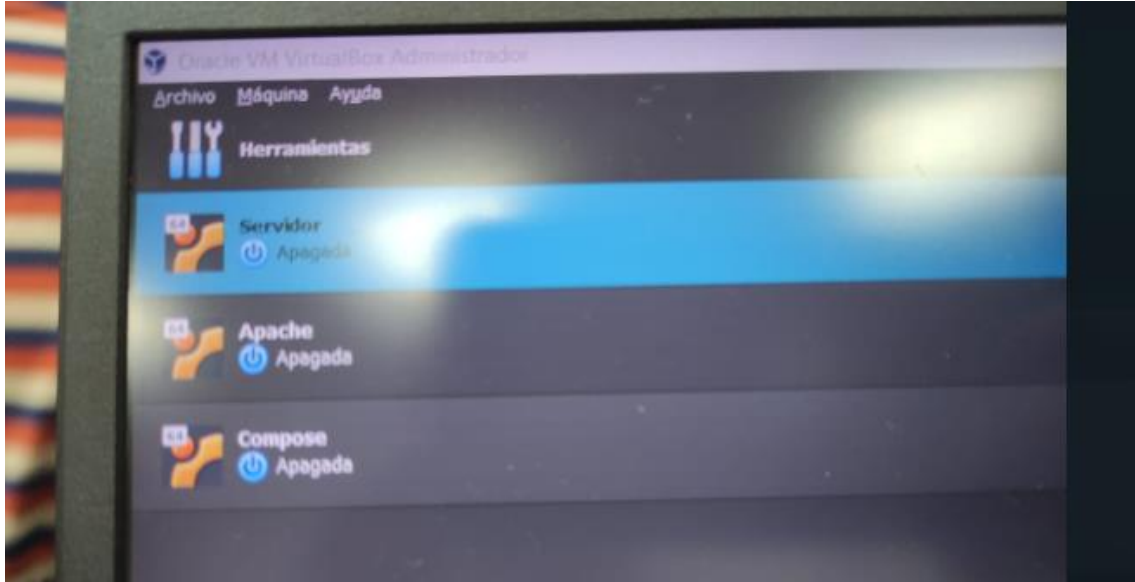


Informe Computación Distribuida

Todo lo realizado, se encuentra en este GitHub: <https://github.com/Miguel-Angel-Nivia/Sistemas-Operativos.git>

Primeramente, para desarrollar este ejercicio, se instaló el virtual box de la página oficial de virtual box: <https://www.virtualbox.org/ticket/20545>, generando los 3 virtuales boxes:



Uno para cada uno de los 3 servicios que tendremos para usar, siendo el principal el de Nginx.

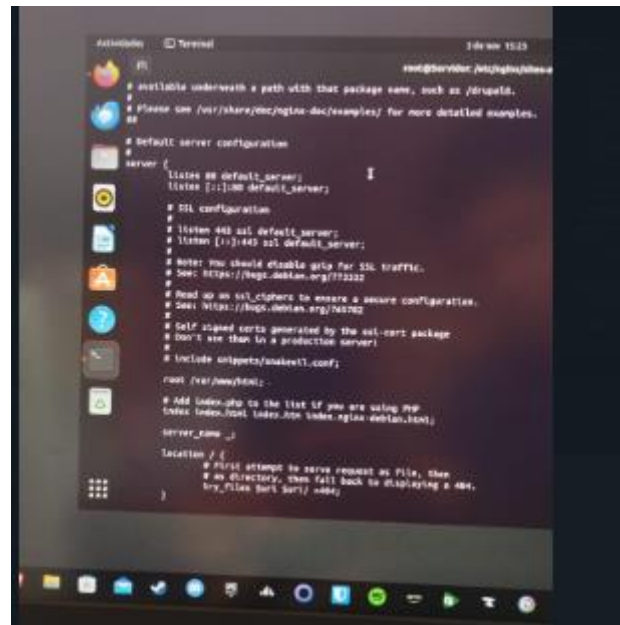
Posteriormente en el asignado se descarga Nginx que sería nuestro primer espacio de contenido de los ejercicios, el cual se realizó los siguientes pasos:

1. Se instaló la ISO requerido a el sistema operativo del PC.
2. Después se instaló el Servidor Nginx

A screenshot of a terminal window. The title bar shows 'Actividades' and 'Terminal'. The terminal content shows the command 'systemctl status nginx' and its output. The output indicates that the nginx service is loaded, active, and running. It also shows the process details for the nginx master and worker processes. The terminal window has a dark background with light-colored text. The prompt is 'root@Servidor: /home/dok'.

Revisando en el estatus que este correcto

3. Después se verificó su dirección para funcionar:



```
root@server:~# cat /etc/nginx/nginx.conf
# available under a path with that package name, such as /usr/share/nginx.
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/772232
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/746702
    #
    # Self signed certs generated by the ssl-cert package
    # don't see them in a production server)
    #
    # Include snippets/*.conf;
    include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```



```
root@server:~# cat /etc/nginx/sites-enabled/default
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

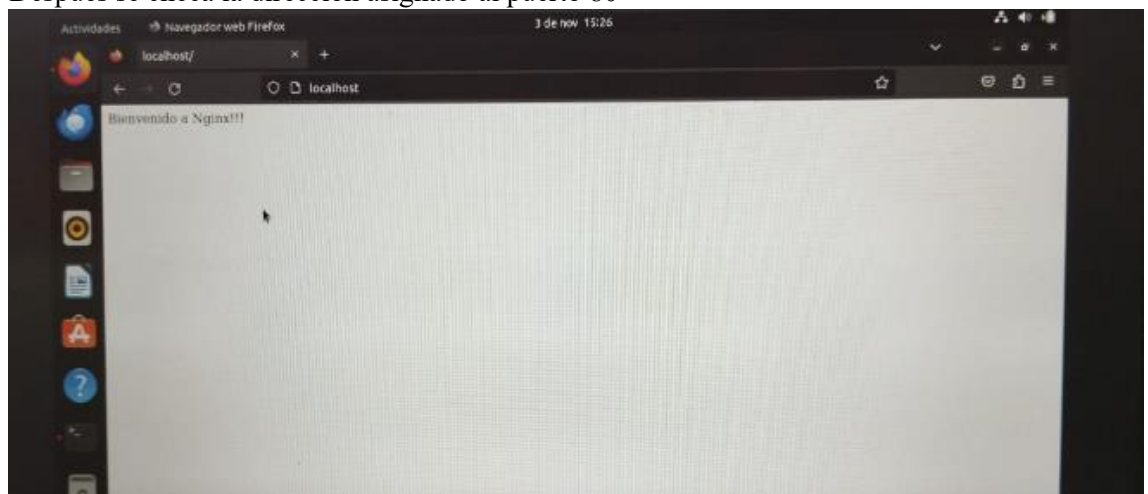
# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    # With php-cgi (or other tcp sockets):
    fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
location ~ /\.ht {
    deny all;
}

