112Kubernetes 系列（一零五）Labels 和 annotations 用法

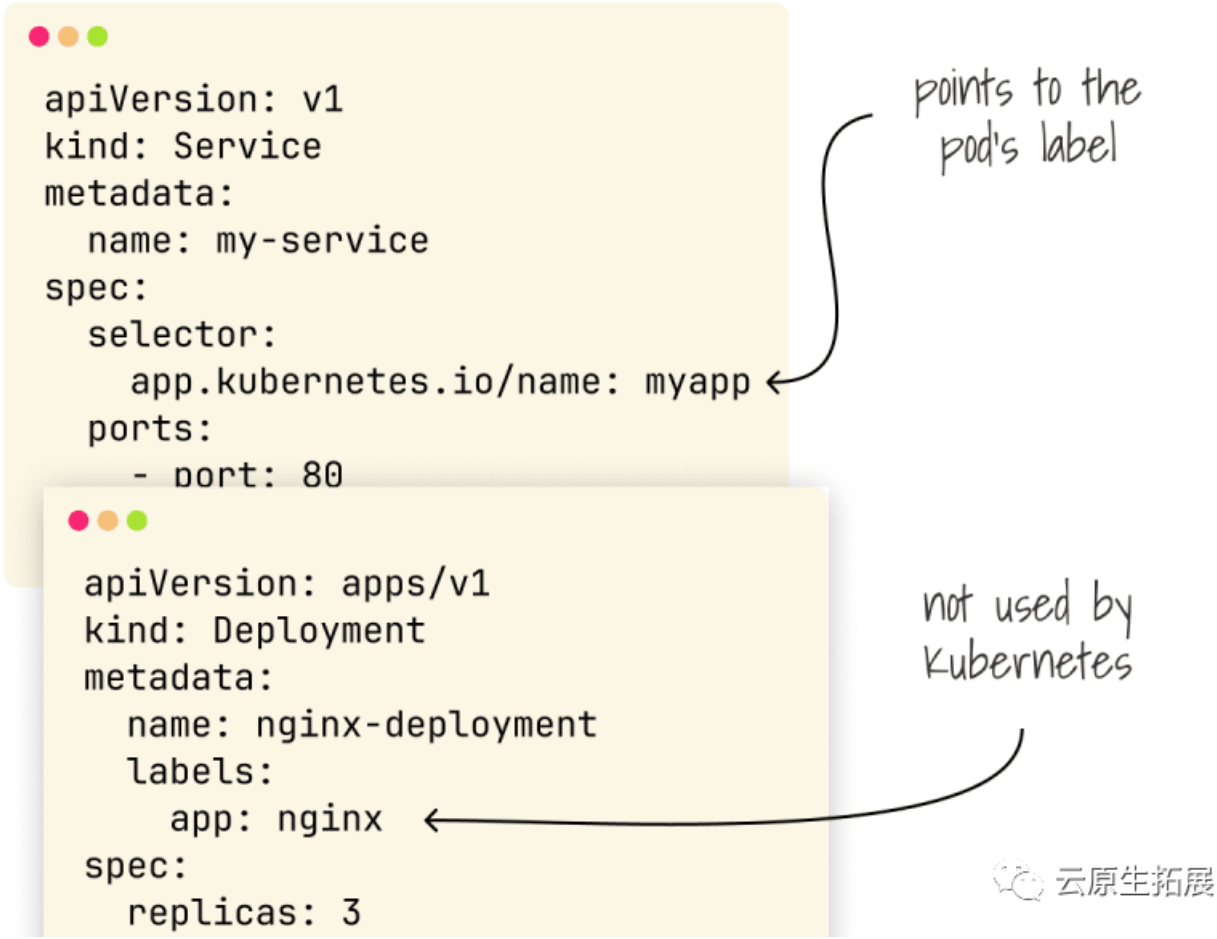
在 Kubernetes 中，您可以使用标签（Labels）将键值对配置到任何资源上。

标签无处不在，对于创建 Services 等日常运营对象来说是必需的。

但是，您应该如何命名和使用这些标签呢？

Kubernetes 中的任何资源都可以有标签。

有些标签至关重要（例如 Services 的selector,operator等），而其他标签对于标记资源很有用（例如标记部署）。



`Kubectl` 提供了一个 `--show-labels` 标志来帮助您列出资源及其标签。

如果您在空集群中列出 Pod、Deployment 和 Service，您可能会注意到 Kubernetes 使用 `component=<name>` 标签来标记 Pod。

