CMM707 – Cloud Computing

Coursework Report

Name: Avishka Vithanage

RGU ID: 2506663

IIT ID: 20241889

Introduction

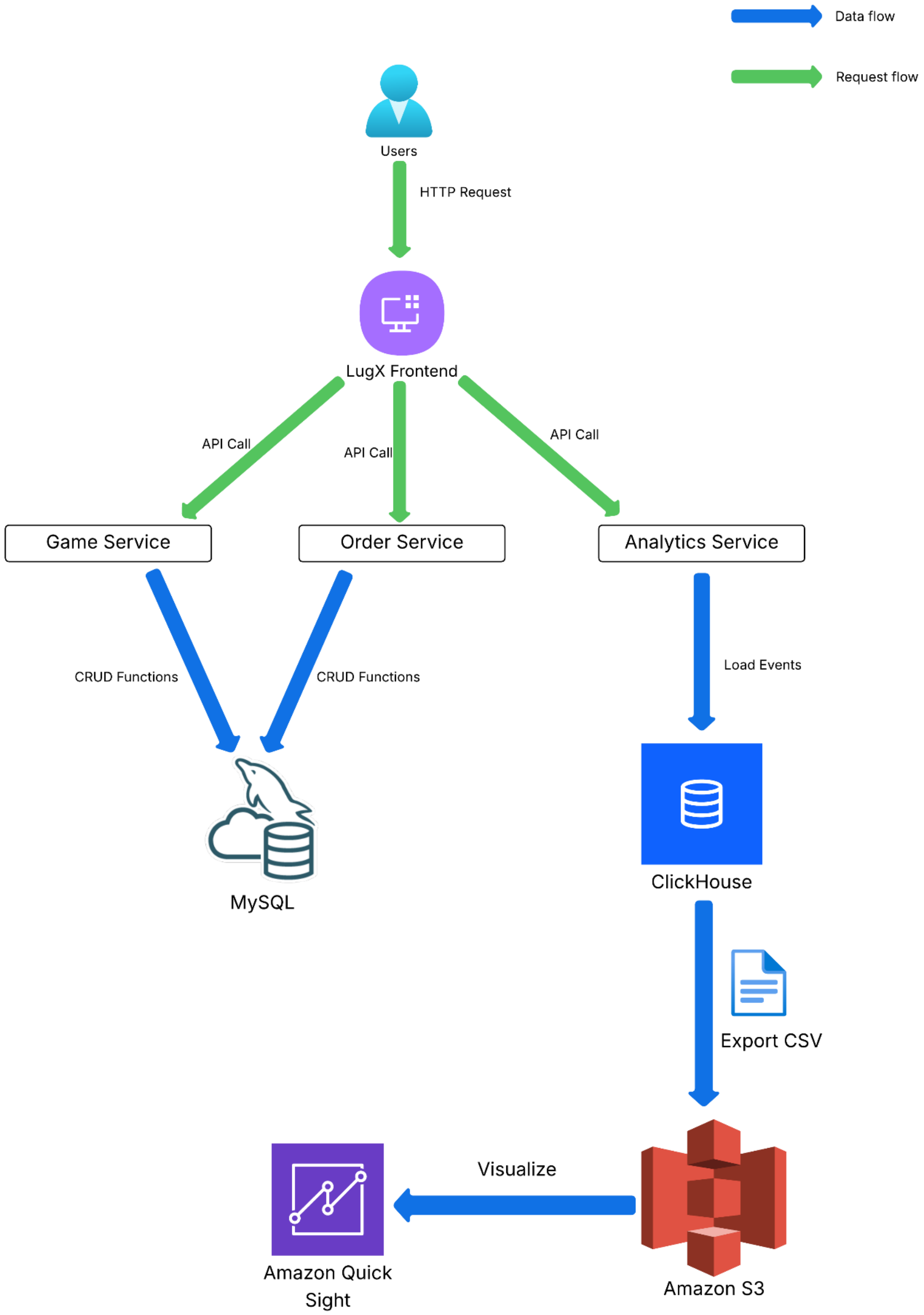
The design and implementation of a microservices-based deployment for the Lugx Gaming platform, created as a component of the CMM707 Cloud Computing course, are presented in this report. In keeping with the module's ability to use both local and cloud-native Kubernetes environments, the solution was created to run exclusively on a locally deployed Kubernetes cluster using Minikube. This strategy avoided the expense and account suspension risks that are occasionally connected to public cloud accounts near assessment deadlines while guaranteeing complete functionality.

The platform architecture consists of multiple containerized microservices:

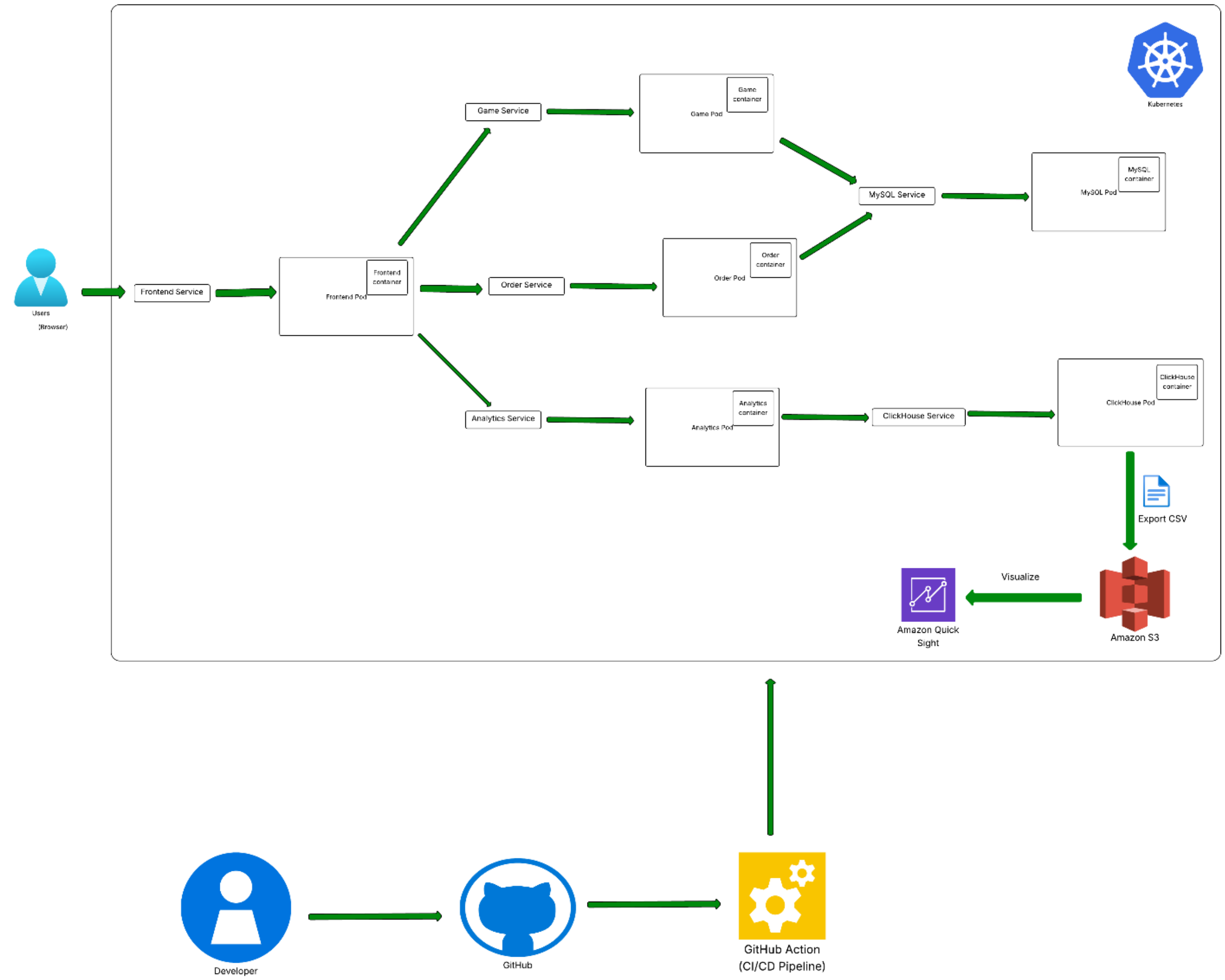
- **Frontend** - a static web application serving the Lugx Gaming user interface.
- **Game Service** - a REST API for storing and updating game details.
- **Order Service** - a REST API for managing orders and cart transactions.
- **Analytics Service** - an API integrated with ClickHouse to capture web analytics events such as page views and clicks.

