DevOps Sesión 4 (2022-02-21)

Documentación relacionada

- Traefik1.7.txt
- Laboratorio Traefik Docker/Docker Traefik.pdf
- Laboratorios docker-compose/Lab Docker-Compose-miaplicacion/*
- Laboratorios docker-compose/Lab Docker-Compose-miaplicacion/docker-compose acciones.txt

Clase

traefik

- https://traefik.io
- https://doc.traefik.io/traefik/v1.7/
- `docker run -d -p 8580:8080 -p 80:80 –network traefik –name traefik -v $PWD/traefik.toml:/etc/traefik/traefik.toml -v /var/run/docker.sock:/var/run/docker.sock traefik:1.7`
    - sticky sesion: en el caso de balanceo, fija una sesión a un contenedor
- escalar:
    - `docker-compose scale <SERVICIO>=3`
    - `docker-compose up --scale <SERVICIO>=3 -d`
- `docker compose up -d --force-recreate --build …`

    - ; traefik.toml

```
###############################################################
# Global configuration
###############################################################

# Duration to give active requests a chance to finish during hot-
reloads.
# Can be provided in a format supported by Go's time.ParseDuration
function or
# as raw values (digits). If no units are provided, the value is
parsed assuming
# seconds.
#
# Optional
# Default: "10s"
#
# graceTimeOut = "10s"

# Enable debug mode
#
# Optional
# Default: false
#
# debug = true

# Periodically check if a new version has been released
```

```
#
# Optional
# Default: true
#
# checkNewVersion = false

# Traefik logs file
# If not defined, logs to stdout
#
# Optional
#
# traefikLogsFile = "log/traefik.log"

# Access logs file
#
# Optional
# Deprecated - see [accessLog] lower down
#
accessLogsFile = "log/access.log"

# Log level
#
# Optional
# Default: "ERROR"
#
logLevel = "DEBUG"

# Backends throttle duration: minimum duration in seconds between 2
events from providers
# before applying a new configuration. It avoids unnecessary reloads
if multiples events
# are sent in a short amount of time.
# Can be provided in a format supported by Go's time.ParseDuration
function or
# as raw values (digits). If no units are provided, the value is
parsed assuming
# seconds.
#
# Optional
# Default: "2s"
#
# ProvidersThrottleDuration = "5s"

# Controls the maximum idle (keep-alive) connections to keep per-
host. If zero, DefaultMaxIdleConnsPerHost
# from the Go standard library net/http module is used.
# If you encounter 'too many open files' errors, you can either
increase this
# value or change the `ulimit`.
#
# Optional
# Default: 200
#
# MaxIdleConnsPerHost = 200

# If set to true invalid SSL certificates are accepted for backends.
```

```
# Note: This disables detection of man-in-the-middle attacks so
should only be used on secure backend networks.
# Optional
# Default: false
#
# InsecureSkipVerify = true

# Entrypoints to be used by frontends that do not specify any
entrypoint.
# Each frontend can specify its own entrypoints.
#
# Optional
# Default: ["http"]
#
# defaultEntryPoints = ["http", "https"]
defaultEntryPoints = ["http"]

# Constraints definition
#
# Optional
#
# Simple matching constraint
# constraints = ["tag==api"]
#
# Simple mismatching constraint
# constraints = ["tag!=api"]
#
# Globbing
# constraints = ["tag==us-*"]
#
# Backend-specific constraint
# [consulCatalog]
#   endpoint = "127.0.0.1:8500"
#   constraints = ["tag==api"]
#
# Multiple constraints
#   - "tag==" must match with at least one tag
#   - "tag!=" must match with none of tags
# constraints = ["tag!=us-*", "tag!=asia-*"]
# [consulCatalog]
#   endpoint = "127.0.0.1:8500"
#   constraints = ["tag==api", "tag!=v*-beta"]

# Enable ACME (Let's Encrypt): automatic SSL
#
# Optional
#
# [acme]

# Email address used for registration
#
# Required
#
# email = "test@traefik.io"

