

# UNIXサーバー構築Ⅱ

---

## 第11章 サーバー(リソース)監視

# Prometheus

---

## ■ Prometheusとは

プル型のリソース監視ツール(ソフトウェア)

## ■ 特徴

監視対象サーバーから情報を取得し、保管する。

集計など保管したデータを使用して発行できる。

アラート機能、しきい値を超えた場合、メールなどに送ることができる。

# コンポーネント

---

## ■ Prometheusのコンポーネント

Prometheusは複数のコンポーネント(Prometheus、exporterなど)で構成される。

## ■ Prometheus

監視サーバープログラム、定期的にexporterにポーリングを行うことで情報を収集する。取得したデータはprometheus内のデータベースに保存される。

---

## ■ exporter

対象サーバーで動作させるプログラム(エージェント)、テキスト形式でリソース情報を公開する。管理対象のリソースごとにexporterが用意されている。

### ① node\_exporter

CPU、メモリなどのリソース情報を取得するためのexporter

### ② nvidia\_exporter

GPUの利用状況を取得するためのexporter、nvidia\_smiで情報されるデータを取得する。

---

## ■ Alertmanager

Prometheusからのアラートデータを処理する。メールやSlackなどのへアラートを送るアラートマネージャー。

## ■ Grafana

ログなどのデータを可視化するためのツール。ただし、デフォルトで付属しているGUIでグラフ表示することができる。

# Prometheusのインストール

---

## ■事前準備

UFWなどのファイアウォールを動作させている場合、事前に許可をしておく必要がある。

**`sudo ufw allow 9090/tcp`    ...使用するポートを許可しておく**

## ■Prometheusのインストール

2つのコンポーネント(Prometheus、node-exporter)をインストールする。

**`sudo apt -y install prometheus prometheus-node-exporter`**

# Prometheusの設定

---

## ■ Prometheusの設定ファイルを変更

/etc/prometheus/prometheus.ymlファイルを変更する。

**sudo vi /etc/prometheus/prometheus.yml**

\* デフォルトの設定でも動作するため、省略可。

## ■ 有効化

**sudo systemctl enable prometheus prometheus-node-exporter**

## ■ ブラウザで確認

http://IPアドレス:9090



The screenshot shows the Prometheus web interface. At the top is a dark navigation bar with the 'Prometheus' logo and links for 'Alerts', 'Graph', 'Status', and 'Help'. Below this is a checkbox labeled 'Enable query history'. The main query editor consists of a text input field with the placeholder 'Expression (press Shift+Enter for newlines)', a blue 'Execute' button, and a dropdown menu currently showing '- insert metric at cursor -'. To the right of the editor is a 'Remove Graph' link. Below the editor are two tabs: 'Graph' (selected) and 'Console'. The 'Graph' tab contains a time range selector with left and right arrow buttons and a 'Moment' button. Below the selector is a table with two columns: 'Element' and 'Value'. The table currently displays 'no data'. At the bottom left of the interface is a blue 'Add Graph' button.



# HTTPSと認証

---

Prometheusにアクセスする時にBasic認証を有効にする。また、HTTPSでのアクセスを有効にする。

## ■事前準備

自己証明書の作成、HTTPSで使用する証明書を作成しておく。

## ■Basic認証用のユーザ名、パスワードの作成

bcrypt hash でパスワード生成する。

**htpasswd -nB ユーザ名**

\* オプション ... n(ファイルを作成せず、結果を画面に表示する)

B(bcryptでパスワードを作成)

# パスワード作成例

---

## ■ **htpasswd**コマンドで作成した例

ユーザID (ecc)、パスワード(123qwe)で設定

**sudo htpasswd -nB ecc**

New password: Re-type new password:

`ecc:$2y$05$DnS2RcdW4CoiUQYdenR1F.cAd6OHNUxEzSwxa6zW0uP0Ed`

`ZeyQcWu`

---

## ■ 証明書のコピーと権限設定

作成した証明書、鍵をPrometheusのディレクトリ(/etc/prometheus/)に移動する。

<例>

```
sudo cp /ca/server.pem /etc/prometheus/
```

```
sudo cp /ca/private/server.key /etc/prometheus/
```

```
sudo chown prometheus:prometheus /etc/prometheus/server.pem
```

```
sudo chown prometheus:prometheus /etc/prometheus/server.key
```

---

## ■ 設定ファイル(/etc/prometheus/web.yml)を新規作成

HTTPSで使用する証明書、鍵、ユーザ名、パスワードを指定する。

<例>

#証明書と鍵の指定

**tls\_server\_config:**

**cert\_file: server.pem**

... 証明書

**key\_file: server.key**

... 鍵

#ユーザ名とパスワードを指定する(htpasswdコマンドで作成したもの)

