Entregable Final - Cloud Computing en AWS

Proyecto de migración a la nube: PagosOnline S.A.

Santiago Gegenschatz

1- Introducción

Para realizar la migración a la nube vamos a utilizar un *framework* de cinco pasos para ordenar las ideas y los pasos a seguir. En particular, vamos a describir las estrategias utilizadas en la migración en cinco secciones principales a lo largo de este trabajo. A saber:

- 2- Oportunidad
- 3- Análisis
- 4- Planificación de tiempos de implementación
- 5 - Migración y resultados
- 6- Optimizaciones Futuras y Mejora Continua

Adicionalmente, el trabajo cuenta con una séptima sección de mejores prácticas, una octava sección de conclusiones y una novena sección de bibliografía.

2- Oportunidad

2.1 - Descripción de la compañía

Pagosonline S.A. es una empresa dedicada a ofrecer servicios financieros de forma digital. Su principal línea de productos consiste en una pasarela de pagos para que distintos comercios puedan cobrar a sus usuarios sin tener que desarrollar su propia infraestructura. Adicionalmente, los usuarios pueden tener sus propias cuentas en la aplicación, guardando sus fondos y transfiriendo a distintas cuentas de otros contactos. La aplicación es accesible mediante web y mobile. Ambos modos de acceso consultan a un backend que maneja la autenticación, la seguridad y las transacciones entre cuentas y comercios.

Actualmente, toda su infraestructura digital está montada *on premise* puesto que en el momento en el que se fundó la compañía hace más de una década, los servicios de cloud computing eran relativamente poco utilizados.

2.2 - Contexto

La compañía, nacida en el año 2006 desarrolló sus operaciones inicialmente en Argentina, pero en los últimos años comenzó a expandirse a países de la región. Esta expansión, sumada a la creciente popularidad del producto de pagos ha llevado a la empresa a experimentar fallos en sus sistemas en momentos críticos, como por ejemplo cuando los comercios electrónicos realizan muchas peticiones a la API de la pasarela de pagos en fechas como CyberWeek o Black Friday.

La empresa está en búsqueda de una solución para poder escalar la potencia de sus equipos durante esas fechas críticas, sin tener que invertir grandes cantidades de capital en infraestructura que quedará ociosa durante la mayor parte del año. Es por esto que la empresa se encuentra interesada en realizar un proceso de migración a la nube.

También, dada la naturaleza sensible de los datos manejados por las empresa, contar con respaldos adecuados (persistentes, seguros y redundantes) resulta difícil ya que implica la construcción de un datacenter propio con distintos servicios especializados de alto coste (seguridad, red eléctrica especial, etc).

Entonces, queda claro que la migración a la nube ofrece claros beneficios en términos de reducción de costos, aumento de la escalabilidad y mejoras en la seguridad.

3- Análisis

3.1 - Descripción de la solución

La solución propuesta cuenta con varios servicios de AWS para replicar la arquitectura que la empresa posee en on premise en la nube. Entre ellos:

- **EC2**

  El backend de la aplicación principal de PagosOnline S.A. está escrito fundamentalmente en Node.js. Actualmente, el backend corre en servidores propios de la empresa, que no son fácilmente escalables y no están protegidos contra eventos adversos como fallas en la red eléctrica.

  Para replicar esta estructura de backend en la nube, crearemos una serie de instancias EC2 capaces de ejecutar los módulos escritos en Node.js. Dada la naturaleza genérica del backend, utilizaremos la **familia M** de instancias que ofrece EC2. En particular, utilizaremos las instancias

m5.xlarge, que ofrecen potencia de cómputo similar a la que la empresa posee on premise, con la ventaja de que pueden ser fácilmente escaladas durante picos de demanda como Black Friday.

- **RDS**

  Para realizar las transacciones es necesario contar con sistemas de bases de datos que soporten transacciones. Para eso se migrarán las bases de datos SQL que la empresa utiliza actualmente a bases de datos, también de tipo SQL, pero alojadas en la nube mediante el sistema RDS de AWS.

