



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**PROCESO DE AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE  
APLICACIONES Y ENTORNO DE APLICACIONES UTILIZANDO CONCEPTOS Y  
HERRAMIENTAS DE COMPUTACIÓN EN NUBE (*CLOUD COMPUTING*)**

**José Luis García Orrego**

Asesorado por el Ing. Sergio René Orrego Soto

Guatemala, noviembre de 2012



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROCESO DE AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE  
APLICACIONES Y ENTORNO DE APLICACIONES UTILIZANDO CONCEPTOS Y  
HERRAMIENTAS DE COMPUTACIÓN EN NUBE (*CLOUD COMPUTING*)**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JOSÉ LUIS GARCÍA ORREGO**  
ASESORADO POR EL ING. SERGIO RENÉ ORREGO SOTO

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, NOVIEMBRE DE 2012



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Herbert René Miranda Barrios
EXAMINADOR	Ing. Carlos Alfredo Azurdia Morales
EXAMINADOR	Ing. Everest Darwin Medinilla Rodríguez
EXAMINADOR	Ing. Jorge Luis Álvarez Mejía
SECRETARIA	Inga. Gilda Marina Castellanos de Illescas



## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **PROCESO DE AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE APLICACIONES Y ENTORNO DE APLICACIONES UTILIZANDO CONCEPTOS Y HERRAMIENTAS DE COMPUTACIÓN EN NUBE (*CLOUD COMPUTING*)**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 03 de agosto de 2011.

**José Luis García Orrego**





## **ACTO QUE DEDICO A:**

- Dios** Por permitirme alcanzar mis metas.
- Mis padres y abuelos** Oscar Ernesto García Aguirre (q.e.p.d.), Sara Margarita Orrego Soto, Ismael Orrego Pineda (q.e.p.d.), Angelina Soto Ovalle (q.e.p.d.). Ya que con este acto de graduación me permito honrarlos en vida por todo su sacrificio y esfuerzo que han hecho por mi durante toda mi vida.
- Mi esposa y mi hija** María Isabel Pérez Valencia, María Alejandra García Pérez, el regalo más bello que Dios me ha dado.
- Mi bebé** Mariana, que está por nacer.
- Mis hermanos** Julio Ernesto y Mónica García, por su apoyo.



## **AGRADECIMIENTOS A:**

- Dios** Por darme la vida y sabiduría para poder alcanzar esta meta.
- Mis padres y abuelos** Oscar Ernesto García Aguirre (q.e.p.d.), Sara Margarita Orrego Soto, Ismael Orrego Pineda (q.e.p.d.), Angelina Soto Ovalle (q.e.p.d.), por su amor, esfuerzo, entrega y dedicación incondicional en mi vida.
- Mi esposa** María Isabel Pérez Valencia, por estar siempre a mi lado, su apoyo, amor y amistad.
- Mis hermanos** Julio Ernesto y Mónica García, quienes me motivaron a alcanzar el éxito, gracias por su apoyo.
- Mis amigos** Por su amistad y colaboración.



## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
GLOSARIO .....	VII
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN .....	XVII
1. MARCO TEÓRICO PARA EL PROCESO DE AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE APLICACIONES Y SU ENTORNO EN LA NUBE .....	1
1.1. Conceptos de computación en nube.....	1
1.2. Características de la nube .....	2
1.3. Características complementarias .....	6
1.4. Ventajas de negocio al utilizar la computación en nube.....	9
1.5. Ventajas técnicas de computación en nube.....	11
1.6. Conceptos de computación <i>grid</i> o computación distribuida .....	12
1.7. Topología de redes .....	14
1.8. Tipos de nube .....	18
1.8.1. Nubes públicas .....	18
1.8.2. Nubes privadas.....	19
1.8.3. Nubes híbridas.....	19
1.9. Capas.....	20
1.9.1. <i>Software</i> como servicio ( <i>Software as a Service</i> , SaaS).....	20
1.9.2. Infraestructura como servicio ( <i>Infrastructure as a</i> <i>Service</i> , <i>IaaS</i> ) .....	21

1.9.3.	Plataforma como servicio (Platform as a Service, PaaS).....	21
1.10.	Aplicaciones y entorno de aplicaciones en la nube.....	21
1.11.	Virtualización.....	22
1.11.1.	Qué es la virtualización.....	22
1.11.2.	Cómo funciona la virtualización .....	23
1.11.3.	Ventajas y desventajas de la virtualización .....	25
1.12.	Máquinas virtuales .....	28
1.12.1.	Qué es una máquina virtual .....	28
1.12.2.	Tipos de máquinas virtuales .....	29
1.12.3.	Ventajas y desventajas de las máquinas virtuales....	30
1.13.	Infraestructura virtual .....	31
1.13.1.	Que es la infraestructura virtual .....	32
1.13.2.	Componentes .....	32
1.13.3.	Ventajas y desventajas.....	33
2.	FACTORES IMPORTANTES PREVIO A ESTUDIAR EL PROCESO DE AUTOMATIZACIÓN DE APLICACIONES Y ENTORNOS EN LA NUBE .....	37
2.1.	Prioridades de aplicaciones y negocios .....	38
2.2.	Presupuesto.....	43
2.3.	Infraestructura y redes .....	47
2.4.	Consideraciones de seguridad.....	48
3.	PROCESO DE ADMINISTRACIÓN Y AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE APLICACIONES Y ENTORNO DE APLICACIONES EN LA NUBE.....	57
3.1.	Virtualización para la administración y automatización de aplicaciones y sus entornos en la nube .....	57

3.2.	Servicios estándar en la virtualización para eliminar la constante reconfiguración estática de infraestructura para un entorno informático dinámico .....	58
3.3.	Algunas herramientas para la administración de máquinas virtuales e infraestructura virtual.....	63
3.4.	Monitoreo de rendimiento de máquinas virtuales.....	76
3.5.	Herramientas que permitan lograr a la empresa obtener máximos rendimientos, escalabilidad y disponibilidad de su infraestructura virtual de forma automática .....	82
3.6.	Procesos y herramientas de <i>software</i> utilizadas en la automatización para la construcción de aplicaciones y entornos de aplicaciones .....	97
3.7.	Tareas para la administración de aplicaciones y entorno de aplicaciones al inicio y durante todo el ciclo de vida de las aplicaciones .....	100
3.8.	Ventajas y desventajas existentes al momento de utilizar las diferentes herramientas para la administración, construcción y despliegue de aplicaciones y sus entornos en la nube de manera automática.....	105
3.9.	Resumen de los pasos a seguir y herramientas a utilizar adecuadamente para el proceso de automatización en la construcción de aplicaciones y sus entornos en la nube .....	107
	CONCLUSIONES .....	125
	RECOMENDACIONES .....	127
	BIBLIOGRAFÍA.....	129
	APÉNDICE.....	131





# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Topología de redes estáticas .....	16
2.	Presupuesto mensual de instancias en <i>Amazon EC2</i> .....	46
3.	Cuenta con grupos de usuarios de <i>Amazon</i> .....	52
4.	Muestra los grupos y los usuarios en consola de <i>Amazon IAM</i> .....	52
5.	Muestra los usuarios creados en la consola de <i>Amazon IAM</i> .....	53
6.	Grupo de seguridad de <i>Amazon EC2</i> para aplicaciones <i>web</i> .....	54
7.	Múltiples instancias de una AMI.....	59
8.	Inicio de diferentes tipos de instancia .....	60
9.	Volúmenes de instancia .....	61
10.	Instantánea de un volumen .....	61
11.	Adjuntar un volumen nuevo a otra instancia .....	62
12.	Separar un volumen de una instancia.....	62
13.	Consola de AWS desplegando AMIs con el servicio de EC2.....	64
14.	Consola AWS desplegando instancias de AMIs con el servicio de EC2 .....	65
15.	Gráficas del CPU, el disco y la red para la instancia seleccionada.....	67
16.	Gráfica de utilización promedio del CPU de la instancia seleccionada .....	67
17.	Volumen de 10 GiB asociado a instancia actualmente en uso.....	69
18.	<i>Snapshots</i> de volúmenes de AMIs que tienen configurados sistemas operativos, aplicaciones, bases de datos, lenguajes de programacion, etc .....	70
19.	Ejemplo de almacenamiento en <i>SIMPLE STORAGE SERVICE (S3)</i> .	71
20.	Métricas de utilización de CPU .....	77

21.	<i>Amazon CloudWatch</i> .....	81
22.	Elasticidad y escalabilidad automatizada .....	84
23.	<i>Trigger</i> que dispara el servicio <i>Auto Scaling</i> .....	89
24.	Tareas o pasos para crear una nube privada .....	109
25.	Inicio de aplicación .....	110
26.	Lista básica .....	111
27.	Selección de una AMI .....	112
28.	Configuraciones adicionales .....	113
29.	Adición de etiquetas .....	114
30.	Ingreso de credenciales de seguridad .....	115
31.	Elección de grupo de seguridad .....	116
32.	Resumen de ingreso de datos .....	117
33.	Estatus de consola .....	118
34.	Conexión remota .....	119
35.	Acceso a la aplicación instalada .....	120
36.	Corriendo aplicación .....	121
37.	Información de cuentas .....	122

## TABLAS

I.	Inversión inicial aproximada para compra de equipo de una empresa ..	43
----	--	----

## GLOSARIO

<b>Amazon EBS</b>	Un tipo de almacenamiento que permite crear volúmenes que se pueden montar como dispositivos de instancias de Amazon EC2. Los volúmenes de Amazon EBS se comportan como dispositivos de bloque externo sin formato.
<b>AMI</b>	<i>Amazon Machine Image</i> , es un tipo especial de sistema operativo preconfigurado y software de aplicación virtual que se utiliza para crear una máquina virtual dentro de la <i>Amazon Elastic Compute Cloud(</i> EC2 <i>)</i> . Además, es la unidad básica de la implementación de servicios prestados utilizando EC2.
<b>API</b>	<i>Application Programming Interface</i> o interfaz de programación de aplicaciones, es una interfaz de programación que representa la capacidad de comunicación entre componentes de <i>software</i> .
<b>Base 64</b>	Es un sistema de numeración posicional que usa 64 como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de ASCII.
<b>DBA</b>	Administrador de base de datos, por sus siglas en inglés <i>Data Base Administrator</i> .

<b>DNS</b>	<i>Domain Name System</i> , sistema de nombres de dominio, es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada.
<b>Elastic Compute Cloud (EC2)</b>	Es un servicio web que proporciona capacidad informática con tamaño modificable en la nube.
<b>Elastic IPAddress</b>	Es una dirección IP pública estática diseñada para la computación en nube dinámica. Las <i>elastic IP address</i> están asociadas a una cuenta de Amazon, no a una instancia específica.
<b>Firewall</b>	Cortafuegos, <i>firewall</i> en inglés, es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i> , protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la <i>World Wide Web</i> .
<b>ICMP</b>	El Protocolo de Mensajes de Control de Internet o ICMP (por sus siglas de <i>Internet Control Message Protocol</i> ), es el sub protocolo de control y notificación de errores del protocolo de internet (IP).

<b>Instancia EC2</b>	Una instancia es una copia de una AMI. La instancia que se crea tiene las mismas características y propiedades de la AMI a la que pertenece. Una vez que la AMI se ha puesto en marcha, el sistema en funcionamiento resultante se conoce como una instancia. Todas las instancias basadas en la misma AMI inician idénticas y cualquier información sobre ellas se pierde cuando las instancias se terminan o fallan.
<b>IPsec</b>	Abreviatura de Internet <i>Protocol security</i> , es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos.
<b>LAN</b>	Red de área local o LAN, del inglés <i>local area network</i> .
<b>Métrica</b>	Es un concepto fundamental y representa un conjunto de puntos de datos ordenados en el tiempo. Puede pensar en una métrica, como una variable que se desea controlar. Los puntos de datos representan los valores de esa variable en el tiempo. Por ejemplo, el uso de la CPU de una instancia en particular es una métrica.
<b>NTFS</b>	Del inglés <i>New Technology File System</i> , es un sistema de archivos de <i>Windows NT</i> incluido en las versiones de <i>Windows 2000</i> , <i>Windows XP</i> , <i>Windows Server 2003</i> y <i>2008</i> , <i>Windows Vista</i> y <i>Windows 7</i> . Permite definir el tamaño del <i>clúster</i> , a partir de 512 bytes (tamaño mínimo

de un sector) de forma independiente al tamaño de la partición. Es un sistema adecuado para las particiones de gran tamaño requeridas en estaciones de trabajo de alto rendimiento y servidores.

**Paquetes de red** Reciben este nombre cada uno de los bloques en que se divide, en el nivel de Red, la información a enviar.

**QA** Aseguramiento de Calidad, por sus siglas en inglés, *Quality Assurance*.

**Refactorizar** Refactorizar el código consiste en cambiar su estructura interna, especialmente el diseño de sus objetos, para hacerlo más comprensible, fácil de mantener y eficaz, sin cambiar su comportamiento visible.

**Snapshot** Son copias de seguridad instantáneas de volúmenes de Amazon EBS y que son almacenados en *Amazon S3*. Usted puede utilizar estos *Snapshots* como punto de partida para nuevos volúmenes de *Amazon EBS* y para proteger sus datos para una mayor durabilidad a largo plazo.

**SOA** Arquitectura orientada a servicios de cliente, en inglés *Service Oriented Architecture*, es un concepto de arquitectura de *software* que define la utilización de servicios para dar soporte a los requisitos del negocio.

<b><i>SugarCRM</i></b>	Aplicación de código abierto que permite llevar la administración de la relación con los clientes.
<b>TI</b>	Tecnología de la Información.
<b>UPD</b>	<i>User Datagram Protocol</i> , es un protocolo del nivel de transporte basado en el intercambio de datagramas.
<b>ZFS</b>	Es un sistema de archivos desarrollado por <i>Sun Microsystems</i> para su sistema operativo Solaris.





## RESUMEN

La computación en nube es un conjunto de tecnologías maduras ya disponibles para ser utilizadas por personas y organizaciones de cualquier tamaño.

Los conceptos y herramientas de computación en nube juegan un papel clave en el proceso de automatización en la construcción de aplicaciones y sus entornos.

La correcta aplicación de estos conceptos y utilización de las herramientas en las diferentes tareas que tienen que desarrollar las organizaciones empresariales de TI, se refleja en la eficiencia para configurar y desplegar aplicaciones y entornos de aplicaciones de forma exitosa.

Actualmente las aplicaciones comerciales tradicionales han sido siempre demasiado complicadas y caras, la cantidad y variedad necesaria de *hardware* y *software* para ejecutarlas son abrumadoras. Se necesita todo un equipo de expertos para que las pueda instalar, configurar, probar, ejecutar, asegurar y actualizar.

Al igual que con las aplicaciones comerciales tradicionales, la construcción, el despliegue y la configuración de infraestructura de aplicaciones y las aplicaciones que se ejecutan en la infraestructura a menudo son tareas fundamentales para las organizaciones empresariales de TI.

Los enfoques tradicionales para la realización de estas tareas pueden resultar problemáticos y provocar ineficiencias y pérdidas tales como tiempos de espera extensos para la configuración de los entornos; imposibilidad de recrear en forma constante entornos cuando sea necesario y la necesidad de que el sector de pruebas y desarrollo tenga que tener demasiado conocimiento sobre infraestructura.

Actualmente la creación, el despliegue, la configuración y luego la recreación de entornos de aplicaciones son tareas críticas para las organizaciones de TI, adicionalmente a esto, la ineficiencia a menudo rodea este conjunto de tareas.

Por tal razón, se explicarán los conceptos y herramientas de computación en nube necesarios para poder administrar y automatizar una solución integrada para afrontar estas ineficiencias, y se aplicarán los conceptos de computación en nube, utilizando la plataforma de servicios de nube de Amazon denominados EC2 (*Elastic Compute Cloud*), y utilizando la aplicación de código abierto llamada *SugarCRM*, montada en dicha plataforma tecnológica.

# OBJETIVOS

## General

Realizar un estudio bibliográfico sobre el enfoque virtualizado y automatizado para afrontar las ineficiencias de los enfoques tradicionales de despliegue y configuración de infraestructura de aplicaciones, y conocer el proceso de automatización para construcción de aplicaciones y sus entornos, mostrando cómo las herramientas de computación en nube pueden combinarse para fortalecer la infraestructura dinámica y las aplicaciones que se ejecutan en dicha infraestructura. Así mismo, investigar las herramientas que proporcionen una forma sencilla de configurar los entornos de las aplicaciones, usando una interfaz gráfica basada en imágenes virtuales, que encapsulan muchos de los conocimientos sobre la infraestructura necesaria para crear de forma automática entornos de aplicaciones, los que permiten el despliegue y la inicialización en cuestión de minutos y no de semanas, de entornos completos.

## Específicos

1. Identificar y documentar los diferentes conceptos, procesos y herramientas utilizadas para la configuración y despliegue de aplicaciones e infraestructura de aplicaciones en la nube.
2. Explicar la importancia de la automatización en la construcción de aplicaciones y entorno de aplicaciones utilizando conceptos y herramientas de computación en nube.

3. Factores a tomar en cuenta previo a poder desplegar aplicaciones y sus entornos en la nube de manera sencilla y automática.
4. Indicar la importancia de la virtualización para administrar y automatizar aplicaciones y sus entornos en la nube.
5. Importancia de los servicios que presta la computación en nube, para eliminar la constante reconfiguración estática de infraestructura para un entorno informático dinámico.
6. Evaluación de herramientas de computación en nube, que nos permitan automatizar y representar de forma gráfica, la configuración de las aplicaciones y su entorno.
7. Importancia de una buena combinación de herramientas para la automatización de la construcción de aplicaciones y sus entornos.
8. Identificar las ventajas y desventajas existentes al momento de administrar y automatizar la construcción de aplicaciones y sus entornos en la nube.
9. Redactar un resumen que indique los factores a tomar en cuenta, pasos a seguir y herramientas a utilizar adecuadamente, para la automatización en la construcción de aplicaciones y sus entornos en la nube.

## INTRODUCCIÓN

La computación en nube es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado.

Es un nuevo modelo de aprovisionamiento de tecnología informática que permite proporcionar de manera dinámica recursos virtuales, cuyo aprovisionamiento puede controlarse programáticamente y por lo tanto permitir la creación del ecosistema que se presta al modelo de pago por uso de recursos.

La computación en nube trae beneficios reales y a corto plazo para los ejecutivos tanto de empresas pequeñas como grandes que reciben directamente las ventajas de la agilidad, maximización de recursos de cómputo y reducción de costos. Internamente dentro de las empresas se ven beneficiados los roles técnicos como el arquitecto de *software*, el programador de *software*, el DBA, el líder de QA, diseño y construcción.

La computación en nube permite aumentar el número de servicios basados en la red, lo que genera beneficios tanto para los proveedores, que pueden ofrecer de forma rápida y eficiente, un mayor número de servicios, así como para los usuarios, que tienen la posibilidad de acceder a ellos con transparencia e inmediatez del sistema de un modelo de pago por consumo.

En el primer capítulo se explican los conceptos básicos de computación en nube, tipos de nube, topologías de redes, capas dentro de la nube, virtualización, máquinas virtuales, así como la infraestructura virtual.

En el siguiente capítulo, se explican los factores a tomar en cuenta para la automatización de las aplicaciones y sus entornos dentro de la nube, tales como presupuesto, infraestructura, seguridad, etc.

En el tercero, que constituye la administración y automatización en la construcción de las aplicaciones y sus entornos, se explica la aplicación de los conceptos y las herramientas, así como los pasos necesarios a seguir adecuadamente para obtener máximos rendimientos, escalabilidad y disponibilidad de aplicaciones y su infraestructura. Adicionalmente, se explica con un ejemplo, cómo crear una infraestructura en *Amazon Elastic Compute Cloud* (EC2), así como instalar la aplicación de código abierto *Sugar CR Men* en la infraestructura de *Amazon*.

# **1. MARCO TEÓRICO PARA EL PROCESO DE AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE APLICACIONES Y SU ENTORNO EN LA NUBE**

## **1.1. Conceptos de computación en nube**

La computación en nube, es el acceso bajo demanda a recursos virtualizados de TI que se encuentran hospedados fuera del propio centro de datos de las empresas o personas que los utilizan. Dichos recursos son compartidos con otros, son fáciles de usar, pagados a través de suscripciones, y se acceden a través de la *web*.

La computación en nube es un tipo de computación que proporciona un acceso simple bajo demanda a los recursos informáticos de alta elasticidad. Estos recursos se ofrecen como un servicio en la red (con frecuencia en internet), y son ahora posibles gracias a una serie de innovaciones en las tecnologías de computación, operaciones y modelos de negocio. Permite a los consumidores de tecnología a pensar en la forma más eficaz de la informática sin límites, con un costo mínimo y confiable, así como a no preocuparse de la forma en cómo fue construida, cómo funciona, quién la administra o dónde se encuentra.

Está basado en Internet y en centros de datos remotos para gestionar servicios de información y aplicaciones. Permite que los consumidores y las empresas gestionen archivos y utilicen aplicaciones sin necesidad de instalarlas en cualquier computadora con acceso a Internet.

Ésta tecnología ofrece un uso más eficiente de recursos, como almacenamiento, memoria, procesamiento y ancho de banda.

El término nube, se utiliza como una metáfora de internet y se origina en la nube utilizada para representar internet en los diagramas de red como una abstracción de la infraestructura que representa.

## **1.2. Características de la nube**

Escalabilidad: aunque muchas características vienen a la mente cuando se habla de la nube, probablemente la primera en fijarse firmemente en la mente, es la relativa falta de preocupación por si una instalación se puede ampliar para atender cualquier demanda particular. La suposición implícita, es que siempre se puede ampliar según sea necesario, al menos para satisfacer la demanda en su conjunto. Tenga en cuenta que la escalabilidad de las aplicaciones individuales no es en absoluto requisito, hay nubes que cuentan con un gran número de aplicaciones relativamente modestas, y por lo tanto, todavía necesitan escala en su conjunto. De hecho, este es el caso común en muchas empresas, sobre todo en las primeras etapas del ciclo de adopción.

La escalabilidad simplemente trata de la ya muy conocida capacidad de una infraestructura o solución para responder ante cualquier demanda, sin que sus prestaciones sufran una degradación.

Elasticidad: una de las mayores críticas de la infraestructura tradicional de tecnología e información (TI), es lo difícil que es escalar recursos, ya sea hacia arriba o hacia abajo, frente a un cambio en la demanda de una aplicación. Ambos problemas conducen a un exceso de asignación de recursos, por lo



general resultando en una baja utilización de los recursos totales, a fin de tratar con picos de carga.

Con el fin de lograr un alto uso, es crucial que la nube sea elástica, es decir, que debe ser capaz fácilmente de aumentar o disminuir de forma automática, sin esfuerzo por parte del personal operativo en el momento de necesidad, y preferentemente, un mínimo o ningún esfuerzo por parte de los desarrolladores de aplicaciones.

Hay que tomar en cuenta que en el mejor de los casos, este nivel de flexibilidad se proporciona para todos y cada uno de los trabajos realizados en la nube, por lo general se mide en pequeñas fracciones de segundo.

Básicamente, la elasticidad es la capacidad para escalar tanto hacia arriba como hacia abajo de forma automática, sin esfuerzo por parte del personal operativo. Este concepto ya no es tan conocido. En el fondo parece asemejarse al anterior de la escalabilidad. En realidad existen sólo un par de matices que los diferencian. El primero es que se considera explícitamente la escalabilidad hacia abajo. Normalmente la escalabilidad se enfoca hacia la absorción de demandas crecientes. Eso puede conducir, y conduce de hecho, a sobredimensionamientos antieconómicos.

En el caso de la elasticidad, sin embargo, se contempla también explícitamente la escalabilidad hacia menores demandas. Otro matiz haría referencia a la relativa facilidad (en realidad los autores hablan de una facilidad casi absoluta), para conseguir esa adaptación sin intervención del personal operativo. A pesar de los dos matices mencionados, es discutible si la elasticidad es realmente una característica diferente de la escalabilidad o es sólo un grado de excelencia de la misma.

Autoservicio: tal vez ninguna otra característica de la computación en la nube ha capturado la imaginación de cualquiera que haya tratado de implementar una nueva aplicación de *software* antes de lo previsto, y se prepare para un aumento previsto de la demanda, o hacer frente a un flujo repentino de la demanda durante el uso de un modelo tradicional de TI que tenga autoservicio.

El proceso para agregar la capacidad en el modelo tradicional, por lo general, implica presupuesto, adquisiciones, planificación de instalaciones, personal, formación y más, con tiempos de espera que suelen ser de meses o años.

En contraste, la capacidad de autoservicio permite que el dueño de una aplicación obtenga los recursos de computación necesarios, o al menos la posibilidad de utilizar ciertos recursos de computación, con una simple solicitud de no más que unos minutos antes de la necesidad.

Este tipo de capacidad a menudo se implementa en los portales *web*, y se deriva de la capacidad que las primeras nubes públicas proporcionaron: la adquisición de infraestructura no fue más difícil que ordenar un libro de Amazon.

Aunque está estrechamente relacionado con la elasticidad, el autoservicio es diferente en ambos períodos de tiempo – minutos, horas, e incluso días en lugar de fracciones de segundo – y la intención principalmente es sobre cómo prepararse para una amplia gama de capacidades, frente a responder a las necesidades en cualquier momento en particular.

El autoservicio es la característica mediante la cual, el poseedor de una aplicación puede obtener los recursos computacionales necesarios mediante una solicitud atendida en pocos minutos. Los autores ponen esta característica en relación, de nuevo, con la elasticidad. Aparte de requerir de dicha elasticidad, lo que se está demandando es una alta automatización de la provisión de servicios y, sobre todo, de la provisión de capacidad.

Acceso ubicuo (servicios y más): es una característica esencialmente heredada como un subproducto de la nube, con la posibilidad de que todas las capacidades de la nube sean accesibles en cualquier lugar y mediante cualquier tipo de terminal (por lo menos a las capacidades de dicha terminal) o aplicación.

Virtualización completa (actúa como una): se trata de desacoplar completamente las aplicaciones de las infraestructuras que las soportan, convirtiendo toda esa infraestructura en una única máquina virtual sencilla de operar, y sobre la que resulte fácil el desarrollar o evolucionar aplicaciones. Los autores reconocen algunos éxitos en el camino de la virtualización, especialmente en lo relativo a los procesadores, pero observan menos éxitos en la virtualización del almacenamiento o las redes, y mucho menos aún en las aplicaciones.

En otras palabras, no importa cuánto una nube particular tenga que escalar, sigue siendo tan simple de operar y fácil de desarrollar aplicaciones, como si fuera solo un único servidor.

Esto es lo que se quiere decir con la virtualización completa. Hay que tener en cuenta, que en combinación con el acceso ubicuo, esto puede llevar a un verdadero sentido de la flexibilidad de ubicación.

Consistencia relativa: como una cuestión práctica, sobre todo a la complejidad operacional, incluso las primeras nubes se han construido a partir de un número relativamente pequeño de componentes únicos. Ni siquiera las operaciones avanzadas de TI, que se basan en la virtualización convencional, pueden tener cientos e incluso miles de componentes básicos de infraestructura única (por ejemplo, servidores, sistemas operativos, componentes de redes, etc.), que deben desplegarse y se encuentran en algún lugar del centro de datos. Lo que se sugiere, es la simplificación de las estructuras de la nube, mediante el uso de un número limitado de componentes.

Comoditización (productos básicos): lo que se sugiere, es evitar el uso de componentes y equipos muy especializados, por el contrario, utilizar en la construcción de las nubes, servidores, elementos de red o elementos de almacenamiento de propósito general, no muy especializados y, por tanto, más abundantes y a menor costo.

### **1.3. Características complementarias**

Mientras que las características anteriores son esenciales para cualquier esfuerzo de computación en nube, las siguientes características son opcionales, por lo menos en este punto en el tiempo. Es bastante claro que cada una de éstas, probablemente se convertirá en estándar para la mayoría de las nubes en los próximos años, sino es que antes.

Servicio medido: casi todas las nubes públicas siempre han tenido la capacidad de facturar precisamente por la cantidad de recursos consumidos, sin compromiso previo, es decir, la capacidad de medir el uso de los recursos que hace el cliente o usuario y facturar por ese uso, de forma similar a como

tradicionalmente se ha hecho en el caso de servicios de telecomunicaciones. Esta característica parece irrenunciable para construir el correlato económico del concepto '*As a Service*'.

Multi cliente: la presencia de múltiples clientes en la misma nube es ciertamente el caso de casi todas las nubes públicas, es simplemente una cuestión de economía, se trata de que la nube soporte a múltiples clientes evitando el uso de recursos dedicados.

Multi aplicaciones: casi todas las nubes son inherentemente multi-aplicaciones, es decir, ejecutar varias aplicaciones de software individuales en la misma infraestructura. Sin embargo, hay ciertas aplicaciones de alto valor para las que una nube dedicada tiene sentido.

Escalabilidad de aplicaciones: si bien todas las nubes necesitan tener la capacidad innata de escalar fácilmente como un conjunto, la habilidad para permitir a las aplicaciones individuales lograr la escala *web*, puede claramente estar reservada para aquellos casos donde sea necesario.

Como cuestión práctica, esto conduce a un modelo de adopción donde una organización puede adoptar la nube y comenzar con las aplicaciones, excepto las que se están ejecutando en la nube, y a continuación, hacer éstas aplicaciones individuales más nativas en la nube, y por lo tanto más escalables, como dicten las necesidades y lo permita el tiempo y el presupuesto.

Dicho esto, muchas organizaciones fueron dirigidas inicialmente a desarrollar y/o adoptar la computación en nube por la necesidad de escalar una aplicación individual, ya sea, la búsqueda para *Google*, comercio electrónico (e-

*commerce*) para Amazon, o de optimizaciones de entrega para *Federal Express*.

Se trata, no sólo de que las infraestructuras sean escalables, sino de que cada aplicación, considerada individualmente, tenga esas características de escalabilidad, lo cual parece conducir a unos diseños y arquitecturas de aplicaciones más orientados a la nube.

Fiabilidad: a primera vista esto puede parecer un castillo en el aire, tal vez una capacidad que es mejor dejar para los casos exigentes, los difíciles, de alto costo que atraen solo a los más atrevidos. Sin embargo hay dos aspectos sorprendentes de discutir con la alta fiabilidad del *software* basado en la nube.

En primer lugar, las aplicaciones que son capaces de garantizar su propio funcionamiento fiable, pueden ser fácilmente desplegadas a un menor costo, infraestructura completa de productos básicos. En otras palabras, no importa si fallan los componentes subyacentes. Por lo tanto, la construcción de la fiabilidad en las aplicaciones realmente permitirá a una organización reducir sus costos.

En segundo lugar, debido al gran número de componentes que son usados para construir una infraestructura en la nube (en comparación con una infraestructura de TI tradicional), es actualmente posible para el *software* de nube inteligente, desarrollar un mayor nivel de confiabilidad. Aunque este tipo de capacidad no es todavía común en la computación en nube. Con la llegada de plataformas de aplicaciones en nube más sofisticadas, será posible garantizar la fiabilidad de forma rutinaria para casi todas las aplicaciones, sin importar cuán agresivo sea el consumo masivo de la infraestructura subyacente.

Es una característica que se considera aún poco desarrollada y que, sin embargo, puede paradójicamente, hacer reducir costos debido a las menores necesidades de operación.

Es difícil decir si éstas características son las que realmente definen lo que se necesita en la computación en nube, si se menciona todo lo necesario, o si alguna característica pudiera no ser tan relevante. Lo que sí parece, es que ayudan a centrar y entender éste emergente mundo de la computación en nube.

