AUTOMATED LAB MONITORING SYSTEM

- - - - - - - - - - - - - -

As a Lab Assistant at Jetking, I was responsible for maintaining 50+ lab PCs, which required time-consuming manual checks for hardware, software, and performance issues. To solve this, I developed an Automated Lab Monitoring System that collects system data on every boot and sends it to AWS for centralized storage and monitoring. A real-time dashboard displays the health of all PCs and generates alerts for critical issues, allowing proactive maintenance. This project not only saved time and effort but also ensured that students always had reliable systems for their learning.

Goal:

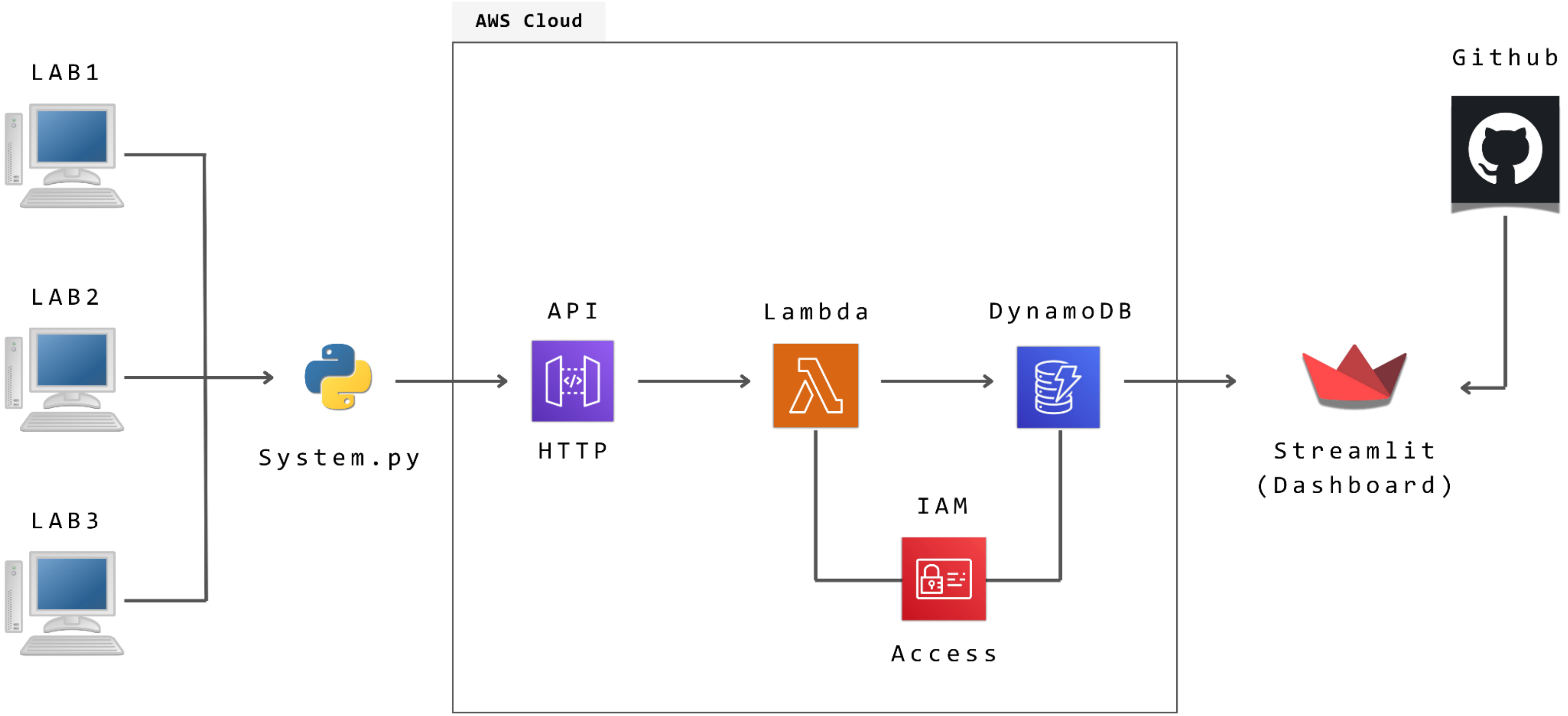
Monitor 50+ lab PCs automatically. Collect hardware, OS, software, and network details. Send this data to AWS and display it on a dashboard.

