**Build a Plasma Donor App**

**With API**

**And deploy it to Kubernetes**

**Final Project-Team A**

**From:**

**PRANAV B (18MIS0072)**

**KISHORE D (18MIS0154)**

**PRITHIVRAJ S (18MIS0320)**

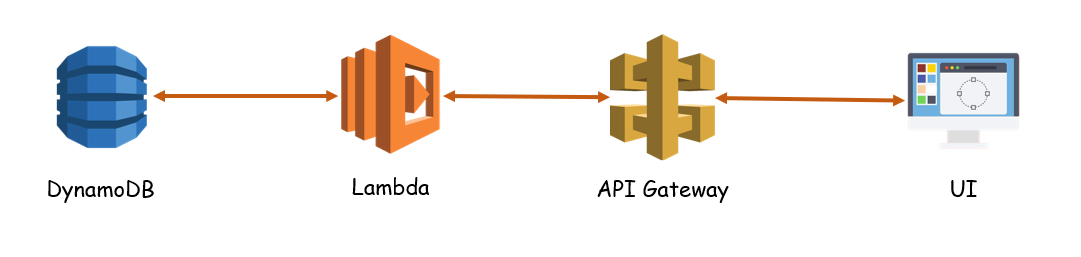
**BIPIN K (18MIS0338)**

**AMARTYA GUNJAN (18BEI0077)**

During the COVID 19 crisis, the requirement of plasma became high and the donor count being low. Saving the donor information and helping the need by notifying the current donors would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details store it and inform them upon a request.

Serverless computing is the current trend of software application development. Micro services are a popular new approach for building maintainable, scalable, cloud-based applications. AWS is the perfect platform for hosting micro-services. In this project we will be building a plasma donor app with AWS services like lambda functions, API gateway and DynamoDB.

**Architecture:**



**Learning Outcomes:**

By the end of this project:

* You’ll be able to work DynamoDB, lambda functions and API Gateway.
* Build a flask application which will take the user inputs, update the DynamoDB and notify the user upon request.

**Project Work Flow:**

* User interacts with the application.
* Registers by giving the details as a donor.
* Database will have all the details and if a user posts a request then the concerned blood group donors will get notified about it.

**Tasks:**

1. Required initial steps
2. Sign in to AWS console
3. Create a database
4. Create APIs to

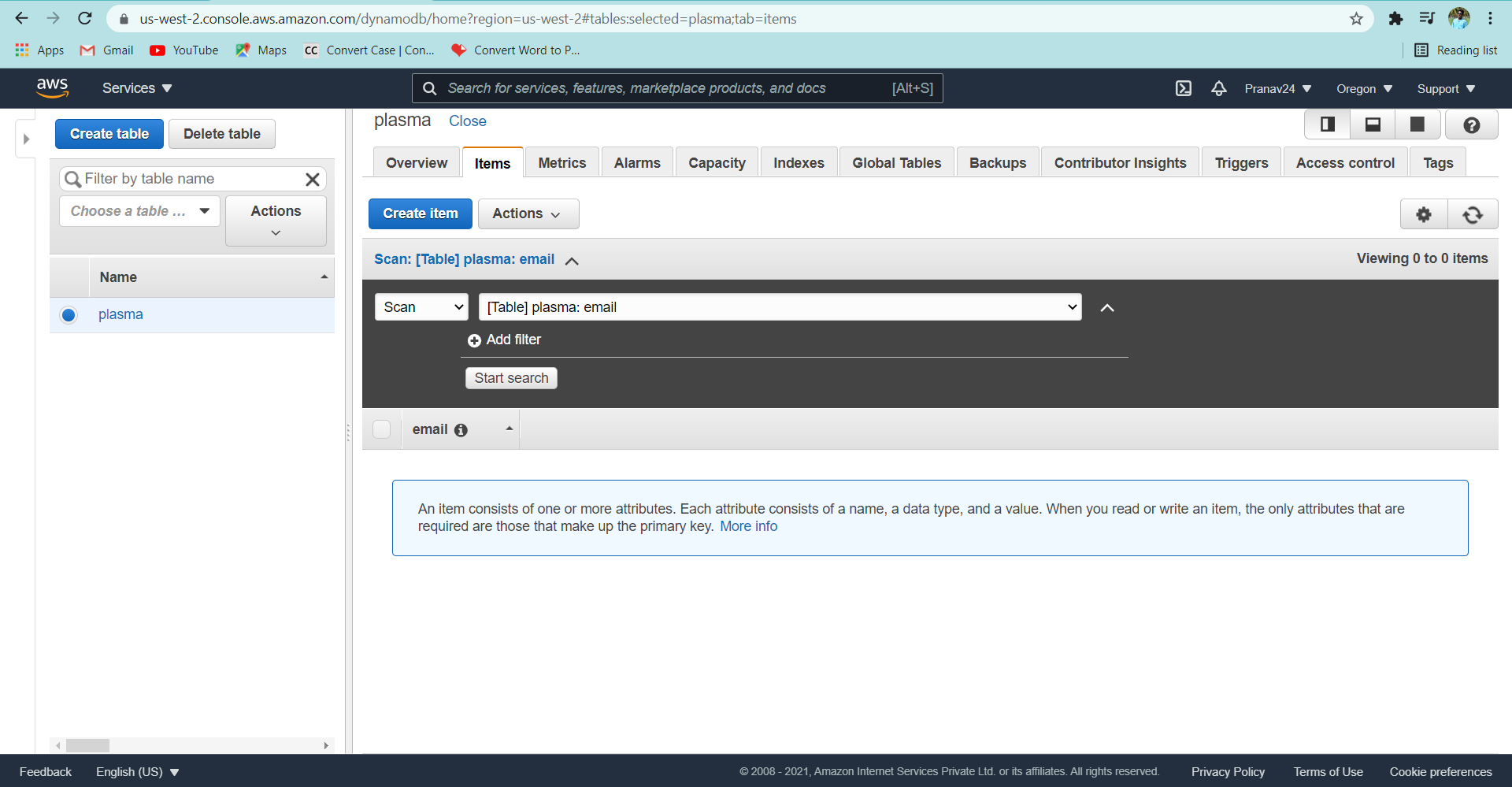
* Push the registration data into database.
* Fetch the data upon login.
* Display the stats of different blood groups.
* Upon request, take the required blood group and return the details

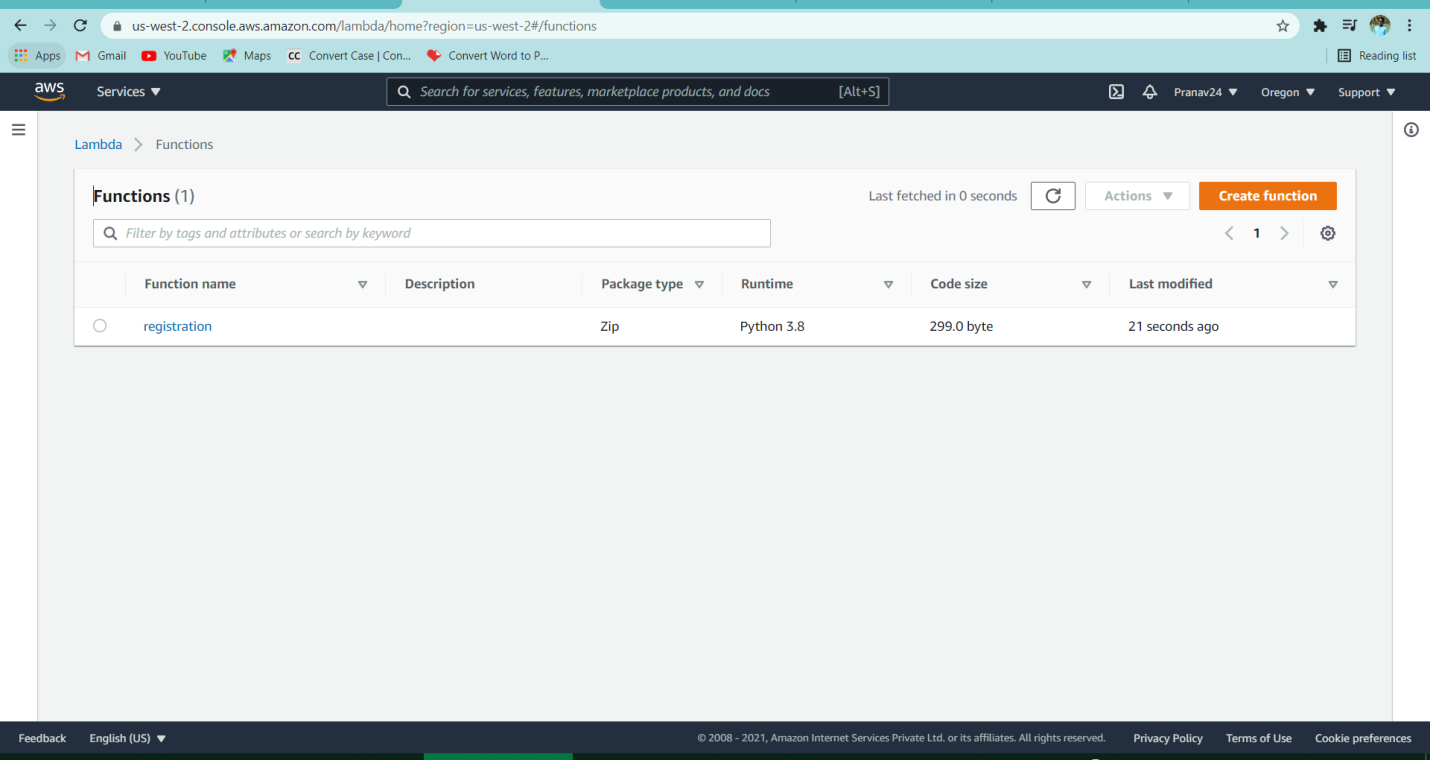
1. Create a flask application

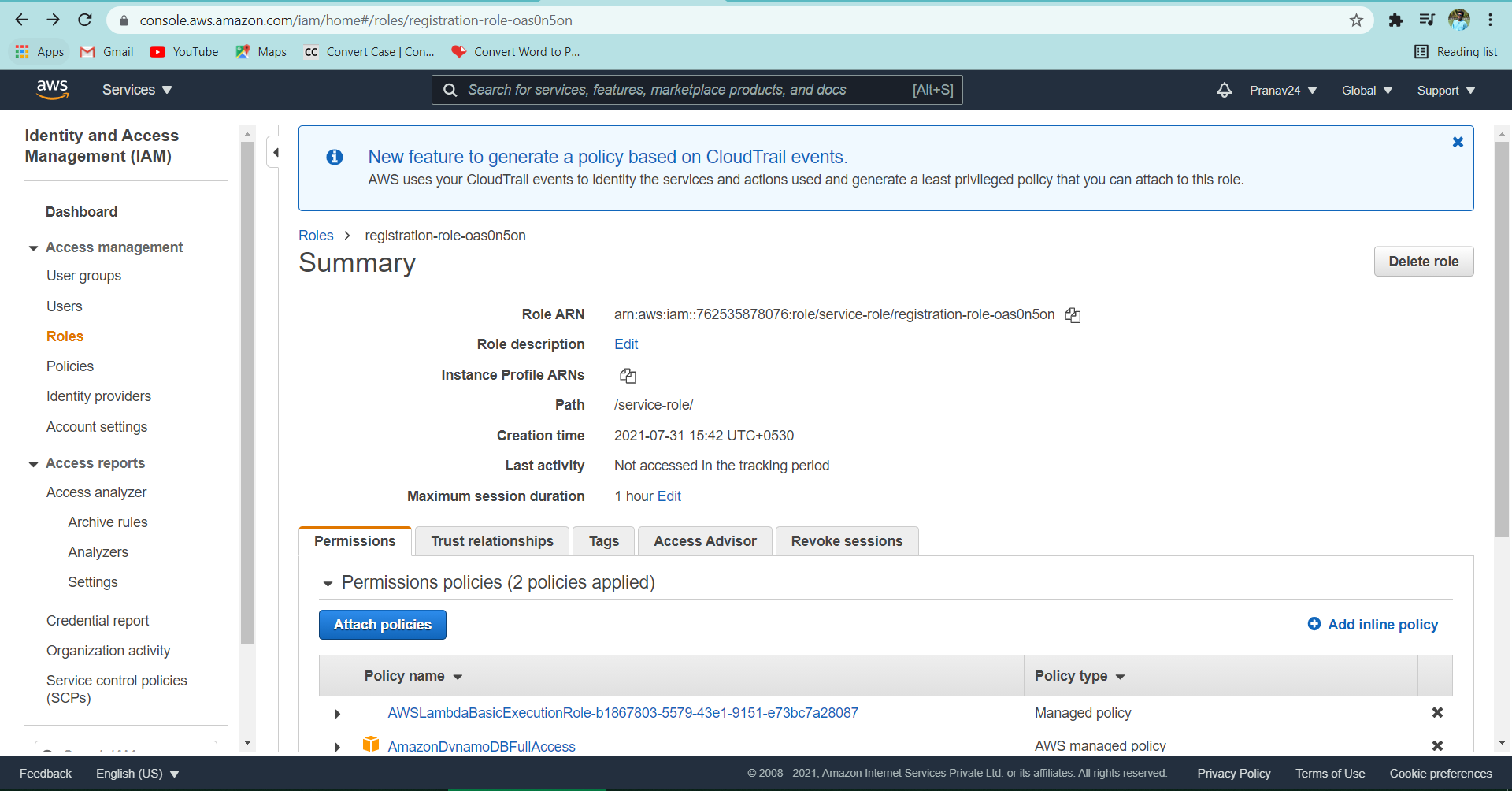
* Registration page
* Login page
* Stats page to display the count
* Request page
* Test it

