# Host a website

# With a disaster recovery environment

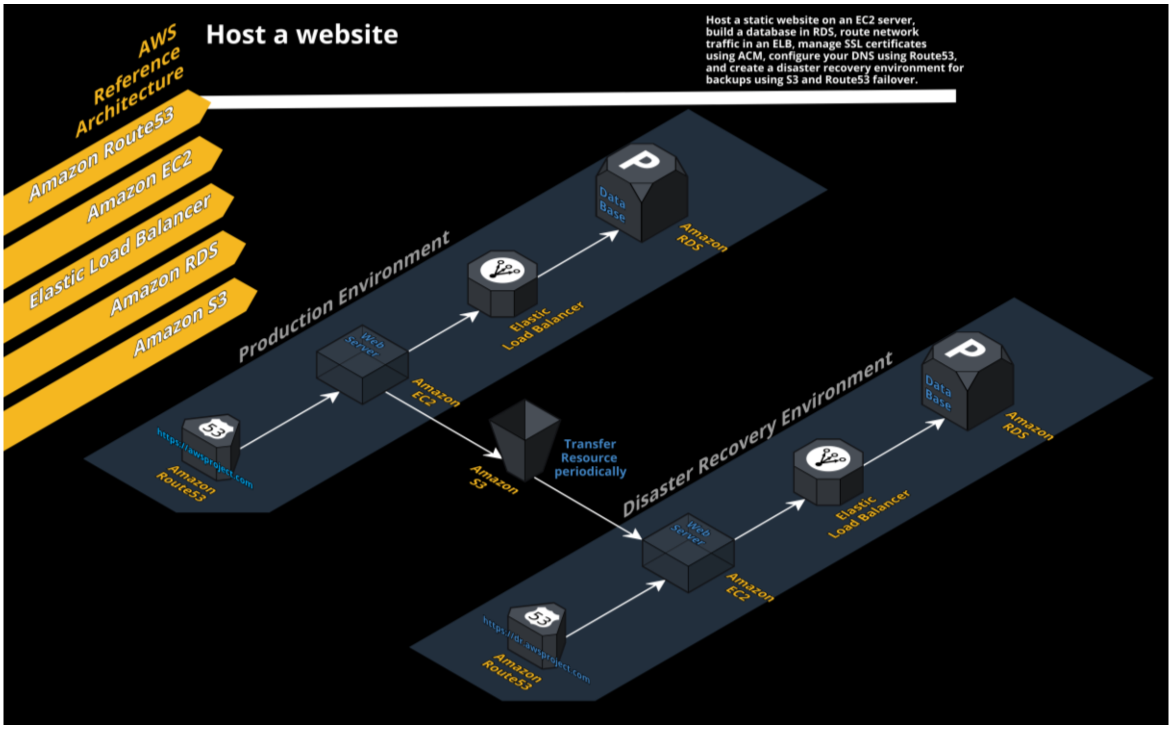
## Project Overview:

Host a static website on an EC2 server, build a database in RDS, route network traffic in an ELB, manage SSL certificates using ACM, configure your DNS using Route53, and create a disaster recovery environment for backups using S3 and Route53 failover.

## Tools & Services we use:

- ACM - Public SSL certificates
- Route53 - DNS routing
- EC2 - Virtual server in the cloud
- ELB - Monitors the health and routes traffic only to the healthy targets
- S3 - Transfer Data Production server to DR server
- RDS - Database
- WordPress - Website Builder

