

Prometheus报警AlertManager实战

Posted on December 27, 2017

标签: [kubernetes \(https://blog.qikqiak.com/tags/kubernetes/\)](https://blog.qikqiak.com/tags/kubernetes/) [prometheus \(https://blog.qikqiak.com/tags/prometheus/\)](https://blog.qikqiak.com/tags/prometheus/) [alertmanager \(https://blog.qikqiak.com/tags/alertmanager/\)](https://blog.qikqiak.com/tags/alertmanager/)

在前面一文Kubernetes使用Prometheus搭建监控平台 (<https://blog.qikqiak.com/post/kubernetes-monitor-prometheus-grafana/>)中我们知道了怎么使用 Prometheus 来搭建监控平台，也了解了 grafana 的使用。这篇文章就来说说报警系统的搭建，有人说

报警用 grafana 就行了，实际上 grafana 对报警的支持真的很弱，而 Prometheus 提供的报警系统就强大很多，今天我们的主角就是 AlertManager 。

欢迎大家加入我们的知识星球： Kubernetes 。

AlertManager 简介

安装和配置 AlertManager

配置 Prometheus 来和 AlertManager 通信

在 Prometheus 中创建报警规则

全部配置

参考文档

kubernetes

微信扫一扫加入星球

 知识星球



扫描下面的二维码(或微信搜索 k8s技术圈)关注我们的微信公众帐号，在微信公众帐号中回复 加群 即可加入到我们的 kubernetes 讨论群里面共同学习。



AlertManager 简介

Prometheus 将数据采集和报警分成了两个模块。报警规则配置在 Prometheus Servers 上，然后发送报警信息到 AlertManager，然后我们的 AlertManager 就来管理这些报警信息，包括 **silencing**、**inhibition**，聚合报警信息过后通过 email、PagerDuty、HipChat、Slack 等方式发送消息提示。

让 AlertManager 提供服务总的来说就下面3步： * 安装和配置 AlertManager * 配置 Prometheus 来和 AlertManager 通信 * 在 Prometheus 中创建报警规则

安装和配置 AlertManager

从官方文档<https://prometheus.io/docs/alerting/configuration/> (<https://prometheus.io/docs/alerting/configuration/>)中我们可以看到下载 alertmanager 二进制文件过后，通过下面的命令就可运行

```
$ ./alertmanager -config.file=simple.yml
```

可以通过 `-config.file` 来指定相应的配置文件，我们这里在 `kubernetes` 平台上面来运行我们的 AlertManager，所以我们直接用 `docker` 镜像的方式来安装，使用的镜像是：`quay.io/prometheus/alertmanager:v0.12.0`，如果你因为某些原因不能拉取该镜像，可以直接去下载 AlertManager 的源代码 (<https://github.com/prometheus/alertmanager>)，然后自己构建打包一个 `docker` 镜像也是一样的。

通过 `ConfigMap` 来指定配置文件，如下：

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: alertmanager
  namespace: kube-ops
data:
  config.yml: |-
    global:
      resolve_timeout: 5m
    route:
      receiver: webhook
      group_wait: 30s
      group_interval: 5m
      repeat_interval: 4h
      group_by: [alertname]
      routes:
        - receiver: webhook
          group_wait: 10s
          match:
            team: node
    receivers:
      - name: webhook
        webhook_configs:
          - url: 'http://apollo/hooks/dingtalk/'
            send_resolved: true
          - url: 'http://apollo/hooks/prometheus/'
            send_resolved: true
```

我们这里定义了两个 webhook，其中一个钉钉，另外一个email。关于钉钉机器人可以查看官方文档 (<https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.6LZJ6G&treeId=257&articleId=105735&docType=1>)，钉钉的报警通知很简单，在你们的钉钉群里面添加一个机器人然后就可以获取到一个带 token 的URL，然后根据上面的文档往这个URL发送数据即可。

配置 Prometheus 来和 AlertManager 通信

在我们之前的 Prometheus 的 ConfigMap 中增加 AlertManager 的配置：

