Komodo

https://komo.do/

- Komodo - Background
- Komodo - Setup with Postgres
- Komodo - Advanced Configuration OIDC / Oauth2

Komodo - Background

Komo.do adalah sebuah sistem untuk membangun dan menerapkan perangkat lunak di banyak server. Sistem ini menawarkan pembuatan otomatis gambar Docker yang diberi versi otomatis dari repositori git, serta pemicu pembuatan saat ada push pada git. Selain itu, Komo.do memungkinkan penerapan kontainer Docker, termasuk Docker Compose, dan menyediakan fitur untuk memantau uptime serta log di semua server yang Anda miliki. Inti API dan agen periphery dari Komo.do ditulis dalam bahasa pemrograman Rust.

Komodo - Setup with Postgres

Sebelum menginstal Komo.do, pastikan Docker sudah terinstal. Panduan instalasi Docker dapat ditemukan di tautan berikut : Docker Install CE & Docker Compose Install.

1. Salin `komodo/postgres.compose.yaml` dan `komodo/compose.env` ke host Anda:

```
wget -P komodo https://raw.githubusercontent.com/moghtech/komodo/main/compose/postgres.compose.yaml && \
  wget -P komodo https://raw.githubusercontent.com/moghtech/komodo/main/compose/compose.env
```

2. Edit variabel dalam `komodo/compose.env`.
3. Menyebarkan:

```
docker compose -p komodo -f komodo/postgres.compose.yaml --env-file komodo/compose.env up -d
```

Config Default postgres.compose.yml (Github)

```
###################################
# □□ KOMODO COMPOSE - POSTGRES □□ #
###################################

## This compose file will deploy:
##   1. Postgres + FerretDB Mongo adapter (https://www.ferretdb.com)
##   2. Komodo Core
##   3. Komodo Periphery

services:
  postgres:
    image: postgres:17
    labels:
      komodo.skip: # Prevent Komodo from stopping with StopAllContainers
    restart: unless-stopped
    logging:
      driver: ${COMPOSE_LOGGING_DRIVER:-local}
```

```
  networks:
    - default
  # ports:
  #   - 5432:5432
  volumes:
    - pg-data:/var/lib/postgresql/data
  environment:
    - POSTGRES_USER=${KOMODO_DB_USERNAME}
    - POSTGRES_PASSWORD=${KOMODO_DB_PASSWORD}
    - POSTGRES_DB=${KOMODO_DATABASE_DB_NAME:-komodo}

ferretdb:
  image: ghcr.io/ferretdb/ferretdb:1
  labels:
    komodo.skip: # Prevent Komodo from stopping with StopAllContainers
  restart: unless-stopped
  depends_on:
    - postgres
  logging:
    driver: ${COMPOSE_LOGGING_DRIVER:-local}
  networks:
    - default
  # ports:
  #   - 27017:27017
  environment:
    - FERRETDB_POSTGRESQL_URL=postgres://postgres:5432/${KOMODO_DATABASE_DB_NAME:-komodo}

core:
  image: ghcr.io/mbecker20/komodo:${COMPOSE_KOMODO_IMAGE_TAG:-latest}
  labels:
    komodo.skip: # Prevent Komodo from stopping with StopAllContainers
  restart: unless-stopped
  depends_on:
    - ferretdb
  logging:
    driver: ${COMPOSE_LOGGING_DRIVER:-local}
  networks:
    - default
  ports:
    - 9120:9120
```

```
    env_file: ./compose.env
    environment:
      KOMODO_DATABASE_URI:
mongodb://${KOMODO_DB_USERNAME}:${KOMODO_DB_PASSWORD}@ferretdb:27017/${KOMODO_DATABASE_
DB_NAME:-komodo}?authMechanism=PLAIN
    volumes:
      ## Core cache for repos for latest commit hash / contents
      - repo-cache:/repo-cache
      ## Store sync files on server
      # - /path/to/syncs:/syncs
      ## Optionally mount a custom core.config.toml
      # - /path/to/core.config.toml:/config/config.toml
    ## Allows for systemd Periphery connection at
    ## "http://host.docker.internal:8120"
    # extra_hosts:
    #   - host.docker.internal:host-gateway

  ## Deploy Periphery container using this block,
  ## or deploy the Periphery binary with systemd using
  ## https://github.com/mbecker20/komodo/tree/main/scripts
  periphery:
    image: ghcr.io/mbecker20/periphery:${COMPOSE_KOMODO_IMAGE_TAG:-latest}
    labels:
      komodo.skip: # Prevent Komodo from stopping with StopAllContainers
    restart: unless-stopped
    logging:
      driver: ${COMPOSE_LOGGING_DRIVER:-local}
    networks:
      - default
    env_file: ./compose.env
    environment:
      PERIPHERY_REPO_DIR: ${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}/repos
      PERIPHERY_STACK_DIR: ${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}/stacks
      PERIPHERY_SSL_KEY_FILE: ${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}/ssl/key.pem
      PERIPHERY_SSL_CERT_FILE: ${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}/ssl/cert.pem
    volumes:
      ## Mount external docker socket
      - /var/run/docker.sock:/var/run/docker.sock
      ## Allow Periphery to see processes outside of container
      - /proc:/proc
```

```
      ## Specify the Periphery agent root directory.
      ## Must be the same inside and outside the container,
      ## or docker will get confused. See https://github.com/mbecker20/komodo/discussions/180.
      ## Default: /etc/komodo.
      - ${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}:${PERIPHERY_ROOT_DIRECTORY:-/etc/komodo}

volumes:
  # Postgres
  pg-data:
  # Core
  repo-cache:

networks:
  default: {}
```

