Primer Bootcamp de DevOps para beginners
Organizado por: Venezolanas in Tech
Desafío Final: La empresa ZERO Technology solicita la contenerización de su aplicación "Products DevOps" en la que incluye Frontend y Backend (products & shopping-cart)

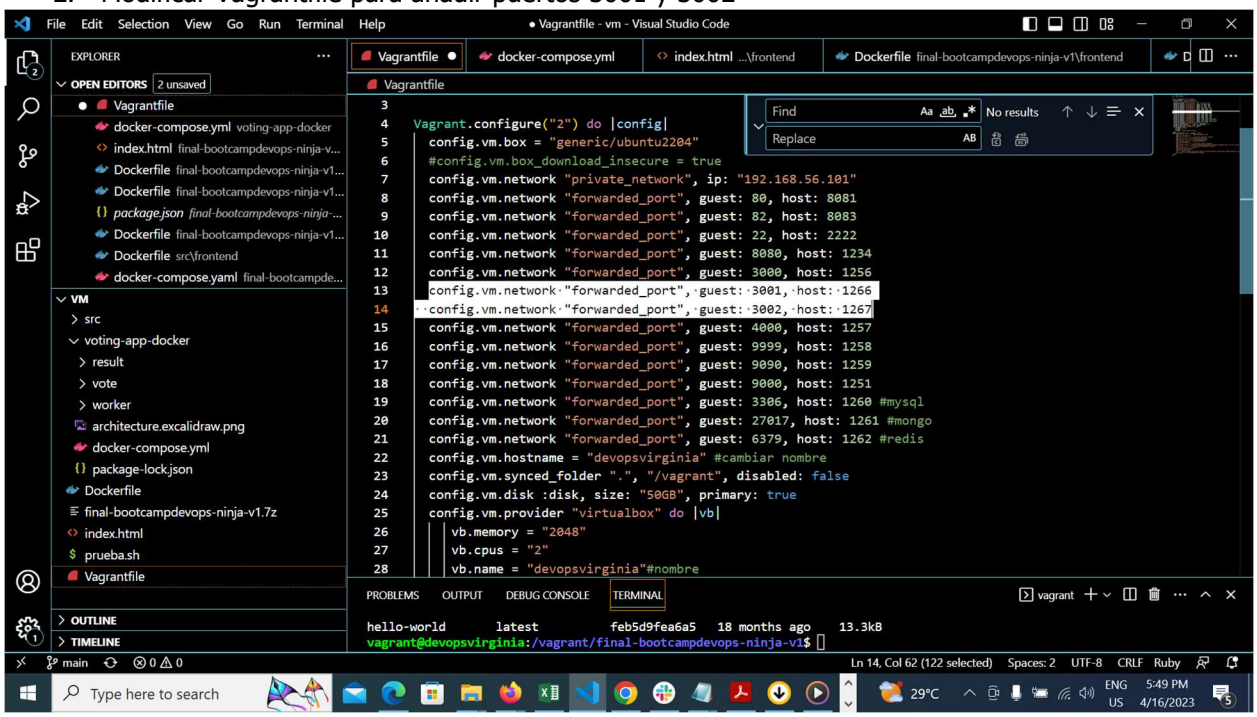
Documentación

1. Instalar Node.js y NPM en Ubuntu 22.04 LTS
   Se constató que los archivos recibidos corresponden scripts de Javascript. Por lo cual se decidió instalar la versión más actualizada

```
vagrant@devopsvirginia:~$ curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.0.2 zlib/1.2.11 brotli/1.0.9 zstd/1.4.8
libidn2/2.3.2 libpsl/0.21.0 (+libidn2/2.3.2) libssh/0.9.6/openssl/zlib nghttp2/1.43.0 librtmp/2.3
OpenLDAP/2.5.13
Release-Date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp
rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz
NTLM NTLM_WB PSL SPNEGO SSL TLS-SRP UnixSockets zstd
vagrant@devopsvirginia:~$
vagrant@devopsvirginia:~$
vagrant@devopsvirginia:~$ curl -fsSL https://deb.nodesource.com/setup_current.x | sudo -E bash -

## Installing the NodeSource Node.js 19.x repo...
## Populating apt-get cache...
+ apt-get update
## Confirming "jammy" is supported...
+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_19.x/dists/jammy/Release'
## Adding the NodeSource signing key to your keyring...
+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | gpg --dearmor | tee
/usr/share/keyrings/nodesource.gpg >/dev/null
## Creating apt sources list file for the NodeSource Node.js 19.x repo...
+ echo 'deb [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_19.x
jammy main' > /etc/apt/sources.list.d/nodesource.list
+ echo 'deb-src [signed-by=/usr/share/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_19.x jammy main' >> /etc/apt/sources.list.d/nodesource.list
## Running `apt-get update` for you...
+ apt-get update
## Run `sudo apt-get install -y nodejs` to install Node.js 19.x and npm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmor | sudo tee
/usr/share/keyrings/yarnkey.gpg >/dev/null
     echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl.yarnpkg.com/debian stable
main" | sudo tee /etc/apt/sources.list.d/yarn.list
     sudo apt-get update && sudo apt-get install yarn
vagrant@devopsvirginia:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 88 not upgraded.
Need to get 29.3 MB of archives.
```

```
After this operation, 189 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_19.x jammy/main amd64 nodejs amd64 19.9.0-deb-1nodesource1
[29.3 MB]
Fetched 29.3 MB in 11s (2,604 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 82999 files and directories currently installed.)
Preparing to unpack .../nodejs_19.9.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (19.9.0-deb-1nodesource1) ...
Setting up nodejs (19.9.0-deb-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
vagrant@devopsvirginia:~$ node --version
v19.9.0
vagrant@devopsvirginia:~$ npm --version
9.6.3
vagrant@devopsvirginia:~$ sudo npm install -g npm

changed 25 packages in 5s
18 packages are looking for funding
  run `npm fund` for details
vagrant@devopsvirginia:~$ npm --version
9.6.4
```

2. Modificar Vagrantfile para añadir puertos 3001 y 3002



Se observó que el desafío final los puertos a utilizar son los siguientes: 3000, 3001 y 3002. Por eso se verificó el Vagrantfile y se adecuó para este fin

En FILE pulsar SAVE para guardar cambio.

Cerrar TERMINAL para abrir uno nuevo.

Hacer click sobre el Vagrantfile y pulsar botón derecho, seleccionando opción OPEN IN INTEGRATED TERMINAL.

Ejecutar comando “vagrant provision”. Como estoy en Venezuela debo activar la VPN para evitar errores en las descargas.
Luego ejecutar comando “vagrant up”, seguido de “vagrant ssh”.
Verificar la versión de Docker ya que se trabajará con Dockerfile, comando “docker - -version”.

3. Dockerfile para el “frontend”

Dockerfile final-bootcampdevops-ninja-v1\frontend | server.js | Dockerfile ...\pr

final-bootcampdevops-ninja-v1 > frontend > Dockerfile

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json /app
RUN npm install
COPY . /app

#with docker-compose: container-name
#ENV PRODUCTS_SERVICE=productsEndpoint container-name
#ENV SHOPPING_CART_SERVICE=shoppingCartEndpoint container-name
#Network Name app

EXPOSE 3000
CMD ["npm","run","start"]
```

FROM
Es una opción de Dockerfile que debe presentarse como la primera instrucción. Cumple con la función de establecer la imagen sobre la cual los pasos e imágenes siguientes se desarrollan en el sistema. Se parte de una Imagen Base, que en este caso es un lenguaje: Javascript.
Se seleccionó esta imagen porque es versión ALPINE: es liviana, no tiene vulnerabilidades y está actualizada.

Pull command copied

TAG
18-alpine
Last pushed 6 days ago by doijanky

docker pull node:18-alpine

DIGEST	OS/ARCH	VULNERABILITIES	COMPRESSED SIZE
44aaf1ccc80e	linux/amd64	None found	50.74 MB
25ba459b834c	linux/arm/v6	None found	49.39 MB
b4c7692301c8	linux/arm/v7	None found	48.7 MB

+3 more...

WORKDIR
Define el directorio donde vamos a agregar nuestro código separando el directorio netamente de la fuente del sistema operativo, estableciendo un área de trabajo.
COPY
Copia del paquete y script hacia el directorio de trabajo.
El asterisco actúa como máscara. El punto asegura que todos los archivos se copien en el directorio de Trabajo, incluyendo cualquier nombre o extensión.

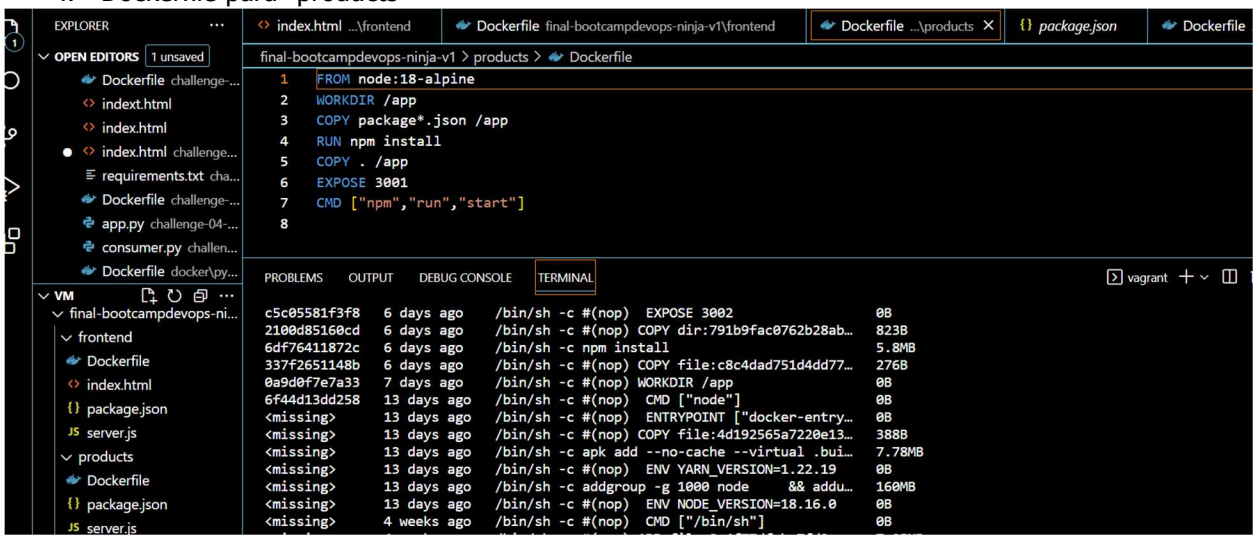
RUN
Ejecuta cualquier comando que sea necesario, en este caso la generación de carpeta “node_modules” porque allí se almacenan todas las librerías descargadas. Se utiliza para crear la imagen de un contenedor.

EXPOSE 3000
Puerto asignado por el desarrollador

CMD
Se encarga del proceso de definición del ejecutable predeterminado de la imagen de Docker, y se usa para añadir (o no) parámetros a dicho ejecutable. En este caso la aplicacion está basada en Node.js, y los argumentos que corresponden a éste para iniciarla.

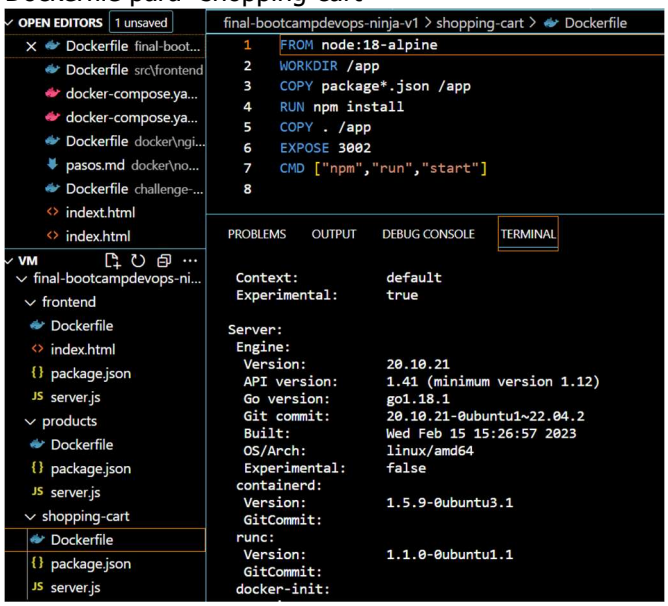
ENV (como comentario, porque se incluirán en la ejecución del comando "docker run")
Los valores que necesita recibir la aplicación para trabajar correctamente.
En este caso, los nombres de los contenedores almacenados en las variables PRODUCTS_SERVICE y SHOPPING_CART_SERVICE.

4. Dockerfile para "products"



La única diferencia con respecto al anterior, es el puerto
EXPOSE 3001
Puerto asignado por el desarrollador

5. Dockerfile para "shopping-cart"



La única diferencia con respecto al anterior, es el puerto
EXPOSE 3002
Puerto asignado por el desarrollador

6. Creación de las Imágenes

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker build -t ms-frontend:1.0
frontend
Sending build context to Docker daemon  10.24kB
Step 1/7 : FROM node:18-alpine
18-alpine: Pulling from library/node
f56be85fc22e: Already exists
931b0e865bc2: Pull complete
60542df8b663: Pull complete
062e26bc2446: Pull complete
Digest: sha256:ca5d399560a9d239cbfa28eec00417f1505e5e108f3ec6938d230767eaa81f61
Status: Downloaded newer image for node:18-alpine
 ---> 6f44d13dd258
Step 2/7 : WORKDIR /app
 ---> Running in df4a22af3165
Removing intermediate container df4a22af3165
 ---> 0a9d0f7e7a33
Step 3/7 : COPY package*.json /app
 ---> 1868d60650f2
Step 4/7 : RUN npm install
 ---> Running in 648963ccf1ab

added 57 packages, and audited 58 packages in 8s

7 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container 648963ccf1ab
 ---> 2aec88a9c1e5
Step 5/7 : COPY . /app
 ---> 10003084ea66
Step 6/7 : EXPOSE 3000
 ---> Running in 109681869f21
Removing intermediate container 109681869f21
 ---> d559d3a8accb
Step 7/7 : CMD ["npm","run","start"]
 ---> Running in 2fe8cf9630ac
Removing intermediate container 2fe8cf9630ac
 ---> a24395126af1
Successfully built a24395126af1
Successfully tagged ms-frontend:1.0

vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker build -t ms-products:1.0
products
Sending build context to Docker daemon  4.096kB
Step 1/7 : FROM node:18-alpine
 ---> 6f44d13dd258
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 0a9d0f7e7a33
Step 3/7 : COPY package*.json /app
 ---> 7b010b8c7926
Step 4/7 : RUN npm install
 ---> Running in b9257e9ee2af
added 50 packages, and audited 51 packages in 4s
```

```
3 high severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container b9257e9ee2af
 ---> 8cddd3fc3558
Step 5/7 : COPY . /app
 ---> efe864dafb0d
Step 6/7 : EXPOSE 3001
 ---> Running in 65a264af7c9e
Removing intermediate container 65a264af7c9e
 ---> f4f1de006b16
Step 7/7 : CMD ["npm","run","start"]
 ---> Running in b0eba8650573
Removing intermediate container b0eba8650573
 ---> 446584114dc6
Successfully built 446584114dc6
Successfully tagged ms-products:1.0

vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker build -t ms-shopping-
cart:1.0 shopping-cart
Sending build context to Docker daemon  4.096kB
Step 1/7 : FROM node:18-alpine
 ---> 6f44d13dd258
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 0a9d0f7e7a33
Step 3/7 : COPY package*.json /app
 ---> 337f2651148b
Step 4/7 : RUN npm install
 ---> Running in d8494fedc6e1

added 50 packages, and audited 51 packages in 4s

3 high severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container d8494fedc6e1
 ---> 6df76411872c
Step 5/7 : COPY . /app
 ---> 2100d85160cd
Step 6/7 : EXPOSE 3002
 ---> Running in d2d902222e5f
Removing intermediate container d2d902222e5f
 ---> c5c05581f3f8
```

```
Step 7/7 : CMD ["npm","run","start"]
 ---> Running in 60fd523a0893
Removing intermediate container 60fd523a0893
 ---> 51df421d5eeb
Successfully built 51df421d5eeb
Successfully tagged ms-shopping-cart:1.0
```

