

## Project Monitoring System

### 1. Grafana monitor CPU dan Memory

Setup Grafana, Prometheus dan node exporter menggunakan docker compose.

```
ubuntu@ip-10-130-15-144:~/grafana$ tree
.
├── docker-compose.yml
├── grafana-provisioning
│   ├── dashboards
│   │   ├── dashboards.yaml
│   │   └── my_dashboard.json
│   └── datasources
│       └── datasources.yaml
└── prometheus.yml
```

#### \$ nano docker-compose.yml

```
version: '3'
services:
  prometheus:
    image: prom/prometheus:latest
    command:
      - --storage.tsdb.retention.time=7d
      - --config.file=/etc/prometheus/prometheus.yml
    ports:
      - "9090:9090"
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
  grafana:
    image: grafana/grafana:latest
    ports:
      - "3000:3000"
    volumes:
      - ./grafana-provisioning/dashboards:/etc/grafana/provisioning/dashboards
      - ./grafana-provisioning/datasources:/etc/grafana/provisioning/datasources
  node_exporter_1:
    image: prom/node-exporter:latest
    ports:
      - "9100:9100"
  node_exporter_2:
    image: prom/node-exporter:latest
    ports:
      - "9101:9100"
```

#### \$ nano prometheus.yml

```
# prometheus.yml
scrape_configs:
  - job_name: 'node'
    scrape_interval: 5s
    static_configs:
      - targets: ['node_exporter_1:9100', 'node_exporter_2:9100']
```

## \$ nano dashboards.yaml

```
apiVersion: 1
providers:
- name: 'default'
  orgId: 1
  folder: ""
  type: file
  disableDeletion: false
  editable: true
  options:
    path: /etc/grafana/provisioning/dashboards
```

## \$ nano my\_dashboard.json

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": {
          "type": "grafana",
          "uid": "-- Grafana --"
        },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 0,
  "links": [],
  "panels": [
    {
      "datasource": {
        "type": "prometheus",
        "uid": "PBFA97CFB590B2093"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
```

```
"barWidthFactor": 0.6,
"drawStyle": "line",
"fillOpacity": 0,
"gradientMode": "none",
"hideFrom": {
  "legend": false,
  "tooltip": false,
  "viz": false
},
"insertNulls": false,
"lineInterpolation": "linear",
"lineWidth": 1,
"pointSize": 5,
"scaleDistribution": {
  "type": "linear"
},
"showPoints": "auto",
"spanNulls": false,
"stacking": {
  "group": "A",
  "mode": "none"
},
"thresholdsStyle": {
  "mode": "off"
}
},
"mappings": [],
"thresholds": {
  "mode": "absolute",
  "steps": [
    {
      "color": "green",
      "value": null
    },
    {
      "color": "red",
      "value": 80
    }
  ]
}
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 0
},
"id": 1,
"options": {
  "legend": {
```

```

    "calcs": [],
    "displayMode": "list",
    "placement": "bottom",
    "showLegend": true
  },
  "tooltip": {
    "mode": "single",
    "sort": "none"
  }
},
"targets": [
  {
    "datasource": {
      "type": "prometheus",
      "uid": "PBFA97CFB590B2093"
    },
    "expr": "100 - (avg(irate(node_cpu_seconds_total{mode='idle'}[5m])) by (instance) * 100)",
    "legendFormat": "{{ instance }}",
    "refId": "A"
  }
],
"timeFrom": "1h",
"title": "CPU Usage",
"type": "timeseries"
},
{
  "datasource": {
    "type": "prometheus",
    "uid": "PBFA97CFB590B2093"
  },
  "fieldConfig": {
    "defaults": {
      "color": {
        "mode": "palette-classic"
      },
    },
    "custom": {
      "axisBorderShow": false,
      "axisCenteredZero": false,
      "axisColorMode": "text",
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "barWidthFactor": 0.6,
      "drawStyle": "line",
      "fillOpacity": 0,
      "gradientMode": "none",
      "hideFrom": {
        "legend": false,
        "tooltip": false,
        "viz": false
      },
    },
    "insertNulls": false,
  }
}

```