Config Defauft compose.env ([Github]())

```
###################################
# □□ KOMODO COMPOSE - VARIABLES □□ #
###################################

## These compose variables can be used with all Komodo deployment options.
## Pass these variables to the compose up command using `--env-file komodo/compose.env`.
## Additionally, they are passed to both Komodo Core and Komodo Periphery with `env_file: ./compose.env`,
## so you can pass any additional environment variables to Core / Periphery directly in this file as well.

## Stick to a specific version, or use `latest`
COMPOSE_KOMODO_IMAGE_TAG=latest

## Note: □□ Podman does NOT support local logging driver □□. See Podman options here:
## `https://docs.podman.io/en/v4.6.1/markdown/podman-run.1.html#log-driver-driver`
COMPOSE_LOGGING_DRIVER=local # Enable log rotation with the local driver.

## DB credentials - Ignored for Sqlite
KOMODO_DB_USERNAME=admin
KOMODO_DB_PASSWORD=admin

## Configure a secure passkey to authenticate between Core / Periphery.
KOMODO_PASSKEY=a_random_passkey
```

```
#=--------------------------=#
#= Komodo Core Environment =#
#=--------------------------=#

## Full variable list + descriptions are available here:
## 🦎 https://github.com/mbecker20/komodo/blob/main/config/core.config.toml 🦎

## Note. Secret variables also support `${VARIABLE}_FILE` syntax to pass docker compose secrets.
## Docs: https://docs.docker.com/compose/how-tos/use-secrets/#examples

## Used for Oauth / Webhook url suggestion / Caddy reverse proxy.
KOMODO_HOST=https://demo.komo.do
## Displayed in the browser tab.
KOMODO_TITLE=Komodo
## Create a server matching this address as the "first server".
## Use `https://host.docker.internal:8120` when using systemd-managed Periphery.
KOMODO_FIRST_SERVER=https://periphery:8120
## Make all buttons just double-click, rather than the full confirmation dialog.
KOMODO_DISABLE_CONFIRM_DIALOG=false

## Rate Komodo polls your servers for
## status / container status / system stats / alerting.
## Options: 1-sec, 5-sec, 15-sec, 1-min, 5-min.
## Default: 15-sec
KOMODO_MONITORING_INTERVAL="15-sec"
## Rate Komodo polls Resources for updates,
## like outdated commit hash.
## Options: 1-min, 5-min, 15-min, 30-min, 1-hr.
## Default: 5-min
KOMODO_RESOURCE_POLL_INTERVAL="5-min"

## Used to auth incoming webhooks. Alt: KOMODO_WEBHOOK_SECRET_FILE
KOMODO_WEBHOOK_SECRET=a_random_secret
## Used to generate jwt. Alt: KOMODO_JWT_SECRET_FILE
KOMODO_JWT_SECRET=a_random_jwt_secret

## Enable login with username + password.
KOMODO_LOCAL_AUTH=true
## Disable new user signups.
KOMODO_DISABLE_USER_REGISTRATION=false
```

```
## All new logins are auto enabled
KOMODO_ENABLE_NEW_USERS=false
