

```
api
root@ip-172-31-37-156:~/python-reach-sample/api-server-flask# cd ..
root@ip-172-31-37-156:~/python-reach-sample# ls
CHANGELOG.md  LICENSE.md  README.md  api-server-flask  install-steps.txt  package.json  react-ui
root@ip-172-31-37-156:~/python-reach-sample# cd react-ui/
root@ip-172-31-37-156:~/python-reach-sample/react-ui# ls
CHANGELOG.md  Dockerfile  LICENSE.md  docker-compose.yml  env.sample  package.json  public  src  web.config  yarn.lock
root@ip-172-31-37-156:~/python-reach-sample/react-ui# cd src/
root@ip-172-31-37-156:~/python-reach-sample/react-ui/src# ls
App.js  config.js  hooks  index.js  layout  menu-items  routes  serviceWorker.js  store  themes  ui-component  utils  views
```

Under config.js file change the public ip with you instance public ip

```
let BACKEND_SERVER = null;
if (process.env.REACT_APP_BACKEND_SERVER) {
  BACKEND_SERVER = process.env.REACT_APP_BACKEND_SERVER;
} else {
  BACKEND_SERVER = "http://13.201.48.10:5000/api/";
}

const config = {
  // basename: only at build time to set, and don't add '/' at end off BASENAME for breadcrumbs, also don't put only '/' use blank('') instead,
  // like '/berry-material-react/react/default'
  basename: '',
  defaultPath: '/dashboard/default',
  fontFamily: 'Roboto',
  borderRadius: 12,
  API_SERVER: BACKEND_SERVER
};

export default config;
```