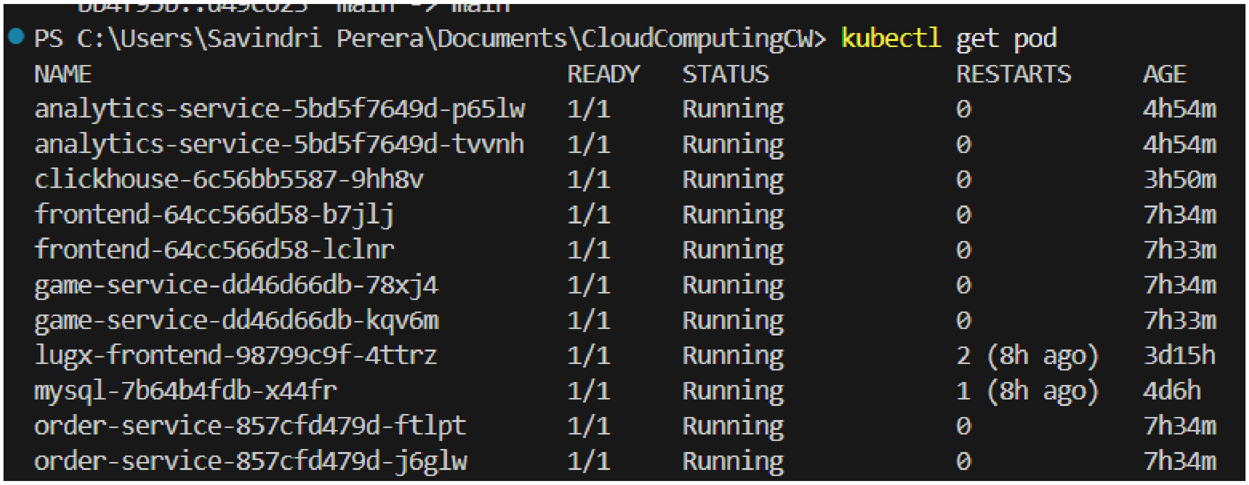
Docker was used to containerize each microservice, and Kubernetes manifests were used to deploy them to Minikube. Grafana and Prometheus were used to achieve observability, and ClickHouse was used to store and visualize analytics data using AWS QuickSight. Utilizing a CI/CD pipeline with GitHub Actions to automate builds, tests, and deployments to the local cluster, the deployment adhered to best practices for scalability, resilience, and maintainability. To ensure high availability, rolling updates and optional blue-green deployment techniques were included.

Solution Architecture Diagram



Deployment Architecture Diagram





```
   bb4f95b..d49c023  main -> main
● PS C:\Users\Savindri Perera\Documents\CloudComputingCW> kubectl get pod
  NAME                                 READY   STATUS    RESTARTS     AGE
  analytics-service-5bd5f7649d-p65lw   1/1     Running   0            4h54m
  analytics-service-5bd5f7649d-tvvnh   1/1     Running   0            4h54m
  clickhouse-6c56bb5587-9hh8v          1/1     Running   0            3h50m
  frontend-64cc566d58-b7jlj            1/1     Running   0            7h34m
  frontend-64cc566d58-lclnr            1/1     Running   0            7h33m
  game-service-dd46d66db-78xj4         1/1     Running   0            7h34m
  game-service-dd46d66db-kqv6m         1/1     Running   0            7h33m
  lugx-frontend-98799c9f-4ttrz         1/1     Running   2 (8h ago)   3d15h
  mysql-7b64b4fdb-x44fr                1/1     Running   1 (8h ago)   4d6h
  order-service-857cfd479d-ftlpt       1/1     Running   0            7h34m
  order-service-857cfd479d-j6glw       1/1     Running   0            7h34m
```

Security and Ethics Challenges in the Lugx Gaming Platform

Security Risks

- Insecure API Endpoints: Public APIs (like /orders, /track) that are not properly authenticated are vulnerable to misuse or illegal access.
- SQL Injection or Input Exploits: Attackers may run malicious SQL or script commands if input is not cleaned up.
- Data Leakage: HTTPS must be used to securely store and transfer sensitive data, such as session IDs and payment-related metadata.
- Improper Secret Handling: It is possible to extract and abuse hard-coded secrets or API tokens from code or Docker files.

Ethical Concerns

- Privacy Violations: Without clear user consent, tracking user clicks, session length, and scroll depth may violate privacy laws and expectations.
- Over-collection of Data: It may be unethical to record more user behavior than is required, particularly in the absence of a clear goal.
- Data Retention Without Justification: It is unethical to keep web activity logs for an extended period as this raises risks and goes against ethical data minimization guidelines.

Mitigation Strategies

- Secure APIs with appropriate authorization and authentication.
- Use environment variables or Kubernetes Secrets to secure secrets and encrypt all traffic using HTTPS.
- Clearly explain analytics collection in a consent banner or user-facing privacy policy.
- Establish time-based guidelines for analytics log data retention.

