Projecte iPXE (PXE Cloud)

Projecte de final de de grau mitjà SMX

Néfix Estrada

Jorge Pastor

Curs 2017 - 2018

Índex

1- Estudi Inicial

El nostre projecte va començar quan *Néfix Estrada* i els seus tutors de pràctiques (Alberto Larraz i Josep Maria Viñolas) estaven parlant sobre el projecte de final de grau. Al final, van acabar reunint-se conjuntament amb *Jorge Pastor* per a parlar sobre el tema i algunes idees que ells tenien. Les propostes que ens van fer van ser dues:

- Millorar l'actual implementació del sistema de "Su Turno", tot utilitzant unes Raspberry Pi, unes impressores tèrmiques per a imprimir els tiquets…
- Una mena de repositori central d'imatges d'iPXE on poguéssim arrencar des de dins d'una LAN les imatges d'aquest. A part, necessitava poder tenir algun tipus d'autentificació, grups…

Al final ens vam decidir per la segona, ja que a part que ens semblava més interessant en relació a les tecnologies utilitzades (es toca tot un stack de tecnologies molt ampli i interessant), feia poc que havíem estat treballant el muntatge d'un servidor PXE a classe i teníem ganes de continuar aprenent en aquesta línia.

També vam pensar que seria una bona idea perquè, si desenvolupàvem correctament la idea, es podia arribar a aplicar dins de l'entorn educatiu. Podria ser un gran substitut al tradicional PXE i a la seva manca de flexibilitat, que l'iPXE ens aporta amb característiques com ara arrencar imatges directament des dels mirrors oficials. A part, podia aportar un factor molt interessant com ara la compartició de menús i imatges entre diferents usuaris, grups i fins i tot organitzacions (diferents escoles)

Un cop vam haver decidit embarcar-nos amb aquest projecte, vam començar creant un Trello compartit. Allà hi vam anar fent una llista de tots els requeriments del projecte, organitzant idees i tasques que necessitàvem fer. També vam començar a separar les diferents parts del sistema; especificant les tasques que havia de dur a terme cada una, acabant així amb una idea bastant definida de què volíem / havíem de prioritzar per a poder acabar amb un projecte amb cara i ulls.

2- Anàlisi del Sistema

Hem organitzat el nostre projecte en 5 grans fases:

Fase 1: Visualitzar menús existents

- Poder arrencar un menú d'iPXE
- Autenticació d'usuaris estàtics

Fase 2: Creació de menús

- Crear menús de forma dinàmica llegint-los de la base de dades
- Dockerització dels serveis per a facilitar la instal·lació i el manteniment de la infraestructura
- Començar a tenir una API que pugui interactuar amb la base de dades

Fase 3: Gestió d'imatges

- Poder arrencar imatges des dels menús generats de forma dinàmica
- Gestionar les imatges i menús a través de l'API
- Començar amb el desenvolupament de la interfície web
- Implantar l'autenticació d'usuaris (encriptant les contrasenyes a la base de dades)

Fase 4: Gestió d'usuaris i permisos

- Implantar permisos als usuaris

Aquesta fase l'hem saltat, perquè hem vist que ens portaria massa temps, i en anar justos de temps hem pensat que seria més important millorar el funcionament tant de l'API com de la web i passar a la fase 5 per depurar errors. En el cas d'acabar la cinquena fase, retornaríem a la fase quatre.

Fase 5: Millorar codi i treure errors.

- Millorar codi fer-lo més llegible
- Depurar bugs

3- Eines i conceptes

Hem utilitzat diferents tipus d'eines englobant serveis, llenguatges de programació i eines d'administració.

3.1- iPXE

iPXE és una millora de l'anterior sistema d'arrencada per xarxa PXE. Les seves millores més notòries són la possibilitat de poder arrencar un menú o imatge a través d'Internet (el seu rang d'abast ara no es limita només a la LAN), la possibilitat d'autenticar-te contra un servidor extern, més opcions de personalització dels menús...

3.2- Docker

Docker és un programa de contenització de serveis. Executa cadascun dels serveis virtualitzant un petit sistema operatiu de forma aïllada al host. Per aquesta mateixa raó, pots tenir diferents serveis sense conflictes entre ells.

3.3- DHCP

DHCP és un servei que utilitza la xarxa per a repartir les adreces IP de la xarxa als diferents hosts d'aquesta. Amb aquest servei podem distribuir adreces IP de forma dinàmica o estàtica.

