DAY 6:

Prometheus

Prometheus is an open-source system monitoring and alerting toolkit originally built at
SoundCloud. It is now a standalone open source project . Prometheus joined the Cloud Native
Computing Foundation in 2016 as the second hosted project, after Kubernetes.

Prometheus Architecture

Prometheus Server – Collects and stores metrics.

Pushgateway – Receives metrics from short-lived jobs.

Exporters – Agents that expose metrics (e.g., Node Exporter for system stats).

Alertmanager – Handles alerts based on defined rules

Grafana (Optional) – For visualization.

Common Prometheus Commands

sh
CopyEdit
```
prometheus --config.file=prometheus.yml
curl http://localhost:9090/metrics
promtool check config prometheus.yml
promtool query instant up
```

Common Prometheus Use Cases

- Monitoring Kubernetes clusters
- Tracking system health (CPU, RAM, disk, network)
- Alerting on performance issues
- Logging API response times
- Monitoring microservices

Features

1. a multi-dimensional data model with time series data identified by metric name and

key/value pairs

2. PromQL, a flexible query language to leverage this dimensionality

3. no reliance on distributed storage; single server nodes are autonomous

4. time series collection happens via a pull model over HTTP

5. pushing time series is supported via an intermediary gateway

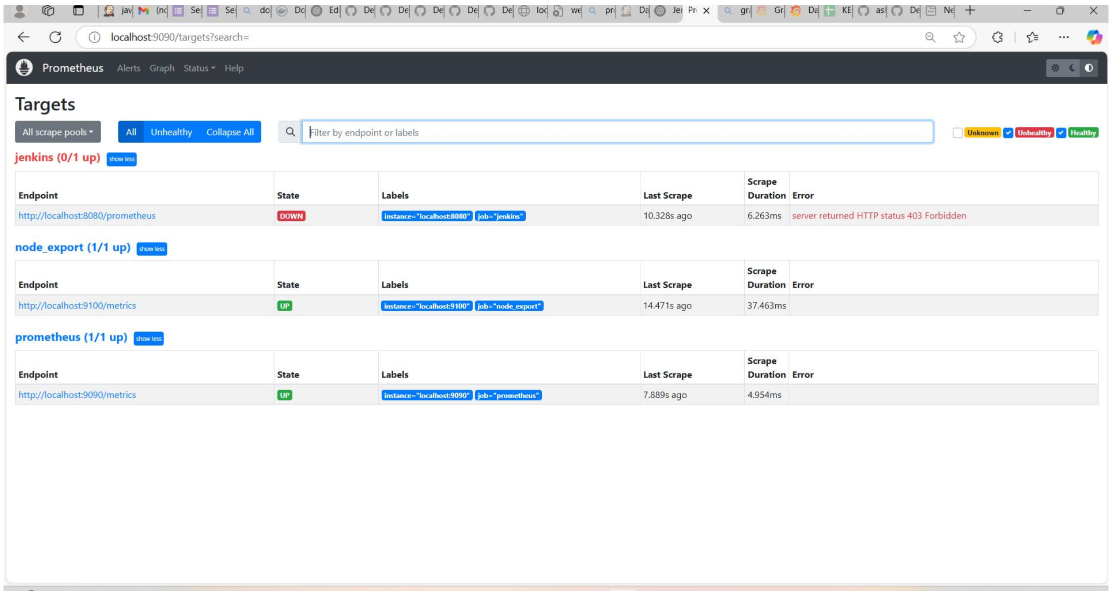
6. targets are discovered via service discovery or static configuration

