

★ HTTP Load Testing Practical

- How can we deploy our HTML code, CSS code, Markdown code, etc. using Apache?

1. System Requirement

- a) podman version 3.4.4
- b) Distributor ID: Ubuntu
Description: Ubuntu 22.04.3 LTS
Release: 22.04
Codename: jammy
- c) Date: Mon, 23 Oct 2023 05:45:39 GMT
Server: Apache/2.4.58 (Unix)
Last-Modified: Sun, 22 Oct 2023 09:30:04 GMT
ETag: "4634-6084ac01d1f00"
Accept-Ranges: bytes
Content-Length: 17972
Content-Type: text/html

2. Definition of Apache .

Apache is a popular open-source web server software that serves web pages and content to users when they visit a website. It handles incoming web requests and delivers the requested web pages to the user's browser.

3. System update and Upgrade .

- `sudo apt update`

```

himanshu@123:~/Desktop/grafana-prometheus-blackbox$ sudo apt update
[sudo] password for himanshu: 
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:1 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-proposed InRelease [270 kB]
Fetched 499 kB in 2s (208 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
himanshu@123:~/Desktop/grafana-prometheus-blackbox$

```

- **sudo apt upgrade**

```

himanshu@123:~/Desktop/grafana-prometheus-blackbox$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libgtkglext1
Use 'sudo apt autoremove' to remove it.
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libmagickcore-6.q16-dev python2.7-minimal libmagickwand-dev imagemagick
  libopenexr-dev libopenexr25 libpostproc55 libmagickcore-dev
  libmagickcore-6.q16-6-extra libavcodec58 libmagickwand-6.q16-6 libpython2.7
  libavutil56 imagemagick-6.q16 libswscale5 libmagickcore-6.q16-6
  libswresample3 imagemagick-6-common libmagickcore-6-arch-config ruby-rack
  libavformat58 python2.7-dev libpython2.7-dev libmagickwand-6-headers
  python2.7 libpython2.7-minimal libmagickwand-6.q16-dev
  libmagickcore-6-headers libpython2.7-stdlib libavfilter7
Learn more about Ubuntu Pro at https://ubuntu.com/pro
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
himanshu@123:~/Desktop/grafana-prometheus-blackbox$

```

4. Create File and Folder

- **cd Desktop**
- **mkdir grafana-prometheus-blackbox**
- **cd grafana-prometheus-blackbox**

```

himanshu@123:~/Desktop/grafana-prometheus-blackbox$ ls
10.png 13.png 16.png 19.png 21.png 24.png 27.png 2.png 32.png 35.png 5.png 8.png index.html README.pdf
11.png 14.png 17.png 1.png 22.png 25.png 28.png 30.png 33.png 3.png 6.png 9.png README.jpeg README.png
12.png 15.png 18.png 20.png 23.png 26.png 29.png 31.png 34.png 4.png 7.png Dockerfile README.md

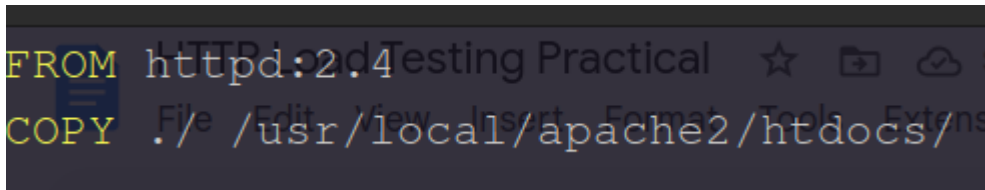
```

5. Create Dockerfile and Paste this Code

- **vim Dockerfile**
- **Paste code in Dockerfile**

```
FROM httpd:2.4
```

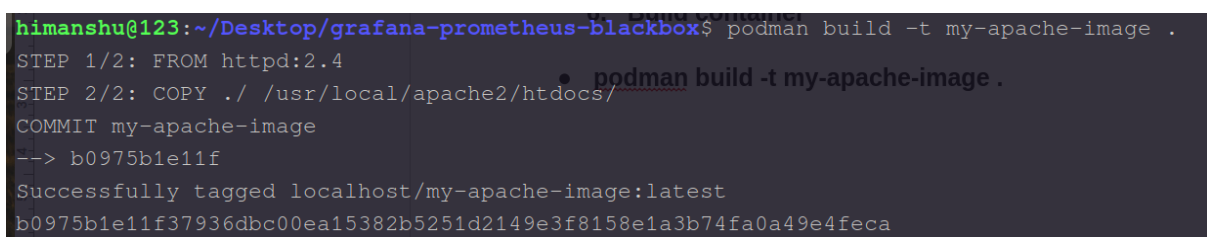
```
COPY ./ /usr/local/apache2/htdocs/
```