3.4- TFTP

TFTP (Trivial File Transfer Protocol) és un servei de transferència de fitxers entre diferents hosts. En el nostre cas, l'utilitzem per a carregar la "ROM" de l'iPXE al host.

3.5- Nginx

Nginx és un servidor web Open Source multiplataforma que permet allotjar i administrar pàgines web. A més, té una molt bona estabilitat i una fàcil configuració.

3.6- Python

Python és un llenguatge de programació interpretat, amb un especial èmfasi en ser un llenguatge de lectura llegible. Gràcies a això, Python facilita la iniciació a la programació sent un dels llenguatges més adequats per la iniciació a programar.

3.6.1- Flask

Flask és un framework de Python que facilita la creació d'aplicacions web. Entre els seus avantatges, hi podem trobar la facilitat que ens dóna a l'hora de rebre requests del tipus, get, post, put i delete. Això el fa perfecte per a utilitzar-lo com a servidor d'una API. A part, té tots els avantatges que ens proporciona Pyhton.

3.7- HTML

HTML (HyperText Markup Language) és el llenguatge que s'utilitza per a l'elaboració de pàgines web. En realitat, només s'utilitza per a afegir el contingut de la pàgina web, les imatges, i els vídeos (entre d'altres).

3.7.1- Bootstrap

Bootstrap és un framework de CSS i JavaScript pensat per a facilitar el disseny de les pàgines web. Compta amb un seguit de classes CSS que estan predissenyades per a una fàcil i ràpida implementació d'estils i per a una agilització de la definició, el posicionament i/o la maquetació de la pàgina web. Tens l'opció d'importar el framework via una URL o descarregar-los i cridar-los localment.

3.8- JavaScript

JavaScript és un llenguatge de programació orientat a la web. Aquest llenguatge et permet dissenyar una pàgina interactiva. És un element fonamental per a la creació d'aplicacions web.

3.8.1- Jquery

Jquery és una llibreria de JavaScript que va ser creada per a facilitar la interacció amb l'arxiu HTML, interactuar amb diversos objectes i manipular l'arbre DOM entre altres. En aquest projecte l'utilitzem per a interactuar des de la web amb l'API.

3.9- Git

Git és un programa de control de versions que va ser creat per Linus Torvalds. És un programa molt útil quan treballes en un projecte, ja que pots anar controlant els canvis que vas fent i en qualsevol moment poder tornar en un moment passat. Encara és més útil quan treballa més d'una persona en un projecte ja que tota la gent pot anar rebent els canvis que van fent la resta i es pot desenvolupar sense "trepitjar" el codi d'altres.

3.9.1- Github

Github és plataforma de desenvolupament col·laboratiu per a allotjar projectes, principalment de codi Open Source, ja que en la seva versió gratuïta només permet crear repositoris públics. Bàsicament és una web per a allotjar projectes que utilitzen Git

3.9.2- Gitkraken

Gitkraken és una eina que facilita la gestió de projectes que utilitzen Git. Et permet, de forma gràfica, accedir al repositori i utilitzar Git de forma fàcil i agradable. També té suport per a Git Flow.

3.10- RethinkDB

RethinkDB és una base de dades Open Source del tipus NoSQL. Emmagatzema les dades en arxius JSON en esquemes dinàmics. Està dissenyada per a facilitar l'enviament d'actualitzacions en temps real i que els clients puguin rebre aquestes actualitzacions.

