

Cloud Essentials

Practico N° 6
Informatica

Repaso

- Información general sobre los servicios de informática
- Amazon EC2
- Optimización de costos con Amazon EC2
- Servicios de contenedores
- Introducción a AWS Lambda
- Introducción a AWS Elastic Beanstalk

Servicios de informática de AWS

Servicios	Conceptos clave	Características	Facilidad de uso
<ul style="list-style-type: none"> Amazon EC2 	<ul style="list-style-type: none"> Infraestructura como servicio (IaaS) Basado en instancias Máquinas virtuales 	<ul style="list-style-type: none"> Aprovisione máquinas virtuales que pueda administrar como lo desee. 	Un concepto familiar para muchos profesionales de TI.
<ul style="list-style-type: none"> AWS Lambda 	<ul style="list-style-type: none"> Informática sin servidor Basado en funciones Bajo costo 	<ul style="list-style-type: none"> Escriba e implemente código que se ejecute de forma programada o que se pueda desencadenar mediante eventos. Utilícelo cuando sea posible (diseñado para la nube). 	Un concepto relativamente nuevo para muchos miembros del personal de TI, pero fácil de usar después de aprender cómo hacerlo.
<ul style="list-style-type: none"> Amazon ECS Amazon EKS AWS Fargate Amazon ECR 	<ul style="list-style-type: none"> Informática basada en contenedores Basado en instancias 	<ul style="list-style-type: none"> Ponga en marcha y ejecute los trabajos con mayor rapidez. 	AWS Fargate reduce la sobrecarga administrativa, pero puede utilizar opciones que le ofrecen mayor control.
<ul style="list-style-type: none"> AWS Elastic Beanstalk 	<ul style="list-style-type: none"> Plataforma como servicio (PaaS) Para aplicaciones web 	<ul style="list-style-type: none"> Concéntrese en el código (la creación de su aplicación). Se puede vincular fácilmente con otros servicios: bases de datos, sistema de nombres de dominio (DNS), etc. 	Es fácil y rápido comenzar a trabajar con este servicio.

Amazon EC2

Características

- Proporciona máquinas virtuales (denominadas **instancias EC2**) en la nube.
- Le ofrece **control total** sobre el sistema operativo invitado (Windows o Linux) en cada instancia.
- Puede lanzar instancias de cualquier tamaño en una zona de disponibilidad en cualquier lugar del mundo.
- Lance instancias desde imágenes de Amazon Machine (AMI).

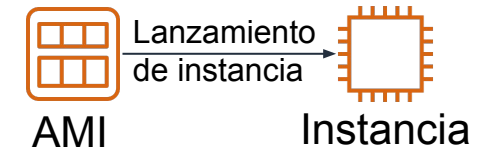
Pasos para lanzar una instancia

1. Seleccionar una AMI
2. Seleccionar un tipo de instancia
3. Especificar la configuración de red
4. Asociar rol de IAM
5. Especificar un script de datos de usuario
6. Especificar el almacenamiento
7. Agregar etiquetas
8. Configurar el grupo de seguridad
9. Identificar o crear el par de claves

1. Seleccionar una AMI

Imagen de Amazon Machine (AMI)

- Es una plantilla que se utiliza para crear una instancia EC2
- Contiene un sistema operativo **Windows, Linux o macOS**.
- También suele tener algún sistema de **software** preinstalado.
- Se pueden lanzar varias instancias a partir de una sola AMI.



Componentes

- Plantilla para el volumen raíz: SO y paquetes necesarios
- Permisos de lanzamiento: controlan que cuenta pueden usar la AMI
- Asignación de dispositivos de bloques: especifica los volúmenes que deben asociarse a la instancia en su lanzamiento, si corresponde

Elecciones de AMI

- **De inicio rápido:** AMI de Linux y Windows proporcionadas por AWS
- **Mis AMI:** todas las AMI que haya creado
- **AWS Marketplace:** catálogo con plantillas preconfiguradas de terceros
- **AMI de la comunidad:** AMI compartidas por otros usuarios

2. Seleccionar un tipo de instancia

	Uso general	Optimizadas para informática	Optimizadas para memoria	Informática acelerada	Optimizadas para almacenamiento
Tipos de instancias	a1, m4, m5, t2, t3	c4, c5	r4, r5, x1, z1	f1, g3, g4, p2, p3	d2, h1, i3
Caso de uso	Amplio	Alto rendimiento	Bases de datos en memoria	Aprendizaje automático	Sistemas de archivos distribuidos

- Cada tipo de instancia varía en
 - Memoria (RAM)
 - CPU
 - Espacio en disco y tipo de disco
 - Rendimiento de la red (ancho de banda en Gbps)
- Se agrupan en categorías
- Los tipos de instancias ofrecen familias, generaciones y tamaños

Características de red

- Ancho de banda en Gbps varía en función del tipo de instancia
- Maximizar rendimiento de red:
 - lanzar instancia en grupo de ubicación en cluster
 - habilitar redes mejoradas
 - Elastic Network Adapter (ENA) – hasta 100 Gbps
 - Interfaz de intel 82599 Virtual Function – hasta 10 Gbps

Nombre de la instancia	vCPU	Memoria (GB)	Almacenamiento
t3.nano	2	0,5	Solo EBS
t3.micro	2	1	Solo EBS
t3.small	2	2	Solo EBS
t3.medium	2	4	Solo EBS
t3.large	2	8	Solo EBS
t3.xlarge	4	16	Solo EBS
t3.2xlarge	8	32	Solo EBS