## The Flow:

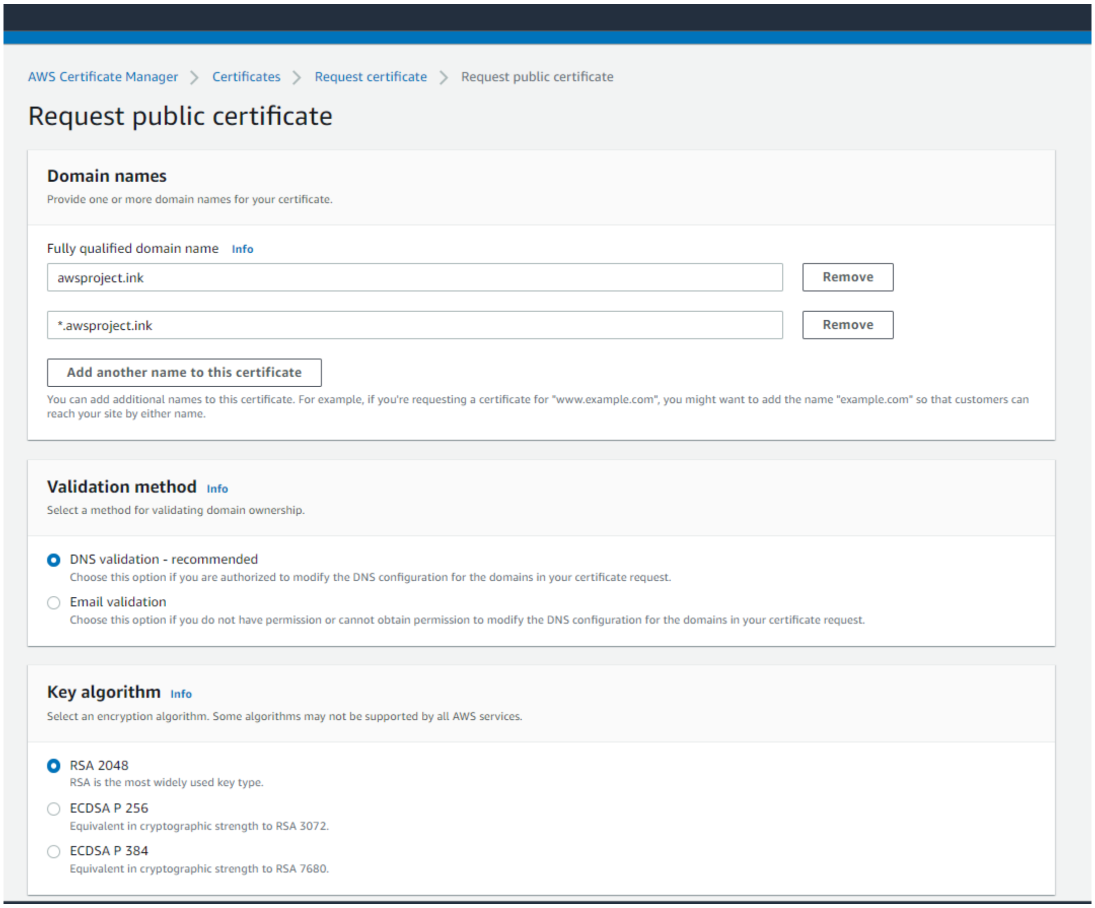


Host a static website on an EC2 server, build a database in RDS, route network traffic in an ELB, manage SSL certificates using ACM, configure your DNS using Route53, and create a disaster recovery environment for backups using S3 and Route53 failover.

# Pre-Requirements: Register the Domain at your DNS provider

## Request an SSL Certificate at ACM:

AWS Certificate Manager → Request a certificate → Certificate type (Public/private) → Enter the Domain name (if use sub-domain use <*.domainName>)



AWS Certificate Manager > Certificates > Request certificate > Request public certificate

Request public certificate

**Domain names**

Provide one or more domain names for your certificate.

Fully qualified domain name Info

awsproject.ink [Remove]

*.awsproject.ink [Remove]

[Add another name to this certificate]

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.

**Key algorithm** Info

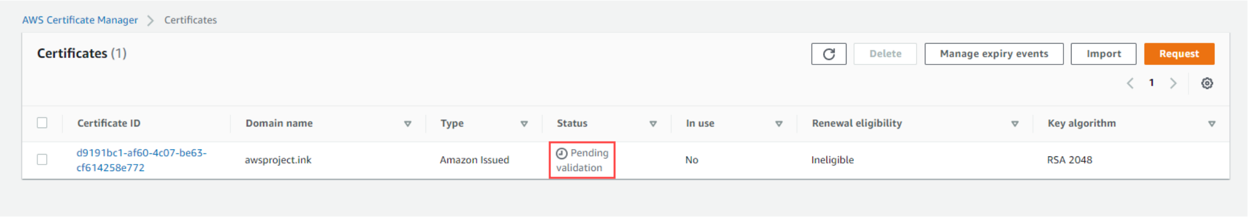
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

● RSA 2048
RSA is the most widely used key type.

○ ECDSA P 256
Equivalent in cryptographic strength to RSA 3072.

○ ECDSA P 384
Equivalent in cryptographic strength to RSA 7680.

Image 1: Request Certificate



AWS Certificate Manager > Certificates

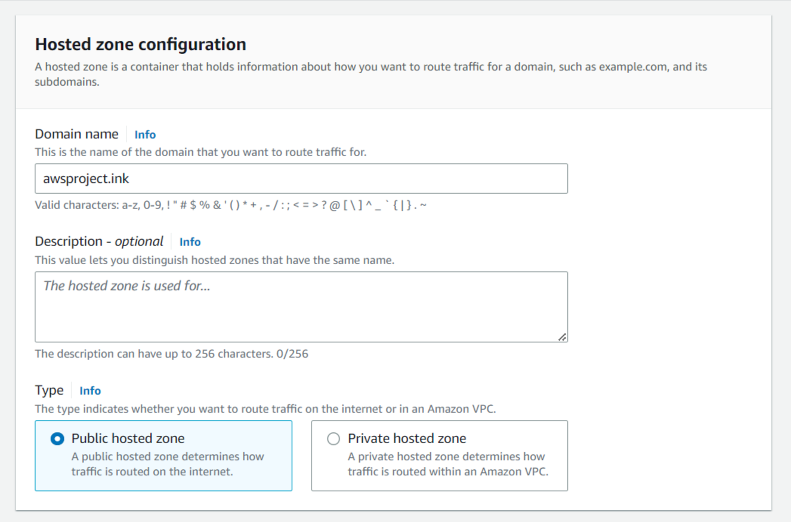
**Certificates** (1)

| □ | Certificate ID | Domain name | Type | Status | In use | Renewal eligibility | Key algorithm |
|---|---|---|---|---|---|---|---|
| □ | d9191bc1-af60-4c07-be63-cf614258e772 | awsproject.ink | Amazon Issued | Pending validation | No | Ineligible | RSA 2048 |

