Promethes监控华为S5720交换机

Promethes监控华为交换机

一、部署Promethes

1.1 下载安装包

```
wget -c
https://github.com/prometheus/snmp_exporter/releases/download/v0.18.0/snmp_exporter
0.18.0.linux-amd64.tar.gz
```

1.2 解压至 /data/tools/

```
[root@snmp:/root]
[root@snmp:/root]
[root@snmp:/root]
[root@snmp:/data/tools]
[root@snmp:/data/tools]
[root@snmp:/data/tools]
[root@snmp:/data/tools/prometheus]
[root@snmp:/data/tools/prometheus]
[root@snmp:/data/tools/prometheus]
[root@snmp:/data/tools/prometheus]
```

1.3 设置环境变量

```
cat >> /etc/profile << EOF
export PROMETHEUS_HOME=/data/tools/prometheus
export PATH=$PROMETHEUS_HOME/bin:$PATH
EOF

[root@snmp:/root]
```

1.4 创建prometheus.service的system unit文件

```
cat > /usr/lib/systemd/system/prometheus.service << EOF
[Unit]
Description=Prometheus
Documentation=https://prometheus.io/
After=network.target

[Service]
Type=simple
User=prometheus
ExecStart=/data/tools/prometheus/bin/prometheus \
--config.file=/data/tools/prometheus/config/prometheus.yml \
--storage.tsdb.path=/data/tools/prometheus/data
Restart=on-failure

[Install]
WantedBy=multi-user.target

EOF
```

1.5 配置prometheus.yml文件

```
vim /data/tools/prometheus/config/prometheus.yml
#末尾处追加如下内容(此处暂时可以省略)
      labels:
        instance: prometheus
```

1.6 启动prometheus服务

```
systemctl daemon-reload
systemctl enable prometheus.service
systemctl start prometheus.service
systemctl status prometheus.service
```

1.7 检查运行情况

```
[root@snmp:/root]
prometh+  48286      1  0 14:20 ?        00:00:00
/data/tools/prometheus/bin/prometheus --
config.file=/data/tools/prometheus/config/prometheus.yml --
storage.tsdb.path=/data/tools/prometheus/data
root      48298   1703  0 14:23 pts/0    00:00:00 grep --color=auto prometheus
[root@snmp:/root]
tcp        0      0 127.0.0.1:38238         127.0.0.1:9090          ESTABLISHED
48286/prometheus
tcp6       0      0 :::9090                 :::*                    LISTEN
48286/prometheus
tcp6       0      0 127.0.0.1:9090          127.0.0.1:38238         ESTABLISHED
48286/prometheus
tcp6       0      0 ::1:9090                ::1:33128               ESTABLISHED
48286/prometheus
tcp6       0      0 ::1:33128               ::1:9090                ESTABLISHED
48286/prometheus
unix  3      [ ]         STREAM     CONNECTED     81862    48286/prometheus
```

1.8 验证

> 关闭防火墙：systemctl stop firewalld
>
> systemctl disable firewalld
>
> 登陆网站：
> http://192.168.10.30:9090/targets
> 可以发现Prometheus Endpoint已UP状态

二、部署snmp_exporter

2.1 下载安装包

```
 wget -c
https://github.com/prometheus/snmp_exporter/releases/download/v0.18.0/snmp_exporter
0.18.0.linux-amd64.tar.gz
```

2.2 解压至 /data/tools/

```
[root@snmp:/root]
[root@snmp:/root]
[root@snmp:/data/tools]
```

2.3 配置snmp_exporter

snmp_exporter的配置文件需要自己通过SNMP Exporter Config Generator 项目编译生成

由于Prometheus使用go语言开发的，所以自己编译生成snmp_exporter的配置文件需要go环境

2.4 安装go并配置环境变量

```
cd /root
tar -xzf go1.13.3.linux-amd64.tar.gz -C /usr/local/
echo 'export PATH=$PATH:/usr/local/go/bin' >> $HOME/.profile
source $HOME/.profile
go version
```

