

Sprint 1 Cloud

INFRAESTRUCTURA I ARQUITECTURA

Rafat Alam
Judit Cerdà
Irene García
Roger Pérez
Gerard Romero

Índex

Docker	2
Capa Fog	3
Adreces IPs	4
Servidor Debian 10	4
Servidor web	5
Servidor Base de dades	10
Servidor DNS	19
Servidor DHCP	33
Burndown	41
Webgrafia	41

Docker

En aquest sprint encara no hem utilitzat Docker, però ens hem començat a informar, de cara als següents sprints.

- Què és Docker?

Docker és un software que consisteix en crear contenidors lleugers (màxima portabilitat), perquè es puguin utilitzar aplicacions en qualsevol màquina independentment del sistema operatiu on s'estiguin executant.

El fet d'utilitzar docker, fa que no importi el sistema base on s'esta executant l'aplicació. No és com una màquina virtual, la quan s'ha d'instal·lar un sistema operatiu i les aplicacions que està sobre aquest sistema, utilitzen les seves característiques. Docker és més com la MV de Java, s'executa sobre el sistema operatiu amfitrió i permet que les aplicacions que s'importin s'utilitzin sobre aquest.

- Com funciona Docker?

La tecnologia Docker usa el kernel de Linux i les funcions d'aquest, com Cgroups i namespaces, per a segregar els processos, de manera que puguin executar-se de manera independent. El propòsit dels contenidors és aquesta independència: la capacitat d'executar diversos processos i aplicacions per separat per a fer un millor ús de la seva infraestructura i, al mateix temps, conservar la seguretat que tindria amb sistemes separats.

Les eines del contenidor, com Docker, ofereixen un model d'implementació basat en imatges. Això permet compartir una aplicació, o un conjunt de serveis, amb totes les seves dependències en diversos entorns. Docker també automatitza la implementació de l'aplicació (o conjunts combinats de processos que constitueixen una aplicació) en aquest entorn de contenidors.

Aquestes eines desenvolupades a partir dels contenidors de Linux, la qual cosa fa a Docker fàcil d'usar i únic, atorguen als usuaris un accés sense precedents a les aplicacions, la capacitat d'implementar ràpidament i control sobre les versions i la seva distribució.

Capa Fog

- **Què és?**

Centres de dades descentralitzades situat en un nivell inferior al cloud, on emmagatzemen petites informacions on després es poden centralitzar en el cloud.

- **Fog com una extensió del cloud:**

En contraposició amb la infraestructura cloud tradicional, basada en grans centres de dades que centralitzen el poder computacional, altres paradigmes com Fog Computing proposen la distribució d'aquesta capacitat de còmput cap als extrems de la xarxa. Aquest paradigma busca solucionar els problemes de comunicació de dades entre els dispositius generadors i consumidors dels mateixos en acostar els centres de processament i anàlisi de dades cap a ells, reduint d'aquesta forma la latència i l'ús de la infraestructura de xarxa, amb el que es millora l'experiència de l'usuari i el rendiment de les aplicacions que fan ús de recursos computacionals externs. Aquest concepte va ser originalment proposat per Cisco com una arquitectura cloud distribuïda i allunyada dels datacenters centralitzats mitjançant un gran nombre de nodes distribuïts per la xarxa. En oferir recursos cloud distribuïts fins a la vora de la xarxa, tot el trànsit generat per l'accés als serveis cloud no necessita travessar tota la xarxa per a accedir als recursos.

No obstant això aquesta nova arquitectura no es planteja com un substitut de les infraestructures Cloud tradicionals, si no com una extensió d'aquestes, la potència de còmput, la versatilitat i les capacitats de compartició de recursos que ofereix un centre de dades cloud continuarà sent imprescindible per a moltes aplicacions. Però aquelles aplicacions que requereixin del processament de dades en temps real (real-estafi) que permetin respondre amb immediatesa a determinats esdeveniments, o que per les seves característiques demandin un gran consum d'amplada de banda, podran treure partit de la proximitat i baixa latència que ofereix un paradigma com Fog Computing per a part de les seves necessitats o la totalitat d'aquestes. A més, aquells serveis i infraestructures crítiques es beneficiaran de la seguretat i resiliència que ofereix, de manera natural, una arquitectura distribuïda. En el següent enllaç trobem informació més detallada sobre les diferents capes del "fog to cloud": <https://www.gradiant.org/blog/edge-fog-computing-cloud/>

Adreces IPs

Per fer una primera implementació de tots els servidors intercomunicats, hem fet servir les següents IPs:

xarxa = 192.168.1.0/24 màscara de xarxa = 255.255.255.0 broadcast
192.168.1.255

Router principal	→ 192.168.1.1
DHCP	→ 192.168.1.2
DNS	→ 192.168.1.3
Servidor Web	→ 192.168.1.4
Servidor APP	→ 192.168.1.5
Servidor BDD	→ 192.168.1.6
Router F2C	→ 192.168.1.7

Servidor Debian 10

- Instal·lació Debian 10

La primera opció és si volem una instal·lació gràfica o una normal, nosaltres hi escollirem una instal·lació normal. Sortirà una llista d'idiomes on escollirem espanyol o el català segons preferència.

Ara escollirem la nostra regió (localització), escollim “Europa” i “Espanya”.

En la següent pantalla hi haurem d'escollir distribució de teclat. Hem de seleccionar la distribució espanyola si no hi ha preferència amb una altra distribució.

A continuació començarà a instal·lar la base del sistema operatiu i, a la següent opció hi haurem d'introduir:

- el nom del host.

- la contrasenya del root (toor) i repetir-la per seguretat.

També el nom complet del nou usuari, el seu compte (administrador), contrasenya (rodartsinimda) i repetir-la per seguretat. Hi escollirem la zona horària (Madrid o Península).

La partició de discs haurem de seleccionar guiada (“Guided - use entire disk”), ja que no ens interessa fer particions personalitzades ni un mànager de volum lògic.

En la següent pantalla hi trobarem el disc que volem instal·lar, escollim sda (que hauria de ser l'únic), i seleccionarem la opció de montar-ho tot a la mateixa partició.

Ara només faltaria assegurar-nos de què els canvis són correctes i confirmar-los. Després d'instal·lar la majoria del sistema operatiu ens sortirà la confirmació de si volem escanejar un altre CD (no), on som i d'on volem descarregar arxius (només hi haurem d'acceptar tot sense afegir res ni quan ens demana una URL).

Durant la següent instal·lació ens hauran preguntant diverses coses: les survey , el software extra a instal·lar (res excepte les “Utilidades estándar del sistema”, ja que hi treballarem sense entorn gràfic, es deselecciona amb espai).

Ara seleccionem que volem GRUB al dispositiu sda (on hi és instal·lat el SO) i ja hi hauríem finalitzat.

Servidor web

Un servidor web s'encarrega de gestionar protocols de transferència d'hipertext HTTP o el segur HTTPS

A l'hora d'implementar un servidor web, s'han de tenir en compte diferents qüestions que veurem a continuació. Però sí que hi han 2 coses que s'hi hauran de tenir en compte. Si en la xarxa es disposa d'un firewall, s'haurà de permetre les connexions pel port 80. A més, per fer més còmode la connexió al nostre servidor, s'haurà de tenir un nom de domini, que s'implementarà amb el DNS.

