# ingress 配置参数详解

## 自定义NGINX的配置方法有以下三种：

- ConfigMap：使用Configmap在NGINX中设置全局配置。https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/
- 注解：如果您要为特定的Ingress规则进行特定的配置，请使用此注解。https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/
- 自定义模板：当需要更具体的设置（例如open_file_cache）时，重写此模板/etc/nginx/template/nginx.tmpl，然后完成挂载到ingress pod中。https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/custom-template/

## nginx 可修改变量说明

> 如下代码中翻译的相关变量说明后面的 json: 后面的变量即为可使用yaml configmap配置变量

```
/*
Copyright 2016 The Kubernetes Authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

package config

import (
    "strconv"
    "time"

    "k8s.io/klog/v2"

    apiv1 "k8s.io/api/core/v1"

    "k8s.io/ingress-nginx/internal/ingress"
    // 导入默认的一部分backend配置参数
    "k8s.io/ingress-nginx/internal/ingress/defaults"
    "k8s.io/ingress-nginx/internal/runtime"
)
// 默认ssl相关功能关闭
var (
    // EnableSSLChainCompletion Autocomplete SSL certificate chains with missing intermediate CA certificates.
    EnableSSLChainCompletion = false
)

// const区域为一些全局默认参数设定值
const (
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#client_max_body_size
    // Sets the maximum allowed size of the client request body
    // client_max_body_size 默认大小设定
    bodySize = "1m"

    // http://nginx.org/en/docs/ngx_core_module.html#error_log
    // Configures logging level [debug | info | notice | warn | error | crit | alert | emerg]
    // Log levels above are listed in the order of increasing severity
    // log级别默认设定
    errorLevel = "notice"

    // HTTP Strict Transport Security (often abbreviated as HSTS) is a security feature (HTTP header)
    // that tell browsers that it should only be communicated with using HTTPS, instead of using HTTP.
    // https://developer.mozilla.org/en-US/docs/Web/Security/HTTP_strict_transport_security
    // max-age is the time, in seconds, that the browser should remember that this site is only to be accessed using HTTPS.
    // hsts 安全协议浏览器保留时间
    hstsMaxAge = "15724800"

    // gzip 压缩的类型
    gzipTypes = "application/atom+xml application/javascript application/x-javascript application/json application/rss+xml application/vnd.ms-fontobject application/x-font-ttf application/x-web-app-manifest+json application/xhtml+xml application/xml font/opentype image/svg+xml image/x-icon text/css text/javascript text/plain text/x-component"
    // brotli 压缩算法的类型，官方介绍优于gzip
    brotliTypes = "application/xml+rss application/atom+xml application/javascript application/x-javascript application/json application/rss+xml application/vnd.ms-fontobject application/x-font-ttf application/x-web-app-manifest+json application/xhtml+xml application/xml font/opentype image/svg+xml image/x-icon text/css text/javascript text/plain text/x-component"

    // log输出格式 包含upstream相关信息
    logFormatUpstream = `$remote_addr - $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http_user_agent" $request_length $request_time [$proxy_upstream_name] [$proxy_alternative_upstream_name] $upstream_addr $upstream_response_length $upstream_response_time $upstream_status $req_id`

    // 请求相关log输出格式
    logFormatStream = `[$remote_addr] [$time_local] $protocol $status $bytes_sent $bytes_received $session_time`

    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_buffer_size
    // Sets the size of the buffer used for sending data.
    // 4k helps NGINX to improve TLS Time To First Byte (TTTFB)
    // https://www.igvita.com/2013/12/16/optimizing-nginx-tls-time-to-first-byte/
    // ssl 缓存空间设定
    sslBufferSize = "4k"

    // Enabled ciphers list to enabled. The ciphers are specified in the format understood by the OpenSSL library
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_ciphers
    // 允许开启的加密类型
    sslCiphers = "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384"

    // SSL enabled protocols to use
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_protocols
    // 允许开启的加密版本
    sslProtocols = "TLSv1.2 TLSv1.3"

    // Disable TLS 1.3 early data
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_early_data
    // 拒绝 tls1.3版本 有安全漏点
    sslEarlyData = false

    // Time during which a client may reuse the session parameters stored in a cache.
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_timeout
    // session 超时时间
    sslSessionTimeout = "10m"

    // Size of the SSL shared cache between all worker processes.
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_cache
    // ssl 缓存大小设定
    sslSessionCacheSize = "10m"

    // Parameters for a shared memory zone that will keep states for various keys.
    // http://nginx.org/en/docs/http/ngx_http_limit_conn_module.html#limit_conn_zone
    // 默认限流变量值设定
    defaultLimitConnZoneVariable = "$binary_remote_addr"
)
```

```
// Configuration represents the content of nginx.conf file
// 配置文件部分可以设定的参数集合，可全局，也可局部设定
// 配置结构体
type Configuration struct {
    导入默认backend 配置参数结构体，见下面
    defaults.Backend `json:",squash"`

    // Sets the name of the configmap that contains the headers to pass to the client
    // 自定义header头
    AddHeaders string `json:"add-headers,omitempty"`

    // AllowBackendServerHeader enables the return of the header Server from the backend
    // instead of the generic nginx string.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_hide_header
    // By default this is disabled
    // 设置允许的到后端的header头限制
    AllowBackendServerHeader bool `json:"allow-backend-server-header"`

    // AccessLogParams sets additionals params for access_log
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#access_log
    // By default it's empty
    // 是否开启请求参数记录到log
    AccessLogParams string `json:"access-log-params,omitempty"`

    // EnableAccessLogForDefaultBackend enable access_log for default backend
    // By default this is disabled
    // 是否开启请求到后端的log记录
    EnableAccessLogForDefaultBackend bool `json:"enable-access-log-for-default-backend"`

    // AccessLogPath sets the path of the access logs for both http and stream contexts if enabled
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#access_log
    // http://nginx.org/en/docs/stream/ngx_stream_log_module.html#access_log
    // By default access logs go to /var/log/nginx/access.log
    // http stream log 路径设定，默认空
    AccessLogPath string `json:"access-log-path,omitempty"`

    // HttpAccessLogPath sets the path of the access logs for http context globally if enabled
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#access_log
    // 记录 http access log设定路径
    HttpAccessLogPath string `json:"http-access-log-path,omitempty"`

    // StreamAccessLogPath sets the path of the access logs for stream context globally if enabled
    // http://nginx.org/en/docs/stream/ngx_stream_log_module.html#access_log
    // stream log 设定路径
    StreamAccessLogPath string `json:"stream-access-log-path,omitempty"`

