

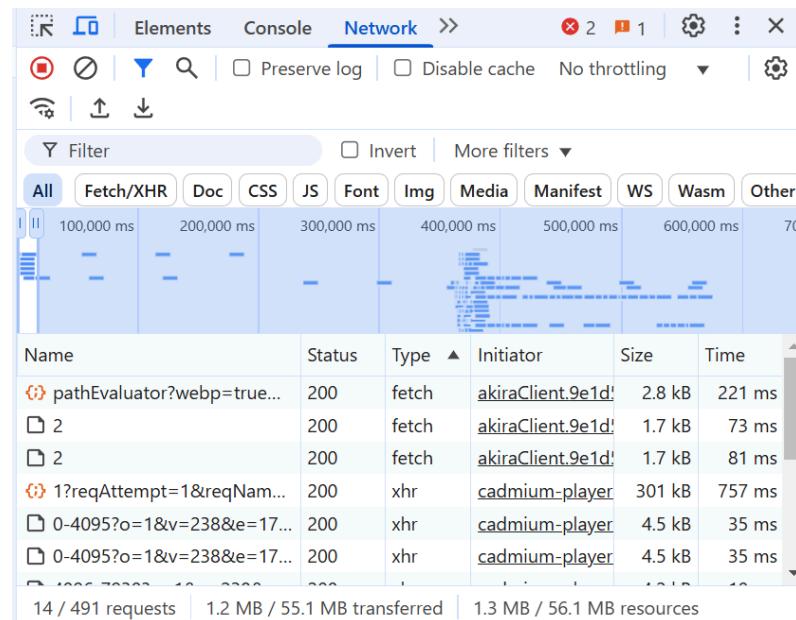
P4 - CDN & Docker

ex 1: Download Docker for the command line or for Desktop and install it.

```
PS C:\Users\isall\OneDrive\UNI\4_uni\P4_Vid3o> ffmpeg -version
ffmpeg version 2024-09-26-git-f43916e217-full_build-www.gyan.dev Copyright (c) 2000-2024 the FFmpeg developers
built with gcc 13.2.0 (Rev5, Built by MSYS2 project)
configuration: --enable-gpl --enable-version3 --enable-static --disable-w32threads --disable-autodetect --enable-fontconfig --enable-iconv --enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-lzma --enable-libsnappy --enable-zlib --enable-librist --enable-libssh --enable-libzmq --enable-avisynth --enable-libbluray --enable-libcaca --enable-sdl2 --enable-libaribbb24 --enable-libaribcaption --enable-libdavid --enable-libdavfs2 --enable-libopenjpeg --enable-libquirc --enable-libuavs3d --enable-libxevd --enable-libzvbi --enable-libqrencode --enable-librav1e --enable-libsvtav1 --enable-libvenc --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxevc --enable-libxvid --enable-libaom --enable-libjxl --enable-libvp8 --enable-medialibrary --enable-libass --enable-frei0r --enable-libfreetype --enable-libfribidi --enable-libharfbuzz --enable-liblensfun --enable-libvidstab --enable-libvmaf --enable-libzimg --enable-amf --enable-cuda-llm --enable-cuvid --enable-dxva2 --enable-d3d12va --enable-d3d11va --enable-fnvcdec --enable-libvpl --enable-nvdec --enable-nvenc --enable-vaapi --enable-libshaderc --enable-vulkan --enable-libplacebo --enable-opencl --enable-libcdio --enable-libgme --enable-libmodplug --enable-libopenpmt --enable-libopencore-amrwb --enable-libmp3lame --enable-libshine --enable-libtheora --enable-libtwolame --enable-libvo-amrwbenc --enable-libcodecs2 --enable-libilbc --enable-libgsm --enable-liblrc3 --enable-libopencore-amrnb --enable-libopus --enable-libspeex --enable-libvorbis --enable-ladspa --enable-libbs2b --enable-libflite --enable-libmysofa --enable-librubberband --enable-libsoxr --enable-chromaprint
libavutil      59. 40.100 / 59. 40.100
libavcodec    61. 20.100 / 61. 20.100
libavformat   61.  8.100 / 61.  8.100
libavdevice   61.  4.100 / 61.  4.100
libavfilter    10.  5.100 / 10.  5.100
libswscale     8.  4.100 /  8.  4.100
libswresample  5.  4.100 /  5.  4.100
libpostproc   58.  4.100 / 58.  4.100
PS C:\Users\isall\OneDrive\UNI\4_uni\P4_Vid3o>
```

