详细地址：

<https://docs.konghq.com/kubernetes-ingress-controller/latest/guides/getting-started/>

**安装kong ingress**

# kubectl apply -f https://bit.ly/k4k8s

使用如下命令查看其创建的service

root@k8s-master:~# kubectl get services -n kong

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kong-proxy LoadBalancer 10.1.85.159 <pending> 80:32268/TCP,443:32148/TCP 10m

kong-validation-webhook ClusterIP 10.1.93.55 <none> 443/TCP 10m

我们设置一个环境变量（10.1.85.159是上面输出的地址）

# export PROXY\_IP=10.1.85.159

**创建示例service**

# kubectl apply -f <https://bit.ly/echo-service>

**创建ingress资源**

# echo "

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: demo

spec:

ingressClassName: kong

rules:

- http:

paths:

- path: /foo

pathType: ImplementationSpecific

backend:

service:

name: echo

port:

number: 80

" | kubectl apply -f -

**访问如下url**

我们如下url即可看到kong将其重定向到后端服务器

# curl -i $PROXY\_IP/foo

**安装konga**

这不是必须的，但图像界面管理起来比较方便，接下来我们在k8s的master节点下安装konga

1. 更改kong-ingress暴露的端口

kong-ingress没有暴露kong admin api，我们需要在集群内部暴露api

所以我们需要更改其Deployment：

# kubectl edit deployments/ingress-kong -n kong

将

- name: KONG\_ADMIN\_LISTEN

value: 127.0.0.1:8444 ssl

修改为

- name: KONG\_ADMIN\_LISTEN

value: 0.0.0.0:8001, 0.0.0.0:8444 ssl

然后我们新建一个konga.yaml文件

kind: Service # 资源类型

apiVersion: v1 # 资源版本

metadata: # 元数据

name: konga # 资源名称

namespace: kong # 命名空间

spec: # 描述

selector: # 标签选择器，用于确定当前service代理哪些pod

app: ingress-kong

type: ClusterIP # 类型，值NodePort允许外部访问，值 ClusterIP 只能在集群内访问

ports: # 端口信息

- name: proxy-admin

port: 8001 # service端口

targetPort: 8001 # pod端口

- name: proxy-admin-ssl

port: 8444 # service端口

targetPort: 8444 # pod端口

执行如下命令应用service

# kubectl apply -f konga.yaml

查看我们konga service的地址

root@k8s-master:~# kubectl get services -A

NAMESPACE NAME TYPE CLUSTER-IP PORT(S)

