

NGINX Docker

NGINX . . .

NGINX C 1.22.0 [github](#), . . . : PKG-OSS.
Makefile.module-*. - Dockerfile .

:

1. (conf.d — , modules-enabled -):

```
sudo mkdir /etc/hydra/nginx-custom && sudo mkdir /etc/hydra/nginx-custom/conf.d /etc/hydra/nginx-custom/modules-enabled
```

2. nginx:

/etc/hydra/nginx-custom/conf.d/nginx.conf

```
resolver 127.0.0.11 valid=10s;
```

3. - :

/etc/hydra/nginx-custom/nginx.conf

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    #gzip on;

    include /etc/nginx/conf.d/*.conf;
}
```

4. docker-compose.yml. ENABLED_MODULES .

/etc/hydra/nginx-custom/docker-compose.yml

```
version: "3.8"
services:
  nginx-custom:
    image: nginx-custom:v1
    build:
      context: .
      args:
        #
        ENABLED_MODULES: "ndk lua" ###
    container_name: nginx-custom
    restart: unless-stopped
    volumes:
      - ./conf.d:/etc/nginx/conf.d:ro
      - /var/log/hydra/nginx-custom:/var/log/nginx
      - ./modules-enabled:/etc/nginx/modules-enabled:ro
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
    networks:
      - default
      - nginx-custom
    ports:
      - 80:80
      - 443:443
    logging:
      driver: "syslog"
      options:
        tag: "hydra/nginx-custom/{{.Name}}"
    #
    ##, -
    ###command: tail -F /remove-after-moving.txt
networks:
  nginx-custom:
    internal: true
    name: nginx-custom
```

5. . ngx_http_lua_module ndk_http_module:

/etc/hydra/nginx-custom/modules-enabled/modules.conf

```
load_module /etc/nginx/modules/ndk_http_module.so;
load_module /etc/nginx/modules/ngx_http_lua_module.so;
```

6. Dockerfile ():

/etc/hydra/nginx-custom/Dockerfile

```
FROM nginx:mainline as builder

ARG ENABLED_MODULES

RUN set -ex \
    && if [ "$ENABLED_MODULES" = "" ]; then \
        echo "No additional modules enabled, exiting"; \
        exit 1; \
    fi

COPY ./ /modules/

RUN set -ex \
    && apt update \
    && apt install -y --no-install-suggests --no-install-recommends \
        patch make wget mercurial devscripts debhelper dpkg-dev \
        quilt lsb-release build-essential libxml2-utils xsltproc \
        equivs git g++ \
```

```

&& hg clone -r ${NGINX_VERSION}-${PKG_RELEASE%%~*} https://hg.nginx.org/pkg-oss/ \
&& cd pkg-oss \
&& mkdir /tmp/packages \
&& for module in $ENABLED_MODULES; do \
    echo "Building $module for nginx-$NGINX_VERSION"; \
    if [ -d /modules/$module ]; then \
        echo "Building $module from user-supplied sources"; \
        # check if module sources file is there and not empty
        if [ ! -s /modules/$module/source ]; then \
            echo "No source file for $module in modules/$module/source, exiting"; \
            exit 1; \
        fi; \
        # some modules require build dependencies
        if [ -f /modules/$module/build-deps ]; then \
            echo "Installing $module build dependencies"; \
            apt update && apt install -y --no-install-suggests --no-install-recommends $(cat /modules/$module/build-deps | xargs); \
        fi; \
        # if a module has a build dependency that is not in a distro, provide a
        # shell script to fetch/build/install those
        # note that shared libraries produced as a result of this script will
        # not be copied from the builder image to the main one so build static
        if [ -x /modules/$module/prebuild ]; then \
            echo "Running prebuild script for $module"; \
            /modules/$module/prebuild; \
        fi; \
        /pkg-oss/build_module.sh -v $NGINX_VERSION -f -y -o /tmp/packages -n $module $(cat /modules/$module/source); \
        BUILT_MODULES="$BUILT_MODULES $(echo $module | tr '[A-Z]' '[a-z]' | tr -d '[_\-. \t ]')"; \
    elif make -C /pkg-oss/debian list | grep -P "^$module\s+\d" > /dev/null; then \
        echo "Building $module from pkg-oss sources"; \
        cd /pkg-oss/debian; \
        make rules-module-$module BASE_VERSION=$NGINX_VERSION NGINX_VERSION=$NGINX_VERSION; \
        mk-build-deps --install --tool="apt-get -o Debug::pkgProblemResolver=yes --no-install-recommends --yes" debuild-module-$module/nginx-$NGINX_VERSION/debian/control; \
        make module-$module BASE_VERSION=$NGINX_VERSION NGINX_VERSION=$NGINX_VERSION; \
        find ../../ -maxdepth 1 -mindepth 1 -type f -name "*.deb" -exec mv -v {} /tmp/packages/ \;; \
        BUILT_MODULES="$BUILT_MODULES $module"; \
    else \
        echo "Don't know how to build $module module, exiting"; \
        exit 1; \
    fi; \
done \
&& echo "BUILT_MODULES=\"$BUILT_MODULES\" > /tmp/packages/modules.env

FROM nginx:mainline
COPY --from=builder /tmp/packages /tmp/packages
RUN set -ex \
    && apt update \
    && . /tmp/packages/modules.env \
    && for module in $BUILT_MODULES; do \
        apt install --no-install-suggests --no-install-recommends -y /tmp/packages/nginx-module-${module}_${NGINX_VERSION}*.deb; \
    done \
    && rm -rf /tmp/packages \
    && rm -rf /var/lib/apt/lists/

```

7. :

```

cd /etc/hydra/nginx-custom/
sudo docker-compose build

```

8. -:

```
cd /etc/hydra/nginx-custom/  
sudo docker-compose up -d
```

NGINX : /var/log/hydra/nginx-custom, :

```
sudo docker logs nginx-custom
```

9. nginx- :

```
sudo docker exec nginx-custom nginx -s reload
```