INTRODUCTION:-

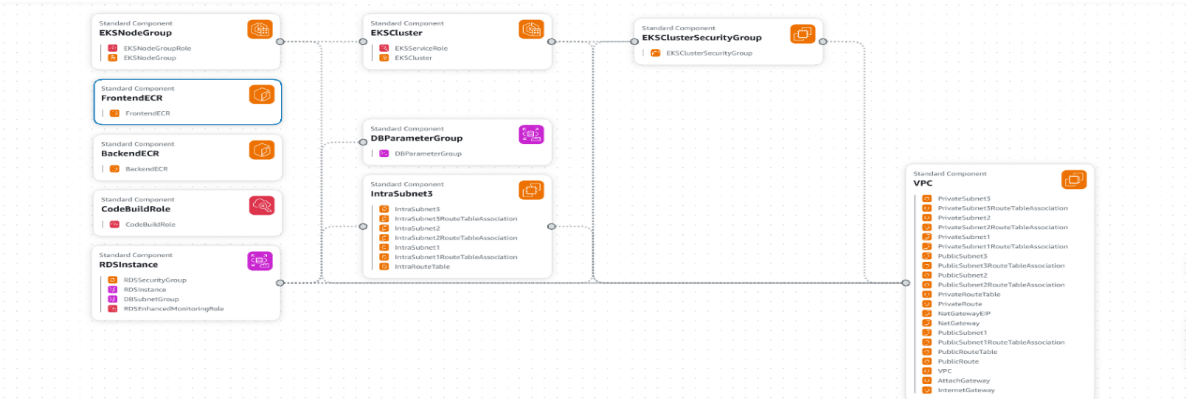
This document provides a comprehensive overview of the **cloudops-demo** three-tier application. It details the architecture, technologies used in each layer, deployment strategy on AWS, and key components, along with guidance for understanding and troubleshooting the system.

Project Purpose:- A web application for managing user profiles, responsive frontend and robust backend.

Three tier Architecture Overview :-

The application is deployed on AWS, leveraging Amazon Elastic Kubernetes Service (EKS) for container orchestration, Amazon RDS for the relational database, and various networking and IAM components to ensure secure and scalable operations.

- **Frontend**: React application
- **Backend**: Node.js/Express API
- **Database**: AWS RDS PostgreSQL



Phase 1:-Deploy an multi tier Application on EKS Cluster

Initially created git repo Devops_main_Project

```
root@ip-172-31-95-241:/home/ubuntu# git clone git@github.com:asnashameel/Devops_main_Project.git
Cloning into 'Devops_main_Project'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
Receiving objects: 100% (47/47), 25.49 KiB | 12.74 MiB/s, done.
Resolving deltas: 100% (1/1), done.
remote: Total 47 (delta 1), reused 47 (delta 1), pack-reused 0 (from 0)
root@ip-172-31-95-241:/home/ubuntu# ls
3-tier-dep  Devops_main_Project  awscliv2.zip  get-docker.sh
root@ip-172-31-95-241:/home/ubuntu# cd Devops_main_Project/
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# ls
README.md  backend  buildspec.yml  database  docker-compose.yml  env.example  frontend  infrastructure  k8s  scripts  terraform
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# cd frontend
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# ls
Dockerfile  nginx.conf  package.json  public  src
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# nano Dockerfile
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend#
```


```
root@ip-172-31-95-241:/home/ubuntu# git clone git@github.com:asnashameel/Devops_main_Project.git
Cloning into 'Devops_main_Project'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
Receiving objects: 100% (47/47), 25.49 KiB | 12.74 MiB/s, done.
Resolving deltas: 100% (1/1), done.
remote: Total 47 (delta 1), reused 47 (delta 1), pack-reused 0 (from 0)
root@ip-172-31-95-241:/home/ubuntu# ls
3-tier-dep  Devops_main_Project  awscliv2.zip  get-docker.sh
root@ip-172-31-95-241:/home/ubuntu# cd Devops_main_Project/
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# ls
README.md  backend  buildspec.yml  database  docker-compose.yml  env.example  frontend  infrastructure  k8s  scripts  terraform
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# cd frontend
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# ls
Dockerfile  nginx.conf  package.json  public  src
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# nano Dockerfile
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend#
```


Created docker image for my frontend application by using docker build -t frontend .

```
[+] Building 175.0s (16/16) FINISHED                                                                                                docker:defau
 => [internal] load build definition from Dockerfile                                                                                          0.
 => => transferring dockerfile: 683B                                                                                                          0.
 => [internal] load metadata for docker.io/library/nginx:alpine                                                                               0.
 => [internal] load metadata for docker.io/library/node:18-alpine                                                                             0.
 => [internal] load .dockerignore                                                                                                             0.
 => => transferring context: 2B                                                                                                               0.
 => [builder 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e              0.
 => [stage-1 1/4] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10                0.
 => [internal] load build context                                                                                                             0.
 => => transferring context: 893B                                                                                                             0.
 => CACHED [stage-1 2/4] COPY nginx.conf /etc/nginx/conf.d/default.conf                                                                       0.
 => CACHED [builder 2/6] WORKDIR /app                                                                                                         0.
 => CACHED [builder 3/6] COPY package*.json ./                                                                                                0.
 => [builder 4/6] RUN npm install                                                                                                            99.
 => [builder 5/6] COPY . .                                                                                                                    0.
 => [builder 6/6] RUN npm run build                                                                                                          72.
 => [stage-1 3/4] COPY --from=builder /app/build /usr/share/nginx/html                                                                        0.
```

```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend/public# docker ps -a
CONTAINER ID   IMAGE                                                                 COMMAND                  CREATED          STATUS
                  PORTS                                                 NAMES
49bd34737f2c   207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend              "/docker-entrypoint.…"   20 seconds ago   Up 20 seconds
 (health: starting)   80/tcp, 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   admiring_brattain
bacc8afde4b7   207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend              "/docker-entrypoint.…"   11 minutes ago   Exited (1) 11
 minutes ago                                                                 vibrant_ganguly
cd24b00dfa61   207567798584.dkr.ecr.us-east-1.amazonaws.com/3-tier-frontend:latest   "docker-entrypoint.s…"   3 days ago       Exited (0) 27
 hours ago                                                                   elegant_driscoll
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend/public# cd ..
```