**basic\_auth\_users:**

**ecc: \$2y\$05\$DnS2RcdW4CoiUQYdenR1F.cAd6OHNUxEzSwxa6zW0uP0EdZeyQcWu**

---

## ■ web.ymlファイルを指定

作成したファイルを指定する。

```
sudo vi /etc/default/prometheus
```

5行目付近に追加する。

```
ARGS="--web.config.file=/etc/prometheus/web.yml"
```

## ■ 設定ファイル(/etc/prometheus/prometheus.yml)を編集

```
sudo vi /etc/prometheus/prometheus.yml
```

---

## ■ prometheus.ymlファイルに追記する内容

<例>-job name “prometheus”内の# scheme defaults to ‘http’.の次行に追加

Scheme: https

tls\_config:                   #証明書と鍵の指定

cert\_file: /etc/prometheus/server.pem

key\_file: /etc/prometheus/server.key

insecure\_skip\_verify: true       #自己証明書場合はtrue

basic\_auth:

#ユーザID、パスワードの指定

username: 'ecc'

password: '123qwe'

---

## ■再起動

**sudo systemctl restart prometheus**

## ■ブラウザでの確認

**https://IPアドレス:9090**

ログイン認証が聞かれる。

# 監視対象の追加

---

## ■ Prometheusで監視するサーバー追加

### ① パッケージのインストール

監視対象にnode-exporterをインストールする。

```
sudo apt -y install prometheus-node-exporter
```

### ② 設定ファイルの変更

追加された監視対象を登録する。

```
sudo vi /etc/prometheus/prometheus.yml
```



---

<例>次の項目を追記する(赤字の部分を追加)

static\_configs:

- targets: ['localhost:9090', 'ie2a99.ecccomp.ac.jp:9090']

\* グループに分けて管理することも可能

- job\_name: グループ名

static\_configs:

- targets: ['ターゲットのURL:ポート番号']

③再起動

**sudo systemctl restart prometheus**

# アラートの設定(メール)

---

## ■アラートの設定

Prometheusのアラートを設定する。アラートの設定はSlackなどのSNSやメールなどを利用することができる。

### ①Alertmanagerをインストール

アラート機能を追加するために**alertmanager**をインストールする。

```
sudo apt -y install prometheus-alertmanager
```

### ②設定ファイル作成

```
sudo mv /etc/prometheus/alertmanager.yml
```

```
/etc/prometheus/alertmanager.yml.org      ... デフォルトのファイル
```

```
sudo vi /etc/prometheus/alertmanager.yml  ... 新規作成する
```

# 設定例

---

global:

smtp\_smarthost: 'ie2a99.ecccomp.ac.jp:25' #SMTPサーバーの指定

smtp\_require\_tls: false #TLSの使用(要/不要)

smtp\_from: 'Alertmanager <ie2a99@ecccomp.ac.jp>' #送信者

# SMTP Auth を設定している場合はアカウント名とパスワードを指定

# smtp\_auth\_username: 'alertmanager'

# smtp\_auth\_password: 'password'

route:

receiver: 'email-notice' #通知先名

group\_by: ['alertname', 'Service', 'Stage', 'Role']

#アラートをグループ化する条件

---

group\_wait: 30s

# グループ化されたアラートを初回通知するまでの待ち時間

group\_interval: 5m

#新しいアラートが追加された場合の次回通知までの待ち時間

repeat\_interval: 4h

#通知済みの同内容のアラートを再送信する間隔

receivers:

- name: 'email-notice' #通知先名

email\_configs:

- to: "ie2a99@ecccomp.ac.jp" #通知先アドレス

---

### ③アラートルールの設定(新規作成)

**sudo vi /etc/prometheus/alert\_rules.yml**

<例>

groups:

- name: Instances

rules:

- alert: InstanceDown

expr: up == 0

for: 5m

---

labels:

severity: critical

annotations:

description: '{{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 5 minutes.'

summary: 'Instance {{ \$labels.instance }} down'

---

#### ④アラートの対象やルールを指定

**sudo vi /etc/prometheus/prometheus.yml**

<例>

# 17行目 : (Alertmanager 稼働ホスト):(ポート)

alerting:

  alertmanagers:

    - static\_configs:

      - targets: ['localhost:9093']

---

# 23行目 : 作成したアラートルールを追記

rule\_files:

# - "first\_rules.yml"

# - "second\_rules.yml"

- "alert\_rules.yml"

## ⑤再起動

**sudo systemctl restart prometheus prometheus-alertmanager**

**sudo systemctl enable prometheus-alertmanager**