**Introducción**

Los contenedores y las máquinas virtuales son formas de virtualización, pero hay algunas diferencias clave entre ellos.

Para proporcionar contexto, consideremos un escenario: Es administrador de Azure responsable de implementar y administrar aplicaciones en un entorno en la nube. Su organización busca una solución que ofrece tiempos de inicio rápidos, administración sencilla y la capacidad de ejecutar aplicaciones en contenedores aislados. Quiere comprender las ventajas de usar Azure Container Instances y cómo se compara con las máquinas virtuales.

En este módulo, aprenderá a usar Azure Container Instances en lugar de máquinas virtuales. También obtendrá información general sobre las características y los casos de uso.

El objetivo de este módulo es presentarle Azure Container Instances.

**Objetivos de aprendizaje**

En este módulo aprenderá a:

* Identificar cuándo debe usar contenedores y cuándo máquinas virtuales.
* Reconocer las características y los casos de uso de Azure Container Instances.
* Implementar grupos de contenedores de Azure.

**Aptitudes evaluadas**

El contenido del módulo le ayuda a prepararse para el [examen AZ-104: Administrador de Microsoft Azure](https://learn.microsoft.com/es-es/certifications/exams/az-104).

**Requisitos previos**

* Conocimientos prácticos de los conceptos y la terminología de la contenedorización.
* Familiaridad con la informática en la nube y experiencia con Azure Portal.

**Comparación de las máquinas virtuales con los contenedores**

La virtualización de hardware ha hecho posible ejecutar varias instancias aisladas de sistemas operativos simultáneamente en el mismo hardware físico. Los contenedores representan la siguiente fase de la virtualización de los recursos informáticos.

La virtualización basada en contenedores le permite virtualizar el sistema operativo. Este enfoque permite ejecutar varias aplicaciones en la misma instancia de un sistema operativo y seguir manteniendo el aislamiento entre las aplicaciones. Los contenedores en una máquina virtual proporcionan una funcionalidad similar a la de las máquinas virtuales en un servidor físico.

**Aspectos que deben conocerse sobre los contenedores frente a las máquinas virtuales**

Para comprender mejor la virtualización basada en contenedores, vamos a comparar los contenedores con las máquinas virtuales.

| **Comparación** | **Containers** | **Máquinas virtuales** |
| --- | --- | --- |
| **Aislamiento** | Un contenedor suele proporcionar un aislamiento ligero del host y otros contenedores, pero no proporciona un límite de seguridad tan sólido como el de una máquina virtual. | Una máquina virtual proporciona un aislamiento completo del sistema operativo host y otras máquinas virtuales. Esta separación es útil cuando un límite de seguridad sólido es fundamental, como en el hospedaje de aplicaciones de empresas de la competencia en el mismo servidor o clúster. |
| **Sistema operativo** | Los contenedores ejecutan la parte del modo de usuario de un sistema operativo y se pueden personalizar para que contengan solo los servicios necesarios para la aplicación. Este enfoque ayuda a usar menos recursos del sistema. | Las máquinas virtuales ejecutan un sistema operativo completo que incluye el kernel, para lo que se necesitan más recursos del sistema (CPU, memoria y almacenamiento). |
| **Implementación** | Puede implementar contenedores individuales usando Docker desde la línea de comandos. Puede implementar varios contenedores usando un orquestador, como Azure Kubernetes Service. | Puede implementar máquinas virtuales individuales mediante Windows Admin Center o el administrador de Hyper-V. Puede implementar varias máquinas virtuales usando PowerShell o System Center Virtual Machine Manager. |
| **Almacenamiento persistente** | Los contenedores usan discos de Azure para el almacenamiento local de un único nodo, o bien Azure Files (recursos compartidos SMB) para el almacenamiento compartido por varios nodos o servidores. | Las máquinas virtuales usan un disco duro virtual (VHD) para el almacenamiento local de una sola máquina virtual, o bien un recurso compartido de archivos SMB para el almacenamiento compartido por varios servidores. |
| **Tolerancia a errores** | Si se produce un error en un nodo del clúster, el orquestador vuelve a crear rápidamente los contenedores que se ejecutan en él en otro nodo del clúster. | Las máquinas virtuales pueden conmutar por error a otro servidor de un clúster y el sistema operativo de la máquina virtual se reinicia en el nuevo servidor. |

**Aspectos que deben tenerse en cuenta cuando se usan contenedores**

Los contenedores ofrecen varias ventajas respecto a las máquinas físicas y virtuales. Revise las siguientes ventajas y piense cómo puede implementar contenedores para las aplicaciones internas de su empresa.