- **DynamoDB**

  Los datos de los usuarios, (identificadores, red de contactos, etc) tienen una naturaleza que favorece su guardado en bases de datos noSQL. Actualmente, la empresa utiliza bases de datos en MongoDB para guardar estos datos. Se procederá a trasladar estas bases de datos a DynamoDB, el servicio de bases de datos noSQL en la nube de AWS.

- **S3**

  Actualmente, PagosOnline.com tiene varios servidores dedicados a guardar información de todo tipo (un *data lake* propio) dentro de servidores on premise. La gestión de este data lake sin herramientas especializadas se vuelve tediosa y costosa. Adicionalmente, este hecho *dificulta el proceso de ETL*. Es por esto que se optará por migrar el data lake de la empresa a S3, donde los accesos podrán ser gestionados con mayor seguridad. Por último, la migración a S3 también permitirá configurar herramientas útiles para la construcción de reportes de business intelligence, como por ejemplo AWS Athena.

- **IAM**

  La empresa cuenta con una creciente cantidad de empleados para atender sus servicios. El manejo de accesos a la información se está volviendo exponencialmente complejo con los servidores on premise. Se utilizará la herramienta IAM de AWS para solucionar este problema mediante la creación de una serie de políticas que ayudará a que solamente ciertos empleados tengan acceso a ciertos datos.

- **VPC**

  Para obtener un mayor nivel de seguridad de los diferentes datos, se implementará una Virtual Private Cloud con AWS. Esta tendrá diferentes subnets públicas y privadas a través de distintas zonas de disponibilidad, lo que garantizará la alta disponibilidad y la privacidad de aquellos servicios que se encuentran en la red privada sin acceso a internet. Adicionalmente, contará con

un balanceador de carga posicionado en frente del grupo de autoescalamiento que contendrá las instancias EC2.

- **CloudWatch, CloudTrail & CloudFormation**
  Para que los ejecutivos *c-level* de PagosOnline S.A. tengan una visión general de los costos asociados con la infraestructura en la nube, se crearán distintos dashboards mediante cloudwatch para mostrar el tiempo real el costo y la cantidad de actividad (usuarios, los, etc) que experimenta la aplicación.

  Por otro lado, mediante CloudTrail se generarán logs de acceso a la API de AWS, que asegurará que los *technical leaders* de PagosOnline S.A. podrán tener un historial de los cambios realizados en la plataforma por los desarrolladores.

  Por último, se utilizará el servicio de CloudFormation para generar de forma automatizada y sistemática los servicios que se utilizarán en cada una de las cuentas de AWS que se crearán para la empresa (Desarrollo, Testing y Producción). Está estrategia se encuentra en línea con las mejores prácticas sugeridas por AWS, que considera que la automatización una parte fundamental de la gestión de los procesos en la nube.

3.2 - Parámetros de la cuenta de AWS

La mayoría de los servicios de PagosOnline S.A son prestados a clientes basados en Latinoamérica. En general, la latencia es un factor determinante para los usuarios de este tipo de servicios financieros, en los que los usuarios requieren realizar pagos de forma rápida. Dado lo anterior, se optará por configurar la cuenta de AWS en la región de Sao Paulo. Si bien es verdad que esto implica un mayor costo comparado con usar la región de Virginia, es necesario destacar que el costo de ofrecer un servicio lento a los usuarios es aún más grande, pues implica clientes insatisfechos y una futura merma en el market share de la empresa.

Además, con el objetivo de maximizar la seguridad, se creará una cuenta de AWS por cada uno de los tres ambientes que la empresa necesita, es decir, una cuenta para Development, una para Testing y una para Producción. Esto garantizará una capa adicional de seguridad para la empresa, que en caso de ser atacada en una de las cuentas tendrá las demás indemnes.

