

Prometheus Hypernode

Prometheus is a monitoring tool originally developed by and for SoundCloud back in 2012. While it has been around for quite some time, its integration into the Cloud Native Computing Foundation (CNCF) has spurred further development. As an open-source project with an active community, Prometheus is also actively supported on the Hypernode platform, something we're eager to utilise.

I'm Jonathan, a DevOps and Full Stack Web Developer at Mooore Digital. In my role, I focus heavily on DevOps and Site Reliability. Given that stability—and thus monitoring—is crucial to our services, the availability of Prometheus on Hypernode is a significant development. I'd like to share more about this.

This post will be a practical deepdive into Prometheus, explaining how to start using Prometheus for enhanced monitoring.

Content

- [About Prometheus](#)

 - [What is Prometheus?](#)

 - [Key Features of Prometheus:](#)

 - [What Were Our Goals with Prometheus?](#)

 - [Benefits](#)

- [Behind the Scenes](#)

 - [Server Metrics](#)

 - [Application Metrics](#)

 - [Notifications on Metrics](#)

 - [Hypernode Metrics](#)

- [Prometheus on Your Hypernode](#)

 - [Prometheus and Hypernode](#)

 - [Setting Up a Prometheus Node-Exporter Daemon](#)

 - [Configuring a Reverse Proxy via an Nginx Vhost](#)

 - [Bonus: Adding Your Own Metrics to Prometheus](#)

 - [Support for Implementing Prometheus](#)

- [About Mooore Digital](#)

About Prometheus

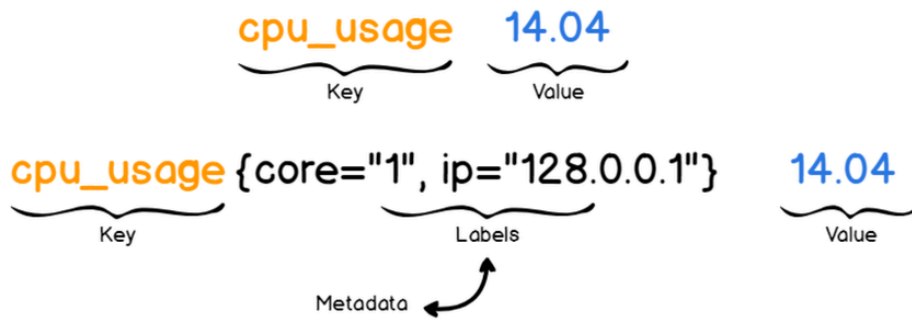
To understand the value of Prometheus, it's useful to delve a bit deeper into what this tool offers. Already familiar with Prometheus and just want implementation details? Feel free to skip ahead.

What is Prometheus?

Prometheus is an open-source technology designed to provide monitoring and alert capabilities. It can collect and store data as time series, recording information with timestamps and optional key-value pairs (labels).

In simple terms, Prometheus works by creating a small text file every minute with keys and values. The central server collects these files at the end of the minute and stores them in the database. This allows us to zoom in on minute-by-minute activity and see exactly what's happening with a system.

1 Prometheus Data Model



2 Data Model Filtering



Key Features of Prometheus:

- **Multidimensional Data Model:** Uses time series data identified by metric names and key-value pairs.
- **PromQL:** A flexible query language for the multidimensional data model.
- **No Reliance on Distributed Storage:** Each server node is autonomous.
- **Pull Model:** Prometheus actively “pulls” time series data via HTTP.
- **Pushing Time Series Data:** Possible using an intermediary gateway.
- **Service Discovery and Static Configuration:** For monitoring target discovery.
- **Visualisation:** Prometheus offers various graphs and dashboards.

What Were Our Goals with Prometheus?

Our primary aim is to actively monitor and quickly respond to potential issues with applications or servers. Tools like New Relic help us monitor at the application level and offer various features to address bottlenecks, allowing us to better track system health and act proactively to prevent problems.

We also value knowing about issues before they escalate. Combining Prometheus with Hypernode enhances our monitoring capabilities, making advanced monitoring more accessible, and allowing us to provide clients with better-performing, more stable environments.