kong konga ClusterIP 10.1.184.146 8001/TCP,8444/TCP

可以看到地址为10.1.184.146

注：之所以要新建一个Service是因为kong-ingress会暴露kong admin api给外部，所以我们新建一个不暴露api的service

1. 创建docker网络

# docker network create -d bridge kongnet

1. 安装postgres数据库

# sudo mkdir -p /opt/postgres/data

# sudo docker run -d --name postgres \

--network kongnet \

--restart=always \

-e POSTGRES\_PASSWORD=kong123 \

-p 5432:5432 \

-v /home/software/postgres/data:/var/lib/postgresql/data \

postgres:9.6

1. 进入数据库创建用户和数据库

// 进入容器

# sudo docker exec -it postgres /bin/bash

// 进入数据库

# psql -U postgres

// 创建用户，创建数据库

# CREATE USER kong WITH PASSWORD 'kong123';

# CREATE DATABASE konga OWNER kong;

// 退出数据库，退出容器

# \q

# exit

1. 安装konga

# docker run -d --name konga \

--network kongnet \

--restart=always \

-p 1337:1337 \

-e "DB\_ADAPTER=postgres" \

-e "DB\_HOST=postgres" \

-e "DB\_PORT=5432" \

-e "DB\_USER=kong" \

-e "DB\_PASSWORD=kong123" \

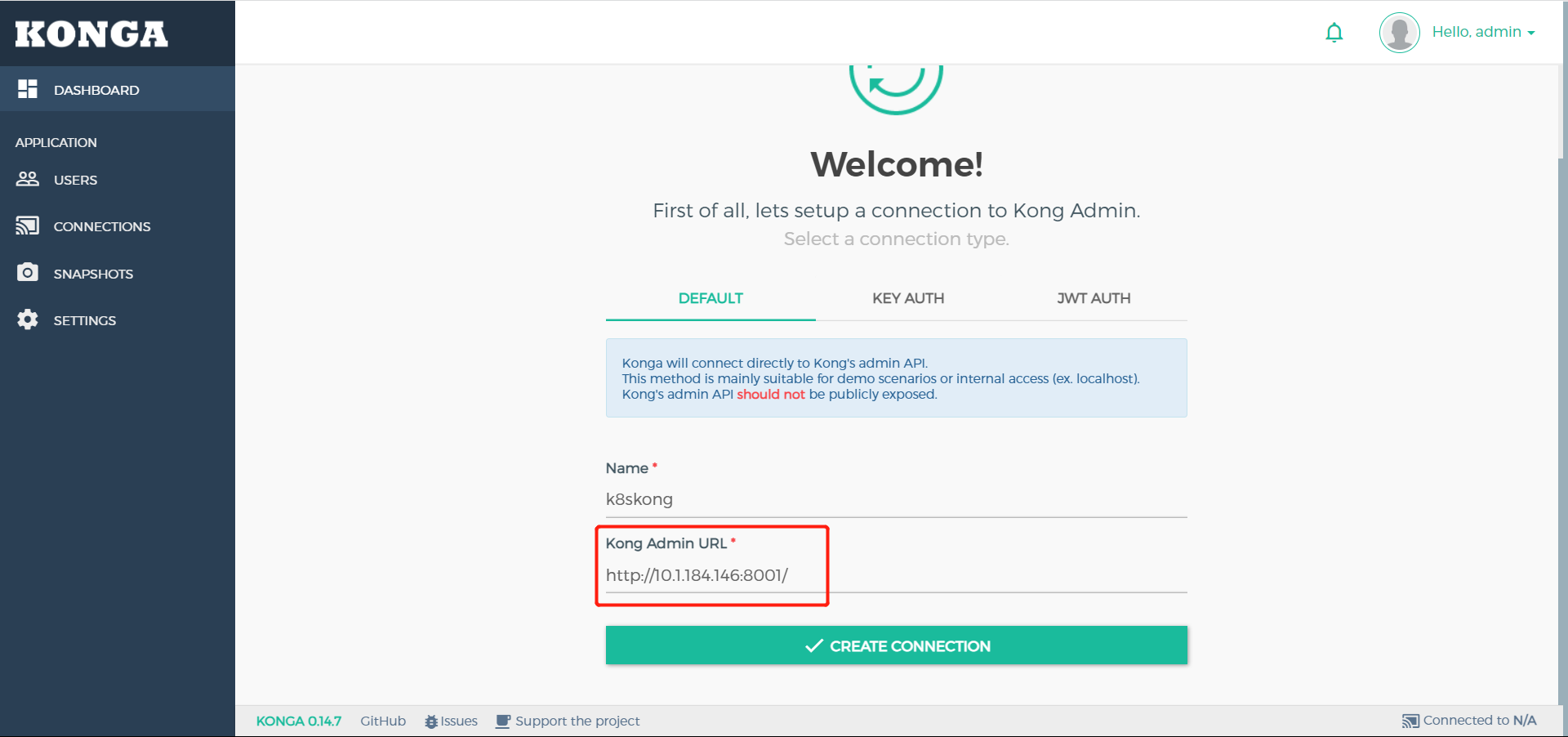
