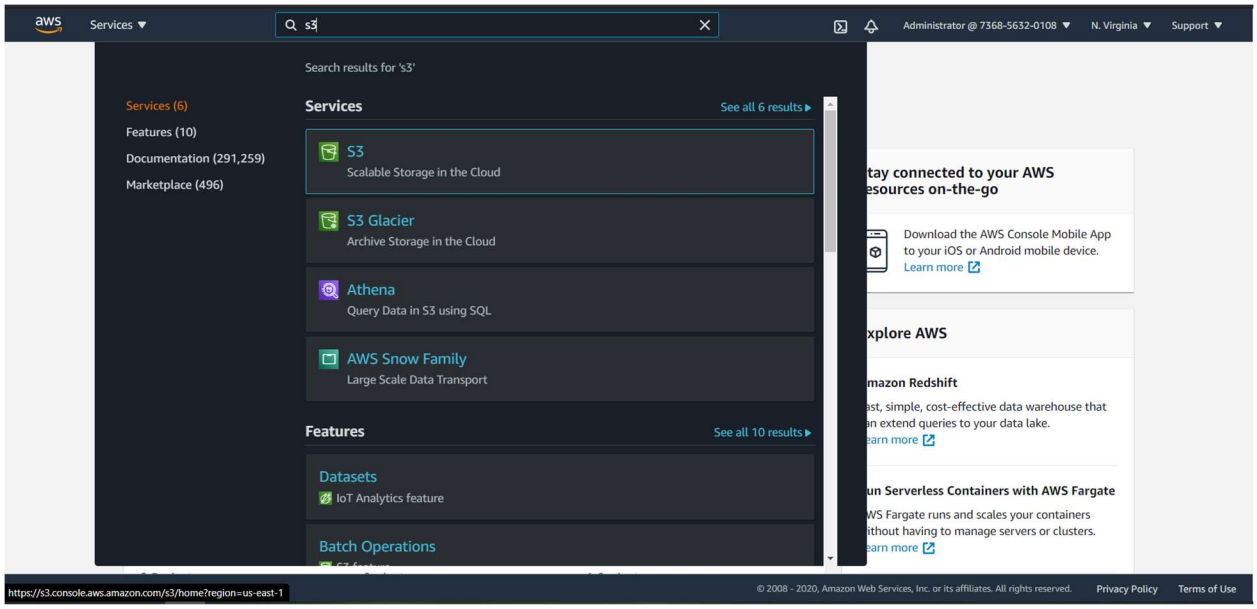
Team 1: Steps to Run Project

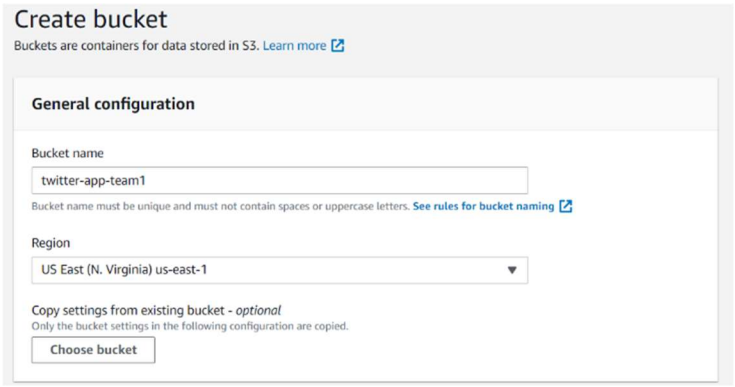
Set Up S3 Bucket

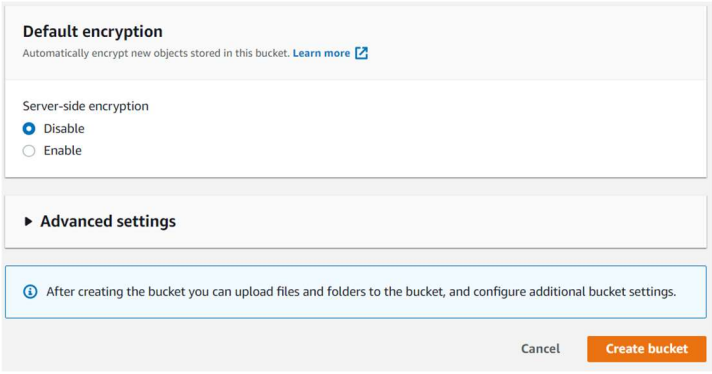
1. These steps can be skipped if you already have a s3 bucket in your account that can be used for testing our project.
2. Login to AWS console and search for S3 in the top search bar.