## Disable non-admins from creating new resources.
KOMODO_DISABLE_NON_ADMIN_CREATE=false
## Allows all users to have Read level access to all resources.
KOMODO_TRANSPARENT_MODE=false

## Time to live for jwt tokens.
## Options: 1-hr, 12-hr, 1-day, 3-day, 1-wk, 2-wk
KOMODO_JWT_TTL="1-day"

## OIDC Login
KOMODO_OIDC_ENABLED=false
## Must reachable from Komodo Core container
# KOMODO_OIDC_PROVIDER=https://oidc.provider.internal/application/o/komodo
## Change the host to one reachable be reachable by users (optional if it is the same as above).
## DO NOT include the `path` part of the URL.
# KOMODO_OIDC_REDIRECT_HOST=https://oidc.provider.external
## Your client credentials
# KOMODO_OIDC_CLIENT_ID= # Alt: KOMODO_OIDC_CLIENT_ID_FILE
# KOMODO_OIDC_CLIENT_SECRET= # Alt: KOMODO_OIDC_CLIENT_SECRET_FILE
## Make usernames the full email.
# KOMODO_OIDC_USE_FULL_EMAIL=true
## Add additional trusted audiences for token claims verification.
## Supports comma separated list, and passing with _FILE (for compose secrets).
# KOMODO_OIDC_ADDITIONAL_AUDIENCES=abc,123 # Alt: KOMODO_OIDC_ADDITIONAL_AUDIENCES_FILE

## Github Oauth
KOMODO_GITHUB_OAUTH_ENABLED=false
# KOMODO_GITHUB_OAUTH_ID= # Alt: KOMODO_GITHUB_OAUTH_ID_FILE
# KOMODO_GITHUB_OAUTH_SECRET= # Alt: KOMODO_GITHUB_OAUTH_SECRET_FILE

## Google Oauth
KOMODO_GOOGLE_OAUTH_ENABLED=false
# KOMODO_GOOGLE_OAUTH_ID= # Alt: KOMODO_GOOGLE_OAUTH_ID_FILE
# KOMODO_GOOGLE_OAUTH_SECRET= # Alt: KOMODO_GOOGLE_OAUTH_SECRET_FILE

## Aws - Used to launch Builder instances and ServerTemplate instances.
KOMODO_AWS_ACCESS_KEY_ID= # Alt: KOMODO_AWS_ACCESS_KEY_ID_FILE
KOMODO_AWS_SECRET_ACCESS_KEY= # Alt: KOMODO_AWS_SECRET_ACCESS_KEY_FILE
```

```
## Hetzner - Used to launch ServerTemplate instances
## Hetzner Builder not supported due to Hetzner pay-by-the-hour pricing model
KOMODO_HETZNER_TOKEN= # Alt: KOMODO_HETZNER_TOKEN_FILE

#=------------------------------=#
#= Komodo Periphery Environment =#
#=------------------------------=#

## Full variable list + descriptions are available here:
## □□ https://github.com/mbecker20/komodo/blob/main/config/periphery.config.toml □□

## Periphery passkeys must include KOMODO_PASSKEY to authenticate.
PERIPHERY_PASSKEYS=${KOMODO_PASSKEY}

## Specify the root directory used by Periphery agent.
PERIPHERY_ROOT_DIRECTORY=/etc/komodo

## Enable SSL using self signed certificates.
## Connect to Periphery at https://address:8120.
PERIPHERY_SSL_ENABLED=true

## If the disk size is overreporting, can use one of these to
## whitelist / blacklist the disks to filter them, whichever is easier.
## Accepts comma separated list of paths.
## Usually whitelisting just /etc/hostname gives correct size.
PERIPHERY_INCLUDE_DISK_MOUNTS=/etc/hostname
# PERIPHERY_EXCLUDE_DISK_MOUNTS=/snap,/etc/repos
```