    // WorkerCPUAffinity bind nginx worker processes to CPUs this will improve response latency
    // http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_affinity
    // By default this is disabled
    WorkerCPUAffinity string `json:"worker-cpu-affinity,omitempty"`
    // ErrorLogPath sets the path of the error logs
    // http://nginx.org/en/docs/ngx_core_module.html#error_log
    // By default error logs go to /var/log/nginx/error.log
    // 错误log路径
    ErrorLogPath string `json:"error-log-path,omitempty"`

    // EnableModsecurity enables the modsecurity module for NGINX
    // By default this is disabled
    // 是否开启waf相关功能
    EnableModsecurity bool `json:"enable-modsecurity"`

    // EnableOCSP enables the OCSP support in SSL connections
    // By default this is disabled
    // 是否开启 ocsp安全连接相关功能
    EnableOCSP bool `json:"enable-ocsp"`

    // EnableOWASPCoreRules enables the OWASP ModSecurity Core Rule Set (CRS)
    // By default this is disabled
    // 是否开启 waf相关的一些cores 规则启用
    EnableOWASPCoreRules bool `json:"enable-owasp-modsecurity-crs"`

    // ModSecuritySnippet adds custom rules to modsecurity section of nginx configuration
    // waf相关规则片段配置
    ModsecuritySnippet string `json:"modsecurity-snippet"`

    // ClientHeaderBufferSize allows to configure a custom buffer
    // size for reading client request header
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#client_header_buffer_size
    // 客户端 请求头缓存大小设定
    ClientHeaderBufferSize string `json:"client-header-buffer-size"`

    // Defines a timeout for reading client request header, in seconds
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#client_header_timeout
    // 客户端请求头超时设定
    ClientHeaderTimeout int `json:"client-header-timeout,omitempty"`

    // Sets buffer size for reading client request body
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#client_body_buffer_size
    // 客户端缓存大小设定
    ClientBodyBufferSize string `json:"client-body-buffer-size,omitempty"`

    // Defines a timeout for reading client request body, in seconds
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#client_body_timeout
    // 客户端主体超时设定
    ClientBodyTimeout int `json:"client-body-timeout,omitempty"`

    // DisableAccessLog disables the Access Log globally for both HTTP and Stream contexts from NGINX ingress controller
    // http://nginx.org/en/docs/http/ngx_http_log_module.html
    // http://nginx.org/en/docs/stream/ngx_stream_log_module.html
    // 禁止access log输出
    DisableAccessLog bool `json:"disable-access-log,omitempty"`

    // DisableHTTPAccessLog disables the Access Log for http context globally from NGINX ingress controller
    // http://nginx.org/en/docs/http/ngx_http_log_module.html
    // 禁止 http access log 输出
    DisableHTTPAccessLog bool `json:"disable-http-access-log,omitempty"`

    // DisableStreamAccessLog disables the Access Log for stream context globally from NGINX ingress controller
    // http://nginx.org/en/docs/stream/ngx_stream_log_module.html
    // 禁止 stream access log 输出
    DisableStreamAccessLog bool `json:"disable-stream-access-log,omitempty"`

    // DisableIpv6DNS disables IPv6 for nginx resolver
    // 禁止 ipv6 dns功能
    DisableIpv6DNS bool `json:"disable-ipv6-dns"`

    // DisableIpv6 disable listening on ipv6 address
    // 禁止监听 ipv6 可为空
    DisableIpv6 bool `json:"disable-ipv6,omitempty"`

    // EnableUnderscoresInHeaders enables underscores in header names
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#underscores_in_headers
    // By default this is disabled
    /*
    HTTP头是可以包含英文字母([A-Za-z])、数字([0-9])、连接号(-)hyphens, 也可义是下划线(_)。在使用nginx的时候应该避免使用包含下划线的HTTP头。主要的原因有以下2点。
    1.默认的情况下nginx引用header变量时不能使用带下划线的变量。要解决这样的问题只能单独配置underscores_in_headers on。
    2.默认的情况下会忽略掉带下划线的变量。要解决这个需要配置ignore_invalid_headers off。
    */
    EnableUnderscoresInHeaders bool `json:"enable-underscores-in-headers"`

    // IgnoreInvalidHeaders set if header fields with invalid names should be ignored
    //
```

```
http://nginx.org/en/docs/http/ngx_http_core_module.html#ignore_invalid_headers
    // By default this is enabled
    IgnoreInvalidHeaders bool `json:"ignore-invalid-headers"`

    // RetryNonIdempotent since 1.9.13 NGINX will not retry non-idempotent requests (POST, LOCK, PATCH)
    // in case of an error. The previous behavior can be restored using the value true
    // 默认不重试 post lock patch等http请求方法，除非程序做到完全解耦. 默认false，如果设为true 比如转账会出现两笔重大故障
    RetryNonIdempotent bool `json:"retry-non-idempotent"`

    // http://nginx.org/en/docs/ngx_core_module.html#error_log
    // Configures logging level [debug | info | notice | warn | error | crit | alert | emerg]
    // Log levels above are listed in the order of increasing severity
    // 错误log级别 可为空
    ErrorLogLevel string `json:"error-log-level,omitempty"`

    // https://nginx.org/en/docs/http/ngx_http_v2_module.html#http2_max_field_size
    // HTTP2MaxFieldSize Limits the maximum size of an HPACK-compressed request header field
    // 设置一个连接的最大并发 HTTP/2 流数量
    HTTP2MaxFieldSize string `json:"http2-max-field-size,omitempty"`

    // https://nginx.org/en/docs/http/ngx_http_v2_module.html#http2_max_header_size
    // HTTP2MaxHeaderSize Limits the maximum size of the entire request header list after HPACK decompression
    // 限制 HPACK 压缩的请求头字段的最大大小（size）
    HTTP2MaxHeaderSize string `json:"http2-max-header-size,omitempty"`

    // http://nginx.org/en/docs/http/ngx_http_v2_module.html#http2_max_requests
    // HTTP2MaxRequests Sets the maximum number of requests (including push requests) that can be served
    // through one HTTP/2 connection, after which the next client request will lead to connection closing
    // and the need of establishing a new connection.
    // HTTP2最大连接
    HTTP2MaxRequests int `json:"http2-max-requests,omitempty"`

    // http://nginx.org/en/docs/http/ngx_http_v2_module.html#http2_max_concurrent_streams
    // Sets the maximum number of concurrent HTTP/2 streams in a connection.
    // 限制一个连接的最大并发推送请求数。
    HTTP2MaxConcurrentStreams int `json:"http2-max-concurrent-streams,omitempty"`
    // Enables or disables the header HSTS in servers running SSL
    // hsts 安全开启
    HSTS bool `json:"hsts,omitempty"`

    // Enables or disables the use of HSTS in all the subdomains of the servername
    // Default: true
    // 允许所有 子域名也开启 hsts
    HSTSIncludeSubdomains bool `json:"hsts-include-subdomains,omitempty"`

    // HTTP Strict Transport Security (often abbreviated as HSTS) is a security feature (HTTP header)
    // that tell browsers that it should only be communicated with using HTTPS, instead of using HTTP.
    // https://developer.mozilla.org/en-US/docs/Web/Security/HTTP_strict_transport_security
    // max-age is the time, in seconds, that the browser should remember that this site is only to be
    // accessed using HTTPS.