Ejemplo: **t3.large**

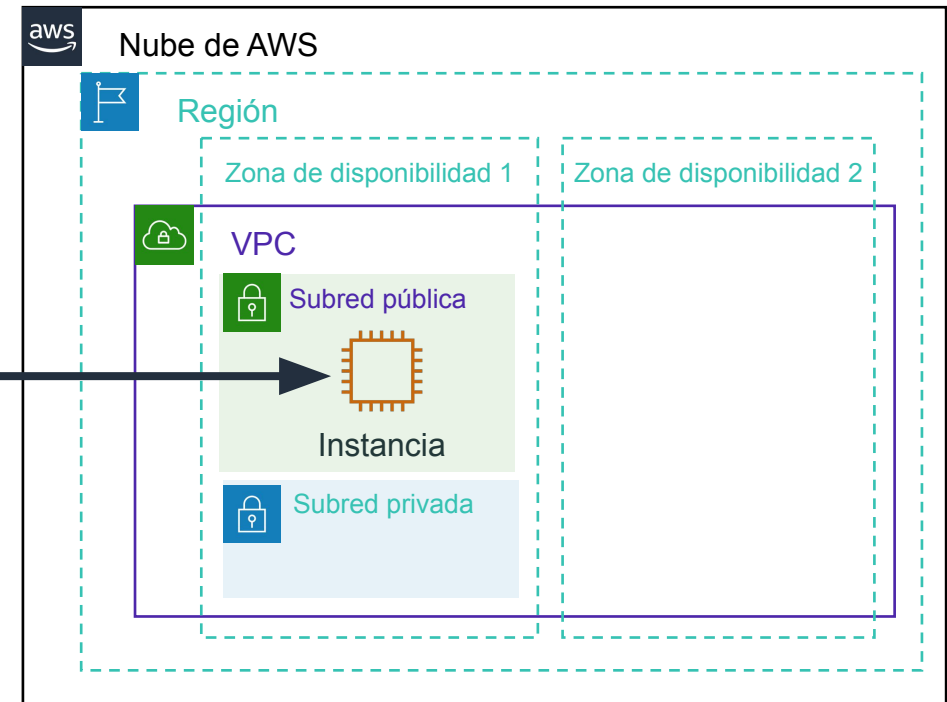
- **T** es la familia
- **3** es la generación
- **Large** es el tamaño

3. Especificar configuración de red

- Escoger región
- Identificar la VPC y/o subred
- Asignar IP pública si se quiere acceder desde internet
- Cuando se lanza una instancia en una **VPC predeterminada**, AWS le asigna una **dirección IP** pública de forma predeterminada.
- Cuando se lanza una instancia en una **VPC no predeterminada**, la subred tiene un atributo que determina si las instancias lanzadas en esa subred reciben una dirección IP pública del grupo de direcciones IPv4 públicas.

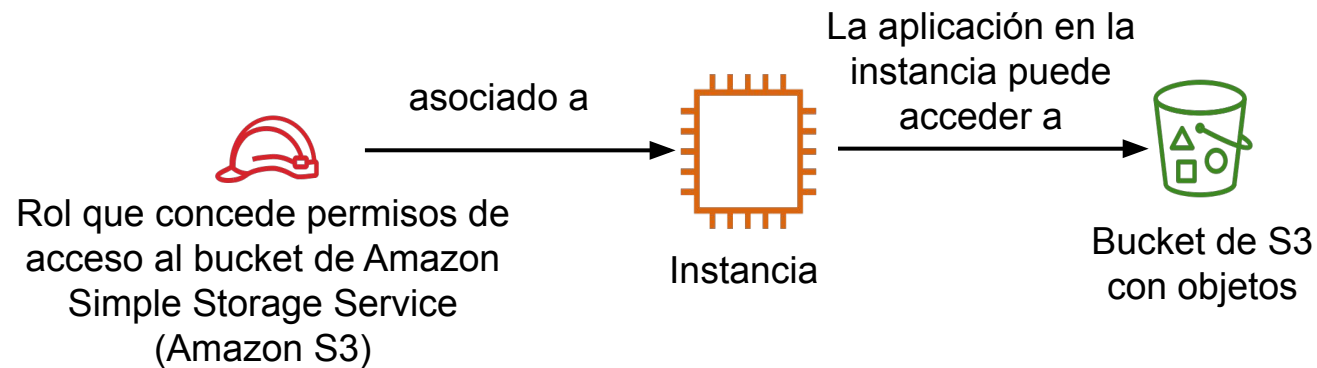
*Ejemplo:
especifique que
se implemente
la instancia aquí*

- De forma predeterminada, AWS no asignará una dirección IP pública a las instancias que se lancen en una subred no predeterminada.
- Puede controlar ese comportamiento modificando el atributo de asignación de direcciones IP públicas de la subred o también desde el lanzamiento



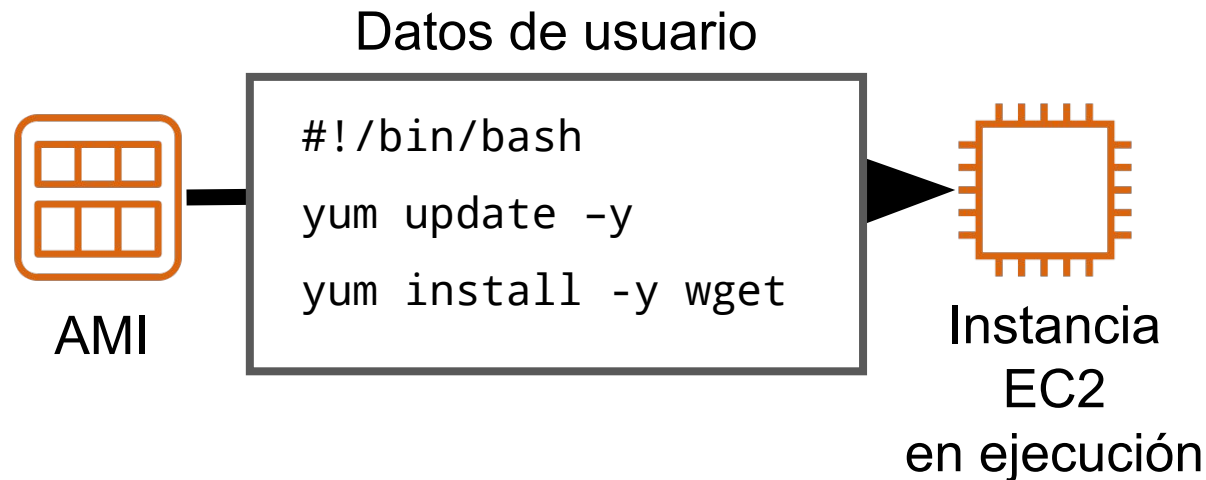
4. Asociar rol de IAM (opcional)

