

Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Redes

PaaS.net

Desarrollado por

- ##### Manuel Ruiz Androvetto (2017074102)
- ##### Kevin Segura Rojas (2017153767)

Alajuela, I-S 2020

Introducción

En este proyecto consta de montar un servidor de servicios seguros por medio de la virtualización. Algunos de los servicios requerirán el ser cifrados, por lo que se deberá instalar una entidad certificadora que firme sus propios certificados. Además, también es necesario montar una central telefónica PBX, que deberá poder realizar llamadas a través de CallMyWay, con al menos dos clientes (computadora de escritorio y teléfono), un cuarto de conferencias, horario establecido para las llamadas, manejo de extensiones y transferencias de llamadas. Finalmente, se deberá realizar una investigación sobre el proceso de instalación de OpenShift, y elementos relacionados a esta herramienta.

Ambiente de Desarrollo

El trabajo se desarrolla en los siguientes ambientes:

- Ubuntu 18.04 LTS: SO operativo principal utilizado por uno de los integrantes.
- AWS Educate: Herramienta principal donde se montará el servidor virtual con los servicios.
- Elastix server (basado en CentOS)
- LinPhone

Diseño

Proyecto:

DataBase, WebServer y CMS:

Iniciando con el WebService, se estudió posibles soluciones, donde la elegida fue AWS Educate por sus facilidades y la ayuda que el TEC brinda para que uno pueda utilizar esta herramienta. Se utilizó el servicio integrado "EC2" para la creación de una instancia virtual que será el servidor donde montar los servicios, esta decisión se hizo tomando en cuenta las características de capacidad informática en la nube segura y de tamaño modificable.

Para la base de datos se utilizó el servicio RDS, que cuenta con una gran facilidad para la creación, utilización y escalado de múltiples tipos de bases de datos. Hasta este punto eran servicios brindados por el mismo AWS, pero ya para la instalación de los servicios se decidió consultar la página <https://hub.docker.com/> en busca de las mejores opciones, para el CMS se decidió utilizar Wordpress al ser de los más reconocidos y ya estar familiarizado con él.

VPN:

Para esta labor se utilizó OpenVPN, específicamente una versión funcional en Ubuntu 18.04 extraída de git.io/vpn.

Acceso al CMS con nombre de dominio (DNS):

Gracias a EC2, que ya cuenta con un generador automático de dominios utilizables en la red, esta labor se pudo hacer sin problemas desde la instancia virtual generada.

2 Servicios con docker-compose:

Por temas de mayor entendimiento de la herramienta por el estudiante, a parte que fueron de los que mejor información se pudo encontrar, se decidió utilizar los servicios de owncloud y Email Server.

Central telefónica:

Para la central telefónica, se trabajará dos elementos principales por aparte que funcionan de manera simbiótica.

Elastix Server: El servidor Elastix es un servidor de comunicaciones basado en CentOS que combina distintos servicios como correo electrónico, fax y PBX. Para el caso preciso de este proyecto, se trabaja principalmente con el PBX. La PBX (*Private Branch Exchange*) es una central telefónica utilizada típicamente a lo interno de una empresa. El servidor es utilizado como ambiente en donde se aloja y funciona la central telefónica. Dentro de la PBX se configuran servicios como menú, correo de voz, horario de atención, desvío de llamadas, transferencias de llamadas, tonos de espera, grabaciones de correo de voz, entre otras cosas. Los servicios del PBX pueden ser modificados a través de la línea de comandos del servidor o bien desde un navegador de internet conectándose directamente al servidor.

LinPhone: Linphone es un softphone (software que simula un teléfono convencional) que utiliza la tecnología VoIP o voz sobre IP para la comunicación. Para la realización del proyecto, se requiere de una línea de la empresa CallMyWay para poder realizar llamadas a usuario fuera de la red en la que establecemos la central telefónica. Linphone fue escogido como softphone porque tiene clientes para sistemas operativos móviles como Android y iOS y también para los sistemas