ex4: We have connected into *Netflix*, a VOD platform. Then, we used the *F12* command to open *Developer Tools*, navigated to the *Network* tab, and examined the *Type* column.

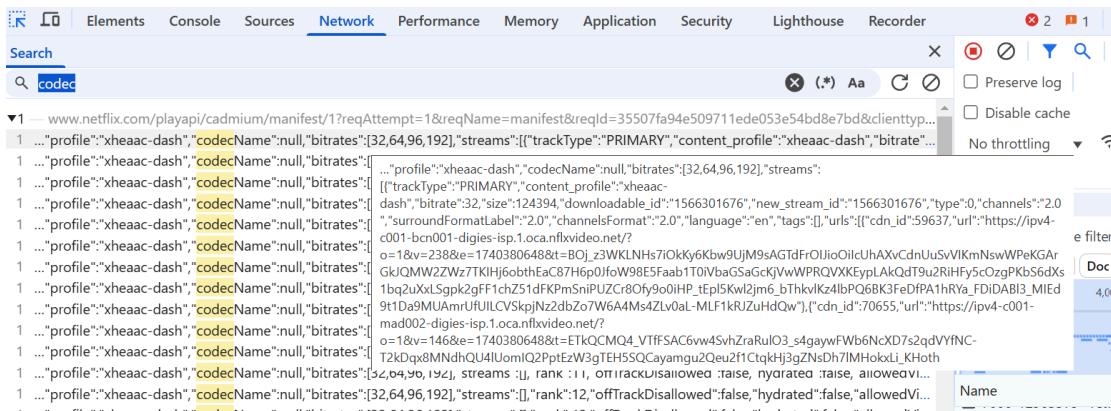
Then we have click play “*The Fast and the Furious*” and observe the following:



We see that the video is not downloaded as a single file but in small chunks. This indicates that Netflix uses **DASH**, a system that adjusts video quality based on internet speed to prevent buffering or interruptions during playback.

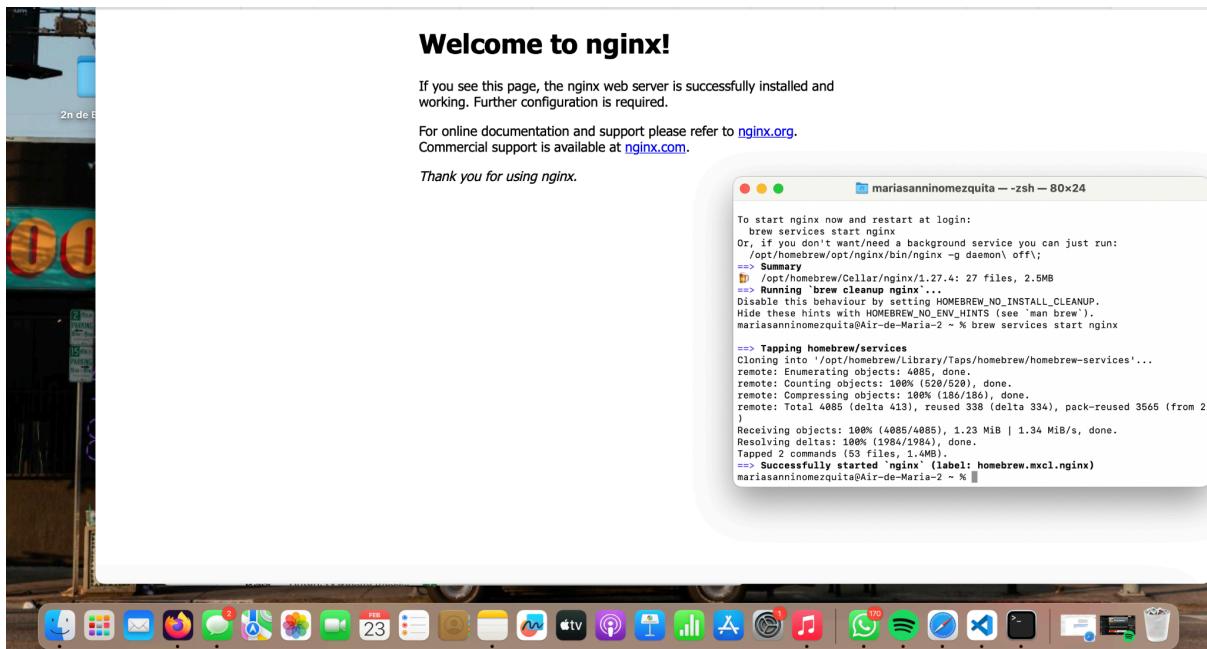
By accessing one of the types, we found the following url:
<https://www.netflix.com/ng/website/memberapi/release/pathEvaluator?webp=true&drmSystem=widevine>... which includes the term "widevine." Widevine is a DRM system developed by Google, confirming that Netflix uses DRM to protect its content.

