

Administración de Sistemas en la Cloud

Instalación y administración en *cloud*

Índice

Esquema	3
Ideas clave	4
10.1. Introducción y objetivos	4
10.2. Despliegue de aplicaciones en la nube	4
10.3. Sysprep	9
10.4. Creación de imágenes en AWS	14
10.5. Creación de imágenes Windows en Azure	18
10.6. Referencias bibliográficas	21
A fondo	23
Test	24

INSTALACIÓN Y ADMINISTRACIÓN EN CLOUD

Despliegue de aplicaciones en la nube

- ▲ Despliegue tradicional: actualización incremental, servidores «mascota».
- ▲ Despliegue basado en instancias: imagen base e instalación de la aplicación de cero.
- ▲ Despliegue basado en imágenes: nueva imagen en cada versión.

Sysprep

- ▲ Herramienta de generalización de servidores Windows.
- ▲ Elimina elementos específicos para poder reusar una instalación de sistema operativo.

Esquema

10.1. Introducción y objetivos

Los comandos y herramientas de administración y automatización no son, salvo en algunos casos, específicos de entornos de nube. Al fin y al cabo, **los sistemas operativos no han cambiado drásticamente** por el hecho de estar disponibles en proveedores de nube. Los administradores pueden, por tanto, seguir trabajando con los sistemas con muchas de las herramientas a las que están acostumbrados.

El apartado de despliegue de servidores, sin embargo, ha cambiado mucho respecto a la instalación de servidores físicos. Las ideas generales siguen en pie (actualización paulatina de aplicaciones, generalización de servidores, etc.), pero las **herramientas se han adaptado** a las características de la nube para acelerar los despliegues.

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

- ▶ Entender los diferentes enfoques de actualización de aplicaciones en entornos de nube.
- ▶ Conocer el concepto de generalización aplicado a imágenes.
- ▶ Tomar contacto con la creación de imágenes en AWS y Azure.

10.2. Despliegue de aplicaciones en la nube

Los pasos para actualizar una aplicación propia en ejecución se podrían generalizar de la siguiente manera:

- ▶ Un desarrollador escribe un código para implementar una nueva característica.

- ▶ Los archivos modificados se incorporan a un sistema de control de versiones tras pasar las pruebas.
- ▶ Dependiendo del lenguaje de programación que se utilice, puede ser necesario compilar los archivos fuente para producir un binario ejecutable.
- ▶ Los archivos fuente modificados (o los ejecutables compilados) están disponibles para las instancias en ejecución. Las instancias pueden extraer los archivos directamente del sistema de control de versiones o, tal vez, usar un repositorio local para paquetes de APT, Yum, Ruby, Python u otro tipo de gestor de paquetes.
- ▶ Una vez los archivos modificados están en las instancias, los servicios en ejecución se reinician para empezar a usar el nuevo código o los nuevos archivos de configuración.

Esto es una descripción general de alto nivel y algunas aplicaciones pueden requerir pasos adicionales.

En un entorno tradicional, con servidores físicos, las tareas de desplegar servidores y desplegar o actualizar aplicaciones eran acciones claramente separadas: un servidor con un rol concreto se desplegaba una vez y la aplicación se actualizaba tantas veces como fuera necesario.

En un entorno virtualizado, ya sea de nube pública o privada, hay **dos enfoques** habituales para estas actualizaciones (Lucifredi y Ryan, 2018):

- ▶ Despliegue basado en instancias: cada instancia se actualiza individualmente en el momento del despliegue de la aplicación.
- ▶ Despliegue basado en imágenes: se crea una nueva imagen cada vez que se lanza una versión de producción, a partir de la cual se crean nuevas instancias y se destruyen las antiguas.

Por supuesto, el proceso tradicional se puede replicar en un entorno de nube, pero estos dos enfoques son los que tienen sentido en un entorno DevOps.

Despliegue basado en instancias

En este modelo de despliegue, todos los roles de servidor se basan en una única **imagen maestra** (o unas pocas, en caso de necesitar diferentes sistemas operativos o distribuciones para diferentes roles). La configuración del rol ocurre durante el arranque de la instancia mediante la ejecución de un *script* a partir de algún mecanismo concreto: *user data* en AWS, extensiones de [máquina virtual](#) de Azure o herramientas con Ansible o Puppet.

Una posible actualización de una aplicación de la versión 1.0 a la versión 1.1 con este modelo seguiría estos pasos:

- ▶ Una vez completados los pasos generales, los binarios o los paquetes de actualización, se etiquetan con el número de versión 1.1.
- ▶ Se modifican los *scripts* de despliegue para hacer referencia a los paquetes de la versión 1.1.
- ▶ Todas las instancias desplegadas a partir de este momento ejecutarían los *scripts* para instalar la versión 1.1, pero las instancias ya desplegadas siguen en la versión 1.0. En un grupo de autoescalado, en el que pueden aparecer instancias nuevas en cualquier momento, se podría llegar a una situación con instancias en ambas versiones simultáneamente.
- ▶ Idealmente, las instancias en la versión 1.0 se destruyen y son sustituidas por instancias nuevas, que tendrán la versión 1.1. Esta sustitución ocurre paulatinamente en un *rolling update*. Para que una sustitución paulatina pueda ocurrir, ambas versiones deben poder convivir, incluso aunque sea momentáneamente. De lo contrario, pueden aparecer problemas con los esquemas de las bases de datos u otras configuraciones. Si no pueden convivir, es probable que sea necesaria una actualización en bloque en la que habrá un intervalo sin servicio.

