Monitoring with Node Exporter, Prometheus, and Grafana [System Level Monitoring]

Monitoring your systems is crucial for maintaining their health and performance. Node Exporter, Prometheus, and Grafana are popular tools used for monitoring and visualization in the world of DevOps. This documentation will guide you through the process of setting up monitoring using these tools, including step-by-step instructions and examples.

Prerequisites

- A Linux-based system (Ubuntu, CentOS, etc.) where you want to set up monitoring.
- Basic knowledge of the Linux command line.
- Docker installed on your system (optional, for simplified setup).

Overview

- **Node Exporter**: A Prometheus exporter for system metrics, providing a wide array of metrics regarding the system's health and performance.
- **Prometheus**: A time-series database and monitoring system that collects metrics from various sources, including Node Exporter.
- **Grafana**: A visualization tool used to create dashboards and graphs based on the metrics collected by Prometheus.

Step 1: Install Node Exporter

Node Exporter collects system metrics and exports them in a format that Prometheus can understand.

Manual Installation (without Docker)

1. Download Node Exporter:

   wget https://github.com/prometheus/node_exporter/releases/download/v1.2.2/node_exporter-1.2.2.linux-amd64.tar.gz

2. Extract the downloaded tarball:

   tar xvfz node_exporter-1.2.2.linux-amd64.tar.gz

3. Move the Node Exporter binary to `/usr/local/bin`:
   sudo mv node_exporter-1.2.2.linux-amd64/node_exporter /usr/local/bin/

4. Create a systemd service file:

   sudo nano /etc/systemd/system/node_exporter.service

5. Paste the following contents into the file:

[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target
[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=default.target

6. Save and close the file.

7. Reload systemd to load the new service file:

   sudo systemctl daemon-reload

8. Start Node Exporter service:

   sudo systemctl start node_exporter

9. Enable Node Exporter to start on boot:

```
sudo systemctl enable node_exporter
```

Docker Installation (simplified)

If you prefer Docker, you can run Node Exporter as a Docker container:

docker run -d --name=node_exporter -p 9100:9100 prom/node-exporter

Step 2: Install Prometheus

Prometheus will scrape metrics from Node Exporter and store them for querying and visualization.

Docker Installation (simplified)

docker run -d --name=prometheus -p 9090:9090 -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus

Step 3: Configure Prometheus

Prometheus needs a configuration file (`prometheus.yml`) to know which targets to scrape.

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets: ['localhost:9100']
```

Replace `localhost:9100` with the IP and port of your Node Exporter if it's running on a different machine.

Step 4: Access Prometheus

- Prometheus UI is accessible at http://localhost:9090.

Step 5: Install Grafana

Grafana will be used to visualize the metrics collected by Prometheus.

Docker Installation (simplified)

docker run -d --name=grafana -p 3000:3000 grafana/grafana

Step 6: Configure Grafana

1. Access Grafana UI at `http://localhost:3000`.
2. Log in with the default credentials (admin/admin).
3. Add Prometheus as a data source:
    - Go to Configuration > Data Sources > Add data source.
    - Choose Prometheus.
    - Set URL to `http://localhost:9090`.
    - Click Save & Test.

Step 7: Create Dashboards

Now that Prometheus is collecting data and Grafana is connected, it's time to create some dashboards!

1. Click on the "+" icon on the left sidebar and choose Dashboard > Import.
2. Use Grafana's built-in dashboard repository or import custom dashboards from the community.
3. Explore metrics and create your custom dashboards as per your requirements.

Conclusion

You have now successfully set up a monitoring stack using Node Exporter, Prometheus, and Grafana. You can further customize and expand your monitoring setup by adding additional exporters, configuring alerts, and creating more advanced dashboards. Monitoring is a continuous process, so make sure to regularly review and update your monitoring setup to meet the changing needs of your systems.

Monitoring with Blackbox Exporter, Prometheus, and Grafana

Introduction

Monitoring is a critical aspect of managing any system or application. It allows you to collect, process, and visualize metrics to gain insights into the health and performance of your infrastructure. Blackbox Exporter, Prometheus, and Grafana are popular open-source tools commonly used for monitoring purposes. In this documentation, we will guide you through setting up monitoring using these tools, including detailed steps and examples.

Components Overview

- **Blackbox Exporter**: Blackbox Exporter is a tool that allows probing endpoints over HTTP, HTTPS, DNS, TCP, and ICMP protocols to collect metrics about their availability and response times.
- **Prometheus**: Prometheus is an open-source monitoring and alerting toolkit designed for reliability and scalability. It collects metrics from monitored targets by scraping metrics HTTP endpoints.
- **Grafana**: Grafana is a visualization and analytics platform that allows you to query, visualize, and alert on metrics gathered from various data sources, including Prometheus.

Setup Steps

Step 1: Install Prometheus

1. Download Prometheus from the official website: Prometheus Downloads.
2. Extract the downloaded archive to a suitable location on your server.
3. Navigate to the Prometheus directory and modify the `prometheus.yml` configuration file to define scrape configurations for your targets.

For example:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx]
    static_configs:
      - targets:
          - https://example.com
          - https://example2.com
```

