**DevOps Test Requirements**

This README.md provides an explanation of the requirements for a simple DevOps test, including Dockerizing the application, deploying with Kubernetes, setting up Prometheus monitoring, and integrating with the ELK stack.

**1. Dockerize the Application**

**Dockerfile**

Q) Create a Dockerfile for the Python application. This file should include instructions to install all necessary dependencies, expose the correct port, and incorporate configurations managed by an appropriate .env file.

A) I have created a Dockerfile for the python application including necessary configurations, dependencies and correct port exposed.

Note: Dockerfile and dependencies are copied in GitHub Repository

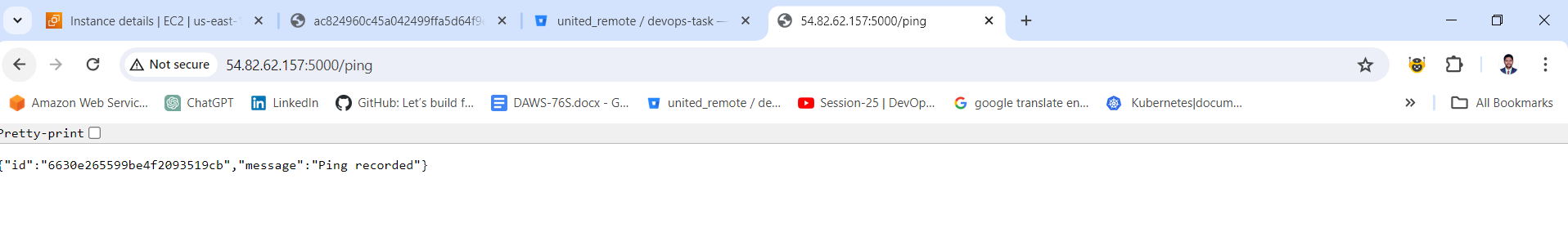
**Build and Test Locally**

Q) Use Docker commands to build an image from the Dockerfile and run a container locally. Ensure the application functions correctly within the container.

A) I have created the image using Dockerfile and run the container locally and over internet and it works fine. Please find below the screenshot of application output .

[centos@ip-172-31-30-37 dockercode]$ curl 54.82.62.157:5000/ping

{"id":"6630e216599be4f2093519ca","message":"Ping recorded"}



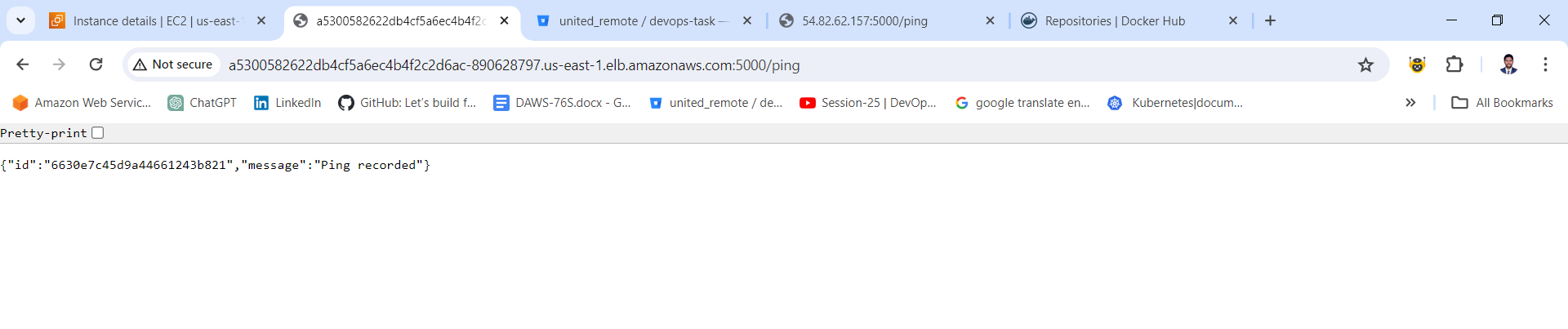
**2. Deploy with Kubernetes**

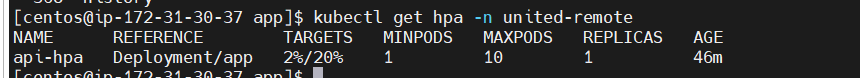
**Kubernetes Deployment YAML File**

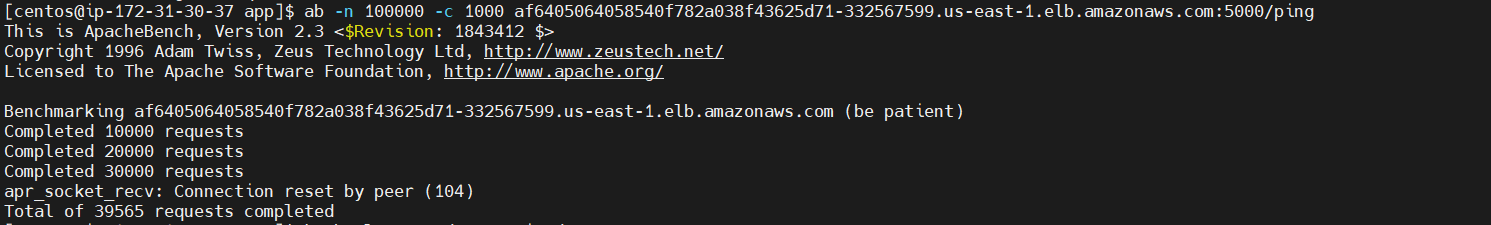
Q)Write a YAML file to describe how Kubernetes should deploy the Docker container. Include specifications such as the container image, resource requirements, number of replicas, and other configurations.