En ambos casos, es habitual que las instancias estén detrás de un balanceador de carga que solo dirige tráfico a las instancias que pasan una llamada de prueba o *healthcheck* (que puede ser, por ejemplo, una petición REST a una ruta diseñada expresamente para validar que la instancia funciona bien). No todas las instancias tienen por qué alojar un servidor web: un servicio que se alimente de una cola de mensajes puede evitar la llamada de *healthcheck*, ya que el propio servicio no recogerá mensajes hasta que no esté listo para hacerlo.

Se podría pensar que se podrían aplicar los *scripts* de la versión 1.1 en las instancias con la versión 1.0 para evitar un despliegue completo. Este enfoque es realmente más parecido al tradicional, en el que se aplican parches sobre un servidor indefinidamente.

Despliegue basado en imágenes

La creación de nuevas imágenes proporciona una forma más limpia de ejecutar actualizaciones. Estas imágenes, o plantillas, consisten en una máquina virtual con una configuración y unas aplicaciones concretas, que se apaga y se «congela», en el sentido de que no puede ser arrancada. Esta imagen mantiene la configuración del *hardware* virtual y los discos de la máquina sin modificaciones desde el apagado. Las instancias se despliegan «clonando» esta imagen, es decir, creando una nueva máquina virtual con el mismo *hardware* virtual y con un disco que es una copia del disco de la imagen.

Los **pasos** para una actualización de 1.0 a 1.1 son los siguientes:

- ▶ Al igual que con el enfoque anterior, una vez completados los pasos generales, se generan los paquetes de instalación con el número de versión 1.1.
- ▶ Se arranca una instancia con una instalación básica del sistema operativo, que podría hacerse con los últimos parches de seguridad. Este sistema operativo suele

ser el mismo que el de la imagen de la versión 1.0, pero no es estrictamente necesario.

- ▶ Se instala la aplicación, versión 1.1, desde cero. Los *scripts* de instalación serán parecidos a los del modelo basado en instancias, pero no tienen que tratar con actualizaciones de la aplicación, solo con la instalación inicial, por lo que pueden ser más sencillos.
- ▶ Una vez completada la instalación, se «generaliza» la imagen y se apaga la máquina virtual (en algunos entornos este proceso es automático).
- ▶ Dependiendo del entorno de nube, la imagen se genera a partir de la máquina virtual, dejando la máquina intacta, o la propia máquina virtual se convierte en una plantilla. En ambos casos, la imagen es inmutable y no se puede arrancar, salvo mediante un clonado.
- ▶ Las instancias en ejecución, basadas en la imagen de la versión 1.0, se sustituyen por instancias nuevas con la versión 1.1. Al igual que en el modelo anterior, se puede seguir un proceso de *rolling update* o una actualización en bloque con parada del servicio. La problemática de convivencia de varias versiones existe también en este modelo de despliegue.

La actualización basada en imágenes es más estable, porque no hay actualizaciones incrementales, sino instalaciones de cero. Como contrapartida, generar una imagen nueva, aun cuando el proceso esté automatizado, lleva más tiempo que cambiar un *script* de despliegue. En cualquier caso, el despliegue basado en imágenes puede seguir haciendo uso de *scripts* de arranque (es decir, *user data*, extensiones, etc.) para finalizar la configuración de la aplicación, en caso de que sea necesario.

Inmutabilidad de la aplicación

El despliegue tradicional sufre de un hecho conocido como **derivado de la configuración**. Esta situación ocurre en el escenario de servidores «mascota», en el que los administradores tratan con cariño de mantener los servidores con vida a toda costa. Cualquier diferencia en la aplicación de una configuración puede dar lugar a horas de investigación para resolver un problema. Cuando los administradores trabajan con servidores «ganado», todas las instancias están en un estado conocido, ya que todas parten de un mismo punto de partida, sea un *script* de instalación o una imagen maestra. El corolario es que, una vez que se despliega un servidor, nunca debe modificarse, solo debe reemplazarse. Esto se conoce como el patrón de diseño del servidor inmutable.

La inmutabilidad de la aplicación se logra automatizando las operaciones realizadas en las instancias. Estas operaciones se limitan, exclusivamente, al aprovisionamiento, reemplazo y desaprovisionamiento. Si se consigue implementar este patrón por completo, la aplicación se puede desplegar de manera predecible, los cambios se pueden revertir con la misma facilidad y su estado completo, incluida la configuración de seguridad, se entiende claramente y se puede reproducir de manera exacta en cualquier punto del ciclo de vida de la instancia.

10.3. Sysprep

Una de las principales ventajas de los entornos de nube es la capacidad de desplegar nuevos servidores rápidamente. Si bien se podría seguir el procedimiento tradicional de crear una máquina virtual, conectar una imagen ISO, instalar el sistema operativo, instalar las aplicaciones, etc., las nubes privadas y públicas ofrecen funcionalidades que aceleran este proceso. El *user data* de AWS es un ejemplo, como lo son las imágenes personalizadas.

Windows está preparado para estas tecnologías. Este capítulo trata sobre una herramienta que facilita la automatización del despliegue en cualquier entorno de nube pública o privada. Está integrada en todas las versiones de Windows, por lo que puede llevar a cabo este mismo proceso en cualquier máquina Windows actual, ya sea un cliente o un servidor.

Sysprep (Krause, 2019) es una herramienta que prepara el sistema para el clonado, aplicando un proceso llamado **generalización**. Su nombre oficial es Herramienta de preparación del sistema de Microsoft. En resumen, permite crear una imagen maestra de un servidor que puede reutilizarse tantas veces como sea necesario para desplegar servidores adicionales.

Una **ventaja** clave del uso de Sysprep es que es posible añadir configuraciones personalizadas en el servidor maestro e instalar elementos, como las actualizaciones de Windows, y todas estas configuraciones y parches existirán en la imagen maestra. El uso de Sysprep ahorra tiempo, al no tener que recorrer el proceso de instalación del sistema operativo, pero ahorra aún más al no tener que esperar a que las actualizaciones de Windows desplieguen todos los parches actuales en cada nuevo sistema.

