



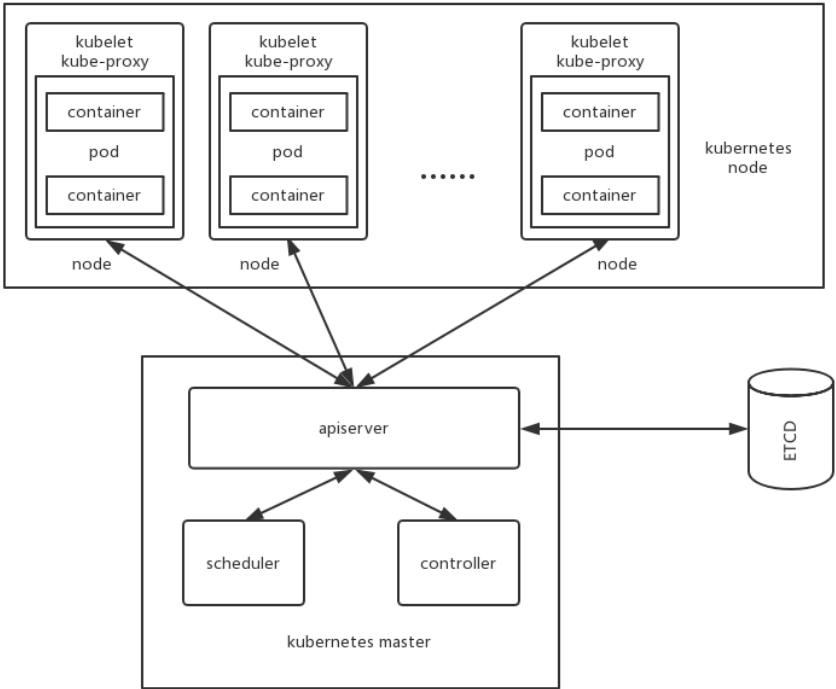
(https://yc.163yun.com/?tag=M_jssq_bky_dt001_ycdh)

使用 **Prometheus + Grafana** 对 **Kubernetes** 进行性能监控的实践 (<https://www.cnblogs.com/163yun/p/7716253.html>)
本文由 网易云 (https://www.163yun.com/?tag=M_cnblogs_7716253)发布。

1 什么是 Kubernetes?

Kubernetes 是 Google 开源的容器集群管理系统，其管理操作包括部署，调度和节点集群间扩展等。

如下图所示为目前 Kubernetes 的架构图，由 master 和 node 端构成，调度部署和扩展由 master 发起，node 协助 master 实现这些功能。



使用 Kubernetes 可以做到：

- 自动化容器的部署和复制；
- 随时扩展或收缩容器规模；
- 将容器组织成组，并且提供容器间的负载均衡；
- 提供容器弹性，如果容器失效就替换它等等。

2 Prometheus + Grafana

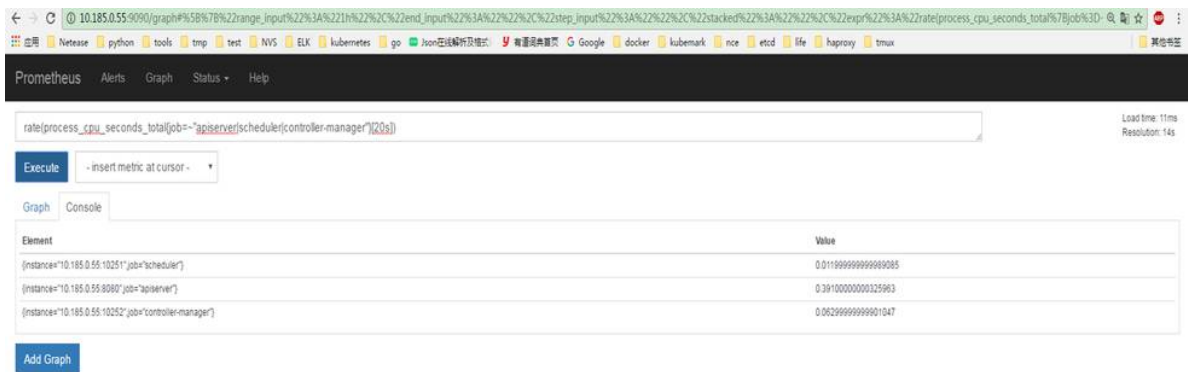
针对 Kubernetes master 端，如何评估各个组件的性能呢？目前社区提供一种搭建便捷、实用性强的监控方案：Prometheus + Grafana。