- **vim index.html**
- Here is the code of the index. Html paste it in this configuration file :- [index.html code](#)

6. Build container

- **podman build -t my-apache-image .**



- **podman build:** This is the command to initiate the image-building process.
- **-t my-apache-image:** The -t flag is used to set the name and optionally a tag for the image being built. In this case, "my-apache-image" is the name you've given to the image. You can think of it as a user-defined name for the image.

- **.: The dot (.)** at the end of the command represents the build context. It specifies the path to the directory containing the Dockerfile for Container File and any associated files needed to build the image. In this case, it is set to the current directory, indicating that the Dockerfile for Container File is in the same directory where the podman build command is being run.
- **podman images : Check Images**

```
himanshu@123:~/Desktop/grafana-prometheus-blackbox$ podman images
REPOSITORY                                TAG                IMAGE ID           CREATED            SIZE
localhost/my-apache-image                 latest            b0975b1e11f3      3 minutes ago     197 MB
docker.io/library/httpd                   2.4               75a48b16cd56      3 days ago        174 MB
```

7. Run Container

- **podman run -d -p 8080:80 my-apache-image**

```
himanshu@123:~/Desktop/grafana-prometheus-blackbox$ podman run -d -p 8080:80 my-apache-image
acc820708ebc6d825a691ab6276f8d18b7a49877574f5ea8089c85e2a1557825
```

- **podman run:** This is the command to create and run a container.
- **-d:** The -d flag is used to run the container in detached mode. This means the container runs in the background, and you get your command prompt back in the terminal. The container continues to run independently.
- **-p 8080:80:** The -p flag is used to map ports between the host and the container. In this case, it is mapping port 8080 on the host to port 80 in the container. This means that you can access the Apache web server running in the container by connecting to port 8080 on your host machine. Port 80 inside the container is where the Apache web server typically listens.
- **my-apache-image:** This is the name of the image you want to use to create the container. The container will be based on the image named "my-apache-image."

- **podman ps : Check container is Running or Not .**

```
himanshu@123:~/Desktop/grafana-prometheus-blackbox$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
acc820708ebc	localhost/my-apache-image:latest	httpd-foreground	3 minutes ago	Up 3 minutes ago	0.0.0.0:8080->80/tcp	kind_mclar

8. Check Localhost

- **curl -I http://localhost:8080**

```
himanshu@123:~/Desktop/grafana-prometheus-blackbox$ curl -I http://localhost:8080
HTTP/1.1 200 OK
Date: Mon, 23 Oct 2023 06:23:45 GMT
Server: Apache/2.4.58 (Unix)
Last-Modified: Sun, 22 Oct 2023 09:30:04 GMT
ETag: "4634-6084ac01d1f00"
Accept-Ranges: bytes
Content-Length: 17972
Content-Type: text/html
```

- **curl:** Curl is a command-line tool and library for making HTTP requests. In this case, it's being used to interact with a web server.
- **-I:** The -I flag instructs curl to make an HTTP HEAD request instead of a full GET request. An HTTP HEAD request only retrieves the headers of a web page, not the content. This is useful for checking the server's response headers, such as the status code and various header fields, without actually downloading the page content.
- **http://localhost:8080:** This is the URL to which the HTTP HEAD request is being made. In this case, it's set to http://localhost:8080, which typically corresponds to a web server running on your local machine and listening on port 8080.
- **Show on Web Browser**

Set up Grafana , Prometheus and Blackbox Exporter

Table of Content

1- Definition of Grafana , Prometheus and Blackbox Exporter

2- How to set up Grafana , Prometheus and Blackbox Exporter on Podman Container ?

1. Definition of Grafana , Prometheus and Blackbox Exporter

a. Grafana: Grafana is an open-source analytics and monitoring platform that integrates with various data sources, allowing users to visualize and understand metrics through customizable dashboards.

b. Prometheus: Prometheus is an open-source monitoring and alerting toolkit designed for reliability and scalability. It collects and stores time-series data, offering powerful querying and alerting capabilities.

c. Blackbox Exporter: Blackbox Exporter is a Prometheus exporter designed for probing and monitoring external services. It allows users to check the availability and response of endpoints, such as HTTP, TCP, ICMP, and DNS, and generates metrics based on the results.

2. How to set up Grafana , Prometheus and Blackbox Exporter on Podman Container ?

System Requirement

- **Distributor ID:** Ubuntu **Description:** Ubuntu 22.04.3 LTS **Release:** 22.04 **Codename:** jammy
- **podman version 3.4.4**

Step 1 . First, we will update and upgrade our system.

- `sudo apt update`

