](#) > [WebAppBackend](#)

WebAppBackend

Function overview [Info](#) [Throttle](#) [Copy ARN](#) [Actions](#)

[Diagram](#) [Template](#)

WebAppBackend

Layers (0)

[+ Add trigger](#) [+ Add destination](#)

Description

Last modified
4 seconds ago

Function ARN
[arn:aws:lambda:us-east-1:390402567136:function:WebAppBackend](#)

Function URL [Info](#)

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Code source [Info](#) [Open in Visual Studio Code](#) [Upload from](#)

EXPLORER **WEBAPPBACKEND**

13) Type the Code and deploy the code.

```

 1 import json
 2 import boto3
 3
 4 dynamodb = boto3.resource('dynamodb')
 5 table = dynamodb.Table('WebappData')
 6
 7 def lambda_handler(event, context):
 8     user_id = event.get("user_id", "default-user")
 9     name = event.get("name", "default-name")
10
11     table.put_item(
12         Item={
13             "user_id": user_id, # Partition Key
14             "name": name, # Sort Key
15             "status": "active"
16         }
17     )
18
19     return {

```

Testing: Test the code with sample json

Event sharing settings

- Private**
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)
- Shareable**
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

Hello World

Event JSON

```

1 * {
2   "user_id": "jeffersen",
3   "name": "cloud-project"
4 }
5

```

Execution function: succeeded (logs)

Details

```

{
    "statusCode": 200,
    "body": "\u201cItem Inserted successfully\u201d"
}

```

Summary

Code SHA-256 ljk9VV5/W8JqfF14nuvzC/VCDRnXfepxW1FU8rUlo=	Execution time 1 minute ago
Function version \$LATEST	Request ID 215e1846-6d43-4bdf-aa87-8c0b373d7782
Duration 295.90 ms	Billed duration 760 ms
Resources configured 128 MB	Max memory used 87 MB
Init duration 463.46 ms	
Log output	

The area below shows the last 4 KB of the execution log. [Click here](#) to view the corresponding CloudWatch log group.

```

START RequestId: 215e1846-6d43-4bdf-aa87-8c0b373d7782 Version: $LATEST

```

14) Build HTTP API from **API Gateway** connect to EC2 and give DynamoDB table details.

API Gateway > APIs > Create API

Choose an API type [Info](#)

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

[Import](#) [Build](#)

WebSocket API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

[Build](#)

REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

API Gateway > APIs > [Create API](#) > Create HTTP API

Step 1 Configure API
Step 2 - optional Configure routes
Step 3 - optional Define stages
Step 4 Review and create

Review and create

API name and integrations

API name: WebApp-Lambda-API IP address type: IPv4
Integrations: • WebAppBackend (Lambda)

Routes

Routes: • POST /create --> WebAppBackend (Lambda)

Stages

Stages: • prod (Auto-deploy: enabled)

[Cancel](#) [Previous](#) [Create](#) [Stage: prod](#) [Delete](#) [Edit](#)

Stages

Stages for WebApp-Lambda-API [Create](#)

Search
 prod

Stage	Actions
prod	Edit Delete

Stage details

Details

Name: prod	Created: December 25, 2025 2:52 PM	Last updated: December 25, 2025 2:52 PM
Invoke URL: https://13zpm52fb1.execute-api.us-east-1.amazonaws.com/prod		
Description: None		
Attached deployment		
Automatic Deployment		
<input checked="" type="radio"/> Enabled		
Deployment ID: 2tivie	Deployment created: December 25, 2025 2:52 PM	
Deployment description		
Automatic deployment triggered by changes to the API configuration		
Stage variables		
<input type="text"/> Find resources		
Key	Value	

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

ast login: Thu Dec 25 13:35:52 2025 from 18.206.107.28
ec2-user@ip-10-0-31-166 ~$ curl -X POST \
-H "Content-Type: application/json" \
-d '{"name": "Jeffersen"}' \
"https://13zpm52fb1.execute-api.us-east-1.amazonaws.com/prod/create"
"message": "Item created successfully", "item": {"user_id": "80ee9b71-3db4-4492-959c-c1e74e865b6d", "name": "Jeffersen"} }{ec2-user@ip-10-0-31-166 ~}$ 

```

WebAppData

TOPICS (1)

Filter by tag Any tag key

Filter by tag value Any tag value

Find tables

WebAppData

▼ Scan or query items

Scan Query

Select a table or index Table - WebAppData

Select attribute projection All attributes

► Filters - optional

Run Reset

Completed - Items returned: 2 - Items scanned: 2 - Efficiency: 100% - RCU's consumed: 2

Table: WebAppData - Items returned (2)

Scan started on December 25, 2025, 20:15:59

	user_id (String)	name (String)	status
<input type="checkbox"/>	80ee9b71-3db4-4492...	Jeffersen	
<input type="checkbox"/>	jeffersen	cloud-project	active

15) In EC2 connect enter the website code

(I changed the security group settings in order to SSH into the instance instead of using session manager in this place)

