Lab 4.1 - Planning the Deployment

Overview

In this exercise, we will investigate common network plugins. Each *kubelet* agent uses one plugin at a time. Due to complexity, the entire cluster uses one plugin which is configured prior to application deployment. Some plugins don't honor security configurations like network policies. Should you design a deployment which uses a network policy, there wouldn't be an error, you just would not be protected.

While still new, the community is moving towards the Container Network Interface (CNI) specification (https://github.com/containernetworking/cni). This provides the most flexibility and features. A common alternative is *kubenet*, a basic plugin which relies on the cloud provider to handle routing and cross-node networking. In a previous lab exercise, we configured Project Calico. Classic and external modes are also possible.

Evaluate Network Plugins

Verify your nodes are using a CNI plugin. Look for options passed to the *kubelet*. You may see other lines, including the **grep** command itself and a shell script running in a container which configures Calico.

```
student@ckad-2-nzjr:~$ ps -ef | grep cni
student  13473 13442  0 22:55 pts/1    00:00:00 grep --color=auto cni
root     14118     1  2 Mar30 ?        02:57:27 /usr/bin/kubelet
--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf
--kubeconfig=/etc/kubernetes/kubelet.conf
--pod-manifest-path=/etc/kubernetes/manifests --allow-privileged=true
--network-plugin=cni --cni-conf-dir=/etc/cni/net.d
--cni-bin-dir=/opt/cni/bin --cluster-dns=10.96.0.10
```

```
--cluster-domain=cluster.local --authorization-mode=Webhook
--client-ca-file=/etc/kubernetes/pki/ca.crt --cadvisor-port=0
--rotate-certificates=true --cert-dir=/var/lib/kubelet/pki
root     30591 30570  0 Mar30 ?        00:00:10 /bin/sh /install-cni.sh
```

View the details of the **`install-cni.sh`** script. The script runs in a container, the path to which will be different than the example below. Read through the script to see what it does on our behalf:

```
student@ckad-2-nzjr:~$ sudo find / -name install-cni.sh
/var/lib/docker/aufs/mnt/e95a30499a76e79027502bbb8ee4eeb8464a657e276a493249
f573f5d86e19b3/install-cni.sh
/var/lib/docker/aufs/diff/a7cd14de39089493b793f135ec965f0f3f79eeec8f9d78e67
9d7be1a3bdf3345/install-cni.sh
```

```
student@ckad-2-nzjr:~$ sudo less
/var/lib/docker/aufs/diff/a7cd14de39089493b793f135ec965f0f3f79eeec8f9d78e67
9d7be1a3bdf3345/install-cni.sh
```

There are many CNI providers possible. The following list represents some of the more common choices, but it is not exhaustive. With many new plugins being developed, there may be another which better serves your needs. Use these websites to answer the questions which follow. While we strive to keep the answers accurate, please be aware that this area receives a lot of attention and development, and changes often.

- Project Calico https://docs.projectcalico.org/v3.0/introduction/
- Calico with Canal https://docs.projectcalico.org/v3.0/getting-started/kubernetes/installation/hosted/canal/
- Weave Works https://www.weave.works/docs/net/latest/kubernetes/kube-addon
- Flannel https://github.com/coreos/flannel
- Romana http://romana.io/how/romana_basics/
- Kube Router: https://www.kube-router.io
- Kopeio https://github.com/kopeio/networking

Which of the plugins allow vxlans?

Which are layer 2 plugins?

Which are layer 3?

Which allow network policies?

Which can encrypt all TCP and UDP traffic?

Answers:

Which of the plugins allow vxlans?
Canal, Flannel, Kopeio-networking and Weave Net.

Which are layer 2 plugins?
Canal, Flannel, Kopeio-networking and Weave Net.

Which are layer 3?
Project Calico, Romana, and Kube Router

Which allow network policies?
Project Calico, Canal, Kube Router, Romana and Weave Net

Which can encrypt all TCP and UDP traffic?
Project Calico, Kopeio, and weave Net.

Multi-Container Pod Considerations

Using the information learned from this chapter, consider the following questions:

Which deployment method would allow the most flexibility, multiple applications per pod or one per Pod?

Which deployment method allows for the most granular scalability?

Which have the best performance?

How many IP addresses are assigned per pod?

What are some ways containers can communicate within the same pod?

What are some reasons you should have multiple containers per pod?

Answers:

Which deployment method would allow the most flexibility, multiple applications per pod or one per Pod?
One per pod.

Which deployment method allows for the most granular scalability?
One per pod

Which have the best performance?
Multiple per pod.

How many IP addresses are assigned per pod?
One

What are some ways containers can communicate within the same pod?
IPC, loopback or shared filesystem access.

What are some reasons you should have multiple containers per pod?
Lean containers may not have functionality like logging. Able to maintain lean execution, but add functionality as necessary.