```
alerting:
  alertmanagers:
    - static_configs:
      - targets: ["localhost:9093"]
```

由于我们这个地方将 Prometheus 和 AlertManager 部署在同一个 POD 中，而 AlertManager 的默认端口是9093，所以直接使用 localhost:9093 就可以互相通信了。

Containers

prometheus

Image: prom/prometheus:v2.0.0-rc.3

Environment variables: -

Commands: /bin/prometheus

Args: --config.file=/etc/prometheus/prometheus.yml
--storage.tsdb.path=/prometheus
--storage.tsdb.retention=24h

alertmanager

Image: quay.io/prometheus/alertmanager:v0.12.0

Environment variables: -

Commands: -

Args: -config.file=/etc/alertmanager/config.yml
-storage.path=/alertmanager

在 Prometheus 中创建报警规则

同样的我们还需要添加报警规则：

```
rule_files:
  - /etc/prometheus/rules.yml
```

其中 rule_files 就是用来指定报警规则的，这里我们将 rules.yml 用 ConfigMap 的形式挂载到 /etc/prometheus 目录下面即可：

```
rules.yml: |
  groups:
  - name: test-rule
    rules:
    - alert: NodeFilesystemUsage
      expr: (node_filesystem_size{device="rootfs"} - node_filesystem_free
      for: 2m
      labels:
        team: node
      annotations:
        summary: "{{ $labels.instance }}: High Filesystem usage detected"
        description: "{{ $labels.instance }}: Filesystem usage is above 80%
    - alert: NodeMemoryUsage
      expr: (node_memory_MemTotal - (node_memory_MemFree+node_memory_Buff
      for: 2m
      labels:
        team: node
      annotations:
        summary: "{{ $labels.instance }}: High Memory usage detected"
        description: "{{ $labels.instance }}: Memory usage is above 80% (cu
    - alert: NodeCPUUsage
      expr: (100 - (avg by (instance) (irate(node_cpu{job="kubernetes-nod
      for: 2m
      labels:
        team: node
      annotations:
        summary: "{{ $labels.instance }}: High CPU usage detected"
        description: "{{ $labels.instance }}: CPU usage is above 80% (curre
```

我们这里添加了3条测试的报警规则，分别是节点的文件系统、节点内存和CPU的使用量，如果大于了80%的话就触发label为 team=node 的 receiver (上面alertmanager 配置文件中)，我们可以看到上面的配置就会匹配 webhook 这个 receiver，然后将报警信息 **POST** 到我们提供的两个HOOK 接口中(一个是通知钉钉的，另外一个发发邮件的)。

一个报警信息在生命周期内有下面3中状态：

- inactive: 表示当前报警信息既不是 firing 状态也不是 pending 状态
- pending: 表示在设置的阈值时间范围内被激活了
- firing: 表示超过设置的阈值时间被激活了

Prometheus Alerts Graph Status ▾ Help

Alerts

NodeCPUUsage (0 active)