Start Prometheus by running the `./prometheus` command.

Step 2: Install Blackbox Exporter

1. Download Blackbox Exporter from the official website: Blackbox Exporter Downloads.
2. Extract the downloaded archive to a suitable location on your server.
3. Navigate to the Blackbox Exporter directory and create a configuration file named `blackbox.yml`. Define the probes you want to perform, such as HTTP, HTTPS, TCP, or ICMP. Example configuration:

```
modules:
  http_2xx_example:
    prober: http
    timeout: 5s
    http:
      valid_http_versions: ["HTTP/1.1", "HTTP/2"]
      valid_status_codes: []
      method: GET
      headers:
        Host: "example.com"
      fail_if_ssl: false
      fail_if_not_ssl: false
      fail_if_matches_regexp:
        - "Could not connect to database"
        - "Permission denied"
```

```
tls_config:
  insecure_skip_verify: false
preferred_ip_protocol: "ip4"
```

4. Start Blackbox Exporter by running the `./blackbox_exporter --config.file=blackbox.yml` command.

Step 3: Install Grafana

1. Download Grafana from the official website: Grafana Downloads.
2. Install Grafana following the installation instructions for your operating system.
3. Start Grafana service by running the appropriate command for your system.

Step 4: Configure Grafana with Prometheus as a Data Source

1. Open Grafana in your web browser.
2. Log in with your credentials (default username/password is admin/admin).
3. Navigate to Configuration > Data Sources.
4. Click on "Add data source" and select Prometheus.
5. Configure the Prometheus data source by providing the URL where Prometheus is running (e.g., http://localhost:9090).
6. Save and test the data source to ensure it's properly configured.

Step 5: Create Dashboards and Alerts in Grafana

1. Navigate to the "+" icon on the left panel and select "Dashboard" to create a new dashboard.
2. Add panels to the dashboard by clicking on the "Add panel" button.
3. Configure each panel to visualize the metrics you're interested in. You can query metrics from Prometheus data source using PromQL (Prometheus Query Language).
4. Customize the dashboard layout, titles, and styles according to your preferences.
5. Once you have created the desired dashboard, you can set up alerts by configuring alert rules in Grafana. Define conditions for triggering alerts based on specific metrics and thresholds.

Example Queries and Visualizations

Querying HTTP Response Time

```
probe_duration_seconds{job="blackbox", module="http_2xx_example"}
```

Visualizing HTTP Response Time

1. Create a new panel in Grafana.
2. Choose the Prometheus data source.
3. Enter the above query in the "Query" field.
4. Select appropriate visualization type (e.g., Graph).
5. Customize visualization settings like axes, legend, and colors.

Setting Up Alerts

avg(probe_duration_seconds{job="blackbox", module="http_2xx_example"}) > 2

This alert rule triggers when the average HTTP response time exceeds 2 seconds.

Conclusion

By following the steps outlined in this documentation, you can set up monitoring using Blackbox Exporter, Prometheus, and Grafana. These tools provide a powerful combination for collecting, storing, querying, and visualizing metrics, allowing you to gain valuable insights into the performance and availability of your systems. Regularly monitor your infrastructure and adjust configurations as needed to ensure optimal performance and reliability.