    // hsts 最大缓存时间
    HSTSMaxAge string `json:"hsts-max-age,omitempty"`

    // Enables or disables the preload attribute in HSTS feature
    // hsts 预载入功能开启
    HSTSPreload bool `json:"hsts-preload,omitempty"`

    // Time during which a keep-alive client connection will stay open on the server side.
    // The zero value disables keep-alive client connections
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#keepalive_timeout
    // 长连接保持连接超时
    KeepAlive int `json:"keep-alive,omitempty"`

    // Sets the maximum number of requests that can be served through one keep-alive connection.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#keepalive_requests
    // 一个长连接可以处理多少个请求
    KeepAliveRequests int `json:"keep-alive-requests,omitempty"`

    // LargeClientHeaderBuffers Sets the maximum number and size of buffers used for reading
    // large client request header.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#large_client_header_buffers
    // Default: 4 8k
    // 最大客户端头部缓冲设定
    LargeClientHeaderBuffers string `json:"large-client-header-buffers"`

    // Enable json escaping
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#log_format
    // 开启json log格式
    LogFormatEscapeJSON bool `json:"log-format-escape-json,omitempty"`

    // Customize upstream log_format
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#log_format
    // 最定义log 格式 for 7层代理 http
    LogFormatUpstream string `json:"log-format-upstream,omitempty"`

    // Customize stream log_format
    // http://nginx.org/en/docs/http/ngx_http_log_module.html#log_format
    // 自定义log格式 for 四层代理 比如mysql
    LogFormatStream string `json:"log-format-stream,omitempty"`

    // If disabled, a worker process will accept one new connection at a time.
    // Otherwise, a worker process will accept all new connections at a time.
    // http://nginx.org/en/docs/ngx_core_module.html#multi_accept
    // Default: true
    // 一个work 接受同时多个请求连接
    EnableMultiAccept bool `json:"enable-multi-accept,omitempty"`

    // Maximum number of simultaneous connections that can be opened by each worker process
    // http://nginx.org/en/docs/ngx_core_module.html#worker_connections
    // 最大worker数设定
    MaxWorkerConnections int `json:"max-worker-connections,omitempty"`

    // Maximum number of files that can be opened by each worker process.
    // http://nginx.org/en/docs/ngx_core_module.html#worker_rlimit_nofile
    // worker最大文件打开数设定
    MaxWorkerOpenFiles int `json:"max-worker-open-files,omitempty"`

    // Sets the bucket size for the map variables hash tables.
    // Default value depends on the processor’s cache line size.
    // http://nginx.org/en/docs/http/ngx_http_map_module.html#map_hash_bucket_size
    // nginx 维护的map字典空间，正常情况不需要额外设定，如果超出了可以适当扩展下
    MapHashBucketSize int `json:"map-hash-bucket-size,omitempty"`

    // NginxStatusIpv4Whitelist has the list of cidr that are allowed to access
    // the /nginx_status endpoint of the "_" server
    // nginx 基于ipv4 ipv6 的白名单列表维护默认localhost，比如你访问 默认的 nginx_status 就会被限制
    NginxStatusIpv4Whitelist []string `json:"nginx-status-ipv4-whitelist,omitempty"`
    NginxStatusIpv6Whitelist []string `json:"nginx-status-ipv6-whitelist,omitempty"`

    // Plugins configures plugins to use placed in the directory /etc/nginx/lua/plugins.
    // Every plugin has to have main.lua in the root. Every plugin has to bundle all of its dependencies.
    // The execution order follows the definition.
    // lua插件加载列表
    Plugins []string `json:"plugins,omitempty"`

    // If UseProxyProtocol is enabled ProxyRealIPCIDR defines the default the IP/network address
    // of your external load balancer
    // 真实IP设定网络cidr
    ProxyRealIPCIDR []string `json:"proxy-real-ip-cidr,omitempty"`

    // Sets the name of the configmap that contains the headers to pass to the backend
    // 代理请求头设定
    ProxySetHeaders string `json:"proxy-set-headers,omitempty"`

    // Maximum size of the server names hash tables used in server names, map directive’s values,
    // MIME types, names of request header strings, etcd.
    // http://nginx.org/en/docs/hash.html
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#server_names_hash_max_size
    // servername 最大空间设定
    ServerNameHashMaxSize int `json:"server-name-hash-max-size,omitempty"`

    // Size of the bucket for the server names hash tables
    // http://nginx.org/en/docs/hash.html
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#server_names_hash_bucket_size
    // servername 空间大小设定，上限为上面的那个值
    ServerNameHashBucketSize int `json:"server-name-hash-bucket-size,omitempty"`

    // Size of the bucket for the proxy headers hash tables
    // http://nginx.org/en/docs/hash.html
    // https://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_headers_hash_max_size
    // 最大代理请求头大小设定
    ProxyHeadersHashMaxSize int `json:"proxy-headers-hash-max-size,omitempty"`

    // Maximum size of the bucket for the proxy headers hash tables
    // http://nginx.org/en/docs/hash.html
    // https://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_headers_hash_bucket_size
    ProxyHeadersHashBucketSize int `json:"proxy-headers-hash-bucket-size,omitempty"`

    // Enables or disables emitting nginx version in error messages and in the “Server” response header field.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#server_tokens
    // Default: true
    // nginx 安全设定，是否暴露nginx版本
    ShowServerTokens bool `json:"server-tokens"`

    // Enabled ciphers list to enabled. The ciphers are specified in the
format understood by
    // the OpenSSL library
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_ciphers
    // ssl 加密算法设定
    SSLCiphers string `json:"ssl-ciphers,omitempty"`

    // Specifies a curve for ECDHE ciphers.
    //
http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_ecdh_curve
    // ssl ECDHC加密算法 密码
    SSLECDHCurve string `json:"ssl-ecdh-curve,omitempty"`

    // The secret that contains Diffie-Hellman key to help with "Perfect
Forward Secrecy"
    // https://wiki.openssl.org/index.php/Diffie-Hellman_parameters
    //
https://wiki.mozilla.org/Security/Server_Side_TLS#DHE_handshake_and_dhparam
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_dhparam
    // 指定 ssl dhp加密秘钥
    SSLDHParam string `json:"ssl-dh-param,omitempty"`

    // SSL enabled protocols to use
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_protocols
    // 指定加密协议版本
    SSLProtocols string `json:"ssl-protocols,omitempty"`

    // Enables or disable TLS 1.3 early data.
    //
http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_early_data
    // 允许或者禁止 TLS 1.3 不安全
    SSLEarlyData bool `json:"ssl-early-data,omitempty"`

    // Enables or disables the use of shared SSL cache among worker
processes.
    //
http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_cache
    // 开启加密session缓存共享空间
    SSLSessionCache bool `json:"ssl-session-cache,omitempty"`

    // Size of the SSL shared cache between all worker processes.