Image 2: ACM Certificate and Validation Pending

# Configure DNS:

Route 53 → Create hosted zone → Enter the Domain name → Create hosted zone → copy name servers and paste your DNS provider (without (.))



**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.

awsproject.ink

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [ \ ] ^ _ ` { | } . ~

Description - *optional* | Info

This value lets you distinguish hosted zones that have the same name.

*The hosted zone is used for...*

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

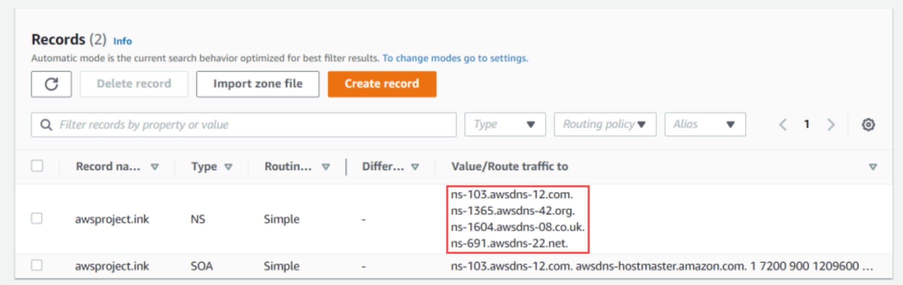
Type | Info

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.

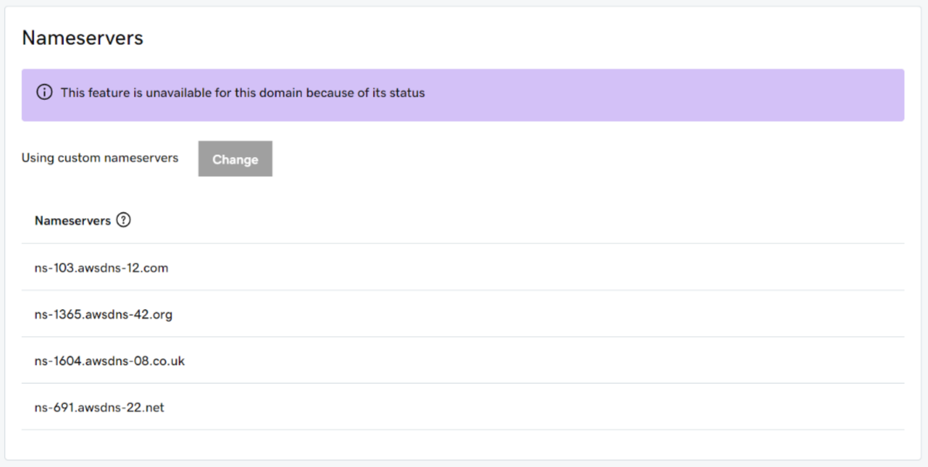
Image 3: Hosted zone configuration



Records (2) Info
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.
Delete record
Import zone file
Create record
Filter records by property or value
Type
Routing policy
Alias
1

| □ | Record na... | Type | Routin... | Differ... | Value/Route traffic to |
|---|---|---|---|---|---|
| □ | awsproject.ink | NS | Simple | - | ns-103.awsdns-12.com.<br>ns-1365.awsdns-42.org.<br>ns-1604.awsdns-08.co.uk.<br>ns-691.awsdns-22.net. |
| □ | awsproject.ink | SOA | Simple | - | ns-103.awsdns-12.com. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 ... |

