TLS

Subtitle

2022/10/05

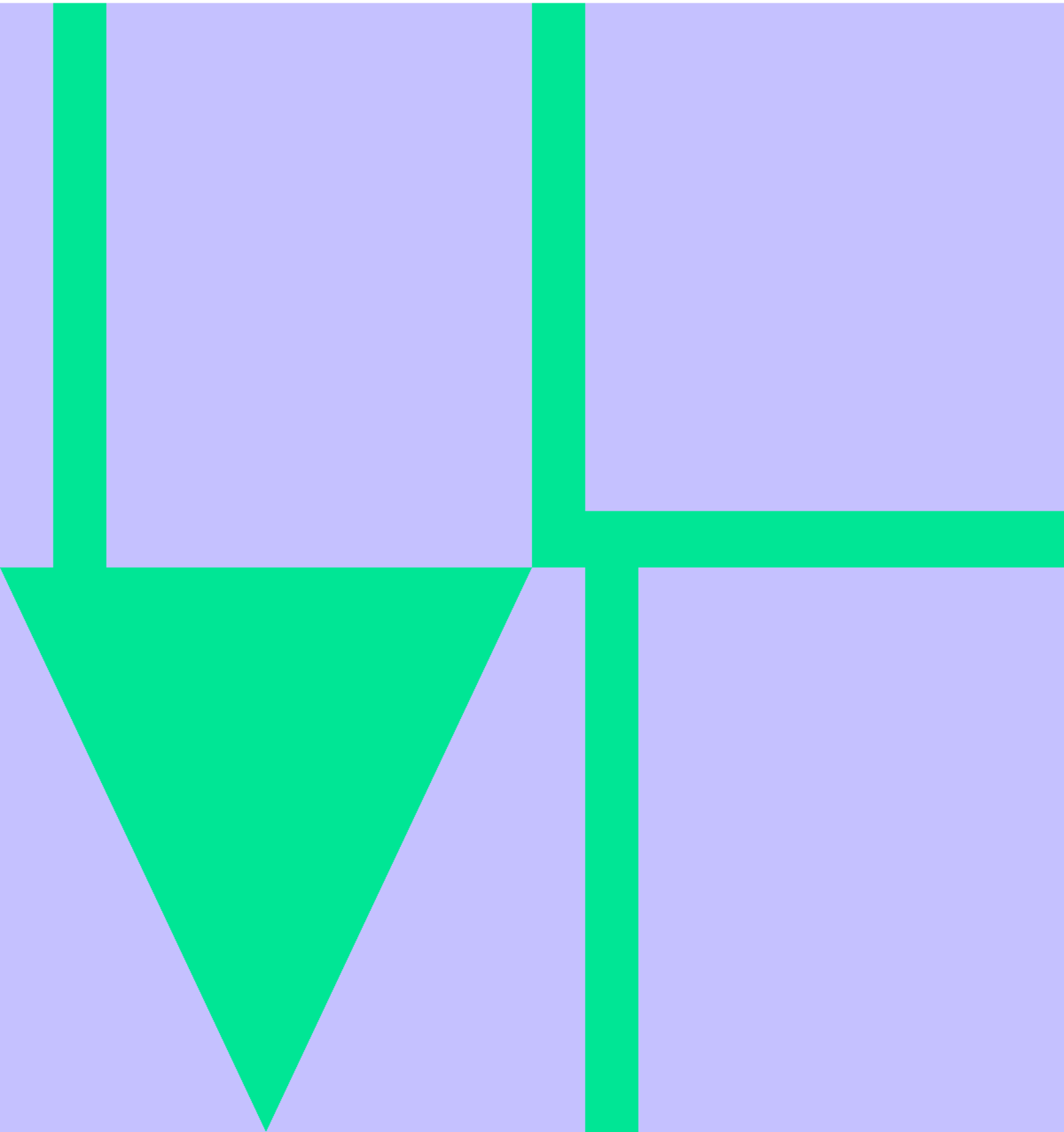

Table of Contents

TLS

Links

- https://mritd.me/2018/01/07/kubernetes-tls-bootstrapping-note/

TLS bootstrapping

需要重复强调一个问题是: TLS bootstrapping 时的证书实际是由 kube-controller-manager 组件来签署的，也就是说证书有效期是 kube-controller-manager 组件控制的；所以在 1.7 版本以后(我查文档发现的从1.7开始有) kube-controller-manager 组件提供了一个 – experimental-cluster-signing-duration 参数来设置签署的证书有效时间；默认为 8760h0m0s，将其改为 87600h0m0s 即 10 年后再进行 TLS bootstrapping 签署证书即可。

TLS bootstrapping only sets up api client certificates for the kubelet currently. If you want a serving cert for the kubelet that is signed by the apiserver's –kubelet-certificate-authority you must provide it. Otherwise the kubelet generates a self-signed serving cert.

https://github.com/kubernetes/kubernetes/issues/63164

```
[root@k8s-master-etcd.10-12-3-12 cert]# openssl verify -CAfile ca.pem kubelet.crt
kubelet.crt: CN = 10.12.3.12@1535266843
error 20 at 0 depth lookup:unable to get local issuer certificate
[root@k8s-master-etcd.10-12-3-12 cert]#
[root@k8s-master-etcd.10-12-3-12 cert]# openssl verify -CAfile ca.pem kubelet-client-current.pem
kubelet-client-current.pem: OK
```