Core Components:

Client side (system.py) → *Runs on every PC, collects info.* **|** Cloud side (AWS) → *Stores and processes data.*

Streamlit → *Monitoring System* **|** Automation → *Runs automatically on system boot.*

Architecture & Workflow:



Project Setup Guide:

Before you begin, ensure you have:

- AWS Account (Free Tier is enough) - https://aws.amazon.com/console
- Python 3.8+ installed on lab PCs - https://www.python.org/downloads
- GitHub access to clone/download project files - https://github.com/rajgaudev/lab-monitoring-system

1. *system.py*

2. *lambda_function.py*

3. *dashboard.py*

4. *install_monitoring.bat*

AWS Configuration:

Step 1: Create DynamoDB Table

Go to AWS Console → DynamoDB.

- Click Create Table.
- Table Name: LabMonitoring
- Partition Key: device_name (String)
- Leave default settings → Create Table.

Successfully created LabMonitoring table

Step 2: Create Lambda Function

Go to AWS Lambda → Create Function.

- Name: storeSystemInfo
- Runtime: Python 3.9
- Upload your lambda_function.py code.
- Add DynamoDB full access permission to the Lambda IAM Role.

Successfully configured Lamda_function.py

Step 3: Create API Gateway

Go to AWS API Gateway → Create API.

- Select HTTP API.
- Create Resource: /monitor → Method: POST.
- Integration: Select your Lambda Function (storeSystemInfo).
- Deploy API → Note the Invoke URL (example: https://abc123.execute-api.ap-south-1.amazonaws.com/monitor).

Successfully created API

Step 4: Create Lambda function triggers

Go to AWS Lambda → Functions → storeSystemInfo → Configuration

- Select add triggers
- Trigger configuration → API Gateway
- Integration: Select your Lambda Function (storeSystemInfo).
- Deploy API → Note the Invoke URL (example: https://abc123.execute-api.ap-south-1.amazonaws.com/monitor).
- Copy & Save “API_URL” Link
- Preview Function overview, API Gateway linked.

Successfully added triggers

Step 5: IAM Configuration for Dashboard

Go to IAM → Policies → Create Policy

- Select a service: DynamoDB
- Actions allowed: Read(GetItem, Scan) | Write(PutItem, UpdateItem) | List (ListTables)
- Resources: All
- Policy name: LabMonitoring-DynamoDBAccess → Create Policy

Successfully LabMonitoring-DynamoDBAccess policy created

Go to IAM → Users → Create user

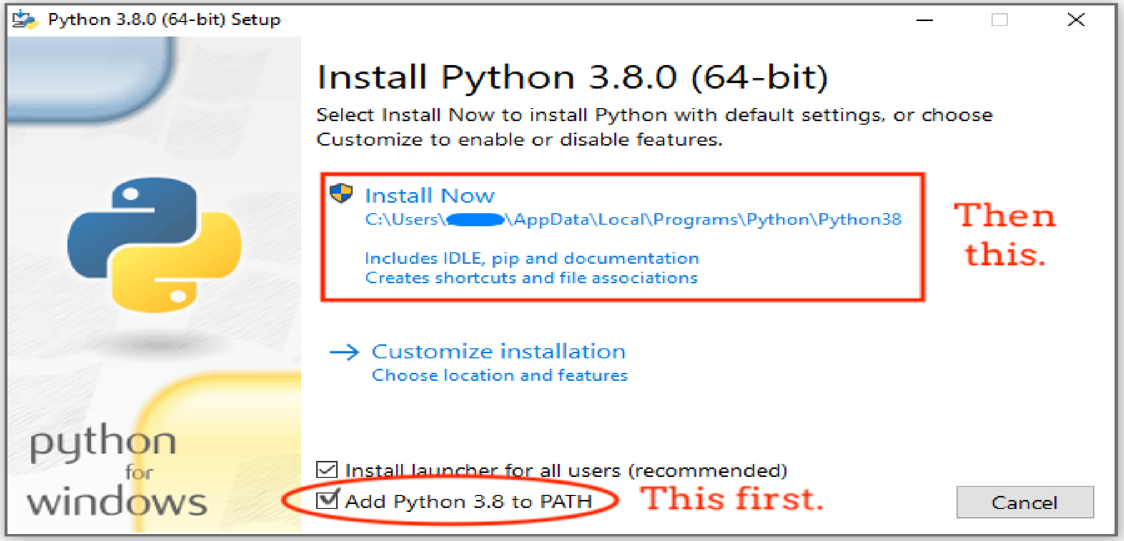
- User name: LabMonitoring
- Select: Attach policies directly
- Search policy: LabMonitoring-DynamoDBAccess
- Review and create
- Access key 1: Create access key
- Use case: Thrid-party service
- Retrieve access key: Copy/Download Access & Secret Key (important)

Successfully LabMonitoring user created

System PC Configuration (After AWS Setup):

Step 1: Install Python

- Ensure Python 3.8+ is installed on the PC.