```

      #_
     ~\ _###_          Amazon Linux 2023
    ~~ \_####\ \
    ~~  \###| \
    ~~   \#/  ____ https://aws.amazon.com/linux/amazon-linux-2023
    ~~    V~' '-->
    ~~~   /
    ~~ .- /_/
    /m/ .-
last login: Thu Dec 25 14:43:03 2025 from 18.206.107.27
[ec2-user@ip-10-0-31-166 ~]$ sudo mkdir -p /var/www/html
[ec2-user@ip-10-0-31-166 ~]$ cd /var/www/html
[ec2-user@ip-10-0-31-166 html]$ sudo nano index.html
[ec2-user@ip-10-0-31-166 html]$ ls
index.html
[ec2-user@ip-10-0-31-166 html]$ sudo nano app.js
[ec2-user@ip-10-0-31-166 html]$ ls
app.js  index.html
[ec2-user@ip-10-0-31-166 html]$ sudo yum install httpd -y
Last metadata expiration check: 1 day, 6:29:11 ago on Thu Dec 25 04:35:28 2025.
Package httpd-2.4.65-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-31-166 html]$ sudo systemctl start httpd
[ec2-user@ip-10-0-31-166 html]$ sudo systemctl enable httpd
[ec2-user@ip-10-0-31-166 html]$ curl localhost

```

▼ Details

Load balancer type Application	Status Active	VPC vpc-0e3848fb897ad3fa6	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDXTRQ7X7K	Availability Zones subnet-04a963ee334de1b69 us-east-1b (use1-az2) subnet-0b593eaff3ff5a525 us-east-1a	Date created December 24, 2025, 18:13 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:390402567136:loadbalancer/app/WebApp-ALB/66f0fe75e49e8089	DNS name copied WebApp-ALB-468916764.us-east-1.elb.amazonaws.com (A Record)		

Listeners and rules Network mapping Resource map Security Monitoring Integrations Attributes Capacity Tags

Listeners and rules (1) Info Manage rules Manage listener Add listener

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

← → ⚡ ⚠ Not secure [webapp-alb-468916764.us-east-1.elb.amazonaws.com](#)

Array Leaders | Prac... Quiz - Data Integrit... Solve Fundamentals...

WebApp running via ALB + ASG

16) Auto healing is enabled and scaled in and out as required

Find Instance by attribute or tag (case-sensitive)

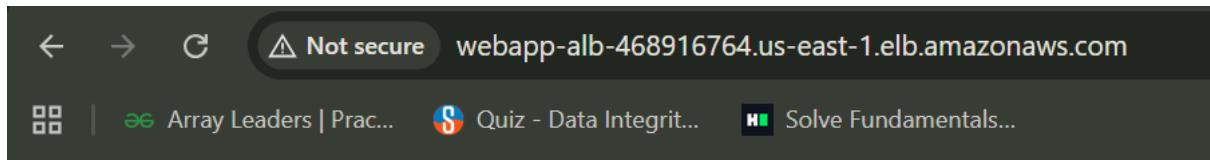
All states ▾

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
<input type="checkbox"/>	i-06e0b3ef72a77f5da	i-06e0b3ef72a77f5da	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-44-204-156-84.co...	44.204
<input type="checkbox"/>	i-07f4ad5abab148b0c	i-07f4ad5abab148b0c	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-44-202-133-21.co...	44.202

Activity history (100+)

Filter activity history

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-07f4ad5abab148b0c	At 2025-12-26T11:27:20Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 December 26, 04:57:22 PM +05:30	2025 December 26, 04:57:54 PM +05:30
Connection draining in progress	Terminating EC2 instance: i-06e0b3ef72a77f5da - Waiting For ELB Connection Draining.	At 2025-12-26T11:27:20Z an instance was taken out of service in response to an ELB system health check failure.	2025 December 26, 04:57:20 PM +05:30	
Successful	Launching a new EC2 instance: i-06e0b3ef72a77f5da	At 2025-12-25T04:34:37Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 December 25, 10:04:38 AM +05:30	2025 December 25, 10:05:10 AM +05:30
Successful	Terminating EC2 instance: i-	At 2025-12-25T04:34:36Z an instance was taken out of service in response to an ELB system health	2025 December 25,	2025 December 25,



WebApp running via ALB + ASG

Now let's move to terraform since we are familiar with the details and flow as well as the configuration. In the current version with terraform, backend data is not replicated, so there is a risk of data loss during a full regional outage. This was an intentional Pilot Light design to focus on infrastructure automation and failover. **Pilot Light disaster recovery strategy**, where critical infrastructure is pre-provisioned across regions, but stateful backend services are activated only when required.

17) In terminal enter **terraform init**, **terraform validate** commands after scripting for creating new setup and give **terraform apply** command.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ... +

```
○ PS E:\Projects\aws-multi-region-auto-healing> Set-Location -Path 'E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform' ; terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/aws v5.100.0...
```

```
'E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform' ; terraform init
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform> terraform validate
Success! The configuration is valid.

PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform>
```

IAM > Users > Jeffersen

Identity and Access Management (IAM)

Multi-factor authentication (MFA) (0)

Access keys (0)

API keys for Amazon Bedrock (0)

Actions ▾ Generate API Key

CloudShell Feedback Console Mobile App

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    ...    +  
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut  
o-healing\7_terraform> aws configure  
>>  
FMQDx  
Default region name [None]: us-east-1  
Default output format [None]: json  
○ PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut  
o-healing\7_terraform> █  
Ln 1, Col 1    Spaces: 0
```

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut  
o-healing\7_terraform> terraform apply  
  
Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.  
  
Enter a value: yes █
```

18) Then use terraform to create VPC and subnets

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut  
o-healing\7_terraform> terraform apply  
public_route_table_id = "rtb-0756f6ed26bb05724"  
public_subnet_1_id = "subnet-0a8d7f82b6d778152"  
public_subnet_2_id = "subnet-07392eba8c5c70a04"  
vpc_id = "vpc-055f46c4ba74c5f63"  
○ PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut  
o-healing\7_terraform> █
```

Your VPCs

VPCs | VPC encryption controls

Your VPCs (3) Info

Find VPCs by attribute or tag

<input type="checkbox"/>	Name	VPC ID	State	Encryption c...	Encrypti
<input type="checkbox"/>	-	vpc-05cf6d0c539860401	Available	-	-
<input type="checkbox"/>	WebApp-VPC-vpc	vpc-0e3848fb897ad3fa6	Available	-	-
<input type="checkbox"/>	multi-region-vpc	vpc-055f46c4ba74c5f63	Available	-	-



Subnets (6) Info

Find subnets by attribute or tag

<input type="checkbox"/>	Name	Subnet ID	State	VPC
<input type="checkbox"/>	public-subnet-2	subnet-07392eba8c5c70a04	Available	vpc-055f46c4ba74c5f63 multi..
<input type="checkbox"/>	WebApp-VPC-subnet-public2-us-east-1b	subnet-04a963ee334de1b69	Available	vpc-0e3848fb897ad3fa6 Web...
<input type="checkbox"/>	public-subnet-1	subnet-0a8d7f82b6d778152	Available	vpc-055f46c4ba74c5f63 multi..
<input type="checkbox"/>	WebApp-VPC-subnet-public1-us-east-1a	subnet-0b593eaff3ff5a525	Available	vpc-0e3848fb897ad3fa6 Web...
<input type="checkbox"/>	port2	subnet-084b300117329f3a5	Available	vpc-05cf6d0c539860401
<input type="checkbox"/>	port1	subnet-087b5c89e67a0050b	Available	vpc-05cf6d0c539860401



19) Security Group is created followed by the other creations.

AWS-MULTI-REGION-AUTO-HEALING	aws-multi-region-auto-healing > 7_terraform > security_groups.tf
└ aws-multi-region-auto-healing	
└ 1_docs	
└ 2_architecture	
└ 3_app	
└ 4_lambda	
└ 5_screenshots	
└ 6_notes	
└ learning-notes.md	
└ 7_terraform	
└ .terraform\providers\registry.terra...	
└ LICENSE.txt	
└ terraform-provider-aws_v5.100.0_x5...	
└ .terraform.lock.hcl	
└ outputs.tf	
└ provider.tf	
└ routes.tf	
└ security_groups.tf	
└ subnets.tf	
└ terraform.tfstate	
└ variables.tf	
└ vpc.tf	
└ README.md	

```

1  # ALB Security Group
2  resource "aws_security_group" "alb_sg" {
3    name          = "alb-sg"
4    description   = "Allow HTTP from internet"
5    vpc_id        = aws_vpc.main.id
6
7    ingress {
8      description = "HTTP from Internet"
9      from_port   = 80
10     to_port     = 80
11     protocol    = "tcp"
12     cidr_blocks = ["0.0.0.0/0"]
13   }
14
15   egress {
16     from_port   = 0
17     to_port     = 0
18     protocol    = "-1"
19     cidr_blocks = ["0.0.0.0/0"]
20   }
21
22   tags = {
23     Name = "alb-sg"
24   }

```

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform plan
Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# aws_security_group.alb_sg will be created
+ resource "aws_security_group" "alb_sg" {
    + arn                      = (known after apply)
    + description              = "Allow HTTP from internet"
    + egress                   = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
            ]
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids = []
            + protocol         = "-1"
            + security_groups  = []
            + self              = false
            + to_port           = 0
            # (1 unchanged attribute hidden)
        },
    ]
    + id                      = (known after apply)
    + ingress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
            ]
            + description      = "HTTP from Internet"
            + from_port        = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids = []
        }
    ]
}
```

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform> terraform apply
>>
    },
]
+ name                  = "ec2-sg"
+ name_prefix          = (known after apply)
+ owner_id              = (known after apply)
+ revoke_rules_on_delete = false
+ tags                  = {
    + "Name" = "ec2-sg"
}
+ tags_all              = {
    + "Name" = "ec2-sg"
}
+ vpc_id                = "vpc-055f46c4ba74c5f63"
}