¿Es Sysprep necesario? Se podría pensar que, para clonar un servidor maestro, simplemente se podría usar una herramienta de clonado de disco duro o, si se tratara de máquinas virtuales, simplemente se podría copiar y pegar el fichero de disco virtual para hacer una copia de su nuevo servidor. Esto puede funcionar sobre el papel, pero el principal problema es que la nueva imagen sería una réplica exacta de la original: el nombre del *host* sería el mismo y, lo que es más importante, cierta información de identificación central en Windows, como el número de identificador de seguridad (SID) del sistema operativo, sería exactamente el mismo. Si activara tanto el servidor maestro original como un servidor nuevo basado en esta réplica exacta, aparecerían conflictos y colisiones en la red, ya que estos dos servidores luchan por su derecho a ser el único servidor con ese nombre único y SID.

Este problema es especialmente relevante en los entornos de Directorio Activo, donde es aún más importante que cada equipo miembro del dominio red tenga un SID único. Sysprep corrige todos estos problemas inherentes al proceso de duplicación del sistema al generar identificadores únicos aleatorios durante el primer arranque de los servidores clonados.

Las **fases** para preparar una imagen maestra con Sysprep son las siguientes:

- ▶ Instalación del sistema operativo en un nuevo servidor.
- ▶ Configuración y actualización.
- ▶ Ejecución de Sysprep y apagado del servidor.
- ▶ Creación de imagen maestra del disco duro.
- ▶ Clonado del disco para desplegar nuevos servidores.

Instalación, configuración y actualización

Para la instalación y configuración del equipo, se puede seguir el proceso habitual, pero en general no es recomendable instalar roles de Windows, porque, en función del rol, el proceso de Sysprep puede causar problema con la configuración. Esto se debe a que una instalación de roles realiza numerosos cambios en el sistema operativo y algunos de los roles bloquean el nombre de *host* particular del servidor maestro, aun cuando el nombre será diferente en el clon. Las personalizaciones que consisten en crear archivos y carpetas, arrancar o detener servicios o cambiar configuraciones del registro suelen ser seguras. La instalación de parches de seguridad es fundamental, ya que es una de las fases que más tiempo consume tras el despliegue.

Ejecutar Sysprep

Ahora que el servidor maestro está preparado, es hora de ejecutar Sysprep. El ejecutable se encuentra en `C:\Windows\System32\Sysprep` (ver Figura 1).

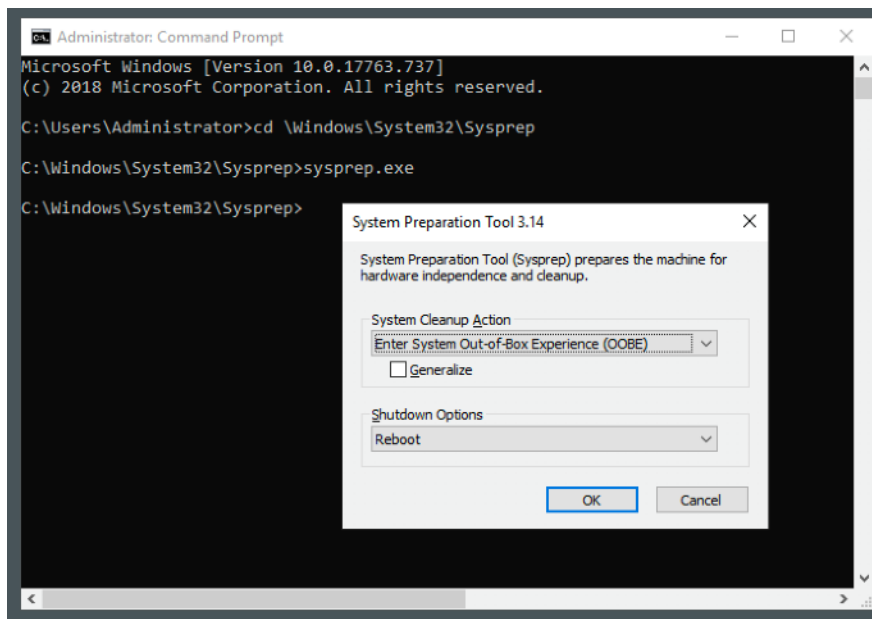


Figura 1. Ejecución gráfica de Sysprep. Fuente: elaboración propia.

Sysprep también admite modificadores de línea de comandos para configurar las mismas opciones que el cuadro de diálogo gráfico:

- ▶ `/generalize`: Sysprep eliminará toda la información única del sistema (SID) de la instalación de Windows, haciendo que la imagen final sea utilizable en múltiples máquinas, ya que cada nuevo servidor creado a partir de la imagen obtendrá un SID nuevo único.
- ▶ `/audit`: esto reinicia la máquina en un modo de auditoría especial, donde es posible agregar controladores adicionales a Windows antes de completar el proceso.
- ▶ `/oobe`: con este modificador activado, el servidor iniciará el asistente de mini configuración (el *out-of-box-experience*) la próxima vez que arranque Windows, es decir, en el primer arranque de cada servidor clonado.
- ▶ `/quiet`: que se ejecute Sysprep sin mensajes de estado en la pantalla.
- ▶ `/shutdown`: esto apaga el sistema cuando Sysprep finaliza.

Los dos que son más importantes para el propósito de crear una imagen maestra son los modificadores `/generalize` y `/shutdown`. El primero es fundamental, porque reemplaza toda la información de identificación única, la información SID, en las

nuevas copias de Windows de los servidores clonados. El modificador de apagado también es muy importante, ya que, en caso de un reinicio, el servidor generará un nuevo SID y habría que ejecutar Sysprep de nuevo. De no usarlo siempre, es posible apagar el servidor manualmente.

