INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE (TECNOLOGÍA - GESTIÓN)

MODERNIZACIÓN Y AUTOMATIZACIÓN DE PROCESOS DE PROVISIONAMIENTO DE INFRAESTRUCTURA

AUTOR: Marzullo, Federico Agustín (Leg. Nº 101825)

DIRECTOR DE TESIS: Almada, Jorge

TESIS PRESENTADA PARA LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN DIRECCIÓN ESTRATÉGICA Y TECNOLÓGICA

BUENOS AIRES

SEGUNDO CUATRIMESTRE, 2022

Agradecimientos

Ante todo a Dios, por sobre todas las cosas, por acompañarme siempre y en todo lugar, por todo cuanto dispone y hace posible en mi vida.

A Verizon Business por costear la cursada de la maestría en un momento económico-financieramente muy complejo de mi vida en el cual por mi cuenta, sin dicha ayuda, me habría sido imposible tomar el programa de maestría.

A la dirección de escuela de postgrado de ITBA por la oportunidad de estudiar ahí como analista de sistemas y por haberme permitido continuar con mi trabajo de tesis varios años después de finalizada mi cursada.

A los docentes maravillosos con los cuales tuve el placer de compartir el programa y a mis compañeros de cohorte de quienes también aprendí muchísimas lecciones y experiencias importantes.

Especialmente agradecer al Mg. Jorge Almada por tutelarme durante este trabajo de tesis.

A mis colegas de IT a lo largo de mi carrera que por fortuna conocí y colaboraron con mi desarrollo profesional siempre, pero fundamental y especialmente a Andrés Nicolás, Ariel Masciotta y Alejandro Loiacono, expertos en el área cubierta por este trabajo de tesis y que han dedicado tiempo personal para brindarme su conocimiento y experiencia.

A JP Morgan Chase por proveerme de las plataformas de acceso a libros técnicos que facilitaron efectuar parte del desarrollo teórico de este trabajo.

Por último, celebro haber podido cumplir con este objetivo personal y profesional; y al fuerte impulso motivacional y de constancia que requerí a lo largo del programa para sostener el compromiso de lectura, análisis, construcción y desarrollo, esperando el presente trabajo se encuentre a la altura del nivel académico que recibí de esta prestigiosa casa de estudio.

Dedicatoria

En este apartado introductorio del trabajo de tesis, quisiera permitirme ahondar en algunos aspectos que considero pequeñas grandes lecciones, luego de haber procrastinado por más de una década la conclusión del trabajo que me permitiría recibir mi título de magíster y quisiera compartir con el lector.

Un doctor llamado Ali Binazir intentó una vez responder a la pregunta "por qué existimos" o "por qué nacimos" cada uno de nosotros, y así escribió un artículo que terminó publicado en un blog de la Universidad de Harvard e intentaré resumir para darle forma al punto en cuestión.

Para que un hombre o mujer entre millones pudiera conocer a un hombre o mujer entre millones dentro de su comunidad, supongamos unos 200 millones; y luego dieran lugar a la concepción (tarea que también conlleva un estudio aparte), que al mismo tiempo la mujer nace con una reserva ovárica media de 100.000 óvulos fértiles y el hombre produce aproximadamente unos 400.000 trillones de espermatozoides totalmente diferentes, nos lleva a una probabilidad de ser quienes somos al momento de la concepción de **1 entre 400.000 trillones**.

Si agregamos el tiempo medio de vida del hombre en la tierra, la cantidad de generaciones desde su aparición, la expectativa media de vida y que cada 20 años aparece una nueva generación, estamos sumando otra variable más de 1 entre 10 elevado a la **45.000**.

Al agregar todo esto, el resultado final de la probabilidad de que hoy hayamos existido es de $\mathbf{1x10^{2.685.000}}$. Sí, es correcto, esto es una probabilidad representada con un 1 y 2.685.000 ceros detrás.

Ahora bien, en Argentina la expectativa media de vida publicada en 2021 es de 80 años para mujeres y 73 para hombres, pero a efectos prácticos imaginemos un punto medio de 77 años.

Luego, las probabilidades de ganarse la lotería en Argentina, más específicamente el llamado Quini 6, son de 6.744.109.680 a una. Ganársela 1 vez al mes a lo largo de toda la vida representa una probabilidad de una chance en $8.4x10^{9828}$. Es una cantidad de dinero incalculable (y que probablemente no exista en el planeta), pero lo que resulta fascinante es que aun pareciendo

imposible que esto ocurra, ganarse la lotería una vez al mes a lo largo de toda nuestra vida desde nuestro nacimiento hasta nuestra muerte sigue siendo unas 273 veces más fácil que haber llegado al mundo.

Esto en lo personal me obliga a reflexionar acerca de diferentes cuestiones, ya que al margen de que esa probabilidad justifica nuestra llegada, no habla de lo que hacemos en vida ni de las decisiones que tomamos y nos llevaron hasta donde estamos hoy ni a lo que somos hoy.

En el complejo entramado que nos presenta la vida, sin importar qué tan abrumados nos sentimos por aquello que nos sucede, ya sea por providencia, elección, imposición social o consecuencia (entre otras), nos deja huella, nos forma, nos forja, nos enseña a aprender, desaprender y reaprender permanentemente y algunas otras, por simple presión evolutiva también nos hacen "cambiar de rumbo" y dar el siguiente paso hacia un nuevo paradigma de nosotros mismos.

A lo largo de la evolución humana, nuestro cerebro ha sido constantemente forzado a adaptarse para sobrevivir. Este órgano es una máquina de predicción, que busca repetir aquello que le da satisfacción y brinda felicidad y al mismo tiempo busca evitar aquello que le hace daño (ya sea mental, físico o emocional – busca protegernos). Su sistema límbico consta de lo que conocemos como la amígdala, el hipocampo, el septo, el hipotálamo y son los encargados de regular los estados emocionales tales como la felicidad, el miedo y la agresión; por lo tanto son los que entre otros se activan ante una situación de peligro y en modo de supervivencia trabaja con un sistema llamado FFFS (por su sigla en inglés *Freeze*, *Flight*, *Fight System* – Congelarse, Huir, Luchar). Así, me resulta inexorable meditar acerca de cómo los seres humanos nos posicionamos frente a la adversidad, frente a las probabilidades de cometer errores, a las amenazas, a los miedos.

Lo mencionado anteriormente me permite compartir por qué al comienzo de esta dedicatoria se invita al lector a reflexionar. En mi caso, acerca de la postergación y la resiliencia frente las situaciones que se nos plantean a lo largo de la vida, de cuántas veces nos paralizamos, de cuántas huimos y en cuántas luchamos o simplemente priorizamos diferente o postergamos o por qué no

también nos posicionamos diferente frente a situaciones ya conocidas pero a las que ya no reaccionamos igual.

Personalmente considero que la resiliencia es ese estado de lucha, de perseverancia, de romper con la somnolencia abrumante del bloqueo ante el reconocimiento de lo adverso y tomar la decisión de seguir adelante sabiendo que la energía necesaria para lograrlo debe mayor a la que tracciona para que nos mantengamos en el mismo estado de nulidad o huida. Es saber que algún día, en algún momento, se deberá romper con la quietud o la huida y cambiar nuestro momento inercial.

La resiliencia es esa sensación de entereza con la capacidad para ayudarnos a adaptar a lo que estamos enfrentando (ver *luz al final del túnel*) y que nos permite alcanzar nuestros objetivos.

Es la resultante de ese proceso de decisión, sin importar el resultado (ya que la resiliencia no garantiza el éxito), una invitación al aprendizaje que fortalece el espíritu y carácter del individuo, algo que las personas nos debemos a nosotros mismos y se deben desde la oportunidad que nos brinda la experiencia del ahora para seguir construyéndonos, con la motivación como motor, generalmente más intrínseca que extrínseca, pero sin dejar de pensar en la fuerza inconmensurable que vive en la voluntad.

Por esto es que la dedicatoria de este trabajo de tesis es a la resiliencia y a la vida, por todo aquello que me ha enseñado hasta acá, porque la aventura no se termina hasta que se termina y el aprendizaje es constante, porque la determinación y el propósito son aquello que nos empuja a levantarnos y seguir adelante y realizarnos, a perseguir sueños, con gratitud, entrega y humildad; recordando:

- Si depende de nosotros, el camino hacia lo que deseamos alcanzar se encuentra a una decisión de distancia,
- que con el tiempo incluso algo tan efímero como una gota de agua puede deformar y romper una roca,
- y que el único error real en la vida es la inacción.

Índice de contenidos

Índice de figuras

Índice de tablas

Introducción

Más y más se evidencia la búsqueda de las empresas en achicar costos y maximizar la eficiencia operativa en sus procesos y procedimientos. Con el avance de la ciencia y la tecnología nuevos paradigmas se hacen presentes en el universo de soluciones y con ello otros negocios emergen naturalmente como el provisionamiento de infraestructura virtual que permite comoditizar servicios de infraestructura apalancándose en la capacidad de servicios en la nube y por otro lado la automatización de procesos a través de la IA (Inteligencia Artificial) mediante RPA (*Robotic Process Automation*).

La economía digital juega a partir de lo expuesto arriba un rol preponderante, los bienes y servicios ya no necesitan estar dentro de la infraestructura de las empresas, sino que contratan a terceras y provisionan ambos fácilmente en cuestión de minutos como si estuvieran comprando una pizza desde una aplicación de celular. Las consecuencias de esto son muchas, apalancándose en la evolución tecnológica más capacidad de procesamiento y almacenamiento permiten bajar los costos incrementando la oferta y haciendo más viable incluso para nuevos jugadores en la industria poder comenzar a proveer estos servicios, bajando las barreras de ingreso al nicho de negocio.

No obstante, también es importante remarcar una alerta que está llamando a reflexión a muchas empresas y parte de la comunidad científica respecto del impacto ambiental que esto conlleva. Entre el negocio de las criptomonedas, la "internet de las cosas" (IoT) y la virtualización que requieren las empresas para mantener sus negocios funcionando el crecimiento de la industria de los Data Centers está creciendo en forma exponencial, y a la par, el impacto ambiental.

De acuerdo con un reporte reciente de "*Research and Markets*"[1], gracias a IoT y a la industria 4.0 el mercado de los Data Centers crecerá a interés compuesto a una tasa del 2% entre 2019 y 2025; pero aun así resulta mucho más costo-efectivo hacer subcontratación de gestión de datos que hacerlo

[1] Comprehensive Data Center Market Outlook and Forecast 2020-2025. (Markets, 2020)

dentro de las empresas mismas, al punto tal que incluso Cisco[2] está considerando apagar algunos de sus propios Data Centers para ahorrar energía y bajar costos de infraestructura.

Durante el curso de esta tesis el maestrando se propone inicialmente presentar las bases y fundamentos de las tecnologías subyacentes en torno a los componentes de todas las alternativas actualmente vigentes para resolver la problemática, junto con las metodologías y tecnologías implicadas en la automatización y modernización de un proceso de provisionamiento de infraestructura, proponiendo un marco de trabajo sobre la alternativa seleccionada que facilite su implementación haciendo uso de bibliografía académica de relevancia para el sector, partiendo de la utilización de mejores prácticas y finalizando con la aplicación directa sobre un sistema de automatización de provisionamiento de infraestructura.

Relevancia

A medida que crece la necesidad de automatizar, modernizar y llevar adelante más análisis de datos a escala (con Cloud, IA, Big Data y Data Analytics a la cabeza) más y más compañías están tercerizando sus servicios de Data Center para poder mejorar la eficiencia, productividad, seguridad y la costo-efectividad de sus operaciones; lo cual les permite al mismo tiempo ahorrar en espacio físico, energía, sistemas de refrigeración para servidores y conectividad.

Además de una reducción de costos en torno al *hardware* y servicios asociados a este, se busca automatizar y modernizar el provisionamiento de los elementos que soportan los servicios de infraestructura para mejorar la eficiencia operativa, reducir o eliminar el error humano y consecuentemente lograr un impacto positivo en el "*time-to-market*", siendo éstos el valor agregado que persigue el propósito de esta tesis.

[2] Data Center Market is blooming amid shift to cloud. (Loten, 2019)

Objetivo principal

El objetivo principal de esta tesis es el de formular los pasos y metodología de trabajo hacia una modernización y automatización del provisionamiento de infraestructura, identificando los problemas habituales y las nuevas oportunidades en la implementación de tecnología de vanguardia que permite reducir errores, bajar costos, incrementar la eficiencia operativa y agilizar el "*time-to-market*".

Por último, se propondrá una metodología de trabajo para implementar el proyecto de desarrollo la solución.

Objetivos específicos

Para este trabajo de modernización de proceso se plantearán 4 objetivos específicos:

- Relevar la problemática a resolver y las variables involucradas en el contexto del problema y que intervienen en el proceso.
- Analizar soluciones viables para la automatización del proceso en función de los recursos, tecnología, metodología y costos que resuelvan el objetivo anterior.
- Proponer un marco de trabajo para la solución elegida, que incluya el *stack* de tecnología, la arquitectura de ambientes, la gestión del proyecto, el *roadmap* y el plan de implementación.
- Evaluar resultados obtenidos basados en el *testing* de la solución y las experiencias de usuario post-implementación.

Antecedentes para la Hipótesis

En 2002 Amazon crea una subsidiaria llamada "Amazon Web Services", con la meta de permitir a los desarrolladores y emprendedores construir aplicaciones innovadoras por sí mismos. La nube bajo el concepto de "Cloud Computing" se popularizó con Amazon en el año 2006 cuando hace el lanzamiento de "Simple Storage Service" o S3, seguido por de "Elastic Compute Cloud", aunque las

primeras referencias al concepto de nube computacional aparecen una década antes en 1996. Estos productos fueron quienes dieron empuje como pioneros a la tecnología de virtualización de servidores para dar génesis al concepto de "*Infrastructure as a Service*" (IaaS) como la forma económica de provisionar infraestructura "on-demand" con precios regularizados por utilización.

Para el año 2008 fue Google quien propone el servicio "*Platform as a Service*" (PaaS) como extensión de IaaS pero bajo una plataforma que permitiera facilitar a los clientes de una interfaz capaz de provisionar, instanciar, ejecutar y gestionar paquetes modulares que compongan una plataforma de Computing sin toda la complejidad inherente al mantenimiento de la infraestructura típicamente asociado con el desarrollo e implementación de las aplicaciones.

Es en 2010 cuando Microsoft lanza su primera versión de Microsoft Azure, el cual integra todo lo anterior junto con una suite de desarrollo de software basada en Visual Studio y un año después IBM hace lo suyo bajo el nombre de Smart Cloud y Oracle un año después lanza Oracle Cloud, quien integra no sólo todo lo existente hasta el momento en materia de servicios montados sobre la nube sino que además agrega "*Software as a Service*" (SaaS) para disponibilizar soluciones de aplicaciones como una capa adicional de servicios.

Estas tecnologías permitieron a las empresas incluso construir nubes propietarias dentro de sus propias infraestructuras, pero por lo expuesto anteriormente en la introducción de esta tesis, más y más empresas se replantean los costos y complejidades inherentes a dichas prácticas y consideran migrar sus nubes a Data Centers externos.

La escalabilidad que permiten estas soluciones posibilita el crecimiento horizontal y vertical en las implementaciones, dando la capacidad de controlar en forma efectiva los costos de provisionamiento en función de la utilización real de recursos, lo cual abre la puerta a un universo de desarrollo de soluciones de IT a medida que se adapten a los requerimientos puntuales de cada cliente.

La hipótesis de esta tesis partirá de la necesidad de plantear un marco de trabajo que permita facilitar la implementación de proyectos que busquen modernización y automatización de procesos

de provisionamiento de infraestructura como medio de mejora en la eficiencia operativa y *time-to-market* y por otro lado apuntará a la reducción de costos y error humano, permitiendo hacer mejor y más preciso seguimiento logístico en cada paso del proceso de provisionamiento a escala.

Hipótesis

La hipótesis plantea que el provisionamiento tradicional de infraestructura como se conoce hoy en la forma de abastecimiento, construcción y ampliación de potencia computacional de Data Centers es obsoleta, inefectiva e ineficiente; no sólo desde los costos, sino también desde la praxis. El maestrando se propone demostrar a lo largo de este trabajo de tesis que la migración a la nube es viable y que una solución basada en estas tecnologías es capaz de responder a la versatilidad, costos y tiempos que necesitan tanto el negocio como las empresa prestadoras de servicios basados en IT.

Metodología

Para demostrar mi hipótesis me propongo:

i) Hacer un análisis de situación actual como punto de partida y descripción de la problemática, proponiendo distintas tecnologías como parte de solución a cada uno de los aspectos individualmente con sus respectivos funcionamientos y principios.

ii) Seleccionar un enfoque y conjunto de tecnologías para resolver el problema, plantear un marco metodológico como objetivo sobre el cual se va a formular la solución a proponer, basándome en los fundamentos presentados dentro del marco teórico de esta tesis.

iii) Desarrollar un plan de implementación.

iv) Describir los resultados esperados y cerrar este trabajo con las conclusiones obtenidas.

Revisión bibliográfica

Para realizar este trabajo me basaré en bibliografía especializada y en publicaciones de expertos, teniendo en especial consideración que fueran de publicaciones de fecha reciente relacionadas con el estado de las tecnologías y metodologías en cada uno de los aspectos cubiertos a lo largo de este trabajo.

Especialmente para el marco teórico de este trabajo realizaré una revisión bibliográfica y me basaré en los siguientes temas como ejes fundamentales:

- Infrastructure as Code
- Ansible
- Docker
- Kubernetes
- Migración de aplicaciones a la nube

También se revisarán artículos, *papers*, publicaciones y videos de referentes de la industria para temas específicos comprendidos en el marco de la tesis y por último, también se mantuvieron entrevistas con expertos para validar el marco de estudio y solución propuesta.

Limitaciones y restricciones

El presente trabajo se basa en bibliografía especializada y en experiencia profesional de expertos en los temas abarcados a lo largo de este trabajo, divulgadas en libros, *papers* y en disertaciones online, como así también entrevistas con expertos en los temas tratados en este trabajo de tesis. La problemática, soluciones viables, marco de trabajo e implementación de la solución seleccionada se ven delimitadas y adaptadas a la medida de los marcos legales y tecnológicos de la empresa donde se desarrollará este trabajo y el alcance y propuesta consecuentemente se ven afectados por ellos. Este trabajo de tesis de modernización no pretende proponer la solución a implementar como la mejor existente, sino resolver una problemática de la manera más eficiente y eficaz posible, teniendo en cuenta todos los elementos que incluso exceden la voluntad del maestrando desde lo legal, metodológico y tecnológico tal cual expresado anteriormente.

Articulación

En el Capítulo 1, se da el marco teórico de la investigación. Se definen conceptos de automatización de procesos, RPA (*Robotic Process Automation*) y se abordan Data Centers, *Containers* y *Cloud Computing* como ejes fundamentales de este trabajo; para sentar luego las justificaciones de la decisión en las tecnologías seleccionadas sobre las cuales finalmente se construirá el entorno de trabajo e implementación.

En el Capítulo 2, se analiza la problemática central de este trabajo de tesis, haciendo un relevamiento de la dinámica actual y revisando los elementos y variables intervinientes y sus complejidades inherentes.

En el Capítulo 3, se revisan los hallazgos del capítulo anterior y se trabaja en plantear objetivos y una propuesta de modernización del proceso en función de los recursos, tecnología, metodología y costos que resuelvan el problema.

En el Capítulo 4 se presenta el plan de implementación de la solución, incluyendo el *stack* de tecnología, la arquitectura de ambientes, la gestión del proyecto, el *roadmap* y el plan de implementación en sí mismo. Finalmente se describirá el plan de soporte post-implementación que se encargará de mantener la solución luego de finalizado el proyecto.

En el Capítulo 5, se analiza el impacto económico/financiero de la solución en dos dimensiones: análisis de demanda y análisis de costos.

En el Capítulo 6, se describen los resultados esperados y se agregan revisiones de las experiencias de usuario para evaluar la efectividad de la solución elegida.

En el Capítulo 7 se elaboran las conclusiones personales.

1. Marco teórico

"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."

(Bill Gates)

1.1. ¿Por qué automatizar un proceso?

Son muchas las razones por las cuales es provechoso automatizar un proceso. La automatización es esencial para gestionar, cambiar y adecuar no sólo infraestructura y procesos de IT, sino también todos los procesos que atraviesan a la compañía en cuestión. Simplificando los cambios en dichos procesos a través de la automatización se pueden ahorrar grandes cantidades de tiempo, dinero y energía (entre otros) que pueden dedicarse a la innovación tecnológica que permitirá otras sucesivas y necesarias transformaciones.

En su libro "*Learning Robotic Process Automation*" (Tripathi, 2018), el autor describe algunos de los beneficios obtenidos por la implementación de RPA, observables en distintas dimensiones, que nos permiten entender la necesidad de la automatización como inversión y eje de transformación tecnológica de una empresa:

- **Reducir costos operativos:** Los robots pueden realizar el trabajo de varias personas, dependiendo de la tarea. Asimismo también pueden ejecutar múltiples tareas en hebras (*threads*), permitiendo escalar rápidamente el modelo de producción, orquestando los flujos de trabajo entre tarea y tarea con múltiples procesos ejecutándose en paralelo. Cuando hablamos de líneas de producción este aspecto se ve mucho más claro aún, tanto en la reducción de costo de mano de obra y consecuente eliminación de la probabilidad de error humano como en incrementos en la precisión de trabajo y que por último se traducen en minimizar el desperdicio de materiales.

- **Mejorar la seguridad del empleado:** En líneas de producción, células automatizadas permiten remover a los empleados de ejecutar tareas peligrosas.

- **Reducción de tiempos de espera:** La automatización permite mantener procesos *in-house*, mejorando los procesos de control y reducir significativamente los tiempos de espera comparados con el *outsourcing*.

- **ROI más rápido:** Las soluciones de automatización están basadas en necesidades puntuales y a medida, que, aprovechando la reducción de costos de operación y tiempos de espera permiten mejorar y agilizar los retornos.

- **Habilidad de ser más competitivo:** Las células de automatización en líneas de producción permiten reducir los ciclos de tiempo y los costos por pieza y al mismo tiempo mejorar la calidad. Esto permite competir mejor a escala global. Adicionalmente, la flexibilidad de este tipo de soluciones permite escalar fácilmente la implementación de tecnología más moderna y transformarla en ventaja competitiva frente a los demás jugadores en el mercado.

- **Aumento de la producción:** Un robot tiene la habilidad de trabajar a velocidad constante, sin intervención humana, 24 horas al día 7 días a la semana. Esto significa que tenemos potencial para producir más. Nuevos productos pueden ser introducidos más rápidamente en el proceso de producción y la programación de nuevos productos puede realizarse desconectada de la línea de producción sin necesidad de interrumpir procesos existentes.

- **Calidad y producción de partes mejorada y consistente:** Células de automatización realizan típicamente los procesos de manufacturación con mucha menor variabilidad que un empleado humano. Esto resulta en un mejor nivel de control y consistencia en la calidad del producto.

- **Huella ambiental más pequeña:** A través de la optimización de equipamiento y procesos, reduciendo la chatarra y sobrantes y utilizando menos espacio, la automatización utiliza menor cantidad de energía; todo en suma permite bajar al mismo tiempo los costos y transformarse en ahorro para la compañía.

- **Mejor planificación:** La producción consistente llevada adelante por robots permite a una fábrica, taller o línea de producción predecir confiablemente los tiempos y costos. Esa predictibilidad permite ajustar los márgenes.

- **Reducir la necesidad de hacer *outsourcing*:** Las células de automatización tienen grandes cantidades de capacidad potencial concentrada en un solo sistema compacto. Esto permite producir partes *in-house* que previamente solían ser producidas fuera.

- **Optimización de utilización del espacio:** Los robots están diseñados para desempeñar su rol sobre bases compactas con la finalidad de poder encajar en espacios confinados. Adicionalmente a estar montados sobre el piso, también pueden estarlo sobre paredes, techos, rieles y estantes. Pueden realizar tareas en espacios pequeños, permitiendo ahorrar mucho dinero en espacio físico de planta.

- **Fácil integración:** La productividad permitirá integrar un Sistema completo hardware-software con controles incluidos. La célula de automatización llega a la línea de producción ya probada, con lo cual una vez instalada ya se encuentra lista para comenzar a producir.

- **Maximizar el trabajo:** Durante las próximas 3 décadas, las estadísticas muestran que más de 76 millones de *baby boomers* se retirarán y que sólo 46 millones de nuevos trabajadores los reemplazarán. Durante este tiempo, la demanda de trabajo continuará, haciendo de la automatización una solución real y viable.

- **Incrementar la productividad y la eficiencia:**
    - Producción 24/7 y manufactura amigable con procesos JIT (*Just-in-Time*).
    - Registros de *uptime* de producción con eficiencia en valores por encima del 90%.
    - Capacidad para poder realizar operaciones secundarias como por ejemplo tratamientos de piezas o servicios de IT derivados de otros que se pueden agregar a la cadena de valor.
    - Comunicación en tiempo real con máquinas y células automatizadas.
    - Capacidad de responder ante improvistos con cambios rápidos para partes, piezas, herramientas y software.
    - Capacidad de poder desarrollar operaciones múltiples de manera flexible.

- **Incrementar la versatilidad del sistema:**
    - Flexibilidad del sistema, fácilmente adaptable a nuevas herramientas (incluyendo capacidad de transición/migración hacia otras más modernas).

- Los robots son flexibles y pueden fácilmente ser desplegados cuantas veces sea necesario en nuevas aplicaciones.
- Los robots tienen la habilidad de permutar fácilmente dentro de una amplia gama de productos sin tener que reconstruir completamente las líneas de producción.
- En las máquinas-herramienta en líneas de producción, tener la habilidad de efectuar cambio rápido de las mismas para poder adaptarse a distintas piezas o tareas sin la necesidad de cambiar de robot.
- Enfoque de *mixed-flow production* para agregar flexibilidad a las fluctuaciones de demanda.
- Los robots son capaces de aprender instantáneamente nuevos procesos.
- Reducir los tiempos de cambio.

Luego de lo expuesto en los puntos anteriores y poniendo foco en la problemática que esta tesis abarca y resuelve es importante abordar el concepto de RPA (*Robotic Process Automation*) como una potente herramienta que brindará una posible pero no completa solución evaluar el proceso a automatizar y desarrollar la solución.

Como se planteaba anteriormente, son muchas las razones por las cuales resulta provechoso implementar soluciones de software basadas en RPA. El concepto de RPA va más allá de lo netamente tecnológico y se extiende hasta el objetivo final de optimización de procesos de negocio.

Este tipo de tecnologías permiten además, por su propia naturaleza, monitorear constantemente los flujos y manejos de información; lo cual resulta en un *governance* del dominio del problema más abarcativo sobre los propios procesos y por lo tanto la posibilidad de inyectar cambios en forma de acciones de acuerdo a determinadas situaciones y flujos alternativos, pudiendo intervenir con tomas de decisiones en tiempo real que permiten mejorar el desempeño general del proceso y ajustarlo mientras se ejecutan las tareas.

Respecto de las bondades que otorga RPA, el autor concluye: "*With the inclusion of AI technologies, RPA now has the capability to read from images or scanned documents, and it can interpret unstructured data and formats as well. However, most of the implementation is happening with structure and digital data.*" (sic) (Tripathi, 2018)

1.2. Robotic Process Automation

La propuesta de valor de RPA es seductoramente simple. Un incansable ejército de robots (*bots*) que trabajarán día y noche para abordar la montaña laboriosa de trabajo en carga de datos que sostiene nuestro mundo digital. Y RPA debería ser fácil de implementar, sin tener que esperar meses para que costosas APIs (*Application Programming Interface*) estén programadas y listas para utilizarse en la integración con sistemas *legacy*, ya que los *bots* interactúan con los sistemas tal y como un ser humano lo haría. Al observar a un *bot* copiar y pegar algunos cientos de campos en una fracción de segundo es difícil no emocionarse un poco. (Taulli, 2020)

Las primeras ventajas experimentadas por las empresas que utilizan RPA si bien son relativas a procesos de automatización de procesos rutinarios los campos de adopción son muy variados: un departamento administrativo puede implicar una gestión más eficiente de ciclos activos y pasivos, mientras provee soporte a toda la cadena de suministro de la empresa, desde los procesos de compra, gestión de stock, seguimiento logístico de envíos, etc.

