

[Tony Bai](#)

一个程序员的心路历程

- [关于我](#)
- [文章列表](#)

使用kubectl访问Kubernetes集群时的身份验证和授权

- 六月 14, 2018
- [2 条评论](#)

[kubectl](#)是日常访问和管理[Kubernetes集群](#)最为常用的工具。

当我们使用[kubeadm](#)成功引导启动(init)一个[Kubernetes集群的控制平面](#)后，kubeadm会在init的输出结果中给予我们下面这样的“指示”：

```
... ..
Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

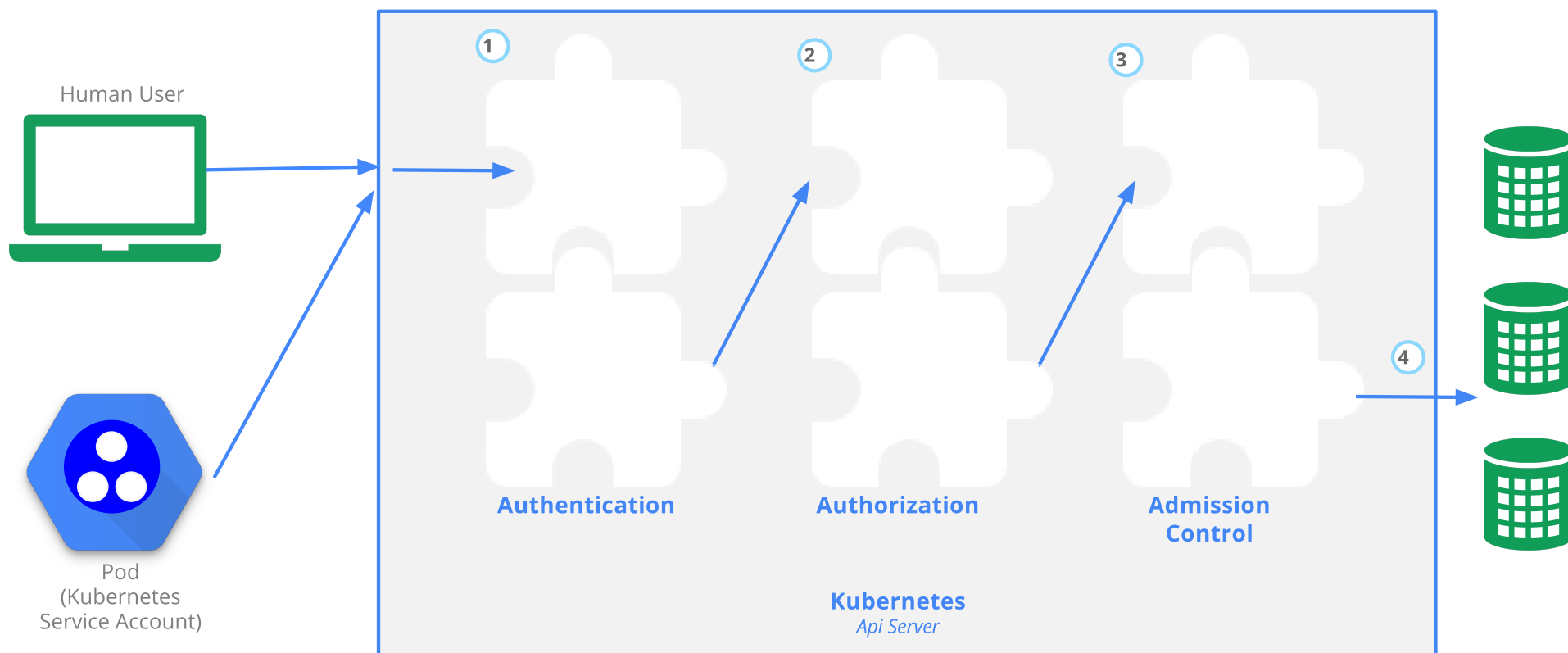
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
... ..
```

[kubeadm init](#)在结尾处输出的这些信息是在告知我们如何配置kubeconfig文件。按照上述命令配置后，master节点上的kubectl就可以直接使用\$HOME/.kube/config的信息访问k8s cluster了。并且，通过这种配置方式，kubectl也拥有了整个集群的[管理员\(root\)权限](#)。