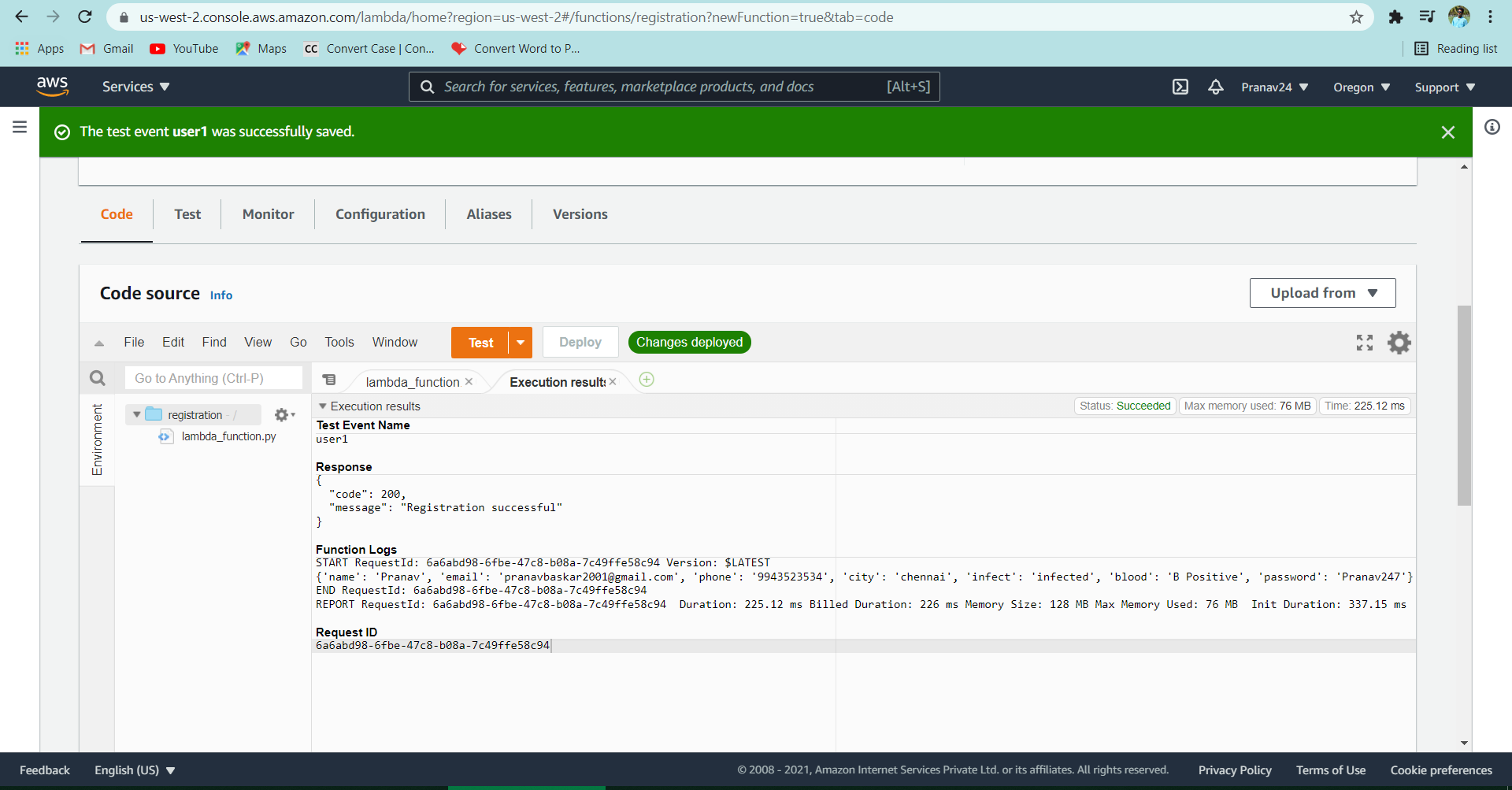
1. Deploy the application

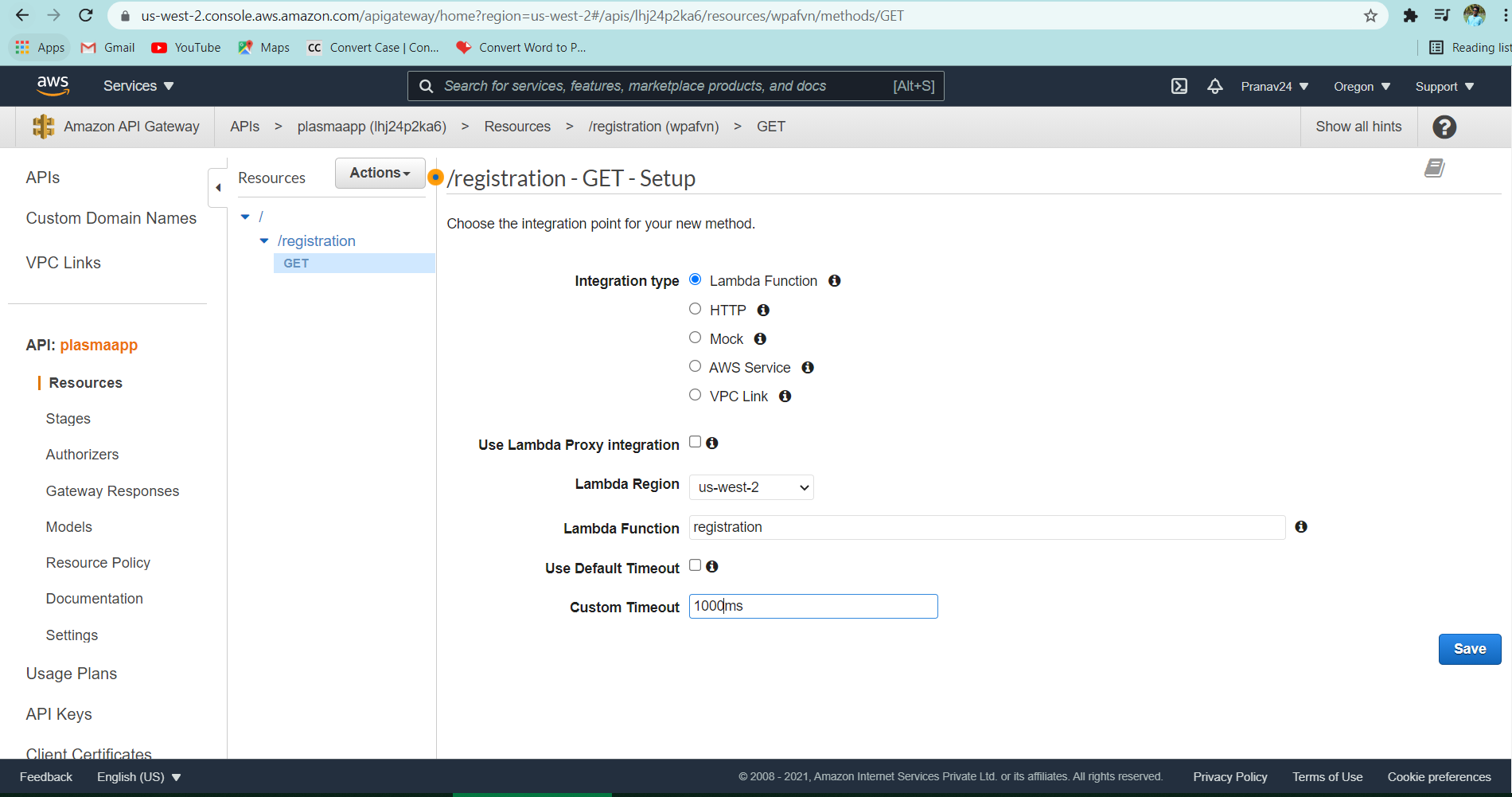
* Create an EC2 instance.
* Deploy the app.