Son estas empresas, que gracias a RPA, pueden en la práctica reducir costos globales, gestionar más fácilmente la redundancia, pueden ser más veloces en llevar a término determinadas prácticas (que incluso carecerán de errores humanos por distracción) y, al optar por la adopción de la solución de RPA más adecuada a las exigencias y normativas de su negocio e industria, tienen la garantía de poder trabajar de acuerdo a las mismas sin desvíos.

Por otro lado, las soluciones de RPA que incorporan inteligencia artificial son entonces capaces de aprender en el desempeño de su trabajo (basándose en datos adquiridos durante la experiencia y

recolectando datos en forma paulatina; esto es ML (*Machine Learning*), para trabajar cada vez de forma más autónoma y continuar mejorando y optimizándose, sin necesidad de la intervención de personas (usuarios o programadores), lo cual resulta muy útil desde un punto de vista de mejora continua de los modelos y procesos de negocio.

1.2.1. Errores en la implementación de RPA

En este aspecto, Taulli expresa lo siguiente: "*Now, let's be honest about what RPA doesn't do. It doesn't transform your organization all by itself, and it's not a fix for Enterprise-wide broken processes and systems. For that, you'll need end-to-end intelligent automation*". (sic)[3] (Taulli, 2020)

En el primer capítulo de su libro (*RPA Foundations*) dedica un apartado específico a las desventajas y errores en la implementación de estas tecnologías: *The Downsides of RPA*.

A priori, listando brevemente identifica los siguientes aspectos/dimensiones importantes a tener en cuenta: c*ost-of-ownership* (incluyendo entrenamiento y mantenimiento), deuda técnica, escala organizacional, seguridad, expectativas, preparación, límites y ambientes virtualizados.

Por cuestiones de alcance de este trabajo de tesis, se abordarán algunos puntos clave en torno a los problemas durante la implementación de RPA, a fin de plantear las bases de una correcta adopción de la tecnología; pero sólo a efecto declarativo de lo que deseamos evitar y para que sirva de guía en cualquier implementación de RPA que se considere llevar adelante:

- La adopción de RPA en los procesos de las empresas es seguramente una ventaja, no obstante también es claro que las bases de la empresa son determinantes para una correcta implementación de RPA y que sus procesos cumplan plenamente su función.
- Probablemente los errores más comunes son cometidos por la empresa, dependiendo en el modo en el cual éstas adoptan a RPA y la incorporan a sus procesos.

[3] www.pega.com/rpa

- El error más común de todos es de hecho creer que RPA puede funcionar para cualquier empresa y para automatizar cualquier proceso.
- En la mayor parte de los casos no es así: no todos los procesos, sobretodo hablando técnicamente. Por lo tanto, en un primer análisis, es necesario evaluar con atención si la propia firma tiene realmente necesidad de implementar RPA antes de invertir en esta tecnología.
- No tener bien claros los procesos de automatización y por lo tanto no tener conciencia de lo que realmente se simplificará y que esto requiere la intervención de la inteligencia humana.
- No estar en condiciones de gestionar adecuadamente los recursos internos de la empresa, creando así conflictos inútiles – nacidos en muchas ocasiones del propio miedo de la gente de ser sustituida por un "*software robot*".
- No tener competencias técnicas adecuadas para la gestión de RPA y negarse a confiar el proceso a una empresa tercera especializada en RPA expertos; lo cual conlleva a inversiones inútiles de dinero, tiempo en acciones repetitivas que no resuelven problemas y que no son productivas.
- No estar en condiciones de seleccionar, en modo adecuado, la plataforma más adecuada sobre la cual instalar RPA.
- A los puntos arriba indicados también se debe agregar la poca atención de las empresas a los costos de desarrollo de automatizar procesos. La integración de estos *softwares* con aplicaciones de la empresa podrían dificultarse, algo no siempre considerado.

Finalmente, al adoptar RPA en la empresa como solución de automatización, se debe siempre tener en cuenta que la misma no será óptima para siempre y que deberá readaptarse a las nuevas

aplicaciones en el futuro. *"Automation is good, so long as you know exactly where to put the machine."*

(Bill Gates)

1.3. Data Centers

En esta sección el maestrando se propone a dar una breve introducción al concepto de Data center, como uno de los elementos fundamentales al tema central de la tesis.

Nuestras compañías dependen de equipamiento de infraestructura para funcionar eficientemente y desarrollar sus actividades, lo cual torna en una prioridad el contar con los dispositivos y recursos funcionando en forma segura y efectiva. Mientras muchas compañías instalan su *hardware* directamente en sus instalaciones, es importante destacar que cada día, más y más organizaciones están optando por co-locar sus recursos de infraestructura en Data Centers.

En un artículo publicado por (Palo Alto Networks), se define al Data Center como: "*…a facility that centralizes an organization's shared IT operations and equipment for the purposes of storing, processing, and disseminating data and applications. Because they house an organization's most critical and proprietary assets, data centers are vital to the continuity of daily operations*".

Se puede decir entonces que las instalaciones de Data Centers aseguran que bienes físicos que mantienen los negocios en movimiento operan eficientemente y que los equipamientos críticos para la misión de la empresa están protegidos.

En el nivel más básico descriptible, un Data Center es una instalación que alberga las operaciones de IT (*Information Technology*), sus datos y equipamiento. Así como más y más de las funciones de negocio modernas se vuelven dependientes de servicios de *Cloud Computing* y *Data Storage*, el número de Data Centers alrededor del mundo se ha incrementado exponencialmente como hemos visto en la introducción de este trabajo de tesis.

Según los autores del artículo, en el pasado, los Data Centers eran infraestructuras físicas altamente controladas, pero la nube pública ha cambiado ese modelo. Excepto donde

restricciones regulatorias requieran que el Data Center se encuentre sin ningún tipo de conexión a internet, la mayoría de las infraestructuras modernas de Data Centers han evolucionado de servidores físicos "*on-prem*" hacia infraestructura virtualizada que soporta aplicaciones y cargas de trabajo a través de ambientes multi-cloud, concepto que se visitará luego dentro de este trabajo.

Los Data centers, entonces, soportan organizaciones interesadas en incrementar sus niveles de seguridad, potencia y funcionalidad para el equipamiento más importante que poseen.

1.3.1. ¿Qué hace un Data Center?

Resulta sumamente relevante ahora comenzar a describir las funciones de un Data center típico. En vez de tener el equipamiento en las instalaciones de las compañías (*on-premise*), los Data centers toman la pesada carga de las empresas y albergan la infraestructura por ellas.

Palo Alto Networks describe el rol del Data Center como una parte integral de la compañía, diseñada para soportar las aplicaciones del negocio y proveer servicios como:

- Almacenamiento de datos, gestión, *backups* y recuperación.
- Aplicaciones de productividad, como el *email*.
- Transacciones de *e-commerce* de alto volumen.
- Alimentación de comunidades de juegos *online*.
- *Big Data, machine learning* e inteligencia artificial

Según especifican, existen reportados más de 7 millones de Data Centers en todo el mundo.

Por su lado, (Cisco) en su *Hybrid Cloud e-book*, lista los componentes fundamentales presentes en los data centers, que incluyen los siguientes componentes críticos:

- *Routers*
- *Switches*

- Sistemas de almacenamiento (*Data Storage*)
- *Firewalls*
- *Servers*
- Controladores de *application-delivery*

Éstos también pueden albergar datos y servicios críticos para las compañías tales como *Email*, sitios web y servidores de *software*; pero al mismo tiempo incluyen otros elementos que soportan a los arriba listados, tales como subsistemas de energía, fuentes ininterrumpidas de energía (UPS), ventilación, sistemas de enfriamiento, supresión de fuego, generadores de respaldo y conexiones a redes externas.

1.3.2. Beneficios de un Data Center

Según indica Palo Alto Networks, almacenar equipamiento directamente en las instalaciones de las compañías presenta muchos desafíos y hay una considerable cantidad de oportunidades de cometer errores, tales como la alimentación eléctrica, la seguridad de los datos, la escalabilidad; los cuales pueden afectar cada aspecto de la organización desde finanzas hasta el staff que trabaja en la primera línea de contacto con los clientes.

Afortunadamente, indican, justamente estos aspectos son en los cuales los Data centers operan con excelencia:

- **Potencia eléctrica:** Los Data centers se aseguran de que exista una cantidad suficiente y constante de provisionamiento eléctrico al edificio para minimizar el riesgo de cortes eléctricos o merma de velocidad de respuesta en los equipos.
- **Seguridad:** La mayoría de las instalaciones requiere una muy extensiva seguridad de acceso al edificio, usualmente controlada a través de tarjetas de acceso, códigos o huellas digitales para ingresar.

- **Controles ambientales:** *Partners* de co-locación en las instalaciones se aseguran de que la temperatura del edificio sea la correcta para prevenir que el equipamiento se sobrecaliente.
- **Espacio:** Las instalaciones pueden variar sus tamaños desde grandes habitaciones a edificios de varios pisos, y por lo tanto ser eficientes a la hora de poder disponer de más espacio donde instalar su *hardware*.
- **Cumplimiento de normas/estándares:** Muchos Data centers han obtenido certificaciones de normas como SOC, HIPAA, ISAE, SAE 70 y PCI con el fin de alcanzar las necesidades de seguridad.
- **Costos:** Almacenar el *hardware* en un Data center típicamente significa un ahorro de dinero para la compañía en el largo plazo, por varios motivos (además de los arriba expuestos), a saber: Renovación de *hardware* tanto operativo como de respaldo, reposición por fallas de los dispositivos, servicios de soporte técnico, costos de técnicos especializados, compra de *hardware*, planeamiento y logística.

1.3.3. Clasificación de Data Centers

Cada Data center cae dentro de un nivel (*tier* - capa) específico, y cada nivel representa diferentes niveles de servicio para cada uno de ellos. Los Data Centers están certificados y regulados por la TIA[4] (Telecommunications Industry Association).

Así, los tipos de niveles de servicios que los Data Centers pueden proveer según dicha asociación son:

- **Nivel 1 (*tier* 1):** Es el nivel más bajo de todos, y es típicamente pequeño en cuanto a su espacio físico. Estos son usualmente pequeñas compañías con poco equipamiento. Estas instalaciones poseen un solo circuito de suministro eléctrico y enfriamiento sin

[4] Telecommunications Industry Association: https://tiaonline.org/

componentes redundantes en caso de caídas. Este nivel provee un 99.671% de disponibilidad.

- **Nivel 2 (*tier* 2):** Este nivel provee un balance entre gestión de costos y performance. Consisten en un solo circuito de suministro eléctrico y enfriamiento como el anterior, pero con componentes redundantes. Este nivel provee 99.741% de disponibilidad.
- **Nivel 3 (*tier* 3):** Estos Data Centers son mejores para compañías grandes con más equipamiento que precisa ser almacenado. Poseen múltiples circuitos de suministro eléctrico y de enfriamiento con sólo uno activo a la vez, con componentes redundantes en caso de interrupciones de suministro y pueden sostener sus servicios en caso de cortes por hasta 72 horas. Este nivel provee 99.982% de disponibilidad.
- **Nivel 4 (*tier* 4):** Finalmente, este tipo de Data Center típicamente brinda servicios de IT a negocios multimillonarios, y provee el nivel más alto de servicio del mercado. Estos Data centers poseen múltiples circuitos activos de suministro eléctrico y enfriamiento, componentes redundantes y son capaces de entregar 96 horas continuas de servicio en caso de cortes de los mismos. Este nivel provee 99.995% de disponibilidad.

1.3.4. Resumen - Ventajas y desventajas de los Data Centers

En la segunda parte del libro "*Cloud Data Centers and Cost Modeling*", llamada "*Data Center Facilities and Cost*" los autores (Wu & Buyya, 2018) se extienden vastamente a lo largo de los beneficios y problemáticas que enfrentan los Data Centers, algunos de los cuales ya se han visitado en el curso de este apartado, y que en conjunto con otras fuentes previamente citadas se resumirán y adaptarán brevemente a continuación.

En cuanto a los beneficios y ventajas de optar por utilizar Datacenters *on-prem*, se identifican:

- Ofrecer servicios a los clientes expresados en valores asequibles, separados en varios planes, de distintos niveles de disponibilidad.

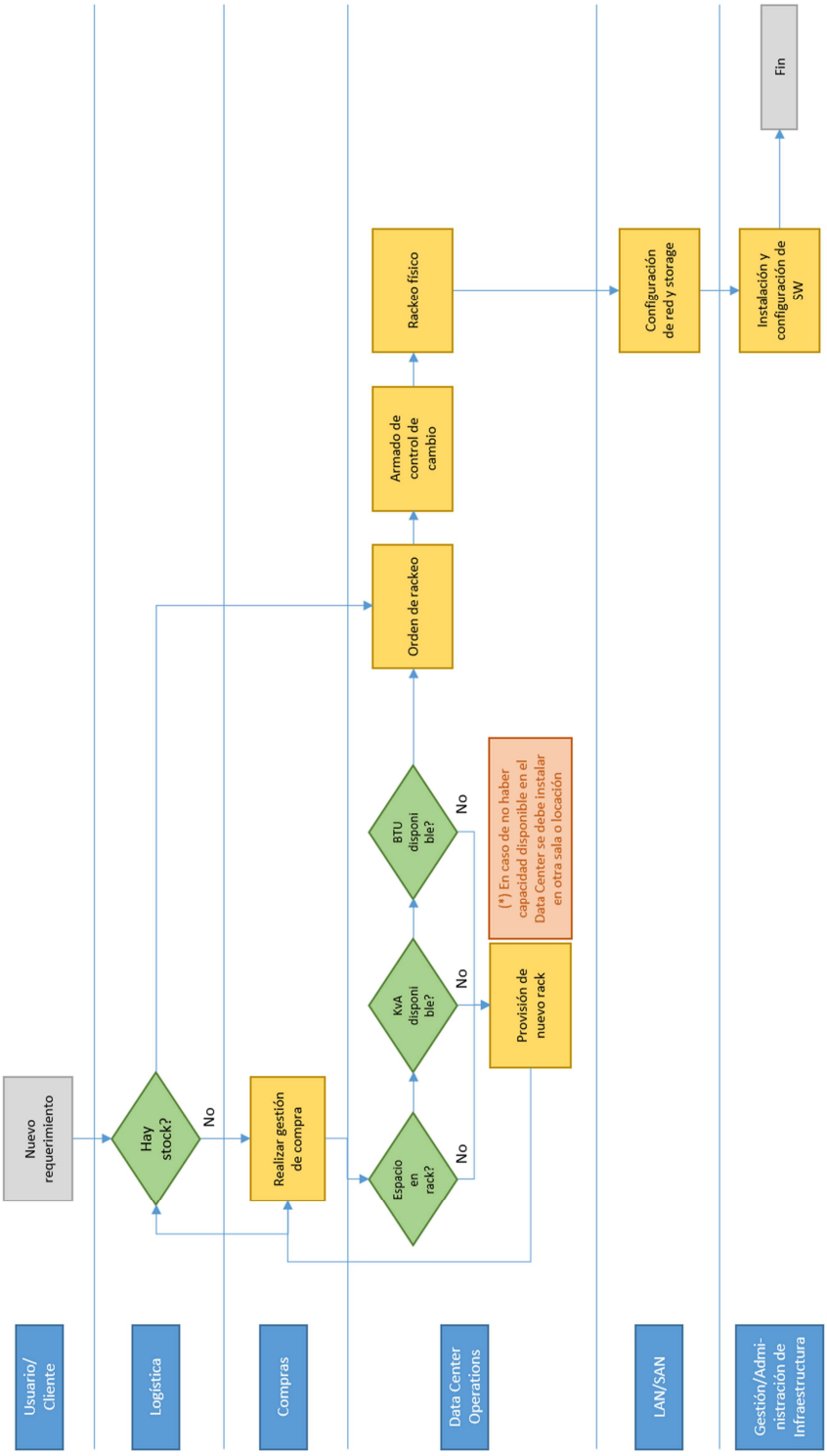
- Ofrecer un ecosistema de IT robusto tanto para *software* como para *hardware*.
- Ofrecer alto desempeño (*performance*) de servicios al distribuir la carga de la demanda a través de nodos de *clusters*.
- Los usuarios o cliente no necesitan preocuparse por contratar staff especializado para mantenimiento que se encargue de gestionar y dar soporte a sus Data centers.
- Los Data centers ofrecen escalabilidad instantánea en repuesta a necesidades de cambios de capacidad solicitados por clientes.
- Los servicios están disponibles siempre y sin caídas gracias a sistemas de respaldo y redundancia.
- Las empresas no necesitan pensar en problemas tales como espacio disponible para escalar sus necesidades de tecnología, o de capacidad de provisionamiento de suministro eléctrico o de enfriamiento, al delegar 100% esto a los proveedores de servicios de Data centers.

Por último entonces, los inconvenientes y desventajas interpretados de los autores respecto de los Data centers son:

- Al contratar servicios de proveedores de Data centers, las compañías no tendrán control completo de lo que sucede en su infraestructura. Esto se debe al hecho de que los recursos humanos involucrados en las tareas de operaciones y el *hardware* propiamente dicho se encuentran fuera de sus instalaciones, muchas veces en locaciones remotas.
- El uso y calidad de los servicios de Data centers pueden variar basados en la conectividad a Internet y a las instalaciones del cliente.
- Existen limitaciones en las características y prestaciones de seguridad ofrecidas por los proveedores de servicios de Data centers.
- Algunas compañías pueden agregar costos adicionales de soporte técnico a los clientes.

- En caso de resolución de problemas e incidentes, los clientes deben confiar en el personal de soporte de los proveedores de Data Centers. Por lo tanto, la resolución de los mismos depende de las habilidades y conocimiento del personal de soporte.

1.3.5. Diagrama de flujos típico de provisionamiento de infraestructura en un Data Center



1.4. Figura 1: Diagrama de flujos del provisionamiento de un Data Center. Imagen propiedad del maestrando.Cloud Computing

Se comenzará por definir qué es el Cloud Computing como punto de partida de este apartado. Cuando se habla de estas tecnologías, en su esencia más elemental, refiere a la entrega de servicios tales como *servers*, almacenamiento, bases de datos, infraestructura, *software*, análisis de datos, inteligencia artificial (y otros), utilizando la nube (internet) como plataforma. (Javatpoint, n.d.)

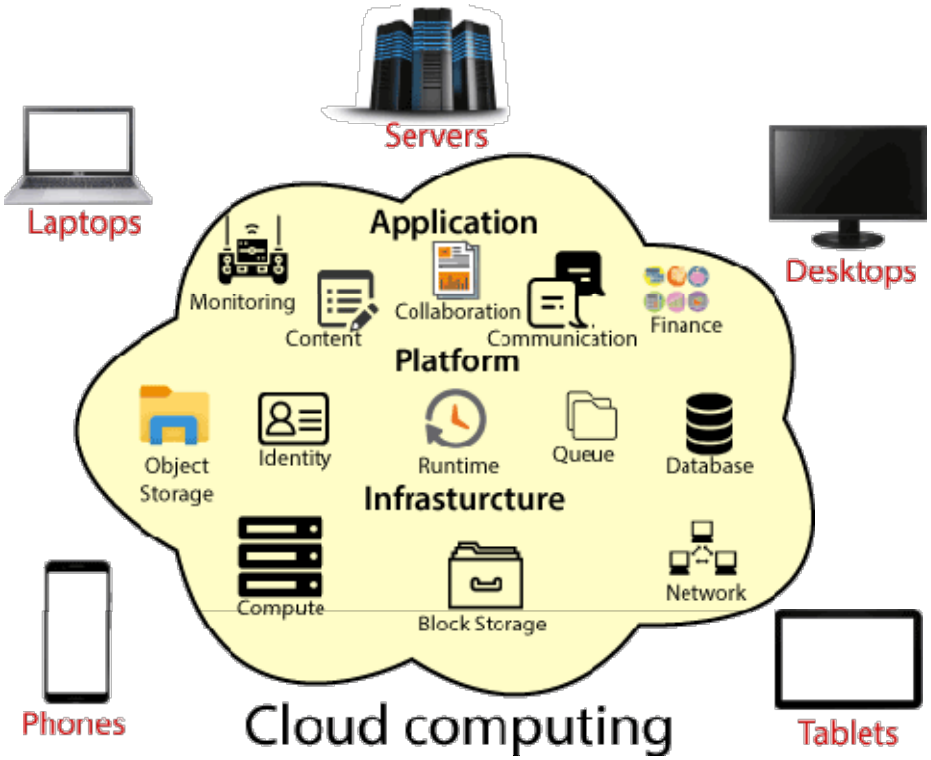


Figura 2: Servicios y dispositivos interoperables en Cloud Computing. Fuente: (Javatpoint)

Según relata (Prabhu) la historia de este término se origina con el nacimiento de la ARPANET (en el año 1977) y la CSNET (en 1981). El término de nube fue también utilizado en 1993 para aludir a plataformas de computación distribuida. Mientras tanto, fue en la década del 1990, donde las empresas de telecomunicaciones vieron una oportunidad en las redes privadas virtuales (VPN) para balancear carga en las redes y servidores de la época y así gestionar el ancho de banda (escaso en esa época) de forma más efectiva.

Sin embargo, el término fue popularizado por Amazon como se introdujo en al inicio de este capítulo, a través de su subsidiaria *Amazon Web Services* (AWS), en el año 2006; cuando lanzó su *Elastic Compute Cloud* product.

Reformulando el planteo de Javatpoint, las tecnologías de la nube proveen una alternativa a los Data Centers en las instalaciones de las compañías (*on-prem*), tal y como se desarrolló en el capítulo anterior; escenario en el cual la empresa debe gestionar todo, desde la compra e instalación de *hardware* (y su virtualización si corresponde), instalación del sistema operativo y otras aplicaciones requeridas por el negocio, configuración de los elementos de red, configuración de firewalls y finalmente instalar y configurar los repositorios de datos. Luego de todo esto, la empresa se hace responsable por mantener todos estos elementos funcionales y en óptimo estado a través de sus ciclos de vida completos.

Al optar por tecnologías de nube (luego se tratarán los tipos de nube existentes), un proveedor es responsable por la compra y mantenimiento del hardware, lo cual evidencia una gran ventaja. Incluso más, estos proveedores tienen la posibilidad de brindar una gran variedad de distintos *software* y plataformas dentro de su portfolio de servicios, lo cual también resulta en una ventaja ya que la empresa no necesita incurrir en gastos de licencias ni instalaciones ni mantenimientos; no obstante, estos servicios son generalmente rentados, pero asimismo facturados en función de su utilización, lo cual resulta en otra ventaja de escalabilidad.

1.4.1. Características principales de Cloud Computing

Como se ha expresado previamente en esta sección de la tesis, las tecnologías basadas en la nube proponen un cambio de paradigma y con ello una batería de posibilidades dentro de su portfolio de servicios, que apuntan a simplificar el acceso a los mismos con mayor control en torno a la escalabilidad, provisionamiento, gestión y facturación y otras más que se detallan a continuación, según las define (Javatpoint) en su apartado acerca de las características de *Cloud Computing*:

- **Resource Pooling**: Esto significa que el proveedor extrajo los recursos computacionales para brindar servicios a múltiples clientes a través de un modelo *multi-tenant* (que permite múltiples clientes en paralelo). Existen diferentes recursos tanto físicos como virtuales asignados y reasignados en forma dinámica dependiendo de la demanda de cada cliente dentro del modelo de servicios. El cliente generalmente no tiene ni control ni información respecto de la locación desde donde se están provisionando dichos servicios, pero a veces es capaz de especificar la locación donde desea que estos servicios sean virtualmente instalados, a un alto nivel de abstracción.
- **On-Demand Self-Service:** Esta resulta una de las más valiosas e importantes características de los servicios de nube, ya que el usuario puede continuamente monitorear varios factores de los servidores por ejemplo (tales como *uptime*, capacidades de procesamiento y cantidad de espacio de almacenamiento provisto). Con esta característica, el usuario puede también monitorear el rendimiento computacional del servidor.
- **Fácil mantenimiento:** Los *servers* son fácilmente mantenidos y el tiempo fuera de servicio es muy bajo, e incluso en algunos casos inexistente. En este punto, las mejoras son constantes y se actualizan por sí solas dentro del server, generando mejoras *on-the-fly* y sin intervención del usuario, e incluyen mejoras de rendimiento y corrección de errores.
- **Acceso a grandes redes:** El usuario puede acceder a los datos en la nube o subir la información a la nube desde cualquier punto, sólo necesitando un dispositivo electrónico (como una PC, tablet o teléfono móvil) y conexión a Internet.
- **Disponibilidad:** Las capacidades de la nube pueden ser modificadas en función del uso y pueden ser extendidas ampliamente. Estas tecnologías permiten analizar la utilización de capacidad de almacenamiento y permiten al usuario comprar extensiones en caso de ser necesarias por bajos montos de dinero.

- **Sistema automático:** La plataforma de nube permite analizar automáticamente la información necesaria y soporta la capacidad de implementación de métricas en algunos niveles de servicios. Se pueden monitorear, controlar y reportar los usos de cada uno de ellos, lo cual se verá traducido en transparencia tanto para el proveedor como para el usuario.
- **Económico:** Representa una única inversión ya que el proveedor es quien se encarga de comprar el almacenamiento y éste luego repartido virtualmente y asignado a todos los clientes de acuerdo a sus necesidades. Los clientes asimismo sólo pagan por lo que consumen.
- **Seguridad:** La seguridad en la nube es una de las mejores características de Cloud Computing. Permite crear una foto (*snapshot*) de toda la información almacenada para que la misma no se pierda incluso si uno de los *servers* se daña. La información es almacenada dentro de dispositivos de almacenamiento, los cuales no pueden ser hackeados ni utilizados por otras personas. El servicio de almacenamiento es rápido y confiable.
- **Pay-as-you-go:** En Cloud Computing, el usuario debe pagar solo por el servicio o espacio que ha utilizado. No existen cargos escondidos ni extra que deba pagar. El servicio es económico y la mayoría del tiempo los proveedores brindan un poco de espacio de almacenamiento adicional gratuito.
- **Servicio medido:** Como se mencionó anteriormente, el monitoreo de recursos utilizados es analizado por herramientas de soporte gratuitas dentro de la plataforma.

En resumen, reformulando el planteo del texto, es la compañía proveedora de servicios de nube la que mantiene los *servers*, se responsabiliza de las caídas de los servicios de tecnología y los resuelve. La compañía también compra el software y las licencias para la operación de su negocio.

Todo eso mantenido por un costo mensual que pasan en forma de tarifas a las empresas a las que sirven.

Finalmente, el maestrando agrega que existen una gran cantidad de características de seguridad, lo cual es un punto sumamente positivo y que junto con el tiempo de acceso sumamente bajo a la plataforma, esto permite subir y bajar información de la nube con gran rapidez.