* **Considere la flexibilidad y la velocidad**. Obtenga mayor flexibilidad y velocidad al desarrollar y compartir el código de una aplicación contenerizada.
* **Considere la realización de pruebas**. Elija contenedores para su configuración y simplificar las pruebas en sus aplicaciones.
* **Considere la implementación de aplicaciones**. Implemente contenedores para lograr una implementación simplificada y acelerada de sus aplicaciones.
* **Considere la densidad de la carga de trabajo**. Trabaje con contenedores para poder admitir una mayor densidad de la carga de trabajo y mejorar el uso de los recursos.

**Revisión de Azure Container Instances**

Los contenedores están convirtiéndose en la manera preferida de empaquetar, implementar y administrar aplicaciones en la nube. Azure Container Instances ofrece la forma más rápida y sencilla de ejecutar un contenedor en Azure, sin tener que administrar ninguna máquina virtual y sin necesidad de adoptar un servicio de nivel superior. Azure Container Instances es una excelente solución para cualquier escenario que pueda funcionar en contenedores aislados, incluidas las aplicaciones simples, la automatización de tareas y los trabajos de compilación.

En la siguiente ilustración se muestra un contenedor de servidor web creado con Azure Container Instances. El contenedor se ejecuta en una máquina virtual de una red virtual.

**Aspectos que hay que saber sobre Azure Container Instances**

Veamos algunas de las ventajas de usar Azure Container Instances. A medida que revise estos puntos, piense en cómo puede implementar Container Instances para sus aplicaciones internas.

* **Tiempos de inicio rápidos**. Los contenedores pueden iniciarse en segundos sin necesidad de aprovisionar y administrar máquinas virtuales.
* **Conectividad con IP pública y nombres DNS**. Los contenedores se pueden exponer directamente a Internet con una dirección IP y un nombre de dominio completo (FQDN).
* **Tamaños personalizados**. Los nodos de contenedor se pueden escalar dinámicamente para que coincidan con las demandas de recursos reales de una aplicación.
* **Almacenamiento persistente**. Los contenedores admiten el montaje directo de recursos compartidos de archivos de Azure Files.
* **Contenedores de Linux y Windows**. Las instancias de contenedor pueden programar contenedores tanto de Windows como de Linux. Especifique el tipo de sistema operativo cuando cree los grupos de contenedores.
* **Grupos con programación compartida**. Container Instances admite la programación de grupos con varios contenedores que comparten recursos de máquinas host.
* **Implementación en red virtual**. Las instancias de Container Instances se pueden implementar en una red virtual de Azure.

**Implementación de grupos de contenedores**

El recurso de nivel superior de Azure Container Instances es el **grupo de contenedores**. Un grupo de contenedores es una colección de contenedores que se programan en la misma máquina host. Los contenedores de un grupo comparten un ciclo de vida, los recursos, la red local y los volúmenes de almacenamiento.

**Cosas que debe saber sobre los grupos de contenedores**

Veamos algunos detalles sobre los grupos de contenedores para Azure Container Instances.

* Un grupo de contenedores es similar a un pod en Kubernetes. Normalmente, un pod se asigna 1:1 con un contenedor, pero un pod puede contener varios contenedores. Los contenedores de un pod multicontenedor pueden compartir recursos relacionados.
* Azure Container Instances asigna recursos a un grupo multicontenedor agregando las solicitudes de recursos de todos los contenedores del grupo. Los recursos pueden incluir elementos como CPU, memoria y GPU.

Imagine un grupo de contenedores que tiene dos contenedores que requieren recursos de CPU. Cada contenedor solicita una CPU. Azure Container Instances asigna dos CPU para el grupo de contenedores.

* Hay dos maneras habituales de implementar un grupo multicontenedor: con plantillas de Azure Resource Manager (ARM) y con archivos YAML.
  + **Plantilla de ARM**. Se recomienda usar una plantilla de ARM para implementar otros recursos de servicios de Azure cuando se implementan instancias de contenedor, como un recurso compartido de archivos de Azure Files.
  + **Archivo YAML**. Dada la naturaleza concisa del formato YAML, se recomienda usar un archivo YAML cuando la implementación incluya únicamente instancias de contenedor.
* Los grupos de contenedores pueden compartir una dirección IP externa, uno o varios puertos de esa dirección IP y una etiqueta DNS con un FQDN.
  + **Acceso de cliente externo**. Debe exponer el puerto en la dirección IP y desde el contenedor para permitir que clientes externos lleguen a un contenedor del grupo.
  + **Asignación de puertos**. No se admite la asignación de puertos, porque los contenedores de un grupo comparten un espacio de nombres de puerto.
  + **Grupos eliminados**. Cuando se elimina un grupo de contenedores, se liberan su dirección IP y su FQDN.

