



Team DevOn



K8S CLUSTER MONITORING

STGI Hackathon

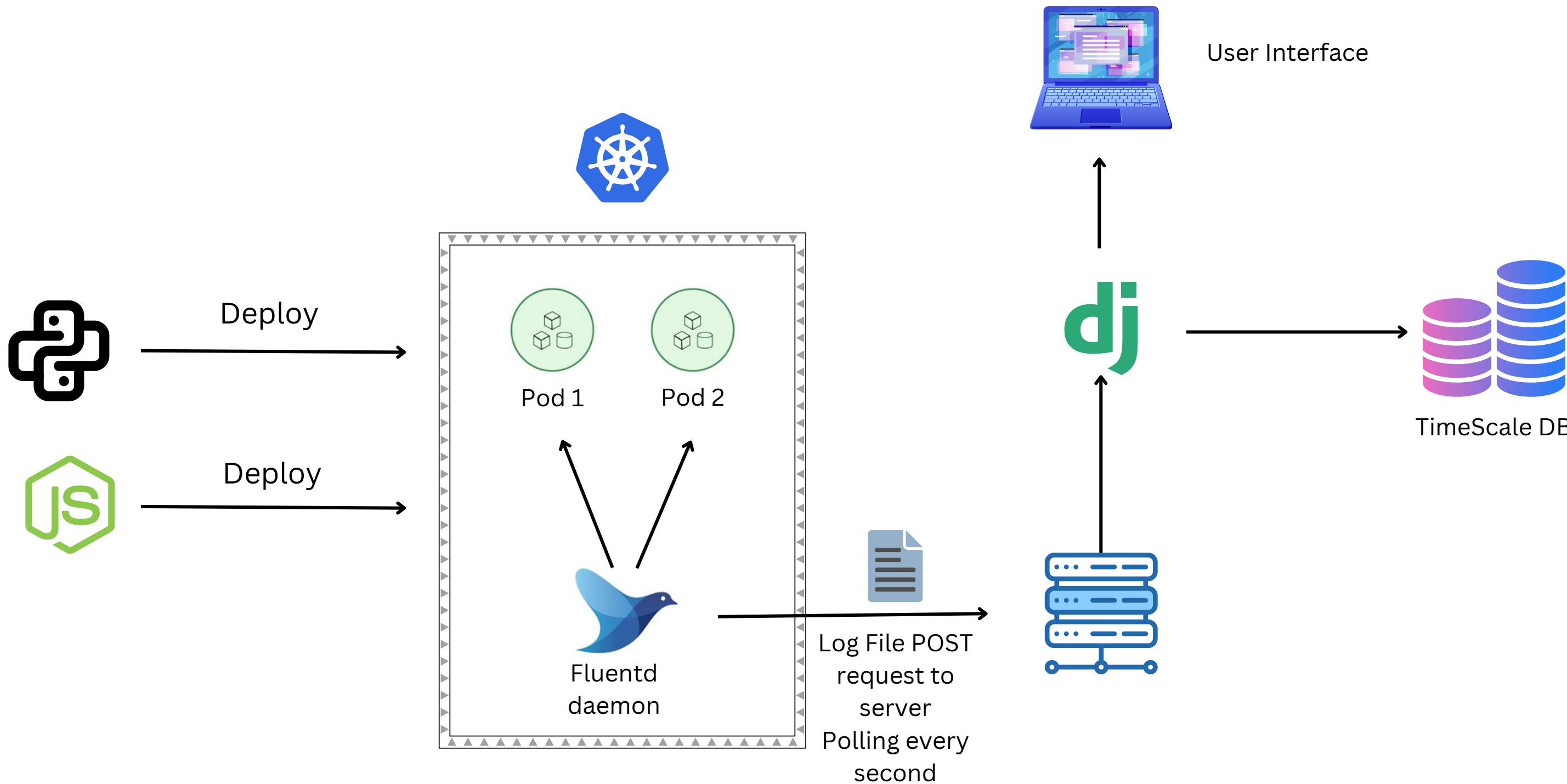


Problem Statement

InfraMonitor

You will set up a Kubernetes cluster (using MiniKube) with two applications, a PostgreSQL and a MySQL database, running on your local machine. Your task is to build a monitoring solution that collects logs from the applications running in the Minikube cluster, authenticates them, stores them in a time series database, and visualizes them in a custom dashboard.

