- Digipay
    - A Blockchain-Based Wallet Website
    - Features
    - Technology Stack
    - Version 1:
    - Deployed in render
    - Network Deployment Summary
    - Version 2:
        - Integration with Superfluid is done!.
        - Demo Video Link (v2):
    - Version 3:
    - Devops (Deploying and Managing Services):
    - Autoscaling and Loadbalancer (accessing private subnets (where we have our servers)):
        - 1. Creating a VPC (Virtual Private Cloud) where we configure 2 availability zones, 2 public subnets, 2 private subnets and 1 NAT Gateway per availability zone. (Size of the VPC: 65536 ips)
        - 2. Creating Autoscaling group in the VPC we created above
        - 3. Now, Let's create a Loadbalancer and map it to two public subnets in the VPC
        - 4. Now, we can see that I implemented the full architecture as shown in architecture image:
        - Remember, I am running the server in one private instance only, the other is for now unhealthy in target group
        - Let's Try using someother python basic server in other instance and check what happens

Digipay

A Blockchain-Based Wallet Website

This project aims to introduce users to blockchain and Web3 functionalities by integrating with MetaMask, allowing transactions via a custom website wallet, and utilizing Superfluid for stream transactions. It features a comprehensive user system with registration and login capabilities, transaction history, and the generation and use of test tokens for transactions.

Features

1. **MetaMask Integration**: Prompt users to connect their MetaMask wallet upon visiting the website.
2. **Display Account Details**: Show the connected MetaMask account's address and balance.
3. **Website Wallet Functionality**: Enable users to send cryptocurrency from their MetaMask account to another recipient's address. These transactions are recorded on the blockchain.
4. **Transaction History**: Display a history of all user transactions made through the website.
5. **User Authentication System**: Include a registration and login system, with MongoDB or MySQL as the database backend. Unique email addresses are used for registration. Transactions can be filtered by the user's email address (stored in the blockchain transaction in a secure manner).
6. **Superfluid Integration**: Implement a separate wallet feature using Superfluid for creating, updating, and deleting money flows.
7. **Test Token Generation**: Allow users to generate test tokens for use in transaction testing.
8. **Educational Content**: Provide users with information on blockchain and Web3 to educate them on the technology's uses and implications.
9. **Superfluid Wallet Functions**: Detailed functionalities for creating, updating, and deleting flows with Superfluid.

Technology Stack

- **Frontend**: React, Vite
- **Blockchain Interaction**: Web3.js or Ethers.js, Solidity for smart contracts
- **Backend**: Node.js, Express.js for server-side logic
- **Database**: MongoDB or MySQL
- **Blockchain Technologies**: MetaMask, Superfluid
- **Testing**: Hardhat or Truffle for smart contract testing

Version 1:

This Web Application is currently limited to interacting with MATIC because the contract is designed for operations with the native token, and the RPC URL connects to the Mumbai network where MATIC is the native currency.

Deployed in render

https://digipay-app.onrender.com

Network Deployment Summary

- **Mumbai**:
  - TransactionRecorder deployed to: `contract_address`
- **Polygon Mainnet**:
  - Insufficient funds in my wallet account provided to deploy.
- **Ethereum Mainnet**:
  - Insufficient funds in my wallet account provided to deploy.
- **Goerli Testnet**:
  - Insufficient funds in my wallet account provided to deploy.
  - Note: To receive Goerli ETH, your wallet must have a minimum balance of 0.001 ETH on mainnet. Please add ETH to your wallet and try again.
- **Sepolia Testnet**:
  - Taking so much time and giving network errors while adding Sepolia ETH to wallet (not added). So, while deploying the contract for Sepolia RPC, it showed insufficient funds.
- **Linea Mainnet**:
  - Insufficient funds in my wallet account provided to deploy.

For now, I am able to get Mumbai faucet MATIC tokens only, and those I am using now in this application with a contract deployed on the Mumbai network. These tokens are also limited to get sufficient gas amount; if our account already has that much amount, then it won’t give more.

Version 2:

Integration with Superfluid is done!.

- You can use the same link: https://digipay-app.onrender.com
- Now users can create streams, update streams, delete streams
- **Note to Users:** For streaming transactions in Superfluid, ensure you use Super Tokens, typically ending with 'x' (e.g., fDAIx). These tokens are "wrapped" versions of regular tokens, designed for fluid, continuous transactions. To convert regular tokens into Super Tokens or vice versa, visit Superfluid App, where you can easily wrap and unwrap tokens for use in streams. Always confirm you're interacting with the correct token for seamless transactions.

Demo Video Link (v2):

- Click Here

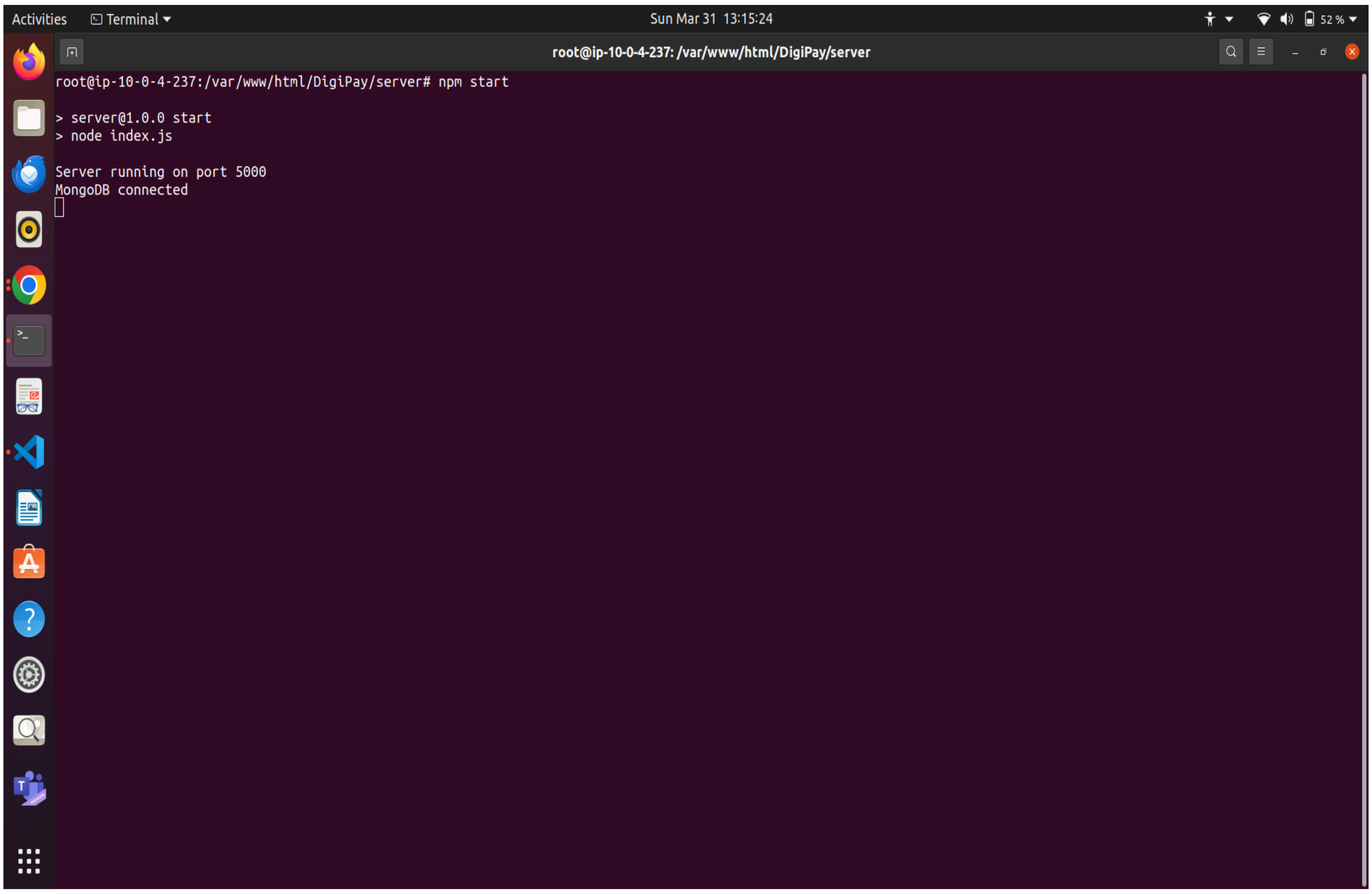
Version 3:

- Deployed a contract of ERC20 tokens DIGI tokens where a user of Digi App can claim those in the Application for usage, testing purpose.
- An account can claim only 100 tokens per week and the account should also have less than 100 tokens if they are claiming.
- Future Implementations: Want to make transactions done using our Digi Tokens in our application. And want to implement a wrapping function to interact with superfluid with these tokens only.

