Les damos la bienvenida – CLASE 10

Desarrollo e Implementación de Sistemas en la Nube

2do año / 2do Tramo
2do Cuatrimestre 2025

NOS PRESENTAMOS…

Rodrigo San Roman

Docente PP III: Desarrollo en la Nube

Ingeniero en Electrónica

Especialista en Telecomunicaciones y Redes de Datos

Líder Ciberseguridad Canales Digitales / Especialista y Arquitecto Cloud / Networking / Ciberseguridad

CONTENIDOS

Unidad N°6 – 2da Parte – Profundización Contenedores

- Conceptos básicos de contenedores
- Contenedores rootless
- Contenedores vs. máquinas virtuales
- ¿Qué es Docker?
- Orquestador contenedores
- ¿Qué es Kubernetes?
- Pods y Workloads
- Aseguramiento contenedores
- Servicio de Contenedores

Unidad N°6 – 2da Parte – Contenedores

Contenedores

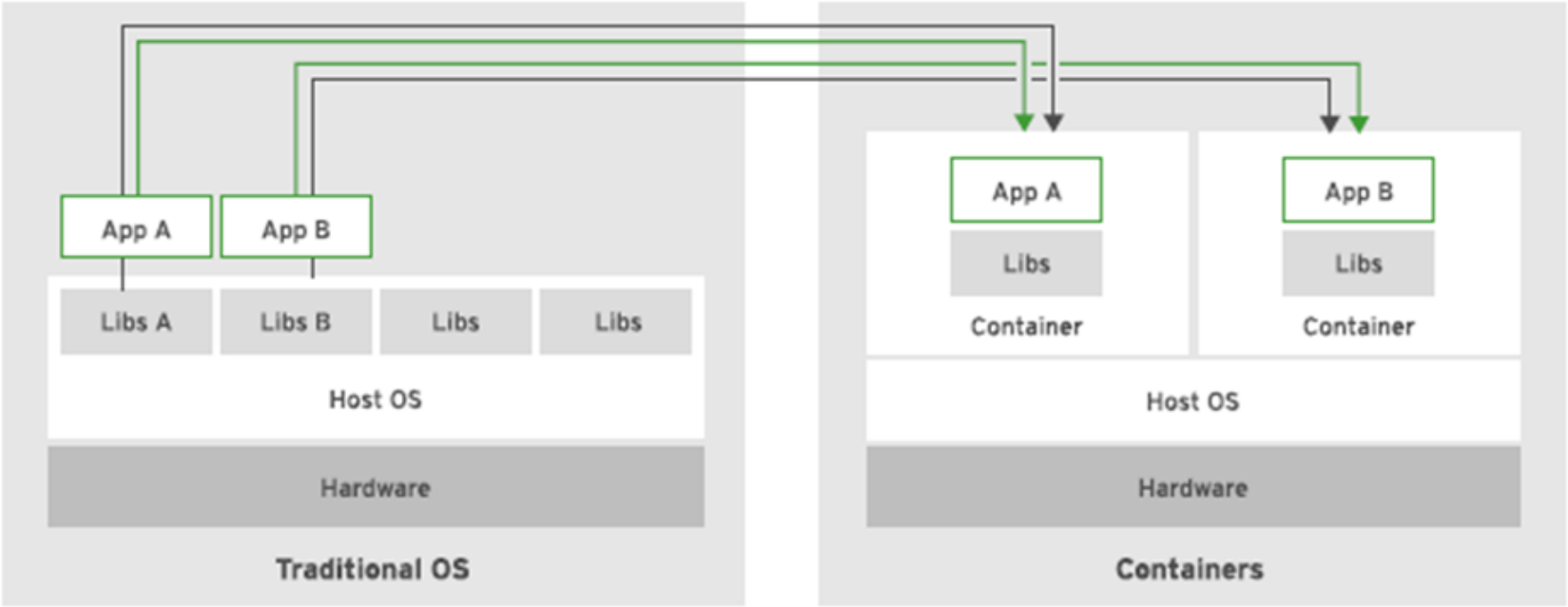
Contenedores – Conceptos básicos

Conceptos básicos de contenedores

- Los contenedores son un método de virtualización del sistema operativo.
- Beneficios:
  - Repetible
  - Entornos de ejecución autónomos
  - Software que se ejecuta de la misma manera en diferentes entornos
    - En el equipo portátil del desarrollador, entornos de prueba y entornos de producción
  - Lanzamiento y detención o terminación más rápidos que las máquinas virtuales