- Informació de plataformes que donen suport:

Llenguatges que s'utilitzaran per la creació de la pàgina web:

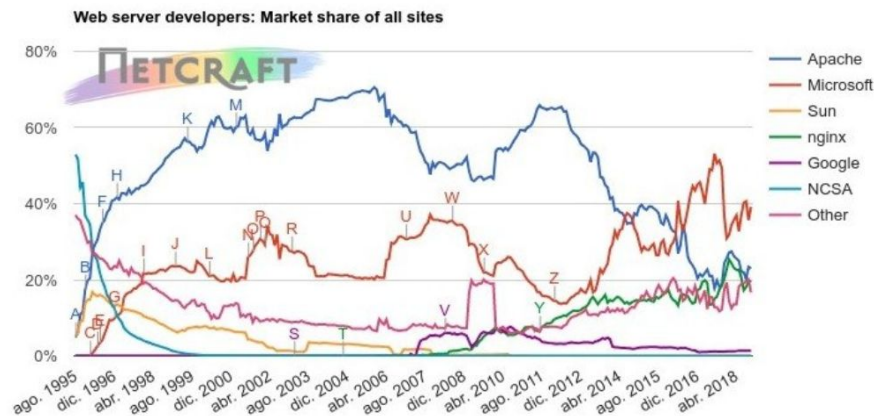
- PHP
- Laravel

Optarem per utilitzar software lliure i de codi obert, dintre d'aquestes opcions podem observar les més utilitzades en el mercat:

-Apache Server: Dintre dels servidors Linux és el líder, és multiplataforma. Té un rendiment molt baix.

-NGINX: Servidor web/proxy: té un alt rendiment. La configuració és simple i permet integrar-se nativament amb quasi qualsevol tecnologia i llenguatge. Consumeix pocs recursos quan ha de gestionar moltes visites. Bona alternativa a Apache.

-Lighttpd: Abans de NGINX, era una bona alternativa a apache, ja que era millor a l'hora de donar suport a múltiples usuaris. Consumeix poca CPU i RAM



- Decidir la plataforma i dir perquè:

Nosaltres utilitzarem Apache, és el que actualment s'utilitza més en el mercat, per sota de Microsoft IIS, al ser compatible amb qualsevol SO, es pot migrar en qualsevol moment i també l'hem escollit perquè els nostres desenvolupadors és el que coneixen.

Apache incorpora un conjunt de mòduls externs per suportar diferents llenguatges com: Perl, PHP, Python, Ruby ...

- Descripció requisits mínims i aconsellables del hardware

Requirements mínims per instal·lar Apache:

- Processador: Pentium
- Memòria RAM: 64 MB
- Sistema Operatiu: Microsoft Windows, GNU/Linux
- Mida de la instal·lació: 50 MB

Qüestions que hem de tenir en compte per saber el hardware necessari.

Quina càrrega d'usuaris al dia rebrà?

Quin tipus de pàgina web carregarà, simple o complexa?, estàtica o dinàmica?

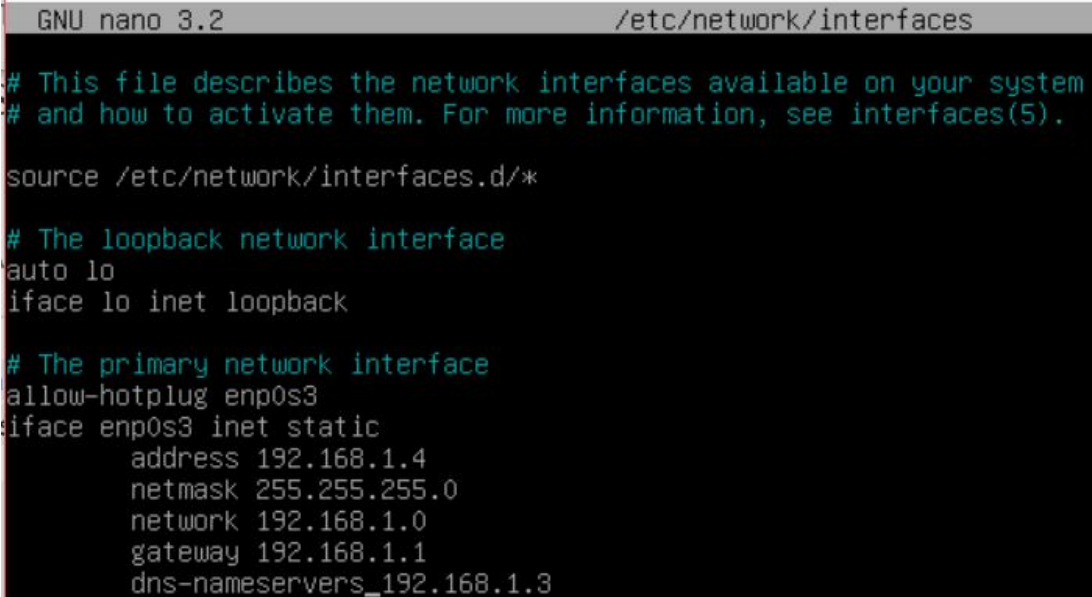
En el projecte que estem desenvolupant, hem de pensar que el nostre servidor rebrà milers de peticions dels usuaris per consultar la pàgina web, una altra part de gestoria.

Per això, necessitem un hardware resistent a aquestes múltiples peticions que rebrà. Per posar una RAM raonable 4GB per poder donar suport a les peticions i una CPU amb 1 core.

A part, hem de pensar que només tindrem allotjada la pàgina web en aquest servidor, tot l'espai del disc dur restant després de les instal·lacions serà per allotjar tota l'estructura i informació de la pàgina web. Hem pensat en posar de moment un disc de 16 GB.

Tots els requisits anteriors s'aniran ampliant sota demanda.

- **Configurem la xarxa**



```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.1.4
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    dns-nameservers 192.168.1.3
```

- **Instal·lar software necessari**

1. Actualitzem el repositori de paquets:
apt-get update
2. Actualitzem els paquets necessaris:
apt-get update
3. Instal·lem Apache:
apt-get install apache2
4. Comprovem el estat:
systemctl status apache2

```

• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2020-03-24 17:59:10 CET; 6min ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 3059 (apache2)
    Tasks: 55 (limit: 1150)
  Memory: 9.0M
  CGroup: /system.slice/apache2.service
          └─3059 /usr/sbin/apache2 -k start
            └─3061 /usr/sbin/apache2 -k start
              └─3062 /usr/sbin/apache2 -k start

mar 24 17:59:10 seax systemd[1]: Starting The Apache HTTP Server...
mar 24 17:59:10 seax systemd[1]: Started The Apache HTTP Server.