#### **1.4. Ventajas de negocio al utilizar la computación en nube**

Hay beneficios claros para la creación de aplicaciones en la nube, de los cuales algunos se enumeran a continuación:

- Inversión casi nula en infraestructura por adelantado: si se tiene que construir un sistema a gran escala que puede costar una fortuna para invertir en bienes raíces, seguridad física del *hardware* (*racks*, servidores, ruteadores, suministros de energía de reserva –UPS–), administración de *hardware* (administración de energía, refrigeración), y el personal de operaciones. Debido a los altos costos iniciales, el proyecto normalmente requiere de varias rondas de gestión para la aprobación antes de que el proyecto pueda incluso empezar. Ahora, con los servicios públicos de la computación en la nube no hay costo fijo o costo inicial.
- Infraestructura justo a tiempo: en el pasado, si una aplicación se hizo popular y los sistemas o la infraestructura no escalan, la empresa se convirtió en una víctima de su propio éxito. Por el contrario, si se hizo

una gran inversión y no se hizo popular, la empresa se convirtió en una víctima de su fracaso. Mediante el despliegue de aplicaciones en la nube con auto aprovisionamiento justo a tiempo, no hay que preocuparse antes de adquirir la capacidad que se necesita para sistemas a gran escala. Esto aumenta la agilidad, disminuye el riesgo y reduce los costos operativos, ya que solo escala a medida que crece y se paga solo por lo que se usa.

- Utilización más eficiente de los recursos: los administradores de sistemas suelen preocuparse por la adquisición de hardware (cuando se les acaba la capacidad), y una mayor utilización de la infraestructura (cuando tienen exceso de capacidad y reposo). Con la nube, se pueden administrar los recursos más eficaz y eficientemente al tener la solicitud de las aplicaciones y renunciar a la demanda de recursos.
- Costo basado en el uso: con un precio de estilo de servicios públicos, se factura solo por la infraestructura que se ha utilizado. No se paga por los recursos asignados y no utilizados de infraestructura. Esto añade una nueva dimensión al ahorro de costos. Se puede obtener un ahorro inmediato de costos (a veces tan temprano como la factura del próximo mes), al implementar un parche de optimización para actualizar la aplicación de nube.

Por ejemplo, si una capa de almacenamiento en caché puede reducir las solicitudes de datos en un 70 por ciento, los ahorros comienzan a acumularse inmediatamente y se verá la recompensa en la siguiente factura. Por otra parte, si se está construyendo plataformas en la parte superior de la nube, se puede pasar la misma estructura de costos variable y flexible en base al uso a los propios clientes.



- **Tiempos reducidos de comercialización:** la paralelización es una de las grandes maneras de acelerar el procesamiento. Si un trabajo de cálculo intensivo o datos intensivos, que se puede ejecutar en paralelo necesita 500 horas para el proceso en una máquina, con la arquitectura de nube sería posible generar y poner en marcha 500 instancias y procesar el trabajo en 1 hora. Disponer de una infraestructura elástica proporciona a la aplicación, la capacidad de explotar la paralelización de una manera costo-efectiva reduciendo el tiempo en el mercado.

### **1.5. Ventajas técnicas de computación en nube**

Algunas de las ventajas técnicas de la computación en nube incluyen:

- **Automatización:** se pueden crear sistemas repetibles de construcción y despliegue mediante el aprovechamiento de la infraestructura programable de secuencia de comandos (API).
- **Autoescala:** se puede cambiar el tamaño de las aplicaciones hacia arriba y hacia abajo para que coincida con la demanda inesperada, sin ninguna intervención humana. Auto escala permite la automatización y conduce a una mayor eficiencia.
- **Escala proactiva:** permite escalar la aplicación hacia arriba y hacia abajo, para satisfacer la demanda prevista con la comprensión de una adecuada planificación de los patrones de tráfico, para mantener bajos los costos mientras escala.
- **Ciclo de vida de desarrollo más eficiente:** los sistemas de producción pueden ser clonados fácilmente para su uso como entornos de

desarrollo y pruebas. Los entornos de ensayo pueden ser promovidos fácilmente a producción.

- Capacidad de prueba mejorada: nunca se queda sin equipo para las pruebas. Inyecta y automatiza las pruebas en todas las etapas durante el proceso de desarrollo. Se puede generar un laboratorio de pruebas instantáneo con ambientes preconfigurados sólo para la duración de la fase de prueba.
- Recuperación de desastres y continuidad del negocio: la nube proporciona una opción de menor costo para el mantenimiento de una flota de servidores y almacenamiento de datos. Con la nube, se puede tomar ventaja de la geodistribución y replicar el entorno en otro lugar en cuestión de minutos.
- Desbordamiento del tráfico hacia la nube: con unos pocos clics y tácticas efectivas de balanceo de cargas, se puede crear una aplicación completa a prueba de desbordamientos, enviando el exceso de tráfico hacia la nube.

#### **1.6. Conceptos de computación *grid* o computación distribuida**

La computación distribuida es un conjunto de computadoras conectadas entre ellas que comparten recursos; es un modelo de procesamiento que permite resultados más rápidos que los que se podrían lograr con un modelo cliente-servidor, en el cual existe un servidor central (*Mainframe*), que concentra todas las transacciones y cuenta con toda la información y datos, así como con la lógica del negocio; mientras que cuando el cliente (usuario), hace una petición a la nube, es atendido por muchos computadores, logrando así

tiempos de respuesta mucho más rápidos y una mayor disponibilidad, puesto que el sistema no depende de un solo servidor.

### *Grid* y computación de alto rendimiento

La búsqueda de la escala es muy antigua en la informática, es decir, siempre ha sido un intento por hacer más. Más a menudo, más datos y más análisis. Estos objetivos han sido siempre la presa de la industria de la computación y de los científicos de investigación.

A finales de 1970 y principios de 1980, esto condujo al incremento de las supercomputadoras. Durante la década de 1980 y principios de 1990, otra idea persistente se mantuvo emergente: ¿Se pueden hacer con un gran número de componentes básicos y más baratos?

Inicialmente esto condujo a una serie de nuevas empresas que construyeron más y más grandes servidores con múltiples procesadores relativamente baratos. Esta fue la primera motivación para servidores de nivel de muchas entradas y servidores de nivel medio para muchas empresas. Eventualmente, incluso muchos grandes proveedores de servidores de la nube, afirmaron que estaban utilizando los componentes básicos, en cierto sentido.

En cualquier caso, el sueño de construir colecciones de un gran número de computadoras básicas en una instalación, útil para resolver grandes problemas, finalmente echó raíces serias a finales de 1990 y la primera parte de la década siguiente.

Debido a esta historia, estos sistemas tienden a ser orientados a lotes, sin mucho sentido ya sea de fiabilidad o de integridad de transacciones; estos

fueron, después de todo, generalmente grandes problemas informáticos; y es también complejo para los desarrolladores y los operadores. Todas estas consideraciones fueron generalmente aceptables por varias razones: En primer lugar, estaba bien para los problemas en cuestión; y segundo, los laboratorios, instituciones de seguridad y universidades en general tenían el personal para hacer frente a la complejidad.

Dado que muchas de las infraestructuras eran relativamente poco frecuentes o caras, con el tiempo hubo una campaña para crear redes (*grids*), en las que el exceso de capacidad podía ser compartido (con el mismo nombre que viene la idea de las redes de energía eléctrica, que se utiliza a menudo como una metáfora para las nubes públicas). Desafortunadamente, mientras que parecían atractivas en un primer momento, el concepto de *grid* usualmente contribuía a la complejidad operacional inherente y al tiempo de previsibilidad de estos sistemas. Como resultado, éstas tendían a ser útiles solamente para preguntas de investigación científica a largo plazo, como la búsqueda de curas para el cáncer o la vida inteligente más allá de la tierra.

Con el auge de la computación en nube algunos de estos esfuerzos son naturalmente enfocados a ser en sí mismos amigables con la nube, y están en varias etapas de progreso en este esfuerzo. Como cuestión práctica, algunos de los modelos de programación y APIs pueden hacer que la transición para los problemas de investigación especializados se dirija bien, pero la mayoría de los otros tienden a ser suplantados por tecnologías basadas en la nube.

## **1.7. Topología de redes**

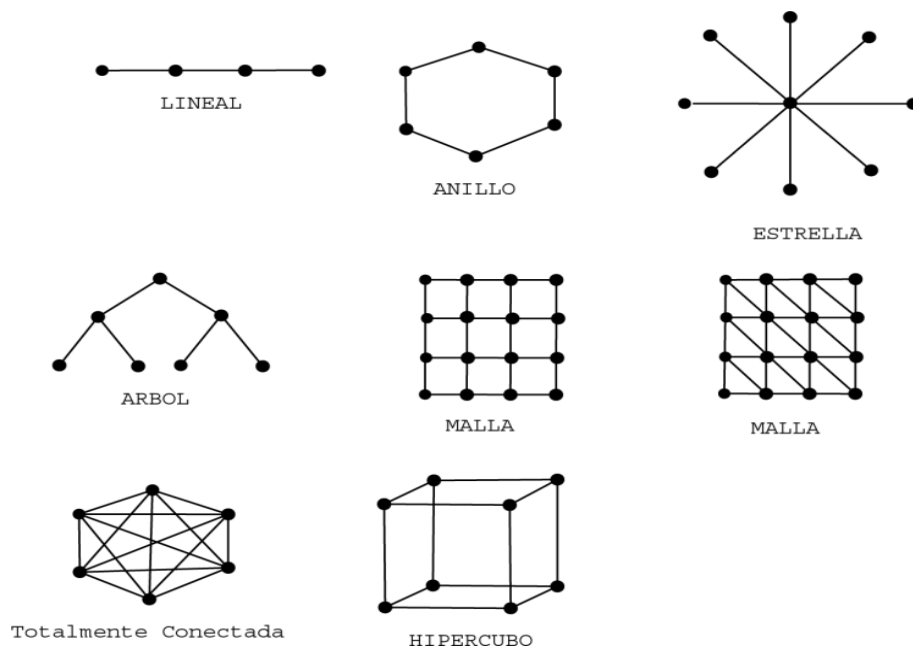
La topología de red se define como la cadena de comunicación usada por los nodos que conforman una red física para comunicarse.

- En *bus*: red cuya topología se caracteriza por tener un único canal de comunicaciones (denominado bus o troncal), al cual se conectan los diferentes dispositivos. De esta forma todos los dispositivos comparten el mismo canal para comunicarse entre sí.
- En *estrella*: es una red en la cual las estaciones están conectadas directamente a un punto central, y todas las comunicaciones se han de hacer necesariamente a través de éste. Los dispositivos no están directamente conectados entre sí, además de que no se permite tanto tráfico de información. Dada su transmisión, una red en estrella activa, tiene un nodo central activo, que normalmente tiene los medios para prevenir problemas relacionados con el eco. Se utiliza sobre todo para redes locales. La mayoría de las redes de área local que tienen un enrutador (*router*), un conmutador (*switch*), o un concentrador (*hub*), siguen esta topología. El nodo central en éstas sería el enrutador, el conmutador o el concentrador por el que pasan todos los paquetes de red.
- *Árbol*: topología de red en la que los nodos están colocados en forma de árbol. Desde una visión topológica, la conexión en árbol es parecida a una serie de redes en estrella interconectadas, salvo en que no tiene un nodo central. En cambio, tiene un nodo de enlace troncal, generalmente ocupado por un concentrador o conmutador, desde el que se ramifican los demás nodos. Es una variación de la red en bus, la falla de un nodo, no implica interrupción en las comunicaciones. Se comparte el mismo canal de comunicaciones.
- *Anillo*: topología de red en la que cada estación está conectada a la siguiente y la última está conectada a la primera; cada estación tiene un

receptor y un transmisor que hace la función de repetidor, pasando la señal a la siguiente estación. En un anillo doble, dos anillos permiten que los datos se envíen en ambas direcciones, ésta configuración crea redundancia (tolerancia a fallos).

- Malla: topología de red en la que cada nodo está conectado a todos los nodos; de ésta manera es posible llevar los mensajes de un nodo a otro por diferentes caminos. Si la red de malla está completamente conectada, no puede existir absolutamente ninguna interrupción en las comunicaciones; cada servidor tiene sus propias conexiones con todos los demás servidores.

Figura 1. Topología de redes estáticas



Fuente: elaboración propia.

La red virtual: una red virtual, es una red de computadoras creada por encima de otra red. Los nodos en la red virtual pueden ser pensados como conectados por vínculos virtuales o lógicos, cada uno de los cuales corresponde a una vía, quizás a través de muchos vínculos físicos en la red subyacente. Las redes virtuales basadas en la nube pueden ser utilizadas para mantener el control en múltiples lugares, permitiéndole al cliente controlar la topología de su propia red, el direccionamiento de la misma, la comunicación cifrada y los protocolos de red deseados, todo de una forma escalable y altamente redundante.

Las redes virtuales están basadas en conexiones punto-a-punto redundantes y cifradas de los servidores, basados en la red a los dispositivos virtuales híbridos que se ejecutan en la nube. Estos dispositivos híbridos actúan como conmutadores y ruteadores virtuales de esta red virtual, la cual esencialmente, se encuentra sobre su red física. Una red virtual le da a un cliente una red común como la LAN, en donde los servidores pueden ser ubicados en el centro de datos físico (virtualizado o no), en la nube privada y en la pública.

Las ventajas de las redes virtuales en nube son:

- Le permiten al usuario el control sobre:
  - El direccionamiento (direccionamiento privado y adaptado para los servidores basados en la nube).
  - La topología son las redes virtuales compuestas de conmutadores virtuales, puentes virtuales y ruteadores virtuales.

- Los protocolos están disponibles para la elección/el descubrimiento de los servicios.
- La comunicación cifrada entre los servidores desplegados en nube.
- La conexión segura con las soluciones extranet basadas en el centro de datos existente a través de un túnel *IPsec*.
- La abstracción del lugar desde la identidad de la red, la cual permite una recuperación de desastres veloz y sencilla.
- Provee un nivel de seguridad controlado por el usuario para la creación de informes y la auditoría de cumplimiento.
- Permite monitorear y administrar los servidores basados en la red.

## **1.8. Tipos de nube**

Existen tres tipos de nube dependiendo de la necesidad de cada organización y el modelo de servicio ofrecido, básicamente son:

### **1.8.1. Nubes públicas**

Se refieren al modelo estándar de computación en nube, donde los servicios que se ofrecen se encuentran en servidores externos al usuario, donde se puede tener acceso de forma gratuita o de pago a las aplicaciones.



### **1.8.2. Nubes privadas**

La plataforma de la nube privada, se encuentra dentro de las instalaciones de la empresa u organización, y normalmente no da servicios a terceros.

Es una plataforma para la obtención de *hardware* solamente, es decir, máquinas, almacenamiento e infraestructura de red (en inglés *Infrastructure as a Service, IaaS*), pero también se puede tener una nube privada que despliegue aplicaciones (en inglés *Platform as a Service, PaaS*), e incluso software (en inglés *Software as a Service, SaaS*).

Son una buena opción para la protección de datos y ediciones de servicio para compañías que lo necesitan.

En las nubes privadas, el cliente controla qué aplicaciones usa y cómo las usa. La empresa es la propietaria de la infraestructura y puede decidir qué usuarios están autorizados para utilizarla.

### **1.8.3. Nubes híbridas**

Este tipo de nube combina recursos locales de una nube privada con la nube pública.

La infraestructura privada, se ve aumentada con los servicios de computación en nube, de la infraestructura de la nube pública, esto permite, aprovechar la computación en nube pública cuando resulte necesario, pero permite que la empresa mantenga el control de sus principales aplicaciones.

## 1.9. Capas

El uso de la virtualización en las nubes ha creado un nuevo conjunto de capas: aplicaciones, servicios, e infraestructura. Estas capas no sólo encapsulan los recursos en demanda, también definen un nuevo modelo de desarrollo de aplicación.

### 1.9.1. **Software como servicio (Software as a Service, SaaS)**

Esta es la capa más alta y consiste en la entrega de una aplicación completa como un servicio.

El proveedor SaaS, dispone de una aplicación estándar desarrollada, en algunos casos, por él mismo, que se encarga de operar y mantener, y con la que da servicio a multitud de clientes a través de la red, sin que estos tengan que instalar ningún *software* adicional.

La distribución de la aplicación tiene el modelo de uno a muchos, es decir, se realiza un producto y el mismo lo usan varios clientes.

Los proveedores de SaaS son responsables de la disponibilidad y funcionalidad de sus servicios, no dejando de lado las necesidades de los clientes que son, al fin y al cabo, los que usaran el *software*.

Un ejemplo claro, es la aplicación para el manejo del correo electrónico (como *Gmail*, *Hotmail*, *Yahoo*, etc.), por medio de un navegador *web*.

### **1.9.2. Infraestructura como servicio (Infrastructure as a Service, IaaS)**

Es un modelo de prestación de servicios en la que una organización subcontrata el equipo utilizado para soportar las operaciones, incluyendo el almacenamiento de información, *hardware*, servidores y componentes de red.

### **1.9.3. Plataforma como servicio (Platform as a Service, PaaS)**

Es una forma de alquiler de *hardware*, sistemas operativos, capacidad de almacenamiento y de red en *Internet*. El modelo de entrega de servicio permite al cliente, alquilar servidores virtualizados y servicios para la ejecución de las aplicaciones existentes, o desarrollar y probar los nuevos asociados.

## **1.10. Aplicaciones y entorno de aplicaciones en la nube**

La aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o varios tipos de trabajo.

Suelen resultar una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén.

Ciertas aplicaciones desarrolladas a la medida pueden ofrecer una gran potencia, ya que están exclusivamente diseñadas para resolver un problema específico. Existen otros llamados paquetes integrados de *software* que ofrecen menos potencia, pero a cambio incluyen varias aplicaciones, como un programa procesador de textos, hojas de cálculo y base de datos.

El entorno de aplicaciones se refiere a la infraestructura de soporte a través del ciclo de vida del *software*, o etapas a través de las cuales se mueven la mayoría de aplicaciones, los entornos de aplicaciones pueden variar su configuración dependiendo de la etapa de desarrollo, etapa de verificación y la etapa de producción de las aplicaciones.

## **1.11. Virtualización**

Es la creación -a través de *software*- de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de *hardware*, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red.

### **1.11.1. Qué es la virtualización**

La virtualización es esencial en el desarrollo óptimo de la computación en la nube, y está referida principalmente al tema de plataforma.

Se puede decir que la virtualización, es una abstracción de los recursos tecnológicos que permite a los servidores crear dispositivos virtuales, los cuales pueden ser usados para aumentar los recursos.

En la computación en nube, es interesante el tema de la virtualización que permite tratar a un servidor como muchos servidores.

Otro tema interesante es la agrupación (*clustering*), que permite tratar a muchos servidores como uno solo. Esto permite muchos mejoramientos tales como:

- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Reducción de los costos de espacio y consumo.
- Administración global centralizada y simplificada.
- Mayor facilidad para la creación de entornos de pruebas que permitan poner en marcha nuevas aplicaciones sin impactar a la producción, agilizando el proceso de las pruebas.
- Aislamiento: un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales.

No sólo aporta el beneficio directo en la reducción del *hardware* necesario, sino también de los costos asociados.

### **1.11.2. Cómo funciona la virtualización**

Una pieza clave en la marcha hacia la computación en nube, ha venido con una mayor adopción de la virtualización de sistemas operativos para PC, comenzando con el trabajo realizado en la comunidad Linux, y que lleva a las versiones comerciales de *VMWare*, *Microsoft* y otros.

Siguiendo el ejemplo del mundo del mainframe, los arquitectos de sistemas operativos para PCs, permiten a sistemas completos de servidores existir dentro de un contenedor virtual en un host, lo que permite una utilización más completa de los recursos del sistema anfitrión mientras se mantiene el control.

Piense en esto como poner una casa dentro de una gran bodega: es una casa completa, pero está conectada con el mundo exterior a través de las conexiones que son propiedad de la bodega.

Ahora, por primera vez, todo un sistema operativo, aplicaciones y accesorios se pueden empaquetar en un conjunto de archivos que se pueden ejecutar en cualquier medio ambiente compatible virtualizado. Un medio ambiente que para todos los efectos, se ve y se siente igual a un ambiente no virtualizado.

Estos entornos virtualizados pueden albergar múltiples máquinas virtuales para utilizar *hardware* de manera más eficiente y aumentar dramáticamente los rendimientos potenciales.

La virtualización está de moda. Si antes era una tecnología al alcance solamente de grandes compañías, ahora cualquier empresa pequeña o mediana tiene al alcance los beneficios de la virtualización.

La reducción de precios en el *hardware* y la presencia de productos *software* de virtualización cada vez más asequibles, han democratizado ésta tecnología que, bien utilizada, puede poner a nuestra disposición más opciones de sistemas operativos, costes menores y un mayor control sobre nuestra estructura.

Dentro de la estructura informática de una empresa, se ha visto que los servidores pueden desempeñar un papel importante, para centralizar recursos y utilizar herramientas específicas que incrementan la productividad, a través de la mejora de los procesos de producción, planificación o comunicación.

La tecnología nos ofrece sin embargo, la posibilidad de disponer de varios servidores con características muy distintas, pero instalando físicamente en una sola máquina. Este sistema se llama virtualización. A través de esta tecnología, es posible hacer que los recursos de un ordenador, en este caso un servidor, puedan ser compartidos por una o más máquinas virtuales, que se comportarán a su vez, como servidores reales.

A cada una de estas máquinas virtuales, se les pueden asignar recursos de *hardware*, diseñando distintas configuraciones con características independientes. Estos recursos pueden ser compartidos, o se pueden bloquear de forma que cada máquina virtual tenga su propia memoria *RAM*, CPU, disco duro, recursos de red, etc. En cada una de estas máquinas, podemos instalar su propio sistema operativo y sus aplicaciones independientes

### **1.11.3. Ventajas y desventajas de la virtualización**

Es la reutilización de hardware existente para optimizar el aprovechamiento de todos los recursos de hardware. Sirve para realizar una rápida incorporación de nuevos recursos para los servidores virtualizados.

#### Ventajas

- Ahorro de costos: se puede adquirir un solo servidor, aunque más potente, y no tener que comprar más servidores, sino solamente ir creándolos en el gestor de máquinas virtuales. También permite ahorros en el costo de mantenimiento y en el de personal, además de ahorrar espacio. La virtualización es una excelente opción hoy día, ya que las máquinas actuales (*laptops*, *desktops*, servidores), en la mayoría de los casos están siendo subutilizados (gran capacidad de disco duro,

memoria *RAM*, etc.), llegando a un uso de entre 30 a 60 por ciento de su capacidad. Al virtualizar, la necesidad de nuevas máquinas en una ya existente permite un ahorro considerable de los costos asociados (energía, mantenimiento, espacio, etc.).

- Crecimiento más flexible: instalar un nuevo servidor es mucho más sencillo y rápido frente a hacerlo con un servidor físico.
- Administración simplificada: desde la consola del gestor de máquinas virtuales se puede aumentar o reducir los recursos para una determinada máquina, reiniciarla, instalar parches o simplemente borrarla en caso de problemas.
- Aprovechamiento de aplicaciones antiguas: posibilidad de conservar aplicaciones que funcionan en sistemas antiguos y aún así modernizar la infraestructura informática de la empresa. Esa aplicación puede sobrevivir en una máquina virtual independiente sin que haga falta conservar el ordenador antiguo.
- Centralización de tareas de mantenimiento: se pueden realizar copias de seguridad de un solo golpe de todas las máquinas, programar actualizaciones y otras actividades desde el gestor de máquinas virtuales. También se pueden centralizar otras funciones.
- Disminuye tiempos de parada: una ventaja importante, solucionar problemas o realizar copias de seguridad son tareas que se realizan en mucho menos tiempo. Por ejemplo, se puede clonar una máquina y seguir dando servicio mientras se realiza mantenimiento de la máquina virtual de producción como actualizaciones.



- Mejor gestión de recursos: se puede aumentar la memoria o almacenamiento de la máquina huésped para aumentar los recursos de todas las máquinas virtuales a la vez, por lo que se aprovecha mucho mejor las inversiones en *hardware*.
- Balanceo de recursos: es posible asignar un grupo de servidores físicos, para que proporcionen recursos a las máquinas virtuales, y asignar una aplicación que haga un balance de los mismos, otorgando más memoria, recursos de la CPU, almacenamiento o ancho de banda de la red a la máquina virtual que lo necesite.

#### Desventajas

- Rendimiento inferior: un sistema operativo virtualizado nunca alcanzará el mismo rendimiento que si estuviera directamente instalado en la computadora físicamente. Dado que el hipervisor introduce una capa intermedia en la gestión del *hardware* para las peticiones de acceso y la concurrencia al mismo, el rendimiento de la máquina virtual se ve muy afectado.
- *Hardware* virtual obsoleto: algunos de los dispositivos tales como *USB 1.0*, *Firewire 400*, *Ethernet 100* son a los que se verán sometidos. Los hipervisores se van renovando, pero hay que trasladarse al pasado para utilizar estas características.
- No se puede utilizar *hardware* que no esté soportado por el hipervisor: es decir, no se puede utilizar arbitrariamente hardware instalado físicamente en la máquina anfitrión desde el entorno virtual. El *software*

de virtualización impondrá una serie de dispositivos virtuales como tarjetas de vídeo y red de las que no se puede salir.

- Proliferación de máquinas virtuales: como no hay que comprar físicamente los ordenadores, el número de máquinas y servidores virtuales se dispara en todos los ámbitos. Los efectos colaterales se perciben después: aumenta el trabajo de administración, gestión de licencias, riesgos de seguridad, etc.
- Desperdicio de recursos: crear máquinas virtuales innecesarias tiene un costo en ocupación de recursos, principalmente en espacio en disco, memoria *RAM* y capacidad de procesamiento.
- Un daño en el servidor anfitrión de virtualización afecta a todas las máquinas virtuales alojadas en él: se debe adoptar ahora más que nunca, soluciones de alta disponibilidad como agrupamiento y replicación para evitar caídas de servicio de múltiples servidores con una única avería.

## **1.12. Máquinas virtuales**

Brevemente significa que es un duplicado eficiente y aislado de una máquina física. La acepción del término actualmente incluye a máquinas virtuales que no tienen ninguna equivalencia directa con ningún *hardware* real.

### **1.12.1. Qué es una máquina virtual**

Este *software* en un principio, fue definido como un duplicado eficiente y aislado de una máquina física. Una característica esencial de las máquinas

virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas. Estos procesos no pueden escaparse de esta computadora virtual.

### 1.12.2. Tipos de máquinas virtuales

#### Máquinas virtuales de sistema

También llamadas máquinas virtuales de *hardware*, permiten a la máquina física subyacente multiplicarse entre varias máquinas virtuales, cada una ejecutando su propio sistema operativo. A la capa de *software* que permite la virtualización se la llama monitor de máquina virtual o hipervisor. Un monitor de máquina virtual puede ejecutarse o bien directamente sobre el *hardware* o bien sobre un sistema operativo.

#### Máquinas virtuales de proceso

Una máquina virtual de proceso, a veces llamada máquina virtual de aplicación, se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo proceso. La máquina se inicia automáticamente cuando se inicia el proceso que se desea ejecutar y se detiene para cuando éste finaliza. Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de *hardware* y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma. El ejemplo más conocido actualmente de este tipo de máquina virtual es la máquina virtual de java. Otra máquina virtual muy conocida es la del entorno *.Net de Microsoft* que se llama *Common Language Runtime*.

### 1.12.3. Ventajas y desventajas de las máquinas virtuales

#### Ventajas

- Varios sistemas operativos distintos pueden coexistir sobre la misma computadora, en sólido aislamiento el uno del otro, por ejemplo para probar un sistema operativo nuevo sin necesidad de instalarlo directamente.
- Varias máquinas virtuales (cada una con su propio sistema operativo llamado sistema operativo invitado o *guest*), pueden ser utilizadas para consolidar servidores. Esto permite que servicios que normalmente se tengan que ejecutar en computadoras distintas para evitar interferencias, se puedan ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. La consolidación de servidores a menudo contribuye a reducir el costo total de las instalaciones necesarias para mantener los servicios, dado que permiten ahorrar en *hardware*.
- La máquina virtual puede proporcionar una arquitectura de instrucciones que sea algo distinta de la verdadera máquina. Es decir, podemos simular hardware.
- Reduce los costos y riesgos al tiempo que aumenta la calidad y agilidad de su infraestructura de TI.

## Desventajas

- Agregan gran complejidad al sistema en tiempo de ejecución, esto tiene como efecto la ralentización del sistema, es decir, el sistema no alcanzará la misma velocidad de ejecución que si se instala directamente en el sistema operativo anfitrión o directamente sobre la plataforma de *hardware*. Sin embargo, a menudo la flexibilidad que ofrecen compensa esta pérdida de eficiencia.
- Los lenguajes basados en máquina virtual, poseen la desventaja que son más lentos que los lenguajes completamente compilados, debido a la sobrecarga que genera tener una capa de *software* intermedia entre la aplicación y el *hardware* de la computadora, pero esta desventaja no es crítica.
- Una falla de *hardware* equivale a la falla de todos los sistemas virtuales que operan sobre el mismo.
- Para asegurar los servicios se debe adquirir *hardware* de gran capacidad y por lo tanto de alto costo.

### **1.13. Infraestructura virtual**

La infraestructura virtual permite compartir los recursos físicos de varias máquinas en toda una infraestructura de una empresa, La infraestructura virtual simplifica la tecnología TI de tal manera que las compañías aprovechan sus recursos computacionales.

### **1.13.1. Que es la infraestructura virtual**

Una infraestructura virtual consiste en el mapeo dinámico de recursos físicos en función de las necesidades de la empresa. Una máquina virtual representa los recursos físicos de un único ordenador, mientras que una infraestructura virtual representa los recursos físicos de la totalidad del entorno de TI, aglutinando ordenadores, así como su red y almacenamiento asociados, en un grupo unificado de recursos de TI.

### **1.13.2. Componentes**

Estructuralmente, una infraestructura virtual consta de los siguientes componentes:

- Hipervisor de un solo nodo para hacer posible la virtualización de todos los ordenadores.
- Un conjunto de servicios de infraestructura de sistemas distribuida basada en la virtualización, como gestión de recursos, para optimizar los recursos disponibles entre las máquinas virtuales.
- Soluciones de automatización que proporcionen capacidades especiales para optimizar un proceso de TI concreto, como aprovisionamiento o recuperación ante desastres. Mediante la separación de la totalidad del entorno de *software* de su infraestructura de *hardware* subyacente, la virtualización hace posible la reunión de varios servidores, estructuras de almacenamiento y redes en grupos compartidos de recursos que se pueden asignar de forma dinámica, segura y fiable a las aplicaciones según sea necesario. Este enfoque innovador permite a las

organizaciones crear una infraestructura informática con altos niveles de utilización, disponibilidad, automatización y flexibilidad utilizando componentes básicos de servidores económicos y estándar del sector.

### **1.13.3. Ventajas y desventajas**

#### Ventajas

- Índices de utilización del 60 al 80 por ciento para servidores (frente al 5 a 15 por ciento en *hardware* no virtualizado)
- Capacidad para el aprovisionamiento de nuevas aplicaciones en cuestión de minutos, en lugar de días o semanas, 85 por ciento de mejora en tiempo de recuperación de paradas imprevistas.
- Costos substancialmente más bajos: es decir, se reduce la complejidad operacional a través de la automatización y el manejo fácil de los ambientes de trabajo, así como la mejor utilización de los recursos de toda la infraestructura de computación de la empresa.
- Disponibilidad mejorada: ya que se eliminan los calendarios de tiempo de inactividad y se tiene mayor control sobre los niveles de servicio.
- Flexibilidad del negocio: capacidad de soportar cambios en las políticas y prioridades del negocio de forma dinámica, así como la capacidad de focalizar los recursos en el negocio.