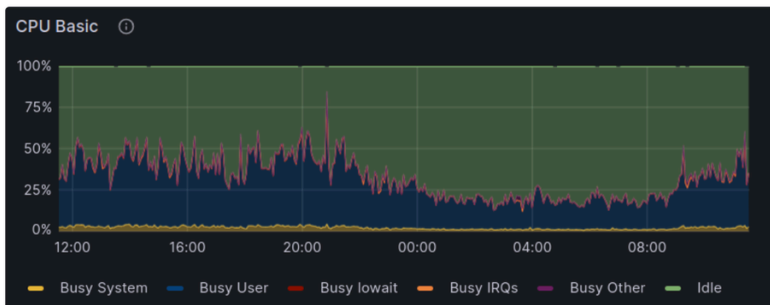
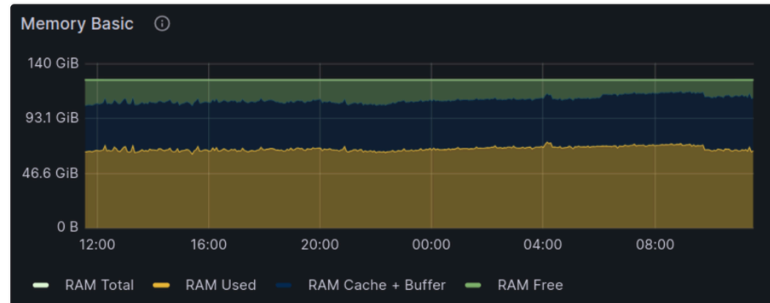
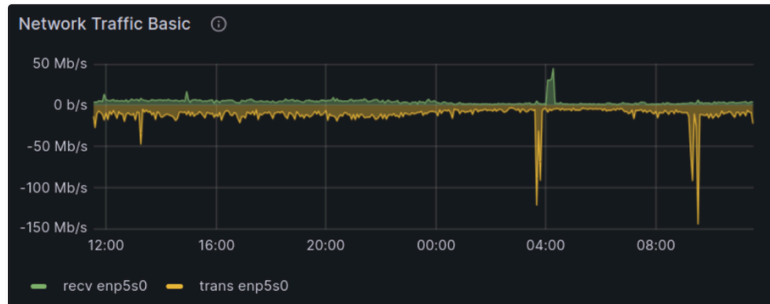
Benefits

Implementing Prometheus' data exporter on a Hypernode server offers several advantages. It enables us to perform active monitoring and respond swiftly to any application or server issues, thereby maintaining system health and acting proactively to prevent problems. This also allows us to offer clients a better-performing and more stable environment. Making advanced monitoring more accessible to all colleagues helps optimise server performance effectively.

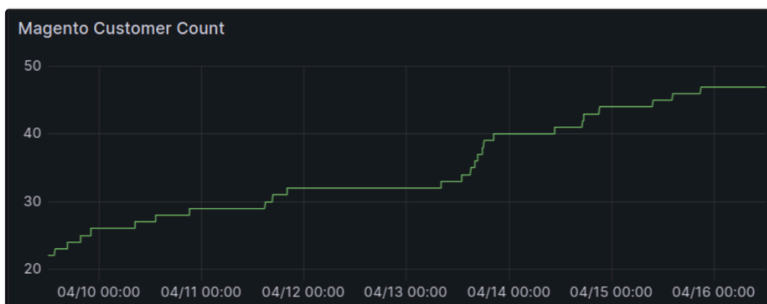
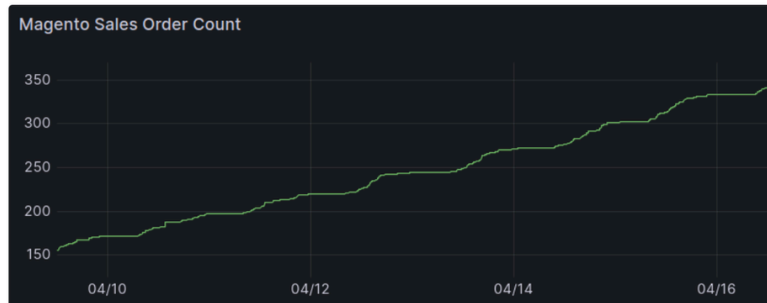
Behind the Scenes

In our practice, we monitor several key aspects and set up notifications accordingly. There's always room to expand.