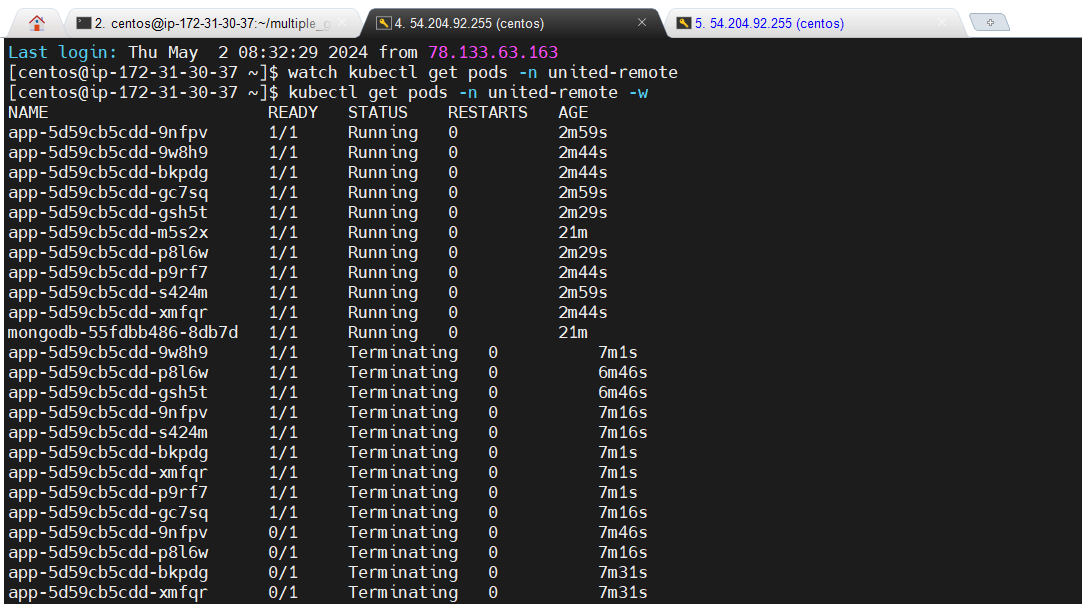
A) I have written a YAML manifest files for Deployment ,Service,ConfigMap and HPA specifications and configured LoadBalancer and Horizontal Pod AutoScaler for our application and the application works fine over internet with LoadBalancer and HPA. The HPA has been tested based on CPU utilization using Apache benchmark tool.

Please find below the s of application output .









Note: Kubernetes specifications are copied in GitHub Repository.

**LoadBalancer Service**

Configure a LoadBalancer service in Kubernetes to enable external access to the application.

**Horizontal Pod Autoscaler (HPA)**

Include a configuration for Horizontal Pod Autoscaler (HPA) to automatically scale the application based on CPU usage.