```

5. Instal·lar php:
apt-get install php
6. Instal·lem els mòduls per connectar php amb apache:
apt-get install libapache2-mod-php
7. Per comprovar que s'ha carregat el mòdul:
ls /etc/apache2/mods-enabled/

```


access_compat.load  authz_core.load  deflate.load  mime.load  php7.3.load  status.load
alias.conf          authz_host.load  dir.conf     mpm_prefork.conf  reqtimeout.conf
alias.load          authz_user.load  dir.load     mpm_prefork.load  reqtimeout.load
auth_basic.load     autoindex.conf  env.load     negotiation.conf  setenvif.conf
authn_core.load     autoindex.load  filter.load  negotiation.load  setenvif.load
authn_file.load     deflate.conf     mime.conf    php7.3.conf       status.conf

```

- Proves
 - Probar que funciona apache2

← → ↺ 192.168.1.4

Aplicaciones Campus Virtual UP...



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

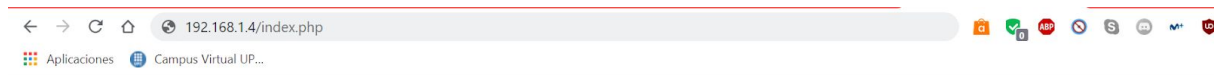
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.Load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf

```


- Probar que funciona php7

Creem un index.php per veure el seu funcionament:

```
GNU nano 3.2 /var/www/html/index.php
<?php phpinfo(); ?>
```



PHP Version 7.3.14-1~deb10u1	
System	Linux seax 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64
Build Date	Feb 16 2020 15:07:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini

- Configurar el servidor d'apache.

Aquest servidor té els següents fitxers de configuració:

- Arxiu d'allotjament virtual:

/etc/apache2/sites-available/000-default.conf

Si fos necessari allotjat més d'una pàgina web, s'hauria de tocar aquest arxiu.

- Arxius de configuració:

Directori on estan tots aquest arxius: /etc/apache2/

★ apache2.conf: arxiu principal de configuració.

★ ports.conf: especifica els ports pels quals escoltarà apache.

Altres directoris de configuració dintre del directori principal:

★ sites-available/: s'allotgen els diferents llocs virtuals.

★ sites-enabled/: s'allotgen els diferents llocs virtuals habilitats.

★ conf-available/ i conf-enabled/: tenen relació amb els anteriors, però aquí es guarden els fitxers de configuració.

★ mods-available/ i mods-enabled/: tenen els mòduls.

- Configurar el mòdul de php:

Els fitxer de configuració de php estan en el directori /etc/php/7.3/ i el fitxer principal és el /etc/php/7.3/apache2/php.ini, tot el que es necessiti configurar respecte a php s'ha d'anar aquí.

- **Com utilitzar Apache:**

El directori on s'han d'allotjar les pàgines web és el: /var/www/html, es podrien allotjar diferents pàgines.

Servidor Base de dades

Una base de dades és, com el nom indica, un lloc on guardar dades que tinguin relació entre elles per després utilitzar-les.

- **Informació de plataformes que donen suport:**

En aquest cas, utilitzarem MongoDB com a gestor de la base de dades perquè és un requisit tècnic del projecte.

Es recomana per a MongoDB:

- Amazon Linux 2
- Debian 9 and 10
- RHEL / CentOS 6, 7, and 8
- SLES 12 and 15
- Ubuntu LTS 16.04 and 18.04
- Windows Server 2016 and 2019

Utilitzarem MongoDB sobre Debian 10 per la base de dades, ja que és altament recomanat i perquè Debian 10 ens proporciona seguretat i robustesa i a més aquest SO té molts pocs requisits mínims.

A més, MongoDB és de codi obert, escalable i d'alt rendiment. Per gestionar les dades d'un aeroport, lloc que sol estar molt concorrit i per tant, té molts usuaris, ens va perfecte.

- **Descripció requisits mínims i aconsellables del hardware**

MongoDB no té com a tal uns "requisits mínims", però el que volem sobretot és RAM. Per tant el que farem serà tenir una màquina amb:

- Memòria RAM: 5GB
- Sistema Operatiu: Debian 10
- Espai per al disc: 20 GB

- Configurem la xarxa

Afegim una ip estàtica al fitxer /etc/network/interfaces. L'adreça del servidor de la base de dades serà: **192.168.1.6**

```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.1.6
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
```

Després fem:

```
# ifdown enp0s3
# ifup enp0s3
```

I comprovem que la ip s'ha canviat correctament amb # ip address:

```
root@debian:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:9a:9b:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9a:9b28/64 scope link
        valid_lft forever preferred_lft forever
```

- Instal·lar software necessari

1. Actualitzem repositoris:

```
# apt-get update
# apt-get upgrade
```

2. Importem la clau pública:

```
# apt-get install gnupg
# wget -qO-
https://www.mongodb.org/static/pgp/server-4.2.asc | sudo
apt-key add
```

```
root@debian:~# wget -qO- https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add
OK
```

Pot donar error de que no tinguem instal·lat:

```
# apt-get install sudo
```

Si ens dona un **OK** es que tot ha anat correctament.

3. Creem una font d'apt per a MongoDB:

```
# nano /etc/apt/sources.list.d/mongodb.org-4.2.list
```

A aquest fitxer li posem la línia:

```
deb http://repo.mongodb.org/apt/debian buster/mongodb-org/4.2
main
```

```
GNU nano 3.2 /etc/apt/sources.list.d/mongodb.org-4.2.list Modificado
deb http://repo.mongodb.org/apt/debian buster/mongodb-org/4.2 main
```

4. Instal·lem el paquet:

Actualitzem el repositori:

```
# apt-get update
```

Instal·lem el paquet que ens interessa:

```
# sudo apt-get install -y mongodb-org
```

- Posant mongodb en marxa

Ara tindrem un nou servei anomenat mongod

Si mirem l'estat amb:

```
# systemctl status mongod
```

Veurem que està inactiu

```
root@debian:~# systemctl status mongod
• mongod.service - MongoDB Database Server
  Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: https://docs.mongodb.org/manual

mar 26 18:53:23 debian systemd[1]: /lib/systemd/system/mongod.service:10: PIDFile= references path b
lines 1-6/6 (END)
```

L'habilitem per a que arranqui junt a Debian 10 amb:

```
# systemctl enable mongod
```

```
root@debian:~# systemctl enable mongod
Created symlink /etc/systemd/system/multi-user.target.wants/mongod.service → /lib/systemd/system/mongod.service.
```

Arrenquem el servei manualment per a no haver de rebootar la màquina i consultem el seu estat, que ara veurem que està active(running).

```
# systemctl start mongod
```

```
# systemctl status mongod
```

```
root@debian:~# systemctl start mongod
root@debian:~# systemctl status mongod
• mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-03-26 18:56:00 CET; 2s ago
     Docs: https://docs.mongodb.org/manual
  Main PID: 3893 (mongod)
    Memory: 79.2M
    CGroup: /system.slice/mongod.service
            └─3893 /usr/bin/mongod --config /etc/mongod.conf

mar 26 18:56:00 debian systemd[1]: Started MongoDB Database Server.
```

Podem accedir ara al shell mongo:

```
# mongo
```

```
root@debian:~# mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("10521173-87cf-40a8-8040-a44f66a2160a") }
MongoDB server version: 4.2.5
Server has startup warnings:
2020-03-26T18:56:00.586+0100 I STORAGE [initandlisten]
2020-03-26T18:56:00.586+0100 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-03-26T18:56:00.586+0100 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten]
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten]
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten]
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2020-03-26T18:56:01.308+0100 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

Si volem llistar les bases de dades:

```
# show dbs
```

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
```

Creem una base de dades per les proves de més endavant:

```
# use prova
```

Una vegada inserim un element, la base de dades quedarà guardada.

