

**International Institute of Information
Technology, Bangalore**

Project Elective Report

OLD AGE HOME

Under the guidance of : Thangaraju Sir

Teaching Assistant : Suhruth Raghavendra



**IASH KUMAR TRIPATHY
MT2021056**

**GURURAJ MUJUMDAR
MT2021049**

Table of content

1. Abstract	3
2. Introduction	4
3. Project Description	5
3.1 Architecture diagram	6
3.2 Steps to setup Mongodb in Cloud	7
3.3 Code Structure	12
3.4 API Documentation	24
4. Software Development Life Cycle	26
4.1 Source Code Management	27
4.2 Build Tool	27
4.3 Testing Tool	28
4.4 Artifact	30
4.5 Deploy	31
4.6 Monitor	39
4.7 CI/CD Pipeline	45
5. Conclusion	53
6. Scope For Future Work	53
7. Appendix	54
8. References	55

1) Abstract

We have tried to create an Old Age Home Web Application to help the homeless senior citizens connect with each other, create a safe place for them to spend their remaining life. We have tried to reduce the crimes happening against the senior citizens.

We tried to implement various functionalities through this online platform so as to make all the paperwork and formalities as minimal as possible and the required help would be provided in the minimum time as possible.

We took advantage of various devops tools while developing our application in order to facilitate source code management, building, testing, deployment and monitoring in a convenient way.

2) Introduction:

The various functionalities which we tried to implement in our Web Application are as follows-

a) Contact-Us: This is a form provided at the bottom of the homepage of our website which the user can fill and ask about any relevant information about our initiative.

b) Adopt: Using this, If a family wants to adopt a senior citizen or wants to look after him/her then they can provide their basic information about their family background on our application itself.

Following their request a one to one meeting would be set up to discuss the process further.

c) Report Homeless Senior Citizens: Using this any person can report a crime against any senior citizen. For example, if any one sees any senior citizen being ill treated or beaten at home or at any location then that person could inform the details to us following which we shall proceed ahead by protecting them and then do a legal battle against the one who commits the crime. We check whether the complaint is genuine or not following which we shall inform the officials too.

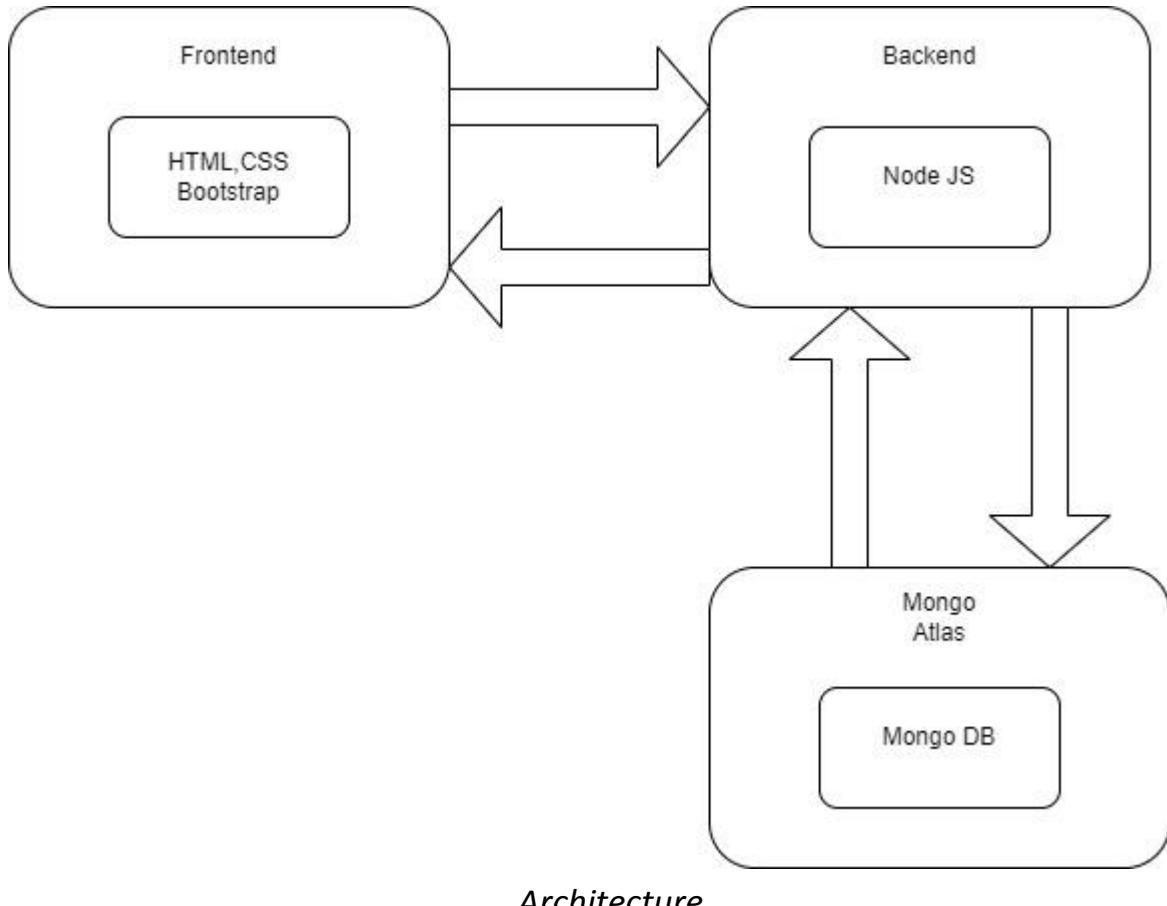
d) Donate: If anyone wants to provide a helping hand by either donating money for the upliftment of the senior citizens or wants to donate some items which are not useful for that person but could be utilized by poor senior citizens like books, clothes, furniture items etc then the required donation could be made on our web- application.

3) Project Description:

In this report we demonstrate some part of our web application and how we used DevOps toolchain for development of our project. We describe how we used different tools such as Git, Jenkins, Docker, Ansible, etc to achieve Continuous Integration, Continuous Deployment and Continuous Monitoring. We also illustrate how to install the above tools along with their configuration so that it can be easily replicated.

We concluded that using the DevOps tools made it easy for us to not worry about building, testing, deploying and monitoring once we had set up our pipeline. So we were able to focus more on writing code and building our website.

a) Architecture Diagram



- HTML5 + CSS3 + BOOTSTRAP : Front-end
- JavaScript : To handle the dynamic content in the html.
- EJS : html templating engine.
- Express : Server side framework of node js.
- MongoDb : Document Oriented Database.

Steps to setup MongoDB Database in Cloud :

The screenshot shows the MongoDB Atlas landing page. At the top, there's a navigation bar with links for MongoDB, Cloud, Software, Learn, Solutions, and Docs. A banner at the top right encourages users to "Join us for the MongoDB.live series beginning November 10!". On the left, there's a section titled "MongoDB Atlas" with a "Start free" button. In the center, there's a "Cloud Provider & Region" configuration panel. It shows "AWS, N. Virginia (us-east-1)" selected. Below this, there's a list of regions for AWS, Azure, and GCP. At the bottom of the page, there are links for Pricing, Getting started, Migrate to MongoDB Atlas, and Frequently Asked Questions. There are also several partner logos: eharmony, invision, SEGA, 7-ELEVEN, and KPMG.

MongoDB Atlas

Move faster with a cloud MongoDB service. Built for agile teams who'd rather spend time building apps than managing databases. Available on AWS, Azure, and GCP.

[Start free](#)

Already have an account? [Log in here](#) →

Cloud Provider & Region
Choose your preferred cloud provider and the region nearest to clients

AWS AWS, N. Virginia (us-east-1) >

Select a cloud provider to see its region availability

aws Azure

Configure a New Cluster by first selecting a region labeled with [SELECT REGION](#). Then add all nodes to the Cluster. To learn more about regions and clusters, click here.

▪ recommended region (1)

AWS Regions (1)
N. Virginia (us-east-1) (1) •
Ohio (us-west-1) (1)
N. California (us-west-1) (1)
Oregon (us-west-2) (1)
Montreal (ca-central-1) (1)

Azure Regions (1)
Ireland (eu-west-1) (1) •
London (eu-west-2) (1)
West Europe (eu-west-3) (1)
East US (east-us-1) (1)
Central US (east-us-2) (1)
Southeast Asia (asia-southeast-1) (1)
South America (sa-south-1) (1)
Brazil (sa-east-1) (1)

GCP Regions (1)
Tokyo (ap-northeast-1) (1) •
Seoul (ap-northeast-2) (1)
Singapore (ap-southeast-1) (1)
Ulm (eu-central-1) (1)

Pricing Getting started Migrate to MongoDB Atlas Frequently Asked Questions

eharmony invision SEGA 7-ELEVEN KPMG

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards.

Get started free
No credit card required

[Sign up with Google](#)

or

Your Company (optional)

How are you using MongoDB?

Your Work Email

First Name

Last Name

Password

8 characters minimum

I agree to the terms of service and privacy policy.

[Get started free](#)

Already have an account? [Sign in](#).

Choose a path. Adjust anytime.

Available as a fully managed service across 60+ regions on AWS, Azure, and Google Cloud

Shared Clusters

For teams learning MongoDB or developing small applications.

- ✓ Highly available auto-healing cluster
- ✓ End-to-end encryption
- ✓ Role-based access control

Create a cluster

Starting at
FREE

Dedicated Clusters

For teams building applications that need advanced development and production-ready environments.

- ✓ Includes all features from Shared Clusters
- ✓ Auto-scaling
- ✓ Network isolation
- ✓ Realtime performance metrics

Create a cluster

Starting at
\$0.08/hr*
*estimated cost \$56.94/month

Dedicated Multi-Region Clusters

For teams developing world-class applications that require multi-region resiliency or ultra-low latency.

- ✓ Includes all features from Shared and Dedicated Clusters
- ✓ Replicate data across multiple regions
- ✓ Globally distributed read and write operations
- ✓ Control data residency at the document level

Create a cluster

Starting at
\$0.13/hr*
*estimated cost \$98.55/month

[Dismiss](#)

[Advanced Configuration Options](#)

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Cloud Provider & Region

AWS, N. Virginia (us-east-1) ▾



★ Recommended region ⓘ

NORTH AMERICA

N. Virginia (us-east-1) ★

Oregon (us-west-2) ★

EUROPE

Ireland (eu-west-1) ★

Frankfurt (eu-central-1) ★

ASIA

Singapore (ap-southeast-1) ★

Mumbai (ap-south-1)

AUSTRALIA

Sydney (ap-southeast-2) ★

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage) >
Encrypted

Additional Settings

MongoDB 4.2, No Backup >

Cluster Name

Cluster0 >

Colt's Org - 2020-09... Access Manager Support Billing

All Clusters Colt

Project 0 Atlas Realm Charts

DATA STORAGE

- Clusters
- Triggers
- Data Lake

SECURITY

- Database Access
- Network Access
- Advanced

COLTS ONG - 2020-09-20 > PROJECT 0

Clusters

Find a cluster...