很多K8s初学者在这里都会有疑问：当kubectl使用这种kubeconfig方式访问集群时，Kubernetes的kube-apiserver是如何对来自kubectl的访问进行[身份验证\(authentication\)](#)和[授权\(authorization\)](#)的呢？为什么来自kubectl的请求拥有最高的管理员权限呢？在本文中，我们就来分析说明一下这个过程。

一. Kubernetes API的访问控制原理回顾

在[《Kubernetes的安全设置》](#)一文中我曾介绍过Kubernetes集群的访问权限控制由kube-apiserver负责，kube-apiserver的访问权限控制由身份验证(authentication)、授权(authorization)和准入控制(admission control) 三步骤组成，这三步骤是按序进行的：



要想搞明白kubectl访问Kubernetes集群时的身份验证和授权，就是要弄清kube-apiserver在进行身份验证和授权两个环节都做了什么：

- **Authentication**：即身份验证，这个环节它面对的输入是整个http request，它负责对来自client的请求进行身份校验，支持的方法包括：client证书验证（https双向验证）、basic auth、普通token以及jwt token(用于serviceaccount)。APIServer启动时，可以指定一种Authentication方法，也可以指定多种方法。如果指定了多种方法，那么APIServer将会逐个使用这些方法对客户端请求进行验证，只要请求数据通过其中一种方法的验证，APIServer就会认为Authentication成功；在较新版本kubeadm引导启动的k8s集群的apiserver初始配置中，默认支持client证书验证和serviceaccount两种身份验证方式。在这个环节，apiserver会通过client证书或http header中的字段(比如serviceaccount的jwt token)来识别出请求的“用户身份”，包括“user”、“group”等，这些信息将在后面的authorization环节用到。

- [Authorization](#)：授权。这个环节面对的输入是http request context中的各种属性，包括：user、group、request path（比如：/api/v1、/healthz、/version等）、request verb(比如：get、list、create等)。APIServer会将这些属性值与事先配置好的访问策略(access policy) 相比较。APIServer支持多种authorization mode，包括[Node](#)、[RBAC](#)、Webhook等。APIServer启动时，可以指定一种authorization mode，也可以指定多种authorization mode，如果是后者，只要Request通过了其中一种mode的授权，那么该环节的最终结果就是授权成功。在较新版本kubeadm引导启动的k8s集群的apiserver初始配置中，authorization-mode的默认配置是”Node,RBAC”。Node授权器主要用于各个node上的kubelet访问apiserver时使用的，其他一般均由RBAC授权器来授权。

RBAC，Role-Based Access Control即Role-Based Access Control，它使用”rbac.authorization.k8s.io”实现授权决策，允许管理员通过Kubernetes API动态配置策略。在RBAC API中，一个角色(Role)包含了一组权限规则。Role有两种：Role和ClusterRole。一个Role对象只能用于授予对某一单一命名空间（namespace）中资源的访问权限。ClusterRole对象可以授予与Role对象相同的权限，但由于它们属于集群范围对象，也可以使用它们授予对以下几种资源的访问权限：

- 集群范围资源（例如节点，即node）
- 非资源类型endpoint（例如”/healthz”）
- 跨所有命名空间的命名空间范围资源（例如所有命名空间下的pod资源）

rolebinding，角色绑定则是定义了将一个角色的各种权限授予一个或者一组用户。角色绑定包含了一组相关主体（即subject, 包括用户——User、用户组——Group、或者服务账户——Service Account）以及对被授予角色的引用。在命名空间中可以通过RoleBinding对象进行用户授权，而集群范围的用户授权则可以通过ClusterRoleBinding对象完成。

好了，有了上面这些知识基础，要搞清楚kubectl访问集群的身份验证和授权过程，我们只需要逐一解决下面的一些问题即可：

- 1、authencation中识别出了哪些http request context中的信息？
- 2、authorization中RBAC authorizer找到的对应的rolebinding或clusterrolebinding是什么？
- 3、对应的role或clusterrole的权限规则？

二. 在身份验证(authentication)识别出Group

我们先从kubectl使用的kubeconfig入手。kubectl使用的kubeconfig文件实质上就是kubeadm init过程中生成的/etc/kubernetes/admin.conf，我们查看一下该kubeconfig文件的内容：