3.11- Postman

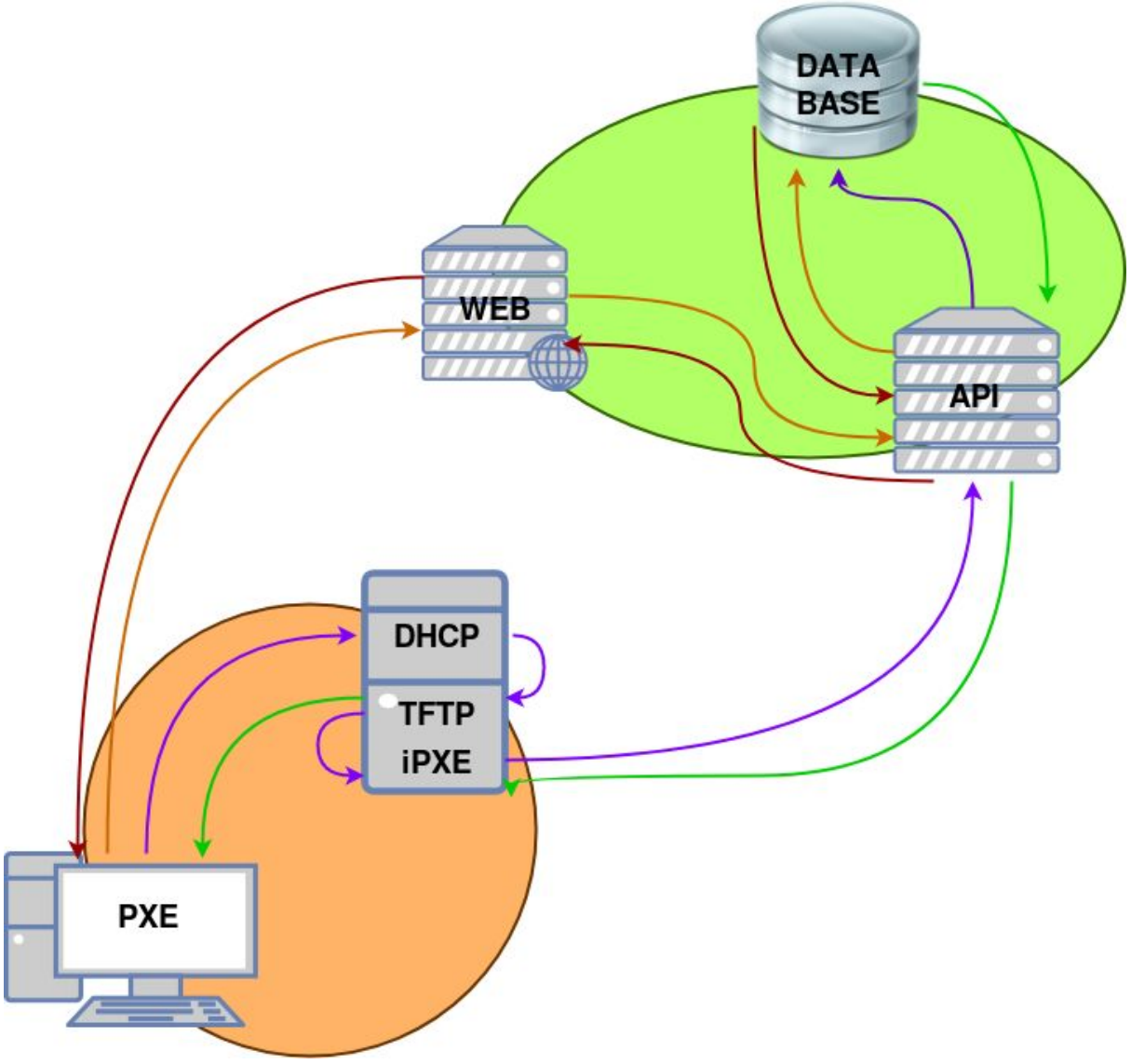
Postman és una eina dissenyada per al desenvolupament d'APIs. Al projecte, l'hem utilitzat per a provar els nous fragments de codi i les noves funcions que anàvem implementant i per a l'automatització de tests contra l'API per a detectar possibles errors introduïts en les noves versions o errors ja existents que havien passat per alt.

3.12- Trello

Trello és una aplicació web pensada per a l'organització de projectes i la productivitat. Funciona per pissarres. Cada projecte té la seva pissarra assignada. Cada pissarra té un seguit de llistes. Un exemple de llistes serien "Idees", "To Do", "Doing" i "Done". Dins d'aquestes llistes, hi ha cartes. Aquestes són les que guarden el contingut. A part, pots assignar les cartes a diferents usuaris, moure-les de llista...

4- Disseny del Sistema

Funcionament del sistema



Base de dades

5- Desenvolupament

Repartiment de les feines

En el repartiment de les feines inicialment els dos hem hagut de buscar informació general del projecte, el que necessitaríem i com ho muntaríem. Un cop amb la informació general recollida i sabent el què necessitaríem ens hem repartit les feines de la següent manera:

Néfix Estrada:

- Creació del Trello per a poder anotar dades y fer seguiment del projecte
- Creació de Github col·laboratiu per a poder treballar conjuntament.
- Desenvolupament de Dockers per a poder implementar una instal·lació ràpida i senzilla de cara a l'usuari.
- Desenvolupament de l'API conjuntament amb la base de dades i totes les seves funcions.
- Documentació del projecte.