CLUSTER TAB

Cluster0 Version 4.2.9

CONNECT **METRICS** **COLLECTIONS** ...

Last 6 Hours

Operations R: W: 100.0%

Logical Size 0.0 B

512.0 MB Max

Connections 0

500 MAX

Last 6 Hours

Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Create a New Cluster

Last 6 Hours

Connect to Atlas

Follow this checklist to get started.

20%

Build your first cluster

Create your first database user

Whitelist your IP address

Load Sample Data [Optional]

Connect to your cluster

No thanks

Colt's Org - 2020-09... Access Manager Support Billing

All Clusters

Project 0 Atlas Realm Charts

DATA STORAGE

- Clusters
- Triggers
- Data Lake

SECURITY

- Database Access
- Network Access
- Advanced

COLTS ONG - 2020-09-20 > PROJECT 0

Database Access

Database Users Custom Roles

Add New Database User

Create a database user to grant an application or user, access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding Access Manager.

Authentication Method

Password Certificate AWS IAM (MongoDB 4.4, M10 and up)

MongoDB uses SCRAM as its default authentication method.

Password Authentication

Enter password SHOW

Database User Privileges

Select a built-in role or privileges for this user.

Restrict Access to Specific Clusters/Data Lakes

Enable to specify the resources this user can access. By default, all resources in this project are accessible.

OFF

Temporary User

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week.

OFF

System Status: All Good

©2020 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Get Started Feature Requests

The screenshot shows the 'Add IP Access List Entry' dialog box over a blurred background of the MongoDB Atlas interface. The dialog box has a title 'Add IP Access List Entry'. It contains instructions: 'Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. Learn more.' Below this are two buttons: 'ADD CURRENT ADDRESS' and 'ALLOW ACCESS FROM ANYWHERE'. A text input field labeled 'Access List Entry:' contains the placeholder 'Enter IP Address or CIDR Notation'. A 'Comment:' input field has the placeholder 'Optional comment describing this entry'. At the bottom, there is a toggle switch for 'This entry is temporary and will be deleted in' followed by a dropdown menu set to '6 hours', and a 'Cancel' button next to a 'Confirm' button.

The screenshot shows the 'Clusters' page in the MongoDB Atlas interface. A modal dialog box titled 'Connect to Cluster' is open. It has tabs: 'Setup connection security' (selected), 'Choose a connection method' (disabled), and 'Connect' (disabled). Under 'Choose a connection method', there are three options: 'Connect with the mongo shell' (disabled), 'Connect your application' (disabled), and 'Connect using MongoDB Compass' (disabled). Below these options are 'Go Back' and 'Close' buttons. In the background, the cluster details for 'Cluster0' are visible, including its tier (M0 Sandbox), region (AWS N. Virginia), type (Replica Set - 3 nodes), and linked realm app (None Listed). The cluster has 0 operations, 0 connections, and was last updated 6 hours ago.

The screenshot shows a modal dialog box titled 'Connect to Atlas'. It displays a checklist titled 'Follow this checklist to get started.' with a progress bar at 80%. The checklist items are: 'Build your first cluster' (checked), 'Create your first database user' (checked), 'Whitelist your IP address' (checked), 'Load Sample Data (Optional)' (unchecked), and 'Connect to your cluster' (checked). At the bottom of the dialog are 'No thanks' and 'Get Started' buttons.

Colt's Org - 2020-09... Access Manager Support Billing

Project 0 Atlas Realm Charts

DATA STORAGE

Clusters

We are deploying your changes (current action: configuring MongoDB)

COLT'S ORG - 2020-09-20 > PROJECT 0

Clusters

Find a cluster...

SANDBOX

Cluster0 Version 4.2.9

Operations R: 0 W: 0

CONNECT METRICS COLLECTIONS ...

Last 6 Hours

Connections 0

Last 6 Hours

CLUSTER TIER M0 Sandbox (General)

REGION AWS US N. Virginia (us-east-1)

TYPE Replica Set - 3 nodes

LINKED REALM APP None Linked

Connect to Cluster0

Setup connection security Choose a connection method Connect

Select your driver and version

DRIVER Node.js VERSION 3.6 or later

Add your connection string into your application code

Include full driver code example

mongodbs://our-first-user:<password>@cluster0.pilic.mongodb.net

Replace <password> with the password for the our-first-user user. Replace <dbname> with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Enhance Your Experience

For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Upgrade

Go Back Close

bbroy's Org - 2021-0... Access Manager Billing

All Clusters Get Help bbroy

SPE-Final-Project Atlas Realm Charts

DEPLOYMENT

Database

Data Lake

DATA SERVICES

Triggers

Data API PREVIEW

SECURITY

Database Access

Network Access

Advanced

New On Atlas

BBROY'S ORG - 2021-05-31 > SPE-FINAL-PROJECT

Database Deployments

Find a database deployment...

+ Create

Cluster0 Connect View Monitoring Browse Collections ...

Enhance Your Experience

R 0.03 W 0 Last 0 hours 0.03/s

Connections 8.0 Last 6 hours 10.0

In 293.7 B/s Out 2.5 KB/s Last 24 hours 2.5 KB/s

Data Size 153.3 KB Last 24 days 512.0 MB

Upgrade

VERSION 5.0.8 REGION AWS / Mumbai (ap-south-1) CLUSTER TIER M0 Sandbox (General) TYPE Replica Set - 3 nodes BACKUPS Inactive LINKED REALM APP None Linked ATLAS SEARCH Create Index

FREE SHARED

bbroy's Org - 2021-0... Access Manager Billing

All Clusters Get Help bbroy

SPE-Final-Project Atlas Realm Charts

DEPLOYMENT

Database

Data Lake

DATA SERVICES

Triggers

Data API PREVIEW

SECURITY

Database Access

Network Access

Advanced

Cluster0

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 1 COLLECTIONS: 4

+ Create Database

NAMESPACES

homeDB

admincredentials contacts labour meetings

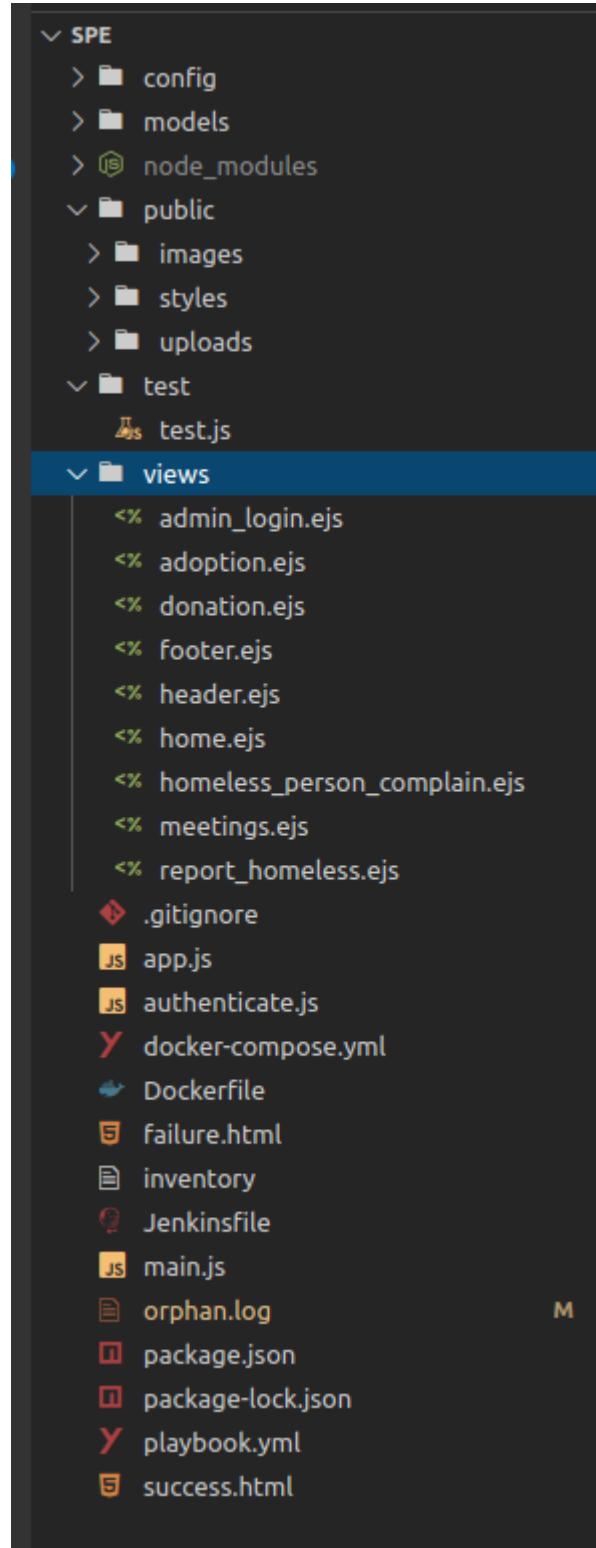
VISUALIZE YOUR DATA REFRESH

CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
admincredentials	1	79B	79B	3	60KB	20KB
contacts	1	186B	186B	1	20KB	20KB
labour	2	535B	268B	1	36KB	36KB
meetings	2	501B	251B	1	36KB	36KB

b)Code Structure

This is the structure of our NodeJS application-



- Node_modules - contains libraries which are required for the app.
- Test - contains unit test javascript files.
- Views - contains various ejs files which are using ejs engine.

Home.ejs = Contains the Home page of our application

Home Page :

 Old Age Home

Home Adopt Donate Report Homeless Admin Login

WHO ARE WE?

Poor old aged people undergoing great difficulties lacking & longing in love and affection and always are into mental stress, strain, for this reason, we started "Old Age Home Website for Poor Elders" with humanity and service to cater to the needs of such unfortunate elders. The aim of this website is to connect the homeless old people to the people who would wish to adopt them and also keep a track of the homeless person. Reporting any crimes being committed against the old homeless people. Helping them financially and by providing the basic needs. We sponsor elders boarding & lodging facilities with breakfast, lunch, dinner, Medical care, T.V for recreation.

ADOPTION

There are lakhs of homeless people in the country. Even we are able to change the life of 10% of them by adopting them and providing them with the basic needs then that would be a huge contribute to the society.

localhost:3000



 Old Age Home

Home Adopt Donate Report Homeless Admin Login

Get In Touch With Us.

If you would like to contact our organization, if you have any questions/concerns/thoughts/suggestions, feel free to contact us using the form below. We would love to hear from you and will endeavor to reply as soon as possible.

FULL NAME*

EMAIL

SUBJECT*

YOUR MESSAGE

SEND

Adoption Page :

 Old Age Home

Home Adopt Donate Report Homeless Admin Login

Book a Meeting

Kindly fill this form to book a meeting with us and we will guide you in this process of adopting a person from beginning to end with no charge at all.

