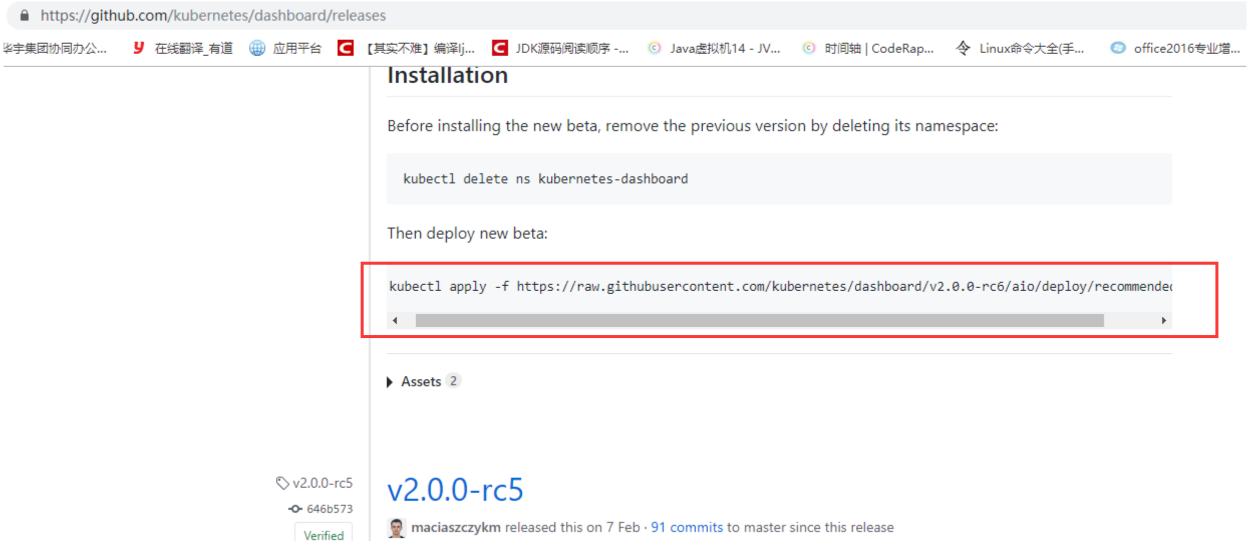
k8s-4、K8S+dashboard安装管理界面软件

笔记本:	<Inbox>		
创建时间:	2020/4/2 21:06	**更新时间:**	2020/4/3 13:46
作者:	王鹏		
URL:	https://www.cnblogs.com/weiBlog/p/10503779.html		

一、安装dashboard

（1）首先我们需要准备一个dashboard.yaml，首先访问dashboard的github官网，

```
https://github.com/kubernetes/dashboard/releases
```



Installation

Before installing the new beta, remove the previous version by deleting its namespace:

```
kubectl delete ns kubernetes-dashboard
```

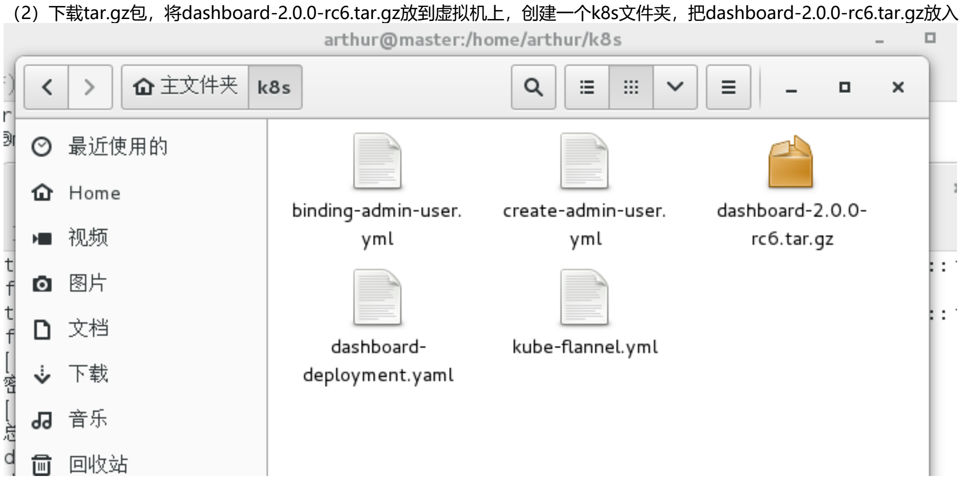
Then deploy new beta:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc6/aio/deploy/recommende
```

v2.0.0-rc5

maciaszczykm released this on 7 Feb · 91 commits to master since this release

（2）下载tar.gz包，将dashboard-2.0.0-rc6.tar.gz放到虚拟机上，创建一个k8s文件夹，把dashboard-2.0.0-rc6.tar.gz放入k8s文件夹中



（3）再浏览器复制连接打开yaml，复制yaml文件上的内容，复制到dashboard-deployment.yaml，并且把文件放到k8s文件夹中

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc6/aio/deploy/recommended.yaml
```

