Administración de Sistemas de la Cloud

Introducción

Índice

Esquema 3

Ideas clave 4

1.1. Introducción y objetivos 4

1.2. Escenarios: migración 5

1.3. Escenarios: entornos híbridos 6

1.4. Escenarios: aprovisionamiento 6

1.5. Escenarios: seguridad 7

1.6. Escenarios: disponibilidad 7

1.7. Servidores *vs.* equipos de escritorio 8

1.8. Linux *vs.* Windows 9

1.9. Tareas tradicionales 10

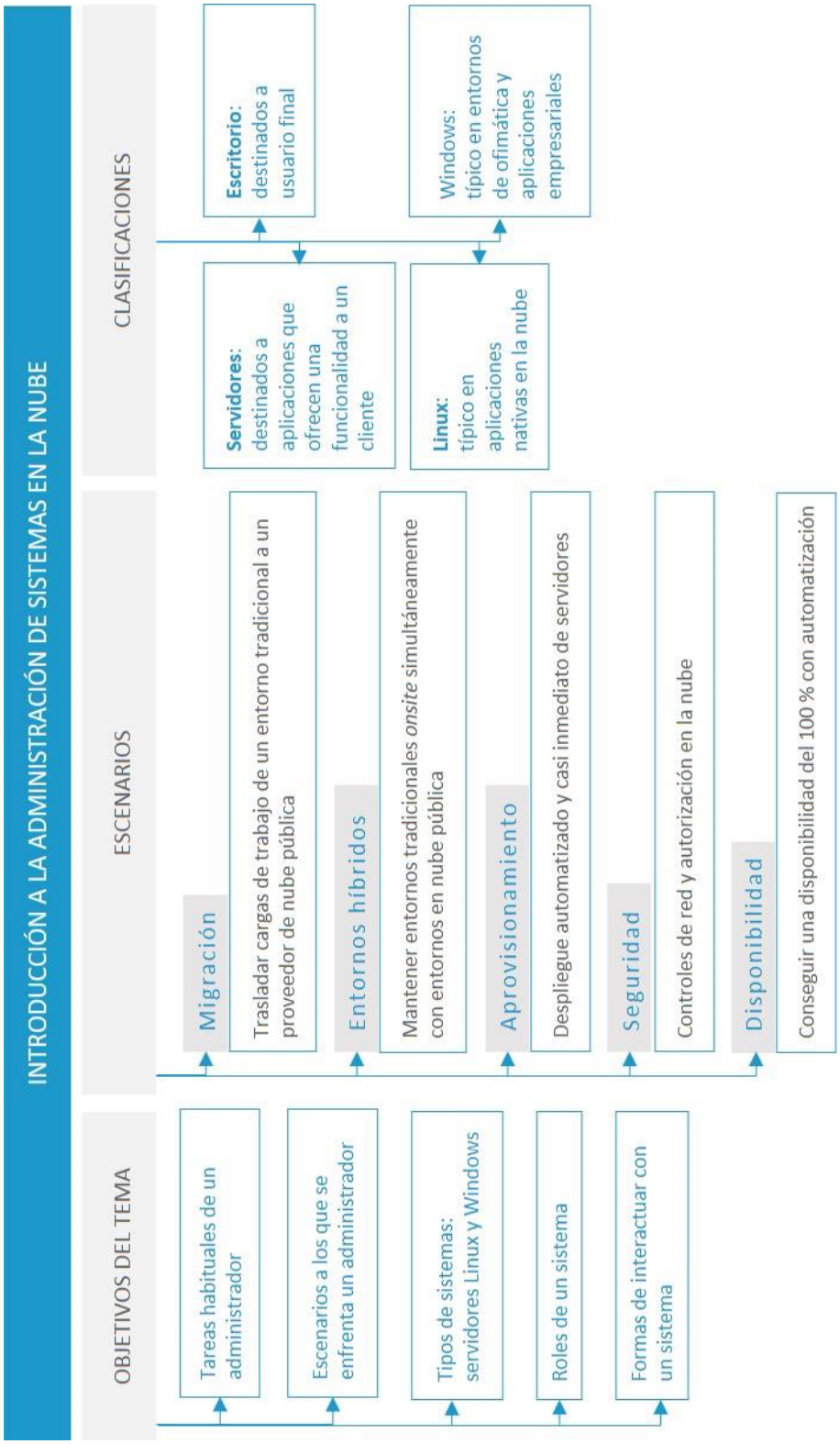
1.10. Tareas en la nube 15

1.11. Referencias bibliográficas 16

A fondo 19

Test 20

Esquema



Ideas clave

1.1. Introducción y objetivos

En este tema, se presenta el concepto de administración de sistemas en el contexto de sistemas operativos y se plantean las tareas que se esperan de un administrador de sistemas. También se presentarán otros sistemas.