CI/CD pipeline Designs Diagram

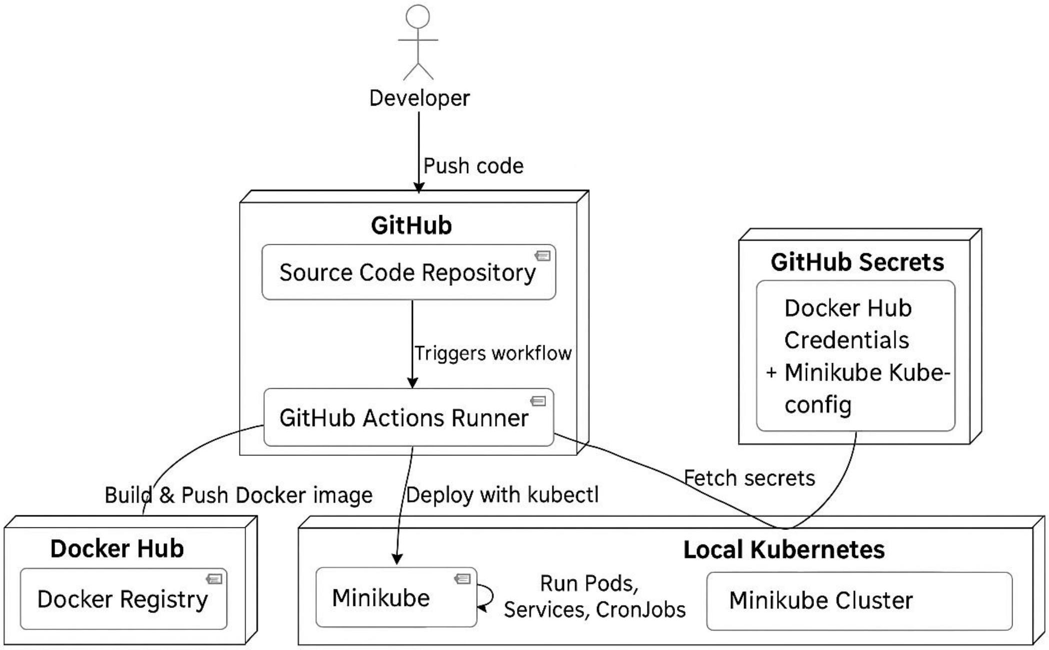


Figure 1 C/CD Pipeline Diagram

Process Description

The Lugx Gaming platform's CI/CD pipeline has been set up to operate with a local Kubernetes cluster that is powered by Minikube. The steps are as follows:

1. Trigger: Code pushed to the main branch or manually dispatched via workflow initiates the pipeline.

2. Build: A GitHub Actions runner builds Docker images for each microservice (Frontend, Game Service, Order Service, and Analytics Service) after checking out the repository.

3. Push: Using credentials safely kept in GitHub Secrets, the constructed images are tagged and pushed to Docker Hub.

4. Deploy: The runner uses a base64-encoded kubeconfig that is kept in GitHub Secrets to establish a connection to the nearby Minikube cluster. Update deployments use Kubernetes manifests.

5. Deployment Strategy: By defaulting to rolling updates, there is no downtime. For safer version switching, a Blue-Green deployment strategy is also supported.

6. Testing: Automated smoke tests examine service health endpoints following deployment. Periodically, integration tests are conducted to confirm functionality.

7. Scheduled Runs: To guarantee ongoing system health, the workflow is set to run every night.

8. Monitoring: To confirm a successful deployment, logs, Grafana dashboards, and rollout status are utilized.

Security and Ethics in CI/CD and Cloud Deployment

CI/CD and Cloud Security Challenges

- The storage of AWS credentials, database passwords, or tokens in plaintext within docker-compose.yml or CI files (such as GitHub Actions) can lead to significant security breaches.

- Absence of Pipeline Integrity: Malicious code may infiltrate the production environment if the pipeline is not secured by version control permissions or does not enforce commit signatures.

- Misconfigured Ingress or Open Kubernetes Ports: When Kubernetes services are not configured properly, they may be exposed to the outside world, which can lead to brute-force or denial-of-service (DoS) attacks.

- Absence of Audit Logging: Deployments and configuration modifications are not accountable in the absence of audit logs.

CI/CD and Cloud Ethical Concerns

- Abuse of Continuous Testing: If integration tests are periodically started without throttle controls, they may inadvertently overload services or waste cloud resources.

- Deploying Without Verification: When updates are released without user approval, end users may experience a worsening of their experience or encounter bugs.

- Ignoring Cost and Environmental Impact: Over-provisioned cloud resources and inefficient build pipelines lead to excessive energy use, which raises sustainability-related ethical questions.

Mitigation Strategies

- Make use of safe secret management solutions, such as AWS Secrets Manager and GitHub Secrets.
- For CI/CD service accounts, apply the least privilege principle.
- To keep track of deployment activities, including audit logging and monitoring.

Utilize scheduling and cost optimization tools to handle computation in an ethical manner.

CI/CD Pipeline Scripts

Build.yml

```
name: Build & Push Images

on:
  push:
    branches: [ main ]
    paths:
      - "frontend/**"
      - "game-service/**"
      - "order-service/**"
      - "analytics-service/**"
      - ".github/workflows/build.yml"
  workflow_dispatch:

jobs:
  build:
    # use GitHub's ubuntu runner for clean, fast Docker builds
    runs-on: ubuntu-latest

    strategy:
      matrix:
        service:
          - frontend
          - game-service
          - order-service
```

```yaml
          - analytics-service

    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Log in to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3

      - name: Build & Push ${{ matrix.service }}
        uses: docker/build-push-action@v5
        with:
          context: ./${{ matrix.service }}
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/${{ matrix.service }}:latest
          cache-from: type=gha
          cache-to: type=gha,mode=max
```