Prometheus 是使用 Golang 开发的开源监控系统，被人称为下一代监控系统，是为数不多的适合 Docker、Mesos、Kubernetes 环境的监控系统之一。

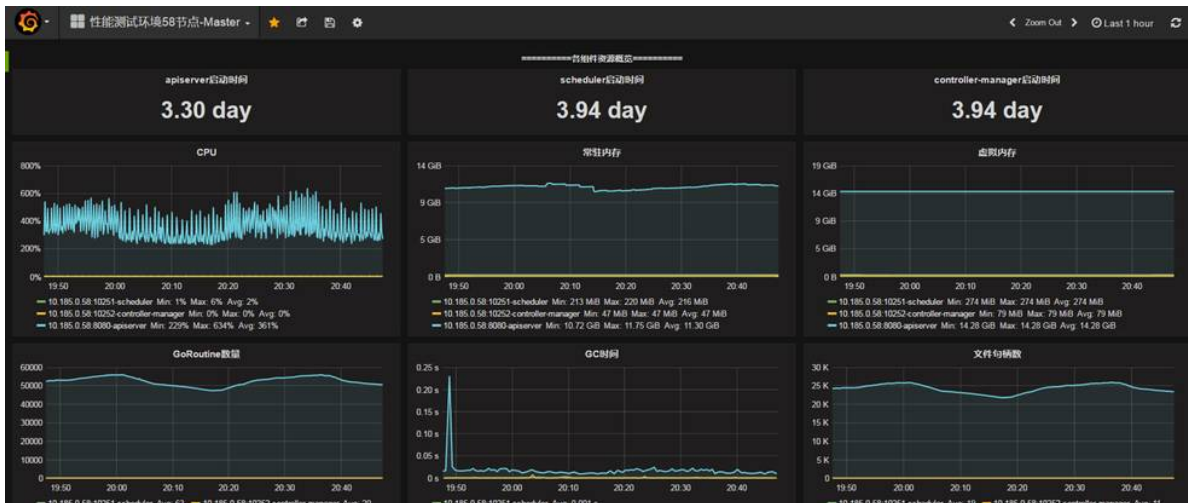
Grafana 是一个开源的图表可视化系统，简言之，其特点在于图表配置比较方便、生成的图表漂亮。

Prometheus + Grafana 监控系统的组合中，前者负责采样数据并存储这些数据；后者则侧重于形象生动的展示数据。

搭建好的这两个长（下面）这个样子，是不是感觉 grafana 的图形化展示能力很强大呢？



prometheus 截图



grafana 截图

那么它们要如何安装和配置？下面就分别对这两者进行个详细的介绍。

3 Prometheus

概念

Prometheus 是源于 Google Borgmon 的一个系统监控和报警工具，用 Golang 语言开发。基本原理是通过 HTTP 协议周期性地抓取被监控组件的状态（pull 方式），这样做的好处是任意组件只要提供 HTTP 接口就可以接入监控系统，不需要任何 SDK 或者其他的集成过程。