- Si la instancia EC2 tiene que interactuar con otros servicios de AWS, es necesario asociar un rol
- Un rol de IAM asociado a una instancia se llama **Instance Profile**
- Se puede asociar antes o después del lanzamiento
- Ejemplo:



5. Especificar un script de datos de usuario (opcional)

- Utilice scripts de datos de usuario para personalizar el entorno de tiempo de ejecución de la instancia.
- Pueden automatizar la finalización de las instalaciones y las configuraciones durante el lanzamiento de la instancia.
- El script se ejecuta la primera vez que se inicia la instancia.
- Para una instancia de Windows, el script de datos de usuario debe escribirse en un formato compatible con cmd o Windows PowerShell
- Cuando se cree la instancia EC2, el script se ejecutará con privilegios raíz durante las fases finales del proceso de arranque.
 - En las instancias de Linux, se ejecuta a través del servicio cloud-init.
 - En las instancias de Windows, se ejecuta mediante la utilidad EC2Config o EC2Launch.



6. Especificar el almacenamiento

- Configure el volumen raíz.
 - Lugar donde está instalado el sistema operativo invitado
- Se puede adjuntar volúmenes de almacenamiento adicionales (opcional).
- Para cada volumen, especifique lo siguiente:
 - El tamaño del disco (en GB)
 - El tipo de volumen
 - unidades de estado sólido (SSD) (varios disponibles)
 - unidades de disco duro (HDD) (varios disponibles)
 - Eliminación del mismo al terminarse la instancia
 - Cifrado
- Opciones de almacenamiento
 - **Amazon Elastic Block Store (Amazon EBS):**
 - Almacenamiento de nivel de bloque persistente.
 - Puede detener la instancia e iniciarla de nuevo sin perder los datos.
 - **Almacén de instancias de Amazon EC2:**
 - Discos asociados al equipo host en el que se ejecuta la instancia EC2.
 - Si la instancia se detiene, se eliminarán los datos almacenados aquí.
 - Útil para información que cambia con frecuencia, como buffers, memorias caché, etc
 - Otras opciones de almacenamiento (no para el volumen raíz):
 - **Amazon Elastic File System (Amazon EFS)**
 - Conectarse a **Amazon Simple Storage Service (Amazon S3)**

7. Agregar tags

- Los tags o etiquetas son marcas de un recurso, que consta de una clave-valor
 - { "Name": "Instancia01" }
 - { "Environment": "dev" }
- Funcionan como metadatos para la instancia

Las claves son *case sensitive*
- Ayuda al filtrado, automatización, asignación de costos y control de acceso

8. Configurar el grupo de seguridad

- **Grupo de seguridad:** conjunto de **reglas** de firewall que controlan el tráfico a la instancia.
 - Se encuentra fuera del sistema operativo invitado de la instancia.
- Las **reglas** habilitan el tráfico hacia o desde las instancias asociadas
- El **origen** puede ser una dirección IP, un intervalo de direcciones IP, otro grupo de seguridad, un punto de enlace de la VPC de gateway o cualquier lugar (lo que significa que se permitirán todos los orígenes).
- Por defecto, ya existe la regla que permite todo el tráfico saliente, pero es modificable

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
SSH ▾	TCP	22	My IP ▾ 72.21.198.67/32

9. Identificar o crear par de claves

- Un par de claves consta de lo siguiente:
 - Una clave pública que AWS almacena.
 - Un archivo de clave privada que el usuario almacena.
- Se debe especificar en el lanzamiento de la instancia, usar uno ya creado o crear uno nuevo
- **Posibilita las conexiones seguras a la instancia.**
- Para las **AMI de Windows:**
 - Utilice la clave privada para obtener la contraseña de administrador que necesita para iniciar sesión en la instancia.
 - Puede conectarse mediante SSH con una herramienta como PuTTY usando la clave
- Para las **AMI de Linux:**
 - Utilice la clave privada para utilizar SSH y conectarse de forma segura a la instancia.



mykey.pem

Vista de una instancia EC2 en ejecución

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a search bar. The left sidebar lists various AWS services, with 'Instances' selected under the 'EC2 Dashboard' section. The main content area shows a list of instances with a search filter applied to the instance ID 'i-092b6f3efba959a53'. The instance is in a 'running' state. Below the list, the 'Description' tab is active, showing detailed information about the instance, including its ID, state, type, availability zone, security groups, and network interfaces.

Name	Instance ID	Instance Type	Instance State	Status Checks	Public DNS (IPv4)	IPv4 Public IP
	i-092b6f3efba959a53	t2.micro	running	Initializing	ec2-54-159-171-63.co...	54.159.171.63

Instance: **i-092b6f3efba959a53** Public DNS: **ec2-54-159-171-63.compute-1.amazonaws.com**

Description		Status Checks	Monitoring	Tags
Instance ID	i-092b6f3efba959a53			
Instance state	running			
Instance type	t2.micro			
Elastic IPs				
Availability zone	us-east-1c			
Security groups	launch-wizard-1. view inbound rules. view outbound rules			
Scheduled events	No scheduled events			
AMI ID	amzn2-ami-hvm-2.0.20190823.1-x86_64-gp2 (ami-0b69ea66ff7391e80)			
Platform	-			
Public DNS (IPv4)	ec2-54-159-171-63.compute-1.amazonaws.com			
IPv4 Public IP	54.159.171.63			
IPv6 IPs	-			
Private DNS	ip-172-31-82-44.ec2.internal			
Private IPs	172.31.82.44			
Secondary private IPs				
VPC ID	vpc-e4e9859e			
Subnet ID	subnet-d22779fc			
Network interfaces	eth0			

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Despliegue de instancia EC2 por medio de AWS CLI