Jorge pastor:

- Recerca d'informació i comprovació de la mateixa en relació amb l'arrancada de l'iPXE, els serveis TFTP, DHCP, la generació de menús...
- Recerca d'informació i comprovació de la mateixa en relació amb els paràmetres dels menús de l'iPXE i de l'arrancada d'imatges. Suport del protocol HTTPS
- Desenvolupament de la pàgina web per a poder fer crides a l'API i i gestionar la base de dades de forma senzilla, eficaç i directament des de qualsevol navegador.
- Documentació del projecte.

Desenvolupament

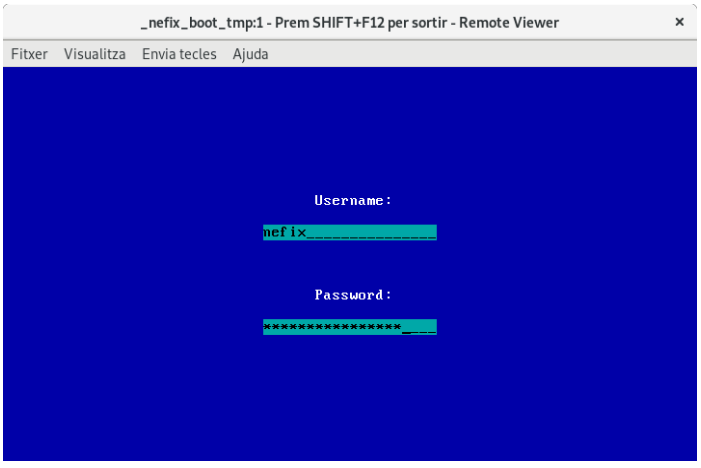
En un inici mentre *Néfix Estrada* creava les imatges de Docker per a facilitar la instal·lació de la plataforma. Mentrestant, *Jorge Pastor* instal·lava i creava una configuració bàsica per al funcionament dels diferents serveis. Un cop els serveis van ser funcionals, *Néfix Estrada* els "Dockeritzar".

Un cop arribats en aquest punt, ja érem capaços d'arrencar un binari d'iPXE (que prèviament havíem compilat) i visualitzar un menú senzill.

Posteriorment, *Néfix Estrada* va "Dockeritzar" la base de dades RethinkDB i va començar el procés de creació de l'API. Al mateix temps, *Jorge Pastor* va verificar tota la informació recol·lectada anteriorment en relació a la generació de menús d'iPXE, a l'arrencada d'imatges, el suport del protocol HTTPS, de menús amb imatges de fons… Seguidament, es va anar implementat tota la informació que s'havia anat recol·lectant a l'API.

Arribats a aquest punt, ja es podia arrancar menús que es generaven dinàmicament, i arrancar imatges. Tot això interactuant amb l'API, que aquesta anava llegint la Base de Dades. També s'havia implementat un sistema d'autentificació, que permetia als usuaris iniciar sessió.

Finalment *Néfix Estrada* va continuar amb la implementació de funcions a l'API. Mentrestant, *Jorge pastor* va començar (i acabar) de programar la pàgina web. Ara ja es podia administrar de forma senzilla el sistema.



Organització del projecte

Per organitzar-nos la feina i fer més fàcil el desenvolupament d'aquest projecte, hem fet ús de les següents eines:

Github

- Repositori col·laboratiu on hi hem allotjat tot el codi. Aquesta eina ens ha sigut molt útil; no només per a tenir el codi accessible ràpidament, sinó per a poder tenir una còpia de seguretat del projecte en cas de pèrdua d'informació.

GitFlow

- Extensió de Git que permet agilitzar la creació de branques de treball. També és molt útil per a gestionar les noves idees i unificar les branques un cop desenvolupades.

