

Table of Contents

Stack used	1
Versions	1
Structure	1
Run as Go Application	2
Build docker image	3
Run container	3
Minikube	3
Helm charts	3
Deploy charts on minikube with Helmfile	4
Verify API's are accessible	5
Destroy resources	5

Stack used

- [gorilla-mux](#) - Request router and dispatcher
- [gorm](#) - ORM library for Golang
- [sqlite3](#) - sqlite3 driver for go using database/sql
- [helm](#) - package manager for K8s
- [helmfile](#) - declarative spec for deploying helm charts
- [minikube](#) - local k8s cluster

Versions

```
Go Version - go1.16.6 darwin/amd64
helm version - v3.6.3
helmfile version - v0.140.0
```

Structure

```

.
├── Dockerfile
├── README.md
├── Solution.adoc
├── charts
│   ├── app
│   │   ├── Chart.yaml
│   │   ├── charts
│   │   ├── templates
│   │   │   ├── connfigmap.yaml
│   │   │   ├── deployment.yaml
│   │   │   └── service.yaml
│   │   └── values.yaml
│   ├── ingress
│   │   ├── Chart.yaml
│   │   ├── charts
│   │   ├── templates
│   │   │   └── ingress.yaml
│   │   └── values.yaml
│   └── charts-values
│       ├── app.yaml
│       └── ingress.yaml
├── database
│   ├── configs.go
│   └── db.go
├── go.mod
├── go.sum
├── handlers
│   └── handlers.go
├── helmfile-deploy.sh
├── helmfile-destroy.sh
├── helmfile.yaml
├── main.go
├── models
│   └── config.go
├── routers
│   └── routers.go
└── sqlite.db

```

Run as Go Application

- Set `SERVE_PORT` env variable
- run/build with `--tags="json1"` to enable `sqlite3` JSON extension

```
$ export SERVE_PORT=8090
$ go run --tags="json1" main.go

$ go build --tags="json1" main.go
```

Build docker image

```
$ docker build -t config-service -t karant/config-service:latest .
```

```
Step 1/9 : FROM golang:latest as builder
Successfully tagged config-service:latest
Successfully tagged karant/config-service:latest
```

Image is available on <https://hub.docker.com>

Run container

```
$ docker run -d -p 8080:8090 -e SERVE_PORT=8090 --name config-service-container
karant/config-service
Unable to find image 'karant/config-service:latest' locally
latest: Pulling from karant/config-service

$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
e2e6430a2dd0	karant/config-service	"/main"	About a minute ago	Up About a minute
	8080/tcp, 0.0.0.0:8080->8090/tcp	config-service-container		

Minikube

*Note - To enable minikube ingress addons, start minikube with flags **--vm=true** **--driver=hyperkit** otherwise due to networking limitations of driver docker on darwin, ingress addon is not supported.*

```
$ minikube start --vm=true --driver=hyperkit

$ minikube addons enable ingress
The 'ingress' addon is enabled
```

Helm charts

- app - contains helm templates for deployment, service and configMap
- ingress - contains helm templates for nginx ingress

Deploy charts on minikube with Helmfile

Helmfile makes easy to deploy multiple charts and take care of dependencies

To apply helmfile run `./helmfile-deploy.sh` it will also update `/etc/hosts` to add CNAME entry

```
$ ./helmfile-deploy.sh
```

verify created resources

```
kubectl get svc,po,deployment,ingress
```

```
□ □ □ 10192
```

NAME	EXTERNAL-IP	PORT(S)	AGE	TYPE	CLUSTER-IP
service/config-service-app				LoadBalancer	10.106.204.196
<pending>		9090:30769/TCP	2m21s		
service/ingress-backend-nginx-ingress-controller				LoadBalancer	10.97.224.70
<pending>		80:30089/TCP,443:32512/TCP	2m17s		
service/ingress-backend-nginx-ingress-default-backend				ClusterIP	10.104.231.213
<none>		80/TCP	2m17s		
service/kubernetes				ClusterIP	10.96.0.1
<none>		443/TCP	121m		

NAME	RESTARTS	AGE	READY	STATUS
pod/config-service-app-76df44dc6b-2zs2g			1/1	Running
0		2m21s		
pod/config-service-app-76df44dc6b-7vbgm			1/1	Running
0		2m21s		
pod/ingress-backend-nginx-ingress-controller-56cbbfdb96-5ftpx			1/1	Running
0		2m17s		
pod/ingress-backend-nginx-ingress-default-backend-58b4f9b66-jbqhs			1/1	Running
0		2m17s		

NAME	AVAILABLE	AGE	READY	UP-TO-DATE
deployment.apps/config-service-app			2/2	2
2m21s				
deployment.apps/ingress-backend-nginx-ingress-controller			1/1	1
2m17s				
deployment.apps/ingress-backend-nginx-ingress-default-backend			1/1	1
2m17s				

NAME	CLASS	HOSTS	ADDRESS
ingress.networking.k8s.io/config-service-ingress	<none>	config-service	
192.168.64.2 80		2m21s	

Verify API's are accessible

```
curl -X GET http://config-service/configs/burger-nutrition | jq .
```

```
{
  "ID": 1,
  "CreatedAt": "2021-07-26T00:58:54.596612+02:00",
  "UpdatedAt": "2021-07-26T00:58:54.596612+02:00",
  "DeletedAt": null,
  "name": "burger-nutrition",
  "metadata": {
    "allergens": {
      "eggs": "true",
      "nuts": "false",
      "seafood": "false"
    },
    "calories": 230,
    "carbohydrates": {
      "dietary-fiber": "4g",
      "sugars": "1g"
    },
    "fats": {
      "saturated-fat": "0g",
      "trans-fat": "1g"
    }
  }
}
```

Destroy resources

To destroy run `./helmfile-destroy.sh`

```
$ ./helmfile-destroy.sh
```