

# Load Balanced Architecture with Advanced Request Routing

This lab guide creates a simple website, hosted on EC2, behind an Application Load Balancer. Host-based and path-based routing rules will then be configured to route based on information in the host header or URL path.

Host-based routing allows you to route to multiple domains on a single load balancer by routing to a different set of EC2 instances or containers based on information in the host header.

Path-based routing is also referred to as URL-based routing. The Application load Balancer will forward the requests to the specific targets based on the rules configured on the load balancer. We will be doing this first.

## Requirements (Prerequisites)

- *AWS Free Tier Account*
- *Basic Exposure to the AWS Console and completion of prior days learning*
- *Follow the steps in Section 8 of the course to register your own domain name in Route 53*

## Resources

Download the "[advanced-request-routing-code.zip](#)" file.

## Exercise Overview

**Exercise 1** - Create the Red and Blue EC2 instances

**Exercise 2** – Enable Path-based Routing

**Exercise 3** – Enable Host-based Routing

**Exercise 4** – Clean up your resources

## Exercise 1 - Create the Red and Blue EC2 instances

### Task 1 - Create the S3 Bucket and Upload the code

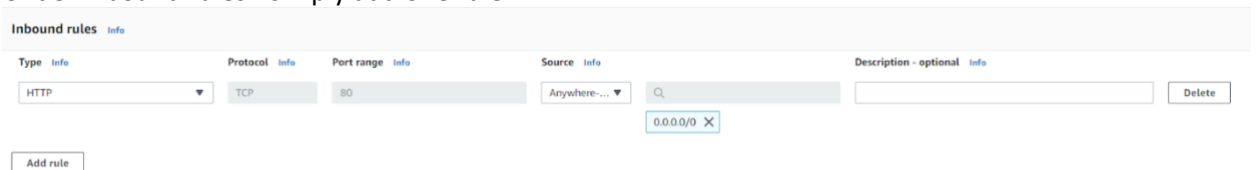
The first step is to upload the code we will use to create the websites into an S3 bucket.

1. Go to the S3 console and click 'Create Bucket'.
2. We will call out bucket 'advanced-request-routing-123456' with the numbers at the end being random to make the bucket globally unique. Take a note of the bucket name as we will be referring to it again shortly.
3. Scroll down and click 'Create Bucket'.
4. Locate the files you have downloaded as part of the course and select all files except the user data files and bucket-permissions file and upload them to the bucket.

### Task 2 - Create your Security Group

The second step is to pre-emptively create the Security Group for your EC2 instances.

1. Go to the EC2 console, scroll down to the 'Security Group' column under 'Network Security'. Click 'Create Security Group'. We will call it 'WebsiteSG'. Populate the description field with 'WebsiteSG' also.
2. Under inbound rules - simply add one rule:



The screenshot shows the 'Inbound rules' section of the AWS EC2 console. It features a table with columns: Type, Protocol, Port range, Source, and Description - optional. The 'Type' column has a dropdown menu set to 'HTTP'. The 'Protocol' column has a dropdown menu set to 'TCP'. The 'Port range' column has a text input set to '80'. The 'Source' column has a dropdown menu set to 'Anywhere...'. Below the 'Source' dropdown, there is a search bar containing '0.0.0.0/0' and a delete icon. At the bottom left, there is an 'Add rule' button. At the bottom right, there is a 'Delete' button.

3. This is an HTTP rule, with an 'Anywhere IPv4' source. Create the Security Group.

### Task 3 - Create the Red EC2 instance.

We will now launch the 'Red' EC2 instance and check to see if it has successfully retrieved the code from S3.

1. Head over to the EC2 console and select 'Launch Instance'.
2. Call your first instance 'Red'.
3. Select the AMI, and under 'Instance type', select 't2.micro'.
4. For 'Key pair', leave the default setting 'Proceed without a key pair'.
5. Under 'Network settings' select the security group you created earlier and choose the subnet us-east-1a.
6. Expand 'Advanced Details' and under 'IAM instance profile' select 'Create a new IAM profile.'
7. A new page will open. Click 'Create Role'.
8. Under common use cases, select 'EC2' and click next.