Desventajas: hay algunos obstáculos con la infraestructura virtual, que pueden afectar la adopción de ésta en una organización y hacer lenta la migración de la industria hacia las iniciativas empresariales basadas en la nube: Algunos de estos son:

- Desafíos de seguridad y privacidad: la seguridad de la infraestructura virtual (nube), y las preocupaciones asociadas a la privacidad, hace reflexionar a muchas organizaciones sobre sus preocupaciones particulares de la computación en nube. Las preocupaciones de seguridad y privacidad incluye la seguridad física y fácil acceso a las instalaciones y equipos, así como la seguridad lógica, los requisitos de cumplimiento de la industria, auditabilidad y más. Hay también dos perspectivas: (1) donde el vaso de seguridad esta medio lleno y (2) en el que está medio vacío.

El punto de vista del vaso medio lleno considera que las preocupaciones de seguridad en la infraestructura virtual (nube), son manejables, y de hecho, son mejores cuando son manejadas por un tercero proveedor de servicios de nube.

El punto de vista del vaso medio vacío, ve todos los desafíos de seguridad como vallas que son inamovibles y no se pueden superar, independientemente del perfil de la empresa que amerita el abordaje en una nube. Al igual que con los desafíos de seguridad que acompañaron a los servicios web y SOA, la arquitectura de seguridad y los modelos asociados con la infraestructura virtual, se debatieron de forma similar y se superaron tranquilamente con las soluciones de seguridad tal como evoluciona la industria.



- **Fiabilidad y confianza:** las interrupciones de la infraestructura virtual (nube), están bien documentadas y altamente publicitadas, especialmente cuando los defensores principales de computación en nube, como *Amazon*, *Google*, *Rackspace* y otras experimentan estos desafíos. Si el tono de llamada a la nube no puede ser asegurado, de manera que los consumidores de nubes sepan que sus recursos habilitados en la nube siempre estarán ahí para ellos, entonces la nube será relegada a nichos donde la disponibilidad de la red se puede asegurar. La nube como SOA, es un modelo basado en la confianza, donde la falta de confianza severamente paralizará la adopción de la infraestructura virtual (nube), por las masas.
- **Integración e interoperabilidad de la nube:** la integración e interoperabilidad de nubes privadas a privadas, de nubes públicas a privadas, públicas a públicas, e híbridas, representan un gran desafío en ausencia de estándares de la industria para las APIs y las interfaces de la nube, estándares de interoperabilidad y estándares técnicos relacionados.
- **Composición cruzada de nube, colaboración e instrumentación de aplicaciones:** el concepto de componer aplicaciones de negocios distribuidas a través de las nubes, la instrumentación de procesos de negocio desde servicios alojados en diferentes nubes, tanto públicas como privadas, y de la integración de múltiples nubes híbridas, juntas en una fábrica de aplicaciones de negocios es nuevo. Habrá progreso en estos frentes como surjan estándares de la industria para hacer frente a estas necesidades potenciales.



## **2. FACTORES IMPORTANTES PREVIO A ESTUDIAR EL PROCESO DE AUTOMATIZACIÓN DE APLICACIONES Y ENTORNOS EN LA NUBE**

La nube refuerza algunos conceptos antiguos de la construcción de arquitecturas de internet de alta escalabilidad [2] e introduce algunos conceptos nuevos que cambian completamente la forma en que las aplicaciones son fabricadas y distribuidas. Por lo tanto cuando se avanza desde el concepto hasta la implementación, es posible que se tenga la sensación de que Todo ha cambiado, sin embargo, nada es diferente. La nube cambia varios procesos, patrones, prácticas, filosofías y refuerza algunos principios de arquitectura tradicional orientada a servicios que ha aprendido, ya que son aún más importantes que antes.

Las aplicaciones tradicionales fueron construidas con alguna mentalidad pre-concebida, que tenía sentido económico y de arquitectura en el momento que se desarrollaron.

La computación de nube es un conjunto de tecnologías maduras ya disponibles para ser utilizadas por personas y organizaciones de cualquier tamaño, en este capítulo y los siguientes se basarán en la plataforma de servicios de nube de *Amazon* denominados *Elastic Compute Cloud (EC2)*.

## 2.1. Prioridades de aplicaciones y negocios

La difícil economía hizo que la computación en nube fuera especialmente relevante. El negocio y el potencial financiero de las nubes, hace que sea una tendencia especial.

En primer lugar, con la computación en nube, se tiene otra vía para la realización de la agilidad del negocio. Al igual que con todas las tendencias de la tecnología, la agilidad del negocio es probablemente el objetivo más mencionado de los ejecutivos de tecnología cuando describen sus estrategias, y sin embargo, sigue siendo el menos tomado en cuenta en términos de ejecución.

La computación en nube ofrece la agilidad del negocio de una manera sencilla con un modelo fácilmente comprensible. Para un nuevo inicio o para los requisitos de negocios emergentes de empresas establecidas, la computación en nube permite a una organización implementar un modelo de tiempo rápido de salida al mercado accediendo de forma segura a un entorno de infraestructura TI listo para usar, alojado y administrado por un tercero de confianza, con el tamaño adecuado, computación escalable, capacidad de almacenamiento y de red, y que se paga solamente por la cantidad que se usa y como se usa.

No hay que construir o ampliar el centro de datos (sin construcción de edificios, piso elevado, equipo de energía y refrigeración, automatización de edificios, equipos de monitoreo y el personal); no hay que comprar ningún *hardware*, *software* o infraestructura de red (ni tratar con problemas de contratación a los que se están acostumbrados, especialmente con los inevitables retrasos en la adquisición de TI); se puede implementar

rápidamente un nuevo modelo de negocios o iniciar una nueva empresa para hacer frente a un nuevo mercado yendo más rápido de lo normal, y no hay que seguir pagando por la infraestructura y recursos de la nube si se suspende el proyecto o si la empresa falla. Desde una perspectiva del negocio y del ejecutivo de TI, ¿qué es lo que no gusta de esta viñeta de negocios?

Hay un sin número de nuevas empresas que inician, que han aprovechado los modelos de computación en nube para obtener su infraestructura de TI como un servicio, por lo tanto, les permite centrar sus limitados fondos y recursos de ancho de banda en su única tecnología y la innovación del modelo del negocio. Las limitaciones de recursos se liberan en este sentido, ya que fuerza a nuevas empresas a aprovechar los recursos de la nube listos para usar, en lugar de construir un centro de datos.

Este tipo de escenarios, por supuesto, plantea una serie de negocios y las consecuencias financieras deben estudiarse más a fondo.

#### Convertir costos fijos en costos variables

En primer lugar, la computación en nube ofrece a un ejecutivo de negocios la oportunidad de, convertir lo que tradicionalmente han sido importantes costos fijos de operación y poseer un centro de datos, en un costo variable pagando solamente por el volumen de recursos de TI que son actualmente utilizados.

Los costos de los centros de datos se pagan por adelantado, son capital desde una perspectiva contable, donde los activos físicos se deprecian durante su vida útil. Por lo tanto, los costos de los centros de datos son fijos y los

gastos que se pagan mensualmente serán relativamente fijos comparado con el volumen de negocios.

Los costos fijos son los gastos que se mantienen relativamente constantes independientemente del nivel de ventas. Por ejemplo, el costo de alquiler de una sede de una empresa, es probable que sea una cantidad constante, por ejemplo \$10 000,00 por mes, independientemente de la cantidad de ingresos que la empresa genere.

Los centros de datos, los recursos de computación, equipos de administración de energía y refrigeración, el apoyo a la automatización de edificios y equipos de seguridad física que figuran en él, se consideran costos fijos, se consideran gastos de capital en términos de contabilidad y se amortizan a lo largo de su vida útil por los principios contables generalmente aceptados. Para los efectos contables trimestrales o anuales, el gasto mensual para centros de datos será un costo fijo, o la misma cantidad de gastos, independientemente de la eficacia con la que se contribuye al volumen de ingresos. Por lo tanto, si las ventas han bajado, todavía se tiene que pagar los mismos costos fijos. Si las ventas han subido, se mantienen las mismas obligaciones de gastos fijos.

Los costos variables, sin embargo, varían con la cantidad de producción o las ventas que se generan. Ejemplos comunes de los costos variables incluyen materia prima, envases y mano de obra directamente implicada en el proceso de fabricación de una compañía. Estos costos varían según la cantidad de la producción y/o volumen de ventas que genera una empresa. Más ventas, más costos variables, pero están alineados con las ventas y el volumen de producción. Menos ventas o producción, menores serán los costos variables.

Los modelos de computación en nube, basado en la modalidad de pago según el consumo, ofrecido a través de los beneficios de computación pública, significa que los gastos asociados con los recursos de la nube, por ejemplo, la infraestructura de TI, plataformas como servicio (*Platform as a Service, PaaS*), *software* como servicio (*Software as a Service, SaaS*), varía directamente con la producción o volumen de ventas y se puede agregar o reducir la capacidad basado en el volumen de ventas o producción. Así los gastos de computación en nube para un consumidor de nube son costos variables en lugar de costos fijos.

Desde una perspectiva de flujo de caja y financiera, convertir los costos fijos en costos variables es mucho mejor para la empresa. La computación en nube es especialmente atractiva ya que permite al costo fijo hacer una conversión y ser un costo variable.

#### Agilidad de negocios y comercialización más rápida

La computación en nube ofrece una nueva vía para la agilidad del negocio y el tiempo más rápido al mercado, ofreciendo recursos listos para consumir habilitados en la nube, así como infraestructura de TI, plataformas de *software* y aplicaciones de negocio que se pueden acceder y operar en apoyo a un requisito de un nuevo negocio, mucho más rápido que adquiriendo, instalando, configurando y operando estos recursos de TI por cuenta propia.

Para el ejecutivo de negocios, nube significa la capacidad de obtener rápidamente un nuevo modelo de negocio de comercialización sin los retrasos típicos de adquisiciones de TI, infraestructura de ingeniería y las necesidades de gestión de configuración, configuración de *software* y requisitos de mantenimiento. En cambio, la empresa define su computación,

almacenamiento, red, plataforma y los requerimientos de recursos de la aplicación; va a un tercero proveedor de nube; y obtiene los recursos necesarios con una transacción de tarjeta de crédito. Los recursos estarán disponibles en cuestión de horas, en comparación con semanas y meses.

Hay muchos escenarios en la nube donde la nube tiene mucho sentido para un ejecutivo de negocios. A menudo los nuevos conceptos de modelo de negocio, son probados en forma limitada con el fin de sacar las torceduras y matices de un nuevo modelo de negocios, o aspectos limitados de una nueva estrategia de negocios. Muchas veces, la velocidad de comercialización es un importante motor de la innovación del modelo del nuevo negocio, y junto con eso viene la seguridad y protección de propiedad intelectual.

Cuanto más tiempo se espere para poner en marcha los secretos comerciales más críticos, pueden caer en manos de la competencia. Para un ejecutivo de negocios, la nube es un medio para llevar a cabo experimentos del modelo del negocio con un alcance limitado para poner a prueba un nuevo servicio o producto, de forma rápida y segura, sin tener que realizar la adquisición normal de *hardware*, *software* e infraestructura de red. A medida que avanza la innovación del modelo del negocio, y si tiene éxito, la nube ofrece un modelo escalable bajo demanda para añadir nueva capacidad con la precisión que se necesita, ni más ni menos.

Si el piloto de modelo de negocio se cancela, la capacidad de la nube es abandonada, y se deja de pagar de inmediato por los recursos de la nube. Si se ha adquirido la infraestructura de TI para apoyar la construcción del modelo de negocio nuevo, todavía se tendría que pagar por ella, y todavía se tendría la capacidad, probablemente la capacidad no usada, en el futuro. El exceso de capacidad es un desperdicio, tanto en términos de dólares, computación física,



condiciones de almacenamiento, mano de obra y condiciones de recursos humanos de TI.

La agilidad del negocio y el valor de la rápida comercialización de la nube, es particularmente atractivo como un medio para responder rápidamente a nuevos mercados, para innovar y probar nuevos conceptos del modelo de negocio rápidamente y ofrecer un modelo nuevo de inicio rápido para ir al mercado sin costos por adelantado y las demoras en la adquisición y operación de infraestructura de TI. Todas estas son razones por las que a un ejecutivo de negocios debería importarle la nube.

## 2.2. Presupuesto

Dependiendo de la capacidad informática que demande y otro tipo necesidades propias de la empresa, ésta puede planificar el presupuesto para la compra y mantenimiento del *hardware* tradicional, lo que puede provocar normalmente un gran costo fijo para la empresa.

A continuación se presenta un cuadro con una inversión inicial aproximada que tendría que hacer la empresa, suponiendo que se desea implementar una aplicación *web* y correo electrónico.

Tabla I. **Inversión inicial aproximada para compra de equipo de una empresa**

4 servidores: base de datos, aplicaciones, servidor web y correo electrónico.	US \$20,000.00
Software para el equipo	US \$2,000.00
Configuración del equipo	US \$2,000.00
Instalación y configuración de red	US \$1,500.00
Equipo adicional (Ej. enrutadores, regletas, ups, etc.)	US \$1,000.00
<b>TOTAL</b>	<b>US \$26,000.00</b>

Fuente: elaboración propia.

Adicionalmente a esto, hay que sumar los costos de salarios de empleados que administrarán el equipo, servicios de energía eléctrica, mantenimiento preventivo, etc., lo que hace que las pequeñas y medianas empresas tengan que hacer una inversión inicial muy elevada en tecnología con riesgos muy grandes.

Por otro lado, si se considera un servicio de computación en nube o servidor *cloud*, que es una máquina virtual configurada a la medida de las necesidades de la empresa, puede resultar que la misma se libere de las complejidades de la planificación, la compra y el mantenimiento del *hardware*, tener el mínimo soporte y costos de administración y puede transformar los grandes costos fijos en costos variables mucho más pequeños.

Los costos pueden variar dependiendo del sistema operativo que tengan las máquinas virtuales que se van a ejecutar, bases de datos configuradas, procesador, capacidad de almacenamiento, tiempo de ejecución de cada máquina virtual en la nube, etc. Las diferentes empresas que brindan este servicio, proporcionan tablas con las diferentes tarifas por hora de uso de instancia de máquina virtual consumida, con lo cual se pueden hacer los cálculos respectivos de los costos promedios mensuales que puede tener la compañía al adquirir estos servicios en la nube.

Existen diferentes servicios que se ofrecen tales como el uso de instancias de máquinas virtuales bajo demanda, lo que hace que la empresa no tenga ningún tipo de compromiso, solamente se paga por la hora consumida de la instancia de la máquina virtual.

Existe el concepto de instancias reservadas, que consiste en la opción de realizar un pago puntual reducido por cada instancia que desee reservar y a cambio recibir un descuento importante en el cargo de uso por horas de dicha instancia, pueden reservarse las instancias por periodos, por ejemplo de 1 hasta 3 años.

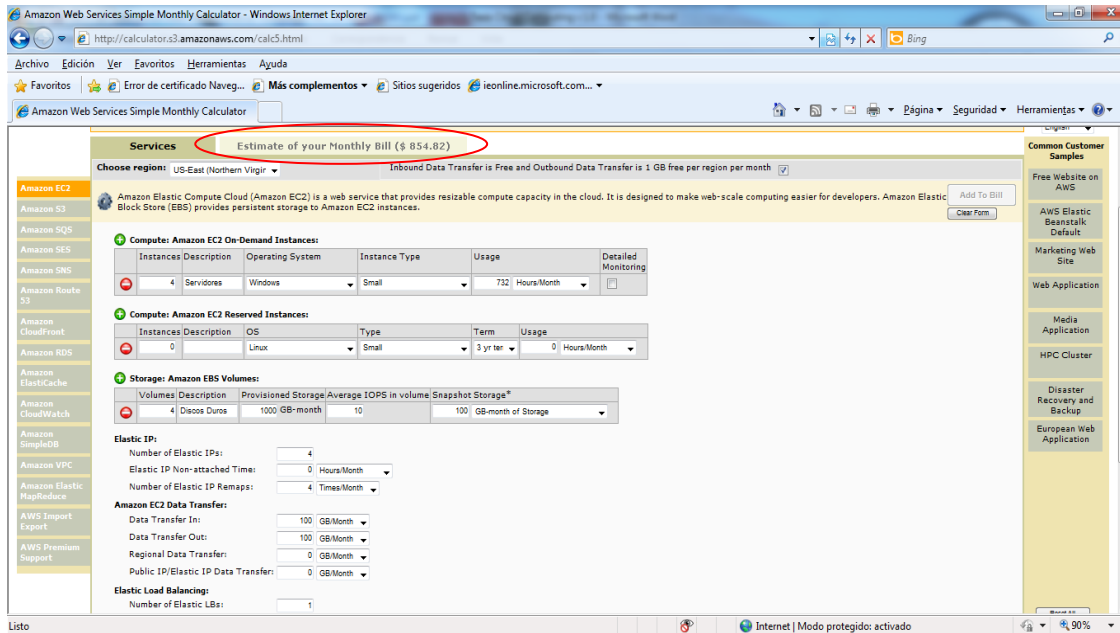
Adicionalmente a los servicios descritos anteriormente, también existen servicios de transferencia de datos tanto entrantes como salientes, con precios por rango de transferencia de datos, medidos en GB/mes o TB/mes, lo cual también ayuda a definir el presupuesto mensual que se adapte a las necesidades de la empresa.

Otro servicio importante, es el que se refiere al almacenamiento que puede suministrar dicha infraestructura, midiendo las tarifas en GB/mes de datos almacenados, así como las solicitudes de entrada y salida para poner y obtener información respectivamente.

Servicios adicionales tales como asignación de IP dinámicas, monitores detallados para ver el rendimiento de instancias, el servicio de balanceo de cargas de trabajo, tienen definida una tarificación, por lo cual es importante tomarlos en cuenta para la planificación del presupuesto.

La siguiente pantalla es un ejemplo de presupuesto mensual estimado en Amazon (\$ 854,82), de cuatro instancias tipo *Small Instance*, cuatro volúmenes de 1 000 GB cada uno, cuatro *Elastic IP* y 100 GB de transferencia de entrada y salida entre las instancias que tenga una cuenta determinada; 1 000 mensajes de email por día y transferencia de datos de entrada y salida en los correos (datos adjuntos) de 1 000 GB por mes.

Figura 2. Presupuesto mensual de Instancias en Amazon EC2



Fuente: elaboración propia.

Si se multiplica por 12 meses el valor presupuestado resulta en US \$ 10 257,84 al año, asumiendo que los precios se mantienen constantes durante ese período de tiempo. Esto es más atractivo para una pequeña o mediana empresa en el sentido que, no tiene que hacer una inversión inicial muy grande, sino que tendrá un costo mensual variable y mucho más reducido.

Para calcular de una mejor manera el presupuesto mensual que se utilizaría con los servicios de Amazon, también se puede incluir almacenamiento adicional para desarrollo de aplicaciones con el servicio Simple Storage Service (S3), incluir más capacidad en el servicio de correo electrónico Amazon Simple Email Service (SES), que es altamente escalable y rentable; el servicio de Amazon Relational Database Service (Amazon RDS), que es un servicio web que facilita las tareas de configuración, utilización

y escalado de una base de datos relacional en la nube; incluir también el servicio de *Amazon Simple DB*, que es un almacén de datos no relacionales altamente disponible, ampliable y flexible, que alivia carga de trabajo de la administración de bases de datos.

En el capítulo 3.3 se describen varios de los servicios más importantes que proporciona Amazon para la nube.

### **2.3. Infraestructura y redes**

En la actualidad, el 70 por ciento de las inversiones en TI se centran en el mantenimiento, con lo cual queda poco tiempo para dedicar a otros proyectos comerciales estratégicos. Las infraestructuras de TI se han vuelto demasiado frágiles y complejas. Dado que los usuarios exigen tiempos de respuesta más rápidos, y el departamento de administración reclama costos más bajos, el área de TI necesita una mejor estrategia. La computación en nube es un nuevo enfoque que reduce la complejidad de TI aprovechando la eficiente creación de depósitos de recursos compartidos de infraestructura virtual auto administrada según demanda, consumidos como un servicio. La computación en nube es fundamental para lograr esa mejor estrategia.

Ya no son necesarios los dispositivos de red físicos: no es posible implementar dispositivos de red físicos en *Amazon Web Services (AWS)*. Por ejemplo, los cortafuegos, enrutadores y equilibradores de cargas de las aplicaciones de AWS ya no pueden residir en dispositivos físicos, sino que es necesario sustituirlos con soluciones de *software*. Existe una amplia gama de soluciones de calidad empresarial para estos problemas, ya se trate de equilibrar cargas (por ejemplo, *Zeus*, *HAProxy*, *Nginx*, *Pound*), o de establecer una conexión de red privada virtual VPN (por ejemplo, *OpenVPN*, *OpenSwan*,

*Vyatta*). No hay limitaciones respecto a qué es posible ejecutar en la nube de AWS, pero representará un cambio de la arquitectura de su aplicación si utiliza estos dispositivos en la actualidad.

## **2.4. Consideraciones de seguridad**

En un entorno multiusuario, los arquitectos de la nube, a menudo expresan preocupaciones sobre seguridad. La seguridad debe aplicarse en todas las capas de la arquitectura de aplicaciones de la nube. La seguridad física se maneja normalmente por el proveedor de servicios, pero la red y la seguridad a nivel de aplicación es responsabilidad de la aplicación y se deben implementar las mejores prácticas aplicables al negocio. En este capítulo se muestran algunas herramientas específicas que permiten asegurar las aplicaciones en el entorno de nube.

Proteger los datos en tránsito: si se necesita intercambiar información sensible o confidencial entre un navegador y un servidor web, puede configurar *Security Socket Layer (SSL)*, en la instancia del servidor. Se necesitará un certificado de una entidad de certificación externa como *VeriSign* o *En trust*. La clave pública incluida en el certificado autentica el servidor hacia el navegador y sirve como base para la creación de la clave de sesión compartida utilizada para cifrar o encriptar los datos en ambas direcciones.

Crear una nube privada permite utilizar sus propios recursos lógicamente aislados dentro de la nube y luego conectar esos recursos directamente al propio centro de datos usando cifrado estándar IPsec, con conexiones de una red privada virtual (VPN, del inglés *virtual private network*). También se puede configurar un servidor *OpenVPN* en una instancia de Amazon EC2 e instalar el cliente *OpenVPN* en todos los PCs de los usuarios.

Proteger los datos en reposo: si se necesita almacenar datos sensibles y confidenciales en la nube, se deben cifrar los datos (archivos individuales), antes de subirlos a la nube. Por ejemplo, cifrar los datos utilizando cualquier herramienta de código abierto o comercial antes de almacenarlos como objetos de la nube y descifrarlos después de la descarga. Esto a menudo es una buena práctica en la construcción de aplicaciones de cualquier tipo.

En la nube, el cifrado de archivos depende del sistema operativo. Las instancias que se ejecutan en *Windows* pueden utilizar la característica del sistema integrado de archivos de cifrado (*Encrypting File System, EFS*). Esta función se encargará de la codificación y decodificación de archivos y carpetas de forma automática y hace que el proceso sea transparente para los usuarios.

Sin embargo, a pesar de su nombre, EFS no cifra el sistema de archivos, sino que cifra los archivos individuales. Si necesita un cifrado completo, puede utilizar el producto de código abierto llamado *TrueCrypt* que se integra muy bien con volúmenes en la nube con formato NTFS. Instancias corriendo Linux, pueden montar volúmenes usando sistemas de archivos encriptados utilizando distintos métodos (*EncFS, Loop-AES, dm-crypt, TrueCrypt*). Del mismo modo, instancias corriendo *Open Solaris*, pueden aprovechar el soporte de cifrado ZFS. Independientemente del método que se utilice, la encriptación de archivos y volumen es ayuda a proteger archivos y datos de registro de manera que solo los usuarios y los procesos en el servidor pueden ver los datos en texto claro, pero nada ni nadie fuera del servidor verá solo los datos cifrados.

No importa qué sistema operativo o tecnología se elija, el cifrado de datos en reposo representa un reto: la administración de las claves utilizadas para cifrar los datos. Si se pierden las claves, se perderán los datos para siempre y

si las claves se ven comprometidas, los datos pueden estar en riesgo. Por lo tanto, hay que asegurarse de estudiar las capacidades de administración de claves de cualquier producto que se elija, y hay que establecer un procedimiento que reduzca al mínimo el riesgo de perder las claves.

Además de proteger los datos de espionaje, también se debe considerar la forma de protegerse de los desastres, haciendo copias instantáneas periódicas de volúmenes, para asegurar que es altamente durable y disponible. Las copias instantáneas son de naturaleza gradual y son almacenadas en la nube, regularmente en un área geográfica separada, y se pueden restaurar de nuevo con unos pocos clics o llamadas de línea de comandos.

Proteger las credenciales: AWS ofrece dos tipos de credenciales de seguridad: las claves de acceso de AWS y los certificados X.509. La clave de acceso de AWS tiene dos partes: el ID de la clave de acceso y la clave de acceso secreta.

Si la clave de acceso secreta se ve comprometida, se debe obtener una nueva clave y cambiarse a la nueva clave de acceso.

Como una buena práctica, se recomienda que se incorpore un mecanismo de rotación de clave dentro de la arquitectura de la aplicación, de manera que se pueda usar en forma regular u ocasionalmente (cuando los empleados descontentos salen de la empresa), para asegurar que las claves comprometidas no pueden durar para siempre.

Alternativamente se pueden utilizar los certificados X.509 para la autenticación en determinados servicios. El archivo de certificado contiene la



clave pública en un órgano de certificación codificado en base 64. Un archivo separado contiene la correspondiente clave privada codificada en base 64.

Manejar varios usuarios y sus permisos: el servicio de Amazon, Identidad y Gestión de Acceso (*Identity and Access Management, IAM*), permite crear varios usuarios y gestionar los permisos para cada uno de estos usuarios dentro de la cuenta de AWS. Un usuario es una identidad (dentro de la cuenta AWS), con credenciales de seguridad únicas que se pueden utilizar para acceder a los servicios. IAM elimina la necesidad de compartir las contraseñas o claves de acceso y hace que sea fácil habilitar o deshabilitar el acceso del usuario según corresponda.

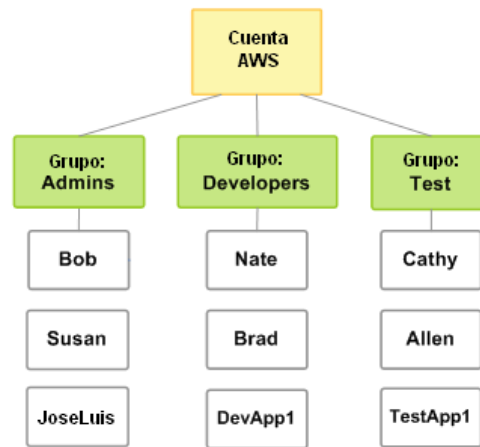
IAM permite implementar las mejores prácticas de seguridad, mediante el otorgamiento de credenciales únicas para cada usuario en una cuenta AWS, y solo concede permiso para acceder a los servicios y los recursos necesarios para los usuarios, para realizar su trabajo. IAM es seguro de forma predeterminada, los usuarios nuevos no tienen acceso a los servicios hasta que los permisos se autoricen de forma expresa.

Se debe minimizar el uso de las credenciales de una cuenta tanto como sea posible, en la interacción con sus servicios y aprovechar las credenciales de usuario de IAM, para acceder a los servicios y recursos.

La siguiente figura muestra un ejemplo sencillo de una cuenta AWS con tres grupos. Un grupo es un conjunto de usuarios que tienen responsabilidades similares. En este ejemplo, hay un grupo de administradores (llamado *Admins*). También hay un grupo de desarrolladores (*Developers*), y un grupo de prueba (*Test*). Cada grupo tiene varios usuarios y cada usuario puede estar en más de

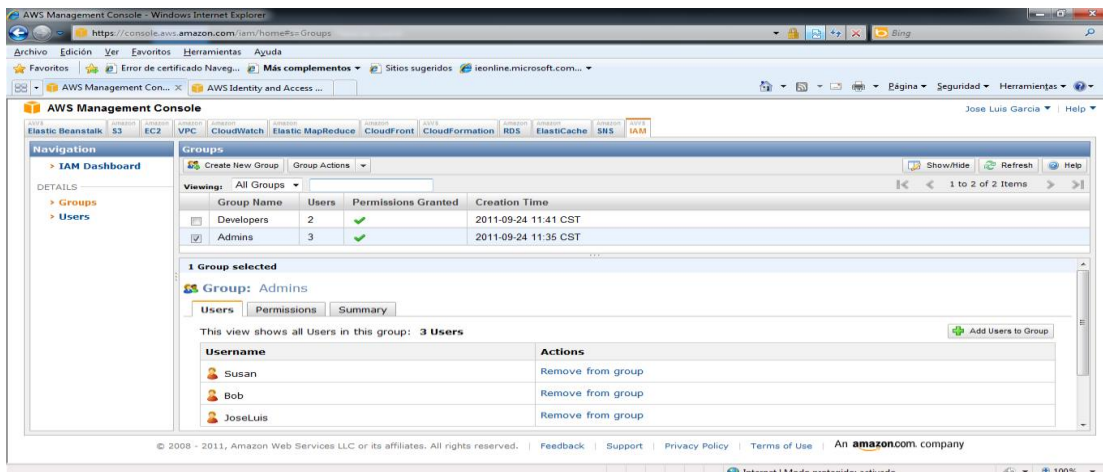
un grupo, aunque la figura no ilustra esto. No se pueden poner grupos dentro de otros grupos.

Figura 3. Cuenta con grupos de usuarios de Amazon



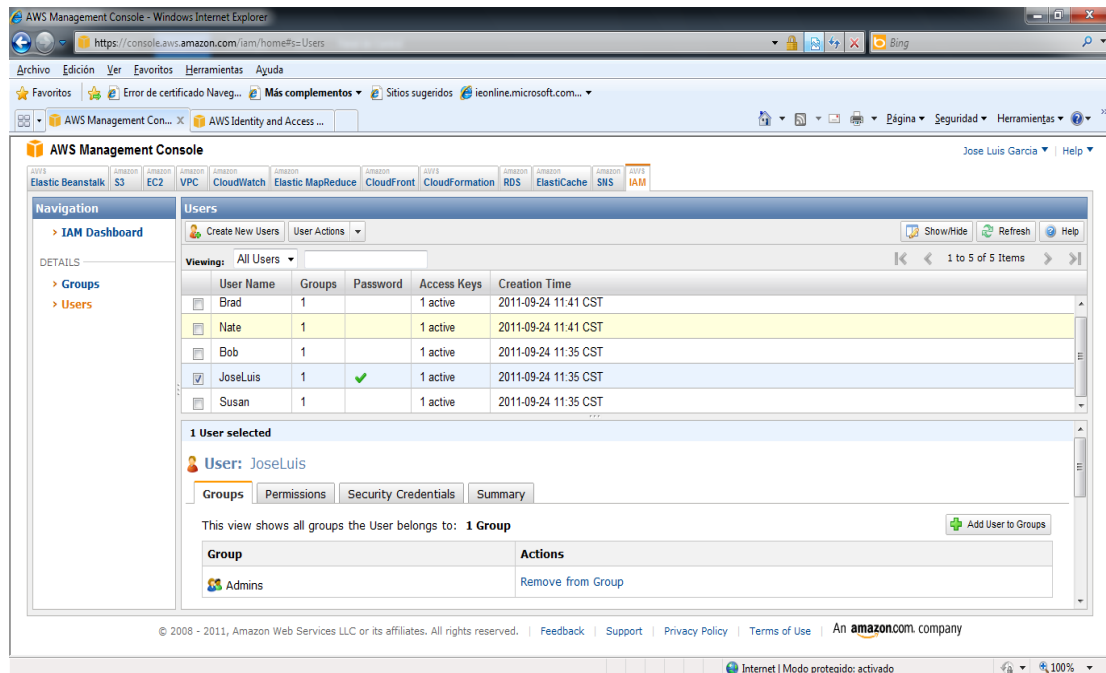
Fuente: elaboración propia.