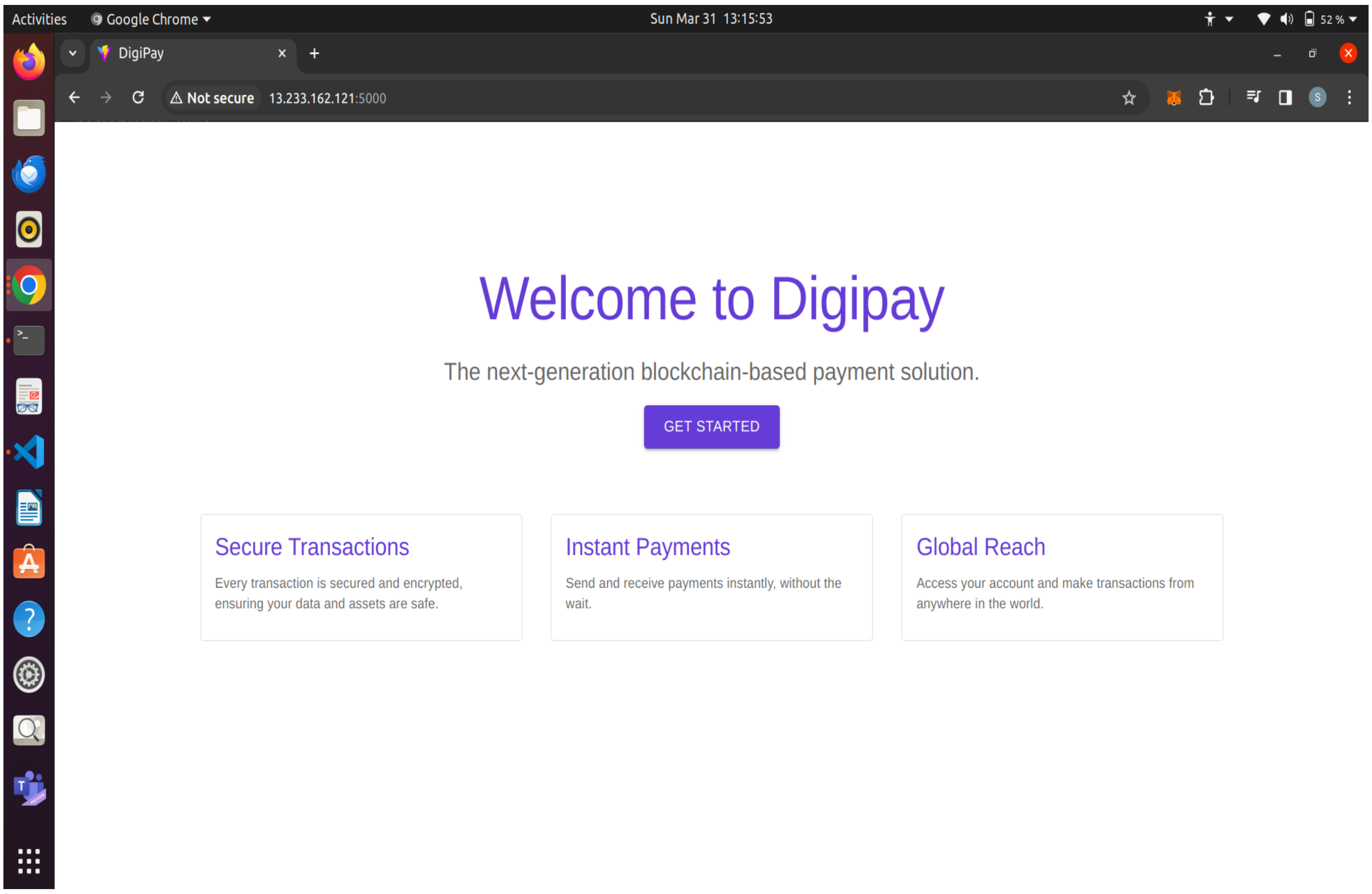
Devops (Deploying and Managing

Services):

- I deployed this DigiPay application in an AWS EC2 Instance where we can access something like here: http://`<instance-public-address>`:`<port>`/
- At first for subnet in vpc, NACL is by default allowing all traffic, but security group in instance is allowing ssh port only using which we can login in terminal to that instance and run our applications in ports we want (for eg: I run DigiPay in port 5000), and as port 5000 is allowed by nacl now but not yet by security groups. So, it was not opening and after editing inbound traffic where to allow custom tcp port 5000 from anywhere (ipv4). The application started running then.
- **Note:** The link might not work when i stopped running in that instance, and ip address changes whenever we start it after stopping it for sometime.



```
root@ip-10-0-4-237:/var/www/html/DigiPay/server# npm start

> server@1.0.0 start
> node index.js

Server running on port 5000
MongoDB connected
```



Welcome to Digipay

The next-generation blockchain-based payment solution.

GET STARTED

Secure Transactions

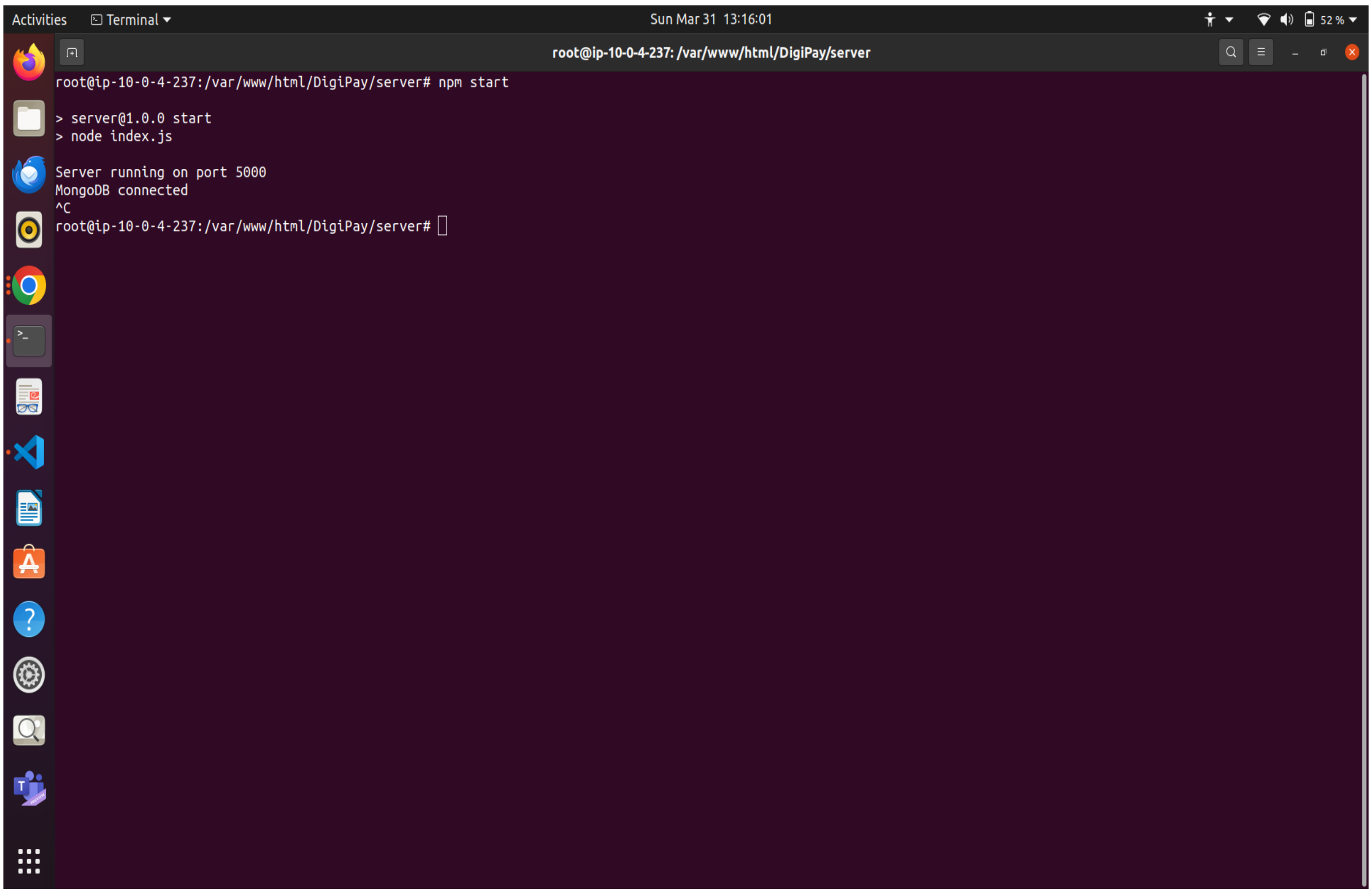
Every transaction is secured and encrypted, ensuring your data and assets are safe.

Instant Payments

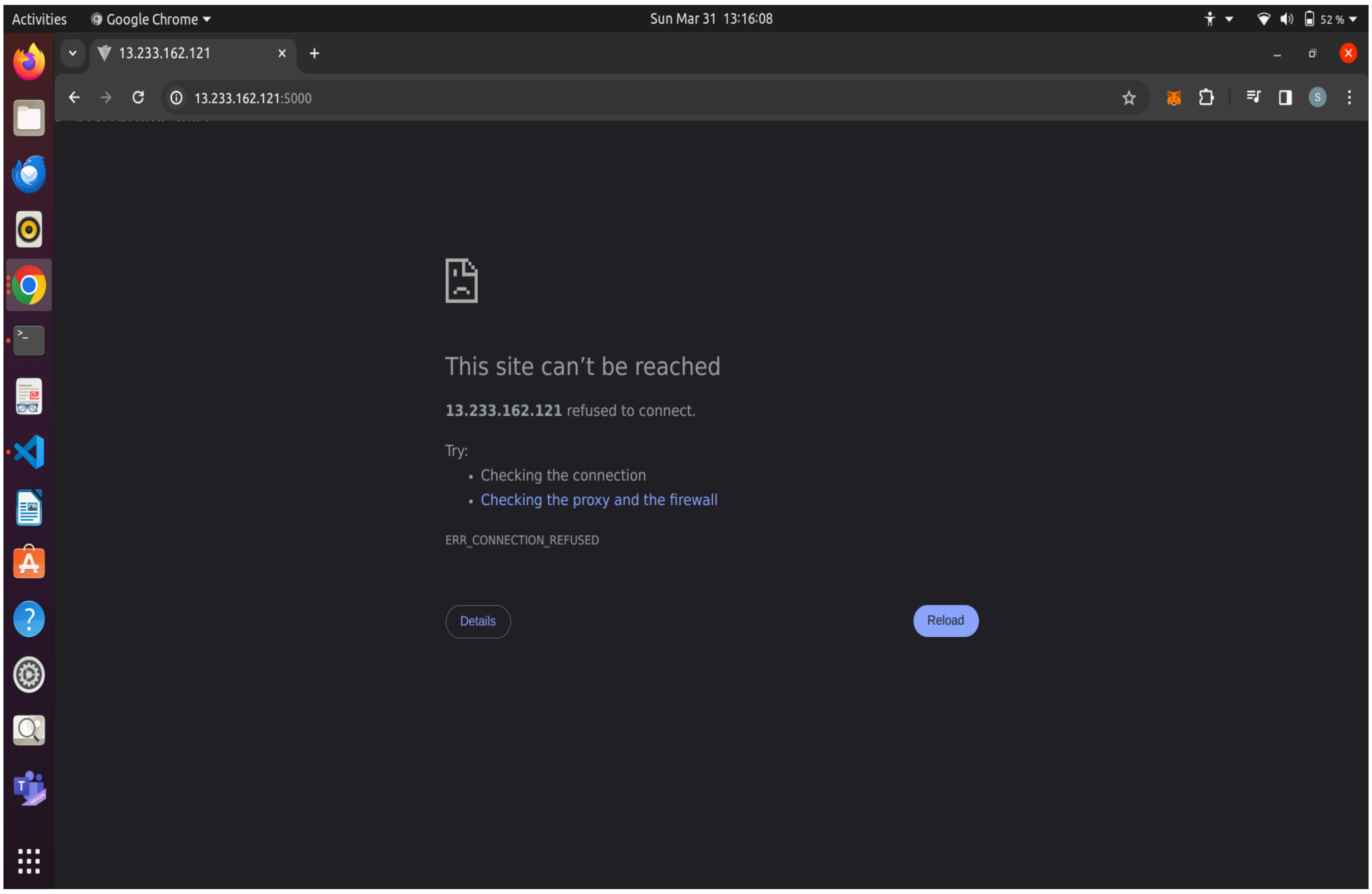
Send and receive payments instantly, without the wait.