Kubernetes 为您的资源推荐了六个标签：

- Name
- Instance
- Version
- Component
- Part of

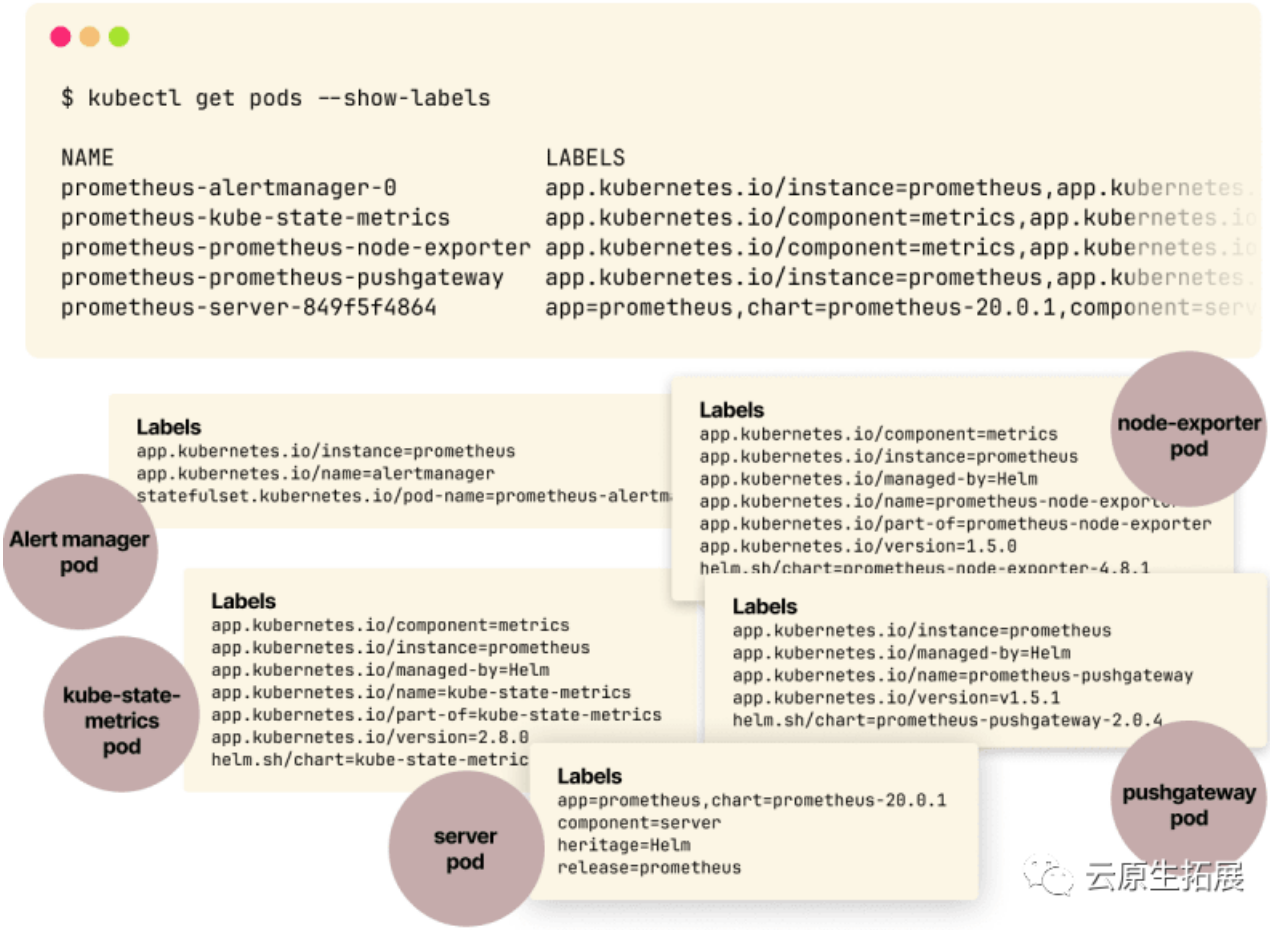
- Managed By

Key	Description	Example	Type
app.kubernetes.io/name	The name of the application	mysql	string
app.kubernetes.io/instance	A unique name identifying the instance of an application	mysql-abcxzy	string
app.kubernetes.io/version	The current version of the application (e.g., a SemVer 1.0, revision hash, etc.)	5.7.21	string
app.kubernetes.io/component	The component within the architecture	database	string
app.kubernetes.io/part-of	The name of a higher level application this one is part of	wordpress	string
app.kubernetes.io/managed-by	The tool being used to manage the operation of an application	helm	string

