

Eugenia Mawuenya Akpo

AIOps

Lab 7: From anomaly detection to incidents

Task 1 deliverables

1. Code for your generalized anomaly monitor: monitor/monitor.py
2. YAML deployment code available in frontend_shipping.yaml

```
deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: frontend-shipping-monitor
5    labels:
6      app: monitor-app
7  spec:
8    replicas: 1 # Set the desired number of replicas
9    selector:
10     matchLabels:
11       app: monitor-app
12    template:
13     metadata:
14       labels:
15         app: monitor-app
16     annotations:
17       prometheus.io/scrape: "true"
18       prometheus.io/port: "8080" # Adjust the port to your needs
19       prometheus.io/path: "/metrics"
20     spec:
21       containers:
22         - name: monitor-app
23           image: index.docker.io/eugeniaakpo/monitor:latest
24           imagePullPolicy: Always
25           ports:
26             - containerPort: 8080 # Expose the port where your monitor is running internally
27           resources:
28             requests:
29               cpu: 200m
30           command:
31             - "python"
32           args:
33             - "monitor.py"
34             - "--service1=frontend"
35             - "--service2=shippingservice"
36             - "--training_data_file=boutique_training.json"
37             - "--prometheus_port=8080"
```

3. Console log of running the test outlined in Task 1

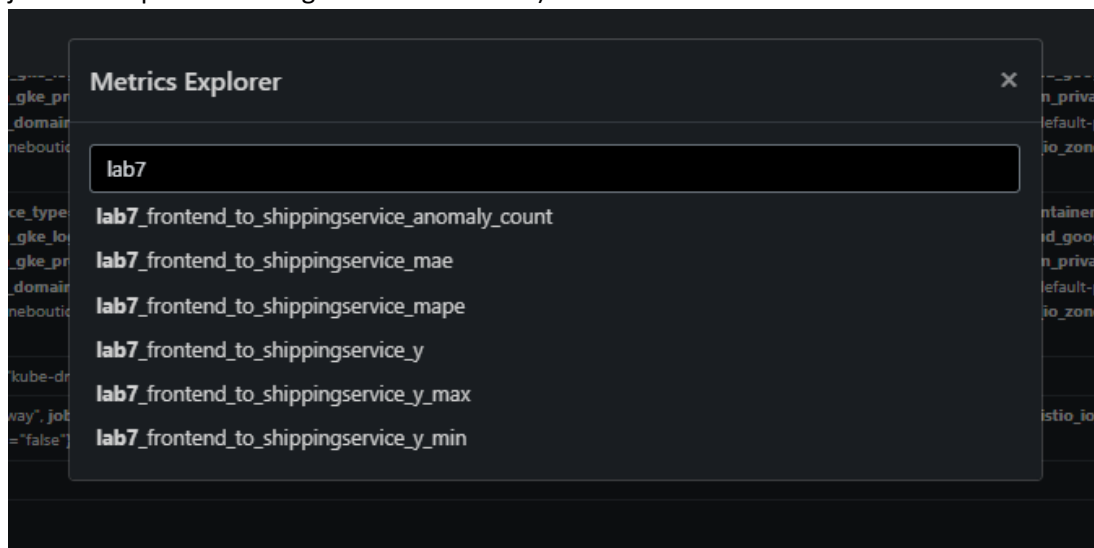
```

Timestamp Anomalies MAE MAPE
1152 2023-12-12 09:22:37.627 1 153.327726 0.876158
1153 2023-12-12 09:22:38.781 1 153.376724 0.876438
1154 2023-12-12 09:22:39.775 1 153.426650 0.876724
1155 2023-12-12 09:22:40.849 1 153.477496 0.877014
1156 2023-12-12 09:22:41.922 1 153.529251 0.877310
1157 2023-12-12 09:22:42.998 1 153.581988 0.877611
1158 2023-12-12 09:22:44.069 1 153.662563 0.878072
1159 2023-12-12 09:22:45.145 1 153.717433 0.878385
1160 2023-12-12 09:22:46.223 1 153.773170 0.878704
1161 2023-12-12 09:22:47.294 1 153.829767 0.879027
1162 2023-12-12 09:22:48.367 1 153.887212 0.879355
1163 2023-12-12 09:22:49.440 1 153.945497 0.879689
1164 2023-12-12 09:22:50.519 1 154.034476 0.880197
1165 2023-12-12 09:22:51.591 1 154.084816 0.880542
1166 2023-12-12 09:22:52.673 1 154.155959 0.880891
1167 2023-12-12 09:22:53.749 1 154.217897 0.881245
1168 2023-12-12 09:22:54.823 1 154.280618 0.881604
1169 2023-12-12 09:22:55.904 1 154.344113 0.881966
1170 2023-12-12 09:22:57.006 1 154.440781 0.882519
1171 2023-12-12 09:22:58.078 1 154.506163 0.882892
1172 2023-12-12 09:22:59.147 1 154.572281 0.883270
1173 2023-12-12 09:23:00.233 1 154.639124 0.883652
1174 2023-12-12 09:23:01.308 1 154.706682 0.884038
1175 2023-12-12 09:23:02.376 1 154.774943 0.884428
1176 2023-12-12 09:23:03.448 1 154.843897 0.884822
1177 2023-12-12 09:23:04.523 1 154.948683 0.885421
1178 2023-12-12 09:23:05.597 1 155.019240 0.885824
1179 2023-12-12 09:23:06.667 1 155.090530 0.886232
1180 2023-12-12 09:23:07.742 1 155.162463 0.886643
1181 2023-12-12 09:23:08.820 1 155.235027 0.887057

/app/monitor.py:118: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent
t datetimes in UTC: datetime.datetime.fromtimestamp(timestamp, datetime.UTC).
latest_request_datetime = datetime.datetime.fromtimestamp(float(latest_request_time))
^C
mlie@mauenya:~/AI0ps_Lab6$ kubectl logs -f frontend-shipping-monitor-64969c4d7-28x05