# File or key used for certificates storage.
```

```
# WARNING, if you use Traefik in Docker, you have 2 options:
#  - create a file on your host and mount it as a volume
#      storageFile = "acme.json"
#      $ docker run -v "/my/host/acme.json:acme.json" traefik
#  - mount the folder containing the file as a volume
#      storageFile = "/etc/traefik/acme/acme.json"
#      $ docker run -v "/my/host/acme:/etc/traefik/acme" traefik
#
# Required
#
# storage = "acme.json" # or "traefik/acme/account" if using KV
store

# Entrypoint to proxy acme challenge/apply certificates to.
# WARNING, must point to an entrypoint on port 443
#
# Required
#
# entryPoint = "https"

# Use a DNS based acme challenge rather than external HTTPS access,
e.g. for a firewalled server
# Select the provider that matches the DNS domain that will host the
challenge TXT record,
# and provide environment variables with access keys to enable
setting it:
#  - cloudflare: CLOUDFLARE_EMAIL, CLOUDFLARE_API_KEY
#  - digitalocean: DO_AUTH_TOKEN
#  - dnsimple: DNSIMPLE_EMAIL, DNSIMPLE_API_KEY
#  - dnsmadeeasy: DNSMADEEASY_API_KEY, DNSMADEEASY_API_SECRET
#  - exoscale: EXOSCALE_API_KEY, EXOSCALE_API_SECRET
#  - gandi: GANDI_API_KEY
#  - linode: LINODE_API_KEY
#  - manual: none, but run traefik interactively & turn on
acmeLogging to see instructions & press Enter
#  - namecheap: NAMECHEAP_API_USER, NAMECHEAP_API_KEY
#  - rfc2136: RFC2136_TSIG_KEY, RFC2136_TSIG_SECRET,
RFC2136_TSIG_ALGORITHM, RFC2136_NAMESERVER
#  - route53: AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, AWS_REGION,
or configured user/instance IAM profile
#  - dyn: DYN_CUSTOMER_NAME, DYN_USER_NAME, DYN_PASSWORD
#  - vultr: VULTR_API_KEY
#  - ovh: OVH_ENDPOINT, OVH_APPLICATION_KEY, OVH_APPLICATION_SECRET,
OVH_CONSUMER_KEY
#  - pdns: PDNS_API_KEY, PDNS_API_URL
#
# Optional
#
# dnsProvider = "digitalocean"

# By default, the dnsProvider will verify the TXT DNS challenge
record before letting ACME verify
# If delayDontCheckDNS is greater than zero, avoid this & instead
just wait so many seconds.
# Useful if internal networks block external DNS queries
#
```

```
# Optional
#
# delayDontCheckDNS = 0

# If true, display debug log messages from the acme client library
#
# Optional
#
# acmeLogging = true

# Enable on demand certificate. This will request a certificate from
Let's Encrypt during the first TLS handshake for a hostname that
does not yet have a certificate.
# WARNING, TLS handshakes will be slow when requesting a hostname
certificate for the first time, this can leads to DoS attacks.
# WARNING, Take note that Let's Encrypt have rate limiting:
https://letsencrypt.org/docs/rate-limits
#
# Optional
#
# onDemand = true

# Enable certificate generation on frontends Host rules. This will
request a certificate from Let's Encrypt for each frontend with a
Host rule.
# For example, a rule Host:test1.traefik.io,test2.traefik.io will
request a certificate with main domain test1.traefik.io and SAN
test2.traefik.io.