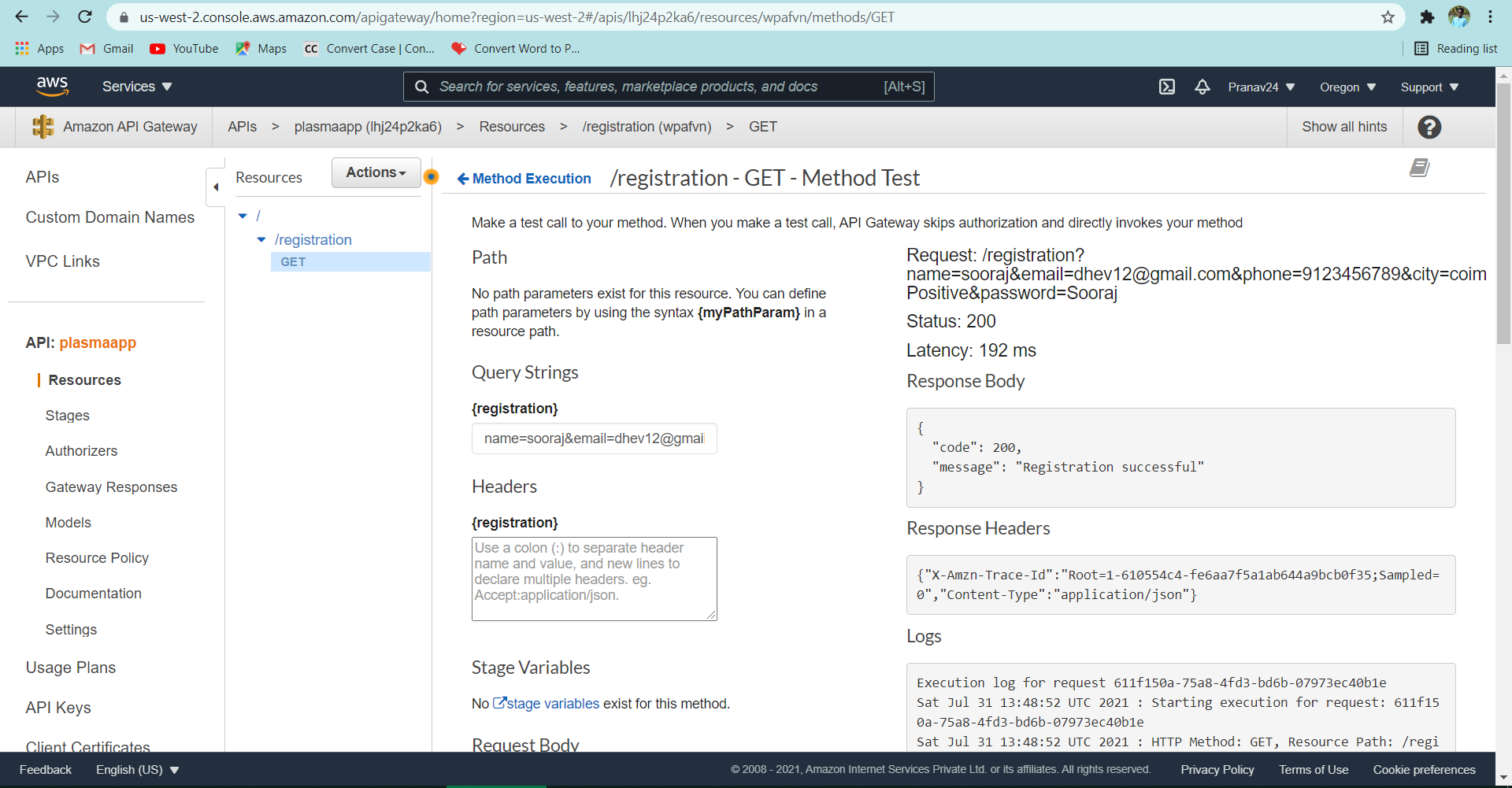


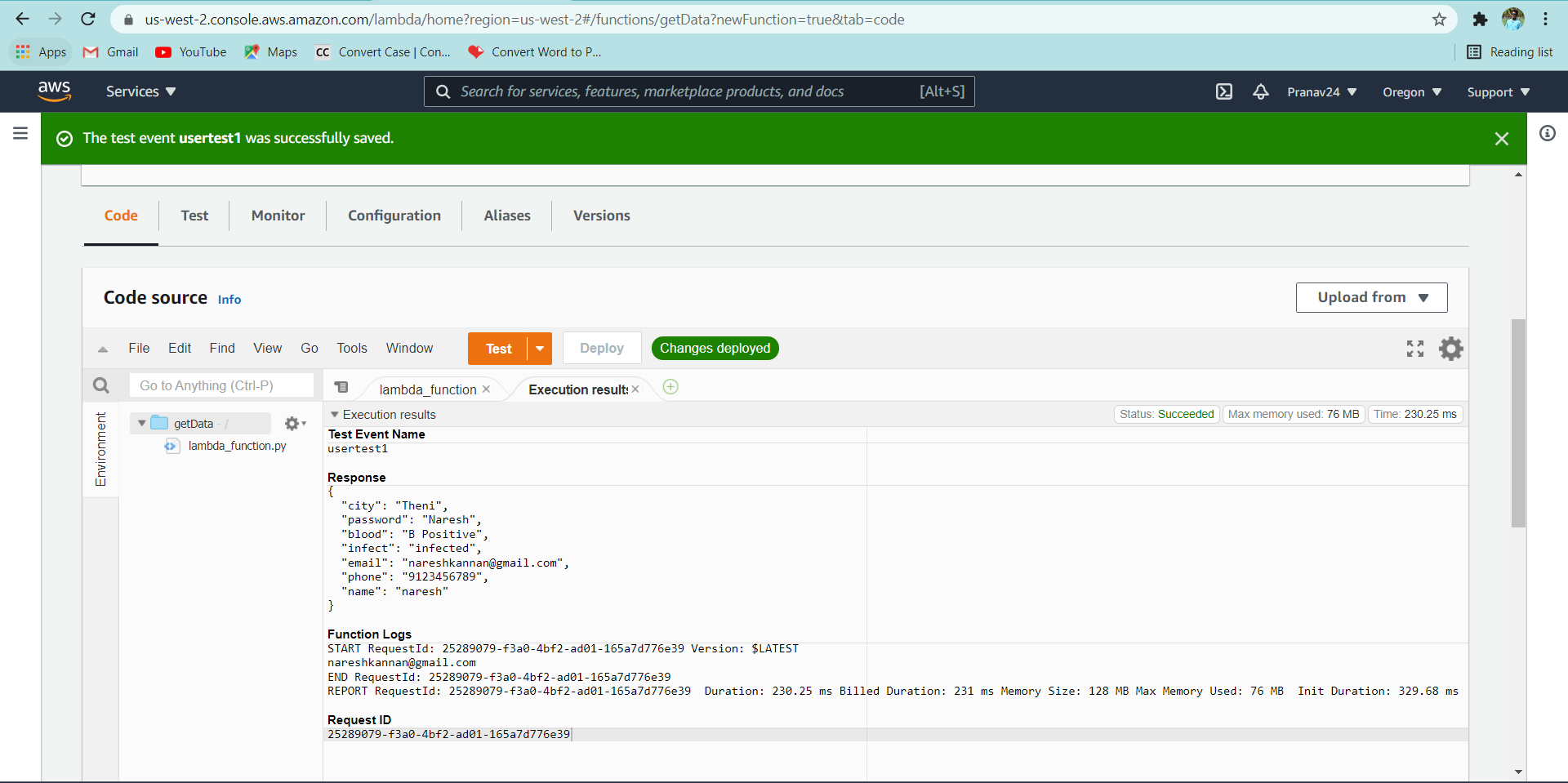


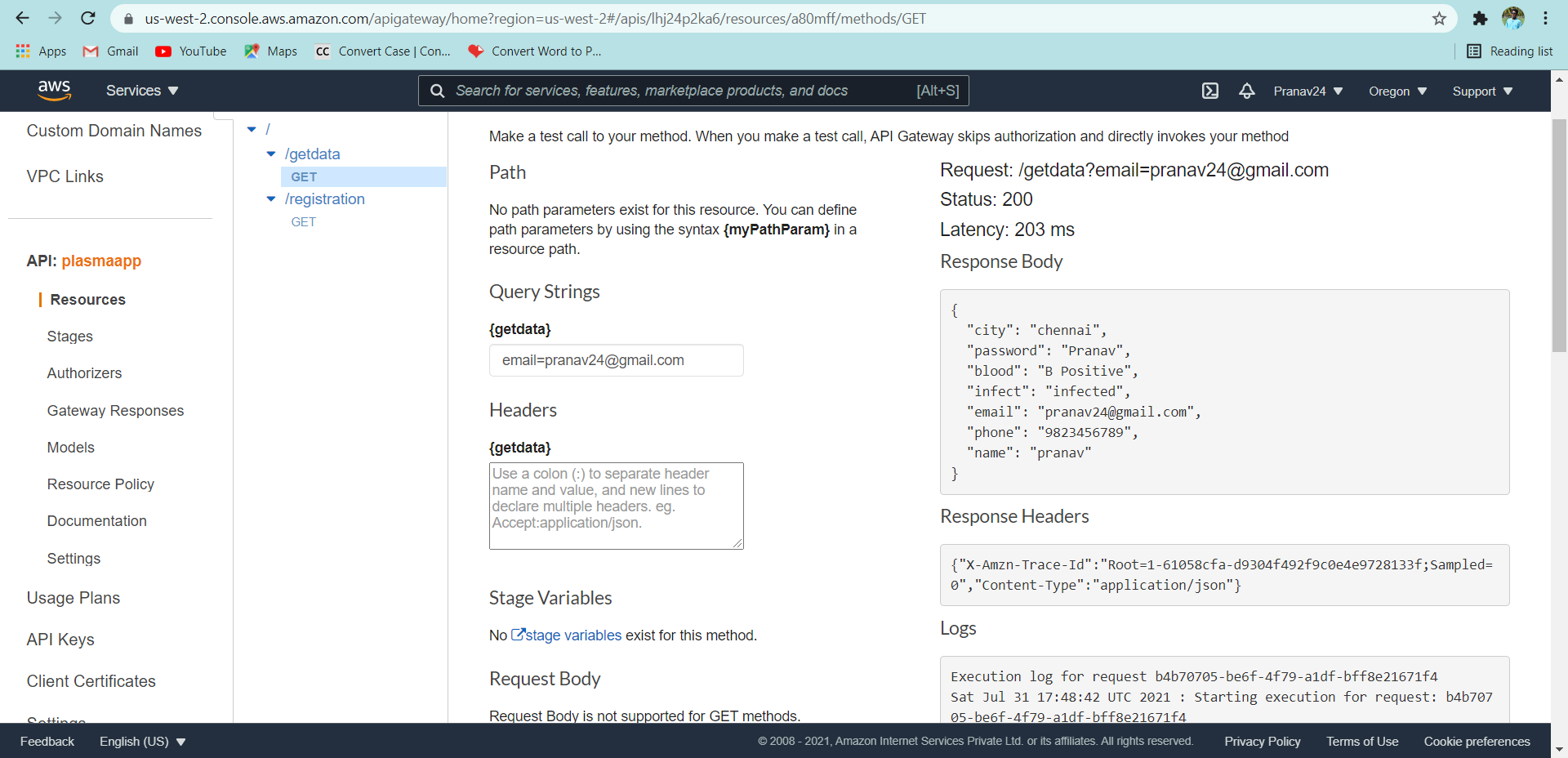


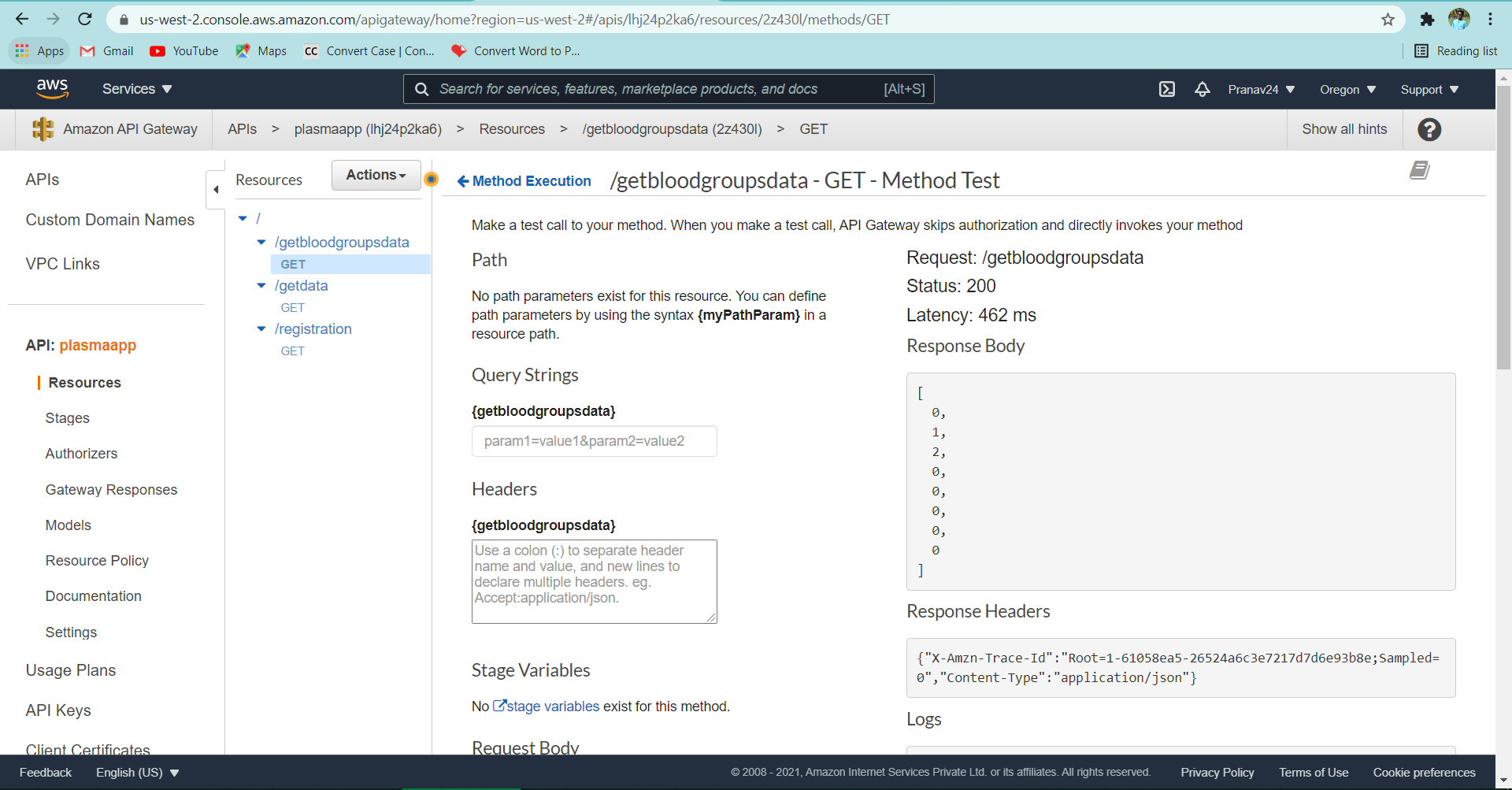


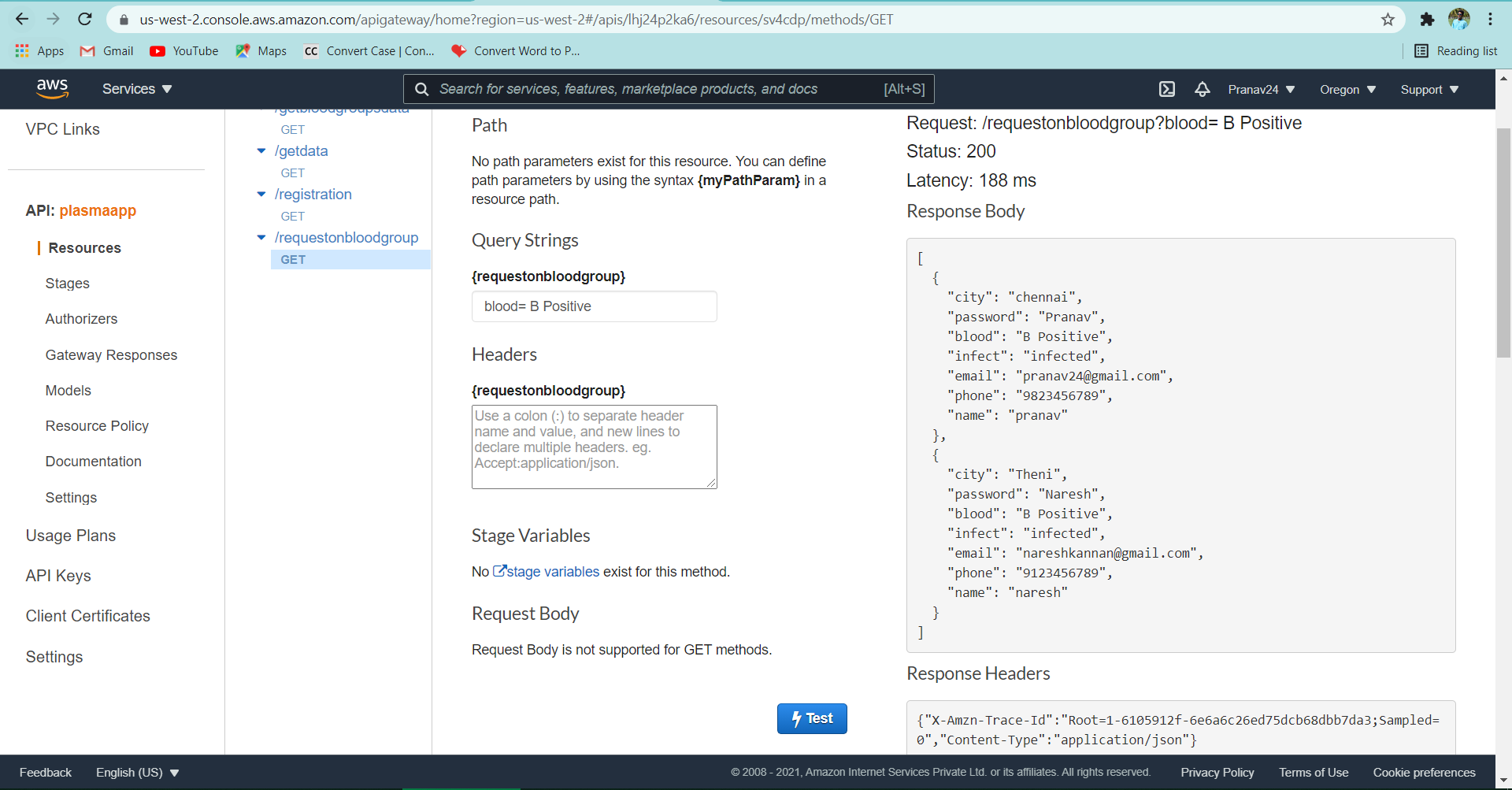




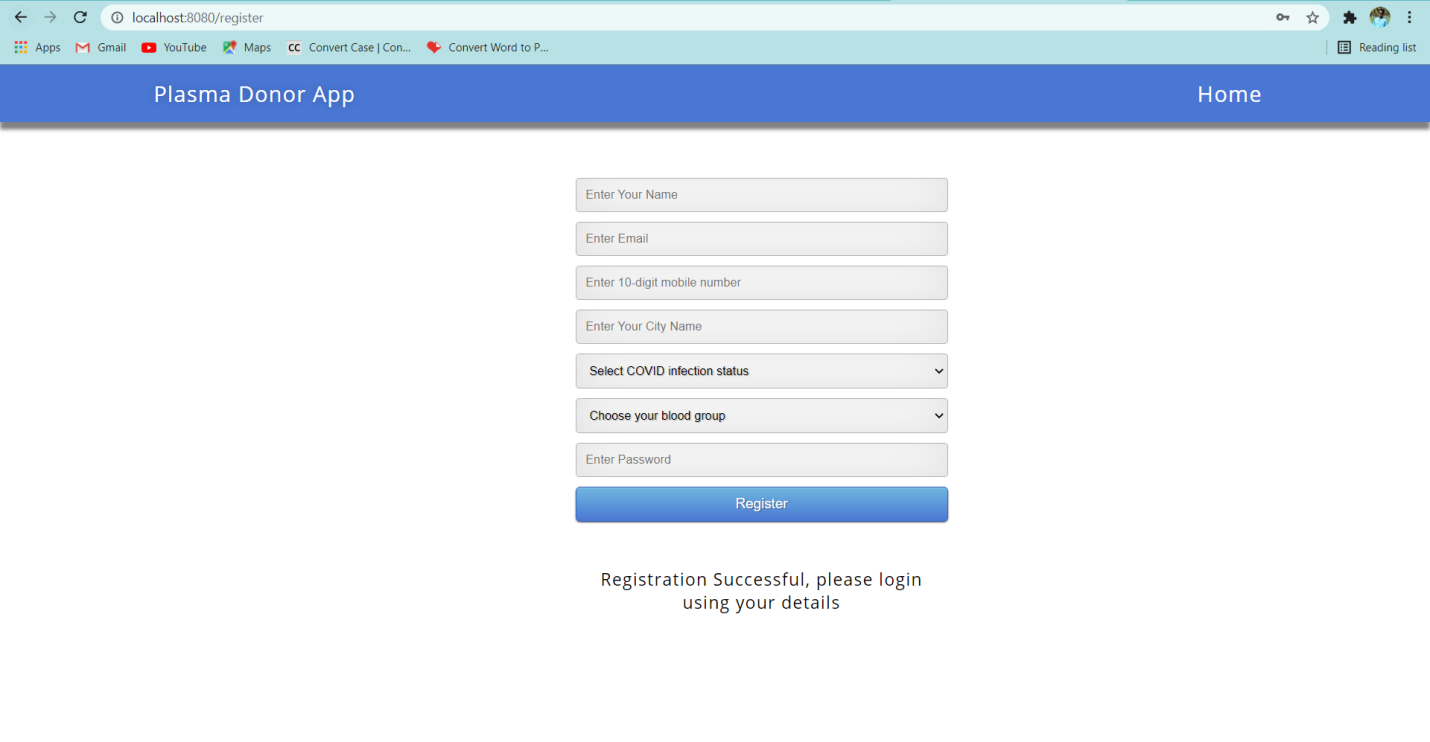


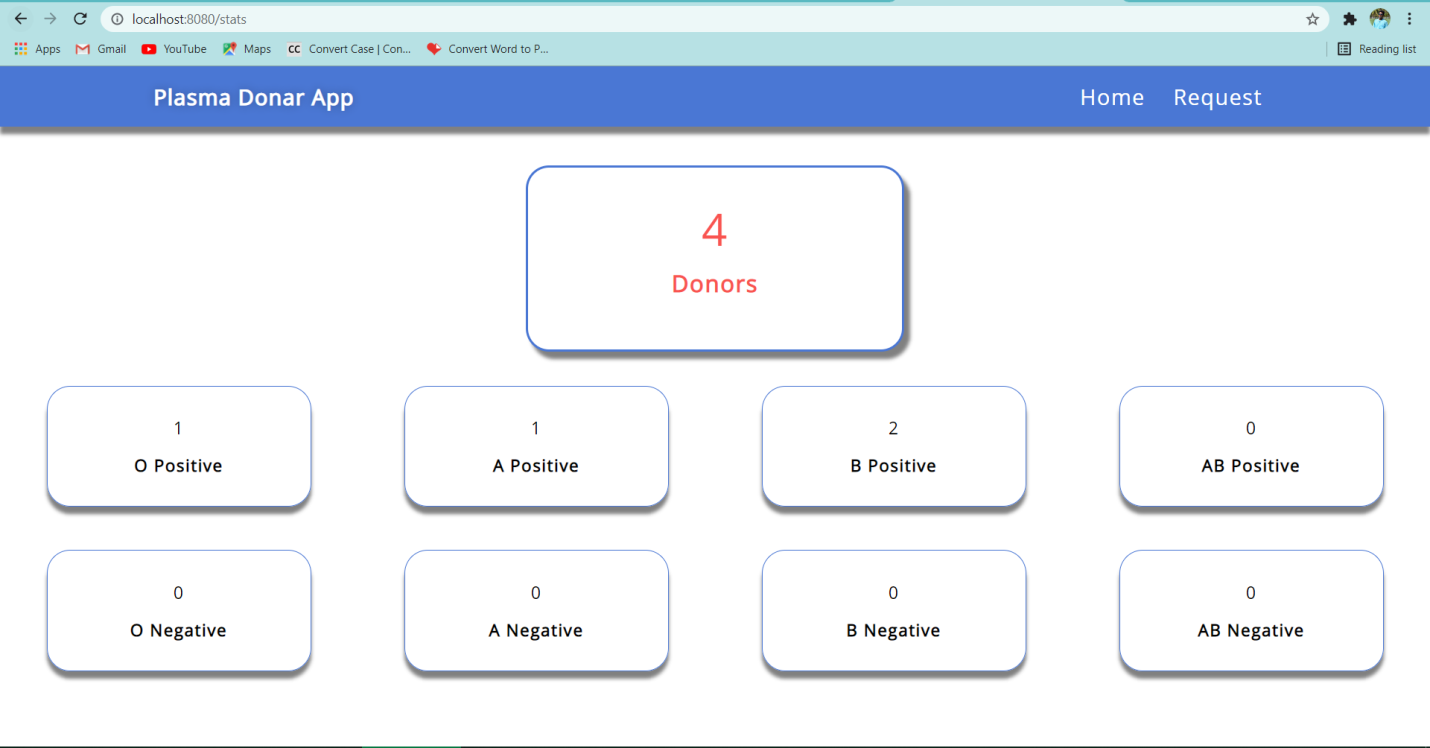


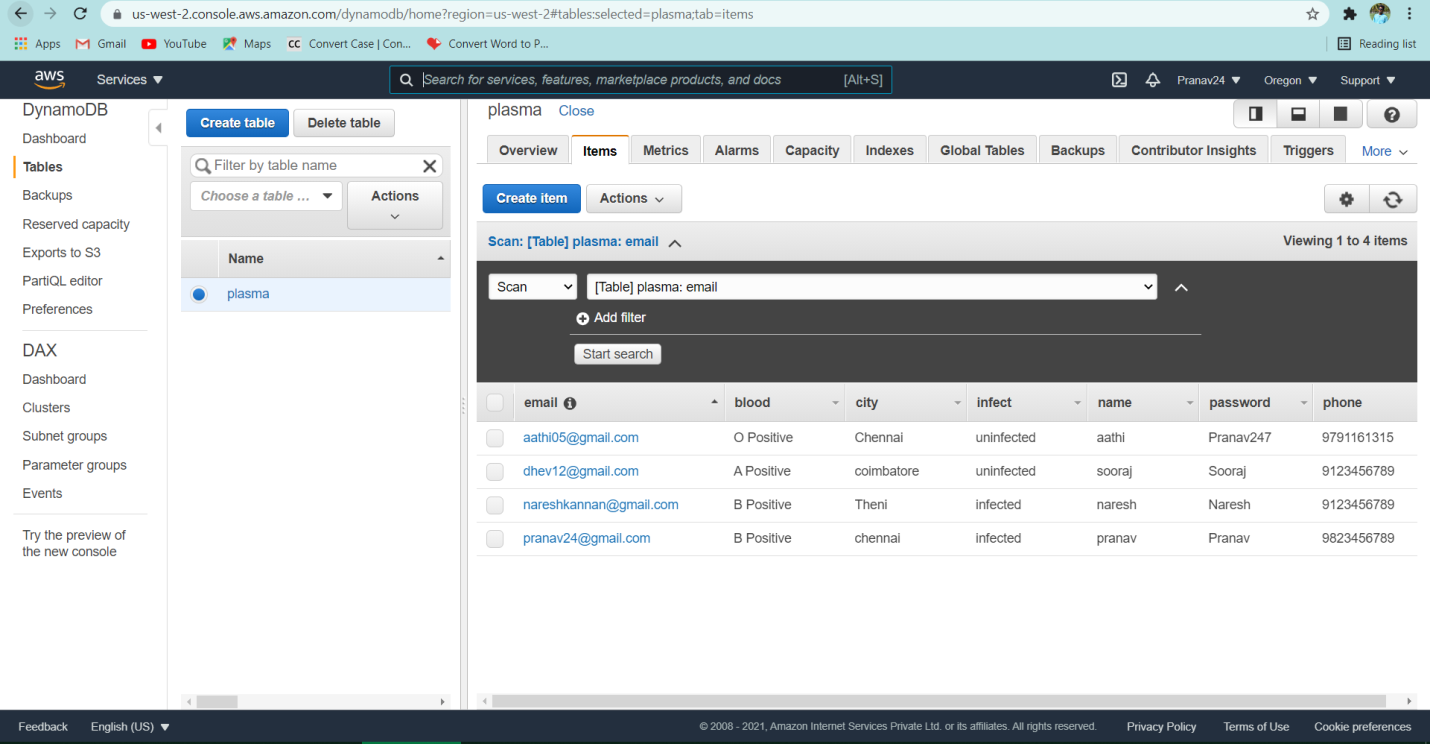


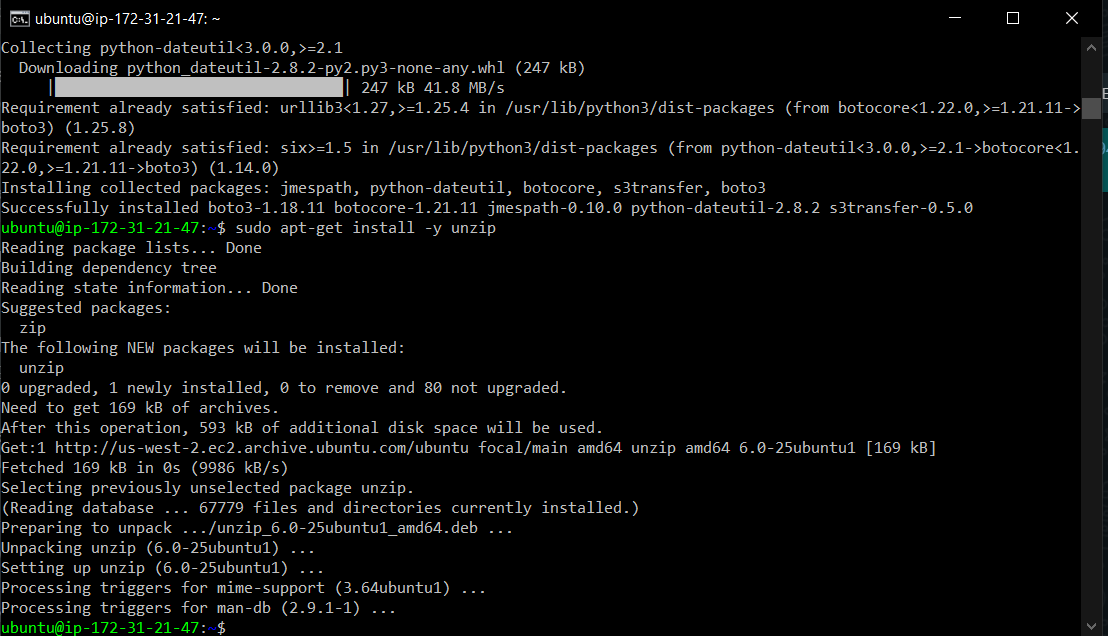


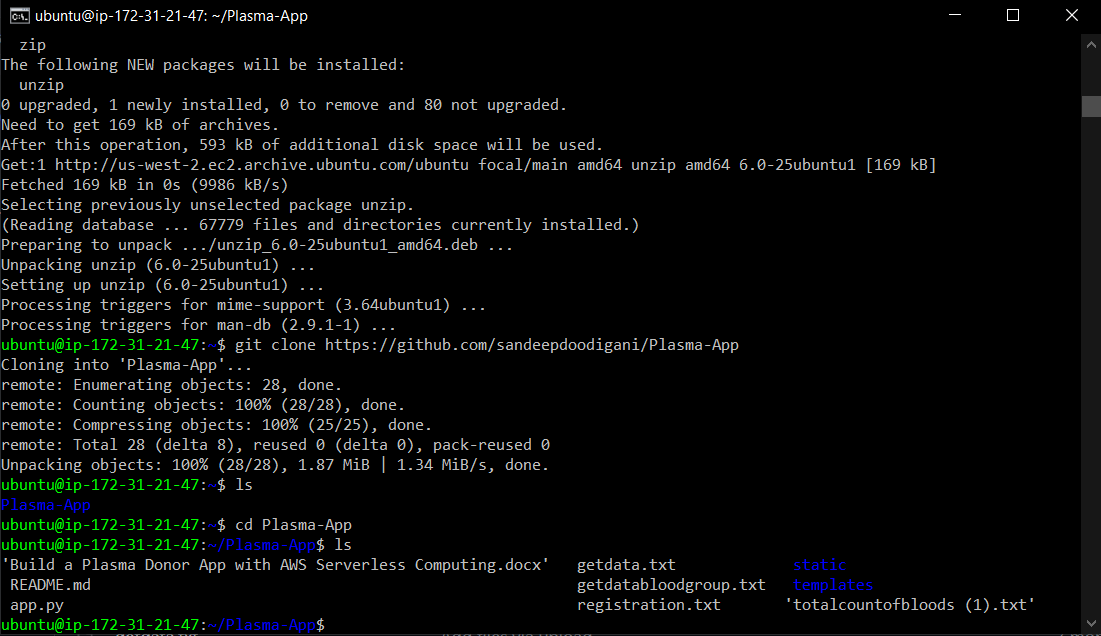


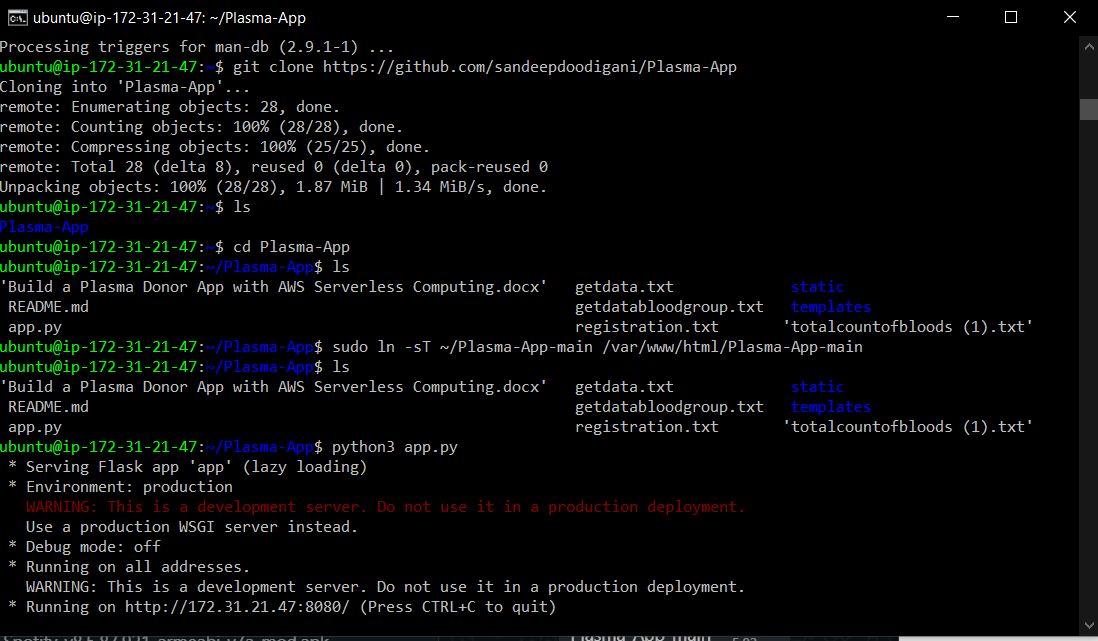


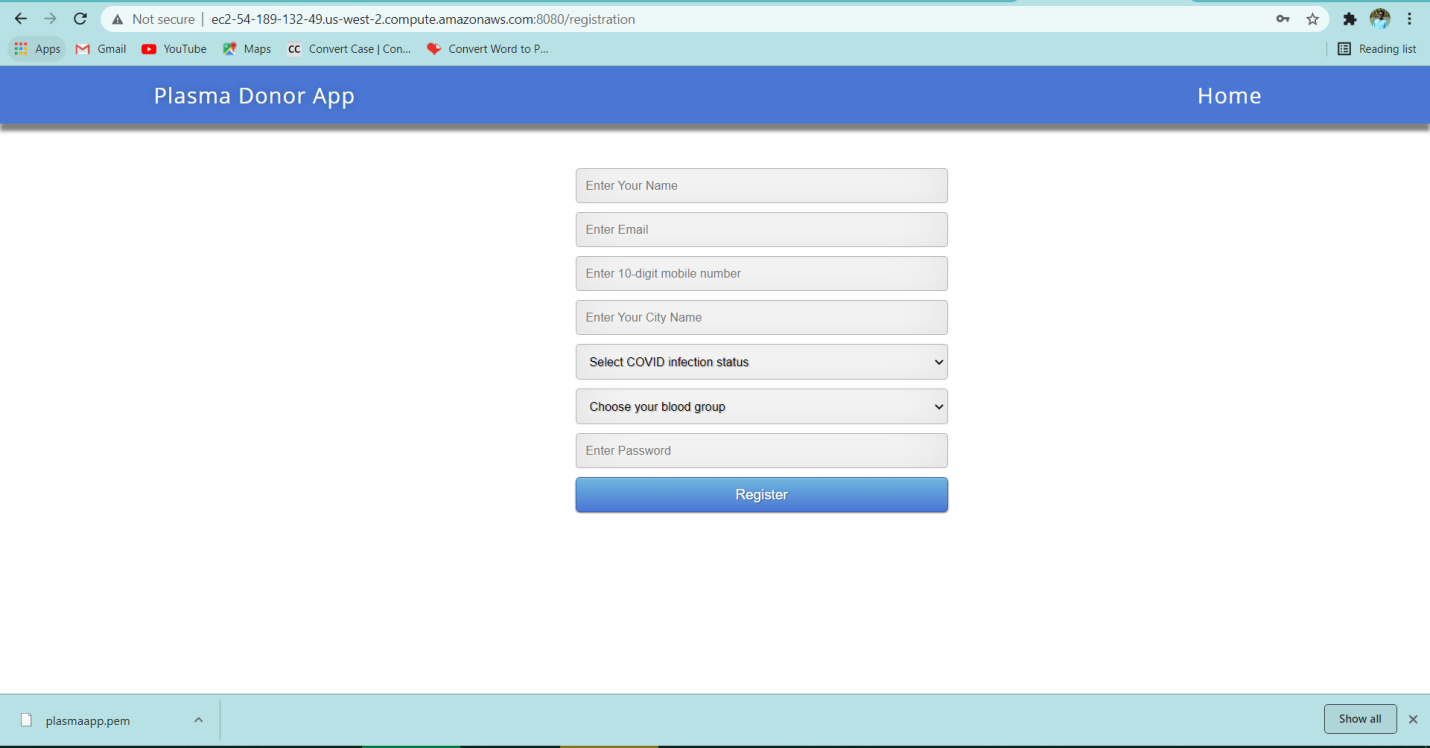


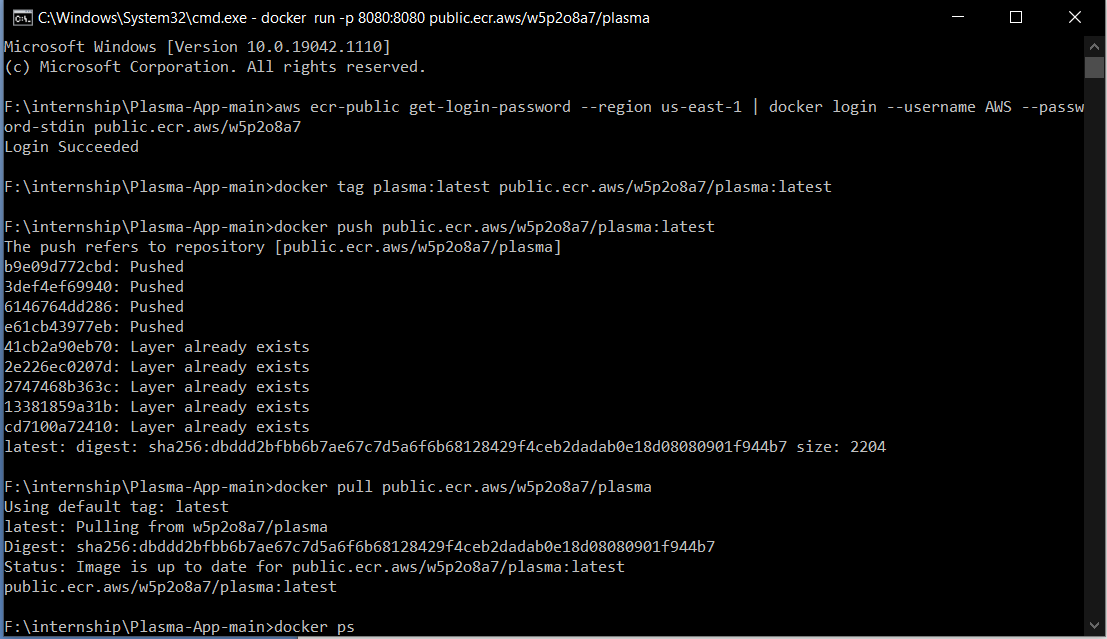


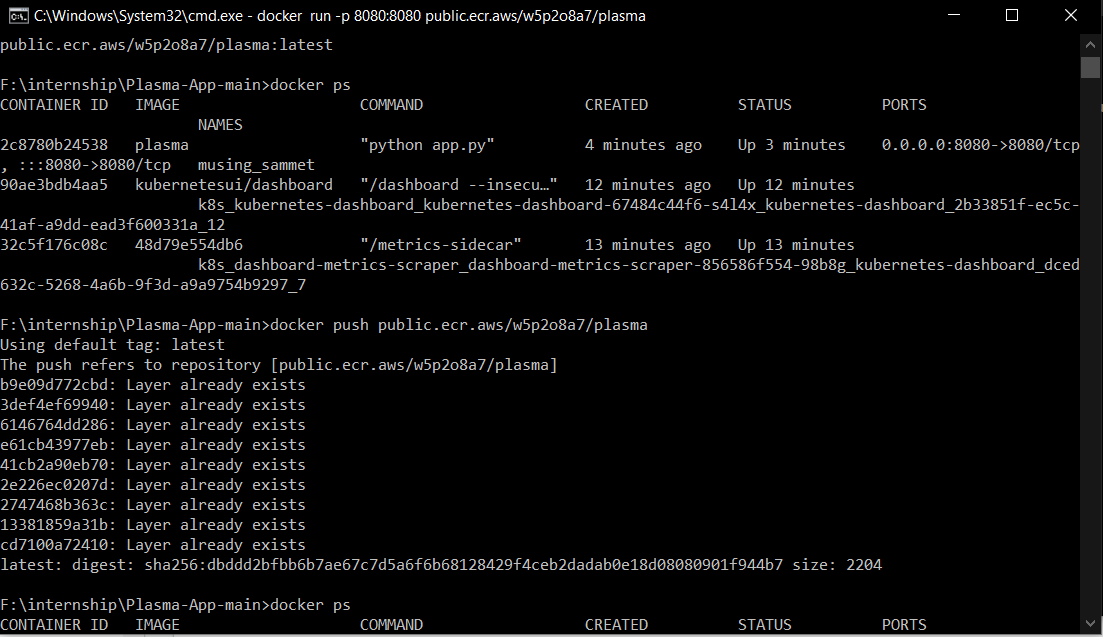




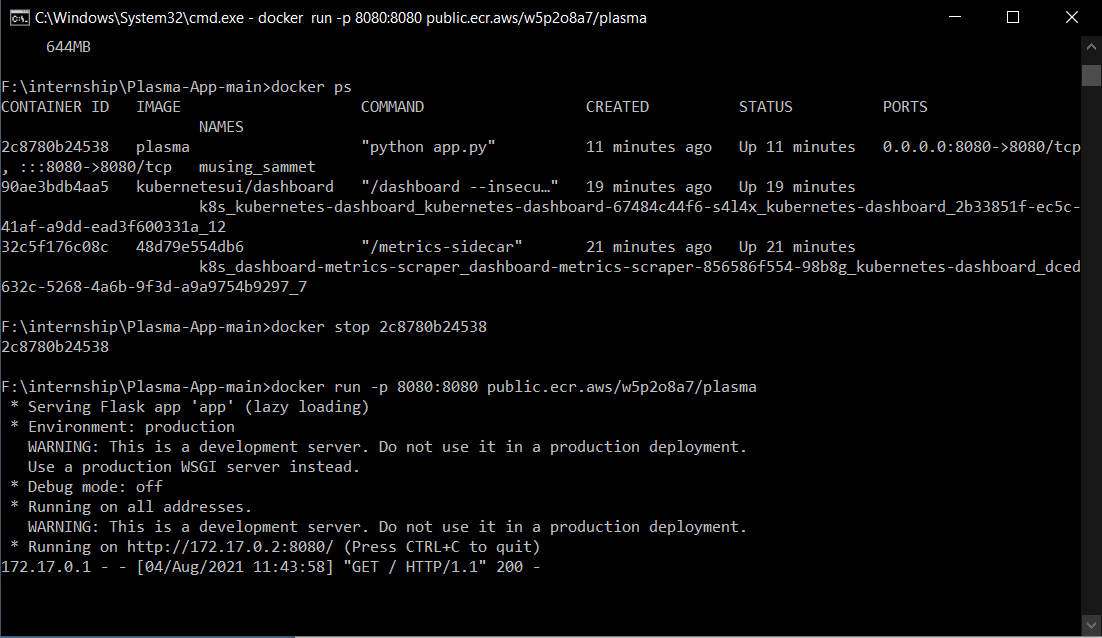


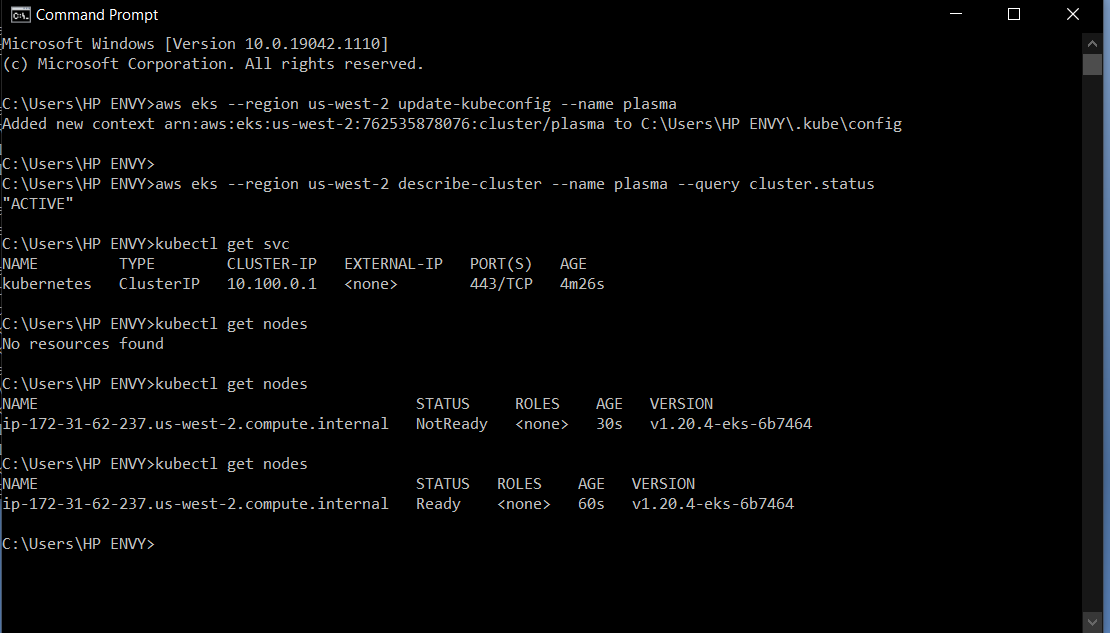


