Name : Hamza Jamshaid

Project tittle:

Cloud Monitoring and Log Analysis

Introduction:

The Internee.pk system is a web application used by interns and companies to manage internship processes. To ensure its performance, reliability, and security, we implemented **cloud monitoring and log analysis** using **Azure Monitor and Application Insights**. This allows real-time tracking of system metrics, error detection, and proactive alerting for system failures.

Objective:

- Monitor the Internee.pk backend for **performance issues** such as CPU usage, memory usage, and response time.
- Track **system reliability** by analyzing failed requests and downtime.
- Identify **errors and security anomalies** using log analysis.
- Configure **alerts** to notify the team of high CPU, failed requests, or system outages.

Project Structure:

internee-cloud-monitoring/

```
├── backend/
│   └── server.js
├── monitoring/
│   ├── metrics.md
│   ├── alerts.md
│   └── kql-queries.md
├── screenshots/
└── README.md
```

GITHUB LINK HERE..

Project Procedure:

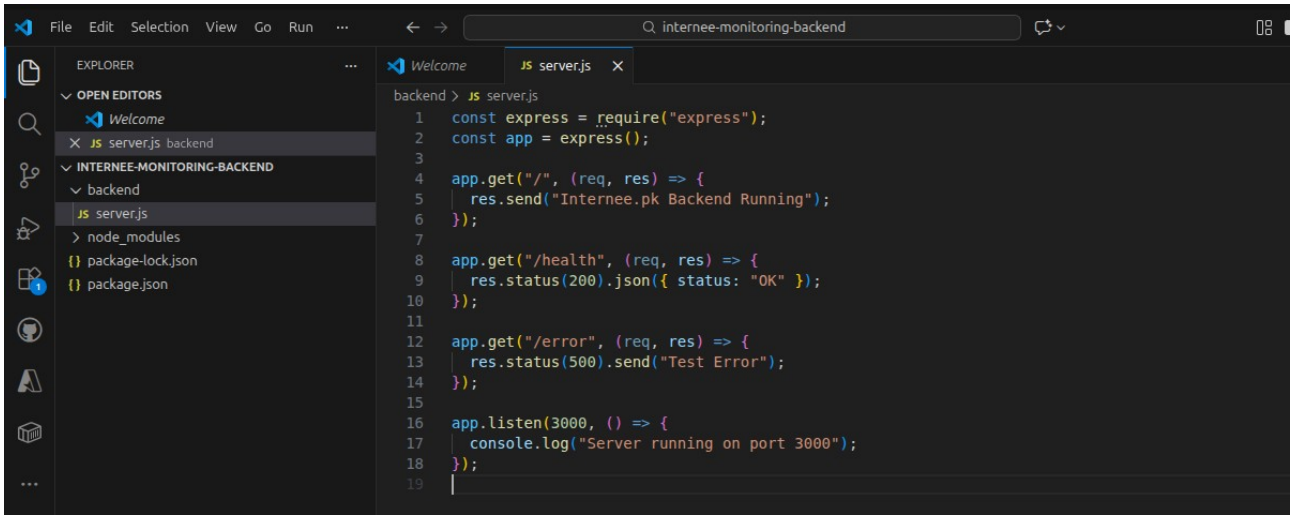
Below is a high-level breakdown of all procedures performed in the project. Each step mentions only actions and file names.

1. Create Backend first :

- Create folder and initialize it.
- Install express that helps in building process

