1. replicated load balanced dictionary service

vi dictionary-deploy.yaml

```
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
dictionary-deploy.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: dictionary-server
  namespace: tina
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: dictionary-server
    spec:
      containers:
      - name: server
        image: brendanburns/dictionary-server
        ports:
        - containerPort: 8080
        readinessProbe:
          httpGet:
            path: /ready
            port: 8080
          initialDelaySeconds: 5
          periodSeconds: 5
```

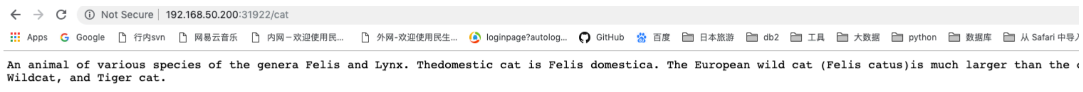
```
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
kubectl create -f dictionary-deploy.yaml
deployment.extensions "dictionary-server" created

wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
```

```
kubectl get deployment -n tina
NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
dictionary-server   3         3         3            0           17s

wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
kubectl get pods -n tina
NAME                                READY     STATUS              RESTARTS   AGE
adapter-example                     2/2       Running             0          45m
dictionary-server-65cc45d7f4-qzcwg   0/1       Running             0          55s
dictionary-server-65cc45d7f4-s94vh   0/1       Running             0          55s
dictionary-server-65cc45d7f4-sj64n   0/1       ContainerCreating   0          55s

vi dictionary-service.yaml
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
dictionary-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: dictionary-server-service
  namespace: tina
spec:
  type: NodePort
  selector:
    app: dictionary-server
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 31922

wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
kubectl create -f dictionary-service.yaml
service "dictionary-server-service" created
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
```

kubectl get services -n tina

```
NAME                     TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
adapter-redis-service      NodePort   10.109.130.100   <none>
9121:31921/TCP   58m
dictionary-server-service   NodePort   10.97.41.62     <none>
8080:31922/TCP   9s
```

访问http://192.168.50.200:31922/cat



```
An animal of various species of the genera Felis and Lynx. Thedomestic cat is Felis domestica. The European wild cat (Felis catus)is much larger than the
Wildcat, and Tiger cat.
```

2. Caching layer

```
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
default.vcl
vcl 4.0;
backend default {
  #.host = "dictionary-server-service";
  .host = "192.168.50.200";
  .port = "31922";
}
kubectl create configmap varnish-config --from-file=default.vcl -n tina
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
kubectl create configmap varnish-config --from-file=default.vcl -n tina
configmap "varnish-config" created
```

```
vi varnish-deploy.yaml
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
varnish-deploy.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: varnish-cache
  namespace: tina
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: varnish-cache
    spec:
      containers:
      - name: cache
        resources:
          requests:
            memory: 2Gi
        image: brendanburns/varnish
        command:
        - varnishd
        - -F
        - -f
        - /etc/varnish-config/default.vcl
        - -a
        - 0.0.0.0:8080
        - -s
        # This should match the 'memory' request above.
        - malloc,2G
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: varnish
          mountPath: /etc/varnish-config
      volumes:
      - name: varnish
        configMap:
          name: varnish-config
```

```
kubectl create -f varnish-deploy.yaml
kubectl create -f varnish-service.yaml
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
varnish-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: varnish-service
  namespace: tina
spec:
  selector:
    app: varnish-cache
  type: NodePort
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
      nodePort: 31923
```

访问：http://192.168.50.200:31923/dog

3.Expanding the caching layer

generate a certificate

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /media/ssd/wenqiao/distributed-systems/replicated-load-balanced/ssl/server.key -out /media/ssd/wenqiao/distributed-systems/replicated-load-balanced/ssl/server.crt
```

没有sudo权限，我在我本地生成再上传：

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ~/Downloads/server.key -out ~/Downloads/server.crt
wenqiaodeMacBook-Pro-2:data wenqiao$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ~/Downloads/server.key -out
```

```
~/Downloads/server.crt
Password:
Generating a 2048 bit RSA private key
..+++
.............................................................................+++
writing new private key to '/Users/wenqiao/Downloads/server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:CN
State or Province Name (full name) []:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:localhost
Email Address []:
```

Upload crt and key as a secret to k8s：

```
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced/ssl$
kubectl create secret tls ssl --cert=server.crt --key=server.key -n tina
secret "ssl" created
```

```
vi nginx.conf
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
nginx.conf
events {
  worker_connections  1024;
}
```

```
http {
  server {
    listen 443 ssl;
    server_name localhost;
    ssl on;
    ssl_certificate         /etc/certs/tls.crt;
    ssl_certificate_key     /etc/certs/tls.key;
    location / {
        proxy_pass http://192.168.50.200:31923;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
    }
  }
}
```