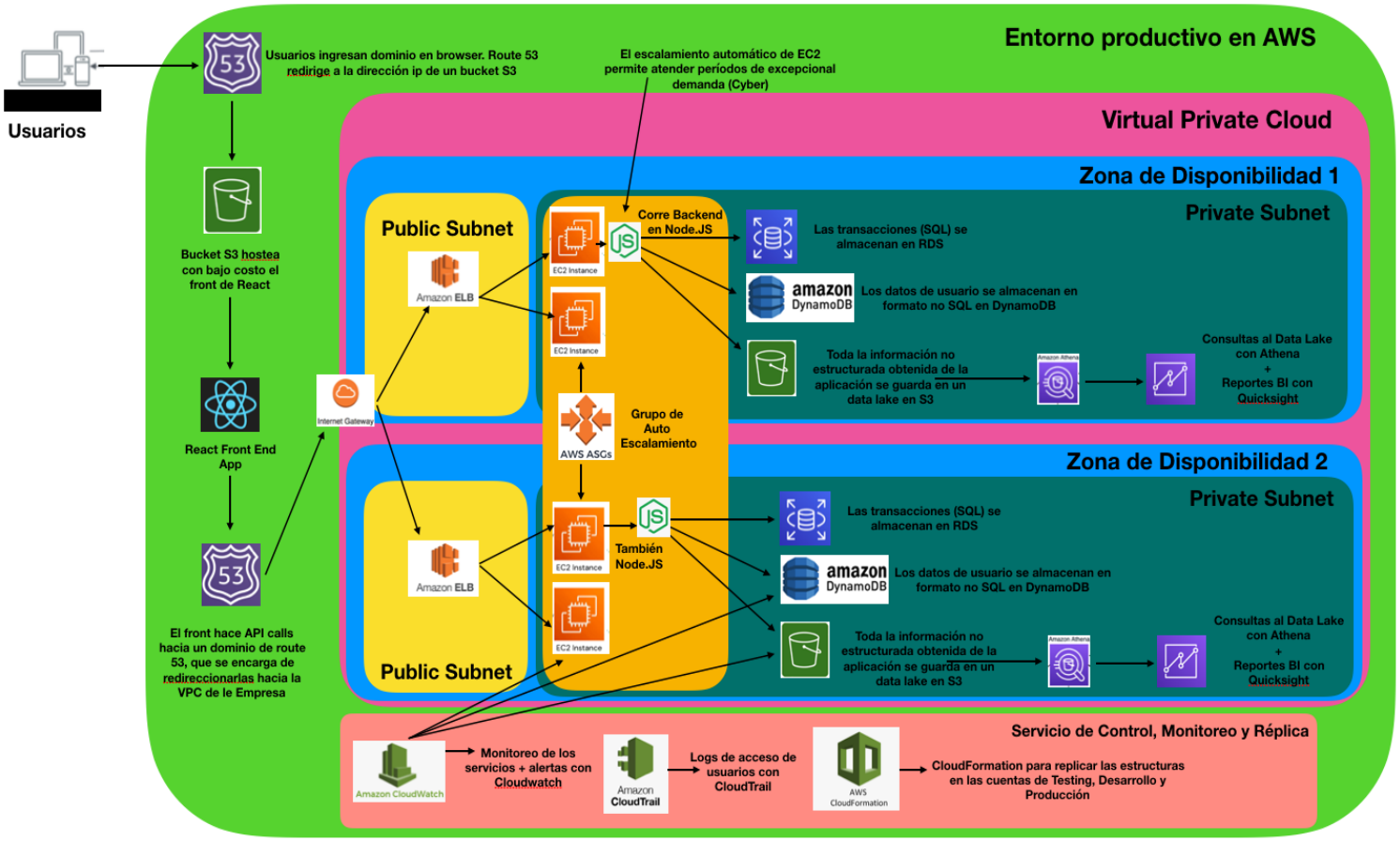
3.3 - Estrategias de Migración

La principal estrategia utilizada durante el proceso de migración a la nube será el *REHOSTING*, puesto que la mayoría de las aplicaciones de PagosOnline S.A están empaquetadas y pueden correr sin problemas en las instancias de EC2 provistos de los entornos adecuados (Node.js por ejemplo).