#
# Optional
#
# OnHostRule = true

# CA server to use
# Uncomment the line to run on the staging let's encrypt server
# Leave comment to go to prod
#
# Optional
#
# caServer = "https://acme-staging.api.letsencrypt.org/directory"

# Domains list
# You can provide SANs (alternative domains) to each main domain
# All domains must have A/AAAA records pointing to Traefik
# WARNING, Take note that Let's Encrypt have rate limiting:
https://letsencrypt.org/docs/rate-limits
# Each domain & SANs will lead to a certificate request.
#
# [[acme.domains]]
#   main = "local1.com"
#   sans = ["test1.local1.com", "test2.local1.com"]
# [[acme.domains]]
#   main = "local2.com"
#   sans = ["test1.local2.com", "test2x.local2.com"]
# [[acme.domains]]
#   main = "local3.com"
```

```
# [[acme.domains]]
#   main = "local4.com"


# Set access log options
#
# Optional
#
# [accessLog]

# Sets the file path for the access log. If none is given (the
default)
# no access logs are produced. Intermediate directories are created
if
# necessary.
#
# Optional
# Default: ""
#
# filePath = "/path/to/log/log.txt"
#

# Format is either "json" or "common".
#
# Optional
# Default: "common"
#
# format = "common"
#

# Entrypoints definition
#
# Optional
# Default:
[entryPoints]
  [entryPoints.http]
    address = ":80"
#
# To redirect an http entrypoint to an https entrypoint (with SNI
support):
# [entryPoints]
#   [entryPoints.http]
#   address = ":80"
#     [entryPoints.http.redirect]
#       entryPoint = "https"
#   [entryPoints.https]
#   address = ":443"
#     [entryPoints.https.tls]
#       [[entryPoints.https.tls.certificates]]
#       CertFile = "integration/fixtures/https/snitest.com.cert"
#       KeyFile = "integration/fixtures/https/snitest.com.key"
#       [[entryPoints.https.tls.certificates]]
#       CertFile = "integration/fixtures/https/snitest.org.cert"
#       KeyFile = "integration/fixtures/https/snitest.org.key"
#
# To redirect an entrypoint rewriting the URL:
```

```
# [entryPoints]
#   [entryPoints.http]
#   address = ":80"
#     [entryPoints.http.redirect]
#       regex = "^http://localhost/(.*)"
#       replacement = "http://mydomain/$1"
#
# To enable basic auth on an entrypoint
# with 2 user/pass: test:test and test2:test2
# Passwords can be encoded in MD5, SHA1 and BCrypt: you can use
htpasswd to generate those ones
# Users can be specified directly in the toml file, or indirectly by
referencing an external file; if both are provided, the two are
merged, with external file contents having precedence
# [entryPoints]
#   [entryPoints.http]
#   address = ":80"
#   [entryPoints.http.auth.basic]
#   users = ["test:$apr1$H6uskkkW$IgXLP6ewTrSuBkTrqE8wj/",
"test2:$apr1$d9hr9HBB$4HxwgUir3HP4EsggP/QNo0"]
#   usersFile = "/path/to/.htpasswd"
#
# To enable digest auth on an entrypoint
# with 2 user/realm/pass: test:traefik:test and test2:traefik:test2
# You can use htdigest to generate those ones
# Users can be specified directly in the toml file, or indirectly by
referencing an external file; if both are provided, the two are
merged, with external file contents having precedence
# [entryPoints]
#   [entryPoints.http]
#   address = ":80"
#   [entryPoints.http.auth.basic]
#   users = ["test:traefik:a2688e031edb4be6a3797f3882655c05 ",
"test2:traefik:518845800f9e2bfb1f1f740ec24f074e"]
#   usersFile = "/path/to/.htdigest"
#
# To specify an https entrypoint with a minimum TLS version, and
specifying an array of cipher suites (from crypto/tls):
# [entryPoints]
#   [entryPoints.https]
#   address = ":443"
#     [entryPoints.https.tls]
#     MinVersion = "VersionTLS12"
#     CipherSuites = ["TLS_RSA_WITH_AES_256_GCM_SHA384"]
#       [[entryPoints.https.tls.certificates]]
#       CertFile = "integration/fixtures/https/snitest.com.cert"
#       KeyFile = "integration/fixtures/https/snitest.com.key"
#       [[entryPoints.https.tls.certificates]]
#       CertFile = "integration/fixtures/https/snitest.org.cert"
#       KeyFile = "integration/fixtures/https/snitest.org.key"

# To enable compression support using gzip format:
# [entryPoints]
#   [entryPoints.http]
#   address = ":80"
#   compress = true
```

```
# To bind to a particular IP address only:
# [entryPoints]
#   [entryPoints.http]
#   address = "10.42.13.37:80"

# Enable retry sending request if network error
#
# Optional
#
# [retry]

# Number of attempts
#
# Optional
# Default: (number servers in backend) -1
#
# attempts = 3