FULL NAME*

Gender
 Male Female

Married
 Yes No

AGE*(ENTER BETWEEN 25 AND 50)

EMAIL

ADDRESS

YOUR JOB DESCRIPTION

YOUR ANNUAL INCOME*

PHONE NUMBER*

Your preferred date for meeting
dd/mm/yyyy

Your preferred time for meeting
 Morning Afternoon Evening

REASON FOR ADOPTION

localhost:3000/adoption

Donations Page :

 Old Age Home

Home Adopt **Donate** Report Homeless Admin Login

We request you to help and donate online to charity generously to feed the hungry poor oldage persons in elderly home. Your donation helps to make a difference in someone's lives and create hope on their lives. Beneficiaries always expressing sincere gratitude towards your kind generous donations. Donations in the form of money,basic needs are accepted.Kindly donate to our noble cause of the poor old age people in oldagehome.

DONATE FINANCIALLY

 **Donate Now**
Secured by Razorpay

Get 80G Tax Exemption By Donation: - While donating money online to charity through milaap you are eligible to claim tax exemption under section 80G of Income Tax Act, 1961. - All Offline / Online Donations are Tax deductible u/s 80G. - Seruds India Charity registered under FCRA (Foreign Contribution Regulation Act), eligible to receive donations online from abroad from NRIs, Foreign Nationals. Make a Donation to best Charity in India & help aged people for their daily needs in oldagehome through Milaap.

 Old Age Home

Home Adopt **Donate** Report Homeless Admin Log **Test Mode**

We request you to help and donate online to charity generously to feed the hungry poor oldage persons in elderly home. Your donation helps to make a difference in someone's lives. Beneficiaries always expressing sincere gratitude towards your kind generous donations. Donations in the form of money,basic needs are accepted.Kindly donate to our noble cause of the poor old age people in oldagehome.

DONATE FINANCIALLY

DONATION AMOUNT ● 0 0 X
Milaap
₹ 5,000.00

Donate an Amount of your Choice*
₹ 5000

Get 80G Tax Exemption By Donation: - While donating money online to charity through milaap you are eligible to claim tax exemption under section 80G of Income Tax Act, 1961. - All Offline / Online Donations are Tax deductible u/s 80G. - Seruds India Charity registered under FCRA (Foreign Contribution Regulation Act), eligible to receive donations online from abroad from NRIs, Foreign Nationals. Make a Donation to best Charity in India & help aged people for their daily needs in oldagehome through Milaap.

NEXT

Secured by 

We request you to help and donate online to our cause. Your donation helps to make a difference in someone's life. Your gratitude towards your kind generous donations. Our cause is noble.

← DONOR DETAILS X

Iash

₹ 5,000.00

Help poor oldage persons in elderly home. Your lives. Beneficiaries always expressing sincere needs are accepted. Kindly donate to our noble cause.

Email * Iash.Tripathy@iiitb.ac.in

Phone * 9087654321

Name * Iash

Address * IIIT Bangalore, Electronic City

City * Bangalore

PROCEED TO PAY

Milaap.

Secured by Razorpay

We request you to help and donate online to our cause. Your donation helps to make a difference in someone's life. Your gratitude towards your kind generous donations. Our cause is noble.

← PAYMENT METHOD X

Iash

₹ 5,000

English ▾

+919087654321 | Iash.Tripathy@iiitb.ac.in

CARDS, UPI & MORE

Card Visa, MasterCard, RuPay, and Maestro

UPI / QR

Netbanking All Indian banks

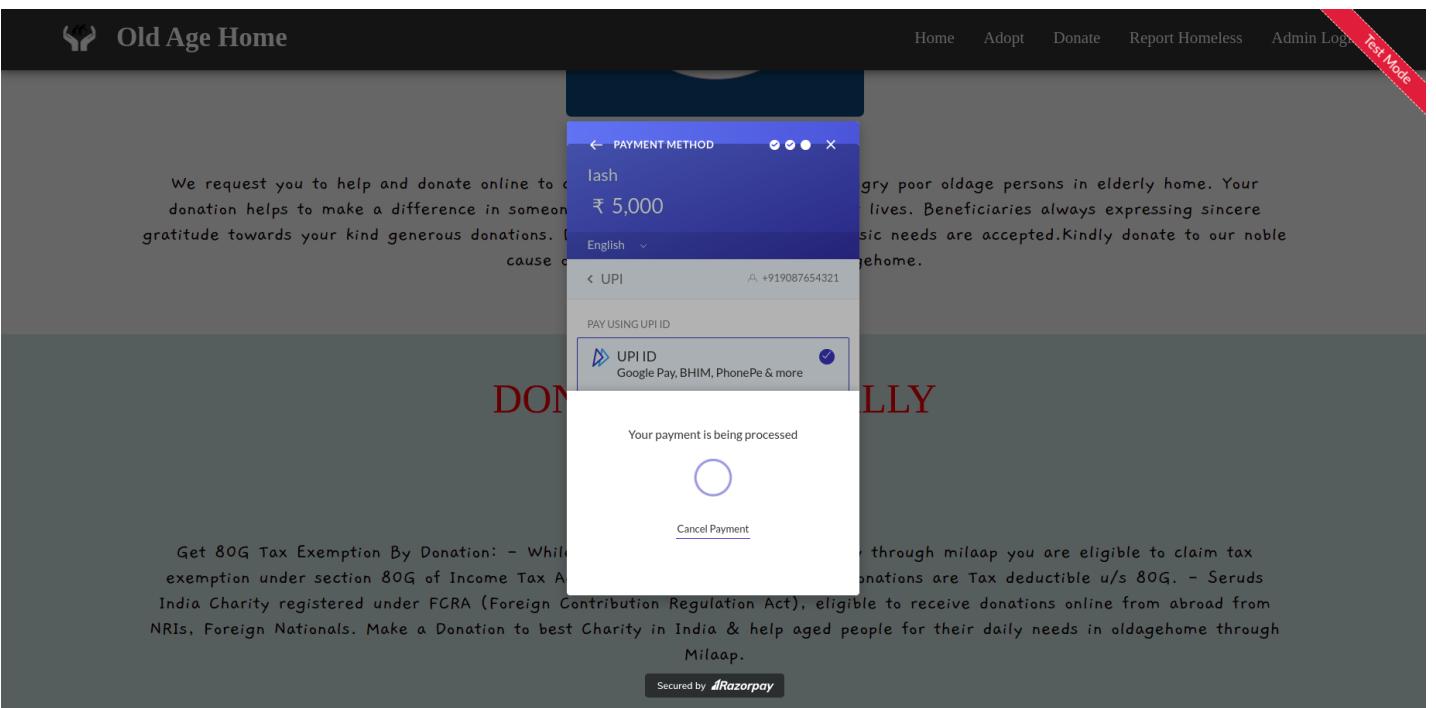
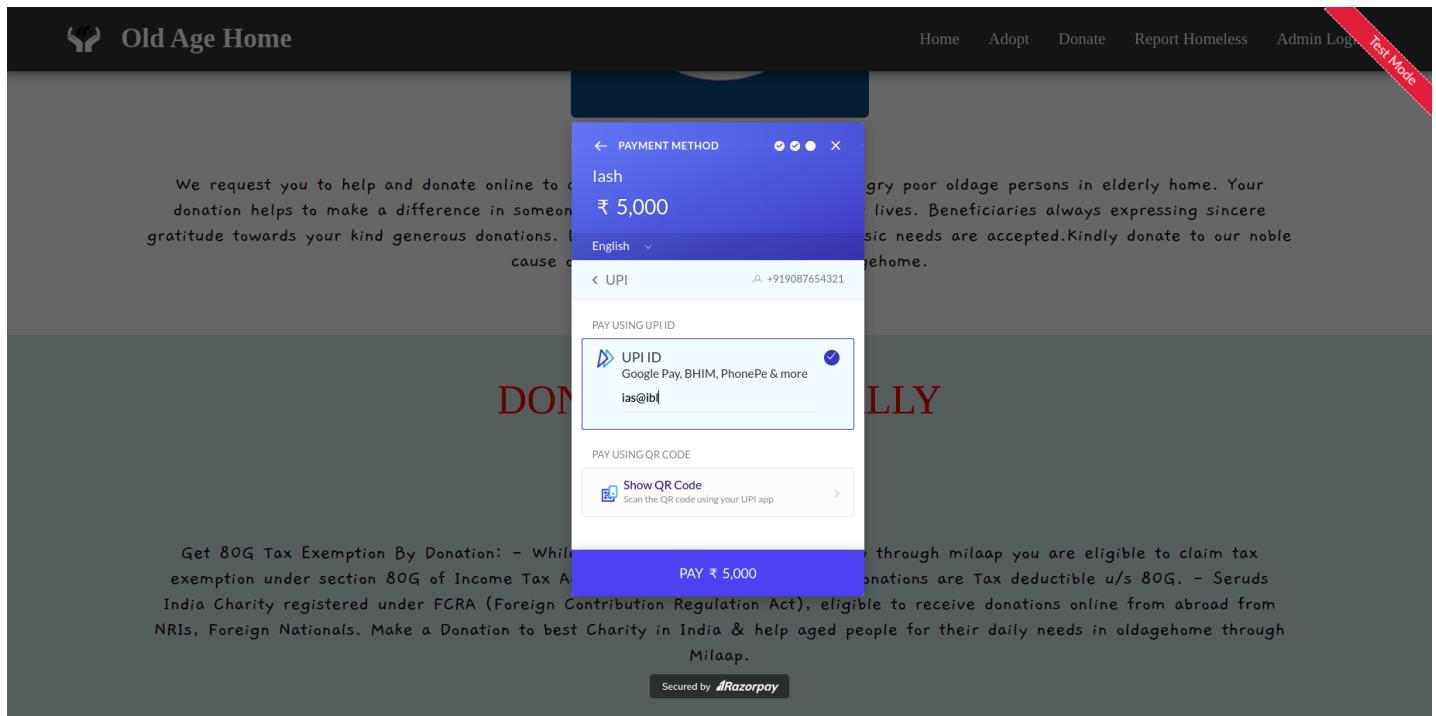
Wallet MobiKwik & More

Help poor oldage persons in elderly home. Your lives. Beneficiaries always expressing sincere needs are accepted. Kindly donate to our noble cause.

Get 80G Tax Exemption By Donation: - While exemption under section 80G of Income Tax Act, India Charity registered under FCRA (Foreign C NRIs, Foreign Nationals. Make a Donation to best

Milaap.

Secured by Razorpay





We request you to help and donate online to our cause. Your donation helps to make a difference in someone's life. We are grateful towards your kind generous donations. Kindly donate to our noble cause.

gry poor oldage persons in elderly home. Your lives. Beneficiaries always expressing sincere basic needs are accepted.Kindly donate to our noble home.



₹ 5000.00

Payment Successful!

DONATE

ALLY

A confirmation email has been sent to you
Payment ID: pay_JPq8A25MXzmfk

DONE

Get 80G Tax Exemption By Donation: - While making a donation through milaap you are eligible to claim tax exemption under section 80G of Income Tax Act. - Seruds India Charity registered under FCRA (Foreign Contribution Regulation Act), eligible to receive donations online from abroad from NRIs, Foreign Nationals. Make a Donation to best Charity in India & help aged people for their daily needs in oldagehome through Milaap.