Komodo - Advanced Configuration OIDC / Oauth2

Untuk mengaktifkan login OAuth2, Anda harus membuat klien pada penyedia OAuth masing-masing, misalnya Github atau Google .

Komodo juga mendukung penyedia Oauth2 yang dihosting sendiri seperti Authentik atau Gitea .

- Komodo menggunakan `web application` alur login.
- Uri pengalihannya adalah:
  - `<KOMODO_HOST>/auth/github/callback` untuk Github.
  - `<KOMODO_HOST>/auth/google/callback` untuk Google.
  - `<KOMODO_HOST>/auth/oidc/callback` untuk OIDC.

Jika Anda lebih suka merahasiakan informasi sensitif dari variabel lingkungan, Anda dapat secara opsional menulis file konfigurasi di host Anda, dan memasangnya `/config/config.toml` di kontainer inti Komodo.

!! INFO !!
Konfigurasi masih dapat diteruskan dalam variabel lingkungan, dan akan diutamakan daripada yang diteruskan dalam berkas.

Download ke `./komodo/core.config.toml`:

```
wget -P komodo https://raw.githubusercontent.com/moghtech/komodo/main/config/core.config.toml
```

Config File

```
############################
# □□ KOMODO CORE CONFIG □□ #
############################

## This is the offical "Default" config file for Komodo Core.
## It serves as documentation for the meaning of the fields.
## It is located at `https://github.com/mbecker20/komodo/blob/main/config/core.config.toml`.
```

```
## All fields with a "Default" provided are optional. If they are
## left out of the file, the "Default" value will be used.

## This file is bundled into the official image, `ghcr.io/mbecker20/komodo`,
## as the default config at `/config/config.toml`.
## Komodo can start with no external config file mounted.

## There is usually no need to create this file on your host.
## Most fields can instead be configured using environment variables.
## Environment variables will override values set in this file.

## This will be the document title on the web page.
## Env: KOMODO_TITLE
## Default: 'Komodo'
title = "Komodo"

## This should be the url used to access Komodo in browser, potentially behind DNS.
## Eg https://komodo.example.com or http://12.34.56.78:9120. This should match the address configured in your Oauth app.
## Env: KOMODO_HOST
## Required, no default.
host = "https://demo.komo.do"

## The port the core system will run on.
## Env: KOMODO_PORT
## Default: 9120
port = 9120

## This is the token used to authenticate core requests to periphery.
## Ensure this matches a passkey in the connected periphery configs.
## If the periphery servers don't have passkeys configured, this doesn't need to be changed.
## Env: KOMODO_PASSKEY or KOMODO_PASSKEY_FILE
## Required, no default
passkey = "a_random_passkey"

## Ensure a server with this address exists on Core
## upon first startup. Example: `https://periphery:8120`
## Env: KOMODO_FIRST_SERVER
## Optional, no default.
first_server = ""
```

```
## Disables write support on resources in the UI.
## This protects users that that would normally have write priviledges during their UI usage,
## when they intend to fully rely on ResourceSyncs to manage config.
## Env: KOMODO_UI_WRITE_DISABLED
## Default: false
ui_write_disabled = false

## Disables the confirm dialogs on all actions. All buttons will now be double-click.
## Useful when only having http connection to core, as UI quick-copy button won't work.
## Env: KOMODO_DISABLE_CONFIRM_DIALOG
## Default: false
disable_confirm_dialog = false

## Configure the directory for sync files (inside the container).
## There shouldn't be a need to change this, just mount a volume.
## Env: KOMODO_SYNC_DIRECTORY
## Default: /syncs
sync_directory = "/syncs"

## Configure the repo directory (inside the container).
## There shouldn't be a need to change this, just mount a volume.
## Env: KOMODO_REPO_DIRECTORY
## Default: /repo-cache
repo_directory = "/repo-cache"

## Configure the action directory (inside the container).
## There shouldn't be a need to change this, or even mount a volume.
## Env: KOMODO_ACTION_DIRECTORY
## Default: /action-cache
action_directory = "/action-cache"

################
# AUTH / LOGIN #
################

## Allow user login with a username / password.
## The password will be hashed and stored in the db for login comparison.
##
## NOTE:
```

Komodo has no API to recover account logins, but if this happens you can doctor the database using Mongo Compass.
Create a new Komodo user (Sign Up button), login to the database with Compass, note down your old users username and _id.
Then delete the old user, and update the new user to have the same username and _id.
Make sure to set `enabled: true` and maybe `admin: true` on the new user as well, while using Compass.
##
Env: KOMODO_LOCAL_AUTH
Default: false
local_auth = false

Normally new users will be registered, but not enabled until an Admin enables them.
With `disable_user_registration = true`, only the first user to log in will registered as a user.
Env: KOMODO_DISABLE_USER_REGISTRATION
Default: false
disable_user_registration = false

