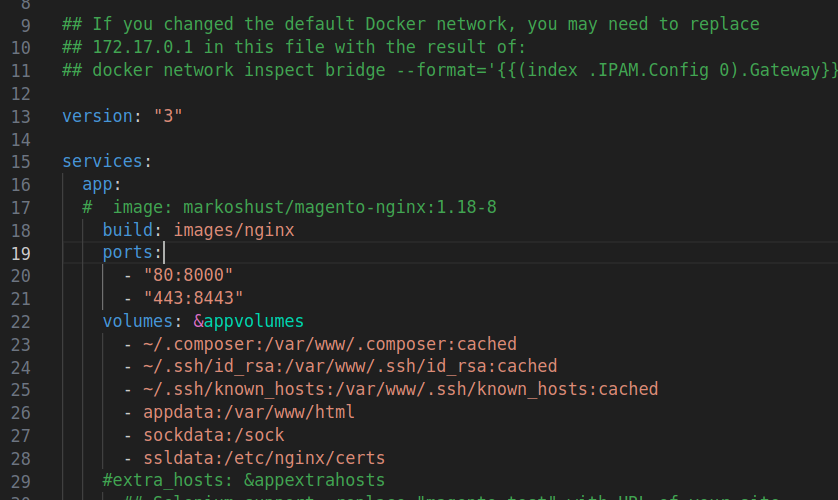
## How to setup Multi website on Nginx server in Docker in magento2

Step1 - first we need to change - compose.yaml file -



Changes according to line no 18

step2 -

Create a folder inside your project directory

inside

images/nginx

Create DockerFile

Sites/magento244p6/images/nginx/

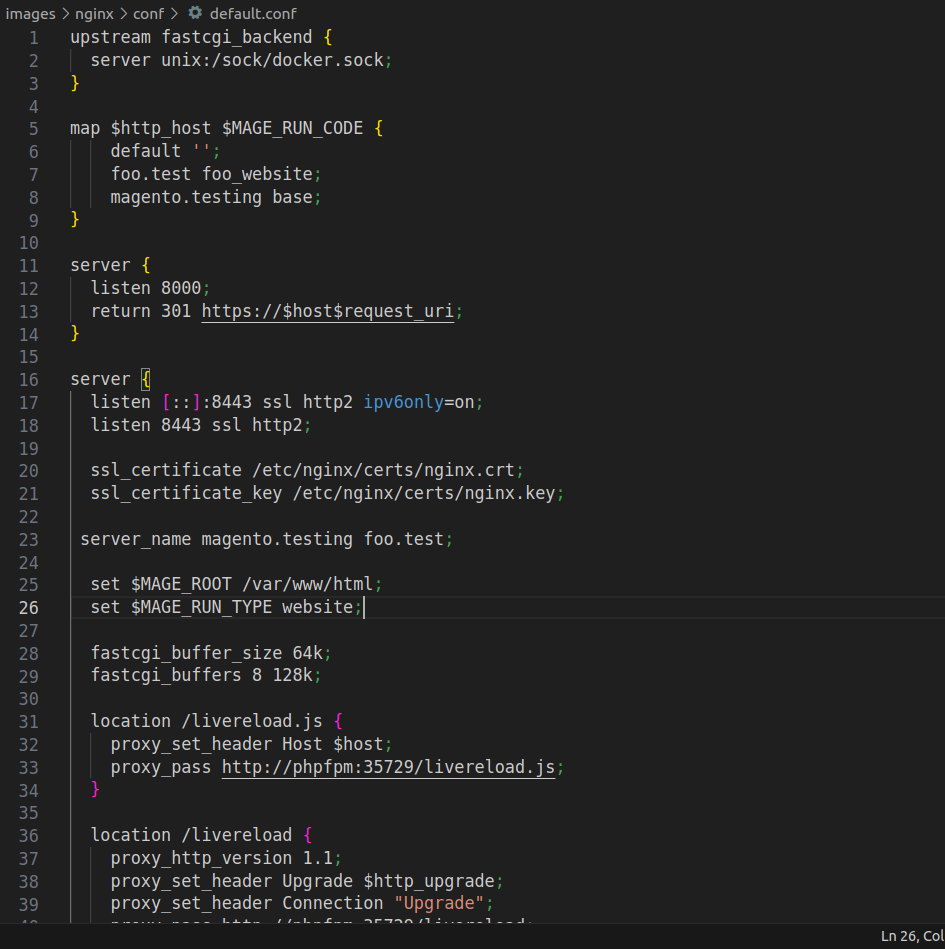
Dockerfile(filename)

| FROM markoshust/magento-nginx:1.18-8  COPY ./conf/**default**.conf /etc/nginx/conf.d/ |
| --- |

