



Route 53: Failover routing Policy

1. In this lab you are going to learn about failover routing policy.
2. Now go to AWS Console and navigate to EC2 and launch 2 instances of Ubuntu OS.
3. Select different subnet for both of them and same security group with HTTP assigned to both of them.
4. Once they are launched go to Route 53 and delete the last record which you created.

Records (3) | DNSSEC signing | Hosted zone tags (0)

Records (1/3) **Info**
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

C Delete record Import zone file Create record

Filter records by property or value Type Routing pol... Alias Value/Route traffic to TTL (s...) Health

Record name	Type	Routing pol...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health
cloudservicesdemo.in	A	Simple	-	No	18.130.4.147	300	-

5. After that you need to log in to your both of the instances using Putty tool or command prompt.

```
System information as of Mon Feb  5 17:12:31 UTC 2024

System load:  0.0          Processes:            95
Usage of /:   20.5% of 7.57GB    Users logged in:      0
Memory usage: 20%          IPv4 address for eth0: 172.31.25.100
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-25-100:~$ |
```

6. Now run update command and update your instance. Then install the nginx web server.

```
sudo apt update  
sudo apt install nginx
```

7. Then you need to create a index.html file here.

```
cd /var/www/html  
sudo chmod 777 /var/www/html  
echo "<h1> This is the instance with name demo-route-01</h1>" > Index.html
```

```
ubuntu@ip-172-31-25-100:~$ cd /var/www/html  
ubuntu@ip-172-31-25-100:/var/www/html$ sudo chmod 777 /var/www/html  
ubuntu@ip-172-31-25-100:/var/www/html$ echo "<h1> This is the instance with name demo-route-01</h1>" > Index.html  
ubuntu@ip-172-31-25-100:/var/www/html$ |
```

8. Just to confirm copy the public IP address of your instance and paste it in a new tab.



This is the instance with name demo-route-01

9. Now you need to do the same thing on the other instance.



10. So, let's simulate the scenario in which our web application is running on this primary web server, and all of the requests for clouddemo.in should go onto this server. If this server goes down for any reason, then start routing requests onto a secondary web server.
11. Now in Route 53 from the hamburger icon go to health check. In health check click on create health check.

Dashboard

Hosted zones

Health checks

The screenshot shows the AWS Route 53 Health Checks list interface. At the top left, there is a blue 'Create health check' button with a red border. To its right are 'Delete health check' and 'Edit health check' buttons. The main area has a table header with columns: Name, Status, Description, Alarms, and ID. Below the header, it says 'No results'. There are also navigation links like '<< < No health checks to view > >>' and a search bar labeled 'Filter by keyword'.

12. Now here first you need to give a name to your health check, then you need to give the public IP address of your instance.
13. Then in the advanced configuration keep them to default. Then just create your health check.
14. Once your health check is created you will see that its status is unknown. It will take some time for its status to get healthy.

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	<input type="text" value="demo-route-01-health-check"/>	
What to monitor	<input checked="" type="radio"/> Endpoint <input type="radio"/> Status of other health checks (calculated health check) <input type="radio"/> 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	<input checked="" type="radio"/> IP address	<input type="radio"/> Domain name		
Protocol	<input style="border: 1px solid #ccc; padding: 2px 5px; width: 150px; height: 20px; border-radius: 5px; font-size: 10px; margin-right: 10px;" type="button" value="HTTP"/>			
IP address *	<input type="text" value="3.8.122.201"/>			
Host name	<input type="text" value="www.example.com"/>			
Port *	<input type="text" value="80"/>			
Path	<input type="text" value="/images"/>			

Advanced configuration

Request interval	<input checked="" type="radio"/> Standard (30 seconds)	<input type="radio"/> Fast (10 seconds)		
Failure threshold *	<input type="text" value="3"/>			
String matching	<input checked="" type="radio"/> No	<input type="radio"/> Yes		
Latency graphs	<input type="checkbox"/>			
Invert health check status	<input type="checkbox"/>			
Disable health check	<input type="checkbox"/> By default, disabled health checks are considered healthy. Learn more			

The screenshot shows the AWS Health Checks console. At the top, a green notification bar says "Health check with id 091b6812-ed30-4202-b396-19f172c91a5f has been created successfully". Below it, a blue info icon通知 "Health checks console feedback collection" with the subtext "To help us improve the Health Check user experience, please take 5 minutes to complete this survey." A table lists one health check entry:

Name	Status	Description	Alarms	ID
demo-route-01-health-check	Unknown	http://3.8.122.201:80/	No alarms configured.	091b6812-ed30-4202-b396-19f172c91a5f

15. After come back to Hosted zones and click on create record. Choose the record type as A. Then give the public IP address of your instance.
16. In the routing policy choose Failover policy. Give health check ID which you just created.
17. Then choose failover record type to primary and give record ID as 1.

The screenshot shows the "Quick create record" interface for Route 53. It's set up to create a new record named "subdomain" pointing to "cloudservicesdemo.in". The "Record type" is selected as "A - Routes traffic to an IPv4 address and some AWS resources". The "Value" field contains "3.8.122.201". The "TTL (seconds)" is set to "300". The "Routing policy" dropdown is set to "Failover" and is highlighted with a red box. The "Health check ID" dropdown contains "091b6812-ed30-4202-b396-19f172c91a5f". The "Failover record type" dropdown is set to "Primary". The "Record ID" field is set to "1".

18. After that click on create another record. This time give the public IP address of your secondary instance. Again, choose policy as Failover.
19. Then record type as Secondary and record ID as 2.
20. Click on create record.

