

# AMI – AMI

Authored by  
**memjavad**

October 24, 2025

## RECOMMENDED CITATION

memjavad (2025). *AMI – AMI*. Spanish Psychological Databases. Retrieved from <https://spanish.arabpsychology.com/?p=1471>

## Amazon Machine Image (AMI)

**Primary Disciplinary Field(s):** Computación en la Nube, Infraestructura como Servicio (IaaS), Amazon Web Services (AWS)

### 1. Definición Central

El **Amazon Machine Image (AMI)**, o Imagen de Máquina de Amazon, constituye uno de los pilares fundacionales del servicio de computación elástica, conocido como [Amazon Elastic Compute Cloud \(EC2\)](#), proporcionado por Amazon Web Services (AWS). En esencia, una AMI es una plantilla preconfigurada que funciona como el plano fundamental para lanzar una instancia de EC2, la cual es una máquina virtual que opera en el entorno de la nube de AWS. Esta plantilla encapsula todos los elementos necesarios para que un servidor operativo se inicie de manera inmediata y consistente, eliminando la necesidad de instalación manual de sistemas operativos y aplicaciones base cada vez que se requiere una nueva instancia de servidor. La funcionalidad primordial de la AMI radica en garantizar la uniformidad y la reproducibilidad de los entornos de servidor, lo cual es vital para arquitecturas que demandan escalabilidad horizontal y despliegues rápidos.

La AMI no es simplemente un archivo de imagen estático, sino un conjunto de metadatos y punteros que definen el sistema operativo (SO), el software de aplicación, las bibliotecas, las configuraciones de usuario, y la estructura de almacenamiento necesaria para la instancia. Cuando un usuario selecciona una AMI para lanzar una instancia, AWS utiliza esta plantilla para aprovisionar el disco raíz (root volume) y configurar la máquina virtual subyacente. Este proceso minimiza drásticamente el tiempo de inicio (boot time) y asegura que cada instancia lanzada a partir de la misma AMI sea idéntica en su configuración inicial, facilitando así las prácticas de [DevOps](#) y la gestión de la configuración a gran escala. La capacidad de empaquetar un entorno de aplicación complejo en una única unidad reutilizable es lo que confiere a la AMI su inmenso valor dentro del ecosistema de la computación en la nube moderna.

Es crucial entender que la AMI actúa como un artefacto inmutable en el contexto de la infraestructura como código. Una vez que la AMI ha sido creada y registrada, no puede ser modificada directamente; cualquier cambio requerido en la configuración debe resultar en la creación de una nueva AMI. Esta inmutabilidad promueve la fiabilidad y simplifica los procesos de reversión, ya que si un nuevo despliegue basado en una AMI actualizada presenta problemas, es sencillo volver a lanzar instancias utilizando la versión anterior y probada. Las AMI son fundamentales para la estrategia de 'baking' o cocción de imágenes, donde las configuraciones y dependencias se integran en la imagen base antes del despliegue, en contraste con el enfoque de 'golden image' o la configuración en tiempo de ejecución.

## 2. Estructura y Componentes Clave

La arquitectura interna de una AMI se compone de varios elementos interconectados que definen cómo se comportará la instancia resultante. El componente más crítico es la instantánea del volumen de almacenamiento (**EBS snapshot**) o, en el caso de las AMI más antiguas basadas en almacenamiento de instancia, la plantilla de almacenamiento. Esta instantánea contiene la imagen del disco raíz, incluyendo el sistema operativo y todo el contenido instalado. Cuando se lanza una instancia, AWS utiliza esta instantánea para crear el volumen de disco raíz de la nueva máquina virtual. La eficiencia de este proceso se debe a que AWS solo carga los bloques de datos necesarios a medida que la instancia los solicita, lo que permite un inicio rápido incluso con imágenes de gran tamaño.