Image 4: Name Server in Console



Nameservers

ⓘ This feature is unavailable for this domain because of its status

Using custom nameservers Change

Nameservers ⓘ

ns-103.awsdns-12.com

ns-1365.awsdns-42.org

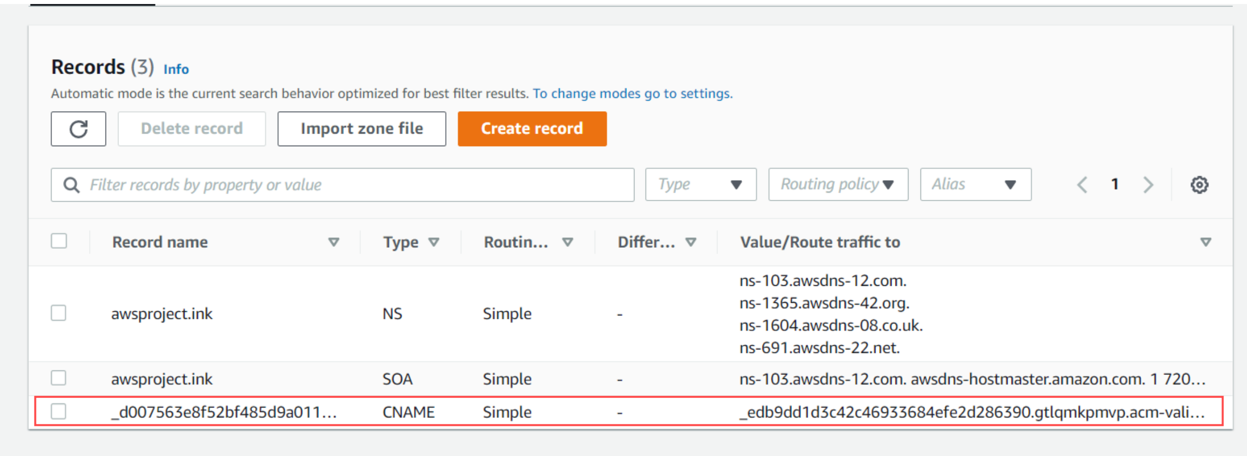
ns-1604.awsdns-08.co.uk

ns-691.awsdns-22.net

Image 5: Adding Name Server in DNS Provider

## Certification Validation:

Goto ACM → select your certificate → Create records in Route 53 → Create records



Records (3) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

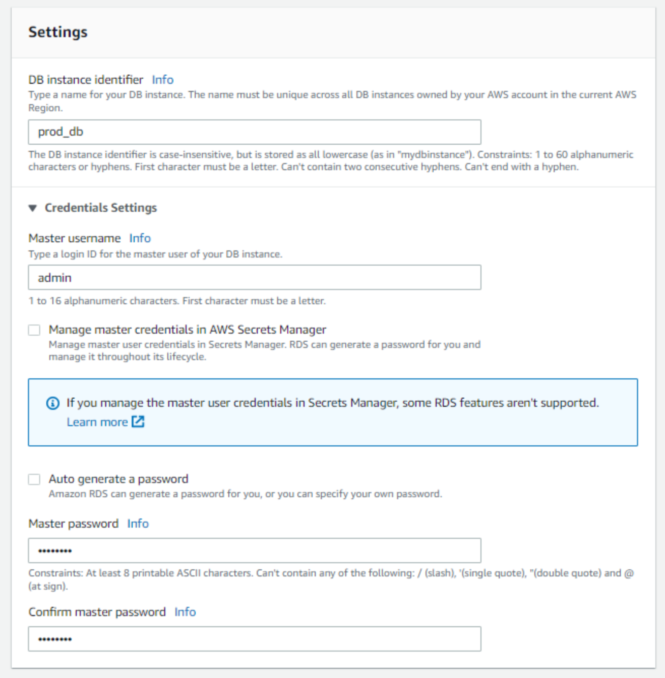
| | Record name | Type | Routin… | Differ… | Value/Route traffic to |
|---|---|---|---|---|---|
| □ | awsproject.ink | NS | Simple | - | ns-103.awsdns-12.com.<br>ns-1365.awsdns-42.org.<br>ns-1604.awsdns-08.co.uk.<br>ns-691.awsdns-22.net. |
| □ | awsproject.ink | SOA | Simple | - | ns-103.awsdns-12.com. awsdns-hostmaster.amazon.com. 1 720… |
| □ | _d007563e8f52bf485d9a011… | CNAME | Simple | - | _edb9dd1d3c42c46933684efe2d286390.gtlqmkpmvp.acm-vali… |

Image 6: After validation new record is created Type: CNAME

## Database Creation in RDS

RDS → Create database → Select (DB: MySQL/ Ver: MySQL 5.7.*/ Free tier) → give (DB_name/ username/ password) → Additional configuration -→ Initial database name (DB_name) →Create database

Create two databases using these steps



**Settings**

DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

prod_db

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter.

□ Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
Learn more

□ Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

•••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

•••••••••

Image7: Adding Credentials

## IAM Role:

Identity and Access Management (IAM) → Role → Create role → AWS service → EC2 → Select preferred policies → give roll name → create

