

Exercise

1. Write a YAML file that deploys the following:
 - a. two applications that is routed via ingress controller
 - b. the application is about cheese store.
 - c. the application address is: `https://ams.cheese.com`
 - d. the application will have two deployments:
 - i. cheddar
 - ii. gauda
 - e. setup an ingress controller that will route the following:
 - i. `https://ams.cheese.com/cedder` -> will go to the cheddar deployment
 - ii. `https://ams.cheese.com/gauda` -> will go to the gauda deployment
2. Mount as config map the attached file to an nginx container under the path `/usr/share/nginx/html`
- 3.
4. BONUS: write your images that will return images of those cheeses and use them in your YAML files.

What to send me?

1. YAML files
2. Screenshots from chrome browser

Answers

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: cheddar
spec:
  replicas: 1
  selector:
    matchLabels:
      app: cheddar
  template:
    metadata:
      labels:
        app: cheddar
    spec:
      containers:
        - name: cheddar
          image: nginxdemos/hello:plain-text
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: cheddar-svc
spec:
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
  selector:
    app: cheddar
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: gauda
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gauda
```

```
template:
  metadata:
    labels:
      app: gauda
  spec:
    containers:
      - name: gauda
        image: nginxdemos/hello:plain-text
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: gauda-svc
  labels:
spec:
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
  selector:
    app: gauda
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cheese-ingress
spec:
  tls:
    - hosts:
        - ams.cheese.com
      secretName: cheese-secret
  rules:
    - host: ams.cheese.com
      http:
        paths:
          - path: /gauda
            backend:
              serviceName: gauda-svc
              servicePort: 80
          - path: /cheddar
            backend:
              serviceName: cheddar-svc
              servicePort: 80
apiVersion: v1
kind: Secret
```

metadata:

name: cheese-secret

type: Opaque

data:

tls.crt:

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURMakNDQWhZQ0NRREFPRjI0THN
hWFdqQU5CZ2txaGtpRzl3MEJBUXNGQURCYU1Rc3dDUVIEVFR0V3SIYKVXpFTE1Ba0d
BMVVFQ0F3Q1EwRXhJVEFmQmdOVkJBb01HRWx1ZEEdWeWJtVjBJRmRwWkdkcGRITWd
VSFI1SUV4MApaREViTUJrR0ExVUVBd3dTWtJGbvPnWxIR0Z0Y0d4bExtTnZiU0FnTUI0
WERURTRNRGt4TWpFMk1UVXpOVm9YCKRUSXpNRGt4TVRFMk1UVXpOVm93V0RFE1
Ba0dBMVVFQmhNQ1ZWtXhDekFKQmdOVkJBZ01Ba05CTVNFd0h3WUQKVIFRS0RCaEpi
blJsY201bGRDQlhhV1JuYVhSeklGQjBIU0JNZEEdReEdUQVhCZ05WQkFNTUVHTmhabVV1
WlhoaApiWEJzWIM1amlyMHdnZ0VpTUEwR0NTcUdTSWlZRFFFQkFRVUFBNEICRhdBd2d
nRUtBb0ICQVFDcDZLbkdzeTgxGnAwanVKL2N5ayt2Q0FtbHNmanRGTTJtdVpOSzBLdGVjc
UcyZmpXUWI1NXhRMVIGQTTJYT1N3SEFZdINkd0kyaloKcnVXOHFYWENMMnJiNENaQ0Z4
d3BWRUNyY3hkam0zdGVWaVJYVnNZSW1tSkhQUFN5UWdwaW9iczl4N0RsTGM2SQpCQ
TBaalVPeWwwUHFHOVNKZXhNVjczV0IJYTVyRFZTRjJyNGtTa2JBajREY2o3TFhIRmxWWE
gySTVYd1hDcHRDCm42N0pDZzQyZitrOHdnemNSVnA4WFprWldaVmp3cTISVUtEWG1GQj
JZeU4xWEVXZFowZXdsdUtZVUpSc202OTIKc2tPcktRajB2a29QbjQxRUUvK1RhVkvVwcUxU
Um9VWTNyemc3RGtkemZkQml6Rk8yZHNQTkZ4MkNXMGpYa05MdgpLbzl1Q1pyT2hYQUh
BZ01CQUFFd0RRWUplb1pJaHZjTkFRRUxCUUFEZ2dFQkFLSEZDY3IPalp2b0hzd1VCTW
