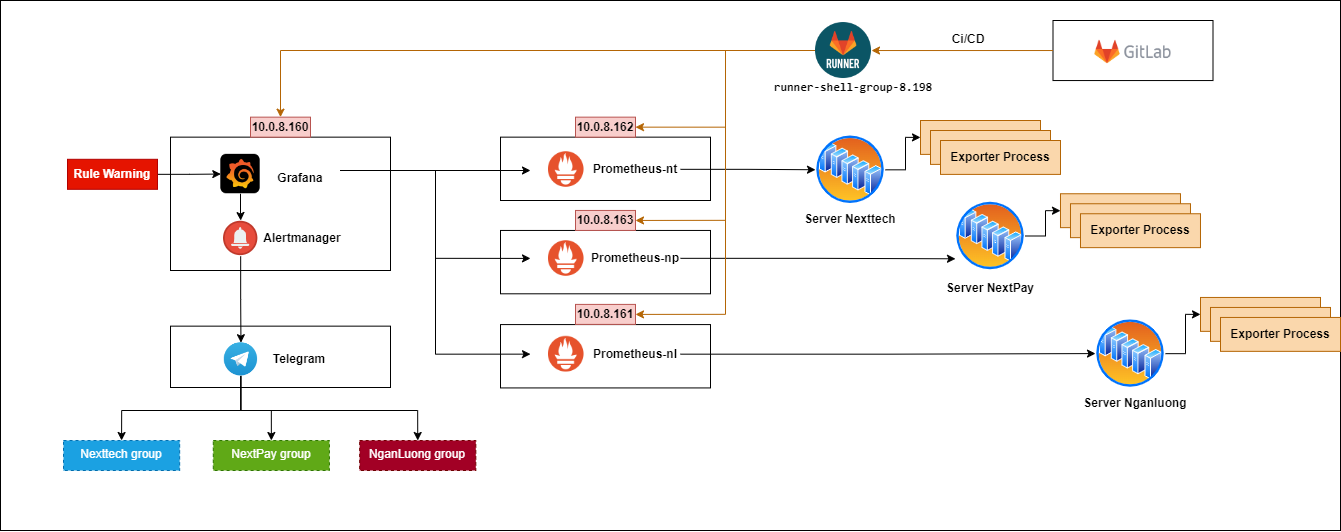
**1. Mô hình triển khai**

[](https://book.saobang.vn/uploads/images/gallery/2024-03/t2AQHgJ02MnfFba7-image-1709700175690-grafana-drawio-5.png)

Server **10.0.8.160** : Grafana + Alertmanager

* Grafana : Hiển thị biểu đồ monitor của hệ thống, quản lý rule của toàn bộ hệ thống
* Alertmanager : Gửi rule qua kênh telegram với các group telegram có trên hình

Các server prometheus đảm nhiệm chức năng monitor hệ thống của toàn dịch vụ như

* Server **10.0.8.161** : Thu thâp các dữ liệu từ tất cả hệ thống server của ngân lượng
* Server **10.0.8.162** : Thu thập các dữ liệu từ tất cả hệ thống server của Nexttech
* Server **10.0.8.163**: Thu thập các dữ liệu từ tất cả hệ thống server của Nextpay

**runner-shell-group-8.198** : Chịu trách nhiệm chạy CI/CD khi cấu hình các dịch vụ trên

**Gitlab :** Chứa toàn bộ file cấu hình của các server trên.

1

#Server 10.0.8.161 "Chỉ đặt file cấu hình docker-compose tại server", Cấu hình bên trong dịch vụ đặt tại gitlab repo

2

https://gitlab.saobang.vn/system/prometheus-nl

3

​

4

#Server 10.0.8.162 "Chỉ đặt file cấu hình docker-compose tại server", Cấu hình bên trong dịch vụ đặt tại gitlab repo

5

https://gitlab.saobang.vn/system/prometheus-nt

6

​

7

#Server 10.0.8.163 "Chỉ đặt file cấu hình docker-compose tại server", Cấu hình bên trong dịch vụ đặt tại gitlab repo

8

https://gitlab.saobang.vn/system/prometheus-np

9

​

10

#Server 10.0.8.160 "Chỉ đặt file cấu hình docker-compose tại server", Cấu hình bên trong dịch vụ đặt tại gitlab repo

11

https://gitlab.saobang.vn/system/grafana

12

https://gitlab.saobang.vn/system/alertmanager

**Exporter Process :** Được cài trên các server client như nexttech, nextpay, ngân lượng, Các client chứa tất cả các Exporter để prometheus thu thập thông tin qua cổng chỉ định điển hình như port 9100