Tag the image

```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# docker tag dev-frontend 207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-fr
ontend:latest
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# docker push 207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend:lates
t
The push refers to repository [207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend]
5a9884b19b06: Pushed
69a9ffc140c3: Pushed
03fcfef73e03: Pushed
0d853d50b128: Pushed
947e805a4ac7: Pushed
811a4dbbf4a5: Pushed
b8d7d1d22634: Pushed
e244aa659f61: Pushed
c56f134d3805: Pushed
d71eae0084c1: Pushed
08000c18d16d: Pushed
latest: digest: sha256:51dce67f288098bdaead3bd2f414bac5c69a3ad51ff94a5a8449483136048e7c size: 2615
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend#
```


```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# docker tag dev-frontend 207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-fr
ontend:latest
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend# docker push 207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend:lates
t
The push refers to repository [207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-frontend]
5a9884b19b06: Pushed
69a9ffc140c3: Pushed
03fcfef73e03: Pushed
0d853d50b128: Pushed
947e805a4ac7: Pushed
811a4dbbf4a5: Pushed
b8d7d1d22634: Pushed
e244aa659f61: Pushed
c56f134d3805: Pushed
d71eae0084c1: Pushed
08000c18d16d: Pushed
latest: digest: sha256:51dce67f288098bdaead3bd2f414bac5c69a3ad51ff94a5a8449483136048e7c size: 2615
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend#
```



```
[+] Building 175.0s (16/16) FINISHED                                                                                                docker:defau
 => [internal] load build definition from Dockerfile                                                                                          0.
 => => transferring dockerfile: 683B                                                                                                          0.
 => [internal] load metadata for docker.io/library/nginx:alpine                                                                               0.
 => [internal] load metadata for docker.io/library/node:18-alpine                                                                             0.
 => [internal] load .dockerignore                                                                                                             0.
 => => transferring context: 2B                                                                                                               0.
 => [builder 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e              0.
 => [stage-1 1/4] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10                0.
 => [internal] load build context                                                                                                             0.
 => => transferring context: 893B                                                                                                             0.
 => CACHED [stage-1 2/4] COPY nginx.conf /etc/nginx/conf.d/default.conf                                                                       0.
 => CACHED [builder 2/6] WORKDIR /app                                                                                                         0.
 => CACHED [builder 3/6] COPY package*.json ./                                                                                                0.
 => [builder 4/6] RUN npm install                                                                                                            99.
 => [builder 5/6] COPY . .                                                                                                                    0.
 => [builder 6/6] RUN npm run build                                                                                                          72.
 => [stage-1 3/4] COPY --from=builder /app/build /usr/share/nginx/html                                                                        0.
```


```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/frontend/public# docker run -d -p 3000:3000 207567798584.dkr.ecr.us-east-1.amazonaws.
com/dev-frontend sleep 2000
49bd34737f2ce7963adb86dc125f838478557e731851c4d16dfefac188d4e9c6
```

container created by using image

```
[+] Building 13.3s (13/13) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                            0.0s
 => => transferring dockerfile: 584B                                                                            0.0s
 => [internal] load metadata for docker.io/library/node:18-alpine                                               0.1s
 => [internal] load .dockerignore                                                                               0.0s
 => => transferring context: 2B                                                                                 0.0s
 => [1/8] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e  0.0s
 => [internal] load build context                                                                               0.0s
 => => transferring context: 7.33kB                                                                             0.0s
 => CACHED [2/8] WORKDIR /app                                                                                   0.0s
 => [3/8] RUN addgroup -g 1001 -S nodejs                                                                        0.2s
 => [4/8] RUN adduser -S nodejs -u 1001                                                                         0.2s
 => [5/8] COPY package*.json ./                                                                                 0.1s
 => [6/8] RUN npm install                                                                                       7.7s
 => [7/8] COPY . .                                                                                              0.0s
 => [8/8] RUN chown -R nodejs:nodejs /app                                                                       4.3s
 => exporting to image                                                                                          0.5s
 => => exporting layers                                                                                         0.4s
 => => writing image sha256:faf11f7ae12c99bb3d284cd0d2f317cfe4bd1c3cd39b9238deb49fdb914d93ea                    0.0s
 => => naming to docker.io/library/dev-backend                                                                  0.0s
root@ip-172-31-95-241:/home/ubuntu/Devops main Project/backend#
```

Build docker image for frontend,tagged the image and pushed to ecr repo

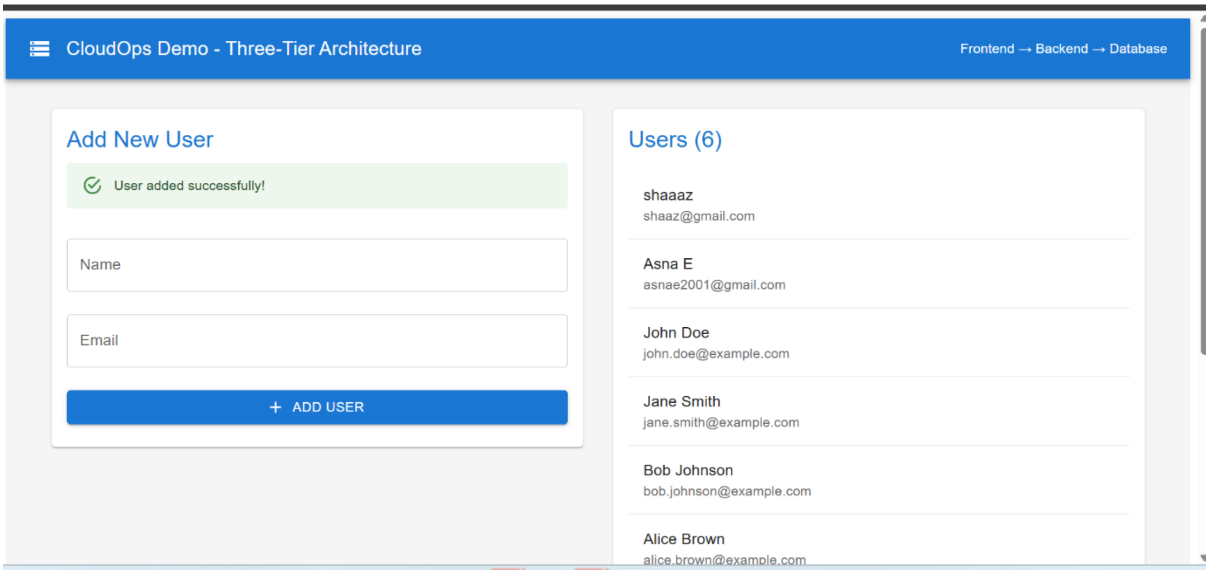
```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/backend# docker tag dev-backend  207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-bac
kend:latest
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/backend# docker push 207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-backend:latest
The push refers to repository [207567798584.dkr.ecr.us-east-1.amazonaws.com/dev-backend]
af32f1b7d4a7: Pushed
545c6ea37ffa: Pushed
123c077b176d: Pushed
5513b7967802: Pushed
f53fc8f26ed6: Pushed
ca5cd0fd36fa: Pushed
777dc634dbf5: Pushed
82140d9a70a7: Pushed
f3b40b0cdb1c: Pushed
0b1f26057bd0: Pushing [=================================>                  ]  74.58MB/113.8MB
08000c18d16d: Pushed
```