In order to find the type of codec, we have search for it in the search bar, and have found:



But we can see that it says that it is "Null". So Netflix is not giving us the details. We have searched for other ways to find it, but still no answer. So we searched on the internet and found out that Netflix uses Av1.

ex5:



The screenshot shows a Mac desktop with a terminal window and a browser window. The terminal window is titled 'cdn-up-and-running' and shows a command-line session for setting up a Docker compose service. The browser window displays a GitHub README page for a 'cdn-up-and-running' project, specifically the 'Adding caching capabilities' section.

Terminal Session:

```
-/Desktop/cdn-up-and-running --zsh ...cuments/GitHub/P4_Vid3o --zsh
mariasanninomezquita@Air-de-Maria-2 ~ % cd /Users/mariasanninomezquita/Desktop/cdn-up-and-running
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running % git checkout 1.0.0
Note: switching to '1.0.0'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.
If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:
  git switch -c new-branch-name
Or undo this operation with:
  git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at ca308e6 add bare minimum nginx
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running %
```

Browser Window (GitHub README):

Adding caching capabilities

For the backend service to be cacheable we need to set up the caching policy. We'll use the HTTP header [Cache-Control](#) to setup what caching behavior we want.

```
-- we want the content to be cached by 10 seconds OR the provided max_age (ex: nginx.header['Cache-Control'] = 'public, max-age=' .. (ngx.var.arg_max_age or 10)
```

And, if you want, make sure to check the returned response header [Cache-Control](#).

```
git checkout 1.0.1 # going back to specific configuration
docker-compose run --rm --service-ports backend
http "http://localhost:8080/path/to/my/content.ext?max_age=30"
```

The screenshot shows a Mac desktop with a terminal window and a browser window. The terminal window is titled 'cdn-up-and-running' and shows a command-line session for setting up a Docker compose service. The browser window displays a GitHub README page for a 'cdn-up-and-running' project, specifically the 'Adding caching capabilities' section.

Terminal Session:

```
-/Desktop/cdn-up-and-running --zsh ...cuments/GitHub/P4_Vid3o --zsh
mariasanninomezquita@Air-de-Maria-2 ~ % cd /Users/mariasanninomezquita/Desktop/cdn-up-and-running
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running % docker compose run --rm --service-ports backend
WARN[0000] /Users/mariasanninomezquita/Desktop/cdn-up-and-running/docker-compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
Cannot connect to the Docker daemon at unix:///Users/mariasanninomezquita/.docke/r/run/docker.sock. Is the docker daemon running?
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running % ln -s $(which docker) /usr/local/bin/docker-compose
ln: /usr/local/bin/docker-compose: File exists
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running % docker compose version
Docker Compose version v2.31.0-desktop.2
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running % docker compose run --rm --service-ports backend
WARN[0000] /Users/mariasanninomezquita/Desktop/cdn-up-and-running/docker-compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
Cannot connect to the Docker daemon at unix:///Users/mariasanninomezquita/.docke/r/run/docker.sock. Is the docker daemon running?
mariasanninomezquita@Air-de-Maria-2 cdn-up-and-running %
```