```
himanshu@123:~/Desktop/grafana-prometheus-blackbox$ sudo apt update
[sudo] password for himanshu: our system
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:1 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-proposed InRelease [270 kB]
Fetched 499 kB in 2s (208 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
← → ↻ localhost:8080
config.yml prometheus.yml
himanshu@123:~/Desktop/grafana$
```

Step 3 . Paste code in prometheus.yml and config.yml file .

- `vim prometheus.yml`

Paste code in this configuration file .

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['192.168.1.113:9090', '192.168.1.113:9115']

  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx]
    static_configs:
      - targets:
        - https://www.google.com

relabel_configs:
  - source_labels: [__address__]
    target_label: __param_target
  - source_labels: [__param_target]
    target_label: instance
  - target_label: __address__
    replacement: 192.168.1.113:9115
```

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s
```

```
modules:
  http_2xx:
    prober: http
  http_post_2xx:
    prober: http
    method: POST
  tcp_connect:
    prober: tcp
  pop3s_banner:
    prober: tcp
    tcp:
      query_response:
        - expect: "+OK"
      tls: true
      tls_config:
        insecure_skip_verify: false
  grpc:
    prober: grpc
    grpc:
      tls: true
      preferred_ip_protocol: "ip4"
  grpc_plain:
    prober: grpc
    grpc:
      tls: false
      service: "service1"
  ssh_banner:
    prober: tcp
    tcp:
      query_response:
        - expect: "SSH-2.0-"
        - send: "SSH-2.0-blackbox-ssh-check"
  irc_banner:
    prober: tcp
    tcp:
      query_response:
        - send: "NICK prober"
        - send: "USER prober prober prober :prober"
        - expect: "PING :([ ]+)"
        - send: "PONG $1)"
        - expect: "^[ ]+ 001"
  icmp:
    prober: icmp
  icmp_ttl5:
    prober: icmp
    timeout: 5s
  icmp:
```

- The content has been written in HTML code and has been set up on Apache. Now, we will check the response of this content.

➤ What are HTTP responses and how many types of HTTP responses are there?