Trello

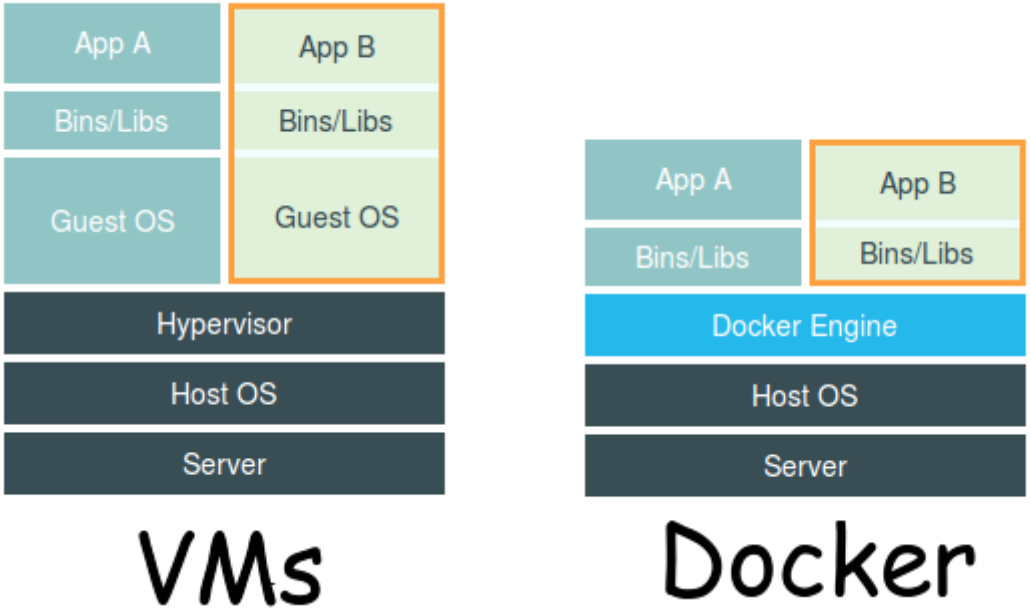
- "Bloc de notes" en línia i col·laboratiu. Ens ha sigut molt útil per a gestionar idees del projecte, establir prioritats i gestionar el treball diari de cada persona.

6- Implantació

La implantació de la plataforma l'hem dut a terme amb Docker.

6.1- Docker

Docker és un programa de contenització de serveis. Per a fer una comparativa, un contenidor de Docker seria com una màquina virtual. Està aïllada del host, en pots executar moltes simultàniament, es poden connectar a la xarxa de diverses maneres… La seva principal diferència és que el seu impacte al rendiment del host és nul ja que no és una virtualització d'un sistema operatiu complet:



Font de la imatge: https://www.cloudbunny.net/ss/docker-vs-vm.png

6.1.1- Avantatges de Docker

Resumint, utilitzar Docker per a executar serveis ens dóna els següents avantatges

- Aïllament dels serveis
- Facilitat de gestió
- Fàcil desplegament i creixement
- Impacte al rendiment quasi nul
- …

6.2- Docker Compose

Docker és una eina molt útil, però té un problema molt gran: un cop comences a tenir més volum de contenidors, es fa complicat de mantenir. Aquí és on apareix Docker Compose. Docker Compose és una eina que ens permet, de forma molt senzilla i intuïtiva, gestionar els nostres grups de serveis utilitzant un sol fitxer de format YAML. El que ens permet és arrencar, aturar, recrear i escalar tots els contenidors de forma simultània i sense haver de passar ni un sol paràmetre.