```

- Prometheus list of metric names created by your monitor (e.g. screen shot of pulldown with just “lab7” prefix showing all related metrics)



Task 2 deliverables

- Prometheus list of metric names from both instances running of your monitor (screenshot of all “lab7” prefix metrics as in above)

```
lab7_checkoutservice_to_paymentservice_anomaly_count
lab7_checkoutservice_to_paymentservice_mae
lab7_checkoutservice_to_paymentservice_mape
lab7_checkoutservice_to_paymentservice_y
lab7_checkoutservice_to_paymentservice_y_max
lab7_checkoutservice_to_paymentservice_y_min
lab7_frontend_to_cartservice_anomaly_count
lab7_frontend_to_cartservice_mae
lab7_frontend_to_cartservice_mape
lab7_frontend_to_cartservice_y
lab7_frontend_to_cartservice_y_max
lab7_frontend_to_cartservice_y_min
lab7_frontend_to_shippingservice_anomaly_count
lab7_frontend_to_shippingservice_mae
lab7_frontend_to_shippingservice_mape
lab7_frontend_to_shippingservice_y
lab7_frontend_to_shippingservice_y_max
lab7_frontend_to_shippingservice_y_min
```

Task 3 deliverables

1. Code for your incident detector in Python: incident/incident.py
2. Explanation of how you extended or refined the flowchart logic and why
 - a. Fetch Anomaly Counts: The code fetches anomaly counts for both services (sev1 and sev2) from Prometheus using the specified queries.
 - a. Original Flowchart: The flowchart assumes the existence of anomaly counts for services sev1 and sev2. However, it does not specify how these counts are obtained.
 - b. Extension: I added a step to fetch anomaly counts from Prometheus using specific queries. This step ensures that the incident detection is based on real-time data reflecting the anomalies in the monitored services.
 - b. Update Accumulators: The accumulators (accumulator_service1 and accumulator_service2) are updated based on the fetched anomaly counts. If an anomaly is detected, the code increments the respective accumulator by the anomaly count. If no anomaly is detected, it decrements the accumulator by 2 but ensures it never goes below zero.
 - c. Update Gauges: Prometheus gauges (sev1_gauge and sev2_gauge) are updated with the values of the respective accumulators.
 - a. Original Flowchart: The flowchart does not calculate incident temperature.
 - b. Extension: I add the code calculating the incident temperature as the sum of both accumulators. This makes it clear how the incident temperature is composed and reflects the severity of anomalies in both services.
 - d. Calculate Incident Temperature: The incident temperature is calculated as the sum of both accumulators (accumulator_service1 + accumulator_service2).