Architecture



Solution

Step 1

deployment of app1 and app2 in
Kubernetes

Step 2

capturing the logs using FluentD

Step 3

Inserting the logs in a time-series database via rest API

Step 4

Fetching the data via Django backend and visualizing on the dashboard in the form of graphs and tables, sorting and filtering the data

LOGS

```
2024-09-29 02:41:16 WARNING [django.server] Server started handling POST request to /api/v1/login. Status 403.
2024-09-29 02:41:18 INFO [django.db.backends] Query executed on /api/v1/cart. Response: 503 in 1.201s.
nkgupta@Nalins-MacBook-Pro fluentd % kubectl logs logger-hackathon-598c79f695-qc4sf
2024-09-29 02:40:52 ERROR [django.request] PUT /api/v1/users - Status: 400 - Response Time: 2.903s - IP: 198.51.100.12
2024-09-29 02:40:53 INFO [django.request] POST /api/v1/cart - Status: 400 - Response Time: 1.826s - IP: 198.51.100.12
2024-09-29 02:40:55 DEBUG [django.db.backends] Query executed on /api/v1/login. Response: 503 in 2.951s.
2024-09-29 02:40:56 DEBUG [django.security] Potential security issue detected on /api/v1/cart from 10.0.0.8. Status 200.
2024-09-29 02:40:57 DEBUG [django.db.backends] Query executed on /api/v1/logout. Response: 400 in 2.492s.
2024-09-29 02:40:58 INFO [django.security] Potential security issue detected on /api/v1/login from 172.16.0.23. Status 403.
2024-09-29 02:41:00 WARNING [django.server] Server started handling PUT request to /api/v1/orders. Status 200.
2024-09-29 02:41:02 ERROR [django] Processing POST request to /api/v1/cart. Status 404.
2024-09-29 02:41:02 ERROR [django.db.backends] Query executed on /api/v1/users. Response: 201 in 2.02s.
2024-09-29 02:41:04 WARNING [django.request] PATCH /api/v1/notifications - Status: 503 - Response Time: 0.935s - IP: 192.168.1.10
2024-09-29 02:41:05 ERROR [django] Processing POST request to /api/v1/cart. Status 400.
2024-09-29 02:41:07 ERROR [django] Processing POST request to /api/v1/cart. Status 201.
2024-09-29 02:41:09 ERROR [django.server] Server started handling GET request to /api/v1/products. Status 503.
2024-09-29 02:41:09 INFO [django.server] Server started handling PUT request to /api/v1/products. Status 503.
2024-09-29 02:41:11 ERROR [django.request] POST /api/v1/products - Status: 403 - Response Time: 0.633s - IP: 172.16.0.23
2024-09-29 02:41:11 WARNING [django] Processing PUT request to /api/v1/logout. Status 401.
2024-09-29 02:41:13 WARNING [django.security] Potential security issue detected on /api/v1/cart from 172.16.0.23. Status 200.
2024-09-29 02:41:14 INFO [django.security] Potential security issue detected on /api/v1/products from 172.16.0.23. Status 400.
2024-09-29 02:41:16 DEBUG [django.db.backends] Query executed on /api/v1/cart. Response: 503 in 0.514s.
2024-09-29 02:41:16 WARNING [django.server] Server started handling POST request to /api/v1/login. Status 403.
2024-09-29 02:41:18 INFO [django.db.backends] Query executed on /api/v1/cart. Response: 503 in 1.201s.
2024-09-29 02:41:19 ERROR [django.db.backends] Query executed on /api/v1/products. Response: 503 in 2.168s.
nkgupta@Nalins-MacBook-Pro fluentd % kubectl logs logger-hackathon-598c79f695-qc4sf
2024-09-29 02:40:52 ERROR [django.request] PUT /api/v1/users - Status: 400 - Response Time: 2.903s - IP: 198.51.100.12
2024-09-29 02:40:53 INFO [django.request] POST /api/v1/cart - Status: 400 - Response Time: 1.826s - IP: 198.51.100.12
2024-09-29 02:40:55 DEBUG [django.db.backends] Query executed on /api/v1/login. Response: 503 in 2.951s.