Donde se considera lo siguiente:
t corresponde al tag (etiqueta)
ms-<NOMBRE SERVICIO> es el nombre del micro servicio
:1.0.0 es la versión
<NOMBRE CARPETA> ubicación del Dockerfile, cada micro servicio tiene su carpeta correspondiente

Para verificar la creación de la imagen o imágenes:

```
frontend
  Dockerfile
  index.html
  package.json
  server.js
products
```

```
Successfully built 51df421d5eeb
Successfully tagged ms-shopping-cart:1.0
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker images
REPOSITORY          TAG             IMAGE ID       CREATED              SIZE
ms-shopping-cart    1.0             51df421d5eeb   8 seconds ago        181MB
ms-products         1.0             446584114dc6   34 seconds ago       181MB
ms-frontend         1.0             a24395126af1   About a minute ago   181MB
node                18-alpine       6f44d13dd258   6 days ago           175MB
```

7. Creación de la red (network)
   Tiene la función de crear nuevas nuevas redes en el sistema. Esta opción también resulta de gran utilidad para cumplir labores como conectar los contenedores. De lo contrario, se crearán en sub-distintas por defecto y no podrán comunicarse los contenedores. En este caso, el nombre asignado es "app" y será tipo "bridge".

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker network create app
9cf5e7572278bd387708c2e1b6dc7e6e6ef3f3c1472d6b9487c3a45bb427c884

vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker network ls
NETWORK ID     NAME                              DRIVER    SCOPE
9cf5e7572278   app                               bridge    local
```

8. Creación e inicio de los contenedores

Existen varios aspectos que se consideran obligatorios, a saber:

- Las variables de entorno están definidas en el archive "server.js", a las cuales se les debe asignar un valor (-e PRODUCTS_SERVICE=<NOMBRE CONTENEDOR USADO POR PRODUCTS> y –e SHOPPING_CART_SERVICE=<NOMBRE CONTENEDOR USADO POR SHOPPING-CART>. Como lo especifica el mencionado archivo, almacenan el nombre de cada contenedor que representa a su vez cada micro-servicio.

```
final-bootcampdevops-ninja-v1 > frontend > server.js > ...
let express = require('express');
let path = require('path');
let app = express();

// with docker-compose: container-name, with K8s: service-name
let productsEndpoint = process.env.PRODUCTS_SERVICE || 'localhost'
let shoppingCartEndpoint = process.env.SHOPPING_CART_SERVICE || 'localhost'

app.get('/', function (req, res) {
    res.sendFile(path.join(__dirname, "index.html"));
});

app.get('/get-products', function (req, res) {
    var http = require('http');

    var options = {
        host: productsEndpoint,
        path: '/',
        port: '3001',
        method: 'GET'
    };

    callback = function(response) {
        var str = ''
        response.on('data', function (chunk) {
            str += chunk;
        });

        response.on('end', function () {
            console.log(str);
```

- Asignar nombres a los contenedores, siendo seleccionados los siguientes: "ecom-frontend" para el contenedor que conteneriza el micro servicio del "frontend"; "ecom-products" está vinculado al contenedor de "products" y "ecom-cart" al contenedor de "shopping-cart". Se especifican en el comando con el parámetro "- -name".

- Uso de una única network para todos los contenedores, por las razones mencionadas en el punto anterior

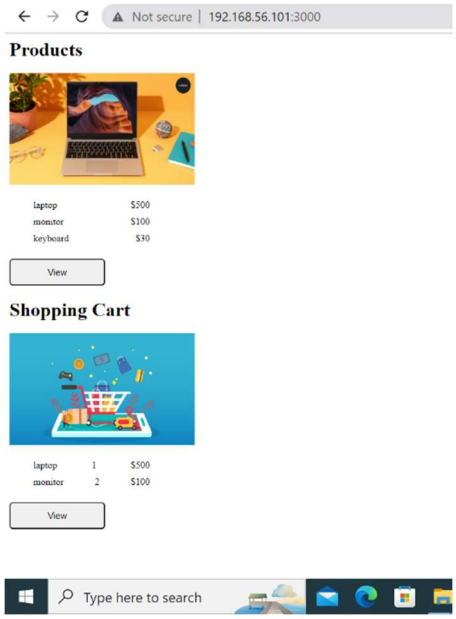
Una vez ejecutado el comando de "docker run", se verifica la creación de los container y su estatus a través del comando "docker ps". En la última columna se puede verificar los nombres de cada contenedor y que éstos coinciden con los definidos en los comandos, parámetro "name". Además se comprueban los valores requeridos para los demás parámetros: IMAGE coincidente (nombre de la imagen), STATUS valor correcto en funcionamiento, PORTS valores coinciden con los enviados por el programador.

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker run -d --name ecom-frontend -p 3000:3000 \
> -e PRODUCTS_SERVICE=ecom-products \
> -e SHOPPING_CART_SERVICE=ecom-cart \
> --network app ms-frontend:1.0
7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker run -d --name ecom-products -p 3001:3001 --network app ms-pr
:1.0
29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker run -d --name ecom-cart -p 3002:3002 --network app ms-shoppi
t:1.0
f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS          PORTS                    NAMES
f47a7e2016b8   ms-shopping-cart:1.0  "docker-entrypoint.s…"   5 seconds ago    Up 4 seconds    0.0.0.0:3002->3002/tcp   ecom-cart
29dd75656d1e   ms-products:1.0       "docker-entrypoint.s…"   22 seconds ago   Up 22 seconds   0.0.0.0:3001->3001/tcp   ecom-prod
7daa88ed5079   ms-frontend:1.0       "docker-entrypoint.s…"   41 seconds ago   Up 40 seconds   0.0.0.0:3000->3000/tcp   ecom-fron
```

9. Resultado

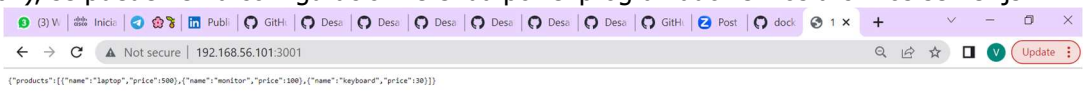
Así se muestra la imagen principal cuando se accesa desde el localhost a través del puerto asignado.

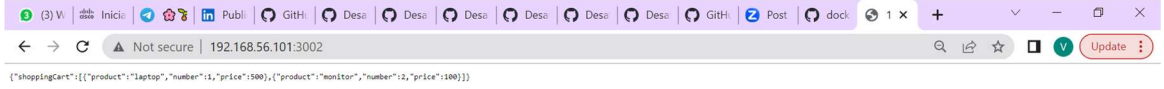
Al hacer click sobre el botón de "View" de cada sección se muestran el listado y precio de cada ítem



10. Revisión de Configuración y estatus

Si accede desde el localhost a través de los puertos asignados para Products (3001) y Shopping-cart (3002), se puede ver la configuración referida por el programador en los archivos server.js.





Al ejecutar el comando de "docker logs", no se encuentra algún error.