# Enable custom health check options.
#
# Optional
#
# [healthcheck]

# Set the default health check interval. Will only be effective if
health check
# paths are defined. Given provider-specific support, the value may
be
# overridden on a per-backend basis.
# Can be provided in a format supported by Go's time.ParseDuration
function or
# as raw values (digits). If no units are provided, the value is
parsed assuming
# seconds.
#
# Optional
# Default: "30s"
#
# interval = "30s"

###############################################################
# Web configuration backend
###############################################################

# Enable web configuration backend
#
# Optional
#
 [web]

# Web administration port
#
# Required
#
 address = ":8080"
```

```
# SSL certificate and key used
#
# Optional
#
# CertFile = "traefik.crt"
# KeyFile = "traefik.key"
#
# Set REST API to read-only mode
#
# Optional
# ReadOnly = false
#
# Enable more detailed statistics
# [web.statistics]
#   RecentErrors = 10
#
# To enable Traefik to export internal metrics to Prometheus
# [web.metrics.prometheus]
#   Buckets=[0.1,0.3,1.2,5.0]
#

# To enable basic auth on the webui
# with 2 user/pass: test:test and test2:test2
# Passwords can be encoded in MD5, SHA1 and BCrypt: you can use
htpasswd to generate those ones
# Users can be specified directly in the toml file, or indirectly by
referencing an external file; if both are provided, the two are
merged, with external file contents having precedence
#   [web.auth.basic]
#     users = ["test:$apr1$H6uskkkW$IgXLP6ewTrSuBkTrqE8wj/",
"test2:$apr1$d9hr9HBB$4HxwgUir3HP4EsggP/QNo0"]
#     usersFile = "/path/to/.htpasswd"
# To enable digest auth on the webui
# with 2 user/realm/pass: test:traefik:test and test2:traefik:test2
# You can use htdigest to generate those ones
# Users can be specified directly in the toml file, or indirectly by
referencing an external file; if both are provided, the two are
merged, with external file contents having precedence
#   [web.auth.digest]
#     users = ["test:traefik:a2688e031edb4be6a3797f3882655c05 ",
"test2:traefik:518845800f9e2bfb1f1f740ec24f074e"]
#     usersFile = "/path/to/.htdigest"


################################################################
# File configuration backend
################################################################

# Enable file configuration backend
#
# Optional
#
# [file]

# Rules file
```

```
# If defined, traefik will load rules from this file,
# otherwise, it will load rules from current file (cf Sample rules
below).
#
# Optional
#
# filename = "rules.toml"

# Enable watch file changes
#
# Optional
#
# watch = true


################################################################
# Docker configuration backend
################################################################

# Enable Docker configuration backend
#
# Optional
#
 [docker]

# Docker server endpoint. Can be a tcp or a unix socket endpoint.
#
# Required
#
 endpoint = "unix:///var/run/docker.sock"

# Default domain used.
# Can be overridden by setting the "traefik.domain" label on a
container.
#
# Required
#
 domain = "curso.local"

# Enable watch docker changes
#
# Optional
#
 watch = true

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "docker.tmpl"

# Expose containers by default in traefik
#
# Optional
# Default: true
#
```

```
# exposedbydefault = true
usebindportip = true
# Enable docker TLS connection
#
# Optional
#
#  [docker.tls]
#  ca = "/etc/ssl/ca.crt"
#  cert = "/etc/ssl/docker.crt"
#  key = "/etc/ssl/docker.key"
#  insecureskipverify = true



################################################################
# Docker Swarmmode configuration backend
################################################################

# Enable Docker configuration backend
#
# Optional
#
# [docker]

# Docker server endpoint. Can be a tcp or a unix socket endpoint.
#
# Required
#
# endpoint = "tcp://127.0.0.1:2375"

# Default domain used.
# Can be overridden by setting the "traefik.domain" label on a
services.