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1  
--instance-type c3.large --key-name MyKeyPair  
--security-groups MySecurityGroup --region us-east-1
```

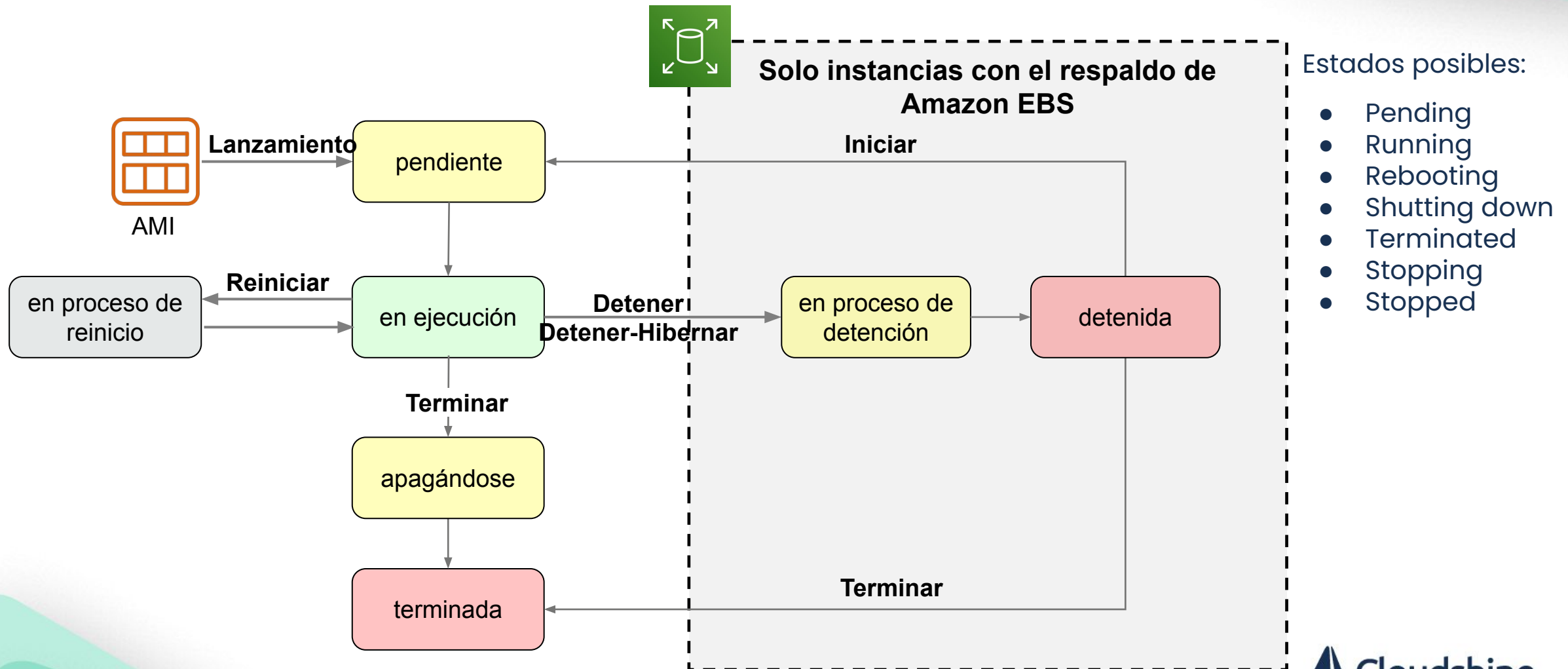
El comando incluye la siguiente información:

- **aws:** especifica la invocación de la utilidad de *línea de comandos* aws.
- **ec2:** especifica la invocación del *comando del servicio* ec2.
- **run-instances:** es el subcomando que se invoca.

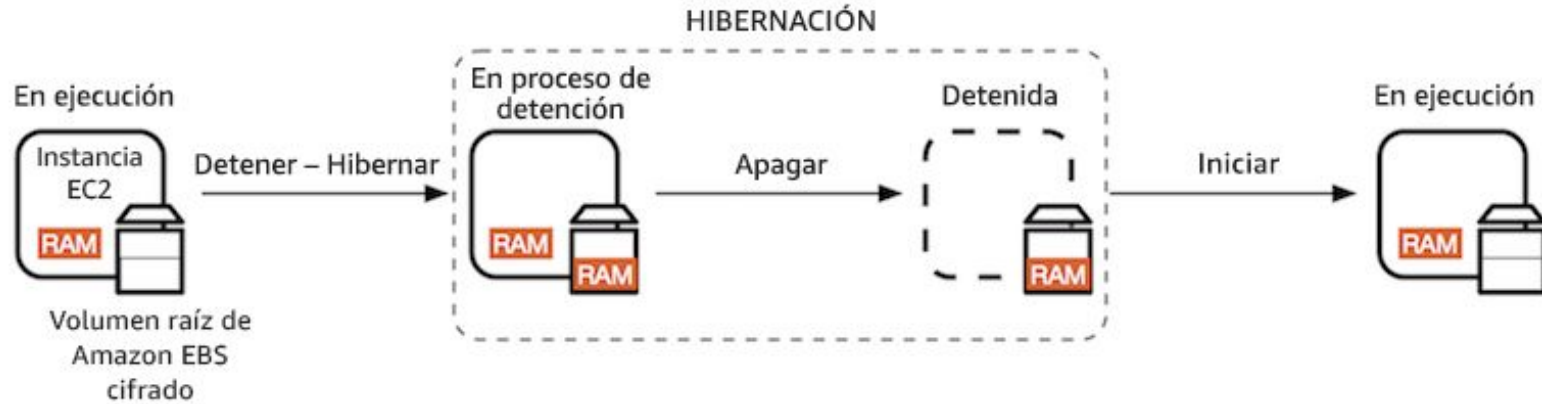
El resto del comando especifica varios parámetros, entre los que se incluyen los siguientes:

- **image-id:** este parámetro va seguido de un ID de AMI. Todas las AMI tienen un ID de único.
- **count:** puede especificar más de una instancia.
- **instance-type:** puede especificar el tipo de instancia que se creará, como una instancia c3.large.
- **key-name:** en el ejemplo, supongamos que *MyKeyPair* ya existe.
- **security-groups:** en este ejemplo, supongamos que *MySecurityGroup* ya existe.
- **region:** las AMI se encuentran en una región de AWS, por lo que debe especificar la región donde la CLI de AWS encontrará la AMI y lanzará la instancia EC2.

Ciclo de vida de las instancias EC2



Opción de hibernación de instancias



- **Beneficios**
 - Guarda el contenido de la memoria de la instancia (RAM).
 - Cuando se reinicia la instancia, se reanudan los procesos anteriores
 - Los costos son similares a los de una instancia detenida).
- **Requisitos previos**
 - algunas AMI de Linux (como Amazon Linux 2) y ciertas familias de instancias la admiten.
 - La instancia debe tener un volumen raíz de Amazon EBS cifrado y un máximo de 150 GB de RAM.
 - La opción de hibernación debe estar habilitada al momento de lanzar la instancia.

Uso de IP Elastica

- A cada instancia que recibe una **dirección IP pública** se le asigna también un **nombre de host DNS externo**. Por ejemplo, si la dirección IP pública asignada a la instancia es 203.0.113.25, el nombre de host DNS externo podría ser ec2-203-0-113-25.compute-1.amazonaws.com
- Si especifica que se debe asignar una dirección IP pública a la instancia, esta se asigna desde el grupo de direcciones IPv4 públicas de AWS. La dirección IP pública **no está asociada a su cuenta de AWS**. Cuando se desvincula una dirección IP pública de la instancia, **se vuelve a liberar** en el grupo de direcciones IPv4 públicas, y no podrá indicar que desea volver a utilizarla
- AWS libera la dirección IP pública de la instancia cuando la instancia se **detiene o se termina**. La instancia detenida **recibe una dirección IP pública nueva cuando se reinicia**