- e. Update Incident Temperature Gauge: The Prometheus gauge (`incident_temperature_gauge`) is updated with the calculated incident temperature.
 - a. Original Flowchart: The flowchart does not mention updating any gauge specifically for incident temperature.
 - b. Extension: I added a Prometheus gauge (`incident_temperature_gauge`) to visualize and monitor the incident temperature. This provides insights into the overall severity of incidents, offering a high-level perspective.
- f. Check for Incident: The code checks if the incident temperature exceeds the incident threshold. If an incident is detected, it then checks the severity based on the values of `accumulator_service1` and `accumulator_service2`. It prints incident details, including the severity.
- g. Sleep for 5 Seconds: The monitoring loop sleeps for 5 seconds before repeating the process.
- h. The threshold for the accumulators are different and act independently but their results are used for the temperature calculation.

3. Console log of running incident detector with custom load but no Istio faults. Should show no or minimal anomalies and incidents

Task 4

1. Istio delay injection YAML spec's: frontend_cartservice_delay.yaml and check_payment_delay.yaml

```
kind: VirtualService
metadata:
  name : cartservice-fault-injection
spec:
  hosts:
  - cartservice
  http:
  - fault:
      delay:
        fixedDelay: .2s
        percent: 100
      match:
      - sourceLabels:
          app: frontend
      route:
      - destination:
          host: cartservice

  - fault:
      delay:
        percent: 100
        fixedDelay: .2s
      match:
      - uri:
          regex: ".*\/GetQuote"
      route:
      - destination:
          host: cartservice

  - route:
      - destination:
          host: cartservice
```

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name : payment-service-fault-injection
spec:
  hosts:
  - payment-service
  http:
  - fault:
      delay:
        fixedDelay: .2s
        percent: 100
      match:
      - sourceLabels:
          app: checkout-service
      route:
      - destination:
          host: payment-service

  - fault:
      delay:
        percent: 100
        fixedDelay: .2s
      match:
      - uri:
          regex: ".*\/GetQuote"
      route:
      - destination:
          host: payment-service

  - route:
      - destination:
          host: payment-service
```

2. Console log showing detected anomalies and Sev 2 (for single delay) and Sev 1 (for double delay) scenarios
 - a. Double delay severity 1

[illegible]

- b. Single delay Severity 2

[illegible]

Task 5

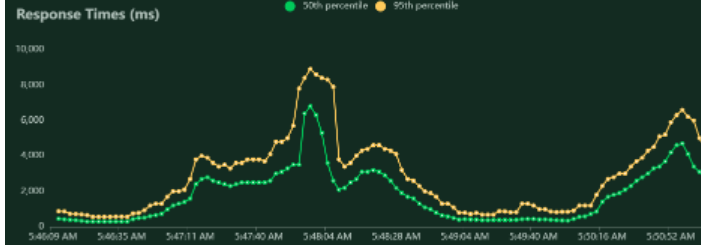
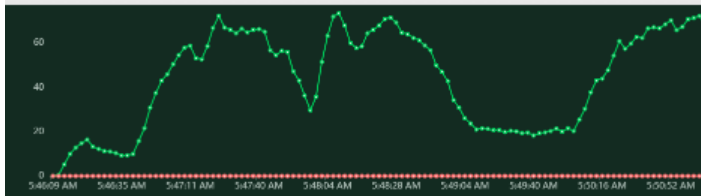
1. Console logs showing transient generation and associated anomaly and incident detection.
 - a. Applying 400 users between 150 to 250