3.4 - Equipo de COE

Las migraciones son procesos complejos que requieren de mucha coordinación entre diferentes personas y equipos para resultar exitosas. El equipo de COE se encargará de facilitar la migración y estará compuesto por los **directores de infraestructura, seguridad y soporte técnico** de PagosOnline S.A.

3.5 - Visualización de la arquitectura de la solución



3.6 Estrategias de ahorro en la nube

Al ser una empresa con fines de lucro, es de **fundamental** importancia ahorrar costos para el cliente. Para esto se implementarán las siguientes políticas:

- Los ambientes de desarrollo y testing van a tener menor poder para ahorrar costos. Por ejemplo en vez de usar EC2 **m5.2xlarge** se utilizarán las **t3.medium** que son más baratas.

- Los ambientes de desarrollo y testing **van a estar apagados durante los fines de semana y entre las 8 pm y las 8 am**, puesto que en esas horas los empleados duermen o descansan. Esto representa un ahorro del 50% aproximadamente en los entornos de Desarrollo y Testing solamente por organizar mejor los horarios de encendido y apagado.

- El ambiente de producción solo se va a prender cuando sea necesario hacia el final del proyecto, una vez que todos los otros procesos de la migración hayan sido ejecutados satisfactoriamente.

3.7 - Calculadora de costos

Para estimar los costos de la migración a la nube se tendrán en cuenta los siguientes datos brindados por PagosOnline S.A. Actualmente la empresa cuenta con:

- 30 servidores on premise para correr los servicios de backend. Sin embargo, solamente están en uso unos 15. Para evitar costos asociados a contar con recursos ociosos, se procederá a configurar 15 instancias EC2 **m5.xlarge.** Cada una tendrá 10GB de almacenamiento EBS para guardar los recursos necesarios para correr el backend.

- La capacidad total de almacenamiento de los servidores on premise es de 250TB, pero solamente se están utilizando unos 80TB. Entonces, se configurará un almacenamiento total de

100TB en el servicio S3 de AWS. De esta forma, se logrará contar con espacio suficiente para el data lake de la empresa sin tener una alta capacidad ociosa.

- La plataforma de pagos cuenta con 10 millones de usuarios. Cada uno de ellos hace unas 5 transacciones por día, y en cada transacción se almacenan en una base de datos SQL unos 100 bytes de información. Esto implica que se generan unos 5GB de transacciones nuevos todos los días. Adicionalmente, por motivos regulatorios las transacciones de los últimos tres años tienen que estar guardadas para que los usuarios puedan acceder a ellas. En consecuencia, se crearán 15 potentes bases de datos RDS **m5.large,** cada una de ellas conectada a una instancia EC2 corriendo el backend. Cada una contará con 600GB de almacenamiento, lo que permitirá que se almacenen en total los registros correspondientes a cuatro años en el sistema (los tres anteriores más el actual).

- Los datos de los usuarios que no son transacciones son guardados en formato noSQL. Se estima que la información correspondiente a cada usuario ocupa unos 5 KB de almacenamiento. Entonces, se procederá a crear una base de datos en DynamoDB con 60GB de almacenamiento, asegurando que haya espacio para el crecimiento de la aplicación.

- La VPC se configurará con un balanceador de carga capaz de manejar 700 llamados por segundo, resultante de promediar a lo largo del día las 50 millones de transacciones realizadas, sumado a un pequeño margen de seguridad para garantizar buenos tiempos de respuesta.

- AWS Athena será configurado para que se puedan realizar hasta 45.000 *queries* por semana contra el data lake de la empresa.

- Los ambientes de desarrollo y testing, como se explicó más arriba, contarán con la misma estructura (generada a través de templates de cloudformation) pero con menor potencia y tiempos de encendido, para ahorrar costos.