- Si necesita una dirección IP pública permanente, le recomendamos asociar una **dirección IP elástica** a la instancia. Para hacerlo, primero debe asignar una nueva dirección IP elástica en la región donde se encuentra la instancia. Una vez asignada, puede asociar la dirección IP elástica a una instancia EC2.
- De manera predeterminada, todas las cuentas de AWS **están limitadas a cinco (5) direcciones IP elásticas por región** porque las direcciones públicas de Internet (IPv4) son un recurso público escaso. Sin embargo, este límite no es fijo, y se puede solicitar un aumento del límite (que podría aprobarse).

Monitoreo con Amazon Cloudwatch

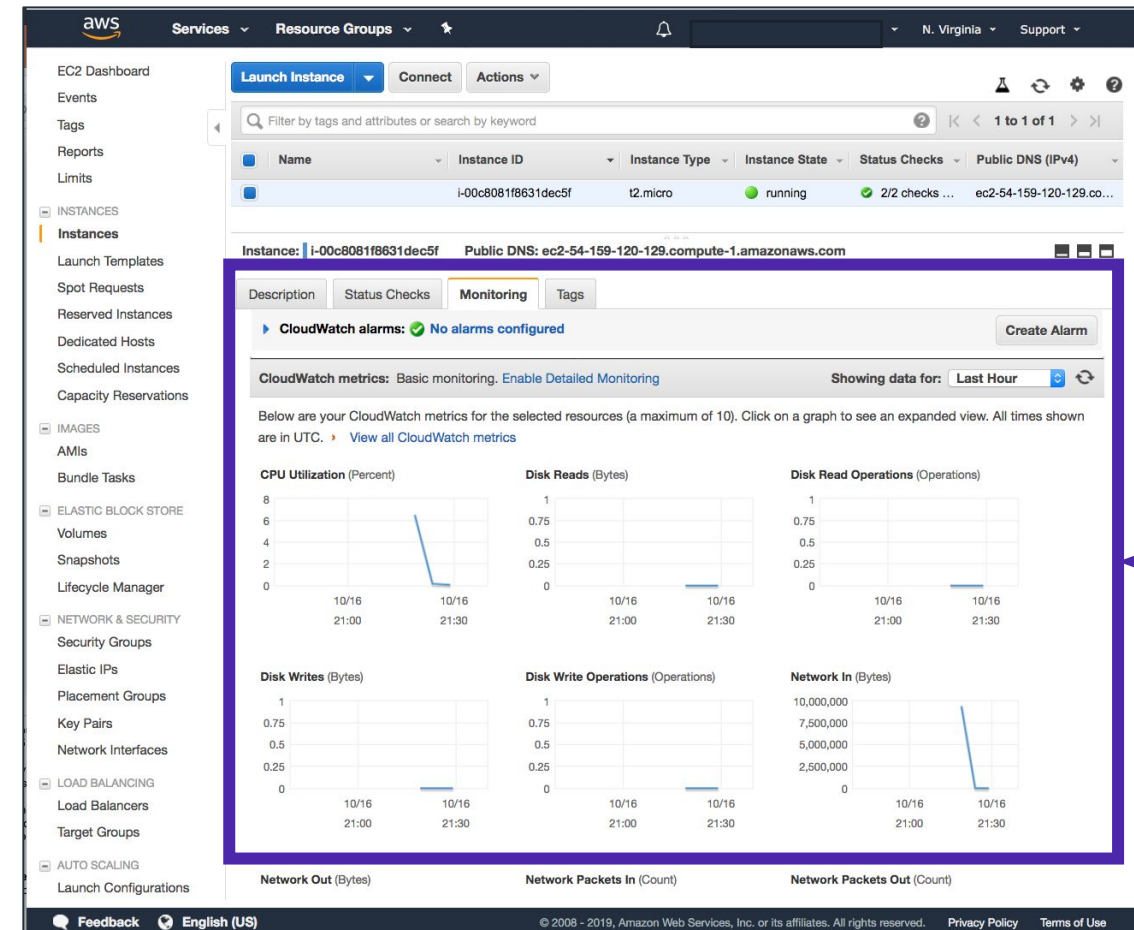
- Proporciona métricas casi en tiempo real.
- Brinda gráficos que puede ver en la pestaña Monitoring (Monitoreo) de la consola de Amazon EC2.
- Mantiene los datos históricos durante 15 meses.
- **Monitoreo básico**
 - Valor predeterminado, sin costo adicional
 - Datos de métricas que se envían a CloudWatch cada 5 minutos
- **Monitoreo detallado**
 - Tarifa mensual fija para siete métricas preseleccionadas
 - Datos de métricas que se envían cada 1 minuto
- De forma predeterminada, Amazon CloudWatch **no proporciona métricas de la RAM** para las instancias EC2, pero puede configurar esta opción si desea que CloudWatch recopile estos datos.



Amazon CloudWatch



Instancia con CloudWatch



Optimización de costos con Amazon EC2

	Modelos de precios			
	On-demand	Spot	Reservadas	Hosts dedicados
Características del modelo de precio	<ul style="list-style-type: none"> - Pago por hora - Sin compromiso a largo plazo - Apto capa gratuita AWS 	<ul style="list-style-type: none"> - Ejecución en base a disponibilidad y oferta superior - AWS puede interrumpirlas (terminación, detención o hibernación) con una notificación de 2 minutos. - Mucho más económicas que las on-demand 	<ul style="list-style-type: none"> - Pago inicial completo, parcial o nulo - Descuento en el cargo por hora - Plazo de 1 a 3 años - Standard - Programadas: - Reserva de capacidad que esté disponible y según la programación periódica (día, semana o mes) - Plazo de 1 año - Convertibles: - Capacidad de cambio de atributos siempre por igual o mayor valor 	<p>Hosts dedicados: Servidor físico con capacidad de instancias EC2 totalmente dedicado a su uso</p> <p>Instancias dedicadas: Instancias que se ejecutan en una VPC en el hardware dedicado a un solo cliente</p>
Beneficios	Bajo costo y flexibilidad	Carga de trabajo dinámica y a gran escala	Predictibilidad asegura capacidad de cómputo disponible cuando se la necesita	Ahorro de dinero en costos de licencia Asistencia para cumplir los requisitos normativos y de conformidad