    //
http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_cache
    // 开启加密session缓存共享空间大小
    SSLSessionCacheSize string `json:"ssl-session-cache-size,omitempty"`

    // Enables or disables session resumption through TLS session tickets.
    //
http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_tickets
    // ssl session 粘贴开启 会话保持
    SSLSessionTickets bool `json:"ssl-session-tickets,omitempty"`

    // Sets the secret key used to encrypt and decrypt TLS session tickets.
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_tickets
    // By default, a randomly generated key is used.
    // Example: openssl rand 80 | openssl enc -A -base64
    // 指定加密key
    SSLSessionTicketKey string `json:"ssl-session-ticket-key,omitempty"`

    // Time during which a client may reuse the session parameters stored in a cache.
    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_session_timeout
    // 加密连接超时
    SSLSessionTimeout string `json:"ssl-session-timeout,omitempty"`

    // http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_buffer_size
    // Sets the size of the buffer used for sending data.
    // 4k helps NGINX to improve TLS Time To First Byte (TTTFB)
    // https://www.igvita.com/2013/12/16/optimizing-nginx-tls-time-to-first-byte/
    // ssl 缓冲大小设定
    SSLBufferSize string `json:"ssl-buffer-size,omitempty"`

    // Enables or disables the use of the PROXY protocol to receive client connection
    // (real IP address) information passed through proxy servers and load balancers
    // such as HAproxy and Amazon Elastic Load Balancer (ELB).
    // https://www.nginx.com/resources/admin-guide/proxy-protocol/
    // 使用代理协议版本
    UseProxyProtocol bool `json:"use-proxy-protocol,omitempty"`

    // When use-proxy-protocol is enabled, sets the maximum time the connection handler will wait
    // to receive proxy headers.
    // Example '60s'
    // 代理协议头超时
    ProxyProtocolHeaderTimeout time.Duration `json:"proxy-protocol-header-timeout,omitempty"`

    // Enables or disables the use of the nginx module that compresses responses using the "gzip" method
    // http://nginx.org/en/docs/http/ngx_http_gzip_module.html
    // 是否开启gzip压缩
    UseGzip bool `json:"use-gzip,omitempty"`

    // Enables or disables the use of the nginx geoip module that creates variables with values depending on the client IP
    // http://nginx.org/en/docs/http/ngx_http_geoip_module.html
    // 是否启用geoip解析客户区域
    UseGeoIP bool `json:"use-geoip,omitempty"`
    // UseGeoIP2 enables the geoip2 module for NGINX
    // By default this is disabled
    // 启用geoip2 解析客户区域
    UseGeoIP2 bool `json:"use-geoip2,omitempty"`

    // Enables or disables the use of the NGINX Brotli Module for
compression
    // https://github.com/google/ngx_brotli
    // 启用brotli 加密算法
    EnableBrotli bool `json:"enable-brotli,omitempty"`

    // Brotli Compression Level that will be used
    // brotli 加密级别
    BrotliLevel int `json:"brotli-level,omitempty"`

    // MIME Types that will be compressed on-the-fly using Brotli module
    // brotli 加密文件类型
    BrotliTypes string `json:"brotli-types,omitempty"`

    // Enables or disables the HTTP/2 support in secure connections
    // http://nginx.org/en/docs/http/ngx_http_v2_module.html
    // Default: true
    // 是否启用http2
    UseHTTP2 bool `json:"use-http2,omitempty"`

    // gzip Compression Level that will be used
    // gzip 启用级别
    GzipLevel int `json:"gzip-level,omitempty"`

    // Minimum length of responses to be sent to the client before it is
eligible
    // for gzip compression, in bytes.
    // gzip 最小长度
    GzipMinLength int `json:"gzip-min-length,omitempty"`

    // MIME types in addition to "text/html" to compress. The special value
“*” matches any MIME type.
    // Responses with the “text/html” type are always compressed if UseGzip
is enabled
    // gzip 加密文件类型
    GzipTypes string `json:"gzip-types,omitempty"`

    // Defines the number of worker processes. By default auto means number
of available CPU cores
    // http://nginx.org/en/docs/ngx_core_module.html#worker_processes
    // 工作进程设定，默认是cpu核心数
    WorkerProcesses string `json:"worker-processes,omitempty"`

    // Defines a timeout for a graceful shutdown of worker processes
    //
http://nginx.org/en/docs/ngx_core_module.html#worker_shutdown_timeout
    // 优雅关闭工作进程等待时间
    WorkerShutdownTimeout string `json:"worker-shutdown-timeout,omitempty"`
    // Sets the bucket size for the variables hash table.
    // http://nginx.org/en/docs/http/ngx_http_map_module.html#variables_hash_bucket_size
    // 变量空间设定大小
    VariablesHashBucketSize int `json:"variables-hash-bucket-size,omitempty"`

    // Sets the maximum size of the variables hash table.
    // http://nginx.org/en/docs/http/ngx_http_map_module.html#variables_hash_max_size
    // 变量空间最大设定
    VariablesHashMaxSize int `json:"variables-hash-max-size,omitempty"`

    // Activates the cache for connections to upstream servers.
    // The connections parameter sets the maximum number of idle keepalive connections to
    // upstream servers that are preserved in the cache of each worker process. When this
    // number is exceeded, the least recently used connections are closed.
    // http://nginx.org/en/docs/http/ngx_http_upstream_module.html#keepalive
    // upstream最大限制连接数维护设定
    UpstreamKeepaliveConnections int `json:"upstream-keepalive-connections,omitempty"`

    // Sets a timeout during which an idle keepalive connection to an upstream server will stay open.
    // http://nginx.org/en/docs/http/ngx_http_upstream_module.html#keepalive_timeout
    // upstream连接超时设定
    UpstreamKeepaliveTimeout int `json:"upstream-keepalive-timeout,omitempty"`

    // Sets the maximum number of requests that can be served through one keepalive connection.
    // After the maximum number of requests is made, the connection is closed.
    // http://nginx.org/en/docs/http/ngx_http_upstream_module.html#keepalive_requests
    // 一个保持连接最大处理线程数量
    UpstreamKeepaliveRequests int `json:"upstream-keepalive-requests,omitempty"`

    // Sets the maximum size of the variables hash table.
    // http://nginx.org/en/docs/http/ngx_http_map_module.html#variables_hash_max_size
    // 限制连接空间变量
    LimitConnZoneVariable string `json:"limit-conn-zone-
```

```
variable,omitempty"`

    // Sets the timeout between two successive read or write operations on client or proxied server connections.
    // If no data is transmitted within this time, the connection is closed.
    // http://nginx.org/en/docs/stream/ngx_stream_proxy_module.html#proxy_timeout
    // 代理stream超时 四层的
    ProxyStreamTimeout string `json:"proxy-stream-timeout,omitempty"`

    // Sets the number of datagrams expected from the proxied server in response
    // to the client request if the UDP protocol is used.
