



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

TAREA 1

VICENTE FUENZALIDA MARÍN

IIC2173 – ARQUITECTURA DE SISTEMAS DE SOFTWARE (II/2017)

PROFESOR: HANS FINDEL

10 DE SEPTIEMBRE DE 2017

DESCRIPCIÓN DE LA APLICACIÓN

Creé una aplicación utilizando el framework Express.js sobre Node.js, utilizando una base de datos no relacional MongoDB para la capa de persistencia.

La aplicación es sencilla, simplemente maneja dos rutas ('/' y 'list'), y cada vez que se maneja una request a cualquiera de las dos, se almacenan datos de dicha request en la base de datos.

Para la base de datos, se utilizó el servicio DaaS *mLab*, que provee un tier gratuito.

Para el deploy de la aplicación, escogí el IaaS Elastic Compute Cloud de *Amazon Web Services* (EC2), ya que me pareció interesante aprender al respecto (nunca lo había utilizado).

El servidor corre en una instancia de tipo t2.micro (gratuita) que corre Ubuntu 16.04. La aplicación se puede ver en la siguiente URL:

ec2-13-59-118-191.us-east-2.compute.amazonaws.com

La configuración del servidor consistió en:

- Habilitar los puertos mediante los que el servidor puede recibir o enviar información (ya sea por HTTP u otro protocolo). AWS provee una interfaz web para configurar esto.
- Mediante SSH, conectarse a la instancia provista por EC2 e instalar el software necesario para que la aplicación se pueda ejecutar correctamente (python, node, npm, git, etc).
- Clonar el repositorio donde tenía alojada mi aplicación (<https://github.com/vjfuenzalida/IIC2173-tarea1>) a la máquina (instancia EC2).
- Instalar las dependencias/paquetes requeridos por la aplicación (npm install).
- Redirigir el tráfico entrante por el puerto 80 hacia el puerto 3000 (escogido por mí y configurado en la aplicación Express).
- Iniciar la aplicación como un *job* en *background* para que sea 'permanente' utilizando el comando 'nohup' (ya que de otro modo al cerrar la conexión SSH, el proceso de la aplicación se terminaba por la señal SIGHUP).

REFERENCIAS

En su mayoría, recurrí a tutoriales de diverso tipo (videos, blogs, tutoriales oficiales) y a la documentación disponible tanto de los paquetes de *node* utilizados, como de los servicios necesarios para la aplicación misma (EC2, mLab).

A continuación, incluyo un listado de algunos de los recursos consultados:

- [HTTPS://WWW.YOUTUBE.COM/WATCH?V=WxhFQ64FQzA](https://www.youtube.com/watch?v=WxhFQ64FQzA)
- [HTTPS://DEVELOPER.MOZILLA.ORG/ES/DOCS/LEARN/SERVER-SIDE/EXPRESS_NODEJS](https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs)
- [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/LEARN/SERVER-SIDE/EXPRESS_NODEJS/TUTORIAL_LOCAL_LIBRARY_WEBSITE](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Tutorial_local_library_website)
- [HTTPS://SCOTCH.IO/TUTORIALS/MEAN-APP-WITH-ANGULAR-2-AND-THE-ANGULAR-CLI](https://scotch.io/tutorials/mean-app-with-angular-2-and-the-angular-cli)
- [HTTPS://WWW.NPMJS.COM/](https://www.npmjs.com/)
- [HTTP://DOCS.AWS.AMAZON.COM/AWSEC2/LATEST/USERGUIDE/EC2_GETSTARTED.HTML](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)