2024-09-29 02:40:56 DEBUG [django.security] Potential security issue detected on /api/v1/cart from 10.0.0.8. Status 200.
2024-09-29 02:40:57 DEBUG [django.db.backends] Query executed on /api/v1/logout. Response: 400 in 2.492s.
2024-09-29 02:40:58 INFO [django.security] Potential security issue detected on /api/v1/login from 172.16.0.23. Status 403.
2024-09-29 02:41:00 WARNING [django.server] Server started handling PUT request to /api/v1/orders. Status 200.
2024-09-29 02:41:02 ERROR [django] Processing POST request to /api/v1/cart. Status 404.
2024-09-29 02:41:02 ERROR [django.db.backends] Query executed on /api/v1/users. Response: 201 in 2.02s.
2024-09-29 02:41:04 WARNING [django.request] PATCH /api/v1/notifications - Status: 503 - Response Time: 0.935s - IP: 192.168.1.10
2024-09-29 02:41:05 ERROR [django] Processing POST request to /api/v1/cart. Status 400.
2024-09-29 02:41:07 ERROR [django] Processing POST request to /api/v1/cart. Status 201.
2024-09-29 02:41:09 ERROR [django.server] Server started handling GET request to /api/v1/products. Status 503.
2024-09-29 02:41:09 INFO [django.server] Server started handling PUT request to /api/v1/products. Status 503.
2024-09-29 02:41:11 ERROR [django.request] POST /api/v1/products - Status: 403 - Response Time: 0.633s - IP: 172.16.0.23
2024-09-29 02:41:11 WARNING [django] Processing PUT request to /api/v1/logout. Status 401.
2024-09-29 02:41:13 WARNING [django.security] Potential security issue detected on /api/v1/cart from 172.16.0.23. Status 200.
2024-09-29 02:41:14 INFO [django.security] Potential security issue detected on /api/v1/products from 172.16.0.23. Status 400.
2024-09-29 02:41:16 DEBUG [django.db.backends] Query executed on /api/v1/cart. Response: 503 in 0.514s.
2024-09-29 02:41:16 WARNING [django.server] Server started handling POST request to /api/v1/login. Status 403.
2024-09-29 02:41:18 INFO [django.db.backends] Query executed on /api/v1/cart. Response: 503 in 1.201s.
2024-09-29 02:41:19 ERROR [django.db.backends] Query executed on /api/v1/products. Response: 503 in 2.168s.
2024-09-29 02:41:20 ERROR [django.security] Potential security issue detected on /api/v1/reports from 10.0.0.8. Status 401.
nkgupta@Nalins-MacBook-Pro fluentd % kubectl logs logger-hackathon-598c79f695-qc4sf
2024-09-29 02:40:52 ERROR [django.request] PUT /api/v1/users - Status: 400 - Response Time: 2.903s - IP: 198.51.100.12
2024-09-29 02:40:53 INFO [django.request] POST /api/v1/cart - Status: 400 - Response Time: 1.826s - IP: 198.51.100.12
2024-09-29 02:40:55 DEBUG [django.db.backends] Query executed on /api/v1/login. Response: 503 in 2.951s.
2024-09-29 02:40:56 DEBUG [django.security] Potential security issue detected on /api/v1/cart from 10.0.0.8. Status 200.
2024-09-29 02:40:57 DEBUG [django.db.backends] Query executed on /api/v1/logout. Response: 400 in 2.492s.
2024-09-29 02:40:58 INFO [django.security] Potential security issue detected on /api/v1/login from 172.16.0.23. Status 403.
2024-09-29 02:41:00 WARNING [django.server] Server started handling PUT request to /api/v1/orders. Status 200.
2024-09-29 02:41:02 ERROR [django] Processing POST request to /api/v1/cart. Status 404.
2024-09-29 02:41:02 ERROR [django.db.backends] Query executed on /api/v1/users. Response: 201 in 2.02s.
2024-09-29 02:41:04 WARNING [django.request] PATCH /api/v1/notifications - Status: 503 - Response Time: 0.935s - IP: 192.168.1.10
```