Figura 4. Muestra los grupos y los usuarios en consola de Amazon IAM



Fuente: elaboración propia.

Figura 5. Muestra los usuarios creados en la consola de Amazon IAM

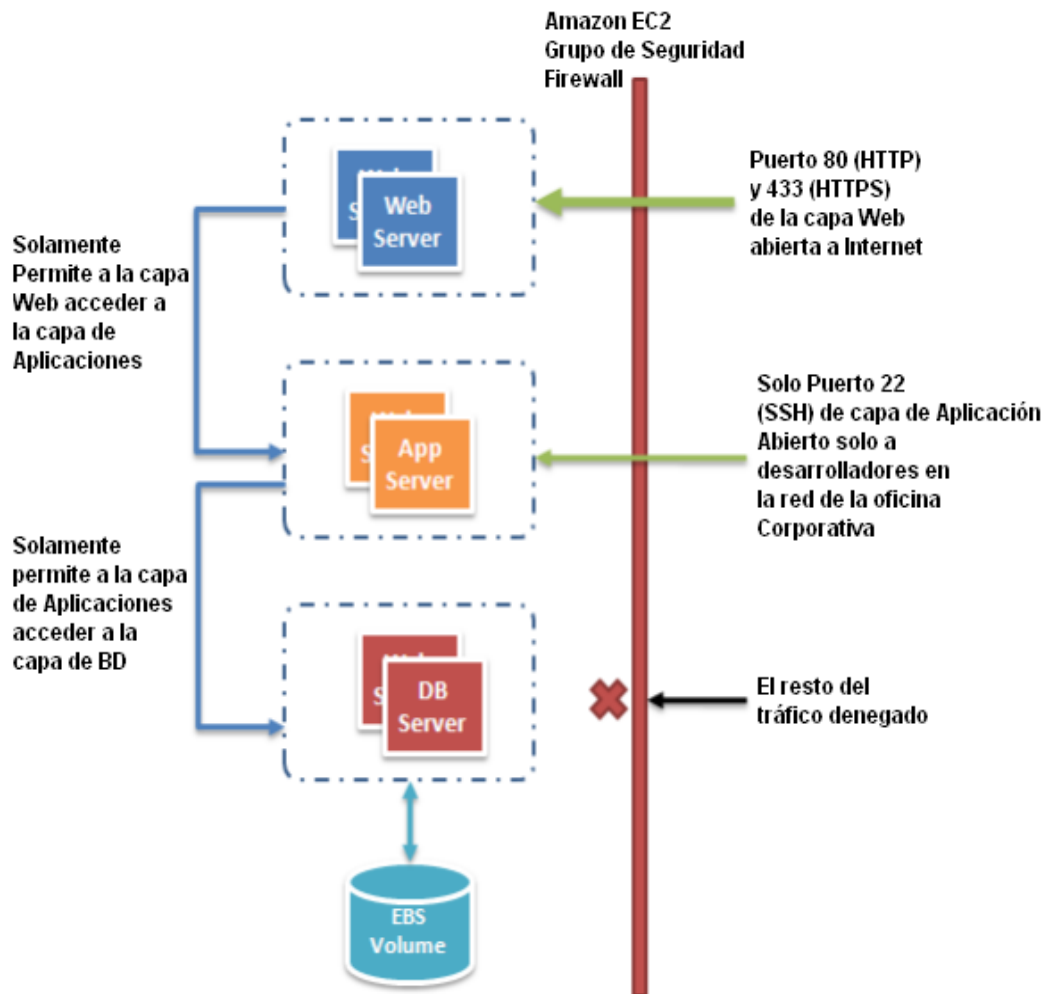


Fuente: elaboración propia.

Asegurar la aplicación: cada instancia de *Amazon EC2* está protegida por uno o más grupos de seguridad, llamados grupos de reglas, que especifican qué entrada de tráfico de red debe ser entregada a la instancia.

Se pueden especificar los puertos TCP y UDP, y los tipos y códigos ICMP y las direcciones de origen. Los grupos de seguridad dan seguridad básica como un *firewall* para instancias en ejecución. Por ejemplo, las instancias que pertenecen a una aplicación *web* pueden tener la siguiente configuración de grupo de seguridad, tal como se muestra en la siguiente figura:

Figura 6. Grupo de seguridad de Amazon EC2 para aplicaciones web



Fuente: elaboración propia.

Otra forma de restringir el tráfico entrante es configurar *firewalls* basados en *software* en las instancias. Las instancias de *Windows* pueden utilizar el *firewall* incorporado, las instancias de *Linux* pueden utilizar *NetFilter* e *IPTables*.

Con el tiempo, los errores de *software* se descubren y se requieren parches para repararlos. A continuación se verán algunas sugerencias básicas para mejorar la seguridad de las aplicaciones:

- Regularmente descargar los parches desde el sitio *web* del proveedor y actualizar las AMIs.
- Volver a implementar o desplegar las instancias de las nuevas AMIs y probar sus aplicaciones para garantizar que los parches no dañaron nada. Hay que asegurarse de que el último AMI se ha implementado en todas las instancias.
- Invertir en scripts de prueba para poder ejecutar los controles de seguridad periódicamente y automatizar el proceso.
- Asegurarse de que el *software* de terceros tiene la configuración de la forma más segura.
- Nunca correr los procesos como usuario raíz (*root*), o administrador a menos que sea absolutamente necesario.

La nube abstrae la complejidad de la seguridad física, y proporciona el control a través de herramientas y funciones, para que se puedan asegurar las aplicaciones.



### **3. PROCESO DE ADMINISTRACIÓN Y AUTOMATIZACIÓN EN LA CONSTRUCCIÓN DE APLICACIONES Y ENTORNO DE APLICACIONES EN LA NUBE**

#### **3.1. Virtualización para la administración y automatización de aplicaciones y sus entornos en la nube**

Administración virtual: la llegada de la nube ha cambiado el papel de administrador del sistema a un Administrador del Sistema Virtual. Esto simplemente significa que las tareas diarias realizadas por estos administradores se han convertido aún más interesantes a medida que aprenden más acerca de las aplicaciones y deciden qué es lo mejor para el negocio como un todo. El administrador del sistema ya no tiene la necesidad de aprovisionar servidores e instalar el *software* y el cable de los dispositivos de la red, ya que todo el trabajo duro se sustituye por unos pocos clics y llamadas a la línea de comando. La nube anima a la automatización, ya que la infraestructura es programable. Los administradores de sistemas necesitan ascender en la pila de la tecnología y aprender a manejar los recursos abstractos de la nube mediante *scripts*.

Del mismo modo, el papel de administrador de bases de datos se transforma en un Administrador de Bases de Datos Virtual, en la que maneja los recursos a través de una consola basada en *web*, ejecuta *scripts* que añaden nuevas capacidades programáticamente en caso de que el *hardware* de la base de datos se quede sin capacidad, y automatiza los procesos del día a día. El DBA virtual tiene que aprender ahora nuevos métodos de despliegue (imágenes de máquinas virtuales), adoptar nuevos modelos (consulta en

paralelo, georedundancia y replicación asincrónica), repensar el enfoque de la arquitectura de datos (*sharding* [1], el particionamiento horizontal [2], la federación [3]), y las diferentes opciones disponibles en la nube para los diferentes tipos de grupos de datos.

En la empresa tradicional, los desarrolladores de aplicaciones pueden no funcionar en estrecha colaboración con los administradores de red y los administradores de red pueden no tener una pista acerca de la aplicación. Como resultado, varias posibles optimizaciones en la capa de red y la capa de arquitectura de la aplicación, son pasadas por alto. Con la nube, los dos roles se han fusionado en uno hasta cierto punto. Al diseñar las futuras aplicaciones, las empresas deben fomentar más la polinización cruzada de conocimientos entre los dos roles, y entender que se están fusionando.

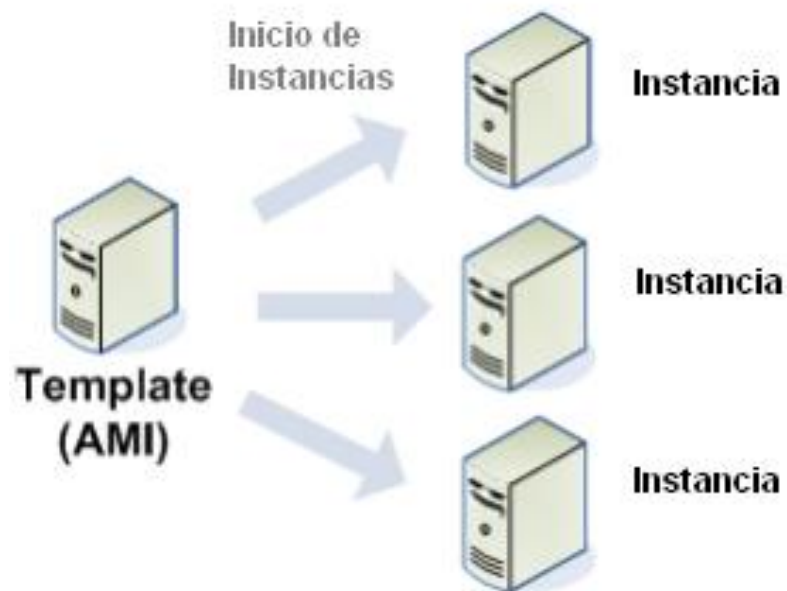
### **3.2. Servicios estándar en la virtualización para eliminar la constante reconfiguración estática de infraestructura para un entorno informático dinámico**

Para poder evitar o eliminar la constante reconfiguración estática de infraestructura, se mencionan varios servicios y componentes básicos de infraestructura que se toman en cuenta para lograr obtener un entorno informático dinámico.

El servicio de Amazon EC2, permite la creación de AMIs (*Amazon Machine Image*), que son básicamente plantillas que contienen configuración de *software* (Ejemplo, sistema operativo, servidor de aplicaciones y aplicaciones). Desde una AMI, se pueden iniciar instancias que son copias de la AMI. Se pueden iniciar múltiples instancias de una AMI tal como se muestra en la siguiente figura.



Figura 7. **Múltiples instancias de una AMI**

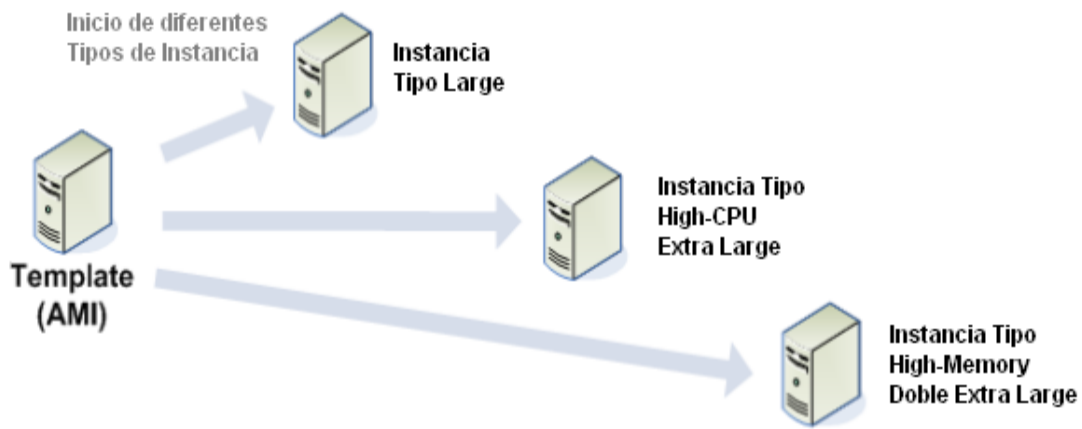


Fuente: elaboración propia.

Las instancias se mantienen corriendo hasta que el administrador las detiene o hasta que ellas fallan, pero si una instancia falla, se puede iniciar una nueva a partir de la AMI.

Se puede usar una única AMI o múltiples AMIs dependiendo de las necesidades que se tengan. Desde una única AMI, se pueden iniciar diferentes tipos de instancias. Un tipo de instancia es esencialmente un prototipo de hardware, tal como se ilustra en la siguiente figura, la cual ilustra cómo se selecciona un tipo de instancia particular, basado en la cantidad de memoria y potencia de computación que necesita, para la aplicación o *software* que vaya a ejecutarse en la instancia.

Figura 8. Inicio de diferentes tipos de instancia



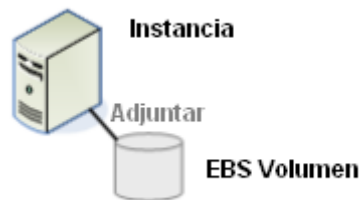
Fuente: elaboración propia.

Amazon publica muchas AMIs que contienen configuraciones comunes de *software* para uso público. Es posible que solo se necesiten usar AMIs que Amazon u otras fuentes confiables ofrecen, solo se tienen que personalizar las instancias que se necesiten, por ejemplo, ejecutar un *script* para proporcionar los datos o el *software* que se necesita cada vez que se inicie una instancia. También se pueden crear AMIs propias o AMIs personalizadas, y posteriormente se puede ejecutar la aplicación con el inicio de una de las AMIs personalizadas.

Por ejemplo, si la aplicación es un sitio web o servicio *web*, la AMI puede ser pre configurada con un servidor *web*, el contenido estático asociado, y el código para todas las páginas dinámicas. Como alternativa, se puede configurar la AMI para instalar todos los componentes de software y contenido, mediante la ejecución de un script de arranque, tan pronto como se inicia la instancia. Como resultado, después del inicio de la AMI, el servidor web se iniciará y la aplicación puede comenzar a aceptar solicitudes.

Otros componentes importantes son los servicios *Amazon EBS Volúmenes*, que proporcionan a las instancias almacenamiento a nivel de bloque con persistencia, es decir, esencialmente discos duros que se pueden adjuntar a una instancia en ejecución.

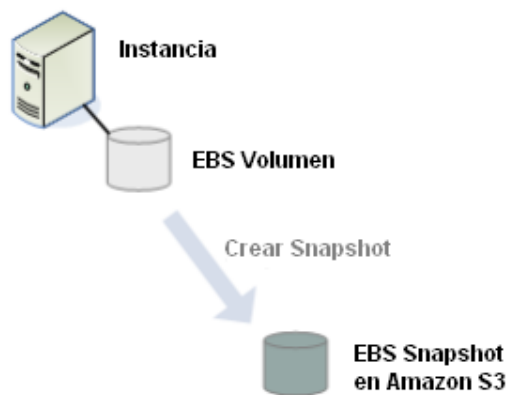
Figura 9. **Volúmenes de instancia**



Fuente: elaboración propia.

Los volúmenes son especialmente adecuados para aplicaciones que requieren una base de datos, un sistema de archivos, etc. Se pueden adjuntar varios volúmenes a una instancia. Para conservar una copia de respaldo, se puede crear una instantánea del volumen como se muestra en la figura a continuación. La instantánea se almacena en *Amazon S3*.

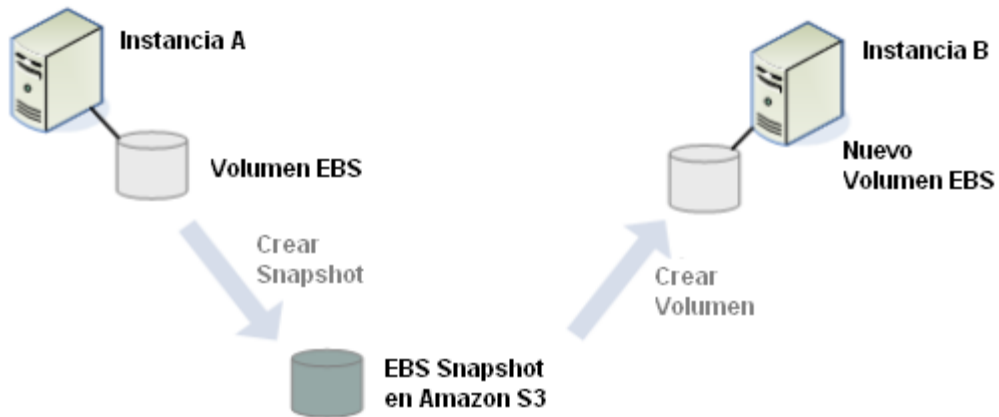
Figura 10. **Instantánea de un volumen**



Fuente: elaboración propia.

Puede crearse un nuevo volumen de *Amazon EBS* a partir de una instantánea, y adjuntarlo a otra instancia como se muestra a continuación.

Figura 11. **Adjuntar un volumen nuevo a otra instancia**



Fuente: elaboración propia.

También puede separar un volumen de una instancia y adjuntarlo a otra diferente como se muestra a continuación.

Figura 12. **Separar un volumen de una instancia**



Fuente: elaboración propia.

### 3.3. Algunas herramientas para la administración de máquinas virtuales e infraestructura virtual

A continuación se dará una breve descripción de las características y funcionamiento de algunas herramientas o servicios utilizados por *Amazon Web Services* (AWS), para la administración del entorno virtual.

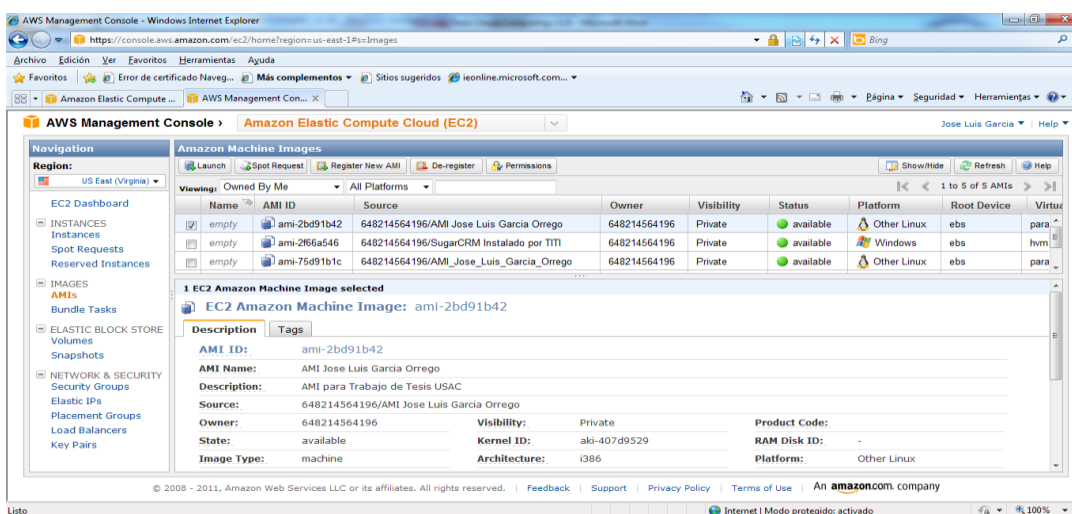
- *Amazon Elastic Compute Cloud* (EC2): es un servicio *web* que permite cambiar el tamaño de la capacidad de cómputo en la nube. Se puede incluir el sistema operativo, el *software* de aplicaciones y los ajustes de configuraciones asociadas a una imagen de máquina de Amazon (*Amazon Machine Image*, AMI). Posteriormente, se pueden utilizar estas Amis para el uso de múltiples instancias virtuales, así como retirar dichas instancias, utilizando llamadas simples de los servicios web para subir la capacidad y bajarla rápidamente a medida que cambian los requisitos informáticos.

Las instancias pueden ser iniciadas en una o más regiones geográficas; cada región tiene varias zonas de disponibilidad, las zonas de disponibilidad son lugares distintos que se han diseñado para aislarse de las fallas de otras zonas de disponibilidad, proporcionando a un bajo costo, conectividad de baja latencia en la red, hacia otras zonas de disponibilidad en la misma región. Amazon EC2, proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores, y les permite aislarse de los casos de error más comunes.

Amazon EC2, presenta un auténtico entorno informático virtual, que permite utilizar interfaces de servicio web, para iniciar instancias con distintos sistemas operativos, cargarlas con un entorno de aplicaciones personalizadas, gestionar los permisos de acceso a la red y ejecutar la imagen, utilizando los sistemas que desee. Permite aumentar o reducir la capacidad en cuestión de minutos, sin esperar horas ni días. Puede enviar una, cientos o incluso miles de instancias del servidor simultáneamente. Tiene la opción de varios tipos de instancias, sistemas operativos y paquetes de software, además permite seleccionar una configuración de memoria, CPU, almacenamiento de instancias y el tamaño de la partición de arranque óptimo para el sistema operativo y la aplicación.

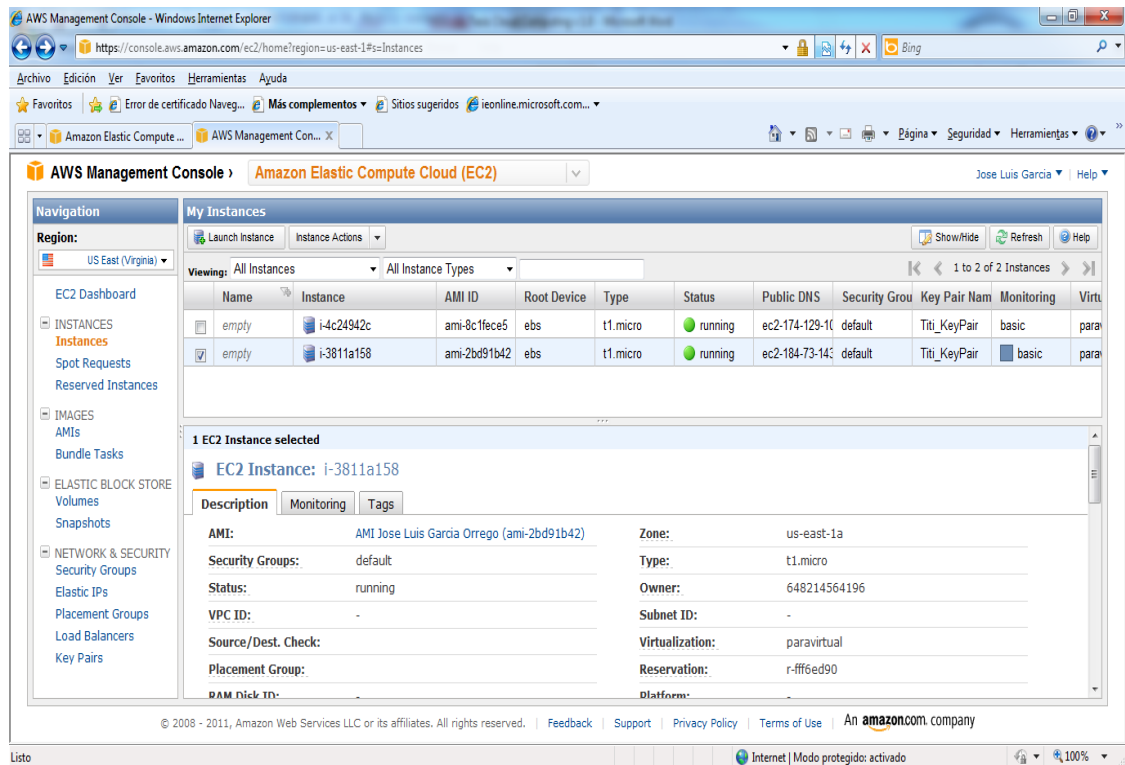
A continuación se presentan dos pantallas que contienen la consola de AWS, con la información de varias instancias virtuales corriendo en la nube.

Figura 13. **Consola de AWS desplegando AMIs con el servicio de EC2**



Fuente: elaboración propia.

Figura 14. Consola AWS desplegando Instancias de AMIs con el servicio de EC2



Fuente: elaboración propia.

- **Elastic IP Address:** permite asignar una dirección IP estática, y programáticamente asignarla a una instancia. Son direcciones IP estáticas, diseñadas para la informática dinámica en nube. Una dirección *Elastic IP*, está asociada a una cuenta, no a una instancia concreta, y se puede controlar ésta dirección hasta que se decida, explícitamente liberarla. Al contrario de las direcciones IP estáticas tradicionales, las *Elastic IP Addresses*, permiten disimular los errores en instancias o zonas de disponibilidad, al reasignar de forma programada sus direcciones IP públicas a cualquier instancia de la cuenta.

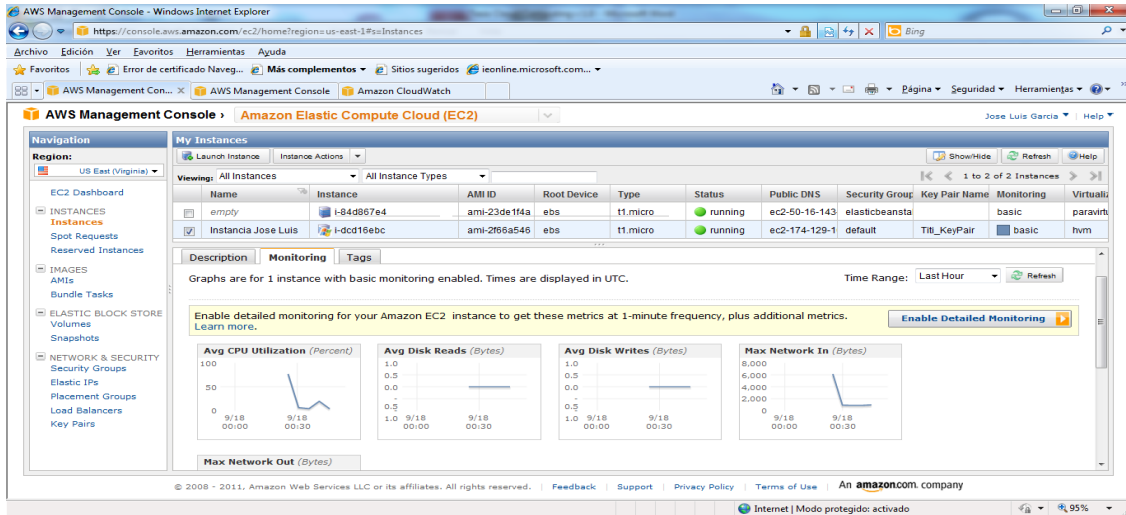
En lugar de esperar a que un técnico de datos reconfigure o reemplace el host, o bien esperar a que el DNS se propague a todos los clientes, *AmazonEC2*, permite solucionar los problemas con la instancia o el *software*, mediante la reasignación rápida de la dirección *Elastic IP*, a una instancia de sustitución. Estas direcciones son muy importantes cuando se inician instancias en nubes privadas virtuales, ya que para poder acceder a dichas instancias a través de *Internet*, se hace por medio de la dirección IP elástica previamente definida y asociada a la instancia en la nube privada.

- *Amazon CloudWatch*: es un servicio que permite habilitar el monitoreo sobre una instancia de *Amazon EC2*, con el fin de ganar visibilidad en la utilización de recursos, el rendimiento operativo, y los patrones generales de la demanda (incluyendo métricas como el uso del CPU, lecturas y escrituras de disco y el tráfico de red), además permite supervisar las instancias de *Amazon EC2*, volúmenes de *Amazon EBS*, *Elastic Load Balancers* y las instancias de bases de datos RDS en tiempo real. Este servicio habilita el servicio *Auto Scaling*, que le permite añadir o eliminar de forma dinámica instancias de *Amazon EC2*, según los indicadores.

A continuación se presentan 2 pantallas que contienen las gráficas con información sobre la utilización del CPU, disco y red de una instancia corriendo en la nube de *Amazon EC2*.

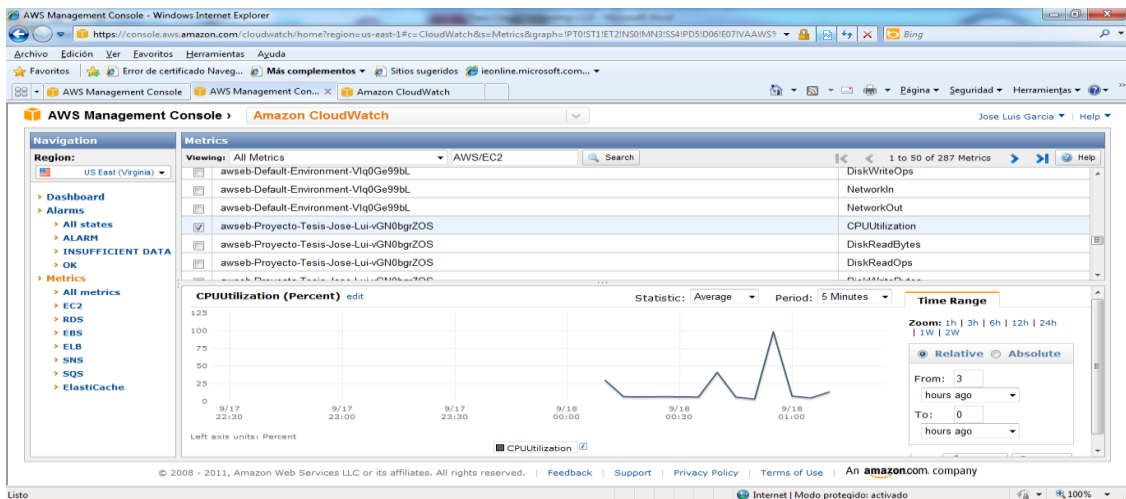


Figura 15. Gráficas del CPU, el disco y la red para la instancia seleccionada



Fuente: elaboración propia.

Figura 16. Gráfica de utilización promedio del CPU de la instancia seleccionada



Fuente: elaboración propia.

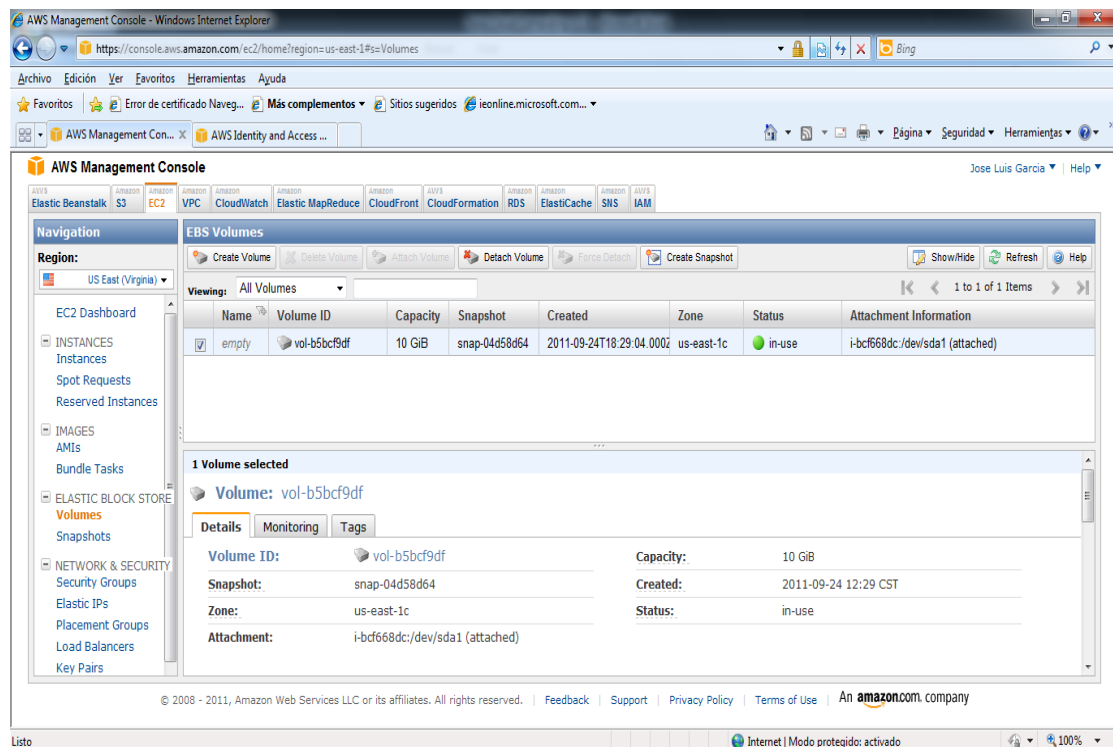
- *Auto Scaling*: con este servicio se pueden crear grupos de auto escala y automáticamente escalar la capacidad en ciertas condiciones, basadas en métricas que *Amazon CloudWatch* recopila. Permite escalar automáticamente la capacidad de *Amazon EC2*, ya sea para aumentarla o reducirla, de acuerdo con las condiciones que se definan. Con este servicio puede asegurarse de que el número de instancias que se estén utilizando, aumente sin interrupciones durante los picos de demanda, a fin de mantener el rendimiento; y se reduzca automáticamente, durante los períodos de calma en la demanda, para minimizar los costes.