Run docker-compose up -d

```

root@ip-172-31-37-156: /python-reach-sample/react-ui/src/ vi config.js
root@ip-172-31-37-156: /python-reach-sample/react-ui/src/ cd ..
root@ip-172-31-37-156: /python-reach-sample/react-ui# docker-compose up -d
WARN[0000] /root/.python-reach-sample/react-ui/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 111.9s (8/10)
=> [sample internal] load build definition from Dockerfile
=> => transferring dockerfile: 407B
=> [sample internal] load metadata for docker.io/library/node:l4-alpine
=> [sample internal] load .dockerignore
=> transferring context: 79B
=> [sample internal] load build context
=> => transferring context: 1.53MB
=> [sample 1/6] FROM docker.io/library/node:l4-alpine@sha256:434215b487a329c9e867202f7f89e704d3a75e554822e07f3e0cf9e606121b33
=> resolve docker.io/library/node:l4-alpine@sha256:434215b487a329c9e867202f7f89e704d3a75e554822e07f3e0cf9e606121b33
=> sha256:434215b487a329c9e867202f7f89e704d3a75e554822e07f3e0cf9e606121b33 2.37MB / 2.37MB
=> sha256:434215b487a329c9e867202f7f89e704d3a75e554822e07f3e0cf9e606121b33 1.43KB / 1.43KB
=> sha256:4e84c956c6d2726af9ed14a8b2939a7343642cb0d4248b5e90eb19717573dfdc548 1.16KB / 1.16KB
=> sha256:0dad3dc27b1ad570e6c3a7f7cd29e88e7130ff0cad3b12bac5a0f222f7e971b0b 6.44KB / 6.44KB
=> sha256:f56be855fc2e246face30e2cd3e37f7fe7c15f8fd7c4e5add28d7f6487abddaa09 3.37MB / 3.37MB

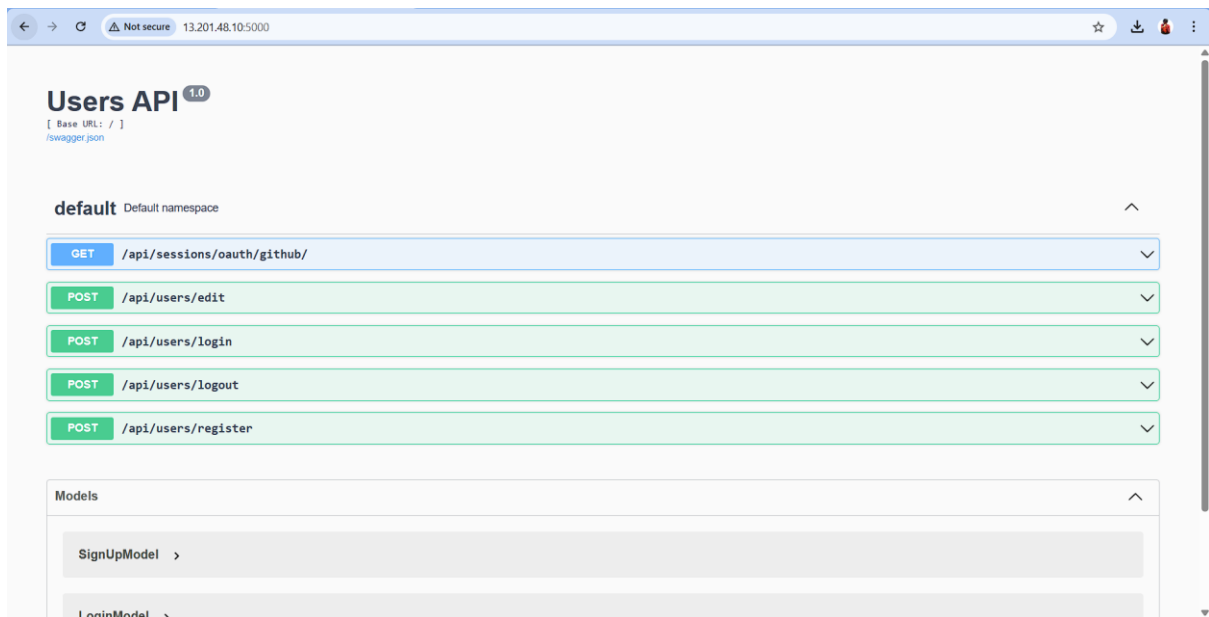
```

```

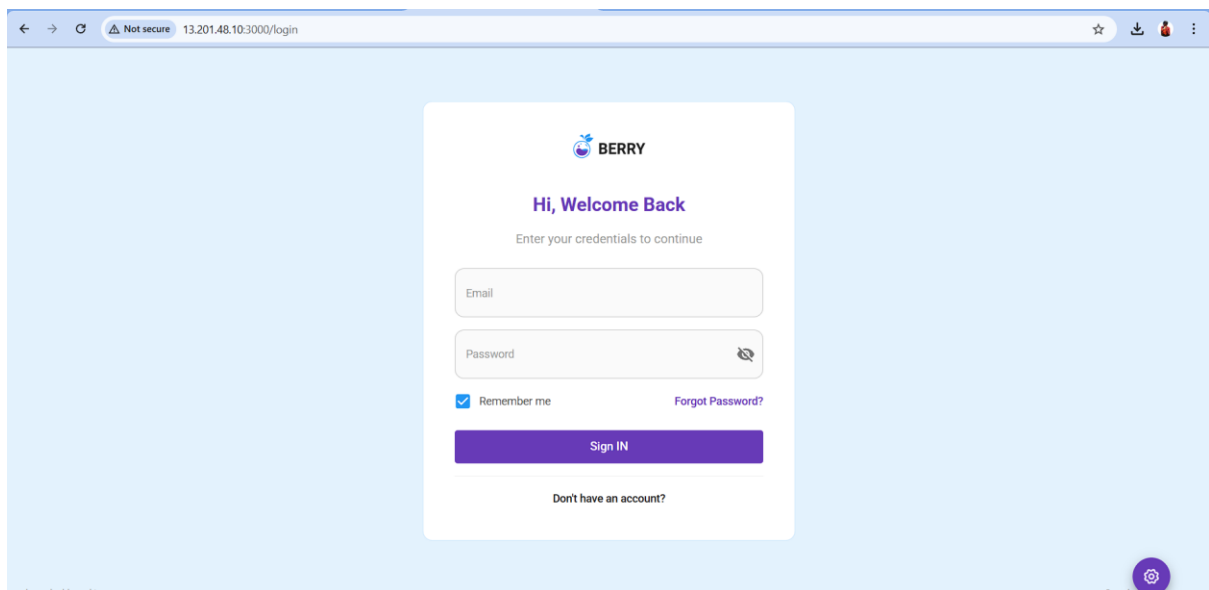
-> sha256sum sha256:5f16b69c53dca97c5b5e02128e496fd434ab4d4dc2f2086d79a2255fear 0.0s
-> [sample 2/6] WORKDIR /app 0.2s
-> [sample 3/6] COPY package.json ./ 0.0s
-> [sample 4/6] RUN npm install --silent 73.0s
-> [sample 5/6] RUN npm install react-scripts@4.0.0 -g --silent 57.0s
-> [sample 6/6] COPY ./ 0.1s
-> [sample] exporting to image 19.3s
-> exporting layers 19.3s
-> writing image sha256:d76ef263b1618abd439cd56bb8eb172f561b7d4d8065f1e3d12ac47bc9e75b 0.0s
-> naming to docker.io/library/react-ui-sample 0.0s
-> [sample] resolving provenance for metadata file 0.0s
[+] Running 3/3 0.0s
✓ sample Built 0.0s
✓ Network react-ui_default Created 0.1s
✓ Container react-Berry Started 12.0s
root@ip-172-31-37-156:~/python-reach-sample/react-ui#