Debido a la aparición de la computación en la nube, el trabajo de un administrador de sistemas está cambiando y está derivando, en muchos casos, en la reconversión hacia los **DevOps,** *development opereations*, (Both, 2018). A medida que crece la adopción de la nube, la atención cambia de la administración de recursos físicos a la administración de sistemas virtuales y del mantenimiento de las máquinas a ser capaz de replicar los entornos para garantizar el servicio.

Las empresas buscarán a los administradores de sistemas y DevOps con la capacidad de liderar la transformación y el despliegue de **entornos en la nube.** Nos centraremos en los elementos generales que pueden resultar de utilidad para cualquier entorno y que serán demandados por un entorno de DevOps.

Los **objetivos** que se pretenden conseguir en este tema son:

- Conocer el alcance de las tareas habituales de un administrador de sistemas.
- Conocer los escenarios a los que se enfrenta un DevOps en un entorno de nube.
- Identificar los diferentes tipos de sistemas operativos.
- Diferenciar los roles que puede tener un sistema.
- Identificar las formas de interactuar con un sistema.

A continuación, en el vídeo *Introducción a Administración de Sistemas para Cloud*, se resumen los principales puntos que se desarrollarán en este tema.


Accede al vídeo

1.2. Escenarios: migración

Uno de los primeros retos de un administrador será la migración de cargas de trabajo de infraestructuras físicas y virtualizadas de un centro de datos privado, probablemente *onsite,* a un **proveedor de nube pública.** *Onsite* tiende a referirse a una localización propia de la organización, frente a una ubicación *offsite*, a la que la organización tiene acceso por red. Un entorno *offsite* tradicional podía ser un centro de datos o una parte de un centro de datos subalquilado a un proveedor. En estos casos, la organización cliente podía llegar a optar por administrar los servidores físicos y subcontratar el espacio y la electricidad.

¿Tiene sentido reconstruir todo el sistema desde cero? ¿Es posible crear una imagen de sistema existente que ofrece el 80 % más de la funcionalidad necesaria? La solución más sencilla puede ser hacer una copia del sistema, copiarla al proveedor de la nube y arrancarla en la nueva ubicación. Esta opción permite, entre otras cosas, **delegar la administración** de la infraestructura física y virtual al proveedor y mantener el funcionamiento familiar al que están acostumbrados los administradores. No obstante, esta opción no permitirá gestionar el ciclo de vida de las aplicaciones existentes.

En un entorno DevOps, en el que se tienda a la automatización, se debería apostar por **automatizar el despliegue,** al menos, de todas las aplicaciones nuevas y, en lo posible, de las existentes, para poder disfrutar de todas las ventajas de un entorno de nube.

1.3. Escenarios: entornos híbridos

La integración con los entornos existentes es casi inevitable cuando se emprende la migración de entornos tradicionales a la nube. Esto requiere el aprovisionamiento de recursos, tanto locales como de la nube, con protocolos compatibles. Un DevOps tendrá que combinar sus conocimientos de desarrollo y administración con **conocimientos de red,** para establecer las conexiones adecuadas. Estos recursos de red serán en muchos casos virtuales y, por lo tanto, también deben ser automatizados.

1.4. Escenarios: aprovisionamiento

La implantación del *cloud computing* permite, finalmente, que los DevOps dispongan de **recursos renovables, reciclables y fáciles de proveer.** Crear un nuevo servidor es tan simple como unos pocos clics en una consola web. El reemplazo de un servidor existente es sencillo, si está automatizado su despliegue y configuración. Además de la facilidad, el aprovisionamiento es mucho más rápido en la nube que en los entornos tradicionales. Si la carga de un servidor es muy alta, es suficiente con lanzar más instancias o, en el peor de los casos, lanzar un servidor nuevo de mayor tamaño, utilizando las automatizaciones.

