

Nginx Unknown directive "auth_request"

Asked 4 years, 5 months ago Modified 1 year, 7 months ago Viewed 2k times



How to include this auth_request module to my custom nginx buildpack?

1

This is my configuration:



```
daemon off;
# Heroku dynos have at least 4 cores.
worker_processes <%= ENV['NGINX_WORKERS'] || 4 %>;

events {
    use epoll;
    accept_mutex on;
    worker_connections <%= ENV['NGINX_WORKER_CONNECTIONS'] || 1024 %>;
}

http {
    gzip on;
    gzip_comp_level 2;
    gzip_min_length 512;

    server_tokens off;

    log_format l2met 'measure#nginx.service=$request_time
request_id=$http_x_request_id';
    access_log <%= ENV['NGINX_ACCESS_LOG_PATH'] || 'logs/nginx/access.log' %>
l2met;
    error_log <%= ENV['NGINX_ERROR_LOG_PATH'] || 'logs/nginx/error.log' %>;

    include mime.types;
    default_type application/octet-stream;
    sendfile on;

    # Must read the body in 5 seconds.
    client_body_timeout 5;

    upstream app_server {
        server unix:/tmp/nginx.socket fail_timeout=0;
    }

    server {
        listen <%= ENV["PORT"] %>;
        server_name _;
        keepalive_timeout 5;

        location / {
            auth_request /_oauth2_token_introspection;
            proxy_pass http://my_backend;
        }

        location /_oauth2_send_request {
            internal;
```

```

        proxy_method      POST;
        proxy_set_header  Authorization "Bearer SecretForOAuthServer";
        proxy_set_header  Content-Type "application/x-www-form-urlencoded";
        proxy_set_body    "token=$http_apikey&token_hint=access_token";
        proxy_pass         https://idp.example.com/oauth/token/introspect;
    }

}
}

```

I have added the `--with-http_auth_request_module` in my `build_nginx` file at `/configure`:

```

#!/bin/bash
# Build NGINX and modules for Heroku.
# This script is designed to run in a Heroku Stack Docker
# image. More information on the Heroku Stack can be found
# at https://devcenter.heroku.com/articles/stack

NGINX_VERSION=${NGINX_VERSION-1.18.0}
PCRE_VERSION=${PCRE_VERSION-8.44}
HEADERS_MORE_VERSION=${HEADERS_MORE_VERSION-0.33}
ZLIB_VERSION=${ZLIB_VERSION-1.2.11}
UUID4_VERSION=${UUID4_VERSION-master}

nginx_tarball_url=https://nginx.org/download/nginx-${NGINX_VERSION}.tar.gz
pcre_tarball_url=https://ftp.pcre.org/pub/pcre/pcre-${PCRE_VERSION}.tar.gz
headers_more_nginx_module_url=https://github.com/openresty/headers-more-nginx-
module/archive/v${HEADERS_MORE_VERSION}.tar.gz
uuid4_url=https://github.com/cybozu/nginx-uuid4-
module/archive/${UUID4_VERSION}.tar.gz
zlib_url=http://zlib.net/zlib-${ZLIB_VERSION}.tar.gz

temp_dir=$(mktemp -d /tmp/nginx.XXXXXXXXXX)

cd $temp_dir
echo "Temp dir: $temp_dir"

echo "Downloading $nginx_tarball_url"
curl -L $nginx_tarball_url | tar xzv

echo "Downloading $pcre_tarball_url"
(cd nginx-${NGINX_VERSION} && curl -L $pcre_tarball_url | tar xzv )

echo "Downloading $headers_more_nginx_module_url"
(cd nginx-${NGINX_VERSION} && curl -L $headers_more_nginx_module_url | tar xzv
)

echo "Downloading $zlib_url"
(cd nginx-${NGINX_VERSION} && curl -L $zlib_url | tar xzv )

echo "Downloading $uuid4_url"
(cd nginx-${NGINX_VERSION} && curl -L $uuid4_url | tar xzv )

# This will build `nginx`
(
    cd nginx-${NGINX_VERSION}
    ./configure \
        --with-pcre=pcre-${PCRE_VERSION} \
        --with-zlib=zlib-${ZLIB_VERSION} \
        --with-http_gzip_static_module \

```

```

--with-http_realip_module \
--with-http_ssl_module \
--with-http_auth_request_module \
--prefix=/tmp/nginx \
--add-module=${temp_dir}/nginx-${NGINX_VERSION}/headers-more-nginx-
module-${HEADERS_MORE_VERSION} \
--add-module=${temp_dir}/nginx-${NGINX_VERSION}/nginx-uuid4-
module-${UUID4_VERSION}
make install
)

# This will build `nginx-debug`
(
  cd nginx-${NGINX_VERSION}
  ./configure \
    --with-debug \
    --with-pcre=pcre-${PCRE_VERSION} \
    --with-zlib=zlib-${ZLIB_VERSION} \
    --with-http_gzip_static_module \
    --with-http_auth_request_module \
    --with-http_realip_module \
    --with-http_ssl_module \
    --prefix=/tmp/nginx-debug \
    --add-module=${temp_dir}/nginx-${NGINX_VERSION}/headers-more-nginx-
module-${HEADERS_MORE_VERSION} \
    --add-module=${temp_dir}/nginx-${NGINX_VERSION}/nginx-uuid4-
module-${UUID4_VERSION}
  make install
)

release_dir=$(mktemp -d /tmp/nginx.XXXXXXXXXX)

cp /tmp/nginx/sbin/nginx $release_dir/nginx
cp /tmp/nginx-debug/sbin/nginx $release_dir/nginx-debug
cp /tmp/nginx/conf/mime.types $release_dir/mime.types
tar -zcvf /tmp/nginx-"${STACK}".tgz -C $release_dir .
cp /tmp/nginx-"${STACK}".tgz $1

```

Regardless the error still persists: `nginx: [emerg] unknown directive "auth_request"`

[ruby](#)
[nginx](#)
[heroku](#)
[buildpack](#)

Share Improve this question Follow

asked Nov 10, 2020 at 18:46



Gustavo

3,811 ● 7 ● 29 ● 44

2 Answers

Sorted by: Highest score (default)



1

I think this sounds like you might have nginx installed somewhere else, and its in your PATH? I had this same problem today and this was the issue.

Share Improve this answer Follow



0



[nginx buildpack](#) includes three `.tgz` archives (`nginx-heroku-18.tgz` , `nginx-heroku-20.tgz` and `nginx-heroku-22.tgz`) with compiled versions of nginx for 3 supported Heroku stacks. The buildpack doesn't download and compile nginx when buildpack is compiled, it just extracts nginx binaries from one of these `.tgz` archives.

So the script `scripts/build_nginx` that you have modified is not run when buildpack is compiled. You need to run `make` on your machine to run this script. `Makefile` included in the repo will run the script three times within proper Docker images for the three Heroku stacks and build new `.tgz` archives, that can be then pushed to your custom buildpack repo.

[Share](#) [Improve this answer](#) [Follow](#)

edited Aug 28, 2023 at 11:11

answered Aug 28, 2023 at 10:45



Jan Wrobel

7,119 ● 3 ● 42 ● 57

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

[ruby](#) [nginx](#) [heroku](#) [buildpack](#)

See similar questions with these tags.