HTTP responses are messages sent by web servers to a client, like a web browser, to show the result of an HTTP request. They convey whether the request was successful ("200 OK"), encountered an error ("404 Not Found" for missing pages), or faced server problems ("500 Internal Server Error").

➤ HTTP Responses

- After a server receives a request (like a GET request), it sends back an HTTP response.
- The response contains information about whether the request was successful and, if so, the data you requested.

There are several common HTTP response types:

- 200 OK:** This means the request was successful, and the server is sending back the requested data. For example, when you load a webpage, you receive a 200 OK response along with the webpage's content.

← → ↻ localhost:8080

Set up Grafana , Prometheus and Blackbox Exporter

Inspector Console Debugger Network Performance Memory Storage Style Editor Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	
200	GET	localhost:8080	33.png	img	png	143.76 kB	143.47 kB	225
200	GET	localhost:8080	34.png	img	png	210.10 kB	209.82 kB	224
200	GET	localhost:8080	10.png	img	png	28.20 kB	27.91 kB	223
200	GET	localhost:8080	11.png	img	png	168.61 kB	168.32 kB	222
200	GET	localhost:8080	12.png	img	png	136.12 kB	135.83 kB	221
200	GET	localhost:8080	13.png	img	png	114.24 kB	113.96 kB	220
200	GET	localhost:8080	14.png	img	png	9.29 kB	9 kB	221
200	GET	localhost:8080	17.png	img	png	37.38 kB	37.09 kB	221
200	GET	localhost:8080	20.png	img	png	77.31 kB	77.03 kB	221
200	GET	localhost:8080	5.png	img	png	133.51 kB	133.23 kB	220
200	GET	localhost:8080	16.png	img	png	11.42 kB	11.13 kB	221
200	GET	localhost:8080	35.png	img	png	51.16 kB	50.87 kB	220
200	GET	localhost:8080	22.png	img	png	112.16 kB	111.88 kB	220
200	GET	localhost:8080	3.png	img	png	227.67 kB	227.38 kB	220
200	GET	localhost:8080	15.png	img	png	21.34 kB	21.06 kB	219
200	GET	localhost:8080	18.png	img	png	49.14 kB	48.85 kB	217
200	GET	localhost:8080	29.png	img	png	314.14 kB	313.85 kB	216
200	GET	localhost:8080	30.png	img	png	161.47 kB	161.18 kB	214
200	GET	localhost:8080	32.png	img	png	73.70 kB	73.41 kB	215
200	GET	localhost:8080	31.png	img	png	162.13 kB	161.84 kB	215
200	GET	localhost:8080	33.png	img	png	cached	147.57 kB	0 ms
200	GET	localhost:8080	34.png	img	png	cached	213.10 kB	0 ms
200	GET	localhost:8080	10.png	img	png	cached	28.78 kB	0 ms