Casos de uso	Cargos de trabajo de corto plazo, con picos o impredecibles Desarrollo o prueba de aplicaciones	<ul style="list-style-type: none"> - Aplicaciones con horarios flexibles de inicio y finalización - Aplicaciones que solo son viables con precios de computación muy bajos - Usuarios con necesidades de computación urgentes de grandes cantidades de capacidad adicional 	<ul style="list-style-type: none"> - Cargas de trabajo de uso predecible o estado estable - Aplicaciones que requieren capacidad reservada, incluida la recuperación de desastres - Usuarios capaces de afrontar pagos iniciales para reducir aún más los costos de computación 	<ul style="list-style-type: none"> - Licencia "Bring your own" (BYOL) - Conformidad y restricciones normativas - Seguimiento del uso y las licencias - Control de la ubicación de instancias

Los 4 pilares de la optimización de costos

Para optimizar los costos, debe tener en cuenta cuatro factores constantes y muy importantes:

- **Adaptación del tamaño:**

- Aproveche instancias para satisfacer las necesidades (uso de cpu y memoria)
- Use métricas de Amazon CloudWatch.
- Práctica recomendada: adapte el tamaño, luego reserve.

- **Aumento de elasticidad:**

- Detenga o ponga a hibernar instancias con EBS que no están usándose de forma activa.
- Use el escalado automático para satisfacer las necesidades en función del uso.

- **Modelo de precios óptimo:**

- Aproveche el modelo de precios adecuado para su caso de uso.
- Optimice y combine los tipos de compras, de acuerdo a cargas variables (instancias a demanda o spot) o fijas (instancias reservadas)
- Considere utilizar soluciones sin servidor (AWS Lambda)

- **Optimización de las opciones de almacenamiento:**

- Reduzca los costos y mantenga el rendimiento y la disponibilidad del almacenamiento.
- Cambie el tamaño de los volúmenes de EBS.
- Cambie los tipos de volúmenes de EBS (pasar de SSD a HDD).
- Elimine las instantáneas de EBS que ya no necesita.
- Identifique el destino más adecuado para determinados tipos de datos.

- ¿Es necesario que la instancia resida en Amazon EBS para la aplicación?

- Las opciones de almacenamiento de Amazon S3 con políticas de ciclo de vida pueden reducir los costos.

Servicios de contenedores

Amazon Elastic Container Service (Amazon ECS)

- Es un servicio de administración de contenedores altamente escalable y rápido
- Las características básicas de Amazon ECS incluyen la posibilidad de hacer lo siguiente:
 - **Lanzar** hasta decenas de miles de contenedores de Docker en cuestión de segundos
 - **Monitorear** la implementación de contenedores
 - **Administrar** el estado del clúster que ejecuta los contenedores
 - **Programar** contenedores con un programador integrado o de terceros (por ejemplo, Apache Mesos o Blox)
- Los clústeres de Amazon ECS también pueden utilizar instancias de spot e instancias reservadas.
- Beneficios clave:
 - Organiza la ejecución de contenedores de Docker.
