

FAUCET SDN on Kubernetes in the cloud

(aka "Toward peak buzzword in our talk titles")

Simeon Miteff - 23 October 2019

Objective

Run FAUCET controller(s) as containers on top of Kubernetes in Google Cloud. Load-balance OpenFlow TCP control connection from data planes running "on prem".



Steven R. Baker
@srbaker

Kubernetes is an Ancient Greek word meaning "More containers than customers."

1:32 PM · May 31, 2018

Deployment resource YAML

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: faucet-controllers
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: faucet
    spec:
      containers:
        - name: faucet-controller
          image: faucet/faucet:latest
          imagePullPolicy: Always
          env:
            - name: FAUCET_LOG
              value: STDOUT
            - name: FAUCET_EXCEPTION_LOG
              value: STDOUT
          ports:
            - name: openflow
              containerPort: 6653
          volumeMounts:
            - name: faucet-controller-config
              mountPath: /etc/faucet/faucet.yaml
              subPath: faucet.yaml
      volumes:
        - name: faucet-controller-config
          configMap:
            name: faucet-config
```

ConfigMap resource YAML

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: faucet-config
data:
  faucet.yaml: |
    vlans:
      office:
        vid: 100
        description: "office network"

    dps:
      sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
          1:
            name: "host1"
            description: "host1 network namespace"
            native_vlan: office
          2:
            name: "host2"
            description: "host2 network namespace"
            native_vlan: office
```

Service (LoadBalancer) resource YAML

```
apiVersion: v1
kind: Service
metadata:
  name: faucet-lb-service
spec:
  ports:
    - port: 6653
      protocol: TCP
      targetPort: openflow
  selector:
    app: faucet
  type: LoadBalancer
```

Deal with Google Cloud bureaucracy

```
$ gcloud config configurations create personal
Created [personal].
Activated [personal].

$ gcloud auth login
Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/...

$ gcloud projects create faucet-cloud0 --name=faucet
Create in progress for \
    [https://cloudresourcemanager.googleapis.com/v1/projects/faucet-cloud0].
Waiting for [operations/cp.5742752346006269608] to finish...done.
Enabling service [cloudapis.googleapis.com] on project [faucet-cloud0]...
Operation "operations/acf.aeeb9d7d-d2e1-4a26-8e5a-d67876683753" \
    finished successfully.

simeon@feijoa:~$ gcloud config set project faucet
Updated property [core/project].
```

- Enable billing account
- Enable GKE API

Setting up GKE (Google cloud managed Kubernetes)

```
$ gcloud container clusters create faucet-test --num-nodes=1 --zone us-east1-b
Creating cluster faucet-test in us-east1-b... Cluster is being health-checked \
(master is healthy).
..done.
Created [https://container.googleapis.com/v1/projects/faucet-cloud0/zones/
us-east1-b/clusters/faucet-test].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/
kubernetes/workload_/gcloud/us-east1-b/faucet-test?project=faucet-cloud0
kubeconfig entry generated for faucet-test.
NAME          LOCATION    MASTER_VERSION  MASTER_IP      MACHINE_TYPE
faucet-test   us-east1-b  1.13.10-gke.0   104.196.125.8  n1-standard-1
NODE_VERSION  NUM_NODES   STATUS
1.13.10-gke.0 1           RUNNING
```

```
$ kubectl get nodes
NAME                                     STATUS ROLES  AGE   VERSION
gke-faucet-test-default-pool-d0d2ce3d-0gcm Ready  <none> 2m56s v1.13.10-gke.0
```

Create the resources to run FAUCET

```
$ kubectl create -f config.yaml  
configmap/faucet-config created  
  
$ kubectl create -f deployment.yaml  
deployment.extensions/faucet-controllers created  
  
$ kubectl create -f loadbalancer.yaml  
service/faucet-lb-service created
```


Did it work? 1/2

Check the container:

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
faucet-controllers-5b86f6fd78-klwf6	0/1	ContainerCreating	0	11s