▼ Record 2

Record name [Info](#) **subdomain** [cloudservicesdemo.in](#)

Keep blank to create a record for the root domain.

Alias

Value [Info](#) **13.40.210.49**

Enter multiple values on separate lines.

TTL (seconds) [Info](#) **300** **1m** **1h** **1d**

Recommended values: 60 to 172800 (two days)

Routing policy [Info](#) **Failover**

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.

Secondary

Health check ID - optional [Info](#) **Choose health check**

Record ID [Info](#) **2**

Delete

<input type="checkbox"/>	Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (seconds)	Health
<input type="checkbox"/>	cloudservicesdemo.in	A	Failover	Primary	No	3.8.122.201	300	091b68
<input type="checkbox"/>	cloudservicesdemo.in	A	Failover	Secondary	No	13.40.210.49	300	-

21. Now go to your domain name **cloudservicesdemo.in/Index.html**

22. You will see that it is working as expected and we are routed onto the primary instance server.



This is the instance with name demo-route-01

23. Now if you will go to the health checks. You will see that it is in healthy state.

[Create health check](#) [Delete health check](#) [Edit health check](#)

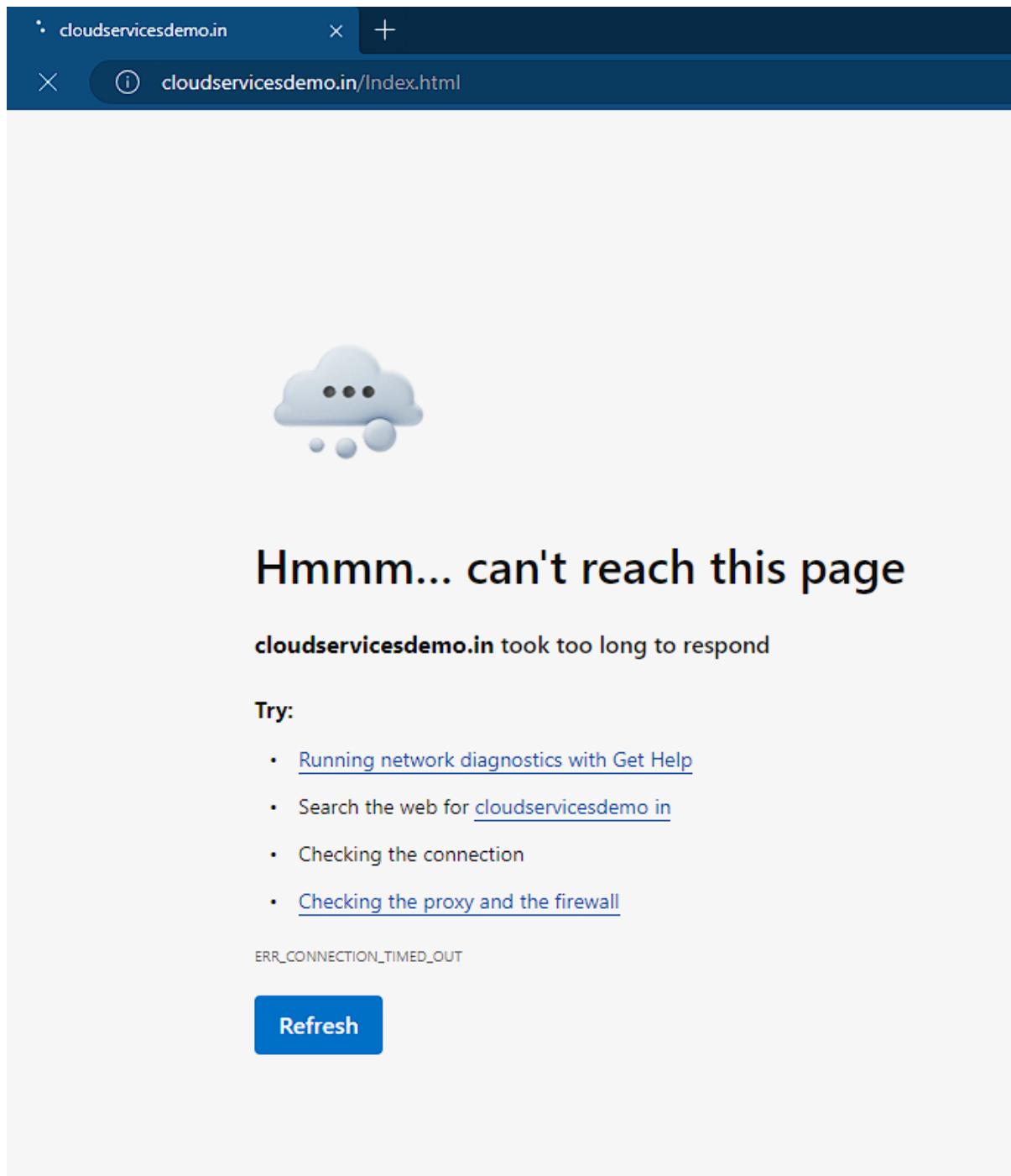
Filter by keyword

Name	Status	Description	Alarms	ID
<input type="checkbox"/> demo-route-01-health-check	15 minutes ago now Healthy	http://3.8.122.201:80/	No alarms configured.	091b6812-ed30-4202-b396-19f172c91a5f

24. To simulate the failure go to EC2 instances and stop your primary instance then wait for some time till it get stopped.

25. Now if you'll go back to your domain page and refresh the page you will see that it trying to get to the server but it can not reach it.

26. You'll see that it has failed.



27. After couple of minutes it has redirected towards your secondary instance.