Global Reach

Access your account and make transactions from anywhere in the world.



```
root@ip-10-0-4-237:/var/www/html/DigiPay/server# npm start

> server@1.0.0 start
> node index.js

Server running on port 5000
MongoDB connected
^C
root@ip-10-0-4-237:/var/www/html/DigiPay/server#
```



This site can’t be reached

13.233.162.121 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

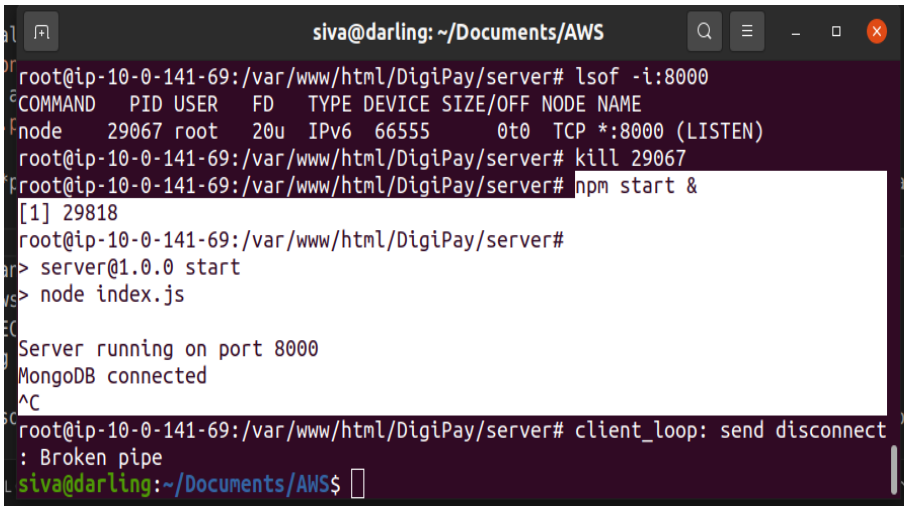
ERR_CONNECTION_REFUSED

Figure 1: Server Running in EC2 Instance on Port 5000

Figure 2: Application Running in Browser on http://`<instance-public-ip-address>`:5000

Figure 3: Server stopped Running in EC2 Instance

Figure 4: Application stopped Running in Browser

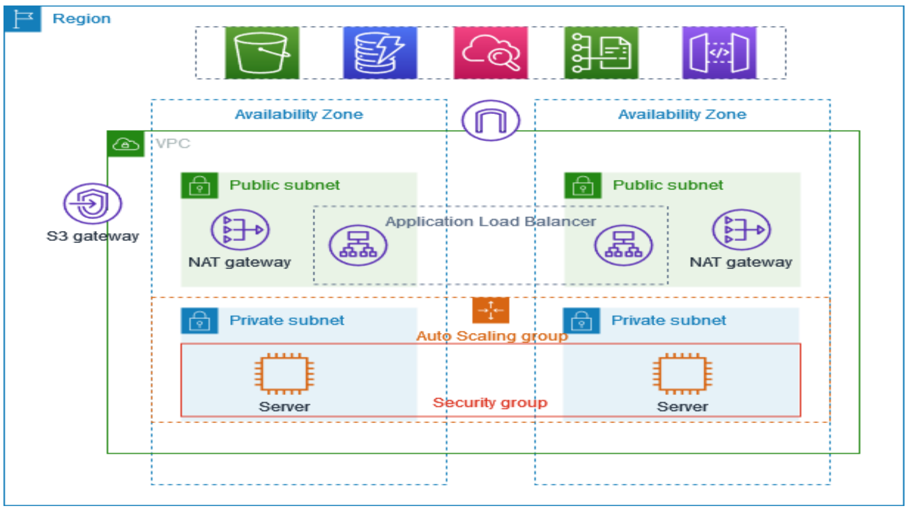


```
siva@darling: ~/Documents/AWS
root@ip-10-0-141-69:/var/www/html/DigiPay/server# lsof -i:8000
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
node    29067 root   20u  IPv6  66555      0t0  TCP *:8000 (LISTEN)
root@ip-10-0-141-69:/var/www/html/DigiPay/server# kill 29067
root@ip-10-0-141-69:/var/www/html/DigiPay/server# npm start &
[1] 29818
root@ip-10-0-141-69:/var/www/html/DigiPay/server#
> server@1.0.0 start
> node index.js

Server running on port 8000
MongoDB connected
^C
root@ip-10-0-141-69:/var/www/html/DigiPay/server# client_loop: send disconnect
: Broken pipe
siva@darling:~/Documents/AWS$
```

Figure 5: If we can run the server in background **placing & after npm start (process running in bacground)**, then even if you terminate there in terminal, the application still runs.

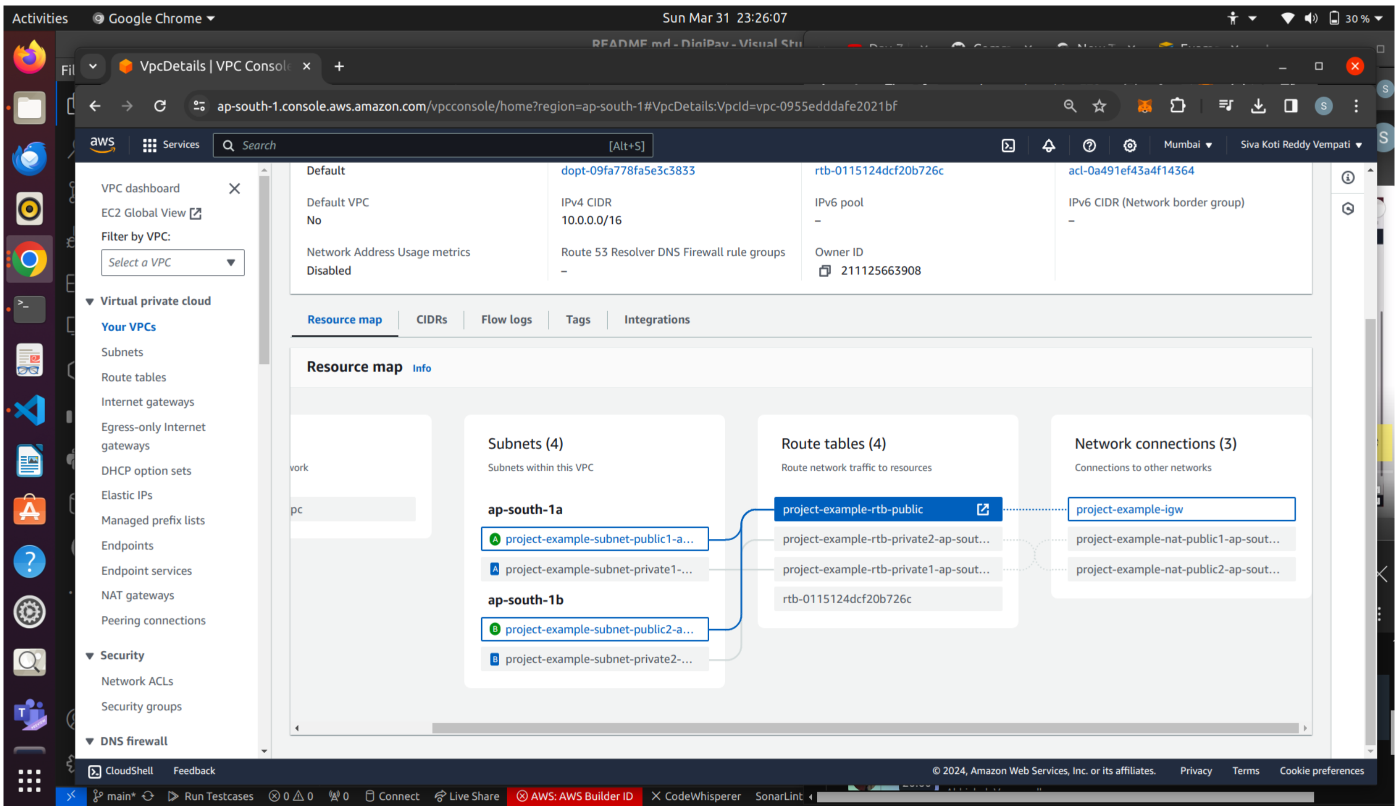
- One problem can be only if I stopped running that instance and if I try to start it again the IP iddress changes and so the link changes.