9. Click 'Create Policy' and move to the JSON section. Copy and paste the code from the bucket-permissions.json file into the code window, replacing 'YOUR-BUCKET-ARN' with the ARN of the bucket you have just created.
10. Click review policy, and Create policy, call it 'mys3to3c2policy'.
11. Go back to where we are creating the role, refresh the policies and find and select the policy we just created. Click next and call it 'mys3to3c2role'.
12. Go back to the launch instance page, and refresh the IAM instance profile, and locate the role, and click attach.
13. Next, copy the user data from the 'user-data-red' file into the 'User data' field. It should look like the image below. You will also need to edit the name 'YOUR-BUCKET-NAME' in the user data to the name of your S3 bucket.

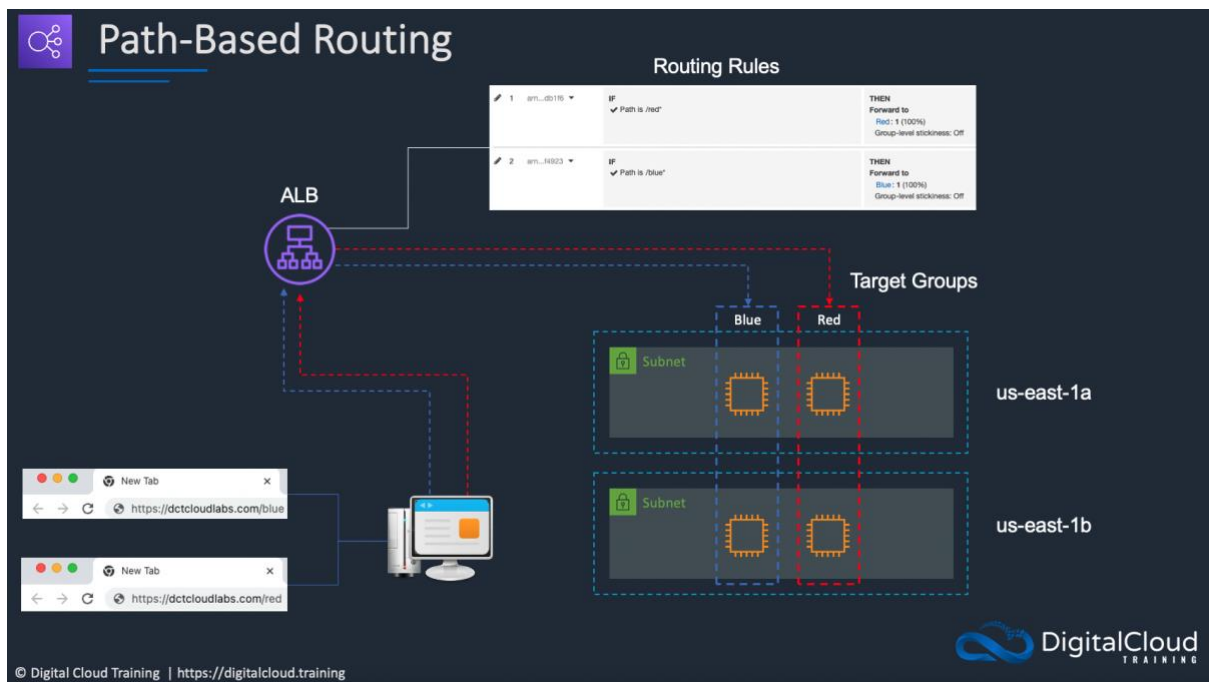
```
#!/bin/bash
yum update -y
yum install -y httpd
# below line starts httpd daemon
systemctl start httpd
# below line sets http daemon to start automatically at boot time
systemctl enable httpd
mkdir /var/www/html/red
# populate /red directory of server with web page
cd /var/www/html/red
aws s3 cp s3://YOUR-BUCKET-NAME/hw-red.css ./
aws s3 cp s3://YOUR-BUCKET-NAME/hw-red-py.css ./
aws s3 cp s3://YOUR-BUCKET-NAME/python.png ./
aws s3 cp s3://YOUR-BUCKET-NAME/apache.svg ./
aws s3 cp s3://YOUR-BUCKET-NAME/red-index.html ./index.html
# populate root of server with web page (replaces apache default page)
cd /var/www/html
aws s3 cp s3://YOUR-BUCKET-NAME/hw-red.css ./
aws s3 cp s3://YOUR-BUCKET-NAME/hw-red-py.css ./
aws s3 cp s3://YOUR-BUCKET-NAME/python.png ./
aws s3 cp s3://YOUR-BUCKET-NAME/apache.svg ./
aws s3 cp s3://YOUR-BUCKET-NAME/red-root-index.html ./index.html
# optional restart of httpd daemon
systemctl restart httpd
```

14. Create the instance.
15. Follow the exact same steps to create the Blue instance with the blue user data (user-data-blue) and select subnet us-east-1b.