    // http://nginx.org/en/docs/stream/ngx_stream_proxy_module.html#proxy_responses
    // Default: 1
    // 设置四层相应报文数量，默认是1
    ProxyStreamResponses int `json:"proxy-stream-responses,omitempty"`

    // Modifies the HTTP version the proxy uses to interact with the backend.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_http_version
    // 设定代理http版本 1.1 或者1.0 1.2这些
    ProxyHTTPVersion string `json:"proxy-http-version"`

    // Sets the ipv4 addresses on which the server will accept requests.
    // 绑定监听ip地址
    BindAddressIpv4 []string `json:"bind-address-ipv4,omitempty"`

    // Sets the ipv6 addresses on which the server will accept requests.
    // 绑定监听IPV6地址
    BindAddressIpv6 []string `json:"bind-address-ipv6,omitempty"`

    // Sets whether to use incoming X-Forwarded headers.
    是否开启xff转发真实IP功能
    UseForwardedHeaders bool `json:"use-forwarded-headers"`

    // Sets whether to enable the real ip module
    // 启动获取真实IP
    EnableRealIp bool `json:"enable-real-ip"`

    // Sets the header field for identifying the originating IP address of a client
    // Default is X-Forwarded-For
    // 设置代理xff 头覆盖，默认 X-Forwarded-For
    ForwardedForHeader string `json:"forwarded-for-header,omitempty"`

    // Append the remote address to the X-Forwarded-For header instead of replacing it
    // Default: false
    // 追加IP 传递方式开启
    ComputeFullForwardedFor bool `json:"compute-full-forwarded-for,omitempty"`

    // If the request does not have a request-id, should we generate a random value?
    // Default: true
    // 生成请求ID 默认开启，可在log中打印。
    GenerateRequestID bool `json:"generate-request-id,omitempty"`

    // Adds an X-Original-Uri header with the original request URI to the backend request
    // Default: true
    // 开启原始uri X-Original-Uri 传递，默认开启
    ProxyAddOriginalURIHeader bool `json:"proxy-add-original-uri-header"`

    // EnableOpentracing enables the nginx Opentracing extension
    // https://github.com/opentracing-contrib/nginx-opentracing
    // By default this is disabled
    // 允许 opentracing 追踪
    EnableOpentracing bool `json:"enable-opentracing"`

    // OpentracingOperationName specifies a custom name for the server span
    // 指定opentracing 相关名字
    OpentracingOperationName string `json:"opentracing-operation-name"`

    // OpentracingOperationName specifies a custom name for the location span
    OpentracingLocationOperationName string `json:"opentracing-location-operation-name"`

    // zipking 相关设定
    // ZipkinCollectorHost specifies the host to use when uploading traces
    ZipkinCollectorHost string `json:"zipkin-collector-host"`

    // ZipkinCollectorPort specifies the port to use when uploading traces
    // Default: 9411
    ZipkinCollectorPort int `json:"zipkin-collector-port"`

    // ZipkinServiceName specifies the service name to use for any traces created
    // Default: nginx
    ZipkinServiceName string `json:"zipkin-service-name"`

    // ZipkinSampleRate specifies sampling rate for traces
    // Default: 1.0
    ZipkinSampleRate float32 `json:"zipkin-sample-rate"`

    // jeager 相关设定
    // JaegerCollectorHost specifies the host to use when uploading traces
    JaegerCollectorHost string `json:"jaeger-collector-host"`

    // JaegerCollectorPort specifies the port to use when uploading traces
    // Default: 6831
    JaegerCollectorPort int `json:"jaeger-collector-port"`
    // JaegerServiceName specifies the service name to use for any traces created
    // Default: nginx
    JaegerServiceName string `json:"jaeger-service-name"`

    // JaegerSamplerType specifies the sampler to be used when sampling traces.
    // The available samplers are: const, probabilistic, ratelimiting, remote
    // Default: const
    JaegerSamplerType string `json:"jaeger-sampler-type"`

    // JaegerSamplerParam specifies the argument to be passed to the sampler constructor
    // Default: 1
    JaegerSamplerParam string `json:"jaeger-sampler-param"`

    // JaegerSamplerHost specifies the host used for remote sampling consultation
    // Default: http://127.0.0.1
    JaegerSamplerHost string `json:"jaeger-sampler-host"`

    // JaegerSamplerHost specifies the host used for remote sampling consultation
    // Default: 5778
    JaegerSamplerPort int `json:"jaeger-sampler-port"`

    // JaegerTraceContextHeaderName specifies the header name used for passing trace context
    // Default: uber-trace-id
    JaegerTraceContextHeaderName string `json:"jaeger-trace-context-header-name"`

    // JaegerDebugHeader specifies the header name used for force sampling
    // Default: jaeger-debug-id
    JaegerDebugHeader string `json:"jaeger-debug-header"`

    // JaegerBaggageHeader specifies the header name used to submit baggage if there is no root span
    // Default: jaeger-baggage
    JaegerBaggageHeader string `json:"jaeger-baggage-header"`

    // TraceBaggageHeaderPrefix specifies the header prefix used to propagate baggage
    // Default: uberctx-
    JaegerTraceBaggageHeaderPrefix string `json:"jaeger-tracer-baggage-header-prefix"`

    // datadog 相关设定
    // DatadogCollectorHost specifies the datadog agent host to use when uploading traces
    DatadogCollectorHost string `json:"datadog-collector-host"`
    // DatadogCollectorPort specifies the port to use when uploading traces
    // Default: 8126
    DatadogCollectorPort int `json:"datadog-collector-port"`

    // DatadogEnvironment specifies the environment this trace belongs to.
    // Default: prod
    DatadogEnvironment string `json:"datadog-environment"`

    // DatadogServiceName specifies the service name to use for any traces created
    // Default: nginx
    DatadogServiceName string `json:"datadog-service-name"`

    // DatadogOperationNameOverride overrides the operation naem to use for any traces crated
    // Default: nginx.handle
    DatadogOperationNameOverride string `json:"datadog-operation-name-override"`

    // DatadogPrioritySampling specifies to use client-side sampling
    // If true disables client-side sampling (thus ignoring sample_rate) and enables distributed
    // priority sampling, where traces are sampled based on a combination of user-assigned
    // Default: true
    DatadogPrioritySampling bool `json:"datadog-priority-sampling"`

    // DatadogSampleRate specifies sample rate for any traces created.
    // This is effective only when datadog-priority-sampling is false
    // Default: 1.0
    DatadogSampleRate float32 `json:"datadog-sample-rate"`

    // main 区域配置片段设定
    // MainSnippet adds custom configuration to the main section of the nginx configuration
    MainSnippet string `json:"main-snippet"`

    // http 区域配置片段设定
    // HTTPSnippet adds custom configuration to the http section of the nginx configuration
    HTTPSnippet string `json:"http-snippet"`

    // http 区域配置片段设定
    // ServerSnippet adds custom configuration to all the servers in the nginx configuration
    ServerSnippet string `json:"server-snippet"`

    // location 区域相关配置片段设定
    // LocationSnippet adds custom configuration to all the locations in the nginx configuration
    LocationSnippet string `json:"location-snippet"`

    // HTTPRedirectCode sets the HTTP status code to be used in redirects.