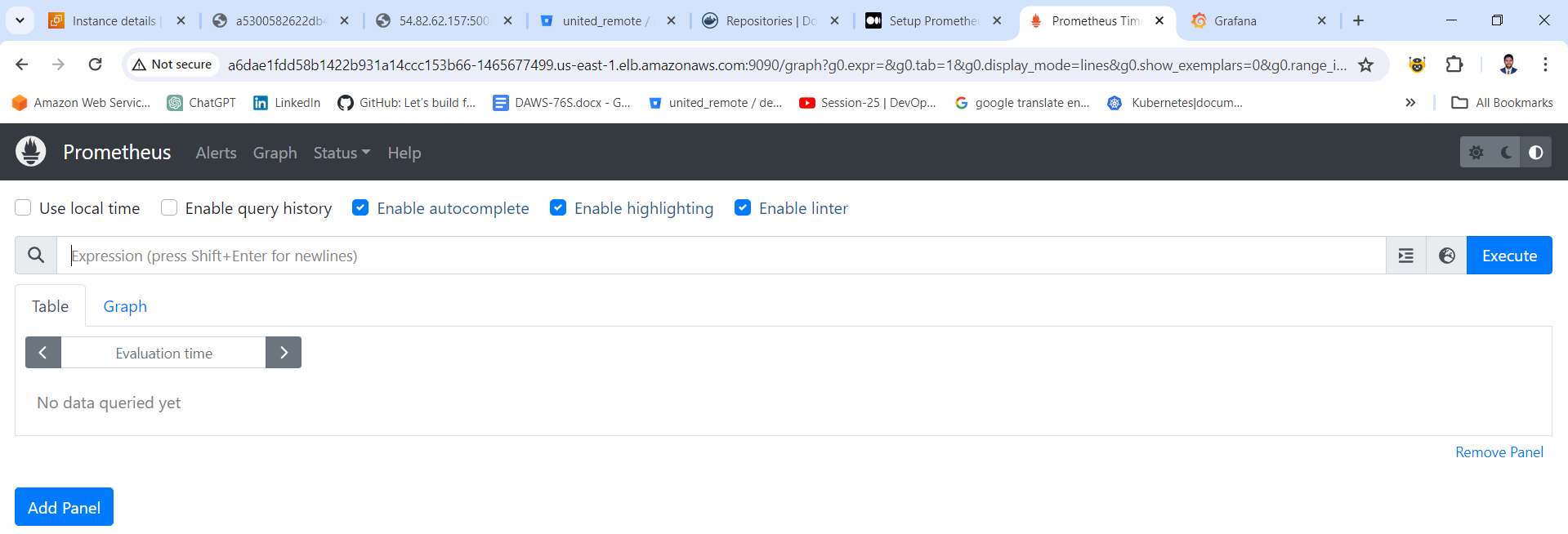
**3. Setup Prometheus Monitoring**

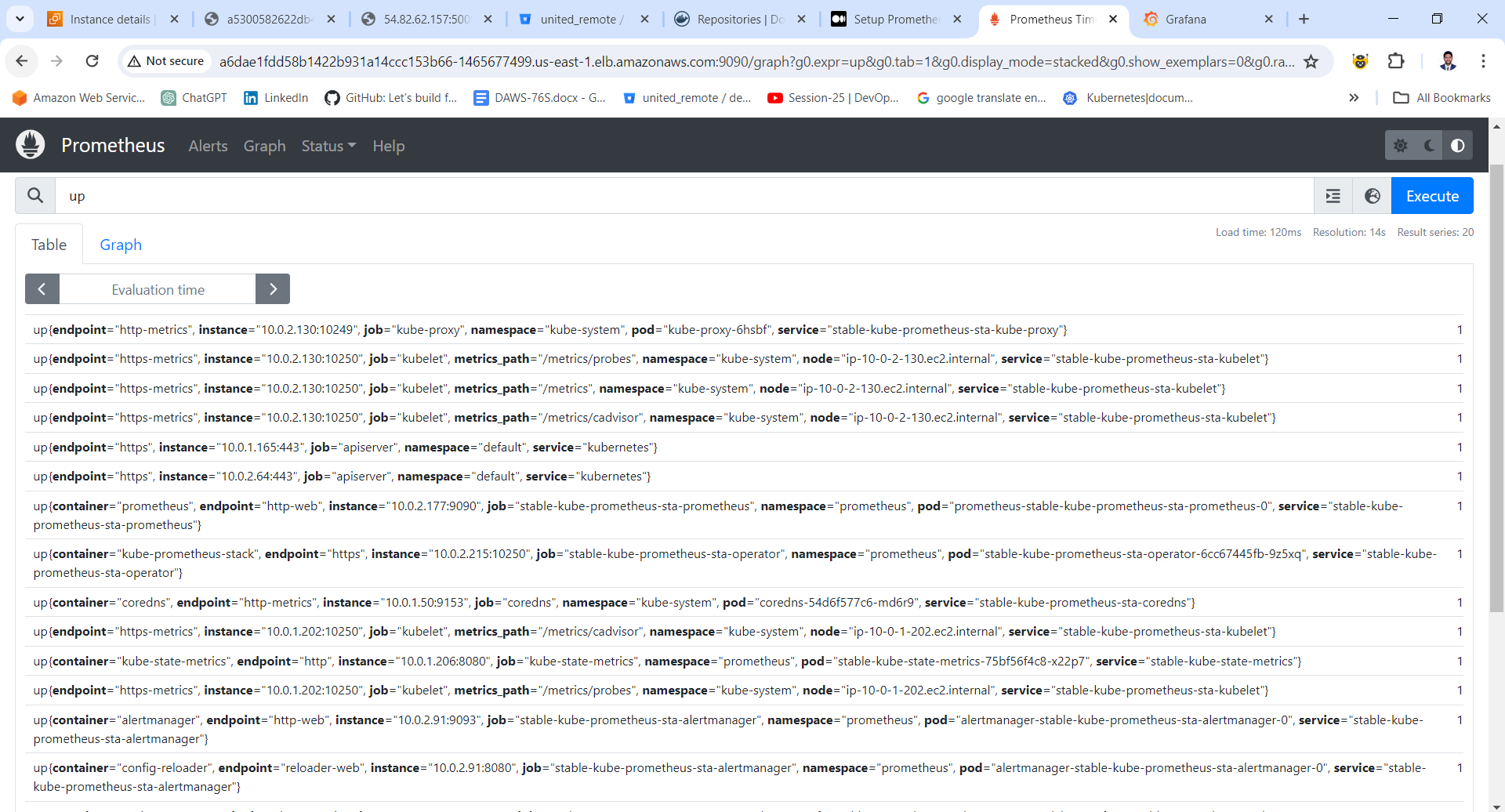
**Prometheus Configuration**

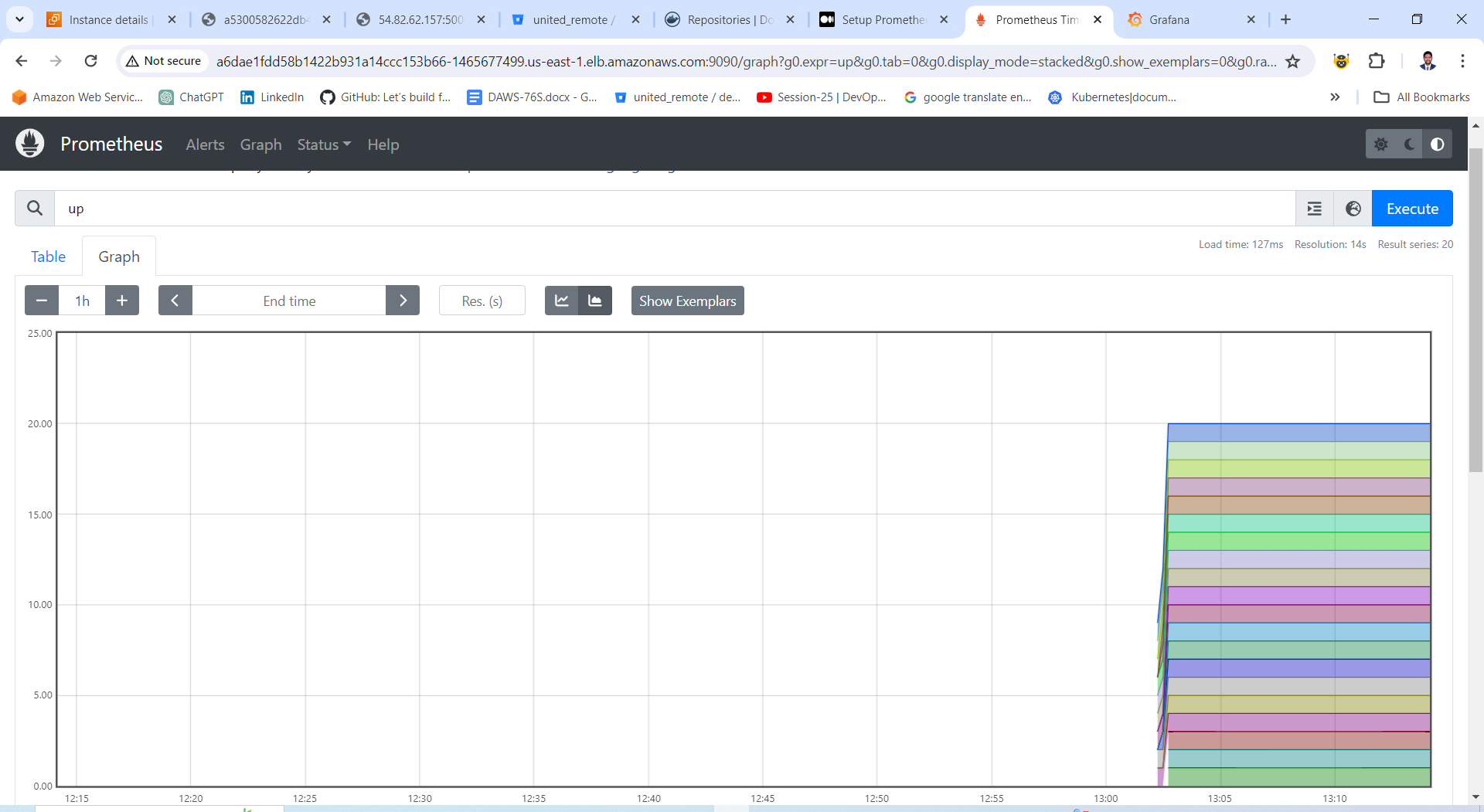
Q) Configure Prometheus to monitor the application by specifying which metrics to scrape. This may include system metrics or custom application metrics.