deploy.yml

```
name: Deploy & Test

on:
  workflow_run:
    workflows: ["Build & Push Images"]
    types: [completed]
  workflow_dispatch:

jobs:
  deploy:
    runs-on: self-hosted

    steps:
      - name: Checkout
        uses: actions/checkout@v3

      # kubectl on Windows self-hosted runner
      - name: Install kubectl (Windows)
        shell: powershell
        run: choco install kubernetes-cli -y

      - name: Configure kubeconfig
        shell: powershell
        run: |
          New-Item -ItemType Directory -Force -Path "$HOME\.kube" | Out-Null
          # write kubeconfig from secret (already base64)
          [IO.File]::WriteAllBytes("$HOME\.kube\config",[Convert]::FromBase64String("${{ secrets.KUBE_CONFIG_DATA }}"))

      - name: Set images to latest (no YAML changes required)
        shell: powershell
        run: |
          kubectl set image deploy/game-service      game-service=${{ secrets.DOCKERHUB_USERNAME }}/game-service:latest
          kubectl set image deploy/order-service     order-service=${{ secrets.DOCKERHUB_USERNAME }}/order-service:latest
          kubectl set image deploy/analytics-service analytics-service=${{ secrets.DOCKERHUB_USERNAME }}/analytics-service:latest
          kubectl set image deploy/frontend          frontend=${{ secrets.DOCKERHUB_USERNAME }}/frontend:latest

      - name: Wait for rollouts (rolling update, zero downtime)
        shell: powershell
        run: |
          kubectl rollout status deploy/game-service      --timeout=180s
          kubectl rollout status deploy/order-service     --timeout=180s
          kubectl rollout status deploy/analytics-service --timeout=180s
          kubectl rollout status deploy/frontend          --timeout=180s

      - name: Smoke test – analytics POST
        shell: powershell
        run: |
          # forward analytics-service:8000 -> localhost:18002
          $pf = Start-Process -PassThru -WindowStyle Hidden powershell -ArgumentList
'kubectl port-forward svc/analytics-service 18002:8000'
          Start-Sleep -Seconds 3

          $payload = @{
            event_type = "page_view"
            page_url   = "/"
            user_agent = "ci-test"
            session_id = "ci-${{ github.run_id }}"
            ts         = (Get-Date).ToString("o")
          } | ConvertTo-Json

          try {
            $res = Invoke-RestMethod -Uri http://localhost:18002/track -Method POST -
ContentType 'application/json' -Body $payload
            Write-Host "Analytics POST OK"
          } finally {
            Stop-Process -Id $pf.Id -Force
          }

      - name: Smoke test – frontend HTTP 200
        shell: powershell
        run: |
          # get NodePort for 'frontend'
          $nodePort = kubectl get svc frontend -o jsonpath='{.spec.ports[0].nodePort}'
          $url = "http://127.0.0.1:$nodePort"
```

```
Write-Host "Testing $url"
$r = Invoke-WebRequest -Uri $url -UseBasicParsing
if ($r.StatusCode -ne 200) { throw "Frontend returned $($r.StatusCode)" }
Write-Host "Frontend OK"
```

Runbook (Deployment Steps)

1. The developer manually initiates workflow or pushes code changes to the main.

2. Docker images for updated services are created and pushed by GitHub Actions.

3. Workflow applies the most recent Kubernetes manifests after connecting to Minikube.

4. Without any downtime, rolling updates swap out outdated pods for new ones.

5. Service availability is verified by automated smoke tests.

6. The developer uses Kubectl get pods and Grafana to confirm rollout.

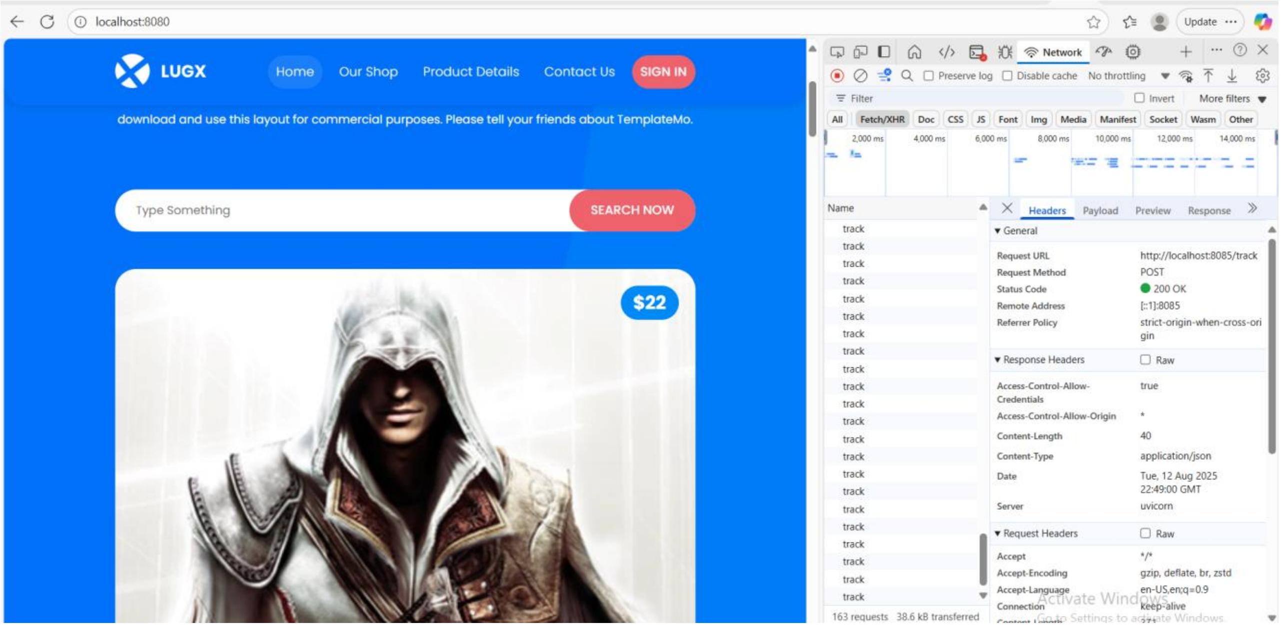
7. If issues arise, rollback using:

```
kubectl rollout undo deploy/<service>
```

GitHub Repo

[AvishkaPereraV/cmm707-cloudcomputing-cw: CMM707 - Cloud Computing Coursework]()