1.4.2. Tipos de nube

Entre las fuentes investigadas por el maestrando, se identificaron esencialmente 4 tipos distintos de nube según describen (Javatpoint) y (Red Hat):

- **Pública:** Los recursos de nube pertenecen y son operados por un proveedor (third-party) de servicios de nube y se denominan nubes públicas (*Public Cloud*). Este proveedor entrega recursos computacionales tales como *servers*, *software* y almacenamiento sobre internet.
- **Privada:** Los recursos de la nube que son exclusivamente utilizados dentro de un único negocio u organización se denominan nube privada (*Private Cloud*). Una nube de este tipo puede estar físicamente alocada en el Data Center dentro de las instalaciones de la compañía o bien dentro de las instalaciones de un proveedor de servicios de terceros.
- **Híbrida:** Es la combinación de nubes privadas y públicas, lo cual es conectado a través de tecnología que permite a aplicaciones de datos ser compartidas entre ellas. Este tipo de nube provee flexibilidad y más opciones de implementación al negocio.
- ***Multicloud*:** Se refiere al enfoque de nube compuesto por más de un servicio en la nube, de más de un proveedor, ya sea público o privado. Todas las nubes híbridas son nubes *multicloud*, pero no todas las *multicloud* son nubes híbridas. Las nubes de tipo *multicloud* se vuelven híbridas cuando múltiples nubes son conectadas por alguna forma de integración u orquestación.

1.4.3. Tipos de servicios de nube

Según describe RedHat (Red Hat), existen básicamente 3 tipos de modelos de servicio distintos de nube, que luego en este apartado serán cubiertos uno a uno:

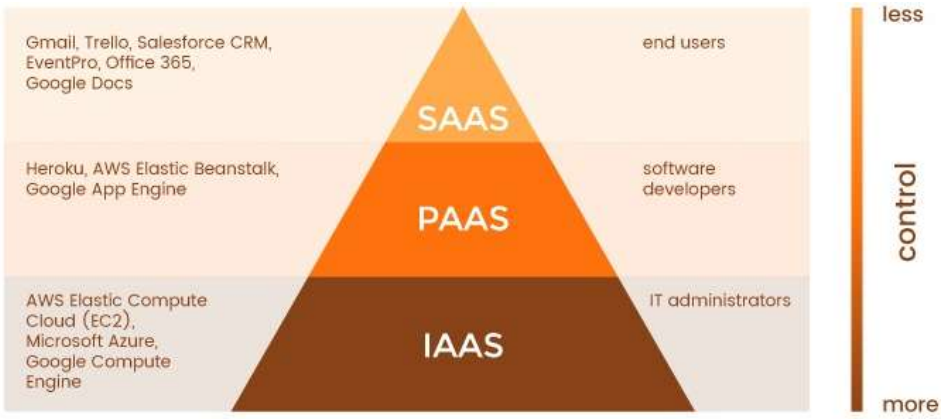


Figura 3: Tipos de servicio provistos por la nube. (Jimenez)

Los servicios en la nube son infraestructura, plataformas o software que es hosteado por proveedores terceros (*3rd party*) y que son disponibilizados a los usuarios a través de Internet. Cada uno de los servicios ilustrados en el gráfico superior facilitan el flujo de información de usuarios desde los clientes que consumen las aplicaciones (*front-end*) hasta los sistemas de los proveedores de servicios en la nube que se ejecutan por detrás (*back-end*).

Así, según comenta el artículo, a medida que subimos de capa en el modelo obtenemos menos gobierno (*governance*) de los servicios, configuraciones y características pero son los proveedores quienes facilitan el acceso a servicios que pueden hacerse disponibles con muy poco esfuerzo, como en el caso de suites de aplicaciones como manejo de correo, *backups* de información o gestión de almacenamiento:

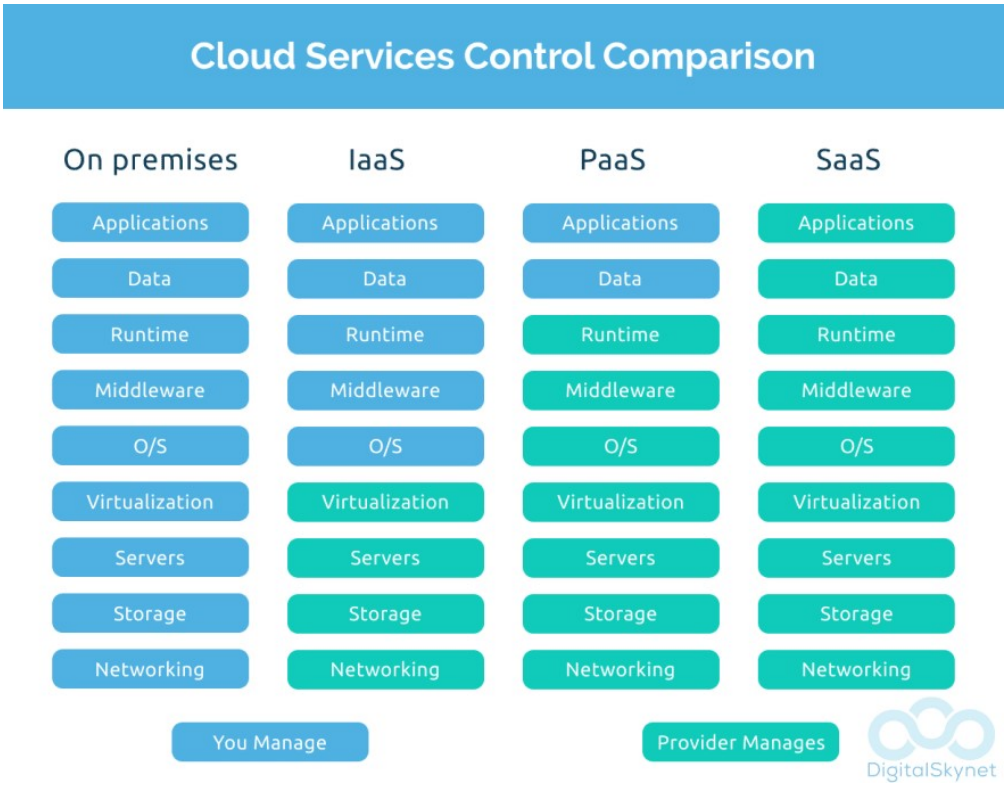
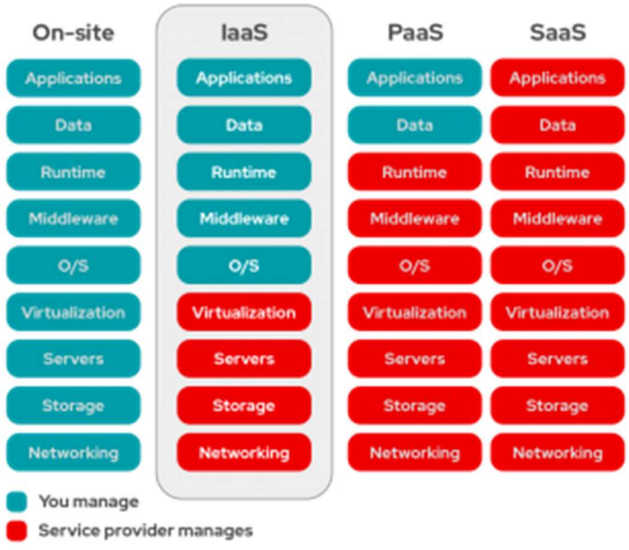


Figura 4: Comparación de servicios de nube y sus niveles de control. (Kobilinskiy)

A continuación se describirán los tipos de servicios en la nube, según los describe Red Hat.

1.4.3.1. Infrastructure as a Service (IaaS)

Este tipo de servicio implica que un proveedor de servicios de nube maneje la infraestructura por nosotros (servidores, redes, virtualización y almacenamiento reales) a través de una conexión a internet. Así, el usuario tiene acceso a través de una API (*Application Programming Interface)* o

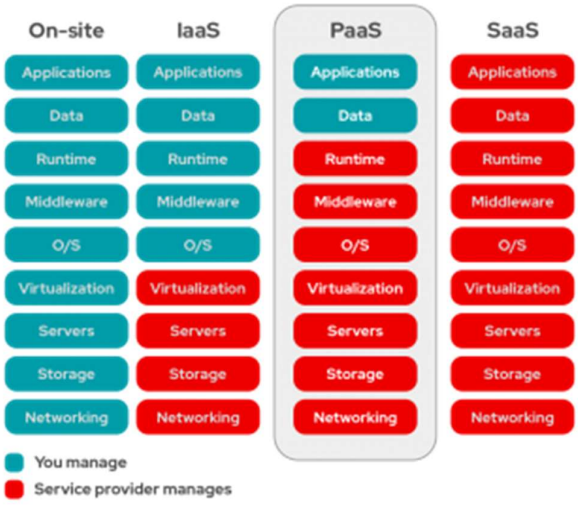


tablero de control y básicamente renta la infraestructura. El usuario gestiona cosas como el sistema operativo, aplicaciones y *middleware* mientras el proveedor se ocupa del *hardware*, redes, discos rígidos, almacenamiento de información y servidores; y tiene la responsabilidad de hacerse cargo de las caídas de servicio, las reparaciones y los problemas en el *hardware*. Este es el modelo típico de implementación de los proveedores de almacenamiento en nube.

Figura 5: Diagrama IaaS. (Red Hat)

1.4.3.2. Platform as a Service (PaaS)

En este modelo de servicio, el hardware y una plataforma de aplicación-software son provistas y gestionadas por un proveedor de servicios de nube externo, pero el usuario maneja las aplicaciones que corren por encima de la plataforma y los datos que precisan las aplicaciones. Ante todo para desarrolladores y programadores,



PaaS les da a los usuarios una plataforma compartida en la nube para gestión y desarrollo de aplicaciones (una componente importante de DevOps) sin tener que construir y mantener la infraestructura usualmente asociada con el proceso.

Figura 6: Diagrama PaaS. (Red Hat)

1.4.3.3. Software as a Service (SaaS)

SaaS es un servicio que entrega una aplicación de software – la cual es gestionada por el proveedor de servicios en la nube – a sus usuarios. Típicamente, las aplicaciones SaaS son aplicaciones *web* o *mobile* que los usuarios pueden acceder a través de un navegador web. Actualizaciones de software, corrección de errores y otras tareas de mantenimiento de software son resueltas al usuario, y éstos se conectan a las aplicaciones en la nube a través de un tablero o API. SaaS también elimina la necesidad de tener una aplicación instalada localmente en las computadoras de cada individuo, permitiendo una mejor gestión de grupos o equipos de trabajo que acceden al software.

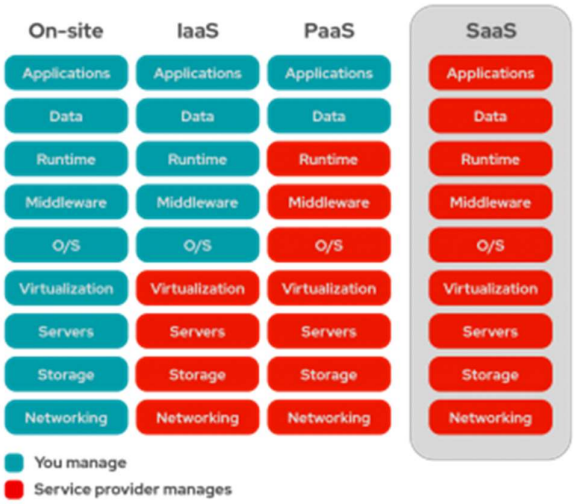


Figura 7: Diagrama SaaS. (Red Hat)

En el cuadro a continuación se identifican los actuales proveedores de mercado por tipo de servicio:

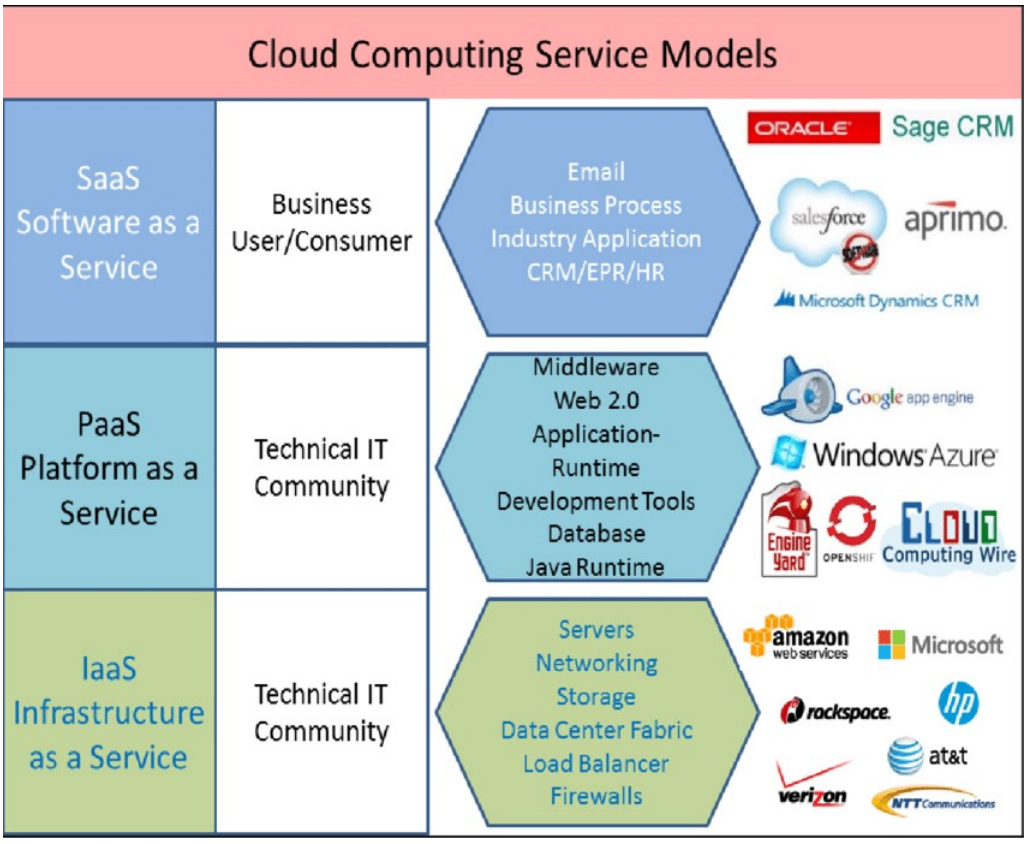


Figura 8: Modelos de servicio por tipo de servicio en Cloud Computing. (Kumar)

1.4.3.4. Infrastructure as Code (IaC)

Kief Morris (Morris) define a esta tecnología como el enfoque de la automatización de infraestructura basado en prácticas del desarrollo de *software*. Enfatiza en las rutinas consistentes y repetibles para provisionar y cambiar sistemas y sus configuraciones. El usuario realiza los cambios al código, utiliza la automatización para testear y posteriormente aplicar esos cambios a los sistemas.

Asimismo enumera los beneficios de esta tecnología, resumiendo en que las organizaciones adoptan a *Infrastructure as Code* para gestionar dinámicamente la infraestructura, obteniendo los siguientes beneficios:

- Utilizar la infraestructura de IT como un facilitador rápido de la entrega de valor.
- Reducir el esfuerzo y riesgo de hacer cambios a la infraestructura.
- Permitir a los usuarios de la infraestructura obtener los recursos que necesitan, cuando los necesitan.
- Proveer las mismas herramientas a todos: desarrolladores, operaciones y otros interesados.
- Crear sistemas que son confiables, seguros y costo-efectivos.
- Llevar adelante los controles de *governance*, seguridad y *compliance* de forma visible.
- Mejorar la velocidad de resolución de problemas y fallas.

Finalmente, define las 3 principales prácticas al trabajar en esta tecnología:

- **Definir todo como código:** Las claves son la reusabilidad, la consistencia y la transparencia.
- **Continuamente testear e implementar todo el trabajo en progreso:** La idea es construir calidad en vez de intentar testear la calidad.
- **Construir piezas pequeñas y simples:** Fundamentalmente con el objetivo de reducir el acoplamiento y las trabas a los cambios que se presentan en sistemas más grandes, lo cual también los hace más permeables a romperse.

En los subsiguientes subcapítulos se procederá a definir brevemente cada una de las tecnologías que serán eje fundamental de este trabajo de tesis.

1.4.3.4.1. Configuration Management – Ansible

Ansible, desarrollado por Red Hat (Red Hat), es un motor radicalmente simple de automatización de IT que permite automatizar provisionamiento en la nube, gestionar la configuración, implementar aplicaciones, efectuar orquestación intra-servicios, entre otros.

Diseñado para gestionar implementaciones multicapa desde sus inicios, Ansible modela la infraestructura de IT describiendo cómo todos los sistemas se interrelacionan, en vez de gestionar un solo sistema a la vez.

Según especifica la fuente, esta tecnología no utiliza agentes ni requiere ningún tipo de infraestructura de seguridad a medida, lo cual lo hace fácil de implementar – y más importante aún, utiliza un lenguaje muy simple (YAML[5], en la forma de *playbooks* de Ansible) que permiten describir los trabajos de automatización de una manera muy cercana a lenguaje natural en inglés.

Adicionalmente, su versatilidad y capacidades, entre otras, permiten:

- Subir y bajar máquinas de balanceadores de carga y ventanas de monitoreo.
- Hacer que un servidor conozca las direcciones IP de todos los demás utilizando hechos recolectados de esos servidores en particular y utilizarlas para construir dinámicamente otros archivos de configuración.
- Asignar algunas variables y solicitar otras, y asignar valores por defecto cuando no están definidas.
- Utilizar el resultado de un comando para decidir si ejecutar otro o no.

[5] YAML (https://yaml.org/): Originariamente su nombre deriva de un acrónimo en inglés que significa "Yet Another Markup Language" aunque posteriormente conservó su acrónimo pero pasó a llamarse "Ain't Markup Language". Es un lenguaje human-friendly de serialización de datos para todos los lenguajes de programación; de marcado ligero, sintaxis relativamente sencilla y diseñado teniendo en cuenta que fuera muy legible pero a la vez fácilmente *mappeable* a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel.

1.4.3.4.1.1. Arquitectura eficiente

Reformulando el texto en la fuente citada al inicio de este apartado, Ansible trabaja conectando a los nodos y aplicándoles pequeños programas, llamados módulos Ansible. Estos programas están escritos para ser modelos de recursos de los estados deseados del sistema. Ansible entonces ejecuta estos módulos (utilizando SSH[6] por defecto), y los remueve cuando finaliza.

Cabe destacar que si bien Ansible trabaja por defecto con SSH como se mencionó anteriormente, también soporta Kerberos[7].

1.4.3.4.1.2. Ansible Playbooks

Según describe la fuente, los *playbooks* pueden orquestar finamente muchas capas de la topología de infraestructura, con un control muy detallado sobre las máquinas en las que debe tomar acción en cada instante.

La solución de Ansible a la orquestación está enfocada en la simplicidad, entendiendo que el código de automatización no debería ser un obstáculo en torno a la sintaxis o características, sino enfocarse en resolver la necesidad.

Por último, a través de *playbooks*, es posible ejecutar tareas en forma asincrónica y en aquellas circunstancias donde no se ejecutan tareas en secuencia, se torna en una herramienta muy poderosa a la hora de construir y ejecutar rutinas con flujos alternativos en función de los resultados y situaciones a medida que se desarrollan.

[6] SSH (Massachusetts Institute of Technology, n.d.): Su acrónimo significa “Secure SHell”, y es el nombre del protocolo y del programa que lo implementa y cuya principal función es el acceso remoto a un servidor por medio de un canal seguro en el que toda la información está cifrada.

[7] Kerberos (Massachusetts Institute of Technology, n.d.): Es un protocolo de autenticación de redes de computadores creado por el MIT que permite a dos computadores en una red insegura demostrar su identidad mutuamente de manera segura.

1.4.3.4.1.3. Ansible Automation Platform

Para este apartado se consultaron varias fuentes. Adaptando de la página oficial de Ansible Automation Platform (Red Hat), mientras que Ansible puede ser una herramienta poderosa de gestión de configuración, puede parecer algo intimidante para los administradores de IT que no se sienten familiarizados con las herramientas de línea de comando. Ansible Automation Platform intenta resolver esto con una interfaz gráfica, y extendiendo la funcionalidad de Ansible agregando más capacidades de gestión de la infraestructura.

Ansible requiere que el usuario utilice la CLI (*Command-Line Interface*, interfaz de línea de comando). En algunas situaciones, donde esto no es la mejor opción, como en los casos donde se necesita ejecutar un trabajo en Ansible desde otro trabajo (caso en el que sería mucho más efectivo utilizar una API) o en casos donde la persona que desea ejecutar el trabajo debería estar habilitada para ejecutar sólo ese trabajo específico. Para este tipo de casos, Ansible Automation Controller (su herramienta de gestión de automatización *on-demand* escalable) es la mejor opción a utilizar.

Este producto fue diseñado para ser el concentrador (*hub*) de todas las tareas de automatización; permite controlar los accesos a los objetos e incluso permitir que se compartan credenciales SSH sin que alguien físicamente lo haga. Al mismo tiempo, los inventarios pueden ser gestionados visualmente o sincronizados con una amplia variedad de fuentes de servicios de nube. Sin dudas una de sus principales ventajas es el poder de su REST API, que además es explorable a través de un navegador.

No obstante, para los autores del libro "*Practical Ansible 2*" (Daniel Oh, James Freeman, & Fabio Alessandro Locati, 2020), otra ventaja es que cuenta con herramientas de integración con Jenkins, que luego facilitarán todas las tareas de *testing* automatizado antes de su implementación al ambiente productivo.

La arquitectura de Ansible Automation Platform se beneficia de los desarrollos en tecnología de contenedores. Es más escalable y seguro que su generación anterior y su diferencia más grande es

que desacopla la capa de control de los ambientes de ejecución, como se puede visualizar en la imagen debajo:

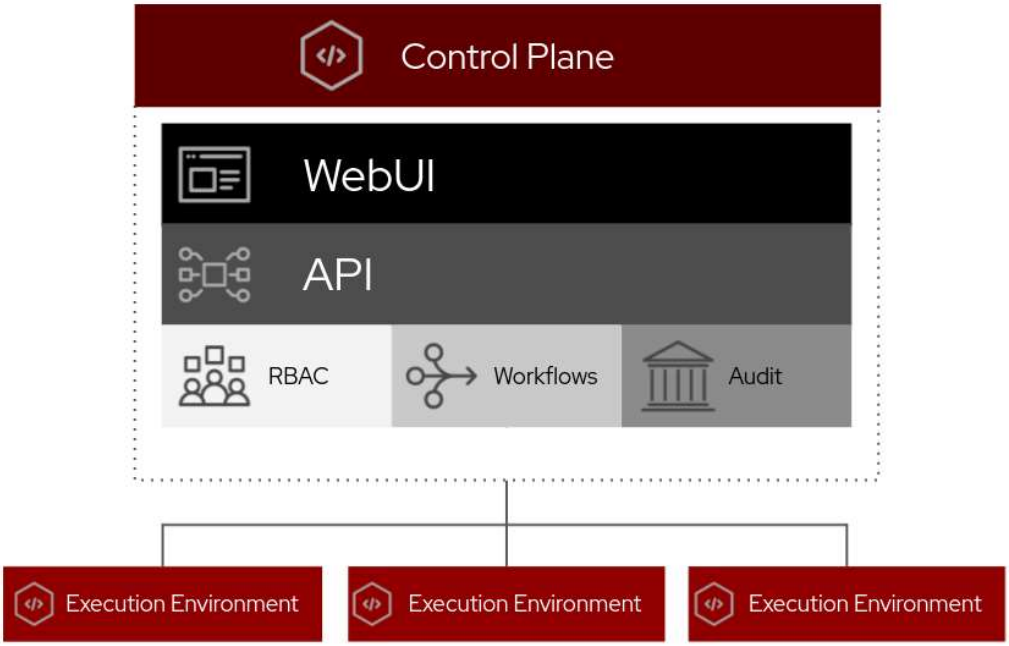


Figura 9: Arquitectura de capas de Ansible Automation Platform. (Prog.World)

Nota: RBAC (*Role Based Access Control*) es un enfoque para restringir el acceso al Sistema a los usuarios autorizados y sólo en función de su rol específico dentro del mismo.

Así, por otro lado (y coincidiendo con Red Hat) los autores de "*Ansible 2 Cloud Automation Cookbook*" (Aggarwal, Aditya Patawari, & Vikas) y de "*Ansible: Up and Running, 3rd edition*" (Bas Meijer, Lorin Hochstein, & René Moser) acuerdan en este aspecto como una ventaja determinante al introducir la ejecución de automatización a nivel de ambiente, que significa la ejecución de automatización en imágenes de contenedores; con la posibilidad de hacerlo a través de su interfaz gráfica o también (como se comentó anteriormente) a través de un uso más granular efectuado por Automation Controller con RBAC y una REST API. Cabe destacar que la REST (*Representational State Transfer*) API que expone la plataforma permite integrar con *pipelines* de CI/CD.

De esta manera, logra extender la funcionalidad de Ansible agregando control de accesos, proyectos, gestión de inventario y la posibilidad de ejecutar trabajos en forma de modelos (*template jobs*).

A continuación se adaptan brevemente cada uno de ellos según se describen en el capítulo 23 de "*Ansible: Up and Running, 3rd edition"*:

- **Control de Accesos:** Implementa RBAC bajo el concepto de separación de funciones (separation of duties), con el objetivo de tener más de una persona requerida para completar una tarea. Esto sirve a efectos de incrementar el control y mejorar la seguridad, previniendo fraudes, sabotajes, robos y mal uso de la información. Dicho esto, la plataforma actúa como un guardia activo, ningún equipo o empleado requiere tener acceso directo a los *hosts* que gestiona.
- **Proyectos:** Su estructura está compuesta por un repositorio que contiene *playbooks* y roles lógicamente vinculados. Si bien en los proyectos clásicos de Ansible los inventarios estáticos son a menudo mantenidos junto a los *playbooks* y roles, Ansible Automation Platform maneja los inventarios por separado. Todo lo relacionado a inventarios y variables de inventarios que se mantiene dentro de los proyectos, como variables de grupos y de *hosts* no serán accesibles luego. Es importante destacar como mejor práctica el mantener los proyectos bajo esquemas de control de versiones.
- **Gestión de inventario:** Como se indicó anteriormente, la plataforma permite gestionar inventarios como recursos dedicados, incluyendo la gestión del control de accesos. Como mejor práctica se propone que los *hosts* de producción, *staging* y pruebas estén en inventarios separados. A cada uno de estos inventarios se les adicionará luego variables por defecto y se podrá agregar manualmente luego tanto grupos como *hosts*.
- ***Template jobs*:** Estos vinculan proyectos con inventarios y definen cómo los usuarios están habilitados para ejecutar un *playbook* de un proyecto a destinos específicos de un

determinado inventario. La herramienta permite efectuar ciertos refinamientos a nivel de *playbook*, tales como agregar parámetros adicionales y *tags*.

1.4.3.4.2. Contenedores – Docker

En la experiencia del maestrando y adaptando de (Red Hat), Docker es una herramienta que hace que adoptar mejores prácticas de empacado del *software*, su distribución y utilización sean económicas y simples. Esto lo logra proveyendo una visión completa de contenedores de procesos y herramientas simples para construirlos y trabajar en ellos.

Es un proyecto *open-source* (de código abierto) que permite construir, transportar y ejecutar programas. Es un programa de línea de comando, un proceso que se ejecuta en *background* (por detrás de lo que ve el usuario), y un conjunto de servicios remotos que constituyen un enfoque logístico para resolver problemas comunes de *software* y simplificar la experiencia de instalación, ejecución, publicación y remoción de *software*. Y todo esto lo logra utilizando una tecnología de sistema operativo llamada **contenedores**.

Por lo arriba expuesto, y agregando lo enunciado por los autores de "*Docker in* action", podría considerarse a Docker como un proveedor de logística de *software* que permitirá ahorrar tiempo, recursos y permitir poner foco en las competencias centrales del negocio. Se puede utilizar con aplicaciones de red como servidores web, bases de datos y servidores de correo, y por qué no también con aplicaciones de usuario que incluyen editores de texto, compiladores, herramientas de análisis de red y *scripts*; en algunos casos, incluso se utiliza para ejecutar aplicaciones de interfaz como exploradores *web* y *software* de productividad.