环境k8s 1.10.3：

```
# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: REDACTED
    server: https://172.16.66.101:6443
    name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
    name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

关于kubeconfig文件的解释，可以在[这里](#)自行脑补。在这些输出信息中，我们着重提取到两个信息：

```
user name: kubernetes-admin
client-certificate-date: XXXX
```

前面提到过apiserver的authentication支持通过tls client certificate、basic auth、token等方式对客户端发起的请求进行身份校验，从kubeconfig信息来看，kubectl显然在请求中使用了tls client certificate的方式，即客户端的证书。另外我们知道Kubernetes是没有[user](#)这种资源的，通过k8s API也无法创建user。那么kubectl的身份信息就应该“隐藏”在client-certificate的数据中，我们来查看一下。

首先我们将 /etc/kubernetes/admin.conf中client-certificate-data的数据内容保存到一个临时文件admin-client-certificate.txt中：

```
// admin-client-certificate.txt
```

```
LS0tLS1CRUdJTIhbnRvJUSUZJQ0FURSB0tLS0tCk1JSUM4akNDQWRxZ0F3SUJBZ01JZjJkVlJqYThFTFF3RFFZSktyWk1odmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVm1aWEp1WlhSbGN6QWVGdzb4T0RBMU1UUXdPREUZTVROyUz3MHhPVEExTVRRd09ERTNN
```

然后针对该文件数据做base64解码，得到client certificate文件：

```
cat admin-client-certificate.txt | base64 -d > admin-client.crt
```

```
# cat admin-client.crt
```

```
-----BEGIN CERTIFICATE-----
MIIC8jCCAdggAwIBAgIIf2dVRjm8ELQwDQYJKoZIhvcNAQELBQAwFTETMBEGA1UE
AxMKa3ViZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJl
FzAVBgNVBAoTdnN5c3RlbTptYXN0ZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJlZXJl
bWluMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAXbn3jdw80b1Gfb6s
w2NrgpLotMT4nyAf2HhQMrXjn0+wnaK1AITow/22mDj0rwIuJwdQIj5/BaF63pPE
lpU0vhIPVK4n6JI4dmMzo/1R3jZpGeZW1zdXaCovw9c7clbiHo/mFG4xqytVLfX4
/S8mFp2A9QcieJGIo5S0BR3FZlU1PM7DRbLDVVq1PdYNY2GfsbGrHlGgXvWAKCd/
H79gAgVoTXjSIWCVYuYcoLvdvVXQSiIYlpXFP1jBQLvcU7vrqtb12RmrnxpkW4p1
dGEOX2sLmfYZ5TiFpkRwz2GxsmWyRbt6OuISJFI6RZ0r+Rn4yMDKPrYlEngDVc5K
PZ5zmwIDAQABoycwJTAOBgNVHQ8BAf8EBAMCBAAwEwYDVRO1BAwwCgYIKwYBBQUH
AwIwDQYJKoZIhvcNAQELBQADggEBAEZNTvTz2OgzCudvMFbrhPsp+mD2vPjMRCxi
3Bka10vICOSfdymMn8aw0IbKYz2gQbXqUfQzQmQfa3if+QYBkb+77zfsv9am4EP/
e6Tg52tqV2P7s2eF7tNAe20GyV6yF1Q1QUW5/M4M+Jm1V+BUbl9yEyQ1ENucKf+u
TPyKKTUtzzvUYr5E3EJkt84EQINvw2nR2jNveZlXWoliUrKfjHhtfvO/n56USuI4w
u2LTlICRcj4g+ZWlIjeMFkGyPbJyJAQ65P2sGrZm1klGGH3mzwO5CP1yZWvoUjjP
jzSjMCiAK/fR8eRAJ6q1tT6bG26L+njkKCQDWKpjAWOapuROcbk=
-----END CERTIFICATE-----
```

查看证书内容：

```
# openssl x509 -in ./admin-client.crt -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9180400125522743476 (0x7f67554639bc10b4)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=kubernetes
        Validity
            Not Before: May 14 08:17:13 2018 GMT
            Not After : May 14 08:17:17 2019 GMT
        Subject: O=system:masters, CN=kubernetes-admin
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)

