

**Name : Hamza Jamshaid**

**Project title:**

**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"
}

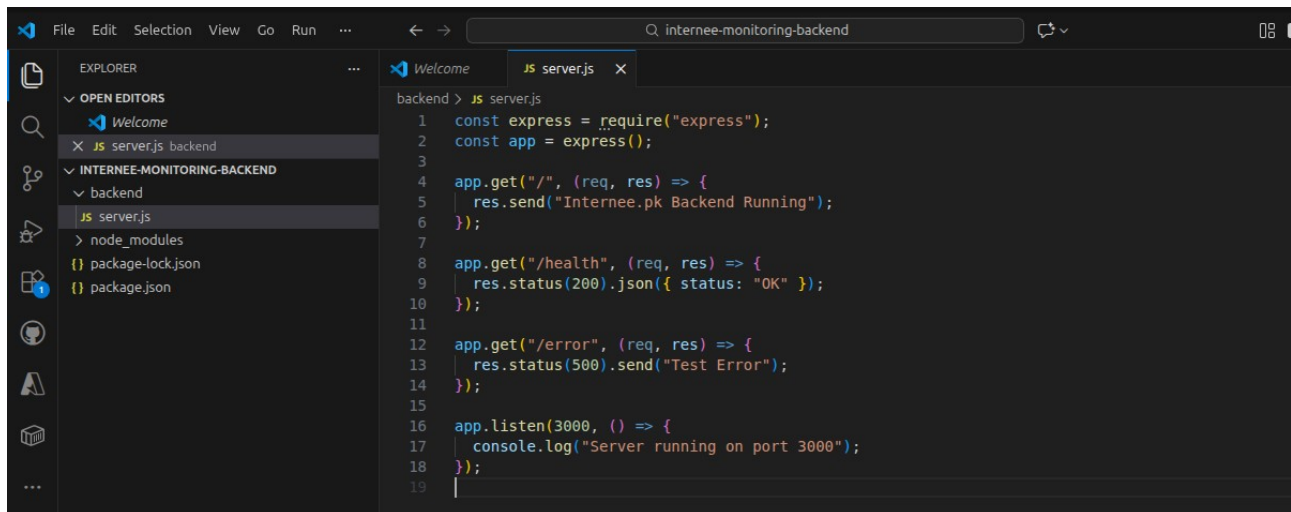
hamza@hamza-HP-EliteBook-840-G4:~/internee-monitoring-backend$ npm install express

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:

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows the project structure: 'INTERNEE-MONITORING-BACKEND' containing a 'backend' folder with 'JS server.js' and 'node\_modules', and a 'package.json' file. The main editor area displays the 'JS server.js' file with the following code:

```
1 const express = require("express");
2 const app = express();
3
4 app.get("/", (req, res) => {
5   res.send("Internee.pk Backend Running");
6 });
7
8 app.get("/health", (req, res) => {
9   res.status(200).json({ status: "OK" });
10 });
11
12 app.get("/error", (req, res) => {
13   res.status(500).send("Test Error");
14 });
15
16 app.listen(3000, () => {
17   console.log("Server running on port 3000");
18 });
19
```

## 2. Deploy Backend to Azure App Service:

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

The screenshot shows the Azure portal interface. The top navigation bar includes the Microsoft Azure logo, an 'Upgrade' button, a search bar, and a Copilot button. The user's profile 'hamzajamshaid339@g...' is visible in the top right. The main content area is titled 'Microsoft.Web-WebApp-Portal-78f94668-a040 | Overview'. A left sidebar contains links for 'Overview', 'Inputs', 'Outputs', and 'Template'. The main area displays a green checkmark and the message 'Your deployment is complete'. Below this, deployment details are listed: Deployment name: Microsoft.Web-WebApp-Portal-78f94..., Start time: 1/13/2026, 12:34:04 AM, Subscription: Azure subscription 1, Correlation ID: deda4d04-c380-472a-b522-2fcf8d37..., and Resource group: NetworkWatcherRG. A table titled 'Deployment details' lists resources and their status:

Resource	Type	Status	Operation details
Monitor/ftp	Microsoft.Web/sites/basicPubli...	OK	<a href="#">Operation details</a>
Monitor/scm	Microsoft.Web/sites/basicPubli...	OK	<a href="#">Operation details</a>
Monitor	Microsoft.Web/sites	OK	<a href="#">Operation details</a>
Monitor	Application Insights	OK	<a href="#">Operation details</a>
Monitor	Application Insights	OK	<a href="#">Operation details</a>

On the right, there are three informational cards: 'Cost management' (Get notified to stay within your budget...), 'Microsoft Defender for Cloud' (Secure your apps and infrastructure...), and 'Free Microsoft tutorials' (Start learning today...). At the bottom, it says 'Work with an expert'.

