Kubernetes 인증서가 만료되면 클러스터 전반의 통신이 중지됩니다.

목차

소개

이 문서에서는 고객이 365일 이상 설치된 Kubernetes 기반 시스템을 사용할 때 발생할 수 있는 중단 문제에 대해 설명합니다.또한 상황을 해결하고 Kubernetes 기반 시스템을 다시 가동하고 실행하는 데 필요한 단계를 거칩니다.

문제

기본 설치된 Kubernetes 클러스터의 1년 후 클라이언트 인증서가 만료됩니다.Cisco CCS(CloudCenter Suite)에 액세스할 수 없습니다. 계속 나타나지만 로그인할 수 없습니다.kubectl CLI로 이동하면 "서버에 연결할 수 없습니다.x509:인증서가 만료되었거나 아직 유효하지 않습니다."

이 bash 스크립트를 실행하여 인증서의 만료 날짜를 확인할 수 있습니다.

```
for crt in /etc/kubernetes/pki/*.crt; do
     printf '%s: %s\n' \
     "$(date --date="$(openssl x509 -enddate -noout -in "$crt"|cut -d= -f 2)" --iso-8601)" \
     "$crt"
done | sort
```

또한 Action Orchestrator에 대한 opensource 워크플로를 찾아 이를 매일 모니터링하고 문제를 알립니다.

https://github.com/cisco-cx-workflows/cx-ao-shared-workflows/tree/master/CCSCheckKubernetesExpiration__definition_workflow_01E01VIRWZDE24mWlsHrqCGB9xUix0f9ZxG

솔루션

클러스터 전체에서 Kubeadm을 통해 새 인증서를 다시 발급해야 하며, 그런 다음 작업자 노드를 마스터에 다시 조인해야 합니다.

1. 마스터 노드에 로그인합니다.

2. IP 주소 show를 **통해** IP 주소**를 가져옵니다**.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3 kubernetes]# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8920 qdisc pfifo_fast state UP group default
qlen 1000
link/ether fa:16:3e:19:63:a2 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.20/24 brd 192.168.1.255 scope global dynamic eth0
valid_lft 37806sec preferred_lft 37806sec
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
default
link/ether 02:42:d0:29:ce:5e brd ff:ff:ff:ff:ff:ff
inet 172.17.0.1/16 scope global docker0
valid_lft forever preferred_lft forever
13: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1430 qdisc noqueue state UNKNOWN group default qlen
1000
link/ipip 0.0.0.0 brd 0.0.0.0
inet 172.16.176.128/32 brd 172.16.176.128 scope global tunl0
valid_lft forever preferred_lft forever
14: cali65453a0219d@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1430 qdisc noqueue state UP
group default
link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netnsid 4
```

3. cd /etc/kubernetes를 통해 Kubernetes 디렉토리로 이동합니다.

4. vi kubeadmCERT.yaml이라는 파일을 vi kubeadmCERT.yaml을 통해 생성합니다.

5. 파일은 다음과 같아야 합니다.

```
apiVersion: kubeadm.k8s.io/v1alpha1
kind: MasterConfiguration
api:
  advertiseAddress: <IP ADDRESS FROM STEP 2>
kubernetesVersion: v1.11.6
#NOTE: If the customer is running a load balancer VM then you must add these lines after...
#apiServerCertSANs:
#- <load balancer IP>
```

6. 이전 인증서 및 키를 백업합니다.이는 필수는 아니지만 권장됩니다.백업 디렉터리를 만들고 이 파일을 백업 디렉터리에 복사합니다.

```
#Files
#apiserver.crt
#apiserver.key
#apiserver-kubelet-client.crt
#apiserver-kubelet-client.key
#front-proxy-client.crt
#front-proxy-client.key

#ie
cd /etc/kubernetes/pki
mkdir backup
mv apiserver.key backup/apiserver.key.bak
```

7. 6단계를 건너뛴 경우 rm apiserver.crt와 같은 rm 명령을 통해 이전에 언급한 파일을 삭제할 수 있습니다.

8. kubeadmCERT.yaml 파일이 있는 위치로 다시 이동합니다.kubeadm —config

kubeadmCERT.yaml alpha phase certs apiserver를 통해 새 apiserver 인증서를 생성합니다.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3 kubernetes]# kubeadm --
config kubeadmCERT.yaml alpha phase certs apiserver
[certificates] Generated apiserver certificate and key.
[certificates] apiserver serving cert is signed for DNS names [cx-ccs-prod-master-d7f34f25-
f524-4f90-9037-7286202ed13a3 kubernetes kubernetes.default kubernetes.default.svc
kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 192.168.1.20]
```

9. kubeadm —config kubeadmCERT.yaml alpha phase certs apiserver-kubelet-client를 통해 새 apiserver kubelet 인증서를 생성합니다.

10. kubeadm —config kubeadmCERT.yaml alpha phase certs front-proxy-client를 통해 새 front-proxy-client 인증서를 생성합니다.

11. /etc/kubernetes 폴더에서 .conf 파일을 백업합니다.필수 사항은 아니지만 권장 사항입니다 .kubelet.conf, controller-manager.conf, scheduler.conf 및 가능한 admin.conf가 있어야 합니다.백업하지 않으려면 삭제할 수 있습니다.

12. kubeadm —config kubeadmCERT.yaml alpha phase kubeconfig all을 통해 새 컨피그레이션 파일을 생성합니다.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3 kubernetes]# kubeadm --
config kubeadmCERT.yaml alpha phase kubeconfig all
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/admin.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/controller-manager.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/scheduler.conf"
```

13. 새 admin.conf 파일을 호스트로 내보냅니다.

