CS571
MANSI SHAH
(19526)

Creating services

```
cs571@ubuntu:~/app_js$ vim kubia-svc.yaml
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc.yaml
service/kubia created
cs571@ubuntu:~/app_js$ cat kubia-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: kubia
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ vim kubia-svc.yaml
cs571@ubuntu:~/app_js$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   23d
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc.yaml
service/kubia created
cs571@ubuntu:~/app_js$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   23d
kubia        ClusterIP   10.99.241.94   <none>        80/TCP    4s
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-42qpz   1/1     Running   0          59m
kubia-8frcm   1/1     Running   0          59m
kubia-s8mc4   1/1     Running   0          69m
cs571@ubuntu:~/app_js$ kubectl exec kubia-s8mc4 -- curl -s http://10.99.241.94
You've hit kubia-42qpz
cs571@ubuntu:~/app_js$
```

Session Affinity:

```
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc.yaml
service/kubia created
cs571@ubuntu:~/app_js$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP   23d
kubia        ClusterIP   10.103.60.117   <none>        80/TCP    5s
cs571@ubuntu:~/app_js$ kubectl exec kubia-s8mc4 -- curl -s http://10.103.60.117
You've hit kubia-8frcm
cs571@ubuntu:~/app_js$ kubectl exec kubia-s8mc4 -- curl -s http://10.103.60.117
You've hit kubia-8frcm
cs571@ubuntu:~/app_js$ kubectl exec kubia-s8mc4 -- curl -s http://10.103.60.117
You've hit kubia-8frcm
cs571@ubuntu:~/app_js$ cat kubia-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  sessionAffinity: ClientIP
  ports:
  - port: 80
    targetPort: 8080
  selector:
    run: kubia
cs571@ubuntu:~/app_js$
```

Exposing multiple ports to service

```
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc.yaml
service/kubia created
cs571@ubuntu:~/app_js$ cat kubia-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  sessionAffinity: ClientIP
  ports:
  - name: http
    port: 80
    targetPort: 8080
  - name: https
    port: 443
    targetPort: 8443
  selector:
    run: kubia
cs571@ubuntu:~/app_js$
```

Specifying port names in a pod definition

```
cs571@ubuntu:~/app_js$ cat kubia-manual.yaml
apiVersion: v1
kind: Pod
metadata:
  name: kubia
  labels:
    app: kubia
spec:
  containers:
  - image: mansi2210/shahm888:kubia
    name: kubia
    ports:
    - name: http
      containerPort: 8080
    - name: https
      containerPort: 8443
cs571@ubuntu:~/app_js$ kubectl create -f kubia-manual.yaml
pod/kubia created
cs571@ubuntu:~/app_js$
```

Referring to named ports in a service

```
cs571@ubuntu:~/app_js$ cat kubia-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  sessionAffinity: ClientIP
  ports:
  - name: http
    port: 80
    targetPort: http
  - name: https
    port: 443
    targetPort: https
  selector:
    run: kubia
cs571@ubuntu:~/app_js$
```

Discovering services through environment variables

```
cs571@ubuntu:~/app_js$ kubectl delete pods --all
pod "kubia" deleted
pod "kubia-42qpz" deleted
pod "kubia-8frcm" deleted
pod "kubia-s8mc4" deleted
```

Service-related environment variables in a container

```
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-72d2n   1/1     Running   0          68s
kubia-fbt2q   1/1     Running   0          68s
kubia-t5lvg   1/1     Running   0          68s
cs571@ubuntu:~/app_js$ kubectl exec kubia-t5lvg env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=kubia-t5lvg
KUBIA_SERVICE_PORT_HTTPS=443
KUBIA_PORT_443_TCP_PROTO=tcp
KUBIA_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBIA_SERVICE_PORT_HTTP=80
KUBIA_PORT_80_TCP_PROTO=tcp
KUBIA_PORT_80_TCP_PORT=80
KUBIA_PORT_80_TCP_ADDR=10.97.167.40
KUBIA_PORT_443_TCP=tcp://10.97.167.40:443
KUBIA_PORT_443_TCP_ADDR=10.97.167.40
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBIA_SERVICE_HOST=10.97.167.40
KUBIA_PORT=tcp://10.97.167.40:80
KUBIA_PORT_80_TCP=tcp://10.97.167.40:80
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
KUBIA_SERVICE_PORT=80
NPM_CONFIG_LOGLEVEL=info
NODE_VERSION=7.10.1
YARN_VERSION=0.24.4
HOME=/root
```