El servicio resulta especialmente adecuado para aquellas aplicaciones que muestran variaciones de uso según la hora, el día o la semana. *Auto Scaling* está habilitado por *Amazon CloudWatch*. Permite definir las condiciones en las que desea aumentar o reducir el uso de *Amazon EC2*; cuando se cumple una de las condiciones, este servicio aplica automáticamente la acción que se ha definido.

- *Elastic Load Balancing*: puede distribuirse el tráfico de entrada a varias instancias, mediante la creación de un equilibrador de carga elástico, usando el servicio de *Elastic Load Balancing*, que permite conseguir aún más tolerancia a fallos en las aplicaciones, al proporcionar la capacidad de equilibrio de carga necesaria como respuesta al tráfico entrante de aplicaciones. Este servicio detecta instancias en mal estado dentro de un conjunto, y redirige automáticamente el tráfico hacia las instancias que se encuentran en buen estado, hasta que se restauran las instancias que están en mal estado.

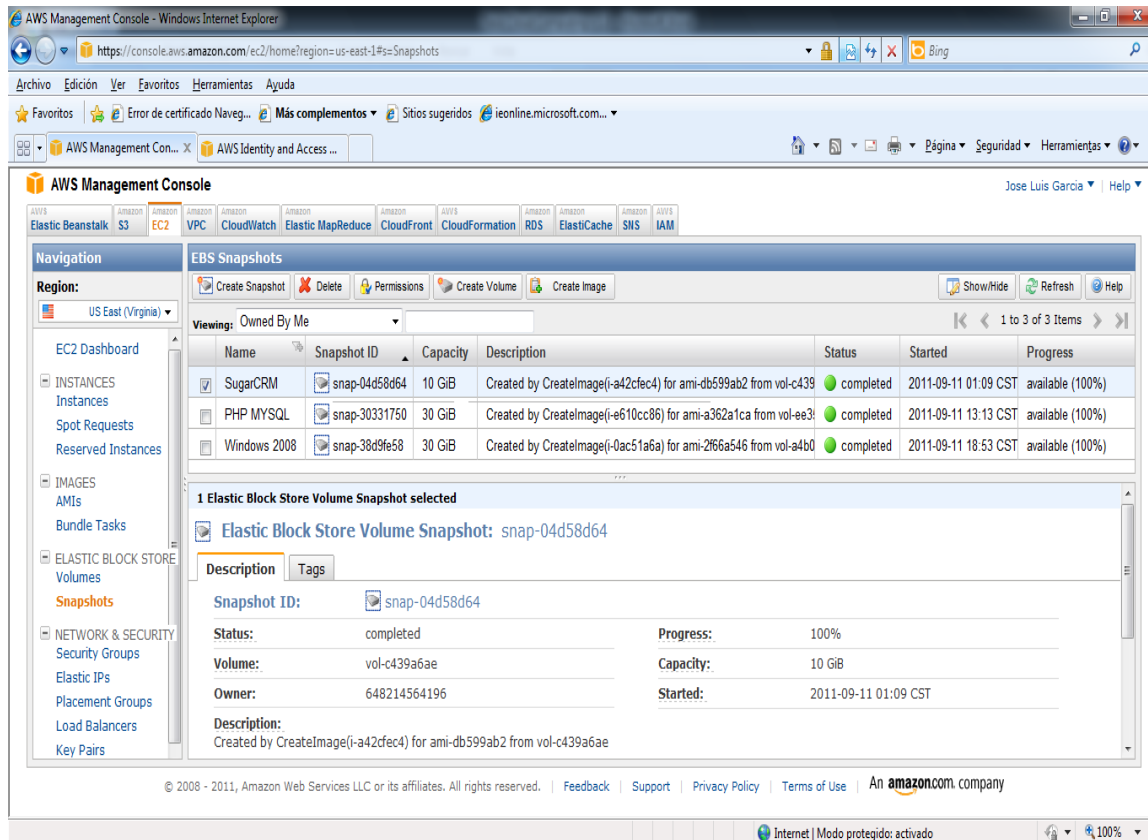
- Amazon Elastic Block Storage (EBS)*: proporciona volúmenes conectados a la red de almacenamiento persistente para instancias de Amazon EC2. Los volúmenes de *Amazon EBS*, constituyen almacenamiento fuera de la instancia que persisten con independencia de la vida de una instancia. Replicas instantáneas (*Snapshots*), consistentes en los volúmenes de EBS, pueden ser creadas y almacenadas con el servicio de *Amazon Simple Storage Service(S3)*. La próxima figura presenta una pantalla donde se ven las propiedades de un volumen de *Amazon EBS* de 10 gigabytes y la siguiente pantalla, las propiedades de las instantáneas hechas a tres volúmenes de EBS, que contienen sus propias configuraciones.

Figura 17. **Volumen de 10 GiB asociado a instancia actualmente en uso**



Fuente: elaboración propia.

Figura 18. **Snapshots de volúmenes de AMIs que tienen configurados sistemas operativos, aplicaciones, bases de datos, lenguajes de programación, etc**



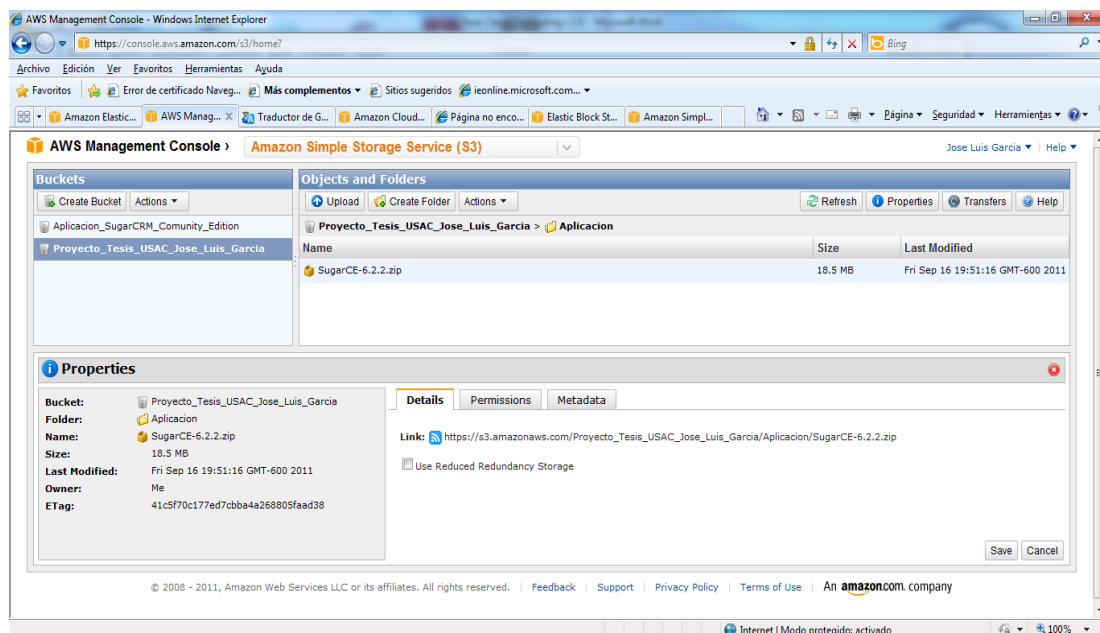
Fuente: elaboración propia.

- Amazon Simple Storage Service (S3):** es un almacén de datos distribuido y de alta duración. Con una simple interfaz de servicios web, se pueden almacenar y recuperar grandes cantidades de datos, como objetos en carpetas o contenedores (*buckets*), en cualquier momento, desde cualquier lugar en la web, utilizando verbos HTTP estándar. Incluye mecanismos de autenticación, diseñados para garantizar que los datos se mantienen seguros frente a accesos no autorizados.

Los objetos pueden hacerse privados o públicos, y pueden otorgarse derechos a usuarios determinados. Los objetos se almacenan de forma redundante en varios dispositivos de diversas instalaciones dentro de una región de *Amazon S3*. Para ayudar a garantizar la durabilidad, las operaciones *PUT* y *COPY* de *Amazon S3*, almacenan de forma sincrónica los datos en varias instalaciones antes de devolver *SUCCESS*. Una vez almacenados, *Amazon S3* mantiene la durabilidad de los objetos, detectando y reparando rápidamente cualquier pérdida de redundancia.

Del mismo modo, *Amazon S3*, comprueba de forma regular la integridad de los datos almacenados mediante sumas de comprobación. La siguiente figura contiene una pantalla con un ejemplo de estructura de almacenamiento en *Amazon S3*.

Figura 19. **Ejemplo de almacenamiento en *Simple Storage Service (S3)***



Fuente: elaboración propia.

- *Amazon CloudFront*: es un servicio *web* diseñado para la distribución de contenido. Se integra con otros servicios de *Amazon*, ofreciendo una forma sencilla de distribuir contenido a los usuarios finales con baja latencia, altas velocidades de transferencia de datos y sin compromisos. Trabaja a la perfección con *Amazon Simple Storage Service (Amazon S3)*, que almacena con altos niveles de durabilidad, las versiones originales definitivas de sus archivos.

Con este servicio, los objetos están organizados en distribuciones. Una distribución especifica la ubicación de la versión original de los objetos. Una distribución cuenta con un nombre de dominio *CloudFront.net* único (por ejemplo, *abc123.cloudfront.net*), que puede utilizar para hacer referencia a los objetos a través de la red de ubicaciones. Si se desea, puede también asignarse a la distribución, un nombre propio de dominio (por ejemplo, *imágenes.ejemplo.com*).

- *SimpleDB*: es un servicio *web* que proporciona la funcionalidad básica de una base de datos, búsquedas en tiempo real y consultas simples de datos estructurados sin la complejidad operativa. Se puede organizar el conjunto de datos en dominios y puede hacer búsquedas a través de todos los datos almacenados en un dominio particular. Los dominios son colecciones de objetos que se describen por pares atributo-valor. Es un almacén de datos no relacionales altamente disponible, ampliable y flexible que alivia la carga de trabajo de la administración de bases de datos.

Los desarrolladores simplemente almacenan elementos de datos y los consultan mediante solicitudes de servicios *web*, además, se encarga de crear y gestionar varias réplicas de los datos y las distribuye

geográficamente, para permitir alta disponibilidad y capacidad de duración.

- *Amazon Relational Database Service (RDS)*: es un servicio *web* que facilita las tareas de configuración, utilización y escalado de una base de datos relacional en la nube. Proporciona tamaño modificable y permite acceder a todas las funciones de una base de datos *Oracle* o *MySQL* convencionales. Este servicio puede hacer revisiones automáticamente en el *software* de base de datos, y puede realizar copias de seguridad de la misma, almacenando las copias durante un período de retención definido por el usuario, permitiendo la recuperación de puntos en el tiempo.

Además, facilita la tarea de utilización de la replicación para mejorar los niveles de disponibilidad y de fiabilidad de las bases de datos de producción. Si en cualquier momento se necesita capacidad adicional, se puede escalar los recursos informáticos y de almacenamiento asociados con la instancia de la base de datos con solo unos clics de la consola, o una sencilla llamada a API.

Toda aquella herramienta diseñada para el motor *Oracle* o *MySQL* funciona con *Amazon RDS* sin ningún tipo de modificación. Adicionalmente tiene la característica principal de ser una base de datos relacional lista para utilizarla en producción, sin tener que preocuparse del aprovisionamiento de la infraestructura, ni de la instalación y el mantenimiento del *software* de base de datos. Otra característica muy importante, es que se pueden tener copias de seguridad automatizadas y copias instantáneas de base de datos que se ejecutan en la infraestructura.

- *Amazon Simple Notification Service (SNS)*: es un servicio *web* que proporciona una manera simple de notificar a las aplicaciones o a las personas desde la nube, creando temas de los que desea informar a las aplicaciones o a las personas, utilizando un protocolo de publicación-suscripción. Facilita las tareas de configuración, utilización y envío de notificaciones desde la nube. Los mensajes se pueden entregar a través del protocolo que elija el cliente, como por ejemplo, correo electrónico. Este servicio puede utilizarse para crear flujos de trabajo y aplicaciones de mensajería de alta fiabilidad, gestionadas mediante eventos. Entre sus usos potenciales se incluyen las aplicaciones de supervisión, sistemas de flujo de trabajo, actualizaciones de información en las que el tiempo es un aspecto a tener en cuenta, aplicaciones móviles, etc.
- *Elastic Beanstalk*: es un servicio que permite una manera muy fácil de administrar y desplegar rápidamente las aplicaciones en la nube. Solo tiene que cargar la aplicación, y el servicio automáticamente maneja los detalles de la implementación de la capacidad de aprovisionamiento, balanceo de carga, auto escala, y la vigilancia de la salud de la aplicación. Al mismo tiempo, con éste servicio, se conserva el control total de los recursos, y puede acceder a los recursos subyacentes en cualquier momento.

El servicio aprovecha los servicios de *Amazon S3*, *Amazon Simple Notification Service (SNS)*, *Elastic Load Balancing* y *Auto Scaling*, para ofrecer la misma infraestructura altamente confiable, escalable y rentable que cientos de miles de empresas dependen hoy en día. Con este servicio se conserva el control total de los recursos al impulsar la aplicación, ya que puede hacerse cargo de algunos o todos los elementos de la infraestructura. Actualmente la versión de este servicio



está construida para los desarrolladores de Java con Apache *Tomcat* como servidor web para asegurar la portabilidad de las aplicaciones.

- *CloudFormation*: es un servicio que ofrece a los desarrolladores y administradores de sistemas, una forma fácil de crear una colección de recursos relacionados, y presentarlos en forma ordenada y predecible. Los desarrolladores pueden utilizar las plantillas de muestra de este servicio, o crear sus propias plantillas para describir los recursos, así como las dependencias asociadas o los parámetros de tiempo de ejecución necesarios para ejecutar su aplicación.
- Los usuarios pueden crear una plantilla y desplegar su colección asociada de recursos (llamada pila), a través de la consola de administración de AWS, la herramienta de línea de comando de *Cloud Formation*, o por medio de APIs. El servicio se integra con el servicio de notificación de *Amazon, Simple Notification Service (SNS)*, que le permite recibir notificaciones, como el avance de la creación y eliminación de la pila, además de que le proporciona el estado; esto también permite que otros programas estén pendientes de eventos dentro del servicio y respondan, o incluso participen, en el proceso de configuración de la pila de recursos.
- *Amazon Virtual Private Cloud (VPC)*: este servicio permite definir una sección privada y aislada de la nube, donde se pueden iniciar recursos en una red virtual que se defina. Controla todos los aspectos del entorno de la red virtual, incluida la selección del rango de direcciones IP, la creación de subredes y la configuración de tablas de rutas y puertas de enlace de red. Con este servicio se puede crear una subred de cara al público para los servidores *web* con acceso a *Internet*, y colocar los

sistemas de fondo, como bases de datos o servidores de aplicaciones, en una subred de uso privado sin acceso a *Internet*. Se pueden definir varias capas de seguridad, así como grupos de seguridad y listas de control de acceso a red, para ayudar a controlar el acceso a las instancias desde cada subred.

### **3.4. Monitoreo de rendimiento de máquinas virtuales**

Para obtener un mejor rendimiento en el funcionamiento de la infraestructura virtual, es necesario contar con la supervisión de cada uno de los recursos en la nube, proporcionando datos sobre su utilización, el rendimiento operativo y los patrones globales de demanda, incluidos indicadores como la utilización de la CPU, las lecturas y escrituras de disco y el tráfico de red, ancho de banda, latencia, etc.

Es importante conservar o contar con los datos de supervisión durante algunos días o semanas, con el objetivo de comprobar en los indicadores precedentes cualquier suceso de interés, para habilitar las medidas preventivas o correctivas correspondientes que permitan añadir o eliminar en forma dinámica, de acuerdo a condiciones previamente definidas, las instancias o volúmenes de disco necesarios. Esta capacidad para añadir o eliminar de forma dinámica las instancias o los volúmenes, permite seguir la curva de demanda de las aplicaciones con mucho mayor detalle.

Las herramientas que pueden proporcionar las métricas, pueden desplegar una serie de gráficas basadas en los datos recolectados, o dependiendo de las necesidades, se pueden utilizarlas métricas directamente para procesar la información y ejecutar procesos dinámicos para añadir o eliminar recursos.

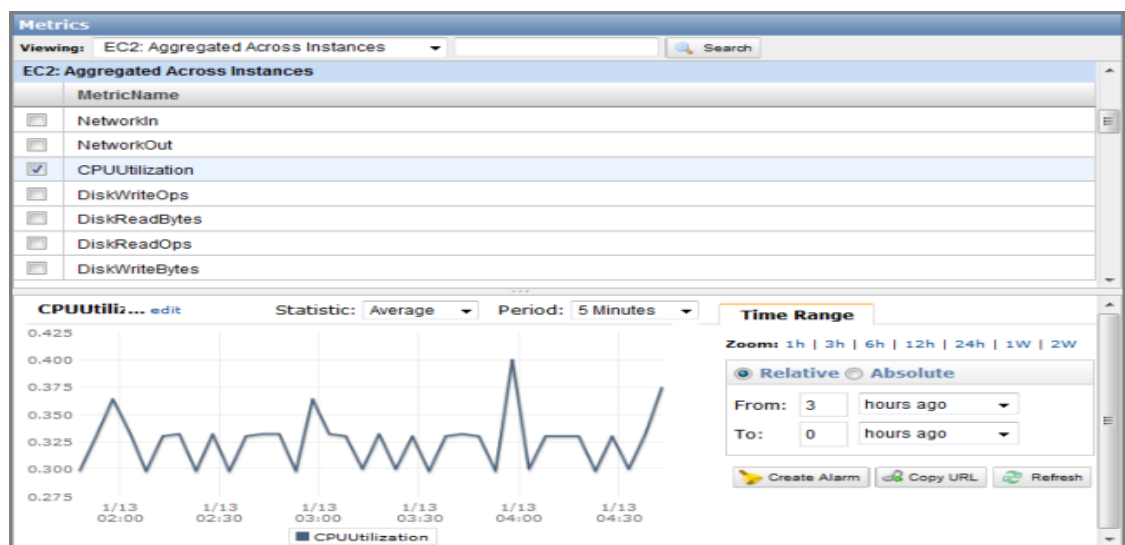
Las gráficas de monitoreo regularmente despliegan información sobre:

- Porcentaje promedio de utilización del CPU
- Promedio de lecturas a disco (*Bytes*)
- Promedio de escrituras a disco (*Bytes*)
- Máxima entrada a la red (*Bytes*)
- Máxima salida de la red (*Bytes*)
- Lectura y escritura de ancho de banda (KiB/s)

Las gráficas de las métricas ayudan a solucionar problemas y descubrir las tendencias, crear y definir las alarmas que notifiquen de los problemas, y dar un vistazo general del comportamiento de los recursos en la nube.

La siguiente figura, muestra el gráfico de la métrica de la utilización de CPU de una instancia en la nube.

Figura 20. **Métricas de utilización de CPU**



Fuente: elaboración propia.

Algunas de las métricas utilizadas regularmente son:

- Utilización del CPU: porcentaje de unidades de cómputo asignadas que están actualmente en uso en la instancia. Esta medida indica el poder de procesamiento requerido para ejecutar una aplicación en una instancia.
- Lecturas de disco: cantidad de operaciones completas de lectura de todos los discos disponibles para las instancias. Esta métrica identifica la velocidad a la que una aplicación lee los datos de un disco duro.
- Escrituras a disco: cantidad de operaciones completas de escritura de todos los discos duros disponibles para las instancias. Esta métrica identifica la velocidad a la que una aplicación escribe los datos en un disco duro.
- Lecturas de disco (*Bytes*): cantidad de *bytes* leídos de todos los discos disponibles para la instancia. Esta métrica se utiliza para determinar el volumen de los datos que la aplicación lee desde el disco duro de la instancia. Esto puede ser usado para determinar la velocidad de la aplicación.
- Escrituras a disco (*Bytes*): cantidad de *bytes* escritos a todos los discos disponibles para la instancia. esta métrica se utiliza para determinar el volumen de los datos que la aplicación escribe al disco duro de la instancia. Esto puede ser usado para determinar la velocidad de la aplicación.

- Entradas de red: número de *bytes* recibidos en todas las interfaces de red de la instancia. Esta medida identifica el volumen de tráfico de red entrante para una aplicación en una sola instancia.
- Salidas de red: número de *bytes* enviados en todas las interfaces de red de la instancia. Esta medida identifica el volumen de tráfico de red saliente para una aplicación en una sola instancia. Conexiones a la BD: cantidad o número de conexiones de base de datos en uso.
- Memoria disponible: cantidad disponible de memoria de acceso aleatorio.
- Uso de *swap*: cantidad de espacio de intercambio utilizado en la instancia de la base de datos.

Utilizar el monitoreo del rendimiento de máquinas virtuales, permite recopilar, analizar y ver las métricas del sistema y la aplicación, para poder tomar decisiones operativas y de negocio más rápidamente y con mayor confianza.

Al tener los datos de las diferentes métricas, tal como se mencionó anteriormente, se pueden definir alarmas que pueden ayudar a implementar las decisiones con más facilidad, ejecutando procesos de forma automática para realizar cambios en los recursos de acuerdo a reglas definidas, para mantener las aplicaciones sanas y en perfecto funcionamiento. Por ejemplo, las métricas se pueden utilizar para descubrir que un sitio web funciona mejor cuando el tráfico de red se mantiene por debajo de un cierto parámetro en las instancias. A continuación se puede crear un procedimiento automatizado, para asegurar

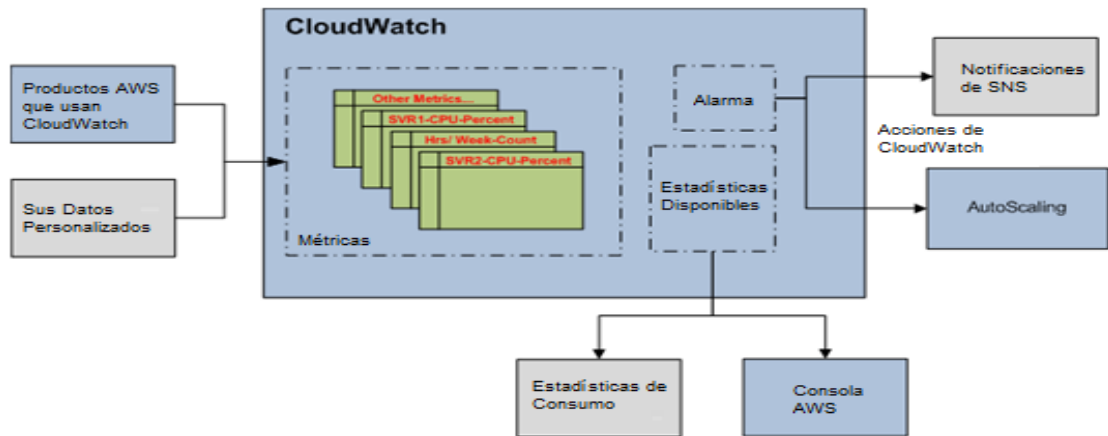
que siempre se tiene el número adecuado de recursos, para que coincida con la cantidad de tráfico que se tiene.

Las métricas también se pueden utilizar para diagnosticar problemas observando el desempeño del sistema antes y después de que un problema ocurra.

El servicio *Amazon CloudWatch*, permite llevar el control de todas las métricas que se definan para controlar las aplicaciones y entorno virtual, ya que permite monitorear, manejar y publicar varias métricas, así como configurar las acciones de alarma basados en los datos de las métricas. Almacena los datos de las mediciones durante 2 semanas. Puede publicar los datos de las métricas provenientes de múltiples fuentes, tales como el tráfico de entrada de decenas de diferentes instancias, o solicitudes de visitas a páginas de varias aplicaciones *web* diferentes.

Como se puede ver en la siguiente figura, *Amazon CloudWatch* es básicamente un repositorio de métricas. Un producto de AWS, así como *Amazon EC2*, pone métricas en el repositorio y se recuperan las estadísticas basadas en las métricas. Si se ponen las métricas personalizadas en el repositorio, también puede recuperar las estadísticas basadas en las métricas personalizadas.

Figura 21. **Amazon CloudWatch**



Fuente: elaboración propia.

*CloudWatch* actualmente proporciona monitoreo para los siguientes servicios:

- *Amazon Elastic Compute Cloud (EC2)*: comienza el envío de estadísticas automáticamente al iniciar cualquier instancia.
- *Amazon Elastic Block Store*: se inicia el envío de estadísticas de forma automática al momento de montar el volumen.
- *Amazon Relational Database Service (Amazon RDS)*: comienza a enviar las estadísticas específicas cuando comienza a utilizar RDS.
- *Elastic Load Balancing*: comienza a enviar las estadísticas cuando se comienza a utilizar.
- *Auto Scaling*: comienza a enviar las estadísticas cuando se comienza a utilizar.

### **3.5. Herramientas que permitan lograr a la empresa obtener máximos rendimientos, escalabilidad y disponibilidad de su infraestructura virtual de forma automática**

Antes de iniciar con la explicación de las herramientas que permiten tener máximos rendimientos, escalabilidad y disponibilidad en la nube, se verán algunos nuevos conceptos que cambiarán la forma en que las aplicaciones son construidas e implementadas.

Construcción de una arquitectura escalable: es fundamental construir una arquitectura escalable con el fin de tomar ventaja de dicha infraestructura.

La nube está diseñada para proporcionar la escalabilidad conceptualmente infinita. Sin embargo, no se puede aprovechar toda esa escalabilidad en la infraestructura, si la arquitectura no es escalable. Ambas tienen que trabajar juntas. Se tienen que identificar los componentes monolíticos y cuellos de botella en la arquitectura, identificar las áreas en las que no pueden aprovecharse las capacidades de aprovisionamiento en la arquitectura, y trabajar para refactorizar la aplicación con el fin de aprovechar la infraestructura escalable y tomar ventaja de la nube.

Las siguientes son características de una aplicación verdaderamente escalable:

- El aumento de recursos resulta en un incremento proporcional en el rendimiento
- Un servicio escalable es capaz de manejar la heterogeneidad



- Un servicio escalable es operativamente eficiente
- Un servicio escalable es resistente
- Un servicio escalable debería hacerse más rentable cuando crece (costo por unidad disminuye a medida que aumenta el número de unidades)

Estas son cosas que deben convertirse en una parte inherente de la aplicación, y si se diseña una arquitectura con las características anteriores en mente, tanto la arquitectura e infraestructura, trabajaran juntas para dar la escalabilidad que se está buscando.

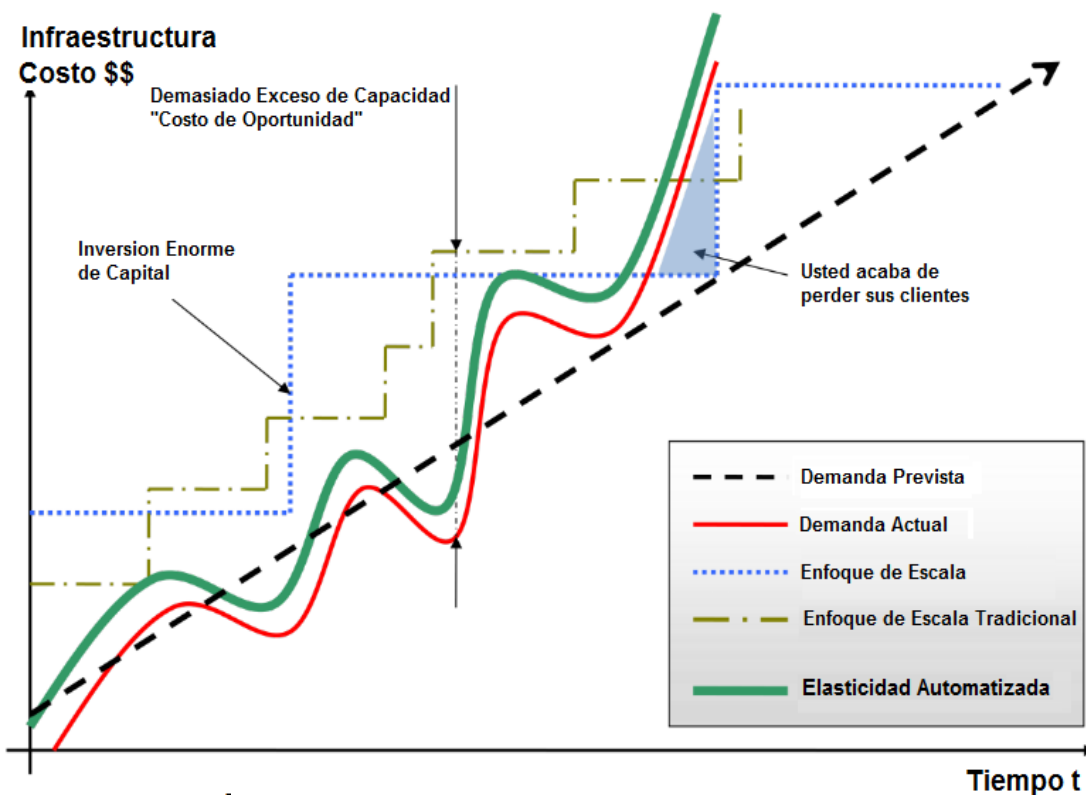
Enfoque de escala: no preocuparse por la arquitectura de aplicaciones escalables, y de grandes inversiones en muy potentes ordenadores (escala vertical), para dar cabida a la demanda. Este enfoque usualmente trabaja hasta cierto punto, pero bien podría costar una fortuna (vea Inversión Enorme de Capital en el diagrama), o la demanda podría superar la capacidad de crecer, antes de que la nueva infraestructura sea implementada (vea Usted acaba de perder sus clientes en el diagrama).

Enfoque de escala tradicional: crea una arquitectura que escala horizontalmente e invierte en infraestructura en pequeñas porciones. La mayoría de las empresas y las grandes aplicaciones *web* siguen este patrón de distribución de sus componentes de la aplicación, que reunirá a sus conjuntos de datos y utilizará un diseño orientado al servicio. Este enfoque es a menudo más eficaz que un enfoque de escala. Sin embargo, esto requiere la predicción de la demanda en intervalos regulares y el despliegue de la infraestructura en porciones para satisfacer la demanda. Esto a menudo conduce a un exceso de capacidad (quema de efectivo) y el constante control manual (quema de los

ciclos humanos). Además, por lo general no funciona si la aplicación es una víctima de un incendio viral, a menudo referida como el efecto *Slashdot*. ([http://en.wikipedia.org/wiki/Slashdot\\_effect](http://en.wikipedia.org/wiki/Slashdot_effect))

Comprendiendo la elasticidad: el siguiente gráfico ilustra los diferentes enfoques que un arquitecto de la nube puede tomar para escalar sus aplicaciones para satisfacer la demanda.