 - Mantiene y escala la flota de nodos que ejecutan sus contenedores.
 - Elimina la complejidad de poner en marcha la infraestructura.
- Integración con:
 - Elastic Load Balancing
 - Grupos de seguridad de Amazon EC2
 - Volúmenes de Amazon EBS
 - Roles de IAM

Servicios de contenedores

Amazon Elastic Container Service (Amazon ECS)

- Para preparar la aplicación con el fin de ejecutarla en Amazon ECS, debe crear una **definición de tarea**, es decir, un archivo de texto que describe uno o varios contenedores, hasta un **máximo de diez**, que componen la aplicación..
- Una **tarea** es la instancia creada de una definición de tarea dentro de un clúster. Puede especificar el número de tareas que se ejecutarán en el clúster. El programador de tareas de Amazon ECS es responsable de colocar las tareas dentro del **clúster**. Una tarea se ejecutará en uno a diez contenedores, según la definición de tarea que haya establecido.
- Cuando Amazon ECS ejecuta los contenedores que componen la tarea, los coloca en un **clúster** de ECS. El clúster (cuando elige el tipo de lanzamiento de EC2) consta de un **grupo de instancias EC2**, y **cada una ejecuta un agente de contenedor** de Amazon ECS.
- Amazon ECS ofrece varias **estrategias de programación** que colocarán los contenedores en clústeres en función de las necesidades de recursos (por ejemplo, CPU o RAM) y los requisitos de disponibilidad.
- Administración del cluster
 - respaldado por **Amazon EC2** (mayor control sobre la infraestructura)
 - respaldado por **AWS Fargate** (fácil de mantener y par enfocarse en las aplicaciones)

Servicios de contenedores

Amazon Elastic Kubernetes Service (Amazon EKS)

- **Kubernetes** es un software de código abierto para la **organización de contenedores** (orquestación).
- Complementa a Docker, ya que puede organizar varios hosts de Docker (nodos)
- Automatiza estos procesos:
 - El aprovisionamiento de contenedores
 - La redes
 - La distribución de carga
 - El escalado
- Amazon EKS permite ejecutar Kubernetes en AWS.
 - Cuenta con el **certificado de conformidad** de Kubernetes (admite una migración sencilla).
 - Admite contenedores de Linux y Windows.
 - Es compatible con las herramientas de la **comunidad de Kubernetes** y admite complementos populares de Kubernetes.
- Use Amazon EKS para lo siguiente:
 - Administrar clústeres de instancias de informática de Amazon EC2
 - Ejecutar contenedores organizados por Kubernetes en esas instancias

Servicios de contenedores

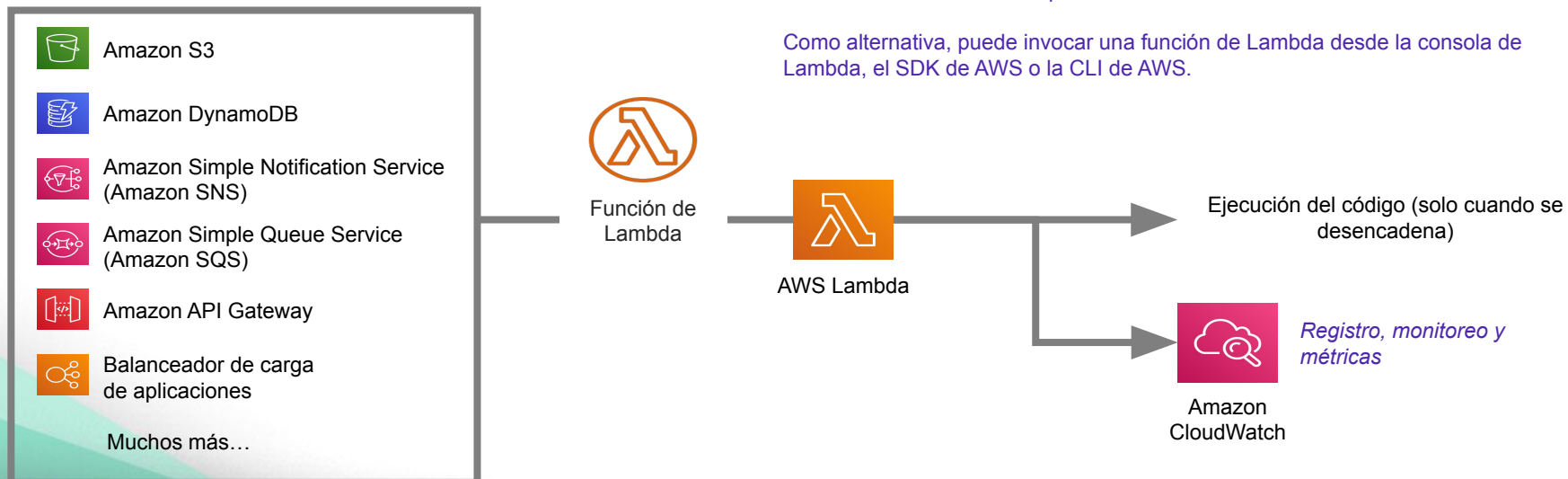
Amazon Elastic Container Registry (Amazon ECR)

- Amazon ECR es un **registro de contenedores de Docker** completamente administrado
- Facilita a los desarrolladores las tareas de almacenamiento, administración e implementación de imágenes de contenedores de Docker.