dashboard-deployment(v2.0.0-rc6)版本的yaml文件内容，标红色的注意地方

```
# Copyright 2017 The Kubernetes Authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.


apiVersion: v1
kind: Namespace
metadata:
  name: kubernetes-dashboard


---


apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard


---


kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  type: NodePort
  ports:
    - nodePort: 30888
      port: 443
      targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard


---


apiVersion: v1
kind: Secret
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard-certs
  namespace: kubernetes-dashboard
type: Opaque


---


apiVersion: v1
kind: Secret
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard-csrf
  namespace: kubernetes-dashboard
type: Opaque
data:
  csrf: ""


---


apiVersion: v1
kind: Secret
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard-key-holder
  namespace: kubernetes-dashboard
type: Opaque


---


kind: ConfigMap
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard-settings
  namespace: kubernetes-dashboard


---


kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
rules:
  # Allow Dashboard to get, update and delete Dashboard exclusive secrets.
  - apiGroups: [""]
    resources: ["secrets"]
    resourceNames: ["kubernetes-dashboard-key-holder", "kubernetes-dashboard-certs", "kubernetes-dashboard-csrf"]
    verbs: ["get", "update", "delete"]
    # Allow Dashboard to get and update 'kubernetes-dashboard-settings' config map.
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["kubernetes-dashboard-settings"]
    verbs: ["get", "update"]
    # Allow Dashboard to get metrics.
  - apiGroups: [""]
    resources: ["services"]
    resourceNames: ["heapster", "dashboard-metrics-scraper"]
    verbs: ["proxy"]
  - apiGroups: [""]
    resources: ["services/proxy"]
    resourceNames: ["heapster", "http:heapster:", "https:heapster:", "dashboard-metrics-scraper", "http:dashboard-metrics-scraper"]
    verbs: ["get"]


---


kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
rules:
  # Allow Metrics Scraper to get metrics from the Metrics server
  - apiGroups: ["metrics.k8s.io"]
    resources: ["pods", "nodes"]
    verbs: ["get", "list", "watch"]


---


apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: kubernetes-dashboard
subjects:
  - kind: ServiceAccount
    name: kubernetes-dashboard
    namespace: kubernetes-dashboard


---


apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: kubernetes-dashboard
subjects:
  - kind: ServiceAccount
    name: kubernetes-dashboard
    namespace: kubernetes-dashboard


---


kind: Deployment
apiVersion: apps/v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      k8s-app: kubernetes-dashboard
  template:
    metadata:
      labels:
        k8s-app: kubernetes-dashboard
    spec:
      containers:
        - name: kubernetes-dashboard
          #此处可以修改为自己可以访问的镜像地址
          image: kubernetesui/dashboard:v2.0.0-rc6
          imagePullPolicy: Always
          ports:
            - containerPort: 8443
              protocol: TCP
          args:
            - --auto-generate-certificates
            - --namespace=kubernetes-dashboard
            # Uncomment the following line to manually specify Kubernetes API server Host
            # If not specified, Dashboard will attempt to auto discover the API server and connect
            # to it. Uncomment only if the default does not work.
            # - --apiserver-host=http://my-address:port
          volumeMounts:
            - name: kubernetes-dashboard-certs
              mountPath: /certs
              # Create on-disk volume to store exec logs
            - mountPath: /tmp
              name: tmp-volume
          livenessProbe:
            httpGet:
              scheme: HTTPS
              path: /
              port: 8443
            initialDelaySeconds: 30
            timeoutSeconds: 30
          securityContext:
            allowPrivilegeEscalation: false
            readOnlyRootFilesystem: true
            runAsUser: 1001
            runAsGroup: 2001
      volumes:
        - name: kubernetes-dashboard-certs
          secret:
            secretName: kubernetes-dashboard-certs
        - name: tmp-volume
          emptyDir: {}
      serviceAccountName: kubernetes-dashboard
      nodeSelector:
        "beta.kubernetes.io/os": linux
      # Comment the following tolerations if Dashboard must not be deployed on master
      tolerations:
        - key: node-role.kubernetes.io/master
          effect: NoSchedule


---


kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: dashboard-metrics-scraper
  name: dashboard-metrics-scraper
  namespace: kubernetes-dashboard
spec:

  ports:
      port: 8000
      targetPort: 8000
  selector:
    k8s-app: dashboard-metrics-scraper


---


kind: Deployment
apiVersion: apps/v1
metadata:
  labels:
    k8s-app: dashboard-metrics-scraper
  name: dashboard-metrics-scraper
  namespace: kubernetes-dashboard
spec:
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      k8s-app: dashboard-metrics-scraper
  template:
    metadata:
      labels:
        k8s-app: dashboard-metrics-scraper
      annotations:
        seccomp.security.alpha.kubernetes.io/pod: 'runtime/default'
    spec:
      containers:
        - name: dashboard-metrics-scraper
          image: kubernetesui/metrics-scraper:v1.0.3
          ports:
            - containerPort: 8000
              protocol: TCP
          livenessProbe:
            httpGet:
              scheme: HTTP
              path: /
              port: 8000
            initialDelaySeconds: 30
            timeoutSeconds: 30
          volumeMounts:
          - mountPath: /tmp
            name: tmp-volume
          securityContext:
            allowPrivilegeEscalation: false
            readOnlyRootFilesystem: true
            runAsUser: 1001
            runAsGroup: 2001
      serviceAccountName: kubernetes-dashboard
      nodeSelector:
        "beta.kubernetes.io/os": linux
      # Comment the following tolerations if Dashboard must not be deployed on master
      tolerations:
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      volumes:
        - name: tmp-volume
          emptyDir: {}
```

