

# Monitoring CI/CD PHP Web Application with Grafana and Prometheus

<b>Version</b>	1.0.0
<b>Prepared by</b>	Rimah Houssameldine, Wissam Hassan
<b>Audience</b>	ParkInnovation Team.
<b>Date</b>	2024-3-3

## Table of Contents

Introduction: .....	2
Technologies Used: .....	2
Prerequisites .....	2
Steps for Monitoring CI/CD PHP Web Application with Grafana and Prometheus .....	4
Setting Up Prometheus .....	4
Docker Engine Configuration.....	8
Setting Up Grafana .....	9
Docker metrics visualization .....	11
Jenkins metrics visualization .....	11
Windows Server metrics visualization .....	12
Prometheus server metrics visualization .....	13
Alerting Setup.....	14
Best Practices .....	17
Conclusion .....	17

## Introduction:

This documentation provides step-by-step instructions for setting up monitoring for a CI/CD PHP web application using Grafana and Prometheus. Monitoring your CI/CD pipeline is crucial for ensuring the reliability and performance of your application.

## Technologies Used:

- Grafana: A visualization and monitoring tool.
- Prometheus: A monitoring and alerting toolkit.
- Docker: For containerization.
- Jenkins: CI/CD automation server.
- Windows Server: Host for the PHP web application.

## Prerequisites

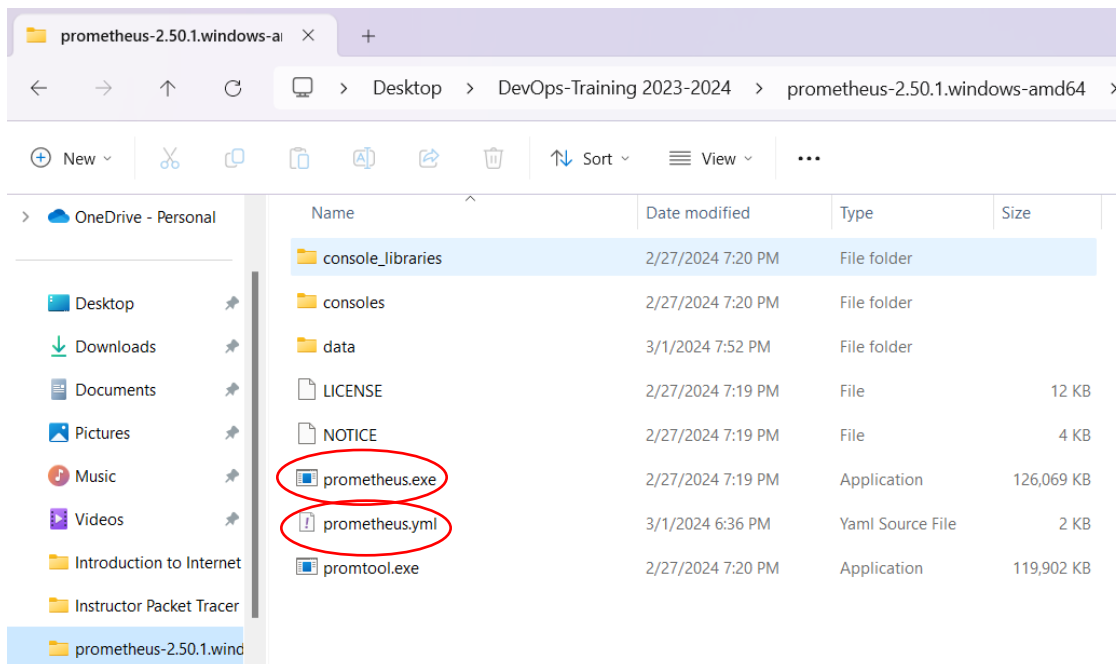
Before proceeding, ensure you have the following:

- Access to Grafana and Prometheus servers.
- Docker installed on the target hosts.
- Jenkins server up and running on localhost:8080.
- Prometheus server up and running on localhost:9090.
- Windows Server with appropriate metrics enabled.

# Steps for Monitoring CI/CD PHP Web Application with Grafana and Prometheus

## Setting Up Prometheus

Configure Prometheus scrape jobs for Docker, Jenkins, Windows Server, and Prometheus itself.



```
! prometheus.yml X
C: > Users > user > Desktop > DevOps-Training 2023-2024 > prometheus-2.50.1.windows-amd64 > ! prometheus.yml > ...
1  # my global config
2  global:
3      scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
4      evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
5      # scrape_timeout is set to the global default (10s).
6
7  # Alertmanager configuration
8  alerting:
9      alertmanagers:
10         - static_configs:
11             - targets:
12                 # - alertmanager:9093
13
14  # Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
15  rule_files:
16      # - "first_rules.yml"
17      # - "second_rules.yml"
18
19  # A scrape configuration containing exactly one endpoint to scrape:
20  # Here it's Prometheus itself.
21  scrape_configs:
22      # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
23      - job_name: "prometheus"
24
25        # metrics_path defaults to '/metrics'
26        # scheme defaults to 'http'.
27
28        static_configs:
29            - targets: ["localhost:9090"]
30
31            # Add this for Jenkins
32      - job_name: "jenkins"
33        metrics_path: "/prometheus"
34        static_configs:
35            - targets: ["localhost:8080"]
36
37      - job_name: "windows_exporter"
38        metrics_path: "/metrics"
39        static_configs:
40            - targets: ["localhost:9182"]
41  ..
```

```
! prometheus.yml X
C: > Users > user > Desktop > DevOps-Training 2023-2024 > prometheus-2.50.1.windows-amd64 > ! prometheus.yml > ...
21 scrape_configs:
42   - job_name: 'php_app'
43     static_configs:
44       - targets: ["127.0.0.1:55832"] # Change this to match your PHP application's address
45         metrics_path: "/metrics" # Change this to match the path where your metrics are exposed
46
47   - job_name: "docker_desktop"
48     metrics_path: "/metrics"
49     static_configs:
50       - targets: ["127.0.0.1:9323"]
51     relabel_configs:
52       - source_labels: [__address__]
53         target_label: instance
54         replacement: "minikube"
55     metric_relabel_configs:
56       - source_labels: [node]
57         target_label: node
58         regex: '(minikube)'
59         replacement: $1
60       - source_labels: [container_name]
61         target_label: container
62         regex: '.*'
63         replacement: "minikube"
64       - source_labels: [__name__]
65         target_label: __name__
66         regex: 'node_memory_.*'
67         action: keep
68
69
70
```