## Server Creation and WordPress installation in EC2

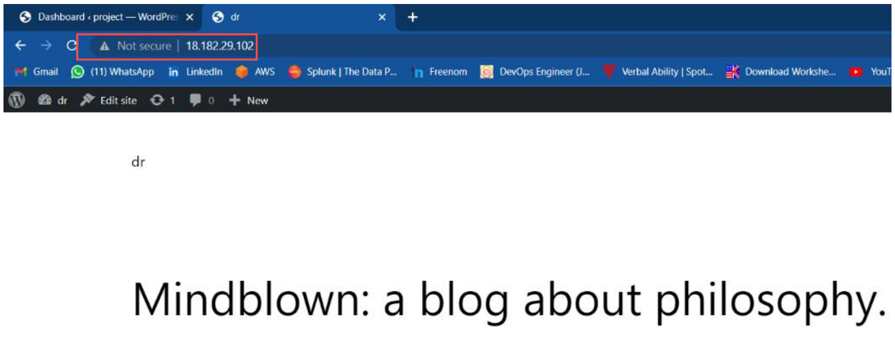
EC2 → Launch instances → Number of instances (2) → Select key pair → IAM instance profile (Select IAM Role) → User data (fill Bash script) → Launch instance

User Data (bash script)

```
#! /bin/bash
yum install httpd php-mysql -y
amazon-linux-extras install -y php7.3
cd /var/www/html
echo "healthy" > healthy.html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
rm -rf wordpress
rm -rf latest.tar.gz
chmod -R 755 wp-content
chown -R apache:apache wp-content
wget https://s3.amazonaws.com/bucketforwordpresslab-
donotdelete/htaccess.txt
mv htaccess.txt .htaccess
chkconfig httpd on
service httpd start
```

## Config WordPress with Database:

- Open WordPress by hitting Public-IP in the browser
- Now it Requires Database credentials
  (DB_Name/ Username/ password/ Database Host: <DatabaseEndpoint:3306>)
- Then it generates a PHP script
- Copy and paste a script in /var/www/html/wp-config.php
- Now fill in some details wordpress requires
- And log-In using username and password



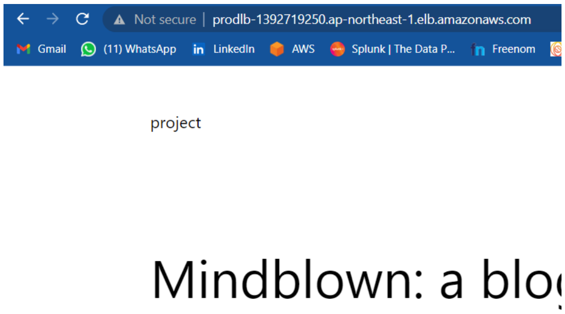
dr

Mindblown: a blog about philosophy.

Image 8: WordPress index page using public-IP

## Elastic Load Balancer (ELB):

Create load balancer → Classic Load Balancer → Create → give name → Next: Assign Security Groups → Select Security Groups → Next: Configure Security Settings → Next: Configure Health Check → Configure Health Check details (Ping Path: /healthy.html)→ Next: Add EC2 Instances → Select Instances → Review and Create → Create



Not secure | prodlb-1392719250.ap-northeast-1.elb.amazonaws.com

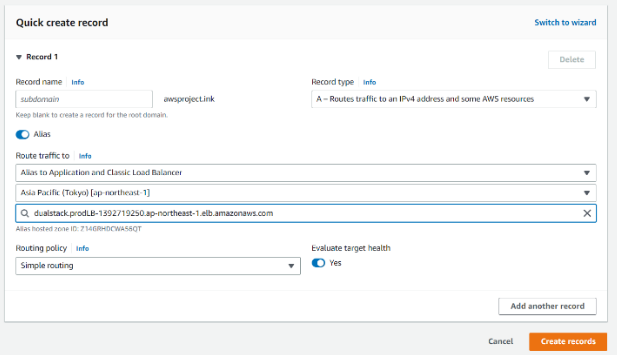
project

Mindblown: a blog

Image 9: WordPress index page using ELB DNS Name

## Attach DNS with ELB:

Goto Route53 -→ Create record → Record name (if have sub domain mention it) → enable Alias → Route traffic to (Alias to Application and Classic Load Balancer/ region/ LB DNS name) → Routing policy (which is needed) → Create records



Quick create record

Record 1

Record name: subdomain awsproject.ink

Keep blank to create a record for the root domain.

Alias

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

Route traffic to: Alias to Application and Classic Load Balancer

Asia Pacific (Tokyo) [ap-northeast-1]

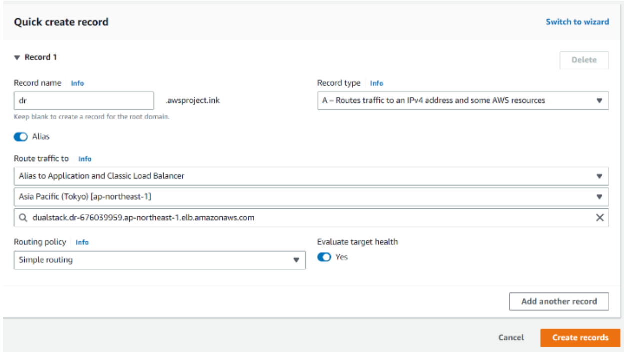
dualstack.prodLB-1392719250.ap-northeast-1.elb.amazonaws.com

Alias hosted zone ID: Z14GRHDCWA56QT

Routing policy: Simple routing

Evaluate target health: Yes

Image 10: Create a record for the production server



Quick create record

Record 1

Record name: dr .awsproject.ink

Keep blank to create a record for the root domain.