Para automatizar el arranque de los nuevos servidores lo más posible, el comando de Sysprep sería el siguiente:

```
sysprep.exe /generalize /shutdown
```

Después de iniciar este comando, Sysprep mostrará algunos mensajes y, finalmente, apagará el servidor.

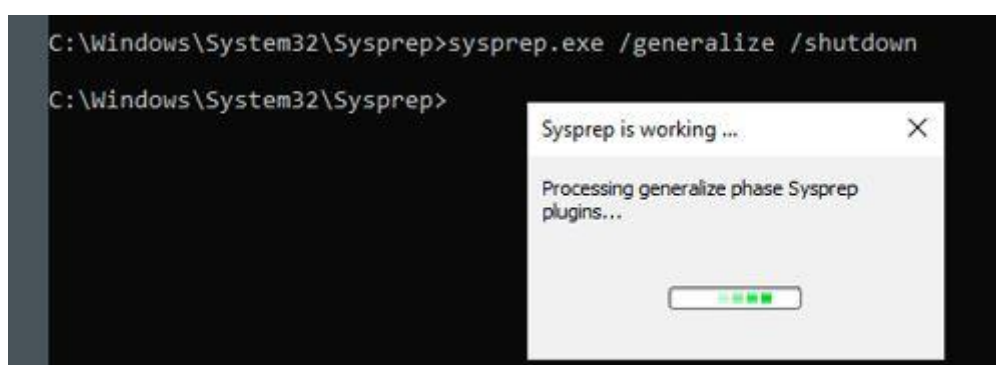


Figura 2. Sysprep preparando Windows. Fuente: elaboración propia.

Creación de la imagen maestra

Una vez el servidor está apagado, ha llegado el momento de crear la imagen maestra. En los servidores físicos tradicionales, había que arrancar desde un CD con herramientas que clonaban el contenido del disco a una unidad de red o a un disco duro externo. En un entorno virtual, la situación es más sencilla: basta con copiar el archivo del disco duro (por ejemplo, el fichero `.vmtx` en vSphere o `.vhdx` en Hyper-V) o, lo que es más habitual, crear una plantilla a partir de la máquina virtual.

Las plantillas son similares a las máquinas virtuales, en cuanto a que consisten en una configuración de *hardware* virtual y uno o varios archivos de disco. Pero, en vez de

ofrecer la opción de arranque, el sistema de virtualización solo permite clonar plantillas en una máquina virtual nueva. De esta manera, los discos de la plantilla no se ven afectados por un arranque accidental que podría echar a perder el sellado del Sysprep.

Sysprep en la nube

Algunos proveedores de nube integran Sysprep en sus herramientas de creación de imágenes. Por ejemplo, [EC2Launch](#) o [EC2 Image Builder](#) en AWS son capaces de ejecutar Sysprep automáticamente. VMware Horizon soporta Sysprep, además de [QuickPrep](#), una variante propia que no aplica los mismos cambios que Sysprep, pero es útil en el despliegue de sistemas operativos de escritorio.

El cualquier caso, Sysprep es una herramienta general y, como tal, puede ejecutarse independientemente del proveedor de nube, incluso aunque este soporte Sysprep de manera integrada. Tanto [Azure](#) como AWS disponen de documentación sobre Sysprep que, a grandes rasgos, sigue el procedimiento descrito en este capítulo.

10.4. Creación de imágenes en AWS

Este capítulo muestra paso a paso cómo crear una [AMI](#), o Amazon *machine image*, a partir de una máquina virtual Linux en ejecución. El proceso en instancias Windows es algo más complejo, ya que requiere un proceso de generalización y no soporta Cloud-init directamente. En la sección A fondo hay un guía para crear imágenes Windows en AWS con EC2Launch.

A continuación, en el vídeo, *Despliegue de sistemas operativos en la nube*, se verán demostrados los pasos explicados a lo largo del tema.



El primer paso es arrancar la imagen con la configuración necesaria. El tamaño, las opciones de redes y las etiquetas no formarán parte de la nueva AMI. La AMI inicial (ver Figura 3) sí que es relevante, ya que la nueva AMI será un incremento sobre la AMI base. El almacenamiento también es importante, ya que los discos EBS formarán parte de la AMI.

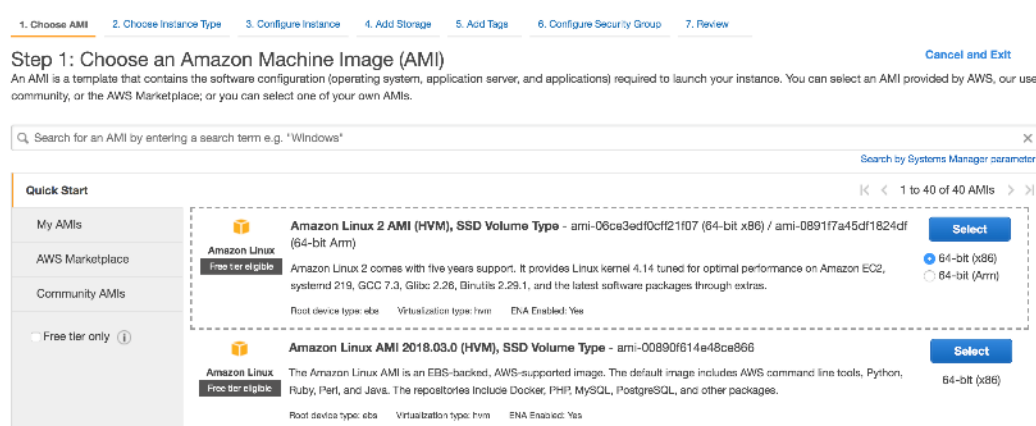


Figura 3. Selección de AMI base. Fuente: elaboración propia.