Secured by

Report Homeless Senior Citizens Page :

 Old Age Home

[Home](#) [Adopt](#) [Donate](#) [Report Homeless](#) [Admin Login](#)



STOP THE CRIME!

The rapidly increasing population of elderly people has been accompanied by a number of challenges to the elderly, communities, and the state. The elderly have increasingly become more vulnerable to physical, financial, and emotional abuse. In many cases, citizens who are aware of these abuses may fail to report to the police or the elderly people themselves may fear engaging in legal battles. A salient factor which has been cited as aggravating the abuse of the senior citizens is the limited social services which the elderly people are receiving. One should know that these crimes are punishable seriously under the law. Kindly report such crimes to us.



LET'S TAKE A STAND !

Elder abuse is an important public health problem. A 2017 review of 52 studies in 28 countries from diverse regions estimated that over the past year 1 in 6 people (15.7%) aged 60 years and older were subjected to some form of abuse (1). Although rigorous data are limited, the review provides prevalence estimates of the proportion of older people affected by different types of abuse. We should take a stand to prevent it as much as possible.

 Old Age Home

[Home](#) [Adopt](#) [Donate](#) [Report Homeless](#) [Admin Login](#)

HELP THE HOMELESS PERSON

If you see this happening anywhere, please report us and we will try our best to help them and help them to live a better life.

YOUR FULL NAME*

YOUR EMAIL

YOUR PHONE NUMBER*

STATE WHERE YOU SAW THE OLD PERSON*

DISTRICT WHERE YOU SAW THE OLD PERSON*

ADDRESS/LANDMARK WHERE YOU SAW THE OLD PERSON

Image (if available)

No file chosen

DESCRIPTION OF THE OLD PERSON

Admin Login Page :

The screenshot shows a web page titled "Old Age Home" with a dark header bar. The header includes a logo of a dog, the text "Old Age Home", and navigation links for "Home", "Adopt", "Donate", "Report Homeless", and "Admin Login". Below the header is a large, light-colored rectangular area containing an "ADMIN LOGIN" form. The form has two input fields: "Admin Email" and "Enter password", both with placeholder text. A blue "SUBMIT" button is positioned below the password field. At the bottom of the page is a dark footer bar with the copyright notice "© Copyright 2022 - Old Age Home" and a URL "localhost:3000/admin/login".

Old Age Home

Home Adopt Donate Report Homeless Admin Login

ADMIN LOGIN

Admin Email _____

Enter password _____

SUBMIT

© Copyright 2022 - Old Age Home

localhost:3000/admin/login

Complaints Page (Only Visible to Admin) :

 Old Age Home

Home Complaints Meeting logout

HOMELESS SENIOR CITIZENS DATA

REPORTED PERSON NAME	EMAIL	STATE	DISTRICT	PHONE NO	ADDRESS	DESCRIPTION	IMAGE
dsd	h@gmail.com	AP	vizag	1234567890	FF-102,Sri Enclave,Infront of Astalakshmi Temple	dsdsdds	
Jinghan	jing@gmail.com	AP	Vizag	8907654321	Near YSR Cricket Stadium	The old person is female may be between 50 to 70 years of age.	

Meeting details Page (Only Visible to Admin) :

 Old Age Home

Home Complaints Meeting logout

ALL MEETING REQUESTS

NAME	EMAIL	GENDER	MARRIED(YES/NO)	AGE	JOB	ADDRESS
Rajat	raj@gmail.com	male	yes	27	Govt employee	FF201,Ram Nagar
Bharat	g@gmail.com	male	yes	45	dsfdgfd	sds

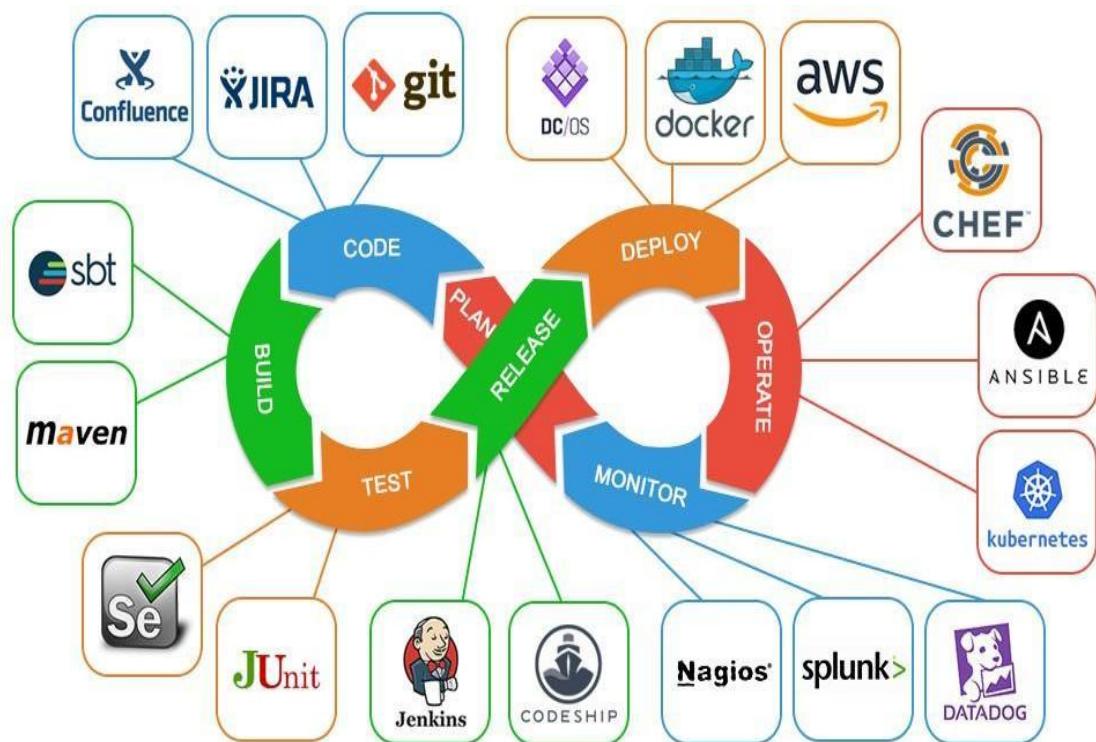
API Documentation

API	Method	Request Body	Request Header	Description	Response
/adoption	GET	NA	NA	get adoption page	adoption page
/adoption	POST	adoption details	NA	store adoption details in DB	Success/ Failure
/report-homeless	GET	NA	NA	get report homeless page	report homeless page
/report-homeless	POST	homeless details	NA	store details in DB	Success/ Failure
/donation	GET	NA	NA	get donation page	donations page
/admin-login	GET	NA	NA	get admin login page	admin login page
/admin-login	POST	credentials	NA	validate login credentials	JWT token

/homeless-persons-data	GET	NA	JWT token	fetch homeless persons data from DB	Page with homeless persons data
/meeting-data	GET	NA	JWT token	fetch meeting data from DB	Page with meeting data
/admin-logout	GET	NA	JWT token	admin logout	Success/ Failure

4) Software Development Life Cycle

In DevOps, the gap between development and operations team is bridged by following continuous integration / continuous deployment.



4.1)Source Code Management

Source Code Management tools are used for collaborative and parallel development, maintaining versions, etc. Git was used for SCM, and Github was used for remote repository management which was then used by Jenkins for continuous integration. Following is the link for the Github repository:

<https://github.com/iashtripathy/SPE.git>

Build Tool:

Our website runs on a NodeJs server. We used npm to install and manage all the dependencies for the project. **npm** creates a package.json file, which is responsible for maintaining version information of all the packages included , running shell commands/scripts , maintaining author/licensing information etc.

Testing Tool:

Mocha is a JavaScript testing framework running on Node.js. Mocha was used for the testing stage for this project. This was done by adding the test command script – ‘mocha –exit’ as the testing command in the package.json file.

Here are the screenshots of unit-test-cases written in mocha:

```
test > test.js > ...
1  const chai = require('chai');
2  const chaiHttp = require('chai-http');
3  const server = require('../app');
4  const should = chai.should();
5
6  chai.use(chaiHttp);
7
8
9  /*-----Test Case-1-----*/
10
11 describe('/ home route', () => {
12   it('it should check the home page', (done) => {
13     chai.request(server)
14       .get('/')
15       .end((err, res) => {
16         res.should.have.status(200);
17         done();
18       });
19   });
20 });
21
22 /*-----Test Case-2-----*/
23
24 describe('/ adoption route', () => {
25   it('it should check the adoption page', (done) => {
26     chai.request(server)
27       .get('/adoption')
28       .end((err, res) => {
29         res.should.have.status(200);
30         done();
31       });
32   });
33 });
34
```

```
35
36  /*-----Test Case-3-----*/
37
38  describe('/ report-homeless route', () => {
39    it('it should Report Homeless People route', (done) => {
40      chai.request(server)
41        .get('/report-homeless')
42        .end((err, res) => {
43          res.should.have.status(200);
44          done();
45        });
46    });
47 });
48
49  /*-----Test Case-4-----*/
50
51  describe('/ donation route', () => {
52    it('it should check the donation page', (done) => {
53      chai.request(server)
54        .get('/donation')
55        .end((err, res) => {
56          res.should.have.status(200);
57          done();
58        });
59    });
60 });
61
```

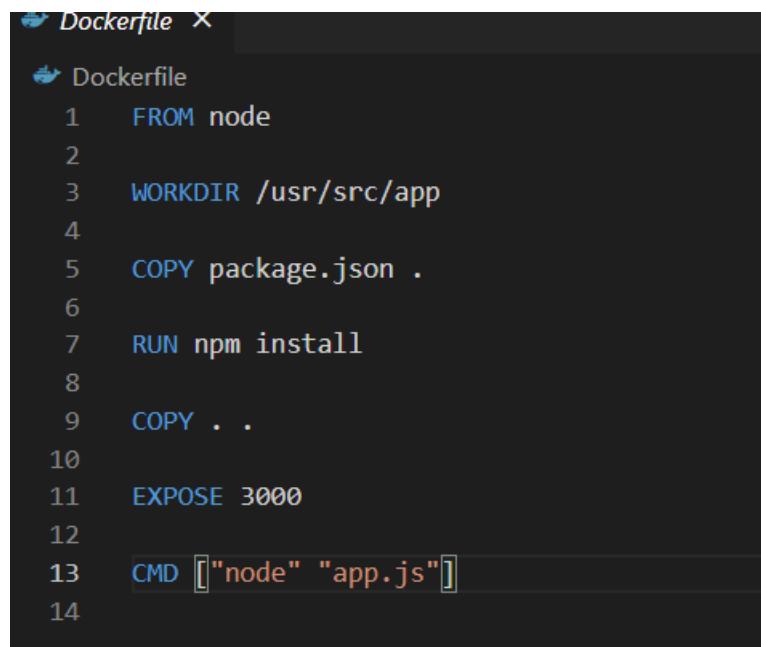
```
61
62 /*-----Test Case-5-----*/
63
64 describe('/ Admin Login route', () => {
65   it('It checks Admin Login page', (done) => {
66     chai.request(server)
67       .get('/admin-login')
68       .end((err, res) => {
69         res.should.have.status(200);
70         done();
71       });
72   });
73 });
74
75
76
77 /*-----Test Case-6-----*/
78
79 describe('/ Admin Logout route', () => {
80   it('It checks admin Logout', (done) => {
81     chai.request(server)
82       .get('/admin-logout')
83       .end((err, res) => {
84         res.should.have.status(200);
85         done();
86       });
87   });
88 });
```

