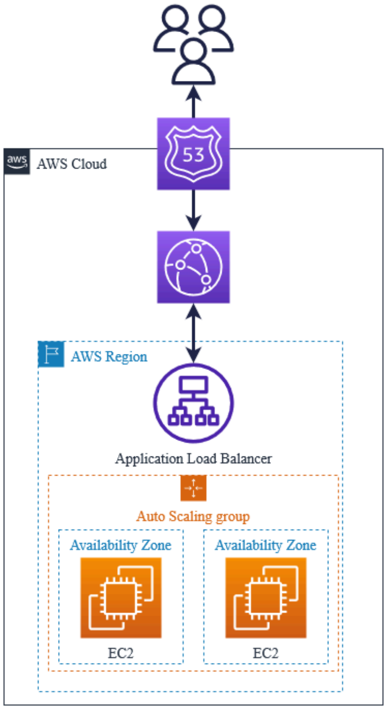
Runbook: Setting Up a Dynamic Website with Route 53, CloudFront

Introduction

This runbook provides a detailed step-by-step guide to set up a dynamic website using Amazon EC2, CloudFront, ALB, and configure Amazon Route 53 with specific routing policies.

Architecture :



By the end of this runbook, you will:

- Set up EC2 instances for your website. ((VPC/subnet required. Use VPC and more to create infrastructure)
- Configure a target group and an application load balancer.
- Request a public certificate using AWS Certificate Manager (ACM).
- Create a CloudFront distribution.
- Configure Amazon Route 53 to route DNS traffic to your CloudFront distribution using specific routing policies.
- Understand and apply specific routing policies in Route 53.

Steps

1. Spin up EC2 instances.
2. Create a target group.

3. Create an application load balancer.
4. Request a public certificate in ACM.
5. Create a CloudFront distribution.
6. Configure routing policies in Route 53 and create records.
7. Test your website.

Step 1: Spin Up EC2 Instances in Pub subnet

Create EC2 Instances

1. **Navigate to EC2 Console:**
   - Open the AWS Management Console.
   - Go to the EC2 console by searching for "EC2" in the search bar.
2. **Launch EC2 Instances :**
   - Click on "Launch Instance".
   - Choose an AMI (Amazon Linux 2 is recommended).
   - Choose an instance type (t2.micro is sufficient for this exercise).
   - Configure instance details:
     - Network: Choose your VPC
     - Subnet: Choose a public subnet and enable Pub IP
     - Add the following user data to each instance:

User Data for Server 1:

Copy code

```
#!/bin/bash
yum update -y
yum install httpd -y
cd /var/www/html
echo "Today, The Awesome JJTech Model batch B students are now studying Route53, and will explore all the routing policies including failover based routing, weighted based routing, latency based routing, geolocation and simple based routing policies" > index.html
service httpd start
chkconfig httpd on
```

User Data for Server 2:

Copy code

```
#!/bin/bash
yum update -y
yum install httpd -y
cd /var/www/html
echo "JJ Tech Inc Disaster Recovery strategy include using Route53 Failover Based Routing" > index.html
service httpd start
chkconfig httpd on
```

3. **Configure Security Group:**
   - Add rules to allow traffic on port 22 (SSH), port 443 (HTTPS) and port 80 (HTTP) from anywhere.
4. **Launch Instances:**
   - Review and launch the instances.

Step 2: Create a Target Group

Create Target Group

1. **Navigate to Target Groups:**
   - Open the AWS Management Console.
   - Go to the EC2 console, then select "Target Groups" from the left-hand menu.
2. **Create a Target Group:**
   - Click "Create target group".
   - Choose "Instances" as the target type.
   - Give your target group a name (e.g., r53TargetGroup).
   - Choose the protocol as HTTP and port 80.
   - Select your VPC (the VPC you created)
   - Click "Next".
3. **Register Targets:**
   - Select the instances you created earlier and click "Include as pending below".
   - Click "Create target group".

Step 3: Create an Application Load Balancer

Create Application Load Balancer

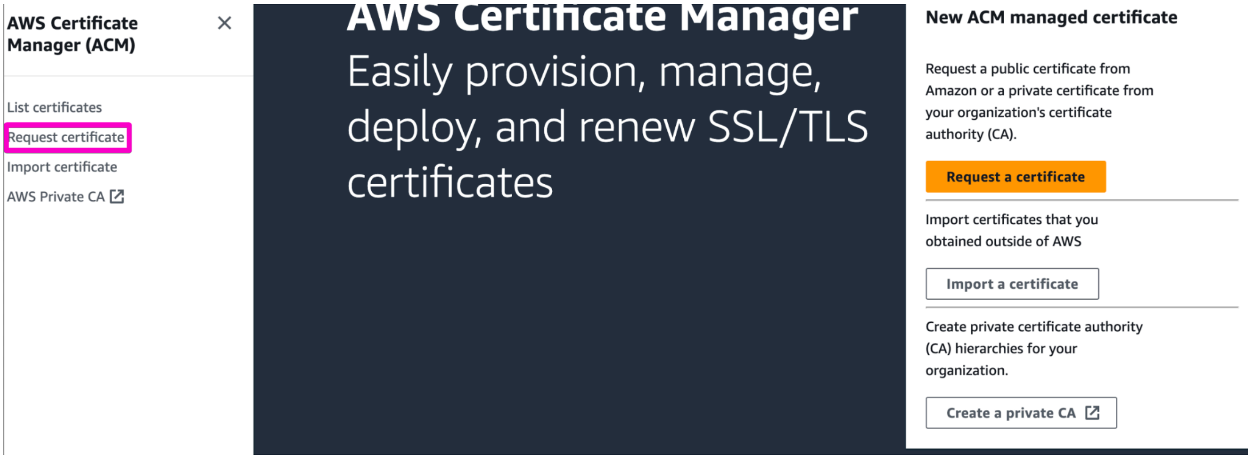
1. **Navigate to Load Balancers:**
   - Open the AWS Management Console.
   - Go to the EC2 console, then select "Load Balancers" from the left-hand menu.
2. **Create a Load Balancer:**
   - Click "Create Load Balancer".
   - Choose "Application Load Balancer".
   - Give your load balancer a name (e.g., r53ALB).
   - Choose the scheme as "Internet-facing".
   - Choose the IP address type as "IPv4".
   - Select your VPC and enable ateast two Availability Zones by selecting Pub subnets in these AZs
3. **Configure Security Group:**
   - Select the security group you created earlier that allows traffic on port 80.
4. **Configure Listeners and Routing:**
   - Select the default listener protocol as HTTP and port 80.
   - Under Default actions, select your target group.

5. **Review and Create:**
   - Review your configuration and click "Create load balancer".

Step 4: Request a Public Certificate in ACM

Request a Public Certificate

1. **Navigate to ACM Console:**
   - Open the AWS Management Console.
   - Go to the ACM console by searching for "ACM" in the search bar.
2. **Request a Certificate:**
   - Click "Request a certificate".