    ... ..
```

从证书输出的信息中，我们看到了下面这行：

```
Subject: O=system:masters, CN=kubernetes-admin
```

k8s apiserver对kubectl的请求进行client certificate验证(通过ca证书—client-ca-file=/etc/kubernetes/pki/ca.crt对其进行校验)，验证通过后kube-apiserver会得到：**group = system:masters**的http上下文信息，并传给后续的authorizers。

三. 在授权(authorization)时根据Group确定所绑定的角色(Role)

kubeadm在init初始引导集群启动过程中，创建了许多default的role、clusterrole、rolebinding和clusterrolebinding，在k8s有关[RBAC的官方文档](#)中，我们看到下面一些default clusterrole列表：

Default ClusterRole	Default ClusterRoleBinding	Description
cluster-admin	system:masters group	Allows super-user access to perform any action on any resource. When used in a ClusterRoleBinding , it gives full control over every resource in the cluster and in all namespaces. When used in a RoleBinding , it gives full control over every resource in the rolebinding's namespace, including the namespace itself.
admin	None	Allows admin access, intended to be granted within a namespace using a RoleBinding . If used in a RoleBinding , allows read/write access to most resources in a namespace, including the ability to create roles and rolebindings within the namespace. It does not allow write access to resource quota or to the namespace itself.
edit	None	Allows read/write access to most objects in a namespace. It does not allow viewing or modifying roles or rolebindings.
view	None	Allows read-only access to see most objects in a namespace. It does not allow viewing roles or rolebindings. It does not allow viewing secrets, since those are escalating.

其中第一个**cluster-admin**这个cluster role binding绑定了**system:masters** group，这和authentication环节传递过来的身份信息不谋而合。沿着**system:masters** group对应的cluster-admin clusterrolebinding“追查”下去，真相就会浮出水面。

我们查看一下这一binding：

```
# kubectl get clusterrolebinding/cluster-admin -n kube-system -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  creationTimestamp: 2018-06-07T06:14:55Z
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: cluster-admin
  resourceVersion: "103"
  selfLink: /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/cluster-admin
  uid: 18c89690-6a1a-11e8-a0e8-00163e0cd764
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:masters
```

我们看到在kube-system名字空间中，一个名为cluster-admin的clusterrolebinding将cluster-admin cluster role与system:masters Group绑定到了一起，赋予了所有归属于system:masters Group中用户cluster-admin角色所拥有的权限。

我们再来查看一下cluster-admin这个role的具体权限信息：

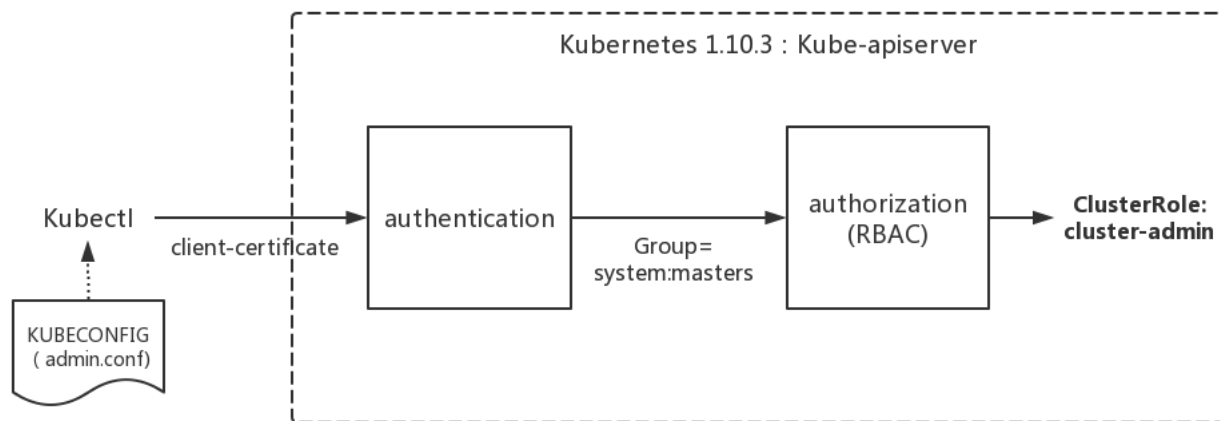
```
# kubectl get clusterrole/cluster-admin -n kube-system -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
```