Web analytics + ClickHouse



▾ General

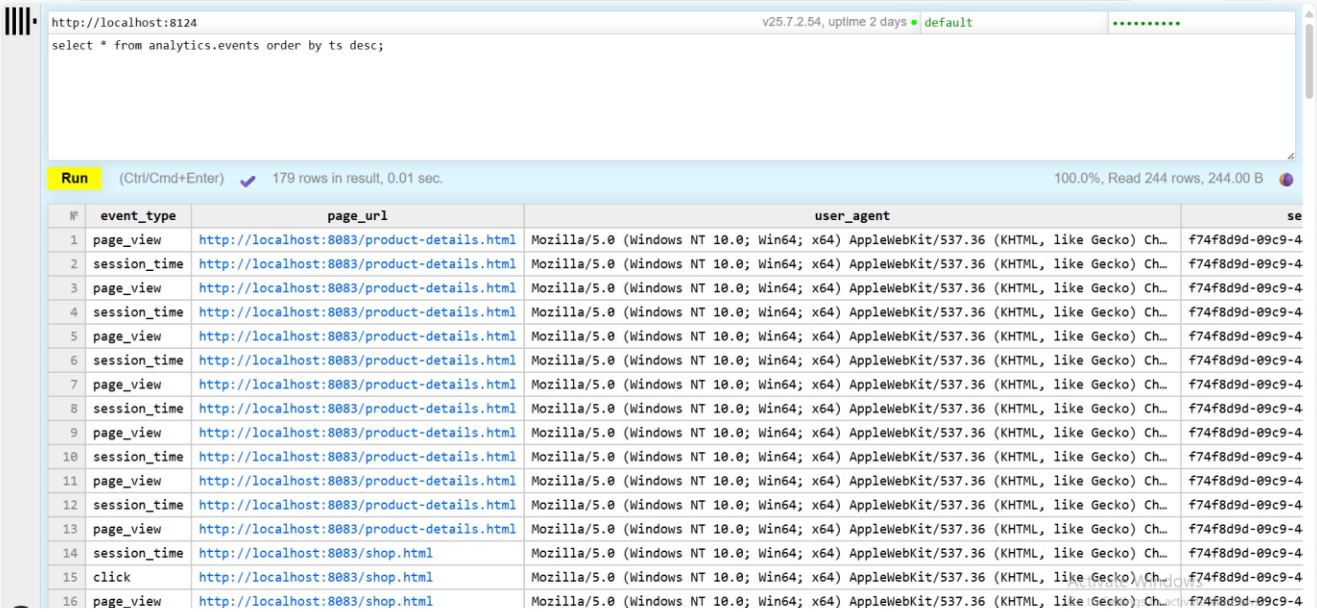
Request URL	http://localhost:8085/track
Request Method	POST
Status Code	● 200 OK
Remote Address	[::1]:8085
Referrer Policy	strict-origin-when-cross-origin

▾ Response Headers ☐ Raw

Access-Control-Allow-Credentials	true
Access-Control-Allow-Origin	*
Content-Length	40
Content-Type	application/json
Date	Tue, 12 Aug 2025 22:49:00 GMT
Server	uvicorn

▾ Request Headers ☐ Raw

Accept	*/*
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en-US,en;q=0.9
Connection	keep-alive

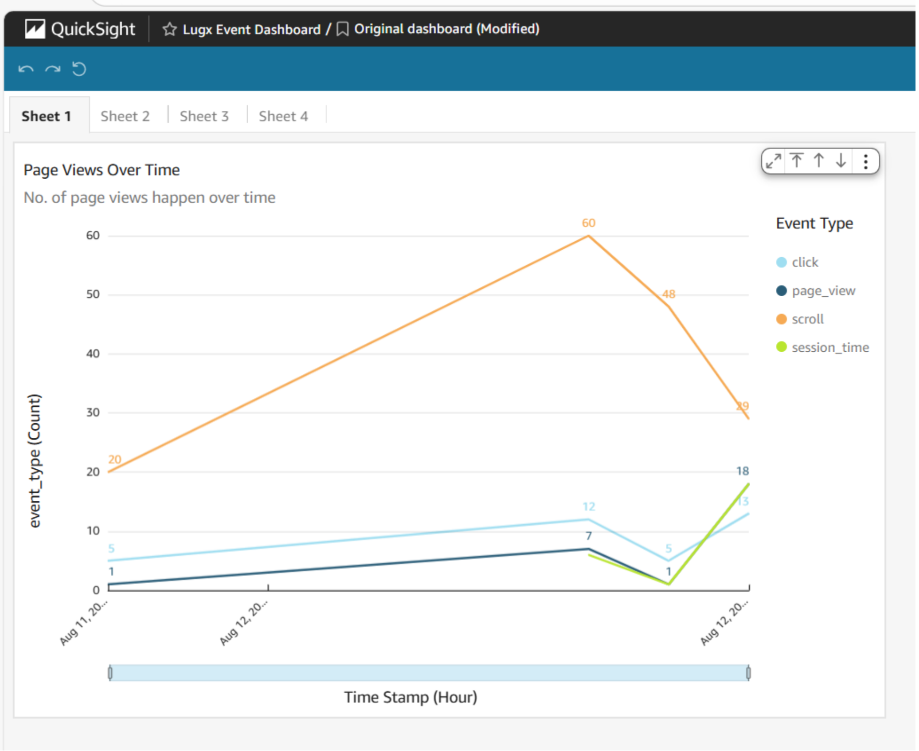


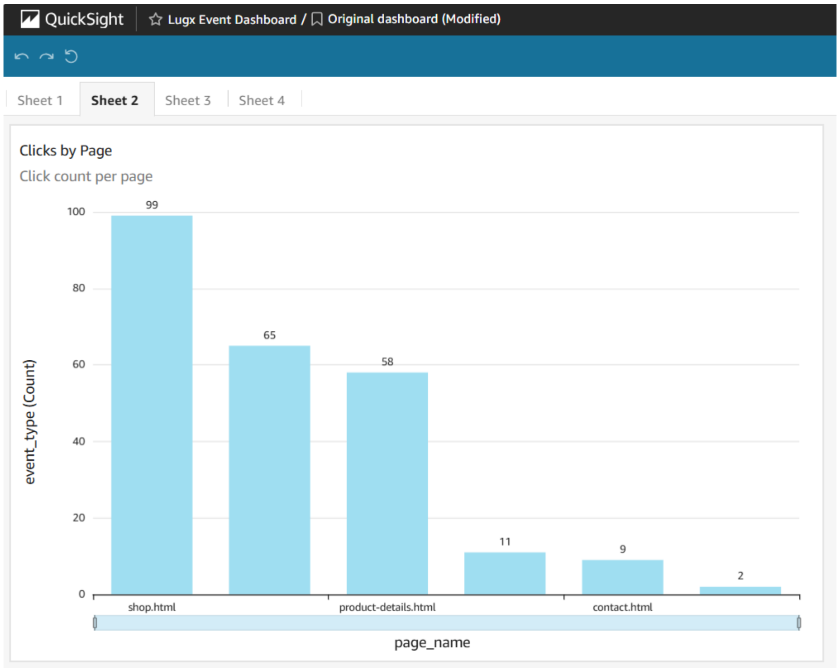
http://localhost:8124
v25.7.2.54, uptime 2 days ● default
••••••••••
Run
(Ctrl/Cmd+Enter)
179 rows in result, 0.01 sec.
100.0%, Read 244 rows, 244.00 B