Autoscaling and Loadbalancer (accessing private subnets (where we have our servers)):

- **Here is the reference architecture diagram, the same I am implementing:** (Ignore S3 Gateway)
- 

1. Creating a VPC (Virtual Private Cloud) where we configure 2 availability zones, 2 public subnets, 2 private subnets and 1 NAT Gateway per availability zone. (Size of the VPC: 65536 ips)

- With this VPC, we created some basic layout as shown in the above architecture for now and will create autoscaling group and load balancer further



Default | dopt-09fa778fa5e3c3833 | rtb-0115124dcf20b726c | acl-0a491ef43a4f14364

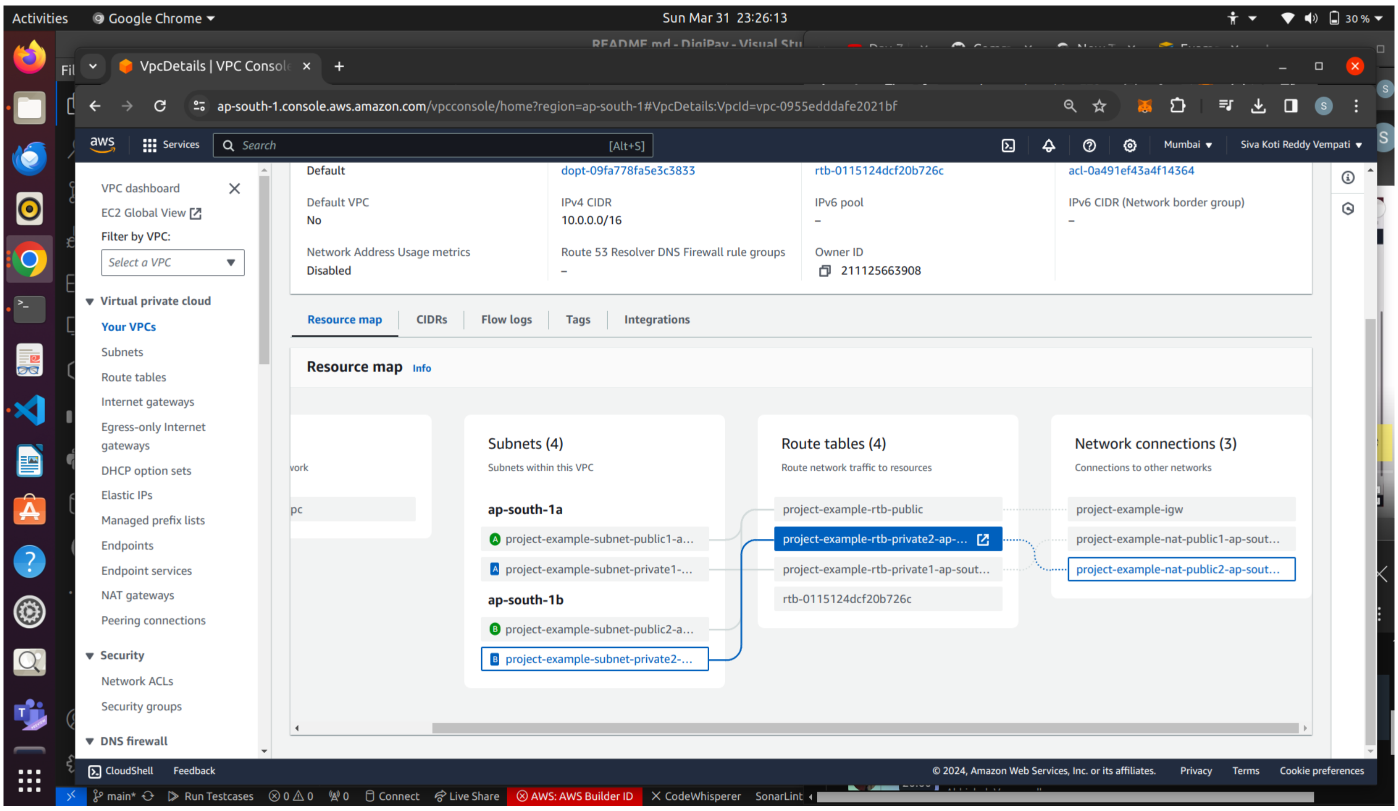
Default VPC: No
IPv4 CIDR: 10.0.0.0/16
IPv6 pool: –
IPv6 CIDR (Network border group): –
Network Address Usage metrics: Disabled
Route 53 Resolver DNS Firewall rule groups: –
Owner ID: 211125663908

Subnets (4)
Subnets within this VPC
ap-south-1a
project-example-subnet-public1-a...
project-example-subnet-private1-...
ap-south-1b
project-example-subnet-public2-a...
project-example-subnet-private2-...

Route tables (4)
Route network traffic to resources
project-example-rtb-public
project-example-rtb-private2-ap-sout...
project-example-rtb-private1-ap-sout...
rtb-0115124dcf20b726c

Network connections (3)
Connections to other networks
project-example-igw
project-example-nat-public1-ap-sout...
project-example-nat-public2-ap-sout...

Public subnets attached to the route table and the internet Gateway



Default | dopt-09fa778fa5e3c3833 | rtb-0115124dcf20b726c | acl-0a491ef43a4f14364

Default VPC: No
IPv4 CIDR: 10.0.0.0/16
IPv6 pool: –
IPv6 CIDR (Network border group): –
Network Address Usage metrics: Disabled
Route 53 Resolver DNS Firewall rule groups: –
Owner ID: 211125663908

Resource map

Subnets (4)
Subnets within this VPC
ap-south-1a
project-example-subnet-public1-a...
project-example-subnet-private1-...
ap-south-1b
project-example-subnet-public2-a...
project-example-subnet-private2-...

Route tables (4)
Route network traffic to resources
project-example-rtb-public
project-example-rtb-private2-ap-...
project-example-rtb-private1-ap-sout...
rtb-0115124dcf20b726c

Network connections (3)
Connections to other networks
project-example-igw
project-example-nat-public1-ap-sout...
project-example-nat-public2-ap-sout...

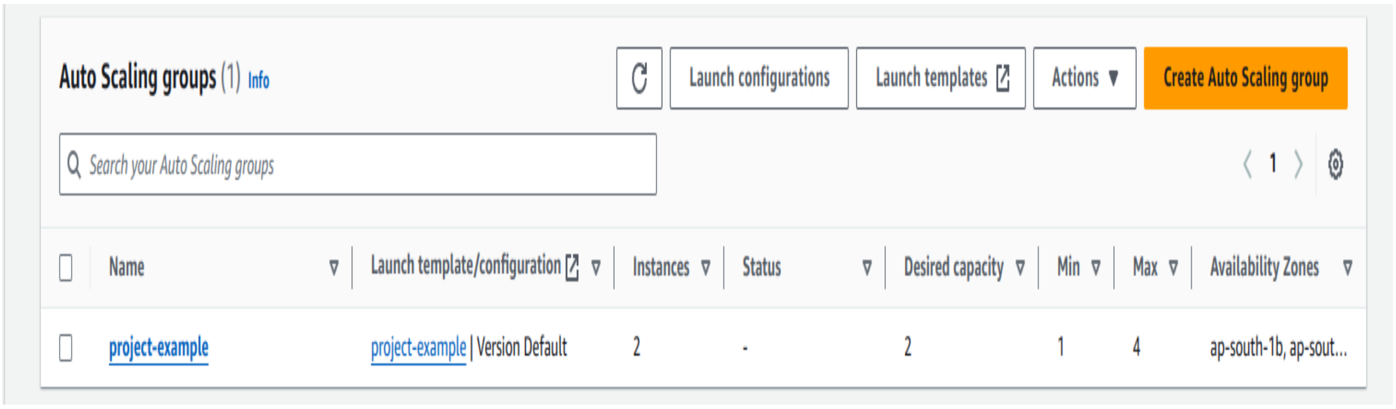
Private subnets attached to different route tables and to NAT Gateways in public subnets

Now, the VPC created subnets, internet gateway, NAT gateways, route tables, routes and associated elastic ip addresses, route tables and verified

2. Creating Autoscaling group in the VPC we created above

- First Launch a template where we will configure what all ports to allow to the instances we create, and in which vpc, we want to launch this autoscaling.
- After creating template, while creating autoscaling group, I configured that I want 2 ec2 instances to be in 2 private subnets.
- I am not attaching any load balancer now, I want to create it later in the public subnets.
- Now, as the name suggests, autoscaling, that means it can dynamically create more instances if there is so much traffic incoming and can remove instances if there is less traffic.
- So, created a desired capacity of 2 instances (to start with 2 instances). And configured maximum instances it can go in peak time are 4, and minimum it can go is 1.

- And, now as we created autoscaling group, we can check whether it created two instances for us or not in two private subnets (as my region ap-south-1, one instance i need in private ap-south-1a, other in private ap-south-1b as configured earlier in autoscaling creation).



Auto Scaling groups (1) Info

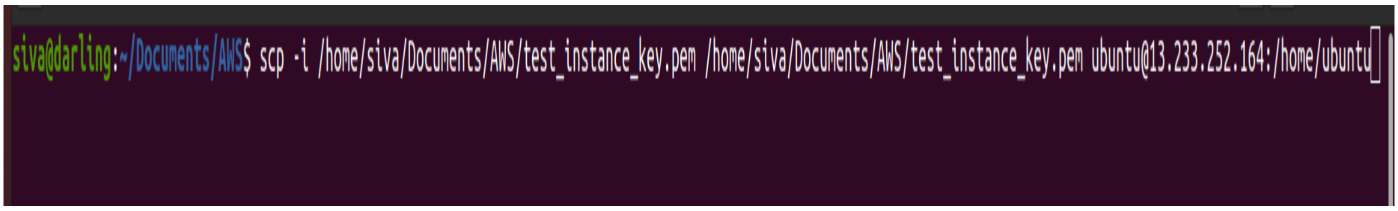
	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
☐	project-example	project-example | Version Default	2	-	2	1	4	ap-south-1b, ap-sout...

Now the question comes, As we don't have public ipv4 addresses to these instances as we created those in private subnets, How do I login into those instances as I want to create my servers there only

- Now, I created a bastion-host instance which can act as mediator between public and private subnets
- So, Inside the VPC, I created this bastion host giving permission to allow ssh port (where we login) (enabled auto-assign public ip address, as we have to login using that)
- As, I launched this host instance, I will first login to this host instance and from there I login to private instance.