**2. Cách thức triển khai**

* **Trên server 10.0.8.160**

Sử dụng docker-compose để triển khai dịch vụ

#vi /data/docker-compose.yml

version: '3'

​

networks:

  monitoring:

    driver: bridge

volumes:

  grafana\_data:

    driver: local

    driver\_opts:

      type: none

      o: bind

      device: /data/grafana\_data

  alertmanager\_data:

    driver: local

    driver\_opts:

      type: none

      o: bind

      device: /data/alertmanager\_data

​

services:

​

  grafana:

    image: hub.saobang.vn/monitoring/grafana:master-v.1.2.2

    container\_name: grafana

    user: root

    volumes:

      - grafana\_data:/var/lib/grafana:rw

    environment:

      - GF\_SECURITY\_ADMIN\_USER=${ADMIN\_USER:-admin}

      - GF\_SECURITY\_ADMIN\_PASSWORD=${ADMIN\_PASSWORD:-admin}

      - GF\_AUTH\_ANONYMOUS\_ENABLED=true

      - GF\_AUTH\_ANONYMOUS\_ORG\_ROLE=Editor

      - GF\_SERVER\_HTTP\_PORT=3000

      - GF\_USERS\_ALLOW\_SIGN\_UP=false

    restart: unless-stopped

    ports:

      - "3000:3000"

        #    labels:

        #      org.label-schema.group: "monitoring"

    networks:

      - monitoring

​

  alertmanager:

    image: hub.saobang.vn/monitoring/alertmanager:master-v.1.1

    container\_name: alertmanager

    user: root

    hostname: localhost

    restart: always

    ports:

      - '9093:9093'

   # Mount config file from local to Container

    volumes:

      - 'alertmanager\_data:/alertmanager/data'

   # Container start CMD to load config file

    command:

      - '--config.file=/config/alertmanager.yml'

    networks:

      - monitoring

Bật dịch vụ lên bằng cách chạy CI/CDsau:

Truy cập vào repo và đánh tag cho hệ thống trên các dịch vụ sau

https://gitlab.saobang.vn/system/alertmanager

https://gitlab.saobang.vn/system/grafana

​

Hệ thống sẽ tự chạy và cấu hình sau khi chạy pipelines

Lưu ý : Nếu là lần đầu chạy CI/CD thì cần phải chạy docker-compose bằng command sau

docker-compose up -d

trên server để có sẵn container trên hệ thống mới sử dụng chạy CI/CD được, các lần sau chỉ cần cấu hình trên gitlab và chạy CI/CD là hệ thống sẽ tự deploy cấu hình.

**Note Cấu hình Grafana:**

Repo sau chứa toàn bộ cấu hình rule của grafana, muốn sửa rule trên grafana cần phải vào đây. "Phần này tìm hiểu thêm trên trang chủ để đưa ra cấu hình phù hợp"

https://gitlab.saobang.vn/system/grafana/-/tree/main/alerting?ref\_type=heads

​

Sau Khi cấu hình xong đừng quên đánh tag và chạy CI/CD nhé, vậy là xong phần cấu hình grafana

**Note Cấu hình Alertmanager:**

Repo sau chứa toàn bộ cấu hình rule của alertmanager, muốn sửa rule trên alertmanager cần phải vào đây. "Phần này tìm hiểu thêm trên trang chủ để đưa ra cấu hình phù hợp"

https://gitlab.saobang.vn/system/alertmanager/-/blob/main/config/alertmanager.yml?ref\_type=heads

​

Sau Khi cấu hình xong đừng quên đánh tag và chạy CI/CD nhé, vậy là xong phần cấu hình alertmanager

* **Trên server 10.0.8.161**

Sử dụng docker-compose để triển khai prometheus trên server 10.0.8.161

#vi docker-compose.yml

version: '3'

​

networks:

  monitoring:

6

    driver: bridge

​

volumes:

  prometheus\_data:

    driver: local

    driver\_opts:

      type: none

      o: bind

      device: /data/prometheus\_data

​

services:

​

  prometheus:

    image: hub.saobang.vn/monitoring/prometheus-nl:master-2.3

    container\_name: prometheus

    user: root

    ports:

      - "9090:9090"

    restart: always

    volumes:

      - prometheus\_data:/prometheus:rw

    command:

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

      - '--storage.tsdb.path=/prometheus'

      - '--web.console.libraries=/etc/prometheus/console\_libraries'

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

      - '--web.enable-lifecycle'

      - '--storage.tsdb.retention.time=90d'

    networks:

      - monitoring

**Note cấu hình prometheus:**

Cấu hình prometheus thêm hosts mới cần monitor ở repo sau, nhớ cấu hình đúng định dạng như file mẫu đã có

https://gitlab.saobang.vn/system/prometheus-nl/-/tree/main/config/hosts?ref\_type=heads

​

Cấu hình prometheus thêm job mới cho prometheus cần cấu hình repo sau

https://gitlab.saobang.vn/system/prometheus-nl/-/blob/main/config/prometheus.yml?ref\_type=heads

​

Sau khi cấu hình xong host hoặc job mới, nhớ đánh tag và chạy CI/CD để deploy cấu hình mới lên hệ thống

* **Trên server 10.0.8.162**

Sử dụng docker-compose để triển khai prometheus trên server 10.0.8.162

#vi /data/docker-compose.yml

version: '3'

​

networks:

  monitoring:

    driver: bridge

​

volumes:

  prometheus\_data:

    driver: local

    driver\_opts:

      type: none

      o: bind

      device: /data/prometheus\_data

​

services:

​

  prometheus:

    image: hub.saobang.vn/monitoring/prometheus-nt:master-v.1.4

    container\_name: prometheus

    user: root

    ports:

      - "9090:9090"

    restart: always

    volumes:

      - prometheus\_data:/prometheus:rw

    command:

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

      - '--storage.tsdb.path=/prometheus'

      - '--web.console.libraries=/etc/prometheus/console\_libraries'

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

      - '--web.enable-lifecycle'

      - '--storage.tsdb.retention.time=90d'

    networks:

      - monitoring

**Note cấu hình prometheus:**

Cấu hình prometheus thêm hosts mới cần monitor ở repo sau, nhớ cấu hình đúng định dạng như file mẫu đã có

https://gitlab.saobang.vn/system/prometheus-nt/-/tree/main/config/hosts?ref\_type=heads

​

Cấu hình prometheus thêm job mới cho prometheus cần cấu hình repo sau

https://gitlab.saobang.vn/system/prometheus-nt/-/blob/main/config/prometheus.yml?ref\_type=heads

​

Sau khi cấu hình xong host hoặc job mới, nhớ đánh tag và chạy CI/CD để deploy cấu hình mới lên hệ thống

* **Trên server 10.0.8.163**

Sử dụng docker-compose để triển khai prometheus trên server 10.0.8.163

vi /data/docker-compose.yml

version: '3'

​

networks:

  monitoring:

    driver: bridge

​

volumes:

  prometheus\_data:

    driver: local

    driver\_opts:

      type: none

      o: bind

      device: /data/prometheus\_data

​

services:

​

  prometheus:

    image: hub.saobang.vn/monitoring/prometheus-np:master-v.2.7

    container\_name: prometheus

    user: root

    ports:

      - "9090:9090"

    restart: always

    volumes:

      - /data/prometheus\_data:/prometheus:rw

    command:

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

      - '--storage.tsdb.path=/prometheus'

      - '--web.console.libraries=/etc/prometheus/console\_libraries'

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

      - '--web.enable-lifecycle'

      - '--storage.tsdb.retention.time=60d'

    networks:

      - monitoring

**Note cấu hình prometheus:**

Cấu hình prometheus thêm hosts mới cần monitor ở repo sau, nhớ cấu hình đúng định dạng như file mẫu đã có

https://gitlab.saobang.vn/system/prometheus-np/-/tree/main/config/hosts?ref\_type=heads

​

Cấu hình prometheus thêm job mới cho prometheus cần cấu hình repo sau

https://gitlab.saobang.vn/system/prometheus-np/-/blob/main/config/prometheus.yml?ref\_type=heads

​

Sau khi cấu hình xong host hoặc job mới, nhớ đánh tag và chạy CI/CD để deploy cấu hình mới lên hệ thống

* **Cách triển khai Exporter trên client điển hình như Node-Exporter**

#wget https://github.com/prometheus/node\_exporter/releases/download/v1.3.1/node\_exporter-1.3.1.linux-amd64.tar.gz

#tar xvf node\_exporter-1.3.1.linux-amd64.tar.gz

#cd node\_exporter-1.3.1.linux-amd64

#cp node\_exporter /usr/local/bin

#useradd --no-create-home --shell /bin/false node\_exporter

#chown node\_exporter:node\_exporter /usr/local/bin/node\_exporter

#vi /etc/systemd/system/node\_exporter.service

​

[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

Restart=always

RestartSec=3

​

[Install]

WantedBy=multi-user.target

​

#systemctl daemon-reload

#systemctl start node\_exporter