Did it work? 2/2

Now check the logs:

```
$ kubectl logs -f faucet-controllers-5b86f6fd78-klwf6
Starting with UID=0 GID=0
loading app faucet.faucet
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context faucet_experimental_api
instantiating app faucet.faucet of Faucet
instantiating app ryu.controller.ofp_handler of OFPHandler
Oct 22 18:57:23 faucet INFO      Reloading configuration
Oct 22 18:57:23 faucet INFO      configuration /etc/faucet/faucet.yaml changed, ana
Oct 22 18:57:23 faucet INFO      Add new datapath DPID 1 (0x1)
Oct 22 18:57:23 faucet.valve INFO      DPID 1 (0x1) sw1 table ID 0 table config mat
Oct 22 18:57:23 faucet.valve INFO      DPID 1 (0x1) sw1 table ID 1 table config mat
Oct 22 18:57:23 faucet.valve INFO      DPID 1 (0x1) sw1 table ID 2 table config exa
Oct 22 18:57:23 faucet.valve INFO      DPID 1 (0x1) sw1 table ID 3 table config mat
(1) wsgi starting up on http://0.0.0.0:9302
```

How do I point my DP to the cloud?

```
$ kubectl describe service/faucet-lb-service
Name:          faucet-lb-service
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=faucet
Type:          LoadBalancer
IP:           10.11.250.40
Port:         <unset> 6653/TCP
TargetPort:    openflow/TCP
NodePort:      <unset> 32102/TCP
Endpoints:     10.8.0.11:6653
Session Affinity: None
External Traffic Policy: Cluster
Events:
  Type    Reason              Age   From                    Message
  ----    -
  Normal  EnsuringLoadBalancer 46s   service-controller     Ensuring load balancer
```

Ah, there is our endpoint....

```
$ gcloud compute forwarding-rules list
NAME                                REGION    IP_ADDRESS    IP_PROTOCOL
a27ea9a46f4fe11e9a6bf42010a8e01c  us-east1  34.74.88.153  TCP
TARGET
us-east1/targetPools/a27ea9a46f4fe11e9a6bf42010a8e01c
```

```
$ gcloud compute forwarding-rules describe a27ea9a46f4fe11e9a6bf42010a8e01c \
  --region us-east1
IPAddress: 34.74.88.153
IPProtocol: TCP
creationTimestamp: '2019-10-22T12:00:49.610-07:00'
description: '{"kubernetes.io/service-name":"default/faucet-lb-service"}'
id: '729954497589341966'
kind: compute#forwardingRule
loadBalancingScheme: EXTERNAL
name: a27ea9a46f4fe11e9a6bf42010a8e01c
networkTier: PREMIUM
portRange: 6653-6653
region: https://www.googleapis.com/compute/v1/projects/faucet-cloud0/
  regions/us-east1
selfLink: https://www.googleapis.com/compute/v1/projects/faucet-cloud0/
  regions/us-east1/forwardingRules/a27ea9a46f4fe11e9a6bf42010a8e01c
target: https://www.googleapis.com/compute/v1/projects/faucet-cloud0/
  regions/us-east1/targetPools/a27ea9a46f4fe11e9a6bf42010a8e01c
```

Set up test OVS DP (Hi Ben and Brad!)

```
$ sudo ovs-vsctl add-br br0 \  
-- set bridge br0 other-config:datapath-id=000000000000000001 \  
-- set bridge br0 other-config:disable-in-band=true \  
-- set bridge br0 fail_mode=secure \  
-- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \  
-- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \  
-- set-controller br0 tcp:34.74.88.153:6653
```

See the DP connecting in FAUCET pod logs

```
Oct 22 20:28:50 faucet.valve INFO
    DPID 1 (0x1) sw1 Cold start configuring DP
Oct 22 20:28:50 faucet.valve INFO
    DPID 1 (0x1) sw1 Port 1 (host1 network namespace) configured
Oct 22 20:28:50 faucet.valve INFO
    DPID 1 (0x1) sw1 Port 2 (host2 network namespace) configured
Oct 22 20:28:50 faucet.valve INFO
    DPID 1 (0x1) sw1 Configuring VLAN office vid:100 untagged: Port 1,Port 2
```

Ping something (Hi Brad!)