```
> use prova
switched to db prova
> db.altre_prova.insert({"name":"hola"})
WriteResult({ "nInserted" : 1 })
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
prova    0.000GB
```

Creem usuaris:

- admin: L'hem de crear a la base de dades "admin".

```
> db.createUser({
  user: "admin",
  pwd: "admin",
  roles: [
    { role: "userAdminAnyDatabase", db: "admin" },
    { role: "readWriteAnyDatabase", db: "admin" },
    { role: "dbAdminAnyDatabase", db: "admin" }
  ]
});
```

```

> use admin;
switched to db admin
> db.createUser({
...     user: "admin",
...     pwd: "admin",
...     roles: [
...     { role: "userAdminAnyDatabase", db: "admin" },
...     { role: "readWriteAnyDatabase", db: "admin" },
...     { role: "dbAdminAnyDatabase", db: "admin" }
...     ]
... });
Successfully added user: {
  "user" : "admin",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    },
    {
      "role" : "readWriteAnyDatabase",
      "db" : "admin"
    },
    {
      "role" : "dbAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
> 

```

- user1: El crearem a la base de dades prova.

El creem com abans hem creat a admin però amb els seus respectius permisos:

```

> db.createUser({
  user: "user1",
  pwd: "user1",
  roles: [
    { role: "userAdmin", db: "prova" },
    { role: "dbAdmin", db: "prova" },
    { role: "readWrite", db: "prova" }
  ]
});

```

```

> db.createUser({
...   user: "user1",
...   pwd: "user1",
...   roles: [
...     { role: "userAdmin", db: "prova" },
...     { role: "dbAdmin", db: "prova" },
...     { role: "readWrite", db: "prova" }
...   ]
... });
Successfully added user: {
  "user" : "user1",
  "roles" : [
    {
      "role" : "userAdmin",
      "db" : "prova"
    },
    {
      "role" : "dbAdmin",
      "db" : "prova"
    },
    {
      "role" : "readWrite",
      "db" : "prova"
    }
  ]
}

```

Per a que els rols que hem assignat funcionin hem de modificar el fitxer
 /etc/mongod.conf i afegir:

```

security:
  authorization: enabled

```

```

security:
  authorization: enabled

```

Per defecte mongoDB només accepta connexions de localhost. Hem de permetre
 acceptar connexions remotes.

En el mateix fitxer d'abans, /etc/mongod.conf, substituïm:

```

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

```


per

```
# network interfaces
```

```
net:
```

```
port: 27017
```

```
bindIp: 0.0.0.0
```

```
GNU nano 3.2 /etc/mongod.conf

# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true
#   engine:
#   mmapv1:
#   wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0
```

Reiniciem el servei per aplicar els canvis:

```
# systemctl restart mongod
```

Comprovem que està actiu i no hem escrit res malament al fitxer de configuració:

```
# systemctl status mongod
```

```
root@debian:~# systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-03-28 11:43:37 CET; 1min 44s ago
     Docs: https://docs.mongodb.org/manual
  Main PID: 805 (mongod)
    Memory: 178.0M
    CGroup: /system.slice/mongod.service
            └─805 /usr/bin/mongod --config /etc/mongod.conf

mar 28 11:43:37 debian systemd[1]: Started MongoDB Database Server.
```

- Proves

- En local:

Ara si accedim només amb `# mongo` veurem que no podem accedir a les bases de dades existents:

```
root@debian:~# mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a4bd6dc7-999f-42d2-b0b5-b4e81b398891") }
MongoDB server version: 4.2.5
> show dbs
> 
```

Per accedir amb admin:

```
# mongo -u admin -p adminpassword 127.0.0.1/admin
```

```
root@debian:~# mongo -u admin -p admin 127.0.0.1/admin
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/admin?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("09a9c9ee-e90d-4ffa-abe4-d4f80d6d35d5") }
MongoDB server version: 4.2.5
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB
prova    0.000GB
```

Per accedir amb un altre usuari:

```
# mongo -u username -p userpassword 127.0.0.1/database
```

```
root@debian:~# mongo -u user1 -p user1 127.0.0.1/prova
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/prova?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("ae12101c-ad08-46ff-bc72-a5c2d3ade547") }
MongoDB server version: 4.2.5
> show dbs
prova 0.000GB
> 
```

- En remot: Igual però canviant la ip per la del nostre servidor.

```
# mongo -u username -p userpassword ip-servidor/database
```

```
root@debian:~# mongo -u user1 -p user1 192.168.1.6/prova
MongoDB shell version v4.2.5
connecting to: mongodb://192.168.1.6:27017/prova?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("43c83dfe-1636-4af9-b72c-03c032c46af4") }
MongoDB server version: 4.2.5
> show dbs
prova 0.000GB
>
```

Servidor DNS

Un servidor DNS (sistema de noms de domini) és un sistema de noms pels ordinadors, la seva funció és traduir una adreça IP a una adreça entenedora pels humans.

- Tipus de servidors DNS:

Hi ha tres tipus de servidors DNS que són:

- Servidors DNS primaris: contenen els fitxers de configuració del domini i responen a les consultes DNS.
- Servidors DNS secundaris: funcionen com una còpia de seguretat i distribució de la càrrega. Els servidors DNS primaris envien les actualitzacions als servidors DNS secundaris.
- Emmagatzematge en cache del servidor DNS: la seva tasca és emmagatzemar en cache les respostes DNS per evitar tornar a preguntar al servidor DNS primari o secundari.

- Hardware pel servidor DNS de Linux:

El hardware necessari per a un sistema operatiu "debian 10" on instal·larem i configurarem el nostre servidor DNS a més d'altres servidors, depenent de la instal·lació que volem fer (amb escriptori o sense) és el següent:

Tipo de instalación	RAM	Disco duro
Sin escritorio	24 Megabytes	450 Megabytes
Con escritorio	64 Megabytes	1 Gigabyte
Servidor	128 Megabytes	4 Gigabytes

El requisit del hardware està relacionat a l'ús que es vol fer amb el servidor DNS, és a dir com més zones tinguem més RAM ens farà falta i com més consultes més processadors ens farà falta. Una estimació del RAM que ens faria falta si tenim 5000-6000 zones seria de 512 MB però si tenim més zones, llavors amb 1024 MB de RAM ens aniria millor.

- **Software pel servidor DNS de Linux:**

Hi ha molts paquets en Linux que implementen la funcionalitat DNS, però nosaltres farem servir el servidor DNS BIND perquè és la més utilitzada en els servidors DNS de tot el món. Per fer la instal·lació del servidor DNS BIND en Linux fem el següent:

```
$ apt-get install bind9
```

```
$ apt-get install dnsutils
```

Una vegada feta la instal·lació, per iniciar el servei i habilitar-ho per tal que s'executi en el moment d'iniciar el sistema s'ha d'executar les següents comandes:

```
$ systemctl start named
```

```
$ systemctl enable named
```

- **Versió del servidor Debian per DNS:**

Nosaltres farem la instal·lació i configuració del servidor DNS amb la versió actual de Debian que és la 10 amb el nom en clau "buster". Aquesta versió va ser llançada el 6 de juliol de 2019.