Server Metrics

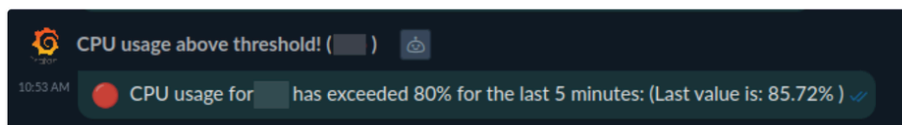
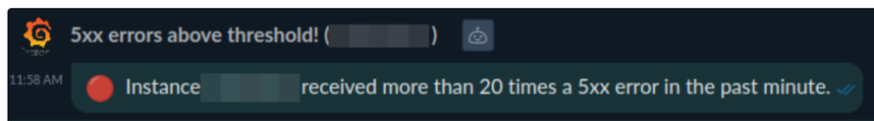


Application Metrics



Notifications on Metrics

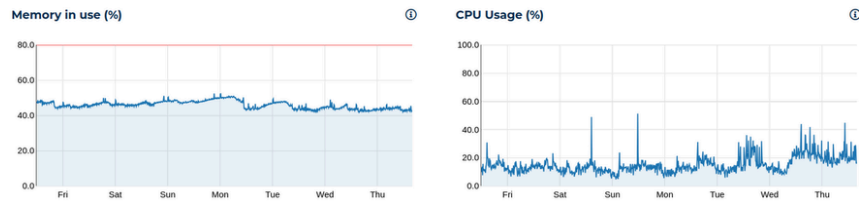
By setting thresholds and notifications on monitored metrics, we ensure stability and address anomalies, such as slow-loading webshops or unusual order patterns.



Hypernode Metrics

Hypernode also provides server state monitoring, but Prometheus offers more advanced active monitoring and alerting capabilities. Using Prometheus on Hypernode enhances our ability to monitor and alert at a higher level.

Statistics



Prometheus on Your Hypernode

IMPORTANT: THIS IS FOR TECHNICAL USERS ONLY!!!

To set up a working Prometheus node-exporter on Hypernode, follow these steps. If you have any questions, please contact us.

<https://moore.atlassian.net/wiki/external/NDZiMDM0MDQ0NDZINDgzNzhjZjFjNDI2OTIzZmYxYmE#Prometheus-op-jouw-Hypernode%5BhardBreak%5Dlet-op!-Dit-is-voor-technische-mensen-only!!!-%3Ao>

Prometheus and Hypernode

Currently, all Hypernodes run a Prometheus node-exporter (located at `/var/lib/prometheus`) with a configuration set up via a private public key pair shared across all Hypernodes and kept by the Hypernode team. However, you can run a second, less secure version of the node-exporter yourself. Feel free to contribute or leave comments for feedback and/or questions!

Before proceeding, be mindful of the security risks. We'll describe how to proxy the Prometheus node-exporter via nginx as a subdomain of the Hypernode URL, protected only by HTTP Authentication—less secure than a dedicated private key. The dedicated 'hypernode' node-export runs under its own user, and our export will run under the app user.

Setting Up a Prometheus Node-Exporter Daemon

Activate the Hypernode Supervisor to keep the daemon running:

```
1 mkdir -p /data/web/supervisor
```

Activate supervisor

```
1 hypernode-systemctl settings supervisor_enabled True
```

Wait for this process to complete by the performing the following

```
1 hypernode-log | head -n2
```

Create a Supervisor program to run our Prometheus exporter (Port 19100 is used by the Hypernode node-exporter):

```
1 echo "[program:client_prometheus]
2 command=/usr/bin/prometheus-node-exporter --web.listen-address=:29100 --collector.disable-defaults --web.disable-
3 autostart=true" > /data/web/supervisor/client-prometheus.conf
```

Make sure the supervisor program is loaded and running:

```
1 supervisorctl reread
2 supervisorctl add client_prometheus
3 supervisorctl start client_prometheus
```

Verify the process is running at <http://localhost:29100>:

```
1 curl http://localhost:29100/metrics
```