Alias

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

Route traffic to: Alias to Application and Classic Load Balancer

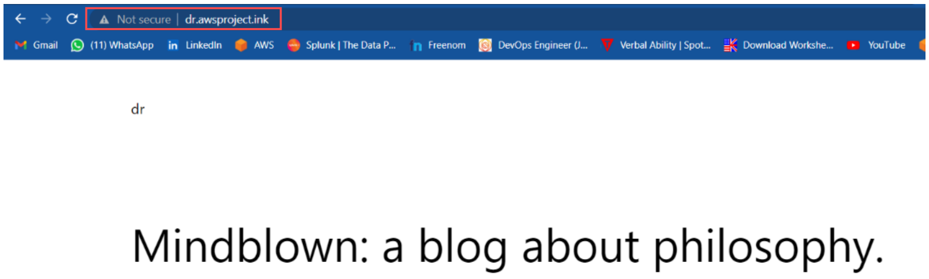
Asia Pacific (Tokyo) [ap-northeast-1]

dualstack.dr-676039959.ap-northeast-1.elb.amazonaws.com

Routing policy: Simple routing

Evaluate target health: Yes

Image 11: Create a record for the Disaster recovery server



Not secure | dr.awsproject.ink

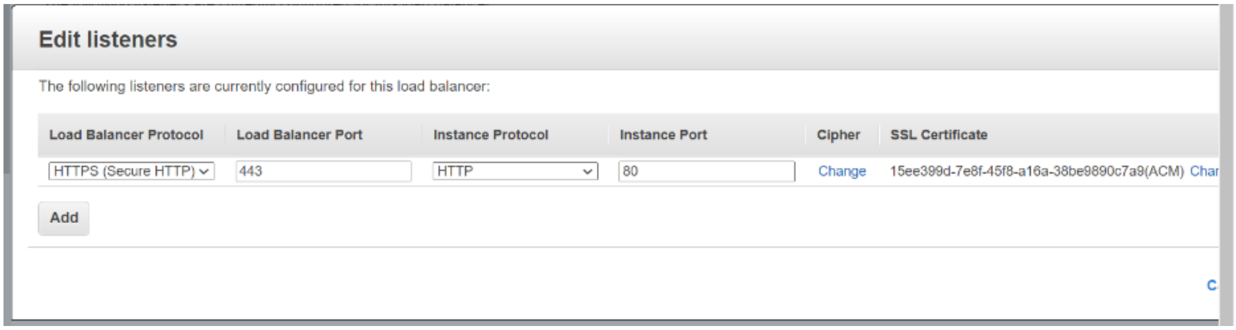
dr

Mindblown: a blog about philosophy.

Image 12: After DNS attach with ELB (http)

## Attach SSL Certificate with DNS:

Goto Load Balancer → Select Load Balancer as we need → Listeners → Edit → Load Balancer Protocol (HTTPS) → SSL Certificate → change → select Choose a certificate from ACM (recommended) → Certificate → save → save

