

# **Services and Networking**

This chapter contains the following topics:

- Load Balancing Kubernetes Services using NGINX, on page 1
- L7 Ingress, on page 1
- L4 Ingress, on page 3
- Ingress CA, on page 4
- Network Policies, on page 7
- Load Balancer Services, on page 7

# 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 -n ccp.

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.

## **L7 Ingress**

Cisco Container Platform supports the following types of L7 Ingresses:

Simple fanout

It enables you to access the website using http.

### Example

cafe.test.com -> 10.1.1.1

tea-svc:80

/coffee coffee-svc:80

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

/t.ea

Sample yaml file

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

Simple fanout with SSL termination

It enables you to access the website using https.

#### 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

#### 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

### Sample yaml file

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: cafe-ingress
spec:
 tls:
 -hosts:
  -cafe.test.com
  secretName: cafe-secret
 rules:
 -host: cafe.example.com
 http:
  paths:
   -path:/
  backend:
    serviceName: tea-svc
    sevicePort: 80
   -path:/
   backend:
```

```
serviceName: coffee-svc
servicePort: 80
```

#### Name based virtual hosting

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

#### Example

tea.test.com --| |-> tea.test.com s1:80 |10.1.1.1 | coffee.test.com --| |-> coffee.test.com s2:80

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

#### Sample yaml file

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



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.

### L4 Ingress

NGINX supports L4 TCP and UDP Ingress load balancing. It uses the NGINX helm chart that contains the TCP or UDP service mappings, instead of the Ingress resources as in the case of L7 support.

### **Configuring L4 Load Balancing**

	Note	NGINX supports either TCP or UDP L4 load balancing, but not both simultaneously.				
Step 1	A coord th	a Kuhamatan aluatar mada using sah				
Step I	Access the Kubernetes cluster master node using ssh.					
	Note	username> <ip address="" master="" node="" of=""> Once you create a Kubernetes cluster, it may take a few minutes for the necessary services to start. If ssh to a cluster fails, we recommend that you try again after a few minutes.</ip>				
Step 2	Get the current helm configuration values.					
	helm get	valuesall nginx-ingress > 14.yaml				
Step 3	You can s	4.yaml file. earch for <i>tcp</i> or <i>udp</i> in the 14.yaml file, and then add your L4 services. wing example shows adding the tcp-test-svc TCP service that uses port 3333.				
	tcp: "900	0": default/tcp-test-svc:3333				
	The follow	wing example shows adding the udp-test-svc UDP service that uses port 5005.				
	udp: "900	1": default/udp-test-svc:5005				
Step 4	Update th	Update the NGINX helm chart with the L4 service mappings.				
	helm upg	radeinstall nginx-ingress /opt/ccp/charts/nginx-ingress.tgz -f 14.yaml				
	Note	You need to restart the NGINX Ingress controller pods for the new configuration to take effect.				
Step 5		t ingress has successfully mapped the port. get services -o wide -w nginx-ingress-controller				

# **Ingress CA**

Cisco Container Platform by default creates an L7 Ingress service in order to support Monitoring Health of Cluster Deployments, Monitoring Logs from Cluster Deployments, and Setting up Kubernetes Dashboard. All of these services are exposed with TLS enabled, and the certificate authority (CA) that is used to sign the Ingress controller server certificate is self-signed and per cluster based.

In order to reach the services without triggering SSL warning, you can either add the CA as part of your application that needs to interact with services behind Cisco Container Platform ingress (preferred), or add the CA to your system trusted CA list. The following section describes how to obtain the CA certificate.

**Step 1** Log in to the Kubernetes dashboard from browser as described in Setting up Kubernetes Dashboard section, download the kubeconfig file, and then use it to login to the Kubernetes dashboard.

Step 2From the right pane, click the dropdown box under Namespace, click the ccp namespace.Figure 1: Kubernetes Dashboard

🛞 kubernetes
≡ Overview
Cluster
Namespaces
Nodes
Persistent Volumes
Select namespace
All namespaces
NAMESPACES
сср
default
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets

Step 3Click the Secrets tab.The Secrets pane appears.

#### Figure 2: Secrets Pane

Cluster	Secrets				-
Namespaces					
Nodes	Name ‡	Туре	Age ≑		
Persistent Volumes	ccp-ingress-tis	kubernetes.io/tls	10 hours		
Roles Storage Classes	ccp-ingress-tls-ca	kubernetes.io/tls	10 hours		:
	ccp-addons-grafana	Opaque	10 hours		
Namespace	ccp-addons-kubernetes-dashboard-token-wvzcj	kubernetes.io/service-account-token	10 hours		:
	ccp-addons-metallb-controller-token-7qzft	kubernetes.io/service-account-token	10 hours		
Overview	ccp-addons-metallb-speaker-token-vvhmt	kubernetes.io/service-account-token	10 hours		
Workloads	ccp-addons-nginx-ingress-token-8jsj8	kubernetes.io/service-account-token	10 hours		
Cron Jobs Daemon Sets	ccp-addons-prometheus-alertmanager-token-7sg77	kubernetes.lo/service-account-token	10 hours		:
Deployments	ccp-addons-prometheus-kube-state-metrics-token-7lpbw	kubernetes.io/service-account-token	10 hours		:
Jobs	ccp-addons-prometheus-node-exporter-token-5kxvj	kubernetes.lo/service-account-token	10 hours		:
Pods				1 - 10 of 17  < < >	×
Replica Sets					
Replication Controllers					
Stateful Sets					
Discovery and Load Balancing					
Ingresses					
Services					
Config and Storage					
Config Maps					
Persistent Volume Claims					
Secrets					

- **Step 4** Open the ccp-ingress-tls-ca secret and find the data for tls.crt.
- **Step 5** Click the Eye icon to view the details of a tls.crt.

### Figure 3: Secrets Pane Showing Details of tls.crt

🛞 kubernetes	Q Search	+ CREATE   😝
	iecrets > ccp-ingress-tls-ca	🖍 EDIT 🧻 DELETE
Cluster Namespaces Nodes Persistent Volumes Roles Storage Classes	Details Name: ccp-ingress-tls-ca Namespace: ccp Creation Time: 2018-07-30T08-47 UTC Type: kubernetes: io/tls	
Namespace	Data	
Config Maps Persistent Volume Claims Secrets	pcotvt81UfLgvvCpXED5hSqxdr72a0HjxBmLagg6fYuSDUg4XHYV110vOd5a472e 3p3zBUg5VTVx105ft+5m4715VZ8viDb#fFNsJyNDhYTXFVTq5CR51X1cKv+4x mCFXN66HhyTPc4/taez2k/cklBnTtv82bG6UX5jsfxNtChuB4423FFTRa90 pOG29hp08D03g3T1+4k3LYXDNMX9CX5655fPcRko/NDK71a/TgLC95+VvfsFmA wtWm2trvy16jKWy4pGLTVu	

You can save the CA data into a file, and use it when a client is trying to connect to the Ingress service.

The following example uses curl to get to the dashboard using the saved CA certificate.