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1/frontend$ docker logs ecom-frontend

> microservice-frontend@1.0.0 start
> node server.js

app listening on port 3000!
{"products":[{"name":"laptop","price":500},{"name":"monitor","price":100},{"name":"keyboard","price":30}]}
{"shoppingCart":[{"product":"laptop","number":1,"price":500},{"product":"monitor","number":2,"price":100}]}
```

Con el comando "docker inspect" para cada contenedor se puede hacer una revisión exhaustiva de la configuración completa

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker inspect ecom-frontend
[
    {
        "Id": "7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee",
        "Created": "2023-04-20T05:47:46.316927363Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "npm",
            "run",
            "start"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 2831,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-04-20T05:47:46.82057939Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:a24395126af1d075737a673fecc78944da7df59a8ab9d361b4b5531c679fcedf",
        "ResolvConfPath":
"/var/lib/docker/containers/7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee/reso
lv.conf",
        "HostnamePath":
"/var/lib/docker/containers/7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee/host
name",
        "HostsPath":
"/var/lib/docker/containers/7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee/host
s",
        "LogPath":
"/var/lib/docker/containers/7daa88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee/7daa
88ed50792b00ba31c2ad37a9d7c0b4db4a543b974be4a0eb3090beb9ebee-json.log",
        "Name": "/ecom-frontend",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
```

```
            "NetworkMode": "app",
            "PortBindings": {
                "3000/tcp": [
                    {
                        "HostIp": "",
                        "HostPort": "3000"
                    }
                ]
            },
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
            "VolumeDriver": "",
            "VolumesFrom": null,
            "CapAdd": null,
            "CapDrop": null,
            "CgroupnsMode": "private",
            "Dns": [],
            "DnsOptions": [],
            "DnsSearch": [],
            "ExtraHosts": null,
            "GroupAdd": null,
            "IpcMode": "private",
            "Cgroup": "",
            "Links": null,
            "OomScoreAdj": 0,
            "PidMode": "",
            "Privileged": false,
            "PublishAllPorts": false,
            "ReadonlyRootfs": false,
            "SecurityOpt": null,
            "UTSMode": "",
            "UsernsMode": "",
            "ShmSize": 67108864,
            "Runtime": "runc",
            "ConsoleSize": [
                0,
                0
            ],
            "Isolation": "",
            "CpuShares": 0,
            "Memory": 0,
            "NanoCpus": 0,
            "CgroupParent": "",
            "BlkioWeight": 0,
            "BlkioWeightDevice": [],
            "BlkioDeviceReadBps": null,
            "BlkioDeviceWriteBps": null,
            "BlkioDeviceReadIOps": null,
            "BlkioDeviceWriteIOps": null,
            "CpuPeriod": 0,
            "CpuQuota": 0,
            "CpuRealtimePeriod": 0,
            "CpuRealtimeRuntime": 0,
            "CpusetCpus": "",
            "CpusetMems": "",
            "Devices": [],
            "DeviceCgroupRules": null,
            "DeviceRequests": null,
            "KernelMemory": 0,
            "KernelMemoryTCP": 0,
            "MemoryReservation": 0,
            "MemorySwap": 0,
            "MemorySwappiness": null,
```

```
            "OomKillDisable": null,
            "PidsLimit": null,
            "Ulimits": null,
            "CpuCount": 0,
            "CpuPercent": 0,
            "IOMaximumIOps": 0,
            "IOMaximumBandwidth": 0,
            "MaskedPaths": [
                "/proc/asound",
                "/proc/acpi",
                "/proc/kcore",
                "/proc/keys",
                "/proc/latency_stats",
                "/proc/timer_list",
                "/proc/timer_stats",
                "/proc/sched_debug",
                "/proc/scsi",
                "/sys/firmware"
            ],
            "ReadonlyPaths": [
                "/proc/bus",
                "/proc/fs",
                "/proc/irq",
                "/proc/sys",
                "/proc/sysrq-trigger"
            ]
        },
        "GraphDriver": {
            "Data": {
                "LowerDir":
"/var/lib/docker/overlay2/e7a2a6eb9bbd2cf30445983e4597a128c8f529ef3f8d97e9f4fe1b2c6ebbbd0c-
init/diff:/var/lib/docker/overlay2/70d675b137dc9e04c457df670e792a0b6ce929c904babec22fe739e2226ad3
18/diff:/var/lib/docker/overlay2/218971fbecb79c0581cf30f63b6e85c6b6538afc622ce76008893b2498039d23
/diff:/var/lib/docker/overlay2/7e9fa2bc70b3cf5d9f48d34f35328d8748d4c99bf002f6bfc17cf18fb6c9c33d/d
iff:/var/lib/docker/overlay2/2d835bf0ef7b833df095d5e291079558d6caacb9c50d0795acba343e3ec5fa57/dif
f:/var/lib/docker/overlay2/f71983fbfddbca753233a479f6b9273ee43ec7f2c119580af7e77c55ec0dd06c/diff:
/var/lib/docker/overlay2/ce5e8d16f1ba75fce9ab0e39942bccecc97112ace54c2af1704ce2fa67a16a3f/diff:/v
ar/lib/docker/overlay2/7e386ab1dc51a48a70efcc86cb949f875d3fd0cd9db98eddbea43e1f313d648d/diff:/var
/lib/docker/overlay2/f35099929628161892cb4d9a8bc3ca59ae9ff5fafcfef85890749780c0282693/diff",
                "MergedDir":
"/var/lib/docker/overlay2/e7a2a6eb9bbd2cf30445983e4597a128c8f529ef3f8d97e9f4fe1b2c6ebbbd0c/merged
",
                "UpperDir":
"/var/lib/docker/overlay2/e7a2a6eb9bbd2cf30445983e4597a128c8f529ef3f8d97e9f4fe1b2c6ebbbd0c/diff",
                "WorkDir":
"/var/lib/docker/overlay2/e7a2a6eb9bbd2cf30445983e4597a128c8f529ef3f8d97e9f4fe1b2c6ebbbd0c/work"
            },
            "Name": "overlay2"
        },
        "Mounts": [],
        "Config": {
            "Hostname": "7daa88ed5079",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "3000/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PRODUCTS_SERVICE=ecom-products",
                "SHOPPING_CART_SERVICE=ecom-cart",
```

```
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "NODE_VERSION=18.16.0",
                "YARN_VERSION=1.22.19"
            ],
            "Cmd": [
                "npm",
                "run",
                "start"
            ],
            "Image": "ms-frontend:1.0",
            "Volumes": null,
            "WorkingDir": "/app",
            "Entrypoint": [
                "docker-entrypoint.sh"
            ],
            "OnBuild": null,
            "Labels": {}
        },
        "NetworkSettings": {
            "Bridge": "",
            "SandboxID": "2121c440356912c647af2c3f12a264a138a40925859fed9894a8ea2f4bc44b47",
            "HairpinMode": false,
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "Ports": {
                "3000/tcp": [
                    {
                        "HostIp": "0.0.0.0",
                        "HostPort": "3000"
                    }
                ]
            },
            "SandboxKey": "/var/run/docker/netns/2121c4403569",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "",
            "Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "MacAddress": "",
            "Networks": {
                "app": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "7daa88ed5079"
                    ],
                    "NetworkID":
"9cf5e7572278bd387708c2e1b6dc7e6e6ef3f3c1472d6b9487c3a45bb427c884",
                    "EndpointID":
"caa0f8e258190ee1ca0b58cab4658dfe56887fca6754aa3dcfb236513b44a21d",
                    "Gateway": "172.20.0.1",
                    "IPAddress": "172.20.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:14:00:02",
                    "DriverOpts": null
                }
            }
        }
    }
```

```
]
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker inspect ecom-products
[
    {
        "Id": "29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6",
        "Created": "2023-04-20T05:48:05.074839286Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "npm",
            "run",
            "start"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 2975,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-04-20T05:48:05.645253309Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:446584114dc6924682d12ef9fed344355f891bc212bad3e7a09e1506463d6d6c",
        "ResolvConfPath":
"/var/lib/docker/containers/29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6/reso
lv.conf",
        "HostnamePath":
"/var/lib/docker/containers/29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6/host
name",
        "HostsPath":
"/var/lib/docker/containers/29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6/host
s",
        "LogPath":
"/var/lib/docker/containers/29dd75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6/29dd
75656d1eb1ba33d804d3205551aad486d8f4dd64a392de76c2f8b0bea2f6-json.log",
        "Name": "/ecom-products",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "app",
            "PortBindings": {
                "3001/tcp": [
                    {
                        "HostIp": "",
                        "HostPort": "3001"
                    }
                ]
            },
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
```