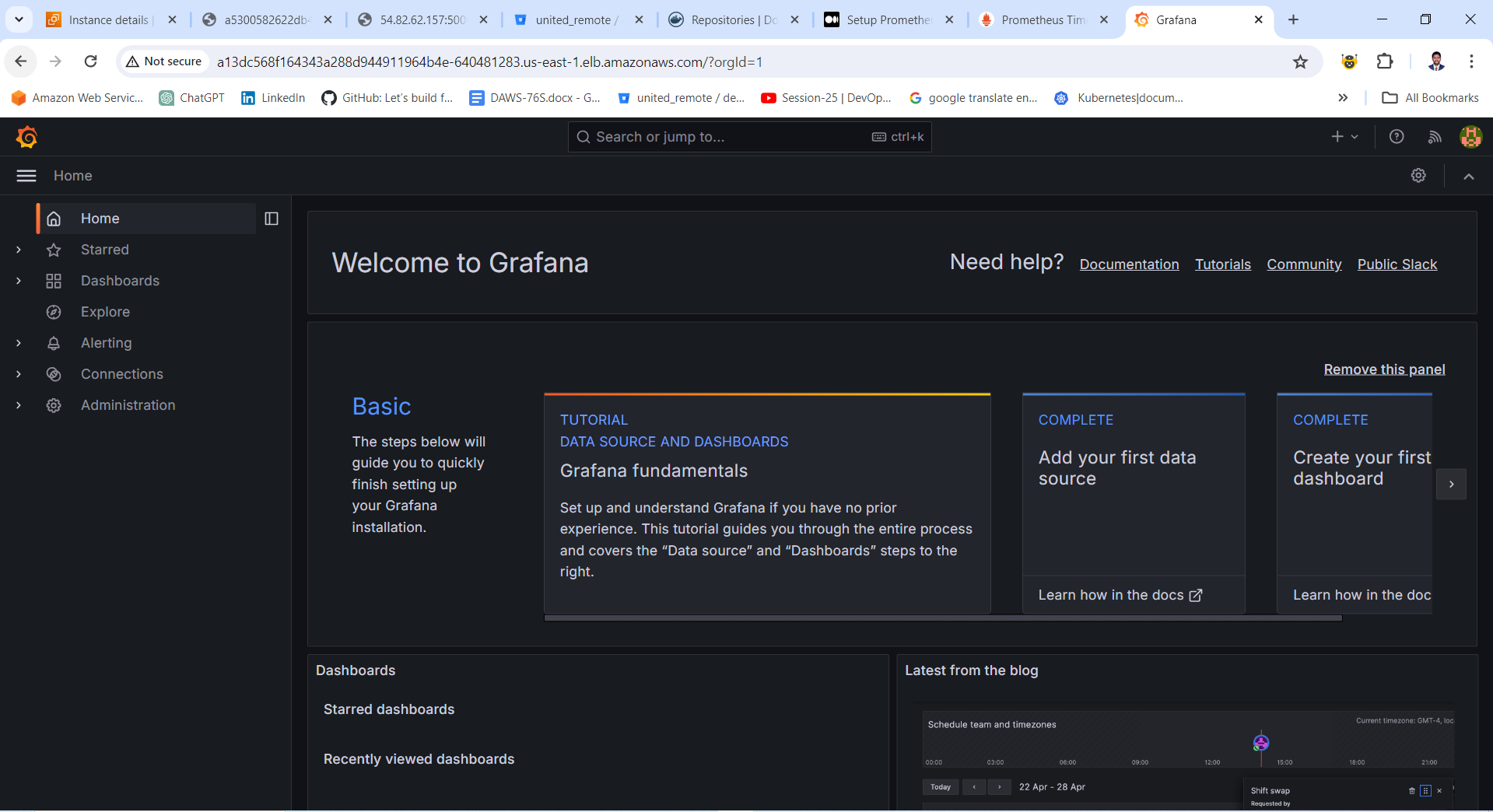
A) I have configured Prometheus to monitor the system metrics of Kubernetes Cluster instance and also deploy the Prometheus as a container with in the Kubernetes Cluster for monitoring of application and configured Grafana for visualize the Prometheus metrics with simple dashboard.

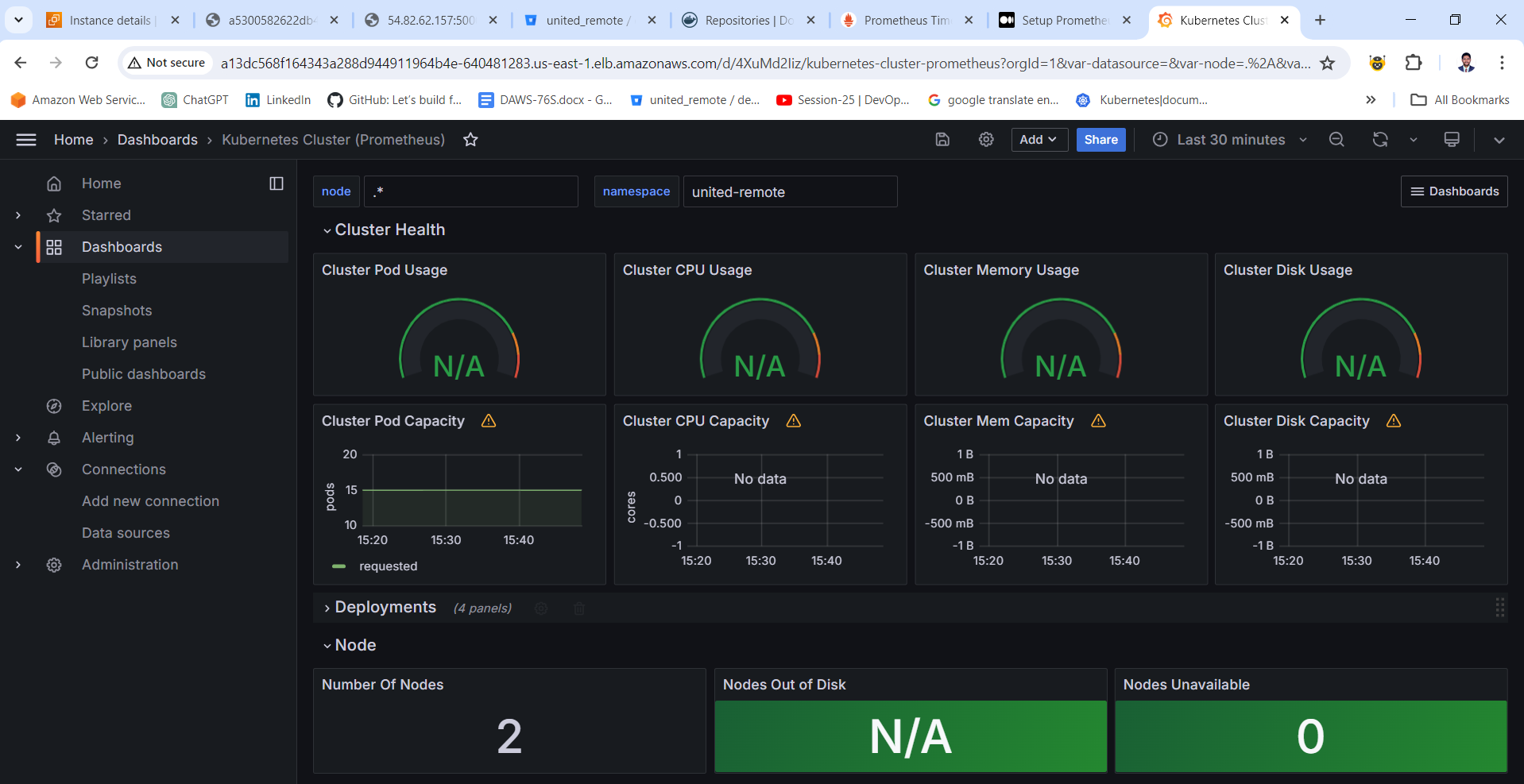
Please find below the screenshots of server and application logs output from Prometheus and Grafana.

