**Trabajo Final**

Kevin Alejandro Vasco Hurtado

Escuela de Ingenierías, Universidad Pontificia Bolivariana

Telemática

Leonardo Betancur Agudelo

3 de mayo, 2022

Contents

[Contents 2](#_Toc102466373)

[Objetivos 3](#_Toc102466374)

[Descripción General 3](#_Toc102466375)

[Topologia y Direccionamiento 5](#_Toc102466376)

[EC2 Name 6](#_Toc102466377)

[OS Images and Architecture 6](#_Toc102466378)

[Instance Type 7](#_Toc102466379)

[Key Pair 7](#_Toc102466380)

[Network Settings 8](#_Toc102466381)

[Configure Storage 10](#_Toc102466382)

[Firewall – Reglas de Seguridad 10](#_Toc102466383)

[EC2 Extra Information 12](#_Toc102466384)

[Services 13](#_Toc102466385)

[DNS 13](#_Toc102466386)

[Web 20](#_Toc102466387)

[Comprobacion Funcionamiento en Docker 21](#_Toc102466388)

[Comandos para Dockerfile 26](#_Toc102466389)

[Instalacion y Despliegue 26](#_Toc102466390)

[Extra para mi 27](#_Toc102466391)

Objetivos

* Emplear las nuevas tecnologías para desarrollar un proyecto comercializable usando los conocimientos del curso.
* Evaluar la capacidad de usar nuevas tecnologías para resolver problemas y ofrecer servicios.
* Demostrar la capacidad de implementar servicios telemáticos en aplicaciones reales.

Descripción General

Diseñe un servicio telemático a nivel de producción que responda al dominio www.miemprendimiento.com que suministre servicio WEB consumible por usuarios móviles y de equipos de escritorio, empleando la tecnología de contenedores, puede usar el sistema operativo de su predilección, las plataformas y paquetes de software que desee siempre y cuando supla el servicio completo propuesto. Tipo de servicio: la aplicación a desarrollar puede usar cualquiera de las siguientes opciones, escoja una de ellas y desarrolle un aplicativo sencillo y simple, se recomienda que use los ejercicios hechos en clase para construir la solución:

* Un servicio de red social.
* Un servicio de periodico local en linea.
* Un portal para manejar blog de usuarios.
* Un portal Wiki sobre un tema de su predileccion.

Características del servicio: debe ser un servicio escalable que debe brindar un servicio consumi- ble con los siguientes requerimientos:

* El servidor que presta la principal aplicación es un servidor WEB desplegado bajo el concepto de desarrollo continuo.
* El desarrollo está almacenado en un repositorio (el código fuente del desarrollo), el nivel de experiencia de usuario y desarrollo gráfico NO importa si es simple o elaborado.
* El desarrollo debe estar documentado y tener un archivo de instrucción README.MD que explique como desplegar el servicio en el contenedor para un servidor de producción.
* El estudiante debe entregar solo un archivo (El Dockerfile), que tenga automatizado todo el proceso de construcción de la imagen con el servicio desplegado.
* El servicio puede ser escrito en cualquier lenguaje, html o php o python o el lenguaje que ud considere que puede manejar mejor, recuerde que la idea principal del proyecto es usar un despliegue usando contenedores.

Topologia y Direccionamiento

La practica se hara usando los servicio de AWS y tendra la siguiente configuracion.

A picture containing diagram

Description automatically generated

EC2 Name

Application

Description automatically generated with medium confidence

OS Images and Architecture

Graphical user interface, text, application, email

Description automatically generated

Instance Type

Graphical user interface, text, application, email

Description automatically generated

Key Pair

Graphical user interface

Description automatically generated with low confidence

Network Settings

Seleccionamos el botón “Edit” para abrir el apartado para agregar las reglas de seguridad.

Graphical user interface, text, application, email

Description automatically generated

Estas son las configuraciones del “Network Settings”.

Graphical user interface, text, application, email

Description automatically generated

Configure Storage

Dejaremos esto tal cual.

Graphical user interface, text, application, email

Description automatically generated

Firewall – Reglas de Seguridad

SSH

Graphical user interface, text, application, chat or text message

Description automatically generated

DNS (UDP)

Graphical user interface, text, application, chat or text message

Description automatically generated

DNS (TCP)

Graphical user interface, text, application, chat or text message

Description automatically generated

HTTP

Graphical user interface, application

Description automatically generated

HTTPS

Graphical user interface, application

Description automatically generated

EC2 Extra Information

A screenshot of a computer

Description automatically generated

Services

Los pasos mostrados a continuacion son de demostracion. Se mostraran los mismos pasos, pero ya realizados en docker para verificar su funcionalidad y de nuevo, pero solamente los comandos en el Dockerfile.

Antes de iniciar con la instalacion de los servicios se actualizara el sistema

**sudo apt-get update**

DNS

Para instalar el servicio DNS

**sudo apt-get install bind9 -y**

Ante de realizar las configuraciones necesarias haremos una copia de seguridad de un archivo necesario para los servicios de DNS

**sudo cp /etc/bind/named.conf /etc/bind/named.conf.old**

De ante mano se necesita tener la IP de la maquina y el dominio a usar.