You should then be able to get the public IP address from either instance to view the web pages created on each EC2 instance. You will need to add /blue and /red to the appropriate instance to return the custom web page.

## Exercise 2 – Enable Path-based Routing

With path-based routing we will enter a path to our URL and the load balancer will route the request to the appropriate target group based on the rules we create. The architecture looks like this:



## Task 1 – Create your target groups

The first step is to set up the target groups; you need at least 2 target groups to configure Path-based routing.

1. To start things off, click on Target Groups under Load balancing.
2. Click 'Create target group.'
3. Set up 2 target groups, one is called 'Red' which will contain the red targets, and the other is called 'Blue' and will contain the blue targets.
4. Leave all the defaults, except changing the target group name to 'Red' / 'Blue' and change the health check to:
  - a. For Red: /red/index.html
  - b. For Blue: /blue/index.html
5. Register the correct instance i.e., Red and be sure to click 'include as pending below'.

| Target groups (2) <a href="#">Info</a>                      |      |                                |      |          |             |                                 |                       |  |
|---|------|--------------------------------|------|----------|-------------|---------------------------------|-----------------------|--|
| <input type="text" value="Search or filter target groups"/> |      |                                |      |          |             |                                 |                       |  |
| <input type="checkbox"/>                                    | Name | ARN                            | Port | Protocol | Target type | Load balancer                   | VPC ID                |  |
| <input type="checkbox"/>                                    | Blue | arn:aws:elasticloadbalancin... | 80   | HTTP     | Instance    | <a href="#">None associated</a> | vpc-0e71f18342474263f |  |
| <input type="checkbox"/>                                    | Red  | arn:aws:elasticloadbalancin... | 80   | HTTP     | Instance    | <a href="#">None associated</a> | vpc-0e71f18342474263f |  |

## Task 2 – Create your Application Load Balancer

Next step is to create the Application Load Balancer.

1. On the left-hand side of the EC2 console you will find the link for Load Balancers.
2. Click 'Create Load Balancer' and choose the 'Application Load Balancer'.

3. Call the load balancer 'LabLoadBalancer' and leave the Scheme as Internet Facing.
4. Select both us-east-1a and us-east-1b for the subnet mappings. This will allow routing of traffic across the instances in different AZs.
5. Choose the same Security Group as earlier (WebsiteSG).
6. Choose the listener and routing as per the image below; we will configure the routing rules later.

#### 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

Remove

Protocol

HTTP ▼

Port

80

1-65535

Default action

Forward to

Blue

HTTP ▼

⌂

Target type: Instance, IPv4

Create target group [↗](#)

Add listener

7. Click Create load balancer and wait a few minutes for it to turn from 'Provisioning' to 'Active'.
8. Once active, under Listeners, select the first listener, click 'Rules' and click 'Manage rules'.

| Listeners (1)  |                                 |                             |                   |                                    |  |                           | <a href="#">⌂</a>      | Actions ▼ | Add listener |
|--|---------------------------------|-----------------------------|-------------------|------------------------------------|--|---------------------------|------------------------|-----------|--------------|
| A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules. |                                 |                             |                   |                                    |  |                           |                        |           |              |
| <input type="text" value="Search"/>  |                                 |                             |                   |                                    |  |                           | < 1 > ⌂                |           |              |
| <input type="checkbox"/>   | Protocol:Port <a href="#">↗</a> | ARN ▼                       | Security policy ▼ | Default SSL cert <a href="#">↗</a> | Default routing rule <a href="#">↗</a>   | Rules <a href="#">↗</a> ▼ | Tags <a href="#">↗</a> |           |              |
| <input type="checkbox"/>   | HTTP:80                         | <a href="#">arn...db1f6</a> | Not applicable    | Not applicable                     | 1. Forward to<br><ul style="list-style-type: none"> <li>BLUE <a href="#">↗</a>: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul> | 1                         | 0                      |           |              |

9. Edit the paths.

Click the + sign on top and click Insert Rule then select the Rule type as Path, enter /red\* in the select the target group 'Red' in the Forward to column.