#
# Required
#
# domain = "docker.localhost"

# Enable watch docker changes
#
# Optional
#
# watch = true

# Use Docker Swarm Mode as data provider
#
# Optional
#
 swarmmode = false

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "docker.tmpl"
```

```
# Expose services by default in traefik
#
# Optional
# Default: true
#
# exposedbydefault = true

# Enable docker TLS connection
#
# Optional
#
#  [swarm.tls]
#  ca = "/etc/ssl/ca.crt"
#  cert = "/etc/ssl/docker.crt"
#  key = "/etc/ssl/docker.key"
#  insecureskipverify = true

# Constraints
#
# Optional
#
# constraints = ["tag==api", "tag==he*ld"]
# Matching with containers having the label "traefik.tags" set to
"api,helloworld"
# ex: $ docker run -d -P --label traefik.tags=api,helloworld
emilevauge/whoami


################################################################
# Mesos/Marathon configuration backend
################################################################

# Enable Marathon configuration backend
#
# Optional
#
# [marathon]

# Marathon server endpoint.
# You can also specify multiple endpoint for Marathon:
# endpoint :=
"http://10.241.1.71:8080,10.241.1.72:8080,10.241.1.73:8080"
#
# Required
#
# endpoint = "http://127.0.0.1:8080"

# Enable watch Marathon changes
#
# Optional
#
# watch = true

# Default domain used.
# Can be overridden by setting the "traefik.domain" label on an
application.
```

```
#
# Required
#
# domain = "marathon.localhost"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "marathon.tmpl"

# Expose Marathon apps by default in traefik
#
# Optional
# Default: true
#
# exposedByDefault = true

# Convert Marathon groups to subdomains
# Default behavior: /foo/bar/myapp => foo-bar-myapp.{defaultDomain}
# with groupsAsSubDomains enabled: /foo/bar/myapp =>
myapp.bar.foo.{defaultDomain}
#
# Optional
# Default: false
#
# groupsAsSubDomains = true

# Override DialerTimeout
# Amount of time to allow the Marathon provider to wait to open a
TCP connection
# to a Marathon master.
# Can be provided in a format supported by Go's time.ParseDuration
function or
# as raw values (digits). If no units are provided, the value is
parsed assuming
# seconds.
#
# Optional
# Default: "60s"
# dialerTimeout = "60s"

# Enable Marathon basic authentication
#
# Optional
#
#  [marathon.basic]
#  httpBasicAuthUser = "foo"
#  httpBasicPassword = "bar"

# DCOSToken for DCOS environment, This will override the
Authorization header
#
# Optional
#
# dcosToken = "xxxxxx"
```

```
# Set the TCP Keep Alive interval for the Marathon HTTP Client.
# Can be provided in a format supported by Go's time.ParseDuration
function or
# as raw values (digits). If no units are provided, the value is
parsed assuming
# seconds.
#
# Optional
# Default: "10s"
#
# keepAlive = "10s"

# By default, a task's IP address (as returned by the Marathon API)
is used as
# backend server if an IP-per-task configuration can be found;
otherwise, the
# name of the host running the task is used.
# The latter behavior can be enforced by enabling this switch.
#
# Optional
# Default: false
#
# forceTaskHostname: false

################################################################
# Mesos configuration backend
################################################################

# Enable Mesos configuration backend
#
# Optional
#
# [mesos]

# Mesos server endpoint.
# You can also specify multiple endpoint for Mesos:
# endpoint =
"192.168.35.40:5050,192.168.35.41:5050,192.168.35.42:5050"
# endpoint =
"zk://192.168.35.20:2181,192.168.35.21:2181,192.168.35.22:2181/mesos
"
#
# Required
#
# endpoint = "http://127.0.0.1:8080"

# Enable watch Mesos changes
#
# Optional
#
# watch = true

# Default domain used.
# Can be overridden by setting the "traefik.domain" label on an
```

```
application.
