

目前所有的访问都已经转向https了，大势所趋，ssl重要性这里不细说了；我这里是client到traefik加密，后端还是http，有更高要求的时候再来进步优化，先满足功能再说。

*traefik http部署我这里省略，详细可参考[kubeasz](#)中的ingress部分。*

## 流程示意图

```
client ---https---> traefik ---http---> svc (本文)
client ---https---> traefik ---https---> svc
```

## 一、生成证书

我这里使用私签证书,CN是域名，根据实际需求填写

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=tr
```

## 二、创建一个configmap，保存traefix的配置。这里的traefix中配置了把所有http请求全部rewrite为https的规则，并配置相应的证书位置

### 1、配置证书traefik-cert cm

```
kubectl -n kube-system create secret tls traefik-ingress-controller-default-cer
```

### 2、查看证书并且导入到yaml文件

```
kubectl get secret traefik-ingress-controller-default-cert -o yaml >> secret.yaml
```

### 3、复制tls.crt和tls.key的值到traefik-ingress.yaml

```
[root@master ssl]# vim secret.yaml

tls.key: LS0tLS1CRUdJTiBQdWUwVWVFIETfWS0tLS0tCk1JSUV2Z0lCQURBTklna3Foa2lH0XcwQkFRRUZhBQVNDQ
QzhQSwpLQTLJSkYxam02Zn1HaFdaeFZHa1RyS1pNeEdhdUhhjSwtEUTY1WnE5ZC9odnVQSG9QbjJRazBQRAo4VWJ6MURC
kbFIzT0V4ZDlQSTA0ZEZJUDJIdlY2Cl1t0FJXUXZ1N2xKQVYVZ3h5cXV0RjdoZj16YmowM1dUQ3V0UHhYd2xJKzJjRUN
JGcUwyem1Mc0ZUV3BiK3ZHS0ppT2hEQmhwc9VZjRmNnpKTXXVSKtaV2hnU21VWQprczdJZmNiUlZmS2hMjd0U056RW
jhkcVp1bEVDcnc1M25Lb3piQWdN0kFBRUNnZ0VBV2gyNkhWbG84R0tveXV5TThxaFRKcVR5TlU1VnppQS96bzVDM1pXS
QWowChpzcxXfXQ2UweFhQSHBVWGE2T3YrdktKaStZWERhWQpreMNXMVZ3Q3hpV3hkRDZPUVBpcE5xbH16SFJtN2kyUDF5
PalVnV3FxyK5wWUNiditJZX13bXR6TE1qeisyVXN5TWt1VEJUYkhTZWR1YWhGVmxWZVJ0dFYwdHMwMUQKTEsSR25zR1h
FqRnpBZVg1bVh5R2dnNWN0K2N4cgpXTVZYa3lGQ21KbGY3SUZBUhJ4ejRB0G90NWFmNTNCK3N5MERtN1NZR1FLQmdRRQ
2w3YWN0VWFTUFDWcm9nbFZRZk1EWjBRWk1xMktxdWtXaDUyQ1ErSEg5TG1GMXMKQ2taUjJhRE9CRUdTamNvZ2Z0NzhyV
ZUNpWStBZgpTN3FmVlVIQz1MRUdBRCtVTLBxbWfZ1Yvd0tCZ1FEU0NXMm1lU1MNmZaTC9ZemNEdE9XUmdnWjU5VTVV
PenhRNTZzUUprRUx0ZmFFNDk4TVV1cE1zazhNeGY5UESKQ1NUcFU2WStvSwpFVDBTd010c3NQczc0VXg3aVVuWm1kUFd
dhaEpRS0JnRU44Wkh6bUhrUUxIV1oxZE1CMUIxVTR0NjRxMTQ5NlQ1N1h6eWRuR2xXZ1VMT1p1RldSCmZvYzY4cmdtZH
kNoTHZ0aGZhdnFPaUZvY1E5WFQKR3JJ0UpKMkRwTXBLW1BHcz1jbFZPVk1XRmRKeHRzNHQ4T0dWNGVrRFF4NEZ4TWFBY
d1pTc1NIUlprUHlDYWFvNVNHaUQrcmptbjBSaDJpRnJ1aWVINGtTLzNlMEhpCi9Qb21Nank1aVhZVktTG4vRWp6MnVO
IR2cvKzQKbUp0SnZCNlUv0S9ZckQ4eGZ0SWJXSz1VaWgzWEppVUtMVG15V1dyQkFvR0JBTWtDbDdmZms2MkpSSGVYQ1M
BnUk5UUj1ub0JsQ2JiT29yWm91cXlTG5oYVY4bWdlCm1lQUG5S2V2Z2g4VWJWRXd0L3lxdUUzaUlPVmZFR1AwektZYm
1RbYwMTNF04b1RmN2tFUUhhbCj0tLS0tRU5FEFSSVZPVEUcS0VZLS0tLS0t
```

#### 4、配置https重定向配置文件

注意:如果修改了cmp配置，需要停止pod重新启动才生效

```
kubect1 delete -f traefik-ingress.yaml
```

配置文件内容如下:

```
[root@by-deploy01 ingress]# cat traefik.yaml1
```

```
kind: ConfigMap
```

```
metadata:
```

```
  name: traefik-ingress-controller-config-map
```

```
  namespace: kube-system
```

```
apiVersion: v1
```

```
data:
```

```
  traefik.toml: |+
```

```
    logLevel = "INFO"
```

```
    defaultEntryPoints = ["http","https"]
```

```
    [entryPoints]
```

```
      [entryPoints.http]
```

```
        address = ":80"
```

```
        [entryPoints.http.redirect]
```

```
          entryPoint = "https"
```

```
      [entryPoints.https]
```

```
        address = ":443"
```

```
        [entryPoints.https.tls]
```

```
          [[entryPoints.https.tls.certificates]]
```

```
            certFile = "/ssl/tls.crt"
```

```
            keyFile = "/ssl/tls.key"
```

```
      [entryPoints.traefik]
```

```
        address = ":8080"
```

```
[entryPoints.traefik]
  address = ":8000"
[kubernetes]
[traefikLog]
  format = "json"
[api]
  entryPoint = "traefik"
  dashboard = true
```

##其中tls.crt和tls.key就是证书文件，注意必须要改为这个文件名。

# 配置traefik-conf cm

```
kubectl apply -f traefik.yaml
```

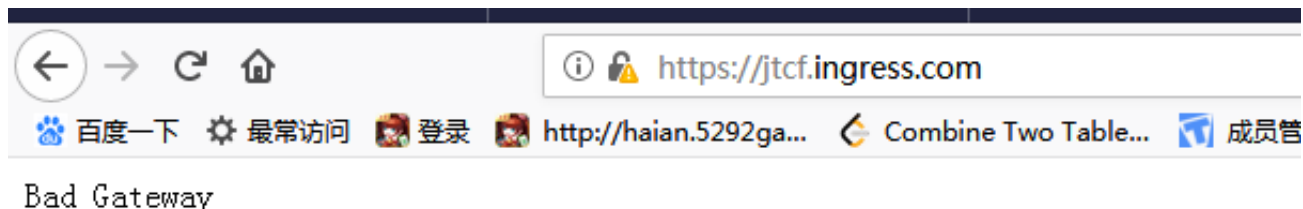
如果是toml文件就用下列命令：

```
kubectl create configmap traefik-conf --from-file=traefik.toml -n kube-system
```

##查看

```
kubectl get cm -n kube-system
```

的在traefik.toml里面定义了端口重定向，如果没设置对应端口https重定向无效，如下



图：

注意：配置文件的目录和证书的目录不能一样

## deploy引用

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: traefik-ingress-controller
```

```
namespace: kube-system
labels:
  k8s-app: traefik-ingress-lb
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: traefik-ingress-lb
  template:
    metadata:
      labels:
        k8s-app: traefik-ingress-lb
        name: traefik-ingress-lb
    spec:
      serviceAccountName: traefik-ingress-controller
      terminationGracePeriodSeconds: 60
      containers:
        - image: traefik:v1.7.2
          imagePullPolicy: IfNotPresent
          name: traefik-ingress-lb
```

#修改了之前的内容

```
    volumeMounts:
      - mountPath: /config
        name: config
      - mountPath: /ssl
        name: ssl
      - mountPath: /etc/localtime
        name: time
    ports:
      - name: http
        containerPort: 80
        hostPort: 80
        protocol: TCP
      - name: https
        containerPort: 443
        hostPort: 443
        protocol: TCP
      - name: traefik-web
        containerPort: 8080
        hostPort: 8080
```

```

    protocol: TCP
  args:
    - --configfile=/config/traefik.toml
  volumes:
    - name: config
      configMap:
        name: traefik-ingress-controller-config-map
    - name: ssl
      secret:
        secretName: traefik-ingress-controller-default-cert
    - name: time
      hostPath:
        path: /etc/localtime

```

# 修改结束

---

kind: Service

.....