```
alert: NodeCPUUsage
expr: (100
  - (avg(irate(node_cpu{job="kubernetes-node-exporter",mode="idle"}[5m]))
  BY (instance) * 100)) > 80
for: 2m
labels:
  team: node
annotations:
  description: '{{$labels.instance}}: CPU usage is above 80% (current value is: {{
    $value }})'
  summary: '{{$labels.instance}}: High CPU usage detected'
```

NodeFilesystemUsage (0 active)

NodeMemoryUsage (0 active)

我们直接点击上面的 `expr` 会直接跳转到 `graph` 页面查询，我们在制定报警规则的时候同样可以先在 Prometheus 里面先测试我们的表达式：

Prometheus Alerts Graph Status ▾ Help

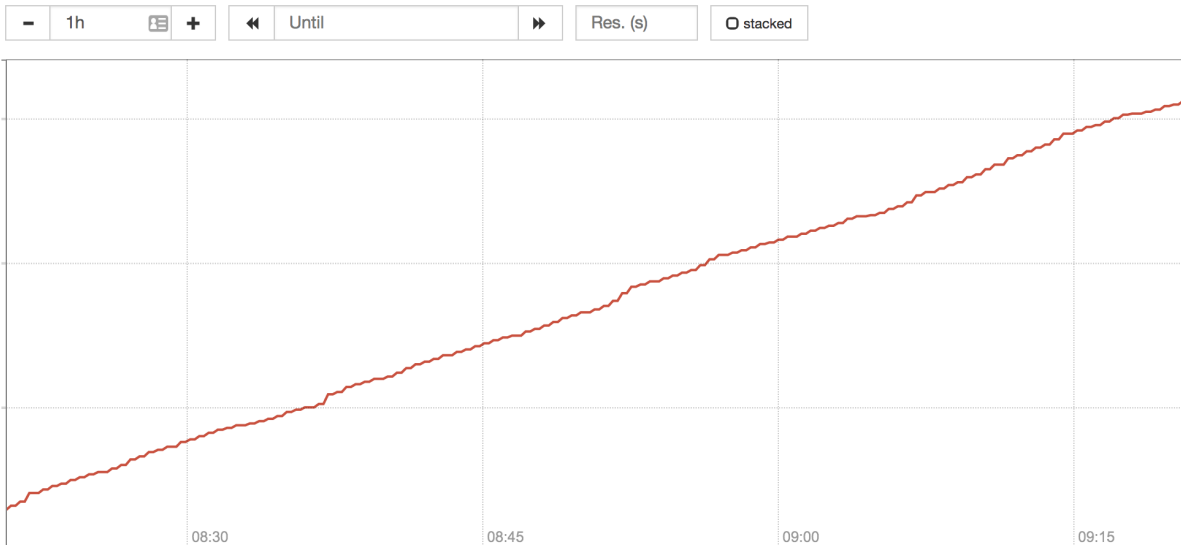
`(node_filesystem_size(device="rootfs") - node_filesystem_free(device="rootfs")) / node_filesystem_size(device="rootfs") * 100 > 70`

Load time: 131ms
Resolution: 14s
Total time series: 1

Execute

- insert metric at cursor -

Graph Console



全部配置

整体的 ConfigMap 配置文件如下 `cm.yaml`：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: kube-ops
data:
  prometheus.yml: |
    global:
      scrape_interval: 30s
      scrape_timeout: 30s

    rule_files:
      - /etc/prometheus/rules.yml

    alerting:
      alertmanagers:
        - static_configs:
            - targets: ["localhost:9093"]

    scrape_configs:
      - job_name: 'prometheus'
        static_configs:
          - targets: ['localhost:9090']

      - job_name: 'kubernetes-apisservers'
        kubernetes_sd_configs:
          - role: endpoints
        scheme: https
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
          bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
        relabel_configs:
          - source_labels: [__meta_kubernetes_namespace, __meta_kubernetes_service]
            action: keep
            regex: default;kubernetes;https

      - job_name: 'kubernetes-nodes'
        scheme: https
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
          bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
        kubernetes_sd_configs:
```



```
- role: node
relabel_configs:
- action: labelmap
  regex: __meta_kubernetes_node_label_(.+)
- target_label: __address__
  replacement: kubernetes.default.svc:443
- source_labels: [__meta_kubernetes_node_name]
  regex: (.+)
  target_label: __metrics_path__
  replacement: /api/v1/nodes/${1}/proxy/metrics

- job_name: 'kubernetes-cadvisor'
  scheme: https
  tls_config:
    ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
  bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
  kubernetes_sd_configs:
  - role: node
    relabel_configs:
    - action: labelmap
      regex: __meta_kubernetes_node_label_(.+)
    - target_label: __address__
      replacement: kubernetes.default.svc:443
    - source_labels: [__meta_kubernetes_node_name]
      regex: (.+)
      target_label: __metrics_path__
      replacement: /api/v1/nodes/${1}/proxy/metrics/cadvisor

- job_name: 'kubernetes-node-exporter'
  scheme: http
  tls_config:
    ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
  bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
  kubernetes_sd_configs:
  - role: node
    relabel_configs:
    - action: labelmap
      regex: __meta_kubernetes_node_label_(.+)
    - source_labels: [__meta_kubernetes_role]
      action: replace
      target_label: kubernetes_role
    - source_labels: [__address__]
      regex: '(.*)10250'
```

```
replacement: '${1}:31672'
```

```
target_label: __address__
```

- job_name: 'kubernetes-service-endpoints'
kubernetes_sd_configs:
 - role: endpointsrelabel_configs:
 - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_
action: keep
regex: true
 - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_
action: replace
target_label: __scheme__
regex: (https?)
 - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_
action: replace
target_label: __metrics_path__
regex: (.+)
 - source_labels: [__address__, __meta_kubernetes_service_annotation_p
action: replace
target_label: __address__
regex: ([^:]+)(?::\d+)?;(\d+)
replacement: \$1:\$2
 - action: labelmap
regex: __meta_kubernetes_service_label_(.+)
 - source_labels: [__meta_kubernetes_namespace]
action: replace
target_label: kubernetes_namespace
 - source_labels: [__meta_kubernetes_service_name]
action: replace
target_label: kubernetes_name
- job_name: 'kubernetes-services'
metrics_path: /probe
params:
 - module: [http_2xx]
kubernetes_sd_configs:
 - role: service
relabel_configs:
 - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_
action: keep

```

    regex: true
- source_labels: [__address__]
  target_label: __param_target
- target_label: __address__
  replacement: blackbox-exporter.example.com:9115
- source_labels: [__param_target]
  target_label: instance
- action: labelmap
  regex: __meta_kubernetes_service_label_(.+)
- source_labels: [__meta_kubernetes_namespace]
  target_label: kubernetes_namespace
- source_labels: [__meta_kubernetes_service_name]
  target_label: kubernetes_name