```
"lineInterpolation": "linear",
"lineWidth": 1,
"pointSize": 5,
"scaleDistribution": {
  "type": "linear"
},
"showPoints": "auto",
"spanNulls": false,
"stacking": {
  "group": "A",
  "mode": "none"
},
"thresholdsStyle": {
  "mode": "off"
}
},
"mappings": [],
"thresholds": {
  "mode": "absolute",
  "steps": [
    {
      "color": "green",
      "value": null
    },
    {
      "color": "red",
      "value": 80
    }
  ]
}
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 8
},
"id": 2,
"options": {
  "legend": {
    "calcs": [],
    "displayMode": "list",
    "placement": "bottom",
    "showLegend": true
  },
  "tooltip": {
    "mode": "single",
    "sort": "none"
  }
},
}
```

```
"targets": [
  {
    "datasource": {
      "type": "prometheus",
      "uid": "PBFA97CFB590B2093"
    },
    "expr": "node_memory_MemTotal_bytes - node_memory_MemFree_bytes",
    "legendFormat": "{{ instance }}",
    "refId": "A"
  }
],
"timeFrom": "1h",
"title": "Memory Usage",
"type": "timeseries"
},
{
  "datasource": {
    "type": "prometheus",
    "uid": "PBFA97CFB590B2093"
  },
  "fieldConfig": {
    "defaults": {
      "color": {
        "mode": "palette-classic"
      },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "barWidthFactor": 0.6,
        "drawStyle": "bars",
        "fillOpacity": 0,
        "gradientMode": "none",
        "hideFrom": {
          "legend": false,
          "tooltip": false,
          "viz": false
        },
        "insertNulls": false,
        "lineInterpolation": "linear",
        "lineWidth": 1,
        "pointSize": 5,
        "scaleDistribution": {
          "type": "linear"
        },
        "showPoints": "auto",
        "spanNulls": false,
        "stacking": {
          "group": "A",
```

```
    "mode": "none"
  },
  "thresholdsStyle": {
    "mode": "off"
  }
},
"mappings": [],
"thresholds": {
  "mode": "absolute",
  "steps": [
    {
      "color": "green",
      "value": null
    },
    {
      "color": "red",
      "value": 80
    }
  ]
}
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 16
},
"id": 3,
"options": {
  "legend": {
    "calcs": [],
    "displayMode": "list",
    "placement": "bottom",
    "showLegend": true
  },
  "tooltip": {
    "mode": "single",
    "sort": "none"
  }
},
"targets": [
  {
    "datasource": {
      "type": "prometheus",
      "uid": "PBFA97CFB590B2093"
    },
    "expr": "node_filesystem_size_bytes{fstype!~'tmpfs|squashfs'} -
node_filesystem_free_bytes{fstype!~'tmpfs|squashfs'}",
    "legendFormat": "{{ instance }}",
    "refId": "A"
  }
]
```