#
# Required
#
# domain = "mesos.localhost"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "mesos.tmpl"

# Expose Mesos apps by default in traefik
#
# Optional
# Default: false
#
# ExposedByDefault = true

# TLS client configuration.
https://golang.org/pkg/crypto/tls/#Config
#
# Optional
#
# [mesos.TLS]
# InsecureSkipVerify = true

# Zookeeper timeout (in seconds)
#
# Optional
# Default: 30
#
# ZkDetectionTimeout = 30

# Polling interval (in seconds)
#
# Optional
# Default: 30
#
# RefreshSeconds = 30

# IP sources (e.g. host, docker, mesos, rkt)
#
# Optional
#
# IPSources = "host"

# HTTP Timeout (in seconds)
#
# Optional
# Default: 30
#
# StateTimeoutSecond = "30"

################################################################
# Kubernetes Ingress configuration backend
```

```
################################################################
# Enable Kubernetes Ingress configuration backend
#
# Optional
#
# [kubernetes]

# Kubernetes server endpoint
#
# When deployed as a replication controller in Kubernetes, Traefik
will use
# the environment variables KUBERNETES_SERVICE_HOST and
KUBERNETES_SERVICE_PORT
# to construct the endpoint.
# Secure token will be found in
/var/run/secrets/kubernetes.io/serviceaccount/token
# and SSL CA cert in
/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
#
# The endpoint may be given to override the environment variable
values.
#
# When the environment variables are not found, Traefik will try to
connect to
# the Kubernetes API server with an external-cluster client. In this
case, the
# endpoint is required. Specifically, it may be set to the URL used
by
# `kubectl proxy` to connect to a Kubernetes cluster from localhost.
#
# Optional for in-cluster configuration, required otherwise
# Default: empty
#
# endpoint = "http://127.0.0.1:8001"

# Bearer token used for the Kubernetes client configuration.
#
# Optional
# Default: empty
#
# token = "my token"

# Path to the certificate authority file used for the Kubernetes
client
# configuration.
#
# Optional
# Default: empty
#
# certAuthFilePath = "/my/ca.crt"

# Array of namespaces to watch.
#
# Optional
# Default: ["default"].
#
```

```
# namespaces = ["default"]

# See:
http://kubernetes.io/docs/user-guide/labels/#list-and-watch-filterin
g
# labelselector = "A and not B"

################################################################
# Consul KV configuration backend
################################################################

# Enable Consul KV configuration backend
#
# Optional
#
# [consul]

# Consul server endpoint
#
# Required
#
# endpoint = "127.0.0.1:8500"

# Enable watch Consul changes
#
# Optional
#
# watch = true

# Prefix used for KV store.
#
# Optional
#
# prefix = "traefik"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "consul.tmpl"

# Enable consul TLS connection
#
# Optional
#
# [consul.tls]
# ca = "/etc/ssl/ca.crt"
# cert = "/etc/ssl/consul.crt"
# key = "/etc/ssl/consul.key"
# insecureskipverify = true

################################################################
# Consul Catalog configuration backend
################################################################

# Enable Consul Catalog configuration backend
```

```
#
# Optional
#
# [consulCatalog]

# Consul server endpoint
#
# Required
#
# endpoint = "127.0.0.1:8500"

# Default domain used.
#
# Optional
#
# domain = "consul.localhost"

# Prefix for Consul catalog tags
#
# Optional
#
# prefix = "traefik"

# Constraints
#
# Optional
#
# constraints = ["tag==api", "tag==he*ld"]
# Matching with containers having this tag:
"traefik.tags=api,helloworld"

################################################################
# Etcd configuration backend
################################################################

# Enable Etcd configuration backend
#
# Optional
#
# [etcd]

# Etcd server endpoint
#
# Required
#
# endpoint = "127.0.0.1:2379"

# Enable watch Etcd changes
#
# Optional
#
# watch = true