3. Click on S3 to view the S3 Dashboard. Click on the **Create bucket** button.

4. Enter any valid bucket name and click on **Create bucket** at the bottom of the page.

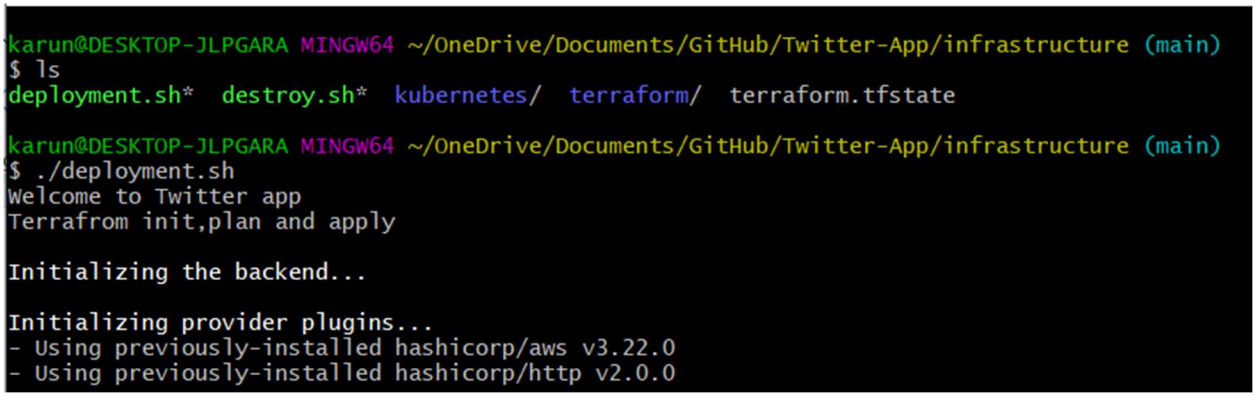




5. Use this s3 bucket name in the creation script.

Deploying the Application

1. Clone the project.
2. Open the terminal (Linux/Mac) or Git bash (Windows) in the infrastructure directory of the project.
3. Run the deployment script using the command "`./deployment.sh`"



4. The script first brings up the EKS cluster using Terraform on the AWS account that AWS CLI has been configured to use and sets up kubectl.
5. Once the cluster is up it will ask for the following details to populate the k8s secret files in the given order:
    - MongoDB Cluster URI: (Enter MongoDB Atlas Cluster URI)
    - MongoDB Database Name: (Enter MongoDB Database Name)
    - Slack Webhook URI: (Slack Webhook URI)
    - JWT Secret: (Enter JWT Secret)
    - RapiAPI Key: (Enter RapidAPI Key)
    - AWS Access Key: (AWS Account Access Key ID for S3 Bucket)
    - AWS Secret Access Key: (AWS Account Access Key ID for S3 Bucket)
    - S3 Bucket Name: (S3 Bucket Name)

6. Once the secrets are populated all the services get deployed to the EKS cluster.

```
clusterrolebinding.rbac.authorization.k8s.io/kube-state-metrics created
clusterrole.rbac.authorization.k8s.io/kube-state-metrics created
deployment.apps/kube-state-metrics created
serviceaccount/kube-state-metrics created
service/kube-state-metrics created
NAME                                  READY   STATUS              RESTARTS   AGE
grafana-86b84774bb-nq9sz              0/1     ContainerCreating   0          6s
prometheus-server-778bd7fb69-rmf2l    0/1     ContainerCreating   0          9s
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP
                           PORT(S)          AGE
grafana              LoadBalancer   172.20.151.11    ab40f5b0f37e546d587b36bd64e6e9d2-399495971.us-e
ast-1.elb.amazonaws.com    3000:31677/TCP   6s
prometheus-service   LoadBalancer   172.20.155.187   a44a12b8144694885881354f8d689b11-1233806019.us-
east-1.elb.amazonaws.com   80:30068/TCP     8s

karun@DESKTOP-JLPGARA MINGW64 ~/OneDrive/Documents/GitHub/Twitter-App/infrastructure (main)
$
```