Figura 22. **Elasticidad y escalabilidad automatizada**



Fuente: elaboración propia.

La infraestructura tradicional generalmente necesita la predicción de la cantidad de recursos de computación que su aplicación va a utilizar durante un periodo de varios años. Si usted subestima, sus aplicaciones no tendrán la potencia necesaria para manejar el tráfico inesperado, resultando potencialmente la insatisfacción del cliente. Si usted, sobreestima, usted está desperdiciando dinero con los recursos superfluos.

La naturaleza en la demanda y la elasticidad del enfoque de la nube (elasticidad automatizada), no obstante, permite que la infraestructura necesaria, sea alineada (como se expande y contrae), con la demanda real, lo que aumenta la utilización global y la reducción de costos.

La elasticidad es una de las propiedades fundamentales de la nube, es la capacidad de escalar los recursos de computación de arriba a abajo con facilidad y con una fracción mínima. Es importante entender que la elasticidad en última instancia, conducirá la mayor parte de los beneficios de la nube. Como arquitecto de la nube, es necesario internalizar este concepto y trabajarlo en la arquitectura de su aplicación con el fin de obtener el máximo beneficio de la nube.

Tradicionalmente, las aplicaciones se han construido para la infraestructura fija, rígida y pre provisionada. Las compañías nunca tuvieron la necesidad de provisionar e instalar los servidores a diario. Como resultado, la mayoría de las arquitecturas de *software* no se refieren al despliegue rápido o a la reducción de *hardware*. Desde que los tiempos de aprovisionamiento y la inversión inicial para la adquisición de nuevos recursos fue muy alta, los arquitectos de *software* nunca han invertido tiempo y recursos en la optimización de la utilización del *hardware*. Era aceptable si el hardware en el que se ejecuta la aplicación era subutilizado. La noción de elasticidad dentro de

una arquitectura fue pasada por alto, porque la idea de tener nuevos recursos en cuestión de minutos, no era posible.

Con la nube, esta mentalidad necesita cambiar. La computación en nube hace más eficiente el proceso de adquisición de los recursos necesarios, ya no hay ninguna necesidad de hacer pedidos antes de tiempo y mantener cautivo el *hardware* utilizado. En cambio, los arquitectos de nube, pueden solicitar lo que necesitan sólo unos minutos antes de que lo necesite, o automatizar el proceso de adquisiciones, aprovechando la vasta escala y tiempo de respuesta rápido de la nube. Lo mismo es aplicable a la liberación de los recursos innecesarios o subutilizados cuando no los necesitan.

*Auto Scaling*: es un servicio web que permite automáticamente poner en marcha o terminar instancias sobre la base de políticas definidas por el usuario, controles del estado de salud y horarios. Este servicio se usa junto con los servicios *Amazon CloudWatch* y *Elastic Load Balancing*. Es útil para mantener una flota de instancias que puede manejar la carga presentada. Como su nombre lo indica, *Auto Scaling* responde automáticamente a las condiciones cambiantes, todo lo que se necesita hacer es especificar cómo debería responder a esos cambios. Por ejemplo, puede indicarse a *Auto Scaling* iniciar una instancia adicional siempre que el uso del CPU exceda el 90 por ciento durante 10 minutos o terminar la mitad de las instancias de el sitio *web* el fin de semana cuando el tráfico se espere que sea bajo.

También se puede utilizar para asegurarse de que las instancias de la flota están funcionando de manera óptima, por lo que las aplicaciones continuarán funcionando de manera eficiente. Grupos de auto escala pueden trabajar a través de la disponibilidad de múltiples zonas, distintas ubicaciones físicas, de manera que si una zona de disponibilidad no está disponible, *Auto*

*Scaling* automáticamente redistribuirá las aplicaciones a una zona de disponibilidad diferente.

## Características

*Auto Scaling* ofrece varias características que ayudan a ahorrar tiempo y dinero, lo que provoca obtener máximos rendimientos.

- Capacidad elástica: añade automáticamente capacidad de cálculo cuando el uso de la aplicación se levanta y lo retira cuando el uso cae.
- Fácil de usar: administrar las instancias distribuidas en una o varias zonas de disponibilidad, como una entidad colectiva única utilizando sencillas herramientas de líneas de comandos, o mediante programación a través de un servicio *web* fácil de usar (API).
- Ahorro de costos: ahorrar en costos de cómputo por terminación de instancias subutilizadas de forma automática e inicio de nuevas instancias cuando las necesite, sin necesidad de intervención manual.
- Redundancia geográfica y escalabilidad: distribuir, escalar y balancear aplicaciones automáticamente sobre múltiples zonas de disponibilidad dentro de una región.
- Fácil mantenimiento: reemplazar automáticamente instancias perdidas o no saludables en función de las alarmas predefinidas.

- Acciones programadas: planifique acciones de escala para los tiempos futuros y las fechas cuando se espera necesitar más o menos capacidad.

Este servicio está diseñado para ayudar a hacer el uso de *Amazon EC2* más fácil, ayudando a reducir la carga operativa de despliegue y mantenimiento de aplicaciones, ayuda a supervisar el estado de cada instancia que se inicia, si alguna instancia termina inesperadamente detecta la terminación e inicia una instancia de reemplazo, ayudando a mantener un número fijo deseado de instancias automáticamente.

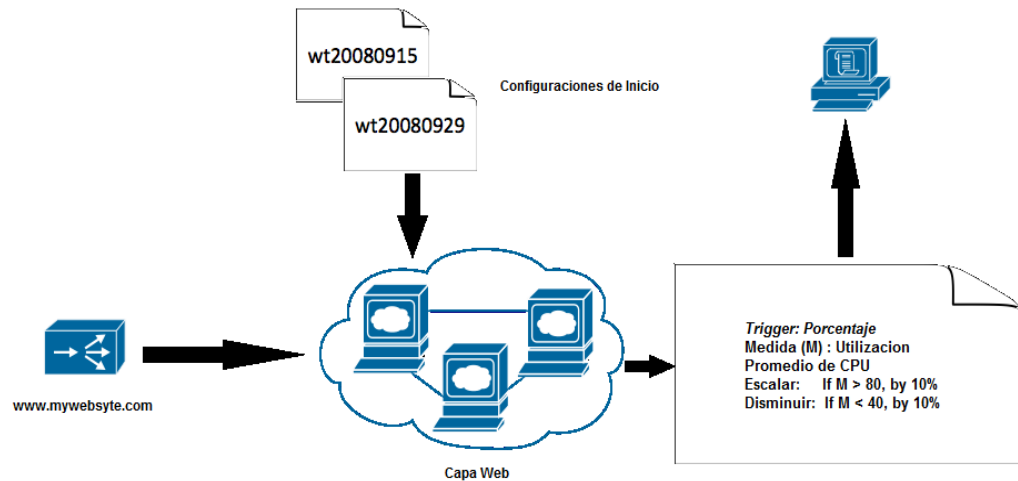
En un escenario común de EC2, varias copias de una aplicación se ejecutan al mismo tiempo para cubrir el volumen de tráfico de los clientes, ellos ven una sola dirección para la aplicación, pero detrás de ese simple URL hay muchas instancias idénticas, cada una de ellas manejando las solicitudes de los clientes.

Estas instancias se clasifican en grupos de escala automáticos, que es el concepto básico del servicio, los cuales se definen con un número mínimo y máximo de instancias. El servicio inicia más instancias (hasta el máximo definido), para el grupo de escala automático para manejar un incremento en el tráfico y, a medida que disminuye la demanda, pone instancias fuera de servicio para utilizar más eficientemente los recursos informáticos.

En la siguiente ilustración, el tráfico de internet se dirige desde la dirección URL pública hacia un grupo de auto escala llamado *capa web*, el grupo tiene disparadores que incrementan o disminuyen el tamaño del grupo automáticamente basado en la utilización promedio del CPU para todo el

grupo. Cuando un disparador se dispara, *Auto Scaling* utiliza una configuración de inicio para crear una nueva instancia.

Figura 23. **Trigger que dispara el servicio *Auto Scaling***



Fuente elaboración propia.

Cada grupo de auto escala, que se crea tiene una configuración de inicio, esta configuración permite describir cada instancia que *Auto Scaling* creará cuando un disparador se dispare.

Como se ha mencionado, *Auto Scaling* utiliza disparadores para indicar cuándo iniciar nuevas instancias y cuando ponerlas fuera de servicio. Un disparador es un mecanismo que se establece para indicar al sistema cuando se desea aumentar o disminuir el número de instancias. Puede configurar un disparador, para activarse en cualquier métrica publicada en *Amazon CloudWatch*, tal como *CPU Utilization*. Cuando se activa, el disparador pone en marcha un proceso de larga duración llamado *Scaling Activity*.

Se puede configurar el grupo de auto escala para que las peticiones de usuarios se distribuyan a través de un grupo de instancias. Se puede agregar un balanceador de cargas (*Load Balancer*), al grupo de auto escala y utilizar las métricas de *Elastic Load Balancing* para escalar la aplicación.

Grupo de auto escala: es una representación de *Amazon EC2*, que comparten características similares, y que son tratados como una agrupación lógica de los efectos de la ampliación y gestión de la instancia. Por ejemplo, si una sola aplicación opera a través de varias instancias, es posible que se desee disminuir o aumentar el número de instancias en ese grupo para mejorar el rendimiento de la aplicación. Se puede usar el grupo de auto escala para escalar de forma automática el número de instancias o mantener un número determinado de instancias. Este grupo puede contener instancias que provienen de una o más zonas de disponibilidad.

*HealthCheck*: es una llamada para verificar el estado de salud de cada instancia de un grupo de auto escala. Si una instancia reporta bajo rendimiento *Auto Scaling* termina la instancia e inicia otra para ocupar su lugar, esto asegura que el grupo de auto escala es consistente y funciona con normalidad.

Configuración de inicio: captura los parámetros necesarios para crear nuevas instancias. Solo se puede adjuntar una configuración de inicio a un grupo de auto escala a la vez. Al adjuntar una configuración de inicio nueva o actualizada al grupo de auto escala, las nuevas instancias se pondrán en marcha utilizando los nuevos parámetros de configuración. Las instancias existentes no se verán afectadas. Cuando *Auto Scaling* necesite reducir, primero termina las instancias que tienen configuración de inicio más vieja.



Disparador: un disparador es un concepto que combina dos funciones: una alarma *CloudWatch* (configurada para ver una métrica específica), y una política de *Auto Scaling*, que describe lo que debe suceder cuando la condición de alarma se cruce.

En la mayoría de los casos, se necesitan dos disparadores, un disparador para escalar y otro para reducir. Por ejemplo, si desea escalar cuando el uso del CPU aumenta al 80 por ciento, es necesario configurar una alarma *CloudWatch* y crear una política de escala automática. La alarma detecta cuando el uso del CPU ha alcanzado el 80 por ciento y envía un mensaje a *Auto Scaling*, el cual determina lo que debe hacer siguiendo las instrucciones en la política de escala. Si también se quiere reducir cuando el uso del CPU llega al 40 por ciento, se necesita un segundo disparador. En otras palabras, es necesario configurar una alarma separada *CloudWatch* para detectar la condición del 40 por ciento y crear una política independiente que escale hacia abajo.

Política: es un conjunto de instrucciones para *Auto Scaling*, que le indica al servicio cómo responder a los mensajes de alarma de *CloudWatch*. Se puede configurar una alarma *CloudWatch* para enviar un mensaje a *Auto Scaling*, cada vez que una métrica específica ha alcanzado un determinado valor de disparo. Cuando la alarma envía el mensaje, *Auto Scaling* ejecuta la política asociada a un grupo de auto escala para escalar el grupo hacia arriba o hacia abajo.

Alarma *CloudWatch*: es un objeto que vela por una única métrica. Una alarma puede cambiar de estado dependiendo del valor de la métrica. Cuando cambia un estado de la alarma se ejecuta una o más acciones.

Una alarma tiene tres posibles estados:

- *OK* – La métrica está dentro de las condiciones definidas.
- *ALARM* – La métrica se encuentra fuera de las condiciones definidas.
- *ISUFFICIENT\_DATA*- La métrica no está disponible o no hay suficientes datos disponibles para la métrica para establecer el estado de alarma *OK*.

Se invoca una acción cuando una alarma cambia de estado y permanece en ese estado por un número de períodos de tiempo. El cambio de estado tiene que ser mantenido por el número de períodos de tiempo que se especifique.

Terminación de instancia: una instancia se termina por respuesta a las llamadas específicas a una acción de terminación de instancia y también en respuesta a otras actividades de escalamiento. Por ejemplo, se puede terminar una instancia cuando se reequilibra una zona de disponibilidad o cuando disminuye el tamaño del grupo de escala.

Para terminar una instancia del grupo de auto escala, dicha instancia está sujeta a las siguientes condiciones:

- Cuando se intenta preservar instancias con la última configuración de inicio. En otras palabras, si el grupo de escala contiene instancias con configuraciones de inicio diferentes, *Auto Scaling* termina la instancia con una configuración de inicio que no está actualmente asociada con el grupo de auto escala.

- Se puede terminar una instancia en una zona de disponibilidad específica para mantener el balance a través de las zonas.

Después que auto escala determina que instancia específica va a terminar, comprueba si la instancia es parte de un grupo de *Elastic Load Balancing*, si es así, instruye al balanceador de carga para eliminar la instancia del grupo de balanceo de carga y espera a que la eliminación se complete. Si se determina que la instancia no es parte de un grupo de *Elastic Load Balancing*, se intenta terminar la instancia mediante la ejecución de scripts de apagado del sistema.

Tipos de escala: hay tres tipos de escala para el sistema: manual, por horario y por política.

Escala manual: es la forma más básica de la escala de los recursos. Hacer una llamada API o usar la interface de línea de comandos de *Auto Scaling* para iniciar o terminar una instancia. Solo se necesita especificar el cambio en la capacidad que se quiere y el proceso crea o destruye las instancias.

Escala por horario: a veces, se sabe exactamente cuándo tendrá que aumentar o disminuir el número de instancias en el grupo, simplemente porque esa necesidad surge en un horario predecible. Esto significa que las acciones de ampliación se realizan de forma automática en función de la hora y la fecha.

Escala por política: una forma más avanzada para escalar los recursos, escala por política, le permite definir los parámetros que informan al proceso de *Auto Scaling*. Por ejemplo, se puede crear una política que exige la ampliación de la flota cada vez que la tasa media de utilización de CPU se mantiene por

encima del noventa por ciento durante quince minutos. Esto es útil cuando se puede definir la forma en que se desea escalar en respuesta a las condiciones cambiantes, pero no se sabe cuándo las condiciones cambiarán, es decir, se puede configurar el escalado automático para responder por usted.

Hay que tener en cuenta que debería haber dos políticas, una de ampliación y una de reducción, para cada evento que se desee controlar. Por ejemplo, si se desea ampliar cuando el ancho de banda alcanza cierto nivel, se creará una política para indicarle a *Auto Scaling* que inicie cierto número de instancias para ayudarlo con el tráfico. Pero también se quiere una política de acompañamiento para reducir cierto número de instancias cuando el nivel de ancho de banda de la red vuelva a bajar.

#### Zonas de disponibilidad y regiones

Los recursos de computación en nube, están alojados en instalaciones de centros de datos de alta disponibilidad. Para proporcionar escalabilidad y fiabilidad adicionales, estos centros de datos se encuentran en varias ubicaciones físicas diferentes. Estos lugares se clasifican por regiones y zonas de disponibilidad. Las regiones son muy grandes y están dispersas en distintos puntos geográficos, en este momento, Amazon cuenta con cinco regiones:

- Noreste de Virginia, Estados Unidos
- California del norte, Estados Unidos
- Pacífico de Asia, Singapur
- Pacífico de Asia, Tokio
- Unión Europea, Irlanda

Las zonas de disponibilidad son lugares distintos dentro de una región que se han diseñado para aislarse de las fallas en otras zonas de disponibilidad, proporcionando bajo costo, baja latencia y conectividad de la red hacia otras zonas de disponibilidad en la misma región.

*Auto Scaling* permite tomar ventaja de la seguridad y fiabilidad de redundancia geográfica que benefician a grupos de auto escala en múltiples zonas de disponibilidad dentro de una región. Cuando una zona de disponibilidad se torna poco saludable o no está disponible, *Auto Scaling* inicia nuevas instancias en una zona de disponibilidad no afectada. Cuando una zona de disponibilidad poco saludable vuelve a un estado saludable, *Auto Scaling* automáticamente redistribuye las instancias de aplicación uniformemente a través de todas las zonas de disponibilidad designadas.

Distribución y balance de instancias a través de múltiples zonas

*Auto Scaling* intenta distribuir las instancias uniformemente entre las zonas de disponibilidad que están habilitadas para el grupo de auto escala. *Auto Scaling* intenta iniciar nuevas instancias en la zona de disponibilidad con el menor número de instancias, si el intento falla intentará iniciarlas en otras zonas hasta que tenga éxito.

Ciertas operaciones y condiciones pueden causar que el grupo de auto escala se pueda salir de balance. *Auto Scaling*, compensa mediante la creación de una actividad de reequilibrio en cualquier de las siguientes condiciones:

- Se emite una solicitud de cambio de zona de disponibilidad para su grupo.
- Hace un llamado al comando *Terminate Instance In Auto Scaling Group* que hace que el grupo se desequilibre.
- Una zona de disponibilidad que anteriormente tenía una capacidad insuficiente se recupera y tiene una capacidad adicional disponible.

*Auto Scaling* siempre inicia nuevas instancias antes de intentar poner fin a las antiguas, por lo que una actividad de reequilibrio no pondrá en peligro el rendimiento o la disponibilidad de la aplicación.

Instancias multi zona cuentan cuando se aproxima la capacidad

Debido a que *Auto Scaling* siempre intenta iniciar nuevas instancias antes de terminar las antiguas, estar en o cerca de la capacidad máxima indicada, podría impedir o detener por completo las actividades de reequilibrio. Para evitar este problema, puede exceder temporalmente la capacidad máxima especificada de un grupo por un margen del 10 por ciento durante una actividad de reequilibrio, o por un margen de una instancia que sea mayor. El margen se extiende sólo si el grupo está en o cerca de su máxima capacidad y necesita de reequilibrio, ya sea como resultado de la re zonificación solicitada por el usuario o para compensar los problemas de disponibilidad de la zona. La extensión solo dura el tiempo necesario para reequilibrar el grupo, por lo general unos pocos minutos.

### **3.6. Procesos y herramientas de software utilizadas en la automatización para la construcción de aplicaciones y entornos de aplicaciones**

Al automatizar la infraestructura, una de las ventajas más importantes de la utilización de un entorno de computación en nube es la capacidad de utilizar las API (*Application Program Interface*), de la nube para automatizar el proceso de implementación. Se recomienda que se tome el tiempo para crear un proceso automatizado de implementación desde el principio en el proceso de migración y no esperar hasta el final. La creación de un proceso de implementación automatizada y repetible, ayudará a reducir los errores y facilitar un proceso de actualización eficiente y escalable.

Para automatizar el proceso de implementación es necesario:

- Crear una biblioteca de recetas, es decir, pequeños scripts de uso frecuente para la instalación y configuración.
- Gestionar el proceso de configuración y despliegue usando agentes agrupados dentro de una AMI (*Amazon Machine Image*).
- Iniciar las instancias.

Iniciar las instancias

Deje que la instancia haga una pregunta en el arranque, ¿Quién soy y cuál es mi papel?, cada instancia debe tener un papel, (servidor de base de datos, servidor de aplicaciones, servidor esclavo en el caso de una aplicación *web*), que jugará en el entorno. Este papel o rol, puede ser pasado como

argumento durante el inicio que instruye a la AMI al ejecutar los pasos a seguir después que se haya iniciado. En el arranque, las instancias deben tomar los recursos necesarios (código, *scripts*, configuración), basado en el papel o rol y adjuntarse a un grupo de instancias para servir a sus funciones.

Beneficios de iniciar las instancias:

- Recrear el ambiente (desarrollo, puesta en escena, producción), con unos pocos clics y el mínimo esfuerzo.
- Un mayor control sobre sus recursos abstractos basados en la nube.
- Reducir los errores inducidos por el hombre al momento del despliegue.
- Crear un entorno auto detectable y auto corregible que es más resistente a fallos del *hardware*.

Tácticas específicas para automatizarla infraestructura:

- Definir grupos de instancias de auto escala, para diferentes clústeres utilizando la característica de auto escala en *Amazon EC2*.
- Monitorear las métricas del sistema (CPU, memoria, disco E/S, redes E/S), usando *CloudWatch* y tomar las medidas adecuadas (el inicio de nuevas AMIs de forma dinámica utilizando el servicio de auto escala), o enviar notificaciones.



- Almacenar y recuperar información de configuración de la máquina de forma dinámica: utilizar *Amazon SimpleDB* para obtener los datos de configuración durante el arranque de una instancia (por ejemplo, las cadenas de conexión de bases de datos). *SimpleDB* también se puede utilizar para almacenar información sobre una instancia como la dirección IP, nombre de la máquina y el rol.
- Diseñar un proceso de construcción de tal manera que vuelque las últimas versiones a una carpeta (*buckets*), en *Amazon Simple Storage Service* (S3); descargando la última versión de una aplicación desde el arranque del sistema.
- Invertir en la construcción de herramientas de gestión de recursos (scripts automatizados, imágenes de máquinas virtuales pre configuradas), o el uso inteligente de herramientas de administración de configuración de código abierto como *Chef*, *Puppet*, *CFEngine* o *Genome*.
- Incluir solo el sistema operativo suficiente y las dependencias de software en una AMI (*Amazon Machine Image*), para que sea más fácil de manejar y mantener. Pasar archivos de configuración o parámetros en tiempo de inicio y recuperar datos de usuario y metadatos de la instancia después de haber iniciado.
- Reducir el tiempo de inicio al arrancar desde volúmenes de *Amazon EBS* (*Elastic Block Store*) al adjuntar varios volúmenes de *Amazon EBS* a una instancia. Crear replicas instantáneas de volúmenes comunes y compartir dichas replicas instantáneas entre cuentas cuando sea apropiado.

### **3.7. Tareas para la administración de aplicaciones y entorno de aplicaciones al inicio y durante todo el ciclo de vida de las aplicaciones**

Para lograr administrar de una mejor manera las aplicaciones es importante automatizar tanto como sea posible: esto es aplicable en todo lo que hacemos y requiere una mención especial debido a que la automatización es a menudo ignorada. Una de las mayores características de *Amazon EC2*, es que se pueden aprovisionar cualquier número de instancias mediante una simple llamada de servicio *web*. La automatización dará poder a los desarrolladores para ejecutar un centro de datos dinámico programable que se expande y contrae sobre la base de sus necesidades.

Por ejemplo, la automatización del ciclo construcción-pruebas-implementación en la forma de una *Amazon Machine Image* (AMI), y luego ejecutarla automáticamente sobre *Amazon EC2* todas las noches (usando una tarea cron), ahorrará mucho tiempo. Al automatizar el proceso de creación del AMI, se puede ahorrar mucho tiempo en la configuración y optimización.

Además es importante para una mejor administración, construir aplicaciones y sus entornos en base a las mejores prácticas de la nube, luego las tareas administrativas se tornan sencillas utilizando las diferentes herramientas y servicios que presta la nube.

A continuación se presentan algunas de las mejores prácticas para utilizar en la nube.

## Diseño para el fracaso y nada fallará

Regla de oro: hay que ser pesimista en el diseño de arquitecturas de la nube, hay que asumir que las cosas van a fallar. En otras palabras, siempre hay que diseñar, implementar y desplegar para la recuperación automática de las fallas.

En particular se supone que el *hardware* fallará. Hay que suponer que los apagones se producirán, que algún desastre golpeará la aplicación. Hay que suponer que habrá más de la cantidad esperada de solicitudes por segundo en algún día. Hay que suponer que con el tiempo, el *software* de aplicación fallará también. Al ser pesimista, se termina pensando en las estrategias de recuperación en tiempo de diseño, lo que ayuda en el diseño de un mejor sistema global.

Si se da cuenta de que las cosas fallan con el tiempo, y se incorpora ese pensamiento en la arquitectura, para construir mecanismos que manejen ese error antes de que ocurra un desastre haciéndole frente con una infraestructura escalable, el resultado final será la creación de una arquitectura tolerante a fallos que se ha optimizado para la nube.

Preguntas que se deben hacer:

- ¿Qué pasa si un nodo en el sistema falla?
- ¿Cómo reconocer esa falla?
- ¿Cómo se puede reemplazar ese nodo?

- ¿Qué tipo de escenarios se tienen que planear?
- ¿Cuáles son los puntos únicos de fallo?

Si un equilibrador de carga está sentado frente a una gran variedad de servidores de aplicaciones, ¿qué pasa si ese equilibrador de carga falla?, si hay maestros y esclavos en la arquitectura, ¿qué pasa si el nodo maestro falla? ¿Cómo se produce la conmutación por error y cómo es instanciado un nuevo esclavo y puesto en sintonía con el maestro?

Al igual que el diseño de un fallo de hardware, también se tiene que diseñar para fallo de *software*. Las preguntas que se deben hacer son las siguientes:

- ¿Qué pasa con la aplicación si los servicios dependientes cambian su interfaz?
- ¿Qué pasa si las claves de cache crecen más allá del límite de memoria de una instancia?

Construir mecanismos para manejar la falla. Por ejemplo, las siguientes estrategias pueden ayudar en caso de fallo.

- Tener una copia de seguridad coherente y una estrategia de restauración de los datos y automatizarla.
- Construir subprocesos que reanuden el reinicio.

- Permitir que el estado del sistema se pueda volver a sincronizar mediante la recarga de mensajes de las colas.
- Mantener las imágenes virtuales pre-configuradas y pre-optimizadas para soportar (2) y (3) sobre inicio/arranque.
- Evitar sesiones en memoria, mover éstas a los almacenes de datos.

Las buenas arquitecturas de nube deben ser impermeables para reinicios. Mediante el uso de una combinación de *Amazon SQS* y *SimpleDB*, la arquitectura general del controlador es muy resistente a los tipos de fallas. Por ejemplo, si la instancia en la que se estaba ejecutando el controlador se muere, se puede traer y volver a su estado anterior como si nada hubiera pasado. Esto se logra mediante la creación de una AMI (*Amazon Machine Image*) pre configurada, que cuando se inicia, saca de la cola todos los mensajes de *Amazon SQS*, y lee sus estados desde un dominio de *Amazon SimpleDB* en el reinicio.

Diseñar con el supuesto de que el *hardware* subyacente fallará, permite prepararse para el futuro cuando realmente falle.

Este principio de diseño ayudará a diseñar aplicaciones con operaciones amigables. Si se puede extender éste principio en una medida pro activa y balancear la carga dinámicamente, es posible que pueda hacer frente a la variación en la red y el rendimiento del disco que existe debido a la naturaleza multiusuario de la nube.

## Tareas para administración de aplicaciones y sus entornos:

- Migrar elegantemente después de un error usando IP *elásticos*: un IP elástico, es una dirección IP estática que es dinámicamente reasignable, que permite rápidamente reasignar y migrar después de un error a otro conjunto de servidores, de modo que el tráfico se dirige a los nuevos servidores. Funciona muy bien cuando se quiere actualizar de antiguas a nuevas versiones o en caso de fallos de *hardware*.
- Utilizar varias zonas de disponibilidad: son conceptualmente como centros de datos lógicos (*data centers lógicos*). Mediante la implementación de la arquitectura a las múltiples zonas de disponibilidad, se puede asegurar una alta disponibilidad.
- Utilizar *Amazon RDS* con la funcionalidad Multi-AZ, que permite replicar automáticamente las actualizaciones de base de datos a través de múltiples zonas de disponibilidad.
- Mantener una AMI (*Amazon Machine Image*), para que pueda restaurar y clonar entornos muy fácilmente en una zona de disponibilidad diferente; mantener varias bases de datos esclavas a través de zonas de disponibilidad y replicación de configuración caliente.
- Utilizar *Amazon CloudWatch* o varias herramientas de *software* libre para seguimiento en tiempo real, para obtener mayor visibilidad y tomar las medidas adecuadas en caso de fallo de *hardware* o degradación del rendimiento. Configure un grupo de escala automático, para mantener fijo el tamaño de la flota para que sustituya instancias EC2 poco saludables por otras nuevas.

- Utilizar *Amazon* EBS y configurar trabajos cron, para que las instantáneas incrementales se carguen automáticamente en *Amazon* S3 y los datos se conserven independientemente de las instancias.
- Utilizar *Amazon* RDS y establecer el período de retención de copias de seguridad, de modo que pueda realizar copias de seguridad automatizadas.

### **3.8. Ventajas y desventajas existentes al momento de utilizar las diferentes herramientas para la administración, construcción y despliegue de aplicaciones y sus entornos en la nube de manera automática**

#### Ventajas

- La fácil utilización de las herramientas: presentan de forma gráfica la forma de administrar y configurar el entorno y las aplicaciones. Dichas herramientas permiten por ejemplo, seleccionar una imagen de plantilla pre configurada para empezar a funcionar inmediatamente, configurar la seguridad y acceso a la red, seleccionar sistemas operativos, seleccionar el tamaño de la instancia, si se desea ejecutar la instancia en varias localizaciones geográficas, etc.
- Acceso a todas las herramientas en una misma consola: desde una misma pantalla, se puede acceder directamente a los diferentes servicios y herramientas que nos proporciona la nube, permitiendo monitorear y administrar todos los recursos de manera sencilla, ya que la consola está diseñada para ser usada con otros servicios *web*.

- Control total: se tiene el control sobre todas las instancias, con acceso a los usuarios de raíz de todas las instancias, pudiendo iniciar, detener, reiniciar y finalizar cualquier instancia de forma remota.
- Automatización: por medio de *scripts* definidos y utilizando las herramientas API proporcionadas por *Amazon*, se puede automatizar la administración del entorno y las aplicaciones en la nube.

### Desventajas

- No hay métricas para medir todos los parámetros de rendimiento: es decir, que el proveedor de recursos de la nube, puede jugar con ellos quitando capacidad de proceso para dársela a otros sin que la empresa se entere.
- Subir código compilado a un servidor ajeno: cualquier persona que tenga acceso al servidor puede descompilar el código.
- El almacenamiento no es 100 por ciento fiable: esto muchas veces no se logra en un 100 por ciento dentro de la empresa, muy seguramente eso no pasa tampoco en las empresas que manejan la nube.
- Pérdida de correos: en algún momento le ha pasado a más de algún proveedor de *cloud*, tal como *Google* y *Microsoft*, que han perdido correos de sus clientes, tal como lo indica un post que dice: *Storing Your Data In The Cloud – Google Deletes Emails* (<http://www.techieinsider.com/news/6625>)



### **3.9. Resumen de los pasos a seguir y herramientas a utilizar adecuadamente para el proceso de automatización en la construcción de aplicaciones y sus entornos en la nube**

A continuación se hará en un ejemplo práctico para demostrar cómo crear paso a paso, un entorno de aplicaciones y su respectiva aplicación en la nube, utilizando la plataforma de servicios de nube de *Amazon*, y utilizando la aplicación de código abierto *SugarCRM* montada en dicha plataforma tecnológica.

La instancia que se utilizará en la nube tiene como sistema operativo *Windows 2008 Server*, y las herramientas de código abierto: base de datos relación al *MySQL*, lenguaje de programación *PHP* y *Apache* como *Web Server*. La aplicación *SugarCRM* que utilizaremos en este ejemplo, está desarrollada para funcionar con las anteriores herramientas instaladas en la instancia.