Browser Window (GitHub README):

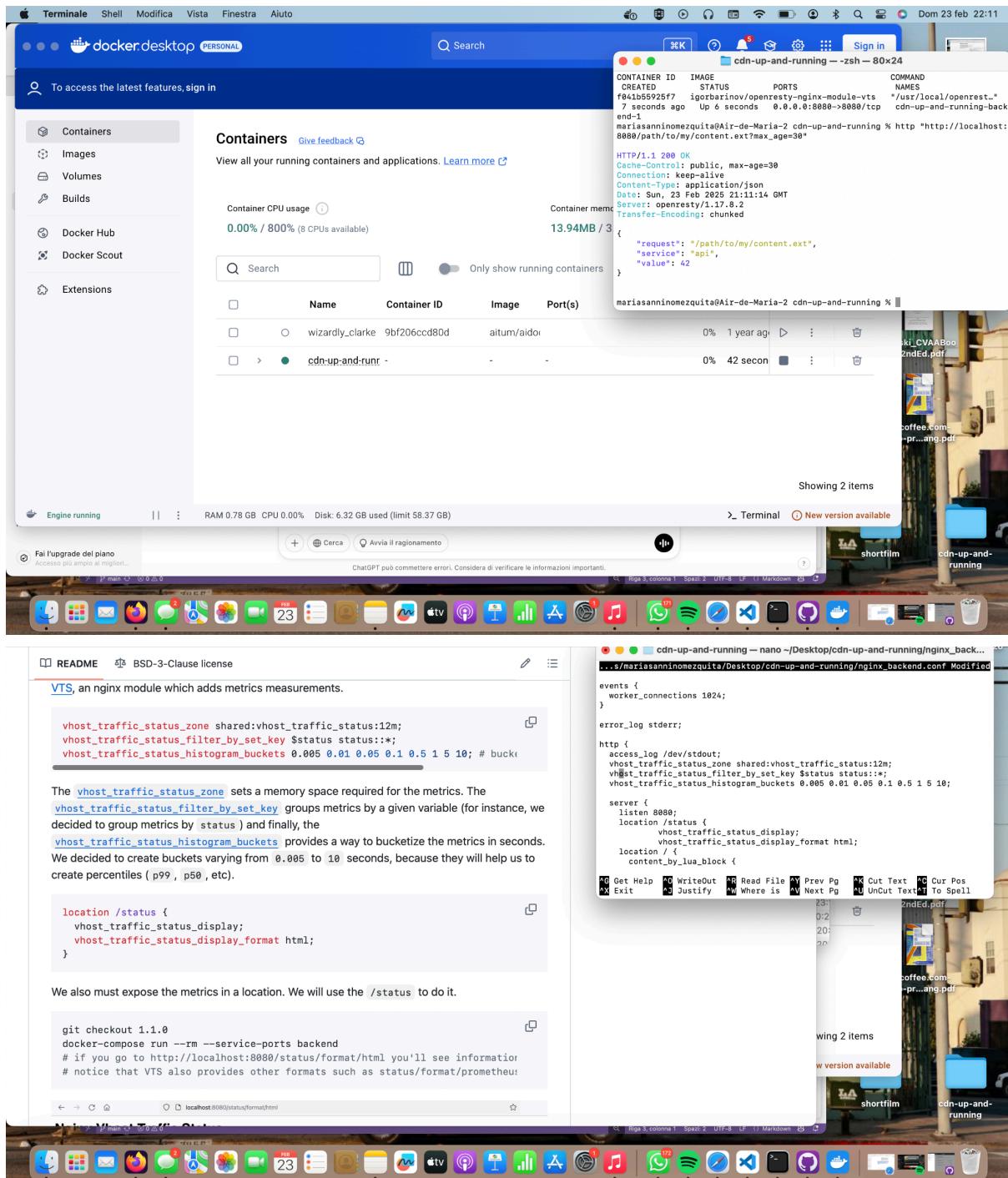
Adding caching capabilities

For the backend service to be cacheable we need to set up the caching policy. We'll use the HTTP header [Cache-Control](#) to setup what caching behavior we want.

```
-- we want the content to be cached by 10 seconds OR the provided max_age (ex: nginx.header['Cache-Control'] = 'public, max-age=' .. (ngx.var.arg_max_age or 10)
```

And, if you want, make sure to check the returned response header [Cache-Control](#).

```
git checkout 1.0.1 # going back to specific configuration
docker-compose run --rm --service-ports backend
http "http://localhost:8080/path/to/my/content.ext?max_age=30"
```



```
{"service": "api", "value": 42, "request": "/"}

cdn-up-and-running --zsh - 80x24
zsh: command not found: vhost_traffic_status_zone
zsh: no matches found: status::*
zsh: command not found: vhost_traffic_status_histogram_buckets
zsh: command not found: #
mariasaninomezquita@Air-de-Maria-2 cdn-up-and-running % vhost_traffic_status_zone
no shared: vhost_traffic_status_zone:12m
vhost_traffic_status_zone set key Status status::+
vhost_traffic_status_histogram_buckets 0.005 0.01 0.05 0.1 0.5 1 5 10;
zsh: command not found: vhost_traffic_status_zone
zsh: no matches found: status::*
zsh: command not found: vhost_traffic_status_histogram_buckets
mariasaninomezquita@Air-de-Maria-2 cdn-up-and-running % cdn-up-and-running/nginx.conf