Administrar un entorno en la nube significa más que mantener todos los sistemas funcionando. Los DevOps deben ahora **vigilar el coste de funcionamiento** del entorno. Los costes de *hardware* se han reemplazado por el pago por uso, a veces, pero no siempre, con el coste de licencia incluido.

Los contratos de soporte se han trasladado de los proveedores de *hardware* a los proveedores de infraestructura. La comprensión de este equilibrio ayudará a los DevOps a trabajar con otros para construir una solución rentable y confiable.

1.5. Escenarios: seguridad

La seguridad es una preocupación importante en la nube. Los proveedores son responsables de asegurar el acceso físico al *hardware.* Sin embargo, la infraestructura debe ser accedida de forma remota por definición. Los DevOps deben estar familiarizados con el **modelo de seguridad** de su proveedor de nube.

Los roles, grupos, usuarios y políticas son mecanismos comunes para **otorgar y restringir el acceso.** La mayoría de los proveedores de nube incorporan estas posibilidades de granularidad entre sus herramientas de administración. Por ejemplo, una directiva puede conceder permiso para que un usuario cree una instancia en una región y la deniegue en otra. Al igual que los entornos tradicionales, la seguridad debe ser una consideración inicial.

Permitir o bloquear el acceso a la red es un trabajo tradicionalmente de los administradores de red. En la nube es habitual que este trabajo recaiga en los DevOps, que aprovecharán la automatización para reducir los riegos de seguridad de red.

1.6. Escenarios: disponibilidad

Uno de los objetivos de contar con recursos en la nube será que tengamos una **disponibilidad** (*uptime*) cercana al 100 %. Esto no es algo que pueda lograrse de forma automática por el mero hecho de utilizar la nube. Si bien es cierto que se podría pensar que al usar la nube no habrá caídas en los sistemas por fallos de *hardware*, no es menos cierto que las máquinas en la nube a veces necesitan ser movidas para tareas de mantenimiento o incluso apagadas por parte del proveedor. En este escenario, muy cercano a la realidad, la única forma de garantizar el *uptime* es contar con mecanismos que permitan desplegar flotas de servidores autogestionados.

El concepto de **flota** se puede concretar en un **grupo de autoescalado** en AWS (Amazon Web Services) o en un **despliegue** en Kubernetes, por citar algunos. Los objetivos detrás de un grupo de servidores autogestionados son:

- Si un servidor deja de funcionar, debe ser posible apagarlo sin intervención humana.
- Si la flota actual no tiene capacidad suficiente, debe ser posible añadir un servidor nuevo sin intervención humana.
- Una actualización incremental no debe implicar una caída del sistema.

Esta idea de flota, en la que los servidores aparecen y desaparecen sin intervención humana, queda reflejada en el **paradigma de las mascotas y el ganado** (*pets vs. cattle*) (Bias, 2016). Los servidores tradicionales eran tratados como mascotas: provisionar uno llevaba mucho tiempo y los administradores hacían lo posible para mantenerlo funcionando todo el tiempo posible. Los servidores modernos en la nube se parecen más al ganado: es posible tener tantos que no merece la pena preocuparse por cada uno en particular.

Una disponibilidad del 100 % no significa que los DevOps trabajen 24/7 para garantizarla. La automatización es la base sobre la que se construyen las aplicaciones de escalado y de despliegue.

1.7. Servidores *vs.* equipos de escritorio

Los apartados anteriores mencionaban tareas como el despliegue de servidores. En un entorno tradicional, un servidor podría referirse tanto al equipo informático físico, formado por la CPU (*central processing unit*), memoria, discos e interfaces de red, como al proceso de *software* que ofrece una funcionalidad a un cliente a través de la red. Aquí, el concepto de servidor se limita al de una instalación de un sistema operativo pensado para ofrecer ejecutar aplicaciones para clientes.

Los sistemas operativos usados como servidores no son muy diferente de los que ejecutan los ordenadores de sobremesa o los portátiles. Ambos tipos tienen un núcleo, una arquitectura de procesador habitualmente compatible (por ejemplo, x86), controladores de *hardware* y utilidades de *software*. Muchos de los ejemplos descritos pueden ejecutarse también en un entorno de escritorio sin cambio alguno.