- Es **compatible con la API HTTP de Docker Registry**, versión 2, que le permite interactuar con Amazon ECR utilizando los comandos de la CLI de Docker o las herramientas de Docker que prefiera
- Puede **transferir las imágenes** del contenedor desde y hacia Amazon ECS mediante HTTPS. Las imágenes también se **cifran** automáticamente en reposo mediante el cifrado del lado del servidor de Amazon S3.
- Integración con Amazon ECS y EKS
- Compatibilidad con Docker
- Colaboración en equipo
- Control de acceso
- Integración con terceros

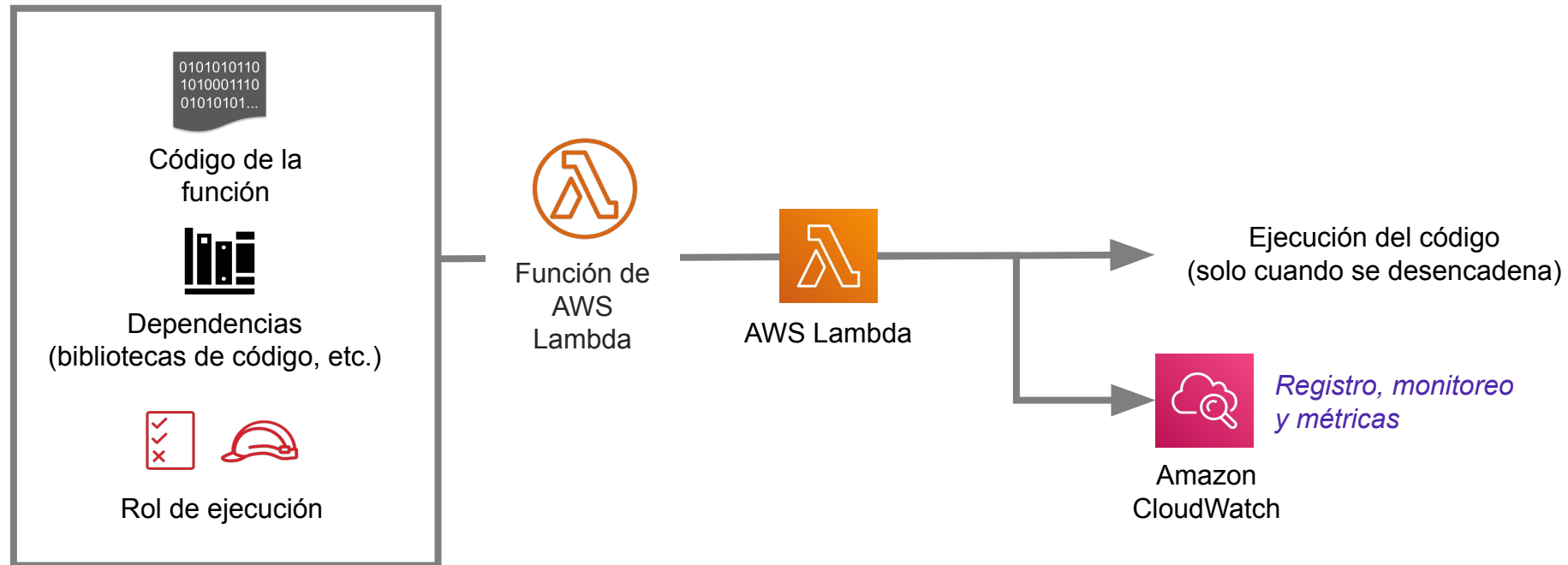
AWS Lambda

- **AWS Lambda** es un **servicio de informática sin servidor** basado en eventos. Este servicio le permite ejecutar código sin aprovisionar ni administrar servidores.
- Características:
 - Compatibilidad con varios lenguajes de programación (Java, Go PowerShell, Node.js, C#, Python y Ruby)
 - Administración totalmente automatizada
 - Tolerancia a errores integrada
 - Admite la organización de varias funciones (AWS Step Functions)
 - Precios de pago por el uso

Orígenes de eventos



AWS Lambda - Configuración

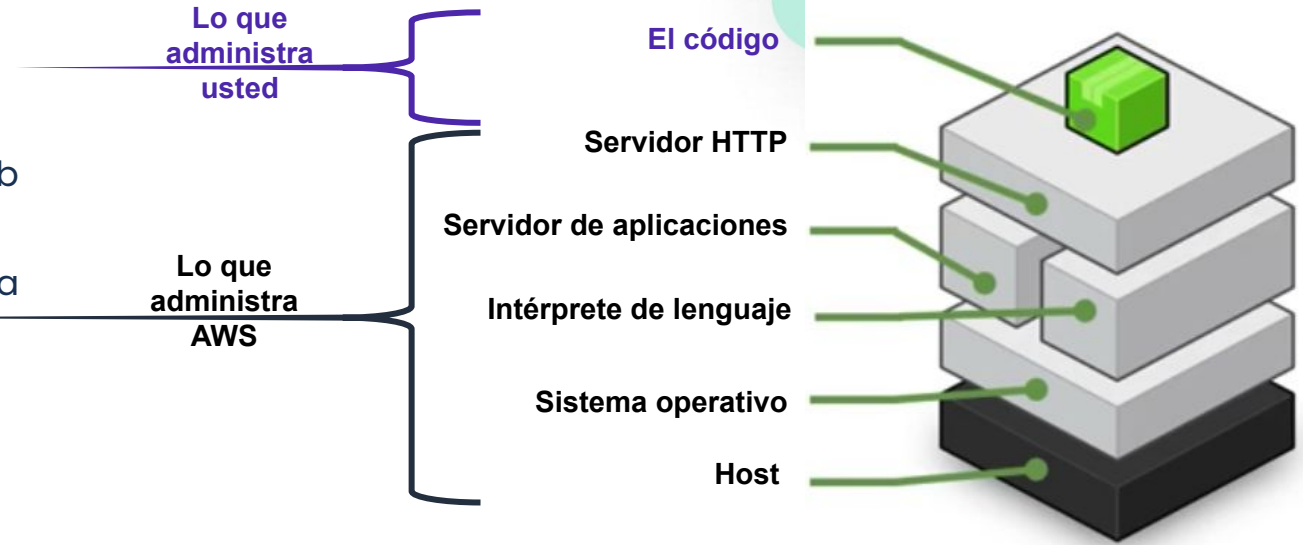


Limites de AWS Lambda

- Límites flexibles por región:
 - Ejecuciones simultáneas = 1000
 - Almacenamiento de funciones y capas = 75 GB
- Límites invariables para funciones individuales:
 - Asignación máxima de memoria de funciones = 3008 MB
 - Tiempo de espera de la función = 15 minutos
 - Tamaño del paquete de implementación = 250 MB sin comprimir, incluidas las capas

AWS Elastic Beanstalk

- Una forma sencilla de poner en marcha las aplicaciones web
- Servicio administrado que gestiona automáticamente:
 - Aprovisionamiento y configuración de la infraestructura
 - Implementación
 - Balanceo de cargas
 - Escalado automático
 - Monitoreo de estado
 - Análisis y depuración
 - Registros
- No se aplican cargos adicionales por utilizar Elastic Beanstalk, solo por los recursos que termina utilizando el servicio (Instancias EC2, buckets S3, etc)
- Admite aplicaciones web escritas para plataformas comunes (Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker)
- Con solo cargar el código, el servicio administra la implementación
 - Se implementa en servidores como Apache, NGINX, Passenger, Puma y Microsoft Internet Information Services (IIS).
- Beneficios:
 - Rápida y sencilla puesta en marcha
 - Mayor productividad
 - Los desarrolladores se centran solamente en el código
 - Difícil de superar
 - Su aplicación puede manejar las cargas máximas de trabajo o tráfico, y usted puede minimizar sus costos al mismo tiempo
 - Control total de los recursos
 - En caso de que quiera encargarse de algunos elementos de la infraestructura (o de todos), puede hacerlo sin problemas



Gracias