让我们看一下使用这些标签的一个很好的例子：Prometheus Helm Chart(https://github.com/prometheus-community/helm-charts)。

这些 Chart安装了五个 pod（即 Server、alert manager、node exporter、push gateway和 kube state metrics）。

请注意并非所有标签都应用于所有 Pod。

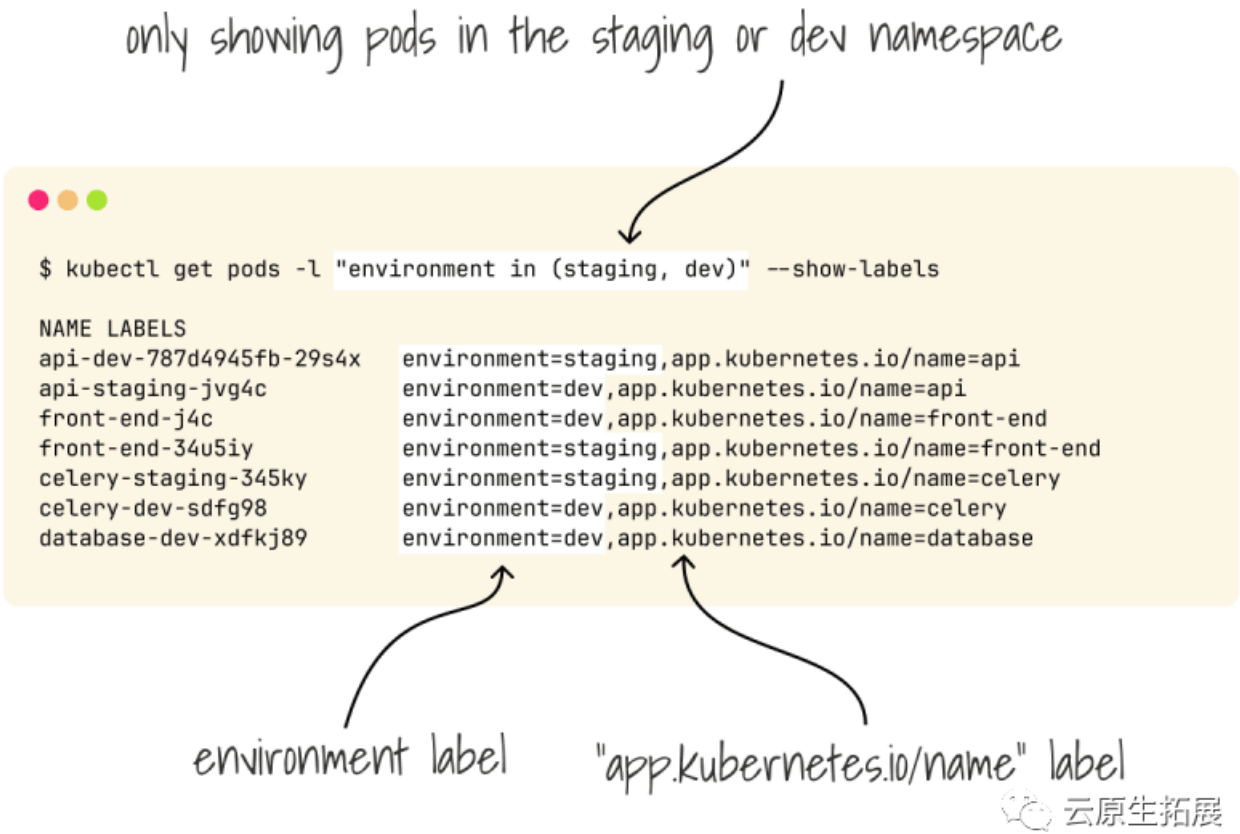


正确标记资源可以帮助您了解所部署的内容。

例如，您可以使用 kubectl 过滤结果：

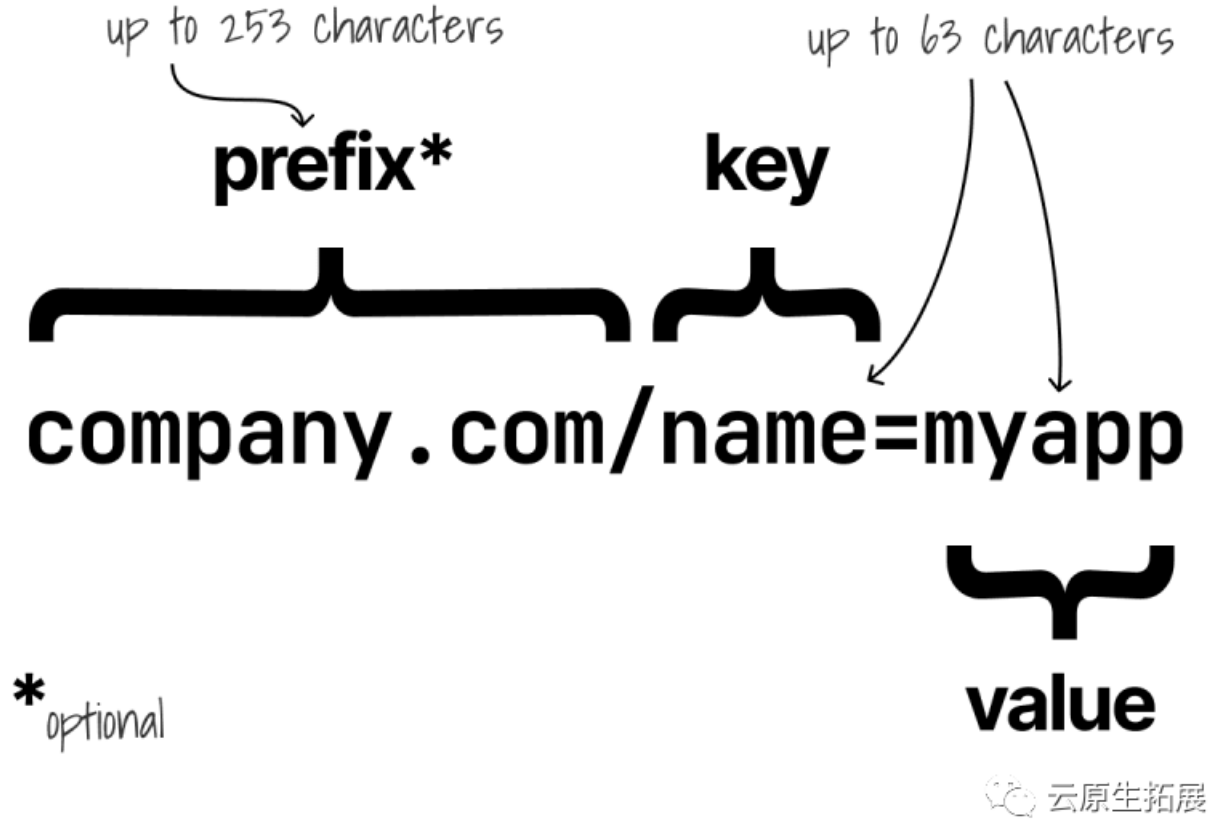
```
kubectl get pods -l "environment in (staging, dev)"
```