    // Supported codes are 301,302,307 and 308
    // Default: 308
    // http 重定向 代码状态指定， 默认308
    HTTPRedirectCode int `json:"http-redirect-code"`

    // ReusePort instructs NGINX to create an individual listening socket
for
    // each worker process (using the SO_REUSEPORT socket option), allowing
a
    // kernel to distribute incoming connections between worker processes
    // Default: true
    // 提高相关socket连接处理性能 默认开启
    ReusePort bool `json:"reuse-port"`

    // HideHeaders sets additional header that will not be passed from the
upstream
    // server to the client response
    // Default: empty
    // 隐藏header头
    HideHeaders []string `json:"hide-headers"`

    // LimitReqStatusCode Sets the status code to return in response to
rejected requests.
    //
http://nginx.org/en/docs/http/ngx_http_limit_req_module.html#limit_req_stat
us
    // Default: 503
    // 限制请求状态码
    LimitReqStatusCode int `json:"limit-req-status-code"`

    // LimitConnStatusCode Sets the status code to return in response to
rejected connections.
    //
http://nginx.org/en/docs/http/ngx_http_limit_conn_module.html#limit_conn_st
atus
    // Default: 503
    // 限制连接状态码
    LimitConnStatusCode int `json:"limit-conn-status-code"`

    // EnableSyslog enables the configuration for remote logging in NGINX
    EnableSyslog bool `json:"enable-syslog"`
    // SyslogHost FQDN or IP address where the logs should be sent
    SyslogHost string `json:"syslog-host"`
    // SyslogPort port
    // 支持log远程syslog传输
    SyslogPort int `json:"syslog-port"`

    // NoTLSRedirectLocations is a comma-separated list of locations
    // that should not get redirected to TLS
    // locations区域没有使用ssl连接的强制重定向配置
    NoTLSRedirectLocations string `json:"no-tls-redirect-locations"`

    // NoAuthLocations is a comma-separated list of locations that
    // should not get authenticated
    // 不需要认知的location配置区域
    NoAuthLocations string `json:"no-auth-locations"`

    // GlobalExternalAuth indicates the access to all locations requires
    // authentication using an external provider
    // +optional
    // 全局扩展认知配置
    GlobalExternalAuth GlobalExternalAuth `json:"global-external-auth"`

    // Checksum contains a checksum of the configmap configuration
    // 检测configmap hash值的
    Checksum string `json:"-"`

    // Block all requests from given IPs
    // 黑名单指定
    BlockCIDRs []string `json:"block-cidrs"`

    // Block all requests with given User-Agent headers
    // 请求ua封杀 类似爬虫
    BlockUserAgents []string `json:"block-user-agents"`

    // Block all requests with given Referer headers
    // 封杀来源referer
    BlockReferers []string `json:"block-referers"`

    // Lua shared dict configuration data / certificate data
    // lua共享自动设定
    LuaSharedDicts map[string]int `json:"lua-shared-dicts"`

    // DefaultSSLCertificate holds the default SSL certificate to use in the configuration
    // It can be the fake certificate or the one behind the flag --default-ssl-certificate
    // 默认ssl凭证设定
    DefaultSSLCertificate *ingress.SSLCert `json:"-"`

    // ProxySSLLocationOnly controls whether the proxy-ssl parameters defined in the
    // proxy-ssl-* annotations are applied on on location level only in the nginx.conf file
    // Default is that those are applied on server level, too
    // 代理ssl location 开启
    ProxySSLLocationOnly bool `json:"proxy-ssl-location-only"`

    // DefaultType Sets the default MIME type of a response.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#default_type
    // Default: text/html
    // 返回内容类型设定 默认 text/html 文本
    DefaultType string `json:"default-type"`
}

// 初始化生成一个上面的配置参数结构体，里面的内容可以参照上面的注释进行解读
// NewDefault returns the default nginx configuration
func NewDefault() Configuration {
    defIPCIDR := make([]string, 0)
    defBindAddress := make([]string, 0)
    defBlockEntity := make([]string, 0)
    defNginxStatusIpv4Whitelist := make([]string, 0)
    defNginxStatusIpv6Whitelist := make([]string, 0)
    defResponseHeaders := make([]string, 0)

    defIPCIDR = append(defIPCIDR, "0.0.0.0/0")
    defNginxStatusIpv4Whitelist = append(defNginxStatusIpv4Whitelist,
"127.0.0.1")
    defNginxStatusIpv6Whitelist = append(defNginxStatusIpv6Whitelist,
"::1")
    defProxyDeadlineDuration := time.Duration(5) * time.Second
    defGlobalExternalAuth := GlobalExternalAuth{"", "", "", "",
append(defResponseHeaders, ""), "", "", "", []string{},
map[string]string{}}

    cfg := Configuration{
        AllowBackendServerHeader:         false,
        AccessLogPath:                    "/var/log/nginx/access.log",
        AccessLogParams:                  "",
        EnableAccessLogForDefaultBackend: false,
        WorkerCPUAffinity:                "",
        ErrorLogPath:                     "/var/log/nginx/error.log",
        BlockCIDRs:                       defBlockEntity,
        BlockUserAgents:                  defBlockEntity,
        BlockReferers:                    defBlockEntity,
        BrotliLevel:                      4,
        BrotliTypes:                      brotliTypes,
        ClientHeaderBufferSize:           "1k",
        ClientHeaderTimeout:              60,
        ClientBodyBufferSize:             "8k",
        ClientBodyTimeout:                60,
        EnableUnderscoresInHeaders:       false,
        ErrorLogLevel:                    errorLevel,
        UseForwardedHeaders:              false,
        EnableRealIp:                     false,
        ForwardedForHeader:               "X-Forwarded-For",
        ComputeFullForwardedFor:          false,
        ProxyAddOriginalURIHeader:        false,
        GenerateRequestID:                true,
        HTTP2MaxFieldSize:                "4k",
        HTTP2MaxHeaderSize:               "16k",
        HTTP2MaxRequests:                 1000,
        HTTP2MaxConcurrentStreams:        128,
        HTTPRedirectCode:                 308,
        HSTS:                             true,
        HSTSIncludeSubdomains:            true,
        HSTSMaxAge:                       hstsMaxAge,
        HSTSPreload:                      false,
        IgnoreInvalidHeaders:             true,
        GzipLevel:                        1,
        GzipMinLength:                    256,
        GzipTypes:                        gzipTypes,
        KeepAlive:                        75,
        KeepAliveRequests:                100,
        LargeClientHeaderBuffers:         "4 8k",
        LogFormatEscapeJSON:              false,
        LogFormatStream:                  logFormatStream,
        LogFormatUpstream:                logFormatUpstream,
        EnableMultiAccept:                true,
        MaxWorkerConnections:             16384,
        MaxWorkerOpenFiles:               0,
        MapHashBucketSize:                64,
        NginxStatusIpv4Whitelist:         defNginxStatusIpv4Whitelist,
        NginxStatusIpv6Whitelist:         defNginxStatusIpv6Whitelist,
        ProxyRealIPCIDR:                  defIPCIDR,
        ProxyProtocolHeaderTimeout:       defProxyDeadlineDuration,
        ServerNameHashMaxSize:            1024,
        ProxyHeadersHashMaxSize:          512,
        ProxyHeadersHashBucketSize:       64,
        ProxyStreamResponses:             1,
        ReusePort:                        true,
        ShowServerTokens:                 false,