Connecting to the service through its FQDN

```
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-72d2n   1/1     Running   0          7m19s
kubia-fbt2q   1/1     Running   0          7m19s
kubia-t5lvg   1/1     Running   0          7m19s
cs571@ubuntu:~/app_js$ kubectl -it exec kubia-72d2n bash
root@kubia-72d2n:/# curl http://kubia.default.svc.cluster.local
You've hit kubia-t5lvg
root@kubia-72d2n:/# curl http://kubia.default
You've hit kubia-t5lvg
root@kubia-72d2n:/# curl http://kubia
You've hit kubia-t5lvg
root@kubia-72d2n:/# cat /etc/resolv.conf
nameserver 10.96.0.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
root@kubia-72d2n:/#
```

Understanding why you can't ping a service IP

```
root@kubia-72d2n:/# ping kubia
PING kubia.default.svc.cluster.local (10.101.50.41): 56 data bytes
^C--- kubia.default.svc.cluster.local ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
root@kubia-72d2n:/#
```

Connecting to services living outside the cluster

Introducing service endpoints

```
cs571@ubuntu:~/app_js$ kubectl describe svc kubia
Name:              kubia
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          run=kubia
Type:              ClusterIP
IP:                10.101.50.41
Port:              <unset>  80/TCP
TargetPort:        8080/TCP
Endpoints:         172.17.0.10:8080,172.17.0.8:8080,172.17.0.9:8080
Session Affinity:  None
Events:            <none>
```

```
cs571@ubuntu:~/app_js$ kubectl get endpoints kubia
NAME    ENDPOINTS                                         AGE
kubia   172.17.0.10:8080,172.17.0.8:8080,172.17.0.9:8080   5m57s
```

Creating a service without a selector

```
cs571@ubuntu:~/app_js$ kubectl get endpoints kubia
NAME    ENDPOINTS                                         AGE
kubia   172.17.0.10:8080,172.17.0.8:8080,172.17.0.9:8080   5m57s
cs571@ubuntu:~/app_js$ vim external-service.yaml
cs571@ubuntu:~/app_js$ cat external-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: external-service
spec:
  ports:
  - port: 80
cs571@ubuntu:~/app_js$ kubectl create -f external-service.yaml
service/external-service created
```

Creating an Endpoints resource for a service without a selector

```
cs571@ubuntu:~/app_js$ vim external-service-endpoints.yaml
cs571@ubuntu:~/app_js$ kubectl create -f external-service-endpoints.yaml
endpoints/external-service created
cs571@ubuntu:~/app_js$ cat external-service-endpoints.yaml
apiVersion: v1
kind: Endpoints
metadata:
  name: external-service
subsets:
  - addresses:
    - ip: 172.17.0.10
    - ip: 172.17.0.8
    - ip: 172.17.0.9
    ports:
    - port: 80
```

Creating an ExternalName service

```
cs571@ubuntu:~/app_js$ cat external-service-externalname.yaml
apiVersion: v1
kind: Service
metadata:
  name: external-service
spec:
  type: ExternalName
  externalName: api.somecompany.com
  ports:
  - port: 80
cs571@ubuntu:~/app_js$ kubectl create -f external-service-externalname.yaml
service/external-service created
```

Creating a NodePort service

```
cs571@ubuntu:~/app_js$ vim kubia-svc-nodeport.yaml
cs571@ubuntu:~/app_js$ cat kubia-svc-nodeport.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia-nodeport
spec:
  type: NodePort
  ports:
  - port: 80
    targetPort: 8080
    nodePort: 30123
  selector:
    run: kubia
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc-nodeport.yaml
service/kubia-nodeport created
cs571@ubuntu:~/app_js$ kubectl get svc
NAME               TYPE           CLUSTER-IP       EXTERNAL-IP           PORT(S)        AGE
external-service   ExternalName   <none>           api.somecompany.com   80/TCP         3m24s
kubernetes         ClusterIP      10.96.0.1        <none>                443/TCP        23d
kubia              ClusterIP      10.101.50.41     <none>                80/TCP         22m
kubia-nodeport     NodePort       10.103.144.117   <none>                80:30123/TCP   6s
```

Creating a LoadBalancer service

```
cs571@ubuntu:~/app_js$ vim kubia-svc-loadbalancer.yaml
cs571@ubuntu:~/app_js$ cat kubia-svc-loadbalancer.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia-loadbalancer
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 8080
  selector:
    run: kubia

cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc-loadbalancer.yaml
service/kubia-loadbalancer created
cs571@ubuntu:~/app_js$ kubectl get svc kubia-loadbalancer
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubia-loadbalancer   LoadBalancer   10.106.52.26   <pending>     80:31583/TCP   13s
cs571@ubuntu:~/app_js$
```