这样做非常适合虚拟化环境比如 VM 或者 Docker，故其为为数不多的适合 Docker、Mesos、Kubernetes 环境的监控系统之一，被很多人称为下一代监控系统。

pull 方式

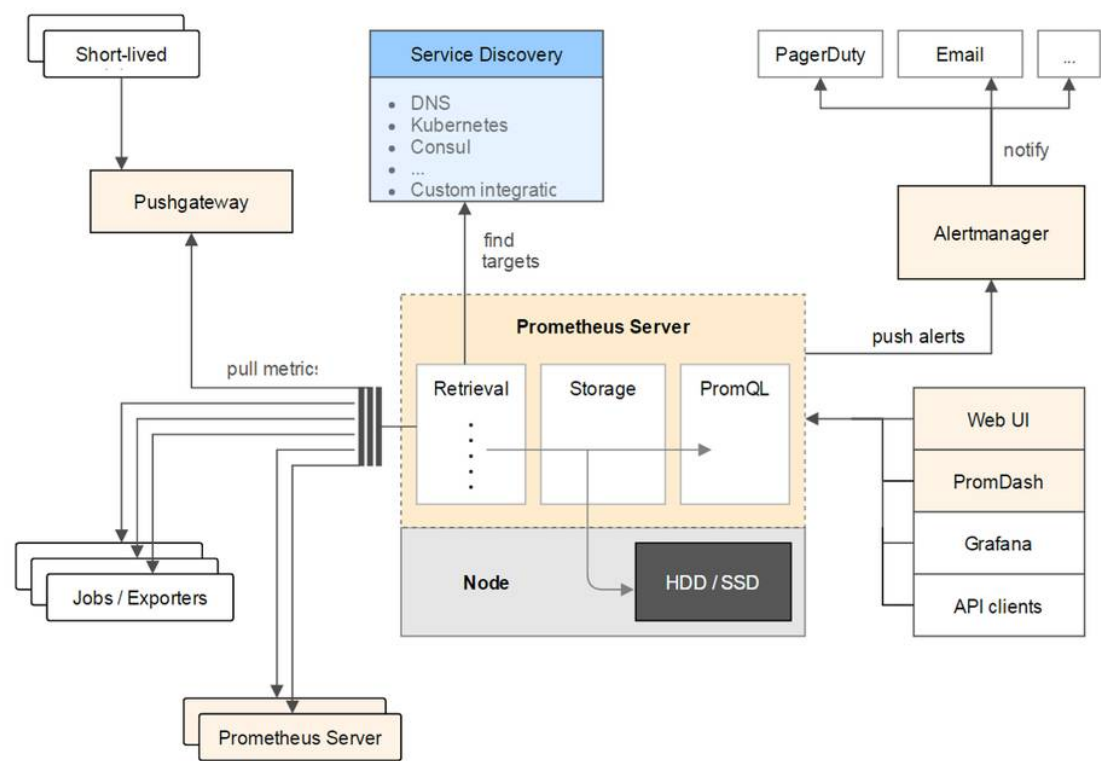
Prometheus 采集数据用的是 pull 也就是拉模型,通过 HTTP 协议去采集指标,只要应用系统能够提供 HTTP 接口就可以接入监控系统,相比于私有协议或二进制协议来说开发简单。

push 方式

对于定时任务这种短周期的指标采集，如果采用 pull 模式，可能造成任务结束了 Prometheus 还没有来得及采集的情况，这个时候可以使用加一个中转层，客户端推数据到 Push Gateway 缓存一下，由 Prometheus 从 push gateway pull 指标过来。

组成及架构

- Prometheus server：主要负责数据采集和存储，提供 PromQL 查询语言的支持；
- Push Gateway：支持临时性 Job 主动推送指标的中间网关；
- exporters：提供被监控组件信息的 HTTP 接口被叫做 exporter，目前互联网公司常用的组件大部分都有 exporter 可以直接使用，比如 Varnish、Haproxy、Nginx、MySQL、Linux 系统信息 (包括磁盘、内存、CPU、网络等等)；
- PromDash：使用 rails 开发的 dashboard，用于可视化指标数据；
- WebUI：9090 端口提供的图形化功能；
- alertmanager：实验性组件、用来进行报警；
- APIclients：提供 HTTPAPI 接口



安装与配置

下载 Prometheus

在官网 <https://prometheus.io/download/> 中选择合适的版本下载，解压。

配置文件

配置 job 和每个 job 要收集的目标 metric 数据源即可。配置文件分为 job、targets 两级，Kubernetes 的监控中主要配置 api-server 和 etcd 的 metrics 地址。