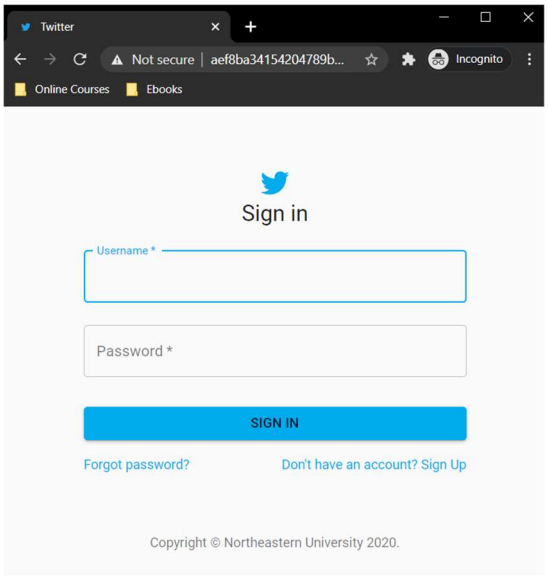
7. Wait for all the pods to be ready. Run the command "**kubectl get pods"** to check the status of the pods.

```
karun@DESKTOP-JLPGARA MINGW64 ~/OneDrive/Documents/GitHub/Twitter-App/infrastructure (main)
$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
twitter-backend-7cfdc86d4c-8zzlp     1/1     Running   0          6m34s
twitter-backend-7cfdc86d4c-z4vz4     1/1     Running   0          6m34s
twitter-frontend-789798c686-8d8bj    1/1     Running   0          6m26s
twitter-frontend-789798c686-fgqxr    1/1     Running   0          6m26s
twitter-middleware-6bc9cf4f5b-jcrbz  1/1     Running   0          6m27s
twitter-queue-68bf798fcf-lhx2t       1/1     Running   0          6m30s
twitter-tweet-864cdb79b7-sllhh       1/1     Running   0          6m32s
twitter-tweet-864cdb79b7-t4bj8       1/1     Running   0          6m32s
```

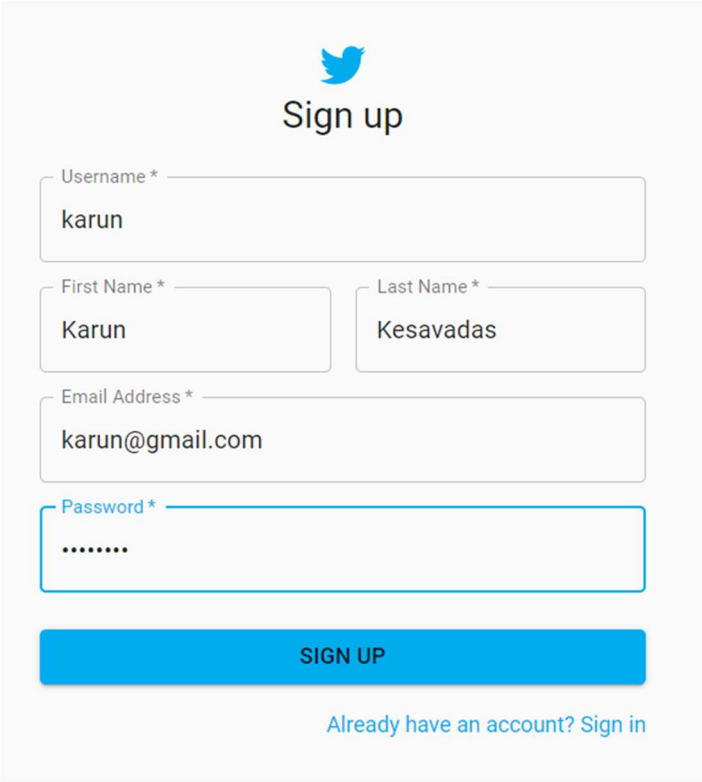
8. Once the pods are ready run the command "**`kubectl get svc twitter-frontend`**" to get the `Front-end loadbalancer IP.`

```
karun@DESKTOP-JLPGARA MINGW64 ~/OneDrive/Documents/GitHub/Twitter-App/infrastructure (main)
$ kubectl get svc twitter-frontend
NAME               TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)        AGE
twitter-frontend   LoadBalancer   172.20.14.215   aef8ba34154204789bd92dbf529c5b07-1819330803.us-east-1.elb.amazonaws.com   80:30123/TCP   8m6s
```