This YAML configuration file for Prometheus sets the global scrape and evaluation intervals to 15 seconds each, along with a default scrape timeout of 10 seconds. Alerting configurations, although present, are commented out, indicating no specific alert manager is configured. Rule files are also commented out, suggesting that no additional rule files are loaded. The file then defines scrape configurations for various job types including Prometheus itself, Jenkins, Windows Server, a PHP application, and Docker Desktop. Each job specifies the targets and metrics paths for scraping metrics, with additional configurations for modifying labels and filtering metrics as required.

Verify Prometheus configuration by accessing Prometheus web UI (<http://prometheus-server-ip:9090>).

```
C:\Users\User\Desktop\DevOp >
ts=2024-03-02T09:41:49.694Z caller=head.go:736 level=info component=tsdb msg="WAL checkpoint loaded"
ts=2024-03-02T09:41:49.702Z caller=head.go:771 level=info component=tsdb msg="WAL segment loaded" segment=53 maxSegment=57
ts=2024-03-02T09:41:49.711Z caller=head.go:771 level=info component=tsdb msg="WAL segment loaded" segment=54 maxSegment=57
ts=2024-03-02T09:41:49.735Z caller=head.go:771 level=info component=tsdb msg="WAL segment loaded" segment=55 maxSegment=57
ts=2024-03-02T09:41:49.784Z caller=head.go:771 level=info component=tsdb msg="WAL segment loaded" segment=56 maxSegment=57
ts=2024-03-02T09:41:49.785Z caller=head.go:771 level=info component=tsdb msg="WAL segment loaded" segment=57 maxSegment=57
ts=2024-03-02T09:41:49.785Z caller=head.go:808 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=126.7134ms wal_replay_duration=90.8666ms wbl_replay_duration=0s total_replay_duration=227.9983ms
ts=2024-03-02T09:41:49.795Z caller=main.go:1139 level=info fs_type=unknown
ts=2024-03-02T09:41:49.795Z caller=main.go:1142 level=info msg="TSDB started"
ts=2024-03-02T09:41:49.795Z caller=main.go:1324 level=info msg="Loading configuration file" filename=prometheus.yml
ts=2024-03-02T09:41:50.734Z caller=main.go:1361 level=info msg="Completed loading of configuration file" filename=prometheus.yml totalDuration=939.3073ms db_storage=0s remote_storage=0s web_handler=0s query_engine=0s scrape=936.8019ms scrape_sd=0s notify=0s notify_sd=0s rules=0s tracing=502.3µs
ts=2024-03-02T09:41:50.734Z caller=main.go:1103 level=info msg="Server is ready to receive web requests."
ts=2024-03-02T09:41:50.734Z caller=manager.go:146 level=info component="rule manager" msg="Starting rule manager..."
ts=2024-03-02T09:41:56.686Z caller=compact.go:569 level=info component=tsdb msg="write block" mint=1709308800000 maxt=1709316000000 ulid=01HQZ8NA80NPYVWKECNSY0BQJ duration=78.6408ms
ts=2024-03-02T09:41:56.700Z caller=head.go:1331 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=5.3854ms
ts=2024-03-02T09:41:56.702Z caller=checkpoint.go:100 level=info component=tsdb msg="Creating checkpoint" from_segment=53 to_segment=55 mint=1709316000000
ts=2024-03-02T09:41:56.797Z caller=head.go:1293 level=info component=tsdb msg="WAL checkpoint complete" first=53 last=55 duration=95.2804ms
```

localhost:9090/targets?search=

Prometheus Alerts Graph Status Help

### Targets

All scrape pools All Unhealthy Expand All Filter by endpoint or labels Unknown Unhealthy Healthy

#### docker\_desktop (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://127.0.0.1:9323/metrics	UP	instance="minikube" job="docker_desktop"	13.340s ago	2.686ms	

#### jenkins (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8080/prometheus	UP	instance="localhost:8080" job="jenkins"	5.108s ago	11.299ms	

#### php\_app (0/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://127.0.0.1:55832/metrics	DOWN	instance="127.0.0.1:55832" job="php_app"	1.967s ago	0s	Get "http://127.0.0.1:55832/metrics": dial tcp 127.0.0.1:55832: connect: No connection could be made because the target machine actively refused it.

prometheus (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	<div>UP</div>	<div>instance="localhost:9090"</div> <div>job="prometheus"</div> <div></div>	4.513s ago	10.151ms	

windows\_exporter (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9182/metrics">http://localhost:9182/metrics</a>	<div>UP</div>	<div>instance="localhost:9182"</div> <div>job="windows_exporter"</div> <div></div>	5.889s ago	607.328ms	

## Docker Engine Configuration

### Docker Engine v24.0.7

Configure the Docker daemon by typing a json Docker daemon [configuration file](#) [↗](#).  
This can prevent Docker from starting. Use at your own risk.

