

# **Modelo de una plataforma de aprendizaje (LMS) robusta y en alta disponibilidad para apoyar el proceso de enseñanza en instituciones de educación.**



**Dr. Carlos Enrique Montenegro Marín**

*Facultad de Ingeniería*



**Dr. Paulo Alonso Gaona García**

*Facultad de Ingeniería*



**Ing. Fernando Torres Naranjo**

*Oficina Asesora de Sistemas*

**Universidad Distrital Francisco José de Caldas, Bogotá, Colombia**

## **Introducción**

Dadas las actuales condiciones sanitarias que vivimos en todo el mundo a causa de la pandemia del COVID, todas las instituciones se han visto abocadas a usar herramientas tecnológicas que suplan la presencialidad en los procesos de enseñanza, en estos grandes esfuerzos que se han hecho los sistemas de gestión del aprendizaje (más conocidos como LMS por sus siglas en inglés), han tomado un papel relevante, pues fueron concebidos precisamente como herramientas que apoyaran el proceso de aprendizaje de manera remota, los LMS presentan diversos problemas como por ejemplo la compatibilidad entre ellos, en un trabajo previo (Cueva-Lovelle et al., 2010) comentan los diversos problemas de compatibilidad que existen entre los diversos LMS, allí se plantea una solución generando un meta-modelo en el cual converjan los módulos comunes de los LMS, ahora si bien este tipo de trabajo solucionan la portabilidad de cursos de un LMS a otro, existe un gran problema en la actualidad y es la capacidad que tienen esos LMS de soportar usuarios persistentes, esto debido a la gran afluencia de usuario que genero la pandemia.

Es allí en donde aparece el concepto de Nube, AWS (Amazon Web Services) define la nube como la distribución de recursos de TI bajo demanda a través de Internet mediante un esquema de pago por uso. En vez de comprar, poseer y mantener servidores y centros de datos físicos, puede obtener acceso a servicios tecnológi-

cos, como capacidad informática, almacenamiento y bases de datos, en función de sus necesidades a través de un proveedor de la nube. (AWS, 2021a). Como vemos este modelo permite que cualquier persona u organización sin mayores esfuerzos o costos pueda tener una infraestructura computacional robusta y en alta disponibilidad, basados en esta premisa el presente artículo muestra como se puede implementar un LMS en una infraestructura de nube pública con alta disponibilidad, garantizando que los usuarios siempre tengan acceso a los recursos de estudio.

## **Modelo de alta disponibilidad para un LMS en una Nube pública**

Para el caso de estudio y la implantación del modelo tomaremos la Universidad Distrital Francisco José de Caldas, esta es una universidad pública del Distrito Capital, ubicada en Bogotá - Colombia, posee 5 Facultades distribuidas por todo el territorio capitalino, con más de 46 grados (pregrados) y 42 postgrados, 121.000 metros cuadrado de infraestructura física distribuidos en 23 inmuebles, 30.000 estudiantes activos y 90.000 egresados aproximadamente (Universidad Distrital, 2021), la Universidad Distrital actualmente tiene más de 50 sistemas de información alojados en a nube pública de AWS que atienden a toda la comunidad administrativa, académica y externa, eso quiere decir que estos sistemas están disponibles para más de 150.000 usuarios, debido a que muchos de estos son sistemas misionales, se requiere que su disponibilidad sea de 24/7/365,

en otras palabras que todo el año deben estar funcionando (AWS, 2021b), ahora si bien estos sistemas misionales estaban alojados en la Nube, el LMS que utiliza la universidad es Moodle y hasta antes de la pandemia se tenían 5 instancias de Moodle montadas en el data center de la institución, con la siguiente arquitectura:

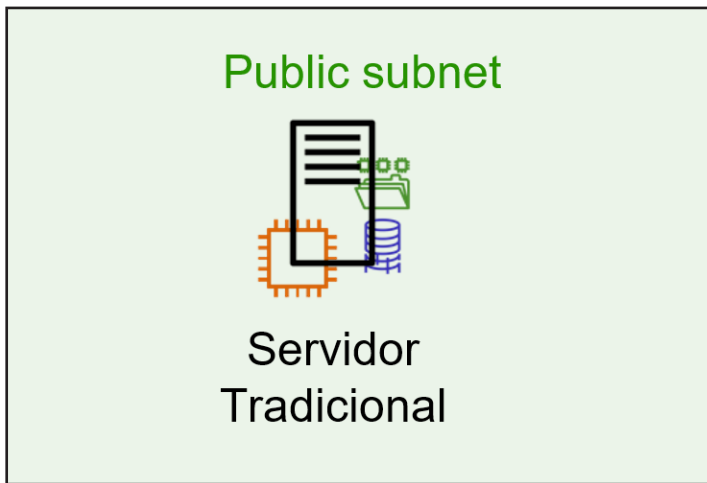


Figura 1: Arquitectura tradicional de Moodle

Básicamente se tenía una instalación tradicional de Moodle, en donde en la misma máquina corría el servidor web, la base de datos y se alojaban todos los archivos, una solución que cubría la baja demanda considerando que todos los programas eran presenciales, con el inicio de la pandemia esta arquitectura empezó a tener problemas de disponibilidad del servicio debido al alto número de usuarios que la empezó a utilizar, como primera medida a corto plazo, se migro tal cual la arquitectura de cada facultad a la nube de pública con la diferencia que se aumentaron los recursos computacionales de manera horizontal, en otras palabras, se utilizó un servidor con mayores recursos computacionales.

El crecimiento acelerado en el uso de los recursos por parte de toda la comunidad académica, trajo consigo otros inconvenientes, como la mantenibilidad de los recursos instalados, el esfuerzo que suponía mantener varias infraestructuras que no permitían la comunicación unas con otras, las diversas versiones de los componentes, el incremento en la facturación por el uso de grandes recursos computacionales entre otras, debido a ello se tomó la decisión en la Universidad de unificar todos los LMSs en este caso Moodle en único sistema de gestión del aprendizaje, que proporcionara un único punto de acceso para los usuarios de la institución, en este sentido basados en la experiencia que la Universidad había adquirido en el uso de tecnologías de nube pública se propuso el siguiente modelo para tener un único LMS en alta disponibilidad:

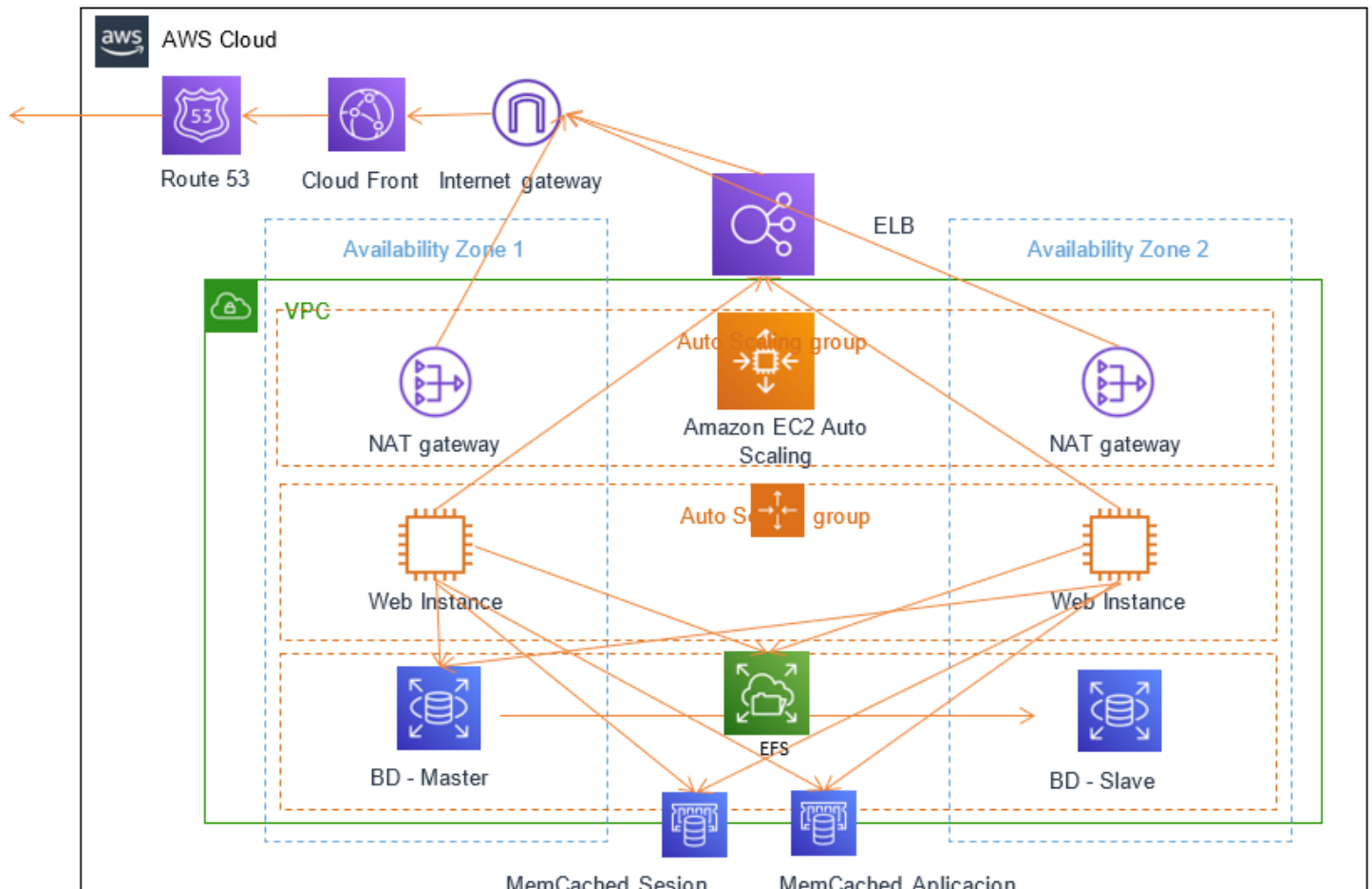


Figura 2: Arquitectura de Moodle en alta disponibilidad sobre una nube pública.

La arquitectura se compone de:

**Route 53:** permite gestionar el dominio a través del cual la aplicación responde.

**CloutFront:** permite la entrega de contenido dinámico y estático, añade una capa de protección contra ataques de denegación de servicios.

**Internet Gateway:** permite que exista comunicación entre las subredes privadas e internet.

**NAT Gateway:** da acceso a internet a las instancias de EC2 en subredes privadas.

**ELB:** Es un Balanceador de Carga de Aplicación se usa para distribuir el tráfico a través del Auto Scaling Group, de las instancias Ec2 que están en múltiples zonas de disponibilidad.

**ElastiCache:** Se utilizan para cachear las sesiones y las aplicaciones mas utilizadas, una para cada una, la tecnología utilizada es MemCached.