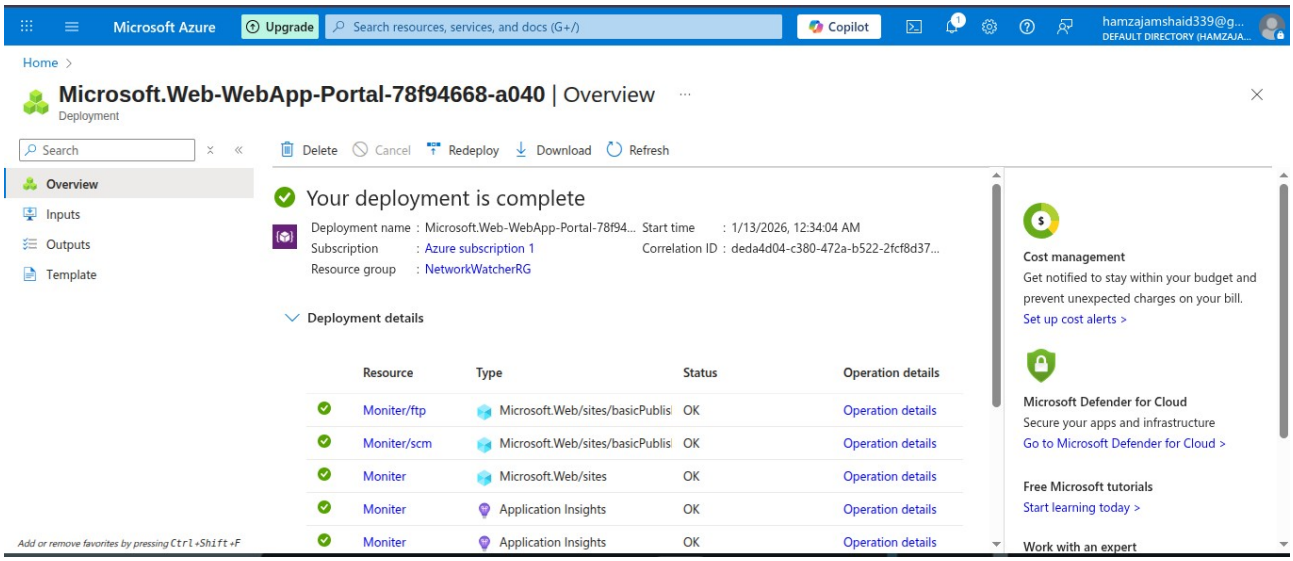
```
hamza@hamza-HP-EliteBook-840-G4:~$ cd internee-monitoring-backend
hamza@hamza-HP-EliteBook-840-G4:~/internee-monitoring-backend$ npm init -y
Wrote to /home/hamza/internee-monitoring-backend/package.json:

{
  "name": "internee-monitoring-backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}



hamza@hamza-HP-EliteBook-840-G4:~/internee-monitoring-backend$ npm install express

added 65 packages, and audited 66 packages in 13s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
hamza@hamza-HP-EliteBook-840-G4:~/internee-monitoring-backend$
```

Create Server.js:



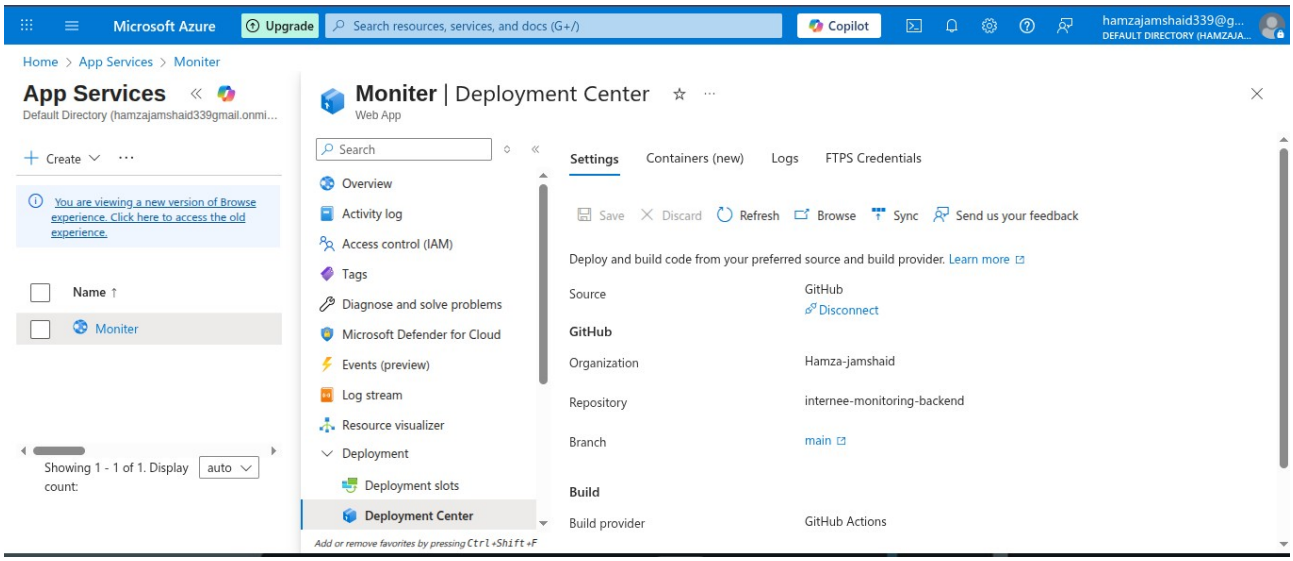
2. Deploy Backend to Azure App Service:

- Create App Services with Runtime “Node 22 LTS”.
- Region “Canada Central” and Name “Moniter”.



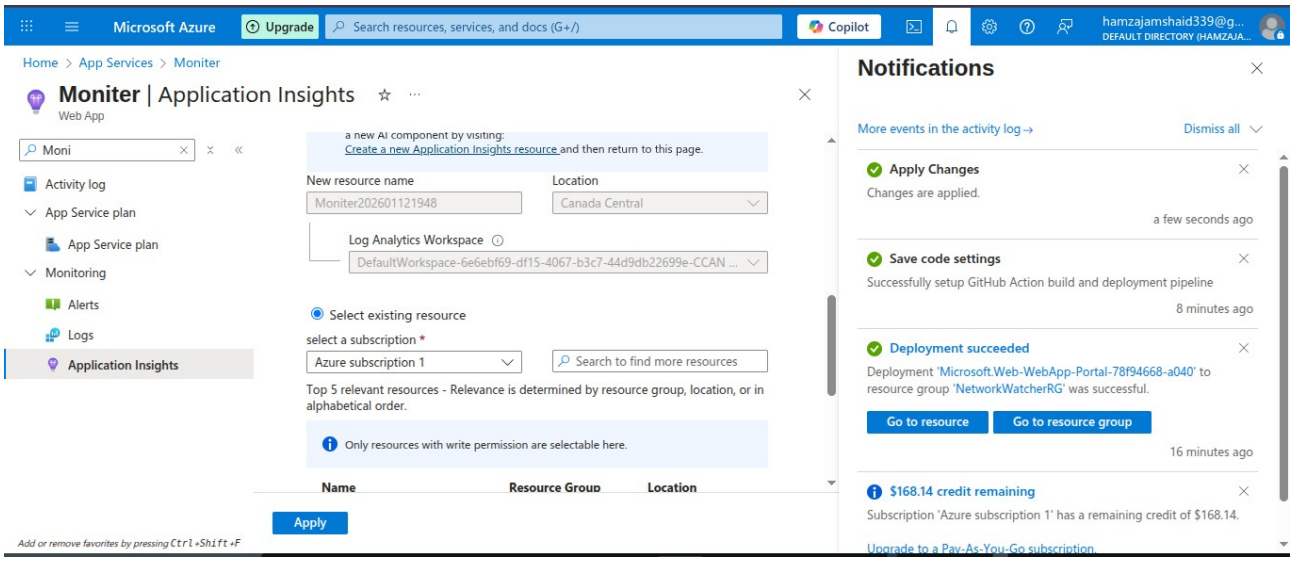
3.Connect GitHub:

- Attach GITHUB to link with Azure and App services.
- Attach related Repo.