Configuring a Reverse Proxy via an Nginx Vhost

Use nginx to proxy, encrypt (HTTPS), and password-protect the node-export application:

```
1 export HYPERNODE_DOMAIN=$(cat /etc/hypernode/app.json | jq -r ".hn_fqdn")
2 export PROMETHEUS_DOMAIN="prom.$HYPERNODE_DOMAIN"
3 hypernode-manage-vhosts --type generic-php --https --force-https $PROMETHEUS_DOMAIN
4
5 rm ~/nginx/$PROMETHEUS_DOMAIN/varnish*
6 rm ~/nginx/$PROMETHEUS_DOMAIN/staging*
```

Configure nginx to proxy to the Prometheus server:

```
1 echo "location / {
2     proxy_pass http://127.0.0.1:29100;
3     proxy_set_header Host \$host;
4     proxy_set_header X-Real-IP \$remote_addr;
5     proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
6     proxy_set_header X-Forwarded-Proto \$scheme;
7 }" > ~/nginx/$PROMETHEUS_DOMAIN/public.genericphp.conf
```

Create an .htpasswd file for HTTP Prompt:

```
1 HTTP_PROMPT_USERNAME="prometheus-client"
2 HTTP_PROMPT_PASSWORD="ZEERVEILIG-pxy-npx0xdn5PCKtce"
3
4 htpasswd -bc ~/nginx/$PROMETHEUS_DOMAIN/.htpasswd $HTTP_PROMPT_USERNAME "$HTTP_PROMPT_PASSWORD"
5
6 echo "" > ~/nginx/$PROMETHEUS_DOMAIN/.htpasswd
7 sh -c "echo -n '$HTTP_PROMPT_USERNAME:' >> ~/nginx/$PROMETHEUS_DOMAIN/.htpasswd"
8 sh -c "openssl passwd -apr1 '$HTTP_PROMPT_PASSWORD' >> ~/nginx/$PROMETHEUS_DOMAIN/.htpasswd"
```

Create the server.basicauth.conf nginx configuration file:

```
1 echo "auth_basic \"Alleen geautoriseerde toegang\";
2 auth_basic_user_file /data/web/nginx/$PROMETHEUS_DOMAIN/.htpasswd;" > ~/nginx/$PROMETHEUS_DOMAIN/server.basicauth
```

Verify the exporter is running at <https://prom.YOURAPP.hypernode.io/> with HTTP Prompt enabled. When setting the export node in your Prometheus client, ensure you set the `basic_auth` property. More information can be found [here](#).

Bonus: Adding Your Own Metrics to Prometheus

The Hypernode node-exporter already adds some metrics: Web status codes, nginx errors, and various Redis information. To expand on this, create a new directory for metrics used by the client-exporter to avoid potentially breaking the Hypernode node-exporter.

Create a new directory for local Prometheus node-export metrics:

```
1 mkdir -p /data/web/.prometheus/node-export
```

Create symlinks for Hypernode metrics (updated every minute):

```
1 ln -s /var/lib/prometheus/node-exporter/* /data/web/.prometheus/node-exporter
```

Modify the path in the Supervisor process to `/data/web/.prometheus/node-exporter` in `/data/web/supervisor/client-prometheus.conf`

```
1 nano /data/web/supervisor/client-prometheus.conf
2 # Vervang --collector.textfile.directory=/var/lib/prometheus/node-exporter door --collector.textfile.directory=/d
```

Restart Supervisor

```
1 supervisorctl reload
```

Now, you can create new metrics in `~/prometheus/node-exporter`. Any new files with the `.prom` extension will be automatically included.

Support for Implementing Prometheus

We hope this post provides practical guidance to elevate your monitoring capabilities. If your development team (as an e-commerce business or agency) needs assistance implementing Prometheus, feel free to contact us. We're happy to take it off your hands or help your team through hands-on workshops [here](#). Reach out to us to use Prometheus with Hypernode.

About Moore Digital

We specialise in building robust e-commerce landscapes for distinctive brands and organisations. Our expertise spans UX design, integrations, (Magento) development, and (headless) front-ends. We are renowned for our deep knowledge of [Magento \(Hyvä\)](#) and extensive experience with Hypernode. For over 15 years, we have developed e-commerce solutions for well-known brands that sell directly to consumers and robust B2B platforms combined with Magento.

With a strong background in DevOps and CI/CD, our colleagues have profound knowledge in these areas, enabling us to help other agencies or development teams professionalise their development processes, such as with Prometheus.