上面的命令仅列出 staging 和 dev 中的 pod。



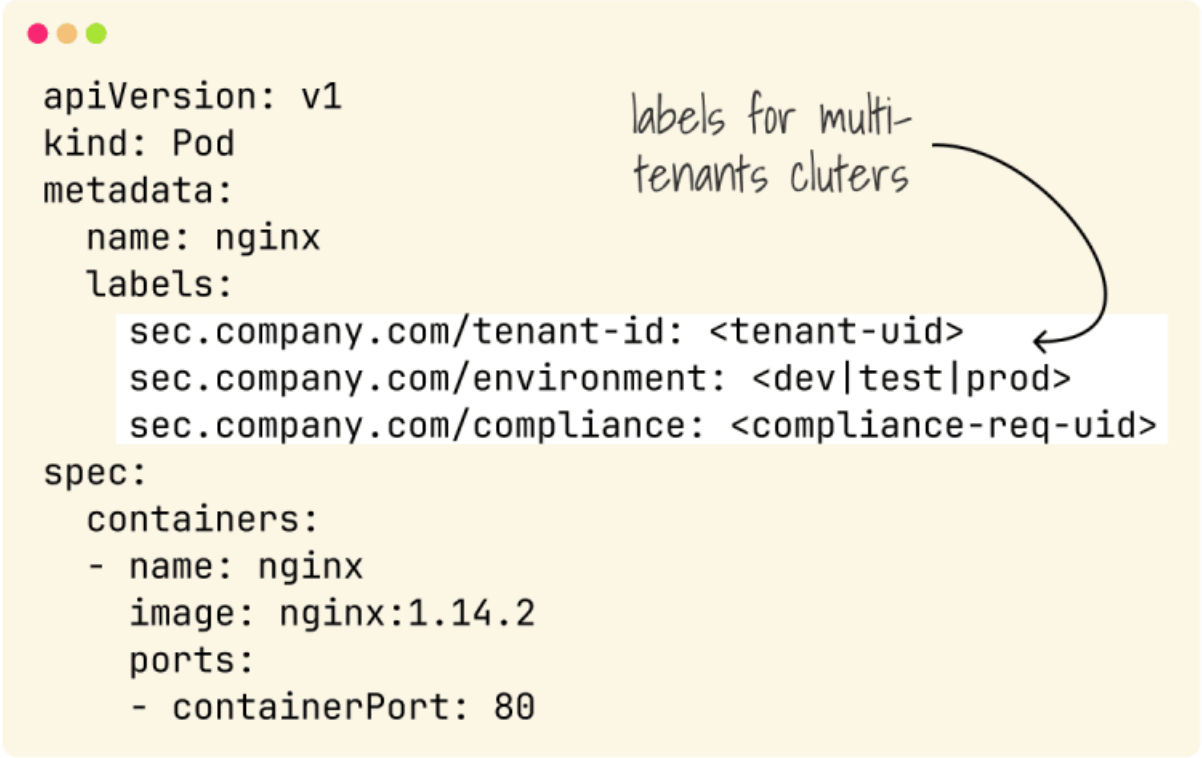
如果这些标签不是您想要的，您可以随时创建自己的标签。

建议使用 `<prefix>/<name>` 键 - 例如 `company.com/database` 。



以下标签可以在多租户集群中使用：

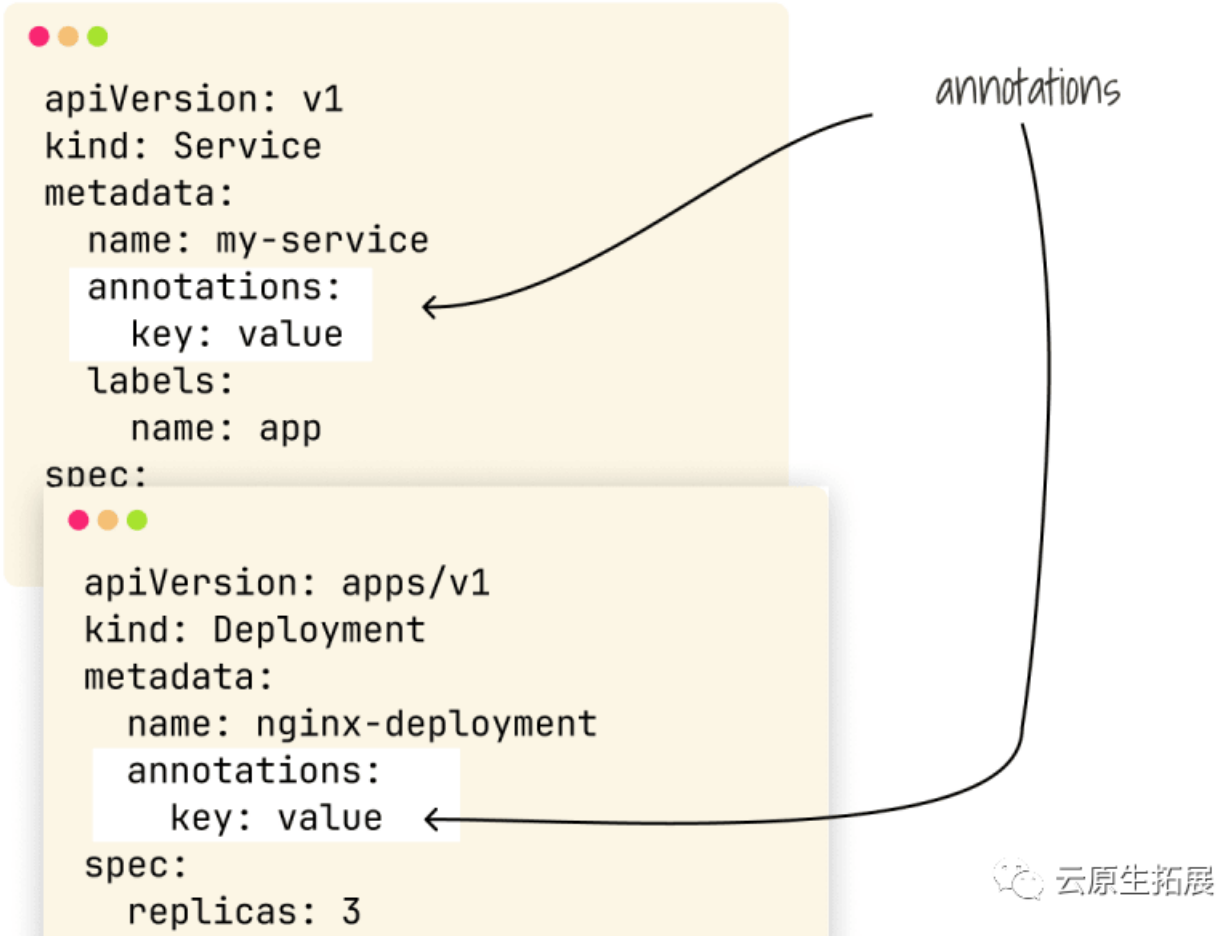
- Business unit
- Development team
- Application
- Client
- Shared services
- Environment
- Compliance
- Asset classification



