Prometheus监控Ceph

Prometheus Server

1. 创建运行Prometheus Server进程的系统用户，并为其创建家目录/var/lib/prometheus作为数据存储目录。 ~]# useradd -r -m -d /var/lib/prometheus prometheus ~]#
2. 下载并安装Prometheus Server，以2.7.2版为例： ~]# wget https://github.com/prometheus/prometheus/releases/download/v2.7.2/prometheus-2.7.2.linux-amd64.tar.gz ~]# tar xf prometheus-2.7.2.linux-amd64.tar.gz -C /usr/local/

   ~]# ln /usr/local/prometheus-2.7.2.linux-amd64 /usr/local/prometheus
3. 创建Unit File 创建Prometheus专用的Unit File，文件路径为/usr/lib/systemd/system/prometheus.service

```
[Unit]
Description=The Prometheus 2 monitoring system and time series database.
Documentation=https://prometheus.io
After=network.target

[Service]
EnvironmentFile=-/etc/sysconfig/prometheus
User=prometheus
ExecStart=/usr/local/prometheus/prometheus \
        --storage.tsdb.path=/var/lib/prometheus \
        --config.file=/usr/local/prometheus/prometheus.yml \
        --web.listen-address=0.0.0.0:9090 \
        --web.external-url=
Restart=on-failure
StartLimitInterval=1
RestartSec=3

[Install]
WantedBy=multi-user.target
```

4. 编辑配置文件 Prometheus的主配置文件为prometheus.yml，它主要由global、rule_files、scrape_configs、alerting、remote_write和remote_read几个配置段组成：
   - global：全局配置段；
   - scrape_configs：scrape配置集合，用于定义监控的目标对象（target）的集合，以及描述如何抓取（scrape）相关指标数据的配置参数；通常，每个scrape配置对应于一个单独的作业（job），而每个targets可通过静态配置（static_configs）直接给出定义，也可基于Prometheus支持的服务发现机制进行自动配置；
   - alertmanager_configs：可由Prometheus使用的Alertmanager实例的集合，以及如何同这些Alertmanager交互的配置参数；每个Alertmanager可通过静态配置（static_configs）直接给出定义，也可基于Prometheus支持的服务发现机制进行自动配置；
   - remote_write：配置“远程写”机制，Prometheus需要将数据保存于外部的存储系统（例如InfluxDB）时定义此配置段，随后Prometheus将样本数据通过HTTP协议发送给由URL指定适配器(Adaptor);

  - remote_read：配置“远程读”机制，Prometheus将接收到的查询请求交给由URL指定适配器（Adpater）执行，Adapter将请求条件转换为远程存储服务中的查询请求，并将获取的响应数据转换为Prometheus可用的格式；

配置文件组成格式及常用的全局配置参数如下所示：

```
global:
  # How frequently to scrape targets by default.
  [ scrape_interval: <duration> | default = 1m ]

  # How long until a scrape request times out.
  [ scrape_timeout: <duration> | default = 10s ]

  # How frequently to evaluate rules.
  [ evaluation_interval: <duration> | default = 1m ]

  # The labels to add to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
    [ <labelname>: <labelvalue> ... ]

# Rule files specifies a list of globs. Rules and alerts are read from
# all matching files.
rule_files:
  [ - <filepath_glob> ... ]

# A list of scrape configurations.
scrape_configs:
  [ - <scrape_config> ... ]

# Alerting specifies settings related to the Alertmanager.
alerting:
  alert_relabel_configs:
    [ - <relabel_config> ... ]
  alertmanagers:
    [ - <alertmanager_config> ... ]

# Settings related to the remote write feature.
remote_write:
  [ - <remote_write> ... ]

# Settings related to the remote read feature.
remote_read:
  [ - <remote_read> ... ]
```

scrape配置段中，使用static_configs配置Job的语法格式：

```
    # The targets specified by the static config.
    targets:
      [ - '<host>' ]

    # Labels assigned to all metrics scraped from the targets.
    labels:
      [ <labelname>: <labelvalue> ... ]
```

使用file_sd_configs配置Job的语法格式：

```
    [
      {
        "targets": [ "<host>", ... ],
        "labels": {
          "<labelname>": "<labelvalue>", ...
        }
      },
      ...
    ]
```

配置文件中的默认配置仅支持以静态方式通过node_exporter监控Prometheus Server本机的系统指标，其配置如下所示。

```
    scrape_configs:
      - job_name: 'prometheus'

        # metrics_path defaults to '/metrics'
        # scheme defaults to 'http'.

        static_configs:
        - targets: ['localhost:9090']
```

5. 启动服务 ~]# systemctl daemon-reload ~]# systemctl start prometheus.service
6.

node_exporter

1. 安装程序包 ~]# wget https://github.com/prometheus/node_exporter/releases/download/v0.17.0/node_exporter-0.17.0.linux-amd64.tar.gz ~]# tar xf node_exporter-0.17.0.linux-amd64.tar.gz -C /usr/local/ ~]# ln -sv /usr/local/node_exporter-0.17.0.linux-amd64 /usr/local/node_exporter ~]#
2. 创建运行Prometheus Server进程的系统用户，并为其创建家目录/var/lib/prometheus作为数据存储目录。 ~]# useradd -r -m -d /var/lib/prometheus prometheus ~]#
3. 创建Unit File

文件路径为/usr/lib/systemd/system/node_exporter.service

```
 [Unit]
```

```
Description=Prometheus exporter for machine metrics, written in Go with pluggable
metric collectors.
Documentation=https://github.com/prometheus/node_exporter
After=network.target

[Service]
EnvironmentFile=-/etc/sysconfig/node_exporter
User=prometheus
ExecStart=/usr/local/node_exporter/node_exporter \
    $NODE_EXPORTER_OPTS
Restart=on-failure
StartLimitInterval=1
RestartSec=3

[Install]
WantedBy=multi-user.target
```

4. 启动服务 ~]# systemctl daemon-reload ~]# systemctl start node_exporter.service ~]# systemctl enable node_exporter.service

设定Ceph Mgr

Ceph Manager内置了众多模块，包括prometheus模块，用于直接输出Prometheus风格的指标数据。

~]# ceph mgr module enable prometheus

Prometheus模块默认监听于TCP协议的9283端口。

配置Prometheus Job

修改Prometheus的配置文件，添加与Ceph相关的Job。

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from
this config.
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['localhost:9090']
  - job_name: 'node'
    static_configs:
    - targets: ['stor03.ilinux.io:9100']
  - job_name: 'ceph'
    static_configs:
      - targets: ['stor03.ilinux.io:9283']
```

参考文档：

http://docs.ceph.com/docs/mimic/mgr/prometheus/