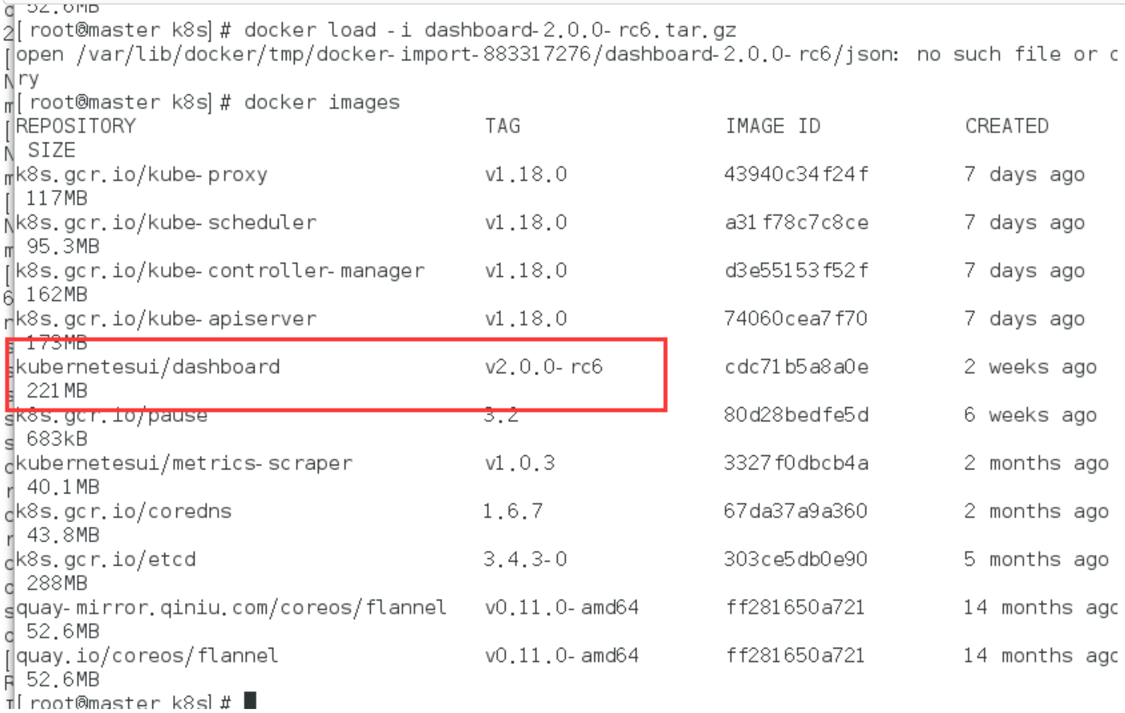
介绍下三种port：
nodePort：实际物理机上的端口，供外部访问
port： service端口，访问nodePort会被代理到service端口
targetPort: pod端口
client访问：nodeIP:port->serviceIP:port->podIP:port

介绍下imagePullPolicy

```
#总是拉取镜像
imagePullPolicy: Always
#默认值，本地有则使用本地镜像，不拉取
imagePullPolicy: IfNotPresent
#只使用本地镜像，从不拉取
imagePullPolicy: Never
```

（4）把dashboard-2.0.0-rc6.tar.gz加载到docker 镜像中

```
[root@master k8s]# docker load -i dashboard-2.0.0-rc6.tar.gz
```



```
 52.6MB
[root@master k8s]# docker load -i dashboard-2.0.0-rc6.tar.gz
open /var/lib/docker/tmp/docker-import-883317276/dashboard-2.0.0-rc6/json: no such file or c
ry
[root@master k8s]# docker images
REPOSITORY                           TAG                 IMAGE ID            CREATED
 SIZE
k8s.gcr.io/kube-proxy                v1.18.0             43940c34f24f        7 days ago
 117MB
k8s.gcr.io/kube-scheduler            v1.18.0             a31f78c7c8ce        7 days ago
 95.3MB
k8s.gcr.io/kube-controller-manager   v1.18.0             d3e55153f52f        7 days ago
 162MB
k8s.gcr.io/kube-apiserver            v1.18.0             74060cea7f70        7 days ago
 173MB
kubernetesui/dashboard               v2.0.0-rc6          cdc71b5a8a0e        2 weeks ago
 221MB
k8s.gcr.io/pause                     3.2                 80d28bedfe5d        6 weeks ago
 683kB
kubernetesui/metrics-scraper         v1.0.3              3327f0dbcb4a        2 months ago
 40.1MB
k8s.gcr.io/coredns                   1.6.7               67da37a9a360        2 months ago
 43.8MB
k8s.gcr.io/etcd                      3.4.3-0             303ce5db0e90        5 months ago
 288MB
quay-mirror.qiniu.com/coreos/flannel v0.11.0-amd64       ff281650a721        14 months agc
 52.6MB
quay.io/coreos/flannel               v0.11.0-amd64       ff281650a721        14 months agc
 52.6MB
[root@master k8s]# █
```