200	GET	localhost:8080	11.png	img	png	cached	172.14 kB	0 ms
200	GET	localhost:8080	12.png	img	png	cached	139.38 kB	0 ms
200	GET	localhost:8080	13.png	img	png	cached	114.80 kB	0 ms
200	GET	localhost:8080	14.png	img	png	cached	12.40 kB	0 ms

41 requests 4.52 MB / 2.24 MB transferred Finish: 1.22 s DOMContentLoaded: 1.20 s load: 1.22 s

← → ↻ localhost:8080

Set up Grafana , Prometheus and Blackbox Exporter

Inspector Console Debugger Network Performance Memory Storage Style Editor Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	
200	GET	localhost:8080	33.png	img	png	143.76 kB	143.47 kB	225
200	GET	localhost:8080	34.png	img	png	210.10 kB	209.82 kB	224
200	GET	localhost:8080	10.png	img	png	28.20 kB	27.91 kB	223
200	GET	localhost:8080	11.png	img	png	168.61 kB	168.32 kB	222
200	GET	localhost:8080	12.png	img	png	136.12 kB	135.83 kB	221
200	GET	localhost:8080	13.png	img	png	114.24 kB	113.96 kB	220
200	GET	localhost:8080	14.png	img	png	9.29 kB	9 kB	221
200	GET	localhost:8080	17.png	img	png	37.38 kB	37.09 kB	221
200	GET	localhost:8080	20.png	img	png	77.31 kB	77.03 kB	221
200	GET	localhost:8080	5.png	img	png	133.51 kB	133.23 kB	220
200	GET	localhost:8080	16.png	img	png	11.42 kB	11.13 kB	221
200	GET	localhost:8080	35.png	img	png	51.16 kB	50.87 kB	220
200	GET	localhost:8080	22.png	img	png	112.16 kB	111.88 kB	220
200	GET	localhost:8080	3.png	img	png	227.67 kB	227.38 kB	220
200	GET	localhost:8080	15.png	img	png	21.34 kB	21.06 kB	219
200	GET	localhost:8080	18.png	img	png	49.14 kB	48.85 kB	217
200	GET	localhost:8080	29.png	img	png	314.14 kB	313.85 kB	216
200	GET	localhost:8080	30.png	img	png	161.47 kB	161.18 kB	214
200	GET	localhost:8080	32.png	img	png	73.70 kB	73.41 kB	215
200	GET	localhost:8080	31.png	img	png	162.13 kB	161.84 kB	215
200	GET	localhost:8080	33.png	img	png	cached	147.57 kB	0 ms
200	GET	localhost:8080	34.png	img	png	cached	213.10 kB	0 ms
200	GET	localhost:8080	10.png	img	png	cached	28.78 kB	0 ms
200	GET	localhost:8080	11.png	img	png	cached	172.14 kB	0 ms
200	GET	localhost:8080	12.png	img	png	cached	139.38 kB	0 ms
200	GET	localhost:8080	13.png	img	png	cached	114.80 kB	0 ms
200	GET	localhost:8080	14.png	img	png	cached	12.40 kB	0 ms