RMCiJKSElImgzczFdGeW5acS9MdVvvdnNWQTU4QjBDZzdCRWZ5NXZXVIZycTVSSWt2N
GxaODFOMjI4MjFkMUpiNnIKalNuUXgrRFhDTy9USkVWNWxTQ1VwSUd6RVVZYVQZ1J5
anNNL05VZENKOHVIVmhaSitTNkZBK0NuT0Q5cm4yaQpaQmVQQ0k1ckh3RVh3bm5sOHl3
aWozdnZRNXplSXV5QmdsV3lvUXI1aTlmalBwd1dVdVtNG52NVNNRzl6Q1Y3CIBwdXd2dW
F0cWpPMTlwOEJqZkUvY1pISWc4SHc5bXZXOXg5YqtJU1J1TURFN2lvZzZPY0s3TEdUTH
dsRnh2QTgKN1dqRWVxdW5heUlwae1oS1JYVmYxTjM0OWVVOOThFejM4Zk9USFRQYmR
KakZBL1BjQytHeW1IK2IHdDVPUPWRGaAp5UkU9Ci0tLS0tRU5EIEIENFUIRJRkIDQVRFLS0tLS
OK

tls.key:

LS0tLS1CRUdJTiBSU0EgUFJJVkfURSBLRVktLS0tLQpNSUJfY3dJQkFBS0NBUEVBCWVpc
Cs3TXZOYWRJN2ImM01wUHJ3Z0pwYkg0N1JUTnBybVVRTdENyWG5LaHRuNDFrCkcrZWN
VTldCUU5semtzQndHTDBuY0NObzJhN2x2S2wxd2k5cTlrQW1RaGNjS1ZSQXEzTVhZNXQ
3WGxZa1YxYkckQ0pwaVJ6ejBza0ILWFXHN1BjZxc1UzNPauFRTkdZMURzcGRENmh2VWI
Yc1RGZTksaUNHdWF3MVVoZHErSkVwRwp3SStBM0kreTEzaFpWVng5aU9WoeZ3cWJRc
Ct1eVFvT05uL3BQTUINM0VWYWZGMlpHVm1WWThLdVWQ2cxNWhRCmRtTWpkVnhGbl
dkSHNFYmltRkNaYkp1dmRySkRxeWtJOuW1S0Q1K05SQIAvazJsUkthaTAwYUZHTjY4NE9
3NUgKYzMzUVIzeFR0bmJEelJjZGdsdEkxNURTN3lxTnVRbWF6b1Z3QndJREFRQUJBb0IC
QVFDUFNkU1luUXRTUHIxbApGZIZGcFRPc29PWVJoZjhzSStpYkZ4SU91UmF1V2VoaEp4Z
G01Uk9ScEF6bUNMeUw1VmhdqEptZTlyM2dMcncyCk45OUVqVUtiL1ZPbVp1RHNCYzZvQ
0Y2UU5SNThkejhbK9SVGV3Y290c0pSMXBuMWhobG5SNUhxSkpCSmFzazEKWkVuVVFm
Y1hackw5NGxvOUplM0UrVXFqbzFGRnM4eHhFOHdvUEJxalpzVjdwUjVhZ0MzTGh4bndMU
0V4eUZvNApjeGI5U09HNU9tQUpeIN0Rm9RMkdKT2VzOHJKNXFmZHZ5dGdnOXhiTGFR
TC94MGtwUTYyYm9GTUJEZHFPZVBXCKtmUDV6WjYvMDcvdnBqNDh5QTFRMzJQem9id
WJzQkxkM0tjbMyamZtMUU3cHJ0V2wrSmVPRmlPem5CUUZKYk4KNHFQVIJ6NWhBb0dC
QU50V3I4aE5DU0x1NFArWGdLeWNRbGpKNkY1NjY4Zk5qNUN6Z0ZScUowOXpuMFRsc05y
bwpGVExaY3hEcW5SM0hQWU00MkpFUmgySi9xREZaeW5SUW8zY2czb2VpdIVkQIZHWT

grRkxVzBxZHVIL0w5K3l1CmVKT1pUUTVYbUdHcDZyNmpleHltY0ppbS9Pc0IzWm5ZT3BPc
mxEN1NQBbUJ2ek5MazRNRjZneGJYQW9HQkFNWk8KMHA2SGJCbWNQMHRqRIhmY0tFN
zdJbUxtMHNBRzR1SG9VeDBIUgovMnFyblRuT0JCTkU0TXZnRHVUSnp5K2NhVQprOFJxb
WRIQ2JleIRINmZ6WXEvOWI0OHNaNzdLVk4xcWtiSWN1YytSVHhBOW5OaDFUanNSbmU3
NFowajFGQ0xrCmhIY3FIMHJpN1BZU0tIVEU4RnZGQ3haWWRidUI4NENtWmlodnhicFJBb0
dBSWJqcWfNWVBWUXVrbENkYTVTNzkKWVNGSjFKelpIMUtqYS8vdER3MXpGY2dWQ0th
MzFqQXcjaXowZi9sU1JxM0hTMUdHR21lemhQVIRpcUxmZVpxYwpSMGILYmhnYk9jVIZrSk
ozSzB5QXILd1BUdW14S0haNnpJbVpTMGMwYW0rUIk5WUdxNVQ3WXJ6cHpjZnZwaU9VC
mZmZTNSeUZUN2NmQ21mb09oREN0enVrQ2dZQjMwb0xDMVJMRk9ycW40M3ZDUzUxe
mM1em9ZNDR1QnpzcHd3WU4KVHd2UC9FeFdNZjNWSnJEakJDScTULzZzeXNIUGJKRuit
bHpNK0I3eXRGcEFOZmlJWEV0LzQ4WGY2ME54OGdXTQp1SHI4Wlp4L05LdER3MFY4dlg
xUE9ucTJBNWVpS2ErOGpSQVJZS0pMWU5kZkR1d29seHZHNmJaaGtQaS80RXRUCjNZM
ThzUUtCZ0h0S2JrKzdsTkPwZXN3WEU1Y1VHNkVEVXNEZS8yVWE3ZlhwN0ZjanFCRW9h
cDFMU3crNIRYcDAKWmdybUtFOEFsek00NytFSkhVdmlpcS9udXBFMTVnMGtKVzNzeWhw
VTI6WkxPN2x0QjBLSWtPOVpSY21Vam84UQpjcExsSE1BcWJMSjhXWUdKQ2toaVd4eWFs
NmhZVHIXWTRjVmtDMHh0VGwvaFVFOUItktvCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS
0tLS0tCg==

```
apiVersion: v1
kind: Service
metadata:
  name: fe-gate
spec:
  selector:
    name: nginx
  type: NodePort
  ports:
    - name: foo # Actually, no port is needed.
      port: 80
      targetPort: 80
```

```
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    name: nginx
spec:
  containers:
    - name: nginx
      image: nginx
      volumeMounts:
        - name: config-volume
          mountPath: /usr/share/nginx/html
```

volumes:

- name: config-volume

configMap:

- name: indexhtml