Instalación de Servicios

- **Web Server:** Antes que nada, se creo un security Group para el manejo del tráfico, para este ejercicio se permitió todo el tráfico para mayor facilidad. Ya con esto se procedio a especificar las características de la instacia, se escogió hacer una máquina virtual de procesamiento moderado con 15 gigas de almacenamiento, ya que se considero que no es mucho lo que se necesitará a la vez que se deja un margen para que tampoco falte. Finalmente, se crearon una llaves privada SSH para conectarse a la instancia a través de mi máquina local y agregar los servicios por la terminal.
- **MySQL(RDS):** Se busco el servicio dentro de AWS, dentro de él esta la opción "create", ahí pedirá la configuración deseada. Se escogio MySQL como se solicita en el proyecto, el Free Tier que era el que brindaba las condiciones más comodas para el servicio, se nombro como "cmsdatabase" con usuario maestro "admin" y password "123queso123", después de que levanto se procedio a ligarla a la instancia virtual EC2.
- Antes que nada se tuvo que instalar Docker y Docker-compose en la máquina virtual a la vez que se dieron los permisos correspondientes.
- **Wordpress(CMS):** Primer servicio docker, después de hacer el pull, inicialmente se corrió con el comando "sudo docker run -d -p 80:80 -e WORDPRESS_DB_HOST=endpoint:puerto(3306) -e WORDPRESS_DB_USER= -e WORDPRESS_DB_PASSWORD= wordpress", pero luego de investigación se logro ejecutar con docker-compose. Para acceder y usar este servicio se puede hacer a través de dominio <http://ec2-34-232-78-241.compute-1.amazonaws.com/> por el puerto 80.
- **OwnCloud:** Se descargo desde https://raw.githubusercontent.com/owncloud/docs/master/modules/admin_manual/examples/installation/docker/docker-compose.yml, se hizo la respectiva configuraci3n para que funcionara, y finalmente se puso en marcha con "sudo docker-compose up -d". Para acceder al servicio es por medio de la misma direcci3n pero por el puerto 8080.
- **Mail Server:** Se utilizo el servidor de correos poste.io, también encontrado en <https://hub.docker.com/>, se hizo el pull respectivo, y se creo el contenedor docker con el comando "docker run --net=host -e TZ=Europe/Prague -v /your-data-dir/data:/data --name "mailserver" -h "mail.example.com" -t analogic/poste.io", y finalmente se pone a ejecutar con el comando "sudo docker start id-contenedor". Para acceder es por el mismo dominio de la instancia, mediante el puerto 80 o el 443.
- **VPN:** Este se instalo en la propia instancia virtual, por medio de los comandos "wget https://git.io/vpn -O openvpn-install.sh" y "sudo ./openvpn-install.sh", ya una vez instalado es necesario estar en la carpeta donde se ubique el archivo .pem con las claves SSH de la instancia y ejecutar el comando "sudo scp -i [archivo.pem] ubuntu@[direcci3n pública]:/home/ubuntu/myvpn.ovpn /home/william/Escritorio/" para crear el archivo que servirá para la conexi3n con la VPN del servidor. Finalmente se usa "sudo systemctl start openvpn-server@server.service" para activar el servicio, para que los cliente se conecten a este se puede hacer con el comando "sudo systemctl start openvpn@client" después de haber copiado el archivo de conexi3n.
- **OpenSSL:** Se hizo pull al docker de este servicio, a continuaci3n se creo el certificado del server con el comando "docker run -it -v \$(pwd):/export frapsoft/openssl req -nodes -new -newkey rsa:2048 -sha256 -out /export/cert.pem", el cual da como resultado un archivo .pem con el certificado. Para hacer uso del servicio es directamente desde la instancia con el comando "docker run -it --entrypoint /bin/ash frapsoft/openssl".
- **PBX:** Para la instalaci3n de la PBX, es necesario descargar un servidor Elastix de internet. La versi3n por utilizar puede ser la más reciente. Esta se puede instalar como una máquina virtual en un software como VirtualBox o VMWare. Elastix está desarrollado sobre una distribuci3n de CentOS y cuenta con un instalador que se encarga de establecer los ajustes principales del servidor antes de levantarlo. Entre los ajustes más importantes a hacer para una buena conexi3n del servidor, está la de establecer una direcci3n IPv4 estática fija para el servidor. Esto puede facilitar la conexi3n del servidor y los clientes ya que se evita una cambio en la direcci3n del servidor cuando este se apague y encienda nuevamente. Se debe establecer una contraseña para el usuario root que será utilizada al encender el servidor y al conectarse desde un navegador al mismo. Al finalizar la instalaci3n el servidor pedirá una contraseña para la base de datos de MySQL y para el servicio freePBX de la central telef3nica. A continuaci3n es necesario ingresar al servidor y para comprobar que esté conectado a la red, se puede hacer uso del comando `ifconfig`.