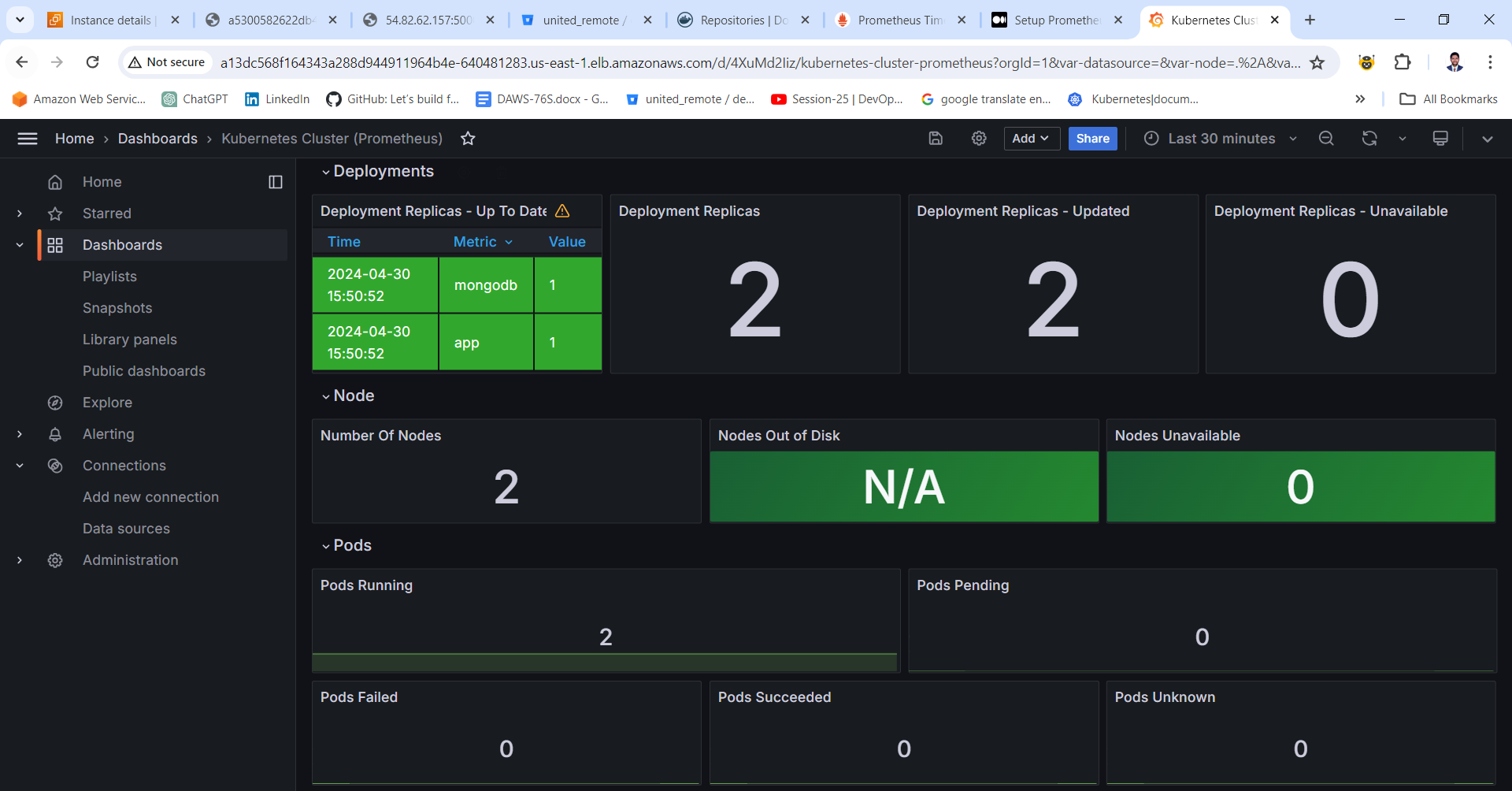


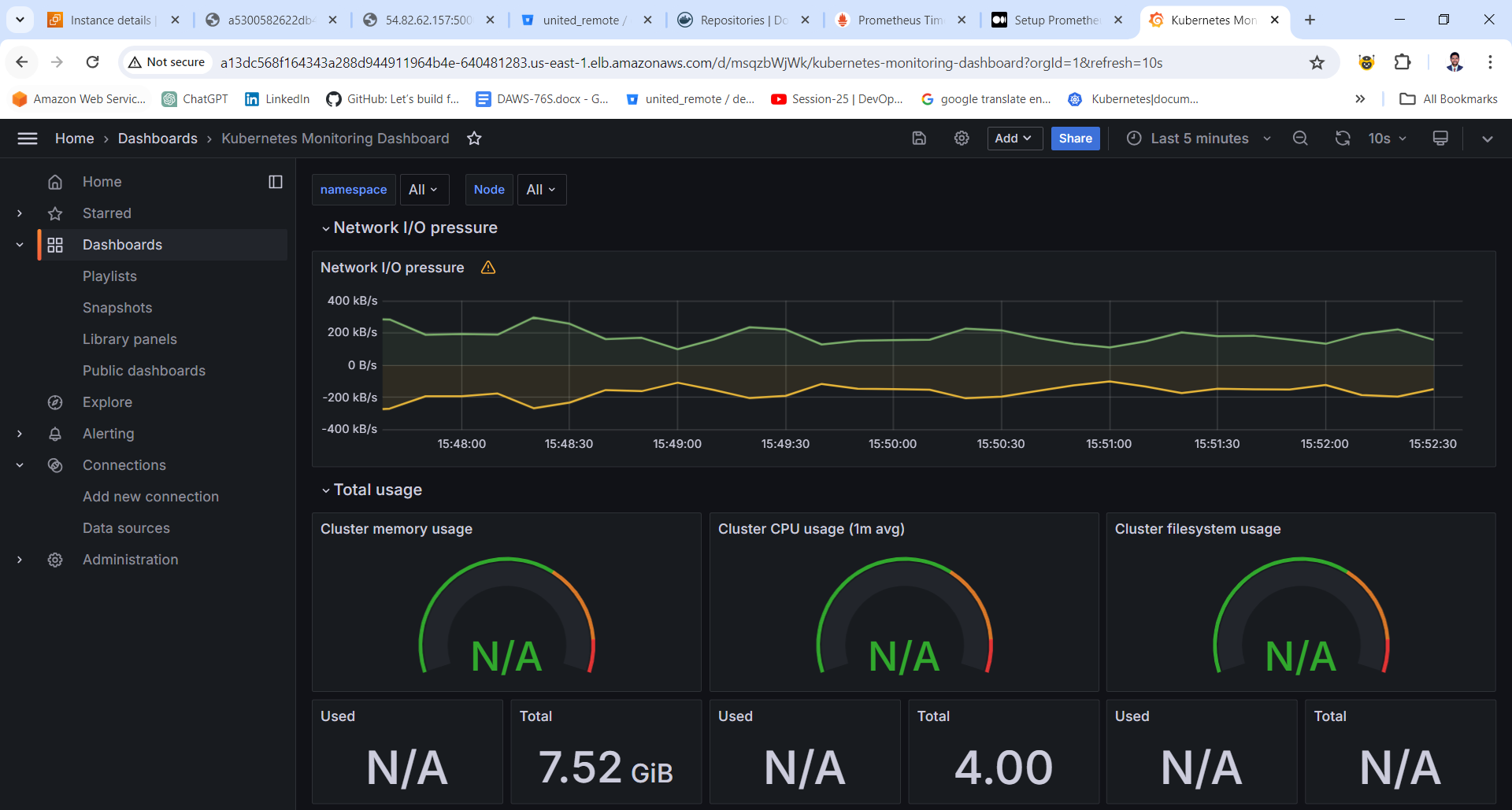












**Deploy Prometheus within Kubernetes Cluster**

Deploy Prometheus as a container within the Kubernetes cluster to enable scraping metrics from the application.