7. multiple modes of graphing and dashboarding support

PROMETHEUS INSTALLATION:

```
sudo useradd \
   --system \
   --no-create-home \
      --shell /bin/false prometheus
   wget
https://github.com/prometheus/prometheus/releases/download/v2.47.1/promethe
us-2.47.1.linux-amd64.tar.gz
   tar -xvf prometheus-2.47.1.linux-amd64.tar.gz
   sudo mkdir -p /data /etc/prometheus
   cd prometheus-2.47.1.linux-amd64/
   sudo mv prometheus promtool /usr/local/bin/
   sudo mv consoles/ console_libraries/ /etc/prometheus/
   sudo mv prometheus.yml /etc/prometheus/prometheus.yml
   sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
   [12:00 PM, 3/22/2025] +91 90928 13114: cd
   rm -rf prometheus-2.47.1.linux-amd64.tar.gz
   prometheus --version
   sudo vim /etc/systemd/system/prometheus.service
   [12:09 PM, 3/22/2025] +91 90928 13114: [Unit]
```

```
Description=Prometheus

Wants=network-online.target

After=network-online.target

StartLimitIntervalSec=500

StartLimitBurst=5

[Service]

User=prometheus

Group=prometheus

Type=simple

Restart=on-failure

RestartSec=5s

ExecStart=/usr/local/bin/prometheus \

  --config.file=/etc/prometheus/prometheus.yml \

  --storage.tsdb.path=/data \

  --web.console.templates=/etc/prometheus/consoles \

  --web.console.libraries=/etc/prometheus/console_libraries \

  --web.listen-address=0.0.0.0:9090 \

  --web.enable-lifecycle

[Install]

WantedBy=multi-user.target

sudo systemctl enable prometheus

sudo systemctl start prometheus

sudo systemctl status prometheus

journalctl -u prometheus -f --no-pager

sudo useradd \
      --system \
      --no-create-home \
      --shell /bin/false node_exporter
wget
https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
  tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
  sudo mv \
    node_exporter-1.6.1.linux-amd64/node_exporter \
    /usr/local/bin/
   rm -rf node_exporter*
   node_exporter --version
  sudo vim /etc/systemd/system/node_exporter.service
  Description=Node Exporter
  Wants=network-online.target
  After=network-online.target
  StartLimitIntervalSec=500
  StartLimitBurst=5
  [Service]
  User=node_exporter
  Group=node_exporter
  Type=simple
  Restart=on-failure
  RestartSec=5s
  ExecStart=/usr/local/bin/node_exporter \
      --collector.logind
  [Install]

  WantedBy=multi-user.target

  sudo systemctl enable node_exporter

  sudo systemctl start node_exporter

  sudo systemctl status node_exporter

  journalctl -u node_exporter -f --no-pager

   - job_name: 'jenkins'

     metrics_path: '/prometheus'

     static_configs:

       - targets: ['<jenkins-ip>:8080promtool check config
/etc/prometheus/prometheus.yml

   curl -X POST http://localhost:9090/-/reload

sudo apt-get install -y apt-transport-https software-properties-common

  wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

  echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list

  sudo apt-get update

  sudo apt-get -y install grafana

  sudo systemctl enable grafana-server

  sudo systemctl start grafana-server

  sudo systemctl status grafana-server
```



localhost:9090/targets?search=

Prometheus Alerts Graph Status Help

Targets

All scrape pools · All · Unhealthy · Collapse All · Filter by endpoint or labels · Unknown · Unhealthy · Healthy

jenkins (0/1 up) show less

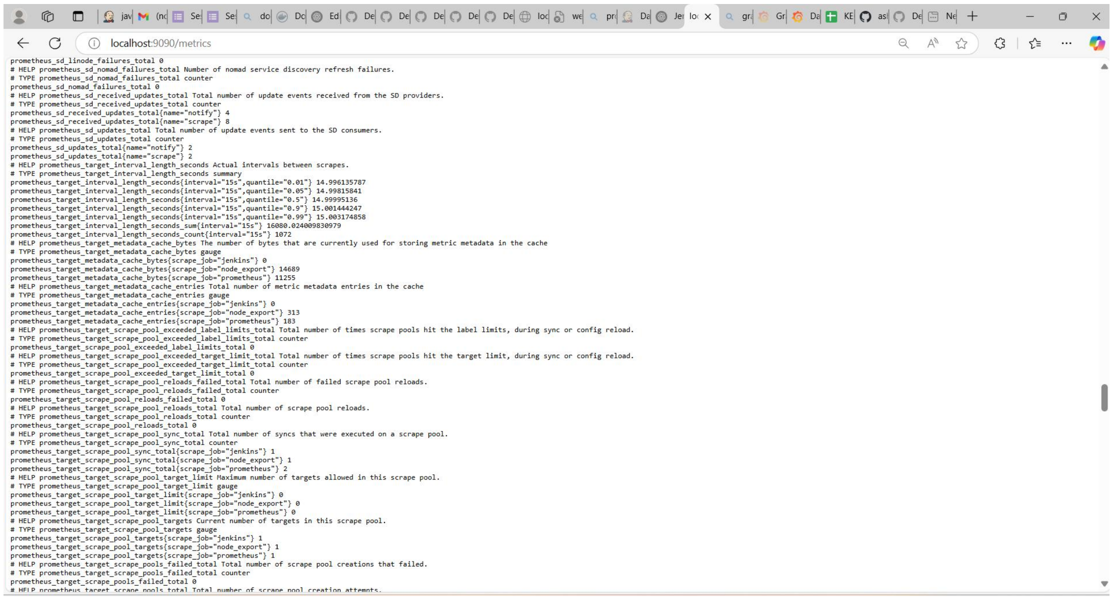
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8080/prometheus	DOWN	instance="localhost:8080" job="jenkins"	10.328s ago	6.263ms	server returned HTTP status 403 Forbidden