Artifact:

Docker was used to ship our application along with all its dependencies in a docker image. The advantage of using Docker is that we can guarantee that the app will run on any platform provided that docker is installed. Docker images are used as an artifact of this project. Docker image is built and uploaded on Docker Hub. Following is the link for Docker Hub image:

<https://hub.docker.com/repository/docker/gururajmujumdar/spefinal>

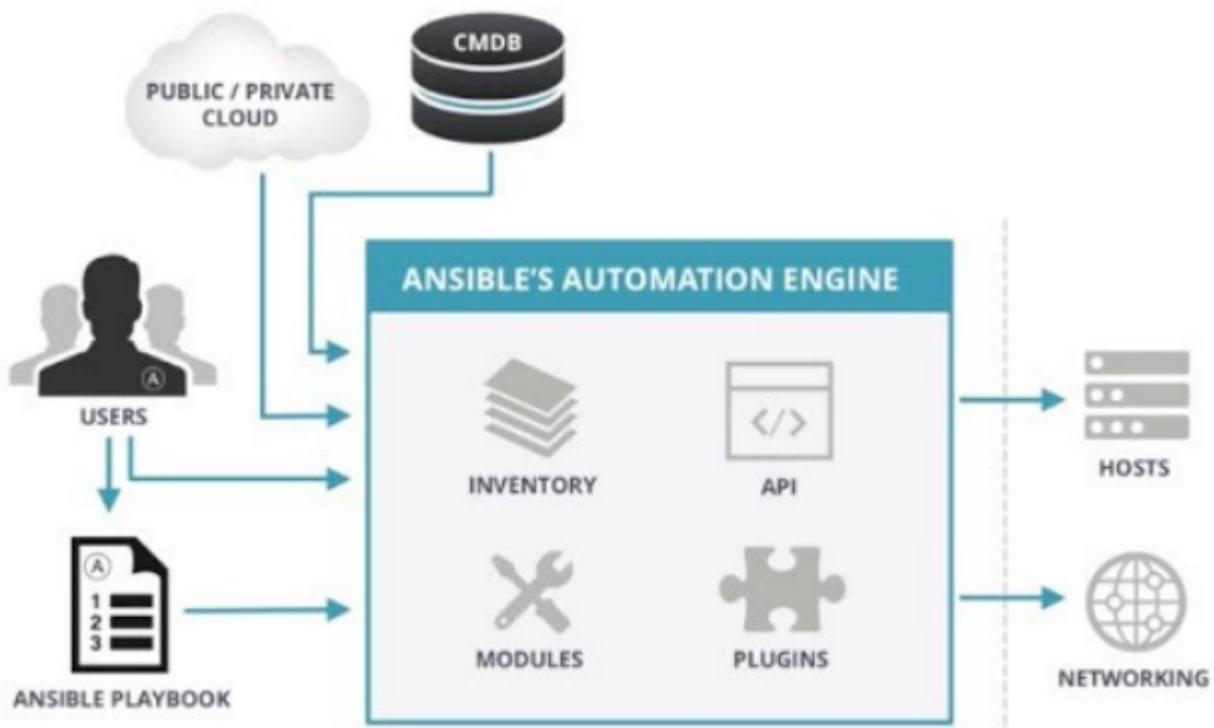
Here are the contents of our Dockerfile



```
1  FROM node
2
3  WORKDIR /usr/src/app
4
5  COPY package.json .
6
7  RUN npm install
8
9  COPY . .
10
11 EXPOSE 3000
12
13 CMD ["node", "app.js"]
14
```

Deploy:

Ansible: Ansible is an IT automation tool that can configure systems, deploy software and orchestrate advanced IT tasks such as continuous deployment. Ansible offers Infrastructure As Code capabilities.



Ansible Automation Engine

Image: <https://docs.ansible.com/>

The Ansible architecture has two categories of systems:

- Controller Node
- Managed Hosts

Controller Node: Controller node is a device that runs ansible files which executes commands on other hosts.

Managed Hosts: Any device that is managed by the controller node is called a managed host.

Ansible works by connecting to the hosts using a protocol like SSH and then sending small programs called Ansible modules to these hosts. Ansible module describes the desired state of the system.

Ansible consists of two configuration files: inventory file and Ansible playbook. The inventory file contains details of the managed hosts that the controller node can manage and communicate with. Ansible playbook is a configuration file that provides instructions on what needs to be done in order to bring the managed host to a desired state.

Below are the screenshots of inventory file and Ansible playbook



```
inventory
[aws]
3.109.153.193 ansible_user=ec2-user ansible_python_interpreter=/usr/bin/python3 ansible_ssh_private_key_file=/home/rocky/Des
```

A screenshot of a terminal window titled "inventory". The window displays a single line of text: "[aws] 3.109.153.193 ansible_user=ec2-user ansible_python_interpreter=/usr/bin/python3 ansible_ssh_private_key_file=/home/rocky/Des". The terminal has a dark background with white text.

In our project, an AWS EC2 instance is used as a managed host. AWS EC2 provides virtual machines as a scalable service on which one can deploy their own applications. The EC2 instance can be launched from the AWS console.

```

-- 
- name: Pull docker image from docker hub
  hosts: aws
  remote_user: ec2-user
  tasks:
    - name: stop container
      command: docker kill spe_final

    - name: remove container
      command: docker rm spe_final

    - name: deleting image
      command: docker rmi gururajmujumdar/spefinal:latest

    - name: pull new image
      command: docker pull gururajmujumdar/spefinal:latest

    - name: run image
      command: docker run -d --name spe_final -p 3000:3000 gururajmujumdar/spefinal:latest

```

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

[Cancel and Exit](#)

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"
Search by Systems Manager parameter

Quick Start	
<input type="checkbox"/> My AMIs	
<input type="checkbox"/> AWS Marketplace	
<input type="checkbox"/> Community AMIs	
<input type="checkbox"/> Free tier only <small>(i)</small>	

 Amazon Linux Free tier eligible	Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-079b5e5b3971bd10d (64-bit x86) / ami-0b5794cc5d751f003 (64-bit Arm) Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	<input type="button" value="Select"/>
 Amazon Linux Free tier eligible	Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-08f3712c8ca5af75e (64-bit x86) / ami-0494f7473bf694265 (64-bit Arm) Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/> Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot Instances
Network	Network: <input type="text" value="vpc-e05c9a8b (default)"/> Create new VPC Subnet: <input type="text" value="No preference (default subnet in any Availability Zone)"/> Create new subnet Auto-assign Public IP: <input type="text" value="Use subnet setting (Enable)"/>
Hostname type	<input type="text" value="Use subnet setting (IP name)"/>
DNS Hostname	<input type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests
Placement group	<input type="checkbox"/> Add Instance to placement group
Capacity Reservation	<input type="text" value="Open"/>
Domain join directory	<input type="text" value="No directory"/> Create new directory

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and several filter options: 'Instance state = running' (selected), 'Clear filters', 'Name' dropdown, 'Instance ID' (i-055610df09fc3217c), 'Instance state' (Running), 'Instance type' (t2.micro), 'Status check' (2/2 checks passed), 'Alarm status' (No alarms), 'Availability Zone' (ap-south-1b), and 'Public IPv4 DNS' (ec2-3-109-153-193.ap-south-1.compute.amazonaws.com). Below the filters is a table with one row for the selected instance. The table columns include Name, Instance ID, Instance state, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The instance details page below shows the instance summary with fields like Instance ID (i-055610df09fc3217c), Public IPv4 address (3.109.153.193), Instance state (Running), Private IPv4 addresses (172.31.13.133), and Public IPv4 DNS (ec2-3-109-153-193.ap-south-1.compute.amazonaws.com).

After launching the instance, AWS will provide a PEM file that can be used to log in to the instance. AWS will also assign a public IPv4 address to this instance. We can connect to the instance using the '**EC2 instance connect**' option in the AWS console.

Connect to instance Info

Connect to your instance i-055610df09fc3217c using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 Serial Console

Instance ID

i-055610df09fc3217c

Public IP address

3.109.153.193

User name

ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

Connect

```
Last login: Thu May  5 08:39:27 2022 from 103.156.19.229
```

```
 _ | _ | _ )  
_ | ( _ | _ /  Amazon Linux 2 AMI  
_ | \_ | _ |
```

```
https://aws.amazon.com/amazon-linux-2/  
22 package(s) needed for security, out of 46 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-13-133 ~]$ █
```

i-055610df09fc3217c

Public IPs: 3.109.153.193 Private IPs: 172.31.13.133

Docker can be installed on this EC2 instance using below command

```
sudo amazon-linux-extras install docker
```

In the inventory file, the public IPv4 address of the EC2 instance is specified along with the PEM file path. Ansible will use this PEM file to log in to the instance using SSH.

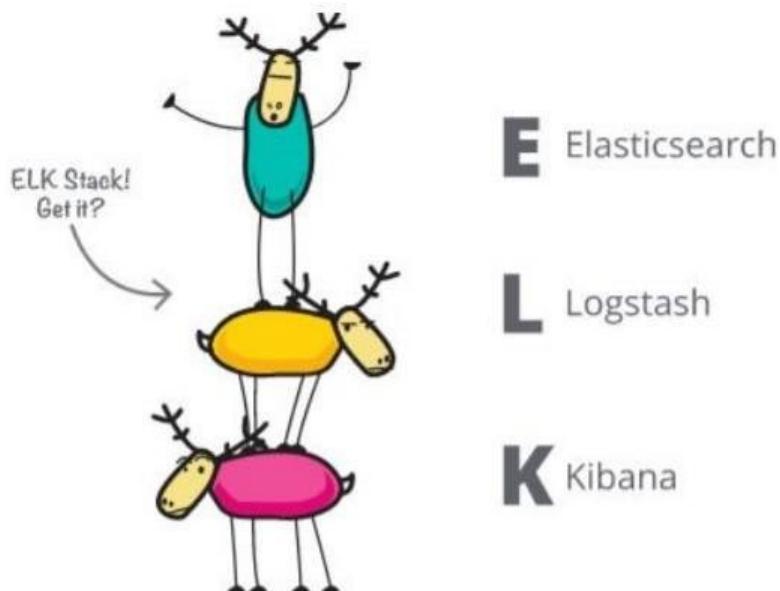
The Ansible playbook will perform below tasks on the EC2 instance

- Stop running container
- Remove the container
- Delete existing application image
- Pull new image from Docker hub
- Run the image in a container

Monitoring using ELK stack:

Continuous Monitoring comes in at the end of the DevOps pipeline. Once the software is released into production, Continuous Monitoring will notify dev and QA teams in the event of specific issues arising in the prod environment. It provides feedback on what is going wrong, which allows the relevant people to work on necessary fixes as soon as possible.