Python 3.8.0 (64-bit) Setup

Install Python 3.8.0 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\[redacted]\AppData\Local\Programs\Python\Python38

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Then this.

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.8 to PATH

This first.

- Verify in CMD: python --version

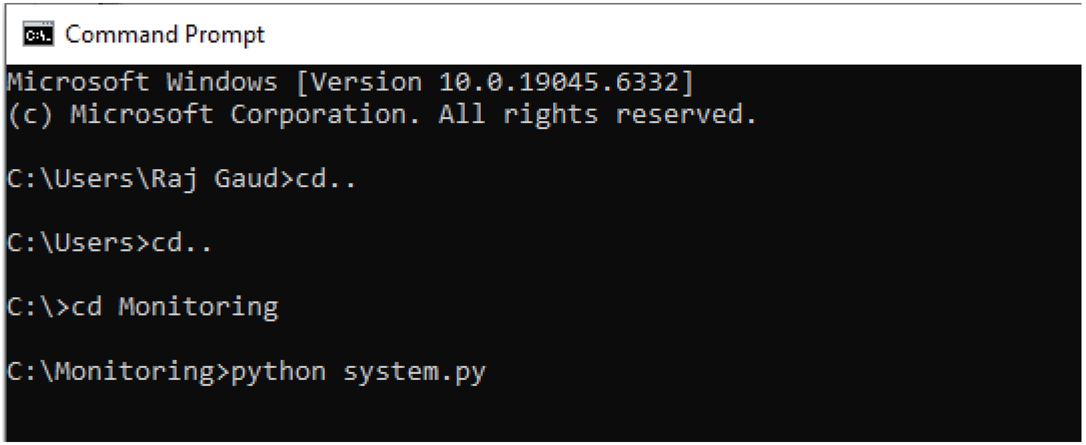
Step 2: Install Script (system.py) & Required Libraries

- Create a new folder named lab.

- Place the following files inside it:
  A. *system.py (the monitoring script)*
  B. *install_monitoring.bat (to auto-install required libraries)*
  C. *edit system.py* → replace "API_URL" → Save it

- Run install_monitoring.bat as Administrator.
  A. *This will automatically install all required Python libraries (psutil, requests, etc.).*

- Test Run: Go to C:\Monitoring>python system.py



```
Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Raj Gaud>cd..

C:\Users>cd..

C:\>cd Monitoring

C:\Monitoring>python system.py
```

- Expected Output: ✓ Data sent to AWS: "Data stored successfully!"

Step 3: Configure Automation (Auto Run on Boot)

- Inside the Monitoring folder, go to the dist subfolder.
- Copy the file system.exe.
- Press Win + R → Type: shell:startup
- Paste system.exe inside the Startup folder.
- ✅ Now, every time the PC cold boots (startup), the script will run automatically.

(Alternative) Task Scheduler

If you prefer Task Scheduler instead of Startup Folder:

- Open Task Scheduler → Create Task.
- General: Name → LabMonitor.
- Trigger → At Startup.
- Action → Run program: C:\Monitoring\dist\system.exe
- Save → Done.

Deploy Dashboard via GitHub + Streamlit Cloud:

––––

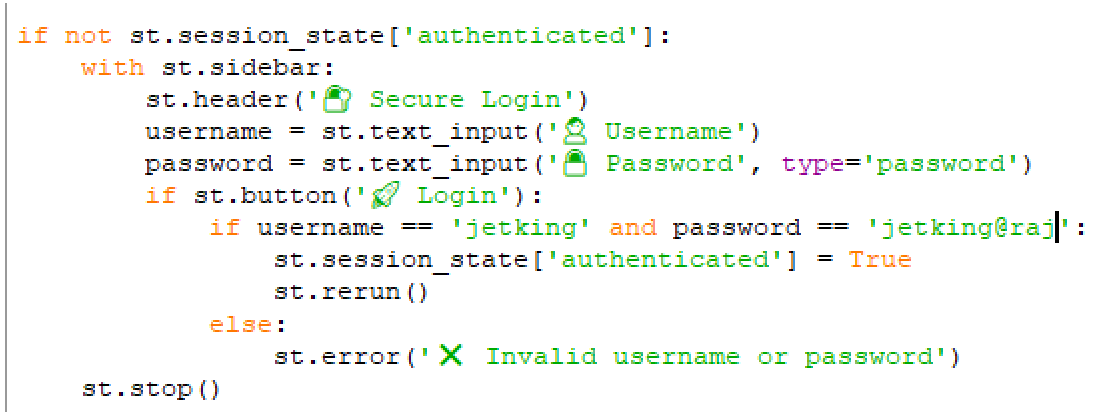
Step 1: Upload Dashboard.py & Requirements.txt to GitHub

- Create a new GitHub repo → Example: lab-monitoring-system
- Upload:

  A. dashboard.py

  B. requirements.txt

- Note: Before Upload edit Username & Password as per your need
- Default: if username == 'jetking' and password == 'jetking@raj':

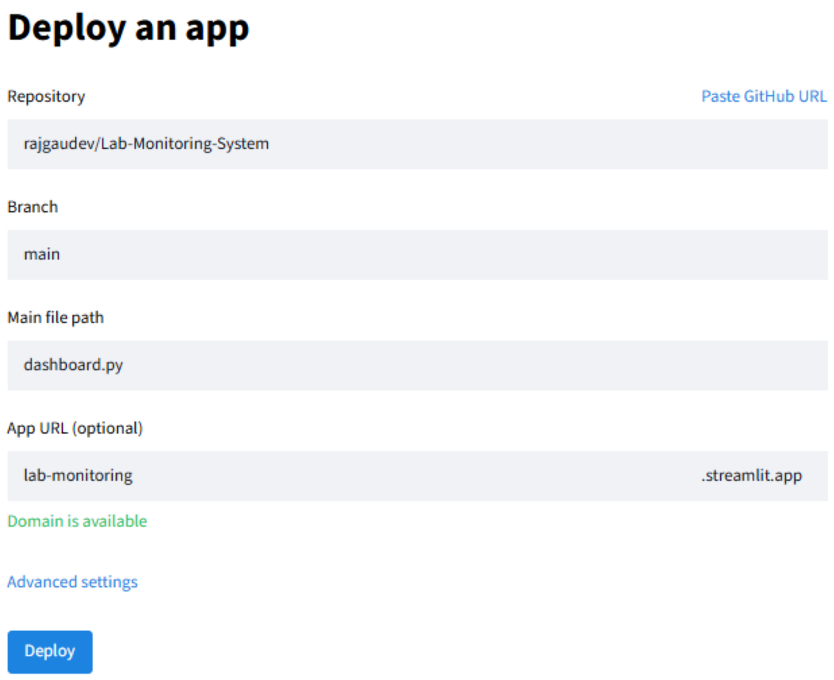


```python
if not st.session_state['authenticated']:
    with st.sidebar:
        st.header('🔐 Secure Login')
        username = st.text_input('👤 Username')
        password = st.text_input('🔒 Password', type='password')
        if st.button('🚀 Login'):
            if username == 'jetking' and password == 'jetking@raj':
                st.session_state['authenticated'] = True
                st.rerun()
            else:
                st.error('❌ Invalid username or password')
    st.stop()
```

Step 2: Deploy on Streamlit Cloud

Go to Streamlit Cloud.

- Log in with GitHub account.
- Click New App.
- Select: Repository → your repo (lab-monitoring-system)
- Branch → main
- File path → dashboard.py
- Click Deploy