（5）或者执行加载yaml文件命令

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc6/aio/deploy/recommended.yaml
```

（6） 然后我们还需要一个用户权限的user.yaml，它用来绑定角色权限：文件放到k8s文件夹上

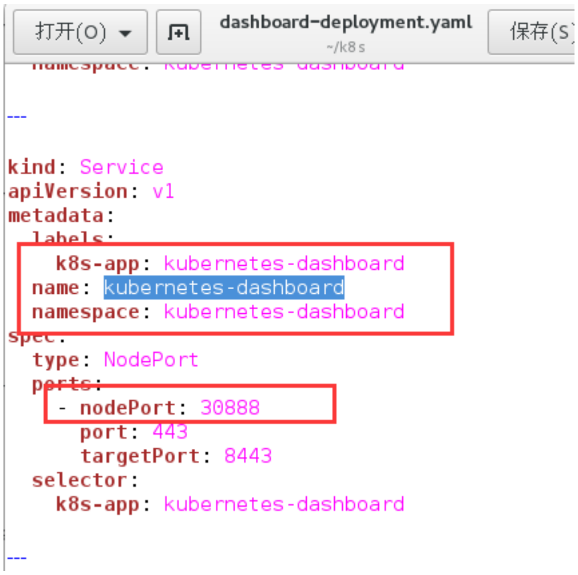
```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: kubernetes-dashboard
  labels:
    k8s-app: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: default
  namespace: default
```

（7）运行user.yaml

```
kubectl apply -f user.yaml
```

(8) 查看dashborad的端口：找到yaml中的 namedata 和端口号

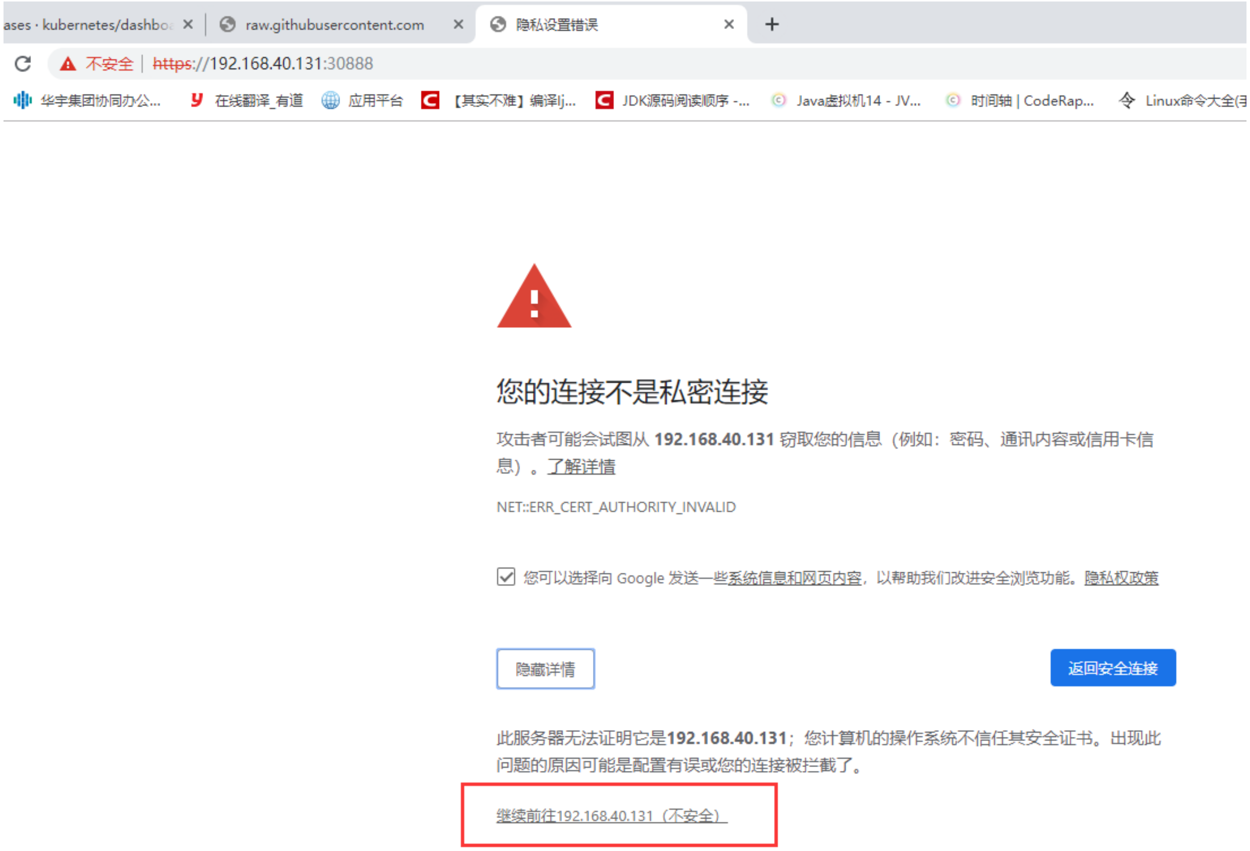
```
# kubectl get svc -n [yaml 中 metadata]
kubectl get svc -n kubernetes-dashboard
```