```

```

rules.yml: |
groups:
- name: test-rule
  rules:
- alert: NodeFilesystemUsage
  expr: (node_filesystem_size{device="rootfs"} - node_filesystem_free
  for: 2m
  labels:
    team: node
  annotations:
    summary: "{{ $labels.instance }}: High Filesystem usage detected"
    description: "{{ $labels.instance }}: Filesystem usage is above 80%
- alert: NodeMemoryUsage
  expr: (node_memory_MemTotal - (node_memory_MemFree+node_memory_Buff
  for: 2m
  labels:
    team: node
  annotations:
    summary: "{{ $labels.instance }}: High Memory usage detected"
    description: "{{ $labels.instance }}: Memory usage is above 80% (cu
- alert: NodeCPUUsage
  expr: (100 - (avg by (instance) (irate(node_cpu{job="kubernetes-nod
  for: 2m
  labels:
    team: node
  annotations:
    summary: "{{ $labels.instance }}: High CPU usage detected"
    description: "{{ $labels.instance }}: CPU usage is above 80% (curre

```

```
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: alertmanager
  namespace: kube-ops
data:
  config.yml: |-
    global:
      resolve_timeout: 5m
    route:
      receiver: webhook
      group_wait: 30s
      group_interval: 5m
      repeat_interval: 4h
      group_by: [alertname]
      routes:
      - receiver: webhook
        group_wait: 10s
        match:
          team: node
    receivers:
    - name: webhook
      webhook_configs:
      - url: 'http://apollo/hooks/dingtalk/'
        send_resolved: true
      - url: 'http://apollo/hooks/prometheus/'
        send_resolved: true
```