zsh: no such file or directory: cdn-up-and-running/nginx.conf
mariasaninomezquita@Air-de-Maria-2 cdn-up-and-running % /Users/mariasaninomezquita/Desktop/cdn-up-and-running/nginx_backend.conf
zsh: permission denied: /Users/mariasaninomezquita/Desktop/cdn-up-and-running/nginx_backend.conf
mariasaninomezquita@Air-de-Maria-2 cdn-up-and-running % nano /Users/mariasaninomezquita/Desktop/cdn-up-and-running/nginx_backend.conf

mariasaninomezquita@Air-de-Maria-2 cdn-up-and-running % docker-compose down
docker-compose up -d
```

Nginx Vhost Traffic Status

Server main

Host	Version	Uptime	Connections					Requests					Shared memory				
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode			
1284b8bb67e2	1.17.8	1m 38s	1	0	1	0	1	1	vhost_traffic_status	12.0 MB	7.0 KB	2					

Server zones

Zone	Requests					Responses					Traffic					Cache										
	Total	Req/s	Time	4xx	500	502	504	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
-	8	0	0ms	0	0	0	0	0	8	0	0	0	8	164.2	2.7	3.4 kB	339 B	0	0	0	0	0	0	0	0	0
*	8	0	0ms	0	0	0	0	0	8	0	0	0	8	164.2	2.7	3.4 kB	339 B	0	0	0	0	0	0	0	0	0

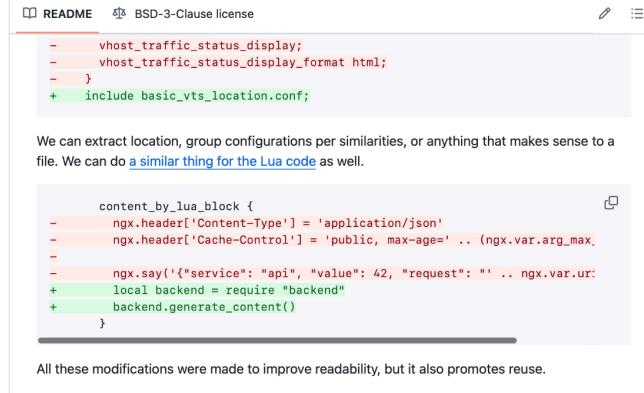
Filters

status::*

Zone	Requests					Responses					Traffic					Cache										
	Total	Req/s	Time	4xx	500	502	504	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
200	8	0	0ms	0	0	0	0	0	8	0	0	0	8	164.2	2.7	3.4 kB	339 B	0	0	0	0	0	0	0	0	0

