# Kubernetes DIY Homework

Author: Bokotei Oleksandr

GitHub: https://github.com/nonamecoder2002/GLBaseCamp2021

## [Base]:

Let's create VPC network kubernetes-the-hard-way with subnet kubernetes:

| | | | | | | |
|---|---|---|---|---|---|---|
| kubernetes-the-hard-way | | 1 | 1460 | Custom | | |
| | us-west1 | kubernetes | | | 11.240.0.0/24 | 11.240.0.1 |

Then create firewall rules:

| Name | Type | Targets | Filters | Protocols / ports | Action |
|---|---|---|---|---|---|
| kubernetes-the-hard-way-allow-external | Ingress | Apply to all | IP ranges: 0.0.0.0/0 | tcp:22;tcp:6443<br>icmp | Allow |

| | | | | | |
|---|---|---|---|---|---|
| kubernetes-the-hard-way-allow-internal | Ingress | Apply to all | IP ranges: 11.240.0.0/24, 11.200.0.0/16 | tcp<br>udp<br>icmp | Allow |

Then reserve static external IP address:

| Name | External Address | Region | Type ↓ | Version | In use by |
|---|---|---|---|---|---|
| kubernetes-the-hard-way | 35.233.246.54 | us-west1 | Static | IPv4 | Forwarding rule kubernetes-forwarding-rule |

Then create Kubernetes controller instances:

| Status | Name ↑ | Zone | Recommendations | In use by | Internal IP | External IP |
|---|---|---|---|---|---|---|
| ✓ | controller-0 | us-west1-a | | kubernetes-target-pool | 11.240.0.10 (nic0) | 34.83.159.81 |
| ✓ | controller-1 | us-west1-a | | kubernetes-target-pool | 11.240.0.11 (nic0) | 34.82.16.41 |
| ✓ | controller-2 | us-west1-a | | kubernetes-target-pool | 11.240.0.12 (nic0) | 34.82.248.162 |

Then create Kubernetes worker instances:

| | | | | |
|---|---|---|---|---|
| ✓ | worker-0 | us-west1-a | 11.240.0.20 (nic0) | 34.127.110.9 |
| ✓ | worker-1 | us-west1-a | 11.240.0.21 (nic0) | 34.83.164.175 |
| ✓ | worker-2 | us-west1-a | 11.240.0.22 (nic0) | 34.82.199.247 |

List of all created instances:

```
NAME          ZONE        MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP   EXTERNAL_IP     STATUS
controller-0  us-west1-a  e2-small                   11.240.0.10   34.83.159.81    RUNNING
controller-1  us-west1-a  e2-small                   11.240.0.11   34.82.16.41     RUNNING
controller-2  us-west1-a  e2-small                   11.240.0.12   34.82.248.162   RUNNING
worker-0      us-west1-a  e2-small                   11.240.0.20   34.127.110.9    RUNNING
worker-1      us-west1-a  e2-small                   11.240.0.21   34.83.164.175   RUNNING
worker-2      us-west1-a  e2-small                   11.240.0.22   34.82.199.247   RUNNING
```

