UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria
de Vilanova i la Geltrú

Desplegament de la Infraestructura i Arquitectura, l'aplicació mòbil, i la Base de Dades

Índex...

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria
de Vilanova i la Geltrú

Descripció general Sprint

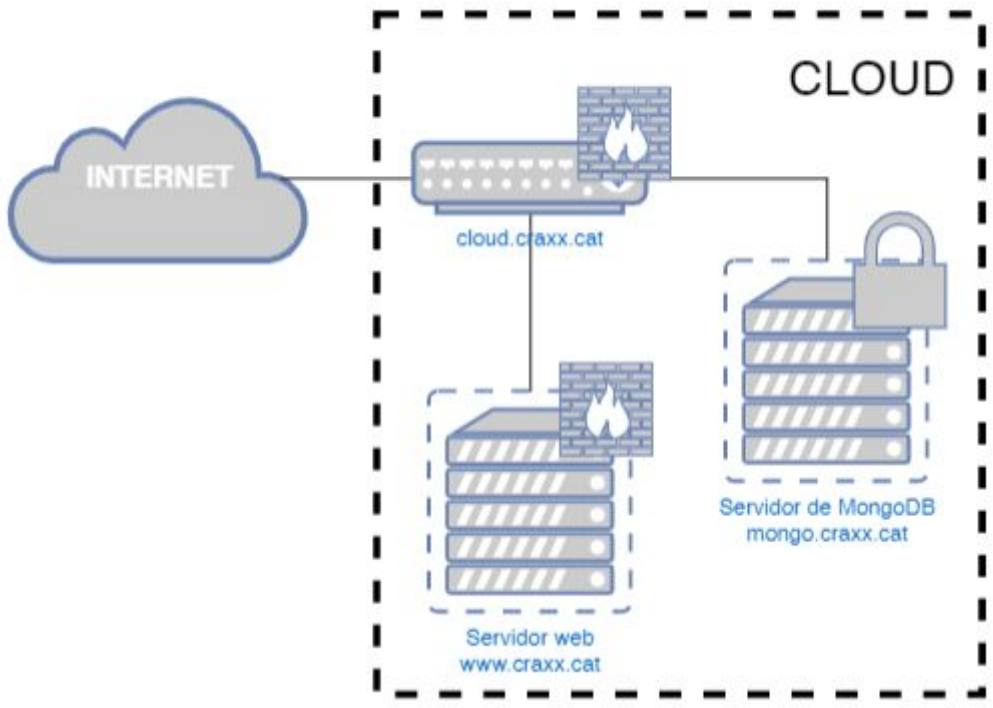
//Descripció global de les tasques realitzades i l'organització del grup. Part del projecte realitzada, estructuració del treball...

Desplegament de la Infraestructura i Arquitectura

Descripció general

La infraestructura ara mateix es divideix en dos components clars:

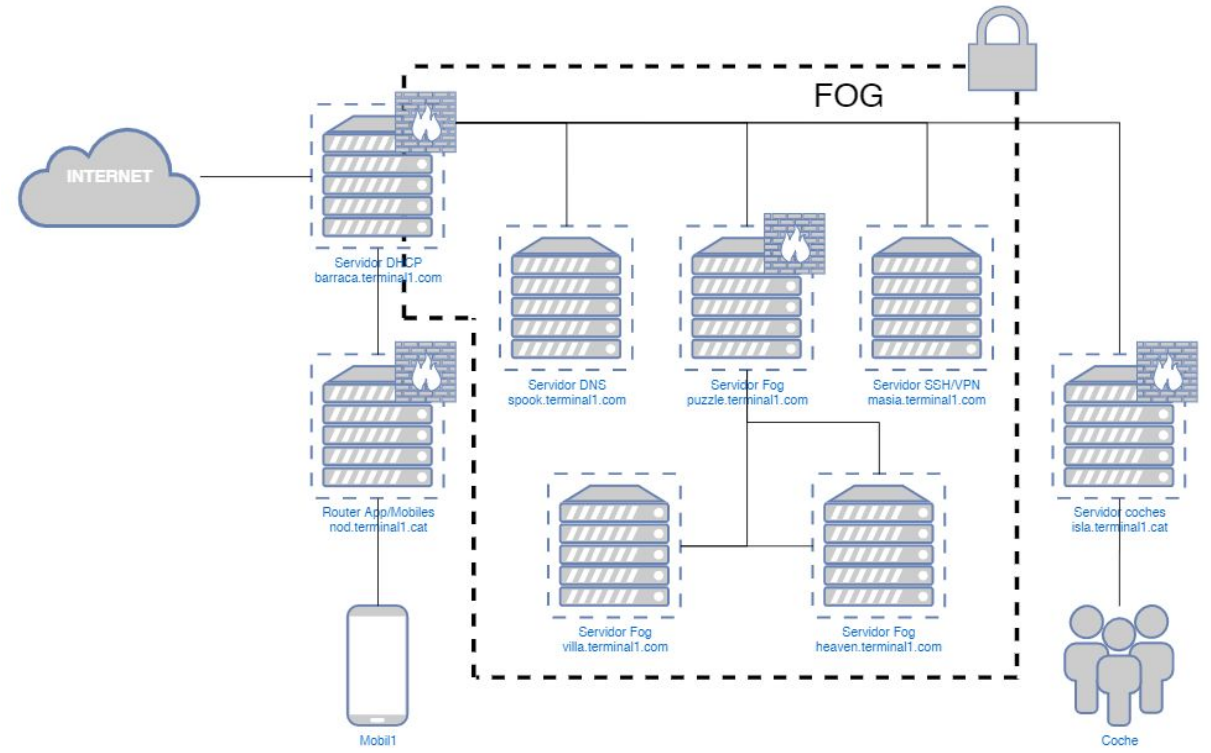
1- Una MV Cloud amb Debian 10 : Aquesta contindrà dos dockers, un per a contenir el servidor de la base de dades, i un altre per contenir el servidor de la pàgina web.
Ports necessaris: 80, 443 (Http I Https) i 27017 (MongoDB).



2- 1 MV amb Debian 10: Servidor que contindrà quatre dockers:

- Un será el servidor DHCP
- Un será el servidor DNS
- Un encarregat de la gestió dels cotxes
- Un altre encarregat de la gestió dels mòbils.

Ports necessaris: 67,68,53. Mes algún més encara per definir.



Explicació software/hardware utilitzat

Manual d'usuari

Seguretat Servidors Debian:

En seguretat, és una bona pràctica, primer de tot, assegurar el servidor on es farà el desplegament de tots els serveis de l'aplicació. Sobre aquest servidor es recomana aplicar aquestes mesures:

1. Instal·leu el que necessiteu.

La primera regla és mantenir el servidor el més fluid i lleuger possible. Instal·leu només aquells paquets que realment necessiteu. Si hi ha paquets no desitjats; purga. Com menys paquets, menys possibilitat de codi no segur.

2. Activar SELinux

Security-Enhanced Linux (SELinux) és un mecanisme de seguretat de control d'accés proporcionat pel nucli (Kernel).

SELinux ofereix 3 modes bàsics de funcionament:

- Imposant(Enforcing) : és el mode per defecte que permet activar i implementar la política de seguretat SELinux a la màquina.
- Permisiva: En aquest mode, SELinux no aplicarà la política de seguretat del sistema, només avisarà i registrarà les accions.

- Desactivat: SELinux està desactivat.

SELinux és un sistema de seguretat que permet crear "rols" o "modes d'execució" a usuaris o processos. Permet saber si algun procés o servei pot ser executat depenent del seu "context d'execució". A la pràctica, és una manera d'assegurar que qualsevol procés només tingui els accessos i permisos mínims per poder executar les seves tasques sense sortir del seu àmbit d'execució i produir canvis no desitjats.

Ara mateix, tenim una política d'accés molt limitada per les característiques dels recursos presentats pels clients, on només hi ha un usuari desenvolupador. Si bé es poden crear rols o contextos de seguretat, la configuració no és gens senzilla, i es poden crear més problemes dels que solucionen. La seguretat en aquest cas, es pot donar en altres àmbits.

Es pot gestionar des del fitxer "/ etc / selinux / config", on el podeu activar o desactivar.