2.5 go环境安装以后，构建snmp exporter config Generator

```
[root@snmp:/root]
[root@snmp:/root]
//使用GOPROXY 或直接上github
//export GOPROXY=http://go.****iot**.com:8081
[root@snmp:/root]
[root@snmp:/root]
[root@snmp:/root]
[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/go-kit/kit v0.9.0
go: downloading github.com/prometheus/common v0.9.1
go: downloading gopkg.in/alecthomas/kingpin.v2 v2.2.6
go: downloading github.com/soniah/gosnmp v1.23.1-0.20200214014533-6d3944030084
go: extracting gopkg.in/alecthomas/kingpin.v2 v2.2.6
go: downloading github.com/alecthomas/units v0.0.0-20190924025748-f65c72e2690d
go: downloading github.com/alecthomas/template v0.0.0-20190718012654-fb15b899a751
go: extracting gopkg.in/yaml.v2 v2.2.8
go: extracting github.com/alecthomas/units v0.0.0-20190924025748-f65c72e2690d
go: extracting github.com/alecthomas/template v0.0.0-20190718012654-fb15b899a751
go: extracting github.com/soniah/gosnmp v1.23.1-0.20200214014533-6d3944030084
go: extracting github.com/prometheus/common v0.9.1
go: extracting github.com/go-kit/kit v0.9.0
go: downloading github.com/pkg/errors v0.9.1
go: downloading github.com/go-logfmt/logfmt v0.5.0
go: extracting github.com/pkg/errors v0.9.1
go: extracting github.com/go-logfmt/logfmt v0.5.0
go: finding github.com/go-kit/kit v0.9.0
go: finding github.com/prometheus/common v0.9.1
go: finding gopkg.in/alecthomas/kingpin.v2 v2.2.6
go: finding gopkg.in/yaml.v2 v2.2.8
go: finding github.com/soniah/gosnmp v1.23.1-0.20200214014533-6d3944030084
go: finding github.com/pkg/errors v0.9.1
go: finding github.com/go-logfmt/logfmt v0.5.0
go: finding github.com/alecthomas/template v0.0.0-20190718012654-fb15b899a751
go: finding github.com/alecthomas/units v0.0.0-20190924025748-f65c72e2690d

[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
```

2.6 编辑 generator.yml

```
[root@snmp:/root/go/../generator]

modules:

  if_mib:
    walk:
      - 1.3.6.1.2.1.2
      - sysUpTime
      - 1.3.6.1.2.1.31.1.1.1.6.5
    version: 2
    auth:
      community: testsnmp**
    lookups:
      - source_indexes: [ifIndex]
        lookup: ifAlias
      - source_indexes: [ifIndex]

        lookup: 1.3.6.1.2.1.2.2.1.2
      - source_indexes: [ifIndex]

        lookup: 1.3.6.1.2.1.31.1.1.1.1
    overrides:
      ifAlias:
        ignore: true
      ifDescr:
        ignore: true
      ifName:
        ignore: true
      ifType:
        type: EnumAsInfo


[root@snmp:/root/go/../generator]
[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
level=info ts=2020-06-03T02:38:30.388Z caller=net_snmp.go:142 msg="Loading MIBs"
from=mibs
level=info ts=2020-06-03T02:38:30.512Z caller=main.go:52 msg="Generating config
for module" module=if_mib
level=info ts=2020-06-03T02:38:30.737Z caller=main.go:67 msg="Generated metrics"
module=if_mib metrics=24
level=info ts=2020-06-03T02:38:30.748Z caller=main.go:92 msg="Config written"
file=/root/go/src/github.com/prometheus/snmp_exporter/generator/snmp.yml


[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
config.go  Dockerfile  FORMAT.md  generator  generator.yml  generator.yml.bak
main.go  Makefile  mibs  net_snmp.go  README.md  snmp.yml  tree.go  tree_test.go

[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
cp: overwrite ‘/data/tools/snmp_exporter/snmp.yml’? y
```

2.7 验证并测试snmp_exporter服务

```
[root@snmp:/root/go/src/github.com/prometheus/snmp_exporter/generator]
[root@snmp:/data/tools/snmp_exporter]
LICENSE  NOTICE  snmp_exporter  snmp.yml
[root@snmp:/data/tools/snmp_exporter]
level=info ts=2020-06-03T02:45:56.193Z caller=main.go:149 msg="Starting
snmp_exporter" version="(version=0.18.0, branch=HEAD,
revision=9a2ff257dd2e8cdb2a4c88b18df668e2008c2cd6)"
level=info ts=2020-06-03T02:45:56.193Z caller=main.go:150 build_context="
(go=go1.14.3, user=root@84ec61d89273, date=20200526-08:26:47)"
level=info ts=2020-06-03T02:45:56.196Z caller=main.go:243 msg="Listening on
address" address=:9116
```