```
applying 400 users
run time = 160.16072475199985
current users = 150
last_target_users = 50
applying 400 users
run time = 161.1613818100002
current users = 160
last_target_users = 50
applying 400 users
run time = 162.16167483499999
current users = 170
last_target_users = 50
applying 400 users
run time = 163.16227496999977
current users = 180
last_target_users = 50
applying 400 users
run time = 164.16296954799986
current users = 190
last_target_users = 50
applying 400 users
run time = 165.16388169200013
current users = 200
last_target_users = 50
applying 400 users
run time = 166.1648853250001
current users = 210
last_target_users = 50
applying 400 users
run time = 167.16572748800013
current users = 220
last_target_users = 50
applying 400 users
run time = 168.16607656300016
current users = 230
last_target_users = 50
applying 400 users
run time = 169.16629817299986
current users = 240
last_target_users = 50
applying 400 users
```



7





	Timestamp	Anomalies	MAE	MAPE
13557	2023-12-14 21:45:30.597	1	22.029529	0.701700
13558	2023-12-14 21:45:31.777	1	22.119621	0.704570
13559	2023-12-14 21:45:32.947	1	22.209311	0.707427
13560	2023-12-14 21:45:34.142	1	22.343068	0.711687
13561	2023-12-14 21:45:35.465	1	22.431706	0.714511
13562	2023-12-14 21:45:36.634	1	22.563837	0.718719
13563	2023-12-14 21:45:37.765	1	22.651359	0.721507
13564	2023-12-14 21:45:38.970	1	22.738416	0.724280
13565	2023-12-14 21:45:40.226	1	22.868107	0.728411
13566	2023-12-14 21:45:41.353	1	22.953957	0.731146
13567	2023-12-14 21:45:42.636	0	10.957700	0.568628
13568	2023-12-14 21:45:43.784	0	11.042279	0.573017
13569	2023-12-14 21:45:44.926	0	11.126332	0.577379
13570	2023-12-14 21:45:46.185	0	11.251403	0.583869
13571	2023-12-14 21:45:47.323	0	11.334096	0.588160
13572	2023-12-14 21:45:48.477	0	11.416228	0.592422
13573	2023-12-14 21:45:49.606	0	11.538351	0.598760
13574	2023-12-14 21:45:50.821	0	11.619035	0.602947
13575	2023-12-14 21:45:52.201	0	11.738940	0.609169
13576	2023-12-14 21:45:53.320	0	11.818115	0.613278
13577	2023-12-14 21:45:54.472	0	11.896668	0.617354
13578	2023-12-14 21:45:55.707	0	12.013313	0.623407
13579	2023-12-14 21:45:56.905	0	9.894891	0.579494
13580	2023-12-14 21:45:58.219	0	10.846971	0.605539
13581	2023-12-14 21:45:59.453	0	10.922273	0.609743
13582	2023-12-14 21:46:00.617	0	11.033950	0.615977
13583	2023-12-14 21:46:01.764	0	11.107536	0.620085
13584	2023-12-14 21:46:02.893	0	11.180420	0.624154
13585	2023-12-14 21:46:04.125	0	11.288407	0.630183
13586	2023-12-14 21:46:05.440	0	11.359493	0.634151

/app/monitor.py:100: DeprecationWarning: datetime.datetime.utcfromtimestamp() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.fromtimestamp(timestamp, datetime.UTC)

[illegible]

- c. Applying 100 users between 500 and 550 seconds

```

applying 1000 users
run time = 525.4821235699999
current users = 450
last_target_users = 200
applying 1000 users
run time = 526.4836650079999
current users = 460
last_target_users = 200
applying 1000 users
run time = 527.4843648939998
current users = 470
last_target_users = 200
applying 1000 users
run time = 528.485383938
current users = 480
last_target_users = 200
applying 1000 users
run time = 529.485862541
current users = 490
last_target_users = 200
applying 1000 users
run time = 530.486632594
current users = 500
last_target_users = 200
applying 1000 users
run time = 531.4874863920002
current users = 510
last_target_users = 200
applying 1000 users
run time = 532.4879643679999
current users = 520
last_target_users = 200
applying 1000 users
run time = 533.4887514989998
current users = 530
last_target_users = 200
applying 1000 users
run time = 534.489794504
current users = 540
last_target_users = 200
applying 1000 users