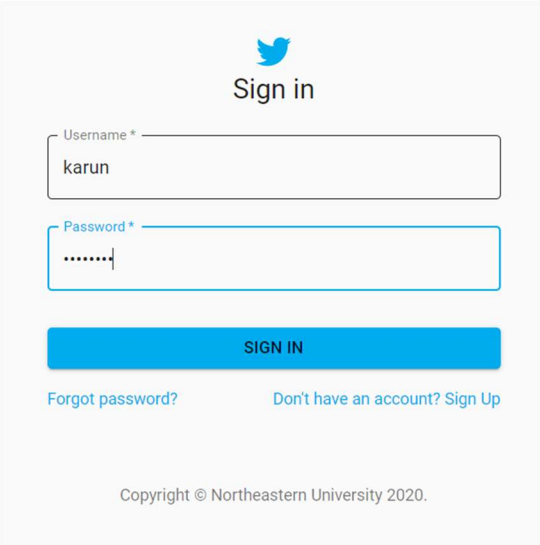
9. Access the front-end in a web browser using the EXTERNAL-IP from the previous command.



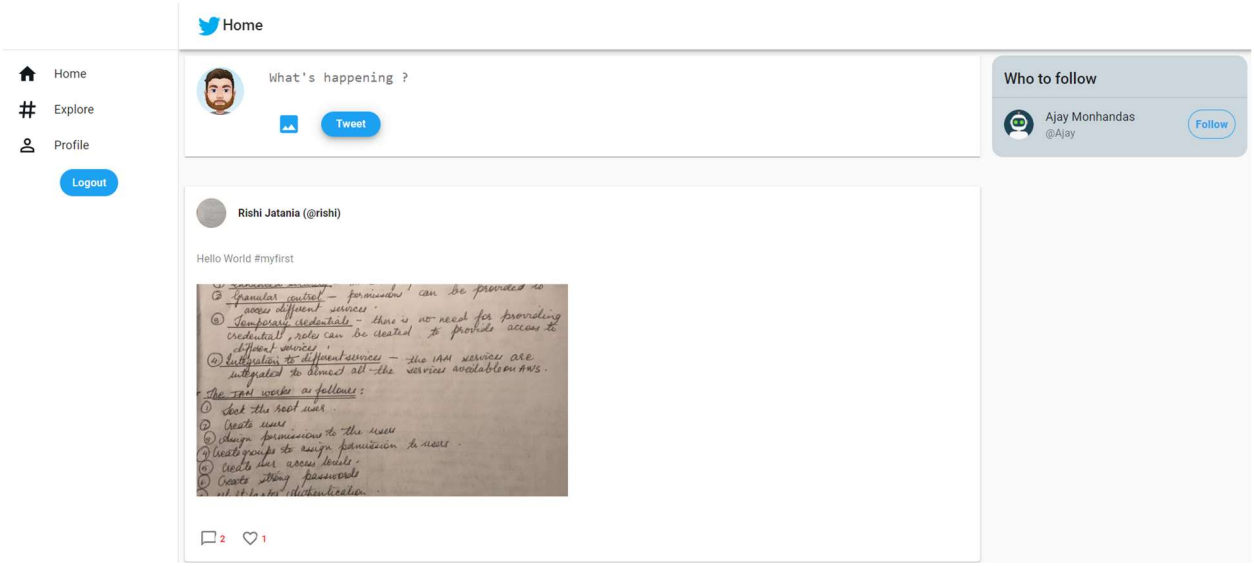
10. Click on “Don’t have account? Sign Up” to go the sign-up page and sign up for an account.