A continuación el link a la calculadora de costos de AWS, con todos los ítems descriptos arriba detallados:

https://calculator.aws/#/estimate?id=362c0b3caef57771d518ae97996ba46a43924a5b

Como se puede observar, el costo total anual de la solución planteada es de aproximadamente 282.000 USD anuales, lo que equivale a 2 centavos de dólar por usuario por mes. Si bien la solución implica una considerable suma de dinero, al compararla con los costos on premise resulta clara la ventaja de los

servicios en la nube. Actualmente, la empresa cuenta con 14 empleados, cada uno de ellos con un sueldo promedio de 50.000 USD anuales, necesarios para el mantenimiento de los sistemas on premise. Se estima que con una migración a cloud solamente serán necesarios 5 empleados para manejar la misma cantidad de sistemas (puesto que AWS se encarga de una gran parte de la carga de trabajo).

Esto significa que la migración a cloud requerirá una inversión de 532.000 USD anuales, comparados con los 700.000 USD invertidos de forma anual en honorarios actualmente. A esta ecuación hay que sumarle los potenciales beneficios de mayor seguridad ante fallas en el sistema eléctrico y las ventajas de no contar con una gran cantidad de equipamiento electrónico que se deprecia año tras año. Por último, la migración a cloud también permitirá que la empresa escale rápidamente en fechas de alta demanda, como Black Friday, sin incurrir en grandes inversiones de capital.

4- Planificación de tiempos de implementación

4.1 - Descripción del cronograma de implementación

Obviamente, la creación de todos estos servicios en la nube lleva una cantidad no trivial de tiempo. Tanto la migración a las diferentes bases de datos para consumo instantáneo (Dynamo y RDS) como la creación de un nuevo nuevo data lake en S3 requieren varios semanas de trabajo. Adicionalmente, es necesario calcular una cierta cantidad de tiempo para testing una vez que todos los sistemas sen encuentren correctamente estructurados. Aquí un cronograma de seis semanas para la migración.

- **Semana 1:**
    - Creación de cuenta de AWS, usuario Root, usuarios IAM y políticas de permisos.
    - Creación de la VPC con subredes públicas y privadas en dos zonas de disponibilidad.
    - Configuración de logs de acceso a través de CloudTrail.
    - Creación de instancia EC2 y de imagen de sistema para correr el backend.
    - Prueba de funcionamiento del backend en la nueva instancia EC2.

- **Semana 2:**
    - Creación de bases de datos en RDS para modelar las transacciones.
    - Creación de bases de datos en DynamoDB para modelar los datos personales de los usuarios.
    - Testeo de funcionalidad del backend corriendo en EC2 mediante la ejecución de llamadas a las bases de datos en RDS y DynamoDB.
    - Creación de alertas en CloudWatch y de templates en CloudFormation.

- **Semana 3:**
    - Creación de distintos *buckets* de S3 para almacenar la información de tipo permanente.
    - Testeo de conexión entre los backends corriendo en EC2 y los datos almacenados en S3.
    - Testeo general de accesos al sistema y del backend.

- **Semana 4:**
    - Migración parcial de datos a RDS, DynamoDB y S3.
    - Testing intensivo y automatizado de todas las funciones del backend tras la migración final.
    - Creación de servicio de Athena para ejecutar *queries* contra el data lake en S3.

- **Semana 5:**
    - Migración total de los datos tras el testeo de la semana anterior.
    - Tests automatizados en el entorno de testing.

- **Semana 6:**
    - Levantamiento de los nuevos sistemas en producción.
    - Revisión de los objetivos alcanzados y seteo de nuevas métricas.

4.2 - Tabla Gráfica del Cronograma de Implementación

	Mes	1				2	
Actividad	Semana	1	2	3	4	5	6
Creación de cuenta AWS root + Usuarios IAM							
Creación de políticas y permisos para desarrolladores y empleados administrativos							
Creación de instancia EC2 con imagen Linux							
Creación de VPC con subredes públicas y privadas + ELB							
Configuración de logs de acceso con CloudTrail							
Subida preliminar y testing del backend Node.JS empaquetado en EC2							