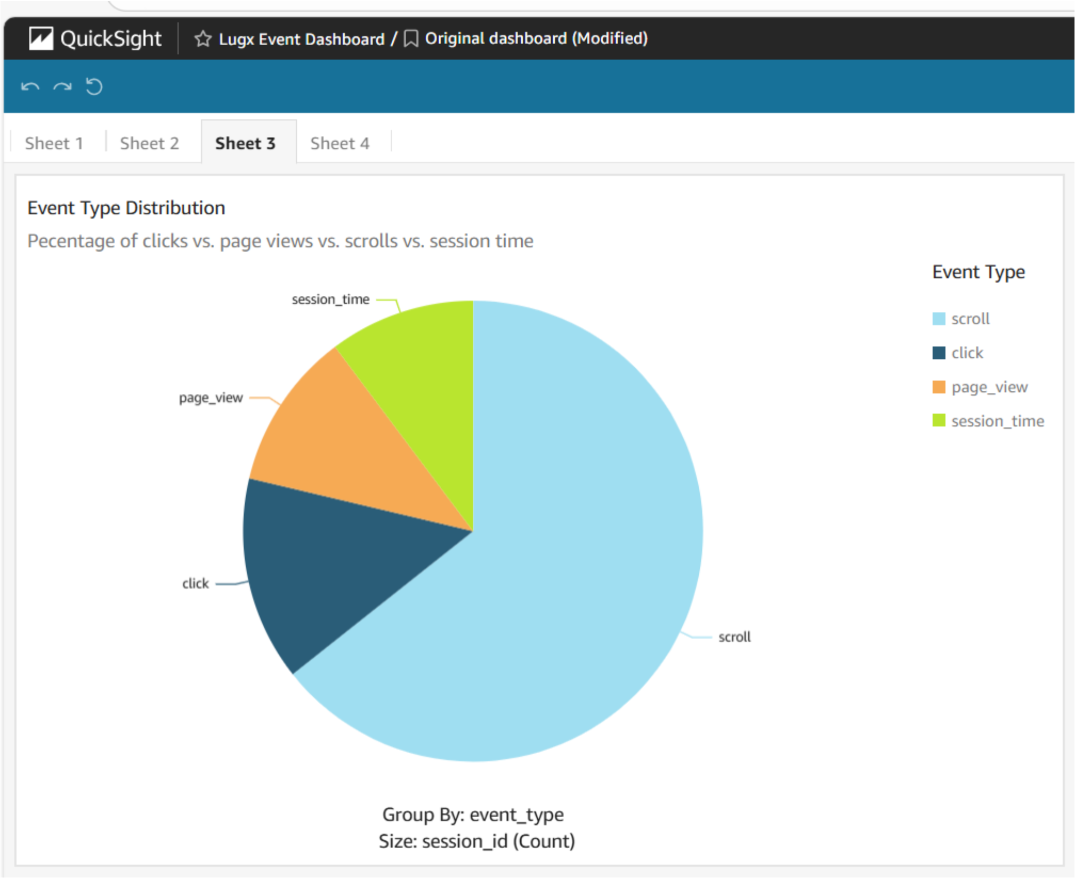
```
select * from analytics.events order by ts desc;
```

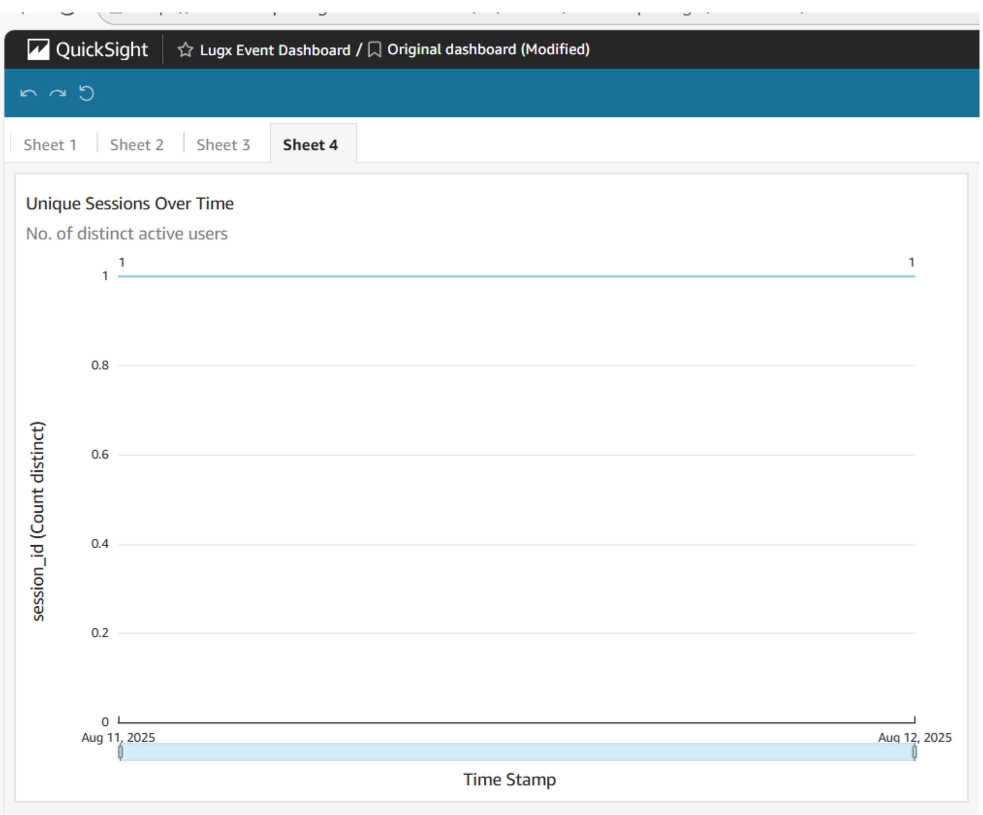
№	event_type	page_url	user_agent	se
1	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
2	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
3	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
4	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
5	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
6	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
7	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
8	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
9	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
10	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
11	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
12	session_time	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
13	page_view	http://localhost:8083/product-details.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
14	session_time	http://localhost:8083/shop.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
15	click	http://localhost:8083/shop.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4
16	page_view	http://localhost:8083/shop.html	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch…	f74f8d9d-09c9-4