> 登陆网址
> 浏览器访问 http://192.168.10.30:9116
> 在 Target 输入框中填入交换机IP地址，点击 submit 按钮，出现数据，在snmp_exporter 配置成功。

2.8 创建snmp_exporter的system unit文件

（设置snmp_exporter以服务方式开机启动）

```
cat > /usr/lib/systemd/system/snmp_exporter.service << EOF
[Unit]
Description=snmp_exporter
After=network.target

[Service]
ExecStart=/data/tools/snmp_exporter/snmp_exporter \
--config.file=/data/tools/snmp_exporter/snmp.yml
Restart=on-failure

[Install]
WantedBy=multi-user.target

EOF
```

2.9 启动snmp_exporter服务

```
systemctl daemon-reload
systemctl enable snmp_exporter.service
systemctl start snmp_exporter.service
systemctl status snmp_exporter.service
```

2.10 添加snmp_exporter至prometheus配置文件

```
[root@snmp:/data/tools/prometheus/config]


global:
  scrape_interval:     15s
  evaluation_interval: 15s



alerting:
  alertmanagers:
  - static_configs:
    - targets:



rule_files:




scrape_configs:

  - job_name: 'prometheus'



    static_configs:
    - targets: ['192.168.10.30:9090']
      labels:
        instance: prometheus
  - job_name: 'snmp_exporter'
    scrape_interval: 10s
    static_configs:
    - targets: ['192.168.100.13']
      labels:
        instance: snmp_exporter
    metrics_path: /snmp
    relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 192.168.10.30:9116
```

2.11 重启Prometheus并验证

```
systemctl restart prometheus.service
systemctl status prometheus.service
```

- 1
- 2

> 登陆网址
> http://192.168.10.30:9090/targets
>
> - 1
> - 2
>
> [外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-9qNLikAF-1591261962649)(截图/targetsNO01.png)]

三、部署Node_exporter

（注：这个可以不用部署，不影响监控交换机）

3.1 安装node_export及配置环境 ID 8919

```
cd /root/
tar -xzf node_exporter-1.0.0.linux-amd64.tar.gz -C /data/tools/
mv /data/tools/node_exporter-1.0.0.linux-amd64/ /data/tools/node_exporter
cd /data/tools/node_exporter/
mkdir bin
mv node_exporter bin/
chown -R prometheus:prometheus /data/tools/node_exporter
vim /etc/profile
export NODE_EXPORTER_HOME=/data/tools/node_exporter
export PATH=$NODE_EXPORTER_HOME/bin:$PATH
:x
source /etc/profile
```

3.2 创建node_exporter.service的 systemd unit 文件

```
cat > /usr/lib/systemd/system/node_exporter.service << EOF
[Unit]
Description=node_exporter
Documentation=https://prometheus.io/
After=network.target

[Service]
Type=simple
User=prometheus
ExecStart=/data/tools/node_exporter/bin/node_exporter
Restart=on-failure

[Install]
WantedBy=multi-user.target

EOF
```

3.3 启动node_exporter服务

```
systemctl daemon-reload
systemctl enable node_exporter.service
systemctl start node_exporter.service
systemctl status node_exporter.service
```

- 4

3.4 验证

> 登陆网站：
> http://192.168.10.30:9090/targets
> 可以发现Node Endpoint已UP状态
>
> - 3

3.5 添加node_exporter入prometheus配置文件

追加

```
  - job_name: node
    static_configs:
    - targets:
      - 192.168.10.30:9100
```

3.6 重启Prometheus并验证

```
systemctl restart prometheus.service
systemctl status prometheus.service
```