Otro componente esencial es el manifiesto de la AMI, un archivo XML que describe la imagen y sus parámetros de lanzamiento. Este manifiesto incluye información vital como la arquitectura del procesador (por ejemplo, x86\_64 o arm64), el tipo de virtualización requerido (paravirtual o HVM), la ubicación del kernel, y los identificadores de las instantáneas de EBS asociadas. Además, la AMI incluye permisos de lanzamiento, que especifican qué cuentas de AWS tienen autorización para utilizar esa plantilla para iniciar nuevas instancias. Estos permisos pueden ser públicos (accesibles por todos), privados (solo accesibles por la cuenta propietaria), o compartidos con cuentas específicas, lo cual es fundamental para la colaboración segura y la distribución de software.

Finalmente, la AMI está intrínsecamente ligada al concepto de región de AWS. Una AMI es específica de la región en la que fue creada. Si una organización necesita utilizar la misma plantilla en múltiples regiones geográficas (por ejemplo, para redundancia o para servir a clientes en diferentes continentes), debe copiar explícitamente la AMI a cada región deseada. Este requisito de regionalización asegura que los datos y las configuraciones permanezcan cerca de los recursos de cómputo, optimizando la latencia y cumpliendo con las regulaciones de soberanía de datos que a menudo varían por jurisdicción. La gestión eficiente de las AMI a través de múltiples regiones es un desafío clave para las empresas con huellas globales en la nube.

## 3. Tipos y Modelos de AMI

Las AMI se clasifican generalmente en tres categorías principales, determinadas por su origen y propósito, ofreciendo a los usuarios una flexibilidad considerable para el aprovisionamiento de infraestructura. La primera categoría son las **AMI proporcionadas por AWS**, también conocidas como AMI públicas o básicas. Estas son mantenidas y optimizadas directamente por Amazon e incluyen imágenes de sistemas operativos populares y estables, como Amazon Linux (actualmente Amazon Linux 2 y 2023), Windows Server, y versiones base de distribuciones de código abierto como Ubuntu y Red Hat Enterprise Linux. Estas AMI base son seguras, reciben

parches regularmente y están optimizadas para un rendimiento óptimo en el entorno EC2.

La segunda categoría son las **AMI de la Comunidad**, que son imágenes creadas y compartidas por otros usuarios de AWS. Estas pueden contener configuraciones especializadas, software preinstalado o entornos de desarrollo específicos. Si bien ofrecen una gran variedad y ahorran tiempo de configuración inicial, su uso requiere precaución, ya que la responsabilidad de la seguridad y la integridad del software recae completamente en el usuario. AWS advierte que las AMI de la comunidad deben ser inspeccionadas cuidadosamente para evitar la ejecución de código malicioso o la exposición accidental de credenciales. No obstante, son una herramienta poderosa para el intercambio de soluciones estandarizadas dentro de la comunidad de desarrolladores.

La tercera y más utilizada categoría en entornos empresariales son las **AMI Personalizadas (Custom AMIs)**. Estas son creadas por los propios usuarios a partir de una AMI base (de AWS o de la Comunidad) a la que se le añaden capas de personalización. Este proceso implica lanzar una instancia a partir de una AMI base, instalar software específico (servidores web, bases de datos, agentes de monitoreo), realizar configuraciones de seguridad detalladas y, finalmente, crear una nueva AMI a partir de esa instancia modificada. Las AMI personalizadas son el corazón de la estrategia de inmutabilidad de la infraestructura, ya que permiten a las organizaciones estandarizar sus entornos de aplicación, asegurando que el código se ejecute en un entorno predecible y validado.

#### 4. Ciclo de Vida de una AMI

El ciclo de vida de una **Amazon Machine Image** es un proceso estructurado que garantiza la gestión eficiente de las plantillas de infraestructura a lo largo del tiempo. Comienza con la **Creación**, que generalmente se realiza a partir de una instancia EC2 existente que ha sido configurada y probada meticulosamente. Una vez creada la imagen, se produce la fase de **Registro**, donde la AMI se asocia con un ID único dentro de la región de AWS. Este ID (AMI ID) es la referencia que los usuarios utilizan para lanzar nuevas instancias. Durante la fase activa, la AMI se utiliza repetidamente para el **Lanzamiento de Instancias**, sirviendo como la fuente de verdad para el despliegue de servidores.