New users will be automatically enabled when they sign up.
Otherwise, new users will be disabled on first login.
The first user to login will always be enabled on creation.
Env: KOMODO_ENABLE_NEW_USERS
Default: false
enable_new_users = false

Allows all users to have Read level access to all resources.
Env: KOMODO_TRANSPARENT_MODE
Default: false
transparent_mode = false

Normally all enabled users can create resources.
If `disable_non_admin_create = true`, only admin users can create resources.
Env: KOMODO_DISABLE_NON_ADMIN_CREATE
Default: false
disable_non_admin_create = false

Optionally provide a specific jwt secret.
Passing nothing or an empty string will cause one to be generated on every startup.
This means users will have to log in again if Komodo restarts.
Env: KOMODO_JWT_SECRET or KOMODO_JWT_SECRET_FILE
Default: empty string, meaning a random secret will be generated at startup.

```
jwt_secret = ""

## Specify how long a user can stay logged in before they have to log in again.
## All jwts are invalidated on application restart unless `jwt_secret` is set.
## Env: KOMODO_JWT_TTL
## Options: 1-hr, 12-hr, 1-day, 3-day, 1-wk, 2-wk, 30-day
## Default: 1-day.
jwt_ttl = "1-day"

#############
# OIDC Auth #
#############

## Enable logins with configured OIDC provider.
## Env: KOMODO_OIDC_ENABLED
## Default: false
oidc_enabled = false

## Give the provider address.
##
## The path, ie /application/o/komodo for Authentik,
## is provider and configuration specific.
##
## Note. this address must be reachable from Komodo Core container.
##
## Env: KOMODO_OIDC_PROVIDER
## Optional, no default.
oidc_provider = "https://oidc.provider.internal/application/o/komodo"

## Configure OIDC user redirect host.
##
## This is the host address users are redirected to in their browser,
## and may be different from `oidc_provider` host depending on your networking.
## If not provided (or empty string ""), the `oidc_provider` will be used.
##
## Note. DO NOT include the `path` part of the URL.
## Example: `https://oidc.provider.external`
##
## Env: KOMODO_OIDC_REDIRECT_HOST
## Optional, no default.
```

```
oidc_redirect_host = ""

## Give the OIDC Client ID.
## Env: KOMODO_OIDC_CLIENT_ID or KOMODO_OIDC_CLIENT_ID_FILE
oidc_client_id = ""

## Give the OIDC Client Secret.
## Env: KOMODO_OIDC_CLIENT_SECRET or KOMODO_OIDC_CLIENT_SECRET_FILE
oidc_client_secret = ""

## If true, use the full email for usernames.
## Otherwise, the @address will be stripped,
## making usernames more concise.
## Env: KOMODO_OIDC_USE_FULL_EMAIL
## Default: false.
oidc_use_full_email = false

## Some providers attach other audiences in addition to the client_id.
## If you have this issue, `Invalid audiences: `...` is not a trusted audience"`,
## you can add the audience `...` to the list here (assuming it should be trusted).
## Env: KOMODO_OIDC_ADDITIONAL_AUDIENCES or KOMODO_OIDC_ADDITIONAL_AUDIENCES_FILE
## Default: empty
oidc_additional_audiences = []

#########
# OAUTH #
#########

## Google

## Env: KOMODO_GOOGLE_OAUTH_ENABLED
## Default: false
google_oauth.enabled = false

## Env: KOMODO_GOOGLE_OAUTH_ID or KOMODO_GOOGLE_OAUTH_ID_FILE
## Required if google_oauth is enabled.
google_oauth.id = ""

## Env: KOMODO_GOOGLE_OAUTH_SECRET or KOMODO_GOOGLE_OAUTH_SECRET_FILE
## Required if google_oauth is enabled.
```

```
google_oauth.secret = ""

## Github

## Env: KOMODO_GITHUB_OAUTH_ENABLED
## Default: false
github_oauth.enabled = false

## Env: KOMODO_GITHUB_OAUTH_ID or KOMODO_GITHUB_OAUTH_ID_FILE
## Required if github_oauth is enabled.
github_oauth.id = ""

## Env: KOMODO_GITHUB_OAUTH_SECRET or KOMODO_GITHUB_OAUTH_SECRET_FILE
## Required if github_oauth is enabled.
github_oauth.secret = ""

############
# Security #
############

## Enable HTTPS server using the given key and cert.
## Env: KOMODO_SSL_ENABLED
## Default: false
ssl_enabled = false

## Path to the ssl key.
## Env: KOMODO_SSL_KEY_FILE
## Default: /config/ssl/key.pem
ssl_key_file = "/config/ssl/key.pem"

