

Using Helm to Deploy Micro-Services Application

Table of Contents

I. Introduction.....	3
I.1. Target.....	3
I.2. Demo Overview.....	3
I.3. Prequiresites.....	3
II. Instruction.....	4
II.1. General steps.....	4
II.2. Execution.....	4
III. Frequently Asked Questions.....	6
III.1. Demo source code.....	6
IV. References.....	7

I. Introduction

I.1. Target

The document shows readers how to deploy applications with **Helm**.

I.2. Demo Overview

In using Helm, we can simplify the process of application deployment.

In this demo, we will create the application on EKS.

The project used in this demo consists of 3 services:

- database
- backend
- frontend

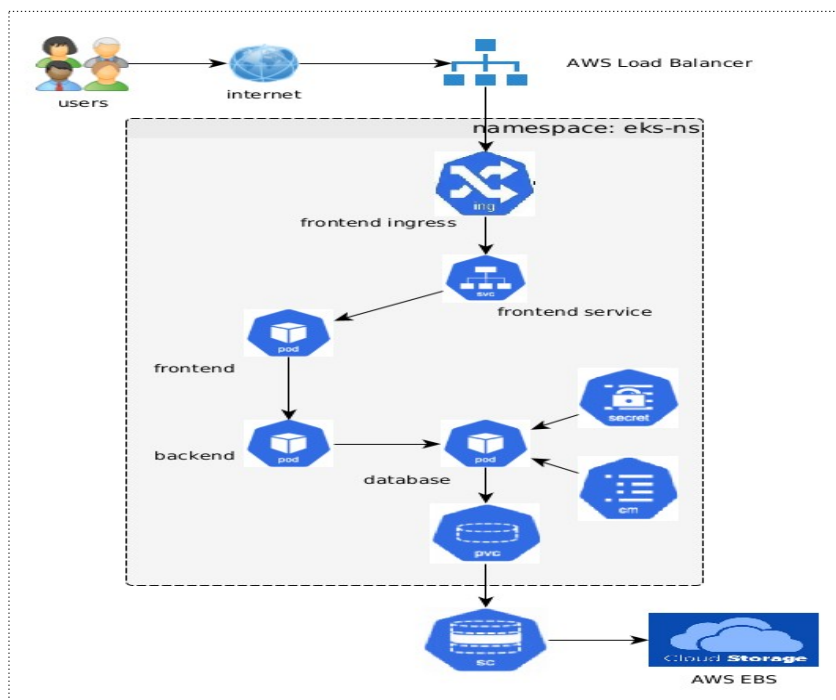


Figure 1: Application Components

We will deploy a helm chart that consists of 3 subcharts (frontend, backend and mongo) and 1 dependency (NGINX controller). They can be found in “[Demo source code](#)”

Note: For the production environment, the database should be an external database service.

I.3. Prequisites

- Helm CLI is locally installed
- EKS cluster is already created (refer to [Create EKS Cluster](#)).
- Kubernetes CLI gets configured (kubectl) (refer [Configure kubectl](#))

II. Instruction

II.1. General steps

In order to get the application running, the following steps are required:

- Deploy the helm chart
- Verify the chart installation

II.2. Execution

- **Deploy the helm chart:**

in the root folder

helm install my-release .

```
NAME: my-release
LAST DEPLOYED: Tue Sep  5 16:50:42 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

- **Verify the chart installation**

verify the release

helm list

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
my-release	default	1	2023-09-05 17:13:40.864098954 +0700 +07	deployed	apps-0.1.0	1.16.0

verify the application resources

kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/backend-app-58f958b745-cvzr5	1/1	Running	0	2m2s
pod/backend-app-58f958b745-qzspk	1/1	Running	0	2m2s
pod/backend-app-58f958b745-tk7zw	1/1	Running	0	2m2s
pod/database-0	1/1	Running	0	2m2s
pod/frontend-app-79dc46ddb5-7w4np	1/1	Running	0	2m2s
pod/frontend-app-79dc46ddb5-94thd	1/1	Running	0	2m2s
pod/frontend-app-79dc46ddb5-rsrhb	1/1	Running	0	2m2s
pod/my-release-ingress-nginx-controller-b9f8867b6-9nzc	1/1	Running	0	2m2s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/backend	ClusterIP	10.100.39.61	<none>	3000/TCP	2m3s
service/frontend	ClusterIP	10.100.197.186	<none>	3000/TCP	2m3s
service/kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	5h4m
service/mongo	ClusterIP	10.100.140.218	<none>	27017/TCP	2m3s
service/my-release-ingress-nginx-controller	LoadBalancer	10.100.113.93	a6c4199044e4346db0057b1a6be194d-1726000723.ap-southeast-1.elb.amazonaws.com	80:31017/TCP,443:32542/TCP	2m3s
service/my-release-ingress-nginx-controller-admission	ClusterIP	10.100.116.158	<none>	443/TCP	2m3s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/backend-app	3/3	3	3	2m4s
deployment.apps/frontend-app	3/3	3	3	2m4s
deployment.apps/my-release-ingress-nginx-controller	1/1	1	1	2m4s

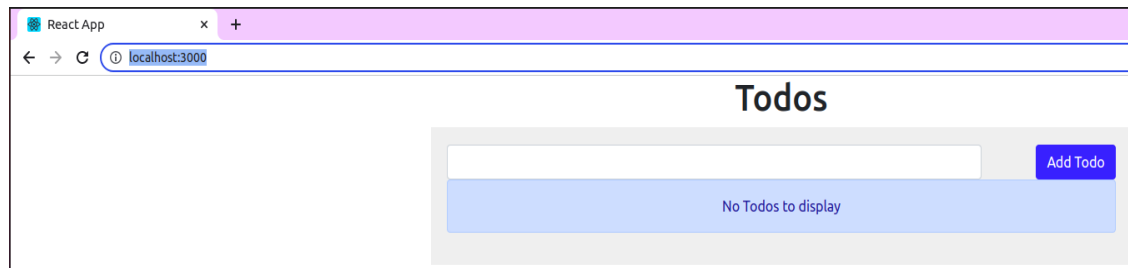
NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/backend-app-58f958b745	3	3	3	2m5s
replicaset.apps/frontend-app-79dc46ddb5	3	3	3	2m5s
replicaset.apps/my-release-ingress-nginx-controller-b9f8867b6	1	1	1	2m5s

NAME	READY	AGE
statefulset.apps/database	1/1	2m5s

expose frontend service to access application
kubectl port-forward service/frontend 3000:3000

```
hatnguyencanh@vnlap03333:~/Documents/K8s/DEMO/k8s$ kubectl port-forward service/frontend 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
```

open browser to access application at URL: localhost:3000

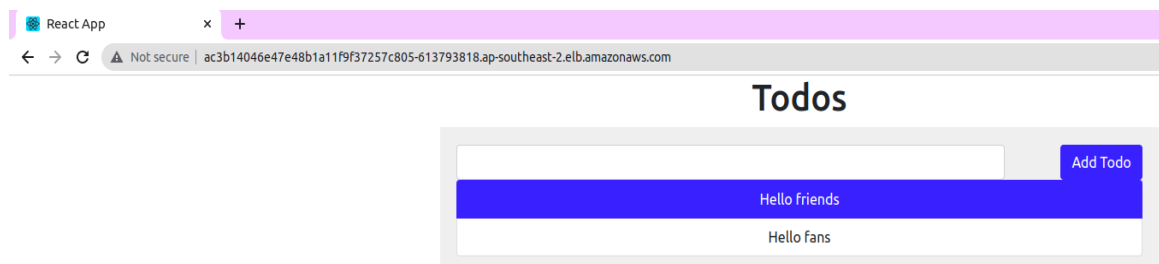


verify application ingress
kubectl get ingress -o wide

```
hatnguyencanh@vnlap03333:~/Documents/K8s/DEMO/k8s$ kubectl get ingress -o wide
```

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
ingress	nginx	*	ac3b14046e47e48b1a11f9f37257c805-613793818.ap-southeast-2.elb.amazonaws.com	80	42s

open browser to access application at URL created by ingress controller (e.g:
ac3b14046e47e48b1a11f9f37257c805-613793818.ap-southeast-2.elb.amazonaws.com)



III. Frequently Asked Questions

III.1. Demo source code

- <https://github.com/nashtech-garage/azure-devops-ci-cd/tree/main/helm>

IV. References

- Source code for application: <https://github.com/docker/awesome-compose/tree/master/react-express-mongodb>