```
    }
  ],
  "timeFrom": "1h",
  "title": "Disk Usage",
  "type": "timeseries"
}
],
"schemaVersion": 39,
"tags": [],
"templating": {
  "list": []
},
"time": {
  "from": "now-1h",
  "to": "now"
},
"timepicker": {
  "refresh_intervals": [
    "5s",
    "10s",
    "30s",
    "1m",
    "5m",
    "15m",
    "30m",
    "1h",
    "2h",
    "1d"
  ],
  "time_options": [
    "5m",
    "15m",
    "1h",
    "6h",
    "12h",
    "24h",
    "2d",
    "7d",
    "30d"
  ]
},
"timezone": "browser",
"title": "Node Monitoring",
"uid": "prod-node-monitoring",
"version": 1,
"weekStart": ""
}
```



\$ nano datasources.yaml

apiVersion: 1

datasources:

- name: Prometheus

type: prometheus

url: http://prometheus:9090

access: proxy

\$ docker-compose up -d

\$ docker compose ps -a

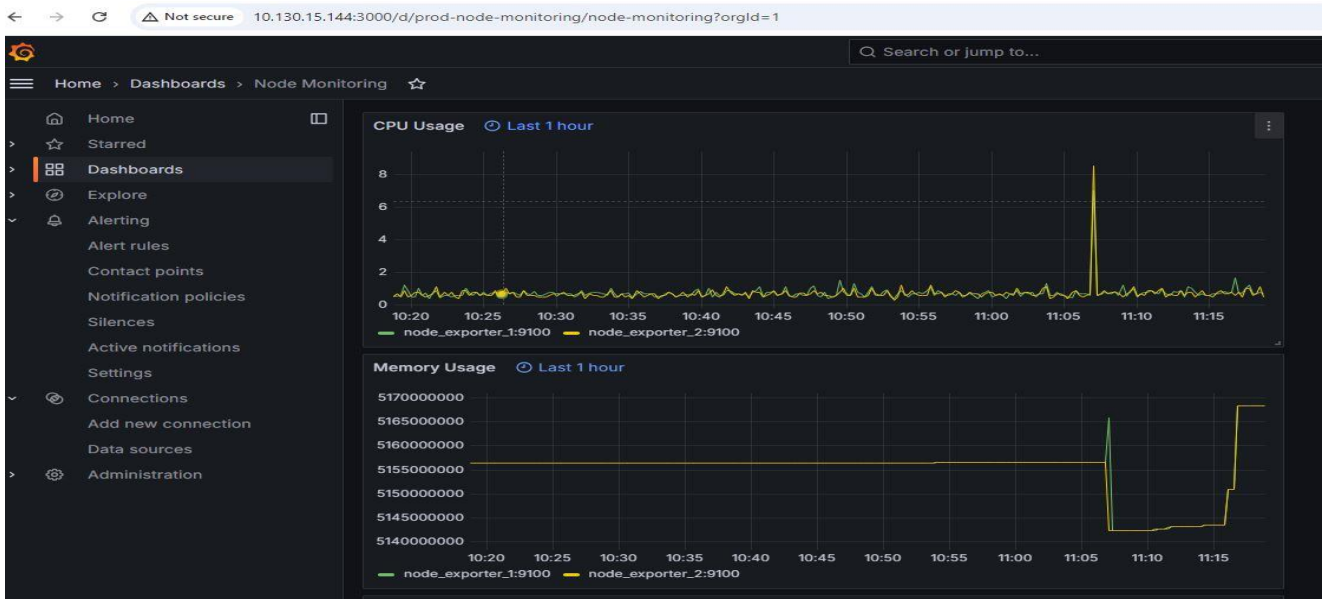
Mari kita cek via browser apakah sudah bisa diakses untuk Grafana dan prometheus.