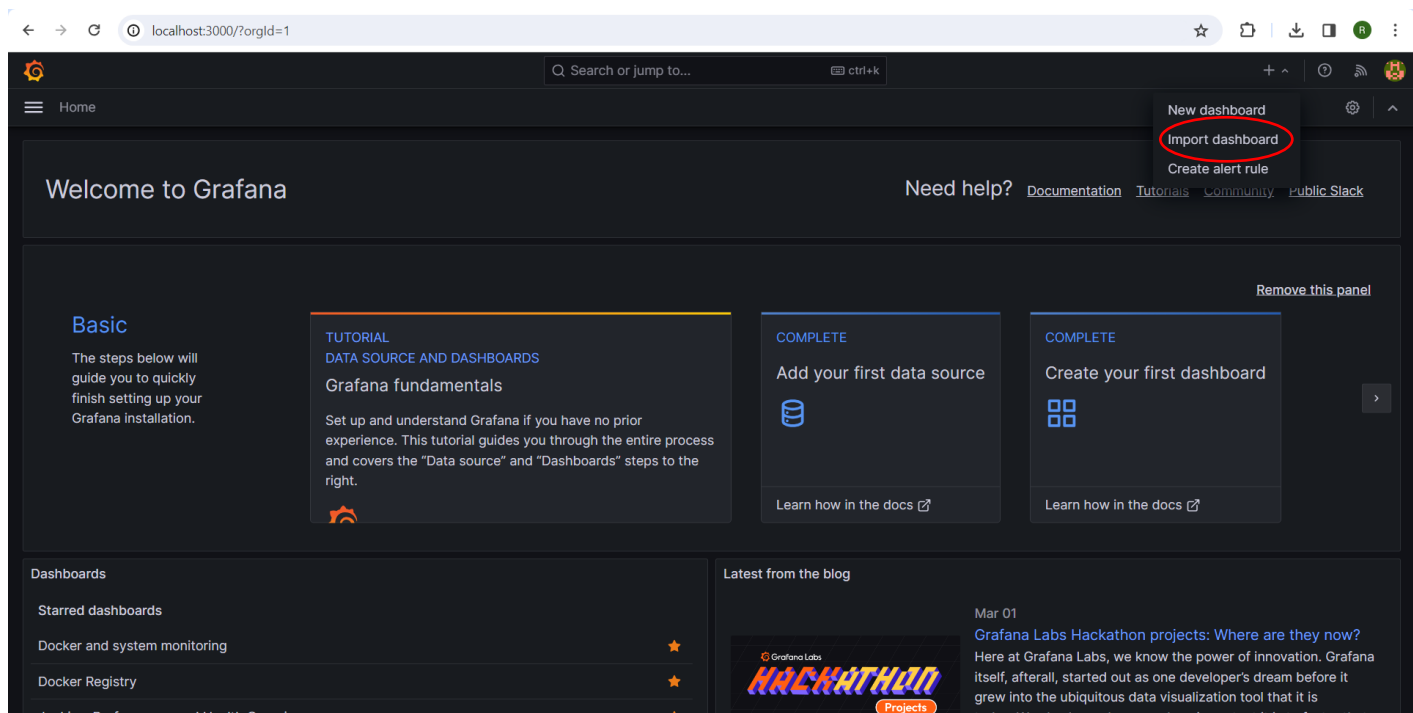
```
{
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "experimental": false,
  "metrics-addr": "127.0.0.1:9323"
}
```

This JSON snippet configures settings for a container runtime environment, likely Docker. Within the "builder" section, it defines parameters related to garbage collection, including the default amount of storage to retain ("20GB") and whether garbage collection is enabled ("true"). The "experimental" parameter is set to "false," indicating that experimental features are disabled. Lastly, the "metrics-add" parameter specifies the address and port ("127.0.0.1:9323") where metrics are exposed, facilitating monitoring of the container runtime environment. Overall, this configuration snippet enables efficient resource management, ensures stability by disabling experimental features, and provides metrics for monitoring purposes.