Conceptos básicos de contenedores

- Los contenedores **son un método de virtualización** del **sistema operativo** que le permite ejecutar una aplicación y sus dependencias en procesos de recursos aislados, es decir, son un **conjunto de uno o más procesos que están aislados del resto del sistema**.



Diferencias entre contenedor y sistema operativo

Conceptos básicos de contenedores

- Un solo **contenedor** se puede usar para **ejecutar** cualquier cosa, desde un **microservicio** o un **proceso** de **software** a una **aplicación** de **mayor tamaño**.
- **Dentro** de un **contenedor** se **encuentran** todos los **ejecutables**, el código **binario**, las **bibliotecas** y los **archivos** de **configuración** necesarios.
- En comparación con los métodos de virtualización de máquinas o servidores, los **contenedores no contienen imágenes** del **sistema operativo**. Esto los hace más ligeros y portátiles, con una sobrecarga significativamente menor.
- En implementaciones de aplicaciones de mayor tamaño, se pueden **poner** en marcha **varios contenedores** como uno o varios **clústeres** de contenedores.
- Estos **clústeres** se pueden **gestionar** mediante un **orquestador** de contenedores, como **Kubernetes**.

Conceptos básicos de contenedores

- **Menos sobrecarga**
  Los contenedores requieren menos recursos del sistema que los entornos de máquinas virtuales tradicionales o de hardware porque no incluyen imágenes del sistema operativo.

- **Mayor portabilidad**
  Las aplicaciones que se ejecutan en contenedores se pueden poner en marcha fácilmente en sistemas operativos y plataformas de hardware diferentes.

Conceptos básicos de contenedores

- **Funcionamiento más constante**
  Los equipos de DevOps saben que las aplicaciones en contenedores van a ejecutarse igual, independientemente de dónde se pongan en marcha.

- **Mayor eficiencia**
  Los contenedores permiten poner en marcha, aplicar parches o escalar las aplicaciones con mayor rapidez.

- **Mejor desarrollo de aplicaciones**
  Los contenedores respaldan los esfuerzos ágiles y de DevOps para acelerar los ciclos de desarrollo, prueba y producción.