```

```
Plan: 2 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
```

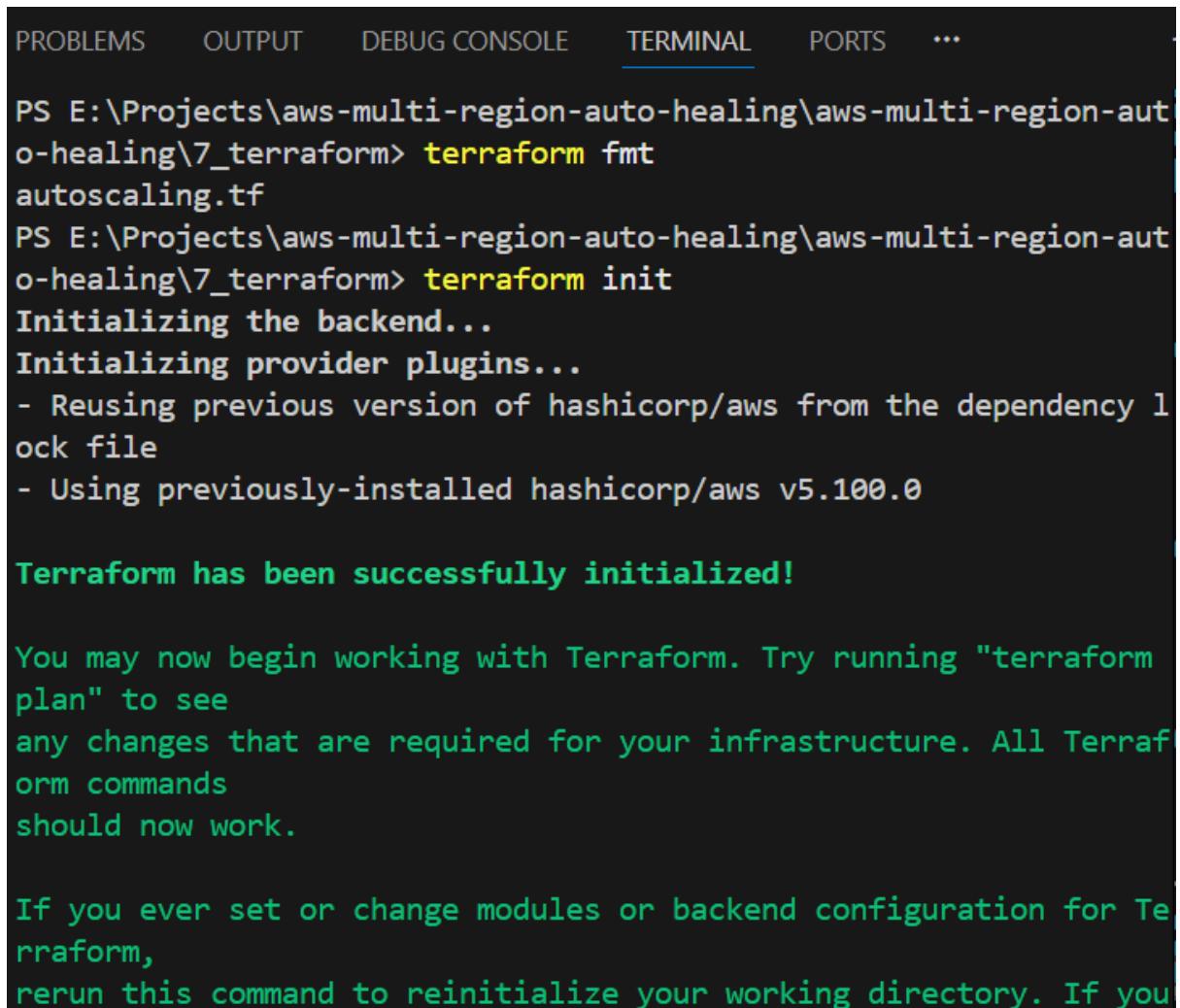
```
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.
```

```
Enter a value: yes
```

```
aws_security_group.alb_sg: Still creating... [00m10s elapsed]
aws_security_group.alb_sg: Creation complete after 10s [id=sg-094
...
aws_security_group.alb_sg: Still creating... [00m10s elapsed]
aws_security_group.alb_sg: Creation complete after 10s [id=sg-094
fb86dd55e907a8]
aws_security_group.ec2_sg: Creating...
aws_security_group.ec2_sg: Creation complete after 8s [id=sg-0405
cf6a4babf18c1]
```

ec2-sg	sg-0405cf6a4babf18c1	ec2-sg
-	sg-0227db797abe6605f	WebApp-ALB-SG
-	sg-061df2f8f7f864e9f	default
-	sg-0fff9aaf4ba6e5e0	db-init-sg
alb-sg	sg-094fb86dd55e907a8	alb-sg

20) Then create ASG using terraform.



```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    ...
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform fmt
autoscaling.tf
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency l
ock file
- Using previously-installed hashicorp/aws v5.100.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform
plan" to see
any changes that are required for your infrastructure. All Terrafo
rm commands
should now work.

If you ever set or change modules or backend configuration for Te
rraform,
rerun this command to reinitialize your working directory. If you