Enabling the Ingress add-on in Minikube

```
cs571@ubuntu:~/app_js$ minikube addons list
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| dashboard                   | minikube | disabled     |
| default-storageclass        | minikube | enabled ✔    |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | disabled     |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metrics-server              | minikube | disabled     |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ✔    |
| storage-provisioner-gluster | minikube | disabled     |
|-----------------------------|----------|--------------|
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ minikube addons enable ingress
🌟  The 'ingress' addon is enabled
cs571@ubuntu:~/app_js$ kubectl get pod --all-namespaces
NAMESPACE     NAME                                        READY   STATUS              RESTARTS   AGE
default       kubia-72d2n                                 1/1     Running             0          36m
default       kubia-fbt2q                                 1/1     Running             0          36m
default       kubia-t5lvg                                 1/1     Running             0          36m
kube-system   coredns-6955765f44-5ztmr                    1/1     Running             4          23d
kube-system   coredns-6955765f44-m6vvr                    1/1     Running             4          23d
kube-system   etcd-minikube                               1/1     Running             5          23d
kube-system   kube-apiserver-minikube                     1/1     Running             6          23d
kube-system   kube-controller-manager-minikube            1/1     Running             14         23d
kube-system   kube-proxy-hrfm2                            1/1     Running             3          23d
kube-system   kube-scheduler-minikube                     1/1     Running             12         23d
kube-system   nginx-ingress-controller-6fc5bcc8c9-45bsc   0/1     ContainerCreating   0          25s
kube-system   storage-provisioner                         1/1     Running             3          23d
```

Creating an Ingress resource

```
cs571@ubuntu:~/app_js$ vim kubia-ingress.yaml
cs571@ubuntu:~/app_js$ cat kubia-ingress.yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: kubia
spec:
  rules:
  - host: kubia.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: kubia-nodeport
          servicePort: 80
cs571@ubuntu:~/app_js$ kubectl create -f kubia-ingress.yaml
ingress.extensions/kubia created
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ kubectl get ingress
NAME    HOSTS               ADDRESS         PORTS   AGE
kubia   kubia.example.com   192.168.39.20   80      79s
```

```
cs571@ubuntu:~/app_js$ curl http://kubia.example.com
You've hit kubia-fbt2q
```

Mapping different services to different paths of the same host

```
cs571@ubuntu:~/app_js$ cat kubia-ingress.yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: kubia
spec:
  rules:
  - host: kubia.example.com
    http:
      paths:
      - path: /kubia
        backend:
          serviceName: kubia-nodeport
          servicePort: 80
      - path: /foo
        backend:
          serviceName: bar
          servicePort: 80
```

Mapping different services to different hosts

```
cs571@ubuntu:~/app_js$ cat kubia-ingress.yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: kubia
spec:
  rules:
  - host: foo.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: foo
          servicePort: 80
  - host: bar.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: bar
          servicePort: 80
```

Configuring Ingress to handle TLS traffic

Creating a TLS certificate for the Ingress

```
cs571@ubuntu:~/app_js$ openssl genrsa -out tls.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................+++++
.....................................+++++
e is 65537 (0x010001)
cs571@ubuntu:~/app_js$ openssl req -new -x509 -key tls.key -out tls.cert -days 360 -subj /CN=kubia.example.com
cs571@ubuntu:~/app_js$ kubectl create secret tls tls-secret --cert=tls.cert --key=tls.key
secret/tls-secret created
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ cat kubia-ingress-tls.yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: kubia
spec:
  tls:
  - hosts:
    - kubia.example.com
    secretName: tls-secret
  rules:
  - host: kubia.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: kubia-nodeport
          servicePort: 80
cs571@ubuntu:~/app_js$ kubectl create -f kubia-ingress-tls.yaml
ingress.extensions/kubia created
cs571@ubuntu:~/app_js$ kubectl get ingress
NAME    HOSTS               ADDRESS   PORTS     AGE
kubia   kubia.example.com             80, 443   4s
```

```
cs571@ubuntu:~/app_js$ curl -k -v https://kubia.example.com/kubia
*   Trying 192.168.39.20...
* TCP_NODELAY set
* Connected to kubia.example.com (192.168.39.20) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* ALPN, server accepted to use h2
* Server certificate:
*  subject: CN=kubia.example.com
*  start date: Mar  9 04:26:49 2020 GMT
*  expire date: Mar  4 04:26:49 2021 GMT
*  issuer: CN=kubia.example.com
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* Using Stream ID: 1 (easy handle 0x557f0516d580)
> GET /kubia HTTP/2
> Host: kubia.example.com
> User-Agent: curl/7.58.0
> Accept: */*
>
* Connection state changed (MAX_CONCURRENT_STREAMS updated)!
< HTTP/2 200
< server: openresty/1.15.8.2
< date: Mon, 09 Mar 2020 04:33:42 GMT
<
You've hit kubia-72d2n
* Connection #0 to host kubia.example.com left intact
cs571@ubuntu:~/app_js$
```

