Exercise 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 such as network policies. Should you design a deployment which and use a network policy there wouldn't be an error; the policy would have no effect.
>
> 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 in the fast changing space of container networking.
>
> 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. Several software defined network projects intended for Kubernetes have been created recently, with new features added regularly.

Evaluate Network Plugins

1. Verify your nodes are using a CNI plugin. Look for options passed to kubelet. You may see other lines including the grep command itself.

    ```
    student@ckad-1:~$ ps -ef | grep cni

    student   2518 30263  0 15:48 pts/0    00:00:00 grep --color=auto cni
    root     13578     1  3 Nov01 ?        01:28:45 /usr/bin/kubelet
    --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf
    --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml
    --cgroup-driver=cgroupfs --network-plugin=cni
    --pod-infra-container-image=k8s.gcr.io/pause:3.1
    --resolv-conf=/run/systemd/resolve/resolv.conf
    ```

2. 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-1:~$ sudo find / -name install-cni.sh

    /var/lib/docker/overlay2/e95a30499a76e79027502bbb8ee4eeb8464a657e276a493249f573f5d86e19b3/diff/install-cni.sh


    student@ckad-2-nzjr:~$ sudo less \
    /var/lib/docker/overlay2/e95a30499a76e79027502bbb8ee4eeb8464a657e276a493249f573f5d86e19b3/diff/install-cni.sh
    ```

3. 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 questions which follow. While we strive to keep the answers accurate, please be aware that this area has 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

4. Which of the plugins allow vxlans?

5. Which are layer 2 plugins?

6. Which are layer 3?

7. Which allow network policies?

8. Which can encrypt all TCP and UDP traffic?

Multi-container Pod Considerations

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

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

2. Which deployment method allows for the most granular scalability?

3. Which have the best performance?

4. How many IP addresses are assigned per pod?

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

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

Do you really know?

When and why would you use a multi-container pod?

Have you found a YAML example online?

Go back and review multi-container pod types and content on decoupling if you can't easily answer these questions. We touched on adding a second logging and a readiness container in a previous chapter and will work more with logging a future exercise.

✅ Solution 4.1

Plugin Answers

1. Which of the plugins allow vxlans?

   Canal, Flannel, Kopeio-networking, Weave Net

                                 THE LINUX FOUNDATION

2. Which are layer 2 plugins?

   Canal, Flannel, Kopeio-networking, Weave Net

3. Which are layer 3?

   Project Calico, Romana, Kube Router

4. Which allow network policies?

   Project Calico, Canal, Kube Router, Romana Weave Net

5. Which can encrypt all TCP and UDP traffic?

   Project Calico, Kopeio, Weave Net

Multi Pod Answers

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

   One per pod

2. Which deployment method allows for the most granular scalability?

   One per pod

3. Which have the best inter-container performance?

   Multiple per pod.

4. How many IP addresses are assigned per pod?

   One

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

   IPC, loopback or shared filesystem access.

6. 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, like Ambassadors and Sidecar containers.