4. Enable Azure Monitor:

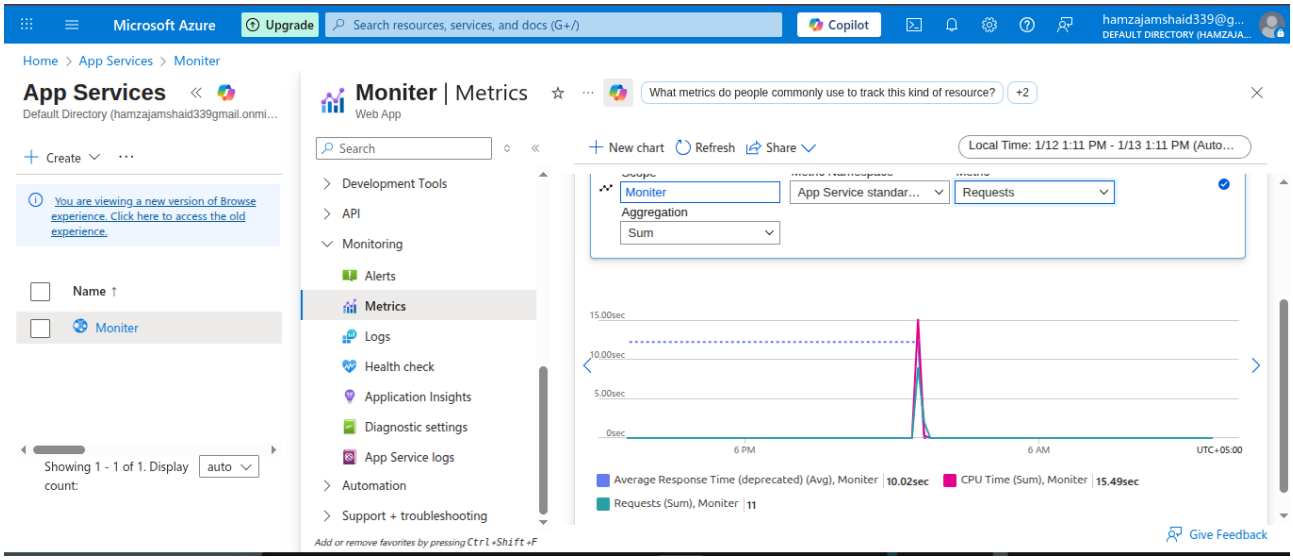
- Turn on Application insight
- Create new log Analytics Workspace



5: Monitor Metrics (CPU, Response Time, Errors):

Add metrics:

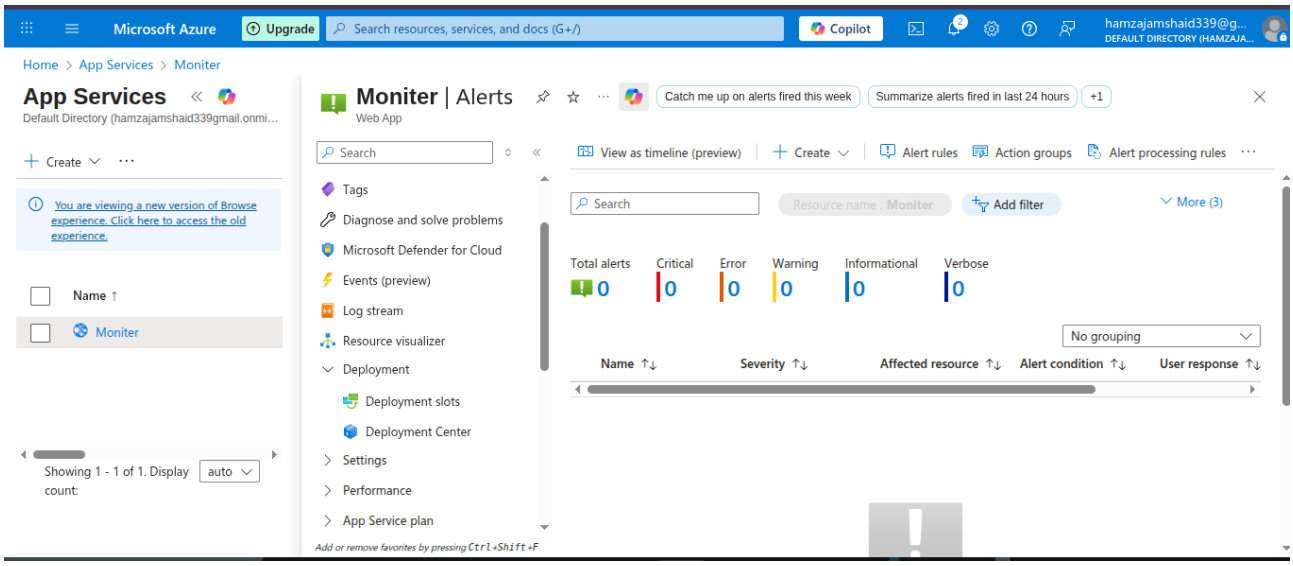
- CPU Percentage
- Requests
- Average Response Time
- Failed Requests