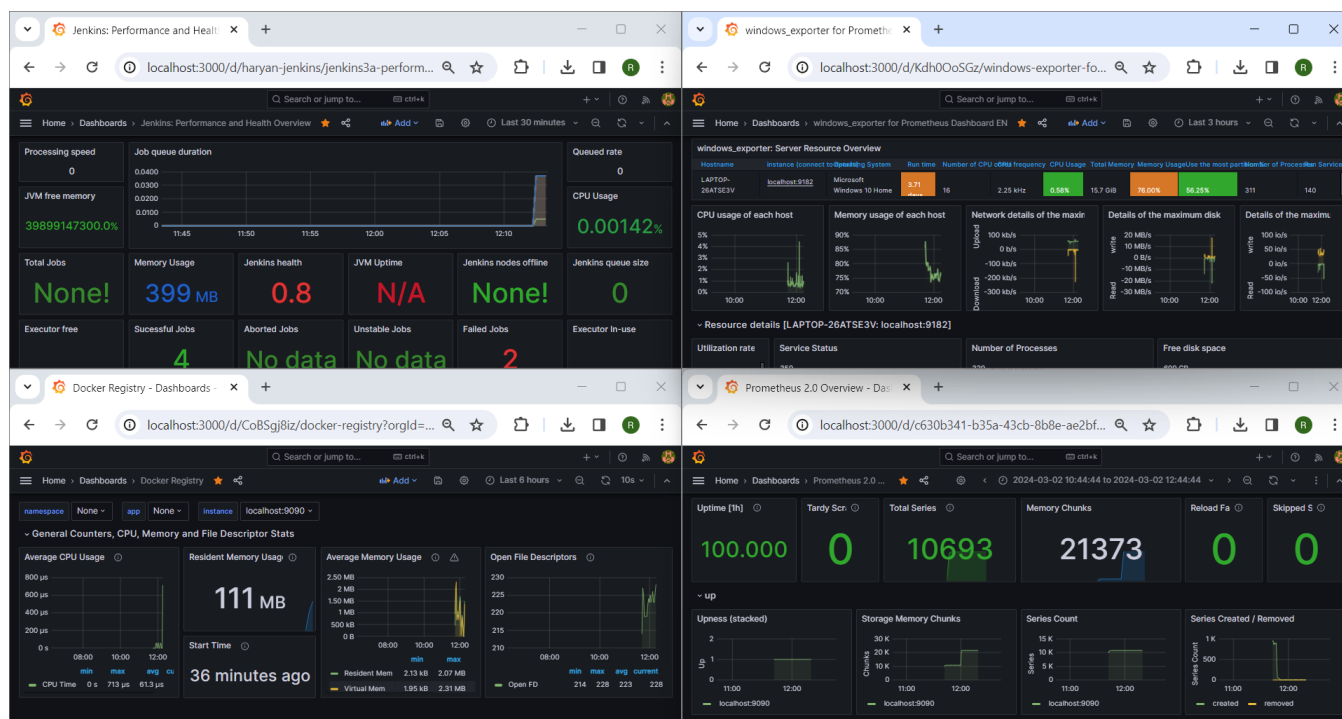
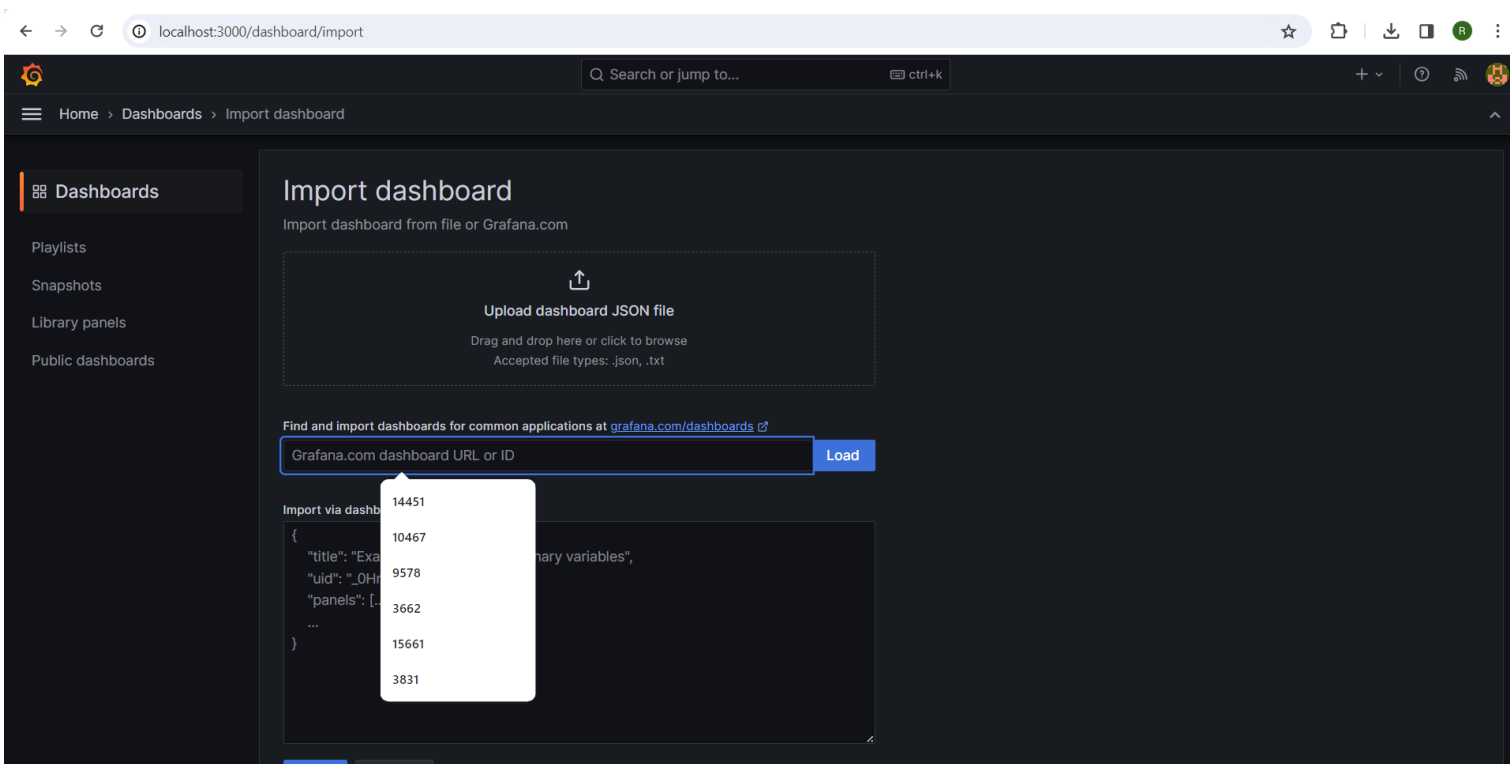


## Setting Up Grafana

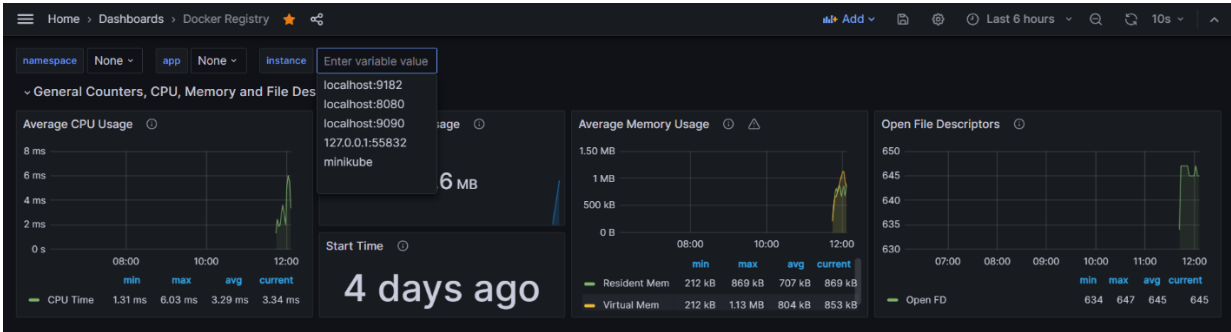
Verify Grafana configuration by accessing Grafana web UI (<http://grafana-server-ip:3000>).