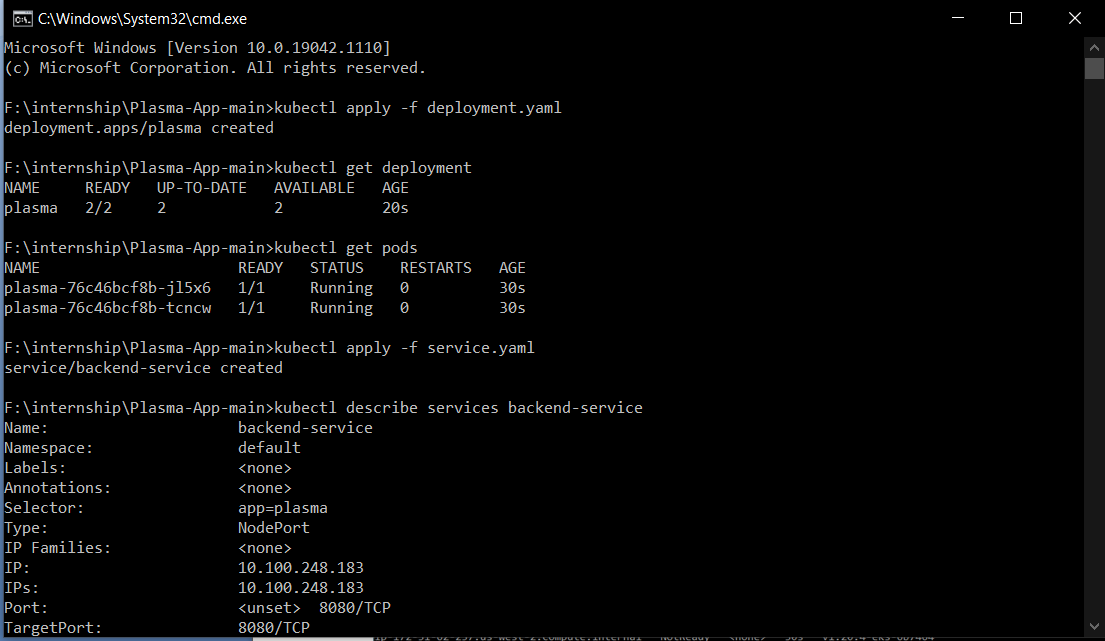


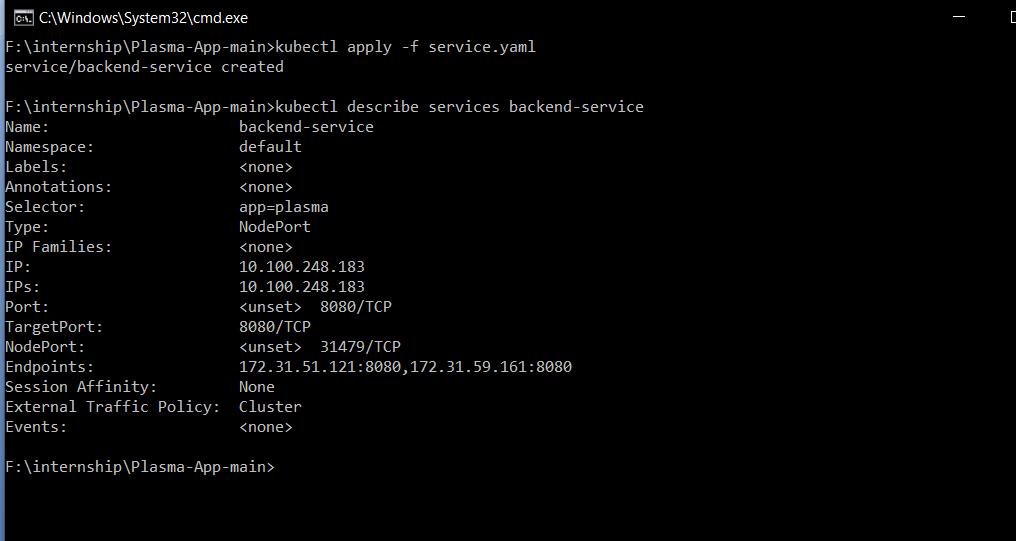


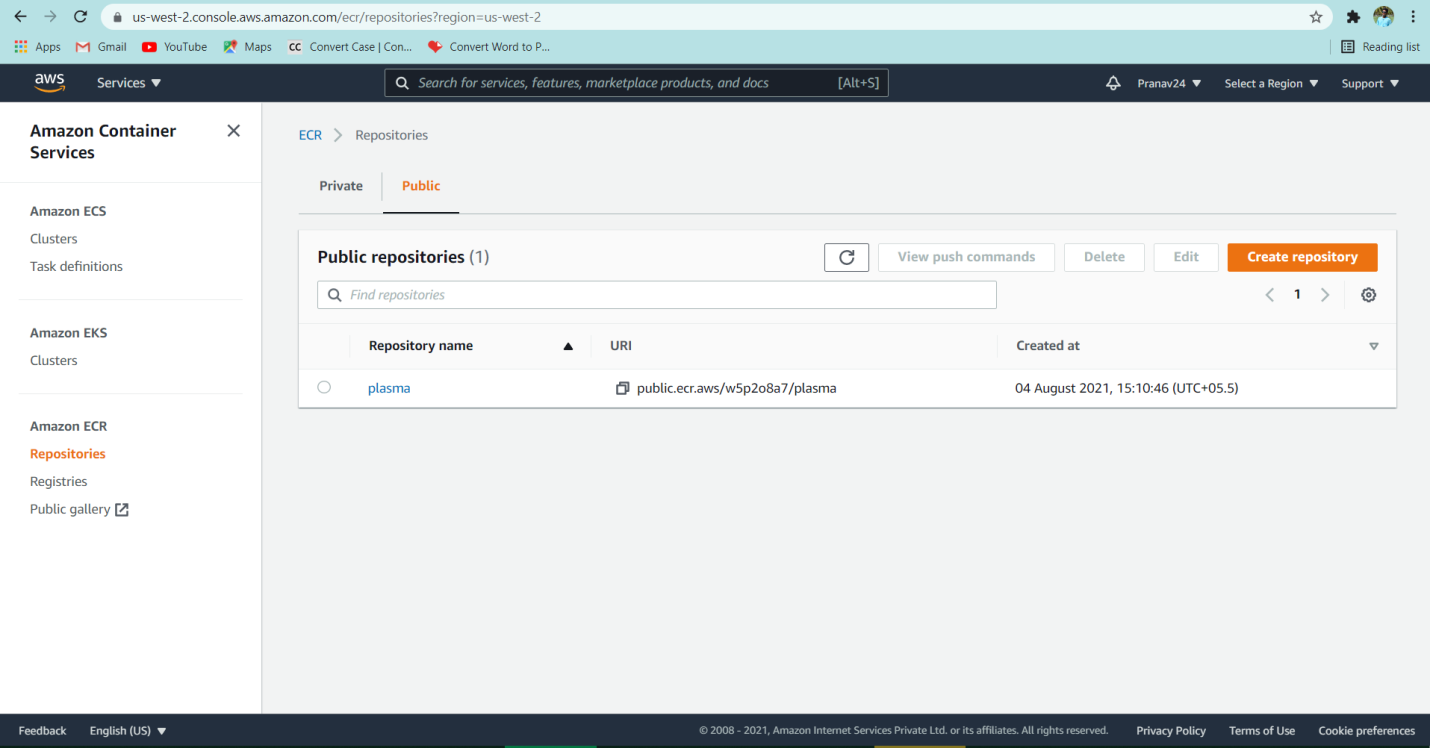


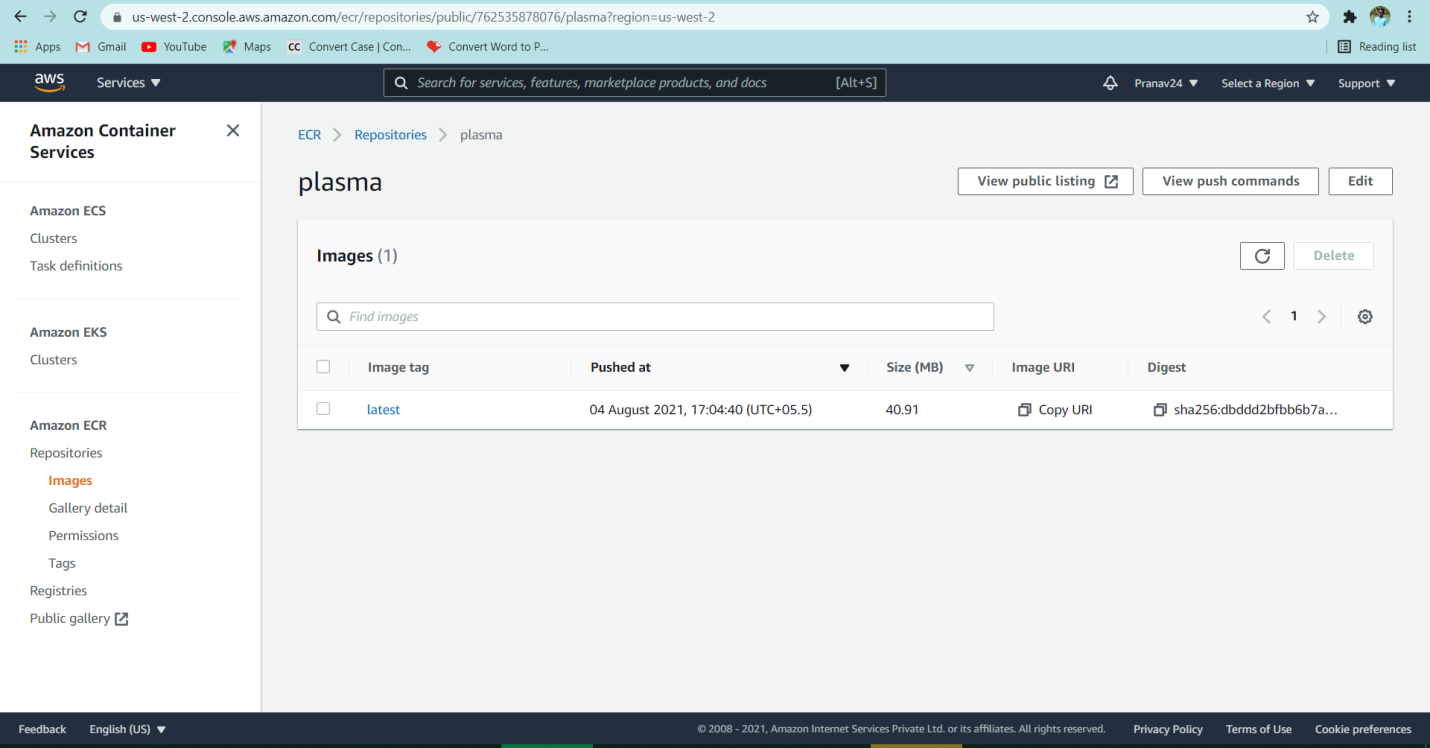


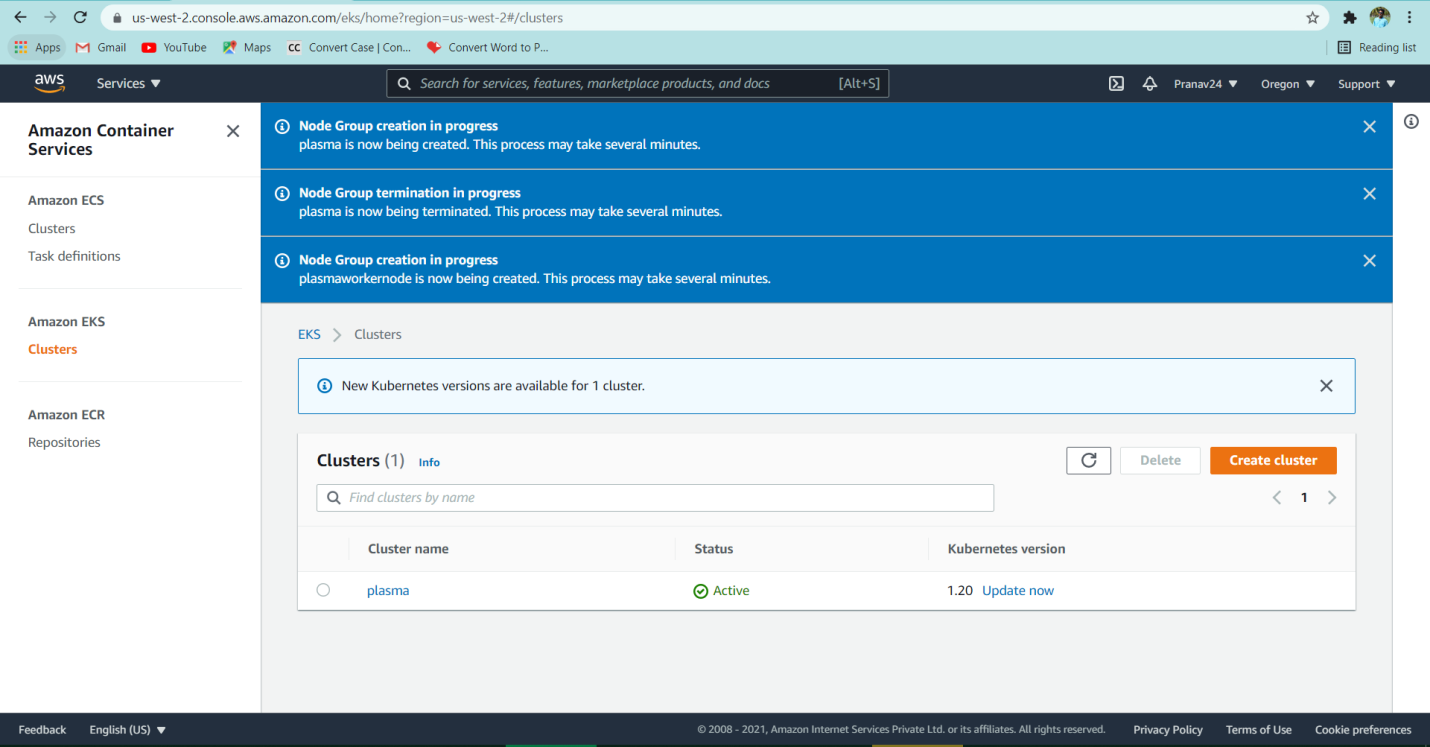


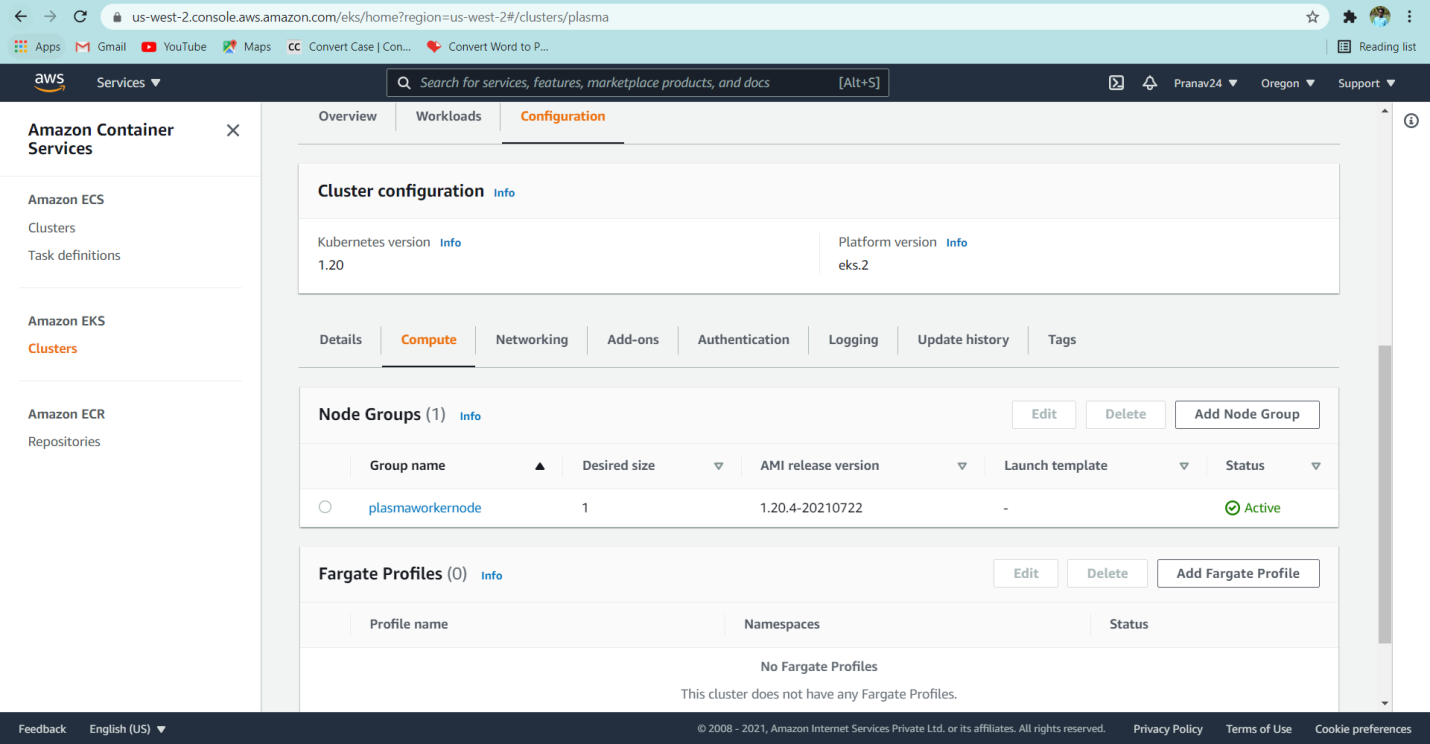


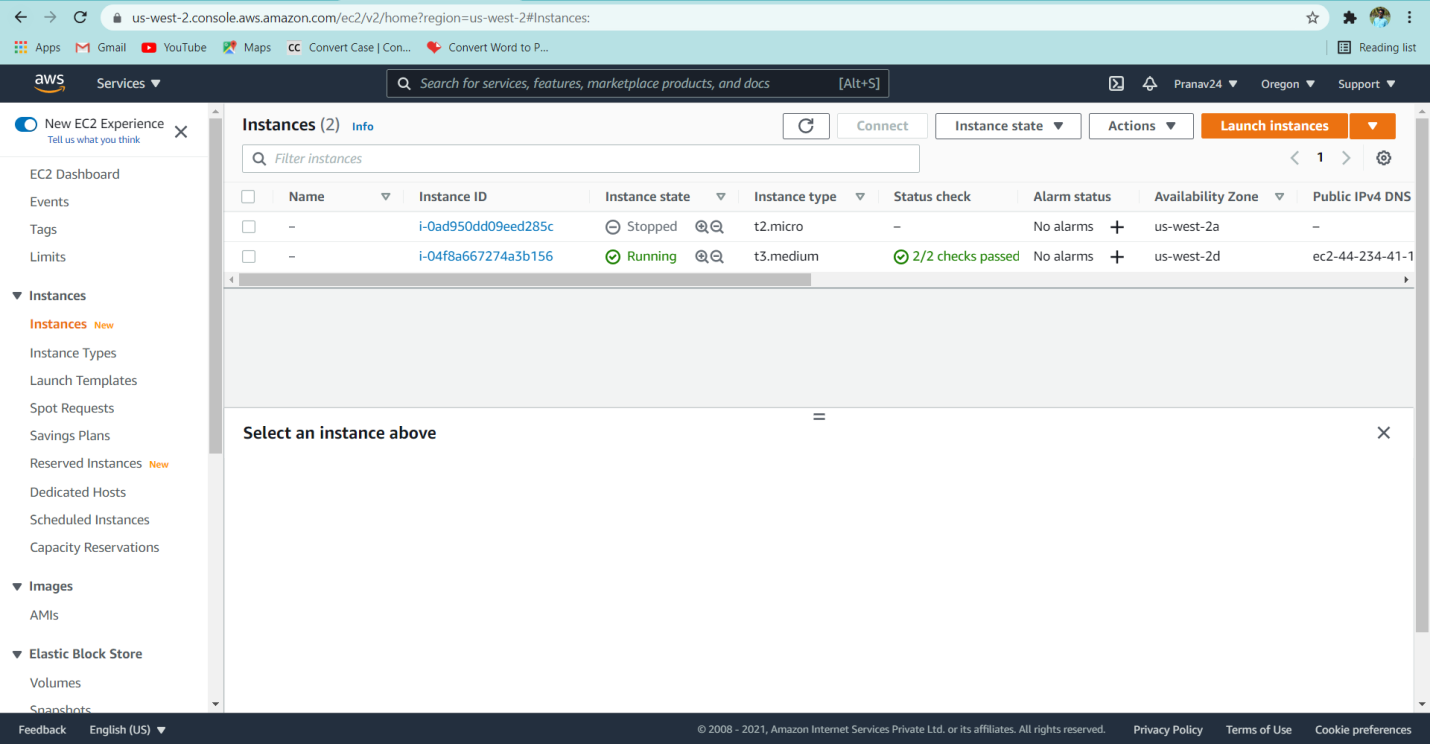


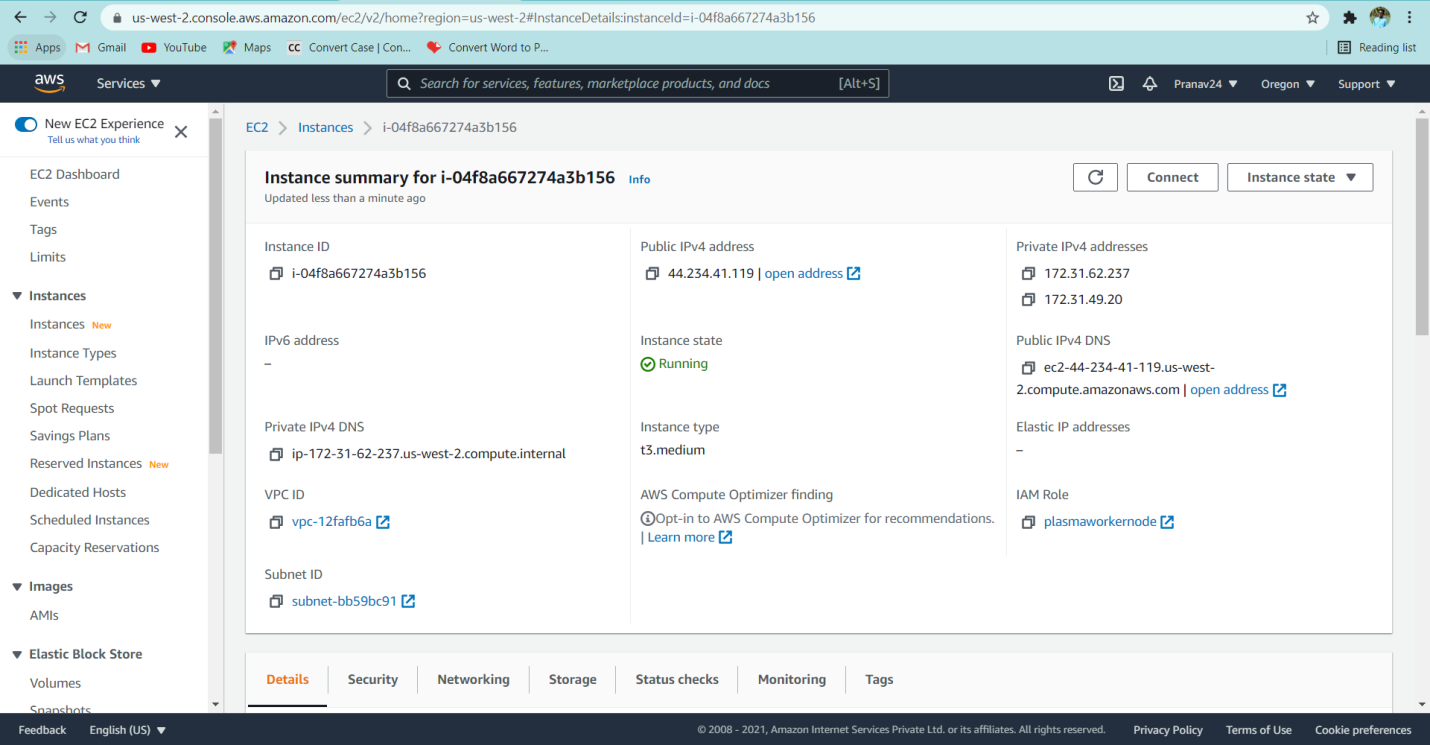












we used DevOps tools which were very useful and simplified in automating the whole deployment process, we are using GitHub and GitBash for version control of the code, for containerizing the application and docker and docker hub to build image, and finally used CI/CD pipeline tool Jenkins to automate the application and deploy the application in EC2 instance, we also used GitHub-webhook to the pipeline build.