Font:
https://debian-handbook.info/browse/es-ES/stable/sect.selinux.html

3. Accés segur a la consola

Hem de protegir l'accés a la consola dels servidors Linux desactivant l'arrencada des de dispositius externs com ara DVD / CD / pen USB després de la configuració del BIOS. A més, configurar el BIOS i la contrasenya del carregador d'arrencada grub per protegir aquesta configuració.

Ara mateix, amb els servidors allotjats sobre màquines virtuals en un servidor del CRAAX, entenem que el servidor en si es troba protegit dels accessos físics no desitjats.

4. Restringir l'ús de contrasenyes antigues

L'ús de contrasenyes antigues pot portar a terme vulnerabilitats en el sistema. Es recomana sempre tindre un mòdul d'autenticació PAM per limitar l'ús de contrasenyes repetides, entre altres coses.

Ara mateix, amb la configuració obtinguda dels clients tècnics, aquesta configuració no és necessària, però podrà ser útil en el futur.

PAM manual: https://wiki.debian.org/LDAP/PAM

5. Permetre només els ports necessaris al serveis prestats o necessaris:

Modificar el firewall per permetre el pas dels serveis que es faran servir dins del servidor i denegar l'accés a la resta.

En cas de Debian tenim el firewall UFW:
https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-debian-10

6. Desactivar el login de Root per SSH

No permetre l'entrada mitjançant la consola SSH al sistema fent servir l'usuari de root.
Aquesta mesura ja està presa en el nostre entorn.

7. Canvi dels ports per defecte del servei SSH

Una mesura recomanada de seguretat és canviar el port per defecte de SSH, a un altre modificant l'arxiu */etc/ssh/sshd_config* .
Aquesta configuració queda subjecte als criteris dels clients tècnics.

8. Desactivar la drecera de reiniciar el servidor.

Fent Ctrl+Alt+Delete es pot reiniciar el servidor, el que és perillós en cas que es faci accidentalment. Cal desactivar aquesta opció del sistema en el fitxer /etc/init/control-alt-delete.conf i comentar la següent línia.

```
#start on control-alt-delete
```

9. Login mitjançant claus SSH.

Cal configurar el servidor per al login amb claus SSH.

Guía:
https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-debian-10

10. Fail2Ban per a logins amb SSH

Fail2Ban modifica el firewall dinàmicament per a prohibir l'accés a les adreces que han fallat el log in un cert nombre de cops.

https://servidordebian.org/es/buster/security/brute_force_attack/fail2ban

Seguretat a nivell d'aplicació/servei:

Una de les altres parts a protegir és l'aplicació/servei que utilitzarem en el nostre projecte. Amb l'infraestructura presentada pel grup d'arquitectura i amb les Tecnologies utilitzades pels grups de desenvolupament, presentem aquestes solucions per als següents serveis:

Servidor Web:

Per al servidor web, a més del certificat digital per al ús del HTTPS, recomanem l'ús d'un WAF o "web application firewall".

Nosaltres recomanem el ModSecurity amb regles de seguretat (Core rule set, CRS) proporcionades per OWASP .

Amb aquesta configuració, protegirem el servidor d'una gran part d'atacs.

Per a la nostra infraestructura, on cada servei està desplegat sobre un Docker, hem trobat aquest tutorial que permet fer el desplegament amb docker-compose d'un servidor web Nginx , amb ModSecurity i el CRS proporcionat per OWASP.

https://janikvonrotz.ch/2020/02/26/nginx-waf-with-modsecurity-and-owasp-crs/

Un cop està desplegat el servidor Web, podem demanar un domini web per poder fer els passos necessaris per crear el certificat digital que es necessita per encriptar les comunicacions amb HTTPS.

Una proposta és fer servir algun dels dominis gratuïts proporcionats per NOIP o qualsevol altra web de les que es llisten en aquesta llista de dominis gratuïts.

Un cop adquirit el domini i registat la web, ja és pot demanar el certificat. Nosaltres hem trobat aquesta documentació per poder fer la instal·lació sobre un servidor Nginx o sobre un servidor Apache.

La particularitat de la nostra infraestructura, és el ús de Docker, el que complica el poder gestionar els certificats. Pel que es recomana el seguiment d'aquest manual, que ens proporciona aquests scripts de configuració que podem modificar per a les nostres necessitats.