其中，api-server 和 etcd 的 metrics 地址为：

```
1 {apiserver_ip}:{apiserver_port}/metrics、{etcd_ip1}:{etcd_port1}/metrics, ... , {etcd_ipX}:{etcd_portX}/metrics
2
3 #my global config
4 global:
5   scrape_interval: 15s # By default, scrape targets every 15 seconds.pull数据的间隔时间——默认 evaluation_interval: 15s # By default, scrape targets every
6   external_labels:
7   monitor: 'codelab-monitor'# Load and evaluate rules in this file every 'evaluation_interval' seconds.
8   rule_files:
9     #- "first.rules"# - "second.rules"
10
11 #A scrape configuration containing exactly one endpoint to scrape:# Here it's Prometheus itself.
12 scrape_configs:
13   #The job name is added as a label `job=` to any timeseries scraped from this config.
14   - job_name: 'etcd-server-v2'# Override the global default and scrape targets from this job every 5 seconds.
15     scrape_interval: 10s #给每个job设置pull数据的间隔时间
16     # metrics_path defaults to '/metrics'# scheme defaults to 'http'.
17     static_configs:
18       - targets: ['{etcd_ip1}:{etcd_port1}','{etcd_ip2}:{etcd_port2}', ... , '{etcd_ipX}:{etcd_portX}']
19
20   - job_name: 'apiserver'# Override the global default and scrape targets from this job every 5 seconds.
21     scrape_interval: 10s
22     #metrics_path defaults to '/metrics'# scheme defaults to 'http'.
23     static_configs:
24       - targets: ['{apiserver_ip}:{apiserver_port}']
25
```

启动方式

源码启动——当前采用这个方式：直接启动。 参数里面指定配置文件路径、监听端口号。
nohup ./prometheus -config.file=prometheus.yml -web.listen-address ":9090" -log.level=debug 2>&1 >> run.log &

启动成功后的效果

- 访问监控页面： [http://\[host_ip\]:9090/](http://[host_ip]:9090/)（WebUI——9090 端口提供图形化功能）， status->targets 可以看到 job 下面各个 metric 的状态信息。