Next are the deployment steps

Initially created cluster

Deployed all yaml files

```
root@ip-172-31-95-241:/home/ubuntu# eksctl create cluster --name kube-cluster --region us-east-1 --node-type t2.medium --nodes-min 2 --nod
es
-max 2
2025-06-19 16:51:09 [ℹ]  eksctl version 0.210.0
2025-06-19 16:51:09 [ℹ]  using region us-east-1
2025-06-19 16:51:09 [ℹ]  setting availability zones to [us-east-1c us-east-1d]
2025-06-19 16:51:09 [ℹ]  subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2025-06-19 16:51:09 [ℹ]  subnets for us-east-1d - public:192.168.32.0/19 private:192.168.96.0/19
2025-06-19 16:51:09 [ℹ]  nodegroup "ng-6659c31a" will use "" [AmazonLinux2023/1.32]
2025-06-19 16:51:09 [ℹ]  using Kubernetes version 1.32
2025-06-19 16:51:09 [ℹ]  creating EKS cluster "kube-cluster" in "us-east-1" region with managed nodes
2025-06-19 16:51:09 [ℹ]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-06-19 16:51:09 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1
 --cluster=kube-cluster'
2025-06-19 16:51:09 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "kube-clu
ster" in "us-east-1"
2025-06-19 16:51:09 [ℹ]  CloudWatch logging will not be enabled for cluster "kube-cluster" in "us-east-1"
2025-06-19 16:51:09 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all
)} --region=us-east-1 --cluster=kube-cluster'
2025-06-19 16:51:09 [ℹ]  default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
```

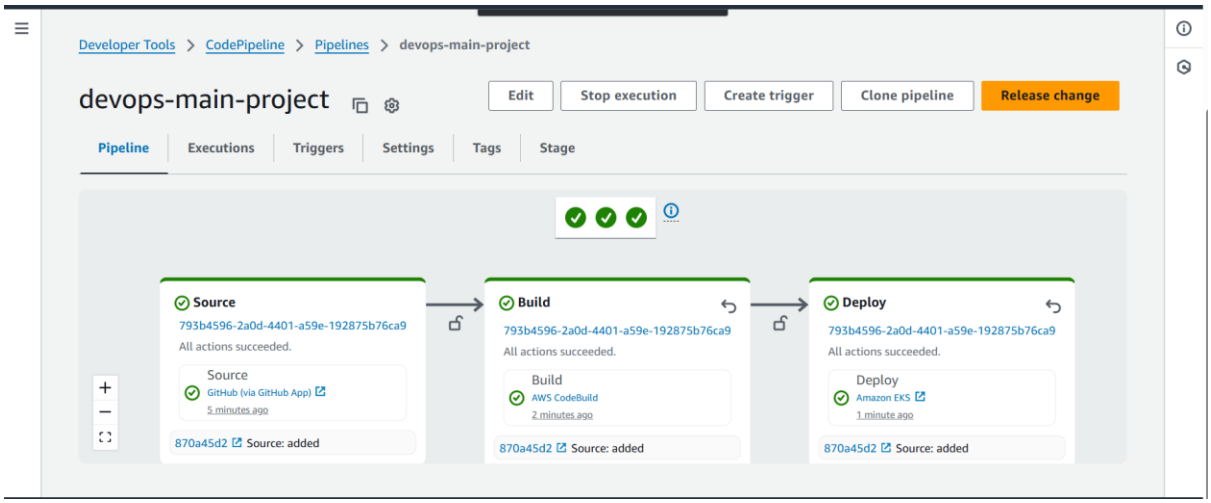


Accessed the application by using host address

Found application is working fine

```
2025-06-19 17:04:54 [ℹ]  cluster should be functional despite missing (or misconfigured) client binaries
2025-06-19 17:04:54 [✓]  EKS cluster "kube-cluster" in "us-east-1" region is ready
root@ip-172-31-95-241:/home/ubuntu# aws eks update-kubeconfig --region us-east-1 --name kube-cluster
Added new context arn:aws:eks:us-east-1:207567798584:cluster/kube-cluster to /root/.kube/config
root@ip-172-31-95-241:/home/ubuntu# kubectl get nodes
NAME                          STATUS   ROLES    AGE   VERSION
ip-192-168-24-77.ec2.internal   Ready    <none>   16m   v1.32.3-eks-473151a
ip-192-168-55-63.ec2.internal   Ready    <none>   16m   v1.32.3-eks-473151a
root@ip-172-31-95-241:/home/ubuntu# cd Devops_main_Project/
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# ls
README.md  backend  buildspec.yml  database  docker-compose.yml  env.example  frontend  infrastructure  k8s  scripts  terraform
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project# cd k8s/
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s# ls
00-namespace.yaml  02-secrets.yaml            08-backend-service.yaml      10-frontend-service.yaml  12-hpa.yaml
01-configmap.yaml  07-backend-deployment.yaml  09-frontend-deployment.yaml  11-ingress.yaml           13-network-policy.yaml
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s# nano 09-frontend-deployment.yaml
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s# kubectl create namespace cloudops-demo
namespace/cloudops-demo created
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s#
```