Continuando con lo interpretado en dicha fuente bibliográfica, los autores sostienen que Docker no es un lenguaje de programación y tampoco es un *framework* para construir *software*. Más bien es una herramienta que ayuda a resolver problemas comunes como instalación, remoción,

actualización, distribución, seguridad y ejecución de *software*; con la ventaja de además ser *open-source*, lo cual significa que cualquier persona puede contribuir, lo cual hace que toda la comunidad se beneficie de ello.

Una de las características más poderosas de Docker es que permite operar servicios de *software* con escalamiento dinámico, lo cual es fundamental a la hora de apalancarse en las ventajas de las tecnologías de la nube, especialmente en la maximización del uso del hardware y consiguientemente de la reducción de costos aparejada. Esto es porque los contenedores se ponen en servicio más rápido y consumen menos recursos que las máquinas virtuales.

Los autores sostienen que resulta ventajoso en aquellos casos donde los equipos de desarrollo de *software* implementen técnicas de CI/CD[8] pueden construir *pipelines* más robustos y ambientes de testeo más funcionales si utilizan Docker, punto en el que también coincide el maestrando. Los contenedores testeados tienen el mismo *software* que luego será implementado en el ambiente productivo; lo cual se traduce en más confianza en la introducción de cambios a producción, controles más ajustados de cambios al ambiente y por último: iteraciones más rápidas.

En el caso de los ambientes locales de desarrollo, utilizar Docker permitirá disminuir los tiempos de incorporación de nuevos integrantes de equipo y al mismo tiempo eliminar las inconsistencias que los retrasan, teniendo en cuenta que estos mismos ambientes además permiten gestionar versionados de código.

Distribuir el *software* a través de Docker es más sencillo de instalar y ejecutar, aprovechando las herramientas de configuración y cualquier material adicional que el desarrollador quiera agregar.

[8] CI/CD (Red Hat, 2018): Del acrónimo Continuous Integration/Continuous Deployment, se refiere a las prácticas combinadas de integración continua y entrega continua. El proceso contrasta con los métodos tradicionales, en los que una colección de actualizaciones de software se integraba en un gran lote antes de desplegar la nueva versión. Las prácticas modernas de DevOps implican el desarrollo continuo, las pruebas continuas, la integración continua, el despliegue continuo y la supervisión continua de las aplicaciones de software a lo lasrgo de su ciclo de vida de desarrollo. Estas prácticas constituyen la columna vertebral de las operaciones de DevOps actuales. Dentro de nuestro modelo de solución detallado en las conclusiones de este capítulo, esta función es cubierta por la herramienta Jenkins.

Esto también permite reducir a la mínima expresión la típica guía de instalación del *software*, transformándose en una sola línea de comando y una única dependencia portable.

Para los autores, cabe destacar que si bien los desarrolladores de *software* entienden de dependencias, instalación y empaquetado, son los administradores de sistemas los que entienden de los sistemas donde el *software* será ejecutado. En este aspecto, Docker provee un lenguaje expresivo para ejecutar *software* en contenedores. Este lenguaje permite a los administradores de sistemas inyectar configuraciones específicas para el ambiente en cuestión y al mismo tiempo controlar minuciosamente el acceso a los recursos. Todo esto, en conjunto, sumado a la capacidad de gestionar los empaquetados en forma nativa dentro de Docker, las herramientas y la distribución de infraestructura, resultan en implementaciones declarativas, repetibles y confiables. Promueve paradigmas de sistemas descartables, aislamiento de estados persistentes y otras mejores prácticas que alivian las tareas de los administradores de sistemas.

1.4.3.4.2.1. Flujo típico de ejecución de Docker

Los autores de "*Docker in Action"* dedican un apartado específico en el primer capítulo del mencionado libro a describir un flujo típico de ejecución de Docker. A continuación se muestra un flujo de ejecución típico por primera vez:

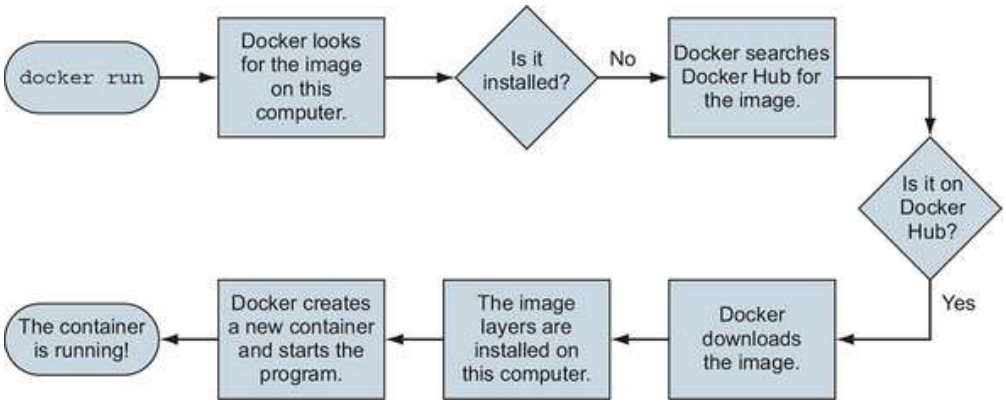


Figura 10: Flujo de ejecución de pedido de imagen inexistente en Docker Registry. (Jeffrey Nickoloff & Stephen Kuenzli)

Se solicita la ejecución de una aplicación, que resulta no estar instalada en el sistema. Docker a continuación procede a buscarla en el repositorio, instalarla en un contenedor nuevo e iniciarlo.

A continuación, el flujo de ejecución del mismo comando que antes descrito por los autores, solicitando la misma aplicación, que ahora resulta estar instalada:

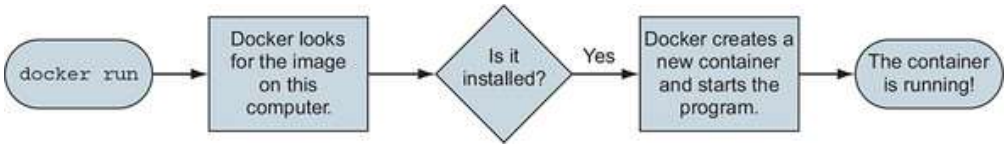


Figura 11: Flujo de ejecución de pedido de imagen existente en Docker Registry. (Jeffrey Nickoloff & Stephen Kuenzli)

1.4.3.4.2.2. Los contenedores no son virtualización

Continuando, (Jeffrey Nickoloff & Stephen Kuenzli) afirman que la gente tiende a pensar en las máquinas virtuales como unidades de implementación, donde poner productivo un proceso implica crear una máquina virtual dedicada y conectarla a la red. Las máquinas virtuales proveen *hardware* virtual (o *hardware* en el cual un sistema operativo y otros programas pueden ser instalados). Éstos toman un largo tiempo en ser creados y requieren una cantidad significativa de sobre dimensionamiento y por ende gastos en asignación de recursos físicos de hardware, incluyendo a veces licencias, debido a que necesitan soportar un sistema operativo completo en conjunto con el software que se supone ejecutarán.

Este trabajo las máquinas virtuales lo hacen de manera óptima, una vez que están funcionando correctamente y todo ha sido configurado, pero los retrasos acumulados a lo largo de

todo el proceso de provisionamiento de la misma las tornan en un pobre intento de JIT (*Just-In-Time*) o sólo aptos para escenarios de implementación/provisionamiento reactivos. Esto las convierte en obsoletas en un paradigma como el que plantea la computación en la nube y dificulta la escalabilidad y dinamismo que propone.

Por otro lado, los contenedores de Docker no utilizan ningún tipo de virtualización de *hardware*. Los programas se ejecutan dentro de los contenedores mismos e interactúan a través de una interfaz directamente con el kernel del sistema operativo. La ventaja es que los programas pueden ejecutarse aislados sin tener que depender de sistemas operativos redundantes o sufrir por las demoras en las secuencias de arranque (boot) de los mismos. Por ende, y siendo una distinción de gran importancia, Docker no es una tecnología de virtualización de *hardware* sino que ayuda a utilizar tecnología de contenedores ya construido dentro del kernel del sistema operativo.

Finalmente, concluyen, las máquinas virtuales proveen abstracciones de hardware para que puedan ejecutarse sistemas operativos. Los contenedores, en cambio, son una prestación más del sistema operativo.

1.4.3.4.2.3. Evolución de los paradigmas de virtualización hacia tecnología de contenedores

El siguiente diagrama pretende ilustrar la diferenciación y evolución del provisionamiento de servicios, comprendiendo los esquemas físicos (a), virtuales (b) y contenedores (c):

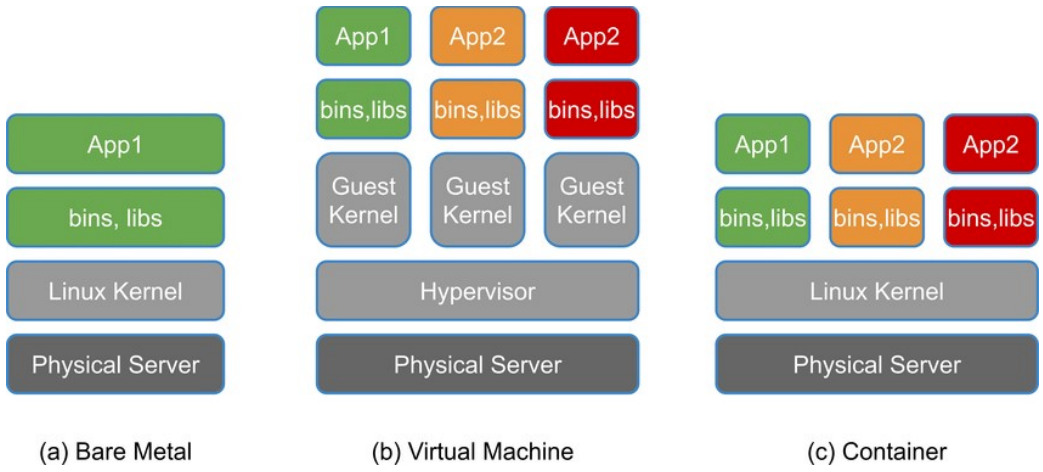


Figura 12: Diferenciación y evolución de provisionamiento de servicios. (Takahashi)

Del gráfico resulta sencillo identificar la simplificación de complejidad y optimización de recursos que propone la tecnología.

Se procederá a mostrar a continuación cómo funciona a nivel procesamiento en ambos escenarios de tecnología (físico/virtual vs contenedores), para lo cual se agregará además otra fuente, en este caso de Nigel Poulton en su libro "*Docker Deep Dive*" (Nigel Poulton).

Esquema de un computador tradicional (físico/virtual) en el cual se ejecuta una aplicación:

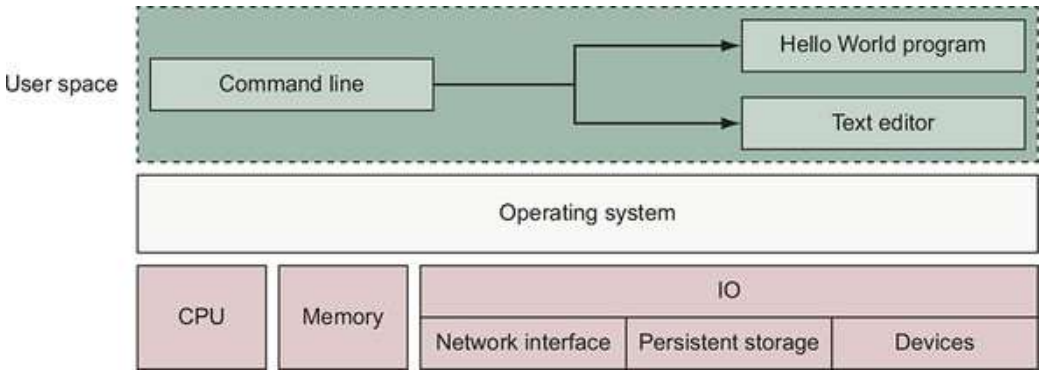


Figura 13: Diagrama en bloques de ejecución de procesos en físicos y virtuales. (Jeffrey Nickoloff & Stephen Kuenzli)

Esquema en tecnología de contenedores, para la misma ejecución:

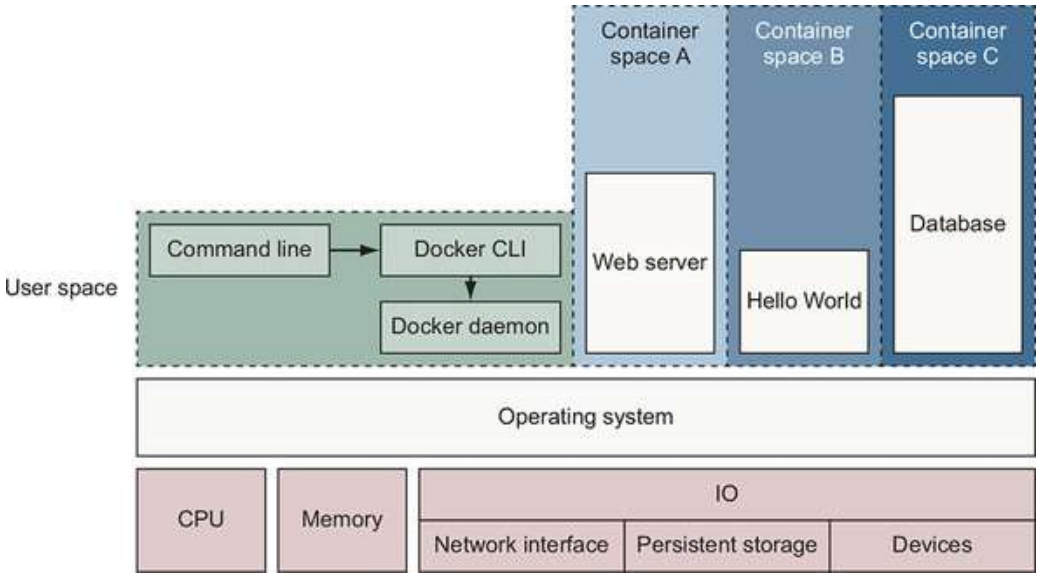


Figura 14: Diagrama en bloques de ejecución de procesos en contenedores. (Jeffrey Nickoloff & Stephen Kuenzli)

En este caso, (Jeffrey Nickoloff & Stephen Kuenzli) ilustran 3 contenedores, cada uno con distintas funcionalidades, todas sirviéndose de la misma plataforma de contenedores.

Por su lado, y antes de continuar al siguiente concepto, es importante destacar que existe otro modelo-principio que propone Nigel Poulton en este caso en el capítulo 10 de su libro, y es el de "*swarm cluster*": "*On the clustering front, a swarm consists of one or more Docker nodes. These can be physical servers, VMs, Raspberry Pi's, or cloud instances. The only requirement is that all nodes have Docker installed and can communicate over reliable networks. Nodes are configured as managers or workers. Managers look after the control plane of the cluster, meaning things like the state of the cluster and dispatching tasks to workers. Workers accept tasks from managers and execute them.*" (Nigel Poulton)

Lo que este concepto propone es la construcción de un enjambre de contenedores orquestados de forma tal de poder mejorar el rendimiento del conjunto, y según describe el autor, la mejor parte de esto es que sucede tan exitosamente que resulta imperceptible para el usuario final.

Al mismo tiempo, menciona que del lado de la orquestación de aplicación, la unidad atómica de programación dentro de un enjambre es el servicio y que algunas de las ventajas obtenidas por su implementación incluyen escalamiento, implementación de actualizaciones y facilitar los *rollbacks* en caso de necesitarlo.

Volviendo al punto anterior a la utilización de enjambres, es oportuno mencionar, indican Nickoloff y Kuenzli, también el resaltar la ventaja que propone la "contenedor-ización" de servicios y aplicaciones a la hora de hablar de seguridad. En un esquema donde un servidor físico o virtual comparte el mismo sistema operativo para todas sus aplicaciones, información y servicios, esto deja vulnerable a todos ellos ante un ataque o virus. Mientras que en enfoque de contenedores, cada contenedor maneja sus propios recursos y es completamente estanco en sí mismo (en un esquema de jaula), siendo imposible que este tipo de programas maliciosos infecten otros contenedores como se muestra a continuación:

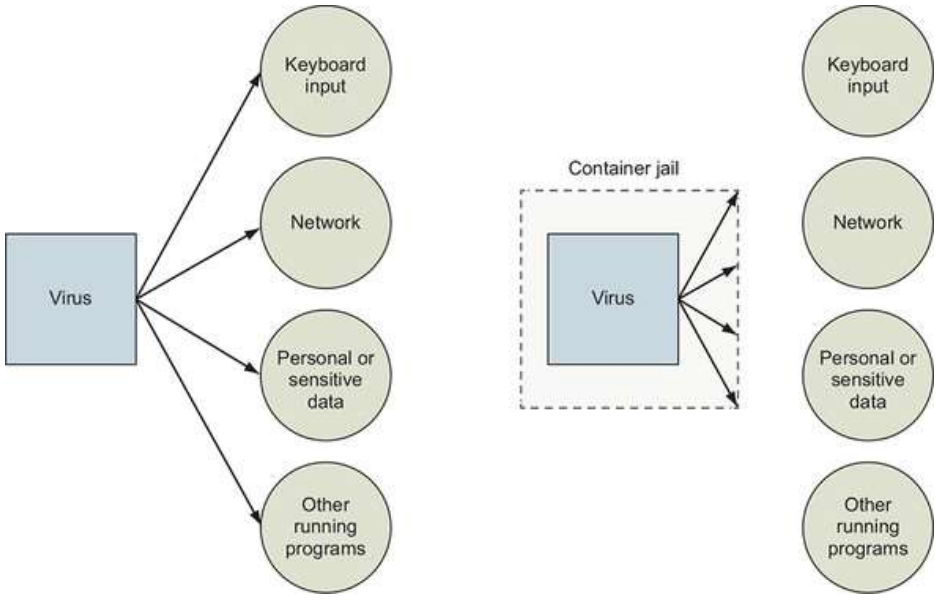


Figura 15: Comparación de aislamiento sin y con contenedores. (Jeffrey Nickoloff & Stephen Kuenzli)

1.4.3.4.2.4. Por qué Docker es importante

Nickoloff y Kuenzli dedican un apartado completo a lo que entienden como la relevancia de Docker, atribuyendo en parte esto a que Docker provee abstracción, y éstas permiten trabajar en cosas complejas en términos simples.

Como una grúa que carga contenedores dentro de un barco de carga (de ahí su característico logo), el proceso de instalación de cualquier software con Docker es idéntico a cualquier otro. El tamaño o forma de la "cosa" dentro del contenedor puede variar, pero la forma en la cual la grúa los levanta y deposita en el barco será siempre la misma. Las herramientas son todas reutilizables por cualquier contenedor.

Interpretando a los autores, esto también aplica al caso de remoción, cuando se desea remover un *software*, simplemente se le pide a Docker cuál remover. No quedarán artefactos residuales porque todos se encontraban dentro del mismo contenedor del cual Docker tenía conocimiento. Así, el sistema operativo quedará limpio como antes de la instalación del *software*.

La abstracción de contenedores y las herramientas que provee Docker para trabajar con los mismos han cambiado el paisaje de los administradores de sistemas y del desarrollo de *software*. Docker es importante porque deja disponibles los contenedores para cualquiera y al utilizarlo ahorra tiempo, dinero y energía.

Otra razón por la cual Docker es importante, explican, es que existe suficiente presión en la comunidad del *software* hacia la adopción de contenedores y Docker. Es tan fuerte que las compañías como Amazon, Microsoft y Google todas han trabajado en conjunto para contribuir a su desarrollo y adopción en sus propias ofertas de servicios en la nube.

Otra razón por la cual Docker es importante es que ha logrado para las computadoras lo que los *App Stores* hicieron por los servicios de telefonía móvil. Ha hecho la instalación de software, compartimentación y remoción simples. Mejor aún, lo hace cros-plataforma y de forma abierta. O mejor aún, en el mismo ejemplo, imagínese a todas las grandes compañías de telefonía móvil compartiendo el mismo App Store. Eso es lo que Docker logró, borrando la línea entre los sistemas operativos que existió por décadas, lo cual elegir uno para montar el negocio y servicios ya no será un factor relevante.

Finalmente, concluyen atribuyendo su estabilidad y "securitización" a través de lo expuesto en el subcapítulo anterior, afirmando que está empujando más y más a las empresas hacia un esquema de contenedores debido a las bondades que brinda el aislamiento entre ellos.

1.4.3.4.2.5. Conclusiones y próximos pasos

Para Nickoloff y Kuenzli hoy el ecosistema de contenedores es rico a través de herramientas que solucionan problemas nuevos o de alto nivel. Estos problemas incluyen la orquestación de contenedores, "clusterización" de alta disponibilidad, gestión del ciclo de vida de microservicios y visibilidad. Puede ser dificultoso navegar el mercado sin depender de asociación de algunos términos pero resulta más dificultoso aún entender cómo Docker y esos productos trabajan en conjunto.

Estos productos trabajan con Docker en la forma de *plugins* o proveen algún tipo de funcionalidad de alto nivel y dependen de Docker. Algunas herramientas sólo utilizan algunos de sus subcomponentes. Estos subcomponentes son proyectos independientes.

Por otro lado, los autores indican que Kubernetes, proyecto del cual nos expandiremos en el siguiente subcapítulo, es el proyecto más notable en el ecosistema sin contar a Docker. Provee una plataforma extensible para orquestación de servicios como contenedores en ambientes "clusterizados" y se encuentra creciendo hacia una suerte de sistema operativo de Data Centers. Pero

éste depende de un motor de contenedores, como Docker, y así los contenedores e imágenes que se construyen se ejecutarán en Kubernetes.

El maestrando agrega a los autores que en su experiencia se necesitan considerar varios "*trade-offs*" cuando se elige cualquier herramienta. Kubernetes se potencia en su extensibilidad, pero al costo de su curva de aprendizaje y los esfuerzos requeridos para darle soporte. Hoy construir, adaptar o extender *clusters* de Kubernetes es un trabajo de día completo. No obstante, utilizar los *clusters* existentes de Kubernetes para implementar aplicaciones es bastante sencillo con muy poca investigación.

1.4.3.4.3. Orquestación de Contenedores – Kubernetes

Como se expresó en el subcapítulo anterior acerca de los contenedores, éstos han traído una flexibilidad tremenda a las organizaciones, pero han permanecido cuestionados por un largo tiempo respecto del desafío que representaba implementarlos en ambientes productivos. Por años, indican Kebbani, Tylenda y McKendrick en su libro "*The Kubernetes Bible"*, las compañías utilizaron contenedores para laboratorios, ambientes de prueba y desarrollo, desarrollo local y similares; pero el uso de contenedores para ambientes productivos era inconcebible para muchas organizaciones. Desde sus inicios, fue pensado como una solución para implementar una enorme cantidad de contenedores en infraestructuras masivamente distribuidas.

Para ellos, la orquestación de contenedores fue el *game-changer*, y fue sin dudas Kubernetes quien estuvo al frente del desafío.

Originalmente construido por Google, Kubernetes es hoy el orquestador de contenedores líder del mercado, proveyendo todas las características necesarias para implementarlos en ambientes productivos a escala.

Algo en lo que todos los autores revisados y expertos consultados coinciden, es en que Kubernetes es tan bueno y popular como complejo. Aterrizar en esta herramienta tan versátil es sumamente sencillo, pero convertirse en usuario avanzado no es tarea fácil ya que no es una herramienta fácil de aprender y operar.

Para Luksa, autor de "*Kubernetes in Action*", como orquestador Kubernetes tiene sus propios conceptos independientes de cualquier otro motor de contenedores, como Docker. Pero, cuando son utilizados en conjuntos, se obtiene una plataforma muy fuerte y robusta lista para implementar aplicaciones *cloud-native* en producción.

1.4.3.4.3.1. Explorando los problemas que resuelve Kubernetes

Es evidente que lanzar contenedores en una máquina local o en un ambiente de desarrollo no requerirá el mismo nivel de planificación como lanzar los mismos en máquinas remotas o servidores productivos, los cuales podrían ser impactados potencialmente por millones de usuarios.

En este aspecto, Kebbani, Tylenda y McKendrick dedican un apartado específico en el primer capítulo de su libro a explorar estos problemas y se describirán a continuación.

Problemas específicos a producción aparecerán, comentan, para los cuales Kubernetes está diseñado resolver, tales como:

- **Aseguramiento de alta disponibilidad:** La arquitectura de microservicios es la manera de mitigar el riesgo de una caída total ante un evento de falla. Al utilizarlos, la falla de un microservicio individual no afectará la estabilidad general de la aplicación. En este sentido, Kubernetes incluye una batería completa de funcionalidades para hacer que los contenedores de Docker de alta disponibilidad repliquen en varios *hosts*, monitoreando su estado de salud frecuentemente. Es de vital importancia entender que la accesibilidad de la aplicación dependerá directamente de la salud de los contenedores. En caso de que algún contenedor estuviera inaccesible, Docker por sí solo no es capaz de destruir el

contenedor y levantar uno nuevo; pero Kubernetes sí, ya que posee la capacidad de reparar automáticamente aplicaciones a través de tareas automatizadas como revisiones de salud y reemplazo de contenedores (concepto de "*pave and repave*"). Si una máquina en el *cluster* fallara, todos los contenedores ejecutándose en ella desaparecerían. No obstante, Kubernetes inmediatamente lo notaría y reprogramaría la construcción de todos ellos en otra máquina, logrando así alta disponibilidad y tolerancia a fallas.

- **Manejar la gestión de lanzamientos (*release management*) e implementación de contenedores:** En este aspecto, los autores afirman que el objetivo es el de mantener la aplicación en producción actualizada, reemplazando versiones antiguas de un microservicio con una versión nueva. Esto supone un desafío desde el punto de vista de la no disponibilidad durante la ventana de mantenimiento/actualización de la aplicación en producción. En caso de fallar, el cambio tiene que volverse atrás y nuevamente levantar el ambiente con la versión anterior. Todo esto sin perder de vista que se espera que las implementaciones necesitan ser lo menos visibles posible para el usuario final, con la menor fricción posible. Kubernetes permite gestionar las implementaciones y *rollbacks* de contenedores de Docker a través de un único comando. De esta manera, irá paulatinamente reemplazando contenedores "viejos" con otros "nuevos" ya ejecutando versiones nuevas de los microservicios en todas las máquinas.
- **Auto-escalamiento de contenedores:** Interpretando a los autores, en este aspecto el escalamiento es otro problema específico de producción que ha sido vastamente tratado entre los proveedores de nube pública más relevantes del mercado. Se trata de la habilidad de adaptar el poder computacional a la carga que enfrenta la aplicación (siempre manteniendo el criterio fundamental de alta disponibilidad). Cuando las máquinas en producción enfrentan picos de tráfico y uno de los contenedores ya no está disponible para enfrentar la carga, se debe encontrar una manera de identificar el

contenedor que falla. La decisión en este punto debe ser de escalar vertical u horizontalmente; de otra manera, si no se toma una decisión y la carga no decrece, el contenedor o incluso la máquina *host* podría fallar y la aplicación podría volverse inaccesible:

- **Escalamiento vertical:** Esto permite al contenedor utilizar más potencia computacional ofrecida por la máquina *host*.
- **Escalamiento horizontal:** Puede duplicar el contenedor a otra máquina, y puede balancear el tráfico entre los dos contenedores.

Nuevamente, Docker no es capaz de responder a este problema por sí solo; pero en combinación con Kubernetes sí, ya que el segundo es capaz de gestionar ambos tanto el escalamiento vertical como el horizontal automáticamente, ya sea ampliando su capacidad computacional o mismo permitiendo crear contenedores adicionales que pueden ser implementados en otro nodo del *cluster*. Incluso en el caso en que Kubernetes no pudiera manejar más contenedores porque todos los nodos están llenos, podría ser capaz de lanzar nuevas máquinas virtuales conectándose con el proveedor de nube de forma automatizada y transparente a través de un componente llamado *Cluster Autoscaler*.