```

In browser hit 13.201.48.10:5000 and check api page is working



Run 13.201.48.10:3000 and check ui page is working



Use response API and send request to register

POST

/api/users/register

Parameters

Cancel

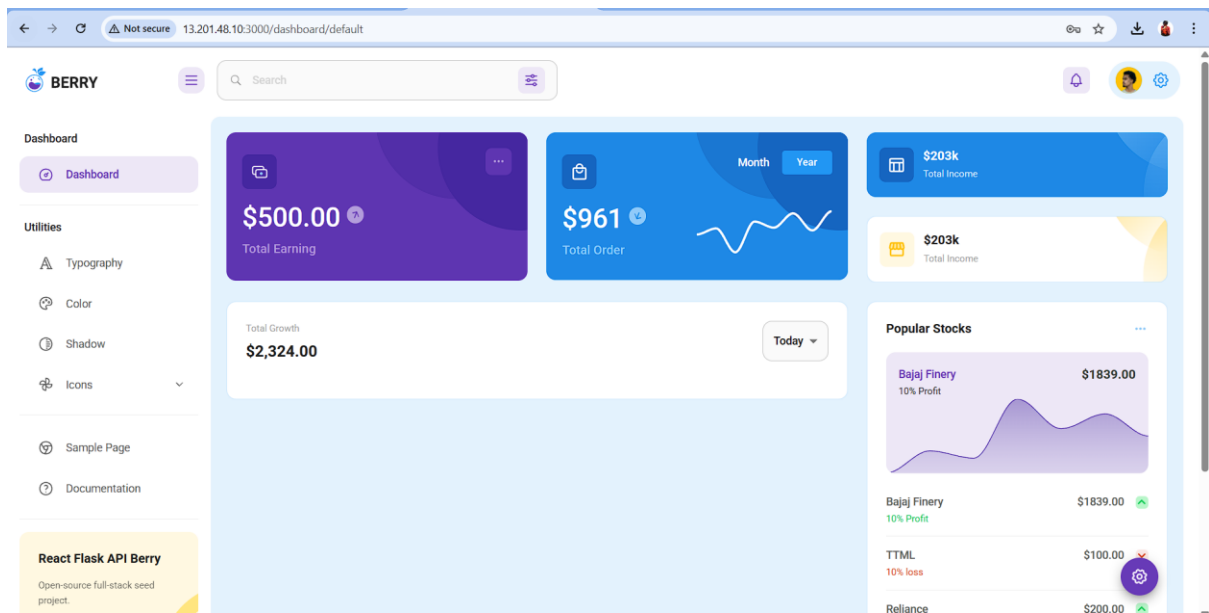
Name	Description
<div><div>payload</div><div>object</div><div>(body)</div></div> <div><div>Edit Value</div><div>Model</div></div>	<div><pre>{ "username": "Nikhil Patil", "email": "nikhilpatil1807@gmail.com", "password": "Password@123" }</pre></div> <div><div>Cancel</div></div> <div><div>Parameter content type</div><div>application/json</div></div>

Execute

Clear

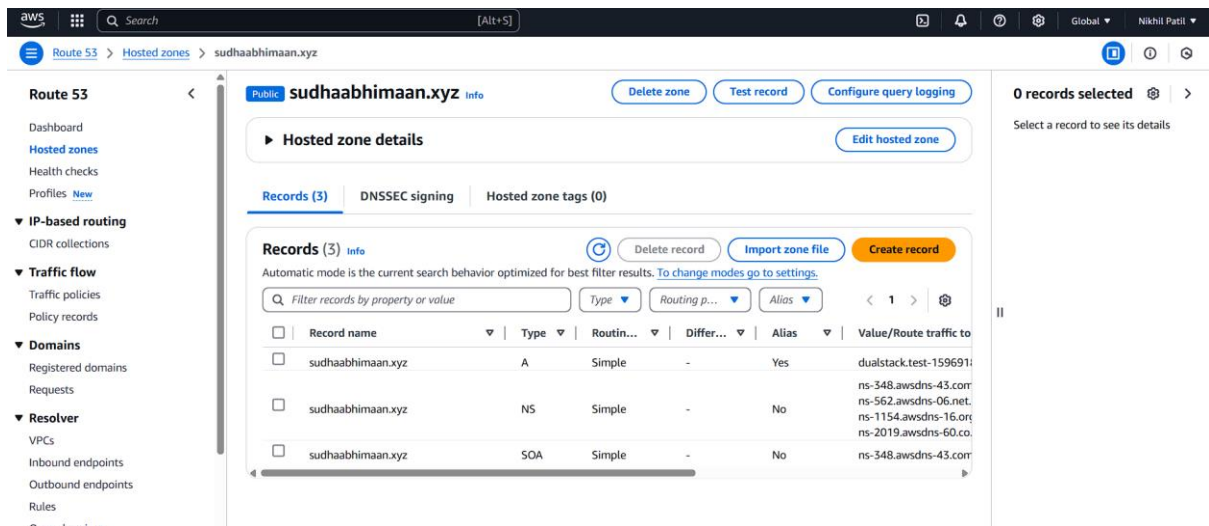
[illegible]

Try login in using email id and password



Map the docker machine to the Route 53 by Load balancer with targeted group to route the traffic.