Contenedores rootless

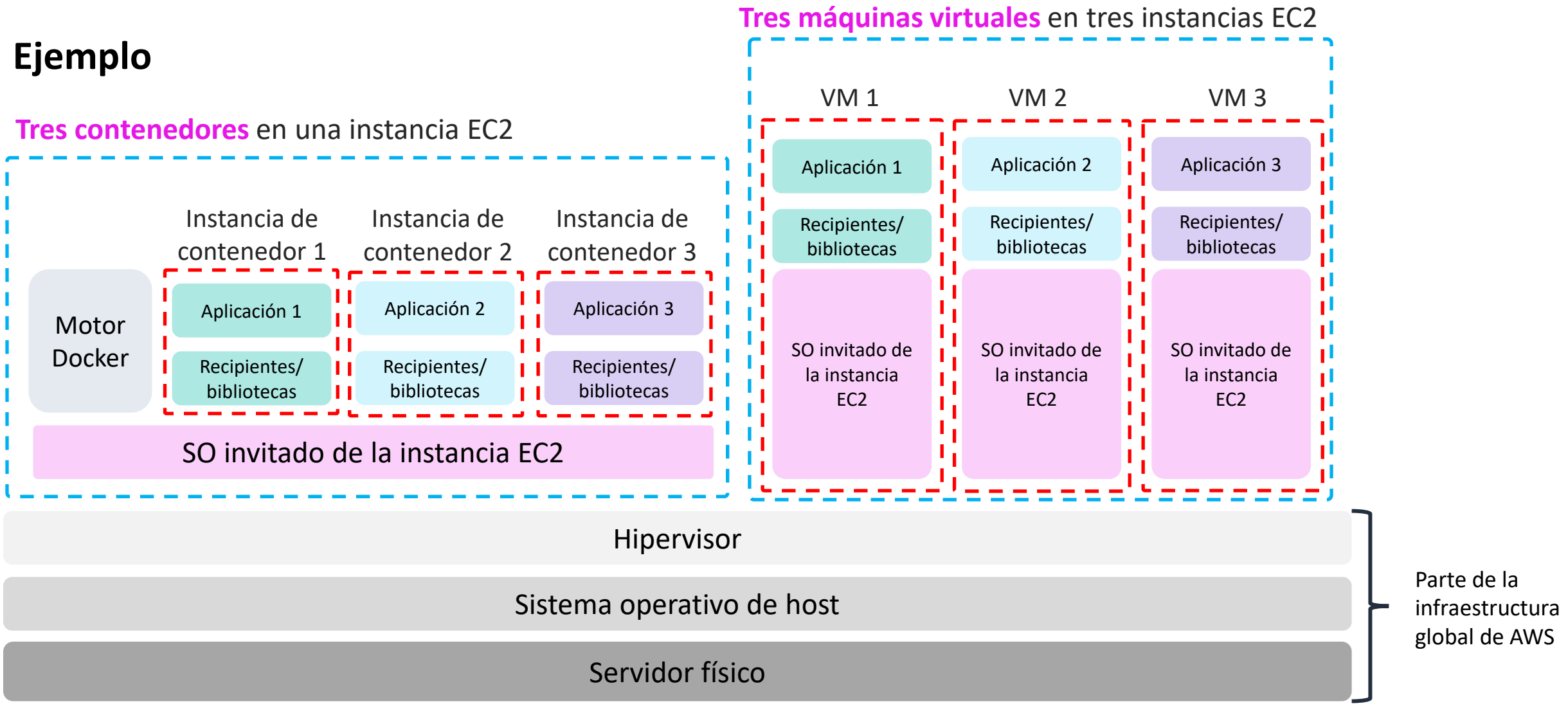
No requieren privilegios de root para ejecutarse.

Ventajas:

- Permite que el código se ejecute dentro de un contenedor rootless con privilegios de root, sin tener que ejecutarse como usuario root del host
- Agrega una nueva capa de seguridad; si el motor de contenedores está comprometido, el atacante no obtendrá privilegios root en el host
- Permite que varios usuarios sin privilegios ejecuten contenedores en la misma máquina
- Permite el aislamiento dentro de contenedores anidados

Contenedores frente a máquinas virtuales

Ejemplo



Contenedores frente a máquinas virtuales

- En la **derecha** del **diagrama**, se muestra una implementación basada en máquinas virtuales (**VM**).
- Cada una de las **tres instancias EC2** se ejecuta directamente en el hipervisor que proporciona la infraestructura global de AWS.
- **Cada instancia EC2 ejecuta** una máquina **virtual**. Cada una de las **tres aplicaciones** se ejecuta en su **propia** máquina **virtual**, lo que proporciona **aislamiento de procesos**.