# 生效

```
kubectl apply -f traefik_deploy.yaml
```

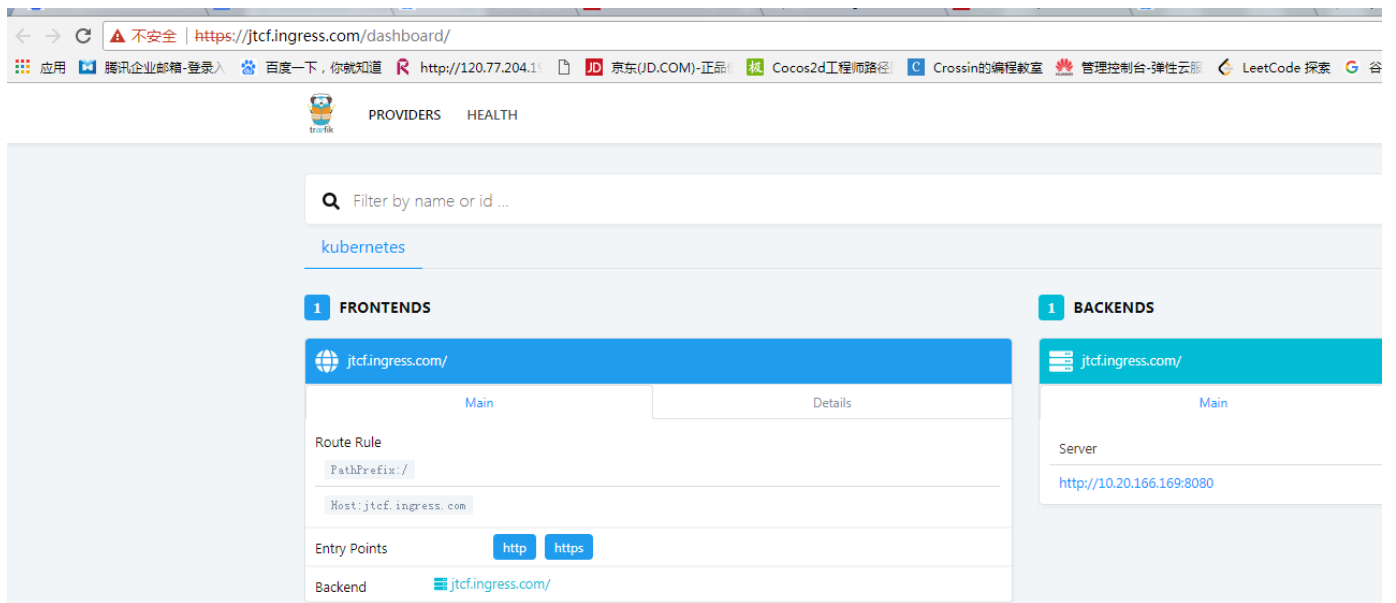
至此配置完成可以用浏览器测试下！

最后我们来测试下是否成功，这里我们可以登陆traefik-ui界面，可以看到原本http的访问，traefik会直接给我们重定向至https。

输入jtcf.ingress.com::23456



自动重定向到https



注意：如果在ingress上开启了tls，那么证书必须与域名相关

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: traefik-web-ui
  namespace: kube-system
spec:
  tls:
  - secretName: traefik-ingress-controller-default-cert
  rules:
  - host: jtcf.ingress.com
    http:
      paths:
      - path: /
        backend:
          serviceName: traefik-ingress-service
          servicePort: 8080
[root@master ss11#
```

在ingress上不开启tls则使用的是默认证书，下面是注释了tls

```

kind: Ingress
metadata:
  name: my-ingress
# namespace: kube-system
spec:
#  tls:
# - secretName: traefik-ingress-controller
rules:
- host: jtcf.nginx.com
  http:
    paths:
    - path: /
      backend:
        serviceName: nginx
        servicePort: 8000
- host: jtcf.tomcat.com
  http:
    paths:
    - path: /
      backend:
        serviceName: myweb
        servicePort: 8080

```

**证书必须和ingress在一个命名空间，不然无法访问，报错如下：比如我们证书是在kube-system**

```

{"level": "error", "msg": "Error configuring TLS for ingress default/my-ingress: secret default/traefik-ingress-controller-default-cert does not exist", "time": "2018-10-26T14:09:07+08:00"}
{"level": "error", "msg": "Error configuring TLS for ingress default/my-ingress: secret default/traefik-ingress-controller-default-cert does not exist", "time": "2018-10-26T14:09:08+08:00"}
{"level": "error", "msg": "Error configuring TLS for ingress default/my-ingress: secret default/traefik-ingress-controller-default-cert does not exist", "time": "2018-10-26T14:09:09+08:00"}

```

Error configuring TLS for ingress default/my-ingress: secret **default**/traefik-ingress-controller-default-cert does not exist

单独创建证书只需要执行：

```
kubectl -n default create secret tls nginx-secret --key=tls.key --cert=tls.crt
```

创建好之后直接在ingress填写证书名称就可以使用

下面是我单独创建了一根default空间的证书给Ingress使用

```

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
spec:
  tls:
  - secretName: nginx-secret
  rules:
  - host: jtcf.nginx.com
    http:
      paths:
      - path: /
        backend:
          serviceName: nginx
          servicePort: 8000
  - host: jtcf.tomcat.com
    http:
      paths:
      - path: /
        backend:
          serviceName: myweb
          servicePort: 8080

```

或者把证书创建与ingress放一个yaml：

Secret:

```

[root@master ~]# cat ingress-yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: my-ingress-secret
  namespace: default
type: Opaque
data:
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk
VEFsaFlnU1V3RXdZRFZRUUhEQXhFWldaaGRXeDBJRU5wZEhre
reE1ESTENRE0wTlRBMTdqOktNNlXN3CkNRWURWUVEHRXdKwVdE

```

ingress：



```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
spec:
  tls:
    - secretName: my-ingress-secret
  rules:
    - host: jtcf.nginx.com
      http:
        paths:
          - path: /
            backend:
              serviceName: nginx
              servicePort: 8000
    - host: jtcf.tomcat.com
      http:
        paths:
          - path: /
            backend:
              serviceName: myweb
              servicePort: 8080
```