**ADVANTAGES & DISADVANTAGES:**

**ADVANTAGES:**

* Easy to find plasma donors
* Easy to tack donors and acceptors number
* Availability of specific blood group donors
* Easy to manage

**DISADVANTAGES:**

* If in the wrong hands, privacy issues arises
* Black marketing of plasma via some organisation can also be done
* Lack of reliablity in informations provided by donors

**APPLICATION:**

This application can be helping hand for Covid patients in need of plasma. Via this app we are storing information of plasma doners i.e their blood group, covid infection status and contact details. While someone in need we help them with these donors contact details.

**Conclusion:**

When someone starts the app we ask them to signup with their name, email and basic information covid infection status, blood group, contact details which will be proved helpful in plasma donation. We maintain these records on our dashboard with blood groups and with the no. of donors available. Where the other users can also login and see the donors availability. Those in need can contact these available donors via their contact details.

**FUTURE SCOPE:**

We can improve the user interface of the application and increase the size of the database. We can also check number of successful plasma donation and patients recovery. We can also increase database to store these world wide information. Using these information we can find out the percentage of population recovered from covid19 and whether they are getting infected again or not and how long their immunity lasts. While the vaccination we can also maintain database and notify people when there is vaccine available in their locality.

**BIBILOGRAPHY:**

1.[Coaching on DevOps and Cloud Computing: Create Jenkins Pipeline for automating Docker image creation and push docker image into Docker Hub | Dockerize Python App (coachdevops.com)](https://www.coachdevops.com/2020/05/automate-docker-builds-using-jenkins.html)

2.[(7) Create an AWS Account(Free) - YouTube](https://www.youtube.com/watch?v=oUoJBayrJT4)

3.**Amazon rekognition Service**

Link : <https://www.youtube.com/watch?v=SZa2HfR-9Xc&t=4s>

**Amazon S3 Bucket :**

Link : <https://www.youtube.com/watch?v=_I14_sXHO8U>

**API gateway :**

Link : <https://www.youtube.com/watch?v=vHQqQBYJtLI>

**Lambda :**

Link : <https://www.youtube.com/watch?v=eOBq__h4OJ4>

4.https://github.com/Guided-Projects/AI-based-attendance-presence-calculation-using-AWS

**APPENDIX**:

GitHub repository for source code: <https://github.com/Pranav2477/plasmadonorapp>

**DEMONSTRATION VIDEO LINK:**

<https://drive.google.com/file/d/1-p_kZ3oLCaT5jUzxiyvUE0mnKHvA_kJ2/view?usp=sharing>