A medida que el software evoluciona y los requisitos de seguridad cambian, es inevitable que las AMI necesiten ser actualizadas. En lugar de modificar una AMI existente (lo cual no es posible), el flujo de trabajo requiere la creación de una **Nueva Versión**. La organización crea una nueva AMI con parches de seguridad actualizados, versiones de software más recientes y nuevas configuraciones. Esta nueva AMI se somete a pruebas rigurosas antes de ser promovida. Una vez que la nueva AMI está lista para producción, la versión anterior entra en la fase de **Desuso o Deprecación**.

La etapa final es la **Anulación del Registro (Deregistration)**. Anular el registro de una AMI significa que ya no se puede utilizar para lanzar nuevas instancias. Sin embargo, es fundamental notar que la anulación del registro de la AMI no elimina automáticamente las instantáneas de EBS subyacentes. Para evitar cargos de almacenamiento innecesarios y mantener la higiene de la infraestructura, el administrador debe eliminar explícitamente las instantáneas de EBS asociadas después de anular el registro de la AMI. Una política de gestión de ciclo de vida bien definida es vital para controlar los costos y garantizar que solo se utilicen imágenes aprobadas y seguras en el entorno de producción.

## 5. Creación y Personalización

La personalización y creación de AMI es un proceso que se ha automatizado significativamente en el ecosistema de AWS para soportar la infraestructura como código. Tradicionalmente, la creación de una AMI personalizada implicaba lanzar manualmente una instancia base, conectarse a ella (vía SSH o RDP), instalar software, aplicar configuraciones, apagarla y luego usar la consola de AWS o la API para crear la imagen. Este proceso manual, sin embargo, es propenso a errores y no es escalable.

Para estandarizar y automatizar esta tarea, herramientas de terceros como **HashiCorp Packer** han ganado popularidad. Packer permite a los desarrolladores definir la configuración de la AMI en un archivo de plantilla declarativo (por ejemplo, JSON). Este archivo especifica la AMI base, los comandos de aprovisionamiento (como scripts Shell, Ansible o Chef) y el resultado deseado. Packer ejecuta estos pasos automáticamente, construye la imagen, la prueba y, finalmente, la registra como una nueva AMI en AWS. Esta automatización garantiza que el proceso de 'baking' sea consistente y repetible, integrándose perfectamente en las tuberías de Integración Continua/Despliegue Continuo (CI/CD).

Además de las herramientas externas, AWS ofrece servicios nativos como **AWS Image Builder**, un servicio totalmente gestionado que simplifica la creación, el mantenimiento y la distribución de imágenes de servidor seguras y actualizadas. Image Builder automatiza el proceso de construcción, parcheo, prueba y distribución de AMI, asegurando que las imágenes cumplan con los estándares de seguridad corporativos. Utilizando estos enfoques automatizados, las organizaciones pueden reducir la deriva de configuración (configuration drift) y asegurarse de que cada servidor de producción se lance a partir de una imagen que ha sido probada y validada sistemáticamente.

## 6. Seguridad y Compartición

La seguridad de las AMI es un aspecto crítico, ya que una imagen comprometida o mal configurada puede resultar en la exposición de toda la infraestructura. El primer punto de control

de seguridad es el **Endurecimiento de la Imagen (Image Hardening)**. Esto implica asegurarse de que la AMI base tenga el mínimo de software necesario para su función, que todos los paquetes estén actualizados y que las configuraciones de seguridad, como los cortafuegos locales y las políticas de acceso, se hayan implementado correctamente antes de la creación de la imagen. Las AMI de AWS Marketplace, aunque convenientes, deben ser revisadas para asegurar que cumplen con las políticas internas de la organización.