```

```
21  Output  public_route_table_id  ⌂
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ... +

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform init
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform plan
var.key_name
  EC2 Key Pair Name

  Enter a value: defo

aws_vpc.main: Refreshing state... [id=vpc-055f46c4ba74c5f63]
aws_subnet.public_1: Refreshing state... [id=subnet-0a8d7f82b6d77
8152]
aws_internet_gateway.igw: Refreshing state... [id=igw-00099113734
5d0dce]
aws_subnet.public_2: Refreshing state... [id=subnet-07392eba8c5c7
0a04]
aws_security_group.alb_sg: Refreshing state... [id=sg-094fb86dd55
e907a8]
aws_route_table.public_rt: Refreshing state... [id=rtb-0756f6ed26
bb05724]
aws_route_table_association.public_1_assoc: Refreshing state... [
id=rtbassoc-0d41da6741408eeba]
aws_route_table_association.public_2_assoc: Refreshing state... [
id=rtbassoc-0d3fe7d84c9d610fd]
aws_security_group.ec2_sg: Refreshing state... [id=sg-0405cf6a4ba
bf18c1]

Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ... +

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform> terraform apply
+ create

Terraform will perform the following actions:

# aws_autoscaling_attachment.asg_attach will be created
+ resource "aws_autoscaling_attachment" "asg_attach" {
    + autoscaling_group_name = "web-asg"
    + id                      = (known after apply)
    + lb_target_group_arn     = (known after apply)
}

# aws_autoscaling_group.web_asg will be created
+ resource "aws_autoscaling_group" "web_asg" {
    + arn                    = (known after apply)
    + availability_zones     = (known after apply)
    + default_cooldown       = (known after apply)
    + desired_capacity        = 2
    + force_delete            = false
    + force_delete_warm_pool  = false
    + health_check_grace_period = 300
    + health_check_type      = "EC2"
    + id                      = (known after apply)
    + ignore_failed_scaling_activities = false
    + load_balancers          = (known after apply)
    + max_size                = 3
    + metrics_granularity     = "1Minute"
    + min_size                = 1
}
```

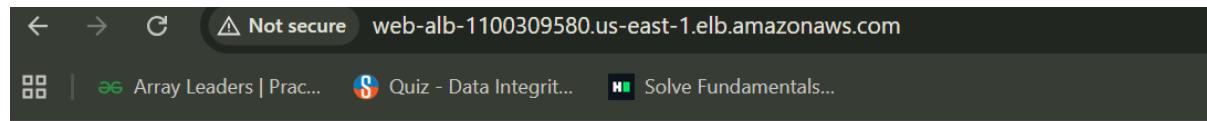
Auto Scaling groups (2) Info								
Last updated less than a minute ago C Launch configurations Launch templates Actions Create Auto Scaling group								
<input type="text"/> Search your Auto Scaling groups								
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	
web-asg	web-lt-20251226162545799300000001	2	-	2	1	3	2 Availability Zones	Actions
WebApp-ASG	WebApp-LT Version 3	1	-	1	1	1	2 Availability Zones	Actions

Load balancers (2) What's new?								
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.								
<input type="text"/> Filter load balancers								
Name	State	Type	Scheme	IP address type	VPC ID	Availability Zones	Security groups	
WebApp-ALB	Active	application	Internet-facing	IPv4	vpc-0e3848fb897ad3fa6	2 Availability Zones	sg-0227db797abi	Actions
web-alb	Active	application	Internet-facing	IPv4	vpc-055f46c4ba74c5f63	2 Availability Zones	sg-094fb86dd55c	Actions

Target groups (2) Info What's new?						
<input type="text"/> Filter target groups						
<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer
<input type="checkbox"/>	WebApp-TG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/WebApp-TG/5678901234567890	80	HTTP	Instance	WebApp-ALB
<input type="checkbox"/>	web-tg	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/web-tg/5678901234567890	80	HTTP	Instance	web-alb

Launch Templates (3) Info						
<input type="text"/> Search						
<input type="checkbox"/>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
<input type="checkbox"/>	lt-0ddd80bcf73eb449	320421174	1	1	2025-12-21T06:09:57.000Z	arn:aws:iam::390402567136:us...
<input type="checkbox"/>	lt-07503d2e61bb8bd5	web-lt-2025122616254579930...	1	1	2025-12-26T16:25:46.000Z	arn:aws:iam::390402567136:us...
<input type="checkbox"/>	lt-0631880ab255526e7	WebApp-LT	1	3	2025-12-24T11:45:25.000Z	arn:aws:iam::390402567136:us...

