



Universidad Privada

Domingo Savio

Facultad de Ingeniería

“PROCESOS REALIZADOS”

- Juan Carlos Ramírez Pedraza



Docente: Jared Lopez

Materia: Tecnología Web I

2024

MAQUINAS VIRTUALES

LAS SIGUIENTES MAQUINAS VIRTUALES SERAN CLONADAS A PARTIR DE MINIOS-BASE

MINIOS-BASE → 1 PROCESADOR / 1.5GB RAM / 192.168.56.202

MINIOS-DNS → 1 PROCESADOR / 1GB RAM / 192.168.56.10 / dns.ramirez.lan

MINIOS-DK → 1 PROCESADOR / 1.5GB RAM / 192.168.56.250 / dk.ramirez.lan

MINIOS-CMS → 1 PROCESADOR / 1.5GB RAM / 192.168.56.251 / cms.ramirez.lan

MINIOS-BASE ADAPTADORES

ADAPTADOR 1 → NAT

ADAPTADOR 2 → ADAPT. SOLO ANFITRION(192.168.56.202)

HABILITACION SSH

Comando `sudo systemctl enable ssh` para que el servicio ssh se active cada que el sistema operativo se inicia

Comando `sudo service ssh start` para levantar el servicio ssh

Comando `sudo service ssh status` para ver el estado del servicio ssh

ACTUALIZACION DE PAQUETES

Comando `sudo apt update` para actualizar la lista de paquetes disponibles en los repositorios

Comando `sudo apt upgrade -y` para instalar las actualizaciones disponibles para los paquetes instalados en tu sistema

APT FAST

Comando `sudo add-apt-repository ppa:apt-fast/stable`

Comando `sudo apt-get -y install apt-fast`

Comando `sudo apt-fast upgrade -y`

Reiniciamos la maquina

GIT

Comando `sudo apt-fast install git` para instalar git

Clonamos el repositorio con los archivos .yaml con el comando `git clone https://github.com/docentegit/202405TecnologiaWebI.git`

RESOLUCION DE DOMINIOS

Editamos el archivo `resolv.conf` `sudo mousepad /etc/resolv.conf`

Dentro del archivo anadimos `nameserver 192.168.56.10` y borramos los que estaban

MINIOS-DNS

Clonamos la maquina virtual MINIOS-Base
ADAPTADORES

ADAPTADOR 1 → NAT

ADAPTADOR 2 → ADAPT. SOLO ANFITRION(192.168.56.10)

PI-HOLE

copiamos el archivo docker-compose.pihole.yaml

Comando `cp 202405TecnologiaWebI/docker-compose.pihole.yaml docker/dns/docker-compose.yaml`
para copiar el archivo a la carpeta nueva creada docker/dns

Ejecutar el comando para levantar el contenedor

`cd docker/dns`

`docker compose up -d`

Tendremos que eliminar el cache de dns con el comando `sudo systemctl disable systemd-resolved`

Luego editamos el archivo `/etc/NetworkManager/NetworkManager.conf` y hacemos estas correcciones:
`dns=dnsmasq` cambiar por `dns=default`

Reiniciamos el sistema `sudo shutdown -r now`

ADMIN PI-HOLE

Ingresamos a `http://192.168.56.10/admin` (pass:123123)

ADICIONAR UN REGISTRO DE DNS

INGRESAR Local DNS → DNS Records

Dominio: `dns.ramirez.lan` → `192.168.56.10`

Adicionarlo

Dominio: `dk.ramirez.lan` → `192.168.56.250`

Adicionarlo

Dominio: `cms.ramirez.lan` → `192.168.56.251`

Adicionarlo

Dominio: `wp.ramirez.lan` → `192.168.56.252`

Adicionarlo

VERIFICACION

INSTALAR: `sudo apt-fast install dnsutils`

En la terminal ejecutar `dig @192.168.56.10 dns.ramirez.lan`

VERIFICACION EN OTRA MAQUINA VIRTUAL (MINIOS-DOCKER-BASE)

Ingresar a la url: `dns.ramirez.lan`

Si presentamos problema revisar el archivo `/etc/resolv.conf` que apunte a `192.168.56.10`

ARCHIVO YML

version: "3"

More info at <https://github.com/pi-hole/docker-pi-hole/> and [https://docs.pi-hole.net/ services](https://docs.pi-hole.net/services):

pihole:

container_name: pihole

image: pihole/pihole:latest

For DHCP it is recommended to remove these ports and instead add: network_mode: "host"

ports:

- "53:53/tcp"

- "53:53/udp"

- "67:67/udp" # Only required if you are using Pi-hole as your DHCP server

- "80:80/tcp"

environment:

TZ: 'America/La_Paz'

WEBPASSWORD: '123123'

Volumes store your data between container upgrades

volumes:

- './etc-pihole:/etc/pihole'

- './etc-dnsmasq.d:/etc/dnsmasq.d'

<https://github.com/pi-hole/docker-pi-hole#note-on-capabilities>

cap_add:

- NET_ADMIN # Required if you are using Pi-hole as your DHCP server, else not needed

restart: unless-stopped

MINIOS-DOKUWIKI

Clonamos la maquina virtual MINIOS-Base

ADAPTADORES

ADAPTADOR 1 → NAT

ADAPTADOR 2 → ADAPT. SOLO ANFITRION(192.168.56.250)

