Services and Networking

This chapter contains the following topics:

Load Balancing Kubernetes Services using NGINX

Cisco Container Platform uses NGINX to offer advanced layer 7 load balancing solutions. NGINX can handle a large number of requests and at the same time, it can be run on Kubernetes containers.

The NGINX load balancer is automatically provisioned as part of Kubernetes cluster creation. Each Kubernetes cluster is provisioned with a single L7 NGINX load balancer. You can access the load balancer using its virtual IP address, which can be found by running the command `kubectl get svc`.

To use the NGINX load balancer, you must create an Ingress resource. Ingress is a Kubernetes object that allows you to define HTTP load balancing rules to allow inbound connections to reach the cluster services. You can configure Ingress to create external URLs for services, load balance traffic, terminate SSL, offer name-based virtual hosting, and so on.

Types of Ingress

Cisco Container Platform supports the following types of Ingresses:

- **Simple fanout**

  It enables you to access the website using http.

  For example:

```
cafe.test.com ->   10.1.1.1  ->  / tea      tea-svc:80
                                 / coffee   coffee-svc:80
```

  For this type of Ingress, you need to create a yaml file that defines the Ingress rules.

Figure 1: Sample yaml file

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: cafe.test.com
    http:
      paths:
      - path: /tea
        backend:
          serviceName: tea-svc
          servicePort: 80
      - path: /coffee
        backend:
          serviceName: coffee-svc
          servicePort: 80
```

- **Simple fanout with SSL termination**

  It enables you to access the website using https.

  For example:

```
https://cafe.test.com ->    10.1.1.1   -> / tea       tea-svc:80
                                          / coffee    coffee-svc:80
```

  For this type of Ingress, you need to create the following yaml files:

  - A yaml file that defines the Secret

    Figure 2: Sample yaml file

```
apiVersion: v1
kind: Secret
metadata:
  name: cafe-secret
type: Opaque
data:
  tls.crt:  base64 encoded cert
  tls.key: base64 encoded key
```

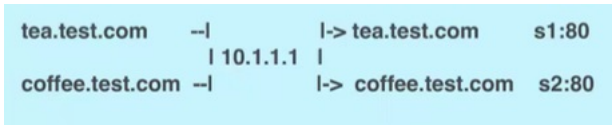
  - A yaml file that defines the Ingress rules

Figure 3: Sample yaml file

- **Name based virtual hosting**

  It enables you to access the website using multiple host names.

  For example:



  For this type of Ingress, you need to create a yaml file that defines the Ingress rules.

Figure 4: Sample yaml file

Note You can download the yaml files that are shown in this topic from the following link:

https://github.com/nginxinc/kubernetes-ingress/tree/master/examples/complete-example

For more information on a sample scenario of implementing Ingress, see Deploying Cafe Application with Ingress.

Network Policies

Cisco Container Platform supports Kubernetes NetworkPolicies. The NetworkPolicies are independent of the underlying container network plugin–ACI CNI, Contiv VPP, or Calico.