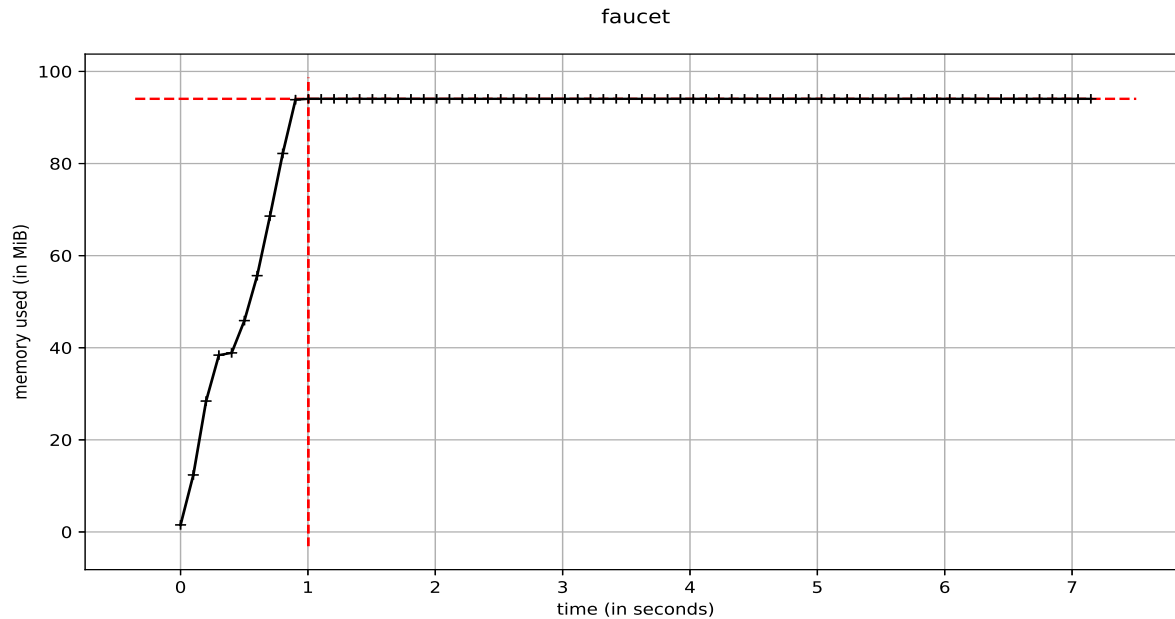
```
$ as_ns host1 ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.659 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.040 ms
```

Log:

```
Oct 22 20:30:58 faucet.valve INFO
DPID 1 (0x1) sw1 L2 learned on Port 1 4e:27:a0:18:02:de (L2 type 0x0800,
L2 dst 72:2f:5b:26:0a:4c, L3 src 192.168.0.1, L3 dst 192.168.0.2)
Port 1 VLAN 100 (1 hosts total)
Oct 22 20:30:58 faucet.valve INFO
DPID 1 (0x1) sw1 L2 learned on Port 2 72:2f:5b:26:0a:4c (L2 type 0x0800,
L2 dst 4e:27:a0:18:02:de, L3 src 192.168.0.2, L3 dst 192.168.0.1)
Port 2 VLAN 100 (2 hosts total)
```

Important missing things

- Resource request/limits (128MB? how does FAUCET memory scale?).



- Readiness and liveness tests (OpenFlow is not HTTP).
- What happens if we add more replicas to the Deployment?
- Restart vs. reload on config update (break-before-make?).

Concluding thoughts

- So, you *can* do this, but *should* you?
- The GCP load balancer seems to work fine for OpenFlow.

This presentation and the Kubernetes manifest YAML files are on github, here:

<https://github.com/simeonmteff/faucet-kubernetes>