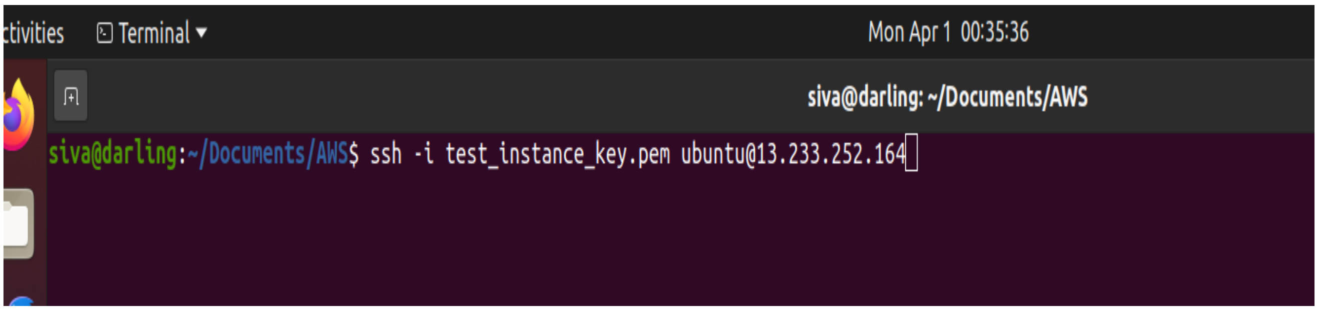
But to login, I need encrypted key right?, I have that in locally to login to host instance, but from there how to login to private instance without this key?

So, What I did is copied the key from local machine to host instance using scp (secure copy protocol)

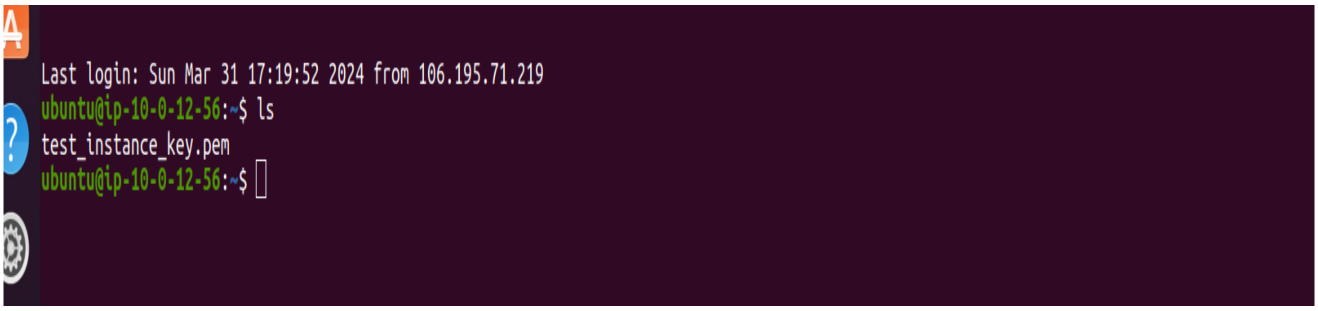


```
siva@darling:~/Documents/AWS$ scp -i /home/siva/Documents/AWS/test_instance_key.pem /home/siva/Documents/AWS/test_instance_key.pem ubuntu@13.233.252.164:/home/ubuntu
```

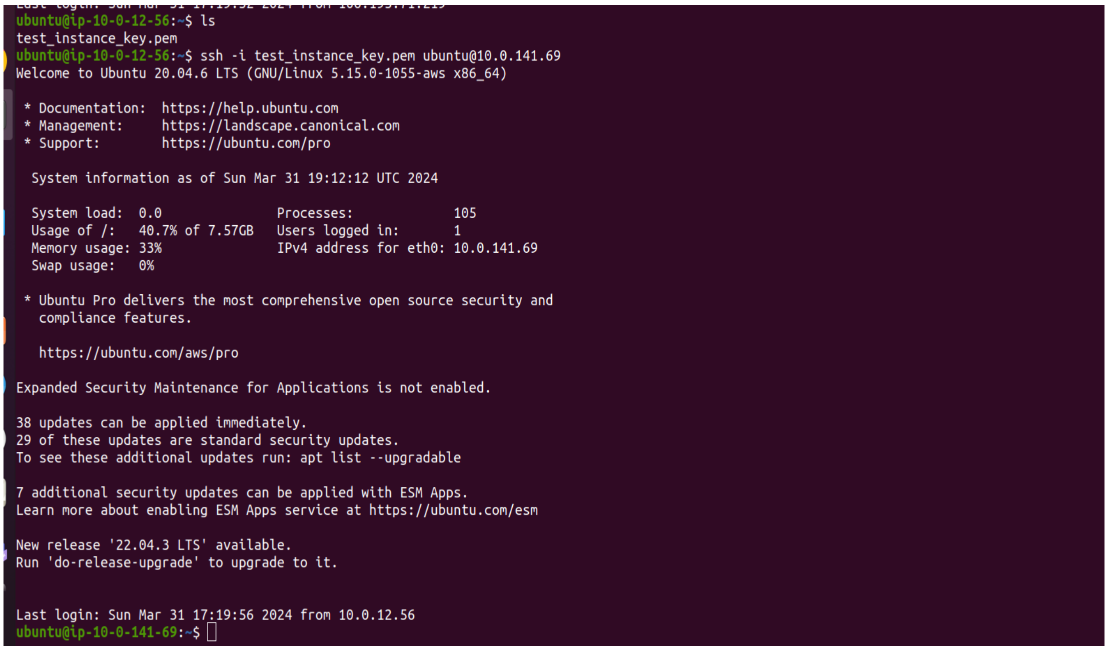
- Now, I logged into host using it's public ipv4 address

- 

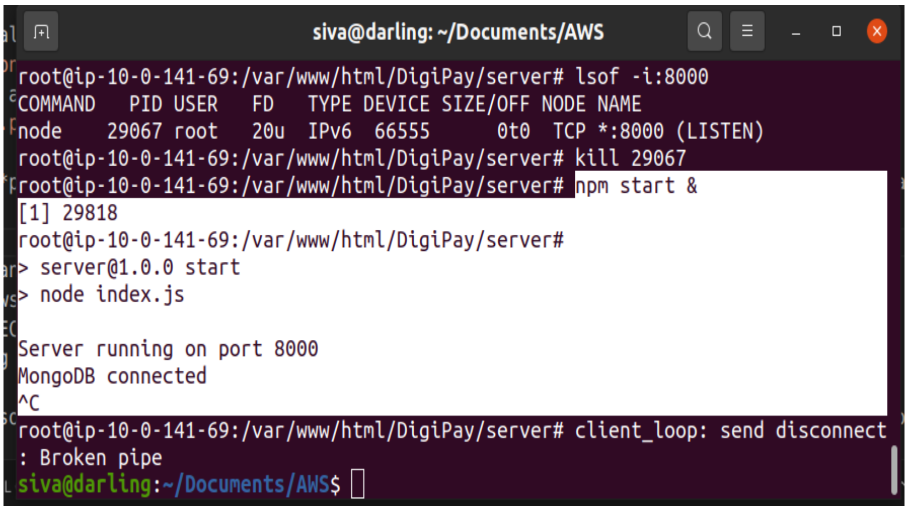
```
siva@darling:~/Documents/AWS$ ssh -i test_instance_key.pem ubuntu@13.233.252.164
```

- See now, As I copied, I can see the key here
- 

```
Last login: Sun Mar 31 17:19:52 2024 from 106.195.71.219
ubuntu@ip-10-0-12-56:~$ ls
test_instance_key.pem
ubuntu@ip-10-0-12-56:~$
```

- And I use this key to login to private instance (instance which is in private subnet) using private ip address
- 

```
ubuntu@ip-10-0-12-56:~$ ls
test_instance_key.pem
ubuntu@ip-10-0-12-56:~$ ssh -i test_instance_key.pem ubuntu@10.0.141.69
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1055-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Sun Mar 31 19:12:12 UTC 2024

  System load:  0.0               Processes:             105
  Usage of /:   40.7% of 7.57GB   Users logged in:       1
  Memory usage: 33%               IPv4 address for eth0: 10.0.141.69
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

38 updates can be applied immediately.
29 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sun Mar 31 17:19:56 2024 from 10.0.12.56
ubuntu@ip-10-0-141-69:~$
```

- And I ran my server inside that private instance
- 

```
root@ip-10-0-141-69:/var/www/html/DigiPay/server# lsof -i:8000
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
node    29067 root   20u  IPv6  66555      0t0  TCP *:8000 (LISTEN)
root@ip-10-0-141-69:/var/www/html/DigiPay/server# kill 29067
root@ip-10-0-141-69:/var/www/html/DigiPay/server# npm start &
[1] 29818
root@ip-10-0-141-69:/var/www/html/DigiPay/server#
> server@1.0.0 start
> node index.js

Server running on port 8000
MongoDB connected
^C
root@ip-10-0-141-69:/var/www/html/DigiPay/server# client_loop: send disconnect
: Broken pipe
siva@darling:~/Documents/AWS$
```