# Prefix used for KV store.
#
# Optional
```

```
#
# prefix = "/traefik"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "etcd.tmpl"

# Use etcd user/pass authentication
#
# Optional
#
# username = foo
# password = bar

# Enable etcd TLS connection
#
# Optional
#
# [etcd.tls]
# ca = "/etc/ssl/ca.crt"
# cert = "/etc/ssl/etcd.crt"
# key = "/etc/ssl/etcd.key"
# insecureskipverify = true

##################################################################
# Zookeeper configuration backend
##################################################################

# Enable Zookeeperconfiguration backend
#
# Optional
#
# [zookeeper]

# Zookeeper server endpoint
#
# Required
#
# endpoint = "127.0.0.1:2181"

# Enable watch Zookeeper changes
#
# Optional
#
# watch = true

# Prefix used for KV store.
#
# Optional
#
# prefix = "/traefik"

# Override default configuration template. For advanced users :)
#
```

```
# Optional
#
# filename = "zookeeper.tmpl"


################################################################
# BoltDB configuration backend
################################################################

# Enable BoltDB configuration backend
#
# Optional
#
# [boltdb]

# BoltDB file
#
# Required
#
# endpoint = "/my.db"

# Enable watch BoltDB changes
#
# Optional
#
# watch = true

# Prefix used for KV store.
#
# Optional
#
# prefix = "/traefik"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "boltdb.tmpl"


################################################################
# ECS configuration backend
################################################################

# Enable ECS configuration backend
#
# Optional
#
# [ecs]

# ECS Cluster Name
#
# Optional
# Default: "default"
#
# Cluster = "default"
```

```
# Enable watch ECS changes
#
# Optional
# Default: true
#
# Watch = true

# Polling interval (in seconds)
#
# Optional
# Default: 15
#
# RefreshSeconds = 15

# Expose ECS services by default in traefik
#
# Optional
# Default: true
#
# ExposedByDefault = false

# Region to use when connecting to AWS
#
# Optional
#
# Region = "us-east-1"

# AccessKeyID to use when connecting to AWS
#
# Optional
#
# AccessKeyID = "abc"

# SecretAccessKey to use when connecting to AWS
#
# Optional
#
# SecretAccessKey = "123"

# Override default configuration template. For advanced users :)
#
# Optional
#
# filename = "ecs.tmpl"

################################################################
# Rancher configuration backend
################################################################

# Enable Rancher configuration backend
#
# Optional
#
# [rancher]
```

```
# Default domain used.
# Can be overridden by setting the "traefik.domain" label on an
service.