ClickHouse + QuickSight Visualizations

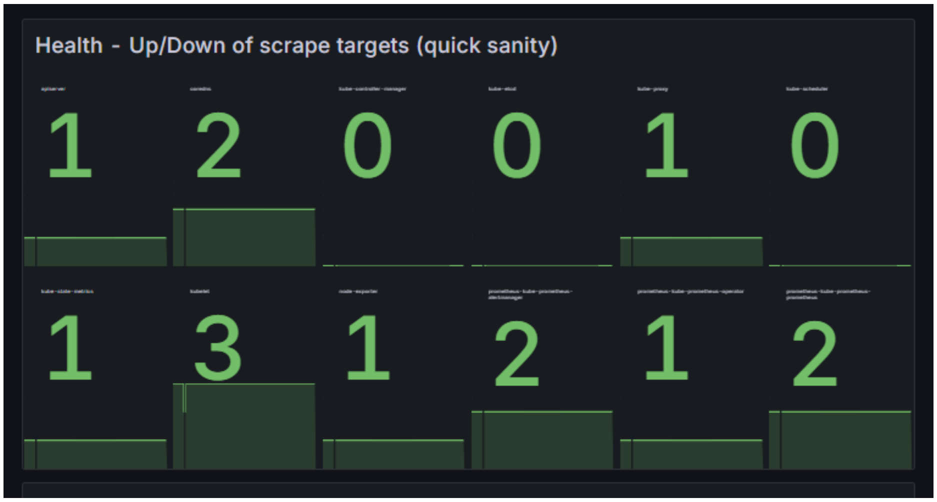


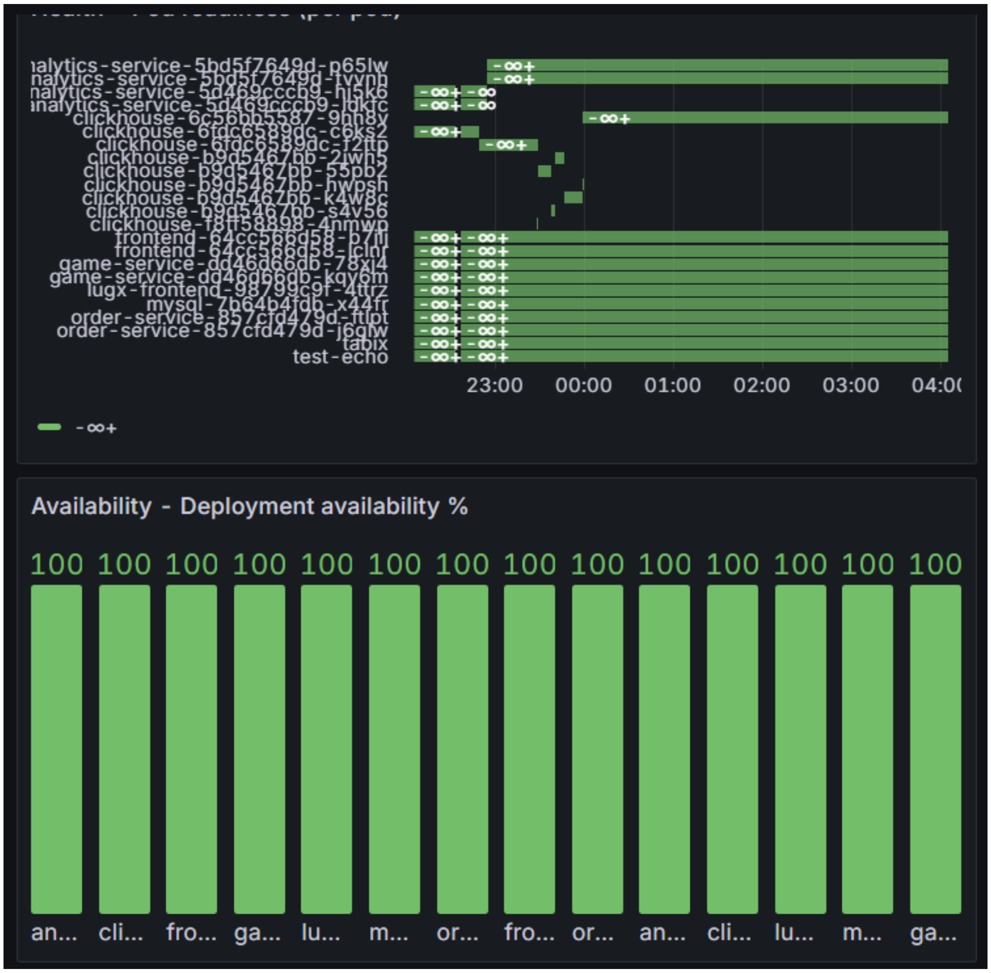

QuickSight
Lugx Event Dashboard / Original dashboard (Modified)
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Clicks by Page
Click count per page
event_type (Count)
100
80
60
40
20
0
99
65
58
11
9
2
shop.html
product-details.html
contact.html
page_name

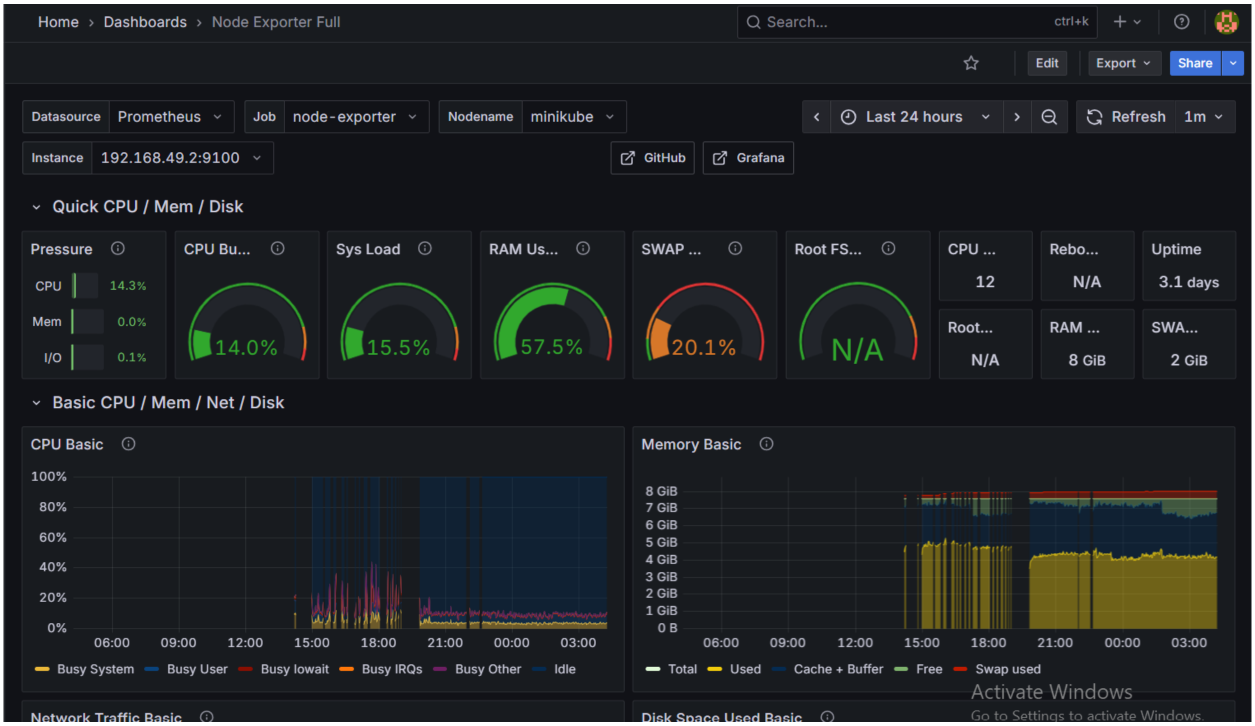

QuickSight
Lugx Event Dashboard / Original dashboard (Modified)
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Event Type Distribution
Pecentage of clicks vs. page views vs. scrolls vs. session time
session_time
page_view
click
scroll
Event Type
scroll
click
page_view
session_time
Group By: event_type
Size: session_id (Count)