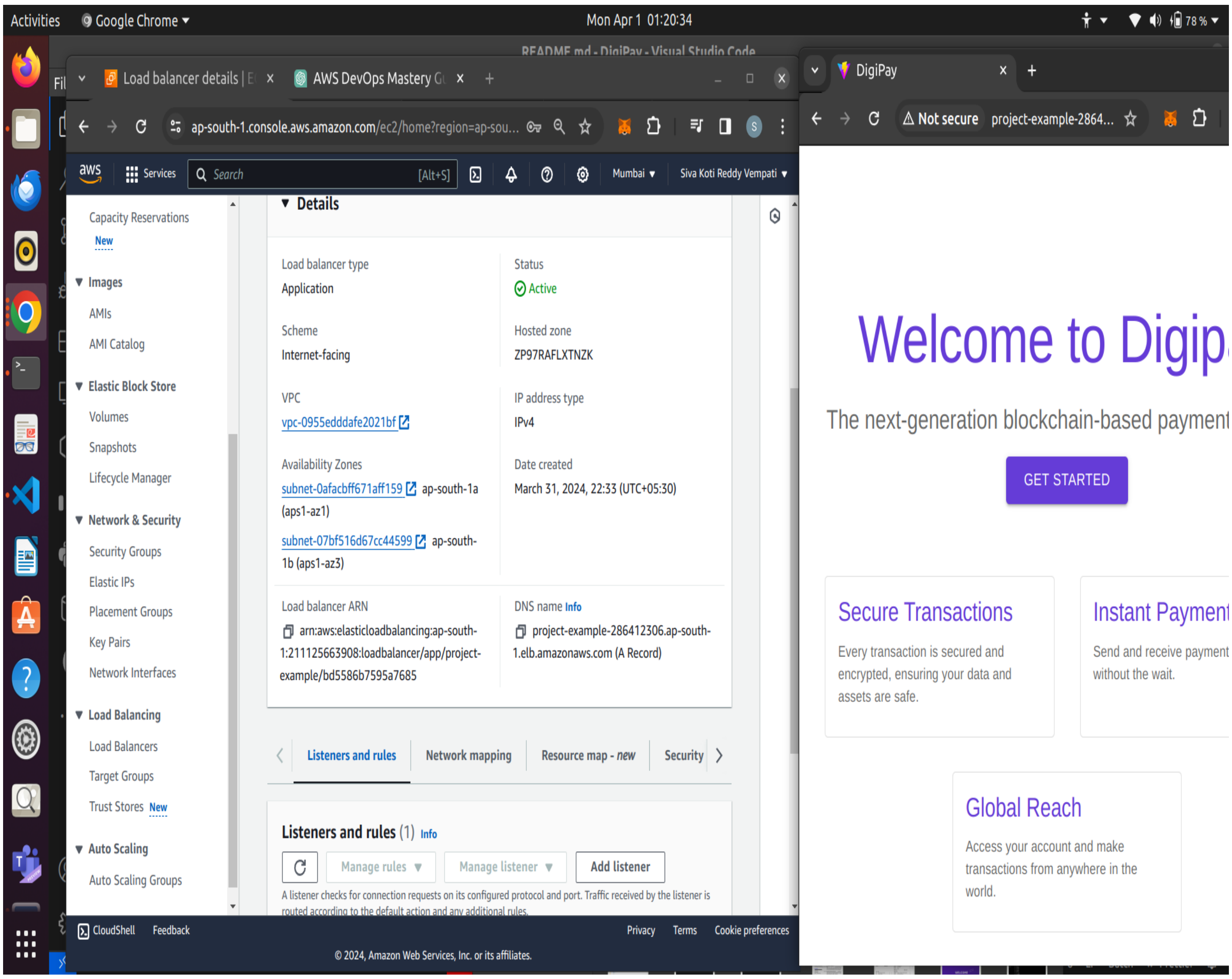
- For now, I am running the server only in 1 private instance and there is no server in the other instance.

3. Now, Let's create a Loadbalancer and map it to two public subnets in the VPC

- Created an Application Loadbalancer (Layer 7 Loadbalancer), configured vpc, security group (for allowing specific traffic).
- I added a security group which allows ssh port 22, and http port 8000, where I am running server in that port 8000.
- Now, created a target group with specific instances we select to support loadbalancing (using http port: 8000, as I allowed already).
- Now, Create Loadbalancer by mapping to both public subnets and attach target group we created, and let's leave the accessible port of loadbalancer as 80 (HTTP port 80) for now.
- And the load balancer is created, but it shows that port is not accessible as we didn't allow http port 80 in any security group yet.
- Now change the configuration of security group of loadbalancer to allow traffic from http port 80, then that error is gone :)

4. Now, we can see that I implemented the full architecture as shown in architecture image:

- Abstraction: The DNS name abstracts the IP addresses of the individual servers behind the load balancer. Users or client applications don't need to know about the specific servers or their IP addresses; they only need to know the DNS name.
- Now, we use the DNS name of our loadbalancer to access the server.



Details

Load balancer type
Application

Status
Active

Scheme
Internet-facing

Hosted zone
ZP97RAFLXTNZK

VPC
vpc-0955edddafe2021bf

IP address type
IPv4

Availability Zones
subnet-0afacbff671aff159 ap-south-1a (aps1-az1)
subnet-07bf516d67cc44599 ap-south-1b (aps1-az3)

Date created
March 31, 2024, 22:33 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:ap-south-1:211125663908:loadbalancer/app/project-example/bd5586b7595a7685

DNS name Info
project-example-286412306.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules | Network mapping | Resource map - new | Security

Listeners and rules (1) Info

Manage rules | Manage listener | Add listener

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

© 2024, Amazon Web Services, Inc. or its affiliates.

Welcome to Digip

The next-generation blockchain-based payment

GET STARTED

Secure Transactions

Every transaction is secured and encrypted, ensuring your data and assets are safe.

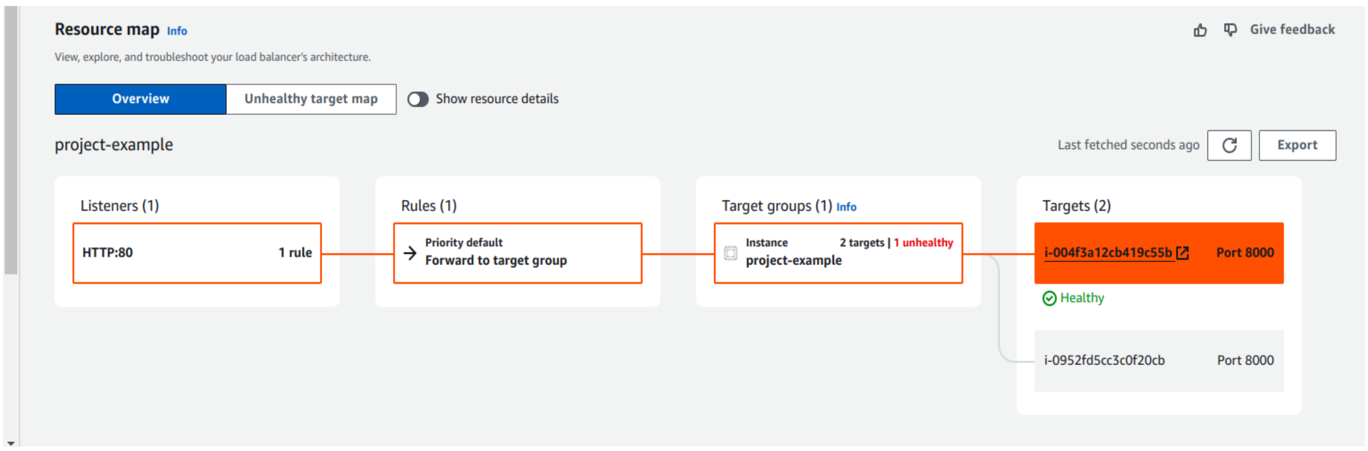
Instant Payment

Send and receive payment without the wait.

Global Reach

Access your account and make transactions from anywhere in the world.

Remember, I am running the server in one private instance only, the other is for now unhealthy in target group



Resource map Info

View, explore, and troubleshoot your load balancer's architecture.

project-example

Last fetched seconds ago

Listeners (1)
HTTP:80 1 rule

Rules (1)
Priority default
Forward to target group

Target groups (1) Info
Instance 2 targets | 1 unhealthy
project-example

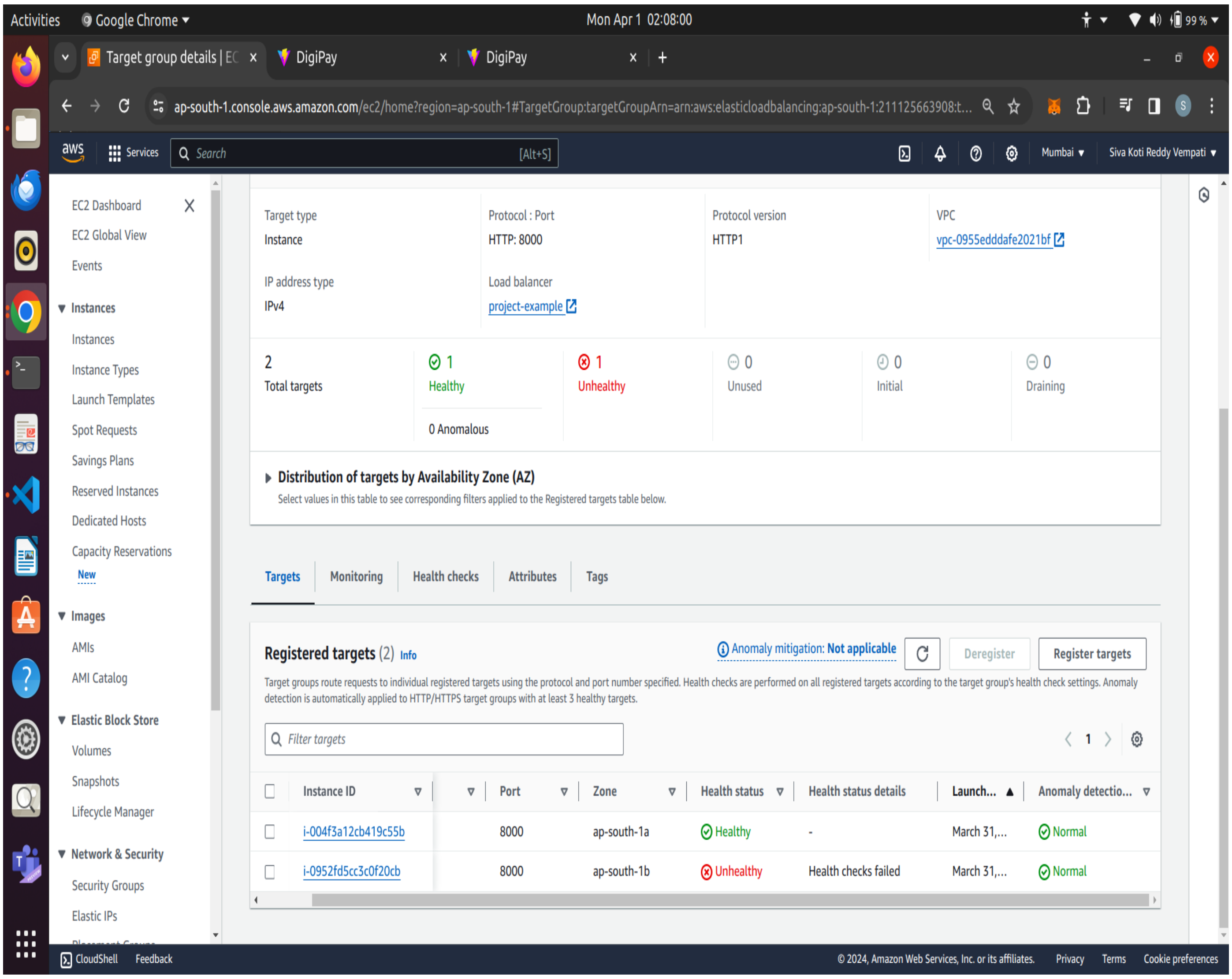
Targets (2)
i-004f3a12cb419c55b Port 8000
Healthy
i-0952fd5cc3c0f20cb Port 8000