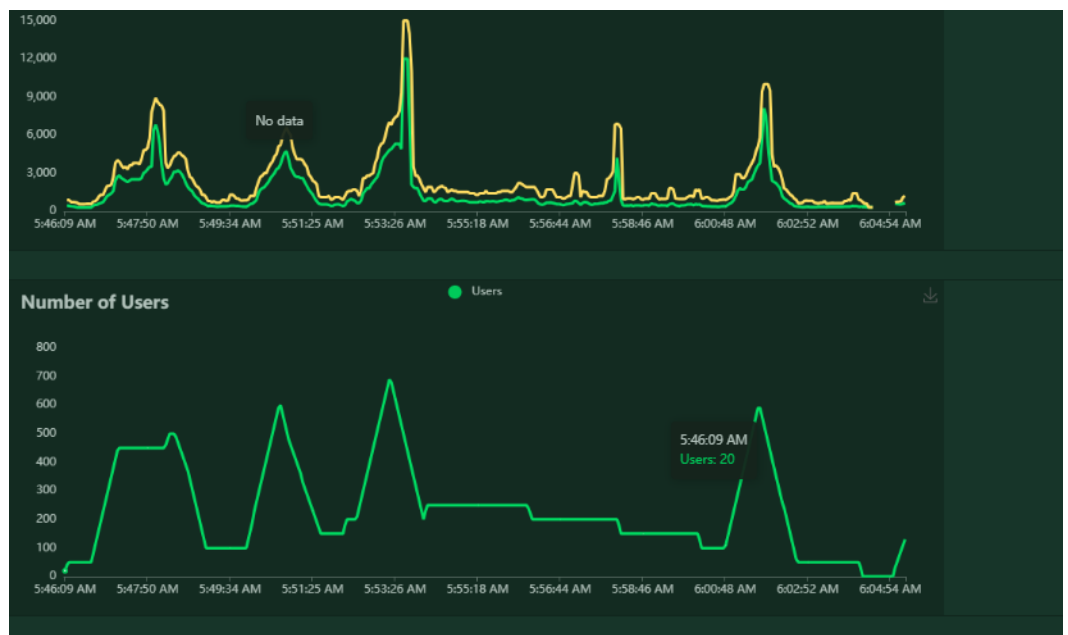
```



```

last_target_users = 100
applying 800 users
run time = 1026.9164231890002
current users = 360
last_target_users = 100
applying 800 users
run time = 1027.917304459
current users = 370
last_target_users = 100
applying 800 users
run time = 1028.917880316
current users = 380
last_target_users = 100
applying 800 users
run time = 1029.9193592180002
current users = 390
last_target_users = 100
applying 800 users
run time = 1030.9194360029999
current users = 400
last_target_users = 100
applying 800 users
run time = 1031.920662142
current users = 410
last_target_users = 100
applying 800 users
run time = 1032.921773765
current users = 420
last_target_users = 100
applying 800 users
run time = 1033.9227960219998
current users = 430
last_target_users = 100
applying 800 users

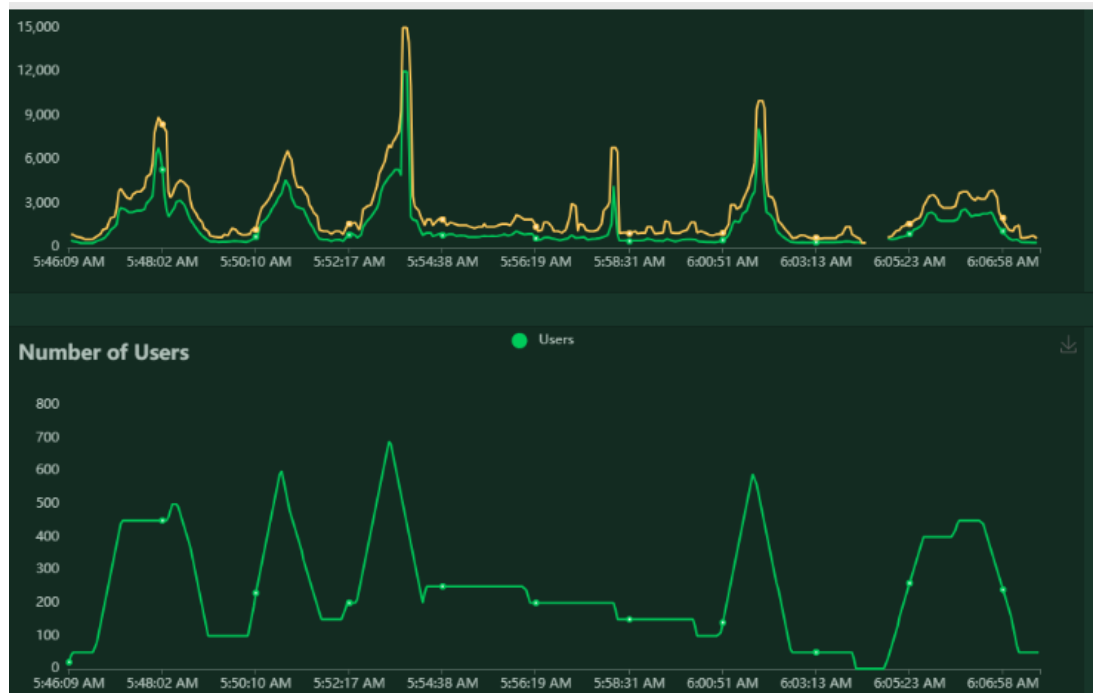
```



- e. Applying 400 users between 1250 secs and 1350 secs

