本篇文件介绍kong配置文件的配置

**配置文件**

Kong启动时，如果存在 /etc/kong/kong.conf 文件，将会使用该文件的配置，该文件由开发者自己生成

/etc/kong/kong.conf.default是kong提供的模板配置，开发者可参考该配置

**基于配置文件的Docker启动**

如下，我们使用配置文件启动Kong Docker

1. 新建一个TestKong目录
2. 目录下新建kong.conf，其内容如下

database = postgres

pg\_host = postgres

pg\_port = 5432

pg\_user = kong

pg\_password = kong123

pg\_database = kong

1. 目录下新建Dockerfile文件，其内容如下

FROM kong:2.0

COPY kong.conf /etc/kong/kong.conf

1. 执行如下命令生产镜像

# docker build -t iceemblem/kongtest:0.1 .

1. 执行如下命令运行测试kong docker

# docker run -d --name kongtest \

--network kongnet \

--restart=always \

-e "KONG\_PROXY\_ACCESS\_LOG=/dev/stdout" \

-e "KONG\_ADMIN\_ACCESS\_LOG=/dev/stdout" \

-e "KONG\_PROXY\_ERROR\_LOG=/dev/stderr" \

-e "KONG\_ADMIN\_ERROR\_LOG=/dev/stderr" \

-e "KONG\_ADMIN\_LISTEN=0.0.0.0:8001" \

-e "KONG\_ADMIN\_LISTEN\_SSL=0.0.0.0:8444" \

-p 7000:8000 \

-p 7443:8443 \

-p 7001:8001 \

-p 7444:8444 \

iceemblem/kongtest:0.1

1. 访问测试kong

如下，我们可以访问127.0.0.1:7000可以查看kong已经运行，如果你跟着直接的文章教程走，那么你访问http://127.0.0.1:7000/netkongtest/WeatherForecast将会看到测试站点的结果

**Kong的配置**

如下，我列出了一下常用的配置，详细配置需要查看kong容器的/etc/kong/kong.conf.default文件

#------------------------------------------------------------------------------

# 常规配置

#------------------------------------------------------------------------------

#prefix = /usr/local/kong/       # 工作目录，包含临时文件和日志

#log\_level = notice              # 日志等级，日志路径为 `<prefix>/logs/error.log`

#proxy\_access\_log = logs/access.log       # 请求访问的日志，设为 off 可以关闭

#proxy\_error\_log = logs/error.log         # 请求错误的日志，路径前缀为 <prefix>

#admin\_access\_log = logs/admin\_access.log # Admin Api 的请求访问日志，设为 off 可以关闭

#admin\_error\_log = logs/error.log         # Admin Api 的请求错误日志，路径前缀为 <prefix>

#plugins = bundled               # kong 启动时加载的插件名称列表，以英文","分隔，如 bundled,custom-auth,custom-log

#------------------------------------------------------------------------------

# NGINX 代理配置

#------------------------------------------------------------------------------

#proxy\_listen = 0.0.0.0:8000 reuseport backlog=16384, 0.0.0.0:8443 http2 ssl reuseport backlog=16384

                         # 代理监听地址，以","分隔

                         #

                         # 一些选项的解释

                         # - `0.0.0.0:8000` 监听所有地址的 8000 端口

                         # - `ssl` 使用 TLS 握手

                         # - `http2` 使用 HTTP/2 协议

                         # - `reuseport` 为每个进程创建单独的套接字

                         # - `backlog=N` 挂起的TCP连接队列的最大长度

#admin\_listen = 127.0.0.1:8001 reuseport backlog=16384, 127.0.0.1:8444 http2 ssl reuseport backlog=16384

                         # Admin API 的配置，与 proxy\_listen 相同

#nginx\_user = nobody nobody      # 定义工作进程使用的用户和用户组，默认为启动时的用户和用户组

#nginx\_worker\_processes = auto   # 定义工作进程数

#------------------------------------------------------------------------------

# SSL 配置

#------------------------------------------------------------------------------

#ssl\_cert =                      # `proxy\_listen` 地址的 SSL 证书绝对路径

#ssl\_cert\_key =                  # `proxy\_listen` 地址的 SSL 密钥绝对路径

#admin\_ssl\_cert =                # `admin\_listen` 地址的 SSL 证书绝对路径

#admin\_ssl\_cert\_key =            # `admin\_listen` 地址的 SSL 密钥绝对路径

#------------------------------------------------------------------------------

# Postgres 数据库配置

#------------------------------------------------------------------------------

#database = postgres             # 是数据库类型，此处为 postgres

#pg\_host = 127.0.0.1             # Postgres 数据库地址

#pg\_port = 5432                  # Postgres 数据库端口

#pg\_timeout = 5000               # Postgres 数据库连接超时时间，毫秒

#pg\_user = kong                  # Postgres 用户名

#pg\_password =                   # Postgres 密码

#pg\_database = kong              # Postgres 数据库名

#------------------------------------------------------------------------------

# 其他配置

#------------------------------------------------------------------------------

#trusted\_ips =                   # 受信任的IP列表，由该列表的IP发送的请求将原样转发 `X-Forwarded-\*`

                                 # 否则 kong 插入自己的 `X-Forwarded-\*`，以","分隔

                                 # 若信任所有地址，则使用 0.0.0.0/0,::/0

#real\_ip\_header = X-Real-IP      # 真实IP头，想要获取客户端的IP地址可利用该字段

**指令注入**

Kong基于nginx，想要配置nginx可使用一定命名规则的配置达到配置nginx

nginx\_http\_<directive>：将<directive>注入到nginx的http区块

nginx\_proxy\_<directive>：将<directive>注入到nginx的server区块

nginx\_http\_upstream\_<directive>：将<directive>注入到nginx的upstream区块

示例kong.conf配置：

# 配置nginx的http区块ssl\_protocols

nginx\_http\_ssl\_protocols = TLSv1.1 TLSv1.2 TLSv1.3