Phase2:- Automate the Previous deployment using CodePipeline

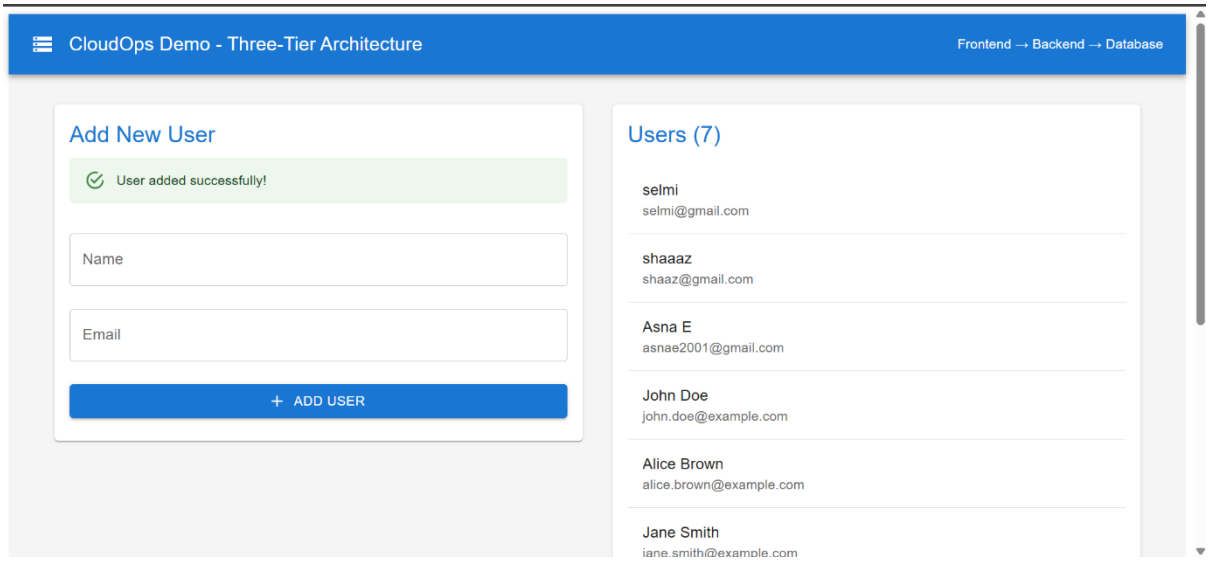


It will deploy all yaml files in kubernetes manifests

```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s# kubectl get ing -n cloudops-d
NAME                CLASS    HOSTS
cloudops-ingress    <none>   a4160f7605eda49cfa12fbfbbcd5a3a6-1839d92ed829f63a.elb.us-east
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s#
```

Kubectl get ing -n cloudops-demo,will get host address of ingress

```
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s# kubectl get ing -n cloudops-demo
NAME                CLASS    HOSTS                                                                  ADDRESS   PORTS   AGE
cloudops-ingress    <none>   a4160f7605eda49cfa12fbfbbcd5a3a6-1839d92ed829f63a.elb.us-east-1.amazonaws.com             80      40h
root@ip-172-31-95-241:/home/ubuntu/Devops_main_Project/k8s#
```

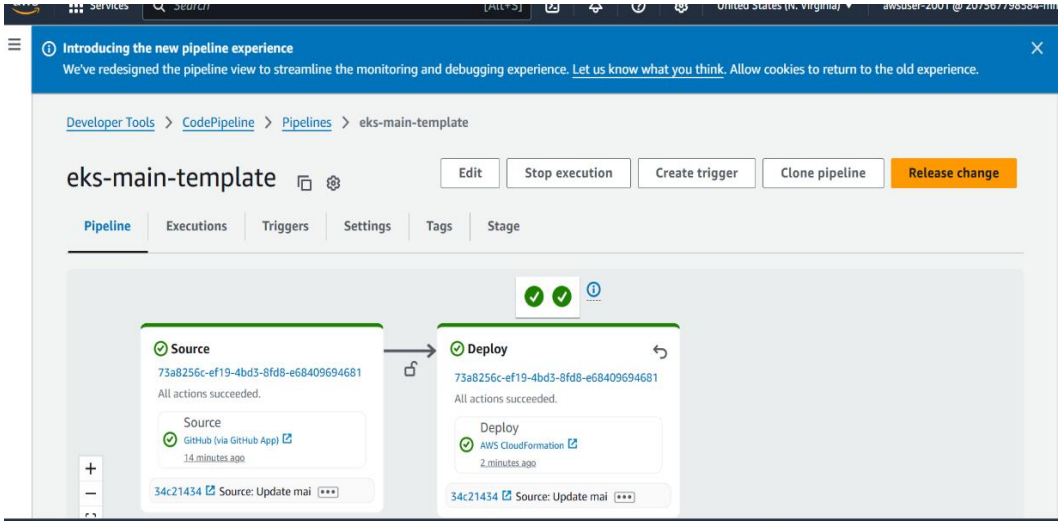


Accessed the application,successfully getting it

Phase3:-Expose the deployment in multiple AWS region (Terraform in one region and cloudformation in another region)