```
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected  
No incident detected
```

```
last_target_users = 50  
applying 400 users  
run time = 1343.191535624  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1344.1920307029998  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1345.1930334770002  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1346.1938470809998  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1347.193917453  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1348.194725607  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1349.1954926800001  
current users = 450  
last_target_users = 50  
applying 400 users  
run time = 1350.19592068  
current users = 450  
last_target_users = 50
```



f. Severity Changes

```
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
Incident detected! Severity: Sev 2
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
No incident detected
```

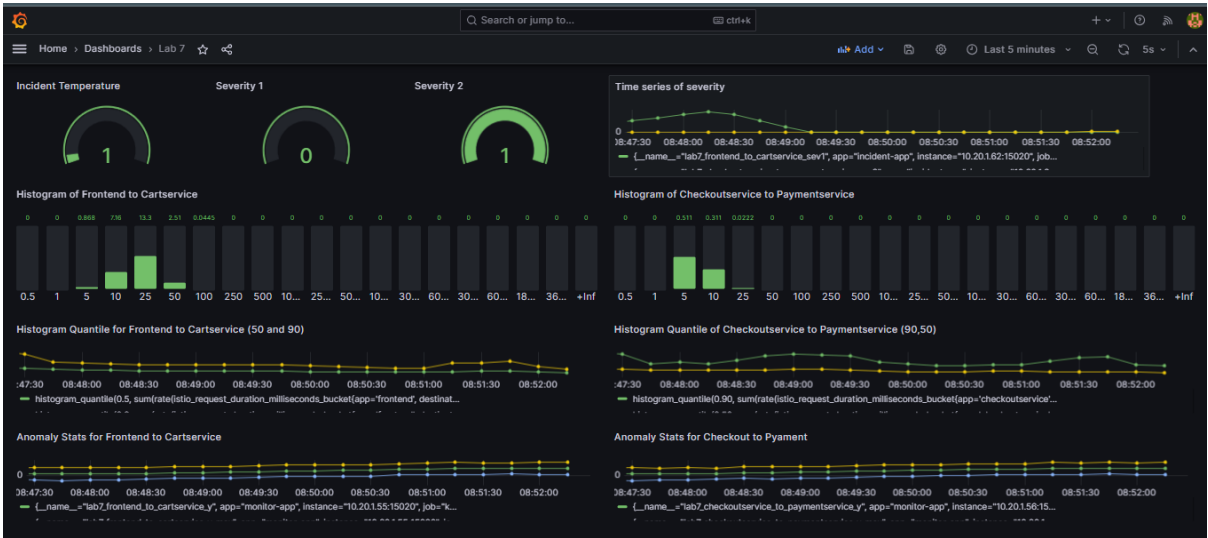
g. Final look on chart



2. Locustfile source showing the transient profile you have chosen and implemented:
locust_step_template.py
3. Explanation of how you configured the transients and why
In configuring the transients for the load generator, I carefully selected parameters to simulate diverse scenarios of transient anomalies. The configuration involves three key aspects:
 - a. **Frequency:** I defined five distinct ranges, each specifying a different frequency range for transient anomalies. These ranges, represented by min_run_secs and max_run_secs, dictate when transient anomalies occur during the test.
 - b. **Height:** The intensity or height of transient anomalies is controlled by the ANOMALY_SURGES list. Each element in this list corresponds to the surge in users during an anomaly. Varied surge values enable the simulation of different levels of system stress.
 - c. **Duration:** Duration is determined by the range specified in ANOMALY_RANGES. The difference between min_run_secs and max_run_secs in each range defines how long each anomaly lasts.

Task 6


1. Grafana Dashboard



Task 7

If you want to cancel all activity or usage, make a payment on [Payment overview](#) to cover any remaining balance, and then visit [Account management](#) to close the billing account.

PRIMARY




Visa •••• 2399
Expires 07/28

Primary

RemoveEdit

BACKUP

PRIMARY

	Visa •••• 3039 Expires 06/25
Primary	Remove Edit

BACKUP

[+ Add a backup payment method](#)

When the primary payment method fails, a backup payment method pays the balance automatically. [Learn more](#)