- Target group is doing health check and it is only forwarding the request to healthy loadbalancer.
- If we disable that health check, then it will send the server request to both instances. So, sometimes we get error and sometimes, we get the application.

Let's Try using someother python basic server in other instance and check what happens

- The DNS Name Link in Loadbalancer using which one can access the servers and through which we send requests for accessing servers: http://project-example-286412306.ap-south-1.elb.amazonaws.com/

As mentioned, the health check fails in target group and always sends the request to the healthy instance



Activities Google Chrome Mon Apr 1 02:08:00

Target group details | EC DigiPay DigiPay

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:211125663908:t...

aws Services Search [Alt+S] Mumbai Siva Koti Reddy Vempati

EC2 Dashboard
EC2 Global View
Events
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
▼ Images
AMIs
AMI Catalog
▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
▼ Network & Security
Security Groups
Elastic IPs

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 8000	HTTP1	vpc-0955edddafe2021bf
IP address type	Load balancer		
IPv4	project-example		

2	1	1	0	0	0
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
	0 Anomalous				

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets Monitoring Health checks Attributes Tags

Registered targets (2) Info   Anomaly mitigation: Not applicable   Deregister   Register targets

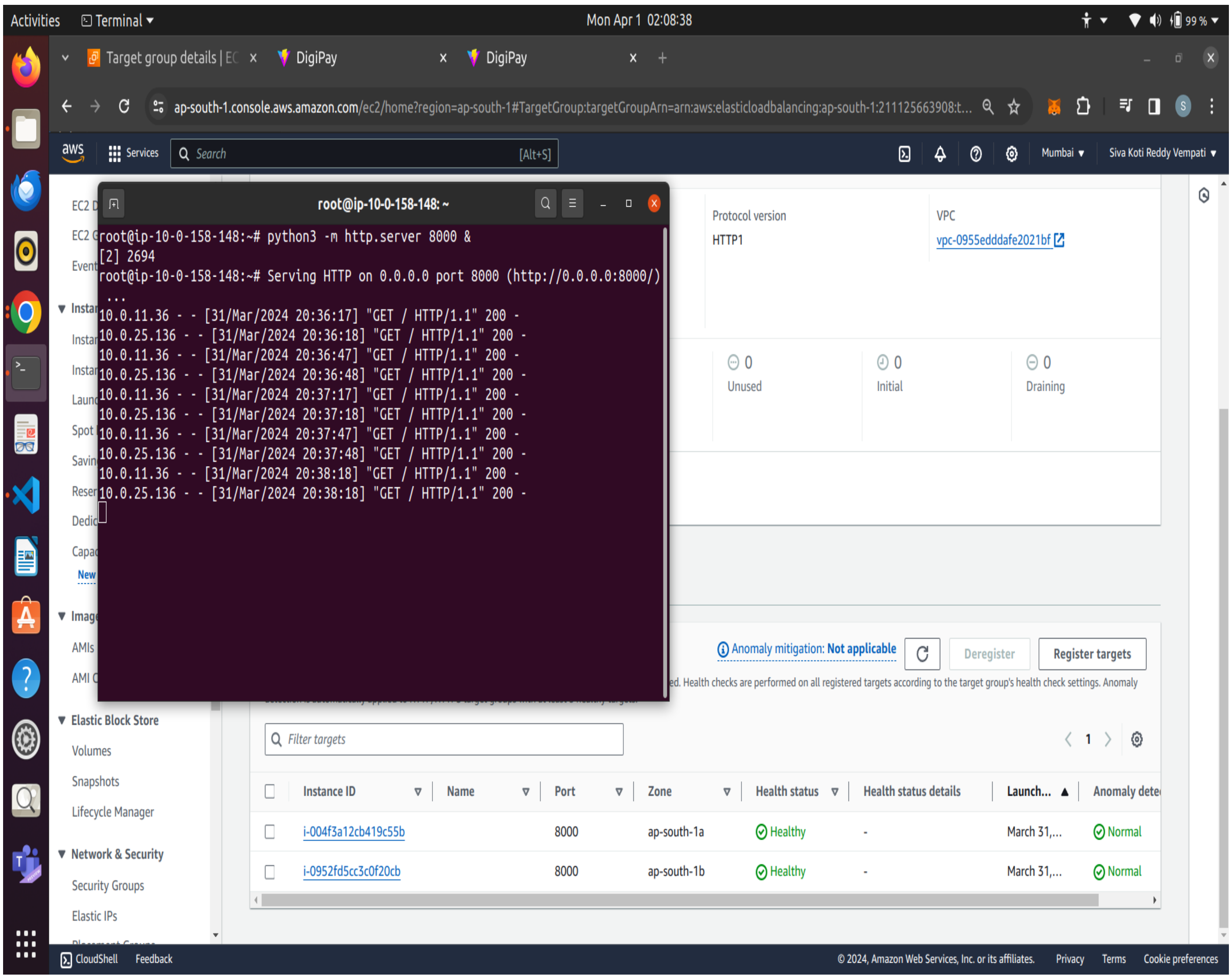
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

Instance ID		Port	Zone	Health status	Health status details	Launch...	Anomaly detectio...
i-004f3a12cb419c55b		8000	ap-south-1a	Healthy	-	March 31,...	Normal
i-0952fd5cc3c0f20cb		8000	ap-south-1b	Unhealthy	Health checks failed	March 31,...	Normal

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now, As I logged into the 2nd private instance and ran a simple html page using python3 in server port 8000

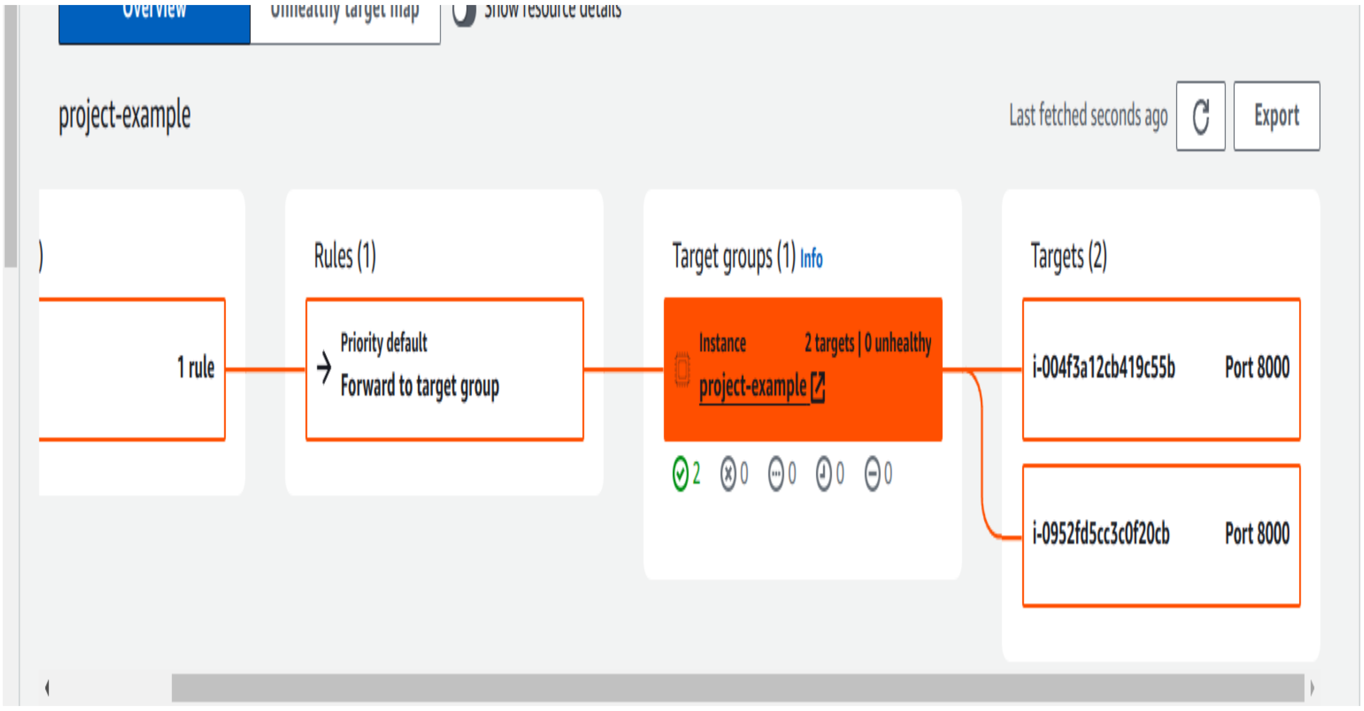


```
root@ip-10-0-158-148:~# python3 -m http.server 8000 &
[2] 2694
root@ip-10-0-158-148:~# Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
 ...
10.0.11.36 - - [31/Mar/2024 20:36:17] "GET / HTTP/1.1" 200 -
10.0.25.136 - - [31/Mar/2024 20:36:18] "GET / HTTP/1.1" 200 -
10.0.11.36 - - [31/Mar/2024 20:36:47] "GET / HTTP/1.1" 200 -
10.0.25.136 - - [31/Mar/2024 20:36:48] "GET / HTTP/1.1" 200 -
10.0.11.36 - - [31/Mar/2024 20:37:17] "GET / HTTP/1.1" 200 -
10.0.25.136 - - [31/Mar/2024 20:37:18] "GET / HTTP/1.1" 200 -
10.0.11.36 - - [31/Mar/2024 20:37:47] "GET / HTTP/1.1" 200 -
10.0.25.136 - - [31/Mar/2024 20:37:48] "GET / HTTP/1.1" 200 -
10.0.11.36 - - [31/Mar/2024 20:38:18] "GET / HTTP/1.1" 200 -
10.0.25.136 - - [31/Mar/2024 20:38:18] "GET / HTTP/1.1" 200 -
```