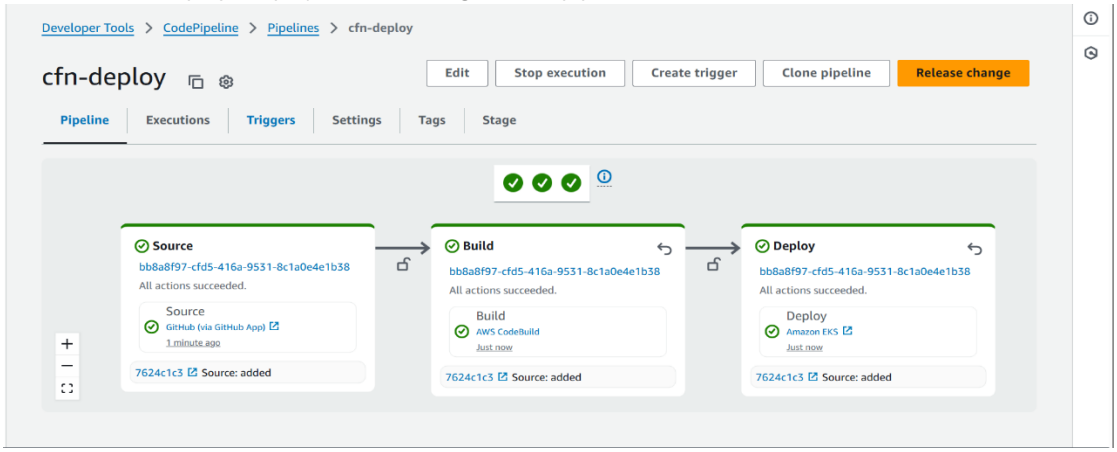
Created infrastructure using cloudformation in us-east-1 region

Created template for that eks-main-template.yaml

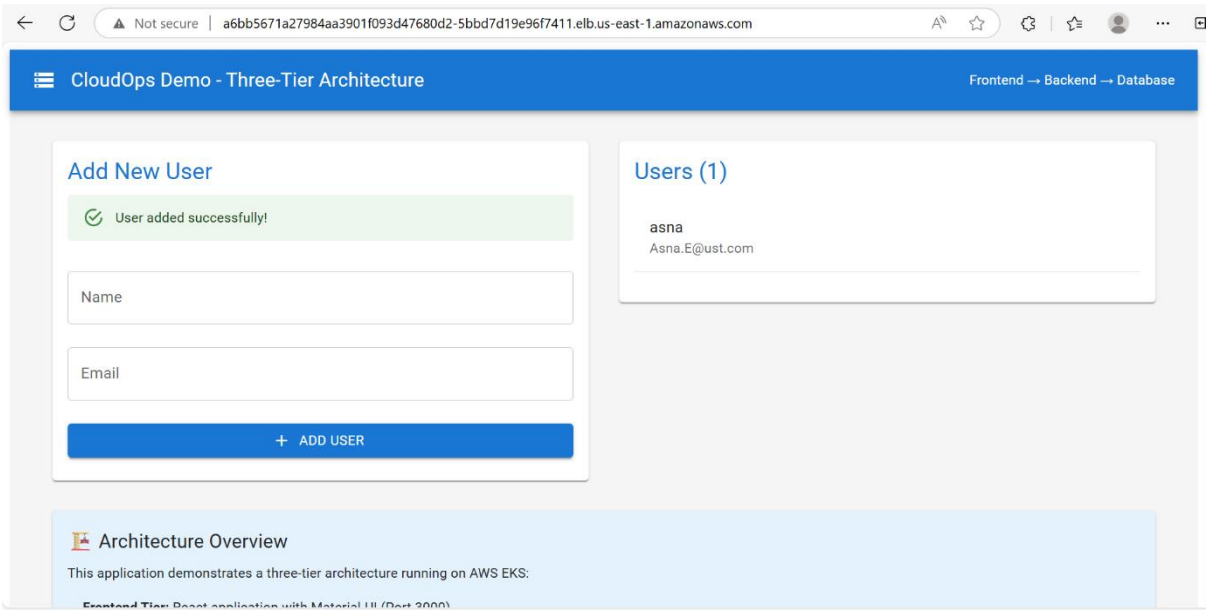


Initially created one pipeline for creating infrastructure and deployed in AWS cloudformation