Deploy an app

Repository: rajgaudev/Lab-Monitoring-System

Branch: main

Main file path: dashboard.py

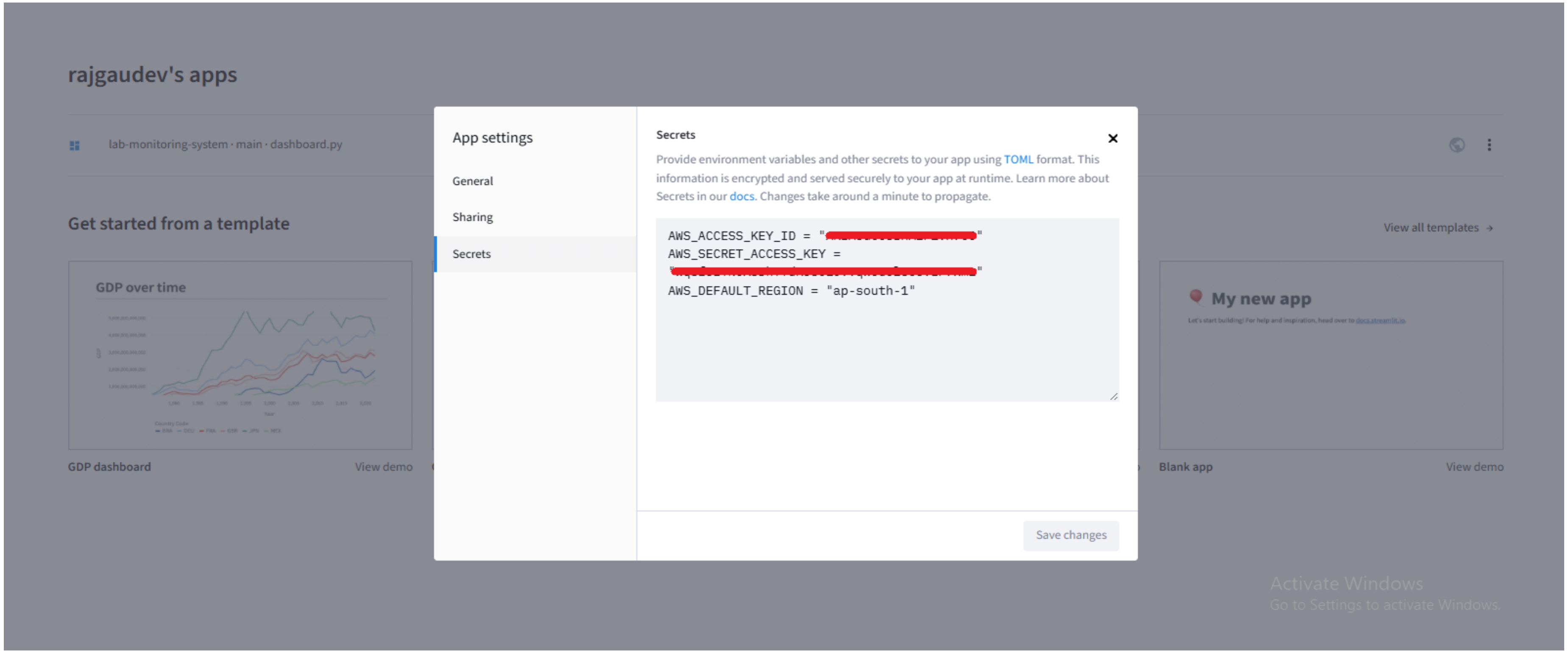
App URL (optional): lab-monitoring .streamlit.app

Domain is available

Step 3: AWS Credentials for Streamlit

Since your dashboard reads data from DynamoDB via boto3, Streamlit Cloud needs AWS credentials.

- In Streamlit Cloud → Go to App Settings → Secrets.
- Add:
    A. AWS_ACCESS_KEY_ID = "your_access_key"
    B. AWS_SECRET_ACCESS_KEY = "your_secret_key"
    C. AWS_REGION = "ap-south-1"



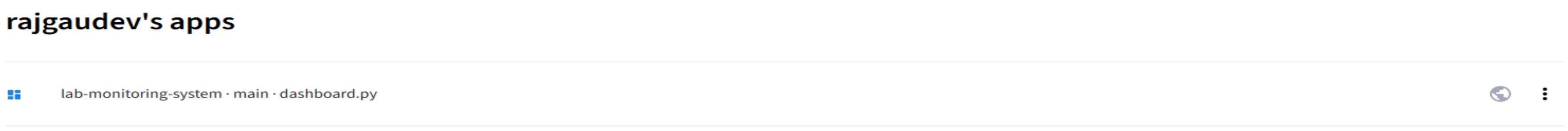
Secrets

Provide environment variables and other secrets to your app using TOML format. This information is encrypted and served securely to your app at runtime. Learn more about Secrets in our docs. Changes take around a minute to propagate.

```
AWS_ACCESS_KEY_ID = "[redacted]"
AWS_SECRET_ACCESS_KEY =
"[redacted]"
AWS_DEFAULT_REGION = "ap-south-1"
```

