Devops practical 9

Got it — let's rewind and treat this as if you're starting from **zero setup**. I'll give you:

1. **Prerequisites (what you need installed first)**
2. **Possible platforms you can work within**
3. **Step-by-step setup flow (from scratch)**

---

1. Prerequisites

- **Hardware/OS**:
  - Works on **Windows, Linux, macOS** (including your **MacBook Air M3**).
- **Software needed**:
  0. **Docker Desktop** → Easiest way to run Prometheus + Grafana without installing them manually.
     - Download Docker Desktop and install.
     - Verify install:
     - docker --version
     - docker compose version
  1. **Python 3.10+** (already pre-installed on macOS; check with python3 --version).
  2. **Text editor/IDE** → VS Code (recommended).
  3. **Git** (optional, for cloning examples).

🔹 2. Platforms you can choose

You have **three good options** depending on how you want to run things:

A. Local with Docker (recommended for you)

- Run **Prometheus & Grafana** in Docker containers.
- Run your **Python app** locally or in Docker.
  ✅ Simple, portable, works cross-platform.

B. Cloud Playground (no installs)

- Use **GitHub Codespaces** or **Gitpod** (online dev environments).
- Spin up containers in-browser.
  ✅ No local setup, but limited free resources.

C. Kubernetes (advanced)

- Run Prometheus & Grafana inside **Kubernetes** with Helm charts.
  ✅ Closer to production setups.
  ❌ Overkill if you're just learning.

👉 Since you said you have *nothing installed yet*, I suggest **Option A: Local with Docker**. That's the most straightforward.

🔹 3. Step-by-Step Setup

Here's the complete flow for **Option A (Local with Docker)**:

Step 1: Install Docker Desktop

- Download & install from docker.com.
- After install → open terminal and check:
- docker run hello-world

✅ If you see "Hello from Docker!", you're good.

Step 2: Install Python + pip

- On macOS:
- brew install python

or just check if already installed:

python3 --version

pip3 --version

Step 3: Create your project folder (in terminal)

mkdir monitoring-demo

```
cd monitoring-demo
```

(on terminal)

```
pip install prometheus-flask-exporter
```

Step 4: Create your Python app (in vscode)

Inside monitoring-demo/app.py:

```
from flask import Flask, jsonify
from prometheus_flask_exporter import PrometheusMetrics

app = Flask(__name__)
metrics = PrometheusMetrics(app, path="/metrics")

@app.route("/hello")
def hello():
   return jsonify(message="Hello from monitored app!")

if __name__ == "__main__":
   app.run(host="0.0.0.0", port=5000)
```

Dependencies in requirements.txt:

```
Flask
prometheus-flask-exporter
```

Step 5: Add Docker + Prometheus + Grafana configs (in vscode)

Dockerfile (for your Python app):

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
```

```
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]
```

prometheus.yml:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "python-app"
    metrics_path: /metrics
    static_configs:
      - targets: ["python-app:5000"]
```

docker-compose.yml:

```
services:
  python-app:
    build: .
    ports:
      - "5000:5000"

  prometheus:
    image: prom/prometheus:latest
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
    ports:
      - "9090:9090"

  grafana:
    image: grafana/grafana-oss:latest
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=admin
    ports:
      - "3000:3000"
```

Step 6: Run everything (in terminal)

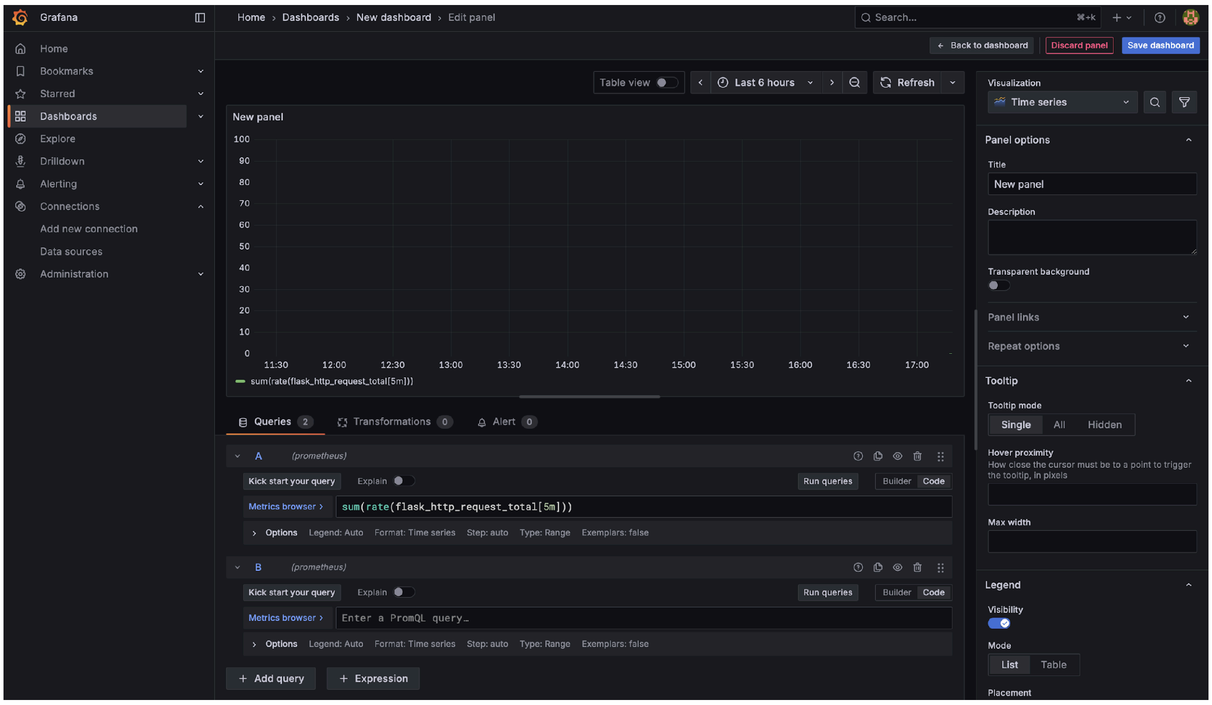
```
docker compose up -d --build
```

(open on browser)

- Python app → http://localhost:5000/hello
- Metrics → http://localhost:5000/metrics
- Prometheus UI → http://localhost:9090
- Grafana → http://localhost:3000 (login: admin/admin)

Step 7: Configure Grafana (in browser)

1. Go to Grafana → Add **Prometheus** as data source (http://prometheus:9090). (in add connections in left heirarchy)
2. Create a new dashboard.
3. Add panels for:
   - **API request rate**
4. sum(rate(flask_http_request_total[5m]))
   - **API latency (p95)**
5. histogram_quantile(0.95, sum(rate(flask_http_request_duration_seconds_bucket[5m])) by (le))
   - **CPU usage (process)**
6. rate(process_cpu_seconds_total[1m])
   - **Memory (RSS MB)**
7. process_resident_memory_bytes / 1024 / 1024



Query A (prometheus):

```
sum(rate(flask_http_request_total[5m]))
```

Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

Query B (prometheus): Enter a PromQL query…

Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

Title: New panel
Transparent background: off
Tooltip mode: Single
Legend Visibility: on
Legend Mode: List

✅ At this point you'll have:

- Prometheus scraping metrics from your Python app.
- Grafana dashboards showing CPU, memory, request rate, and latency.

Do you want me to **prepare a one-shot starter repo** (zip or GitHub-ready) with all files (app + Docker + configs), so you just clone & run without touching configs?