Creación de bases de Datos RDS para modelar transacciones con SQL						
Creación de Bases de Datos DynamoDB para modelar datos personales						
Testeo de Conexión del Backend con las bases						
Creación de alertas en CloudWatch y Templates en CloudFormation						
Creación de Buckets S3						
Testeo de conexión Backend con Buckets S3						
Testeo general del funcionamiento del sistema						
Configuración de Athena para reportes de BI						
Migración de datos desde on premise a Dynamo y RDS tras testeo exitoso						
Testeo general tras la migración						
Levantamiento en producción						
Revisión de objetivos alcanzados y definición de nuevas metas						

5- Migración y resultados

5.1 Resultados Estratégicos

Con la implementación de un sistema como el propuesto, se lograrían mejores resultados en términos de latencia durante períodos de alta intensidad en la API. Adicionalmente, habría mejor control de accesos a los datos gracias al sistema IAM de AWS. En total, se estima una reducción de costos de alrededor del 25%, sumado a las mejoras intangibles en capacidad de escalamiento.

6- Optimizaciones Futuras

En el futuro, es posible realizar distintas optimizaciones a la arquitectura. Entre ellas:

- Aprovechar varios servicios de machine learning ofrecidos por AWS, como por ejemplo Sagemaker, para entrenar modelos que detecten y prevengan transacciones fraudulentas.

- Utilizar herramientas como AWS RDS Performance insights para monitorear la carga sobre las bases RDS y optimizar el diseño de las *queries* contra la base de datos.

- Hacer uso de servicios como AWS Elasticaché para guardar consultas frecuentes en caché, ofreciendo menor latencia a los usuarios y disminuyendo el impacto sobre las bases de datos RDS y DynamoDB.

7- Mejores Prácticas Recomendadas

- Guardar cuidadosamente el usuario y clave *root* del nuevo sistema. Este usuario posee permisos de administrador superiores a los de cualquier usuario IAM. De esta forma, un compromiso de la cuenta *root* representaría un gran desafío para la integridad y seguridad de los sistemas informáticos de la empresa.

- Delegar un equipo de personas dentro de la empresa para trabajar, modificar y mantener el sistema de permisos IAM. Controlar la seguridad de este tipo de datos es fundamental, y AWS ofrece un sistema muy poderoso para manejarlos. Es indispensable, no obstante, que quienes manejen este sistema estén capacitados en su uso.

- Etiquetar todos los servicios creados y poner en uso políticas que fuercen el uso de etiquetas con la creación de cada servicio. De esta manera, en el futuro será posible filtrar los diferentes centros de costos por etiquetas para tomar mejores decisiones financieras para la empresa.

- Eliminar las AMI antiguas.

- Cambiar las contraseñas de los usuarios IAM cada una cantidad predeterminada de tiempo, para fortalecer la seguridad del sistema.

8- Conclusiones

Actualmente, la empresa enfrenta grandes obstáculos en el escalamiento de sus sistemas informáticos. La migración a la nube de los sistemas ofrece en este caso una interesante alternativa para resolver este desafío.

En primer lugar, la migración permitirá el escalamiento de los servidores con mayor facilidad durante fechas críticas como Cyber o Black Friday. En segundo lugar, abaratará los costos de infraestructura en un 25% o más, ya que será necesario menos personal para mantener la misma cantidad de sistemas. Adicionalmente, AWS es quien se ocupa del mantenimiento de los datacenters, la redundancia y la seguridad del hardware. En tercer lugar, la migración resolverá problemas de accesos y seguridad a los datos mediante el poderoso sistema de políticas de accesos y permisos ofrecido por AWS IAM. En cuarto lugar, mediante S3 será posible construir un data lake propio del que la empresa podría obtener valuables *insights* mediante servicios como Athena y Quicksight.

En definitiva, la migración representará una reducción de costos y una ventaja competitiva para la empresa, que podrá escalar su infraestructura en función de la demanda. Este hecho permitirá a la empresa ofrecer un mejor servicio a sus clientes durante momentos de alto volumen, generando clientes más satisfechos y una probable expansión del market share.

9- Bibliografía Recomendada

- AMAZON (2023). Six advantages of cloud Computing. Recuperado de: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html

- AMAZON (2023). Cost optimization Pillar - AWS Well-Architected Framework. Recuperado de: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html