**Ejemplo de configuración**

Vea el siguiente ejemplo de un grupo multicontenedor con dos contenedores.

El grupo multicontenedor tiene las siguientes características y configuración:

* El grupo de contenedores está programado en una sola máquina host y tiene asignada una etiqueta de nombre DNS.
* El grupo de contenedores expone una única dirección IP pública con un puerto expuesto.
* Un contenedor del grupo escucha en el puerto 80. El otro contenedor escucha en el puerto 1433.
* El grupo incluye dos recursos compartidos de archivos de Azure Files como montajes de volumen. Cada contenedor del grupo monta uno de los recursos compartidos de archivos localmente.

**Aspectos que deben tenerse en cuenta cuando se usan grupos de contenedores**

Los grupos multicontenedor son útiles cuando se quiere dividir una sola tarea funcional en varias imágenes de contenedor. Estas imágenes las pueden entregar diferentes equipos y pueden tener requisitos de recursos diferentes.

Tenga en cuenta los siguientes escenarios para trabajar con grupos multicontenedor. Piense en qué opciones pueden sustentar sus aplicaciones internas para el comerciante minorista en línea.

* **Considere las actualizaciones de las aplicaciones web**. Admita actualizaciones de las aplicaciones web implementando un grupo multicontenedor. Un contenedor del grupo sirve la aplicación web y otro contenedor extrae el contenido más reciente del control de código fuente.
* **Considere la recopilación de datos de registros**. Use un grupo multicontenedor para capturar datos de registros y métricas sobre la aplicación. El contenedor de la aplicación genera registros y métricas. Un contenedor de registro recopila los datos de salida y escribe los datos en el almacenamiento a largo plazo.
* **Considere la supervisión de las aplicaciones**. Habilite la supervisión de su aplicación con un grupo multicontenedor. Un contenedor de supervisión realiza periódicamente una solicitud al contenedor de la aplicación para asegurarse de que la aplicación se ejecuta y responde correctamente. El contenedor de supervisión genera una alerta si identifica posibles problemas con la aplicación.
* **Considere la funcionalidad de front-end y back-end**. Cree un grupo multicontenedor para albergar el contenedor de front-end y el de back-end. El contenedor de front-end puede servir una aplicación web. El contenedor de back-end puede ejecutar un servicio para recuperar datos.

**Revisión de la plataforma Docker**

Docker es una plataforma que permite a los desarrolladores hospedar aplicaciones en un contenedor. Un contenedor en Docker es básicamente un paquete independiente que contiene todo lo necesario para ejecutar un componente de software.

Docker Hub proporciona un gran repositorio global de imágenes de contenedor de desarrolladores, proyectos de código abierto y fabricantes de software independientes. Puede acceder a Docker Hub para buscar y compartir imágenes de contenedor para su aplicación y sus contenedores. Los hosts Docker son máquinas que ejecutan Docker y permiten ejecutar aplicaciones como contenedores.

En la siguiente ilustración se muestra cómo se comunica Docker Hub con el host Docker.

Diagrama

Descripción generada automáticamente

**Inserción de una imagen de contenedor en un repositorio de Docker**

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

<https://youtu.be/r_tGl4zF1ZQ>

**Cosas que debe saber sobre Docker y los contenedores**

Examine las siguientes características de la plataforma y los contenedores Docker.

* La plataforma Docker está disponible en Linux y Windows y se puede hospedar en Azure.
* Un paquete de contenedor Docker incluye el código ejecutable de la aplicación, el entorno en tiempo de ejecución, como .NET Core, las herramientas del sistema y la configuración.
* Un **Dockerfile** es un archivo de texto con instrucciones sobre cómo compilar una imagen de Docker. Dockerfile es como un script por lotes. La imagen base se muestra primero. El resto del archivo incluye las acciones de compilación.
* La principal característica de Docker es la garantía de que el software contenedorizado se ejecuta siempre igual de forma local en Windows o Linux, o en la nube de Azure.
* Desarrolle el código localmente en un contenedor Docker, comparta el código con recursos de Control de calidad para realizar pruebas e impleméntelo en producción en la nube de Azure. Una vez implementado el código, la aplicación se puede escalar fácilmente con Azure Container Instances.