四、部署Grafana

4.1 下载安装包并安装

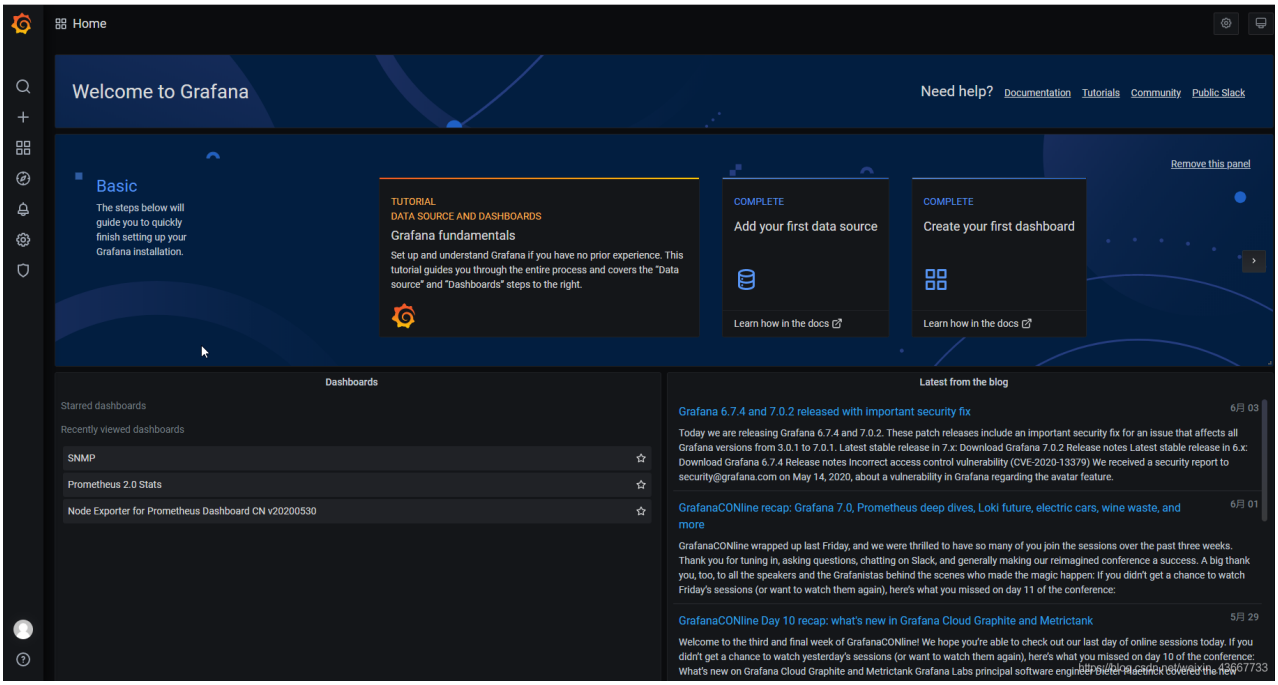
```
wget https://dl.grafana.com/oss/release/grafana-7.0.1-1.x86_64.rpm
sudo yum localinstall grafana-7.0.1-1.x86_64.rpm -y
systemctl enable grafana-server
systemctl start grafana-server
systemctl status grafana-server
```