# Virtual host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.

#server {
#    listen 80;
#    listen [::]:80;
```

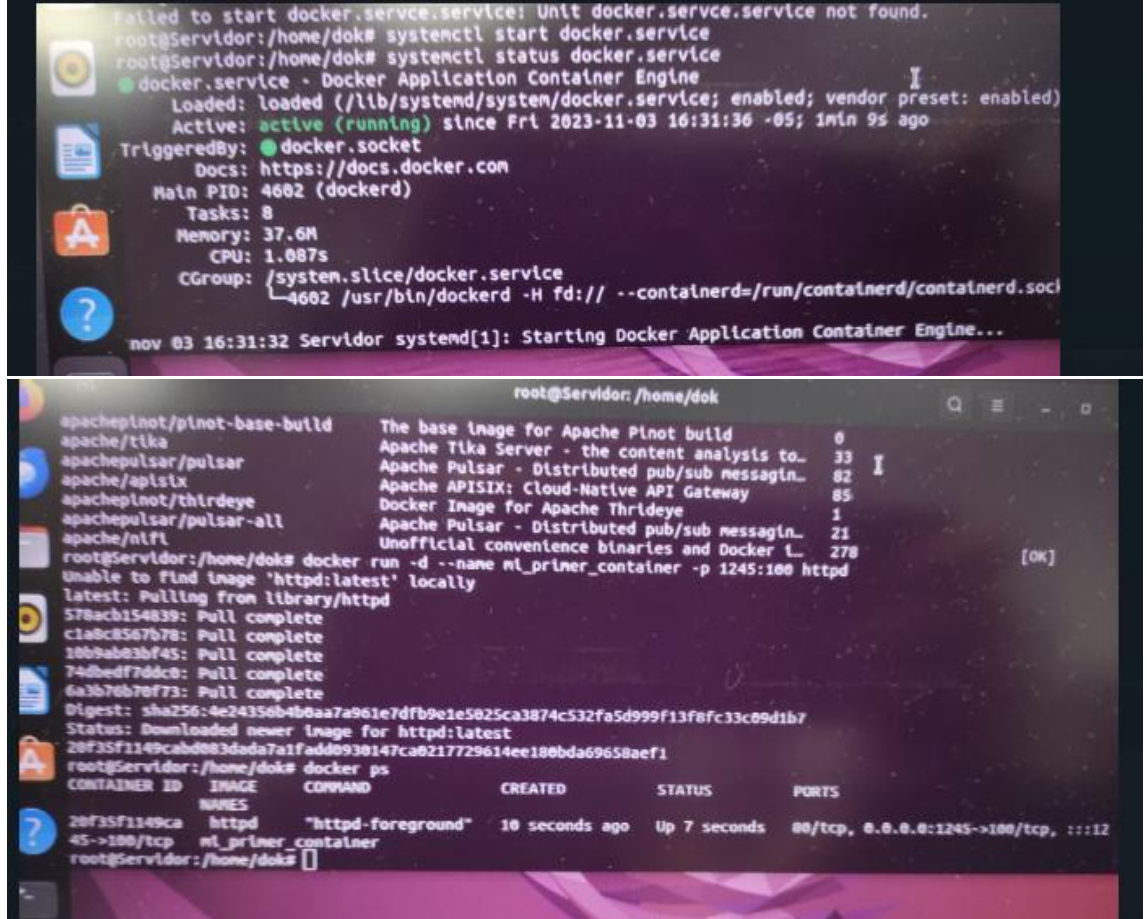
4. Después se checa la dirección asignado al puerto 80



5. Ya con esto solo faltaba cambiar los puertos de red para hacer puente de este con los otros servicios.

Después de creo el segundo servicio de Apache en el virtual box asignado, así siguiendo los siguientes pasos:

1. Primero se descargo el servicio Apache en la ISO del sistema.



The first screenshot shows the Docker service status in a terminal window. The output indicates that the service is loaded and active (running) since Friday, 2023-11-03 at 16:31:36 -05. The main PID is 4602 (dockerd), and it has 8 tasks, 37.6M memory, and 1.087s CPU usage. The second screenshot shows the process of pulling the 'httpd:latest' image from Docker Hub. It lists the image's SHA256 digest and shows the download progress. Finally, it displays a table of running containers, including the 'httpd' container which is running on port 80/tcp.