La instalación de aplicaciones y configuración es el siguiente paso. Se pueden crear usuarios por defecto, crear carpetas, añadir repositorios, etc. (ver Figura 4). En este paso, **no** es recomendable aplicar configuraciones dependientes de la instancia, como el nombre de *host* o configuraciones dependientes de la IP (por ejemplo, configurar un servidor para que escuche en una IP concreta), ya que esos valores cambiarán cuando arranquen las instancias basadas en esta imagen.

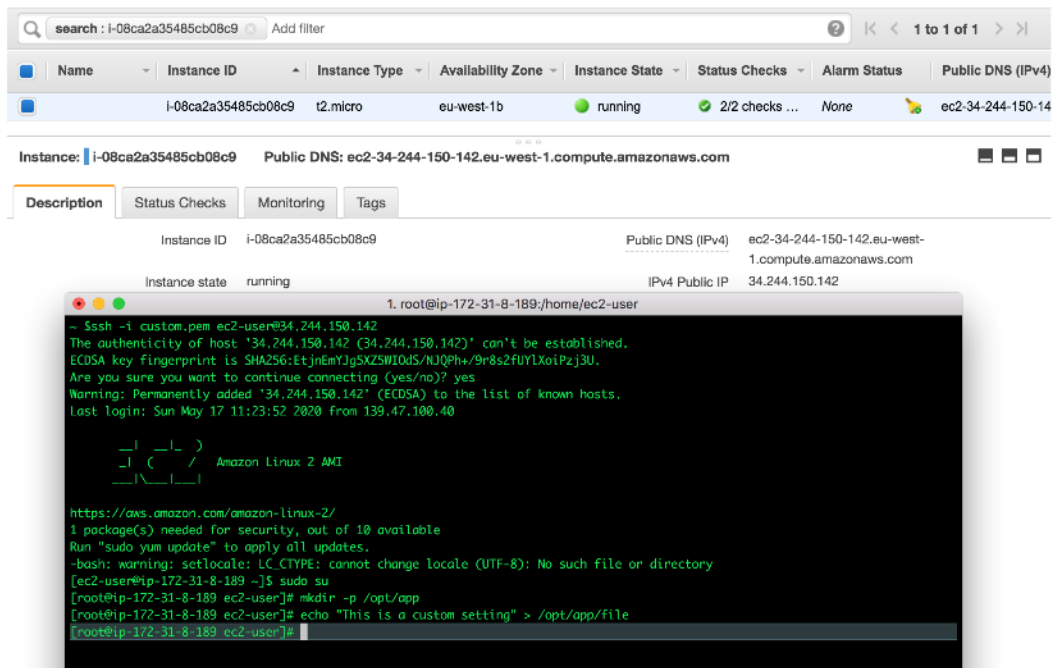


Figura 4. Conexión a la instancia, instalación y configuración de aplicaciones. Fuente: elaboración propia.

Cuando la instancia está lista, el siguiente paso es crear la imagen en el menú Actions > Image > Create Image (Figura 5). En el siguiente cuadro de diálogo, no hay más que introducir un nombre de imagen y una descripción. También permite añadir discos EBS adicionales, que estarán disponibles en todas las instancias arrancadas a partir de esta AMI.

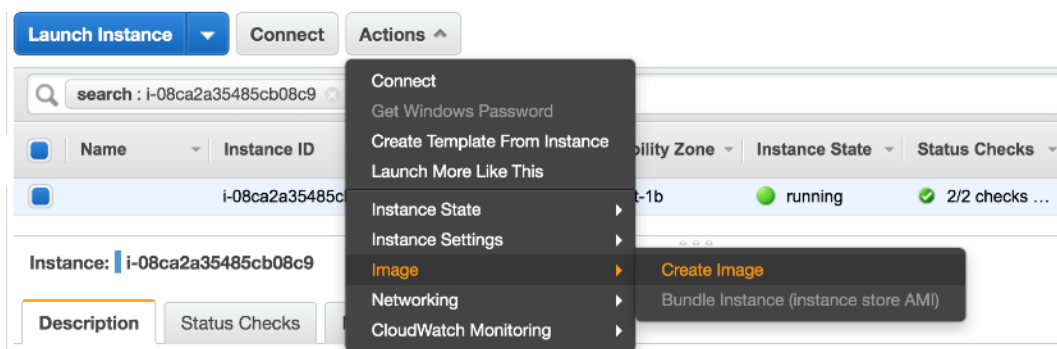


Figura 5. Menú de creación de instancia. Fuente: elaboración propia.

Como resultado de este proceso, aparecerán dos recursos: la propia AMI (ver Figura 6) y una captura de disco o *snapshot* (Figura 7). Esta captura de disco es equivalente a las imágenes de disco de servidores físicos o a los archivos de discos virtuales de otros entornos de virtualización.

Owned by me

	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
	custom-ami	ami-05c5d35c7f6b1c528	014641795089/c...	014641795089	Private	pending	May 17, 2020 at 1:26	

Image: ami-05c5d35c7f6b1c528

Details

Permissions

Tags

AMI ID	ami-05c5d35c7f6b1c528	AMI Name	custom-ami
Owner	014641795089	Source	014641795089/custom-ami
Status	pending	State Reason	-
Creation date	May 17, 2020 at 1:26:33 PM UTC+2	Platform details	Linux/UNIX
Architecture	x86_64	Usage operation	RunInstances

Edit

Figura 6. Listado de AMI privadas. Fuente: elaboración propia.