```
  namespace: kubernetes-dashboard

---

kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  type: NodePort
  ports:
    - nodePort: 30888
      port: 443
      targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard

---
```

（9）输入dashborad的网址：网址地址为https:// + 本地ip + 服务设置的端口号

必须要添加https，页面会被阻止，然后点击高级，点击继续访问。



您的连接不是私密连接

攻击者可能会试图从 **192.168.40.131** 窃取您的信息（例如：密码、通讯内容或信用卡信息）。了解详情

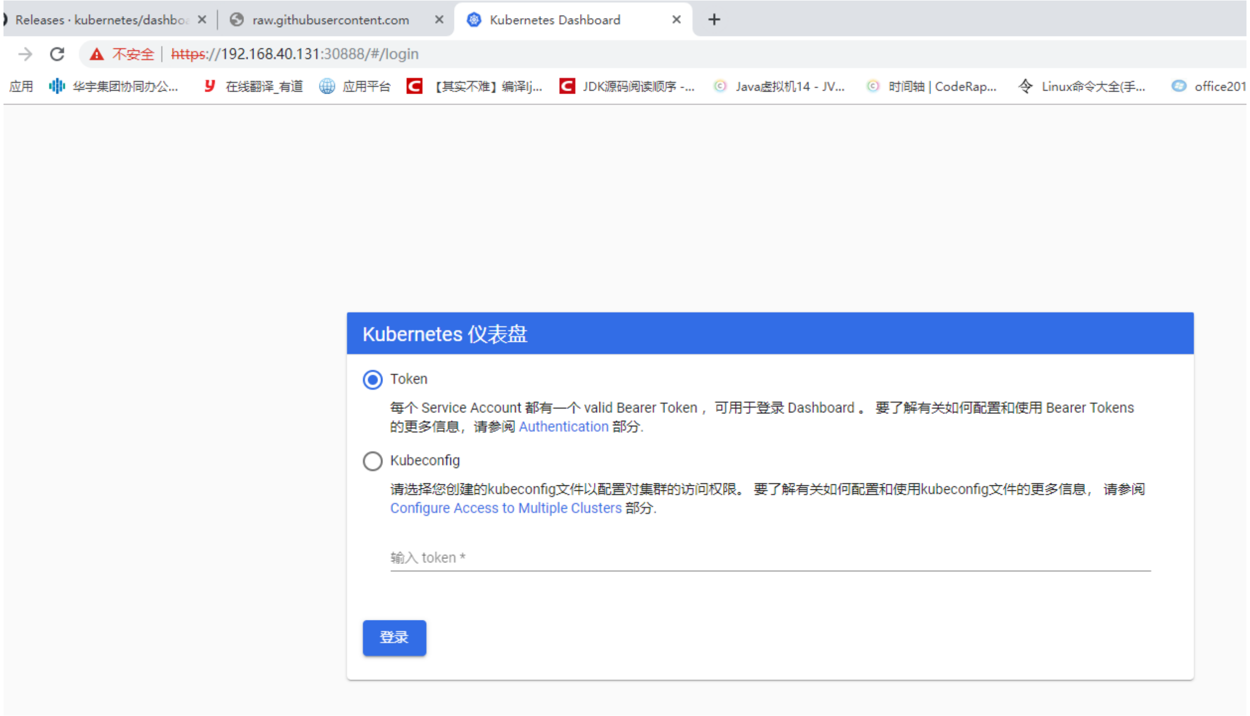
NET::ERR_CERT_AUTHORITY_INVALID

☑ 您可以选择向 Google 发送一些系统信息和网页内容，以帮助我们改进安全浏览功能。隐私权政策

此服务器无法证明它是**192.168.40.131**；您计算机的操作系统不信任其安全证书。出现此问题的原因可能是配置有误或您的连接被拦截了。

继续前往192.168.40.131（不安全）

（10）进入到登录界面



Kubernetes 仪表盘

● Token

每个 Service Account 都有一个 valid Bearer Token ，可用于登录 Dashboard 。要了解有关如何配置和使用 Bearer Tokens 的更多信息，请参阅 Authentication 部分.

○ Kubeconfig

请选择您创建的kubeconfig文件以配置对集群的访问权限。要了解有关如何配置和使用kubeconfig文件的更多信息，请参阅 Configure Access to Multiple Clusters 部分.

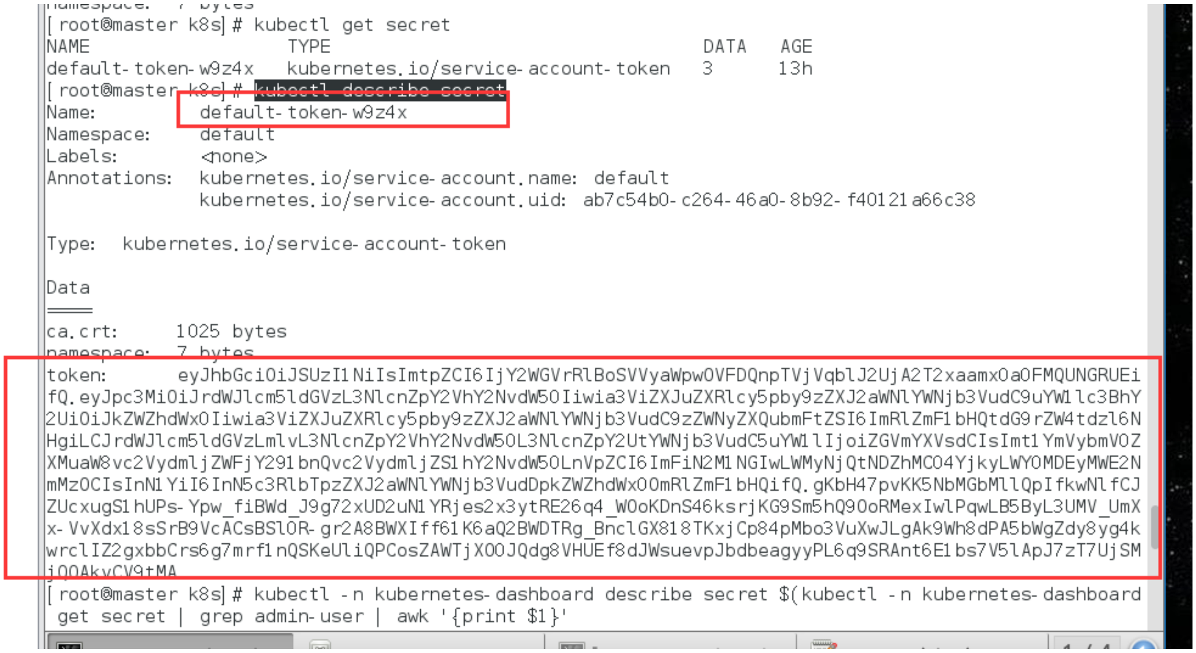
输入 token *

登录

（11）配置登录用户，通过命令获取令牌信息