node_export (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_export"	14.471s ago	37.463ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	7.889s ago	4.954ms	



localhost:9090/metrics

```
prometheus_sd_linode_failures_total 0
# HELP prometheus_sd_nomad_failures_total Number of nomad service discovery refresh failures.
# TYPE prometheus_sd_nomad_failures_total counter
prometheus_sd_nomad_failures_total 0
# HELP prometheus_sd_received_updates_total Total number of update events received from the SD providers.
# TYPE prometheus_sd_received_updates_total counter
prometheus_sd_received_updates_total{name="notify"} 4
prometheus_sd_received_updates_total{name="scrape"} 8
# HELP prometheus_sd_updates_total Total number of update events sent to the SD consumers.
# TYPE prometheus_sd_updates_total counter
prometheus_sd_updates_total{name="notify"} 2
prometheus_sd_updates_total{name="scrape"} 2
# HELP prometheus_target_interval_length_seconds Actual intervals between scrapes.
# TYPE prometheus_target_interval_length_seconds summary
prometheus_target_interval_length_seconds{interval="15s",quantile="0.01"} 14.996135787
prometheus_target_interval_length_seconds{interval="15s",quantile="0.05"} 14.99815841
prometheus_target_interval_length_seconds{interval="15s",quantile="0.5"} 14.99995136
prometheus_target_interval_length_seconds{interval="15s",quantile="0.9"} 15.001444247
prometheus_target_interval_length_seconds{interval="15s",quantile="0.99"} 15.003174858
prometheus_target_interval_length_seconds_sum{interval="15s"} 16080.024009830979
prometheus_target_interval_length_seconds_count{interval="15s"} 1072
# HELP prometheus_target_metadata_cache_bytes The number of bytes that are currently used for storing metric metadata in the cache
# TYPE prometheus_target_metadata_cache_bytes gauge
prometheus_target_metadata_cache_bytes{scrape_job="jenkins"} 0
prometheus_target_metadata_cache_bytes{scrape_job="node_export"} 14689
prometheus_target_metadata_cache_bytes{scrape_job="prometheus"} 11255
# HELP prometheus_target_metadata_cache_entries Total number of metric metadata entries in the cache
# TYPE prometheus_target_metadata_cache_entries gauge
prometheus_target_metadata_cache_entries{scrape_job="jenkins"} 0
prometheus_target_metadata_cache_entries{scrape_job="node_export"} 313
prometheus_target_metadata_cache_entries{scrape_job="prometheus"} 183
# HELP prometheus_target_scrape_pool_exceeded_label_limits_total Total number of times scrape pools hit the label limits, during sync or config reload.
# TYPE prometheus_target_scrape_pool_exceeded_label_limits_total counter
prometheus_target_scrape_pool_exceeded_label_limits_total 0
# HELP prometheus_target_scrape_pool_exceeded_target_limit_total Total number of times scrape pools hit the target limit, during sync or config reload.
# TYPE prometheus_target_scrape_pool_exceeded_target_limit_total counter
prometheus_target_scrape_pool_exceeded_target_limit_total 0
# HELP prometheus_target_scrape_pool_reloads_failed_total Total number of failed scrape pool reloads.
# TYPE prometheus_target_scrape_pool_reloads_failed_total counter
prometheus_target_scrape_pool_reloads_failed_total 0
# HELP prometheus_target_scrape_pool_reloads_total Total number of scrape pool reloads.
# TYPE prometheus_target_scrape_pool_reloads_total counter
prometheus_target_scrape_pool_reloads_total 0
# HELP prometheus_target_scrape_pool_sync_total Total number of syncs that were executed on a scrape pool.
# TYPE prometheus_target_scrape_pool_sync_total counter
prometheus_target_scrape_pool_sync_total{scrape_job="jenkins"} 1
prometheus_target_scrape_pool_sync_total{scrape_job="node_export"} 1
prometheus_target_scrape_pool_sync_total{scrape_job="prometheus"} 2
# HELP prometheus_target_scrape_pool_target_limit Maximum number of targets allowed in this scrape pool.
# TYPE prometheus_target_scrape_pool_target_limit gauge
prometheus_target_scrape_pool_target_limit{scrape_job="jenkins"} 0
prometheus_target_scrape_pool_target_limit{scrape_job="node_export"} 0
prometheus_target_scrape_pool_target_limit{scrape_job="prometheus"} 0
# HELP prometheus_target_scrape_pool_targets Current number of targets in this scrape pool.
# TYPE prometheus_target_scrape_pool_targets gauge
prometheus_target_scrape_pool_targets{scrape_job="jenkins"} 1
prometheus_target_scrape_pool_targets{scrape_job="node_export"} 1
prometheus_target_scrape_pool_targets{scrape_job="prometheus"} 1
# HELP prometheus_target_scrape_pools_failed_total Total number of scrape pool creations that failed.
# TYPE prometheus_target_scrape_pools_failed_total counter
prometheus_target_scrape_pools_failed_total 0
# HELP prometheus_target_scrape_pools_total Total number of scrape pool creation attempts.
```