41 requests 4.52 MB / 2.24 MB transferred Finish: 922 ms DOMContentLoaded: 907 ms load: 921 ms

Headers Cookies Request Response Timings

Filter Headers

GET http://localhost:8080/11.png

Status: 200 OK

Version: HTTP/1.1

Transferred: 168.61 kB (168.32 kB size)

Referrer Policy: strict-origin-when-cross-origin

Request Priority: Low

DNS Resolution: System

Response Headers (287 B)

Accept-Ranges: bytes

Connection: Keep-Alive

Content-Length: 168323

Content-Type: image/png

Date: Mon, 23 Oct 2023 06:52:30 GMT

ETag: "29183-6080be337a080"

Keep-Alive: timeout=5, max=98

Last-Modified: Thu, 19 Oct 2023 06:30:10 GMT

Server: Apache/2.4.58 (Ubuntu)

Request Headers (559 B)

Accept: image/avif,image/webp,*/*

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.5

Cache-Control: no-cache

Connection: keep-alive

Cookie: grafana_session=7f9a589890adfa3d9cce9d9cd16c6f0; grafana_session_expiry=1698038820; splunkweb_csrf_token_8000=6599350883249603606

Host: localhost:8080

Pragma: no-cache

Referer: http://localhost:8080/

Sec-Fetch-Dest: image

Sec-Fetch-Mode: no-cors

Sec-Fetch-Site: same-origin

Status	Method	Domain	File
200	GET	🔒 localhost:8080	33.png
200	GET	🔒 localhost:8080	34.png
200	GET	🔒 localhost:8080	10.png
200	GET	🔒 localhost:8080	11.png
200	GET	🔒 localhost:8080	12.png
200	GET	🔒 localhost:8080	13.png
200	GET	🔒 localhost:8080	14.png
200	GET	🔒 localhost:8080	17.png
200	GET	🔒 localhost:8080	20.png
200	GET	🔒 localhost:8080	5.png
200	GET	🔒 localhost:8080	16.png
200	GET	🔒 localhost:8080	35.png
200	GET	🔒 localhost:8080	22.png
200	GET	🔒 localhost:8080	3.png
200	GET	🔒 localhost:8080	15.png
200	GET	🔒 localhost:8080	18.png
200	GET	🔒 localhost:8080	29.png
200	GET	🔒 localhost:8080	30.png
200	GET	🔒 localhost:8080	32.png
200	GET	🔒 localhost:8080	31.png
200	GET	🔒 localhost:8080	33.png
200	GET	🔒 localhost:8080	34.png
200	GET	🔒 localhost:8080	10.png
200	GET	🔒 localhost:8080	11.png
200	GET	🔒 localhost:8080	12.png
200	GET	🔒 localhost:8080	13.png
200	GET	🔒 localhost:8080	14.png

or


Accessibility

Application

1

AllHTMLCSSJSXHRFontsImagesMediaWSOther☐ Disable CacheNo Throttling

HeadersCookiesRequestResponseTimings



Name: 30.png
Dimensions: 1855 x 416
MIME Type: image/png

or

Accessibility

Application

1

AllHTMLCSSJSXHRFontsImagesMediaWSOther☐ Disable CacheNo Throttling