Generate all needed certificates:

```
-rw------- 1 alex alex 1679 Jun  2 16:00 admin-key.pem
-rw-r--r-- 1 alex alex 1428 Jun  2 16:00 admin.pem
-rw------- 1 alex alex 1675 Jun  2 16:00 ca-key.pem
-rw-r--r-- 1 alex alex 1318 Jun  2 16:00 ca.pem
-rw------- 1 alex alex 1679 Jun  2 16:02 kube-controller-manager-key.pem
-rw-r--r-- 1 alex alex 1484 Jun  2 16:02 kube-controller-manager.pem
-rw------- 1 alex alex 1675 Jun  2 16:02 kube-proxy-key.pem
-rw-r--r-- 1 alex alex 1452 Jun  2 16:02 kube-proxy.pem
-rw------- 1 alex alex 1675 Jun  2 16:02 kube-scheduler-key.pem
-rw-r--r-- 1 alex alex 1460 Jun  2 16:02 kube-scheduler.pem
-rw------- 1 alex alex 1679 Jun  2 16:05 kubernetes-key.pem
-rw-r--r-- 1 alex alex 1663 Jun  2 16:05 kubernetes.pem
-rw------- 1 alex alex 1679 Jun  2 16:05 service-account-key.pem
-rw-r--r-- 1 alex alex 1440 Jun  2 16:05 service-account.pem
```

```
-rw------- 1 alex alex 1679 Jun  2 16:01 worker-0-key.pem
-rw-r--r-- 1 alex alex 1493 Jun  2 16:01 worker-0.pem
-rw------- 1 alex alex 1679 Jun  2 16:01 worker-1-key.pem
-rw-r--r-- 1 alex alex 1493 Jun  2 16:01 worker-1.pem
-rw------- 1 alex alex 1679 Jun  2 16:01 worker-2-key.pem
-rw-r--r-- 1 alex alex 1493 Jun  2 16:01 worker-2.pem
```

**And copy them to controller & worker instances**

**Then using this certificates create Kubernetes config files for authentication:**

```
-rw------- 1 alex alex 6265 Jun  2 16:13 admin.kubeconfig
-rw------- 1 alex alex 6391 Jun  2 16:13 kube-controller-manager.kubeconfig
-rw------- 1 alex alex 6321 Jun  2 16:12 kube-proxy.kubeconfig
-rw------- 1 alex alex 6337 Jun  2 16:13 kube-scheduler.kubeconfig
-rw------- 1 alex alex 6387 Jun  2 16:11 worker-0.kubeconfig
-rw------- 1 alex alex 6387 Jun  2 16:11 worker-1.kubeconfig
-rw------- 1 alex alex 6387 Jun  2 16:11 worker-2.kubeconfig
```

**And distribute them to controller & worker instances**

**Create data encryption config and send it to all controller instances:**

```
-rw-r--r-- 1 alex alex  240 Jun  2 16:16 encryption-config.yaml
```

**Then install & start etcd Server on each controller instance:**

```
5b2ead4ad6a47c97, started, controller-2, https://11.240.0.12:2380, https://11.240.0.12:2379, false
c2b91050f00f9313, started, controller-1, https://11.240.0.11:2380, https://11.240.0.11:2379, false
f9e0a1e379a1d360, started, controller-0, https://11.240.0.10:2380, https://11.240.0.10:2379, false
```

Start kube-apiserver, kube-controller-manager, kube-scheduler on the controller instances

```
alex@controller-0:~$ kubectl cluster-info --kubeconfig admin.kubeconfig
Kubernetes control plane is running at https://127.0.0.1:6443
```

Create external Load Balancer:

```
alex@DESKTOP-LBU2UOH:~/kuber_hardway/sertificates$ curl --cacert ca.pem https://${KUBERNETES_PUBLIC_ADDRESS}:6443/version
{
  "major": "1",
  "minor": "21",
  "gitVersion": "v1.21.0",
  "gitCommit": "cb303e613a121a29364f75cc67d3d580833a7479",
  "gitTreeState": "clean",
  "buildDate": "2021-04-08T16:25:06Z",
  "goVersion": "go1.16.1",
  "compiler": "gc",
  "platform": "linux/amd64"
```

Then configure CNI Networking, containerd, the Kubelet, the Kubernetes Proxy on worker instances:

```
NAME       STATUS   ROLES    AGE     VERSION
worker-0   Ready    <none>   4h22m   v1.21.0
worker-1   Ready    <none>   4h22m   v1.21.0
worker-2   Ready    <none>   4h22m   v1.21.0
alex@DESKTOP-LBU2UOH:~$
```