        SSLBufferSize:                    sslBufferSize,
        SSLCiphers:                       sslCiphers,
        SSLECDHCurve:                     "auto",
        SSLProtocols:                     sslProtocols,
        SSLEarlyData:                     sslEarlyData,
        SSLSessionCache:                  true,
        SSLSessionCacheSize:              sslSessionCacheSize,
        SSLSessionTickets:                false,
        SSLSessionTimeout:                sslSessionTimeout,
        EnableBrotli:                     false,
        UseGzip:                          false,
        UseGeoIP:                         true,
        UseGeoIP2:                        false,
        WorkerProcesses:                  strconv.Itoa(runtime.NumCPU()),
        WorkerShutdownTimeout:            "240s",
        VariablesHashBucketSize:          256,
        VariablesHashMaxSize:             2048,
        UseHTTP2:                         true,
        ProxyStreamTimeout:               "600s",
        Backend: defaults.Backend{
            ProxyBodySize:            bodySize,
            ProxyConnectTimeout:      5,
            ProxyReadTimeout:         60,
            ProxySendTimeout:         60,
            ProxyBuffersNumber:       4,
            ProxyBufferSize:          "4k",
            ProxyCookieDomain:        "off",
            ProxyCookiePath:          "off",
            ProxyNextUpstream:        "error timeout",
            ProxyNextUpstreamTimeout: 0,
            ProxyNextUpstreamTries:   3,
            ProxyRequestBuffering:    "on",
            ProxyRedirectFrom:        "off",
            ProxyRedirectTo:          "off",
            SSLRedirect:              true,
            CustomHTTPErrors:         []int{},
            WhitelistSourceRange:     []string{},
            SkipAccessLogURLs:        []string{},
            LimitRate:                0,
            LimitRateAfter:           0,
            ProxyBuffering:           "off",
            ProxyHTTPVersion:         "1.1",
            ProxyMaxTempFileSize:     "1024m",
        },
        UpstreamKeepaliveConnections: 320,
        UpstreamKeepaliveTimeout:     60,
        UpstreamKeepaliveRequests:    10000,
        LimitConnZoneVariable:        defaultLimitConnZoneVariable,
        BindAddressIpv4:              defBindAddress,
        BindAddressIpv6:              defBindAddress,
        ZipkinCollectorPort:          9411,
        ZipkinServiceName:            "nginx",
        ZipkinSampleRate:             1.0,
        JaegerCollectorPort:          6831,
        JaegerServiceName:            "nginx",
        JaegerSamplerType:            "const",
        JaegerSamplerParam:           "1",
        JaegerSamplerPort:            5778,
        JaegerSamplerHost:            "http://127.0.0.1",
        DatadogServiceName:           "nginx",
        DatadogEnvironment:           "prod",
        DatadogCollectorPort:         8126,
        DatadogOperationNameOverride: "nginx.handle",
        DatadogSampleRate:            1.0,
        DatadogPrioritySampling:      true,
        LimitReqStatusCode:           503,
        LimitConnStatusCode:          503,
        SyslogPort:                   514,
        NoTLSRedirectLocations:       "/.well-known/acme-challenge",
        NoAuthLocations:              "/.well-known/acme-challenge",
        GlobalExternalAuth:           defGlobalExternalAuth,
        ProxySSLLocationOnly:         false,
        DefaultType:                  "text/html",
    }

    if klog.V(5).Enabled() {
        cfg.ErrorLogLevel = "debug"
    }

    return cfg
}

// 最终传递到nginx.tmpl 那边去的配置结构体参数组装
// 后端upstream部分不需要过分关注，因为那部分是由lua去etcd数据库动态获取维护的。
// 整个配置是 nginx http全局性参数传递
// TemplateConfig contains the nginx configuration to render the file nginx.conf
type TemplateConfig struct {
    ProxySetHeaders          map[string]string
    AddHeaders               map[string]string
    BacklogSize              int
    Backends                 []*ingress.Backend //收集的backend相关列表被包含进去
    PassthroughBackends      []*ingress.SSLPassthroughBackend
    Servers                  []*ingress.Server //搜集的 ingress server类型相关参数 列表
    TCPBackends              []ingress.L4Service
    UDPBackends              []ingress.L4Service
    HealthzURI               string
    // 如果是全局性的直接可以参照上面可用参数设定调整 官方文档性说明 https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/
    // 如果只需要局部生效，需要对照 https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/ 进行投入投入使用，这些注解都是局部设定使用的。
    Cfg                      Configuration  //configuration类型的 Cfg 被包含在里面，这里面的参数就是上面解说的那些参数，基本都在全局可用，
    IsIPV6Enabled            bool
    IsSSLPassthroughEnabled  bool
    NginxStatusIpv4Whitelist []string
    NginxStatusIpv6Whitelist []string
    RedirectServers          interface{}
    ListenPorts              *ListenPorts
    PublishService           *apiv1.Service
    EnableMetrics            bool
    MaxmindEditionFiles      []string
    MonitorMaxBatchSize      int

    PID        string
    StatusPath string
    StatusPort int
    StreamPort int
}

// ListenPorts describe the ports required to run the
// NGINX Ingress controller
type ListenPorts struct {
    HTTP     int
    HTTPS    int
    Health   int
    Default  int
    SSLProxy int
}

// GlobalExternalAuth describe external authentication configuration for the
// NGINX Ingress controller
type GlobalExternalAuth struct {
    URL string `json:"url"`
    // Host contains the hostname defined in the URL
    Host              string            `json:"host"`
    SigninURL         string            `json:"signinUrl"`
    Method            string            `json:"method"`
    ResponseHeaders   []string          `json:"responseHeaders,omitempty"`
    RequestRedirect   string            `json:"requestRedirect"`
    AuthSnippet       string            `json:"authSnippet"`
    AuthCacheKey      string            `json:"authCacheKey"`
    AuthCacheDuration []string          `json:"authCacheDuration"`
    ProxySetHeaders   map[string]string `json:"proxySetHeaders,omitempty"`
}
```

// 后端backend结构参数说明

```
/*
Copyright 2017 The Kubernetes Authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

package defaults

import "net"

// Backend defines the mandatory configuration that an Ingress controller must provide
// The reason of this requirements is the annotations are generic. If some implementation do not supports
// one or more annotations it just can provides defaults
type Backend struct {
    // AppRoot contains the AppRoot for apps that doesn't exposes its content in the 'root' context
    // 设定 nginx root路径
    AppRoot string `json:"app-root"`

    // enables which HTTP codes should be passed for processing with the error_page directive
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_intercept_errors
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#error_page
    // By default this is disabled
    // 自定义http错误页面， 默认关闭
    CustomHTTPErrors []int `json:"custom-http-errors"`

    //
http://nginx.org/en/docs/http/ngx_http_core_module.html#client_max_body_siz
e
    // Sets the maximum allowed size of the client request body
    // 代理主体大小设定
    ProxyBodySize string `json:"proxy-body-size"`

    // Defines a timeout for establishing a connection with a proxied
server.