```
kubectl get secret
或者
kubectl describe secret default-token-w9z4x
```

将token保存起来。此默认用户权限较低。



```
namespace:   7 bytes
[root@master k8s]# kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-w9z4x   kubernetes.io/service-account-token   3      13h
[root@master k8s]# kubectl describe secret
Name:         default-token-w9z4x
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: default
              kubernetes.io/service-account.uid: ab7c54b0-c264-46a0-8b92-f40121a66c38

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1025 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IjY2WGVrRlBoSVVyaWpwOVFDQnpTVjVqblJ2UjA2T2xaamx0a0FMQUNGRUEi
fQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY
2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImRlZmF1bHQtdG9rZW4tdzl6N
HgiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoiZGVmYXVsdCIsImt1YmVybmV0Z
XMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6ImFiN2M1NGIwLWMyNjQtNDZhMC04YjkyLWY0MDEyMWE2N
mMzOCIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmRlZmF1bHQifQ.gKbH47pvKK5NbMGbMllQpIfkwNlfCJ
ZUcxugS1hUPs-Ypw_fiBWd_J9g72xUD2uN1YRjes2x3ytRE26q4_WOoKDnS46ksrjKG9Sm5hQ9OoRMexIwlPqwLB5ByL3UMV_UmX
x-VvXdx18sSrB9VcACsBSl0R-gr2A8BWXIff61K6aQ2BWDTRg_BnclGX818TKxjCp84pMbo3VuXwJLgAk9Wh8dPA5bWgZdy8yg4k
wrclIZ2gxbbCrs6g7mrf1nQSKeUliQPCosZAWTjXO0JQdg8VHUEf8dJWsuevpJbdbeagyyPL6q9SRAnt6E1bs7V5lApJ7zT7UjSM
jQ0AkyCV9tMA
[root@master k8s]# kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard
 get secret | grep admin-user | awk '{print $1}'
```

用于用户登录的token：

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjY2WGVrRlBoSVVyaWpwOVFDQnpTVjVqblJ2UjA2T2xaamx0a0FMQUNGRUEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5p
Ypw_fiBWd_J9g72xUD2uN1YRjes2x3ytRE26q4_WOoKDnS46ksrjKG9Sm5hQ9OoRMexIwlPqwLB5ByL3UMV_UmXx-VvXdx18sSrB9VcACsBSl0R-
gr2A8BWXIff61K6aQ2BWDTRg_BnclGX818TKxjCp84pMbo3VuXwJLgAk9Wh8dPA5bWgZdy8yg4kwrclIZ2gxbbCrs6g7mrf1nQSKeUliQPCosZAWTjXO0JQdg8VHUEf8dJWsuevpJbdbeagyyPL6q9SRAnt
```