Then build and deploy the project in eks using another pipeline

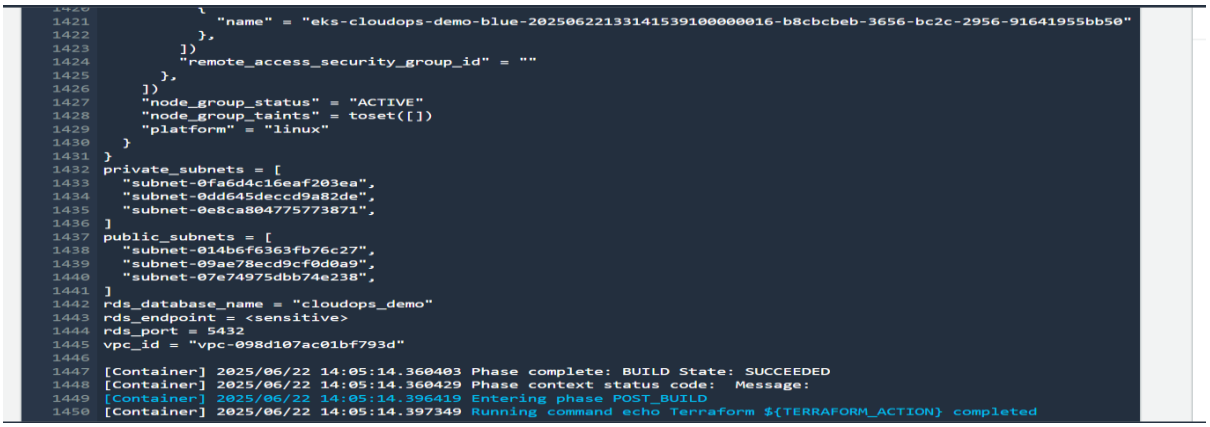


Accessed the application ,yeah it is working fine



Terraform:-

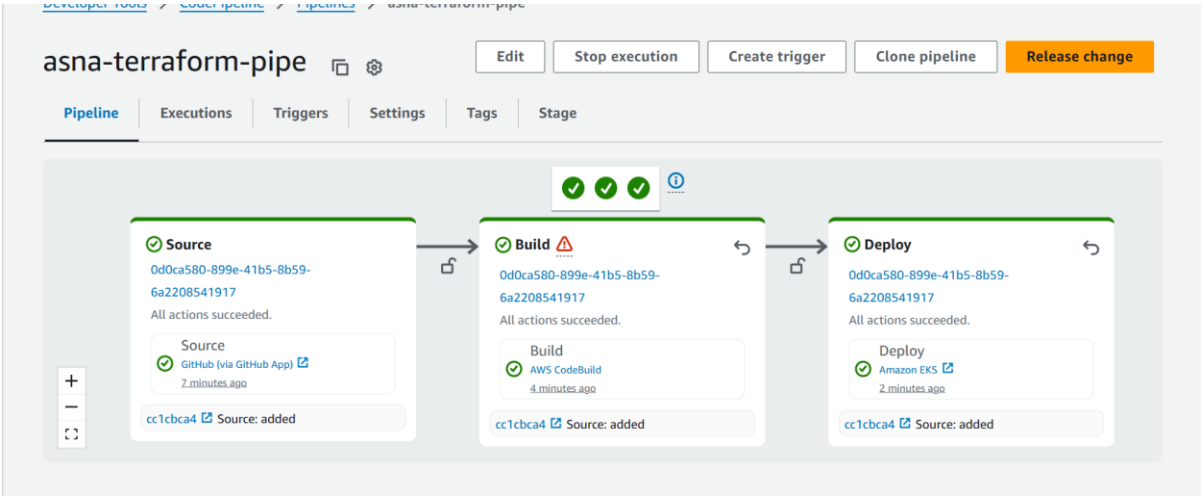
Created terraform infrastructure using one pipeline that contain buildspec.yaml for terraform init plan and apply



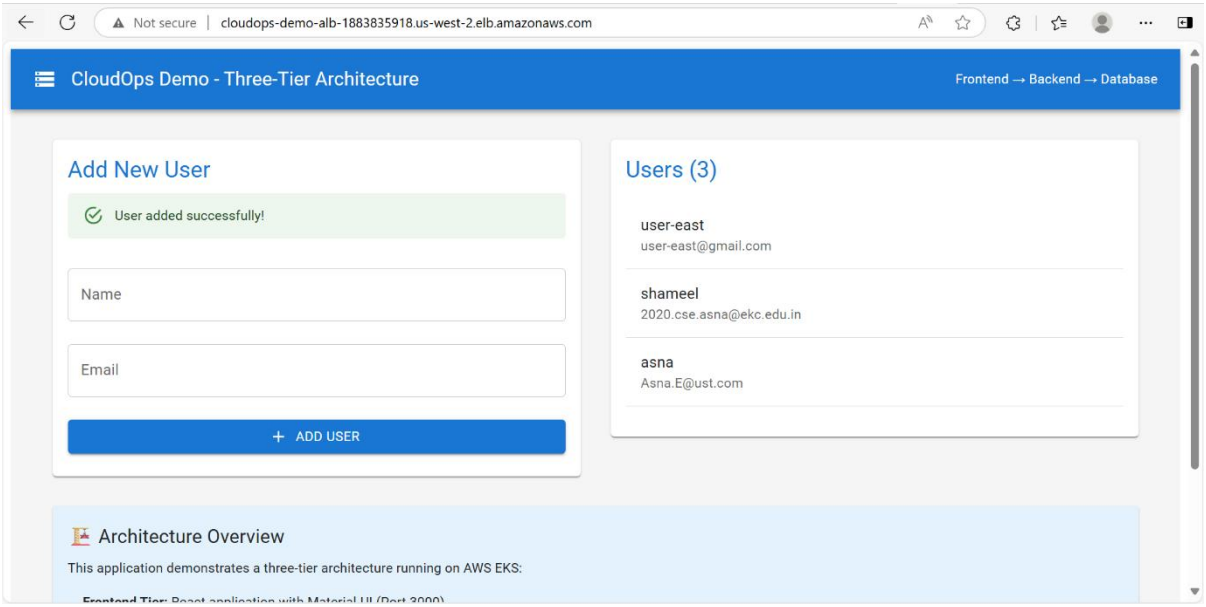
```
fi

=== Infrastructure Outputs ===
cluster_certificate_authority_data =
"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJT1lHZWtYN1FMYWt3RFFZSktvWklodmNOQVFFTEJRQXdG
VEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TlRBMk1qSXhNekl3TkRKYUZ3MHpOVEEyTWpBeE16STFOREphTUJVeApFekFSQ
mdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURZOHdneUVZcnBEdmpodX
ExWDgrOW5QZGxGaldLRXlMakRHQk1XSm9OOVpWTjFTdFhMMml3eEh1bk4KMldFSktFNEF3djVuY29mVlZKZE9vOXEwZ24wRE9jMU5nU011L3p
mSTlNZU1WZmhHTklmTEpLaERaaHFOZzJOKwpQc2FsMis1UXg5Q2xESFJtZGVqd0RON2FiMDg4NmU4ZG1lNzhFanJNeFpVNE1YZ1p4NStMS0x1
TURnWU9tc29XCkRES1VMNERMNVl2SHRhTSs3Q3p5bzJ6cGJyVjBVRG00OEZ2Ujd3dXoySmN3RDgzMFUxYmg3MTFWeFVialBrMjcKdUNNVHZQO
TJuNDlnV3VTWS9Za1o1M1dTRDB6a0JvV3RTam9pbnhxWDhqdkxTeTh2czdoVzZvMXd2SFdsd3NnVQo4T0lhZytIQVdTZk1OdHc0Tk84aEJnSk
JDOGkvQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSdTZYT2t
2ZTBUWW9oUmRyZ3d4QWdFSFdKZGFqQVYKQmdOVkhSUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRRFBC
ZVFJSjFJZApHcDdVUTZ3RVJibnRRd2ErVDJNQXJ2OHU5Z0pzaWhHZzVLTHVJc0Vic3BFVTROQ3lGYWVIWHBnMWpsd0RaNUFvCmtnS0ZVMXFvT
1J6MHFMdm1qcUlmUjlyV2NUUlJGcU1zejkybENwOEU1aDhLRXFpWGFtNGU4RlQ1RDFvbGxWdUEKcDRtRmJqMVN0SkNSVWM3K2o0a25mT1JHVX
pLZjdMTFBqL1M3VzE4NFhxVUZHcHNvQVUvczlPaWpVd1Nnd1MxYwpsQUwra3VjbFpxRG5CNmpLKzFnREVXbmdTcmUzYXY2K240V1c4YURDaTY
ySTNPVUE2UDdwNmhDMStGWGRPdEI0ClVaZDFBVVRZQU1KdHNveHJiS2hnOGdIN3ZpOEJOUWZnV1RabmdzcFYzOEdjSEhCME96bDVJUVAwOHhF
N1NMUXAKQVdIU2U5OWFoN0lTCi0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K"
cluster_endpoint = "https://5BB8C62322E206B71C17C0EA7E0CC9F1.yl4.us-west-2.eks.amazonaws.com"
cluster_iam_role_name = "cloudops-demo-cluster-20250622132054187300000001"
cluster_name = "cloudops-demo"
cluster_oidc_issuer_url = "https://oidc.eks.us-west-2.amazonaws.com/id/5BB8C62322E206B71C17C0EA7E0CC9F1"
cluster_security_group_id = "sg-0c839e72a573f26e9"
codebuild_role_arn = "arn:aws:iam::715841362372:role/cloudops-demo-codebuild-role"
ecr_repositories = {
  "cloudops-backend" = "715841362372.dkr.ecr.us-west-2.amazonaws.com/cloudops-backend"
  "cloudops-frontend" = "715841362372.dkr.ecr.us-west-2.amazonaws.com/cloudops-frontend"
}
kubectl_config_command = "aws eks update-kubeconfig --region us-west-2 --name cloudops-demo"
```

```
1510    "subnet-0dd645deccd9a82de",
1511    "subnet-0e8ca804775773871",
1512  ]
1513  public_subnets = [
1514    "subnet-014b6f6363fb76c27",
1515    "subnet-09ae78ecd9cf0d0a9",
1516    "subnet-07e74975dbb74e238",
1517  ]
1518  rds_database_name = "cloudops_demo"
1519  rds_endpoint = <sensitive>
1520  rds_port = 5432
1521  vpc_id = "vpc-098d107ac01bf793d"
1522  === Setting up kubectl ===
1523  Added new context arn:aws:eks:us-west-2:715841362372:cluster/cloudops-demo to /root/.kube/config
1524  NAME                                          STATUS   ROLES    AGE   VERSION
1525  ip-10-0-24-188.us-west-2.compute.internal     Ready    <none>   32m   v1.27.16-eks-aeac579
1526  ip-10-0-40-222.us-west-2.compute.internal     Ready    <none>   32m   v1.27.16-eks-aeac579
1527  ip-10-0-8-49.us-west-2.compute.internal       Ready    <none>   32m   v1.27.16-eks-aeac579
1528
1529  [Container] 2025/06/22 14:05:23.039594 Phase complete: POST_BUILD State: SUCCEEDED
1530  [Container] 2025/06/22 14:05:23.039616 Phase context status code:  Message:
1531  [Container] 2025/06/22 14:05:23.361846 Set report auto-discover timeout to 5 seconds
1532  [Container] 2025/06/22 14:05:23.361901 Expanding base directory path:  .
1533  [Container] 2025/06/22 14:05:23.365651 Assembling file list
1534  [Container] 2025/06/22 14:05:23.365665 Expanding .
1535  [Container] 2025/06/22 14:05:23.368903 Expanding file paths for base directory .
1536  [Container] 2025/06/22 14:05:23.368926 Assembling file list
1537  [Container] 2025/06/22 14:05:23.368929 Expanding **/*
1538  [Container] 2025/06/22 14:05:23.428871 No matching auto-discover report paths found
```