## Path to the ssl cert.
## Env: KOMODO_SSL_CERT_FILE
## Default: /config/ssl/cert.pem
ssl_cert_file = "/config/ssl/cert.pem"

############
# DATABASE #
############

## Configure the database connection in one of the following ways:
```

```
## Pass a full Mongo URI to the database.
## Example: mongodb://username:password@localhost:27017
## Env: KOMODO_DATABASE_URI or KOMODO_DATABASE_URI_FILE
## Optional, can usually use `address`, `username`, `password` instead.
database.uri = ""

## ==== * OR * ==== ##

# Construct the address as mongodb://{username}:{password}@{address}
## Env: KOMODO_DATABASE_ADDRESS
database.address = "localhost:27017"
## Env: KOMODO_DATABASE_USERNAME or KOMODO_DATABASE_USERNAME_FILE
database.username = ""
## Env: KOMODO_DATABASE_PASSWORD or KOMODO_DATABASE_PASSWORD_FILE
database.password = ""

## ==== other ====

## Komodo will create its collections under this database name.
## The only reason to change this is if multiple Komodo Cores share the same db.
## Env: KOMODO_DATABASE_DB_NAME
## Default: komodo.
database.db_name = "komodo"

## This is the assigned app_name of the mongo client.
## The only reason to change this is if multiple Komodo Cores share the same db.
## Env: KOMODO_DATABASE_APP_NAME
## Default: komodo_core.
database.app_name = "komodo_core"

############
# WEBHOOKS #
############

## This token must be given to git provider during repo webhook config.
## The secret configured on the git provider side must match the secret configured here.
## If not provided,
## Env: KOMODO_WEBHOOK_SECRET or KOMODO_WEBHOOK_SECRET_FILE
## Optional, no default.
```

```
webhook_secret = "a_random_webhook_secret"

## An alternate base url that is used to recieve git webhook requests.
## If empty or not specified, will use 'host' address as base.
## This is useful if Komodo is on an internal network, but can have a
## proxy just allowing through the webhook listener api using NGINX.
## Env: KOMODO_WEBHOOK_BASE_URL
## Default: empty (none)
webhook_base_url = ""

## Configure Github webhook app. Enables webhook management apis.
## <INSERT LINK TO GUIDE>
## Env: KOMODO_GITHUB_WEBHOOK_APP_APP_ID or KOMODO_GITHUB_WEBHOOK_APP_APP_ID_FILE
# github_webhook_app.app_id = 1234455 # Find on the app page.
## Env:
##   - KOMODO_GITHUB_WEBHOOK_APP_INSTALLATIONS_IDS or
KOMODO_GITHUB_WEBHOOK_APP_INSTALLATIONS_IDS_FILE
##   - KOMODO_GITHUB_WEBHOOK_APP_INSTALLATIONS_NAMESPACES
# github_webhook_app.installations = [
#   ## Find the id after installing the app to user / organization. "namespace" is the username / organization
name.
#   { id = 1234, namespace = "mbecker20" }
# ]

## The path to Github webhook app private key. <INSERT LINK TO GUIDE>
## This is defaulted to `/github/private-key.pem`, and doesn't need to be changed if running core in Docker.
## Just mount the private key pem file on the host to `/github/private-key.pem` in the container.
## Eg. `/your/path/to/key.pem : /github/private-key.pem`
## Env: KOMODO_GITHUB_WEBHOOK_APP_PK_PATH
# github_webhook_app.pk_path = "/path/to/pk.pem"

############
# LOGGING #
############

## Specify the logging verbosity
## Env: KOMODO_LOGGING_LEVEL
## Options: off, error, warn, info, debug, trace
## Default: info
logging.level = "info"
```

```
## Specify the logging format for stdout / stderr.
## Env: KOMODO_LOGGING_STDIO
## Options: standard, json, none
## Default: standard
logging.stdio = "standard"

## Optionally specify a opentelemetry otlp endpoint to send traces to.
## Example: http://localhost:4317
## Env: KOMODO_LOGGING_OTLP_ENDPOINT
logging.otlp_endpoint = ""

## Set the opentelemetry service name.
## This will be attached to the telemetry Komodo will send.
## Env: KOMODO_LOGGING_OPENTELEMETRY_SERVICE_NAME
## Default: "Komodo"
logging.opentelemetry_service_name = "Komodo"

############
# PRUNING #
############