```
curl --cacert ./ca.crt -I https://10.10.99.185/dashboard
HTTP/1.1 200 OK
Server: nginx/1.13.12
Date: Mon, 30 Jul 2018 19:08:11 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Vary: Accept-Encoding
Accept-Ranges: bytes
Cache-Control: no-store
Strict-Transport-Security: max-age=15724800; includeSubDomains
```

## **Network Policies**

Cisco Container Platform supports Kubernetes NetworkPolicies. The NetworkPolicies are independent of the underlying container network plugin.

## Load Balancer Services

Cisco Container Platform supports load balancer services on tenant clusters.

While creating a tenant cluster, you need to choose the number of load balancer IP addresses that you want to allocate for a tenant cluster from a VIP pool that you want to use.



**Note** The cluster creation operation fails if the number of requested load balancer IP addresses is more than the available IP addresses in the pool.

For more information, see Creating Kubernetes Clusters on vSphere On-prem Clusters.

Once load balancer IP addresses are allocated for a tenant cluster, externally reachable load balancer IP addresses are automatically provisioned for the load balancer services.

The following code provides an example of creating a service of type **LoadBalancer**.

```
apiVersion: v1
kind: Service
metadata:
name: frontend
labels:
app: guestbook
tier: frontend
type: LoadBalancer
```

You can update the number of available load balancer IP addresses from the **Edit Cluster** screen. You need to be aware of the number of used addresses in order to update the number of allocated load balancer IP addresses.

For example:

Suppose the current tenant is allocated with five load balancer IP addresses. If there are three load balanced services running, you cannot reduce the number of load balancer IP addresses to three or less as there are services using those IP addresses already.



Note

When you delete a tenant cluster, the allocated load balancer IP addresses are recycled to the VIP pool.