Seguretat API:

Gràcies a la col·laboració del equip de desenvolupament Web, sabem que l'API funciona amb Node.js muntat sobre una infraestructura web Express.js .

Nosaltres, hem trobat diferents solucions que permeten assegurar el nostre projecte de diferents atacs maliciosos com podrien ser el SQLi(SQl injection), XSS(Cross Site Scripting), etc.

Una d'aquestes solucions és Helmet, un conjunt de petits "middlewares" que s'encarreguen de gestionar els headers de HTTP per evitar forats de seguretat.

Helmet no és tot poderós, soluciona molts dels forats de seguretat, però no tots. Per protegir-nos de la majoria de vulnerabilitats, es recomana seguir aquest manual de seguretat proporcionat per OWASP sobre seguretat en Node.js

És un manual extremadament exhaustiu, i el seguiment d'aquest, proporciona una seguretat que protegeix els serveis de pràcticament el 90% dels atacs informàtics.

Dins d'aquest manual, hi ha tots els recursos necessaris per protegir totes les parts de la infraestructura de possibles atacs. Amb la col·laboració de la resta d'equips de desenvolupament, el següent Sprint, implementarem aquelles que trobem més apremiants.

Seguretat amb Docker:

Docker és un servei que és bastant segur, les configuracions necessàries per assegurar els contenidors, tenen més a veure amb confirmar que les imatges descarregades d'internet no tinguin vulnerabilitats conegudes.

Aquí ajuntem un manual que descriu quines son les millors pràctiques a l'hora de crear contenidors.

https://snyk.io/blog/10-docker-image-security-best-practices/

Conclusions

En aquest Sprint de desenvolupament proposem solucions de seguretat que es poden prendre sobre la nostra infraestructura i serveis.

En el següent Sprint, conjuntament amb el equip de desenvolupament, implementarem aquestes tecnologies i tècniques per assegurar el producte.

Documentació adicional, enllaços interessants

Com crear un servei de Node.js amb MongoDB:
https://medium.com/@kahana.hagai/docker-compose-with-node-js-and-mongodb-dbdadab5ce0a

Com forçar a la API l'ús de HTTPS:
https://stackoverflow.com/questions/8605720/how-to-force-ssl-https-in-express-js/11033289#11033289

Llistat de les vulnerabilitats més freqüents en desenvolupament web:

https://www.cloudflare.com/learning/security/threats/owasp-top-10/

Com protegir un servidor de Producció:

https://medium.com/viithiisys/10-steps-to-secure-linux-server-for-production-environment-a135109a57c5

L'aplicació mòbil

Descripció general

En aquest Sprint s'ha prioritzat la comunicació amb altres equips i l'implementació de les parades dels cotxes al mapa.

Explicació software/hardware utilitzat

Manual d'usuari

S´ha implementat el mapa a la app amb les parades dels cotxes

Conclusions

El resultado ha sido bastante intuitivo y completo.

Base de Dades

Descripció general

En aquest sprint se ha modificat la base de dades per peticions dels altres grups

Col·lecció administrators

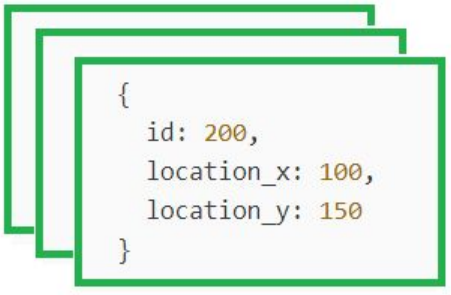
- S'han eliminat els atributs phone i bithdate perquè donaven alguns problemes o ames eren atributs innecessaris per a un admin.
- S'ha afegit l'atribut job

Col·lecció flights

- Hem afegit un nou camp dins la col·lecció flights amb un nom 'state', aquest nou atribut ens indicarà l'estat del vol.

Col·lecció stops

- Aquest és una nova col·lecció que serveix per a que els cotxes tinguin un destí al que puguin anar, ja que hem decidit que els cotxes no poden parar en qualsevol lloc, han de parar en les diferents parades(stops) que hi haurà en la terminal.



un exemple de la col·lecció stops.