```
rbac.authorization.kubernetes.io/autoupdate: "true"
creationTimestamp: 2018-06-07T06:14:55Z
labels:
  kubernetes.io/bootstrapping: rbac-defaults
name: cluster-admin
resourceVersion: "52"
selfLink: /apis/rbac.authorization.k8s.io/v1/clusterroles/cluster-admin
uid: 18abe535-6a1a-11e8-a0e8-a0163e0cd764
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

从rules列表来看，cluster-admin这个角色对所有resources、verbs、apiGroups均有无限制的操作权限，即整个集群的root权限。于是kubectl的请求就可以操控和管理整个集群了。

四. 小结

至此，我们应该明确了为什么采用了admin.conf kubeconfig的kubectl拥有root权限了。下面是一幅示意图，简要总结了对kubectl访问请求的身份验证和授权过程：



大家可以结合这幅图，重温一下上面的文字描述，加深一下理解。

更多内容可以通过我在慕课网开设的实战课程[《Kubernetes实战 高可用集群搭建、配置、运维与应用》](#)学习。

[51短信平台](https://51smspush.com/)：企业级短信平台定制开发专家 <https://51smspush.com/>

smspush：可部署在企业内部的定制化短信平台，三网覆盖，不惧大并发接入，可定制扩展； 短信内容你来定，不再受约束，接口丰富，支持长短信，签名可选。

著名云主机服务厂商DigitalOcean发布最新的主机计划，入门级Droplet配置升级为：1 core CPU、1G内存、25G高速SSD，价格5\$/月。有使用DigitalOcean需求的朋友，可以打开这个[链接地址](https://m.do.co/c/bff6eed92687)：<https://m.do.co/c/bff6eed92687> 开启你的DO主机之路。

我的联系方式：

<https://tonybai.com/2018/06/14/the-authentication-and-authorization-of-kubectl-when-accessing-k8s-cluster/>

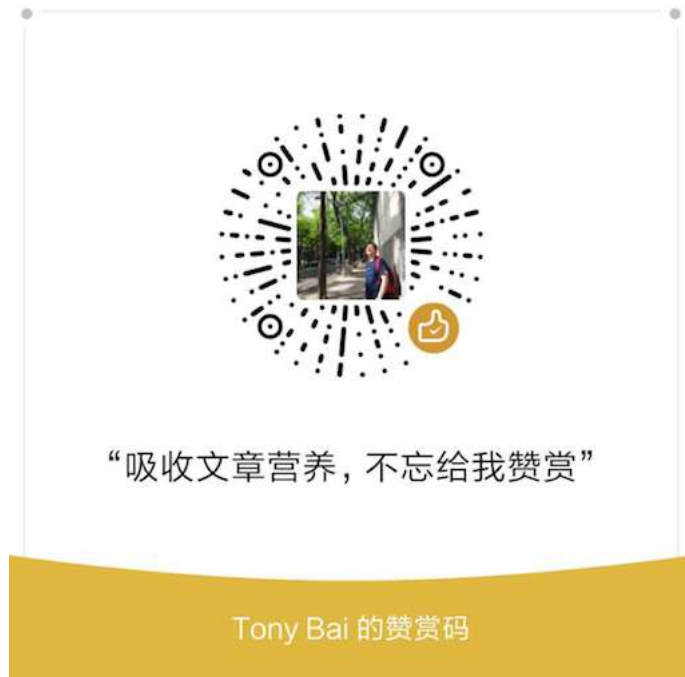
微博: <https://weibo.com/bigwhite20xx>

微信公众号: iamtonybai

博客: tonybai.com

github: <https://github.com/bigwhite>

微信赞赏:



商务合作方式: 撰稿、出书、培训、在线课程、合伙创业、咨询、广告合作。

© 2018, bigwhite. 版权所有.