Create Snapshot

Actions

Owned By Me

Filter by tags and attributes or search by keyword

1 to 2 of 2

<input type="checkbox"/>	Name	Snapshot ID	Size	Description	Status	Started
<input type="checkbox"/>		snap-08bebddc2b4...	8 GiB	Created by CreateImage(i-074e7041194cd1393) for ami-00706...	completed	May 17, 2020
<input type="checkbox"/>		snap-0f43f1434889...	8 GiB	Created by CreateImage(i-08ca2a35485cb08c9) for ami-05c5d...	completed	May 17, 2020

Figura 7. Listado de capturas de disco (*snapshots*). Fuente: elaboración propia.

La nueva AMI estará disponible en la pestaña My AMIs en el asistente de despliegue de la nueva instancia. Si el despliegue se automatiza a través de un SDK o de la herramienta de línea de comandos `awscli`, solo es necesario usar el ID de la AMI, sin especificar si es pública o privada.

Step 1: Choose an Amazon Machine Image (AMI) [Cancel and Exit](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Search by Systems Manager parameter

Quick Start

My AMIs

AWS Marketplace

Community AMIs

custom-ami - ami-05c5d35c7f6b1c528

Custom Amazon Linux 2

Root device type: ebs Virtualization type: hvm Owner: 014641795089 ENA Enabled: Yes

Select

64-bit (x86)

Figura 8. AMI privada disponible para desplegar una instancia nueva. Fuente: elaboración propia.

```
~ $ssh -i custom.pem ec2-user@34.253.184.136
The authenticity of host '34.253.184.136 (34.253.184.136)' can't be established.
ECDSA key fingerprint is SHA256:EtjnEmYJg5XZ5WIOdS/NJQPh+/9r8s2FUYlXoiPzj3U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '34.253.184.136' (ECDSA) to the list of known hosts.
Last login: Sun May 17 11:25:09 2020 from 139.47.100.40

  _I_  _I_  )
  _I_  C   /   Amazon Linux 2 AMI
  __I\__I__I

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-12-170 ~]$ cat /opt/app/file
This is a custom setting
[ec2-user@ip-172-31-12-170 ~]$
```

Figura 9. Opciones personalizadas en la nueva instancia. Fuente: elaboración propia.

En AWS, las AMI son un recurso específico de una región. Es decir, esta AMI se podrá desplegar en la misma región en la que se creó. Para poder desplegarla en otras [regiones](#), es necesario copiarla primero.

10.5. Creación de imágenes Windows en Azure

Este capítulo muestra la creación de una imagen [Windows en Azure](#). En este caso, el proceso sí incluye el proceso de generalización.

Al igual que en AWS, el primer paso es desplegar la máquina virtual (Figura 10). De nuevo, las opciones como configuración de red, tamaño y etiquetas no son relevantes, ya que se podrán decidir en las futuras imágenes. La imagen inicial sí afectará a la imagen nueva.

Instance details

Virtual machine name * ⓘ ✓

Region * ⓘ ✓

Availability options ⓘ ✓

Image * ⓘ ✓
[Browse all public and private images](#)

Azure Spot instance ⓘ ☐ Yes ☒ No

Size * ⓘ **Standard B2s**
 2 vcpus, 4 GiB memory (€36.94/month)
[Change size](#)

Administrator account

Username * ⓘ ✓

Password * ⓘ ✓

Confirm password * ⓘ ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ ☐ None ☒ Allow selected ports

Select inbound ports * ✓

Figura 10. Creación de VM inicial. Fuente: elaboración propia.

Tras instalar las aplicaciones y aplicar las configuraciones necesarias, es necesario ejecutar Sysprep siguiendo los pasos indicados en el apartado *Despliegue de aplicaciones en la nube*.

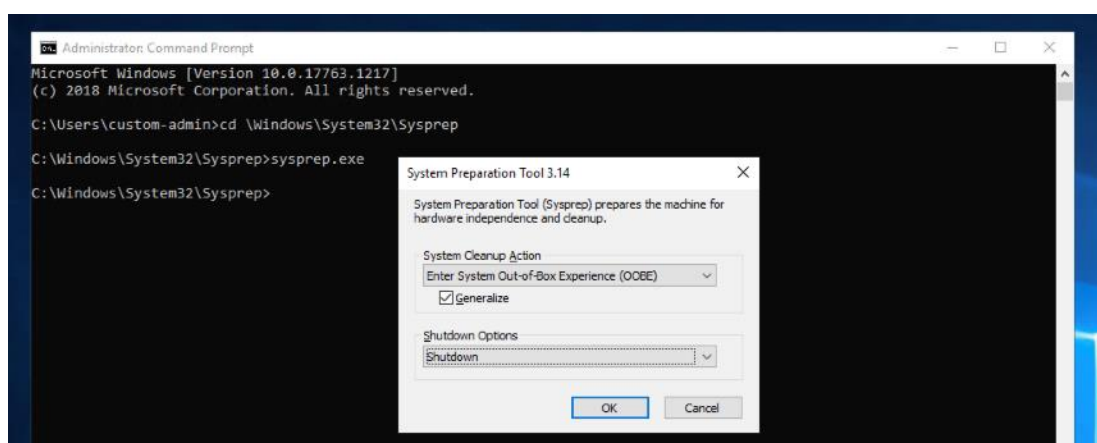


Figura 11. Generalización con Sysprep. Fuente: elaboración propia.

La ejecución de Sysprep debe terminar con un apagado de la VM. Una vez apagada, hay que pulsar el botón de Capture (Figura 12).