QUERY:

rate(node_cpu_seconds_total{mode="system"}[1m])

node_cpu_seconds_total: This metric represents the total CPU time spent in different modes (user, system, idle, etc.).

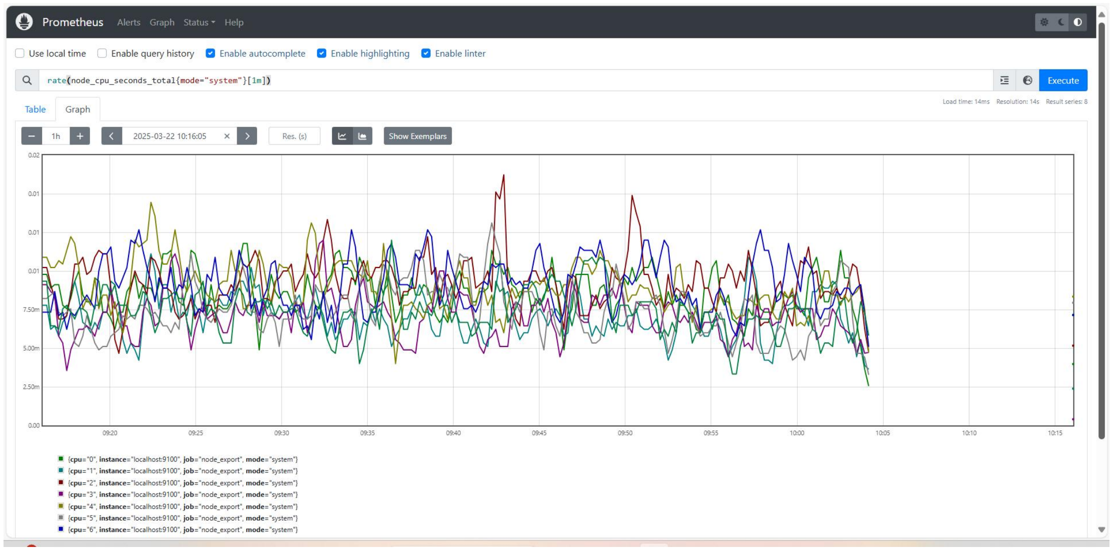
mode="system": Filters only CPU time spent in **system/kernel mode**.

rate(...[1m]): Calculates the **per-second increase** of this metric over the last **1 minute**.

What it does:

This query shows the **CPU usage in system mode per second** over the past 1 minute.

Useful for detecting high system resource consumption by kernel processes.

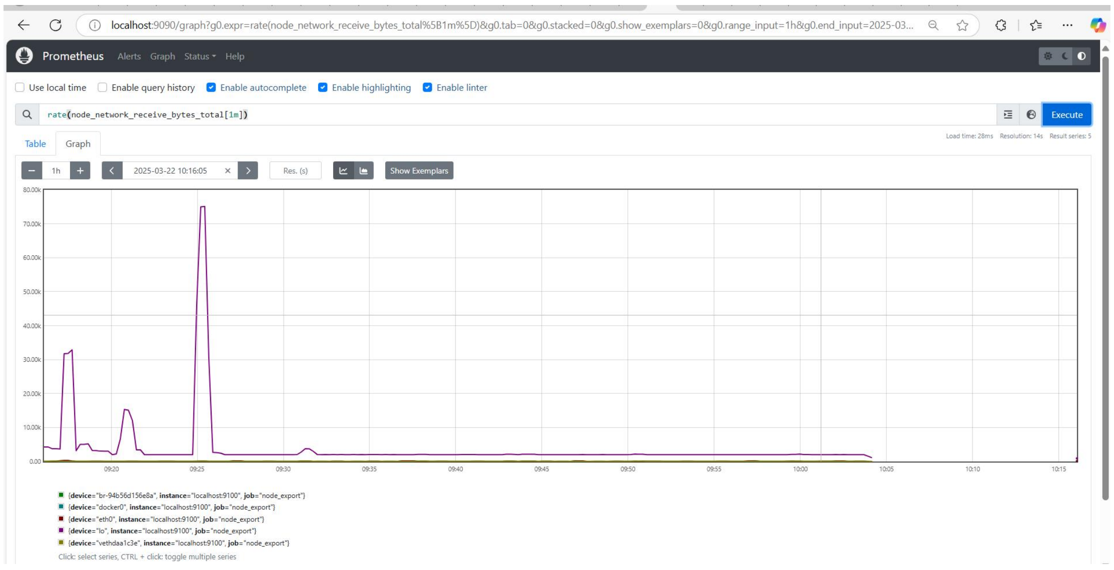


```
rate(node_cpu_seconds_total{mode="system"}[1m])
```