El segundo aspecto fundamental es la gestión de los **Permisos de Lanzamiento**. Por defecto, una AMI personalizada es privada y solo puede ser utilizada por la cuenta de AWS que la creó. Si es necesario compartir la AMI con otras cuentas (por ejemplo, entre equipos de desarrollo y producción, o con socios), los permisos deben ser configurados explícitamente. Sin embargo, compartir una AMI conlleva riesgos significativos. Si la AMI contiene datos sensibles, claves SSH o credenciales codificadas, estas serán accesibles para cualquier cuenta con la que se comparta. Por lo tanto, la práctica estándar de seguridad dicta que las credenciales nunca deben integrarse en la imagen de la AMI, sino que deben inyectarse de forma segura en tiempo de ejecución utilizando servicios como [AWS Systems Manager Parameter Store](#) o AWS Secrets Manager.

Finalmente, la gestión de las AMI públicas y de la comunidad exige una vigilancia constante. Si bien las AMI de la comunidad ofrecen soluciones preconstruidas, existe el riesgo de que contengan puertas traseras o malware. Los equipos de seguridad deben establecer políticas estrictas que limiten el uso de AMI externas o, en su defecto, que requieran un análisis de vulnerabilidades exhaustivo antes de su aprobación. La utilización de **AMI encriptadas**, mediante el uso de AWS Key Management Service (KMS), es una práctica recomendada para proteger los datos en reposo contenidos en la instantánea de EBS subyacente de la AMI.

## 7. Importancia e Impacto en la Computación en la Nube

El concepto de **Amazon Machine Image** ha tenido un impacto transformador en la forma en que las empresas abordan el despliegue y la gestión de la infraestructura en la nube. Su importancia radica en su capacidad para catalizar la **inmutabilidad de la infraestructura**. Antes de la estandarización de las imágenes de máquina, era común que los administradores de sistemas aplicaran parches y modificaciones directamente a los servidores en ejecución ('mutabilidad'), lo que conducía a la temida "deriva de configuración" y al problema del "funciona en mi máquina". La AMI resuelve esto al garantizar que cada servidor se inicie a partir de una plantilla idéntica y validada, promoviendo la consistencia en todos los entornos (desarrollo, prueba y producción).

Además de la consistencia, las AMI son cruciales para la **escalabilidad y la elasticidad**. En un entorno de escalado automático, donde cientos o miles de instancias pueden ser lanzadas en minutos en respuesta a picos de tráfico, la velocidad de aprovisionamiento es esencial. Dado que la AMI ya contiene el SO y las aplicaciones preinstaladas, las instancias se inician

significativamente más rápido que si tuvieran que pasar por un proceso de instalación y configuración en tiempo de ejecución. Esta eficiencia de lanzamiento es lo que permite a los servicios de AWS, como Auto Scaling, responder de manera casi instantánea a las demandas fluctuantes del mercado.

A nivel operativo, las AMI simplifican enormemente la **recuperación ante desastres (Disaster Recovery)** y la gestión de parches. En lugar de aplicar parches a instancias en ejecución, el método moderno consiste en crear una nueva AMI parcheada y luego reemplazar progresivamente las instancias antiguas con las nuevas (actualización rodante). Esto reduce el riesgo de fallos durante el parcheo y facilita la reversión a una versión anterior conocida si el nuevo parche causa problemas. Por lo tanto, la AMI no es solo una plantilla de lanzamiento, sino un componente central de las estrategias de fiabilidad, seguridad y eficiencia operativa en el entorno de AWS.

## 8. Lecturas Adicionales

[AWS Documentation: Amazon Machine Images \(AMIs\)](#)

[Wikipedia: Amazon Web Services \(AWS\)](#)

[HashiCorp Packer Official Documentation](#)