AWS Certificate Manager

Easily provision, manage, deploy, and renew SSL/TLS certificates

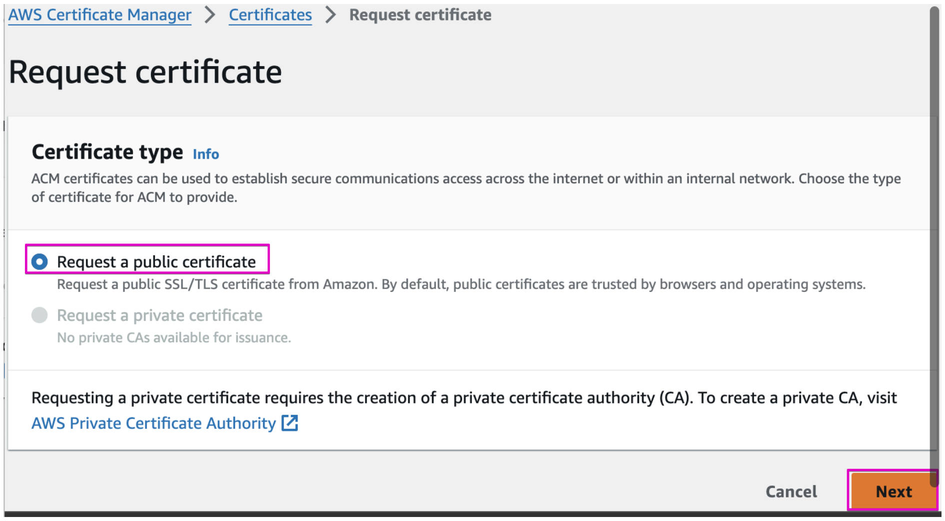
New ACM managed certificate

Request a public certificate from Amazon or a private certificate from your organization's certificate authority (CA).

Import certificates that you obtained outside of AWS

Create private certificate authority (CA) hierarchies for your organization.

   - Choose "Request a public certificate" and then click "Next".


AWS Certificate Manager > Certificates > Request certificate
Info
Cancel
Next

Request certificate

Certificate type

ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

● Request a public certificate

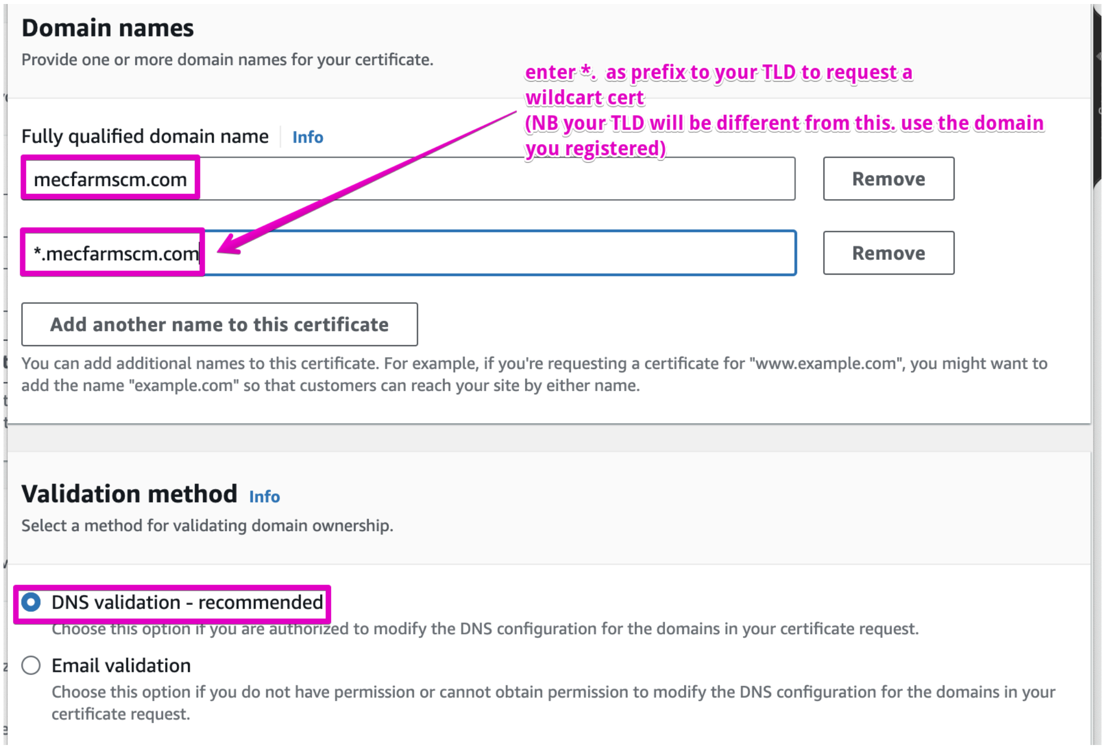
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

○ Request a private certificate

No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority

3. **Enter Domain Names:**
   - Enter your domain name (e.g., jjtech.com).
   - Under "Add another name to this certificate", enter an asterisk in front of the domain name to request a wildcard certificate (e.g., *.jjtech.com).
   - Under Validation Method, choose "DNS validation"



Domain names

Provide one or more domain names for your certificate.

enter *. as prefix to your TLD to request a wildcart cert
(NB your TLD will be different from this. use the domain you registered)

Fully qualified domain name Info

mecfarmscm.com

*.mecfarmscm.com

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Validation method Info

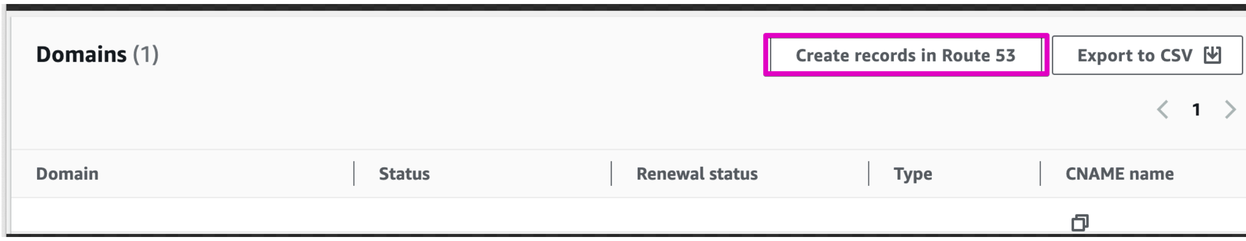
Select a method for validating domain ownership.

● DNS validation - recommended
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

○ Email validation
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

4. Review and **Request:**

5. **Validate Domain:**
   - On the Validation page, expand both domains and choose **"Create record in Route 53**" to automatically add the CNAME records for your domains.