6. Create Alerts (System Failure):

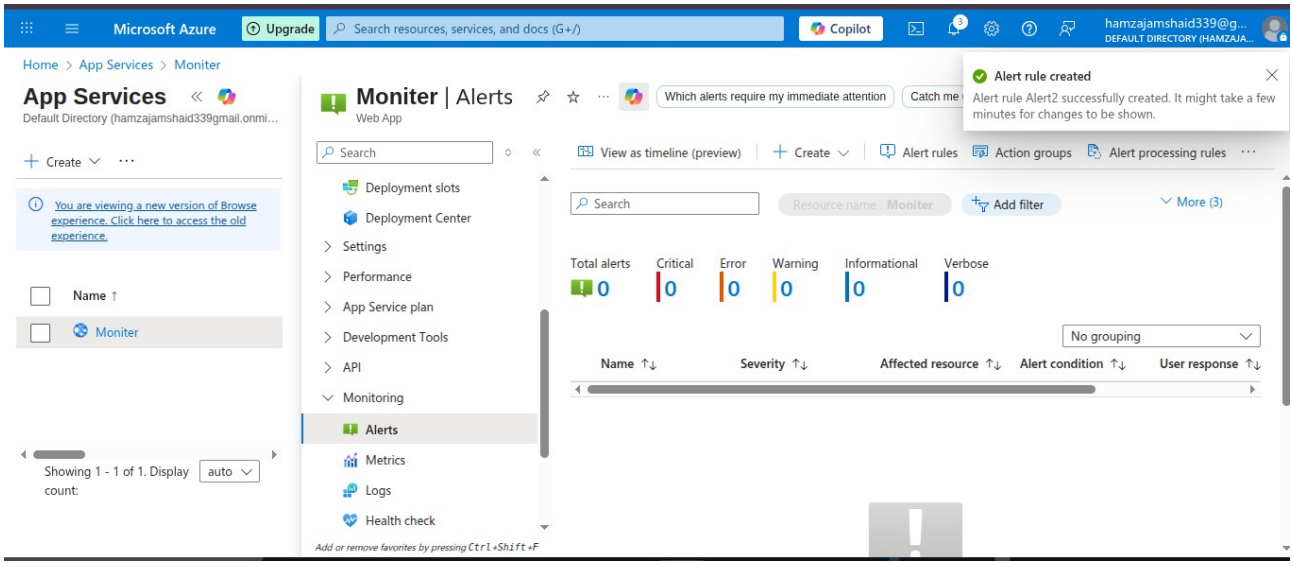
Alert 1: High CPU

- Create Alert Rule
- Metric: CPU Percentage
- Condition: > 80% for 5 minutes
- Action: Email



Alert 2: Errors

- Metric: Failed Requests
- Threshold: > 2 in 5 minutes
- Action: Email



Results:

- Successfully deployed **Node.js backend** on **Azure App Service**.
- Enabled **Azure Monitor and Application Insights**, which tracked:
    - CPU and memory usage
    - Request count and response times
    - Failed requests and 500 errors
- Configured **alerts** for high CPU and failed requests.
- Log queries provided insights into performance bottlenecks and error patterns.
- Generated a professional **GitHub repository** with metrics, alerts, and screenshots for documentation.

Conclusion:

The project demonstrates the importance of **cloud-based monitoring** for web applications. By implementing Azure Monitor and log analysis, the Internee.pk system became more **reliable and observable**, enabling proactive detection of issues. This approach ensures **better performance, faster troubleshooting**, and prepares the system for **scalable production deployment**.