**IP privada del EC2: Varia**

**Dominio: www.miemprendimiento.com**

Modificamos el anterior archivo **named.conf** con lo siguiente:

**zone "miemprendimiento.com" {**

**type master;**

**file "mapadirecto.miemprendimiento.com";**

**};**

**zone "29.31.172.inaddr.arpa" {**

**type master;**

**file "inverso.172.31.29";**

**};**

A screenshot of a computer

Description automatically generated

Verificamos que el archivo este escrito correctamente con el comando **named-checkconf**. En dado caso que arroje un mensaje, hay un error de sintaxis, en el caso contrario no hay problemas con el archivo.

Ahora se crea los archivos dichos en **named.conf**.

**nano /var/cache/bind/miemprendimiento.com** & **nano /var/cache/bind/inverso.172.31.29**

En el archivo **miemprendimiento.com** se agregara la configuracion necesaria para la **zona directa**.

**$ORIGIN miemprendimiento.com.**

**$TTL 86400 ;**

**@ IN SOA servidor postmaster (**

**202204301 ;**

**6H ;**

**1H ;**

**2W ;**

**3H ;**

**)**

**IN NS www**

**www IN A 172.31.29.96**

A screenshot of a computer

Description automatically generated

Verificamos que el archivo quedo correcto.

**named-checkzone miemprendimiento.com /var/cache/bind/mapadirecto.miemprendimiento.com**

Text

Description automatically generated

Haremos los mismos pasos con el archivo **inverso.172.31.29**

**$ORIGIN 29.31.172.inaddr.arpa.**

**$TTL 86400 ;**

**@ IN SOA servidor postmaster (**

**202204301 ;**

**6H ;**

**1H ;**

**2W ;**

**3H ;**

**)**

**IN NS wwww.miemprendimiento.com.**

**96 IN PTR www.miemprendimiento.com.**

A screenshot of a computer

Description automatically generated

Verificamos que el archivo quedo correcto.

**named-checkzone 29.31.172.inaddr.arpa /var/cache/bind/inverso.172.31.29**

Text

Description automatically generated

Verificaremos que todo esté funcionando correctamente. Primero, reiniciamos el servicio DNS.

**sudo service bind9 restart**

Luego, validaremos que esté funcionando el servicio.

**sudo service bind9 status**

A screenshot of a computer

Description automatically generated with medium confidence

Ahora validamos el correcto funcionamiento.

**dig @172.31.29.96 www.miemprendimiento.com**

Text

Description automatically generated

Web

El servicio web se implementara usando dash, o sea Python. Para implementar su funcionalidad y los demas servicios necesarios para el funcionamiento del servicio, se instalan las demas librerias.

**sudo apt install python3 -y**

**sudo apt install python3-pip -y**

**pip3 install dash**

**pip3 install scipy**

**pip3 install dash\_bootstrap\_components**

**pip3 install pandas**

Comprobacion Funcionamiento en Docker

Se actualiza el sistema.

A screenshot of a computer

Description automatically generated with medium confidence

Se instala el servicio de Docker.

A screenshot of a computer

Description automatically generated with medium confidence

Se jala la imagen que vamos a usar, en este caso un Ubuntu al raz.

A screenshot of a computer

Description automatically generated

Dentro del contenedor actualizamos el sistema y configuramos el servicio DNS como anteriormente.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Comprobamos que este correctamente funcionando.

Text

Description automatically generated

Instalamos la librerias necesarias.

Pasamos el archivo app.py al docker.

A screenshot of a computer

Description automatically generated

Anteriormente al abrir el docker no le indique que mapeara el puerto 80 del contenedor al 80 de la maquina local, en este caso la instancia, por lo tanto salimos del contenedor y volvemos a ejecutar este mismo.

**sudo docker run -it -p 80:80 exe:prueba**

Ya hehco esto y estando dentro del contenedor ejecutamos el archivo python.

**python3 app.py**

Verificamos su funcionamiento.

Text

Description automatically generated

Comandos para Dockerfile

Ya habiendo comprobado el funcionamiento de los servicios de DNS y Web, vamos a crear el [Dockerfile](https://github.com/KevinVasco/Trabajo_Final_Telematica/blob/main/Dockerfile) con los comandos necesarios para la instalacion de los servicios, como tambien la descarga de los archivos necesarios para el funcionamiento de los servicios.

Instalacion y Despliegue

Para saber como instalar y desplegar el contenedor y sus servicios redirigase a este enlace del repositorio y leer el [README.md](https://github.com/KevinVasco/Trabajo_Final_Telematica/blob/main/README.md).

Extra para mi

En el caso que en el container no este funcionando el DNS:

**service named restart** en vez de **service bind9 restart**

**dig @172.17.0.2** [**www.miemprendimiento.com**](http://www.miemprendimiento.com)

Instalar ifconfig

**sudo apt install net-tools**

Instalar nano

**sudo apt install nano**