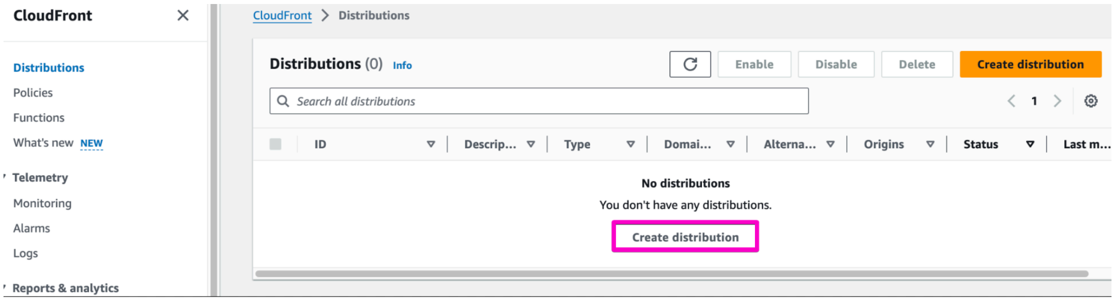
Domains (1)

Domain	Status	Renewal status	Type	CNAME name

Step 5: Create CloudFront Distribution

Create CloudFront Distribution

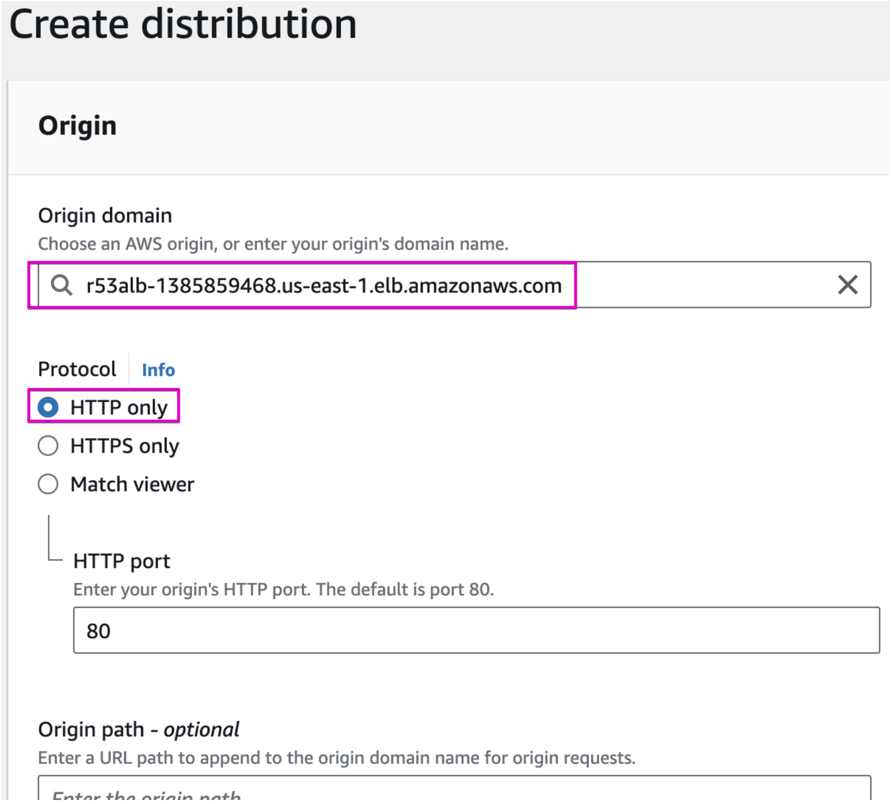
1. **Navigate to CloudFront Console:**
   - Open the AWS Management Console.
   - Go to the CloudFront console by searching for "CloudFront" in the search bar.
2. **Create Distribution:**
   - Click "Create Distribution".



No distributions

You don't have any distributions.

3. **Configure Origin Settings:**
   - For Origin Domain Name, under Elastic Load Balancer, select the load balancer you created earlier.
   - Set Origin Protocol Policy to HTTP only.
   -



Create distribution

Origin

Origin domain

Choose an AWS origin, or enter your origin's domain name.

r53alb-1385859468.us-east-1.elb.amazonaws.com

Protocol Info

● HTTP only

○ HTTPS only

○ Match viewer

HTTP port

Enter your origin's HTTP port. The default is port 80.

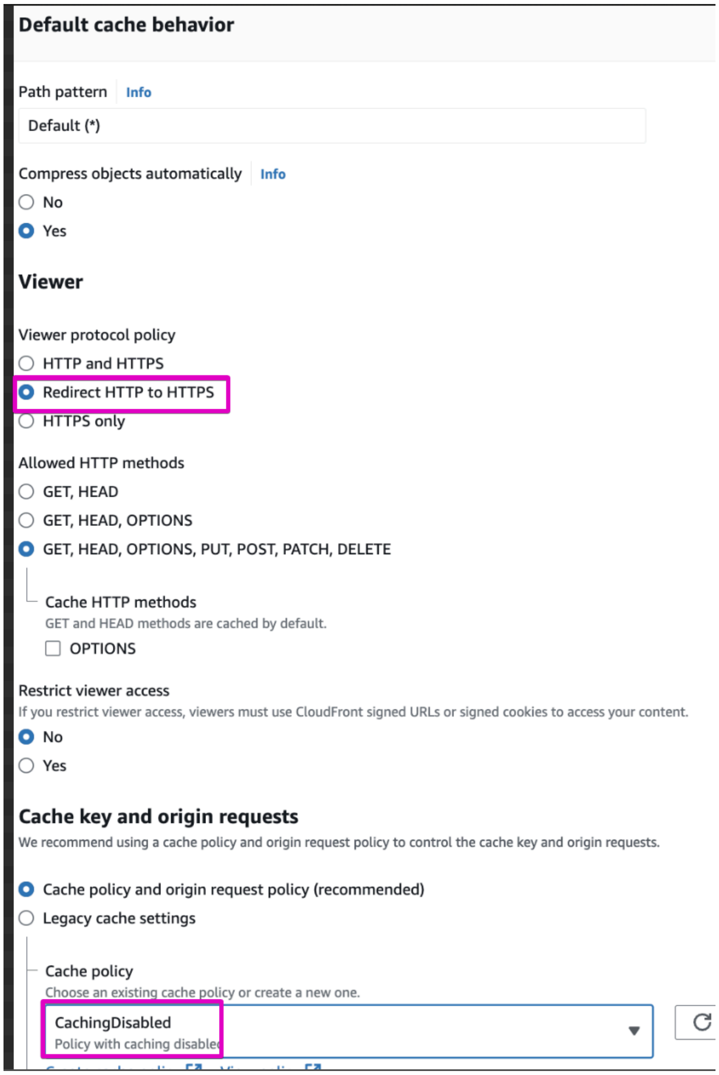
80

Origin path - *optional*

Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

4. **Configure Default Cache Behavior Settings:**
   - Under Viewer, set Viewer protocol policy to "Redirect HTTP to HTTPS".
   - Set Cache settings to "CachingDisabled".