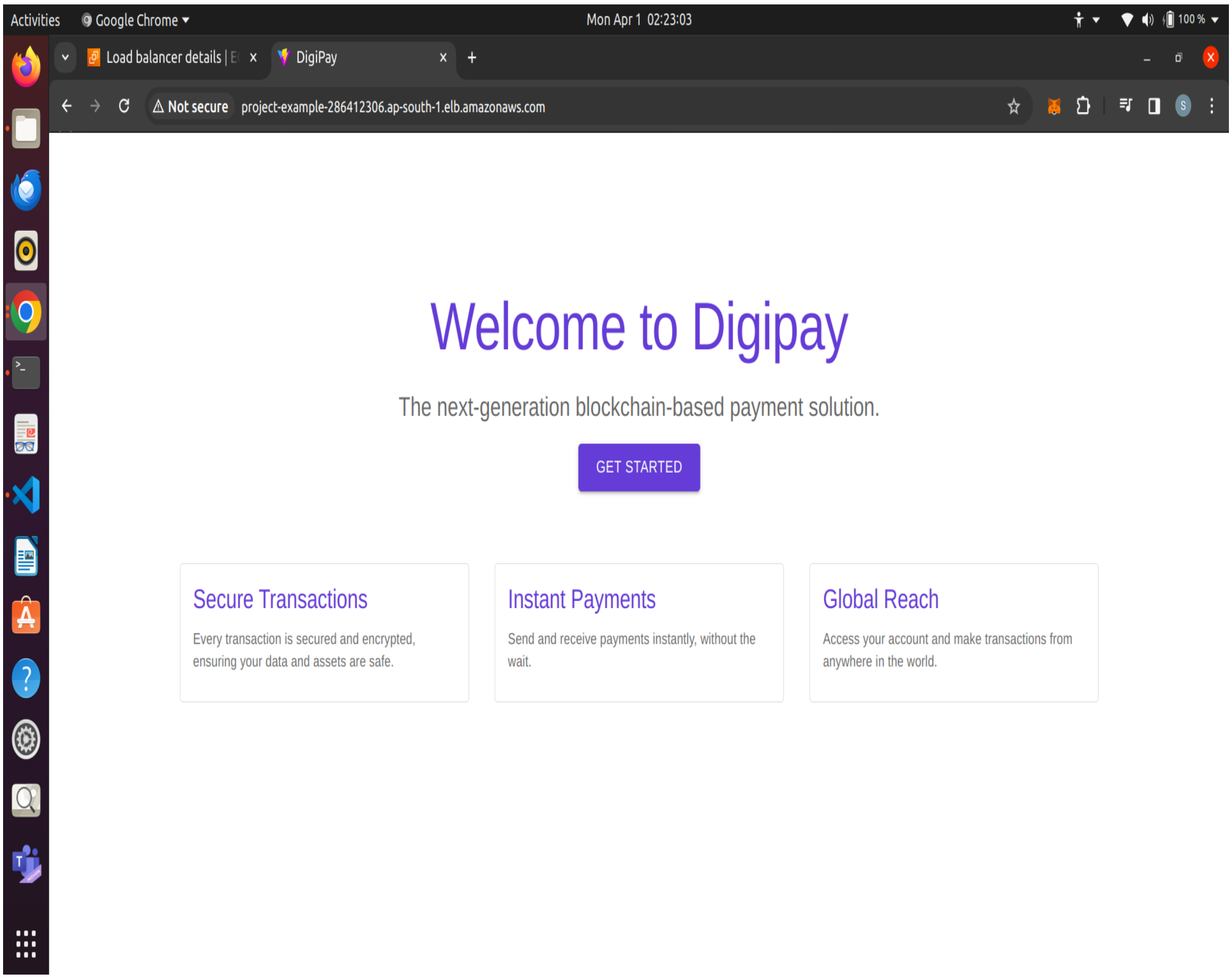
...ed. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly

Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly dete
i-004f3a12cb419c55b		8000	ap-south-1a	Healthy	-	March 31,...	Normal
i-0952fd5cc3c0f20cb		8000	ap-south-1b	Healthy	-	March 31,...	Normal

- The target group health check succeeds and this instance will also become healthy and requests will be sent evenly to both the servers (which are in private instances).



Now, If we see, sometimes the link directs to application: (running in private instance in private subnet south-1a)


Activities
Google Chrome
Mon Apr 1 02:23:03
100 %
Load balancer details | E
DigiPay
Not secure
project-example-286412306.ap-south-1.elb.amazonaws.com

Welcome to Digipay

The next-generation blockchain-based payment solution.

GET STARTED

Secure Transactions

Every transaction is secured and encrypted, ensuring your data and assets are safe.

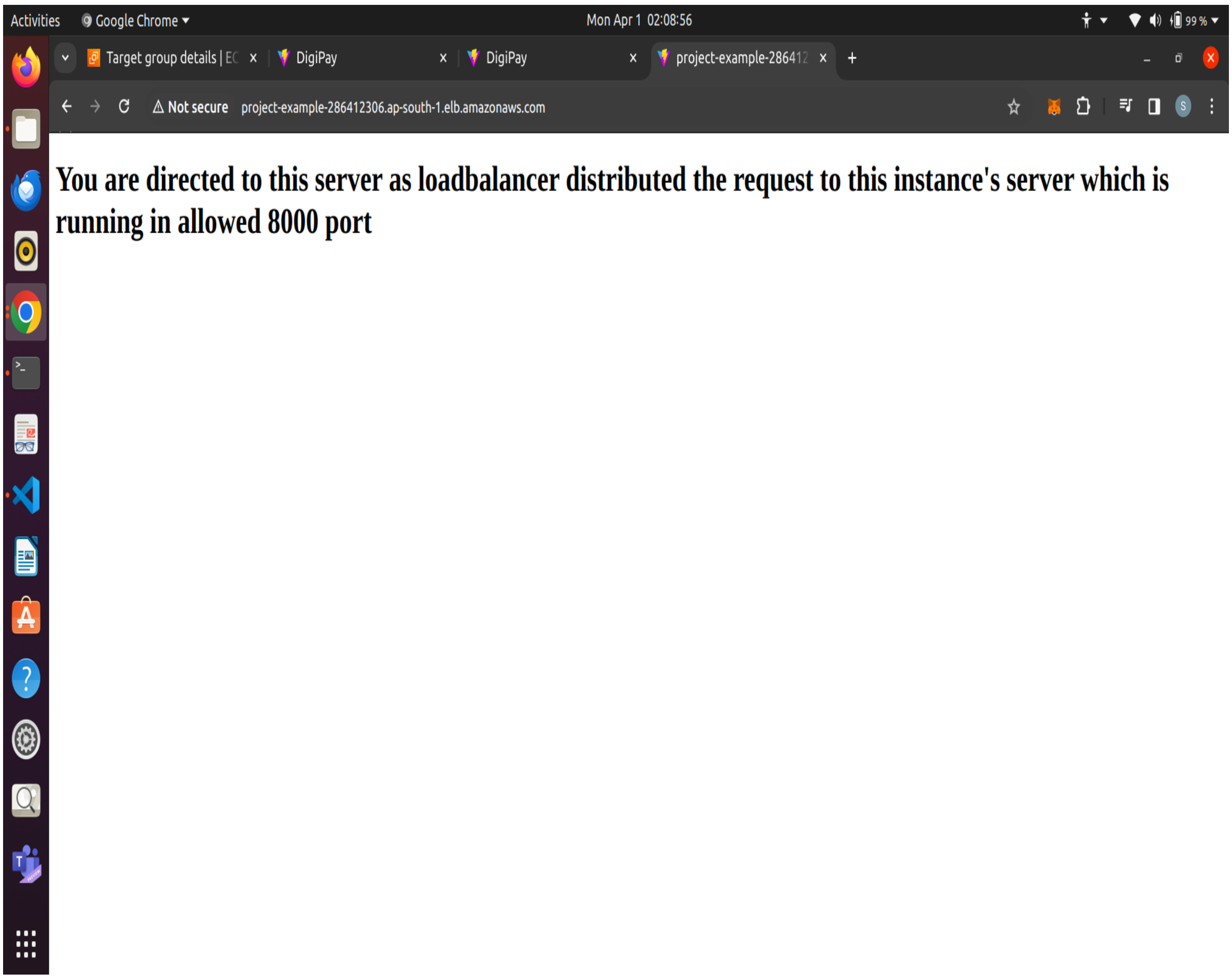
Instant Payments

Send and receive payments instantly, without the wait.

Global Reach

Access your account and make transactions from anywhere in the world.

And sometimes, the link directs to this page where I ran using python3 in 2nd private instance in private subnet south-1b



You are directed to this server as loadbalancer distributed the request to this instance's server which is running in allowed 8000 port