Sin embargo, una instalación de servidor suele tener **dos características relevantes:** solo es posible interactuar con él por red y no tiene entorno gráfico. Un DevOps deberá escoger herramientas que le permitan adaptarse a ambas características. Como normal general, cuantas menos piezas tenga un sistema, menos pueden fallar. Si los servidores no tienen una pantalla conectada, ¿por qué no obviar el sistema gráfico totalmente?

La **línea de comandos,** esa pantalla en negro con texto que tan bien han vendido las películas sobre *hackers*, es el entorno habitual de trabajo de un administrador. Incluso Windows, que mantiene el concepto de interfaz gráfica en su nombre, introdujo PowerShell en 2007. La línea de comandos ofrece la posibilidad de escribir *scripts*: archivos con comandos y funciones propias de Bash o PowerShell que se pueden ejecutar en bloque. Efectivamente, facilitan la automatización.

1.8. Linux *vs.* Windows

La discusión entre **qué sistema operativo es mejor** ha alimentado multitud de foros, blog y artículos desde hace años (Economides y Evangelos, 2006; Casadesus-Masanell y Jordan, 2006; Newell, 2020). Los argumentos giran en torno al coste, la libertad de acceso al código fuente, la seguridad, la usabilidad, etc. Aquí se cubren ambos sistemas, porque, al fin y al cabo, son una herramienta y no un fin en sí mismo. Un argumento a favor en una situación (la usabilidad de Windows en un entorno de escritorio) puede no serlo en otra (en un entorno de servidor, la usabilidad del usuario puede pasar a un segundo plano).

La elección de uno u otro debe estar marcada por las necesidades de la **aplicación,** la **organización** y los **usuarios.** Por ejemplo, Linux es una gran opción para una aplicación nueva, diseñada con una arquitectura nativa en la nube, gracias al soporte nativo de contenedores. En el caso de una empresa con una gran base de usuarios que necesitan una *suite* ofimática y acceso a unas pocas herramientas corporativas, Windows parte con la ventaja de facilitar la administración de políticas de seguridad de manera centralizada.

En ambos ejemplos, se pueden dar razones a favor y en contra de uno y otro. Cuantas más herramientas domine un administrador, más preparado estará para poder evaluar dichas razones y tomar una decisión.

1.9. Tareas tradicionales

Cada organización define los roles y las responsabilidades de sus administradores. El ámbito de las tareas cambia de organización a organización y también cambia según evoluciona la organización, pero hay algunas tareas que cualquier administrador tiene (Kralicek, 2016).

- Instalación de servidores y clientes.
- Instalación y mantenimiento de aplicaciones.
- Creación de usuarios y grupos.
- Soporte a usuarios.
- Copias de seguridad y recuperación frente a desastres.
- Seguridad.
- Automatización de tareas.
- Instalación de periféricos como impresoras.
- Gestión de cambios.
- Configuración de equipos de red local.

Las organizaciones necesitarán más o menos tareas en función de su propio tamaño y del tamaño de sus departamentos. Además, en organizaciones suficientemente grandes, en entornos tradicionales que se acogen a ITIL, es habitual que se implanten roles y procesos estándares.

Muchos grupos de soporte de IT (*information technology*) **dividen las tareas en roles.** Esto facilita que cada rol cree una base de conocimiento profunda para resolver problemas y ejecutar tareas. Una separación de tareas en roles podría ser la siguiente:

- Servicios de usuario:
  - Nivel 2 de soporte a usuario.
  - Instalación de periféricos.
  - Instalación y mantenimiento de aplicaciones.
  - Instalación de equipos de usuario.

- Administración de servidores:
  - Instalación de servidores.
  - Creación de usuarios y grupos.
  - Copias de seguridad y recuperación frente a desastres.
  - Automatización de tareas.

- Seguridad IT:
  - Soporte de red.
  - Gestión de cambios.

En este ejemplo, las tareas se separan en **tres roles,** que facilitan la separación de responsabilidades y, por tanto, las operaciones y la resolución de problemas. Esto, a su vez, incrementa las habilidades y el entrenamiento técnico del personal en cada rol. Las organizaciones más pequeñas suelen unir estos roles. No hay una división ideal para todos los tamaños de organización y cada una adaptará su estructura de roles al tamaño de su flota de equipos, a la complejidad de las tareas o a ambos.