```
Linux seax 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64 GNU/Linux
root@seax:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@seax:/#
```

- **Instal·lant i configurant el servidor DNS de Linux:**

Abans de començar amb la instal·lació hem de dir el nom del nostre domini intern que és "VIA.com". Fem la instal·lació del servidor DNS BIND, en el nostre sistema operatiu (Debian 10), de la següent manera:

```
root@seax:/# apt-get install bind9
```

```
root@seax:/# apt-get install dnstools
```

Una vegada finalitzada la instal·lació comencem amb la configuració del servidor. Per començar la configuració li donem al nostre servidor una IP estàtica, per fer-ho editem el fitxer `/etc/network/interfaces` de la següent manera:

```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.3
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
```

Després fem:

```
# ifdown enp0s3
```

```
# ifup enp0s3
```

I comprovem que tenim la nova amb la comanda “ip address”:

```
# ip address
```

```
root@seax:~# ifdown enp0s3
root@seax:~# ifup enp0s3
root@seax:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:00:00:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
root@seax:~#
```

Com podem observar la nostra ip modificada com volíem.

Una vegada tenim la IP estàtica ens posem al directori /etc/bind aquí és on tenim els fitxers de configuració del servidor DNS, per fer modificacions al fitxer /etc/bind s'hauria de fer amb permisos de root (sudo su). El nostre servidor DNS consultarà el fitxer /etc/bind/named.conf.options quan no trobi la resposta a la nostra petició, editem el fitxer named.conf.options de la següent manera:

Accedim al fitxer /etc/bind/named.conf.options:

```
root@seax:/etc/bind# ls
bind.keys  db.255      db.local    named.conf.local  zones.rfc1
db.0       db.empty    named.conf   named.conf.options
db.127     db.gestioVIA  named.conf.default-zones  rndc.key
root@seax:/etc/bind# cp named.conf.options named.conf.options.old
root@seax:/etc/bind# nano named.conf.options_
```

El fitxer named.conf.options hauria de tenir aquest contingut abans de fer cap modificació:

```
// forwarders {
//     0.0.0.0;
// };

//=====
// If BIND logs error messages about the root key
// you will need to update your keys.  See https://
//=====
dnssec-validation auto;

listen-on-v6 { any; };
```

Llavors afegim les següents línies:

```

// If your ISP provided one or more IP addresses for stable
// nameservers, you probably want to use them as forwarders.
// Uncomment the following block, and insert the addresses replacing
// the all-0's placeholder.

forwarders {
    192.168.1.3;
    8.8.8.8;
};

//=====
// If BIND logs error messages about the root key being expired,
// you will need to update your keys. See https://www.isc.org/bind-keys
//=====
dnssec-validation auto;
auth-nxdomain no;_
listen-on-v6 { any; };
};

```

Ara editem el fitxer `/etc/bind/named.conf.local`, en aquest fitxer hem de posar les zones de cerca directa i inversa del nostre servei DNS, el domini de la nostra zona directa i la subxarxa de la zona inversa. També hem d'incloure el tipus de servei i en quins fitxers farà la cerca de noms:

El contingut del fitxer `/etc/bind/named.conf.local` sense haver fet cap modificació és el següent:

```

root@seax:/etc/bind# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
root@seax:/etc/bind# _

```

Modifiquem el fitxer `/etc/bind/named.conf.local` afegint les següents línies:


```

GNU nano 3.2 /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
//Arxiu de cerques directes
zone "VIA.com"{
    type master;
    file "/etc/bind/VIA.db";

};
//Arxiu de cerques inverser
zone "1.168.192.in-addr.arpa"{
    type master;
    file "/etc/bind/192.rev";
};

```

Comprovem que no tenim errors de sintaxis al fitxer /etc/bind/named.conf.local amb la comanda “named-checkconf”:

```

root@seax:/etc/bind# named-checkconf
root@seax:/etc/bind#

```

Com podem observar no ens ha sortit cap missatge d’error per tant no tenim cap error de sintaxi.

Després creem els fitxers VIA.db i 192.rev que hem posat al fitxer /etc/bind/named.conf.local:

Creem el fitxer VIA.db a partir del fitxer /etc/bind/db.local ja existent:

```

root@seax:/etc/bind# cp db.local VIA.db
root@seax:/etc/bind# ls
bind.keys  db.255    named.conf      named.conf.options  VIA.db
db.0       db.empty  named.conf.default-zones  named.conf.options.old  zones.rfc
db.127     db.local  named.conf.local      rndc.key
root@seax:/etc/bind#

```

El contingut del VIA.db sense haver fet cap modificació és el següent:


```

root@seax:/etc/bind# cat VIA.db
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
root@seax:/etc/bind#

```

Llavors editem el fitxer /etc/bind/VIA.db de la següent manera:

```

GNU nano 3.2 /etc/bind/VIA.db
;
; BIND data file for VIA.db
;
$TTL      86400
@         IN      SOA      VIA.com. root.VIA.com. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        86400 ) ; Negative Cache TTL
;
@         IN      NS       VIA.com.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
www       IN      A        192.168.1.4
dns       IN      A        192.168.1.3
bdd       IN      A        192.168.1.6
dhcp      IN      A        192.168.1.2
servidordns IN CNAME     dns.VIA.com.

```

Comprovem que no tenim cap error de sintaxi en el fitxer /etc/bind/VIA.db amb la comanda “named-checkzone” de la següent manera:

```

root@seax:/# cd /etc/bind
root@seax:/etc/bind# named-checkzone VIA.com /etc/bind/VIA.db
zone VIA.com/IN: loaded serial 2
OK
root@seax:/etc/bind# _

```

Com podem observar he obtingut un “OK” això vol dir que no tenim cap error de sintaxi en el fitxer /etc/bind/VIA.db

Creem el fitxer 192.rev a partir del fitxer /etc/bind/db.local ja existent:

```

root@seax:/etc/bind# ls
bind.keys  db.255    named.conf      named.conf.options  VIA.db
db.0       db.empty  named.conf.default-zones  named.conf.options.old  zones.rfc1918
db.127     db.local  named.conf.local      rndc.key
root@seax:/etc/bind# cp db.local 192.rev
root@seax:/etc/bind# _

```

El contingut del 192.rev sense haver fet cap modificació és el següent:

```

root@seax:/etc/bind# cat 192.rev
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
root@seax:/etc/bind# _

```

Llavors editem el fitxer /etc/bind/192.rev de la següent manera:

```
GNU nano 3.2 /etc/bind/192.rev
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      VIA.com. root.VIA.com. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       VIA.com.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
4         IN      PTR      www.VIA.com
3         IN      PTR      dns.VIA.com
6         IN      PTR      bdd.VIA.com
2         IN      PTR      dhcp.VIA.com
```

Comprovem que no tenim cap error de sintaxi en el fitxer /etc/bind/192.rev amb la comanda “named-checkzone” de la següent manera:

```
root@seax:/etc/bind# named-checkzone VIA.com /etc/bind/192.rev
zone VIA.com/IN: loaded serial 1
OK
root@seax:/etc/bind# _
```

Com podem observar he obtingut un “OK” això vol dir que no tenim cap error de sintaxi en el fitxer /etc/bind/192.rev

Ara editem el fitxer /etc/network/interfaces de la següent manera:

El contingut del fitxer /etc/network/interfaces sense haver fet cap modificació és el següent:

```
root@seax:/etc/bind# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.3
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
root@seax:/etc/bind# _
```

Modifiquem el fitxer `/etc/network/interfaces` afegint la següent línia:

```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.3
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    dns-nameservers 192.168.1.3
```

Una vegada hem guardat els fitxers de configuracions `VIA.db`, `192.rev` i el fitxer `/etc/network/interfaces`, reiniciem el procés `bind` amb la següent comanda:

```

root@seax:/# service bind9 restart
root@seax:/# service bind9 status
• bind9.service - BIND Domain Name Server
  Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2020-03-24 21:14:52 CET; 1s ago
  Docs: man:named(8)
  Process: 446 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 447 (named)
  Tasks: 4 (limit: 1150)
  Memory: 11.7M
  CGroup: /system.slice/bind9.service
          └─447 /usr/sbin/named -u bind

mar 24 21:14:52 seax named[447]: managed-keys-zone: loaded serial 12
mar 24 21:14:52 seax named[447]: zone 0.in-addr.arpa/IN: loaded serial 1
mar 24 21:14:52 seax named[447]: zone 127.in-addr.arpa/IN: loaded serial 1
mar 24 21:14:52 seax named[447]: zone 1.168.192.in-addr.arpa/IN: loaded serial 1
mar 24 21:14:52 seax named[447]: zone 255.in-addr.arpa/IN: loaded serial 1
mar 24 21:14:52 seax named[447]: zone localhost/IN: loaded serial 2
mar 24 21:14:52 seax named[447]: zone VIA.com/IN: loaded serial 2
mar 24 21:14:52 seax named[447]: all zones loaded
mar 24 21:14:52 seax systemd[1]: Started BIND Domain Name Server.
mar 24 21:14:52 seax named[447]: running
root@seax:/#

```

Després hem de modificar el fitxer `/etc/resolv.conf`, perquè el nostre ordinador entengui que el nostre servidor DNS és ell mateix.

El contingut del fitxer `/etc/resolv.conf` sense haver fet cap canvi és el següent:

```

root@seax:/etc/bind# cat /etc/resolv.conf
domain home
search home
nameserver 10.0.2.3
root@seax:/etc/bind# _

```

Modifiquem el fitxer `/etc/resolv.conf` de la següent manera:

```

root@seax:/etc/bind# cat /etc/resolv.conf
nameserver 192.168.1.3
search VIA.com
root@seax:/etc/bind# _

```

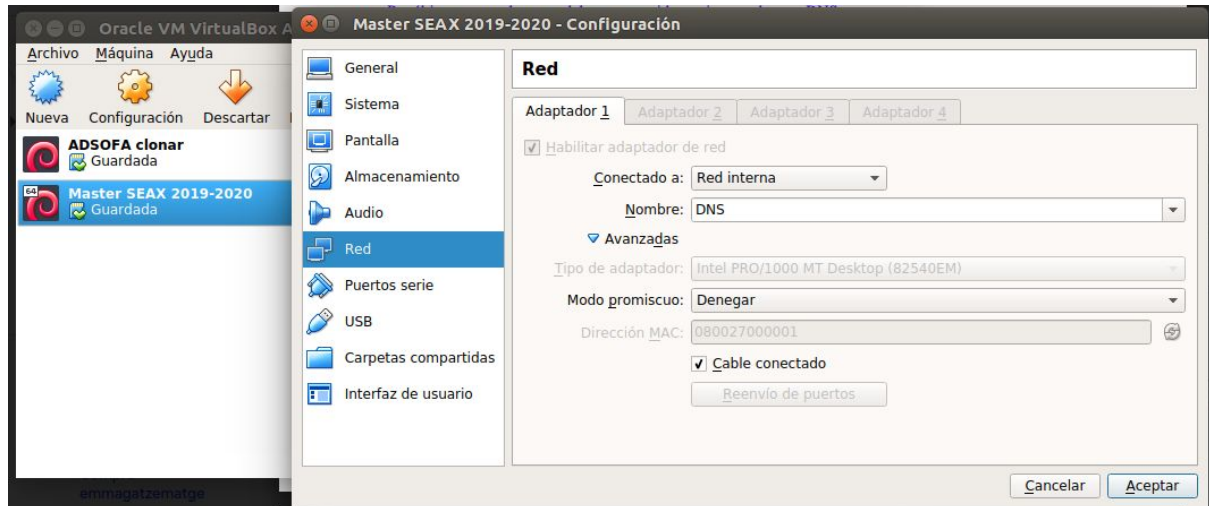
Després d'haver modificat el fitxer `/etc/resolv.conf` reiniciem el bind9 de la següent manera:

```

root@seax:/# /etc/init.d/bind9 restart
[ ok ] Restarting bind9 (via systemctl): bind9.service.
root@seax:/#

```

Per acabar amb la configuració del servidor DNS, definim la xarxa del nostre servidor com interna i de nom DNS:



- **Proves per comprovar el correcte funcionament del servidor DNS:**

Ja tenim el nostre servidor DNS funcionant, per comprovar que ho fa correctament, realitzarem algunes comprovacions amb la comanda “nslookup”. Amb aquesta comanda podem comprovar que es resolen les peticions de la zona directa:

```
root@seax:/# nslookup VIA.com
Server:      192.168.1.3
Address:     192.168.1.3#53

Name:   VIA.com
Address: 127.0.0.1
Name:   VIA.com
Address: ::1

root@seax:/# nslookup www.VIA.com
Server:      192.168.1.3
Address:     192.168.1.3#53

Name:   www.VIA.com
Address: 192.168.1.4
```

```
root@seax:/# nslookup dns.VIA.com
Server:      192.168.1.3
Address:     192.168.1.3#53

Name:   dns.VIA.com
Address: 192.168.1.3

root@seax:/# _
```

I també la zona inversa:


```
root@seax:/# nslookup 192.168.1.3
3.1.168.192.in-addr.arpa      name = dns.VIA.com.1.168.192.in-addr.arpa.
root@seax:/# _
```

Fem el mateix amb la comanda ping, amb aquesta comanda podem comprovar que no hi ha errors de xarxa a l'accedir al nostre servidor DNS:

Accedim a les zones directes:

```
root@seax:/# ping www.VIA.com
PING www.VIA.com (192.168.1.4) 56(84) bytes of data.
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=1 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=2 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=3 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=4 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=5 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=6 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=7 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=8 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=9 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=10 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=11 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=12 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=13 Destination Host Unreachable
From dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3) icmp_seq=14 Destination Host Unreachable
```