"AutoRemove": false,
"VolumeDriver": "",
"VolumesFrom": null,
"CapAdd": null,
"CapDrop": null,
"CgroupnsMode": "private",
"Dns": [],
"DnsOptions": [],
"DnsSearch": [],
"ExtraHosts": null,
"GroupAdd": null,
"IpcMode": "private",
"Cgroup": "",
"Links": null,
"OomScoreAdj": 0,
"PidMode": "",
"Privileged": false,
"PublishAllPorts": false,
"ReadonlyRootfs": false,
"SecurityOpt": null,
"UTSMode": "",
"UsernsMode": "",
"ShmSize": 67108864,
"Runtime": "runc",
"ConsoleSize": [
    0,
    0
],
"Isolation": "",
"CpuShares": 0,
"Memory": 0,
"NanoCpus": 0,
"CgroupParent": "",
"BlkioWeight": 0,
"BlkioWeightDevice": [],
"BlkioDeviceReadBps": null,
"BlkioDeviceWriteBps": null,
"BlkioDeviceReadIOps": null,
"BlkioDeviceWriteIOps": null,
"CpuPeriod": 0,
"CpuQuota": 0,
"CpuRealtimePeriod": 0,
"CpuRealtimeRuntime": 0,
"CpusetCpus": "",
"CpusetMems": "",
"Devices": [],
"DeviceCgroupRules": null,
"DeviceRequests": null,
"KernelMemory": 0,
"KernelMemoryTCP": 0,
"MemoryReservation": 0,
"MemorySwap": 0,
"MemorySwappiness": null,
"OomKillDisable": null,
"PidsLimit": null,
"Ulimits": null,
"CpuCount": 0,
"CpuPercent": 0,
"IOMaximumIOps": 0,
"IOMaximumBandwidth": 0,
"MaskedPaths": [
    "/proc/asound",
    "/proc/acpi",
    "/proc/kcore",
    "/proc/keys",
    "/proc/latency_stats",

```
                "/proc/timer_list",
                "/proc/timer_stats",
                "/proc/sched_debug",
                "/proc/scsi",
                "/sys/firmware"
            ],
            "ReadonlyPaths": [
                "/proc/bus",
                "/proc/fs",
                "/proc/irq",
                "/proc/sys",
                "/proc/sysrq-trigger"
            ]
        },
        "GraphDriver": {
            "Data": {
                "LowerDir":
"/var/lib/docker/overlay2/26adf9d670b4ad516c6bf7de7b233b0f1c6bb14e4fa1b26027ea03d805cab769-
init/diff:/var/lib/docker/overlay2/329d76ef8afc59871e18d53b9cf5b9ce41446e7610fb31d9c8424f09804cae
7a/diff:/var/lib/docker/overlay2/9f3c8bd2e14a3af935f268337707c7f5028f2c16a79a6e24e4924857a4c8bc00
/diff:/var/lib/docker/overlay2/8ceab86ab136a7167b46951d7d03aba0f5a0ea04a9b602a053ec66b092294ff1/d
iff:/var/lib/docker/overlay2/2d835bf0ef7b833df095d5e291079558d6caacb9c50d0795acba343e3ec5fa57/dif
f:/var/lib/docker/overlay2/f71983fbfddbca753233a479f6b9273ee43ec7f2c119580af7e77c55ec0dd06c/diff:
/var/lib/docker/overlay2/ce5e8d16f1ba75fce9ab0e39942bccecc97112ace54c2af1704ce2fa67a16a3f/diff:/v
ar/lib/docker/overlay2/7e386ab1dc51a48a70efcc86cb949f875d3fd0cd9db98eddbea43e1f313d648d/diff:/var
/lib/docker/overlay2/f35099929628161892cb4d9a8bc3ca59ae9ff5fafcfef85890749780c0282693/diff",
                "MergedDir":
"/var/lib/docker/overlay2/26adf9d670b4ad516c6bf7de7b233b0f1c6bb14e4fa1b26027ea03d805cab769/merged
",
                "UpperDir":
"/var/lib/docker/overlay2/26adf9d670b4ad516c6bf7de7b233b0f1c6bb14e4fa1b26027ea03d805cab769/diff",
                "WorkDir":
"/var/lib/docker/overlay2/26adf9d670b4ad516c6bf7de7b233b0f1c6bb14e4fa1b26027ea03d805cab769/work"
            },
            "Name": "overlay2"
        },
        "Mounts": [],
        "Config": {
            "Hostname": "29dd75656d1e",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "3001/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "NODE_VERSION=18.16.0",
                "YARN_VERSION=1.22.19"
            ],
            "Cmd": [
                "npm",
                "run",
                "start"
            ],
            "Image": "ms-products:1.0",
            "Volumes": null,
            "WorkingDir": "/app",
            "Entrypoint": [
                "docker-entrypoint.sh"
            ],
```

```
            "OnBuild": null,
            "Labels": {}
        },
        "NetworkSettings": {
            "Bridge": "",
            "SandboxID": "fc55652e0bb5991506e315e6a57ec1fd85cfb7cb9d4b2a31c21d575fa6df9367",
            "HairpinMode": false,
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "Ports": {
                "3001/tcp": [
                    {
                        "HostIp": "0.0.0.0",
                        "HostPort": "3001"
                    }
                ]
            },
            "SandboxKey": "/var/run/docker/netns/fc55652e0bb5",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "",
            "Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "MacAddress": "",
            "Networks": {
                "app": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "29dd75656d1e"
                    ],
                    "NetworkID":
"9cf5e7572278bd387708c2e1b6dc7e6e6ef3f3c1472d6b9487c3a45bb427c884",
                    "EndpointID":
"533b462d3ac732b7f5cad015344940f307a14e7ec5157b955b00c2dca6094074",
                    "Gateway": "172.20.0.1",
                    "IPAddress": "172.20.0.3",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:14:00:03",
                    "DriverOpts": null
                }
            }
        }
    }
]
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$
```


```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker inspect ecom-cart
```


```
[
    {
        "Id": "f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756",
        "Created": "2023-04-20T05:48:22.336531921Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "npm",
            "run",
            "start"
        ],
```

```
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 3126,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-04-20T05:48:22.884386942Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:51df421d5eeb8b4fc5b8dc62068853297420b36caeb07e3f46c2d6027fd0104b",
        "ResolvConfPath":
"/var/lib/docker/containers/f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756/reso
lv.conf",
        "HostnamePath":
"/var/lib/docker/containers/f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756/host
name",
        "HostsPath":
"/var/lib/docker/containers/f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756/host
s",
        "LogPath":
"/var/lib/docker/containers/f47a7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756/f47a
7e2016b8c9b9764ca790d1db1e0e917070a157b808385aba434484bca756-json.log",
        "Name": "/ecom-cart",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "app",
            "PortBindings": {
                "3002/tcp": [
                    {
                        "HostIp": "",
                        "HostPort": "3002"
                    }
                ]
            },
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
            "VolumeDriver": "",
            "VolumesFrom": null,
            "CapAdd": null,
            "CapDrop": null,
            "CgroupnsMode": "private",
            "Dns": [],
            "DnsOptions": [],
            "DnsSearch": [],
            "ExtraHosts": null,
            "GroupAdd": null,
            "IpcMode": "private",
```

"Cgroup": "",
"Links": null,
"OomScoreAdj": 0,
"PidMode": "",
"Privileged": false,
"PublishAllPorts": false,
"ReadonlyRootfs": false,
"SecurityOpt": null,
"UTSMode": "",
"UsernsMode": "",
"ShmSize": 67108864,
"Runtime": "runc",
"ConsoleSize": [
    0,
    0
],
"Isolation": "",
"CpuShares": 0,
"Memory": 0,
"NanoCpus": 0,
"CgroupParent": "",
"BlkioWeight": 0,
"BlkioWeightDevice": [],
"BlkioDeviceReadBps": null,
"BlkioDeviceWriteBps": null,
"BlkioDeviceReadIOps": null,
"BlkioDeviceWriteIOps": null,
"CpuPeriod": 0,
"CpuQuota": 0,
"CpuRealtimePeriod": 0,
"CpuRealtimeRuntime": 0,
"CpusetCpus": "",
"CpusetMems": "",
"Devices": [],
"DeviceCgroupRules": null,
"DeviceRequests": null,
"KernelMemory": 0,
"KernelMemoryTCP": 0,
"MemoryReservation": 0,
"MemorySwap": 0,
"MemorySwappiness": null,
"OomKillDisable": null,
"PidsLimit": null,
"Ulimits": null,
"CpuCount": 0,
"CpuPercent": 0,
"IOMaximumIOps": 0,
"IOMaximumBandwidth": 0,
"MaskedPaths": [
    "/proc/asound",
    "/proc/acpi",
    "/proc/kcore",
    "/proc/keys",
    "/proc/latency_stats",
    "/proc/timer_list",
    "/proc/timer_stats",
    "/proc/sched_debug",
    "/proc/scsi",
    "/sys/firmware"
],
"ReadonlyPaths": [
    "/proc/bus",
    "/proc/fs",
    "/proc/irq",
    "/proc/sys",
    "/proc/sysrq-trigger"

```
            ]
        },
        "GraphDriver": {
            "Data": {
                "LowerDir":
"/var/lib/docker/overlay2/3fe91c0afd832c8f59a3f28f777491363cff20a36da20e483a40dbf306de851a-
init/diff:/var/lib/docker/overlay2/1a27ec2766bb2660174f7c2d112baa8788c1cbfab393db141ffe43a6a23ae0
7a/diff:/var/lib/docker/overlay2/2badc668f071be1783389c5764799d134a703031f73fe52ad5946db942a53bf5
/diff:/var/lib/docker/overlay2/08e195660b2197cafadfb2f4a3864d2e19c9862b42cbb7d93bce19f105f8b21a/d
iff:/var/lib/docker/overlay2/2d835bf0ef7b833df095d5e291079558d6caacb9c50d0795acba343e3ec5fa57/dif
f:/var/lib/docker/overlay2/f71983fbfddbca753233a479f6b9273ee43ec7f2c119580af7e77c55ec0dd06c/diff:
/var/lib/docker/overlay2/ce5e8d16f1ba75fce9ab0e39942bcccecc97112ace54c2af1704ce2fa67a16a3f/diff:/v
ar/lib/docker/overlay2/7e386ab1dc51a48a70efcc86cb949f875d3fd0cd9db98eddbea43e1f313d648d/diff:/var
/lib/docker/overlay2/f35099929628161892cb4d9a8bc3ca59ae9ff5fafcfef85890749780c0282693/diff",
                "MergedDir":
"/var/lib/docker/overlay2/3fe91c0afd832c8f59a3f28f777491363cff20a36da20e483a40dbf306de851a/merged
",
                "UpperDir":
"/var/lib/docker/overlay2/3fe91c0afd832c8f59a3f28f777491363cff20a36da20e483a40dbf306de851a/diff",
                "WorkDir":
"/var/lib/docker/overlay2/3fe91c0afd832c8f59a3f28f777491363cff20a36da20e483a40dbf306de851a/work"
            },
            "Name": "overlay2"
        },
        "Mounts": [],
        "Config": {
            "Hostname": "f47a7e2016b8",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "3002/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "NODE_VERSION=18.16.0",
                "YARN_VERSION=1.22.19"
            ],
            "Cmd": [
                "npm",
                "run",
                "start"
            ],
            "Image": "ms-shopping-cart:1.0",
            "Volumes": null,
            "WorkingDir": "/app",
            "Entrypoint": [
                "docker-entrypoint.sh"
            ],
            "OnBuild": null,
            "Labels": {}
        },
        "NetworkSettings": {
            "Bridge": "",
            "SandboxID": "0e60df8aead17e21ed5c9bff7952e7966e6382b42a56d4ca4f8ec71fc7dc7b75",
            "HairpinMode": false,
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "Ports": {
                "3002/tcp": [
                    {
```

```
                        "HostIp": "0.0.0.0",
                        "HostPort": "3002"
                    }
                ]
            },
            "SandboxKey": "/var/run/docker/netns/0e60df8aead1",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "",
            "Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "MacAddress": "",
            "Networks": {
                "app": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "f47a7e2016b8"
                    ],
                    "NetworkID":
"9cf5e7572278bd387708c2e1b6dc7e6e6ef3f3c1472d6b9487c3a45bb427c884",
                    "EndpointID":
"9d553d9853bc83562ebcdeb0b5a45d3fceaec3362b69660108fa3974528c7870",
                    "Gateway": "172.20.0.1",
                    "IPAddress": "172.20.0.4",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:14:00:04",
                    "DriverOpts": null
                }
            }
        }
    }
]
```

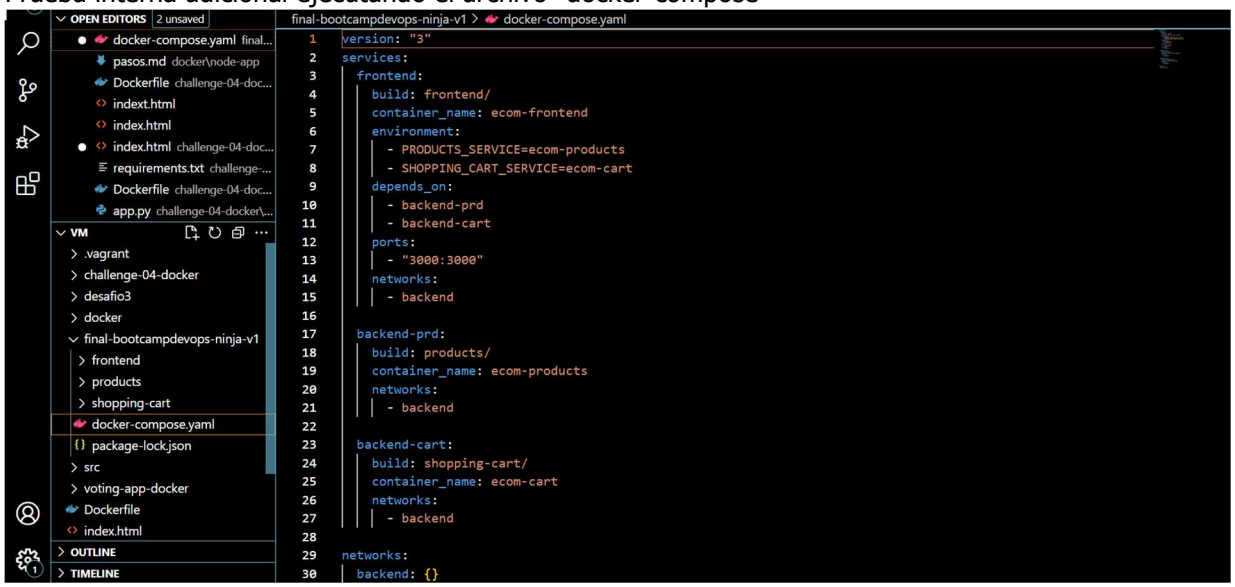

Como los contenedores se están ejecutando, con el comando "docker exec" se puede verificar en el directorio de trabajo las librerías. ¡Se ve ordenado y separado del código fuente!