update interval: sec

[JSON](#) | [GITHUB](#)



We can extract location, group configurations per similarities, or anything that makes sense to a file. We can do a [similar thing for the Lua code](#) as well.

```

content_by_lua_block {
    ngx.header['Content-Type'] = 'application/json'
    ngx.header['Cache-Control'] = 'public, max-age=' .. (ngx.var.arg_max,
    -
    -
    ngx.say('{"service": "api", "value": 42, "request": "' .. ngx.var.ur:
+     local backend = require "backend"
+     backend.generate_content()
}

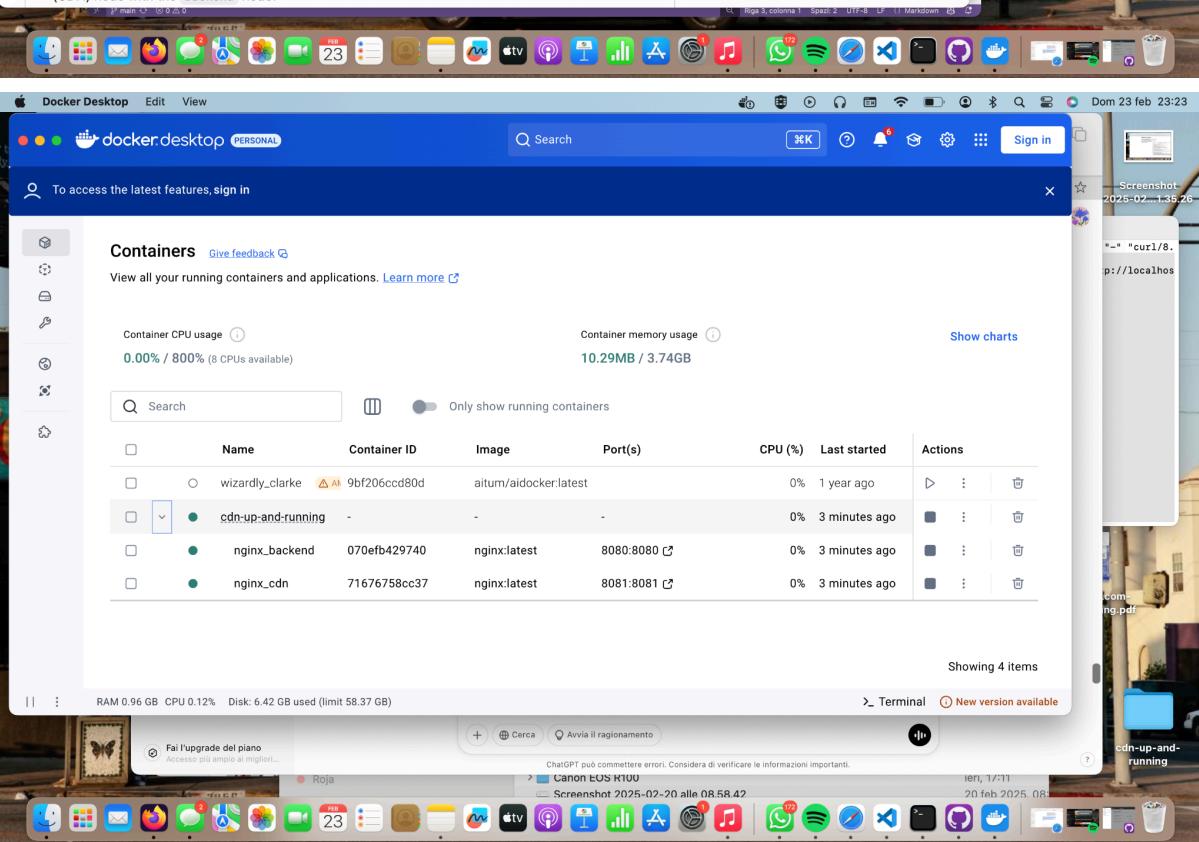
```

All these modifications were made to improve readability, but it also promotes reuse.