除了标签之外，还有注释（annotation）。

标签用于选择资源，而注释则使用元数据来装饰资源。

您无法选择带注释的资源。



管理员可以为任何工作负载分配注释。

然而，更常见的是，Kubernetes 和 Operators 用额外的注释来装饰资源。

一个很好的例子是为 Pod 分配带宽的注释 `kubernetes.io/ingress-bandwidth` 。

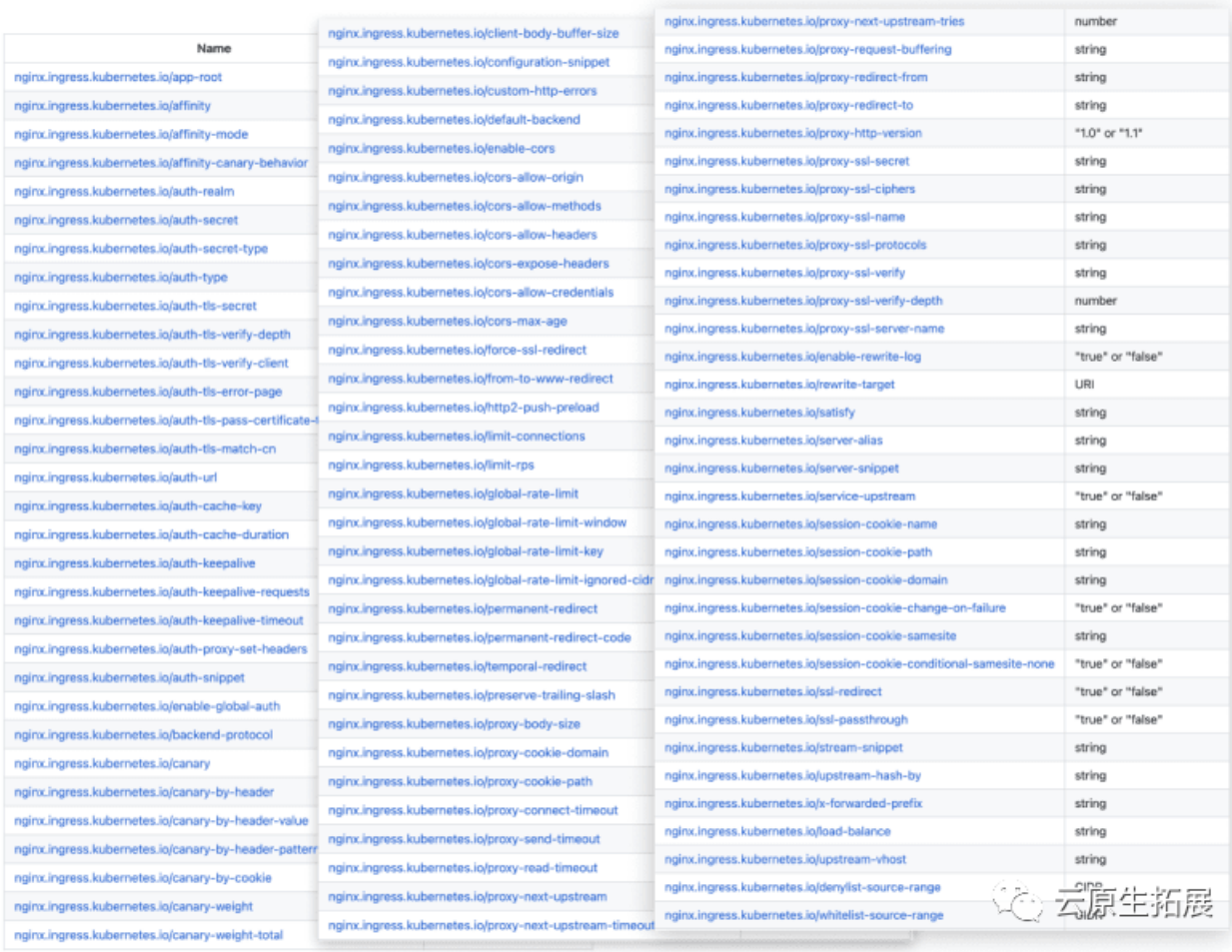
官方文档有一个众所周知的标签和注释的列表(https://kubernetes.io/docs/reference/labels-annotations-taints/)。