Click Save, insert another rule, and configure forwarding to the Blue target group too. The configuration should look like the image below:

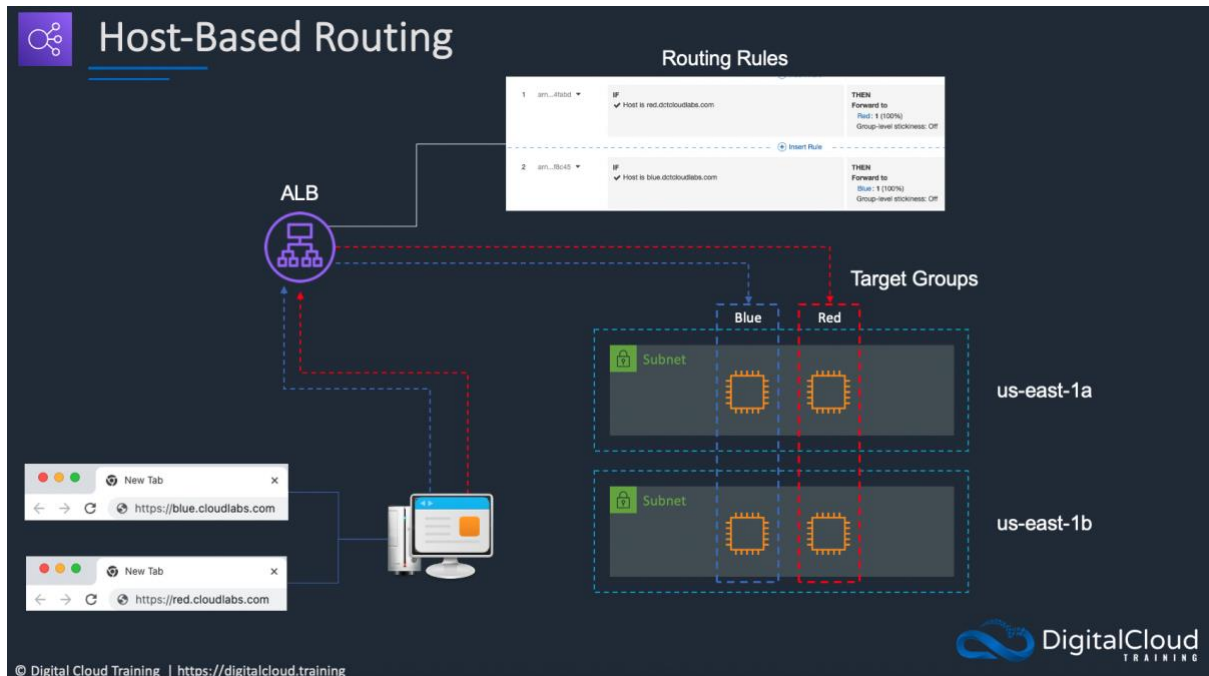
|      |  |  |  |
|------|--|--|--|
| 1    | arn...db1f6 ▼  | <div>IF</div> <div>✓ Path is /red*</div>                 | <div>THEN</div> <div>Forward to</div> <div>Red: 1 (100%)</div> <div>Group-level stickiness: Off</div>  |
| 2    | arn...f4923 ▼  | <div>IF</div> <div>✓ Path is /blue*</div>                | <div>THEN</div> <div>Forward to</div> <div>Blue: 1 (100%)</div> <div>Group-level stickiness: Off</div> |
| last | HTTP 80: default action<br><i>This rule cannot be moved or deleted</i> | <div>IF</div> <div>✓ Requests otherwise not routed</div> | <div>THEN</div> <div>Forward to</div> <div>Blue: 1 (100%)</div> <div>Group-level stickiness: Off</div> |

10. We can then test the load balancer's path-based routing by copying the DNS name from the Application Load Balancer and append either /blue or /red on the URL and see what happens. You should see the different colored custom web pages we added to our instances.

## Exercise 3 – Enable Host-based Routing

With host-based routing we will enter a subdomain to the domain name and the load balancer will route the request to the appropriate target group based on the rules we create.

The architecture looks like this:



## Task 1 – Edit host-based forwarding rules

The first step is to set up the forwarding rules. You need at least 2 target groups to configure host-based routing.