1.4.3.4.3.2. Arquitectura de Kubernetes

En este apartado se examinará cómo Kubernetes habilita la gestión de contenedores distribuidos en diferentes máquinas tal y como se describe en "*Kubernetes: Up and Running, 3rd Edition*" (Brendan Burns, Joe Beda, & Kelsey Hightower and Lach).

Respecto de la anatomía de sus *clusters*, Kubernetes está constituido por varios componentes distribuidos, cada uno de los cuales juega un rol específico en la ejecución de los contenedores de Docker. Para entender el rol de cada uno de sus componentes, se seguirá el ciclo de vida de un

contenedor Docker como es creado y gestionado por Kubernetes. Esto es, desde el momento en que se ejecuta el comando de creación del mismo hasta el punto donde es realmente ejecutado en una máquina que es parte del *cluster* de Kubernetes.

Para poder ejecutar la herramienta, se requerirá de máquinas con Linux, llamadas **nodos**. Un nodo puede ser una máquina virtual en un proveedor en la nube o física.

Según explican los autores del libro "*The Kubernetes Bible*" (Nassim Kebbani, Piotr Tylenda, & Russ McKendrick) en el segundo capítulo, existen 2 tipos de nodos: *Master* y *Worker*. Mientras los primeros son responsables de mantener el estado del *cluster*, los segundos son responsables de ejecutar los contenedores de Docker.

Y así, como se ilustra debajo, es como se ejecuta un flujo de comandos típico:

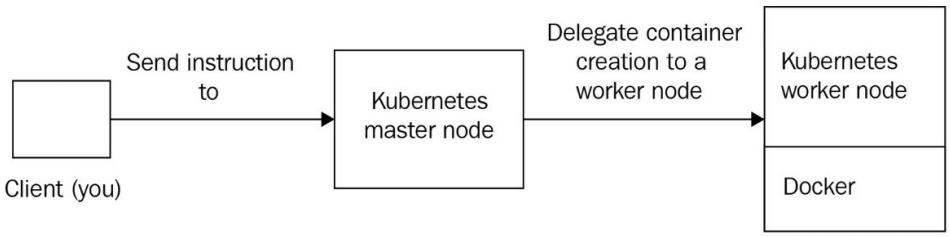


Figura 16: Flujos de ejecución de comandos en Kubernetes. (Nassim Kebbani, Piotr Tylenda, & Russ McKendrick)

Como se puede visualizar, el cliente interactúa con el nodo Master y sus componentes del plano de control (*Control Plane*), los cuales delegan la creación del contenedor a un nodo *Worker*. Asimismo, no existe comunicación entre el cliente y el nodo *Worker*.

Una de las ventajas que propone el modelo de *Control Plane* es que sus componentes pueden ser implementados en distintas máquinas, con el propósito de alcanzar el máximo nivel de tolerancia a fallas. De esta manera, cada una de ellas tiene la responsabilidad de ejecutar un único componente que permite la gestión del *cluster*.

En el siguiente ejemplo, todos los componentes del *Control Plane* están instalados en una máquina con un único nodo Master:

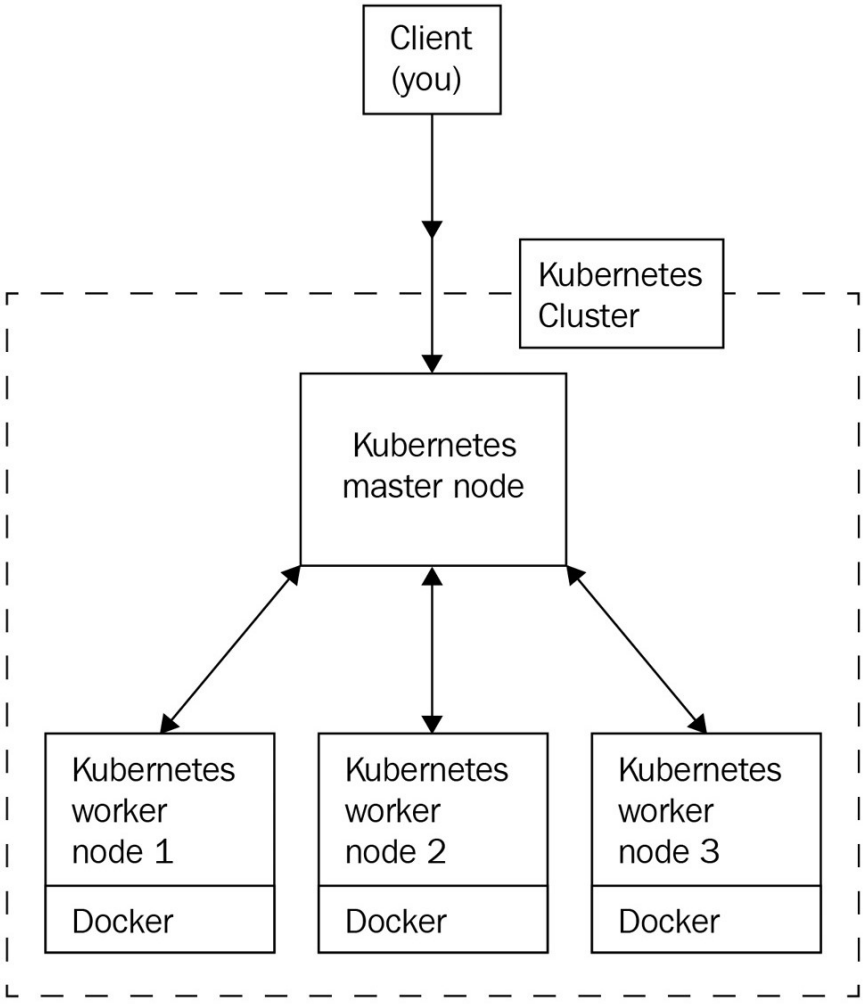


Figura 17: Diagrama de componentes del Control Plane en nodos Master únicos. (Nassim Kebbani, Piotr Tylenda, & Russ McKendrick)

1.4.3.4.3.3. Concepto de pods

Como mencionamos al comienzo de este tema, Kubernetes fue desarrollado con la experiencia ganada por Google para gestionar contenedores en producción. Más importante aún, heredó ideas de Borg y Omega, conceptos y arquitectura.

Entre otros elementos, trajo consigo el concepto de *pod*, y poseen las siguientes características, cubiertas por Kebbani, Tylenda y McKendrick:

- Los *pods* son un tipo de recurso que permite gestionar los contenedores, Kubernetes utiliza un objeto lógico llamado *pod*, para crear, actualizar y eliminar los contenedores.
- Cada *pod* tiene su propia dirección IP en el *cluster*.
- Existen componentes distribuidos que observan a la API central de Kubernetes con el objetivo de recuperar el estado del *cluster*.
- Existe un balanceo de carga interno entre los *pods* y los servicios.
- Las etiquetas y selectores son dos metadatos utilizados juntos para construir interacción entre Kubernetes.

Diagrama de flujos de creación de pods en Kubernetes:

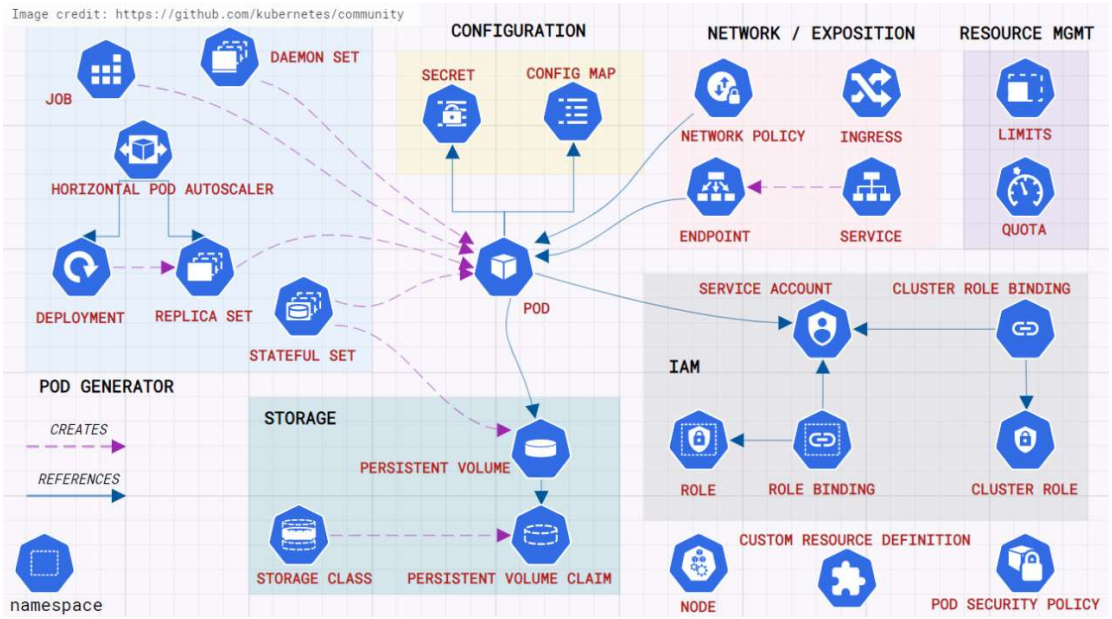


Figura 18: Diagrama de componentes intervinientes en la creación de pods de Kubernetes.

(CloudSkew)

1.4.3.4.3.4. Arquitectura típica de una solución basada en Kubernetes

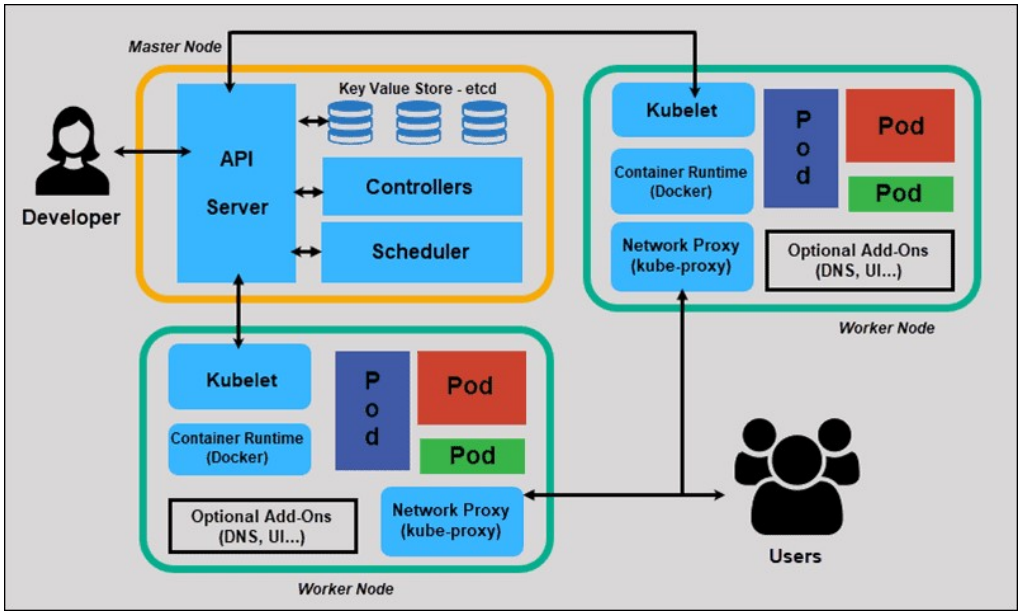


Figura 19: Diagrama en bloques de arquitectura de solución basada en Kubernetes

(Kaplarevic).

1.4.3.4.4. Provisionamiento de Infraestructura - Terraform

Para este apartado de Terraform se recurrre a la página oficial del producto como fuente (Terraform), y define a esta herramienta de IaC como una que permite definir soluciones tanto *on-prem* como en la nube, utilizando archivos de configuración escritos en un lenguaje casi natural; capaces de ser versionados, reutilizados y compartidos. A través de flujos de trabajo (*workflows*), esta herramienta permite provisionar y gestionar toda la infraestructura a lo largo de su ciclo de vida.

Puede gestionar componentes tanto de bajo nivel, tales como computacionales, almacenamiento y recursos de conectividad, como de alto nivel, tales como entradas de DNS y características de SaaS.

Al mismo tiempo, el propietario manifiesta también que crea y gestiona recursos en plataformas de nube y otros servicios a través de su API. Los proveedores habilitan a Terraform para trabajar con virtualmente casi cualquier plataforma o servicio con una API disponible.

Figura 20: Diagrama en bloques de flujos entre el Terraform Provider y la API destino.

(Terraform)

Finalmente, el workflow principal de terraform consiste de 3 etapas:

- ***Write:*** Se definen recursos, los cuales pueden estar distribuidos entre múltiples proveedores y servicios de nube.
- ***Plan:*** Terraform crea un plan de ejecución describiendo la infraestructura que creará, actualizará o destruirá basada en la infraestructura existente y la nueva configuración propuesta.
- ***Apply:*** Una vez aprobado, Terraform ejecuta las operaciones propuestas en el orden correcto, considerando las dependencias entre los recursos.

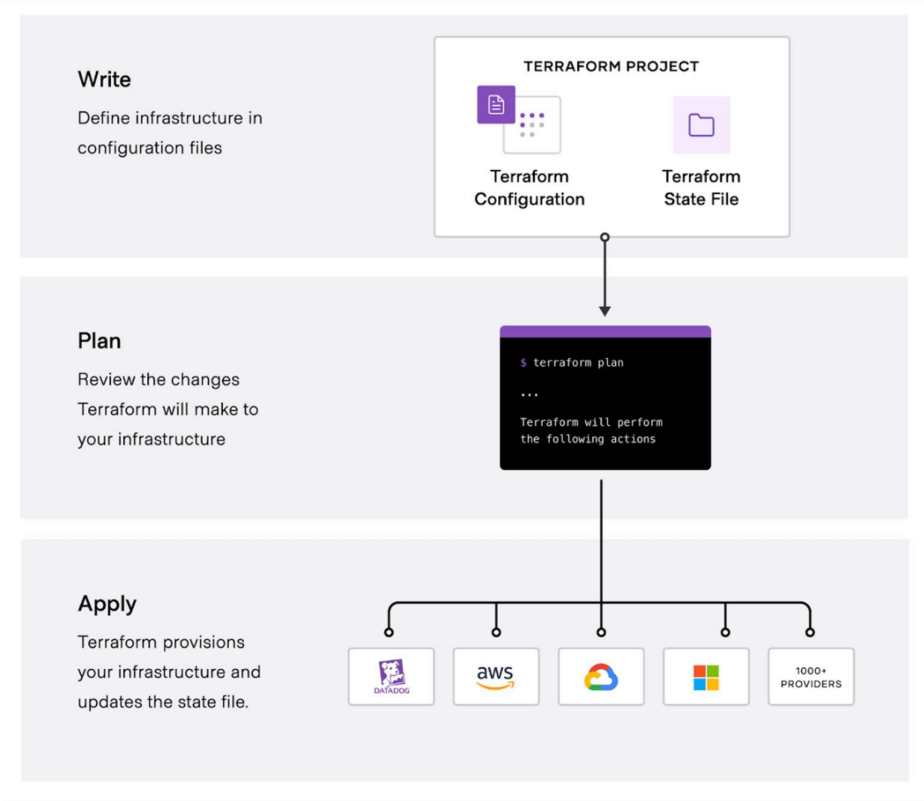


Figura 21: Diagrama de etapas de un workflow de Terraform. (Terraform)

Durante este apartado se presentó a Terraform como una opción al provisionamiento de infraestructura, pero a efectos prácticos no será utilizado dentro del conjunto de tecnologías de la solución a desarrollar en próximos capítulos de este trabajo de tesis puesto que sus funciones ya son cubiertas por Ansible Automation Platform.

1.4.4. Conclusiones

Como se ha descrito hasta acá en el apartado de *Cloud Computing*, son muchas las ventajas que brinda este paradigma de gestión y provisionamiento de servicios de infraestructura. Fundamentalmente, en la opinión del maestrando, implica comparar 2 eras; tal cual las describe Kief Morris (la de hierro y la de la nube), con las siguientes particularidades características:

Edad de hierro	Edad de la nube
Hardware físico	Recursos virtualizados
El provisionamiento toma semanas	El provisionamiento toma minutos
Procesos manuales	Procesos automatizados

Tabla 1: Diferencias entre Data Centers *legacy* y la nube. (Kief Morris)

Steve Swoyer, en su libro "*Migrating Applications to the Cloud*" (Swoyer), describe en el primer capítulo del mismo (*The Cloud and Business Transformation*) el objetivo de la migración hacia esta plataforma; el cual no es sólo mover estas aplicaciones a la nube sino transformar la organización y sus operaciones. Lo cual también acarrea no sólo sus aplicaciones y arquitecturas de *software* (como por ejemplo al transportar las aplicaciones, servicios y también las cargas de trabajo de todos los ambientes) sino también las aplicaciones y servicios que fueron construidos alrededor de la mismas.

A continuación describe dos maneras de realizar esto y sus ventajas:

- En primera instancia, el proceso de migración da a la organización la oportunidad de gestionar problemas con su propia infraestructura de *software* y con sus áreas de negocio funcionales, procesos de negocio y flujos de trabajo que esta infraestructura soporta.
- En segunda instancia, la nube no es sólo un medio para reducir costos o resolver problemas, sino el de la transformación de las operaciones de IT – y a la vez, en el proceso, también transformar las operaciones del negocio.

El autor resalta que el proceso de migración le da al negocio y a los interesados de IT una oportunidad de hablar acerca de lo nuevo, diferente y las ventajas acerca de la nube. Por ejemplo, cuáles son los beneficios que el negocio obtendría al re-alocar sus aplicaciones de negocio en la nube. Cómo características de la nube tales como la elasticidad, la redundancia, la seguridad y la habilidad

de provisionar rápidamente (*on-demand*) nuevos recursos tecnológicos y cómo eso se traduce en beneficios para el negocio.

Al mismo tiempo, también las logísticas de la nube abren nuevas oportunidades para exponer funciones de servicios a clientes, *partners*, proveedores y otros consumidores; la co-localización de la nube sumada a servicios de *Machine Learning* (ML), Inteligencia Artificial (AI), seguridad automatizada, integración de datos, desarrollo de *software* y otros servicios hacen más fácil para los desarrolladores incorporar características avanzadas dentro de las aplicaciones y servicios que construyen.

Finalmente, concluye, la combinación de estos beneficios hacen posible para los negocios desarrollar nuevos productos y servicios, perseguir nuevas iniciativas, establecer nuevos tipos de *partnerships* y perseguir otras nuevas oportunidades de innovación tecnológica.

2. Descripción de la problemática

"The people who built Silicon Valley were engineers. They learned business, they learned a lot of different things, but they had a real belief that humans, if they worked hard with other creative, smart people, could solve most of humankind's problems. I believe that very much."

(Jobs, Steve)

Más allá de los evidentes riegos que representan la logística, la gestión de tiempos (no sólo los operativos sino también los muertos por demoras de terceros tales como proveedores), la gestión de compras, la capacidad misma del Data Center en sí mismo (falta de espacio físico, capacidad energética o de refrigeración) para albergar crecimiento, los cableados y puesta en funcionamiento de los equipos; existe una serie de componentes que, apalancados en la potencia de la nube transforman a los Data Centers en una opción a abandonar o al menos repensar a la hora de incrementar la capacidad computacional de una compañía (que en un escenario normal de Data Center puede tomar semanas o meses contra segundos o minutos en la nube), y se irán cubriendo a lo largo de este capítulo.

De acuerdo a una encuesta realizada por Spice Works (Spice Works, 2019), aunque alrededor del 98% de las compañías tienen sus propios servidores físicos *on-prem* para mantener su infraestructura de IT, la pandemia ha forzado y colaborado en algunos ajustes. Ahora, las compañías están migrando hacia la nube y rehusándose a seguir manteniendo sistemas *legacy* para apostar a la continuidad de negocio. La firma Flexera (Adler, 2022), por otro lado, encuestó a tomadores de decisiones de todo el mundo y el 50% de ellos confirmó que su utilización de servicios y gastos en la nube se incrementará.

2.1. Estado de situación actual – Relevamiento

Dado que para el desarrollo de este trabajo no ha sido posible acceder a información fehaciente del estado de situación problemático a resolver dentro de la empresa a estudiar por cuestiones de confidencialidad, el maestrando se basará en información testimonial basada en experiencias personales, de expertos y del mercado en general.

En el apartado 1.3.5 de este trabajo de tesis se formuló un diagrama de flujos típico de provisionamiento de un Data Center.

Basándose en experiencia del maestrando y de otros expertos consultados, un proceso de principio a fin que comprenda la adquisición de hardware para ser instalado y puesto en funcionamiento dentro de un Data Center toma en promedio de 60 a 90 días, esto suponiendo que haya capacidad para instalarlo en el Data Center y que el proveedor disponga de stock para poder ser entregado.

Lo arriba expuesto es sólo una parte del problema que enfrentan las empresas a la hora de ampliar su capacidad computacional o de almacenamiento en Data Centers, existen numerosas cuestiones que luego serán abordadas en este capítulo que ejemplificarán y justificarán el postulado.

Frente a las ventajas que presenta migrar a la nube, nos encontramos con problemas de disponibilidad de Data Center, costos, agilidad, *time-to-market*, seguridad, capacidad de adaptación a nuevas tecnologías y cambios de paradigmas tecnológicos, logísticos, etc.

Todo esto debe sumarse a la necesidad imperativa de personal capacitado dentro de la empresa, y multiplicarlo por numerosas oportunidades de riesgos de gestión/instalación/mantenimiento de error humano, sin siquiera (en la gran mayoría de los casos) poder disponer de herramientas de *software* que faciliten y agilicen estas gestiones.

Como por si fuera poco, no basta con hacer todo esto bien, teniendo en cuenta que para poder expandir la capacidad de un Data Center se necesita además de mayor cantidad de personal y que también tomará tiempo contratar y capacitar para que esté listo en el momento en que sea necesario.

Finalmente, se agrega una capa de complejidad muy importante que es de factibilidad técnica, donde en muchas ocasiones se mal-diseña o mal-dimensiona una solución, acarreando pérdidas de muchísimo dinero y tiempo y que se traducen también en demoras y engorrosas gestiones para mitigar y corregir cualquiera de estas situaciones comentadas.

Las trabas que presentan los modelos clásicos de ampliación y mantenimiento de Data Centers pueden transformarse en las peores pesadillas de muchos gerentes y altos directivos de logística y operaciones; presionados en ambientes de baja tolerancia al error por parte de los departamentos de ventas, tecnología y producto (entre otros), que "por culpa de" ven sus operaciones y objetivos comprometidos.

El maestrando agrega además de su propia experiencia profesional el haberse encontrado en situaciones donde se hicieron compras equivocadas de *hardware* que no era necesario y se omitió comprar el que sí se necesitaba porque el sistema donde se cargaban las solicitudes de compra de equipamiento no contaban con el modelo exacto que se deseaba comprar de dichos dispositivos; y que luego del arribo a la empresa de los mismos hubo que hacer una reversión de la entrega al proveedor asumiendo pérdida de parte del valor de la compra ya que el proveedor no estaba en condiciones de hacer la devolución de esa magnitud ni quedarse trabado con tanta cantidad de stock que no necesitaba ni para el cual disponía de espacio suficiente en sus depósitos.

La anécdota finalizó con gente despedida, incluyendo al gerente de compras de la empresa, la misma debió incurrir en más gastos para volver a hacer una compra asumiendo un costo hundido por el error y llegando tarde a la salida a producción del producto que luego implicó más inversión de parte de otras áreas para recapturar parte del *market-share* que perdió por la demora.

El arriba expuesto es sólo un ejemplo de miles en las situaciones provocadas por mala gestión de logística, capacidad, tiempos, demanda, y costos; en un aspecto crucial para las operaciones de cualquier empresa que utilice sistemas tecnológicos como parte de su columna vertebral sobre la cual

se apalanque su negocio, ya sea ésta *core* o no en su actividad y que en algunos sectores resulta binaria, o se adopta y se llega a tiempo o no, con el precio que ello conlleva.

Tal y como se puede apreciar, son cada vez más complejos los escenarios donde toda esta complejidad arriba ilustrada justifica el continuar aferrándose a la idea de mantener Data Centers como parte de los bienes de una empresa.

El maestrando quisiera además agregar que esto pasa en empresas de todos los tamaños, y que se hace más evidente aún en empresas pequeñas, donde quizás 1-2 personas hacen la planificación, logística, compras, recepción, instalación y puesta en funcionamiento de los dispositivos y que justamente por ser más pequeños los tiempos, costos y errores se pueden pagar más caros por la falta de estándares, procesos y mecanismos de control. En este caso, si una empresa de 100.000 empleados tiene problemas para instalar 50 routers adicionales quizás pueda compensar los tráficos compensando carga hacia otros circuitos de infraestructura, mientras que para una empresa de 30 empleados no tener 1-2 routers nuevos que encargó a su proveedor no le permiten operar en otro edificio o piso o sector de una planta de producción, frenando su capacidad productiva.

A continuación, se visitarán algunos de los aspectos que el maestrando considera de mayor importancia a la hora de estudiar la problemática central de los Data Centers.

2.2. Reducción de costos

Una de las razones más comunes por las cuales las organizaciones migran hacia la nube es para reducir los costos de infraestructura de IT. En la nube, se pueden ajustar fácilmente los recursos de acuerdo a los requerimientos del negocio y cortar los gastos innecesarios. En vez de estimar las necesidades de capacidad por adelantado, las organizaciones pueden ajustar dinámicamente, como

también al mismo tiempo eliminar hardware innecesario o dispositivos/bienes rígidos que se encuentren físicamente dentro de sus instalaciones.

El planteo es sencillo, una empresa no necesita comprar equipamiento de servidores altamente costoso, hacerse cargo de su mantenimiento y pagar grandes facturas de electricidad. Además, esto sirve para recortar gastos operacionales ya que los especialistas de DevOps y administradores de sistemas no pierden tiempo en mantenimiento de *backups* y *hardware*. La oferta planteada por los proveedores de servicios de nube funciona bajo precios de *pay-as-you-go* (pagar por lo que se consume a medida que se consume).

En 2020, Emirates migró de *on-prem* a la nube (Amazon, 2020). Estaban en busca de formas de reducir los costos de mantenimiento de *hardware* y responder al incremento de tráfico y al mismo tiempo, su caída causada por la pandemia y las restricciones turísticas. La compañía esperaba tener ahorros anuales por valor de 1 millón de dólares luego de retirar por completo su *hardware* obsoleto.

La plataforma de pagos Yedpay, decidió migrar a la nube (Amazon, 2020) luego de experimentar problemas en sus Data Centers. Sin necesidad de grandes inversiones en IT ni de personal para mantenimiento de infraestructura física, lograron reducir sus costos en un 40%.

2.3. Incrementar la agilidad de negocio

La agilidad del negocio es clave en la economía moderna global. Tener acceso a recursos de tecnología flexible es crucial para mantener el paso con los competidores y la dinámica de la industria rápidamente cambiante. En la nube, más del 99% de lo que se necesita está disponible *on-demand*. Las organizaciones no tienen que esperar semanas o meses por componentes e instalaciones, lo cual supone una problemática adicional de logística y cadena de abastecimiento de los proveedores. En cambio, la opción de la nube permite alquilar capacidades de valor para el negocio directamente a proveedores de servicios en la nube y llegar al mercado mucho más rápidamente.