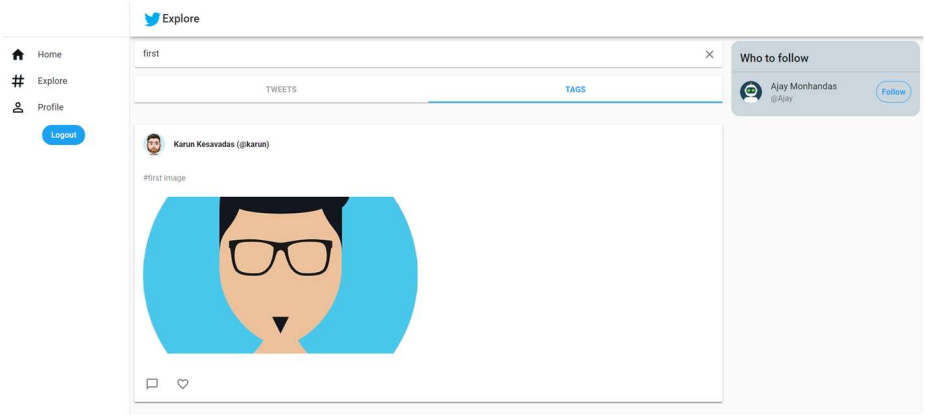
11. Login using the credentials



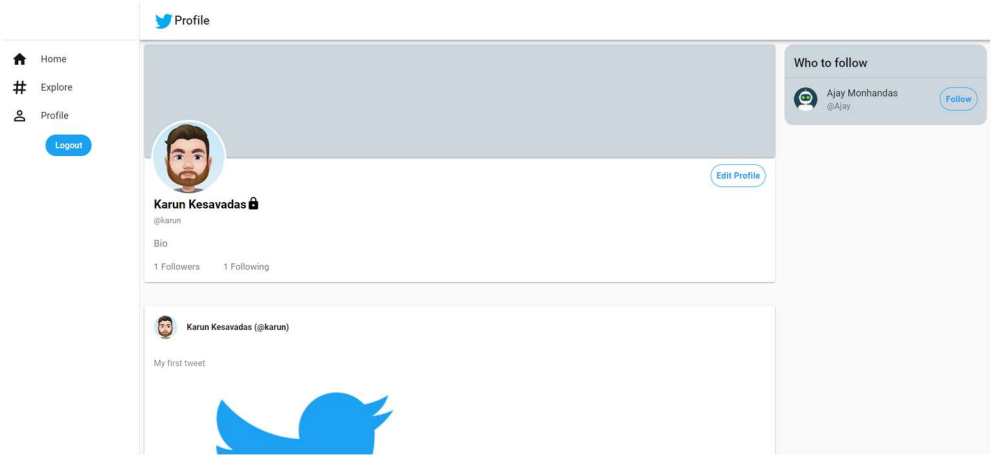
12. On logging in you will see the home page where you can post tweets, follow other users, Comment and like tweets in your user feed.



13. On the explore tab you can search for tweets by text or tags.



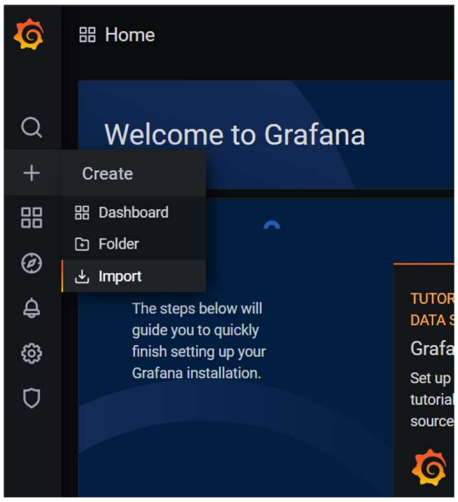
14. On the profile page you can see the tweets posted by you and edit your profile.