**Visualize Metrics**

Visualize the Prometheus metrics through a simple dashboard

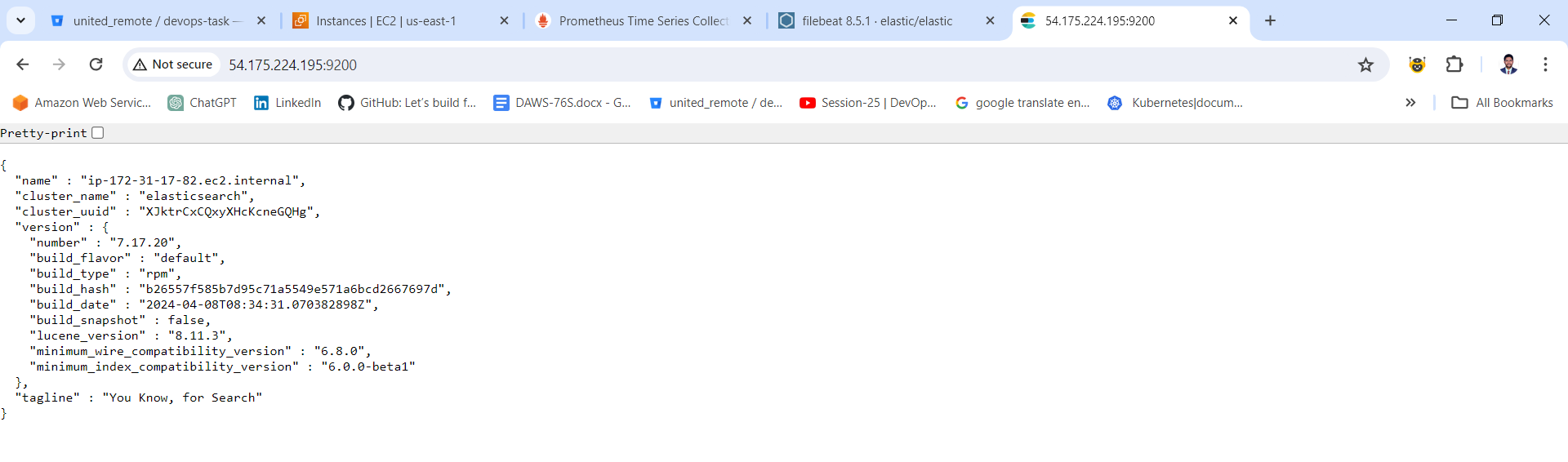
**4. Integrate with ELK Stack**

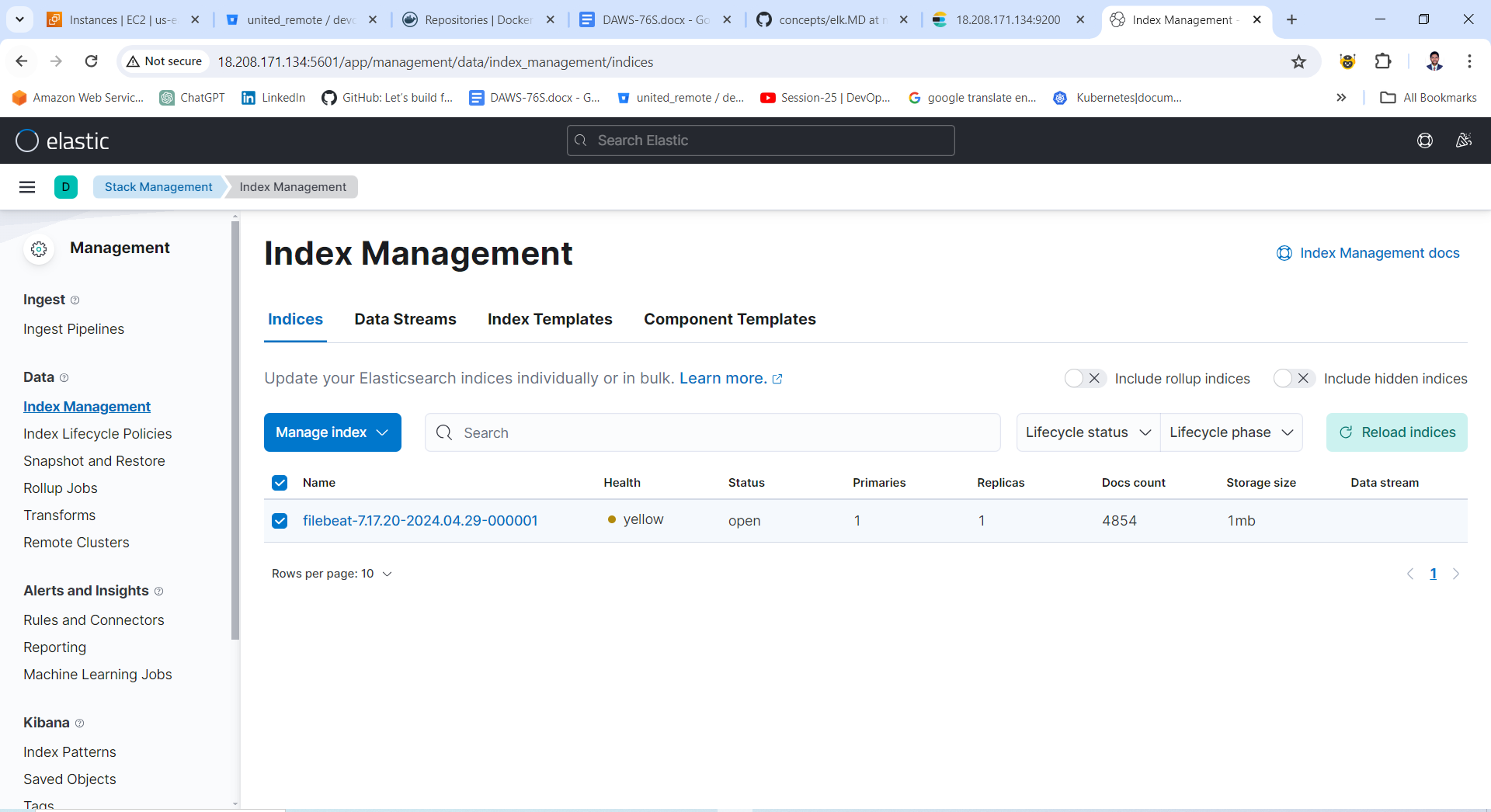
**Filebeat Configuration**

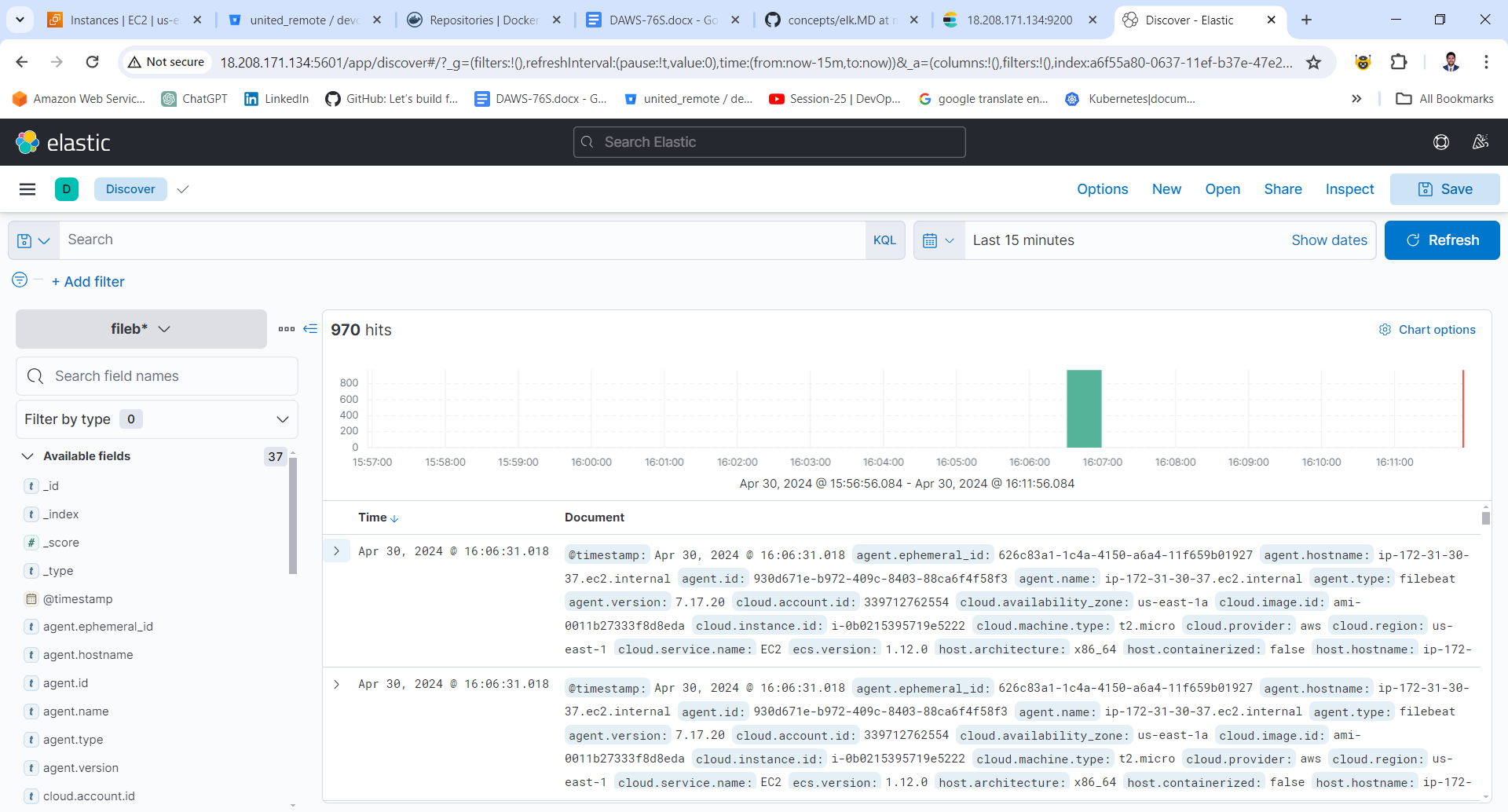
Q) Configure Filebeat to watch the application's log files and ship them to Elasticsearch.