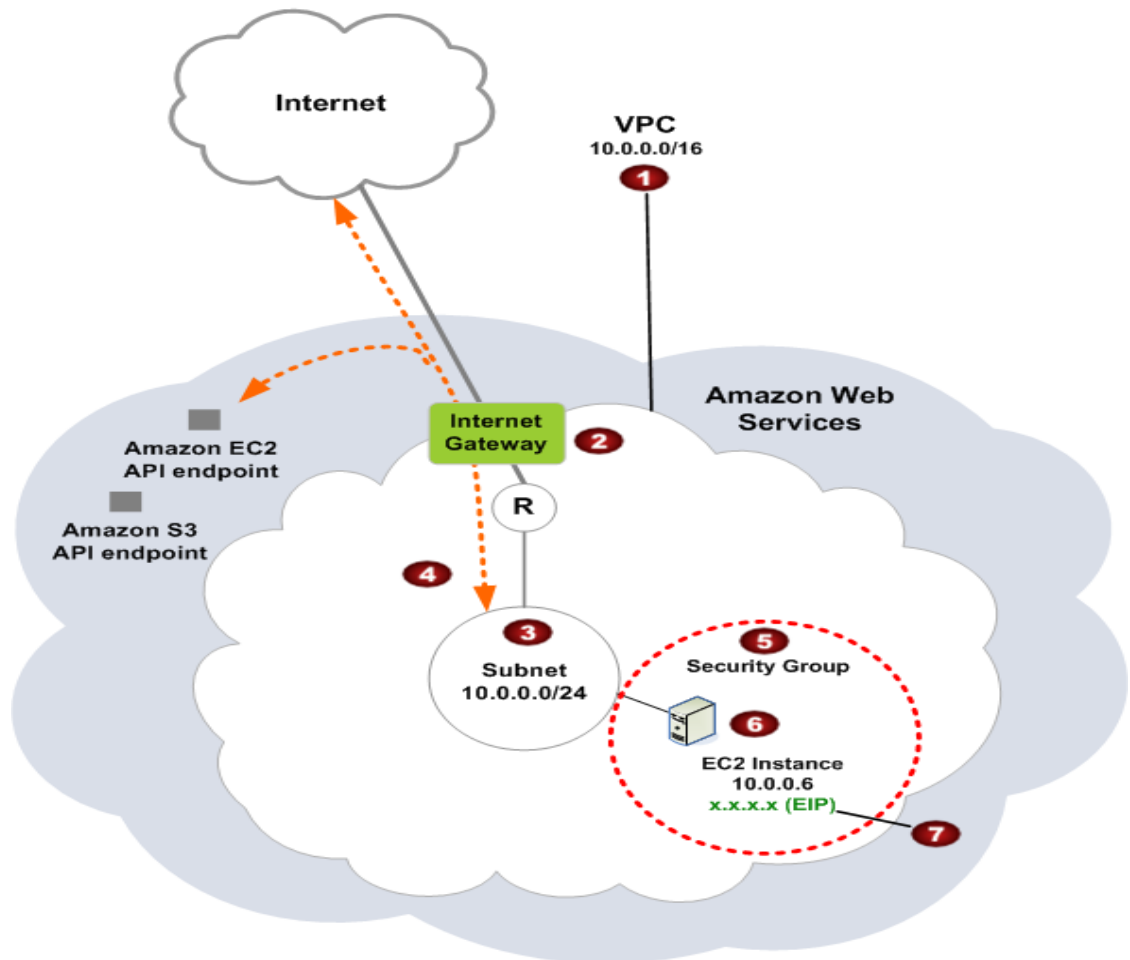
La configuración para el funcionamiento de la aplicación está contenida en una instancia de *Amazon*. El entorno estará dentro de en una nube privada que es la primera parte que construiremos, para poder montar nuestra aplicación

#### Construcción de la nube privada

A continuación se enumera un resumen de las tareas que se ejecutan en la herramienta *Amazon Virtual Private Cloud*, para crear una nube privada.

- Crear una nube privada que es una parte aislada de la nube.
- Crear y adjuntar un portal de internet (*Internet Gateway*) de la nube privada, que conecta directamente a internet y proporciona acceso a otros recursos como *Amazon Simple Storage Service* (Amazon S3).
- Crear una subred de *Amazon VPC*, que es un segmento de una nube privada que tiene un rango de direcciones IP que se pueden iniciar dentro de de las instancias.
- Configurar el enrutamiento en la nube privada para permitir que el tráfico fluya entre la subred y el *internet*.
- Establecer un grupo de seguridad para controlar el tráfico entrante y saliente de las instancias que se inicien.
- Iniciar una instancia en la subred (en este caso es una instancia de *Windows*), la instancia tiene una dirección IP privada del rango de direcciones de la subred.
- Asignar una dirección IP elástica a la instancia. Una dirección IP elástica es una dirección estática y pública que se puede asignar a una instancia en la nube privada. Esta asignación da a la instancia una dirección IP pública, además de su dirección privada. Una instancia en la nube privada para ser accedida desde internet, debe tener una dirección IP elástica.

Figura 24. Tareas o pasos para crear una nube privada



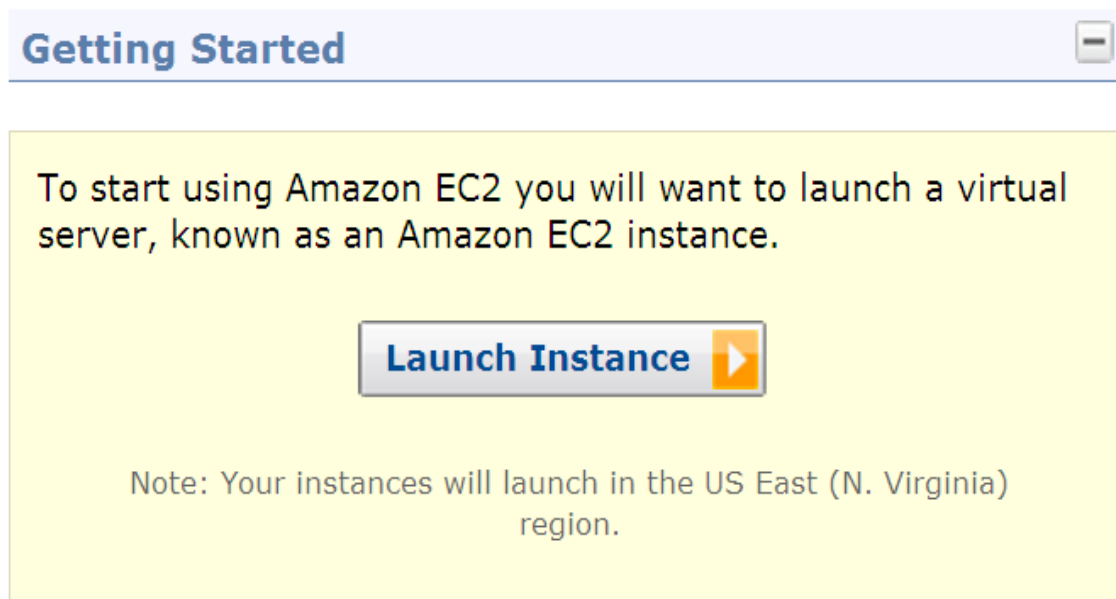
Fuente: elaboración propia.

### Iniciar la instancia

Para iniciar una instancia, lo primero que hay que hacer, es ingresar con el usuario y clave a la consola o tablero de instrumentos de *Amazon Web Services* (AWS), posteriormente, ingresar a la sección que contiene el servicio y herramientas para administrar las instancias llamado *Elastic Compute Cloud*

(EC2). Una vez aquí, se presiona el botón iniciar una instancia, tal como se indica a continuación.

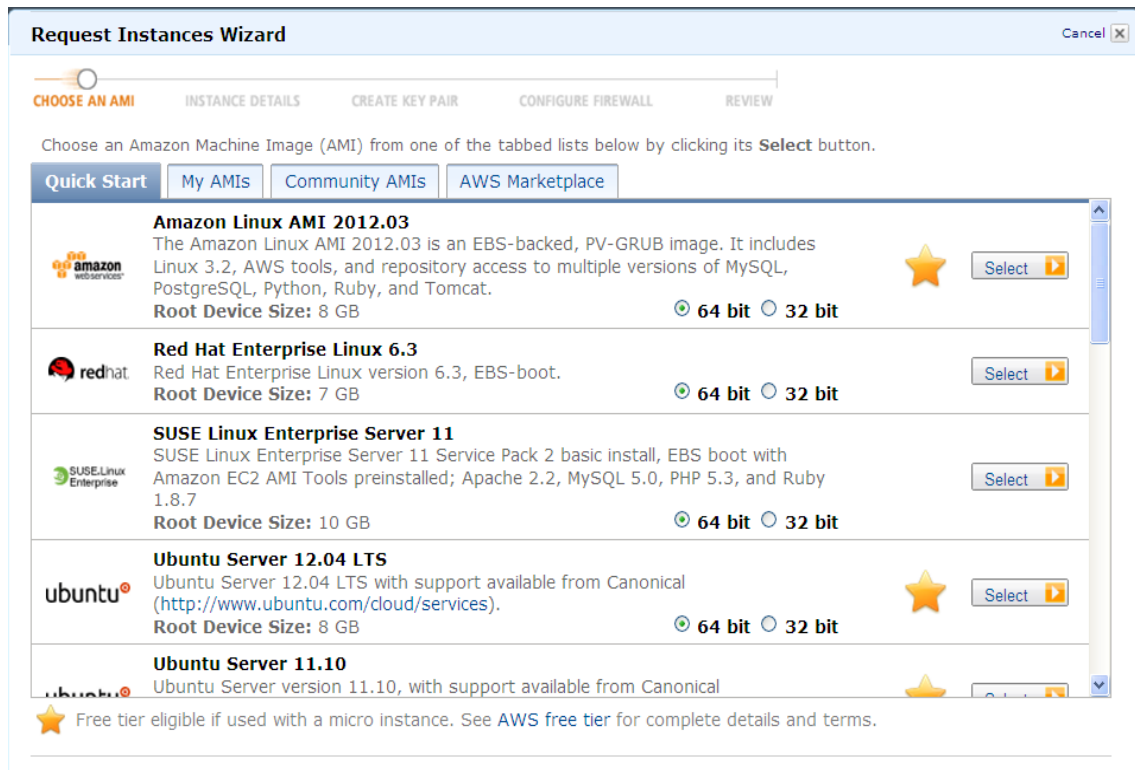
Figura 25. Inicio de aplicación



Fuente elaboración propia.

El primer *tab* de la página, como se muestra en la siguiente figura, despliega una lista de AMIs básicas que contienen toda la información necesaria para crear una nueva instancia de un servidor, por ejemplo una AMI, puede contener todo el *software* para actuar como un servidor web (Linux, Apache y el Sitio *Web*).

Figura 26. Lista básica

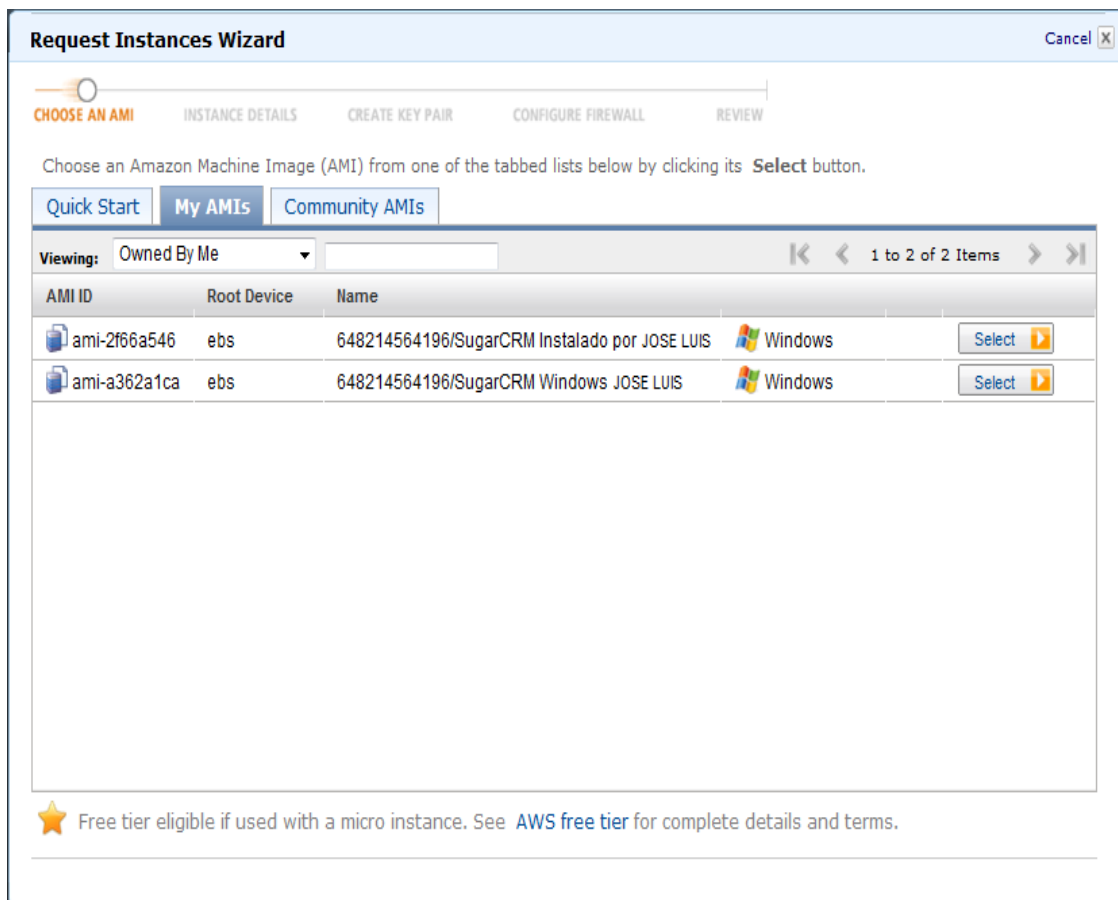


Fuente: elaboración propia.

Se debe seleccionar una AMI, que en este caso, será una que tenga instalado *Windows 2008 server* y los programas necesarios para ejecutar la aplicación *SugarCRM*. Para el presente ejemplo, ya se tiene definida una AMI, que previamente se configuró, con todos los programas necesarios (*Windows 2008 Server*, PHP, MySQL, Apache y la aplicación *SugarCRM*). Como bien se sabe, se pueden tener definidas AMIs personalizadas con todos los programas instalados y configuraciones que se necesiten en determinado momento, es decir, estas AMIs son plantillas pre- configuradas, a partir de las cuales se inician nuevas instancias.

Para iniciar una instancia, se selecciona el tab My AMIs, tal como se muestra en la siguiente figura, el cual despliega todas las AMIs que están definidas y pre configuradas en mi cuenta de Amazon EC2, y que tienen instalados programas particulares, en este caso para el ejemplo, se utiliza la AMI llamada ami-2f66a546/648214564196/SugarCRM Instalada por José Luis, que contiene la configuración mencionada anteriormente y que se necesita para correr la aplicación *SugarCRM*.

Figura 27. Selección de una AMI



Fuente: elaboración propia.

Antes de iniciar la instancia, se incluyen algunas configuraciones adicionales tales como, el número y tamaño de instancias a iniciar; si la instancia estará dentro de una nube privada (VPC); si se habilitará el monitoreo detallado de la instancia, etc.

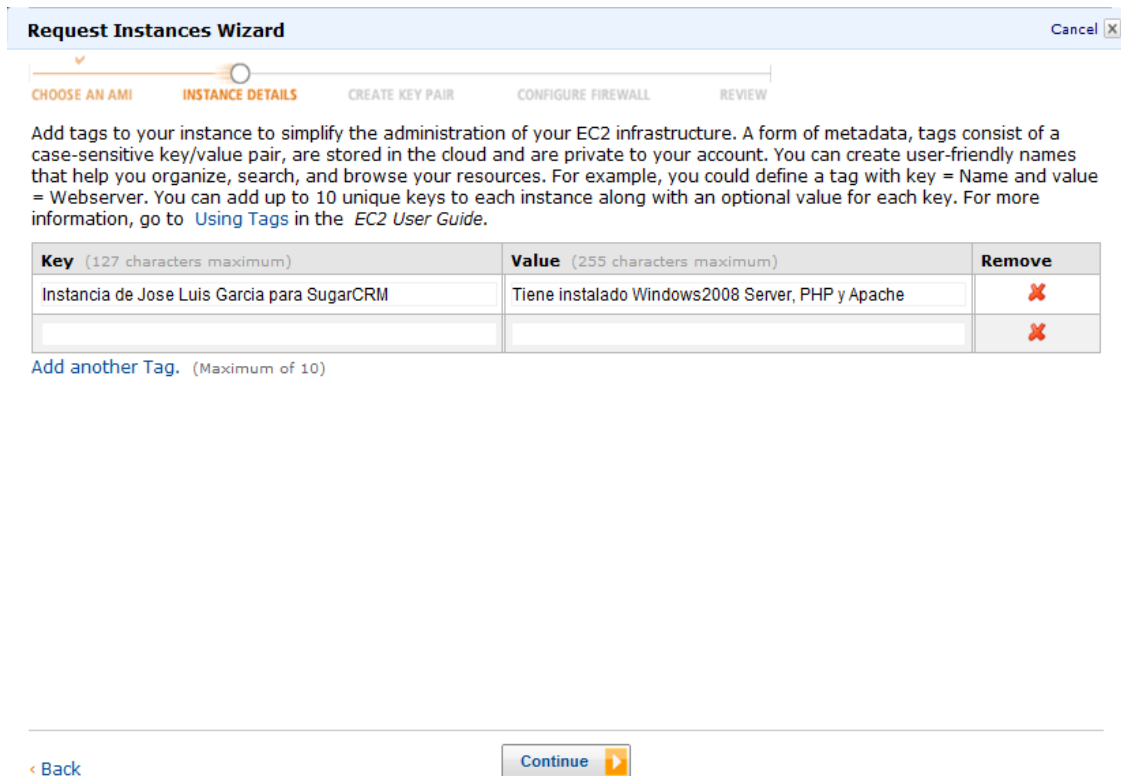
Figura 28. Configuraciones adicionales

The screenshot shows the 'Request Instances Wizard' interface. At the top, there is a progress bar with five steps: 'CHOOSE AN AMI', 'INSTANCE DETAILS', 'CREATE KEY PAIR', 'CONFIGURE FIREWALL', and 'REVIEW'. The 'INSTANCE DETAILS' step is currently active. Below the progress bar, the 'Number of Instances' is set to 1 and the 'Availability Zone' is set to us-east-1a. The 'Advanced Instance Options' section includes: 'Kernel ID' and 'RAM Disk ID' both set to 'Use Default'; 'Monitoring' checked with a note that additional charges will apply; 'User Data' containing a text area with the description 'Instancia que contiene la configuración adecuada para ejecutar la aplicación de código abierto SugarCRM. Dicha aplicación corre en esta instancia que tiene instalado Windows 2008 Server, PHP, Apache y la aplicación SugarCRM.' and a checkbox for 'base64 encoded'; 'Termination Protection' unchecked with a note 'Prevention against accidental termination.'; and 'Shutdown Behavior' set to 'Stop' with a note 'Choose the behavior when the instance is shutdown from within the instance.' and a tooltip that says 'The behavior of the instance for an instance initiated shutdown.' At the bottom, there are 'Back' and 'Continue' buttons.

Fuente: elaboración propia.

Posteriormente, se agregan etiquetas a la instancia, para poder administrar de mejor manera la infraestructura de EC2. Esto sirve para organizar, navegar y buscar los recursos en la nube. Se pueden agregar hasta 10 etiquetas con información importante de cada una de las instancias que se van a iniciar, tal como lo muestra la siguiente figura.

Figura 29. Adición de etiquetas



Fuente: elaboración propia.

Luego se ingresan las credenciales de seguridad, como se indica en la siguiente figura, en este caso, llamadas Jose Luis\_KeyPair. Éstas credenciales fueron previamente definidas, generadas y almacenadas en un lugar seguro dentro de la computadora. Las credenciales se utilizarán en el momento que se haga la conexión hacia la instancia, y son similares a un *password*, que se utilizan para conectarse de forma segura a la instancia después que la misma se ha iniciado y está corriendo.



Figura 30. Ingreso de credenciales de seguridad

**Request Instances Wizard** Cancel X

CHOOSE AN AMI    INSTANCE DETAILS    **CREATE KEY PAIR**    CONFIGURE FIREWALL    REVIEW

Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

**Choose from your existing Key Pairs**

Your existing Key Pairs\*: JoseLuis\_KeyPair ▾

Create a new Key Pair

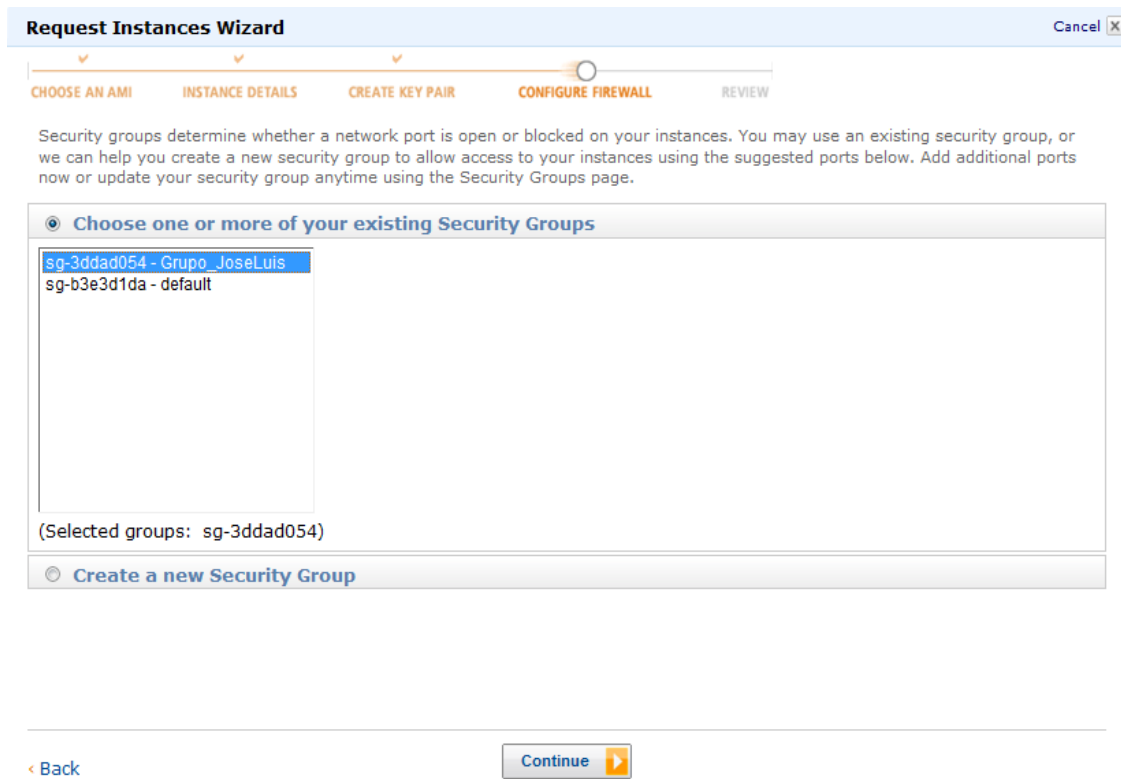
Proceed without a Key Pair

[< Back](#)    [Continue >](#)

Fuente: elaboración propia.

Una vez que se han indicado las credenciales (*Key Pair*), se debe indicar un grupo de seguridad, que es el que define las reglas del *firewall* para las instancias. Estas reglas especifican qué tráfico de red entrante debe ser entregado a la instancia (por ejemplo, aceptar el tráfico *web* en el puerto 80). El resto del tráfico se ignora. Estas reglas para un grupo pueden ser modificadas en cualquier momento y se aplican automáticamente a todas las instancias. En este ejemplo, se selecciona el grupo de seguridad llamado Grupo\_JoseLuis, que tiene configuradas las reglas del *firewall* para la instancia que se va a utilizar. La siguiente figura muestra los grupos de seguridad.

Figura 31. Elección de grupo de seguridad



Fuente: elaboración propia.

Previo al último paso que es iniciar la instancia, se despliega una ventana que contiene toda la información de la instancia que se va a iniciar. Básicamente la información contiene el nombre de la instancia, el sistema operativo, el nombre de la nube privada donde se inicia la instancia, si aplica; el tipo de instancia, si la instancia será monitoreada detalladamente, las credenciales de seguridad, el grupo de seguridad que maneja las reglas del *firewall* para la instancia, etc. Una vez revisada la importante información de la configuración de la instancia, se presiona el botón *Launch* y se comienza el proceso de inicio de la misma. La siguiente figura muestra toda la información importante previa a iniciar la instancia.

Figura 32. Resumen de ingreso de datos

**Request Instances Wizard** Cancel

CHOOSE AN AMI   INSTANCE DETAILS   CREATE KEY PAIR   CONFIGURE FIREWALL   **REVIEW**

Please review the information below, then click **Launch**.

**AMI:** Windows AMI ID ami-2f66a546 (i386) [Edit AMI](#)

**Number of Instances:** 1

**VPC ID:** vpc-bf6c6fd6

**VPC Subnet:** subnet-b56c6fdc (10.0.0.0/24)

**Availability Zone:** No Preference

**Instance Type:** Small (m1.small)

**Instance Class:** On Demand [Edit Instance Details](#)

---

**Monitoring:** Disabled      **Termination Protection:** Disabled

**Tenancy:** Default

**Kernel ID:** Use Default      **Shutdown Behavior:** Stop

**RAM Disk ID:** Use Default

**IP Address:** Automatically Assigned

**User Data:** Instancia que conti... [Edit Advanced Details](#)

---

**Key Pair Name:** JoseLuis\_KeyPair [Edit Key Pair](#)

---

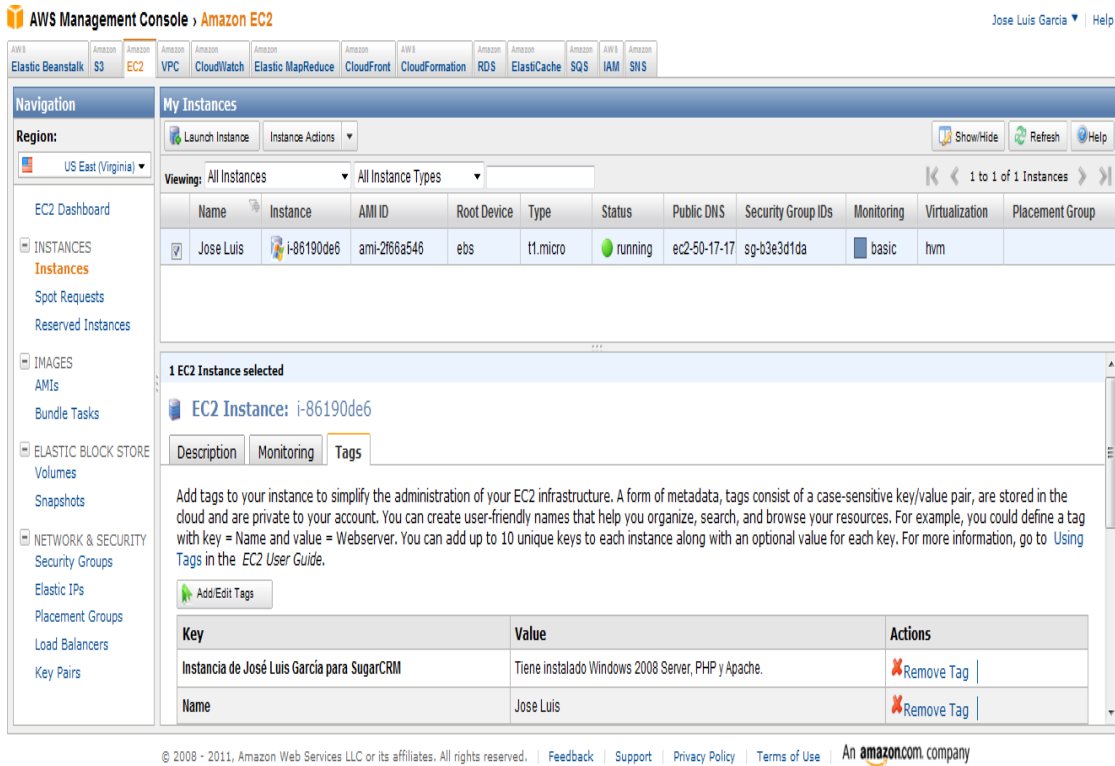
**Security Group(s):** sg-70809d1c [Edit Firewall](#)

[< Back](#)      **Launch**

Fuente: elaboración propia.

Cuando la instancia se ha iniciado, se puede ir a la consola para consultar las instancias y ver el estatus en el que se encuentran. El estatus de la instancia que se acaba de iniciar, está en *running*, lo que indica que se puede hacer una conexión a dicha instancia, y trabajar con ella como si fuera cualquier servidor de *Windows* normal. La siguiente pantalla muestra la instancia que se acaba de iniciar.

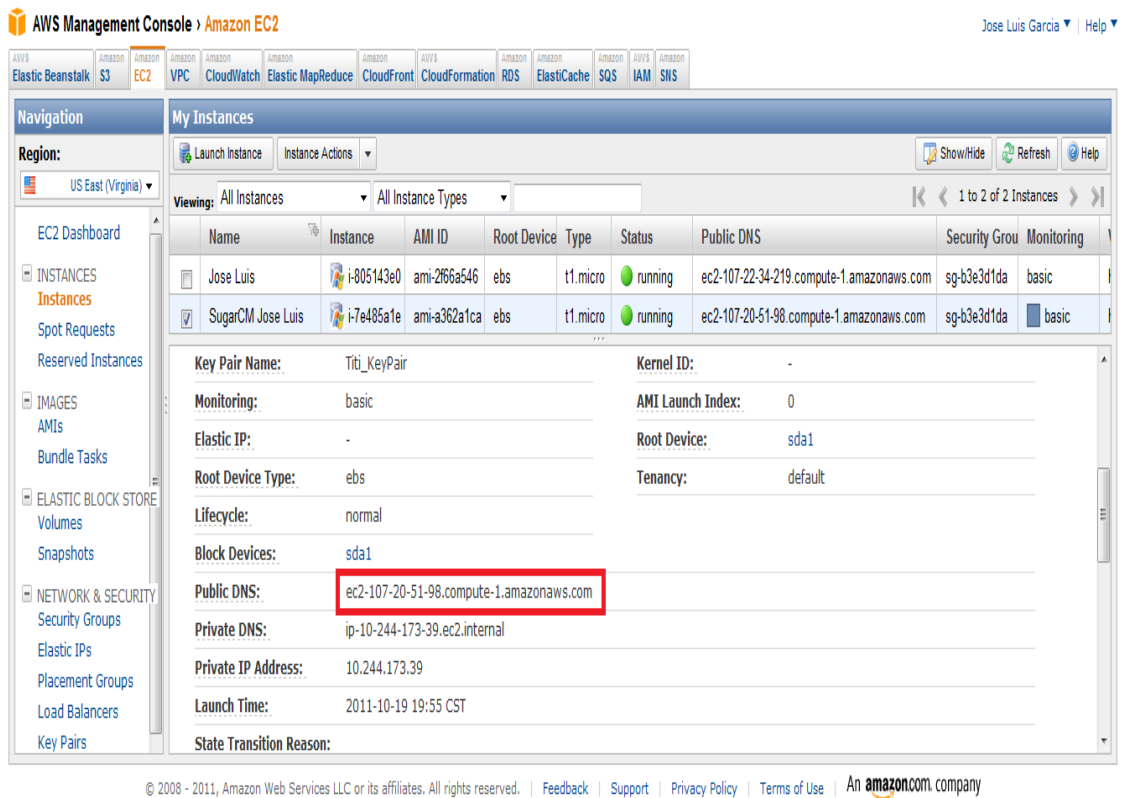
Figura 33. **Estatus de consola**



Fuente: elaboración propia.