La capacidad de implementar rápidamente cambios es crucial para cualquier negocio que requiera de servicios basados en IT, y desde una perspectiva de crecimiento de negocio, la nube trae consigo oportunidades ilimitadas para las empresas. Esto también involucra el ritmo de innovación digital. Con la flexibilidad que ofrecen los proveedores de nube, los equipos de trabajo pueden optimizar y acelerar sus flujos de trabajo para llevar más rápidamente sus ciclos de desarrollo hacia ambientes productivos.

Un ejemplo de esto es el caso de Capital One (Amazon, 2020), uno de los bancos más grandes de Estados Unidos, que logró reducir los tiempos de construcción de ambientes de desarrollo de varios meses a sólo algunos minutos luego de migrar a la nube.

2.4. Acorralando los riesgos de Seguridad

La seguridad continuará siendo un área de foco importante para las organizaciones. Al migrar a la nube, las organizaciones pueden modernizar su infraestructura de IT de acuerdo con las mejores prácticas y proteger sus aplicaciones de intentos de hacking maliciosos.

Una cosa a tener en cuenta es que muchos líderes erróneamente piensan que la nube por sí misma es lo que hace seguras a las organizaciones. En realidad, la nube empodera grupos para implementar políticas seguras, incrementar el *governance*, y estándares a cumplir que precisan para sus operaciones particulares.

Los proveedores de nube confiables actualizan regularmente sus servicios siguiendo los estándares de industria más modernos, cumpliendo además con regulaciones. Estas medidas apuntan a reducir los riesgos de ciberataques a sus clientes. De acuerdo con las predicciones hechas por Gartner (Panetta, 2019), hasta el 99% de las fallas de seguridad en la nube hasta 2025 serán por culpa de los clientes.

En ambientes *on-prem*, las organizaciones tienen la responsabilidad de salvaguardar la integridad y consistencia de sus aplicaciones y datos. Necesitan ser cuidadosos acerca de cómo persisten y almacenan sus datos sensibles, no sólo para protegerlos ante posibles filtraciones de datos sino para poder también cumplir con las regulaciones aplicables a leyes de privacidad de la información.

Como se mencionó anteriormente, en la nube, muchos de estos requerimientos o desaparecen se transforman en sutiles y substanciales.

La pregunta entonces es cómo resuelve este problema la tecnología de nube. Los proveedores encriptan toda la información por defecto, una práctica que no es consistentemente aplicada en entornos *on-prem*. Algo similar hace como mejor práctica al aislar datos sensibles de no sensibles, que una vez más su contraparte tampoco hace. Las bases de datos en la nube, integración de datos, calidad de datos, identidad y gestión de accesos, encriptación y servicios similares incorporan mecanismos de fábrica para reforzar el *governance* de datos entre y a través de regiones dispares, países, estados dentro de países, uniones políticas, etc. Y una vez que las aplicaciones y datos son movidos a un servicio en la nube, el suscriptor ya no sufre por la seguridad subyacente en *hardware* y *software* asociado a ellas.

2.5. Eliminar las preocupaciones por obsolescencia

Para muchas organizaciones, la decisión de migrar a la nube es influenciada por las líneas de tiempo de los ciclos de vida de su *software* y *hardware* (mejor conocidos como *End-of-Life*). Los líderes de IT hoy día no quieren tener que lidiar con acuerdos rígidos de licenciamientos y contratos a largo plazo.

Afortunadamente, la nube permite esquivar ese problema. La organizaciones no tienen que preocuparse por los ciclos de vida de las aplicaciones o términos de contratos, ya que la tecnología de

la nube permite consumir bajo el modelo *pay-as-you-go* para aquellas capacidades cruciales de servicios y tomar ventaja de las actualizaciones que versátilmente las plataformas de gestión de la nube pueden ejecutar por sí solas automáticamente una vez que éstas son liberadas por los fabricantes del *software* y aplicarlas a la infraestructura. De esta forma, las compañías pueden acceder a las últimas tecnologías inmediatamente sin tener que atarse a contratos inflexibles.

La confiabilidad es un factor clave, y es cierto que no todas las implementaciones en la nube salen bien fácilmente. Pueden darse problemas de hardware y caídas en el servicio. De todas maneras, las migraciones a la nube son un paso confiable hacia la reducción de caídas y disminución de riesgos de pérdida de datos a futuro.

La mayoría de los proveedores tienen SLAs (*Service Level Agreements*) que garantizan un 99% de *uptime* (tiempo de servicio disponible), y los proveedores cargan con la responsabilidad de ejecutar *backups* y de los procesos de recuperación ante caídas (*disaster recovery*), lo cual ahorra mucho tiempo a un equipo de trabajo.

Otro caso es el de la plataforma de la empresa Under Armour (Amazon, n.d.), en el cual tuvo que enfrentar un desafío de confiabilidad en sus Data Centers. La empresa disponía de 2 Data Centers, naturalmente y por una cuestión de redundancia, si había problemas en el primario, se activaba el secundario. Pero cuando había problemas en el primario, tenían caídas. Entonces la firma decidió moverse a la nube, lo cual resolvió su inconveniente.

2.6. Consolidar los Data Centers

Gracias al poder computacional de la nube, las compañías ya no tienen que gestionar sus propios Data Centers *on-prem*. De esta manera, los líderes de IT pueden tercerizar responsabilidades de gestión a proveedores terceros y reubicar recursos en actividades de mayor valor (tareas de soporte de niveles bajos, tales como las que ejecutaría un soporte de nivel 1 o 2 de operaciones –

GNOC). Adicionalmente, las organizaciones pueden consolidar operaciones y distribuir accesos a los servicios de la nube según sea necesario, y por lo tanto incrementar la eficiencia.

2.7. Fomentar la transformación digital

Muchas organizaciones están realizando transformaciones digitales para obtener valor incremental de sus bienes existentes. Gracias a los avances que otorgan las tecnologías de la nube, más y más empresas eligen digitalizar sus funciones *core* de negocio, incluyendo las migraciones de herramientas tales como CRM, SAP, data analytics, etc.

Aquellas que migran lejos de tecnologías obsoletas pueden incrementar su productividad y la de sus fuerzas de trabajo, permitiéndose innovar y desbloquear nuevas fuentes de ingresos comparadas con sus pares.

2.8. Acelerar el crecimiento

La tecnología es más importante que nunca en el rol que juega dentro del crecimiento organizacional, ya sea a través de crecimiento orgánico o fusiones y adquisiciones (*Mergers and Acquisitions*). Las empresas en la nube pueden integrar nuevas adquisiciones a plataformas ya existentes de una manera mucho más fácil. Ellas pueden incluso escalar rápidamente, apalancándose en las funcionalidades de crecimiento automático de la nube y otros servicios flexibles de gestión de datos.

2.9. Apalancamiento en las nuevas tecnologías

Migrar a la nube abre numerosas puertas de oportunidades cuando se trata de apalancarse en tecnologías modernas. Por ejemplo, las organizaciones que migran pueden tomar ventaja de *Machine Learning* y *AI*, tal y como se comentó anteriormente en este trabajo, lo cual no es posible en arquitecturas *on-prem*. Ellas incluso pueden dar un salto hacia tecnologías altamente complejas, tales como Docker, Kubernetes y *data lakes*, en minutos.

2.10. La encrucijada de la capacidad en *On-Prem*

El desafío que enfrentan las empresas que aún no han migrado a la nube es en la búsqueda de escalabilidad y dinamismo, sin poner en riesgo la adquisición de hardware y los costos de mantenimiento aparejados pero al mismo tiempo no sacrificando alta disponibilidad en función de la demanda de recursos computacionales.

Para Steve Swoyer (Steve Swoyer, 2021), el tener la carga de recursos en la nube, permite responder rápidamente a picos de demanda y baja capacidad cuando se necesita. Todo ello se efectúa dinámicamente y no requiere de mucho tiempo y esfuerzo. En un Data Center, normalmente esto requeriría comprar equipamiento adicional e instalarlo para incrementar la capacidad. El inconveniente aquí resulta en que una vez pasado el pico de carga, todavía la empresa tendría que pagar por los recursos redundantes que consume.

Dicho esto, la escalabilidad es siempre limitada en los ambientes *on-prem* por restricciones duras que son menos concernientes a las limitaciones de *hardware* físico (en casi todos los casos, el *hardware on-prem* no virtualizado tiene mejor rendimiento que recursos virtualizados en nube) que a las limitaciones prácticas respecto del espacio físico disponible en el Data Center, la densidad de los recursos de *hardware* (limitados por la cantidad máxima de servidores y arreglos de almacenamiento), y, más importante aún: el presupuesto para gastos en IT.

En contraste, los recursos de nube, son virtualmente inagotables; en la mayoría de los casos y para la mayoría de las cargas de trabajo, los suscriptores pueden provisionar recursos en las plataformas a bajo costo y así escalar fácilmente hasta la performance necesitada que muy probablemente en un esquema *on-prem* habría sido una odisea alcanzar. El punto crítico a favor de la nube, es que aun habiendo hecho esa expansión, la nube todavía tendría capacidad de sobra para afrontar los picos subsiguientes de demandas de carga aperiódicas que se presenten.

Es evidente a simple vista, la nube permite una elasticidad de recursos que no tiene análoga en ambientes *on-prem*. Los recursos virtuales escalan independientemente unos de otros. Mejor aún, los recursos puede ser encendidos o apagados o pausados para luego ser reanudados según conveniencia. Algunos proveedores incluso facturan a los suscriptores sólo por la capacidad que realmente consumieron y sólo cuando lo hicieron, tal y como se explicó en el concepto de *pay-as-you-go*.

2.11. Logística

En este aspecto, sin lugar a dudas los Data Centers presentan un problema que en la nube no existe a la hora de ampliar capacidad. En el caso del primero, se deben contemplar tiempos de compras, adquisición, transporte, coordinación de entrega de equipamiento, disponibilidad de stock de parte del proveedor, llegada a depósito, instalación y puesta en marcha. Todos estos pasos pueden tomar semanas o hasta meses, incluso sin contar tiempos muertos ni demoras por error humano.

En el caso del segundo y como se ha cubierto a lo largo de este capítulo y los anteriores, la misma tarea toma segundos y algunos clics de mouse o incluso nada si el provisionamiento está completamente automatizado para escalar horizontal o verticalmente en función de la demanda.

3. Análisis y solución propuesta

"Sometimes problems don't require a solution to solve them; instead they require maturity to outgrow them."

(Maraboli, Steve)

3.1. Objetivo

El objetivo fundamental de este trabajo de tesis es el de proveer una solución que permita automatizar de principio a fin la problemática del provisionamiento de infraestructura, apalancándose en las ventajas de las tecnologías de la nube y al mismo tiempo buscando eficiencia y eficacia operativa, reduciendo costos y mejorando las respuestas a la volatilidad de las necesidades del negocio, siendo el eje de la propuesta resolver las problemáticas enunciadas a lo largo del capítulo 2 de este trabajo, fundamentalmente sobre el supuesto de la carencia de muchas empresas de procesos de compras y gestión bien definidos y al mismo tiempo de herramientas que permitan minimizar la posibilidad de errores voluntarios o involuntarios del personal involucrado en la gestión de un Data Center y las consecuencias de ello, siendo probos ejemplos ya presentados que justifican la necesidad de una solución basada en tecnologías de la nube.

De esta manera, todo lo expuesto en el capítulo anterior es fácilmente trasladable al proveedor de tecnología en la nube, permitiendo a la empresa poner foco en lo que es verdaderamente importante para su negocio y no en las problemáticas típicas de un Data Center.

La propuesta a continuación es el resultado de amplio debate con expertos en infraestructura y tecnologías de nube sumado a la experiencia profesional del maestrando y todo el material revisado a lo largo de este trabajo de investigación.

3.2. Propuesta de Modernización

En términos simples, la propuesta de modernización consiste en armar un Marketplace de provisionamiento de servicios en la nube, capaz de ser consumido por cualquier área relacionada con IT y que busque implementar sus productos y/o servicio en dicha plataforma; todo esto contemplando la remoción completa del trabajo manual que requiera la creación, modificación, eliminación e implementaciones en todos los ambientes que comprendan a la arquitectura necesaria para cumplir con los propósitos de dichos productos y/o servicios.

Cabe destacar que en las conversaciones con expertos, algunos comentan que este tipo de soluciones están empezando a desarrollarse en algunas empresas, en respuesta a las problemáticas ya cubiertas a lo largo del marco teórico en esta tesis.

La propuesta en cuestión pretende acelerar el uso de productos modernos de ingeniería, plataformas y mejores prácticas en forma sostenida a través de la tecnología, que conforma la piedra angular del programa; y se plantea por parte del maestrando como solución al estado actual de los problemas que padecen los Data Centers tradicionales/*legacy*.

Las metas se describen a continuación:

- Acelerar el uso de plataformas fundacionales, productos y prácticas de forma tal que pueda capitalizarse el camino ya allanado por evidencia dura a través del uso de la comunidad de ingeniería de *software*.
- Guiar desde la adopción consistente y sostenible y con esto crear una comunidad de expertos rica en feedback con el fin de mejorar la entrega de producto y el crecimiento basado en tecnología.
- Crear visiones que favorezcan la cohesión de principio a fin y el uso de productos estandarizados para toda la empresa que a su vez permitan encontrar oportunidades de aceleramiento de adopción e implementación para la mejora continua de la experiencia de ingeniería y usuarios.

- Reducir, siempre que sea posible, el trabajo manual en aquellas tareas de operación, mantenimiento o gestión que sea repetitiva; priorizando las que otorgan mayor valor al negocio con poco nivel de esfuerzo o tienen mayor impacto en costos.
- La propuesta además resalta la importancia del alineamiento estratégico de todo el negocio, incluyendo la capacitación de los equipos de trabajo en metodologías ágiles, tecnologías de desarrollo e ingeniería de software y de la nube (DevOps). Es importante destacar que se pretende construir con un enfoque ágil y cros-funcional fácilmente escalable y que para ello, esta estrategia debe tener en cuenta los siguientes factores como pilares núcleo:

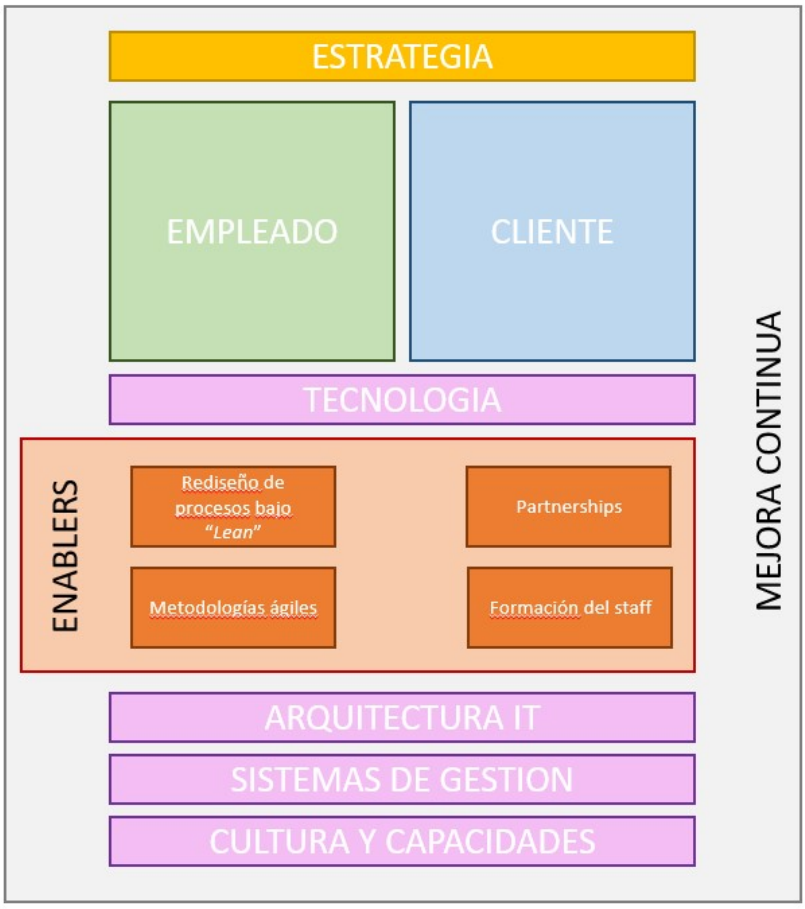


Figura 22: Pilares estratégicos para ejecutar la solución propuesta. Imagen propiedad del maestrando.

La solución que se propone en este trabajo de tesis es la de servir a los usuarios de IT de un portal donde podrán gestionar el provisionamiento de la infraestructura *on-demand* tal y como harían hoy desde sus celulares en el portal de aplicaciones (como el Apple Store o el Google Play Store). El funcionamiento es simple, el usuario selecciona qué quiere instalar y el sistema operativo se encarga de la instalación en forma transparente y totalmente desatendida.

El diagrama en bloques de la arquitectura a implementar es el siguiente:

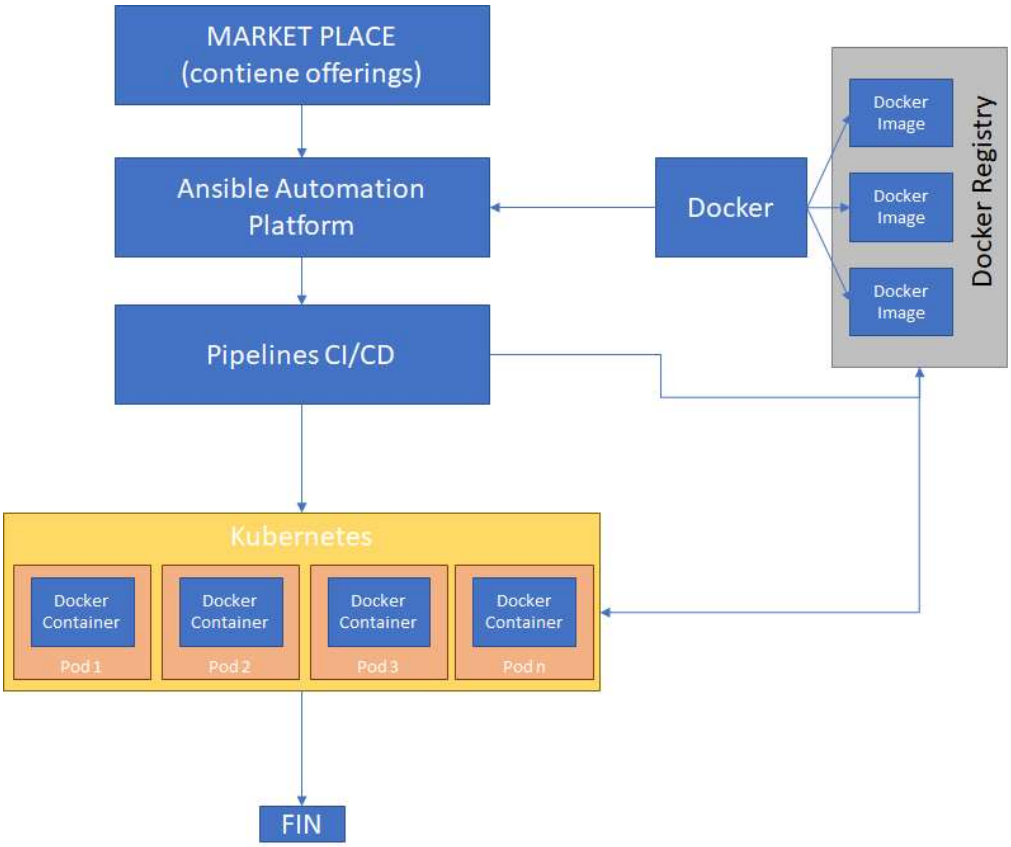


Figura 23: Diagrama en bloques de arquitectura de la solución propuesta. Imagen propiedad del maestrando.

El funcionamiento de cada bloque es el siguiente:

- ***Market Place:*** es un portal que da acceso a usuarios de IT y gestionan o infraestructura o proyectos de desarrollo o cualquier otro producto relacionado a los anteriores. Dentro del mismo se encuentran *offerings* (la oferta de servicios), cada uno de los cuales conteniendo parámetros fundamentales tales como nombre y locación, para que luego puedan pasar al siguiente bloque a través de una llamada a su API.

- **Ansible Automation Platform:** Cumple función de gestionador de configuración y orquestador de Ansible que utiliza YAML y workflows según las solicitudes que recibió por parte del *Market Place.* Lo que hace es cargar la *metadata* del requerimiento (de su propia orquestación y credenciales), luego crea la suscripción usando *job templates* (plantillas) y con ellas construye los building blocks. Las suscripciones son compuestas por los *resource groups*, VNETs, subredes, cuentas de Storage y una *key vault* para almacenar todas las credenciales necesarias para operar el contenido de la suscripción. Por último, es quien se encarga de notificar al usuario acerca del *pricing* (cómo se va a cobrar, cuánto se va a cobrar) y de gestionar el *scale up* y *scale out* (vertical y horizontal).

- **Docker:** Responsable por almacenar y gestionar los microservicios contenidos dentro de las *docker images*.

- **Pipelines CI/CD:** Son construidos por los equipos de DevOps (construyen los repositorios de código para llevar control de versionado, la infraestructura de red, los ambientes con sus respectivas bases de datos y servicios de hosting web y las variables de entorno. Éstos luego son ejecutados para así construir el código, ejecutar los *tests* de integración y cobertura y finalmente transformarse en *docker images*.

- **Kubernetes:** Gestiona los requerimientos dentro de un *namespace* específico donde hará las implementaciones. Las *docker images* se despliegan mediante implementaciones a Kubernetes que son disparadas por los *pipelines*. Para realizar esto, Kubernetes se conecta con el *Docker Registry* (repositorio de contenedores) a buscar los microservicios que debe instalar en forma de contenedores dentro de los pods que creó y para ello utiliza los archivos *manifest* que le fueron provistos anteriormente.

El proceso se cierra con la notificación al usuario de que su solicitud de *offering* fue implementada y está lista para ser consumida.
El maestrando sugiere además que se asocie los *offerings* instalados a cuentas o centros de costo para que luego los costos puedan ser imputados correctamente al área que incurrió en el gasto y que los mismos manejen criterios de aprobación de parte de directivos o sponsors senior de los proyectos de forma tal de poder responder ante auditorías y seguimiento de costos de los proyectos.

Finalmente, dada la extensión del plan de transformación que atravesará la organización, se sugiere por su relevancia además revisar el modelo propuesto por Mc Kinsey titulado “Next-Gen Operating Model”.

4. Plan de implementación

"Organizations are successful because of good implementation, not good business plans."

(Kawasaki, Guy)

4.1. Technology Stack

Resulta muy complejo delimitar las tecnologías a utilizar para poner en marcha la solución, así como en muchos otros aspectos no existe una sola forma de realizar algo, de la misma manera algunas de las tecnologías disponibles en el mercado se solapan entre sí debido a que desde diferentes enfoques han podido resolver las problemáticas que son comunes a la comunidad de IT, de la misma manera el maestrando seleccionará algunas tecnologías que considera simplifican la búsqueda por resolver la problemática planteada en este trabajo de tesis.

Por lo arriba expuesto, sometida a riguroso estudio, la propuesta puede que resulte de agrado a algunos expertos y a otros no tanto, y siendo totalmente subjetivo, se deja expreso que la propuesta no pretende ser ni la única ni la mejor manera de resolver la problemática planteada.

A continuación se presenta un mapa *full-stack* típico de tecnologías de DevOps, distribuidas por función:

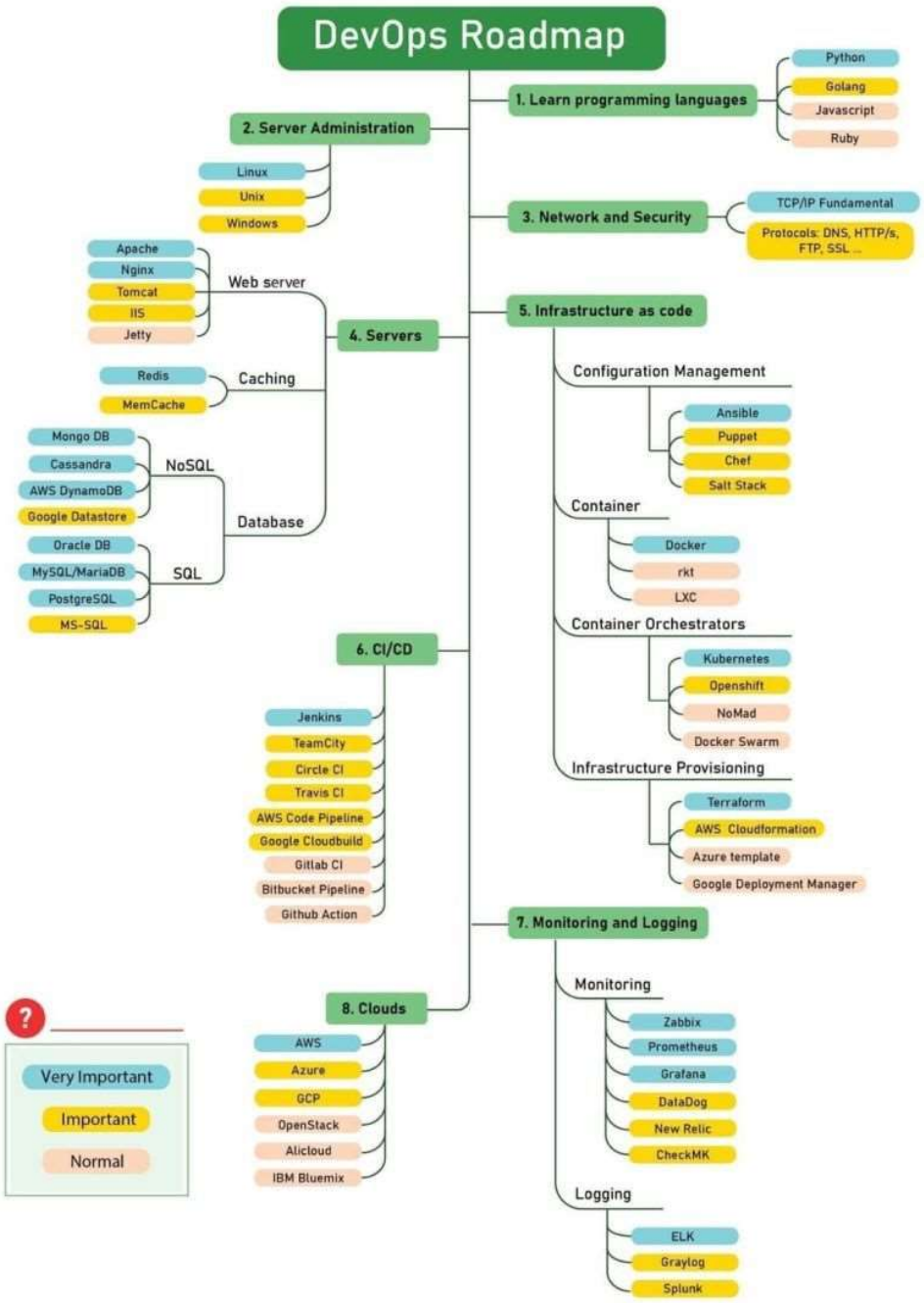


Figura 24: Mapa de tecnologías de DevOps (StarAgile).

Aunque se encuentra por fuera del alcance de este trabajo de tesis el cubrir todas las tecnologías descritas en la imagen de la página anterior, el maestrando considera de mucha

importancia para entender la amplitud de las tecnologías de la nube el poder distinguirlas en un mapa como parte del contexto en torno al problema planteado.

Es importante asimismo destacar, que este trabajo de tesis se centrará mayormente en el desarrollo de una solución basada en las tecnologías de fondo **celeste** que se encuentran en el punto 5 del gráfico (*Infrastructure as Code*) y que fueron cubiertas a lo largo del marco teórico con la suficiente profundidad como para poder plantear la propuesta pero que no obstante no serán suficientes para implementarla ya que necesitará también de tecnologías citadas en los puntos 1, 4, 6, 7 y 8 del mismo gráfico. Todas ellas, por razones de alcance y vasta cantidad de tecnologías y conceptos tecnológicos involucrados, serán asumidas como conocidas, como así también la existencia de los elementos que se encuentran en los puntos 2 y 3.