HeadersCookiesRequestResponseTimings

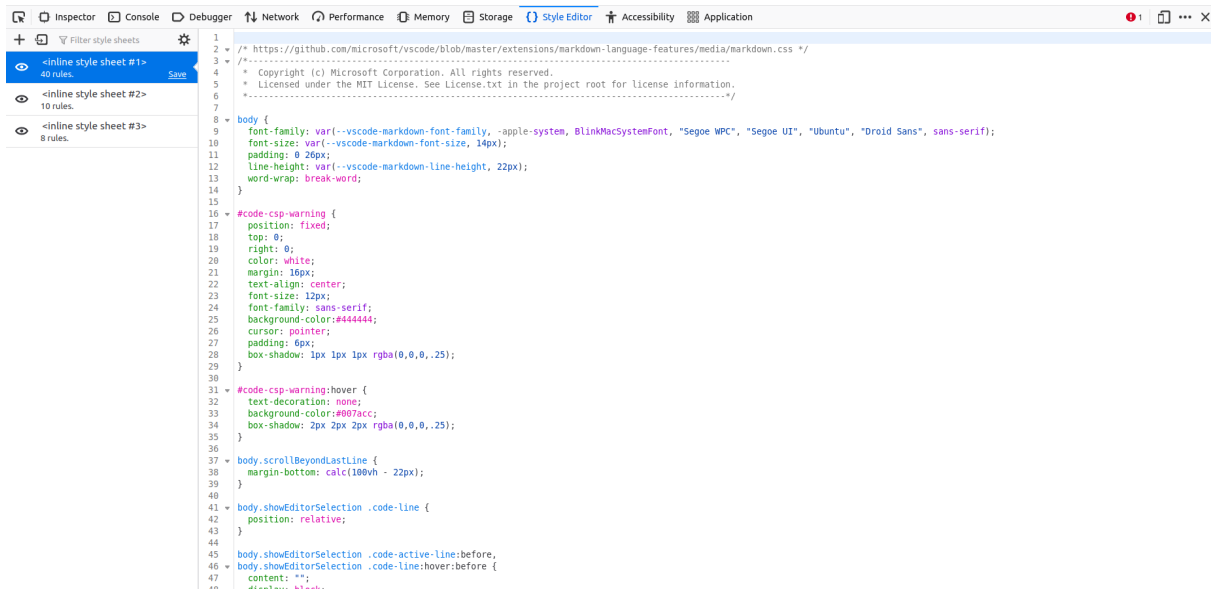
Filter Cookies

Request Cookies

grafana_session: "7f9a589890adfa3d9cce9df9cd16c6f0"

grafana_session_expiry: "1698038820"

splunkweb_csrf_token_8000: "6599350883249603606"



2. 404 Not Found: This response indicates that the server couldn't find the requested resource. It's like a "page not found" error.

or Accessibility Application 1

|| + 🔍 All HTML CSS JS XHR Fonts Images Media WS Other ☐ Disable Cache No Throttling ⚙

📄 Headers Cookies Request Response Timings Stack Trace

🔍 Filter Headers Block Resend

▶ GET http://localhost:8080/favicon.ico

Status **404 Not Found** ⓘ
Version HTTP/1.1
Transferred 410 B (196 B size)
Referrer Policy strict-origin-when-cross-origin
DNS Resolution System

▼ Response Headers (214 B) Raw ☐

- Connection: Keep-Alive
- Content-Length: 196
- Content-Type: text/html; charset=iso-8859-1
- Date: Mon, 23 Oct 2023 06:47:10 GMT
- Keep-Alive: timeout=5, max=97
- Server: Apache/2.4.58 (Unix)

▼ Request Headers (564 B) Raw ☐

- Accept: image/avif,image/webp,*/*
- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-US,en;q=0.5
- Cache-Control: no-cache
- Connection: keep-alive
- Cookie: grafana_session=7f9a589890adfa3d9cce9df9cd16c6f0; grafana_session_expiry=1698038820; splunkweb_csrf_token_8000=6599350883249603606
- Host: localhost:8080
- Pragma: no-cache
- Referer: http://localhost:8080/
- Sec-Fetch-Dest: image
- Sec-Fetch-Mode: no-cors
- Sec-Fetch-Site: same-origin
- User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0

or Accessibility Application 1

|| + 🔍 All HTML CSS JS XHR Fonts Images Media WS Other ☐ Disable Cache No Throttling ⚙

📄 Headers Cookies Request Response Timings Stack Trace

HTML Raw ☐

Not Found

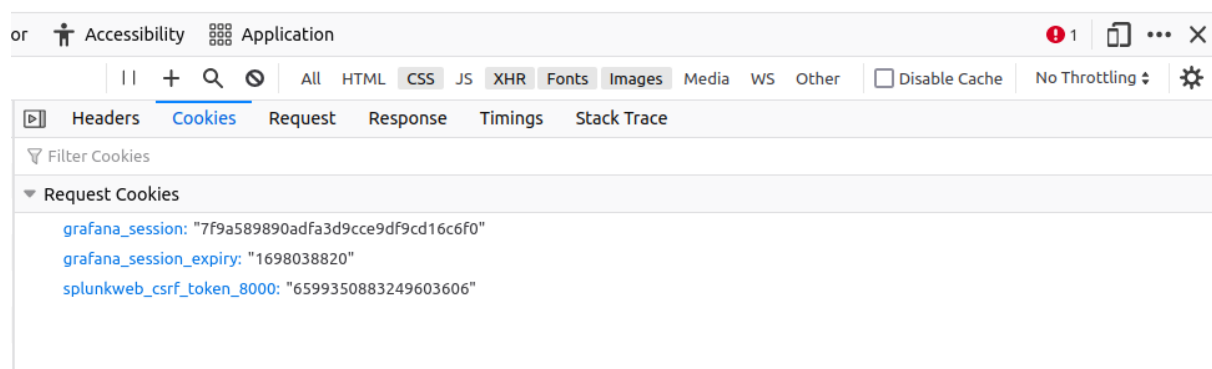
The requested URL was not found on this server.

3. **500 Internal Server Error:** This response indicates that something went wrong on the server while processing your request. It's a generic error message for server-side issues.

4. **301 Moved Permanently:** This is a redirection response. It tells your browser that the requested resource has moved to a new location, and it provides the new URL.