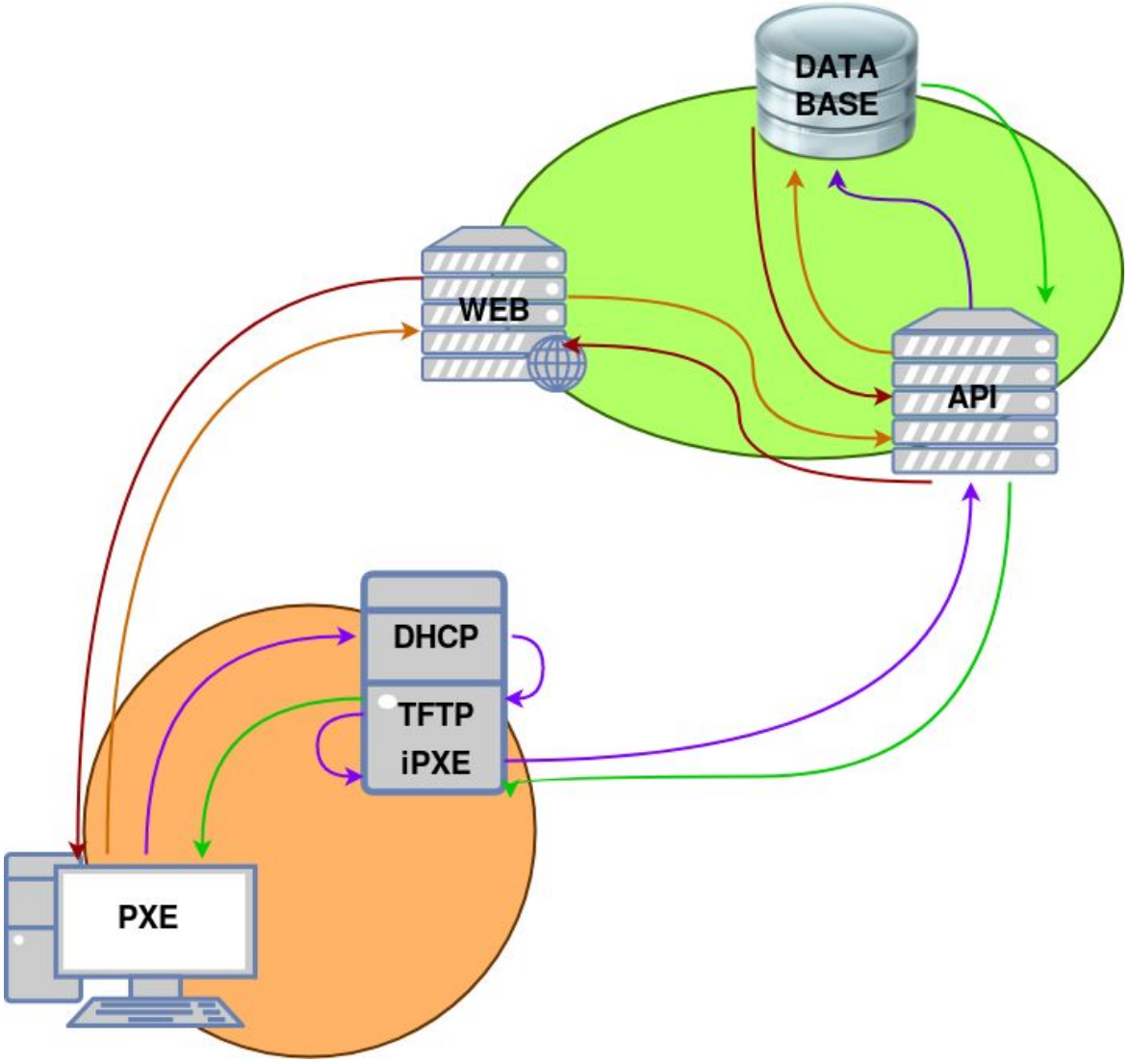
A part, Docker Compose fa la instal·lació i configuració de tots els serveis trivial; només necessitem executar la següent comanda:

```
sudo docker-compose up -d
```

Un cop executada, ja tindríem tots els serveis necessaris per a fer funcionar la plataforma executant-se a la màquina servidora.

6.3- Infraestructura

La infraestructura de la nostra solució és la següent:



El servidor principal és una Raspberry Pi 3B que es troba a casa de Néfix Estrada, executant tots els contenidors.

6.4- Desplegament

Finalment bé el desplegament. És un dels passos més senzills de tot el procés. Podem distingir dues instal·lacions diferents:

6.4.1- Instal·lació del servidor

La instal·lació del servidor només ens servirà si volem muntar-nos la nostra plataforma de PXE Cloud. Normalment, només voldríem configurar el client perquè es connectés a la plataforma original. La instal·lació del servidor és una mica més complicada que la del client. Haurem d'executar les següents comandes:

```
git clone https://github.com/pxe-cloud/pxe-cloud-docker
cd pxe-cloud-docker/server
cp settings.yml.example settings.yml
vim settings.yml
```

Nota: al pas que editem el fitxer, hem de canviar els paràmetres perquè s'adeqüin a les nostres necessitats.

Tot seguit haurem de configurar el nostre servidor Nginx perquè faci un proxy_pass cap al servidor:

```
location / {
  proxy_pass http://192.168.1.5:8001;
}
```

Nota: si volem utilitzar HTTPS, haurem de tenir els certificats a la carpeta de certificats de l'Nginx (nginx-server/certs). Aquesta es muntarà a /etc/pxec-cloud/certs/

Finalment, haurem de tornar a compilar l'iPXE perquè l'script que carrega apunti a la web corresponent.

```
python3 compile_ipxe.py
```

Nota: abans de compilar l'iPXE, haurem de tenir els certificats HTTPS a la carpeta de certificats de l'Nginx (nginx-server/certs).