Step 4: Verify Streamlit Dashboard

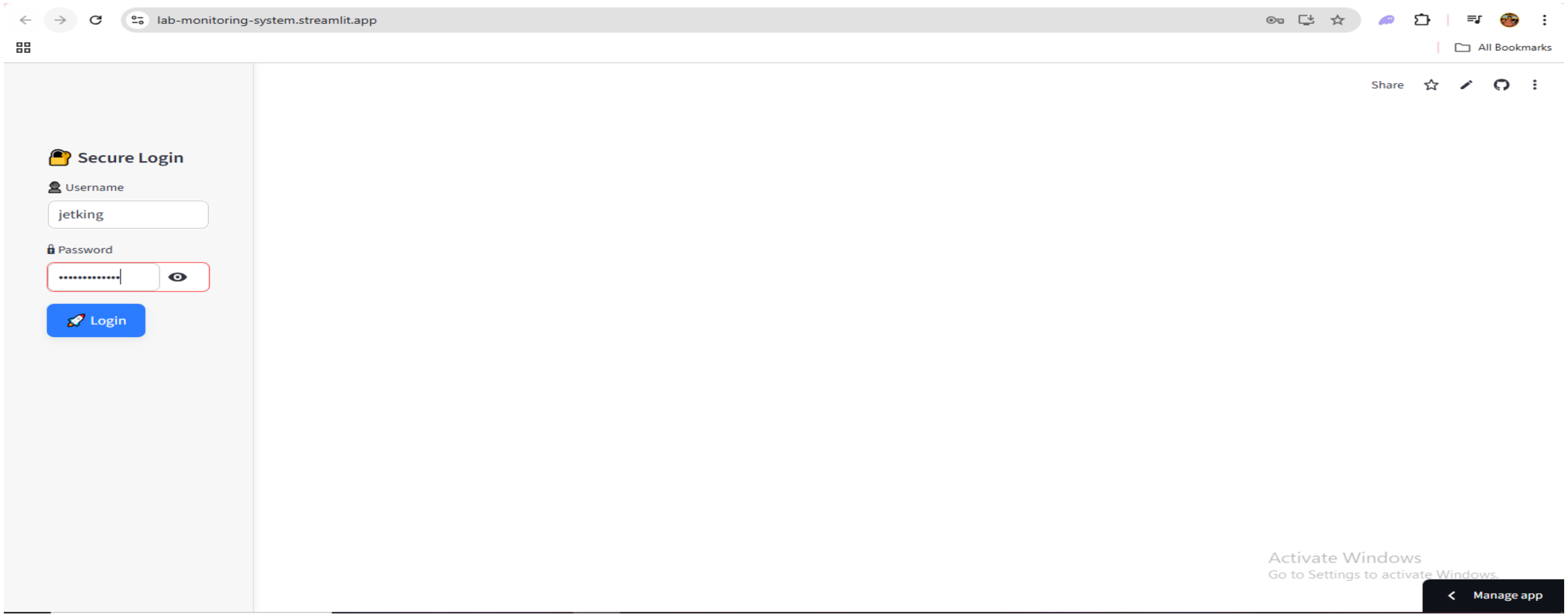
- Click on APP Link



rajgaudev's apps

lab-monitoring-system · main · dashboard.py

- Enter Username & Password



lab-monitoring-system.streamlit.app

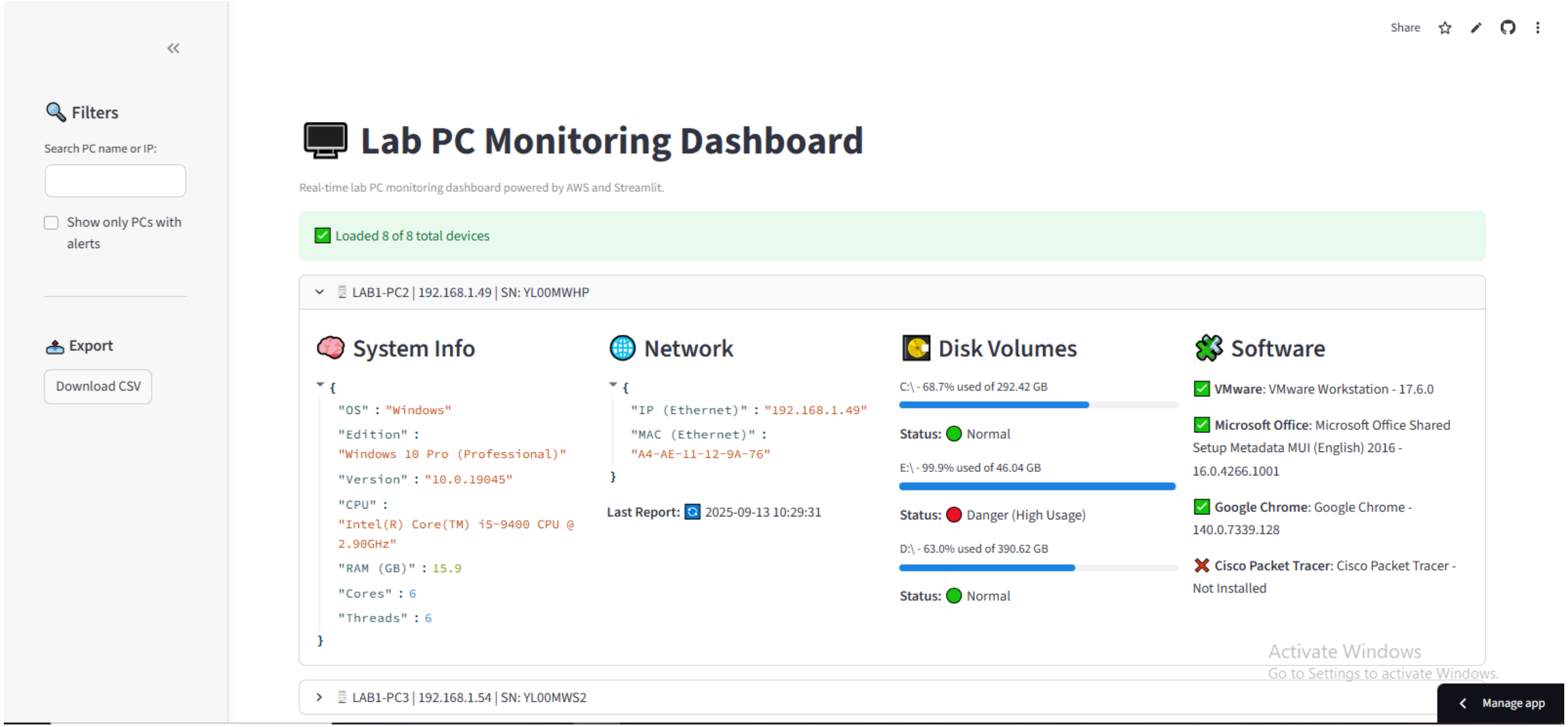
Secure Login

Username

jetking

Password

·············



🔍 Filters

Search PC name or IP:

□ Show only PCs with alerts

Export

Download CSV

Lab PC Monitoring Dashboard

Real-time lab PC monitoring dashboard powered by AWS and Streamlit.

✅ Loaded 8 of 8 total devices

⌄ LAB1-PC2 | 192.168.1.49 | SN: YL00MWHP

System Info

```
{
  "OS" : "Windows"
  "Edition" :
  "Windows 10 Pro (Professional)"
  "Version" : "10.0.19045"
  "CPU" :
  "Intel(R) Core(TM) i5-9400 CPU @
  2.90GHz"
  "RAM (GB)" : 15.9
  "Cores" : 6
  "Threads" : 6
}
```

Network

```
{
  "IP (Ethernet)" : "192.168.1.49"
  "MAC (Ethernet)" :
  "A4-AE-11-12-9A-76"
}
```

Last Report: 🔄 2025-09-13 10:29:31

Disk Volumes

C:\ - 68.7% used of 292.42 GB

Status: 🟢 Normal

E:\ - 99.9% used of 46.04 GB

Status: 🔴 Danger (High Usage)

D:\ - 63.0% used of 390.62 GB

Status: 🟢 Normal

Software

✅ **VMware:** VMware Workstation - 17.6.0

✅ **Microsoft Office:** Microsoft Office Shared Setup Metadata MUI (English) 2016 - 16.0.4266.1001

✅ **Google Chrome:** Google Chrome - 140.0.7339.128

❌ **Cisco Packet Tracer:** Cisco Packet Tracer - Not Installed

Activate Windows
Go to Settings to activate Windows.

› LAB1-PC3 | 192.168.1.54 | SN: YL00MWS2

Successfully Lab Monitoring Dashboard Deployed

Dashboard: lab-monitoring-system.streamlit.app

Linkedin: linkedin.com/in/rajgaud