1. Navigate back to the listener.
2. Click on 'Manage rules' under the 'Rules' tab.
3. Remove the path-based routing rules created earlier.
4. Click on the '+' symbol followed by 'Insert Rule'.
5. Under the 'IF(all match)' column, click on the '+ Add condition' drop-down arrow and select 'Host header' as the Rule type and put your domain name with the blue or red subdomains in front. For example, blue.dctcloudlabs.com and red.dctcloudlabs.com
6. In the 'Then' column, click on '+Add action' drop-down arrow and select 'Forward to' as the action. Here select the appropriate target group (Red / Blue).
7. Make sure you repeat the above to have one rule for each target group / subdomain.

The rules should look something like this:

|               |   |  |
|---------------|---|--|
| + Insert Rule |   |  |
| 1             | arn...4fabd ▾   | <div> <b>IF</b><br/> ✓ Host is red.dctcloudlabs.com </div> <div> <b>THEN</b><br/> <b>Forward to</b><br/> Red: 1 (100%)<br/> Group-level stickiness: Off </div>   |
| + Insert Rule |   |  |
| 2             | arn...f8c45 ▾   | <div> <b>IF</b><br/> ✓ Host is blue.dctcloudlabs.com </div> <div> <b>THEN</b><br/> <b>Forward to</b><br/> Blue: 1 (100%)<br/> Group-level stickiness: Off </div> |
| + Insert Rule |   |  |
| last          | <b>HTTP 80: default action</b><br><i>This rule cannot be moved or deleted</i> | <div> <b>IF</b><br/> ✓ Requests otherwise not routed </div> <div> <b>THEN</b><br/> <b>Forward to</b><br/> Red: 1 (100%)<br/> Group-level stickiness: Off </div>  |

## Task 2 – Configure Records in Route 53

In this task we need to create the relevant subdomain DNS records in Amazon Route 53 and configure the load balancer as the target.

1. Open Route 53 dashboard from the management console, find your public domain name under hosted zones. Click on it and select 'Create Record.'
2. Enter various details for this record:
  - subdomain:** blue or red
  - Record type:** Select A type here.
  - Value/Route traffic to:**
    - Select 'Alias to Application and Classic Load Balancer'
    - Select region N.Virginia
    - Select the target load balancer we made earlier; it should all look like this.

## Define simple record

Record name

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.

Keep blank to create a record for the root domain.

Record type

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

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 Application and Classic Load Balancer

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

Alias hosted zone ID: Z35SXDOTRQ7X7K

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.

☒ Yes

Cancel

Define simple record

- Finally click 'Define simple record'.
- Make sure you repeat the above steps to ensure you have one DNS record for each of the blue and red subdomains.
- The DNS records will look something like this:

Records (4)

Info

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

Type

Routing policy

Alias

< 1 >

| <input type="checkbox"/> | Record name           | Type | Routin... | Differ... | Value/Route traffic to  |
|--------------------------|-----------------------|------|-----------|-----------|---|
| <input type="checkbox"/> | dctcloudlabs.com      | NS   | Simple    | -         | ns-74.awsdns-09.com.<br>ns-1291.awsdns-33.org.<br>ns-653.awsdns-17.net.<br>ns-1973.awsdns-54.co.uk. |
| <input type="checkbox"/> | dctcloudlabs.com      | SOA  | Simple    | -         | ns-74.awsdns-09.com. awsdns-hostmaster.amazon.com. 1 7200 900 12096                                 |
| <input type="checkbox"/> | blue.dctcloudlabs.com | A    | Simple    | -         | dualstack.labloadbalancer1-1797931097.us-east-1.elb.amazonaws.com.                                  |
| <input type="checkbox"/> | red.dctcloudlabs.com  | A    | Simple    | -         | dualstack.labloadbalancer1-1797931097.us-east-1.elb.amazonaws.com.                                  |



6. To check if everything is working as expected, open a web browser, and paste the DNS name of the load balancer and append it with either Red or Blue, like this:

[red.mydomainname.com/red](http://red.mydomainname.com/red)

[blue.mydomainname.com/blue](http://blue.mydomainname.com/blue)

You should see the webpages load and be populated with content. Bear in mind that the custom web pages exist under the /blue and /red paths, so you'll need to add those to see that the correct pages are loading.

## Exercise 4 – Clean up your resources

### Task 1 – Delete your resources

You can now delete the resources you created:

- Application Load Balancer (chargeable).
- Target groups (not chargeable).
- Security Group (not chargeable).
- Amazon EC2 Instances (chargeable).
- Amazon S3 bucket (chargeable).
- Amazon Route 53 Alias records (not chargeable).