Default cache behavior

Path pattern Info

Default (*)

Compress objects automatically Info

○ No
● Yes

Viewer

Viewer protocol policy

○ HTTP and HTTPS
● Redirect HTTP to HTTPS
○ HTTPS only

Allowed HTTP methods

○ GET, HEAD
○ GET, HEAD, OPTIONS
● GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods
GET and HEAD methods are cached by default.
□ OPTIONS

Restrict viewer access
If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

● No
○ Yes

Cache key and origin requests

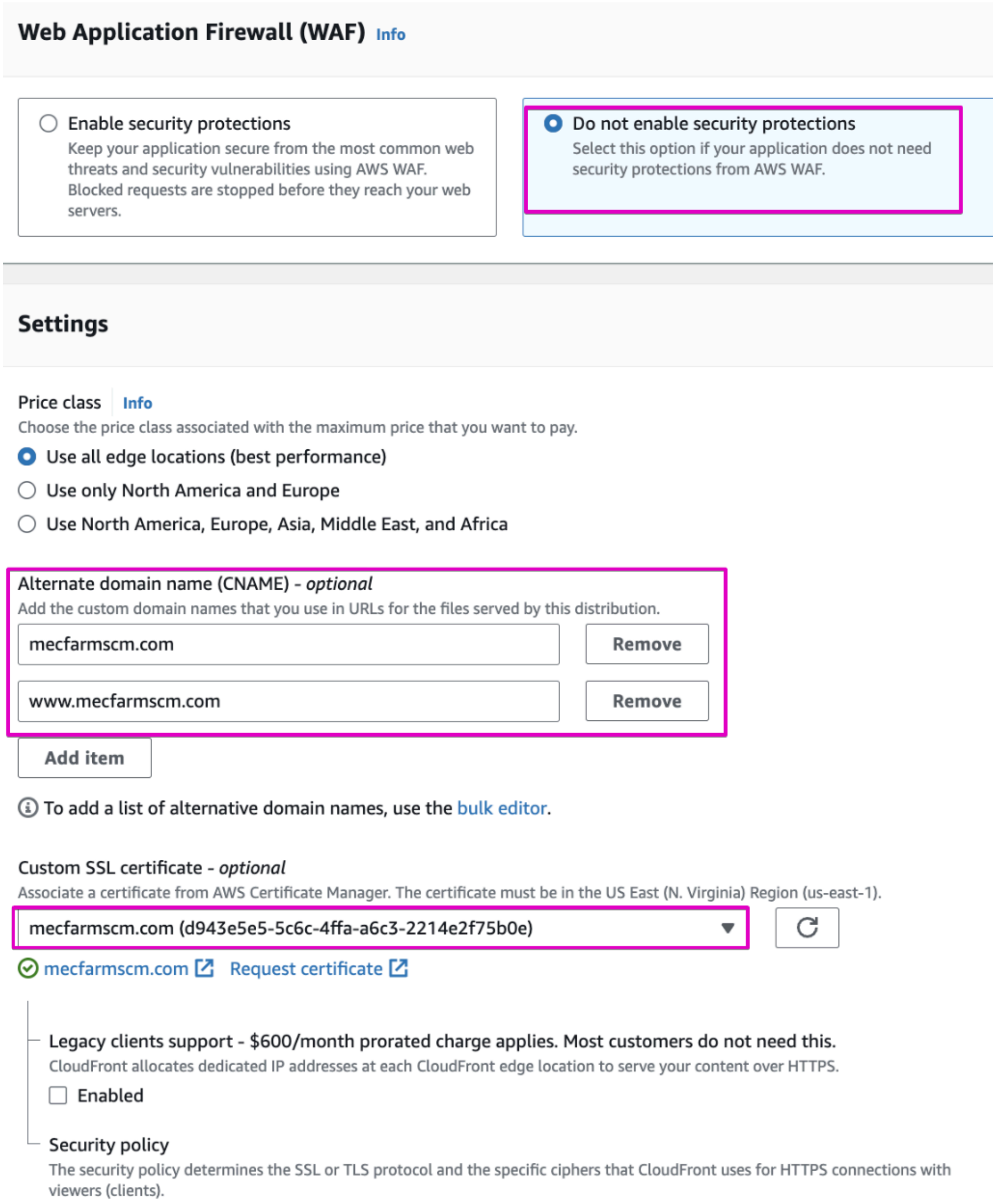
We recommend using a cache policy and origin request policy to control the cache key and origin requests.

● Cache policy and origin request policy (recommended)
○ Legacy cache settings

Cache policy
Choose an existing cache policy or create a new one.

CachingDisabled
Policy with caching disabled

5. **Configure WAF and Distribution Settings:**
   - For WAF, select “Do not enable security protections”
   - Alternate domain name (CNAME), enter your root domain (e.g., jjtech.com).
   - For Custom SSL Certificate, choose the certificate you created in ACM.



Web Application Firewall (WAF) Info

○ Enable security protections
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

● Do not enable security protections
Select this option if your application does not need security protections from AWS WAF.

Settings

Price class Info
Choose the price class associated with the maximum price that you want to pay.

● Use all edge locations (best performance)
○ Use only North America and Europe
○ Use North America, Europe, Asia, Middle East, and Africa

Alternate domain name (CNAME) - *optional*
Add the custom domain names that you use in URLs for the files served by this distribution.

mecfarmscm.com — Remove
www.mecfarmscm.com — Remove

Add item

ⓘ To add a list of alternative domain names, use the bulk editor.

Custom SSL certificate - *optional*
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

mecfarmscm.com (d943e5e5-5c6c-4ffa-a6c3-2214e2f75b0e) ▼

✓ mecfarmscm.com   Request certificate

Legacy clients support - $600/month prorated charge applies. Most customers do not need this.
CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over HTTPS.
☐ Enabled

Security policy
The security policy determines the SSL or TLS protocol and the specific ciphers that CloudFront uses for HTTPS connections with viewers (clients).

- Leave other settings as default.

6. **Create Distribution:**
   - Click "Create Distribution". Wait for the distribution to be deployed before proceeding to the next step.

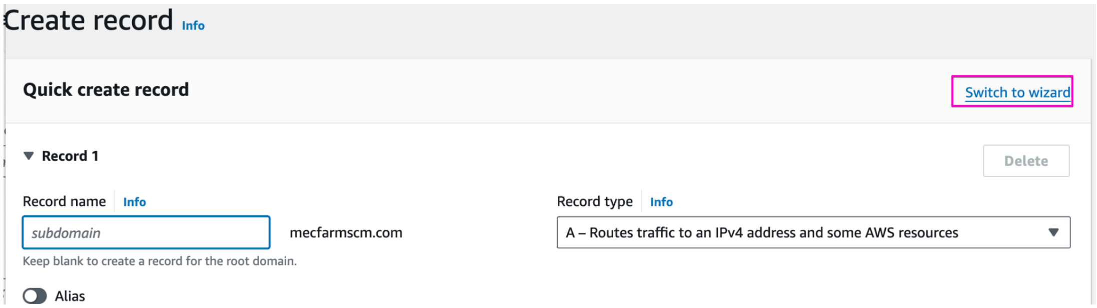
Step 6: Configure Routing Policies in Route 53 and Create Records

Routing Policies Overview

1. **Simple Routing**
   - **Definition**: Directs traffic to a single resource, such as an EC2 instance or an ELB.
   - **Use Case**: Best for single server websites or applications.
2. **Weighted Routing**
   - **Definition**: Distributes traffic across multiple resources based on assigned weights.
   - **Use Case**: Useful for A/B testing or gradual traffic migration.
3. **Latency Routing**
   - **Definition**: Routes traffic to the resource that provides the lowest latency.
   - **Use Case**: Ideal for improving user experience by minimizing latency.
4. **Failover Routing**
   - **Definition**: Automatically switches traffic to a standby resource in case the primary resource becomes unavailable.
   - **Use Case**: Ensures high availability and disaster recovery.