Al conectarse a la instancia, en este caso, con sistema operativo *Windows2008 Server*, se hace utilizando el escritorio remoto de *Windows (Remote Desktop)*. El nombre del equipo es: *ec2-107-20-51-98.compute-1.amazonaws.com*, que es el que está definido en *Public DNS*, el usuario es administrador. La siguiente pantalla muestra el nombre del equipo al que se hará la conexión.

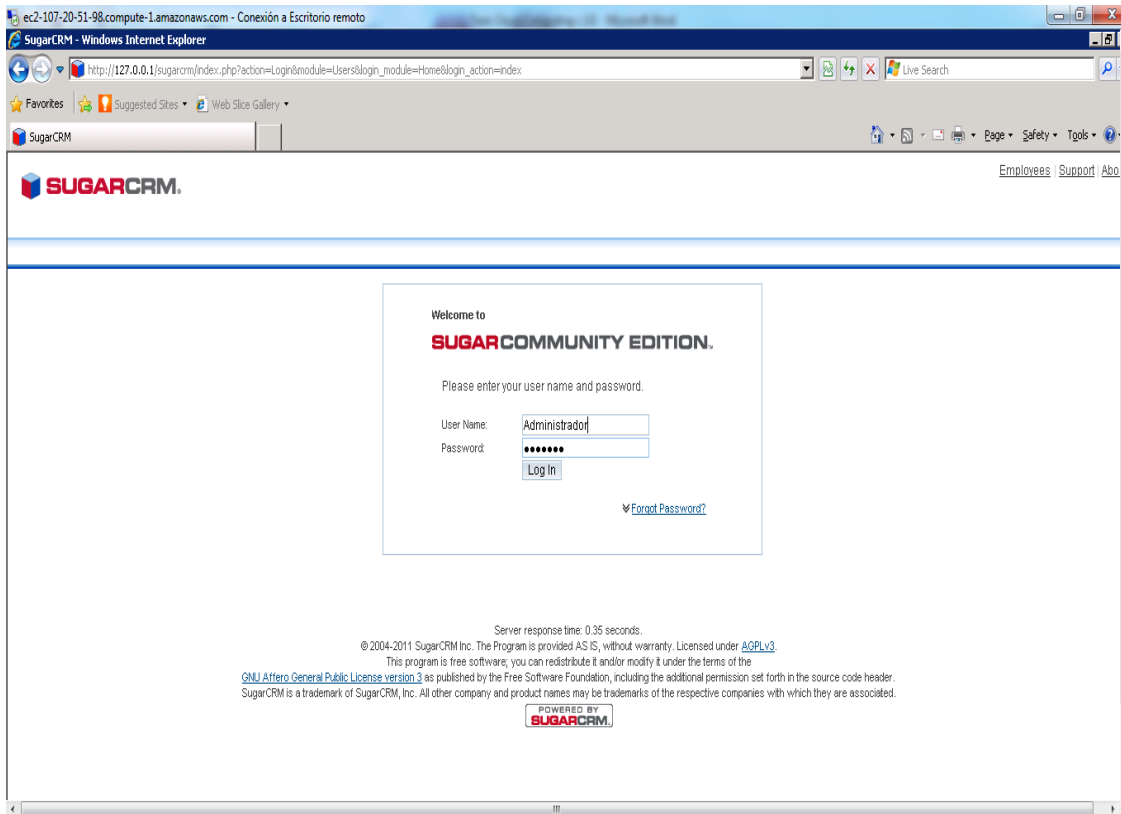
Figura 34. Conexión remota



Fuente: elaboración propia.

Una vez conectado al servidor, se puede acceder a la aplicación instalada previamente junto con el resto de programas. Al entrar a la aplicación *SugarCRM*, se verá la siguiente pantalla de inicio que permitirá navegar a través de la aplicación. Observe que en la esquina superior izquierda aparece el nombre del equipo al que se está conectado, *ec2-107-20-51-98.compute-1.amazonaws.com*

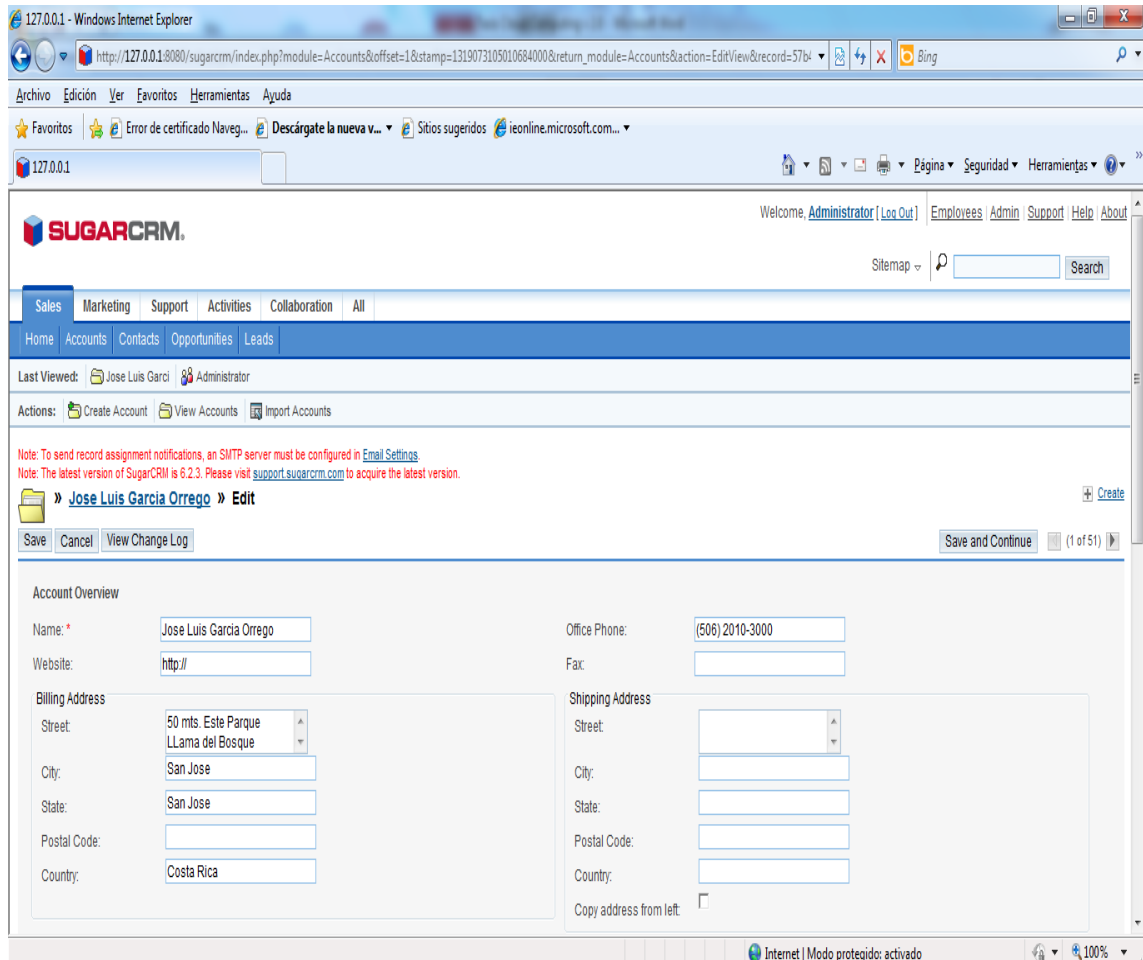
Figura 35. Acceso a la aplicación instalada



Fuente: elaboración propia.

A continuación se muestran unas pantallas que contienen la aplicación corriendo en la instancia que se acaba de iniciar. La siguiente figura, muestra los datos para crear una cuenta de ventas en *SugarCRM*. Se creó una cuenta de ejemplo para José Luis García Orrego.

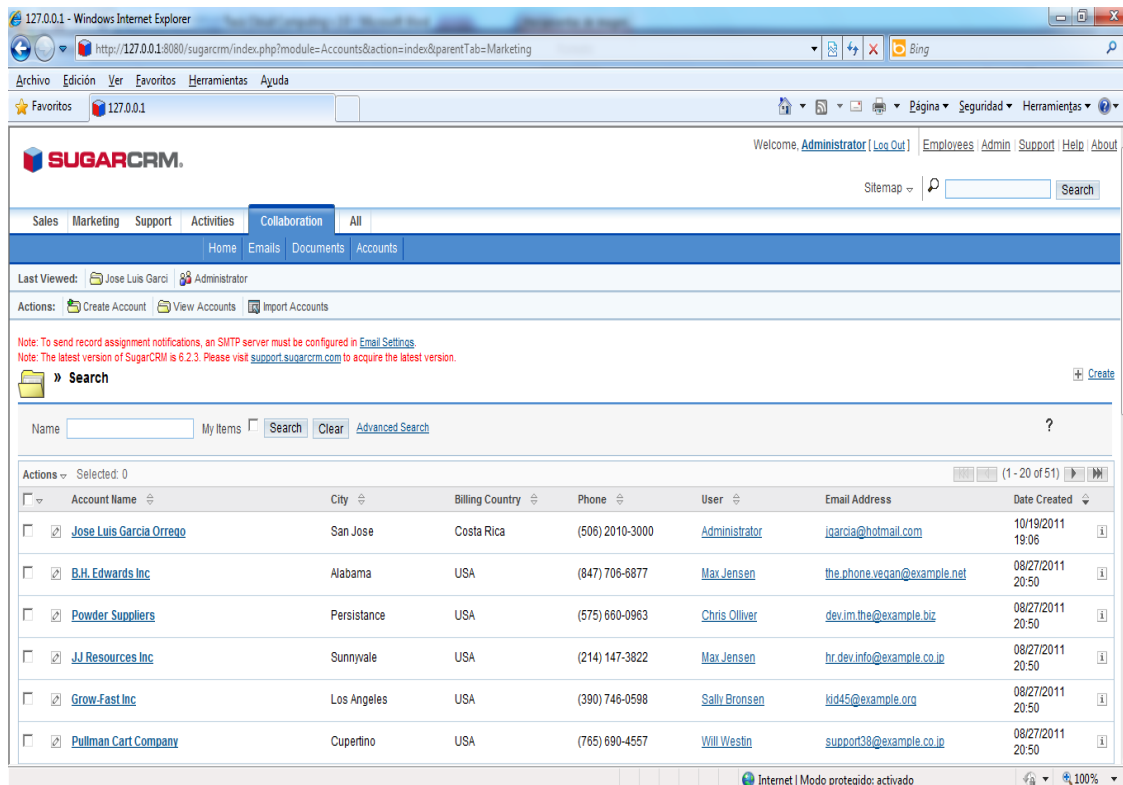
Figura 36. Corriendo aplicación



Fuente: elaboración propia.

La siguiente pantalla, muestra la información de la nueva cuenta creada, así como otras cuentas que existían previamente en la base de datos de la aplicación.

Figura 37. Información de cuentas



Fuente: elaboración propia.

La aplicación *SugarCRM* tiene todas las características propias de un sistema que maneja la administración de la relación con los clientes (CRM por sus siglas en Inglés), sin embargo, para efectos de este proyecto de tesis, no se entrará en más detalle en el funcionamiento de ésta aplicación, ya que el objetivo básicamente es mostrar cómo se puede soportar dicha aplicación en la plataforma tecnológica de *Amazon*.

En esta sección, se presentó un resumen de los pasos a seguir y las herramientas a utilizar para poder crear una nube privada, iniciar una instancia en base a una AMI, conectarse a dicha instancia y ejecutar una aplicación en la instancia.



Los pasos para crear una instancia, y configurarla con los programas necesarios para correr la aplicación *SugarCRM*, no fueron incluidos en este ejemplo, ya que la instalación de *software* en las instancias no es relevante para este ejemplo. Sin embargo, una vez configurada la instancia e instalado el *software* necesario, se creó una AMI que fue la base de para el presente ejemplo. Esta AMI es parte fundamental de la automatización, ya que cada vez que se necesita crear una o varias instancias nuevas, se hace en base a dicha AMI. Las instancias al momento de iniciarse, automáticamente heredaron las características de la AMI, lo que evita la reconfiguración manual para correr la aplicación *SugarCRM*.

En el ejemplo, no se incluye cómo se crearon las credenciales y los grupos de seguridad, la forma para definir alarmas en base a métricas para escalar e iniciar nuevas instancias, balanceo de cargas, creación de nuevos volúmenes de almacenamiento por llegar al límite de espacio en disco, envío de correos electrónicos, etc., sin embargo, son tareas muy sencillas que se ejecutan en cada una de las herramientas encargadas de brindar éste servicio y con las que cuenta Amazon. Estos servicios se pueden combinar perfectamente con todo el resto de herramientas y servicios de Amazon, para ofrecer un control completo sobre todos los recursos informáticos que se necesiten.

En el apéndice 1, se presenta un ejemplo de automatización en la construcción de un entorno de aplicaciones e instalación de programas, que hace básicamente lo mismo que el ejemplo anterior, solamente que de forma automática, ejecutando plantillas (scripts), previamente definidas, con las capacidades informáticas que se necesitan en la nube.



## CONCLUSIONES

1. El uso de las herramientas de computación en nube, permiten la facilidad para iniciar y finalizar de forma sencilla entornos completos en la nube.
2. No toda la información que está almacenada en la nube esta cien por ciento protegida en cuanto a seguridad y confidencialidad de los datos.
3. Se comprobó que la aplicación de código abierto llamada *SugarCRM* funcionó correctamente en la infraestructura que se construyó en la nube con las herramientas de *Amazon EC2*.
4. La capacidad de escalar y disminuir en la nube, se hace de manera muy sencilla, ya que en el ejemplo práctico, se utilizó una instancia y posteriormente se agregó una instancia adicional con las mismas características, dedicando de esta manera, más recursos a la nube sin necesidad de realizar ajustes de ningún tipo.
5. Hay muchos aspectos arquitectónicos y conceptuales que deben tenerse en cuenta a la hora de hacer una migración de una aplicación *web* a la nube, sin embargo, las ventajas de disponer de una infraestructura con alta capacidad de ajuste de dimensiones y tolerante a fallos, compensa los esfuerzos de migrar a la nube.

6. Las empresas pequeñas o nuevas empresas, no necesitan hacer inversiones grandes en tecnología, sino que a medida que la van necesitando, la van usando y pagando por ella, lo cual representa una gran flexibilidad, al punto que no se requiere ningún contrato previo con el proveedor del servicio, sino que simplemente se hace la adquisición por la *web*, utilizando una tarjeta de crédito.
  
7. Las herramientas y servicios que ofrece *Amazon* para la computación en nube, tienen una manera muy sencilla de administrar y automatizar la infraestructura y las aplicaciones. Además, por medio de la línea de comandos (API de *Amazon*), se pueden ejecutar *scripts* previamente definidos, con las capacidades informáticas que se necesitan en la infraestructura de la nube, de forma programada y automática.
  
8. Todos los servicios que presta Amazon EC2 son muy importantes para la automatización, sin embargo, la utilización de AMIs, es la base que permite evitar o eliminar la constante reconfiguración estática de la infraestructura.

## RECOMENDACIONES

1. Se debería automatizar lo más posible las diferentes configuraciones de la infraestructura y sus aplicaciones por medio de *scripts* que se ejecuten en lote o con tareas *cron*, para evitar la constante reconfiguración manual.
2. Se deberían evaluar las diferentes empresas que prestan el servicio de computación en nube, en cuanto a beneficios de solidez, calidad, precio, herramientas y servicios, que faciliten la administración y automatización de infraestructura y las aplicaciones.
3. Sería necesario conocer el detalle de los costos actuales y tiempos de implementación en que está incurriendo la empresa por servicios como el de procesamiento, almacenamiento y comunicaciones, para tomar decisiones acertadas con respecto al uso de los servicios en la nube para automatización y administración.
4. Se debería comparar la facilidad con la que se puede construir un entorno de aplicaciones en la nube, contra la construcción del mismo entorno de manera física, para poder tomar decisiones en cuanto a, qué es lo más conveniente para la empresa dependiendo de sus necesidades.



## BIBLIOGRAFÍA

1. *Amazon Web Services*. [en línea] <http://aws.amazon.com/es/> [Consulta: 10 de agosto de 2011].
2. BARR Jeff; NARIN, Attila; VARIA, Jinesh. *Building Fault Tolerant Applications on AWS*. [en línea] <http://aws.amazon.com/es/whitepapers/> [Consulta: 25 de septiembre de 2012].
3. CHEE, Brian J.S.; FRANKLIN, Curtis Jr. *Cloud Computing. Technologies and Strategies of the Ubiquitous Data Center*. Boca Ratón, USA. Taylor & Francis Group. 2010. 263 p.
4. LURIE, M. *The Federation: database interoperability*. [en línea] <http://www.ibm.com/developerworks/data/library> [Consulta: 23 de abril de 2003].
5. MARKS, Erick A.; LOZANO, Bob. *Executive's guide to cloud computing*. Hoboken, New Jersey, USA. John Wiley & Sons, Inc. 2010. 302 p.
6. OBASANJO D. *Building scalable databases: pros and cons of various database sharding schemes*. [en línea] <http://www.25hoursaday.com/weblog/2009/01/16/BuildingScalableDatabasesProsAndConsOfVariousDatabaseShardingSchemes.aspx>. [Consulta: 16 de enero de 2009].

7. SCHLOSSNAGLE T. *Scalable Internet Architectures*. Sams Publishing. [en línea] [http://sqs-public-images.s3.amazonaws.com/Building\\_Scalabale\\_EC2\\_applications\\_with\\_SQS2.pdf](http://sqs-public-images.s3.amazonaws.com/Building_Scalabale_EC2_applications_with_SQS2.pdf). [Consulta: 31 de Julio de 2006].
8. *SugarCRM. Aplicación de código abierto*. [en línea] <http://www.sugarcrm.com/crm/> [Consulta: 15 de agosto de 2011].
9. TAVIS, Matt. *Alojamiento de aplicaciones Web en la nube*. [en línea] <http://aws.amazon.com/es/whitepapers/web-application-hosting-best-practices/> [Consulta: 5 de octubre de 2011].
10. VARIA, Jinesh. *Architecting for the Cloud: best practices*. [en línea] <http://aws.amazon.com/es/whitepapers/architecting-for-the-aws-cloud-best-practices/> [Consulta: 5 de septiembre de 2011].
11. VMWare and Cloud Computing. *An Evolutionary Approach to an IT Revolution*. Palo Alto, USA. VMware, Inc. 2010. 20 p.



## APÉNDICE

### Ejemplos de automatización en base a plantillas (*scripts*)

Los mismos pasos que se mencionaron anteriormente de forma manual con unos cuantos clics en el ejemplo práctico, se pueden ejecutar de forma automática, corriendo *scripts* previamente definidos con las capacidades informáticas que se necesitan en la infraestructura de la nube. Los *scripts* se ejecutan con las herramientas API o herramientas de línea de comandos proporcionadas por *Amazon*.

Esto permite crear los entornos de aplicaciones y desplegar las aplicaciones automáticamente en la nube, por medio de procesos en lote o cron. A continuación se presentan dos ejemplos de *scripts* para crear entornos en la nube.

Script de ejemplo que muestra cómo crear una instancia EC2 con sistema operativo Linux en la nube.

```
{
  "Description" : "Crea una instancia EC2 corriendo en base a una AMI con Linux 32
bit"
  "Parameters" : {
    "KeyPair" : {
      "Description" : "El EC2 Key Pair permite acceso via SSH a la instancia",
      "Type" : "String"
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
```

```

    "KeyName" : { "Ref" : "KeyPair" },
    "ImageId" : "ami-75g0061f"
  }
},
"Outputs" : {
  "InstancedId" : {
    "Description" : "El Id de la Instancia EC2 recientemente creada",
    "Value" : { "Ref" : "Ec2Instance" }
  },
  "AWSTemplateFormatVersion" : "2011-09-09"
}

```

El siguiente *script*, contiene la creación de una instancia tipo m1.small, con una base de datos MYSQL, a la cual se le asigna un usuario y *password*, con lenguaje de programación PHP, envío de correo electrónico a jose.garcia@amazon.com, con habilitación de puertos, con definición de *firewall*, con definición de grupos de seguridad, definición alarmas para auto escala, etc.

Script para crear recursos en la nube de forma automática.

```

{
  "AWSTemplateFormatVersion": "2011-09-09",
  "Description": "Ejemplo de aplicación PHP HelloWorld que conecta a una instancia con servicio de base de datos, Amazon RelationalDatabase y despliega información sobre el servidor web.",
  "Parameters": {
    "DatabaseName": {
      "Default": "EjemploPHP",
      "Type": "String",
      "Description": "Nombre base de datos de prueba",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription": "debe empezar con una letra y contener solo caracteres alfanumericos."
    },
    "DatabaseUser": {
      "Default": "admin",

```

```

    "NoEcho": "true",
    "Type": "String",
    "Description" : "Nombre decuenta admin de base de datos de prueba",
    "MinLength": "1",
    "MaxLength": "16",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
    "ConstraintDescription" : " debe empezar con una letra y contener solo caracteres
alfanumericos."
},
    "DatabasePassword": {
        "Default": "admin",
        "NoEcho": "true",
    "Type": "String",
    "Description" : " Password decuenta admin de base de datos de prueba",
    "MinLength": "1",
    "MaxLength": "41",
    "AllowedPattern" : "[a-zA-Z0-9]*",
    "ConstraintDescription" : "debe contener solo caracteres alfanumericos."
},
    "InstanceType": {
        "Default": "m1.small",
        "Type": "String",
    "Description" : "Tipo de instancia EC2 para el servidor web"
},
    "OperatorEmail": {
        "Default": "jose.garcia@amazon.com",
    "Description": "Direccion Email para notificar si hay problemas operacionales",
    "Type": "String"
    },
    "WebServerPort": {
        "Default": "8888",
        "Type": "Number",
    "MinValue": "1",
    "MaxValue": "65535",
    "Description" : "Puerto TCP/IP para el servidor web"
},
    "DatabasePort": {
        "Default": "3306",
        "Type": "Number",
        "MinValue": "1150",
    "MaxValue": "65535",
    "Description" : "Puerto TCP/IP para la base de datos RDS"
}
},
    "Resources": {
        "CPUAlarmHigh": {
            "Properties": {

```

```

    "EvaluationPeriods": "1",
    "Statistic": "Average",
    "Threshold": "10",
    "AlarmDescription": "Alarma si el uso del CPU está muy alto o si desaparecen las
métricas indicando que la instancia esta caída",
    "Period": "60",
    "AlarmActions": [
      {
        "Ref": "AlarmTopic"
      }
    ],
    "Namespace": "AWS/EC2",
    "InsufficientDataActions": [
      {
        "Ref": "AlarmTopic"
      }
    ],
    "Dimensions": [
      {
        "Name": "AutoScalingGroupName",
        "Value": {
          "Ref": "WebServerGroup"
        }
      }
    ],
    "ComparisonOperator": "GreaterThanThreshold",
    "MetricName": "CPUUtilization"
  },
  "Type": "AWS::CloudWatch::Alarm"
},
"TooManyUnhealthyHostsAlarm": {
  "Properties": {
    "EvaluationPeriods": "1",
    "Statistic": "Average",
    "Threshold": "0",
    "AlarmDescription": "Alarma que indica si hay muchas instancias no saludables.",
    "Period": "60",
    "AlarmActions": [
      {
        "Ref": "AlarmTopic"
      }
    ],
    "Namespace": "AWS/ELB",
    "InsufficientDataActions": [
      {
        "Ref": "AlarmTopic"
      }
    ]
  }
}

```

```

    ],
    "Dimensions": [
      {
        "Name": "LoadBalancerName",
        "Value": {
          "Ref": "ElasticLoadBalancer"
        }
      }
    ],
    "ComparisonOperator": "GreaterThanOrEqualTo",
    "MetricName": "UnHealthyHostCount"
  },
  "Type": "AWS::CloudWatch::Alarm"
},
"EC2SecurityGroup": {
  "Properties": {
    "SecurityGroupIngress": [
      {
        "FromPort": "22",
        "CidrIp": "0.0.0.0/0",
        "ToPort": "22",
        "IpProtocol": "tcp"
      },
      {
        "FromPort": {
          "Ref": "WebServerPort"
        },
        "CidrIp": "0.0.0.0/0",
        "ToPort": {
          "Ref": "WebServerPort"
        },
        "IpProtocol": "tcp"
      }
    ]
  },
  "GroupDescription": "HTTP and SSH access"
},
"Type": "AWS::EC2::SecurityGroup"
},
"RequestLatencyAlarmHigh": {
  "Properties": {
    "EvaluationPeriods": "1",
    "Statistic": "Average",
    "Threshold": "1",
    "AlarmDescription": "Alarma si no hay solicitudes que llegan",
    "Period": "60",
    "AlarmActions": [
      {

```

```

    "Ref": "AlarmTopic"
  }
],
"Namespace": "AWS/ELB",
"InsufficientDataActions": [
  {
    "Ref": "AlarmTopic"
  }
],
"Dimensions": [
  {
    "Name": "LoadBalancerName",
    "Value": {
      "Ref": "ElasticLoadBalancer"
    }
  }
],
"ComparisonOperator": "GreaterThanThreshold",
"MetricName": "Latency"
},
"Type": "AWS::CloudWatch::Alarm"
},
"ElasticLoadBalancer": {
  "Properties": {
    "Listeners": [
      {
        "InstancePort": {
          "Ref": "WebServerPort"
        },
        "PolicyNames": [
          "p1"
        ],
        "Protocol": "HTTP",
        "LoadBalancerPort": "80"
      }
    ],
    "HealthCheck": {
      "HealthyThreshold": "2",
      "Timeout": "5",
      "Interval": "10",
      "UnhealthyThreshold": "5",
      "Target": {
        "Fn::Join": [
          "",
          [
            "HTTP:",
            {

```

```

        "Ref": "WebServerPort"
      },
    ],
  ],
},
"AvailabilityZones": [
  {
    "Fn::Join": [
      "",
      [
        {
          "Ref": "AWS::Region"
        },
        "a"
      ]
    ]
  }
],
"LBCookieStickinessPolicy": [
  {
    "CookieExpirationPeriod": "30",
    "PolicyName": "p1"
  }
],
"Type": "AWS::ElasticLoadBalancing::LoadBalancer"
},
"DBSecurityGroup": {
  "Properties": {
    "DBSecurityGroupIngress": {
      "EC2SecurityGroupName": {
        "Ref": "EC2SecurityGroup"
      }
    }
  }
},
"GroupDescription": "Acceso a la base de datos"
},
"Type": "AWS::RDS::DBSecurityGroup"
},
"SampleDB": {
  "Properties": {
    "Engine": "MySQL",
    "DBName": {
      "Ref": "DatabaseName"
    }
  },
  "Port": { "Ref": "DatabasePort" },

```

```

    "MultiAZ" : { "Fn::FindInMap" : [ "AWSRegionCapabilities", { "Ref" : "AWS::Region"
}, "RDSMultiAZ"] },
    "MasterUsername": {
      "Ref": "DatabaseUser"
    },
    "DBInstanceClass": "db.m1.small",
    "DBSecurityGroups": [
      {
        "Ref": "DBSecurityGroup"
      }
    ],
    "AllocatedStorage": "5",
    "MasterUserPassword": {
      "Ref": "DatabasePassword"
    }
  },
  "Type": "AWS::RDS::DBInstance"
},
"WebServerGroup": {
  "Properties": {
    "LoadBalancerNames": [
      {
        "Ref": "ElasticLoadBalancer"
      }
    ],
    "MinSize": "1",
    "LaunchConfigurationName": {
      "Ref": "LaunchConfig"
    },
    "AvailabilityZones": [
      {
        "Fn::Join": [
          "",
          [
            {
              "Ref": "AWS::Region"
            },
            "a"
          ]
        ]
      }
    ]
  },
  "MaxSize": "3"
},
"Type": "AWS::AutoScaling::AutoScalingGroup"
},
"AlarmTopic": {

```



```

"Properties": {
  "Subscription": [
    {
      "Endpoint": {
        "Ref": "OperatorEmail"
      },
      "Protocol": "email"
    }
  ]
},
>Type": "AWS::SNS::Topic"
},
"LaunchConfig": {
  "Properties": {
    "SecurityGroups": [
      {
        "Ref": "EC2SecurityGroup"
      }
    ],
    "ImageId": {
      "Fn::FindInMap": [
        "AWSRegionArch2AMI",
        {
          "Ref": "AWS::Region"
        }
      ],
      {
        "Fn::FindInMap": [
          "AWSInstanceType2Arch",
          {
            "Ref": "InstanceType"
          }
        ],
        "Arch"
      }
    ]
  }
},
"UserData": {
  "Fn::Base64": {
    "Fn::Join": [
      "",
      [
        "Database=",
        {
          "Fn::GetAtt": [
            "SampleDB",
            "Endpoint.Address"
          ]
        }
      ]
    ]
  }
}

```

```

    },
    ".":
    {
        "Ref": "DatabasePort"
    },
    "&DBUser=",
    {
        "Ref": "DatabaseUser"
    },
    "&DBPassword=",
    {
        "Ref": "DatabasePassword"
    },
    "&WSPort=",
    {
        "Ref": "WebServerPort"
    }
    ]
}
},
"InstanceType": {
    "Ref": "InstanceType"
}
},
"Type": "AWS::AutoScaling::LaunchConfiguration"
}
},

```

```

"Mappings": {
    "AWSInstanceType2Arch": {
        "t1.micro" : { "Arch": "64" },
        "m1.small" : { "Arch": "32" },
        "m1.large" : { "Arch": "64" },
        "m1.xlarge" : { "Arch": "64" },
        "m2.xlarge" : { "Arch": "64" },
        "m2.2xlarge" : { "Arch": "64" },
        "m2.4xlarge" : { "Arch": "64" },
        "c1.medium" : { "Arch": "32" },
        "c1.xlarge" : { "Arch": "64" },
        "cc1.4xlarge" : { "Arch": "64" }
    },
    "AWSRegionArch2AMI": {
        "us-east-1" : { "32": "ami-c813e0a1", "64": "ami-c213e0ab" },
        "us-west-1" : { "32": "ami-09c7974c", "64": "ami-0bc7974e" },
        "eu-west-1" : { "32": "ami-6bc2f61f", "64": "ami-6fc2f61b" },
        "ap-southeast-1" : { "32": "ami-70f28c22", "64": "ami-72f28c20" },
        "ap-northeast-1" : { "32": "ami-ac03a8ad", "64": "ami-b003a8b1" }
    }
}

```

```

    },
    "AWSRegionCapabilities" : {
      "us-east-1" : { "RDSMultiAZ" : "true" },
      "us-west-1" : { "RDSMultiAZ" : "true" },
      "eu-west-1" : { "RDSMultiAZ" : "true" },
      "ap-southeast-1" : { "RDSMultiAZ" : "true" },
      "ap-northeast-1" : { "RDSMultiAZ" : "true" }
    }
  },
  "Outputs": {
    "URL": {
      "Value": {
        "Fn::Join": [
          "",
          [
            "http://",
            {
              "Fn::GetAtt": [
                "ElasticLoadBalancer",
                "DNSName"
              ]
            },
            "/"
          ]
        ]
      },
      "Description" : "URL del website"
    }
  }
}

```