这里有些例子：

- kubectl.kubernetes.io/default-container
- topology.kubernetes.io/region
- node.kubernetes.io/instance-type
- kubernetes.io/egress-bandwidth

注释在 Operators 中广泛使用。

查看可与 ingress-nginx 控制器一起使用的所有注释（https://github.com/kubernetes/ingress-nginx/blob/main/docs/user-guide/nginx-configuration/annotations.md）。



Name			
nginx.ingress.kubernetes.io/app-root	nginx.ingress.kubernetes.io/client-body-buffer-size	nginx.ingress.kubernetes.io/proxy-next-upstream-tries	number
nginx.ingress.kubernetes.io/affinity	nginx.ingress.kubernetes.io/configuration-snippet	nginx.ingress.kubernetes.io/proxy-request-buffering	string
nginx.ingress.kubernetes.io/affinity-mode	nginx.ingress.kubernetes.io/custom-http-errors	nginx.ingress.kubernetes.io/proxy-redirect-from	string
nginx.ingress.kubernetes.io/affinity-canary-behavior	nginx.ingress.kubernetes.io/default-backend	nginx.ingress.kubernetes.io/proxy-redirect-to	string
nginx.ingress.kubernetes.io/auth-realm	nginx.ingress.kubernetes.io/enable-cors	nginx.ingress.kubernetes.io/proxy-http-version	"1.0" or "1.1"
nginx.ingress.kubernetes.io/auth-secret	nginx.ingress.kubernetes.io/cors-allow-origin	nginx.ingress.kubernetes.io/proxy-ssl-secret	string
nginx.ingress.kubernetes.io/auth-secret-type	nginx.ingress.kubernetes.io/cors-allow-methods	nginx.ingress.kubernetes.io/proxy-ssl-ciphers	string
nginx.ingress.kubernetes.io/auth-type	nginx.ingress.kubernetes.io/cors-allow-headers	nginx.ingress.kubernetes.io/proxy-ssl-name	string
nginx.ingress.kubernetes.io/auth-tls-secret	nginx.ingress.kubernetes.io/cors-expose-headers	nginx.ingress.kubernetes.io/proxy-ssl-protocols	string
nginx.ingress.kubernetes.io/auth-tls-verify-depth	nginx.ingress.kubernetes.io/cors-allow-credentials	nginx.ingress.kubernetes.io/proxy-ssl-verify	string
nginx.ingress.kubernetes.io/auth-tls-verify-client	nginx.ingress.kubernetes.io/cors-max-age	nginx.ingress.kubernetes.io/proxy-ssl-verify-depth	number
nginx.ingress.kubernetes.io/auth-tls-error-page	nginx.ingress.kubernetes.io/force-ssl-redirect	nginx.ingress.kubernetes.io/proxy-ssl-server-name	string
nginx.ingress.kubernetes.io/auth-tls-pass-certificate-	nginx.ingress.kubernetes.io/from-to-www-redirect	nginx.ingress.kubernetes.io/enable-rewrite-log	"true" or "false"
nginx.ingress.kubernetes.io/auth-tls-match-cn	nginx.ingress.kubernetes.io/http2-push-preload	nginx.ingress.kubernetes.io/rewrite-target	URI
nginx.ingress.kubernetes.io/auth-url	nginx.ingress.kubernetes.io/limit-connections	nginx.ingress.kubernetes.io/satisfy	string
nginx.ingress.kubernetes.io/auth-cache-key	nginx.ingress.kubernetes.io/limit-rps	nginx.ingress.kubernetes.io/server-alias	string
nginx.ingress.kubernetes.io/auth-cache-duration	nginx.ingress.kubernetes.io/global-rate-limit	nginx.ingress.kubernetes.io/server-snippet	string