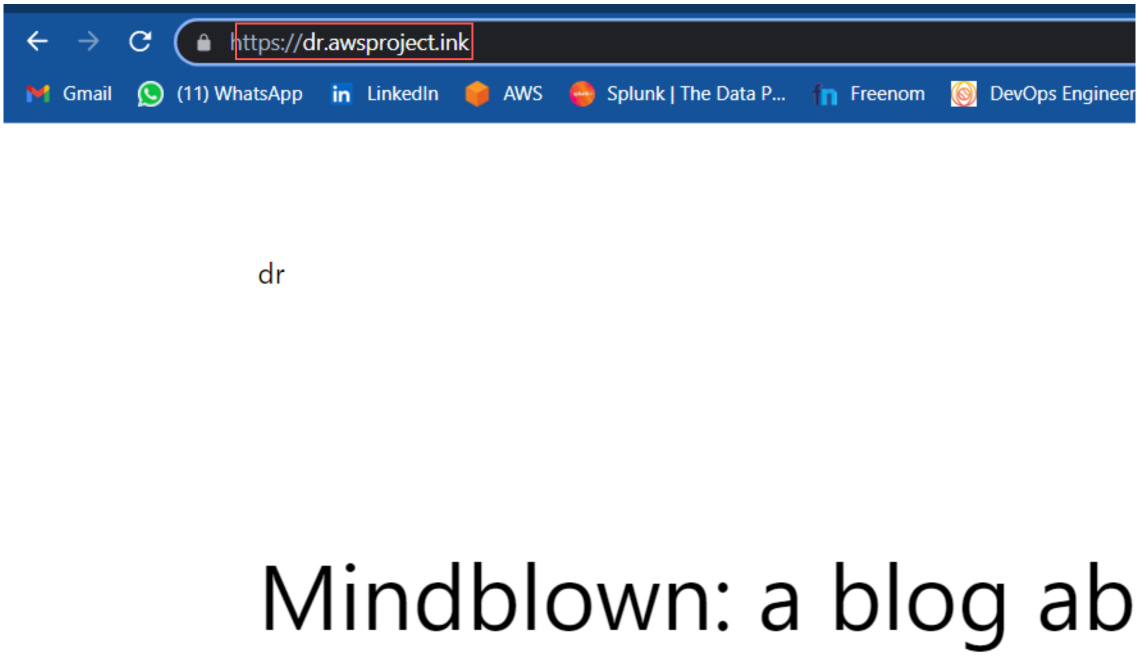


Edit listeners

The following listeners are currently configured for this load balancer:

| Load Balancer Protocol | Load Balancer Port | Instance Protocol | Instance Port | Cipher | SSL Certificate |
|---|---|---|---|---|---|
| HTTPS (Secure HTTP) | 443 | HTTP | 80 | Change | 15ee399d-7e8f-45f8-a16a-38be9890c7a9(ACM) Char |

Image 13: Attach SSL Certificate



https://dr.awsproject.ink

dr

Mindblown: a blog ab

Image 14: After Attach SSL Certificate DNS (https)

## Setup Route53-Failover:

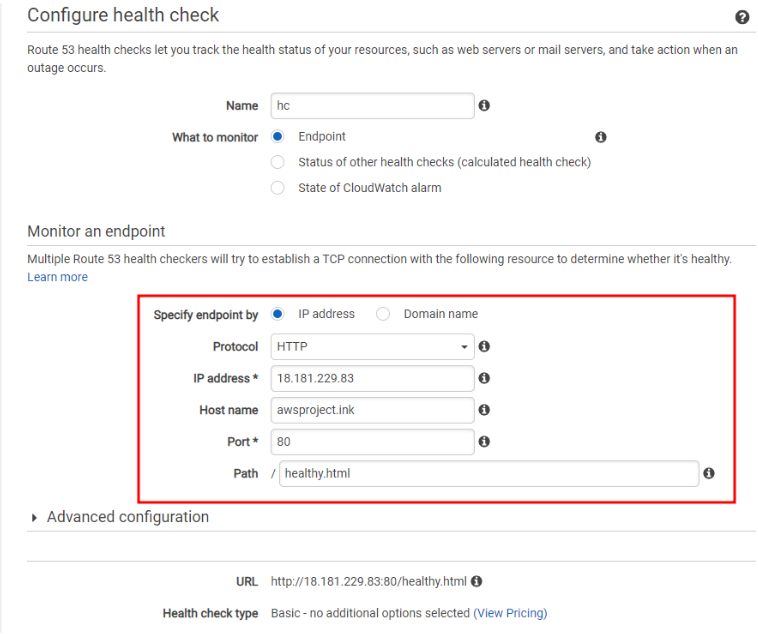
- Create a Health Check
- Create two records under the failover routing policy

## Create Health Check:

Route53 → Health checks → Create health check → Monitor an endpoint

- Specify endpoint by: IP address
- IP address: public Ip or Elastic IP
- Hostname: Domain name
- Path: HealthCheck file name

→ Next → 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: hc

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 *: 18.181.229.83

Host name: awsproject.ink

Port *: 80

Path: / healthy.html

Advanced configuration

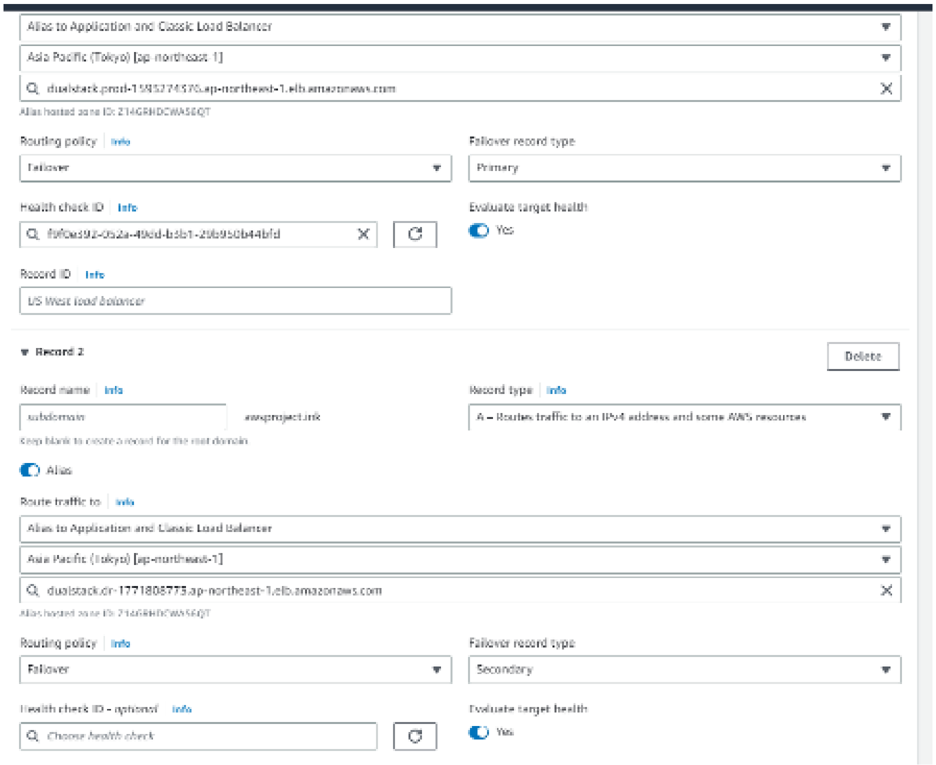
URL: http://18.181.229.83:80/healthy.html

