

Monitoring Stack

Building a Docker-Based Monitoring Stack for Comprehensive Infrastructure Management

Monitoring is crucial for any system administrator or DevOps engineer who needs to maintain the health and performance of servers, containers, and services. A well-configured monitoring stack provides deep insights into resource usage, network performance, and other system metrics. In this blog, we'll explore how to set up a powerful Docker-based monitoring stack using Prometheus, Grafana, Node Exporter, PVE (Proxmox Virtual Environment) Exporter, and cAdvisor.

This stack will allow you to monitor physical and virtual systems, containers, and network performance efficiently. We'll go over the features, use cases, and detailed setup instructions using Docker Compose to get everything up and running smoothly.

Key Components of the Monitoring Stack

1. **Prometheus:** An open-source monitoring and alerting toolkit that collects and stores time-series data metrics. It is highly customizable and scalable, making it ideal for monitoring infrastructure of all sizes.
 - [Official Prometheus Docs](#)
2. **Grafana:** A data visualization tool that integrates seamlessly with Prometheus to provide real-time graphs, charts, and dashboards. Grafana is highly customizable and supports

alerts, making it a go-to choice for visualizing system metrics.

- [Official Grafana Docs](#)
3. **Node Exporter:** A Prometheus exporter that collects metrics from your Linux system (CPU, memory, disk, and network usage). It exposes these metrics to Prometheus for monitoring.
 - [Node Exporter GitHub](#)
 4. **PVE Exporter:** A custom exporter designed specifically for monitoring Proxmox Virtual Environments, exposing VM-specific metrics like CPU, memory, and disk usage to Prometheus.
 - [PVE Exporter GitHub](#)
 5. **cAdvisor:** A tool that provides insights into resource usage and performance characteristics of running containers. cAdvisor collects data on CPU, memory, network, and filesystem usage by containers.
 - [cAdvisor GitHub](#)

Features and Use Cases

1. Prometheus

- **Use Case:** Prometheus is the core component of the monitoring stack and is responsible for scraping metrics from various exporters and storing them. Prometheus is designed for reliability and scalability, making it suitable for monitoring everything from small servers to large-scale environments.
- **Features:**
 - Powerful query language (PromQL)
 - Time-series database built-in
 - Highly customizable and modular
 - Alerting capabilities through Alertmanager

2. Grafana

- **Use Case:** Grafana is used to visualize the metrics stored in Prometheus. It provides interactive dashboards that can be customized to monitor specific metrics across your infrastructure.
- **Features:**
 - Supports multiple data sources, not just Prometheus
 - Customizable dashboards and graphs
 - Real-time monitoring and alerting
 - User-friendly interface

3. Node Exporter

- **Use Case:** Node Exporter collects Linux system-level metrics. It provides essential information about hardware and operating system performance, making it perfect for server monitoring.
- **Features:**
 - Collects CPU, memory, disk I/O, and network statistics
 - Exposes hardware and operating system metrics to Prometheus

4. PVE Exporter

- **Use Case:** Specifically designed for Proxmox environments, PVE Exporter collects detailed information about VMs and containers running in Proxmox and exposes them to Prometheus.
- **Features:**
 - VM and container resource usage
 - Provides metrics for CPU, memory, disk, and network for each VM and container
 - Ideal for monitoring virtualized environments

5. cAdvisor

- **Use Case:** cAdvisor is designed for container monitoring. It provides insights into container resource usage and performance metrics, making it ideal for Docker-heavy environments.
- **Features:**
 - Tracks container CPU, memory, network, and disk usage
 - Supports container management platforms like Docker and Kubernetes
 - Real-time monitoring of container performance

Setting Up the Monitoring Stack with Docker Compose

Here's a step-by-step guide to setting up this monitoring stack using Docker Compose.

Prerequisites

- Docker and Docker Compose installed on your server.

- Basic knowledge of Docker and Prometheus/Grafana.

Docker Compose File

Create a `docker-compose.yml` file with the following content:

```
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus_data:/prometheus
    ports:
      - "9090:9090"

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana_data:/var/lib/grafana

  node_exporter:
    image: prom/node-exporter:latest
    container_name: node_exporter
    network_mode: host
    command:
      - '--path.procfs=/host/proc'
      - '--path.sysfs=/host/sys'
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro

  cadvisor:
    image: google/cadvisor:latest
    container_name: cadvisor
    ports:
```

```
- "8080:8080"
```

```
volumes:
```

- /:/rootfs:ro
- /var/run:/var/run:ro
- /sys:/sys:ro
- /var/lib/docker:/var/lib/docker:ro

```
pve_exporter:
```

```
image: prom/pve-exporter:latest
```

```
container_name: pve_exporter
```

```
ports:
```

- "9221:9221"

```
volumes:
```

```
prometheus_data:
```

```
grafana_data:
```

Prometheus Configuration

In the same directory as your `docker-compose.yml`, create a `prometheus.yml` file to define how Prometheus scrapes the exporters:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['localhost:8080']

  - job_name: 'pve_exporter'
```

```
static_configs:
  - targets: ['localhost:9221']
```

Bring Up the Stack

Run the following command to start the entire stack:

```
docker-compose up -d
```

This will start all the services: Prometheus, Grafana, Node Exporter, PVE Exporter, and cAdvisor.

Accessing the Services

- **Prometheus:** Visit `http://<your-server-ip>:9090`
- **Grafana:** Visit `http://<your-server-ip>:3000` (default login: admin/admin)
- **Node Exporter:** Visit `http://<your-server-ip>:9100/metrics`
- **cAdvisor:** Visit `http://<your-server-ip>:8080`
- **PVE Exporter:** Visit `http://<your-server-ip>:9221/metrics`

Basic Setup Instructions

Prometheus

1. Navigate to Prometheus' UI at `http://<your-server-ip>:9090`.
2. Verify that Prometheus is scraping metrics from all the configured exporters (Node Exporter, cAdvisor, and PVE Exporter).
3. Use PromQL to query specific metrics or monitor systems in real-time.

Grafana

1. Navigate to Grafana's UI at `http://<your-server-ip>:3000`.
2. Login with default credentials (`admin/admin`).
3. Add Prometheus as a data source:
 - Go to **Configuration > Data Sources**.
 - Select **Prometheus** and enter the Prometheus server URL (`http://prometheus:9090`).
4. Create custom dashboards to visualize CPU, memory, and container metrics.
5. Import pre-built dashboards from Grafana's community for easier monitoring.

cAdvisor and Node Exporter

- **Node Exporter** will start monitoring Linux system metrics automatically.
- **cAdvisor** provides detailed container insights; access it at `http://<your-server-ip>:8080`.

PVE Exporter

1. Ensure you have Proxmox running on your environment.
2. The PVE Exporter will gather VM and container metrics and expose them at `http://<your-server-ip>:9221/metrics`.

Conclusion

Building a Docker monitoring stack with Prometheus, Grafana, Node Exporter, PVE Exporter, and cAdvisor is a robust and scalable way to keep track of your infrastructure's performance. Whether you're monitoring physical servers, virtual machines, or Docker containers, this setup gives you a complete picture of system health and resource usage. By following the steps outlined above, you can set up and start monitoring in no time!

Revision #1

Created 2024-09-17 18:16:05 UTC by thesabear

Updated 2024-09-17 18:30:41 UTC by thesabear