Related posts:

1. [解决Kubernetes 1.6.4 Dashboard无法访问的问题](#)
2. [在Kubernetes Pod中使用Service Account访问API Server](#)
3. [Kubernetes集群的安全配置](#)
4. [以Kubeadm方式安装的Kubernetes集群的探索](#)
5. [Kubernetes集群Dashboard插件安装](#)

[已有 2 条评论](#)

1.  *pzhang*
2019/03/29

k8s apiserver对kubectl的请求进行client certificate验证(通过ca证书—client-ca-file=/etc/kubernetes/pki/ca.crt对其进行校验)

“apiserver 对 kubectl 的请求”这里是不是写反了？

[回复](#)



wp

2019/04/29

确实是这样，没有反，apiserve也需要验证kubectl的，通过证书或token。源码中共有4种方式的

[回复](#)

添加新评论

称呼

邮箱

网站

提交评论

Capatcha

搜索

如发现本站页面被黑，比如：挂载广告、挖矿等恶意代码，请朋友们及时[联系我](#)。十分感谢！

赞助商广告位1

图片广告+链接跳转

广告首页展示，欢迎合作 商务合作请联系bigwhite.cn AT aliyun.com

欢迎使用邮件订阅我的博客

输入邮箱订阅本站，只要有新文章发布，就会第一时间发送邮件通知你哦！

名字：

邮箱：

马上订阅



这里是 [Tony Bai](#)的个人Blog, 欢迎访问、订阅和留言! [订阅Feed请点击上面图片。](#)

如果您觉得这里的文章对您有帮助, 请扫描上方二维码进行捐赠, 加油后的Tony Bai将会为您呈现更多精彩的文章, 谢谢!

如果您希望通过微信捐赠, 请用微信客户端扫描下方二维码:



如果您希望通过比特币或以太币捐赠, 可以扫描下方二维码:

比特币:



以太币:



如果您喜欢通过微信浏览本站内容，可以扫描下方二维码，订阅本站官方微信订阅号“iamtonybai”；点击二维码，可直达本人官方微博主页^_^：



本站Powered by Digital Ocean VPS。

[选择Digital Ocean VPS主机，即可获得10美元现金充值，可免费使用两个月哟！](#) 著名主机提供商Linode 10\$优惠码：linode10，在 [这里注册](#)即可免费获得。阿里云推荐码：**1WFZ0V**，**立享9折！**



我的业余项目

- [smspush短信发送平台](#)

文章

- [Go modules：最小版本选择](#)
- [Kubernetes Deployment故障排除图解指南](#)
- [计算重现性：一些挑战](#)
- [Go官方发布的go.dev给gopher们带来了什么](#)
- [Go语言开源十周年](#)
- [Go语言项目的安全评估技术](#)
- [图解中文字符编码-Go语言例解](#)
- [Go语言的遗产](#)
- [Go 1.13中值得关注的几个变化](#)
- [如何在Ubuntu 18.04 Server上部署Kubernetes集群](#)

评论

-  bigwhite 在 [Go语言TCP Socket编程](#)
这个就是Wordpress自带的功能啊。
-  PierreVon 在 [Go语言TCP Socket编程](#)
你这个评论回复功能做的好厉害，请问是用的sql还是nosql存储的评论，前端是怎么渲染的呀？
-  爱玩编程的大二学弟 在 [也谈Go的可移植性](#)
受益匪浅

-  [Chris](#) 在 [关于我](#)
你好，请问是否能交换博客链接？ <https://jingine.com>
-  [yyli](#) 在 [理解Golang包导入](#)
謝謝樓主
-  [BillZong](#) 在 [Go modules：最小版本选择](#)
翻译很准确到位，谢谢又Get到不少知识点
-  [bigwhite](#) 在 [如何在Go语言中使用Websockets：最佳工具与行动指南](#)
文中有标注：这是一篇译文。原文在<https://yalantis.com/blog/how-to-b...>
-  [wei](#) 在 [如何在Go语言中使用Websockets：最佳工具与行动指南](#)
本文的源码能否公开一下？有些片段文中没有展示.....
-  [lostpg](#) 在 [也谈goroutine调度器](#)
请问 G 的抢占调度那部分，「可以看出，如果一个 G 任务运行 10ms，sysmon 就会认为其运...
-  [bigwhite](#) 在 [HTTPS服务的Kubernetes ingress配置实践](#)
<https://github.com/bigwhite/experiments>下面的ingress-...
- [下一页](#) »

分类

- [光影汇](#) (7)
- [影音坊](#) (36)
- [思考控](#) (66)
- [技术志](#) (601)
- [教育记](#) (1)
- [杂货铺](#) (75)
- [生活簿](#) (154)
- [职场录](#) (14)
- [读书吧](#) (14)
- [运动迷](#) (107)
- [驴友秀](#) (40)