QuickSight
Lugx Event Dashboard / Original dashboard (Modified)
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Unique Sessions Over Time
No. of distinct active users
1
1
1
0.8
0.6
0.4
0.2
0
session_id (Count distinct)
Aug 11, 2025
Aug 12, 2025
Time Stamp

Prometheus + Grafana Visualizations


Health - Up/Down of scrape targets (quick sanity)
apiserver
1
coredns
2
kube-controller-manager
0
kube-etcd
0
kube-proxy
1
kube-scheduler
0
kube-state-metrics
1
kubelet
3
node-exporter
1
prometheus-kube-prometheus-alertmanager
2
prometheus-kube-prometheus-operator
1
prometheus-kube-prometheus-prometheus
2


Health - Pod readiness (per pod)
ıalytics-service-5bd5f7649d-p65lw
nalytics-service-5bd5f7649d-tvynh
nalytics-service-5d469cccb9-hj5k6
ınalytics-service-5d469cccb9-ldkfc
clickhouse-6c56bb5587-9hh8y
clickhouse-6fdc6589dc-c6ks2
clickhouse-6fdc6589dc-f2ttp
clickhouse-b9d5467bb-2jwh5
clickhouse-b9d5467bb-55pb2
clickhouse-b9d5467bb-hwpsh
clickhouse-b9d5467bb-k4w8c
clickhouse-b9d5467bb-s4v56
clickhouse-f8ff58898-4nmwp
frontend-64cc566d58-b7jlj
frontend-64cc566d58-lclnr
game-service-dd46d66db-78xj4
game-service-dd46d66db-kqy6m
lugx-frontend-98799c9f-4ttrz
mysql-7b64b4fdb-x44fr
order-service-857cfd479d-ftlpt
order-service-857cfd479d-j6qlw
tabix
test-echo
-∞+
-∞
23:00
00:00
01:00
02:00
03:00
04:0(
-∞+
Availability - Deployment availability %
100
100
100
100
100
100
100
100
100
100
100
100
100
100
an...
cli...
fro...
ga...
lu...
m...
or...
fro...
or...
an...
cli...
lu...
m...
ga...


Home › Dashboards › Node Exporter Full

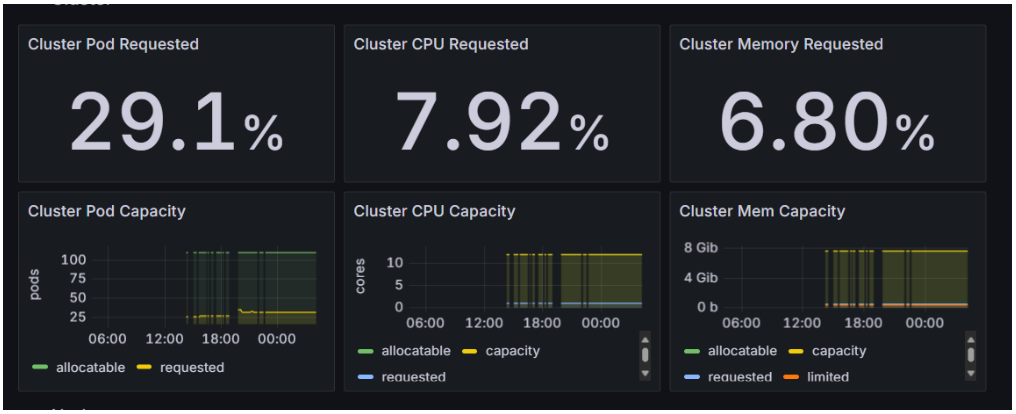
Datasource: Prometheus
Job: node-exporter
Nodename: minikube
Instance: 192.168.49.2:9100
Last 24 hours
Refresh 1m

Quick CPU / Mem / Disk

Pressure	CPU Bu...	Sys Load	RAM Us...	SWAP ...	Root FS...
CPU 14.3% Mem 0.0% I/O 0.1%	14.0%	15.5%	57.5%	20.1%	N/A

CPU ...	Rebo...	Uptime
12	N/A	3.1 days
Root...	**RAM ...**	**SWA...**
N/A	8 GiB	2 GiB

Basic CPU / Mem / Net / Disk



Cluster Pod Requested

29.1%

Cluster CPU Requested

7.92%

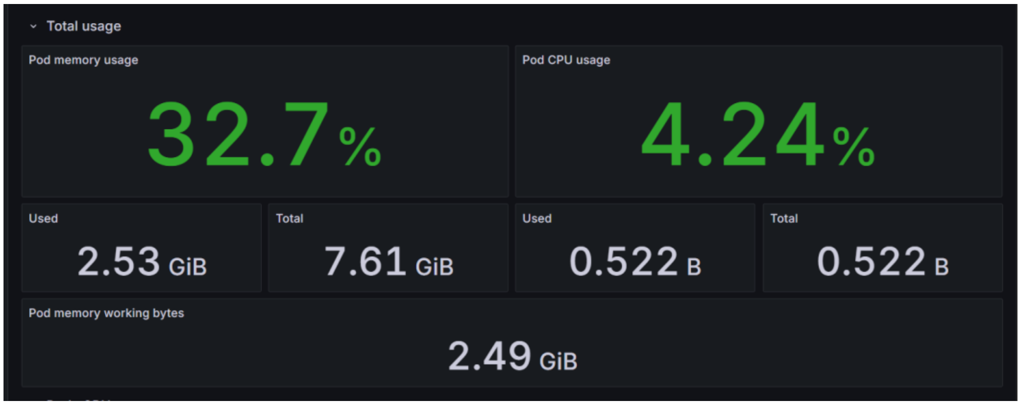
Cluster Memory Requested

6.80%

Cluster Pod Capacity

Cluster CPU Capacity

Cluster Mem Capacity



Total usage

Pod memory usage

32.7%

Pod CPU usage

4.24%

Used

2.53 GiB

Total

7.61 GiB

Used

0.522 B

Total

0.522 B

Pod memory working bytes

2.49 GiB