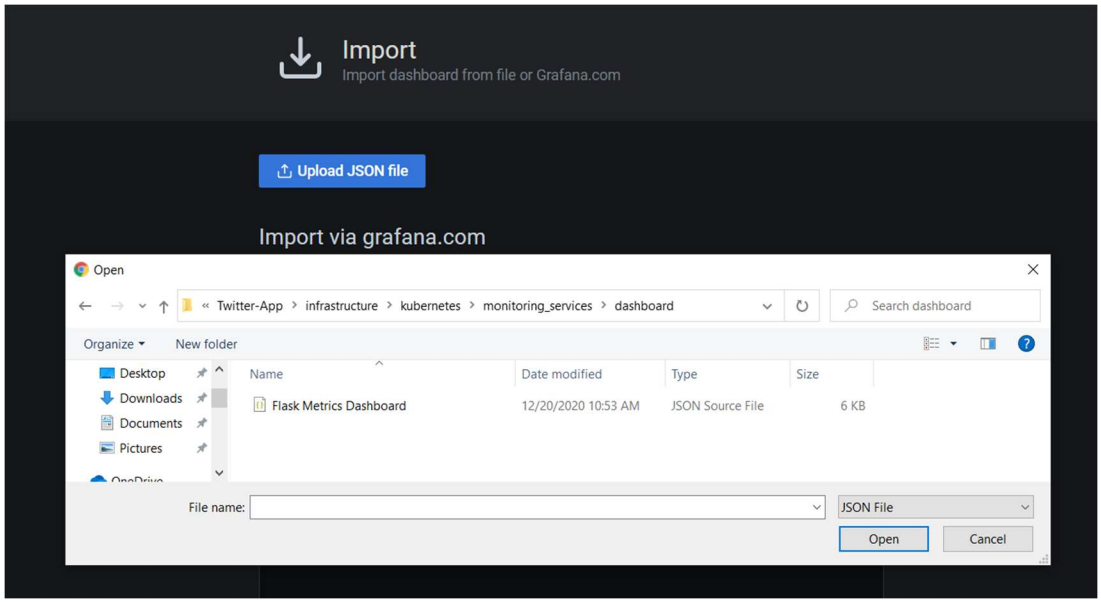
15. Logout of the application by clicking the Logout button on the sidebar.
16. To access the Grafana dashboard get the EXTERNAL IP and PORT of the Grafana load balancer using the command "**`kubectl get svc -n monitoring`**"

```
karun@DESKTOP-JLPGARA MINGW64 ~/OneDrive/Documents/GitHub/Twitter-App/infrastructure/kubernetes/deployment_services (main)
$ kubectl get svc -n monitoring
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)          AGE
grafana              LoadBalancer   172.20.151.11    ab40f5b0f37e546d587b36bd64e6e9d2-399495971.us-east-1.elb.amazonaws.com     3000:31677/TCP   69m
prometheus-service   LoadBalancer   172.20.155.187   a44a12b81446948858813354f8d689b11-1233806019.us-east-1.elb.amazonaws.com   80:30068/TCP     69m
```