Load time: 14ms   Resolution: 14s   Result series: 8

- 🟩 {**cpu**="0", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- 🟦 {**cpu**="1", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- 🟥 {**cpu**="2", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- 🟪 {**cpu**="3", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- 🟨 {**cpu**="4", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- ⬜ {**cpu**="5", **instance**="localhost:9100", **job**="node_export", **mode**="system"}
- 🟦 {**cpu**="6", **instance**="localhost:9100", **job**="node_export", **mode**="system"}

rate(node_network_receive_bytes_total[1m])



localhost:9090/graph?g0.expr=rate(node_network_receive_bytes_total%5B1m%5D)&g0.tab=0&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h&g0.end_input=2025-03…

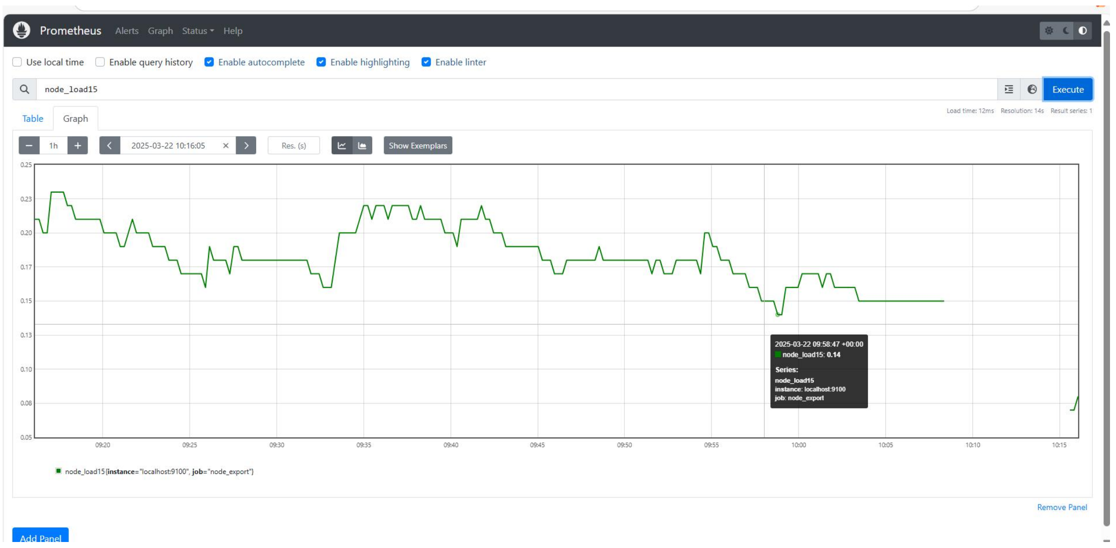
```
rate(node_network_receive_bytes_total[1m])
```

Load time: 28ms   Resolution: 14s   Result series: 5

- 🟩 {**device**="br-94b56d156e8a", **instance**="localhost:9100", **job**="node_export"}
- 🟦 {**device**="docker0", **instance**="localhost:9100", **job**="node_export"}
- 🟥 {**device**="eth0", **instance**="localhost:9100", **job**="node_export"}
- 🟪 {**device**="lo", **instance**="localhost:9100", **job**="node_export"}
- 🟨 {**device**="vethdaa1c3e", **instance**="localhost:9100", **job**="node_export"}

Click: select series, CTRL + click: toggle multiple series

node_load15



Use local time
Enable query history
[✓ Enable autocomplete]
[✓ Enable highlighting]
[✓ Enable linter]

```
node_load15
```

Load time: 12ms Resolution: 14s Result series: 1

1h 2025-03-22 10:16:05

```
2025-03-22 09:58:47 +00:00
node_load15: 0.14

Series:
node_load15
instance: localhost:9100
job: node_export
```

🟩 node_load15{**instance**="localhost:9100", **job**="node_export"}