Create Grafana dashboard for Docker metrics visualization, Jenkins metrics visualization, Windows Server metrics visualization, and Prometheus server metrics visualization.



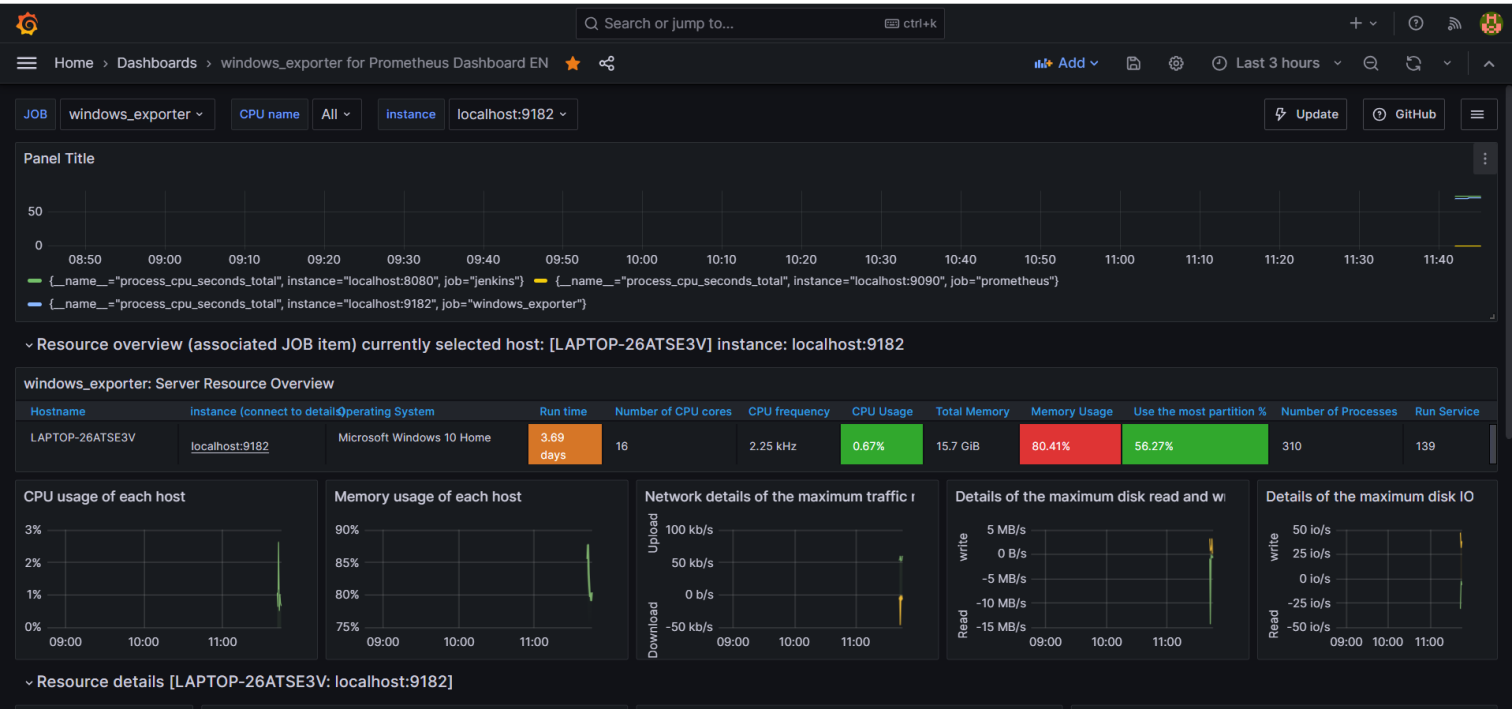
Docker metrics visualization



Jenkins metrics visualization

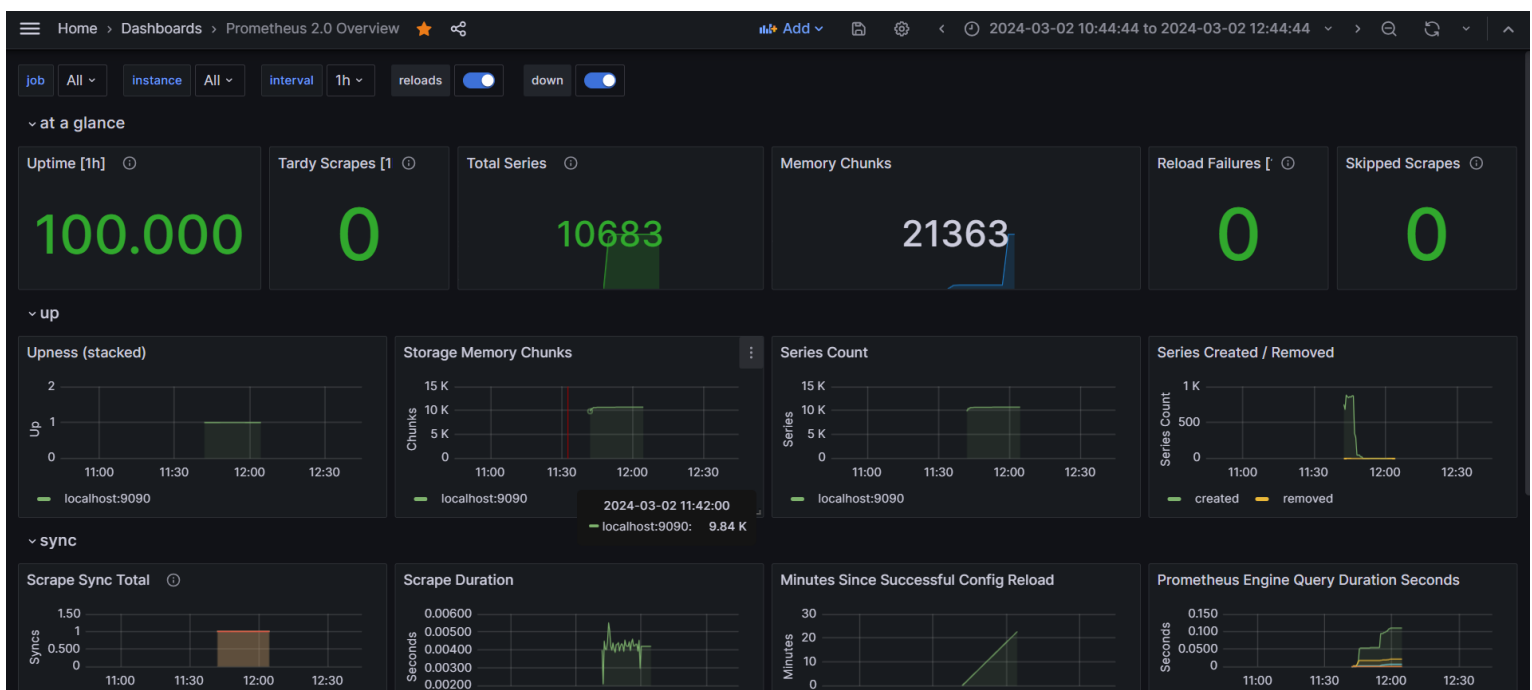


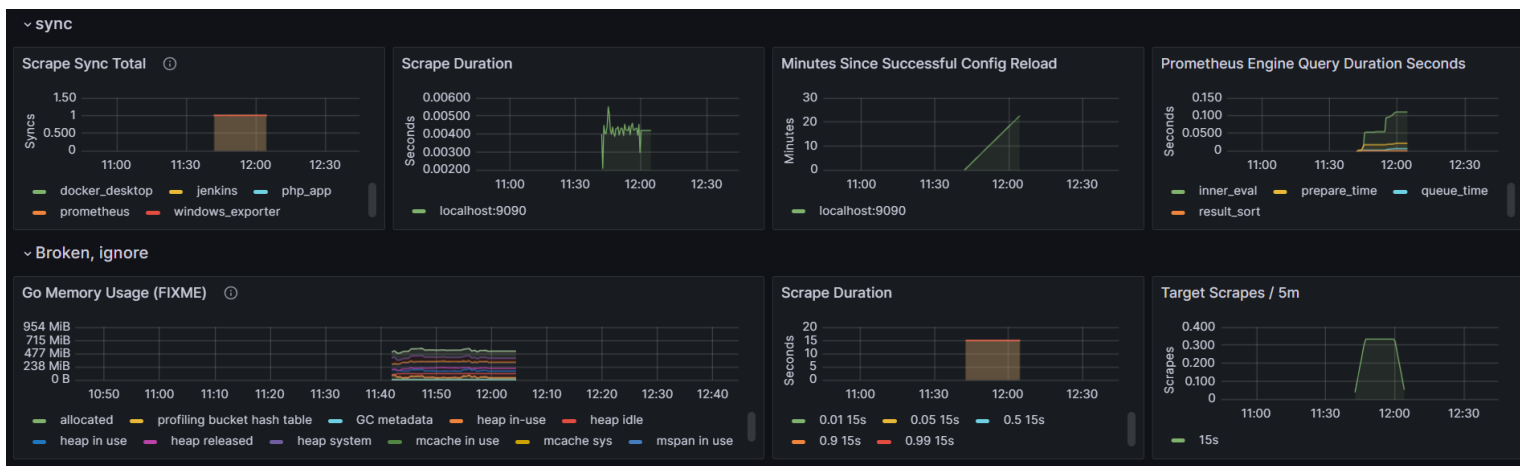
Windows Server metrics visualization