    // It should be noted that this timeout cannot usually exceed 75
seconds.
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_connect_time
out
    // 代理连接超时设定
    ProxyConnectTimeout int `json:"proxy-connect-timeout"`

    // Timeout in seconds for reading a response from the proxied server.
The timeout is set only between
    // two successive read operations, not for the transmission of the
whole response
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_read_timeout
    // 代理读取超时
    ProxyReadTimeout int `json:"proxy-read-timeout"`

    // Timeout in seconds for transmitting a request to the proxied server.
The timeout is set only between
    // two successive write operations, not for the transmission of the
whole request.
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_send_timeout
    // 代理发送超时
    ProxySendTimeout int `json:"proxy-send-timeout"`

    // Sets the number of the buffers used for reading a response from the
proxied server
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_buffers
    // 搭理缓冲数量
    ProxyBuffersNumber int `json:"proxy-buffers-number"`

    // Sets the size of the buffer used for reading the first part of the
response received from the
    // proxied server. This part usually contains a small response header.
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_buffer_size)
    // 代理缓冲大小
    ProxyBufferSize string `json:"proxy-buffer-size"`

    // Sets a text that should be changed in the path attribute of the
“Set-Cookie” header fields of
    // a proxied server response.
    //
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_cookie_path
```

```
    // 代理cookie路径
    ProxyCookiePath string `json:"proxy-cookie-path"`

    // Sets a text that should be changed in the domain attribute of the “Set-Cookie” header fields
    // of a proxied server response.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_cookie_domain
    // 代理cookie域名
    ProxyCookieDomain string `json:"proxy-cookie-domain"`

    // Specifies in which cases a request should be passed to the next server.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_next_upstream
    // 代理进行下个节点状态设定 比如 error 503 500
    ProxyNextUpstream string `json:"proxy-next-upstream"`

    // Limits the time during which a request can be passed to the next server.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_next_upstream_timeout
    // 检测故障超时进行切换
    ProxyNextUpstreamTimeout int `json:"proxy-next-upstream-timeout"`

    // Limits the number of possible tries for passing a request to the next server.
    // https://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_next_upstream_tries
    // 切换重试次数
    ProxyNextUpstreamTries int `json:"proxy-next-upstream-tries"`

    // Sets the original text that should be changed in the "Location" and "Refresh" header fields of a proxied server response.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_redirect
    // Default: off
    // 代理重定向来源
    ProxyRedirectFrom string `json:"proxy-redirect-from"`

    // Sets the replacement text that should be changed in the "Location" and "Refresh" header fields of a proxied server response.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_redirect
    // Default: off
    // 代理重定向去哪里
    ProxyRedirectTo string `json:"proxy-redirect-to"`

    // Enables or disables buffering of a client request body.
    //
```

```
http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_request_buff
ering
    // 开启代理请求缓冲
    ProxyRequestBuffering string `json:"proxy-request-buffering"`

    // Name server/s used to resolve names of upstream servers into IP
addresses.
    // The file /etc/resolv.conf is used as DNS resolution configuration.
    // dns 解析ip 列表
    Resolver []net.IP

    // SkipAccessLogURLs sets a list of URLs that should not appear in the
NGINX access log
    // This is useful with urls like `/health` or `health-check` that make
"complex" reading the logs
    // By default this list is empty
    // 那些url 跳过log记录
    SkipAccessLogURLs []string `json:"skip-access-log-urls"`

    // Enables or disables the redirect (301) to the HTTPS port
    // 开启自动跳转 ssl 301状态码
    SSLRedirect bool `json:"ssl-redirect"`

    // Enables or disables the redirect (301) to the HTTPS port even
without TLS cert
    // This is useful if doing SSL offloading outside of cluster eg AWS ELB
    // 强制ssl重定向
    ForceSSLRedirect bool `json:"force-ssl-redirect"`

    // Enables or disables the specification of port in redirects
    // Default: false
    // 开启指定端口重定向
    UsePortInRedirects bool `json:"use-port-in-redirects"`

    // Enable stickiness by client-server mapping based on a NGINX
variable, text or a combination of both.
    // A consistent hashing method will be used which ensures only a few
keys would be remapped to different
    // servers on upstream group changes
    // http://nginx.org/en/docs/http/ngx_http_upstream_module.html#hash
    // 开启upstream hash by 什么指定变量 粘连
    UpstreamHashBy string `json:"upstream-hash-by"`

    // Consistent hashing subset flag.
    // Default: false
    // 基于子网节点做hash一致性平衡
    UpstreamHashBySubset bool `json:"upstream-hash-by-subset"`

    // Subset consistent hashing, subset size.
    // Default 3
    // 基于子网节点做hash一致性平衡
    UpstreamHashBySubsetSize int `json:"upstream-hash-by-subset-size"`

    // Let's us choose a load balancing algorithm per ingress
    // 选择LB机制 rr least-conn 等等
    LoadBalancing string `json:"load-balance"`

    // WhitelistSourceRange allows limiting access to certain client addresses
    // http://nginx.org/en/docs/http/ngx_http_access_module.html
    // 白名单相关功能 基于cidr
    WhitelistSourceRange []string `json:"whitelist-source-range"`

    // Limits the rate of response transmission to a client.
    // The rate is specified in bytes per second. The zero value disables rate limiting.
    // The limit is set per a request, and so if a client simultaneously opens two connections,
    // the overall rate will be twice as much as the specified limit.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#limit_rate
    // 限流
    LimitRate int `json:"limit-rate"`

    // Sets the initial amount after which the further transmission of a response to a client will be rate limited.
    // http://nginx.org/en/docs/http/ngx_http_core_module.html#limit_rate_after
    // 限流
    LimitRateAfter int `json:"limit-rate-after"`

    // Enables or disables buffering of responses from the proxied server.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_buffering
    // 代理buffer设定开启
    ProxyBuffering string `json:"proxy-buffering"`

    // Modifies the HTTP version the proxy uses to interact with the backend.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_http_version
    // 代理版本开启
    ProxyHTTPVersion string `json:"proxy-http-version"`

    // Sets the maximum temp file size when proxy-buffers capacity is exceeded.
    // http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_max_temp_file_size
    // 代理最大临时文件大小设定
    ProxyMaxTempFileSize string `json:"proxy-max-temp-file-size"`
}
```