nginx.ingress.kubernetes.io/auth-keepalive	nginx.ingress.kubernetes.io/global-rate-limit-window	nginx.ingress.kubernetes.io/service-upstream	"true" or "false"
nginx.ingress.kubernetes.io/auth-keepalive-requests	nginx.ingress.kubernetes.io/global-rate-limit-key	nginx.ingress.kubernetes.io/session-cookie-name	string
nginx.ingress.kubernetes.io/auth-keepalive-timeout	nginx.ingress.kubernetes.io/global-rate-limit-ignored-cidr	nginx.ingress.kubernetes.io/session-cookie-path	string
nginx.ingress.kubernetes.io/auth-proxy-set-headers	nginx.ingress.kubernetes.io/permanent-redirect	nginx.ingress.kubernetes.io/session-cookie-domain	string
nginx.ingress.kubernetes.io/auth-snippet	nginx.ingress.kubernetes.io/permanent-redirect-code	nginx.ingress.kubernetes.io/session-cookie-change-on-failure	"true" or "false"
nginx.ingress.kubernetes.io/enable-global-auth	nginx.ingress.kubernetes.io/temporal-redirect	nginx.ingress.kubernetes.io/session-cookie-samesite	string
nginx.ingress.kubernetes.io/backend-protocol	nginx.ingress.kubernetes.io/preserve-trailing-slash	nginx.ingress.kubernetes.io/session-cookie-conditional-samesite-none	"true" or "false"
nginx.ingress.kubernetes.io/canary	nginx.ingress.kubernetes.io/proxy-body-size	nginx.ingress.kubernetes.io/ssl-redirect	"true" or "false"
nginx.ingress.kubernetes.io/canary-by-header	nginx.ingress.kubernetes.io/proxy-cookie-domain	nginx.ingress.kubernetes.io/ssl-passthrough	"true" or "false"
nginx.ingress.kubernetes.io/canary-by-header-value	nginx.ingress.kubernetes.io/proxy-cookie-path	nginx.ingress.kubernetes.io/stream-snippet	string
nginx.ingress.kubernetes.io/canary-by-header-patterr	nginx.ingress.kubernetes.io/proxy-connect-timeout	nginx.ingress.kubernetes.io/upstream-hash-by	string
nginx.ingress.kubernetes.io/canary-by-cookie	nginx.ingress.kubernetes.io/proxy-send-timeout	nginx.ingress.kubernetes.io/x-forwarded-prefix	string
nginx.ingress.kubernetes.io/canary-weight	nginx.ingress.kubernetes.io/proxy-read-timeout	nginx.ingress.kubernetes.io/load-balance	string
nginx.ingress.kubernetes.io/canary-weight-total	nginx.ingress.kubernetes.io/proxy-next-upstream	nginx.ingress.kubernetes.io/upstream-vhost	string
	nginx.ingress.kubernetes.io/proxy-next-upstream-timeout	nginx.ingress.kubernetes.io/denylist-source-range	CIDR
		nginx.ingress.kubernetes.io/whitelist-source-range	CIDR

不幸的是，使用Operator/云提供商/等。如果您希望保持供应商中立，注释并不总是一个好主意。

但是，有时它也是唯一的选择（例如，在使用 LoadBalancer 类型的服务时，将 AWS ALB 部署在正确的子网中）。

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nlb-sample-service
  namespace: nlb-sample-app
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: external
    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip
    service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing
spec:
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
  type: LoadBalancer
  selector:
    app: nginx
```

AWS EKS annotations for provisioning an NLB