## The number of days to keep historical system stats around, or 0 to disable pruning.
## Stats older that are than this number of days are deleted on a daily cycle.
## Env: KOMODO_KEEP_STATS_FOR_DAYS
## Default: 14
keep_stats_for_days = 14

## The number of days to keep alerts around, or 0 to disable pruning.
## Alerts older that are than this number of days are deleted on a daily cycle.
## Env: KOMODO_KEEP_ALERTS_FOR_DAYS
## Default: 14
keep_alerts_for_days = 14

###################
# POLL INTERVALS #
###################

## Controls the rate at which servers are polled for health, system stats, and container status.
## This affects network usage, and the size of the stats stored in mongo.
```

```
## Env: KOMODO_MONITORING_INTERVAL
## Options: 1-sec, 5-sec, 15-sec, 30-sec, 1-min, 2-min, 5-min, 15-min
## Default: 15-sec
monitoring_interval = "15-sec"

## Interval at which to poll Resources for any updates / automated actions.
## Env: KOMODO_RESOURCE_POLL_INTERVAL
## Options: `15-sec`, `1-min`, `5-min`, `15-min`, `1-hr`.
## Default: 5-min
resource_poll_interval = "5-min"

####################
# CLOUD PROVIDERS #
####################

## Komodo can build images by deploying AWS EC2 instances,
## running the build, and afterwards destroying the instance.

## Additionally, Komodo can deploy cloud VPS on AWS EC2 and Hetzner.
## Use the Template resource to configure launch preferences.
## Hetzner is not supported for builds as their pricing model is by the hour,
## while AWS is by the minute. This is very important for builds.

## Provide AWS api keys for ephemeral builders / server launch
## Env: KOMODO_AWS_ACCESS_KEY_ID or KOMODO_AWS_ACCESS_KEY_ID_FILE
aws.access_key_id = ""
## Env: KOMODO_AWS_SECRET_ACCESS_KEY or KOMODO_AWS_SECRET_ACCESS_KEY_FILE
aws.secret_access_key = ""

## Provide Hetzner api token for server launch
## Env: KOMODO_HETZNER_TOKEN or KOMODO_HETZNER_TOKEN_FILE
hetzner.token = ""

#################
# GIT PROVIDERS #
#################

## These will be available to attach to Builds, Repos, Stacks, and Syncs.
## They allow these Resources to clone private repositories.
## They cannot be configured on the environment.
```

```
## configure git providers
# [[git_provider]]
# domain = "github.com"
# accounts = [
# □{ username = "mbecker20", token = "access_token_for_account" },
# □{ username = "moghtech", token = "access_token_for_other_account" },
# ]

# [[git_provider]]
# domain = "git.mogh.tech" # use a custom provider, like self-hosted gitea
# accounts = [
# □{ username = "mbecker20", token = "access_token_for_account" },
# ]

# [[git_provider]]
# domain = "localhost:8000" # use a custom provider, like self-hosted gitea
# https = false # use http://localhost:8000 as base-url for clone
# accounts = [
# □{ username = "mbecker20", token = "access_token_for_account" },
# ]

########################
# REGISTRY PROVIDERS #
########################

## These will be available to attach to Builds and Stacks.
## They allow these Resources to pull private images.
## They cannot be configured on the environment.

## configure docker registries
# [[docker_registry]]
# domain = "docker.io"
# accounts = [
# □{ username = "mbecker2020", token = "access_token_for_account" }
# ]
# organizations = ["DockerhubOrganization"]

# [[docker_registry]]
# domain = "git.mogh.tech" # use a custom provider, like self-hosted gitea
```

```
# accounts = [
# □{ username = "mbecker20", token = "access_token_for_account" },
# ]
# organizations = ["Mogh"] # These become available in the UI

###########
# SECRETS #
###########

## Provide Core based secrets.
## These will be available to interpolate into your Deployment / Stack environments,
## and will be hidden in the UI and logs.
## These are available to use on any Periphery (Server),
## but you can also limit access more by placing them in a single Periphery's config file instead.
## These cannot be configured in the Komodo Core environment, they must be passed in the file.

# [secrets]
# SECRET_1 = "value_1"
# SECRET_2 = "value_2"
```