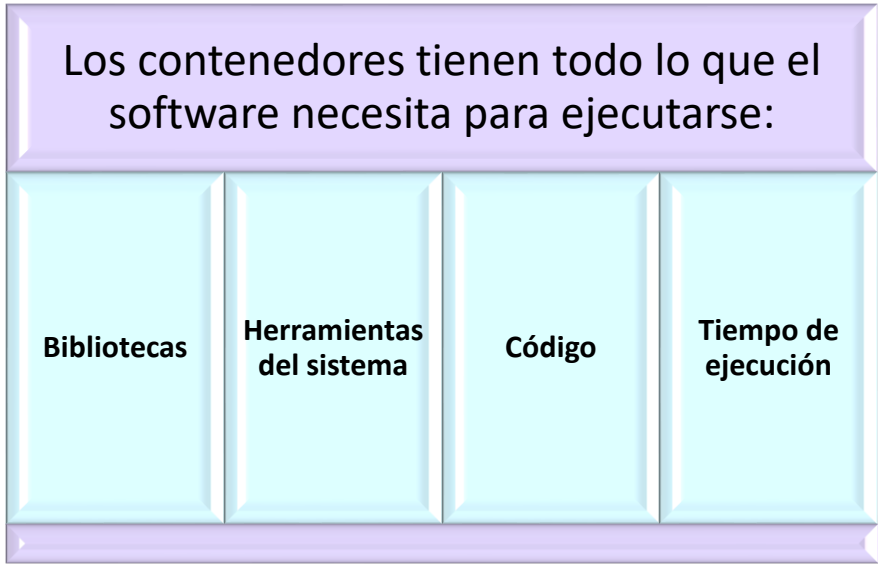
Contenedores frente a máquinas virtuales

- En la parte **izquierda** del diagrama, se muestra una implementación basada en **contenedores**.
- Hay una **sola** instancia **EC2** que ejecuta una máquina **virtual**.
- El motor **Docker** se instala en el sistema operativo Linux de la instancia EC2 y **hay tres contenedores**.
- Cada **aplicación** se ejecuta en su **propio contenedor** (que proporciona **aislamiento de procesos**).

Entonces…¿Qué es Docker?

- **Docker** es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente.
- Puede ejecutar contenedores en Docker
  - Los contenedores se crean a partir de una plantilla denominada *imagen*.
- Un **contenedor** tiene todo lo que una aplicación de software necesita para ejecutarse.

Contenedor



Gestión varios contenedores

- **Docker** es un popular **entorno** en **tiempo de ejecución** que se usa para **crear y construir software dentro de contenedores**.
- **Docker** se basa en **estándares abiertos** y **funciona** en la mayoría de los **entornos operativos más comunes**, incluidos Linux, Microsoft Windows y otras infraestructuras locales o basadas en la nube.
- Las **aplicaciones** en contenedores pueden ser complicadas, pudiendo **requerir** cientos o **miles** de **contenedores** independientes.

Gestión varios contenedores

- En este sentido, los **entornos** en tiempo de ejecución de contenedores, como **Docker**, se **benefician** del uso de otras **herramientas** para **orquestar** o gestionar todos los **contenedores** en funcionamiento.
- Una de las **herramientas** más **populares** para este fin es **Kubernetes**, un orquestador de contenedores que reconoce varios entornos en tiempo de ejecución de contenedores, incluido Docker.
- **Facilita** la **automatización** y el **escalado** de cargas de trabajo.

Entonces…¿Qué es Kubernetes?

- Kubernetes es un software de código abierto para la organización de contenedores
  - Implementa y **administra aplicaciones en contenedores** a escala
  - El mismo conjunto de herramientas se puede usar on-prem o cloud
- Complementa a Docker
  - Docker le permite ejecutar varios contenedores en un solo host del sistema operativo
  - Kubernetes **organiza** varios hosts de Docker (nodos).
- Automatiza estos procesos:
  - El aprovisionamiento de contenedores
  - La redes
  - La distribución de carga
  - El escalado

Pods y Workloads

- ¿Qué es un **pod**? Los pods son los **objetos más pequeños** y básicos que se pueden implementar en Kubernetes. Un pod representa una instancia única de un proceso en ejecución en un clúster. Podemos pensar que un pod es un "**host lógico**" autónomo y aislado.
- Un **workload** en **Kubernetes** es **cualquier cosa** que esté siendo **ejecutada** en el **clúster de Kubernetes**. La plataforma de Kubernetes se encarga de gestionarlos, asegurando que se ejecuten de manera confiable y escalable.

Pods y Workloads

- Los pods contienen uno o más contenedores, como los contenedores de Docker.
- Cuando un ***pod ejecuta varios contenedores***, estos se administran como una ***sola entidad y comparten los recursos del pod***. En general, ejecutar varios contenedores en un solo pod representa un caso práctico avanzado.