Implementing Routing Policies

Simple Routing

1. **Create Simple Routing Record:**
   - Navigate to the Route 53 console.
   - Go to your hosted zone.
   - Click "Create record" and click on “switch to the wizard”.
   - 

Create record Info

Quick create record

Record 1

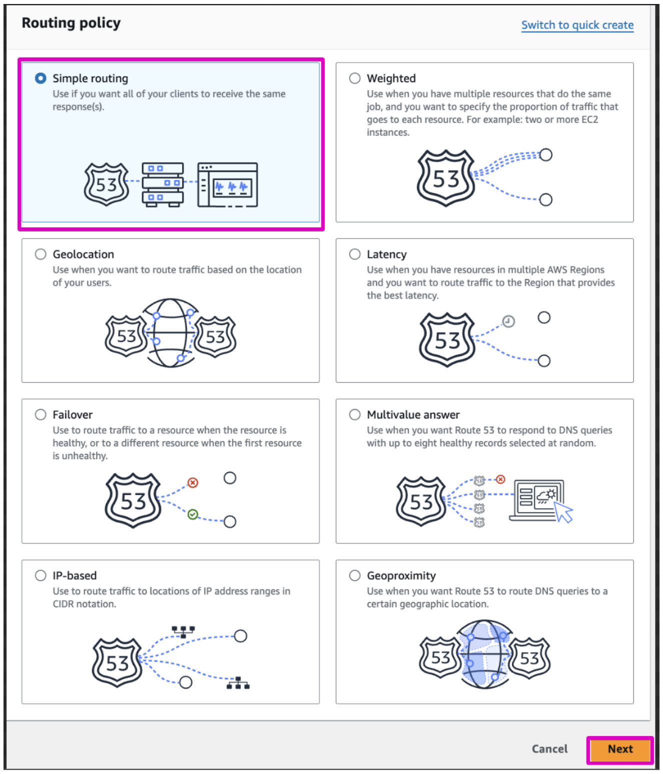
Record name: subdomain mecfarmscm.com

Keep blank to create a record for the root domain.

Record type: A – Routes traffic to an IPv4 address and some AWS resources

Alias

   - Select "Simple routing" policy and click "Next".



Routing policy

Switch to quick create

● Simple routing
Use if you want all of your clients to receive the same response(s).

○ Weighted
Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

○ Geolocation
Use when you want to route traffic based on the location of your users.

○ Latency
Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

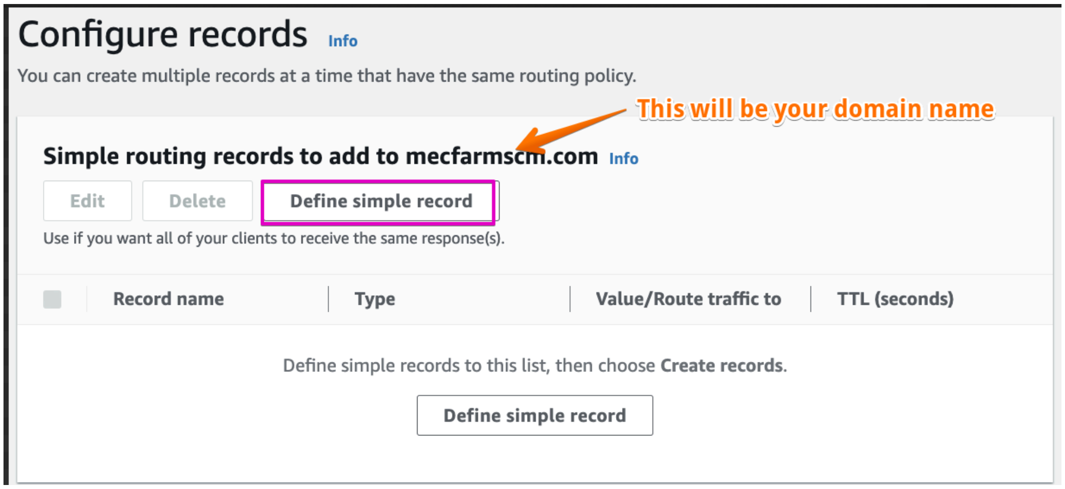
○ Failover
Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

○ Multivalue answer
Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

○ IP-based
Use to route traffic to locations of IP address ranges in CIDR notation.

○ Geoproximity
Use when you want Route 53 to route DNS queries to a certain geographic location.

- Choose "Define simple record".



Configure records Info

You can create multiple records at a time that have the same routing policy.

This will be your domain name

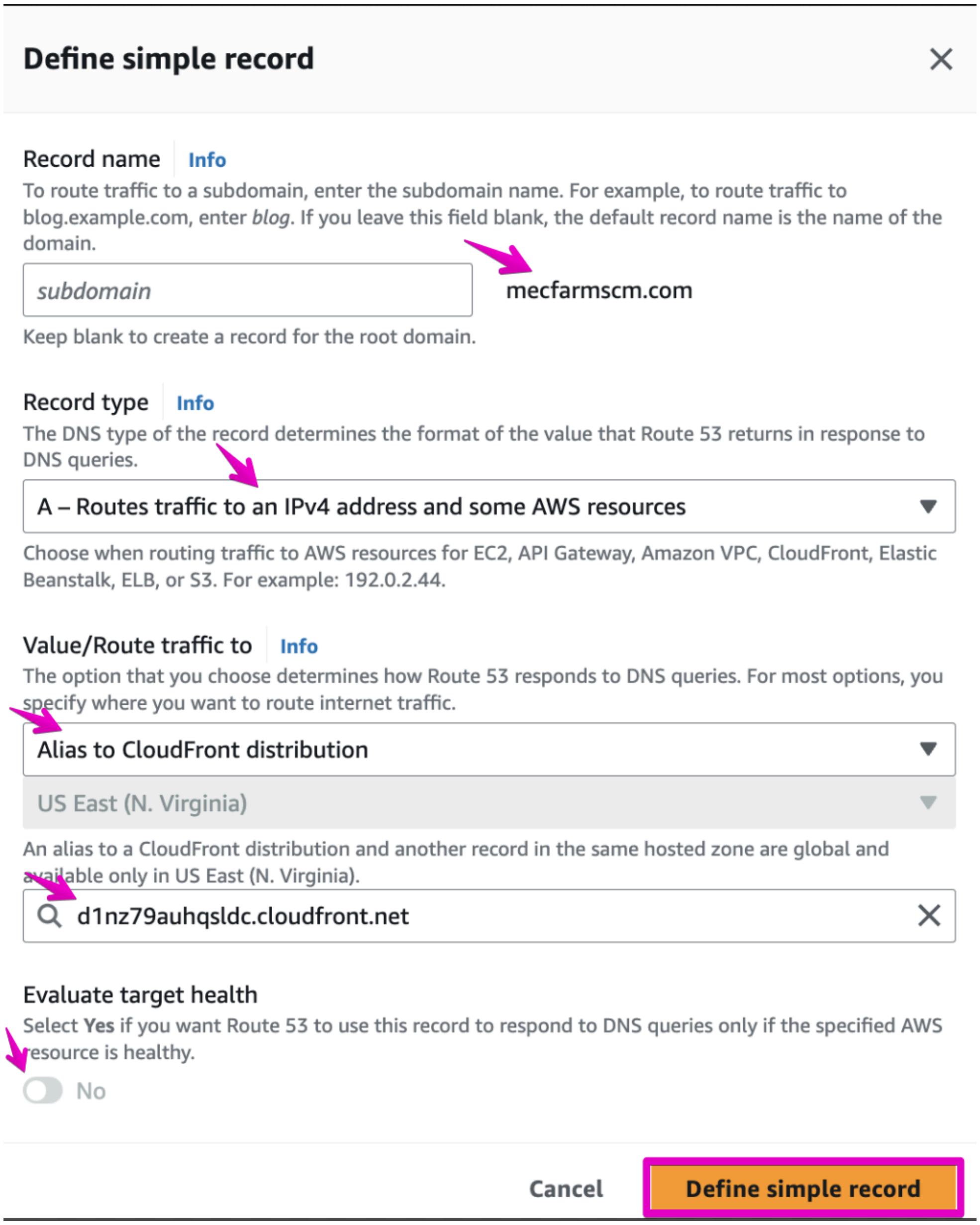
Simple routing records to add to mecfarmscm.com Info

Use if you want all of your clients to receive the same response(s).