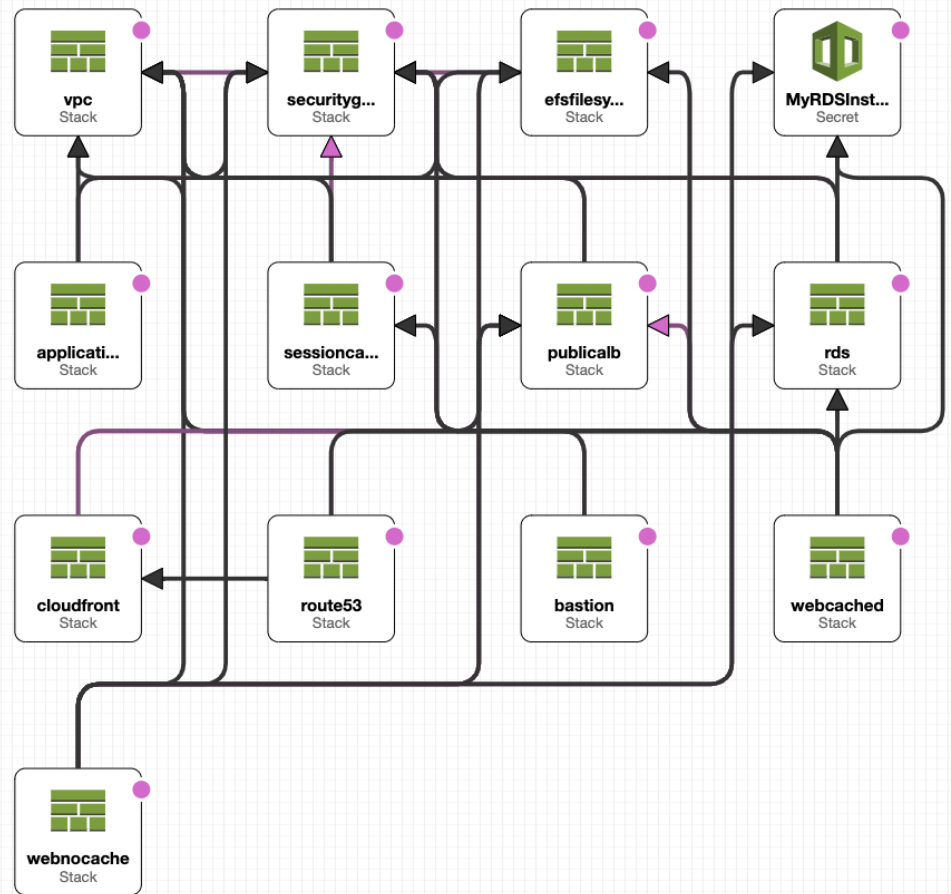
**EFS:** Es un sistema de archivos de red simple, de alta disponibilidad y escalable. Permite acceso a las instancias de Moodle a un almacenamiento compartido de crecimiento dinámico.

**Auto Scaling Group:** permite el escalado y desescalado horizontal de instancias web que atiendan a los usuarios, la elasticidad es una de las grandes características de este modelo.

**RDS:** La base de datos es única y se despliega en un Cluster de Amazon RDS Aurora MySQL, lo que permite tener los datos salvaguardados y en alta disponibilidad para su uso, así mismo permite el escaldo rápido a la demanda elástica de usuarios.

Esta arquitectura se ha desplegado empleando CloudFomation, que se encargará del aprovisionamiento y la configuración de dichos recursos de manera automática, evitando la creación y configuración individual de cada recurso, ahorrando tiempo, recursos y los más importante eliminando errores en un despliegue manual, en este mismo sentido la automatización con CloudFomation permite tener varios ambientes para realizar desarrollo y

pruebas que son fiel copia del de producción, de tal forma que cualquier cambio que se haga será probado de manera rigurosa antes de ponerlo en producción, a continuación, se muestra un diagrama que muestra la implementación de esta arquitectura en un ambiente de producción.



**Figura 3: Despliegue en producción de una Arquitectura de Moodle en alta disponibilidad sobre una nube pública automatizada utilizando CloudFormation.**

## Conclusiones

El uso adecuado de una nube pública, ahorra costos significativamente para suplir las necesidades a nivel de plataformas de apoyo al aprendizaje como los LMSs, en cualquier institución, con el modelo presentado y herramientas como CloudFormation, fácilmente y en cuestión de minutos cualquier institución puede tener su propio sistema de gestión del aprendizaje en alta disponibilidad.

El crecimiento vertical es una buena solución a corto plazo, pero los costos crecen rápidamente y la infraestructura puede limitarse también, una buena arquitectura debe permitir crecer horizontalmente, que los costos sean mas bajos y atender mayor cantidad de usuarios.

Definitivamente tener todos los servicios en una sola máquina, es un riesgo demasiado alto, ya que al tener cualquier daño en la máquina todo el sistema se vería afectado en su disponibilidad, una buena manera de prevenir este alto riesgo, es dividir la arquitectura de tal forma que no solo exista un punto de fallo si no que se distribuya, aumentando el porcentaje de disponibilidad del servicio, además hacer que todos los servicios siempre tengan por lo menos un respaldo, este respaldo entraría en funcionamiento cuando el principal componente falle.