After this,

insame location like images/nginx

inside create one folder conf and inside create default.conf



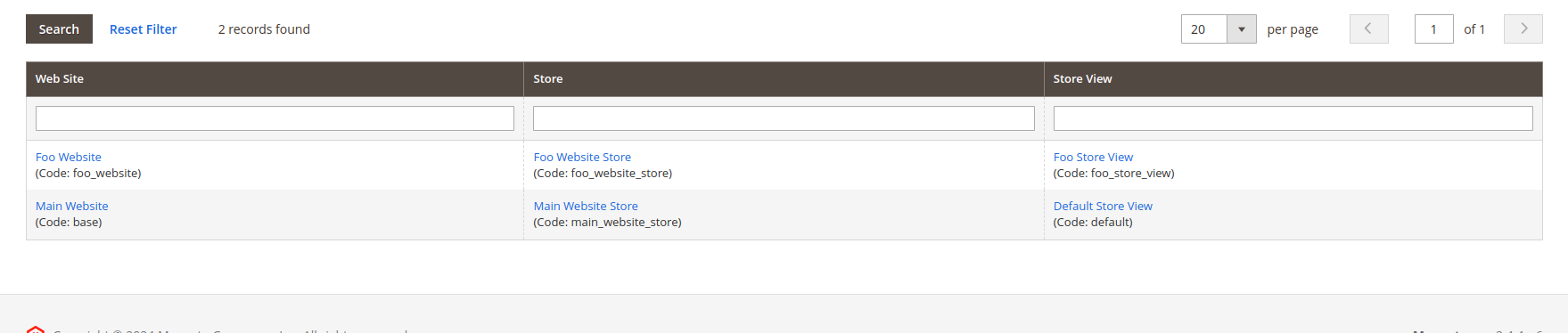
This one is modified file

In this file I have added something

Line no 5 to 9

In line no 7 first is given website\_url and second is website\_code

Like as I have givin foo.test foo\_website



And also I have added line no 23

And also I have added 26

Step3:-

I have create a nginx.conf file inside src/nginx.conf

If file is already exists then only paste the above code

| ## Example configuration: # upstream fastcgi\_backend { # # use tcp connection # # server 127.0.0.1:9000; # # or socket # server unix:/var/run/php/php7.4-fpm.sock; # } # server { # listen 80; # server\_name mage.dev; # set $MAGE\_ROOT /var/www/magento2; # set $MAGE\_DEBUG\_SHOW\_ARGS 0; # include /vagrant/magento2/nginx.conf.sample; # } # ## Optional override of deployment mode. We recommend you use the ## command 'bin/magento deploy:mode:set' to switch modes instead. ## ## set $MAGE\_MODE default; # or production or developer ## ## If you set MAGE\_MODE in server config, you must pass the variable into the ## PHP entry point blocks, which are indicated below. You can pass ## it in using: ## ## fastcgi\_param MAGE\_MODE $MAGE\_MODE; ## ## In production mode, you should uncomment the 'expires' directive in the /static/ location block  # Modules can be loaded only at the very beginning of the Nginx config file, please move the line below to the main config file # load\_module /etc/nginx/modules/ngx\_http\_image\_filter\_module.so;  root $MAGE\_ROOT/pub;  index index.php; autoindex **off**; charset UTF-8; error\_page 404 403 = /errors/404.php; #add\_header "X-UA-Compatible" "IE=Edge";   # Deny access to sensitive files location /.user.ini {  deny all; }  # PHP entry point for setup application location ~\* ^/setup($|/) {  root $MAGE\_ROOT;  location ~ ^/setup/index.php {  fastcgi\_pass fastcgi\_backend;   fastcgi\_param PHP\_FLAG "session.auto\_start=off \n suhosin.session.cryptua=off";  fastcgi\_param PHP\_VALUE "memory\_limit=756M \n max\_execution\_time=600";  fastcgi\_read\_timeout 600s;  fastcgi\_connect\_timeout 600s;   fastcgi\_index index.php;  fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  include fastcgi\_params;  }   location ~ ^/setup/(?!pub/). {  deny all;  }   location ~ ^/setup/pub/ {  add\_header X-Frame-Options "SAMEORIGIN";  } }  # PHP entry point for update application location ~\* ^/update($|/) {  root $MAGE\_ROOT;   location ~ ^/update/index.php {  fastcgi\_split\_path\_info ^(/update/index.php)(/.+)$;  fastcgi\_pass fastcgi\_backend;  fastcgi\_index index.php;  fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  fastcgi\_param PATH\_INFO $fastcgi\_path\_info;  include fastcgi\_params;  }   # Deny everything but index.php  location ~ ^/update/(?!pub/). {  deny all;  }   location ~ ^/update/pub/ {  add\_header X-Frame-Options "SAMEORIGIN";  } }  location / {  try\_files $uri $uri/ /index.php$is\_args$args; }  location /pub/ {  location ~ ^/pub/media/(downloadable|customer|import|custom\_options|theme\_customization/.\*\.xml) {  deny all;  }  alias $MAGE\_ROOT/pub/;  add\_header X-Frame-Options "SAMEORIGIN"; }  location /static/ {  # Uncomment the following line in production mode  # expires max;   # Remove signature of the static files that is used to overcome the browser cache  location ~ ^/static/version\d\*/ {  rewrite ^/static/version\d\*/(.\*)$ /static/$1 **last**;  }   location ~\* \.