Sending logs securely

```
3 metadata:  
4   name: fluentd-config  
5   namespace: default  
6 data:  
7   fluent.conf: |  
8     <source>  
9       @type tail  
0       path /var/log/pods/default_logger-hackathon-*/*/*.log  
1       pos_file /var/log/fluentd-containers.log.pos  
2       tag kubernetes.*  
3     <parse>  
4       @type "json"  
5       time_key "time"  
6       time_format "%Y-%m-%dT%H:%M:%S.%NZ"  
7     </parse>  
8   </source>  
9  
0   <match kubernetes.**>  
1     @type http  
2     endpoint "http://drf-backend-web:8000/api/capture/" # Replace with your Django API endpoint  
3     http_method post  
4     serializer json  
5     open_timeout 5  
6     read_timeout 5  
7     flush_interval 1s # Adjust as needed  
8   </match>
```

Dashboard

STGI | InfraMonitor

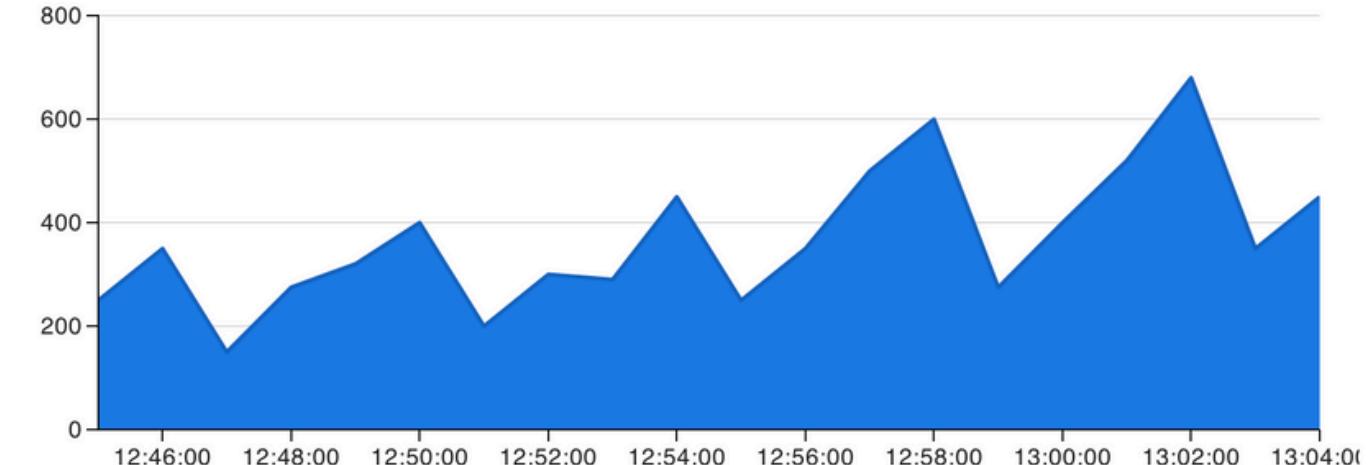
Large Text Theme  

Dashboard >  Secondary App
App 2  SEP 29, 2024

Past 24 hours Past Week Past Month

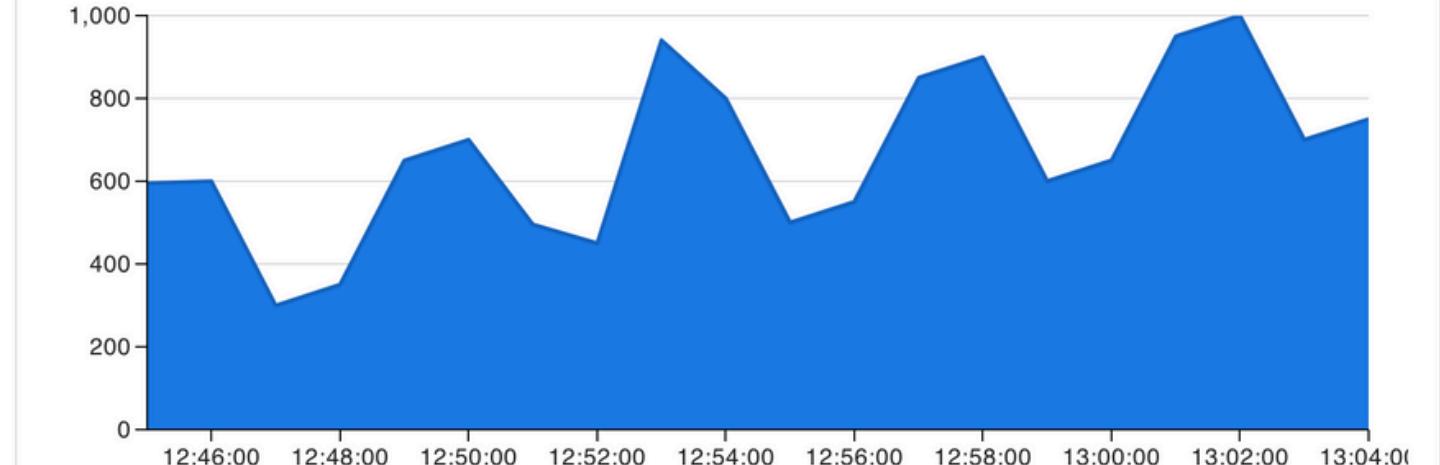
CPU Usage

CPU Usage of last 20 Minutes (MilliCores per minute)



Memory Usage

Memory Usage of last 20 Minutes (MB per minute)



Logs

Time Stamp ↓	Method	Endpoint	Status Code	Message
2024-09-29 17:33:42.628603260	PUT	/api/v1/orders	500	DEBUG
2024-09-28 17:56:30.392104390	DELETE	/api/v1/products	500	ERROR
2024-09-28 17:55:45.645839201	POST	/api/v1/users	201	INFO
2024-09-28 17:54:59.423715612	PATCH	/api/v1/customers/78	200	SUCCESS
2024-09-28 17:53:10.512874932	GET	/api/v1/products	304	DEBUG

Dashboard UI Features

Themes -
dark and light modes
Large Text Theme
Small Text Themes