En este caso, muchas de las tareas de los administradores deberán ser automatizadas y genéricas; es decir, con la posibilidad de reusar gran parte de los *scripts* y de usar *scripts* disponibles *online* y en libros del sector.

Una manera de optimizar las horas de un equipo pequeño es **estandarizar el despliegue de equipos,** ya sean servidores o de escritorio. Cuanto más se parecen los servidores entre sí, más fácil es resolver los problemas que puedan aparecer en el día a día. Esto, además, facilita el soporte y mejora su calidad.

Operaciones

En departamentos grandes y maduros, los procedimientos y los mecanismos están bien establecidos. Estos procesos están integrados en aplicaciones de soporte y enlazan tanto aspectos operacionales como procesos clave de la administración del servicio, como gestión de cambios, control de configuración, inventario, catálogo de servicios, gestión de incidencias, etc. Estas aplicaciones son complejas y caras y requieren un conocimiento avanzado de los procesos de negocio y de la naturaleza técnica del mismo.

En departamentos pequeños, sin embargo, el nivel de madurez de la documentación de los procesos y de la gestión del conocimiento puede no ser homogénea. En estos casos, un administrador puede usar una lista diaria o semanal para gestionar sus tareas, además de coordinarse con otros equipos para asegurarse de que todos los sistemas funcionan correctamente.

Comunicación

Es habitual no tener en cuenta que **la información debe fluir en dos sentidos.** En muchos departamentos, los administradores reciben peticiones de soporte y se ven obligados a cerrarlas rápido, penalizando la calidad del soporte, debido a las métricas con las que se los evalúa a final de año. Esto limita la involucración del usuario que

inició la comunicación, así como del resto de los individuos que han formado parte de la resolución.

Tanto la comunicación con otros equipos como la interna benefician a todas las partes. Internamente, estar al tanto de lo que ocurre o de cómo se ha solucionado un problema puede ayudar en futuros problemas. La documentación de estas soluciones se suele llevar a cabo en un *knowledge base* (KB), o **base o biblioteca de conocimiento.** Incluso, algunas organizaciones publican estos KB al exterior (bien públicamente o al menos a un sector interno más amplio que el propio equipo de IT), para facilitar la resolución proactiva de problemas.

Esto último enlaza con la necesidad de educar e informar al usuario. Si los usuarios finales reciben información sobre la resolución de sus problemas, a largo plazo facilitan el trabajo de los equipos de soporte. También ayuda al hacer que todos estén al tanto de las tareas del día a día y el mantenimiento preventivo. Un objetivo de los departamentos es tener una **política 90-8-2:**

- Los usuarios resuelven por sí mismos el 90 % de las incidencias.
- Los grupos de soporte intermedios se encargan de resolver un 8 %, que serán situaciones más complicadas, pero normalmente documentadas y en las que no hay que involucrar a un experto.
- Los administradores solo reciben un 2 % de los problemas. Así se pueden dedicar a tareas en las que añaden valor a la organización.

Investigación

Estar al tanto de las novedades en IT ayuda a ser proactivo con posibles problemas externos. Hay muchas revistas técnicas gratuitas (*Information Week*, *Redmond Magazine*, *Information Security Magazine*), así como excelentes sitios de investigación en la web (Whatis.com, EventID.net, Tech Republic). Ampliar los conocimientos con este tipo de recursos mejora la sensibilidad operacional.

Formación

Asistir a **cursos específicos de proveedores** para la certificación no solo ayuda al desarrollo profesional del administrador, sino que también beneficia a la organización, al aumentar su conocimiento en la gestión de problemas. De hecho, las certificaciones hacen que los administradores y la operativa de IT sean más respetadas por sus clientes y, en algunos casos, es un requisito para optar a un contrato. Proporciona experiencia interna reconocida por la industria y asegura a quienes utilizan sus servicios que la organización cumple con los estándares. Además, facilita el *networking* y no en el sentido técnico, sino en cuanto a las relaciones profesionales con otros individuos del sector.