#
# Required
#
# domain = "rancher.localhost"

# Enable watch Rancher changes
#
# Optional
# Default: true
#
# Watch = true

# Polling interval (in seconds)
#
# Optional
#
# RefreshSeconds = 15

# Expose Rancher services by default in traefik
#
# Optional
# Default: true
#
# ExposedByDefault = false

# Filter services with unhealthy states and health states
#
# Optional
# Default: false
#
# EnableServiceHealthFilter = false

# Endpoint to use when connecting to Rancher
#
# Required
# Endpoint = "http://rancherserver.example.com/v1"

# AccessKey to use when connecting to Rancher
#
# Required
# AccessKey = "XXXXXXXXXXXXXXXXXXX"

# SecretKey to use when connecting to Rancher
#
# Required
# SecretKey = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

# Constraints
#
# Optional
#
# constraints = ["tag==api", "tag==chee*e"]
# Matching with containers having this tag:
```

```
"traefik.tags=cheese,wine"

################################################################
# DynamoDB configuration backend
################################################################

# Enable DynamoDB configuration backend
#
# Optional
#
# [dynamodb]

# DynamoDB Table Name
#
# Optional
#
# TableName = "traefik"

# Enable watch DynamoDB changes
#
# Optional
#
# Watch = true

# Polling interval (in seconds)
#
# Optional
#
# RefreshSeconds = 15

# Region to use when connecting to AWS
#
# Required
#
# Region = "us-east-1"

# AccessKeyID to use when connecting to AWS
#
# Optional
#
# AccessKeyID = "abc"

# SecretAccessKey to use when connecting to AWS
#
# Optional
#
# SecretAccessKey = "123"

# Endpoint of dynamodb when testing locally
#
# Optional
#
# Endpoint = "http://localhost:8080"


################################################################
```

```
# Sample rules
################################################################
# [backends]
#   [backends.backend1]
#     [backends.backend1.circuitbreaker]
#       expression = "NetworkErrorRatio() > 0.5"
#     [backends.backend1.servers.server1]
#     url = "http://172.17.0.2:80"
#     weight = 10
#     [backends.backend1.servers.server2]
#     url = "http://172.17.0.3:80"
#     weight = 1
#   [backends.backend2]
#     [backends.backend2.LoadBalancer]
#       method = "drr"
#     [backends.backend2.servers.server1]
#     url = "http://172.17.0.4:80"
#     weight = 1
#     [backends.backend2.servers.server2]
#     url = "http://172.17.0.5:80"
#     weight = 2
#
# [frontends]
#   [frontends.frontend1]
#   backend = "backend2"
#     [frontends.frontend1.routes.test_1]
#     rule = "Host: test.localhost, other.localhost"
#   [frontends.frontend2]
#   backend = "backend1"
#   passHostHeader = true
#   entrypoints = ["https"] # overrides defaultEntryPoints
#     [frontends.frontend2.routes.test_1]
#     rule = "Host:{subdomain:[a-z]+}.localhost"
#   [frontends.frontend3]
#   entrypoints = ["http", "https"] # overrides defaultEntryPoints
#   backend = "backend2"
#     rule = "Path: /test, /other"
```

- https://www.avante.es/video-killed-the-radio-star-o-como-kubernetes-ha-matado-a-docker-o-no/

swarm

- DOCs:
  - 2-Docker Swarm.pdf
  - Clase explicacion Swarm.txt
  - Seminario Orquestadores de Contenedores Swarn kubernetes .pdf
  - https://www.youtube.com/channel/UCaL3Poyh7AWECpI5Ee_38xw/featured
- orquestador
- ejecución contenedores en varios nodos
- stacks ≡ docker-compose.yml
- servicios = unidad mínima
- master/esclavo
  - bdd:
    - **etcd**

        - distribuida (entre los masters)
        - clave=valor
    - protocolo ralph para escoger master
    - servidor DNS
- **/var/lib/docker/swarm**: información del swarm, por cada nodo.
- `docker swarm init`
    - `docker swarm join-token worker`
    - `docker swarm join-token master`
- creación red **overlay** → `docker network ls`
- `docker node inspect <NODO> [–pretty]]`
- `docker network inspect ingress`
- servicios
    - modos:
        - replicado (por defecto)
        - global: entornos de monitorización. se crea un contenedor en cada nodo. no permite escalado
- técnicas desplagado
    - blue-green: cambio en ingress de versiones (implica tener las 2 versiones en marcha)
    - canary: cambio progresivo
    - A/B
- `docker node update –availabilty`
    - active: modo normal de funcionamiento
    - pause: no acepta nuevos servicios
    - drain: se va vaciando el nodo de servicios
- constrains
    - `docker service create –name cluster1 –constraint «node.role == worker» -p:81:80/tcp russmckendrick/cluster`
- `-p:8080:80` → `--publish published=8080,target=80`

Extra

- Rancher Desktop → https://rancherdesktop.io
    - sustituto de docker Desktop (licencia)
    - docker, compose, kubernetes
- nip.io
    - traefik.me

TODO