(ico|jpg|jpeg|png|gif|svg|svgz|webp|avif|avifs|js|css|eot|ttf|otf|woff|woff2|html|json|webmanifest)$ {  add\_header Cache-Control "public";  add\_header X-Frame-Options "SAMEORIGIN";  expires +1y;   if (!-f $request\_filename) {  rewrite ^/static/(version\d\*/)?(.\*)$ /static.php?resource=$2 **last**;  }  }  location ~\* \.(zip|gz|gzip|bz2|csv|xml)$ {  add\_header Cache-Control "no-store";  add\_header X-Frame-Options "SAMEORIGIN";  expires **off**;   if (!-f $request\_filename) {  rewrite ^/static/(version\d\*/)?(.\*)$ /static.php?resource=$2 **last**;  }  }  if (!-f $request\_filename) {  rewrite ^/static/(version\d\*/)?(.\*)$ /static.php?resource=$2 **last**;  }  add\_header X-Frame-Options "SAMEORIGIN"; }  location /media/ {  ## The following section allows to offload image resizing from Magento instance to the Nginx. ## Catalog image URL format should be set accordingly. ## See https://docs.magento.com/user-guide/configuration/general/web.html#url-options # location ~\* ^/media/catalog/.\* { # # # Replace placeholders and uncomment the line below to serve product images from public S3 # # See examples of S3 authentication at https://github.com/anomalizer/ngx\_aws\_auth # # resolver 8.8.8.8; # # proxy\_pass https://<bucket-name>.<region-name>.amazonaws.com; # # set $width "-"; # set $height "-"; # if ($arg\_width != '') { # set $width $arg\_width; # } # if ($arg\_height != '') { # set $height $arg\_height; # } # image\_filter resize $width $height; # image\_filter\_jpeg\_quality 90; # }   try\_files $uri $uri/ /get.php$is\_args$args;   location ~ ^/media/theme\_customization/.\*\.xml {  deny all;  }   location ~\* \.(ico|jpg|jpeg|png|gif|svg|svgz|webp|avif|avifs|js|css|eot|ttf|otf|woff|woff2)$ {  add\_header Cache-Control "public";  add\_header X-Frame-Options "SAMEORIGIN";  expires +1y;  try\_files $uri $uri/ /get.php$is\_args$args;  }  location ~\* \.(zip|gz|gzip|bz2|csv|xml)$ {  add\_header Cache-Control "no-store";  add\_header X-Frame-Options "SAMEORIGIN";  expires **off**;  try\_files $uri $uri/ /get.php$is\_args$args;  }  add\_header X-Frame-Options "SAMEORIGIN"; }  location /media/customer/ {  deny all; }  location /media/downloadable/ {  deny all; }  location /media/import/ {  deny all; }  location /media/custom\_options/ {  deny all; }  location /errors/ {  location ~\* \.xml$ {  deny all;  } }  # PHP entry point for main application location ~ ^/(index|get|static|errors/report|errors/404|errors/503|health\_check)\.php$ {  try\_files $uri =404;  fastcgi\_pass fastcgi\_backend;  fastcgi\_buffers 16 16k;  fastcgi\_buffer\_size 32k;   fastcgi\_param PHP\_FLAG "session.auto\_start=off \n suhosin.session.cryptua=off";  fastcgi\_param PHP\_VALUE "memory\_limit=756M \n max\_execution\_time=18000";  fastcgi\_read\_timeout 600s;  fastcgi\_connect\_timeout 600s;   fastcgi\_index index.php;  fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  fastcgi\_param MAGE\_RUN\_TYPE $MAGE\_RUN\_TYPE;  fastcgi\_param MAGE\_RUN\_CODE $MAGE\_RUN\_CODE;  include fastcgi\_params; }  gzip **on**; gzip\_disable "msie6";  gzip\_comp\_level 6; gzip\_min\_length 1100; gzip\_buffers 16 8k; gzip\_proxied any; gzip\_types  text/plain  text/css  text/js  text/xml  text/javascript  application/javascript  application/x-javascript  application/json  application/xml  application/xml+rss  image/svg+xml; gzip\_vary **on**;  # Banned locations (only reached if the earlier PHP entry point regexes don't match) location ~\* (\.php$|\.phtml$|\.htaccess$|\.git) {  deny all; } |
| --- |