Contenedores

Contenedores - Conceptos de seguridad

Aseguramiento contenedores

- **Imagen contenedor**
  - Imágenes mínimas
  - Software actualizado
- **Registro imágenes**
  - Registro privado
  - Hardenizar repositorio registro
  - Control acceso repositorio
  - Depuración imágenes obsoletas
  - Usar únicamente imágenes confiables

Aseguramiento contenedores

- **Runtime contenedor**
  - Controlar el tráfico de salida enviado por contenedores
  - IP dinámica asignada automáticamente
- **Host OS**
  - Protecciones SO (ejemplo: SELinux)
- **Orquestador**
  - Limitar usuarios privilegiados
  - Encriptar volúmenes
  - Segmentación

OWASP Docker Top 10

D01 - Secure User Mapping
D02 - Patch Management Strategy
D03 - Network Segmentation and Firewalling
D04 - Secure Defaults and Hardening
D05 - Maintain Security Contexts
D06 - Protect Secrets
D07 - Resource Protection
D08 - Container Image Integrity and Origin
D09 - Follow Immutable Paradigm
D10 - Logging

CIS Docker Benchmark

1. **Instalación y configuración segura**
2. **Configuración de imágenes seguras**
3. **Configuración del demonio de Docker**
    1. Restringir el acceso a interfaces de red
    2. Habilitar el registro de actividades y limitar los recursos del sistema que puede utilizar Docker.
4. **Configuración de contenedores seguros**
    1. Rootless
5. **Gestión de usuarios y permisos**
6. **Auditoría y registro**
7. **Protección del demonio de Docker**
8. **Actualizaciones y mantenimiento**

Contenedores

Servicios de contenedores (AWS)

Amazon Elastic Container Service (Amazon ECS)

Para evitar lanzar una o varias instancias de Amazon EC2, instalar Docker en cada instancia y administrar y ejecutar los contenedores de Docker en dichas instancias de Amazon EC2, AWS ofrece un servicio denominado ***Amazon Elastic Container Service*** (Amazon ***ECS***) que simplifica la administración de los contenedores.

Amazon Elastic Container Service (Amazon ECS)

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 (Apache Mesos, Blox, etc.)

Amazon Elastic Container Service (Amazon ECS)