```
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
chown $(id -u):$(id -g) $HOME/.kube/config
chmod 777 $HOME/.kube/config
export KUBECONFIG=.kube/config
```

14. shutdown -r을 통해 마스터 노드를 지금 재부팅합니다.

15. 마스터가 백업되면 kubelet이 systemctl 상태 kubelet을 통해 실행되고 있는지 확인합니다.

16. kubectl get 노드를 통해 Kubernetes를 확인합니다.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3 ~]# kubectl get nodes
NAME                                                  STATUS     ROLES     AGE
VERSION
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a1   Ready      master    1y
v1.11.6
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a2   Ready      master    1y
v1.11.6
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3   Ready      master    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a1   NotReady   <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a2   NotReady   <none>    1y
```

```
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a3   NotReady   <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a4   NotReady   <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a5   NotReady   <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a6   NotReady   <none>    1y
v1.11.6
```

17. 각 마스터 노드에 대해 1~16단계를 반복합니다.

18. 하나의 마스터에서 **kubeadm 토큰 create —print-join-command를** 통해 새 조인 토큰**을 생성합니다**.나중에 사용할 수 있도록 해당 명령을 복사합니다.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a1 k8s-mgmt]# kubeadm token
create
--print-join-command kubeadm join 192.168.1.14:6443 --token m1ynvj.f4n3et3poki88ry4
--discovery-token-ca-cert-hash
sha256:4d0c569985c1d460ef74dc01c85740285e4af2c2369ff833eed1ba86e1167575
```

19. **노드**를 **전체**로 **구베틀**에서 **근로자**의 IP**를 가져옵니다**.

20. ssh **-i /home/cloud-user/keys/gen3-ao-prod.key cloud-user@192.168.1.17**과 같은 작업자에 로그인하고 루트 액세스로 이동합니다.

21. systemctl **stop kubelet**을 통해 **kubelet 서비스를 중지합니다**.

22. 이전 컨피그레이션 파일을 제거합니다. 여기에는 **ca.crt**, **kubelet.conf** 및 **bootstrap-kubelet.conf가 포함됩니다**.

```
rm /etc/kubernetes/pki/ca.crt
rm /etc/kubernetes/kubelet.conf
rm /etc/kubernetes/bootstrap-kubelet.conf
```

23. 19단계에서 노드의 이름을 가져옵니다.

24. 클러스터에 다시 참가하려면 작업자에 대한 명령을 실행합니다.18.에서 명령을 사용하되, **—node-name <name of node>**를 끝에 추가합니다.

```
[root@cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a1 kubernetes]# kubeadm join
192.168.1.14:6443 --token m1ynvj.f4n3et3poki88ry4 --discovery-token-ca-cert-hash
sha256:4d0c569985c1d460ef74dc01c85740285e4af2c2369ff833eed1ba86e1167575 --node-name cx-
ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a1
[preflight] running pre-flight checks
        [WARNING RequiredIPVSKernelModulesAvailable]: the IPVS proxier will not be used,
because the following required kernel modules are not loaded: [ip_vs_rr ip_vs_wrr
ip_vs_sh] or no builtin kernel ipvs support: map[ip_vs:{} ip_vs_rr:{} ip_vs_wrr:{}
ip_vs_sh:{} nf_conntrack_ipv4:{}]
you can solve this problem with following methods:
 1. Run 'modprobe -- ' to load missing kernel modules;
2. Provide the missing builtin kernel ipvs support
```

```
I0226 17:59:52.644282   19170 kernel_validator.go:81] Validating kernel version
I0226 17:59:52.644421   19170 kernel_validator.go:96] Validating kernel config
[discovery] Trying to connect to API Server "192.168.1.14:6443"
[discovery] Created cluster-info discovery client, requesting info from
"https://192.168.1.14:6443"
[discovery] Requesting info from "https://192.168.1.14:6443" again to validate TLS against
the pinned public key
[discovery] Cluster info signature and contents are valid and TLS certificate validates
against pinned roots, will use API Server "192.168.1.14:6443"
[discovery] Successfully established connection with API Server "192.168.1.14:6443"
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.11"
ConfigMap in the kube-system namespace
[kubelet] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-
flags.env"
[preflight] Activating the kubelet service
[tlsbootstrap] Waiting for the kubelet to perform the TLS Bootstrap...
[patchnode] Uploading the CRI Socket information "/var/run/dockershim.sock" to the Node
API object "cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a1" as an annotation

This node has joined the cluster:
* Certificate signing request was sent to master and a response
  was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the master to see this node join the cluster.
```

25. 작업자를 종료하고 kubectl get **노드**를 통해 마스터의 상태를 **확인합니다**.준비 상태여야 합니다.

26. 각 근로자에 대해 20~25단계를 반복합니다.

27. 마지막 **kubectl get 노드는** 모든 노드가 "준비" 상태이고, 다시 온라인 상태이며, 클러스터에 연결되었음을 표시해야 합니다.

```
[root@cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a1 ~]# kubectl get nodes
NAME                                                  STATUS    ROLES     AGE
VERSION
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a1   Ready     master    1y
v1.11.6
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a2   Ready     master    1y
v1.11.6
cx-ccs-prod-master-d7f34f25-f524-4f90-9037-7286202ed13a3   Ready     master    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a1   Ready     <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a2   Ready     <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a3   Ready     <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a4   Ready     <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a5   Ready     <none>    1y
v1.11.6
cx-ccs-prod-worker-d7f34f25-f524-4f90-9037-7286202ed13a6   Ready     <none>    1y
v1.11.6
```