```
failed to start docker.service: Unit docker.service.service not found.
root@Servidor:/home/dok# systemctl start docker.service
root@Servidor:/home/dok# systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-11-03 16:31:36 -05; 1min 9s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 4602 (dockerd)
         Tasks: 8
        Memory: 37.6M
          CPU: 1.087s
       CGroup: /system.slice/docker.service
               └─4602 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

nov 03 16:31:32 Servidor systemd[1]: Starting Docker Application Container Engine...

root@Servidor:/home/dok#
apache/pinot/pinot-base-build The base image for Apache Pinot build 0
apache/tika Apache Tika Server - the content analysis to... 33
apache/pulsar/pulsar Apache Pulsar - Distributed pub/sub messagin... 82
apache/apisix Apache APISIX: Cloud-Native API Gateway 85
apache/pinot/thrdeye Docker Image for Apache Thrdeye 1
apache/pulsar/pulsar-all Apache Pulsar - Distributed pub/sub messagin... 21
apache/nifi Unofficial convenience binaries and Docker L... 278
root@Servidor:/home/doks docker run -d --name ni_primer_container -p 1245:100 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
578acb154839: Pull complete
c1a8c8567b78: Pull complete
10b9ab63bf45: Pull complete
74dbedf7ddc8: Pull complete
6a3b76b7bf73: Pull complete
Digest: sha256:4e24356b4b0aa7a961e7dfb9e1e5025ca3874c532fa5d999f13f8fc33c09d1b7
Status: Downloaded newer image for httpd:latest
20f35f1149cabd083dada7a1fadd0930147ca0217729614ee180bda69658aef1
root@Servidor:/home/doks# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
20f35f1149ca   httpd     "httpd-foreground"      10 seconds ago Up 7 seconds  80/tcp, 0.0.0.0:1245->100/tcp, :::12
45->100/tcp    ni_primer_container
root@Servidor:/home/doks#
```

2. Después se verifica su estatus, siendo correcto.
3. Posteriormente se le asigna el puerto deseado que seria el 8080 para este servicio y se verifica en el buscador si está funcionando adecuadamente.



- Después de esto se activo el puente de los puertos de red para que se conecte con los otros.

Posteriormente se realiza la última instalación, referenciado con docker compose, en su virtual box asignada:

- Primeramente, se debe descargar docker.



```
failed to start docker.service.service: Unit docker.service.service not found.
root@Servidor:/home/dok# systemctl start docker.service
root@Servidor:/home/dok# systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-11-03 16:31:36 -05; 1min 9s ago
     TriggeredBy: ● docker.socket
        Docs: https://docs.docker.com
       Main PID: 4602 (dockerd)
         Tasks: 8
        Memory: 37.6M
         CPU: 1.087s
        CGroup: /system.slice/docker.service
                └─4602 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

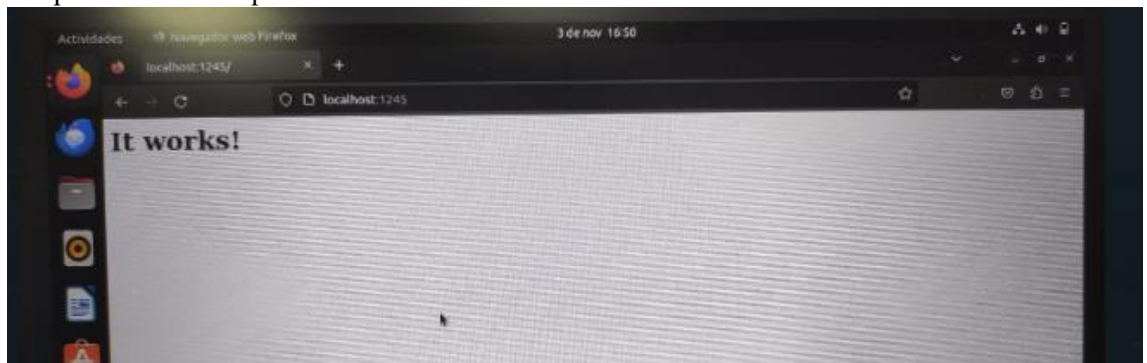
nov 03 16:31:32 Servidor systemd[1]: Starting Docker Application Container Engine...
```

- Verificamos que el estatus del servicio este activo.
- Después se le asigna el puerto deseado, en este caso será 1245.