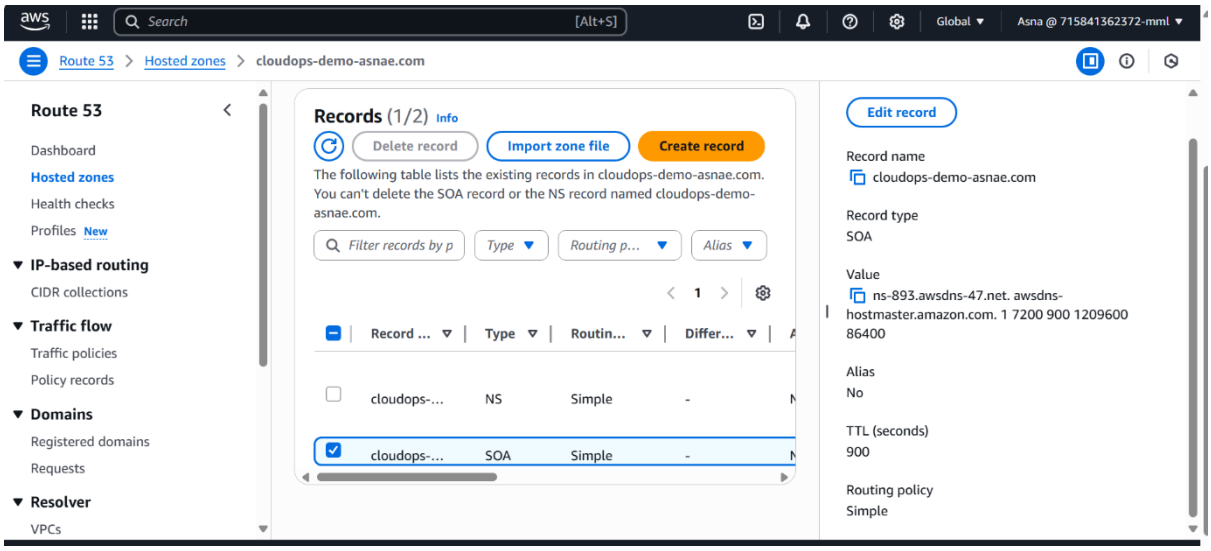
After creating the infrastructure deployed the application in eks using another pipeline