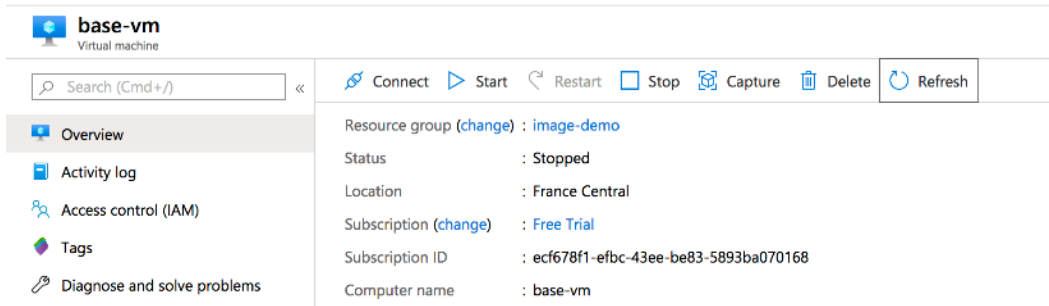


Figura 12. Botón Capture en la VM apagada. Fuente: elaboración propia.

A diferencia de AWS, donde la instancia convertida en AMI sigue funcionando normalmente, la creación de la imagen en Azure destruye la VM original, tal como indica el cuadro de diálogo de la Figura 13.

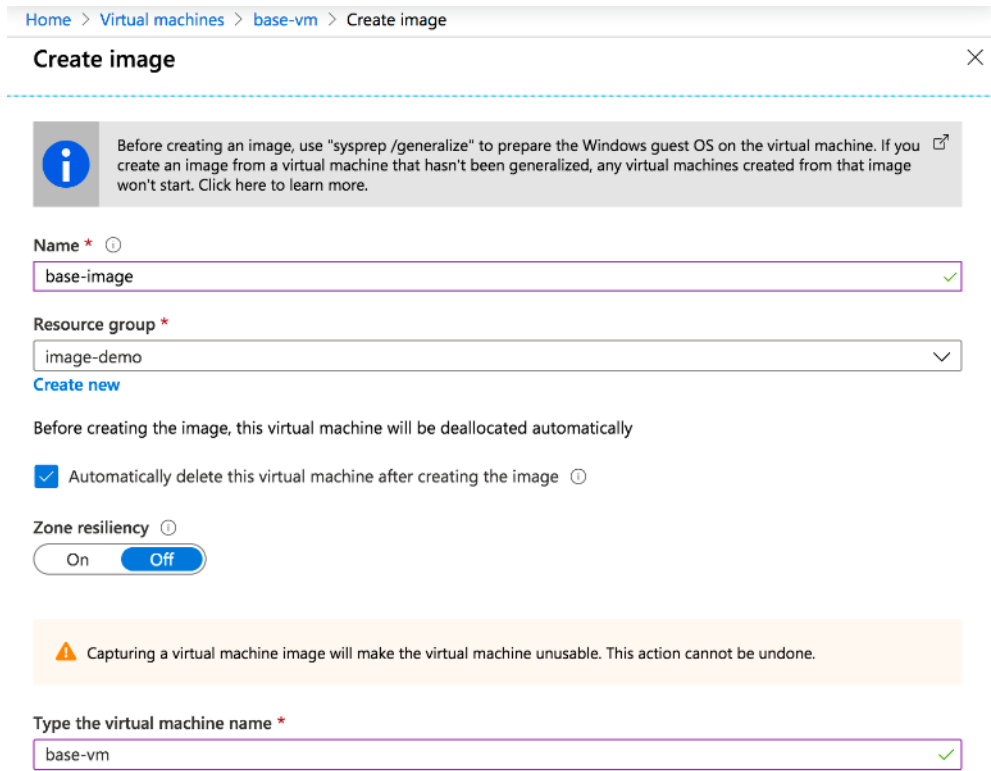


Figura 13. Menú de creación de imagen a partir de la VM. Fuente: elaboración propia.

Una vez completado el proceso, la imagen estará disponible para su despliegue como VM. El botón Create VM de la Figura 14 iniciará el asistente de despliegue con esta imagen ya seleccionada.

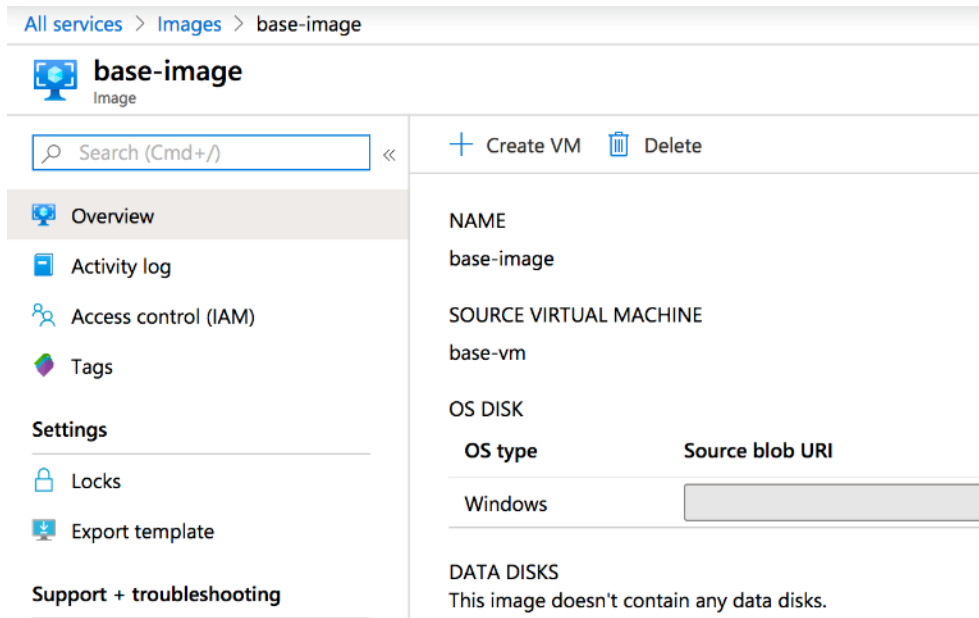


Figura 14. Imagen lista para ser desplegada. Fuente: elaboración propia.