标签

[Blog](#) [Blogger](#) [C](#) [container](#) [C++](#) [docker](#) [GCC](#) [github](#) [GNU](#) [Go](#) [Golang](#) [Google](#) [Java](#) [k8s](#) [Kernel](#) [Kubernetes](#) [Linux](#) [M10](#) [Opensource](#) [Programmer](#) [Python](#) [Solaris](#) [Subversion](#) [Ubuntu](#) [Unix](#) [Windows](#) [世界杯](#) [博](#)
[客](#) [学习](#) [容器](#) [工作](#) [巴萨](#) [开源](#) [思考](#) [感悟](#) [摄影](#) [旅游](#) [梅西](#) [球王](#) [生活](#) [程序员](#) [编译器](#) [翻译](#) [西甲](#) [足球](#)

归档

- [2019 年十二月](#) (2)
- [2019 年十一月](#) (6)
- [2019 年十月](#) (5)
- [2019 年九月](#) (4)
- [2019 年八月](#) (5)
- [2019 年七月](#) (1)
- [2019 年六月](#) (2)
- [2019 年五月](#) (1)
- [2019 年四月](#) (4)
- [2019 年三月](#) (2)
- [2019 年二月](#) (1)
- [2019 年一月](#) (2)
- [2018 年十一月](#) (3)
- [2018 年十月](#) (1)
- [2018 年九月](#) (1)
- [2018 年七月](#) (1)
- [2018 年六月](#) (4)
- [2018 年五月](#) (2)
- [2018 年四月](#) (1)
- [2018 年三月](#) (3)
- [2018 年二月](#) (3)
- [2018 年一月](#) (7)
- [2017 年十二月](#) (5)
- [2017 年十一月](#) (4)
- [2017 年十月](#) (3)
- [2017 年九月](#) (2)
- [2017 年八月](#) (3)
- [2017 年七月](#) (4)
- [2017 年六月](#) (8)
- [2017 年五月](#) (5)
- [2017 年四月](#) (3)
- [2017 年三月](#) (2)
- [2017 年二月](#) (5)
- [2017 年一月](#) (7)
- [2016 年十二月](#) (7)
- [2016 年十一月](#) (7)
- [2016 年十月](#) (3)
- [2016 年九月](#) (2)
- [2016 年八月](#) (1)
- [2016 年六月](#) (2)
- [2016 年五月](#) (2)
- [2016 年四月](#) (2)
- [2016 年三月](#) (2)
- [2016 年二月](#) (3)
- [2016 年一月](#) (2)
- [2015 年十二月](#) (1)
- [2015 年十一月](#) (1)

- [2015 年十月](#) (1)
- [2015 年九月](#) (3)
- [2015 年八月](#) (5)
- [2015 年七月](#) (6)
- [2015 年六月](#) (4)
- [2015 年五月](#) (1)
- [2015 年四月](#) (2)
- [2015 年三月](#) (2)
- [2015 年一月](#) (2)
- [2014 年十二月](#) (5)
- [2014 年十一月](#) (8)
- [2014 年十月](#) (9)
- [2014 年九月](#) (2)
- [2014 年八月](#) (1)
- [2014 年七月](#) (1)
- [2014 年五月](#) (2)
- [2014 年四月](#) (5)
- [2014 年三月](#) (4)
- [2014 年二月](#) (1)
- [2014 年一月](#) (1)
- [2013 年十二月](#) (3)
- [2013 年十一月](#) (5)
- [2013 年十月](#) (6)
- [2013 年九月](#) (4)
- [2013 年八月](#) (5)
- [2013 年七月](#) (6)
- [2013 年六月](#) (2)
- [2013 年五月](#) (6)
- [2013 年四月](#) (3)
- [2013 年三月](#) (7)
- [2013 年二月](#) (4)
- [2013 年一月](#) (6)
- [2012 年十二月](#) (8)
- [2012 年十一月](#) (10)
- [2012 年十月](#) (5)
- [2012 年九月](#) (3)
- [2012 年八月](#) (10)
- [2012 年七月](#) (4)
- [2012 年六月](#) (2)
- [2012 年五月](#) (4)
- [2012 年四月](#) (10)
- [2012 年三月](#) (8)
- [2012 年二月](#) (6)
- [2012 年一月](#) (6)
- [2011 年十二月](#) (4)
- [2011 年十一月](#) (4)
- [2011 年十月](#) (5)
- [2011 年九月](#) (8)
- [2011 年八月](#) (7)