Adding a readiness probe to the pod template

```
cs571@ubuntu:~/app_js$ kubectl edit rc kubia
replicationcontroller/kubia edited
cs571@ubuntu:~/app_js$ kubectl get pds
error: the server doesn't have a resource type "pds"
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-72d2n   1/1     Running   0          66m
kubia-fbt2q   1/1     Running   0          66m
kubia-t5lvg   1/1     Running   0          66m
cs571@ubuntu:~/app_js$ kubectl delete pods --all
pod "kubia-72d2n" deleted
pod "kubia-fbt2q" deleted
pod "kubia-t5lvg" deleted


cs571@ubuntu:~/app_js$
cs571@ubuntu:~/app_js$
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-46676   0/1     Running   0          41s
kubia-7dbxq   0/1     Running   0          41s
kubia-z42hr   0/1     Running   0          41s
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ kubectl exec kubia-46676 -- touch /var/ready
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-46676   0/1     Running   0          2m11s
kubia-7dbxq   0/1     Running   0          2m11s
kubia-z42hr   0/1     Running   0          2m11s
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-46676   1/1     Running   0          2m35s
kubia-7dbxq   0/1     Running   0          2m35s
kubia-z42hr   0/1     Running   0          2m35s
cs571@ubuntu:~/app_js$
```

```
cs571@ubuntu:~/app_js$ kubectl exec kubia-7dbxq -- touch /var/ready
cs571@ubuntu:~/app_js$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-46676   1/1     Running   0          7m31s
kubia-7dbxq   1/1     Running   0          7m31s
kubia-z42hr   0/1     Running   0          7m31s
cs571@ubuntu:~/app_js$ kubectl exec kubia-46676 -- curl -s http://10.105.92.78
You've hit kubia-7dbxq
cs571@ubuntu:~/app_js$ kubectl exec kubia-46676 -- curl -s http://10.105.92.78
You've hit kubia-7dbxq
cs571@ubuntu:~/app_js$
```

Creating a headless service

```
cs571@ubuntu:~/app_js$ vim kubia-svc-headless.yaml
cs571@ubuntu:~/app_js$ cat kubia-svc-headless.yaml
apiVersion: v1
kind: Service
metadata:
  name: kubia-headless
spec:
  clusterIP: None
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: kubia
cs571@ubuntu:~/app_js$ kubectl create -f kubia-svc-headless.yaml
service/kubia-headless created
cs571@ubuntu:~/app_js$ kubectl get svc
NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP           PORT(S)        AGE
external-service     ExternalName   <none>          api.somecompany.com   80/TCP         51m
kubernetes           ClusterIP      10.96.0.1       <none>                443/TCP        23d
kubia                ClusterIP      10.105.92.78    <none>                80/TCP         5m14s
kubia-headless       ClusterIP      None            <none>                80/TCP         7s
kubia-loadbalancer   LoadBalancer   10.106.52.26    <pending>             80:31583/TCP   44m
kubia-nodeport       NodePort       10.100.43.75    <none>                80:30123/TCP   15m
```

```
cs571@ubuntu:~/app_js$ kubectl exec kubia-z42hr -- touch /var/ready
cs571@ubuntu:~/app_js$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
kubia-46676   1/1     Running   0          12m
kubia-7dbxq   1/1     Running   0          12m
kubia-z42hr   1/1     Running   0          12m
```

Discovering pods through DNS

```
cs571@ubuntu:~/app_js$ kubectl run dnsutils --image=tutum/dnsutils --generator=run/v1 --command -- sleep infinity
kubectl run --generator=run/v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
replicationcontroller/dnsutils created
cs571@ubuntu:~/app_js$ kubectl exec dnsutils nslookup kubia-headless
Server:         10.96.0.10
Address:        10.96.0.10#53
```

```
cs571@ubuntu:~/app_js$ kubectl exec dnsutils nslookup kubia
Server:         10.96.0.10
Address:        10.96.0.10#53

Name:   kubia.default.svc.cluster.local
Address: 10.105.92.78
```