10.6. Referencias bibliográficas

AWS. (s. f.). Configure EC2Launch. En *Configure a Windows instance using EC2Launch*. <https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2launch.html#ec2launch-config>

AWS. (s. f.). *Copy an AMI*. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html>

AWS. (s. f.). *Create an Amazon EBS-backed Linux AMI*. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html>

AWS. (s. f.). *How EC2 Image Builder works*.

<https://docs.aws.amazon.com/imagebuilder/latest/userguide/how-image-builder-works.html>

Krause, J. (2019). *Mastering Windows Server 2019: The Complete Guide for IT Professionals to Install and Manage Windows Server 2019 and Deploy New Capabilities* (2.ª ed.). Packt Publishing.

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

Microsoft. (2018, marzo 30). *Virtual machine extensions and features for Windows*.

<https://docs.microsoft.com/en-us/azure/virtual-machines/extensions/features-windows>

Microsoft. (2018, septiembre 27). *Create a managed image of a generalized VM in Azure*.

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/capture-image-resource>

VMware. (2014, mayo 30). *Choosing QuickPrep or Sysprep to Customize Linked-Clone Machines*.

<https://docs.vmware.com/en/VMware-Horizon-7/7.3/horizon-virtual-desktops/GUID-A9C86030-D3E3-4266-9581-E3EAFB8D4BD2.html>

Creación de imágenes Windows en AWS

AWS. (s. f.). *Create a custom Windows AMI.*

https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/Creating_EBSbacked_WinAMI.html

La creación de AMI Windows es más compleja que para imágenes Linux. La guía de la documentación de AWS explica el proceso con las diferentes opciones de generalización, en función de la versión de Windows Server.

Automatización del arranque con ficheros de respuesta

Microsoft. (2017, mayo 2). *Use Answer Files with Sysprep.*

<https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/use-answer-files-with-sysprep>

Sysprep no se limita a la generalización del sistema operativo y puede automatizar totalmente el primer arranque de la máquina virtual con los llamados ficheros de respuesta. La documentación enlazada contiene más información al respecto.

1. ¿Cómo se logra la inmutabilidad de los servidores?
 - A. Desplegando desde imágenes que ya incluyen la versión concreta de la aplicación.
 - B. Desplegando desde imágenes base, siempre instalando la aplicación de cero.
 - C. A y B son correctas.
 - D. Actualizando la aplicación incrementalmente.

2. ¿Qué parámetro de Sysprep aplica la generalización del sistema operativo?
 - A. /generalize.
 - B. /oobe.
 - C. /shutdown.
 - D. /restart.

3. ¿Cuál de estas opciones no está incluida en la definición de la AMI?
 - A. Tamaño.
 - B. Opciones de red.
 - C. Etiquetas.
 - D. Todas las anteriores.

4. ¿Qué ocurre con la instancia original en AWS cuando se crea una AMI a partir de ella?
 - A. La instancia desaparece y se crea una AMI con el mismo ID que la instancia original.
 - B. Nada, la instancia sigue en ejecución.
 - C. No es posible crear AMI a partir de instancias.
 - D. La instancia se apaga durante la creación y se vuelve a encender al terminar.

5. ¿Qué ocurre con la instancia original en Azure cuando se crea una AMI a partir de ella?
- A. La instancia desaparece y se crea una AMI con el mismo ID que la instancia original.
 - B. Nada, la instancia sigue en ejecución.
 - C. No es posible crear AMI a partir de instancias.
 - D. La instancia desaparece.
6. ¿Soportan los proveedores de nube el uso de Sysprep?
- A. Sí, siempre es posible ejecutar Sysprep en una máquina virtual, sea cual sea el proveedor.
 - B. Sí, además algunos proveedores ofrecen servicios que integran la ejecución de Sysprep.
 - C. Sí, además algunos proveedores ofrecen alternativas propias a Sysprep.
 - D. Todas las anteriores.
7. ¿Por qué en un despliegue basado en imágenes normalmente las instancias están disponibles más rápido que en uno basado en instancias?
- A. En general tardan por igual.
 - B. Porque las aplicaciones están instaladas en la imagen y, al arrancar las instancias, solo es necesario configurar algunos aspectos del sistema operativo y de la propia aplicación.
 - C. Porque las imágenes privadas tardan menos en arrancar que las públicas.
 - D. Ninguna las anteriores.

8. ¿Por qué no se puede seleccionar el sistema operativo al desplegar una instancia a partir de una imagen privada?
- A. Sí se puede, la opción aparece en el siguiente paso.
 - B. Por motivos de licencia.
 - C. Porque el sistema operativo forma parte del disco de la instancia, al contrario que el *hardware* virtual, que sí se puede seleccionar al desplegar la instancia.
 - D. Ninguna las anteriores.
9. ¿Qué es el clonado?
- A. Es un proceso por el cual se crea una imagen exacta del disco duro de una máquina virtual.
 - B. No es un término aplicado a la nube.
 - C. Copiar los archivos del sistema operativo exclusivamente.
 - D. permite duplicar las interfaces de red.
10. ¿Qué es una imagen maestra?
- A. Es una imagen creada para poder desplegar máquinas virtuales a partir de ella.
 - B. No es un término aplicable a la virtualización.
 - C. Es una imagen que permite crear imágenes secundarias.
 - D. Es una imagen protegida.