- [2011 年七月](#) (6)
- [2011 年六月](#) (7)
- [2011 年五月](#) (8)
- [2011 年四月](#) (6)
- [2011 年三月](#) (10)
- [2011 年二月](#) (7)
- [2011 年一月](#) (10)
- [2010 年十二月](#) (7)
- [2010 年十一月](#) (6)
- [2010 年十月](#) (7)
- [2010 年九月](#) (12)
- [2010 年八月](#) (8)
- [2010 年七月](#) (3)
- [2010 年六月](#) (5)
- [2010 年五月](#) (4)
- [2010 年四月](#) (2)
- [2010 年三月](#) (6)
- [2010 年二月](#) (4)
- [2010 年一月](#) (6)
- [2009 年十二月](#) (6)
- [2009 年十一月](#) (6)
- [2009 年十月](#) (5)
- [2009 年九月](#) (8)
- [2009 年八月](#) (8)
- [2009 年七月](#) (8)
- [2009 年六月](#) (2)
- [2009 年五月](#) (5)
- [2009 年四月](#) (7)
- [2009 年三月](#) (12)
- [2009 年二月](#) (9)
- [2009 年一月](#) (15)
- [2008 年十二月](#) (9)
- [2008 年十一月](#) (5)
- [2008 年十月](#) (10)
- [2008 年九月](#) (13)
- [2008 年八月](#) (13)
- [2008 年七月](#) (3)
- [2008 年六月](#) (1)
- [2008 年五月](#) (7)
- [2008 年四月](#) (4)
- [2008 年三月](#) (9)
- [2008 年二月](#) (11)
- [2008 年一月](#) (15)
- [2007 年十二月](#) (11)
- [2007 年十一月](#) (14)
- [2007 年十月](#) (4)
- [2007 年九月](#) (5)
- [2007 年八月](#) (1)
- [2007 年七月](#) (10)

- [2007 年六月](#) (10)
- [2007 年五月](#) (10)
- [2007 年四月](#) (8)
- [2007 年三月](#) (15)
- [2007 年二月](#) (4)
- [2007 年一月](#) (17)
- [2006 年十二月](#) (18)
- [2006 年十一月](#) (9)
- [2006 年十月](#) (11)
- [2006 年九月](#) (6)
- [2006 年八月](#) (5)
- [2006 年七月](#) (22)
- [2006 年六月](#) (35)
- [2006 年五月](#) (24)
- [2006 年四月](#) (26)
- [2006 年三月](#) (25)
- [2006 年二月](#) (18)
- [2006 年一月](#) (15)
- [2005 年十二月](#) (10)
- [2005 年十一月](#) (10)
- [2005 年九月](#) (13)
- [2005 年八月](#) (11)
- [2005 年七月](#) (6)
- [2005 年六月](#) (2)
- [2005 年五月](#) (3)
- [2005 年四月](#) (6)
- [2005 年三月](#) (1)
- [2005 年一月](#) (15)
- [2004 年十二月](#) (9)
- [2004 年十一月](#) (14)
- [2004 年十月](#) (2)
- [2004 年九月](#) (2)

私人

- [我的女儿](#)

链接

- [@douban](#)
- [@flickr](#)
- [@github](#)
- [@googlecode](#)
- [@picasa](#)
- [@slideshare](#)
- [@twitter](#)
- [@weibo](#)
- [Hoterran](#)
- [Lionel Messi](#)

- [Puras He](#)
- [梦想风暴](#)
- [磊磊落落的博客](#)
- [过眼云烟](#)

开源项目

- [buildc](#)
- [cbehave](#)
- [lcut](#)

翻译项目

- [C语言编码风格和标准](#)
- [《Programming in Haskell》中文翻译项目](#)

02262000 [View My Stats](#)

© 2020 [Tony Bai](#). 由 [Wordpress](#) 强力驱动. 模板由[cho](#)制作.