Atribut location:

Abans l'atribut location era un string amb les coordenades x i y, en aquest sprint les hem canviat per dos atributs de nom location_x i location_y, ara son enters i no strings.

col·lecció node(cotxe):

state es un atribut de node(cotxe), l'hem canviat de string a int

0 En repació o no funciona
1 Disponible
2 Ocupat
3 Carregant

col·lecció passengers:

Hem afegit un nou atribut amb el nom url_image, aquest nou atribut conte la url de la foto del passatger.

Un altre atribut afegit es el de el type_user, quest atribut ens indica quin tipus d'usuari és, es un valor numero de 0 a 3:

0 usuari corrent
1 usuari or
2 usuari platino
3 usuari traballador

codi dels passatgers:

Hem canviat els codis a variables numeriques, ja que abans eren strings.

col·leccio boarding_passes:

Hem afegit una nova col·lecció amb el nom boarding_passes que fa referencia a la targeta d'embarcament, amb aquesta col·lecció un passatger podrà veure la informació del seu vol així com abordar amb aquesta informació.

También hemos generado un nuevo archivo que contiene suficientes datos de prueba para la Base de datos, así podemos hacer más pruebas para asegurar la robusteza de de la API y APP.

Explicació software/hardware utilitzat

MongoDB

Manual d'usuari

//Interaccions que pot realitzar l'usuari amb el projecte actual.

Podem veure les col·leccions que tenim amb la comanda

show collections

```
> show collections
administrators
boarding_passes
flights
nodes
operators
passengers
shops
stops
```

Podem veure que s'han afegit dos noves col·leccions que son stops i boarding_passes

cadascun amb dades de proves per veure les dades que enmagatzmen tenim que utilizar la comanda bd.nom_col·lecció.find()

```
> db.boarding_passes.find()
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d7e"), "id_hash" : 75092425388432490000, "seat" : "1A", "id_passenger" : 56764564, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d7f"), "id_hash" : 55853654337661990000, "seat" : "1B", "id_passenger" : 3344699, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d80"), "id_hash" : 21636962957484995000, "seat" : "1C", "id_passenger" : 45364678, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d81"), "id_hash" : 60119888056922500000, "seat" : "1D", "id_passenger" : 56764564, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d82"), "id_hash" : 73504170301113360000, "seat" : "2A", "id_passenger" : 3344949, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d83"), "id_hash" : 75092425388432490000, "seat" : "2B", "id_passenger" : 957832, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d84"), "id_hash" : 89735804919991140000, "seat" : "2C", "id_passenger" : 4334499, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d85"), "id_hash" : 58146554997115600000, "seat" : "2D", "id_passenger" : 9275399, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d86"), "id_hash" : 27398768418673500000, "seat" : "3A", "id_passenger" : 454104, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d87"), "id_hash" : 54918526570878290000, "seat" : "3B", "id_passenger" : 909736885, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d88"), "id_hash" : 18616712873936028000, "seat" : "3C", "id_passenger" : 91488945, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d89"), "id_hash" : 74228460332806280000, "seat" : "3D", "id_passenger" : 863077323, "flights" : "VL 203" }
{ "_id" : ObjectId("5ebd1ace8eeeb82159b01d8a"), "id_hash" : 30557208176679860000, "seat" : "4A", "id_passenger" : 459131612, "flights" : "VL 203" }
>
```

Son les dades de prova de boarding_passes

```
> db.stops.find()
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d68"), "id" : 100, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d69"), "id" : 200, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6a"), "id" : 300, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6b"), "id" : 400, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6c"), "id" : 500, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6d"), "id" : 600, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6e"), "id" : 700, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d6f"), "id" : 800, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d70"), "id" : 900, "location_x" : 100, "location_y" : 150 }
{ "_id" : ObjectId("5ebd1acd8eeeb82159b01d71"), "id" : 950, "location_x" : 100, "location_y" : 150 }
```

I aquí les dades de stops.

Conclusions

En aquest sprint s'ha modificat bastan la base de dades així com s'ha afegit 2 noves col·leccions, al principi fa ser un gran canvi perquè no ho havien plantejat la possibilitat d'afegir aquestes noves col·leccions però amb això ja tenim una base de dades més robusta i que dóna més informació i ajuda als passatgers.