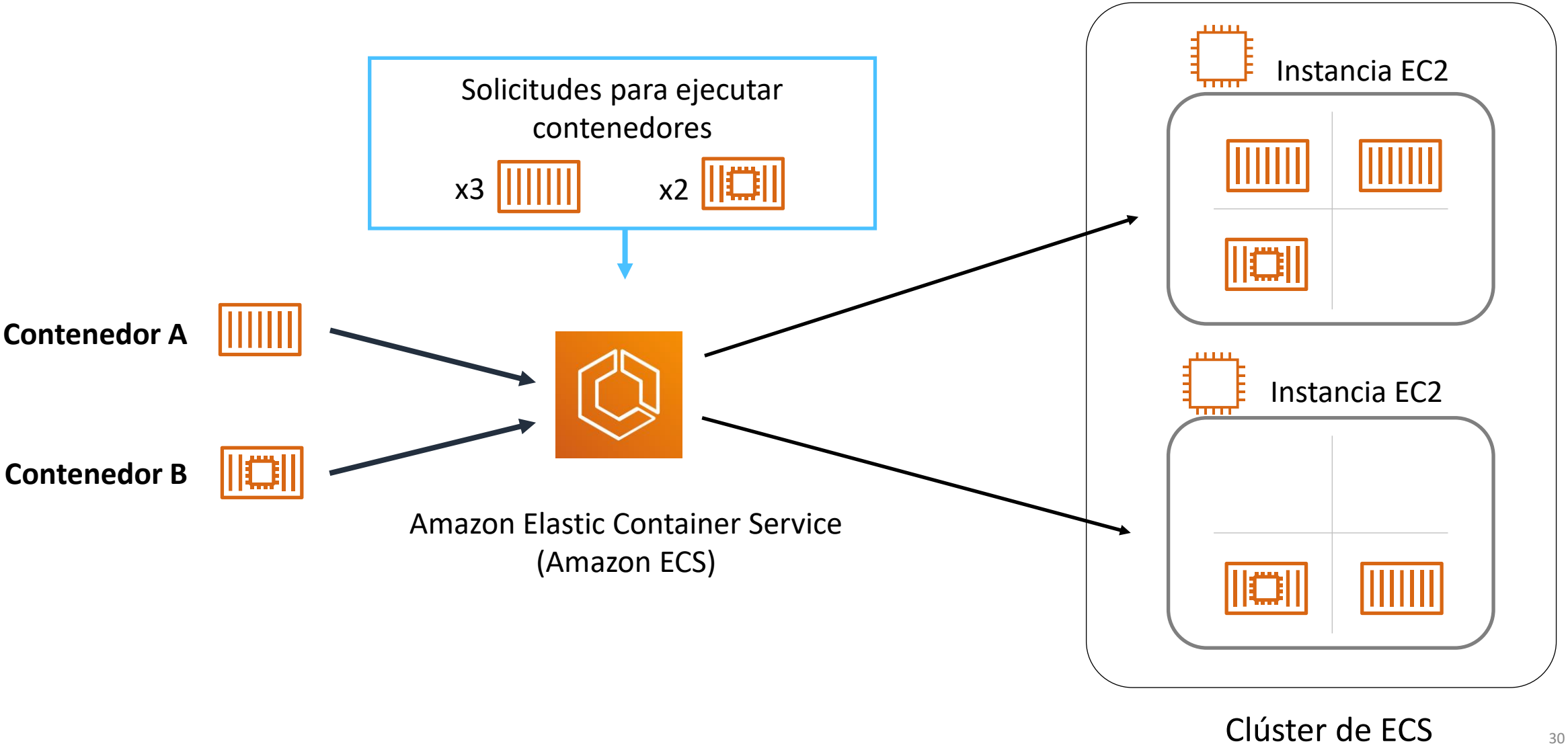
- Amazon Elastic Container Service (**Amazon ECS**):
  - Un servicio de administración de contenedores altamente escalable y rápido

Amazon Elastic Container Service

- 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 características que los usuarios de servicios de Amazon EC2 conocen:
  - Elastic Load Balancing
  - Grupos de seguridad de Amazon EC2
  - Volúmenes de Amazon EBS
  - Roles de IAM

Amazon ECS organiza contenedores



Amazon ECS organiza contenedores

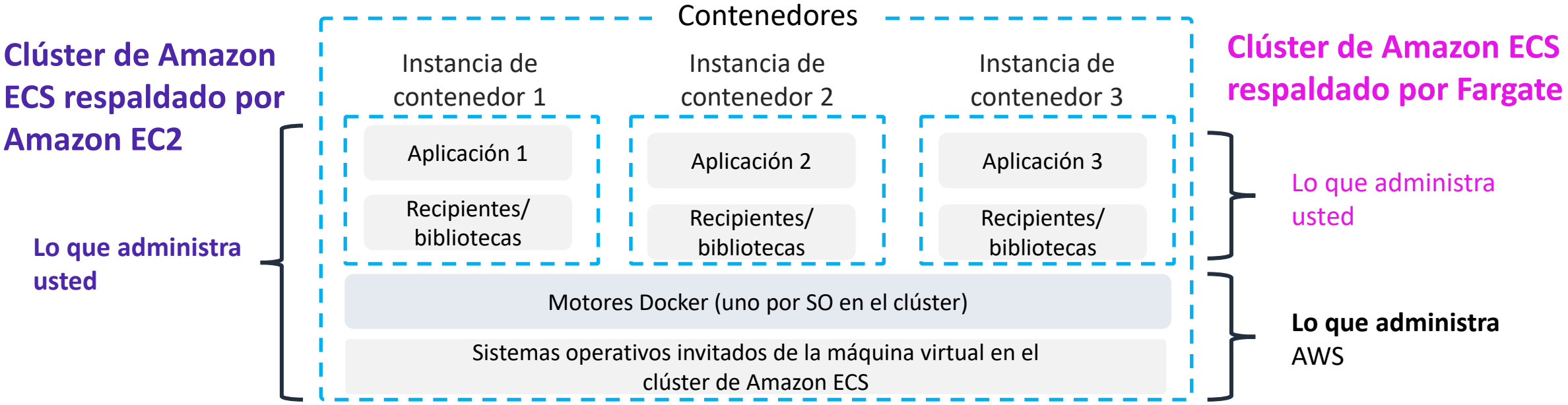
- Para preparar una aplicación con el fin de ejecutarla en Amazon ECS, se debe crear una **definición de tarea**, es decir, un archivo de texto que **describe uno o varios contenedores**, hasta un máximo de diez, que componen la aplicación.
- Podemos decir que es un plano técnico de la aplicación. La definición de tareas especifica los parámetros para la aplicación, como los contenedores que se utilizarán, los puertos que se deben abrir para la aplicación y los volúmenes de datos que se deben utilizar con los contenedores en la tarea.