Confianza

Un sistema bien administrado permite que las organizaciones lleven a cabo sus negocios de manera estable, sin necesidad de que los usuarios de las aplicaciones tengan que comprender ni su arquitectura ni la operativa necesaria. Cuando se consiguen estos objetivos, se promueve la confianza de la organización y de los usuarios en el equipo de IT y viceversa.

¿Qué hay que cambiar?

Las ideas mencionadas en esta sección no son exclusivas de entornos tradicionales con centros de datos propios, servidores físicos y equipos de escritorio Windows, por citar un ejemplo lo más tradicional posible. Un equipo DevOps aplicará muchos de estos principios en su día a día: deberá organizar sus tareas, probablemente se especializarán en roles en función de sus fortalezas y, en cierta medida, estarán involucrados en tareas de soporte. Quizás no sigan ITIL a rajatabla, pero eso no impide que documenten sus procesos o estandaricen sus roles, tareas y herramientas.

Aunque las tareas técnicas cambien, los administradores de sistemas siguen teniendo el mismo objetivo común: apoyar a las organizaciones a conseguir sus objetivos de negocio.

1.10. Tareas en la nube

La función del administrador del sistema está cambiando. Hace solo unos años, la mayoría de los sistemas se ocupaban de granjas de servidores de *hardware* físico y realizaban una planificación detallada de la capacidad. Ampliar su aplicación significaba comprar un nuevo *hardware* y, tal vez, pasar tiempo acumulándolo en el centro de datos. Actualmente, hay un gran porcentaje de la industria que nunca ha tocado el *hardware* físico. Es posible desplegar servidores con una llamada a API (*application programming interfaces*) o haciendo clic en un botón en una página web (Lucifredi y Ryan, 2018).

Aunque el término ha sido apropiado por los equipos de *marketing*, la **nube** es algo sorprendente. En este contexto, se usa el término «nube» para referirse a la idea de servicios informáticos y de aplicaciones escalables a demanda, en lugar de servicios «basados» en la nube, como Google Mail.

A medida que aumenta la competencia en el espacio del mercado de la nube, su **atractivo,** para los administradores de sistemas y los propietarios de negocios, **aumenta casi a diario.** Amazon Web Services continúa impulsando el mercado de la computación en la nube al introducir con frecuencia nuevas herramientas y servicios (no hay más que repasar la velocidad con la que publican las novedades en What's New).

Las economías de escala están constantemente bajando el precio de los servicios en la nube. Aunque los entornos como AWS o Google Compute Engine aún no son adecuados para todas las aplicaciones, cada vez es más claro que las habilidades en

la nube se están convirtiendo en una **parte necesaria** de un conjunto de herramientas completo del administrador de sistemas.

Para las empresas, la nube abre nuevas **vías de flexibilidad.** Los equipos tecnológicos pueden hacer cosas que hubieran sido prohibitivamente caras hace solo unos años. Los juegos y las aplicaciones que tienen la suerte de convertirse en éxitos desbocados a menudo requieren una gran cantidad de capacidad de cómputo. Provisionar esta capacidad en horas en lugar de semanas permite a estas compañías responder rápidamente al éxito, sin entrar en inversiones y compromisos de gasto por varios años o gastos iniciales de capital.

En la era DevOps, los desarrolladores y la dirección saben lo importante que es iterar y mejorar rápidamente el código de aplicación. Los servicios de los proveedores de nube permiten tratar la infraestructura de la misma manera, permitiendo que un equipo relativamente pequeño administre infraestructuras de aplicaciones masivamente escalables.

1.11. Referencias bibliográficas

Axelos. (s. f.). *ITIL*. https://www.axelos.com/best-practice-solutions/itil

AWS. (s. f.). *Novedades de AWS*. https://aws.amazon.com/es/new/?whats-new-content-all.sort-by=item.additionalFields.postDateTime&whats-new-content-all.sort-order=desc&awsf.whats-new-analytics=*all&awsf.whats-new-app-integration=*all&awsf.whats-new-arvr=*all&awsf.whats-new-cost-management=*all&awsf.whats-new-blockchain=*all&awsf.whats-new-business-applications=*all&awsf.whats-new-compute=*all&awsf.whats-new-containers=*all&awsf.whats-new-customer-enablement=*all&awsf.whats-new-customer%20engagement=*all&awsf.whats-new-database=*all&awsf.whats-new-developer-tools=*all&awsf.whats-new-end-user-computing=*all&awsf.whats-new-

mobile=*all&awsf.whats-new-gametech=*all&awsf.whats-new-iot=*all&awsf.whats-new-machine-learning=*all&awsf.whats-new-management-governance=*all&awsf.whats-new-media-services=*all&awsf.whats-new-migration-transfer=*all&awsf.whats-new-networking-content-delivery=*all&awsf.whats-new-quantum-tech=*all&awsf.whats-new-robotics=*all&awsf.whats-new-satellite=*all&awsf.whats-new-security-id-compliance=*all&awsf.whats-new-serverless=*all&awsf.whats-new-storage=*all