Then configure the remote admin access to the cluster:

```
alex@DESKTOP-LBU2UOH:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.1", GitCommit:"5e58841cce77d4bc13713ad2b91fa0d961e69192", G
itTreeState:"clean", BuildDate:"2021-05-12T14:18:45Z", GoVersion:"go1.16.4", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.0", GitCommit:"cb303e613a121a29364f75cc67d3d580833a7479", G
itTreeState:"clean", BuildDate:"2021-04-08T16:25:06Z", GoVersion:"go1.16.1", Compiler:"gc", Platform:"linux/amd64"}
```

```
alex@DESKTOP-LBU2UOH:~$ kubectl get nodes
NAME       STATUS   ROLES    AGE     VERSION
worker-0   Ready    <none>   4h26m   v1.21.0
worker-1   Ready    <none>   4h26m   v1.21.0
worker-2   Ready    <none>   4h26m   v1.21.0
```

Let's create network routes for each worker instance:

```
NAME                            NETWORK                  DEST_RANGE      NEXT_HOP
default-route-518dfd7c09563df0  kubernetes-the-hard-way  0.0.0.0/0       default-internet-gateway
default-route-d69b192c3d337d03  kubernetes-the-hard-way  11.240.0.0/24   kubernetes-the-hard-way
kube-route-11-200-0-11-240      kubernetes-the-hard-way  11.200.0.0/24   11.240.0.20
kube-route-11-200-1-11-240      kubernetes-the-hard-way  11.200.1.0/24   11.240.0.21
kube-route-11-200-2-11-240      kubernetes-the-hard-way  11.200.2.0/24   11.240.0.22
```

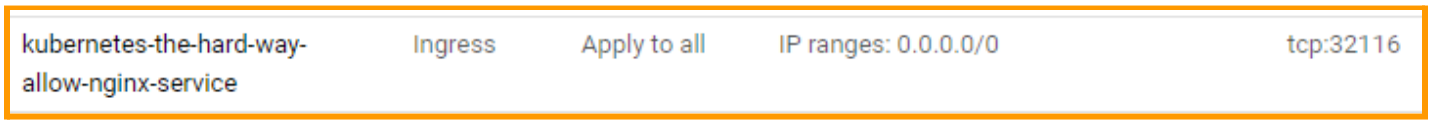
Then add cluster DNS add-on:

```
NAME                       READY   STATUS    RESTARTS   AGE
coredns-8494f9c688-295pc   1/1     Running   0          17h
coredns-8494f9c688-djxrs   1/1     Running   0          17h
```

To verify that all works OK, let`s deploy nginx:

```
alex@DESKTOP-LBU2UOH:~$ kubectl get po
NAME                     READY   STATUS    RESTARTS   AGE
nginx-6799fc88d8-q6tlz   1/1     Running   0          18h
```

**Expose its port 80 via NodePort & create firewall rule:**




**Make http request to worker external IP on the NodePort:**

```
alex@DESKTOP-LBU2UOH:~$ curl --head 34.83.164.175:32116
HTTP/1.1 200 OK
Server: nginx/1.21.0
Date: Thu, 03 Jun 2021 11:54:07 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 25 May 2021 12:28:56 GMT
Connection: keep-alive
ETag: "60aced88-264"
Accept-Ranges: bytes
```

# [EXT]:

**Let's create the virtual service with Round Robin scheduler:**

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.10.10.10:http rr
```

**Next create 2 docker containers with nginx running in both:**

```
IMAGE               COMMAND                  CREATED             STATUS
nginx               "/docker-entrypoint.…"   20 minutes ago      Up 20 minutes
nginx               "/docker-entrypoint.…"   21 minutes ago      Up 21 minutes
```

**Then add them to the virtual service ipvsadm:**

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.10.10.10:http rr
  -> 172.17.0.2:http              Masq    1      0          0
  -> 172.17.0.3:http              Masq    1      0          0
```

**To verify if everything is OK let`s "curl" the service IP:**

```
root@remote1noodle:/# curl --head 10.10.10.10
HTTP/1.1 200 OK
Server: nginx/1.21.0
Date: Thu, 03 Jun 2021 13:10:22 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 25 May 2021 12:28:56 GMT
Connection: keep-alive
ETag: "60aced88-264"
Accept-Ranges: bytes
```