ELK Stack



"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana.

- Elasticsearch is a search and analytics engine.
- Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch.
- Kibana lets users visualize data with charts and graphs in Elasticsearch.

Logging:

In our application, we have used winston library for logging. winston is a simple and universal logging library with multiple transports. A transport is a storage space for the logs. Below is a snapshot of the log configuration file

```

1 const {
2   createLogger,
3   transports,
4   format
5 } = require("winston");
6
7 const logger = createLogger({
8
9   transports: [
10     new transports.File([
11       filename: 'orphan.log',
12       //format: format.timestamp(),
13       format: format.combine(format.timestamp(), format.json())
14       //json: false
15     ])
16   ]
17 });
18
19
20 module.exports = logger;

```

Using this configuration, all the logs will be stored in a log file named ‘**orphan.log**’. This configuration also specifies that the logs will be stored in JSON format. Below is a snapshot of the log file

```
☰ orphan.log
1  [{"level": "info", "message": "Server started on port 3000", "timestamp": "2022-04-26T21:31:57.608Z"}]
2  {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:32:20.854Z"}
3  {"level": "info", "message": "Entry saved for contact-us", "timestamp": "2022-04-26T21:33:43.818Z"}
4  {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:33:47.017Z"}
5  {"level": "info", "message": "Donation page request", "timestamp": "2022-04-26T21:34:02.700Z"}
6  {"level": "info", "message": "Report homeless person page request", "timestamp": "2022-04-26T21:34:20.935Z"}
7  {"level": "info", "message": "Report homeless post request saved", "timestamp": "2022-04-26T21:35:52.714Z"}
8  {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:35:54.615Z"}
9  {"level": "info", "message": "Report homeless person page request", "timestamp": "2022-04-26T21:36:06.755Z"}
10 {"level": "info", "message": "Admin Login page request", "timestamp": "2022-04-26T21:36:10.446Z"}
11 {"level": "error", "message": "Admin Not Found", "timestamp": "2022-04-26T21:36:30.409Z"}
12 {"level": "info", "message": "Admin Login Successful", "timestamp": "2022-04-26T21:36:55.240Z"}
13 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:36:55.294Z"}
14 {"level": "info", "message": "homeless person data display page request", "timestamp": "2022-04-26T21:37:15.482Z"}
15 {"level": "info", "message": "All meetings data display page request", "timestamp": "2022-04-26T21:37:33.046Z"}
16 {"level": "info", "message": "Loggin out admin", "timestamp": "2022-04-26T21:37:44.300Z"}
17 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:37:44.396Z"}
18 {"level": "info", "message": "Server started on port 3000", "timestamp": "2022-04-26T21:38:04.363Z"}
19 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:41:00.256Z"}
20 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:41:17.271Z"}
21 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:44:12.208Z"}
22 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:54:30.252Z"}
23 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:55:32.320Z"}
24 {"level": "info", "message": "Adoption page request", "timestamp": "2022-04-26T21:55:38.279Z"}
25 {"level": "info", "message": "Donation page request", "timestamp": "2022-04-26T21:55:43.679Z"}
26 {"level": "info", "message": "Report homeless person page request", "timestamp": "2022-04-26T21:55:48.967Z"}
27 {"level": "info", "message": "Admin Login page request", "timestamp": "2022-04-26T21:55:54.725Z"}
28 {"level": "info", "message": "Admin Login Successful", "timestamp": "2022-04-26T21:56:06.048Z"}
29 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:56:06.101Z"}
30 {"level": "info", "message": "homeless person data display page request", "timestamp": "2022-04-26T21:56:13.710Z"}
31 {"level": "info", "message": "All meetings data display page request", "timestamp": "2022-04-26T21:56:18.821Z"}
32 {"level": "info", "message": "Logging out admin", "timestamp": "2022-04-26T21:56:21.463Z"}
33 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:56:21.531Z"}
34 {"level": "info", "message": "Home page request", "timestamp": "2022-04-26T21:57:41.516Z"}
35 {"level": "info", "message": "Admin Login page request", "timestamp": "2022-04-26T21:57:44.022Z"}
36 {"level": "info", "message": "Server started on port 3000", "timestamp": "2022-04-26T22:00:31.469Z"}
37 {"level": "info", "message": "Admin Login page request", "timestamp": "2022-04-26T22:00:36.733Z"}
38 {"level": "info", "message": "Admin Login page request", "timestamp": "2022-04-26T22:02:00.909Z"}
```

Data Visualization using Kibana:

1) Create the index pattern of the logs file

The screenshot shows the Elasticsearch 'Import data' interface for the 'orphan.log' file. The 'Simple' tab is selected. The 'Index name' field contains 'oldagehome_index'. A checked checkbox labeled 'Create data view' is present. A 'Reset' button is at the bottom left. Below the form, a progress bar indicates the import status with five green circular markers and the text: 'File processed', 'Index created', 'Ingest pipeline created', 'Data uploaded', and 'Data view created'. A light blue banner at the bottom states '✓ Import complete' and 'Index oldagehome_index'. At the bottom are 'Back' and 'Cancel' buttons. The top navigation bar includes 'Find apps, content, and more. Ex: Discover', a user profile icon, and a search bar.

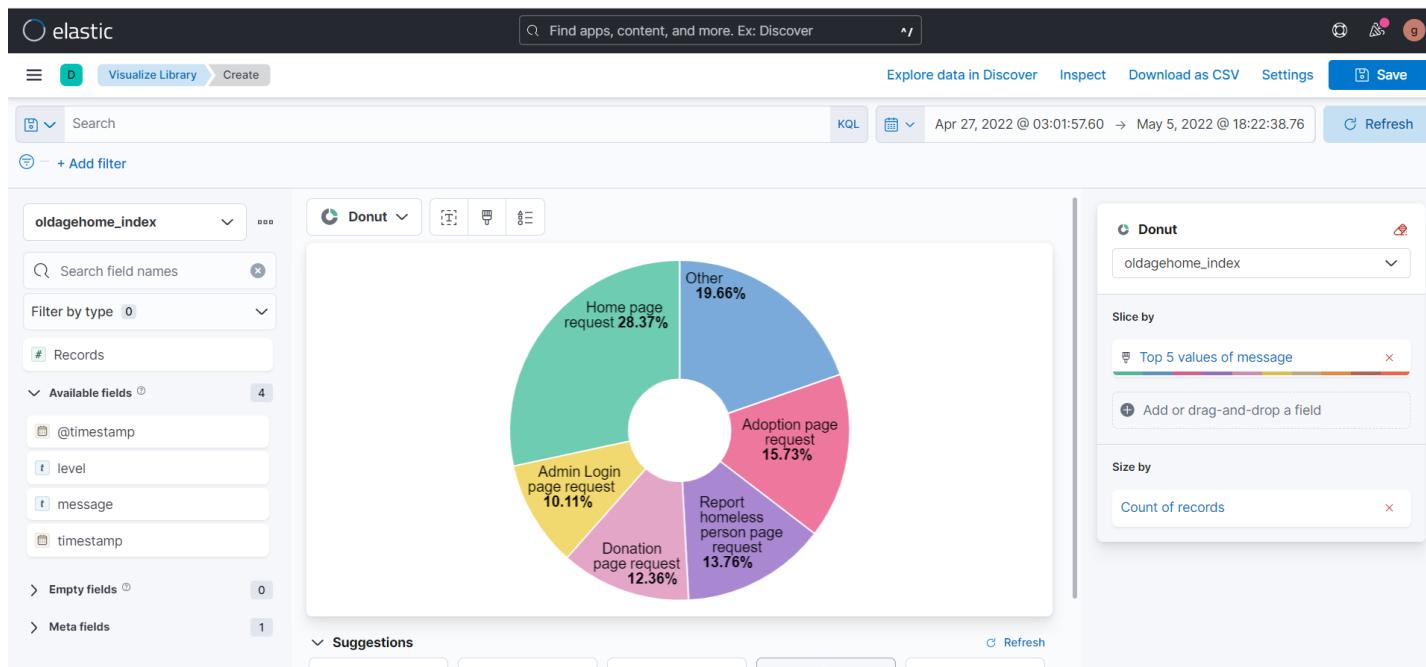
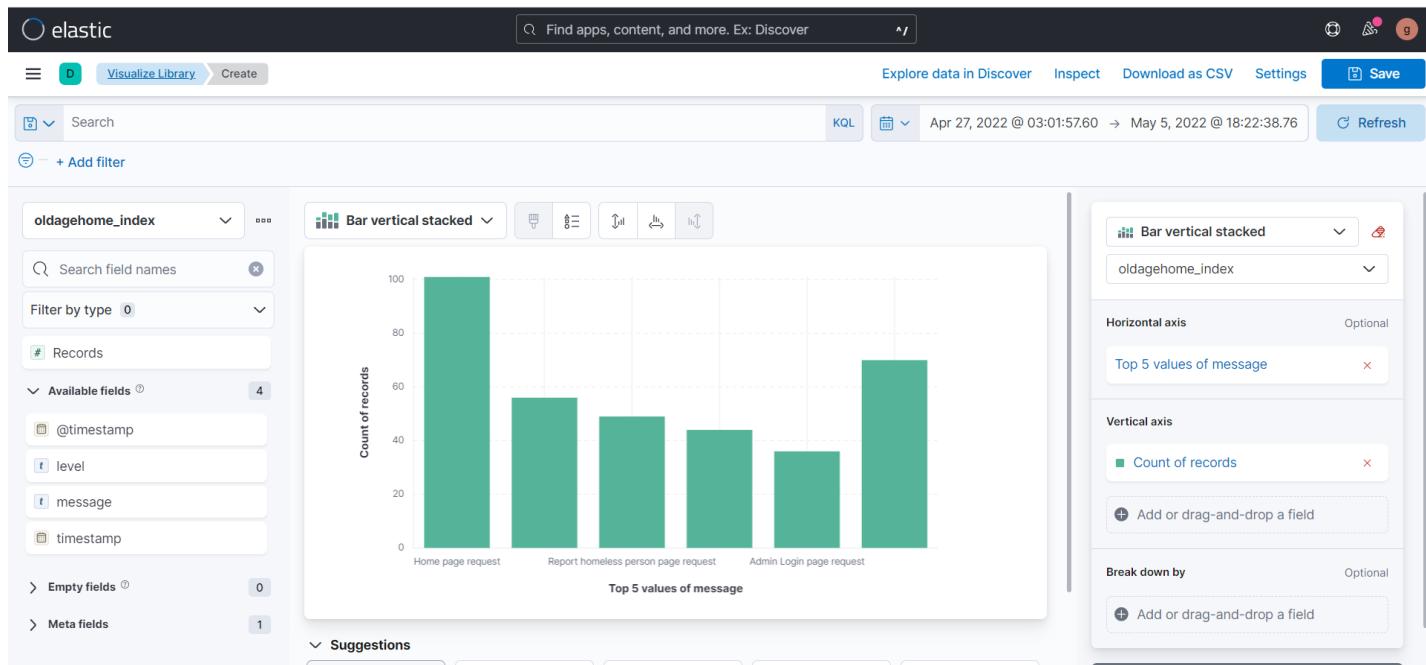
2) Discover the logs using Elastic search after creating the index pattern

The screenshot shows the Elastic Discover interface. At the top, there's a search bar with placeholder text "Find apps, content, and more. Ex: Discover". Below it, the index name "oldagehome_index" is selected. The main area displays a histogram titled "356 hits" showing the distribution of documents over time from April 27th to May 5th. A tooltip "A better way to explore" points to the "Field statistics" section, which is currently set to "Documents". On the right, a sidebar shows user information: "gururaj.mujumdar@iiitb.ac.in", "Profile", "Account & Billing", "Preferences", and "Log out". The bottom section shows a table of documents sorted by timestamp.