执行下面的命令即可：

```
$ kubectl apply -f cm.yaml
```

然后部署的文件 `deploy.yaml` 如下：

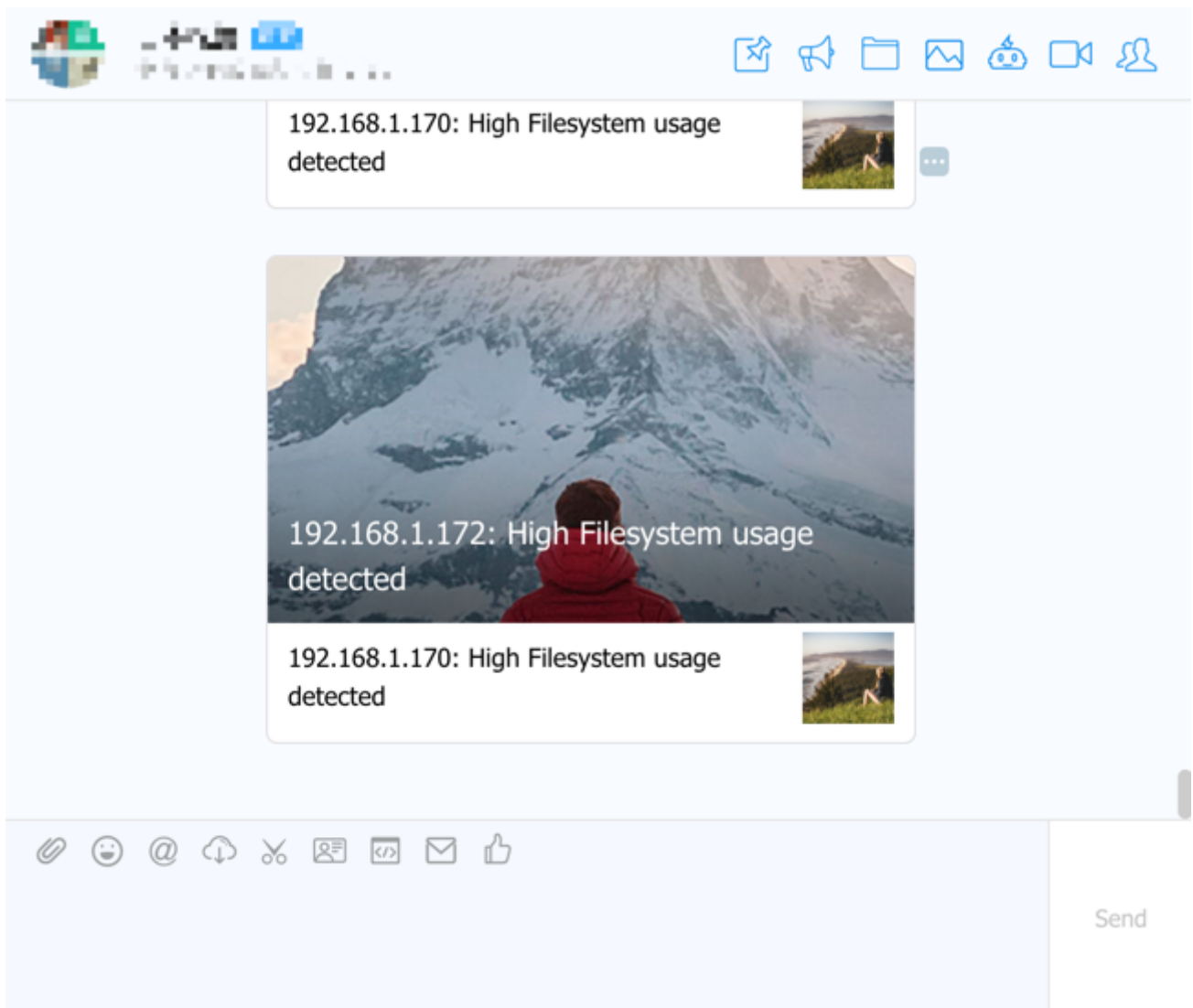
```
kind: Deployment
metadata:
  labels:
    k8s-app: prometheus
  name: prometheus
  namespace: kube-ops
spec:
  replicas: 1
  template:
    metadata:
      labels:
        k8s-app: prometheus
    spec:
      serviceAccountName: prometheus
      containers:
        - image: prom/prometheus:v2.0.0-rc.3
          name: prometheus
          command:
            - "/bin/prometheus"
          args:
            - "--config.file=/etc/prometheus/prometheus.yml"
            - "--storage.tsdb.path=/prometheus"
            - "--storage.tsdb.retention=24h"
          ports:
            - containerPort: 9090
              protocol: TCP
              name: http
          volumeMounts:
            - mountPath: "/prometheus"
              name: data
              subPath: prometheus/data
            - mountPath: "/etc/prometheus"
              name: config-volume
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
        limits:
          cpu: 200m
          memory: 1Gi
        - image: quay.io/prometheus/alertmanager:v0.12.0
          name: alertmanager
```

```
args:
- "-config.file=/etc/alertmanager/config.yml"
- "-storage.path=/alertmanager"
ports:
- containerPort: 9093
  protocol: TCP
  name: http
volumeMounts:
- name: alertmanager-config-volume
  mountPath: /etc/alertmanager
resources:
  requests:
    cpu: 50m
    memory: 50Mi
  limits:
    cpu: 200m
    memory: 200Mi
volumes:
- name: data
  emptyDir: {}
- configMap:
    name: prometheus-config
    name: config-volume
- name: alertmanager-config-volume
  configMap:
    name: alertmanager
```

执行下面的命令即可： 执行下面的命令即可：

```
$ kubectl apply -f deploy.yaml
```

最后来看一张我们钉钉收到的报警信息吧：



注意： 本文章的实现都是针对 *Prometheus2.x* 版本，1.x 版本配置会有所不同

本文中用到的 `kubernetes` 相关的 `yaml` 文件我已经归档到 `github` 上，可以在这里<https://github.com/cnych/k8s-repo/tree/master/prometheus> (<https://github.com/cnych/k8s-repo/tree/master/prometheus>)查看

参考文档

- <https://prometheus.io/docs/alerting/configuration/> (<https://prometheus.io/docs/alerting/configuration/>)
- Prometheus: understanding the delays on alerting (<https://pracucci.com/prometheus-understanding-the-delays-on-alerting.html>)