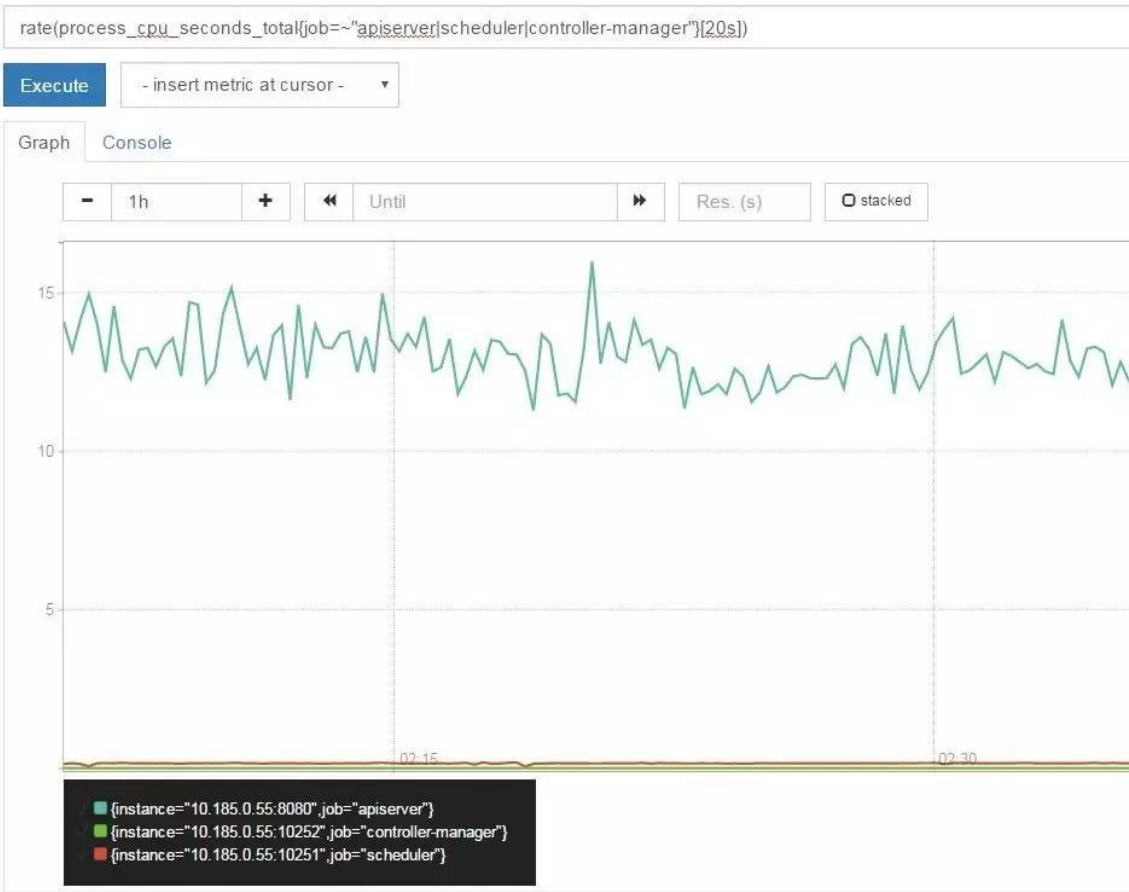
Prometheus Alerts Graph Status Help			
Targets			
apiserver			
Endpoint	State	Labels	Last Scrape
http://10.185.0.55:8080/metrics	UP	none	782ms ago
controller-manager			
Endpoint	State	Labels	Last Scrape
http://10.185.0.55:10252/metrics	UP	none	5.228s ago
etcd-server-v2			
Endpoint	State	Labels	Last Scrape
http://10.185.3.96:2379/metrics	UP	none	7.747s ago
http://10.185.3.97:2379/metrics	UP	none	9.323s ago
http://10.185.3.98:2379/metrics	UP	none	8.942s ago
http://10.185.3.99:2379/metrics	UP	none	3.305s ago

在页面 `http://[host_ip]:9090/graph` 中输入针对该 target 的 Prometheus query 语句即可实时绘图，但是趋势图不能保存，使用起来不便。

如以下 query 语句，用于计算名为 `apiserver`、`scheduler` 和 `controller-manager` 的 job 在 20s 内平均 cpu 使用率：
`rate(process_cpu_seconds_total[job=~"apiserver|scheduler|controller-manager"])[20s]`，Prometheus query 语法见第 5 节。

以上请求得到的图形可以认为是以下两个步骤的组合：

- 根据 http 请求：`http://[host_ip]:9090/api/v1/query?query=rate(process_cpu_seconds_total[job=~"apiserver|scheduler|controller-manager"])[20s]` 后得到的 json 数据（包括 job 名称、时间信息和 cpu 使用率信息等）
- 解析 json 内容，然后进行绘图



4 Grafana

Grafana 是一个开源的图表可视化系统，与 Kibana 类似，能够对后端的数据进行实时展示，简单地设图表配置比较方便、生成的图表比较漂亮。它一般和一些时间序列数据库进行配合来展示数据，例如：Graphite、OpenTSDB、InfluxDB 和 Elasticsearch 等。

安装与配置

下载 Grafana：

在官网 `http://grafana.org/download/` 选择合适版本下载、解压。

启动方式：

- 源码启动（当前采用这个方式）

配置文件在 ./conf/defaults.ini，比如默认的监听端口是 3000，data、log 之类的路径等，我们这边均使用默认配置。

```
nohup ./bin/grafana-server -homepath ./ 2>&1 >> run.log &
```