Time Stamp	Method	Endpoint	Status Code
2024-09-29 17:33:42.628603260	PUT	/api/v1/orders	500
2024-09-28 17:56:30.392104390	DELETE	/api/v1/products	500
2024-09-28 17:55:45.645839201	POST	/api/v1/users	201
2024-09-28 17:54:59.423715612	PATCH	/api/v1/customers/78	200
2024-09-28 17:53:10.512874932	GET	/api/v1/products	304
2024-09-28 17:52:45.758492340	PUT	/api/v1/orders	500

Time Stamp ↓	Method	Endpoint	Status Code
2024-09-29 17:33:42.628603260	PUT	/api/v1/orders	500
2024-09-28 17:56:30.392104390	DELETE	/api/v1/products	500
2024-09-28 17:55:45.645839201	POST	/api/v1/users	201
2024-09-28 17:54:59.423715612	PATCH	/api/v1/customers/78	200
2024-09-28 17:53:10.512874932	GET	/api/v1/products	304



Filtering And Sorting

After Getting Logs We Can Use Multiple Filters i.e contain , is any of, start with , end with etc..

Dynamic table column width

Time Stamp	↓ :	Method	Endpoint
2024-09	↑ Sort by ASC	PUT	/api/v1/orders
2024-09	Unsort	DELETE	/api/v1/products
2024-09	Filter	POST	/api/v1/users
2024-09		PATCH	/api/v1/customers/78
2024-09	Hide column	GET	/api/v1/products
2024-09	Manage columns	PUT	/api/v1/orders
2024-09-28 17:51:32.982371090		GET	/api/v1/orders/1027
2024-09-28 17:50:55.483921213		DELETE	/api/v1/products/57



Color Coding

Methods of endpoints and status
Codes are color coded.

Method	Endpoint
PUT	/api/v1/orders
DELETE	/api/v1/products
POST	/api/v1/users
PATCH	/api/v1/customers/78
GET	/api/v1/products
PUT	/api/v1/orders
GET	/api/v1/orders/1027
DELETE	/api/v1/products/57

**Thank
You**

Sehaj Kahlon

Nitin

Nalin Kumar Gupta

Akshit Saini