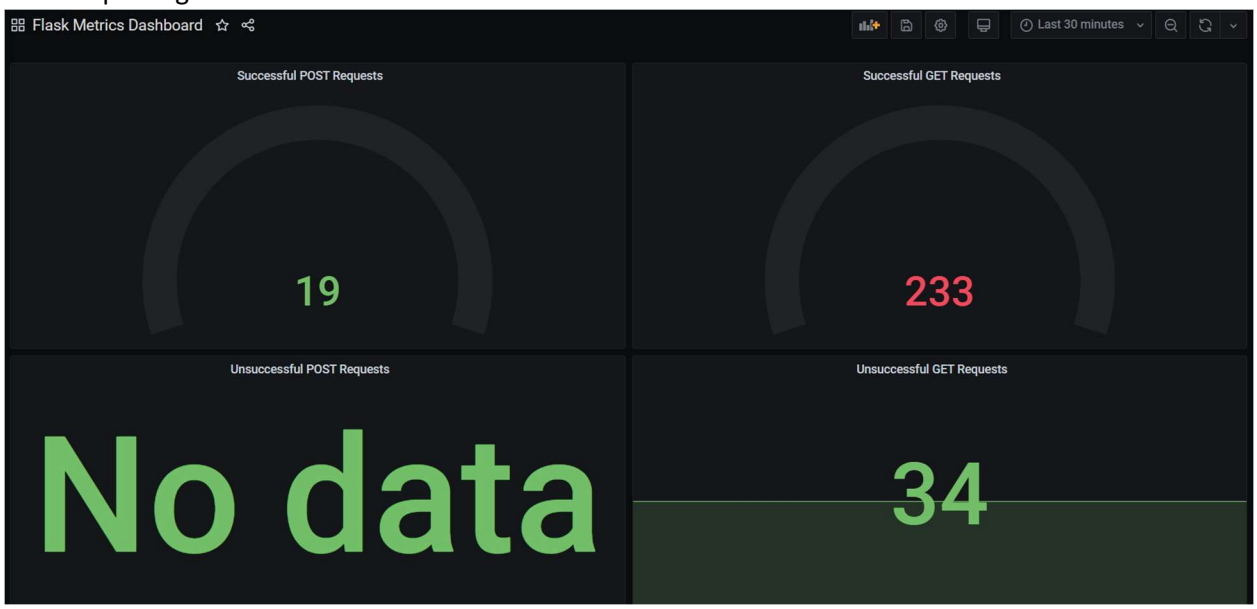
17. Open the Grafana dashboard in a browser using the details obtained in the previous step.
18. Login to the dashboard using credentials "admin" and "admin".
19. Set new password for the Grafana dashboard.
20. Hover over the plus icon on the side bar and click on import.



21. Click on Upload JSON file and browse into the Project_Folder/infrastructure/Kubernetes/monitoring_services/dashboard directory and select the Flask Metrics Dashboard file. After selecting the file, click on Import.



22. After importing the Flask Metrics Dashboard will be visible.



PWA Steps

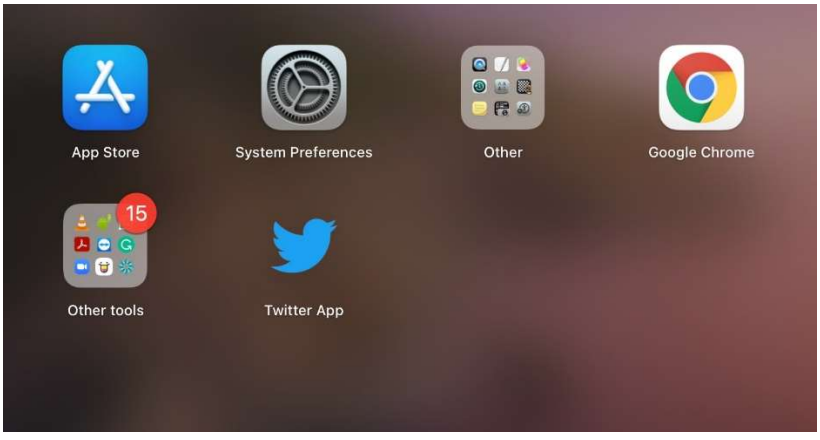
You will find a '+' symbol to right of the URL. Click on it and install the app on to your device.

If you don't find a '+' to the right of the URL bar after going into the deployed website, then follow these steps.

1. Open Chrome browser and type the following in the URL (chrome://flags/#unsafely-treat-insecure-origin-as-secure)

2. Enable the "Insecure origins treated as secure" by adding the deployed application's URL in the input field.

3. Now reload browser and you will see '+' symbol to right of the URL. Click on it and install the app on to your device.



Destroying the Resources

To bring down all the resources open Terminal (Linux/MacOS) or Git Bash (Windows) in the infrastructure directory of the project and run the command "**./destroy.sh**"