**Aspectos que deben tenerse en cuenta al usar Docker**

Antes de empezar a usar Docker y Azure Container Instances para crear, compilar y probar contenedores, resulta útil estar familiarizado con la terminología y los conceptos.

* **Contenedor**: una instancia de una imagen de Docker. Un contenedor representa la ejecución de una sola aplicación, proceso o servicio. Está formado por el contenido de una imagen de Docker, un entorno de ejecución y un conjunto estándar de instrucciones. Al escalar un servicio, crea varias instancias de un contenedor a partir de la misma imagen. Un trabajo por lotes puede crear varios contenedores a partir de la misma imagen y pasar diferentes parámetros a cada instancia.
* **Imagen de contenedor**: paquete con todas las dependencias y la información necesarias para crear un contenedor. Las dependencias incluyen marcos de trabajo y la configuración de implementación y ejecución que usa un entorno de ejecución de contenedor. Normalmente, una imagen se deriva de varias imágenes base que son capas que se apilan unas encima de otras para formar el sistema de archivos del contenedor.
* **Compilación**: proceso de creación de una imagen de contenedor basada en la información y el contexto que proporciona el Dockerfile. La compilación también incluye cualquier otro archivo necesario. Las imágenes se compilan con el comando docker build de Docker.
* **Extracción** (pull): proceso de descarga de una imagen de contenedor de un registro de contenedores.
* **Inserción** (push): proceso de carga de una imagen de contenedor en un registro de contenedores.

**Prueba de conocimientos**

Está trabajando en la estrategia de implementación y administración de aplicaciones para el sitio de venta minorista en línea de su empresa. Está revisando algunos escenarios como parte del planeamiento de la implementación.

* El equipo de administración ha solicitado información sobre dónde puede ser más ventajoso tener contenedores de Windows en lugar de máquinas virtuales.
* Está examinando las características de Azure Container Instances para sustentar aplicaciones relacionadas con el sitio hospedadas en el entorno local en Azure, para compartir recursos de hardware, uso de red y volúmenes de almacenamiento.
* El equipo de desarrollo necesita una implementación de contenedores que pueda garantizar que el software se ejecute siempre igual de forma local y en la nube.

**Responda a las siguientes preguntas**

Elija la respuesta más adecuada para cada una de las siguientes preguntas. Después, seleccione **Comprobar las respuestas**.

Principio del formulario

**1. ¿Por qué debe seleccionar máquinas virtuales en lugar de contenedores para la configuración?**

Las máquinas virtuales ejecutan la parte del modo de usuario de un sistema operativo y se pueden personalizar para que contengan solo los servicios necesarios para la aplicación.

Las máquinas virtuales proporcionan un aislamiento completo del sistema operativo host y otras máquinas virtuales.

Las máquinas virtuales usan Azure Disks para el almacenamiento local de un único nodo.

**2. ¿Cuál de las siguientes opciones es una característica de Azure Container Instances?**

Las instancias de Container Instances requieren varios minutos para cargarse.

Las instancias de Container Instances usan Azure Blob Storage para recuperar y mantener el estado.

Container Instances se factura cuando los contenedores están en uso.

**3. ¿Qué implementación garantiza que el software contenedorizado se ejecute igual de forma local y en la nube de Azure?**

Docker

Grupos de contenedores

Container Instances

1. B
2. C
3. A

Final del formulario

**Resumen y recursos**

En este módulo, ha aprendido a identificar cuándo usar instancias de Azure Container Instances en lugar de máquinas virtuales de Azure. Ha explorado las características y los casos de uso de Azure Container Instances. Ha descubierto cómo implementar grupos de contenedores de Azure.

Las principales conclusiones de este módulo son:

* Los contenedores proporcionan aislamiento ligero y usan menos recursos del sistema en comparación con las máquinas virtuales.
* Los contenedores se pueden implementar individualmente mediante Docker o con un orquestador como Azure Container Apps.
* Los contenedores usan Azure Disks o Azure Files para el almacenamiento.
* Un grupo de contenedores es una colección de contenedores que se programan en la misma máquina host.
* Los contenedores se pueden volver a crear rápidamente en otro nodo de clúster si se produce un error en un nodo.