```
root@seax:/# ping VIA.com
PING VIA.com(localhost (:::1)) 56 data bytes
64 bytes from localhost (:::1): icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from localhost (:::1): icmp_seq=2 ttl=64 time=0.091 ms
64 bytes from localhost (:::1): icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from localhost (:::1): icmp_seq=4 ttl=64 time=0.113 ms
64 bytes from localhost (:::1): icmp_seq=5 ttl=64 time=0.086 ms
64 bytes from localhost (:::1): icmp_seq=6 ttl=64 time=0.122 ms
64 bytes from localhost (:::1): icmp_seq=7 ttl=64 time=0.114 ms
64 bytes from localhost (:::1): icmp_seq=8 ttl=64 time=0.101 ms
64 bytes from localhost (:::1): icmp_seq=9 ttl=64 time=0.094 ms
_
```

```
root@seax:/# ping dns.VIA.com
PING dns.VIA.com (192.168.1.3) 56(84) bytes of data.
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=1 ttl=64 time=0.038 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=4 ttl=64 time=0.088 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=6 ttl=64 time=0.086 ms
64 bytes from dns.VIA.com.1.168.192.in-addr.arpa (192.168.1.3): icmp_seq=7 ttl=64 time=0.086 ms
_
```

I també a la zona inversa:


```
root@seax:/# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.059 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.089 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.089 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=0.119 ms
64 bytes from 192.168.1.3: icmp_seq=6 ttl=64 time=0.079 ms
64 bytes from 192.168.1.3: icmp_seq=7 ttl=64 time=0.082 ms
```

Finalment podem afirmar que el nostre servidor DNS està instal·lat, configurat i funcionant correctament.

Servidor DHCP

DHCP (Dinamic Host Configuration Protocol) és un protocol de xarxa del tipus client-servidor en el qual el servidor assigna dinàmicament una direcció IP als diferents dispositius de la mateixa xarxa.

Requisits del Sistema:

Un servidor de DHCP varia en funció dels nostres clients esperats, però generalment necessita pocs recursos i gairebé qualsevol màquina serviria. En el nostre cas farem servir una màquina virtual amb:

El S.O serà un Debian 10 - Buster

1GB de RAM(512MB són suficients)

8GB de disc (almenys 4GB)

Per fer la instal·lació i configuració del servidor és necessari accedir a la màquina amb l'usuari root. Ja sigui amb un 'su root', fent un login directe com a root o utilitzant la comanda 'sudo ' al davant de la resta. Aquí es treballarà com a usuari root durant tot el procés.

Instal·lació del software

Només caldrà instal·lar el paquet isc-dhcp-server en la màquina per poder obtenir un DHCP funcional

Per això farem la comanda 'sudo apt update' per actualitzar el repositori.

```
root@seax:~# apt update
Des:1 http://security.debian.org/debian-security buster/updates InRelease [65,4 kB]
Des:2 http://security.debian.org/debian-security buster/updates/main Sources [106 kB]
Des:3 http://security.debian.org/debian-security buster/updates/main amd64 Packages [185 kB]
Des:4 http://security.debian.org/debian-security buster/updates/main Translation-en [98,9 kB]
Obj:5 http://deb.debian.org/debian buster InRelease
Des:6 http://deb.debian.org/debian buster-updates InRelease [49,3 kB]
Descargados 505 kB en 5s (94,2 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
root@seax:~#
```

Un cop els paquets estan actualitzats, cal continuar amb la comanda 'apt install isc-dhcp-server -y'

L'opció -y serveix per estalviar la confirmació del paquet i fer la descàrrega directament

```
root@seax:~# apt install isc-dhcp-server -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
isc-dhcp-server ya está en su versión más reciente (4.4.1-2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@seax:~#
```

Configuració del servidor

-Xarxa estàtica

Ara comencem la configuració del servidor. Primer donarem a la nostra màquina una direcció estàtica en la interfície que usarem. En aquesta màquina només trobarem la interfície 'enp0s3' i serà la que es modificarà.

L'adreça és donada estàtica ja que se suposa que el servidor sempre estarà en aquesta adreça i no hauria d'haver més màquines associades.

Per donar-li la direcció ens caldrà modificar el fitxer /etc/network/interfaces i deixar-lo així:

```
GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

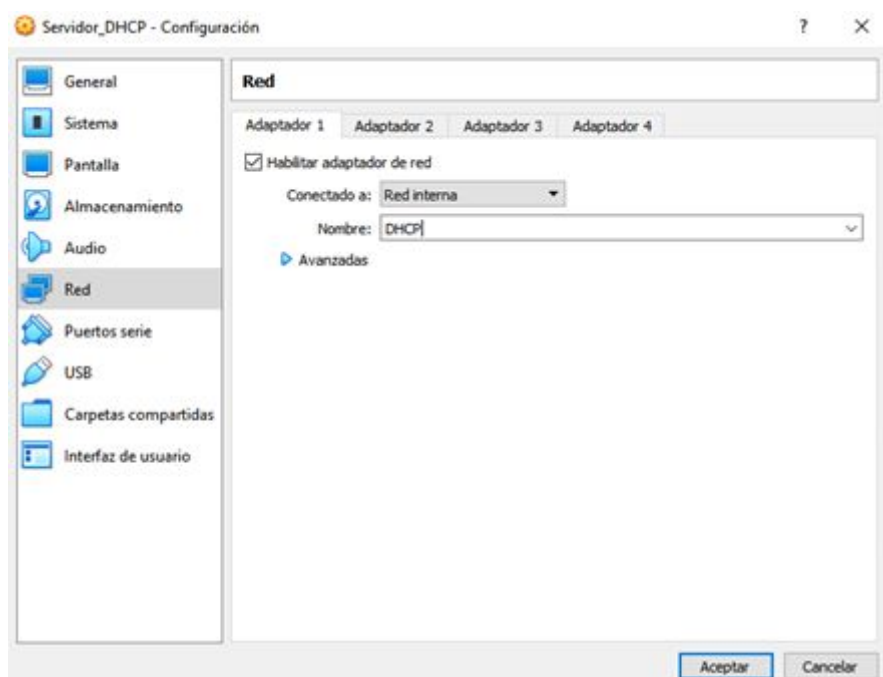
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3
auto enp0s3
iface enp0s3 inet static_
    address 192.168.1.2
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    broadcast 192.168.1.255
    dns-nameservers 192.168.1.3
```

-Xarxa interna

Un cop modificada la interfície tancarem la màquina i a la configuració de VirtualBox sel·leccionarem l'opció Red Interna a l'apartat de Red. Aquesta xarxa li donarem el nom 'DHCP'.



-Configuració de les opcions DHCP

Un cop la màquina està en Xarxa Interna, l'encenem i modificarem els fitxers /etc/dhcp/dhcpd.conf i el /etc/default/isc-dhcp-server com a usuari root i hi afegim el següent:

```
GNU nano 3.2 /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.8 192.168.1.250;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.3;
}
```

Per assegurar que tenim la sintaxis ben escrita en el dhcpd.conf, podem utilitzar la comanda '/usr/sbin/dhcpd -t' i ens indicarà si tenim problemes en el fitxer.