Document	Timestamp	Level	Message
May 5, 2022 @ 18:22:38.762	info	Server started on port 3000	@timestamp May 5, 2022 @ 18:22:38.762 _id tFbnYABSHyaLw7kLWKS _index oldagehome_index _score -
May 5, 2022 @ 18:20:21.631	info	Server started on port 3000	@timestamp May 5, 2022 @ 18:20:21.631 _id tFbnYABSHyaLw7kLWKS _index oldagehome_index _score -

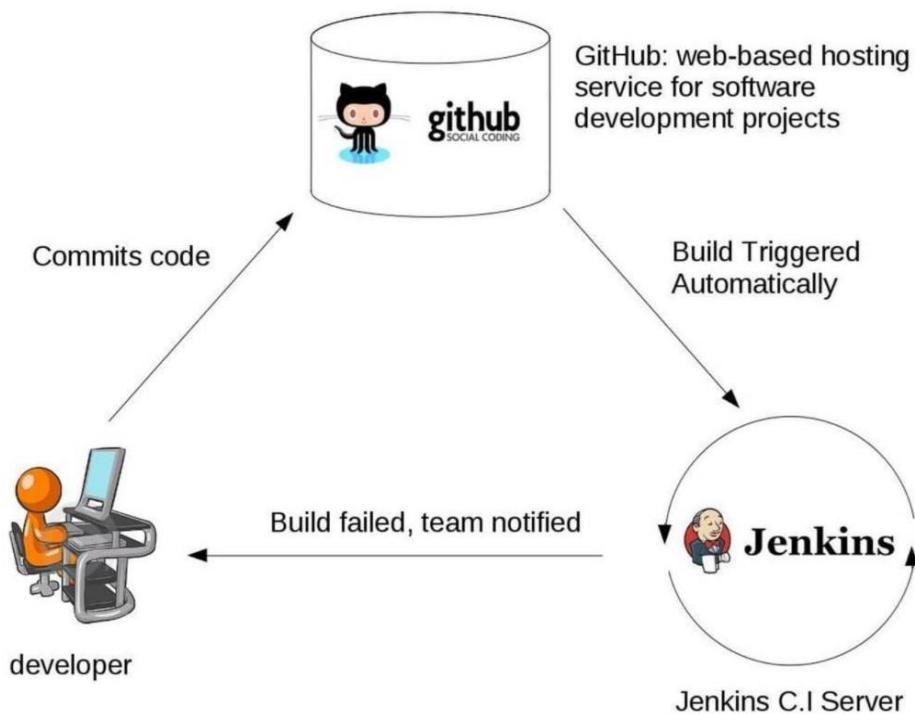
This screenshot shows the same Elastic Discover interface as above, but with a detailed view of the "message" field. A modal window titled "message" displays the "Top 5 values" for this field. The values are: "Home page request" (28.4%), "Adoption page request" (15.7%), "Report homeless person pag..." (13.8%), "Donation page request" (12.4%), and "Admin Login page request" (10.1%). The modal also states "Exists in 356 / 356 records". The background histogram and document list are visible but dimmed.

3) Visualization of Data(Logs) using Kibana:



CI/CD Pipeline Using Jenkins:

In Continuous Integration after a code commit, the software is built and tested immediately. In a large project with many developers, commits are made many times during a day. With each commit code is built and tested. If the test is passed, the build is tested for deployment. If deployment is a success, the code is pushed to production. This commit, build, test, and deploy is a continuous process and hence the name continuous integration/deployment



Steps to start a Jenkins Pipeline

1. Click on *New Item* -> Enter the name for the project ->Select *Pipeline*
2. In the General section add relevant descriptions and select GitHub project and provide the GitHub repository link.
3. In the build triggers section, select GITScm polling and add how frequently we want the Jenkins to look for changes in Github.This will trigger our Jenkins pipeline after every commit to the repository.
4. In the pipeline section, select *Pipeline script from SCM* from the Definition menu. Select Git as SCM, and provide a repository URL. We are using Pipeline as code to define the pipeline. For this purpose, Jenkinsfile will be used which will contain a pipeline script. Jenkinsfile is in the root of the Github repository.

Next we need configure the pipeline: Here are the contents of the Jenkinsfile:

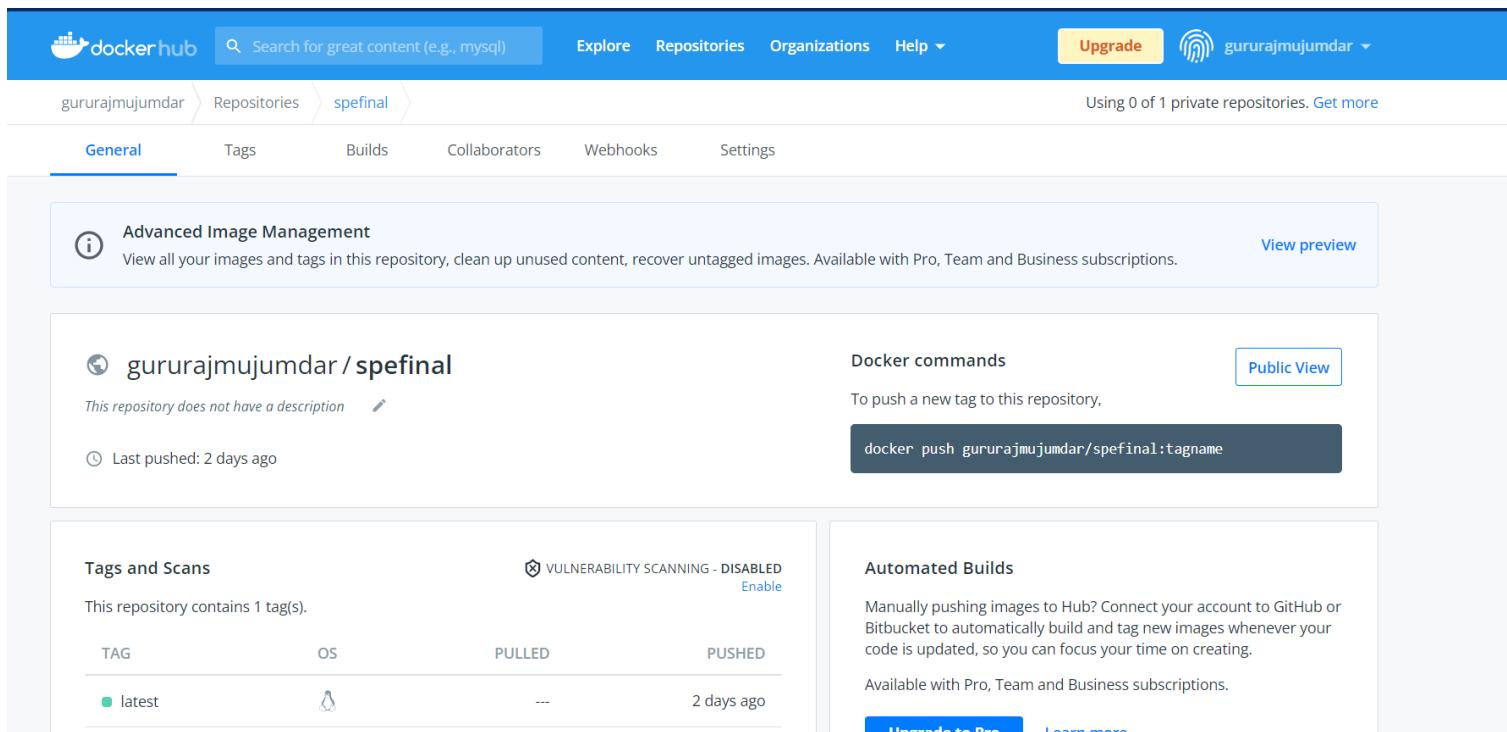
The screenshot shows a code editor with a tab bar at the top containing 'Welcome to Settings Sync', 'test.js', and 'Jenkinsfile'. The 'Jenkinsfile' tab is active, displaying the following Groovy script:

```
1 pipeline {
2     agent any
3     environment{
4         DOCKERHUB_CREDENTIALS=credentials('docker-hub-creds')
5     }
6     options{
7         skipDefaultCheckout(true)
8     }
9     stages {
10         stage('clean workspace'){
11             steps{
12                 |     cleanWs()
13             }
14         }
15         stage('clone repo and install') {
16             steps {
17                 |         git url: 'https://github.com/iashtripathy/SPE.git', branch: 'main'
18                 |         sh "npm install"
19             }
20         }
21     }
22 }
```

The code editor highlights the 'cleanWs()' and 'git' steps with blue underlines, indicating they are part of an external library or suggestion list. The cursor is positioned at the end of the 'sh "npm install"' line.

```
26
27     stage('Testing'){
28         steps{
29             sh "npm test"
30         }
31     }
32     stage('Build docker image'){
33         steps{
34             sh 'docker build -t spefinal:latest .'
35         }
36     }
37     stage('Push to Dockerhub'){
38         steps{
39             echo 'docker tag'
40             sh 'docker tag spefinal gururajmujumdar/spefinal:latest'
41             echo 'docker login'
42             sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
43             echo 'Pushing image to hub'
44             sh 'docker push gururajmujumdar/spefinal'
45             echo 'docker logout'
46             sh 'docker logout'
47         }
48     }
49     stage('Pull image from docker hub'){
50         steps {
51             ansiblePlaybook colorized: true, disableHostKeyChecking: true, installation: 'Ansible', inventory: 'inventory', playbook: 'playbook.yml'
52         }
53     }
54 }
```

1. We can clearly see in our Jenkins file that in the very first step we are telling Jenkins from where to get the docker credentials
2. Following that we first clone the repo where our project is present.
3. We then perform the required tests on the project.
4. Then we build the image of our project.
5. We now push the image to our docker hub. Below is the screenshot of it.