☐	Record name	Type	Value/Route traffic to	TTL (seconds)

Define simple records to this list, then choose **Create records**.

2. **Configure Simple Record:**
   - In Record type, choose A - Routes traffic to an IPv4 address and some AWS resources.
   - In Value/Route traffic to, choose Alias to CloudFront distribution and select your distribution.
   - For “Evaluate target health”, choose No.
   - Click "Define simple record" and “Create records”..



Define simple record

Record name Info

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

subdomain mecfarmscm.com

Keep blank to create a record for the root domain.

Record type Info

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources ▼

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to Info

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

Alias to CloudFront distribution ▼

US East (N. Virginia) ▼

An alias to a CloudFront distribution and another record in the same hosted zone are global and available only in US East (N. Virginia).

d1nz79auhqsldc.cloudfront.net

Evaluate target health

Select **Yes** if you want Route 53 to use this record to respond to DNS queries only if the specified AWS resource is healthy.

○ No

o

NB:

- **Check the hosted zone and new records added to the hosted zone**.
- **you cannot create different records with the same name (exception: Failover Records).**

Weighted Routing

1. **Create Weighted Routing Policy:**
   - Go to your Route 53 hosted zone.
   - Click "Create record".
   - Choose "Switch to wizard".
   - Choose "Weighted routing" and click "Next".
   - Choose "Define weighted record".
2. **Configure Weighted Record:**
   - Enter your domain name.
   - Choose A - Routes traffic to an IPv4 address and some AWS resources.
   - In Value/Route traffic to, choose Alias to CloudFront distribution and select your distribution.
   - Set the weight to 50 and click "Define weighted record".
3. **Add Another Weighted Record:**
   - Repeat the above steps to add another weighted record with the same settings but different weight (e.g., 50).

Latency Routing

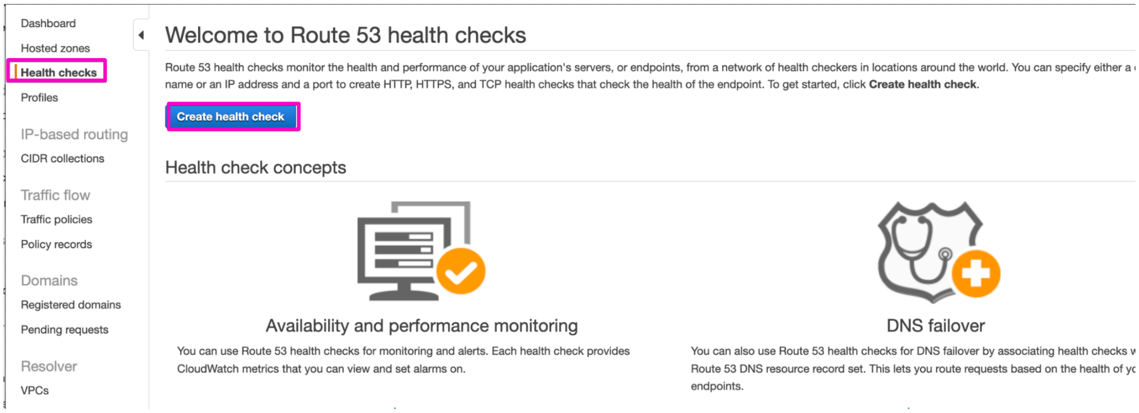
1. **Create Latency Routing Policy:**
   - Go to your Route 53 hosted zone.
   - Click "Create record".
   - Choose "Switch to wizard".
   - Choose "Latency routing" and click "Next".
   - Choose "Define latency record".
2. **Configure Latency Record:**
   - Enter your domain name.
   - Choose A - Routes traffic to an IPv4 address and some AWS resources.
   - In Value/Route traffic to, choose Alias to CloudFront distribution and select your distribution.
   - Choose the region for the latency record.
   - Click "Define latency record".
3. **Add Another Latency Record:**
   - Repeat the above steps to add another latency record for a different region.

Failover Routing

1. **Create R53 Health Check for an Instance** (also possible with ELBs and other AWS resources)

   - go to the R53 console and select Health Checks

- click on Create health check



Welcome to Route 53 health checks

Route 53 health checks monitor the health and performance of your application's servers, or endpoints, from a network of health checkers in locations around the world. You can specify either a name or an IP address and a port to create HTTP, HTTPS, and TCP health checks that check the health of the endpoint. To get started, click **Create health check**.

Health check concepts

Availability and performance monitoring

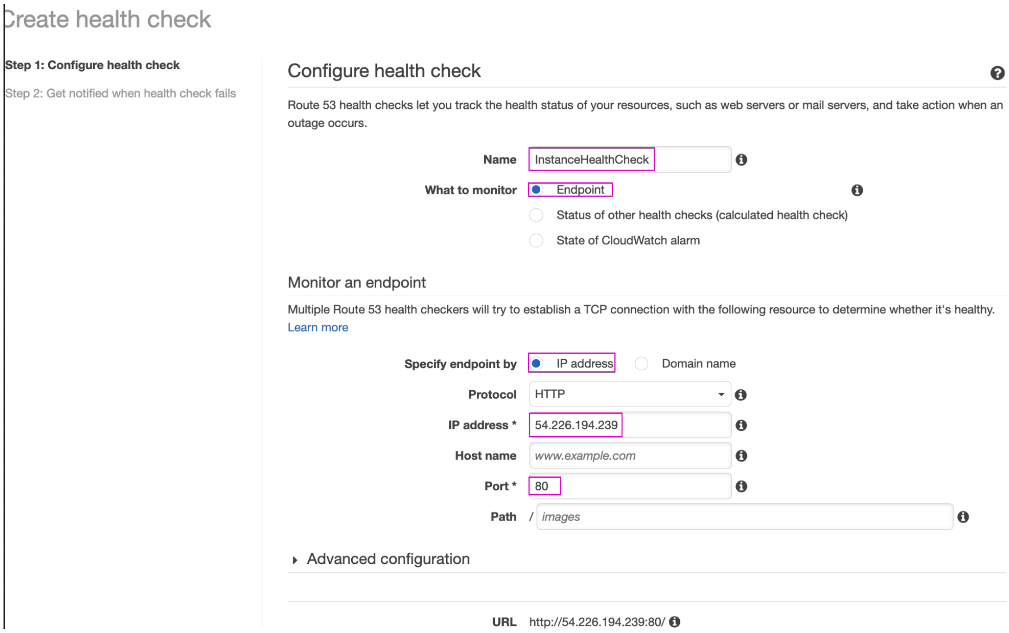
You can use Route 53 health checks for monitoring and alerts. Each health check provides CloudWatch metrics that you can view and set alarms on.