## 3. Connect GitHub:

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

The screenshot shows the Azure portal interface for the 'Monitor' Web App. The top navigation bar is the same as the previous screenshot. The main content area is titled 'Monitor | Deployment Center'. A left sidebar contains links for 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Microsoft Defender for Cloud', 'Events (preview)', 'Log stream', 'Resource visualizer', 'Deployment', 'Deployment slots', and 'Deployment Center'. The main area shows the 'Settings' tab for the 'Deployment Center'. It includes a search bar, a 'Save' button, and a 'Discard' button. Below, it says 'Deploy and build code from your preferred source and build provider. Learn more'. The 'Source' is set to 'GitHub'. The 'Organization' is 'Hamza-jamshaid'. The 'Repository' is 'internee-monitoring-backend'. The 'Branch' is 'main'. The 'Build' provider is 'GitHub Actions'.

## 4. Enable Azure Monitor:

- Turn on Application insight
- Create new log Analytics Workspace

Microsoft Azure | Upgrade | Search resources, services, and docs (G+)

Home > App Services > Monitor

**Monitor | Application Insights** ☆ ...

Web App

Moni

Activity log

App Service plan

App Service plan

Monitoring

Alerts

Logs

Application Insights

a new AI component by visiting:  
[Create a new Application Insights resource](#) and then return to this page.

New resource name: Monitor202601121948 Location: Canada Central

Log Analytics Workspace: DefaultWorkspace-6e6ebf69-df15-4067-b3c7-44d9db22699e-CCAN ...

Select existing resource

select a subscription \*

Azure subscription 1

Search to find more resources

Top 5 relevant resources - Relevance is determined by resource group, location, or in alphabetical order.

Only resources with write permission are selectable here.

Name	Resource Group	Location
------	----------------	----------

Apply

Notifications

More events in the activity log → Dismiss all

- ✓ Apply Changes  
Changes are applied.  
a few seconds ago
- ✓ Save code settings  
Successfully setup GitHub Action build and deployment pipeline  
8 minutes ago
- ✓ Deployment succeeded  
Deployment 'Microsoft.Web-WebApp-Portal-78f94668-a040' to resource group 'NetworkWatcherRG' was successful.  
Go to resource Go to resource group  
16 minutes ago
- ! \$168.14 credit remaining  
Subscription 'Azure subscription 1' has a remaining credit of \$168.14.  
Upgrade to a Pay-As-You-Go subscription

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

### Add metrics:

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

Microsoft Azure | Upgrade | Search resources, services, and docs (G+)

Home > App Services > Monitor

**App Services** << >>

Default Directory (hamzajamshaid339@gmail.onmi...)

+ Create ...

You are viewing a new version of Browse experience. Click here to access the old experience.

Name ↑

Monitor

Showing 1 - 1 of 1. Display count: auto

**Monitor | Metrics** ☆ ...

Web App

Search

Development Tools

API

Monitoring

Alerts

Metrics

Logs

Health check

Application Insights

Diagnostic settings

App Service logs

Automation

Support + troubleshooting

What metrics do people commonly use to track this kind of resource? +2

New chart Refresh Share

Local Time: 1/12 1:11 PM - 1/13 1:11 PM (Auto...)

Monitor App Service standar... Requests

Aggregation Sum

15.00sec

10.00sec

5.00sec

0sec

6 PM 6 AM UTC+05:00

Average Response Time (deprecated) (Avg), Monitor | 10.02sec CPU Time (Sum), Monitor | 15.49sec