Bias, R. (2016, septiembre 29). *The History of Pets vs Cattle and How to Use the Analogy Properly*. Cloudscaling. https://cloudscaling.com/blog/cloud-computing/the-history-of-pets-vs-cattle/

Both, D. (2018). *The Linux Philosophy for SysAdmins: And Everyone Who Wants To Be One* (cap. 1, pp. 3-14). Apress.

Casadesus-Masanell, R. y Jordan, M. (2006). *Linux vs. Windows*. Harvard Business School. https://www.hbs.edu/faculty/Pages/item.aspx?num=33719

Economides, N. y Evangelos, K. (2006). Linux vs. Windows: A comparison of application and platform innovation incentives for open source and proprietary software platforms. En Bitzer, J. y Srhroder, P. (Eds.), *The Economics of Open Source Software Development* (pp. 207-218). Elsevier.

Kralicek, E. (2016). *The Accidental Sysadmin Handbook* (2. ª ed.). Apress.

Lucifredi, F. y Ryan, M. (2018). *AWS System Administration*. O'Reilly Media.

Newell, G. (2020, mayo 29). *12 Reasons Why Linux Is Better Than Windows 10*. Lifewire. https://www.lifewire.com/windows-vs-linux-mint-2200609

Página de *Information Week* (https://www.informationweek.com/).

Página de *Infosecurity* (https://www.infosecurity-magazine.com/).

Página de *Redmond* (https://redmondmag.com/Home.aspx).

Página de TechRepublic (https://www.techrepublic.com/).

Página de WhatIs.com (https://whatis.techtarget.com/).

A fondo

La filosofía del SysAdmin

Both, D. (2018). *The Linux Philosophy for SysAdmins: And Everyone Who Wants To Be One* (cap. 1, pp. 3-14). Apress.

David Both comparte su forma de ver su profesión. Presenta las ideas que un administrador debe tener claras en el día a día, sin llegar a entrar en aspectos técnicos. Además, hace referencia a otros recursos bibliográficos que se usaran más adelante.

DevOps y Agile

Kim, G., Debois, P., Willis, J., Humble, J. y Allspaw, J. (2016). *The DevOps Handbook: How to Create World-class Agility, Reliability, & Security in Technology Organizations* (1. ª ed., parte I). IT Revolution Press.

Los autores ponen el papel de administrador en contexto con la evolución del desarrollo del *software* y cómo las metodologías ágiles han cambiado el paradigma de los sistemas de *software.* Es una lectura entretenida, con poco contenido técnico.

What is DevOps?

Loukides, M. (2012). *What Is DevOps?*. O'Reilly Media.

Este pequeño libro con aspecto de artículo de prensa explica de primera mano, en apenas siete páginas, la reconversión del rol de los administradores, desde los primeros ordenadores de tubos de vacío hasta la actualidad.

Test

1. ¿Qué significa administrar un entorno en la nube?

A. Mantener todos los sistemas funcionando.

B. Crear un nuevo servidor.

C. Asegurar el funcionamiento de todos los sistemas, además de vigilar el coste del entorno.

D. Comprobar que la compañía no dispone de ningún recurso físico en propiedad.

2. ¿Cuál de los siguientes enunciados es correcto?

A. Los proveedores son los responsables de asegurar el acceso físico al *hardware* y, por tanto, los asuntos relativos a la seguridad recaen en ellos.

B. La organización debe proteger la red y sus recursos, y sus equipos deben estar familiarizados con el modelo de seguridad de su proveedor de nube.

C. Los proveedores no ofrecen opciones adecuadas para restringir los accesos y permisos de usuarios y la organización tiene esta responsabilidad.