◦ docker 启动

```
docker run --name grafana \ -d \
-p 3000:3000
-v $DATAPATH:/var/lib/grafana
grafana/grafana

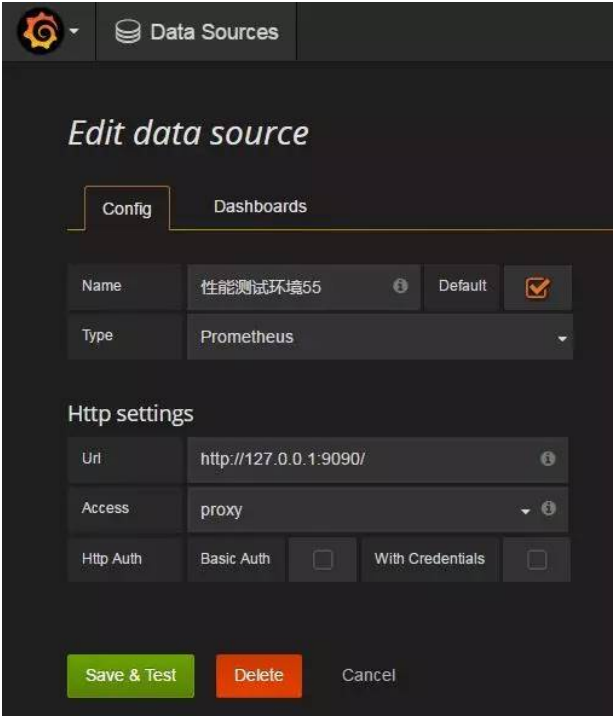
启动成功后效果
```

访问页面 `http://{host_ip}:3000/`，默认情况下管理员的账号和密码均为 admin，登录即可。

使用指南

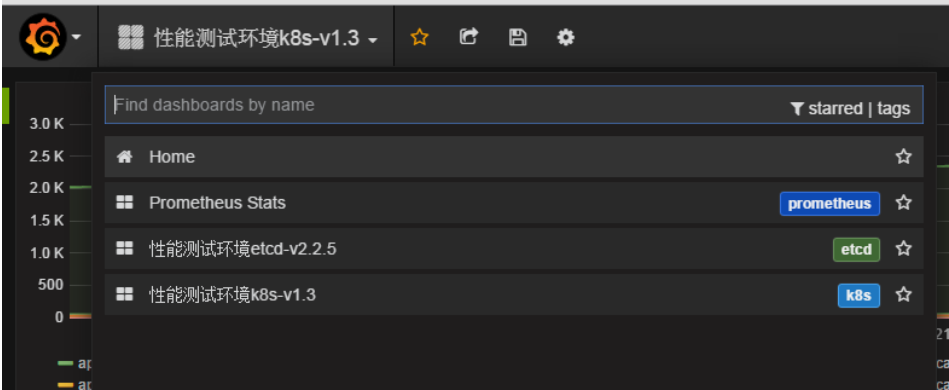
◦ 添加数据源 datasource

添加 prometheus 地址，作为一个数据源，数据源类型选择的是 prometheus。当前 prometheus 内包含了3个 job（一个 Kubernetes、一个 etcd、一个 prometheus 自身），其中 etcd 的 job 里面又有四个 target，在 grafana 中称其为 instance。



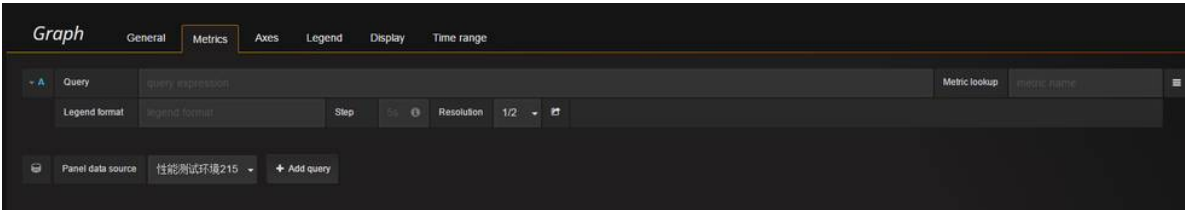
添加 dashboard

添加 dashboard，通过名字区分不同的 dashboard 即可。



◦ 为 dashboard 添加监控 panel

常见的 panel 是 graph，其中 metric 配置为核心配置。Query 中使用的是 Prometheus query 语言，一个 panel 中可以添加 n 多的 query，以图形化方式显示。（所以为了视觉美观和直观，建议图中的线条不要太多）