DNS failover

You can also use Route 53 health checks for DNS failover by associating health checks w Route 53 DNS resource record set. This lets you route requests based on the health of yo endpoints.

- Configure Health Check

  1. Name health check e.g**. s1InstanceHealthCheck**
  2. what to monitor select **Endpoint**
  3. specify endpoint by **IP**, and enter the ***public IP of one of the EC2*** instances, **port 80**,
  4. click on Next, Review and **Create health check**


Step 1: Configure health check
Step 2: Get notified when health check fails
Advanced configuration

Create health check

Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name InstanceHealthCheck

What to monitor ● Endpoint
○ Status of other health checks (calculated health check)
○ State of CloudWatch alarm

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy. Learn more

Specify endpoint by ● IP address ○ Domain name

Protocol HTTP

IP address * 54.226.194.239

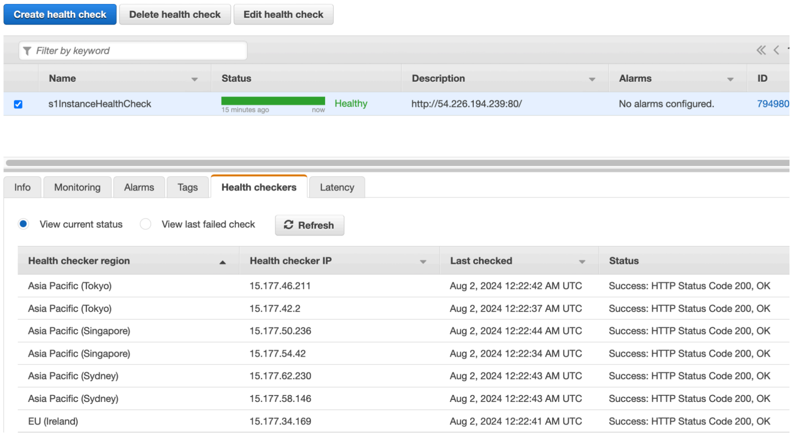
Host name www.example.com

Port * 80

Path / images

URL http://54.226.194.239:80/

HealthChecks

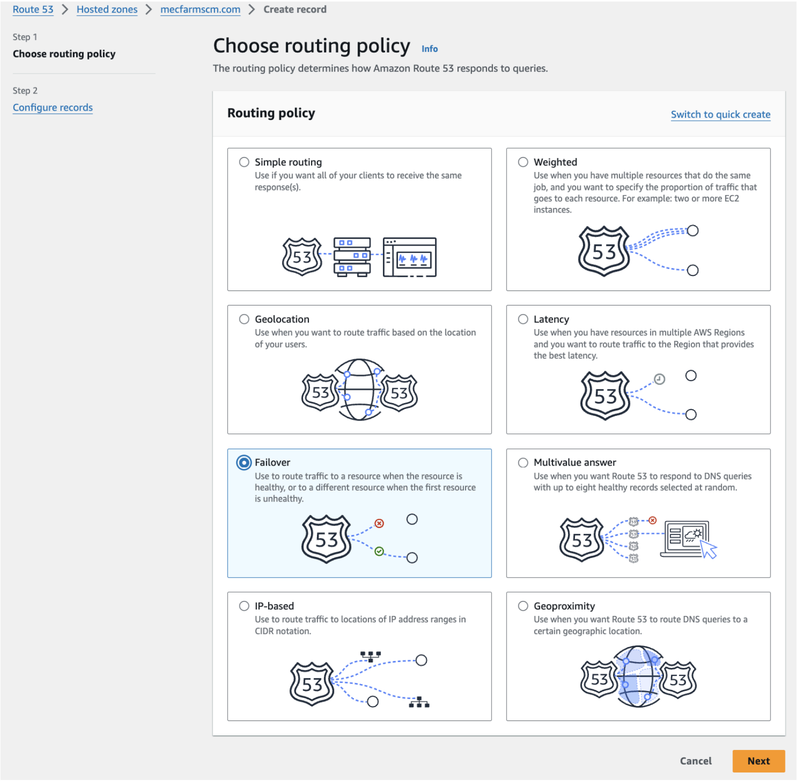


	Name	Status	Description	Alarms	ID
☑	s1InstanceHealthCheck	Healthy (15 minutes ago – now)	http://54.226.194.239:80/	No alarms configured.	794980

● View current status ○ View last failed check

Health checker region	Health checker IP	Last checked	Status
Asia Pacific (Tokyo)	15.177.46.211	Aug 2, 2024 12:22:42 AM UTC	Success: HTTP Status Code 200, OK
Asia Pacific (Tokyo)	15.177.42.2	Aug 2, 2024 12:22:37 AM UTC	Success: HTTP Status Code 200, OK
Asia Pacific (Singapore)	15.177.50.236	Aug 2, 2024 12:22:44 AM UTC	Success: HTTP Status Code 200, OK
Asia Pacific (Singapore)	15.177.54.42	Aug 2, 2024 12:22:34 AM UTC	Success: HTTP Status Code 200, OK
Asia Pacific (Sydney)	15.177.62.230	Aug 2, 2024 12:22:43 AM UTC	Success: HTTP Status Code 200, OK
Asia Pacific (Sydney)	15.177.58.146	Aug 2, 2024 12:22:43 AM UTC	Success: HTTP Status Code 200, OK
EU (Ireland)	15.177.34.169	Aug 2, 2024 12:22:41 AM UTC	Success: HTTP Status Code 200, OK

2. **Create Failover Routing Policy:**
   - Go to your Route 53 hosted zone.
   - Click "Create record".
   - Choose "Switch to wizard".
   - Choose "Failover routing" and click "Next".



Choose routing policy

The routing policy determines how Amazon Route 53 responds to queries.

Routing policy

○ Simple routing
Use if you want all of your clients to receive the same response(s).

○ Weighted
Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

○ Geolocation
Use when you want to route traffic based on the location of your users.

○ Latency
Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

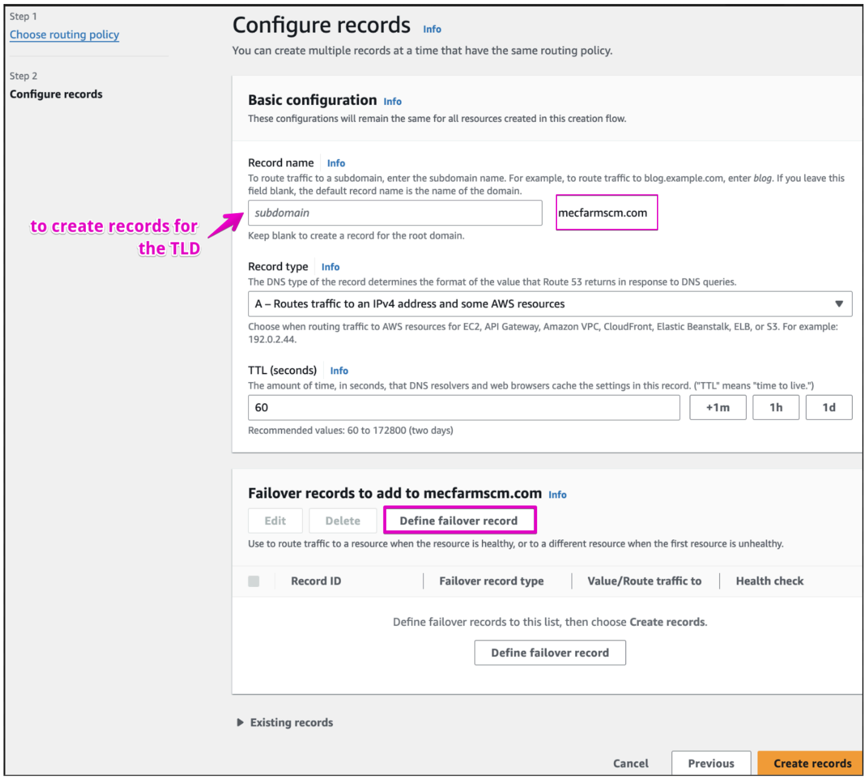
● Failover
Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