Transform nginx file into a configmap object

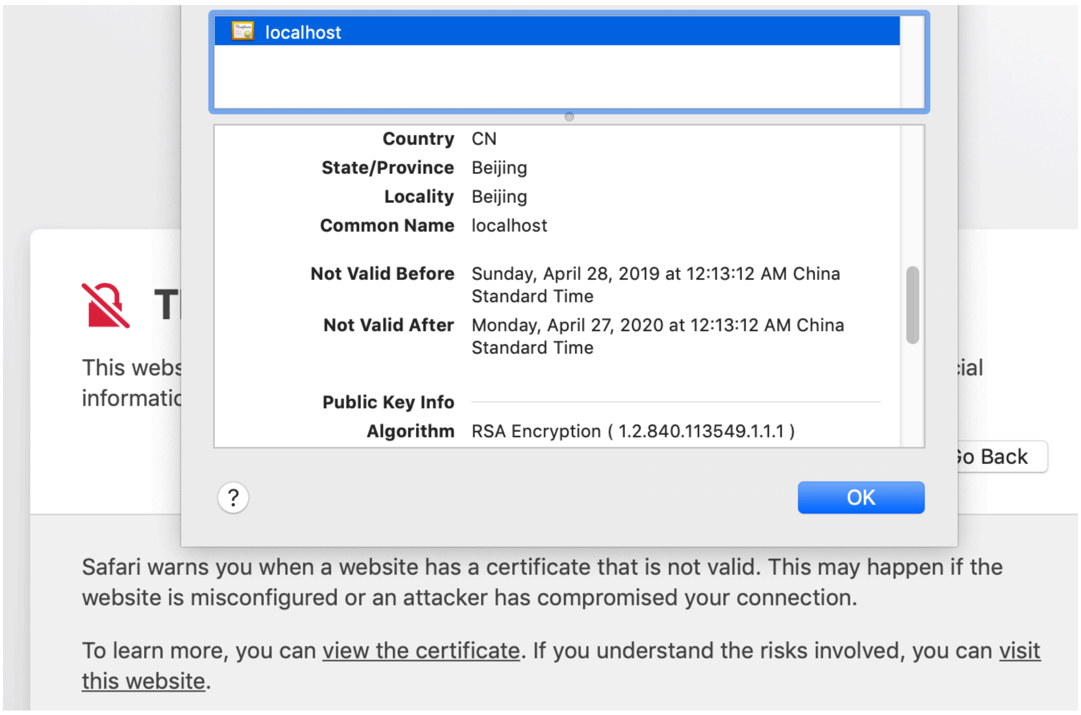
```
kubectl create configmap nginx-conf --from-file=nginx.conf -n tina
```

```
vi nginx-deploy.yaml
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
nginx-deploy.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ssl
  namespace: tina
spec:
  replicas: 4
  template:
    metadata:
      labels:
        app: nginx-ssl
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 443
        volumeMounts:
        - name: conf
          mountPath: /etc/nginx
        - name: certs
          mountPath: /etc/certs
      volumes:
      - name: conf
        configMap:
          # This is the ConfigMap for nginx we created previously
          name: nginx-conf
      - name: certs
        secret:
          # This is the secret we created above
          secretName: ssl
```

```
kubectl create -f nginx-deploy.yaml
vi nginx-service.yaml
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$ cat
nginx-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: tina
spec:
  selector:
    app: nginx-ssl
  type: LoadBalancer
  ports:
    - protocol: TCP
      port: 443
      targetPort: 443
```

```
wenqiao@gnuhpc-pc:~/distributed-systems/replicated-load-balanced$
kubectl describe service nginx-service -n tina
Name:                     nginx-service
Namespace:                tina
Labels:                   <none>
Annotations:              <none>
Selector:                 app=nginx-ssl
Type:                     LoadBalancer
IP:                       10.97.42.20
Port:                     <unset>  443/TCP
TargetPort:               443/TCP
NodePort:                 <unset>  32135/TCP
Endpoints:                10.244.0.249:443,10.244.1.203:443,10.244.2.135:443 + 1
more...
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

访问：https://192.168.50.200:32135/dog
会提示证书is not valid

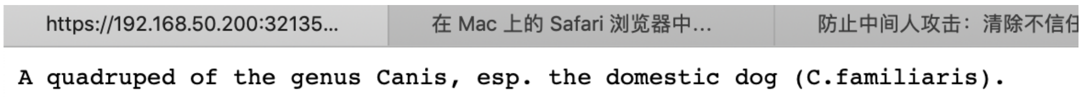

localhost

Country	CN
State/Province	Beijing
Locality	Beijing
Common Name	localhost
Not Valid Before	Sunday, April 28, 2019 at 12:13:12 AM China Standard Time
Not Valid After	Monday, April 27, 2020 at 12:13:12 AM China Standard Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)

Safari warns you when a website has a certificate that is not valid. This may happen if the website is misconfigured or an attacker has compromised your connection.

To learn more, you can view the certificate. If you understand the risks involved, you can visit this website.

继续浏览网页，可以查询词典：


https://192.168.50.200:32135...
在 Mac 上的 Safari 浏览器中...
防止中间人攻击：清除不信任

```
A quadruped of the genus Canis, esp. the domestic dog (C.familiaris).
```