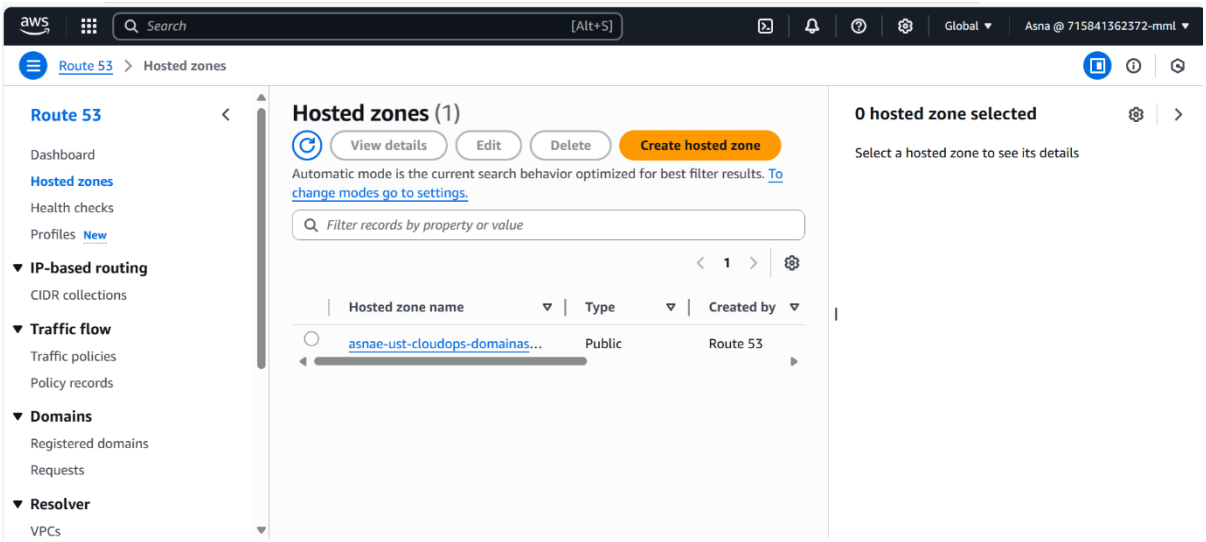


After successful deployment ,accessed the application,showed it is working fine

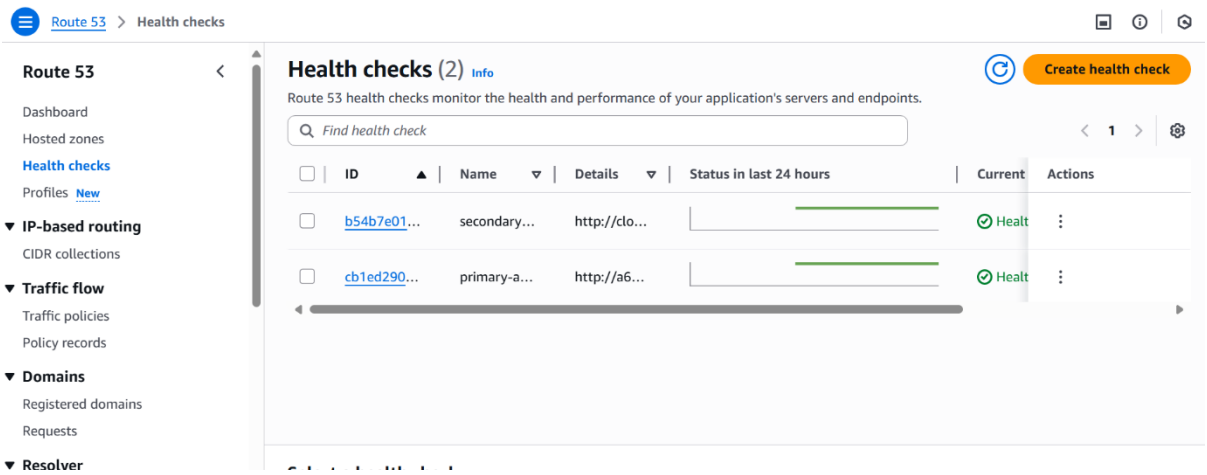


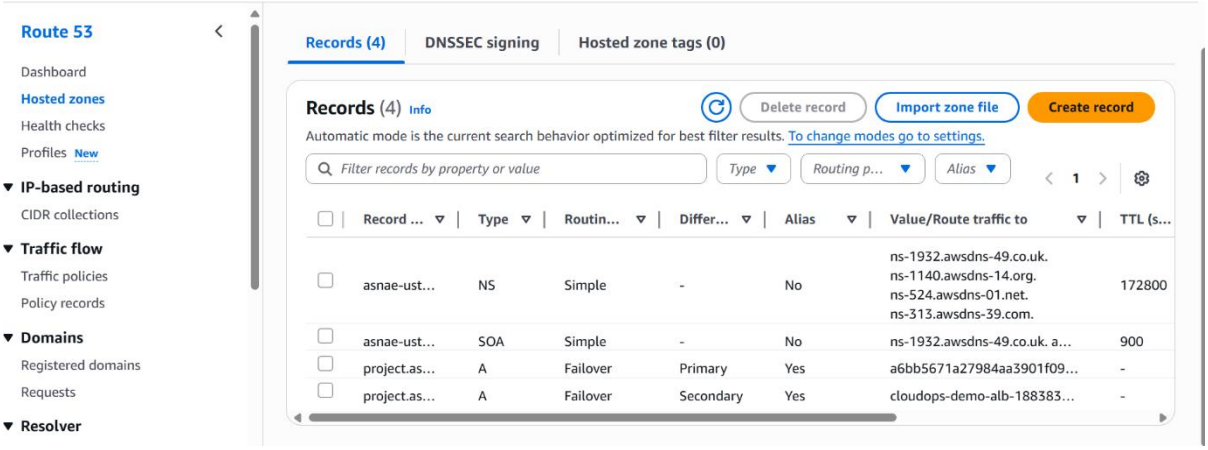
Phase4:- Route53





Created Health checks





Brought a domain from go daddy which is “project.asnae-ust-cloudops-domainasna.xyz”

Route 53 hosted zone, create an Alias record (A type) for application

Alias record should point directly to the DNS name of the AWS Application Load Balancer that is fronting Kubernetes Ingress.

Phase5:- Cloudwatch Monitoring