```
root@seax:~# /usr/sbin/dhcpd -t
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
root@seax:~# _
```

En la nostra xarxa necessitarem reservar diverses adreces per el servidor DNS, el web i per la base de dades. Necessitarem donar al DNS la direcció 192.168.1.3, el servidor web necessitarà la 192.168.1.4 i la base de dades la direcció 192.168.1.6.

Per donar una direcció estàtica una màquina tenim dos alternatives, la primera seria modificar el fitxer d'interfaces (com s'ha fet anteriorment) o podem configurar-ho al dhcpd.conf del servidor dhcp. Per poder assignar la IP estàtica, necessitarem el nom de la màquina i la MAC del adaptador que es connecta a la xarxa DHCP. El nom s'obté amb la comanda 'hostname' i l'adreça MAC amb la comanda 'ip a' podem conèixer la MAC dels adaptadors de xarxa.

```
host dns-server{
    hardware ethernet 08:00:27:00:00:01;
    fixed-address 192.168.1.3;
}
host web-server{
    hardware ethernet 08:00:27:dd:52:af;
    fixed-address 192.168.1.4;
}
host bdd-server{
    hardware ethernet 08:00:27:d2:22:3d;
    fixed-address 192.168.1.6;
}
```

En el fitxer /etc/default/isc-dhcp-server li indicarem la interfície que farà ús del dhcp ja que el sistema pot tenir-ne d'altres i en aquest cas caldria diferenciar-les.

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 3.2 /etc/default/isc-dhcp-server

# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
DHCPDv4_PID=/var/run/dhcpd.pid
DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""
```

Si s'han seguit els passos correctament, ara ens caldrà reiniciar el servei del dhcp per aplicar els canvis amb 'systemctl restart isc-dhcp-server' i 'systemctl status isc-dhcp-server'. Si hi ha algun problema un cop es fa el restart tindrem un parell de línies indicant que quelcom ha sortit malament i no està funcionant. En cas contrari es veurà així:

```
root@seax:~# systemctl restart isc-dhcp-server
root@seax:~# systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Sun 2020-03-29 15:19:27 CEST; 1min 24s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 424 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 1150)
   Memory: 4.6M
    CGroup: /system.slice/isc-dhcp-server.service
           └─436 /usr/sbin/dhcpd -4 -g -cf /etc/dhcp/dhcpd.conf enp0s3

mar 29 15:19:25 seax systemd[1]: Starting LSB: DHCP server...
mar 29 15:19:25 seax isc-dhcp-server[424]: Launching IPv4 server only.
mar 29 15:19:25 seax dhcpd[436]: Wrote 0 leases to leases file.
mar 29 15:19:25 seax dhcpd[436]: Server starting service.
mar 29 15:19:27 seax isc-dhcp-server[424]: Starting ISC DHCPv4 server: dhcpd.
mar 29 15:19:27 seax systemd[1]: Started LSB: DHCP server.
root@seax:~# _
```

El requadre vermell emmarca l'estat del servidor i podem observar les connexions dels clients.

Ara ja tenim el servidor preparat i ja podem començar les proves.

Proves de funcionament

Per demostrar el funcionament necessitarem dues màquines addicionals. En aquest cas s'han fet 2 clons de la maquina original abans del procés de instal·lació del software del dhcp. L'únic canvi ha estat el de Xarxa NAT a Xarxa Interna (com s'ha fet anteriorment).

Dit això, farem dues de proves de validació. La primera consistirà en connectar les dues màquines a la xarxa DHCP amb una IP diferent i entre 192.168.1.8 i 192.168.1.250. En el status del servidor (el requadre vermell de l'apartat anterior) hauran d'aparèixer les noves màquines.

La segona prova consisteix en fer un ping entre les dues màquines. Si obtenim resposta indicarà que tot funciona correctament.

-Prova1

Només cal tenir les màquines en xarxa Interna i el servidor obert abans d'encendre-les. Un cop enceses només cal veure la seva adreça amb 'ip a'

Màquina1

```
entel@seax:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:1a:8b:bd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.9/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 487sec preferred_lft 487sec
    inet6 fe80::a00:27ff:fe1a:8bbd/64 scope link
        valid_lft forever preferred_lft forever
```

Màquina2

```
root@seax:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:91:ba:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 464sec preferred_lft 464sec
    inet6 fe80::a00:27ff:fe91:ba00/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:8b:46:f5 brd ff:ff:ff:ff:ff:ff
root@seax:~#
```

Servidor


```

root@seax:~# systemctl status isc-dhcp-server
• isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Sun 2020-03-29 15:34:57 CEST; 5min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 458 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 1150)
   Memory: 4.6M
    CGroup: /system.slice/isc-dhcp-server.service
            └─470 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp0s3

mar 29 15:34:54 seax dhcpd[470]: Wrote 1 leases to leases file.
mar 29 15:34:55 seax dhcpd[470]: Server starting service.
mar 29 15:34:57 seax isc-dhcp-server[458]: Starting ISC DHCPv4 server: dhcpd.
mar 29 15:36:16 seax dhcpd[470]: DHCPDISCOVER from 08:00:27:91:ba:00 via enp0s3
mar 29 15:36:17 seax dhcpd[470]: DHCPOFFER on 192.168.1.10 to 08:00:27:91:ba:00 (seax) via enp0s3
mar 29 15:36:17 seax dhcpd[470]: DHCPREQUEST for 192.168.1.10 (192.168.1.2) from 08:00:27:91:ba:00
mar 29 15:36:17 seax dhcpd[470]: DHCPACK on 192.168.1.10 to 08:00:27:91:ba:00 (seax) via enp0s3
mar 29 15:36:31 seax dhcpd[470]: DHCPREQUEST for 192.168.1.9 from 08:00:27:1a:8b:bd (seax) via enp0s3
mar 29 15:36:31 seax dhcpd[470]: DHCPACK on 192.168.1.9 to 08:00:27:1a:8b:bd (seax) via enp0s3

```

El requadre vermell emmarca el procés de connexió de les dues màquines i com es pot veure, les dues màquines aprèixen connectades, per tant la primera prova està superada.

-Prova2

Ara donem pas a la segona prova: un ping entre Màquina2 i Màquina1:

Això ho farem amb la comanda 'ping ip_maquina' en aquest cas la de la Màquina2 és la 192.168.1.9

```

root@seax:~# ping 192.168.1.9
PING 192.168.1.9 (192.168.1.9) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_seq=1 ttl=64 time=0.950 ms
64 bytes from 192.168.1.9: icmp_seq=2 ttl=64 time=0.504 ms
^C
--- 192.168.1.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 0.504/0.727/0.950/0.223 ms
root@seax:~#

```

Com es pot veure, obtenim resposta i per tant aquesta prova també està superada.

Amb la superació de les dues proves ja queda demostrat que el servidor està funcionant correctament.

Burndown



Webgrafia

- **Docker:**
<https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>
<https://www.redhat.com/es/topics/containers/what-is-docker>
- **Fog:**
<https://www.ionos.es/digitalguide/servidores/know-how/fog-computing/>
<https://www.gradiant.org/blog/edge-fog-computing-cloud/>
- **Base de dades**
<https://docs.mongodb.com/manual/administration/production-notes/>
<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/>
<https://chachocool.com/como-instalar-mongodb-en-debian-9-stretch/>