CONTENEDOR

copiamos el archivo docker-compose.dokuwiki.yml

Comando cp 202405TecnologiaWebI/docker-compose.dokuwiki.yml docker/dokuwiki/docker-compose.yml para copiar el archivo a la carpeta nueva creada docker/dokuwiki

Ejecutar el comando para levantar el contenedor

cd docker/dokuwiki

docker compose up -d

Luego entrar a la url dk.ramirez.lan y completar la instalacion

VERIFICACION

Entrar a la url dk.ramirez.lan

VERIFICACION EN OTRA MAQUINA VIRTUAL (MINIOS-DOCKER-BASE)

Ingresa a la url: dk.ramirez.lan

Si presentamos problema revisar el archivo /etc/resolv.conf que apunte a 192.168.56.10

ARCHIVO YML

version: "2.1" services:

dokuwiki:

image: lscr.io/linuxserver/dokuwiki:latest

container_name: dokuwiki

environment:

- PUID=1000

- PGID=1000

- TZ=Bolivia/La_Paz

volumes:

- ./config:/config

ports:

- 80:80

- 443:443 #optional

restart: unless-stopped

MINIOS-CMSIMPLE

Clonamos la maquina virtual MINIOS-Base

ADAPTADORES

ADAPTADOR 1 → NAT

ADAPTADOR 2 → ADAPT. SOLO ANFITRION(192.168.56.251)

CONTENEDOR

Clonamos el repositorio con git clone <https://github.com/risingisland/GetSimpleCMS-CE-3.3.20.git>

Copiamos el contenido a nuestra carpeta docker/cms

Dentro de la carpeta docker/cms abrir una terminal ejecutamos el comando sudo chown -R live:live html/ para cambiar de propietario la carpeta html y luego para darle permisos de escritura ejecutamos sudo chmod -R 777 html/

Luego ya podremos terminar nuestra instalacion

VERIFICACION

Entrar a la url dk.ramirez.lan

VERIFICACION EN OTRA MAQUINA VIRTUAL (MINIOS-DOCKER-BASE)

Ingresar a la url: dk.ramirez.lan

Si presentamos problema revisar el archivo /etc/resolv.conf que apunte a 192.168.56.10

ARCHIVO YML

version: '3.8'

services:

Apache and PHP service

web:

image: php:8-apache

container_name: php-apache-container

ports:

- "80:80"

volumes:

- ./html:/var/www/html

Define networks

networks:

my-network:

MINIOS-WORDPRESS-MARIADB-PHPMYADMIN

Clonamos la maquina virtual MINIOS-Base

ADAPTADORES

ADAPTADOR 1 → NAT

ADAPTADOR 2 → ADAPT. SOLO ANFITRION(192.168.56.252)

CONTENEDOR

copiamos el archivo docker-compose.phpmyadmin.yml y docker-compos.wordpressMariadb.yml

Comando cp 202405TecnologiaWebI/docker-compose.phpmyadmin.yml docker/wordpress/docker-compose.phpmyadmin.yml para copiar el archivo a la carpeta nueva creada docker/wordpress

Comando cp 202405TecnologiaWebI/docker-compos.wordpressMariadb.yml

docker/wordpress/docker-compos.wordpressMariadb.yml para copiar el archivo a la carpeta nueva creada docker/wordpress

Creamos un documento nuevo vacio llamado docker-compose.yml y fusionamos los dos archivos yml que podremos ver en esta misma pagina en la parte de ARCHIVO YML

Ejecutar el comando para levantar el contenedor

cd docker/wordpress

docker compose up -d

VERIFICACION

Entrar a la url wp.ramirez.lan para verificar el wordpress

Entrar a la url wp.ramirez.lan:8082 para verificar el phpmyadmin

VERIFICACION EN OTRA MAQUINA VIRTUAL (MINIOS-DOCKER-BASE)

Ingresar a la url: wp.ramirez.lan para wordpress y wp.ramirez.lan:8082 para phpmyadmin
Si presentamos problema revisar el archivo /etc/resolv.conf que apunte a 192.168.56.10

ARCHIVO YML

version: '3.1'

services:

wordpress:

image: wordpress

restart: always

ports:

- 80:80

- 443:443 # HTTPS

environment:

WORDPRESS_DB_HOST: db

WORDPRESS_DB_USER: exampleuser

WORDPRESS_DB_PASSWORD: examplepass

WORDPRESS_DB_NAME: exampledb

volumes:

- ./wp:/wp

db:

image: mariadb:latest

container_name: db

restart: always

environment:

MYSQL_ROOT_PASSWORD: 123456

MYSQL_DATABASE: exampledb

MYSQL_USER: exampleuser

MYSQL_PASSWORD: examplepass

MYSQL_RANDOM_ROOT_PASSWORD: '1'

ports:

- "3306:3306"

volumes:

- db:/var/lib/mysql

phpmyadmin:

image: phpmyadmin

container_name: pma

links:

- db

environment:

PMA_HOST: db

PMA_PORT: 3306

PMA_USER: exampleuser

PMA_PASSWORD: examplepass

PMA_ARBITRARY: 1

```
restart: always
ports:
  - 8082:80
```

```
volumes:
  db:
```

PING.SH

Este archivo nos sirve para hacerle ping a nuestras 4 maquinas virtuales desde un solo comando

En la carpeta live creamos un documento llamado ping.sh

```
ping dns.ramirez.lan -c 4
ping dk.ramirez.lan -c 4
ping cms.ramirez.lan -c 4
ping wp.ramirez.lan -c 4
```

Para ejecutarlo bash ping.sh