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker exec -it ecom-frontend sh
/app # printenv
NODE_VERSION=18.16.0
HOSTNAME=7daa88ed5079
YARN_VERSION=1.22.19
SHLVL=1
HOME=/root
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/app
PRODUCTS_SERVICE=ecom-products
SHOPPING_CART_SERVICE=ecom-cart
/app # ls
Dockerfile         index.html         node_modules       package-lock.json  package.json       server.js
/app # cd node_modules
/app/node_modules # ls
accepts            depd               function-bind      methods            proxy-addr         statuses
array-flatten      destroy            get-intrinsic      mime               qs                 toidentifier
body-parser        ee-first           has                mime-db            range-parser       type-is
bytes              encodeurl          has-symbols        mime-types         raw-body           unpipe
call-bind          escape-html        http-errors        ms                 safe-buffer        utils-merge
content-disposition etag              iconv-lite         negotiator         safer-buffer       vary
content-type       express            inherits           object-inspect     send
cookie             finalhandler       ipaddr.js          on-finished        serve-static
cookie-signature   forwarded          media-typer        parseurl           setprototypeof
debug              fresh              merge-descriptors  path-to-regexp     side-channel
/app/node_modules #
```

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker exec -it ecom-products sh
/app # printenv
NODE_VERSION=18.16.0
HOSTNAME=29dd75656d1e
YARN_VERSION=1.22.19
SHLVL=1
HOME=/root
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/app
/app # ls
Dockerfile         node_modules       package-lock.json  package.json       server.js
/app # cd node_modules
/app/node_modules # ls
accepts            depd               fresh              mime-db            range-parser       type-is
array-flatten      destroy            http-errors        mime-types         raw-body           unpipe
body-parser        ee-first           iconv-lite         ms                 safe-buffer        utils-merge
bytes              encodeurl          inherits           negotiator         safer-buffer       vary
content-disposition escape-html       ipaddr.js          on-finished        send
content-type       etag               media-typer        parseurl           serve-static
cookie             express            merge-descriptors  path-to-regexp     setprototypeof
cookie-signature   finalhandler       methods            proxy-addr         statuses
debug              forwarded          mime               qs                 toidentifier
/app/node_modules #
```

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker exec -it ecom-cart sh
/app # printenv
NODE_VERSION=18.16.0
HOSTNAME=f47a7e2016b8
YARN_VERSION=1.22.19
SHLVL=1
HOME=/root
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/app
/app # ls
Dockerfile         node_modules       package-lock.json  package.json       server.js
/app # cd node_modules
/app/node_modules # ls
accepts            depd               fresh              mime-db            range-parser       type-is
array-flatten      destroy            http-errors        mime-types         raw-body           unpipe
body-parser        ee-first           iconv-lite         ms                 safe-buffer        utils-merge
bytes              encodeurl          inherits           negotiator         safer-buffer       vary
content-disposition escape-html       ipaddr.js          on-finished        send
content-type       etag               media-typer        parseurl           serve-static
cookie             express            merge-descriptors  path-to-regexp     setprototypeof
cookie-signature   finalhandler       methods            proxy-addr         statuses
debug              forwarded          mime               qs                 toidentifier
/app/node_modules #
```

11. Prueba interna adicional ejecutando el archivo "docker-compose"



Tomar en cuenta las siguientes consideraciones:

- Versión máxima a usar será la número 3.
- Se puede definir todos los servicios que necesita mi aplicación en un único archivo, extensión YAML.
- El contenido de este archivo son las variables o parámetros del comando docker build y docker run.
- En el docker-compose se utilizan los nombres de los contenedores, como se indica en el archivo server.js añadido dentro de la carpeta relacionada al servicio "frontend".
- Las variables de entorno especificadas en el archivo server.js añadido dentro de la carpeta relacionada al servicio "frontend", se denominan PRODUCTS_SERVICE y SHOPPING_CART_SERVICE. La primera relacionada al host Products y la segunda la Shopping-cart.
- Se definen los nombres de los contenedores para ser utilizados por las variables de entorno.
- El servicio Frontend depende del estado del Backend (Products y Shopping-Cart). Si éstos dos no están habilitados, el frontend no puede iniciarse.

Se ejecuta el comando "docker-compose build" para construir las imágenes

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker-compose build
[+] Building 0.0s (0/0)
[+] Building 0.0s (0/1)
 => [internal] load build definition from Dockerfile                                    0.0s
[+] Building 0.1s (1/2)
 => [internal] load build definition from Dockerfile                                    0.0s
[+] Building 0.2s (4/9)
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 31B                                                     0.0s
 => [internal] load .dockerignore                                                       0.0s
[+] Building 0.4s (7/9)
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 31B                                                     0.0s
[+] Building 10.2s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 31B                                                     0.0s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 2B                                                         0.0s
[+] Building 10.8s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 31B                                                     0.0s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 2B                                                         0.0s
 => [internal] load metadata for docker.io/library/node:18-alpine                       0.0s
 => [1/5] FROM docker.io/library/node:18-alpine                                         0.0s
 => [internal] load build context                                                       0.1s
 => => transferring context: 92B                                                        0.0s
 => CACHED [2/5] WORKDIR /app                                                           0.0s
 => CACHED [3/5] COPY package*.json /app                                                0.0s
 => [4/5] RUN npm install                                                              10.2s
 => [5/5] COPY . /app                                                                   0.1s
 => exporting to image                                                                  0.2s
 => [5/5] COPY . /app                                                                   0.1s
 => exporting to image                                                                  0.3s
 => => exporting layers                                                                 0.3s
 => => writing image sha256:adb8839afba6c19b617bfc5714658663938d8bfc0098f90876145cb7a0366616   0.0s
 => => naming to docker.io/library/final-bootcampdevops-ninja-v1-frontend               0.0s
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$
```

Verificar si las imágenes se crearon, una para cada servicio

```
 => => writing image sha256:adb8839afba6c19b617bfc5714658663938d8bfc0098f90876145cb7a0366616   0.0s
 => => naming to docker.io/library/final-bootcampdevops-ninja-v1-frontend                      0.0s
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker images
REPOSITORY                                  TAG       IMAGE ID       CREATED          SIZE
final-bootcampdevops-ninja-v1-frontend      latest    adb8839afba6   8 minutes ago    181MB
final-bootcampdevops-ninja-v1-backend-cart  latest    ce636671593e   8 minutes ago    181MB
final-bootcampdevops-ninja-v1-backend-prd   latest    1c978f95bb55   8 minutes ago    181MB
```

Se ejecuta comando "docker-cpmpose up –d" para crear e iniciar los contenedores. Este comando equivale al equivale a "doker run –it"

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker-compose up -d
[+] Running 4/4
 ⠿ Network final-bootcampdevops-ninja-v1_backend  Created                                                    0.1s
 ⠿ Container ecom-products                        Started                                                    1.1s
 ⠿ Container ecom-cart                            Started                                                    1.1s
 ⠿ Container ecom-frontend                        Started                                                    2.1s
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker ps
CONTAINER ID   IMAGE                                        COMMAND                  CREATED          STATUS          PORTS
               NAMES
4c473933b501   final-bootcampdevops-ninja-v1-frontend       "docker-entrypoint.s…"   11 seconds ago   Up 8 seconds    0.0.0.0:3000
->3000/tcp   ecom-frontend
00a12cc1be3c   final-bootcampdevops-ninja-v1-backend-prd    "docker-entrypoint.s…"   11 seconds ago   Up 10 seconds   3001/tcp
               ecom-products
bbb654cf33fc   final-bootcampdevops-ninja-v1-backend-cart   "docker-entrypoint.s…"   11 seconds ago   Up 10 seconds   3002/tcp
               ecom-cart
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$
```

Con el comando "docker ps" se confirma el estatus de los contenedores y la configuración de los puertos más los nombres de cada contenedor.

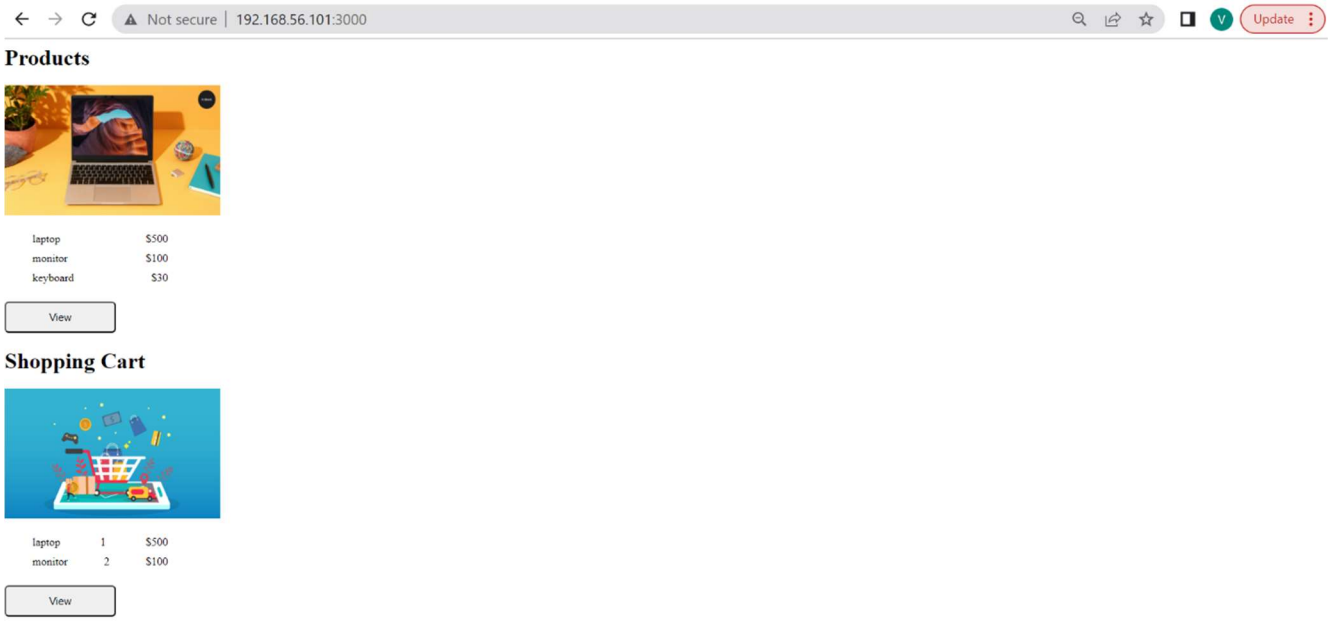
Se verifica que en log no muestre algún error.

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker-compose logs
ecom-cart  |
ecom-cart  | > microservice-shopping-cart@1.0.0 start
ecom-cart  | > node server.js
ecom-cart  |
ecom-cart  | app listening on port 3002!
ecom-products  |
ecom-products  | > microservice-products@1.0.0 start
ecom-products  | > node server.js
ecom-products  |
ecom-products  | app listening on port 3001!
ecom-frontend  |
ecom-frontend  | > microservice-frontend@1.0.0 start
ecom-frontend  | > node server.js
ecom-frontend  |
ecom-frontend  | app listening on port 3000!
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$
```

Se verifica si la red fue creada, para comunicación de los contenedores

```
vagrant@devopsvirginia:/vagrant/final-bootcampdevops-ninja-v1$ docker network ls
NETWORK ID     NAME                                   DRIVER    SCOPE
9cf5e7572278   app                                    bridge    local
20ff9550107f   bridge                                 bridge    local
5dc99bfe2109   final-bootcampdevops-ninja-v1_backend  bridge    local
3330742a71ed   host                                   host      local
b68815c078b7   none                                   null      local
```

¡El resultado es el esperado!