Por otra parte, se debe ir a un navegador de internet para acceder al servidor utilizando la interfaz gráfica que ofrece Elastix. Para esto basta con conectarse a <https://<direcci3n-IP-servidor>> y colocar el nombre de usuario admin con la contraseña seteada anteriormente.

Una vez en el servidor de Elastix, se deben crear extensiones para cada uno de los clientes por utilizar. Esto se realiza dentro de la opción de Extensiones en la pestaña de PBX. A cada extensi3n creada se le agrega un nombre identificador, un número de extensi3n, se le habilita el correo de voz y se agrega una contraseña para el correo de voz. Además, se borra la casilla de secret y por último en las opciones de Email attachment, Play CID y Play Envelope se setea en Si. Al terminar la configuraci3n es importante guardar y aplicar la nueva configuraci3n

Una vez listas las extensiones, se crean usuarios para cada extensi3n en la opción de Users dentro de la pestaña de System. A cada usuario se le asigna una contraseña y su respectivo nombre y extensi3n.

Para crear un correo de voz, es necesario ir a la secci3n de correo electr3nica y establecer un correo electr3nico por cada extensi3n creada (el correo electr3nico puede ser ficticio). Una vez creado el correo y habilitado el correo de voz en las extensiones, un usuario puede llamar al *97 para escuchar los mensajes almacenados en su buz3n.

Para agregar una nueva grabaci3n a utilizarse en el correo de voz o menú principal, es necesario crear un IVR. Si esta IVR va a ser utilizada como menú principal, por ejemplo, se debe crear una extensi3n. A esta extensi3n se le asigna un nombre, una extensi3n propia y se le elimina la contraseña o "secret". Una vez listo, se crea una grabaci3n agregando la extensi3n desde la cual grabar en el menú de grabaciones. Luego, desde el softphone se hace una llamada a *77 y se graba el audio deseado. Al finalizar, se vuelve al PBX para guardar la grabaci3n. Después se abrí el menú de IVR y se crea una nueva IVR asignándola un nuevo nombre, el audio recientemente grabado y guardado y por último, indicando las acciones a realizar de acuerdo a los botones presionados durante la llamada.

- **LinPhone:** es un softphone que se puede descargar directamente desde su página web. Una vez descargado el programa, se instala y se agrega una nueva cuenta a la cuál se le asigna la extensi3n respectiva del dispositivo por usar y la IP del servidor de Elastix. Cuando se desee realizar una llamada, basta con marcar el número deseado desde linphone y este realizará la llamada.

Actividades Realizadas por Estudiante

Kevin Segura:

- Búsqueda de informaci3n para la investigaci3n de OpenShift. Horas invertidas: 2:30. Fecha: Lunes 3 de Agosto.
- Búsqueda de un poco más de informaci3n y escritura de parte de la investigaci3n. Horas invertidas: 3. Fecha: Miercoles 5 de Agosto.
- Estudio de opciones para montar el Web Server. Horas invertidas: 1:30. Fecha: Lunes 10 de Agosto.
- Estudio básico de como usar AWS y posibles formas de montar los servicios. Horas invertidas: 2. Fecha: Martes 11 de Agosto.
- Creaci3n de la instancia EC2, creaci3n de la base de datos con RDS e investigaci3n de dockers para los servicios descritos. Horas: 4:30. Fecha: Jueves 13 de Agosto.
- Instalaci3n y comprobaci3n del funcionamiento de los servicios Wordpress y OwnCloud. Horas invertidas: 3. Fecha: Viernes 14 de Agosto.

- Instalación de los servicios OpenSSL y OpenVPN, además de pruebas. Horas invertidas: 2:30. Fecha: Sabado 15 de Agosto.
- Instalación y pruebas del servicio Mail Server, además de investigar como utilizar docker-compose a partir de los contenedores creados. Horas invertidas: 4. Fecha: Lunes 17 de Agosto.
- Continuación de la creación de los archivos docker-compose. Problemas con la autenticación con OwnCloud y con la creación del archivo docker-compose para el Mail Server, escritura de la documentación. Horas invertidas: 6. Fecha: Martes 18 de Agosto.
- Horast totales invertidas: 29 horas.

Manuel Ruiz:

- Investigación sobre PBX: Martes 4 de agosto, 1.5 Horas
- Investigación sobre Elastix: Martes 4 de agosto, 1.5 horas
- Seteo de servidor de Elastix: Viernes 14 de agosto, 5 horas
- Inicio de investigación de OpenShift: Viernes 14 de agosto, 3 Horas
- Instalación de OpenShift: lunes 17 de agosto, 5 Horas
- Creación de servicios de PBX: domingo 17, lunes 17 y martes 18 de agosto: 8 horas.