21) ALB DNS works well



Auto Scaling Web App - Terraform

22) Our aim is to create a secondary region for multi region implementation for disaster recovery.

```
notes  
learning-notes.md  
terraform  
terraform\providers\registry.terra... ●  
LICENSE.txt U  
terraform-provider-aws_v5.100.0_x5... U  
terraform.lock.hcl U  
lb.tf U  
autoscaling.tf U  
outputs.tf U  
provider.tf U  
outes.tf ●  
security_groups.tf U  
ubnets.tf U  
terraform.tfstate U  
terraform.tfstate.backup U  
9  
10 provider "aws" {  
11   region = "us-east-1"  
12 }  
13  
14 provider "aws" {  
15   alias  = "use1"  
16   region = "us-east-1"  
17 }  
18  
19 provider "aws" {  
20   alias  = "usw2"  
21   region = "us-west-2"  
22 }  
23 | Amazon Q Tip 2/3: Invoke s
```

```
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform> terraform init
- Reusing previous version of hashicorp/aws from the dependency l
ock file
- Using previously-installed hashicorp/aws v5.100.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform
plan" to see
any changes that are required for your infrastructure. All Terraf
orm commands
should now work.

If you ever set or change modules or backend configuration for Te
rraform,
rerun this command to reinitialize your working directory. If you
forget, other
commands will detect it and remind you to do so if necessary.
○ PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-aut
o-healing\7_terraform>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... terraform - 7_terraform + ▾ []

PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7_terraform> terraform apply
error: [id=rtbassoc-0096162a0753f8cc9]
aws_launch_template.web_lt_secondary: Modifications complete after 9s [id=1
t-0b88fe8b9cac30967]
aws_autoscaling_group.secondary_asg: Creating...
aws_lb.secondary_alb: Still creating... [00m10s elapsed]
aws_lb.secondary_alb: Still creating... [00m20s elapsed]
aws_lb.secondary_alb: Still creating... [00m30s elapsed]
aws_lb.secondary_alb: Still creating... [00m40s elapsed]
aws_lb.secondary_alb: Still creating... [00m50s elapsed]
aws_lb.secondary_alb: Still creating... [01m00s elapsed]
aws_lb.secondary_alb: Still creating... [01m10s elapsed]
aws_lb.secondary_alb: Still creating... [01m20s elapsed]
aws_lb.secondary_alb: Still creating... [01m30s elapsed]
aws_lb.secondary_alb: Still creating... [01m40s elapsed]
aws_lb.secondary_alb: Still creating... [01m50s elapsed]
aws_lb.secondary_alb: Still creating... [02m00s elapsed]
aws_lb.secondary_alb: Still creating... [02m10s elapsed]
aws_lb.secondary_alb: Still creating... [02m20s elapsed]
aws_lb.secondary_alb: Still creating... [02m30s elapsed]
```

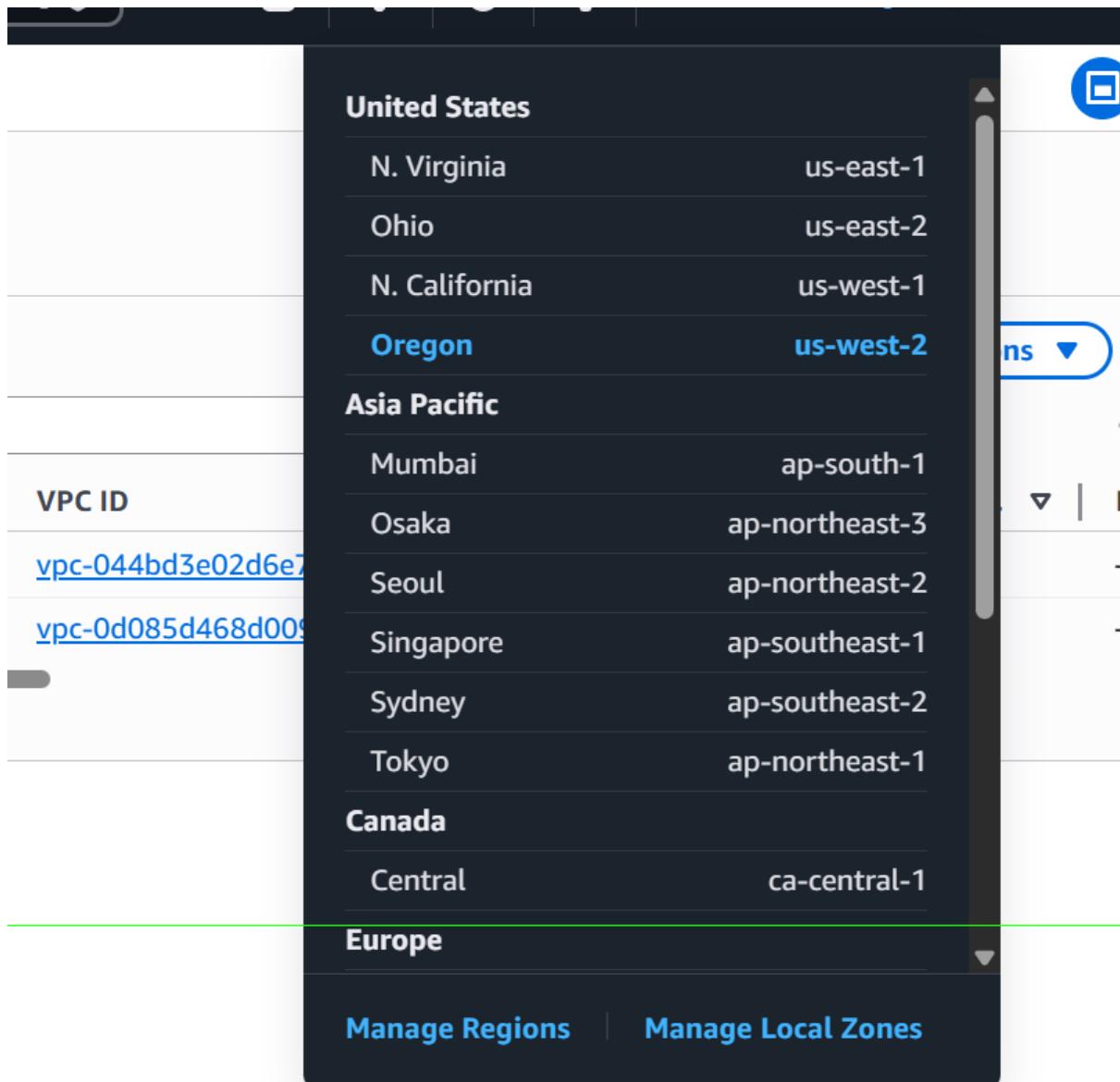
```
aws_autoscaling_group.secondary_asg: Still creating... [00m30s elapsed]
aws_autoscaling_group.secondary_asg: Still creating... [00m40s elapsed]
aws_autoscaling_group.secondary_asg: Creation complete after 45s [id=secondary-asg]
```