The CDN - siting in front of the backend

Proxying

What we did so far has nothing to do with the CDN. Now it's time to start building the CDN. For that, we'll create another node with nginx, just adding a few new directives to connect the edge (CDN) node with the backend node.



Docker Desktop

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
wizardly_clarke	9bf206cccd80d	aitum/aidocker:latest	-	0%	1 year ago	[...]
cdn-up-and-running	-	-	-	0%	3 minutes ago	[...]
nginx_backend	070efb429740	nginx:latest	8080:8080	0%	3 minutes ago	[...]
nginx_cdn	71676758cc37	nginx:latest	8081:8081	0%	3 minutes ago	[...]

Showing 4 items

RAM 0.96 GB CPU 0.12% Disk: 6.42 GB used (limit 58.37 GB)

Terminal

```

Last login: Sun Feb 23 21:48:02 on ttys002
mariasanninomezquita@Air-de-Maria-2: ~$ zshrc:[1]:1: '!' expected
mariasanninomezquita@Air-de-Maria-2: ~$ command not found: ]
mariasanninomezquita@Air-de-Maria-2: ~$ cdn-up-and-running % nano /Users/mariasanninomezquita/Desktop/cdn-up-and-running/nginx_backend.conf
mariasanninomezquita@Air-de-Maria-2: ~$ cdn-up-and-running % curl -X GET http://localhost:8080/
{"service": "api", "value": 42, "request": "/"}

mariasanninomezquita@Air-de-Maria-2: ~$ curl -X GET http://localhost:8081/

```

```

listen 8080;

location / {
    proxy_pass http://backend;
    add_header X-Cache-Status $upstream_cache_status;
}

We also added a new header (X-Cache-Status) to indicate whether the cache was used or not.
• HIT: when the content is in the CDN, the X-Cache-Status should return a hit.
• MISS: when the content isn't in the CDN, the X-Cache-Status should return a miss.

git checkout 2.0.0
docker-compose up
# we still can fetch the content from the backend
http "http://localhost:8080/path/to/my/content.ext"
# but we really want to access the content through the edge (CDN)
http "http://localhost:8081/path/to/my/content.ext"

```

Caching

When we try to fetch content, the X-Cache-Status header is absent. It seems that the edge node is always invariably requesting the backend. This is not the way a CDN should work, right?

```

backend_1 | 172.22.0.4 - - [05/Jan/2022:17:24:48 +0000] "GET /path/to/my/c [D]
edge_1   | 172.22.0.1 - - [05/Jan/2022:17:24:48 +0000] "GET /path/to/my/c [D]

```

getting an error:

```

leandro.moreira at C02D557BML7L in ~/tmp/cdn-up-and-running (tags/2.0.0)
$ http "http://localhost:8081/path/to/something.txt"
HTTP/1.1 200 OK
Cache-Control: public, max-age=10
Connection: keep-alive
Content-Type: application/json
Date: Thu, 06 Jan 2022 10:17:31 GMT
Server: openresty/1.17.8.2
Transfer-Encoding: chunked
X-Cache-Status: HIT
X-Edge: Server

{
    "request": "/path/to/something.txt",
    "service": "api",
    "value": 42
}

```

Monitoring Tools

Checking the cache effectiveness by looking at the command line isn't efficient. It's better if we use a tool for that. **Prometheus** will be used to scrape metrics on all servers, and **Grafana** will show graphics based on the metrics collected by the prometheus.