Horas totales: 24 horas

Autoevaluación

Kevin Segura:

Se termino la investigación, mientras que el servidor web quedo habilitado para el uso del CMS-Wordpress con su respectiva base de datos, un servidor de correos electronicos, OwnCloud, conexión por VPN y un domino conocido por los servidores DNS. La principal limitante sin dudas fue el tiempo, ya que había que investigar mucho a la vez que se tenían trabajos y exámenes finales de otros cursos, también el desconocimiento de muchas herramientas como AWS, Docker, y el uso de los servicios como tal, provocaron diversos problemas que alerdaron el proceso. En este caso no se escribio código como tal por lo que no se requirió la herramienta git, y todo el proceso fue almacenado en los respectivos ambientes de trabajos.

AutoEvaluación:

- Aprendizaje de Servicios: 10
- Aprendizaje de Central Telefónica: 8
- Aprendizaje de laC(VPN): 9
- Aprendizaje de PaaS(Plataforma como servicio): 10

Manuel Ruiz:

No fue posible establecer un servicio haciendo uso de OpenShift, no obstante su instalación se completó correctamente. Por otra parte, la central telefónica tiene un comportamiento apropiado, sin embargo existe un problema al recibir llamadas. Las llamadas son colgadas en el primer segundo en que se reciben.

AutoEvaluación:

- Aprendizaje de Servicios: 8
- Aprendizaje de Central Telefónica: 10
- Aprendizaje de laC(VPN): 9
- Aprendizaje de PaaS(Plataforma como servicio): 9

Lecciones Aprendidas

- Se aprendio como montar un servidor web para servicios con AWS, montar y utilizar algunos servicios a la vez que estos se implementaban con la herramienta docker.
- Se aprendió un poco de OpenStack, kubernetes y del uso de docker, su gran utilidad para la portabilidad y retrocompatibilidad.
- Se pudo observar que no siempre todo es escribir código, más aún en redes, es requerido tener dedicación para aprender a utilizar las herramientas y servicios que están disponibles, y de saberse usar correctamente, representan un poderoso aliado para nuestras futuras labores como profesionales.
- Se aprecio el proceso de cómo un servicio puede ser puesto en marcha de manera simple, pero a su vez gracias a la gran infraestructura e implementación de Internet es puesto a marcha para el uso de cualquier usuario que lo necesite.
- Las centrales telefónicas ofrecen una enorme cantidad de servicios para las empresas y son fáciles de implementar. La interfaz es intuitiva y rápida de comprender.
- Existen muchos softPhones gratuitos para distintas plataformas tanto móviles como computadoras de escritorio que ofrecen la oportunidad de establecer comunicaciones a través de internet desde cualquier parte del mundo con una conexión a la red.
- A pesar de girar la investigación en torno a OpenShift, Rancher pareciera ser una mejor opción por su comparación directa a partir de un estudio realizado en internet. OpenShift por su parte, tiene la ventaja del respaldo que una empresa como IBM le puede dar, además de las conexiones directas con socios estratégicos de la multinacional.

Bibliografía

- AWS.(2020). Encuentre los tutoriales prácticos para sus necesidades de AWS. Recuperado de: <https://aws.amazon.com/es/getting-started/hands-on/>
- Martinez, P.(2018). Comandos Docker y Docker-Compose. Recuperado de: <https://medium.com/@pablomp/comandos-docker-y-docker-compose-636b119b4e5>
- Git Hub.(2020). Wordpress. Recuperado de: https://hub.docker.com/_/wordpress
- Git Hub.(2020). owncloud. Recuperado de: https://hub.docker.com/_/owncloud
- Git Hub.(2020). analogic/poste.io. Recuperado de: <https://hub.docker.com/r/analogic/poste.io>
- Git Hub.(2020). frapsoft/openssl. Recuperado de: <https://hub.docker.com/r/frapsoft/openssl>
- Poster.io.(2020). Getting started. Recuperado de: <https://poste.io/doc/>
- Docker Tips.(2020). Aprendiendo a utilizar Docker Compose. Recuperado de: <https://dockertips.com/utilizando-docker-compose>
- nixCraft.(2020). Ubuntu 18.04 LTS Set Up OpenVPN Server In 5 Minutes. Recuperado de: <https://www.cyberciti.biz/faq/ubuntu-18-04-lts-set-up-openvpn-server-in-5-minutes/>