由于TLS bootstrapping生成的kubelet的服务端证书（serving cert）是一个自签名证书，metrics-server 访问kubelet 10250端口时，会在日志中显示“ x509: certificate signed by unknown authority ”。因此需要设置metrics-server参数&insecure=true

证书生成

使用 cfssl 工具

CA命令

snippet.bash

```
#cfssl gencert -initca ca-csr.json |cfssljson -bare ca
```

生成命令示例

snippet.bash

```
#cfssl gencert -ca=ca.pem -ca-key=ca-key.pem -config=ca-config.json -profile=k8sdev etcd-csr.json |cfssljson -bare etcd
```

ca-config.json

snippet.json

```
{
   "signing": {
      "default": {
         "expiry": "87600h"
    },
      "profiles": {
         "k8sdev": {
            "expiry": "87600h",
            "usages": [
               "signing",
               "key encipherment",
               "server auth",
               "client auth"
         ]
      }
    }
  }
}
```

ca-csr.json

snippet.json

```
{
   "CN": "k8sdev",
   "key": {
      "algo": "rsa",
      "size": 2048
  },
   "names": [
    {
         "C": "CN",
         "ST": "Beijing",
         "L": "Beijing",
         "O": "k8s",
```

```
            "OU": "Scloud"
        }
    ],
    "ca": {
        "expiry": "87600h"
    }
}
```

etcd-csr.json

snippet.json

```
{
  "CN": "etcd",
  "hosts": [
    "127.0.0.1",
    "192.168.53.205",
    "192.168.52.63",
    "192.168.56.143"
  ],
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
    {
      "C": "CN",
      "ST": "BeiJing",
      "L": "BeiJing",
      "O": "k8s",
      "OU": "Scloud"
    }
  ]
}
```

kubernetes-csr.json

这时给 apiserver 用的证书，可以加上 apiserver 的 vip 和 域名。

snippet.json

```
{
  "CN": "k8sdev",
  "hosts": [
    "127.0.0.1",
    "192.168.53.205",
    "192.168.52.63",
    "192.168.56.143",
```

```
   "10.110.11.123",
   "169.169.0.1",
    "realdomain.xxx.com",
    "kubernetes",
    "kubernetes.default",
    "kubernetes.default.svc",
    "kubernetes.default.svc.cluster",
    "kubernetes.default.svc.cluster.local"
  ],
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
   {
      "C": "CN",
      "ST": "BeiJing",
      "L": "BeiJing",
      "O": "k8s",
      "OU": "Scloud"
   }
  ]
}
```

admin-csr.json

snippet.json

```
{
  "CN": "admin",
  "hosts": [],
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
   {
      "C": "CN",
      "ST": "BeiJing",
      "L": "BeiJing",
      "O": "system:masters",
      "OU": "Scloud"
   }
  ]
}
```

kube-controller-manager-csr.json

snippet.json

```
{
    "CN": "system:kube-controller-manager",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "hosts": [
    "127.0.0.1",
    "192.168.56.143",
    "192.168.52.63",
    "192.168.53.205"
    ],
    "names": [
    {
        "C": "CN",
        "ST": "BeiJing",
        "L": "BeiJing",
        "O": "system:kube-controller-manager",
        "OU": "Scloud"
    }
    ]
}
```

kube-proxy-csr.json

snippet.json

```
{
  "CN": "system:kube-proxy",
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
  {
      "C": "CN",
      "ST": "BeiJing",
      "L": "BeiJing",
      "O": "k8s",
      "OU": "Scloud"
  }
  ]
}
```

kube-scheduler-csr.json

snippet.json

```
{
    "CN": "system:kube-scheduler",
    "hosts": [
    "127.0.0.1",
    "192.168.52.63",
    "192.168.56.143",
    "192.168.53.205"
  ],
    "key": {
        "algo": "rsa",
        "size": 2048
  },
    "names": [
    {
        "C": "CN",
        "ST": "BeiJing",
        "L": "BeiJing",
        "O": "system:kube-scheduler",
        "OU": "Scloud"
    }
  ]
}
```

metrics-server-csr.json

snippet.json

```
{
  "CN": "aggregator",
  "hosts": [],
  "key": {
    "algo": "rsa",
    "size": 2048
 },
  "names": [
   {
     "C": "CN",
     "ST": "BeiJing",
     "L": "BeiJing",
     "O": "k8s",
     "OU": "Scloud"
   }
 ]
}
```

front-proxy-ca-csr.json

snippet.json

```
{
   "CN": "k8sdev",
   "key": {
      "algo": "rsa",
      "size": 2048
  },
   "names": [
     {
          "C": "CN",
          "ST": "Beijing",
          "L": "Beijing",
          "O": "k8s",
          "OU": "Scloud"
     }
  ],
   "ca": {
      "expiry": "87600h"
  }
}
```

浏览器访问 dashboard

基于 admin 证书 生成 .pfx 证书

snippet.bash

```
openssl pkcs12 -inkey admin-key.pem -in admin.pem -export -out k8sdev-admin.pfx
```

然后浏览器上执行以下步骤

1. 导入 ca.pem 到 受信任的根证书
2. 导入 .pfx 证书

Printed on: 2022/10/05 17:38

Convert to img Failed!