Health check type: Basic - no additional options selected (View Pricing)

Image 15: Health check creation

## Create Failover Record:

Route 53--> Hosted zones --> select Hosted zone --> Create record --> enable Alias --> Route traffic to (Alias to Application and Classic Load Balancer/ region/ prod_LB DNS name) --> Routing policy (failover) --> Failover record type(Primary) --> select Health check ID -->Add another record --> enable Alias --> Route traffic to (Alias to Application and Classic Load Balancer/ region/ dr_LB DNS name) --> Routing policy (failover) --> Failover record type(Secondary) --> Create records



Alias to Application and Classic Load Balancer

Asia Pacific (Tokyo) [ap-northeast-1]

dualstack.prod-1595274570.ap-northeast-1.elb.amazonaws.com

Alias hosted zone ID: Z14GRHDCWA56QT

Routing policy: Failover

Failover record type: Primary

Health check ID: f9f0ad92-052a-49dd-b5b1-20b950b44bfd

Evaluate target health: Yes

Record ID: US West load balancer

Record 2

Record name: subdomain .awsproject.link

Keep blank to create a record for the root domain.

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

Alias

Route traffic to

Alias to Application and Classic Load Balancer

Asia Pacific (Tokyo) [ap-northeast-1]

dualstack.dr-1771808775.ap-northeast-1.elb.amazonaws.com

Alias hosted zone ID: Z14GRHDCWA56QT

Routing policy: Failover

Failover record type: Secondary

Health check ID - optional: Choose health check

Evaluate target health: Yes

Image 16: Failover Record creation

## Create S3 Bucket:

S3 → Create bucket → Bucket name → Select Region → give public access → Create bucket

- Create 2 Buckets
- One For Transfer Codes and Text File
- Another one for Transfer media files

## Transfer Resources Production server to Disaster Recovery Server:

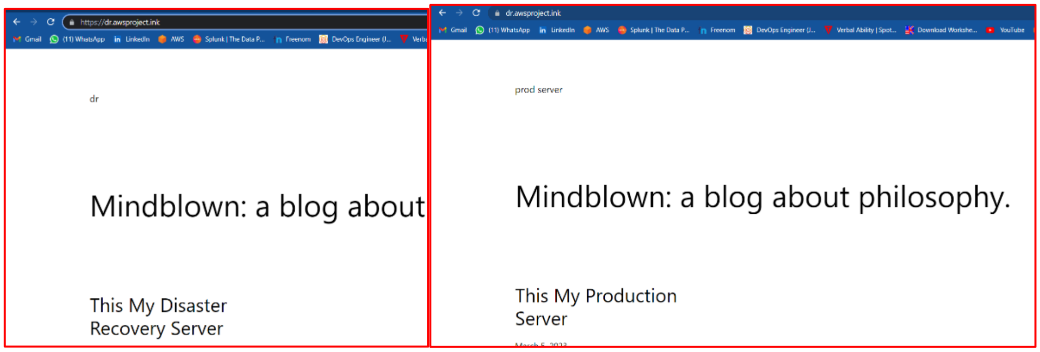
- Open Both servers and Enter the below Commends

```
Open crontab
# crontab -e

For Production Server
*/2 * * * * aws s3 sync --delete /var/www/html/wp-content/uploads s3://<BucketNameForMedia>
*/2 * * * * aws s3 sync --delete /var/www/html/ s3://<BucketNameForCode>

For Disaster Recovery Server
*/2 * * * * aws s3 sync --delete s3://<BucketNameForMedia> /var/www/html/wp-content/uploads
*/2 * * * * aws s3 sync --delete s3://<BucketNameForCode> /var/www/html/
```

- It will take for few minutes



dr

Mindblown: a blog about

This My Disaster Recovery Server

prod server

Mindblown: a blog about philosophy.

This My Production Server

Image 17: Before And After Disaster Recovery Server