DockerHub Image

6. Finally using ansible we pull the image from docker hub.

Below are screenshots of Jenkins Pipeline and Console output :

Jenkins

Dashboard > SPEFinalProject >

Back to Dashboard Status Changes Build Now Configure Delete Pipeline Full Stage View Rename Pipeline Syntax Build History trend ▾ Filter builds... #20 May 5, 2022, 2:07 PM #19 May 5, 2022, 2:04 PM #18 May 5, 2022, 1:59 PM #17 Apr 22, 2022, 11:01 AM #16 Apr 11, 2022, 4:17 PM #15 Apr 11, 2022, 4:10 PM #14 Apr 11, 2022, 3:57 PM #13 Apr 11, 2022, 3:49 PM #12 Apr 11, 2022, 2:59 PM #11 Apr 11, 2022, 2:33 PM

Pipeline SPEFinalProject

Recent Changes

Stage View

	clean workspace	clone repo and install	Testing	Build docker image	Push to Dockerhub	Pull image from docker hub
Average stage times: (Average full run time: ~3min 57s)	1s	19s	3s	1min 22s	49s	24s
#20 May 05 14:07 1 commit	756ms	8s	2s	58s	43s	25s
#19 May 05 14:04 No Changes	835ms	10s	2s	25s	32s	28s
#18 May 05 13:59 2 commits	1s	23s	2s	1min 24s	36s	50s
#17 Apr 22 11:01 2 commits	3s	1min 24s	16s	7min 28s	1min 41s	50s
#16 Apr 11 16:17 1 commit	837ms	6s	3s	23s	44s	8s

Jenkins

Dashboard > SPEFinalProject > #20

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build '#20' Git Build Data Restart from Stage Replay Pipeline Steps Workspaces Previous Build

Console Output

Started by user Iash Kumar Tripathy
Obtained Jenkinsfile from git <https://github.com/iashtripathy/SPE.git>
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/SPEFinalProject
[Pipeline]
[Pipeline] withCredentials
Masking supported pattern matches of \$DOCKERHUB_CREDENTIALS or \$DOCKERHUB_CREDENTIALS_PSW
[Pipeline]
[Pipeline] stage
[Pipeline] { (clean workspace)
[Pipeline] cleanWS
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (clone repo and install)
[Pipeline] git
The recommended git tool is: git
No credentials specified
Cloning the remote Git repository <https://github.com/iashtripathy/SPE.git>
> git init /var/lib/jenkins/workspace/SPEFinalProject # timeout=10
Fetching upstream changes from <https://github.com/iashtripathy/SPE.git>
> git --version # timeout=10
> git -v version 2.25.1'
> git fetch -tags --force --progress -- <https://github.com/iashtripathy/SPE.git> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url <https://github.com/iashtripathy/SPE.git> # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^~(commit) # timeout=10
Checking out Revision 32833a9a92bffddea0469504e8f5302256c603c7 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 32833a9a92bffddea0469504e8f5302256c603c7 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b main 32833a9a92bffddea0469504e8f5302256c603c7 # timeout=10

Dashboard > SPEFinalProject > #20

```
Checking out Revision 32833a9a92bffddea0469504e8f5302256c603c7 (refs/remotes/origin/main)
> git config core.sparseCheckout # timeout=10
> git checkout -f 32833a9a92bffddea0469504e8f5302256c603c7 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b main 32833a9a92bffddea0469504e8f5302256c603c7 # timeout=10
Commit message: "Update package.json"
> git rev-list --no-walk 0af8f80bb612844b71808171d581a0bd5e24a457 # timeout=10
[Pipeline] sh
+ npm install
npm WARN deprecated formidable@1.2.6: Please upgrade to latest, formidable@v2 or formidable@v3! Check these notes: https://bit.ly/2ZEQIAu
npm WARN deprecated superagent@3.8.3: Please upgrade to v7.0.2+ of superagent. We have fixed numerous issues with streams, form-data, attach(), filesystem errors not bubbling up (ENOENT on attach()), and all tests are now passing. See the releases tab for more information at <https://github.com/visionmedia/superagent/releases>.

added 281 packages, and audited 282 packages in 3s

27 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (2 moderate, 3 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Dashboard > SPEFinalProject > #20

```
+ npm test
> spe@1.0.0 test
> mocha --exit

Server started on port 3000

/ home route
  ✓ it should check the home page (51ms)

/ adoption route
  ✓ it should check the adoption page

/ report-homeless route
  ✓ it should Report Homeless People route

/ donation route
  ✓ it should check the donation page

/ Admin Login route
  ✓ It checks Admin Login page

/ Admin Logout route
{
  host: '127.0.0.1:40175',
  'accept-encoding': 'gzip, deflate',
  'user-agent': 'node-supertest/3.8.3',
  connection: 'close'
}
(node:215446) [DEP0066] DeprecationWarning: OutgoingMessage.prototype._headers is deprecated
(Use `node --trace-deprecation ...` to show where the warning was created)
  ✓ It checks admin Logout

6 passing (116ms)
```

Dashboard > SPEFinalProject > #20

```

[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build docker image)
[Pipeline] sh
+ docker build -t spefinal:latest .
Sending build context to Docker daemon 65.98MB

Step 1/7 : FROM node:17-alpine
--> dca41997cd4
Step 2/7 : WORKDIR /usr/src/app
--> Using cache
--> fc63d48d34a3
Step 3/7 : COPY package.json .
--> d6f85fb3cf84
Step 4/7 : RUN npm install
--> Running in 3d69b56ed363
[1mWARN deprecate[d@1.2.6: Please upgrade to latest, formidable@v2 or formidable@v3! Check these notes: https://bit.ly/2ZEqlau
[0m[1mWARN deprecate[d@1.2.3: Please upgrade to v7.0.2+ of superagent. We have fixed numerous issues with streams, form-data, attach(), filesystem errors not bubbling up (ENOENT on attach()), and all tests are now passing. See the releases tab for more information at <https://github.com/visionmedia/superagent/releases>.
[0m
added 270 packages, and audited 271 packages in 15s

26 packages are looking for funding
  run `npm fund` for details

2 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
[1mnotice New minor version of npm available! 8.5.5 -> 8.9.0
[0m[1mnotice Changelog: <https://github.com/npm/cli/releases/tag/v8.9.0>
[0m[1mnotice Run `npm install -g npm@8.9.0` to update!
[0m[1mnotice[0m
[0m[0mRemoving intermediate container 3d69b56ed363
--> 984abb4a98e
Step 5/7 : COPY .
--> 9b5497da12e3
Step 6/7 : EXPOSE 3000
--> Running in 924b59616b91
Removing intermediate container 924b59616b91
--> 061debaba58d

```

Dashboard > SPEFinalProject > #20

```

Removing intermediate container 924b59616b91
--> 061debaba58d
Step 7/7 : CMD ["node","app.js"]
--> Running in bed4fcf79c34
Removing intermediate container bed4fcf79c34
--> 156023e9ae88
Successfully built 156023e9ae88
Successfully tagged spefinal:latest
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push to Dockerhub)
[Pipeline] echo
docker tag
[Pipeline] sh
+ docker tag spefinal gururajmujumdar/spefinal:latest
[Pipeline] echo
docker login
[Pipeline] sh
+ docker login -u gururajmujumdar --password-stdin
+ echo ****
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] echo
Pushing image to hub
[Pipeline] sh
+ docker push gururajmujumdar/spefinal
Using default tag: latest
The push refers to repository [docker.io/gururajmujumdar/spefinal]
57c7077c8e00: Preparing
3787d9008a58: Preparing
b53c8d627bc6: Preparing
e28188451ac: Preparing
71aa656b6aa1: Preparing
30a116ea8c57: Preparing
f909274da3b9: Preparing
4fc242d58285: Preparing
30a116ea8c57: Waiting
f909274da3b9: Waiting
4fc242d58285: Waiting
e28188451ac: Layer already exists
71aa656b6aa1: Layer already exists

```

Dashboard > SPEFinalProject > #20

```
latest: digest: sha256:68125cteeac4abeetc8a9548e9et10383/cd4tt06/c430t2a9de1t9509aa33ee size: 1996
[Pipeline] echo
docker logout
[Pipeline] sh
+ docker logout
Removing login credentials for https://index.docker.io/v1/
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Pull image from docker hub)
[Pipeline] ansiblePlaybook
[SPEFinalProject] $ ansible-playbook playbook.yml -i inventory

PLAY [Pull docker image from docker hub] *****

TASK [Gathering Facts] *****
: [0;32mok: [3.109.153.193]:[0m
: [0;32m:[0m
TASK [stop container] *****
: [0;33mchanged: [3.109.153.193]:[0m
: [0;33m:[0m
TASK [remove container] *****
: [0;33mchanged: [3.109.153.193]:[0m
: [0;33m:[0m
TASK [deleting image] *****
: [0;33mchanged: [3.109.153.193]:[0m
: [0;33m:[0m
TASK [pull new image] *****
: [0;33mchanged: [3.109.153.193]:[0m
: [0;33m:[0m
TASK [run image] *****
: [0;33mchanged: [3.109.153.193]:[0m
: [0;33m:[0m
PLAY RECAP *****
: [0;33m3.109.153.193:[0m : [0;32mok=6 [0m [0;33mchanged=5 [0m unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4) Conclusion:

This project is automatically built, tested and deployed with CI/CD pipeline. Various DevOps tools are used in this project. It is useful to understand the whole DevOps pipeline easily. Continuous deployment is done by Ansible. Docker containers are used to deploy this project which is very light weight compared to a virtual environment.

5) Scope for future work:

There are several other features that can be added to this. Currently it is limited to only a few features.

6) Appendix:

- **Operating system used:** Ubuntu version 18.04 LTS
- **CPU used:** 4 cores (oracle virtualbox)
- **RAM:** 8 GB
- **Kernel version:** 4.15.0-101-generic
- **Languages used:** HTML,CSS,JavaScript,Bootstrap,Node Js,Mongodb,Express

References

1. "Installing Jenkins"
<https://www.jenkins.io/doc/book/%20installing>
2. "Installing on Ubuntu or Debian Linux distributions"
<https://docs.rundeck.com/docs/administration/install/linux-deb.html>
3. "Install Docker Engine on Ubuntu"
<https://docs.docker.com/engine/install/ubuntu/>
4. "Install Elasticsearch with Docker"
<https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html>
5. Richard Umoffia "Visualizing Logs from a Dockerized Node Application Using the Elastic Stack"
<https://blog.soshace.com/visualizing-logs-from-a-dockerized-node-application-using-the-elastic-stack/>
6. Josh Reichardt "Setting up a GitHub webhook in Jenkins"
<https://thepracticalsysadmin.com/setting-up-a-github-webhook-in-jenkins/>
7. "Dockerizing a Node.js web app"
<https://nodejs.org/fr/docs/guides/nodejs-docker-webapp/>
8. Brent Laster "Introduction to writing pipelines-as-code and implementing DevOps with Jenkins 2"
<https://opensource.com/article/18/8/devops-jenkins-2>

Important Links:

GitHub link:

<https://github.com/iashtripathy/SPE.git>

Docker Hub:

[https://hub.docker.com/repository/docker/gururajmuju
mdar/spefinal](https://hub.docker.com/repository/docker/gururajmuju/mdar/spefinal)