## Prometheus server metrics visualization

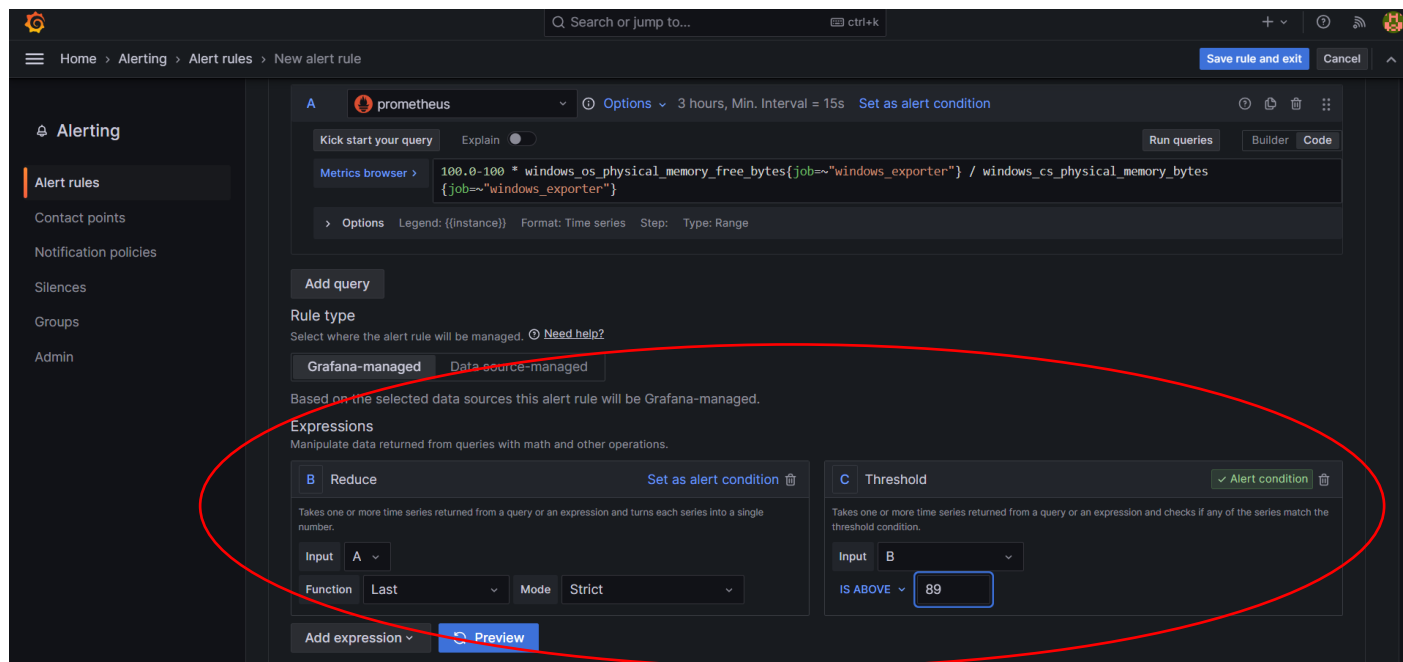




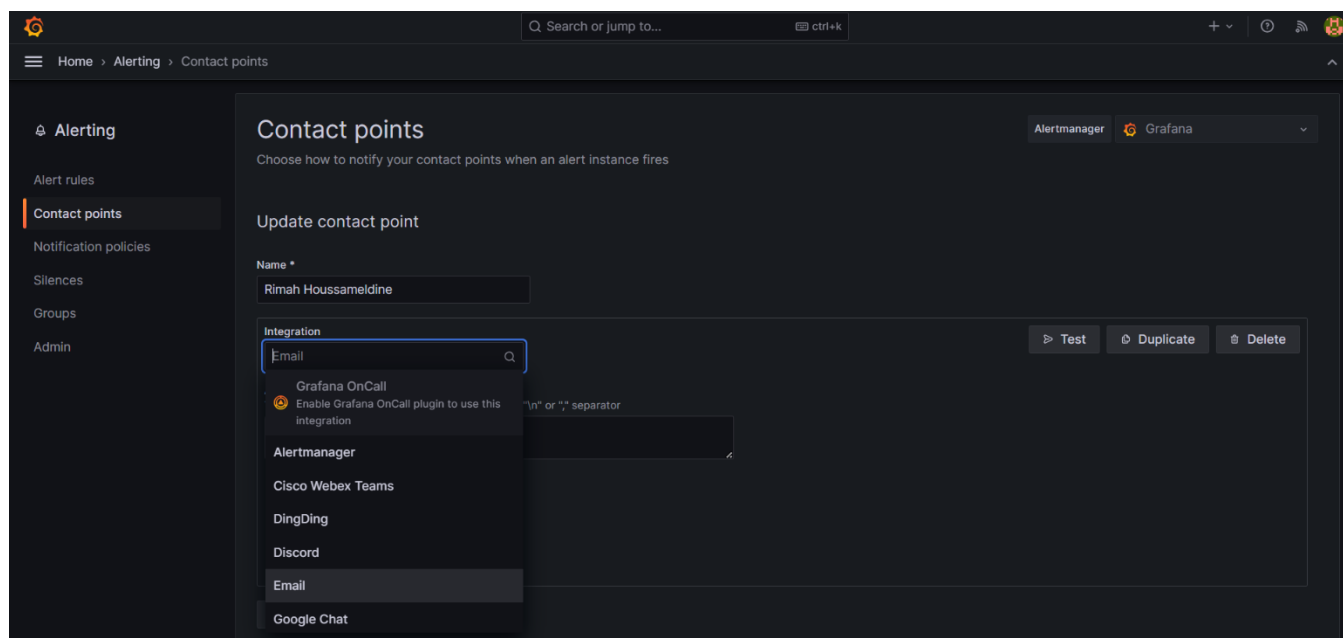
## Alerting Setup

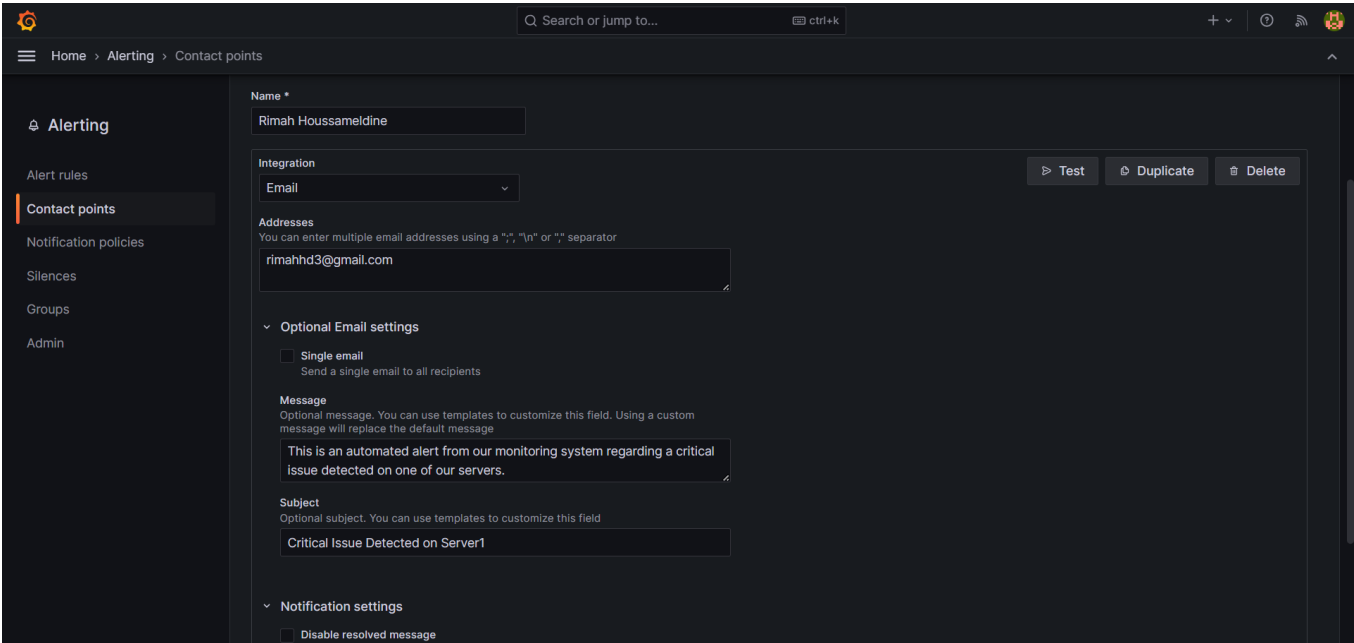
Set up alerting rules in Prometheus for critical metrics.

The screenshot shows the Prometheus Alerting Setup interface. The 'Expressions' section is highlighted with a red circle, showing the configuration for a 'Reduce' expression and a 'Threshold' condition. The 'Reduce' expression is configured