Amazon ECS organiza contenedores

- 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**.

Opciones de clúster de Amazon ECS

- **Pregunta clave**: ¿***Desea*** administrar el clúster de Amazon ECS que ejecuta los contenedores?
  - En caso **afirmativo**, cree un **clúster de Amazon ECS respaldado por Amazon EC2,** que proporciona un control más detallado sobre la infraestructura.
  - De lo contrario, cree un clúster de Amazon ECS respaldado por AWS Fargate, que es más fácil de mantener y le permite centrarse en las aplicaciones.



Opciones de clúster de Amazon ECS

- Cuando crea un clúster de Amazon ECS, tiene tres opciones:
- Un clúster **de solo redes** (con tecnología de AWS Fargate)
- Un clúster **de redes + EC2 Linux**
- Un clúster **de redes** + **EC2 Windows**

¿Qué es Kubernetes?

- Kubernetes es un software de código abierto para la organización de contenedores.
  - Implemente y **administre aplicaciones en contenedores** a escala.
  - El mismo conjunto de herramientas se puede usar en las instalacioens y en la nube.
- Complementa a Docker.
  - Docker le permite ejecutar varios contenedores en un solo host del sistema operativo.
  - Kubernetes **organiza** varios hosts de Docker (nodos).
- Automatiza estos procesos:
  - El aprovisionamiento de contenedores
  - La redes
  - La distribución de carga
  - El escalado