欢迎大家加入我们的知识星球： `Kubernetes` 。

kubernetes

微信扫一扫加入星球

 知识星球



扫描下面的二维码(或微信搜索 **k8s技术圈**)关注我们的微信公众帐号，在微信公众帐号中回复 **加群** 即可加入到我们的 **kubernetes** 讨论群里面共同学习。



相关文章

- Kubernetes使用Prometheus搭建监控平台 (<https://blog.qikqiak.com/post/kubernetes-monitor-prometheus-grafana/>)

- Kubernetes 下升级Prometheus2.0 (<https://blog.qikqiak.com/post/update-prometheus-2-in-kubernetes/>)
- kubernetes 日志收集方案 (<https://blog.qikqiak.com/post/kubernetes-logs-collect/>)
- kubernetes 下实现socket.io 的集群模式 (<https://blog.qikqiak.com/post/socketio-multiple-nodes-in-kubernetes/>)
- 在kubernetes 集群上搭建docker 私有仓库Harbor (<https://blog.qikqiak.com/post/install-docker-registry-harbor-in-kubernetes/>)
- kubernetes dashboard 升级之路 (<https://blog.qikqiak.com/post/update-kubernetes-dashboard-more-secure/>)
- 手动搭建高可用的kubernetes 集群 (<https://blog.qikqiak.com/post/manual-install-high-available-kubernetes-cluster/>)
- Kubernetes Deployment滚动升级 (<https://blog.qikqiak.com/post/kubernetes-rollout-update/>)
- Kubernetes Helm 初体验 (<https://blog.qikqiak.com/post/first-use-helm-on-kubernetes/>)
- 为kubernetes dashboard访问用户添加权限控制 (<https://blog.qikqiak.com/post/add-authorization-for-kubernetes-dashboard/>)

← **PREVIOUS POST** ([HTTPS://BLOG.QIKQIAK.COM/POST/WEBPACK3-GET-STARTED/](https://blog.qikqiak.com/post/webpack3-get-started/))

NEXT POST → ([HTTPS://BLOG.QIKQIAK.COM/POST/TDD-DEVELOP-PYTHON-MICROSERVICE-APP/](https://blog.qikqiak.com/post/tdd-develop-python-microservice-app/))

♡ Like • 1 Liked • 2 Comments

Issue Page (<https://github.com/cnych/qikqiak-blog/issues/11>)



(<https://github.com>)



(<https://github.com/cnych>)

rolay (<https://github.com/rolay>) commented on Wed Feb 07 2018



钉钉格式有要求, 测试告警是否还要进行数据处理转发呢?

cnych (<https://github.com/cnych>) commented on Thu Mar 22 2018



@rolay (<https://github.com/rolay>) 那是必须的噻, 需要写一个server将数据转换成钉钉要求的格式~~~

(https://github.com/login/oauth/authorize?scope=public_repo&redirect_uri=https%3A%2F%2Fblog.qikqiak.com%2Fpost%2Falertmanager-of-prometheus-in-practice%2F&client_id=bdb76dbb2e9d0786e350&client_secret=b454b2a08013fd0e32013be7a63fa8fcb262b6c4)

Write Project
Login (https://github.com/login/oauth/authorize?scope=public_repo&redirect_uri=https%3A%2F%2Fblog.qikqiak.com%2Fpost%2Falertmanager-of-prometheus-in-practice%2F&client_id=bdb76dbb2e9d0786e350&client_secret=b454b2a08013fd0e32013be7a63fa8fcb262b6c4)

ent_id=bdb76dbb2e9d0786e350&client_secret=b454b2a08013fd0e32013be7a63fa8fcb262b6c4) with GitHub

Leave a comment

Styling with Markdown is supported (<https://guides.github.com/features/mastering-markdown/>)

Comment

Powered by Gitment (<https://github.com/imsun/gitment>)



(mailto:icnych@gmail.com)



(<https://github.com/cnych>)



(<https://weibo.com/cnych>)



(<https://instagram.com/cnych>)



(<https://blog.qikqiak.com/index.xml>)

• 2018 • 阳明的博客 (<https://blog.qikqiak.com/>) • 友链 (<https://blog.qikqiak.com/page/friend>)

Hugo v0.41 (<http://gohugo.io>) powered • Theme by Beautiful Jekyll (<http://deanattali.com/beautiful-jekyll/>) adapted to Beautiful Hugo (<https://github.com/halogenica/beautifulhugo>)

本站总访问量21912次 访客数8462人