-e "DB\_DATABASE=konga" \

-e "NODE\_ENV=development" \

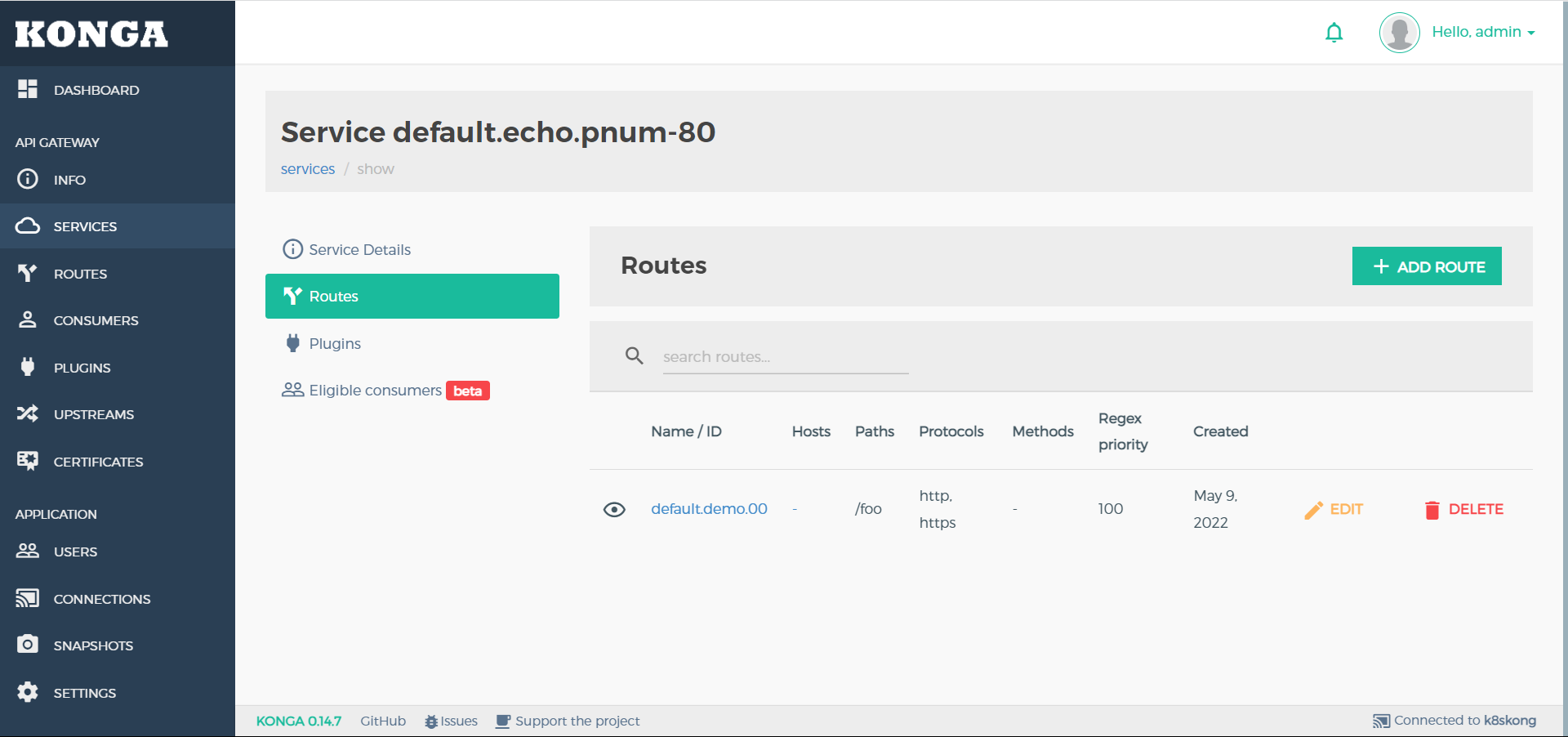
docker.io/pantsel/konga:0.14.7

1. 访问http://192.168.102.130:1337，（192.168.102.130是我k8s master的地址）

登录进去后连接kong，这里的连接地址是我们前面创建的k8s服务的地址10.1.184.146



1. 我就进入Services页面可以看到kong-ingress创建的Service



**Kong-ingress yaml文件模板示例**

apiVersion: apps/v1

kind: Deployment

metadata:

name: mynginx

spec:

replicas: 1

selector:

matchLabels:

app: mynginx

template:

metadata:

labels:

app: mynginx

spec:

containers:

- image: nginx:1.17.1

name: nginx

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: mynginx-svc

spec:

ports:

- port: 80

name: 80port

protocol: TCP

targetPort: 80

selector:

app: mynginx

---

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: mynginx-ingress

spec:

ingressClassName: kong

rules:

- http:

paths:

- path: /mynginx

pathType: ImplementationSpecific

backend:

service:

name: mynginx-svc

port:

number: 80