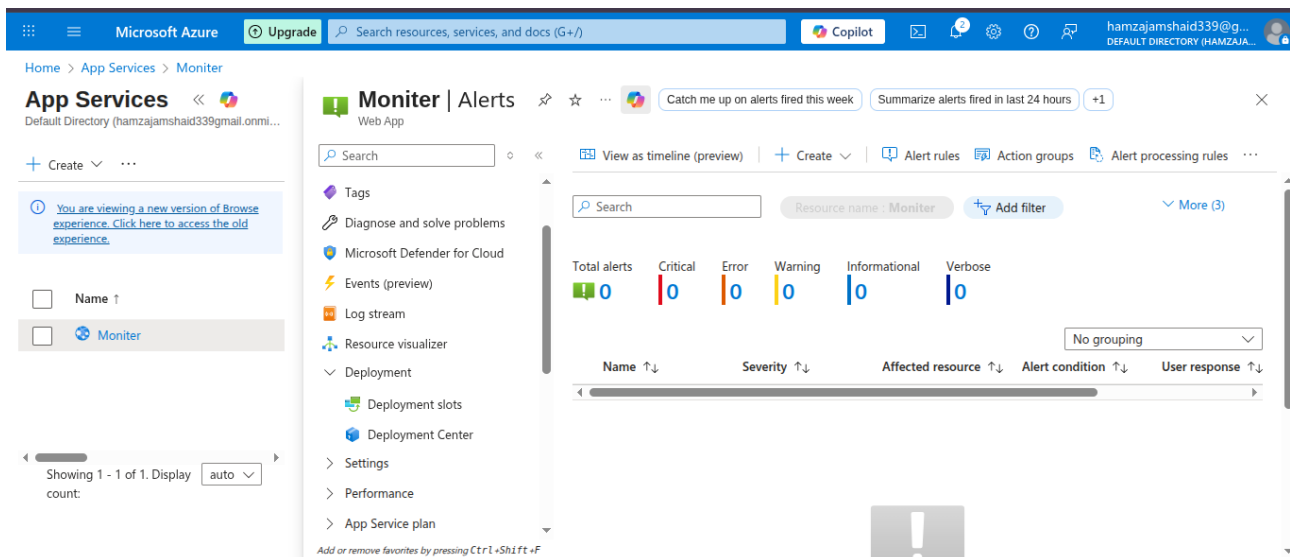
Requests (Sum), Monitor | 11

Give Feedback

## 6. Create Alerts (System Failure):

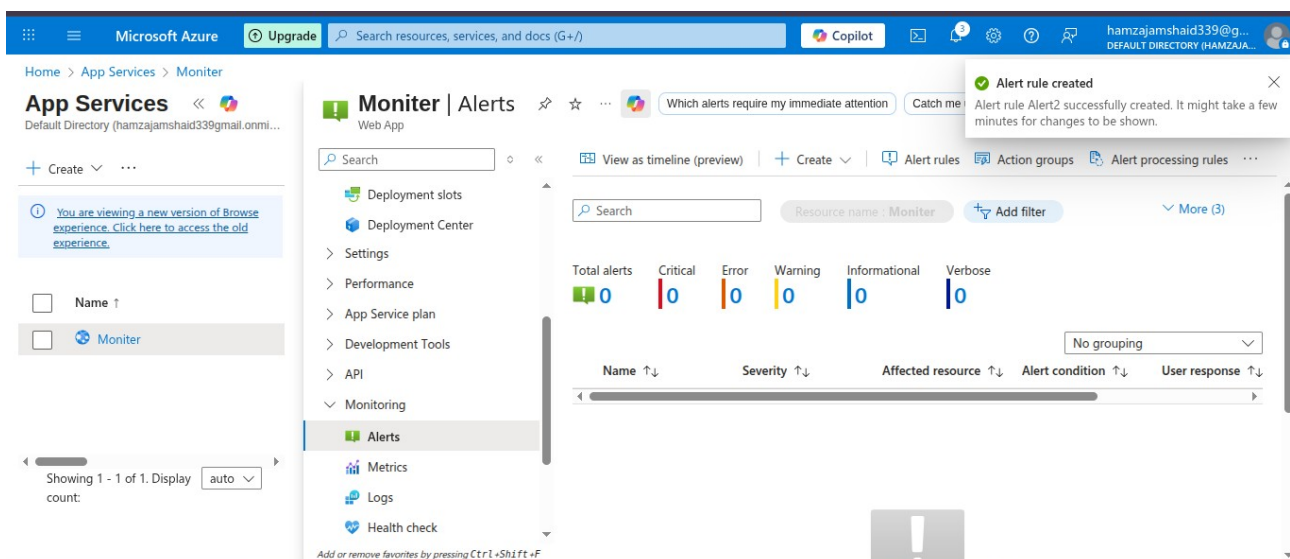
### Alert 1: High CPU

- 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**.