I ja podrem arrencar tots els serveis

```
sudo docker-compose up -d
```

6.4.2- Instal·lació del client

El client és el que la majoria de gent s'instal·larà. En realitat, només necessitarem un servidor DHCP amb el que apuntarem a un servidor TFTP i un servidor TFTP que servirà el binari (precompilat amb les opcions que necessitem) de l'iPXE amb el qual els clients arrencaran. Per a instal·lar-los, haurem de seguir els següents passos:

1. Ens instal·lem Docker i Docker Compose a la màquina que farà de DHCP i de servidor TFTP (aquest pas varia segons la distribució)
2. Executem les següents comandes:

```
git clone https://github.com/pxe-cloud/pxe-cloud-docker
cd pxe-cloud-docker/local/
cp dhcp-server/conf/dhcpd.conf.example dhcp-server/conf/dhcpd.conf
vim dhcp-server/conf/dhcpd.conf
sudo docker-compose up -d
```

Nota: al pas que editem el fitxer, hem de canviar els paràmetres perquè s'adeqüin a la nostra xarxa.

7- Manteniment

Amb Docker i Docker Compose és molt senzill gestionar el manteniment de tota la solució. Tan sols necessitem executar la comanda que hi ha a continuació cada vegada que vulguem actualitzar el sistema i automàticament tindrem l'última versió de tot el software (ens descarregarem les imatges actualitzades del DockerHub, una plataforma on tenim les nostres imatges de Docker de forma centralitzada i anem actualitzant segons anem traient versions noves del software).

```
sudo docker-compose down # Només en cas de que els serveis
s'estiguessin executant
sudo docker-compose up -d --build --force-recreate
```

Automàticament s'actualitzen tots els contenidors i es tornen a arrencar

8- Conclusions

- Ens ha agradat molt treballar en aquest projecte, ja que hem tocat i après molts tipus de tecnologies diferents. Hem tingut l'oportunitat de treballar amb un stack molt ric i variat

- Ens hauria agradat disposar de més temps per a desenvolupar el projecte perquè hauríem pogut fer un codi més net i més fàcil de mantenir i perquè hauríem pogut afegir més idees i funcionalitats que li donarien molta més robustesa. A part, podríem haver depurat molt millor els errors del codi

- Valorem molt positivament el fet d'haver pogut aprendre mentre anàvem treballant en el projecte, perquè de no haver estat així, el projecte hauria perdut una mica el sentit i no ens hauria motivat tant. També ens ha agradat treballar en grup perquè, a part d'avançar molt més, els projectes adquireixen un caire molt diferent

- Si tornéssim a començar amb el projecte, li dedicaríem molt més temps a la planificació i disseny de la base de dades i dels "endpoints" de l'API. Ens hem trobat que per culpa de tenir una idea poc definida al cap sobre què volíem fer, hem generat molts errors i hem acabat perdent molt de temps

9- Bibliografia

iPXE

http://ipxe.org/docs

Iniciar un menú iPXE

https://www.palepurple.co.uk/ipxe-network-booting-iso-images

Login amb iPXE

http://ipxe.org/cmd/login

Colors i fons del menú iPXE

http://ipxe.org/cmd/cpair

http://ipxe.org/cmd/colour

Menús

https://github.com/skunkie/ipxe-phpmenu

https://gist.github.com/tuxfight3r/877b2a3bf6ae818ce1077684a4e42ad4

CSS i JS.

https://getbootstrap.com/docs/4.1/getting-started/introduction/

https://www.w3schools.com/js/default.asp

https://www.w3schools.com/jquery/default.asp

Base de Dades

https://www.rethinkdb.com/docs/

Docker

https://docs.docker.com/

10- Annexes

Repositori del codi

https://github.com/pxe-cloud/pxe-cloud/tree/develop

Repositori de les imatges de Docker

https://github.com/pxe-cloud/pxe-cloud-docker/tree/develop

Repositori d'imatges de Docker

https://hub.docker.com/r/pxecloud/

Documentació API

https://pxecloud.docs.apiary.io

Web

https://pxecloud.tk

API

https://api.pxecloud.tk

Trello

https://trello.com/b/nPYXiDJS

Demo funcionament ipxe

https://drive.google.com/open?id=1UCSllBgUFexU7o6h5OQzIy2zNjsNmCAw