```
ubuntu@ip-10-130-15-144:~/grafana$ docker compose ps -a
NAME                                COMMAND                                SERVICE    STATUS    PORTS
grafana-grafana-1                   "/run.sh"                              grafana    running   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
grafana-node_exporter_1-1           "/bin/node_exporter"                   node_exporter_1  running   0.0.0.0:9100->9100/tcp, :::9100->9100/tcp
grafana-node_exporter_2-1           "/bin/node_exporter"                   node_exporter_2  running   0.0.0.0:9101->9100/tcp, :::9101->9100/tcp
grafana-prometheus-1                "/bin/prometheus --s..."             prometheus    running   0.0.0.0:9090->9090/tcp, :::9090->9090/tcp
```

The screenshot shows the Prometheus web interface. The 'Targets' section is active, displaying a table with two targets. Both targets are in the 'UP' state. The first target is 'http://node\_exporter\_2:9100/metrics' with a last scrape of 1.738s ago and a duration of 21.886ms. The second target is 'http://node\_exporter\_1:9100/metrics' with a last scrape of 4.365s ago and a duration of 19.457ms. The interface includes a search bar and buttons for 'All', 'Unhealthy', and 'Collapse All'.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://node_exporter_2:9100/metrics	UP	instance="node_exporter_2:9100" job="node"	1.738s ago	21.886ms	
http://node_exporter_1:9100/metrics	UP	instance="node_exporter_1:9100" job="node"	4.365s ago	19.457ms	

The screenshot shows the Grafana login page. It features the Grafana logo at the top, followed by the text 'Welcome to Grafana'. Below this, there are input fields for 'Email or username' and 'Password'. A 'Log in' button is positioned below the password field. A link for 'Forgot your password?' is located at the bottom right of the login form.



## 2. Create alert CPU dan notifikasi alert Buat contact points telegram\_alert

**Contact points**  
Choose how to notify your contact points when an alert instance fires

Update contact point

Name \*  
telegram\_alert

Integration  
Telegram

BOT API Token  
Configured [Clear](#)

Chat ID  
Integer Telegram Chat Identifier  
254541939

> Optional Telegram settings

> Notification settings

[+ Add contact point integration](#)

# Alert rules

## 2. Define query and alert condition

Define query and alert condition [Need help?](#)

Prometheus Options an hour, Min. Interval = 15s Set as alert condition

Kick start your query Explain

Run queries Builder Code

Metrics browser `100 - (avg(irate(node_cpu_seconds_total{mode='idle'}[5m])) by (instance)) * 100`

Options Legend: {{ instance }} Format: Time series Step: Type: Instant

Graph Lines Bars Points Stacked lines Stacked bars

node\_exporter\_1:9100 node\_exporter\_2:9100

Add query

Rule type Select where the alert rule will be managed. [Need help?](#)

Grafana-managed Data source-managed

Based on the selected data sources this alert rule will be Grafana-managed.

## Expressions

Manipulate data returned from queries with math and other operations.

### B Reduce

Set as alert condition

Takes one or more time series returned from a query or an expression and turns each series into a single number.

Input A

Function Last Mode Drop Non-numeric V...

{instance="node_exporter_1:9100"}	1
{instance="node_exporter_2:9100"}	0.9

2 series

Add expression Preview

### C Threshold

Alert condition

Takes one or more time series returned from a query or an expression and checks if any of the series match the threshold condition.

Input B

IS ABOVE 80

Custom recovery threshold

{instance="node_exporter_1:9100"}	0	Normal
{instance="node_exporter_2:9100"}	0	Normal

2 series: 0 firing, 2 normal

## 3. Set evaluation behavior

Define how the alert rule is evaluated. [Need help?](#)

Folder Select a folder to store your rule.

alert or + New folder

Evaluation group and interval Define how often the alert rule is evaluated.

alert or + New evaluation group

All rules in the selected group are evaluated every 1m.

Pending period Period the threshold condition must be met to trigger the alert. Selecting "None" triggers the alert immediately once the condition is met.

1m None 1m 2m 3m 4m 5m

> Configure no data and error handling

## 4. Configure labels and notifications

Select who should receive a notification when an alert rule fires.

### Labels

Add labels to your rule for searching, silencing, or routing to a notification policy. [Need help?](#)


No labels selected [+ Add labels](#)

### Notifications

Select who should receive a notification when an alert rule fires.

[Select contact point](#) [Use notification policy](#)

Notifications for firing alerts are routed to a selected contact point. [Need help?](#)

Alertmanager:  grafana

#### Contact point

telegram\_alert [View or create contact points](#)

 Telegram

Muting, grouping and timings (optional) [v](#)

Home > Alerting > Alert rules

Grafana

[Export rules](#)

alert > alert

1 normal | 1m | [🔍](#) | [🔗](#) | [🗑️](#)

State	Name	Health	Summary	Next evaluation	Actions
Normal	CPU Usage	ok		in a minute	<a href="#">🔍</a> <a href="#">✎</a> <a href="#">More</a>

[Go to dashboard](#) [Go to panel](#) [Show state history](#)

Evaluate Every 1m

Pending period 1m

Last evaluation a few seconds ago

Evaluation time 0s

Dashboard UID [prod-node-monitoring](#)

Panel ID [1](#)

Instances

2 normal

State	Labels	Created
Normal	instance node_exporter_1:9100 +4 common labels	2024-09-13 18:44:30
Normal	instance node_exporter_2:9100 +4 common labels	2024-09-13 18:44:30


## Test alert ke telegram

### Test contact point

#### 5. Add annotations

Add more context in your notification messages. [Need help?](#)

**Summary (optional)**  
Short summary of what happened and why.

Test Alert CPU 

**Description (optional)**  
Description of what the alert rule does.

Enter a description...


**Runbook URL (optional)**  
Webpage where you keep your runbook for the alert.

Please open <http://testCPU.only>

[+ Add custom annotation](#) [Link dashboard and panel](#)

#### Labels

Add labels to your rule for searching, silencing, or routing to a notification policy. [Need help?](#)

key = value 

[+ Add more](#)

[Send test notification](#)