with 'Input: Choose', 'Function: Last', and 'Mode: Strict'. The 'Threshold' condition is configured with 'Input: A' and 'IS ABOVE 60'. The 'Rule type' is set to 'Grafana-managed'.

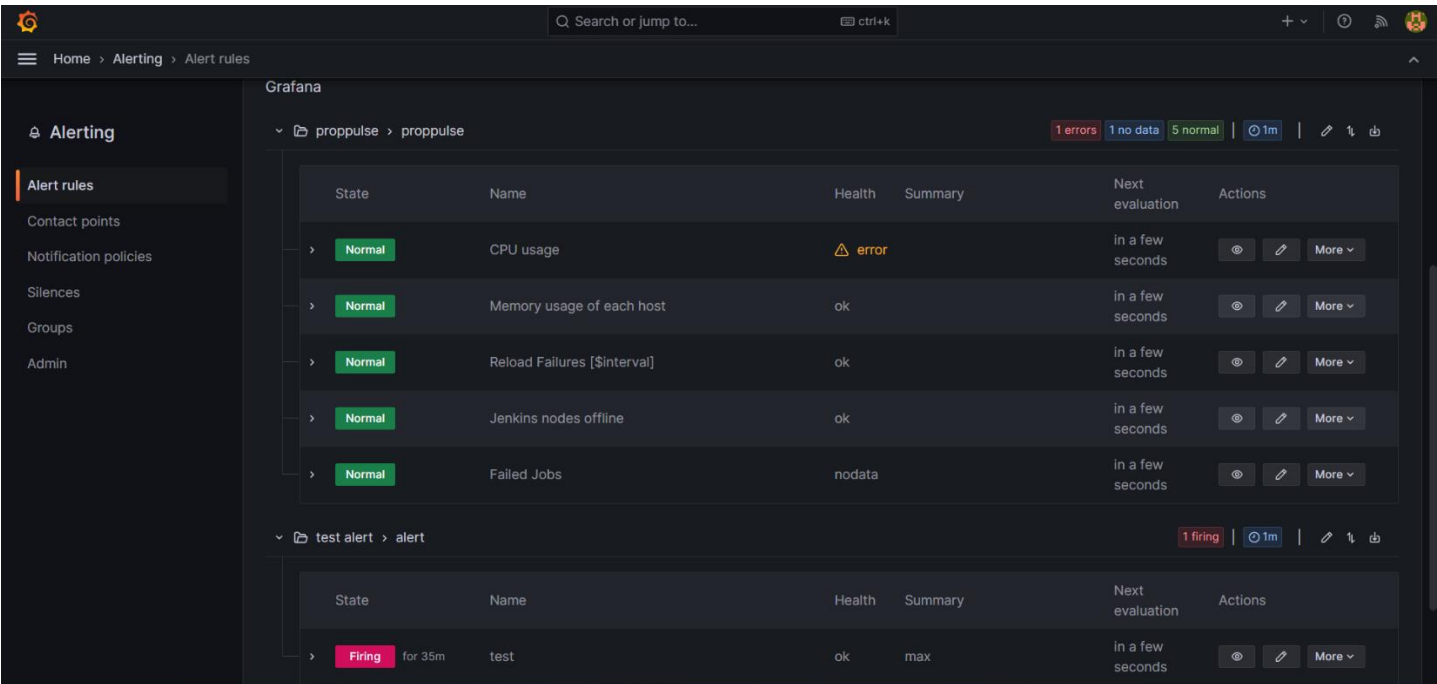


Configure alert notification channels in Grafana (email, Slack, etc.).





Test alerting setup to ensure proper functioning.





## Best Practices

Regularly review and optimize dashboards and alerting rules.  
Ensure monitoring covers key performance indicators (KPIs) for your application.  
Implement automated monitoring checks as part of your CI/CD pipeline.

## Conclusion

Monitoring your CI/CD PHP web application with Grafana and Prometheus enhances visibility into your deployment pipeline, enabling proactive issue detection and resolution. Follow the steps outlined in this documentation to establish a robust monitoring infrastructure for your application.