- 4

4.2 配置Dashboard

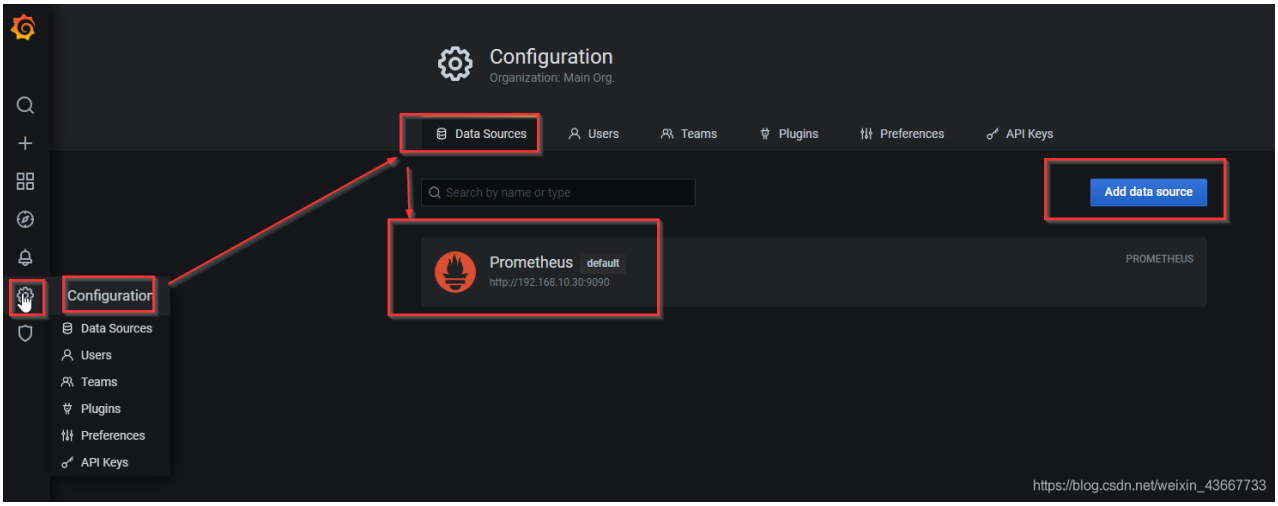
访问http://192.168.10.30:3000

4.2.1 访问Grafana页面，默认端口是3000，默认账户：admin，默认密码admin。登录成功之后修改账户密码，页面如下：

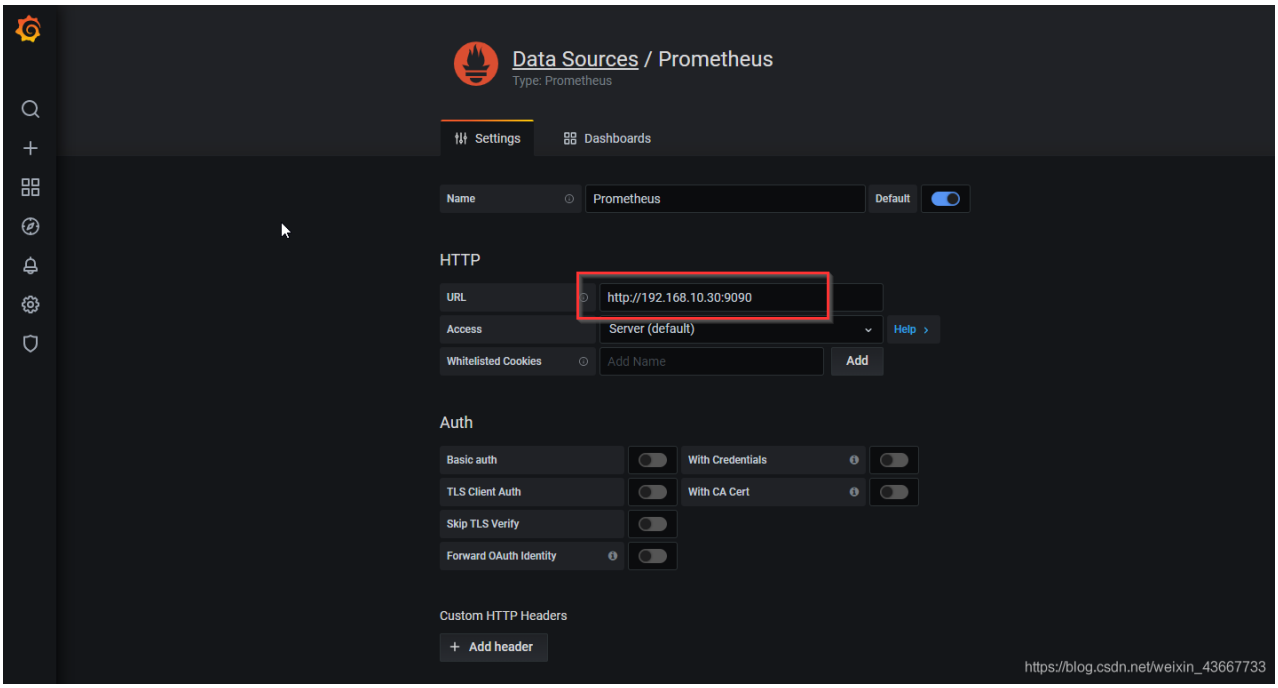


4.2.2 配置prometheus数据源

选择Data Source，如下：



在URL配置prometheus的地址