○ Multivalue answer
Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

○ IP-based
Use to route traffic to locations of IP address ranges in CIDR notation.

○ Geoproximity
Use when you want Route 53 to route DNS queries to a certain geographic location.

   - Choose "Define failover record".



Step 1
Choose routing policy

Step 2
Configure records

to create records for the TLD

Configure records Info

You can create multiple records at a time that have the same routing policy.

Basic configuration Info

These configurations will remain the same for all resources created in this creation flow.

Record name Info

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

subdomain mecfarmscm.com

Keep blank to create a record for the root domain.

Record type Info

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources ▼

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

TTL (seconds) Info

The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. ("TTL" means "time to live.")

60 | +1m | 1h | 1d

Recommended values: 60 to 172800 (two days)

Failover records to add to mecfarmscm.com Info

Edit | Delete | Define failover record

Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

□	Record ID	Failover record type	Value/Route traffic to	Health check

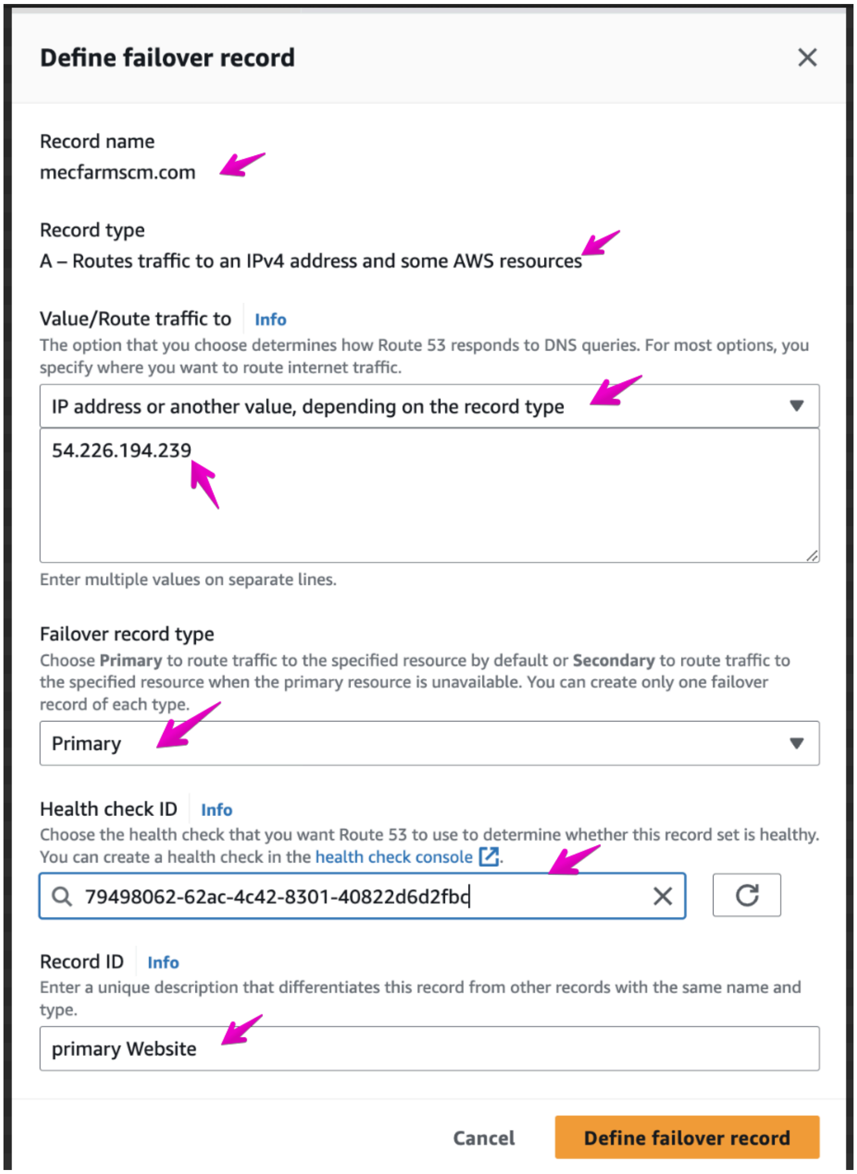
Define failover records to this list, then choose **Create records**.

Define failover record

▶ Existing records

Cancel | Previous | Create records

3. **Configure Failover Record: Primary record**
   - Enter your domain name.
   - Choose A - Routes traffic to an IPv4 address and some AWS resources.
   - In Value/Route traffic to (select the type of endpoint to send traffic) choose *"IP address or another value, depending on the record type"*
   - Choose "Primary" for the failover record type.
   - Add the health check ID
   - Enter a unique description for this record and
   - Click "Define failover record".

- 

Define failover record

Record name
mecfarmscm.com

Record type
A – Routes traffic to an IPv4 address and some AWS resources

Value/Route traffic to | Info
The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value, depending on the record type

54.226.194.239

Enter multiple values on separate lines.

Failover record type
Choose **Primary** to route traffic to the specified resource by default or **Secondary** to route traffic to the specified resource when the primary resource is unavailable. You can create only one failover record of each type.

Primary

Health check ID | Info
Choose the health check that you want Route 53 to use to determine whether this record set is healthy. You can create a health check in the health check console.

79498062-62ac-4c42-8301-40822d6d2fbc

Record ID | Info
Enter a unique description that differentiates this record from other records with the same name and type.

primary Website

4. **Add Another Failover Record:** **secondary record**

Repeat the above steps to add another failover record with the same settings but choose "Secondary".

- Enter your domain name.
- Choose A - Routes traffic to an IPv4 address and some AWS resources.
- In Value/Route traffic to (select the type of endpoint to send traffic) choose *"IP address or another value, depending on the record type"* and add the IP address
- Choose **"secondary"** for the failover record type.
- Optional: Add the health check ID
- Enter a unique description for this record and

- Click "Define failover record" and create record.

  Simulate Failover by stopping the Instance in the Primary record and observe your change change the backend server it routes traffic to.

Step 7: Test Your Website

Verify Website Functionality

1. **Open a Web Browser:**
   - Open a web browser and browse to the following URLs:
     - https://your-domain-name (e.g., https://jjtech.com).