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Outputs:

```
alb_dns_name = "web-alb-1100309580.us-east-1.elb.amazonaws.com"
internet_gateway_id = "igw-000991137345d0dce"
primary_alb_dns = "web-alb-1100309580.us-east-1.elb.amazonaws.com"
public_route_table_id = "rtb-0756f6ed26bb05724"
public_subnet_1_id = "subnet-0a8d7f82b6d778152"
public_subnet_2_id = "subnet-07392eba8c5c70a04"
secondary_alb_dns = "secondary-alb-543721480.us-west-2.elb.amazonaws.com"
vpc_id = "vpc-055f46c4ba74c5f63"
PS E:\Projects\aws-multi-region-auto-healing\aws-multi-region-auto-healing\7 terraform> █
```

23) We implement the same architecture in secondary region also.



Your VPCs

VPCs | VPC encryption controls

Your VPCs (2) Info Last updated less than a minute ago Actions Create VPC

Name	VPC ID	State	Encryption c...	Encryption contr...
-	vpc-044bd3e02d6e7ca09	Available	-	-
secondary-vpc	vpc-0d085d468d009faf1	Available	-	-

Subnets (6) Info Last updated less than a minute ago Actions Create subnet

Name	Subnet ID	State	VPC
secondary-public-subnet-2	subnet-09c0702bac09fb04e	Available	vpc-0d085d468d009faf1 seco...
-	subnet-0d2e44d350575cdb3	Available	vpc-044bd3e02d6e7ca09
-	subnet-01e2919a796a71970	Available	vpc-044bd3e02d6e7ca09
	subnet-04652007f00fa01e	Available	vpc-0d085d468d009faf1 sec...
	subnet-04652007f00fa01e	Available	vpc-0d085d468d009faf1 sec...
	subnet-04652007f00fa01e	Available	vpc-0d085d468d009faf1 sec...

Account ID: 3904-0256-7
Jefferson

Auto Scaling groups (1) Info

Last updated less than a minute ago

Search your Auto Scaling groups 1

<input type="checkbox"/> Name	Launch template/configuration	Instances	Status
<input type="checkbox"/> secondary-asg	web-lt-secondary-202512261745495732	2	-

Account ID: 3904-0256-7136
Jefferson

Instances (2) Info

Find Instance by attribute or tag (case-sensitive) All states

<input type="checkbox"/> Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/> secondary-asg	i-05205274e4ce21122	Running	t2.micro	Initializing	View alarms +
<input type="checkbox"/> secondary-asg	i-03a9218c0d1c967df	Running	t2.micro	Initializing	View alarms +

Select an instance

aws Account ID: 3904-0256-7136
Jefferson

EC2 > Load balancers > secondary-alb

AMIs
AMI Catalog

Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

Load Balancing
Load Balancers

Scheme Internet-facing	Hosted zone Z1H1FL5HABSF5	Availability Zones subnet-09c0702bac09fb04e us-west-2b (usw2-az1) subnet-0fb5bc983f08f4945 us-west-2a (usw2-az2)	Date created December 26, 2025, 23:33 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-west-2:390402567136:loadbalancer/app/secondary-alb/f8b042681a9e68db		DNS name copied secondary-alb-543721480.us-west-2.elb.amazonaws.com (A Record)	

Listeners and rules (0) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default

24) Since we have different DNS we need a common link for multi region access we use route 53 for this. Created hosted zone and traffic policy.