In this file I have chnages only one thing

I have added

Note this is added only

# PHP entry point for main application

| fastcgi\_param MAGE\_RUN\_TYPE $MAGE\_RUN\_TYPE;  fastcgi\_param MAGE\_RUN\_CODE $MAGE\_RUN\_CODE; |
| --- |

After this cached this file compose.dev.yaml

version: "3"

| version: "3"  services:  app:  volumes: &appvolumes  ## Host mounts with performance penalty, only put what is necessary here  - ./src/app/code:/var/www/html/app/code:cached  - ./src/app/design:/var/www/html/app/design:cached  - ./src/app/etc:/var/www/html/app/etc:cached  - ./src/composer.json:/var/www/html/composer.json:cached  - ./src/composer.lock:/var/www/html/composer.lock:cached  - ./src/grunt-config.json.sample:/var/www/html/grunt-config.json:cached  - ./src/Gruntfile.js.sample:/var/www/html/Gruntfile.js:cached  - ./src/dev/tools/grunt/configs:/var/www/html/dev/tools/grunt/configs:cached  - ./src/nginx.conf:/var/www/html/nginx.conf:cached  - ./src/package.json.sample:/var/www/html/package.json:cached  - ./src/vendor/magento/module-catalog:/var/www/html/vendor/magento/module-catalog:cached  - ./src/vendor/magento/module-sales-rule:/var/www/html/vendor/magento/module-sales-rule:cached  #- ./src/auth.json:/var/www/html/auth.json:cached  #- ./src/m2-hotfixes:/var/www/html/m2-hotfixes:cached  #- ./src/patches:/var/www/html/patches:cached  #- ./src/var/log:/var/www/html/var/log:cached  #- ./src/var/report:/var/www/html/var/report:cached   phpfpm:  volumes: \*appvolumes   phpmyadmin:  image: linuxserver/phpmyadmin  env\_file: env/db.env  ports:  - "8080:80"  depends\_on:  - db |
| --- |

Here is the link :

<https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/multi-sites/ms-nginx.html?lang=en>

**Here is the link of video -**

<https://courses.m.academy/courses/set-up-magento-2-development-environment-docker/lectures/14780970>

Here is the github link:-