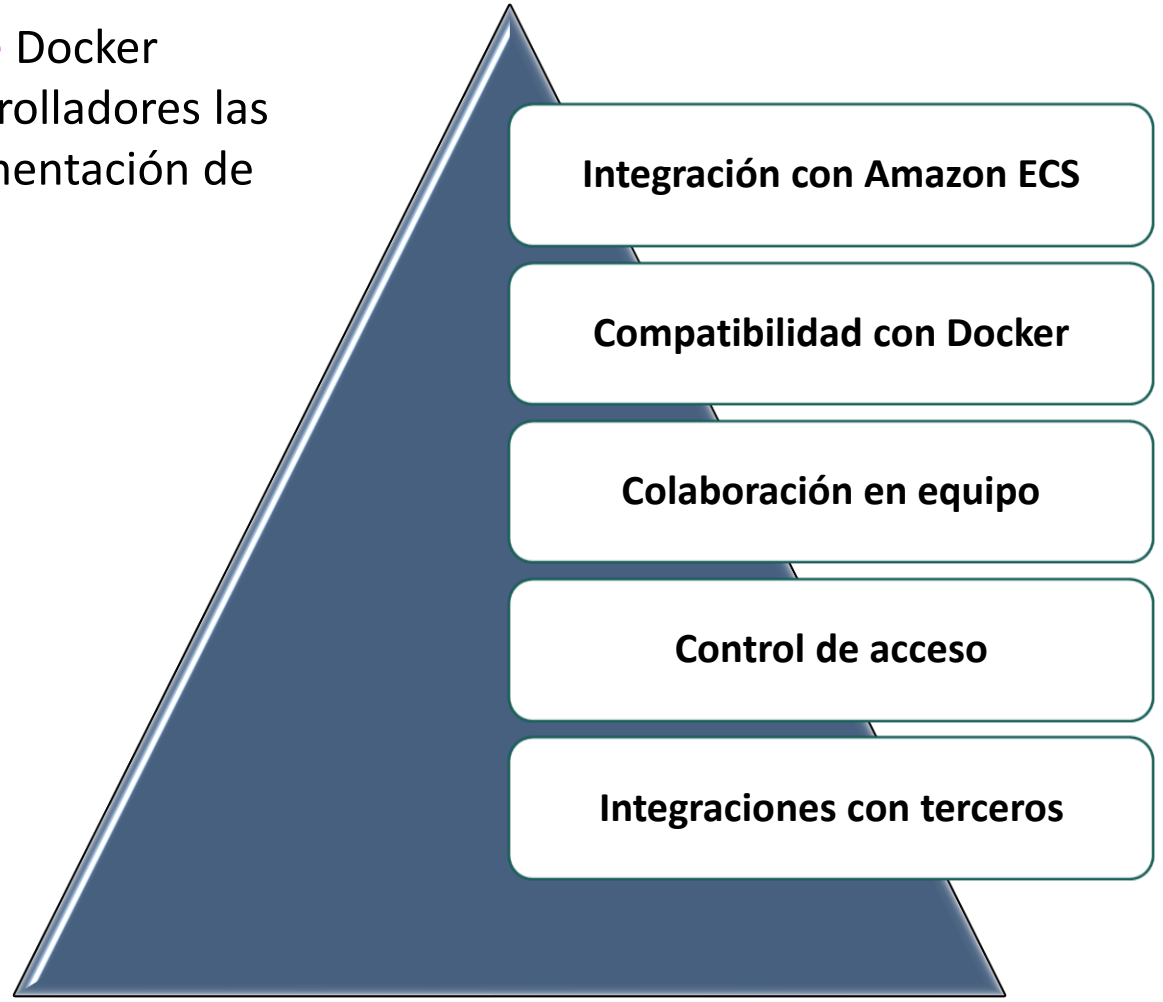
Amazon Elastic Container Registry (Amazon ECR)

Amazon ECR es un registro de contenedores de Docker completamente administrado que facilita a los desarrolladores las tareas de almacenamiento, administración e implementación de imágenes de contenedores de Docker.

Amazon Elastic Container Registry

Imagen

Registro



Amazon Elastic Kubernetes Service (Amazon EKS)

- Amazon Elastic Kubernetes Service (**Amazon EKS**)
  - Le 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.

Amazon Elastic Kubernetes Service

- Use Amazon EKS para lo siguiente:
  - Administrar clústeres de instancias de informática de Amazon EC2
  - Ejecutar contenedores organizados por Kubernetes en esas instancias

Amazon Elastic Kubernetes Service (Amazon EKS)

Amazon EKS es un servicio administrado de Kubernetes que permite ejecutar Kubernetes en AWS fácilmente, sin necesidad de instalar, gestionar ni mantener su propio plano de control.

Amazon EKS administra automáticamente la disponibilidad y escalabilidad de los nodos del clúster encargados de iniciar y detener contenedores, programar contenedores en máquinas virtuales y almacenar datos de clústeres, entre otras tareas.

Conclusiones importantes

- Los contenedores pueden abarcar todo lo que una aplicación necesita para ejecutarse.
- **Docker** es una plataforma de software que empaqueta software en contenedores.
  - Una sola aplicación puede abarcar varios contenedores.
- Amazon Elastic Container Service (**Amazon ECS**) organiza la ejecución de los contenedores de Docker.
- **Kubernetes** es un software de código abierto para la organización de contenedores.
- Amazon Elastic Kubernetes Service (**Amazon EKS**) le permite ejecutar Kubernetes en AWS.
- Amazon Elastic Container Registry (**Amazon ECR**) le permite almacenar, administrar e implementar sus contenedores de Docker.

MUCHAS GRACIAS