Route 53 > Hosted zones > Create hosted zone

Create hosted zone Info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name Info
This is the name of the domain that you want to route traffic for.
 Valid characters: a-z, 0-9, !# \$ % & ' () * + , - / ; ; < = > ? @ [\] ^ _ ` { } , ~

Description - optional Info
This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

Type Info
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.
 Public hosted zone
A public hosted zone determines how traffic is routed on the internet.
 Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Route 53 > Traffic policies > Create traffic policy

Create a new traffic policy X

Name
 The name can have up to 512 characters.

Description - optional
A short description to help identify the traffic policy among other versions when creating a policy record.
 The description can have up to 1,024 characters.

Cancel **Next**

Route 53 > Traffic policies > Create traffic policy

multi-region-alb-policy Info Version 1 Draft **Create policy**

Properties

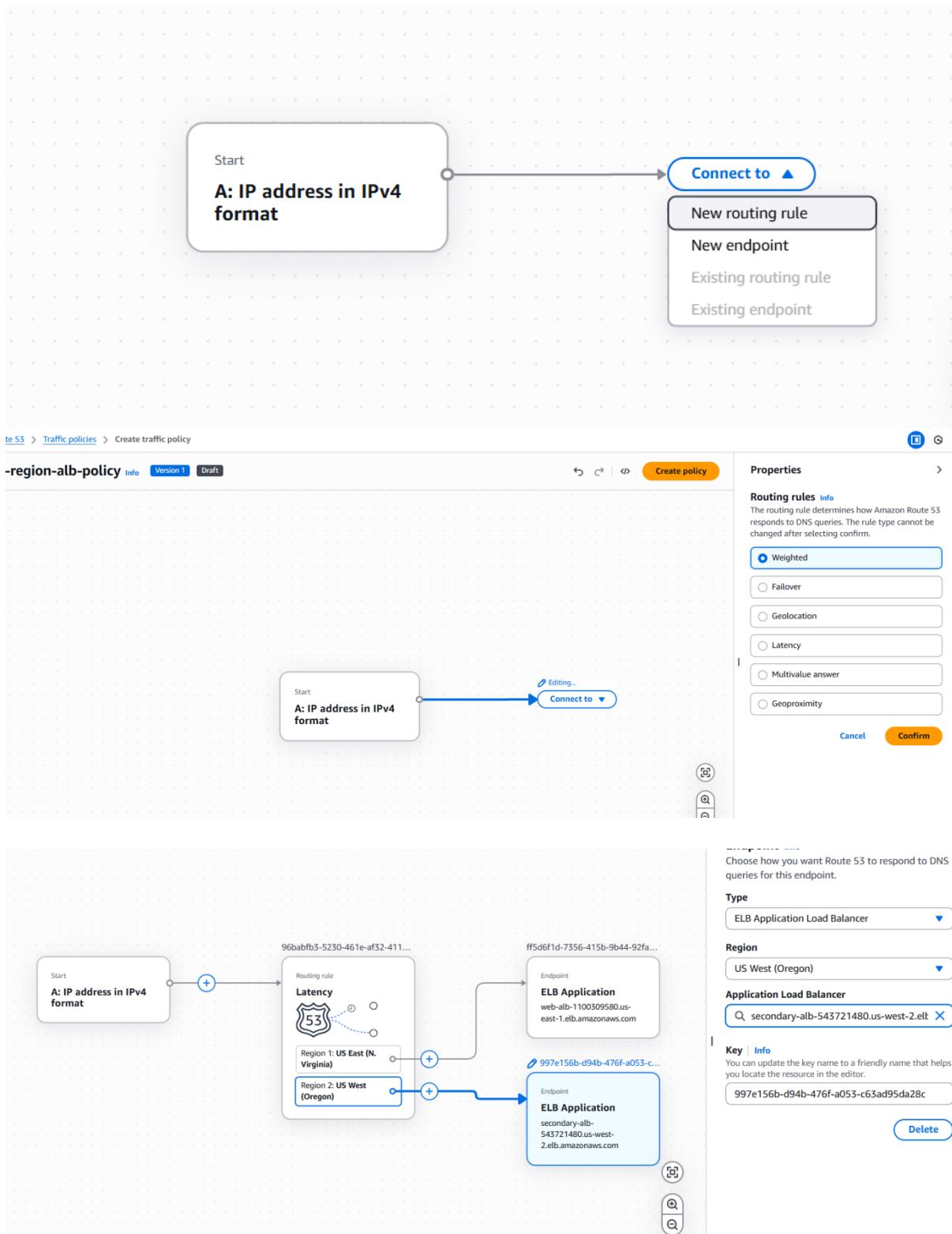
Select a DNS record type Info
Select the DNS record type you want to assign to all the resource record sets created by this traffic policy.

A: IP address in IPv4 format

AAAA: IP address in IPv6 format
 CNAME: Canonical name
 MX: Mail exchange
 PTR: Pointer
 SPF: Sender Policy Framework
 SRV: Service locator
 TXT: Text

Editing... **+ Choose DNS record type**

Cancel **Confirm**



Now our application is auto-healing as well as multi-region accessible.