12. Kubernetes

Paso 1
Levantar las imágenes con los okerfile previamente creados

Paso 2
Agregar las etiquetas a las imágenes
docker tag ms-frontend:1.0 vickyaurfali/ms-frontend:1.0
docker tag ms-products:1.0 vickyaurfali/ms-products:1.0
docker tag ms-shopping-cart:1.0 vickyaurfali/ms-shopping-cart:1.0

Paso 3
Loguearse en dockerhub para hacer un push de las imágenes construidas
Para realizar el logueo
docker login luego colocar el usuario y contraseña
docker push miusuario/miimagen
Ejemplo:
docker push vickyaurfali/ms-frontend:1.0
docker push vickyaurfali/ms-products:1.0
docker push vickyaurfali/ms-shopping-cart:1.0

Paso 4
crear yaml para el namespace
crear yaml para el frontend - (al estar expuesto de debe especificar el tipo Nodeport)
crear yaml para el products
crear yaml para el shopping
Para cada archivo colocar el nombre del namespace
y en imagen el nombre de las imagenes en Docker

Paso 5
Ejecutar el comando para crear el namespace
kubectl apply -f 01-ninja_ns.yaml

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl apply -f 01-ninja_ns.yaml
namespace/grpvnzla-ninja created
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$
```

Paso 6
Ejecuto los siguientes comandos para hacer el deploy y el servicio
kubectl get all -n grpvnzla-ninja
kubectl apply -f 03-ninja-products.yaml
kubectl apply -f 02-ninja-frontend.yaml
kubectl apply -f 04-ninja-shopping-cart.yaml

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl apply -f 04-ninja-shopping-cart.yaml
deployment.apps/ms-shopping-cart created
service/ms-shopping-cart-svc created
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl apply -f 03-ninja-products.yaml
deployment.apps/ms-products created
service/ms-products-svc created
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl apply -f 02-ninja-frontend.yaml
deployment.apps/ms-frontend created
service/ms-frontend-svc created
```

Paso 7
Ejecuto el siguiente comando para ver qué tengo en mi namespace
kubectl get all -n grpvnzla-ninja

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl get all -n grpvnzla-ninja
NAME                                    READY   STATUS    RESTARTS   AGE
pod/ms-frontend-55bc84b56b-zcmd5        1/1     Running   0          5h4m
pod/ms-products-9544bcf8b-zltz7         1/1     Running   0          5h4m
pod/ms-shopping-cart-8487ddd446-pqlwx   1/1     Running   0          5h4m

NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/ms-frontend-svc       NodePort    10.96.73.212    <none>        3000:30005/TCP   5h4m
service/ms-products-svc       ClusterIP   10.96.101.239   <none>        3001/TCP         5h4m
service/ms-shopping-cart-svc  ClusterIP   10.96.245.251   <none>        3002/TCP         5h4m

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ms-frontend       1/1     1            1           5h4m
deployment.apps/ms-products       1/1     1            1           5h4m
deployment.apps/ms-shopping-cart  1/1     1            1           5h4m

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/ms-frontend-55bc84b56b       1         1         1       5h4m
replicaset.apps/ms-products-9544bcf8b        1         1         1       5h4m
replicaset.apps/ms-shopping-cart-8487ddd446  1         1         1       5h4m
```

Paso 8
Se realiza un port-forward para el servicio
kubectl -n grpvnzla-ninja port-forward --address 0.0.0.0 service/ms-frontend-svc 3000:3000

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl -n grpvnzla-ninja port-forward service/ms-fro
ntend-svc 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
```

Hago un curl http://localhost:3000

```
vagrant@kindk8s:/$ curl http://localhost:3000

<html lang="en">
<style>
    .container {
        margin: 20px auto;
    }
    .button {
        width: 160px;
        height: 45px;
        border-radius: 6px;
        font-size: 15px;
        margin-top: 20px;
    }
    img {
        width: 310px;
        height: 187px;
        display: block;
        margin-bottom: 20px;
    }
    hr {
        width: 400px;
        margin-left: 0;
    }
    h3 {
        display: inline-block;
    }
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        display: flex;
        flex-direction: column;
    }
    ul li {
        width: 240px;
        display: flex;
        justify-content: space-between;
    }
    ul li span {
        padding: 5px;
        margin-left: 35px;
    }
```

```
        let spanPrice = document.createElement('span');

        spanName.setAttribute("class", "name");
        spanNumber.setAttribute("class", "name");
        spanPrice.setAttribute("class", "price");
        li.setAttribute("class", "shopping-cart-item");

        spanName.innerText = item.product;
        spanNumber.innerText = item.number;
        spanPrice.innerText = `$${item.price}`;

        li.appendChild(spanName);
        li.appendChild(spanNumber);
        li.appendChild(spanPrice);
        productsList.appendChild(li);

      });
    }
  </script>
  <body>
  <div>
      <h1>Products</h1>
      <img src='https://images.prismic.io/frameworkmarketplace/5dc5fc06-aec5-4f28-a924-0230aa91a360_Pre-Marketplac
  e+-+image_02.jpg?auto=compress,format' alt="user-profile">
  </div>

  <div class='container' id='products-container'>
      <ul id="products"></ul>
      <button class='button' onclick="handleProductsRequest()">View</button>
  </div>

  <div>
      <h1>Shopping Cart</h1>
      <img src='https://www.statcan.gc.ca/o1/sites/default/files/2021-11/shopping_2.jpg' alt="user-profile"></div>


  <div class='container' id='shopping-cart-container'>
      <ul id="shopping-cart"></ul>
      <button class='button' onclick="handleShoppingCartRequest()">View</button>
  </div>
  </body>
  </html>
  vagrant@kindk8s:/$
  vagrant@kindk8s:/$
```

Paso 9

kubectl apply -f 05-ninja-local-ingress_rc.yaml

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl apply -f 05-ninja-local-ingress_rc.yaml
ingress.networking.k8s.io/nginx-ingress created
```

Notas:

Para configurar el controlador de ingreso debo saber cuál es el tipo de Ingress Class

Puedes tener una o más depende de la configuración que se esté utilizando o del servidor donde se está desplegando

El ingress debe estar en el mismo namespace de donde está mi aplicación

Si se ejecuta un kubectl get all -A puedo ver la que tiene el controlador por defecto

```
NAMESPACE           NAME                                          DESIRED   CURRENT   READY   AGE
grpvnzla-ninja      replicaset.apps/ms-frontend-55bc84b56b        1         1         1       6h43m
grpvnzla-ninja      replicaset.apps/ms-products-9544bcf8b         1         1         1       6h43m
grpvnzla-ninja      replicaset.apps/ms-shopping-cart-8487ddd446   1         1         1       6h44m
ingress-nginx       replicaset.apps/ingress-nginx-controller-69dfcc796b   1   1         1       18h
kube-system         replicaset.apps/coredns-565d847f94            2         2         2       18h
local-path-storage  replicaset.apps/local-path-provisioner-684f458cdd   1   1         1       18h
```

kubectl describe replicaset.apps/ingress-nginx-controller-69dfcc796b -n ingress-nginx

Me va mostrar mi ingress class , la misma debe estar definida en mi archivo yaml

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl describe replicaset.apps/ingress-nginx-controller-69dfcc796b -n ingress-nginx
Name:           ingress-nginx-controller-69dfcc796b
Namespace:      ingress-nginx
Selector:       app.kubernetes.io/component=controller,app.kubernetes.io/instance=ingress-nginx,app.kubernetes.io/name=ingress-nginx,pod-template-hash=69dfcc796b
Labels:         app.kubernetes.io/component=controller
                app.kubernetes.io/instance=ingress-nginx
                app.kubernetes.io/name=ingress-nginx
                app.kubernetes.io/part-of=ingress-nginx
                app.kubernetes.io/version=1.7.0
                pod-template-hash=69dfcc796b
Annotations:    deployment.kubernetes.io/desired-replicas: 1
                deployment.kubernetes.io/max-replicas: 2
                deployment.kubernetes.io/revision: 1
Controlled By:  Deployment/ingress-nginx-controller
Replicas:       1 current / 1 desired
Pods Status:    1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:           app.kubernetes.io/component=controller
                    app.kubernetes.io/instance=ingress-nginx
                    app.kubernetes.io/name=ingress-nginx
                    app.kubernetes.io/part-of=ingress-nginx
                    app.kubernetes.io/version=1.7.0
                    pod-template-hash=69dfcc796b
  Service Account:  ingress-nginx
  Containers:
   controller:
    Image:       registry.k8s.io/ingress-nginx/controller:v1.7.0@sha256:7612338342a1e7b8090bef78f2a04fffcadd548ccaabe8a47bf7758ff549a5f7
    Ports:       80/TCP, 443/TCP, 8443/TCP
    Host Ports:  80/TCP, 443/TCP, 0/TCP
    Args:
      /nginx-ingress-controller
      --election-id=ingress-nginx-leader
      --controller-class=k8s.io/ingress-nginx
      --ingress-class=nginx
      --configmap=$(POD_NAMESPACE)/ingress-nginx-controller
      --validating-webhook=:8443
      --validating-webhook-certificate=/usr/local/certificates/cert
      --validating-webhook-key=/usr/local/certificates/key
      --watch-ingress-without-class=true
      --publish-status-address=localhost
    Requests:
      cpu:      100m
      memory:   90Mi
    Liveness:   http-get http://:10254/healthz delay=10s timeout=1s period=10s #success=1 #failure=5
    Readiness:  http-get http://:10254/healthz delay=10s timeout=1s period=10s #success=1 #failure=3
    Environment:
      POD_NAME:       (v1:metadata.name)
```