5. **403 Forbidden:** This means that you don't have permission to access the requested resource. It's like an "access denied" error.
6. **502 Bad Gateway:** This response indicates that a server acting as a gateway or proxy received an invalid response from an upstream server.

➤ What are Cookies and How can they work ?

Cookies are small pieces of data that websites store on your computer. They help websites remember information about you, like login status or preferences. When you revisit the site, your browser sends the cookie data back, so the website recognizes you without needing to log in again.

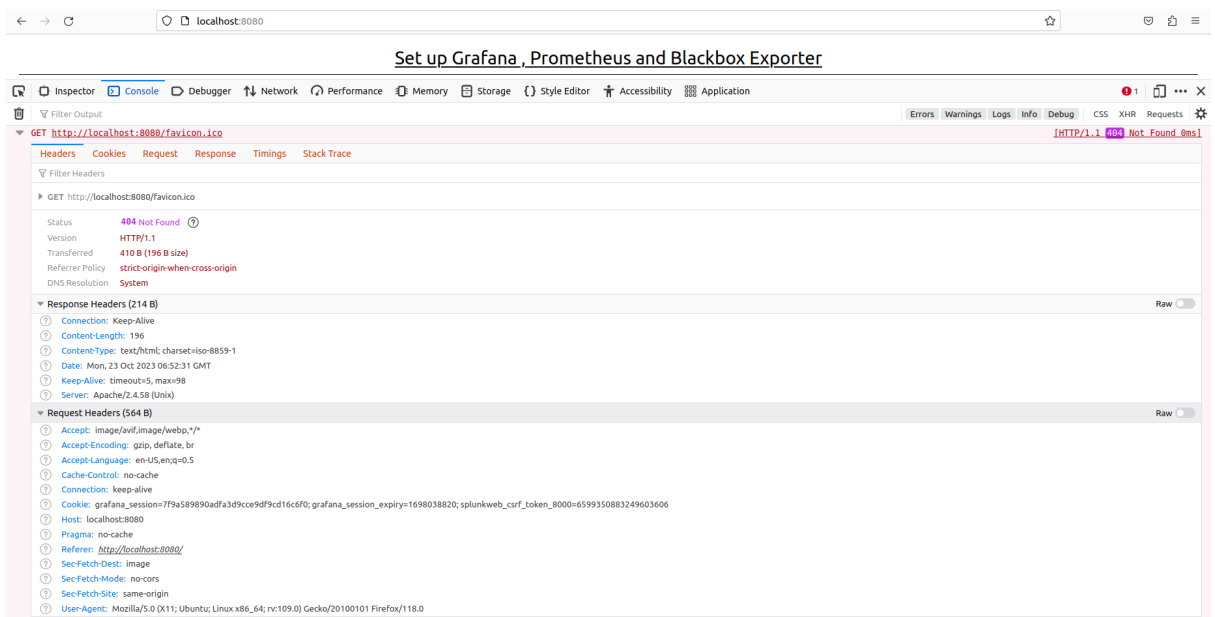
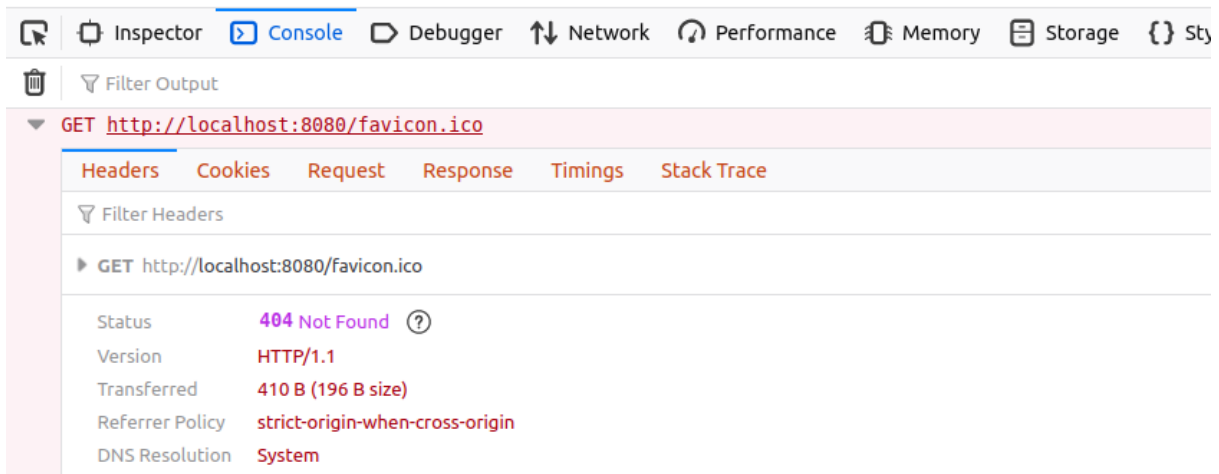


➤ What is Console and How can it work ?

The **console** in web development is like a special tool that helps web developers find and fix problems in their websites. It's like a secret window where you can see errors, test code, and check how the website talks to the internet. It's very useful for making sure websites work correctly.



Set up Grafana , Prometheus



➤ What are Common Headers ?

Common HTTP headers include **"Content-Type"** for specifying the type of data in the response, **"User-Agent"** to identify the client making the request, and **"Authorization"** for authentication. These headers help in data interpretation, client identification, and security within HTTP requests and responses.

Response Headers (287 B)

Raw 

? Accept-Ranges: bytes
? Connection: Keep-Alive
? Content-Length: 168323
? Content-Type: image/png
? Date: Mon, 23 Oct 2023 06:52:30 GMT
? ETag: "29183-6080be337a080"
? Keep-Alive: timeout=5, max=98
? Last-Modified: Thu, 19 Oct 2023 06:30:10 GMT
? Server: Apache/2.4.58 (Unix)

Request Headers (559 B)

Raw 

? Accept: image/avif,image/webp,*/
? Accept-Encoding: gzip, deflate, br
? Accept-Language: en-US,en;q=0.5
? Cache-Control: no-cache
? Connection: keep-alive
? Cookie: grafana_session=7f9a589890adfa3d9cce9df9cd16c6f0; grafana_session_expiry=1698038820; splunkweb_csrf_token_8000=6599350883249603606
? Host: localhost:8080
? Pragma: no-cache
? Referer: <http://localhost:8080/>
? Sec-Fetch-Dest: image
? Sec-Fetch-Mode: no-cors
? Sec-Fetch-Site: same-origin
? User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0