创建一个用户权限比较高的用户：就是上面第（6）（7）步。然后执行查询用户token操作

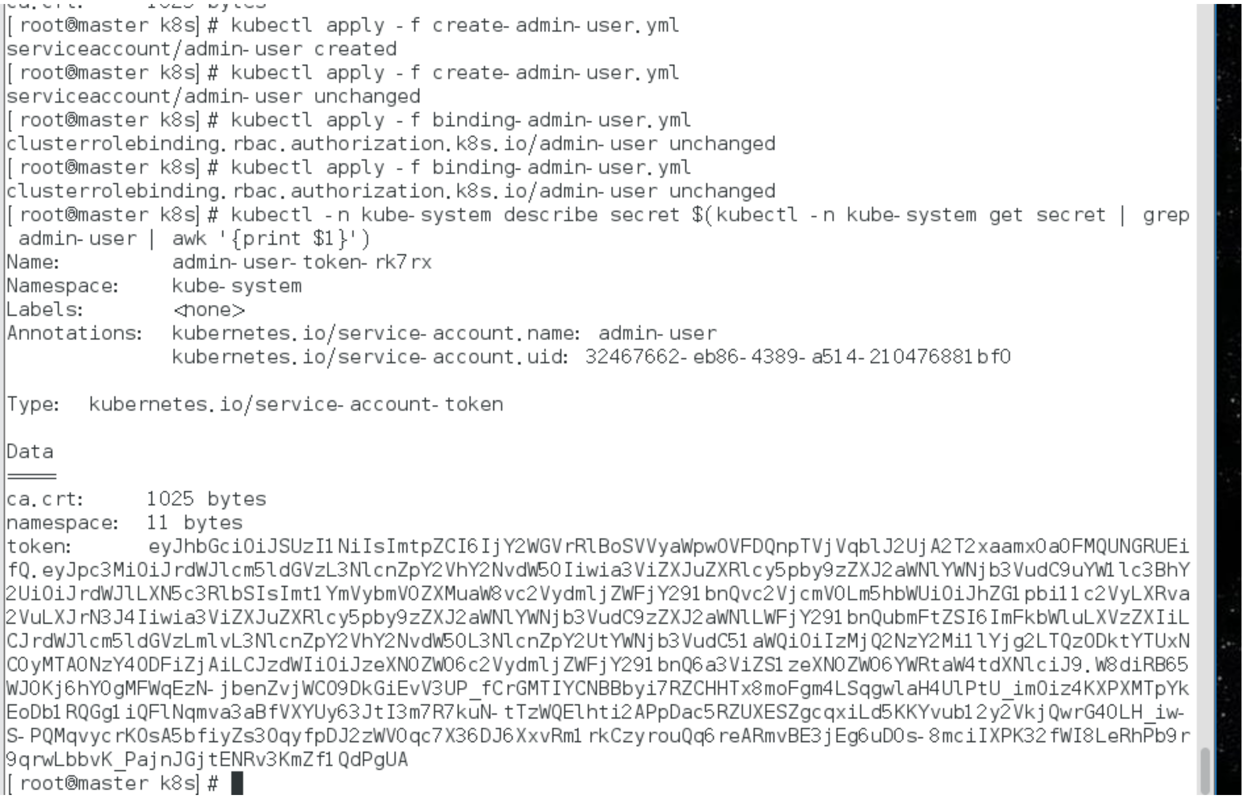
创建一个文件create-admin-user.yml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kube-system
```

创建一个文件binding-admin-user.yml

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: admin-user
  namespace: kube-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kube-system
```

```
#红色部分代表空间（下一章讲解namespace），admin-user代码哪个用户
kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user | awk '{print $1}')
```