4.2. Arquitectura de Ambientes

El maestrando considera importante construirse una arquitectura que contemple los siguientes ambientes de trabajo:

- Instancias locales en las estaciones de trabajo de cada miembro del equipo.
- Un ambiente de desarrollo donde se irán volcando e integrando los aportes de todos.
- Un ambiente de pruebas de integración para que los usuarios finales puedan efectuar las pruebas que consideren pertinentes.
- Un ambiente de *Staging*, o también conocido como pre-producción, cuya infraestructura es un espejo de la productiva a fin de realizar una prueba de estabilidad final.
- Un ambiente productivo.

Asimismo, se sugiere se maneje control de versiones en todos los niveles y que los pasajes entre ambientes sean realizados a través de los *pipelines* de CI/CD que se describieron anteriormente.

4.3. Gestión del Proyecto

Para la ejecución del proyecto el maestrando propone la adopción de metodologías ágiles y la adopción de un plan que permita la transformación del *mindset* de los miembros del equipo de trabajo y principales interesados. La cultura de trabajo será fundamental para el éxito del mismo, por ello se propone el siguiente plan como "*Modernization Career Path*", que podría resultar no sólo para alinear a los involucrados sino también como el siguiente paso hacia una cultura más eficiente de trabajo dentro de proyectos de desarrollo de software. Esta currícula persigue sentar las bases de la transformación de los involucrados invirtiendo en capacitar a los ingenieros de infraestructura y desarrolladores existentes y comenzar a prepararlos para los nuevos desafíos que residen en el cambio de paradigma:

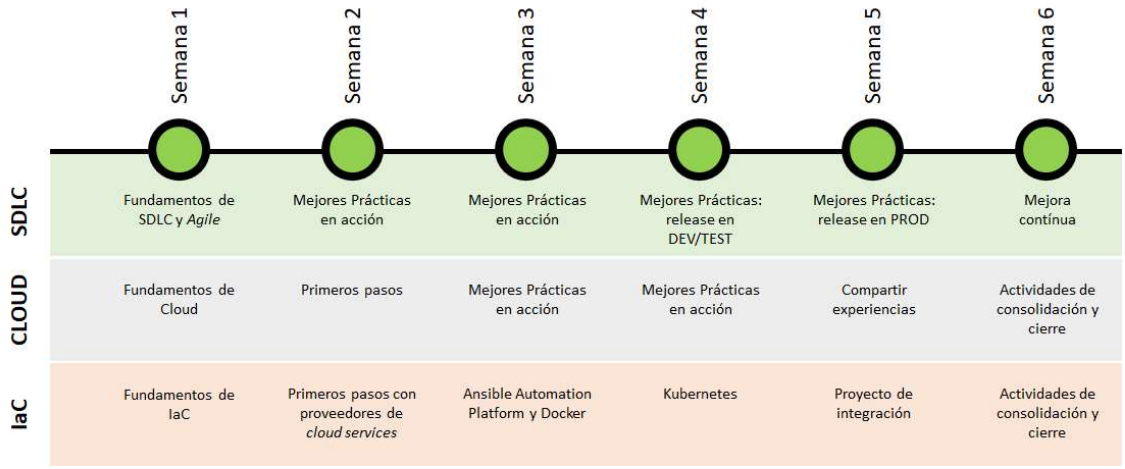


Figura 25: Plan de carrera del "Modernization Career Path". Imagen propiedad del maestrando.

Como se ve en el cuadro, los focos son:

- Desarrollar conocimientos en el área de ciclo de vida de software, las metodologías ágiles, comprender la arquitectura y separación de ambientes, qué es un pipeline de implementaciones.
- Introducir a los participantes en tecnologías de la nube desde la práctica en laboratorios.

- Incorporar conocimientos de las tecnologías involucradas en *Infrastructure as Code* como eje fundamental de la solución propuesta.

Se propone que todos los interesados participen de este plan de introducción a las tecnologías de la nube, de forma tal que todos los involucrados en el proyecto hablen el mismo idioma, entiendan las limitaciones y problemáticas asociadas a la naturaleza de los problemas y puedan trabajar dentro de la misma cultura.

4.4. Roadmap y Plan de implementación (CI/CD)

En este punto el maestrando considera clave priorizar las necesidades del negocio. Al mismo tiempo, un MVP (*Minimum Viable Product*) debería contemplar el *market place* funcionando y luego determinar con las áreas de interesados cuáles son los *offerings* que se irán agregando a la herramienta en función de la importancia, la demanda, el costo y los tiempos de construcción de cada uno de ellos.

Cabe destacar que cada uno de los *offerings* será implementado a lo largo de todo el ciclo de vida del software y llegará al ambiente productivo a través de *pipelines* de CI/CD.

El gráfico a continuación (sólo con propósitos ilustrativos) muestra un ejemplo de implementación de *pipelines* de CI/CD utilizando Jenkins:

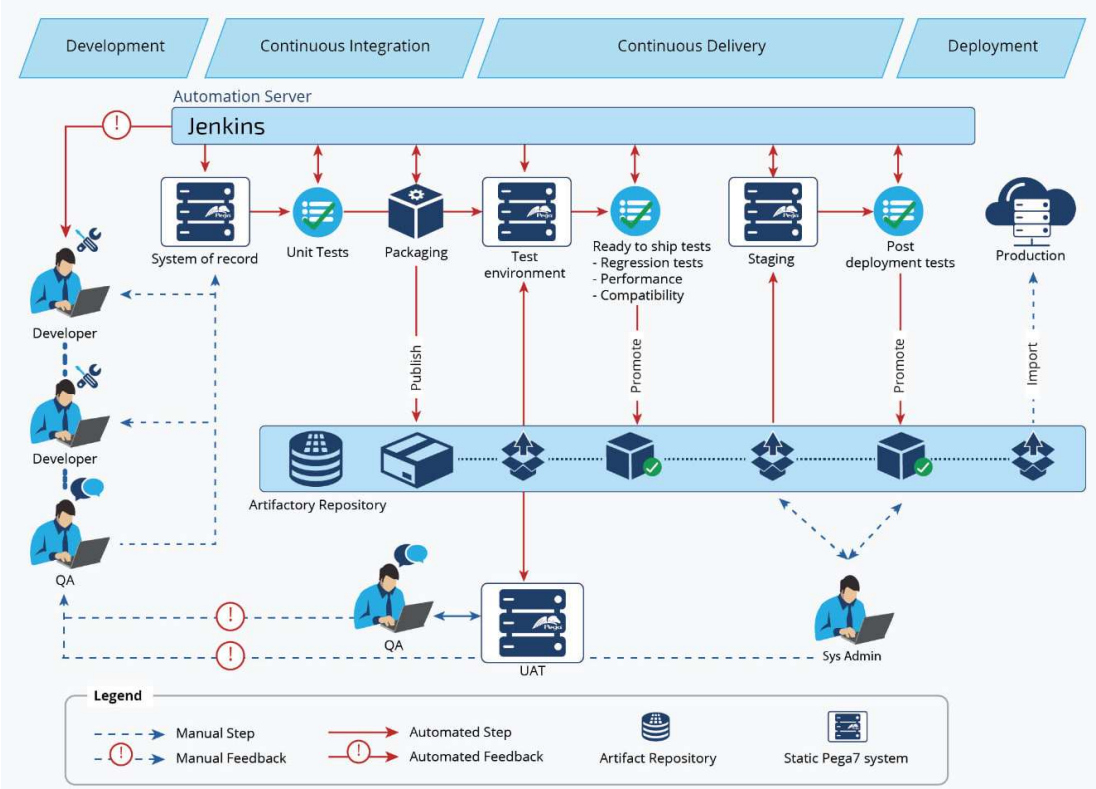


Figura 26: Ejemplo de implementación de pipelines con Jenkins (amith136).

En el ejemplo a continuación se muestran los pasajes entre ambientes utilizando la herramienta GitHub y la implementación en esquemas de preproducción en nube y para producción un esquema de nube *multi-cloud* (Amazon y Google):

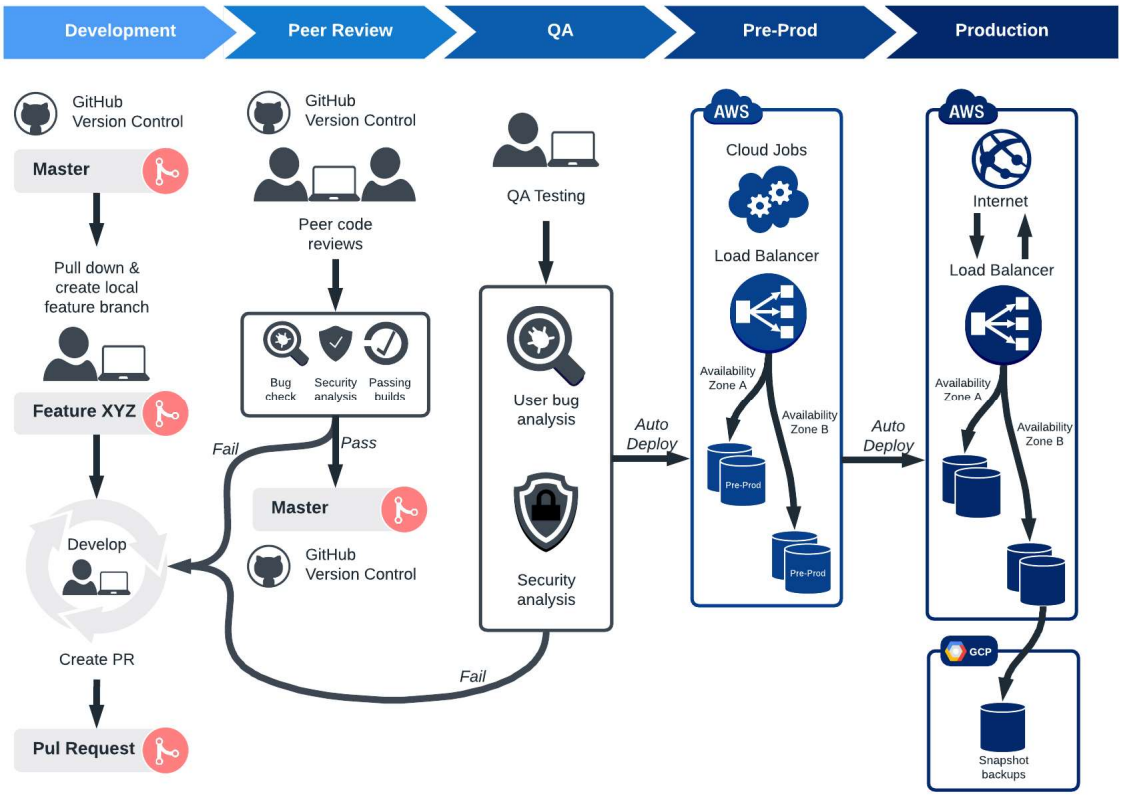


Figura 27: Ejemplo de implementación de migración de ambientes con GitHub (Ramchandani).

4.5. Soporte post-implementación y mantenimiento

Con la implementación de la propuesta, el actual modelo de soporte y escalamiento de IT deberá también evolucionar. Esto es, empezando porque los niveles más bajos de soporte serán trasladados al proveedor de servicios de la nube (quedan completamente tercerizados); con lo cual en este caso la propuesta del maestrando es que se invierta en seleccionar de dichos niveles a los perfiles de mayor potencial y rendimiento para transformarlos en ingenieros de DevOps que luego conformarán el nivel 3 de escalamiento y que el nivel 4 se asigne a ingeniería específica de producto, todos bajo el modelo de gestión de problemas, incidentes y cambios de ITIL.

5. Análisis de impacto económico/financiero

"Management is efficiency in climbing the ladder of success; leadership determines whether the ladder is leaning against the right wall." (Covey, Steven)

5.1. Análisis de demanda

Entendiendo que estamos hablando de una propuesta sin información de negocio, la estimación de la demanda se hará en función de supuestos y basados únicamente en la opinión y experiencia del maestrando.

La propuesta detallada en el capítulo anterior sugiere una arquitectura de 4 ambientes, supongamos que para los ambientes de desarrollo y *testing* tenemos 1 máquina virtual para los servidores web y luego las bases de datos comparten la instancia del motor con otras aplicaciones, a efectos prácticos se asumirá que cada base de datos consume un quinto de la capacidad del servidor. Para los ambientes de *Staging* y Producción, se asume que tienen redundancia en *clusters* con *failover* y balanceo de carga y poder garantizar el *uptime* de la aplicación.

En el análisis de costos a continuación se procederá a detallar la comparación de costos de una solución en Data Centers legacy versus la misma solución implementada 100% en la nube.

5.2. Análisis de costos

Si bien existe la posibilidad de construir Data Centers con nubes privadas, cabe destacar que el trabajo es el mismo con algunas diferencias mínimas; la clave pasa porque la inversión inicial cuando se comienza de cero o se desea ampliar capacidad actual representa una inversión sumamente elevada (en el orden de millones de dólares). En el caso de no desear esto, las empresas hacen un *re-purpose* de su infraestructura existente y la migran hacia un esquema de nube privada, el problema en ese caso sigue siendo la renovación por obsolescencia pasado un período de aproximadamente 5

años para renovar todo el *hardware*, se mantienen problemas de soporte, *backups*, actualización de *software*, licencias, etc.

Esto, si todavía existe espacio físico en el edificio del Data Center para ampliar la capacidad computacional. Caso contrario se debe contemplar la posibilidad de tener que adquirir o rentar un nuevo espacio o construirlo, con todos los gastos que ello implica de *real state*, logística, vínculos entre Data Centers, costos de mantenimiento, staff *on-site* para el nuevo lugar que haga el mantenimiento, personal de seguridad, etc. Los costos son fácilmente multiplicables, y no estamos ponderando los costos asociados al *time-to-market* o costo de oportunidad.

Por otro lado, dependiendo de cómo se construya una solución, puede que esta termine siendo más cara instalada en la nube que *on-prem*.

Ejemplo de esto es una aplicación basada en servicio IaaS, cuando se la migre a la nube haciendo lo que se conoce como *lift&shift*, pueden existir costos adicionales de licenciamiento y redundancia de servidores.

De las entrevistas con expertos, surgió un dato no menor: Comprar un servidor económico para *middleware* y ponerlo en marcha listo para brindar servicios cuesta no menos de 50.000 USD y puede tomar hasta 6 meses desde que se inicia el proceso de compra. Esto en comparación a los 100 USD (más soporte) que puede llegar a salir la misma configuración de potencia computacional en la nube por mes.

Si hacemos el cálculo, a una amortización a 5 años, son 10.000 USD al año en el caso de un servidor *on-prem* contra 1.200 USD en la nube (más soporte). En este caso la diferencia es abismal.

Se pasará ahora a un caso más práctico, un caso de una adquisición de *hardware* suficiente como para tener un número real de ambos mundos.

Se contó para este análisis con un presupuesto para armar 3 Racks basados en servidores Dell, que consta de 4 chasis y 44 servidores *blade* de la línea PowerEdge y tiene una capacidad total de

procesamiento de 126 núcleos Intel Xeon y 36 TB de RAM y una capacidad de almacenamiento de 200 TB.

5.2.1. Costos de Hardware

Concepto	Cantidad	Costo Unitario (en u$s)	Costo Total (en u$s)
PowerEdge M1000e Chassis	4	USD 33,739	USD 134,954
Servidores Blade PowerEdge M630	40	USD 11,062	USD 442,484
Servidores Blade PowerEdge M830	4	USD 34,385	USD 137,541
PDUs	12	USD 373	USD 4,474
Memoria 32GB RDIMM 2133 MT/s	48	USD 300	USD 14,386
Balanceo de Carga - Citrix Netscalers MPX8005	3	USD 34,762	USD 104,285
Servicios de implementación Netscalers	1	USD 7,720	USD 7,720
Switches de agregación para LAN	4	USD 9,595	USD 38,380
1 VMAX 200K para el sitio principal	1	USD 622,841	USD 622,841
Integración a la Nube (1000-2000 VM's)	1	USD 43,402	USD 43,402
Switches de agregación de SAN FC 16 Gb	2	USD 24,875	USD 49,750
Switches Piso (Sitio)	14	USD 4,000	USD 56,000

Subtotal 1: USD 1.656.216

Tabla 2: Costos de *hardware* para solución *on-prem*.

5.2.2. Costos de Servicios DC

Concepto		Cantidad	Costo Unitario (en u$s)	Costo Total (en u$s)
Racks	Con soporte a 5 años	3	USD 2,500	USD 7,500
Conexión Fibra	Plan 10 x Fibras SAN DC = Total USD 3000	15	USD 300	USD 4,500

Subtotal 2: USD 12.000

Tabla 3: Costos de servicios de DC para solución *on-prem*.

5.2.3. Costos de licenciamiento

Concepto	Cantidad	Costo Unitario (en u$s)	Costo Total (en u$s)
MSSQL (72 Cores x lic Enterprise)	1	USD 611,424	USD 611,424
HiperV (40 lic)	40	USD 5,000	USD 200,000
Windows	600	USD 140	USD 84,000

Windows Terminal Server	1000	USD 200	USD 200,000

Subtotal 3: USD 1.095.424

Tabla 4: Costos de licenciamientos para solución *on-prem*.

5.2.4. Costos de staff (por 5 años)

Concepto	Cantidad	Costo Unitario (en ARS)	Costo Total (en ARS)
1 x Gerente de Operaciones	60	$ 750,000.00	$ 45,000,000.00
1 x SR Infra	60	$ 500,000.00	$ 30,000,000.00
1 x SR Infra	60	$ 500,000.00	$ 30,000,000.00
1 x JR Infra	60	$ 250,000.00	$ 15,000,000.00
1 x SR Redes	60	$ 450,000.00	$ 27,000,000.00
1 x JR Instalaciones	60	$ 180,000.00	$ 10,800,000.00
1 x SSR Operaciones	60	$ 450,000.00	$ 27,000,000.00
1 x SSR Operaciones	60	$ 450,000.00	$ 27,000,000.00
1 x JR Operaciones	60	$ 250,000.00	$ 15,000,000.00
1 x JR Operaciones	60	$ 250,000.00	$ 15,000,000.00
Aportes patronales (24%)			$ 58,032,000.00

Subtotal 4: USD 2.418.000 (124 ARS por dólar, a valor USD oficial Mayo 2022)

Tabla 5: Costos de staff para solución *on-prem*.

Nota: No se tienen en cuenta ni variaciones salariales por inflación ni los costos del *site*, *facilities*, etc.

Costo Total *on-prem* por 5 años: USD 5.181.640 (sin contar costos de metro cúbico de Data Center, asumiendo que existen ya suficientes para adicionar estos 3 racks y que hay asimismo suficiente potencia eléctrica y BTUs disponibles).

Para la segunda parte, se intentará comparar la misma potencia computacional pero ahora utilizando la calculadora *online* provista por Amazon en su sitio de AWS: https://calculator.aws/ (Amazon, n.d.)

Esta herramienta permite personalizar mejor la solución apalancándose en las ventajas de la nube tales como la carga de trabajo de la infraestructura virtual.

La configuración que se utiliza en la calculadora es la siguiente:

- Carga de trabajo variable a lo largo del día con algunos picos de consumo, de lunes a viernes, y dichos picos pueden tener hasta 8 horas de duración a lo largo del día.
- Se solicitarán inicialmente 100 máquinas virtuales con un pico de 1000 cuando alcanza el máximo de carga (aprovechando el escalamiento horizontal que permite la plataforma). Cabe destacar que sería lo mismo que cargar 25 máquinas virtuales por cada *blade* en el esquema *on-prem*.
- Cada máquina virtual tendrá 4 núcleos asignados y 16 GB de RAM (que en un pico de carga corresponden a 4000 núcleos de procesador virtual en la nube contra los 2048 de la solución *on-prem*, mientras que se baja la cantidad de memoria RAM a la mitad apostando al escalamiento horizontal y balanceo de carga).
- Se le asignan 200 GB de disco a cada máquina virtual (misma capacidad de almacenamiento que en la opción *on-prem*) y se solicitan 2 *snapshots* de *backup* al día con un límite de 3 GB de cambios como máximo en cada uno. Esto no estaba contemplado en la estimación *on-prem* pero la nube lo brinda como servicio adicional sin costo en este caso.

Para esta configuración, la herramienta cotiza un valor final de: **USD 35.040** mensuales, y un **total a 5 años de: USD 2.102.400**

Como se puede apreciar, la diferencia de costos es substancial, en el ejemplo ilustrado el ahorro por implementar la solución completamente en la nube representa un **40.57%**. No obstante, ajustando aún más en función del análisis de carga real podría reducirse en al menos otro **30%** adicional.

5.3. Conclusión

Como conclusión, se puede agregar que cuantitativamente, es importante aclarar que no existen fórmulas mágicas ni cálculos que puedan realizarse a la ligera ya que todo depende de la estrategia que se utilice a la hora de analizar los modelos de arquitectura y sus correspondientes opciones según el proveedor de nube que se adopte.

En el aspecto cualitativo, el impacto en el negocio está dado fundamentalmente por las ventajas que brinda el *time-to-market*, la versatilidad para adaptarse a la demanda y la capacidad para cerrar el negocio *overnight* si así se deseara con cero impacto, aunque sin lugar a dudas la ventaja más grande está en que al delegar todas las responsabilidades de la gestión de la infraestructura y su mantenimiento a un proveedor, el foco puede estar 100% en el negocio y en el cliente. Mientras que por el aspecto económico, el impacto está en la gestión del Capex y Opex, el costo de oportunidad y los incurridos por las gestiones de personal.

Finalmente, se observa que aun sobrecargando el modelo de nube con más potencia computacional y permitiendo su escalabilidad ante picos de demanda, es mucho más rentable y permite pagar por lo que realmente se consume; con la ventaja de poder achicar la infraestructura solicitada sin que eso signifique tener capacidad computacional ociosa como podría ocurrir en un Data Center *on-prem legacy*.

6. Resultados esperados

"There are no secrets to success, it is the result of preparation, hard work, and learning from failure."

(Powell, Colin)

Como se ha podido visualizar a lo largo de este trabajo de tesis, las ventajas de la nube por sobre arquitecturas de infraestructura *on-prem* (especialmente en Data Centers de tipo *legacy*) son múltiples, y en algunos casos determinantes y contundentes.

Su flexibilidad permite que las empresas pongan foco en el negocio y no tanto en la tecnología, que a final de cuentas es un *enabler* más en la ecuación. Dependiendo del tamaño de las empresas, en las grandes puede implicar un ahorro significativo en su cuadro de resultados al final del año, pero en el caso de una *start up* es crucial, donde cada centavo cuenta puede olvidarse de desembolsar cientos de miles de dólares en hardware y contratación de empleados o consultores para poner en funcionamiento su infraestructura y a través de la nube puede comenzar a comercializar sus productos y servicios en cuestión de minutos por muy poco dinero que pagan como un abono mensual al proveedor.

La transformación de empresas propietarias de Data Centers hacia tecnologías de la nube ya ha comenzado y paulatinamente su adopción se irá incrementando. Su capacidad de crecimiento es teóricamente infinita y no supone limitantes, día a día más servicios se van agregando a su portfolio de *offerings* y sea cual fuere el proveedor, todas las plataformas integran entre sí, permitiendo la sinergia.

En esta instancia es fundamental poder definir indicadores para poder observar y evaluar los resultados de las migraciones a la nube, los KPIs (*Key Primary Indicators*) son una parte esencial del proceso. Tener los KPIs correctos ayuda a todas las partes involucradas a monitorear los procesos de migración y ver cuáles áreas están rindiendo según lo esperado y cuáles otras tienen dificultades.

Asimismo, establecer indicadores es la mejor manera de medir el progreso de la adopción de la nube. Los mismos "operacionalizan" diferentes conceptos que, de otra manera, son difíciles de medir, como por ejemplo el monitoreo de los niveles de utilización de servidores. Su aplicación permite incluso monitorear el progreso y resolver problemas antes de que se vuelvan demasiado severos como para determinar el éxito o fracaso de la plataforma de nube.

Existen distintos tipos de KPIs que pueden adoptarse, y a continuación se listarán sólo algunos como parte de la propuesta, dentro de las categorías de seguridad, experiencia de usuario, rendimiento y respuesta:

- **KPIs de Seguridad:**
    - Network I/O
    - Auditorías de usuario
    - Exposición de datos
    - Accesibilidad S3
- **KPIs de rendimiento y respuesta:**
    - Utilización de CPU
    - Utilización de Memoria
    - Tiempos de respuesta
    - *Uptime*
- **KPIs de experiencia de usuario:**
    - Latencia
    - Encuesta de satisfacción de cliente (CSAT)
    - Tasas de error
    - Tipos de error

Conclusiones

"People do not like to think. If one thinks, one must reach conclusions.

Conclusions are not always pleasant."

(Keller, Helen)

La migración a la nube es inevitable. Esto no implica que toda la infraestructura de IT *on-prem* vaya a migrarse a la nube pública, simplemente quiere decir que la mayoría de los recursos *on-prem* se moverán a un contexto de nube de algún tipo, ya sea una nube privada *on-prem*, un servicio de infraestructura en una nube pública, servicios públicos PaaS o SaaS o alguna mezcla de ambos, como ser un *blend* de nube pública-privada *on-prem*.

Las aplicaciones de escala Enterprise no son diferentes. Las funciones provistas por aplicaciones o sistemas tales como ERPs, SCMs, CRMs, SAP y otros también serán migrados oportunamente, ya sea total o parcialmente hacia la nube.

Bajo circunstancias normales, resulta asombrosamente difícil convencer a grandes organizaciones de mover sus *stacks* de aplicaciones *core* de negocio hacia otro *stack* similar pero provisto por un proveedor externo. La cuestión es que los sistemas en sí mismos son tan complejos y tan profundamente entrelazados con las operaciones de negocio que ello los constituye práctica y contextualmente en objetos inamovibles.

Pero en el caso de la nube, es una excepción; y es una vez por generación en que se tienen a veces este tipo de oportunidades de transformación, tanto para clientes como para proveedores.

El maestrando considera que la clave es el equilibrio basado en las prioridades estratégicas que defina el negocio con su *partner* de IT en la nube. Definir dónde está el *trade-off*, que la empresa decida en cuáles ventajas estratégicas conviene apalancarse y en cuáles otras no se permite ceder *governance*.

Desde un punto de vista de costos, tal y como se evidenció en el capítulo 5 de este trabajo de tesis, la reducción podría estimarse entre un 40 a un 70% en la gestión tecnológica de la empresa, dependiendo del alcance de su implementación y la configuración que se utilice para el diseño de la solución en la nube; lo cual, sumado a las mejoras en las métricas propuestas en el capítulo 6 permitirán comprobar que la transformación hacia la nube es no sólo viable sino estratégicamente acertado.

El lector sabrá apreciar que dentro de las métricas no se agregó indicadores de comparativa de time-to-market, esto se debe a que este indicador en la solución propuesta basada en la nube es obsoleto, debido a que el provisionamiento de infraestructura en la nube se automatizaría por completo y donde en un Data Center tradicional el tiempo de gestión, traslados, tiempos muertos, recepción, instalación y puesta en funcionamiento que toma el proceso completo hasta que está listo para operar se mide en meses mientras que en la nube se mide en segundos.

Otro indicador que mostrará substanciales mejoras será el que mida los costos al cierre del año en los cuadros de estado de resultado y los balances.

A continuación se procederá a cerrar este trabajo de tesis poniendo foco en algunos aspectos en los que el maestrando coincide en su opinión y experiencia son relevantes, sumando aportes del último capítulo del libro "*Migrating Applications to the Cloud*" de Steve Swoyer.