D. Los DevOps no necesitan preocuparse por la seguridad en la nube, ya que es una tarea de los administradores de seguridad.

3. Relaciona:

Migración	1	A	Porcentaje del tiempo en el que una aplicación ofrece un servicio
Disponibilidad	2	B	Trasladar aplicaciones de un entorno tradicional a un entorno de nube
Entorno híbrido	3	C	Instancia de sistema operativo que sirve un rol concreto para una aplicación específica
Servidor	4	D	Modelo de despliegue donde parte de los recursos residen en las oficinas de la compañía y parte en la nube

4. Al hablar de entornos, un ingeniero DevOps debe:

A. Saber cómo configurar recursos de computación para establecer y mantener una conexión entre los diferentes entornos.

B. Colaborar con su equipo dentro de la organización, pero no es necesario que los recursos de red sean automatizados y, por tanto, un ingeniero DevOps no necesariamente debe estar familiarizado con ellos y sus protocolos.

C. Combinar sus conocimientos de desarrollo y administración con conocimientos de red, para establecer las conexiones adecuadas.

D. Ninguna de las anteriores.

5. ¿Cuál de los siguientes enunciados es correcto al hablar de disponibilidad?

A. Se espera que un DevOps trabaje 24/7 para garantizar el tiempo de actividad.

B. Las aplicaciones de escalado funcionan independientemente de la automatización.

C. Es necesario asegurar que ningún servidor se apaga nunca, incluso actualizando a mano las aplicaciones y paquetes del sistema operativo.

D. Hoy por hoy existen métricas para detectar eventos que pueden activar tareas de autoescalado y notificaciones, como mensajes de correo electrónico y mensajes de texto.

6. ¿Cuál de las siguientes afirmaciones es correcta?

A. Permitir o bloquear el acceso a la red ahora resulta mucho más sencillo, gracias a la nube.

B. La automatización siempre facilita la eliminación de errores y la restricción de accesos a los elementos que realmente son necesarios.

C. Los costes de *hardware* han reemplazado al pago por uso y licencias.

D. Los DevOps necesitan aprender nuevas herramientas, exclusivas de la nube, que no estaban presentes en entornos tradicionales.

7. ¿Qué sistema operativo es mejor: Linux o Windows?

A. Depende de la tarea que vaya a cumplir.

B. Linux, ya que es gratis.

C. Windows, ya que el sistema de ventanas es mucho más fácil de usar.

D. Linux, ya que es lo que usan los informáticos más avanzados.

8. ¿Cuál es la mejor manera de afrontar una migración de una aplicación de un despliegue tradicional *onsite* a un despliegue en la nube?

A. Cambiar el modelo de despliegue para usar herramientas diferentes, como contenedores o grupos de autodespliegue.

B. Hacer una copia de seguridad de la aplicación, copiarla a la nube y recrear la aplicación a partir de la copia.

C. Crear una réplica de los discos duros en la nube que esté sincronizada con cada escritura en los discos *onsite*. Apagar la aplicación *onsite* y arrancarla con los discos en la nube.

D. Todas las opciones son válidas, la mejor depende del tipo de aplicación, la capacidad de tener caída de servicio y el tipo y cantidad de recursos disponibles.

9. En el modelo mascotas *vs.* ganado (*pets vs. cattle*), ¿qué es cada concepto?

A. *Pet* se refiere a un servidor que nunca se apaga, que se actualiza con parches de seguridad y cuyo estado es muy difícil de recuperar si el servidor deja de ser accesible. *Cattle* se refiere a servidores de usar y tirar: si un servidor deja de funciona, se cambia por otro; una actualización se aplica cambiando un servidor por otro.

B. Un servidor «mascota» tiene un nombre familiar y un servidor «ganado» tiene un nombre genérico.

C. Un servidor «mascota» es siempre físico y un servidor «ganado» es siempre virtual.

D. Un servidor «mascota» es siempre Windows y un servidor «ganado» es siempre Linux.

10. ¿Qué se entiende por entorno híbrido?

A. Aquel en el que todos los recursos están en la nube.

B. Aquel en el que todos los recursos son físicos.

C. Aquel en el que hay recursos físicos y virtuales.

D. Aquel en el que parte de los recursos están en las ubicaciones de la compañía y parte residen en la nube.