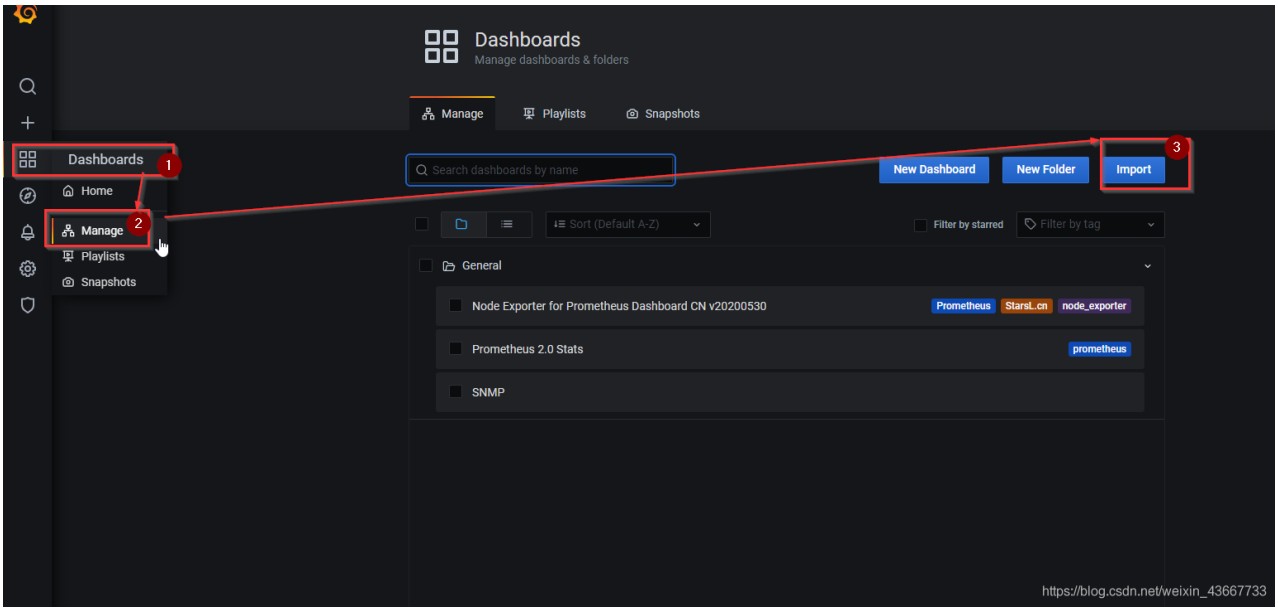


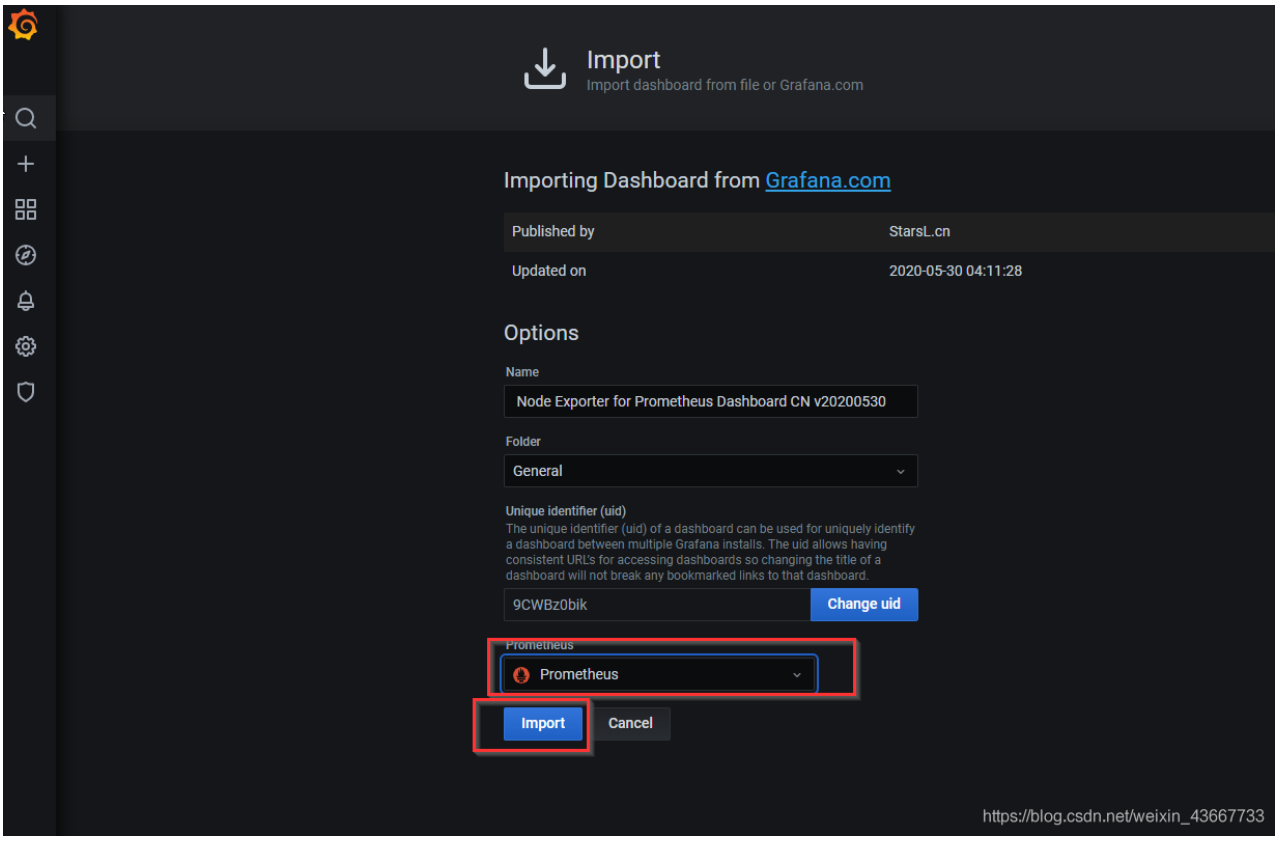
然后点击保存

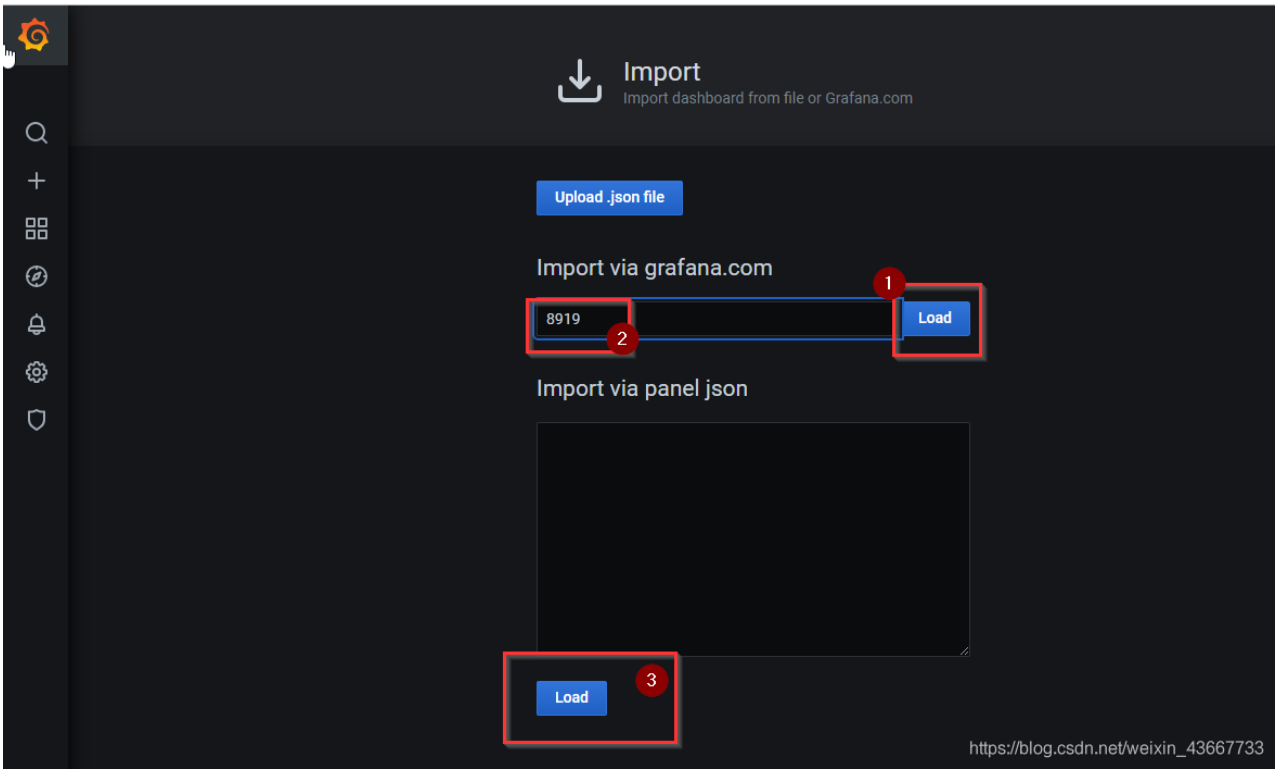
4.2.3 导入Dashboard编号

推荐dashboard id （node_exporter 8919）(emqttd 9963) (kafka:721)(process 249)

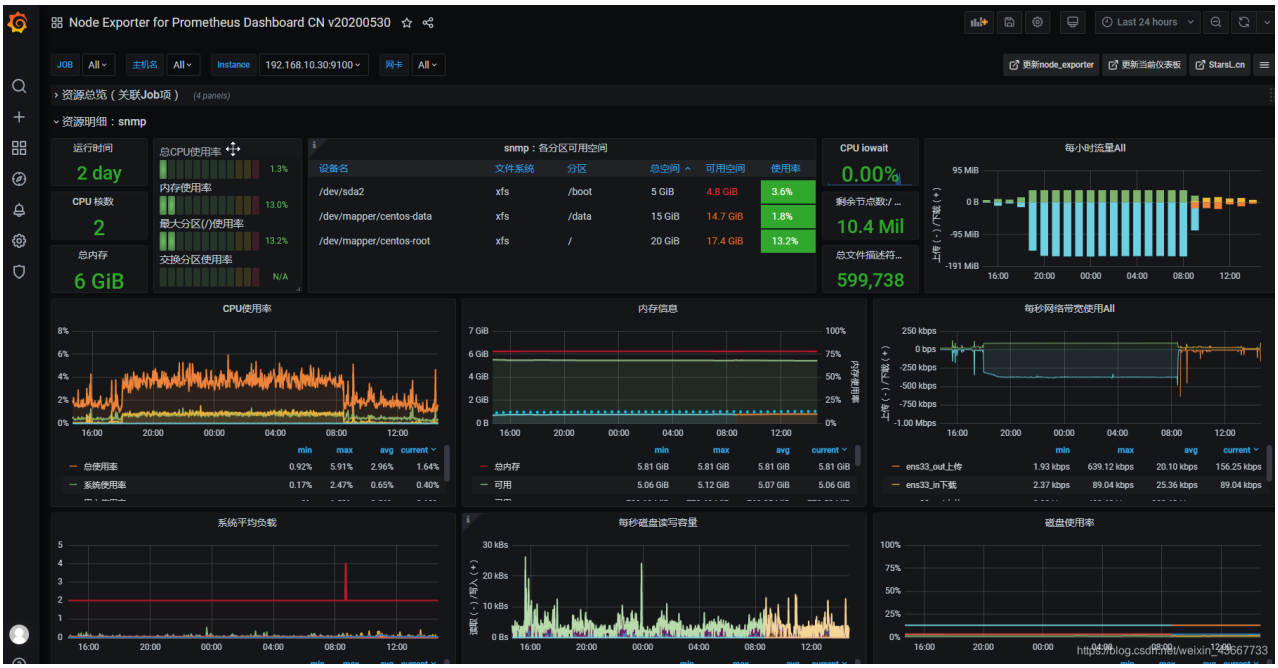
已 8919为例子




Import
Import dashboard from file or Grafana.com
Importing Dashboard from Grafana.com
Published by
StarsL.cn
Updated on
2020-05-30 04:11:28
Options
Name
Node Exporter for Prometheus Dashboard CN v20200530
Folder
General
Unique identifier (uid)
The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
9CWBz0bik
Change uid
Prometheus
Prometheus
Import
Cancel
https://blog.csdn.net/weixin_43667733


Import
Import dashboard from file or Grafana.com
Upload .json file
Import via grafana.com
1
8919
2
Load
Import via panel json
Load
3
https://blog.csdn.net/weixin_43667733

编号8919效果



交换机监控，已8570为例，进行部分修改。