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ kubectl describe ing nginx-ingress -n grpvnzla-ninja
Name:             nginx-ingress
Labels:           <none>
Namespace:        grpvnzla-ninja
Address:          localhost
Ingress Class:    nginx
Default backend:  <default>
Rules:
  Host        Path  Backends
  ----        ----  --------
  *
              /   ms-frontend-svc:3000 (10.244.1.5:3000)
Annotations:  <none>
Events:
  Type    Reason  Age                From                      Message
  ----    ------  ----               ----                      -------
  Normal  Sync    15m (x2 over 16m)  nginx-ingress-controller  Scheduled for sync
```

Paso 10

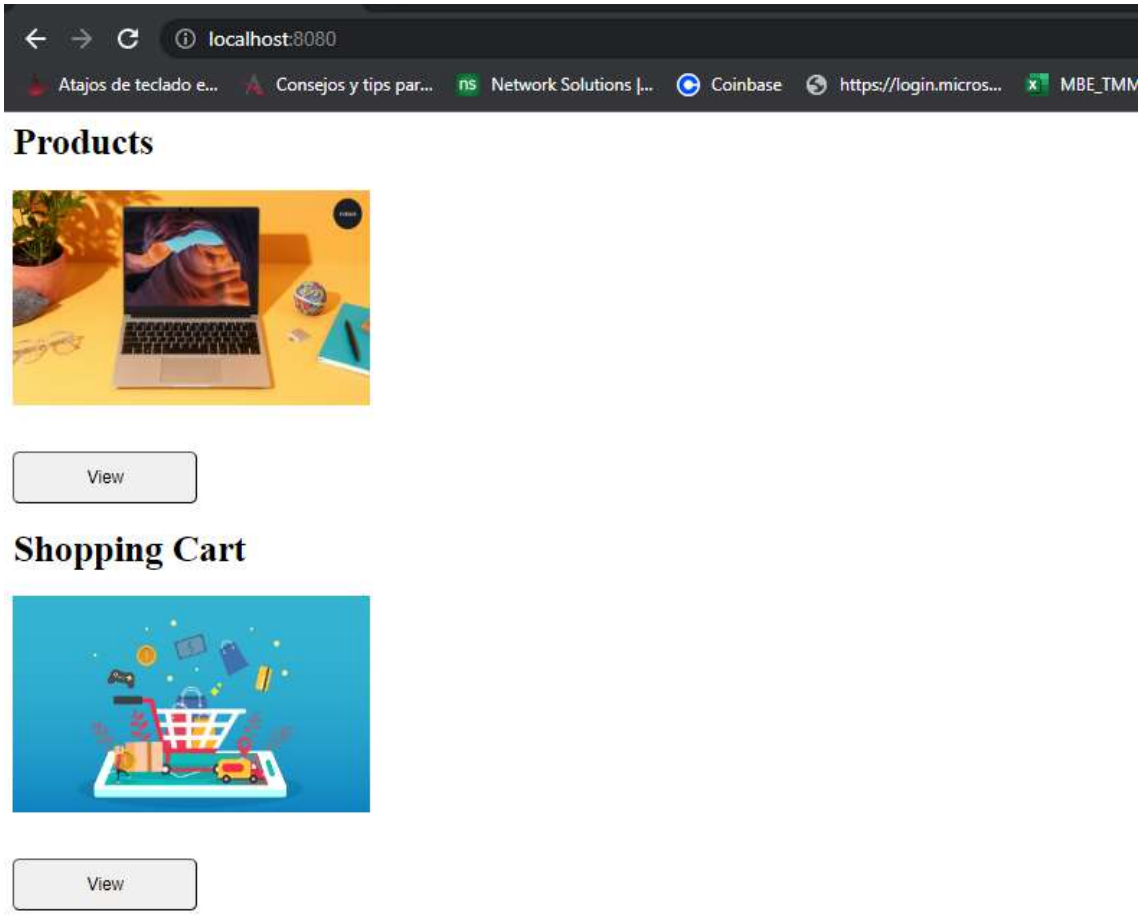
Luego si yo hago un curl http://localhost

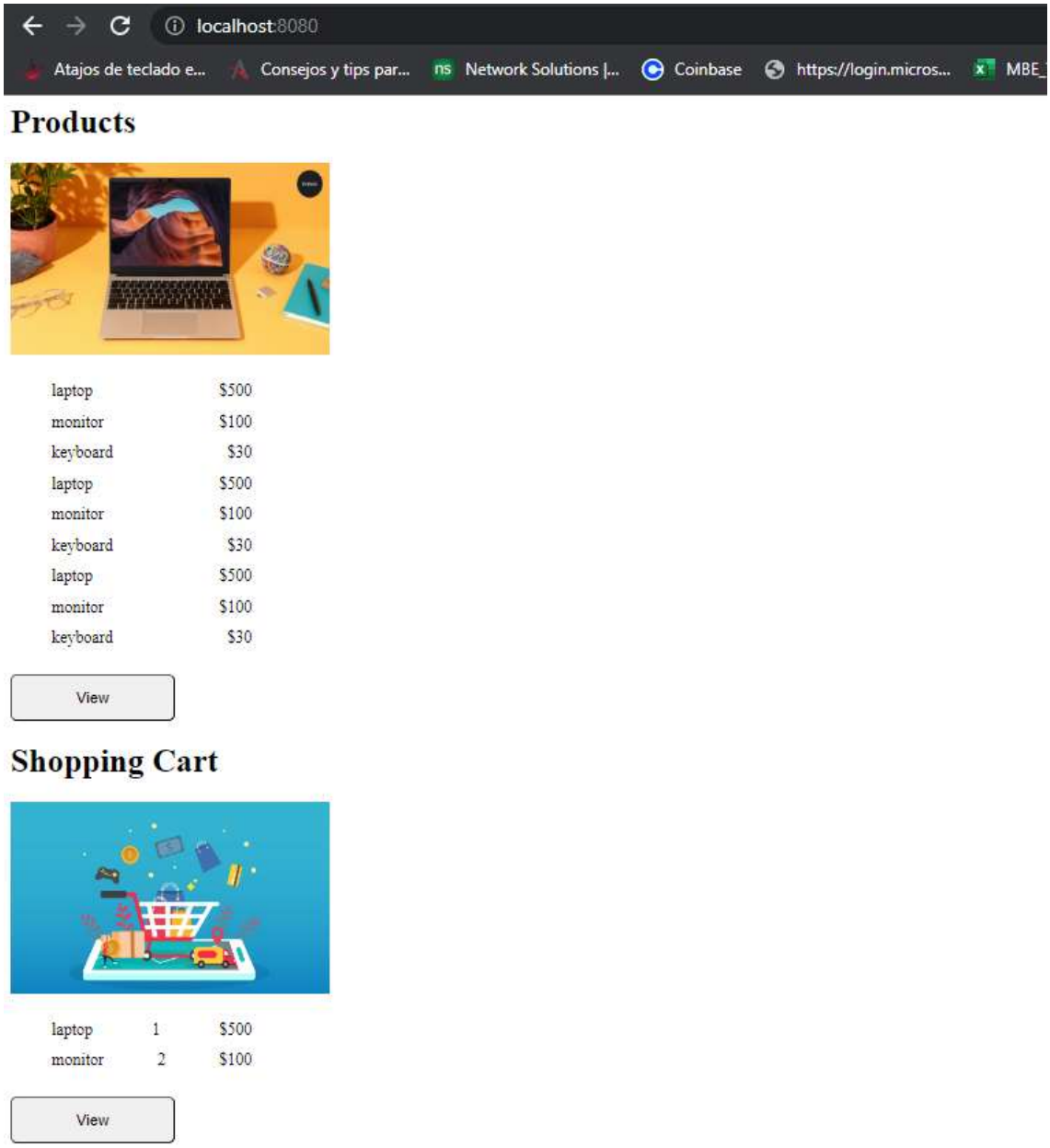
se utiliza para hacer una solicitud HTTP a la dirección local y la respuesta que recibimos es un archivo HTML que contiene código JavaScript para solicitar y mostrar productos y carritos de compras.

```
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$ curl http://localhost
<html lang="en">
<style>
    .container {
        margin: 20px auto;
    }
    .button {
        width: 160px;
        height: 45px;
        border-radius: 6px;
        font-size: 15px;
        margin-top: 20px;
    }
    img {
        width: 310px;
        height: 187px;
        display: block;
        margin-bottom: 20px;
    }
    hr {
        width: 400px;
        margin-left: 0;
    }
    h3 {
        display: inline-block;
    }
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        display: flex;
        flex-direction: column;
    }
    ul li {
        width: 240px;
        display: flex;
        justify-content: space-between;
    }
```

```
      spanNumber.setAttribute("class", "name");
      spanPrice.setAttribute("class", "price");
      li.setAttribute("class", "shopping-cart-item");

      spanName.innerText = item.product;
      spanNumber.innerText = item.number;
      spanPrice.innerText = `$${item.price}`;

      li.appendChild(spanName);
      li.appendChild(spanNumber);
      li.appendChild(spanPrice);
      productsList.appendChild(li);

    });
  }
</script>
<body>
<div>
    <h1>Products</h1>
    <img src='https://images.prismic.io/frameworkmarketplace/5dc5fc06-aec5-4f28-a924-0230aa91a360_Pre-Marketplace+-+image_02.jpg?auto=compress,format' alt="user-profile">
</div>

<div class='container' id='products-container'>
    <ul id="products"></ul>
    <button class='button' onclick="handleProductsRequest()">View</button>
</div>

<div>
    <h1>Shopping Cart</h1>
    <img src='https://www.statcan.gc.ca/o1/sites/default/files/2021-11/shopping_2.jpg' alt="user-profile">
</div>

<div class='container' id='shopping-cart-container'>
    <ul id="shopping-cart"></ul>
    <button class='button' onclick="handleShoppingCartRequest()">View</button>
</div>
</body>
</html>
vagrant@kindk8s:/vagrant/final-devops-ninja-v2/kubernetes$
```

Si voy a mi navegador localhost:8080 puedo ver mi página web


localhost:8080
Atajos de teclado e...
Consejos y tips par...
Network Solutions |...
Coinbase
https://login.micros...
MBE_TMM
Products
View
Shopping Cart
View



¿Por qué se debería configurar un Ingress?

Un Ingress Controller es necesario en Kubernetes para permitir el acceso a los servicios que se ejecutan dentro del clúster Kubernetes desde fuera del clúster.

Un Ingress Controller es un componente de Kubernetes que actúa como una capa de entrada para el tráfico HTTP y HTTPS hacia los servicios dentro del clúster. Permite que se configuren las reglas de enrutamiento y las políticas de seguridad para el tráfico que entra en el clúster, y permite a los desarrolladores y administradores de Kubernetes exponer servicios de forma más fácil y segura.

Sin un Ingress Controller, se debe configurar manualmente el acceso a cada servicio individual. Con un Ingress Controller, se pueden exponer múltiples servicios a través de una única dirección IP y un puerto, y se puede enrutar el tráfico a diferentes servicios basándose en el nombre del host o la ruta del URI.

En resumen, la configuración de un Ingress Controller en Kubernetes es necesaria para exponer servicios de forma más fácil y segura, y para permitir el acceso a los servicios desde fuera del clúster Kubernetes.

Buenas Prácticas

- Seleccionar una imagen que sea la más adecuada con la aplicación o lenguaje.
- Usar imágenes que tengan el aviso "Docker Official Image" ya que son más seguras, de lo contrario pueden tener vulnerabilidades.
- Evitar utilizar la etiqueta de imagen base latest, porque apuntará a una imagen diferente cuando se publique una nueva versión y la construcción no será la más estable.
- Utilizar imágenes basadas en ALPINE ya que son más livianas o de bases reducidas.
- Que los archivos de mi imagen queden en un directorio de trabajo separado del código fuente o directorios propios del sistema. Esto provoca que tener ficheros en dicho directorio que no son necesarios para la construcción de la imagen provocará contexto de construcción más grandes y, en consecuencia, imágenes de mayor tamaño. También, resulta en un tiempo de construcción mayor y en un contenedor (una vez ejecutado) con mayor uso de memoria.
- En este caso, podemos aislar dentro de nuestro proyecto el fichero Dockerfile dentro de un subdirectorio que contendrá sólo aquello necesario.
- Cuando se ejecuta el comando CMD en el Dockerfile, es mejor indicar a los argumentos que vayan directamente al apartado del script. Porque si el desarrollador hace cambios, incluyendo el nombre, la imagen sigue siendo utilizable.
- Llevar la aplicación al estado “En Producción” mediante Dockerfile ya que es más seguro, porque los valores de las variables de entorno quedan expuestas en caso de que se usen en docker-compose.yaml. Preferiblemente usarlo en ambientes de prueba sólo para extrapolar el ambiente en producción.
- Una aplicación o imagen por contenedor, podamos ejecutar o destruir uno en cualquier momento sin afectar otras. Si existen varias aplicaciones en un contenedor, una de ellas podría dejar de funcionar y sería difícil conocer su estado.
- Optimización de la caché de construcción de Docker. Docker utiliza una memoria caché con la idea de agilizar la construcción de imágenes. Las imágenes son construidas por capas, cada instrucción dentro de un fichero Dockerfile resulta en una capa de la imagen. Durante la construcción, siempre que se pueda, Docker tiende a reutilizar las capas de una imagen de una construcción anterior, obviando un paso que podría resultar costoso. Por lo antes expuesto es recomendable colocar las instrucciones del Dockerfile que tienden a cambiar en la parte final del fichero. De este modo, Docker podrá reutilizar las capas anteriores. También agrupar instrucciones en una misma capa (instrucción del fichero Dockerfile) si corresponden al mismo comando.
- Usar adecuadamente los tags de la imagen
- Las imagenes de Docker se identifican de dos maneras: el nombre y el tag. El formato que sigue es el siguiente: nginx:1.15.5 donde “nginx” es el nombre y “1.15.5” el tag.
- Aprovechar los tags para versionar. De este modo, se facilita la liberación de código. Es un método muy flexible, puesto que podemos utilizar las tags para obtener variantes muy diversas, no sólo

indicando incrementalmente y con una política adecuada nuestra versión (esto en sí mismo es una buena práctica), sino indicar diferentes variantes para una misma versión.

- Se realizó prueba de laboratorio utilizando la versión de express 4.17.1, si bien es una versión que data de hace aproximadamente cinco años y está caduca, ésta se encuentra referenciada en el archivos packages.json de las aplicaciones a desplegar. Se conoce dentro de las buenas prácticas para creación de imágenes en Docker, la importancia de utilizar versiones de software y sistemas operativos recientes, estables y oficiales; no obstante se respetó los requerimientos del mencionado archivo .json para evaluar comportamiento y características de las imágenes, contendores y aplicaciones.

  La generación de las imágenes Docker generó una cantidad importe de errores de auditoría y fue necesario deshabilitar esta función, con el fin de poder construir dichas imágenes, también se observó la creación de contenedores temporales, lo que según se investigó, es causado por la utilización de versiones antiguas de software en la construcción de estas imágenes. El despliegue de las aplicaciones luego de este punto, fue fluido y exitoso, funcionando de forma correcta, lo cual debe generar alertas y despertar la conciencia, ya que de cara al usuario todo pareciera estar correcto, pero tanto las imágenes generadas con software de vieja data, como los contenedores y aplicaciones, presentan importantes vulnerabilidades de seguridad. Muy interesante el desafío, en especial en este sentido.

  Se anexa la salida de consola del despliegue manual de las aplicaciones, los Dockerfile utilizados y la evidencia de la funcionalidad de las aplicaciones en navegador web.

###FRONTEND