```
apachepinot/pinot-base-build The base image for Apache Pinot build 0
apachetika Apache Tika Server - the content analysis to... 33
apachepulsar/pulsar Apache Pulsar - Distributed pub/sub messagin... 82
apachepulsar/pulsar-all Apache APISIX: Cloud-Native API Gateway 85
apachepulsar/pulsar-all Docker Image for Apache Thrdeye 1
apachepulsar/pulsar-all Apache Pulsar - Distributed pub/sub messagin... 21
apachepulsar/pulsar-all Unofficial convenience binaries and Docker I... 278
root@Servidor:/home/dok# docker run -d --name nl_primer_container -p 1245:100 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
578acb154839: Pull complete
c1a8c8567b78: Pull complete
18b9ab93bf45: Pull complete
74dbedf7ddc0: Pull complete
6a3b78b78f73: Pull complete
Digest: sha256:4e24356b4b6aa7a961e7dfb9e1e5825ca3874c532fa5d999f13f8fc33c89d1b7
Status: Downloaded newer image for httpd:latest
28f35f1149cabd883dada7a1fad0930147ca0217729614ee188bda69658aef1
root@Servidor:/home/dok# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
28f35f1149ca   httpd      "httpd-foreground"      10 seconds ago Up 7 seconds   80/tcp, 0.0.0.0:1245->100/tcp, :::12
45->100/tcp    nl_primer_container
root@Servidor:/home/dok#
```

- Después se verifica que funcione esto en el buscador.



- Finalmente se activa el puente entre puertos para así poder ya tener la última maquina conectada.

Nota: El proceso del paso a paso de todo esto, se realizo gracias a la ayuda de unos videos, los cuales están referenciados al final del texto.

Ya con todo esto, se tiene las tres maquinas hechas, y ahora desde la de Nginx, se verifica que, al cambiar el puerto, este cambiara de página entre Apache, Nginx y compose, pero antes se debe modificar el archivo de conexión de las IP's entre cada uno:

```
1  server {
2      listen 80;
3
4      location /apache/ {
5          proxy_pass http://apache_server:3000/;
6          proxy_redirect    off;
7          proxy_set_header  Host $host;
8          proxy_set_header  X-Real-IP $remote_addr;
9          proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
10         proxy_set_header  X-Forwarded-Host $server_name;
11         client_max_body_size 100M;
12     }
13
14     location /compose/ {
15         proxy_pass http://compose_server:3000/;
16         proxy_redirect    off;
17         proxy_set_header  Host $host;
18         proxy_set_header  X-Real-IP $remote_addr;
19         proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
20         proxy_set_header  X-Forwarded-Host $server_name;
21         client_max_body_size 100M;
22     }
23
24 }
```

En base al código del profesor que nos dio para esto, solamente se cambio el numero de puerto, en el cual, el de apache es con 8080, el de compose con 1245 y el de Nginx con el de 80, con esto desde el virtual box de Nginx se puede interactuar entre cada uno de estos 3 servicios haciendo:

LocalHost:xxx -> en el cual xxx significa el puerto deseado a realizar.

Nota: El segundo laboratorio de esta entrega por temas de tiempo frente a parciales u otros proyectos y dificultad de este mismo laboratorio, se decidió a no realizarlo ya que, al buscar información, no se encontró mucha de esta, más que teórico y como desconocemos de esto, no lo podemos hacer.

REFERENCIAS:

- Proceso Descarga VirtualBox – Compose:
<https://www.youtube.com/watch?v=BSgAhEKoDuw>
- Proceso Descarga VirtualBox – Nginx:
<https://www.youtube.com/watch?v=fF2VZGOBWpc>
- Proceso Descarga VirtualBox – Apache:
<https://www.youtube.com/watch?v=Az3SRuHlqDM>