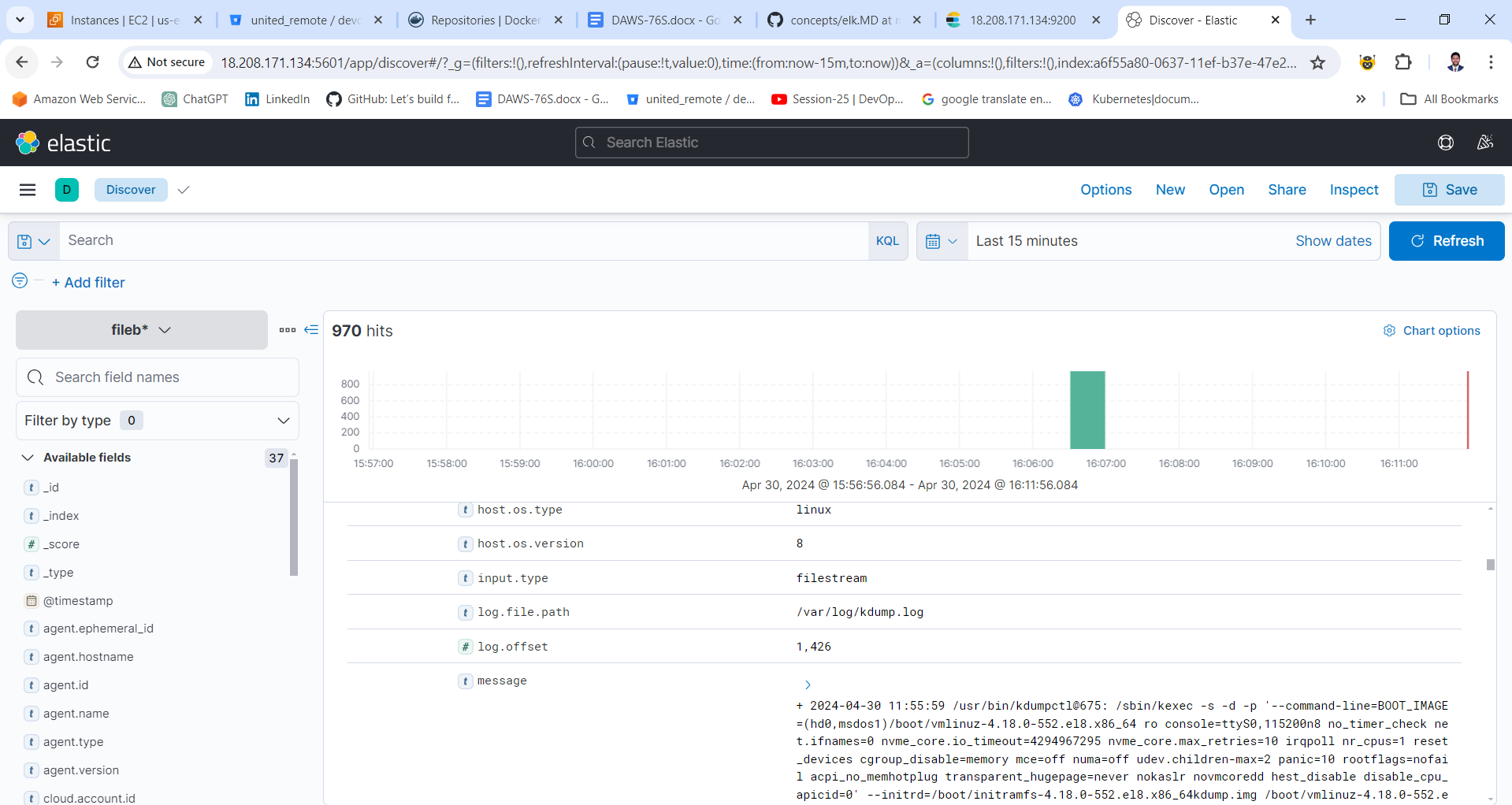
A) I have installed the Filebeat in Kubernetes Cluster and ElasticSearch IP address is configured on Filebeat configuration file to ship the application logs into ElasticSearch tool and ElasticSearch and Kibana are installed in separate instance and the ElasticSearch IP is configured on Kibana configuration file to visualize the application logs on Kibana dashboard.

Please find below the screenshots of server logs output from Kubernetes Cluster on Kibana Dashboard.









**Elasticsearch**

Ensure Elasticsearch is running within the environment to store and index the logs sent from Filebeat.

**Kibana**

Set up Kibana to visualize the logs stored in Elasticsearch. Create a dashboard in Kibana to display key information from the application's logs.

By following these steps, you'll effectively Dockerize your application, deploy it using Kubernetes, set up monitoring with Prometheus, and integrate logging with the ELK stack, enabling efficient management, monitoring, and troubleshooting of your application in a DevOps environment.

**Evaluation Criteria**

* Code quality, structure and organization
* Docker & Kubernetes best practices
* Application metrics, collect and visualize
* Logs Managment, collect and visualize logs

**Submission**

Please create a public GitHub repository and send the link to your repository once you have completed the project to [development@united-remote.com](https://bitbucket.org/united_remote/devops-task/src/ff9596189d47f5031ccd13a84f956d11da9a6473/mailto:development@united-remote.com).