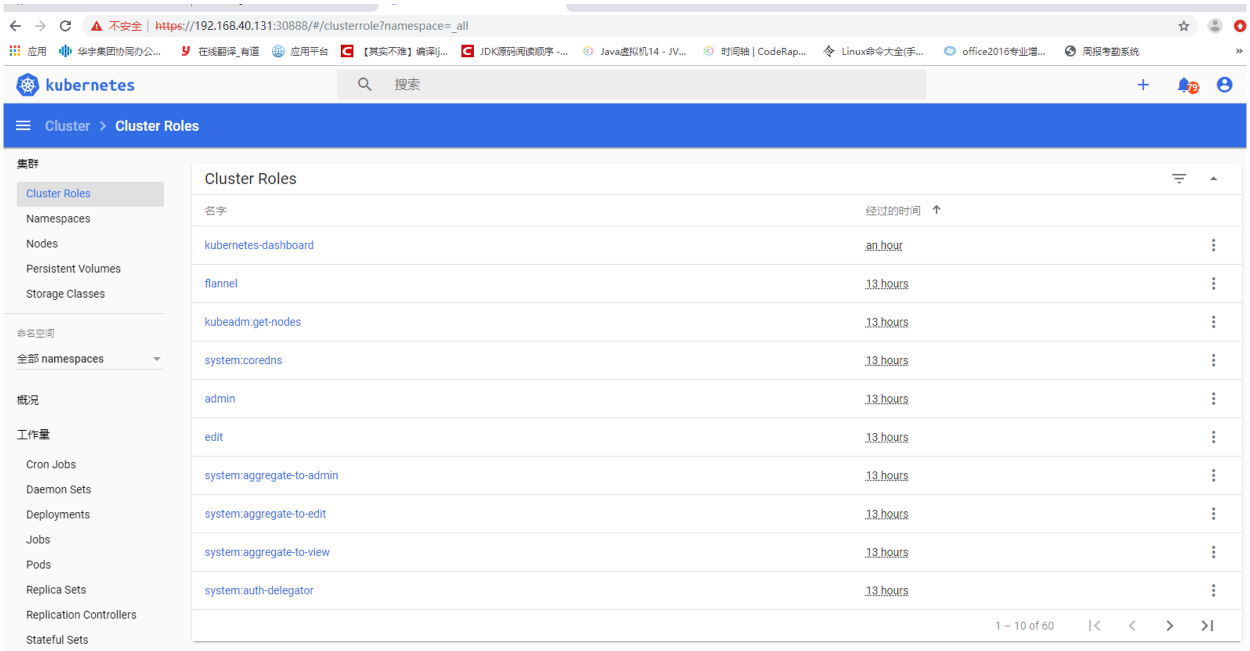


```
ca.crt:      1025 bytes
[root@master k8s]# kubectl apply -f create-admin-user.yml
serviceaccount/admin-user created
[root@master k8s]# kubectl apply -f create-admin-user.yml
serviceaccount/admin-user unchanged
[root@master k8s]# kubectl apply -f binding-admin-user.yml
clusterrolebinding.rbac.authorization.k8s.io/admin-user unchanged
[root@master k8s]# kubectl apply -f binding-admin-user.yml
clusterrolebinding.rbac.authorization.k8s.io/admin-user unchanged
[root@master k8s]# kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep
 admin-user | awk '{print $1}')
Name:         admin-user-token-rk7rx
Namespace:    kube-system
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 32467662-eb86-4389-a514-210476881bf0

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1025 bytes
namespace:  11 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IjY2WGVrRlBoSVVyaWpwOVFDQnpTVjVqblJ2UjA2T2xaamx0a0FMQUNGRUEi
fQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY
2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva
2VuLXJrN3J4Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiL
CJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiIzMjQ2NzY2Mi1lYjg2LTQzODktYTUxN
C0yMTA0NzY4ODFiZjAiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.W8diRB65
WJ0Kj6hY0gMFWqEzN-jbenZvjWCO9DkGiEvV3UP_fCrGMTIYCNBByi7RZCHHTx8moFgm4LSqgwlaH4UlPtU_im0iz4KXPXMTpYk
EoDb1RQGg1iQFlNqmva3aBfVXYUy63JtI3m7R7kuN-tTzWQElhti2APpDac5RZUXESZgcqxiLd5KKYvub12y2VkjQwrG40LH_iw-
S-PQMqvycrKOsA5bfiyZs30qyfpDJ2zWV0qc7X36DJ6XxvRm1rkCzyrouQq6reARmvBE3jEg6uD0s-8mciIXPK32fWI8LeRhPb9r
9qrwLbbvK_PajnJGjtENRv3KmZf1QdPgUA
[root@master k8s]# 
```

保存admin-user的token：

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjY2WGVrRlBoSVVyaWpwOVFDQnpTVjVqblJ2UjA2T2xaamx0a0FMQUNGRUEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5p
jbenZvjWCO9DkGiEvV3UP_fCrGMTIYCNBByi7RZCHHTx8moFgm4LSqgwlaH4UlPtU_imOiz4KXPXMTpYkEoDb1RQGg1iQFlNqmva3aBfVXYUy63JtI3m7R7kuN-
tTzWQElhti2APpDac5RZUXESZgcqxiLd5KKYvub12y2VkjQwrG40LH_iw-S-PQMqvycrKOsA5bfiyZs3OqyfpDJ2zWV0qc7X36DJ6XxvRm1rkCzyrouQq6reARmvBE3jEg6uD0s-
8mciIXPK32fWI8LeRhPb9r9qrwLbbvK_PajnJGjtENRv3KmZf1QdPgUA
```

首页



Cluster Roles

名字	经过的时间 ↑
kubernetes-dashboard	an hour
flannel	13 hours
kubeadm:get-nodes	13 hours
system:coredns	13 hours
admin	13 hours
edit	13 hours
system:aggregate-to-admin	13 hours
system:aggregate-to-edit	13 hours
system:aggregate-to-view	13 hours
system:auth-delegator	13 hours

1 – 10 of 60