El modelo de la nube: oportunidad y responsabilidad compartida

Tal y como se ha visto en el capítulo anterior, las ventajas de costos de la nube son imposibles de ignorar, como también lo son sus ventajas con respecto a la agilidad y flexibilidad de IT, por no mencionar también su rol como catalizador de transformaciones de negocios. La lógica irrefutable de la nube, con su énfasis en su capacidad de abstracción, desacoplamiento y *pooling* de servicios, han

ayudado a purgar el mundo de IT de los prejuicios (y los del negocio) que lo reprimía y le permitió volver a recuperar su libertad de acción hacia esta transformación y renovación de paradigma.

Cabe destacar algo: y es que sin lugar a dudas los problemas ya existentes tales como la deuda técnica no desaparecerán por migrar a la nube; en la nube, y en las infraestructuras *on-prem*, una organización que diseña y mantiene sus aplicaciones o servicios de los cuales es propietaria incurrirá en deuda técnica.

De todas maneras, adaptado del autor: el modelo tiene una responsabilidad compartida. El proveedor de servicios de nube, no el suscriptor, es responsable de la seguridad, la gestión, el mantenimiento de la infraestructura de hardware y software; los suscriptores tienen sus propias responsabilidades con respecto a la definición (y refuerzo) de mejores prácticas en lo que respecta a la seguridad de la información, el *governance* de datos y otros tantos esenciales.

De la misma manera, Steve Swoyer sostiene que incumbe a los suscriptores (o por qué no arquitectos y desarrolladores) que refactoricen y, de ser necesario, reescriban las aplicaciones y servicios para que se comporten de acuerdo a los cambios que introducen los proveedores de la nube. Este es el caso por ejemplo de una librería API que es modificada o dada de baja por estar deprecada y es reemplazada por una nueva. Si el equipo de desarrollo que es dueño del producto o aplicación no tiene esto en cuenta, la misma fallará luego de efectuado el reemplazo.

Sin lugar a dudas, existen riesgos. Supóngase una empresa que consumía un *offering* en particular del proveedor de nube y súbitamente el mismo decide eliminarla del portfolio o por qué no también dejar de brindarla en forma gratuita y empezar a cobrarla. Si bien es difícil que ocurra en el sector de grandes empresas puesto que para ellas existen modelos de servicio con contratos, términos de servicio, SLAs (Service Level Agreements) que en definitiva protegen a ambas partes y sientan los lineamientos del *partnership*, otro riesgo también podría ser que el proveedor fuera comprado por un competidor o que dejara de brindar servicios (concluye el autor).

Aun así, es importante remarcar que todos estos riesgos son conmensurables frente a riesgos similares que podrían ocurrir en ambientes *on-prem*. Súbitamente un proveedor de una herramienta de ERP o CRM es adquirido por un tercero o competencia, y modifica las condiciones del servicio o decide retirar el producto del mercado o características de los mismos que son clave para que el negocio siga funcionando. Todos estos, son riesgos conocidos. En ambos contextos, las organizaciones emplean estrategias para medir y mitigar riesgos ante externalidades negativas y/o eventos inesperados que pudieran surgir durante el curso del negocio o ciclo de vida de las aplicaciones y servicios.

Consideraciones importantes adicionales de una migración a la nube

Las organizaciones deben considerar la gran cantidad de factores importantes antes de realmente considerar migrar aplicaciones a la nube. Por ejemplo:

- ¿En el caso de servicios PaaS y SaaS, el servicio ofrecido está disponible en más de un proveedor?
- ¿El servicio ofrecido está disponible en múltiples regiones nacional e internacionalmente?
- ¿La misma versión del mismo servicio se encuentra disponible en una o más regiones del mismo país?
- ¿En cuáles regiones alrededor del mundo?
- ¿El proveedor garantiza la misma experiencia (controles de seguridad, cumplimiento de estándares, rendimiento, características, etc) en todas las regiones?

Las fuentes revisadas a lo largo de este trabajo demuestran que incluso en proveedores a hiper-escala, la disponibilidad de servicios y la consistencia de la experiencia de usuarios todavía varían de región a región (incluso dentro de los Estados Unidos). La mayoría de ellos opera más de un Data Center a los cuales se los llama a veces zonas en cada una de esas regiones. La cobertura regional es una cuestión importante a analizar, ya que lo más cerca que se encuentre el suscriptor de una zona

de disponibilidad dentro de la región, más rápida y más consistente será la entrega de servicios. Para mitigar estas situaciones, algunos operadores realizan tareas de caché local para ayudar a reducir las latencias de datos consultados con más frecuencia; lo cual además permite a los proveedores alcanzar más fácilmente las condiciones de SLA de sus acuerdos con clientes.

Para Swoyer, otro punto importante a considerar debe ser el rendimiento de sistemas críticos (tales como bases de datos, *data warehouses* y aplicaciones Enterprise), los proveedores de PaaS tienden a ofrecer SLAs que son más afines a los clientes que aquellos basados en servicios IaaS. La razón de esto es simple: en el modelo PaaS, el proveedor es el responsable de gestionar y escalar el servicio (como así también de los recursos virtuales necesarios para alimentarlo), lo cual hace posible para los proveedores de PaaS soportar criterios de rendimiento y disponibilidad más granulares. En contraste, en el modelo IaaS, a los suscriptores se les requiere que configuren, mantengan y escalen los recursos virtuales y el *software*.

La vertiginosa evolución y mejora de la nube

En este aspecto, el autor manifiesta que el problema con los SLAs tiene otro inconveniente más: la constante e inexorable mejora de los servicios de infraestructura, sobre todo en el caso de PaaS. Los proveedores continúan poniendo foco en los puntos débiles del paradigma de la nube, especialmente respecto del rendimiento y a las latencias sub-óptimas de los recursos. Hoy por hoy los servicios PaaS están siendo provistos por interfaces de usuario de alto nivel para poder gestionar sus *offerings* y al mismo tiempo empiezan a adoptar e incorporar *Machine Learning* para incorporar más poder a sus servicios, o por ejemplo agregar automatización de tipo *rule-driven* para monitorear el rendimiento, detectar inconvenientes y disparar una o más acciones predefinidas. Esto resulta en una ventaja competitiva que es de gran ayuda al negocio, puesto que grandes proveedores ponen a disposición este tipo de tecnologías integradas con modelos de ML e IA y de esta forma entrenan a sus motores para poder disparar mejores y más acertadas decisiones para remediar los problemas y

mejorar así la calidad del servicio de automatización con el que resuelven las necesidades de soporte de sus clientes y de los clientes de sus clientes. Así, la misma información permitirá evolutivamente captar mejores pistas a tareas generales, procesos o casos de uso que los proveedores pueden automatizar o simplificar.

Comentarios finales

Una organización no está partida a la mitad si queda a mitad de camino y extiende no todos sus servicios sino solamente sus servicios de infraestructura y operaciones de IT a un contexto de nube; sino que se transforma en una totalmente nueva. La nube, en esencia, también tiene una componente de *on-prem* ya que lo que la empresa deja de tener dentro de sus instalaciones para subcontratarse a un proveedor tercero, termina estando en las instalaciones de ellos. Por ello, la nube no escapa al concepto de Data Center por más que delegue sus funciones a otro. Una estrategia de nube debe tener esto en mente. El punto de extender las operaciones de la nube (ya sea pública o privada) no debe ser solamente deshacerse de lo *legacy*, de los recursos costosos o poco confiables; ni tampoco el querer posicionarse mejor para navegar golpes o disrupciones. Mucho menos ser simplemente adjuntada a un esfuerzo estratégico por mejorar o transformar el negocio.

Estos son, sin dudas, todos beneficios de la nube, pero la razón por la cual se migra a la misma debe ser más que simplemente aprovechar sus bondades, y hacerlo como parte de un crecimiento estratégico-orgánico, dentro de un plan donde la migración hacia la nube deberá ir de la mano de un crecimiento y desarrollo de toda la empresa en conjunto.

7. Anexos

7.1. Entrevistas a expertos

7.1.1. Participantes

Andrés Nicolás:

- **Posición:** Head of Technology, Security and Infrastructure.
- **Empresa:** Koibanx

Alejandro Loiacono:

- **Posición:** DevOps Lead
- **Empresa:** Ernst & Young

Ariel Masciotta:

- **Posición:** Lead DevOps Engineer
- **Empresa:** Ernst & Young

7.1.2. Preguntas

- **Cuáles piensa son los desafíos más grandes que enfrentan los Data Centers?**

Andrés: En primer lugar me gustaría hacer una distinción entre los Data Centers *Legacy* y los Data Centers modernos utilizados por los proveedores de nube.

Es cada vez más frecuente la utilización de servicios nube, teniendo así las empresas más tiempo para enfocarse en su *core-business*, sin tener que preocuparse por mantenimiento eléctrico, cableado, temperatura, seguridad física, espacio ocioso, compra de hardware, etc.

Esta ventaja competitiva de los servicios nube es, sin duda, uno de los desafíos más grandes que enfrentan los Data Centers *Legacy*.

Asimismo los Data Center nube que dan flexibilidad a las empresas, *pay-as-you-go*, portales administrativos donde se puede auto-provisionar máquinas virtuales, *bare-metal* o arquitecturas *serverless*, también tienen desafíos que afrontar: el escepticismo, marco regulatorio-legal de industrias y países.

Alejandro: Escalabilidad, costos, inversión inicial. Existen diferentes generaciones de Data Centers *on-prem*. Pero al mismo tiempo existen nuevas generaciones de Data Centers en nubes privadas. En ese caso se puede hacer un *re-purpose* de toda esa infraestructura y rearmar una nube privada *on-prem* para brindar un nuevo negocio. Se mueve de CapEx a Opex. Con lo cual te podés enfocar en tu negocio. Dejás así de preocuparte de problemas de infraestructura y pasás a ocuparte del negocio puramente. Se habla de 2 a 6 meses como tiempo necesario para dejar un middleware listo para operar iniciando desde la primera orden de compra. Para lo cual muchas empresas compran hardware de más para adelantarse al

crecimiento. Hablamos de costos de 50.000 USD para comprar e instalar un server físico contra 100 USD al mes de gasto en la nube por mes (más soporte aparte). Para empresas grandes, la ventaja es mucho mejor, por la escala. A partir de que se contrata un servicio en la nube, las empresas se olvidan de problemas como la redundancia. Muchas veces, la ventaja de la nube se da porque ahí el foco es en negocio, no en ahorro de dinero, gano en *time-to-market*.

Ariel: Un Data Center requiere personal idóneo en múltiples especialidades para poder operar y ser mantenido, proveedores, espacio físico y planeamiento. A su vez, soportar la rápida expansión de los requerimientos de infraestructura con la velocidad de aprovisionamiento demandada en los últimos años no es una tarea fácil para un centro de procesamiento que está limitado por espacio físico. Agregar hardware en un Data Center puede ser una tarea ardua y lenta, aún para una empresa ágil. A su vez un Data Center solamente puede escalar hacia arriba, y en general, escalar hacia abajo significa una pérdida de recursos y tiempo.

- **Cuáles piensa son los desafíos más grandes que enfrentan las tecnologías de la nube?**

Andrés: Mejor eficiencia y sostenibilidad ecológica, optimización de costos y Adopción de nuevas tecnologías.

Alejandro: La seguridad por sobre todas las cosas es una ventaja. Pero también los beneficios son un *trade-off*, porque lo que gano por un lado lo pierdo por otro al ser cliente cautivo del proveedor. Los *offerings* que tienen los proveedores no son los mismos, cada uno maneja un *stack* de tecnologías y cambiarme a otro es complejo. Las *features* entre sí no son intercambiables. Cuanto más exprimo los beneficios de esa nube, más cautivo soy.

Ariel: Las tecnologías en la nube requieren de aplicaciones desarrolladas para tal fin, y no simples adaptaciones de las aplicaciones clásicas que abundan en una empresa acostumbrada a basar sus cómputos en un Data Center. Las apps deben estar desarrolladas íntegramente como micro servicios para poder hacer el mejor uso de la infraestructura provista por los proveedores de servicios. Sin tampoco dejar de lado las restricciones de algunas compañías que por motivos (algunas veces faltos de fundamentos sólidos) de seguridad no pueden migrar su cómputo a la nube.

- **En su opinión, una solución que pretenda automatizar de principio a fin el provisionamiento de infraestructura, qué *stack* de tecnologías debería contemplar y con qué propósito?**

Andrés: Sin lugar a duda debería contemplar tecnologías *serverless*, git-ops, WAF, *honey-pot*, bases de datos, etc. Poder desplegar soluciones de manera segura, dinámica, escalable con mínima intervención de equipos de infraestructura es un modelo operativo con ventajas competitivas estratégicas que permite a las empresas crecer en servicios sin necesariamente aumentar sus equipos de infraestructura.

Alejandro: Sí o sí se debe entrar por una nube, con lo cual hay que conocer el lenguaje de esa nube (ARM, Terraform, etc). GitHub o similar para manejar el versionado de código, Jenkins para manejar los pipelines de CI/CD, la parte

de contenedores con Docker y Kubernetes completo, notificación gestionada a través de la API de SendGrid y la parte de *workflows* con PowerAutomate para gestionarlos gráficamente (*codeless*).

Ariel: Kubernetes – Encargado de manejar los contenedores y deployments de los microservicios. jFrog Artifactory/DockerHub – Hospedar las imágenes de Docker y/o los paquetes npm, o Nuget producidos por los distintos *stacks*. Github – Encargado de hospedar los repositorios de código. Azure – Encargado de proveer la infraestructura (AKS, Gateways, VNETs, VMs, Azure Functions, etc). Azure DevOps – Encargado de proveer la plataforma de automatización para crear los pipelines asociados a los repositorios.

Lenguajes: DotNet, NodeJs, Python. (Abunda personal capacitado en el mercado).

- **En su opinión, cómo debería estar conformado un modelo de soporte con escalamiento ideal en una empresa cuya infraestructura está totalmente automatizada en la nube?**

Andrés: Idealmente deberían existir 3 cosas: 1-Sistema de monitoreo capaz de generar alertas y *triggers*. 2- La plataforma debería tener la suficiente resiliencia y autonomía para tomar acciones según los *triggers* que reciba de monitoreo.

3- Un equipo de soporte que pueda lidiar con los problemas que escapen la automatización y que a su vez sea capaz de automatizar nuevos hallazgos de manera tal que la intervención humana sea cada vez menos frecuente.

Alejandro: Depende de la escala de la empresa. En determinadas fechas del mes, Ernst & Young puede llegar a ser el primer, segundo o tercer consumidor de Azure del mundo porque la demanda fluctúa dependiendo de la parte del mes donde se esté. A partir de que nos movemos a la nube, los soportes son de producto y los equipos trabajan en función de esa premisa. El equipo de soporte tendrá Dev, QA y DevOps/SRE para soporte a producción, que se encargará del monitoreo y mejoramiento de la solución. Que la infra sea redundante no quiere decir que el producto sea redundante. En esta instancia es más importante pensar más en un modelo de soporte de infra en el soporte de producto. Agregaría una herramienta de tracking de errores, problemas y necesidades (herramientas de *backlog* o un *chatbot* combinado con PowerAutomate que me genere los tickets automáticamente.

Ariel: Debe contar con personal idóneo desde su nivel 1 para poder reportar los posibles problemas de sus aplicaciones.

Soporte Nivel 1 – Descartar problemas de usuarios/Workstations

Soporte Nivel 2 – Analizar posibles errores en la aplicación y descartar los falsos positivos

Soporte Nivel 3 – Especialistas en los principales *stacks* mencionados anteriormente que puedan investigar cualquier escalamiento de los niveles anteriores. Estar en contacto con los proveedores en caso de necesitar soporte de los proveedores.

Ingeniería – Equipo encargado del planeamiento y desarrollo de la arquitectura y automatizaciones requeridas para el correcto ciclo de vida de una aplicación.

- **Cuáles visualiza como los problemas del futuro respecto del provisionamiento de infraestructura? Hacia dónde apuntan las tendencias?**

Andrés: En líneas generales no veo problemas con el provisionamiento de infraestructura, más que nada porque es cada vez menos frecuente su uso, en camino a ser un *commodity* manejado detrás de la "automagia" de la nube. Las tendencias apuntan hacia soluciones de *infraestructure-as-a-code*, terraform, cloudformation, contenedores, microservicios, etc.

Alejandro: Nos alejamos de la infraestructura y nos enfocamos en servicios. Hay que tomar una decisión en algún punto de hasta dónde quiero tener control/governance. Ver cuál es el *trade-off* de beneficios y responsabilidades delegadas a terceros.

Ariel: Los Data Centers tienen un punto finito de expansión, el espacio es uno de ellos. La infraestructura como servicio (IaaS) ya es algo que existe hace mucho tiempo y cada vez más empresas se dan cuenta que las Plataformas como Servicio (PaaS) es algo actual y que seguirá siendo así. Esto permite delegar la administración, mantenimiento y seguridad de las plataformas a un proveedor, ahorrando importante dinero y tiempo en contratar los administradores necesarios para poder crearla/mantenerla/soportarla. De esta manera la empresa solo necesita un equipo de arquitectos/desarrolladores/DevOps/QA que pueden dedicarse únicamente a desarrollar su aplicación/conjunto de aplicaciones, automatización y testeo.

- **Comentarios adicionales del tema en cuestión?**

Andrés: No se debe subestimar la velocidad con la que está avanzando la tecnología y la resistencia de empresas, industrias y sectores, muchas veces esta resistencia ha dejado a grandes jugadores fuera del mercado.

Cabe destacar que las ventajas técnico-financieras de la nube han permitido el desarrollo vertiginoso de nuevas empresas e industrias.

Alejandro: Las nubes cuando el cliente se vuelve muy grande no están preparadas para que se haga *reselling* de sus servicios. Están pensadas para interactuar dentro de la empresa que consume el servicio. La capa que gestionaría el *reselling* no existe en el proveedor y debe construirse. Por otro lado, muchas de las *features* que otorgan los proveedores dentro de su *offering* ya funcionan *out-of-the-box*, pero cualquier cambio debe desarrollarse, como así también cualquier modificación al standard; lo cual también requiere constante actualización técnica porque los proveedores sacan

regularmente nuevos *offerings* que compiten o coinciden con las personalizaciones que cada cliente ya tiene desarrolladas. El negocio de las nubes en IaC era hace 5 años, hoy son los productos y servicios y cómo los clientes interactúan con la nube.

Ariel: En algunas localizaciones puede no haber buenos enlaces o el cambio puede no ser favorable para contratar servicios de la nube. Esto puede complicar la adopción de micro servicios, ya que conseguir profesionales especializados en Kubernetes es una tarea difícil y no es personal de bajo costo.

La diferencia entre un proveedor de nube de primer nivel (Azure, AWS, GCP) y los locales suele ser extremadamente grande, con lo cual, la falta de posibilidad de ir con uno de ellos, puede marcar el destino del desarrollo de aplicaciones de la compañía en cuestión.

8. Referencias y Bibliografía

Adler, B. (21 de Marzo de 2022). *Cloud Computing Trends: Flexera 2022 State of the Cloud Report*. Obtenido de Flexera: https://www.flexera.com/blog/cloud/cloud-computing-trends-2022-state-of-the-cloud-report

Aggarwal, Aditya Patawari, & Vikas. (2018). *Ansible 2 Cloud Automation Cookbook.* Packt Publishing.

Amazon. (2020). *Capital One Completes Migration from Data Centers to AWS, Becomes First US Bank to Announce Going All In on the Cloud*. Obtenido de Amazon AWS: https://aws.amazon.com/solutions/case-studies/capital-one-all-in-on-aws/?did=cr_card&trk=cr_card

Amazon. (2020). *Emirates Flies High After Massive AWS Migration*. Obtenido de Amazon AWS: https://aws.amazon.com/solutions/case-studies/emirates-case-study/?did=cr_card&trk=cr_card

Amazon. (2020). *Yedpay Equips Small Businesses with a Highly Secure Payment Platform Using AWS*. Obtenido de Amazon AWS: https://aws.amazon.com/solutions/case-studies/yedpay/?did=cr_card&trk=cr_card

Amazon. (s.f.). *AWS Pricing Calculator*. Obtenido de Amazon AWS: https://calculator.aws/

Amazon. (s.f.). *Under Armour Case Study*. Obtenido de Amazon AWS: https://aws.amazon.com/solutions/case-studies/under-armour/

amith136. (s.f.). *I will setup entire devops working platform on cloud or on premises*. Obtenido de fiverr: https://www.fiverr.com/amith136/setup-entire-devops-working-platform-on-cloud-or-on-premises

Bas Meijer, Lorin Hochstein, & René Moser. (2022). *Ansible: Up and Running, 3rd edition.* O'Reilly.

Bill Gates. (n.d.). Retrieved from https://ai100.stanford.edu/

Brendan Burns, Joe Beda, & Kelsey Hightower and Lach. (2022). *Kubernetes: Up and Running, 3rd Edition.* O'Reilly.

Cisco. (s.f.). *Hybrid Cloud e-book*. Obtenido de Cisco: https://www.cisco.com/c/m/en_us/solutions/cloud/hybrid-cloud-ebook.html

Cloud. (s.f.).

CloudSkew. (s.f.). *Free Kubernetes Architecture Diagram tool*. Obtenido de CloudSkew: https://docs.cloudskew.com/free-kubernetes-architecture-diagram-tool.html

Covey, Steven. (s.f.).

Daniel Oh, James Freeman, & Fabio Alessandro Locati. (2020). *Practical Ansible 2.* O'Reilly.

Javatpoint. (s.f.). *Features of Cloud Computing*. Obtenido de Javatpoint: https://www.javatpoint.com/features-of-cloud-computing

Javatpoint. (s.f.). *Introduction to Cloud Computing*. Obtenido de Javatpoint: https://www.javatpoint.com/introduction-to-cloud-computing

Jeffrey Nickoloff, & Stephen Kuenzli. (2019). *Docker in Action, 2nd Edition.* O'Reilly.

Jimenez, A. (s.f.). *Que es IaaS, PaaS y SaaS*. Obtenido de Comparativa Clouds: https://comparacloud.com/servicios-clouds/iaas-paas-saas/

Jobs, Steve. (s.f.).

Kaplarevic, V. (19 de Noviembre de 2019). *Understanding Kubernetes Architecture with Diagrams*. Obtenido de phoenixNAP: https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams

Kawasaki, Guy. (s.f.).

Keller, Helen. (s.f.).

Kobilinskiy, A. (18 de December de 2019). *Cloud Service Models: SAAS, PAAS, IAAS - Which Is Better For Business*. Obtenido de Dev.to: https://dev.to/artemkobilinskiy/cloud-service-models-saas-paas-iaas-which-is-better-for-business-574k

Kumar, V. &. (2019). *Comparison of Fog Computing & Cloud Computing. International Journal of Mathematical Sciences and Computing. 5. 31-41. 10.5815/ijmsc.2019.01.03*. Obtenido de

Researchgate: https://www.researchgate.net/figure/Cloud-Computing-Service-Model-15_fig1_330090457

Loten, A. (19 de Agosto de 2019). *Data Center Market is blooming amid shift to cloud*. Obtenido de The Wall Street Journal: https://www.wsj.com/articles/data-center-market-is-booming-amid-shift-to-cloud-11566252481

Luksa, M. (2018). *Kubernetes in Action.* Manning Publications.

Maraboli, Steve. (s.f.).

Markets, R. a. (13 de Febrero de 2020). *Comprehensive Data Center Market Outlook and Forecast 2020-2025*. Obtenido de GlobeNewswire: https://www.globenewswire.com/news-release/2020/02/13/1984742/0/en/Comprehensive-Data-Center-Market-Outlook-and-Forecast-2020-2025.html

Massachusetts Institute of Technology. (s.f.). *Kerberos: The Network Authentication Protocol*. Obtenido de https://web.mit.edu/kerberos/

Massachusetts Institute of Technology. (s.f.). *Red Hat Enterprise Linux 4: Manual de referencia*. Obtenido de https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html

Morris, K. (2020). *Infrastructure as Code, 2nd Edition.* O'Reilly.

Nassim Kebbani, Piotr Tylenda, & Russ McKendrick. (2022). *The Kubernetes Bible.* Packt Publishing.

Nigel Poulton. (2020). *Docker Deep Dive.* O'Reilly.

Palo Alto Networks. (s.f.). *What is a Data Center*. Obtenido de Palo Alto Networks cyberpedia: https://www.paloaltonetworks.com/cyberpedia/what-is-a-data-center#:~:text=A%20data%20center%20is%20a,the%20continuity%20of%20daily%20operations.

Panetta, K. (10 de Octubre de 2019). *Is the Cloud Secure?* Obtenido de Gartner: https://www.gartner.com/smarterwithgartner/is-the-cloud-secure/

Powell, Colin. (s.f.).

Prabhu, V. (19 de February de 2019). *WHAT IS CLOUD COMPUTING - AN INTRODUCTION FOR ABSOLUTE BEGINNERS*. Obtenido de YoungWonks: https://www.youngwonks.com/blog/What-is-Cloud-Computing---An-Introduction-for-Absolute-Beginners#:~:text=And%20cloud%20computing%20means%20storing,level%20services%20over%20the%20Internet.

Prog.World. (s.f.). *Introducing Ansible Automation Platform 2*. Obtenido de Prog.World: https://prog.world/introducing-ansible-automation-platform-2/

Ramchandani, L. (26 de Agosto de 2020). *The Importance of a QA Environment* . Obtenido de TestProject by Tricentis: https://blog.testproject.io/2020/08/26/the-importance-of-a-qa-environment

Red Hat. (31 de enero de 2018). *¿Qué son la integración y la distribución continuas (CI/CD)?* Obtenido de Red Hat: https://www.redhat.com/es/topics/devops/what-is-ci-cd

Red Hat. (15 de Marzo de 2018). *Types of cloud computing*. Obtenido de RedHat: https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud

Red Hat. (9 de Enero de 2018). *What is Docker*. Obtenido de Red Hat: https://www.redhat.com/en/topics/containers/what-is-docker

Red Hat. (21 de Julio de 2022). *Public cloud vs private cloud and hybrid cloud*. Obtenido de Red Hat: https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud

Red Hat. (11 de Mayo de 2022). *Red Hat Ansible Automation Platform*. Obtenido de Red Hat: https://www.redhat.com/en/resources/ansible-automation-platform-datasheet?extIdCarryOver=true&sc_cid=701f2000001OH7EAAW

Red Hat. (s.f.). *How Ansible Works*. Obtenido de Red Hat Ansible: https://www.ansible.com/overview/how-ansible-works?hsLang=en-us

Spice Works. (4 de Marzo de 2019). *The 2019 State of Servers*. Obtenido de Spiceworks: https://community.spiceworks.com/blog/3182-the-2019-state-of-servers

StarAgile. (22 de Marzo de 2021). *DevOps Roadmap 2022: How to Become a DevOps Engineer in 2022*. Obtenido de StarAgile: https://staragile.com/blog/devops-roadmap

Steve Swoyer. (2021). *Migrating Applications to the Cloud.* O'Reilly.

Swoyer, S. (2021). *Migrating Applications to the Cloud.* O'Reilly.

Takahashi, K. (Septiembre de 2019). *A Study on Portable Load Balancer for Container Clusters*. Obtenido de ResearchGate: https://www.researchgate.net/publication/337971562_A_Study_on_Portable_Load_Balancer_for_Container_Clusters

Taulli, T. (2020). *The Robotic Process Automation Guideline.* Apress.

Telecommunications Industry Association. (s.f.). Obtenido de Telecommunications Industry Association: https://tiaonline.org/

Terraform. (s.f.). *Intro to Terraform*. Obtenido de Terraform - HashiCorp: https://www.terraform.io/intro

Tripathi, A. M. (2018). *Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool – UiPath.* Packt Publishing.

Wu, C., & Buyya, R. (2018). *Cloud Data Centers and Cost Modeling.* Morgan Kaufmann.