```
FROM node:alpine
WORKDIR /usr/src/app
COPY package*.json .
RUN npm i webpack-4.17.1 --no-bin-links
COPY index.html .
COPY server.js .
EXPOSE 3000
CMD ["npm", "start"]
```

###PRODUCTS

```
FROM node:alpine
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm i webpack-4.17.1 --no-bin-links
COPY server.js .
EXPOSE 3001
CMD ["npm", "start"]
```

###SHOPPING CART

```
FROM node:alpine
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm i webpack-4.17.1 --no-bin-links
COPY server.js .
EXPOSE 3002
CMD ["npm", "start"]
```

```
vagrant@mhoheb:/vagrant/REPO/src$ docker build -t ms-frontend:1.0 frontend
Sending build context to Docker daemon  33.02MB
Step 1/8 : FROM node:alpine
 ---> 182dfd1d5db3
Step 2/8 : WORKDIR /usr/src/app
 ---> Running in 223446765cc4
Removing intermediate container 223446765cc4
 ---> 9d2a3658a4f2
Step 3/8 : COPY package*.json ./
```

```
 ---> b12f08ce38d2
Step 4/8 : RUN npm i webpack-4.17.1 --no-bin-links
 ---> Running in 09fe87b5ce2b
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated uglify-es@3.3.9: support for ECMAScript is superseded by `uglify-js` as
of v3.13.0
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-
url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-
resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code
should use the URLSearchParams API instead.
npm WARN deprecated acorn-dynamic-import@3.0.0: This is probably built in to whatever tool
you're using. If you still need
it... idk
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019.
Upgrade to chokidar 3 with 15x fewer dependencies

added 470 packages, and audited 471 packages in 26s

11 packages are looking for funding
  run `npm fund` for details

13 high severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible, run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container 09fe87b5ce2b
 ---> 00a1e972be2e
Step 5/8 : COPY index.html .
 ---> 339e78f571d8
Step 6/8 : COPY server.js .
 ---> 6648d87f8cda
Step 7/8 : EXPOSE 3000
 ---> Running in d8a5f7d72934
Removing intermediate container d8a5f7d72934
 ---> 6bf712b47c69
Step 8/8 : CMD ["npm", "start"]
 ---> Running in 5eded8708276
Removing intermediate container 5eded8708276
 ---> fed7423a31ad
Successfully built fed7423a31ad
Successfully tagged ms-frontend:1.0

vagrant@mhoheb:/vagrant/REPO/src$ docker build -t ms-products:1.0 products
Sending build context to Docker daemon  33.01MB
Step 1/7 : FROM node:alpine
 ---> 182dfd1d5db3
Step 2/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> 9d2a3658a4f2
Step 3/7 : COPY package*.json ./
```

```
 ---> 085bf7eaeedf
Step 4/7 : RUN npm i webpack-4.17.1 --no-bin-links
 ---> Running in c3755c5cebbb
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated uglify-es@3.3.9: support for ECMAScript is superseded by `uglify-js` as
of v3.13.0
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-
url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-
resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code
should use the URLSearchParams API instead.
npm WARN deprecated acorn-dynamic-import@3.0.0: This is probably built in to whatever tool
you're using. If you still need
it... idk
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019.
Upgrade to chokidar 3 with 15x fewer dependencies

added 470 packages, and audited 471 packages in 28s

11 packages are looking for funding
  run `npm fund` for details

13 high severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible, run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container c3755c5cebbb
 ---> a735bc196de8
Step 5/7 : COPY server.js .
 ---> 58f683b45386
Step 6/7 : EXPOSE 3001
 ---> Running in ea7348b5ae70
Removing intermediate container ea7348b5ae70
 ---> 66713938a9b8
Step 7/7 : CMD ["npm", "start"]
 ---> Running in b76e67232867
Removing intermediate container b76e67232867
 ---> 0e3083a28490
Successfully built 0e3083a28490
Successfully tagged ms-products:1.0

vagrant@mhoheb:/vagrant/REPO/src$ docker build -t ms-shopping-cart:1.0 shopping-cart
Sending build context to Docker daemon  33.01MB
Step 1/7 : FROM node:alpine
 ---> 182dfd1d5db3
Step 2/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> 9d2a3658a4f2
Step 3/7 : COPY package*.json ./
 ---> 36ddf6fac44f
Step 4/7 : RUN npm i webpack-4.17.1 --no-bin-links
```

```
 ---> Running in d2a703d45de4
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated uglify-es@3.3.9: support for ECMAScript is superseded by `uglify-js` as
of v3.13.0
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-
url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-
resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code
should use the URLSearchParams API instead.
npm WARN deprecated acorn-dynamic-import@3.0.0: This is probably built in to whatever tool
you're using. If you still need
it... idk
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019.
Upgrade to chokidar 3 with 15x fewer dependencies

added 470 packages, and audited 471 packages in 38s

11 packages are looking for funding
  run `npm fund` for details

13 high severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible, run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.5
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.5>
npm notice Run `npm install -g npm@9.6.5` to update!
npm notice
Removing intermediate container d2a703d45de4
 ---> f39a1a195849
Step 5/7 : COPY server.js .
 ---> 3a8df055444d
Step 6/7 : EXPOSE 3002
 ---> Running in ef844883dfe1
Removing intermediate container ef844883dfe1
 ---> 392f97d60f07
Step 7/7 : CMD ["npm", "start"]
 ---> Running in 42850b2fb5f2
Removing intermediate container 42850b2fb5f2
 ---> 4f43de819feb
Successfully built 4f43de819feb
Successfully tagged ms-shopping-cart:1.0

vagrant@mhoheb:/vagrant/REPO/src$ docker images -a
REPOSITORY         TAG       IMAGE ID       CREATED              SIZE
ms-shopping-cart   1.0       4f43de819feb   9 seconds ago        220MB
<none>             <none>    392f97d60f07   10 seconds ago       220MB
<none>             <none>    3a8df055444d   11 seconds ago       220MB
<none>             <none>    f39a1a195849   15 seconds ago       220MB
<none>             <none>    36ddf6fac44f   About a minute ago   180MB
ms-products        1.0       0e3083a28490   2 minutes ago        220MB
<none>             <none>    66713938a9b8   2 minutes ago        220MB
<none>             <none>    58f683b45386   2 minutes ago        220MB
<none>             <none>    a735bc196de8   3 minutes ago        220MB
<none>             <none>    085bf7eaeedf   3 minutes ago        180MB
```

```
ms-frontend        1.0       fed7423a31ad   6 minutes ago        220MB
<none>             <none>    6648d87f8cda   6 minutes ago        220MB
<none>             <none>    6bf712b47c69   6 minutes ago        220MB
<none>             <none>    339e78f571d8   6 minutes ago        220MB
<none>             <none>    00a1e972be2e   6 minutes ago        220MB
<none>             <none>    9d2a3658a4f2   6 minutes ago        179MB
<none>             <none>    b12f08ce38d2   6 minutes ago        180MB
node               alpine    182dfd1d5db3   4 days ago           179MB
hello-world        latest    feb5d9fea6a5   19 months ago        13.3kB

vagrant@mhoheb:/vagrant/REPO/src/shopping-cart$ cd /vagrant/REPO/src/

vagrant@mhoheb:/vagrant/REPO/src$ docker network create products-devops
d5c8b37aaa4d8e06b4b50297a657dcb1387a599b64ad315e8e6d107bcf8ebd07

vagrant@mhoheb:/vagrant/REPO/src$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
d5c8b37aaa4d   products-devops   bridge    local

vagrant@mhoheb:/vagrant/REPO/src$ docker run -d --name frontend --network products-devops
-p 3000:3000 \-e PRODUCTS_SERVICE=products \-e SHOPPING_CART_SERVICE=shopping-cart ms-
frontend:1.0
853b40d79e4f7ff1db18c5505d41b6331e4c54f0a64012f9b80bcea2616c5a2d

vagrant@mhoheb:/vagrant/REPO/src$ docker run -d --name products --network products-devops
-p 3001:3001 ms-products:1.0
588867600b7af6d5f94191841cc4d7fe4e5ee38fbb5c642dee99e6f45f4884c4

vagrant@mhoheb:/vagrant/REPO/src$ docker run -d --name shopping-cart  --network products-
devops -p 3002:3002 ms-shopping-cart:1.0
35a393f84b011f809fd18cd0763b1bc9e14fe55489b9b24f263dc3add1409e02

vagrant@mhoheb:/vagrant/REPO/src$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS
PORTS
  NAMES
35a393f84b01   ms-shopping-cart:1.0   "docker-entrypoint.s…"   8 seconds ago        Up 6
seconds    0.0.0.0:3002->3002/tcp
  shopping-cart
588867600b7a   ms-products:1.0        "docker-entrypoint.s…"   43 seconds ago       Up 41
seconds   0.0.0.0:3001->3001/tcp
  products
853b40d79e4f   ms-frontend:1.0        "docker-entrypoint.s…"   About a minute ago   Up 59
seconds   0.0.0.0:3000->3000/tcp
  frontend

vagrant@mhoheb:/vagrant/REPO/src$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS
PORTS
      NAMES
35a393f84b01   ms-shopping-cart:1.0   "docker-entrypoint.s…"   About a minute ago   Up About
a minute   0.0.0.0:3002->3002/tcp   shopping-cart
588867600b7a   ms-products:1.0        "docker-entrypoint.s…"   2 minutes ago        Up 2
minutes        0.0.0.0:3001->3001/tcp   products
853b40d79e4f   ms-frontend:1.0        "docker-entrypoint.s…"   2 minutes ago        Up 2
minutes        0.0.0.0:3000->3000/tcp   frontend

vagrant@mhoheb:/vagrant/REPO/src$ docker logs 35a393f84b01



> microservice-shopping-cart@1.0.0 start
> node server.js

app listening on port 3002!
vagrant@mhoheb:/vagrant/REPO/src$ docker logs 588867600b7a
```

```
> microservice-products@1.0.0 start
> node server.js

app listening on port 3001!
vagrant@mhoheb:/vagrant/REPO/src$ docker logs 853b40d79e4f

> microservice-frontend@1.0.0 start
> node server.js

app listening on port 3000!
{"products":[{"name":"laptop","price":500},{"name":"monitor","price":100},{"name":"keyboa
rd","price":30}]}
{"shoppingCart":[{"product":"laptop","number":1,"price":500},{"product":"monitor","number
":2,"price":100}]}

vagrant@mhoheb:/vagrant/REPO/src$
```