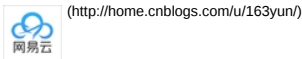
5 Prometheus Query 语言

- prometheus 的查询语法基础：<https://prometheus.io/docs/querying/basics/>
- prometheus 查询语法的操作符：<https://prometheus.io/docs/querying/operators/>
- prometheus 的函数：<https://prometheus.io/docs/querying/functions>
- prometheus 官方最佳实践：<https://prometheus.io/docs/practices/histograms/>，其中有个例子和 Kubernetes 计算平均延迟时间比较类似。

来自： 网易云-共创云上精彩世界 (https://www.163yun.com/?id=cnblogs&tag=M_cnblogs)

了解 网易云：
网易云官网：<https://www.163yun.com/> (https://www.163yun.com/?tag=M_cnblogs_7716253)
新用户大礼包：<https://www.163yun.com/gift> (https://www.163yun.com/gift?tag=M_cnblogs_7716253)
网易云社区：<https://sq.163yun.com/> (https://sq.163yun.com/?tag=M_cnblogs_7716253)

标签: [kubernetes](http://www.cnblogs.com/163yun/tag/kubernetes/) (<http://www.cnblogs.com/163yun/tag/kubernetes/>), [监控](http://www.cnblogs.com/163yun/tag/%E7%9B%91%E6%8E%A7/) (<http://www.cnblogs.com/163yun/tag/%E7%9B%91%E6%8E%A7/>)



网易云 (<http://home.cnblogs.com/u/163yun/>)
关注 - 0
(<http://home.cnblogs.com/u/163yun/followees>)
粉丝 - 69
(<http://home.cnblogs.com/u/163yun/followers>)
[+加关注](#)

« (<http://www.cnblogs.com/163yun/p/7697996.html>) 上一篇：反射型 DDoS 攻击的原理和防范措施 (<http://www.cnblogs.com/163yun/p/7697996.html>)
» (<http://www.cnblogs.com/163yun/p/8242151.html>) 下一篇：2017年内容安全十大事件盘点 (<http://www.cnblogs.com/163yun/p/8242151.html>)
posted @ 2017-10-23 13:11 网易云 (<http://www.cnblogs.com/163yun/>) 阅读(1098) 评论(1) 编辑 (<https://i.cnblogs.com/EditPosts.aspx?postid=7716253>) 收藏

评论列表

#1楼 2018-06-01 08:51 江湖一浪子 (<http://www.cnblogs.com/geekmao/>) (<http://msg.cnblogs.com/send/%E6%B1%9F%E6%B9%96%E4%B8%80%E6%B5%AA%E5%AD%90>)
图片都不显示了

[支持\(0\)](#) [反对\(0\)](#)
[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](http://www.cnblogs.com)(<http://www.cnblogs.com>)网站首页。

关注网易云



微信公众号

推荐产品



云安全

(https://www.163yun.com/product-secure?tag=M_cnblogs_zq_401)

为互联网各行业提供反垃圾、验证码、注册保护、登录保护、活动反作弊、应用加固、DDoS防护等整体安全解决方案。



网易有料

(https://youliao.163yun.com/?tag=M_cnblogs_zq_402)

将定制化信息流和精准广告快速对接到您的产品中，向终端用户提供个性化内容分发服务。



通信与视频

(https://www.163yun.com/product-media?tag=M_cnblogs_zq_403)

打造真正稳定易用的多媒体通信云平台，为您提供 IM、音视频、直播、点播、短信等多种通信能力。



云基础服务

(https://www.163yun.com/product-cloudcompute?tag=M_cnblogs_zq_404)
深度整合了 IaaS、PaaS 及容器技术，提供了弹性计算，DevOps 工具链及微服务基础设施等服务。



大数据

(https://bigdata.163yun.com/?tag=M_cnblogs_zq_405)

一站式大数据管理和应用开发平台，企业级大数据可视化分析平台。