Create the route 53 and map the DNS as mentioned below:



Create target group with the docker machine for UI and API with 3000 and 5000 ports

API

The screenshot shows the AWS Management Console interface for a target group named 'ui'. The left sidebar contains navigation links for EC2, Instances, Images, and Elastic Block Store. The main content area displays the 'Details' tab for the target group. The details include the ARN, target type (Instance), IP address type (IPv4), protocol (HTTP), port (80), protocol version (HTTP1), and VPC ID. A table below shows the status of the target group: 1 total target, 0 healthy, 0 unhealthy, 1 unused, 0 initial, and 0 draining. The 'Distribution of targets by Availability Zone (AZ)' section is also visible.

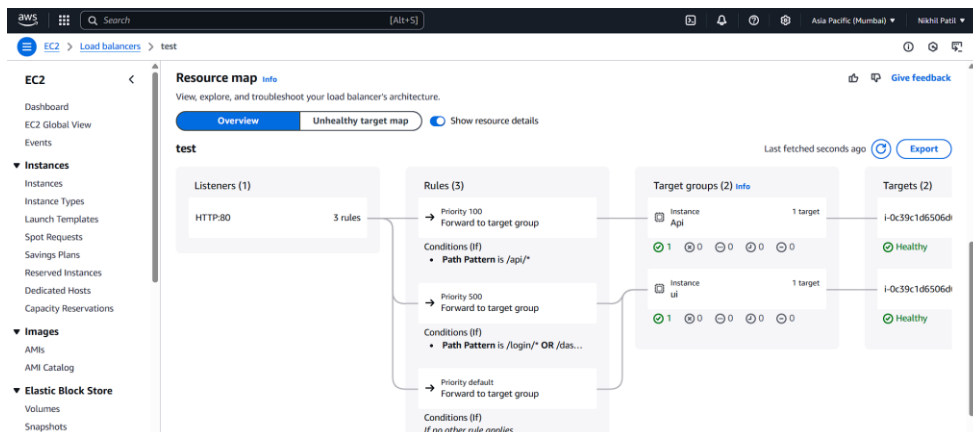
Target type	Healthy	Unhealthy	Unused	Initial	Draining
1 Total targets	0	0	1	0	0

UI

The screenshot shows the AWS Management Console interface for a target group named 'ui'. The left sidebar contains navigation links for Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'Details' tab for the target group. The details include the ARN, target type (Instance), IP address type (IPv4), protocol (HTTP), port (80), protocol version (HTTP1), and VPC ID. A table below shows the status of the target group: 1 total target, 1 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining. The 'Distribution of targets by Availability Zone (AZ)' section is also visible.

Target type	Healthy	Unhealthy	Unused	Initial	Draining
1 Total targets	1	0	0	0	0

Create a load balancer and add the target groups



Add the Listeners and rules

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

...

RulesAttributesTags

Listener rules (3) [Info](#)

Rule limits ⓘ Actions ▼ Add rule

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)	ARN	Tag
<input type="checkbox"/>	api-path	100	Path Pattern is /api/*	Forward to target group <ul style="list-style-type: none">Api 1 (100%)Target group stickiness: Off	ARN	1 ta
<input type="checkbox"/>	ui	500	Path Pattern is /login/* OR /dashboard/*	Forward to target group <ul style="list-style-type: none">ui 1 (100%)Target group stickiness: Off	ARN	1 ta
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none">Api 1 (100%)Target group stickiness: Off	ARN	0 ta

Add the Load balancer to the Route 53

Route 53Hosted zonessudhaabhimaan.xyzCreate record

Create record [Info](#)

Quick create record

Switch to wizard

▼ Record 1Delete

Record name [Info](#)

sudhaabhimaan.xyz

Keep blank to create a record for the root domain.

Record type [Info](#)

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

Alias

Route traffic to [Info](#)

Alias to Application and Classic Load Balancer

Asia Pacific (Mumbai)

Alias hosted zone ID: ZP97RAFLXTNZK

Routing policy [Info](#)

Simple routing

Evaluate target health

☒ Yes

Try accessing the application with DNS name :

Not secure sudhaabhimaan.xyz/login

BERRY

Hi, Welcome Back

Enter your credentials to continue

Email
nikhilpatil027@gmail.com

Password
Password@123

☒ Remember me [Forgot Password?](#)

Sign IN

Don't have an account?

berrydashboard.io © codeshapes.com

